PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Curso de Graduação em Engenharia de Computação

Vinícius Ferreira Salgado

CRION: DATASET PARA ANÁLISE E PREDIÇÃO DE CRIMES

Belo Horizonte

2017

Vinícius Ferreira Salgado

CRION: DATASET PARA ANÁLISE E PREDIÇÃO DE CRIMES

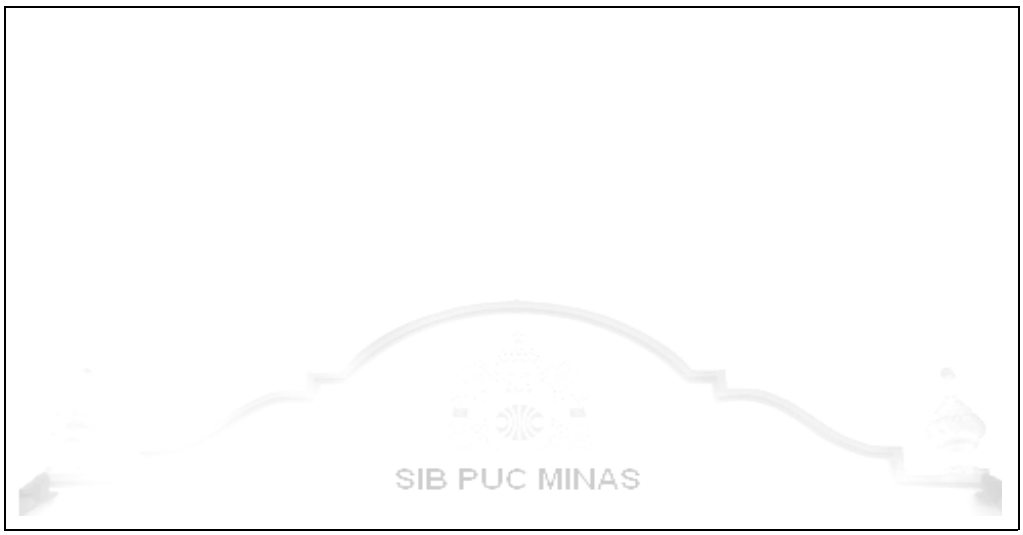
Monografia apresentada ao Curso de Engenharia da Computação da Pontifícia Universidade Católica de Minas Gerais, para obtenção do título de Engenheiro da Computação.

Orientador: Prof. Dr. Wladmir Cardoso Brandão

Belo Horizonte

2017

FICHA CATALOGRÁFICA
Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais



Vinícius Ferreira Salgado

CRION: DATASET PARA ANÁLISE E PREDIÇÃO DE CRIMES

Monografia apresentada ao Curso de Engenharia da Computação da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Engenheiro da Computação.

Prof. Dr. Wladmir Cardoso Brandão –
PUC Minas

Prof. Dr. Alexandre Texeira – PUC Minas

Belo Horizonte, 2017.

Dedico esse trabalho aos meus Pais, sem eles seria impossível chegar aqui e a minha noiva Luciana Lírio pela força e incentivo para seguir em frente.

AGRADECIMENTOS

Aos professores do COTEMIG, por guiarem meus primeiros passos na área da tecnologia.

A todos os professores que foram exemplos de postura ética e profissional.

Ao Professor Wladmir Cardoso Brandão, pela diligente orientação.

"O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis."

José de Alencar

RESUMO

O presente trabalho tem como objetivo construir uma base de dados para realização de experimentos de análise e predição de padrões criminais a partir de dados da *Web*. Em particular, coletamos, filtramos e caracterizamos dados relacionados a crimes da rede social *Twitter*. Além disso, realizamos experimentos com usuários para rotulação dos dados da base de dados para que possam ser efetivamente utilizados em experimentos de classificação automática.

Para avaliar a qualidade na base de dados, realizamos um experimento de classificação. Há um estudo de validação realizado em cada um dos *kernels*, onde é comparada a eficiência da classificação. Foram utilizadas 2 classes para o treinamento dos classificadores, avaliou-se o desempenho de classificadores treinados com dados por usuário e pela base de dados conjunta. Pôde-se avaliar que o kernel Linear obteve o melhor desempenho em relação aos demais, assim aplicando *S-Fold Cross validation* para averiguar a melhor acurácia, tendo o 5-fold Cross como melhor resultado.

Palavras-chave: Crime. *Dataset*. Classificação.

ABSTRACT

The present work aims to construct a database for performing experiments of analysis and prediction of criminal patterns from data of the web. In particular, we collect, filter and characterize data related to crimes on the social network Twitter. In addition, we perform experiments with users for labeling database data so that they can be effectively used in automatic sorting experiments.

To evaluate the quality in the database, we performed a classification experiment. There is a validation study carried out in each of the kernels, where the classification efficiency is compared. Two classes were used to train the classifiers, and the performance of trained classifiers with data per user and the joint database was evaluated. It was possible to evaluate that the Linear kernel obtained the best performance in relation to the others, thus applying S-Fold Cross validation to verify the best accuracy, with 5-fold Cross as the best result.

Keywords: Crime. Dataset. Classification.

LISTA DE FIGURAS

FIGURA 1 – Estrutura do Projeto 14
FIGURA 2 – Website Twitter 16
FIGURA 3 – Hiperplano 18
FIGURA 4 – Hiperplano Ideal e Margem máxima 19
FIGURA 5 – Representação da margem rígida 20
FIGURA 6 – Representação margem suave 21
FIGURA 7 – Dimensional 22
FIGURA 8 – Validação Cruzada 23
FIGURA 9 – Metodologia modelo Cascata 26
FIGURA 10 – Metodologia do Projeto 27
FIGURA 11 – Diagrama E-R 28
FIGURA 12 – Gerar chave twitter 29
FIGURA 13 – Votação 31
FIGURA 14 – Periodicidade de relado 37
FIGURA 15 – Relatos de crime por usuário 37
FIGURA 16 – Classes de crime 38
FIGURA 17 – Análise Crion 39
FIGURA 18 – Ackernel 42
FIGURA 19 – Resultado grid.py 43

LISTA DE TABELAS

TABELA 1 – Funções Kernel LIBSVM 22
TABELA 2 – Matriz de confusão 23
TABELA 3 – Entidades CRION 27
TABELA 4 – Votação 31
TABELA 5 – Amostras após votação 32
TABELA 6 – Quantidade crime e não crime na classe fácil 32
TABELA 7 – Termos de coleta 33
TABELA 8 – Frequência do termo 35
TABELA 9 – Matriz de confusão (80% base) 44
TABELA 10 – Matriz de confusão (60% base) 44
TABELA 11 – Matriz de confusão (40% base) 45
TABELA 12 – Matriz de confusão (20% base) 45

LISTA DE SIGLAS

SVM *Máquina de Vetores de Suporte*

API *Application Programming Interface*

KDT *Knowledge Discovery in Text*

TF *Term Frequency*

IDF *Inverse Document Frequency*

MOOCs *Massive Open Online Courses*

CONTEÚDO

1 INTRODUÇÃO **12**

1.1 Motivação 12

1.2 Objetivos 13

1.2.1 Objetivo Geral 13

1.2.2 Objetivos Específicos 13

1.3 Descrição resumida do Projeto 13

1.4 Estrutura do trabalho 14

2 REVISÃO BIBLIOGRÁFICA **15**

2.1 Redes Sociais 15

2.2 Crime e Criminalidade 16

2.3 Aprendizado de Máquina 17

2.3.1 Máquina de vetores de Suporte 17

2.3.2 Classificação Linear 18

2.3.3 Margens Rígidas 19

2.3.4 Margens Suaves 20

2.3.5 Kernel SVM 21

2.3.6 Validação cruzada 22

2.3.7 Medidas de Performance 23

2.4 Trabalhos Relacionados 24

3 CRION DATASET **26**

3.1 Metodologia da etapa Projeto 26

3.2 Estrutura CRION 27

3.3 Coletas de *tweet* 28

3.3.1 Uso da Application Programming Interface (API) do Twitter 28

3.4 Banco de Dados Inicial 30

3.4.1 Votação 30

4 CARACTERIZAÇÃO DA BASE DE DADOS 33
4.1 Termos para Coleta 33
4.2 Região a ser estudada 34
4.3 Pré-Classificação dos textos 34
4.3.1 Tokenização 34
4.3.2 Remoção de stopword 34
4.3.3 Modelo de Espaço de Vetores 34
4.4 Distribuição de postagens por usuários 36
4.4.1 Periodicidade de relato 36
4.4.2 Números de relatos de crimes 37
4.5 Caracterização de crimes 37
4.6 Protótipo de Aplicação Crion 39
4.7 Técnica de Classificação 39

5 ANÁLISES E RESULTADOS 41
5.1 Testes em Kernels Distintos 41
5.2 Refinamento de parâmetro 42
5.3 Avaliação classificação SVM Linear 43
5.3.1 Classificação em 80% 44
5.3.2 Classificação em 60% 44
5.3.3 Classificação em 40% 45
5.3.4 Classificação em 20% 45
5.4 Conclusão 45

REFERÊNCIAS 47

1 INTRODUÇÃO

É perceptível o crescimento das tecnologias e sobretudo da *Internet* nos últimos anos, atualmente a mesma encontra-se presente em todos os contextos sociais, desde uma simples busca até mesmo grandes pesquisas. É inegável o uso cada vez mais frequente dessa tecnologia que, por consequência modifica a forma com a qual as pessoas se relacionam e realizam suas tarefas.

Essa rápida evolução tecnológica trouxe diversos benefícios, tais como a facilidade de comunicação e informação, porém conforme menciona Moraes (2014) percebe-se um grande número de adultos, jovens e crianças passando a maior parte do tempo conectadas não estabelecendo uma comunicação saudável com o ambiente em que estão inseridos.

Em toda sociedade há conflitos de interesses e não seria diferente nesta nova sociedade, chamada sociedade da informação, isto é, aquela presente no âmbito virtual. Todo este avanço tecnológico trouxe incontáveis benefícios, como exposto acima, contudo esse fácil acesso à *Internet* revela-se um forte instrumento para a ocorrência de delitos, sobretudo nas redes sociais, onde é notória a ocorrência cada vez maior de crimes praticados neste espaço, os chamados crimes virtuais. A *Internet* também abriu espaço para relatos de crimes, onde pessoas informam nas redes sociais o acontecido sem registrarem um boletim de ocorrência.

Devido a grande ocorrência e relatos de crimes nas redes sociais, faz-se necessário a criação de mecanismos capazes de identificá-los de forma ágil e eficaz, delineando assim, dados relativos aos mesmos e extraindo informações para a prevenção de crimes.

À vista disso, este trabalho apresenta o *dataset* de crimes virtuais, para reunir informações da rede social Twitter e fornecer um efetivo modo de predições de crime.

1.1 Motivação

Atualmente, as ocorrências de infrações penais (contravenções penais e crimes) cometidas nas redes sociais, são analisadas e classificadas manualmente por uma equipe de segurança. Por depender de esforço humano, esta análise é suscetível a incontáveis erros, ressalte-se ainda o tempo consumido para classificar a infração para depois tomar as providências cabíveis ao caso concreto.

Todos os dados são informados somente por órgão governamental. A maior motivação desse trabalho é disseminar o conhecimento sobre o relato de crime para todos sem passar por nenhum filtro e/ou terem a chance de não serem relatados.

