Instituto Politécnico
de Viana do Castelo

DETEÇÃO DE E-MAILS DE PHISHING APLICANDO MACHINE LEARNING AO CONTEÚDO

Marco Saraiva - João Paulo Magalhães - Silvestre Malta

Escola Superior de Tecnologia e Gestão

Instituto Politécnico
de Viana do Castelo

Nome completo do candidato(a)

MARCO ANTÓNIO CARVALHOSA SARAIVA

Nome do curso de Mestrado

Mestrado em Cibersegurança

Trabalho efetuado sob a supervisão de

Professor João Paulo Magalhães

Professor Silvestre Malta

Junho de 2022

Mestrado em Cibersegurança
Master in Cybersecurity

Deteção de e-mails de phishing aplicando Machine Learning ao conteúdo

a master's thesis authored by

Marco António Carvalhosa Saraiva

and supervised by

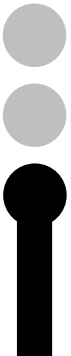
João Paulo Magalhães
CIICESI, ESTG, Politécnico do Porto

Silvestre Malta
AdiT-Lab ESTG, Instituto Politécnico de Viana do Castelo

This thesis was submitted in partial fulfilment of the requirements for the Master's degree in Cybersecurity at the Instituto Politécnico de Viana do Castelo

ipvc

27 of October, 2022



Abstract

Social engineering is a concept in which psychological manipulation is applied to get the victim to perform actions on behalf of the malicious actor. One of the most common forms of social engineering is *phishing*. In the cyber world, *phishing* is used to manipulate users into extortion and theft of sensitive data. This practice is increasingly used, which makes it worrying and alarming how it is possible to be the target of this attack. Reports in the area reveal that users are constantly being targeted by e-mails that pretend to be legitimate, but are actually victims of a *phishing* attack. The number of *phishing* websites and e-mail campaigns continues to grow year after year, and in 2021, *phishing* attacks grew by 200% due to to take advantage of the Covid-19 pandemic with campaigns for vaccines and treatment for the virus.

There is great concern from the academic community and the industry to mitigate the problem, but the challenges are many. To a certain extent, this is due to the fact that part of the solution involves human beings, developing their ability to be aware and make correct decisions to prevent the attack from being successful.

Addressing a problem like *phishing* requires people, procedural, and technology (PPT) action. The human side has been the target of constant training and awareness actions, but even so the phenomenon continues to grow. In this work we propose a technological solution to reinforce the ability to mitigate *phishing* attacks, that is, to create a line of defense so that the end user does not have to deal with e-mails *phishing* on a daily basis, in order to avoid human error and create possible damage and losses.. The proposal presented involves the creation of a *dataset* from e-mails previously classified as *phishing* and not *phishing*. To create the *dataset*, linguistic aspects of the e-mail itself were considered. For this, an automated information extraction technique, called Named-entity Recognition (NER) was applied. This technique removes the characteristics that form the data set

from the body of the e-mail. The resulting dataset was analyzed, treated and submitted to Machine Learning (ML) algorithms, more specifically to classification algorithms.

The analysis of results allows us to conclude that, through this method, it is possible to determine if an e-mail is from *phishing* and or with a hit rate of 91.13%. It was also possible to conclude that the choice of *features* for the training phase of ML models has a preponderant role in maximizing the hit rate. It should be noted that the proposal presented here to determine whether an e-mail is from *phishing* or not can simply be integrated with other solutions, thus improving the ability to detect and prevent this type of attack.

Palavras-chave: phishing. ML. *dataset*. e-mails.

Resumo

A engenharia social é um conceito no qual é aplicado a manipulação psicológica para levar a vítima a executar ações em prol do ator malicioso. Uma das formas mais comuns de praticar atos de engenharia social é o *phishing*. No mundo cibernético, o *phishing* é usado para manipular os utilizadores para a prática de extorsão e roubo de dados sensíveis. Esta prática é cada vez mais usada, o que torna preocupante e alarmante a forma de como é possível ser alvo deste ataque. Relatórios na área revelam que os utilizadores estão constantemente a ser alvo de e-mails que fingem ser legítimos, mas que na verdade estão a ser vítimas de um ataque *phishing*. O número de sites e de campanhas de e-mail de *phishing* continua a crescer ano após ano, sendo que, e a título de exemplo, no ano de 2021, os ataques *phishing* cresceram 200% devido ao aproveitamento da pandemia Covid-19 com campanhas de vacinas e tratamento para o vírus.

Existe uma grande preocupação da comunidade académica e da indústria em mitigar o problema porém os desafios são muitos. Tal deve-se em certa medida ao fato de que parte da solução passa pelo ser humano, desenvolvendo a capacidade do mesmo ter consciência e tomar decisões corretas para evitar que o ataque seja bem-sucedido.

Endereçar um problema como o do *phishing* requer ações ao nível pessoas, procedimental e tecnológico (PPT). O lado humano tem sido alvo de constantes ações de treino e consciencialização, mas mesmo assim o fenómeno não para de crescer. Neste trabalho propomos uma solução tecnológica para reforçar a capacidade de mitigar ataques de *phishing*, ou seja, criar uma linha de defesa para que o utilizador final não tenha de lidar com e-mails *phishing* no dia a dia, de forma a evitar o erro humano e assim criar possíveis estragos e prejuízos. A proposta apresentada envolve a criação de um *dataset* a partir de e-mails previamente classificados como sendo de *phishing* e não *phishing*. Para a criação do *dataset* foram considerados aspetos linguísticos do próprio e-mail. Para tal foi apli-

cada uma técnica de extração de informação automatizada, denominada de Named-entity Recognition (NER). Esta técnica retira do corpo do e-mail características que formam o conjunto de dados. O conjunto de dados resultado foi analisado, tratado e submetido a algoritmos de ML, mais propriamente a algoritmos de classificação.

A análise de resultados permite concluir que, através deste método é possível determinar se um e-mail é de *phishing* e ou com uma taxa de acerto de 91.13%. Foi ainda possível concluir que a escolha das *features* para a fase de treino dos modelos de ML tem um papel preponderante para maximizar a taxa de acerto. Salienta-se que a proposta aqui apresentada para determinar se um e-mail é de *phishing* ou não poderá de forma simples ser integrada com outras soluções, melhorando assim a capacidade de detetar e evitar este tipo de ataques.

Keywords: phishing. ML. *dataset*. e-mails.

Agradecimentos

Em primeiro, agradeço aos professores João Paulo Magalhães e Silvestre Malta, pela ajuda, amizade, incentivo e rigor na realização desta fase importante da minha vida.

Ao meu antigo curso, Engenharia Eletrónica e Redes de Computadores, pelo incentivo e gosto de ter continuado a minha formação e atingir esta fase.

À minha namorada, por todo o apoio e ajuda nos momentos mais difíceis desta caminhada.

A toda a minha família e amigos que me fizeram continuar e nunca ter desistido da minha formação académica.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

APWG Anti-Phishing Working Group

CPO Position of the domain token

DTC DecisionTreeClassifiers

FBI Federal Bureau of Investigation

GBM Gradient Boosted Decision

GNB Gaussian Naïve Bayes

HTTPS Hyper Text Transfer Protocol Secure

IA Inteligência Artificial

ISTR Internet Security Threat Report

KNN K-Nearest Neighbor

LIR Linear Regression

LOR Logistic Regression

MCyber Master in Cybersecurity

ML Machine Learning

MLPC Multi-layer Perceptron Classifier

NB Naïve Bayes

NER Named Entity Recognition

NLTK Natural Language Toolkit

nm NearMiss

NN Neural Network

PDR Public Domain Registry

PLN Processamento de Linguagem Natural

POS Part of Speech

RDT Ratio of the found domain token

RF Random Forest

ROS Random Over-Sampling

RoSI Random Over-Sampling with imblearn

RU Randomness of the URL

RUS Random Under-Sampling

RUSI Random Under-Sampling with imblearn

SKB SelectKBest

SMOTE Synthetic Minority Oversampling Technique

SVM Support Vector Machine

TF-IDF Term Frequency Inverse Document Frequency

TLD Top-level domain

URL Uniform Resource Locator

USTL Under-sampling: Tomek links

Capítulo 1

Introdução

Foi a 29 de Outubro de 1969 que foi partilhado aquilo que é considerado primeiro e-mail da história [1]. Desde então, o volume de e-mails trocados online não pára de evoluir e ainda nos dias de hoje, o e-mail faz parte da nossa vida pessoal e profissional. De acordo com as estatísticas publicadas em [**e-mailporminuto**] estima-se que, por minuto, sejam enviados mais de 196 milhões de e-mails.

Se por um lado o crescimento do mundo cibernético trás novas funcionalidades, interligando pessoas e negócios a uma escala global e de forma simples, por outro é usado com fins menos lícitos. Este mundo cibernético é usado também para práticas criminosas na tentativa de obter informação e/ou dados de forma ilícita, para cometer fraude, para extorquir dinheiro, por exemplo, através de ataques de ransomware. A forma como os atores maliciosos atuam é variada. A disseminação de software malicioso através da Internet é amplamente utilizada, a exploração de vulnerabilidades em sistemas e aplicações permite também a execução de ciberataques e o lado humano é também explorado no âmbito de ataques de engenharia social. Dentro da engenharia social há um tipo de ataque que se destaca pela expressão que tem. Este tipo de ataque denomina-se de *phishing* e, de acordo com [2] é uma atividade que teve origem em 1996 onde um grupo de atores maliciosos roubava contas no America Online (AOL). O grupo enviava mensagens de e-mail com links falsos para a recolha dos dados dos utilizadores. Hoje em dia, este tipo de ataques causam prejuízos elevados. De acordo com o relatório do Federal Bureau of Investigation (FBI) [3] no ano de 2020, juntando os ataques de /*Business e-mail Comproiese* (BEC) e os ataques de *phishing* tradicionais verifica-se que as empresas foram lesadas em mais de

1.8 mil milhões de euros. Ainda, e de acordo com o mesmo relatório, os ataques causam sobretudo:

- Roubo de dados;
- Roubo de contas e credenciais de utilizadores;
- Distribuição de malware, incluindo ransomware (Restringe o acesso ao sistema infetado, em troca de um pagamento para remover o Ransomware);
- Perdas financeiras (e.g. CEO-Fraud)[3].

O *phishing* não é um fenómeno novo. Trata-se de uma prática que visa iludir um utilizador, de forma que ele partilhe informações confidenciais, como palavras-passe, números de cartões de crédito, entre outros. No passado foi usado para falsificar cheques, iludir inimigos em ataques de guerra, viajar gratuitamente em transportes públicos entre muitos outros. Esta prática enquadra-se nas técnicas de engenharia social que, de forma genérica, fazem manipulação psicológica de pessoas para que estas executem ações em prol do engenheiro social, acreditando no mesmo como alguém de bem. Por sua vez, os ataques de *phishing* são divididos em vários tipos. Os mais comuns são:

- Spear*phishing*: Tratam-se de ataques de *phishing* dirigidos e um alvo específico, podendo o alvo ser uma pessoa, uma organização ou um determinado setor de negócio. Este tipo de ataque requer um reconhecimento do alvo, para ser o mais credível possível;
- Blind*phishing*: Consiste no envio, em massa, de e-mails. Este ataque tem como alvo pessoas que não tem conhecimento na área da cibersegurança e que estão mais suscetíveis de ser enganadas;
- Smishing: Tem o mesmo propósito do Blind*phishing* sendo o canal o envio de mensagens para o telemóvel;
- Whaling: É usado quando o alvo é uma pessoa de destaque numa empresa (e.g., CEO, CFO). Neste tipo de ataque o ator malicioso faz-se por exemplo passar pelo CEO indicando ao CFO o pagamento ou transferência monetária para uma determinada conta bancária. Este ataque torna-se particularmente perigoso quando

combinado com BEC (Business e-mail Compromise) na medida em que o remetente da mensagem para a ser o endereço legítimo, maximizando o sucesso do ataque;

- Scam: Quando os atores maliciosos tentam obter informações das vítimas através de links e ficheiros, podendo ser realizado por telefone, e-mail, mensagem de texto ou redes sociais;

- Vishing: É efetuado, por meio de uma chamada de voz, de forma a fazer se passar por um representante de uma entidade, alegando, por exemplo, que existe algum problema na máquina do utilizador. Tipicamente pedem dados do cartão de crédito para instalação de algum software, pedem também acesso remoto à máquina da vítimas e o pagamento simbólico (que é depois alterado) de um valor para aquisição de software de proteção.

É importante indicar que o estudo adotado, faz referência apenas aos ataques *phishing* que são efetuados via e-mail. O exemplo do Vishing e o Smishing não se aplicam ao estudo, apenas servem de exemplos de ataques phishing existentes. A maioria dos ataques *phishing*, tal como referido em [4], provém dos e-mails que são enviados diariamente. De acordo com os dados, 96% dos ataques são entregues via e-mail, 3% chegam aos utilizadores através de sites maliciosos e 1% via telefone. O estudo indica ainda que, 1 em cada 4200 e-mails é considerado *phishing*. Cruzando esta informação com a referida em [**e-mailporminuto**], pode-se concluir que, por minuto são enviados mais de 46666 e-mails de *phishing*.

Segundo o relatório [3] do FBI, o *phishing* foi a atividade de cibercrime mais comum em 2020, tendo o dobro dos incidentes (mais de 240 mil) face ao ano 2019 (mais de 114 mil) e cerca de 11 vezes mais quando comparado com 2016. Um dos fatores que afetou este crescimento, foi a questão da doença Covid-19, onde as pessoas eram bombardeadas com e-mails com propagandas de vacinas, curas, entre outros. Esta evolução é ilustrada na Figura 1.1.

Segundo o 2021 State of the Phish Report [5], 74% dos ataques *phishing* aos utilizadores dos Estados Unidos da América foram executados com sucesso, seguido do Reino Unido com cerca de 66% dos ataques sucedidos e da Austrália com 60%. Esta amostra é representativa a cada país.

Existem estudos no sentido de perceber o porquê dos utilizadores serem afetados por

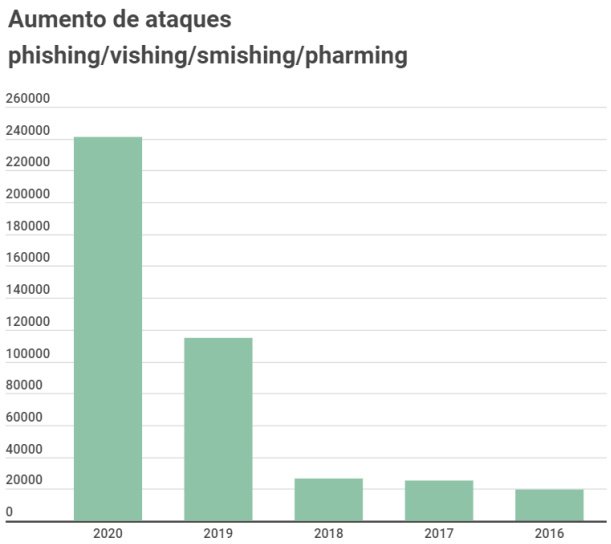


Figura 1.1: Aumento dos ataques *phishing* entre 2020 e 2016 [3]

ataques de *phishing*. Por exemplo, o relatório da Symantec's 2019 Internet Security Threat Report (ISTR) [6] refere que os atores maliciosos utilizam palavras-chave para chamar a atenção do utilizador. Entre as palavras encontram-se:

- Urgente;
- Pedido;
- Importante;
- Pagamentos;
- Prémio;
- Atenção.

