Larissa Sayuri Futino Castro dos Santos

Estudo Online da Dinâmica Espaço-temporal de Crimes através de Dados da Rede Social Twitter

Belo Horizonte

2015, Fevereiro

Larissa Sayuri Futino Castro dos Santos

Estudo Online da Dinâmica Espaço-temporal de Crimes através de Dados da Rede Social Twitter

Universidade Federal de Minas Gerais

Instituto de Ciências Exatas

Programa de Pós-Graduaçõo

Orientador: Marcos Oliveira Prates

Coorientador: Erica Castilho Rodrigues

Belo Horizonte

2015, Fevereiro

Este trabalho é dedicado aos meus pais pelo incentivo e respeito aos meus sonhos.

Agradecimentos

Agradeço aos meus pais, Matico e Gilberto, por incentivarem a minha educação e felicidade toda a minha vida e tornarem possível essa etapa.

Agradeço aos meus orientadores, Marcos e Erica, por acreditarem no projeto, em mim, pelos e-mails tão prontamente atendidos e o encorajamento a fazer o Doutorado.

Agradeço aos meus familiares pelo apoio e motivação sobretudo minha Madrinha Solange e meu tio Sebastião os quais mostraram-se curiosos com a pesquisa e meus rumos acadêmicos.

Agradeço minhas amigas de infância - Fernanda, Gabriela, Helena e Luciana - e amigos de graduação - Thais, Maisa e Thiago - pelo esforço de manterem-se presentes na minha vida, pelas risadas e fofocas que me alegram e pelo apoio, direto ou indireto, a encarar o desafio de sair de casa.

Agradeço a todos os moradores da minha casa em BH pela companhia e respeito, sobretudo ao Carlos pela solicitude.

Agradeço às minhas "amigas do Mestrado" Juliane, Lívia e Rachel por terem ajudado a tornar esses dois anos inesquecíveis. Muitos almoços, muitos exercícios, vários lanches de tarde, histórias pra contar, forrós, uma crise de riso, uma viagem excelente, lágrimas de tristeza e alegria.

Agradeço aos meus colegas de laboratório Zayda, Raquel, Maurício, Douglas e Luis por serem meus exemplos de alunos e pela humildade que possuem. Em especial, à Raquel eu agradeço pela paciência de me ajudar em praticamente tudo no início do meu projeto e ao Maurício pela calma e estabilidade. Ao Luis eu agradeço todas as dúvidas, conselhos, a ajuda gigantesca no sonhado Shiny e o ambiente descontraído e leve do LESTE.

Ao Douglas eu agradeço o carinho, a companhia, o respeito e calma nesses meses mas sobretudo nos momentos mais sensíveis e estressantes. Em vários dias foram os gestos dele que me alegraram e foram a postura e personalidade dele que me serviram de exemplo.

Agradeço aos funcionários e Professores do Departamento por todo o trabalho e esclarecimentos prestados.

"And the most important:
Have the courage to follow your own heart and intuition.
They somehow already know what you truly want to become."
(Steve Jobs)

Resumo

Entender a dinâmica criminal é essencial para criação de políticas públicas mais adequadas para o controle dos diversos tipos de crimes. Neste estudo, procura-se mapear a ocorrência de crimes no estado de São Paulo através da coleta de postagens na rede social Twitter. A partir da informação dos dados coletados e através de métodos de aprendizado de máquina, o trabalho tem como objetivo classificar, de forma inteiramente automatizada, a ocorrência ou não de um evento de crime na região citada. Assim, pode-se visualizar aspectos espaço tempo da distribuição dos diversos tipos de crimes de maneira dinâmica, pois as coletas podem ser realizadas em tempo real. Nesse trabalho, apresentamos a forma empregada para coleta de *tweets* e os métodos de aprendizado de máquina para classificação dos *tweets*. Inicialmente, são utilizadas e apresentadas três técnicas de classificação de textos conhecidas como: Naive-Bayes, Árvore de Decisão e Máquinas de Vetores de Suporte (SVM). Um estudo de validação cruzada é realizado em cada uma das técnicas e essas são comparadas sob o ponto de vista da eficiência de classificação e tempo computacional.

Palavras-chaves: Mineração de Texto, Twitter, Naive-Bayes, Árvore de Decisão, SVM.

Abstract

To understand crime dynamic's is essential for the development of public politics to control many types os crimes. In this study, we aim to map the crime occurrences at the state of São Paulo by collecting posts from the Twitter Social Web. Using the coollected data and machine learning techniques this study aim to classify, in an automatic way, the occurrences of crimes in the cited area. This way, we are able to dinamically visualize space time aspects of the crime distribution due to the possibility of real time collection of data. In this work, we present how to collect tweets and the machine learning methodology for the tweet classification. At first, we present and use three text classification techniques, known as, Naive-Bayes, Decision Trees and Support Vector Machines (SVM). Next, a cross validation study is performed for each technique and they are compared by classification eficiency and computational time.

Key-words: Text mining, Twitter, Naive- Bayes, Decision Trees, SVM.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

1.1 Crime e Criminalidade

Os seres humanos, agrupados em sociedade, vivem em condições propícias a gerar conflitos de interesses. Entende-se que existem pressupostos imprescindíveis para uma existência em comum como: a vida, a integridade física, a liberdade de atuação, a propriedade, os chamados bens jurídicos. Esses merecem atenção integral para a sua não violação.

Crime e criminalidade são termos para expressar ou caracterizar condutas, fatos que vão de encontro aos bens jurídicos. Pode-se definir crime como "ato ilícito, de consequências desagradáveis, contravenção" (AURÉLIO, 2013) designando, portanto, eventos que ocorrem em desajuste com alguma norma ou regra. No contexto abordado nesse trabalho, um crime viola uma lei civil sendo o conceito comumente utilizado de contravenção perante às leis tipificadas no Código Penal brasileiro. Do conceito de crime surge a definição de criminalidade a qual pode ser considerada como "natureza ou estado do que é criminal ou ainda o conjunto de atos criminosos cometidos em um meio dado" (AURÉLIO, 2013).

Segundo o que a própria Constituição prevê, em consonância com o Código Penal: "Não há crime sem lei anterior que o defina, nem pena sem prévia cominação legal". Isto quer dizer que é necessário que um dado crime e sua pena estejam exatamente e literalmente tipificados no Código Penal brasileiro para eventual aplicação legal. No atual Código Penal brasileiro são descritos 53 tipos distintos de crime tais como "crimes contra a vida" e "lesões corporais" sob a forma de 241 artigos.

Um ato criminoso sempre está associado a uma perda seja ela material ou imaterial correspondendo a criminalidade a um dos maiores problemas do Brasil e do mundo hoje e sempre, o que justifica o seu estudo.

1.2 Twitter

O Twitter é uma rede social e *microblogging* criada em 2006 por Jack Dorsey, Evan Williams e Biz Stone nos Estados Unidos cujo serviço é gratuito pela internet e as atualizações são exibidas no perfil de um usuário em tempo real e também enviadas a outros usuários seguidores que tenham assinado para recebê-las. A principal característica da rede social é a limitação de 140 caracteres nos seus textos. Na sua concepção a idéia era possibilitar uma comunicação rápida, similar à dos textos de SMS (*Short Message Service*).

O Twitter ganhou tanta notoriedade entre as redes sociais que em 14 de Setembro de 2010, divulgou que seu número total de usuários era 175 milhões. Hoje a rede abriga a imensa maioria dos veículos de comunicação e atrai muitas empresas que visam conhecer seus consumidores e

estabelecer comunicação com eles via Web (*World Wide Web*). É notória a participação de pessoas públicas como artistas e políticos, por exemplo, o Papa Francisco e o presidente norte-americano Barack Obama além da sua utilização em movimentos políticos e sociais pelo mundo.

O intuito desse trabalho é descrever o comportamento de variáveis de crime e criminalidade através de *tweets*. Como típica variável sócio econômica, a ocorrência de crimes tem potencial de ser explorada a partir de características espaciais, o que justifica a validade do emprego da rede social descrita já que o Twitter repassa, quando autorizado, a localidade em que determinado *tweet* foi postado. Além disso, o microblogging permite a coleta de suas informações de maneira clara a partir de um tipo de aplicativo denominado API (*Application Programming Interface*) a qual será descrita com detalhes a seguir no texto.

1.3 Objetivos

O presente trabalho tem como principal objetivo analisar relatos de crime a partir de tweets no estado de São Paulo, Brasil.

Tal tarefa compreende obter tais postagens de maneira rápida e eficiente, classificá-las segundo técnicas de classificação de texto difundidas na literatura e disponibilizar as estatísticas e análises provenientes delas sob a forma de produtos claros, concisos e de apelo visual.

Este trabalho divide-se em mais sete capítulos principais; no Capítulo 2 são introduzidos conceitos iniciais de Aprendizado de Máquina. No Capítulo 3 são introduzidos os conceitos, características e etapas a serem seguidas em um problema de Classificação Textual. Já no Capítulo 4 descreve-se com detalhes o processo de Coleta das Postagens em estudo e os seus resultados. Em seguida, no Capítulo 5, apresenta-se a teoria resumida das técnicas empregadas Naive-Bayes, Árvore de decisão e Máquina de Vetores de Suporte (do inglês *Support Vector Machine* abreviado para SVM). Já no Capítulo 6 constam resultados da utilização da técnica no banco de treinamento adotado. No capítulo seguinte (7) são apresentadas figuras descrevendo o tipo de visualização pretendida para as postagens já classificadas. O Capítulo 8 explicita as conclusões do presente trabalho.

2 Aprendizado de Máquina

Conforme o que foi apresentado, este trabalho visa analisar relatos de crime a partir de *tweets* em São Paulo. Como consequência do dinamismo e subjetividade da linguagem, sobretudo na Web, é comum que para um determinado termo observe-se postagens de diferentes assuntos. Observa-se os quatro *tweets* reais que compoem o banco de dados inicial:

1. "Esconde o Game of Thrones dentro do Celso Furtado pra poder ler na aula, mas não sigam meu exemplo. O Furtado é muito mais que formidável."
2. "Estuprou o português essa daí"'
3. "Mano que droga, assaltaram a casa da minha tia, roubaram TUDO, levaram até o cachorro :/ "
4. "Galera, quem aqui tem cartão de crédito Amex? E quem já sofreu clonagem? Ou é só comigo?"

Os dois primeiros foram coletados por conterem as palavras "furtado" e "estuprou", termos que designam a ocorrência de crimes bem conhecidos. Todavia, Furtado é nome de um autor e pesquisador da Economia e a palavra "estuprou" aparece em um contexto metaforizado para designar um erro grave que diz respeito à língua portuguesa.

Já nos *tweets* 3 e 4 a situação é oposta; eles exemplificam o uso da rede para o fim que está em estudo nesse trabalho sendo, perfeitamente, o relato de um crime ocorrido ou de conhecimento do usuário da rede social Twitter.

Dessa forma, é pertinente a classificação das postagens quanto à adequabilidade do contexto pretendido segundo técnicas para texto consolidadas na literatura. Em linhas gerais pode-se dizer que tal processo de classificação dá-se através da leitura de um conjunto de *tweets* iniciais em que cada um deles será classificado como sendo de "crime" ou "não crime". Esse conjunto de *tweets* rotulados servirá de subsídio para que técnicas computacionais criem uma regra de predição que possa ser utilizada na previsão do rótulo de novos dados.

Este capítulo intui formalizar a descrição fornecida no parágrafo anterior apresentando parte da nomenclatura usual empregada.

O interesse do problema em análise é obter classificadores sendo a teoria de Aprendizado Estatístico a que visa obter aqueles com boa generalização; que prevêem corretamente a classe de novos dados do mesmo domínio que o aprendizado ocorreu (LORENA; CARVALHO, 2007).

Em uma situação típica tem-se uma medida de interesse, quantitativa ou categórica, sob a qual se quer fazer previsão tomando-se como base um conjunto de características ou atributos. Tem-se um conjunto de dados de treinamento em que se observa, para um conjunto de elementos, a medida de interesse e o respectivo conjunto de atributos. Esses são os dados empregados

na construção de um modelo de predição o qual permitirá prever a medida de interesse para novos objetos desconhecidos. Um bom classificador é aquele que prevê essa medida com acurácia (HASTIE et al., 2009).

As técnicas de Aprendizado de Máquina empregam o princípio inferencial chamado indução no qual obtém-se conclusões genéricas a partir de um conjunto particular de exemplos (LORENA; CARVALHO, 2007). Este trabalho aborda o tipo supervisionado de aprendizado indutivo, conceito esclarecido no Capítulo 3). Considere um conjunto de exemplos com seus respectivos rótulos (x, y) (x_i é o exemplo i e y_i o rótulo associado). O intuito é produzir um classificador (ou modelo preditor) que possa prever o rótulo de novos dados. Chama-se de treinamento o processo de indução de um classificador a partir de uma amostra de dados. Pode-se dizer ainda que x é o domínio da função f representativa do classificador a qual fornece a predição y.

No contexto desse trabalho os rótulos representam o fenômeno de interesse assumindo as classes "crime" e "não crime". Cada tweet lido e classificado, o exemplo x_i, é representado por um vetor de características ou atributos, que correspondem aos termos ou palavras da postagem.

É importante estimar as taxas de predição (acerto e erro) do classificador encontrado quando utilizado em novos dados. É visando essa etapa que divide-se, em geral, o banco de dados inicial em banco de treino e banco de teste. Medidas para se averiguar a adequabilidade do classificador são apresentadas na Subseção 3.1.5 do Capítulo 3 de Classificação Textual.

Voltando ao conceito de generalização há três possíveis relações entre o classificador e os dados de treino. Considere o conjunto de treinamento da Figura 1 nela vêem-se duas classes, círculo e triângulo e possíveis fronteiras de decisão. Na Figura 1(a) a hipótese é de um classificador especialista no conjunto de treinamento de modo que até os ruídos são corretamente classificados. Tamanha especificidade implica baixa taxa de acerto quando o classificador for empregado a novos dados, situação denominada superajustamento ou *overfitting*. Também pode acontecer de o classificador ter baixo desempenho para o conjunto de treinamento que o induziu. Lorena e Carvalho (2007) afirmam que tal situação pode ocorrer quando os exemplos de treinamento empregados forem pouco representativos ou quando o modelo obtido é muito simples. Esse é o caso da Figura 1(c) em que o segmento de reta desconsidera que apesar de próximos entre si há pontos de classes opostas. Diz-se que ocorreu um sub-ajustamento ou *underfitting*. Vê-se, por conseguinte, que o classificador na Figura 1(b) equilibra as situações apresentadas com uma complexidade intermediária, classificação da maioria dos exemplos sem ter expertise nos mesmos.

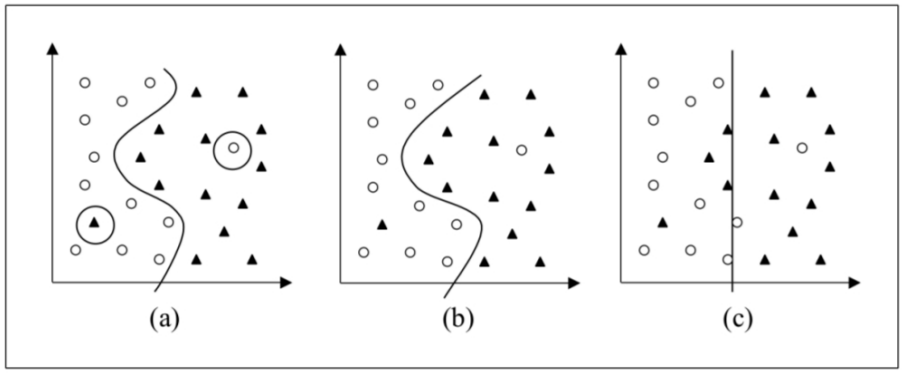

Figura 1 – Três possíveis situações para classificador de duas classes na fase de treinamento: (a) - Superajustamento ou *overfitting*: Regra de classificação especialista, (b) - Ajuste ideal, regra de classificação não simplista e não especialista e (c) - *Underfitting*: Regra de classificação simplista.

O que se pretende esclarecer com a Figura 1 é o que a teoria de Aprendizado Estatístico considera na construção de um classificador: desempenho e complexidade.

Em linhas gerais, pode-se dizer que para qualquer conjunto de treinamento é factível a determinação de uma boa função discriminatória. Todavia essa classificação precisa, de alto desempenho, costuma estar associada a um modelo preditor especialista, com grande complexidade que se torna pouco poderoso em termos de generalização. A teoria de Aprendizado Estatístico define condições para se encontrar o classificador que equilibre o conflito de interesses apresentado.

3 Classificação de Texto

A tarefa de classificação ou categorização de texto pode ser descrita brevemente como a "automatização do processo de organização de textos em um conjunto de categorias pré-definidas" (TOKER; KIRMEMIS,). Tais categorias representam o fenômeno de interesse sobre o qual se deseja fazer previsões. Shimodaira (2014) acrescenta que a tarefa de classificação de documentos se dá a partir de seu conteúdo, ou seja, pelas palavras que os compõem.

Tradicionalmente cada novo documento é analisado e classificado manualmente por experts no domínio/assunto baseados no conteúdo do documento. Tal tarefa consome muito tempo e recursos humanos e com o intuito de facilitar a mesma, esquemas para classificação automática são necessários. Com o rápido crescimento da informação *online* o tratamento e a classificação de textos tornou-se uma das técnicas chaves para organizar textos.

O emprego mais comum de classificação de texto é o problema de filtragem de *spam* (SHIMODAIRA, 2014) mas ela também pode ser empregada para reconhecimento da língua do texto, da idade ou gênero do autor ou até mesmo a sua identidade (como em situações que deseja-se saber o autor de determinado documento após a sua morte). Também é possível empregá-la com o intuito de achar informação na Web ou guiar usuários na procura por *websites* (JOACHIMS, 1998).

As tarefas de classificação de textos tem, dentre outras características, alta dimensão do espaço de características e vetores esparsos conceitos descritos na Subseção 3.1.3.6.

Nesse capítulo descreve-se as etapas da categorização de texto - pré-processamento dos textos, escolha e ajuste da técnica de classificação a ser empregada e verificação da qualidade da classificação feita além dos seus jargões. Uma vez que todas as técnicas de categorização de textos empregadas no presente trabalho são supervisionadas a etapa de treinamento será incluída nas descrições.

3.1 Etapas para Classificação de Texto

3.1.1 Jargões

Algumas das expressões que serão empregadas nas tarefas de categorização de textos:

- Documento: Texto que será classificado. No nosso caso, cada tweet a ser estudado será um documento.
- *Corpus*: Conjunto de todos os documentos para classificação. Se em um único dia são coletados 100 *tweets* de crime todos eles conjuntamente compõem o *Corpus* para classificação.
- Banco de Dados/Textos: Banco com os Dados disponíveis para a construção da regra de classificação. Constitui-se de documentos classificados manualmente por experts no domínio/assunto e divide-se em dois, o de Treinamento e o de Teste.

- Banco de Treinamento: Parte do Banco de Textos utilizada para a construção da regra de classificação.
- Banco de Teste: Parte do Banco de Textos que avalia a qualidade da regra de classificação obtida.
- Termo: Uma única palavra do texto em classificação.
- Vocabulário/ Dicionário: Todas as diferentes palavras que compoem um Corpus.

3.1.2 Etapa I: Classificação Manual/ Supervisão

Essa é a etapa que diferencia os métodos Supervisionados dos Não Supervisionados. Nela, um expert no assunto lê e classifica uma coleção de documentos, o Banco de Dados/Textos. Esse banco é dividido em dois, o banco de treinamento e o banco de teste. É com base no primeiro, também conhecido como banco de exemplos, que a técnica procura a melhor regra de classificação. Neste trabalho foram lidos e classificados 26.503 documentos para compor o Banco de Dados.

3.1.3 Etapa II: Pré-Processamento do Texto

A representação de um problema tem um forte impacto na acurácia da generalização de um sistema de aprendizado. Documentos, que tipicamente são strings ou caracteres, tem de ser transformados em uma representação adequada para o algoritmo de aprendizagem e a tarefa de classificação (JOACHIMS, 1998).

De forma simplificada, pode-se dizer que na etapa de pré-processamento almeja-se a melhor representação da informação textual para o computador. Ou seja, como transformar o conjunto de documentos numa informação compreesível para a máquina. Isso passa ainda, por deixar apenas os termos realmente informativos o que implica a exclusão de alguns deles como será descrito a seguir.

A etapa de pré processamento do texto foi realizada a partir do pacote *tm* (FEINERER; HORNIK, 2014) do software estatístico R, versão 3.1.1 (R Core Team, 2014).

3.1.3.1 Termos indesejados

É comum que em textos da web apareçam termos ou expressões não desejadas quando intui-se classificá-los. Apesar de comporem os documentos, são considerados não informativos e, portanto, devem ser excluídos dos mesmos. Nesse trabalho excluem-se links para páginas externas e a menção a outro usuário da rede (por exemplo, @MenezesMyWorld).

3.1.3.2 Remoção de Acentos, Pontuações e Caracteres não Alfa-Numéricos

Pontuações são imprescindíveis para a expressão do autor. Entretanto, as técnicas de classificação automática são incapazes de compreendê-los a menos de meros caracteres. Dessa forma, torna-se conveniente excluí-las dos documentos em classificação. O mesmo pode ser aplicado aos acentos a caracteres não alfa-numéricos (letras e números) da língua.

3.1.3.3 Tokenização

Para que um documento tenha representação em vetor deve-se ler cada documento a partir das suas palavras individuais. Ou seja, garantir que o computador não vai ler cada documento como uma única *string*, quebrando-o em palavras individuais. Por exemplo, a frase "Assalto na rua" deve ser representada por "Assalto" "na" "rua".

3.1.3.4 Remoção das *Stop-Words*

"As palavras mais frequentes usualmente não trazem muito sentido ao texto" (ZHU, 2014) essa é a idéia para a remoção de palavras como "de", "a", "aquela", "também", "está", "tiveram". As *stop-words* são arbitrárias cabendo a cada contexto a definição de termos que de tão comuns, não são considerados úteis para a classificação.

3.1.3.5 Remoção de Termos de baixa frequência

Na Web a linguagem empregada não segue a norma culta da língua. A espontaneidade criou uma linguagem com expressões próprias e a possibilidade de que usuários criem termos e erros de digitação também são comuns dado o dinamismo da rede. Essas características implicam em uma enorme quantidade de termos que aparecem pouquíssimas vezes relativamente ao *Corpus*. É prática comum considerar apenas os termos com uma frequência mínima a fim de se diminuir a dimensão do vocabulário em uso e, consequentemente, o tempo de processamento e a qualidade das classificações. Como essa etapa é subjetiva e tem fortes implicações na qualidade da classificação optou-se por avaliar, em cada técnica, o número para frequência mínima do termo para que entre na análise assumindo os valores 1, 3 e 5.

3.1.3.6 Modelo de Espaço de Vetores

Etapa mais importante no pré-processamente de um texto por traduzir a informação textual em linguagem máquina. Classificadores de texto não costumam usar nenhum tipo de representação complexa para linguagem: é comum que um documento tenha representação conhecida como *bag of words*. Essa é a representação mais simplista possível pois só demarca quais palavras estão incluídas no documento e quantas vezes cada palavra ocorre, desconsiderando a ordem dos termos.

O mais usual é considerar um Modelo de Espaço de Vetores. Nesse caso, cada documento é representado por um vetor de palavras sendo cada palavra considerada um atributo/característica.

Cada atributo, por sua vez, recebe uma ponderação. Existem três tipos de pesos que relacionam cada termo (palavra) na especificação de um documento ou das categorias em análise.

Começa-se exemplificando conjuntamente a representação vetorial dos documentos e o tipo de ponderação mais intuitivo: pela frequência dos termos.

Considere três possíveis *tweets*:

1. Assalto na rua de casa!
2. Assaltaram a vizinha na rua de casa!

3. Assalto à mão armada em casa na Pampulha hoje.

Para sermos mais realistas às práticas descritas retiram-se artigos, preposições, acentuações, pontuações chegando a:
1. Assalto rua casa
2. Assaltaram vizinha rua casa
3. Assalto mao armada casa Pampulha hoje

Não fizemos a exclusão de termos com frequência inferior a um dado número porque o exemplo tem poucos documentos o que implica em termos com baixas frequências no *Corpus*.

Tem-se então: 3 documentos e 9 diferentes termos (ou vocabulário de tamanho 9). Para representarmos cada um desses documentos na forma vetorial poderíamos pensar nas entradas do vetor como o número de vezes que cada termo ocorreu em cada documento. Na matriz abaixo a linha i corresponde ao documento i e as colunas são nomeadas com o nome do atributo a que faz referência.

armada	*assaltaram*	*assalto*	*casa*	*hoje*	*mao*	*pampulha*	*rua*	*vizinha*
0	0	1	1	0	0	0	1	0
0	1	0	1	0	0	0	1	1
1	0	1	1	1	1	1	0	0

A expressão $f(t, d)$ irá denotar o número de vezes que o termo t ocorre no documento d e essa corresponde à ponderação mais simples; Frequência do Termo que será chamado TF do inglês *term-frequency*.

O peso TF é o mais intuitivo mas documentos que tenham termos mais frequentes tendem a ser enfatizados na análise. Trata-se do mesmo raciocínio da remoção de *stop-words*; palavras mais frequentes informam menos acerca da categoria do documento justamente por serem mais corriqueiras, se enquadrarem numa maior diversidade de contextos.

Pensando nisso surgiu outro tipo de peso para os termos. No Inverso da Frequência nos Documentos (IDF de *Inverse Document Frequency*) o intuito é valorizar termos cuja ocorrência é menor. É uma medida de quanto o termo é comum ou raro dentre todos os documentos. O inverso da frequência do termo t no conjunto de documentos D é dado por:

$$idf(t, D) = log\left(\frac{|D|}{|d \epsilon D : t \epsilon d + 1|}\right) \tag{3.1}$$

Em 3.1, $|D|$ denota o número total de documentos e o denominador o número de documentos d em que o termo t aparece mais um.