Esboçado tal plano, classificação automática torna-se um tema atual, relevante e de extrema importância para o relato de infrações penais nas redes sociais. Referido tema fez surgir o presente trabalho uma vez que para o autor, o assunto é de extrema relevância para a sociedade e para conhecimento próprio na construção de um Dataset.

1.2 Objetivos

1.2.1 Objetivo Geral

Criar um *dataset* com registros de crimes relatados no *Twitter*, identificando-os na rede social para uma análise mais detalhada e precisa das autoridades competentes.

1.2.2 Objetivos Específicos

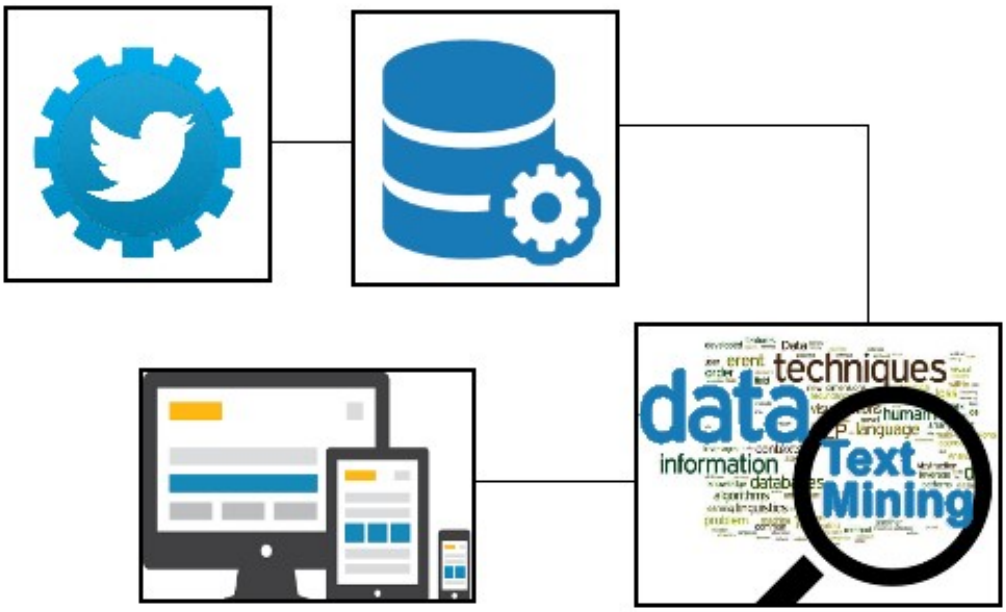
- Obter postagens indicativas de infrações penais, na rede social Twiter, de maneira rápida e eficiente;
- Classificar as postagens;
- Caracterizar as infrações penais, como sua localização e quem o indicou;
- Indicar as regiões bem como as infrações penais que mais ocorrem;
- Disponibilizar as estatísticas e análises de forma clara, concisa e de apelo visual;

1.3 Descrição resumida do Projeto

O presente trabalho, baseia-se na criação de um *dataset* para a análise de crimes na rede social Twitter, com a utilização da tecnologia *Twitter4j*, esta que trata-se de uma biblioteca Java não oficial para o Twitter, e dos algoritmos de classificação de texto *Máquina de Vetores de Suporte* (SVM).

Conforme se verifica na Figura 1, o processo de formação de um *dataset* é composto por algumas etapas. Em linha de princípio as informações (postagens dos usuários) são coletadas com o uso da API *Twitter4j*. Feita a integração da aplicação com o Twitter, as informações coletadas são enviadas para registro diretamente no banco de dados.

Figura 1 – Estrutura do Projeto



Fonte:Desenvolvido pelo autor

Após cumprir-se as etapas de coleta e registro no banco de dados há a etapa de votação dos Tweets coletados, onde 5 experts da área competente determinam a qual classe pertence o Tweet, crime ou não crime.

Após a coleta e registro das informações no banco de dados, a segunda etapa deste processo ocorre por meio da caracterização do *dataset* seguido de um estudo e avaliação do mesmo, aplicando técnica de aprendizagem supervisionada que define critérios específicos para posterior classificação. No presente trabalho, o critério adotado para a classificação das informações obtidas é: "crime" e "não crime". As informações que constam no banco de dados são treinadas para estabelecer um critério e indicar a ocorrência de crimes.

Após a classificação, há uma análise dos resultados obtidos nos testes da classificação. Além de exibir de forma clara e precisa as caracterícticas, estatísticas e análises do *dataset* por meio de um website;

1.4 Estrutura do trabalho

Nesse primeiro Capítulo foram abordado a motivação, objetivos e descrição resumida do projeto, no Capítulo 2 conceitos básicos e revisão bibliográfica, no Capítulo 3 a criação do dataset, no Capítulo 4 a caracterização da base de dados e, no último Capítulo, a análise dos resultados e considerações finais.

2 REVISÃO BIBLIOGRÁFICA

Este Capítulo apresenta algumas tecnologias relacionadas à classificação de textos, bem como uma revisão sobre os conceitos e técnicas necessárias para desenvolvimento do projeto proposto.

2.1 Redes Sociais

As redes sociais figuram como ambientes de divulgação, combinação de interesses, central de informações e conhecimentos. Desde a sua origem houve um crescimento do número de usuários, diferenciados pelos seus perfis, interesses, temas, entre outros. Essa diferenciação de interesses possibilitou o surgimento de diferentes redes: as redes sociais, as profissionais e as acadêmicas Furtado (2017), viabilizando a troca de informações e o aumento do círculos de relacionamentos sociais.

Há fatores positivos e negativos para o uso de rede social. Observa-se que o uso dessa tecnologia de uma forma correta contribui para a conhecimento social, difundindo e compartilhando conhecimento de modo que pontos colocados em questão são discutidos ou promovidos. No entanto, estudam-se os efeitos prejudiciais que o uso indiscriminado dos diversos tipos de rede social pode ocasionar, tais como, a superexposição das crianças Almeida (2016) e a dependência da internet YOUNG (2011).

Os números de usuários de rede sociais no Brasil chega a ser assustador. O brasileiro passa em média 3h14 por dia conectado com o celular, segundo Amaral (2016), sendo as redes sociais mais usadas: Facebook, WhatsApp, Youtube, Instagram, Skype e Twitter, passando dos 100 milhoes de usuários Kemp (2016).

Dentre elas, o Twitter tornou-se uma poderosa ferramenta de comunicação para vários fins, como para busca de informação e disseminação do conhecimento. Ao redor do mundo são mais de 310 milhões de usuários únicos, tendo 83% dos líderes políticos mundiais presentes na rede com contas ativas. A Figura 2 ilustra a página inicial do Twitter, rede social escolhida para a construção do *dataset*.

Figura 2 – Website Twitter

Fonte: Website *twitter.com*

2.2 Crime e Criminalidade

Apesar dos incontáveis benefícios trazidos pelo desenvolvimento da tecnologia da comunicação, incluindo as redes sociais, como citado na sessão anterior, o mau uso dessa tecnologia é capaz de causar diversos tipos de danos aos usuários. Neste ponto, entende-se dano no seu amplo sentido, compreendendo até mesmo a ocorrência de crimes. O Direito é responsável por regular as interações sociais e tem o dever de acompanhar os avanços tecnológicos de modo a proteger os direitos e os interesses do indivíduo bem como da sociedade. O Direito Penal é a parte do ordenamento jurídico responsável por estabelecer normas definidoras de infrações penais bem como as sanções correspondentes. Segundo Cezar Roberto Bitencourt:

> Uma das principais características do moderno Direito Penal é o seu caráter fragmentário, no sentido de que representa a última ratio do sistema para a proteção daqueles bens e interesses de maior importância para o indivíduo e a sociedade à qual pertence BITENCOURT (2013).

Os crimes estão caracterizados no Código Penal (Decreto-Lei n. 2.848, de 07 de dezembro de 1940) e são considerados como uma conduta contrária à Lei penal. Pode-se ressaltar ainda que, apesar dos crimes ocorrerem em um ambiente virtual, eles são crimes reais, descritos no Código Penal e devem ser punidos de acordo com a previsão legal. Conforme levantamentos, determinados crimes possuem incidência frequente na textitInternet, a saber: calúnia, difamação, injúria, falsa identidade, preconceito, iscriminação, pedofilia, dentre outros. Além dos frequêntes relatos dos crimes que não

são cometidos na *Internet*, como roubo, assalto e homicídio. Para uma apuração ágil e eficaz da ocorrência desses crimes e/ou até mesmo na atuação preventiva para que o mesmo não ocorra, é necessária a promoção de uma investigação mais efetiva, o que pode ocorrer com o auxílio da tecnologia de mineração de dados. A utilização de tal ferramenta pode tornar o espaço virtual cada vez mais seguro e garantir o bem estar social bem como a efetiva segurança pública neste meio.

2.3 Aprendizado de Máquina

Com milhões de usuários usando o Twitter simultâneamente, há um grande volume de informações que são armazenadas, onde muitas informações valiosas podem estar implícitas nas bases de dados. Muitas técnicas foram estudadas e desenvolvidas com o objetivo de auxiliar na extração de informações Lago (2008). A Mineração de dados ou *Data Mining* é o processo de exploração de grandes quantidades de dados com o objetivo de encontrar padrão, irregular ou não, e analogia para suportar a tomada de decisões.

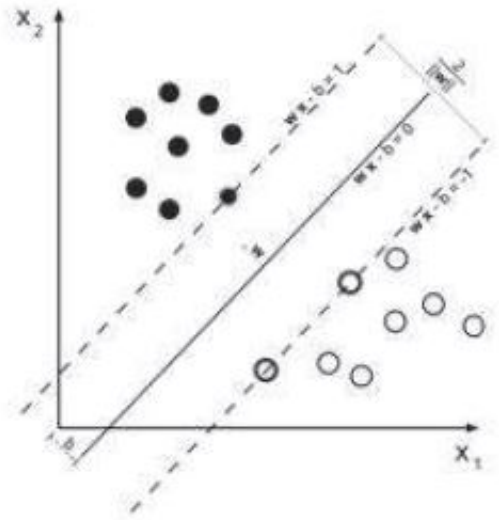
Pelo fato de muitas informações estarem armazenadas em forma de texto Lago (2008), o *Knowledge Discovery in Text* (KDT) combina técnicas e conhecimentos de diversos segmentos como Informática, Estatística, Linguística, Matemática e outras, tendo a capacidade de extrair conhecimento a partir de grandes coleções de texto Silva (2013). Nesse projeto, foi adotada a técnica de Rastreamento de Tópicos e foram definidos tópicos de acordo com estatísticas e estudos descritos no Capítulo 3.

2.3.1 Máquina de vetores de Suporte

SVM é um algoritmo de aprendizado de máquina para classificação binária que tem sido usado para classificação e análise de regressão. Utiliza um conjunto de métodos de aprendizado supervisionado para reconhecer padrões em dados disponibilizados.

O SVM trabalha com dados númericos, ou seja, todo e qualquer dado como textos são obrigatóriamente convertidos em formato númerico. Como explicitado, para o SVM, a entrada é um conjunto de vetores representados por característicase, onde sua saída normalmente são separadas por duas classes: positivo ou negativo. Cada vetor é representado no espaço e, no treinamento do SVM, o mesmo busca o hiperplano definindo a máxima distância Euclidiana entre essas duas classes (Figura 3). Como dito o SVM não consegue processar textos. Para a camada adicional de representação é criado um dicionário de palavras com todas as palavras existentes em todos os documentos. Então cada palavra é associada a uma coordenada do vetor.

