



PEDRO MIGUEL ROCHA CORREIA DE OLIVEIRA
Licenciado em Ciência e Engenharia Informática

**CLASSIFICAÇÃO SUPERVISIONADA DE
DOCUMENTOS DE TEXTO CRU EM
CONTEXTOS DIFÍCEIS.**

MESTRADO EM ENGENHARIA INFORMÁTICA

Universidade NOVA de Lisboa
Setembro, 2022



CLASSIFICAÇÃO SUPERVISIONADA DE DOCUMENTOS DE TEXTO CRU EM CONTEXTOS DIFÍCEIS.

PEDRO MIGUEL ROCHA CORREIA DE OLIVEIRA

Licenciado em Ciência e Engenharia Informática

Orientador: Joaquim Francisco Ferreira da Silva

Assistant professor, NOVA University Lisbon

Júri

Presidente: Name of the committee chairperson

Full Professor, FCT-NOVA

Orientador: Name of the adviser present in defense

Associate Professor, Some University

Vogal: Another member of the committee

Full Professor, Another University

Classificação Supervisionada de Documentos de Texto Cru em contextos difíceis.

Copyright © Pedro Miguel Rocha Correia de Oliveira, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

AGRADECIMENTOS

Em primeiro lugar, gostava de agradecer ao meu orientador o Professor Doutor Joaquim Silva por toda a ajuda e orientação prestada durante a elaboração da presente tese, bem como no esclarecimento de dúvidas a algumas unidades curriculares ao longo do curso. Realizar a tese com o professor, foi sem margem para dúvidas, a decisão acertada. Seguidamente, queria agradecer a minha namorada por toda a ajuda e paciência, pois sempre soube conceder-me o espaço que necessitava para trabalhar no mestrado e alcançar este objetivo. Por fim, mas não menos importante, dedico esta tese à minha família, que sempre me ajudou, em especial aos meus cotas. Mais mestres virão, pois estou certo que o meu irmão Ricardo e irmã Matilde também chegarão lá.

«You cannot teach a man anything; you can only help him discover it in himself.» (Galileo)

RESUMO

A classificação de documentos é uma área com cada vez mais aplicações. Naturalmente, o número de estudos produzidos tem vindo assim a aumentar e, conseqüentemente as propostas e abordagens aos diversos problemas têm vindo a melhorar. A classificação de documentos ramifica-se também pelas diferentes abordagens a cada problema, uma vez que estas abordagens diferem consoante a forma como os dados são apresentados. Os resultados obtidos por cada método apresentam por norma diferenças em termos de precisão. De forma genérica, a classificação supervisionada permite obter resultados melhores em comparação com a classificação não supervisionada, onde *a priori* os dados não têm uma classe conhecida.

Neste sentido, quer a Atribuição de Autoria quer a Verificação de Plágio fazem parte da classificação de documentos. Embora com objetivos fundamentalmente diferentes, ambas têm como objetivo inferir a partir do conjunto de dados que compõem um documento, informação sobre o seu autor. Para a realização da presente dissertação, pretende-se desenvolver um sistema capaz de realizar a atribuição de autoria, mas também o de rejeitar um documento que seja muito dissemelhante de qualquer dos protótipos aprendidos numa fase de treino. Ou seja, o sistema recebe um conjunto de amostras (documentos) produzidos por cada autor, extrai informação útil que represente cada um e após uma fase de treino, recebe novos documentos e tenta atribuir-lhes uma das autorias anteriormente aprendidas na referida fase. Caso o documento seja muito dissemelhante de qualquer dos protótipos aprendidos, o sistema deve ser capaz de rejeitar a atribuição de qualquer autoria a este documento.

De forma genérica, qualquer problema de classificação parte da assunção de que para objetos diferentes existe algo que os permite distinguir. Assim, uma das grandes dificuldades passa pela identificação de quais atributos presentes nos dados permitem a identificação de cada autor. Após esta fase, um autor será representado pelo conjunto de atributos que o descreve. Assim, idealmente, após esta fase será possível agrupar autores através do grupo escolhido de características (atributos).

Palavras-chave: Classificação de Documentos, Atribuição de Autoria, Classificação não Supervisionada, Classificação Supervisionada, Clustering , Extração de atributos

ABSTRACT

Document classification is an area with increasing applications. Naturally, the number of studies produced has been increasing and, consequently, the proposals and approaches to the various problems have been improving. The classification of documents is also ramified by the different approaches to each problem, as these approaches differ depending on how the data is presented. The results obtained by each method usually show differences in terms of precision. In general, supervised classification allows obtaining better results when compared to unsupervised classification, where a priori data do not have a known class.

In this sense, both the Attribution of Authorship and Plagiarism Verification are part of the document classification. Although with fundamentally different goals, both aim to infer from the data set that make up a document, information about its author. In order to carry out this dissertation, the goal was to develop a system capable of attributing authorship, but also of rejecting a document that is very different from any of the learned prototypes in a training phase. That is, the system receives a set of samples (documents) produced by each author, extracts useful information that represents each one and after being trained, receives new documents and tries to assign them one of the authors previously learned in that phase. If the document is very different from any of the learned prototypes, the system must be able to reject the assignment of any authorship of this document.

Generally speaking, any classification problem starts from the assumption that for different objects there is something that allows to distinguish them. So, one of the big difficulties involves the identification of which attributes present in the data allow the identification of each author. After this phase, an author will be represented by the set of attributes that describe it. Thus, ideally, after this phase it will be possible to group authors through the chosen group of characteristics (attributes).

Keywords: Documents Classification, Attribution of Authorship, Supervised Learning, Unsupervised Learning , Clustering , Features Extraction

ÍNDICE

Índice de Figuras	xi
Índice de Tabelas	xv
1 Introdução	1
1.1 Contexto	1
1.2 Motivação	3
1.3 Objetivos	3
1.4 Estrutura do documento	4
2 Trabalho relacionado	6
2.1 A natureza dos dados	6
2.2 Ferramentas relacionadas	6
2.2.1 Distâncias entre pontos	7
2.2.2 Pré-processamento de Texto	8
2.2.3 O poder discriminante das <i>features</i>	8
2.2.4 Redução de dimensionalidade	10
2.2.5 Métricas para avaliação de performance	10
2.2.6 Métodos de Clustering	11
2.2.7 Algoritmos classificadores	13
2.2.8 AdaBoost	14
2.2.9 Decision Tree	15
2.2.10 <i>Random Forests</i>	16
2.2.11 Redes neurais	17
2.2.12 Seleção de modelos	21
2.3 Estado da arte	23
2.3.1 Classificação supervisionada	23
2.3.2 Classificação não supervisionada	34
3 Solução Proposta	36

3.1	Classificação em contextos difíceis	36
3.1.1	Introdução	36
3.1.2	<i>Features</i> suficientemente discriminantes	36
3.2	A classificação de documentos	38
3.3	Classificar documentos utilizando o quadrado da distância de <i>Mahalanobis</i>	39
3.4	A rejeição de documentos	40
3.4.1	Posicionamento do problema	40
3.4.2	Resolução do problema	41
3.4.3	Transformação de Box-Cox	42
4	Avaliação	44
4.1	Introdução	44
4.1.1	Pré-processamento do texto	44
4.2	Diferentes <i>datasets</i> e os resultados obtidos.	45
4.2.1	Autores do séc 19 e 20	45
4.2.2	Classificação por género	47
4.2.3	Classificação de autores de diferentes épocas	49
4.2.4	Dataset Fernando Pessoa	50
4.2.5	Descrição do dataset	50
4.2.6	Redução de dimensionalidade	53
4.2.7	Módulo de rejeição	60
4.2.8	A problemática da evolução da escrita em <i>features</i> estatísticos.	64
5	Conclusões	67
5.1	Conclusões sobre o trabalho produzido	67
5.2	Trabalho Futuro	68
	Bibliografia	70
	Apêndices	
A	Quadros Auxiliares	73
A.1	Dataset de autores do século 19 e 20	73
A.2	Classificação por género	75
A.3	Autores de diferentes épocas	76
A.4	Dataset Fernando Pessoa	78
A.4.1	Dataset completo	78
A.4.2	Dataset incompleto	79
B	Figuras Auxiliares	82
B.1	Datasets de autores do século 10 e 20	82
B.2	Classificação por género	84
B.3	Autores de diferentes épocas	86

B.4	Dataset Fernando Pessoa	88
B.4.1	Dataset completo	88
B.4.2	Dataset incompleto	90

ÍNDICE DE FIGURAS

1.1	Resolução genérica de problemas de classificação.	2
2.1	Exemplo da formação de <i>clusters</i> pelo <i>K-Means</i>	12
2.2	Exemplo da formação de <i>clusters</i> pelo <i>Bisecting K-Means</i>	12
2.3	Exemplo ilustrativo do algoritmo EM.	13
2.4	Árvore de decisão para prever se o dia é bom ou não para praticar ténis. . .	16
2.5	Exemplo de Stochastic gradient descent.	19
2.6	Representação gráfica da função <i>sigmoid</i>	20
2.7	Exemplo de <i>cross-validation</i> com $k=4$	22
2.8	Exemplo de <i>Leave-One-Out</i>	23
2.9	Matriz de confusão obtida para identificação de género; conjunto de teste com 10.000 entradas de <i>blogs</i> do género masculino e 15.00 do género feminino; precisão de 0.65 , <i>Recall</i> de 0.71 , F-measure de 0.68 com α de 0.5. Stop-words não foram removidas.	24
2.10	Resultados obtidos na classificação por género do estudo [26].	25
2.11	Abordagem geral seguida para identificação do género do autor no estudo [5].	26
2.12	Resultados obtidos por classificador no estudo de Na Cheng, Rajarathnam Chandramouli e K.Subbalakshmi [5].	27
2.13	Precisão obtida por <i>threshold</i> usando diferentes atributos no estudo [12]. . .	29
2.14	Grafo resultante do processamento de três frases produzido pelo sistema ISG.	30
2.15	Atributos utilizados no estudo de Fatma Howedi [15].	31
2.16	Resultados obtidos por classificador em [15].	31
2.17	Resultados obtidos por <i>Feature Selection</i> e <i>Value Assignment</i> em [20].	32
2.18	Resultados obtidos com o <i>dataset</i> pessoa em [24].	33
2.19	Ilustração da metodologia seguida em [17].	35
3.1	Processo geral de classificação adotado.	39
3.2	Transformação de uma variável com uma distribuição desconhecida para distribuição normal.	42

4.1	Matriz de confusão obtida através dos resultados do classificador <i>Random Forest</i> - (<i>RF</i>).	47
4.2	Matriz de confusão para o classificador <i>Random Forest</i> para a determinação do género do autor.	48
4.3	Matriz de confusão do classificador <i>Gaussian naive Bayes</i> para o <i>dataset</i> de autores de diferentes épocas.	50
4.4	Descrição do <i>dataset</i> Fernando Pessoa.	51
4.5	Distribuição de documentos por classe no <i>dataset</i> de Fernando Pessoa.	51
4.6	Análise do <i>dataset</i> de Fernando Pessoa.	52
4.7	Execução do algoritmo <i>PCA - Principal Component Analysis</i> para o <i>dataset</i> de Fernando Pessoa.	54
4.8	Matriz de confusão obtida pelo classificador <i>BG</i> para o <i>dataset</i> dos heterónimos normalizado.	55
4.9	Matriz de confusão obtida pelo classificador <i>Bagging Classifier</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por classe/autor.	56
4.10	Matriz de confusão obtida pelo classificador <i>AdaBoost</i> para 2 heterónimos.	57
4.11	Matriz de confusão obtida pelo modelo neuronal para o <i>dataset</i> contendo 4 heterónimos com o número de documentos por classe normalizado.	59
4.12	Matriz de confusão obtida pelo modelo neuronal para o <i>dataset</i> contendo 4 heterónimos com todos os documentos existentes por classe.	60
4.13	Exemplos de transformações <i>Box-Cox</i>	61
4.14	Método do cotovelo para a escolha do número de <i>clusters</i>	65
4.15	Análise de <i>Silhouette</i> em função do número de <i>clusters</i>	65
4.16	Projeção dos documentos representados pelos atributos.	66
B.1	Matriz de confusão para o classificador <i>Bagging Classifier</i> para o <i>dataset</i> contendo autores do século 19 e 20.	82
B.2	Matriz de confusão para o classificador <i>Decision Tree</i> para o <i>dataset</i> contendo autores do século 19 e 20.	82
B.3	Matriz de confusão para o classificador <i>Gaussian Naive Bayes</i> para o <i>dataset</i> contendo autores do século 19 e 20.	83
B.4	Matriz de confusão para o classificador <i>Support Vector Machines</i> para o <i>dataset</i> contendo autores do século 19 e 20.	83
B.5	Matriz de confusão para o classificador <i>AdaBoost</i> para o <i>dataset</i> contendo autores do século 19 e 20.	83
B.6	Matriz de confusão para o classificador <i>Bagging Classifier</i> para a classificação por género.	84
B.7	Matriz de confusão para o classificador <i>Decision Tree</i> para a classificação por género.	84

B.8	Matriz de confusão para o classificador <i>Gaussian Naive Bayes</i> para a classificação por género.	85
B.9	Matriz de confusão para o classificador <i>Support Vector Machines</i> para a classificação por género.	85
B.10	Matriz de confusão para o classificador <i>AdaBoost</i> para a classificação por género.	85
B.11	Matriz de confusão para o classificador <i>Bagging Classifier</i> para o <i>dataset</i> contendo autores de épocas diferentes.	86
B.12	Matriz de confusão para o classificador <i>Decision Tree</i> para o <i>dataset</i> contendo autores de épocas diferentes.	86
B.13	Matriz de confusão para o classificador <i>Random Forest</i> para o <i>dataset</i> contendo autores de épocas diferentes.	87
B.14	Matriz de confusão para o classificador <i>Support Vector Machines</i> para o <i>dataset</i> contendo autores de épocas diferentes.	87
B.15	Matriz de confusão para o classificador <i>AdaBoost</i> para o <i>dataset</i> contendo autores de épocas diferentes.	87
B.16	Matriz de confusão para o classificador <i>Random Forest</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por autor.	88
B.17	Matriz de confusão para o classificador <i>Decision Tree</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por autor.	88
B.18	Matriz de confusão para o classificador <i>Gaussian Naive Bayes</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por autor.	89
B.19	Matriz de confusão para o classificador <i>Support Vector Machines</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por autor.	89
B.20	Matriz de confusão para o classificador <i>AdaBoost</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por autor.	89
B.21	Matriz de confusão para o classificador <i>Random Forest</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com normalização do número de documentos por autor.	90
B.22	Matriz de confusão para o classificador <i>Decision Tree</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com normalização do número de documentos por autor.	90
B.23	Matriz de confusão para o classificador <i>Gaussian Naive Bayes</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com normalização do número de documentos por autor.	91
B.24	Matriz de confusão para o classificador <i>Support Vector Machines</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com normalização do número de documentos por autor.	91

B.25 Matriz de confusão para o classificador <i>AdaBoost</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com normalização do número de documentos por autor.	91
---	----

ÍNDICE DE TABELAS

3.1	Exemplo de dois documentos produzidos por 2 heterónimos.	37
3.2	Atributos utilizados na proposta de solução seguida.	37
4.1	<i>Dataset</i> utilizado.	45
4.2	Resultados de avaliação das <i>features</i>	46
4.3	Tabela de avaliação de <i>performance</i> pelo classificador <i>Random Forest - (RF)</i> . . .	47
4.4	Resultados de avaliação das <i>features</i>	48
4.5	Tabela de avaliação de resultados pelo classificador <i>Random Forest - (RF)</i> . . .	48
4.6	Tabela de avaliação de resultados obtidos pelo classificador <i>Gaussian Naive Bayes</i> para o <i>dataset</i> de autores de diferentes épocas.	49
4.7	Resultados de avaliação das <i>features</i> para o <i>dataset</i> de Fernando Pessoa. . . .	53
4.8	Resultados obtidos pelo classificador <i>BG - Bagging Classifier</i> para o <i>dataset</i> dos heterónimos normalizado.	55
4.9	Resultados obtidos pelo classificador <i>Bagging Classifier</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por classe/autor.	55
4.10	Tabela de resultados obtidos pelo classificador <i>AdaBoost</i> para 2 heterónimos. . .	56
4.11	Modelo neuronal utilizado.	58
4.12	Resultados obtidos pelo modelo neuronal para o <i>dataset</i> contendo 4 heterónimos com o número de documentos por classe normalizado.	59
4.13	Resultados obtidos pelo modelo neuronal para o <i>dataset</i> contendo 4 heterónimos com todos os documentos existentes por classe.	60
4.14	Teste de Shapiro-Wilk.	62
4.15	Classificação obtida com recurso a Distância de <i>Mahalanobis</i>	63
4.16	Performance da rejeição com $\alpha = 0.001$	63
A.1	Atributos utilizados por classificador para o <i>dataset</i> composto por autores do século 19 e 20.	73
A.2	Legenda das Tabelas dos autores do século 19 e 20.	73

A.3	<i>Performance</i> do classificador <i>AdaBoost</i> para o <i>dataset</i> contendo autores do século 19 e 20.	73
A.4	<i>Performance</i> do classificador <i>Decision Tree</i> para o <i>dataset</i> contendo autores do século 19 e 20.	74
A.5	<i>Performance</i> do classificador <i>Gaussian Naive Bayes</i> para o <i>dataset</i> contendo autores do século 19 e 20.	74
A.6	<i>Performance</i> do classificador <i>Support Vector Machines</i> para o <i>dataset</i> contendo autores do século 19 e 20.	74
A.7	<i>Performance</i> do classificador <i>Bagging Classifier</i> para o <i>dataset</i> contendo autores do século 19 e 20.	74
A.8	Atributos utilizados por classificador para a classificação por género.	75
A.9	Desempenho obtido pelo classificador <i>Support Vector Machines</i> para o <i>dataset</i> contendo autores de ambos os géneros.	75
A.10	Desempenho obtido pelo classificador <i>Bagging Classifier</i> para o <i>dataset</i> contendo autores de ambos os géneros.	75
A.11	Desempenho obtido pelo classificador <i>AdaBoost</i> para o <i>dataset</i> contendo autores de ambos os géneros.	75
A.12	Desempenho obtido pelo classificador <i>Gaussian Naive Bayes</i> para o <i>dataset</i> contendo autores de ambos os géneros.	76
A.13	Desempenho obtido pelo classificador <i>Decision Tree</i> para o <i>dataset</i> contendo autores de ambos os géneros.	76
A.14	Atributos utilizados por classificador para o <i>dataset</i> composto por autores de épocas diferentes.	76
A.15	Legenda das Tabelas para o <i>dataset</i> contendo autores de épocas diferentes.	76
A.16	Desempenho obtido pelo classificador <i>Random Forest</i> para o <i>dataset</i> contendo autores de épocas diferentes.	77
A.17	Desempenho obtido pelo classificador <i>AdaBoost</i> para o <i>dataset</i> contendo autores de épocas diferentes.	77
A.18	Desempenho obtido pelo classificador <i>Bagging Classifier</i> para o <i>dataset</i> contendo autores de épocas diferentes.	77
A.19	Desempenho obtido pelo classificador <i>Support Vector Machines</i> para o <i>dataset</i> contendo autores de épocas diferentes.	77
A.20	Desempenho obtido pelo classificador <i>Decision Tree</i> para o <i>dataset</i> contendo autores de épocas diferentes.	77
A.21	Legenda das Tabelas para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa.	78
A.22	Atributos utilizados por classificador para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa.	78
A.23	Desempenho obtido pelo classificador <i>Random Forest</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa.	78

A.24 Desempenho obtido pelo classificador <i>Decision Tree</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa.	78
A.25 Desempenho obtido pelo classificador <i>Gaussian Naive Bayes</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa.	79
A.26 Desempenho obtido pelo classificador <i>Support Vector Machines</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa.	79
A.27 Desempenho obtido pelo classificador <i>AdaBoost</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa.	79
A.28 Melhor combinação de atributos obtida por classificador para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.	79
A.29 Desempenho obtido pelo classificador <i>AdaBoost</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.	80
A.30 Desempenho obtido pelo classificador <i>Random Forest</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.	80
A.31 Desempenho obtido pelo classificador <i>Decision Tree</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.	80
A.32 Desempenho obtido pelo classificador <i>Gaussian Naive Bayes</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.	80
A.33 Desempenho obtido pelo classificador <i>Support Vector Machines</i> para o <i>dataset</i> contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.	81

INTRODUÇÃO

1.1 Contexto

Aprendizagem automática é o termo que se refere aos algoritmos que têm a capacidade de aprender e de usar o conhecimento adquirido para reconhecimento futuro. Este é para mim o campo mais promissor de todos! Vivemos numa época em que cada vez são produzidos mais dados, podendo estes produzir conhecimento e facilitar a tomada de decisões. Dados podem ser definidos como sendo um conjunto de observações/factos. Factos estes que podem ser vistos como o resultado de relações complexas entre os vários campos que descrevem uma amostra. Ao sermos capazes de compreender que tipo de relação existe entre os diversos atributos de uma amostra estamos a produzir conhecimento. Assim, de acordo com esta visão, os dados podem gerar conhecimento. A decisão de elaborar a tese nesta área alicerçou-se nesta linha de pensamento e no interesse constante em saber mais relativamente a como se pode extrair informação útil de entre os diversos dados.

A tarefa simples de classificação é uma atividade realizada pela humanidade, provavelmente desde sempre. Na sua essência, pode ser simplificada como sendo a atribuição de um grupo/classe a um determinado objeto. A classificação de documentos segue o mesmo princípio. Embora de um ponto de vista prático, a tarefa possa parecer simples, na prática, muitas das vezes não é. Algumas destas dificuldades podem ser atribuídas à forma como os dados se encontram estruturados, ou até mesmo a falta de alguns dados em diversas características por cada amostra. Para lidar com os diversos problemas existentes, foram sendo propostas diferentes abordagens para determinados tipos de problemas. Duas destas aproximações de classificação de dados são a supervisionada (aprendizagem supervisionada) e não supervisionada (aprendizagem não supervisionada), pese que embora com o mesmo propósito, devem ser utilizadas para contextos diferentes.

A classificação supervisionada é possível quando temos amostras pertencentes ao nosso conjunto de dados que se encontram já categorizadas. O objetivo é então utilizar os atributos de cada amostra e com estes, construir uma função através da qual seja possível mapear novos exemplos. Por outro lado, a classificação não supervisionada é utilizada

quando nenhuma informação *a priori* relativamente à classe de cada amostra se encontra presente. Como tal, esta abordagem têm que ser capaz de inferir relações entre os dados que permitam depois classifica-los de acordo com a semelhança ou dissemelhança com outras amostras pertencentes ao conjunto de dados.

A resolução genérica de um problema de classificação pode ser vista de grosso modo na Figura 1.1.

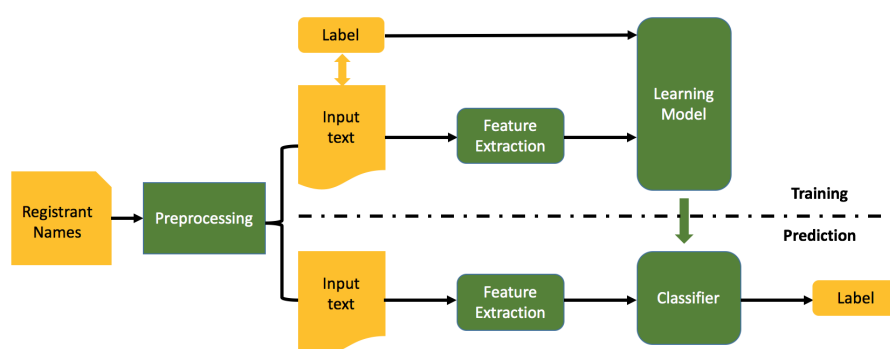


Figura 1.1: Resolução genérica de problemas de classificação.

Na primeira parte, são obtidos atributos/*features* que serão utilizados para a resolução do nosso problema. Este, é um passo de extrema importância, uma vez que nas fases subsequentes do problema passar-se-á a representar cada exemplo/amostra como um vetor destas características, colocando-se em questão quais atributos escolher e quais deixar de fora. Não obstante, importa ter assente que embora cada atributo seja uma métrica de informação sobre a amostra, muitas vezes podemos ter atributos cuja informação dada não é importante para o contexto do problema. Além deste facto, a utilização de muitos atributos aumenta a complexidade da solução, pelo que para lidar com esta questão da melhor forma possível, poderá ser necessário utilizar algoritmos de redução de dimensionalidade ao problema que, embora estes sejam explicados com mais detalhe no próximo capítulo, o que fazem de um modo geral é encontrar uma forma de reduzir o número de atributos usados para o estudo minimizando a perda de informação.

O próximo passo requer a utilização de um classificador que atribua a um novo objeto amostra (nosso caso, documento) uma classificação baseando-se para tal no conjunto de características escolhidas anteriormente. A realização deste processo consiste na divisão do nosso conjunto de dados tipicamente em dois sub-conjuntos, um de treino e um de teste. O algoritmo irá utilizar os dados do conjunto de teste e formular uma hipótese de classificação baseando-se nos mesmos. Posto isto, é necessário avaliar, no contexto do conjunto de teste, qual foi a performance do classificador utilizando algumas métricas para esse efeito, pois diferentes algoritmos classificadores poderão produzir diferentes performances, dependendo estas, em parte, da natureza/distribuição associada aos dados.

1.2 Motivação

Qualquer problema de classificação parte do pressuposto de que objetos diferentes podem ser categorizados como pertencentes a classes diferentes. A Classificação de Autoria em documentos assenta exatamente neste mesmo pressuposto, nomeadamente que existem características que permitem diferenciar entre autores de forma a que, aprendidas as diferenças, a autoria de um futuro documento lido possa ser atribuída de forma automática e com fortes probabilidades de êxito. Para ser possível atribuir um autor a um texto/documento, é então necessário *a priori* ter um conhecimento de características de escrita desse e de outros autores. A escolha destas características, que corresponderão aos atributos na resolução do problema, não é uma tarefa simples, sendo um dos principais desafios da presente tese. A questão principal prende-se com a escolha de atributos com poder discriminante o suficiente para que um classificador possa elaborar uma relação entre os valores dos atributos e uma determinada classe, com uma precisão razoável para que a autoria seja atribuída com confiança, num contexto de discriminação difícil. Na verdade, neste contexto particular de discriminação de autores, o problema não tem uma solução fácil. A título de exemplo, se considerarmos vários autores contemporâneos que escrevem na mesma língua, não surgem com facilidade atributos evidentes com tal poder discriminante. Por outro lado, a identificação do género (sexo) do autor de um documento tem sido considerado um desafio. Trata-se de um problema onde é difícil obter valores elevados de performance na fase da classificação. Este é por isso, um dos desafios que se pretende considerar nesta dissertação.

Durante a fase de classificação, a maioria dos algoritmos atribui à amostra a classificar, a classe aprendida que corresponde ao conjunto/*cluster* cujo protótipo se encontra mais perto dessa amostra representada pelo vetor dos atributos escolhidos.

Particularmente no caso da classificação de autoria, este comportamento nem sempre é o desejado dado que podemos não querer classificar um documento que se encontre muito distante de todos os protótipos aprendidos, embora exista sempre um que se encontra mais próximo da amostra. Num sistema funcional isto significaria que os atributos que caracterizam a escrita do autor da amostra são muito diferentes dos atributos dos demais autores que já conhecemos. Por outro lado, se o objetivo for classificar o autor quanto ao género já não se verifica este problema, uma vez que será sempre pretendida uma classificação de acordo com o protótipo mais perto, assumindo a existência de apenas dois géneros, feminino e masculino.

1.3 Objetivos

- O objetivo fulcral da presente tese consiste na realização de um sistema de classificação de documentos em contextos difíceis, sendo a Identificação da Autoria um exemplo, independente do idioma dos documentos apresentados, devendo este ser capaz de discriminar autorias entre os diversos documentos, classes essas que

estarão naturalmente próximas, i.e., classificar documentos em contextos difíceis. Um contexto difícil é definido, no âmbito do presente trabalho, como se tratando de um ambiente em que os dados não estão categorizados nem é fácil categorizar os mesmos, ou seja, não são fornecidos com um conjunto de atributos, dando assim seguimento para outro desafio deste projeto.

- A escolha dos atributos para a realização do presente trabalho é também um dos desafios mais difíceis. A seleção assenta no princípio de que existem características que serão suficientemente dissemelhantes entre os autores em estudo por forma a poder classificar a autoria de determinado documento. Este fato assenta na convicção pessoal de que cada autor tem uma impressão/relação única entre a sua escrita e sua pessoa. A averiguação desta afirmação será também um dos objetivos desta dissertação.
- Outro objetivo é a implementação de um módulo de rejeição, isto é, uma capacidade associada ao classificador de forma a que este possua o poder de rejeitar a Atribuição de Autoria, a determinado autor no caso de o documento a classificar se encontrar muito distante de todos os protótipos aprendidos.
- Uma das funcionalidades que o sistema terá é a capacidade de discriminar o género do autor dos documentos o que exigirá a avaliação das capacidades discriminantes de atributos candidatos de outra natureza, características estas que não dependem da época ou estilo em que estão escritos os documentos. Na fase de classificação o sistema deverá atribuir, obviamente, um dos géneros (masculino/feminino) ao documento em análise.
- Por fim, são realizadas de uma forma não tão extensa, uma comparação entre os resultados obtidos seguindo uma abordagem utilizando classificadores tradicionais e usando redes neuronais, e uma comparação à abordagem supervisionada *vs* não supervisionada usando os mesmos dados, isto é, às amostras caracterizadas pelos mesmos atributos, mas não usando a anotação da classe. A experiência servirá não só para avaliar a qualidade dos atributos, mas também a qualidade do *clustering* atribuído manualmente na versão supervisionada.

