Bruno Nunes Brasil

DESCOBERTA AUTOMÁTICA DE PADRÕES A PARTIR DE REGISTOS DE AÇÕES COM FINS DE ANÁLISE ORGANIZACIONAL

UNIVERSIDADE DO ALGARVE

Faculdade de Ciências e Tecnologias

2021

Bruno Nunes Brasil

DESCOBERTA AUTOMÁTICA DE PADRÕES A PARTIR DE REGISTOS DE AÇÕES COM FINS DE ANÁLISE ORGANIZACIONAL

Mestrado Integrado em Engenharia Eletrónica e Telecomunicações

Trabalho efetuado sob orientação de:

Professora Doutora Marielba Silva de Zacarias

UNIVERSIDADE DO ALGARVE

Faculdade de Ciências de Tecnologias

2021

Descoberta automática de padrões a partir de registos de ações com fins de análise organizacional

Declaração de autoria de trabalho

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

_____

Dedicado à minha mãe, irmãos, sobrinho e namorada.

AGRADECIMENTO

A realização desta dissertação de mestrado contou com o apoio de inúmeras pessoas, demonstrado das mais variadas formas, e que contribuíram diretamente ou indiretamente para a conclusão deste trabalho.

Agradeço encarecidamente a Professora Doutora Marielba Silva de Zacarias todo o apoio, partilha do saber, críticas, ideias, revisão deste relatório e a grande disponibilidade em me orientar ao longo desta tese. Todos estes contributos foram essenciais para o enriquecimento e realização deste trabalho.

A todos os Professores do curso de Mestrado Integrado em Engenharia Eletrónica e Telecomunicações, por todo o apoio e partilha de conhecimento ao longo desta caminhada.

Aos meus amigos e colegas, André Francisco e João Gonçalves por todas as trocas de ideias, partilhas, dicas e momentos de descontração e amizade.

Agradeço a todos os meus colegas de curso, por todos os momentos de trabalho e diversão ao longo destes anos de vida académica.

Ao meu coordenador técnico João Ribeiro da empresa que trabalho, um agradecimento pela compreensão, apoio e flexibilização laboral.

Aos meus colegas de trabalho por todo o apoio e companheirismo.

Aos meus sogros e restante família, pelo apoio, compreensão e amizade. Foram uma segunda família para mim.

À minha mãe que lhe devo tudo, aos meus irmãos, sobrinho e cunhados que sempre estiveram presentes, mas de uma forma distante, pelo apoio incondicional, incentivo permanente e pelas condições que me facultaram a todos os níveis, sem dúvida foram um pilar fundamental neste meu percurso. Um muito obrigado.

À minha namorada, à "minha" Ana Sofia, por toda a paciência, apoio, compreensão, companheirismo, amizade e amor. Esteve ao meu lado em qualquer fase da minha vida.

RESUMO

A busca pelo aumento da procura de informação preciosa e o aumento do volume de dados não estruturados, faz com que se procurem soluções para o tratamento dessa informação. A integração de estratégias personalizadas de recolha de informação será essencial e é parte integrante nas soluções de onde resultará um aumento de eficiência e rapidez dos processos utilizados na indústria.

O elevado volume de trocas de informações digitais por parte das empresas, torna o acesso às mesmas essencial e de extrema importância.

Essa dificuldade pode ser compensada com a utilização de técnicas de mineração de dados e mineração de textos, que permitem a recolha de informação minimizando os custos e tempo despendido na procura de dados e padrões.

O presente trabalho consiste na deteção de textos e padrões de registos de ações provenientes de análises organizacionais. Para tal, foi tido em consideração os processos de agrupamento e de regra de associação, recorrendo a técnicas de pré-processamento de texto, processo de obtenção de agrupamento e classificação dos agrupamentos e regras de associação com recurso a utilização da ferramenta *Rapidminer Studio* ®.

Os modelos desenvolvidos permitiram obter simulações de cenários que considerem o agrupamento de diferentes ações desenvolvidas por cada ator sendo estas designadas por contextos pessoais, e o agrupamento de ações desenvolvidas por grupo de atores, designado por contextos interpessoais. Como forma de melhorar a análise interpessoal recorreu-se a outra técnica: a regra de associação para utilização nas ações de contexto interpessoais.

A utilização destas técnicas permitirá a avaliação de cada cenário utilizando parâmetros de similaridades e cálculos de distância entre conjuntos de dados.

PALAVRAS-CHAVE: Mineração de dados, Mineração de textos, Padronização de dados, Agrupamento, Regra de Associação, Classificação.

ABSTRACT

The search for an increase in the demand for precious information and an increase in the volume of unstructured data, makes it necessary to search solutions for the treatment of information. The integration of personalized information collection strategies will be essential and is an integral part of the solutions which will result in an increase in efficiency and speed of the processes used in the industry.

The high volume of exchanges of digital information by companies, makes access to them essential and extremely important.

This difficulty can be offset with the use of data mining and text mining techniques, which allow the collection of information minimizing costs and time spent searching for data and standards.

The present work consists of detecting texts and patterns of action records from organizational analyzes. For this, the grouping and association rule processes were taken into association, using text pre-processing techniques, the process of obtaining grouping and classification of clusters and association rules using the Rapidminer Studio ® tool.

The developed models allowed to obtain simulations of scenarios that consider the grouping of different actions developed by each actor, which are designated by personal contexts, and the grouping of actions developed by a group of actors, designated by interpersonal contexts. To improve interpersonal analysis, another technique was used: the association rule for use in interpersonal context actions.

The use of these techniques will allow the evaluation of each scenario using parameters of similarities and calculations of distance between data sets.

KEYWORDS: Data mining, Text mining, Clustering, Data standards, Association Rules, Classification.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CRISP-DM	*Cross Industriy Standard Process for Data Mining*
IA	Inteligência Artificial
KDT	*Knowlegde Discovery in Text*
IE	*Information Extraction*
NLP	*Natural Languague Processing*
YALE	*Yet Another Learning Environment*
TB	*Terabyte*
EUA	Estados Unidos da América
ID	*Identificaction*
FP-Growth	*Frequent Pattern - Growth*
ODM	*Organizational Data Mining*
TI	Tecnologia de informação
OT	*Organizational Data Mining*
VLC	Valor Líquido Contabilístico
IBM	*International Business Machines Corporation*
SEMMA	*Sample, Explore, Modify, Model e Assess*
KDD	*Knowledge-Discovery in Databases*

1. INTRODUÇÃO

1.1. ENQUADRAMENTO

A economia global atual está num nível competitivo tão elevado que requer capacidade de tomada de decisões no imediato [1]. Estudo realizado pela IBM (*International Business Machines Corporation*) indicam que as indústrias apenas analisam no máximo um por cento dos seus dados. No passado, as empresas lutavam para tomar decisões devido à falta de dados, mas atualmente, cada vez mais organizações estão trabalhando na superação da "paralisia de informações", há tantos dados disponíveis que é difícil determinar o que é relevante.

A deteção de um conjunto de dados "especiais" permite resolver problemas de negócio e ajudar na tomada de decisão, providenciando benefícios para uma empresa ou qualquer entidade [2]. Contudo, as atividades e operações digitais nas empresas, instituições públicas e privadas têm vindo a aumentar significativamente nas últimas décadas, o que origina grandes volumes de dados nas bases de dados, colocando desafios cada vez maiores no tratamento e análise desses dados. Em qualquer domínio, uma análise manual de um conjunto de dados é lenta, cara e altamente subjetiva.

Mais do que nunca o ser humano encontra-se na Era dos Dados. Estima-se que pode atingir só através da Internet, diariamente, e a nível global valores na ordem dos 2,5 milhões de TB de dados [3]. Compreende-se que é urgente a existência de métodos que tratem os dados de maneira produtiva para o utilizador. Por outro lado, as organizações procuram saber as opiniões que o consumidor tem sobre estas e recorrem muito aos utilizadores da internet. Um bom exemplo é poder ser demostrado o grande interesse em entender o que as pessoas sentem quando partilham a sua opinião na internet e no mundo das notícias.

A área de mineração de dados (*Data Mining*) providencia meios para contornar estes desafios colocados pela análise de grandes volumes de dados, através do desenvolvimento de técnicas eficazes para extrair conhecimento, relações e padrões, a partir de grandes volumes de dados extraindo informações chave para as instituições num curto espaço de tempo e com elevada fiabilidade [2][3]. Os domínios de *Data Science*, *Machine Learning* e *Artificial Intelligence* tem sido cada mais procurados e investigados para o melhoramento do rendimento dos lucros económicos [3]. Uma preocupação fundamental do *Data Mining* é a qualidade dos

dados e das técnicas de análise. Uma das consequências da utilização de dados sem qualidade para o utilizador/empresa é o prejuízo gerado, só nos EUA estima-se que atingem os 3,1 biliões de dólares/ano de prejuízo, um valor que tende a crescer proporcional com a criação de novos dados [3].

Em Portugal, a investigação sobre este tema também tem vindo a crescer, inclusive existem instituições de ensino universitário que se tem juntado a empresas para formarem profissionais nesta área de estudo. Empresas Portuguesas, nomeadamente no Algarve, começaram a recrutar Cientistas de Dados (*Data Scientists)* que tenham conhecimentos e experiência em aplicações de técnicas de mineração de dados e texto, que possuam um *background* sólido de conceitos de análise de dados e experiência na identificação de *insights* analíticos para o apoio à tomada de decisão em projetos de analítica e gestão de informação [3][4].

Uma vertente do *Data Mining* de desenvolvimento mais recente endereça a mineração de dados textuais (*Text Mining*). O *Text Mining* visa a extração de informações relevantes a partir de textos, transformando palavras, frases ou parágrafos registados em documentos de forma não estruturada, num modo apto para se poder aplicar técnicas de *Data Mining* [5].

Segundo [6], apesar do esforço crescente das organizações na utilização de técnicas de *Data Mining* como forma de manter a sua competitividade, 80% da informação armazenada nas organizações é textual. De acordo com o mesmo autor, este facto, embora difícil de constatar, abre novas vias de investigação, mas também coloca desafios pela dificuldade de processar e analisar informação textual. O *Text Mining* permite ultrapassar algumas destas dificuldades e está a começar a ser utilizado para responder a questões de pesquisa e análise organizacional até agora pouco investigadas, como examinar padrões num dado momento temporal ou investigar mudanças nesses padrões ao longo do tempo. Um exemplo do segundo tipo de pergunta é a análise da evolução de modelos de negócio a partir de relatórios anuais.

1.2. PROBLEMA

O potencial da aplicação de técnicas de *Data* e *Text Mining* nas organizações é um indicador da importância de encontrar ferramentas e definir metodologias que consigam dar resposta a questões para fins de análise organizacional. Um exemplo da utilização de técnicas automatizadas para fins de análise organizacional foi descrito em [7], o qual endereçou a análise de práticas de trabalho a partir de registos diários de ações em equipas de trabalho.

A autora definiu as práticas de trabalho como "padrões recorrentes de ação e interação entre trabalhadores", os contextos pessoais de ação como "agrupamentos de ações associadas a tópicos, recursos de informação e ferramentas semelhantes", os contextos interação interpessoal como "associações recorrentes entre dois contextos de ação pessoais de ação" e as redes de contextos de interação como as "redes geradas por estes contextos". A análise das práticas de trabalho baseou-se na descoberta e caracterização destes três conceitos, tendo explorado o suporte automático do primeiro tipo de contextos com recurso ao *Microsoft Analysis Services* ®. Contudo, a validação, assim como a descoberta e análise dos restantes contextos foi realizada de forma manual. O presente trabalho visa contribuir nesta linha de estudo continuando o trabalho iniciado por [7]. explorando a utilidade da ferramenta de acesso aberto *Rapidminer Studio* ®. Nomeadamente, este trabalho deverá responder ás seguintes questões:

- É possível reproduzir a descoberta automática de contextos de ação pessoal?
- É possível automatizar a descoberta de contextos de interação interpessoal?
- É possível descobrir as redes de contextos de interação geradas por estes contextos?

1.3. OBJETIVOS

Baseados na descrição do problema, os objetivos da dissertação passam por:

1. Estudar os conceitos, metodologias e técnicas de *Data Mining* e *Text Mining*;
2. Analisar e comparar algoritmos de *Text Mining* recorrendo ao software *Rapidminer Studio* ®;
3. Selecionar a metodologia e técnicas que melhor se adaptam para os fins pretendidos;
4. Explorar e avaliar a descoberta de: (a) contextos de ação pessoal, (b) contextos de interação interpessoal e (c) redes de contextos de interação.

1.4. ESTRUTURA DO RELATÓRIO

Este relatório está dividido e organizado em 5 capítulos:

No atual Capítulo 1 é apresentado o contexto, o problema e os objetivos do relatório.

No Capítulo 2 são abordados conceitos gerais e tecnologias associadas a Descoberta de Conhecimento de Bases de Dados, explicações de métodos de *Data* e *Text Mining*, explicação do conceito e algoritmos de *clustering*, análise e explicação sobre a regra de associação envolvendo os métodos de implementação e estudo da ferramenta *Rapidminer Studio* ® e a visualização e comparação dos resultados obtidos.

No Capítulo 3 é apresentada a metodologia de trabalho utilizada na preparação dos dados para análise e organização dos mesmos como também as configurações efetuadas na ferramenta.

O Capítulo 4 contém os critérios de avaliação, implementação e análise dos resultados.

No Capítulo 5 são apresentadas as conclusões e são explicitados quais serão os próximos passos e desenvolvimentos futuros relativos ao modelo proposto.

2. ESTADO DA ARTE

Neste capítulo começaremos por descrever a definição de *Data Mining* e de *Text Mining*, referindo os seus fundamentos, conceitos e aplicações. Em seguida será descrita a metodologia CRISP-DM (*Cross Industry Standard Process for Data Mining)*, que nos indicam os passos a seguir nos projetos de *Data Mining.* Depois serão abordados a metodologia de *Clustering*, *Associaction Rules* e os seus algoritmos associados, incluindo os mais populares e existentes na ferramenta *Rapidminer Studio ®.* Finalizaremos com uma breve análise sobre a ferramenta *Rapidminer Studio ®*.

2.1. MINERAÇÃO DE DADOS (*DATA MINING*)

Qualquer que seja a atividade humana, qualquer que seja a sua escala, esta produz recorrentemente dados eletrónicos, exemplo disso, é de uma pessoa ao comprar um bilhete na internet, são geradas e armazenadas em bases de dados as informações dessa compra [26].

Mais de 80% dos dados eletrónicos que existem atualmente são compostos por dados semiestruturados ou não estruturados. Com a aceleração de técnicas de aquisição de dados digitais, estes provocaram um enorme volume de dados. A realização de pesquisas convencionais torna a procura de termos familiares ao utilizador penosas e complexas, na grande maioria das vezes os resultados da pesquisa não são relevantes para os requisitos do utilizador.

O *Data Mining*, é referido como uma etapa da descoberta de conhecimento em dados e tem como principal objetivo encontrar as melhores e mais recentes informações através do estudo de enormes volumes de dados o que torna o conhecimento e as descobertas científicas mais poderosas [11]. A escolha da técnica correta e própria de mineração de dados é essencial para diminuir o tempo e o esforço e aumentar a velocidade necessários para extrair informações valiosas [8] [27].

O *Data Mining* recorre a diversos processos para analisar e investigar grandes quantidades de dados na procura de padrões, previsões, associações, erros, etc. Este permite realizar operações sobre os dados e que possibilitam categorizar e sumariá-los sobre as mais diferentes dimensões e vistas, segundo as suas relações. O *Data Mining* permite encontrar correlações,

associações, mudanças e anomalias que então implícitas nos dados, não estando percetíveis por uma simples observação humana [12] [13].

Tabela 2.1: Descrição das atividades de *Data Mining*, retirado de [8]

Atividade Preditiva	Classificação	Algoritmos Genéricos
		Redes Neurais
		Árvore de Decisão
		Classificação Baeysiana
		Lógica Fuzzy
		Análise de Vizinhança
	Regressão	Regressão Linear
		Regressão Multipla
		Regressão Não Linear
		Regressão Logistica
		Regressão Poisson
Atividade Descritivas	Regras de Associação	Estatística
		Teoria Conjuntos Aproximados
		Análise de Correlação
	Sumarização	Agregação
		Gráficos diversos
	Clustering	Redes Neurais
		Estatística
	Outras	

Conclui-se a partir da tabela anterior que a tarefa da mineração de dados usufrui de diferentes técnicas associadas, para [9] as mais famosas são: Árvore de Decisão, Agregação e Gráficos diversos, Análise de Vizinhança, Regressão Linear e Não Linear. As utilizações destas técnicas não funcionam só de forma independente, é possível combinar várias técnicas, inclusive a utilização dos dois modelos (preditivo e descritivo) em conjunto de forma a obter os melhores resultados.

2.1.1. METODOLOGIA DO CRISP-DM

Em meados dos anos 90, o *Data Mining* era uma área ainda pouco explorada, ainda numa fase embrionária, contudo ao longo dos anos foram feitos avanços nesta área e aplicados novos contornos ao nível da forma como foi sendo definida na sua documentação. Para isso, houve a necessidade da criação de um processo *standard* em que possibilitasse a partilha de dados gratuitamente, sem direitos de autor, sem interferência industrial e onde existisse a possibilidade de ajudar as indústrias no progresso dos projetos de *Data Mining* com as melhores linhas de orientação. A aplicação da metodologia CRISP-DM permite obter diversas vantagens, naturalmente torna a aplicação de projetos de *Data Mining* mais fácil de orientar, mais económica, mais acessível, e mais rápida. É totalmente independente da indústria e da ferramenta de *Data Mining*, o que torna idêntica à filosofia existente em outros ramos. Isto associado a utilização de forma independente do tipo de negócio (comércio, saúde, retalho, etc.) podendo ser utilizado em todas as ferramentas de *Data Mining* [10].

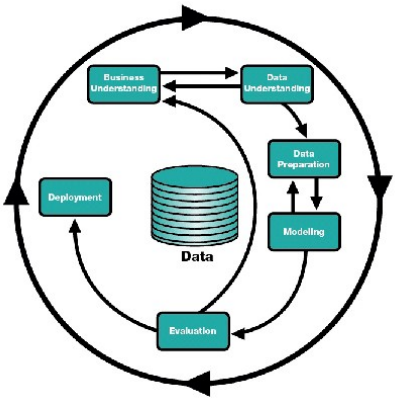


Figura 2.1: Fases do CRISP-DM, retirado de [9]

2.1.1.1. CRISP-DM (*CROSS INDUSTRY STANDARD PROCESS FOR DATA MINING*)

Através da figura 2.1, que são ilustradas as fases principais da CRISP-DM, analisamos em detalhe cada fase:

1. **Compreensão do Negócio (*Business Understanding*)**

Numa primeira abordagem a análise de negócio é verdadeiramente uma fase muito importante para todo o processo de *Data Mining*. Caso a análise inicial seja mal interpretada nesta fase, possivelmente acarretará resultados menos positivos para as restantes fases do processo, colocando em causa o sucesso de todo o projeto. Esta fase inicial divide-se em várias etapas: Começamos com a etapa da análise do problema do ponto de vista do utilizador ou negócio e a compreensão dos objetivos do ponto de vista do cliente, sendo que só uma compreensão realmente profunda do tema em questão permite compreender os objetivos do cliente/utilizador, e assim existe a possibilidade de avançar para a próxima fase. Esta fase permite também definir os objetivos do ponto de vista lógico, como é o caso de um projeto que pretende reduzir o abandono de clientes, por exemplo para uma loja de retalho o objetivo seria reduzir a perda de clientes em 20%. Após conhecido o negócio, passa-se à etapa de avaliação das condições para a execução do projeto, das quais se permite fazer uma avaliação dos recursos humanos e tecnológicos disponíveis, das informações disponíveis, etc. Resumindo, neste ponto é analisada a exequibilidade do que se pretende fazer. Feito isto, com a informação obtida é transformada num problema de *Data Mining* em que é definido o objetivo na visão do técnico, no caso em questão, por exemplo, modelar um modelo preditivo. Após a compreensão do problema com vista no negócio, da avaliação da viabilidade e definidos os objetivos a atingir, é assim elaborado um plano do projeto [11][12].

2. **Análise dos Dados (*Data Understanding*)**

A segunda fase é a análise dos dados, onde o utilizador conhece os dados que vai trabalhar. Fundamental compreender numa primeira fase – a colheita de dados iniciais – onde são feitos o carregamento de dados e a sua integração de forma uniforme que são provenientes de fontes diferentes. Em seguida, realiza-se a identificação dos dados, de uma forma superficial dos dados de modo a ficar a conhecer os mesmos. Assim temos a possibilidade de ficar a conhecer o tipo de dados, o formato, números de registos, quantidade de dados, e outros tipos de informação apelativas e pertinentes.

O mais determinante nesta etapa é perceber se os dados disponíveis contêm os requisitos para a modelação, ou seja, ficar a conhecer as caraterísticas dos dados. De seguida é realizada

a etapa de exploração dos dados, onde os mesmos são investigados de uma forma mais profunda, realizando pesquisas dos dados, visualizando relatórios para que possamos compreender as caraterísticas não superficiais. Aqui deve-se tentar descobrir de forma preliminar, padrões ou relações entre os dados. A última etapa consiste em estudar a qualidade dos dados, aqui são analisadas questões como a falta de valores a nulo e a branco. Também é analisado os valores possíveis para cada atributo, ou seja, se os valores presentes nos dados fazem sentido, como por exemplo se uma variável que indique o peso de uma pessoa seja em valores negativos. Ainda é estudado a existência de frases com o mesmo significado, a existência de *outliers* e valores que contrariem o senso comum como por exemplo jovens com rendimentos elevados, ou idades avançadas e fora do normal [11].

3. Tratamento dos Dados (*Data Preparation*)

Após a compreensão do negócio e de analisados os dados, agora é possível passar à terceira fase que é o tratamento dos dados. Aqui e após o utilizador ter adquirido conhecimento das fases anteriores, este utiliza o conhecimento para preparar os dados. Estes são sujeitos à ferramenta de modelação, ou seja, nesta fase existe o processamento de várias tarefas, das quais a seleção de dados, a limpeza dos dados, transformação, integração e formatação dos dados. (1) Seleção: consiste em utilizar os dados que serão de facto utilizados no projeto e retirado os que são desnecessários. Esta seleção deve ser bem justificada e documentada porque os dados introduzidos e as suas variáveis devem ser uteis para o projeto e evita-se volumetria em excesso o que pode tornar o processamento de informação lento. (2) Limpeza dos dados: Consiste em remover valores em falta e/ou valores em branco como por exemplo os *outliers,* nesta fase ocorre a verificação da qualidade dos dados. (3) Construção dos dados: São construídos registos completamente novos e/ou são gerados outros atributos derivados, estes podem ser transformados de maneira a adaptarem-se aos requisitos das ferramentas de *Data Mining*, um exemplo comum e muito utilizado é a transformação de atributos, quer sejam categóricos, numéricos ou binários que podem ser convertidos entre estes. (4) Integração dos dados: Este permite integrar os mesmos num só espaço, ou seja, os dados originários de fontes diferentes, que existem na mesma plataforma, por exemplo, uma empresa vende artigos num website de vários vendedores de vários armazéns, e até mesmo de vários países. Por último temos (5) Formatação dos dados: Quando ocorre, por alguma razão, existe a necessidade de formatar os dados existentes, como por exemplo, alterar o tamanho, remover algum carater ou número de maneira a tornar os dados viáveis [11][13].

4. Modelação (*Modeling*)

A fase de modelação deve iniciar-se após a fase de tratamento de dados, mas é possível voltar à fase anterior, ao voltar atrás este permite realizar novamente melhoramento nos dados com a possibilidade de melhorar os modelos gerados, sendo este passo importante nos projetos de *Data Mining.* É aqui que são selecionadas as diversas técnicas de *Data Mining* e que são capazes de lidar com os problemas colocados. Estas técnicas ao serem aplicadas e os parâmetros das mesmas definidos de forma a encontrar os melhores valores para o problema.

Nesta fase podemos caraterizá-los por quatro etapas que são: (1) Seleção das técnicas de modelação: esta constitui um modelo para selecionar as várias técnicas de *Data Mining* e as que melhor se adaptam para resolver problemas. (2) Geração do design de teste: Consiste na parametrização do método utilizado para testar os modelos após a sua utilização, o que nem sempre é trivial, após a criação dos modelos é essencial avaliar o desempenho e a qualidade das técnicas de *Data Mining*. Uma forma de prever, se a classificação dos dados tem um bom grau de fiabilidade é utilizar modelos de dados existentes já predefinidos e testados e comparar com dados incógnitos que servirão para prever o futuro. (3) Criação dos modelos: Esta etapa funciona com a execução dos modelos implementados. (4) Avaliação dos modelos: Aqui como o próprio nome indica o analista/investigador avalia os modelos implementados. Com o conhecimento do projeto e com os parâmetros definidos e as etapas anteriormente descritas bem executadas temos todas as condições para avaliar bem o projeto, atenção que esta tarefa deve ser realizada com auxilio de um especialista do projeto ou negócio em análise, de modo a avaliar com firmeza os resultados obtidos porque podem ocorrer problemas nos dados que não são identificados e que não sejam óbvios, com facilidade podem passar despercebidos para um utilizador comum. Nesta fase, também ocorre a classificação dos modelos usando as diferentes técnicas. Os modelos gerados são classificados gerando assim um ranking tendo em conta os critérios de avaliação definidos [10][11].

5. Avaliação (*Evaluation*)

A quinta fase é dedicada à avaliação, aqui é feita uma análise com maior qualidade do modelo escolhido e executado. Aqui é avaliado se são cumpridos todos os objetivos do negócio anteriormente definidos e se nenhum pormenor foi descartado.

No final desta fase, o analista deve decidir como usar os resultados recolhidos. A primeira tarefa é a avaliação dos resultados, nesta altura os resultados são comparados em relação aos objetivos de projeto e consegue-se verificar se o modelo implementado se adequa aos termos do projeto. Esta tarefa apesar de ser idêntica à fase de comparação e avaliação não deve ser

confundida com a fase de modelação, porque nesta fase os modelos são avaliados entre si, de acordo com as métricas implementadas, tais como a precisão dos modelos.

A seguir devemos fazer uma reavaliação, na qual é estudado se algum ponto técnico ou etapa realizada durante o processo foram descorados. Após a avaliação e da reavaliação, é chegada a altura de se definir as próximas tarefas. Deve-se decidir se terminamos o projeto e avançamos para o *deployment* da solução ou voltamos a efetuar novas iterações no modelo [10][11].

6. Instalação (*Deployment*)

Esta última fase é a instalação da solução implementada. O modelo final não representa o fim do projeto, estes resultados ainda devem ser organizados e apresentados ao cliente para que estes sejam compreendidos e corretamente utilizados.

Nesta fase é importante a elaboração de um resumo ou até mesmo de um relatório para o cliente, dependendo da dimensão do projeto em que é importante justificar todas as opções escolhidas e os resultados obtidos, esta etapa é importante porque permite que o processo de *Data Mining* possa ser repetido para toda a empresa. Na maior parte das vezes é o cliente a elaborar essa tarefa.

É importante que o analista familiarize o cliente com as tarefas a elaborar, possibilitando a aprendizagem da utilização correta dos modelos. Para isso, existem algumas etapas que devem ser importantes executar após a obtenção do projeto. As tarefas a executar na fase de instalação são: (1) planificação da instalação em que é planeado todo o processo de instalação; (2) planeamento da monitorização e manutenção dos modelos criados, esta é uma etapa com um elevado grau de importância, se os resultados são planeados para usar diariamente no cliente; (3) relatório final, após as duas etapas de planeamento deve ser realizado o relatório. O relatório poderá ter como conteúdo apenas um resumo do projeto com algumas notas de interesse como pode ser um relatório detalhado com uma exposição exaustiva dos resultados obtidos. A última etapa e não menos importante, é a revisão do projeto, onde é feita uma reflexão de todo o processo e é avaliada os pontos positivos e negativos do projeto, bem como os pontos a melhorar para projetos futuros [11][13].

Tabela 2.2: Subfases da metodologia do CRISP-DM, retirado de [10]

Fases	○ Subfases
Análise do Negócio (Business Understanding)	○ Análise do problema do ponto de vista funcional ○ Avaliação da situação ○ Definição dos objetivos de *Data Mining* ○ Planeamento do projeto
Análise dos Dados (Data Understanding)	○ Recolha de dados iniciais ○ Descrição dos dados ○ Exploração dos dados ○ Qualidade de dados
Tratamento dos Dados (Data Preparation)	○ Seleção de dados ○ Limpeza de dados ○ Transformação de dados ○ Integração dos dados ○ Formatação de dados
Modelação (Modeling)	○ Seleção das técnicas ○ Planificação da avaliação ○ Modelação ○ Avaliação
Avaliação (Evaluation)	○ Avaliação dos resultados ○ Reavaliação dos resultados ○ Decisão das próximas tarefas
Instalação (Deployment)	○ Planeamento da Instalação ○ Planeamento da monitorização e manutenção ○ Realização do relatório final ○ Revisão do projeto.

2.1.2. TÉCNICAS DE DATA MINING

2.1.2.1. CLASSIFICAÇÃO

A classificação estuda os vários atributos e elabora descrições das suas características em cada uma das classes. A aplicação dos vários métodos de classificação é feita em várias áreas, tais como: Análise de crédito, ações de Marketing, deteção de fraudes, telecomunicações, segmentação de clientes, modelagem de negócios e em diagnósticos médicos [14].

A tarefa de previsão consiste na análise de características presentes nos dados X e atribuir uma classe já definida Y, ou seja, consiste na identificação da classe e a que registo a mesma pertence, como representado na seguinte figura 1. Um bom exemplo do explicado acima é: Identificar se uma pessoa é boa ou má pagadora ou se normalmente tem rendas baixas, médias ou altas. Dando assim origem à resposta da pergunta: Deve ser concedido um crédito ou não? [15].

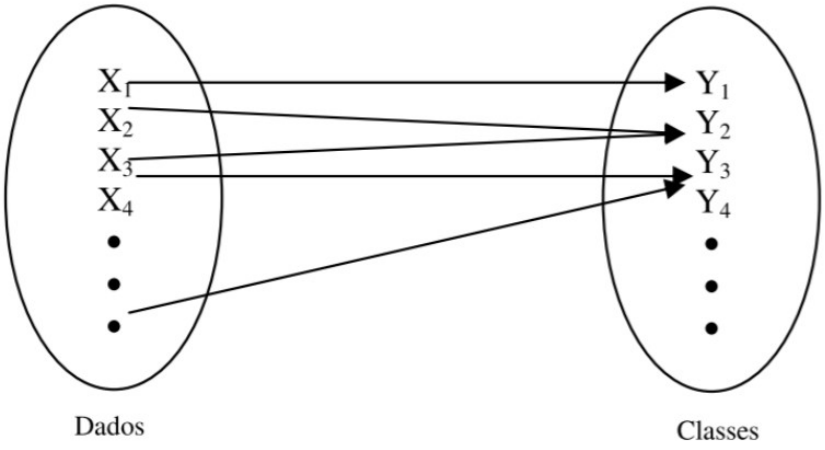


Figura 2.2: Relação entre os dados e as classes, retirado de [12]

2.1.2.2. REGRESSÃO

Segundo [16], a regressão é uma função que liga um determinado item a uma variável real estimada. Pode ser usada para classificar, acreditando-se que diferentes valores de dados podem corresponder também a diferentes classes.

Utiliza-se a regressão para definir valores a variáveis desconhecidas. Ou seja, para determinar valores que acontecerão, mas ainda não se tem dados quantitativos suficientes. Como por exemplo estimar o tempo de vida de um cliente, ou estimar a receita total de uma família baseando-nos nos ordenados recebidos até então ou estimar a probabilidade de um paciente com base no seu historial médico [5][17].

Outra forma de regressão, só que com a única diferença que o atributo é um valor numérico e não categórico. Alguns exemplos: Previsão do tempo ao longo da semana, prever o VLC (Valor Líquido Contabilístico) de um equipamento, prever um desempenho de um aluno, entre outros.

A regressão pode ser classificada como linear e não linear. Segundo [18], numa regressão linear as variáveis X e Y possuem uma relação linear entre si como o próprio nome indica, existe sempre uma variável dependente e uma variável independentemente. Por exemplo: Se eu trabalhar mais horas receberei mais ordenado, se eu estudar mais terei mais rendimento escolar. No caso da regressão não linear, é quando as variáveis X e Y não seguem um comportamento linear como se pode ver no gráfico, representado por uma função polinomial. Por exemplo: Aumento horas de estudo, mas nem sempre consigo obter melhores notas.

A figura 2.3 mostra um exemplo de gráfico regressão linear

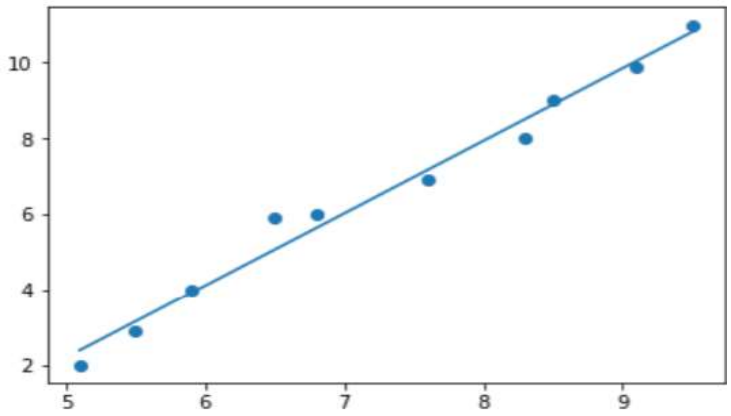


Figura 2.3: Exemplo de uma regra de regressão linear, retirado de [17]

A figura 2.4 mostra um exemplo de gráfico regressão não linear

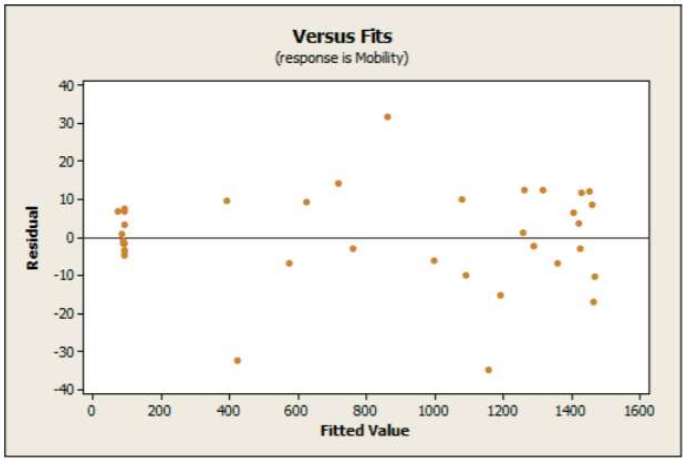


Figura 2.4: Exemplo de uma regra de regressão não linear, retirado de [18]

2.1.2.3. ASSOCIAÇÃO

Segundo [19], as regras de associações anseiam por encontrar semelhanças entre os registos, essas mesmas semelhanças são expressas por regras. A associação é algo bem definido, determinístico e relativamente simples. No entanto, ao contrário da classificação não envolve a previsão. A associação permite encontrar como que uma causa-efeito, ou seja, procura relacionar um conjunto de itens de dados com a ocorrência de outros itens de dados.