Com base nos pesos $TF(t, d)$ e $IDF(t, D)$ é razoável pensar em uma ponderação que procure levar em consideração as duas qualidades descritas. O peso $TF - IDF(t, d, D)$ (*Term Frequency - Inverse Document Frequency*) pode ser definido, então, como uma estatística que reflete quão importante uma palavra é para um documento considerando-se todo o *Corpus*. Ela

aumenta proporcionalmente ao número de vezes que uma palavra aparece no documento mas é penalizada pela frequência desse termo no *Corpus*, o que ajuda a controlar o inevitável efeito das palavras que são mais frequentes que outras.

É a partir desses conceitos que entende-se as características de alta dimensão do espaço de características e vetores esparsos apresentadas anteriormente. A representação sob a forma de espaço de vetores associa cada termo a uma dimensão o que implica em espaços com dimensão próxima do vocabulário do contexto e consequentemente a representação vetorial de cada documento contendo apenas algumas poucas entradas não nulas.

3.1.3.7 Extração/Seleção de Características

"Na categorização de textos é comum confrontar-se com espaços de características com mais de 10000 dimensões, usualmente excedendo o número de exemplos de treinamento presentes. Percebeu-se a necessidade de uma seleção de características para viabilizar o uso de métodos convencionais de aprendizado, para melhorar a acurácia da generalização e para evitar *overfitting*", (JOACHIMS, 1998).

Alternativas para evitar espaços de grande dimensão existem sob a forma de abordagens para extração das características mais relevantes. Todavia, em classificação de texto existem pouquíssimas características irrelevantes. Joachims (1998) demonstra que mesmo características consideradas pouco informativas ainda contêm informação considerável e de alguma forma relevante de sorte que mesmo um classificador que utiliza apenas as piores características tem performance muito superior à aleatória. Em suma, é possível que a seleção de características prejudique a performance do classificador como consequência de perda de informação.

Sendo assim, optou-se no presente trabalho por não fazer seleção de termos considerados mais relevantes.

3.1.4 Etapa III: Técnicas de Classificação

Para o presente trabalho foram escolhidos três diferentes métodos para proceder a tarefa de classificação textual: Naive-Bayes, Árvore de Decisão e Máquinas de Vetores de Suporte (do inglês *Support Vector Machine*, SVM). Cada um deles tem revisão bibliográfica e apresentação em Seções do Capítulo 5.

3.1.5 Etapa IV: Medidas de Performance

Essa constitui a fase de verificação da qualidade da classificação feita. Inicialmente, tem-se um banco de dados o qual é dividido em banco de treinamento e banco de teste. Ambos passam pela classificação manual por um expert no domínio em análise. Usa-se o banco de treinamento para construir a regra de classificação e o banco de teste para avaliá-la.

São várias as medidas que servem a esse propósito, todas relacionando quantidades relativas à classificação computacional com relação à manual.

Na Tabela 1 apresenta-se uma Tabela de Contingência 2x2 que ilustra essas relações, conhecida como Matriz de Confusão:

Tabela 1 – Classficação Manual vs Classificação da Técnica empregada

	Manual do Expert		Total
Técnica/ Sistema	Crime	Não Crime	
Crime	a	b	a+b
Não Crime	c	d	c+d
Total	a+c	b+d	n

As entradas b e c denotam, respectivamente, as quantidades de falso positivo e de falso negativo obtidas pelo classificador. Algumas medidas comumente empregadas para avaliar e comparar classificadores são:

$$PRECISAO : p = \frac{a}{a+b}$$
$$RECALL : r = \frac{a}{a+c}$$
$$F1 = \frac{2rp}{r+p}$$

As medidas mais populares são a de precisão e o *recall* o qual será denotado por revocação em que relacionam-se os documentos corretamente classificados como "crime" com respeito a tudo que a técnica classificou como crime e a todos os documentos que realmente são de crime (segundo o expert), respectivamente.

Se um classificador tem um alto valor de falsos negativos ele não classifica corretamente quem deveria ser classificado apresentando uma baixa revocação. Se um classificador tem alto valor de falsos positivos ele classifica elementos que não deveriam ser classificados apresentando baixa precisão.

A média harmônica entre a precisão e a revocação corresponde à medida $F1$. Lembra-se que a média harmônica entre dois números x e y tende a ser próxima de $min(x, y)$ de sorte que obtem-se um valor de F1 alto quando a precisão e a revocação são conjuntamente elevadas.

4 Coletas de *Tweets*

O Twitter, rede social e *microblogging*, permite que usuários coletem as postagens realizadas em seu ambiente. As etapas para coleta são descritas em detalhes a seguir. Elas incluem a definição da região de estudo e de termos ou expressões que caracterizem o contexto desejado. Em seguida, apresenta-se a etapa de coleta dos *tweets* que constitui o banco de textos desse trabalho.

4.1 API do Twitter

A Interface de Programação de Aplicativos (API) corresponde a um "conjunto de rotinas e padrões estabelecidos por um *Software* para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do *software*, mas apenas usar seus serviços"(WIKIPEDIA, 2013).

Perfis que coletam *tweets* fazem uso da rede de forma mais avançada que meros usuários mas completamente sujeitos às regras dos desenvolvedores, apresentadas na documentação da API.

Atualmente, é a API 1.1 que está vigente para o Twitter. Ela está disponível online (TWITTER, 2013) e as coletas feitas para esse trabalho enquadram-se na categoria '*search*' a qual retorna *tweets* baseadas em termos desejados. Ou seja, especifica-se uma palavra (termo) e o Twitter retorna postagens nas quais tal termo estava presente, literalmente.

A API 1.1 do Twitter diferencia-se das anteriores, principalmente, por exigir identificação do usuário. Na prática, durante a sua vigência, cada pedido de coleta à rede social chama-se requisição e é registrado na conta do aplicativo do usuário de modo que a rede social tem mais controle sobre quando e quantos *tweets* estão sendo repassados a qual usuário.

O Twitter impoẽ a taxa máxima de 180 requisições a cada 15 minutos corridos, sendo que cada requisição pode voltar, no máximo 100 *tweets* por página com as características especificadas. O tipo *search* permite que se especifique um ou mais parâmetros dentre todos os possíveis (número de *tweets*, termo de pesquisa, região/ área circular, língua, dentre outros).

4.1.1 Termos para Coleta

Parte central das discussões do trabalho dá-se na definição de quais termos servem à coleta de tweets de relato de crimes.

O Código Penal tipifica, em sua parte especial, cerca de 241 crimes sob a forma dos artigos de 121 a 361 o que dimensiona a extensão e variedade da variável pretendida para estudo.

Tendo em vista que a língua e a linguagem passam por constantes mudanças e são instrumento de comunicação de todas as camadas econômicas, sociais, etárias e com níveis diversos de educação e acrescentando-se a isso a flexibilidade e espontaneidade da comunicação

Web é de se esperar uma imensa variabilidade nos termos e formas de usá-los para se referir à ocorrência de um crime.

Na fase inicial do trabalho foram definidas algumas palavras comumente associadas a crimes. A lista de 131 palavras encontra-se no Anexo, Seção 9.1 desse trabalho na Tabela 9.1. De forma geral, a lista foi construída a partir de variações de número e grau de 25 radicais considerados mais populares para designar crimes. Os principais substantivos desses radicais seguem listados na Tabela 4.1.1 a seguir.

Agressão	Atentado violento ao pudor	Furto	Pedofilia	Saída de banco
Arrastão	Clonagem	Gangue	Prisão	Sequestro
Arrombamento	Estelionato	Golpe	Quadrilha	Tráfico
Assalto	Estupro	Ladrão	Refém	Violência doméstica
Assassinato	Fraude	Morte	Roubo	Vítima

Tabela 2 – Vinte e cinco substantivos principais dos radicais que deram origem às palavras-chave para coletas no Twitter.

4.1.2 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 que o uso de redes sociais é recente e associado a características sócio-econômicas a distruibuição dos usuários do Twitter se dá conforme tais características. Dessa forma optou-se por iniciar o estudo no estado de São Paulo, Brasil. Acredita-se que a maior população e quantidade de usuários permitirá a obtenção de mais dados o que corresponde a mais insumo para as técnicas de classificação e análise.

O estado de São Paulo situa-se na região Sudeste do Brasil e é o mais populoso do país com cerca de 43 milhões de habitantes, ou 22% do total nacional. Atualmente, São Paulo possui o maior parque industrial e o maior PIB entre todos os estados brasileiros com economia respondendo por cerca de 33% do total de riquezas produzidas no país além do segundo maior Índice de Desenvolvimento Humano (IDH) do país.

A especificação de São Paulo como região de estudo foi possível a partir do centróide geográfico do estado, -22.26527,-48.72848 de latitude e longitude, e a determinação de uma distância definidora do raio a encobrir todo o espaço desejado de 455 km. A Figura 2 a esquerda demarca a região circular em que as coletas foram feitas. Nota-se a cobertura do estado de São Paulo mas também a inclusão de municípios de fronteira com o mesmo.

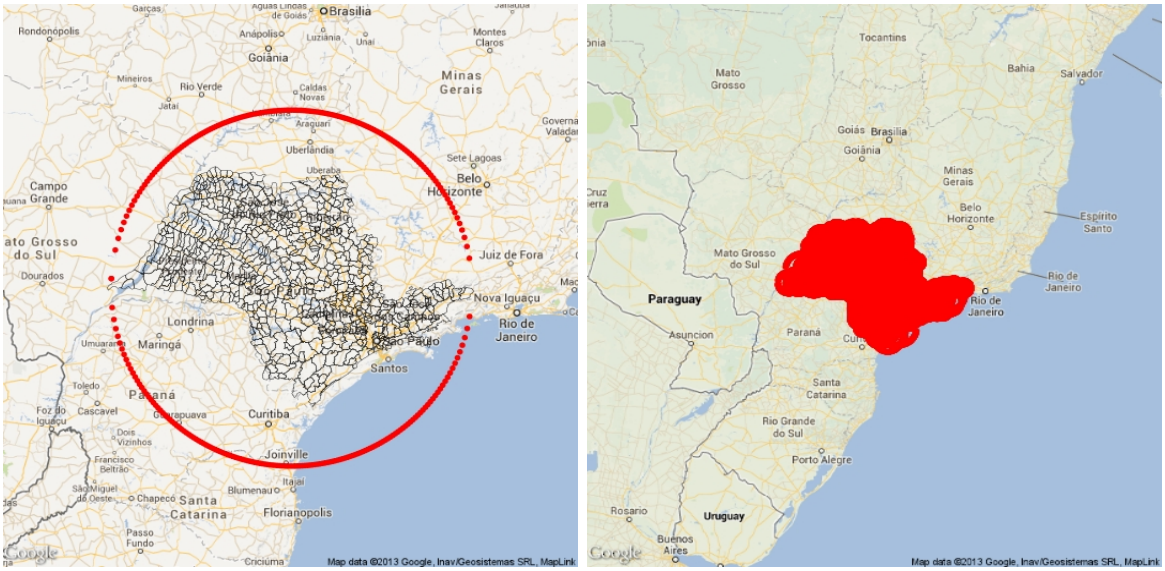

Figura 2 – Esquerda: Região de coleta de tweets, especificação para todo o estado de São Paulo. Direita: Região de coleta de tweets, especificação por municípios de São Paulo.

As coletas equivalentes que deram origem ao primeiro banco de dados foram realizadas a partir de um *script* em linguagem Python nos moldes do que foi descrito. Resultados e discussões dessa etapa são apresentados na Seção seguinte.

4.2 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.

4.2.0.1 Características

O banco de dados inicial desse trabalho foi coletado durante, aproximadamente, nove dias de Setembro de 2013: 03, 04, 05, 06, 10, 12, 13, 16 e 17. Diz-se aproximadamente porque a lista foi crescendo no decorrer do período, à medida que ia-se percebendo a necessidade de inclusão ou exclusão de algum termo.

A cada dia foram feitas 131 requisições ao Twitter, uma para cada palavra listada na Tabela 9.1. Foram requeridas 50 postagens por cada termo. Ao final dessas coletas iniciais obteve-se 50.516 tweets.

4.2.0.2 Filtragens

As coletas de Twitter são disponíveis no formato *json* que é um tipo de arquivo que facilita a extração e filtragem dos dados. A extração dos dados nesse formato foi possível com o uso do pacote *RJSONIO* (LANG, 2014) do *Software* estatístico R, versão 3.1.1 (R Core Team, 2014). A rede social disponibiliza, para cada postagem coletada, uma série de informações. Além

do texto publicado tem-se a data e horário da publicação, dados do usuário e seu perfil e ainda, se autorizado, o georreferenciamento do *tweet*.

Uma breve leitura dos *tweets* coletados explicita a quantidade de postagens que são retornadas de maneira repetida pela rede. Considera-se razoável excluí-los pelo fato de designarem a ocorrência de apenas um crime não sendo interessante mantê-los sob a forma de contagens repetidas. Para tanto criou-se uma sequência de filtragens que são viáveis pelo fato de cada *tweet* e *retweet* ser demarcado por um número distinto, que será chamado nesse texto de ID. Chama-se retweet a divulgação ou compartilhamento de um determinado *tweet*. Por exemplo, se um usuário faz uma postagem cujo conteúdo é o *tweet* de um outro usuário ele fez um retweet e o próprio Twitter controla essa disseminação e indica se tratar de *retweet* explicitando o *tweet* 'mãe' através da sua numeração ID.

As filtragens na sequência exata em que foram feitas são as seguintes:

1. *Tweets* repetidos: Na presença de um mesmo ID mais de uma vez apenas a primeira ocorrência do ID permaneceria na base.

2. *Tweets* de um mesmo usuário com uma dada palavra: Nas situações em que um único usuário posta vários tweets com uma mesma palavra chave fica-se apenas com a primeira ocorrência deles. A leitura de tweets permitiu observar que, na maioria dos casos, essa repetição demonstra um usuário repetitivo o que também implicaria duplas-contagens de relatos de crimes.

3. *Retweets* repetidos: Essa corresponde à situação em que vários usuários repassam uma mesma postagem. Ocorre, na maioria das vezes, com tweets de veículos de comunicação, usuários famosos ou muito atuantes da rede ou relatos de crime que ganharam muita repercussão.

4. *Retweets* de *tweets* que caíram na coleta: Postagens de usuários repassando *tweets* que também constam na coleta são consideradas repetições de uma informação já obtida não sendo necessário, então, mantê-las.

Ao término da etapa de filtragem chegou-se a 26.503 tweets, cerca de 52% do inicial.

4.3 Classificação Manual

Em se tratando de Aprendizado Supervisionado em que busca-se pela definição de um classificador a partir de exemplos obtidos de um agente externo, viu-se a necessidade da classificação manual de textos.

Chama-se atenção para a dificuldade dessa etapa uma vez que há um elevado grau de subjetividade no uso da língua e linguagem e a necessidade de familiarização com essas últimas pelas características do microblogging de ser objetivo e atingir usuários com perfis muito distintos.

4.4 Contexto Temporal

Conforme descrito, os tweets inciais do estudo foram coletados em dias próximos. Espera-se que postagens de dias seguidos tenham relação entre si e caracterizem eventos e acontecimentos do seu período como decorrência da forte vinculação de informação pela Web e seu grande impacto no conteúdo das postagens produzidas.

Nos dias que serviram de amostra os tópicos populares foram “Ariel Castro”, o sequestrador e estuprador de “Cleveland”, o evento “Rock in Rio” no Rio de Janeiro, a absolvição do assassino “Pimenta Neves” dentre outros e os nomes mais citados foram “Putin”, “Obama”, “Messi” e “Robinho”.

4.4.1 Resultados das Coletas Iniciais

Apresenta-se a seguir alguns resultados referentes às coletas iniciais. Observou-se como os filtros realizados impactavam no total de *tweets* e explorou-se o percentual de *tweets* classificados manualmente como crime por cada termo pesquisado o que possibilita determinar quais expressões são mais sujeitas a serem empregadas em outros contextos. Além disso, estudou-se o percentual de informações de geo-referenciamento das postagens.

4.4.1.1 Impacto dos filtros no número de *tweets*

O impacto das filtragens em cada conjunto de *tweets* por termo permitiria diagnosticar termos com muitas repetições nas coletas ou termos com muitos *retweets* (possivelmente os divulgados pela mídia). Não foi possível fazer conclusões sobre filtros ou termos e, portanto, análises descrevendo-os serão omitidas nesse trabalho.

4.4.1.2 Termos com mais *tweets* classificados como crime

Também é natural questionar se existem expressões que tenham mais ou menos *tweets* efetivamente classificados como crime que as demais considerando-se apenas o cojunto de *tweets* que foi para classificação manual (considerada informação não repetida, distinta). Ou seja, há palavras cujo percentual de *tweets* classificados manualmente como crime é muito maior (ou menor) que as demais?

Para averiguar tal indagação apresenta-se na Figura 3 a esquerda uma nuvem de palavras. Essa constitui uma representação usual de estatísticas de texto e palavras e, no caso apresentado, cada termo aparece com uma fonte proporcional ao seu percentual de *tweets* classificados como crime. Além da fonte, há uma escala de cor visando a mesma descrição de sorte que as cores mais escuras coincidem com as palavras escritas com fonte maior.

Logo, os termos “saidinha de banco”, “saidinha de agencia”, “saidinha bancaria”, “tentativa de assalto”, “tentativa de furto”, “furtada” e “furtadas” aparecem como os mais associados a crime. As sete expressões são realmente consideradas muito específicas mas, sobretudo as cinco primeiras listadas. Apesar de serem termos muito bons para designar relatos de crime eles são mais raros entre os usuários e consequentemente entre as amostras que o Twitter retorna de sorte que há retorno de poucos *tweets* por requisição com tais termos. Os termos mais próximos

das bordas são os mais usados em expressões diferentes do contexto de interesse (crime) desse trabalho.

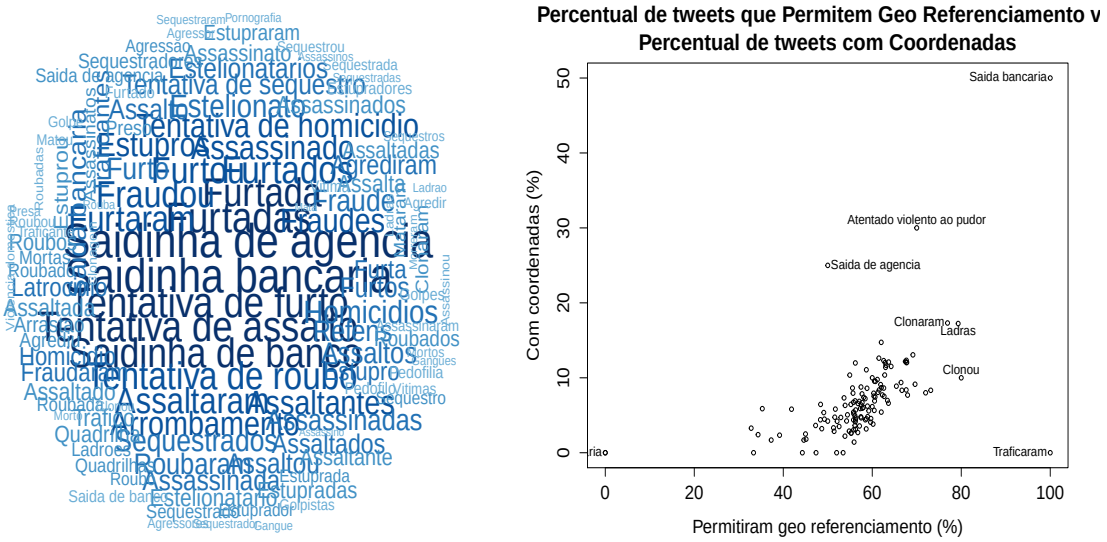

Figura 3 – Esquerda: Nuvem de palavras dos *tweets* classificados como crimes para termos pesquisados. Direita: Relação entre *tweets* que permitem geo-referenciamento e *tweets* com coordenadas.

4.4.1.3 *Tweets* com Georreferenciamento

Como previamente dito, o intuito desse trabalho é analisar relatos de crime a partir de *tweets* no estado de São Paulo. Sendo assim, eles requerem além do texto postado e dados do usuário a posição em latitude/longitude de onde o tweet foi postado.

Ressalta-se que poucos *tweets* tem essa informação de localidade. Uma postagem só pode ter sua localidade de postagem "rastreada" se ele for postado de um dispositivo móvel (celulares e *tablets*) e se o usuário autorizar a rede social a usar suas informações de GPS (do inglês *Global Position System*). Além do mais, o Twitter só repassa a latitude e longitude de uma postagem a terceiros que coletam dados da rede se autorizado pelo usuário autor do conteúdo.

Logo, um *tweets* com dado de georreferenciamento é naturalmente mais raro o que dificulta uma possível investigação espacial.

Na Figura 3 a direita segue gráfico de dispersão relacionando o percentual de *tweets* cujos usuários autorizam o geo-referenciamento e o percentual de *tweets* que efetivamente retornou com coordenadas de latitude e longitude por cada termo pesquisado.

Nota-se uma tendência linear discreta na nuvem de pontos central sendo que no eixo-x do gráfico há maior concentração de dados entre 50% e 70% e no eixo-y a faixa com maior frequência de dados é a de 0 a 10%.

A percepção da dificuldade de encontrar dados georreferenciados nas postagens com os termos pesquisados conduz a outras formas de coletas de *tweets*. A Sub-seção 4.4.2 visa a descrição dessas formas de coleta.

4.4.2 Coleta de Tweets por Município

Uma vez que são retornados poucos *tweets* com informação de latitude-longitude optou-se por uma forma de coleta por município do estado de São Paulo. Desse modo, variam-se as áreas de coleta para que além de informações pontuais tenham-se dados referentes às unidades políticas citadas.

Nesse caso, para cada município coletam-se *tweets* a partir do círculo tomando-se seu centróide e raio. Essa forma mantém coletas com *tweets* com latitude e longitude e acrescenta uma informação por polígono. Mesmo municípios muito pequenos ou com poucas postagens terão alguma informação quanto à postagem de dado termo em determinado dia já que se a requisição retornar sem qualquer *tweet* para o termo é porque simplesmente não houve postagem naquele período. Na Figura 2 a direita demarca-se as regiões de coletas circulares por município. É válido ressaltar que diminuem-se coletas nas regiões de fronteira do Estado problema anteriormente apresentado.

Esse tipo de coleta exige uma requisição de palavra para cada um dos 645 municípios em estudo. Logo, trata-se de 85.140 requisições para a rede social por dia. O Twitter permite apenas 180 requisições a cada 15 minutos de modo que a viabilidade de uma coleta nesses moldes fica comprometida. A solução foi diferenciar coletas em função da população do município e reduzir a quantidade de termos distintos coletados.

É natural pensar que municípios com maior população também tenham mais usuários da rede social em uso e, portanto, mais postagens com o contexto desejado. Pensou-se em coletas mais refinadas para os cinco maiores municípios do estado e coletas mais gerais para todos os demais 640 municípios.

Diz-se mais geral no intuito de fazer uma requisição com mais de uma palavra. Ou seja, coletam-se os *tweets* que tenham um ou outro termo dentre os listados a cada vez. Um exemplo corresponde à seguinte expressão a ser repassada à API do Twitter a qual também define bem a regra lógica sendo adotada na coleta:

"Assaltantes OR Assassinado OR Homicidios OR Golpes OR Sequestrada OR Assassinatos OR Pedofilos OR Assalta OR Estupro OR Roubos OR Arrastao OR Assassinada"

A coleta que recebe essa requisição retornará todos os *tweets* com pelo menos uma das palavras listadas. Entende-se como uma requisição econômica, com maior probabilidade de retornar o número total de *tweets* possível mas com termos variados.

Segundo dados do Censo 2010 do IBGE os cinco maiores municípios são a capital São Paulo, Guarulhos, Campinas, São Bernardo do Campo e Santo André.

Depois de algumas tentativas optou-se por diminuir a quantidade de termos pesquisados. Os termos retirados da lista inicial de 131 são demarcados com * na Tabela 9.1 do Anexo 9.1.

5 Métodos

5.1 Naive Bayes

Os classificadores Naive Bayes estão entre os algoritmos de aprendizado de maior sucesso para classificação de documentos de texto. Os métodos Naive Bayes são um conjunto de algoritmos de aprendizado supervisionado que aplicam conjuntamente o Teorema de Bayes e o pressuposto inocente ("naive") de independência condicional entre pares de atributos (LEARN, 2014).

Considere um documento D, cuja classe é C. Classifica-se D como sendo da classe cuja probabilidade a *posteriori* $P(C|D)$ é a maior, o que pode ser representado usando o Teorema de Bayes: $P(C|D) = P(D|C)P(C)/P(D)$

Como cada documento D está representado sob a forma de atributos vamos fazer a representação vetorial e genérica D = $(F_1, ..., F_n)$.

No contexto em estudo, D corresponde a um tweet em análise, a classe C pode admitir as categorias "crime" e "não crime" e F_i seriam as diferentes palavras (atributos ou características) empregadas para fazer a classificação. Logo:

$$\begin{aligned}
P(C|D) &= P(C|F_1, ..., F_n) \\
&= \frac{P(C, F_1, ..., F_n)}{P(F_1, ..., F_n)} \\
&= \frac{P(C)P(F_1, ..., F_n|C)}{P(F_1, ..., F_n)} \\
&\propto P(C) \cdot P(F_1, ..., F_n|C) \\
&\propto P(C) \cdot P(F_1|C) \cdot P(F_2, ..., F_n|C) \\
&\propto P(C) \cdot P(F_1|C) \cdot P(F_2|C, F_1) \cdot P(F_3, ..., F_n|C, F_1, F_2) \\
&\propto P(C) \cdot P(F_1|C) \cdot P(F_2|C, F_1) \cdot ... \cdot P(F_n|C, F_1, F_2, F_{n-1})
\end{aligned}$$

Mas, sob o pressuposto de independência condicional entre atributos dada uma determinada categoria C temos que $F_i \perp F_j$:

$$\begin{aligned}
&P(F_i|C, F_j) = P(F_i|C),\ i \neq j \\
&P(F_i|C, F_j, F_k) = P(F_i|C),\ i \neq j, k \\
&P(F_i|C, F_j, F_k, F_l) = P(F_i|C),\ i \neq j, k, l
\end{aligned}$$

De modo que

$$
\begin{aligned}
P(C|D) = P(C|F_1, ..., F_n) &\propto P(C, F_1, ..., F_n) \\
&\propto P(C) \cdot P(F_1|C) \cdot P(F_2|C) \cdot ... \cdot P(F_n|C) \\
&\propto P(C) \prod_{i=1}^{n} P(F_i|C)
\end{aligned}
$$

Há dois modelos probabilísticos para documentos usando a representação *bag of words* e a concepção Naive Bayes: o modelo Bernoulli e o modelo Multinomial. Em ambos os modelos há parâmetros para probabilidades a *priori* $P(C)$ e parâmetros para as probabilidades de cada palavra dada a classe do documento. Entretanto, a representação vetorial, discutida na Subsubseção 3.1.3.6 de um documento no modelo Bernoulli é com elementos binários de sorte que a entrada de um dado termo recebe 1 se ele aparece no documento em análise e 0 caso contrário. Já no modelo Multinomial as entradas dos vetores de documentos são inteiros que correspondem às frequências das palavras no documento em análise.