Os números apresentados relativos ao *phishing* reforçam o problema do mesmo e a dificuldade em o tratar. As atividades de consciencialização e treino dos utilizadores para o *phishing* tem vindo a aumentar, mas o que é certo é que o número de ataques *phishing* não pára de crescer. Neste âmbito torna-se necessário encontrar novas soluções tecnológicas para auxiliar o combate ao *phishing*, isto porque o ser humano é suscetível a erros, que podem ser aproveitados pelos atores maliciosos. Estas soluções surgem tanto pela comunidade científica como pela indústria.

Nesta dissertação é apresentado um sistema de classificação dos e-mails, de forma a identificar se os mesmos são de *phishing* ou não. O sistema faz uso de algoritmos de Machine Learning (ML) e tem como ponto de partida para análise o conteúdo do e-mail. Mais especificamente, partindo de um conjunto de e-mails previamente classificados como *phishing* e não *phishing*, o projeto contempla a análise do corpo/conteúdo do e-mail, nomeadamente a forma como é escrito, tirando partido do processamento de linguagem natural. Os dados linguísticos extraídos são incluídos num *dataset* e combinados com diferentes *features* (e.g., número de erros de linguagem, número de referências a unidades monetárias, tempos verbais, localizações). O *dataset* inicial é posteriormente tratado tendo como base a análise exploratória de dados (e.g. verificação e tratamento de valores nulos, remoção de caracteres inválidos, análise da importância das *features*, entre outros). Uma vez tratado o *dataset* é analisado recorrendo a diferentes algoritmos de ML (e.g. Logistic Regression (LOR), DecisionTreeClassifiers (DTC), Support Vector Machine (SVM), Multilayer Perceptron Classifier (MLPC), Random Forest (RF) e Gaussian Naïve Bayes (GNB)).

Para cada algoritmo são recolhidas métricas de desempenho. A eficácia na classificação é usada para comparar os diferentes algoritmos, avaliando quais os que mais se adequam à classificação de e-mails de *phishing* baseado no conteúdo do próprio e-mail.

O principal contributo deste projeto é analisar se a deteção de e-mails de *phishing* pode ser complementada através da análise automática do conteúdo do próprio e-mail. A conjugação de meios técnicos para a deteção de *phishing* contribuirá para a redução do número de e-mails deste tipo que chegam ao utilizador final e/ou apoiando o utilizador com a indicação de potencial e-mail de *phishing*.

Esta dissertação está organizada em capítulos. O Capítulo 2 apresenta trabalho já feito na área, nomeadamente trabalhos relacionados com algoritmos de ML e soluções anti-phishing existentes. O terceiro capítulo, 3, retrata as diferentes etapas do estudo. O capítulo 4 descreve a forma como foi criado e tratado o *dataset*. O capítulo 5 é composto pela seleção das *features* mais significativas, a fase de treino dos algoritmos e o impacto que a seleção de *features* tem na análise. Este capítulo apresenta também os resultados obtidos pelos diferentes algoritmos de ML e considerando os diferentes algoritmos de seleção de *features*. Por fim, no capítulo 6 é feita a conclusão do estudo.

Capítulo 2

Estado da Arte

Os ataques *phishing* são considerados a maior ameaça no ciberespaço. É um problema difícil de controlar e evitar. A melhor abordagem é garantir mecanismos de deteção de *phishing* para mitigar o impacto negativo que este tipo de ataque traz.

2.1 Relatórios Anti-Phishing Working Group

A Anti-Phishing Working Group (APWG) é uma coligação que unifica a resposta global ao crime cibernético em setores industriais e governamentais. Eles fornecem um relatório sobre cada trimestre do ano que fornece informações sobre estatísticas, domínios, entre outros. Nesta secção serão abordados alguns dados, de 2021, para enquadrar a severidade do problema.

No primeiro trimestre [7], verificou-se que nos mês de janeiro foi registado um valor de 245,771 ataques, um valor nunca antes registado. No mês de fevereiro e março houve uma diminuição dos números, contudo março sofreu mais de 200 mil ataques. Segundo o relatório, as indústrias mais afetadas são as financeiras, que representam cerca de 24.9% dos ataques, seguido das redes sociais, com uma percentagem de 23.6% e e-mails com 19.6%. Em relação à tentativa de ataques, os atacantes utilizaram várias formas de chegar aos utilizadores. 54% são enviados cartões de presente para chamar atenção ao utilizador, os restantes 46% são e-mails relacionados com transferências bancárias, pedidos financeiros e desvio das folhas de pagamentos. Foi feita a referência dos domínios utilizados pelos atacantes no tema dos e-mails. Verificou-se que Namecheap representa cerca de 46.3% na

totalidade dos domínios e-mail, seguido do Public Domain Registry (PDR) com 26.7%, entre outros. É feita a referência, também, ao aumento de sites que usam encriptação Hyper Text Transfer Protocol Secure (HTTPS). Apesar de no primeiro trimestre ter havido uma descida na percentagem, ainda se registou que cerca de 83% dos sites *phishing* usam HTTPS. Por último, foi analisado o Top-level domain (TLD). Num estudo de 3054 Uniform Resource Locator (URL), *phishing*, verificou-se que o .COM esteve associado a 1535 URL, seguido do .UK com 77 e do .ORG com 55%

No segundo trimestre [8] verificou-se um máximo de 222,127 ataques, em junho, seguido de abril com 204,050 e maio abaixo dos 200,000. Mais uma vez, a indústria financeira foi a mais afetada com 29.2% dos ataques, com um aumento de 4.3% em relação ao primeiro trimestre, seguido das redes sociais, mas com uma diferença dos serviços de pagamento terem registado uma maior ocorrência a ataques com cerca de 12.2%. Verificou-se um aumento significativo dos ataques *phishing* no tema das criptomoedas com cerca de 7.5% dos ataques direcionadas ás industrias. Novamente verificado que os domínios dominantes nas URL *phishing* são o NameCheap que apresenta 29.6% e o PDR com cerca de 19.5%. Foram analisados 2447, URL *phishing*, para avaliar o TLD mais utilizado e verificou-se, mais uma vez, o .COM com 767 URL, seguido do .XYZ com 42 e .UK com 41, entre outros. Houve uma descida no uso de encriptação HTTPS de 1% (82%)

No terceiro trimestre [9] verificou-se um recorde máximo de ataques *phishing*. Mais de 260,642 ataques registados. Neste trimestre, houve uma alteração nos alvos escolhidos pelos atacantes. Foram registados 29.1% dos ataques a serviços Webmail e SAAS(Software-as-a-service), cerca de 17.8% a instituições financeiras e 13.1% a serviços de pagamento. Verificou-se que os atacantes mantém o .COM como TLD prioritário. Contudo, os atacantes variam entre outros como é o caso do .tk e .gq que seguem no TLD mais usados neste trimestre.

É possível concluir, com os dados apresentados ao longo dos anos, que os ataques tem tendência crescente e não mostram sinais de diminuição. Existe uma enorme necessidade da existência de mecanismos de detenção de e-mails *phishing* para evitar o número elevado dos ataques.

2.2 Piores prejuízos causados pelo *phishing*

Os ataques *phishing* podem causar prejuízos irreais às empresas e/ou utilizadores.

Por exemplo, um simples e-mail fez a empresa FACC perder mais de 50 milhões de euros. A empresa foi alvo de um ataque Whaling *phishing*, onde o autor malicioso se fez passar por um CEO da empresa que pediu uma transferência para uma conta à parte para a "aquisição de um projeto"[10].

Usando o mesmo método, whaling *phishing*, o Banco Crelan Bak sofreu uma perda de 75.6 milhões de euros. O autor malicioso fez-se passar pelo responsável do banco e, novamente, enviou um e-mail a um colaborador para transferir a quantia para uma conta [11].

A Sony Pictures também sofreu prejuízos, cerca de 80 milhões de euros, onde os autores maliciosos enviaram vários e-mails *phishing* aos executivos da Sony para verificar e-mails onde, automaticamente, os reenviava para web sites onde roubavam as credenciais dos utilizadores. Com esses dados, eles foram capazes de implementar um malware que fez com que terabytes de dados da empresa fossem roubados e apagados dos seus sistemas, exigindo que atendessem aos seus pedidos para que os dados não fossem vazados para fora [12].

O Facebook e a Google também sofreram por causa dos ataques *phishing*. Ambos sofreram prejuízos que superam os 90 milhões de euros. O atacante verificou que ambas as empresas utilizavam o mesmo fornecedor de infraestruturas. Fazendo-se passar pelo fornecedor, o atacante enviava faturas milionárias que pareciam ter sido assinadas pelos executivos de ambas as organizações [13].

O ser humano é suscetível a erros, e os atores maliciosos estão sempre a aproveitar esses erros para proveito próprio. Nem sempre a melhor abordagem são os avisos e as chamadas de atenção, a funcionários, para melhorar as capacidades de deteção de ataques *phishing*. É necessário, juntamente com formação dos funcionários, mecanismos que ajudem as empresas a não serem vitimas de fraude, devido aos diferentes tipos de *phishing*, para evitar a perda de dados sensíveis de cliente e capital.

2.3 Soluções anti-*phishing*

Com o grande crescimento da prática do *phishing*, são necessários mecanismos anti-*phishing* que ajudem o utilizador, até o mais experiente, a evitar situações que possam comprometer os seus dados pessoais.

No seguinte artigo [14], os autores propuseram uma técnica que detetava páginas web suspeitas baseado na consistência literal e conceptual entre o URL e o conteúdo web. Este projeto consiste na fase de pré-filtragem que se foca na identidade da página com o nome do domínio associado para filtrar os URL suspeitos dos não suspeitos e na fase de classificação que examina a consistência da página com o conteúdo da mesma. No final usando os diferentes algoritmos utilizados, Randomness of the URL (RU), Ratio of the found domain token (RDT) e Position of the domain token (CPO), conseguiram uma precisão de 98% na deteção de URL *phishing*.

Em [15] os autores desenvolveram uma plataforma que, de forma independente ou através de uma extensão, funciona como uma linha de defesa contra ataques *phishing*. A plataforma regista o domínio em que utilizador se regista e apenas é ativado quando, após monitorizada a URL, a extensão verifica a autenticidade da página, alertando assim o utilizador de um potência ataque *phishing* na página acedida.

No artigo [16] os autores disponibilizam um estudo sobre técnicas de anti-*phishing* comparando dois tipos de detenção de *phishing*, a consciência do utilizador e mecanismos automáticos de deteção de *phishing*. Ao longo do relatório são abordadas formas de prevenir o *phishing*, mecanismos de detenção de *phishing* e, no final, estabelecem a comparação de ambas as categorias com o objetivo, características, vantagens e desvantagens.

No seguimento deste artigo [17], a autora apresenta uma visão geral de vários ataques de *phishing* e técnicas de proteção. Numa fase inicial abordou 11 tipos de *phishing*, tais como Web Trojans, Key-loggers e Screen-Loggers, Deceptive *phishing*, Session Hijacking, entre outros. Numa segunda parte elabora um estado da arte onde retrata vários artigos baseados em técnicas anti-*phishing*.

No seguinte artigo [18], os autores focam-se num mecanismo de análise ao URL como forma de detetar atividade *phishing*, que eles denominam de Enhanced Malicious URLs Detection usando algoritmos de classificação, o Naïve Bayes (NB) e o SVM. Este meca-

nismo proposto pelos autores analisa um conjunto de recursos, que eles denominam de heurística, tais como a idade do domínio, dígitos em hexadecimal, tamanho do URL, caracteres especiais, entre outros, onde o algoritmo devolvia 0 ou 1 a cada resposta. No final submeteram o resultado aos 2 algoritmos de ML, e compararam o mecanismo deles a um mecanismo já existente. Verificaram que, neste estudo, o algoritmo SVM, aplicado com o mecanismo proposto, obtém uma melhor resposta que com o algoritmo NB.

No próximo artigo [19], os autores propõem um conjunto de técnicas anti-*phishing* que se destinam à prática da engenharia social, onde identificam um conjunto de ataques e como se proteger dos mesmos. O artigo começa com uma pequena introdução do *phishing* e que tipos de *phishing* existem. De seguida, descreve como é o processo de ataques *phishing* e o processo do ciclo de vida dos e-mails. Por fim, apresenta um conjunto de técnicas que incluí: Verificação do código-fonte; comparação de páginas legitimas com páginas *phishing*; Aplicação de regras (White e Blacklist); entre outros.

Focando o tema na deteção de *phishing* de web sites, os autores [20] elaboraram um artigo para ajudar o utilizador final a examinar web sites e, verificar se eles são maliciosos ou não, sem a necessidade de mecanismos anti-*phishing*. Eles começam por numerar caraterísticas que os web sites *phishing* contém, como por exemplo: o uso do IP como domínio; a não utilização de certificados SSL; identificação de erros; entre outros. No final, apresentam um conjunto de passos, para o utilizador usar como guia, para detetar se o web site é *phishing* ou não e apresentam a análise, com os passos que eles sugerem, da quantidade de web sites que apresentam essas caraterísticas.

Os autores no seguinte artigo [21] estudaram, analisaram e classificaram as melhores estratégias, na área de *phishing* de web site, e organizaram um conjunto de vantagens e desvantagens dessa estratégias. Os autores começam com uma introdução sobre ataques *phishing* na web, o cenário de um ataque, táticas, estatísticas, soluções anti-*phishing* e esquemas de prevenção. No final, agregaram um conjunto de esquemas de detenção de *phishing* baseado em motores de busca, ML, *phishing* Blacklist e Whitelist, semelhança visual, DNS e URL. Concluíram que esquemas de detenção de *phishing* tem melhor resultado do que prevenção de *phishing* e soluções de treino de utilizadores porque não requer mudanças na plataforma de autenticação e não depende da habilidade do utilizador para detetar *phishing*.

No seguinte artigo [22] foi realizada uma pesquisa sobre vários ataques *phishing*, os problemas que advém e soluções para o problema relacionado com esses ataques. Os autores começam com uma introdução sobre o tema "Internet"e os autores maliciosos que se aproveitam para realizar ataques *phishing*. Depois, descrevem os tipos de e-mail (Legítimos, spam e *phishing*) e tipos de ataques (Malware, Man-in-the-middle, entre outros). No final, fazem uma visão geral sobre deteção, técnicas e ferramentas para evitar os ataques *phishing*, onde sugerem Plugins Anti-*phishing*, filtros de spam, prevenção para softwares que contém malware, entre outros. No final, relatam algumas soluções e dicas para prevenir que os ataques sejam bem sucedidos como por exemplo: usar um browser seguro (e.g Brave, Tor, entre outros), suspeitar dos web sites, cuidado ao responder a e-mails e ignorar frases como "Promoções fantásticas", "Urgente", entre outras.

2.4 Machine Learning e anti-*phishing*

Em [23] os autores apresentam uma proposta que consiste na análise do texto de um e-mail usando Processamento de Linguagem Natural (PLN) que analisa cada frase e identifica relações entre significantes de palavras importantes para determinar se a frase é uma questão ou um comando. Para isso desenvolveram um script que processa linha a linha, do texto, e devolve verdadeiro se o documento contém ataques de engenharia social. O algoritmo é dividido em fases de análise: Verifica uma questão e/ou comando malicioso, algum tipo de saudação genérica e também uma ferramenta chamada Netcraft Anti-*phishing* que verifica a integridade e validade de uma URL. Essa questão e/ou comando são extraídos, avaliados e comparados com uma lista negra, tendo o sistema sido criado recorrendo a um modelo de ML treinado com um conjunto de e-mails benignos e malignos. Além do mecanismo do autor, foi usado um outro (Netcraft) para comparação. Num teste realizado com 5009 e-mails *phishing* e 5000 e-mails benignos verificaram resultados de 4545 verdadeiros positivos, 239 falsos positivos e 464 falsos negativos com uma precisão de 95%.