2.1.2.4. AGRUPAMENTO (*CLUSTERING*)

O agrupamento é um conjunto de categorias ou agrupamentos para descrever os dados. Este identifica os elementos de uma classe que tem mais similaridade entre si e muitas diferenças relativamente a outros elementos [21].

O *Clustering* difere da classificação porque não necessita que os dados sejam classificados. Segundo [16], o agrupamento não prevê dados, apenas agrega grupos de dados iguais, como é demonstrado na figura 6, como por exemplo: o agrupamento de clientes por região ou com comportamento de compras parecidas, entre outros.

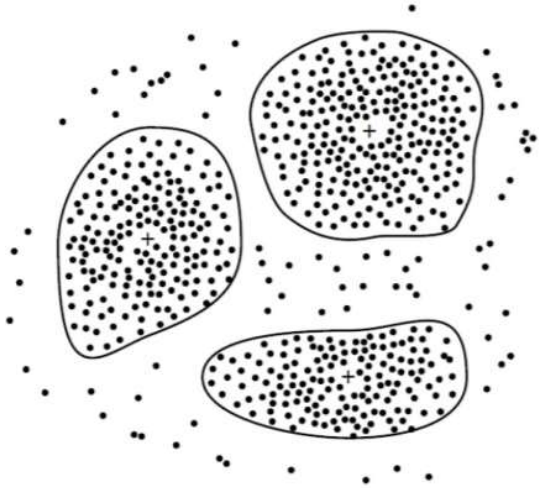


Figura 2.5: Agrupamento com formação de três clusters, retirado de [16]

2.1.2.5. SUMARIZAÇÃO

A sumarização permite descrever compactamente um subconjunto de dados. Consiste em sumarizar (identificar e indicar) similaridades entre conjuntos de dados, exemplos da aplicação destes métodos são a utilizam da média e do desvio padrão para todos os campos. Em métodos mais sofisticados utilizam a derivação de regras de resumo, técnicas de visualização multivariada e a descoberta de relações funcionais entre variáveis. As técnicas de resumo são frequentemente aplicadas à análise exploratória interativa de dados e à geração automatizada de relatórios [19].

2.2. MINERAÇÃO DE TEXTOS (*TEXT MINING*)

Nesta secção iremos abordar a temática do *Text mining* que é uma área de desenvolvimento mais recente, cujo objetivo principal é a análise de textos, e que abrange a extração de informações relevantes a partir de textos, transformando palavras, frases ou parágrafos não estruturadas num modo apto para se poder aplicar técnicas de *Data Mining*.

Os textos são vulneráveis ás mais diversas interpretações, quer pela gíria das palavras, quer seja pela linguagem especifica do utilizador/indústria, quer pelo idioma utilizado, estes podem sofrer várias interpretações [14].

Tanto a mineração de texto como a mineração de dados diferem no tipo de dados em que estes trabalham, a mineração de dados trabalha com dados estruturados, por sua vez a mineração de texto trabalha com dados não estruturados ou dados semiestruturados [26].

Os textos são vulneráveis ás mais diversas interpretações, quer pela gíria das palavras, seja pela linguagem especifica do utilizador/industria, quer pelo idioma utilizado, estes podem sofrer várias interpretações [14].

2.2.1. DESCOBERTA DO *TEXT MINING*

O aparecimento do conceito de *Text Mining* ou KDT (*Knowledge Discovery from Text*) surgiu pela necessidade organizacional existente na carência de se catalogar livros utilizando a sua classificação e sumarização, uma tarefa importante numa biblioteca [23]. Para [23] o primeiro registo remonta a Universidade de Oxford em 1674 onde foi catalogado livros e atribuídos à *Bodelian Library*, mas foi dois anos mais tarde que os cartões de indexação apareceram para originar a catalogação de bibliotecas com cartões.

O enorme avanço da técnica de *Text Mining* aconteceu em 1898 através de uma cooperação conjunta entre a Sociedade Física de Londres e a Instituição de Engenheiros Elétricos. Com as primeiras tentativas de resumir grandes corpos de texto em resumos científicos, um enorme passo no desenvolvimento do processamento de texto que permitiu a sumarização resultando nos resumos.

O primeiro computador utilizado, um IBM 701, para produzir resumos de documentos foi implementado por *Luhn* em 1958, este computador analisava a frequência de palavras [20].

Nos dias de hoje, uma enorme quantidade de informação do governo, indústria, empresas e outras instituições são guardados eletronicamente na forma de banco de dados de texto virtuais [21].

Figura 2.6: Catálogo de cartões da biblioteca e cartão de índice, retirado de [23]

O século atual levou-nos além das limitações da quantidade de informação na web o que tornou as informações com maior conscientização e melhor conhecimento. De uma maneira geral o *text mining* gerou o processo de conhecimento interessantes e não triviais de documentos de texto ou processo de extração de padrões.

Um enorme desafio do processo da mineração de texto é encontrar o conhecimento preciso em determinados documentos de texto e ajudar os usuários a encontrar o que realmente procura. Resume-se que o *Text Mining* é uma variação do campo da *Data Mining* que procura descobrir padrões de elevado interesse nos enormes bancos de dados [22].

O seguinte diagrama de Venn mostra o *Text Mining* e a sua interação com outros campos do universo do tratamento da informação.

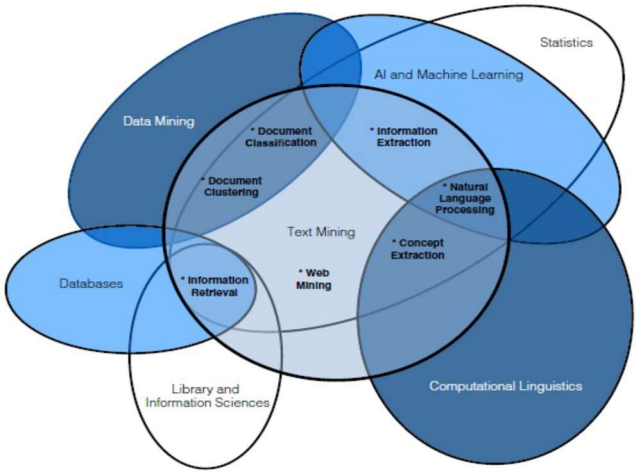


Figura 2.7: Interação do *Text Mining* com outros campos, retirado de [26]

O *Text Mining* encontra-se no centro do processo de conceção da informação, este consegue ter interação com o *Data Mining*, bases de dados, Inteligência Artificial, *Machine Learning*, estatísticas, linguística computacional e biblioteca e informação científica.

2.2.2. TÉCNICAS DE *TEXT MINING*

2.2.2.1. PROCURA/SELEÇÃO DE INFORMAÇÃO (*INFORMATION RETRIEVAL*)

O *Text Mining* e a recuperação de informação nos dados textuais estão inteiramente relacionadas. A recuperação de informação permite fazer extração de padrões associados e relevantes de acordo com um determinado conjunto de palavras ou frases.

As páginas web de pesquisa de informação usam o sistema de recuperação de informação com mais incidência para extrair informações relevantes, estes utilizam algoritmos apropriados em consultas para rastreio de resultados com propensão a informações mais significativas. O utilizador consegue obter informação de maior interesse e adequadas para as necessidades pedidas com recurso a motores de busca consequentemente este obtém as informações relevantes da coleção [23].

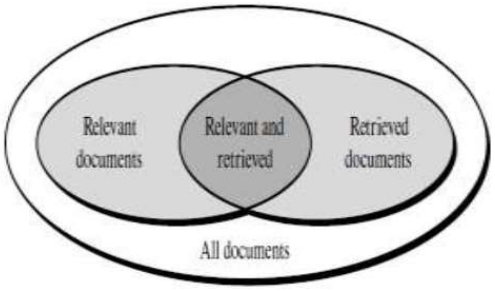


Figura 2.8: Ilustração de recolha de informação relevante, retirado de [24]

A Recuperação de informações é um problema de pesquisa, em que o utilizador extrai um certo tipo de informações mais especificas (ad hoc). Vejamos o seguinte exemplo: Um usuário quer comprar um carro usado, ao procurar por informação fá-lo logo por "Carro Usado". Automaticamente, irão apenas aparecer-lhe carros usados e não todo o género de carros.

Ou quando um utilizador pesquisa por informação a longo prazo, o sistema pode ter a iniciativa de lhe incutir mais informação, se as mesmas forem relevantes [21].

2.2.2.2. EXTRAÇÃO DE INFORMAÇÃO (*INFORMATION EXTRACTION*)

Esta é a etapa inicial para um computador examinar atributos e entidades específicos de textos ou documentos não estruturados chamados de extração de informação, este método identifica as frases-chave e relações no texto. A extração de informação abrange a tokenização, identificação de entidades projetadas, segmentação de sentenças e atribuição de parte do discurso.

Inicialmente as frases e sentenças são analisadas e semanticamente executadas e logo depois as informações necessárias são colocadas no banco de dados [22][23].

O processo de extração de informações gerais é como mostrado na figura 2.9 [25].

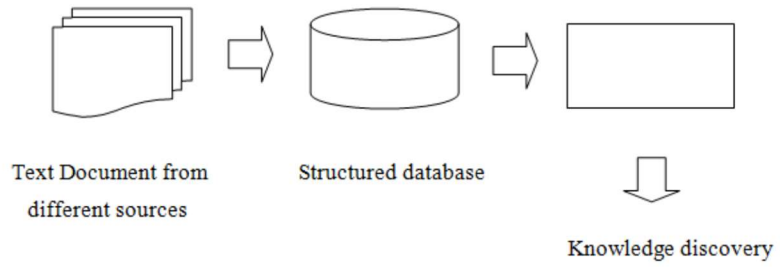


Figura 2.9: Procedimento de Extração de Informação

A extração de informação utiliza relações dentro do texto, simplesmente ocorre a procura de padrões, transformações dos dados disponíveis em forma de tabelas que transformam em texto semiestruturado podendo depois extrair para o conhecimento [24].

2.2.2.3. PROCESSAMENTO DE LINGUAGEM NATURAL (*NATURAL LANGUAGE PROCESSING*)

O Processamento de Linguagem Natural tem o propósito de analisar a linguagem de voz ou escrita com a intenção de ser modificada para uma apresentação mais objetiva para ser mais facilmente manipulada por máquinas virtuais.

As máquinas atualmente conseguem analisar melhor os grandes volumes de dados fundamentados em linguagem do que os próprios seres humanos, sem presença de cansaço, inconsistente e imparcialmente.

O estudo da linguagem humana é complexo e diversificado. Quando expressamos das mais variadas maneiras existem muitas interpretações tanto para a máquina como para o utilizador, por existir inúmeros idiomas e dialetos, como também existem variadas regras gramaticais e de sintaxe, expressões e gírias expressadas por cada ser humano. Na escrita do utilizador poderá ocorrer erros de caligrafia, presença de abreviaturas nas palavras ou ausência de pontuação.

A interpretação das palavras também pode ser dificultada quando exprimimos informações, suportamos sotaques locais, estendemos murmúrios e utilizamos termos de outras línguas.

O processamento de linguagem natural é essencial porque auxilia a resolver a incerteza na linguagem e acrescenta uma estrutura numérica útil aos dados para muitas aplicações de *downstream*, como identificação da língua ou interpretação de texto [25][26].

Figura 2.10: Visão geral do processamento de linguagem natural, retirado de [30]

O processamento de linguagem natural abrange diversos tarefas piloto como por exemplo a análise sintática, *stemming*/lematização, rotulagem dos componentes do discurso, deteção de idiomas e relacionamento de semânticas.

Num modo global as tarefas de processamento de linguagem natural dividem a linguagem em segmentos menores e essenciais para formar relações que colocam as semelhanças a funcionar juntos de maneira a gerar significado.

Existem tarefas implícitas que são maioritariamente usadas em níveis do processamento de linguagem natural com um grau de complexidade superior, como os exemplos a seguir:

Extração Contextual: Em que por norma extrai autonomamente informações estruturadas de origens de texto.

Categorização de conteúdo: Este tem por intuito resumir texto com base na linguística, que engloba pesquisas e indexação, alertas de conteúdos e deteção de repetições.

Descoberta e modulagem de tópicos: Faz a captura com precisão sobre o significado e os tópicos em coleções de texto, este método aplica *advanced analytics* como *forecasting* e otimização.

Tradução de máquina: Traduz a voz e o texto de vários idiomas de forma autónoma.

Análise de sentimentos: Captura a realidade da situação ou opiniões subjetivas de grandes volumes de texto, que incluem a mineração de opinião e o sentimento médio.

Conversão voz-texto e texto-voz: Altera comandos de texto em voz ou vice-versa.

Sumarização: Cria resumos de grandes volumes de texto automático.

Em todas estas situações o principal objetivo ambicionado é colocar os grandes volumes de texto e usar técnicas algorítmicas e linguísticas de forma a obter um rico conjunto de sentenças de modo a obter ótimos resultados.

O estudo do texto é aproveitado para explorar textos e procurar novas variáveis de texto em bruto, que podem ser visualizadas, filtradas ou usadas como entradas para os modelos preditivos ou outros métodos estatísticos.

Do modo geral o processamento de linguagem natural ou o estudo do texto pode ser usado nas seguintes aplicações:

Conhecimento Especializado: Categoriza conteúdos em temas significantes para que se possa tomar decisões acertadas e descobrir novos modelos.

Análise de mídias social: Investiga a relevância e o sentimento sobre informações especificas e identifica *influencers*.

Descoberta investigativa: Procura padrões e pistas em documentos ou correio eletrónico que possam ajudar a detetar e resolver crimes [26].

2.3. O *DATA MINING* E O *TEXT MINING* NAS ORGANIZAÇÕES

As organizações que tomam decisões rápidas e baseadas em fatos, otimizando seus recursos de dados, terão um desempenho melhor do que as organizações que não o fazem [1]. Uma tecnologia robusta que facilita este processo de tomada de decisão ideal é o *Organizational Data Mining*, este elimina as suposições e permeiam grande parte da tomada de decisões corporativas. Ao adotar o *Organizational Data Mining*, os gerentes e funcionários de uma organização são capazes de agir mais cedo ou mais tarde, ser proativos ao invés de reativos e saber ao invés de adivinhar. O *Organizational Data Mining* abrange uma ampla gama de tecnologias, incluindo, mas não se limitando a, *e-business intelligence*, análise de dados, OLAP, CRM, EIS, painéis digitais, portais de informação, etc. ODM permite que as organizações respondam a perguntas sobre o passado (o que aconteceu), o presente (o que está acontecendo) e o futuro (o que pode acontecer). As organizações podem gerar conhecimento valioso a partir de seus dados, o que por sua vez, aprimora as decisões corporativas. Esta tecnologia de aprimoramento de decisão permite muitas vantagens em operações (desenvolvimento mais rápido do produto, maior participação de mercado com tempo de entrada mais rápido no mercado, melhor gestão da cadeia ideal de suplemento). Os elementos da *Organizational Data Mining* que podem ser fundamentais são categorizados em três campos principais: Inteligência artificial, tecnologia de informação e teoria organizacional. A principal diferença entre *Organizational Data Mining* e mineração de dados é a teoria organizacional.

Para atingir esses objetivos, a pesquisa em teoria organizacional sugere que as organizações usam dados em três atividades vitais de criação de conhecimento.

As três atividades de criação de conhecimento são:

- Criação de sentido é a capacidade de interpretar e compreender informações sobre o ambiente e eventos que acontecem dentro e fora da organização.

- A produção de conhecimento é a capacidade de criar novos conhecimentos combinando a experiência dos membros para aprender e inovar.

- A tomada de decisão é a capacidade de processar, analisar informações, conhecimento para selecionar e implementar o curso de ação apropriado.

As tecnologias da *Organizational Data Mining* fornecem a essas organizações de classe mundial maiores oportunidades de entender seus negócios e tomar decisões informadas. A *Organizational Data Mining* também permite que as organizações de classe mundial estimulem os seus recursos internos de forma mais eficiente e eficaz. O número de projetos da *Organizational Data Mining* deve crescer mais de 300% na próxima década. Como a coleta, organização e armazenamento de dados a aumentar rapidamente, a *Organizational Data Mining* será o único meio de extrair conhecimento oportuno e relevante de grandes bancos de dados corporativos [1].

O *Text Mining*, ajuda a acelerar a descoberta do conhecimento das organizações, aumentando radicalmente a quantidade de dados que podem ser analisados. O conhecimento é derivado de padrões e relacionamentos e pode ser usado para revelar fatos, tendências ou construções. Uma técnica relacionada com a qual os investigadores organizacionais podem estar mais familiarizados é a análise de texto assistida por computador. Até o momento, a maioria dos estudos que empregam análise de texto na pesquisa organizacional são baseados na análise de texto assistida por computador, que é uma classe especial do *Text Mining*. Enquanto a maioria destes procedimentos extraem padrões contando frequências de palavras/termos, o *Text Mining* geralmente também capitaliza outras propriedades textuais, como gramática e estrutura, é aplicado técnicas de processamento de linguagem natural, linguística computacional, linguística de corpus, aprendizado de máquina e estatística [27].

2.4. TÉCNICAS DE PRÉ-PROCESSAMENTO DO *TEXT MINING*

Para fazer a mineração ou extração da informação em grandes coleções de textos são necessários o pré-processamento de documentos de texto e o armazenamento dessas informações sob a forma de dados estruturados, mais fáceis de processar do que um ficheiro de texto. A etapa do pré-processamento de texto também chamada etapa de preparação do texto, visa essencialmente fazer a remoção dos dados desnecessários para o entendimento do texto e extração do conhecimento. O processo de preparação do texto inclui um conjunto de ações sobre um texto, nomeadamente: correção ortográfica, remoção de *stopwords*, lematização, definição de *n-grams*, cálculo de peso e seleção de palavras-chave.

2.4.1. TOKENIZE

É uma técnica relativamente simples, que permite a divisão do texto em *tokens* individuais, as frases de um documento serão dividas em *tokens* removendo os espaços. Esta técnica serve de entrada para os próximos processos do método de *Text Mining*. [28].

A ferramenta *Rapidminer* ® oferece três formas de divisão: um é o padrão que é geralmente usado e é chamado de não carateres, onde se divide com base da existência de espaços, vírgulas, pontos finais etc. O segundo modo é especificar caractere, em que podemos especificar carateres que pretendemos dividir a frase em *tokens* e o terceiro é expressão regular, em que a expressão regular é fornecido para dividir a frase em *tokens* [28][29].

2.4.2. REMOÇÃO DE *STOPWORDS*

Esta técnica serve para reduzir a dimensão de um conjunto de documentos e o tamanho do dicionário de palavras que descrevem os documentos de *stemming* [30].

O método de remoção de *stopwords* ou remoção de *stoplist,* não é mais que a remoção de artigos, preposições, conjunções, etc. Palavras que são comuns e que se repetem diversas vezes num texto e que não produzem informações relevantes como por exemplo “o”, “e”, “mas”, “para”, ou “a”, entre outras. Com isto temos a possibilidade de reduzir a dimensão dos documentos indexados, ficando apenas nos documentos expressões que são relevantes para a extração de informação [31].

2.4.3. LEMATIZAÇÃO OU *STEMMING*

A lematização ou *Stemming* é outro método que pode ser utilizado no pré-processamento e que permite diminuir a lista de palavras presentes num documento. A remoção ocorre por palavras do tipo plural, prefixos, sufixos, género, número e gerúndio de modo que a respetiva palavra contenha apenas a *stem* (raiz) [32].

Como mostra a figura 2.11 podem existir várias palavras que mantem o mesmo significado variando apenas a forma do tempo presente [28].

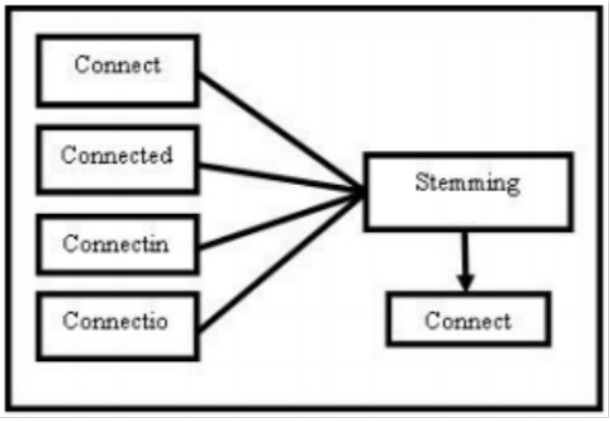


Figura 2.11: Processo de Lematização ou *Stemming*

2.4.4. GENERATE N-GRAMS

Quando se analisa uma palavra isoladamente por vezes perde-se o significado da informação. Para isso existe uma solução que passa pela aplicação deste método do *n-grams.* Esta é uma sequência de Carateres ou Termos consecutivos existindo em um determinado tamanho que podem ser de um até ao infinito. Com isto podemos identificar na linguagem significados da utilização de sequências. No pré-processamento de texto é utilizado para reorganizar um texto lematizado e sem *stopword* de modo a originar um novo texto com alguma sequência lógica compreensível [32]. Podemos assim definir numa amostra de palavras, se pretendemos utilizar um determinado número de termos da amostra ou utilizar um determinado número de carateres.

2.4.5. TRANFORM CASES

A técnica de *transform cases* não é mais do que transformar todas as palavras em maiúsculas ou minúsculas, técnica esta chamada de *lower case* e *upper case*. Isto permite que palavras com o mesmo significado e escrita que contenham carateres maiúsculas e minúsculas sejam reconhecidas como *tokens* diferentes, assim faz aumentar o vocabulário com repetições de palavras com o mesmo significado. Esta técnica permite reduzir o número de *features*. Esta etapa é importante para normalizar o texto [34].

2.5. *CLUSTERING*

O *Clustering*, em inglês, ou o agrupamento é uma das tarefas gerais para sintetizar um conjunto de documentos. O *Clustering* serve, nada mais nada menos, para classificar documentos, agrupando os dados e formando grupos de dados que tenham semelhança entre si. Este processo de agrupamento faz a recolha nos vários documentos termos ou padrões semelhantes. Para isso, recorre-se a medidas de similaridade que calculam as distâncias entre esses mesmos termos ou padrões, tais como: *euclidian*, *manhattan*, *chebychey*, entre outros.

Para se obter esses resultados de similaridade, pode ser necessário recorrer a mudanças nos dados. Quer sejam ordinais, categórigos, binários, ou para uma medida padrão como por exemplo na escala [0.0, 1.0].

A utilização do método de *Clustering* é um processo que envolve otimização multi-objetivo que envolve tentativa e falhanço [22][23][33].

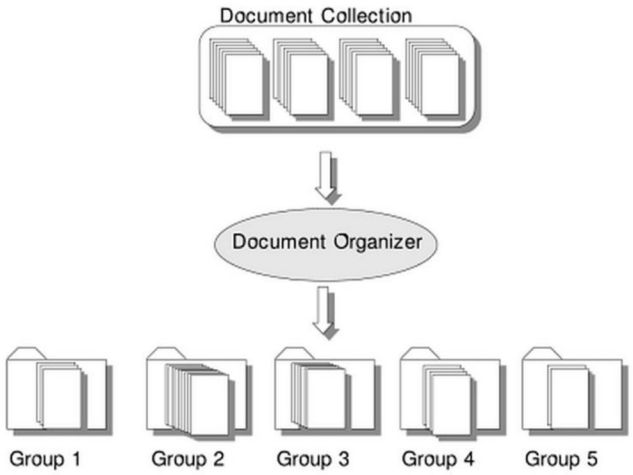


Figura 2.12: Método de *Clustering*, retirado de [34]

2.5.1. MÉTODOS DE *CLUSTERING*

2.5.1.1. MÉTODOS HIERÁRQUICO (*HIERARCHICAL METHOD*)

O método hierárquico não exige estabelecer um número inicial de clusters porque são vistos como inflexíveis e assim não se consegue trocar um elemento de cluster. Estes tipos de métodos podem ser classificados em aglomerativos e divisivos.

Nos métodos aglomerativos, por norma, os elementos começam separados e vão se agrupando em etapas, passo a passo terminando num único cluster com todos os elementos. A constituição do número ideal de clusters é interpretada consoante as opções existentes no processo de cluster. A similaridade é calculada entre um determinado agrupamento e os restantes agrupamentos, o processo termina quando existe apenas um agrupamento principal, assim podemos definir este método como uma estratégia *bottom-up*.

Os algoritmos que utilizam esta estratégia são o AGNES (*AGlomerative NESting*) e a CURE (*Clustering Using REpresentatives*), no *software Rapidminer Studio* ® existe o *Agglomerative Hierarchical Clustering* [16][40].

Já nos métodos divisivos, os elementos têm o seu início num único núcleo, ou seja, estão juntos num único cluster e a medida que o processo vai sendo desenvolvido estes vão se separando um a um até que cada elemento esteja no seu próprio cluster.

Conclusão, nos métodos divisivos o utilizador deve escolher o melhor número de clusters de forma a obter a melhor combinação possível e o pretendido pelo mesmo, estes inicialmente estão todos agrupados e no desenrolar do processo vão acontecendo divisões até que cada objeto represente um agrupamento. O algoritmo que utiliza esta estratégia é o DIANA e no *software Rapidminer* ® não existe algoritmo associado a esta técnica [16] [40].

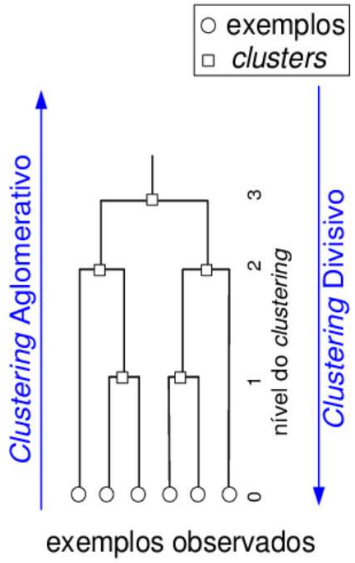


Figura 2.13: Dendograma do método hierárquico, retirado de [41]

2.5.1.2. MÉTODO DE PARTICIONAMENTO (*PARTITIONING METHODS*)

Este método tem a ação de recorrer a um número n de registos de uma base de dados, partindo do pressuposto que se inicia com um valor K de *clustering* a fim de se obter várias partições de dados, onde cada partição representa um grupo de *clustering*. De início é configurado um número de grupos que pretendemos utilizar, valor esse indicado por K, os registos são agrupados numa primeira interação e posteriormente reagrupados novamente de forma interativa. Alguns algoritmos existe uma função de custo, que ajuda a determinar quando o algoritmo deve determinar o seu processo.

Um critério de avaliação para verificar a qualidade dos clusters formados é a verificação através da aproximação e distanciamento dos registos, esta análise é realizada pelas medidas de distância. Os algoritmos mais utilizados por este método de particionamento é o *K-means* e o *K-medoids* [16][40].

2.5.1.3. MÉTODO BASEADO NA DENSIDADE (*DENSITY-BASED METHODS*)

Este método tem por objetivo a ocorrência de um agrupamento crescente até atingir um limite ou "excesso" de um determinado raio. Para este caso os dados são obrigados a encontrarem-se dentro de um determinado raio. Os métodos baseados em densidade por norma regem-se pela deteção de valores discrepantes num conjunto de dados. Para este método existe alguns algoritmos que utilizam este processo de recolha de dados relevantes e que usam a densidade, tais como o algoritmo DBSCAN (*Density Based Spatial Clustering of Applications with Noise*) [35], este recorre a construção de agrupamentos de formas variados. A forma como o agrupamento dos registos é realizada é determinada pela vizinhança de raio R que este não pode ultrapassar o raio delimitado. Este limite do raio R é calculado pela distância euclidiana. Para alem do DBSCAN, existe o OPTICS (*Ordering Points to Identify the Clustering Sctrcture*) e DENCLUE (*DENsity-based Clustering*) que tem o mesmo modo de operação utilizando o método baseado em densidade [18][36].

2.5.1.4. MÉTODO BASEADO EM GRADE (*GRID-BASED METHODS*)

Os métodos baseados em grade partem do plano que os dados são criados por uma série de probabilidade de distribuição. Estes métodos permitem criar um modelo para cada

agrupamento e induzem identificar o melhor modelo para cada objeto. Os algoritmos que utilizam este método são o EM (*Expectation-Maximization*), COBWEB e o CLASSIT. Estes métodos buscam otimizar o auxílio que um determinado conjunto de dados que podem ajudar nos modelos matemáticos, com isto existem duas abordagens, a redes neurais e a estatística.

Um exemplo dado pelas redes neurais passa pela existência de duas camadas, uma de entrada e uma de saída que também podemos chamar de camada competitiva. Começando pela camada de entrada, aqui estão os dados que são os neurônios de entrada e que são os dados introduzidos. Os neurónios que estão presentes na camada de entrada estão ligados a todos os neurónios da camada competitiva em que cada ligação possui um peso, este é calculado pela medida de distância do neurónio mais "perto" da entrada. Um neurónio é dado como vencedor se tem os seus pesos atualizados e que mostra a proximidade ao registo de entrada. A vizinhança do neurónio vencedor vai diminuindo conforme as iterações ou fases do treino da rede. Os vizinhos também possuem pesos atualizados, mas tem menor influência do neurónio vencedor, a distância é aumentada em relação ao neurónio vencedor. Ou seja, a ideia é que um dado que não tenho sido antes submetido à rede neural possa ser colocado na rede e que os neurónios que tem os pesos ajustados para um determinado padrão tenham os pesos com valores aproximados aos valores do novo dado de entrada [43]. A imagem a seguir mostra essa iteração [36].

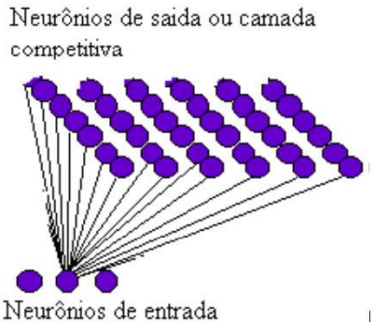


Figura 2.14: Estrutura do método baseado em modelos

2.5.2. ALGORITMOS DE *CLUSTERING*

2.5.2.1. *AGLOMERATIVE HIERARCHICAL CLUSTERING*

O algoritmo *Aglomerative Hierarchical Clustering* é demostrado como um diagrama de estrutura de árvore que pode ser chamado de dendograma, como explicado na secção 2.4.1.1.

Este modo de agrupamento aglomerativo é o mais comum e aplicado em comparação ao método divisivo.

O procedimento do *aglomerative hierarchical Clustering:*

1. Criação de n cluster com observação;
2. Cálculo da matriz de proximidade;
3. Misturação de dois *clusters* vizinhos mais próximos;
4. Cálculo da distância entre os clusters
5. Repetição das etapas anteriores, 3 e 4 várias vezes, até que obtenhamos clusters permanentes.

Estes tipos de algoritmos requisitam, através de calculo computacional, a distância entre dois clusters obtendo assim a principal etapa que é o cálculo da matriz de proximidade. Este é definido por MIN, MAX e *Group Average* que é justificada por clusters obtidos por gráficos. Assim podemos definir os três parâmetros da matriz de proximidade:

1. MIN: através de dois pontos mais próximos de cada cluster, permite calcular a proximidade entre clusters diferentes.
2. MAX: Ao contrário do MIN, este calcula a distância de dois pontos mais distantes entre dois clusters mais próximos.
3. *Group Average*: Por último, este define através das proximidades dos pares médios dos pontos inteiros de clusters próximos e distintos [37].

As equações seguintes mostram os três tipos de medidas da matriz de proximidade.

Para o Min ou Single Link temos:

$$d(A,B) = min\{x \in A, y \in B\}$$

Para Max ou Complete Link temos:

$$d(A,B) = max\{x \in A, y \in B\}$$

Para o Group Average ou Average Link temos:

$$d(A,B) = \frac{1}{|A||B|} \sum y \in A \sum x \in B$$

Podemos observar em termos esquemáticos o calculo da matriz de proximidade [38].

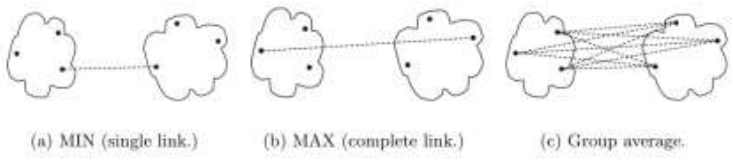


Figura 2.15: Demostração da matriz de proximidade

2.5.2.2. *K-MEANS*

O *K-means* é o algoritmo mais conhecido quando se fala em *clustering*, sendo este simples, popular e o mais usado na aplicação do critério do erro quadrado. Este algoritmo inicia-se com uma distribuição inicial aleatória e mantêm-se atribuído aos novos clusters padrões com suporte na similaridade, de acordo de diálogo com o padrão e com o cluster, terminando quando seja atingido um critério de convergência. Este conclui-se quando já não ocorre reatribuições, ou seja, até que não exista movimentos de dados de uns *clusters* para os outros e isto deve-se quando os centroides estabilizam ou então o algoritmo completa a execução de um limite especificado de iterações [44][46].