Figura 3 – Hiperplano com a Máxima Distância



Fonte: Website *Google images*

O SVM tem a a vantagem de utilizar a proteção contra *overfitting*, ou seja, se ajusta muito bem ao conjunto de dados anteriormente observado, além de trabalhar bem em espaços de grande dimensão o que facilita para o presente trabalho visto a imprevisão de termos a serem trabalhados. Acrescenta-se que o SVM também trata a maioria dos problemas de caegorização de texto de forma linear. JOACHIMS (1998) ressalva essas vantagens do uso de SVMs para classificação textual.

2.3.2 Classificação Linear

Para melhor entendimento das máquinas de vetores de suporte, é necessário entender um pouco da teoria dos classificadores lineares. Considere o caso em que $f(x)$ é uma função linear de $x \in X$, onde a entrada x é atribuída a uma classe representada por 0 ou 1. Essa relação pode ser escrita na forma da Equação:

$$f(x) = \sum_{i=1}^{n} w_i x_i + b \tag{2.1}$$

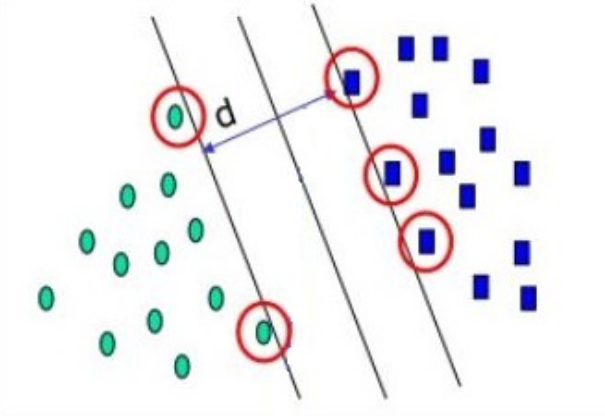
Onde (w, b) são parâmetros que controlam a função e a regra de decisão é dadapela função de atribuição $sgn(f(x))$. Os parâmetros para a regra de decisão são baseados em dados de treinamento Cristianini (2000). Essa hipótese apresentada é interpretada como um espaço de entrada X onde é dividido nas duas classes no hiperplano definido na equação apresentada.

Para o classificador o SVM procura classificar os registros de uma forma que as classes permaneçam o mais longe possível uma da outra, ou seja, maximizando a

capacidade de generalização, definindo a margem para que o plano ótimo maximize a distancia entre as duas classes. A margem é definida pelo dobro da distância entre os vetores de suporte, sendo os pontos mais próximos das diferentes classes, e o hiperplano.

Considera-se como margem o dobro do valor da distância entre o hiperplano e o ponto do conjunto de treinamento mais próximo, sendo que estes pontos são chamados de vetores de suporte. O hiperplano com maior margem de separação tem melhor capacidade de generalização pois diminui a possibilidade de erro. A Figura 4 ilustra o hiperplano ideal com margem máxima, destacando os vetores de suporte.

Figura 4 – Hiperplano Ideal e Margem máxima



Fonte: Website *Google images*

2.3.3 Margens Rígidas

Suponha que um hiperplano pode ser definido pela Equação a seguir:

$$H(x) = wx + b \tag{2.2}$$

Onde w é um vetor n-dimensional (n é o número de características), b é o bias. Considera-se, agora, os planos H_1 e H_2:

$$H_1 : wx_i + b = 1 \tag{2.3}$$

$$H_2 : wx_i + b = -1 \tag{2.4}$$

A distância $p(x)$ entre um ponto x do plano H_1 a H deve ser a mesma que entre um ponto de H_2 a H, o que define as margens. Uma vez que w e b foram escalados de forma a não haver exemplos entre H_1 e H_2, temos a distância mínima entre o hiperplano separador e os dados de treinamento, dado por:

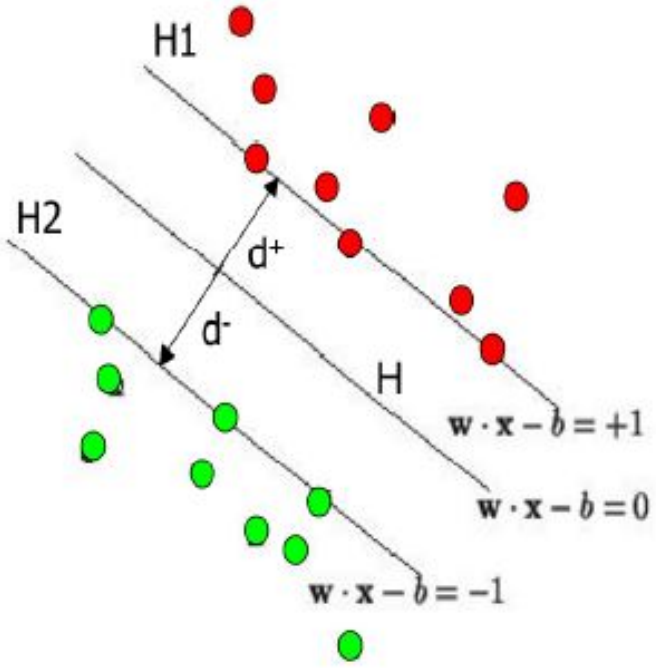
$$minimizar : Q(w, b) = \frac{1}{2}||w||^2 \tag{2.5}$$

E para assegurar que não haja dados de treinamento entre os vetores de suporte incluem-se as restrições:

$$y_i(wx_i + b) - 1 \geq 0, \forall_i = 1, ..., n \tag{2.6}$$

Por se tratar de um problema de programação quadrática, Abe (2010) suegere que o quadrado da norma Euclidiana $\|w\|^1$ na equação de distância mínima converte o problema em otimização quadrática, ou seja, procura determinar os valores extremos da função. A hipótese de separabilidade linear supõe que existam w e b que satisfaçam a restrição definida na Inequação. A Figura 5 ilustra a representação da Margem rígida.

Figura 5 – Representação da margem rígida

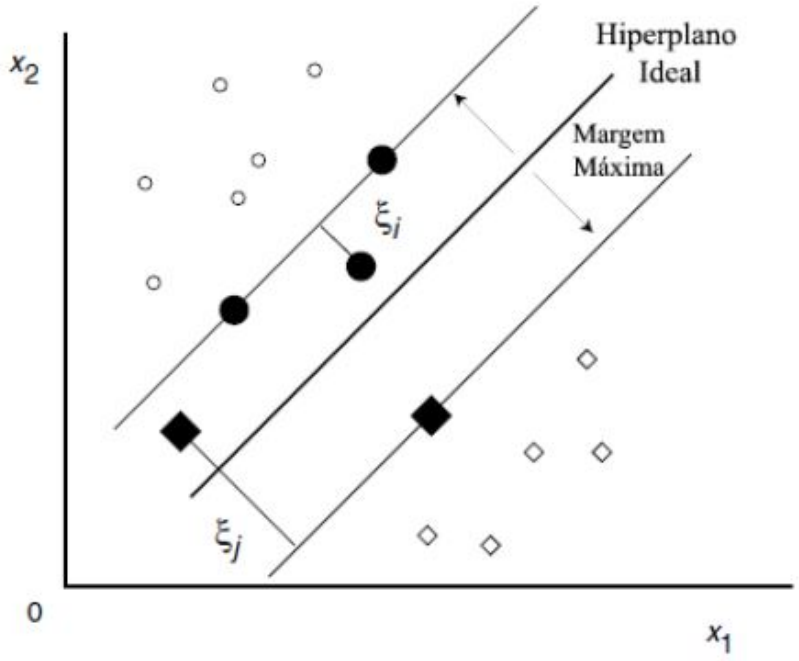


Fonte: Santos (2013)

2.3.4 Margens Suaves

As margens suaves aplicam uma variável de folga, para solucionar o problema se separação linear, quando os dados do conjunto de treinamento não são linearmente separáveis, diferentemente da margem rígida, que supõe que os dados de treinamento são linearmente separáveis. Considerando a variável de folga como ζ_i, observa-se na Figura 6 que para alguns treinamentos o classificador não possui margem máxima, mas é classificado de forma correta. Porém se a variável de folga fugir da margem de $\zeta_i \geq 1$ o dado é classificado de forma incorreta pelo hiperplano ideal.

Figura 6 – Representação margem suave



Fonte: Reis (2016)

Para resolver este problema de otimização, tem-se que:

$$minimizar : Q(w,b) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\zeta_i \tag{2.7}$$

$$y_i(wx_i + b) - 1 \geq -\zeta_i, \forall_i = 1, ..., n \tag{2.8}$$

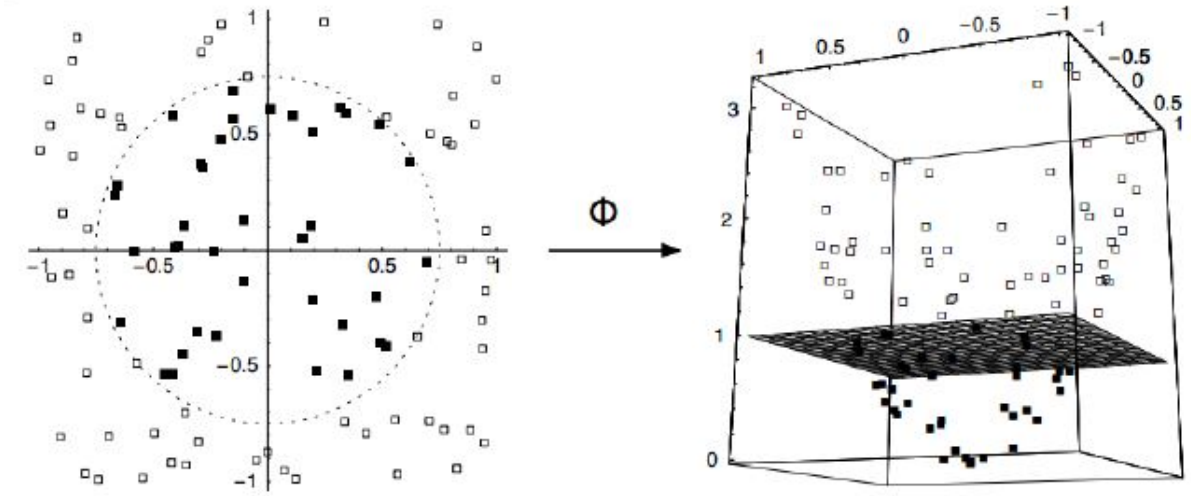
Em que $\zeta_i = (1, ..., n)$ e C é o parâmetro que determina a maximização da margem em contrapartida da minimização do erro de classificação Reis (2016).

2.3.5 Kernel SVM

O conjunto de treinamentos de textos é de difícil treinamento porque possui um conjunto de dados linearmente não separávais, o classificador pode apresentar baixa capacidade de generalização mesmo com o hiperplano ideal, o hiperplano ideal é determinado para maximizar a capacidade de generalização. Assim, para aumentar a separabilidade linear o espaço é mapeado em um espaço de maior dimensão, chamado de espaço característico.

A Figura 7 mostra a transformação de um conjunto de dados não linearmente separáveis em uma espaço bidimensional para um espaço tridimensional onde o conjunto de dados pode ser linearmente separável.

Figura 7 – Transformação dimensional



Fonte: Sullivan (2007)

Para a utilização do SVM foi utilizada a biblioteca LIBSVM, que oferece funções kernel dísponíveis na sua biblioteca, descritos na Tabela 1. Neste trabalho, foram utilizadas as funções Kernel Linear - $x_i * x_j$.

