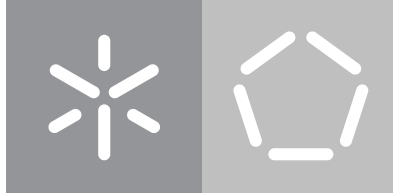


Universidade do Minho

Escola de Engenharia

Tiago Dias de Sousa

Aplicação de técnicas de IA na deteção de irregularidades em contratos públicos



Universidade do Minho

Escola de Engenharia

Tiago Dias de Sousa

Aplicação de técnicas de IA na deteção de irregularidades em contratos públicos

Dissertação de Mestrado

Mestrado Integrado em Engenharia Informática

Trabalho efetuado sob a orientação do(a)

Paulo Novais

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Creative Commons Atribuição-NãoComercial-Compartilhalgual 4.0 Internacional
CC BY-NC-SA 4.0

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.pt>

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Braga, 1 de Junho de 2022

(Localização)

(Data)

Tiago Dias de Sousa

(Tiago Dias de Sousa)

Agradecimentos

Agradeço inicialmente aos meus orientadores, Professor Paulo Novais e Marco Gomes, por promoverem a realização deste projeto e por todas sugestões, críticas construtivas e auxílio prestado durante a escrita desta dissertação.

Por fim, agradeço aos meus amigos e família, especialmente aos meus pais e ao meu irmão, por todas as palavras de apoio para que este projeto pudesse ser levado a cabo com sucesso.

Resumo

Com o passar dos anos, acompanhado pela evolução da tecnologia, existe um aumento acentuado no número de dados acumulados no setor de contratação pública em Portugal. Este aumento abre caminho para a possibilidade de tirar proveito desses dados, com recurso à utilização de técnicas emergentes de inteligência artificial, de modo a melhorar o funcionamento do processo de feitura de contratos públicos em Portugal.

No panorama da contratação pública, um dos grandes problemas que afetam a qualidade dos contratos celebrados é a existência de irregularidades não detetadas aquando da celebração dos contratos, essas irregularidades geram assim contratos que não cumprem as regras definidas para os contratos e que podem comprometer a qualidade dos bens e serviços providenciados pelo Estado.

Portanto, esta dissertação visa recolher um *dataset* de contratos públicos a partir do portal disponibilizado pelo governo português, processar e analisar os dados recolhidos, codificar as regras do Código dos Contratos Públicos num sistema de regras, investigar e utilizar técnicas de Inteligência Artificial de modo a desenvolver um *pipeline* com a finalidade de encontrar padrões de suspeição de conluio e detetar irregularidades, e, por fim, conceber modelos de *Machine Learning* para prever os valores futuros das despesas de cada entidade ou região.

Para sustentar o trabalho desenvolvido foram analisadas e relatadas na dissertação algumas implementações existentes de técnicas de inteligência artificial em contratação pública, juntamente com algumas abordagens de deteção de fraudes, assim como foram analisados diversos paradigmas e algoritmos de ML.

Por fim é demonstrado de que forma os modelos de ML foram concebidos e otimizados, e é feita a análise de resultados dos modelos criados.

A investigação e experimentação realizada abre perspetivas para o futuro da aplicação de soluções de Inteligência Artificial na área da contratação pública.

Palavras-chave: Machine Learning, Data Mining, Contratação Pública, Extração de Conhecimento

Abstract

Over the years, with the evolution of technology, there has been a sharp increase in the number of data accumulated in the public procurement sector in Portugal. This increase paves the way for the possibility of taking advantage of this data by using emerging techniques of artificial intelligence, in order to improve the functioning of the public procurement process in Portugal.

Within the public procurement panorama, one of the major problems affecting the quality of the contracts signed is the existence of irregularities that are not detected when the contracts are made. These irregularities thus generate contracts that do not comply with the rules defined for contracts and that can compromise the quality of the goods and services provided by the State.

Therefore, this dissertation aims to collect a dataset of public contracts from the portal provided by the Portuguese government, process and analyze the collected data, codify the rules in a rule-based system, investigate and use Artificial Intelligence techniques in order to develop a pipeline with the purpose of finding patterns of suspicion of collusion and detecting irregularities. Finally, there is also the objective of designing and developing Machine Learning models to predict the future values of expenditure of each city or region.

In this work were also discussed implementations of artificial intelligence techniques in the public procurement field, along with the analysis of fraud detection approaches, as well as ML paradigms and algorithms.

In addition, it is demonstrated how ML models have been designed and optimized, followed by the analysis of the results of each model.

The research and experimentation conducted opens up perspectives for the future of the application of Artificial Intelligence solutions in the area of public procurement.

Keywords: Machine Learning, Data Mining, Public Procurement, Knowledge Extraction

Índice

Lista de Figuras	ix
Lista de Tabelas	xi
Listagens	xii
Siglas	xiii
1 Introdução	1
1.1 Contextualização e Motivação	1
1.2 Questões de Investigação e Objetivos	2
1.3 O problema e os seus desafios	3
1.3.1 Problema	3
1.3.2 Desafios	4
1.4 Metodologia	4
1.5 Estrutura	5
2 Fundamentação Teórica	6
2.1 Extração de Conhecimento	6
2.1.1 Pré-processamento e transformação de dados	7
2.1.2 Data Mining	10
2.2 Sistemas de Suporte à Decisão	12
2.3 Rule Based System	12
2.4 Web Scraping	13
2.5 Machine Learning	14
2.5.1 Aprendizagem Supervisionada	15
2.5.2 Aprendizagem não Supervisionada	16
2.5.3 Deep Learning	17
2.6 Grafos de Conhecimento	18
2.6.1 <i>Degree centrality</i>	18
2.6.2 <i>Closeness centrality</i>	19

2.6.3	<i>Eigenvector centrality</i>	19
2.6.4	PageRank	19
2.7	Conclusão	21
3	Estado da Arte	22
3.1	Contratação Pública	22
3.1.1	Deteção de Irregularidades em contratos públicos	23
3.1.2	Conluio na Contratação Pública	24
3.2	<i>Machine Learning</i> na Contratação Pública	25
3.3	Sistemas de Suporte à Decisão na Contratação Pública	26
3.4	Análise de Redes Sociais	27
3.5	<i>Time series forecasting</i>	28
3.6	Conclusão	28
4	Arquitetura da Solução e Tratamento de Dados	30
4.1	<i>Pipeline</i> da Abordagem	30
4.2	Tecnologias	31
4.2.1	<i>Beautiful Soup</i>	32
4.2.2	NetworkX	32
4.2.3	Scikit learn	32
4.2.4	Keras	32
4.3	Ferramentas	32
4.4	Aquisição de Dados	33
4.5	Análise de dados	37
4.5.1	Compreensão de dados	37
4.5.2	Exploração do <i>dataset</i>	38
5	Desenvolvimento	42
5.1	Implementação e análise de redes sociais	42
5.2	Deteção de indicadores de conluio	44
5.2.1	Deteção de Red Flags de Conluio	45
5.2.2	Avaliação do risco de conluio	51
5.3	Deteção de Irregularidades	52
5.3.1	<i>Rule Based System</i>	52
5.3.2	Classificação de irregularidades em contratos	53
5.3.3	Conceção dos Modelos	57
5.3.4	Análise dos resultados obtidos	62
5.4	Utilização de <i>Machine Learning</i> para previsão de gastos em contratos públicos	63
5.4.1	Preparação de dados	63

5.4.2	Algoritmos e métricas de <i>performance</i> escolhidas	64
5.4.3	Modelos de <i>Machine Learning</i>	64
5.4.4	Análise dos resultados obtidos	66
5.5	Interface de visualização de dados	69
5.5.1	Visualização geográfica da contratação	69
5.5.2	Visualização gráfica de dados	70
5.5.3	Teste de irregularidade de contratos	71
6	Conclusão e trabalho futuro	72
6.1	Conclusão	72
6.2	Trabalho futuro	73
	Bibliografia	75

Lista de Figuras

2.1	Processo de Extração de Conhecimento[2]	7
2.2	Taxonomia dos métodos de Data Mining [6]	11
2.3	Exemplo de um <i>Rule Based System</i> para escolha de um sistema operativo [35]	13
2.4	Enquadramento de <i>Machine Learning</i> na Inteligência artificial [30]	15
2.5	Demonstração visual do resultado da aplicação do algoritmo de PageRank num grafo [7]	20
4.1	Representação da arquitetura do sistema	31
4.2	Disposição da lista de contratos no portal Base.gov	33
4.3	Disposição da informação de um contrato no portal Base.gov	34
4.4	Dataframe com dados extraídos do Portal	36
4.5	Informação as variáveis dos dados recolhidos	37
4.6	Número total de contratos depositados no portal Base em cada mês	39
4.7	Preço total em euros gasto em contratos em cada mês	39
4.8	Distribuição do dinheiro despendido por cada distrito em cada CPV	40
4.9	Número de contratos por distrito e por tipo de procedimento	41
4.10	Percentagem de despesa referente a cada categoria de CPV	41
5.1	Grafo representativo das entidades dos contratos da localidade de Vila Nova de Famalicão	43
5.2	Top 5 entidades do grafo com maior <i>eigenvector centrality</i>	44
5.3	Top 5 entidades do grafo com maior <i>degree centrality</i>	44
5.4	Exemplos de comunidades encontradas pelo algoritmo Girvan Newman	44
5.5	Curva de Lorenz para os contratos do <i>dataset</i>	46
5.6	Contagem do número de aparições de cada tipo de procedimento no <i>dataset</i>	47
5.7	Contagem do dinheiro gasto em cada tipo de procedimento no <i>dataset</i>	47
5.8	Distribuição do número de concorrentes de cada concurso público presente no <i>dataset</i>	48
5.9	Contagem dos contratos do <i>dataset</i> quanto ao valor do atributo HigherValue	49
5.10	Gráfico do número médio de <i>red flags</i> acionadas pelos contratos de cada uma das empresas referenciadas	52
5.11	Matriz de correlação das variáveis do <i>dataset</i>	55
5.12	Rankeamento de variáveis a partir da classe selectKbest	55
5.13	Balanceamento da classe <i>target</i>	56

5.14	Balanceamento após aplicação de <i>sampling</i>	56
5.15	Variáveis discretas após aplicação de <i>label encoding</i>	57
5.16	Comparação dos valores reais com os valores previstos pela melhor abordagem testada	68
5.17	Aba de visualização geográfica	70
5.18	Aba de gráficos gerais do país	70
5.19	Aba de gráficos de uma localidade específica	71

Lista de Tabelas

5.1	Valores máximos para cada combinação de tipo de contrato e procedimento	53
5.2	Métricas de performance	59
5.3	Resultados do teste de <i>performance</i> dos algoritmos	60
5.4	Hiperparâmetros utilizados no <i>tuning GridSearch</i>	61
5.5	Resultados do teste de <i>performance</i> do algoritmo MLP	61
5.6	Hiperparâmetros utilizados no <i>tuning GridSearch</i>	61
5.7	Resultados do teste de <i>performance</i> do algoritmo <i>Random Forest</i>	61
5.8	Hiperparâmetros utilizados no <i>tuning GridSearch</i>	62
5.9	Resultados do teste de <i>performance</i> do algoritmo <i>Gradient Boosting</i>	62
5.10	Métricas de <i>performance</i> de regressão utilizadas	64
5.11	Resultados do teste de <i>performance</i> dos algoritmos LSTM	67
5.12	Resultados do teste de <i>performance</i> dos algoritmos GRU	67

Listagens

5.1	Função que sinaliza o percentil 90 das entidades com mais dinheiro recebido em contratos	46
5.2	Função de deteção da utilização excessiva de ajustes	47
5.3	Função de deteção de contratos com valor excessivo	48
5.4	Função de contagem de derrotados	49
5.5	Função de vitórias e derrotas de uma empresa numa região	50
5.6	Implementações dos algoritmos de <i>Machine Learning</i>	58
5.7	Cálculo de métricas de avaliação dos algoritmos	58
5.8	Função de implementação da rede LSTM	65
5.9	Função de implementação da rede GRU	65
5.10	Implementações de XGBoost e Random Forest Regressor	66

Siglas

AdC	Autoridade da Concorrência
CCP	Código dos Contratos Públicos
CEIS	Cadastro de Empresas Inidôneas e Suspensas
CPV	Vocabulário Comum para os Contratos Públicos
FN	False Negative
FP	False Positive
GRU	<i>Gated recurrent unit</i>
IA	Inteligência Artificial
LSTM	<i>Long short-term memory</i>
ML	<i>Machine Learning</i>
OCDE	Organização para a Cooperação e Desenvolvimento Económico
OCDS	Open Contracting Data Standard
PIB	Produto Interno Bruto
RNN	<i>Recurrent neural network</i>
TN	True Negative
TP	True Positive
WGI	Worldwide Governance Indicators

Introdução

Nesta secção serão abordadas a motivação e contextualização desta dissertação. Serão também definidos os principais objetivos da mesma e a abordagem planeada para os alcançar e validar. De seguida, serão descritas as questões de investigação e a metodologia utilizada. Por fim será debatido o problema inerente à dissertação e os principais desafios encontrados.

1.1 Contextualização e Motivação

Com a evolução da tecnologia, a quantidade de dados acumulados no mundo duplica a cada vinte meses[1], no entanto, frequentemente esses dados são subaproveitados pelas entidades que os acumulam, sendo apenas utilizados como registos históricos e não como uma forma de entendimento do fenómeno intrínseco à ocorrência dos dados. Posto isto, os dados acumulados podem ter um valor maior se as instituições forem capazes de utilizar estratégias como:

- Analisar e entender os dados;
- Prever comportamentos;
- Identificar tendências;

O setor da contratação pública não é exceção. Há uma enorme quantidade de registos de contratos celebrados que não são aproveitados para extrair conhecimento, tal como existem diversos processos relacionados com a feitura de contratos públicos que poderiam possivelmente beneficiar de ser renovados para estarem enquadrados na nova era digital[25].

Contratação pública pode ser definida como a aquisição de bens ou serviços, feita por um departamento do governo ou por qualquer empresa de propriedade do governo. Esta representa uma grande parte dos gastos de um país, tendo sido estimado pela OCDE[69] que a contratação pública corresponde

a 12% do Produto Interno Bruto e 29% das despesas dos seus estados-membros, dos quais também faz parte Portugal. Para além da sua extrema importância na economia dos países, a contratação pública é também a atividade governamental mais propícia a fraudes e corrupção devido ao grande fluxo monetário gerado [69], o que leva a perdas de biliões de euros todos os anos, tanto em países desenvolvidos como em países subdesenvolvidos.

Devido ao elevado risco de fraudes e corrupção neste setor, é de extrema importância o desenvolvimento de uma ferramenta capaz de detetar indícios destas atividades prejudiciais em contratos públicos[27] para que sejam evitadas perdas monetárias avultadas para o Estado português, e também, para que sejam providenciados bens e serviços públicos com a melhor qualidade possível à população.

A principal motivação deste trabalho é a de realizar uma análise acerca da aplicabilidade de diversas técnicas de Inteligência Artificial no âmbito da otimização dos processos de contratação pública, com o objetivo de apresentar por fim um *pipeline* que quando aplicado a um *dataset* de contratos permita extrair conhecimento capaz de auxiliar a feitura de um contrato de modo a diminuir o número de contratos irregulares, e também, maximizar a qualidade do bem ou serviço contratado.

O facto do objetivo desta dissertação envolver o estudo e implementação de tecnologias emergentes como *Machine Learning* foi uma forte motivação para o seu desenvolvimento, de modo a poder contribuir para a consolidação do uso das técnicas de inteligência artificial no campo da contratação pública.

1.2 Questões de Investigação e Objetivos

Nos parágrafos seguintes serão expostas algumas questões relacionadas com os objetivos desta dissertação que devem ser respondidas durante a conceção da mesma.

Questão de investigação no.1: É possível utilizar técnicas de Inteligência Artificial como forma de detetar irregularidades em contratos públicos?

Objetivo: O objetivo principal desta questão é identificar as técnicas de Inteligência Artificial mais apropriadas à criação de um *pipeline* de identificação de irregularidades em contratos públicos.

Validação: Para validar esta questão será implementado um *pipeline* de deteção de irregularidades num conjunto de contratos de dimensão significativa, retirados do portal Base.gov, e serão posteriormente analisados os resultados obtidos.

Questão de investigação no.2: É possível prever custos futuros em contratação pública com base nos custos do passado de forma a que os valores possam ser usados para otimização do planeamento de gastos pelas entidades responsáveis pela contratação?

Objetivo: O objetivo principal desta questão é implementar um *pipeline* com recurso a técnicas de *Machine Learning* que seja capaz de aprender padrões de compra baseados no passado, e com isso, prever os gastos futuros em contratação pública para uma certa localidade, produto ou setor de contratação.

Validação: Para validar esta questão será implementado o *pipeline* de previsão e serão analisadas as métricas de *performance* de previsão produzidas, de modo a verificar se a taxa de erro produzida

demonstra ou não indícios de aplicabilidade no mundo real como ferramenta de auxílio.

Questão de investigação no.3: É possível utilizar técnicas de Inteligência Artificial para identificar padrões de conluio em contratos públicos?

Objetivo: O objetivo principal desta questão é identificar na literatura existente um conjunto de regras adequadas para a identificação de padrões de conluio em contratos públicos e codificar essas regras num *rule-based system* que permita aplicar essas regras a um conjunto de contratos.

Validação: Para validar esta questão será aplicado o conjunto de regras criado a contratos de empresas multadas pela Autoridade da Concorrência pela prática de conluio, e será analisado se o conjunto de regras foi capaz de identificar algum dos padrões de conluio.

Questão de investigação no.4: Análise de redes sociais pode ser utilizada como forma de detecção de comunidades e padrões de conluio em contratos públicos?

Objetivo: O objetivo principal desta questão é a conceção de grafos de conhecimento a partir de um conjunto de dados de contratação pública e aplicação de algoritmos de análise de redes sociais.

Validação: Esta questão será considerada validada caso os resultados da aplicação dos algoritmos de redes sociais consigam identificar comunidades e produza indicadores correlacionados com os indicadores de conluio criados anteriormente.

1.3 O problema e os seus desafios

O propósito desta secção é explicar o impacto positivo que pode ter a utilização de inteligência artificial no âmbito da contratação pública e de seguida abordar os desafios que serão enfrentados.

1.3.1 Problema

O problema central desta dissertação é o subaproveitamento do sistema atual de contratação pública em extrair conhecimento a partir do grande número de contratos celebrados acumulados no portal Base, algo que poderia ser utilizado como forma de auxílio para mitigar problemas causadores de consequências financeiras avultadas como irregularidades e conluio. Na comunidade tecnológica, existem diversos artigos que apontam para um futuro da contratação pública associado à utilização de inteligência artificial. O U4 *Anti-Corruption Resource Centre*, um centro norueguês com o objetivo de diminuir o impacto da corrupção na sociedade aponta o uso de IA, como uma forma promissora de combate à corrupção na contratação pública, com o potencial de substituir os métodos atuais de feitura de contratos, propensos à corrupção[46]. A capacidade das técnicas de IA de analisar *datasets* demasiado grandes para serem analisados manualmente faz com que estas técnicas sejam consideradas promissoras para encontrar irregularidades e padrões que neste momento são quase impossíveis de detetar.

Deste modo, a implementação de um *pipeline* para extração de conhecimento a partir de contratos públicos, capaz de identificar padrões de fraude e de irregularidades durante a feitura de contratos poderá

contribuir para cimentar o futuro da utilização da IA como contributo para uma melhoria da eficiência do sistema de contratação pública.

1.3.2 Desafios

Um dos desafios desta dissertação é conseguir ultrapassar a escassez de trabalhos, projetos e artigos científicos disponíveis com implementações práticas de inteligência artificial no campo da contratação pública, isto dificultou acentuadamente o processo de criação da arquitetura da solução, levando inicialmente a sucessivas fases de tentativa e erro durante a experimentação de diversas abordagens, algo que poderia ser evitado caso houvesse um maior número de artigos com abordagens semelhantes à que foi desenvolvida neste trabalho pois permitiria que fossem tiradas ilações a partir das conclusões de outros autores.

Outro grande desafio apresentado foi o de dominar todo o contexto à volta da contratação pública, tais como as especificidades de cada conjunto de atributos associado a cada contrato e todas as consequências provocadas por esses atributos nas quantias monetárias mínimas e máximas permitidas para cada contrato. Para isto, foi de elevada importância efetuar várias pesquisas acerca dos significados de diversos termos técnicos desta área, permitindo assim uma leitura eficiente do documento do Código dos Contratos Públicos com vista a sintetizar da melhor forma possível as regras nele presentes.

1.4 Metodologia

A metodologia seguida nesta dissertação foi escolhida tendo em conta a sua aplicabilidade em responder às questões de investigação definidas. A escolha recaiu numa combinação do método indutivo, em que o investigador recolhe dados e fórmula a teoria a partir da análise dos dados recolhidos, com o método dedutivo, em que o investigador formula hipóteses e define uma estratégia para testar as hipóteses formuladas. Estes dois métodos são complementares e podem ser combinados num método hipotético-dedutivo que inicia com a formulação de teoria acerca do problema e de seguida formula e testa hipóteses a partir da teoria formulada. O algoritmo da metodologia escolhida é constituído pelos passos seguintes:

1. Formulação de teoria (contexto geral do problema)
2. Formulação de hipóteses testáveis
3. Realização de previsões baseadas nas hipóteses formuladas
4. Recolha e análise de dados para o teste das hipóteses
5. Teste das hipóteses (caso a hipótese seja refutada é retomado o passo número 2)
6. Confirmação da teoria

1.5 Estrutura

Este documento contém as secções de Introdução, Fundamentação Teórica, Estado da Arte, Arquitetura da Solução, Desenvolvimento e Conclusão e Trabalho Futuro.

Na secção da fundamentação teórica o objetivo é abordar as especificidades acerca dos temas, técnicas e ferramentas essenciais para o desenvolvimento do projeto.

Na secção do estado da arte são descritas diversas abordagens que durante o processo de revisão da literatura foram consideradas relevantes no contexto do desenvolvimento deste estudo, baseiam-se na utilização de *Machine Learning*, redes sociais, e sistemas de suporte à decisão no âmbito da contratação pública.

Na secção de arquitetura da solução e tratamento de dados é abordada a arquitetura do *pipeline* e as tecnologias utilizadas, bem como a captura, pré-processamento e análise de dados.

Na secção de desenvolvimento é apresentada a abordagem proposta, o trabalho feito até ao momento, e a análise dos resultados obtidos.

Por fim, na última secção são apresentadas as conclusões retiradas a partir do trabalho realizado, bem como os próximos passos que devem ser tomados no contexto deste estudo e uma planificação temporal do trabalho futuro.

Fundamentação Teórica

Neste capítulo serão abordados conceitos relacionados com a base teórica do campo informático relacionados com os processos de recolha, tratamento e extração de conhecimento a partir de dados presentes em um *website*. No campo da extração de conhecimento serão abordados sistemas baseados em regras, grafos de conhecimento, e uma visão global acerca dos paradigmas e algoritmos de *Machine Learning*.

2.1 Extração de Conhecimento

Extração de Conhecimento é o nome dado ao processo de descoberta de conhecimento útil a partir de dados, processo que é subdividido nos passos de seleção, pré-processamento e transformação de dados, *data mining* e avaliação de resultados.

Segundo Fayyad et al.[2] o método tradicional de transformar dados em conhecimento é feito através da análise manual e consequente interpretação, como por exemplo, se pode observar na área da medicina, em que os especialistas são capazes de analisar dados imagiológicos, e a partir deles extrair conhecimento e efetuar uma inferência sobre o que essa imagem médica representa para ele. Para além deste exemplo, existem diversas áreas onde a análise de dados é feita manualmente, tais como geologia, finanças ou marketing. O processo de extração de conhecimento, representado na figura 2.1 tem os seguintes passos[2]:

1. Identificação dos objetivos do procedimento e recolha de conhecimento sobre o domínio da aplicação.
2. Escolha do *dataset* para recolha, ou então, de um *subset* de variáveis ou *data samples*, que serão utilizados para a extração.

3. Pré-processamento de dados. Processo que engloba a remoção do *noise*, *outliers* e outros indicadores que afetem negativamente a qualidade dos dados, tal como a decisão do procedimento a tomar com valores em falta.
4. Remoção de dados, por exemplo, remover parâmetros e variáveis que não sejam relevantes para o objetivo proposto.
5. Decisão do método de *data mining* mais ajustado ao objetivo definido para o processo de extração, por exemplo, ponderação da possível aplicação de *clustering*, regressão, classificação ou sumarização.
6. Escolha do algoritmo de *data mining* a utilizar. Esta fase depende da preferência pessoal do utilizador, que pode preferir um algoritmo que maximize a eficácia de previsão ou um algoritmo que facilite o entendimento do modelo.
7. Aplicação do algoritmo escolhido no passo 6 ao *dataset* escolhido no passo 2. O sucesso deste passo está dependente da correta aplicação dos passos anteriores.
8. Interpretação de resultados e possível retorno a passos anteriores de modo a reajustar o modelo.
9. Incorporação do conhecimento gerado num sistema para tomada de decisão face a problemas do mundo real, ou simplesmente documentação do processo e resultados obtidos de modo a auxiliar potenciais interessados no método de extração realizado.

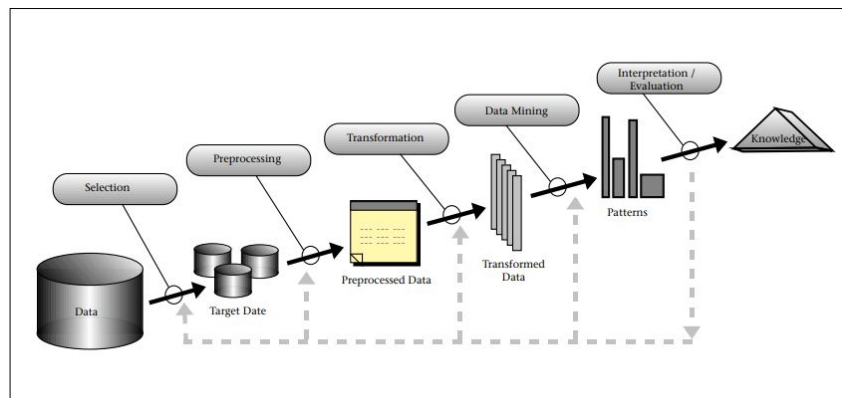


Figura 2.1: Processo de Extração de Conhecimento[2]

2.1.1 Pré-processamento e transformação de dados

Os dados existentes na *Web* são desestruturados e provenientes de fontes heterogêneas, o que leva a que antes de efetuar tarefas de *Data Mining* seja necessário efetuar pré-processamento de dados de modo converter os dados num formato estruturado e sem incongruências[55].

Considerando que a existência de dados incongruentes, incompletos ou imprecisos leva a que os modelos sejam mais menos fiáveis e tenham previsões menos eficazes quando utilizados em algoritmos de *Machine Learning*, o pré-processamento de dados torna-se um passo de extrema importância no *pipeline* de construção de ferramentas de *Data Science*[57]. Segundo Vijayashri et al.[59] acima de 80% do tempo total despendido num projeto de *data mining* é passado na fase de pré-processamento de dados, o que realça a extrema importância deste passo para a qualidade da solução.

Durante a recolha de dados podem surgir dados inconvenientes no *dataset*, provenientes de diversas causas, tais como, erros humanos durante a inserção de dados, informação incorreta inserida deliberadamente por utilizadores, inserção de dados duplicados ou inserção de dados incompletos.

De modo a conseguir a eliminação ou minimização dos problemas do *dataset* que afetam negativamente a *performance* de generalização dos algoritmos de *Machine Learning*, existem diversos métodos de pré-processamento de dados que podem ser utilizados no *dataset*, tais como deteção de *outliers*, balanceamento de dados, tratamento de valores em falta, discretização, normalização de dados, seleção de atributos e construção de atributos[57].