Um grande problema deste algoritmo passa pela sensibilidade que este tem à partição inicial e quando este tem tendência em convergir num mínimo local. Este algoritmo tem a particularidade de somar o quadrado dos erros entre os itens num cluster e o seu centro. Resumindo o algoritmo tem a função de obter excelentes resultados quando os clusters estão compactados e isolados [44].

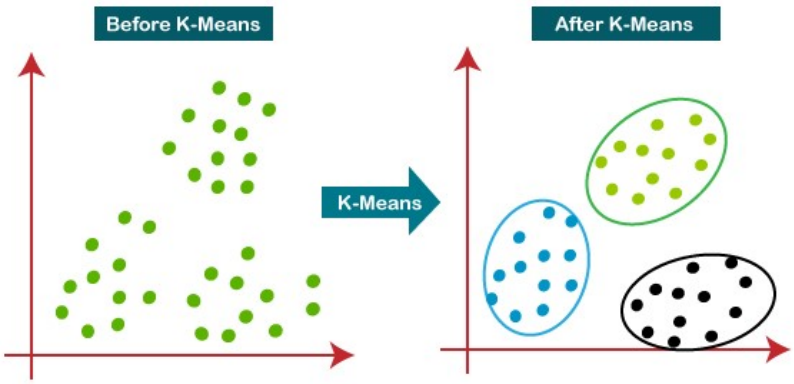


Figura 2.16: Obtenção de conjuntos de *clustering (k-means)*, retirado de [39]

Processo do algoritmo *K-means:*

1. É escolhido o valor k (centroides) de clusters;

2. É atribuído a cada padrão o centro do cluster mais aproximado;

3. É novamente calculado o centro dos clusters já pré-definidos;

4. Enquanto o critério de convergência não é obtido, voltamos ao passo 2 e continua-se os restantes passos.

Existem outras variantes do algoritmo *k-means*, que permitem melhorar não só a qualidade do processamento do algoritmo como a sua rapidez de processamento e eficácia, é o caso do *k-means*++, que permitem selecionar uma ótima partição inicial, com isto, permite-nos encontrar os valores mais prováveis para as formações dos clusters. Resumindo os clusters passam por uma fase de dividir e juntar os clusters, ou seja, quando estes são divididos, este mostra que o seu desencontro que está na parte superior de um limiar predefinido, e quando estão juntos os centroides estão na parte inferior do limiar predefinido [47].

O processo da variante *K-means*++ é descrito nos seguintes passos:

1. Assuma o centro C_1, escolhido uniformemente aleatoriamente a partir de X;
2. Faz um novo centro C_i, escolhendo $x \in X$ com probabilidade $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$;
3. Repete-se o passo 2. até encontrarmos centro do K todo junto;
4. Procede-se as etapas do algoritmo *k-means* padrão.

Usando estas variantes, é possível obter a partição ótima a partir de qualquer partição arbitrária inicial, tendo em conta os valores limites estabelecidos [44][47].

2.5.2.3. *K-MEDOIDS*

O algoritmo *k-medoids* é um algoritmo de agrupamento com a particularidade de ser mais resiliente a *outliers,* quando comparado com o algoritmo *k-means.* Este algoritmo diminui a sensibilidade ao ruido sendo que é mais robusto do que o *k-means*. O principal método para diminuir a existência de ruido é a utilização de um *medoid*, ou seja, a utilização do objeto mais centralizado de um conjunto de objetos que formam o *cluster*, ao contrário do *k-means* que

utiliza o valor médio para calcular o centro de um cluster. Com a utilização do *medoid* podemos minimizar a soma das diferenças entre o ponto de referência e o objeto pertencente ao *cluster*.

A estratégia do algoritmo *k-medoids* passa por encontrar k *clusters* em n objetos de modo aleatório, depois o objetivo passa por encontrar o *medoid* para cada *cluster*. Os objetos restantes passam pelo processo de *cluster* com o *medoid* ao que é mais similar ao mesmo, de seguida a estratégia passa repetir iterativamente trocando um dos *medoids* por outros que não são *medoids* a medida que o resultado da performance do *clustering* vai melhorando. Para calcularmos essa melhoria o próprio algoritmo usa uma função custo em que este mede a similaridade média no próprio cluster com o *medoid* [36][40].

Processo do *K-medoids*

1. Selecione, arbitrariamente, K pontos ou objetos do conjunto como *medoids* iniciais no cluster;
2. Repetido o processo anterior;
3. É atribuído a cada ponto ou objeto restante ao cluster com o *medoid* mais aproximado;
4. De uma forma aleatória, é selecionado um ponto ou objeto que não seja *medoid,* O;
5. É calculado o custo total, X, para trocar o *medoid* pelo objeto O;
6. Se o custo/distancia for menor que o objeto O, é trocado *medoid* pelo objeto O;
7: Este processo é repetido até que não exista mudança de pontos ou objetos entre clusters [40].

2.5.2.4. *X-MEANS*

O *X-means* é um método de agrupamento que pode ser usado para estimar eficientemente o valor de K. Este usa um método chamado de lista negra que serve para identificar conjuntos de centroides entre os existentes, que podem ser divididos para ajustar melhor os dados.

Os valores K para a experiência são escolhidos entre um valor de limite inferior e superior selecionado, reduzindo o espaço de procura com recurso a uma heurística, o *x-means* executa o algoritmo tradicional do *k-means* para recolha dos resultados, apenas difere na escolha quando esta estima o valor K num intervalo estabelecido pelo utilizador.

O objetivo principal deste algoritmo é estimar K de forma eficiente e fornecer um algoritmo de agrupamento *k-means* escalável quando o número de pontos de dados torna-se grande [41][42].

2.5.2.5. *K-MEANS KERNEL*

A existência do algoritmo *k-means kernel* surge pela carência de alterar o algoritmo original *k-means* porque este apresenta dificuldades em separar de forma não linear os pontos em clusters, porque o algoritmo originário realiza uma separação rígida dos elementos quando existem conjuntos de dados não convexos.

Com a utilização deste algoritmo evita-se o conhecimento dos centroides obtidos pelo *k-means,* razão pelo qual o espaço característico permite obter uma função caraterística do *k-means kernel* que utiliza funções *kernels.* Isto é obtido pela seguinte equação [43]:

$$\min_{c} SSE(C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left\| \phi(x_i) - Z_i^{\phi} \right\|^2 \tag{2.1}$$

As obtenções dos clusters finais são projetadas nos dados com um dimensionamento elevado no espaço do *kernel*. As funções mais importantes no algoritmo *K-means kernel* são *kernel polinimial*, o *kernel gaussiano* e o *kernel sigmóide*.

Neste algoritmo a complexidade computacional é mais elevada que o *k-means*, devido a matriz do *kernel* ser gerada com os dados fornecidos a partir da função *kernel.* [44][45].

Na ferramenta *rapidminer Studio* ® tem disponível os seguintes tipos de *kernel* que são (1) *dot*: este é definido por k(x,y) = xk(x, y) = X × Y sendo o produto interno de X e Y, (2) *radial*: este é definido pela expressão $e^{-g\|X \times Y\|^2}$, onde g é o gama, (3) *polynomial*: O *kernel* do polinómio é definido por $k(x, y) = (x \times y + 1)^d$, onde d é o grau do polinómio, (4) *sigmoid*: o núcleo é definido por uma rede de duas camadas, e é calculado pela expressão $tanh\,(ax \times y + b)$ onde a é o valor alfa e b a constante de interceção, (5) *anova:* Este é definido pela expressão $e^{-g(X \times Y)}$, onde g é o valor gama, (6) *epachnenikov*: É uma função definida por $\frac{3}{4}(1 - u^2)$ onde u situa-se entre -1 e 1 e 0 quando o u se encontra fora desse

intervalo, (7) *gaussian combination*: Sendo este um *kernel* da combinação gaussiana e por fim (8) *multiquadric*: Este é definido pela expressão $\sqrt{\|X \times Y\|^2 + C^2}$ [46].

2.5.2.6. *K-MEANS FAST*

Este algoritmo, ao contrário do *k-means*, não necessita que a totalidade dos dados estejam presente na memória, na mesma altura em que a operação é ocorrida numa única passagem sobre os dados, estes podem melhoram a eficiência dos *clusters* dos dados, ou seja, o *k-means fast* não exige que ocorram várias passagens sobre os pontos de dados antes da deteção dos centroides dos clusters. Com isto temos uma redução do tempo de processamento, o que pode chegar a cinco vezes mais em meio milhão de dados no caso se for de um milhão poderá chegar a dez vezes mais [42][47].

2.5.2.7. *RANDOM CLUSTERING*

O *Random Clustering* simplesmente itera todos os exemplos e atribui um cluster aleatório a cada um. Não existe um método específico de agrupamento [48].

2.5.3. TIPOS DE MEDIDAS DE SIMILARIDADE

Existe três tipos de medidas de similaridade: *Mixed Measures*, *Nominal Measures*, *Numerical Measures* e *Bregman Divergences*. Dentro dos tipos de medidas, temos as *Mixed Measure*: Apresenta apenas uma medida mista que é a Distância Euclidiana Mista, este utiliza atributos do tipo nominais e numéricos. A distância euclidiana, em matemática, é a distância entre dois pontos $P = (p_1, p_2, \ldots, p_n)$ e $Q = (q_1, q_2, \ldots, q_n)$ num espaço euclidiano n-dimensional, provado pelo teorema de Pitágoras. Este é calculado através de [49]:

$$Dist.Euclidiana = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} \quad (3.1)$$

Resumindo, pode ser expressa pela seguinte equação:

$$Dist.Euclidiana = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad (3.2)$$

De seguida temos o parâmetro *Numerical Measures*, como o próprio nome indica este é utilizado apenas em atributos numéricos, ou seja, diferentes métricas de distância podem ser usadas para calcular os atributos numéricos. Os atributos numéricos de entrada num *ExampleSet* podem ser definido pelas seguintes variáveis: y(i,j) sendo o valor do j-ésimo atributo do i-ésimo exemplo. Por exemplo, y(1,3) – y(2,3) é calculado a diferença dos valores do terceiro atributo, do primeiro e do segundo exemplo. Em casos de usar a similaridade como medida de distância real, esta é calculada como similaridade negativa. Dentro deste parâmetro temos diversos modelos de medidas, tais como, (i) *Euclidean Distance*: que é a raiz quadrada da soma das diferenças quadráticas sobre todos os atributos, ou seja, é idêntica a medida anteriormente descrita. (ii) *Camberra Distance:* É uma medida numérica de distância entre pares de pontos em um espaço vetorial, basicamente esta é uma versão ponderada da distância de Manhattan, será descrita mais a frente, usa como métricas listas de classificação e deteção de intrusões na segurança de computadores, em termos práticos é definida pela diferença absoluta entre as variáveis dos dois objetos e dividida pela soma dos valores absolutos das variáveis antes da soma [50][51]. Representação da equação, em que p e q são vetores:

$$d(p,q) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|} \tag{3.3}$$

(iii) Chebychev Distance: É uma unidade de medida definida no espaço vetorial em que a distância entre os dois vetores é a maior diferença ao longo de qualquer dimensão. A seguinte equação mostra esta unidade:

$$dist = Max_{j=1}|y(1,j) - y(2-j)| \tag{3.4}$$

(iv) Correlation Similarition: É uma técnica para calcular a relação entre duas variáveis quantitativas e continuas. O coeficiente de correlação de Pearson é uma medida relacionada à força e à direção de um relacionamento linear [52]. Calculamos esta medida por:

$$CORR(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.5}$$

Onde,

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.6}$$

Este pode assumir um intervalo de valores entre -1 e 1.

Podemos verificar exemplos de medida de correlação:

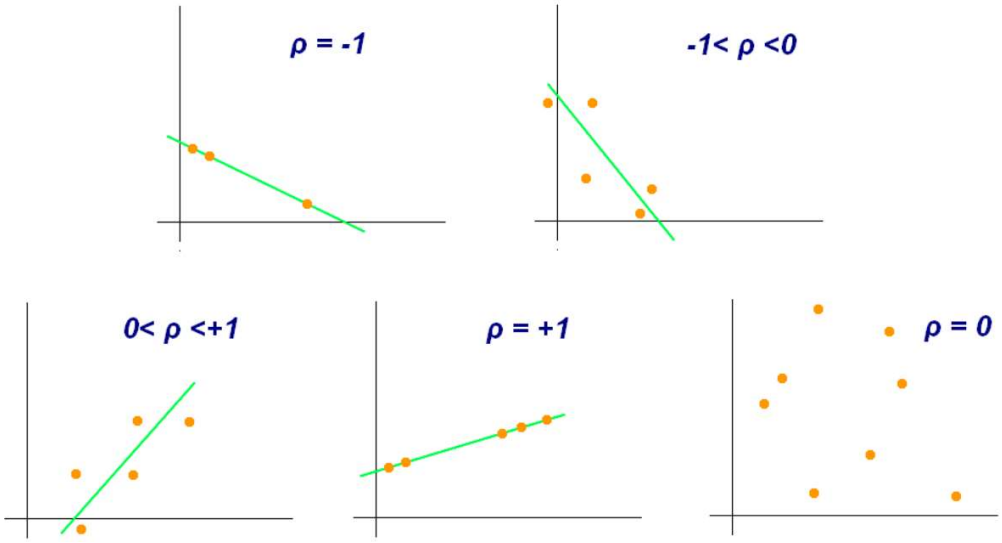


Figura 2.17: Exemplos de medidas de correlação [79]

(v) Cosine Similarition: E a diferença entre dois vetores diferentes de zero de um espaço interno de um produto que mede o cosseno do angulo entre os mesmos. A técnica também é usada para medir a coesão dentro de clusters no campo de mineração de dados. A similaridade de cosseno é muito popular na análise de texto. É usado para determinar a semelhança entre os documentos, independentemente do tamanho [52]. Representado pela seguinte equação:

$$Similarity = \cos(\theta) = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (3.7)$$

Podemos observar um exemplo na figura seguinte:

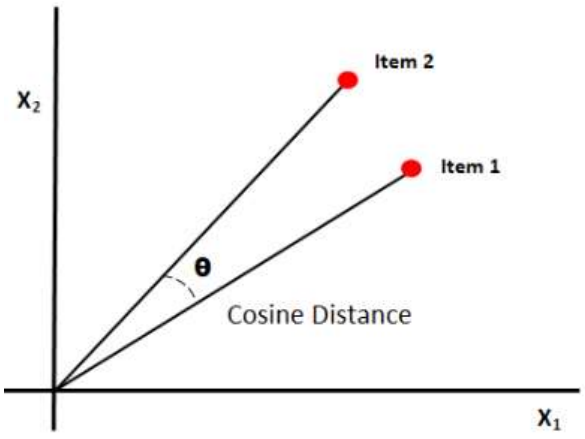


Figura 2.18: Representação gráfica da similaridade do cosseno, retirado de [79]

(vi) Dice Similarition: Também conhecido como índice de *Sørensen – Dice* ou simplesmente coeficiente dos dados, é uma ferramenta estatística que mede a similaridade

entre dois conjuntos de dados. Esse índice tornou-se na ferramenta mais usada na validação de algoritmos de segmentação de imagens criados por Inteligência Artificial, mas é um conceito muito mais geral que pode ser aplicado a conjuntos de dados para uma variedade de aplicações, incluindo processamento de linguagem natural.

$$dice = \frac{2 \times |x| \cap |y|}{|x| + |y|} \tag{3.8}$$

Quando aplicamos os dados booleanos, usando a definição de verdadeiro positivo (TP), falso positivo (FP), falso negativo (FN) e verdadeiro negativo (FN), podendo ser escrito como:

$$dice = \frac{2TP}{2TP + FP + FN} \tag{3.9}$$

(vii) Dynamic Time Warping Distance: É frequentemente usado na análise de séries temporais, para medir a distância entre duas sequências temporais. Aqui é calculada a distância em um caminho "distorcido" ideal do vetor do atributo do primeiro exemplo ao segundo exemplo.

(viii)Inner Product Similarity: A similaridade é calculada como a soma do produto dos vetores de atributos dos dois exemplos. A distância é igual a Similaridade negativa, como demostra a equação:

$$Dist = -\sum_{j=1} y(1,j) \times y(2,j) \tag{3.10}$$

(xiii) Overlap Similarity: Este permite descobrir quais os objetos que são subconjuntos de outros objetos, permitindo assim descobrir a taxonomia a partir de dados assinalados. A similaridade é uma variante da correspondência simples para atributos numéricos e é calculada pelo seguinte fórmula [53]:

$$O(X,Y) = \frac{|X \cap Y|}{\min(|X|,|Y|)} \tag{3.11}$$

Por fim temos a *Bregman Divergences* em que destacamos os parâmetros por serem utilizados, *(i) Mahalanobis Distance:* É a distância entre dois pontos num espaço multivariado, esta é baseada nas correlações entre variáveis de padrões distintos que podem ser identificados e analisados, ou seja, permite determinar a similaridade entre uma amostra desconhecida e uma conhecida, esta é calculada pela fórmula [54].

$$d(M) = [(X_B - X_A)^T \times C^{-1} \times (X_B - X_A)]^{0.5} \tag{3.12}$$

(ii) Squared Euclidean Distance: Quando um evento de erro ocorre é utilizado a distância euclidiana quadrada para a sequência de entrada correta, sendo esta maior do que a sequência de entrada mais provável. Representado pela seguinte fórmula [55].

$$\sum_{i=1}^{n}(r_i - y_i)^2 > \sum_{i=1}^{n}(r_i - \hat{y}_i)^2 \tag{3.13}$$

2.6. MÉTODOS DE AVALIAÇÃO

Ao aplicarmos métodos de *clustering* por mais variados que se utilize é importante perceber qual se adequa melhor aos dados e a qualidade dos mesmos, permitindo assim utilizar um método de avaliação que permita garantir a qualidade dos resultados obtidos. Para isso é necessário avaliar o desempenho de um classificador que encontre o bom grau de *accuracy*. Para isso, necessitamos perceber a contagem de exemplos que se encontram corretamente e incorretamente previsto.

2.6.1. MATRIZ DE CONFUSÃO (*CONFUSION MATRIX*)

A matriz de confusão facilita a visualização do número de classificações corretas e do número de classificações preditas para cada classe, de um determinado conjunto de exemplos, segundo o classificador em análise. Esta torna-se uma ferramenta útil para analisar a qualidade da classificação, quer no reconhecimento de exemplos de diferentes categorias.

Quando um conjunto de dados tem apenas duas categorias, muitas vezes considera-se uma como 'positiva' e a outra como 'negativa'. Desta forma, podemos considerar que a matriz de confusão é uma tabela com duas linhas e duas colunas que regista o número de *True Negatives* (TN), *False Positives* (FP), *False Negatives* (FN) e *True Positives* (TP).

Tabela 2.3: Demostração da *confusion matrix*, retirado de [55]

Categoria	Prevista C^+	Prevista C^-
Verdadeira C^+	TP	FN
Verdadeira C^-	FP	TN

TP refere-se ao número de exemplos da categoria 'positiva' corretamente previstos; FN representa o número de exemplos da categoria 'positiva' incorretamente previstos; FP refere-se ao número de exemplos da categoria 'negativa' incorretamente previstos; e TN representa o número de exemplos da categoria 'negativa' corretamente previsto. Com base nas entradas da matriz de confusão, o número total de previsões corretas feitas pelo classificador é TP+TN e o total número de previsões incorretas é FP+FN [56], assim podemos obter uma medida, chamada de *accuracy*.

$$Accuracy = \frac{TP\ + TN}{TP + TN + FN + FP} \tag{2.2}$$

Podemos calcular o exemplo classificado com incorretos, ou seja, a percentagem de previsões incorretas, pela seguinte formula [56]:

$$Classification\ error = \frac{Incorrect\ Preditions}{Number\ of\ Examples} = \frac{FP + FN}{TP + FP + FN + FN} \tag{2.3}$$

O coeficiente *Kappa* proposto por *Jacob Cohen* em 1060 trata-se de um método estatístico para avaliar o nível de concordância ou reprodutibilidade entre dois conjuntos de dados.

Este coeficiente é considerado como um dado conservador e é descrito pela seguinte equação [56]:

$$Cohen´sKappa = \frac{\text{precisão observada} - \text{precisão esperada}}{1 - \text{precisão esperada}} \tag{2.4}$$

A precisão observada é obtida por:

$$\text{Precisão observada} = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.5}$$

A Precisão esperada é obtida por:

$$\text{Precisão esperada} = \frac{(TP + TN) \times (TP + FN) \times (FN + TN)}{(TP + FP + FN + TN)^2} \tag{2.6}$$

Landis e *Koch* em 1977 classificaram esta regra em diferentes níveis de concordância ou reprodutividade conforme mostra a tabela a baixo [57]:

Tabela 2.4: Níveis de concordância do coeficiente *Kappa*

Valor do Coeficiente *Kappa*	Nível de concordância
< 0	Não existe concordância
0 – 0,20	Concordância Mínima
0,21 – 0,40	Concordância Razoável
0,41 – 0,60	Concordância Moderada
0,61 – 0,80	Concordância Substancial
0,81 – 1,0	Concordância Perfeita

Existe diversos métodos de classificação que se pode utilizar com recurso ao software *Rapidminer Studio* ® tais como: *weighted mean recall, weighted mean precision, spearman rho, kendalltau, absolute error* e *relative error*, entre outros [58].

2.6.2. REGRA DE ASSOCIAÇÃO (*ASSOCIATION RULES*)

Este método tem como finalidade o agrupamento de escolha e previsão. Esta regra de associação carateriza-se pela aquisição de um determinado artigo e associá-lo à aquisição de outro artigo. Esta regra é muito utilizada no comércio, porque faz com que as vendas possam ser realizadas de uma forma lucrativa para a entidade e que desperta a atenção de compra por parte do utilizador.

A aplicação desta regra envolve uma estratégia de associar um produto que influencia a compra de outro, a partir daqui podemos verificar que quando o cliente compra um determinado artigo tem a intenção de conjugar outro artigo na sua compra. Por exemplo uma pessoa vai ao supermercado para comprar frango assado, como forma de comprar algo pronto a comer, este como pretende uma refeição pronta e é de costume associar um acompanhamento, por exemplo batatas fritas, assim o comerciante adota uma estratégia para despertar a venda dos dois ou mais produtos. Deste modo, o comerciante pode gerir a venda com promoções, destaques nas prateleiras e entre outros métodos.

A regra de associação designa-se por uma expressão na qualidade de implicação no formato $X \rightarrow Y$ ou seja que X se associa a Y. Esta regra pode ser avaliada em termos do seu suporte e da confiança [59].

O suporte é determinado pela frequência de um conjunto X no qual uma regra é aplicável de um conjunto de vários itens [60]:

$$Suporte\ (X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \tag{2.7}$$

Já a confiança determina-se pela frequência na qual os itens Y aparecem associados a produtos de X, representa-se assim pela equação da confiança [60]:

$$Confiança(X \rightarrow Y) = \frac{\mathrm{su\,p}(X \cup Y)}{\sup(X)} \tag{2.8}$$

Por palavras podemos dar o exemplo da regra {Frango, Batata Frita} ⟶ {Cerveja}. Assim temos como contador de suporte para {Frango, Batata Frita, Cerveja} é 2, e o número total de transações é 5, logo o suporte da regra é $\frac{2}{5} = 0{,}4$. Assim temos a possibilidade de obter a confiança a partir do contador de suporte {Frango, Batata Frita, Cerveja}. Através da regra de confiança podemos obter o valor $\frac{2}{3} = 0{,}667$, já que existe 3 transações que contem frango e batata frita.

Afinal o que é o suporte e a confiança? O suporte para a regra de associação é uma medida de alta importância porque aos dados da regra em estudo se forem abaixo do valor de suporte, quer dizer que aconteceu apenas por mera coincidência e não ser realmente importante para o cliente, assim o utilizador da regra consegue perceber que não é rentável a utilização desta associação e assim permite eliminar esta combinação.

A confiança tem o intuito de medir a confiabilidade da intervenção feita por uma regra. Para analisar a confiança quanto maior for a probabilidade de acontecer Y quando implica a ocorrência de X, assim temos um forte relacionamento de concorrência entre itens no antecedente e o consequente da regra.

A forma de calcular o suporte e a confiança requer muito esforço por parte da ferramenta *Rapidminer Studio* ®, isto é, só para calcular um conjunto de regras extraídas de um conjunto de dados que contenha d itens é obtida pela seguinte formula [60]:

$$R = 3^d - 2^{d+1} + 1 \tag{2.9}$$

Supondo que temos 5 itens disponíveis que podem ser conjugados numa regra de associação e fazendo os cálculos de modo a calcular o suporte e a confiança obtemos $3^d - 2^{d+1} + 1 = 602$ regras, um valor muito elevado quando que realmente interessa, posto isto, existe a necessidade de aplicar um valor mínimo de suporte e de confiança. Por exemplo se colocarmos

um mínimo de suporte em 20% e um mínimo de confiança em 50%, temos que a maioria do cálculo se torne inutilizável e assim evita o estudo de dados que não são confiáveis [60].

Os algoritmos *Apriori* e o *FP-Growth*, são os mais conhecidos e utilizados para a implementação da *Association Rules*. Em particular e com mais enfase é estudado com mais pormenor o algoritmo *FP-Growth*, este foi utilizado nesta tese e está presente no *Rapidminer Studio* ®.

2.6.2.1. ALGORITMO APRIORI

Este algoritmo foi concebido por *Rakesh Agrawal* no ano 2013 [61], o função deste algoritmo é detetar e agrupar os itens frequentes nos dados e criar itens que são candidatos.

O funcionamento deste algoritmo tem por base contabilizar o suporte de itens individuais e determinar quais são os frequentes, consequentemente os que tem suporte mínimo, após cada passagem, esta inicia-se com um grupo pré-definido de itens estimados como frequentes na passagem anterior. Depois de estabelecido o grupo pré-definido, este é utilizado para gerar potenciais novos itens frequentes, que lhe chamamos de itens candidatos. Chegando ao final da passagem, o conjunto de itens considerados frequentes torna-se pré-definidos para a próxima passagem, este processo continua até que sejam encontrados novos itens frequentes. Este processo termina gerando as regras de associação do tipo X para Y, visualizando as regras que satisfazem a condição de confiança maior para mínima [60].

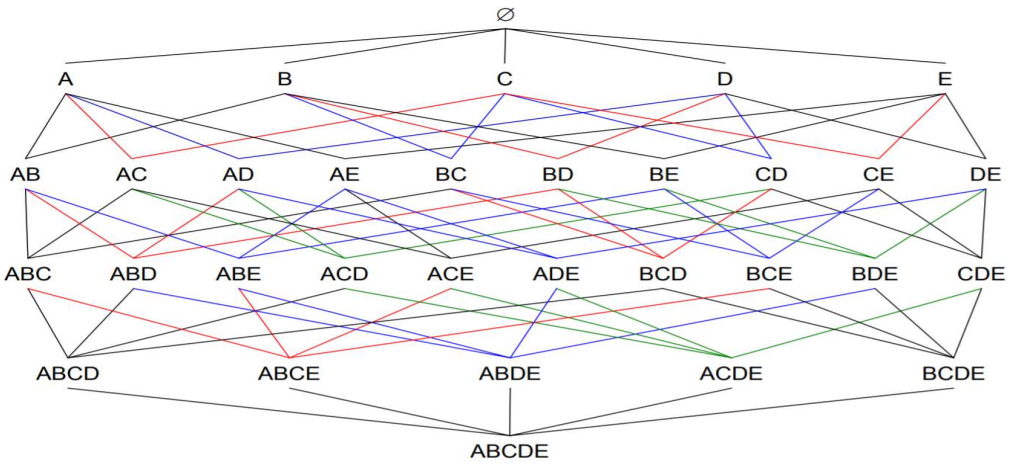


Figura 2.19: Exemplo da aplicação do algoritmo *Apriori*, retirado de [62]

2.6.2.2. ALGORITMO FP-GROWTH

O *FP-Growth* (*Frequent Pattern - Growth*), foi o implementado nesta dissertação por ser um dos algoritmos mais famosos e o único que existe no ferramenta *Rapidminer Studio* ®, este algoritmo difere do *Apriori* porque não utiliza o padrão de gerar e testar, este reúne o conjunto de dados usando uma estrutura de dados comprimida que lhe chamam de árvore FP e que extrai conjuntos de dados frequentes evitando uma nova passagem por todos os dados, assim temos uma leitura mais seletiva e com menos recursos utilizados, tais como memória e tempo de leitura como acontece com o algoritmo *Apriori* [60].

Uma árvore FP é uma representação compacta dos dados de entrada e é construída a partir de um conjunto de dados formados por transações. Uma leitura de cada transação é feita por vez e mapeia cada transação em um caminho na árvore FP.

Quanto mais os caminhos se sobrepõem, mais compressão podemos obter usando a estrutura de árvore FP. Se o tamanho da árvore FP for suficientemente pequeno para caber na memória principal, isto nos permitirá extrair conjuntos de itens frequentes diretamente da estrutura na memória em vez de executar passagens repetidas pelos dados armazenados em disco [60]. Iremos explicar o algoritmo de uma forma simplista.

Temos as seis transações de uma loja fictícia.

Tabela 2.5: Tabela de transação de uma loja

Transação	Pão	Manteiga	Leite	Café	Adoçante
T1	1	1	1	0	0
T2	0	1	1	1	0
T3	0	0	0	1	1
T4	1	1	0	1	0
T5	1	1	1	0	1
T6	1	1	1	1	0

Inicialmente começamos pelo contador de frequências do conjunto de itens, este passo é idêntico ao algoritmo *Apriori*. A tabela seguinte mostra a frequência de itens [62].

Tabela 2.6: Frequência de itens de um conjunto

Itens	Número de transações
Pão	4
Manteiga	5
Leite	4
Café	4
Adoçante	2

Definindo o suporte mínimo de 3 itens, verificamos que o item Adoçante não tem os critérios mínimos para entregar o suporte mínimo, logo ficamos com a seguinte tabela.

Tabela 2.7: Frequência de itens acima de 2

Itens	Número de transações
Pão	4
Manteiga	5
Leite	4
Café	4

Pela ordem de frequência da tabela (anterior), organizamos os itens para cada transação, verificando que o item adoçante está descartado.

Tabela 2.8: Itens ordenados pela frequência em cada transação

Transações	Itens	Itens ordenados
T1	Pão, Manteiga, Leite	Manteiga, Pão, Leite
T2	Manteiga, Leite, Café	Manteiga, Leite, Café
T3	Café, Adoçante	Café
T4	Pão, Manteiga, Café	Manteiga, Pão, Café
T5	Pão, Manteiga, Leite, Adoçante	Manteiga, Pão, Leite
T6	Pão, Manteiga, Leite, Café	Manteiga, Pão, Leite, Café

Após a ordenação de cada transação, começamos a construir a árvore *FP tree.* A raiz da árvore aparecerá sempre como *null* e para cada nó suportará o item e a frequência do padrão. Iniciemos a obtenção das transações, a T1, T2, T3, T4, T5 e T6 e obtemos as seguintes árvores.

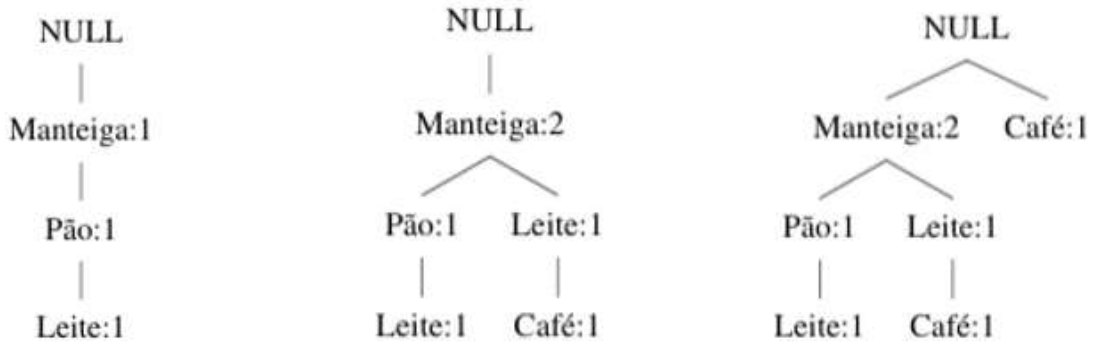


Figura 2.20: Construção da árvore *FP tree* após T1, T2 e T3

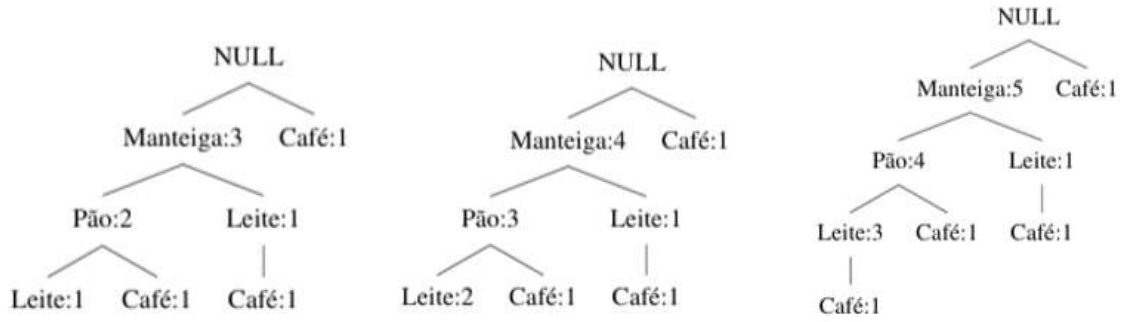


Figura 2.21: Construção da árvore *FP tree* após T4, T5 e T6

Após representar as seis transações, temos agora condições para construir os conjuntos intermédios da *conditional pattern* e da *conditional FP tree*, e assim, conseguimos gerar os itens dos conjuntos frequentes.