No capítulo três do presente documento, será abordada em maior detalhe a metodologia a adotar para a implementação de cada um dos objetivos supramencionados.

1.4 Estrutura do documento

A presente dissertação divide-se em cinco capítulos distintos.

A dissertação inicia-se com a apresentação das motivações que conduziram à elaboração de uma dissertação na área de *Machine Learning*, assim como é feita uma contextualização em que esta se insere e, por fim, é explicada a motivação que antecede os objetivos que se propõem alcançar.

Posteriormente, no segundo capítulo, será aprofundado o rigor sobre algumas definições já mencionadas e, ainda apresentadas ferramentas que se revelaram úteis na realização da tese. No mesmo capítulo, situa-se igualmente o Estado da Arte onde serão analisados diversos estudos com objetivos semelhantes ao da presente dissertação e dadas a conhecer as suas abordagens ao problema, bem como as soluções propostas e os resultados obtidos.

O terceiro capítulo centra-se na apresentação das propostas de soluções desenvolvidas para os vários objetivos propostos de forma detalhada, bem como numa breve explicação sobre cada tomada de decisão sobre os caminhos de solução. Adicionalmente, são também apresentados alguns resultados teóricos que se constituem necessários por forma a sustentar o trabalho prático realizado.

No quarto e penúltimo capítulo, apresenta-se a avaliação, onde é pretendido divulgar os resultados obtidos para cada *dataset* que pretende ilustrar cada problema.

Por fim, no capítulo das conclusões, pretende-se fazer uma resenha sobre o trabalho desenvolvido na presente dissertação, refletindo nomeadamente sobre o cumprimento dos objetivos propostos inicialmente, os resultados obtidos e também sobre o trabalho futuro que pode vir a ser realizado sobre o mesmo tema.

TRABALHO RELACIONADO

2.1 A natureza dos dados

Dados estruturados *vs* não estruturados

Como é sabido, em ambiente de dados estruturados, existe o acesso ao conteúdo de tabelas, matrizes, etc., estruturas estas que contêm atributos normalmente com poder discriminante, e.g., temperatura, pressão arterial, género, peso e altura; estes atributos surgem por norma em número fixo. Por outro lado, a classificação de texto cru insere-se num tipo de problemas em que os dados não são estruturados, já que normalmente não existem as estruturas referidas atrás. Assim, os atributos com poder discriminante têm que ser encontrados no seio do texto sem referências a qualquer estrutura. No contexto particular da caracterização de autoria, terão que ser consideradas características estatísticas associadas, por exemplo, à pontuação da escrita, à extensão do léxico usado por cada autor, entre outros potenciais candidatos a atributos.

2.2 Ferramentas relacionadas

Para a realização de uma tarefa de classificação, existem alguns processos que são amplamente utilizados. Neste capítulo, pretende-se dar a conhecer que processos são esses, bem como fazer uma breve resenha teórica sobre os mesmos. Presumivelmente, alguns não foram utilizados no contexto da presente tese, por diversos motivos, não obstante serão ainda assim apresentados. Estes grupos de ferramentas pretendem dar resposta a problemas diferentes entre si. Subsequentemente, será dada primeiramente uma breve explicação do problema e, posteriormente, a solução oferecida por cada , explicada de forma pormenorizada.

2.2.1 Distâncias entre pontos

Distância euclidiana

Em matemática, a distância euclidiana entre dois pontos num espaço vetorial é definida como sendo o comprimento de um segmento de linha entre os dois pontos, representando assim o comprimento do menor caminho entre estes. Esta distância pode ser calculada a partir de coordenadas cartesianas usando como base o teorema de Pitágoras. No caso geral, para o cálculo de uma distância euclidiana entre dois pontos p e $q \in \mathbb{R}^N$ é definida por:

$$d(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^{i=N} (p_i - q_i)^2}$$

Similaridade por cosseno

A similaridade por cosseno é uma métrica, bastante útil na determinação da similaridade entre objetos independentemente do seu tamanho. Nesta métrica os objetos de determinado *dataset* são tratados como um sendo um vetor. A distância é calculada para 2 vetores x e $y \in \mathbb{R}^N$ da seguinte forma:

$$\text{Cos}(\vec{x}, \vec{y}) = \frac{\vec{x}\vec{y}}{\|\vec{x}\|\|\vec{y}\|}$$

Por contraste, a medida de dissemelhança é definida por $\text{Dis}(\vec{x}, \vec{y}) = 1 - \text{Cos}(\vec{x}, \vec{y})$.

Quadrado da distância de Mahalanobis

O quadrado da distância de Mahalanobis introduzida por P. C. Mahalanobis é uma medida da distância entre um ponto \vec{p} e uma distribuição caracterizada por um centroide μ e uma matriz de covariâncias inversa Σ^{-1} . Esta matriz reflete a distribuição dos elementos de um *cluster* de pontos num espaço vetorial; $\vec{\mu}$ é o vetor médio dos elementos do *cluster*. Trata-se de uma generalização multidimensional da ideia de medir quantos desvios padrão de P está a média de $\vec{\mu}$. Essa distância é zero para \vec{p} igual a $\vec{\mu}$ e cresce à medida que \vec{p} se afasta da média ao longo de cada eixo componente principal. Se cada um desses eixos for redimensionado para ter variância unitária, então a distância de Mahalanobis corresponde à distância euclidiana padrão no espaço transformado. O quadrado da distância de Mahalanobis é, portanto, sem unidade, invariante em escala e leva em consideração as correlações do conjunto de dados. A formula no caso geral é então:

$$M^2(\vec{p}, \vec{\mu}, \Sigma^{-1}) = (\vec{p} - \vec{\mu})^T \Sigma^{-1} (\vec{p} - \vec{\mu})$$

Sendo $\vec{\Sigma}^{-1}$ a matriz de covariâncias. Embora esta métrica seja amplamente utilizada, alguns autores como *William J. Egan e Stephen L. Morgan* em [9] e *Hadi e Simonoff* em [14] questionam-se sobre a confiabilidade dos resultados obtidos. Adicionalmente, para o cálculo desta distância é necessário que exista a matriz de covariâncias inversa, o que nem sempre é possível, especialmente quando as variáveis estão altamente correlacionadas entre si, conforme descrito por *Varmuza, K., e Filzmoser, P.* em [27].

2.2.2 Pré-processamento de Texto

Lematização

Por Lematização entende-se o processo de deflexionar uma palavra por forma a determinar o seu lema, isto é, a determinação da forma canónica. Por exemplo, ao analisarmos as palavras "gato", "gata", "gatos", "gatas", verifica-se que constituem-se todas como formas do mesmo lema: "gato".

Este processo revela-se útil em contextos onde a identificação semântica da raiz das palavras e o seu número de ocorrências tem maior importância do que as suas formas flexionadas. Esta transformação, no que concerne aos termos semanticamente relevantes, pode ajudar no processo de discriminação de tópicos com vista à construção de *clusters*.

Stemming

Esta ferramenta é semelhante à lematização dado que é uma transformação do texto, contudo, o seu objetivo é ligeiramente diferente. Neste processo, cada palavra flexionada é transformada pela sua raiz, por exemplo, na transformação das palavras *cats, catlike e catty*, obter-se-ia sempre a raiz *cat*, que neste caso, coincide com uma palavra. No entanto nem sempre se obtêm esses resultados, concretamente, em casos como *aluno, alunos, aluna, alunas*, resultaria em *alun*, que não coincide com uma palavra.

2.2.3 O poder discriminante das *features*

No contexto desta dissertação é necessário encontrar características/traços capazes de discriminar as diferentes classes em causa, ou seja os diferentes grupos de documentos. Para tal, existem algumas métricas conhecidas para este objetivo. As medidas que se seguem não esgotam o leque de ferramentas utilizadas na presente dissertação mas são muito populares e de grande utilidade.

TF-IDF (term frequency – inverse document frequency)

O TF-IDF é uma medida estatística que tem o intuito de indicar a importância de uma palavra num documento em relação a uma coleção de documentos ou num *corpus*

linguístico (conjunto de documentos). O valor TF-IDF de uma palavra aumenta em proporção à medida que o número de ocorrências da mesma no documento aumenta, no entanto, esse valor é equilibrado pela frequência da palavra no *corpus*. Esta métrica tende a penalizar palavras com poucas ocorrências ou que aparecem num elevado número de documentos que compõem o *corpus*. A métrica revela-se bastante útil para selecionar palavras semanticamente mais relevantes. Sendo assim, o valor de TF-IDF do termo t no documento d pertencente ao conjunto de documentos D , é calculado da seguinte maneira:

$$Tf-Idf(t, d, D) = tf(t, d) \times idf(t, D)$$

$$idf(t, D) = \log\left(\frac{\|D\|}{\|docs \in D : tf(t, docs) > 0\|}\right)$$

Alguns autores preferem a utilização da frequência relativa do termo t em d , isto é, normalizando-se a frequência do termo tendo em conta o tamanho do documento.

Skewness

Para a realização da presente dissertação e de forma mais extensa, para a realização de qualquer problema de classificação que recorra a algoritmos e processos de *Machine Learning*, é essencial conseguir quantificar o grau de significância assumido por cada atributo a ser utilizado. Em particular, no contexto da classificação supervisionada, quais atributos conseguem discriminar documentos entre classes. Para tal, uma das métricas que pode ser utilizada é a medida estatística do terceiro momento ou *skewness* e é definido da seguinte forma:

$$SK = \frac{1}{n} \sum_{i=1}^{i=n} (x_i - \bar{x})^3$$

Esta métrica mede o equilíbrio em torno da média (\bar{x}) de uma distribuição. Desta forma, é então possível detetar *outliers* caso estes existam. A título de exemplo considere-se o conjunto $A = \{0.01, 0.01, 0.01, 0.0105, 0.4\}$ terá um terceiro momento positivo, uma vez que o elemento 0.4 sobressai como sendo significativamente maior que os restantes. Este é um caso típico que deve ser valorizado já que se trata de um atributo que surge com probabilidades baixas na maioria das classes, exceto numa onde assume um valor alto. A métrica também deteta desequilíbrios negativos, no entanto, esses casos não revelam ter interesse. Com efeito um termo é discriminante quando assume valores de frequência relativa altos para uma ou poucas classes, tendo valores baixos ou nulos na maioria das demais classes. Exemplo: A palavra agricultura discrimina documentos (de agricultura) porque tende a ocorrer apenas (ou certamente, com maior probabilidade) nesta classe de documentos.

2.2.4 Redução de dimensionalidade

PCA - Principal Component Analysis e SVD (Singular Value Decomposition)

A *Principal Component Analysis (PCA)*, consiste numa técnica estatística usada para redução de *features* sendo que as *features* originais estão correlacionadas entre si, correspondendo a eixos não ortogonais num espaço vetorial. Aplicando o PCA, são geradas novas *features* ortogonais entre si, tendo por base as originais, o que permite uma redução, por vezes muito significativa, no número de *features* sem que a redução de informação perdida no processo seja significativa. Tal como a técnica PCA, também a SVD permite a redução de *features*. Esta técnica consiste na fatorização da matriz inicial em vectores singulares e valores singulares possuindo inúmeras aplicações imprescindíveis em processamento de sinais e estatística, podendo vir a ser usada no contexto que nos for pertinente. Estas técnicas são particularmente úteis no âmbito de classificação ou agregação/*clustering* de documentos, especialmente dada a quantidade de *features* serem, no contexto deste tipo de dados, numerosos.

T-Distributed Stochastic Neighbor Embedding

A *T-Distributed Stochastic Neighbor Embedding*, técnica que consiste na redução não linear da informação descrita por um número elevado de dimensões, é útil para incorporar dados descritos por diversas dimensões por forma a que possamos ter uma visualização dos mesmos num espaço de baixa dimensão, por norma, de duas ou três dimensões. Mais especificamente, cada objeto descrito inicialmente por muitas dimensões, é modelado por um ponto num espaço de poucas dimensões (duas ou três) de tal modo que objetos semelhantes sejam modelados por pontos relativamente próximos e, objetos dissemelhantes sejam modelados por pontos distantes.

Este algoritmo é composto em duas fases. Numa primeira, constrói-se a distribuição de probabilidade condicionada sobre pares de objetos caracterizados por um número elevado de dimensões de tal modo que os objetos semelhantes tenham maior probabilidade condicionada de estar perto segundo uma distribuição gaussiana, ao passo que objetos diferentes têm menor probabilidade. Posteriormente, na segunda fase, é definida uma distribuição de probabilidade semelhante à gaussiana mas mais apropriada para um espaço de menor dimensão. Deste modo, esta técnica pode ser útil no contexto da classificação de documentos onde tendencialmente existe um valor elevado de dimensões.

2.2.5 Métricas para avaliação de performance

Precisão , Recall , F-score e Accuracy

A *Precisão P* é uma métrica que avalia a qualidade dos resultados obtidos, isto é, mede a taxa de verdadeiros positivos (*TP*) em relação à soma do número de verdadeiros positivos

e de falsos positivos (FP), ou seja:

$$P = \frac{TP}{TP + FP}$$

O *Recall* R , também designado por vezes como cobertura, mede a taxa de verdadeiros positivos em relação à soma do número de verdadeiros positivos e do número de falsos negativos, ou seja:

$$R = \frac{TP}{TP + FN}$$

A métrica *F-score*, também denominada por *F1-score*, é uma medida que têm em conta simultaneamente a *Precisão* e o *Recall* de um modelo. Sendo calculado pela média harmónica entre a *Precisão* e *Recall*, o seu valor aproxima-se mais do menor dos valores de entre a *Precisão* e o *Recall*:

$$F1 = \frac{2}{recall^{-1} + precision^{-1}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Onde FN representa os falsos negativos. A *Accuracy* como métrica é dada pela seguinte taxa, em que TN representa o número de verdadeiros negativos num *corpus* de documentos D .

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

2.2.6 Métodos de Clustering

A fim de poder comparar a abordagem supervisionada a desenvolver nesta dissertação com uma outra não supervisionada, como foi referido na secção dos objetivos, torna-se necessário obter *clusters* (grupos de documentos) a partir dos dados de entrada, isto é, os documentos caracterizados pelas suas *features*. Existem várias abordagens de *clustering*. Seguidamente indicam-se alguns dos algoritmos disponíveis.

2.2.6.1 K-Means

Este algoritmo tem como objetivo particionar n amostras em k grupos, onde cada observação pertence ao grupo cujo centróide se encontra mais próximo. Este método minimiza a variância dentro dos grupos, recorrendo habitualmente à distância Euclidiana para formular a decisão sobre qual dos grupos é atribuído a cada amostra. Um exemplo do resultado deste método de classificação pode ser observado na Figura 2.1. Contudo, esta abordagem tem duas desvantagens principais no contexto desta dissertação: i) É necessário indicar previamente o número de *clusters*. ii) Com a utilização da heurística distância

Euclidiana para o K-Means os *clusters* serão, teoricamente, hiper-esféricos, tendencialmente de volume semelhante, que nem sempre poderá acontecer, por exemplo para *clusters* com formas/volumes irregulares, sendo uns compactos e outros dispersos. Contudo, esta abordagem, embora simplista, geralmente produz bons resultados.

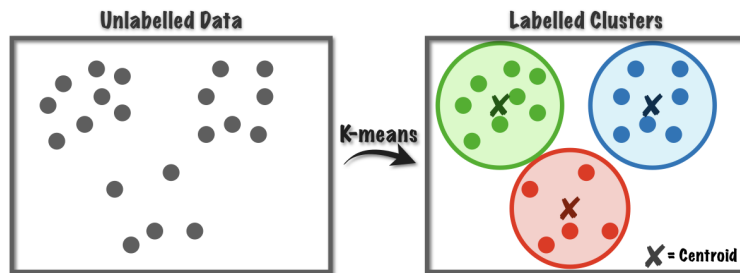


Figura 2.1: Exemplo da formação de *clusters* pelo K-Means.

2.2.6.2 Bisecting K-Means

A *Bisecting K-Means* é uma abordagem que consiste numa modificação do algoritmo K-Means, quer-se com isto dizer, que consegue realizar *clustering* particionado/hierárquico e reconhecer *clusters* de diferentes tamanhos e formas, oferecendo assim uma solução ao principal problema apontado ao algoritmo K-Means. Deste modo, em vez de dividir os dados em K grupos, em cada iteração, o algoritmo divide um *cluster* em dois *sub-clusters* usando o K-Means até o número de *clusters* K ser alcançado. O procedimento utilizado pode ser visualizado na Figura 2.2.

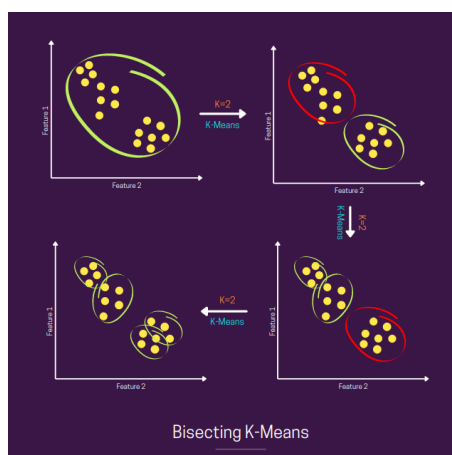


Figura 2.2: Exemplo da formação de *clusters* pelo *Bisecting K-Means*.

EM (Expectation maximization)

Este é um algoritmo que pode ser consultado com maior detalhe em [8] e consiste num

método iterativo para estimar parâmetros em modelos estatísticos, quando o modelo depende de variáveis não observadas. A iteração do EM alterna entre o passo de expectativa (E), e o passo de maximização (M). Relativamente à primeira, etapa de expectativa (E), é então criada uma função para a expectativa da verossimilhança logaritmica usando a estimativa atual para os parâmetros, enquanto que a etapa de maximização (M), calcula os parâmetros que maximizam a verossimilhança logaritmica encontrada na etapa E. Este processo é realizado repetitivamente até que se realizem o número de iterações necessárias. Na Figura 2.3 é possível observar-se o processo utilizado pelo EM.

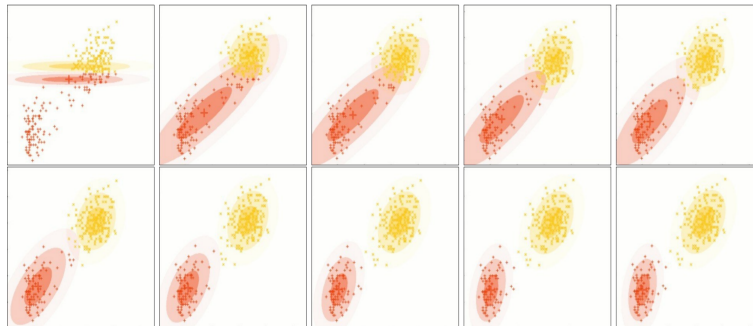


Figura 2.3: Exemplo ilustrativo do algoritmo EM.

2.2.7 Algoritmos classificadores

Sendo esta dissertação focada principalmente numa abordagem supervisionada, será necessário escolher o classificador que melhor performance apresentar face às amostras com que vai lidar. Apresentam-se abaixo alguns dos classificadores disponíveis.

2.2.7.1 K-Nearest Neighbours

O K-NN é um exemplo de um método classificador de *lazy learning*. Mais especificamente, um método de *instance learning* pois este algoritmo envolve comparar novos objetos a serem classificados com objetos do conjunto de treino. Para tal, compara as classes dos K vizinhos mais próximos e atribui ao objeto a classificar a mais comum. Cada parte da superfície de decisão é linear e composta pelos hiperplanos onde o vizinho mais próximo muda. Naturalmente, à medida que o número de vizinhos K aumenta, o classificador torna-se menos propício a ser influenciado por condições locais. Para implementar o classificador, é necessário primeiro escolher uma função de distância. Normalmente, para *features* numéricas e contínuas usa-se a distância de *Minkowski* ou p -norm definida por:

$$D(\vec{x}, \vec{x}') = \sqrt[p]{\sum_d \|x_d - x'_d\|^p}$$

Dependente do valor de p , poderemos ter a distância de *Manhattan* ($p = 1$) ou *Euclidiana* ($p=2$). Para outros tipos de *features*, existem outras funções de distância como a de *Hamming*.

2.2.7.2 Naive bayes

É um classificador que faz parte da família dos classificadores probabilísticos simples. Baseia-se na aplicação do teorema *Bayes* com fortes/naive suposições de independência entre as *features* no contexto de cada classe. Este classificador é um dos mais simplistas da família de modelos *Bayesianos*, no entanto quando utilizado conjuntamente com a estimativa de densidade de kernel (consultar [4] para mais detalhes), consegue alcançar altos níveis de precisão. Este tipo de classificador é altamente escalável, precisando apenas de um número de parâmetros que é proporcional ao número de features do problema. No contexto desta tese, os resultados a obter por este classificador deverão ser comparados com os produzidos por outros classificadores tendo em conta que não é garantido que, para cada classe, as *features* angariadas venham a ser independentes.

2.2.7.3 SVM (Support Vector Machine) e LR (Logistic Regression)

Na sua versão mais simples, o SVM [6] é um classificador binário não probabilístico. No entanto, é possível recorrer a implementações SVM multi classe. A partir de um conjunto de exemplos de treino, marcados com a classe a que pertencem, de um modo geral, o SVM tenta encontrar uma linha de separação (hiperplano) entre os dados de duas classes. Essa linha procura maximizar a distância entre os pontos mais próximos em relação a cada uma das classes. Este classificador dispõe de vários *kernels* que permitem, através da escolha criteriosa de hiperparâmetros adequados, encontrar soluções que melhor se adaptam à natureza dos dados, resultando em performances elevadas. A regressão logística que é essencialmente um modelo de regressão, pode também ser utilizado para classificação. Este modelo, têm o propósito de obter um hiperplano que consegue separar as diferentes classes. Assim, estes classificadores poderão revelar um especial interesse no contexto desta dissertação, sobretudo para a implementação da classificação por género.

2.2.8 AdaBoost

AdaBoost é uma abreviatura de Adaptive Boosting que é um meta-algoritmo de classificação estatística formulado por *Yoav Freund* e *Robert Schapire*. O algoritmo pode ser usado em conjunto com muitos outros tipos de classificadores, for forma a melhorar a prestação individual de cada um. O que o algoritmo faz é formular uma combinação linear dos classificadores fracos, isto é, classificadores com uma baixa *performance*, em que para cada classificador é atribuído um peso, por forma que a média ponderada das respostas do classificador minimize o erro. Este processo é implementado treinando-se cada classificador com o mesmo conjunto de dados mas dando-se diferentes pesos a diferentes amostras/*samples*, atribuindo um maior peso aos exemplos que foram previamente mal classificados. Para tal, primeiramente é então dado o peso de $w_n = \frac{1}{N}$ a cada exemplo, sendo N definido como $N = \|\text{dataset}\|$. Em seguida, treinamos um classificador para minimizar o erro ponderado do conjunto de treino:

$$J_m = \sum_{n=1}^N w_n^m I(y_m(x^n) \neq t^n)$$

Em seguida, calculamos o erro ponderado do classificador e o peso do classificador no *ensemble* classificador.

$$\epsilon_m = \frac{\sum_{n=1}^N w_n^m I(y_m(x^n) \neq t^n)}{\sum_{n=1}^N w_n}$$

$$\alpha = \ln \frac{1 - \epsilon_m}{\epsilon_m}$$

E assim é usado α para atualizar o peso associado a cada *sample*. A função I retorna 1 se os valores forem diferentes, 0 se forem iguais, podendo desta forma aumentar os pesos dos pontos que foram mal classificados. Em seguida, um novo classificador é ajustado ao conjunto de treino com pesos atualizados, e o processo é repetido até que o erro de treino ponderado seja zero ou maior que 0,5. Para usar o classificador *ensemble* final obtido com AdaBoost, calculamos a soma ponderada dos respostas dos classificadores.

$$f(x) = \text{sign} \sum_{m=1}^M \alpha_m y_m(x)$$

2.2.9 Decision Tree

Árvores de decisão são algoritmos de *machine learning* que dividem recursivamente o *feature space* ou espaço de características de acordo com uma declaração condicional para que a homogeneidade seja maximizada dentro das novas regiões criadas. Naturalmente, uma árvore de decisão completa exibirá uma forma de árvore binária, com cada nó sendo um nó de decisão e cada folha o valor previsto. Um aspecto importante na construção de uma árvore de decisão é, portanto, encontrar o melhor valor *feature value* pelo qual se irá ramificar a árvore e, conseqüentemente, dividir os dados. Isso é feito por meio das chamadas medidas de impureza, que avaliam quão pura é uma divisão, isto é, qual é a uniformidade de classe obtida em cada ramo de saída (folhas) de um nó de decisão. Uma divisão pura, i.e, uma divisão perfeita para um problema de classificação binária gerar dois nós em que cada nó teria apenas instâncias de uma das classes. Embora existam muitas métricas, o índice *Gini* visível na Equação (2.1) é a medida de impureza mais comum, e estudos mostram que há poucas razões para usar alternativas [22].

$$GiniIndex = 1 - \sum_{i=1}^{i=C} (p_i)^2 \quad (2.1)$$

Onde p_i^2 corresponde à probabilidade de um objeto ser classificado como pertencente a uma determinada classe e C corresponde ao número de classes existentes.

Na Figura 2.4 pode-se ver uma árvore de decisão que pretende classificar (*Yes/No*) se determinada pessoa deve ou não praticar ténis baseado na previsão do estado do tempo, na temperatura e na humidade.

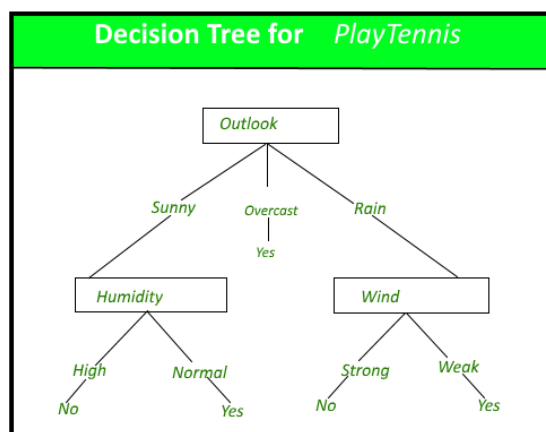


Figura 2.4: Árvore de decisão para prever se o dia é bom ou não para praticar ténis.

2.2.10 *Random Forests*

Florestas aleatórias [3] são um algoritmo de aprendizagem baseado em árvore de decisão (subsecção 2.2.9) que utiliza o conceito de *bagging*, adicionando mais uma camada de aleatoriedade. Além de implementar *subsampling* aos dados de entrada, as florestas aleatórias também fazem uma sub-amostragem dos próprios recursos (conhecido como *random subspace*). Essa extensão decorre do fato de que, se todo o poder preditivo estiver concentrado em apenas alguns subconjuntos de atributos/features, muitas árvores de forma individual poderiam escolher os mesmos atributos para realizar as divisões em cada nó, tornando vários modelos altamente correlacionados entre si. As florestas aleatórias abordam muitos dos problemas em torno das árvores de decisão. A aplicação dos métodos de *bagging* e de *subsampling* tornam o algoritmo robusto para *overfitting*, e ao contrário das árvores de decisão, em que taxas altas de variância nos atributos em utilização tipicamente diminuem a *accuracy* do modelo, as *Random Forests* beneficiam de grandes variâncias entre os dados, visto que a instabilidade é um pré-requisito para *bagging* [3]. Infelizmente, essas melhorias vêm à custa da interpretabilidade dos modelos, pois o número de árvores de decisão aumenta, tornando-se muito difícil entender os motivos por trás de cada decisão. Este pequeno problema, foi de alguma forma disseminado pela capacidade de este classificador estimar a importância de cada atributo por calcular a média de *impurity reductions* de cada *feature* sobre todos os nós em que esta aparece. De forma geral, as *Random Forests* são modelos de *machine learning* robustos, capazes de lidar com diversos formatos de dados, estimando a importância de cada *feature*, capazes de trabalhar com

relações não-lineares, sendo este mais um dos motivos pelo qual o presente classificador é robustos para overfitting. Adicionalmente, a parametrização deste tipo de classificadores no que toca a *hypertunning* é reduzida, o que é um parâmetro importante no contexto temporal da seleção do modelo mais adequado. *Random Forests*, embora não sejam o classificador mais rápido no que concerne ao treino e a realizar previsões, revelam-se eficientes o suficiente para a esmagadora maioria dos problemas e aplicações, desde que o número de estimadores seja ajustado à dimensão do problema.

2.2.11 Redes neuronais

Na sua essência, as *ANN* (Artificial neural networks) procuram simular o comportamento de vários neurónios utilizando um conjunto de entradas, cada uma associada a um peso, juntamente com uma função de ativação. Essas entradas ponderadas são multiplicadas e somadas, e se a sua soma for maior que uma certa *threshold*, então o neurónio irá "disparar". Posteriormente a este processo, os dados podem ser usados para treinar o neurónio, ou seja, atualizar os pesos de forma a disparar apenas quando for o comportamento desejado, usando para tal uma função de erro. Um único neurónio tem um poder de previsão muito limitado, sendo capaz apenas de classificar corretamente classes linearmente separáveis, ou seja, classes que podem ser separadas por um único hiperplano. No entanto, incorporar vários neurónios em uma estrutura de rede em camadas permite a aproximação de qualquer função contínua e a qualquer precisão desejada, conforme declarado pelo teorema da aproximação universal [7]. Estruturalmente, uma rede neuronal pressupõe a existência de uma camada de entrada (que recebe as características dos nossos dados), uma camada de saída (classe/valor previsto) e camadas no meio, denominadas por camadas ocultas. As redes neuronais aprendem de maneira semelhante à de um único neurónio. Uma amostra é passada como entrada, e cada neurónio será acionado, ou não, de acordo com o valor da amostra e o conjunto dos pesos. Uma vez que a camada de saída é alcançada, os pesos de cada neurónio individual serão atualizado por meio do algoritmo de *backpropagation*. Normalmente, as redes neuronais exigem muitas iterações/épocas de treino por forma a conseguirem identificar padrões nos dados. Alguns exemplos de critérios de interrupção, são o número de iterações ou quando o erro de treino é inferior a determinado valor definido como *threshold*. Nas últimas décadas, as redes neuronais popularizaram-se em grande escala devido à imensa variedade de problemas que podem resolver, e subsequentemente à capacidade de serem atualizados em tempo real e aos baixos requisitos de memória, desde que já tenham decorrido as fases de treino. Apesar das vantagens referidas, o treino de uma rede neuronal pode consumir muito tempo e requer, geralmente, grandes *datasets* que muitas vezes não possuímos. Mais se acrescenta que a escolha de uma estrutura de rede correta pode revelar-se uma tarefa árdua e exigir muito tempo devido à hiper-afinação dos diversos hiper-parâmetros da rede.