2.1.1.1 Deteção de *outliers*

O problema dos *outliers* é um dos mais antigos do campo da estatística[52]. Intuitivamente, um *outlier* pode ser definido como uma observação que se desvia tanto do padrão das observações, que levanta suspeições de não ter sido gerada pelo mesmo método que gerou as observações restantes.

Segundo Overbay et al.[5] existe um forte debate quanto ao procedimento a tomar com os *outliers*, sendo que existem argumentos fortes a favor da sua eliminação, e também da sua conservação. Por um lado, existem afirmações de que o *dataset* é mais representativo da sua população caso sejam mantidos os *outliers*, por outro lado, na maioria dos algoritmos de classificação testados no artigo citado, os valores de erro de inferência foram reduzidos quando utilizado o *dataset* sem os *outliers* face ao *dataset* completo.

No caso de atributos exclusivamente numéricos, a deteção de *outliers* pode ser feita de várias formas, entre elas, a comparação do valor do atributo com a mediana, média ou com os percentis 25 e 75 da distribuição estatística de valores desse atributo no *dataset*[57].

2.1.1.2 Balanceamento do *dataset*

Um *dataset* considera-se desbalanceado em problemas de classificação quando a maioria dos elementos está classificado com uma classe, enquanto a classe restante tem um número bastante menor de elementos. O desbalanceamento do *dataset* pode fazer com que diversos algoritmos de classificação, procurando maximizar a sua *accuracy* tendam a ter uma tendência a classificar maioritariamente os objetos como sendo da classe maioritária, e em casos extremos chegam até a ignorar por completo a classe minoritária[24].

Existem diversos métodos criados para mitigar o problema do desbalanceamento, entre eles, as técnicas de *undersampling* e *oversampling*. *Undersampling* é um método não-heurístico que visa balancear

a distribuição do *dataset* através da eliminação aleatória de elementos da classe majoritária. O maior problema deste método é que através da eliminação aleatória podem ser descartados dados potencialmente úteis para o processo de indução[58]. *Oversampling* é um método não-heurístico que visa balancear a distribuição do *dataset* através da replicação aleatória de elementos da classe minoritária. Este método pode causar um aumento da probabilidade de *overfitting* dos modelos de *Machine Learning* devido ao facto de poder inserir no *dataset* diversas cópias exatas de elementos da classe minoritária, dessa forma o algoritmo classificador pode, por exemplo, construir regras que aparentemente são eficazes, mas, na verdade, cobrem apenas um elemento que foi replicado. Este método pode também acarretar um grande esforço computacional adicional para um processo de aprendizagem caso o *dataset* seja de tamanho elevado[58].

2.1.1.3 Tratamento de valores em falta

O tratamento de valores em falta decide de que forma lidar com atributos com valores em falta ou valores nulos num objeto do *dataset*, existem diversas formas de realizar este tratamento, entre elas[19]:

- Métodos de eliminação, que decidem quando eliminar ou manter entradas com dados em falta.
- Substituir pelo valor médio ou mediana da observação.
- Interpolação dos valores tomados na entrada anterior e seguinte do campo em falta, método denominado por interpolação linear.
- Utilizar um modelo de regressão de modo a prever esse valor.
- Procurar o objeto mais semelhante com o objeto em questão e substituir o valor em falta pelo valor tomado nesse mesmo atributo no objeto considerado semelhante, processo chamado de *hot-deck imputation*.
- Substituir o dado em falta por uma constante proveniente de uma fonte externa (e.g versões antigas desse *dataset*), processo chamado de *cold-deck imputation*.

2.1.1.4 Escalonamento de dados

Escalonamento de dados é o processo de tratamento de dados em que é efetuada uma redução na escala dos valores dos atributos, ficando reduzidos a uma escala de valores menor que a inicial. Este processo tem como objetivo colocar todos os atributos na mesma faixa de valores de modo que as diferentes faixas de valores de cada atributo não interfiram na aprendizagem dos algoritmos[21]. Este processo é especialmente importante para redes neuronais e KNN[57], visto que estes algoritmos aprendem a classificar através de cálculos de distâncias entre dados o que leva a que as *features* com maior escala tenham passem indevidamente a ter um peso excessivo na classificação produzida pelos algoritmos.

Escalonamento de dados pode ser feito através da normalização e também da estandardização.

Normalização é o processo de tratamento de dados em que é efetuada uma redução na escala dos valores dos atributos, ficando reduzidos a uma escala específica, normalmente entre 0 e 1 ou -1 e 1.

No processo de standardização a escala de valores do atributo passam a ter uma distribuição com média de 0 e um desvio padrão de 1. Este método é especialmente útil quando os dados seguem uma distribuição gaussiana, caso contrário a sua utilização pode prejudicar a aprendizagem[32].

2.1.1.5 Discretização

Discretização é o processo no qual se substituem conjuntos de valores numéricos contínuos por variáveis discretas, associando cada intervalo de valores a uma categoria. Isto reduz significativamente o número de possíveis valores do atributo, sendo que a existência de uma gama elevada de valores possíveis numa variável contribui para ineficácia e lentidão do processo de aprendizagem de algoritmos indutivos de *Machine Learning*[18].

2.1.1.6 Seleção de atributos

Seleção de atributos é o processo de identificação e remoção de atributos irrelevantes, redundantes ou ruidosos do *dataset*, ou seja, atributos que têm uma influência nula, reduzida ou negativa na classificação dos objetos. Isto reduz a dimensão dos dados e permite aos modelos operar mais rápida e eficazmente e de forma mais facilmente interpretável[16].

Existem diversos métodos de seleção de atributos dependendo do tipo de categoria do *dataset* (e.g discretas ou contínuas), cada método usa uma função de avaliação que decide os atributos mais vantajosos para o *dataset*, a função de avaliação divide-se nos 4 subgrupos seguintes [57]:

- Distância: Num problema de 2 classes possíveis, o atributo X é utilizado em detrimento ao atributo Y se X produz uma maior diferença nas probabilidades de classificação do que Y.
- Informação: O atributo X é preferido em relação ao Y se o ganho de informação com o atributo X for superior ao ganho de informação com o atributo Y.
- Dependência: Se a correlação do atributo X com uma classe C for maior que a correlação de Y com essa classe, é preferido o atributo X.
- Consistência: Dois objetos são inconsistentes se têm os mesmos valores para os mesmos atributos, mas diferem na classe atribuída.

2.1.2 Data Mining

Atualmente com a evolução da tecnologia, os volumes de dados tornaram-se maiores e a sua acumulação tornou-se mais barata, como consequência, as bases de dados de empresas e entidades governamentais

contêm atualmente quantidades massivas de dados, é estimado que a quantidade de dados acumulados no mundo duplica a cada vinte meses [1], neste grande volume de dados pode existir informação potencialmente útil que não é aproveitada.

Data Mining é um processo que recorre a técnicas estatísticas, matemáticas, de inteligência artificial, e de *Machine Learning* de modo a conseguir extrair e identificar informação útil e consequentemente conhecimento a partir de grandes bases de dados.

Data mining foi também definido em Fayyad et al.[2] como o processo não trivial de procura de padrões válidos, novos, e potencialmente úteis num conjunto de dados.

O processo de *Data mining* taxonomicamente, como representado na figura 2.2, pode ser orientado à descoberta e orientado à verificação.

Os métodos de descoberta são métodos capazes de identificar automaticamente padrões num conjunto de dados e são métodos maioritariamente baseados em aprendizagem indutiva, ou seja, os modelos são maioritariamente construídos a partir da generalização de um conjunto de dados de treino.[6]. Os métodos de descoberta são por sua vez subdivididos em métodos de previsão e métodos de descrição. Métodos de previsão têm como objectivo construir um modelo capaz de prever o valor de uma ou mais variáveis de uma amostra estudada. Métodos de descrição por sua vez são métodos mais orientados à interpretação dos dados, focando-se na demonstração de padrões e relacionamentos de um conjunto de dados num formato capaz de ser entendido por um ser humano.

Os métodos orientados à verificação, pelo contrário dos de descoberta, avaliam hipóteses fornecidas por fontes externas, como por exemplo, por especialistas. As técnicas de *Data Mining* têm aplicações em áreas como deteção de fraudes em cartões de crédito, análise de relatórios médicos, previsão de hábitos de compras, previsão de interesses de utilizadores da *Web* e otimização de processos industriais[66].

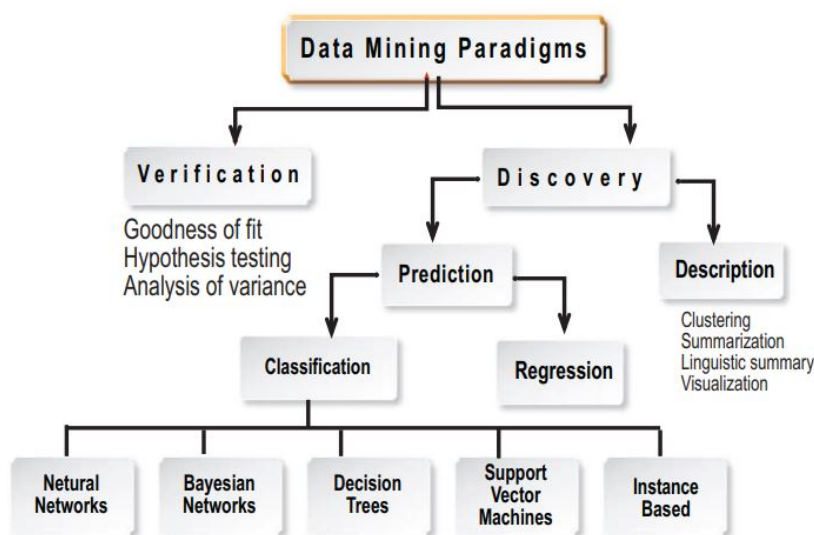


Figura 2.2: Taxonomia dos métodos de Data Mining [6]

2.2 Sistemas de Suporte à Decisão

Marakas[61] definiu um sistema de suporte como um sistema sob controlo de um ou mais tomadores de decisão que assistem na atividade de tomada de decisão, fornecendo um conjunto de ferramentas organizadas com o objetivo de estruturar cada passo de tomada de decisão e melhorar a eficácia da decisão geral tomada. Para além disso também deve ser tomado em consideração aquando do uso de sistema de suporte à decisão que estes não garantem benefícios em todas as decisões tomadas, e que o seu propósito é de providenciar ao utilizador grandes quantidades de informação durante a tomada de decisão [61]. Por outro lado, sistemas de suporte à decisão foram também definidos como sistemas de computação que apoiam a tomada de decisão, providenciando sistemas especializados e análise da decisão a tomar com base em múltiplos critérios[67].

Um sistema de suporte à decisão é necessário quando a tomada de decisão é feita no âmbito da resolução de problemas que combinem a necessidade de julgamento humano com processos de solução padrão (e.g decisões semi-estruturadas), e é tipicamente composto por uma base de dados com informação sobre o problema em questão, um modelo adequado ao funcionamento do problema e uma *user interface* [12].

Este tipo de ferramentas foram inicialmente desenvolvidas para operações empresariais baseadas em atividades relacionadas com investimentos e tendências de mercado, mas atualmente têm aplicações na área do apoio governamental na tomada de decisões estratégicas para fins de planeamento sócio-económico e de desenvolvimento em pública administração [56].

2.3 Rule Based System

Gegov et al.[14] definiu *Rule Based System* como programas baseados num conjunto de regras *if-then*, como o representado na figura 2.3 em que o seu objetivo em particular é retratar todas as condições e respetivas consequências para auxiliar a escolha de um sistema operativo. Esses sistemas têm a capacidade de auxiliar sistemas de suporte à decisão ou tomadas de decisão por inferência em aplicações[14].

A construção destes sistemas pode ser baseada em raciocínio dedutivo encadeado para a frente ou raciocínio dedutivo encadeado para trás.

Raciocínio dedutivo encadeado para a frente ou *Data Based Approach* é o processo de construção de sistemas baseados em regras que é desencadeado pela introdução de dados no sistema pelo utilizador ou por resultados obtidos pelo próprio sistema. À medida que são recebidos factos o sistema verifica as regras que têm condições satisfeitas pelos factos introduzidos, e a partir das regras que têm todas as suas premissas satisfeitas, são produzidas as conclusões, conclusões essas que podem satisfazer premissas de outras regras[35].

Raciocínio dedutivo encadeado para trás ou *Knowledge Based Approach* é o processo de construção de sistemas baseados em regras que é desencadeado através de "perguntas" ao sistema, quando é recebida uma "pergunta" o sistema percorre a sua base de conhecimentos à procura de factos ou regras que

permitam responder à pergunta do utilizador. Se não encontrar, falha. Caso encontre, verifica se os factos recebidos satisfazem as condições de alguma das regras que respondem à pergunta, se esta verificação for bem sucedida, é devolvida a conclusão da regra cujas premissas foram satisfeitas ao utilizador[35].

```

Regra 1
SE
  o ambiente de utilização é um ambiente empresarial; e
  é necessário usar software aplicativo da Microsoft
ENTÃO
  o sistema operativo seleccionado é o Windows NT

Regra 2
SE
  o ambiente de utilização é um ambiente particular; e
  é necessário usar software aplicativo da Microsoft
ENTÃO
  o sistema operativo seleccionado é o Windows98

Regra 3
SE
  o sistema operativo seleccionado é o Windows NT; e
  é necessário disponibilizar serviços a vários clientes
ENTÃO
  a versão do sistema operativo é Windows NT Server

Regra 4
SE
  é necessário usar Excel
ENTÃO
  é necessário usar software aplicativo da Microsoft

Regra 5
SE
  é necessário usar Access
ENTÃO
  é necessário usar software aplicativo da Microsoft

```

Figura 2.3: Exemplo de um *Rule Based System* para escolha de um sistema operativo [35]

2.4 Web Scraping

Web Scraping é uma técnica utilizada para extrair grandes volumes de dados a partir de *websites*, sendo que os dados são armazenados posteriormente localmente no computador ou numa base de dados tabular (do tipo *spreadsheet*). A maioria dos dados dos *websites* só podem ser visualizados através de um *browser* de *web*, mesmo listas de dados como por exemplo, páginas amarelas, e grande parte dos *websites* não fornece ao utilizador uma funcionalidade de descarregar os dados existentes, sendo que guardar manualmente os dados pode demorar horas ou até mesmo dias [37].

O processo de *Web Scraping* é composto por duas técnicas, *crawling* e *parsing*. *Crawling* é o processo no qual o *bot* implementado, denominado de *crawler* entra no URL da página pedida, e através de algoritmos usualmente recursivos, percorre a página, guarda o HTML da página, e procura os *links* existentes e guarda-os numa estrutura de dados. Posteriormente vai às páginas dos *links* obtidos e procura mais *links* existentes nelas, repetindo este passo até que não existam mais *links* referenciados nas páginas. *Parsing* é a técnica na qual o *parser* analisa os conteúdos das páginas obtidas pelo *crawler*, a partir dos quais faz uma extração dos dados da informação não estruturada presente armazenando-a de forma estruturada, seja em CSV, Json, ou outro formato semelhante [60].

Atualmente esta técnica tem aplicação em diversas áreas, entre as quais [71]

- *Marketing*;

- Geração de *leads*;
- Análise de concorrência (eg. comparar preços, artigos, etc.);
- Análise de reputação de empresas, pessoas ou produtos;
- Monitorização de notícias;
- Aglomeração de ofertas de emprego de diferentes *websites*;
- Extração de valores do mercado de ações (eg. valorização de empresas);

2.5 Machine Learning

Segundo Liu et al.[15] *machine learning* é o campo da inteligência artificial com o propósito de desenvolvimento de modelos computacionais de aprendizagem. Do ponto de vista computacional, *machine learning* é referido como a habilidade de uma máquina de melhorar a sua *performance* baseada em resultados anteriores.

Por outro lado, segundo Mitchell[66], cada problema de *Machine Learning* pode ser definido como um problema de melhorar a *performance* ao executar uma tarefa, a partir de um certo treino.

Atualmente, este campo da inteligência artificial tem diversas aplicações como reconhecimento facial, reconhecimento por voz, sistemas de recomendação ou problemas de classificação de objetos.

Como é demonstrado na figura 2.4 *Machine Learning* para além de ser um ramo da Inteligência Artificial, é por sua vez subdividido em diferentes paradigmas não só como *Deep Learning*, mas também *Semi-Supervised Learning*, *Supervised Learning*, *Unsupervised Learning* e *Reinforcement Learning*, que apesar de terem uma origem comum, cada um deles tem diferentes implementações e aplicações em contextos distintos.

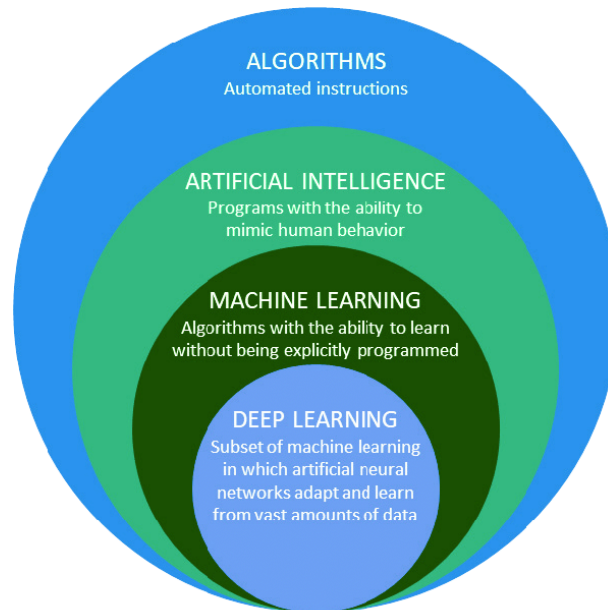


Figura 2.4: Enquadramento de *Machine Learning* na Inteligência artificial [30]

2.5.1 Aprendizagem Supervisionada

Aprendizagem supervisionada, ou na sua designação comum, *Supervised Learning*, é uma subcategoria de *Machine Learning* definida pelo uso de *datasets* previamente classificados no processo de treino de algoritmos de *Machine Learning* para que subsequentemente sejam capazes de classificar novos dados eficazmente [78]. À medida que são inseridos dados no modelo, são ajustados os seus pesos de modo assegurar que o modelo está corretamente ajustado [78].

Utilizando, por exemplo, o caso do filtro de *spam*, um caso de *Supervised Learning* existiria caso o treino do modelo fosse efetuado com um conjunto de emails já classificados com a *label* "*spam*" ou "*não-spam*", classificando posteriormente cada caso de teste com uma das *labels* anteriores que fosse considerada mais adequada pelo processo de inferência.

Os algoritmos podem ser divididos também quanto ao tipo de inferência, podendo ser de classificação ou regressão. Classificação atribui a cada objeto classes discretas, ou seja, o output produzido pelo algoritmo é uma categoria, por exemplo, num algoritmo para classificação com imagens de cães e gatos, os outputs possíveis seriam "cão" e "gato". Regressão atribui a cada objeto classes contínuas, por exemplo, num algoritmo de previsão de preços de artigos um output possível seria o número decimal "35,3" [37].

No conjunto de algoritmos supervisionados encontram-se algoritmos como, regressão linear, árvores de decisão e *support vector machines* [37].

2.5.1.1 Árvores de Decisão e Random Forest

Árvores de decisão são um algoritmo de previsão supervisionada de *Machine Learning* que pode ser usado para classificação ou regressão e representam um modelo hierárquico de decisões possíveis e as respetivas consequências. São frequentemente usadas em áreas como finanças, marketing, engenharia

e medicina [6].

Estas árvores funcionam de forma a que dado um certo *input*, cada nó da árvore será responsável por representar decisões que conduzirão à previsão final, que está presente nas folhas da árvore.

O processo de construção de uma árvore de decisão pode ser *Top Down* ou *Bottom Up*, sendo que o primeiro constrói a árvore desde a raiz até às folhas, e o segundo parte das folhas, ou seja, da classificação de cada objeto, para induzir a estrutura da árvore. [75]

O algoritmo *Random Forest* é uma das formas utilizadas para mitigar as questões de *overfitting* apresentadas pelas árvores de decisão. O algoritmo de *Random Forest* é formado por um conjunto de Árvores de Decisão que são treinadas cada uma em um subconjunto aleatório do conjunto de formação. Desta forma, o modelo faz previsões baseadas em várias árvores de decisão, o que aumenta a *performance* de classificação em relação às árvores de decisão.

2.5.1.2 Regressão Logística e Linear

Regressão linear e regressão logística são dois algoritmos de aprendizagem supervisionada, sendo que o primeiro é um algoritmo de classificação, ou seja, produz um *output* discreto, e o segundo é um algoritmo de regressão, ou seja, produz um *output* contínuo. O algoritmo de regressão linear efetua uma ponderação matemática dos valores dos atributos no *dataset* de treino, definindo os pesos das variáveis que são capazes de produzir melhores resultados de previsão.

O algoritmo de regressão logística é semelhante ao de regressão linear em todo o processo de cálculo dos pesos, diferindo apenas na apresentação do *output*, em que são definidos valores que limitem cada classe possível com base no seu valor numérico, sendo a classificação de cada objeto dada através da comparação do valor produzido na regressão linear com os limites definidos [20].

2.5.2 Aprendizagem não Supervisionada

Aprendizagem não supervisionada, ou na sua designação comum, *Unsupervised Learning*, é uma subcategoria de *Machine Learning* definida pelo uso de *datasets* que não foram previamente classificados, descobrindo padrões escondidos ou agrupamentos de dados sem necessidade de intervenção humana, graças à sua habilidade de descobrir diferenças e similaridades em informação [36].

Este tipo de aprendizagem pode conter problemas pertencentes a duas categorias, *clustering* e associação. Problemas de *clustering* são problemas em que o objetivo é encontrar agrupamentos inerentes aos dados, através de similaridades entre os objetos, por exemplo, agrupar clientes de uma loja com base nas compras efetuadas por eles. Por outro lado, problemas de associação são problemas em que se tenta encontrar correlações em grandes quantidades de dados, por exemplo, descobrir que quando um cliente C compra um produto A, existe uma grande tendência que as suas compras também incluam um produto B[37].

No conjunto de algoritmos não supervisionados encontram-se algoritmos como *k-Means-Clustering* e o algoritmo *a priori* para regras de associação [37].

2.5.3 Deep Learning

Apesar de ser uma ramificação de *Machine Learning*, *Deep Learning* difere das restantes áreas de *Machine Learning* nos dados que usa, e na forma como os usa para aprender. Ao contrário dos algoritmos de classificação que utilizam dados estruturados e classificados para aprender, *Deep Learning* utiliza dados não estruturados nem classificados na sua aprendizagem. A aprendizagem é feita a partir de redes neuronais multicamada que à medida que vão iterando sobre os dados de treino, tornam-se mais eficazes na tarefa de classificação executada.

Estas redes foram criadas com o intuito de emular a forma como os cientistas pensam que o cérebro humano funciona durante a aprendizagem, o algoritmo processa e reprocessa os dados, refinando gradualmente a análise e os resultados de modo a conseguir classificar eficazmente as classes dos objetos. As camadas de redes neuronais consistem em nós conectados entre si, em que cada um usa progressivamente um algoritmo mais complexo para extrair e identificar atributos e padrões nos dados, calculando por fim a confiança do algoritmo na classificação que foi dada ao objeto [68].

No conjunto de algoritmos de *Deep Learning* encontram-se algoritmos como *convolutional neural networks* que são utilizados para classificação de imagens em visão por computador, e *recurrent neural networks* que são utilizados para modelos em que os atributos e padrões alterem ao longo do tempo, das quais fazem parte as redes LSTM e GRU.

2.5.3.1 LSTM

As redes LSTM são um tipo de *recurrent neural networks* capazes de aprender dependências nos dados a longo prazo. Estas redes surgiram como forma de resolver o problema do *vanishing gradient* apresentado pelas RNN, que consiste em que, quando o gradiente passa por muitos *time steps*, tende a desaparecer, o que leva a que as RNN tenham muita dificuldade em conseguir aprender dependências nos dados a longo prazo [64].

Estas redes divergem das RNN devido ao diferente tipo de células neuronais utilizadas, estas células têm três portas diferentes denominadas de *forget*, *input* e *output*[73]. Cada porta contém ativações *sigmoid* que colocam os dados entre 0 e 1. A porta *forget* decide a informação que é retida ou esquecida, a porta *input* decide quais os valores que vão ser atualizados e a porta *output* decide qual o *output* a ser gerado a partir do estado interno da célula.

Graças a estas particularidades que lhes conferem capacidades de aprender dependências a longo prazo, as suas utilizações no mundo real estendem-se a áreas como análise de textos e de discursos[73].

2.5.3.2 GRU

GRU é um tipo especial de rede neural recorrente criada com o objetivo de capturar dependências de diferentes escalas temporais[38]. A estrutura interna de uma rede GRU é semelhante à estrutura interna de uma rede LSTM, exceto que a GRU combina a porta de entrada e a porta de esquecimento da LSTM numa única porta de atualização[73].

Embora sejam um tipo de RNN, as redes GRU apresentam ainda assim a mesma capacidade apresentada pela LSTM de responder ao problema do *vanishing gradient* apresentado pelas RNN[64].

Cada GRU contém uma porta de *update* que controla até que ponto a informação do estado do momento anterior é retida no estado atual, enquanto a porta de *reset* determina se o estado atual deve ser combinado com a informação anterior. A porta de *update* age de forma semelhantes às portas *forget* e *input* da estrutura LSTM, definindo qual a informação a descartar e qual a informação nova a adicionar. A porta de *reset* define a quantidade de informação passada que deve ser esquecida[73].

2.6 Grafos de Conhecimento

Uma definição dada a grafos de conhecimento é que são uma base de conhecimento estruturada sob forma de grafo que armazenam informação factual na forma de relacionamento entre entidades [17]. A teoria dos grafos de conhecimento é um novo ponto de vista usado na área de conhecimento da linguagem humana e traz novas vantagens nesta área, nomeadamente, uma maior capacidade de expressão, a capacidade de representar camadas semânticas mais profundas e de utilizar um conjunto mínimo de relações para imitar o processo de aprendizagem humano. O seu aparecimento deu uma nova forma à investigação de compreensão informática da linguagem humana [3].

Este tipo de grafos está a ganhar cada vez mais popularidade entre grandes empresas mundiais, e em diferentes comunidades científicas como Redes Semânticas, Bases de Dados, *Machine Learning* e *Data Mining* [34]. Entre as empresas que já anunciaram a utilização de grafos de conhecimento estão Airbnb, Amazon, eBay, Facebook, IBM, LinkedIn, Microsoft, Uber, entre outras [53].

A análise de grafos de conhecimento consiste na aplicação de processos analíticos aos grafos, permitindo extrair informação e tirar conclusões sobre os objetos presentes nos nós do grafo e relações entre eles, com base na topologia do grafo. A análise de grafos utiliza muitas técnicas relacionadas com outras áreas de conhecimento onde também são utilizados grafos, tais como redes sociais, a *web*, o *routing* de *internet* e redes de transportes.

Atualmente, existem diversas técnicas de análise de grafos que permitem definir a importância de cada nó com base em contextos de aplicação específicos, entre os quais estão *degree centrality*, *betweenness centrality*, *closeness centrality*, *Eigenvector* e *PageRank* [26].

2.6.1 Degree centrality

Degree centrality é definido pelo número de arestas incidentes num nó de um grafo. Se o grafo foi direcionado, *degree centrality* divide-se em *indegree centrality* e *outdegree centrality*. *Indegree* é o número de arestas do grafo com direção a um nó. *Outdegree* é o número de arestas com saída de um nó.

No caso do grafo ser direcionado, o *degree centrality* é calculado pela soma do valor de *indegree centrality* e *outdegree centrality* [11].