Tabela 2.9: Frequência inversa com a *Conditional Pattern*

Frequência inversa	*Conditional Pattern*
Café	{Manteiga, Pão, Leite: 1} {Manteiga, Leite:1} {Manteiga, Pão:1}
Leite	{Manteiga, Pão: 3} {Manteiga: 1}
Pão	{Manteiga: 4}
Manteiga	-

Para descobrimos o *condittional pattern base* necessitamos de repetir sobre cada item da coluna frequência inversa, e reservar o caminho da raiz até ao nó em questão. Por exemplo, queremos encontrar o nó café então existe a possibilidade através de {Manteiga, Pão, Leite}, {Manteiga, Leite}, {Manteiga, Pão}. Temos a particularidade que a transação T6 o caminho direto da raiz ao café não é contado.

Tendo a *conditional pattern base*, podemos agora gerar o *conditional FP tree* a partir dos valores que se encontram acima do suporte mínimo dos conjuntos de itens.

Tabela 2.10: *Conditional pattern base* com *Conditional FP tree*

Conditional pattern base	*Conditional FP tree*
{Manteiga, Pão, Leite: 1} {Manteiga, Leite:1} {Manteiga, Pão:1}	[Manteiga: 3]
{Manteiga, Pão: 3} {Manteiga: 1}	[Manteiga: 4, Pão: 3]
{Manteiga: 4}	[Manteiga: 4]
-	-

Agora temos todas as condições para encontrar os conjuntos frequentes deste exemplo.

Tabela 2.11: Padrões frequentes (FP)

Frequência inversa	Conditional FP tree	Conjuntos frequentes
Café	[Manteiga]	{Manteiga, Café}
Leite	[Manteiga: 4, Pão: 3]	{Manteiga, Leite}, {Pão, Leite}, {Manteiga, Pão, Leite}
Pão	[Manteiga: 4]	{Manteiga, Pão}
Manteiga	-	-

Para terminar, conseguimos reencontrar os conjuntos do exemplo anterior desde que se realize a combinação da respetiva coluna de frequência com os itens do *conditional FP tree*.

Avaliando estes dois algoritmos descrito na secção 2.4.4.1, este permite referir que o *FP-Growth* é um algoritmo mais eficiente e complexo, porque este necessita apenas de dois varrimentos invés do algoritmo *Apriori* que faz varrimentos para cada iteração, para alem que a utilização da memoria é muito mais compacta. A única desvantagem da utilização do algoritmo *FP-Growth* é na mineração incremental, quando é adicionado transações aos dados, assim refletia numa atualização de novos dados na estrutura [62].

2.6.2.3. QUALIDADE DOS PADRÕES DA *ASSOCIATION RULES*

O algoritmo das regras de associação tem a facilidade de gerar um grande número de padrões, sucessivamente essa quantidade excessiva de regras de associação ocorre quando o *minsup* é muito baixo. Contudo, apenas umas pequenas quantidades dessas regras são realmente interessantes para os investigadores, visto que a grande maioria das regras produzidas são obvias e irrelevantes. Assim, não torna o conhecimento produzido útil. Outra dificuldade é encontrar na medida de confiança, visto que esta medida não consegue calcular a independência entre os participantes da regra de confiança.

Com esta dificuldade, pode acontecer que exista uma falsa ideia de interesse ou apresentar resultados contraintuitivos, um modo de ultrapassar estas dificuldades é a utilização de medidas de interesse que permitem encontrar apenas as regras ou padrões que sejam interessantes ou de relevância para minerar.

Para definir adequadamente as medidas de interesse é necessário estabelecer um conjunto de critérios que sejam bem aceites para avaliar a qualidade dos padrões e das regras. Estes critérios podem ser objetivos e subjetivos.

Os critérios objetivos podem ser estabelecidos por argumentos estatísticos. A utilização de medidas de interesse objetivas pode considerar tais regras banais. Isto é conseguido em virtude de se levar em conta a correlação (co-ausência e co-ocorrência) entre os intervenientes da regra

por meio de técnicas estatísticas. Exemplos dessas medidas são suporte e confiança, que já foram descritos na definição de regras de associação.

Já os critérios subjetivos, são considerados um padrão desinteressante se for obvio, ou seja, um investigador já conhece esse padrão e seu conhecimento não suporta nova informação.

Seguindo o exemplo da regra Manteiga → Pão, é natural não ser considerado interessante porque é considerada uma regra óbvia para um investigador, mesmo apresentando um alto suporte e uma alta confiança. Pelo contrário, existe a regra Fralda → Cerveja, poderá ser interessante. Porque esta regra não é considerada como esperada que se acontece e assim permite ter um novo conhecimento que pode ser rentável e trazer lucros. Esta tarefa de recolher informação subjetivas é algo trabalhosa porque o investigador deverá conhecer previamente as informações [60].

2.6.3. *RAPIDMINER* ®

Teve o seu início no ano 2001, com a designação inicial de *YALE* na Unidade de Inteligência Artificial da Universidade Técnica de Dortmund e foi desenvolvido por *Ralf Klinkenberg*, *Simon Fischer* e *Ingo Mierswa*. Passado cinco anos, ano de 2006, esta sofreu evoluções de desenvolvimento e fundou-se a empresa *Rapid-I* pelas mãos de *Ingo Mierswa* e *Ralf Klinkenberg*. Após um ano, o software desenvolvido com o nome *YALE* foi reformulado e ficou definido como *Rapidminer* ®. Por último, no ano 2013 a empresa alterou o seu nome para *Rapidminer* ® substituindo o *Rapid-I*.

Figura 2.22: Logo do *Rapidminer* ®, retirado de [63]

Atualmente a empresa *Rapidminer* ® ajuda equipas de colaboradores nas tomadas de decisões inteligentes através da utilização de inteligência preditiva e *predactions* (*predictions*

and actions), ou seja, previsões baseadas em ações de forma a melhorar as operações das organizações, permitindo ajudar de forma eficiente as empresas a atingir escolhas inteligentes de negócio [64][65].

O *Rapidminer* ® é uma das inúmeras ferramentas existentes para análise de dados e *business intelligente* existentes no mercado, mas este destaca-se por ser desenvolvido em Java que facilita a sua utilização de um modo versátil em qualquer sistema operativo e ambiente de trabalho.

O *Rapidminer* ® é um software comercial que recorre a um ambiente de desenvolvimento *open-source* que realiza ações de *data mining*, *text mining* aprendizagem máquina, *machine learning*, etc. Este permite a análise de enormes quantidades de dados que incluem bases de dados e textos que são carregados pelos utilizadores, efetuando operações de carregamento de dados, visualização, modelação estatística, análise preditiva, avaliação e *deployment*.

Atualmente o *Rapidminer* ® oferece quatro produtos:

. *Rapidminer ® Studio*

. *Rapidminer ® GO*

. *Rapidminer ® Notebooks*

.*Rapidminer ® Al Hub*

O *Rapidminer* ® tem um programa de Licença Educacional que se estende a toda a plataforma das ferramentas disponíveis (inclui *Rapidminer Studio®,* e o *GO*® como forma de *trial*), com isto, o programa permite oferecer licenças educacionais gratuitas com duração de um ano e com possibilidade de renovação enquanto for membro de uma instituição universitária.

A browser do *Rapidminer* ® tem uma enorme iteração com os utilizadores da ferramenta, este permite dar acesso a cursos *onlines* gratuitos através do *Rapidminer academy* ®, exames de certificação com credenciais e suporte ao utilizador através do *Rapidminer community* ® que permite interação e partilha de informações e ajuda em dúvidas e problemas dos utilizadores [66][67].

2.6.3.1. *RAPIDMINER STUDIO* ®

O *Rapidminer Studio* ® apresenta várias caraterísticas, tais como:

- Disponibilização de uma biblioteca de modelos e funções (cerca de 1500 algoritmos) que permitem utilizar para cada situação apresentada pelo utilizador, visualização de dados e gráficos automáticos;
- Interface gráfica de arrastar e soltar de modo a cria fluxo de processos de análise;
- Modelos pré-construídos para casos gerais e de uso comum incluindo rotatividade de clientes, manutenção preditiva, deteção de fraude e entre outros.
- O sistema "*Wisdom of Crowds*" que fornece conselhos para cada fase de modo a ajudar os iniciantes neste software.
- Facilidade em trabalhar nas mais diversas fontes de dados como por exemplo *Excel, acess, oracle, ibm, Microsft sql, sybase, MySQL, Postgress, SPSS*, ficheiros de texto e entre outros.
- Este *ferramenta* tem as APIs (*Application Programming Interface*) abertas o que permite outros desenvolvedores de *ferramentas* desenvolverem produtos que utilizam os dados do *Rapidminer* ® [65][68].

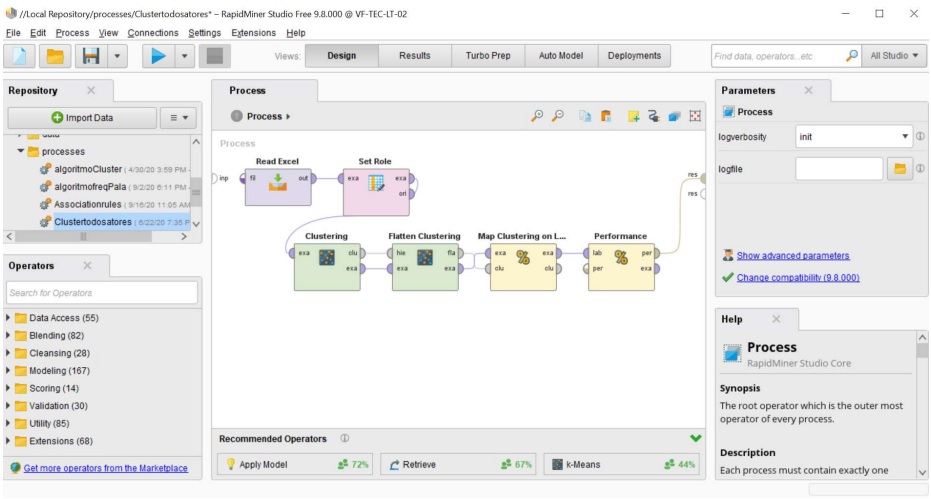


Figura 2.23: Ambiente de trabalho do *Rapidminer Studio* ®

2.6.3.2. *RAPIDMINER AL HUB* ®

O *Rapidminer Al Hub* ® (anteriormente designado por *Rapidminer Server* ®) é um ambiente do tipo servidor que possibilita uma preponderante análise preditiva suportada pela capacidade do computador.

Este apresenta as seguintes caraterísticas:

- Compartilhamento e operacionalização de modelos e resultados que são construídos no *Rapidminer Studio* ®.
- Ferramentas de agendamentos, *triggers*, execução em servidor, controle de versão, acesso remoto a processos e integração com ferramentas de *Business Intelligence*.
- Captação de resultados de predição em tempo real, utilizando aplicações de otimização da *performance;*

Este produto destina-se a empresas de grandes dimensões em que utilizam grandes volumes de dados [65][69][70].

2.6.3.3. *RAPIDMINER GO* ®

Esta é uma ferramenta de utilização online que permite de uma forma rápida e simples o entendimento dos dados, inclinado para a projeção de analistas de negócios, o *Rapidminer GO* ® permite guiar de uma forma rápida utilizando o processo de avaliação dos dados e usá-los para obter previsões.

O *Rapidminer Go* ® permite:

- Utilizar dados no formato *Excel* (xlsx e csv);
- Escolher a opção de destino;
- Escolher os recursos de input que são pretendidos;
- Modelos de regressão instantâneos;
- Avaliação dos modelos criados com base em métricas e gráficos variados;
- Exportação de resultados;

O *Rapidminer GO* ® é um produto comercial, sendo necessário uma subscrição para se poder utilizar, mas permite utilizar durante um mês como forma de teste. Mais informações e download do aplicativo é possível consultar a página web do *Rapidminer GO* ® em: https://rapidminer.com/products/go/ [71].

3. METODOLOGIA

No decorrer desta dissertação foi utilizada algumas fases da metodologia CRISP-DM, nomeadamente as fases de (1) Compreensão dos dados e preparação, (2) Configuração e descoberta do modelo e (3) Avaliação. Estas fases estão descritas na secção 2.1.1.

3.1. ANÁLISE E PREPARAÇÃO DOS DADOS

3.1.1. DESCRIÇÃO DOS DADOS

Os dados foram o resultado de um estudo prévio [7], o qual envolveu a observação de uma equipa de programadores que desenvolveu software para um banco comercial, constituída por uma chefe (Mariana) e quatro programadores (Gonçalo, Carla, Catarina e Alexandre). Os membros da equipa executam tarefas de análise, design, programação, teste e manutenção de sistemas. Durante o intervalo de recolha de informação, a equipa trabalhou nas seguintes aplicações; (1) Fornecedores, (2) Reclamações, (3) Correspondência por Correio dos Clientes (tem como nome aplicação Mail), (4) Despejos e (5) Campanhas de Marketing.

Neste estudo foram registadas 653 ações usando uma estrutura pré-definida e transferidas para um ficheiro Excel (figura 3.1). Os elementos relevantes para o presente trabalho incluíram o nome do indivíduo que realiza a ação, a ação realizada (Action_Interaction), o recetor da ação - no caso da ação ser comunicativa - a ação despoletada por uma ação comunicativa (Related_Action), um conjunto de palavras chaves (Subject_Keywords) que caracterizam o assunto da ação, ferramentas utilizadas na realização da ação (Tools), indivíduos com um papel de suporte na ação (Recursos_Humanos) e o contexto da ação. O contexto representa o conjunto de ações associadas a um dado assunto. Quando as ações são realizadas pelo mesmo indivíduo o contexto é pessoal, quando envolve dois indivíduos o contexto é interpessoal**.**

No ficheiro recebido, a coluna contexto pessoais de ação. Estes contextos foram identificados de acordo com a sua definição, agrupando ações com características semelhantes (ações, palavras chave, recursos humanos e ferramentas). Estes agrupamentos foram identificados de forma manual pela equipa de trabalho que participou no estudo com o objetivo de avaliar os resultados obtidos através de técnicas automáticas. Os agrupamentos foram

criados manualmente, sendo o primeiro objetivo deste trabalho a exploração de algoritmos que permitam a criação automática destes grupos de ações ou contextos.

Num_Seq	Day	follows	Actor_Sender	Context	Receiver	Action_Int	Related_A	Subject Keywords	Tools	Human_Re	Action_ID
1	6	999	mariana	m1	goncalo, catarina,	propose		team meeting,15h	e-mail		1
2	6	999	production team	unknown	mariana, catarina	inform		problem, automatic table	e-mail		2
3	6	2	mariana	m2	catarina	request	find	solution, automatic table u	e-mail		3
4	6	999	mariana	m3	cg team	propose	test	integration tests, claims aj	telephone		4
5	6	4	cg team	unknown	mariana	accept	test	integration tests, claims	application		5
6	6	5	mariana	m3		prepare		tests environment, claims	claims-app software		6
7	6	999	goncalo	g1		program		data management classe	visual studio dotnet, sc		7
8	6	3	catarina	t1		solve		problem, automatic table	sqlserver, message m		8
9	6	8	catarina	t1	mariana	propose		solution, automatic table	update, problem		9
10	6	9	mariana	m2	catarina	accept		solution, automatic table	update problem		10
11	6	999	carla	c1		program		common services applica	visual studio dotnet, sc		11
12	6	999	alexandre	a2		program		mail application	visual studio dotnet, sc		12
13	6	10	catarina	t1	integration team	propose		solution, automatic table u	e-mail		13
14	6	13	integration team	unknown	catarina	accept		solution, automatic table	update		14
15	6	1	catarina	t6	mariana	accept		team meeting,15h	e-mail		15
16	6	15	mariana	m1		ellaborate		project list, human resourc	word, msexcel, ms-pro		16
17	6	5	mariana	m3		test		integration tests, claims aj	claims-app software, s		17
18	6	999	catarina	t2		program		application classes, supp	visual studio dotnet, sc		18
19	6	999	alexandre	a1	mariana	ask		who-is, responsible, cards	application maintenar		19
20	6	19	mariana	m011	maintenance grou	ask		who-is, responsible , card	telephone		20
21	6	999	alexandre	a1	mail-app user	request	send	example mail records, ma	e-mail, mail applicatio		21
22	6	12	alexandre	a2		program		mail application	visual studio dotnet, sc		22
23	6	16	mariana	m1		ellaborate		project list, human resource	distribution		23
24	6	999	claims-app user		mariana	ask		how-to, create users, claims	application		24
25	6	24	mariana	m4	user	answer		how-to, create users, claims	application		25
26	6	20	maintenance chief	unknown	mariana	answer		who-is, responsible, cards	application maintena		26

Figura 3.1: Ficheiro Excel dos dados em estudo

No mesmo estudo, foram identificados os contextos interpessoais de interação. Partindo da sua definição, este segundo tipo de contexto foi identificado analisando ações comunicativas (ações com um ator emissor e um recetor) entre dois contextos pessoais de ação de dados onde se verifica a partilha de recursos semelhantes como ilustrado na figura 3.2. A sua análise por meios manuais requereu agrupamento das ações comunicativas entre dois atores que pertenciam a um mesmo par de contextos de ação pessoal (cpX, cpY). A diferença dos contextos de ação pessoal, os contextos de interação interpessoal não foram explicitamente etiquetados no ficheiro recebido.

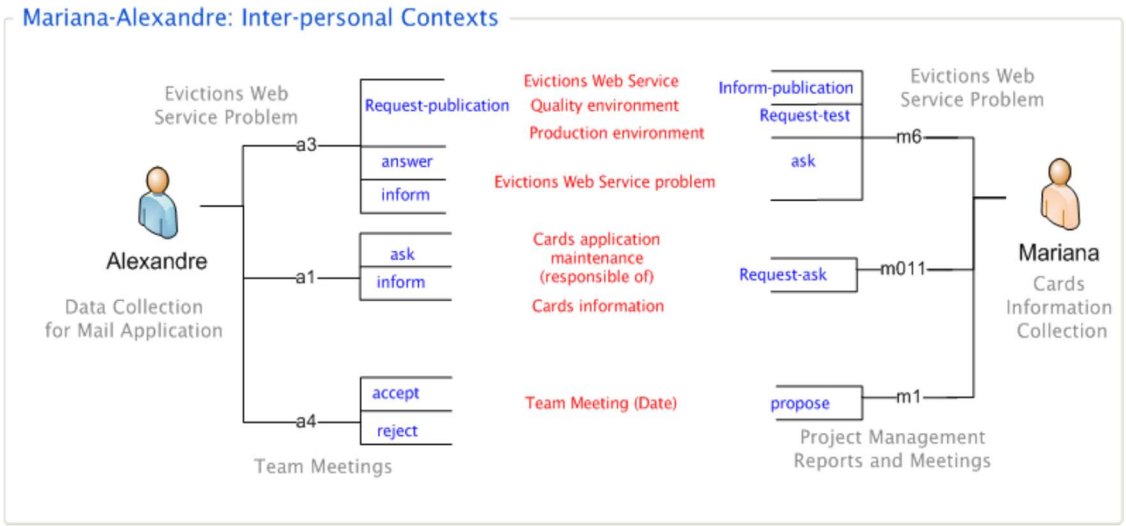


Figura 3.2: Contextos de Interação Interpessoal de interação entre Mariana e Alexandre, retirado de [7]

3.1.2. ESCOLHA DA FERRAMENTA

Das modalidades apresentadas na secção 2.4.5 a escolhida foi o *Rapidminer Studio* ®. Esta ferramenta oferece uma licença educacional destinada a organizações universitárias. O *Rapidminer Studio* ® dispõe de várias técnicas e modelos de descoberta de conhecimento e reitera que o núcleo deste se mantém com o código-fonte aberto [67].

Durante a execução desta dissertação foi usado duas versões do *Rapidminer* ®, a versão 5 e a versão 9.8 (até ao presente é considerado a última versão da ferramenta). A utilização da versão 5 deveu-se a existência da maior disponibilidade de informações e tutoriais de aprendizagem. Após um conhecimento da ferramenta utilizada, houve uma evolução na utilização para a última versão (diferenças essencialmente a nível de estética das ferramentas utilizadas). A utilização de uma versão mais recente foi avaliada em comparação com os resultados obtidos na versão 5 e garantiu-se que a utilização dos dados introduzidos nas duas ferramentas não sofria quaisquer alterações.

3.1.3. SELEÇÃO DOS ALGORITMOS

A forma específica de preparar os dados depende dos algoritmos selecionados para a sua análise. A primeira decisão tomada nesse sentido foi a de explorar algoritmos de *clustering* para implementar o agrupamento de ações com características semelhantes. A segunda decisão foi a escolha dos algoritmos de *clustering* para explorar os algoritmos disponíveis no *Rapidminer* ®. A partir da secção 2.4.2, onde são descritos alguns dos principais algoritmos de *clustering*, decidiu-se explorar para análise os algoritmos *Agglomerative Hierarquial Clustering*, *K-means*, *K-medoids*, *X-means*, *K-means (Kernel) e K-means (Fast)*. O *X-means* permite escolher um intervalo do valor K de modo a ferramenta *Rapidminer Studio* ® alcance o melhor valor para K, para a obtenção do melhor número de clusters para os dados em estudo, este algoritmo implementa o *K-means* e o *K-means (fast),* logo descartou-se a utilização deste algoritmo porque não acrescenta informação ao estudo, devido a obrigatoriedade de selecionar o valor K para servir de analise na avaliação dos algoritmos.

Ao nível do *clustering* existem mais algoritmos que não foram analisados porque a seleção do valor K não existe no algoritmo (DBSCAN e *Expectation Maximization Clustering*), não sendo possível a obtenção da avaliação dos algoritmos empregues nos dados com recurso ao método *confusion matrix*.

Outro método utilizado como alternativa da utilização do *clustering* para avaliação de *contextos de interação interpessoal* e *redes de contexto de interação* foi o uso do método de *Association Rules.* O processo da *Association Rules* descrito na secção 2.4.4 permitiu aplicar o algoritmo *FP-Growth*.

3.1.4. ORGANIZAÇÃO DOS DADOS

Uma vez selecionados os algoritmos, foi necessário organizar o ficheiro em formato *Microsoft Excel* ®, pois este encontrava-se desorganizado. A organização dependeu do objetivo pretendido para análise dos resultados. A primeira tarefa que pretendíamos era obter *clustering* das ações de apenas de cada *Actor_Sender,* obtenção dos contextos pessoais, o respetivo ID do cluster contem os seguintes dados das várias colunas do ficheiro *Excel*, formando assim uma única linha ou "frase", esta linha que é composta pelos itens pertencentes ao *Receiver*, *Action_Interecion*, *Related_Action*, *SubjectKeyword*, *Tools* e *Human_Resources*. Quer estas estejam ou não completas.

No caso da análise com recurso às *Association Rules*, os dados utilizados para a obtenção desta regra foram o o *Actor_sender*, *Receiver* e *Subject_Keywords*. Assim a seleção dos dados envolveu três colunas para a introdução dos mesmos na ferramenta, como mostra a seguinte figura.

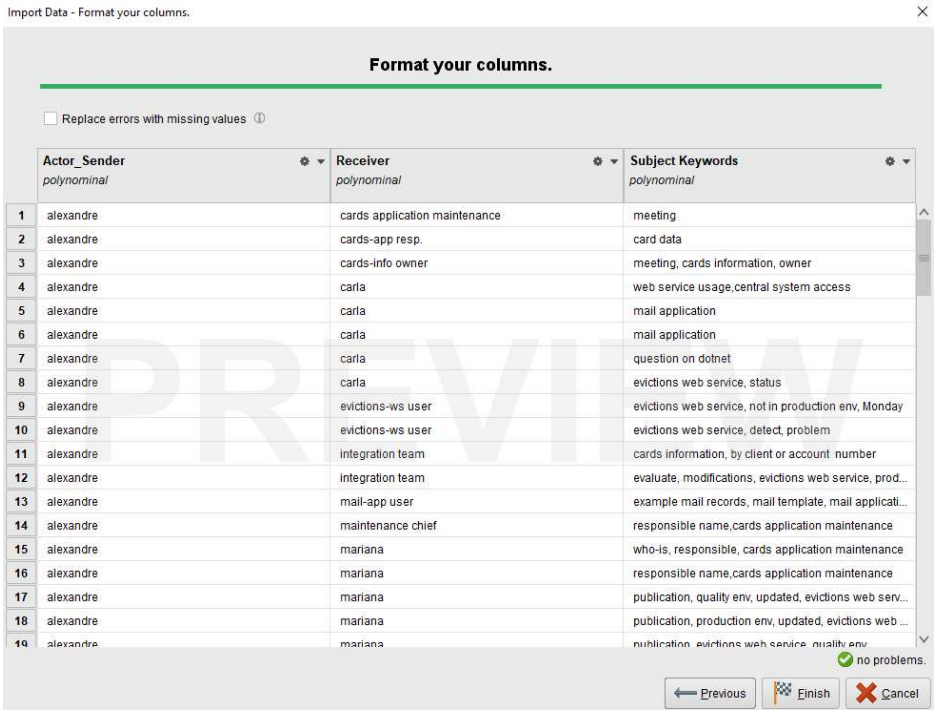


Figura 3.3: Carregamento de dados na ferramenta *Rapidminer Studio* ®

A utilização destas três colunas permitiu identificar o *Actor_Sender* (a pessoa que envia a mensagem) o *Receiver* (a pessoa que recebe a mensagem) e o *Subject Keywords* (o conteúdo da mensagem transmitida).

3.1.5. IMPLEMENTAÇÃO DO PROCESSO DE *CLUSTERING E ASSOCIATION RULES*

Após completar os passos da escolha da ferramenta, da seleção dos algoritmos e da organização dos dados, teve seguimento a implementação do processo de *clustering* com recurso a ferramenta *Rapidminer Studio* ®.

Num primeiro passo, o ficheiro dos dados estavam guardados em formato XLS e XLSX que pertencem ao software *Microsoft Excel* ®, que é um dos formatos aceites para carregar os dados através do operador *Read Excel* disponível no *Rapidminer Studio* ®. Pode ser visualizado através da Figura 3.4, a escolha do operador consoante o tipo de dados carregados.

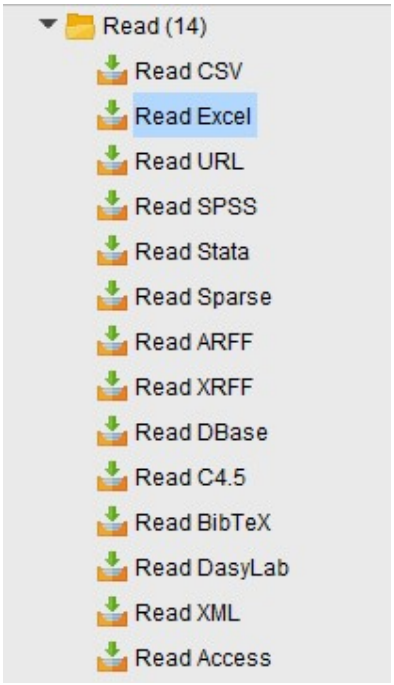
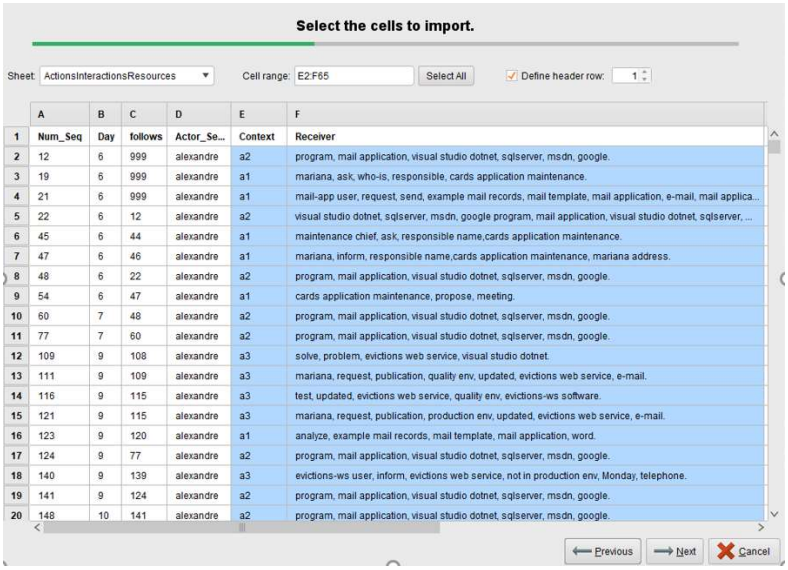


Figura 3.4: Operadores de carregamento de dados no *Rapidminer Studio* ®

Como descrito na secção anterior, utilizou-se os dados de cada ator (Alexandre, Carla, Catarina, Gonçalo e Mariana), e selecionou-se a coluna *Context* e a *Receiver.* Por limitações da ferramenta (valores ausentes nos dados detetados através dos algoritmos de *clustering*), na coluna *Receiver* foram incluídas nos dados as colunas *Receiver, Action_Interaction, Related_Action* e *Subject_Keyword*., De seguida procedeu-se à importação dos dados.



	A	B	C	D	E	F
1	Num_Seq	Day	follows	Actor_Se...	Context	Receiver
2	12	6	999	alexandre	a2	program, mail application, visual studio dotnet, sqlserver, msdn, google.
3	19	6	999	alexandre	a1	mariana, ask, who-is, responsible, cards application maintenance.
4	21	6	999	alexandre	a1	mail-app user, request, send, example mail records, mail template, mail application, e-mail, mail applica...
5	22	6	12	alexandre	a2	visual studio dotnet, sqlserver, msdn, google program, mail application, visual studio dotnet, sqlserver, ...
6	45	6	44	alexandre	a1	maintenance chief, ask, responsible name,cards application maintenance.
7	47	6	46	alexandre	a1	mariana, inform, responsible name,cards application maintenance, mariana address.
8	48	6	22	alexandre	a2	program, mail application, visual studio dotnet, sqlserver, msdn, google.
9	54	6	47	alexandre	a1	cards application maintenance, propose, meeting.
10	60	7	48	alexandre	a2	program, mail application, visual studio dotnet, sqlserver, msdn, google.
11	77	7	60	alexandre	a2	program, mail application, visual studio dotnet, sqlserver, msdn, google.
12	109	9	108	alexandre	a3	solve, problem, evictions web service, visual studio dotnet.
13	111	9	109	alexandre	a3	mariana, request, publication, quality env, updated, evictions web service, e-mail.
14	116	9	115	alexandre	a3	test, updated, evictions web service, quality env, evictions-ws software.
15	121	9	115	alexandre	a3	mariana, request, publication, production env, updated, evictions web service, e-mail.
16	123	9	120	alexandre	a1	analyze, example mail records, mail template, mail application, word.
17	124	9	77	alexandre	a2	program, mail application, visual studio dotnet, sqlserver, msdn, google.
18	140	9	139	alexandre	a3	evictions-ws user, inform, evictions web service, not in production env, Monday, telephone.
19	141	9	124	alexandre	a2	program, mail application, visual studio dotnet, sqlserver, msdn, google.
20	148	10	141	alexandre	a2	program, mail application, visual studio dotnet, sqlserver, msdn, google.

Figura 3.5: Seleção dos dados a importar no *Rapidminer Studio* ®

A figura 3.6 mostra o processo inicial de parametrização de dados desenvolvidos no *Rapidminer Studio* ®.

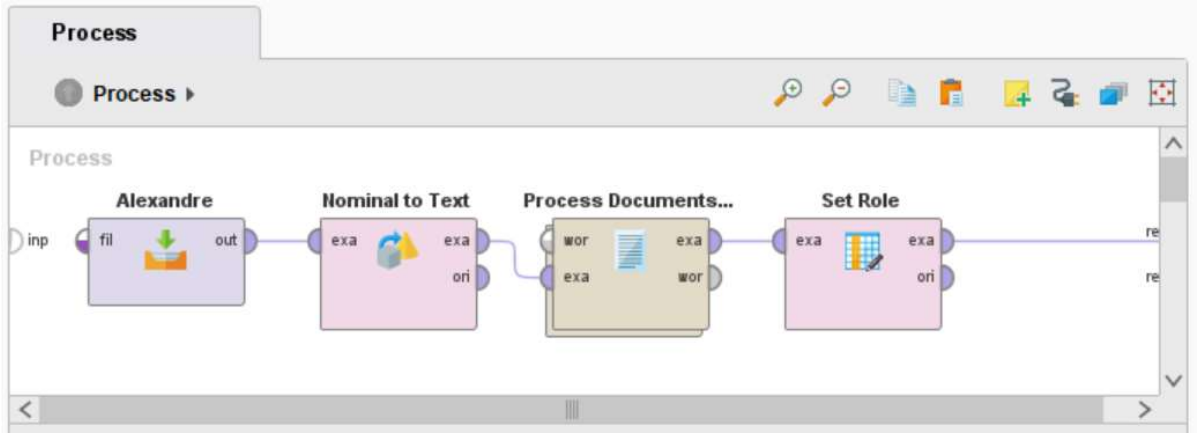


Figura 3.6: Etapas de pré-processamento do *Rapidminer Studio* ®

Após o carregamento dos dados é necessário realizar o tratamento dos dados, a isto chamamos de pré-processamento, como foi explicado na secção 2.3., é necessário instalar um *plugin* na ferramenta *Rapidminer Studio* ® com o nome de *Text Processing*. Antes da utilização dos operadores de processamento de texto é necessário converter os dados, aos dados introduzidos estes estão no formato nominal, que representa uma variável binária e que pode ter mais de dois estados não textuais [72], por forma a garantir que os dados introduzidos estão na forma de *string* utilizamos o operador Nominal to Text [73], este permite converter dos dados introduzidos para o tipo de texto em que os operadores de pré-processamento possam aplicar as suas técnicas nos dados de texto.

No operador *Process Documents from Data*, aqui é realizado toda a operação do tratamento de texto, como explicado na secção 2.4. A seguinte figura mostra os operadores utilizados para o tratamento da informação introduzida.