Tabela 1 – Funções Kernel LIBSVM

Tipo	Função Kernel	Parâmetros
Linear	$x_i * x_j$	Não há
Polinomial	$(\gamma * x_i * x_j + k)^d$	γ, d, k
Base Radial (RBF)	$\exp(-\gamma * \lvert x_i * x_j \rvert^2)$	γ
Sigmoidal	$\tan(\gamma * x_i * x_j + k)$	γ e k

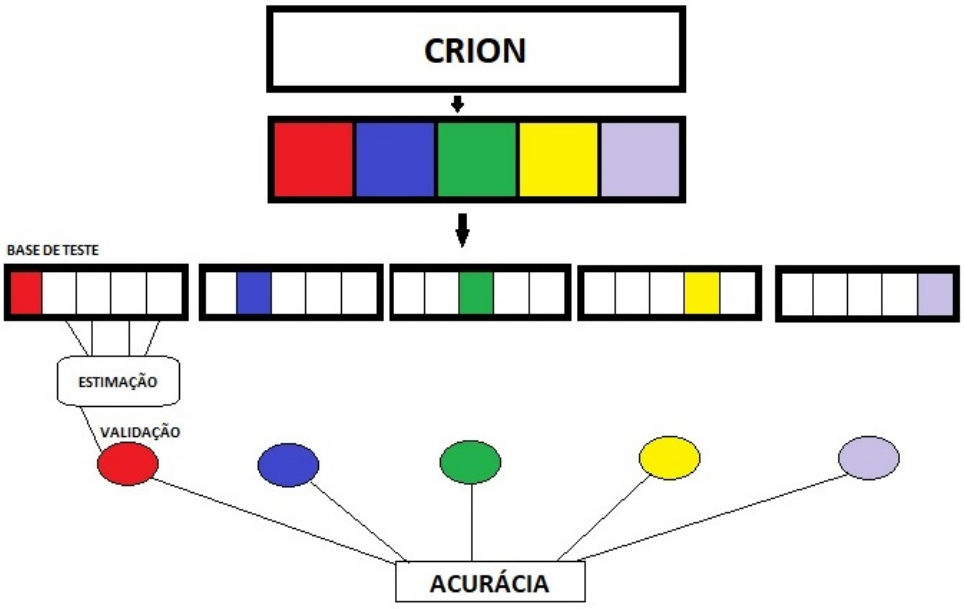
2.3.6 Validação cruzada

A validação cruzada é uma técnica para avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados KOHAVI (1995). Uma técnica comum em modelagem para predição. Do total de *tweets*, foram retirados os primeiros 20000 posts para o processo da validação cruzada e assim definir o desempenho do treinamento do CRION.

Para esse procedimento foi realizado o método k-fold, um método bastante usado na literatura, que consiste em dividir o conjunto total de dados em k subconjuntos do mesmo tamanho e, a partir disto, um subconjunto é utilizado para teste e os $k - 1$ restantes são utilizados para estimação dos parâmetros e resultando na acurácia do modelo. Para esse modelo deve-se realizar o processo k vezes alternando os sub-

conjuuntos de teste. A Figura 8 ilustra o esquema realizado pelo $k - fold$, para esse trabalho foi definido como $k = 5$.

Figura 8 – Validação Cruzada



Fonte: Desenvolvido pelo autos

Para a avaliação de performance de um classificador, resultado da acurácia, temos a porcentagem de predições corretas pelo total de predições. Como descrito na equação a seguir:

$$Acuracia = \frac{n^o.de.predicoes.corretas}{Total.de.amostra.de.teste} x100\% \tag{2.9}$$

2.3.7 Medidas de Performance

Essa constitui a fase de verificação da qualidade da classificação feita, tendo o número de falso positivos obtidos duante o experimento. No primeiro processo de votação, ao ser avaliado por cinco pessoas, foram considerados os 8.672 *tweets*, sendo eles listados na Tabela de Matriz de confusão(Tabela 2), na primeira instância com 60% de acerto e ao passar pela validação cruzada obteve-se uma acurácia de 68,25%.

Tabela 2 – Matriz de confusão

	tweets		
	Positivo	Negativo	Total
Verdadeiro	1.466	662	2.128
Falso	4.504	2040	6.544

A partir da matriz de confusão, pode-se calcular medidas de desempenho da classificação, onde:

$$F - score = \frac{2VP}{(2VP + FP + FN)} \tag{2.10}$$

Onde, VP = Verdadeiro posisitivo, FP = Falso positivo e FN = Falso negativo. Quanto maior o valor do F-score, delimitado entre 0 e 1, maior a capacidade de generalização do classificador.

A medida F-score de certa forma é mais informativa que a acurácia, pois considera as medidas de precisão e revocação (também conhecida como sensibilidade) para avaliar o desempenho de um classificador.

2.4 Trabalhos Relacionados

Santos (2013) ressalta que a tarefa de classificação textual é uma tarefa difícil por ter problemas com o dinamismo da linguagem informal, em que tais postagens podem não denotar uma ocorrência de crime. Mesmo com o uso de técnicas que avaliem a viabilidade do texto com o contexto de criminalidade, a autora concluiu a dificuldade na tentativa de definir a relação de crime ou não crime. Para isso, fez-se uso das ferramentas SVM, Árvore de decisão e Naive bayes. Concluiu-se que o melhor classificador de texto corresponde ao SVM Linear com uso conjunto de PCA que utiliza o número de componentes para explicar 80% da variabilidade total dos dados. Esse classificador teve medida F-score de 0,678.

Em Marani (2016) foi criado um conjunto de dados a partir do rastreamento de dados em *Massive Open Online Courses* (MOOCs) hospedados no *site* Coursera, plataforma de cursos *online*. O artigo DAJEE contém mais de 20.000 recursos, para o rastreamento de outras Plataformas MOOCs. DAJEE fornece dados sobre lições, cursos e instrutores do tempo que os recursos foram entregues. Portanto, as técnicas de mineração foram empregadas para extrair padrões de ensino e as preferências dos instrutores, analisando os recursos e cursos.

Reis (2016) avaliou técnicas utilizadas na literatura para o reconhecimento de gestos, por meio de informações inerciais e eletromiográficas. Foram avaliadas 6 classes (repouso, mão fechada, mão aberta, mão para dentro, mão para fora e duplo clique) em modalidades diferentes. Os resultados foram obtidos pela técnica de SVM, com a melhor modalidade apresentando um resultado de acurácia com cerca de 80, 64% por usuário.

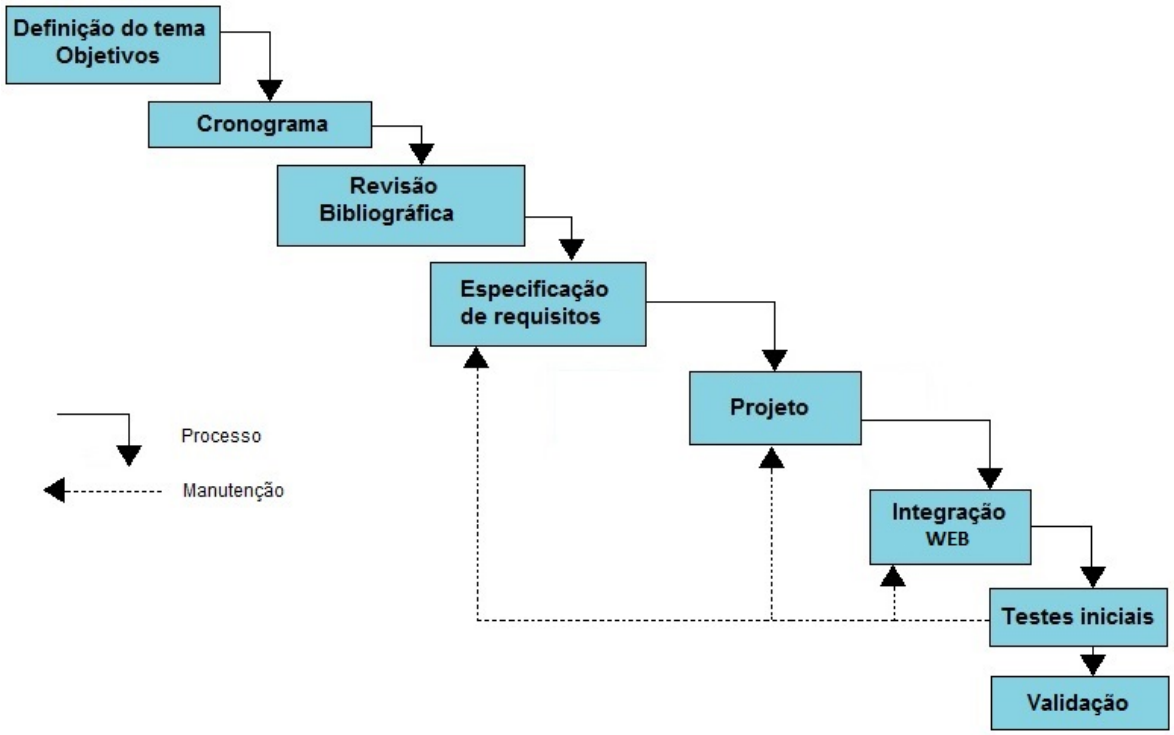
Piczak (2015) cria um *dataset* de som ambiental, onde faz a classificação do som

ambiental de um conjunto de dados publicamente disponíveis. O autor aborda essa questão apresentando uma coleção anotada de 2 000 clipes curtos compreendendo 50 classes de vários sons comuns e uma compilação unificada de 250 000 trechos auditivos não gravados extraídos das gravações. Fornece a avaliação da precisão humana na classificação de sons ambientais, comparando ao desempenho de classificadores SVM, K-NN e Árvore aleatória.

3 CRION DATASET

Neste capítulo serão abordados os métodos e os processos realizados para atingir os objetivos propostos no capítulo 1, ou seja, criar um *dataset* para analises e predição de crimes com dados obtidos da rede social *Twitter*. A Figura 9 demonstra a metodologia adotada.

Figura 9 – Metodologia modelo Cascata



Fonte: Desenvolvida pelo autor

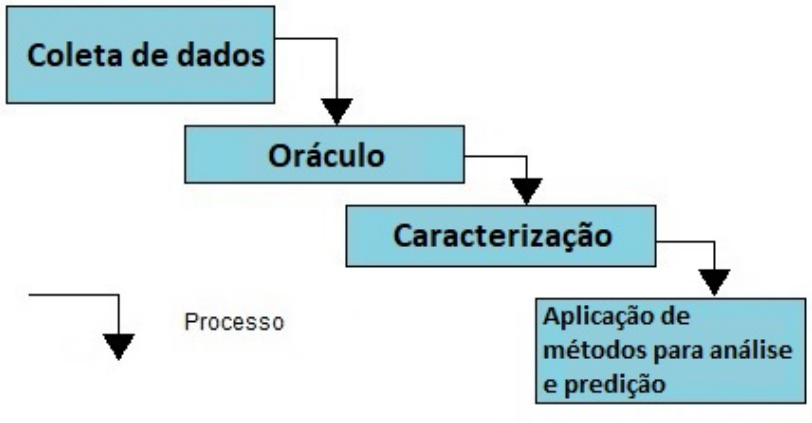
Diante do tema e dos objetivos apresentados foram estipulados as etapas para a entrega final deste projeto de monografia. Em seguida, foi realizada a revisão bibliográfica descrita no capítulo 2 e por sua vez entrando no processo de requisitos, desenvolvimento e testes que serão descritos no dercorrer das seções e capítulos posteriores.