2.6.2 Closeness centrality

Closeness centrality é a métrica de análise de grafos que verifica a centralidade de um nó através do inverso da soma da sua distância aos restantes nós do grafo. O nó em que a soma das distâncias for menor, é o nó com maior centralidade segundo este critério. Esta métrica é útil para comparar a rapidez com que a informação flui no grafo a partir dos diferentes nós do grafo [48].

2.6.3 Eigenvector centrality

Eigenvector centrality é uma métrica de análise de grafos que analisa o nível de influência do nó na rede. Esta métrica atribui a cada nó da rede uma pontuação, que quanto maior, significa que o nó tem maior influência na rede. A pontuação é calculada através do número de conexões que o nó tem com outros nós, sendo que conexões com nós de alta pontuação leva a um maior aumento da pontuação do que conexões com nós de baixa pontuação. Esta métrica difere da *degree centrality*, porque um nó com menos ligações a outros nós, mas ligado a nós de maior pontuação, pode ter uma maior centralidade. Difere também da *Betweenness Centrality* porque um nó de alta *Betweenness Centrality*, ou seja, que ligue várias zonas do grafo, pode mesmo assim estar mais longe dos nós de maior influência no grafo, tendo uma *EigenVector Centrality* mais baixa do que um nó com menor *Betweenness centrality*, mas que esteja perto de nós muito influentes [44].

2.6.4 PageRank

PageRank é um algoritmo criado pela *Google* para definir a ordem de importância das páginas no motor de busca, mas que pode ser aplicado a outros grafos fora desta área. É uma variante do algoritmo *EigenVector Centrality*, usada para grafos direcionados.

O funcionamento deste algoritmo tem por base os passos seguintes[4]:

1. Inicialmente, são recolhidos todos os *websites* resultantes de uma pesquisa no motor de busca
2. É criado um grafo, em que os nós são os *websites* obtidos no passo 1, e as arestas são as hiperligações que outros *websites* têm para esse *website*. (e.g. se um *website* A tiver uma hiperligação para outro *website* B, esta relação é disposta no grafo por A->B)
3. Para cada nó representativo de um *website* obtido no passo 1, são contabilizados quantos são os outros nós que têm arestas a apontar para eles
4. Recursivamente, é realizado o passo anterior, mas para os nós que foram contabilizados, até que algum nó não tenha arestas de entrada
5. Para cada nó, é atribuída uma pontuação que é maior quanto maior for a pontuação e a quantidade de outros nós com arestas a apontar para si

6. Cada *website* é ranqueado por ordem decrescente da pontuação obtida pelo nó que o representa no grafo

No fim do algoritmo descrito anteriormente é possível avaliar a influência relativa dos nós no grafo, como vemos na Figura 2.5, em que um maior tamanho dos nós significa uma maior pontuação de PageRank e consequentemente maior influência no grafo.

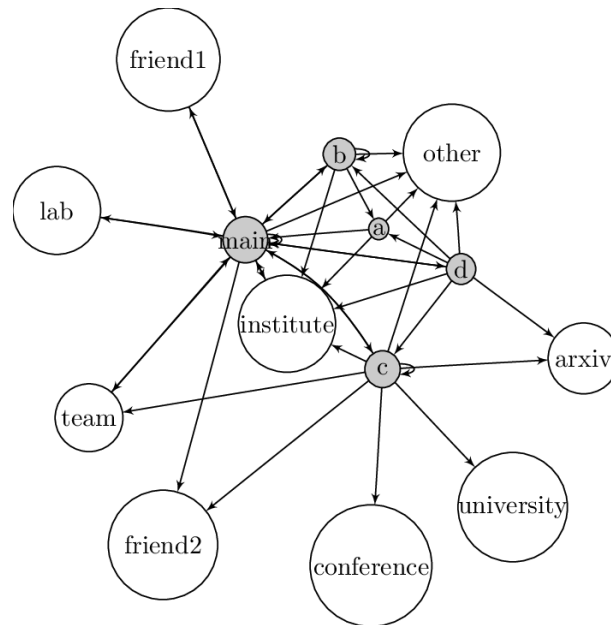


Figura 2.5: Demonstração visual do resultado da aplicação do algoritmo de PageRank num grafo [7]

2.7 Conclusão

Em suma, neste capítulo foram analisadas diversas técnicas e áreas de conhecimento relacionadas com o objetivo deste trabalho, o que permitiu comparar e aprofundar conhecimentos acerca das técnicas, de modo a formular um *pipeline* capaz de responder às questões de investigação propostas.

Partindo do conhecimento mais geral para o mais particular desta área, foi analisado inicialmente o processo de Extração de Conhecimento, processo este que representa a base de todos os projetos semelhantes estudados. Durante o aprofundamento sobre artigos acerca deste processo foi possível concluir a preponderância que tem a fase de pré-processamento de dados para a qualidade dos resultados finais do processo de extração de conhecimento.

De modo a poder alimentar o processo de extração de conhecimento com dados, foi necessário encontrar técnicas que permitissem extrair os dados de um *website* de forma estruturada. Com este propósito foi encontrada unicamente a técnica de *web scraping*, que engloba as técnicas de *crawling* e *parsing* e permite extrair e armazenar dados em diferentes formatos.

Por fim foram analisadas diversas técnicas para a fase de *data mining*, todas com diferentes propósitos, mas com o objetivo comum de encontrar padrões nos dados de forma a poder retirar conclusões sobre os dados que seriam dificilmente obtidas por análise humana dos dados sem recurso a técnicas de Inteligência Artificial.

Uma dessas técnicas foi *rule based system*, que é composta por um conjunto de regras que ao ser aplicado aos dados, permitem classificá-los de forma diferente com base na resposta às condições das regras. Esta técnica tem uma vantagem na área da contratação pública, devido à maioria dos países terem um conjunto de regras definidas para cada contrato, e que podem ser transformadas num *rule based system*.

Outra técnica estudada como forma de detetar padrões, foi a análise de grafos de conhecimento, que permite dispor os dados em forma de grafo e posteriormente aplicar diversas funções para análise do grafo e detetar relacionamentos entre entidades. Esta técnica tem a vantagem no contexto da contratação pública devido a permitir representar e analisar de forma mais compreensível os relacionamentos entre as entidades celebradoras dos contratos.

Foram também aprofundados os conceitos da área de *Machine Learning*, estudando os diferentes paradigmas e algoritmos, dos quais se destacaram *Random Forest* que conta com diversas utilizações no campo da deteção de irregularidades em áreas como a fraude de cartões de crédito.

Estado da Arte

Na secção inicial deste capítulo serão explicados vários conceitos relacionados com o campo da contratação pública.

Inicialmente serão abordados os tipos de contratos e procedimentos e as condições para a aplicação de cada um deles. De seguida serão abordados os problemas causados pela existência de irregularidades e conluio na contratação pública e algumas regras de prevenção e deteção de conluio publicadas pela AdC (Autoridade da Concorrência).

Por fim, serão abordadas utilizações práticas de técnicas de Inteligência Artificial relacionadas com os temas deste trabalho propostos nas questões de investigação, nomeadamente, abordagens gerais de *Machine Learning* na Contratação Pública, utilizações de sistemas de suporte à decisão em contratação pública, deteção de fraudes em redes sociais e *time series forecasting*. Para cada artigo analisado serão abordadas as dificuldades dos autores, as ferramentas utilizadas, os resultados obtidos e as conclusões retiradas.

3.1 Contratação Pública

Por definição contratação pública é o processo de compra de bens ou serviços, feita por entidades governamentais.

De modo a orientar este procedimento, existe o CCP (Código dos Contratos Públicos), onde estão contidas uma série de normas e procedimentos que devem ser obrigatoriamente cumpridos na feitura de cada contrato, tais como as obrigações das partes envolvidas no contrato, o próprio cumprimento do contrato, o prazo de execução, as legalidades por parte do adjudicatário, entre outros fatores importantes para a celebração do contrato.

Segundo o artigo 6 do CCP estas normas devem ser seguidas para todas as contratações que tenham por objetivo as prestações enumeradas de seguida[42] [72]:

- Empreitada de obras públicas
- Concessão de obras públicas
- Concessão de serviços públicos
- Locação ou aquisição de bens móveis
- Aquisição de serviços, e outros contratos submetidos à concorrência;

Os contratos podem ser classificados quanto ao procedimento realizado na sua feitura, entre os quais se destacam o ajuste direto, concurso público e consulta prévia.

De acordo com o n.º 2 do artigo 112.º do CCP, o ajuste direto é o procedimento em que a entidade adjudicante convida diretamente uma entidade, à sua escolha, a apresentar proposta. De salientar que existe também o ajuste direto simplificado que se trata de um procedimento de ajuste direto que dispensa quaisquer formalidades procedimentais consumando-se quando o órgão competente para a decisão de contratar aprova a fatura ou documento equivalente apresentada pela entidade convidada, comprovativa da aquisição.

Concurso público é o procedimento padrão de contratação do estado, e é realizado através da abertura de um concurso para a realização do serviço ou aquisição de bens em questão, em que a empresa que apresentar a proposta adequada aos parâmetros propostos de valor mais baixo é selecionada. Este tipo de procedimento pode ser adotado sempre que a entidade adjudicante o entender, no entanto, quando o valor do contrato a celebrar for superior aos limiares europeus de contratação pública, o anúncio deve ser obrigatoriamente publicado no Diário da República e no Jornal Oficial da União Europeia.

A consulta prévia é o procedimento em que a entidade adjudicante convida diretamente pelo menos três entidades à sua escolha a apresentar proposta, podendo com elas negociar os aspetos da execução do contrato a celebrar, desde que tal possibilidade conste expressamente do convite [42].

Cada um dos procedimentos exceto o concurso público tem diversos critérios que necessita cumprir para que o contrato seja válido, nomeadamente um preço máximo previamente definido para cada tipo de serviço/aquisição e que varia se o adjudicante for o Estado ou outra entidade [72].

Uma vez concluídos, todos os contratos são obrigatoriamente depositados no Portal Base.Gov, onde são discriminados todos os atributos relevantes do contrato, entre os quais o tipo de procedimento, descrição, tipo de serviço prestado, entidade adjudicante, entidade adjudicatária, preço contratual, data de execução, empresas concorrentes, empresas convidadas e cópia do contrato.

3.1.1 Detecção de Irregularidades em contratos públicos

Irregularidades em contratos públicos podem ser entendidas como todas as infrações ao conjunto de regras definidas para esses contratos, independentemente da fase em que o contrato se encontra e da existência de consequências acarretadas por essa irregularidade.

O elevado número de contratos públicos celebrados todos os anos, alguns deles com diversos concorrentes, faz com que o número de propostas para análise seja enorme, isto aliado à grande complexidade do sistema de regras a que todos os contratos se devem sujeitar, torna muito complexa a análise manual de propostas de modo a verificar a sua regularidade, tornando propícia a existência de irregularidades provenientes de lapsos durante a verificação dos contratos por intervenientes humanos.

Essas irregularidades podem ter um grande impacto no resultado do procedimento em questão, por exemplo, nos casos de erros que se fossem detetados a tempo alterariam o vencedor do concurso. Podem também ser irrelevantes para o concurso, por exemplo, nos casos em que mesmo com o erro em questão o vencedor permaneceria inalterado [77].

A complexidade associada à feitura de contratos é uma das causas para a existência de irregularidades nos contratos, segundo um artigo publicado no *World Economic Forum*, apenas 4,5% dos concorrentes a contratos públicos consideram fácil o procedimento de concurso, e 39,4% dos concorrentes considera extremamente difícil. No caso das *startups* apenas 6,1% acha simples o procedimento de trabalhar com o governo [47].

3.1.2 Conluio na Contratação Pública

Segundo a AdC o conluio na contratação pública consiste na concertação de propostas com o objetivo de eliminar ou limitar a concorrência nos procedimentos de contratação[33]. Comportamento esse que leva a condições menos favoráveis para o Estado do que as que resultariam de uma situação de concorrência efetiva, traduzindo-se em preços mais elevados, qualidade inferior, menos inovação, e numa perda de cerca 18 mil milhões de euros dos cofres do estado português anualmente[45].

Em 2015, por exemplo, foram multadas pela AdC cinco empresas portuguesas que confessaram ter efetuado um acordo entre si que visava a repartição e fixação de preços no mercado de fornecimento e montagem de módulos pré-fabricados para a instalação provisória de salas de aula entre 2009 e 2010 prejudicando assim o estado português em avultadas quantias monetárias[74].

Entre as estratégias de conluio mais recorrentes, destacam-se:

- Propostas Rotativas: Esquemas de rotatividade da proposta vencedora, em que duas ou mais empresas combinam ganhar entre si alternadamente os concursos.
- Propostas de Cobertura: Para criar uma ilusão de concorrência, as empresas combinam submeter propostas com um preço mais elevado que o da empresa que escolheram previamente para vencer o procedimento, para que o contrato lhe seja adjudicado.
- Supressão de Propostas: As empresas acordam não submeter ou retirar uma proposta, para que o contrato seja adjudicado à empresa que escolheram para vencer o procedimento.
- Repartição de Mercado: As empresas combinam um esquema de apresentação de propostas com o objetivo de repartir o mercado entre si.

- Subcontratação: As empresas acordam facilitar o sucesso da proposta da empresa que escolhem para vencer o procedimento, em contrapartida, a empresa vencedora garante à empresa que perde a subcontratação de fornecimentos no âmbito do contrato.

De modo a evitar o conluio, existem diversas abordagens expressas pela AdC que podem ser utilizadas para diminuir a possibilidade da sua ocorrência, através de indicadores que tornam um concurso mais suspeito, entre as quais estão [33]:

- Reduzido número de Empresas a concorrer num mercado.
- Condições do mercado estáveis e previsíveis.
- Empresas a concorrer em diversos mercados.
- Associações empresariais.
- Homogeneidade dos produtos ou serviços contratualizados.

3.2 *Machine Learning* na Contratação Pública

Atualmente, existe uma vasta bibliografia de aplicações de técnicas de *Machine Learning* para extração de conhecimento e aprendizagem de padrões para prever diversos acontecimentos, estas aplicações podem, por exemplo, ser encontradas em abordagens como identificação dos padrões de influência de stress nos comportamentos negociais[13] ou classificação dos níveis de confiança entre oponentes em cenários de negociação[62].

Por outro lado, são escassas as aplicações práticas deste tipo de abordagens no campo da contratação pública existentes na literatura científica. Apesar disso, serão abordadas nesta secção algumas utilizações existentes. Para cada artigo abordado serão explicados o objetivo do trabalho, os métodos utilizados e os resultados obtidos.

No caso da contratação pública podemos encontrar aplicações desta área de conhecimento no âmbito do preenchimento automático de propostas.

Por exemplo, numa das abordagens presentes num artigo da empresa multinacional Deloitte[79], acerca das tecnologias emergentes no setor da contratação públicas, foram utilizadas técnicas de *Machine Learning* de modo a conseguir prever o código do CPV de contratos públicos ucranianos. Esta abordagem foi capaz de prever com 70% de acerto os primeiros 4 algarismos do CPV dos contratos de teste. Para isto foi utilizado um método de abordagem supervisionada, ou seja, contratos de treino previamente com o CPV previamente classificado, e é feita uma inferência com base em atributos textuais dos contratos, dos quais são ressaltados no artigo a descrição e o título dos contratos.

Outro estudo de caso presente nesse mesmo artigo da Deloitte, refere a criação de uma aplicação em Python que classifica automaticamente os contratos públicos. Esta ferramenta, de nome *CAITY*, foi treinada com 45 milhões de contratos previamente classificados. Os atributos do contrato utilizados no

treino foram o nome, classificação e número identificador da entidade Adjudicatária, e a descrição do produto contratado. Esta aplicação conseguiu superar em 20% a eficácia de acerto que era obtida pelo processo de classificação manual que é usado tradicionalmente, tendo conseguido cerca de 90% de eficácia na classificação dos contratos. Foi então concluído pelos autores do artigo que a ferramenta *CAITY* permite não só aumentar a eficácia do processo de categorização de contratos, bem como diminuir o tempo despendido pelo staff nesse processo.

Por outro lado, em Sales[10] foram utilizadas técnicas de *credit scoring*, técnicas comumente utilizadas para analisar o risco de empréstimos em instituições financeiras, para prever a probabilidade de um adjudicatário de um concurso não cumprir as obrigações contratuais expressas no concurso celebrado. Para tal foi utilizado um *dataset* previamente classificado, com 1000 empresas consideradas "responsáveis" e outras 1000 consideradas "defraudadoras". O critério de seleção dos contratos do *dataset* foi o seguinte, para encontrar contratos de entidades "defraudadoras" foram recolhidos apenas contratos de empresas proibidas de voltar a concorrer em concursos públicos no Brasil porque constam no CEIS (Cadastro de Empresas Inidóneas e Suspensas). Para obter as empresas "responsáveis", foram recolhidos contratos de empresas cujo nome não conste no CEIS e que tenham no mínimo 5 contratos concluídos com o governo brasileiro nos últimos 5 anos.

Os algoritmos de previsão escolhidos foram árvores de decisão e regressão logística. O primeiro foi escolhido por poder computar variáveis numéricas e categóricas sem afetar a *performance* de previsão e por ter a vantagem de permitir dispor graficamente os caminhos para as decisões tomadas. O segundo foi escolhido por ser usualmente utilizado em *credit scoring*.

Por fim foram avaliados ambos os modelos, e obtida uma eficácia de 62,5% para a regressão logística e 63% para a árvore de decisão.

3.3 Sistemas de Suporte à Decisão na Contratação Pública

A implementação de sistemas de suporte à decisão em contratação pública pode ser um passo importante para influenciar positivamente a qualidade das tomadas de decisão efetuadas. Estes sistemas têm a capacidade de facilitar os processos de contratação providenciando aos funcionários responsáveis pelas tomadas de decisão informações obtidas através da aplicação de técnicas de *data mining* aos contratos existentes.

Em Carpanese et al.[29] são discutidas as possibilidades de implementação de um sistema de suporte à decisão utilizando algoritmos de deteção de conluio como fator de auxílio ao utilizador de modo a encontrar potenciais contratos fraudulentos. Este artigo é baseado em um *dataset* de contratação pública do Brasil, um dos países mais afetados pela corrupção a nível mundial segundo o *Transparency International*[41]. Utiliza como principais variáveis de previsão de conluio as moradas das empresas e os valores das propostas dos concorrentes vencedores e derrotados. Sendo que moradas partilhadas por duas ou mais empresas participantes do mesmo concurso é considerado um indicio de conluio. No caso dos valores das propostas, é analisada a similaridade das propostas derrotadas, por exemplo, caso duas

ou mais empresas apresentarem propostas com exatamente os mesmos valores num concurso, será considerado como um indício de conluio.

É também utilizado neste artigo como possível indicador de conluio a presença de um *Top Loser* num concurso, que consiste numa empresa com um rácio de Vitórias/Concursos Totais inferior a 5% tendo participado em pelo menos 15 concursos, ou de uma empresa *Top Winner*, que neste caso é o oposto da anterior, ou seja, uma empresa com um rácio de vitórias muito elevado.

3.4 Análise de Redes Sociais

Ao longo dos anos, têm surgido diversas aplicações de técnicas de Inteligência Artificial em redes sociais em áreas tão díspares como resolução de conflitos[8], prevenção de esquemas de lavagem de dinheiro[39] ou deteção de discurso de ódio em plataformas de *social media*[76]. Uma das aplicações mais comuns para a análise de redes sociais, e que será abordada nos parágrafos seguintes é a de deteção de fraudes, nestas abordagens são analisadas as ligações entre diferentes entidades de um grafo de modo a tentar encontrar indícios de fraude a partir de técnicas como *queries* ou algoritmos. Em Srivastava et al.[23] foi utilizada análise de redes sociais, com o objetivo de analisar os círculos fraudulentos entre entidades num grafo criado a partir da fuga de informação associada ao escândalo *Panama papers*, contendo cerca de 11 milhões de documentos sobre mais de 250 mil entidades, documentos que incluem informação suspeita acerca de inúmeros indivíduos abastados de diferentes países.

O *dataset* utilizado neste trabalho é um ficheiro CSV que contém emails, contactos, documentos digitalizados e documentos transcritos. Para aglomerar esta informação a estrutura escolhida foi um grafo, escolha que foi justificada pelo formato desta estrutura permitir aglomerar as relações entre as entidades presentes nos vértices, e também por permitir efetuar *queries* complexas de forma eficiente e em tempo constante e independente do tamanho do *dataset*[23].

Para isto foi utilizado Neo4j para o desenho do grafo de conhecimento. Neo4j é uma plataforma utilizada para mapear, analisar, guardar e percorrer informação conectada em formato de grafo com *performance* superior em relação às restantes ferramentas de bases de dados relacionais[23].

Foram utilizados três diferentes algoritmos de grafos de modo a analisar e determinar a importância que cada entidade envolvida tinha no escândalo dos "*Panama Papers*", nomeadamente, *Page Rank*, *Betweenness Centrality* e *Closeness Centrality*.

Foram aplicadas *Cypher Queries*, que são *queries* que permitem consultas de dados expressivas e eficientes em grafos de conhecimento Neo4j. Com isto foi possível descobrir dados relevantes no contexto do problema, como o número de oficiais financeiros que foram intermediários e realizaram assessoria financeira a entidades de forma não autorizada. Foram também obtidos os círculos fraudulentos de entidades fraudulentas que estão relacionadas.

Noutra abordagem foram utilizadas redes sociais e *Machine Learning* de modo a prever fraudes em transações de cartões de crédito [50].

A implementação passou por, criação de um grafo em Neo4j a partir do *dataset* e uso das *Cypher queries* para obter os círculos sociais potencialmente fraudulentos (e.g círculo de clientes com o mesmo email), e posterior uso da linguagem Python para aplicação de técnicas de *sampling* (*Oversampling*, *Undersampling*, *Hybrid*, *SMOTE* e *ROSE*) e de algoritmos de previsão (*Deep Learning*, *Random Forest*, *GLM*, *GBM* e *Ensemble Learning*), por fim foi utilizada a biblioteca *scikit-learn* para implementação dos algoritmos e avaliação dos resultados obtidos.

Em conclusão apesar de o *dataset* estar altamente desbalanceado tal como é referido no artigo, ainda assim o algoritmo de *Deep Learning* consegue ter a maior precisão entre todos, o que significa que foi o que teve o maior número de fraudes classificadas corretamente, apesar de no que toca à eficácia ter ficado um pouco abaixo das abordagens de previsão com recurso a *sampling*, como era expectável devido ao desbalanceamento do *dataset*.

3.5 Time series forecasting

A previsão de séries temporais é a designação dada à tarefa de prever os valores futuros de uma dada sequência utilizando dados históricos utilizando técnicas de previsão que inferem a dependência estocástica entre valores passados e futuros. Recentemente, esta área de conhecimento tem atraído a atenção dos investigadores na área de *Machine Learning* como forma de ultrapassar as limitações dos métodos tradicionais de previsão, que são de complexidade muito elevada e necessitam de um dispêndio elevado de tempo.

Na literatura disponível no âmbito de *Time series forecasting*, uma grande parte está associada ao uso desta tarefa no âmbito da previsão de preço de ações da bolsa, como é o caso de Sethia et al.[22] onde foram utilizados LSTM, GRU, SVM e MLP como forma de prever o valor do *Standard's and Poor's 500 Index*, um índice composto por quinhentos ativos cotados nas bolsas. Para tal, foi utilizado um *dataset* composto pelos preços deste índice entre os anos 2000 e 2017 de modo que os algoritmos pudessem aprender a partir dos preços antigos. Por fim foi concluído que os modelos que utilizavam GRU e LSTM foram superiores aos restantes na capacidade de previsão, sendo que apesar de o modelo de GRU ter sido capaz de prever melhor *trends* de variação de preços, o modelo de LSTM teve uma melhor certeza de previsão. Apesar dos bons resultados destes dois modelos, nenhum dos modelos utilizados foi capaz de prever períodos de queda ou subida acentuada de preços.

3.6 Conclusão

Em suma, o processo contratação pública envolve a compra de bens e serviços e é guiado pelo código dos contratos públicos. Este código contém um conjunto de regras que ditam os limites de preço para cada tipo de contrato formulado. A partir desse código, existem contratos que estão em incumprimento das regras do contrato e são denominados de irregulares. Para além das irregularidades, também a existência

de conluio é um fator prejudicial à qualidade da contratação pública, e tem diversas regras de auxílio à sua detecção e prevenção publicadas pela Autoridade da Concorrência.

De modo a detetar irregularidades em contratos, foi explorada a aplicabilidade de *Machine Learning* neste campo, visto que poderia aprender a identificar contratos irregulares mais facilmente. Como tal foram analisadas diferentes algoritmos para este propósito em artigos da comunidade científica, como por exemplo, em Sales[10], onde foram experimentados alguns algoritmos dos quais se destacaram árvores de decisão.

Outra abordagem com destaque na busca de irregularidades em *datasets* que envolvem relacionamentos entre entidades é a análise de redes sociais. Estas redes em forma de grafo podem ser percorridas com uma *performance* superior às bases de dados relacionais, e permitem a aplicação de algoritmos capazes de encontrar não só comunidades como também os nós mais influentes do grafo.

Por outro lado, foram também analisadas técnicas utilizadas para previsão de valores futuros de conjuntos de dados compostos por sequências numéricas. Neste tema, é recorrente a utilização de LSTM e GRU com *performances* de classificação elevadas em relação às técnicas restantes, como é o caso dos resultados obtidos em Sethia et al.[22] onde estas duas técnicas se destacaram em relação a SVM e MLP, no entanto, nenhuma delas é eficaz em prever quando os valores atingem picos.

Arquitetura da Solução e Tratamento de Dados

Neste capítulo será abordada a arquitetura da solução desenhada para o problema desta dissertação. Será inicialmente explicado o *pipeline* da abordagem utilizada para o problema, abordando resumidamente cada passo tomado na solução desenvolvida. De seguida serão abordadas as tecnologias utilizadas na solução do problema, explicitando para cada uma delas as suas vantagens, desvantagens, utilidades e particularidades.

Por fim será detalhada a fase de captura e pré-processamento de dados, terminando com a explicação acerca da fase de análise de dados, onde foi feito entendimento e tiradas conclusões acerca da distribuição dos dados recolhidos.

4.1 Pipeline da Abordagem

A abordagem deste projeto, representada na figura 4.1, está dividida em 3 fases distintas, aquisição e pré-processamento de dados, extração de conhecimento e criação de uma plataforma de visualização de dados do *dataset*.

Aquisição dos dados é a fase onde é foram extraídos os dados do Portal Base.gov através de *web scraping*, no caso da figura 4.1 a biblioteca utilizada para esse fim é a biblioteca de Python *Beautiful Soup*.

Antes de iniciar a fase de extração de conhecimento, e de modo a otimizar os resultados produzidos, foi necessária a aplicação de técnicas de pré-processamento de dados, a análise dos dados recolhidos, escolha da amostra do *dataset* a ser utilizada para os modelos, entre outras técnicas de pré-processamento consideradas necessárias no contexto do problema.

Posteriormente foi utilizada a biblioteca Networkx para criar grafos a partir da informação dos contratos extraídos, o que permite analisar relações entre as entidades que celebram os contratos.

A fase de extração de conhecimento foi subdividida em quatro subfases que serão descritas de seguida.

Inicialmente foram aplicados algoritmos de análise de grafos ao grafo de conhecimento criado na fase anterior de modo a guardar os contratos extraídos na fase de aquisição de dados, com o objetivo de tentar analisar relacionamentos entre empresas e a importância relativa (tanto ao nível de dinheiro recebido como pelo número de contratos que lhe foram atribuídos) de cada empresa dentro de uma certa localidade ou distrito, para isso foram utilizados algoritmos maioritariamente de centralidade e deteção de comunidades.