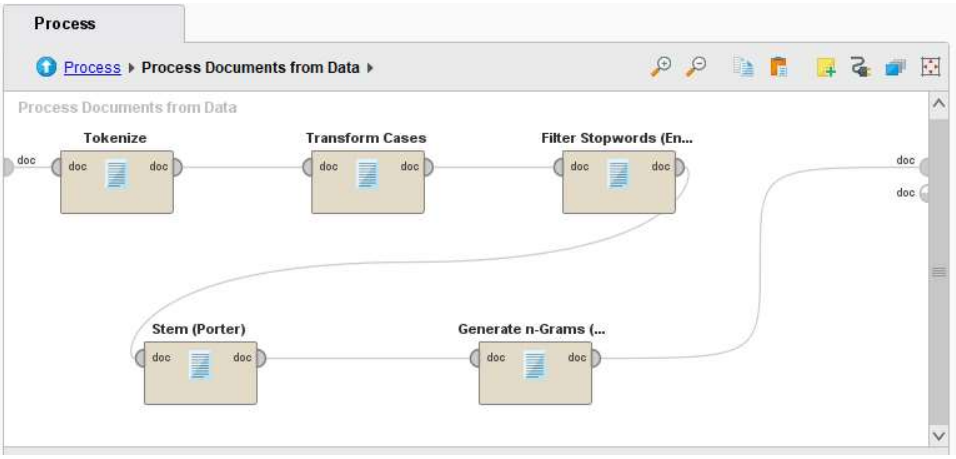


Figura 3.7: Operadores de pré-processamento de documentos

O *Tokenize* como descrito na secção 2.3.1, permite definir a utilização dos dados introduzidos, o término do *subject keywords*, para o nosso caso utilizou-se o *mode*: *specify characters*, isto significa que as frases do *subject keywords* estão definidas para terminar quando deteta o ponto final na frase. A figura 3.8 mostra a seleção destes parâmetros:

Figura 3.8: Parâmetros do *Tokenize*

O *Transform Cases* permite transformar todas os carateres em maiúsculas ou minúsculas, assim como foi explicado na secção 2.3.5, este evita que uma palavra com o mesmo significado seja interpretada pela ferramenta como uma palavra diferente por este apenas ter uma ou mais letras maiúsculas/minúsculas.

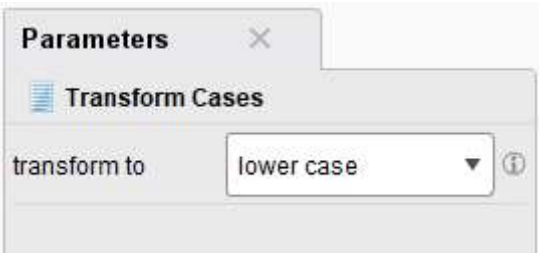


Figura 3.9: Parâmetro de *Transform Cases*

Estes parâmetros descritos na secção 2.3.2 tem a utilidade de simplificar a interpretação da ferramenta utilizada, o *filter stopwords* vai remover palavras sem significados, tais como, *in* ou *or*, que são palavras que não acrescentam significado aos resultados. Já o parâmetro descrito e como explicado na secção 2.3.3, este tem por objetivo eliminar plurais, gerúndio, prefixos, sufixos, género e números, dando valor ao corpo da palavra.

Figura 3.10: Parâmetro *Filter Stop*

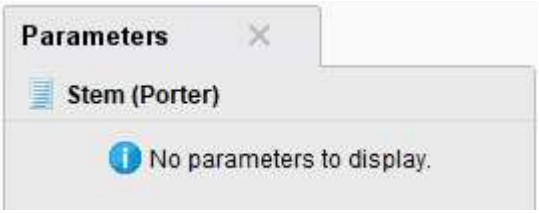


Figura 3.11: Parâmetro *Stem*

O parâmetro do *Generate n-Grams (Terms)* e o *Generate n-Grams (Carateres)* descrito na secção 2.3.4 permite escolher o tamanho de carateres ou termos que pretendemos utilizar de amostra única para o algoritmo, inicialmente utilizou-se o *Generate n-Grams (Terms)* e posteriormente o *Generate n-Grams (Carateres)*.

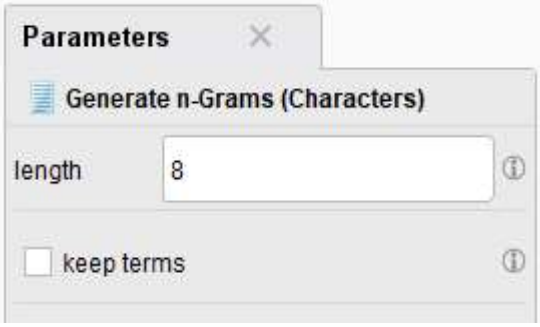


Figura 3.12: Parâmetro *Characters*

Figura 3.13:Parâmetro *Terms*

Após a aplicação de técnicas de processamento, utilizou-se o operador *Set Role*, antes de aplicar qualquer ação de *clustering*, este permite dizer qual a *label* ou rótulo que pretendemos usar como forma de comparação, e assim podermos ter um termo na avaliação dos dados. Os parâmetros existentes no operador são de categorizar os atributos, as categorizações disponíveis são: regular, id, *label*, *prediction*, *cluster*, *weight*, *batch*. O parâmetro utilizado para definir que a identificação dos *clusters* que nos nossos dados são caracterizados por *context*, define os nomes do *cluster*, assim é possível obter a comparação dos resultados com a previsão e assim conseguimos obter a os resultados da *confusion matrix*. O parâmetro *label* indica qual o atributo e a classe prevista para os operadores de modelagem [74].

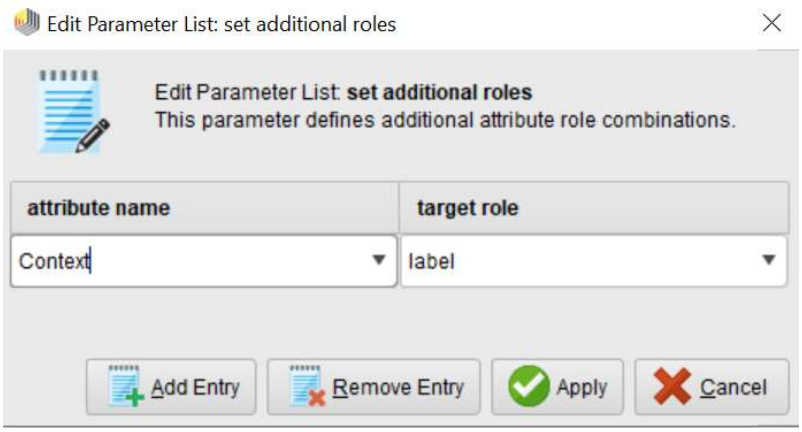


Figura 3.14: Parâmetro do Set Role

O operador *Set Role* foi utilizado na obtenção de contexto de ação pessoal e em contextos de interação interpessoal utilizando o método de *clustering*, que avaliamos os resultados em comparação da coluna do *context* dos dados introduzidos.

Na utilização do método de *association rules* não se utilizou o operador *Set Role* porque a descoberta das redes de contexto de interação apenas permite a visualização dos dados (rede de ações semelhantes) e a avaliação acontece de forma comparativa dos resultados obtidos da ferramenta *Rapidminer Studio* ® com os resultados obtidos pela autora [7].

3.2. ANÁLISE E OBTENÇÃO DE RESULTADOS

3.2.1. PARAMETRIZAÇÃO DOS ALGORITMOS DA FERRAMENTA

Nesta secção iremos visualizar os parâmetros disponíveis nos algoritmos de *clustering* descritos na secção 3.1.3. Todos os algoritmos, à exceção do algoritmo *Aglomerative Hierarquical Clustering*, apresentam os mesmos campos de escolha dos parâmetros e apenas necessitam de um operador para obtenção de *clustering*. No caso do algoritmo *Aglomerative Hierarquical Clustering,* este possui parâmetros diferentes dos restantes e utiliza dois operadores, um para a escolha das medidas de similaridade e outro para a terminação do número de *clusters*. A figura 3.15 mostra a listagem de algoritmos de *clustering* disponíveis na ferramenta *Rapidminer Studio* ®.

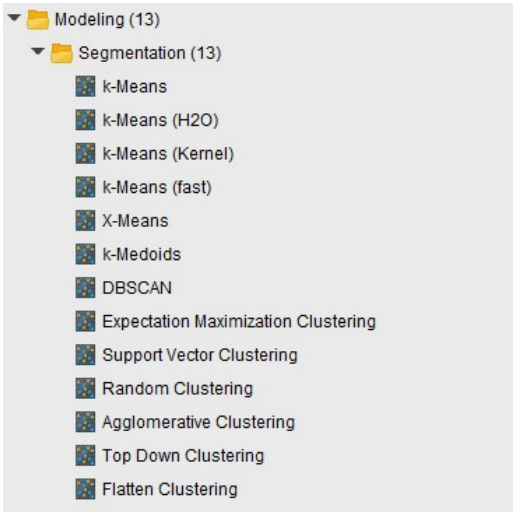


Figura 3.15: Algoritmos de *Clustering* disponíveis no *Rapidminer Studio* ®

Na figura 3.16 conseguimos visualizar os vários parâmetros disponíveis no algoritmo *K-means*, sendo estes iguais para os restantes algoritmos, exceto para o algoritmo *Agglomerative clustering*, que tem a mesma função do algoritmo descrito na secção 2.5.2.1.

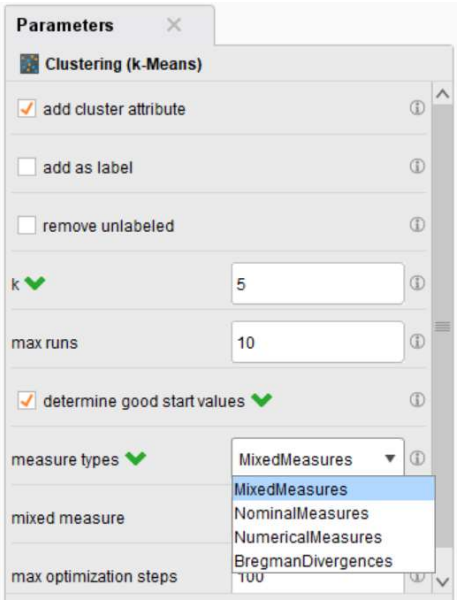


Figura 3.16: Opções do algoritmo *K-means* de *Clustering*

Os algoritmos que apresentam a mesma estrutura de opções do *k-means* e estas são: (1) *add cluster atribute*: ocorre um novo atributo designado por *cluster*, com um *cluster_id* para cada exemplo gerado, ou seja é criado uma função especial de *cluster*, no caso de (2) *add as label:* se for selecionado, o novo atributo é denominado como rótulo e tem a função especial de rótulo [75], neste caso o rótulo é definido no operador *Set Role*, logo esta opção não é selecionada, porque não queremos que os atributos *keywords* sejam do tipo *label*. Os únicos atributos do tipo *label* foram selecionados pelo operador *Set Role* e definiu-se os atributos do *context*, (3) *remove unlabeled*: esta opção permite que os exemplos atribuídos a um *cluster* sejam

removidos no *Example Set*, (4) *K*: Número pré-definido de *clusters* a ser criado pelo algoritmo. (5) *Max Runs*: É indicado o número máximo de execuções do algoritmo com inicializações aleatórias dos pontos pré-definidos. (6) *max optimization steps*: Parâmetro que especifica um número máximo de iterações realizadas pelo algoritmo. (7) *determine good start values*: este é um ponto importante na obtenção dos resultados, esta utiliza uma técnica chamada *K-means*++ que é um melhoramento da precisão do *K-means*, explicado na secção 2.4.2.2. (8) *measure types*: Este parâmetro é utilizado para escolher qual o tipo de medida de similaridade a ser usado para encontrar os vizinhos mais próximos.

No caso da implementação do algoritmo *Aglomerative Hierarchical Clustering* este implementa dois operadores, como mostra a figura 3.17.

Figura 3.17: Operadores do algoritmo *Aglomerative Hierarchical Clustering*

Os parâmetros disponíveis são (1) *mode*: Este permite especificar o critério de ligação entre *clusters* e encontra-se explicado na secção 2.5.2.1, já (2) *measure types:* permite selecionar os diversos tipos de parâmetros.

3.3. OBTENÇÃO DOS RESULTADOS DE *CLUSTERING*

A apresentação dos resultados de *clustering* do *Rapidminer Studio* ® são ilustrados nas figuras desta secção.

No primeiro separador indicado como *Description*, é identificado o número de itens de cada *cluster*, este indica de uma maneira geral a quantidade de itens existentes em cada *cluster*, os números de *clusters* existentes e o número total de itens em estudo.

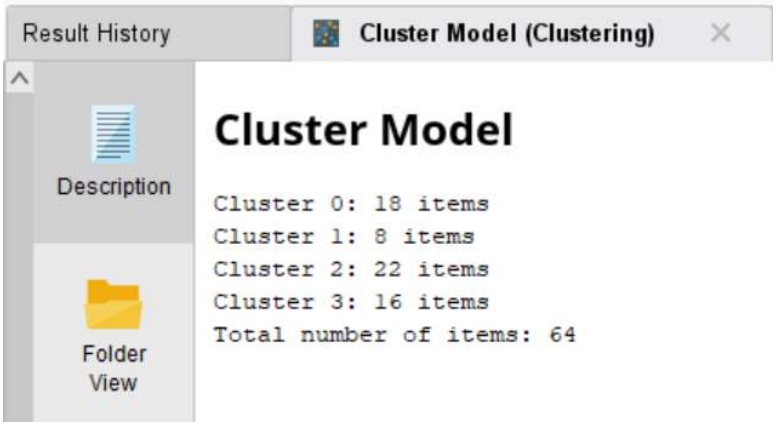


Figura 3.18: Visualização da descrição

O segundo separador (*Folder View)*, permite visualizar os atributos ou itens que ficaram atribuídos a cada pasta de *cluster*, esta vista possibilita a observação do conteúdo de cada item associado aos respetivos *clusters*.

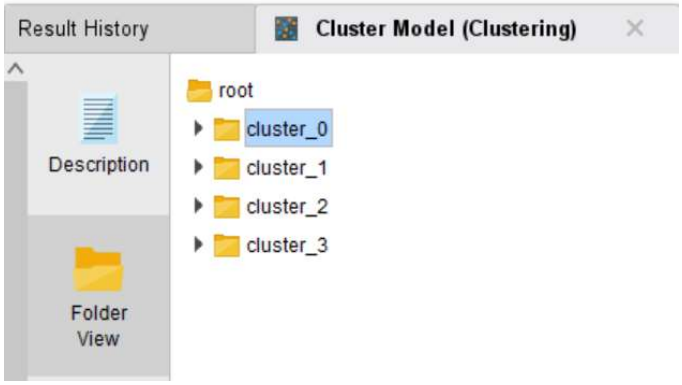


Figura 3.19: Visualização do *Folder View*

Dentro de cada pasta é possível visualizar os atributos, ou seja, as frases que foram dividas para cada *cluster*, estas estão numeradas com um ID que foi atribuído pelo próprio algoritmo de *cluster*.

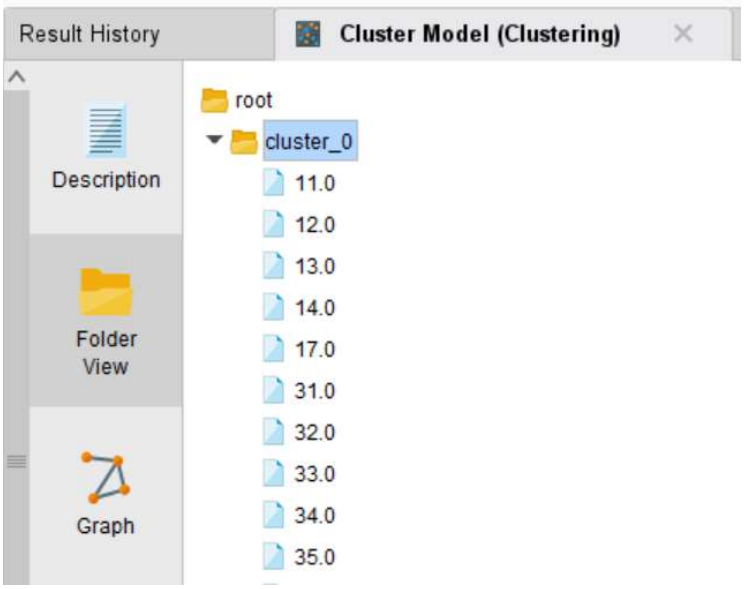


Figura 3.20: Visualização dos itens dentro da *Folder View*

A opção *Graph* é uma representação gráfica dos vários *clusters*. O *Root Set* representa o conjunto total dos dados, os círculos menores indicam os respetivos agrupamentos realizados pelo operador. Quanto maior for o tamanho do círculo, mais frases/palavras são atribuídas ao respetivo cluster. No lado direito da imagem é possível observar os ID dos atributos que foram guardados nos *clusters*, no caso da imagem abaixo, o *cluster* 1 possui oito atributos.

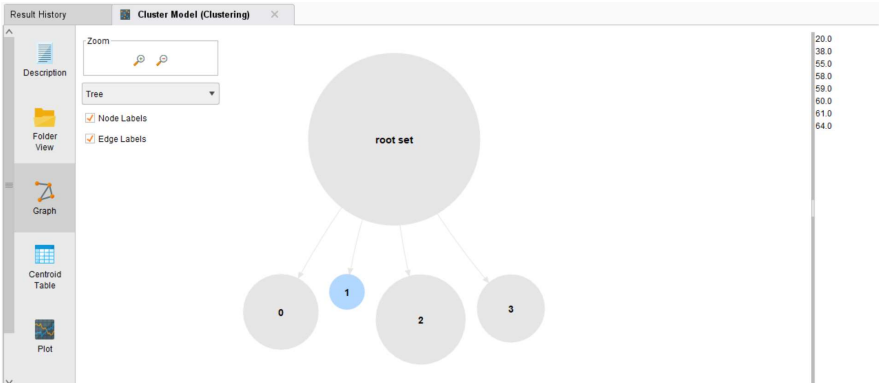
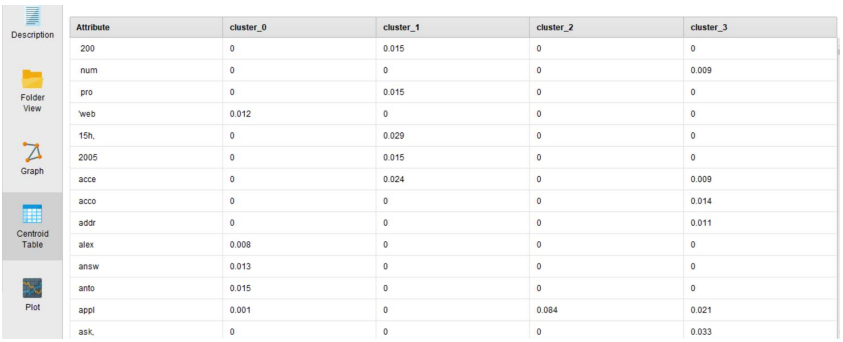


Figura 3.21: Visualização gráfica do *Graph*

De seguida temos o *Centroid Table*, como o próprio nome indica este identifica os atributos e mostra o grau de frequência que cada atributo e a relação ao *cluster* associado.



Attribute	cluster_0	cluster_1	cluster_2	cluster_3
200	0	0.015	0	0
num	0	0	0	0.009
pro	0	0.015	0	0
'web	0.012	0	0	0
15h,	0	0.029	0	0
2005	0	0.015	0	0
acce	0	0.024	0	0.009
acco	0	0	0	0.014
addr	0	0	0	0.011
alex	0.008	0	0	0
answ	0.013	0	0	0
anto	0.015	0	0	0
appl	0.001	0	0.084	0.021
ask,	0	0	0	0.033

Figura 3.22: Visualização do *Centroid Table*

O último separador dos resultados é o *plot*, este de uma forma gráfica é possível verificar as frequências dos atributos das frases no cluster, de certa forma é a mesma representação que a tabela apresentada no *Centroid Table*.

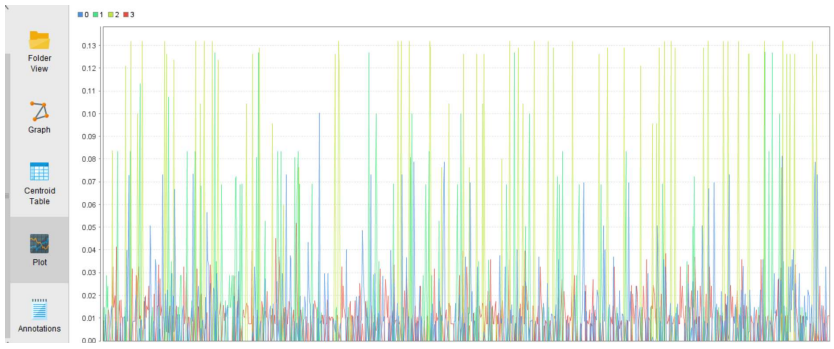


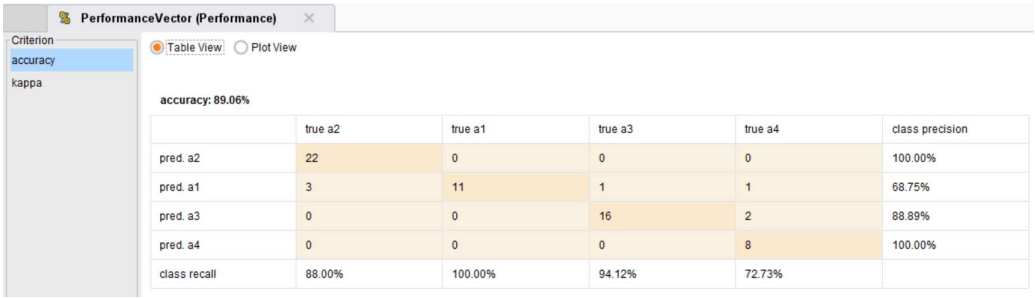
Figura 3.23: Demonstração do *Plot*

3.3.1. PARAMETRIZAÇÃO DO *CONFUSION MATRIX* NO *RAPIDMINER STUDIO* ®

O *confusion matrix* é utilizado para a avaliação dos resultados de *clustering* através da comparação dos agrupamentos formados pelos algoritmos com os agrupamentos realizados manualmente e identificados com o atributo da coluna *context* dos dados introduzidos.

As métricas utilizadas para avaliação dos nossos dados são *a accuracy* e o *kappa* e foram descritas na secção 2.6.1.

A figura 3.24 mostra o *matrix confusion* que nos indica o *accuracy* e o *kappa*, tal como a matriz demostra os *clusters* previstos e os *clusters* verdadeiros, as percentagens mostradas indicamos o valor de acerto relativamente aos valores previsto pela ferramenta e do operador *set role* introduzido que é o *context* dos dados introduzidos, o cruzamento destes dois permite avaliar a qualidade dos algoritmos.



	true a2	true a1	true a3	true a4	class precision
pred. a2	22	0	0	0	100.00%
pred. a1	3	11	1	1	68.75%
pred. a3	0	0	16	2	88.89%
pred. a4	0	0	0	8	100.00%
class recall	88.00%	100.00%	94.12%	72.73%	

Figura 3.24: Visualização do *Confusion Matrix*

3.3.2. PARAMETRIZAÇÃO DA *ASSOCIATION RULES* NO *RAPIDMINER STUDIO* ®

Outra técnica utilizada foi a *Association Rules*, descrita na secção 2.4.4 que demonstra o funcionamento do mesmo. Existem diversos algoritmos sendo os mais conhecidos o *Apriori* que se encontra descrito na secção 2.4.4.1 e o *FP-Growth* na secção 2.4.4.2 que mostram o funcionamento destas técnicas.

Neste trabalho explorou-se o único algoritmo disponibilizado pela ferramenta *Rapidminer Studio* ®.

Após a fase inicial, em que se introduziu os dados como explicado na secção 3.1.5, para este caso introduziu-se o conjunto de todos os atores.

Os dois operadores utilizados pela ferramenta *Rapidminer Studio* ® foram o *FP-Growth* e o *Create Association Rules.*

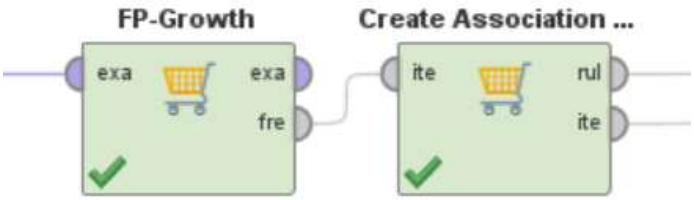


Figura 3.25: Operadores da *Association Rules*

O *FP-Growth* permite indicar os itens frequentes de um conjunto de dados de entrada, descrito na secção 2.4.4.2.

Para obtermos os resultados aceitáveis e confiáveis a partir dos dados inseridos, é preciso parametrizar o algoritmo *FP-Growth*, assim começamos por observar as definições disponíveis deste algoritmo. O primeiro parâmetro, o *(1) Input Format:* permite escolher qual é o formato de dados inseridos no algoritmo e deixa em escolha três opções: (i) Lista de itens em uma coluna: Indica que os itens aparecem em uma única coluna. (ii) Itens separados por colunas: Como a própria opção indica, cada item pertence apenas a uma coluna, ou seja, o

primeiro item pertence à primeira coluna, o segundo item à segunda coluna e assim sucessivamente. (iii) Itens em colunas codificadas fictícias: Neste caso cada item num conjunto de todos os itens tem a sua própria coluna, em que o nome do item é o nome da coluna, as colunas estão introduzidas com valores binominais (*True/False*), indicando que cada item pode ser encontrado no conjunto de dados [76].

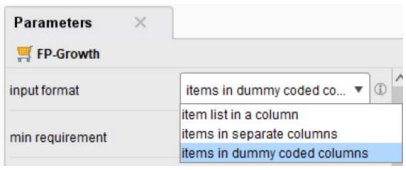


Figura 3.26: Parametrização do *FP-Growth* (*input format*)

(2) *min requirement:* Tem disponível duas opções, o (i) *Support e o (ii) Frequency*, descritos na secção 2.6.2. Ao escolher (3) *min Support* este é calculado pela percentagem do número de ocorrências de um conjunto de itens a dividir pelo tamanho total de dados de entrada. Já o (4) *min frequency* é obtido pela frequência mínima de ocorrências. Quanto mais diminuirmos os valores destes dois parâmetros, maior serão os números de conjuntos de itens nos resultados, o que para análise do utilizador dificultará a análise dos mesmos.

Por fim, temos três parâmetros disponíveis todos estes sobre os itens, o (5) *min items per itemset*: Define o numero mínimo de itens a apresentar nos resultados, para o nosso caso como se pretende observar a iteração do *Sender* com o *Receiver* e a informação (*Keywords*) comunicada entre ambos, (6) *max items per itemset*: Define o numero máximo de itens a apresentar nos resultados, este não é importante para os nosso resultado, porque nos dados introduzidos não revela mais de 3 itens, mas caso pretendêssemos obter menos itens associados no conjunto de itens, seria neste parâmetro a introduzir o valor, no caso da configuração estabelecida por padrão utiliza-se o valor 100, (7) *max number of itemsets*: Define o limite superior para o numero de conjuntos de itens, assim colocamos o valor de zero que indica que não existe limite superior. A figura a seguir mostra o ambiente gráfico dos parâmetros do algoritmo *FP-Growth* [76].

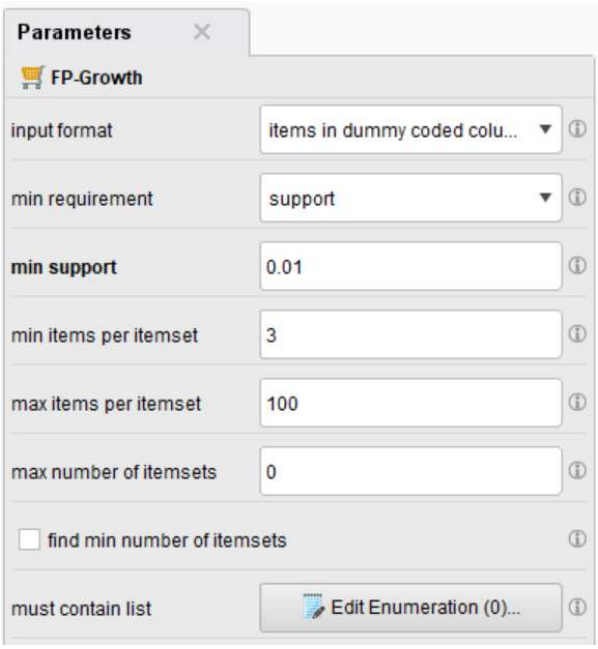


Figura 3.27: Parametrização do *FP-Growth* (geral**)**

O *Association Rules* apresenta nos parâmetros de avaliação dos dados diversas opções, que são, (1) *criterion*: É a área de seleção das regras que caraterizar os resultados, (i) *Confidence:* É a regra de confiança que se define a probabilidade de acontecer o item Y dado o item X, este tem uma variação em percentagem que varia de 0,0 a 1,0, (ii) *conviction*: este é sensível à direção da regra, ou seja, se X implica Y poderá ou não ser que Y implique X. É inspirada na definição lógica de implicação e tenta medir o grau de implicação da regra. Ainda existe o *lift*, *gain*, *laplace* e o *ps* [77].

Figura 3.28: Parametrização do *Create Association Rules*

4. RESULTADOS

4.1. DESCOBERTA DO MODELO

Nesta secção são apresentados os resultados relativos à (1) descoberta automática de contextos pessoais, (2) contextos interpessoais e redes de interação. O primeiro caso foi realizado com recurso à algoritmos de *clustering*. Os resultados de cada algoritmo de *clustering* foram avaliados e comparados utilizando a técnica do *Confusion Matrix*. No segundo caso, recorreu-se à técnica de *Association rules*.

4.1.1. RESULTADOS DE *CLUSTERING* NA DESCOBERTA DE CONTEXTOS PESSOAIS

No decurso da execução dos algoritmos de *clustering*, podemos observar como são atribuídos os elementos a cada cluster. A figura 4.1, verificamos o respetivo atributo e a estimativa em valor decimal pertence a cada cluster, neste caso os atributos estão com o parâmetro de pré-processamento *Generate n-grams (termos)* descrito na secção 2.4.4.

Attribute	cluster_0	cluster_1	cluster_2	cluster_3
mariana, request, publication, evictions web service, quality env	0.026	0	0	0
mariana, request, publication, production env, updated, evictions web service, e-...	0.026	0	0	0
mariana, request, publication, quality env, updated, evictions web service, e-mail	0.026	0	0	0
meet, cards information owner, handle returned mail	0.026	0	0	0
meet, discuss, evictions web service, problem, status, mariana, antonio	0.026	0	0	0
meet, team meeting, goncalo, mariana, carla, catarina	0.026	0	0	0
meet, team meeting, mariana, goncalo, catarina,carla	0	0	0	1
program, mail application, visual studio dotnet, sqlserver, msdn, googl	0	0	1	0
reject, team meeting, todai	0.026	0	0	0
research, evictions web service, problem	0.026	0	0	0
solve, problem, evictions web service, visual studio dotnet	0.026	0	0	0
test, evictions web service, production env, success	0.026	0	0	0
test, evictions web service, quality env, success, 'web servic	0.026	0	0	0
test, updated, evictions web service, quality env, evictions-ws softwar	0.026	0	0	0

Figura 4.1: Resultados de *clustering* para o *Generate n-grams (termos*) do algoritmo *K-means*

Na figura 4.2 visualizamos a obtenção dos resultados de *clustering* para o parâmetro *Generate n-grams (carateres).*

Attribute	cluster_0	cluster_1	cluster_2	cluster_3
acces	0	0	0	0.011
accou	0	0	0	0.013
addre	0	0	0	0.010
alexa	0.007	0	0	0
answe	0.013	0	0	0
anton	0.015	0	0	0
appli	0.002	0	0.095	0.014
ask,	0	0	0	0.031
avail	0	0	0	0.015
by ca	0	0	0	0.007
by cl	0	0	0	0.013
calli	0.008	0	0	0
card	0	0	0	0.015

Figura 4.2: Resultados de *clustering* para o *Generate n-grams (carateres*) do algoritmo *K-means*

Com a obtenção dos resultados descritos anteriormente, validamos a partir do *confusion matrix* a qualidade da medição da *accuracy* e do valor *kappa*, a fim de observar os valores máximos obtidos pela qualidade de medição dos algoritmos utilizados. Como podemos visualizar na figura 4.3.

accuracy: 64.06% **kappa: 0.486**

	true a2	true a1	true a3	true a4	class precision
pred. a2	21	0	0	0	100.00%
pred. a1	1	0	0	0	0.00%
pred. a3	3	11	17	8	43.59%
pred. a4	0	0	0	3	100.00%
class recall	84.00%	0.00%	100.00%	27.27%	

Figura 4.3: Resultado da *matrix confusion* do algoritmo *K-means* para o Ator Alexandre

Na obtenção dos resultados utilizou-se o valor no parâmetro *max run* de 100, assumindo que os algoritmos estabilizam as suas iterações neste valor.

Para os restantes algoritmos, os resultados dos elementos atribuídos seguem o mesmo padrão, como já visualizado no algoritmo *K-means*.