As probabilidades a *priori* das classes C são estimadas como a frequência relativa dos documentos de cada classe no total de documentos de treinamento. Todavia, as quantidades referentes às probabilidades condicionais das palavras dada a classe diferem. Os algoritmos para cada um dos métodos são apresentados a seguir:

Algoritmo para classificação de Texto Bernoulli:

1. Defina o vocabulário V
2. Para o banco de treinamento, faça as contagens:
   - 2.1) N: Número total de documentos
   - 2.2) N_k: Número total de documentos nomeados com a classe k, para k = 1, 2,..., K.
   - 2.3) $n_k(w_t)$: Número de documentos da classe C = k que contém o termo w_t, para todas as classes e para cada um dos termos no vocabulário
3. Estimar as probabilidades condicionais de cada termo dada a classe usando:

   $$\hat{P}(w_t|C = k) = \frac{n_k(w_t)}{N_k}$$

   Em que $n_k(w_t)$ é o número de documentos da classe $C = k$ em que w_t é observado. Ou seja, a frequência relativa dos documentos da classe $C = k$ que contem a palavra w_t.

4. E, para um documento D^i específico, a sua probabilidade condicionada à classe em termos das probabilidades condicionais de cada termo:

   $$P(D^i|C) = \prod_{t=1}^{|V|} [b_{it} P(w_t|C) + (1 - b_{it})(1 - P(w_t|C))]$$

   Em que b_{it} é indicadora do termo w_t no documento D^i.

5. Estimar as probabilidades a *priori* como a frequência relativa dos documentos que são da classe $C = k$:

   $$\hat{P}(C = k) = \frac{N_k}{N}$$

O algoritmo para classificação de Texto Multinomial é similar ao Bernoulli exceto pelos tópicos 2.3, 3 e 4 os quais são substituídos respectivamente por:

- 2.3) x_{it}: A frequência do termo w_t no documento D^i, computado para cada termo w_t em V.
- 3 - Estimar as probabilidades condicionais de cada termo dada a classe usando:

$$\hat{P}(w_t|C=k) = \frac{\sum_{i=1}^{N} x_{it} z_{ik}}{\sum_{s=1}^{|V|} \sum_{i=1}^{N} x_{is} z_{ik}}$$

$$z_{ik} = \begin{cases} 1, \text{ quando } D^i \text{ tem classe C=k} \\ 0, c.c \end{cases}$$

- 4 - Para um documento D^i específico a sua probabilidade condicionada à classe em termos das probabilidades condicionais de cada termo:

$$P(D^i|C) \propto \prod_{t=1}^{|V|} P(w_t|C)^{x_{it}}$$

Em que x_{it} é o número de vezes em que o termo w_t ocorre em D^i e z_{ik} é uma variável indicadora da classe k ($z_{ik} = 1$ quando D^i tem classe k e 0 caso contrário). Ou seja, a fração de vezes que o termo w_t aparece dentre todas as palavras nos documentos da classe k. É como se os dados fossem separados por classes e fossem obtidos os vocabulários em separado de cada uma dessas classes. Para cada um dos sub-bancos faz-se as frequências relativas de cada termo.

5.1.1 O Problema da Probabilidade Zero

Frequências relativas como estimadores de probabilidade têm o inconveniente de resultarem 0 quando contagens são nulas. Na classificação com Naive Bayes há um produto de probabilidades condicionais de palavra dada classe. Se alguma (ou várias delas) for zero o termo de verossimilhança é anulado o que significaria que a probabilidade do documento pertencer à classe em análise é zero. Ou seja, conclui-se que é impossível que um documento com tais atributos seja daquela classe porque apenas um (ou alguns) dos seus atributos não tem referência de classe no banco de treinamento.

"Só porque uma palavra não ocorre em uma classe no banco de treinamento não significa que não possa ocorrer em qualquer documento daquela classe"' (SHIMODAIRA, 2014). Sendo assim, nas situações nas quais um termo w_t não é observado em uma classe $C = k$ do banco de treinamento ainda assim deseja-se que $P(w_t|C = k) > 0$ o que leva a uma correção das probabilidades condicionais conhecida como Lei de Sucessão de Laplace, apresentada na Equação (5.1) que corresponde a somar 1 na frequência de cada palavra w_t no documento d_i:

$$\hat{P_{Lap}}(w_t|C=k) = \frac{1 + \sum_{i=1}^{N} x_{it} z_{ik}}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{N} x_{is} z_{ik}} \tag{5.1}$$

Shimodaira (2014) afirma que a representação Bernoulli é preferível para documentos de texto pequenos mas ressalva que a correção de Laplace é possível no caso Multinomial o que reforça a preferência pela segunda representação.

Nesse trabalho, classificadores concebidos pela técnica Naive Bayes utilizaram a representação Multinomial e a Correção de Laplace.

5.2 Árvore de Decisão

Árvores de Decisão são classificadores concebidos a partir da partição recursiva do espaço de atributos (ROKACH; MAIMON, 2005). A ideia é modelar uma sequência de decisões de um problema descrevendo graficamente as decisões a serem tomadas, e os resultados das combinações das decisões e eventos (TREEPLAN, 2014).

A árvore de decisão constrói uma regra de classificação a partir do processo conhecido como indução da árvore de decisão o qual leva em consideração como e quando parar de dividir os registros, conhecidos critérios de *splitting* e parada, respectivamente. Indutores de árvores de decisão são algoritmos que as constroem sendo os mais conhecidos: ID3, C4.5 do autor Ross Quinlan e o CART, de Breiman.

Em uma árvore de decisão tem-se uma raiz, nós e folhas as quais podem ser visualizadas na Figura 4. Cada nó denota um teste lógico que divide o espaço de atributos em dois ou mais sub-espaços, correspondendo o primeiro deles ao caso mais simples e representado na ilustração. A raiz é um nó sem aresta de entrada (chegada) representando o começo do problema, o nível superior da árvore apresentada. Todos os outros nós tem exatamente uma aresta de entrada sendo aqueles com aresta de saída chamados nós internos ou de teste (na Figura 4, eles correspondem aos retângulos no desenho). Os demais nós são chamados folhas, nós terminais ou nós de decisão e correspondem aos círculos da Figura 4.

Na árvore descrita os nós particionando o banco testam a existência de termos no texto em classificação. Desse modo, as arestas que saem a esquerda (linhas pontilhadas) dos nós indicam a continuidade da árvore para os registros em que observa-se o termo testado e as arestas a direita a continuidade da árvore para os registros sem aquela palavra. Sendo assim, os possíveis tweets "Homem indiciado por estupro assassina jovem em mata ciliar." e "Universitário confessa estupro." seriam classificados como CRIME.

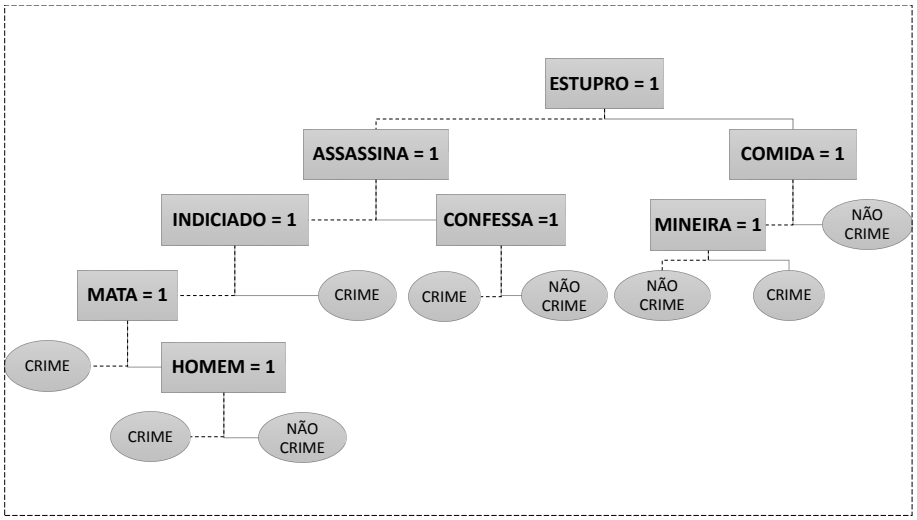

Figura 4 – Exemplo de Árvore de Decisão num contexto de classificação textual para categorias do tipo "crime" e "não crime".

Árvores de Decisão tem as vantagens de ser uma técnica auto-explicativa que pode ser acompanhada por usuários não profissionais, ser um método não paramétrico e ter resposta *online* rápida se o número de categorias é pequeno. Todavia, pode provocar super-ajustamento tanto ao conjunto de treinamento quanto a atributos irrelevantes e a ruídos.

Como típico problema de classificação por aprendizado de máquina, é natural perceber o *trade-off* entre a complexidade da regra criada e o percentual de erros cometidos. Tendo em vista a clara preferência

de usuários por árvores menos complexas e, portanto, mais compreensíveis. Rokach e Maimon (2005) ressalta que a complexidade da árvore tem efeito decisivo na sua acurácia sendo explicitamente controlada pelo critério de parada e o método de *prunning* adotado e acrescenta que a complexidade da árvore pode ser mensurada pelos seguintes critérios:

- número total de nós,
- número total de folhas,
- tamanho da árvore
- número de atributos usados
- todos os registros pertençam à mesma classe
- todos os registros tenham os mesmo valores de atributos

A etapa de *prunning* corresponderia à podagem da árvore de classificação. Ou seja, ela reavaliaria a necessidade da ramificação da árvore em diversos pontos dando indicativos da necessidade de diminuí-los e se for o caso, onde substituir uma sucessão de nós por uma folha.

Como antecipado, a divisão da árvore dá-se gerando nós. Um nó é criado se relativamente ao seu nó mãe ele apresenta melhora na classificação, ou seja, se o novo nó especifica melhor um grupo de observações que o seu nó mãe. Como sinônimo de 'melhor' adota-se o conceito de homogeneidade de classes o que se mensura com uma medida de impureza em cada nó. Por exemplo, considera-se o caso em que é possível sub-dividir uma árvore em dois nós, ambos com 10 registros cada, segundo critérios diferentes. No primeiro deles, chega-se a 5 registros da classe "crime" e 5 registros da classe "não crime". Já no segundo nó tem-se 8 e 2 registros de cada classe respectivamente. Logo, o nó obtido pelo segundo critério é mais homogêneo na sua classificação (baixo grau de impureza).

A impureza em um nó A pode ser definida a partir de uma função da proporção dos elementos de A que pertencem à classe i p_i^A por:

$$I(A) = \sum_{i=1}^{C} f(p_i^A)$$

O presente trabalho adota duas medidas de impureza de um nó: o Índice de Gini e o Índice de Informação com respectivas medidas de impureza $f(p_i^A) = -p_i^A(1 - p_i^A)$ e $f(p_i^A) = -p_i^A \log(p_i^A)$. O parâmetro de complexidade define o valor que quantifica o quão melhor é a entropia entre nós mãe e filhos. Por exemplo, a árvore considera vantajoso abrir mais um nó se a diferença de impureza entre os nós mãe e filhos for pelo menos o valor definido pelo parâmetro de complexidade.

Ainda pode-se adotar como critério para divisão de uma árvore um número mínimo de observações por nó, ou seja, abre-se um novo nó se a frequência de observações dele for maior ou igual a um valor especificado.

Como critério de parada pode-se adotar um número máximo de folhas ou folhas com dado nível de homogeneidade. A indução das árvores desse trabalho não incluiu critério de parada.

5.3 Máquina de Vetores de Suporte

Os classificadores Máquina de Vetores de Suporte, do inglês *Support Vector Machines* (SVM), foram apresentados em Boser, Guyon e Vapnik (1992) seguido do trabalho decisivo de Cortes e Vapnik

(1995). Enquadram-se nas técnicas de aprendizado de máquinas lineares e são continuamente desenvolvidos como promessa de classificadores que trabalham bem em espaços de grande dimensão e evitam super ajustamento.

Joachims (1998) ressalva as vantagens do uso de SVMs para classificação textual. Uma vez que SVMs utilizam proteção contra *overfitting* eles têm o potencial para trabalhar com grandes espaços de atributos. Acrescenta-se, ainda, o fato de que a maioria dos problemas de categorização de textos é linearmente separável possibilitando o emprego das chamadas SVMs lineares, o caso mais simples e computacionalmente mais rápido.

Cortes e Vapnik (1995) motivam sua abordagem apresentando trabalhos em que a construção de regras de decisão foi associada à construção de hiperplanos lineares em algum espaço demonstrando o quão intuitiva tal abordagem é, pelo menos em um primeiro momento. A ideia é construir um hiperplano que permita classificar bem os dados. Todavia, como se pode ver na Figura 5 a esquerda existem inúmeros possíveis hiperplanos entre as classes em estudo.

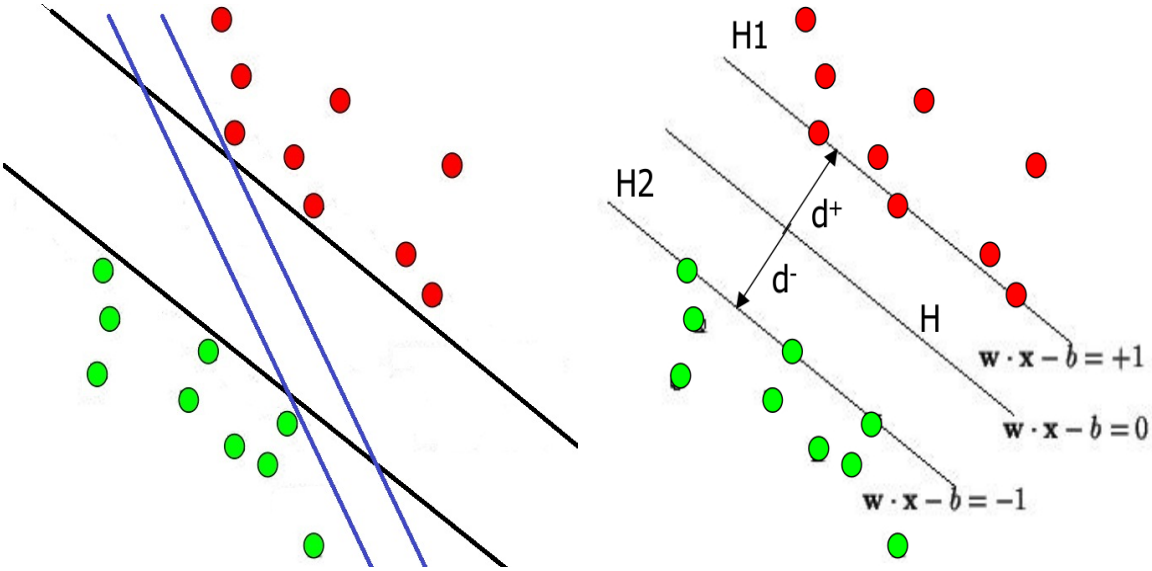

Figura 5 – Esquerda: Problema de classificação linear: Escolha do hiperplano ótimo. Direita: Representação de um problema de classificação linear para duas classes a partir de SVM com hiperplanos marginais H_1 e H_2 e hiperplano ótimo H.

Uma constatação a partir da Figura ilustrativa 5 é que a determinação da regra de classificação baseia-se apenas nas observações mais difíceis de classificar. Ou seja, tanto as observações em vermelho quanto as em verde que são mais próximas da classe oposta. Na Figura 5 a direita, essas observações estão sobre os hiperplanos H_1 e H_2. Essas são as observações com potencial para alteração da fronteira de decisão, uma parcela reduzida dentre os vetores disponíveis chamada de Vetores de Suporte.

Chama-se atenção para a reta paralela na cor preta próxima dos círculos vermelhos. Para os dados de treinamento descritos elas são capazes de discriminar as classes mas lembra-se que a técnica visa a construção de uma regra que classifique dados distintos aos usados na determinação da regra.

Uma nova observação da classe em vermelho ligeiramente diferente das demais de modo que passasse um pouco da fronteira seria classificada equivocadamente já que estaria no outro lado da região de separação sendo que, na realidade, ela tem atributos mais similares aos da sua classe. Sendo assim, percebe-se que para classificar bem ambas as classes o ideal é que o hiperplano separador esteja igualmente distante de ambas, ou ainda, o mais distante possível de cada uma delas simultaneamente.

Os desenvolvimentos que seguem formalizam as discussoẽs apresentadas. Essa construção introdutória recebe o nome de SVM de Margens Rígidas conforme será explicitado no decorrer do texto.

5.3.1 SVM de Margens Rígidas

Considere que para cada rótulo y_i do conjunto de treinamento T esteja associado um vetor de atributos ou características x_i no espaço dos dados X. Um classificador linear também chamado função discriminante linear tem a seguinte equação de hiperplano apresentada na Equação 5.2:

$$f(x) = w \cdot x + b = 0 \tag{5.2}$$

Onde o produto interno ou escalar entre os vetores w e x corresponde a $\sum_i w_i x_i$, $w \epsilon X$ é o vetor normal ao hiperplano descrito e o termo b, chamado de viés, translada o hiperplano com respeito à origem de modo que $\frac{b}{||w||}$ corresponde à distância do hiperplano em relação à origem (SAMMUT, 2010).

A função discriminante divide o espaço dos dados X em duas regiões: $w \cdot x + b > 0$ e $w \cdot x + b < 0$. A partir de $f(x)$ é possível obter um número infinito de hiperplanos equivalentes multiplicando w e b por uma mesma constante (hiperplano invariante à multiplicação por escalar não nulo) (LORENA; CARVALHO, 2007). Define-se, então o hiperplano canônico em relação ao conjunto de treinamento T como aquele em que w e b são escalados de maneira que os exemplos mais próximos de $w \cdot x + b = 0$ satisfaçam:

$$|w \cdot x_i + b| = 1$$

De modo que as classificações ficam:

$$w \cdot x_i + b \geq 1, \text{ se } y_i = 1$$
$$w \cdot x_i + b \leq -1, \text{ se } y_i = -1$$

Ou de modo resumido:

$$y_i(w \cdot x_i + b) - 1 \geq 0, \forall (x_i, y_i) \epsilon T$$

Considera-se, agora, os planos H_1 e H_2 representados na Figura 5 a direita:

$$H_1 : w \cdot x_i + b = 1$$
$$H_2 : w \cdot x_i + b = -1$$

A distância $\rho(x)$ entre um ponto x do plano H_1 a H deve ser a mesma que entre um ponto de H_2 a H, chamando-se margem, e sendo dada por:

$$\rho(x) = \frac{|w \cdot x + b|}{||w||} = \frac{1}{||w||} \tag{5.3}$$

Uma vez que w e b foram escalados de forma a não haver exemplos entre H_1 e H_2, a Equação 5.3 define a distância mínima entre o hiperplano separador e os dados de treinamento (LORENA; CARVALHO,

2007). A maximização dessa distância implica na minimização de $||w||$ ou também $\frac{1}{2}||w||^2$. Chega-se, portanto, no problema:

$$min_{w,b}\frac{1}{2}||w||^2 \tag{5.4}$$

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

$$y_i(w \cdot x_i + b) - 1 \geq 0, \forall i = 1, ..., n \tag{5.5}$$

São essas restrições que nomeiam a técnica descrita como SVM de margens rígidas. Mohri, Rostamizadeh e Talwalkar (2012) afirma se tratar de um problema de programação quadrática convexa o qual pode ser resolvido a partir de uma função Lagrangiana. A introdução dos multiplicadores de Lagrange α_i para relacionar a Função Objetivo 5.4 às restrições 5.5 conduz a:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i(y_i(w \cdot x_i + b) - 1) \tag{5.6}$$

O gradiente da função lagrangiana com respeito às variáveis w e b:

$$\nabla_w \mathcal{L} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{5.7}$$

$$\nabla_b \mathcal{L} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{5.8}$$

Tem-se então que o vetor Normal w solução do problema SVM, descrito na Equação 5.7, é uma combinação linear apenas dos vetores de treino em que $\alpha_i \neq 0$. Ou seja, a determinação do hiperplano ótimo está em função apenas de alguns vetores, os Vetores de Suporte.

E para garantir as condições de Karush-Kuhn-Tucker para a solução de um problema com restrição acrescenta-se ainda a condição de complementariedade:

$$\alpha_i[y_i(w \cdot x_i + b) - 1] = 0 \Rightarrow \alpha_i = 0 \cup y_i(w \cdot x_i + b) = 1 \tag{5.9}$$

O problema especificado nas Equações 5.4 e 5.5 tem a sua versão dual. Basta substituir a Equação 5.8 na Equação 5.6:

$$\begin{aligned}
\mathcal{L}(w, b, \alpha) &= \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i y_i w \cdot x_i - \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i \\
\mathcal{L}(w, b, \alpha) &= \frac{1}{2}\Big|\Big|\sum_{i=1}^{n} \alpha_i y_i x_i\Big|\Big|^2 - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i \\
\mathcal{L}(w, b, \alpha) &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)
\end{aligned} \tag{5.10}$$

De modo que o problema de otimização fica:

$$max_\alpha \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{5.11}$$

$$s.a \begin{cases} \alpha_i \geq 0, \forall i = 1, ..., n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{5.12}$$

A Equação 5.11 designa um problema de otimização em termos de produtos internos entre dados e as restrições das Equações em 5.12 são mais simples que na formulação anterior, a primal.

Da Equação 5.9 vê-se que $\alpha_i^* \neq 0$ somente para os dados que estão sobre os hiperplanos H_1 ou H_2, ou seja $y_i(w \cdot x_i + b) = 1$. Logo, os vetores de suporte, que determinam a equação do hiperplano separador sendo considerados os dados mais informativos do conjunto de treinamento, são aqueles sob os hiperplanos marginais $w \cdot x_i + b = \pm 1$.

Na prática, a maioria, senão todos os conjuntos de dados, não são linearmente separáveis (SAMMUT, 2010). Ou seja, não há w e b que satisfaçam as restrições do problema de otimização descrito pelas Equações 5.4 e 5.5. Isso se deve a diversos fatores entre eles a presença de ruídos e *outliers* nos dados ou à própria natureza do problema, que pode não ser linear (LORENA; CARVALHO, 2007). Todavia, há uma forma de adaptação para o problema de otimização em que permite-se que alguns dados violem as restrições da Equação 5.5. A Sub- Seção 5.3.2 apresenta a extensão das SVMs lineares de margens rígidas para lidar com conjuntos de treinamento mais gerais.

5.3.2 SVM de Margens Suaves

A versão relaxada a que se fez referência corresponde à introdução de variáveis de folga na Equação 5.5, prática comum em problemas de otimização:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ \forall i = 1, ..., n$$

No contexto em estudo, ξ_i mensura a distância que o vetor x_i viola a inequação desejada $y_i(w \cdot x_i + b) \geq 1$. A Figura 6 a esquerda explicita sua ideia. Na situação de margens rígidas quaisquer vetores passando dos hiperplanos marginais eram considerados *outliers*. Nessa formulação permitem-se vetores entre H_1 e H_2 cabendo analisar se estão do lado correto do hiperplano marginal. Ou seja, aqueles em que $0 < y_i(w \cdot x_i + b) < 1$ não são considerados *outliers* e são corretamente classificados. Os casos em que $\xi_i > 1$ são considerados erros no conjunto de treinamento e $\sum_{i=1}^{n} \xi_i$ representa um limite no número de erros de treinamento.

Caracteriza-se, por conseguinte, um problema de conflito de objetivos: deseja-se um hiperplano classificador com a margem o mais larga possível mas que limite o número total de "faltas/erros" decorrentes de *outliers* (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012):

$$min_{w,b,\xi} \left(\frac{1}{2}||w||^2 + C \left(\sum_{i=1}^{n} \xi_i \right) \right) \tag{5.13}$$

$$s.a \begin{cases} y_i(w \cdot x_i + b) \geq 1 - \xi_i, \forall i = 1, ..., n \\ \sum_{i=1}^{n} \xi_i \geq 0 \end{cases} \tag{5.14}$$

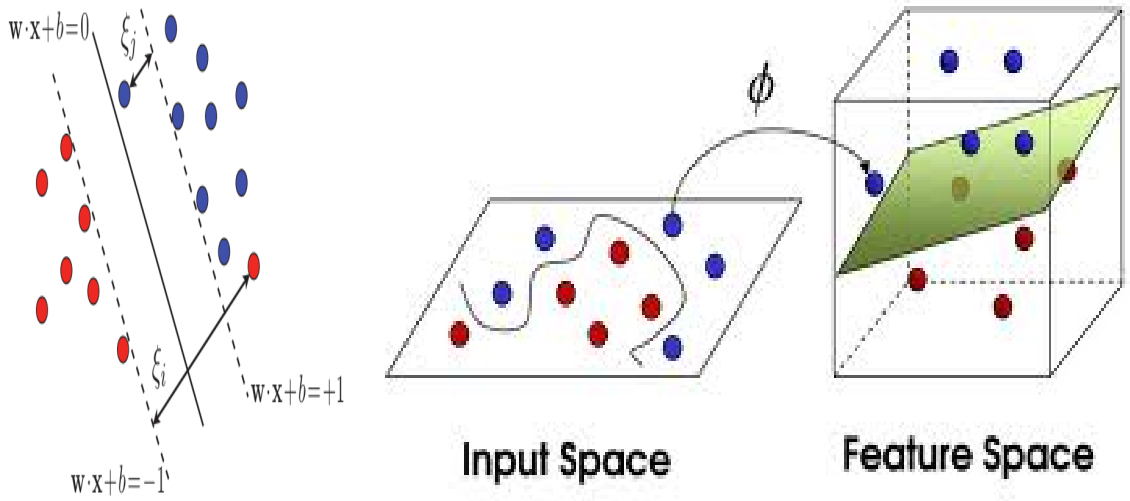

Figura 6 – Esquerda: Variáveis de folga em um problema de classificação linear. Direita: Rótulos não linearmente separáveis classificados segundo técnica SVM não linear.