Posteriormente, no âmbito da deteção de conluio entre empresas intervenientes em contratos, foram aplicadas um conjunto de regras presentes no OCDS e outras definidas pela AdC para identificar contratos potencialmente colusivos. De seguida foi criado um sistema de regras a partir das regras do CCP que foi aplicado aos contratos do *dataset* de modo a sinalizar contratos irregulares, através das classificações geradas foram utilizados esses contratos classificados como *dataset* de treino para um modelo de *Machine Learning* de modo que ele aprenda a classificar contratos quanto à sua regularidade baseando-se num número reduzido de atributos. Por fim, foram também utilizadas técnicas de *Machine Learning* no âmbito de conseguir prever o valor de gastos futuros em cada setor de contratação pública através dos gastos nos meses anteriores, utilizando algoritmos de aprendizagem adequados a problemas de sequências temporais.

Por fim, foram criadas *dashboards* com gráficos de representação de dados do *dataset* que foram considerados como relevantes no âmbito da otimização e perceção dos fluxos de gastos com contratação pública em Portugal.

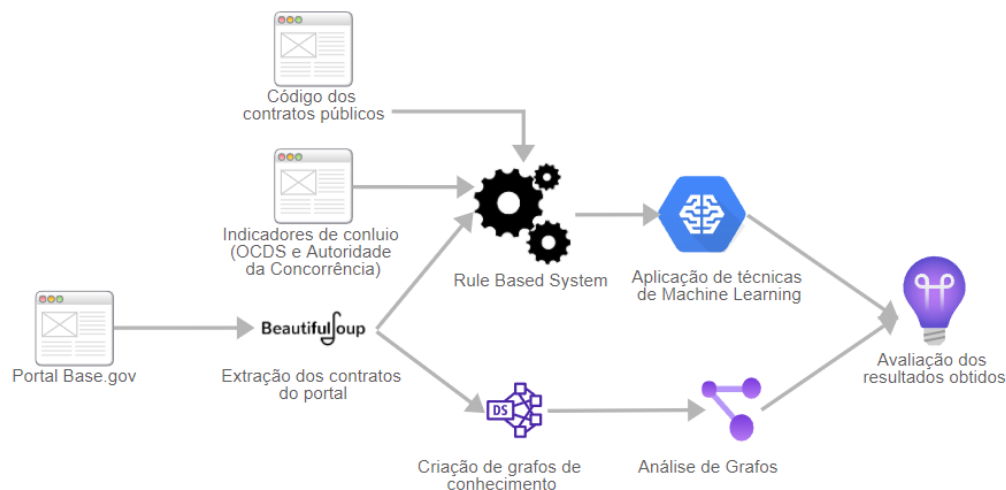


Figura 4.1: Representação da arquitetura do sistema

4.2 Tecnologias

Nas subsecções seguintes serão abordadas de uma forma resumida as particularidades de cada uma das bibliotecas da linguagem *Python* utilizadas na conceção deste trabalho.

4.2.1 *Beautiful Soup*

Beautiful Soup é uma biblioteca de Python utilizada para fazer *web scraping* de páginas *web*. Esta biblioteca permite extrair informação a partir de ficheiros HTML e XML. Tem como vantagens a sua fácil implementação e a sua habilidade de reconhecer automaticamente o *encoding* de textos HTML [63].

4.2.2 *NetworkX*

NetworkX é uma biblioteca da linguagem Python, que pode ser utilizada para implementação e análise dos grafos de conhecimento a partir do *dataset*. Esta biblioteca providencia estruturas de dados capazes de representar redes ou grafos, incluindo grafos orientados e grafos não orientados.

Os nós dos grafos criados em *NetworkX* têm a vantagem de poder conter qualquer objeto *hashable* de Python, e as arestas poderem conter informação arbitrária.

Para além das estruturas de dados, *NetworkX* conta também com diversos algoritmos para análise de grafos tais como *shortest paths*, *betweenness centrality*, *clustering* ou *degree distribution* [51].

4.2.3 *Scikit learn*

Scikit Learn foi a biblioteca de Python indicada para efetuar as previsões de irregularidades em contratos usando *Machine Learning*. É uma biblioteca *open-source* que providencia aos utilizadores várias implementações de diversos algoritmos de *Machine Learning*, tornando a utilização *Machine Learning* acessível a especialistas de diversas áreas de conhecimento (eg. biólogos, físicos, desenvolvedores de *software*) sem que tenham de ter o conhecimento necessário para perceber o funcionamento do algoritmo, mantendo um nível de dificuldade de implementação reduzido e diversas documentações e instruções de uso [9].

4.2.4 *Keras*

Keras é a biblioteca de Python que foi utilizada nos modelos de previsão de custos. É uma API de *deep learning* que corre em cima da plataforma de *deep learning* TensorFlow e foi criada com o objetivo de permitir experiências simples, flexíveis e poderosas de soluções de *deep learning*. Esta biblioteca é usada por empresas como a NASA e o Youtube[31].

4.3 Ferramentas

A experimentação prática presente neste capítulo foi realizada em Jupyter Notebook, uma aplicação da plataforma *open-source* Anaconda que permite a criação de documentos que contêm código e permitem a visualização dos respetivos *outputs* para cada porção de código, o que a torna mais transparente, perceptível, repetível e partilhável. A recolha do *dataset* final foi realizada na plataforma Kaggle, com recurso à funcionalidade de computação em nuvem, devido ao elevado esforço computacional exigido para recolher centenas de milhares de contratos. A linguagem escolhida foi Python devido à sua grande

utilização no contexto de problemas de *Data Science*, e devido às suas bibliotecas para análise e tratamento de dados como Pandas, Numpy e Matplotlib.

O ficheiro Jupyter foi corrido localmente numa máquina com as especificações seguintes: Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz e 16,0 GB de RAM.

4.4 Aquisição de Dados

Os dados necessários para o trabalho foram extraídos a partir do portal Base.Gov através da técnica de *Web Scraping*. Para tal, a linguagem escolhida foi Python, que conta com a biblioteca de *Web Scraping BeautifulSoup* que foi a escolhida para extrair dados a partir de páginas *Web*.

Inicialmente, foi feita uma análise à página do portal onde se encontram os contratos para ver de que forma era possível extrair automaticamente os dados de todos os contratos. Foi verificado que cada página de contratos do Portal contém uma tabela com os contratos, por ordem de introdução do mais recente para o mais antigo, e que cada linha da tabela tem uma hiperligação que leva a uma página com informação mais detalhada acerca do contrato selecionado, incluindo informações sobre os concorrentes, preço, data de execução, anexos, entre outros atributos.

Foram encontrados **1266884** resultados para a sua pesquisa.

Pesquisou por:

 Exportar Resultados (Excel)

Objeto do Contrato	Preço contratual	Publicação	Adjudicante	Adjudicatário	
3020024713/FORNECIMENTO DE PAINEIS DE FIBRA	270,00 €	26-01-2021	Ministério da Defesa Nacional - Marinha	LUSO GERAL LDA	+
Concurso Público para prestação contínua de serviços de manutenção (preventiva...	180.000,00 €	26-01-2021	Fagar - Faro, Gestão de Águas e Resíduos, E. M.	Emplilhadores de Portugal (Algarve) - Comércio de Máquinas, Lda	+
Aquisição de fatos de trabalho, produtos e equipamentos de higiene...	74.337,00 €	26-01-2021	ALTERNÂNCIA - Ensino e Formação Profissional, C. R. L.	Africorte	+
Expansão do Parque Empresarial de Ferreira do Alentejo (Obras de...	1.924.800,00 €	26-01-2021	Município de Ferreira do Alentejo	Tecnovia - Sociedade de Empreitadas, S.A.	+
Prestação de Serviços Jurídicos	26.000,00 €	26-01-2021	Ordem dos Médicos	Luís Filipe Carvalho Pereira	+
3020026891/FORNECIMENTO DE PUBLICAÇÕES	328,00 €	26-01-2021	Ministério da Defesa Nacional - Marinha	J. Garraio & Cª, Lda	+
Aquisição de serviços de apoio técnico na área do Turismo...	19.920,00 €	26-01-2021	Município de Viana do Alentejo	Joséla Adriana Silvestre Bruno	+

[Página Anterior](#) [Próxima Página](#)

Figura 4.2: Disposição da lista de contratos no portal Base.gov


Detalhe do Contrato		 Imprimir
Data de publicação no BASE	03-10-2014	
Tipo(s) de contrato	Locação de bens móveis	
Nº de registo do acordo quadro	614368	
Descrição do acordo quadro	PE. 13125, CAQ - Concurso Público Internacional para a Celebração de um Acordo Quadro para o Fornecimento e Montagem, em Regime de Aluguer, de Monoblocos Pré-Fabricados para a Instalação Provisória de Salas Para o Funcionamento de Atividades Letivas e de Serviços de Apoio nas Escolas do Programa de Modernização das Escolas do Ensino Secundário	
Tipo de procedimento	Ao abrigo de acordo-quadro (art.º 259.º)	
Descrição	Contrato de Fornecimento n.º 14/3177/CA/C	
Fundamentação	Artigo 259.º do Código dos Contratos Públicos	
Fundamentação da necessidade de recurso ao ajuste direto (se aplicável)	Não aplicável	
Entidade adjudicante - Nome, NIF	Parque Escolar, E. P. E. (508069645)	
Entidade adjudicatária - Nome, NIF	NORMETAL - UNIDADE DE ESTRUTURAS METÁLICAS, S.A. (502505729)	
Objeto do Contrato	PE. 14076, AAQ - FORNECIMENTO E MONTAGEM, EM REGIME DE ALUGUER, DE MONOBLOCOS PRÉ-FABRICADOS PARA A INSTALAÇÃO PROVISÓRIA DE SALAS PARA O FUNCIONAMENTO DE ATIVIDADES LETIVAS E DE SERVIÇOS DE APOIO NA ESCOLA SECUNDÁRIA QUINTA DO MARQUÊS EM OEIRAS (ZONÁ 1), AO ABRIGO DO ACORDO QUADRO N.º 17/2014, CELEBRADO COM A PARQUE ESCOLAR, E.P.E.	
Procedimento Centralizado	-	
CPV	44211100-3, Módulos pré-fabricados portáteis	
Data de celebração do contrato	02-10-2014	
Preço contratual	41.180,86 €	
Prazo de execução	517 dias	
Local de execução - País, Distrito, Concelho	Portugal, Lisboa, Oeiras	
Convidados	U.E.M. - UNIDADE DE ESTRUTURAS METÁLICAS, S.A. Algeco - Construções Pré-Fabricadas, S.A. Elevatrans, Lda GRUPO VENDAP, S.A.	
Concorrentes	U.E.M. UNIDADE DE ESTRUTURAS METÁLICAS, S.A.- NORMETAL (502505729) ELEVATRANS PRÉ-FABRICADOS, S.A. (504072811) Algeco - Construções Pré - Fabricadas S.A. (502721871)	

Figura 4.3: Disposição da informação de um contrato no portal Base.gov

A partir da disposição que podemos verificar na figura 4.2 e 4.3 os passos necessários para extrair os dados dos contratos são os seguintes:

1. Fazer o pedido *get* ao servidor para obter o HTML da página;
2. A partir do HTML total extrair apenas a tabela dos contratos dessa página;
3. Para cada entrada da tabela abrir a hiperligação associada;
4. Para cada hiperligação obter a tabela de dentro de cada contrato;
5. Para cada entrada da tabela guardar as informações referentes ao contrato;
6. Abrir o url da página seguinte caso exista;
7. Caso o passo anterior tenha encontrado uma página voltar para o passo 1, caso contrário é terminada a execução;

Um dos problemas que surgiu para implementar este algoritmo foi saber quando não existiriam mais contratos restantes para extrair, visto que o Portal nunca para de apresentar novos contratos. A tentativa inicial foi em cada página de contratos, após a extração de todos os contratos, fazer um pedido *GET* na hiperligação presente no botão de nome "Próxima Página". Esta tentativa foi falhada, pois, mesmo quando já não há mais contratos a mostrar, o botão "Próxima Página" continua a estar presente na página e a

mostrar contratos aleatórios. A solução encontrada consiste em recolher o número de contratos existentes, presente no topo da página, e ir alterando o campo *range* do URL até perfazer o número total de contratos.

Após a aplicação do *Web Scraper* para obter os dados dos contratos, foi necessária a utilização de filtros de texto para retirar as *tags HTML* dos dados e ficar apenas com os valores dos atributos.

Para tal foi utilizada a biblioteca RE de Python que permite utilizar expressões Regex para filtrar apenas caracteres de interesse em textos. Com esta biblioteca foram criados filtros para retirar cada uma das *tags HTML* presentes na tabela, foi também criado um filtro para dividir a localização do contrato em dois atributos separados, localidade e distrito, de modo a facilitar as pesquisas de contratos geograficamente, sendo que na tabela ela está presente apenas como um atributo triplo do tipo (Localidade, Distrito, País).

Para guardar os dados foi utilizada a biblioteca de Python de nome *Pandas*, que é utilizada para manipulação e análise de dados, desta biblioteca a estrutura utilizada para aglomerar os dados extraídos do portal foi um *Dataframe*, foi escolhida devido à sua forma tabular, com linhas e colunas, e à facilidade de selecionar, apagar ou substituir entradas. Na figura 4.4 está uma parte do *Dataframe* resultante de uma extração de dados e remoção das *tags HTML*, em que apesar de não ser totalmente visível, cada linha contém 25 atributos.

Por fim, de modo a melhorar a *performance* da recolha de dados, foi utilizada a biblioteca de multiprocessamento de Python. A decisão da utilização de multiprocessamento surgiu devido ao elevado tempo despendido para a recolha de dados numa abordagem sequencial, que persistiu mesmo utilizando um ambiente de execução em *cloud* com um elevado poder computacional como o Kaggle e o Google Colab. No sentido de averiguar a existência de uma possível diminuição no quesito temporal através da utilização de paralelismo na captura dos contratos, foram efetuadas várias capturas de diferentes volumes de dados e com diferentes números de processos. Inicialmente foram capturados 120 contratos, com 1, 15, 30, 40 e 60 processos em simultâneo, os resultados demonstraram uma superioridade significativa na velocidade de recolha de contratos utilizando multiprocessamento, diminuindo em cerca de 5 vezes o tempo demorado no tempo obtido com 40 processos em relação à abordagem sequencial. De seguida foi tomada a decisão de experimentar o mesmo procedimento para um valor superior de contratos para ver de que forma os valores evoluíam, e para verificar se os ganhos se mantinham. Como pode ser verificado na tabela, os ganhos de tempo na abordagem de paralelismo não só se mantiveram no caso dos 30 e 40 processos, como se acentuaram no caso da utilização de 60 processos, tendo este passado a ser aproximadamente 8 vezes mais rápido do que a abordagem sequencial. Devido a esta descoberta foi então decidido utilizar uma abordagem de paralelismo com 60 processos por página na recolha dos dados.

Número de Processos	Tempo de recolha (120 contratos)	Tempo de recolha (360 contratos)
60	18,90s	53,79s
40	17,69s	62,48s
30	19,81s	77,06s
15	43,02s	149,97s
1	104,77s	408,96s

Para conseguir tirar proveito do paralelismo durante a recolha de dados foi necessário recorrer a uma lista partilhada entre processos que serviu para aglomerar a informação dos contratos recolhidos por cada processo, algo que só foi possível através da classe *Manager*, pertencente ao *package multiprocessing*, que permite criar estruturas que podem ser acedidas por diferentes processos criados por *forks* de um processo inicial, visto que isto não é possível utilizando as estruturas criadas convencionalmente.

Após a criação da lista partilhada, é criado um processo para cada contrato presente nessa página, cada processo coloca as especificações recolhidas acerca do contrato na lista partilhada. Por fim faz uma espera para que todos os processos dessa página terminem, e posteriormente avança para a página seguinte repetindo o processo anteriormente descrito.

	Data de publicação no BASE	Tipo(s) de contrato	Tipo de procedimento	Descrição	Fundamentação	Existência da necessidade de recurso ao ajuste direto se aplicável	Entidade adjudicante - Nome, NIF	Entidade adjudicatária - Nome, NIF	Objeto do Contrato	Procedimento Centralizado
0	29-01-2021	Aquisição de serviços	Ajuste Direto Regime Geral	Aquisição de serviços de apoio ao atendimento ...	Artigo 20.º, n.º 1, alínea d) do Código dos Co...	ausência de recursos próprios	Município de Lisboa	Go Connection S.A.	Aquisição de serviços de apoio ao atendimento ...	None
1	29-01-2021	Aquisição de serviços	Ajuste Direto Regime Geral	Prestação de serviços de assessoria técnica na...	Artigo 27.º, n.º 1, alínea b) do Código dos Co...	Prestação de serviços de assessoria técnica na...	Município de Lisboa	José Eduardo Antunes Romano Pires	Prestação de serviços de assessoria técnica na...	None
2	26-01-2021	Empreitadas de obras públicas	Consulta Prévia	Empreitada n.º14/DMMC/DHM/DIH/2020 - "Obras em...	Artigo 19.º, alínea c) do Código dos Contratos...	Não aplicável	Município de Lisboa	Manuel Pinto Pereira	"Obras em Prédios Municipais Sitos na Rua do L...	None
3	19-01-2021	Aquisição de serviços	Ajuste Direto Regime Geral	Aquisição de Serviços por Ajuste Direto – no á...	Artigo 27.º, n.º 1, alínea b) do Código dos Co...	ausência de recursos próprios	Município de Lisboa	José Leonardo Frutuosa Medinas	Aquisição de Serviços por Ajuste Direto – no á...	None
4	19-01-2021	Aquisição de serviços	Ajuste Direto Regime Geral	Aquisição de Serviços por Ajuste Direto – no á...	Artigo 27.º, n.º 1, alínea b) do Código dos Co...	ausência de recursos próprios	Município de Lisboa	Mário da Luz Antunes Pedro	Aquisição de Serviços por Ajuste Direto – no á...	None

Figura 4.4: Dataframe com dados extraídos do Portal

Desta etapa de aquisição de dados foi criado um *dataset* com todos os dados capturados. Como pode ser verificado na figura 4.5, o *dataset* contém 800333 contratos cada um com 27 *features*. Pode ser também verificado que existem *features* com um número elevado de valores nulos como no caso da *feature* de nome "Incrementos superiores a 15%" que apenas contém 3033 valores não nulos.

O elevado valor de valores nulos é um problema que afeta a qualidade do *dataset* e que deve ser tratado durante a fase de preparação de dados.

```

Int64Index: 800333 entries, 0 to 59999
Data columns (total 27 columns):
#   Column                                                                 Non-Null Count  Dtype
---  -
0   Data de publicação no BASE                                           800255 non-null object
1   Tipo(s) de contrato                                                  799941 non-null object
2   Tipo de procedimento                                                 800170 non-null object
3   Descrição                                                            800105 non-null object
4   Fundamentação                                                        799879 non-null object
5   Fundamentação da necessidade de recurso ao ajuste direto se aplicável 799789 non-null object
6   Entidade adjudicante - Nome, NIF                                    799586 non-null object
7   Entidade adjudicatária - Nome, NIF                                  799524 non-null object
8   Objeto do Contrato                                                  799608 non-null object
9   Procedimento Centralizado                                           17019 non-null  object
10  CPV                                                                  799566 non-null object
11  Data de celebração do contrato                                       799342 non-null object
12  Preço contratual                                                    799380 non-null object
13  Prazo de execução                                                  799382 non-null object
14  Localidade                                                         795027 non-null object
15  Concorrentes                                                       239735 non-null object
16  Anúncio                                                            75549 non-null  object
17  Incrementos superiores a 15%                                         3033 non-null  object
18  Documentos                                                          507575 non-null object
19  Observações                                                         47327 non-null  object
20  Data de fecho do contrato                                           237667 non-null object
21  Preço total efetivo                                                 257112 non-null object
22  Causas das alterações ao prazo                                       52349 non-null  object
23  Causas das alterações ao preço                                       74700 non-null  object
24  Distrito                                                            745266 non-null object
25  Nif Adjudicante                                                     799276 non-null float64
26  Nif Adjudicatario                                                  791210 non-null object
dtypes: float64(1), object(26)

```

Figura 4.5: Informação as variáveis dos dados recolhidos

4.5 Análise de dados

Nesta secção será feita uma análise geral acerca do *dataset* obtido na fase de aquisição de dados. Inicialmente será feita uma descrição da função de cada atributo do *dataset*. De seguida serão feitas análises temporais, geográficas e por CPV aos contratos de modo a verificar de que forma estão distribuídos o número atribuições e o dinheiro público despendido por cada distrito, localidade, categoria de CPV e mês.

4.5.1 Compreensão de dados

Os dados adquiridos através do *web scraper* foram convertidos para um ficheiro CSV e carregados utilizando a função de leitura de ficheiros CSV e conversão *dataframe* da biblioteca Pandas.

O *dataset* contém 770762 entradas e cada uma corresponde a um contrato público depositado no portal Base.Gov.

Cada entrada é caracterizada pelos seguintes atributos: Data de publicação no BASE, Tipo(s) de contrato, Tipo de procedimento, Descrição, Fundamentação, Fundamentação da necessidade de recurso ao ajuste direto se aplicável, Entidade adjudicante - Nome, NIF, Entidade adjudicatária - Nome, NIF, Objeto do Contrato, CPV, Data de celebração do contrato, Preço contratual, Prazo de execução, Localidade e Concorrentes.

Como demonstrado na tabela, os atributos de cada contrato são de 3 tipos, *float*, *int* e *object*. A presença do tipo *object* nos atributos deve-se ao facto de a função de *load* de *datasets* a partir de ficheiros CSV guardar as variáveis textuais com este formato.

Nome do atributo	Tipo	Descrição
Data de publicação no BASE	Object	Data em que o contrato foi publicado no portal Base.gov
Tipo(s) de contrato	Object	Tipo de serviço ou aquisição de bens associado ao contrato
Tipo de procedimento	Object	Tipo de procedimento utilizado para atribuição do contrato
Descrição	Object	Descrição resumida acerca do contrato em questão
Fundamentação	Object	Apresentação dos artigos do CCP que justifiquem os tipos de procedimento e contrato escolhidos
Fundamentação da necessidade de recurso ao ajuste direto se aplicável	Object	Justificação para a utilização de ajuste direto em contratos que o utilizam
Entidade adjudicante - Nome, NIF	Object	Nome e NIF da entidade adjudicante do contrato
Entidade adjudicatária - Nome, NIF	Object	Nome e NIF da entidade adjudicatária do contrato
Objeto do Contrato	Object	Descrição do serviço ou bem adquirido no contrato (frequentemente preenchido de forma igual à descrição)
CPV	Object	Código e descrição referente ao objeto do contrato, utilizando o vocabulário comum de contratação pública da União Europeia
Data de celebração do contrato	Object	Dia em que o contrato foi celebrado
Preço contratual	float64	Preço total da aquisição feita no contrato
Prazo de execução	int64	Número de dias estimados para a finalização do serviço contratado
Localidade	Object	Concelho e Distrito onde será realizado o serviço contratado
Concorrentes	Object	Lista de concorrentes, caso existam, para a atribuição do contrato

4.5.2 Exploração do *dataset*

Nesta secção será explorado o *dataset* de modo a tentar encontrar padrões e/ou desbalanceamentos na distribuição dos dados. Inicialmente será feita a análise temporal, com o objetivo de verificar de que forma o número de contratos e o dinheiro despendido em cada mês estão distribuídos no *dataset*. De seguida será feita uma análise geográfica de modo a verificar quais os distritos mais representados nos gastos de contratação do *dataset*. Por fim será feita uma análise por CPV com o objetivo de saber de que forma as categorias de contrato estão distribuídas no *dataset*.

4.5.2.1 Análise temporal

Os dados recolhidos foram depositados no portal Base.Gov entre Novembro de 2014 e Abril de 2021.

Na figura 4.6 é demonstrado o número de contratos depositados no portal em cada mês durante o intervalo de tempo dos contratos do *dataset*, sendo que pode ser verificado por esta figura que os números de contratos permanecem bastante homogêneos durante todo o espectro temporal avaliado, mas com uma ligeira subida no início de 2021.

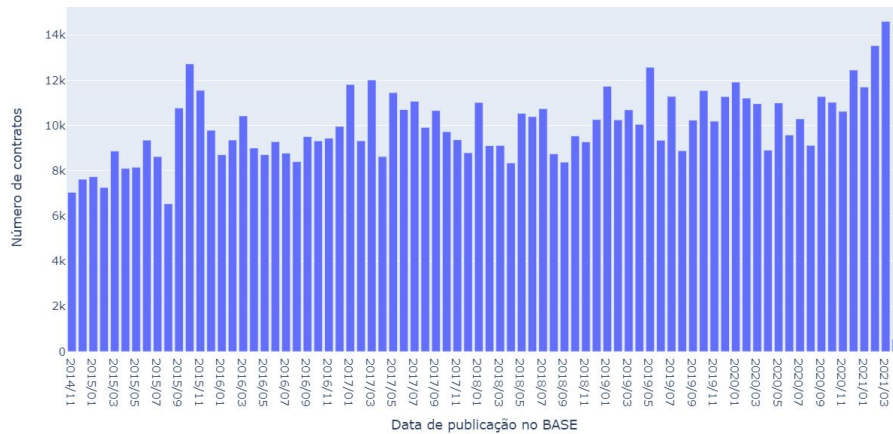


Figura 4.6: Número total de contratos depositados no portal Base em cada mês

A evolução temporal de gastos é um indicador que pode ajudar a encontrar padrões e tendências nos gastos por cada mês. Como pode ser verificado na figura 4.7, dentro do espectro temporal dos contratos do *dataset*, o mês de Janeiro de 2021 foi o mês em que houve uma maior quantidade de despesa em contratação pública o que demonstra uma provável influência do agravamento da situação pandémica vivida nesse período no *dataset*. Nesse período Portugal gastou cerca de 2 mil milhões de euros apenas em um mês sendo que até aí nunca tinha sido despendido mais de mil milhões em nenhum dos meses registados no *dataset*.

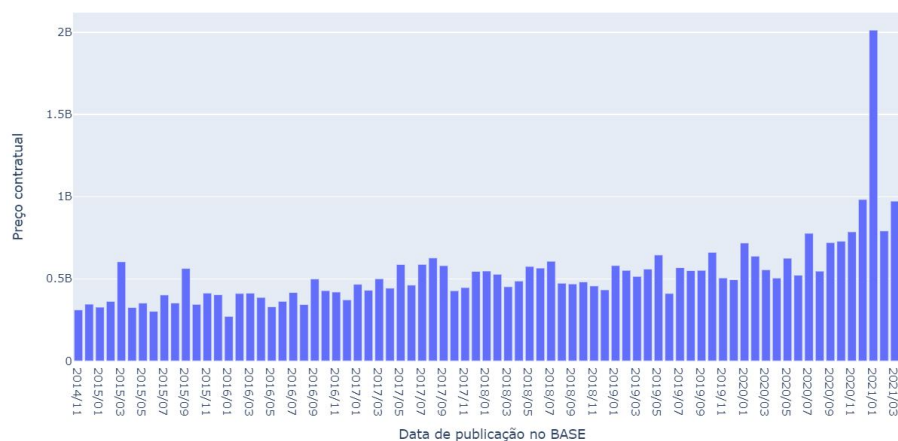


Figura 4.7: Preço total em euros gasto em contratos em cada mês

4.5.2.2 Análise Geográfica

A distribuição geográfica dos contratos é um fator importante para a compreensão do *dataset* visto que permite verificar o equilíbrio da representação de cada distrito e localidade nos contratos do *dataset*. Como

tal foram de seguida analisadas as particularidades geográficas do *dataset*, de modo a compreender a distribuição de contratos e valor monetário despendido em contratação pública por cada localidade e distrito.