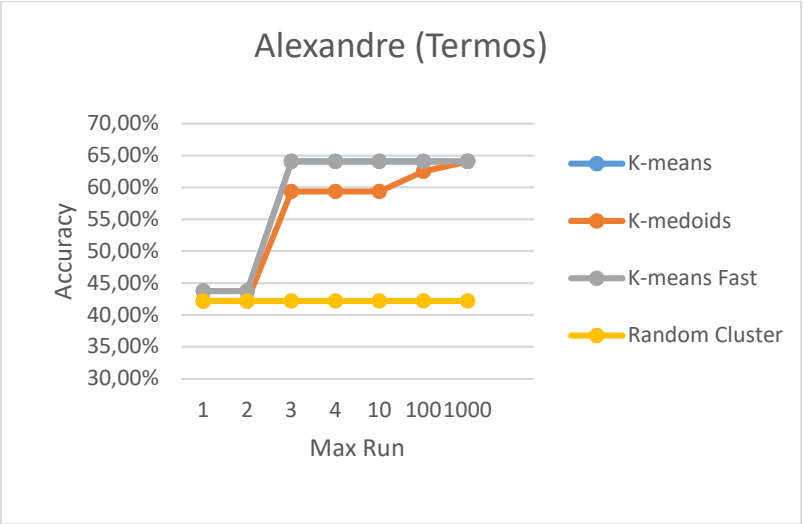


Figura 4.4:*Accuracy* dos primeiros 4 algoritmos de *clustering* usando termos para o Ator Alexandre

Visualizando a figura 4.4, que se refere à obtenção dos *clusters* utilizados apenas com os registos das tarefas desenvolvidas pelo programador Alexandre, estes formam quatro *clusters*, apresentando uma tendência na medição da *accuracy* máxima em torno dos 64,06%. Isto indica que os *clusters* gerados foram previstos e produzidos em quatro *clusters*, verificando que os algoritmos *K-means* e a variante *K-means Fast* atingiram o valor mais elevado de acertos (previstos e verdadeiros), admitindo que o algoritmo estabiliza a formação de cluster em *max run* igual a 100. Já o valor *kappa* estimado para o *K-means* e o *K-means fast* obteve-se um valor de 0.486, ou seja, analisando a tabela 4 da secção 2.4.3.1, verificamos que este resultado tem uma concordância moderada, ou seja, está num grau quarto de seis possíveis.

Utilizando agora outra variante no pré-processamento, o *Generate n-grams* (carateres), para o mesmo ator obtivemos a seguintes resultados.

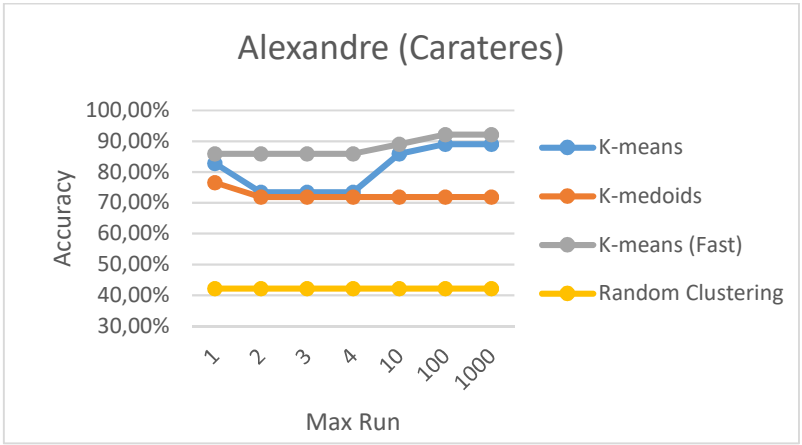


Figura 4.5: *Accuracy* dos primeiros 4 algoritmos de *clustering* usando carateres para o Ator Alexandre

A partir da figura 4.5 é possível notar que a *accuracy* melhorou significativamente em todos os algoritmos de *clustering*. Continuamos a ter o *k-means* e o *k-means fast* como melhores resultados no que respeita a medida da *accuracy*. O *k-means* a obter uns 89,62% de acerto na *confusion matrix* enquanto que o *k-means fast* obtém um resultado ligeiramente melhor, chegando aos 90,62% de acerto. Com estes valores podemos verificar a qualidade do coeficiente *kappa* para o algoritmo *k-means* e para o *k-means fast*, assim temos como valores de *kappa,* respetivamente 0.849 e 0.892, que significa que temos um nível de concordância perfeito, ou seja o nível mais alto de concordância, valor este correspondente ao intervalo entre 0,81 e 1. Em sentido inverso, temos o algoritmo *random clustering* que apresenta o pior resultado, neste caso tem uma medida abaixo dos 50% mais propriamente um valor de 42,19%, isto corresponde a um valor *kappa* de 0.224, o que indica um nível de concordância razoável, ou seja encontra-se no terceiro grau.

De seguida, temos um novo ator, neste caso a Carla, este apresenta a formação de três *clusters*, assim a nossa *confusion matrix* apresentará três variáveis.

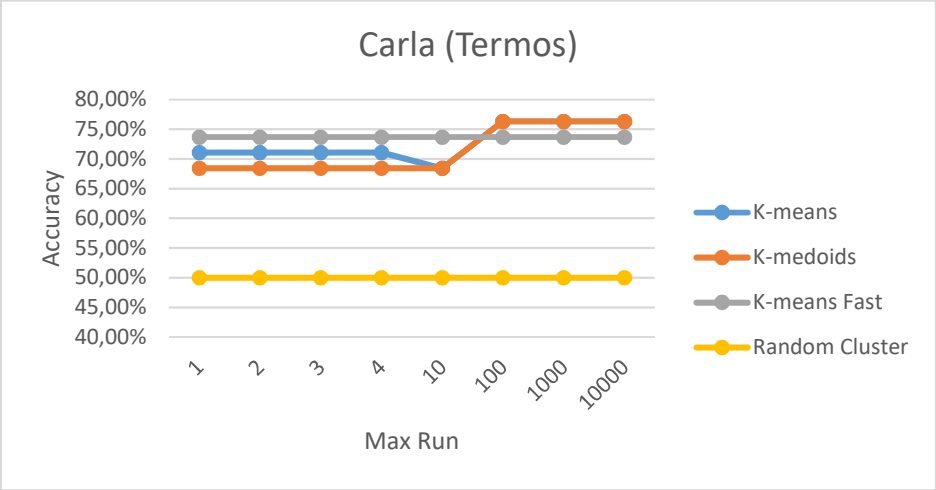


Figura 4.6: *Accuracy* dos primeiros 4 algoritmos de *clustering* usando termos para o Ator Carla

Na figura 4.6 verifica-se a obtenção de *clusters* utilizando apenas os registos das tarefas desenvolvidas pelo programador Carla que apresenta uma tendência na medição da *accuracy* máxima em torno dos 76,32%, verificou-se que os algoritmos *k-medoids* e o *k-means* obtiveram os valores mais altos a partir do *max run* 100, podemos então identificar que a "maturidade" foi conseguida no valor mais alto. O *k-means fast* que atingiu o valor com cerca de 73,68%, apresentou-se com uma *accuracy* constante para qualquer *max run*. Já o valor *kappa* estimado para o *k-medoids* e *k-means* com um *max run* de 100 tem um valor de 0.608,

respeitando a tabela do coeficiente *kappa* garantimos que o valor se encontra no intervalo entre 0,41 e 0,60 e assim verificamos que este resultado tem uma concordância moderada.

Utilizando agora outra variante, o *Generate n-grams* (carateres), para o mesmo ator obtemos os seguintes resultados.

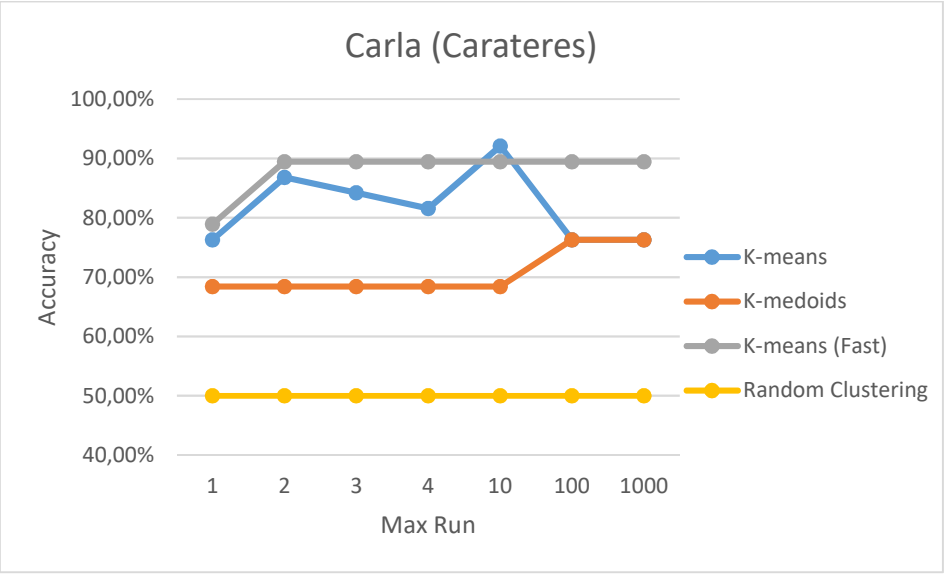


Figura 4.7: *Accuracy* dos primeiros 4 algoritmos de *clustering* usando carateres para o Ator Carla

Na figura 4.7 notamos que a *accuracy* melhorou significativamente na maioria dos algoritmos de *clustering*. O *k-means fast* apresenta melhores resultados e com estabilidade no que respeita a medida de *accuracy* obtendo 89,47%, o *k-means* foi o que obteve o valor mais alto, mas de forma inconstante, apesar de atingir os 92,11% de acerto na *confusion matrix*, este valor não é confiável, porque ainda não tinha ocorrido as várias interações para se obter um valor estável. A partir do valor estável que é o *max run* em 100, este sofreu uma queda percentual para os 76,32% igualando o máximo atingido pelo *k-medoids*. Analisando o valor de *kappa* máximo obtido que foi no algoritmo *k-means fast* de 0.824, assim temos um nível de concordância perfeita.

A seguir temos a programadora Catarina, esta colaboradora apresenta a formação de seis *clusters* para os seus registos. A figura seguinte mostra os respetivos resultados.

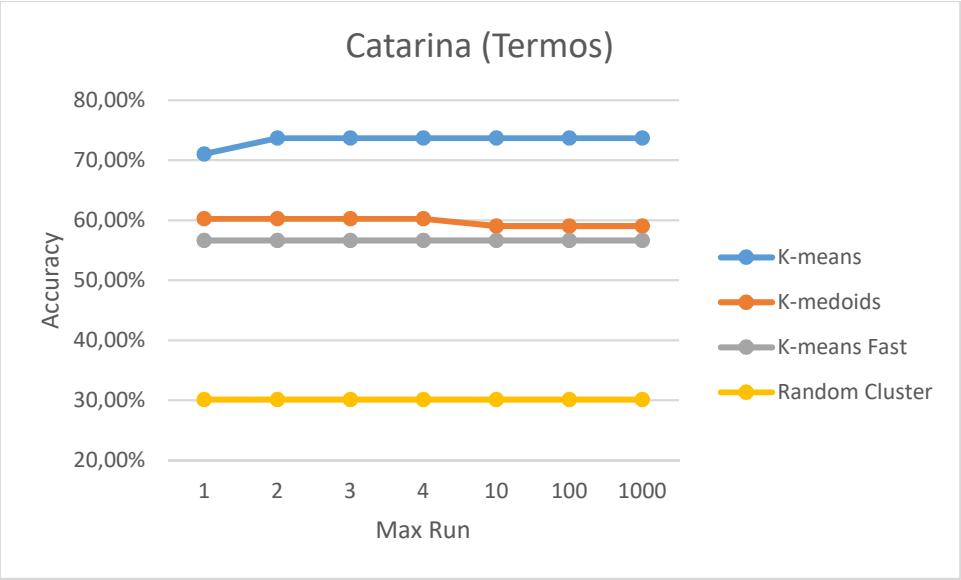


Figura 4.8: *Accuracy* dos primeiros 4 algoritmos de *clustering* usando termos para o Ator Catarina

Analisando a figura 4.8 o gráfico permite logo destacar que o algoritmo *k-means* apresenta o melhor resultado, acima de 70% com um valor exato de 73,68%, analisando o coeficiente *kappa* temos um valor de 0,679 obtendo um nível de concordância substancial, um nível abaixo do nível máximo. Em sentido contrário, temos o algoritmo *Random Clustering* que apresenta uma *accuracy* baixa, com um valor de 30,12%, com isto, verificando o valor *kappa* de 0.159 correspondente a concordância mínima.

Voltando ao programador Catarina, mas desta vez ao uso do parâmetro de pré-processamento *Generate n-gram* (carateres).

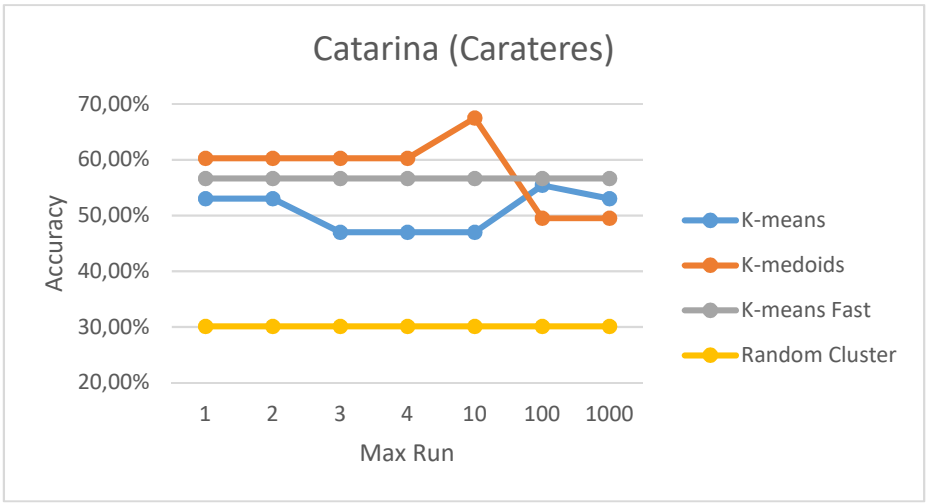


Figura 4.9: *Accuracy* dos primeiros 4 algoritmos de *clustering* usando carateres para o Ator Catarina

A figura 4.9 indica os valores mais instáveis comparado com o gráfico da Catarina (termos), começamos pelo *k-means* que teve várias variações e apresentou o valor mais alto, mas depois desceu a *accuracy* a partir de 100, o único algoritmo estável é o *k-means fast* apresentando um valor continuo para qualquer *max run* que foi de 56,63%, outro algoritmo que apresentou instabilidade foi o *k-medoids*, apesar de tem um pico de *accuracy* de 67,47% baixou para 49,50% para valores a partir de *max run* igual ou superior a 100. Em relação à medida *kappa* temos o algoritmo *k-means fast* que apresenta um valor de 0,299, que nos indica uma concordância razoável.

Analisando agora o programador Gonçalo com a utilização do parâmetro *Generate n-grams* (termos).

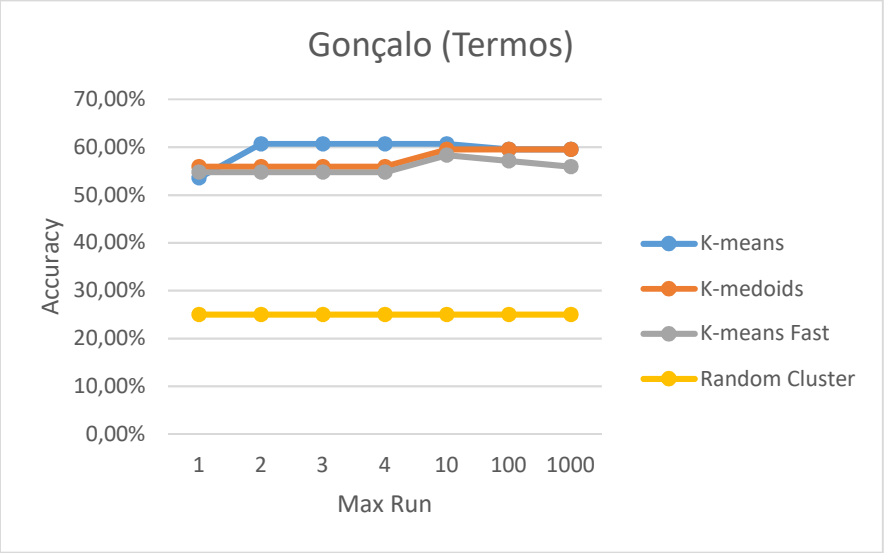


Figura 4.10: *Accuracy* dos primeiros 4 algoritmos de *clustering* usando Termos para o Ator Gonçalo

A figura 4.10 apresenta a criação de seis *clusters*, interpretando os valores obtidos pela avaliação da *accuracy,* permite verificar que os três algoritmos tem valores estáveis acima de 50%, mais propriamente os algoritmos *k-means* e o *k-medoids* que tem um valor máximo de 59,52%, abaixo deste valor aparece o *k-means fast* com uma ligeira descida correspondente a 57,14%, o algoritmo em falta é o *random clustering* que apresenta o valor de 25%. Analisando agora os valores obtidos pelo método de avaliação *kappa* temos um valor de 0.374 que traduz numa concordância razoável para o *k-means*, já para o *k-medoids* temos um *kappa* de 0.374, ou seja, idêntico nos dois algoritmos descritos. Por fim temos o *randon clustering* que não foi alem dos 0.105 e com isto tem uma concordância mínima.

Voltamos novamente ao programador Gonçalo, mas com uso do parâmetro de pré-processamento com respetivo *generate n-grams* (carateres).

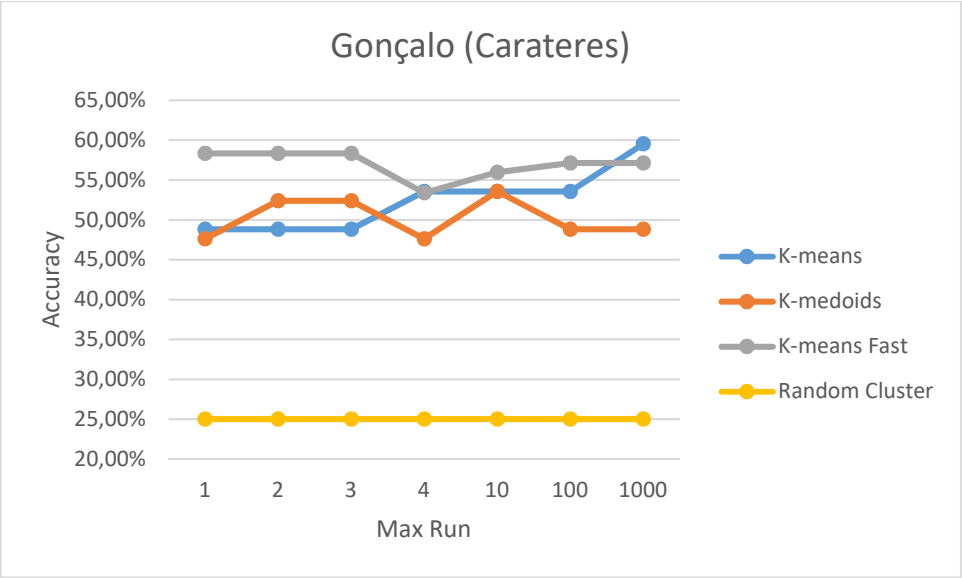


Figura 4.11: *Accuracy* dos primeiros 4 algoritmos de *clustering* usando carateres para o Ator Gonçalo

Ao observar o gráfico da figura 4.11, permito-nos verificar algumas oscilações ao longo do gráfico, chegamos a conclusão que mesmo após o *max run* de 100 verificamos oscilações nomeadamente na subida do algoritmo *k-means* que termina com um *accuracy* de 59,53% e com um *kappa* de 0,384 referente a concordância razoável, já o *k-means (fast)* estabilizou no *max run* 100 e obteve um valor de 57,14% e um *kappa* de 0,419 que significa ter o mesmo nível de concordância que o anterior.

Para finalizar temos a última programadora e chefe Mariana, que tem um papel de líder da equipa e com isto o volume de dados e diversidade de clusters é muito superior, com isto a Mariana tem dezassete *clusters* associados.

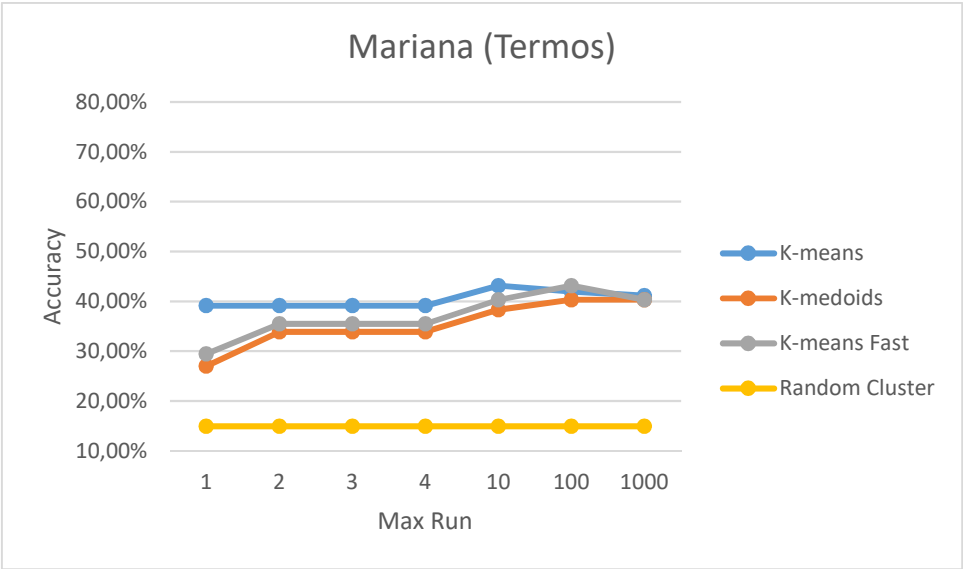


Figura 4.12: *Accuracy* dos primeiros 4 algoritmos de *clustering* usando termos para o Ator Mariana

Olhando para o gráfico 4.12, observamos uma tendência em estabilizar a *accuracy* em torno dos 40%, tendo como valor máximo obtido o *k-means* no valor 100 referente ao *max run,* verifica-se como valor máximo de 43,15%. Olhando para a medida *kappa* que é de 0.335, que significa uma concordância razoável.

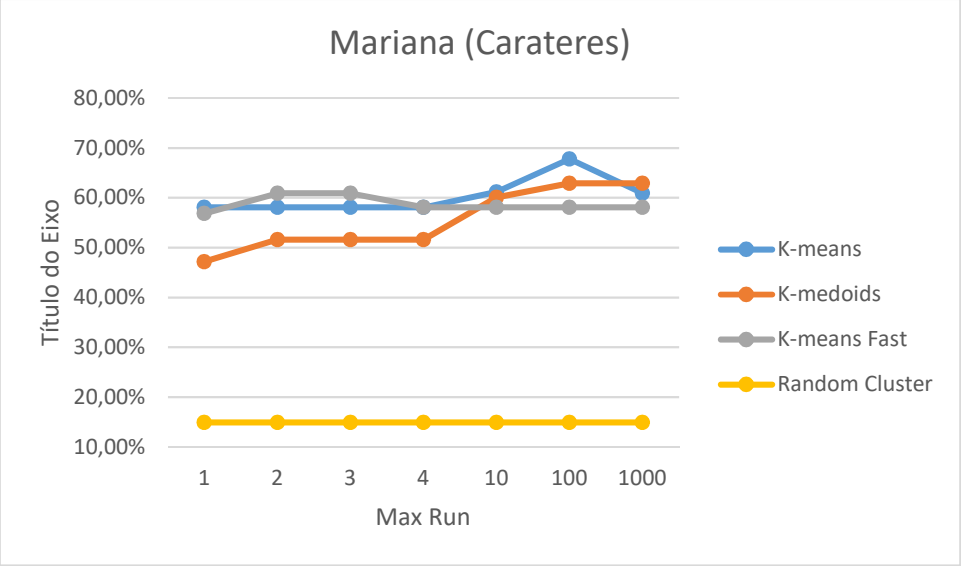


Figura 4.13: *Accuracy* dos primeiros 4 algoritmos de *clustering* usando carateres para o Ator Mariana

Terminamos assim com o parâmetro de pré-processamento referente ao *generate n-grams* (carateres). Assim verificamos que a variação gráfica tende a movimentar-se da mesma forma que a figura 4.12, mas com valores mais elevados, detetamos um pico máximo no *max run* 100 na ordem do valor 67,76%, originando um *kappa* de 0,646 o que significa ter uma concordância substancial.

Abordamos agora o algoritmo *k-means kernel,* este permite escolher o tipo de *kernel* para obtenção de *clusters*.

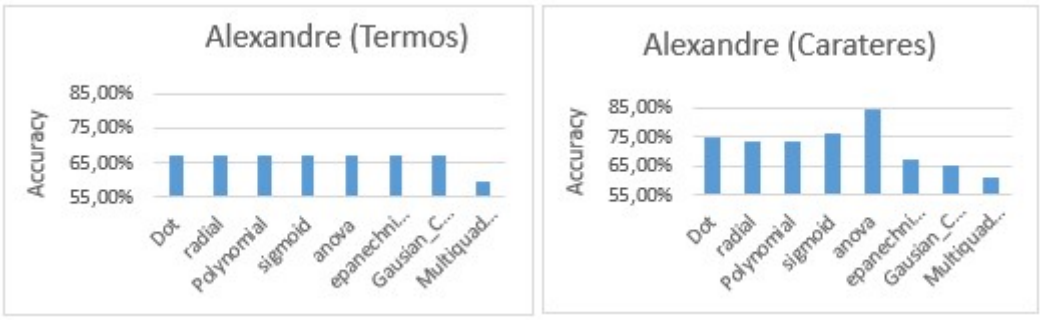


Figura 4.14: *Accuracy* do algoritmo *k-means kernel* usando termos e carateres para o Ator Alexandre

No caso do ator Alexandre os *clusters* obtidos com a utilização do parâmetro *Generate n-grams* (carateres) alcançaram melhores resultados em comparação com o parâmetro *Generate*

n-grams (termos). Temos de destacar o parâmetro *anova,* que mostra um valor de *accuracy* superior aos restantes, o parâmetro *anova* tem uma percentagem de 84,38% e o respetivo valor *kappa* é de 0.785 que corresponde a uma concordância substancial.

Para o caso do Ator Carla podemos visualizar a seguinte figura 4.15.

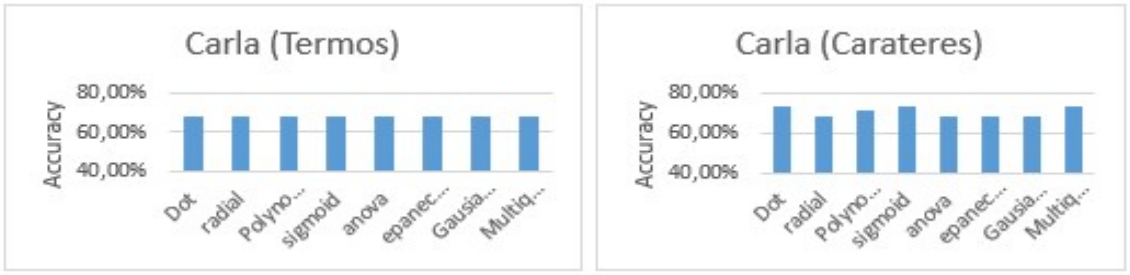


Figura 4.15: *Accuracy* do algoritmo *K-means Kernel* usando termos e carateres para o Ator Carla

No caso do ator Carla os *clusters* obtidos com a utilização do parâmetro *Generate n-grams* (carateres) permite verificar melhores resultados comparados com o parâmetro *Generate n-grams* (termos). Temos de destacar os parâmetros *dot, sigmoid* e o *multiquadric* que mostram um valor de *accuracy* superior aos restantes, estes parâmetros tem uma percentagem de 73,68% e o respetivo valor *kappa* é de 0.548 que corresponde a uma concordância moderada.

Analisando agora o Ator Catarina podemos visualizar a figura 4.16.

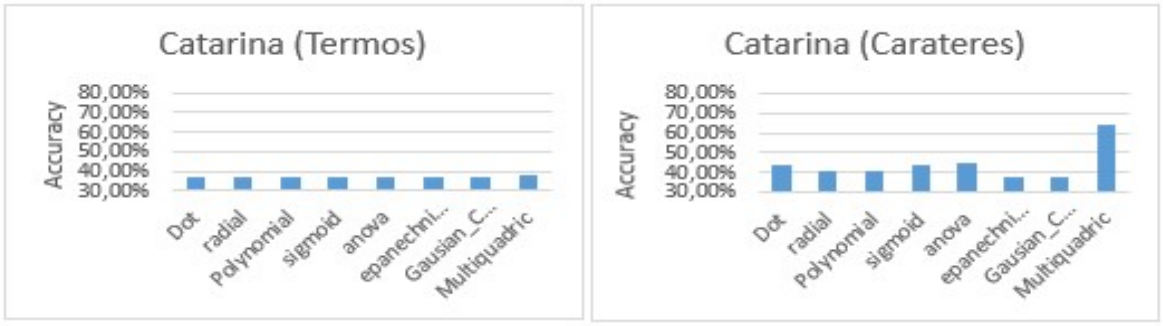


Figura 4.16: *Accuracy* do algoritmo *k-means kernel* usando termos e carateres para o Ator Catarina

A utilização destes dados demostra pouca qualidade na obtenção da *accuracy*, observando os gráficos podemos concluir que os termos tem baixa percentagem, o máximo obtido é de 38,55%, analisando o valor *kappa* temos -0.188, este valor na tabela do coeficiente *kappa* indica-nos que este se encontra no primeiro nível, ou seja para valores negativos, com isto estamos a falar no primeiro nível que significa que não existe concordância. O gráfico que pertence aos Carateres verificamos uma melhoria nos resultados em comparação ao gráfico

por termos, a exceção verifica-se no parâmetro *multiquadratic* que apresenta um valor mais alto de 63,86% que significa um valor de 0,246 pertencendo ao nível de concordância razoável.

A figura agora em análise agora é do ator Gonçalo.

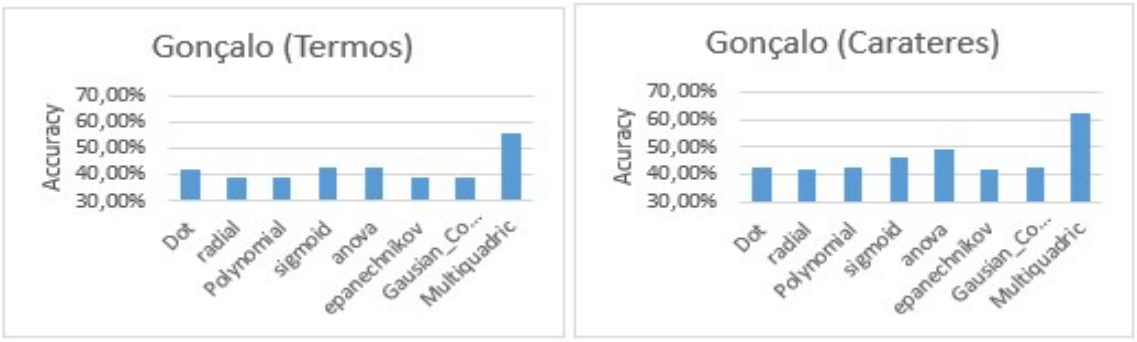


Figura 4.17: *Accuracy* do algoritmo *k-means kernel* usando termos e carateres para o Ator Gonçalo

Aqui temos um claro parâmetro que mostra um melhor resultado em relação a todos os outros, que neste caso é o *multiquadratic* com 55,95%, e um *kappa* de 0.301 correspondente a uma concordância razoável.

Por fim temos os resultados da Ator Mariana.

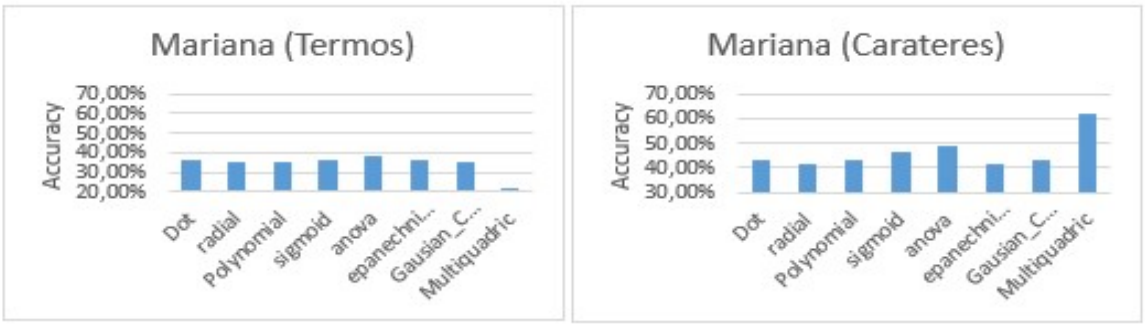


Figura 4.18: *Accuracy* do algoritmo *k-means kernel* usando termos e carateres para o Ator Mariana

Permite-nos avaliar que o parâmetro *multiquadratic* é o que mostra valores mais discrepantes, no gráfico para termos obteve-se um valor bem abaixo dos restantes parâmetros, já no gráfico para carateres o parâmetro em questão melhorou significativamente e superou os restantes parâmetros que se mantiveram estáveis. Analisando o melhor valor foi obtido no gráfico de carateres, com um valor para a *accuracy* de 61,90% e um *kappa* de 0,384, pertencendo a concordância razoável, em sentido inverso temos 21,37% com um *kappa* de 0.064, valor referente a concordância mínima, mostrando uma clara melhoria quando se utiliza o *generate n-grams (carateres)*.

Observando agora o algoritmo *aglomerative hierarchical clustering*, este algoritmo de *cluster* que se encontra explicado na secção 2.4.2.1 é mais um método de agrupamento existente na ferramenta *Rapidminer Studio* ®. Analisando os resultados, e tendo em conta que obtemos estes valores a partir da matriz de proximidade (min/*single*, max/*complete* e média/a*verage*), conjugando com os parâmetros descritos na secção 3.2.2.

Figura 4.19: *Accuracy* do algoritmo *aglomerative hierarchical clustering* usando termos e carateres para o Ator Alexandre

Observando os dois gráficos obtidos, como temos vindo a observar nos outros algoritmos estudados, a utilização do *generate n-grams* (carateres) demostra uma *performance* muito mais elevada em relação ao *generate n-grams* (termos). Analisando a matriz de proximidade de uma forma geral para o gráfico referente aos termos, os resultados foram muito idênticos nos três métodos de proximidade em relação aos parâmetros utilizados, temos o valor mais alto da *accuracy* de 60,94% e referente a um *kappa* de 0.439 que indica-nos uma concordância moderada, em relação ao gráfico dos carateres temos um *performance* maior na matriz de proximidade *average link*, no parâmetro *dice similarity* a atingir o valor de 89,06% e um valor *Kappa* de 0.845 verificando assim um concordância perfeita.