3.1 Metodologia da etapa Projeto

A etapa projeto é destrinchado em quatro fases, como ilustrado na Figura 10. A primeira fase constituí na coleta das informações, em seguida a aplicação do orá-

culo para maior precisão dos dados, e por fim caracterizando o *dataset*. Assim, há a possibilidade de aplicação de métodos para análise e predição em uma base de dados consolidada e caracterizada.

Figura 10 – Metodologia do Projeto



Fonte: Desenvolvida pelo autor

3.2 Estrutura CRION

CRION é um banco de dados MySql (DB) de baixo nível de normalização por ser apenas para fins de consultoria, hospedando os dados coletados e processados nas entidades relatadas na Tabela 1 e descritas a seguir. Para uma melhor compreensão, a Figura 11 mostra a Diagrama E-R do DB que reporta todas as entidades do dataset.

Tabela 3 – Entidades CRION

Entidade	Descrição.
usuario	Usuários CRION
tweet	*tweets* coletados
votacao	*tweets* para votação
stop*tweet*	*tweets* sem Stop Words
treina*tweet*	*tweets* de treinamento
crime*tweet*	*tweets* classificados

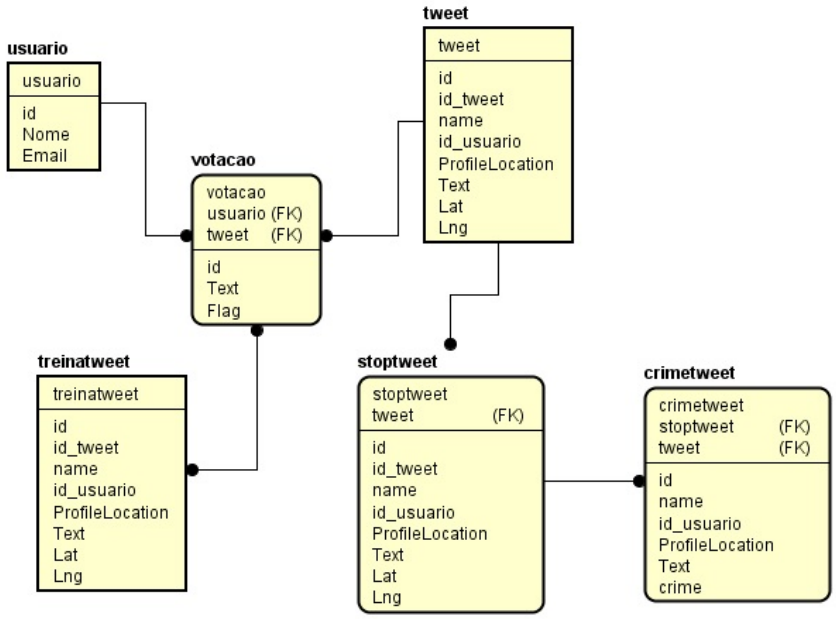
A entidade *tweet* trabalha como uma entidade mãe, contendo os atributos: id, id do *tweet*, *name*, id do usuário, *ProfileLocation* como o local onde o *Profile* está associado, *Text* como o conteúdo o post do usuário, latitude e longitude relativo a cada post coletado.

A entidade *votacao* está relacionada ao processo de votação do post do usuário com o intuito de rotular o dado como crime ou não crime e assim produzir-se a base de

treinamento, detalhada na subseção 3.2. Esse procedimento é permitido somente para usuários cadastrados na entidiade usuario.

A entidade *stoptweet*, que contém o resultado da primeira etapa da classificação, é similar a entidade *tweet*, porém diferenciada pelo atributo Text onde o valor nada mais é que o post com a remoção dos stopwords. Na entidade *treinatweet* está armazenado todo o conteúdo para o processo de treinamento e para o último passo do sistema. A entidade *crimetweet* armazenará todos o *tweets* que tiveram uma classificação suscetível de ser um relato de crime, adotado o critério de 60% em relação ao treinamento.

Figura 11 – Diagrama E-R



Fonte: Desenvolvido pelo autor

3.3 Coletas de *tweet*

O Twitter permite a coleta de postagens realizadas em seu ambiente, podendo ser definidas por região e/ou de termos ou expressões que caracterizem o contexto desejado. A API utilizada no presente trabalho é a *Twitter4J*, uma biblioteca Java não oficial desenvolvida de forma *open source*, onde pode-se integrar a aplicação com o serviço Twitter.

3.3.1 Uso da API do Twitter

Existem várias propriedades disponíveis para configurar o *Twitter4J*, foi utilizada a classe *ConfigurationBuilder* que segundo Sirois (2012) é útil para usuários que desejam realizar testes de unidade, nesse caso cada unidade de coleta representa um

post coletado.

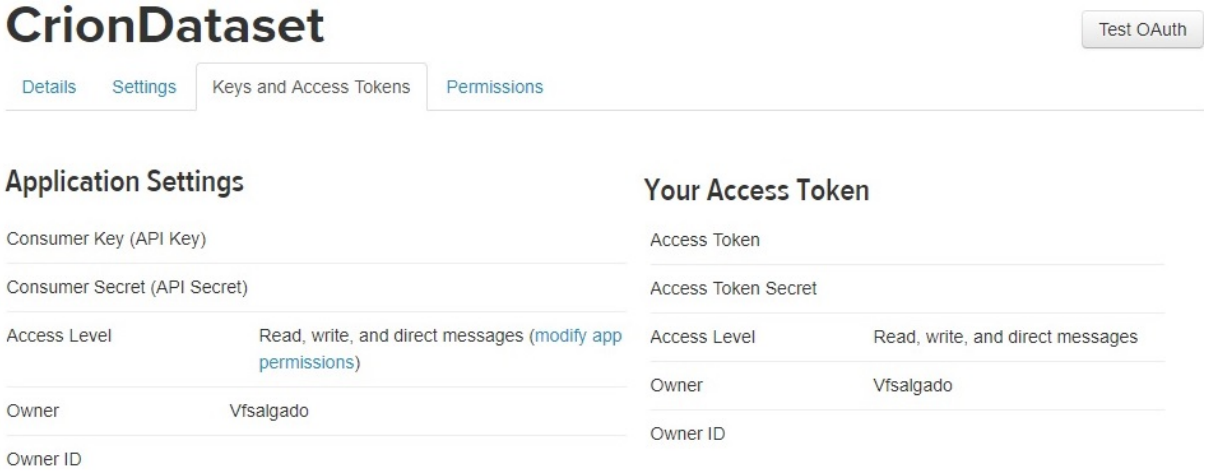
Para determinada rotina o sistema deve seguir os padrões estabelecidos, o Twitter estabelece limites para taxa de requisição, onde são divididos em intervalos de 15 minutos e com limite máximo 180 chamadas, sendo que cada chamada pode voltar, no máximo 100 *tweets* por página com as características especificadas. Para ter esse controle o Twitter exige a identificação do usuário. Na prática, durante o processo, cada pedido de coleta ao Twitter é registrado na conta do aplicativo (*developer.twitter*) do usuário de modo que a rede social tem mais controle sobre o mesmo, registrando os *logs* de data e quantos *tweets* estão sendo repassados.

O código abaixo representa o a configuração da API para instanciar as chaves de acesso para autenticação do Twitter.

```
1 ConfigurationBuilder cb = new ConfigurationBuilder();
2 cb.setDebugEnabled(true)
3     .setOAuthConsumerKey("**********************")
4     .setOAuthConsumerSecret("******************")
5     .setOAuthAccessToken("**********************")
6     .setOAuthAccessTokenSecret("***************");
7 TwitterFactory tf = new TwitterFactory(cb.build());
8 Twitter twitter = tf.getInstance();
```

A Figura 12 exibe a interface da aplicação de desenvolvedor do Twitter onde são geradas as chaves de acesso e consumo de dados da rede social. Para ter acesso o usuário deve ter uma conta associada ao Twitter.

Figura 12 – Gerar chave de acesso dotwitter



Fonte: Twitter Developer

Para delimitar a coleta de *tweets* utiliza-se a classe *FilterQuery*, que oferece opções de filtros como usuário, localização e até limite que os *tweets* aparecem no fluxo da *timeline*. Para este trabalho, foram delimitadas palavras-chave de pesquisa, detalhadas

no capítulo 4, e todos os *tweets* que estão em português.

O cógido abaixo demonstra a pesquisa aplicada ao código desenvolvido:

```
1 FilterQuery fq = new FilterQuery();
2     fq.track(new String[]{"Palavras-chave para pesquisa"});
3     fq.language(new String[]{"pt"});
4 twitterStream.filter(fq);
```

3.4 Banco de Dados Inicial

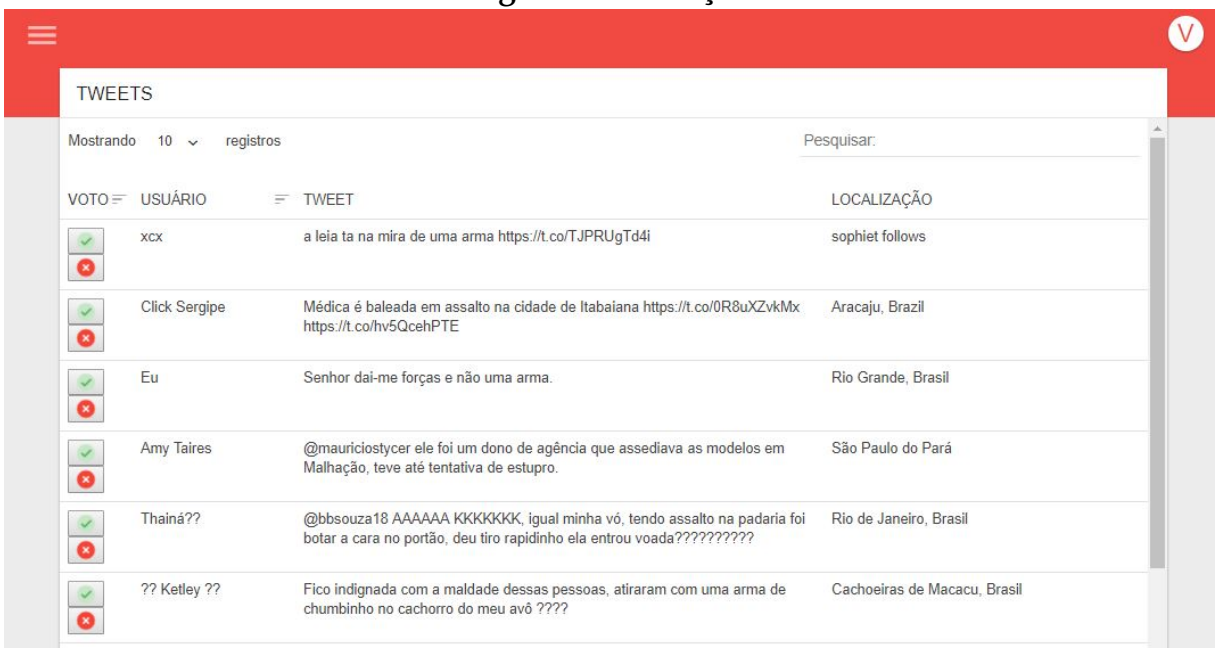
A ideia foi obter amostras iniciais de postagens com relatos de crime que dariam origem ao primeiro banco de treinamento e teste. A partir dessas primeiras coletas teve-se uma percepção das dificuldades encontradas e como melhorar a coleta.