A figura 4.8 demonstra uma grande preponderância do distrito de Lisboa seguido do Porto, no número de contratos celebrados em relação aos restantes distritos, o que por si só não representa um desequilíbrio no *dataset* porque estes são também os distritos mais populosos de Portugal. De realçar também que o principal procedimento escolhido por todos os distritos é o ajuste direto, perfazendo em todos eles mais de 50% dos contratos registados.

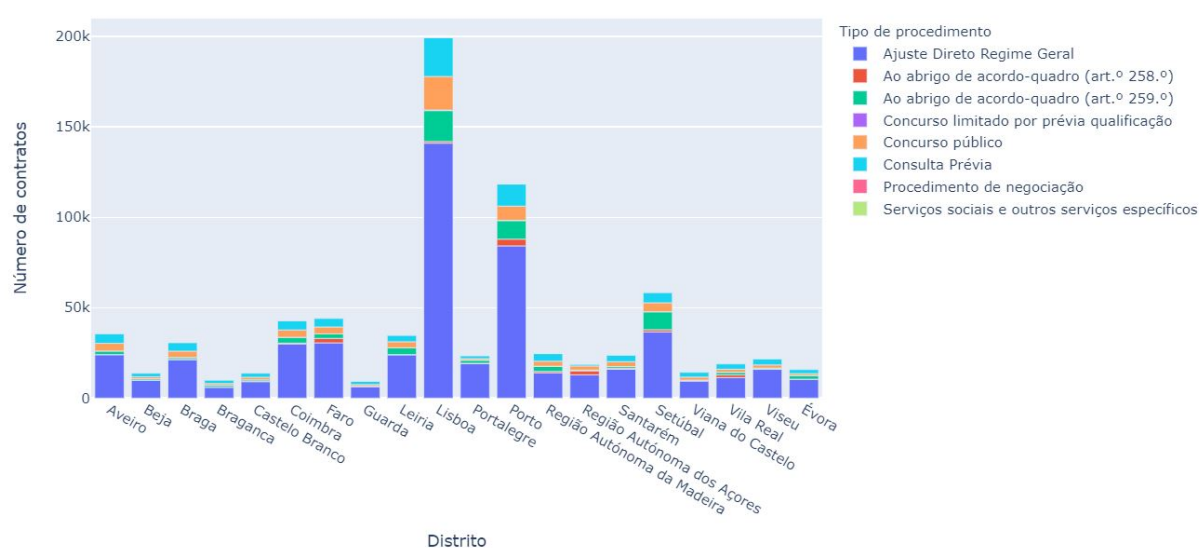


Figura 4.8: Distribuição do dinheiro despendido por cada distrito em cada CPV

4.5.2.3 Análise de CPV

CPV é um vocabulário comum da União Europeia que permite através de um código com 9 algarismos classificar os fornecimentos, obras ou os serviços objeto de um contrato. Por se tratar de um código hierárquico, é possível através dos primeiros 3 algarismos encontrar a categoria a que pertence cada objeto de um contrato, o que permite uma análise estatística de quais as categorias de obras, fornecimentos ou serviços são mais preponderantes no panorama da contratação pública em Portugal.

Na figura 4.9 está representado por cada mês a categoria de CPV onde foi despendido mais dinheiro, sendo que maioritariamente essa categoria são trabalhos de construção. De realçar também que em Janeiro de 2021 foi verificado um valor invulgarmente elevado na categoria dos serviços de transporte.

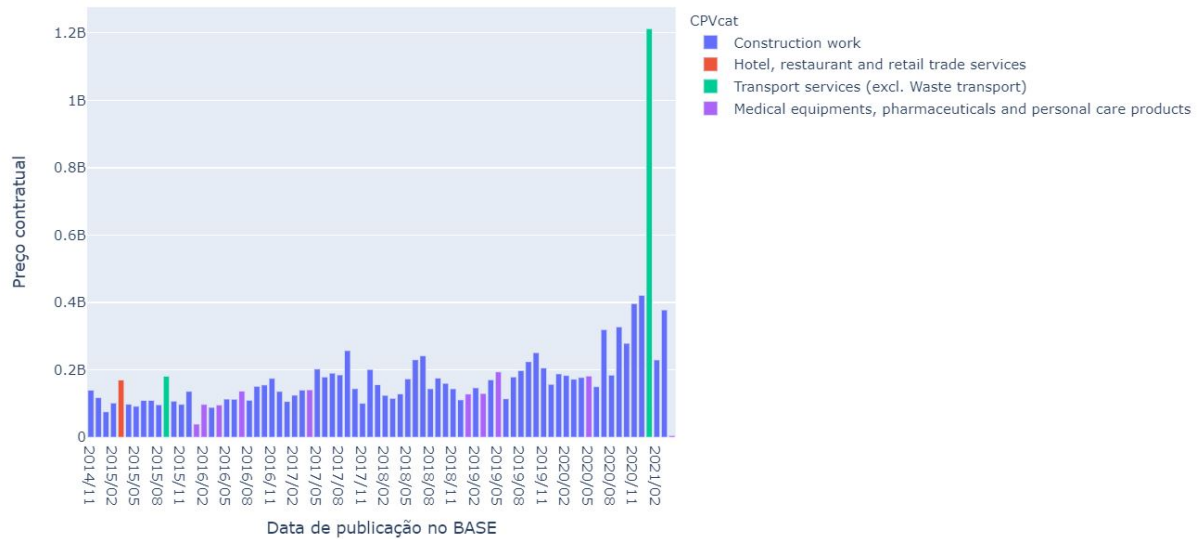


Figura 4.9: Número de contratos por distrito e por tipo de procedimento

De modo a reforçar a análise estatística anterior em que era perceptível uma forte preponderância da categoria de trabalhos de construção, foram colocadas as despesas por categoria de CPV numa *pie chart* que se encontra representada na figura 4.10, de modo a entender se existia ou não uma distribuição igualitária entre categorias de CPV no *dataset*. Esse gráfico permite concluir que existe um gasto significativamente superior nas categorias médica e farmacêutica e trabalhos de construção em relação às restantes categorias de CPV, sendo que estas duas perfazem quase 50% dos gastos de todos os contratos presentes no *dataset*

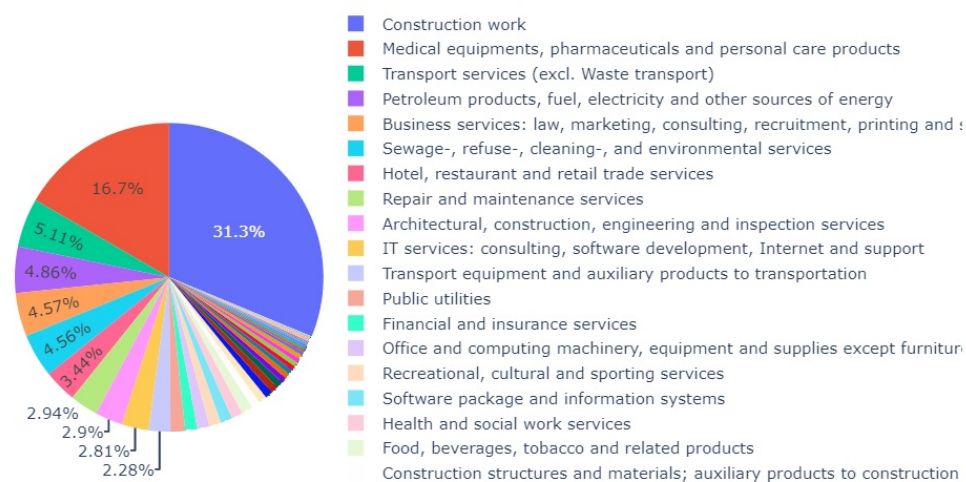


Figura 4.10: Percentagem de despesa referente a cada categoria de CPV

Desenvolvimento

Este capítulo aborda em detalhe as diversas fases de desenvolvimento deste projeto.

Inicialmente serão abordadas detalhadamente cada uma das experimentações feitas para extração de conhecimento, nomeadamente, análise de redes sociais, deteção de conluio e irregularidades em contratos públicos, e previsão de custos futuros em setores de contratação pública. Cada uma das experimentações efetuadas contará com as motivações que levaram à escolha do método ou algoritmos utilizados, código-fonte das fases mais importantes de implementação, resultados obtidos e conclusões retiradas.

Por fim será abordada uma implementação simples para uma *interface* que permite a comunicação de um utilizador com o *dataset*, produzindo não só diversos gráficos com dados estatísticos dos dados, como também uma página de teste de validade de contratos com base no sistema de regras e algoritmos de *Machine Learning* criados para esse fim.

5.1 Implementação e análise de redes sociais

Nesta secção será explicitado todo o processo de criação de grafos de conhecimento através de um *dataset* de contratos públicos portugueses.

O ponto de começo para a criação de um grafo que permita extrair conhecimento acerca das relações e importância de entidades no panorama geral de contratos públicos em Portugal é analisar quais são as variáveis dos dados que permitem relacionar esses dados. Neste caso, as variáveis de relevo para esta análise, são as que contêm a entidade adjudicante e adjudicatária do contrato.

Para cada contrato do *dataset* foi criada uma entrada num *dataframe* que contém 4 variáveis, entidade adjudicante, entidade adjudicatária, uma relação entre os dois denominada de "contratado por", e o preço contratual do contrato em questão. Apenas essas 4 variáveis foram escolhidas devido ao facto de serem as variáveis demonstrativas da relação entre cada par de entidades, tendo sido considerado que as restantes

particularidades do contrato (eg. justificação, objeto, descrição, localidade, etc.) não seriam relevantes para a análise das redes, nem poderiam ser computados pelos algoritmos de análise de grafos, visto que são variáveis nominais. Desse modo, um par de entidades que celebrem um contrato público entre elas, passam a estar relacionadas no grafo por uma relação entre dois nós, cada nó que contém o NIF de cada entidade, e a aresta contém o preço do contrato.

Após criado o *dataframe* com as entidades e as relações entre elas, foi utilizada a biblioteca Networkx de modo a construir um grafo com esses dados.

O resultado final da criação do grafo está representado na figura 5.1, sendo que neste caso foram apenas usados os contratos da localidade de Vila Nova de Famalicão, de forma a ser mais fácil a percepção das ligações, visto que o grafo de todos os contratos do *dataset* é demasiado povoado para serem perceptíveis as relações entre entidades.

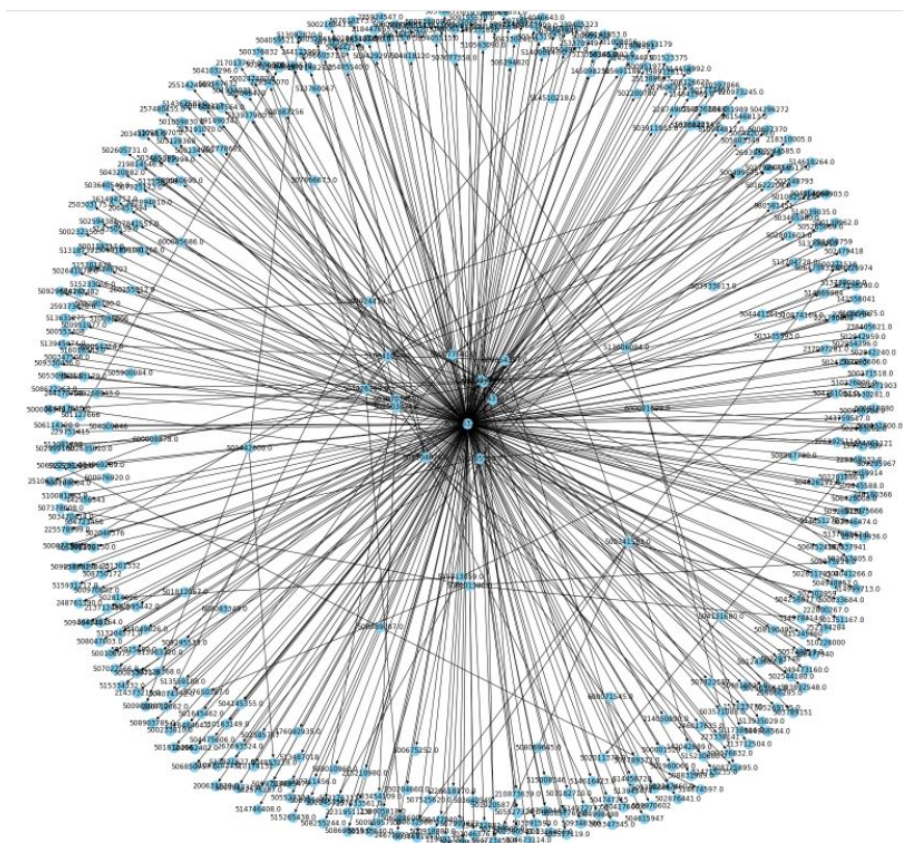


Figura 5.1: Grafo representativo das entidades dos contratos da localidade de Vila Nova de Famalicão

Numa figura como a 5.1 torna-se difícil verificar as entidades mais centrais apenas com observação, o que torna relevante a aplicação de algoritmos de centralidade de modo a observar segundo diversos algoritmos quais são as entidades mais centrais deste conjunto de entidades. Foram aplicados algoritmos de centralidade (*eigenvector* e *degree centrality*) ao grafo da figura 5.1 para verificar se as entidades mais centrais do grafo, os resultados das 5 entidades com maior coeficiente de centralidade do grafo, segundo o algoritmo de *eigenvector* e *degree centrality* estão nas figuras 5.2 e 5.3, respetivamente.

```
[('Município de Vila Nova de Famalicão ', 0.7070823558811804),
 ('EDP Comercial - Comercialização de Energia, S.A. ', 0.01905483873482095),
 ('Larestil - Construções, Lda ', 0.019027775794002597),
 ('Dacop - Construções e Obras Públicas, S.A. ', 0.01902723028379588),
 ('Arnaldo Fernandes & Cª,Lda ', 0.019027230283795868)]
```

Figura 5.2: Top 5 entidades do grafo com maior *eigenvector centrality*

```
[('Município de Vila Nova de Famalicão ', 0.5817649519029694),
 ('Centro Hospitalar do Médio Ave, E. P. E. ', 0.061062317022166454),
 ('Agrupamento de Escolas Padre Benjamim Salgado ', 0.027185278126306982),
 ('Forave - Associação para a Educação Profissional do Vale do Ave ',
 0.025094102885821833),
 ('Agrupamento de Escolas D. Sancho I ', 0.020911752404851526)]
```

Figura 5.3: Top 5 entidades do grafo com maior *degree centrality*

De seguida foi efetuada a tentativa de detetar comunidades dentro do círculo de contratos, ou seja, grupos de entidades com um acentuado número de contratos celebrados interligados entre si. Para isso, foi utilizado o algoritmo de Girvan Newman, que permite detetar comunidades em grafos de conhecimento através da eliminação sucessiva de arestas. Este algoritmo foi aplicado ao grafo anteriormente construído e devolveu a lista de todas as comunidades nele encontradas. Algumas das comunidades encontradas no *subset* do distrito de Braga estão demonstradas na figura 5.4.

```
[('Conecticabo - Instalação de ', 'Infraestruturas de Portugal, S. A. '),
 ('Administração Regional de Saúde do Norte, I. P. ',
 'Advantarget - Climatização e Ventilação, Lda '),
 ('AREAL EDITORES, SA ',
 'Agrupamento de Escolas de Pedome, Vila Nova de Famalicão ',
 'Cantinho Ideal ',
 'Quaselink,Lda '),
 ('Centro Social Panoquial de Ribeirão ',
 'INOVA+ - INNOVATION SERVICES S.A. ',
 'Mentortec - Serviços de Apoio a Projectos Tecnológicos, S.A. '),
 ('REN - Rede Eléctrica Nacional, S. A. ', 'Siemens, S.A. '),
 ('Associação para a Formação Tecnológica e Profissional da Beira Interior ',
 'Centro Tecnológico das Indústrias Têxtil e do Vestuário de Portugal - CITEVE ',
 'Universidade do Minho '),
 ('Ave - Cooperativa de Intervenção Psico-Social, C. R. L. ',
 'Galp Power, S.A. ',
 'Rui Miguel Moreira, Unipessoal, Lda. ',
 'SOFTMAP - Informática e Telecomunicações ',
 'V FONTES CAR - COMERCIO DE AUTOMOVEIS LDA ')]
```

Figura 5.4: Exemplos de comunidades encontradas pelo algoritmo Girvan Newman

Apesar de terem sido criados com sucesso grafos de conhecimento com base nos dados de contratação pública portuguesa, a percepção final foi de que esta implementação teve pouca relevância no âmbito da deteção de conluio a partir dos relacionamentos entre entidades. Os algoritmos de deteção de comunidades identificaram comunidades na sua maioria constituídas por centenas de entidades, o que levantou sérias dúvidas sobre a possibilidade da sua utilização como modo de deteção de conluio, visto que o objetivo era que a deteção de comunidades conseguisse detetar comunidades constituídas por poucas entidades, dentro do panorama geral, para que fosse possível identificar círculos de conluio, mas, isto é algo que com a grande dimensão das comunidades previstas pelo algoritmo de Girvan Newman não pode ser feito.

5.2 Deteção de indicadores de conluio

Como um dos objetivos proposto para este trabalho foi detetar indicadores que possam levar a suspeição de conluio em contratos públicos, foi implementado um sistema de regras baseado em indicadores de

conluio retirados de artigos divulgados pela AdC e pelo OCDS, este sistema de regras permite testar contratos quanto aos indicadores de conluio definidos, e, consoante o número de indicadores acionados, atribui um valor de suspeição entre 0 e 5. É necessário ressaltar que o cenário ideal para conseguir identificar conluio seria treinar um modelo de *Machine Learning* com contratos previamente classificados como irregulares pelas entidades governamentais competentes, mas não foram encontradas quaisquer fontes de dados governamentais na *web* que disponibilizassem tais contratos.

Um grande problema que surge neste âmbito, é que algumas informações acerca das entidades intervenientes nos contratos que foram utilizadas para detetar indicadores de conluio pelos artigos presentes na secção do estado da arte[29] (e.g moradas e contactos) não são disponibilizados pelo portal Base.gov.

Como tal foram implementadas 6 regras que serão descritas nas subsecções seguintes, baseadas nas regras de integridade presentes no padrão de OCDS (*Open Contracting Data Standard*), que é um instrumento desenvolvido pela *Open Contracting Partnership* com o objetivo de que garantir a transparência e a qualidade dos sistemas de *e-procurement* em cada etapa do ciclo de compras públicas.

5.2.1 Detecção de Red Flags de Conluio

Nesta subsecção serão abordadas diversas *red flags* de contratos públicos, estas foram retiradas do documento de *red flags* do OCDS[70] e do Guia das Boas Práticas da AdC[49] e expressam características presentes em certos contratos que fazem com que estes se tornem mais suspeitos de estar presentes em esquemas de conluio.

5.2.1.1 Desigualdade na atribuição

Um dos critérios considerados como *red flag* no OCDS[70] para integridade da contratação pública é a atribuição de um elevado número de contratos a um grupo de restrito de empresas. Para analisar se a contratação pública em Portugal cumpre este critério foi utilizada a curva de *Lorenz* como valor de referência. A curva de *Lorenz* é considerada a curva de igualdade absoluta, ou seja, representa a curva de atribuição de contratos num país totalmente igualitário em termos de atribuição de contratos em que o valor recebido por todas as entidades adjudicantes era o mesmo.

A figura 5.5 demonstra a comparação entre a curva de *Lorenz* e a curva real da contratação em Portugal, onde pode ser verificado que 10% das entidades presentes no *dataset* arrecadaram 80% do dinheiro gasto por Portugal em contratação pública nesse mesmo período, o que pode ser um indício de um desbalanceamento na contratação pública em Portugal entre empresas.

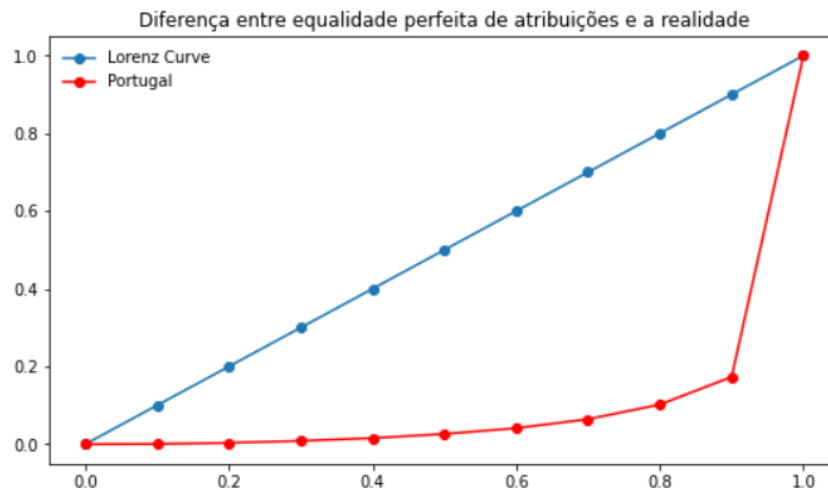


Figura 5.5: Curva de Lorenz para os contratos do *dataset*

Para obter este gráfico que sustenta a afirmação anterior, foram ordenadas as empresas numa lista por ordem crescente do somatório dos valores dos contratos que lhes foram atribuídos. Esta lista foi então subdividida em 10 grupos para cada um desses grupos representar 10% da amostra de empresas. Por fim, para cada um desses grupos foi efetuado o somatório de todas as empresas incluídas. Este processo de captura do top 10% de empresas com mais dinheiro recebido está representado na função em 5.1, função essa que sinaliza essas entidades no *dataset* com uma *red flag*.

```

1 def top10PercentCompanies(df):
2     df2=df[df["Preço contratual"]>0].groupby("Nif Adjudicatario")[["Preço contratual"]].sum().
3     ↪ reset_index().sort_values(by="Preço contratual")[:-1]
4     percentil90=np.percentile(np.array(df2["Preço contratual"]),90)
5     flags=df2[df2["Preço contratual"]>=percentil90]["Nif Adjudicatario"]
6     for ind,val in dt.iterrows():
7         if (val["Nif Adjudicatario"] in list(set(flags))):
8             df.loc[ind,"redFlagCount"]=val["redFlagCount"]+1

```

Listagem 5.1: Função que sinaliza o percentil 90 das entidades com mais dinheiro recebido em contratos

5.2.1.2 Ajuste Direto

Outro critério considerado como *red flag* é a utilização de ajustes diretos como tipo de procedimento, o que leva à falta de concorrência e consequentemente pode lesar o estado português. Na figura 5.8 está demonstrada a frequência de cada tipo de procedimento nos contratos do *dataset*. Como pode ser observado o tipo de procedimento mais utilizado é o ajuste direto, correspondendo a aproximadamente 69% dos contratos realizados.

Quanto aos valores monetários despendidos em ajuste direto, na figura pode ser verificado que apenas concursos públicos têm maior número de quantia despendida, sendo que o ajuste direto representa cerca de 31% dos gastos.

	Tipo de procedimento	Contagem
0	Ajuste Direto Regime Geral	531241
1	Ao abrigo de acordo-quadro (art.º 258.º)	14414
2	Ao abrigo de acordo-quadro (art.º 259.º)	62571
3	Concurso limitado por prévia qualificação	1232
4	Concurso público	69735
5	Consulta Prévia	91262
6	Procedimento de negociação	12
7	Serviços sociais e outros serviços específicos	1

Figura 5.6: Contagem do número de aparições de cada tipo de procedimento no *dataset*

	Tipo de procedimento	Preço contratual
0	Ajuste Direto Regime Geral	1.250572e+10
1	Ao abrigo de acordo-quadro (art.º 258.º)	7.588031e+08
2	Ao abrigo de acordo-quadro (art.º 259.º)	4.628871e+09
3	Concurso limitado por prévia qualificação	1.604716e+09
4	Concurso público	1.844329e+10
5	Consulta Prévia	2.744056e+09
6	Procedimento de negociação	8.087334e+06
7	Serviços sociais e outros serviços específicos	2.105597e+06

Figura 5.7: Contagem do dinheiro gasto em cada tipo de procedimento no *dataset*

De modo a tentar identificar a utilização excessiva de ajustes diretos foram sinalizados com *red flag* os contratos atribuídos a entidades cujos contratos celebrados sejam mais de 50% ajustes diretos, através da função representada em 5.2.

```

1 def ajustesExcessivos(df,nif):
2     tamanho=len(df[df["Nif Adjudicatario"]==nif])
3     valor=len(df[(df["Nif Adjudicatario"]==nif) & (df["Tipo de "procedimento"]==Ajuste Direto
4         ↳ Regime Geral)])
5     if (valor>(tamanho/2)):
6         return True
7     else:
8         return False

```

Listagem 5.2: Função de deteção da utilização excessiva de ajustes

5.2.1.3 Concursos com apenas um concorrente

Tal como o ajuste direto, concursos públicos com apenas um concorrente são uma *Red Flag* devido à falta de concorrência, o que prejudica a eficiência dos gastos na contratação pública de um país. Na

figura pode ser observado que 26900 concursos públicos de 69735 têm apenas um concorrente, o que representa cerca de 39% dos concursos efetuados.

Para sinalizar estes contratos, foram filtrados todos os concursos com apenas 1 concorrente e sinalizados com uma *red flag*.

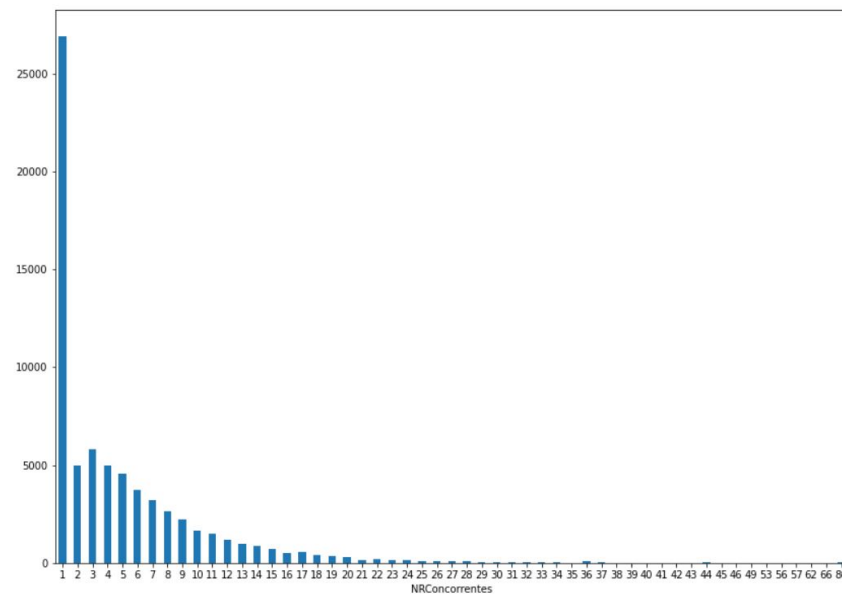


Figura 5.8: Distribuição do número de concorrentes de cada concurso público presente no *dataset*

5.2.1.4 Contratos com valor excessivo

Outra *Red Flag* na contratação pública são contratos com valor excessivo, ou seja, muito superior aos valores comuns de acordo com o respectivo CPV do contrato. Para tal foi desenvolvida uma função de detecção deste tipo de valores elevados, utilizando a fórmula para detecção de *outliers*. Para cada CPV é calculado o *Interquartile Range* a partir da subtração do valor do percentil 75 pelo valor do percentil 25 da distribuição. Se o preço contratual de cada contrato corresponder à condição seguinte é considerado um *outlier*, acionando essa *red flag* caso o preço contratual obedeça à fórmula seguinte:

$$PrecoContratual > Percentil75 + 1,5 * InterquartileRange \quad (5.1)$$

```

1 def valorElevDis(df):
2     hist=[]
3     df.loc[[i for i in range(0,len(df))],"HigherValue"]=False
4     for ind,row in df.iterrows():
5         if((row["CPV"],row["Distrito"]) not in hist):
6             temp=df[(df["CPV"]==row["CPV"])]
7             percentil75=np.percentile(np.array(temp["Preço contratual"]),75)
8             temp=temp[temp["Preço contratual"]>(percentil75+1.5*IQR(temp["Preço contratual"]))]
9             listIndexes=list(temp.index.values)
10            df.loc[listIndexes,"HigherValue"]=True

```



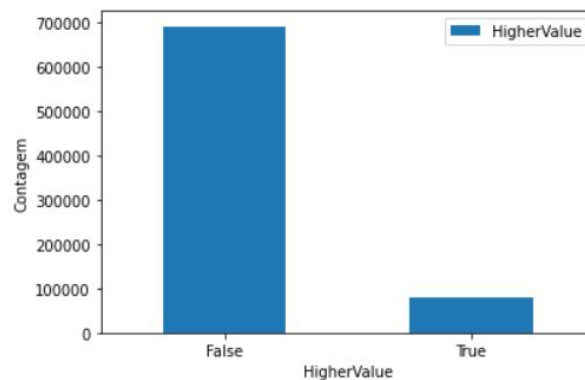
```

11     hist.append((row["CPV"],row["Distrito"]))
12     return df

```

Listagem 5.3: Função de detecção de contratos com valor excessivo

Todos os contratos que cumpram a condição descrita no parágrafo anterior, são classificados numa coluna do *dataset* criada para esse efeito chamada "HigerValue" e é-lhes adicionada uma *red flag*. A distribuição de classificações está representada na figura 5.9

Figura 5.9: Contagem dos contratos do *dataset* quanto ao valor do atributo HigherValue

5.2.1.5 Vencedores e perdedores comuns

Segundo a AdC, padrões repetitivos de vencedores e perdedores é um indício de conluio. Para investigar a existência deste padrão nos contratos do *dataset* foi criada uma função que contabiliza para cada entidade que já venceu pelo menos um concurso, o número de derrotas das outras entidades que perderam os concursos ganhos por ela. A partir do resultado dessa função foram sinalizadas todas as entidades que perdem mais de 33,(3)% dos concursos em que outra determinada entidade é vencedora. Foram para esta sinalização apenas consideradas entidades com uma participação em mais de 10 concursos, de modo a não penalizar entidades com poucos concursos.