Nas Equações 5.13 e 5.14 a constante C é um parâmetro definido pelo usuário chamado parâmetro de complexidade. Ela atua como um termo de regularização que impõe um peso à minimização dos erros no conjunto de treinamento em relação à minimização da complexidade do modelo (LORENA; CARVALHO, 2007). Em outras palavras, C relaciona o *trade-off* entre a maximização da margem de classificação e a minimização dos erros marginais. Chama-se atenção para a formulação mais geral de (5.13) com a expressão $\sum_{i=1}^{n} \xi_i^p, p \geq 1$ em que p designa a agressividade da penalização dada aos termos de folga e $p = 1$ e $p = 2$ são os valores mais empregados.

Similarmente ao desenvolvimento para SVM com margens rígidas chega-se à função lagrangiana:

$$\mathcal{L}(w, b, \alpha, \xi, \beta) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i] - \sum_{i=1}^{n}\beta_i\xi_i \tag{5.15}$$

Os gradientes e condição de complementariedade resultam:

$$\nabla_w \mathcal{L} = w - \sum_{i=1}^{n}\alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{n}\alpha_i y_i x_i \tag{5.16}$$

$$\nabla_b \mathcal{L} = -\sum_{i=1}^{n}\alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{5.17}$$

$$\nabla_{\xi_i} \mathcal{L} = C - \alpha_i - \beta_i = 0 \Rightarrow \alpha_i + \beta_i = C \tag{5.18}$$

$$\alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i]0, \forall i \Rightarrow \alpha_i = 0 \cup y_i(w \cdot x_i + b) = 1 - \xi_i \tag{5.19}$$

$$\beta_i\xi_i = 0 \Rightarrow \beta_i = 0 \cup \xi_i = 0 \tag{5.20}$$

O que permite replicar as conclusões da Seção anterior: as soluções do problema são combinações lineares do vetores de treinamento e os x_i associados aos $\alpha_i \neq 0$ são os Vetores de Suporte. Todavia há mais de um tipo de vetor de suporte para o SVM de Margens Suaves. Nas situações em que $\alpha_i \neq 0$ tem-se $y_i(w \cdot x_i + b) = 1 - \xi_i$. Se a variável de folga $\xi_i = 0$ é porque ela não quantifica uma distância entre o hiperplano marginal e o vetor estando então, o último sobre o primeiro. Quando $\xi \neq 0$ a Equação 5.20

permite deduzir que $\beta_i = 0$ e, pela Equação 5.18, $\alpha_i = C$ de sorte que trata-se de um vetor de suporte entre os hiperplanos marginais, um *outlier*.

A forma dual do problema segue em função da definição de w em termos das variáveis duais e aplicando as restrições descritas nas Equções 5.16 a 5.20 o que resulta na mesma forma da Equação 5.10. Acrescenta-se a isso as restrições $\alpha_i \geq 0$ e $\beta_i \geq 0$ que equivale a $\alpha_i \leq C$. Nos casos em que o conjunto de dados não é linearmente (ou aproximadamente) separável os dados não são satisfatoriamente divididos por um hiperplano. Nessas situações foi desenvolvida uma teoria em que o espaço de atributos é projetado para uma dimensão maior de sorte que nesse novo espaço é possível uma separação linear dos rótulos. A Figura 6 à direita ilustra tal situação. As SVMs não lineares não são usualmente recomendadas em contextos de muitos registros e atributos. Por conseguinte, seu desenvolvimento teórico será omitido podendo ser consultado em Cortes e Vapnik (1995), Lorena e Carvalho (2007), Sammut (2010), Mohri, Rostamizadeh e Talwalkar (2012) e Flach (2012).

Nesse trabalho, a partir do pacote *e1071* (DIMITRIADOU et al., 2008) do *Software* R, versão 3.1.1, estudou-se a factibilidade no uso de classificadores não lineares com *kernel* polinomial e radial. Lembra-se que o contexto de postagens de Twitter, rede social com características muito próprias, se difere dos cenários de classificação de texto maior e estruturado. Esse diferencial justificou a exploração de mais formas de categorização.

Como diversas vezes citado a solução para classificadores por Vetores de Suporte necessita apenas de produtos internos das observações. Mohri, Rostamizadeh e Talwalkar (2012) apresentam o uso de *kernels* como metologia natural na extensão de algoritmos SVMs ou quaisquer outros que dependam apenas de produtos internos entre pontos amostrais para definir fronteiras de decisão não lineares. O autor define um *kernel* em χ como uma função $k : \chi \times \chi \to \mathbb{R}$. A ideia é, então, definir um *kernel* de modo que para quaisquer dois pontos $x, x^{'} \epsilon \chi$ $K(x, x^{'})$ seja igual ao produto interno de vetores $\phi(x)$ e $\phi(x^{'})$:

$$k(x, x^{'}) =< \phi(x), \phi(x^{'}) >$$

para alguma função de mapeamento $\phi : \chi \to \mathbb{H}$ sendo $\mathbb{H}$ o espaço de atributos.

Mohri, Rostamizadeh e Talwalkar (2012) cita as vantagens de eficiência e flexibilidade no uso de *kernels* por evitar o cálculo de produtos internos em $\mathbb{H}$ e não exigir a definição explícita ou cálculo da função ϕ.

De uma forma geral, os *kernels* correspondem a funções que generalizam o produto interno sendo interpretado como uma medida de similaridade entre elementos do espaço de atributos.

Existem inúmeros possíveis *kernels* a serem aplicados no problema desse trabalho.

Também utilizando-se o pacote *e1071* optou-se por testar o polinomial e o radial dados, respetivamente, por:

$$k(x, x^{'}) = (\gamma x x^{'} + c)^d, \ c > 0, \ d \, \epsilon \mathbb{N}, \ \forall x, \ x^{'} \epsilon \, \mathbb{R}^N$$
$$k(x, x^{'}) = exp(-\gamma |x - x^{'}|^2), \ \forall x, \ x^{'} \epsilon \, \mathbb{R}^N$$

Mohri, Rostamizadeh e Talwalkar (2012) afirma que SVMs com *kernels* polinomial são muito empregados e que o *Radial Basis Function* (também chamado Gaussiano) e corresponde a uma combinação linear positiva de *kernels* polinomiais de todos os graus $n \geq 0$. James, Witten e Hastie (2014) acrescenta ainda a consideração que o *kernel* radial tem um comportamento local de modo que só observações de treinamento vizinhas interferem na classificação de uma nova observação como decorrência do emprego de distâncias Euclidianas.

O pacote *e1071* fornece ainda o *kernel* Sigmoide o qual não foi empregado por ser citado como um algoritmo de aprendizado semelhante a redes neurais simples.

5.4 Análise de Componentes Principais

No presente trabalho problemas de natureza computacional são considerados comuns dadas as características da grande dimensão de dados textuais e a grande quantidade de informação disponível e vinculada na *web*.

Originalmente, dispõe-se de um banco de dados de 26503 registros cuja categorização em "crime" e "não crime" foi feita manualmente conforme previamente dito. Esse banco todo apresenta 32536 termos (palavras) diferentes. Para a implementação das técnicas descritas, optou-se por trabalhar com bancos de dados reduzidos. Em um deles, foram amostradas 5000 (cinco mil) do total de observações implicando em vocabulário de 11341 termos distintos. Há também um banco correspondendo a 10000 (dez mil) valores amostrados do total disponível totalizando 17789 palavras diferentes. Logo, explicita-se como a limitação computacional restringe a potencialidade das técnicas em uso. Utiliza-se 20% ou 50% de todo o vocabulário/contexto já classificado pela impossibilidade da máquina.

De forma complementar, pensou-se no uso da técnica de Análise de Componentes Principais (PCA) conjuntamente com as demais técnicas apresentadas no decorrer do texto.

A PCA foi concebida por Karl Pearson em 1901 com o intuito de explicar a estrutura de variância e covariância de um vetor aleatório de p variáveis aleatórias. Para tanto, procede-se com a construção de combinações lineares das variáveis originais. Tais combinações são chamadas Componentes Principais e são não correlacionadas entre si (MINGOTI, 2005). Em geral, intui-se substituir a informação contida nas p variáveis por um número k de Componentes Principais menor que o número de variáveis. Dessa maneira, há uma redução na dimensão do problema em estudo já que substitui-se uma matriz de variância/covariância por outra menor e com menos informação. Mingoti (2005) acrescenta que, uma vez determinadas as componentes principais, os valores numéricos de cada uma delas (escores) podem ser calculados para cada elemento amostral possibilitando o seu emprego em técnicas estatísticas usuais como ANOVA ou Regressão e, nesse trabalho, técnicas de Aprendizado Supervisionado. O desenvolvimento teórico da técnica não será apresentado nesse texto podendo ser consultado em (HÄRDLE; SIMAR, 2007).

Para esse trabalho, a ideia é criar Componentes dos termos do vocabulário em estudo para diminuir a dimensão dos dados, ou ainda, trabalhar com combinações lineares dos termos observados originalmente. As análises que seguem no texto explicitam que o uso da PCA viabilizou o emprego de várias das técnicas em tempo razoável tendo sido de fundamental importância. Em geral, pensou-se no uso da PCA empregando o número de componentes necessárias para explicar 80% e 90% da variabilidade total dos termos.

6 Resultados

Nesse capítulo são apresentados os resultados dos classificadores para cada um dos diferentes cenários que cada técnica de classificação permite.

Dentre os possíveis bancos a serem utilizados, descritos na Seção 5.4, foram utilizados dois: a versão mais reduzida, de 5000 postagens e, para os casos em que a definição de cenários é mais delicada usou-se o banco de dados de tamanho 10000.

Para cada técnica, a avaliação do classificador obtido dá-se através de validação cruzada de tamanho 10. Isso significa que particiona-se o banco em 10 partes de tamanhos parecidos e para cada um dos cenários de todas as técnicas será concebido um classificador para cada uma das 10 partes. Por exemplo, no classificador Naive Bayes são 12 cenários diferentes. Para cada um deles constrói-se um classificador com as suas características tomando cada uma delas (1/10 do banco) como banco de teste e todas as demais juntas (9/10) como banco de treino. A validação cruzada simula contextos com vocabulários diferentes para testar o desempenho classificador.

Todas as técnicas tem dois argumentos em comum os quais dizem respeito às características do *Corpus*: a ponderação dada por termo e o número mínimo de frequência do termo para que entre na análise. Esse último critério é interessante por possibilitar avaliar o ganho de eficiência e tempo computacional ao se excluir termos pouco comuns. Considera-se como prática comum em mineração de texto. A etapa de ponderação dos termos é realizada através do pacote *tm* (FEINERER; HORNIK, 2014) do *Software* estatístico R, versão 3.1.1 (R Core Team, 2014) e que implicará em classificadores com preferência por termos mais ou menos comuns dependendo do peso adotado. Considerando todas as possíveis combinações desses critérios mínimos pode-se pensar em 12 diferentes cenários discriminados na Tabela 3.

	Frequência Mínima do Termo		
Peso	1	3	5
TF	1	2	3
IDF	4	5	6
TF-IDF	7	8	9
Dummy	10	11	12

Tabela 3 – Numeração dos cenários ao se variar a ponderação dada por termo e o número mínimo da frequência do termo.

Nesse trabalho, a avaliação da capacidade de generalização do classificador será baseada na medida F1 visto que engloba as medidas precisão e a revocaçãosua as quais foram descritas no Capítulo 3.

A ideia é comparar os cenários de cada técnica sobre o critério F1 de generalização sendo o critério tempo computacional para classificação sempre abordado como consequência do interesse por um método de classificação quase *online*.

Para tanto, procede-se com uma análise visual do valor de F1 por cenário seguido de testes estatísticos de hipótese visando formalizar as comparações entre cenários. Em geral, apresentam-se ANOVAs de um fator para comparação de médias de múltiplas populações e testes Shapiro Wilk e Levene (do pacote *lawstat* (<KINOGUCHI@UCDAVIS.EDU>, 2013)) para averiguar a obediência dos pressupostos de Normalidade e Homocedasticidade da ANOVA. Adianta-se que a não obediência dessas hipóteses na imensa maioria das análises descritas implica no uso do Teste de Kruskal Wallis para comparação de médias, análogo não paramétrico da Análise de Variância e, em seguida, teste de comparações múltiplas

Kruskal Multiple Comparisons do pacote *pgirmess* (GIRAUDOUX, 2014) do R.

6.1 Naive Bayes

Os resultados dos classificadores da técnica Naive Bayes multinomial com correção de Laplace estão na Figura 7. Dada a sua simplicidade não há argumentos além da ponderação dada por termo e o número mínimo da frequência do termo para que entre na análise. As combinações deles dois a dois correspondem a doze modelos diferentes exatamente os que constam na Tabela 3.

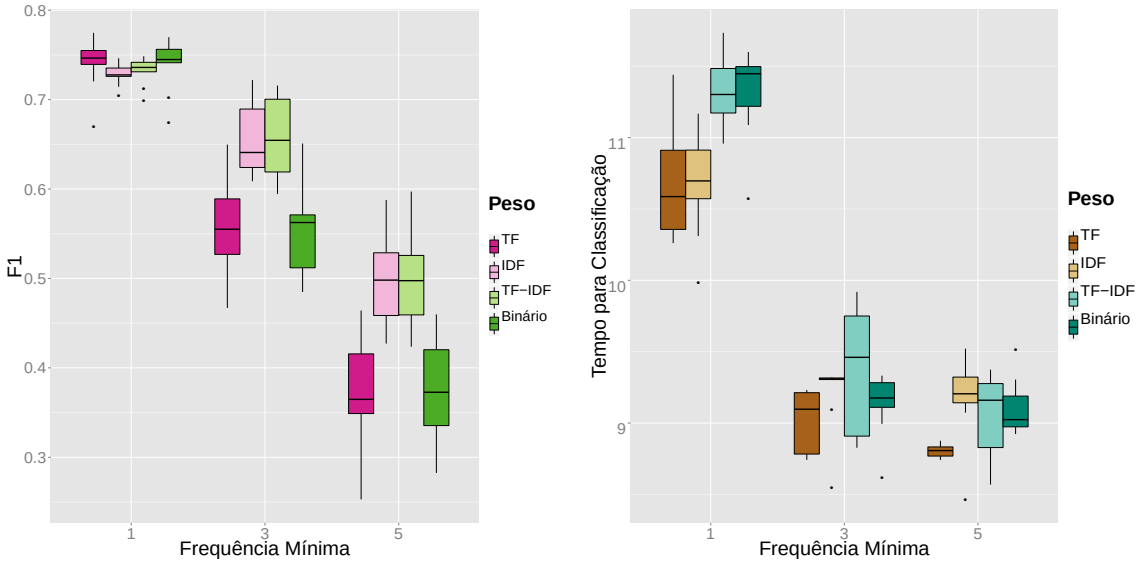

Figura 7 – Medida F1 para generalização (esquerda) e tempos necessário (direita) para construção do classificador por Naive Bayes.

Da Figura 7 vê-se que a diminuição do vocabulário (exigência de um número mínimo de frequência para que um dado termo permaneça na análise) prejudica a qualidade do ajuste, mas diminui o tempo computacional. Basta observar que as caixas decaem no sentido de crescimento do eixo horizontal dos gráficos. Logo, quanto mais se exige de frequência de postagem por termo pior fica o classificador. Vê-se mais semelhança entre as ponderações TF e Binário e entre IDF e $TF - IDF$ além do fato de que os tempos computacionais para classificação dos cenários com frequências mínimas 3 e 5 serem parecidos. Opta-se então pelos classificadores Naive construídos com termos que tem pelo menos três ocorrências no banco por desempenharem satisfatoriamente segundo a medida F1 em um tempo razoavelmente inferior aos classificadores com o vocabulário todo (aqueles de melhor desempenho).

Todavia, a exploração visual exige um rigor estatístico na comparação das médias das medidas de interesse. A ANOVA comparando as médias das medidas F1 por cada cenário é descrita na Tabela 4. Como dito, na Tabela 5 explicita-se a impossibilidade do seu uso pela não obediência dos seus pressupostos. Nessa mesma Tabela vê-se que o teste de Kruskal-Wallis permite afirmar que há diferenças estatisticamente significantes entre as médias da estatística F1 por cenários.

Cabe, então, questionar quais são os cenários que diferem dos demais dando enfoque para aqueles cujo desempenho mostrou-se satisfatório segundo F1 e o tempo para classificação (cenários com frequência mínima 3 e peso IDF ou $TF - IDF$ chamados cenários 5 e 8, respectivamente). O teste de comparações

Técnica		Gl	Soma de Quad.	Quad. Médio	F	P-valor
Naive-Bayes	Cenário	11	2.0338	0.1849	88.17	10^{-16}
	Resíduos	108	0.2265	0.0021		
Árvore	Cenário	287	0.448	0.001560	0.686	1
	Resíduos	2592	5.892	0.002273		
Árvore CenReduz	Cenário	47	0.0034	0.0000716	0.033	1
	Resíduos	432	0.9479	0.0021941		
Árvore PCA 80%	Cenário	47	0.2145	0.004563	2.928	10^{-9}
	Resíduos	432	0.6732	0.001558		
Árvore PCA 90%	Cenário	47	0.3595	0.007649	4.167	10^{-16}
	Resíduos	432	0.7931	0.001836		
SVM Linear	Cenário	131	25.622	0.19559	175.2	10^{-16}
	Resíduos	1188	1.326	0.00112		
SVM Linear PCA 80%	Cenário	131	82.84	0.6324	570.7	10^{-16}
	Resíduos	1188	1.32	0.0011		
SVM Linear PCA 90%	Cenário	131	82.31	0.6283	508.8	10^{-16}
	Resíduos	1188	1.47	0.0012		

Tabela 4 – Análise de Variância para testar diferença nas médias de F1 por cenário de cada técnica adotada.

Técnica	$Shapiro - Wilk$		Levene		Kruskal-Wallis		
	Estatística	P-valor	Estatística	P-valor	Estatística	Gl	P-valor
Naive Bayes	0.9853	0.2208	3.1034	0.0012	106.6396	11	~ 0
Árvore	0.9777	~ 0	0.0818	~ 1	191.8080	287	~ 1
Árvore CenReduz	0.0523	~ 1	0.0523	~ 1	1.7339	47	~ 1
Árvore PCA 80%	0.9776	~ 0	0.9447	0.5797	105.9242	47	~ 0
Árvore PCA 90%	0.9901	0.0026	0.3248	~ 1	145.6579	47	~ 0
SVM Linear	0.9919	~ 0	1.6205	~ 0	880.5813	131	~ 0
SVM Linear PCA 80%	0.9746	~ 0	3.2562	~ 0	985.1233	131	~ 0
SVM Linear PCA 90%	0.9632	~ 0	2.8194	~ 0	917.7607	131	~ 0

Tabela 5 – Testes de Hipóteses Shapiro-Wilk para Normalidade, Levene para Homocedasticidade e Kruskal-Wallis como análogo não paramétrico da ANOVA para as técnicas empregadas nesse trabalho.

múltiplas não paramétrico que sucede o teste de Kruskal Wallis tem os resultados apresentados na Tabela 6. Nela, as entradas denotam o par de cenários que foi comparado entre si através do teste e o símbolo ✓explicita as entradas que nele apresentaram diferenças estatísticas significativas. Não existe diferença significativa entre os cenários 5 e 8 visto que não há sinal para a entrada (5,8). Diz-se que para frequência mínima 3 as ponderações IDF e $TF - IDF$ não implicam classificadores com capacidade de generalização diferentes.

O critério tempo computacional desempata a análise de modo que o classificador Naive-Bayes com melhor desempenho foi construído apenas com os termos do vocabulário com frequência maior ou igual a 3 e ponderação IDF. O F1 médio na validação cruzada foi de 0,6568355 e o tempo total médio para classificação 9,2138 segundos.

6.2 Árvore de Decisão

6.2.1 Usual

As regras de classificação concebidas pelo método da Árvore de Decisão foram feitas no pacote *rpart* (THERNEAU; ATKINSON; RIPLEY, 2014) o qual induz a árvore segundo as descrições dadas em (BREIMAN et al., 1984). Nesse caso os cenários variam de acordo com os seguintes parâmetros:

	Cenário											
Cenário	1	2	3	4	5	6	7	8	9	10	11	12
1	.	✓	✓			✓			✓		✓	✓
2	.	.								✓		
3	.	.	.	✓	✓		✓	✓		✓		
4	.	.	.	.	✓				✓			✓
5	.	.	.	.	.							✓
6	.	.	.	.	.	.	✓			✓		
7	.	.	.	.	.	.	.		✓			✓
8	.	.	.	.	.	.	.	.				✓
9	.	.	.	.	.	.	.	.	.	✓		
10	.	.	.	.	.	.	.	.	.	.	✓	✓
11	.	.	.	.	.	.	.	.	.	.	.	.
12	.	.	.	.	.	.	.	.	.	.	.	.

Tabela 6 – Matriz relacionando comparações múltiplas dos cenários dois a dois da técnica Naive Bayes.

- ponderação dada por termo: TF, IDF, $TF - IDF$ e Binário
- número mínimo de postagens do termo para que entre na análise: 1, 3 e 5
- o critério para divisão dos nós: Gini e Informação
- número mínimo de observações por nó: 10, 20 e 50
- parâmetro de complexidade: $10^{-10}, 10^{-7}, 10^{-5}, 10^{-3}$

Lembra-se que os três últimos critérios servem ao propósito de determinar formas para indução e parada da divisão da árvore e todos foram apresentados na Seção 5.2.

Ao se variar os cinco critérios descritos, chega-se ao total de 288 possíveis cenários para análise. Há clara dificuldade de distinção e avaliação dos mesmos de modo que optou-se por avaliar os cenários a partir dos testes estatísticos de hipótese. Sabendo da impossibilidade do uso da Análise de Variância pelos resultados nas Tabelas 4 e 5 emprega-se o teste de Kruskal Wallis. Os resultados na Tabela 5 elucidam que mesmo numa metodologia não paramétrica não foi possível discriminar os melhores classificadores da técnica Árvore de Decisão.

Considerou-se investigar inicialmente *boxplots* de F1 para os níveis de cada um dos critérios que compõem os cenários. Nos gráficos apresentados na Figura 8 há indícios de que Gini seja mais adequado para indução da árvore de decisão (mediana ligeiramente superior em relação à mediana de "informação"). Da Figura 8 a direita admite-se que um número mínimo de 20 casos por nó corresponde a árvores com maior capacidade de generalização. Os demais critétrios também foram avaliados de maneira univariada não apresentando diferenças nítidas e aparecem na Figura 32 na Sub-Seção 9.2 em Anexo.

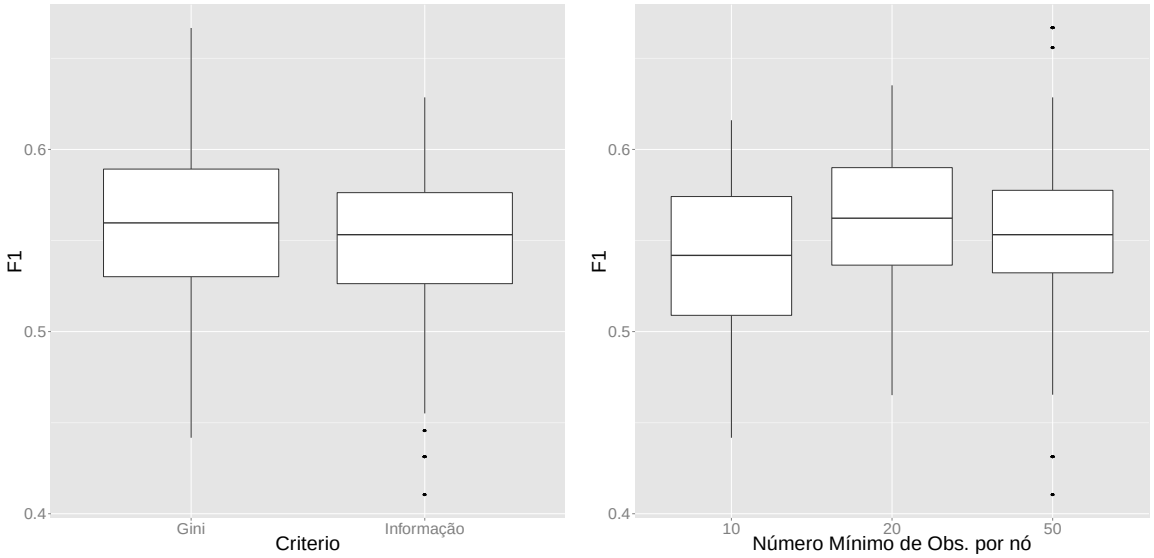

Figura 8 – *Boxplots* para medida Gini de quebra de nós (esquerda) e critério número mínimo de observações por nó (direita).

Logo, entende-se que a procura do classificador ótimo dá-se nos cenários em que o número mínimo de casos por nó é 20 com critério Gini para quebra de nós. A Figura 33 representando o desempenho de F1 nos cenários em que os demais critérios são variados situa-se na Sub-Seção 9.2 em Anexo já que conclusões acerca da generalização do classificador são difíceis.

Em contrapartida, o tempo total empregado na classificação é claramente distinto entre cenários de modo que esse, então, será o critério de desempate. Na figura 9 vêem-se 16 caixas com baixo tempo de categroização. Uma vez que a diferença de resultados é muito grande as caixas aparecem de modo reduzido justificando a observação exclusiva dos 16 cenários mais favoráveis na Figura 9 a direita. Nessa última também é clara a superioridade dos cenários com menor vocabulário (frequência mínima 5) de modo que chega-se na especificação de mais um critério. Para a escolha da ponderação a ser adotada cita-se a nítida similaridade dos cenários. Os tempos para classificação associados ao peso TF tem valores (em segundos) 0.1340, 0.1345, 0.1355 e 0.1360 ao passo que ao peso Binário tem valores 0.1320, 0.1290, 0.1310, 0.1340. A média de tempos é, portanto, muito próxima (0.134 e 0.132, respectivamente) mas há leve superioridade do cenário com peso Binário defindo-se assim o critério de ponderação.