O banco de dados inicial desse trabalho foi coletado durante, aproximadamente, sete dias, começando pelo dia 8 de outubro de 2017. Ao final dessas coletas iniciais obteve-se 98.358 *tweets* e após aplicação de filtros o número caiu para 42289. Foram desconsiderados todos os *retweets* por serem informações duplicadas, o que podem prejudicar a classificação, e também todos os *tweets* que continham os textos 100% iguais.

3.4.1 Votação

Para a predição se tornar mais precisa, foi utilizado como primeira etapa a votação inicial, processo de decisão no qual os usuários expressam a sua opinião por meio de um voto, determinando se o *tweet* coletado é relato de um crime ou não. Como a quantidade de *tweets* não crime era muito maior, foi adotado uma estratégia de balanceamento da base de dados, no qual foram considerados 20.000 *tweets* aleatórios que foram listados para que cinco usuários do CRION expressassem a sua opinião. A Figura 13 ilustra o processo de votação do sistema CRION.

Figura 13 – Votação



Fonte: Sistema CRION

Dessa forma contabilizaremos cinco votos para cada *tweet* e assim determinar quais irão para o processo de treinamento. A Tabela 4 explicita todas as possibilidades de resultado após o procedimento, indicando o número de votos e seu resultado de decisão.

Tabela 4 – Votação

Crime	Não crime	Decisão
5	0	Fácil
4	1	Média
3	2	Difícil
2	3	Difícil
1	4	Média
0	5	Fácil

Foram considerados somente os *tweets* que estavam na classe de decisão *Fácil*, por ser de decisão unânime e assim evitar qualquer conflito. Após esse procedimento obteve-se o resultado de 8.672 *tweets*.

A Tabela 5 , mostra a quantidade de amostras capturadas para o treinamento após o critério de votação e a Tabela 6 quantidade de amostras definidas como crime ou não crime na classe Fácil.

Tabela 5 – Amostras após votação

	Total	Difícil	Média	Fácil
Quantidade	20.000	4.192	7.136	8.672

Tabela 6 – Quantidade crime e não crime na classe fácil

	Total	Crime	Não crime
Quantidade	8.672	2.128	6.544

4 CARACTERIZAÇÃO DA BASE DE DADOS

Como descrito no capítulo anterior, o Twitter permite a coleta de postagens para serem utilizadas pelos desenvolvedores. As etapas de coleta serão abordadas neste capítulo apresentando a definição dos termos e região que caracterizam o contexto desejado e o processo de classificação do posts coletados para o dataset CRION.

4.1 Termos para Coleta

Parte central das discussões do trabalho dá-se na definição de quais termos servem à coleta de *tweets* de relatos de crimes. Na fase inicial do trabalho foram definidas algumas palavras comumente associadas a crimes. Além dos relatos de crime como furto, roubo e assalto, foram escolhidos termos associados aos crimes de:

- Calúnia: inventar histórias falsas sobre alguém;
- Preconceito ou discriminação: fazer comentários de forma negativa sobre religião, etnias, raças, etc;
- Apologia ao crime: criar comunidades que ensinem a burlar normas ou mesmo que divulguem atos ilícitos já realizados;
- Pedofilia: troca de informações e imagens de crianças ou adolescentes.

A polícia divulgou termos e gírias utilizadas por criminosos, encontradas em TERMOS... (2013), nesse trabalho não foram utilizada todas as palavras listadas tendo em vista que esses são jargões mais utilizados em locais como presídios. Porém as palavras selecionadas para o filtro de pesquisa, encontradas na Tabela 7, tem relação aos crimes citados anteriormente.

Tabela 7 – Termos de coleta

arma	assalto	assaltantes	assassinado	homicidios	golpes
sequestrada	assassinatos	pedofilos	assalta	estupro	roubos
pedofilia	preto	tiroteio	apologia	crime	viado
assassinada	arrastao	bicha	corno	pedofilo	

4.2 Região a ser estudada

Seria possível coletar *tweets* com termos associados a crime sem especificar uma dada região. O intuito do trabalho é usar o Twitter como ferramenta para coleta de dados referentes à crime e criminalidade. O interessante é, portanto, acompanhar tais dados o que é facilitado com a definição de uma região mais específica, limitada.

Considerando-se o grande uso do Twitter no Brasil, optou-se por iniciar o estudo em todo território nacional, tendo a informação inicial do local do *Profile* do usuário.

4.3 Pré-Classificação dos textos

4.3.1 Tokenização

Para que um *Tweet* tenha representação em vetor deve-se ler cada palavra individualmente, ou seja, garantir que o computador não leia todo o contexto e o defina como uma única string. Por exemplo, a frase "Médica é baleada em assalto" deve ser representada por "Médica","é","baleada", "em", "assalto".

4.3.2 Remoção de stopword

A idéia é a remoção de palavras como "de", "a", "aquela", "também", "está", "tiveram", já que segundo ZHU (2010) as palavras mais frequentes usualmente não trazem muito sentido ao texto. As *stop-words* são dispensáveis cabendo a cada *tweet* apurado a definição de termos comuns, onde não são considerados úteis para a classificação.

4.3.3 Modelo de Espaço de Vetores

A determinação de um modelo é um passo importante no pré- processamento de um texto, onde há a tradução da informação em linguagem de máquina. A representação utilizada nesse trabalho é a *bag of words* onde demarca quais são as palavras incluídas no texto e quantas vezes cada uma ocorre. Considerando o modelo de espaço de vetores, cada *tweet* é representado por um vetor de palavras e cada palavra considerada um atributo.

Inicia-se o processo ponderando cada atributo de acordo com a sua frequência. Considere três possíveis *tweets*:

1. Médica é baleada em assalto.

2. Manteve em cativeiro e abusou da vítima.

3. Vereador é preso por porte ilegal de arma.

Seguindo às práticas descritas retiram-se artigos, preposições, acentuações, pontuações e colocando-se todas as palavras em minúculo, com o resultado de:

1. medica baleada assalto

2. cativeiro abusou vitima

3. preso ilegal arma

Não há a exclusão de termos com frequência inferior a *um*, porque cada post é limitado 177 caracteres e poeria ser excluido o relato individual de um crime.

Tem-se então: 3 *tweets* e 12 diferentes termos. Para representarmos cada um desses atributos na forma vetorial temos que colocar como entrada do vetor o número de vezes que cada palavra ocorreu em cada *tweet*. Na Tabela 8, representado por uma matriz, a linha corresponde aos *tweets* e as colunas são nomeadas com a palavra e a ocorrência ou não da mesma na coleção de *tweets*.

Tabela 8 – Frequência do termo

medica	baleada	assalto	cativeiro	abusou	vitima	preso	ilegal	arma
1	1	1	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1

Considerando documento como cada tweet, a expressão $f(t, d)$ denota o número de vezes que o termo t ocorre no documento d, ponderando de acordo com a Frequência do Termo (*Term Frequency* (TF)).

O cálculo do peso TF tende a ser o mais intuitivo, mas *tweets* que contenham termos mais frequentes tendem a ser enfatizados no processo de análise. Palavras mais recorrentes informam menos na definição do tweet, justamente por serem mais comuns.

Pensando nisso há outro tipo de peso para os termos coletados. No Inverso da Frequência nos Documentos (*Inverse Document Frequency* (IDF)) o objetivo é valorizar termos que ocorrem com uma menor frequência. É uma medida de quanto o termo

é comum ou raro dentre todos os documentos. O IDF do termo t no conjunto de documentos D (coleção de *tweets*) é dado por:

$$idf(t, D) = log(\frac{|D|}{|d'D : t'd + 1|}) \tag{4.1}$$

Onde |D| denota o número total de documentos e o denominador o número de documentos (*tweets*) d em que o termo t aparece mais um.

O peso $TF - IDF(t, d, D)$ pode ser definido como a importância de uma palavra para documento considerando-se toda a coleção de *tweets* (D). Ela aumenta proporcionalmente ao número de vezes que uma palavra aparece no documento mas é penalizada pela frequência desse termo no *corpus*, ajudando a controlar o efeito das palavras que são mais frequentes que outras.

É a partir desses conceitos que define-se as características do modelo de espaço de vetores, indicando sua importância no contexto.

4.4 Distribuição de postagens por usuários

Power Law é uma relação entre duas quantidades, onde uma variação relativa em uma quantidade resulta em uma variação relativa proporcional na outra quantidade, nesse caso as duas variáveis são *posts* x *usuarios* onde a curva informa a periodicidade em que cada usuário relata um crime.

4.4.1 Periodicidade de relato

A Figura 14 ilustra a distribuição de postagens por usuário do CRION.

Figura 14 – Periodicidade de relado

Fonte: Sistema CRION

4.4.2 Números de relatos de crimes

A Figura 15 ilustra o número de usuário que relatam crimes, quantidade resultante da votação em que os crimes estão na classe *Fácil*, decididas como unanimidade.

Figura 15 – Relatos de crime por usuário

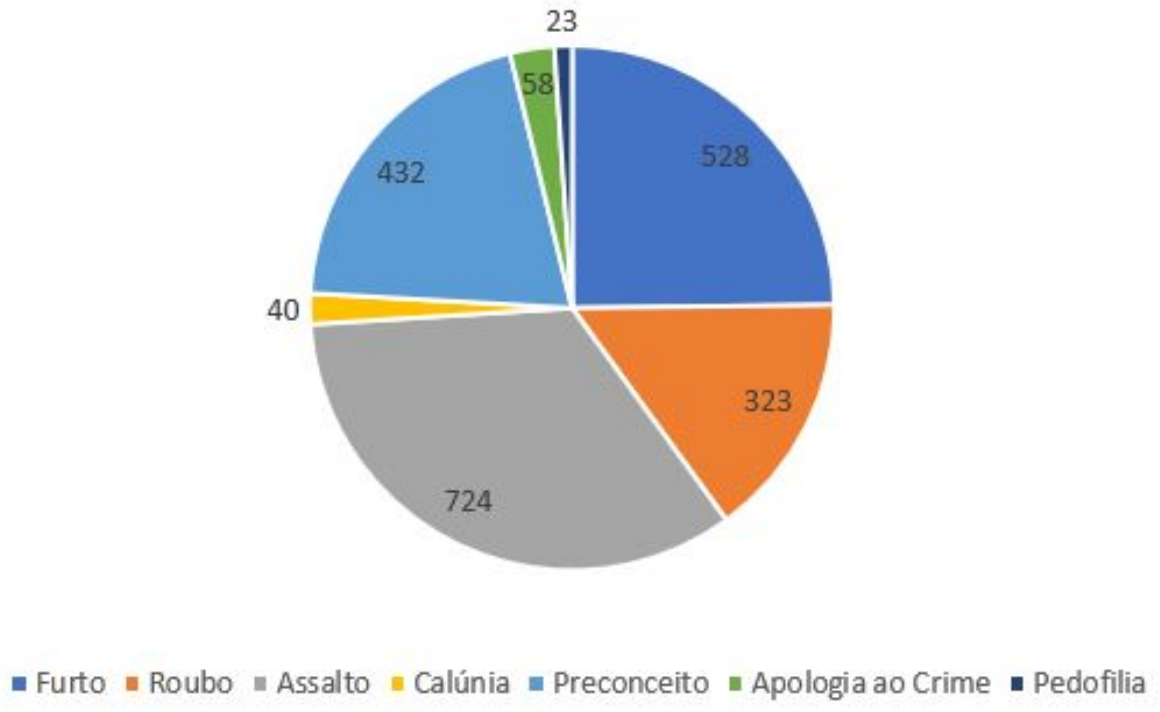
Fonte: Sistema CRION

4.5 Caracterização de crimes

Ao analisar como foi o comportamento dos *tweets* pode-se inspecionar a Figura 16, que é um gráfico em pizza para os dados coletados particionado em fatias de cores diferentes, cada um denotando ao número relativo de postagens referenciado

pelo tipo de crime. Dessa figura pode-se ver que os crime de "Assalto", "Roubo"e "Preconceito"são frequentes durante o período de coleta.