Em 5.4 está a função utilizada para percorrer o *dataset* e contar nos concursos ganhos por uma dada entidade, quais foram as entidades que mais vezes perderam. Esta função retorna um dicionário com o NIF de cada empresa como *key* e o número de vezes que foi derrotada como *value*, para que depois a função de cálculo das *red flags* receba este dicionário e apenas tenha de verificar se cada entidade perdedora esteve como concorrente derrotado em pelo menos um terço dos contratos ganhos pela entidade vencedora.

```

1 def CommonWL(NifEmp, Nome, Dataframe):
2     dft=Dataframe
3     contagem={}
4     temp=False
5     for ind,cont in dft.iterrows():
6         for concorrente in cont["Concorrentes"]:

```



```

7         if(concorrente!=Nome):
8             for key in contagem.keys():
9                 if(concorrente==key):
10                     contagem[key]=contagem[key]+1
11                     temp=True
12                     break
13             if(temp==False):
14                 contagem[concorrente]=1
15             temp=False
16     return contagem

```

Listagem 5.4: Função de contagem de derrotados

5.2.1.6 Partilha de regiões entre entidades

Para além das *red flags* do OCDS descritas nas subsecções anteriores foi considerado também relevante atribuir uma *red flag* a contratos atribuídos a empresas que tenham uma soberania muito elevada na região de celebração desse contrato, algo que é considerado pela AdC como um indício de conluio.

Inicialmente, foi criada uma função para contabilizar o número de vitórias de uma empresa numa determinada região e para um determinado CPV que pode ser vista em 5.5. Esta função tem como objetivo identificar o conluio de repartição de regiões por concorrentes, em que várias empresas decidem dividir o mercado por regiões, recebendo as vitórias na sua região, e perdendo propositadamente os concursos nas regiões das restantes empresas, lesando o estado português.

Por fim, foi definido o *threshold* de que um contrato atribuído a uma empresa que tenha ganho acima de 30% dos contratos de um certo CPV celebrados nessa região seria sinalizado com uma *red flag*.

```

1 def regionWinners(contrato,df,distrito):
2     regiao=classifierRegion(distrito)
3     vencedor=0
4     perdedor=0
5     ds=df[(df['CPV']==contrato['CPV']) & (df2["Localidade"]==regiao)]
6     ds=ds.reset_index(drop=True)
7     for i in range(0,len(ds)):
8         if(ds.loc[i]['Nif Adjudicatario']==contrato['Nif Adjudicatario']):
9             vencedor=vencedor+1
10        else:
11            perdedor=perdedor+1
12    return (vencedor,perdedor,contrato["Entidade adjudicatária - Nome, NIF"],contrato["Nif
    ↪ Adjudicatario"])

```

Listagem 5.5: Função de vitórias e derrotas de uma empresa numa região

5.2.2 Avaliação do risco de conluio

Apesar de já terem sido encontradas algumas *red flags* que permitem suspeitar de conluio, existe a necessidade de aglomerar toda essa informação numa variável única.

Para tal foram somados os valores das *red flags* previstas de modo a sinalizar todos os contratos do *dataset* com um grau de risco correspondente à soma das *red flags* acionadas.

Apesar de, como referido anteriormente, não estarem disponíveis contratos previamente classificados como colusivos, foi utilizada uma notícia[40] do ano 2021 do Jornal Expresso que indica diversas entidades acusadas pela AdC por práticas de conluio em contratos públicos na área da segurança, empresas essas que foram usadas para tentar validar a utilidade dos indicadores de conluio na deteção de conluio.

Para tentar efetuar a validação dos indicadores foram utilizados todos os contratos com o CPV da área de serviços de segurança durante todo o espectro temporal do *dataset* (visto que segundo a notícia estas práticas já se estendem desde 2009). Segundo a mesma notícia, as empresas acusadas são 2045/Gália, Comansegur, Grupo 8, Prestibel, Prosegur, Securitas e Strong Charon, e a prática de conluio feita por elas estendeu-se a todo o território nacional, pelo que não será necessário filtrar contratos por região.

A este *subset* foram aplicadas as funções de deteção de cada *red flag*, e foi calculado para cada uma dessas empresas a média de *red flags* que cada um dos seus contratos contém. A análise foi feita a partir da média visto que cada uma das empresas citadas tem um número elevado de contratos no *dataset* e desta forma conseguimos analisar o risco médio de conluio atribuído pelo método de deteção para cada entidade. Cada uma das entidades tem um valor entre 0 e 5 visto que apesar de serem 6 *red flags*, duas delas são mutuamente exclusivas, nomeadamente a da utilização de ajuste direto e a de concursos públicos com um único concorrente. Na figura 5.10 está demonstrado o gráfico final com o nome que cada uma das empresas contém no portal Base, e para cada uma delas, o valor médio de *red flag* nos seus contratos. Tendo as *red flags* obrigatoriamente um valor de 0 a 5 como já foi explicado no parágrafo anterior, foi previamente definido um valor *threshold* de 2.5 visto ser o ponto que divide o espectro de valores de classificação em metade. Todas as empresas com média acima desse valor seriam consideradas suspeitas para fins de validação.

Todas as empresas no gráfico possuem um valor médio acima de 2.5 *red flags*, apesar de as empresas Securitas e Grupo 8 estarem bastante próximas do *threshold*. Pelo contrário a Prosegur e a Super Charon tiveram uma média de 4.03 e 3.92 respetivamente, o que significa que os indicadores apontam para uma suspeita elevada de conluio visto que o valor máximo é 5.

Esta tentativa de validação foi promissora quanto à validade do sistema de deteção de *red flags*, visto que mesmo num *dataset* de contratos com escassez de dados de relevo para a verificação de conluio, e sem saber quais contratos das empresas é que objetivamente foram fruto de conluio (considerando que provavelmente as empresas não efetuaram práticas de conluio em todos os contratos celebrados), foi possível acionar diversos indicadores de conluio em contratos de empresas que foram processadas pela AdC pela prática de conluio num espectro temporal englobado no *dataset*.

redFlagCount	Nome empresa
2.559322	Securitas - Serviço e Tecnologia de Segurança,...
3.343750	2045-Gália/Serviços de Vigilância e Segurança,...
2.842105	GRUPO 8 - Vigilância e Prevenção Electrónica, ...
3.708333	COMANSEGUR - Segurança Privada, S.A.
3.920000	CHARON - Prestação de Serviços de Segurança e ...
3.561404	Prestibel – Empresa de Segurança, S.A.
4.032787	Prosegur

Figura 5.10: Gráfico do número médio de *red flags* acionadas pelos contratos de cada uma das empresas referenciadas

5.3 Detecção de Irregularidades

Nesta secção serão abordados todos os passos tomados no âmbito da criação de um *pipeline* capaz de classificar contratos como irregulares.

Inicialmente será abordado o *rule based system*, ou seja, o sistema de regras extraídas do CCP a partir do qual se pode verificar quais os contratos que estão nos conformes da regulamentação.

A subsecção seguinte aborda a utilização de um *pipeline* de *Machine Learning*, desde o tratamento de dados até à aplicação de algoritmos e otimização de hiperparâmetros.

5.3.1 Rule Based System

Com o objetivo de classificar os contratos com base na sua regularidade conforme os regulamentos nacionais, foi analisado o CCP e codificadas as suas regras num sistema de regras. Como as condições definidas pelas regras do CCP para classificar um contrato são maioritariamente baseadas em intervalos de valores numéricos, um sistema de regras permite sintetizar a maiorias dessas condições e auxiliar um utilizador no processo classificação de contratos sem necessidade que o utilizador tenha de ler o CCP.

O sistema foi criado também como meio de classificar os contratos existentes no *dataset*, servindo de base para a criação de modelos de previsão de irregularidades em contratos, visto que não foram encontrados contratos objetivamente classificados como irregulares disponibilizados pelo setor de contratação em Portugal para servir como referência.

O sistema de regras criado determina se um contrato se encontra nos parâmetros legislados. Esta avaliação é feita com base no tipo de contrato, tipo de procedimento e artigo do CCP utilizado como fundamentação. Para cada uma das combinações destes atributos anteriores há um valor monetário máximo definido, valor esse que vai ser comparado com o valor do contrato em questão, verificando assim para cada contrato a sua regularidade.

Na tabela 5.1 estão as condições utilizadas para a conceção do *Rule based system* capaz de definir se os contratos são regulares ou irregulares.

Devido ao facto de no *dataset* apenas existirem referências a Ajuste Direto Regime Geral, foram excluídas as regras referentes ao regime simplificado.

Foi também necessário encontrar um método de distinção dos contratos do *dataset* referentes a concurso público nacional e concurso público urgente, visto que ambos estão referenciados apenas com a designação comum "Concurso público" na coluna "Tipo de procedimento". Nesse âmbito foi verificado que essa distinção era possível através da análise do artigo utilizado para fundamentação do contrato em questão, visto que concursos públicos urgentes utilizam como fundamentação o artigo 151º do CCP, artigo esse que se destina exclusivamente a concursos públicos urgentes.

Tabela 5.1: Valores máximos para cada combinação de tipo de contrato e procedimento

Tipo de procedimento	Tipo(s) de contrato	Preço Máximo
Ajuste direto regime geral	Locação ou aquisição de bens móveis e aquisição de serviços	20000€
	Empreitadas de obras públicas	30000€
	Concessão de serviços públicos ou concessão de obras públicas	75000€
	Restantes contratos	50000€
Consulta prévia	Locação ou aquisição de bens móveis e aquisição de serviços	75000€
	Empreitadas de obras públicas	150000€
	Concessão de serviços públicos ou concessão de obras públicas	75000€
	Restantes contratos	100000€
Concurso público nacional	Locação ou aquisição de bens móveis e aquisição de serviços	221000€
	Empreitadas de obras públicas	5548000€
Concurso público urgente	Locação ou aquisição de bens móveis e aquisição de serviços	221000€
	Empreitadas de obras públicas	300000€
Concurso limitado por pré- via qualificação	Locação ou aquisição de bens móveis e aquisição de serviços	221000€
	Empreitadas de obras públicas	5548000€
	Restantes contratos	Qualquer valor

5.3.2 Classificação de irregularidades em contratos

Após a criação do *Rule Based System* no *dataset*, foram classificados todos os contratos do *dataset* com base nas suas regras. Com isto foi obtida uma coluna no *dataset* em que cada contrato está classificado como regular ou irregular.

Apesar de este sistema de regras já permitir classificar todos os contratos completos que lhe sejam inseridos, não consegue lidar com contratos com informação incompleta, o que poderia vir a ser útil no caso de um sistema de auxílio de feitura de contratos em que à medida que o utilizador insere atributos de um contrato, o sistema vai tentando prever se este é regular ou irregular, informando o utilizador.

Neste contexto, foram então utilizadas técnicas de *Machine Learning* para a previsão de irregularidades devido à possibilidade de aprendizagem de padrões de irregularidade com um número mais reduzido de

variáveis, podendo lidar assim com contratos que estão a ser inseridos e contêm informação incompleta.

5.3.2.1 Preparação de dados

A preparação de dados é uma fase crucial para a obtenção de bons resultados na aplicação de modelos de *Machine Learning*. Os passos tomados para a preparação do *dataset* foram:

- Enriquecimento de dados
- Remoção de valores nulos e dados duplicados
- *Feature engineering*
- Aplicação de técnicas de *sampling*
- *Feature scaling*
- *Encoding* de variáveis nominais
- *Train-test split*

Foram inicialmente tratados os valores nulos do *dataset*, tendo sido decidido que seriam eliminados todos os contratos com valores de preço contratual nulos. Uma alternativa possível seria a substituição dos valores nulos pela média para esse atributo, o que causaria contratos com valores fictícios, o que afetaria negativamente a veracidade dos dados, e por essa razão não foi escolhida.

De seguida, foi realizado enriquecimento de dados, ou seja, a partir de *features* presentes no *dataset*, foram criadas novas *features* que podem ser relevantes para a eficácia de previsão do modelo. Um exemplo de enriquecimento de dados realizado foi a criação de uma *feature* com a categoria do CPV do contrato. A categoria do CPV foi extraída através dos primeiros 3 dígitos do CPV geral, e foi utilizada no modelo de *machine learning* em detrimento do código CPV inteiro do contrato, isto deve-se à existência de milhares de diferentes códigos CPV no *dataset*, o que não é favorável para algoritmos de *Machine Learning*, que lidam melhor com *features* com um espectro de valores possíveis menos amplo. Essa nova *feature* criada passou a ter 45 valores únicos no *dataset*, enquanto que a *feature* original do CPV total tinha 6293.

De seguida foram utilizadas técnicas de *Feature engineering* de modo a escolher as variáveis ótimas no contexto do problema, eliminando variáveis com baixos níveis de correlação com a variável que será utilizada como *Target* no modelo de *Machine Learning*, neste caso essa variável chama-se "Irregular".

Na figura 5.11 está representada a matriz de correlação entre as variáveis do *dataset*. A partir dela podemos verificar não só a correlação das variáveis de treino para com a variável *target* do modelo, como também podem ser verificadas correlações entre variáveis de treino, como por exemplo, a existência de uma correlação forte entre o tipo de procedimento e o número de concorrentes (esta correlação deve-se à forte dependência entre as duas variáveis, por exemplo, os contratos realizados com o procedimento de concurso público têm um número médio de concorrentes muito superior aos restantes procedimentos, que na maioria têm apenas um concorrente).

	Tipo(s) de contrato	Tipo de procedimento	Fundamentação	Preço contratual	Prazo de execução	Localidade	Distrito	Nif Adjudicante	Irregular	CPVcat	NRConcorrentes	redFlagCount
Tipo(s) de contrato	1.000000	0.022173	-0.079557	0.018260	0.000965	0.013876	-0.008467	-0.014089	0.097221	-0.110689	0.013035	-0.026770
Tipo de procedimento	0.022173	1.000000	0.119092	0.044010	-0.000741	0.000712	-0.004920	0.044633	0.018474	-0.041874	0.310148	-0.483406
Fundamentação	-0.079557	0.119092	1.000000	0.007909	0.000405	-0.038625	0.048748	-0.044245	0.058840	0.088367	0.092926	-0.098773
Preço contratual	0.018260	0.044010	0.007909	1.000000	0.000102	-0.001032	0.001865	-0.003481	0.098891	-0.003232	0.047612	0.038645
Prazo de execução	0.000965	-0.000741	0.000405	0.000102	1.000000	0.001503	-0.001714	-0.000642	0.005673	-0.001134	-0.000326	0.000450
Localidade	0.013876	0.000712	-0.038625	-0.001032	0.001503	1.000000	0.253276	-0.045684	0.004724	-0.011492	-0.003744	0.000564
Distrito	-0.008467	-0.004920	0.048748	0.001865	-0.001714	0.253276	1.000000	0.037586	0.007808	-0.002659	-0.009562	0.010186
Nif Adjudicante	-0.014089	0.044633	-0.044245	-0.003481	-0.000642	-0.045684	0.037586	1.000000	-0.010679	0.017193	0.050990	-0.042625
Irregular	0.097221	0.018474	0.058840	0.098891	0.005673	0.004724	0.007808	-0.010679	1.000000	-0.074208	0.051323	0.131626
CPVcat	-0.110689	-0.041874	0.088367	-0.003232	-0.001134	-0.011492	-0.002659	0.017193	-0.074208	1.000000	-0.034641	0.095216
NRConcorrentes	0.013035	0.310148	0.092926	0.047612	-0.000326	-0.003744	-0.009562	0.050990	0.051323	-0.034641	1.000000	-0.091043
redFlagCount	-0.026770	-0.483406	-0.098773	0.038645	0.000450	0.000564	0.010186	-0.042625	0.131626	0.095216	-0.091043	1.000000

Figura 5.11: Matriz de correlação das variáveis do *dataset*

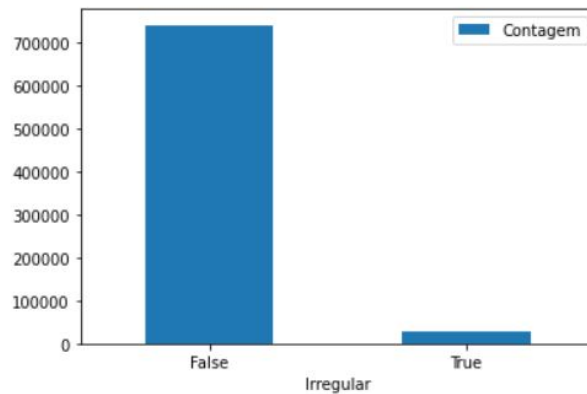
Após a análise da matriz de correlação foi utilizada a classe `selectKbest`, que pertence à biblioteca `Sklearn` e tem como objetivo indicar as melhores *features* para utilizar nos algoritmos de aprendizagem. Para tal, foi necessária a escolha de uma função para produzir os *scores* de cada *feature*, foi escolhida a função `f_classif` por ser indicada para *targets* categóricos.

Os *scores* produzidos pela aplicação do `selectKbest`, representados na figura 5.12, demonstram a existência de 6 *features* com *scores* bastante superiores às restantes, nomeadamente "redFlagCount", "Preço contratual", "Tipo(s) de contrato", "CPVcat", Fundamentação e "NRConcorrentes", pelo que foram apenas mantidas estas *features* e as restantes foram descartadas.

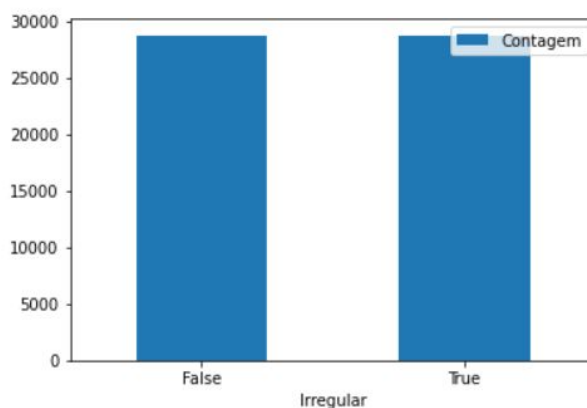
	Feature_Name	Score
11	redFlagCount	13583.828923
3	Preço contratual	7609.054253
0	Tipo(s) de contrato	7351.771167
9	CPVcat	4266.263840
2	Fundamentação	2676.692262
10	NRConcorrentes	2034.783318
1	Tipo de procedimento	263.038383
7	Nif Adjudicante	87.873565
6	Distrito	46.970556
4	Prazo de execução	24.796394
5	Localidade	17.195858
8	Nif Adjudicatario	0.038787

Figura 5.12: Ranqueamento de variáveis a partir da classe `selectKbest`

Após a escolha das variáveis com as quais o modelo treinará, foi analisado o balanceamento do *dataset* quanto ao número de contratos considerados pelo sistema de regras como irregulares e regulares, representado na figura 5.13. Foi encontrado um forte desbalanceamento entre estas duas classes, havendo 741697 contratos considerados regulares e apenas 28768 considerados irregulares, o que trouxe a necessidade de efetuar um balanceamento nos dados, visto que um desbalanceamento tão grande causaria um *bias* na previsão do modelo, dando-lhe sempre tendência a atribuir aos contratos de teste a classe que se encontra em maior número, neste caso, os contratos seriam maioritariamente ou totalmente classificados como regulares.

Figura 5.13: Balanceamento da classe *target*

A técnica de *sampling* escolhida para o *dataset* foi *undersampling*, visto que o *dataset* tem uma quantidade elevada de contratos, mesmo após serem descartados os contratos que causavam o desbalanceamento. Para realizar o *undersampling* foram escolhidos de forma randomizada 28768 contratos "regulares" para ser utilizados no modelo de previsão e descartados os restantes, tornando assim o *dataset* perfeitamente balanceado como se pode observar na figura 5.15.

Figura 5.14: Balanceamento após aplicação de *sampling*

Posteriormente foi realizado *feature scaling*, utilizando a classe *MinMaxScaler* da biblioteca SKlearn, que permite colocar variáveis do tipo numérico numa escala de valores entre 0 e 1 conforme a sua distribuição. Este passo é importante, na medida em que se os dados de diferentes *features* do *dataset* estiverem em diferentes escalas, a ordem de magnitude da variância entre as *features* diferirá, o que causa um *bias* na aprendizagem, e com implicação negativa nos resultados produzidos.

De modo a poder utilizar no modelo variáveis com valores nominais, algo que não é possível sem estas serem transformadas, foi utilizado o *LabelEncoder* da biblioteca SKlearn, que atribui a cada valor nominal um valor inteiro correspondente, convertendo assim os valores nominais de uma variável em inteiros, fazendo com que esses valores passem a ser suportados pelos algoritmos de *Machine Learning*.

	CPVcat	Tipo de procedimiento	Tipo(s) de contrato	Localidade	Distrito	
0	44		4	2	81	14
1	10		5	5	177	0
2	10		5	5	177	0
3	43		0	1	15	15
4	27		4	1	124	9

Figura 5.15: Variáveis discretas após aplicação de *label encoding*

Por fim foi efetuado o *train-test split* de modo a poder ter uma porção do *dataset* destinada a treino e outra destinada à avaliação da *performance* de classificação dos algoritmos. O rácio da divisão foi de 80% dos dados para treino e os restantes 20% para teste.

5.3.3 Conceção dos Modelos

Nesta subsecção serão demonstrados detalhadamente todos os passos seguidos durante a utilização de algoritmos de *Machine Learning* para a deteção de irregularidades em contratos públicos e justificadas as escolhas tomadas.

A abordagem proposta é a de inicialmente escolher vários algoritmos especializados em classificação presentes na biblioteca SKLearn e comparar os seus resultados, eliminar os algoritmos que produzem piores resultados, e posteriormente efetuar *tuning* nos hiperparâmetros dos algoritmos mantidos de modo a, se possível, otimizar a qualidade das classificações produzidas.

5.3.3.1 Algoritmos e métricas de *performance* escolhidas

De modo a escolher algoritmos mais adequados para a classificação dos contratos, foi feita uma pesquisa de artigos de fraudes de cartões de crédito devido à similaridade desse problema com o problema de classificação de contratos irregulares. Estes dois tipos de problema são similares na medida em que ambos contêm usualmente *datasets* altamente desbalanceados em que os contratos irregulares (no caso da contratação pública) e os cartões fraudulentos (no caso das fraudes de cartões de crédito) se encontram em grande minoria face à classe oposta. Outra similaridade é que ambos os problemas consistem prever uma classificação binária a partir de um conjunto de dados descritivos para cada entrada do *dataset*.

Os algoritmos de *Gradient Boosting*, *Random Forest*, *Logistic Regression* e SVM foram utilizados num artigo[65] com o objetivo de deteção de fraudes de cartão de crédito em um *dataset* de cartões de crédito europeus altamente desbalanceado, 492 cartões fraudulentos e 284315 cartões genuínos. Após ter sido efetuado *undersampling* no *dataset* estes algoritmos tiveram todos um *recall* superior a 91% e *precision* superior a 90%.

Noutro artigo com uma abordagem semelhante[43] ao artigo abordado no parágrafo anterior foram comparadas as métricas de *performance* de classificação de diversos algoritmos, entre os quais a maioria está disponível para utilização na biblioteca SKLearn, sendo eles *Random Forest*, *Gaussian Naive Bayes*,

Logistic Regression, SVM, Decision Tree, KNN e Multilayer Perceptron utilizando também um *dataset* de cartões de crédito europeus. Tal como no *paper* referido no parágrafo anterior também foi aplicado *undersampling* ao *dataset*, e no fim foram obtidos valores de *precision*, *recall* e *accuracy* acima de 90% para todos eles, sendo que o *Random Forest* foi o algoritmo que mais se destacou com aproximadamente 95% nas três métricas citadas.

Com base nos bons resultados de *performance* obtidos pelos algoritmos nos trabalhos citados, foram então utilizados estes algoritmos no *dataset* recolhido, com o objetivo de comparar as *performances* de cada um.

As funções da biblioteca SKlearn para implementação de cada algoritmo referido estão demonstradas em 5.6

```

1 classifiers = [
2     KNeighborsClassifier(),
3     RandomForestClassifier(),
4     AdaBoostClassifier(),
5     GaussianNB(),
6     LinearDiscriminantAnalysis(),
7     GradientBoostingClassifier(n_estimators=100, learning_rate=1.0, max_depth=1, random_state
      ↪ =0),
8     MLPClassifier(hidden_layer_sizes=(20))]
```

Listagem 5.6: Implementações dos algoritmos de *Machine Learning*

Para cada um dos algoritmos escolhidos foi efetuada a aprendizagem com o *dataset* de treino, e posterior classificação dos contratos de teste. Após a produção das classificações dos contratos por cada algoritmo, foram extraídas métricas de comparação da *performance* de cada modelo, nomeadamente *Accuracy*, *Precision*, *Recall* e F1-score. No código em 5.7 está representado o ciclo utilizado para calcular e imprimir as métricas para cada algoritmo.

```

1 for clf in classifiers:
2     clf.fit(X_train, y_train)
3     print(clf.__class__.__name__)
4     train_predictions = clf.predict(X_test)
5     print(classification_report(y_test, train_predictions))
6     acc = accuracy_score(y_test, train_predictions)
7     print("Accuracy: {:.4%}".format(acc))
8     train_predictions = clf.predict_proba(X_test)
9     lloss = log_loss(y_test, train_predictions)
10    print("Log Loss: {}".format(lloss))
```

Listagem 5.7: Cálculo de métricas de avaliação dos algoritmos

Na tabela 5.2 estão as fórmulas de cálculo das métricas de avaliação da *performance* de classificação dos algoritmos[28].