Em [24] os autores propuseram uma estrutura para análise sentimental de e-mails que usa um esquema híbrido de algoritmos combinado com K-means cluster e suporte com uma máquina de classificação vetorial, usando um *dataset* constituído por 200 mensagens de e-mails. A estrutura do estudo é composta por: extração de dados mais relevantes; pré-

processamento, que consiste na recolha de dados relevantes no e-mail; extração de recursos; Sentimental Lexicon e classificação sentimental, que consiste em aplicar um rótulo para os dados de modo a serem fornecidos a algoritmos de ML para serem sujeitos a testes. No final, provaram que o método K-means junto com o método Support Vector Machine SVM apresentava uma melhor precisão com 97.7%, comparada com os métodos SentiWordNet, Polarity, LOR, DTC e OneR.

No artigo seguinte [25] os autores usam abordagens como K-means cluster, fuzzy C-means clustering e Neural network para extrair recursos de um e-mail para que possa ser usado para pessoas na área forense. O objetivo era verificar qual dos algoritmos era mais eficiente na classificação da análise sentimental de cada e-mail (positivo, negativo e neutro). A estrutura consiste: na recolha de e-mails; um pré-processamento para remover palavras indesejadas e a extração de palavras-chave para extrair o sentimento que estão escondidos nessas palavras. Neste estudo foi usado um *dataset* constituído por 1412 e-mails, em que 60% constituíram um *dataset* para treino, 20% para validação e 20% para teste. No final, os autores conseguiram comprovar que o mecanismo de Neural Network era mais eficiente em reconhecer o tipo de e-mail com uma taxa de 97.91%, seguido do K-means com 80.22% e, por último, o funny C-means com uma taxa de 79.65%, de média.

No próximo artigo [26] os autores usam 2 tipos de algoritmos de ML: NB e o K-Nearest Neighbor (KNN) para calcular a exatidão e precisão de avaliações da opinião pública de filmes e de experiências em hotéis. A estrutura que os autores utilizaram passou por: recolher de dados; preparaçar dos dados que recolheram; detetar a análise sentimental desses dados e a sua devida classificação. Não foram claros na forma como executaram cada passo. Com um *dataset* constituído por 5000 avaliações negativas e 5000 positivas onde verificaram que nas avaliações dos filmes o mecanismo NB tem uma melhor precisão com uma precisão superior a 80% comparado com a melhor precisão da K-NN de 63.31%. Mas, em contrapartida, o mecanismo K-NN apresenta melhor precisão na avaliação dos hotéis com uma taxa de 70%.

Neste artigo [27] os autores usaram 2 algoritmos de classificação, NB e o LOR para determinar a análise sentimental de tweets provenientes do twitter. A metodologia usada pelos autores consiste no processamento da informação, como a retirada de URL, Emoji's, pontuação, pontos de interrogação e as letras grandes foram todas convertidas para minúsculas,

coletar os recursos chave dos tweets, na extração de recursos usando o mecanismo Term Frequency Inverse Document Frequency (TF-IDF), balanceamento de dados e classificação sentimental dos tweets. Os autores não indicam a quantidade de dados usados no estudo, mas mostraram que o método de seleção LOR tem uma maior precisão que o outro mecanismo com uma percentagem de 88.4%.

Usando técnicas de Processamento de Linguagem Natural [28], os autores desenvolveram um sistema que dispõe ao utilizador uma recomendação clara de qualquer produto computacional usando o mecanismo de SVM usando diferentes avaliações pelos utilizadores. O sistema criado entra com um produto, copiando o URL do produto. Com esse URL o sistema recolhe a informação do site, devidamente processada, e guarda diretamente no sistema e classifica quanto à sua análise sentimental com os algoritmos de ML. Com um total de 2000 avaliações eles obtiveram uma precisão de 70.01% na recomendação de um produto previamente descrito.

Neste estudo [29] foi criada uma nova abordagem de ML que prevê o desempenho de cada estudante na vida académica usando os algoritmos XGboost, KNN e o SVM como modelos de previsão com o objetivo de determinar, a partir da personalidade e aptidão, o desempenho do estudante e descobrir quais os problemas relacionados com o estudante. Os dados utilizados contém 15 atributos de 1500 estudantes. O modelo conta com um pré-processamento da informação em análise que consiste no tratamento de valores em falta, valores inconsistentes, nomeadamente a troca de informação (Idade pelo sexo) e valores duplicados e a seleção e extração de recursos dos dados. Após testes, verificaram que o mecanismo XGboost era o mais apropriado para o estudo em questão seguido do mecanismo KNN e do XGboost. Não foi possível verificar qual a percentagem correta pois apenas é demonstrado num gráfico.

Usando algoritmos de previsão, os autores [30] criaram uma solução que previa, relativamente à performance de cada jogador, um conjunto de jogadores adequados a uma equipa de Cricket usando os classificadores NB, RF, o SVM e o DTC. Neste estudo, os autores consideraram vários status de cada jogador na criação do *dataset*, usando dados de jogos jogados entre 14 de janeiro de 2005 até 10 de Julho de 2017. Concluíram que, neste estudo, o classificador RF era o mais indicado para avaliar a qualidade dos jogadores com uma previsão superior a 90%.

Apesar de vários estudos efetuados na área da análise sentimental de textos com o recurso aos algoritmos de ML são poucos os que apresentam uma análise gramatical ao texto dos e-mails. São vários os e-mails de *phishing* que tentam chegar ao utilizador final sem qualquer rigor na escrita, com erros ortográficos, entre outros, que, com a análise correta, é possível diferenciar os diferentes tipos de e-mails. Com isto, queremos dar a conhecer uma nova perspetiva de análise no mundo da engenharia social e, assim, criar uma nova porta no combate à criminalidade cibernética.

Para a escolha dos algoritmos de ML, e segundo investigação do estudo do estado da arte, existem várias abordagens, conforme é possível ver na tabela 2.1. A escolha destes algoritmos deveu-se ao facto dos resultados obtidos serem de alta taxa de acerto na análise de classificação binária.

Estudo	Abordagem
Machine Learning	
[24]	Support Vector Machine
[26]	Naïve Bayes
[27]	Naïve Bayes e Logistic Regression
[28]	Support Vector Machine
[30]	Naïve Bayes, DecisionTreeClassifiers, Random Forest e Support Vector Machine

Tabela 2.1: Seleção dos algoritmos de ML

Capítulo 3

Metodologia de Trabalho

Neste capítulo é apresentada a metodologia do estudo efetuado. Começam-se por apresentar as várias etapas que o trabalho seguiu e por etapa são descritas as ações levadas a cabo. O estudo subjacente à deteção de e-mails de *phishing*, baseado em reconhecimento de entidades, divide-se em 3 grandes grupos. Os grupos são ilustrados na Figura 3.1 e representados por um esquema de cores: verde, azul e amarelo.

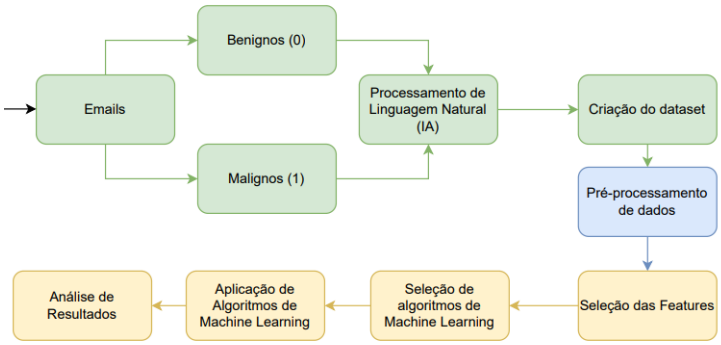


Figura 3.1: Metodologia de trabalho

3.1 Agregação de e-mails, extração e tratamento da informação do e-mail

A cor verde na Figura 3.1 diz respeito ao seguinte conjunto de etapas:

- E-mails;
- Extração de dados a partir do conteúdo do e-mail usando PLN;

- Criação do *dataset*;
- Análise exploratória e tratamento do *dataset*;

A etapa "E-mails", resume-se à recolha de um conjunto de e-mails, pré-definidos como sendo benignos, as quais foi atribuído o valor de classe 0 (zero) e malignos aos quais foi atribuído o valor de classe 1 (um). Estes e-mails são a base do *dataset* a criar. Nesta fase houve a necessidade de efetuar um tratamento inicial, fazendo uma limpeza tanto ao nível dos cabeçalhos como do corpo e a remoção dos ficheiros em anexo. O objetivo desta limpeza é ficar apenas com o corpo de cada e-mail. Tal prende-se com o tipo de abordagem a aplicar para a deteção de *phishing* (baseado no corpo do e-mail através do reconhecimento de identidades).

Na segunda etapa, os e-mails tratados foram submetidos a processos de extração de dados. Para tal extração, usaram-se métodos baseados em PLN. Estes métodos tem a capacidade de identificar texto de uma forma idêntica aos humanos, sendo frequentemente usados para a deteção de SPAM, reconhecimento de entidades (NER - Named-Entity Recognition), usados em tradutores automáticos, usados para análise de sentimentos, entre outros. Estes métodos estão disponíveis para várias linguagens de programação. Por exemplo o Natural Language Toolkit (NLTK) é uma biblioteca Python que fornece módulos de processamento de texto, classificação, entre outros. O MALLET é um componente de software em JAVA que fornece classificação de documentos, extração de informação, entre outros. Neste projeto foi utilizado o módulo spaCy, do Python. Trata-se de um módulo bastante popular, gratuito, de fácil utilização e com alto desempenho. Tem como aspeto menos positivo o facto de apenas suportar conteúdo em inglês.

O terceiro ponto diz respeito à criação do *dataset*. O *dataset* é composto pela informação extraída do corpo do e-mail. Entre os parâmetros extraídos constam os dados relativos ao PLN, o número de erros, o tamanho do e-mail, entre outros.

No quarto é relativo ao pré-processamento dos dados. Este processo engloba o tratamento de valores em falta, o tratamento de valores extremos, a limpeza de dados não enquadrados no texto de e-mail como por exemplo o HTML existente no corpo do e-mail, entre outros. O balanceamento do *dataset* é também considerado nesta fase. Existem vários algoritmos de balanceamento [31]: o Random Under-Sampling (RUS) remove ob-

servações da classe maioritária; o Random Over-Sampling (ROS) adiciona observações à classe minoritária; o Random Under-Sampling with imblearn (RUSI) escolhe aleatoriamente um conjunto de dados de uma classe selecionada; o Random Over-Sampling with imblearn (RoSI) gera novos dados à classe minoritária; o Under-sampling: Tomek links (USTL) remove dados da classe maioritária; o Synthetic Minority Oversampling Technique (SMOTE) adiciona dados sintáticos à classe minoritária. O mecanismo usado neste estudo será o ROS, pois será mais propicio ao algoritmo de ML ter mais dados para analisar que serem eliminados dados. Esta técnica irá acrescentar dados à classe Malignos de forma a balancear o dataset.

3.2 Seleção de Features

A cor azul na Figura 3.1 diz respeito à fase denominada de seleção de *features*. É nesta fase que se verificam quais as *features* mais significativas a considerar para o treino dos algoritmos. Existem várias técnicas que podem ser usadas para levar a cabo este procedimento. A amostragem de atributos é uma técnica que consiste na seleção de atributos significativos para uma determinada previsão, por exemplo, prever a prestação de um determinado atleta onde a idade, bpm (Batimentos por minuto), peso e altura são atributos mais significativos que a religião, gosto musical, entre outros. O registo de amostragem consiste na remoção de valores em falta e menos significativos, entre outros. Para o estudo aqui apresentados foram considerados dois algoritmos de seleção de *features*: o SelectKBest (SKB) e o RF, com recurso à função feature_importance. Estes algoritmos devolvem as *features* identificadas como as mais significativas para a classificação.

3.3 Seleção e aplicação de algoritmos de ML e Análise de Resultados

A cor amarela está representada uma fase que engloba as seguintes etapas:

- Seleção de algoritmos de ML;
- Aplicação de algoritmos de ML;

- Análise de resultados.

Na primeira etapa da fase de seleção de algoritmos foi abordado quais os algoritmos de ML a utilizar. Existe uma variedade de algoritmos [32]: Linear Regression (LIR) método utilizado para modelar a relação passada entre variáveis de entrada independentes e variáveis de saída dependentes; LOR usado em tarefas de classificação; DTC que consiste na divisão de recursos de dados em ramos nos nós de decisão até que uma saída de decisão final seja feita; NB que aplica o teorema de Bayes que permite que a probabilidade de um evento seja calculada com base no conhecimento dos fatores que podem afetar esse evento; RF que melhora a precisão de uma decisão simples gerando várias decisões simples e selecionando a maioria dos votos para prever o resultado; SVM que é usada para classificações e, também, para realizar regressões; Neural Network (NN) que utilizada neurónios artificiais, formados em 3 camadas, que podem ser usados para classificar dados ou encontrar relações entre variáveis; KNN que coloca os dados em vários grupos, cada um contendo características semelhantes, determinado pelo próprio modelo; GNB um modelo genérico do KNN com uma melhoria nos *clusters*; Gradient Boosted Decision (GBM) que utiliza técnicas de classificação ou regressão onde gera várias grupos, onde cada grupo se concentra em corrigir os erros provenientes do grupo anterior. MLPC que depende de uma NN para realizar uma tarefa de classificação. Após análise dos algoritmos optou-se pela utilização de seis algoritmos. Estes algoritmos são os que melhor se enquadram na área em estudo, considerando o estado da arte, e são eles: GNB; LOR; DTC; RF; SVM e MLPC.

Na segunda etapa, desta última fase, o objetivo é aplicar os algoritmos de ML, anteriormente escolhidos, tendo em consideração as *features* mais significativas previamente identificadas. De forma a entender a melhor quantidade de *features* para submeter nos algoritmos de ML, e o impacto que advém da seleção das mesmas, foram adotados 3 grupos de *features*: um grupo com 15 *features*, um grupo com 10 *features* e um com 5 *features*. Uma vez que o RF devolve *features* diferentes a cada iteração, optou-se ainda por fazer diversas experiências, usando cinco conjuntos de *features* diferentes, devolvidas pelo método, para avaliar o impacto que tal poderá ter nos resultados. Em relação ao SKB, este devolve sempre a mesma lista de *features*.

Na terceira etapa, desta última fase, e último ponto, são analisados os resultados devolvidos pelos algoritmos de ML. Existem várias métricas [33] que podem ser usadas para analise de resultados, nomeadamente: Accuracy é uma medida de desempenho que faz a relação entre a observação correta prevista e o total de observações; Precision que faz a relação entre as observações positivas previstas corretamente para o total de observações positivas previstas; Recall é a relação entre as observações positivas corretamente previstas para todas as observações na classe atual; F1-score faz a média, ponderada, entre a Precision e o Recall, que leva em consideração tanto os falsos positivos como os falsos negativos. Para efeitos de análise e comparação optou-se por usar a Accuracy obtida através de Cross-Validation Score.

No próximo capítulo é apresentado a forma como foi criado o *dataset* para, depois, ser submetido a um mecanismo de PLN.

Capítulo 4

Criação do *dataset*

Neste capítulo são apresentados os passos subjacentes à criação do *dataset*. A criação do mesmo, bem como o seu tratamento, é necessário para que os algoritmos de ML treinem e maximizem os resultados de eficácia. Das pesquisas realizadas não se encontrou nenhum conjunto de dados, relacionados com e-mails *phishing*, que nos permita analisar e enriquecer. Por isso foi necessário criar um conjunto de dados, de raiz, e identificar parâmetros, ou *features*, que possam ser relevantes para a análise de dados via algoritmos de ML. O foco da análise é o conteúdo do e-mail e nesse sentido os parâmetros relacionam-se sobretudo com aspetos linguísticos. Uma vez obtido um conjunto de dados inicial, e tal como se apresenta neste capítulo, é necessário efetuar uma análise exploratório para compreender os dados, fazer a limpeza/tratamento dos dados obtendo no final um *dataset* pronto para análise.