Para a programadora Carla podemos verificar.

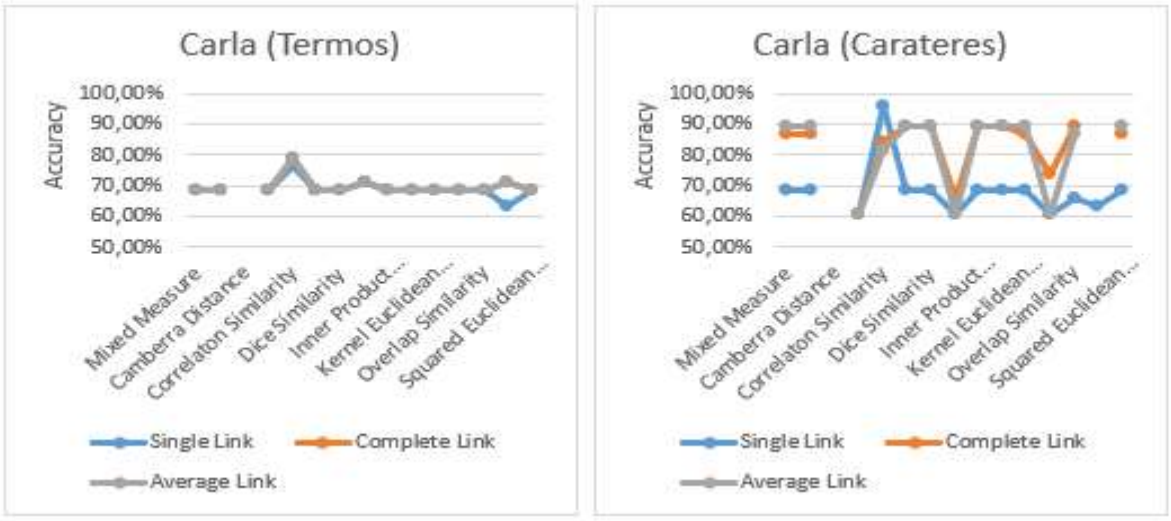


Figura 4.20: *Accuracy* do algoritmo *aglomerative hierarchical clustering* usando termos e carateres para o Ator Carla

Como tem sido normal, utilização da escolha por carateres mostra valores mais elevados, assim podemos referir que para termos temos um pico no parâmetro *correlation similarity* com a utilização da matriz de proximidade de *complete link* e *average link*, e com um *accuracy* de 78,95% e um *kappa* de 0.619 correspondente a uma concordância substancial. No que diz respeito ao gráfico referente aos carateres temos excelentes valores, destaque para a matriz de proximidade *single link* que obtém um valor de 94,74% e um *kappa* de 0,909 ou seja, com concordância perfeita.

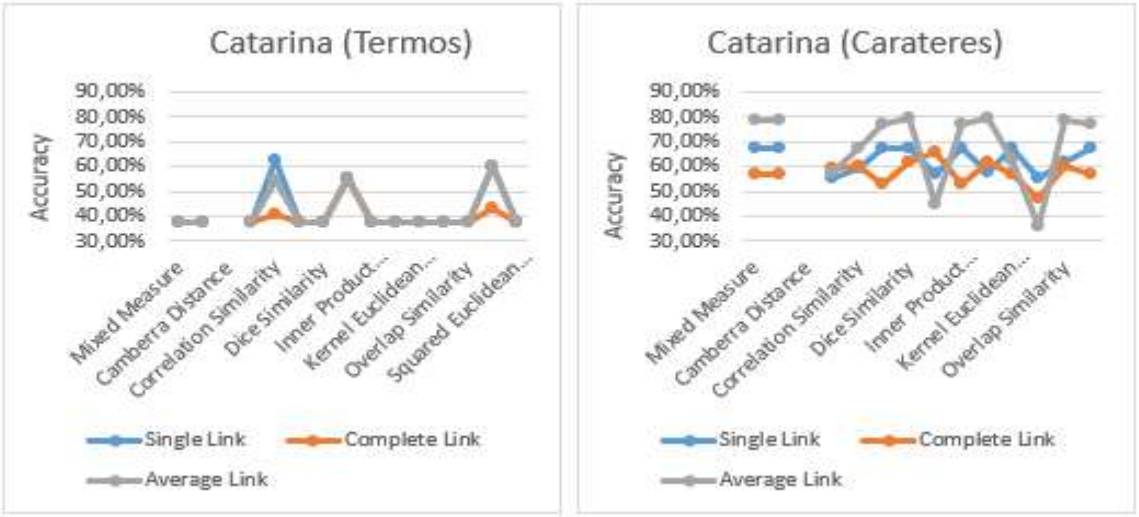


Figura 4.21: *Accuracy* do algoritmo *Aglomerative hierarchical clustering* usando termos e carateres para o Ator Catarina

Analisando agora a programadora Catarina, verificamos no gráfico referente aos carateres que tem o dobro da *performance* da *accuracy* em relação ao gráfico por termos. O valor mais alto

no primeiro gráfico é de 62,65%, correspondendo a um *kappa* de 0.185, o que significa que tem uma concordância mínima, isto acontece na matriz de concordância *single link*, para o parâmetro *correlation similarity*. Já o gráfico por carateres, este atingiu o valor de 79,52% de *accuracy* no na matriz *average link,* mais propriamente nos parâmetros *dice similarity* e no *jaccard similarity,* obtendo um valor *kappa* de 0.684, valor compreendido no nível de concordância substancial.

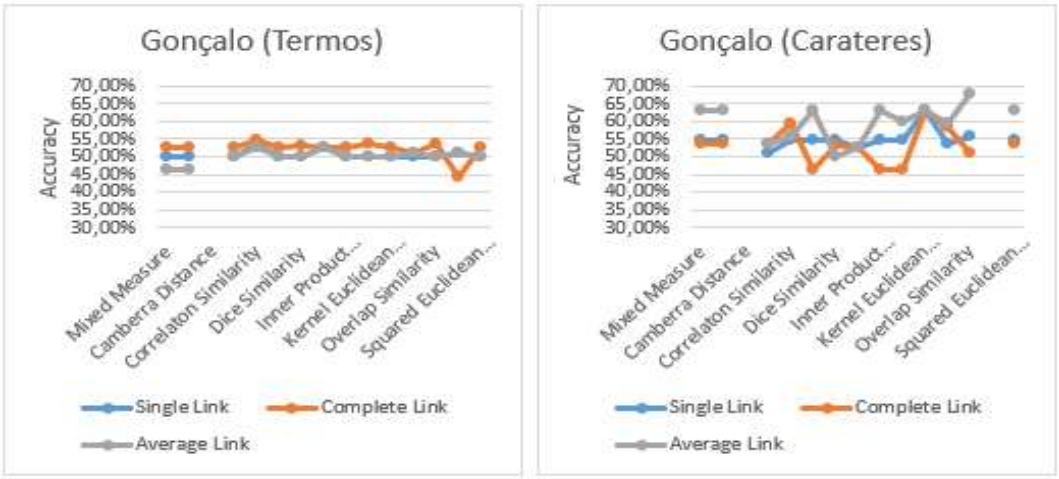


Figura 4.22: *Accuracy* do algoritmo *aglomerative hierarchical clustering* usando termos e carateres para o Ator Gonçalo

O programador Gonçalo, no gráfico de termos tem uma medição praticamente linear, apresentando o valor mais alto no parâmetro *correlation similarity* para a matriz de proximidade *complete link* de 54,76% e um *kappa* de 0.307 que indica o nível de concordância razoável. Para o segundo gráfico apresentam alguns picos, a qual se destaca com 72,62% e com 0.551 referente a concordância moderada na matriz de proximidade de *average link* e com o parâmetro *overlap similarity.*

Figura 4.23: *Accuracy* do algoritmo *Aglomerative Clustering* usando termos e carateres para o Ator Mariana

Por fim, temos a programadora e chefe Mariana, analisando o gráfico por termos nota-se um pico na *accuracy* com o valor mais alto para o *complete link* que traduz nos 54,76% e um *kappa* de 0.365, o que corresponde a uma concordância substancial no parâmetro *correlation similarity*. Analisando agora o gráfico para os carateres temos um pico máximo para o parâmetro *correlation similarity* referente a matriz de proximidade *complete link* de 66,13% originando um *kappa* de 0.627 com concordância substancial.

4.1.1.1. ANALISE DE RESULTADOS DE *CLUSTERS* DO CONJUNTO DE ATORES

Analisando os resultados de *clustering* com recurso à *confusion matrix* na situação em que os dados se encontram todos juntos, ou seja, todos os atores envolvidos, podemos verificar que o valor da *accuracy* não passa acima dos 30%.

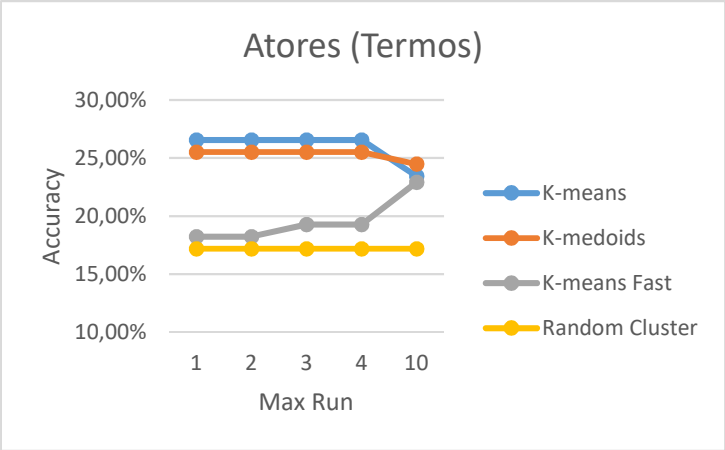


Figura 4.24: *Accuracy* dos primeiros 4 algoritmos usando termos para o conjunto de Atores

Analisando a figura 4.24 permite destacar que ao aproximar o *max run* de 10 temos uma convergência na *accuracy* em torno dos 23%, verifica-se que o algoritmo *k-medoids* tem o valor mais alto, cerca de 24,48% o que indica um valor *kappa* de 0.127, assim conseguimos concluir que temos uma concordância mínima.

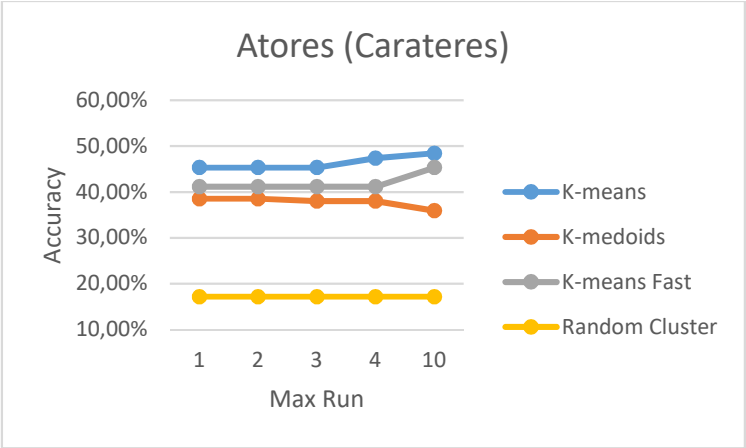


Figura 4.25: *Accuracy* dos primeiros 4 algoritmos usando carateres para o conjunto de Atores

Para o conjunto de atores, com a utilização do parâmetro *generate n-grams* (carateres) temos um melhoramento nos resultados em relação ao parâmetro anterior utilizado. Com isto, verificamos que o *k-means* obtém melhor resultado em relação aos restantes três analisados na figura 4.25. Podemos observar que o valor máximo da *accuracy* é de 48,44%, assim verifica-se um *kappa* correspondente a 0,453 o que nos indica uma concordância moderada.

Agora analisando o algoritmo implementado *k-means (kernel)* para o conjunto de atores.

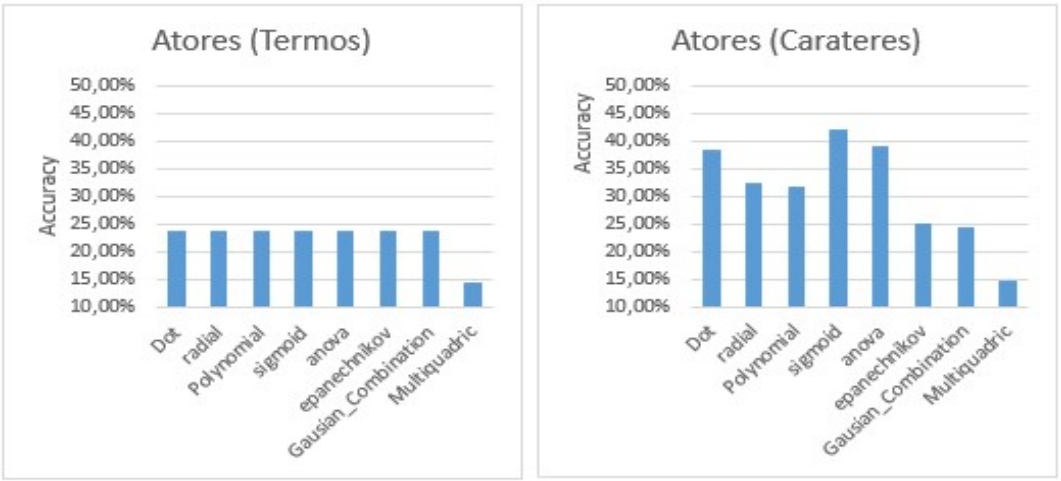


Figura 4.26: *Accuracy* do algoritmo *k-means kernel* usando termos e carateres para o conjunto de Atores

Comparando as duas figuras, podemos logo destacar um melhoramento dos resultados no parâmetro da *accuracy* para o parâmetro de pré-processamento *Generate n-grams* (carateres), visualizando o gráfico dos Atores por termos o valor mais elevado não ultrapassa os 23,96% e tem um *kappa* de 0.198 o que significa corresponder a uma concordância mínima.

Já para o gráfico dos Atores por carateres obtemos um valor máximo no parâmetro *sigmoid* em que se obtém um valor de *accuracy* de 42,19% e um *kappa* de 0.287 correspondente a uma concordância razoável.

Por fim temos a análise do algoritmo *aglomerative hierarchical clustering.*

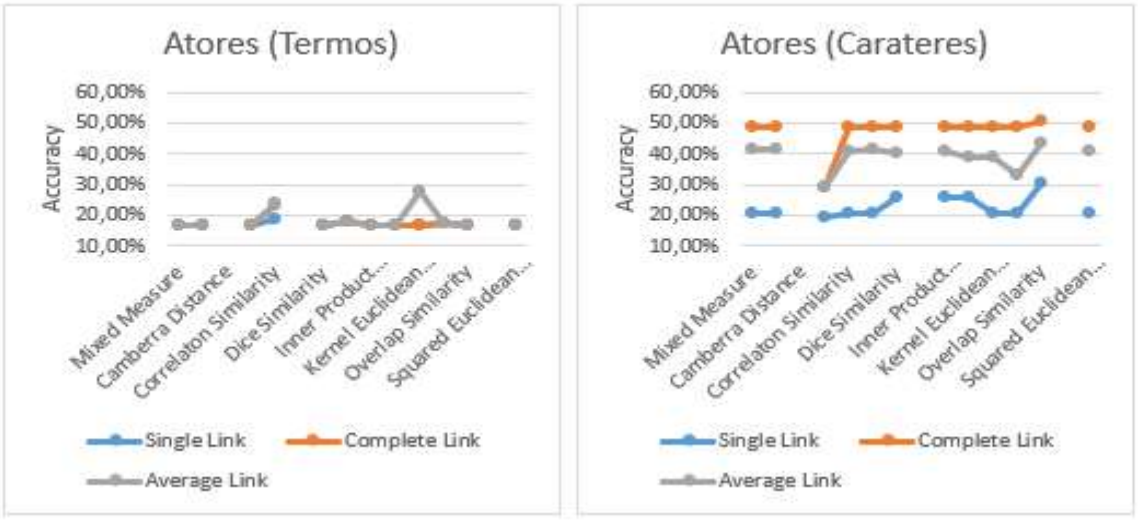


Figura 4.27: *Accuracy* do algoritmo *aglomerative hierarchical clustering* usando termos e carateres para o conjunto de Atores

Como tem sido geralmente avaliado, mais uma vez temos a comparação entre dois gráficos que refletem sempre um melhoramento na utilização do *generate n-grams* (carateres). Analisando a *accuracy* do gráfico que contem o *generate n-grams* (termos) verificamos que o valor máximo atingido é de 28,12%, com a matriz de proximidade *average link* e com o parâmetro *Kernel Euclidean Distance,* refletindo num *kappa* de 0.173, assim temos uma concordância de mínima. Visualizando o gráfico correspondente ao *generate n-grams* (carateres) é mostrado uma ordem de valores praticamente linear, apontado para o parâmetro *overlap similarity* e com o valor mais elevado, com cerca de 50,52% e é representado pela matriz de proximidade *complete link* e que reflete num *kappa* de 0,468 indicando uma concordância moderada.

4.1.2. RESULTADOS DO *CLUSTERING* NA DESCOBERTA DE CONTEXTOS DE INTERAÇÃO INTERPESSOAL

Nesta secção analisamos os resultados de *clustering* na identificação de contextos interpessoais, como descrito na secção 3.1.1 em que verificou-se que existiu partilha de caraterísticas semelhantes entre os contextos pessoais como ilustrado na figura 3.2 da secção 3.1.1, estes resultados foram obtidos de uma forma automática pela ferramenta em estudo

utilizando os algoritmos de *clustering* e depois verificou-se manualmente, não sendo possível a utilização de um método automático de comparação, como por exemplo o método da *confusion matrix.*

A tabela 4.1 ilustra os resultados do algoritmo *k-means kernel,* que foi o que obteve melhor formação de contextos interpessoais. Na primeira coluna ilustra-se o número do cluster e na segunda os contextos pessoais (*label*) associados a cada cluster pelo algoritmo. Os resultados de outros algoritmos encontram-se na secção (anexos).

Tabela 4.1: Resultados do algoritmo *K-means kernel*

Cluster	Contextos pessoais associados a cada cluster
0	<a3-a4-c1-t1-t2-t3-g5-m011-m3-m2-m8-m51>
1	<a3-c2-t4-t5-t1-g1-g2-g3-g5-m1-m012-m3-m8-m9>
2	<a3-a4-c3-t1-g1-g4-g2-m1-m3-m6-m8-m9-m10>
3	<c2-t1-t6-t2-m4-m5-m51-m3-m6-m1-m8-m5>
4	<m1-m9>
5	<a4-a3-c2-t2-t1-t3-g2-g1-g5-m5-m012-m8-m51-m3-m10-m9-m1>
6	<a1-c1-c2-t3-g2-g1-m4-m51-m81-m1-m3-m6-m8-m81-unknown>
7	<a1-a3-a2-a4-c3-c1-t1-t3-t2-t5-t6-g1-g2-g5-m2-m3-m1-m4-m012-m5-m6-m7-m8-m6-m011-m10-m81>
8	<a3-a2-a1-a4-c3-t6-t4-t5-t1-g5-g1-m1-m3-m6-m2-m10-m8>
9	<g1>
10	<t2-t1-m3-m81>
11	<a2-t1-g1-g2-m4-m7-m51-m5-m6-m10-m8-m81-m10-unknown>
12	<a2-c1>
13	<a1-a4-c2-t1-t3-g2-g1-m1-m012-m5-m3-m2-m81-m10-m8-unknown>
14	<a1-c2-t6-t1-t2-t4-g2-m6-m8-m3-m10-m1>
15	<a1-c3-c2-t2-g1-g2-m1-m5-m6-m8-m011-m82>

A tabela 4.1 ilustra os contextos de interação interpessoal identificados de forma manual a partir do registo de ações do caso de estudo, que indicam as iterações ocorridas na realidade entre contextos de ação pessoal dos atores participantes.

ID.	PERSONAL CONTEXTS	DESCRIPTION
ic1	$< a1 - x >$	data collection for mail application
ic2	$< a1 - m011 >$	cards information collection
ic3	$< a3 - m6 >$	evictions web service problem
ic4	$< c2 - a5 >$	web services and mail app. support
ic5	$< c2 - m8 >$	suppliers app. support
ic6	$< g2 - t3 >$	suppliers app. support
ic7	$< g3 - m8/m81 >$	suppliers app. support
ic8	$< m1 - a4/c3/g5/t6 >$	team meetings
ic9	$< m1 - antx >$	project management reports
ic10	$< m2 - t2 >$	automatic table updates problem
ic11	$< m3 - cgTeamx >$	integration tests
ic12	$< m4 - user >$	claims app. user support
ic13	$< m8 - t1/t3 >$	suppliers app. discussions/collaborations
ic14	$< m9 - t4 >$	message maintenance
ic15	$< m10 - antx >$	marketing campaigns app. adjustments
ic16	$< m4/m51/m6 - pubTeamx >$ $< m7/m8/m10 - pubTeamx >$	software publication

Figura 4.28: Contextos pessoais e respetiva *Description*, retirado de [7]

Comparando a tabela 4.1 e a figura 4.2 verificamos que os resultados obtidos pelo algoritmo *k-means (kernel*) não tem correspondência entre os contextos de interação interpessoal identificados manualmente e os contextos de interação interpessoal identificados pelo algoritmo. Por exemplo, o contexto interpessoal ic8 formado por ações dos contextos pessoais <m1 – a4/c3/g5/t6> não está associado a qualquer agrupamento encontrado pelo algoritmo.

4.1.3. DESCOBERTA DE INTERAÇÃO INTERPESSOAL DE CONTEXTOS INTERPESSOAIS ATRAVÉS DA *ASSOCIATION RULES*

Por último, e como alternativa a utilização de *clustering* como meio de obtenção agrupamento de semelhanças com dados interpessoais, recorreu-se a análise com recurso ao método algorítmico *association rules*. A utilização da *association rules*, explicada na secção 2.5.4 vem permitir obter relações de comunicação com semelhanças entre atores.

Como forma de avaliar a qualidade das *association rules* utilizamos a figura 4.29 denominada de *Context-based Interaction Network* que nos indica as associações verdadeira [7].

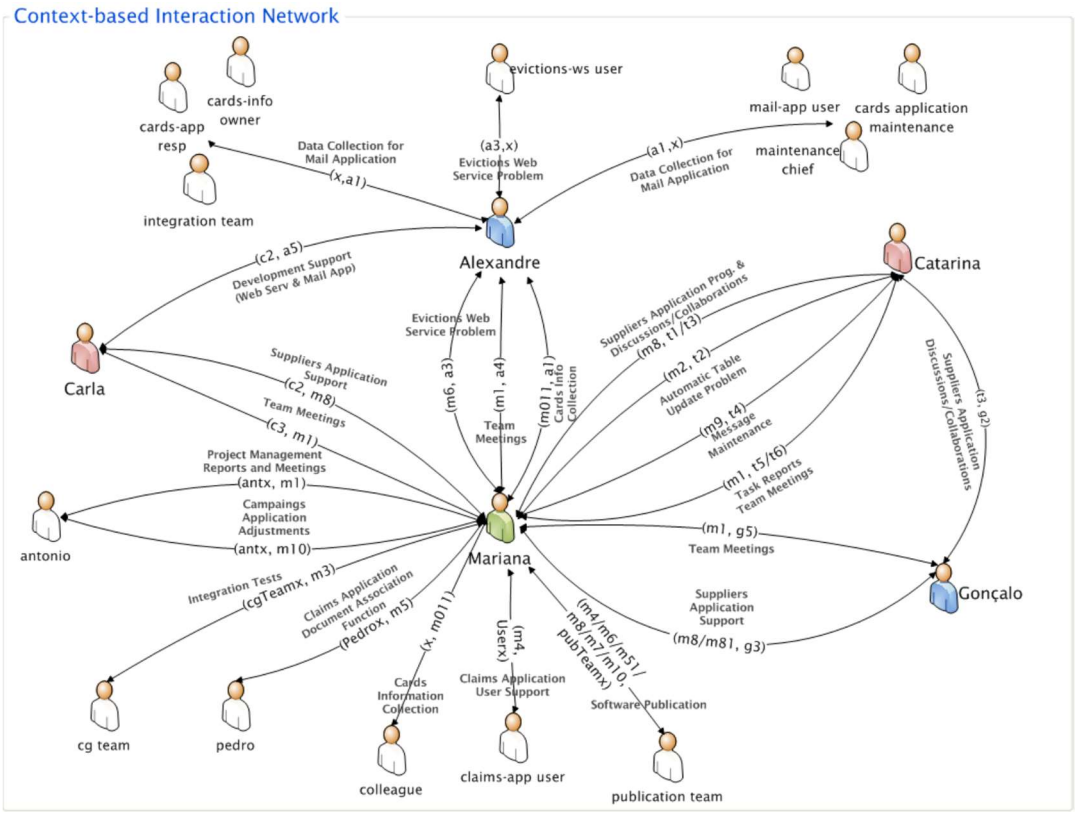


Figura 4.29: Demostração do *Context-based Interaction Network*, retirado de [7]

Para obter a *association rules* entre atores e *users*, utilizamos os operadores que permitem criar e controlar os resultados, o *FP-Growth* e o *Create Association Rules* da ferramenta *Rapidminer Studio* ®. Como forma de obter este resultado começamos com valores de suporte e confiança altos e baixou-se estes parâmetros à medida que capturávamos os resultados por ordem decrescente.

Analisando a figura 4.30 é possível visualizar os conjuntos das interações entre atores/*users* e a respetiva comunicação efetuada por ambos. Os valores mínimos de suporte são relativamente baixo porque a quantidade de informação introduzida é relativamente extensa e existe uma variedade de ações presentes.

Size	Support ↓	Item 1	Item 2	Item 3
3	0.037	mariana	cg team	integration tests, claims appl
3	0.017	alexandr	carla	mail appl
3	0.013	mariana	cg team	test, integration tests, claims appl
3	0.010	mariana	antonio	weekly status report
3	0.010	mariana	claims-app us	how-to, create users, claims appl
3	0.007	mariana	catarina	most recent version, project, suppliers appl
3	0.007	mariana	catarina	query requirements, suppliers application, databas
3	0.007	mariana	catarina	search function, suppliers appl
3	0.007	mariana	alexandr	publication, evictions web service, production env
3	0.007	mariana	alexandr	publication, evictions web service, quality env

Figura 4.30: Resultados em tabela do algoritmo *fp-growth*

O algoritmo implementado *fp-growth* permite mostrar ao nível de tabelas as interações realizadas. A imagem a seguir mostra o resultado com o *support* mais alto, com isto chega-se a conclusão que a comunicação entre a Mariana (gestora da equipa) e o *user cg team* tiveram uma frequência alta de comunicação, o que torna uma ação com muita relevância.

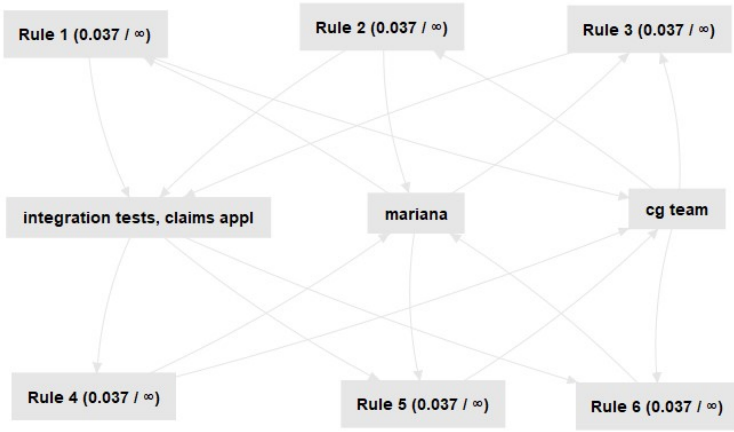


Figura 4.31: Rede da *association rules* (*Integration Tests* (cgTeamx-m3))

Na obtenção desta regra de associação podemos analisar que foi possível validar uma associação existente no contexto de interação utilizado para avaliação e que ocorre com gau de suporte de 3,7%. Representado pelo esquema a seguir:

Figura 4.32: Ligação entre cgTeams e m3

A regra a seguir obtida permite identificar os atores Carla e Alexandre que trocam várias ações de *mail application,* esta regra tem um Support de 1,7%

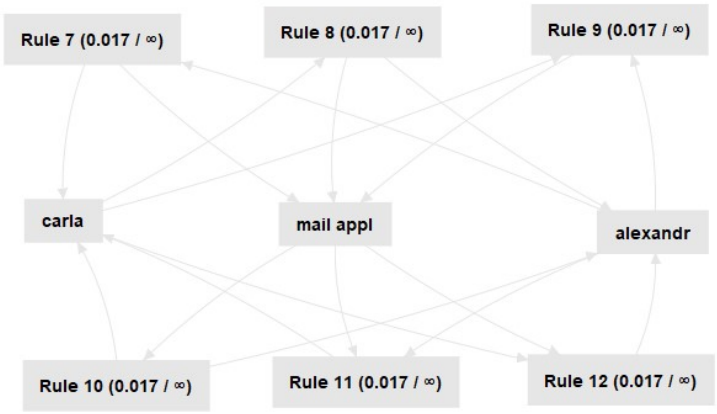


Figura 4.33: Rede da *association rules* (*Developement Support* (c2-a5))

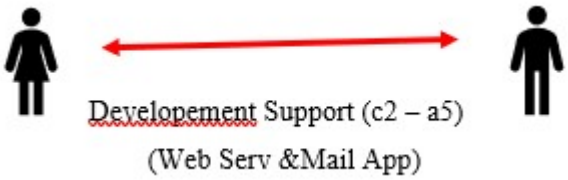


Figura 4.34: Ligação entre cgTeams e m3

Neste caso temos um contexto interpessoal entre o António e a chefe Mariana que trocam informações de "*weekly status report*" que corresponde ao conjunto de ações denominado por "*Reports and Meeting*", esta regra atingiu uma de frequência do *support* de 1%.

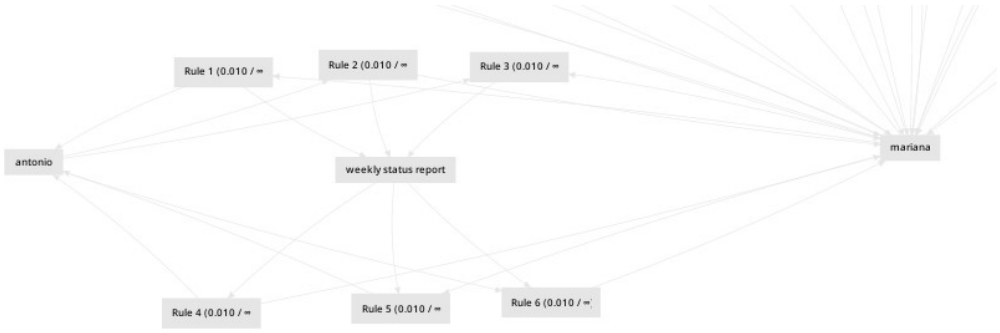


Figura 4.35: Rede da *association rules* (*Project Management* (cgTeams-m3))

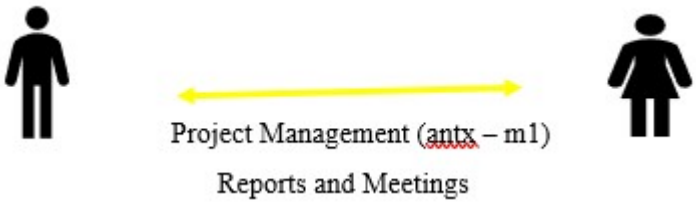


Figura 4.36: Ligação entre antx e m1

Por último, também conseguimos obter a regra entre mariana e *claims-app user* que comunica a ação de "*how-to, create users, claims application*" que corresponde ao conjunto de ações denominado por "*Reports and Meeting*", esta regra atingiu uma de frequência do *support* de 1%.

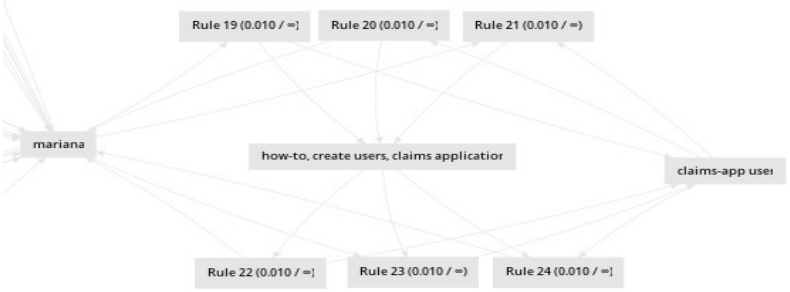


Figura 4.37: Rede da *association rules* (Project Management (cgTeams-m3))

Figura 4.38: Ligação entre m4 e userx

É possível ver todas as associações existentes nos dados baixando a frequência *de support*, mas a visualização da regra de associação torna-se complexa e extensa, ficando em forma de

"esparguete", como se comprova na figura 4.39 em que se torna difícil distinguir as associações a analisar, mas é possível visualizar.

Figura 4.39: Mapa da *association rules* da rede completa

Apesar de não conseguirmos utilizar a medida de confiança para as expressões que são comunicadas entre os atores e *users.* Existe forma de obter a confiança dos resultados, a figura 4.40 mostra as premissas e as conclusões quando obtemos os resultados da *association rules,* estes só são possíveis com a utilização de atores e *users*, não existindo a possibilidade de utilizar a mensagem transmitida entre o *sender* e o *receiver*.