Nome	Função	Fórmula
<i>Accuracy</i>	$\frac{TP + TN}{TP + FP + TN + FN}$	Percentagem de valores previstos iguais aos valores reais
<i>Precision</i>	$\frac{TP}{TP + FP}$	De todos os contratos classificados com uma classe, percentagem de quantos pertencem realmente a essa classe
<i>Recall</i>	$\frac{TP}{TP + FN}$	De todos os contratos pertencentes a uma classe, percentagem dos que foram classificados corretamente
<i>F1-score</i>	$2 * \frac{Precision * Recall}{Precision + Recall}$	Diferença entre a distribuição dos valores previstos e os valores reais

Tabela 5.2: Métricas de performance

Os algoritmos de aprendizagem foram aplicados num *dataset* de teste com 11000 contratos, estando cada classe (regular ou irregular) representada por 5500 contratos. A intenção desta etapa foi comparar os resultados produzidos por cada um dos algoritmos, de forma a poder escolher os 3 algoritmos com melhores resultados, para serem posteriormente otimizados através de *tuning* de hiperparâmetros. O processo de comparação de resultados feito para escolher os melhores algoritmos foi realizado com base nas métricas de *performance* já apresentadas, e também no *log loss* que representa quão perto a previsão probabilística está do correspondente valor real binário de classificação.

Inicialmente, foi descartado o algoritmo de Naive Bayes devido a ter uma *accuracy* significativamente inferior aos restantes, com 50% de *accuracy* quando todos os restantes apresentaram valores acima dos 80%, esta falta de *accuracy* verificada deve-se provavelmente a um *overfit* desse algoritmo que lhe dá a tendência de classificar o *target* sempre como falso, o que pode ser verificado através da *recall* de 99% e 18% para as classes *False* e *True* respetivamente, o que significa que 99% das entradas com classe *False* foram classificadas corretamente e apenas 18% das entradas com classe *True* foram classificadas como tal. A *precision* de 54% para o *target* falso é também um indicador da presença de um *bias* forte na classificação, visto que isto significa que dos contratos classificados como irregulares, 46% foram incorretamente classificados.

Para os restantes algoritmos, visto que possuem valores de *F1-score* bastante próximos, a escolha dos melhores algoritmos foi feita com base na *accuracy* e *log loss*. Através das duas métricas de comparação escolhidas, foram escolhidos os 3 algoritmos que se destacam dos restantes por alta *accuracy* e baixo

log loss em relação aos restantes, nomeadamente *Multilayer Perceptron*, *Gradient Boosting* e *Random Forest*.

	Geral		True			False		
Nome	Accuracy	Loss	F1(T)	F1(F)	Precision(T)	Precision(F)	Recall(T)	Recall(F)
Random Forest	0.97	0.14	0.98	0.98	0.98	0.98	0.98	0.98
GaussianNB	0.57	0.87	0.30	0.69	0.93	0.54	0.18	0.99
KNN	0.94	0.66	0.95	0.94	0.94	0.95	0.95	0.94
MLP	0.97	0.09	0.97	0.97	0.95	0.99	0.99	0.94
AdaBoost	0.95	0.55	0.95	0.95	0.92	0.98	0.99	0.91
Decision Tree	0.97	0.87	0.98	0.97	0.98	0.97	0.97	0.98
GradientBoosting	0.98	0.07	0.98	0.97	0.96	0.99	0.99	0.96
LDA	0.82	0.46	0.82	0.81	0.82	0.81	0.82	0.82

Tabela 5.3: Resultados do teste de *performance* dos algoritmos

5.3.3.2 *Tuning* de Hiperparâmetros

Hiperparâmetros são fatores importantes nos algoritmos de *Machine Learning*, pois permitem controlar diretamente o comportamento do algoritmo e afetam significativamente a *performance* do modelo[54].

Tuning de hiperparâmetros é um processo que tem a finalidade de encontrar os valores ótimos para os hiperparâmetros de um dado algoritmo de *Machine Learning*, este processo quando aplicado de forma eficiente permite melhorar de forma relevante a eficiência de aprendizagem do algoritmo[54].

Foi tomada a decisão de fazer o processo de *tuning* dos hiperparâmetros para tentar maximizar as métricas de *performance* dos algoritmos.

A ferramenta escolhida para efetuar o processo de *tuning* foi a classe GridSearchCV da biblioteca SKlearn, esta ferramenta através do algoritmo e da lista de valores possíveis para os parâmetros, testa automaticamente todas as combinações possíveis de valores para os hiperparâmetros e devolve a combinação que produz a maior *accuracy*. Nos parágrafos seguintes serão descritos os hiperparâmetros utilizados e os valores tomados por cada um para cada algoritmo seguidos da comparação dos resultados obtidos antes e após o *tuning*.

Multilayer Perceptron

Tabela 5.4: Hiperparâmetros utilizados no *tuning GridSearch*

Hiperparâmetro	Descrição	Valores testados	Valor escolhido
<i>Activation</i>	Função de ativação das camadas escondidas	ReLu e Tanh	Tanh
<i>Solver</i>	Algoritmo de otimização dos pesos	Adam e SGD	Adam
Alpha	Penalização para pesos excessivos	10^{-n} para todo n inteiro entre 1 e 10	0.0001
Learning_Rate	Controla a taxa de ajuste dos pesos do algoritmo	<i>constant</i> , <i>invscaling</i> e <i>adaptive</i>	<i>Constant</i>
Max_iter	Máximo de iterações do algoritmo até convergir	200, 500 e 1000	1000
hidden_layer_sizes	Número de neurónios das camadas escondidas da rede neuronal	(10,30,10), (20) e (15,15)	(10,30,10)

	Geral		True			False		
Nome	Accuracy	Loss	F1(T)	F1(F)	Precision(T)	Precision(F)	Recall(T)	Recall(F)
Antes do <i>Tuning</i>	0.97	0.09	0.97	0.97	0.95	0.99	0.99	0.94
Após <i>Tuning</i>	0.98	0.08	0.98	0.97	0.96	0.99	0.99	0.96
Diferença antes/após <i>tuning</i>	0.01	-0.01	0.01	0.00	0.01	0.00	0.00	0.02

Tabela 5.5: Resultados do teste de *performance* do algoritmo MLP**Random Forest**

Hiperparâmetro	Descrição	Valores testados	Valor escolhido
max_depth	Profundidade máxima das árvores	3, 4, 5, 6, 7 e 8	8
max_features	Número de <i>features</i> a considerar	auto, sqrt e log2	auto
n_estimators	Número de árvores da floresta	200, 400, 600 e 800	200
criterion	Crítério de divisão das árvores de decisão	gini e entropia	entropia

Tabela 5.6: Hiperparâmetros utilizados no *tuning GridSearch*

	Geral		True			False		
Nome	Accuracy	Loss	F1(T)	F1(F)	Precision(T)	Precision(F)	Recall(T)	Recall(F)
Antes do <i>Tuning</i>	0.97	0.14	0.98	0.98	0.98	0.98	0.98	0.98
Após <i>Tuning</i>	0.99	0.06	0.98	0.98	0.98	0.99	0.99	0.98
Diferença antes/após <i>tuning</i>	0.02	-0.08	0.00	0.00	0.00	0.01	0.01	0.00

Tabela 5.7: Resultados do teste de *performance* do algoritmo *Random Forest***Gradient Boosting**

Hiperparâmetro	Descrição	Valores testados	Valor escolhido
max_depth	Profundidade máxima das árvores	3, 4, 5, 6, 7, 8	8
max_features	Número de <i>features</i> a considerar	sqrt e log2	sqrt
n_estimators	Número de árvores da floresta	100, 200 e 400	100
criterion	Critério de divisão	mse e friedman_mse	friedman_mse
loss	Função de <i>loss</i>	deviance e exponential	exponential
min_samples_leaf	Número mínimo de samples necessários para estar nas folhas de uma árvore	1, 2, 3, 4, 5, 6	5
min_samples_split	Número mínimo de samples necessários para dividir um nó de uma árvore	2, 3, 4, 5, 6 e 7	2
learning_rate	Controla a taxa de ajuste dos pesos do algoritmo	0.1, 0.05, 0.01	0.1

Tabela 5.8: Hiperparâmetros utilizados no *tuning GridSearch*

	Geral		True			False		
Nome	Accuracy	Loss	F1(T)	F1(F)	Precision(T)	Precision(F)	Recall(T)	Recall(F)
Antes do <i>Tuning</i>	0.98	0.07	0.98	0.97	0.96	0.99	0.99	0.96
Após <i>Tuning</i>	0.99	0.05	0.99	0.98	0.98	0.99	0.99	0.98
Diferença antes/após <i>tuning</i>	0.01	-0.02	0.01	0.02	0.00	0.00	0.00	0.02

Tabela 5.9: Resultados do teste de *performance* do algoritmo *Gradient Boosting*

5.3.4 Análise dos resultados obtidos

Após a testagem do modelo, os resultados obtidos foram bastante satisfatórios no que toca à capacidade de previsão dos contratos considerados irregulares. Uma grande parte dos algoritmos utilizados foi capaz de prever com uma *accuracy* superior a 90% e com elevados valores de *precision* e *recall* irregularidades em contratos que não se encontravam presentes no *dataset* de treino.

Apesar das métricas de *performance* iniciais já serem bastante elevadas, foi feito *tuning* de hiperparâmetros nos três algoritmos que produziram melhores resultados, *Random Forest*, *Gradient Boosting* e *Multilayer Perceptron*. Após o *tuning*, todos os algoritmos apresentaram *accuracy* mais alta e *loss* mais baixa, tendo o *Gradient Boosting* e *Random Forest* alcançado 99% de *accuracy*. Nas tabelas

Por fim, este modelo de detecção de irregularidades produziu resultados bons no contexto do problema, tendo alguns dos algoritmos uma taxa de acerto muito próxima de 100%, o que leva a crer alguns destes algoritmos de *Machine Learning*, principalmente os de *Random Forest*, *Gradient Boosting* e *Multilayer Perceptron*, podem ser usados como fonte confiável de detecção de irregularidades a partir de um conjunto de dados previamente classificados.

5.4 Utilização de *Machine Learning* para previsão de gastos em contratos públicos

Devido a esta dissertação ter como objetivo a extração de conhecimento que possa facilitar e/ou otimizar o processo de contratação pública em Portugal, e também devido ao estudo de técnicas e paradigmas de *Machine Learning* feitos durante a análise do estado da arte, foi analisada a hipótese da utilização de técnicas de *Machine Learning* no campo de *time series forecasting* para conseguir prever gastos futuros associados a um certo CPV, ou a uma certa localidade.

Para isto, nesta secção será abordada a criação de um modelo de previsão de custos, que tem como objetivo auxiliar as entidades responsáveis pela contratação pública, como por exemplo, câmaras municipais ou comunidades intermunicipais, a conseguir efetuar uma previsão dos custos futuros numa certa área de contratos (e.g. educação, saúde, etc.), recorrendo aos custos presentes nos contratos celebrados no passado.

5.4.1 Preparação de dados

A preparação de dados para a conceção dos modelos de previsão de gastos consistiu nos passos seguintes:

- Escolha do *subset*
- Eliminação de *outliers*
- *Scaling* dos dados
- *Train-test split*

Para a fase de escolha do *subset* foram selecionados apenas os dados referentes a contratos com o CPV pertencente à categoria de aquisição de serviços e material médico, categoria esta que foi escolhida devido a ser a uma das categorias de CPV com mais contratos no *dataset*, o que providencia um *subset* com um elevado número de observações.

De seguida, a partir do *subset* escolhido foram agrupados os contratos pelo somatório dos gastos efetuados nessa área no país inteiro para cada semana do ano.

A organização dos dados para estes poderem ser utilizados para o propósito de previsão de custos recorrendo a técnicas de *Machine Learning* consiste na criação de dois *arrays*, um que contém o somatório do valor despendido em contratos durante uma certa semana e será a variável Y do problema, e outro que contém os valores despendidos nas N semanas anteriores e será a variável X do problema. Por exemplo, para o caso de um mês aleatório do *dataset*, considerando um *time step* de 30 semanas, o que seria utilizado como variável Y no algoritmo seria o valor total gasto no presente mês em contratos do CPV escolhido, e o valor X seria uma lista com os valores gastos em cada uma das 30 semanas anteriores.

De modo a eliminar possíveis meses com valores fora do comum, foi utilizada o algoritmo de *isolation forest*, que através dos valores despendidos em cada semana, sinaliza os *outliers* presentes. A principal

causa para serem eliminados os *outliers* foi para que o modelo não sofresse impacto na aprendizagem causado pelo período em que Portugal esteve numa situação pandémica agravada.

5.4.2 Algoritmos e métricas de *performance* escolhidas

O objetivo desta subsecção é justificar as escolhas dos algoritmos utilizados para efetuar as previsões. Para tal teve de ser analisado inicialmente o contexto do problema e a natureza dos dados, e em seguida a seleção de algoritmos utilizados em problemas e tipos de dados semelhantes na literatura disponível. Visto que o objetivo do problema é prever unicamente os gastos para o futuro a partir de valores de gastos anteriores, os algoritmos a ser utilizados têm de ser de algoritmos fortes a lidar com problemas de sequências numéricas. Neste âmbito foi decidido o uso de LSTM e GRU visto que por definição estas são redes com capacidades de memorizar padrões de longo termo em sequências numéricas e que se destacam dos restantes algoritmos em problemas de previsão[22].

Após a escolha do tipo de redes a utilizar para a de previsão de resultados, e devido à sua elevada complexidade, foi decidido implementar também outros algoritmos de regressão mais simples e de fácil implementação, apenas como forma de *benchmarking*, de modo a poder utilizar as suas métricas de *performance* como método de comparação com as abordagens LSTM e GRU. Foram escolhidos XGBoost e Random Forest Regressor por serem dois algoritmos de regressão presentes na biblioteca Keras capazes de lidar com sequências de dados como *input*. Estes algoritmos foram apenas usados pela sua simplicidade de implementação de modo a gerar valores de referência para poderem ser comparados com as implementações de *Deep Learning*, e a partir dessa comparação, tentar melhorar os resultados produzidos.

Na tabela 5.10 está o nome e uma descrição do método de cálculo de cada uma das métricas de *performance* utilizadas para comparação dos resultados produzidos por cada abordagem.

Nome	Descrição
<i>Mean Average Percentage Error</i>	Média das percentagens de erro entre os valores previstos e valores reais
<i>Mean Absolute Error</i>	Média do valor absoluto da diferença entre a distribuição dos valores previstos e os valores reais
<i>Root Mean Square Error</i>	Raiz quadrada da diferença média entre a distribuição dos valores previstos e os valores reais

Tabela 5.10: Métricas de *performance* de regressão utilizadas

5.4.3 Modelos de *Machine Learning*

Nesta subsecção serão abordados os modelos utilizados para previsão de gastos monetários futuros em contratos públicos de diversos setores. Para cada um dos algoritmos utilizados é explicada a sua implementação e demonstrado o código utilizado na sua criação. Inicialmente é abordada a arquitetura

de LSTM, que contou com 2 abordagens distintas, uma com utilização de camadas de *dropout* e outra sem utilização de *dropout*.

De seguida serão abordados os algoritmos de XGBoost e *Random Forest Regressor*, utilizados devido à sua simples implementação de modo a poder usar os seus resultados como termo de comparação com as abordagens de *Deep Learning*, com a finalidade de verificar se as redes neuronais estão a produzir resultados superiores, inferiores ou equiparáveis aos produzidos por algoritmos de aprendizagem supervisionada convencionais no campo das séries temporais.

5.4.3.1 LSTM e GRU

Devido à grande complexidade das redes neuronais e do grande número de alterações que podem ser feitas às estruturas das redes LSTM e GRU de modo a otimizar a sua capacidade aprendizagem, foram realizadas experimentações preliminares com vista a escolher um número de neurónios, de camadas e de percentagem de *dropout* que produzissem melhor *performance* tendo em conta as métricas escolhidas para a fase de avaliação. Para tal experimentação ser possível foi criada uma função capaz de receber como atributo o número de camadas, neurónios e *epochs* desejados e a percentagem de *dropout* desejada, essa função criará uma estrutura comum para diversas implementações em que só variam estes atributos citados anteriormente, permitindo assim comparar o efeito das variações dos atributos no resultado final, escolhendo assim de entre uma série de combinações possíveis a combinação mais favorável em termos de *performance*.

As funções de implementação das redes LSTM e GRU estão em 5.8 e 5.9, respetivamente, onde pode ser verificado que as estruturas são idênticas para facilitar a comparação de resultados obtidos entre abordagens, e apenas são variadas os tipos de RNN utilizados.

```

1 def lstm_network(neurons, lstm_layers, dropout, epochs):
2     model=Sequential()
3     model.add(LSTM(neurons, return_sequences=True, input_shape= (x_train.shape[1], 1)))
4     model.add(Dropout(dropout))
5     for i in range(0,lstm_layers-2):
6         model.add(LSTM(int(neurons/2), return_sequences=True))
7         model.add(Dropout(dropout))
8     model.add(LSTM(neurons, return_sequences=False))
9     model.add(Dropout(dropout))
10    model.add(Dense(1))
11    model.compile(optimizer = 'adam',loss = 'mean_squared_error')
12    history=model.fit(x_train, y_train,epochs=epochs,batch_size=2,validation_data=(x_test,
    ↪ y_test),callbacks=callbacks)

```

Listagem 5.8: Função de implementação da rede LSTM

```

1 def lstm_network(neurons, lstm_layers, dropout, epochs):
2     model=Sequential()
3     model.add(GRU(neurons, return_sequences=True, input_shape= (x_train.shape[1], 1)))

```



```

4     model.add(Dropout(dropout))
5     for i in range(0, lstm_layers-2):
6         model.add(GRU(int(neurons/2), return_sequences=True))
7         model.add(Dropout(dropout))
8     model.add(GRU(neurons, return_sequences=False))
9     model.add(Dropout(dropout))
10    model.add(Dense(1))
11    model.compile(optimizer = 'adam', loss = 'mean_squared_error')
12    history=model.fit(x_train, y_train, epochs=epochs, batch_size=2, validation_data=(x_test,
    ↪ y_test), callbacks=callbacks)

```

Listagem 5.9: Função de implementação da rede GRU

5.4.3.2 XGBoost e Random Forest Regressor

De seguida serão demonstradas as implementações utilizadas nos algoritmos de XGBoost e *Random Forest*.

A criação destes dois algoritmos tal como previsto foi de um nível de dificuldade bastante inferior ao LSTM e GRU visto que o único hiperparâmetro que foi ajustado foi o *max depth* no caso do *Random Forest* e os parâmetros *boost* e *objective* no caso do algoritmo XGBoost, o código de implementação de cada algoritmo está representado em 5.10.

```

1  XGB = XGBRegressor(booster="gblinear", objective="reg:squarederror").fit(X_train, Y_train)
2  RF=RandomForestRegressor(max_depth=2, random_state=0).fit(X_train, y_train)

```

Listagem 5.10: Implementações de XGBoost e Random Forest Regressor

5.4.4 Análise dos resultados obtidos

Antes de analisar os resultados obtidos quanto às métricas escolhidas, é importante referir que durante a fase de teste dos modelos as métricas de *validation loss* e *training loss* apresentadas durante a compilação dos modelos tiveram um papel fulcral na calibração dos parâmetros e estrutura dos modelos, sendo que uma grande discrepância entre os valores destas métricas foi uma forma de perceber que teriam de ser feitas mudanças na estrutura da rede, no caso da *validation loss* ter um valor muito superior à *training loss* foi considerado como um indicador de *overfitting*, e no caso contrário foi considerado indicador de *underfitting*. Um problema frequente que surgiu durante a implementação dos modelos foi a estagnação da aprendizagem do modelo, detetado a partir da constância da *validation loss* e *training loss* em diversas *epochs* seguidas. Um método encontrado para contrariar esta estagnação foi a utilização de uma função de *callback*, chamada *ReduceLROnPlateau*, que permite atualizar a *learning rate* do algoritmo, multiplicando-a por um fator de multiplicação escolhido previamente sempre que a aprendizagem não apresente melhorias durante um certo número de *epochs*.

Durante a implementação das redes de LSTM e GRU, foram testadas diversas combinações de número de camadas, número de neurónios e valores de *dropout* diferentes de modo a ser possível comparar os efeitos da variação desses parâmetros e escolher a combinação de parâmetros que produzam resultados mais satisfatórios. Os resultados dos testes efetuados encontram-se nas tabelas 5.11 e 5.12.

Neurons	Layers	Dropout	MAE	RMSE	MAPE
64	3	0%	28124839	21122076	17.36%
128	3	0%	28259684	21158101	17.52%
256	3	0%	28533906	21597947	17.90%
128	3	10%	29309004	22215952	18.15%
128	3	20%	29351303	22335083	18.38%
128	3	30%	28797050	22004696	18.30%
128	4	0%	28999147	22162583	18.26%
128	5	0%	29031411	22262288	18.32%
128	6	0%	28692671	22103386	18.45%

Tabela 5.11: Resultados do teste de *performance* dos algoritmos LSTM

Neurons	Layers	Dropout	MAE	RMSE	MAPE
64	3	0%	29064632	22305166	18.70%
128	3	0%	28340447	22308803	19.74%
256	3	0%	29588270	22512515	18.77%
128	3	10%	28908728	22152797	18.52%
128	3	20%	28938361	22525578	19.42%
128	3	30%	30200512	22890569	18.75%
128	4	0%	28312003	22083148	18.13%
128	5	0%	29233127	22285157	18.43%
128	6	0%	28848392	21891951	18.59%

Tabela 5.12: Resultados do teste de *performance* dos algoritmos GRU

Como pode ser verificado na tabela 5.11, a combinação de maior *performance* produzida foi coincidentemente também a rede teoricamente mais simples de todas as experimentadas, rede essa que utilizou como parâmetros da função de criação 64 neurónios 3 camadas e inexistência de *dropout*. Esta rede obteve os valores mais baixos para as métricas MAE, RMSE e MAPE entre todas as abordagens LSTM apresentadas.

Uma das conclusões tiradas durante a fase de experimentação com as redes LSTM foi que aumentar o número de camadas e de neurónios pode ter efeitos negativos na *performance* da rede, como pode ser verificado, por exemplo, nas três primeiras entradas, em que para o mesmo número de camadas, o aumento do número de neurónios de 64 para 128 e 256 traduziu-se num decréscimo da MAPE e aumento da MAE e RMSE.

Foi testada a utilização de camadas de *dropout* de modo a contrariar o *overfitting* através da introdução de ruído em algumas das redes testadas, mas, com a sua introdução a *performance* de previsão foi pior que em redes bastante mais simples.

No caso das implementações de redes GRU, como pode ser verificado na tabela 5.12, a abordagem que obteve as melhores métricas de *performance* foi a abordagem com os parâmetros de criação de 0% de *dropout*, 128 neurónios e 4 camadas. Ou seja, a rede é constituída por 4 camadas GRU seguidas de uma *Dense*, sendo a primeira e última GRU de 128 neurónios e as intermédias de 64 neurónios. Apesar de a abordagem anterior ter sido a melhor em termos de *performance* das abordagens GRU, esta abordagem teve piores métricas *performance* do que a melhor abordagem LSTM. Na figura 5.16 pode ser verificada a diferença entre os valores reais e os valores previstos pelo melhor modelo de LSTM. Através da figura é possível perceber que o modelo, apesar de ter apresentado um erro médio percentual de 17,36%, tem dificuldade conseguir valores próximos dos valores reais quando existem variações acentuadas nos valores, o que pode ser um entrave para a sua aplicabilidade como meio de previsão no mundo real. Uma das causas para a incapacidade do modelo para prever picos é possivelmente o número reduzido de casos de treino utilizados, o que faz com que as redes se deparem com poucos picos de gastos durante o treino e consequentemente sejam incapazes de os prever na fase de teste, pelo que futuramente deverá ser reimplementado com uso de um *dataset* maior de modo a verificar se este problema foi mitigado.

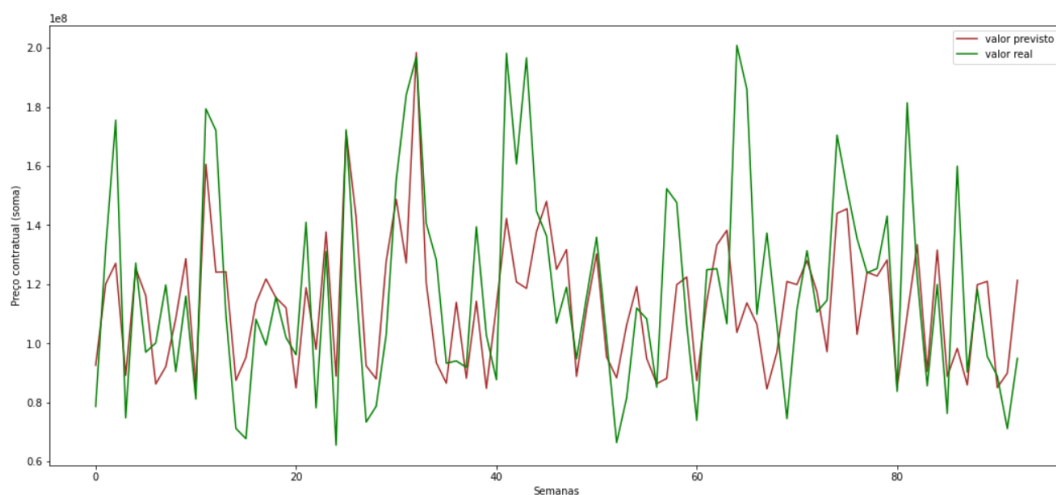


Figura 5.16: Comparação dos valores reais com os valores previstos pela melhor abordagem testada

Em suma, foram utilizadas diversas implementações de GRU e LSTM, bem como duas implementações de XGBoost e Random Forest, de modo a tentar prever o valor despendido em contratação pública numa semana com base nas despesas das 20 semanas anteriores. O modelo que conseguiu melhores métricas de previsão foi o LSTM, tendo conseguido na melhor abordagem um erro médio de 17,3%. Apesar de o erro médio não ser excessivamente alto tendo em conta a enorme imprevisibilidade destes dados em que ocorrem muitos picos de despesas, os modelos viram-se incapazes de prever a maioria das subidas acentuadas, e, por essa razão não podem ser considerados métodos confiáveis de previsão.

5.5 Interface de visualização de dados

Como foi extraída uma grande quantidade de contratos, e pela área da contratação pública se tratar de uma área na qual os erros trazem prejuízos muito avultados, foi tomada a decisão de criar uma plataforma simples que permita auxiliar o utilizador na atribuição de contratos.

O objetivo inicial foi criar um protótipo de uma plataforma que no futuro pudesse auxiliar as entidades responsáveis pela feitura dos contratos públicos, permitindo por um lado o acesso através da *interface* ao algoritmo de classificação de irregularidades, e por outro o acesso a dados estatísticos divididos por localidades e regiões do país. Este acesso a estatísticas permite, por exemplo, que antes da feitura de um contrato, o utilizador verifique o preço médio de contratos com o mesmo código CPV nas regiões vizinhas de modo a otimizar as despesas.

Nesta secção será abordado o processo de criação da *interface* de visualização dos dados de contratação pública nacional.

O *dataset* utilizado para visualização nesta *interface* foi o mesmo que foi obtido na fase de captura de dados.

A *framework* utilizada para a criação dos *dashboards* tem como nome *Dash*, e é uma *framework* de Python utilizada para *deploy* de aplicações de *Machine Learning* e *Data Science*. Uma das grandes vantagens desta *framework* é a utilização da mesma linguagem na qual foi feita a fase de análise dos dados, o que permite o reaproveitamento de alguns gráficos feitos durante essa fase.

A conceção da plataforma de visualização dos dados foi dividida em 4 partes distintas, uma de visualização através de um mapa interativo, outra que contém um *overview* acerca de dados considerados importantes no contexto da contratação pública nacional, outra que permite escolher uma localidade e mostra dados de relevo acerca da contratação nessa localidade específica, e finalmente uma aba que permite inserir dados de um contrato e que corre o algoritmo de *Machine Learning* de deteção de irregularidades e mostra a classificação produzida pelo algoritmo quanto à regularidade do contrato inserido.