4.1 Obtenção de e-mails de *phishing* e não *phishing*

À falta de *datasets* disponíveis foi necessário criar um de raiz. Para o efeito foi necessário encontrar um conjunto de e-mails que, à priori, sejam validados como *phishing* (Malignos) e não *phishing* (Benignos). Estes e-mails foram obtidos através de um investigador que publicou o seu trabalho de investigação na International Workshop on Security and Privacy Analytics (IWSPA) [34] e que gentilmente cede os dados. A partir dos dados fornecidos conseguiram-se 504 e-mails Malignos e 4082 e-mails Benignos. Foram ainda recolhidos 794 e-mails Malignos de um *dataset* disponível no Github [35], perfazendo um

total de 1298 e-mails malignos.

Dado que os e-mail obtidos tinham formatos diferentes houve a necessidade de os converter num único formato facilitando assim o seu processamento. Todos os e-mails foram lidos e convertidos para o formato CSV tornando assim a informação organizada de forma tabular. Este CSV será depois lido através das funções do módulo Pandas, do Python, facilitando a manipulação e analise dos dados [36]. A Figura 4.1 esquematiza a transformação geral efetuada: todos os ficheiros de e-mail são colocados num diretório e há um script que lê esses e-mails, prepara-os e converte-os para um formato CSV. Para a transformação foi usado o módulo "csv" do Python [37].

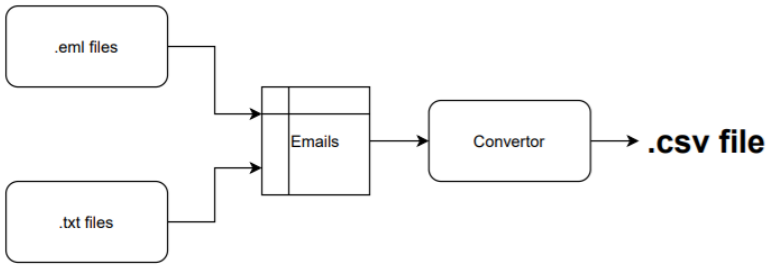


Figura 4.1: Conversão dos ficheiros .txt e .eml para csv

4.2 Preparação dos dados

Antes da leitura dos e-mails e extração dos parâmetros para o formato CSV, houve a necessidade de tratar os e-mails. O tratamento aplicado é apresentado ao longo das próximas sub secções.

4.2.1 Tradução dos e-mails

Verificou-se que o idioma usado dos e-mails não era sempre o mesmo. Tal como referido no Capitulo 3, será adotado um mecanismo PLN para derivar *features* a partir do corpo do e-mail. Este módulo apenas utiliza o idioma inglês e, por isso, houve a necessidade de traduzir todos os e-mails para esta língua. Existe um conjunto de ferramentas que permitem esta tradução [38]. Para o efeito foi usado um módulo Python que tira partido da API do Google Translate [39]. A Figura 4.2 apresenta alguns dos idiomas detetados e o processo de tradução.

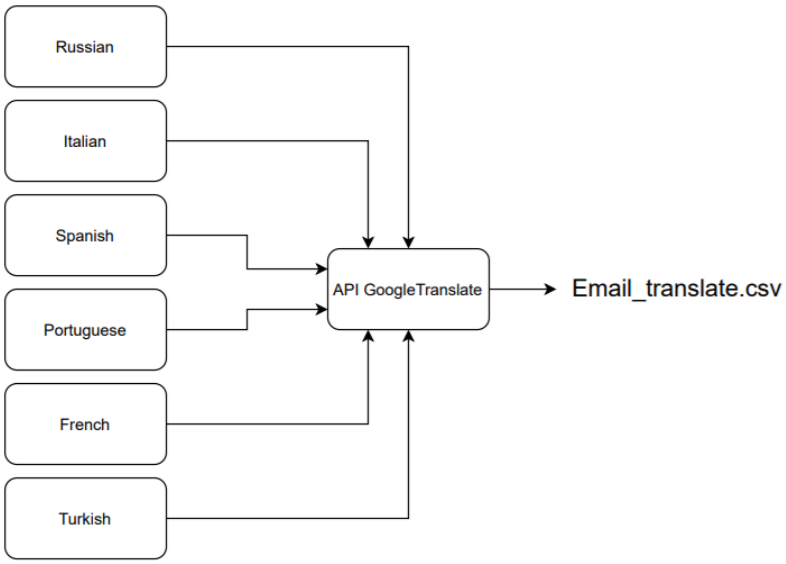


Figura 4.2: Tradução dos e-mails e armazenamento em formato CSV

4.2.2 Remoção dos Cabeçalhos

Considerando que o objetivo neste trabalho é a análise do corpo da mensagem, por forma a detetar se o e-mail é de *phishing* ou não, os dados extra do e-mail foram removidos. Para tal, foi criado um script que faz a extração do corpo da mensagem ignorando os cabeçalhos do e-mail com exceção do "subject". Este processo é ilustrado na Figura 4.3. O script tira partido do módulo "e-mail" do python [40], que permite retornar uma estrutura da mensagem e de onde, posteriormente, se pode retornar apenas o conteúdo relativo ao assunto e do corpo do e-mail [38].

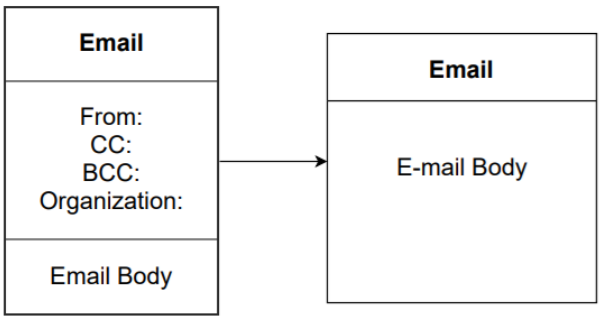


Figura 4.3: Remoção dos cabeçalhos do e-mail

Com o corpo da mensagem filtrado, os dados foram analisados e extraídos para formato CSV.

4.2.3 Caracteres indesejados

Outro tratamento realizado diz respeito à remoção de caracteres indesejados. Nos e-mails base verificou-se que:

- Os URL tinham sido removidos e substituídos pela palavra chave "link". Este valor pouco acrescenta à análise e por isso essa palavra chave foi removida;
- Alguns e-mails usavam o símbolo "=" para definir um parágrafo. Este símbolo foi retirado uma vez que não acrescenta valor à análise;
- Existiam e-mails com conteúdo HTML. Uma vez que a análise se baseia no conteúdo optou-se por retirar o HTML existente nos e-mails.

Com o tratamento dos e-mails base efetuado, passou-se à extração das *features* que, tal como referido, são compiladas num ficheiro com formato CSV.

4.3 Features e Named Entity Recognition

O Reconhecimento de entidades Nomeada (Named Entity Recognition (NER)) é uma das área de aplicação de PLN. Considerando que o objetivo é retirar do conteúdo do e-mail um conjunto de *features* e analisar se as mesmas permitem distinguir e-mails de *phishing* e não-*phishing*, tirou-se partido do NER para o efeito. No Python existe um módulo, denominado spaCy que disponibiliza funções de PLN, permitindo criar uma classe base à qual é acrescentada mais informação [41]. Deste modo, usando o módulo do python 'spaCy' [42] foi criado um script que permite recolher informação gramatical, pessoas, sítios, organizações, entre outros, das palavras provenientes no e-mail. A Figura 4.4 ilustra o processo de recolha desta informação.

As informações obtidas são denominadas de Part of Speech (POS). A POS é uma função gramatical que indica como uma determinada palavra é usada numa frase. Na tabela 4.3, são apresentados todos os POS devolvidos no conjunto de e-mails em análise.

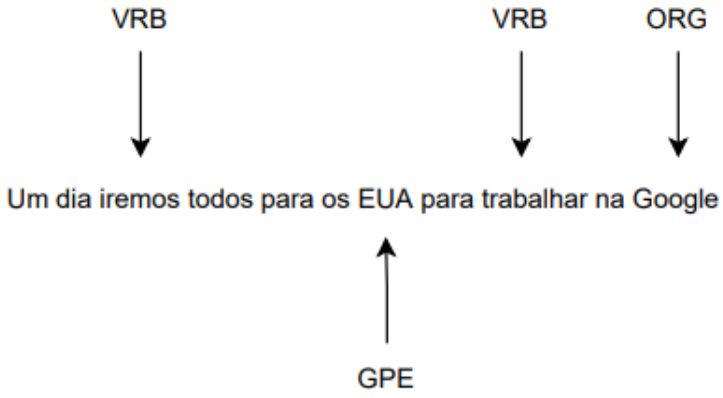


Figura 4.4: Aplicação do módulo spaCy

Tabela 4.1: NER - POS: Part Of Speech

POS	**Descrição**
_SP	none
ADD	e-mail
“	opening quotation mark
-LRB-	left round bracket
UH	interjection
CD	cardinal number
VBP	verb, non-3rd person singular present
-RRB-	right round bracket
WORK_OF_ART	titles of books, songs, etc.
IN	conjunction, subordinating or preposition
MD	verb, modal auxiliary
NNP	noun, proper singular
RP	adverb, particle
:	punctuation mark, colon or ellipsis
$	symbol, currency
Continua na próxima página	

Tabela 4.1 – NER

POS	Descrição
,	punctuation mark, comma
PRP$	pronoun, possessive
RB	adverb
CC	conjunction, coordinating
VBN	verb, past participle
TO	infinitival “to”
JJ	adjective
JJS	adjective, superlative
NN	noun, singular or mass
FW	foreign word
NNS	noun, plural
LS	list item marker
HYPH	punctuation mark, hyphen
WP	wh-pronoun, personal
VBG	verb, gerund or present participle
PRODUCT	objects, vehicles, foods, etc. (not services)
VBZ	verb, 3rd person singular present
MONEY	monetary values, including unit
DT	determiner
PERCENT	percentage, including “%”
VBD	verb, past tense
ORDINAL	first, “second”, etc.
XX	unknown
EVENT	named hurricanes, battles, wars, sports events, etc.
”	closing quotation mark
WDT	wh-determiner
POS	possessive ending

Continua na próxima página

Tabela 4.1 – NER

POS	Descrição
NORP	nationalities or religious or political groups
VB	verb, base form
PDT	predeterminer
PRP	pronoun, personal
RBR	adverb, comparative
NFP	superfluous punctuation
JJR	adjective, comparative
SYM	symbol
WRB	wh-adverb
NNPS	noun, proper plural
EX	existential there
.	punctuation mark, sentence closer
RBS	adverb, superlative
CARDINAL	Numerals that do not fall under another type
LAW	named documents made into laws.
ORG	companies, agencies, institutions, etc.
LOC	non-GPE locations, mountain ranges, bodies of water
DATE	absolute or relative dates or period
FAC	buildings, airports, highways, bridges, etc.
PERSON	people, including fictional
QUANTITY	measurements, as of weight or distance
TIME	times smaller than a day
AFX	affix
GPE	countries, cities, states
LANGUAGE	any named language
WP	wh-pronoun, possessive

Os e-mails, depois de tratados, foram analisados através do módulo spaCy, permitindo retirar as POS. Por cada POS é determinado o valor total, sendo esse mesmo valor registado no ficheiro em formato CSV.

Além dos campos devolvidos pelo módulo spaCy, o o *dataset* foi enriquecido com mais quatro *features*. As quatro *features* extra são:

- Size[S] - Tamanho do e-mail;
- NumberWords[NW] - Conjunto de palavras que constituem o e-mail;
- NumberErrors[NE] - Número de erros tipográficos que o e-mail contém. Foi usando o módulo spellChecker [43] para o efeito;
- SentimentalAnalysis[SA] - Análise sentimental sobre e-mail, usando o módulo textBlob retornando entre um sentimento positivo, negativo ou neutro [44].

Foi ainda necessário inserir uma coluna que diz respeito à variável dependente - "Classe". Esta variável faz a distinção entre os e-mails de *phishing* dos e-mails benignos (1 e 0 respetivamente).

É de referir que, com base no apresentado, o *dataset* final é constituído por 72 *features*.

4.4 Extração dos dados do e-mail para o CSV

Após efetuado o tratamento de dados, o e-mail foi submetido a processos de extração das *features* (e.g., número de erros, número de verbos, número de referências a valores monetários). O resultado da extração, é inicialmente colocado numa estrutura JSON, organizada e separada em chave valor, em que a chave identifica o nome da *feature* e o valor representa o respetivo valor para o e-mail em análise. A estrutura JSON relativa a cada e-mail é guardada num ficheiro cujo o nome corresponde ao valor hash sha256 do seu conteúdo. Este conteúdo foi guardado num formato binário (serializado), tirando partido do módulo de Python pickle [45]. Foi adotado este procedimento ao invés de armazenar o resultados diretamente no CSV, para que, antes de criar o CSV se pudesse ter a lista completa de *features* que resultaram da aplicação dos algoritmos de NER.

No final, cada ficheiro hash contendo a estrutura JSON serializado foi percorrido, desserializado, permitindo criar o ficheiro CSV que agregou todos os dados. Para facilitar

a análise de irregularidades ou valores abusivos, acrescentou-se uma coluna designada ID, que corresponde ao nome do ficheiro hash relativo ao e-mail. Assim, em caso de erro/anomalia nos dados torna-se possível descobrir qual é o e-mail origem.

4.5 Análise Exploratória

A análise exploratório foi feita com o objetivo de perceber o *dataset*. Esta análise permite verificar se o *dataset* se encontra balanceado em relação a cada classe (0 em relação a benignos e 1 em relação a Malignos). Permite também verificar a diferença gramatical utilizada em cada classe.

Facilita também uma análise sobre as *features* evidenciando o seu papel no *dataset* e potencial importância para a análise. Por exemplo, relativa a esta análise, e tal como em [46], verifica-se a existência de um maior número de ocorrências sobre os valores monetários, na tentativa de extorsão, número de erros, que é bastante comum em e-mails *phishing*, entre outros. Através da análise exploratória foi verificado que: Existe uma uma maior ocorrência no número de erros registado na classe maligna, conforme é possível verificar pela Figura 4.5;

Figura 4.5: Somatório da feature "NE" (número erros) em ambas as classes

Existe uma maior ocorrência de valores monetários em e-mails *phishing*, que nos e-mails benignos, conforme é possível verificar pela Figura 4.6;

Verificou-se ainda que os e-mails malignos tem tendência a ser maiores devido pelo facto de conter ficheiros em anexo (eventualmente maliciosos). A Figura 4.7 ilustra a diferença entre o tamanho total dos e-mails *phishing* e dos e-mails benignos.

Figura 4.6: Somatório da feature "MONEY" em ambas as classes

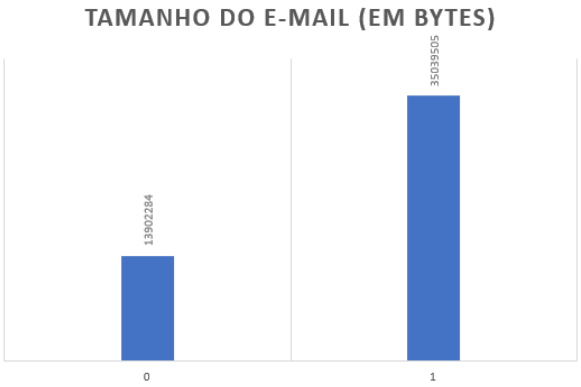


Figura 4.7: Somatório da feature "S" (*size*) em ambas as classes

Outra análise realizada consistiu em agrupar *features* em grupos para analisar o seu eventual peso por classe. Um dos grupos foi: Verbos; Pronomes; Advérbios; Pontuação e Determinantes. A análise permitiu verificar a tendência da utilização dos diferentes tipos de verbos e dois destacam-se no lado dos e-mails malignos: VBG e VBN. Os restantes, tem mais tendência a surgir em e-mails benignos. Esta análise é ilustrada na Figura 4.8.