2.2.11.1 Otimizadores

Por forma a produzir resultados, as variáveis ou parâmetros de uma rede neuronal têm que ser adaptados, por forma a minimizar uma função de custo, pois só assim é que a rede poderá aprender informações presentes nos dados. A questão prende-se então em saber qual é a melhor forma para modificar estes pesos, para tal foram surgindo ao longo dos anos abordagens diferentes.

2.2.11.2 Stochastic Gradient Descent

Uma das formas possíveis de treinar um só neurónio para que este execute classificação binária, por exemplo, consiste na minimização do erro quadrático entre a resposta do neurónio e a classe da resposta. Desta forma procura-se minimizar a seguinte função de erro:

$$E = \frac{1}{2} \sum_{j=1}^{j=N} (t^j - s^j)^2 \quad (2.2)$$

Assim é possível, de forma semelhante ao princípio inicial usando no *perceptron* ajustar os pesos do neurónio em pequenos passos em função do erro para cada exemplo j , $E^j = \frac{1}{2}(t^j - s^j)^2$, onde t^j corresponde a classe do exemplo j e s^j a resposta do neurónio para o exemplo j . Para fazer isto, é então necessário calcular a derivada do erro expressa como função dos pesos do neurónio por forma a ser possível calcular como atualizar os pesos do neurónio. Como o erro é uma função da ativação do neurónio para o exemplo j , s^j , a ativação é então uma função da soma ponderada dos pesos dos *inputs* da rede neuronal (net^j) que por sua vez é uma função dos pesos, podemos utilizar a regra da cadeia para a derivada da função composta por forma a obter o gradiente em função de cada peso:

$$-\frac{\partial E^j}{\partial w} = -\frac{\partial E^j}{\partial s^j} \frac{\partial s^j}{\partial net^j} \frac{\partial net^j}{\partial w} \quad (2.3)$$

Onde

$$s^j = \frac{1}{1 + e^{-net^j}} \quad (2.4)$$

$$net^j = w_0 + \sum_{i=1}^M w_i x_i \quad (2.5)$$

$$\begin{aligned} \frac{\partial net^j}{\partial w} &= x, \\ \frac{\partial s^j}{\partial net^j} &= s^j(1 - s^j), \\ \frac{\partial E^j}{\partial s^j} &= -(t^j - s^j) \end{aligned} \quad (2.6)$$

Assim, é possível obter a regra de atualização do do peso i para o neurónio dado o exemplo j (w_i^j) com uma *learning rate* de η da seguinte forma:

$$\Delta w_i^j = -\eta \frac{\partial E^j}{\partial w_i} = \eta(t^j - s^j)s^j(1 - s^j)x_i^j \quad (2.7)$$

Utilizando esta função de atualização, é minimizada a superfície de erro em pequenos passos em direções diferentes de acordo com cada exemplo apresentado à rede. Com exemplos apresentados aleatoriamente, isto corresponde a *stochastic gradient descent*. Uma representação visual desta otimização pode ser vista na Figura 2.5.

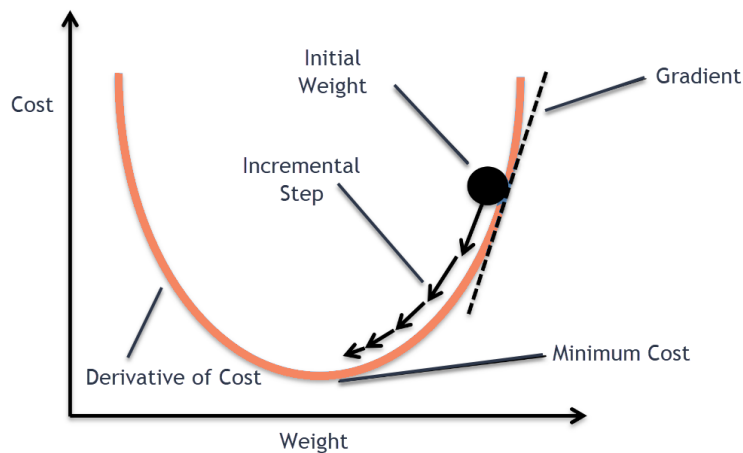


Figura 2.5: Exemplo de Stochastic gradient descent.

2.2.11.3 AdaGrad

Um dos problemas associados ao *stochastic gradient descent* é que a taxa de aprendizagem é constante, prevenindo o otimizador de se adaptar a diferentes condições durante o processo de minimização. Isto é especialmente problemático, quando o gradiente varia de forma diferente ao longo de várias dimensões, o que pode levar a que o *SGD* oscile e não consiga encontrar o caminho para a minimização. A utilização de *momentum* ajuda a resolver este problema, acumulando gradientes de direções de fases anteriores. Isto causa o cancelamento das oscilações e reforça a direção correta, fazendo com que a otimização seja mais rápida. Ainda assim, o problema de se utilizar a mesma taxa de aprendizagem para todos os parâmetros continua e diferentes parâmetros talvez beneficiassem de taxas de atualização diferentes.

O algoritmo *AdaGrad* resolve este problema, pois divide a taxa de aprendizagem de cada parâmetro pela soma dos gradientes quadráticos passados ao parâmetro. Assim, o algoritmo aumenta a taxa de aprendizagem para parâmetros com gradientes relativamente pequenos e reduz a mesma para os que já possuem gradientes elevados, acelerando o processo, com um menor risco de existir um avanço demasiado longo.

2.2.11.4 RMSprop

O *RMSprop* é semelhante ao *AdaGrad*, mas mantém uma média móvel do quadrado dos gradientes e divide o gradiente pela raiz quadrada da média dos quadrados. Assim, é possível equilibrar a magnitude da atualização para cada parâmetro.

2.2.11.5 Funções de ativação

A função de ativação têm essencialmente como objetivo ativar ou não um neurónio consoante o valor fornecido como input. Uma das funções de ativação mais básicas é a *sigmoid*, cuja representação gráfica se encontra na Figura 2.6. O problema associado a esta função é que para valores positivos muito distantes da origem, a função varia pouco, fazendo com que a sua derivada tome valores perto de zero. Com a utilização do algoritmo de *backpropagation*, isto torna-se um problema no sentido em que, embora o neurónio possa estar ativo para um exemplo em particular, todas as derivadas calculadas utilizando esse neurónio serão próximas de zero e como tal não informarão o *gradient descent* para reduzir a função de custo. Com redes formadas por várias camadas de neurónio, este problema torna o treino da rede tão lento, que o torna em contexto prático inviável.

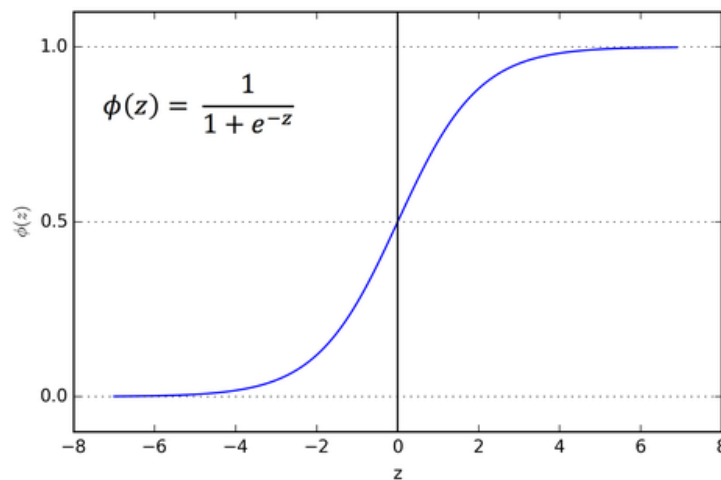


Figura 2.6: Representação gráfica da função *sigmoid*.

Como forma a responder a este problema, surgiu a *ReLU - Rectified Linear Units*, que é uma função linear em cada uma das suas partes, que pode ser na sua forma simplificada definida como:

$$ReLU = \begin{cases} 0 & \text{se } x < 0 \\ x & \text{se } x \geq 0 \end{cases} \quad (2.8)$$

Existem outros tipos de variantes da *ReLU*, tais como a *leaky ReLU* que retorna $a_i x_i$ para valores negativos de x_i e a_i constante. Em 2015, Xu et al compararam a *performance* de diferentes variantes de *ReLU* em redes profundas de convolução no *dataset* de imagens

CIFAR, concluindo que a não utilização da reta $y = 0$ para valores negativos de x melhora a performance sobre a *ReLU* original. Assim, variantes da *Leaky ReLU* são de forma geral, mais usadas nas camadas escondidas de redes profundas.

Quanto a camada de *output*, a função de ativação, de forma geral, depende do tipo de problemática que estejamos a resolver. Para problemas associados a regressões, o neurónio de *output* não deve ter uma função de ativação, retornando assim somente a soma ponderada dos *inputs* multiplicados pelos pesos de cada neurónio. Desta forma, torna-se possível que o neurónio devolva qualquer valor para função de regressão.

Para problemas de classificação binária, é útil ter um valor de *output* que retorne a probabilidade associada ao exemplo pertencer a uma das duas classes. Para este caso, poderemos utilizar a função *sigmoid*. Já para problemas n -ários, a função de ativação mais utilizada é a *softmax* e pode ser definida como:

$$\begin{aligned} \sigma : \mathbb{R}^K &\rightarrow [0, 1]^K \\ \sigma(\vec{x}_j) &= \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}} \end{aligned} \quad (2.9)$$

Esta função retorna um vetor de valores onde $\sigma_j \in [0, 1]$ e $\sum_{k=1}^K \sigma_k = 1$, o que significa que σ_j pode ser interpretado como a probabilidade do exemplo j pertencer a classe k .

2.2.12 Seleção de modelos

Underfitting and Overfitting

Se um modelo não tiver a capacidade de se ajustar ao conjunto de treino, os erros de treino e teste tendem a ser altos porque nenhuma hipótese pode ser instanciada de forma a refletir de forma precisa a relação entre os atributos e a sua classe. A este fenómeno é chamado desubajustamento/*underfitting*. Por forma a resolver este problema, é então necessário substituir o modelo por um que se adapte aos dados, tendo em conta as classes. Desta forma, o erro de treino e de teste serão reduzidos. No entanto, melhorar o ajuste entre o modelo e o conjunto de treino eventualmente começará a aumentar o erro de teste, mesmo que o erro de treino diminua. A este fenómeno é dado o nome de *overfitting*, que é devido ao modelo se adaptar a detalhes do conjunto de treino que não generalizam para o universo a partir do qual os dados foram amostrados. Assim, uma forma possível para abordar o problema de *overfitting* consiste em determinar qual é o modelo, de entre os vários produzidos, que obtém a melhor *performance* no conjunto de validação.

Cross-Validation/ Validação cruzada

Conforme mencionado anteriormente uma maneira simples de resolver o problema de *overfitting*, consiste em selecionar a hipótese que obteve o menor erro de validação. Para fazer isso, é dividido o conjunto de dados em um conjunto de treino e validação (e, se desejado, poder-se-á dividir em mais um conjunto de teste para estimar o verdadeiro erro

da hipótese selecionada). No entanto, todas estas estimativas são amostras aleatórias de alguma distribuição de probabilidade e podemos melhorá-las calculando a média de várias repetições. Além disso, fazer a validação dessa forma só nos permite avaliar hipóteses específicas e não os modelos em si. A utilização de *Cross-Validation* resolve os problemas mencionados. Para fazer a validação cruzada, dividimos os dados em vários conjuntos disjuntos entre si. Por exemplo, se $\|dataset\| = n$ e quisermos usar validação cruzada com k conjuntos, colocamos em cada conjunto n/k elementos. Desta forma, treinamos o nosso modelo em cada instante com $k-1$ conjuntos e validamos os resultados obtidos no conjunto que ficou de fora. Este processo iterativo é repetido k vezes como se pode ver na Figura 2.7, ficando em cada iteração um conjunto de validação diferente de fora. No final das k iterações, realizamos a média dos erros de validação e temos assim uma estimativa do verdadeiro erro que, em média, as hipóteses geradas por este modelo terão neste tipo de dados.

4-fold validation (k=4)



Figura 2.7: Exemplo de *cross-validation* com $k=4$.

Leave-One-Out

Um dos casos específicos de *Cross-Validation* é quando $k = \|dataset\|$, como se pode ver na Figura 2.8. Desta forma, em cada uma das iterações, são utilizados para treino todos os elementos do *dataset* menos um, que irá servir para teste. Este processo, embora computacionalmente bastante mais dispendioso quando comparado com o *Cross-Validation* tradicional, goza das seguintes vantagens: não se coloca a questão da aleatoriedade em usar algumas observações para treino *versus* conjunto de validação, pois cada observação é considerada para treino e validação. Portanto, em média, existe menos variabilidade. Existe também menos *bias*, pois o conjunto de treino é composto por $n-1$ observações/*samples* e devido a este fato, existe também maior confiabilidade nos resultados produzidos.

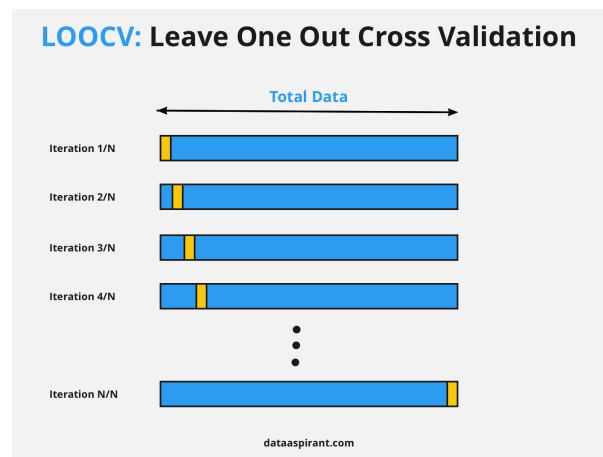


Figura 2.8: Exemplo de *Leave-One-Out*

2.3 Estado da arte

O problema da Atribuição de Autoria faz parte do campo mais generalista da classificação. Com o passar dos anos, novas abordagens aos problemas foram surgindo com o intuito de obter resultados mais promissores. Estas abordagens diferentes dependem essencialmente da estrutura dos dados com que estamos a trabalhar. Desta forma, podemos numa primeira instância decidir sobre se a abordagem a usar será da forma de classificação supervisionada ou não supervisionada. Do ponto de vista de resultados, a classificação supervisionada permite na maioria dos casos a obtenção de melhores resultados quando comparada com a classificação não supervisionada.

2.3.1 Classificação supervisionada

Os autores Xiang Yan e Ling Yan em [28] propuseram um modelo para a classificação de género dos autores de *blogs*. Para tal, utilizaram como atributos unigramas, e um conjunto de propriedades inerente ao *blog* tais como a cor de fundo deste, fonte da letra, pontuação e uso de *emoticons*. Com os atributos angariados foi tentado prever qual o género G , dado uma entrada de blog $B = (C_1, C_2, \dots, C_n)$, onde C_i é uma das características utilizadas e previamente mencionadas. O objetivo é então o calcular de $P(G | B) = P(G)P(B | G)/P(B)$. Os resultados obtidos pelos autores podem ser consultados na tabela reproduzida na Figura 2.9.

Os resultados obtidos não se revelaram muito promissores na medida em que o problema é de classificação binária e a precisão obtida não é suficientemente diferente de uma escolha cega que teria em média cerca de 0.5 de precisão. Os mesmos autores não se focaram em determinar características que permitissem inferir propriedades inerentes ao género, não obstante este é um ponto fundamental para *blogs* cuja dimensão do texto em estudo pode ser muito variável e assim sendo, cada métrica de avaliação textual deve ser utilizada.

	Entries classified as male-blogged	Entries classified as female-blogged
Male-blogged entries	7101	2899
Female-blogged entries	3824	11176

Figura 2.9: Matriz de confusão obtida para identificação de género; conjunto de teste com 10.000 entradas de *blogs* do género masculino e 15.00 do género feminino; precisão de 0.65 , *Recall* de 0.71 , F-measure de 0.68 com α de 0.5. Stop-words não foram removidas.

O estudo realizado por Eric S. Tellez, Sabino Miranda-Jiménez , Mario Graff, e Daniela Moctezuma [26] têm como objetivo a classificação de documentos quanto ao género e idioma de utilizadores da plataforma *Twitter*, contudo somente o primeiro revela interesse para a preparação desta dissertação. Para a concretização do estudo, os autores recorrem ao microTC (μ TC) [25] que é uma *framework* genérica para classificação de texto, i.e., opera independentemente de particularidades inerentes ao idioma em que o documento se encontra escrito. A *framework* μ TC contém os seguintes módulos: i) uma lista de funções que normaliza e transforma o texto presente no documento em *tokens*, ii) um conjunto de funções de *tokens* que transforma o texto pré-processado num multi-conjunto de *tokens*, iii) uma função que retorna um vetor cujas entradas significam os pesos relativos de cada *token*; e finalmente, iv) um classificador capaz de atribuir a cada vetor da etapa anterior uma classe.

Com recurso ao μ TC, todos os *tweets* de cada utilizador passam a ser representados por um vetor de *features* com pesos diferentes. Para o cálculo dos diversos pesos, os autores introduzem o conceito de $entropy_b$, que considera que cada termo é representado por uma distribuição sobre as classes disponíveis (M/F). Desta forma, os pesos são calculados com recurso à $entropy_b$, recorrendo a seguinte fórmula:

$$entropy_b(w) = \log|C| - \sum_{c \in C} p_c(w, b) \cdot \log \frac{1}{p_c(w, b)}$$

Onde C é o conjunto das classes em estudo (M/F) e $p_c(w, b)$ representa a probabilidade de pertença do termo w na classe c parametrizado com b . Mais detalhadamente:

$$p_c(w, b) = \frac{freq_c(w)}{b \cdot |C| + \sum_{c \in C} freq_c(w)}$$

Os classificadores utilizados na etapa iv) são o *Naive Bayes* e o SVM. Os resultados obtidos da classificação por género, podem ser visualizados na Tabela representada na Figura 2.10. Como se pode observar, os resultados obtidos em termos de *accuracy* são bastante bons e consolidados em diversos idiomas.

name	macro-recall	macro-f1	accuracy	improvement
Arabic				
μ TC-FREQ	0.7365	0.7355	0.7369	-
μ TC-TFIDF	0.7190	0.7149	0.7169	↓2.71%
μ TC-entropy+0	0.7030	0.7009	0.7025	↓4.66%
μ TC-entropy+3	0.7591	0.7583	0.7588	↑2.97%
μ TC-entropy+10	0.7577	0.7573	0.7575	↑2.80%
μ TC-entropy+30	0.7460	0.7450	0.7456	↑1.19%
μ TC-entropy+100	0.7259	0.7252	0.7256	↓1.53%
English				
μ TC-FREQ	0.7789	0.7787	0.7788	-
μ TC-TFIDF	0.7750	0.7738	0.7740	↓0.61%
μ TC-entropy+0	0.7626	0.7624	0.7625	↓2.09%
μ TC-entropy+3	0.7897	0.7895	0.7896	↑1.39%
μ TC-entropy+10	0.7788	0.7787	0.7788	0.0%
μ TC-entropy+30	0.7725	0.7725	0.7725	↓0.80%
μ TC-entropy+100	0.7722	0.7721	0.7721	↓0.86%
Spanish				
μ TC-FREQ	0.7364	0.7364	0.7364	-
μ TC-TFIDF	0.7304	0.7294	0.7296	↓0.92%
μ TC-entropy+0	0.7105	0.7092	0.7104	↓3.54%
μ TC-entropy+3	0.7533	0.7527	0.7532	↑2.28%
μ TC-entropy+10	0.7433	0.7430	0.7432	↑0.92%
μ TC-entropy+30	0.7415	0.7411	0.7414	↑0.68%
μ TC-entropy+100	0.7368	0.7366	0.7368	↑0.05%
Portuguese				
μ TC-FREQ	0.8043	0.8034	0.8038	-
μ TC-TFIDF	0.7786	0.7786	0.7788	↓3.11%
μ TC-entropy+0	0.8406	0.8395	0.8400	↑4.51%
μ TC-entropy+3	0.8440	0.8437	0.8438	↑4.98%
μ TC-entropy+10	0.8464	0.8462	0.8463	↑5.29%
μ TC-entropy+30	0.8377	0.8375	0.8375	↑4.20%
μ TC-entropy+100	0.8240	0.8237	0.8238	↑2.49%

Figura 2.10: Resultados obtidos na classificação por género do estudo [26].

A Classificação de Autoria por género, isto é, a Atribuição de Autoria de um documento não a uma pessoa em concreto mas ao seu género baseia-se na premissa que existem características inerentes à produção dos documentos que permitem realizar esta distinção. Estudos como o realizado por Robin Lakoff [18] apresenta um grupo lexical, sintático e pragmático de características que podem ser usadas com o intuito de distinguir o género dos autores. Em [23], Mary Talbot indica que existem padrões na linguagem presente nos documentos que refletem o género do autor. De acordo com o mesmo, mulheres que são subordinadas a homens em termos de trabalho tendem a exibir um uso maior de linguagem educada.

O estudo realizado por Na Cheng, Rajarathnam Chandramouli e K.Subbalakshmi [5] pretende classificar alguns textos de diferentes autores encontrados em aplicações na Internet e responder às seguintes questões:

- As mulheres e os homens usam diferentes estilos de escrita?
- Em caso positivo, quais são as características que os diferenciam?

A estrutura geral desta abordagem seguida pode ser visualizada na Figura 2.11.

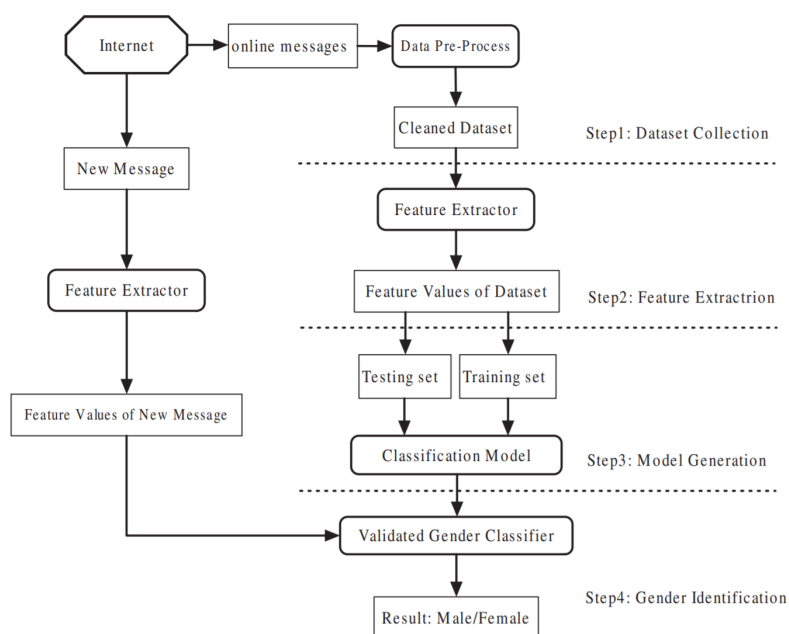


Figura 2.11: Abordagem geral seguida para identificação do género do autor no estudo [5].

Os autores utilizaram como *dataset* tanto o *Reuters newsgroup* como o *Enron newsgroup*. Este último *dataset* consiste num conjunto de reportagens na língua inglesa realizadas por jornalistas no período compreendido entre 20 de Agosto de 1996 a 19 de Agosto de 1997. Para a fase de pré-processamento do texto, foram somente considerados os documentos cujo género pudesse ser inferido pelo nome do autor e cujo comprimento do texto estivesse compreendido entre as 200 e 1000 palavras. Como conjunto de características descritivas dos dados os autores reconhecem, com base em outros estudos, que assentam no pressuposto de que essencialmente existem cinco componentes diferentes que permitem distinguir o género dos autores, que são baseadas em:

- Caracteres utilizados.
- Palavras utilizadas.
- Construção sintática.
- Estrutura.
- Utilização de classes de palavras.

Para este estudo, foram produzidos um total de 545 atributos. Para a fase classificação dos documentos por género foram utilizados os classificadores SVM, *Decicion Tree* e *Bayesian-based logistic regression*. Os resultados obtidos por classificador, podem ser visualizados na Figura 2.12.

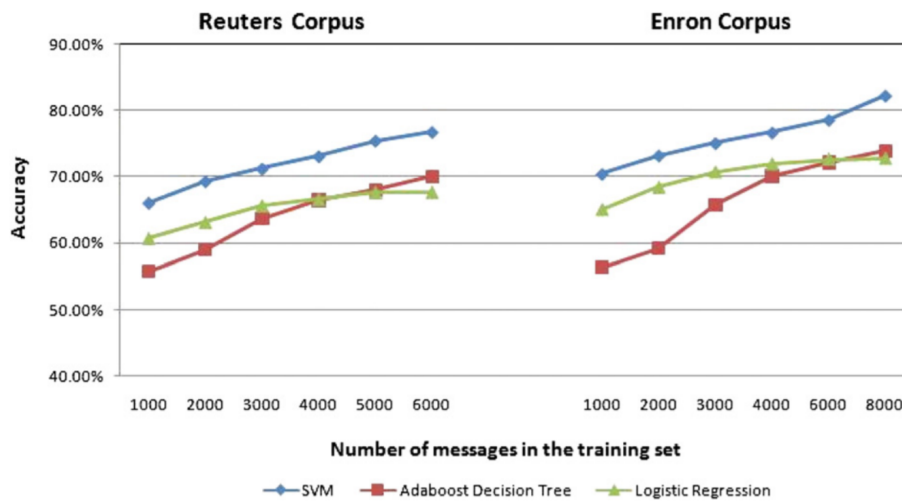


Figura 2.12: Resultados obtidos por classificador no estudo de Na Cheng, Rajarathnam Chandramouli e K.Subbalakshmi [5].

Os resultados obtidos são bastante promissores, atingindo taxas altas de *accuracy*. Outros autores, como é o caso de Sara El Manar El Bouanani e Ismail Kassou [10] realizaram um resumo de diferentes estudos bem como as técnicas utilizadas pelos autores por forma a verificar quais as características que têm um poder discriminativo suficientemente forte para permitir diferenciar entre autores. Os autores defendem que a análise de autoria se divide em:

- Classificação de Autoria: Consiste na determinação da probabilidade de determinado autor ter produzido determinado documento.
- Caracterização de autor: Consiste na determinação do conjunto de características que descreve o autor. Estas características incluem o género, o grau de escolaridade, o grau de cultura geral e a familiarização com a língua.
- Deteção de semelhança: Consiste na comparação de diversos documentos e averigua se foram produzidos por um único autor. Esta abordagem é particularmente útil para efeitos de verificação de plágio.

Estes autores concluem que os resultados obtidos por classificação de autoria são condicionados por alguns parâmetros particularmente, os seguintes: o tamanho do *corpus*, o tamanho dos documentos e o número de autores candidatos. É também salientado o tamanho dos documentos como um dos principais fatores na determinação da abordagem correta do problema. Tal fica a dever-se essencialmente ao facto de que alguns métodos conseguem obter uma melhor precisão de classificação com uma quantidade diferente de informação.

Os autores defendem ainda que o *corpus* de documentos deve ser controlado em parâmetros como idade, nacionalidade e gênero. Adicionalmente, argumentam que os documentos pertencentes ao mesmo autor devem ser sensivelmente do mesmo período, isto para minimizar efeitos como evolução da escrita do mesmo. É ainda referido que o tópico dos documentos que compõem o *corpus* deve ser o mesmo, transmitindo a ideia de que estilos de escrita podem ser também dependentes dos temas abordados.

Ainda que o referido pelos autores faça sentido, no contexto da presente dissertação, que se quer que seja aplicada a contextos difíceis, não fará sentido utilizar *datasets* com as características cujos autores do estudo defendem em [10], pois assim, a tarefa da Atribuição de Autoria seria largamente facilitada, no sentido em que existiria uma redução de parâmetros que poderão influenciar a escrita de cada documento.

Michael Gamon em [12], demonstra uma solução para o problema de classificação de autor díspar à dos autores mencionados supra recorrendo à utilização de técnicas de análise profunda de classificação de atributos em conjunto com algumas abordagens anteriores por forma a verificar se a combinação de diferentes técnicas trará melhores resultados.

O autor começa por enunciar alguns métodos de caracterização do estilo do autor tais como a frequência da utilização de classes de palavras diferentes, tamanho das palavras e das frases.

O *dataset* utilizado é composto por textos produzidos por três irmãs, com o intuito de reduzir fatores como educação, gênero e estilo histórico de escrita, podendo-se assim focar na atribuição de classificação num *dataset* homogêneo do ponto de vista dos parâmetros referidos. Os atributos foram extraídos dos documentos de cada autora com recurso ao sistema NLPWin. O autor, introduz o conceito de *cut-off* para atributos. Na prática, a implementação do conceito traduz-se na restrição para relações existentes no documento cuja frequência absoluta seja inferior a um certo valor (*threshold*), não sejam tidas em conta para efeitos de classificação. Os valores para as (*threshold*) tidos em conta são 5, 10, 20, 50, 75, 100, 200 e 500. Para cada um dos valores de *threshold* foi então aplicado o classificador SVM, tendo sido obtidos os resultados apresentados na Figura 2.13.