Para evitar superajustamento da regra da árvore no banco de treinamento após a sua definição é comum submetê-la a um processo chamado *prunning*. Nele, a ideia é reduzir o número de nós para que as regras de classificação sejam construídas com menos atributos. Essa etapa de aprimoramento da árvore baseou-se no banco de 5000 dados divididos amostrando-se 500 para teste e 4500 para construção da árvore e melhora. Seguindo descrições do próprio pacote *rpart* e (CHU, 2013) o *prunning* da árvore toma como base o gráfico que relaciona a queda do parâmetro de complexidade pelo número de nós e o erro relativo o qual pode ser visualizado na figura a seguir.

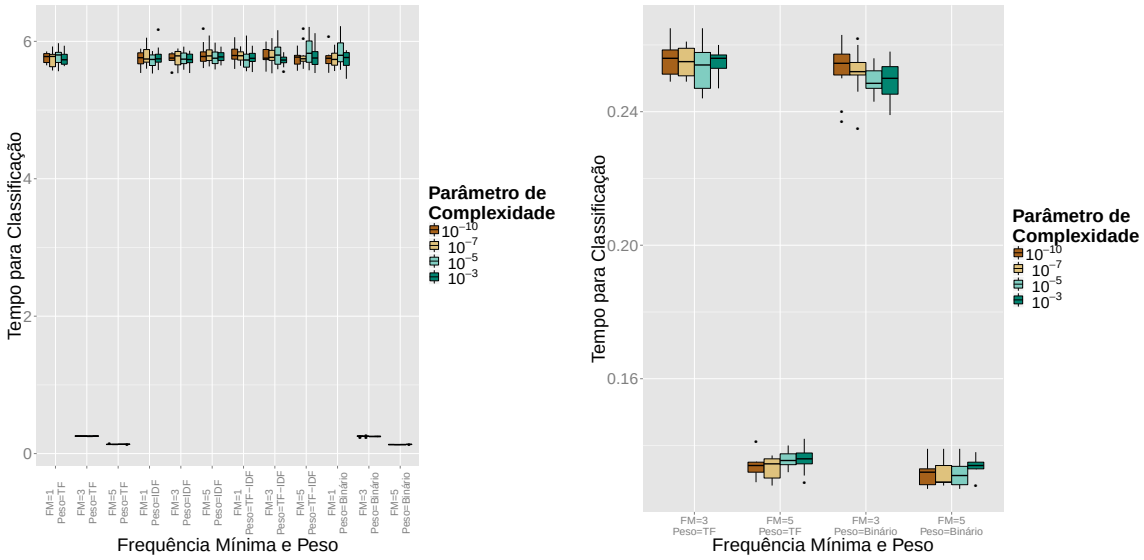

Figura 9 – *Boxplots* comparando tempo total empregado na classificação de Árvores de Decisão fixando critério número mínimo de observações por nó igual a 20 e medida Gini de quebra de nós para todos os cenários (esquerda) e apenas para os cenários com menor tempo de classificação (direita).

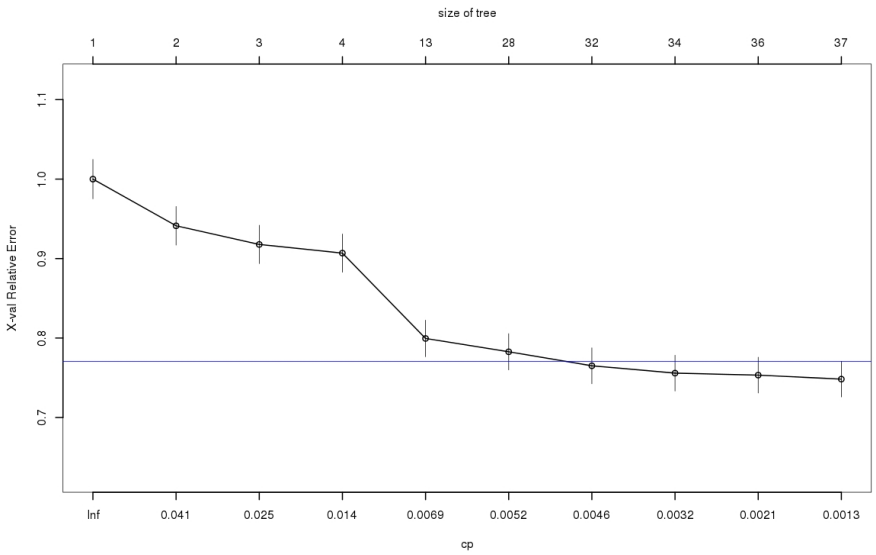

Figura 10 – Relação entre parâmetro de complexidade da árvore e seu número de nós para ajudar na etapa de *prunning* da árvore visando diminuir o número de parâmetross envolvidos.

Cabe discutir que apesar da árvore ter tantos possíveis parâmetros que controlam critérios de crescimento e parada da mesma o parâmetro mais delicado da análise corresponde ao parâmetro de complexidade. É visando a sua estimação que faz-se validação cruzada de tamanho 10 de sorte que a medida de erro relativo diz respeito ao erro relativo designa a média de classificações incorretas nos bancos de validação cruzada.

Chu (2013) afirma que o melhor valor do parâmetro de complexidade cp é o primeiro valor a esquerda para o qual a linha de erro relativo encontra-se inteiramente abaixo da linha pontilhada. Na árvore induzida esse valor corresponde a 0.0046. Então, atualiza-se a árvore induzida com esse novo valor de parâmetro de complexidade através da função *prune* do pacote em uso chegando-se na árvore final que

fica como classificador definitivo.

Para testar sua capacidade de generalização os 500 dados do conjunto de teste foram classificados. A relação entre precisão e revocação resultou na estatística F1 = 0.591 com tempo de classificação 0.147 segundos demonstrando tratar-se de um bom classificador.

6.2.2 Com Análise de Componentes Principais

Como previamente discutido na Seção 5.4 a técnica de Análise de Componentes Principais foi empregada nesse trabalho com a expectativa de reduzir a dimensão do problema em estudo e melhorar o tempo para classificação do banco de teste adotado. Para tanto, foram feitas análises com Componentes Principais de duas maneiras diferentes: usando um número de componentes que explicasse 80% e 90% da variabilidade total dos termos. Sabendo das dificuldades de simulação da Árvore de Decisão (elevado tempo computacional) optou-se pela diminuição do números de cenários em avaliação de modo que os níveis de cada critério permanecem como antes exceto pela fixação do número mínimo de observações por nó em 20 e critério Gini para divisão de nós.

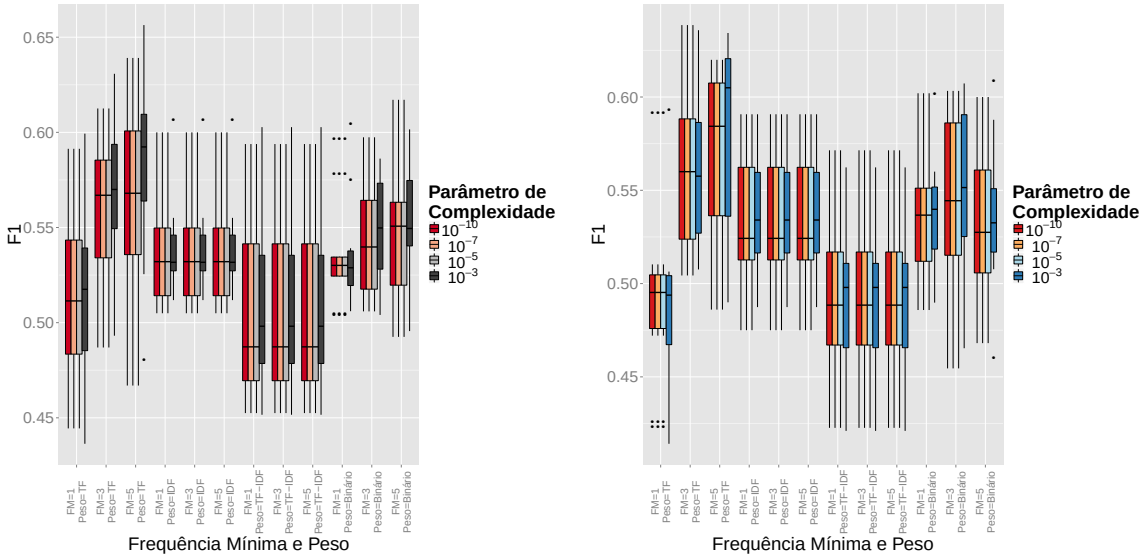

Figura 11 – *Boxplots* por cenários da técnica Árvore de Decisão com critério Gini para divisão dos nós e número mínimo de 20 observações por nó para medida F1 de capacidade de generalização com PCA 80% (esquerda) e PCA 90% (direita).

As Figuras 11 demarcam muita similaridade no desempenho dos classificadores tomando PCA 80% ou PCA 90%. Nas duas imagens para os pesos TF e Binário vê-se que o critério frequência mínima do termo tem papel muito grande e intermediário nas medidas F1 respectivamente. Nesses casos, que correspondem às caixas las laterais das imagens, o valor 10^{-3} para o parâmetro de complexidade (CP) parece superior aos demais sendo que para o peso TF essa diferença é mais acentuada. Já nos cenários com ponderação TF e $TF - IDF$ as caixas não parecem se diferenciar em função dos demais critérios e sendo esse último característico dos classificadores com pior classificação.

A Tabela 12 na Sub-Seção 9.2 em Anexo apresenta as medianas dos *boxplots* apresentados o que explicita o parágrafo anterior. Com base nela, observa-se que 23 dos 48 cenários são melhores com PCA 90% mas trata-se de uma diferença de valores tão pequena que não é possível concluir que o percentual de variabilidade explicada distingue bem a medida F1. Recorre-se à avaliação do tempo para classificação.

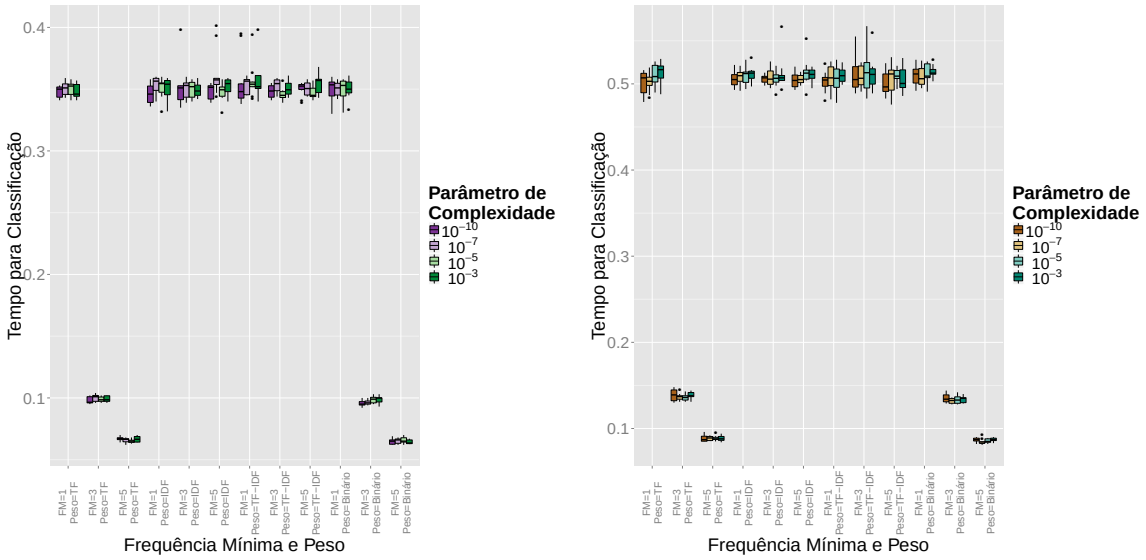

Figura 12 – *Boxplots* por cenários da técnica Árvore de Decisão com critério Gini para divisão dos nós e número mínimo de 20 observações por nó para tempo para classificação com PCA 80% (esquerda) e PCA 90% (direita).

Na Figura 12 é nítida a diferença entre os dois tipos de PCA no que diz respeito ao tempo para classificação. Ao contrário da medida F1 as caixas se diferenciam para um mesmo valor de peso e frequência mínima do termo e os menores tempos ocorrem quando trabalha-se com o menor vocabulário possível (FM = 5). Ainda assim, os menores tempos para classificação foram dos classificadores com ponderação Binária e TF o que vai ao encontro das análises para F1.

Pela análise conjunta das quatro figuras e respectivas medidas levam a concluir que a Árvore de Decisão com componente que explicam 80% da variabilidade total dos dados, com parâmetro de complexidade igual a 10^{-3}, freqência mínima igual a 5, peso TF, critério Gini para divisão dos nós e número mínimo de 20 observações por nó é a que tem melhor desempenho. Essa configuração de árvore tem F1 médio de 0.581 e 0.066 segundos para classificação, em média.

Assim como na análise da Árvore de Decisão usual, procede-se com a etapa de *prunning* da árvore. A Figura 13 não revela exatamente o valor do Parâmetro de Complexidade (CP) mas a inspeção do objeto no *Software* R explicita que o ponto da curva descrita que está concomitantemente mais a esquerda e abaixo da linha vertical é 0.0064.

Um classificador com os mesmos parâmetros que antes e esse novo valor para CP apresentou medida F1=0.644 em um tempo para classificação de 0.123 segundos sendo esse superior e mais rápido do que a análise sem PCA.

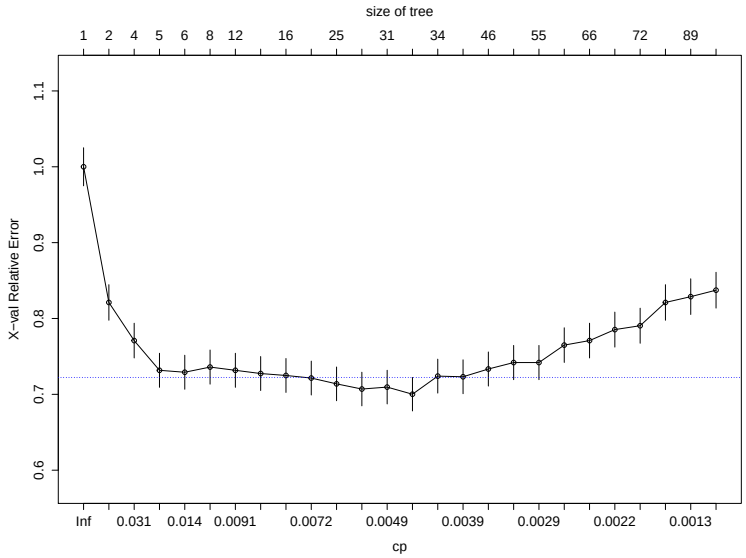

Figura 13 – Relação entre parâmetro de complexidade da árvore com PCA e seu número de nós para ajudar na etapa de *prunning* da mesma visando diminuir o número de parâmetros envolvidos.

6.3 SVM

Para técnica de Máquina de Vetor de Suporte apresentada na Seção 5.3 utilizou-se o pacote *e1071* (DIMITRIADOU et al., 2008) do *software* R. Trata-se do pacote mais tradicional para SVM no R dispondo dos tipos linear e não linear dividindo-se, nesse último, nos *kernels* Polinomial, Radial (ou RBF ou Gaussiano) e Sigmoide.

Os parâmetros da técnica não foram estimados a partir de funções do pacote. Optou-se por variar um *grid* com diferentes possíveis valores para cada um dos parâmetros. Os resultados dos classificadores são apresentados a seguir.

6.3.1 Linear

O SVM linear é o caso mais simples e o que melhor se adequa ao contexto de classificação textual segundo resultados de Joachims (1998). Sendo assim, é natural o seu emprego e para tanto foram estudados inúmeros cenários correspondendo a variações nos valores de três parâmetros: a frequência mínima do termo para que entre na análise, a ponderação do termo adotada (duas características do *Corpus*) e o parâmetro de Custo do SVM. Esse último é considerado primordial no ajuste de um SVM não simplista e que não se superajuste aos dados. Ele corresponde ao valor que penaliza a entrada de observações entre os vetores de suporte, um valor para controlar ar variáveis de folga e, assim, controlar a margem do problema.

Nesse trabalho, os parâmetros de pré processamento variaram conforme a Tabela 3 e o Custo admitiu os valores $10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4, 10^5$ resultando em 132 cenários diferentes.

A medida F1 para avaliar a capacidade de generalização do classificador obtido e seu respectivo tempo de classificação para o Banco de Teste aparecem sob a forma de *boxplots* nas Figuras 14 e 15, respectivamente.

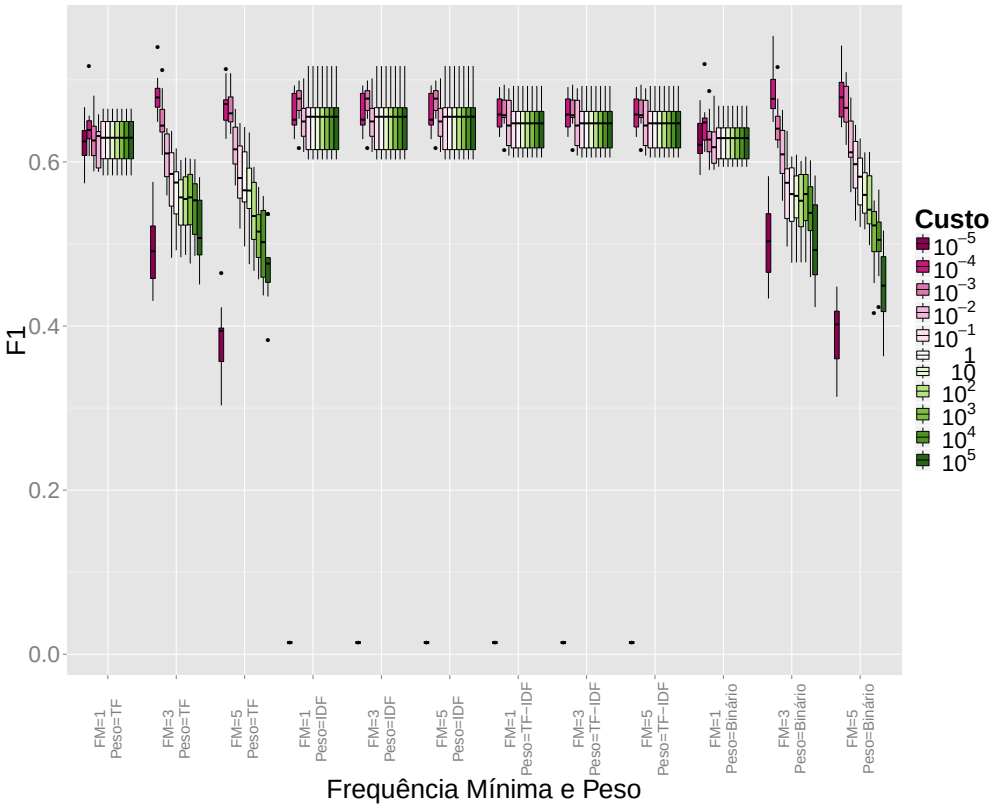

Figura 14 – Medida F1 para generalização por classificador obtido por técnica SVM linear.

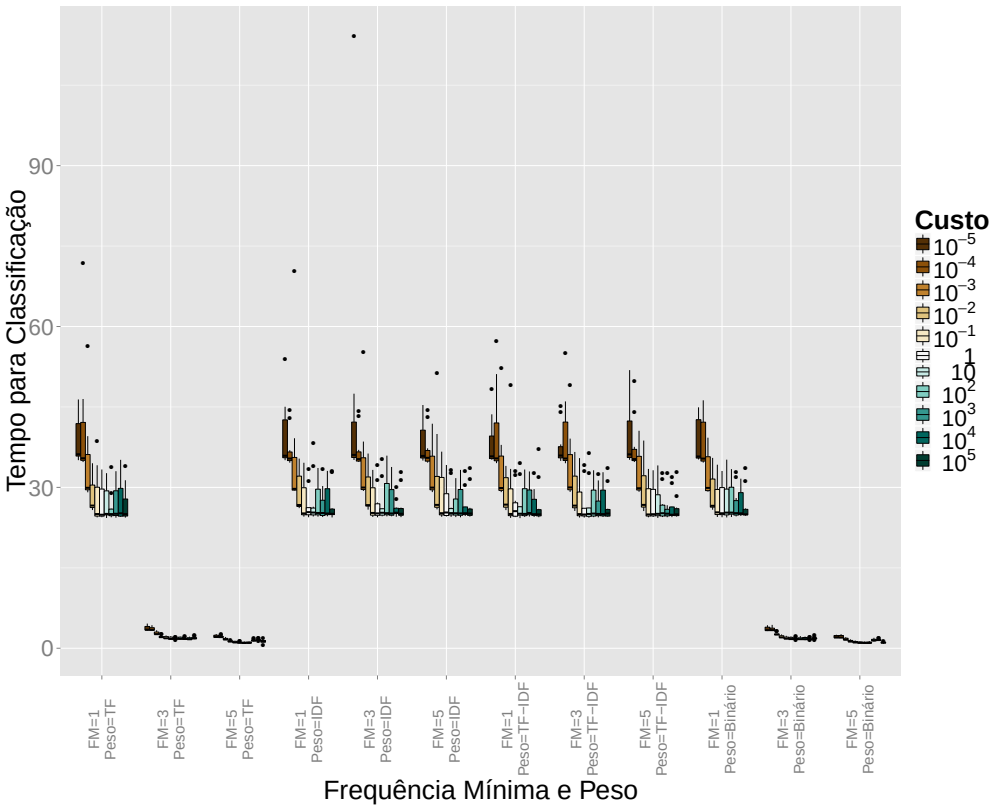

Figura 15 – Tempos necessário para classificação por classificador obtido por técnica SVM linear.

A partir da Figura 14 observa-se um decréscimo do desempenho do classificador (F1) em função dos critérios frequência mínima do termo e ponderação. Nas laterais da Figura 14 essa característica é mais acentuada ao passo que na parte central observa-se um comportamento mais homogêneo da medida por cenário. É possível dizer que os cenários com menores valores para o parâmetro de custo tem desempenho relativamente superior àqueles com o custo mais elevado.

Já no que diz respeito ao tempo necessário para classificação o fato mais marcante é a grande variabilidade de medidas observadas o que compromete a qualidade da visualização dos *boxplots*. Os cenários com diminuição do vocabulário (frequência mínima 3 e 5) e ponderações mais simples (TF e Binário) aparecem como vantajosos segundo esse critério. Nos demais cenários o parâmetro de custo é inversamente proporcional ao tempo para classificação.

A ANOVA na Tabela 4 levaria a concluir pela diferença de médias mas os testes descritos na Tabela 5 não validam o seu uso já que demostram a quebra de tais pressupostos. Segue-se, portanto, com o teste não paramétrico Kruskal-Wallis o qual demarca diferença entre cenários. Todavia a verificação dos pares de cenários que são distintos entre si (teste de comparações múltiplas) é inviável tendo em vista a necessidade de observar as entradas de uma matriz 132x132. Cabe então, proceder como na análise da Árvore de Decisão e inspecionar os critérios de maneira univariada como na Figura 16.

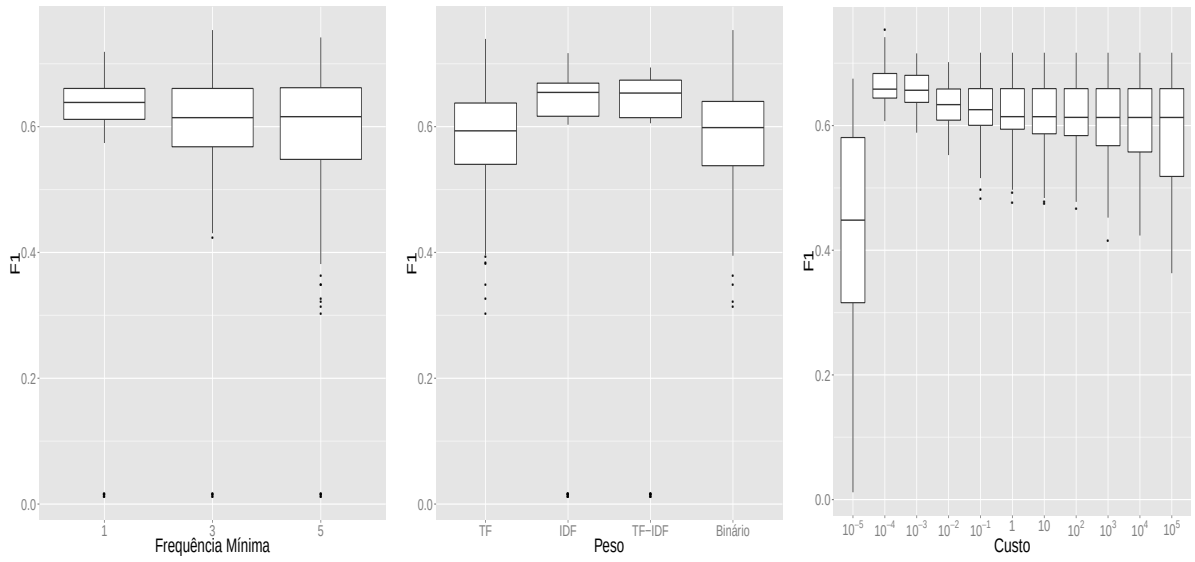

Figura 16 – *Boxplots* para frequência mínima do termo (esquerda), forma de ponderação para os termos (centro) e parâmetro de custo (direita) para classificadores concebido técnica SVM linear.

Na Figura 16, vêem-se caixas muito parecidas sendo os *boxplots* representativos dos pesos os que demarcam diferenças nos níveis mas não permitem concluir qual ponderção seria preferível. A leve superioridade da caixa do custo admitindo 10^{-4} e pela análise das Figuras 14 opta-se pela definição desse parâmetro. Sendo assim, define-se um cenário reduzido para o SVM tomando frequência mínima do termo como 3 e 5, peso TF e binário e custo 10^{-4}.