No conjunto de *features* relativos aos pronomes, à exceção do PRP$, verifica-se que existe um maior número de pronomes a ser utilizados nos e-mails benignos (Figura 4.9).

Relativamente ao conjunto das *features* com pontuação verificou-se uma maior abundância de tipos de pontuação em e-mails de *phishing*. Contudo, verificou-se um maior número de pontuação final nos e-mails benignos. Tal é ilustrado na Figura 4.10.

Relativamente à análise sobre o uso de determinantes, é possível concluir que os e-mails benignos, em todas as categorias, são os e-mails que mais utilizam determinantes no corpo da mensagem (Figura 4.11).

Por fim, foram analisadas as referências a locais, datas, cidades, objetos, organizações, edifícios, entre outros. Esta análise é ilustrada na Figura 4.12.

Evidencia-se pela Figura 4.12 uma maior utilização de referência a organizações nos

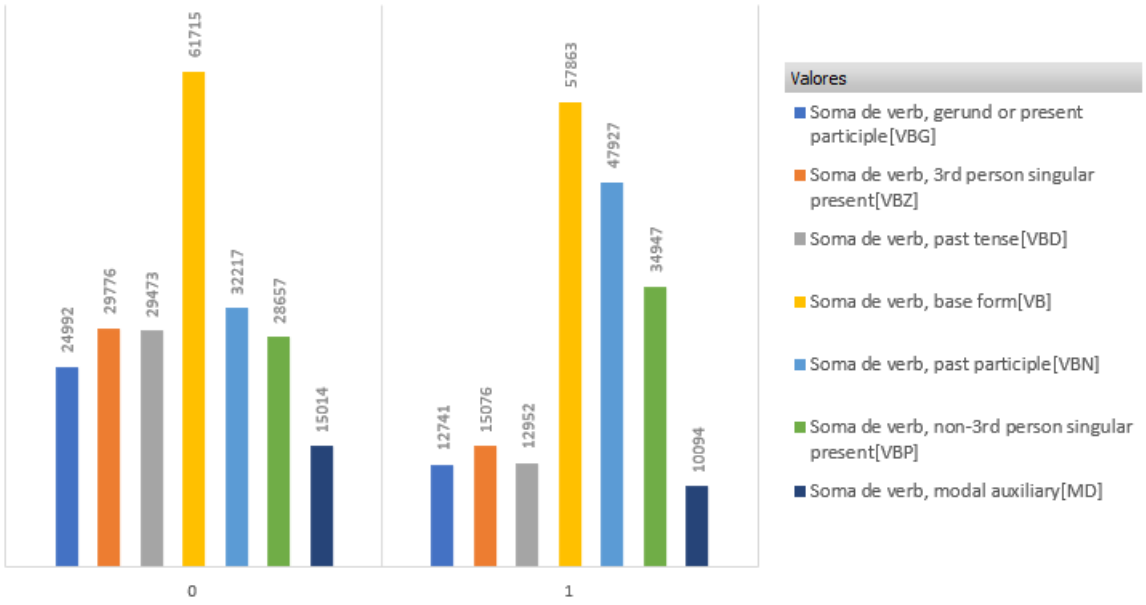


Figura 4.8: Somatório do conjunto das features relacionadas com verbos em ambas as classes

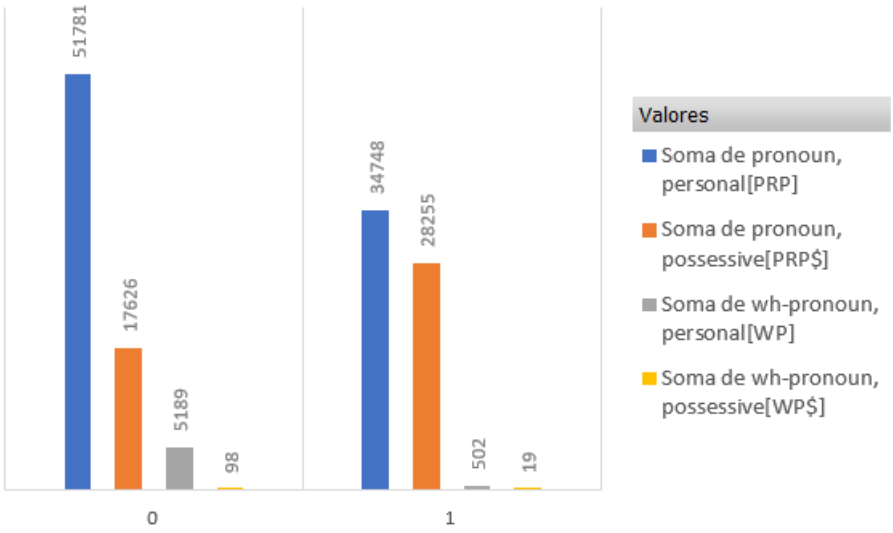


Figura 4.9: Somatório do conjunto das features relacionadas com pronomes em ambas as classes

e-mails de *phishing* e uma maior referência a nomes de pessoas nos e-mails benignos, o que faz sentido na medida em que normalmente os e-mails de *phishing* (a não ser os de Spear-*phishing*) não tratam as vítimas pelo seu nome.

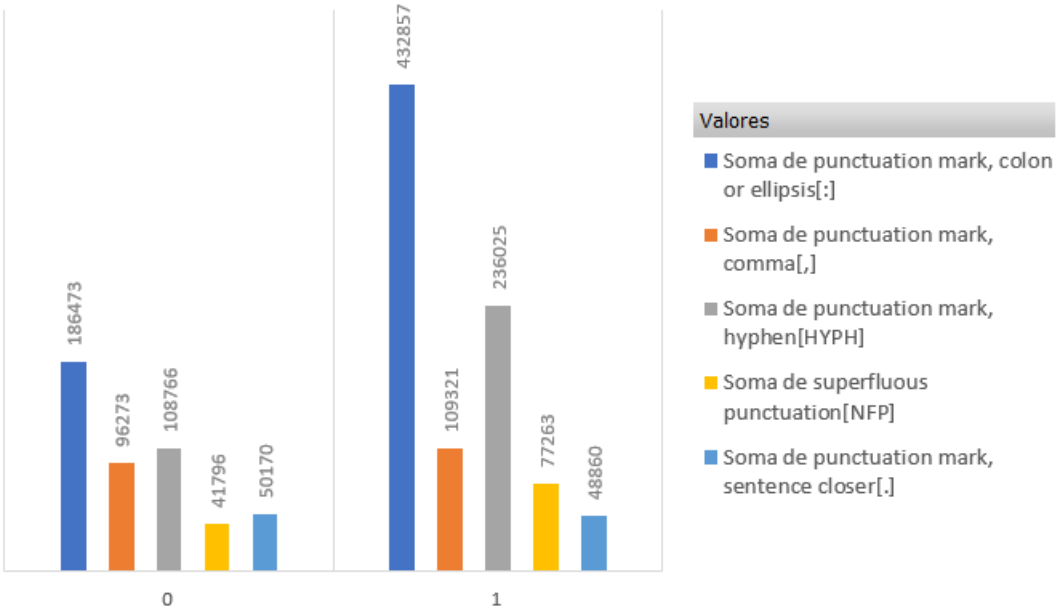


Figura 4.10: Somatório do conjunto das features relacionadas com a utilização de pontuação em ambas as classes

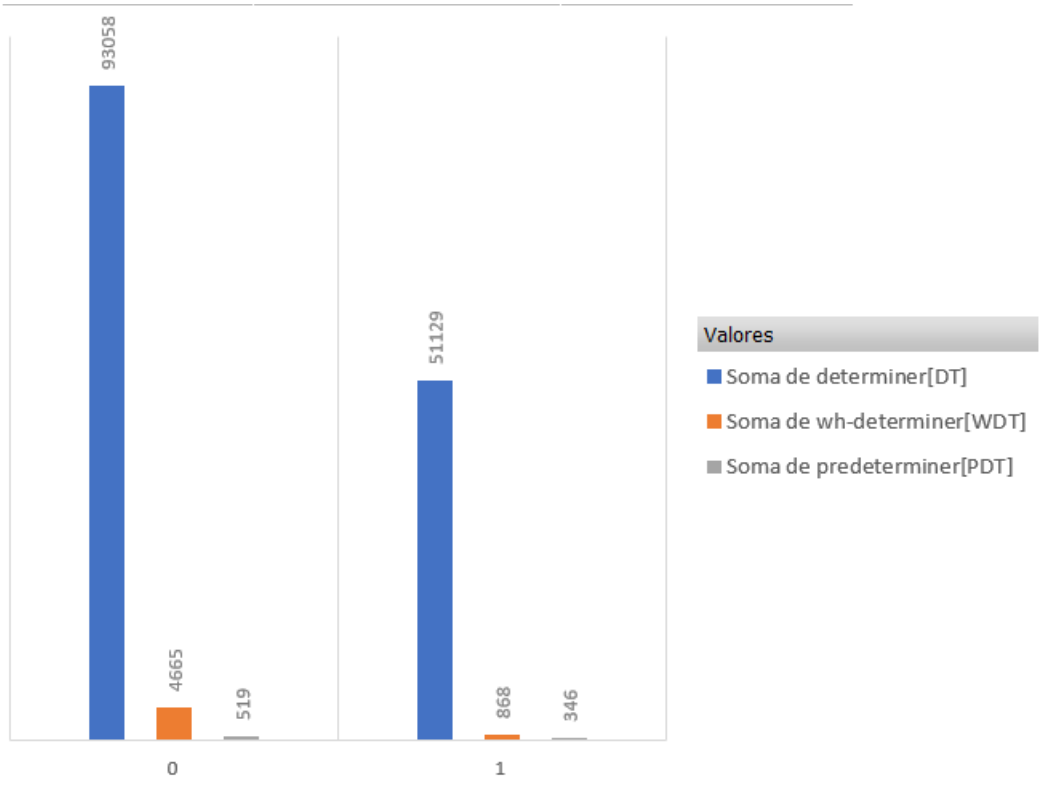


Figura 4.11: Somatório do conjunto das feature relacionadas com determinantes em ambas as classes

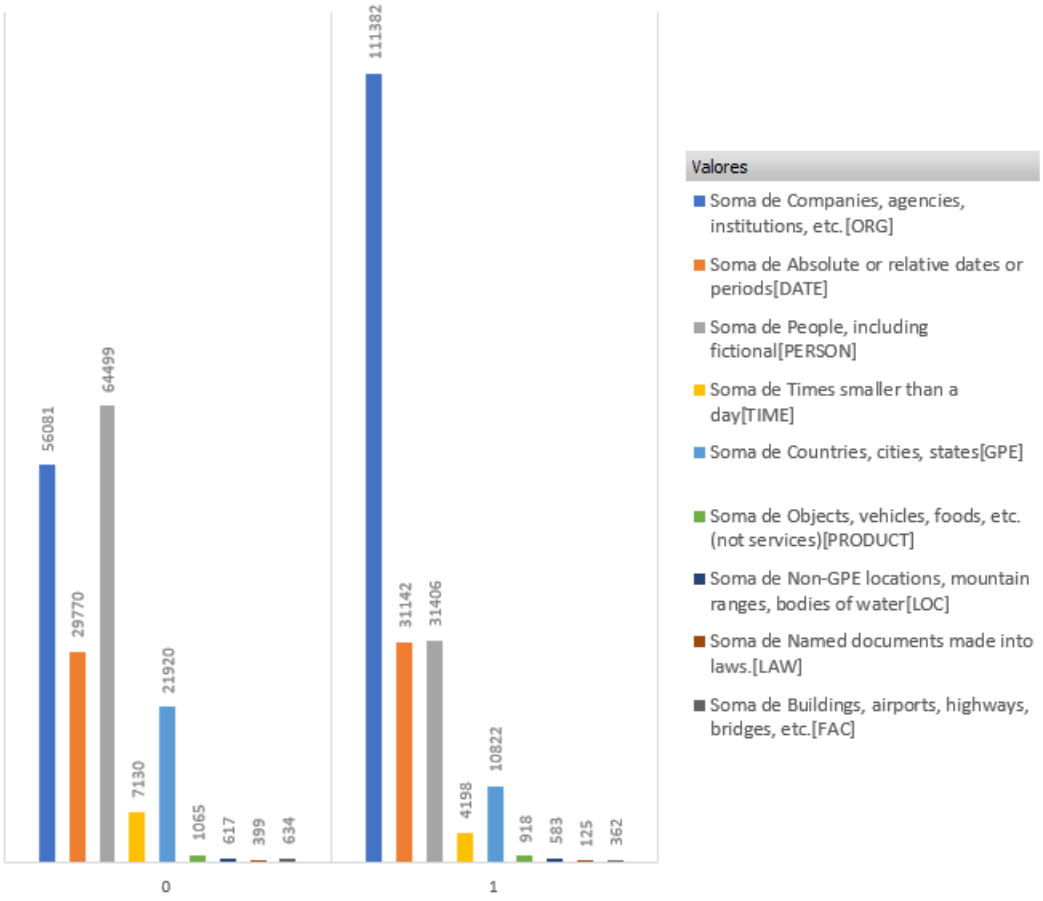


Figura 4.12: Somatório do conjunto das *features* em ambas as classes

4.6 Preparação do *dataset*

A análise exploratória teve por objetivo verificar se o *dataset* criado (a partir dos passos anteriormente descritos) continha valores nulos, se existiam valores extremos indesejáveis para a análise. Feita essa análise, o *dataset* foi preparado de forma a resolver algumas das limitações encontradas bem como para adicionar novas *features* relevantes para a análise e para tratar o balanceamento do *dataset*.

4.6.1 Novas features

Durante esta análise e numa primeira fase foi criada uma nova *feature*, denominada de "PercentageWords". Essa percentagem é calculada multiplicando o número de erros (Ne) com o número de caracteres (Nc) a dividir por 100% que dá o resultado, em percentagem, da relação entre caracteres e erros num determinado e-mail. Esta relação em percentagem é mais representativa do que o valor absoluto, na medida em que é considerado o tamanho do texto para pesar o número de erros.

Após a criação da nova *feature*, verificou-se que, nos e-mail benignos, o e-mail com maior percentagem de erros tinha cerca de 20.98% e a mais baixa apresentava um valor de 1.1%, conforme podemos ver na Tabela 4.2. Já em relação aos malignos, o e-mail com mais erros, apresentava uma percentagem de 50.77% de erros e a mais baixa tem uma percentagem de 0.79%, conforme podemos ver na Tabela 4.3. O número de erros em e-mails de *phishing* poderá ser maior do que e-mails legítimos. A análise efetuada e apresentada nas tabelas 4.2 e 4.3 visa analisar isso e como se pode verificar o número de erros em percentagem é efetivamente maior nos e-mails de *phishing*.

id	Percentagem(%)
res\b6e...	20.98
res\805...	20.96
res\31b...	20.95
...	...
res\a97...	1.40
res\a8e	1.35

id	Percentagem(%)
res\6dc	1.35

Tabela 4.2: Percentagem de erros nos e-mails benignos por e-mail

id	Percentagem(%)
res\ff8...	50.77
res\cd4...	50.63
res\837...	50.50
...	...
res\1b0...	2.56
res\39c	2.47
res\dba	0.79

Tabela 4.3: Percentagem de erros nos e-mails de *phishing* por e-mail

No seguimento da mesma análise, é apresentada, na Figura 4.13, os resultados, em percentagem, da quantidade de erros dos e-mails benignos e malignos. Através da figura é possível fazer uma correspondência entre a percentagem de erros ortográficos em relação ao número total de erros no e-mail. As barras azuis representam os e-mails benignos e as barras laranjas representam os e-mails malignos. Conforme é possível verificar, e pegando nas 2 primeiras amostragem, verificamos que, no eixo do y, 8 e-mails benignos e 3 e-mails malignos contém 2% de erros no corpo do e-mail, representado no eixo do x. Já na segunda amostragem, 50 e-mails benignos e 16 e-mails malignos contém 3% de erros na totalidade de palavras escritas no e-mail.