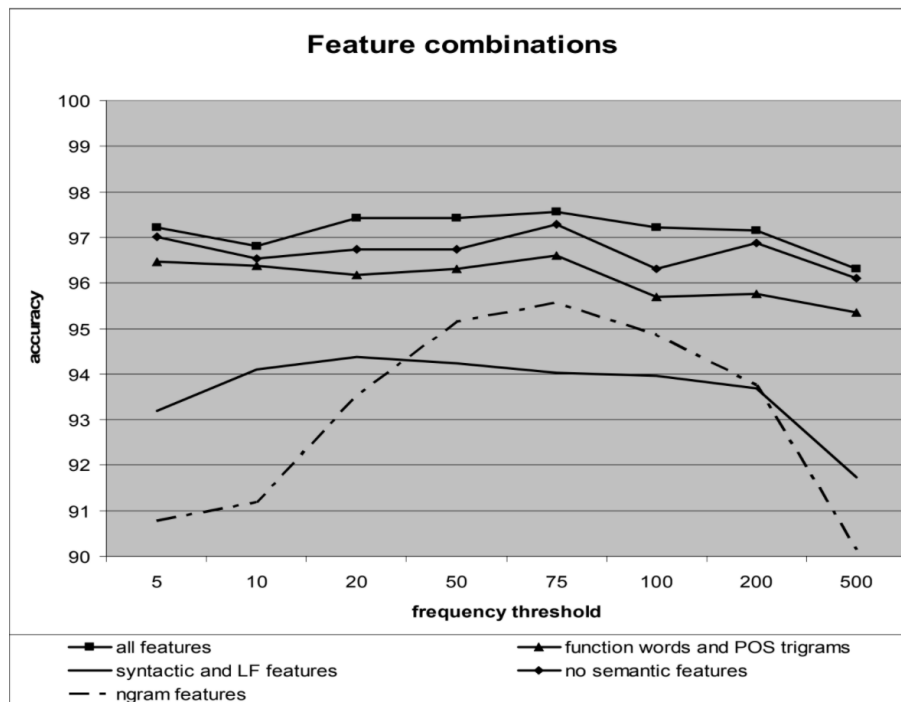


Figura 2.13: Precisão obtida por *threshold* usando diferentes atributos no estudo [12].

O autor conclui assim que o uso de técnicas de análise profunda de classificação de atributos conjuntamente com abordagens mais clássicas aumenta a precisão obtida. No entanto, creio que este resultado deveria ser confirmado com outro *dataset* mais heterogêneo e com mais autores.

Em [1], os autores utilizam cadeias de markov como solução para identificar o ano em que o documento foi escrito, assim como identificar a existência de profundas mudanças na escrita na língua inglesa. Contudo, esta não é uma solução extensível atendendo ao facto de que este tipo de abordagem depende essencialmente da língua em que os documentos estão redigidos.

Outra solução oferecida em [13] passa por tentar decidir se determinado documento foi ou não elaborado por um autor conhecido. Para tal, o documento é processado pelo sistema ISG - *Integrated Syntactic Graph* cujo objetivo é a formação de um grafo que reproduza as várias relações linguísticas presentes no documento. O processamento de três frases pode ver visualizado recorrendo à Figura 2.14. As frases avaliadas são as seguintes: "I'm going to share with you the story as to how I have become an HIV/AIDS campaigner", "And this is the name of my campaign, SING Campaign.", "In November of 2003 I was invited to take part in the launch of Nelson Mandela's 46664 Foundation".

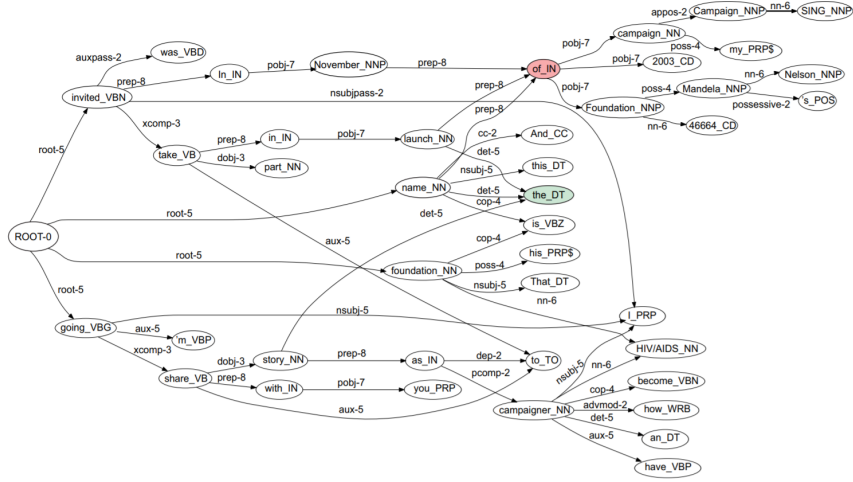


Figura 2.14: Grafo resultante do processamento de três frases produzido pelo sistema ISG.

Para a comparação de documentos é, então, depois utilizada uma métrica de semelhança entre os grafos de cada par de autores.

$$\begin{aligned}
 \text{Similarity}(D^1, D^2) &= \sum_{n=1}^m \text{Cosine}(f\vec{D}_{1,i}, f\vec{D}_{2,i}) \\
 &= \sum_{i=1}^m \frac{f\vec{D}_{1,i} \cdot f\vec{D}_{2,i}}{\|f\vec{D}_{1,i}\| \cdot \|f\vec{D}_{2,i}\|} \\
 &= \sum_{i=1}^m \frac{\sum_{j=1}^{|V|} (f(D1,i),j) \times (f(D2,i),j)}{\sqrt{\sum_{i=1}^m (f(D1,i),j)^2} \times \sqrt{\sum_{i=1}^m (f(D2,i),j)^2}}
 \end{aligned}$$

Onde D_i é um grafo, e m é o número de vetores existentes no grafo.

Baseando-se no resultado obtido pela métrica é então depois feita a classificação, indicando se o documento foi ou não produzido pelo autor.

No paper produzido por Fatma Howedi [15], a autora segue uma abordagem mais clássica com excelentes resultados. É realizado a extração de *features* (para maior detalhe consultar a Figura 2.15) baseando-se essencialmente em :

- *Word N-gram* : Frases de tamanho N.
- *Character N-grams* : Cadeias de caracteres de tamanho N.
- *Rare Words* : Palavras raras.

Feature	N-grams level	Description	Feature Type
Character Uni-gram	1-gram	individual characters	Character
Character Bi-gram	2-grams	two consecutive characters	
Character Tri-gram	3-grams	three consecutive characters	
Character Tetra-gram	4-grams	four consecutive characters	
Word Uni-gram	1-gram	single words	Lexical
Word Bi-gram	2-grams	two consecutive words	
Word Tri-gram	3-grams	three consecutive words	
Word Tetra-gram	4-grams	four consecutive words	
Rare Words		low frequency	

Figura 2.15: Atributos utilizados no estudo de Fatma Howedi [15].

Após esta fase inicial de recolha de *features*, as mesmas são submetidas a alguns testes para verificar o poder descritivo das mesmas no conjunto de dados. Os testes utilizados no estudo foram *Chi-Squared* e *Information Gain*. Como classificadores foram usados quer o SVM quer o *Naive Bayes*, tendo-se obtido os seguintes resultados visíveis na tabela representada pela Figura 2.16.

Feature	Accuracy of good attribution using NB Classifier	Accuracy of good attribution using SVM Classifier
Character Uni-gram	53.33%	50.00%
Character Bi-gram	60.00%	63.33%
Character Tri-gram	93.33%	86.67%
Character Tetra-gram	93.33%	93.33%
Word uni-gram	96.67%	76.67%
Word Bi-gram	76.67%	56.67%
Word Tri-gram	40.00%	20.00%
Word Tetra-gram	40.00%	26.67%
Rare Words	93.33%	93.33%
Average of the all used features	71.85%	62.96%

Figura 2.16: Resultados obtidos por classificador em [15].

Como é possível observar, a precisão obtida utilizando palavras de tamanho um foi acima de 95% o que é um excelente resultado. É ainda sublinhada a importância do poder discriminativo que a pontuação pode ter em atributos baseados em caracteres, tendo-se verificado que em média a precisão aumenta quando estes são incluídos.

Abordagens como a de Arjun Mukherjee e Bing Liu em [20] são diferentes das técnicas tradicionais de resolução de classificação de género. Este estudo pretende utilizar uma nova classe de atributos que são dependentes de relações entre o conteúdo textual - *POS (Part-Of-Speech)*, mas sendo diferente da abordagem tradicional de *N-grams*, pois são de tamanho variável e têm que satisfazer cumulativamente algum critério. Adicionalmente, os autores propõem uma nova técnica híbrida de seleção de atributos, pois é do entendimento dos mesmos que a utilização de uma só técnica e determinados critérios é sempre enviesada para um determinado tipo de atributos.

Os métodos utilizados para a angariação de atributos foram o *mine-POS-pats* e o *EFs (Ensemble Feature Selection)*, sendo que este último classifica os atributos e utiliza uma

métrica por forma a verificar se os mesmos serão úteis. Após esta seleção inicial, os autores ficam com um conjunto grande de atributos e utilizaram técnicas como IG - *Information Gain*, *Mutual Information*, *Xi-Square* e *Cross Entropy* para reduzir o número de atributos.

Após esta fase, os classificadores utilizados foram o NB - *Naive Bayes*, SVM - *Support Vector Machine* e SVM *Regression*.

Os resultados obtidos por método utilizado podem ser vistos na tabela da Figura 2.17.

Feature Selection	Value Assignment	NB	SVM	SVM_R
IG	Boolean	71.32	76.61	78.32
IG	TF	66.01	72.84	74.13
MI	Boolean	72.01	78.62	79.48
MI	TF	70.86	73.14	74.58
χ^2	Boolean	72.90	80.71	81.52
χ^2	TF	71.84	73.57	75.24
EFS	Boolean	73.57	86.24	88.56
EFS	TF	72.82	82.05	83.53

Figura 2.17: Resultados obtidos por *Feature Selection* e *Value Assignment* em [20].

Técnicas de classificação não dependentes do idioma tendem a obter resultados mais modestos, uma vez que não utilizam características inerentes às mesmas. Neste sentido, os autores são Filipa Peleja et. al. em [21] realizam uma comparação entre diversos classificadores. O estudo dos autores é interessante no sentido em que explora uma técnica de angariação de atributos pouco utilizada no contexto do estado de arte que têm o nome de 3M (ver [21] para mais detalhes). Os autores concluem assim que a utilização deste técnica permite impulsionar o uso de atributos raros, permitindo assim uma maior representação mais específica do conjunto de palavras que são atributos do estudo, atingindo assim uma performance sem precedentes quando comparado com outras métricas de seleção de atributos.

Um estudo produzido por João F. Teixeira e Marco Couto [24], ambos investigadores da do laboratório de investigação *High-Assurance Software Laboratory (HASLab)* da Universidade do Minho, realiza a classificação em contextos difíceis de dois heterónimos de Fernando Pessoa. Para tal, utilizaram o repositório Pessoa, disponível em <http://www.dominiopublico.gov.br>. Este repositório contém somente documentos completos e cuja autoria tenha sido atribuída de forma clara a um dos vários heterónimos. Para tal, começaram por seleccionar os heterónimos Ricardo Reis e Álvaro de Campos e os seus respetivos documentos no *dataset*. Após este passo, realizaram o pré-processamento de texto da seguinte forma:

- *S1 - Tokenization*. - Cada documento foi primeiramente transformado numa sequência

de termos de tamanho unitário. Posteriormente, cada termo existente no *corpus* foi extraído para um *Bag-of-Word (BoW)* independente da sua ordenação.

- *S2 - Casing Transformation.* - Cada carácter foi transformado para minúscula, por forma a reduzir o número de termos diferentes. Assim, os termos "Não" e "não", ficam reduzidos ao mesmo - "não".
- *S3 - Length Filtering.* - Foram removidos termos cujo comprimento é inferior a quatro e maior que quinze.
- *S4 - Stemming.* - A transformação de termos na sua raiz. (Ver a subsecção 2.2.2 para maior detalhe.)
- *S5 - Stopword.* - Foram removidas todas as marcas de pontuação, conectores e palavras específicas definidas pelos autores.

Por forma a estimar o poder discriminante que cada termo tem com o intuito de distinguir ambos os autores, os autores aplicaram métricas tais como :

- *Binary Term Occurrence - BTO* - Métrica que devolve o número de documentos no *corpus* em que determinado termo é utilizado.
- *Term Occurrence - TO* - É uma métrica que calcula o número de vezes que determinado termo é utilizado por documento.
- *Term Frequency - TF* - Métrica que avalia o número de ocorrências de determinado termo num determinado documento, tendo em conta o tamanho do mesmo.
- *TF-IDF* - Métrica que, de forma geral, retorna $TF * Inverse Document Frequency (IDF)$.

Posteriormente à angariação de atributos e estimação do seu poder discriminante, foi utilizado o SVM - *Support-Vector-Machine* como classificador, tendo-se obtido os resultados visíveis na tabela da Figura 2.18.

Binary TO			TF			TF-IDF		
Acc	F1 _{RR}	F1 _{AC}	Acc	F1 _{RR}	F1 _{AC}	Acc	F1 _{RR}	F1 _{AC}
66,20	76,47	40,00	92,96	93,83	91,80	97,18	97,44	96,88

Figura 2.18: Resultados obtidos com o *dataset* pessoa em [24].

Muito embora os resultados obtidos sejam bastante bons, com o melhor modelo a ter uma *accuracy* $\approx 97\%$, a complexidade associada à abordagem seguida está longe de ser a ideal. Para este tipo de solução, que depende do peso associado a cada termo, é fulcral classificar muitas palavras e passar muitas features ao classificador, tornando assim o processo lento e praticamente inviável em conteúdo prático. Acresce que os autores apenas consideraram dois dos heterónimos disponíveis.

No contexto da presente tese, não será possível testar a solução produzida pelos autores, pela falta de recursos computacionais com as características necessárias. No

entanto, devido à dificuldade associada ao *dataset* ser elevada, será interessante observar qual é a *performance* da solução proposta na presente dissertação neste *dataset* cujo autor, em rigor, é o mesmo para todas as classes.

2.3.2 Classificação não supervisionada

A utilização de técnicas de classificação não supervisionada difere essencialmente da classificação supervisionada pois os dados não se encontram classificados, isto é, com a notação de classe. Desta forma, numa fase inicial não existe a informação prévia do número de *clusters*/grupos existentes. Assim, terá que ser necessário também inferir o número de grupos da melhor forma possível de acordo com os dados.

Em [11], o objetivo é a criação de um modelo que pretende resolver o problema de questão-pergunta por forma a que perguntas semelhantes possam ser agrupadas e respondidas de uma só vez, evitando assim duplicados de resposta. O conjunto de dados contém mais de 1300 questões de escolha múltipla agrupadas por (questão, par) incluindo a sua explicação, nível de dificuldade e média da qualidade das respostas obtidas pelos estudantes. O modelo utilizado para a classificação foi o LDA - *Latent Dirichlet Allocation*. Este modelo revela algumas propriedades interessantes, pois não é, em termos rigorosos, um algoritmo de *cluster*. Isto deve-se ao facto de que os algoritmos de *cluster* classificam cada amostra como pertencente a um determinado grupo, no entanto, o LDA o que faz é produzir uma distribuição de grupos sobre as amostras a serem agrupadas. Para a formulação do número ideal de tópicos de questões/grupos foi utilizado o HDP - *Hierarchical Dirichlet Process*.

Melhores resultados poderiam ser obtidos se cada pergunta se encontrasse já categorizada de acordo com alguns tópicos relacionados. Assim, um número certo de tópicos já poderia ser inferido. Claro que ao obtermos esta informação estaríamos a cair no escopo de classificação supervisionada. No entanto, esta revela resultados melhores na maioria dos casos.

O estudo realizado por Ms. Meghana. N.Ingole, Mrs.M.S.Bewoor e Mr.S.H.Patil [16] pretendeu realizar a tarefa de sumarização de texto, que exigiu, numa fase inicial, a realização do processamento de texto através do seguinte procedimento: dividir por frases, *Tokenization*, POS, *Chuncking* e *Parsing*.

Após esta fase estar concluída, é utilizado o EMS - *Expectation Maximization Clustering*, onde são formados os *clusters*. Seguidamente, é formalizado o grafo de documentos e assim, calculada a matriz de semelhança.

De acordo com o trabalho de Youngjoong Ko e Jungyun Seo em [17], a solução encontrada para a classificação de documentos, passa pela criação de um conjunto de treino para todas as categorias. Após esta fase, as características são extraídas e usado como algoritmo de classificação o *Naive Bayes*.

O procedimento geral pode ser visualizado na Figura 2.19.

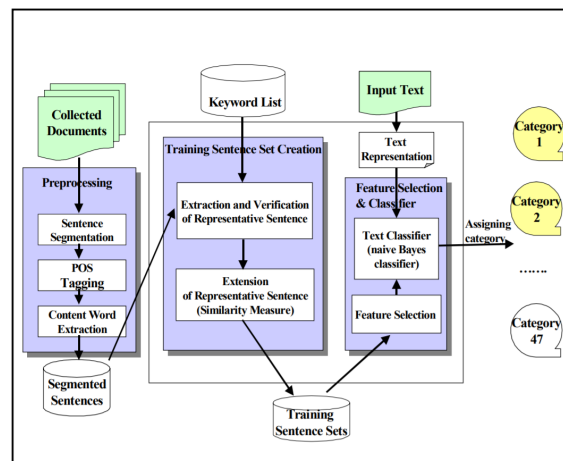


Figura 2.19: Ilustração da metodologia seguida em [17].

Esta abordagem não se enquadra no contexto da presente tese pois fica limitada a categorias previamente criadas, além do mais, esta abordagem não é independente da língua utilizada.

SOLUÇÃO PROPOSTA

3.1 Classificação em contextos difíceis

3.1.1 Introdução

No contexto da utilização de dados estruturados, as *features* são muitas vezes selecionadas a partir dessa estrutura e podemos fazer uso destas para a resolução de determinado problema envolvendo o conjunto de dados/*dataset*. Por outro lado, no caso de dados não estruturados é necessário encontrar atributos que têm que ser angariados no seio do texto sem referências a qualquer estrutura. No contexto particular da caracterização de autoria, será necessário encontrar *features* com um poder discriminante suficiente por forma a ser possível caracterizar cada classe e diferenciá-la das demais. Naturalmente, esta foi uma das principais dificuldades sentidas ao realizar a presente dissertação, pois as diferenças de escrita entre autores são muito ténues e muitas vezes somente detetadas através da compreensão e do conjunto de vivências humanas. É possível, por exemplo tentar detetar estas diferenças com ferramentas como a análise de sentimentos ou polaridade frásica, no entanto, estas ferramentas não são independentes do idioma em que os documentos foram produzidos, pelo que saem fora do escopo da presente tese.

3.1.2 *Features* suficientemente discriminantes

A extração de *features* sobre documentos compostos por texto, sem a utilização de métodos dependentes do idioma recai essencialmente sobre a extração de atributos estatísticos, i.e atributos que descrevem essencialmente com que frequência determinado autor tende a repetir um determinado padrão. A título de exemplo, pode-se observar na Tabela 3.1 dois documentos provenientes do mesmo *dataset* cuja classe não é a mesma, muito embora à primeira vista nos pareçam praticamente indistinguíveis.

Tabela 3.1: Exemplo de dois documentos produzidos por 2 heterónimos.

Diana através dos ramos Espreita a vinda de Endymion Endymion que nunca vem, Endymion, Endymion, Lá longe na floresta. . . E a sua voz chamando Exclama através dos ramos Endymion, Endymion. . . Assim choram os deuses. . .	No tempo em que festejavam o dia dos meus anos, Eu era feliz e ninguém estava morto. Na casa antiga, até eu fazer anos era uma tradição de há séculos, E a alegria de todos, e a minha, estava certa com uma religião qualquer.
---	--

Sendo que uma das premissas para a realização de uma qualquer tarefa de classificação, assenta no pressuposto da existência de elementos inerentes à amostra capazes de, em primeira instância descreverem-na e em segunda instância de a distinguir de entre as demais pertencentes a classes diferentes, foram angariados os seguintes atributos, conforme consta na Tabela 3.2.

Tabela 3.2: Atributos utilizados na proposta de solução seguida.

Feature/ Atributo	Descrição
9-char	Número de palavras por documento cujo comprimento é maior ou igual a nove.
6-char	Número de palavras por documento cujo comprimento é maior ou igual a seis.
3-char	Número de palavras por documento cujo comprimento é menor que três.
5-char	Número de palavras por documento cujo tamanho está compreendido entre três e cinco.
2-char	Número de palavras por documento cujo tamanho é igual a dois.
uni-grams	Frequência do uni-grama mais repetido.
bi-grams	Frequência do bi-grama mais repetido.
tri-grams	Frequência do tri-grama mais repetido.
quad-grams	Frequência do quad-grama mais repetido.
variância-frase	A variância silábica de cada frase.
variância-documento	A variância do tamanho de cada documento em número de palavras.
virgulas	Frequência da utilização de virgulas normalizado.
pontos	Frequência da utilização de pontos normalizado.
hifen	Frequência da utilização de hifens normalizado.
not ascii	Frequência de caracteres não ascii no documento.
maiúsculas	Frequência de utilização de caracteres maiúsculos no documento.
tamanho-medio-palavra	Comprimento médio de cada palavra.
tamanho-frase	Comprimento médio de cada frase.
Exclamação	Frequência da utilização de pontos de exclamação normalizado.
Interrogação	Frequência da utilização de pontos de interrogação normalizado.
ponto e virgula	Frequência da utilização do carácter ";" normalizado.
palavras entre virgulas	Número médio de palavras entre duas virgulas.
palavras entre pontos de interrogação	Número médio de palavras entre pontos de interrogação.

Após a angariação das *features*/atributos surge então a questão de quais terão um maior poder discriminante para cada *dataset*. Como forma de ajudar a responder à questão supra-mencionada, foi então utilizada uma métrica de nome $D(A)$ que pretende averiguar qual o poder de diferenciar um atributo A .

A métrica calcula a capacidade de cada atributo para discriminar através do quociente entre a variância da média do atributo por classe, e a médias das variâncias do atributo por classe, conforme pode ser visto com maior detalhe na Equação (3.1). Naturalmente, o poder de cada *feature* irá variar consoante o *dataset*, podendo existir casos em que determinado atributo seja bastante discriminativo entre as classes e outros em que não.

$$D(A) = \frac{VM(A)}{MV(A)} = \frac{\frac{1}{\|G\|} \sum_{g \in G} (M(A, g) - M(A, \cdot))^2}{\frac{1}{\|G\|} \sum_{g \in G} \frac{1}{\|g\|} \sum_{d \in g} (fr(A, d) - M(A, g))^2} \quad (3.1)$$

Onde $M(A, \cdot)$ representa a média dos valores do atributo A por classe, ou seja:

$$M(A, \cdot) = \frac{1}{\|G\|} \sum_{g \in G} fr(A, d) \quad (3.2)$$

Sendo $fr(A, d)$ a frequência relativa do atributo A no documento d .

Adicionalmente, ao estimarmos o poder discriminante que cada atributo têm para a distinção entre classes, podemos utilizar, sempre que se releve necessário, esta informação da seguinte forma: Seja d_A o valor dado pela Equação (3.1) para a *feature* A , seja ainda a matriz C constituída por um conjunto de colunas, cada uma correspondente a um atributo, e por um conjunto de linhas, cada uma correspondente a um documento; para cada atributo A da matriz C podemos transformar cada elemento $x(A, i)$ correspondente à linha i e portanto ao documento d_i , tal que $x(A, i) = fr(A, d_i) * D(A)$.

Esta métrica também será utilizada no contexto da presente dissertação para calcular o valor de uma medida baseada no 3º momento estatístico (*Skewness*), a que chamamos $Sk(A)$ nesta dissertação, pois basta para tal que o expoente utilizado no numerador da Equação (3.1) seja igual a 3 e transformando o denominador para $VM(A) = \frac{1}{\|G\|} \sum_{g \in G} \frac{1}{\|g\|} \sum_{d \in g} (|fr(A, d) - M(A, g)|)^3$.

3.2 A classificação de documentos

Considerando os objetivos delineados para a realização desta dissertação, tanto o reconhecimento de autores, como a identificação do género a partir de texto não estruturado serão trabalhados principalmente em contexto de classificação supervisionada. Pretende-se de igual modo, dotar a abordagem da capacidade de rejeitar documentos de autores desconhecidos.

Deste modo, pretende-se com a presente dissertação implementar um sistema capaz de atribuir a autoria de determinado documento ao seu autor, para que tal seja concretizável é necessário numa primeira instância seleccionar os atributos que descrevem de forma única o autor, para tal dividimos o *corpus* de cada autor e seleccionamos 80 por cento para treino e o restante para testes. Após a selecção das características, verifica-se o poder discriminativo das mesmas entre os autores em estudo. Neste ponto em particular, os autores passam a ser categorizados por uma matriz de atributos que poderá conter dados que podem não se revelar particularmente úteis para a resolução do problema, pelo que devemos de igual forma realizar a redução de atributos sem perder a capacidade discriminante da matriz original.

3.3. CLASSIFICAR DOCUMENTOS UTILIZANDO O QUADRADO DA DISTÂNCIA DE MAHALANOBIS.

Após esta fase, reúnem-se as condições necessárias para classificar novos documentos não pertencentes ao conjunto de documentos de treino, sendo adotado um processo válido para qualquer problema de classificação e que será usado no âmbito desta tese quer para a Atribuição de Autoria, quer para a atribuição de género dado que estas diferem somente pela escolha dos atributos a serem escolhidos, uma vez que podem não ser os mesmos.

O funcionamento geral do processo de classificação utilizando os classificadores clássicos de uma forma simplificada poderá ser observado na Figura 3.1.

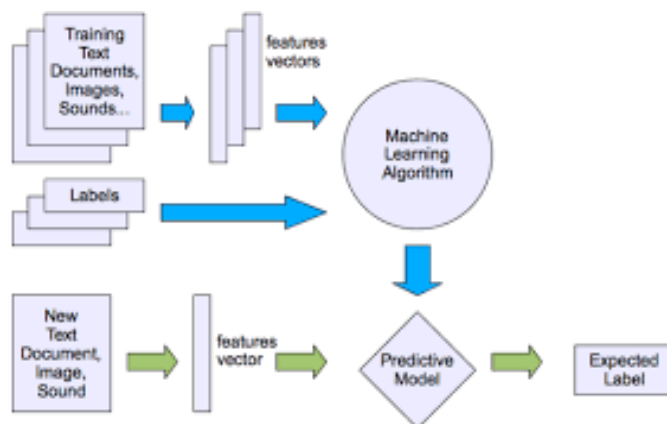


Figura 3.1: Processo geral de classificação adotado.

3.3 Classificar documentos utilizando o quadrado da distância de Mahalanobis.

O quadrado da distância de *Mahalanobis* conforme descrita de forma mais detalhada no Capítulo 2, pode ser utilizada para o processo de classificação de documentos e conseqüentemente para a Atribuição de Autoria. Sendo a distância definida por $M^2(\vec{p}, \vec{\mu}, \Sigma^{-1}) = (\vec{p} - \vec{\mu})^T \Sigma^{-1} (\vec{p} - \vec{\mu})$, é então necessário realizar os seguintes passos:

- Construir o ponto \vec{p} em função das *features* em utilização.
- Construir os centroides para cada classe em função das *features* em utilização.
- Construir a matriz de covariâncias em função das *features* em utilização.
- Construir $\vec{\mu}$ como sendo o vetor médio dos atributos em estudo.

Seja $\|K\|$ o número de classes existentes em determinado *dataset* e seja $\|F\|$ as *features* em utilização. Construir o ponto \vec{p} resume-se a selecionar das coordenadas dos atributos utilizados. Assim, $\vec{p} = [f_1, f_2, \dots, f_{\|F\|}]$ em que cada f_i corresponde ao valor da *feature* i para \vec{p} . Cada classe existente no *dataset* terá como centroide o vetor $\vec{\mu}_i$ com $i \in \{1, \dots, \|K\|\}$ que será construído a partir da média de cada atributo em utilização em cada classe e terá a forma $\vec{\mu}_i = [\overline{f_1^{k_i}}, \overline{f_2^{k_i}}, \overline{f_3^{k_i}}, \dots, \overline{f_{\|F\|}^{k_i}}]$. Finalmente, a matriz de covariâncias Σ^{-1} está

associada as *features* que caracterizam documentos de uma dada classe, normalmente autor ou género. É estimada pela matriz de covariância \vec{E}_i , com base na amostra feita por os documentos (os documentos de treino) do *cluster* i , da seguinte forma:

$$\vec{E}_i = \begin{bmatrix} E_{1,1} & E_{1,2} & \dots & E_{1,\|F\|} \\ E_{1,2} & E_{2,2} & \dots & E_{2,\|F\|} \\ \vdots & \vdots & \ddots & \dots \\ E_{1,\|F\|} & E_{2,\|F\|} & \dots & E_{\|F\|,\|F\|} \end{bmatrix}$$

Sendo n o número de *features* e um elemento genérico de \vec{E}_i descrito da seguinte forma:

$$E_{l,m} = \frac{1}{\|g_i\|} \sum_{d \in g_i} (x(l, d) - x(l, .))(x(m, d) - x(m, .))$$

Onde g corresponde ao grupo dos documentos da classe i , e $x(l, .)$ ao valor médio da componente/atributo l para os documentos pertencentes a classe i , isto é:

$$x(l, .) = \frac{1}{\|g_i\|} \sum_{d \in g_i} x(l, d)$$

Após a conclusão de todas estas etapas, estamos em condições de realizar a classificação de novos documentos, isto é, documentos não pertencentes ao conjunto de treino. Naturalmente, como em qualquer classificador tradicional, queremos atribuir à classe k a um novo documento d cuja distância absoluta seja a menor, ou seja, queremos dizer que d pertence à classe mais próxima que conhecemos segundo as *features* em estudo. Assim, para cada ponto \vec{p} a ser classificado, existirão $\|K\|$ distâncias quadráticas de *Mahalanobis* associadas, que representam a distância do vetor \vec{p} a cada classe conhecida. Dizemos então que o documento \vec{p} pertence a classe k_b se e só se o seu centroide $\vec{\mu}_b = \arg \min_{\vec{\mu}_i} M^2(\vec{p}, \vec{\mu}_i, \Sigma_i^{-1})$. Desta forma, é então possível implementar um classificador baseado na distância quadrática de *Mahalanobis*, no entanto o classificador à semelhança dos classificadores tradicionais, continua a não possuir a capacidade de rejeição.