A Figura 17 demonstra que a capacidade de generalização no que diz respeito à ponderação é semelhante para os dois níveis adotados. Porém, o tempo para classificação quando trabalha-se com o menor vocacbulário possível é bastante inferior ao tempo quando a frequência mínima é 3. Por conseguinte, opta-se pelo classificador com frequência mínima 5. Basta, então selecionar o parâmetro peso. Observando a mediana superior da caixa branca da Figura 17 a esquerda opta-se pela ponderação Binária, a mais simples.

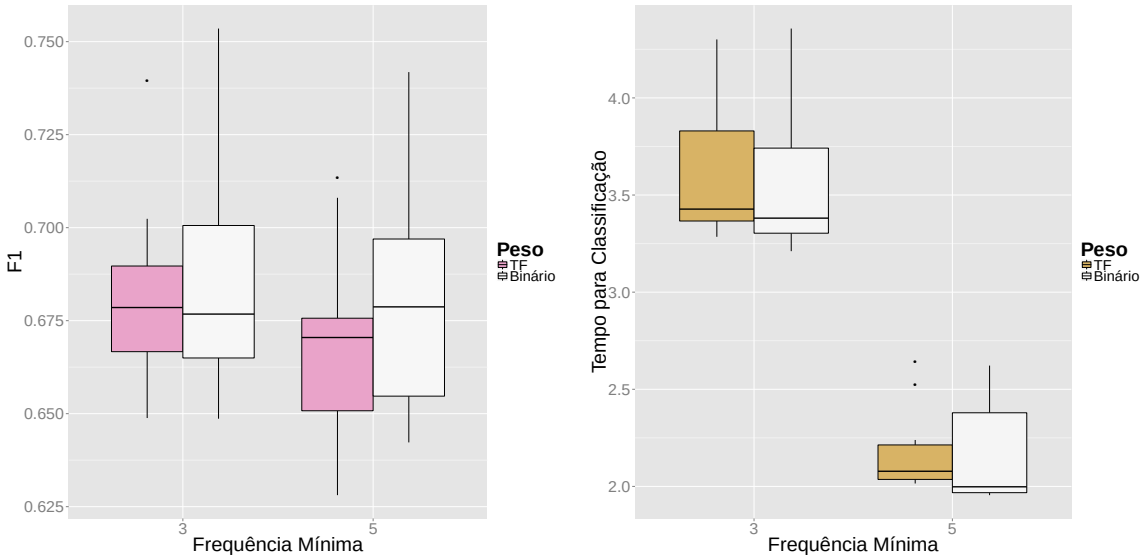

Figura 17 – *Boxplots* para medida F1 de generalização do classificador (esquerda), e tempo para classificação (direita) para classificadores concebidos pela técnica SVM linear com custo 10^{-4}: melhores cenários.

O SVM linear com essas características tem medida F1 média 0.6804898 e tempo médio de classficação 2.1577 segundos.

6.3.2 Linear com Análise de Componentes Principais

Dentre as técnicas abordadas nesse trabalho a que está associada a maior esforço computacional é a SVM. O seu desempenho como classificador é muito favorável e a ideia, bem como no caso de Árvores de Decisão, é de explorar o uso conjunto da técnica com o uso de componentes lineares advindas de uma Análise de Componentes Principais. Essa sub-seção apresenta os resultados dessas análises para PCAs como as já apresentadas.

Nas Figuras 18 e 19 nota-se um comportamento semelhante dos classificadores. Para todos os cenários os classificadores em que o custo assumiu valor 10^{-3} (caixas vermelhas) parecem ser a opção de pior desempenho. Para os pesos TF e Binário nos cenários de vocabulário reduzido observa-se que os classificadores com maiores custos tem desempenho inferior aos demais. Nas caixas da Figura 18 a ponderação $TF - IDF$ apresenta a menor variabilidade que as demais mas tal fato não se repete nos *boxplots* da Figura 19. Em ambas as Figuras os cenários cujo peso é Binário parecem melhores no que diz respeito à medida F1. Contudo, para PCA 80% o parâmetro de custo mais adequado assume valores 10^{-1} e 10^{-2} ao passo que para o PCA 90% parece ser 10^{-2}.

Já nas Figuras 20 e 21 observa-se nitidamente que o custo é inversamente proporcional ao tempo necessário para classificação. Basta observar que para mesmos valores de peso e frequência mínima (conjuntos de caixas próximas) os *boxplots* estão em diferentes níveis no eixo vertical. Também é interessante demarcar que para o custo 10^{-3} vê-se que o tempo para classificação se diferencia, não se agrupando com outros. Assim como em análises anteriores, vê-se que os menores tempos são para classificadores com peso TF ou Binário e frequência mínima 3 ou 5. Além do mais, os cenários para PCA 90% (Figura 21) tem maior variabilidade e apresentam classificadores mais lentos que os cenários para PCA 80%.

Mais uma vez, há impossibilidade no uso das ANOVAs descritas na Tabela 4 pelas quebras dos seus pressupostos explicitadas na Tabela 5. Apesar do teste não paramétrico de Kruskal-Wallis ser indicativo de diferenças nas médias por cenário é inviável explorar diferenças de pares 132x132 de cenários.

Usa-se o tempo para classificação para reduzir o número de cenários em análise de sorte que opta-se pelo estudo detalhado daqueles em que há diminuição do vocabulário (frequência mínima diferente de 1) e as ponderações mais simples: TF e Binário. Mais *boxplots* para esses cenários específicos aparecem nas Figuras 22 e 23.

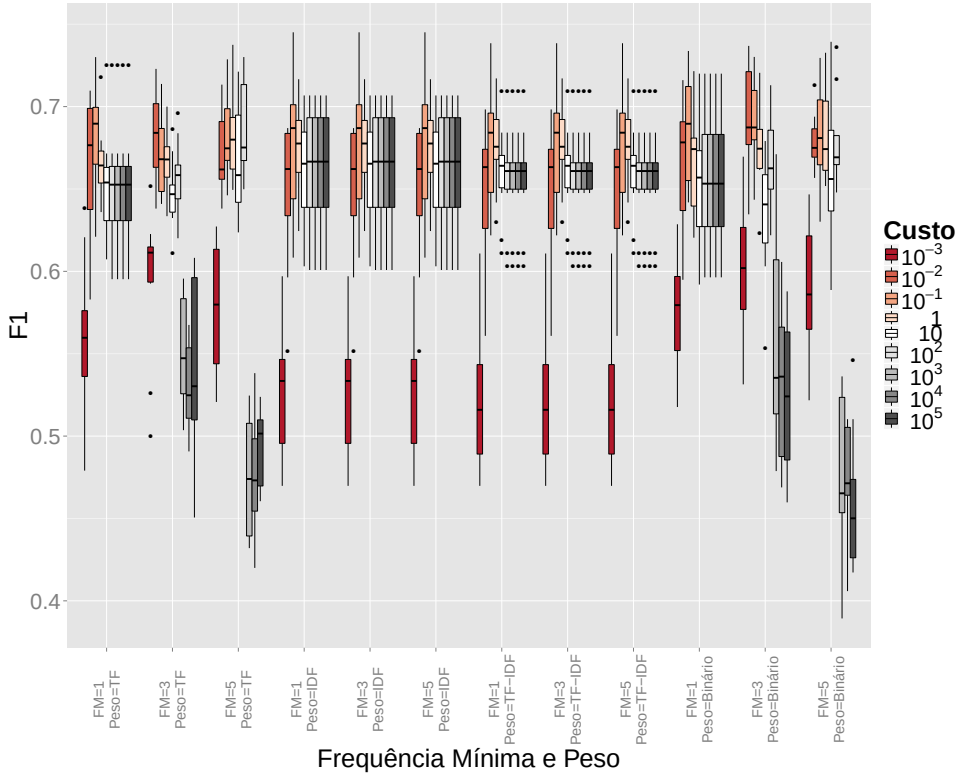

Figura 18 – *Boxplots* por cenários da técnica SVM linear para medida F1 de capacidade de generalização com PCA 80%.

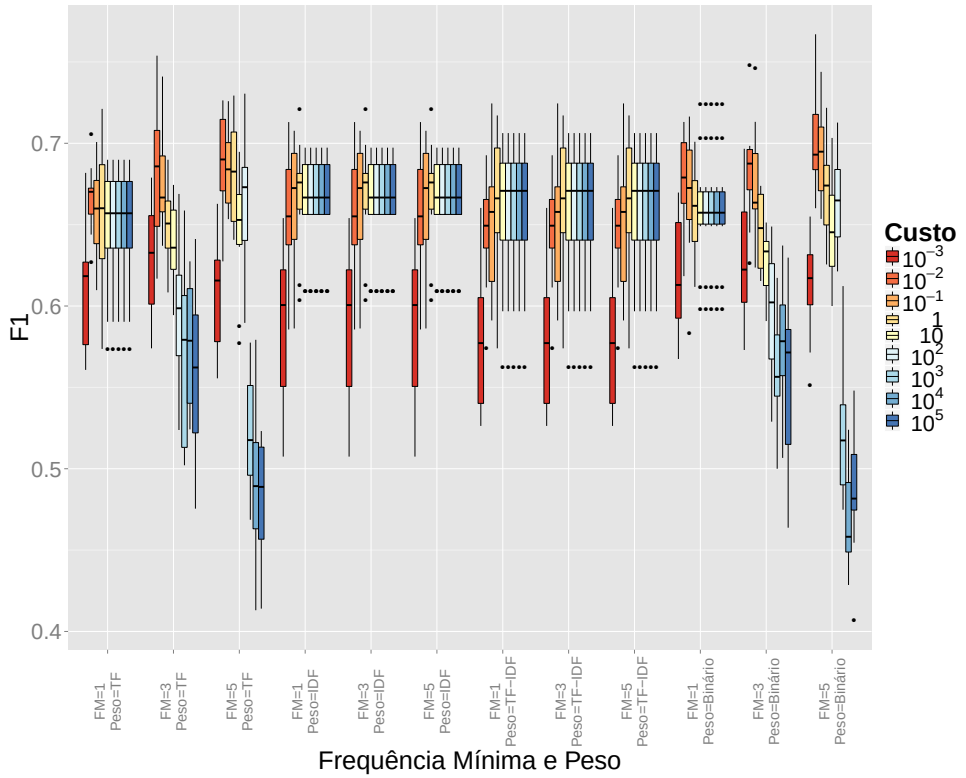

Figura 19 – *Boxplots* por cenários da técnica SVM linear para medida F1 de capacidade de generalização com PCA 90%.

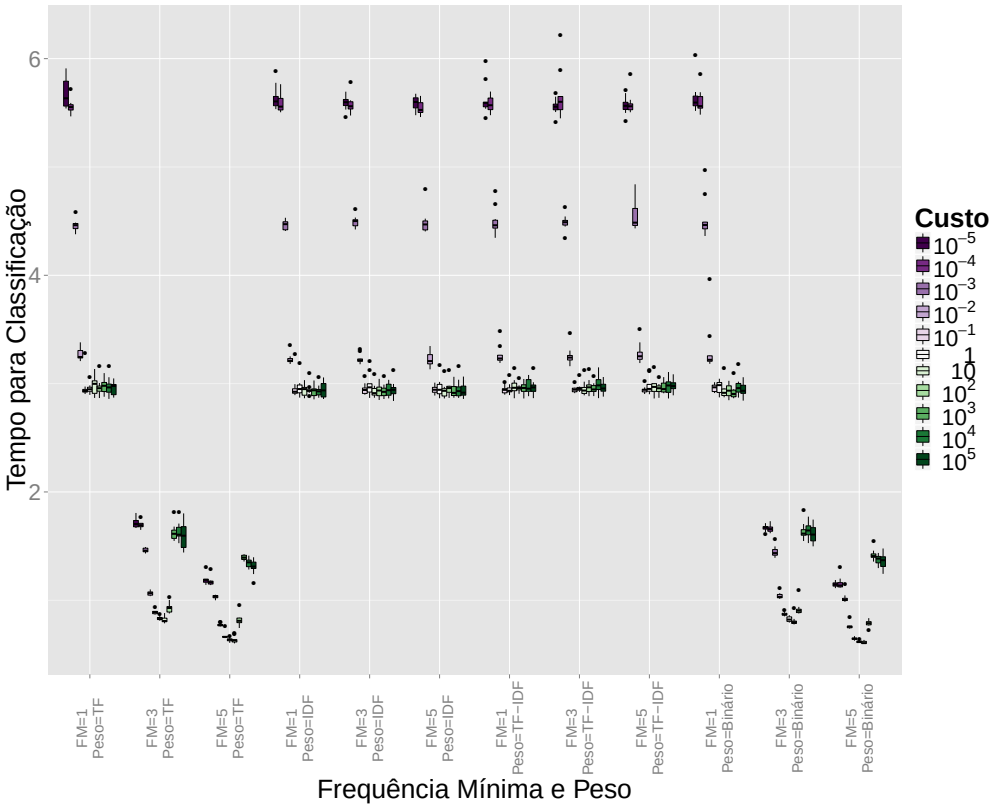

Figura 20 – *Boxplots* por cenários da técnica SVM linear para tempo para classificação com PCA 80%.

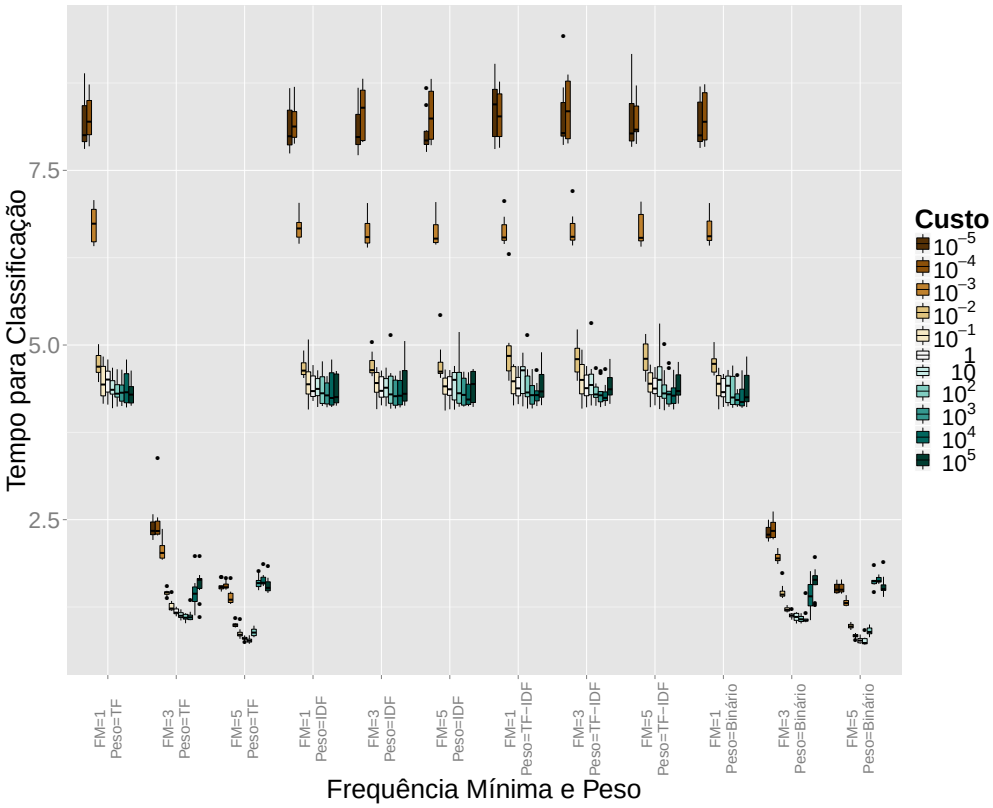

Figura 21 – *Boxplots* por cenários da técnica SVM linear para tempo para classificação com PCA 90%.

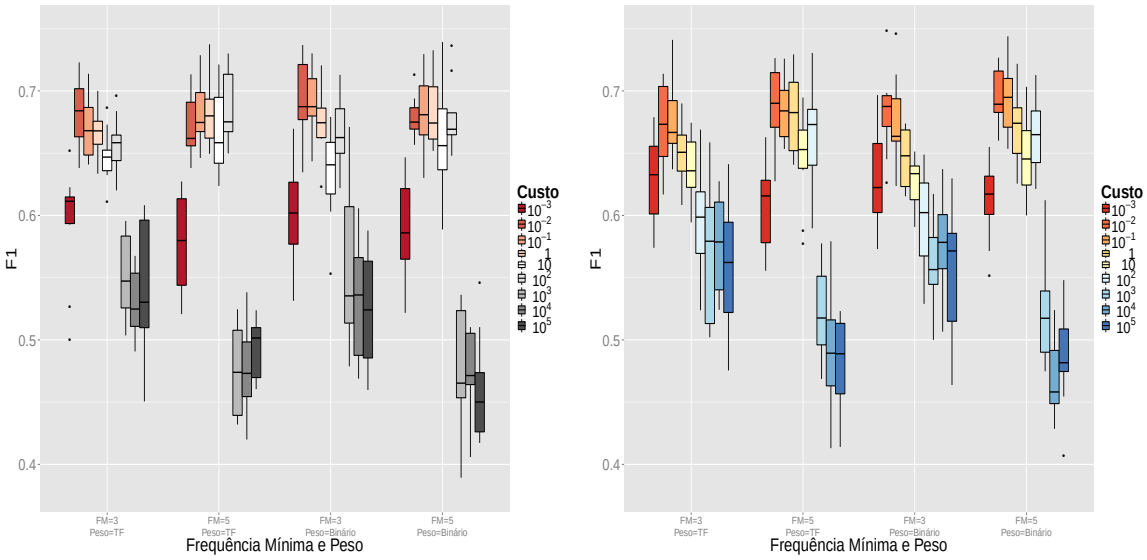

Figura 22 – *Boxplots* por cenários da técnica SVM linear com frequência mínima 3 ou 5 e peso TF ou Binário para medida F1 de capacidade de generalização com PCA 80% (esquerda) e 90% (direita).

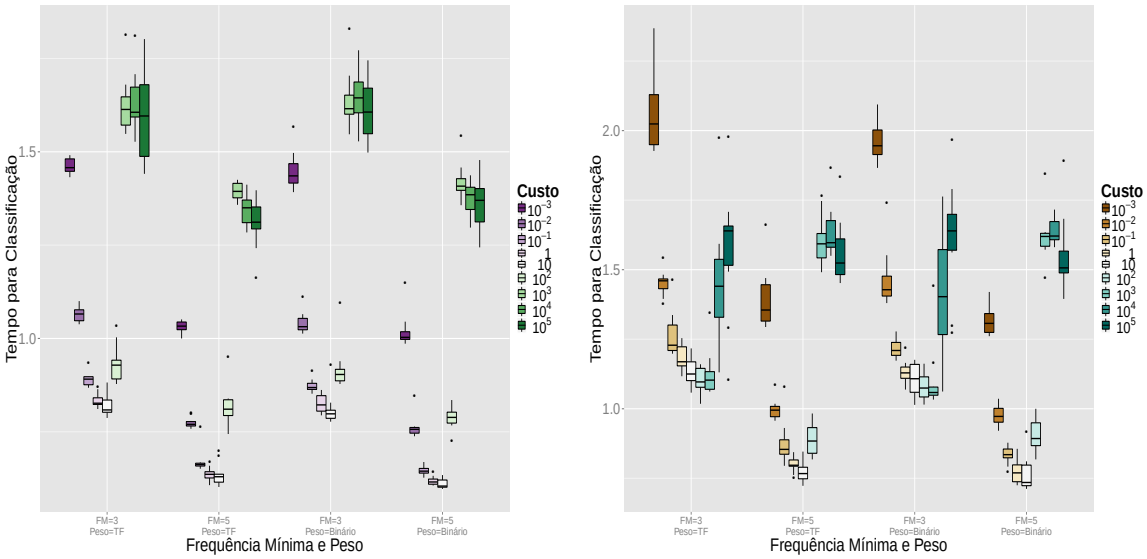

Figura 23 – *Boxplots* por cenários da técnica SVM linear com frequência mínima 3 ou 5 e peso TF ou Binário para tempo para classificação com PCA 80% (esquerda) e 90% (direita).

A Figura 22 explicita que os valores $10^{-3}, 10^{3}, 10^{4}, 10^{5}$ para o parâmetro de custo produzem classificadores com capacidade de generalização inferior aos demais valores. Ao se utilizar PCA de 90% o decréscimo da eficiência do classificador em função do custo é mais suave ao passo que no uso de PCA 80% a queda de eficiência é abrupta entre o custo 10^{2} e 10^{3}. A observação das imagens à esquerda e à direita não possibilita concluir pela superioridade dos classificadores com qual percentual de variância total explicada pelo PCA. Na maioria das combinações de peso e frequência mínima as caixas representativas de custo 10^{-2} superam, em eficiência de classificação as caixas com custo 10^{-1}, mas é importante reforçar que essa diferença é pequena. A leve superioridade das caixas à direita também é muito sutil mas será suficiente para definir a ponderção Binária como a melhor.

Já no estudo das imagens que constam na Figura 23 é claro que o PCA 80% permanece como classificador cuja classificação é mais rápida. A diferença de variabilidade entre as imagens à esquerda e à direita também é muito forte.

Gráficos univariados foram suprimidos dessa análise por serem pouco informativos na definição de qual frequência mínima por termo ou qual o percentual de explicação exigida pela PCA. Ao contrário das metodologias anteriores optou-se por fazer uma análise secundária com um banco de dados com mais registros para o desempate dos parâmetros. Como dito no Capítulo 3 dispoẽ-se de um banco de treino com 10000 postagens e os demais 16000 registros são empregados como banco de teste para validação do classificador encontrado. Em se tratando de uma análise para desempate não se procedeu com a validação cruzada de tamanho 10. Os resultados dessas análises encontram-se a seguir:

Peso	Custo	% PCA	Freq. Mín.	Tempo Classif.	Precisão	Revocação	F1
Binário	0.01	80	3	125.894	0.8012	0.7178	0.7572
Binário	0.01	80	5	93.736	0.8273	0.7206	0.7703

Tabela 7 – Resultados de tempo de classificação e medidas da qualidade do ajuste para cenários SVM linear com PCA 80% cujo peso é binário e o custo é 0.01.

A partir dos resultados da Tabela decide-se pela frequência mínima por termo igual a 5 a qual concilia a melhor capacidade de genralização e o menor tempo de classificação.

6.3.3 Não Linear

Como dito no Capítulo 5, as técnicas SVM são não lineares apresentam um grau de complexidade e tempo computacional muito maior que as lineares. Sabendo disso e lembrando que Joachims (1998) apresenta os SVM lineares como os mais adequados para classificação textual os SVM com *kernel* foram avaliados de maneira mais simplificada do que nas análises que sucedem. Optou-se por não fazer uso de validação cruzada de modo que tomou-se uma única amostra de tamanho 500 dentre os 5000 dados iniciais. Esse 1/10 dos dados constituiu o banco de treino e os demais 9/10 constituiu o banco de teste. Nas análises que seguem não há, portanto, variabilidade de resultados por cenário. A medida F1 e o tempo para classificação por cenário aparecem nas Tabelas 13, 15, 14 e 16 na Sub-Seção 9.2.1, em Anexo.

6.3.3.1 *Kernel* Polinomial

Nesse tipo de SVM não linear especifica-se, além dos parâmetros do *Corpus* (frequência mínima e peso) e do custo controlador do tamanho da margem, o grau do poliônomio desejado. Dessa forma avaliou-se os parâmetros de pré processamento variando conforme a Tabela 3, o Custo admitindo os valores $10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4, 10^5$ e o grau do polinômio assumindo as quantidades 2, 3 e 5. A variação conjunta de todos esses parâmetros resultou em 396 cenários diferentes.

Os resultados da Tabela 13 não permitem definição clara dos cenários mais adequados especialmente dada suas grandes dimensões. Olha-se cada parâmetro separadamente para exploração da tabela. O parâmetro de custo é ligeiramente superior ao nível 10^{-4} apesar de se tratar de uma diferença diminuta. Os pesos IDF e $TF - IDF$ tem desempenho pouco maior que as demais ponderações e para esses dois níveis os níveis da variável frequência mínima não demarcam diferenças na medida F1. As médias de F1 pelo parâmetro grau nos níveis 2, 3 e 5 são 0.611, 0.598 e 0.565 com respectivos tempos de classificação 24.983, 25.667 e 26.236 (em segundos) de modo que o grau 2 é considerado o melhor.

Analisando a Tabela 14 nota-se que o aumento do custo está associado ao decréscimo do tempo para classificação e as ponderações TF e Binário são as mais rápidas. Todavia, quando a frequência mínima do termo é 1 para os pesos TF e Binário o tempo para categorização se assemelha aos tempos dos pesos IDF e $TF - IDF$ assumindo valores muito altos. Dessa forma, apresenta-se uma leve contradição com a análise da medida F1. Não há uma ponderação que atenda aos critérios de bom classificador em um tempo razoável.

Uma inspeção minuciosa da Tabela 13 revela que o maior valor de F1 (0.636) corresponde à entrada cuja frequência mínima é 3, o peso é Binário, o custo é 10^{-4} e o grau é 2 cuja classificação demorou 3.264 segundos. Esse é o cenário considerado melhor segundo a técnica SVM polinomial.

6.3.3.2 *Kernel* Radial

A especificação de cenários para esse *kernel* também é delicada. Nesse caso, além da frequência mínima e ponderação do termo especificou-se o parâmetro custo admitindo os valores $1, 10, 10^2, 10^3, 10^4, 10^5$ e o parâmetro Gamma assumindo as quantidades $10^{-4}, 10^{-3}, 10^{-2}$.

A característica mais marcante dos classificadores dessa técnica é a sua baixa capacidade de generalização. Na Tabela 15 vêem-se inúmeras entradas não preenchidas (-) as quais designam cenários em que o classificador não pode ser avaliado segundo a medida de precisão ou de *recall* resultando no não cálculo de F1. Fora isso, vê-se as entradas da Tabela com valores muito pequenos que só confirmam o desempenho ruim do *kernel*.

É interessante notar que, mais uma vez, o custo é inversamente proporcional à medida F1 e que o peso Binário aparece como superior em relação aos demais. Observa-se a mesma discussão para a

frequência mínima dentro dos pesos IDF e $TF - IDF$ bem como a superioridade da frequência mínima 5 quando a ponderação é TF ou Binário. A visualização atenta da segunda coluna mais a direita explicita que o valor de Gamma como 10^{-4} está associado aos cenários com melhores F1.