5.5.1 Visualização geográfica da contratação

Na aba de visualização geográfica da contratação foi utilizado um mapa interativo que permite saber os gastos totais de cada distrito de Portugal em contratos públicos. Esta aba contém também filtros que permitem filtrar os contratos apresentados segundo diversos fatores, permitindo nomeadamente:

- Escolher as datas inicial e final entre as quais os contratos foram realizados
- Escolher o tipo de procedimento e tipo de contrato presente no contrato
- Escolher o formato de apresentação dos custos (per capita ou total)

Para a criação do mapa interativo foi utilizado um ficheiro Geojson disponibilizado pelo portal Dados.gov e que contém as delimitações dos distritos de Portugal. Este ficheiro é lido pela função `choropleth_mapbox` da biblioteca `plotly express` criando assim um mapa coroplético como pode ser visto na figura 5.17.

Este tipo de apresentação de valores foi escolhido devido à facilidade de comparar valores de distritos distintos através da escala de cores e devido a conferir uma maior facilidade de análise regional da despesa de contratação pública em Portugal.

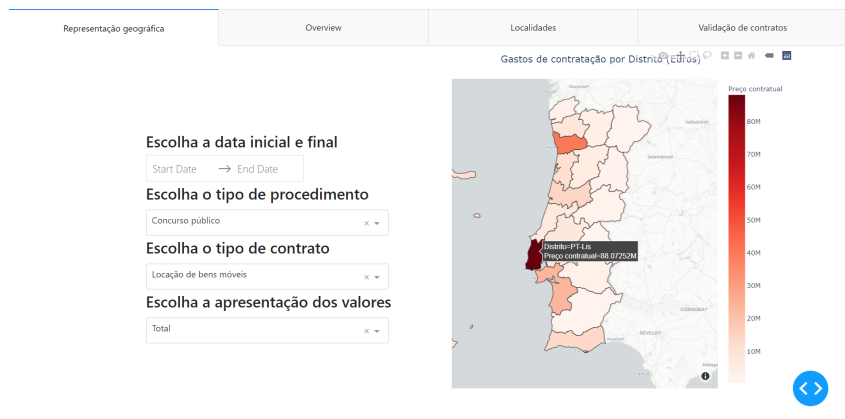


Figura 5.17: Aba de visualização geográfica

5.5.2 Visualização gráfica de dados

Para a visualização gráfica de dados foram criadas duas abas na *interface*, com a finalidade de conferir a qualquer pessoa uma fácil análise de temas como a distribuição dos gastos e das empresas com mais dinheiro recebido e despendido no país inteiro ou também em cada localidade específica do país.

As duas abas contêm informação semelhante, com a diferença que uma delas contêm gráficos gerais da contratação em Portugal como um todo, e a outra permite escolher uma localidade e apresenta posteriormente diversos gráficos de relevo acerca da contratação nessa mesma localidade. Ambas contêm gráficos idênticos, mas com especificidades diferentes, entre os gráficos contidos estão os que podem ser vistos nas figuras 5.18 e 5.19.

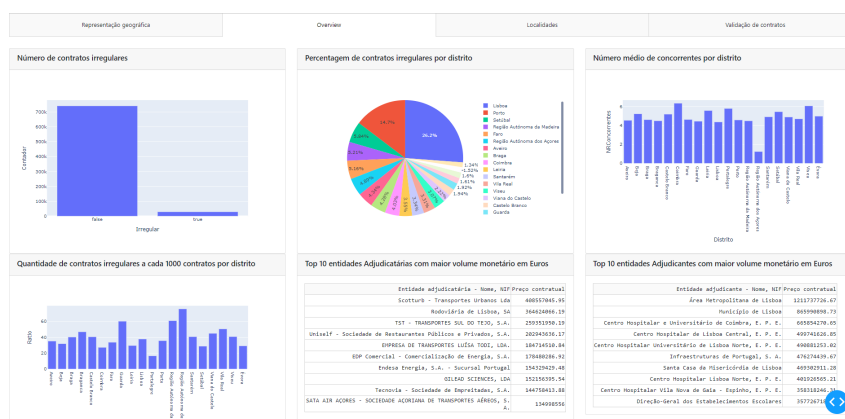


Figura 5.18: Aba de gráficos gerais do país

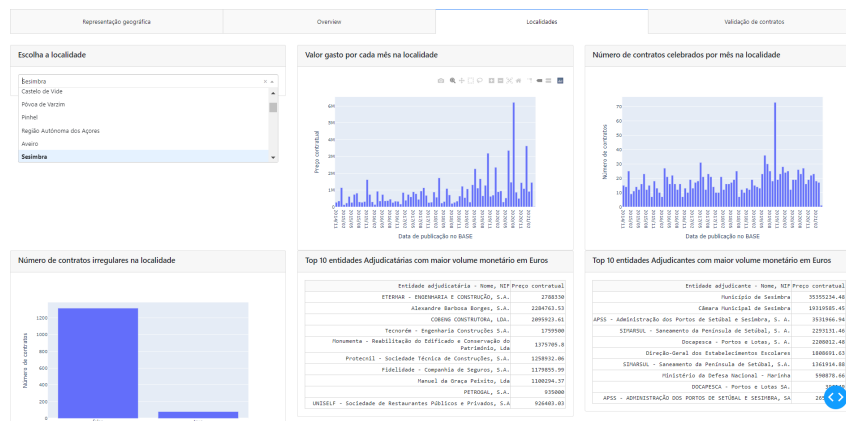


Figura 5.19: Aba de gráficos de uma localidade específica

5.5.3 Teste de irregularidade de contratos

Por último, a plataforma contém uma aba na qual pode ser testada a irregularidade de um contrato segundo o algoritmo de *Gradient Boosting* que foi o que deu melhores resultados na fase de testagem. O utilizador deve inserir o tipo de contrato, o tipo de procedimento, o valor do contrato, e a artigo do CCP de fundamentação do contrato. Estes valores são passados ao algoritmo que já tendo sido treinado previamente em *background* com o *dataset* vai devolver uma classificação para o contrato inserido.

Esta aba tem como finalidade permitir ao utilizador a verificação dos contratos antes da sua feitura, de modo a que fique alerta para uma possível irregularidade. Com isto, o utilizador não necessita de correr o algoritmo manualmente com o *dataset* e com os dados do contrato em questão, beneficiando assim de um dispêndio de tempo mais reduzido em relação a essa abordagem manual.

Para isto foi criada uma página simples, composta por um *form* em que o utilizador insere os dados, e um botão que procederá a enviar os dados inseridos para o algoritmo que corre em *background* e que devolve num balão de resposta a classificação obtida acerca do contrato inserido.

Conclusão e trabalho futuro

Neste capítulo serão revistos os resultados obtidos e verificado se eles correspondem às expectativas inicialmente propostas nos objetivos desta dissertação. De seguida será abordado o que poderia ser feito no futuro de modo a melhorar o trabalho realizado e a sua utilidade.

6.1 Conclusão

A literatura existente na área da deteção de irregularidades e conluio em contratos públicos é muito escassa e com poucas implementações práticas, devido à falta de artigos na comunidade científica com objetivos semelhantes a esta dissertação, o principal trabalho desta dissertação passou pela experimentação e avaliação da utilização de diversas técnicas de aplicação de Inteligência Artificial a *datasets* de contratação pública.

Inicialmente, foi testada a utilização de análise de grafos como forma de detetar conluio em contratos públicos, para isto foi aplicado um leque de algoritmos de análise de grafos de centralidade e de comunidade. As comunidades detetadas pelos algoritmos tinham tamanhos muito díspares, tendo algumas várias centenas de empresas e outras apenas duas empresas, foi também detetado que o algoritmo de deteção de comunidades considerava por vezes como comunidade conjuntos de empresas com um reduzido número de contratos celebrados entre si e de áreas distintas, pelo que foi considerado que análise de grafos não iria ser utilizada como forma de deteção de conluio.

Quanto à questão de investigação sobre a possibilidade de detetar irregularidades e indicadores de conluio nos dados, a resposta divide-se em dois tópicos. No âmbito da deteção de irregularidades o problema era mais literal, pois as irregularidades dependem apenas do cumprimento total das regras expressas no CCP, como tal foram codificadas as regras do CCP que envolvem condições numéricas num sistema de regras que foi capaz de identificar irregularidades nos contratos do *dataset*. Para além disso, foi criado um modelo de *Machine Learning* capaz de identificar com 99% de eficácia os contratos classificados

como irregulares pelo sistema de regras. Este modelo pode constituir um auxílio importante na construção de um sistema de apoio à decisão no futuro de modo a alertar a possível presença irregularidades através de um número reduzido de atributos.

Na área da detecção de conluio, foram extraídas *red flags* a partir de artigos da AdC e do OCDS, que são consideradas por essas entidades como fatores de suspeição de práticas de conluio em contratos públicos. A cada contrato foi atribuído um fator de suspeição constituído pelo número de *red flags* por ele acionadas. Esta abordagem foi capaz de sinalizar contratos de uma série de empresas acusadas publicamente de conluio, mas ainda carece de uma avaliação com uma lista maior de casos comprovados de conluio, de modo a validar a sua ação. Este sistema de sinalização carece também de um *dataset* com alguns dados que permitiriam implementar uma maior quantidade de indicadores de conluio da AdC, dados esses que atualmente não são divulgados pelo portal Base (e.g valores das propostas perdedoras).

No que diz respeito à segunda questão de investigação, sobre a possibilidade de prever gastos futuros, foram utilizadas duas implementações de redes *Deep Learning*, uma delas com a utilização de redes LSTM e outra com utilização de GRUs, e duas implementações de algoritmos de *supervised learning*, nomeadamente XGBoost e Random Forest. A melhor *performance* de previsão foi obtida numa implementação LSTM, e teve um erro médio de 17,36% do valor real. Por outro lado, todas as abordagens foram incapazes de prever subidas e descidas acentuadas nos valores.

Em suma, os objetivos delineados foram alcançados, visto que foi construído um *pipeline* capaz de extrair conhecimento com recurso a diversas técnicas de inteligência artificial. Por outro lado, a abordagem implementada de procura de irregularidades através de análise de grafos de conhecimento não teve sucesso, apresentando resultados incoerentes e ambíguos face aos resultados previstos.

Para culminar este projeto, foi submetido um artigo em coautoria na *23rd International Conference on Intelligent Data Engineering and Automated Learning*, artigo esse em que é abordado, entre outros temas, o trabalho desenvolvido durante esta dissertação.

6.2 Trabalho futuro

Inicialmente, algo que poderia melhorar de forma significativa os resultados obtidos neste trabalho na parte da detecção do conluio, e com o qual seria benéfico reconstruir o modelo de treino no futuro, seria a obtenção de informações mais completas acerca dos contratos e das empresas intervenientes na contratação pública. Informações como morada, nome dos sócios (caso existam), números de telefone e email estavam presentes em alguns dos artigos analisados sobre detecção de conluio em contratação pública onde foram considerados como indicadores de conluio, contribuindo assim para fortalecer a capacidade de detecção de conluio dos modelos. O acesso a estes dados das empresas tornava possível detetar, por exemplo, contratos em que duas das empresas concorrentes partilham a mesma morada ou o mesmo número de telefone, o que foi considerado uma prática suspeita num dos artigos analisados no estado da arte. Caso seja possível o acesso a estes dados mais completos acerca dos contratos, deverá ser feita uma reavaliação dos resultados obtidos, e, se necessário, reimplementação do *pipeline*,

principalmente no campo da análise de grafos de conhecimento, que não teve sucesso em produzir resultados relevantes durante esta dissertação.

Futuramente, seria importante também a reavaliação do modelo de previsão de custos utilizando um conjunto de dados de treino muito maior, para perceber se o problema da dificuldade de previsão nas oscilações acentuadas dos valores existentes durante a fase de análise de resultados teve como causa o reduzido número de dados de treino, o que trará novas conclusões sobre a aplicabilidade deste modelo como método confiável de previsão de custos futuros.

Outro trabalho a ser realizado é o desenvolvimento de um sistema de sugestões que auxilie um utilizador durante a inserção de dados, sistema esse que funcionaria com recurso ao algoritmo de *Machine Learning* já implementado neste trabalho. Este sistema serviria para que durante a inserção dados de um contrato na *interface* por parte de um funcionário da área da contratação pública, fossem corridos os algoritmos de deteção de irregularidades e de *red flags*, avisando-o acerca de possíveis suspeições acerca de irregularidades do contrato.

Por fim, será também necessário modificar a implementação do algoritmo de recolha de dados a partir do portal Base caso seja necessário fazer uma nova captura de dados, visto que durante o período de escrita desta dissertação a interface do portal foi totalmente remodelada, o que leva a que o algoritmo inicialmente implementado já não funcione para capturar contratos atualmente.

Bibliografia

- [1] W. J. Frawley, G. Piatetsky-Shapiro e C. J. Matheus. *Knowledge Discovery in Databases: An Overview*. AI Magazine, vol.13(3), p. 57. 1992.
- [2] U. Fayyad, G. Piatetsky-Shapiro e P. Smyth. *From Data Mining to Knowledge Discovery in Databases*. AI Magazine, vol.17(3), p. 37. 1996.
- [3] L. Zhang. *Knowledge graph theory and structural parsing*. 2002.
- [4] R. Mihalcea, P. Tarau e E. Figa. *PageRank on Semantic Networks, with Application to Word Sense Disambiguation*. 2004.
- [5] A. Overbay e J. W. Osborne. *The Power of Outliers (and Why Researchers Should Always Check for Them)*. Pract. Assess. Res. Eval, vol. 9. 2004.
- [6] L. Rokach e O. Maimon. *Data Mining with Decision Trees*. Series in Machine Perception and Artificial Intelligence vol. 69, p.264. 2007.
- [7] O. Fercoq, M. Akian, M. Bouhtou e S. Gaubert. *Ergodic Control and Polyhedral Approaches to PageRank Optimization*. Computing Research Repository - CORR, vol.58. 2010.
- [8] M. Gomes, J. Alfonso-Cendón, P. Marqués-Sánchez, D. Carneiro e P. Novais. *Improving Conflict Support Environments with Information Regarding Social Relationships*. Advances in Artificial Intelligence – IBERAMIA 2014. 2010.
- [9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot e E. Duchesnay. *Scikit-learn: Machine Learning in Python*. The Journal of Machine Learning Research. vol.12, pp. 2825–2830. 2011.
- [10] L. Sales. *Risk prevention of public procurement in the brazilian government using credit scoring*. OBEGEF-Observatório de Economia e Gestão de Fraude OBEGEF Working Papers on Fraud and Corruption. 2013.
- [11] D. Sharma e A. Surolia. *Degree Centrality*. Encyclopedia of Systems Biology p. 558. 2013.
- [12] M. R. Walsh. *Toward Spatial Decision Support Systems in Water Resources*. Water vol.5 pp.798-818. 2013.
- [13] M. Gomes, T. J. M. Oliveira, D. R. Carneiro, P. Novais e J. Neves. *Studying the effects of stress on negotiation behaviour*. Cybernetics and Systems vol.45(3), pp. 279–291. 2014.

-
- [14] H. Liu, A. Gegov e F. T. Stahl. *Categorization and Construction of Rule Based Systems*. Communications in Computer and Information Science vol.459. 2014.
 - [15] X. Yao e Y. Liu. *Machine Learning*. Search Methodologies pp. 477–517. 2014.
 - [16] J. Miao e L. Niu. *Survey on deep learning with class imbalance*. Procedia Computer Science. vol.91, pp.919-926. 2016.
 - [17] M. Nickel, K. Murphy, V. Tresp e E. Gabrilovich. *A Review of Relational Machine Learning for Knowledge Graphs*. Proceedings of the IEEE, vol. 104, pp. 11-33. 2016.
 - [18] S. Ramírez-Gallego e S. García. *Data discretization: taxonomy and big data challenge*. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery vol.6. 2016.
 - [19] C. M. Salgado, C. Azevedo, H. Proença e S. M. Vieira. *Missing Data*. Secondary Analysis of Electronic Health Records, pp. 441–448. 2016.
 - [20] A. Géron. *Hands-On Machine Learning With Scikit-Learn, Keras, And Tensorflow*. O'Reilly Media. 2017.
 - [21] S. Bhanja e A. Das. *Impact of Data Normalization on Deep Neural Network for Time Series Forecasting*. Deep Neural Network for Time Series Forecasting. 2018.
 - [22] P. Raut e A. Sethia. *Application of LSTM, GRU and ICA for Stock Price Prediction*. Proceedings of ICTIS, vol. 2, pp. 479-487. 2018.
 - [23] S. Srivastava e A. K. Singh. *Graph Based Analysis of Panama Papers*. 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC), pp. 822-827. 2018.
 - [24] J. M. Johnson e T. M. Khoshgoftaar. *Survey on deep learning with class imbalance*. Journal of Big Data. 2019.
 - [25] M. Fazekas e S. Saussier. *Big Data in Public Procurement. Colloquium. Law and Economics of Public Procurement Reforms*, pp.131-146. 2020.
 - [26] S. Iyengar e A. Saxena. *Centrality Measures in Complex Networks: A Survey*. 2020.
 - [27] M. E. K. Niessen, J. M. P. Coronel e J. I. P. Fernandez. *Anomaly Detection in Public Procurements using the Open Contracting Data Standard*, pp.127-134. 2020.
 - [28] M. Vakili, M. Ghamsari e M. Rezaei. *Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification*. 2020.
 - [29] R. B. Velasco, I. Carpanese, R. Interian, O. C. G., P. Neto e C. C. Ribeiro. *A decision support system for fraud detection in public procurement*. International Transactions in Operational Research vol.28. 2020.
 - [30] J. Vrana. *The NDE 4.0: Key Challenges, Use Cases, and Adaption*. 2020.
 - [31] *About Keras*. [Online]. (Date last accessed on Mar. 31, 2022). url: <https://keras.io/about/>.
 - [32] P. J. M. Ali e R. H. Faraj.

- [33] *Autoridade da Concorrência. Guia de boas práticas de combate ao conluio na contratação pública.*[Online]. (Date last accessed on Mar. 31, 2022). url: <https://poise.portugal2020.pt/documents/10180/19827/Guia+de+Boas+Praticas+-+Combate+ao+Conluio+na+Contrata%C3%A7%C3%A3o+P%C3%ABlica.pdf/236cd409-8cd2-4de5-afdf-f9edf51db03e>.
- [34] P. A. Bonatti, S. Decker, A. Polleres e V. Presutti. *Knowledge Graphs: New Directions for Knowledge Representation on the Semantic Web. Dagstuhl Reports*, vol. 8, pp. 29-111. 2018.
- [35] L. M. Botelho. *Sistemas Baseados em Regras.*[Online]. (Date last accessed on Mar. 31, 2022). 2015. url: <http://home.iscte-iul.pt/~luis/aulas/ia/Sistemas%20baseados%20em%20regras.pdf>.
- [36] K. E. Boucheffry e R. S. de Souza. *Chapter 12 - Learning in Big Data: Introduction to Machine Learning. Knowledge Discovery in Big Data from Astronomy and Earth Observation*, pp. 225-249. 2020.
- [37] J. Brownlee. *Supervised and Unsupervised Machine Learning Algorithms.*[Online]. (Date last accessed on Mar. 31, 2022). 2016. url: <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>.
- [38] J. Chung, C. Gulcehre, K. Cho e Y. Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR*. 2014.
- [39] A. F. Colladon e E. Remondi. *Using social network analysis to prevent money laundering. Expert Systems with Applications* vol. 67, pp. 49-58. 2017.
- [40] *Concorrência acusa sete empresas de segurança por cartel em concursos públicos.*[Online]. (Date last accessed on Mar. 31, 2022). 2021. url: <https://www.publico.pt/2021/07/19/economia/noticia/concorrenca-acusa-sete-empresas-seguranca-cartel-concursos-publicos-1970962>.
- [41] *Corruption perceptions index.*[Online]. (Date last accessed on Mar. 31, 2022). url: <https://www.transparency.org/en/cpi/2019/index/nzl>.
- [42] *Código dos Contratos Públicos Decreto-Lei n.º 18/2008.*[Online]. (Date last accessed on Mar. 31, 2022). url: <https://dre.pt/web/guest/legislacao-consolidada/-/lc/149398001/202101201612/73921392/diploma/indice>.
- [43] S. Dhankhad, E. Mohammed e B. Far. *Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study. IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 122-125. 2018.
- [44] *Eigenvector Centrality.*[Online]. (Date last accessed on Mar. 31, 2022). url: <https://www.sci.unich.it/~francesco/teaching/network/eigenvector.html>.

- [45] European Parliament. *The Cost of Non- Europe in the area of Organised Crime and Corruption Annex II - Corruption*. [Online]. (Date last accessed on Mar. 31, 2022). url: https://www.europarl.europa.eu/RegData/etudes/STUD/2016/579319/EPRS_STU%282016%29579319_EN.pdf.
- [46] *Exploring artificial intelligence for anti-corruption*. [Online]. (Date last accessed on Mar. 31, 2022). url: <https://www.u4.no/publications/artificial-intelligence-a-promising-anti-corruption-tool-in-development-settings/shortversion>.
- [47] S. Gerdon e. Valesca Molinari. [Online]. (Date last accessed on Mar. 31. *How governments can use public procurement to shape the future of AI regulation – and boost innovation and growth*. url: <https://www.weforum.org/agenda/2020/06/artificial-intelligence-ai-government-procurement-standards-regulation-economic-growth-covid-19-response/>.
- [48] J. Golbeck. *Chapter 3 - Network Structure and Measures*. *Analyzing the Social Web*, pp. 25-44. 2013.
- [49] *Guia do Combate ao Conluio*. url: <https://www.concorrencia.pt/sites/default/files/Guia%20do%20Combate%20ao%20Conluio.pdf>.
- [50] S. Gupta. *Deep Learning vs. traditional Machine Learning algorithms used in Credit Card Fraud Detection*. 2016.
- [51] A. Hagberg, P. Swart e D. S. Chult. *Exploring Network Structure, Dynamics, and Function Using NetworkX*. *Proceedings of the 7th Python in Science Conference*. 2008.
- [52] D. M. Hawkins. *Identification of Outliers*. *Chapman and Hall*. 1980.
- [53] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A. cyrille Ngonga Ngomo, Dice, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab e A. Zimmermann. *Knowledge Graphs*. *ACM Comput. Surv*, vol.54(4). 2021.
- [54] X.-Y. C. Jia Wu, H. Zhang, L.-D. Xiong, H. Lei e S.-H. Deng. *Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization*. *Journal of Electronic Science and Technology*, vol. 17, pp. 26-40. 2019.
- [55] V. Kalra e D. R. Aggarwal. *Importance of Text Data Preprocessing Implementation in RapidMiner*, pp. 71-75. 2018.
- [56] S. Kamel. *Decision Support in the Governorates Level in Egypt*. *4th Information Resources Management Association International Conference (IRMA) on Challenges for Information Management in a World Economy*, vol. 4. 1993.
- [57] S. B. Kotsiantis, D. Kanellopoulos e P. E. Pintelas. *Data Preprocessing for Supervised Learning*. *International Journal of Computer Science*, vol. 1, pp. 111-117. 2006.

- [58] S. Kotsiantis, D. Kanellopoulos e P. Pintela. *Handling imbalanced datasets: A review. GESTS International Transactions on Computer Science and Engineering*, vol. 30, pp. 25-36. 2005.
- [59] V. Losarwar e D. M. Joshi. *Data Preprocessing in Web Usage Mining. Intelligent Systems and Computing*, vol. 202. 2013.
- [60] D. K. Mahto e L. Singh. *A Dive into Web Scraper World. 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 689-693. 2016.
- [61] G. M. Marakas. *Decision Support Systems in the 21st Century*. 1999.
- [62] J. Z. Marco Gomes Paulo Novais. *A Non-intrusive Approach to Measuring Trust in Opponents in a Negotiation Scenario. In: Pagallo, U., Palmirani, M., Casanovas, P., Sartor, G., Villata, S. (eds) AI Approaches to the Complexity of Legal Systems. AICOL 2015, AICOL 2016, AICOL 2016, AICOL 2017, AICOL 2017. Lecture Notes in Computer Science()*, vol. 10791. 2018.
- [63] S. A. Metwalli. *Choose the Best Python Web Scraping Library for Your Applications.[Online]. (Date last accessed on Mar. 31, 2022). 2020. url: <https://towardsdatascience.com/choose-the-best-python-web-scraping-library-for-your-application-91a68bc81c4f>.*
- [64] D. L. Minh, A. Sadeghi-Niaraki, H. D. Huy, K. Min e H. Moon. *Deep Learning Approach for Short-Term Stock Trends Prediction Based on Two-Stream Gated Recurrent Unit Network. IEEE Access*, vol. 6, pp. 55392-55404. 2018.
- [65] A. Mishra e C. Ghorpade. *Credit Card Fraud Detection on the Skewed Data Using Various Classification and Ensemble Techniques. IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pp. 1-5. 2018.
- [66] T. M. Mitchell. *Machine Learning. McGraw-Hill, Inc., USA, vol.1*. 1997.
- [67] M. Morge e P. Mancarella. *The hedgehog and the fox. An argumentation-based decision support system*, vol. 4946, pp. 114-131. 2007.
- [68] N.L.W.Keijsers. *Neural Networks. Encyclopedia of Movement Disorders*, pp. 257-259. 2010.
- [69] *OECD Support on Public Procurement.[Online]. (Date last accessed on Mar. 31, 2022). url: <https://www.oecd.org/gov/public-procurement/support/>.*
- [70] *Open Contracting Partnership. Red flags for integrity: Giving the green light to open data solutions. url: <https://www.open-contracting.org/wp-content/uploads/2016/11/OCP2016-Red-flags-for-integrityshared-1.pdf>.*
- [71] H. Patel. *How Web Scraping is Transforming the World with its Applications.[Online]. (Date last accessed on Mar. 31, 2022). 2018. url: <https://towardsdatascience.com/https-medium-com-hiren787-patel-web-scraping-applications-a6f370d316f4>.*
- [72] *Perguntas Frequentes.[Online]. (Date last accessed on Mar. 31, 2022). url: <http://www.base.gov.pt/Base/pt/PerguntasFrequentes>.*

-
- [73] M. Phi. *Illustrated Guide to LSTM's and GRU's: A step by step explanation*. [Online]. (Date last accessed on Mar. 31, 2022). 2018. url: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>.
- [74] *Práticas de Colusão* [Online]. (Date last accessed on Mar. 31, 2022). url: <https://www.concorrencia.pt/pt/praticas-de-colusao>.
- [75] J. Quinlan. *Induction of Decision Trees*. Machine Learning, J, pp.81-106.
- [76] J. J. A. P. N. P. H. Ricardo Martins Marco Gomes. *Hate Speech Classification in Social Media Using Emotional Analysis*. 7th Brazilian Conference on Intelligent Systems (BRACIS), pp. 61-66. 2018.
- [77] SIGMA Programme. *Detecting and Correcting Common Errors in Public Procurement*. [Online]. (Date last accessed on Mar. 31, 2022). 2016. url: <http://www.sigmaweb.org/publications/Public-Procurement-Policy-Brief-29-200117.pdf>.
- [78] L. Wang, Z. Zhang, X. Zhang, X. Zhou, P. Wang e Y. Zheng. *Chapter One - A Deep-forest based approach for detecting fraudulent online transaction*. *Advances in Computers*, vol. 120, pp. 1-38. 2021.
- [79] A. Willmer, J. Duhan e L. Gibson. *Deloitte. Robotic Process Automation in the Public Sector*. url: <https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/Innovation/deloitte-uk-innovation-the-new-machinery-of-govt.pdf>.