3.4 A rejeição de documentos

3.4.1 Posicionamento do problema

Em geral, os classificadores conhecidos (*Support Vector Machines*, *Naive Bayes*, *K-nn*, entre outros) atribuem ao elemento a classificar, uma das classes que aprenderam na fase de treino; por norma aquela com características mais semelhantes às do elemento a classificar. Acontece que, por vezes o elemento a classificar é muito dissimilante tendo em conta qualquer das classes; por exemplo, se um classificador for treinado para reconhecer

documentos escritos numa de três classes, digamos Inglês, Francês e Português, caso o classificador tenha que classificar posteriormente um documento escrito em Espanhol, classificá-lo-á provavelmente como Português por razões de maior proximidade relativa; se tiver que classificar um documento escrito em Alemão terá tendência para classificá-lo como Inglês. Em qualquer destes dois casos há de facto uma maior semelhança, embora fraca em valor absoluto, com uma das classes, mas em rigor, qualquer dos documentos deveria ser rejeitado, pois não pertence a nenhuma língua aprendida na fase de treino. Os classificadores clássicos não possuem tal capacidade de rejeição. Este comportamento, em ambientes de situações reais não é o normalmente exigido. Na verdade, não parece aceitável que um classificador atribua a um documento um dos autores que conhece, só porque este configura a menor dissemelhança com a verdadeira autoria que na realidade, é estranha ao que foi aprendido pelo classificador. Para mitigar esta questão, é então necessário dotar o sistema classificador da capacidade de rejeitar também documentos sempre que estes se revelem ser suficientemente diferentes dos produzidos pelos autores que este conhece.

3.4.2 Resolução do problema

Com a finalidade de resolver a questão apresentada no sub-capítulo anterior, podemos usar a teoria que diz se a distribuição associada aos dados em cada *cluster* for gaussiana, isto é, se o *cluster* tiver uma distribuição multi-gaussiana é válido fazer um teste de χ^2 que relaciona a hipótese de um elemento pertencer a uma classe representada por um *cluster*, com a distância quadrática de *Mahalanobis* do elemento ao centróide desse *cluster*. A ideia central é estabelecer uma probabilidade suficientemente grande para aceitar que o elemento ainda deve pertencer ao *cluster*. Há pois uma distância de *Mahalanobis* limite associada a essa probabilidade. Para distâncias maiores que esse limite rejeita-se a hipótese do elemento pertencer à classe representada pelo *cluster*. Assim, conforme referido é possível utilizar um teste de χ^2 para rejeitar a autoria dum documento ou atribuí-la a um dos autores previamente conhecidos, seguindo a seguinte hipótese:

H_0 : seja p a amostra representada pelo vetor \vec{p} pertence a classe k_i representada por um *cluster* dos valores médios de cada atributo na classe e a matriz de covariâncias representados respetivamente por μ_i e $\vec{\Sigma}_i^{-1}$. Assim, aplicando um teste com um nível de confiança de α , podemos afirmar que H_0 não será rejeitada se e só se:

$$(\vec{p} - \vec{\mu}_i)^T \sum_i^{\vec{-1}} (\vec{p} - \vec{\mu}_i) \leq \chi_{df}^2(\alpha) \quad (3.3)$$

Onde df corresponde aos graus de liberdades, valor que é dado pelo número de eixos, isto é o número de dimensões representado pelo número de features em estudo. Então, ao usar-se uma tabela cumulativa de χ^2 e a distância quadrática de *Mahalanobis* do vetor \vec{p} ao centroide $\vec{\mu}_i$ podemos então decidir se o documento é próximo o suficiente para atribuir a autoria a um dos autores já conhecidos pelo sistema ou se é de tal forma dissemelhante que

nos permita rejeitar a autoria. Desta forma: Se $\exists k_i : M^2(\vec{p}, \vec{\mu}, \vec{\Sigma}^{-1}) = \min_{j \in k} M^2(\vec{p}, \vec{\mu}_j, \Sigma_j^{-1}) \wedge \text{In}(\vec{p}, \vec{\mu}_j, \Sigma_j^{-1}, \alpha)$ então $p \in k_i$, caso contrário p pertence a uma classe desconhecida. Nota: $\text{In}(\vec{p}, \vec{\mu}_j, \Sigma_j^{-1}, \alpha)$ é verdadeiro se a condição representada em (3.3) for verificada.

3.4.3 Transformação de Box-Cox

Conforme referido no sub-capítulo anterior, para podermos rejeitar documentos é necessário que os atributos em utilização sigam individualmente uma distribuição normal, isto é, gaussiana. Tipicamente, este caso, não tende a ocorrer para classificação em contextos difíceis. O processo de gaussianização é então descrito como sendo o processo de transformação da distribuição de uma dada variável, que à partida é desconhecido, numa distribuição normal. O processo, pode então ver visualizado na Figura 3.2.

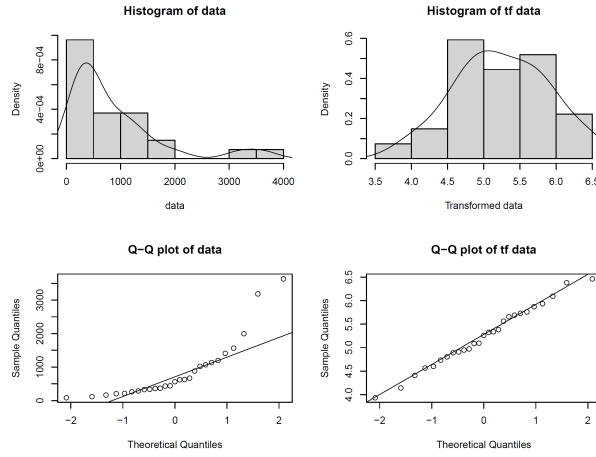


Figura 3.2: Transformação de uma variável com uma distribuição desconhecida para distribuição normal.

Surgem então naturalmente duas questões associadas a este processo:

- Como podemos realizar a transformação ?
- Como podemos verificar se o resultado produzido é suficientemente bom ?

Por forma a responder à primeira questão, foi utilizado o algoritmo de nome *Box-Cox* originalmente descrito em [2]. De forma geral, segundo o algoritmo, seja y um elemento arbitrário de uma determinada matriz C que queremos transformar em gaussiana. Consideramos a seguinte família de potências transformadores de y para $y^{(\lambda)}$:

$$y^{(\lambda)} = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{se } \lambda \neq 0 \\ \ln y & \text{se } \lambda = 0 \end{cases} \quad (3.4)$$

Transformações de potências estão definidas somente para variáveis positivas, isto é, $y \in \mathbb{R}^+$, no entanto isto não é restritivo, pois podemos adicionar uma constante v a elemento

y da matriz C caso exista algum elemento negativo na mesma. Naturalmente a escolha de v tem que verificar a condição $y + v > 0$. Assim, para os elementos $y_1, y_2, y_3, \dots, y_n$ de uma determinada coluna, a solução proposta por *Box-Cox* para a escolha da melhor potência de λ para cada coluna i.e para a distribuição relativa a cada *feature* é a que maximiza a seguinte expressão:

$$l(\lambda) = -\frac{m}{2} \ln \left[\frac{1}{m} \sum_{j=1}^{j=m} (y_j^\lambda - \overline{y^\lambda})^2 \right] + \ln(\lambda - 1) \sum_{j=1}^{j=m} \ln(y_j) \quad (3.5)$$

$$\text{Onde } \overline{y^\lambda} = \frac{1}{m} \sum_{j=1}^{j=m} y_j^\lambda \quad (3.6)$$

Sendo que m é o número de elementos de um determinada coluna e y está definido em Equação (3.4). Desta forma, após ser encontrado o melhor λ recorrendo a Equação (3.5) para cada coluna/*feature*, cada elemento da coluna é transformado segundo a família de transformações na Equação (3.4). Após todos os elementos terem sofrido esta alteração, os dados seguem uma distribuição aproximadamente gaussiana, condição necessária para podermos utilizar o módulo de rejeição. Estamos então em condições de escrever, sendo c_i uma das colunas da matriz C :

$$\forall c_i \in C, \exists \mu, \sigma^2 : c_i \sim \mathcal{N}(\mu, \sigma^2)$$

Posteriormente, o processo de rejeição nada mais é do que a utilização do classificador pela distância quadrática de *Mahalanobis*, utilizando os atributos já gaussianizados e verificando qual a distância ao centroide representativo do *cluster* mais próximo, i.e, com maior semelhança pelas *features* em estudo é próxima o suficiente para a atribuição ou de tal forma dissemelhante que imponha a rejeição.

4.1 Introdução

Pretende-se no presente capítulo apresentar os resultados obtidos de acordo com as problemáticas propostas e que podem ser consultadas pormenorizadamente no Capítulo 1. Desta forma, primeiramente foi necessário angariar documentos de vários autores por forma a ser possível construir um *dataset* com as características apropriadas a cada problema. A construção do referido dataset pode ser encontrado na Tabela 4.1. A disponibilidade de dados ou datasets com alta qualidade permite a obtenção de melhores resultados numa solução de *machine learning*.

Felizmente, foi possível encontrar um *dataset* com textos de vários heterónimos de Fernando Pessoa, que por terem sido, em rigor, produzidos pelo mesmo autor, revelam uma dificuldade acrescida na distinção dos seus autores, sendo por isso um contexto interessante. Nos demais elementos, tentou-se angariar elementos cujo período de lançamento das suas obras em alguns casos fosse próximo o suficiente por forma a que os padrões de escrita não fossem próximos por forma a constituir um desafio. Por exemplo: documentos de Lobo Antunes e José Saramago, sendo do mesmo período, a deteção de padrões de escrita requer capacidade discriminante.

4.1.1 Pré-processamento do texto

Muitos dos autores estudados e referenciados no estado da arte, que é possível consultar no Capítulo 2, realizam um pré-processamento dos documentos por forma a preparar/moldar os textos por diversas razões, tais como preparar o texto para melhor utilização por parte de um algoritmo, remoção de palavras com pouca significância, remoção de caracteres especiais entre outros. Para a realização da presente dissertação, nenhum desses processos foi utilizado, pois é do meu entender que embora exista informação que poderá deter um maior poder discriminante do que outra, toda a informação é potencialmente útil e poderá implicar na prática melhorias no processo de classificação se utilizada corretamente.

Tabela 4.1: *Dataset* utilizado.

Livro	Autor	Ano de Lançamento
A Brusca	Agustina Bessa Luís	1967
Dentes de Rato	Agustina Bessa Luís	1987
Dicionário Imperfeito	Agustina Bessa Luís	2008
Sibila	Agustina Bessa Luís	1954
A relíquia	Eça de Queirós	1887
O Mistério da Estrada de Sintra	Eça de Queirós	1870
Os maias	Eça de Queirós	1888
S. Cristóvão	Eça de Queirós	(1890-1900)
História do Descobrimento e Conquista da Índia	Fernão Lopes de Castanheda	1554
Peregrinação	Fernão Mendes Pinto	1614
Textos de quatros Heterónimos	Fernando Pessoa	(1914-1934)
Desamparo	Inês Pedrosa	2015
Fazes-me falta	Inês Pedrosa	2002
Fica comigo esta noite	Inês Pedrosa	2003
Nas tuas mãos	Inês Pedrosa	1997
Catarina de Bragança	Isabel Stilwell	2008
D.Amélia	Isabel Stilwell	2010
D.Teresa	Isabel Stilwell	2015
Inclita Geração	Isabel Stilwell	2016
As intermitências da morte	José Saramago	2005
Caim	José Saramago	2009
Ensaio sobre a cegueira	José Saramago	1995
O homem duplicado	José Saramago	2002
As Naus	Lobo Antunes	2000
Auto dos danados	Lobo Antunes	1992
Explicação aos pássaros	Lobo Antunes	1981
O arquipélago da insónia	Lobo Antunes	2008
Sermão de São Pedro	Padre António Vieira	1644
Sermão de Santo António	Padre António Vieira	1654
Sermão de Todos os Santos	Padre António Vieira	1643

4.2 Diferentes *datasets* e os resultados obtidos.

A resolução de cada problema referenciado no Capítulo 1 pretende demonstrar a capacidade da solução adotada, recorrendo para tal à sua avaliação em contexto prático. Para tal, da Tabela 4.1, foram formados os seguintes sub-conjuntos.

4.2.1 Autores do séc 19 e 20

O primeiro sub-conjunto pretende reunir autores cujas obras tenham sido escritas com um espaço temporal inferior a 100 anos, concretamente que sejam autores contemporâneos. Desta forma, é esperado, que padrões morfológicos e sintáticos se mantenham, no cômputo geral, inalterados. Assim, a informação estatística extraída por meio de todas as *features* descreve o padrão de escrita de cada autor, representando-o. O conjunto de autores selecionados e suas obras integra os seguintes:

- Agustina Bessa Luís - Todas as obras angariadas.
- Eça de Queirós - Todas as obras angariadas.

- Inês Pedrosa - Todas as obras angariadas.
- Isabel Stilwel - Todas as obras angariadas.
- José Saramago - Todas as obras angariadas.
- Lobo Antunes - Todas as obras angariadas.

Tabela 4.2: Resultados de avaliação das *features*.

Atributos	$D(.)$	$SK(.)$
2-char	1.30	-0.44
4-char	0.53	-0.17
5-char	4.49	1.27
9-char	0.93	-0.2
uni_grams	4.89	1.36
bi_grams	5.98	6.63
tri_grams	7.78	7.60
quad_grams	7.46	5.34
num medio palavras entre pontos exclamação	3.56	-1.09
maior que nove	0.92	0.05
menos que tres	1.31	-0.41
maior que seis	1.02	-0.39
tamanho palavra	0.78	0.09
num medio palavras entre virgulas	4.41	0.4
num medio palavras entre pontos	0.85	-2.04
num medio palavras	5.30	1.74
num medio letras	0.78	0.09
num medio pontos exclamacao	39.74	74.02
num palavras distintas	3.93	-3.23
not_ascii	3.78	1.37

Após a obtenção dos resultados obtidos por cada *feature* (consultar a Tabela 4.2 para mais detalhes), foram então selecionados os melhores atributos, isto é, os cujo valor dado pelo $D(.)$ é mais elevado e que não revelam ter um valor de $SK(.)$ muito negativo. Um forte valor negativo em $SK(.)$ significa que o poder discriminante para a maioria das classes não é muito elevado.

Após o passo anterior, foram testadas várias combinações de atributos que possuíssem as características selecionadas, com o intuito de se obter uma ideia de qual seria a performance obtida pelos classificadores. Finalmente, após selecionados os melhores atributos, foi realizado *Leave-One-Out* com vários classificadores, entre os quais *SVM*, *Gaussian Naive Bayes*, *AdaBoost*, *Bagging Classifier*, *Decision Tree* e *Random Forest*, implementados com recurso à biblioteca *sklearn*, com todo o *dataset*, tendo-se obtido resultados diferentes para alguns classificadores. Os melhores resultados podem ser visualizados na Tabela 4.3 e na Figura 4.1.

Por questões de espaço, serão apenas apresentados os melhores resultados, ficando os demais resultados obtidos nos Anexos A e B.

Tabela 4.3: Tabela de avaliação de *performance* pelo classificador *Random Forest* - (RF).

Métricas	Agustina	Eça	Inês	Isabel	Lobo	Saramago	<i>Accuracy</i>
<i>Precision</i>	1.0	1.0	1.0	1.0	1.0	1.0	1.0
<i>Recall</i>	1.0	1.0	1.0	1.0	1.0	1.0	1.0
<i>F1-score</i>	1.0	1.0	1.0	1.0	1.0	1.0	1.0
<i>Support</i>	50	50	50	50	50	50	1.0

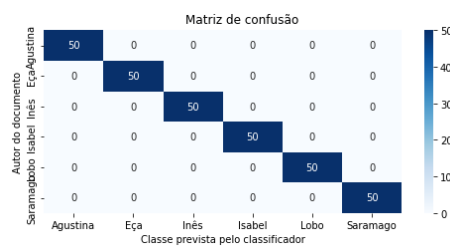


Figura 4.1: Matriz de confusão obtida através dos resultados do classificador *Random Forest* - (RF).

A Tabela 4.3 e a Figura 4.1 mostram resultados de uma classificação sem erros. Tratando-se de autores contemporâneos (à exceção de Eça de Queirós), podemos concluir que as *features* usadas revelam ter um poder discriminante suficientemente alto para poder distinguir os autores.

4.2.2 Classificação por género

Outro *dataset* criado, contém exatamente os mesmos autores do anterior, no entanto as classes são alteradas com o intuito de formar dois grupos, correspondentes ao género dos autores. Para o estudo em questão é utilizado somente o género masculino e feminino. Com a formulação deste conjunto, o objetivo passa por tentar perceber se com o uso de *features* puramente estatísticas é possível diferenciar autores por género, i.e, inferir a partir do *corpus* métricas com poder discriminante para que seja possível classificar documentos pelo género do seu autor. Naturalmente, como em qualquer outro problema de classificação em contextos difíceis, particularmente no caso em questão em que os atributos são puramente estatísticos, é necessário que existam de facto padrões de escrita diferenciados por género, o que é algo que creio que exista, mas que não passa de uma crença. Assim, as classes passam a ser definidas da seguinte forma:

$$\text{Classes} = \begin{cases} \text{Eça de Queirós, José Saramago ou Lobo Antunes} \rightarrow \text{Masculino} \\ \text{Agustina Bessa Luís, Inês Pedrosa ou Isabel Stilwel} \rightarrow \text{Feminino} \end{cases}$$

Desta forma, primeiramente foram criados cinquenta documentos para cada classe a partir do *dataset* inicial, tendo-se posteriormente averiguado qual o poder discriminante de cada atributo visível na Tabela 4.4 e, seguidamente, realizado a classificação. O melhor resultado obtido pode ser visualizado na Tabela 4.5.

Tabela 4.4: Resultados de avaliação das *features*.

Atributos	$D(.)$	$SK(.)$
número médio de palavras entre virgulas	0.69	-0.32
número médio palavras entre pontos	0.47	0.067
número palavras distintas	0.87	-0.69
número médio letras	1.85	-1.093
número médio pontos exclamação	6.30	14.25
número de caracteres não ascii	5.92	9.28
número de virgulas normalizado	0.90	-0.97

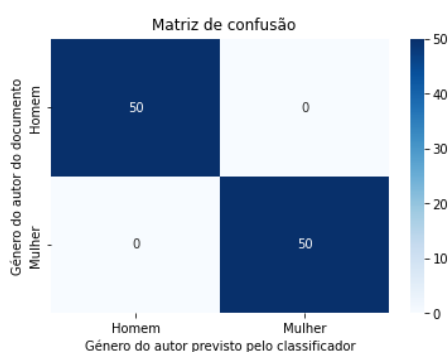


Figura 4.2: Matriz de confusão para o classificador *Random Forest* para a determinação do género do autor.

Tabela 4.5: Tabela de avaliação de resultados pelo classificador *Random Forest* - (RF).

Métricas	Homem	Mulher	<i>Accuracy</i>
<i>Precision</i>	1.0	1.0	1.0
<i>Recall</i>	1.0	1.0	1.0
<i>F1-score</i>	1.0	1.0	1.0
<i>Support</i>	50	50	1.0

Tendo em conta o estado da arte deste domínio (identificação por género) a Tabela 4.5 e a Figura 4.2, permite concluir que o conjunto de atributos selecionados para este fim são suficientemente discriminantes. No entanto, não se pode concluir que para qualquer que seja o *dataset* para treino e classificação, a identificação do género seja sempre conseguida com 100% de *accuracy*.

4.2.3 Classificação de autores de diferentes épocas

Ao longo dos anos e com o passar do tempo, todos os idiomas sofreram alterações. Estas mudanças derivam de mudanças inerentes ao estilo de escrita que se refletem tanto na forma sintática como morfológica. Assim sendo, é expectável que documentos produzidos em determinada época sejam textualmente diferentes de documentos, por exemplo, produzidos 200 anos depois. No entanto, esta tarefa enquadra-se também no processo de classificação em contextos difíceis pois, não são utilizados processos inerentes ao idioma, sendo esta uma abordagem independente do mesmo. Denota-se que no caso das diferenças serem somente ao nível morfológico, isto é, de grosso modo, ao nível das palavras/vocabulário utilizado, a atribuição de autoria é de extrema dificuldade não existindo o recurso a processos como a *lematização* ou *stemming*. Adicionalmente, como é natural, a evolução da escrita não foi um processo, por assim dizer, regular; existiram, naturalmente, fases de maior evolução em termos absolutos, como foi o caso do período do renascimento.

Por forma a verificar se as mudanças são suficientemente grandes de tal modo que um classificador consiga atribuir a autoria de forma correta, foi criado um *dataset* com os seguintes autores:

- Fernão Mendes Pinto - Todas as obras angariadas.
- Fernão Castanhede - Todas as obras angariadas.
- Padre António Vieira - Todas as obras angariadas.
- Lobo Antunes - Todas as obras angariadas.
- Inês Pedrosa - Todas as obras angariadas.
- Isabel Stilwell - Todas as obras angariadas.

O processo seguido foi o descrito anteriormente, onde existiu uma fase de selecção de atributos com capacidade discriminante e onde estes foram posteriormente passados aos classificadores. O melhor resultado obtido foi pelo classificador *Gaussian Naive Bayes*, cujos resultados conseguidos podem ser observados na Tabela 4.6 e a respetiva matriz de confusão na Figura 4.3.

Tabela 4.6: Tabela de avaliação de resultados obtidos pelo classificador *Gaussian Naive Bayes* para o *dataset* de autores de diferentes épocas.

Avaliação	FC	FMP	IP	IS	LA	P	Acuracy	Macro avg	Weighted avg
<i>Precision</i>	1	0.97	0.98	1.0	1.0	0.98	0.99	0.99	0.99
<i>Recall</i>	0.98	0.96	1.0	1.0	1.0	1.0	0.99	0.99	0.99
<i>F1-Score</i>	0.98	0.96	0.99	1.0	1.0	0.99	0.99	0.98	0.98
<i>Support</i>	50	50	50	50	50	50	0.99	300	300

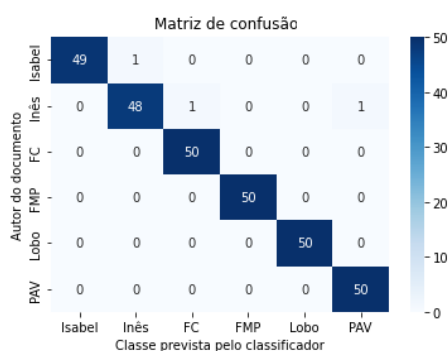


Figura 4.3: Matriz de confusão do classificador *Gaussian naive Bayes* para o *dataset* de autores de diferentes épocas.

A Tabela 4.6 e a Figura 4.3 mostram que apesar de a classificação não ser perfeita, esta apresenta valores altos de *precision*, *recall* e *accuracy*.

4.2.4 Dataset Fernando Pessoa

Conforme referido anteriormente, a angariação de bons *datasets* é fundamental para a realização de tarefas de classificação com suporte a algoritmos de *Machine Learning*. No entanto, estes são difíceis de angariar. Com a finalidade de testar a solução produzida, foi utilizado um conjunto de documentos de vários heterónimos de Fernando Pessoa. Este *dataset*, foi angariado graças a leitura cuidadosa do *paper* de João F. Teixeira e Marco Couto em [24], onde os autores realizam a atribuição de autoria de dois heterónimos. Este contexto de classificação, assume-se diferente dos anteriores no sentido em que, em bom rigor, todos os documentos, embora de heterónimos diferentes, foram produzidos pelo mesmo autor acrescentando assim uma maior dificuldade à tarefa. Adicionalmente, derivado ao facto de o autor ser o mesmo, os pressupostos iniciais de classificação não se aplicam, pois não é garantido que os heterónimos nos seus documentos, possuam características sintáticas e morfológicas distintas.

4.2.5 Descrição do dataset

O *dataset* angariado contém já alguma informação sobre cada documento. Esta informação pode ser visualizada com maior detalhe na Figura 4.4 e descreve-se da seguinte forma:

- **Autor** - Correspondente ao heterónimo a que é atribuída a autoria do documento.
- **Titulo** - O título do documento.
- **Tipo** - Género textual associado ao documento. No contexto do *dataset* é resumido a prosa ou poesia.
- **Data** - Data da produção do documento. Alguns documentos não possuem data, pois não se sabe com rigor a data da sua produção.

- **Bibliografia** - Bibliografia do documento.

	id	autor	título	tipo	texto	data	bibliografia
0	4	Ricardo Reis	Diana através dos ramos	poesia	Diana através dos ramos Espregueira a virde de Erydmion Erydmion que nunca v...	16-6-1914	Poemas de Ricardo Reis, Fer...
1	5	Fernando Pessoa	A REFORMA DO CALENDÁRIO E AS SUAS CON...	prosa	A REFORMA DO CALENDÁRIO E AS SUAS CONSEQUÊNCIAS COMERCIAIS	10-3-1933	Páginas de Pensamento Pol...
2	6	Fernando Pessoa	The Infinite is then the Possible	prosa	The Infinite is then the Possible How does this possible realise itself? How is it re...	1906?	Textos Filosóficos . Vol. II. F...
3	7	Ricardo Reis	Aqui, sem outro Apolo do que Apolo,	poesia	Aqui, sem outro Apolo do que Apolo, Sem um suspiro abandonemos Cristo!	11-8-1914	Poemas de Ricardo Reis, Fer...

Figura 4.4: Descrição do *dataset* Fernando Pessoa.

A construção do sub-conjunto a partir do *dataset* inicial, que utilizaremos como *corpus* no processo de classificação recorrendo a solução proposta, depende do número de documentos que podemos encontrar por cada classe. Naturalmente, uma classe com poucos documentos revela-se menos interessante em comparação com outra com um maior número de elementos. Como tal, foi realizada a assunção de somente utilizarmos heterónimos cuja cardinalidade dos documentos a estes atribuídos seja igual ou superior a 80 documentos. A razão da imposição desta condição prende-se essencialmente com a dificuldade associada ao *dataset* e crê-se que com um número inferior de documentos, os documentos obtidos não sejam suficientes por forma a captar a essência do seu autor. Desta forma, os heterónimos utilizados foram: RR - Ricardo Reis , ÁDC - Álvaro de Campos , AC - Alberto Caeiro e BS - Bernardo Soares, com a seguinte distribuição de documentos visível na Figura 4.5.

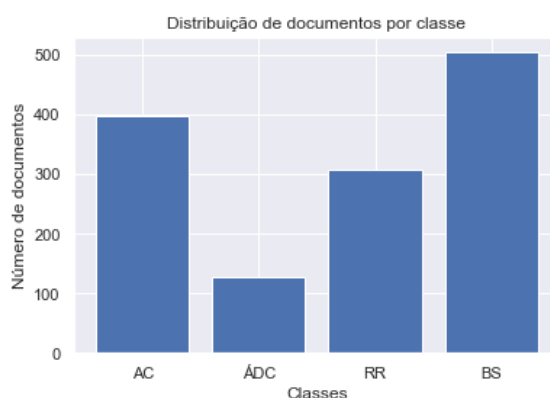


Figura 4.5: Distribuição de documentos por classe no *dataset* de Fernando Pessoa.

Os documentos dos autores/heterónimos utilizados, muitos, infelizmente, não possuem data, tendo a informação na *label* da data como "s.d.". Noutros, depois da data surge um "?", exemplo: "1931?", pelo que foi assumido que a informação correspondente não é passível de ser utilizada. Assim, os documentos produzidos por ano e década, bem como a distribuição do género textual por classe e em termos absolutos podem ser visualizados na Figura 4.6.