Já no que tange ao tempo para classificação, a Tabela 16 tem semelhança com as análises do *kernel* Polinomial: o aumento do parâmetro de Custo associa-se com o decréscimo tanto da medida F1 quanto do tempo para classificação, os níveis Binário e 10^{-4} dos parâmetro peso e Gamma são os mais rápidos computacionalmente.

O maior valor observado na Tabela 15 para qualidade do classificador foi 0.494 com as características: custo = 10^3, Gamma = 10^{-4}, Peso = Binário e frequência mínima do termo = 5. Esse classificador demorou 1.3 segundos na categorização do banco de teste.

6.3.3.3 Resumo

Os melhores classificadores - segundo cenários que variavam aspectos do pré processamento do texto e parâmetros da própria técnica - são explicitados nas Tabelas 8, 9 e 10 resultando num panorama geral do trabalho.

Técnica	Peso	Freq.Mín.	F1	Tempo (em seg.)
Naive Bayes	IDF	3	0.6568	9.2138

Tabela 8 – Resultados dos parâmetros, medida F1 e tempo de classificação do classificador Naive Bayes.

Técnica	Peso	Freq.Mín.	Núm.Mín. Obs.Nó	Crit.Quebra Nós	Parâm. Complex.	F1	Tempo (em seg.)
Árvore de Decisão	Binário	5	20	Gini	0.0046	0.591	0.147
Árvore de Decisão com PCA	TF	5	20	Gini	0.0064	0.644	0.123

Tabela 9 – Resultados dos parâmetros, medida F1 e tempo de classificação dos classificadores Ávore de Decisão.

Técnica	Peso	Freq.Mín.	Custo	Grau	Gamma	F1	Tempo (em seg.)
SVM Linear	Binário	5	10^{-4}	-	-	0.684	2.157
SVM Linear com PCA	Binário	5	10^{-2}	-	-	0.678	0.761
SVM Não Linear com *kernel* Polinomial	Binário	3	10^{-4}	2	-	0.636	3.264
SVM Não Linear com *kernel* Radial	Binário	5	10^3	-	10^{-4}	0.494	1.300

Tabela 10 – Resultados dos parâmetros,medida F1 e tempo de classificação dos classificadores SVM.

Dentre os classificadores é gritante a inferioridade do SVM com *kernel* Radial. Pode-se dizer que os classificadores tem capacidade de generalização de cerca de 60% com destaque para o SVM Linear e SVM Linear com PCA de 80% cujas medidas F1 resultaram 0.68 e 0.67, respectivamente. Todavia, a superioridade do tempo de classificação do segundo sob o primeiro leva a concluir que o SVM Linear com PCA de 80% é o melhor classificador dentre os estudados e, portanto, o que será empregado na classificação das postagens coletadas ao longo dos meses em estudo. Esse classificador, teve medida F1 de 0.678 e um tempo de classificação de 0.761 segundos no banco de 500 postagens.

É relevante ressaltar que os parâmetros de pré processamento para quase todos os classificadores coincidiram: termos com frequências mínimas 3 ou 5 e a ponderação dada a eles, em 5 dos 7 resultados descritos, foi a Binária.

7 Visualização dos Dados

Uma vez coletados e classificados os *tweets* dispõe-se de um conjunto de postagens, na região em estudo, com conteúdo adequado. É pertinente, então, o uso de ferramentas que possibilitem a informação e visualização de uma grande massa de dados com enorme possibilidades de exploração. Lembra-se que, na Seção 1.3, comentou-se sobre o interesse na disponibilização das análises sob a forma de produtos claros, concisos e de apelo visual.

Todos esses objetivos e características implicam no uso de uma ferramenta bastante versátil. Intui-se a visualização desses dados sobre a forma de gráficos, tabelas e mapas em um formato que atenda a curiosidades e especifidades de consultas por usuários. Sendo assim, utilizou-se conjuntamente os pacotes *googleVis* (GESMANN; CASTILLO, 2011) e *shiny* (RStudio; Inc., 2014) do *Software* estatístico R, versão 3.1.1 (R Core Team, 2014). O primeiro deles tem funcionalidades gráficas interativas e a possibilidade do uso dos mapas do Google. Já o segundo, visa a criação de produtos advindos do R para divulgação na Web sem requisitar habilidades em programação na mesma. O emprego de um banco de dados extenso também exigiu pacotes para manipulação de dados. Nesse casos, cita-se o pacote *plyr* (WICKHAM, 2011), o pacote *dplyr* (WICKHAM; FRANCOIS, 2014) e o pacote *reshape* (Wickham; Hadley, 2007).

Algumas formas de visualização pretendidas serão listadas a seguir sob a forma de um único produto: o Aplicativo *TweetsCrimeSP*.

Na Figura 24 observa-se na parte superior uma sucessão de abas. A aba que está em análise na Figura resume o total de coletas (ou seja, para todos os termos pesquisados) por município através de uma taxa de coltas por 100.000 habitantes. No mapa cada polígono é pintado com uma cor que traduz essa taxa de sorte que os municípios considerados mais violentos aparecem em tons mais escuros. Repara-se que na parte mais ao Leste e ao Sul um dos polígonos recebe contorno branco mais acentuado. Trata-se de um município selecionado com o *mouse* pelo usuário. Ainda mais, na caixa superior a direita lê-se o nome do município e suas respectivas taxa de crime por 100.00 habitantes e população.

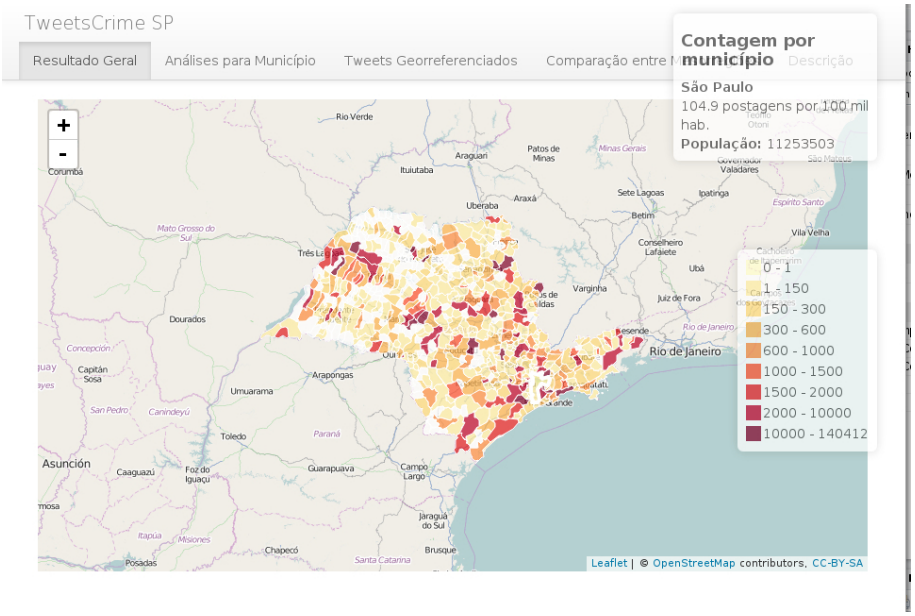

Figura 24 – Visualização do aplicativo *shiny* *TweetsCrimeSP* - Aba: Mapa Total de Coletas por Município.

Já nas Figuras 25, 26, 27, 28 e 29 vêem-se visualizações para características das coletas de um dado município. À esquerda de cada figura nota-se uma caixa que permite a definição do município de interesse. Importante ressaltar que em todos os gráficos tem-se o correspondente do estado. A ideia é possibilitar comparação entre o município e o estado.

Para a cidade escolhida de São Carlos observa-se as séries agrupadas por mês ou diária do total de coletas que possibilitariam análises pelo tempo observado como tendências ou um ponto demarcando uma mudança abrupta no comportamento da variável.

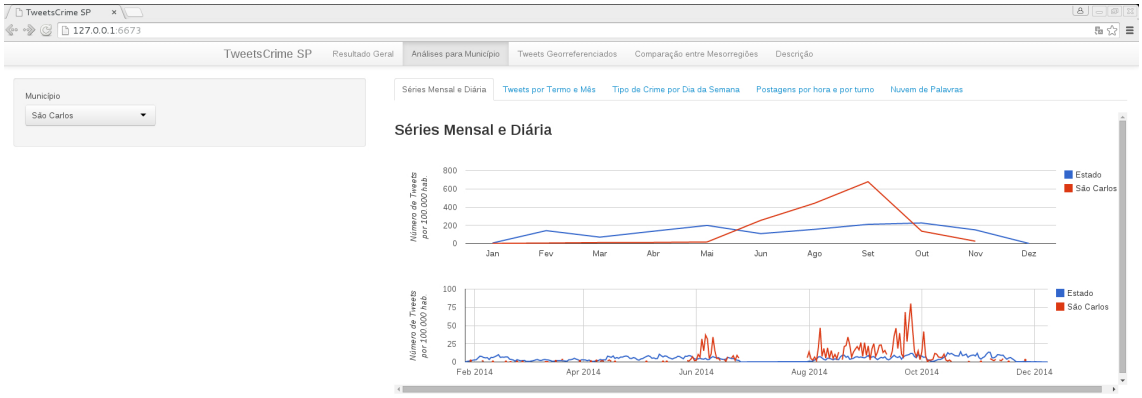

Figura 25 – Visualização do aplicativo *shiny TweetsCrimeSP* - Aba: Análise por Município, séries mensal e diária.

Em se querendo analisar como foi o comportamento dos termos mês a mês pode-se inspecionar a Figura 26. Uma coluna empilhada para um dado mês é particionada em retângulos de cores diferentes, cada um denotando a frequência relativa de postagens por termo pesquisado. Dessa Figura pode-se ver que o termo "roubo" na cor azul é muito frequente durante todos os meses e também pode-se verificar se os termos mais postados em um município equivalem às expressões mais comuns no estado todo.

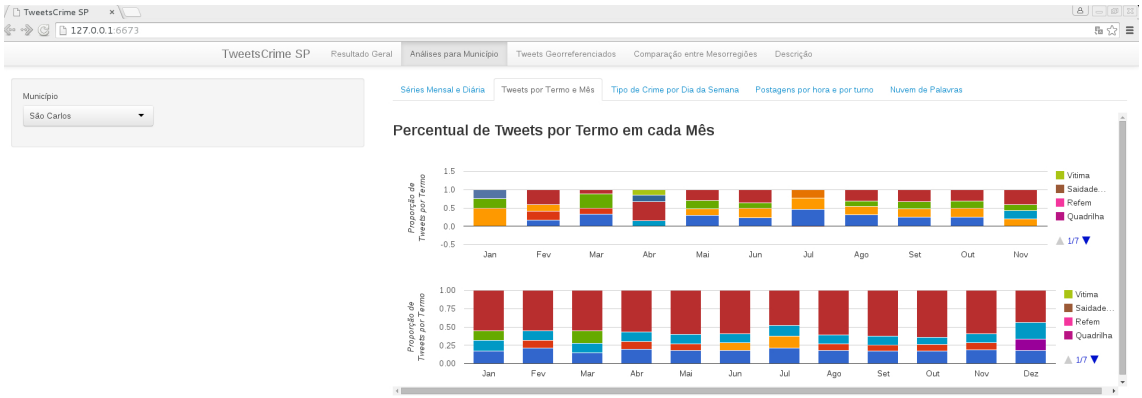

Figura 26 – Visualização do aplicativo *shiny TweetsCrimeSP* - Aba: Análise por Município, colunas empilhadas comparando distribuição dos termos coletados por mês de coleta.

Outra visualização da taxa de criminalidade pelo tempo é possível na aba explicitada na Figura 27. As colunas verticais por tipo de semana representam o total de postagens por tipo de crime, um agrupamento dos 25 termos em 4 categorias, por 100.00 habitantes. É natural que a coluna de crimes contra o patrimônio seja superior às demais pois há mais termos que o denotam.

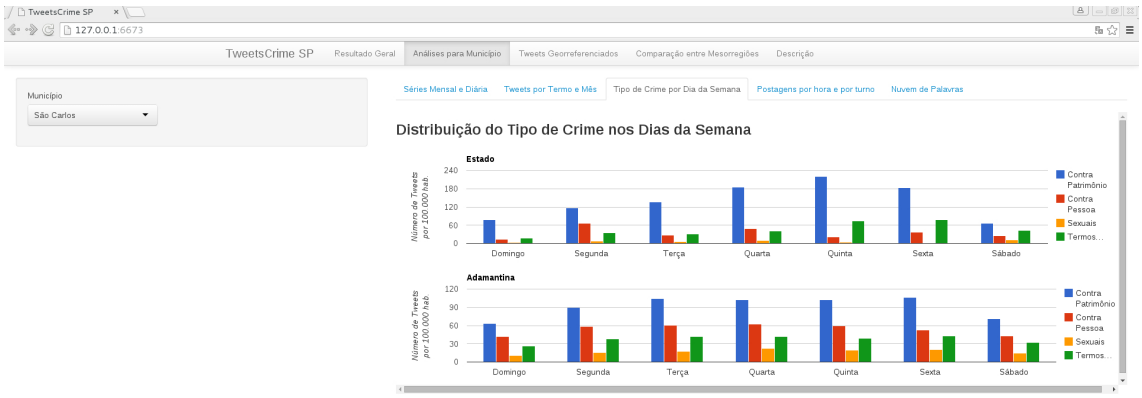

Figura 27 – Visualização do aplicativo *shiny TweetsCrimeSP* - Aba: Análise por Município, distribuição do tipo de crime por dia da semana.

Pensando-se em um refinamento ainda mais detalhado no que tange ao tempo pode-se procurar por uma visualização por hora e por turno como a que consta na Figura 28. As colunas com o as taxas de postagens por 100.00 habitantes por hora explicitam, por exemplo, que entre 11:00 e 13:00 a cidade de São Carlos tem bem mais postagens que o estado de São Paulo mas de uma forma geral há semelhanças entre as duas colunas. O agrupamento das taxas de postagens em 4 categorias similares a turnos (manhã, tarde, noite e madrugada) com os gráficos de setores pode ser razoável em algumas situações justificando a presença de tais gráficos.

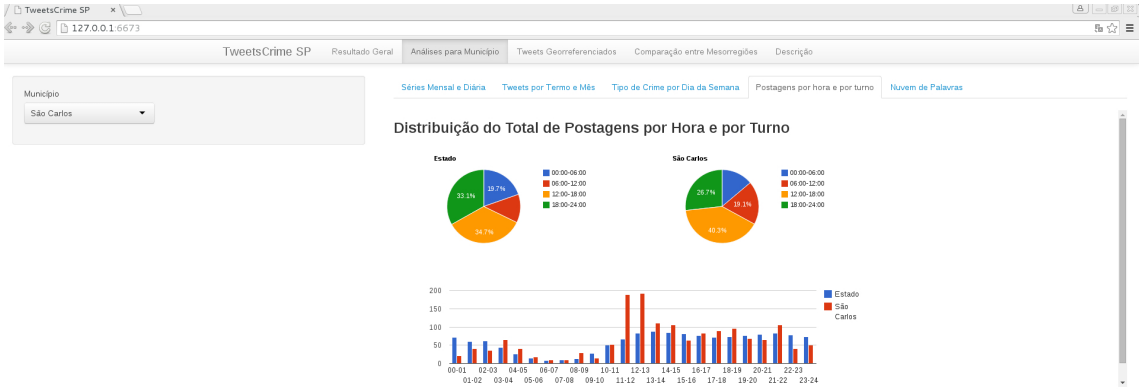

Figura 28 – Visualização do aplicativo *shiny TweetsCrimeSP* - Aba: Análise por Município, distribuição do total de postagens por hora e por turno.

E lembrando que a principal fonte de informação das postagens coletadas é o próprio texto postado pensou-se na visualização de uma nuvem de palavras. Assim, os 25 termos pesquisados aparecem com fonte proporcional à sua frequência relativa salientando que em São Carlos muitos dos relatos de crime se deram com as expressões “furtado” e “tráfico”.

Como já mencionado, postagens com georreferenciamento são naturalmente mais raras. Essas postagens interessantes ocorreram em apenas 5% dos dados e uma aba para elas também é válida no aplicativo ambicionado. Na Figura 30 há uma possibilidade de exploração dessa característica com a definição, pelo usuário, do município, termo e mês em investigação. Com a interface Google, a direita aparece um mapa demarcando com balões vermelhos os locais das postagens. Passando o *mouse* sobre os balões é possível ler o respectivo *tweet*.

Figura 29 – Visualização do aplicativo *shiny TweetsCrimeSP* - Aba: Análise por Município, nuvem de palavras dos termos postados.

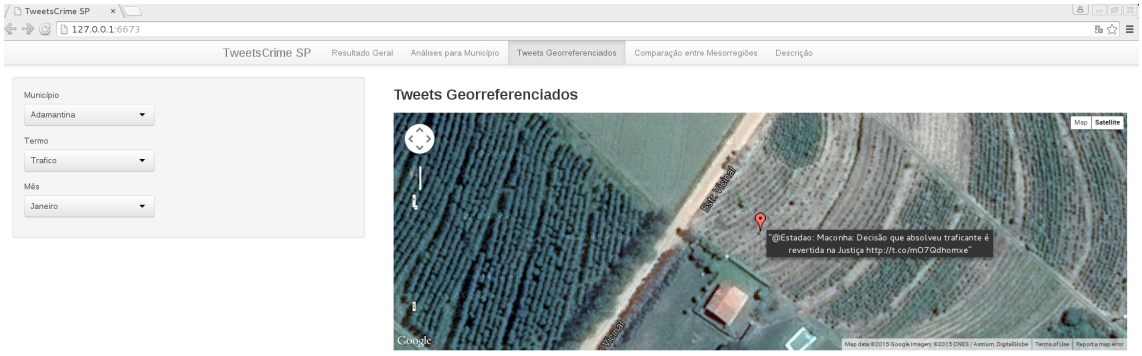

Figura 30 – Visualização do aplicativo *shiny TweetsCrimeSP* - Aba: Postagens Georreferenciadas.

Pensando na curiosidade comum de querer comparar municípios ou regiões concebeu-se a visualização de mais uma aba. Nesse caso, pensou-se na comparação entre Mesorregiões do Estado. Como são apenas 15 mesorregiões, essa divisão geográfica é mais fácil de analisar no gráfico *Motion Chart* que é apresentado na tela da Figura 31. Esse gráfico permite a comparação de municípios segundo mais de um termo de crime de uma forma cronológica. O usuário define quais variáveis vão nos eixos x e y, e quais variáveis as bolinhas, seus tamanhos e suas cores denotam. Há ainda um tipo de marcador que pode acompanhar as bolinhas desejadas. As bolinhas descrevem um percurso à medida que o tempo avança na barra horizontal inferior se assemelhando a um vídeo do gráfico.

O aplicativo descrito permite diferentes formas de visualização com boa estética, atendendo a especifidades de consultas por usuários e interatividade. Logo, trata-se de uma ferramenta bastate adequada às necessidades do trabalho.

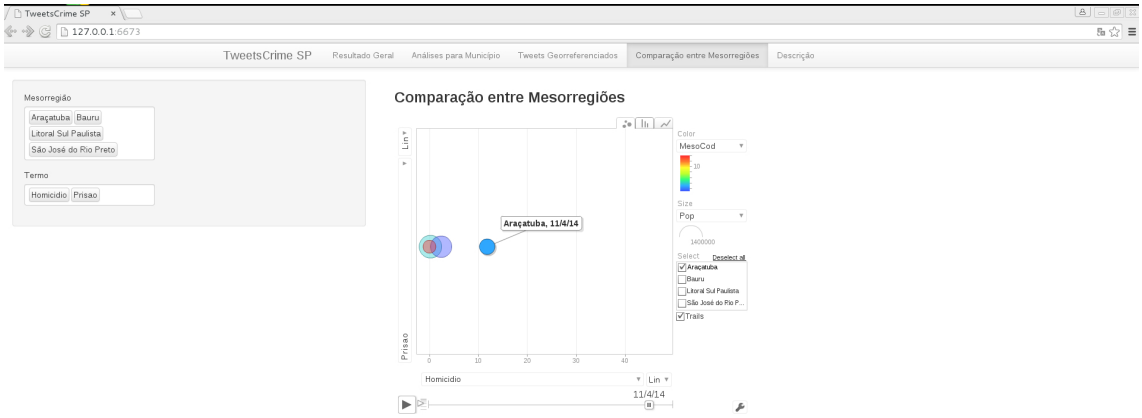

Figura 31 – Visualização do aplicativo *shiny TweetsCrimeSP* - Aba: Análise por Município, nuvem de palavras dos termos postados.

8 Conclusão e Trabalhos Futuros

A tarefa de classificação textual com suas peculiaridades de alta dimensão do espaço de características e vetores esparsos mostra-se tarefa ainda mais difícil em se tratando de postagens na rede social Twitter. Pode-se citar como características agravantes a alta dimensão do espaço de atributos, grande volume de textos e o dinamismo de uma linguagem informal, sujeita a características econômicas, sociais e regionais. Um outro problema ainda é o fato das mensagens serem limitadas a 140 caracteres.

Tais postagens, devido à natureza subjetiva da linguagem, não necessariamente denotam a ocorrência de um crime. Sendo assim, é razoável o uso de técnicas que avaliem a factibilidade do texto com o contexto de criminalidade. Para tanto, utilizou-se três diferentes técnicas de Aprendizado de Máquina Supervisionado: Naive-Bayes, Árvore de Decisão e SVM linear e não linear.

Importante lembrar que o uso dessas metodologias exigiu a leitura e classificação manual de uma coleção de textos (26.503 *tweets*) tarefa que demandou esforço e tempo. Dada as características de um problema de categorização textual, dificuldades do ponto de vista computacional são comuns. Na tentativa de contornar a impossibilidade do uso das técnicas pretendidas ou o elevado tempo necessário, fez-se uso da análise conjunta das técnicas com a técnica multivariada Análise de Componentes Principais.

Uma extensa análise e discussão objetivou conciliar duas caracterísiticas interessantes para o classificador: desempenho e factibilidade do tempo necessário para a classificação de novos registros. Por fim, 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 F1 de 0.678 e um tempo de classificação de 0.761 segundos no banco de 500 postagens.

De posse do melhor classificador para o contexto em estudo procede-se com a classificação automatizada de relatos de crime sob a forma de *tweets* coletados durante o ano de 2014. Mais uma vez, demonstra-se o esforço necessário em tal tarefa pois as coletas de 201 dias de tal ano implicaram no arquivament0 de mais de 700.000 arquivos de coletas e o pré-processamento e classificação de centenas de milhares de *tweets*. E lembra-se, ainda, que todo esse esforço não é sinônimo de contagens expressivas de relatos de crime pois o uso metaforizado das expressões utilizadas é mais comum que o uso real.

A melhor visualização para tantas formas diferentes de análise e interpretação dos dados demandou a procura pelas ferramentas adequadas. Optou-se por uma forma de visualização que concilia agilidade na manipulação do Banco e interatividade com o usuário.

Como parte dos trabalhos futuros cita-se a inclusão automática do número de registros que compõem o banco de treinamento. Essa inclusão dispensa a necessidade de um classificador externo, economizando recursos e tempo e deve ser capaz de diferenciar jargões e expressões momentâneas.

As limitações computacionais exigem a definição de uma metodologia própria que, muito provavelmente, contemple características específicas da Rede Social Twitter ou de textos tão curtos. Deve-se investigar a possibilidade da criação de metodologias e métricas paralelizáveis quem melhorem o tempo computacional sem a perda de acurácia do classificador.

Há ainda a possibilidade do uso de classificadores do tipo *Lazy Associative Classifier* de Veloso, Meira e Zaki (2006) que se caracterizam pelo foco local identificando e utilizando apenas a porção útil da base de treino para a classificação de dados da base de teste.

Essas etapas e ambições qualificam e complementam uma visualização *online* dinâmica que estende a apresentada no Capítulo 7.