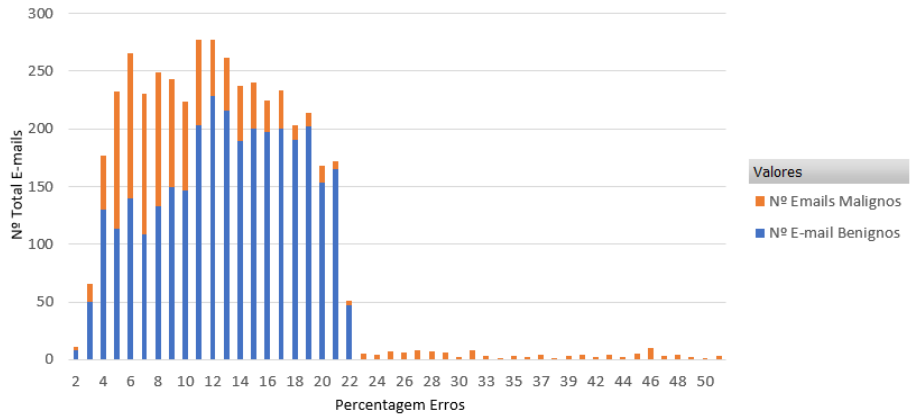


Figura 4.13: Gráfico de frequências relativo à relação percentual do número de erros em relação ás palavras escritas nos e-mails Benignos e Malignos

4.6.2 Análise de valores nulos

Uma das verificações a que se fez, na fase de análise exploratório, foi a existência de valores nulos no *dataset*. Verificou-se nesta análise que os únicos valores em falta diziam respeito ás *features* devolvidas pelo NER, ou seja, como os POS (Part Of Speech) variam de e-mail para e-mail, a junção de todos existem POS que se verificam nuns e-mails mas que não existem noutros, gerando assim os valores nulos. O tratamento dos valores nulos neste caso é simples. A existência de valores nulos significa que foram encontradas zero ocorrências daquela *feature* no e-mail em análise e por isso os valores nulos foram substituídos por zero.

4.6.3 Análise de *outliers*

Durante a análise exploratória verificou-se a existência de um e-mail de *phishing* com um número excessivo de erros. Na Figura 4.14 é ilustrado o boxplot relativo ao número de erros em relação a cada e-mail. O boxplot ajuda a evidenciar a existência de *outliers*. Assim, e após análise, verificou-se que o e-mail em causa ainda continha elementos HTML, sendo estes considerados erros tipográficos. De modo a normalizar o valor excessivo de erros verificado nesse e-mail, e ao invés de processar o *dataset* de novo resolveu-se substituir o número de erros pelo valor médio de erros. Na Figura 4.15 é ilustrado o resultado após a transformação do valor. Existem várias estratégias a adotar na existência de *outliers*, como por exemplo eliminar o e-mail. Contudo, foi feito o balanceamento para estudar

a forma como é feito este processo e, também, para verificar alguma afetação que possa gerar no resultado final. No final, não se verificou relevância pela alteração de um e-mail.

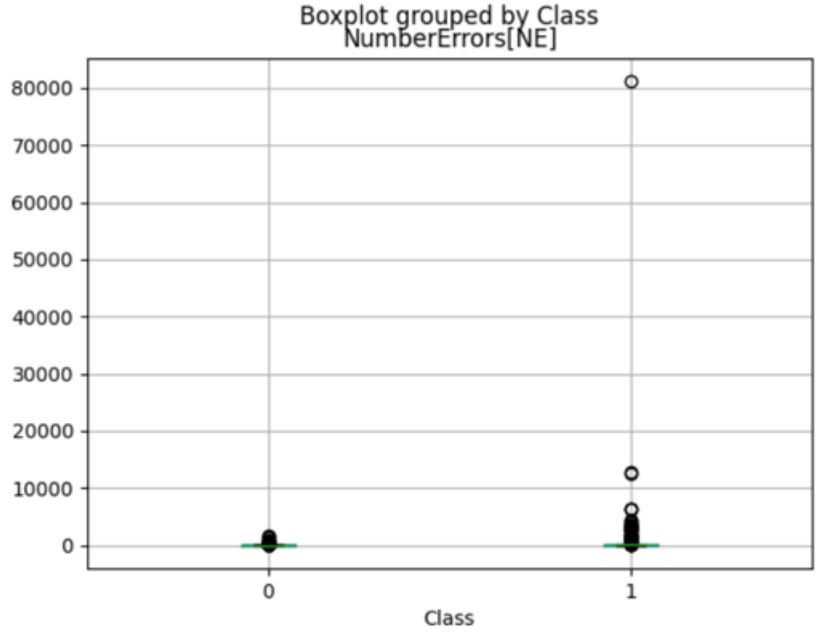


Figura 4.14: Gráfico de número de erros, por e-mail, entre classes

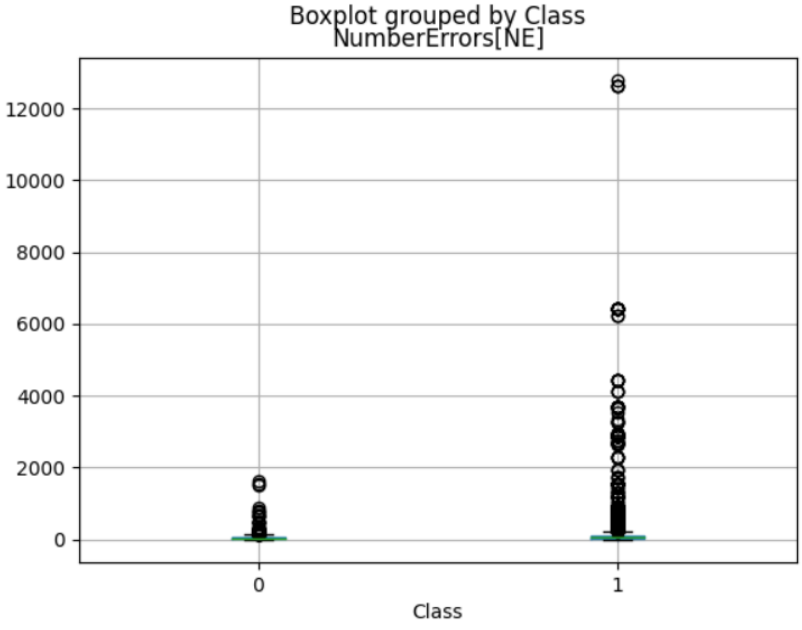


Figura 4.15: Gráfico de número de erros, depois de normalizado

4.6.4 Balanceamento do *dataset*

Conforme indicado no capítulo 4, o *dataset* é constituído por 4082 e-mails Benignos e 1298 e-mails Malignos. Antes de passar à fase da análise de dados é importante balancear os dados, uma vez que os algoritmos de ML são mais eficientes quando o número de amostras, de ambas as classes, são aproximados. Posto isto, foi utilizado um mecanismo de balanceamento de dados denominado de ROS. Este mecanismo, conforme descrito no capítulo 3, consiste em adicionar amostras à classe com menor número de valores/amostras. O que o mecanismo faz é escolher exemplos da classe minoritária e adiciona esses exemplos ao dataset sujeito a treino. O resultado do balanceamento é apresentado na tabela 4.4.

Feature	Antes Balanceamento		Após Balanceamento	
PoS	Benignos	Malignos	Benignos	Malignos
_SP	260695	257511	260695	727657
ADD	39801	25759	39801	67485
-LRB-	34949	27455	34949	69824
CD	134457	90236	134457	249756
-RRB-	37050	31292	37050	81373
IN	173015	49997	173015	132176
NNP	505519	272742	505519	764011
:	186473	160936	186473	426112
,	96273	38902	96273	104934
RB	45136	17455	45136	46332
VBN	32217	18377	32217	48388
JJ	93967	58814	93967	164911
NN	330172	259997	330172	690672
NNS	64141	32235	64141	85370
HYPH	108766	87555	108766	234689
VBG	24992	4837	24992	13044
VBZ	29776	5789	29776	15760
DT	93058	18765	93058	48690

Tabela 4.4 Continuação da página anterior

Feature	Antes Balanceamento		Após Balanceamento	
VBD	29473	5092	29473	13228
XX	60802	89542	60802	226186
”	22423	35256	22423	87105
POS	9660	861	9660	2491
VB	61715	21648	61715	57405
PRP	51781	13466	51781	36441
NFP	41796	28923	41796	75498
SYM	30517	61676	30517	187142
NNPS	10443	3062	10443	8988
.	50170	17784	50170	44676
CARDINAL	42199	36962	42199	100968
ORG	56081	50266	56081	153476
DATE	29770	11924	29770	31679
PERSON	64499	12425	64499	34067
TIME	7130	1603	7130	4271
GPE	21920	3993	21920	10161
S	13902284	14544658	13902284	42320513
NW	1948981	1890590	1948981	5548411
NE	175331	352090	175331	1145632
“	19445	23309	19445	57161
UH	10962	11499	10962	27961
VBP	28657	13875	28657	39100
WORK_OF_ART	2760	1948	2760	4909
MD	15014	3774	15014	9942
RP	5124	566	5124	1410
$	9511	4376	9511	11709
PRP$	17626	10967	17626	28992
CC	31977	8121	31977	21937

Tabela 4.4 Continuação da página anterior

Feature	Antes Balanceamento		Após Balanceamento	
TO	19803	5207	19803	13721
JJS	2099	404	2099	1149
FW	3258	9796	3258	26604
LS	4570	2027	4570	5560
WP	5189	195	5189	492
PRODUCT	1065	393	1065	1049
MONEY	2890	1629	2890	4612
PERCENT	2317	1183	2317	3057
ORDINAL	1433	11859	1433	32888
EVENT	665	43	665	132
WDT	4665	310	4665	772
NORP	8516	748	8516	2119
PDT	519	127	519	300
RBR	1481	279	1481	800
JJR	3422	1722	3422	4380
WRB	4175	458	4175	1197
EX	1668	135	1668	360
RBS	792	237	792	720
LAW	399	54	399	142
LOC	617	214	617	551
FAC	634	137	634	351
QUANTITY	161	109	161	320
AFX	18	1	18	3
LANGUAGE	61	8	61	24
WP$	98	5	98	13

Tabela 4.4: Variação das ocorrências por feature antes e após o balanceamento do *dataset*

Feita a análise exploratória e preparado o *dataset* em conformidade, o mesmo está apto a ser analisado. A análise de dados é apresentada na próximo capítulo.

Capítulo 5

Análise de dados

Neste capítulo é apresentada a análise de dados realizada, tendo por base o *dataset* preparado, de acordo com o descrito no capítulo anterior. A metodologia de análise de dados é classificação supervisionada, ou seja, o modelo aprende a partir de resultados pré-definidos [47]. Assim, e partindo do *dataset* previamente classificado com amostras de e-mails benignos e de e-mails de *phishing*, o foco da análise é determinar métricas de análise e proceder à sua comparação com vista a:

- perceber se a classificação de e-mails a partir de um *dataset* baseado no conteúdo dos e-mails é viável;
- identificar, do conjunto das 72 *features*, quais as mais significativas para a análise, aplicando diferentes metodologias de seleção de *features*
- analisar o impacto que o números de *features* (e.g., 5, 10 e 15 *features*) diferentes poderá ter nos resultados;
- determinar qual ou quais os algoritmos de ML que mais se adequam à classificação dos e-mails de *phishing*.

Na próxima secção é apresentado o método de seleção de *features*.

5.1 Seleção de *features*

A lista das *features* mais significativas para a análise de dados através dos algoritmos de ML é um aspeto importante para a análise. A seleção de *features* permite reduzir as 72

features existentes no *dataset* a um subconjunto de N, tornando dessa forma o processo de análise mais rápido e com o objetivo de não perder eficácia na análise.

Para o processo de seleção de *features* foram usados algoritmos de seleção de *features* que, no âmbito do ML, identificam as variáveis mais significativas, ou seja, aquelas que maximizam o acerto na classificação. Para o efeito, foram usados dois algoritmos de seleção: o RF, com uso da função feature_importante; o SKB. A escolha destes dois algoritmos deveu-se ao facto de:

- o algoritmo RF resolver problemas de classificação, nomeadamente se o e-mail é spam ou não, ter bom desempenho e possuir implementações nas mais diversas bibliotecas de Inteligência Artificial (IA) atualmente [48];
- o SKB tem um bom desempenho e é de simples funcionamento. Apenas é necessário indicar na função o parâmetro k, ou seja, o número de *features* do *dataset* a selecionar [49].

O algoritmo RF faz uso de uma função chamada feature_importance [50] que faz a análise das *features* e determina as *features*, mais importantes a partir do conjunto das mesmas. Conforme é possível ver nas tabelas 5.1, 5.2, 5.3, 5.4 e 5.5, o mecanismo de seleção RF não atribui sempre o mesmo peso ás *features*. Por isso, e para estudar qual o impacto que a seleção de *features* poderá ter, através deste método, optou-se por executar o algoritmo várias vezes (cinco vezes no estudo). O objetivo foi o de verificar, por repetição, quais as *features* mais significativas para a análise e verificar também qual o impacto em termos de resultados que a seleção de *features* tem. De acordo com o resultado do algoritmo, pode-se ver que as *features* -LRB- (left round bracket) e -RRB- (right round bracket), são as que tem maior peso na avaliação das *features*, seguido das *features* NNS, ADD cujo peso varia de iteração para iteração. Existe também uma discrepância em algumas *features* que, comparadas com as que tem mais peso, vão variar de iteração para iteração. Entre estas destacam-se as *features* S e NW.

RF feature_importannce	*Features*	Peso (%)
1ª Iteração	-LRB-	0.073
	-RRB-	0.043
	ADD	0.036
	VBN	0.028
	CARDINAL	0.023
	NNS	0.023
	FW	0.021
	PRP$	0.019
	CD	0.019
	XX	0.018
	WP	0.017
	POS	0.013
	HYPH	0.013
	IN	0.013
	NORP	0.013

Tabela 5.1: *features* mais significativas na primeira iteração do algoritmo RF

RF feature_importannce	*Features*	Peso (%)
2ª Iteração	-RRB-	0.074
	-LRB-	0.065
	NNS	0.038
	ADD	0.034
	HYPH	0.024
	VBN	0.02
	EVENT	0.019
	XX	0.019
	POS	0.018
	CARDINAL	0.018
	PRP$	0.018
	IN	0.017
	S	0.017
	CD	0.016
	NW	0.014

Tabela 5.2: *features* mais significativas na segunda iteração do algoritmo RF

RF feature_importannce	*Features*	Peso (%)
3ª Iteração	-RRB-	0.062
	-LRB-	0.061
	ADD	0.039
	NNS	0.034
	PRP$	0.029
	VBN	0.029
	FW	0.021
	POS	0.021
	CARDINAL	0.020

Tabela 5.3 Continuação da página anterior

RF feature_importannce	*Features*	Peso (%)
	EVENT	0.018
	XX	0.018
	IN	0.016
	HYPH	0.015
	WP	0.015
	NN	0.014

Tabela 5.3: *features* mais significativas na terceira iteração do algoritmo RF

RF feature_importannce	*Features*	Peso (%)
4ª Iteração	-RRB-	0.067
	-LRB-	0.044
	NNS	0.040
	ADD	0.039
	CD	0.031
	XX	0.026
	VBN	0.025
	CARDINAL	0.024
	POS	0.023
	EVENT	0.021
	NN	0.019
	FW	0.018
	PRP$	0.017
	HYPH	0.016
	NORP	0.015

Tabela 5.4: *features* mais significativas na quarta iteração do algoritmo RF

RF feature_importannce	*Features*	Peso (%)
5ª Iteração	-LRB-	0.068
	-RRB-	0.047
	NNS	0.036
	ADD	0.034
	XX	0.028
	VBN	0.026
	CARDINAL	0.023
	HYPH	0.022
	EVENT	0.022
	CD	0.021
	POS	0.020
	PRP$	0.017
	W	0.016
	WP	0.015
	NORP	0.014

Tabela 5.5: *features* mais significativas na quinta iteração do algoritmo RF

Tal como referido anteriormente, a execução das cinco iterações prende-se com o facto do método de seleção de *features*, obtidas a partir do algoritmo RF, não ser linear. Uma vez que a cada iteração são devolvidas diferentes *features*, foram analisas as 15 *features* retornadas a cada iteração e qual o seu peso. Deste modo é possível selecionar as *features* que, de iteração para iteração, se mantiveram como as mais significativas para a análise.