Figura 16 – Classes de crime



Fonte: Desenvolvido pelo autor

Lembrando que esse gráfico foi realizado dentre 2128 crimes citados na votação. As diferentes classes não foram consideradas nos testes de *SVM Linear* e foram classificadas manualmente. A principal fonte de informação das postagens coletadas são as palavras-chave, assim, com os termos pesquisados podemos ver que muitos dos relatos de crime se deram com as expressões “assalto” e “roubo”.

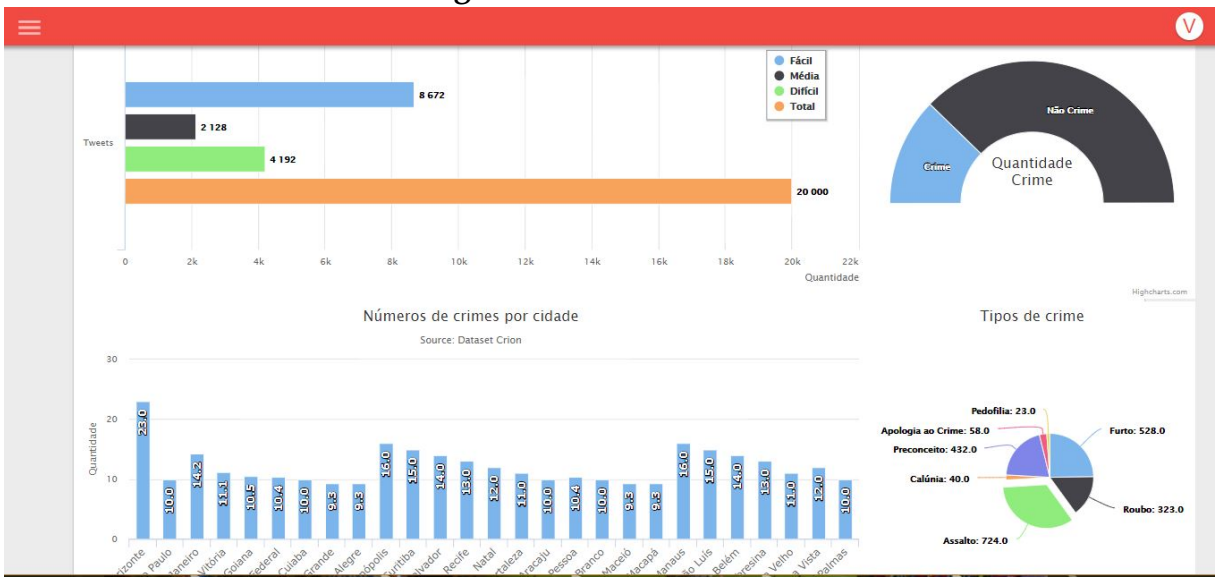
Os *tweets* também foram coletadas com georreferenciamento, porém são raras, ocorrem com pouca frequência. Por esse motivo foi levado em consideração a localização registrada no perfil do usuário, onde muitas vezes são contextos não relacionados à localização, como por exemplo *'meu mundinho'*. No aplicativo você pode ver além da localização, o usuário e seu respectivo ID de identificação na rede social.

O sistema permite a visualização dos dados coletados, atendendo a especifidades de consultas por usuários, localização e *tweets*. Logo, trata-se de uma ferramenta bastante adequada às necessidades do trabalho.

4.6 Protótipo de Aplicação Crion

A Figura 13 e 17 mostra o protótipo de aplicação Crion, que inclui o sistema de votação ilustrado anteriormente e a tela com a representação gráfica dos dados recebidos.

Figura 17 – Análise Crion



Fonte: Sistema CRION

Nesta tela, é possível analisar o local onde ocorreram os relados de crimes. Assim como a quantificação dos *tweets* coletados, informando as informações de crime ou não crime e os tipos de crime cometidos.

4.7 Técnica de Classificação

Após a caracterização do *dataset*, ele se torna habilitado para receber avaliações ou experimentos. Para o presente trabalho foi escolhido o método para proceder a tarefa de classificação textual: Máquinas de Vetores de Suporte (do inglês Support Vector Machine, SVM). Para a classificação leva-se em consideração os seguintes critérios:

- Número de classes: 2 (Crime / Não crime)
- Número de dados: 8,672 (teste)
- Número de características: peso IDF

O SVM exige que cada instância de dados, no caso os *tweets*, sejam representados como um vetor de números reais. Portanto, primeiro devemos convertê-los em números dados como descrito na subseção 4.3.3 e para representar um atributo definimos

o TFIDF do mesmo na coleção de *tweets*. Por exemplo, no conjuunto de três palavras como medica, baleada, assalto foi representado como (*TFIDF*,0,0), (0,*TFIDF*,0) e (0,0,*TFIDF*), se o número de valores em um atributo e ou mesmo o número de atributos não for muito grande, tende a ser mais estável.

Porém esse contexto se torna um problema não linear, onde há um conjunto desconhecido de variáveis reais propondo uma função objetiva não linear, para isso utilizamos a função kernel que converte problemas não lineares em lineares em um espaço l-dimensional, como descrito no capítulo 2.

5 ANÁLISES E RESULTADOS

Este Capítulo apresenta as análises, testes realizados e os resultados obtidos para a classificação dos *tweets* através da técnica SVM. Foi utilizado o banco de versão de teste, no caso, usou-se o banco de dados de tamanho 20000, escolhidos de forma aleatória dos *tweets* pós filtros.

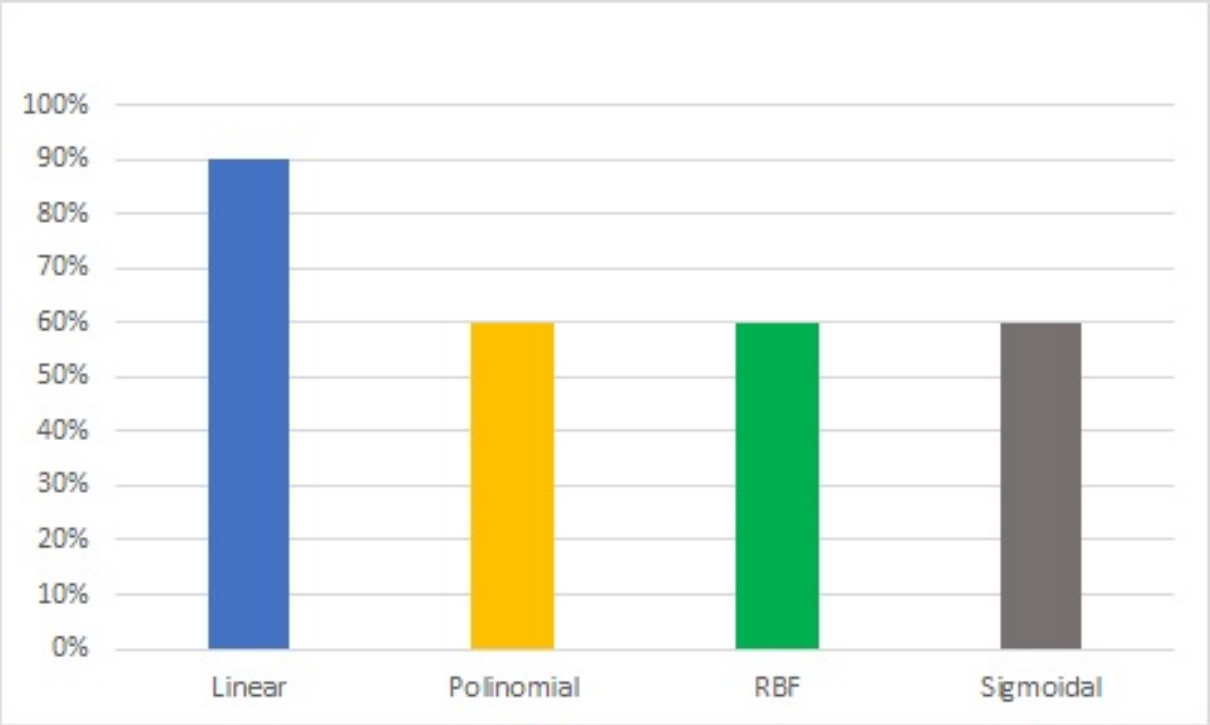
5.1 Testes em Kernels Distintos

Para experimento de treinamento com todos os kernels, utilizou-se os valores padrão para todos os procedimentos:

- Tipo de SVM: C-SVC
- Grau: 3
- $\gamma \frac{1}{n^{o}_{amostras}}$
- k (coeficiente) igual a 0
- C (Custo) igual a 1
- e(erro) igual a 0,001 sendo a tolerância do critério de parada

Foram aplicados as Funções Kernel demonstradas na Tabela 1 da Sessão 2.3.4. Para validar a acurácia, utilizou-se o método k-cross validation nativo da biblioteca libSVM, com k = 5.

Figura 18 – Acurácia por Kernel



Fonte: Desenvolvido pelo autor

Observando o gráfico apresentado na Figura 18, pode-se verificar que todos os kernels, desconsiderando o kernel linear, tiveram um desempenho igual, chegando a obter médias de 60% de acurácia. De uma maneira geral, pode-se dizer que o conjunto de dados de palavras apresentam características de difícil distinção. Esta não variação pode ter ocorrido devido a distribuição de features no espaço dimensional, ou seja, cada *feature* fica distribuida de forma incorreta para definição de classe. Portanto, podemos afirmar a ocorrência do fenômeno de *overfitting* que consiste em um conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados com esses kernels.

5.2 Refinamento de parâmetro

A biblioteca LIBSVM possui um script que implementa uma heurística chamada de Grid Search que otmiza o modelo de predição criado (otimiza o que foi generalizado). Seu objetivo é encontrar os parâmetros C e γ que melhora sua capacidade de classificação para dados desconhecidos.

Por padrão o *Grid Search* usa validação cruzada *5-cross validation* para estimar a precisão de cada combinação de parâmetros, obtendo a acurácia a cada iteração, variando exponencialmente o par(C,γ), que por padrão assume os valores: $C = 2^5, 2^3, ..., 2^{15}$ e $\gamma = 2^{15}, 2^{13}, ..., 2^3$. O código foi aplicado na mesma base de teste da seção anterior

e ao final da busca, o valor de acurácia retornado foi de 68,85%, somente 8,85% em relação ao experimento inicial com o *Kernel RBF*. A Figura 19 ilustra o resultado do experimento, onde a média varia entre 67.2% e 68.8%.