Growth) | AssociationRules (Create Association Rules)

No.	Premises	Conclusion	Support	Confiden...	LaPlace	Gain	p-s	Lift	Convicti...
14	cg team	mariana	0.093	1	1	-0.093	0.021	1.288	∞
13	publication team	mariana	0.147	0.978	0.997	-0.153	0.030	1.259	10.050
12	antonio	mariana	0.083	0.758	0.976	-0.137	-0.002	0.975	0.921
11	catarina	mariana	0.107	0.593	0.938	-0.253	-0.033	0.763	0.548
10	goncalo	mariana	0.083	0.532	0.937	-0.230	-0.038	0.685	0.477
9	alexandr	mariana	0.070	0.429	0.920	-0.257	-0.057	0.552	0.391
8	goncalo	catarina	0.053	0.340	0.911	-0.260	0.025	1.891	1.243
7	catarina	goncalo	0.053	0.296	0.893	-0.307	0.025	1.891	1.198
6	mariana	publication team	0.147	0.189	0.645	-1.407	0.030	1.259	1.048

Figura 4.40: Resultados de suporte e confiança correspondente ao *Sender* e ao *Receiver*

Podemos visualizar na figura 4.40, a percentagem de confiança quando tem uma premissa e uma conclusão, neste caso quando era indicado um ator/*user* associado um outro ator/*user*. Na figura 4.41 visualizamos uma *association rules*, que foi a regra com maior confiança.

Figura 4.41: Esquema da association rules (publication team e mariana)

Verificamos que a obtenção do resultado com um suporte de 10% e uma confiança mínima de 80% permitiu indicar que o *publication team* acontece, então ocorre uma comunicação com Mariana, apresentando um grau de confiança para esta regra de cerca de 97,8%.

4.2. DISCUSSÃO DOS RESULTADOS

Após a descrição dos resultados, analisamos os resultados da secção 4.1.1 referentes aos contextos pessoais, em que se avaliou a *performance* dos *clusters,* inicialmente verificou-se duas escolhas possíveis nos parâmetros de pré-processamento, o *generate n-grams* (termos) e o *generate n-grams* (carateres), que mostram diferenças substanciais nos resultados. A tabela 4.2, indica-nos os melhores resultados obtidos pela técnica utilizada.

Tabela 4.2: Resultados da *Confusion Matrix* do ator Alexandre

Ator	Generate n-grams	Algoritmo	Parâmetro	*Accuracy*	*Kappa*
Alexandre	Carateres	*K-means Fast*	*Squared Euclidean Distance*	92,19%	0.892
Carla	Carateres	*Aglomerative C.*	*Single Link: Correlation Similarity*	94,74%	0.909
Catarina	Carateres	*Aglomerative C.*	*Average link: Dice similarity e jaccard similarity*	79,52%	0.684
Gonçalo	Carateres	*Aglomerative C.*	*Average Link: Overlap Similarity*	72,20%	0.684
Mariana	Carateres	*K-means*	*Manhattam D., Euclidean D. e Squared Euclidean D.*	67,76%	0.646
Atores	Carateres	*Aglomerative C.*	*Complete Link: Overlap Similarity*	50,52%	0.468

A tabela 4.2 permite verificar que a utilização do *generate n-grams* (carateres) de um modo geral melhorou significativamente as medidas de avaliação impostas, isto deveu-se a utilização apenas os primeiros cinco carateres dos atributos de cada ator, ou seja, ao analisar as frases, recorreu-se a utilização do máximo de cinco carateres de cada atributo, assim permitiu descrever com maior qualidade a formação dos *clusters,* comparativamente a utilização do *generate n-grams (termos)*. Para o *gerenarate n-grams (termos),* que se utilizou três termos, notou-se que perdeu alguma interpretação na identificação das expressões para a formação de *clusters*.

Nesta análise comprovou-se que ao restringir a utilização da análise de texto, por exemplo, a utilização de uma palavra ou alguns carateres conseguimos reunir melhores agrupamentos e ganhar melhor significado para as expressões do que ter um conjunto de expressões.

Analisando em pormenor os resultados dos algoritmos de *cluster*, verificamos que o algoritmo como o melhor e com excelente qualidade perante os resultados obtidos, o algoritmo *Aglomerative Hierarchical Clustering* e o *K-means Fast,* foram os que tiveram uma concordância perfeita, atingindo o nível mais elevado da escala da medida de avaliação *kappa*.

O funcionamento deste algoritmo descrito nas secção 2.4.2.1, o A*lgomerative Hierarchical Clustering* mostra a disponibilidade de opção de escolha das três matrizes de proximidade, que permite calcular com maior eficácia as distâncias entre as dados, o que torna mais preciso a obtenção dos *clusters*, para os dados introduzidos verificou-se que a atora Carla obteve uma precisão de 94,74%, que foi estabelecida pela matriz de proximidade *single link*. Isto demostra que dependendo da estrutura dos dados introduzidos, quer pela diversidade das proximidades num espaço n-dimensional das expressões este algoritmo adapta-se bem para uma boa avaliação dos dados.

Na secção 2.4.2.6 temos o *k-means (fast)*, outro algoritmo com bons resultados, este descreve que otimiza a utilização dos dados na memória, mesmo utilizando a base do algoritmo *K-means*, como os dados introduzidos são frases, este algoritmo permite "poupar" nos recursos utilizados da máquina física e analisar mais eficazmente os resultados. A seguir temos o *K-means* e o *K-medoids*, são dois métodos de extrema importância, que apresentaram resultados satisfatórios, foram menos eficazes que os dois descritos anteriormente, o *K-means (fast)* e o *Aglomerative Hierarchichal Clustering*, sendo que estes algoritmos apresentam uma estrutura de avaliação rígida tendo assim mais dificuldade em interpretar as mais variadas tarefas em análise, realçamos que o *K-means* é a base estruturante das variações *fast* e *kernel*.

Por último temos o algoritmo *Random Clustering*, um algoritmo neutro, sem seguir um estilo de agrupamento como todos os outros analisados, e como o próprio nome refere, faz agrupamento aleatório, apresentou resultados fracos, quando interpretados pela matriz de concordância obteve praticamente sempre concordância mínima.

Numa segunda análise realizada, realizou-se a identificação de contextos interpessoais através de *clustering* utilizando o recurso da *confusion matrix,* para o conjunto de todos os atores (Alexandre, Carla, Catarina, Gonçalo e Mariana), que se encontra apresentada e descrita na secção 4.1.2.

Verificou-se que os resultados ficaram a quem do esperado, podemos epilogar que a percentagem de *accuracy* não atingiu os 50%, o máximo atingido foi 48,44% que corresponde a uma concordância moderada para o algoritmo de *clustering K-means*. Este valor relativamente baixo deve-se ao elevado número de expressões textuais em análise, cerca de 517 tarefas, e um elevado número de *clusters* previsto, cerca de 23 *clusters* pretendidos a obter,

o que demostra que os algoritmos no modo geral funcionam bem, mas para uma menor quantidade de dados e de *clusters*.

Uma terceira abordagem realizada na secção 4.1.2 foi verificar a capacidade da ferramenta *Rapidminer Studio* ® foi criar contextos interpessoais automáticos, ou seja, verificar a interação de contextos entre os atores e *users*, analisando a figura 4.28 que diz respeito a *a* interação de contextos pessoais e o algoritmo de cluster *K-means Kernel*, que foi o algoritmo que se aproximou mais da respetiva tabela de comparação, verificamos que os resultados esperados ficaram aquém do desejado, assim conclui-se que o método de obtenção através da utilização de algoritmos de *clustering* existente no ferramenta *Rapidminer Studio* ® não é viável para o pretendido.

Com a análise anterior fracassada, obrigou-nos a recorrer a outra técnica disponível na ferramenta *Rapidminer Studio* ®, a técnica denominada de *Associaction Rules.* Essa verificação teve de ser feita manualmente e é demostrado na figura 3.2 da secção 3.1.1, a ferramenta utiliza as expressões originais dos dados introduzidos.

Assim é possível verificar que as regras de associação encontradas estão presentes no esquema de *Context-based Interaction Network* da figura seguinte.

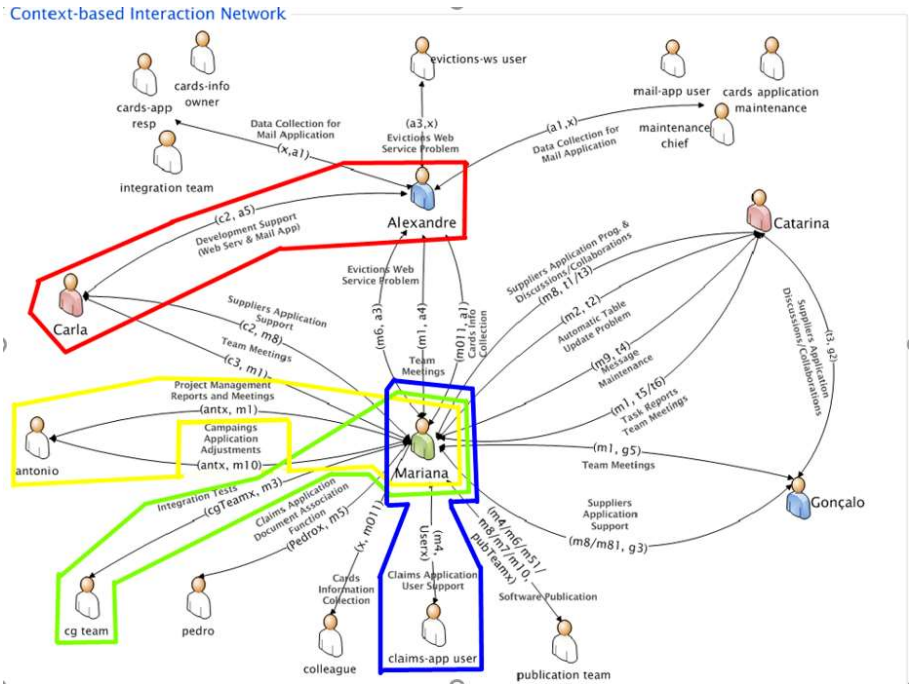


Figura 4.42: Identificação das *association rules* na ferramenta, adaptado de [7]

Vemos que cada conexão está representada como um conjunto de ações e não como apenas uma ação, ao obter a *Association Rules* temos de associar a que grupo pertence esse item

detetado. Com a utilização do algoritmo foi possível encontrar quatro associações, a Carla - Alexandre (c2 – a5) – Development (Web serv & mail app), Antonio – Mariana (antx – m3) – Project Management Report and Meeting e Mariana – Claims-app user (m4 – userx) – Claims Application user support.

A utilização desta técnica permitiu-nos obter as associações, mostradas na figura 4.29 da secção 4.1.3. A análise é feita pelo utilizador, verificando na ferramenta se coincide, só foi possível observar com maior rapidez e facilidade as tarefas com maior frequência, porque estas apresentam um valor de suporte superior. A medida que vamos diminuindo o suporte, vão-se criando mais interações entre clusters, mas em contrapartida a medida que se vai baixando o suporte, a visualização das interações torna-se de difícil visualização e penosa para descodificação do "esparguete" que se vai formando, cada vez de maiores dimensões.

O algoritmo da *Association Rules* detetou setenta e oito conjuntos de itens em que as percentagens de suporte variam entre os valores de 0.3%, e 3,7%, já a nível de confiança resultante foi de infinito, isto deve-se ao *Association Rules* ser calculada de modo iterativo e limitamos a utilização ao mínimo de três itens (*Sender, Receiver* e *Keywords*), e como não existe suporte de itens únicos, não existe a possibilidade de cálculos das confianças para conjuntos iguais ou superiores a dois item [78]. Foi possível calcular valores de confiança, mas sem a utilização das *keywords*, apenas tínhamos associado um item e obtivemos valores variáveis de confiança, sendo que alguns com um excelente nível de confiança. De destacar o *cg team* – Mariana, *publication* – Mariana e António – Mariana que apresentam valores de confiança acima dos 75%, conclui-se que a Marina que é chefe teve uma maior interatividade de tarefas com estes *users*, ao mostrar elevada confiança em troca de informação de atividades

5. CONCLUSÕES E TRABALHO FUTURO

5.1. CONCLUSÕES

Com este trabalho, pretendeu-se explorar técnicas de análise de dados para encontrar padrões de trabalho a partir de registos de ações para fins de análise organizacional. Mais especificamente, testou-se e avaliou-se algoritmos de *Clustering e Association Rules* disponíveis na ferramenta *Rapidminer Studio ®.*

Antes da utilização da ferramenta, os registos de ações foram formatados e inseridos na respetiva ferramenta que serviu para utilização da descoberta de algoritmos e comparação dos mesmos. Ao explorar os operadores, os algoritmos e os parâmetros da ferramenta *Rapidminer Studio* ® foi possível realizar uma comparação entre eles, houve a necessidade de uma avaliação com operadores disponíveis no mesmo. Depois de tudo construído graficamente, foi possível obter resultados e chegar a uma conclusão mais pormenorizada. Durante a elaboração desta dissertação foram realizadas várias experiências com os dados e com a utilização da ferramenta *Rapidminer Studio ®.*

Primeiro, obtivemos *clusters* em contexto pessoal, em que verificamos e avaliamos a qualidade dos *clusters* gerados para cada programador. Com os resultados obtidos através da utilização das ferramentas propostas, conclui-se que no geral apesar do algoritmo *Aglomerative Hierarchical Clustering* e o *K-means (Fast)* ter conseguido melhores resultados, o *K-means* e o *K-medoids* acabaram por apresentar resultados não muito distante dos melhores.

Um outra análise que nos salta a vista é o número de *clusters* obtidos, por norma, e como é natural no quotidiano, quantas mais opções nos temos de escolha na nossa vida, maior é a facilidade de errar, como no ser humano a escolha e decisão torna-se mais dispersa, para a ferramenta usada acontece o mesmo, ou seja, para o *ferramenta* no estudo da previsão, o *machine learning* está inteiramente relacionado com o conhecimento, a aprendizagem acaba por estar relacionado com a estatística, o que quer dizer, quanto mais opções nos questionam, mais confusão criará a maquina e mais falhas ocorrerão, com isto verificamos nos dados que este relacionamento é inversamente proporcional o aumento de *clusters* com o aumento da *accuracy* e o *kappa*.

Numa segunda tarefa, analisamos a obtenção de *clusters* em contexto interpessoal, com os programadores integrados numa única lista de tarefas, gerando vinte e três *clusters* referentes a todos os programadores, o que mostrou fragilidades na qualidade da obtenção dos *clusters*, no modo geral a percentagem da *accuracy* e o valor *kappa* revelaram níveis muito baixos.

Depois, numa terceira tarefa, pretendeu-se verificar as redes de contextos de interação, e aqui verificamos se existia uma similaridade entre *clusters*, tentou-se obter de uma forma automática essas combinações com recurso a técnica de *clustering* disponíveis na ferramenta *Rapidminer Studio ®,* mas sem êxito.

De forma a contornar os resultados menos bons da tarefa anterior, recorreu-se a técnica de *association rules* para a descoberta de redes de contextos de interação, o que demostrou ser uma boa técnica, mas com algumas limitações quando se pretende obter as associações totais, esta tornou-se penoso e extenso. Mas para a descoberta de padrões com elevada frequência é uma ferramenta com muita qualidade, com esta permitiu encontrar comunicações entre programadores e programadores - *users* com maior ou menor qualidade de suporte e confiança.

É possível afirmar que a utilização dos dados e dos algoritmos não vai depender só de si mesmo, para um bom funcionamento e uma boa análise dos resultados é fundamental seguir algum modelo, como é o exemplo da técnica de CRISP-DM, SEMMA e KDD, ou seja, o analista ou o investigador deve ter o conhecimento de todas as fases, das caraterísticas dos dados, do problema e dos resultados que pretende obter.

Será fundamental um conhecimento do ambiente e do contexto, das tarefas a implementar, das ferramentas, algoritmos e parâmetros empregues no projeto para se conseguir bons resultados.

Para trabalho futuro nesta área de estudo seria interessante a utilização de outras ferramentas existentes no mercado, tais como o WEKA ®, KNIME ® e o ORANGE ®, para comparação com a ferramenta RAPIDMINER STUDIO ®. A observação de novos algoritmos que possam existir nas ferramentas não avaliadas. Avaliação e comparação da *performance* de algoritmos de *cluster* que tem o mesmo estilo de funcionamento. Observação do desempenho de novos algoritmos de *clusters* que possam existir em ferramentas desconhecidas, que existam com a utilização dos dados aplicados nesta dissertação.

6. REFERENCIAS BIBLIOGRÁFICAS

[1]H. Nemati, "Organizational Data Mining (ODM): An Introduction," no. November, 2014.

[2]"Big Data: O que é? Conheça seu conceito e definição | Navita." https://navita.com.br/blog/big-data-saiba-mais-sobre-o-conceito-e-definicao/ (accessed Oct. 20, 2020).

[3]"Os dados fazem girar o mundo - DV." https://www.dinheirovivo.pt/opiniao/os-dados-fazem-girar-o-mundo-12896149.html (accessed Oct. 21, 2020).

[4]"Visualforma - Recrutamento." https://www.visualforma.pt/pt/66/recrutamento.aspx (accessed Oct. 21, 2020).

[5]J. P. Jaiwei Han, Micheline Kamber, *Data Mining Concepts and Techniques*, Third Edit. Illinois: TheMorgan Kaufmann Series in DataManagement Systems, 2012.

[6]E. A. M. Morais and A. P. L. Ambrósio, "Mineração de Textos - technical report," p. 29, 2007.

[7]M. S. de Zacarias, "Conceptual Framework Based on Agents and Contexts for the Alignment between Individuals and Organizations PhD Thesis," Instituto Superior Técnico.

[8]C. Regina and M. Rosa, "KNOWLEDGE DISCOVERY IN DATABASE E DATA MINING : UMA," no. Dm, 2018.

[9]M. A. H. Ian H. Witten, Eibe Frank, *DATA MINING Practical Machine Learning Tools and Techniques*, Third Edit. Morgan Kaufmann, 2011.

[10] C. Pete *et al.*, "Crisp-Dm 1.0," *Cris. Consort.*, p. 76, 2000.

[11] M. Boussaa, I. Atouf, M. Atibi, and A. Bennis, "ECG signals classification using MFCC coefficients and ANN classifier," *Proc. 2016 Int. Conf. Electr. Inf. Technol. ICEIT 2016*, vol. 5, no. 4, pp. 480–484, 2016, doi: 10.1109/EITech.2016.7519646.

[12] "CRISP-DM, SEMMA e KDD: conheça as melhores técnicas para exploração de dados | by Paulo Vasconcellos | Paulo Vasconcellos—Cientista de Dados Brasileiro." https://paulovasconcellos.com.br/crisp-dm-semma-e-kdd-conheça-as-melhores-técnicas-para-exploração-de-dados-560d294547d2 (accessed Mar. 03, 2021).

[13] "Crisp DM methodology - Smart Vision Europe." https://www.sv-europe.com/crisp-dm-methodology/#two (accessed Mar. 07, 2021).

[14] B. Kamsu-foguem, F. Rigal, and F. Mauget, "Expert Systems with Applications Mining association rules for the quality improvement of the production process," vol. 40, 2013.

[15] R. F. Cássio Alan Garcia, “SISTEMA DE RECOMENDAÇÃO DE PRODUTOS UTILIZANDO MINERAÇÃO DE DADOS,” *TECNO-LÓGICA, St. Cruz do Sul, v. 17, n. 1, p. 78-90, Jan/jun. 2013.*, pp. 78–90, 2013.

[16] A. Alzghoul, M. Löfstrand, and B. Backe, “Computers & Industrial Engineering Data stream forecasting for system fault prediction,” *Comput. Ind. Eng.*, vol. 62, no. 4, pp. 972–978, 2012, doi: 10.1016/j.cie.2011.12.023.

[17] S. Viaene, M. Ayuso, D. Van Gheel, and G. Dedene, “Strategies for detecting fraudulent claims in the automobile insurance industry,” vol. 176, pp. 565–583, 2007, doi: 10.1016/j.ejor.2005.08.005.

[18] J. C. da S. Cássio Oliveira Camilo, “Mineração de Dados : Conceitos , Tarefas , Métodos e Ferramentas,” 2009.

[19] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *AI Mag.*, vol. 17, no. 3, pp. 37–53, 1996.

[20] G. Miner, J. Elder, R. A. Nisbet, J. Thompson, and R. Foley, “and Statistical Analysis Text Data Applications.”

[21] R. Sagayam, S. Srinivasan, and S. Roshni, “A Survey of Text Mining: Retrieval, Extraction and Indexing Techniques,” *Int. J. Comput. Eng. Res.*, vol. 2, no. 5, pp. 2250–3005, 2012, [Online]. Available: http://pakacademicsearch.com/pdf-files/com/319/1443-1446 Volume 2, Issue 5, September, 2012.pdf.

[22] S. VijayGaikwad, A. Chaugule, and P. Patil, “Text Mining Methods and Techniques,” *Int. J. Comput. Appl.*, vol. 85, no. 17, pp. 42–45, 2014, doi: 10.5120/14937-3507.

[23] R. Talib, M. Kashif, S. Ayesha, and F. Fatima, “Text Mining: Techniques, Applications and Issues,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 11, pp. 414–418, 2016, doi: 10.14569/ijacsa.2016.071153.

[24] A. Kaushik and S. Naithani, “A Comprehensive Study of Text Mining Approach,” *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 16, no. 2, pp. 69–76, 2016.

[25] A. Catarina and B. Forte, “Análise de comentários de clientes com o auxílio a técnicas de Text Mining para determinar o nível de (in) satisfação,” 2015.

[26] “O que é processamento de linguagem natural? | SAS.” https://www.sas.com/pt_br/insights/analytics/processamento-de-linguagem-natural.html (accessed Jun. 09, 2020).

[27] V. Kobayashi, S. T. Mol, H. A. Berkers, and D. N. Den Hartog, *Text Mining in Organizational Research Text Mining in Organizational*, no. August. 2017.

[28] C. M. dos Santos, “Classificação de Documentos com Processamento de Linguagem

Natural," p. 217, 2015, [Online]. Available: http://files.isec.pt/DOCUMENTOS/SERVICOS/BIBLIO/Teses/Tese_Mest_Cedric-Michael-Santos.pdf.

[29] V. Kalra and R. Aggarwal, "Importance of Text Data Preprocessing & Implementation in RapidMiner," *Proc. First Int. Conf. Inf. Technol. Knowl. Manag.*, vol. 14, pp. 71–75, 2018, doi: 10.15439/2017km46.

[30] D. W. FREEMAN and W. A. SISTRUNK, "Effects of Post-Harvest Storage on the Quality of Canned Snap Beans," *J. Food Sci.*, vol. 43, no. 1, pp. 211–214, 1978, doi: 10.1111/j.1365-2621.1978.tb09773.x.

[31] J. L. Solka, "Text data mining: Theory and methods," *Stat. Surv.*, vol. 2, pp. 94–112, 2008, doi: 10.1214/07-SS016.

[32] "Elcelina Rosa Correia Carvalho Silva Elcelina Rosa Correia TÉCNICAS DE DATA E TEXT MINING PARA," 2010.

[33] "Clustering – Wikipédia, a enciclopédia livre." https://pt.wikipedia.org/wiki/Clustering (accessed Mar. 08, 2021).

[34] C. S. M. de Andrade, "Text Mining na Análise de Sentimentos em Contextos de Big Data," p. 99, 2015.

[35] M. Daszykowski and B. Walczak, "Density-Based Clustering Methods," *Compr. Chemom.*, vol. 2, pp. 635–654, 2009, doi: 10.1016/B978-044452701-1.00067-3.

[36] R. L. B. e Silva, "Estudo do desempenho do algoritmo Agrupamento em Duas Etapas através de comparações realizadas sob a metodologia de planejamento de experimentos," p. 94, 2007.

[37] M. A. T. A. Castro, "Agrupamento - 'Clustering,'" p. 56, 2011.

[38] G. N. Product and R. Two, "Chapter 1 : Introduction Chapter 1 : Introduction," *Fluid Mech.*, pp. 1–16, 1966.

[39] "K-Means Clustering Algorithm - Javatpoint." https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning (accessed Mar. 29, 2021).

[40] T. Submetida, A. O. Corpo, D. Da, C. Dos, and E. M. E. Civil, "Novas Metedologias para Clusterização de Dados."

[41] F. B. M. Soares, "Utilizando o processo automático de descoberta de conhecimento para caraterização do perfil das submissões dos pesquisadores," 2015.

[42] C. De Dados, "4. Clusterização de Dados," pp. 59–96, 1948.

[43] D. Henriques and F. Nunes, "Um breve estudo sobre o algoritmo."

[44] G. Tzortzis, "Global Kernel k-Means," no. July 2008, 2014, doi:

10.1109/IJCNN.2008.4634069.

[45] M. Filippone, F. Camastra, and F. Masulli, “A Survey of Kernel Clustering Methods.”

[46] “Support Vector Machine - RapidMiner Documentation.” https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/support_vector_machines/support_vector_machine.html (accessed Apr. 19, 2021).

[47] R. Salman, V. Kecman, and R. Strack, “FAST K-MEANS ALGORITHM CLUSTERING,” no. June 2014, 2011, doi: 10.5121/ijcnc.2011.3402.

[48] “What is random clutering? — RapidMiner Community.” https://community.rapidminer.com/discussion/19447/what-is-random-clutering (accessed Apr. 18, 2021).

[49] “Distância euclidiana – Wikipédia, a enciclopédia livre.” https://pt.wikipedia.org/wiki/Distância_euclidiana (accessed Jan. 21, 2021).

[50] “Canberra distance - Wikipedia.” https://en.wikipedia.org/wiki/Canberra_distance (accessed Jan. 23, 2021).

[51] “Distância de Canberra.” http://www.code10.info/index.php?option=com_content&view=article&id=49:article_canberra-distance&catid=38:cat_coding_algorithms_data-similarity&Itemid=57 (accessed Jan. 23, 2021).

[52] “Calculate Similarity — the most relevant Metrics in a Nutshell | by Marvin Lüthe | Towards Data Science.” https://towardsdatascience.com/calculate-similarity-the-most-relevant-metrics-in-a-nutshell-9a43564f533e (accessed Jan. 24, 2021).

[53] “9.5.5. The Overlap Similarity algorithm - 9.5. Similarity algorithms.” https://neo4j.com/docs/graph-algorithms/current/labs-algorithms/overlap/ (accessed Jan. 27, 2021).

[54] “Mahalanobis Distance: Simple Definition, Examples - Statistics How To.” https://www.statisticshowto.com/mahalanobis-distance/ (accessed May 15, 2021).

[55] “Squared Euclidean Distance - an overview | ScienceDirect Topics.” https://www.sciencedirect.com/topics/computer-science/squared-euclidean-distance (accessed May 25, 2021).

[56] “Performance (Binominal Classification) - RapidMiner Documentation.” https://docs.rapidminer.com/latest/studio/operators/validation/performance/predictive/performance_binominal_classification.html (accessed Mar. 23, 2021).

[57] P. D. G. de B. V. Junior, “Coeficiente kappa,” p. 10, 1996.

[58] “Performance (Classification) - RapidMiner Documentation.”

https://docs.rapidminer.com/latest/studio/operators/validation/performance/predictive/performance_classification.html (accessed Mar. 23, 2021).

[59] J. Barth, “Minera ¸ c ˜ ao de regras de associa ¸ c ˜ ao em servidores Web com RapidMiner ∗ oes na Web,” pp. 1–14.

[60] J. W. dos S. Lima, “Aplicação de técnicas de regras de associação em sistemas de potência,” p. 70, 2014.

[61] R. Agrawal, T. Imieliński, and A. Swami, “Mining Association Rules Between Sets of Items in Large Databases,” *ACM SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, 1993, doi: 10.1145/170036.170072.

[62] “Minerações de dados frequentes com Apriori e FP Growth. | by Abner Suniga | Medium.” https://medium.com/@abnersuniga7/encontre-padrões-nos-seus-dados-com-apriori-e-fp-growth-4a581ec1b22 (accessed Mar. 24, 2021).

[63] “Minerando Dados › Café com Código #07: RapidMiner: Data Science sem escrever uma linha de código.” https://minerandodados.com.br/cafe-com-codigo-07-rapidminer-data-science-sem-escrever-uma-linha-de-codigo/ (accessed Sep. 11, 2020).

[64] “RapidMiner – Wikipédia, a enciclopédia livre.” https://pt.wikipedia.org/wiki/RapidMiner (accessed Sep. 16, 2020).

[65] T. Gomes, “Ferramentas Open Source de Data Mining,” p. 147, 2014, [Online]. Available: http://files.isec.pt/DOCUMENTOS/SERVICOS/BIBLIO/Teses/Tese_Mest_Tania-Gomes.pdf.

[66] “Programa de licença educacional RapidMiner | RapidMiner.” https://rapidminer.com/educational-program/ (accessed Nov. 23, 2020).

[67] “The core of RapidMiner is open source | RapidMiner.” https://rapidminer.com/blog/the-core-of-rapidminer-is-open-source/ (accessed Jan. 13, 2021).

[68] “Software de análise preditiva | RapidMiner Studio.” https://rapidminer.com/products/studio/ (accessed Nov. 23, 2020).

[69] “Mineração de Dados com o RapidMiner – IA Expert.” https://iaexpert.academy/2016/11/01/ferramentas-para-ia-mineracao-de-dados-com-o-rapidminer/ (accessed Nov. 24, 2020).

[70] “RapidMiner AI Hub, anteriormente RapidMiner Server | RapidMiner.” https://rapidminer.com/products/ai-hub/ (accessed Nov. 24, 2020).

[71] “About RapidMiner Go - RapidMiner Documentation.”

https://docs.rapidminer.com/latest/go/overview/ (accessed Nov. 26, 2020).

[72] “Clustering,” [Online]. Available: https://web.fe.up.pt/~ec/files_1112/week_05_Clustering.pdf.

[73] “Nominal para Texto - Documentação RapidMiner.” https://docs.rapidminer.com/latest/studio/operators/blending/attributes/types/nominal_to_text.html (accessed Apr. 03, 2021).

[74] “Set Role - RapidMiner Documentation.” https://docs.rapidminer.com/latest/studio/operators/blending/attributes/names_and_roles/set_role.html (accessed Jan. 19, 2021).

[75] “k-Means - Documentação do RapidMiner.” https://docs.rapidminer.com/latest/studio/operators/modeling/segmentation/k_means.html (accessed Jan. 21, 2021).

[76] “FP-Growth - RapidMiner Documentation.” https://docs.rapidminer.com/latest/studio/operators/modeling/associations/fp_growth.html (accessed Mar. 25, 2021).

[77] “Create Association Rules - RapidMiner Documentation.” https://docs.rapidminer.com/latest/studio/operators/modeling/associations/create_association_rules.html (accessed May 27, 2021).

[78] “FP growth produced infinite value for association rules — RapidMiner Community.” https://community.rapidminer.com/discussion/50468/fp-growth-produced-infinite-value-for-association-rules (accessed Mar. 25, 2021).

7. ANEXOS

Cluster	Contextos pessoais associados a cada cluster. K-means (max run: 1000; Septs:1000)
0	<a1-a2-a3-a4-c3-c2-c1-t1-t6-t2-t3-t4-t5-g2-g1-g3-g5-m1-m2-m3-m011-m4-m5-m012-m6-m7-m8-m51-m81-m10-m9-m82-unknown>
1	<m1 >
2	<m8>
3	<m1>
4	<g1>
5	<t2-t1>
6	<c1>
7	<m81>
8	<m9>
9	<c1>
10	< m011>
11	<m012>
12	<a2>
13	<m3>
14	<t3>
15	<m1>

Cluster	Contextos pessoais associados a cada cluster. K-medoids (max run: 1000; Septs:1000)
0	<m81>
1	<0>
2	<t1>
3	<m1>
4	<g3>
5	<a2>
6	<c1>
7	<m51>
8	<m3-m1>
9	<t6>
10	<g1>
11	<m81>
12	<c2>
13	<g2>
14	<m3>
15	<m1>

Cluster	Contextos pessoais associados a cada cluster. Random Clustering (max run: 1000; Septs:1000)
0	<a3-a2-a4-c1-t1-t2-t3-t6-g1-g5-m011-m3-m1-m5-m2-m8-m51-m81-m9>
1	<a2-a3-c1-c2-t1-t5-g1-g2-g3-g5-m1-m012-m3-m8-m6-m81-m9>
2	<a3-a4-c1-c3-t1-g2-g4-g2-g1-m1-m3-m6-m51-m8-m9-m81-m10>
3	<a2-c2-t2-t1-t6-g1-m4-m5-m3-m51-m81-m6-m1-m8>
4	<a2-a3-a4-c1-t1-t3-g1-g2-m5-m1-m81-m8-m10-m9>
5	< a2-a4-a3-c1-c2-t2-t1-t3-g2-g1-g5-m3-m012-m5-m1-m8-m51-m10-m9>
6	<a2-a1-a4-c1-t1-t3-g2-g1-g5-m4-m51-m81-m1-m3-m6-m8-unknown>
7	<a3-a4-c1-c3-t1-t2-t5-t6-g1-g2-m2-m4-m3-m5-m6-m1-m7-m8-m81-m3-m011-m10>
8	<a3-a2-a1-a4-c3-c1-t6-t1-t4-t5-g2-g5-g1-m1-m3-m81-m6-m2-m10-m8>
9	<a2-a1-c2-c1-t1-t3-t6-g1-g2-g5-m1-m4-m3-m5-m2-m10-m8-m9>

10	<a2-a3-c1-t1-t4-g1-g2-g5-m3-m4-m5-m012-m1-m81-m8-m6-m011-m10>
11	<a2-a3-a4-c3-c1-t1-g1-g2-g5-m3-m2-m5-m011-m8>
12	<a2-a3-a4-c3-c1-t1-g1-g2-g5-m3-m2-m5-m011-m8>
13	<a1-a4-c2-t1-g2-g1-m1-m012-m2-m81-m8-m10-m5-m9-unknown>
14	<a1-a2-c2-t6-t1-t2-t4-g2-m6-m3-m81-m8-m10-m1>
15	<a1-a2-a4-c3-c1-c2-t2-g1-g2-m1-m5-m6-m8-m011-m10-m81-m82>