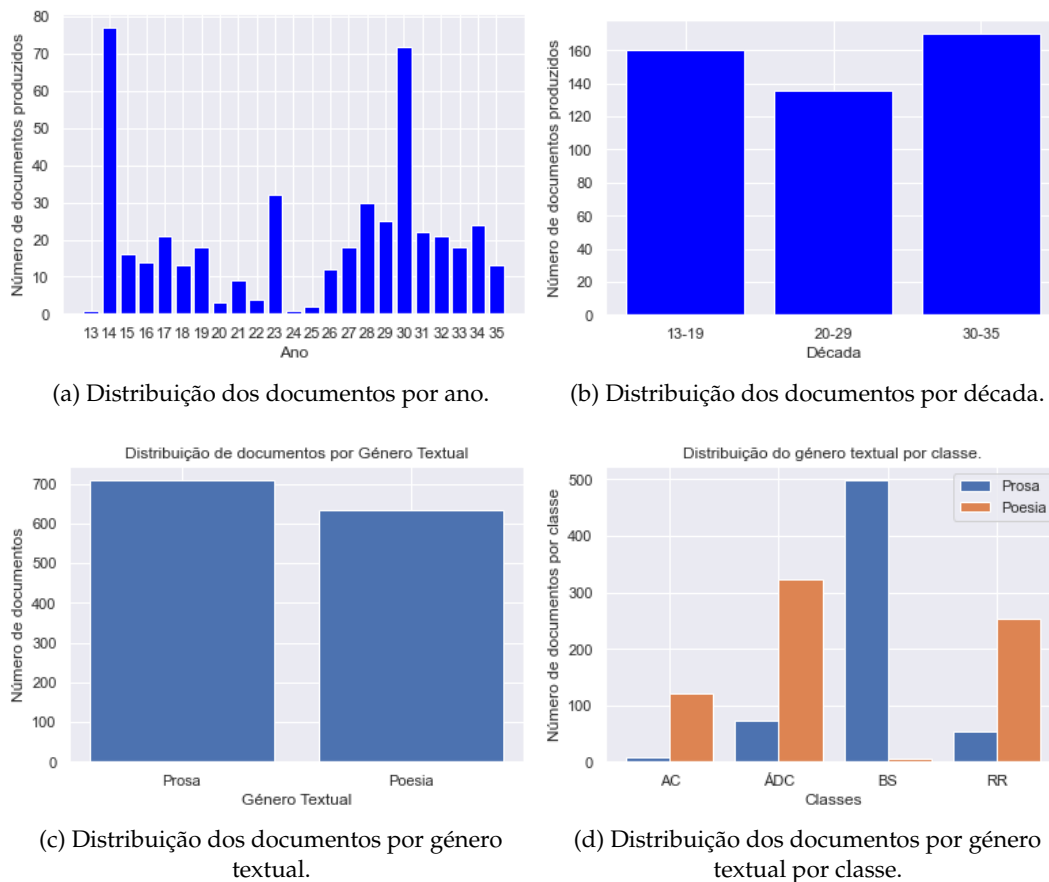


Figura 4.6: Análise do *dataset* de Fernando Pessoa.

Conforme foi possível verificar na composição do *dataset* constituído por documentos de vários heterónimos, nem todas as classes têm a mesma cardinalidade. Esta situação é o que tende a acontecer em conjuntos de dados do mundo real, onde nem sempre a distribuição de dados pelas diversas classes é então homogénea. Foram realizadas duas experiências, uma onde o número de documentos por cada classe é igual ao mínimo dos documentos entre as várias classes e outra em que foram utilizados todos os conjuntos de dados. Os atributos em estudo, bem como o valor associado a cada um através das métricas $D(\cdot)$ e $SK(\cdot)$, angariados através da utilização de cerca de 20%, podem ser vistos na Tabela 4.7.

Tabela 4.7: Resultados de avaliação das *features* para o *dataset* de Fernando Pessoa.

Atributos	$D(.)$	$SK(.)$
tri-grams	12.41	0.23
bi-grams	7.27	0.17
quad-grams	5.31	0.02
variancia-frase	1.23	0.00
uni-grams	0.66	0.00
5-char	0.50	0.00
tamanho-medio-palavra	0.16	0.00
tamanho-frase	0.09	-0.00
Exclamação	0.07	-0.00
Virgulas	0.04	-0.26
variancia-silabas	0.04	-0.33
upper	0.03	-0.43
num-versos	0.03	-3.10
palavras entre virgulas	0.02	-3.94
2-char	0.02	-6.30
num medio letras por doc	0.02	-20.02
6-char	0.02	-21.55
1-char	0.02	-23.24
palavras entre pontos de interrogação	0.02	-50.63
Interrogação	0.01	-71.23
Pontos	0.01	-415.12
ponto e virgula	0.00	-3073.31

4.2.6 Redução de dimensionalidade

Quando o número de atributos angariados a partir dos documentos ou já contidos no próprio *dataset* é muito elevado, a testagem de várias combinações de *features* torna-se complicada, sendo em alguns casos no contexto prático, impossível. Por forma a lidar com esta questão existem os chamados algoritmos de redução de dimensionalidade, que, de grosso modo, reduzem o número de atributos preservando, tanto quanto possível, a informação presente nas mesmas.

Desta forma, a redução de dimensionalidade tem uma *performance* tanto melhor quanto maior o número de atributos finais após a redução preservando a mesma informação. Neste sentido e devido a este *dataset* ter sido aquele onde mais atributos foram angariados, foi executado o algoritmo *PCA - Principal Component Analysis*, tendo-se obtido o resultado visível na Figura 4.7.

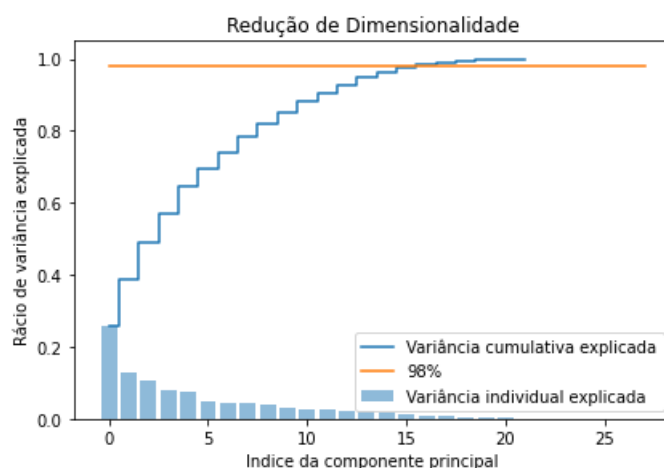


Figura 4.7: Execução do algoritmo *PCA - Principal Component Analysis* para o *dataset* de Fernando Pessoa.

Como é possível verificar, para se obter uma explicação de $\approx 98\%$ da variância, o número de novos atributos gerados pelo *PCA* seria de aproximadamente 16, o que corresponde a uma redução de $\approx 27\%$, o que não é muito significativo, pois não se traduz numa redução de tal forma importante que reduza os custos computacionais associados. Mais se acrescenta, que existe ainda uma perda da variância explicada por estas 16 *features* vs as 22 originais, culminando numa perda de *performance* por parte do classificador, que não se pretende.

De uma forma geral, foi então preferido que o algoritmo demore um pouco mais de tempo a ser executado, isto é, a ser treinado mas que produza resultados mais precisos e credíveis, que é o que se pretende em tarefas de Atribuição de Autoria.

4.2.6.1 Normalização de documentos por classe

Derivado à não existência de homogeneidade na distribuição da cardinalidade dos documentos existentes em cada classe, foi então realizada esta experiência em que todas as classes têm o mesmo número de documentos, número esse correspondente ao mínimo de documentos existentes por classe não normalizada. Foram escolhidos 20% para teste do poder discriminante dos atributos, tendo-se obtido os resultados visíveis na Figura 4.7. Posteriormente, foi realizado *leave-one-out* tendo o melhor classificador, neste caso, o *Bagging Classifier* obtido 95 % de *accuracy*, sendo possível ver a respetiva matriz de confusão na Figura 4.8.

4.2. DIFERENTES DATASETS E OS RESULTADOS OBTIDOS.

Tabela 4.8: Resultados obtidos pelo classificador *BG - Bagging Classifier* para o *dataset* dos heterónimos normalizado.

Métricas	AC	BS	RR	ÁDC	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.93	0.99	0.91	0.96	0.95	0.95	0.95
<i>Recall</i>	0.96	0.99	0.90	0.94	0.95	0.95	0.95
<i>F1-Score</i>	0.94	0.99	0.90	0.95	0.95	0.95	0.95
<i>Support</i>	127	127	127	127	0.95	508	508

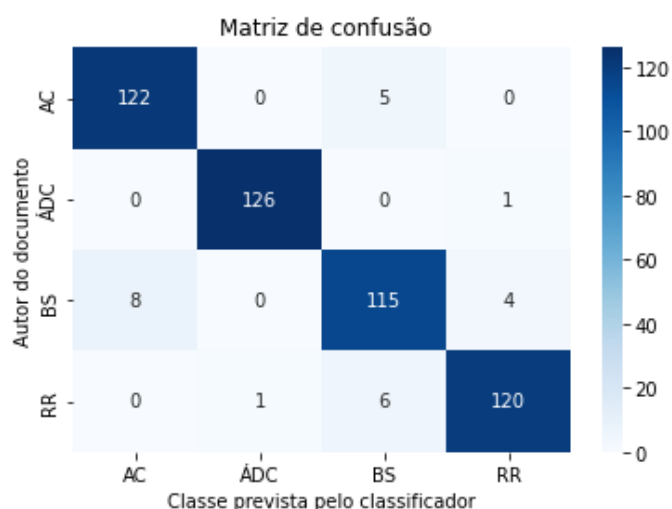


Figura 4.8: Matriz de confusão obtida pelo classificador *BG* para o *dataset* dos heterónimos normalizado.

4.2.6.2 Classificação utilizando os documentos todos

Nesta experiência de classificação, foram utilizados todos os documentos existentes de cada heterónimo. Por forma a realizar a Atribuição de Autoria, foi realizado *leave-one-out* e obtido os resultados disponíveis através de uma tabela de avaliação e matriz de confusão na Tabela 4.9 e na Figura 4.9. A *accuracy* da solução recai para $\approx 90\%$.

Tabela 4.9: Resultados obtidos pelo classificador *Bagging Classifier* para o *dataset* contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por classe/autor.

Métricas	AC	BS	RR	ÁDC	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.87	0.91	0.90	0.90	0.90	0.89	0.90
<i>Recall</i>	0.82	0.93	0.94	0.85	0.90	0.89	0.90
<i>F1-Score</i>	0.85	0.92	0.92	0.88	0.90	0.89	0.90
<i>Support</i>	127	504	307	397	0.90	1335	1335

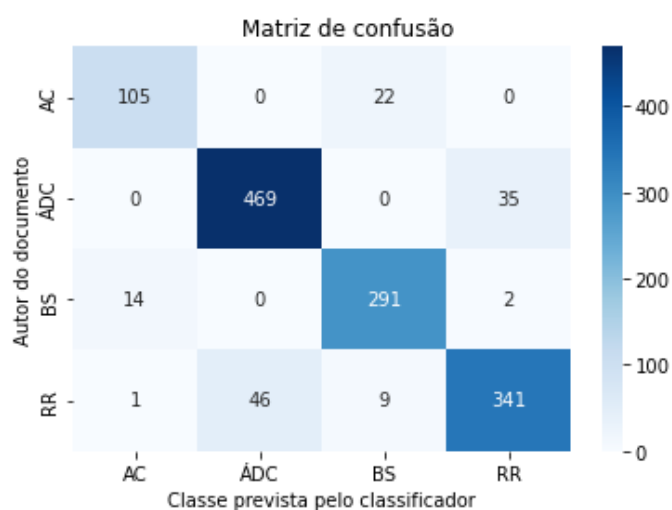


Figura 4.9: Matriz de confusão obtida pelo classificador *Bagging Classifier* para o *dataset* contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por classe/autor.

4.2.6.3 Comparação com o estado da arte

No estado da arte, foi encontrado um estudo recente, produzido por investigadores da Faculdade do Minho, que incide sobre a mesma área da presente dissertação e utiliza também o *dataset* de Fernando Pessoa. De forma geral, os resultados obtidos com a abordagem apresentada foram melhores, não só em termos de classificação tendo aumentado cerca de 1% a *accuracy* com dois heterónimos como é visível na Tabela 4.10 e na Figura 4.10. Adicionalmente, a solução proposta em [24], utiliza já com a redução de dimensional 4398 atributos, constituindo por isso, uma solução demorada e pouco elegante, que necessita de grandes recursos computacionais para ser executada. É de notar que os autores em [24] não utilizaram um *dataset* com quatro heterónimos, o que levaria a utilização de uma quantidade maior de atributos. É também neste sentido, preferida a solução desenvolvida com somente 22 atributos, pois é menos dispendiosa e com um resultado ligeiramente melhor.

Tabela 4.10: Tabela de resultados obtidos pelo classificador *AdaBoost* para 2 heterónimos.

Métrica	Ricardo Reis	Álvaro de Campos	<i>accuracy</i>	<i>macro avg</i>	<i>weighted avg</i>
<i>Precision</i>	0.97	0.98	0.98	0.98	0.98
<i>Recall</i>	0.98	0.98	0.98	0.98	0.98
<i>F1-Score</i>	0.98	0.98	0.98	0.98	0.98
<i>Support</i>	307	397	0.98	704	704



Figura 4.10: Matriz de confusão obtida pelo classificador *AdaBoost* para 2 heterônimos.

4.2.6.4 Abordagem com redes Neurais

Nos últimos anos, diferentes soluções para a área de *Atribuição de Autoria* - (AA) têm sido apresentadas. Em particular, a maioria das novas soluções reside na área da aprendizagem profunda, isto é, de grosso modo, soluções que usam como recurso redes neurais. Com o intuito de cobrir também este tipo de solução, foi então usado uma rede neuronal no *dataset* contendo os heterônimos de Fernando Pessoa. De forma semelhante à abordagem com os classificadores clássicos, foi também realizado duas experiências, uma contendo o *dataset* como um todo e outra em que existia uma normalização de documentos por cada classe. Devido ao funcionamento associado a esta solução recorrendo a redes neurais, é esperado que a *performance* seja, em termos absolutos, ligeiramente inferior a dos classificadores tradicionais, devido essencialmente a quantidade de documentos existentes, sendo que também por este mesmo motivo, é esperado que a *performance* com o conjunto todo seja melhor do que com a normalização de documentos por autor. O processo apresentado em ambas as experiências foi semelhante, tendo contemplado as seguintes fases:

Seleção dos dados:

Os dados começaram por ser divididos essencialmente em dois conjuntos, treino e teste. O conjunto de teste contém 40 amostras de cada classe sendo que todas as restantes amostras pertencem ao conjunto de treino.

One-Hot-Encoding:

No caso concreto dos nossos dados, de momento, as classes são representadas pelo nome do heterônimo. No entanto, alguns algoritmos necessitam que os dados sejam numéricos. Como tal, para que se possa utilizar as redes neurais no *dataset*, têm que existir uma conversão de dados categóricos para numéricos. Uma de várias soluções que realizam esta conversão dá pelo nome de *One-Hot-Encoding*. Assim, desta forma, as classes passam a ser representadas por um vetor, da seguinte forma:

- Alberto Caeiro = [1,0,0,0]
- Bernardo Soares = [0,1,0,0]
- Ricardo Reis = [0,0,1,0]
- Álvaro de Campos = [0,0,0,1]

Normalização:

Tipicamente, em qualquer rede neural os *inputs* são normalizados. Isto acontece, pois é mais fácil para a rede aprender se evitar valores extremos e tiver os dados centrados na origem. Com uma rede profunda, cada camada recebe as ativações da camada anterior como *inputs*. Assim, estas beneficiam também se os valores estiverem normalizados. Adicionalmente, durante o treino as ativações vão variando à medida que os pesos são atualizados, isto faz com que as entradas de cada camada mudem constantemente a sua média e variância, forçando a que cada camada se ajuste a estas mudanças causadas nas camadas anteriores. No caso dos valores não estarem normalizados, estas mudanças afetariam todo o funcionamento da rede neuronal. Desta forma, todos os dados foram normalizados.

Redução de *overfitting*:

Por forma a poder lidar com esta problemática no contexto das redes neuronais, existem diversas soluções. Para ambos os conjuntos, foram testadas soluções recorrendo a *dropout*, *l1* e *l2*, tendo-se obtido, para ambos os casos, melhores resultados com *l1*.

Escolha do modelo:

As redes neuronais não tem todas a mesma constituição, como tal é necessário escolher os melhores parâmetros para cada problema. Neste caso, tentou-se vários modelos com diversas camadas e com um número diferente de neurónios, bem como diferentes otimizadores. Entre todos, o que melhor se adaptou foi o modelo representado na Tabela 4.11.

Tabela 4.11: Modelo neuronal utilizado.

Modelo		
Camada (tipo)	Forma do output	Parâmetros #
flatten (Flatten)	(None, 22)	0
dense (Dense)	(None, 128)	3712
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 4)	260
=====		
Total parâmetros: 12,228		
Parâmetros treináveis: 12,228		
Parâmetros não treináveis: 0		

De entre todos os otimizadores testados, tais como o *stochastic gradient descent*, *texttmsprop* e *adam*, foi o último que revelou produzir melhores resultados. Como função

de *loss*, de acordo com as especificidades do problema foi utilizado a *sparse categorical crossentropy*. Como função de ativação foram usadas nas camadas densas *RELU* e *Softmax* na camada final.

Normalização de documentos por classe

Conforme foi realizado com os classificadores clássicos, também existiu por meio desta abordagem uma normalização da cardinalidade de documentos por classe de forma aleatória. A *accuracy* do modelo com este formato de documentos por classe foi de $\approx 70\%$, onde mais detalhes sobre os resultados podem ser consultados na Tabela 4.12 e na Figura 4.11.

Tabela 4.12: Resultados obtidos pelo modelo neuronal para o *dataset* contendo 4 heterónimos com o número de documentos por classe normalizado.

Avaliação	AC	BS	RR	ADC	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.88	0.70	0.80	0.52	0.7	0.72	0.72
<i>Recall</i>	0.94	0.48	0.56	0.82	0.7	0.70	0.70
<i>F1-Score</i>	0.91	0.57	0.65	0.64	0.7	0.69	0.69
<i>Support</i>	50	50	50	50	0.7	200	200

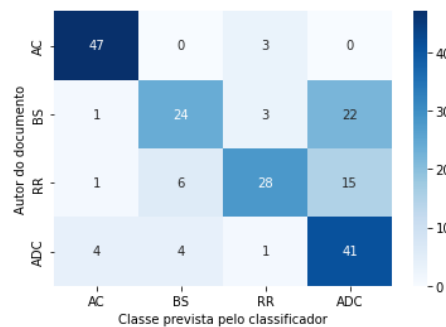


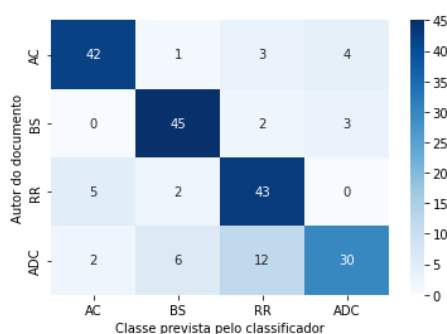
Figura 4.11: Matriz de confusão obtida pelo modelo neuronal para o *dataset* contendo 4 heterónimos com o número de documentos por classe normalizado.

Classificação utilizando os documentos todos

Nesta experiência, foram utilizados todos os documentos disponíveis de cada heterónimo independentemente de existir ou não uma homogeneidade da cardinalidade de cada classe. Assim, a *accuracy* do modelo foi de $\approx 80\%$, onde mais detalhes podem ser consultados na Tabela 4.13 e na Figura 4.12.

Tabela 4.13: Resultados obtidos pelo modelo neuronal para o *dataset* contendo 4 heterónimos com todos os documentos existentes por classe.

Avaliação	AC	BS	RR	ADC	accuracy	macro avg	weighted avg
<i>Precision</i>	0.85	0.83	0.71	0.81	0.8	0.80	0.80
<i>Recall</i>	0.84	0.90	0.86	0.60	0.8	0.80	0.80
<i>F1-Score</i>	0.84	0.86	0.78	0.68	0.8	0.79	0.79
<i>Support</i>	50	50	50	50	0.8	200	200

Figura 4.12: Matriz de confusão obtida pelo modelo neuronal para o *dataset* contendo 4 heterónimos com todos os documentos existentes por classe.

4.2.7 Módulo de rejeição

Para a implementação do módulo de rejeição, o objetivo é a rejeição de documentos estranhos, isto é, documentos que sejam suficientemente dissemelhantes dos aprendidos na fase de treino pelo classificador. Os classificadores tradicionais, infelizmente, não possuem a capacidade de rejeição, o que é uma grande desvantagem para com a tarefa de Atribuição de Autoria, no sentido em que em qualquer situação, estes irão atribuir a autoria sempre a uma das classes (autor) que conhecem independentemente do quão dissemelhante dos demais o documento em teste possa ser. Naturalmente, pela natureza da tarefa, a implementação deste módulo constitui um dos grandes desafios da presente dissertação.

Foi escolhido o *dataset* de Fernando Pessoa como primeira hipótese, para a verificação da *performance* prática obtida pela solução proposta. Como é natural, a capacidade de rejeição associada a um processo de classificação pela distância quadrática de *mahalanobis*, deve em primeira instância, conseguir *gaussianizar* todos os atributos em estudo e, posteriormente, classificar o documento com taxas associadas de *accuracy* tão altas quanto possível. Naturalmente, que com performances más na fase de classificação, todos o processo posterior é posto em causa.

Todo os documentos disponíveis no *dataset* foram utilizados, por forma a que

os centroides calculados não estejam sujeitos a variâncias locais associadas a sub-conjuntos formados. O processo seguido, passou por dividir o conjunto em dois sub-conjuntos, um de treino contendo, 80% dos documentos, e outro de teste, contendo os restantes documentos divididos por autor. Seguidamente, foi realizada a gaussianização dos dados recorrendo ao algoritmo de *Box-Cox*. Os resultados obtidos podem ser vistos na Figura 4.13.

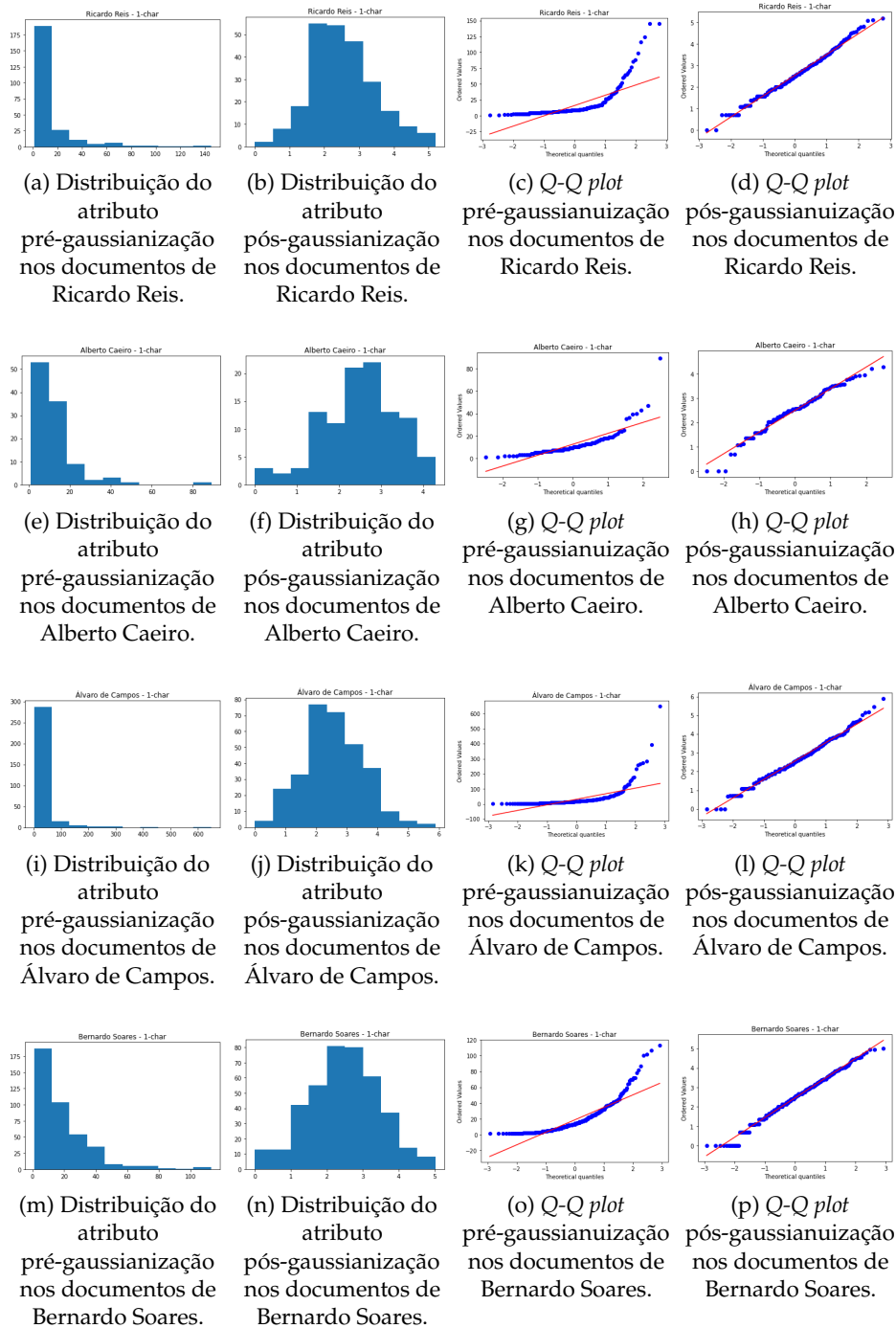


Figura 4.13: Exemplos de transformações *Box-Cox*.

Com a visualização dos respetivos histogramas, é notório que a gaussianização molda os dados por forma a que estes apresentem um aspeto mais normal. Uma das questões associadas a este processo prende-se com a capacidade da transformação em causa, pois não é possível através de histogramas ou *Q-Q plots* afirmar com certeza, se a distribuição é ou não normal. Esta verificação assume uma extrema importância no contexto da presente tese, uma vez que o princípio teórico só é válido na condição de cada *feature* se encontrar com uma distribuição normal por autor.

Para tal, foi utilizado o teste de *Shapiro-Wilk*, que nos permite, de grosso modo, rejeitar que a amostra possui uma distribuição normal se o valor de p for inferior a 0.05. Com a utilização deste recurso, podemos verificar que apenas dois dos quatro atributos estão devidamente gaussianizados, conforme é possível confirmar na Tabela 4.14. Infelizmente, em muitos casos, o algoritmo somente aproxima os dados de uma distribuição normal, não os tornando gaussianos.

Tabela 4.14: Teste de Shapiro-Wilk.

Autor	Statistic	<i>p-value</i>	Distribuição
RR	0.99	0.17	\approx <i>Gaussiana</i>
AC	0.97	0.03	Não <i>Gaussiana</i>
ADC	0.99	0.12	\approx <i>Gaussiana</i>
BS	0.99	0.01	Não <i>Gaussiana</i>

Posteriormente, testes de classificação foram realizados com diversos atributos, com o intuito de se escolher as *features* que produzissem melhores resultados em termos de *performance*, para que depois se realizasse *leave-one-out*, obtendo assim resultados mais fidedignos. Infelizmente, mesmo com a melhor combinação de atributos, a melhor *accuracy* obtida era de $\approx 54\%$, o que não é suficiente para que se possa avançar para uma fase de rejeição, por motivos supracitados. A diminuição da *performance* com os dados gaussianizados, já era algo esperado, uma vez que a utilização deste processo diminui a variância dos dados, fazendo com que os atributos percam parte do seu poder discriminante, o que leva a piores resultados na classificação. Adicionalmente e por este ser um contexto de classificação difícil, os centróides calculados e utilizados na *distância quadrática de mahalanobis* não são muito díspares entre si, contribuindo assim também para o acréscimo de dificuldade na classificação.

Com o intuito de mitigar esta questão, foi construído um conjunto diferente, por forma a ser possível obter uma boa *performance* na fase de classificação e entrar então na componente da rejeição. O *dataset* construído, diz então respeito a 50 documentos do autor Padre António Vieira, 307 documentos de Ricardo Reis

e 50 documentos de Fernão Castanhede. Após a obtenção do melhor conjunto de atributos, foi realizado *leave-one-out*, tendo-se obtido os seguintes resultados visíveis na Tabela 4.15.

Tabela 4.15: Classificação obtida com recurso a Distância de *Mahalanobis*.

Métrica	FC	PAV	RR	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	1.0	0.39	1.0	0.80	0.79	0.92
<i>Recall</i>	1.0	1.0	0.74	0.80	0.91	0.80
<i>F1-Score</i>	1.0	0.56	0.85	0.80	0.80	0.83
<i>Support</i>	50	50	307	0.80	407	407

Assim, com uma *accuracy* de $\approx 80\%$, existe já uma performance razoável e pode-se passar a fase de rejeição de documentos. Assim, foi acrescentado ao *dataset* documentos da Isabel Stilwell, cujos documentos são estranhos ao conjunto de treino. De acordo com os resultados teóricos, se χ^2 com três graus de liberdade para um α graus de sucesso, for maior que a menor das distâncias a cada centroide, então a atribuição de autoria ao documento é rejeitada, caso contrário é atribuída a classe que apresentar uma distância de *Mahalanobis* menor. Os resultados da rejeição, podem ser vistos na Tabela 4.16.

Tabela 4.16: Performance da rejeição com $\alpha = 0.001$.

α	Distância	<i>Accuracy</i>	<i>Support</i>
0.001	16.27	1.0	50

Desta forma, é possível verificar que para autores cuja escrita é suficientemente diferente, o módulo de rejeição funciona. No entanto, como é possível observar para os resultados obtidos para o *dataset* dos heterónimos, a solução infelizmente não possui robustez suficiente para permitir, no contexto geral, realizar a rejeição com qualidade.