Figura 19 – Resultado grid.py

```
log2c=0.0 log2g=-2.0 rate=67.2131
log2c=-3.0 log2g=-2.0 rate=68.8525
log2c=3.0 log2g=-2.0 rate=67.2131
log2c=0.0 log2g=-3.0 rate=67.2131
log2c=-3.0 log2g=-3.0 rate=68.8525
log2c=3.0 log2g=-3.0 rate=65.5738
log2c=-4.0 log2g=-2.0 rate=68.8525
log2c=-4.0 log2g=-3.0 rate=68.8525
log2c=2.0 log2g=-2.0 rate=67.2131
log2c=2.0 log2g=-3.0 rate=65.5738
log2c=0.0 log2g=0.0 rate=67.2131
log2c=-3.0 log2g=0.0 rate=68.8525
log2c=3.0 log2g=0.0 rate=67.2131
log2c=-4.0 log2g=0.0 rate=68.8525
log2c=2.0 log2g=0.0 rate=67.2131
log2c=-1.0 log2g=-2.0 rate=68.8525
log2c=-1.0 log2g=-3.0 rate=68.8525
log2c=-1.0 log2g=0.0 rate=68.8525
log2c=5.0 log2g=-2.0 rate=67.2131
log2c=5.0 log2g=-3.0 rate=65.5738
log2c=5.0 log2g=0.0 rate=67.2131
log2c=0.0 log2g=-4.0 rate=68.8525
log2c=-3.0 log2g=-4.0 rate=68.8525
log2c=3.0 log2g=-4.0 rate=67.2131
log2c=-4.0 log2g=-4.0 rate=68.8525
log2c=2.0 log2g=-4.0 rate=68.8525
log2c=-1.0 log2g=-4.0 rate=68.8525
log2c=5.0 log2g=-4.0 rate=67.2131
```

Fonte: Resultado de tela grid.py

5.3 Avaliação classificação SVM Linear

Os demais testes foram realizados utilizando-se o Kernel Linear. Para este experimento, foram retirados 1000 *tweets* de forma aleatória da classe *Fáceis* e aplicados quatro tipos de testes com o *SVM Linear*:

- Aplicação do *5-Fold cross validation* em 80% da base e a predição em 20%,obteve-se 2848 *features*;
- Aplicação do *5-Fold cross validation* em 60% da base e a predição em 40%,obteve-se 2288 *features*;
- Aplicação do *5-Fold cross validation* em 40% da base e a predição em 60%,obteve-se 1605 *features*;

- Aplicação do *5-Fold cross validation* em 20% da base e a predição em 80%,obteve-se 961 *features*;

Após realizar o refinamento dos parâmetros C e γ para o Kernel, iniciaram se os testes avaliação de desempenho, considerando-se as diferentes divisões do tamanho da base, o método de avaliação de desempenho utilizado foi a matriz de confusão das aplicações de testes.

5.3.1 Classificação em 80%

Para 80% da base, ou seja, 800 *tweets* como treinamento obteve-se $C = 8,0$ e $\gamma = 0,0078125$. O método *5-fold* obteve a acurácia de 90,9887. A partir daí foi aplicado a base de teste para a predição dos *tweets* coletados, a Tabela 9 mostra a matriz de confusão da predição realizada.

Tabela 9 – Matriz de confusão (80% base)

	tweets		
	Positivo	Negativo	Total
Verdadeiro	19	3	22
Falso	74	105	179

A partir da matriz de confusão, calculou-se o F-score de desempenho para o classificador, onde obete-se o resultado de 0,18.

5.3.2 Classificação em 60%

Para 60% da base, obteve-se $C = 512,0$ e $\gamma = 0,0001220703125$, e o método *5-fold* obteve a acurácia de 90,1503. Com a aplicação a base de teste para a predição dos *tweets*, a Tabela 10 mostra a matriz de confusão da predição realizada. Calculou-se o F-score de desempenho com resultado de 0,13.

Tabela 10 – Matriz de confusão (60% base)

	tweets		
	Positivo	Negativo	Total
Verdadeiro	28	6	34
Falso	103	264	367

5.3.3 *Classificação em 40%*

Assim como nos outros testes, 40% da base, obteve-se $C = 128,0$ e $\gamma = 0,001953125$, e o método *5-fold* obteve a acurácia de 89,4737. A Tabela 11 mostra a matriz de confusão da predição realizada. Calculou-se o F-score de desempenho com resultado de 0,10.

Tabela 11 – Matriz de confusão (40% base)

	tweets		
	Positivo	Negativo	Total
Verdadeiro	31	13	44
Falso	119	438	557

5.3.4 *Classificação em 20%*

Para a base de 20% , obteve-se $C = 32,0$ e $\gamma = 0,0078125$, e o método *5-fold* obteve a acurácia de 84,9246. A Tabela 12 mostra a matriz de confusão da predição realizada. Calculou-se o F-score de desempenho com resultado de 0,082.

Tabela 12 – Matriz de confusão (20% base)

	tweets		
	Positivo	Negativo	Total
Verdadeiro	33	36	69
Falso	127	605	732

5.4 Conclusão

Através dos resultados e análises realizadas, pôde-se concluir que os objetivos definidos neste trabalho foram alcançados. O Capítulo 2 apresentou a rede social assim como a definição de crime, e também descreveu o funcionamento do *SVM*. O Capítulo 3 apresentou a estrutura do *dataset* CRION e a técnica realizada para a definição de classe dos dados coletados, assim como a filtragem de *tweets* que resultou em 42289 de onde foram tirados 20000 *tweets* aleaórios para a votação. Nos Capítulos 4 expõe-se a caracterização da base de dados e no Capítulo 5 resultados da classificação, o processo de classificação, treinamento e testes do classificador, além de fornecer uma interface de visualização dos dados recebidos.

Com os resultados obtidos no Capítulo 5, observou-se que a tarefa de criação de um *dataset* requer uma cautela pela individualidade da classificação textual, a sua

alta dimensão do espaço de caracterísiticas mostra-se uma tarefa difícil de se tratar, dificultando ainda mais por se tratar de postagens de rede social. Há fatos que agravam a tarefa de classificação textual como o grande volume de texto, o dinamismo da linguagem informal e as inúmeras variações de assuntos. Também há uma didifuldade relacionado ao Twitter onde temos o grande problema das mensagens serem limitadas à 177 caracteres.

Muitas das postagens não estão relacionadas a um relato de crime, o que faz o número de *tweets* 'não crime' sobrepor a quantidade de 'crimes' que devem ser analisados dificultando a criação de um banco de testes por conter muitas palavras não associadas a crime. Observou-se também o desempenho dos classificadores com o uso da biblioteca LIBSVM, como descrito na seção 5.1.

A dificuldade do classificador se deve pela metodologia adotada, a decisão de tornar cada palavra como uma *feature*, ou seja, um vetor de N palavras com N valores, dificulta a transformação de dados para um modelo l-dimensional conténdo muitas váriaveis de restrição para uma classificação. Pensando em um hiperplano, seria basicamente impossível montar um plano com separação ideal e margem máxima.

Assim, concluiu-se que é possível reconhecer os relatos de crimes utilizando as técnicas descritas no Capítulo 4, com ressalva às dificuldades pela variedade de *features*, onde os termos não possuem um significado semelhante, fazendo que o classificador não fique preciso.

Como parte dos trabalhos futuros cita-se o *feedback* automático do banco de treinamento e com o tempo deve ser capaz de diferenciar jargões e expressões diferenciadas, além de aplicar técnicas como LSA ou PLSA que permite que os conjuntos de observações sejam explicados por grupos não observados. As limitações difícultam a procura características específicas na rede social Twitter pelos textos curtos, pode-se então expandir a possibilidade da criação de um *dataset* para outras redes sociais como *Facebook* e *Instagram*, além de aperfeiçoar as metodologias e as métricas realizadas,sem a perda de acurácia do classificador. Há ainda a possibilidade de comparação de outros classificadores, como Naive Bayes. E como objetivo principal, criar um classificador multiclasses onde cada classe é o crime especificado, além de uma caracterização mais detalhada dos usuários, a própria biblioteca LIBSVM está apta a trabalhar com esse problema.

REFERÊNCIAS

ABE, S. Support Vector Machines for Pattern Classification. 2. ed. London: Springer-Verlag, 2010.

ALMEIDA, T. D. D. S. G. D. S. O uso da internet: A superexposição das crianças nas redes sociais no brasil. Revista Contribuciones a las Ciencias Sociales, 2016.

AMARAL, B. do. 2016. Disponível em: <https://exame.abril.com.br/tecnologia/brasileiro-usa-celular-por-mais-de-tres-horas-por-dia/>. Acesso em: 10 Out. 2017.

BITENCOURT, C. R. Tratado de Direito Penal: parte geral 1. 19. ed. São Paulo: Saraiva, 2013. 37 p.

CRISTIANINI, J. S.-T. N. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press, 2000.

FURTADO, R. R. C. O contributo das redes sociais acadÊmicas para o campo cientÍfico brasileiro na Área de ciÊncia da informaÇÃo. Challenges 2017: Aprender nas Nuvens, Learning in the Clouds, 2017.

JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. European Conference on Machine Learning, 1998.

KEMP, S. 2016. Disponível em: <https://www.slideshare.net/wearesocialsg/digital-in-2016/2-wearesocialsg_2>. Acesso em: 20 Abr. 016.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. International joint Conference on artificial intelligence., 1995.

LAGO, E. C. B. D. Mineração de textos, text mining. Revista de Ciências Exatas e Tecnologia, 2008.

MARANI, V. E.-C. C. L. L. Dajee: A dataset of joint educational entities for information retrieval in technology enhanced learning. Special Interest Group on Information Retrieval, 2016.

MORAES, L. M. da Silva; Marianne F. da S. D. C. A internet como ferramenta tecnolÓgica e as consequencias de seu uso: Aspectos positivos e negativos. Revista Científica Semana Acadêmica, 2014.

PICZAK, K. J. Esc: Dataset for environmental sound classification. Association for Computing Machinery, 2015.

REIS, L. A. M. dos. Uma abordagem para classificação de gesos manuais do Myo Armband. 2016. Monografia Graduação, PUC (Ponifícia Universidade Católica de Minas Gerais), BH, Brazil.

SANTOS, L. S. F. C. dos. Estudo Online da Dinâmica Espaço-temporal de Crimes através de Dados da Rede Social Twitter. 2013. Monografia (Programa de Pós-Graduação), UFMG (Universidade Federal de Minas Gerais), BH, Brazil.

SILVA, G. L. A. da. Text Mining, um estudo a partir da rede social Twitter. 2013. Monografia (Bacharel em Estatística), UFRGS (Universidade Federal do Rio Grande do Sul), RS, Brazil.

SIROIS, J. twitter4j.conf Class ConfigurationBuilder. 2012. Disponível em: <http://twitter4j.org/javadoc/twitter4j/conf/ConfigurationBuilder.html>. Acesso em: 15 Set. 2017.

SULLIVAN, S. L. K. Evolving kernels for support vector machine classification. Conference on Genetic and Evolutionary Computation, 2007.

TERMOS E GÍRIAS UTILIZADOS POR DETENTOS. 2013. Disponível em: <http://tudosobreseguranca.com.br/downloads/girias.pdf>. Acesso em: 17 Set. 2017.

YOUNG, C. N. de A. . K. S. Dependência de Internet: Manual e Guia de Avaliação e Tratamento. 1. ed. São Paulo: ARTMED EDITORA S.A, 2011. 77 p.

ZHU, X. Basic Text Process. 2010. Disponível em: <http://pages.cs.wisc.edu/~jerryzhu/cs769/text_preprocessing.pdf>. Acesso em: 25 Out. 2017.