Relativamente ao algoritmo de seleção de *features* SKB (SelectKBest), este apresentou sempre um *output* constante. Os resultados apresentados na Tabela 5.6, dão a conhecer as 15 *features* que, de acordo com o SKB, são as mais relevantes para a análise. Ao passo que as *features* -LRB- e o -RRB-, foram identificadas pelo algoritmos de seleção de *features* RF como as mais relevantes, o mecanismo SKB nem sequer as considera. Daí a importância de executar e avaliar diferentes mecanismos para determinar de forma eficaz as *features*

mais significativas. Entre os dois algoritmos foram também identificadas *features* comuns (e.g., XX, HYPH, CARDINAL, CD, NNP). Os resultados sugerem que estas *features* tem realmente peso para a classificação.

SKB	*Features*
Top 15 *features*	_SP
	CD
	NNP
	:
	NN
	HYPH
	XX
	”
	SYM
	CARDINAL
	ORG
	S
	NW
	NE
	ORDINAL

Tabela 5.6: Seleção de features através do algoritmo SKB

5.2 Fase de Treino

De forma a classificar se o e-mail é *phishing* ou não, usaram-se vários algoritmos de ML. Cada algoritmo recebe a variável dependente e o conjunto de variáveis independentes, sendo estas as determinadas através dos algoritmos de seleção de *features*. Para treinar os algoritmos de ML adotou-se o mecanismo de K-fold Cross Validation.

O uso do mecanismo de K-fold Cross Validation deveu-se ao facto de:

- ser uma técnica conhecida pela sua eficiência em testes de desempenho de ML;
- este utilizar os dados de forma eficiente, pois faz uso de todos os dados tanto na etapa de treino, como na etapa de teste.

O número de *folds* usado foi 5 ($k = 5$). Deste modo o *dataset* é dividido em cinco

blocos, sendo quatro usados para treino e um para testes/validação. O próprio método faz com que os blocos de treino e teste variem a cada iteração. Este processo é ilustrado na Figura 5.1 com um valor de $k = 4$ para efeitos de apresentação.

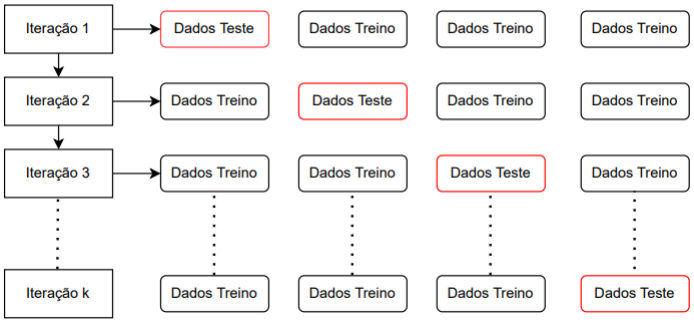


Figura 5.1: Cenário da seleção de dados para treino e teste com k iterações

No âmbito do estudo foram usados seis algoritmos de ML: GNB; LOR; DTC; RF; SVM e MLPC. A escolha destes algoritmos, deriva da análise feita ao estado da arte. De acordo com a mesma estes algoritmos são os que apresentam um melhor desempenho em problemas de classificação supervisionada de dados binários.

A eficácia de cada algoritmo (*accuracy*) resulta do valor médio da eficácia obtida a cada iteração do Cross Validation. O objetivo final é determinar qual o algoritmo com maior nível de eficácia. Esse algoritmo será o eleito para o problema de classificação dos e-mails, podendo ser integrado numa solução final. É de salientar ainda que, uma vez que os algoritmos de seleção de *features* devolveram diferentes *features* como sendo as mais significativas para a análise, o estudo contempla a análise do impacto que o próprio processo de seleção de *features* tem no resultado. Esta análise é apresentada na próxima subsecção.

5.2.1 Análise do impacto do processo de seleção de features no resultado

Esta secção aborda o impacto, em termos de *accuracy*, que as diferentes *features* tem para cada um dos algoritmos de ML usados no estudo.

Tal como referido anteriormente, o algoritmo RF permite, através da função feature_importance, obter a lista de *features* mais relevantes para a análise. Uma vez que este método não devolve uma mesma lista de *features*, decidiu-se analisar o peso que as *features*

tem na análise. Para tal, a análise de dados foi executada cinco vezes (5 iterações), sempre com variáveis independentes diferentes, para verificar se o algoritmo de ML é afetado pelos dados de entrada. Nas tabelas 5.7, 5.8, 5.9 são apresentados os resultados obtidos por cada algoritmo de ML, ao longo das cinco iterações. Cada iteração tem como *input* uma lista de *features* diferente, representadas nas tabelas 5.1, 5.2, 5.3, 5.4 e 5.5. Os resultados contemplam um número diferente de *features* (15, 10 e 5 *features*) permitindo analisar a influência que esse número terá no resultado de *accuracy*. As 10 e 5 *features* correspondem ao top 10 e top 5 de cada iteração realizada aquando da seleção de *features*.

RF com 15 *features*		
	Algoritmo de ML	Accuracy
1ª Iteração	GNB	64%
	LOR	91.14%
	DTC	87.43%
	RF	91.33%
	SVM	62.74%
	MLPC	35.66%
2ª Iteração	GNB	64.01%
	LOR	89.8%
	DTC	86.07%
	RF	90.68%
	SVM	38.39%
	MLPC	51.36%
3ª Iteração	GNB	67.19%
	LOR	87.57%
	DTC	86.29%
	RF	91.81%
	SVM	76.12%
	MLPC	62.84
4ª Iteração	GNB	66.81%
	LOR	89.89%

Tabela 5.7 Continuação da página anterior

RF com 15 *features*		
	DTC	89.34%
	RF	92.13%
	SVM	73.82%
	MLPC	34.34
5ª Iteração	GNB	67.19%
	LOR	87.57%
	DTC	86.29%
	RF	91.81%
	SVM	76.12%
	MLPC	62.84

Tabela 5.7: Desempenho dos diferentes algoritmos com atribuição de 15 features

RF com 10 *features*		
	Algoritmo de ML	Accuracy
1ª Iteração	GNB	64.51%
	LOR	72.60%
	DTC	88.44%
	RF	73.55%
	SVM	73.55%
	MLPC	90.38%
2ª Iteração	GNB	65.96%
	LOR	82.05%
	DTC	89.5%
	RF	90.44%
	SVM	77.26%
	MLPC	89.91%

Tabela 5.8 Continuação da página anterior

RF com 10 *features*		
3ª Iteração	GNB	64.7%
	LOR	78.44%
	DTC	87.48%
	RF	90.35%
	SVM	72.41%
	MLPC	90.13%
4ª Iteração	GNB	65.84%
	LOR	79.45%
	DTC	88.44%
	RF	91.2%
	SVM	77.55%
	MLPC	65.84%
5ª Iteração	GNB	63.88%
	LOR	74.23%
	DTC	83.49%
	RF	89.35%
	SVM	72.84%
	MLPC	89.40%

Tabela 5.8: Desempenho dos diferentes algoritmos com atribuição de 10 features

RF com 5 *features*		
	Algoritmo de ML	Accuracy
1ª Iteração	GNB	62.26%
	LOR	61.72%
	DTC	75.05%
	RF	85.57%

Tabela 5.9 Continuação da página anterior

RF com 5 *features*		
	SVM	76.34%
	MLPC	86.67%
2ª Iteração	GNB	62.26%
	LOR	53.63%
	DTC	79.32%
	RF	85.7%
	SVM	76.34%
	MLPC	86.37%
3ª Iteração	GNB	58.09%
	LOR	63.91%
	DTC	72.52%
	RF	81.12%
	SVM	77.89%
	MLPC	87.4%
4ª Iteração	GNB	55.27%
	LOR	58.42%
	DTC	78.99%
	RF	83.16%
	SVM	71.64%
	MLPC	85.85%
5ª Iteração	GNB	71.75%
	LOR	64.24%
	DTC	84.5%
	RF	87.11%
	SVM	71.75%
	MLPC	87.78%

Tabela 5.9: Desempenho dos diferentes algoritmos com atribuição de 5 features

Da análise dos diferentes algoritmos, e ao longo das iterações, verificou-se um impacto residual no resultado da seleção de *features*, devolvido pelo mecanismo de seleção RF. No conjunto de 15 *features*, o melhor resultado, entre os algoritmos de ML, foi de 92.13% de *accuracy*. No conjunto das 10 *features*, o melhor resultado obtido foi de 91.2% de *accuracy*. Por fim, no conjunto de 5 *features* o melhor resultado de *accuracy* obtido foi de 87.78%. Este resultado permite concluir que o número de *features* tem impacto no resultado da análise e daí ser importante fazer diferentes combinações de *features* para a fase de treino dos modelos.

A escolha de *features* também é um processo significativo no que toca à obtenção de melhores resultados. No conjunto de 15 *features*, existe uma diferença de 1.45% entre a melhor avaliação (4ª iteração) e a pior avaliação (2ª iteração), no algoritmo de ML RF onde foi obtido o melhor resultado da análise dos algoritmos de ML. No Conjunto de 10 *features* existe uma diferença bastante significativa de 17.65% entre a melhor avaliação (4ª iteração) e a pior avaliação (1ª iteração). Após verificado o conjunto de *features*, em ambas as iterações, verificou-se a diferença de duas *features*. O que sugere que a *feature* POS, *possessive ending*, e a *feature* EVENT, não presente na 1ª iteração, são *features* importantes para a análise comparada com a *feature* FW e a *feature* PRP$, presentes na 1ª iteração. No conjunto de 5 *features* existe uma diferença de 0,93% entre a melhor avaliação (5ª iteração) e a pior avaliação (4ª iteração), no algoritmo MLPC, onde foi obtido o melhor resultado.

Relativamente ao conjunto de *features* selecionadas através do algoritmo SKB verificou-se que os resultados são menos satisfatórios, quando comparado com o algoritmo de seleção RF. Para efeitos de comparação, o melhor resultado obtido com a aplicação do SKB foi de 85.63%, conforme é possível verificar na Tabela 5.10.

Nº *features*	Algoritmo de ML	Accuracy
15 *features*	LOR	67.73%
	DTC	66.9%
	RF	66.23%
	GNB	59.26%
	SVM	37.99%

Tabela 5.10 Continuação da página anterior

Nº *features*	Algoritmo de ML	Accuracy
	MLPC	34.62%
10 *features*	RF	83.16%
	DTC	81.83%
	LOR	69.31%
	GNB	60.93%
	MLPC	51.70%
	SVM	38.94%
5 *features*	RF	85.63%
	DTC	84.51%
	LOR	70.77%
	GNB	62.29%
	SVM	38.75%
	MLPC	38.45%

Tabela 5.10: Melhores resultados obtidos com o método de seleção de *features* SKB

5.3 Análise global dos resultados

Nesta secção, é feita a análise global dos resultados. Esta análise baseia-se no resultado obtido na 4ª iteração do conjunto de 15 *features* obtidas através da função feature_importance após a aplicação do algoritmo de ML RF.

Tal como é apresentado na Tabela 5.11, na 4ª iteração do conjunto de 15 *features* o algoritmo de ML RF foi o que apresentou o melhor resultado com 92.13% de acerto. É ainda de destacar que, no conjunto de 10 *features* o mesmo algoritmo obteve uma taxa de acerto de 91.2% o que indica que, com um número de *features* menor o resultado, em termos de acerto não é muito afetado. Optar por um número menor de *features* contribuí para uma maior rapidez na determinação se o e-mail é de *phishing* ou não.

Nº *features*	Algoritmos	Accuracy Cross-Validation
15 *features*	RF	92.13%
	LOR	89.89%
	DTC	89.34%
	SVM	73.81%
	GNB	66.81%
	MLPC	34.34%

Tabela 5.11: Melhores resultados obtidos com nas diferentes iterações com o mecanismo de seleção RF

É ainda importante salientar o impacto que as *features* tem na avaliação dos algoritmos. Conforme referido na secção 5.2.1 a escolha das *features* é um fator importante no que toca à avaliação dos algoritmos de ML. Neste estudo isso é evidente pois as diferentes *features*, devolvidas pelos métodos de seleção, tem um impacto considerável na taxa de acerto. Para um mesmo algoritmo de ML, com um mesmo número de *features*, os métodos de seleção de *features* influenciam a taxa de acerto em mais de 25%. Dentro do mesmo método de seleção de *features* a variação é menor, mas ainda assim de 1.45%.Estas comparações são referentes ao algoritmo de ML RF.

Capítulo 6

Conclusão

A engenharia social é uma arte antiga que tira partido de fragilidades humanas para iludir e enganar. A engenharia social evoluiu ao longo dos tempos e o crescimento da Internet tornou o fenómeno ainda maior. Hoje em dia, um engenheiro social consegue, através da Internet, chegar a um vasto número de vítimas potenciando assim os seus ataques e retorno dos mesmos. Dentro da engenharia social uma técnica muito utilizada é o *phishing*. Vários estudos indicam que o *phishing* é a forma de ciberataque mais popular, sendo o e-mail o canal mais utilizado para o efeito.

Parar/evitar os ataques de *phishing* é uma tarefa complexa. Estes ataques são dirigidos aos humanos que tendem muitas da vezes a ceder aos pedidos dos atores maliciosos. Outro problema é que basta uma pessoa cair no engodo para comprometer toda a organização. As organizações estão cada vez mais conscientes do problema e desenvolvem projetos de treino e programas de consciencialização. Estas iniciativas são extremamente importantes mas não erradicam o problema. Suportar ou complementar estas ações com meios tecnológicos avançados, capaz de detetar se um e-mail é de *phishing* ou não contribuirá diretamente para uma melhor proteção do utilizadores e das organizações.

No trabalho aqui apresentado foi feito um estudo para validar o potencial da utilização de ML aplicado ao conteúdo dos e-mails com o objetivo de classificar estes quando a ser de *phishing* ou não. Assim, uma fase inicial foi feita uma pesquisa de trabalhos que agregam o tema de ML e de anti-*phishing*. Foi enquadrado o tema com uma pequena introdução que realçava os prejuízos, problemas, tipos de *phishing* e da forma como são feitos os ataques. Foi também iniciado o estudo com a agregação de um conjunto de e-mails, benignos e

malignos, que constituíram o *dataset* inicial. Foi necessário proceder a um tratamento dos dados dos e-mails, que incluí a tradução dos e-mails para um único idioma, a remoção do cabeçalho dos e-mails e a remoção de caracteres indesejados dos mesmos. Os e-mails tratados foram depois submetidos a algoritmos de PLN que permitiram a leitura do texto e a extração de um conjunto de *features* relacionados com aspetos linguísticos do e-mail. Esta extração constitui um *dataset* base que foi depois analisado e tratado chegando a um *dataset* final constituído por 72 *features*.