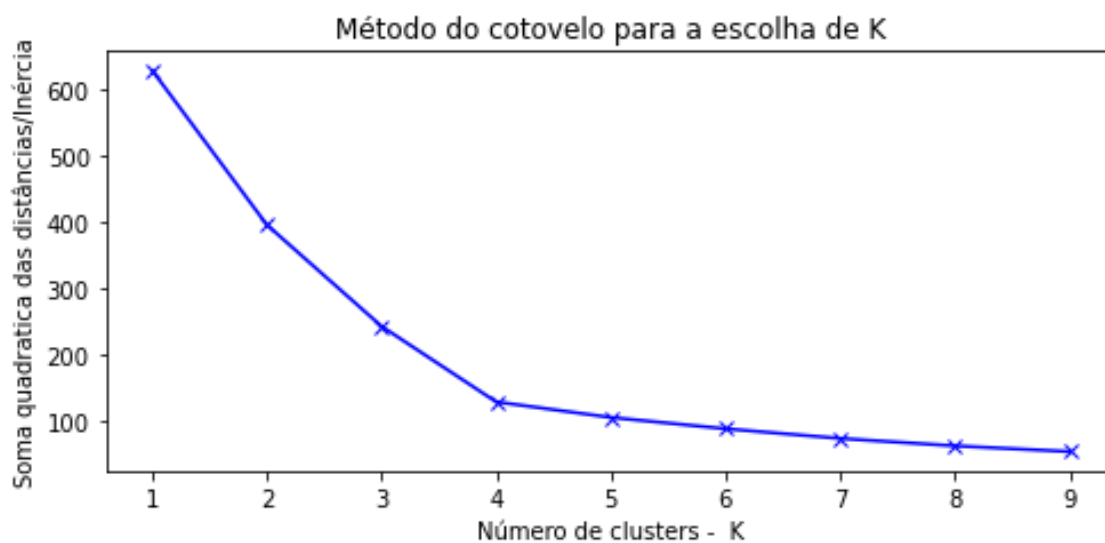
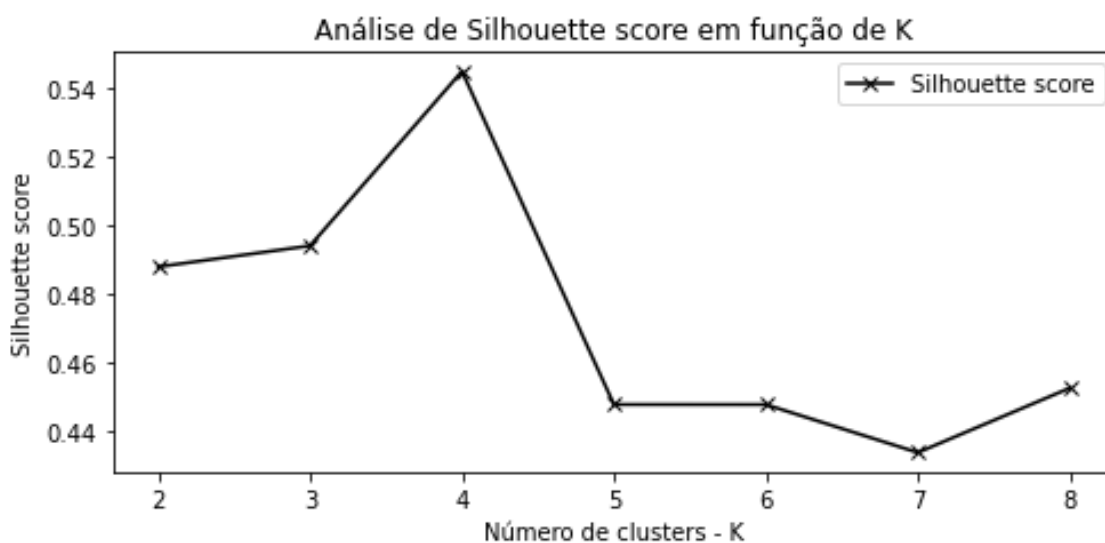
A questão da necessidade dos dados terem que ter uma distribuição normal, faz com que as diferenças existentes entre os autores sejam suavizadas, o que faz com que os atributos percam capacidade discriminante, culminando numa maior dificuldade de distinguir a autoria dos autores. Adicionalmente, questões como a proximidade dos centroides, para contextos de classificação difícil também dificultam a tarefa, tendo-se obtido somente 56% de *accuracy* vs 80% quando estes são mais distantes.

4.2.8 A problemática da evolução da escrita em *features* estatísticos.

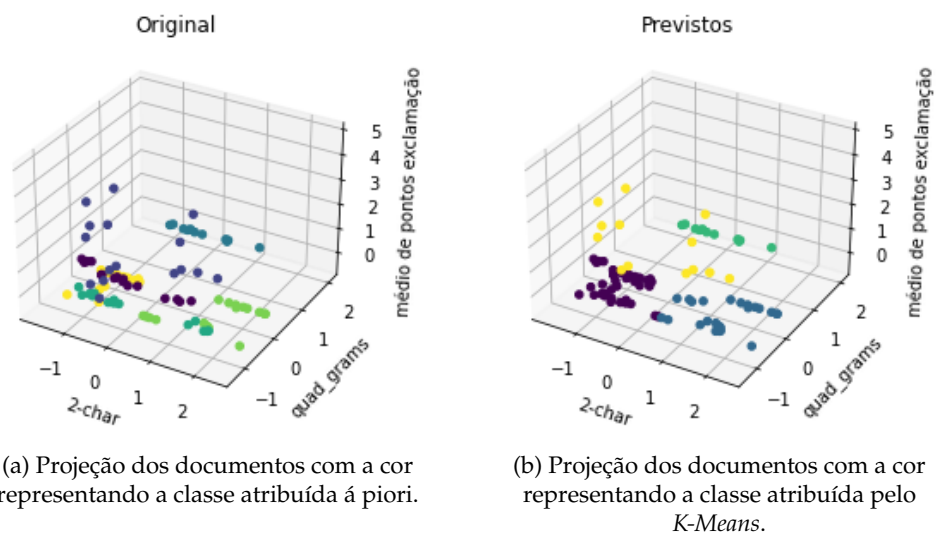
Para a implementação da solução de classificação de documentos de texto cru em contextos difíceis, com a implementação independente do idioma em que o documento tenha sido produzido, foi tomado como pressuposto a existência de *features* estatísticas com capacidade discriminante que permitam descrever um autor de acordo com os seus padrões de escrita. No entanto, conforme referido em [10] os padrões de escrita, entre o mesmo autor, podem mudar por diversos fatores, entre eles o tema do documento. Naturalmente, um autor pode também evoluir ou modificar a sua escrita ao longo do tempo, fazendo com que as características que o descrevem mudem também, introduzindo uma maior variância nestas. Desta forma, os *clusters* tornam-se mais dispersos, em vez de compactos, dificultando assim o processo de classificação. O método utilizado na presente dissertação insere-se no âmbito da classificação supervisionada, uma vez em que é utilizada a informação correspondente ao autor de cada documento. No entanto, pode-se fazer a experiência de não utilizar esta informação e verificar qual o número de *clusters* que existem em determinado *dataset*.

Se considerarmos o *dataset* representado na Tabela 4.2, podemos verificar que o classificador com melhores resultados utiliza três atributos, assim, cada documento passa então a ser descrito por um vetor $\vec{d} = [x_i, y_i, z_i]$ cujas entradas correspondem as *features* em estudo. Desta forma, este vetor pode ser projetado de acordo com as suas coordenadas num plano ortonormado e com a capacidade de visualização por parte dos humanos, uma vez que existem só três eixos.

A abordagem seguida passa então por utilizar o *K-Means* sobre o *dataset* para que possamos inferir qual o melhor número de *clusters*. Para tal, foi dividido o conjunto total dos documentos em treino e teste. A ideia consiste em testar vários números de *clusters* sobre os documentos de treino, em valor não muito distantes do que é o número de autores diferentes dos documentos e posteriormente verificar os resultados obtidos recorrendo para tal a métricas como *Elbow Method* e *Silhouette Score*. No final, é feita a comparação entre a projeção dos documentos originais de acordo com as classes sabidas a priori com a classificação realizada pelo algoritmo. Assim, os resultados obtidos face ao número de *clusters* k foram os que se podem observar nas Figuras 4.14 e 4.15.

Figura 4.14: Método do cotovelo para a escolha do número de *clusters*.Figura 4.15: Análise de *Silhouette* em função do número de *clusters*.

Como é possível verificar, de entre os vários valores de *clusters* possíveis, ambas as métricas fornecem a indicação de que o número ótimo de autores para o conjunto é de quatro autores. Ora, como é sabido, o conjunto de fato têm seis autores diferentes e não quatro. Esta informação indica-nos que existe um conjunto de documentos que são suficientemente próximos, segundo a métrica da distância euclidiana utilizada pelo algoritmo, por forma a que se atribua a sua autoria a somente quatro dos seis autores, como se pode ver na Figura 4.16.



(a) Projeção dos documentos com a cor representando a classe atribuída á priori.

(b) Projeção dos documentos com a cor representando a classe atribuída pelo *K-Means*.

Figura 4.16: Projeção dos documentos representados pelos atributos.

CONCLUSÕES

5.1 Conclusões sobre o trabalho produzido

A presente dissertação, teve como objetivo a classificação de documentos de texto cru, particularmente em contextos difíceis, apresentando para tal uma proposta, em particular, de Atribuição de Autoria a cada documento, independente do idioma em que este se encontra escrito. Para tal, a abordagem carece da obtenção de características aos documentos que sejam independentes do idioma em que os documentos se encontram escritos, o que é de certa forma, uma limitação à informação angariada pelas *features*.

No cômputo geral, os resultados obtidos são bastante satisfatórios, tendo sido obtido taxas de *accuracy* superiores a 95% para a esmagadora maioria dos *datasets*, revelando assim que características compostas por *features* de natureza estatística são capazes de realizar o processo de Atribuição de Autoria. Contudo, estas podem não captar a mudança/evolução eventual da escrita por parte dos autores, mudança esta que introduz variância nos valores das *features* que captam o seu estilo, fazendo com que a tarefa de classificação seja dificultada.

De forma não esperada, em primeira instância, em função da abordagem, os classificadores *Decision Tree* e *Random Forest* obtiveram performances relativamente alta para todos os *datasets* abordados. A explicação reside na forma como estes funcionam, tomando em cada nó decisões, em função dos atributos que captam a essência de escrita dos autores, que conseguem descrever os mesmos de forma bastante precisa.

Outra das principais questões endereçadas pela presente dissertação era a produção de um módulo de rejeição que não atribuísse a autoria quando o documento a ser classificado se demonstrasse demasiado dissemelhante dos demais conhecidos pelo sistema. A solução proposta, parece não ser capaz de desempenhar com rigor a rejeição de documentos quando estes não se revelam, através das *features* em estudo, muito dissemelhantes dos demais, resultando assim, num

módulo que, no contexto prático da problemática trabalhada, não possui robustez suficiente.

Foi também verificado com recurso a *unsupervised learning*, para os atributos através dos quais uma maior *accuracy* foi obtida, qual o melhor número de autores encontrado, justificando assim, quando a cardinalidade de classes é diferente do conhecimento obtido *á priori*, que a classificação de texto cru insere-se num contexto de classificação difícil.

Por fim, foi realizada uma pequena tentativa de classificação usando redes neuronais, fora do escopo do cerne da dissertação, onde foram obtidos resultados promissores, contudo inferiores aos obtidos pelos classificadores tradicionais.

5.2 Trabalho Futuro

A tarefa da Atribuição de Autoria têm, em contexto prático, inúmeras aplicações que beneficiarão se essa atribuição for realizada com uma *accuracy* alta. Deste ponto de vista, qualquer solução que vise esta problemática deve conseguir classificar de forma correta a maioria dos documentos do conjunto de treino, onde o autor é conhecido, para poder *a posteriori* classificar elementos cujo autor é desconhecido.

Como em qualquer tarefa de classificação, as *features* que descrevem um autor, assumem um papel crucial neste processo, pelo que toda e qualquer informação proveniente das mesmas poderá ter um impacto positivo no modelo. De acordo com esta linha de pensamento, abordagens dependentes do idioma, são uma mais valia. A análise de sentimentos, é, no meu entender, uma ferramenta com imenso potencial para a Atribuição de Autoria, pois através da mesma, poderiam ser angariados imensos atributos, que se crê que teriam um poder discriminante alto.

Inerente ao processo de Atribuição de Autoria por parte de um sistema classificador, deve existir também a capacidade do mesmo de rejeitar a atribuição de uma das classes ao documento quando este se revelasse suficientemente diferente de todos os protótipos conhecidos. Caso contrário, somente se existir a garantia que o documento a classificar pertence ao conjunto de classes que o classificador conhece, o que nem sempre é sabido *a priori* em contexto real, é que a solução é passível de ser utilizada, uma vez que em qualquer condição será sempre atribuída ao documento o autor que revele ter as características de escrita mais próximas, em termos relativos, mesmo que estas estejam em termos absolutos muito distantes. Foi, na presente dissertação oferecida uma possível solução teórica que aborda o problema, no entanto, para o caso geral, não oferece ainda uma solução robusta o suficiente, permanecendo assim a problemática por resolver.

A questão da fraca *performance* do módulo de rejeição, deve-se essencialmente

a capacidade da implementação da solução teórica em contexto prático, pois embora existam técnicas de gaussianização dos dados, estas não são, à presente data, passíveis de serem perfeitas.

Assim, em contextos difíceis, onde os centroides apresentam tipicamente uma distribuição espacial semelhante, a juntar ao fato de a gaussianização não ser perfeita, não ser sempre possível obter através da distância quadrática de *mahalanobis* taxas altas de *accuracy* e existir sempre um erro associado ao χ^2 , conduzem à falta de robustez do processo, sendo que, naturalmente, a melhoria destes processos conduziria a obtenção de melhores resultados.

BIBLIOGRAFIA

- [1] A. Belay et al. «English Text Classification by Authorship and Date». Em: () (ver p. 29).
- [2] G. E. P. Box e D. R. Cox. «An Analysis of Transformations». Em: *Journal of the Royal Statistical Society: Series B (Methodological)* 26.2 (1964), pp. 211–243. DOI: <https://doi.org/10.1111/j.2517-6161.1964.tb00553.x>. eprint: <https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1964.tb00553.x>. URL: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1964.tb00553.x> (ver p. 42).
- [3] L. Breiman. «Random forests». Em: *Machine learning* 45.1 (2001), pp. 5–32 (ver p. 16).
- [4] Y.-C. Chen. «A tutorial on kernel density estimation and recent advances». Em: *Biostatistics & Epidemiology* 1.1 (2017), pp. 161–187 (ver p. 14).
- [5] N. Cheng, R. Chandramouli e K. Subbalakshmi. «Author gender identification from text». Em: *Digital Investigation* 8 (2011-07), pp. 78–88. DOI: [10.1016/j.diin.2011.04.002](https://doi.org/10.1016/j.diin.2011.04.002) (ver pp. 25–27).
- [6] C. Cortes e V. Vapnik. «Support vector machine». Em: *Machine learning* 20.3 (1995), pp. 273–297 (ver p. 14).
- [7] G. Cybenko. «Approximation by superpositions of a sigmoidal function». Em: *Mathematics of control, signals and systems* 88.424 (1993), pp. 303–314. DOI: [10.1007/BF02551274](https://doi.org/10.1007/BF02551274) (ver p. 17).
- [8] A. P. Dempster, N. M. Laird e D. B. Rubin. «Maximum likelihood from incomplete data via the EM algorithm». Em: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22 (ver p. 12).
- [9] W. J. Egan e S. L. Morgan. «Outlier Detection in Multivariate Analytical Chemical Data». Em: *Analytical Chemistry* 70.11 (1998). PMID: 21644644, pp. 2372–2379. DOI: [10.1021/ac970763d](https://doi.org/10.1021/ac970763d). URL: <https://doi.org/10.1021/ac970763d> (ver p. 8).

- [10] S. Elmanarelbouanani e I. Kassou. «Authorship Analysis Studies: A Survey». Em: *International Journal of Computer Applications* 86 (2013-12). DOI: [10.5120/15038-3384](https://doi.org/10.5120/15038-3384) (ver pp. 27, 28, 64).
- [11] N. Fernandes et al. *Unification of HDP and LDA Models for Optimal Topic Clustering of Subject Specific Question Banks*. 2020. arXiv: [2011.01035](https://arxiv.org/abs/2011.01035) [cs.IR] (ver p. 34).
- [12] M. Gamon. «Linguistic correlates of style: authorship classification with deep linguistic analysis features». Em: *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*. Geneva, Switzerland: COLING, 2004-08, pp. 611–617. URL: <https://aclanthology.org/C04-1088> (ver pp. 28, 29).
- [13] H. Gomez Adorno et al. «A Graph Based Authorship Identification Approach». Em: 2015-09 (ver p. 29).
- [14] A. S. Hadi e J. S. Simonoff. «Procedures for the Identification of Multiple Outliers in Linear Models». Em: *Journal of the American Statistical Association* 88.424 (1993), pp. 1264–1272. DOI: [10.1080/01621459.1993.10476407](https://doi.org/10.1080/01621459.1993.10476407). eprint: <https://www.tandfonline.com/doi/pdf/10.1080/01621459.1993.10476407>. URL: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1993.10476407> (ver p. 8).
- [15] F. Howedi. «Text Classification for Authorship Attribution Using Naive Bayes Classifier with Limited Training Data». Em: (2014-12) (ver pp. 30, 31).
- [16] M. N. Ingole, M. Bewoor e S. Patil. «Text summarization using expectation maximization clustering algorithm». Em: *International Journal of Engineering Research and Applications* 2.4 (2012), pp. 168–171 (ver p. 34).
- [17] Y. Ko e J. Seo. «Automatic text categorization by unsupervised learning». Em: *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*. 2000 (ver pp. 34, 35).
- [18] R. Lakoff. «Language and woman's place New York». Em: *NY: Harper &* (1975) (ver p. 25).
- [19] J. M. Lourenço. *The NOVAthesis Template User's Manual*. NOVA University Lisbon. 2021. URL: <https://github.com/joaomlourenco/novathesis/raw/master/template.pdf> (ver p. ii).
- [20] A. Mukherjee e B. Liu. «Improving gender classification of blog authors». Em: *Proceedings of the 2010 conference on Empirical Methods in natural Language Processing*. 2010, pp. 207–217 (ver pp. 31, 32).
- [21] F. Peleja, G. P. Lopes e J. Silva. «Text Categorization: A comparison of classifiers, feature selection metrics and document representation». Em: *Proceedings of the 15th Portuguese Conference in Artificial Intelligence*. 2011, pp. 660–674 (ver p. 32).
- [22] S. Srivastava et al. «Germ-line transmission of a mutated p53 gene in a cancer-prone family with Li-Fraumeni syndrome». Em: *Nature* 348.6303 (1990), pp. 747–749 (ver p. 15).

- [23] M. Talbot. *Language and gender: An introduction*. Malden, MA. 1998 (ver p. 25).
- [24] J. F. Teixeira e M. Couto. «Automatic Distinction of Fernando Pessoa's Heteronyms». Em: *Portuguese Conference on Artificial Intelligence*. Springer. 2015, pp. 783–788 (ver pp. 32, 33, 50, 56).
- [25] E. S. Tellez et al. «An automated text categorization framework based on hyperparameter optimization». Em: *Knowledge-Based Systems* 149 (2018), pp. 110–123 (ver p. 24).
- [26] E. S. Tellez et al. «Gender and language-variety Identification with MicroTC.» Em: *CLEF (Working Notes)*. 2017 (ver pp. 24, 25).
- [27] K. Varmuza e P. Filzmoser. *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press, 2009. URL: <https://doi.org/10.1201/9781420059496> (ver p. 8).
- [28] X. Yan e L. Yan. «Gender Classification of Weblog Authors.» Em: *AAAI spring symposium: computational approaches to analyzing weblogs*. Palo Alto, CA. 2006, pp. 228–230 (ver p. 23).

QUADROS AUXILIARES

A.1 Dataset de autores do século 19 e 20

Tabela A.1: Atributos utilizados por classificador para o *dataset* composto por autores do século 19 e 20.

Classificador	Atributos
<i>Support Vector Machines - [SVM]</i>	['2-char', '5-char', 'num medio palavras entre virgulas']
<i>Bagging Classifier - [BG]</i>	['2-char', 'uni-grams', 'quad-grams']
<i>Adaptive Boosting - [ADA]</i>	['2-char', 'quad-grams', 'maior que seis']
<i>Gaussian Naive Bayes - [GNB]</i>	['2-char', '5-char', 'num palavras distintas']
<i>Decision Tree - [DT]</i>	['2-char', 'tri-grams', 'num medio palavras entre pontos exclamação']

Tabela A.2: Legenda das Tabelas dos autores do século 19 e 20.

Sigla	Autor
A	Agustina Bessa Luís
E	Eça de Queirós
In	Inês Pedrosa
L	Lobo Antunes
S	José Saramago

Tabela A.3: *Performance* do classificador *AdaBoost* para o *dataset* contendo autores do século 19 e 20.

Métrica	A	E	In	Is	L	S	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.60	0.92	0.98	0.92	0.87	0.92	0.84	0.87	0.87
<i>Recall</i>	0.88	0.92	1.00	1.00	0.82	0.46	0.84	0.84	0.84
<i>F1-Score</i>	0.71	0.92	0.99	0.96	0.84	0.61	0.84	0.84	0.84
<i>Support</i>	50	50	50	50	50	50	0.84	300	300

Tabela A.4: *Performance* do classificador *Decision Tree* para o *dataset* contendo autores do século 19 e 20.

Métrica	A	E	In	Is	L	S	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	1.00	0.98	1.00	0.98	1.00	1.00	0.99	0.99	0.99
<i>Recall</i>	1.00	0.98	1.00	0.98	1.00	1.00	0.99	0.99	0.99
<i>F1-Score</i>	1.00	0.98	1.00	0.98	1.00	1.00	0.99	0.99	0.99
<i>Support</i>	50	50	50	50	50	50	0.99	300	300

 Tabela A.5: *Performance* do classificador *Gaussian Naive Bayes* para o *dataset* contendo autores do século 19 e 20.

Métrica	A	E	In	Is	L	S	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.94	1.00	1.00	0.92	1.00	0.97	0.97	0.97	0.97
<i>Recall</i>	1.00	1.00	0.96	0.98	0.94	0.96	0.97	0.97	0.97
<i>F1-Score</i>	0.97	1.00	0.97	0.95	0.96	0.96	0.97	0.97	0.97
<i>Support</i>	50	50	50	50	50	50	0.97	300	300

 Tabela A.6: *Performance* do classificador *Support Vector Machines* para o *dataset* contendo autores do século 19 e 20.

Métrica	A	E	In	Is	L	S	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.70	0.90	0.89	0.65	0.75	0.71	0.75	0.77	0.77
<i>Recall</i>	0.82	0.98	0.66	0.98	0.18	0.92	0.75	0.75	0.75
<i>F1-Score</i>	0.75	0.94	0.75	0.78	0.29	0.80	0.75	0.72	0.72
<i>Support</i>	50	50	50	50	50	50	0.75	300	300

 Tabela A.7: *Performance* do classificador *Bagging Classifier* para o *dataset* contendo autores do século 19 e 20.

Métrica	A	E	In	Is	L	S	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	1.00	1.00	1.00	0.98	1.00	1.00	0.99	0.99	0.99
<i>Recall</i>	1.00	1.00	1.00	1.00	0.98	1.00	0.99	0.99	0.99
<i>F1-Score</i>	1.00	1.00	1.00	0.99	0.98	1.00	0.99	0.99	0.99
<i>Support</i>	50	50	50	50	50	50.0	0.99	300	300

A.2 Classificação por género

Tabela A.8: Atributos utilizados por classificador para a classificação por género.

Classificador	Atributos
<i>Support Vector Machines - [SVM]</i>	['num palavras distintas', 'num medio pontos exclamacao', 'not ascii', 'virgulas']
<i>Bagging Classifier - [BG]</i>	['num medio palavras entre pontos', 'num medio pontos exclamacao']
<i>Adaptive Boosting - [ADA]</i>	'num medio palavras entre virgulas', 'num medio pontos exclamacao'
<i>Gaussian Naive Bayes - [GNB]</i>	['num medio palavras entre virgulas', 'num palavras distintas']
<i>Decision Tree - [DT]</i>	[('num medio palavras entre virgulas', 'num medio pontos exclamacao')]

Tabela A.9: Desempenho obtido pelo classificador *Support Vector Machines* para o *dataset* contendo autores de ambos os géneros.

Métrica	Homem	Mulher	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.90	0.93	0.92	0.92	0.92
<i>Recall</i>	0.94	0.90	0.92	0.92	0.92
<i>F1-Score</i>	0.92	0.91	0.92	0.91	0.91
<i>Support</i>	50	50	0.92	100	100

Tabela A.10: Desempenho obtido pelo classificador *Bagging Classifier* para o *dataset* contendo autores de ambos os géneros.

Métrica	Homem	Mulher	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	1.00	1.00	1.00	1.00	1.00
<i>Recall</i>	1.00	1.00	1.00	1.00	1.00
<i>F1-Score</i>	1.00	1.00	1.00	1.00	1.00
<i>Support</i>	50	50	1	100	100

Tabela A.11: Desempenho obtido pelo classificador *AdaBoost* para o *dataset* contendo autores de ambos os géneros.

Métrica	Homem	Mulher	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	1.00	1.00	1.00	1.00	1.00
<i>Recall</i>	1.00	1.00	1.00	1.00	1.00
<i>F1-Score</i>	1.00	1.00	1.00	1.00	1.00
<i>Support</i>	50	50	1	100	100

Tabela A.12: Desempenho obtido pelo classificador *Gaussian Naive Bayes* para o *dataset* contendo autores de ambos os géneros.

Métrica	Homem	Mulher	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.87	0.84	0.86	0.86	0.86
<i>Recall</i>	0.84	0.88	0.86	0.86	0.860
<i>F1-Score</i>	0.85	0.86	0.86	0.85	0.85
<i>Support</i>	50	50	0.86	100	100

 Tabela A.13: Desempenho obtido pelo classificador *Decision Tree* para o *dataset* contendo autores de ambos os géneros.

Métrica	Homem	Mulher	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	1.00	1.00	1.00	1.00	1.00
<i>Recall</i>	1.00	1.00	1.00	1.00	1.00
<i>F1-Score</i>	1.00	1.00	1.00	1.00	1.00
<i>Support</i>	50	50	1	100	100

A.3 Autores de diferentes épocas

 Tabela A.14: Atributos utilizados por classificador para o *dataset* composto por autores de épocas diferentes.

Classificador	Atributos
<i>Support Vector Machines - [SVM]</i>	['num medio palavras entre virgulas', 'num palavras distintas', 'not-ascii']
<i>Bagging Classifier - [BG]</i>	['upper', 'num-virgulas', 'maior que nove', 'num medio palavras entre virgulas']
<i>Adaptive Boosting - [ADA]</i>	['num-e-comercial', 'num-pontos', 'num medio palavras', 'num palavras distintas']
<i>Random Forest - [RF]</i>	['upper', 'num-e-comercial', 'num medio palavras entre pontos exclamação', 'num medio palavras']
<i>Decision Tree - [DT]</i>	['upper', 'num-virgulas', 'num medio palavras entre virgulas', 'num palavras distintas']

 Tabela A.15: Legenda das Tabelas para o *dataset* contendo autores de épocas diferentes.

Sigla	Autor
FC	Fernão Lopes de Castanhede
FMP	Fernão Mendes Pinto
IP	Inês Pedrosa
IS	Isabel Stilwell
LA	Lobo Antunes
P	Padre António Vieira

Tabela A.16: Desempenho obtido pelo classificador *Random Forest* para o *dataset* contendo autores de épocas diferentes.

Métrica	FC	FMP	IP	IS	L	P	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.96	0.97	0.96	1.00	1.00	1.00	0.98	0.98	0.98
<i>Recall</i>	0.98	0.96	1.00	1.00	1.00	0.96	0.98	0.98	0.98
<i>F1 - Score</i>	0.97	0.96	0.98	1.00	1.00	0.97	0.98	0.98	0.98
<i>Support</i>	50	50	50	50.0	50.0	50	0.98	300	300

Tabela A.17: Desempenho obtido pelo classificador *AdaBoost* para o *dataset* contendo autores de épocas diferentes.

Métrica	FC	FMP	IP	IS	LA	P	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.75	0.87	0.94	0.96	0.94	0.96	0.89	0.90	0.90
<i>Recall</i>	0.90	0.94	0.96	0.98	1.0	0.60	0.89	0.89	0.89
<i>F1 - Score</i>	0.81	0.90	0.95	0.97	0.97	0.74	0.89	0.89	0.89
<i>Support</i>	50	50	50	50	50	50	0.89	300	300

Tabela A.18: Desempenho obtido pelo classificador *Bagging Classifier* para o *dataset* contendo autores de épocas diferentes.

Métrica	FC	FMP	IP	IS	LA	P	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.96	0.97	1.00	1.00	1.00	1.00	0.99	0.99	0.99
<i>Recall</i>	0.98	0.96	1.00	1.00	1.00	1.00	0.99	0.99	0.99
<i>F1-Score</i>	0.97	0.96	1.00	1.00	1.00	1.00	0.99	0.98	0.98
<i>Support</i>	50	50	50	50	50	50	0.99	300	300

Tabela A.19: Desempenho obtido pelo classificador *Support Vector Machines* para o *dataset* contendo autores de épocas diferentes.

Métrica	FC	FMP	IP	IS	LA	P	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.96	0.80	0.98	0.87	0.97	0.98	0.92	0.93	0.93
<i>Recall</i>	0.98	0.84	0.98	1	0.78	0.98	0.92	0.92	0.92
<i>F1-Score</i>	0.97	0.82	0.98	0.93	0.86	0.98	0.92	0.92	0.92
<i>Support</i>	50	50	50	50	50	50	0.92	300	300

Tabela A.20: Desempenho obtido pelo classificador *Decision Tree* para o *dataset* contendo autores de épocas diferentes.

Métrica	FC	FMP	IP	IS	LA	P	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.96	0.96	1.00	1.00	1.00	1.00	0.98	0.98	0.98
<i>Recall</i>	0.96	0.96	1.00	1.00	1.00	1.00	0.98	0.98	0.98
<i>F1-Score</i>	0.96	0.96	1.00	1.00	1.00	1.00	0.98	0.98	0.98
<i>Support</i>	50	50	50	50	50	50	0.98	300	300

A.4 Dataset Fernando Pessoa

Tabela A.21: Legenda das Tabelas para o *dataset* contendo quatro heterónimos de Fernando Pessoa.

Sigla	Autor
AC	Alberto Caeiro
BS	Bernardo Soares
RR	Ricardo Reis
ÁDC	Álvaro de Campos

A.4.1 Dataset completo

Tabela A.22: Atributos utilizados por classificador para o *dataset* contendo quatro heterónimos de Fernando Pessoa.

Classificador	Atributos
<i>Support Vector Machines</i> - [SVM]	['quad-grams', 'uni-grams', 'Virgulas', 'Pontos', 'tamanho-frase', 'palavras entre virgulas']
<i>Random Forest</i> - [RF]	['quad-grams', 'uni-grams', 'Virgulas', 'Pontos', 'tamanho-frase', 'palavras entre virgulas']
<i>Adaptive Boosting</i> - [ADA]	['num-versos', 'tamanho-medio-palavra', 'Exclamação', 'Interrogação']
<i>Gaussian Naive Bayes</i> - [GNB]	['quad-grams', 'uni-grams', 'Virgulas', 'Pontos', 'tamanho-frase', 'palavras entre virgulas']
<i>Decision Tree</i> - [DT]	['quad-grams', 'uni-grams', 'Virgulas', 'Pontos', 'tamanho-frase', 'palavras entre virgulas']

Tabela A.23: Desempenho obtido pelo classificador *Random Forest* para o *dataset* contendo quatro heterónimos de Fernando Pessoa.

Métrica	AC	BS	RR	ÁDC	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.89	0.91	0.89	0.88	0.90	0.89	0.90
<i>Recall</i>	0.80	0.91	0.96	0.87	0.90	0.88	0.90
<i>F1-Score</i>	0.84	0.91	0.92	0.88	0.90	0.8	0.90
<i>Support</i>	127	504	307	397	0.90	1335	1335

Tabela A.24: Desempenho obtido pelo classificador *Decision Tree* para o *dataset* contendo quatro heterónimos de Fernando Pessoa.

Métrica	AC	BS	RR	ÁDC	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.76	0.88	0.88	0.82	0.85	0.841744	0.85
<i>Recall</i>	0.77	0.88	0.87	0.83	0.85	0.84	0.85
<i>F1-Score</i>	0.77	0.88	0.88	0.82	0.85	0.84	0.85
<i>Support</i>	127	504	307	397	0.85	1335	1335

Tabela A.25: Desempenho obtido pelo classificador *Gaussian Naive Bayes* para o *dataset* contendo quatro heterónimos de Fernando Pessoa.

Métrica	AC	BS	RR	ÁDC	Accuracy	Macro avg	Weighted avg
<i>Precision</i>	0.49	0.71	0.81	0.92	0.72	0.73	0.77
<i>Recall</i>	0.88	0.99	0.58	0.43	0.72	0.72	0.72
<i>F1-Score</i>	0.63	0.83	0.67	0.59	0.72	0.68	0.70
<i>Support</i>	127	504	307	397	0.72	1335	1335

Tabela A.26: Desempenho obtido pelo classificador *Support Vector Machines* para o *dataset* contendo quatro heterónimos de Fernando Pessoa.

Métrica	AC	BS	RR	ÁDC	Accuracy	Macro avg	Weighted avg
<i>Precision</i>	0.66	0.79	0.62	0.73	0.72	0.70	0.72
<i>Recall</i>	0.015	0.86	0.91	0.62	0.72	0.60	0.72
<i>F1-Score</i>	0.030	0.82	0.74	0.67	0.72	0.57	0.68
<i>Support</i>	127	504	307	397	0.72	1335	1335

Tabela A.27: Desempenho obtido pelo classificador *AdaBoost* para o *dataset* contendo quatro heterónimos de Fernando Pessoa.

Métrica	AC	BS	RR	ÁDC	Accuracy	Macro avg	Weighted avg
<i>Precision</i>	0.31	0.77	0.77	0.63	0.69	0.62	0.68
<i>Recall</i>	0.24	0.84	0.71	0.64	0.69	0.61	0.69
<i>F1-Score</i>	0.27	0.80	0.74	0.63	0.69	0.61	0.69
<i>Support</i>	127	504	307	397	0.69	1335	1335

A.4.2 Dataset incompleto

Tabela A.28: Melhor combinação de atributos obtida por classificador para o *dataset* contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.

Classificador	Atributos
<i>Support Vector Machines - [SVM]</i>	['quad - grams', 'uni - grams', 'Virgulas', 'Pontos', 'tamanho - frase', 'palavrasentrevirgulas']
<i>Gaussian Naive Bayes - [GNB]</i>	['quad - grams', 'uni - grams', 'Virgulas', 'Pontos', 'tamanho - frase', 'palavrasentrevirgulas']
<i>Adaptive Boosting - [ADA]</i>	['quad - grams', 'uni - grams', 'Virgulas', 'Pontos', 'tamanho - frase', 'palavrasentrevirgulas']
<i>Random Forest - [RF]</i>	['quad - grams', 'uni - grams', 'Virgulas', 'Pontos', 'tamanho - frase', 'palavrasentrevirgulas']
<i>Decision Tree - [DT]</i>	['quad - grams', 'uni - grams', 'Virgulas', 'Pontos', 'tamanho - frase', 'palavrasentrevirgulas']

Tabela A.29: Desempenho obtido pelo classificador *AdaBoost* para o *dataset* contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.

Métrica	AC	BS	RR	ÁDC	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.0	0.94	0.46	1.00	0.69	0.60	0.60
<i>Recall</i>	0.0	1.00	1.00	0.79	0.69	0.69	0.69
<i>F1-Score</i>	0.0	0.96	0.63	0.88	0.69	0.62	0.62
<i>Support</i>	127	127	127	127	0.69	508	508

 Tabela A.30: Desempenho obtido pelo classificador *Random Forest* para o *dataset* contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.

Métrica	AC	BS	RR	ÁDC	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.93	0.97	0.89	0.95	0.94	0.94	0.94
<i>Recall</i>	0.95	1.00	0.88	0.92	0.94	0.94	0.94
<i>F1-Score</i>	0.94	0.98	0.89	0.93	0.94	0.94	0.94
<i>Support</i>	127	127	127	127	0.94	508	508

 Tabela A.31: Desempenho obtido pelo classificador *Decision Tree* para o *dataset* contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.

Métrica	AC	BS	RR	ÁDC	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.95	1.00	0.88	0.95	0.94	0.94	0.94
<i>Recall</i>	0.92	0.99	0.91	0.95	0.94	0.94	0.94
<i>F1-Score</i>	0.94	0.99	0.89	0.95	0.94	0.94	0.94
<i>Support</i>	127	127	127	127	0.94	508	508

 Tabela A.32: Desempenho obtido pelo classificador *Gaussian Naive Bayes* para o *dataset* contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.

Métrica	AC	BS	RR	ÁDC	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.66	1.00	0.71	0.93	0.82	0.82	0.82
<i>Recall</i>	0.92	0.99	0.50	0.86	0.82	0.82	0.82
<i>F1-Score</i>	0.77	0.99	0.59	0.89	0.82	0.81	0.81
<i>Support</i>	127	127	127	127	0.82	508	508

Tabela A.33: Desempenho obtido pelo classificador *Support Vector Machines* para o *dataset* contendo quatro heterónimos de Fernando Pessoa com regularização de elementos por autor.

Métrica	AC	BS	RR	ÁDC	<i>Accuracy</i>	<i>Macro avg</i>	<i>Weighted avg</i>
<i>Precision</i>	0.62	0.90	0.63	0.82	0.74	0.74	0.74
<i>Recall</i>	0.70	1.00	0.52	0.75	0.74	0.74	0.74
<i>F1-Score</i>	0.66	0.94	0.57	0.79	0.74	0.74	0.74
<i>Support</i>	127	127	127	127	0.74	508	508

FIGURAS AUXILIARES

B.1 Datasets de autores do século 10 e 20

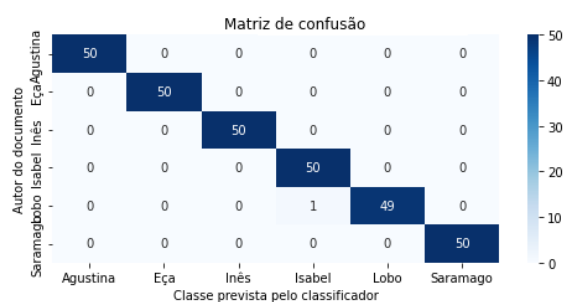


Figura B.1: Matriz de confusão para o classificador *Bagging Classifier* para o *dataset* contendo autores do século 19 e 20.

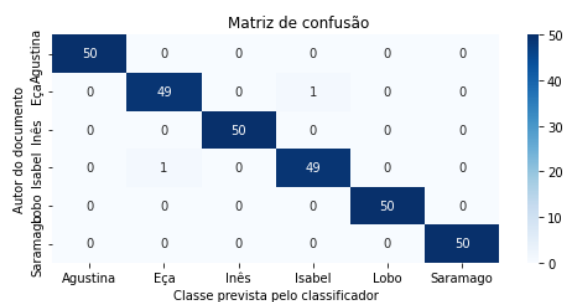


Figura B.2: Matriz de confusão para o classificador *Decision Tree* para o *dataset* contendo autores do século 19 e 20.

B.1. DATASETS DE AUTORES DO SÉCULO 19 E 20

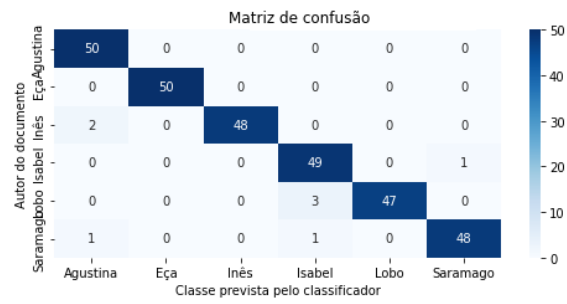


Figura B.3: Matriz de confusão para o classificador *Gaussian Naive Bayes* para o *dataset* contendo autores do século 19 e 20..

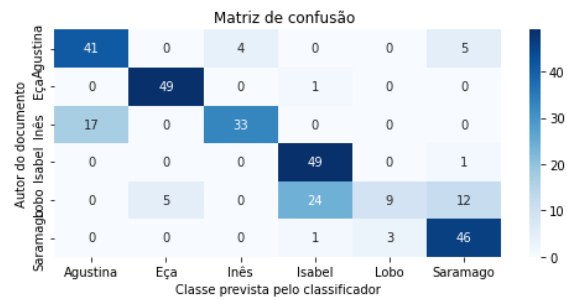


Figura B.4: Matriz de confusão para o classificador *Support Vector Machines* para o *dataset* contendo autores do século 19 e 20..

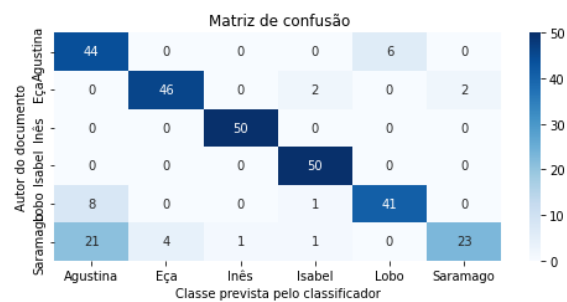


Figura B.5: Matriz de confusão para o classificador *AdaBoost* para o *dataset* contendo autores do século 19 e 20..

B.2 Classificação por género

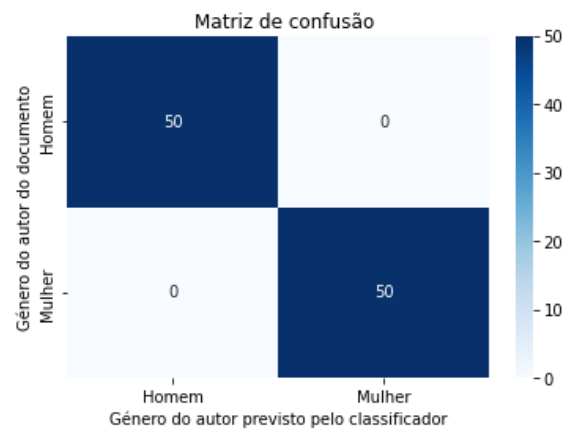


Figura B.6: Matriz de confusão para o classificador *Bagging Classifier* para a classificação por género.

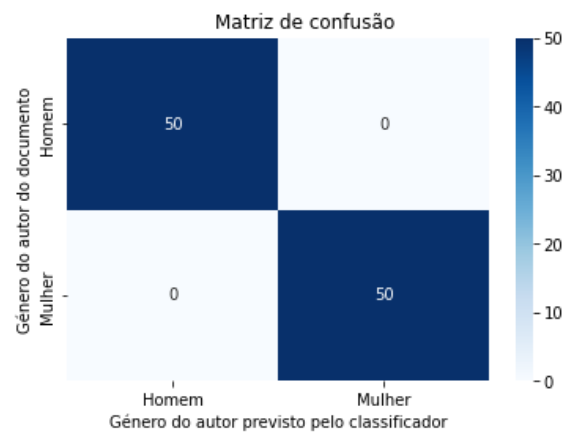


Figura B.7: Matriz de confusão para o classificador *Decision Tree* para a classificação por género.



Figura B.8: Matriz de confusão para o classificador *Gaussian Naive Bayes* para a classificação por gênero.

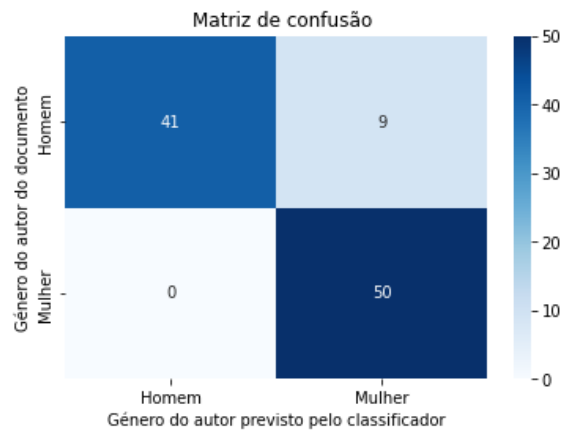


Figura B.9: Matriz de confusão para o classificador *Support Vector Machines* para a classificação por gênero.

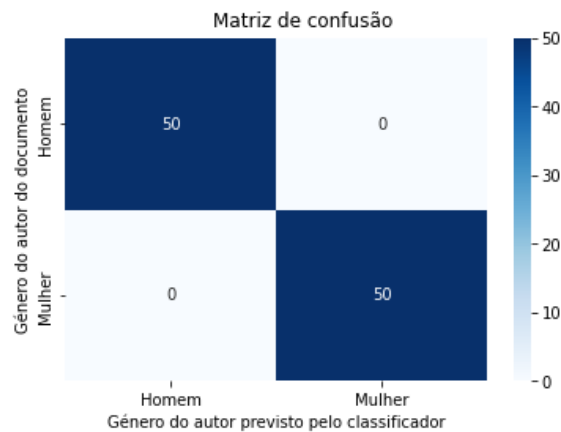


Figura B.10: Matriz de confusão para o classificador *AdaBoost* para a classificação por gênero.

B.3 Autores de diferentes épocas

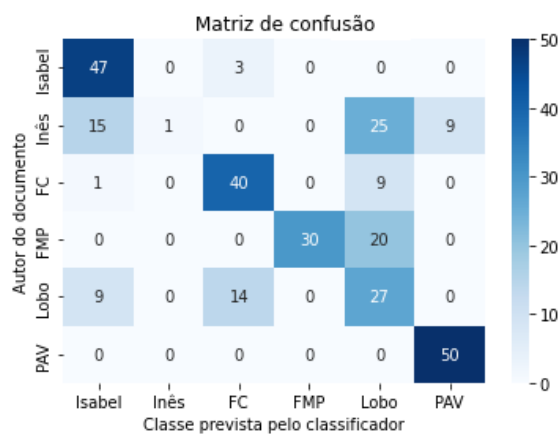


Figura B.11: Matriz de confusão para o classificador *Bagging Classifier* para o *dataset* contendo autores de épocas diferentes.

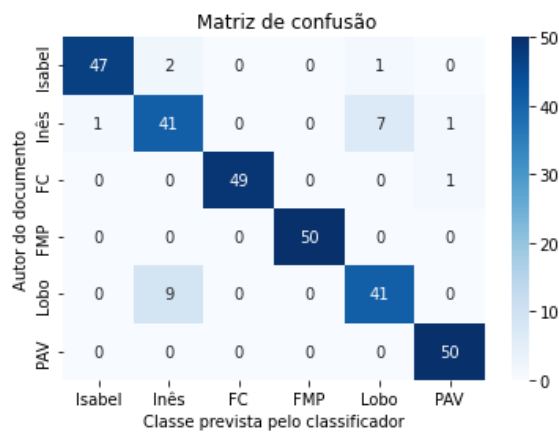


Figura B.12: Matriz de confusão para o classificador *Decision Tree* para o *dataset* contendo autores de épocas diferentes.

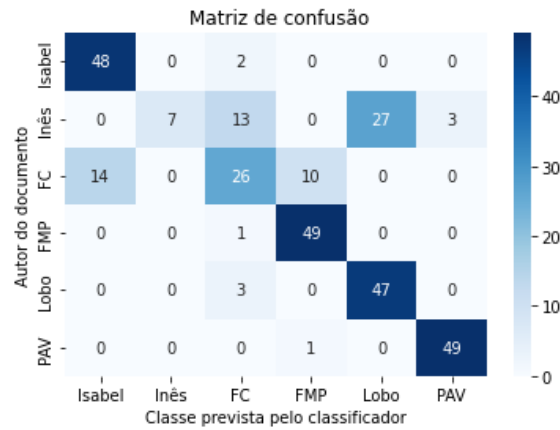


Figura B.13: Matriz de confusão para o classificador *Random Forest* para o *dataset* contendo autores de épocas diferentes.

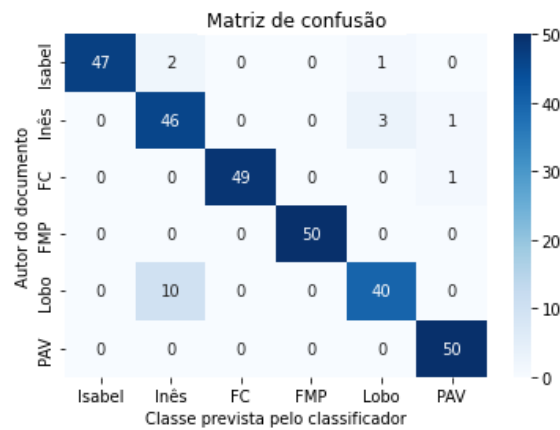


Figura B.14: Matriz de confusão para o classificador *Support Vector Machines* para o *dataset* contendo autores de épocas diferentes.

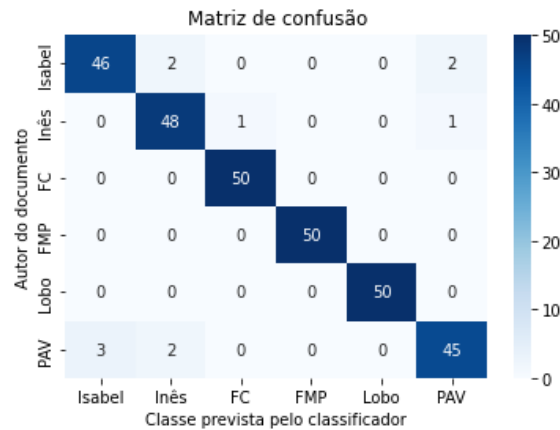


Figura B.15: Matriz de confusão para o classificador *AdaBoost* para o *dataset* contendo autores de épocas diferentes.

B.4 Dataset Fernando Pessoa

B.4.1 Dataset completo

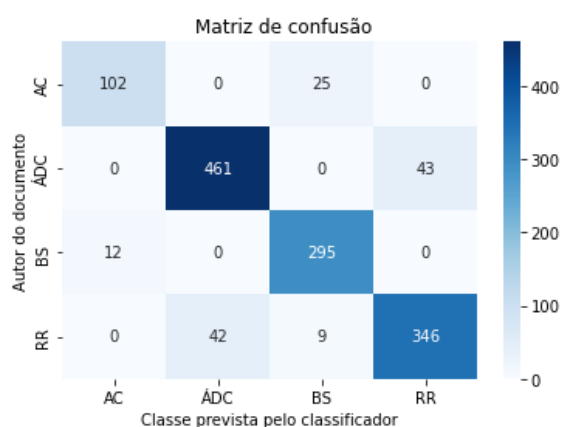


Figura B.16: Matriz de confusão para o classificador *Random Forest* para o *dataset* contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por autor.

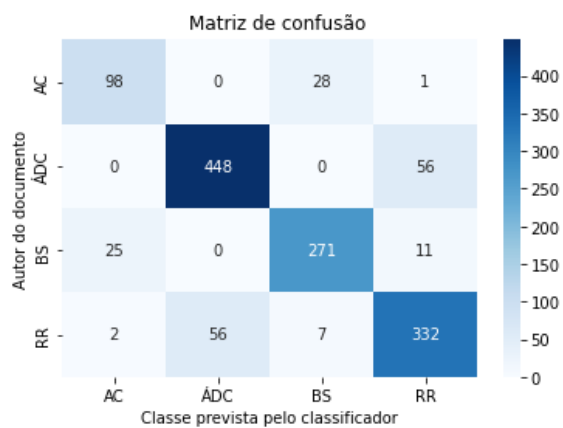


Figura B.17: Matriz de confusão para o classificador *Decision Tree* para o *dataset* contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por autor.

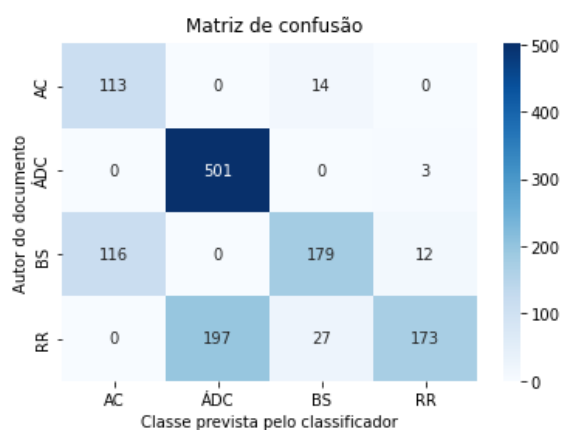


Figura B.18: Matriz de confusão para o classificador *Gaussian Naive Bayes* para o *dataset* contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por autor.

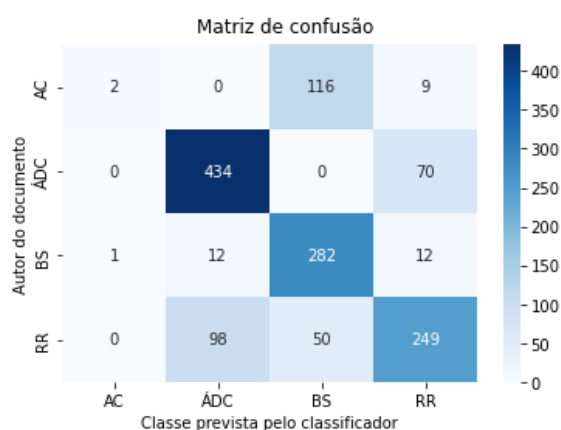


Figura B.19: Matriz de confusão para o classificador *Support Vector Machines* para o *dataset* contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por autor.

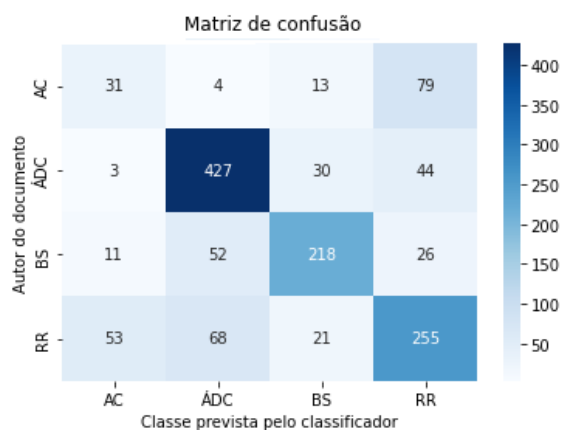


Figura B.20: Matriz de confusão para o classificador *AdaBoost* para o *dataset* contendo quatro heterónimos de Fernando Pessoa utilizando todos os documentos por autor.

B.4.2 Dataset incompleto

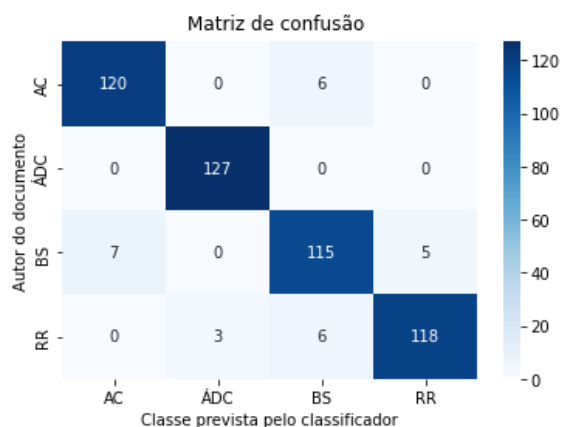


Figura B.21: Matriz de confusão para o classificador *Random Forest* para o *dataset* contendo quatro heterónimos de Fernando Pessoa com normalização do número de documentos por autor.

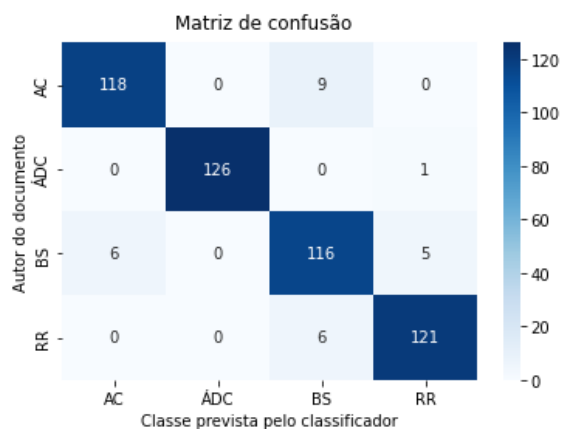


Figura B.22: Matriz de confusão para o classificador *Decision Tree* para o *dataset* contendo quatro heterónimos de Fernando Pessoa com normalização do número de documentos por autor.

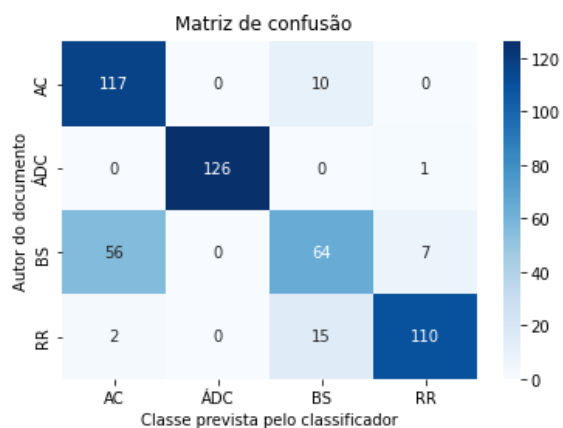


Figura B.23: Matriz de confusão para o classificador *Gaussian Naive Bayes* para o *dataset* contendo quatro heterónimos de Fernando Pessoa com normalização do número de documentos por autor.

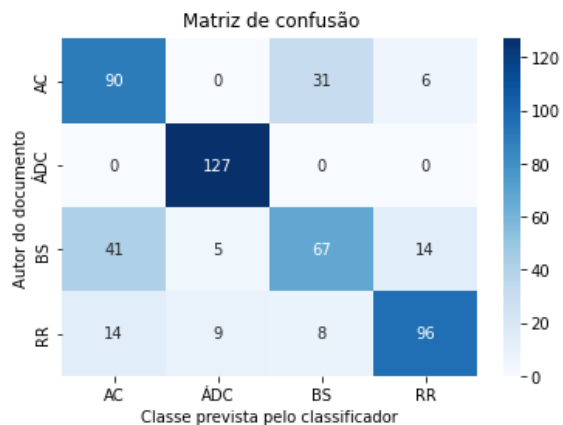


Figura B.24: Matriz de confusão para o classificador *Support Vector Machines* para o *dataset* contendo quatro heterónimos de Fernando Pessoa com normalização do número de documentos por autor.

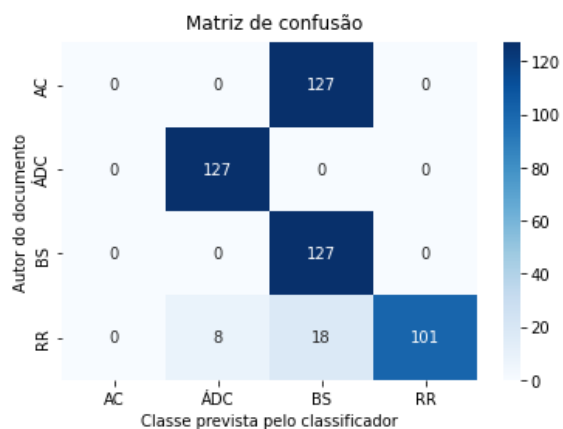


Figura B.25: Matriz de confusão para o classificador *AdaBoost* para o *dataset* contendo quatro heterónimos de Fernando Pessoa com normalização do número de documentos por autor.