O *dataset* foi balanceado e submetido a algoritmos de seleção de *features*, identificando-se o conjunto de *features* mais significativas para análise. As *features* mais significativas foram submetidas a diferentes algoritmos de classificação ML.

Foram usados dos algoritmos de seleção de *features*: O método feature_importance dado pelo algoritmo de RF; o método SelectKBest (SKB). Entre os dois o método feature_importance do RF foi o que deu melhores resultados. Uma vez que este método devolve *features* diferentes a cada execução, procedeu-se a uma execução iterativa para validar até que ponto a lista de *features* influência os resultados. Ainda, e para estudar os algoritmos, foram escolhidos diferentes números de *features* a usar como *input* para os algoritmos de ML. O algoritmo RF, com 15 *features* selecionadas através do método feature_importance foi o que obteve melhores resultados: 92.13% de *accuracy*. A este nível conclui-se que, um algoritmo de seleção necessita de mais *features* que o outro para obtenção de melhores resultados.

De acordo com os resultados, a possibilidade de classificar e-mails como sendo de *phishing* ou não com uma taxa de acerto na ordem dos 92% é um excelente resultado. Reforça-se que a análise baseia-se no conteúdo do e-mail e que a mesma poderá ser complementada com outros estudos que se baseiam na análise da meta-informação do e-mail maximizando assim a capacidade de deteção.

Pessoas, Processos e Tecnologias são os três pilares de uma visão holística da cibersegurança. Neste âmbito, uma solução tecnológica de confiança que apoie a deteção de ataques de e-e-mail *phishing*, combinada com programas de treino e consciencialização das pessoas contribuirá para combater aquele que é o tipo de ciberataque mais frequente nos dias de hoje.

Referências

[1] lg. "O primeiro email foi enviado há 50 anos!" Em: (). (Acedido a 07/06/2022). URL: https://www.lg.com/pt/lg-magazine/tech-story/primeiro-email.

[2] e.Veritas. "A história do Phishing". Em: (). (Acedido a 07/12/2021). URL: https://www.emailveritas.com/pt/blog/historia-do-phishing.

[3] FBI. "Internet Crime Report 2020". Em: (2020). (Acedido a 27/02/2022). URL: https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf.

[4] Threat Hunter Team. "Threat Landscape Trends – Q1 2020". Em: (2020). (Acedido a 27/02/2022). URL: https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/threat-landscape-q1-2020.

[5] Proofpoint. "2021 State of the Phish Report". Em: (2020). (Acedido a 27/02/2022). URL: https://www.proofpoint.com/uk/resources/threat-reports/state-of-phish.

[6] Symantec. "Internet Security Threat Report". Em: (2020). (Acedido a 27/02/2022). URL: https://docs.broadcom.com/doc/istr-24-2019-en.

[7] APWG. "Phishing Activity Trends ReporT - 1st Quarter 2021". Em: *APWG* (2021). (Acedido a 27/02/2022). URL: https://docs.apwg.org/reports/apwg_trends_report_q1_2021.pdf.

[8] APWG. "Phishing Activity Trends ReporT - 2st Quarter 2021". Em: *APWG* (2021). (Acedido a 27/02/2022). URL: https://docs.apwg.org/reports/apwg_trends_report_q2_2021.pdf.

[9] APWG. "Phishing Activity Trends ReporT - 3st Quarter 2021". Em: *APWG* (2021). (Acedido a 27/02/2022). URL: https://docs.apwg.org/reports/apwg_trends_report_q3_2021.pdf.

[10] Gary Cohen. "Throwback Attack: How a single whaling email cost $61 million". Em: *Industrial Cybersecurity Pulse* (2019). (Acedido a 03/03/2021). URL: https://www.industrialcybersecuritypulse.com/throwback-attack-how-a-single-phishing-email-cost-61-million/.

[11] Zeljka Zorz. "Belgian bank Crelan loses €70 million to BEC scammers". Em: *Help Net Security* (2016). (Acedido a 03/03/2021). URL: https://www.helpnetsecurity.com/2016/01/26/belgian-bank-crelan-loses-e70-million-to-bec-scammers/.

[12] Timothy B. Lee Emily St. James. "The 2014 Sony hacks, explained". Em: *Vox* (2015). (Acedido a 03/03/2021). URL: https://www.vox.com/2015/1/20/18089084/sony-hack-north-korea.

[13] Tom Huddleston Jr. "How this scammer used phishing emails to steal over $100 million from Google and Facebook". Em: *CNBC* (2019). (Acedido a 03/03/2021). URL: https://www.cnbc.com/2019/03/27/phishing-email-scam-stole-100-million-from-facebook-and-google.html.

[14] Yi Shin Chen et al. "Detect phishing by checking content consistency". Em: 2014. DOI: 10.1109/IRI.2014.7051880.

[15] Eric Abraham Kalloor, Manoj Kumar Mishra e Joy Paulose. "Phishfort - Anti-phishing framework". Em: *International Journal of Engineering and Technology(UAE)* 7 (3 2018). ISSN: 2227524X. DOI: 10.14419/ijet.v7i3.4.14673.

[16] Siti Hawa Apandi, Jamaludin Sallim e Roslina Mohd Sidek. "Types of anti-phishing solutions for phishing attack". Em: vol. 769. 2020. DOI: 10.1088/1757-899X/769/1/012072.

[17] V. Suganya. "A Review on Phishing Attacks and Various Anti Phishing Techniques". Em: *International Journal of Computer Applications* 139 (1 2016). DOI: 10.5120/ijca2016909084.

[18] Shweta Sankhwar, Dhirendra Pandey e R. A. Khan. "Email Phishing: An Enhanced Classification Model to Detect Malicious URLs". Em: *EAI Endorsed Transactions on Scalable Information Systems* 6 (21 2019). ISSN: 20329407. DOI: 10.4108/eai.13-7-2018.158529.

[19] Minal Chawla e Siddarth Singh Chouhan. "A Survey of Phishing Attack Techniques". Em: *International Journal of Computer Applications* 93 (3 2014). DOI: 10.5120/16197-5460.

[20] Muhammad Dawood, Othman Bin Ibrahim e Waheeb Abdel Rahman Ali Abu-Ulbeh. "Enrich awareness of users to detect phishing websites". Em: *International Journal of Engineering and Advanced Technology* 8 (6 Special Issue 3 2019). ISSN: 22498958. DOI: 10.35940/ijeat.F1119.0986S319.

[21] Gaurav Varshney, Manoj Misra e Pradeep K. Atrey. "A survey and classification of web phishing detection schemes". Em: *Security and Communication Networks* 9 (18 2016). ISSN: 19390122. DOI: 10.1002/sec.1674.

[22] Krutika RaniSahu e Jigyasu Dubey. "A Survey on Phishing Attacks". Em: *International Journal of Computer Applications* 88 (10 2014). DOI: 10.5120/15392-4007.

[23] Tianrui Peng, Ian Harris e Yuki Sawa. "Detecting Phishing Attacks Using Natural Language Processing and Machine Learning". Em: vol. 2018-January. 2018. DOI: 10.1109/ICSC.2018.00056.

[24] Sisi Liu e Ickjai Lee. "A hybrid sentiment analysis framework for large email data". Em: 2016. DOI: 10.1109/ISKE.2015.91.

[25] Pranjal S. Bogawar e K. K. Bhoyar. "Soft computing approaches to classification of emails for sentiment analysis". Em: vol. 25-26-August-2016. 2016. DOI: 10.1145/2980258.2980304.

[26] Lopamudra Dey et al. "Sentiment Analysis of Review Datasets Using Naïve Bayes' and K-NN Classifier". Em: (2016). URL: https://www.mecs-press.org/ijieeb/ijieeb-v8-n4/IJIEEB-V8-N4-7.pdf.

[27] Viranjitha Lakshmi Maturi et al. “Twitter sentimental analysis using machine learning techniques”. Em: *International Journal of Innovative Technology and Exploring Engineering* 8 (6 2019). ISSN: 22783075. DOI: 10.35940/ijeat.c6281.029320.

[28] Adith Shetty et al. “Online Product Grading using Sentimental Analysis with SVM”. Em: 2020. DOI: 10.1109/ICICCS48265.2020.9121098.

[29] R. Geetha, T. Padmavathy e R. Anitha. “Prediction of the academic performance of slow learners using efficient machine learning algorithm”. Em: *Advances in Computational Intelligence* 1 (4 2021). ISSN: 2730-7794. DOI: 10.1007/s43674-021-00005-9.

[30] Kalpdrum Passi e Niravkumar Pandey. “Increased Prediction Accuracy in the Game of Cricket Using Machine Learning”. Em: *International Journal of Data Mining & Knowledge Management Process* 8 (2 2018). ISSN: 2231007X. DOI: 10.5121/ijdkp.2018.8203.

[31] Benai Kumar. “10 Techniques to deal with Imbalanced Classes in Machine Learning”. Em: (). (Acedido a 07/05/2021). URL: https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/.

[32] Pedro César Tebaldi Gomes. “Principais algoritmos de machine learning”. Em: *datageeks* (2019). (Acedido a 10/08/2020). URL: https://www.datageeks.com.br/algoritmos-de-machine-learning.

[33] Renuka Joshi. “Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures”. Em: *Exsilio* (2016). (Acedido a 09/10/2021). URL: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.

[34] Victor Zeng Rakesh Verma e Houtan Faridi. “Data Quality for Security Challenges: Case Studies of Phishing, Malware and Intrusion Detection Datasets”. Em: *ACM SIGSAC Conference on Computer and Communications Security (CCS) 2019* 1 (2019). (Acedido a 05/08/2020). ISSN: 2605–2607. DOI: https://doi.org/10.1145/3319535.3363267.

[35] Tushaar Gangavarapu. “Feature Selection for Spam and Phishing Detection”. Em: *GitHub* (2018). (Acedido a 30/09/2021). URL: https://github.com/TushaarGVS/Phishing.

[36] JournalDev. “Python Pandas Module Tutorial”. Em: (). (Acedido a 29/09/2020). URL: https://www.journaldev.com/29055/python-pandas-module-tutorial.

[37] Python. “csv — CSV File Reading and Writing”. Em: (). (Acedido a 30/09/2020). URL: https://docs.python.org/3/library/csv.html.

[38] Jordan Kalebu. “How to Translate Languages in Python”. Em: (2020). (Acedido a 07/11/2020). URL: https://docs.python.org/3/library/csv.html.

[39] PythonCode. “How to Translate Languages in Python”. Em: (). (Acedido a 07/11/2020). URL: https://docs.python.org/3/library/csv.html.

[40] Python. “ email — An email and MIME handling package”. Em: (). (Acedido a 08/11/2020). URL: https://docs.python.org/3/library/email.html.

[41] João Diogo Coelho Ferreira. “Python para Pré-processamento e Extracção de Características a partir de Texto Português”. Em: (2019). (Acedido a 27/02/2022). URL: https://eg.uc.pt/bitstream/10316/88030/1/Tese_Final_Joao_Ferreira.pdf.

[42] Pypi. “ spaCy: Industrial-strength NLP”. Em: (). (Acedido a 30/09/2020). URL: https://pypi.org/project/spacy/#description.

[43] pyspellchecker. “pyspellchecker”. Em: (). (Acedido a 07/11/2020). URL: https://pyspellchecker.readthedocs.io/en/latest/.

[44] Steven Loria. “TextBlob: Simplified Text Processing”. Em: (). (Acedido a 21/11/2020). URL: https://textblob.readthedocs.io/en/dev/.

[45] Python. “pickle — Python object serialization”. Em: (). (Acedido a 20/01/2021). URL: https://docs.python.org/3/library/pickle.html.

[46] Shivam Aggarwal, Vishal Kumar e S. D. Sudarsan. “Identification and detection of phishing emails using natural language processing techniques”. Em: vol. 2014-September. 2014. DOI: 10.1145/2659651.2659691.

[47] Muhammad Iqbal e Zhu Yan. “Supervised Machine Learning Approaches: A survey”. Em: *International Journal of Soft Computing* 5 (abr. de 2015), pp. 946–952. DOI: 10.21917/ijsc.2015.0133.

[48] Cínthia Pessanha. “Random Forest: como funciona um dos algoritmos mais populares de ML”. Em: *medium* (). (Acedido a 03/02/2021). URL: https://medium.com/cinthiabpessanha/random-forest-como-funciona-um-dos-algoritmos-mais-populares-de-ml-cc1b8a58b3b4.

[49] Paulo Vasconcellos. “Como selecionar as melhores features para seu modelo de Machine Learning”. Em: *medium* (2019). (Acedido a 03/02/2021). URL: https://medium.com/data-hackers/como-selecionar-as-melhores-features-para-seu-modelo-de-machine-learning-faf74e357913.

[50] scikit-learn. “sklearn.ensemble.RandomForestClassifier”. Em: *scikit-learn* (). (Acedido a 25/02/2022). URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

Appendices

Apêndice A

Features recolhidas

Conjunto de *features* recolhidas pelo mecanismo de PLN.

Conjunto de 72 Features	
None[_SP]	NumberErrors[NE]
email[ADD]	SentimentalAnalysis[SA]
left round bracket[-LRB-]	opening quotation mark[``]
cardinal number[CD]	interjection[UH]
right round bracket[-RRB-]	verb, non-3rd person singular present[VBP]
conjunction, subordinating or preposition[IN]	Titles of books, songs, etc.[WORK_OF_ART]
noun, proper singular[NNP]	verb, modal auxiliary[MD]
punctuation mark, colon or ellipsis[:]	adverb, particle[RP]
punctuation mark, comma[,]	symbol, currency[$]
adverb[RB]	pronoun, possessive[PRP$]
verb, past participle[VBN]	conjunction, coordinating[CC]
adjective[JJ]	infinitival "to"[TO]
noun, singular or mass[NN]	adjective, superlative[JJS]
noun, plural[NNS]	foreign word[FW]
punctuation mark, hyphen[HYPH]	list item marker[LS]
verb, gerund or present participle[VBG]	wh-pronoun, personal[WP]
verb, 3rd person singular present[VBZ]	Objects, vehicles, foods, etc. (not services)[PRODUCT]
determiner[DT]	Monetary values, including unit[MONEY]
verb, past tense[VBD]	Percentage, including "%"[PERCENT]
unknown[XX]	first, "second", etc.[ORDINAL]
closing quotation mark["]	Named hurricanes, battles, wars, sports events, etc.[EVENT]
possessive ending[POS]	wh-determiner[WDT]
verb, base form[VB]	Nationalities or religious or political groups[NORP]
pronoun, personal[PRP]	predeterminer[PDT]
superfluous punctuation[NFP]	adverb, comparative[RBR]
symbol[SYM]	adjective, comparative[JJR]
noun, proper plural[NNPS]	wh-adverb[WRB]
punctuation mark, sentence closer[.]	existential there[EX]
Numerals that do not fall under another type[CARDINAL]	adverb, superlative[RBS]
Companies, agencies, institutions, etc.[ORG]	Named documents made into laws.[LAW]
Absolute or relative dates or periods[DATE]	Non-GPE locations, mountain ranges, bodies of water[LOC]
People, including fictional[PERSON]	Buildings, airports, highways, bridges, etc.[FAC]
Times smaller than a day[TIME]	Measurements, as of weight or distance[QUANTITY]
Countries, cities, states[GPE]	affix[AFX]
Size[S]	Any named language[LANGUAGE]
NumberWords[NW]	wh-pronoun, possessive[WP$]

Apêndice B

Dataset final

Dataset final após análise e extração de features pelo mecanismo PLN.