9 Anexo

9.1 Coletas de *Tweets*

Agredir	Clonaram	Homicidios	Roubadas
Agrediram	Clonou	Ladra	Saida bancaria
Agrediu	Estelionatario	Ladras *	Saida de agencia
Agressao	Estelionatarios	Ladrao	Saida de banco *
Agressor	Estelionato	Ladroes	Saidinha bancaria
Agressores *	Estuprador	Latrocinio	Saidinha de agencia
Arrastao	Estupradores	Mata	Sequestra
Arrombamento	Estupro	Mataram	Sequestrador
Assalta	Estupros	Matou	Sequestradores *
Assaltada	Estuprada	Morreu	Sequestraram
Assaltadas	Estupradas	Morreram	Sequestro
Assaltado	Estupraram	Morto	Sequestros
Assaltados	Estuprou	Mortos	Sequestrou
Assaltante	Fraudaram *	Morta	Sequestrado
Assaltantes	Fraude	Mortas	Sequestrados
Assaltaram	Fraudes	Pedofilia	Sequestrada
Assalto	Fraudou *	Pedofilo	Sequestradas
Assaltos	Furta	Pedofilos	Tarado
Assaltou	Furtaram	Pornografia	Tentativa de assalto
Assassina	Furto	Presa	Tentativa de furto *
Assassinas	Furtos	Preso	Tentativa de homicidio *
Assassinada	Furtou	Quadrilha	Tentativa de roubo *
Assassinadas	Furtada	Quadrilhas	Tentativa de sequestro *
Assassinado	Furtadas	Refem	Traficante
Assassinados	Furtado	Refens	Traficantes
Assassinaram	Furtados	Rouba	Traficaram *
Assassinato	Gangue	Roubaram	Trafico
Assassinatos	Gangues	Roubo	Traficos
Assassino	Golpe	Roubos	Traficou *
Assassinos	Golpes	Roubou	Violencia domestica
Assassinou	Golpista	Roubado	Vitima
Atentado violento ao pudor	Golpistas	Roubados	Vitimas
Clonagem	Homicidio	Roubada	

Tabela 11 – Termos que serviram como palavras-chave para coletas no Twitter: azul: crimes contra a pessoa, verde: crimes contra o patrimônio, vermelho: crimes sexuais e roxo: termos gerais. Termos com $*$ são aqueles que não aparecem na coleta das menores cidades na coleta por municípios

9.2 Árvore de decisão

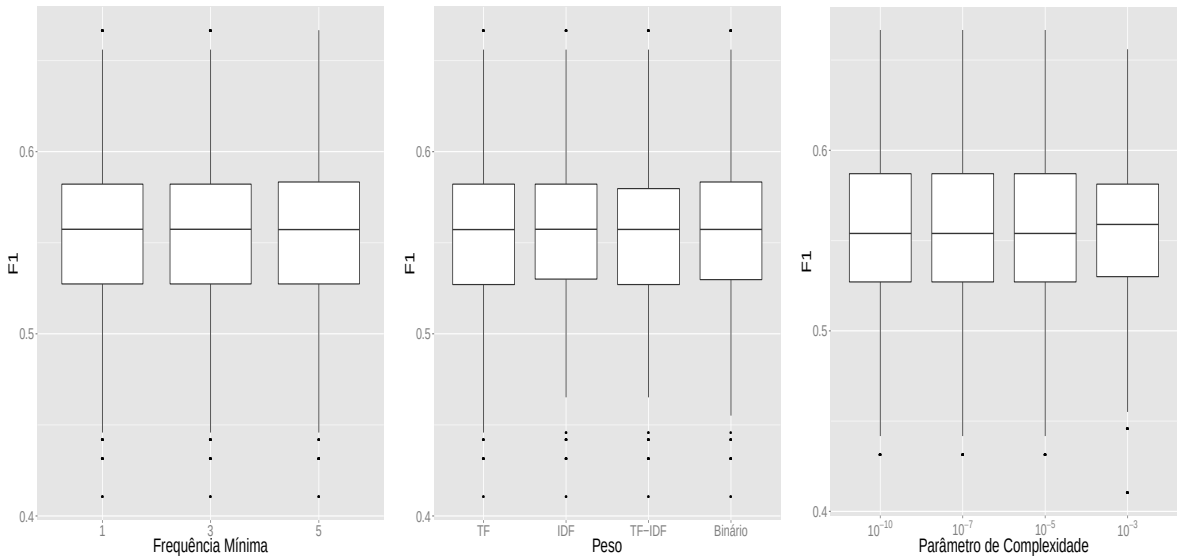

Figura 32 – Boxplots para frequência mínima do termo (esquerda), forma de ponderação para os termos (centro) e parâmetro de complexidade (direita): Critérios em que não se observou diferenças nos valores de F1 por níveis.

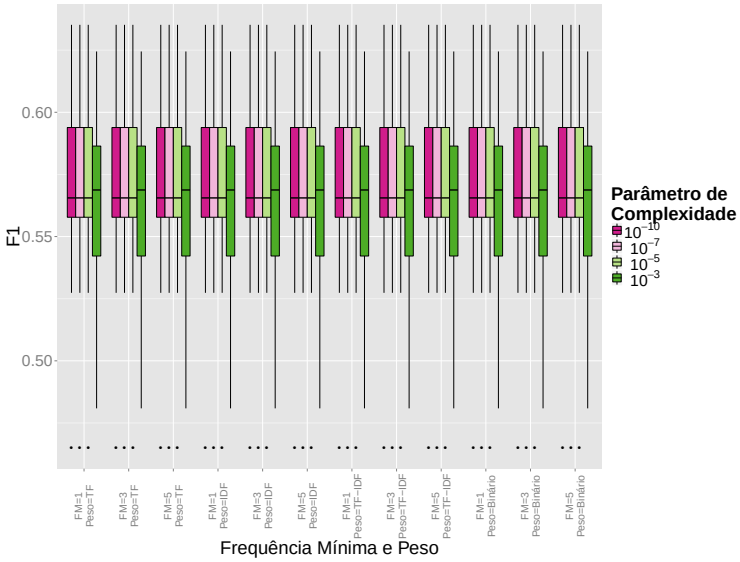

Figura 33 – Árvore de Decisão fixando critério número mínimo de observações por nó igual a 20 e medida Gini de quebra de nós segundo critério F1.

PCA	Par. de Complex.	Peso e Frequência Mínima do Termo											
		TF			IDF			TF-IDF			Binário		
		1	3	5	1	3	5	1	3	5	1	3	5
80%	10^{-10}	0.5114	0.5669	0.5680	0.5320	0.5320	0.5320	0.4872	0.4872	0.4872	0.5300	0.5397	0.5507
	10^{-7}	0.5114	0.5669	0.5680	0.5320	0.5320	0.5320	0.4872	0.4872	0.4872	0.5300	0.5397	0.5507
	10^{-5}	0.5114	0.5669	0.5680	0.5320	0.5320	0.5320	0.4872	0.4872	0.4872	0.5300	0.5397	0.5507
	10^{-3}	0.5175	0.5699	0.5923	0.5317	0.5317	0.5317	0.4981	0.4981	0.4981	0.5288	0.5497	0.5494
90%	10^{-10}	0.4952	0.5599	0.5843	0.5242	0.5242	0.5242	0.4884	0.4884	0.4884	0.5366	0.5444	0.5274
	10^{-7}	0.4952	0.5599	0.5843	0.5242	0.5242	0.5242	0.4884	0.4884	0.4884	0.5366	0.5444	0.5274
	10^{-5}	0.4952	0.5599	0.5843	0.5242	0.5242	0.5242	0.4884	0.4884	0.4884	0.5366	0.5444	0.5274
	10^{-3}	0.4938	0.5575	0.6049	0.5340	0.5340	0.5340	0.4979	0.4979	0.4979	0.5398	0.5514	0.5325

Tabela 12 – Medianas das quantidades F1 por critérios Peso, Frequência Mínima do Termo e Parâmetro de Complexidade para técnica Árvore de Decisão com PCA com número de componentes explicando 80% e 90% da variabilidade total dos termos.

9.2.1 SVM

Custo	Grau	Peso e Frequência Mínima do Termo													
		TF			IDF			TF-IDF			Binário				
		1	3	5	1	3	5	1	3	5	1	3	5		
10^{-5}	2	0.587	0.633	0.579	0.604	0.604	0.604	0.613	0.613	0.613	0.581	0.631	0.597	0.604	
	3	0.589	0.593	0.599	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.608	0.586	0.598	0.595
	5	0.599	0.583	0.577	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.585	0.595	0.582	
10^{-4}	2	0.592	0.632	0.623	0.618	0.618	0.618	0.624	0.624	0.624	0.584	0.636	0.619	0.617	
	3	0.589	0.592	0.599	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.600	0.596	0.598	0.599
	5	0.599	0.583	0.577	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.585	0.595	0.582	
10^{-3}	2	0.593	0.615	0.613	0.618	0.618	0.618	0.624	0.624	0.624	0.580	0.609	0.623	0.613	
	3	0.589	0.592	0.599	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.600	0.596	0.598	0.598
	5	0.599	0.583	0.577	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.585	0.595	0.582	
10^{-2}	2	0.593	0.617	0.604	0.618	0.618	0.618	0.624	0.624	0.624	0.580	0.606	0.611	0.611	
	3	0.589	0.592	0.599	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.600	0.596	0.598	0.597
	5	0.599	0.582	0.577	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.575	0.595	0.581	
10^{-1}	2	0.593	0.617	0.604	0.618	0.618	0.618	0.624	0.624	0.624	0.580	0.606	0.611	0.611	
	3	0.589	0.592	0.599	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.600	0.596	0.598	0.589
	5	0.599	0.507	0.508	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.510	0.499	0.556	
1	2	0.593	0.617	0.604	0.618	0.618	0.618	0.624	0.624	0.624	0.580	0.606	0.611	0.611	
	3	0.589	0.592	0.599	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.600	0.596	0.598	0.589
	5	0.599	0.507	0.508	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.510	0.499	0.556	
10	2	0.593	0.617	0.604	0.618	0.618	0.618	0.624	0.624	0.624	0.580	0.606	0.611	0.611	
	3	0.589	0.592	0.599	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.600	0.596	0.598	0.589
	5	0.599	0.508	0.508	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.510	0.499	0.556	
10^2	2	0.593	0.617	0.604	0.618	0.618	0.618	0.624	0.624	0.624	0.580	0.606	0.611	0.611	
	3	0.589	0.592	0.599	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.600	0.596	0.598	0.588
	5	0.599	0.499	0.512	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.480	0.499	0.553	
10^3	2	0.593	0.617	0.604	0.618	0.618	0.618	0.624	0.624	0.624	0.580	0.606	0.611	0.611	
	3	0.589	0.592	0.599	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.600	0.596	0.598	0.588
	5	0.599	0.504	0.512	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.480	0.500	0.554	
10^4	2	0.593	0.617	0.604	0.618	0.618	0.618	0.624	0.624	0.624	0.580	0.606	0.611	0.611	
	3	0.589	0.592	0.599	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.599	0.596	0.598	0.588
	5	0.599	0.501	0.531	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.480	0.499	0.555	
10^5	2	0.593	0.617	0.604	0.618	0.618	0.618	0.624	0.624	0.624	0.580	0.606	0.611	0.611	
	3	0.589	0.514	0.598	0.602	0.602	0.602	0.607	0.607	0.607	0.585	0.597	0.597	0.592	0.586
	5	0.599	0.504	0.545	0.579	0.579	0.579	0.575	0.575	0.575	0.592	0.480	0.486	0.555	
Médias		0.593	0.579	0.580	0.599	0.599	0.599	0.601	0.601	0.601	0.585	0.578	0.579		
			0.584			0.599			0.601			0.581			

Tabela 13 – Medida F1 para classificador SVM com *kernel* polinomial.

Custo	Grau	Peso e Frequência Mínima do Termo												Média	
		TF			IDF			TF-IDF			Binário				
		1	3	5	1	3	5	1	3	5	1	3	5		
10^{-5}	2	37.923	3.621	2.253	36.671	38.334	37.578	36.192	36.139	36.062	36.085	3.502	2.078	25.536	
	3	36.952	3.444	2.021	37.213	37.419	37.354	37.210	37.421	37.272	37.497	3.450	1.939	25.766	26.00264
	5	38.507	3.685	2.134	38.497	39.387	38.578	38.112	38.084	38.153	39.378	3.764	2.186	26.705	
10^{-4}	2	36.923	3.380	2.008	38.882	36.802	36.144	35.940	36.255	35.723	36.007	3.264	1.874	25.266	
	3	37.205	3.394	1.906	37.140	37.479	37.044	36.955	36.983	36.904	37.345	3.383	1.866	25.633	25.86667
	5	37.879	3.675	2.185	38.453	38.439	38.264	37.993	38.202	39.862	39.558	3.726	2.158	26.699	
10^{-3}	2	36.209	3.075	1.767	36.054	36.181	36.176	35.801	35.810	35.776	35.950	3.055	1.707	24.796	
	3	36.817	3.395	1.910	37.405	37.373	37.139	37.114	36.822	36.936	38.233	3.623	1.918	25.723	25.85567
	5	39.222	3.733	2.170	39.666	39.118	38.955	39.224	38.821	37.955	39.820	3.715	2.159	27.046	
10^{-2}	2	35.721	3.068	1.697	36.152	35.806	36.097	35.768	35.811	35.817	36.427	3.018	1.674	24.754	
	3	38.536	3.550	2.031	37.020	36.969	37.037	36.835	37.057	37.172	37.161	3.399	1.890	25.721	25.61222
	5	38.431	3.569	2.096	38.216	38.148	38.112	38.057	37.967	37.873	38.497	3.245	2.116	26.360	
10^{-1}	2	36.059	3.067	1.703	35.871	36.164	35.951	35.790	35.972	35.847	35.933	3.021	1.672	24.754	
	3	38.945	3.525	1.963	36.954	37.148	37.947	37.377	37.808	36.693	38.184	3.455	1.917	25.993	25.60086
	5	37.980	1.687	0.739	38.640	38.505	38.436	38.395	38.352	38.829	38.756	1.501	0.845	26.055	
1	2	35.790	3.050	1.716	36.299	36.085	35.966	36.162	35.973	35.864	36.938	3.085	1.706	24.886	
	3	37.902	3.548	1.954	38.227	37.073	36.962	36.657	36.641	37.020	37.930	3.473	1.929	25.776	25.54594
	5	38.198	1.648	0.780	38.347	38.585	38.500	38.362	38.069	38.137	38.689	1.527	0.862	25.975	
10	2	35.915	3.095	1.743	36.100	36.089	36.300	36.888	36.188	36.744	36.285	3.047	1.685	25.006	
	3	36.780	3.409	1.982	37.225	37.120	37.432	36.865	36.929	37.126	37.112	3.404	1.856	25.603	25.53547
	5	38.733	1.648	0.729	38.445	38.504	38.584	38.089	38.148	38.285	38.400	1.549	0.844	25.996	
10^2	2	35.971	3.085	1.804	37.113	36.774	36.013	36.100	35.778	36.765	35.889	3.044	1.672	25.000	
	3	36.817	3.438	1.934	37.170	37.040	37.347	36.817	36.762	36.870	36.970	3.400	1.861	25.535	25.56147
	5	38.258	0.503	0.580	38.375	39.989	38.650	38.375	41.244	38.080	38.446	0.419	0.860	26.148	
10^3	2	36.073	3.078	1.715	35.863	35.883	36.927	36.908	36.229	36.365	36.294	3.060	1.693	25.007	
	3	37.004	3.396	1.940	37.124	37.343	37.215	36.962	36.917	37.019	37.335	3.437	1.860	25.629	25.55372
	5	38.305	0.502	0.577	41.485	39.075	38.261	38.251	38.014	38.189	38.364	0.423	0.848	26.024	
10^4	2	36.001	3.104	1.716	36.258	36.362	36.160	36.178	36.015	36.134	36.514	3.023	1.746	24.934	
	3	37.022	3.401	1.899	37.091	36.994	37.193	36.766	36.971	37.162	37.031	3.332	1.871	25.561	25.50875
	5	38.973	0.501	0.408	38.543	38.778	38.932	38.499	38.450	39.791	38.230	0.422	0.844	26.030	
10^5	2	35.966	3.092	1.734	36.067	36.402	36.222	36.127	35.968	35.968	36.279	3.020	1.672	24.876	
	3	36.983	1.473	1.883	37.083	37.192	37.155	36.886	36.870	36.933	37.198	3.241	1.879	25.398	25.27681
	5	37.897	0.496	0.279	38.219	38.234	38.221	37.894	37.998	37.906	38.249	0.421	0.858	25.556	
Média		37.330	2.798	1.635	37.511	37.478	37.359	37.137	37.171	37.188	37.484	2.801	1.652		
			13.921			37.449			37.166			13.979			

Tabela 14 – Tempo para classificação por classificador SVM com *kernel* polinomial.

Custo	Gamma	Peso e Frequência Mínima do Termo													
		TF			IDF			TF-IDF			Binário				
		1	3	5	1	3	5	1	3	5	1	3	5		
1	10^{-4}	-	0.480	0.483	0.150	0.150	0.150	0.138	0.138	0.138	-	0.475	0.482	0.278	0.1790
	10^{-3}	-	-	-	-	-	-	-	-	-	-	-	-	-	
	10^{-2}	-	-	-	0.013	0.013	0.013	0.013	0.013	0.013	-	-	-	0.013	
10	10^{-4}	0.027	0.483	0.488	0.282	0.282	0.282	0.302	0.302	0.302	0.013	0.478	0.484	0.310	0.1652
	10^{-3}	-	-	-	0.026	0.026	0.026	0.026	0.026	0.026	-	-	-	0.026	
	10^{-2}	-	-	-	0.013	0.013	0.013	0.013	0.013	0.013	-	-	-	0.013	
10^2	10^{-4}	0.040	0.098	0.193	0.260	0.260	0.260	0.287	0.287	0.287	0.027	0.483	0.484	0.247	0.1335
	10^{-3}	-	-	-	0.026	0.026	0.026	0.026	0.026	0.026	-	-	-	0.026	
	10^{-2}	-	-	-	0.013	0.013	0.013	0.013	0.013	0.013	-	-	-	0.013	
10^3	10^{-4}	0.040	0.107	0.077	0.260	0.260	0.260	0.287	0.287	0.287	0.027	0.483	0.494	0.239	0.1295
	10^{-3}	-	-	-	0.026	0.026	0.026	0.026	0.026	0.026	-	-	-	0.026	
	10^{-2}	-	-	-	0.013	0.013	0.013	0.013	0.013	0.013	-	-	-	0.013	
10^4	10^{-4}	0.040	0.094	0.051	0.260	0.260	0.260	0.287	0.287	0.287	0.027	0.488	0.072	0.201	0.1105
	10^{-3}	-	-	-	0.026	0.026	0.026	0.026	0.026	0.026	-	-	-	0.026	
	10^{-2}	-	-	-	0.013	0.013	0.013	0.013	0.013	0.013	-	-	-	0.013	
10^5	10^{-4}	0.040	0.038	0.071	0.260	0.260	0.260	0.287	0.287	0.287	0.027	0.038	0.060	0.159	0.0897
	10^{-3}	-	-	-	0.026	0.026	0.026	0.026	0.026	0.026	-	-	-	0.026	
	10^{-2}	-	-	-	0.013	0.013	0.013	0.013	0.013	0.013	-	-	-	0.013	
Média		0.037	0.216	0.227	0.099	0.099	0.099	0.106	0.106	0.106	0.024	0.407	0.346		
		0.1675			0.0991			0.1060			0.2730				

Tabela 15 – Medida F1 para classificador SVM com *kernel* radial.

Custo	Gamma	Peso e Frequência Mínima do Termo													
		TF			IDF			TF-IDF			Binário				
		1	3	5	1	3	5	1	3	5	1	3	5		
1	10^{-4}	39.552	3.514	2.138	39.516	39.568	39.458	39.550	39.775	39.616	39.585	3.457	2.071	27.316	
	10^{-3}	40.582	4.474	2.469	40.905	40.784	40.831	40.518	40.885	40.518	40.361	4.396	2.361	28.257	28.157
	10^{-2}	40.893	5.396	3.335	41.291	41.036	41.187	41.428	41.184	41.177	41.206	5.364	3.291	28.899	
10	10^{-4}	38.165	2.798	1.662	38.375	38.466	38.135	38.446	38.466	38.617	38.271	2.721	1.608	26.310	
	10^{-3}	40.440	4.309	2.313	40.322	40.571	40.666	40.694	40.421	40.434	40.075	4.246	2.212	28.058	27.663
	10^{-2}	41.010	5.356	3.301	40.825	40.752	41.085	40.611	40.661	40.631	40.752	5.239	3.215	28.619	
10^2	10^{-4}	37.567	2.442	1.395	37.282	38.350	37.443	37.809	37.968	37.269	37.325	2.369	1.409	25.719	
	10^{-3}	40.485	4.131	2.169	40.150	40.131	40.541	40.224	40.361	40.588	40.058	4.020	2.101	27.913	27.402
	10^{-2}	40.875	5.330	3.313	40.678	40.709	41.003	40.616	40.712	40.636	40.505	5.298	3.215	28.574	
10^3	10^{-4}	37.389	2.223	1.329	37.669	37.379	37.461	37.482	37.409	37.482	37.353	2.181	1.300	25.554	
	10^{-3}	40.128	4.182	2.130	40.668	40.243	40.323	40.549	40.243	40.618	40.506	4.127	2.066	27.981	27.391
	10^{-2}	40.560	5.366	3.291	41.036	41.013	40.830	40.646	40.729	41.222	40.551	5.227	3.183	28.637	
10^4	10^{-4}	37.353	2.088	1.289	37.329	37.549	37.341	37.329	37.802	37.332	37.179	2.090	1.272	25.496	
	10^{-3}	40.047	4.156	2.142	40.238	40.570	40.380	40.215	40.159	40.482	40.061	4.129	2.068	27.887	27.345
	10^{-2}	40.602	5.317	3.274	40.747	40.715	41.011	40.991	41.049	40.652	41.053	5.232	3.212	28.654	
10^5	10^{-4}	37.561	2.137	1.306	37.297	37.528	37.238	37.692	37.668	37.490	37.236	2.136	1.312	25.550	
	10^{-3}	40.033	4.179	2.123	40.870	40.428	40.155	40.642	40.342	40.239	40.436	4.111	2.070	27.969	27.366
	10^{-2}	40.634	5.317	3.256	40.880	40.647	40.906	40.735	40.804	40.720	40.615	5.218	3.215	28.578	
Média		39.659	4.039	2.034	39.782	39.802	39.777	39.787	39.813	39.762	39.618	3.975	2.287		
		15.348			39.787			39.7877			15.2938				

Tabela 16 – Tempo para classificação por classificador SVM com *kernel* radial.

Referências

AURÉLIO. *Crime e Criminalidade*. 2013. www.dicionariodoaurelio.com. Acessado em 28/08/2013. X

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. New York, NY, USA: ACM, 1992. (COLT '92), p. 144–152. ISBN 0-89791-497-X. Disponível em: <http://doi.acm.org/10.1145/130385.130401>. XXXII

BREIMAN, L. et al. *Classification and regression trees*. [S.l.]: CRC press, 1984. XLII

CHU, S. *Stat 8053, Fall 2013: Recursive Partitioning*. 2013. http://www.amstat.org/publications/jse/v9n2/datasets.chu.html.Dataondiamondpricing. XLIV, XLV

CORTES, C.; VAPNIK, V. Support-vector networks. In: *Machine Learning*. [S.l.: s.n.], 1995. p. 273–297. XXXIII, XXXVIII

DIMITRIADOU, E. et al. Misc functions of the department of statistics (e1071), tu wien. *R package*, p. 1–5, 2008. XXXVIII, XLVIII

FEINERER, I.; HORNIK, K. *tm: Text Mining Package*. [S.l.], 2014. R package version 0.5-10. Disponível em: <http://CRAN.R-project.org/package=tm>. XVI, XL

FLACH, P. *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. [S.l.]: Cambridge, 2012. XXXVIII

GESMANN, M.; CASTILLO, D. de. googlevis: Interface between r and the google visualisation api. *The R Journal*, v. 3, n. 2, p. 40–44, December 2011. Disponível em: <http://journal.r-project.org/archive/2011-2/RJournal_2011-2_Gesmann+de~Castillo.pdf>. LIX

GIRAUDOUX, P. *pgirmess: Data analysis in ecology*. [S.l.], 2014. R package version 1.5.9. Disponível em: <http://CRAN.R-project.org/package=pgirmess>. XLI

HÄRDLE, W.; SIMAR, L. *Applied multivariate statistical analysis*. [S.l.]: Springer, 2007. XXXIX

HASTIE, T. et al. *The elements of statistical learning*. [S.l.]: Springer, 2009. XIII

JAMES, G.; WITTEN, D.; HASTIE, T. *An Introduction to Statistical Learning: With Applications in R*. [S.l.]: Taylor & Francis, 2014. XXXVIII

JOACHIMS, T. *Text categorization with support vector machines: Learning with many relevant features*. [S.l.]: Springer, 1998. XV, XVI, XIX, XXXIII, XLVIII, LVI

<KINOGUCHI@UCDAVIS.EDU>, J. L. G. Y. R. G. <ygl@math.uwaterloo.ca>; W. L. Wallace Hui <wlwhui@uwaterloo.ca>; Vyacheslav Lyubchich <vlyubchich@uwaterloo.ca>; Weiwen Miao <miao@macalester.edu>; K. N. *lawstat: An R package for biostatistics, public policy, and law*. [S.l.], 2013. R package version 2.4.1. Disponível em: <http://CRAN.R-project.org/package=lawstat>. XL

LANG, D. T. *RJSONIO: Serialize R objects to JSON, JavaScript Object Notation*. [S.l.], 2014. R package version 1.3-0. Disponível em: <http://CRAN.R-project.org/package=RJSONIO>. XXIII

LEARN scikit. *http://scikit-learn.org*. 2014. Online. Acessado em 20/02/2014. XXVIII

LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007. XII, XIII, XXXIV, XXXV, XXXVI, XXXVII, XXXVIII

MINGOTI, S. A. *Análise de dados através de métodos de estatística multivariada: uma abordagem aplicada*. [S.l.]: Editora UFMG, 2005. XXXIX

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning*. [S.l.]: The MIT Press, 2012. ISBN 026201825X, 9780262018258. XXXV, XXXVI, XXXVIII

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2014. Disponível em: <http://www.R-project.org>. XVI, XXIII, XL, LIX

ROKACH, L.; MAIMON, O. Decision trees. In: *Data Mining and Knowledge Discovery Handbook*. [S.l.]: Springer, 2005. p. 165–192. XXXI, XXXII

RStudio; Inc. *shiny: Web Application Framework for R*. [S.l.], 2014. R package version 0.10.2.2. Disponível em: <http://CRAN.R-project.org/package=shiny>. LIX

SAMMUT, G. I. W. C. *Encyclopedia of Machine Learning*. [S.l.]: Springer, 2010. XXXIV, XXXVI, XXXVIII

SHIMODAIRA, H. *Text Classification using Naive Bayes*. 2014. http://www.inf.ed.ac.uk/teaching/courses/inf2b/learnnotes/inf2b-learn-note07-2up.pdf. Acessado em 20/03/2014. XV, XXX

THERNEAU, T.; ATKINSON, B.; RIPLEY, B. *rpart: Recursive Partitioning and Regression Trees*. [S.l.], 2014. R package version 4.1-6. Disponível em: <http://CRAN.R-project.org/package=rpart>. XLII

TOKER, G.; KIRMEMIS, O. Text categorization using k nearest neighbor classification. *Survey Paper, Middle East Technical University*. XV

TREEPLAN. *Introduction to decision Trees*. 2014. http://www.treeplan.com/chapters/introduction-to-decision-trees.pdf. Acessado em 02/03/2014. XXXI

TWITTER. *API 1.1 Twitter*. 2013. https://dev.twitter.com/docs/api/1.1. Acessado em 27/08/2013. XXI

VELOSO, A.; MEIRA, W.; ZAKI, M. J. Lazy associative classification. In: IEEE. *Data Mining, 2006. ICDM'06. Sixth International Conference on*. [S.l.], 2006. p. 645–654. LXIV

Wickham; Hadley. Reshaping data with the reshape package. *Journal of Statistical Software*, v. 21, n. 12, 2007. Disponível em: <http://www.jstatsoft.org/v21/i12/paper>. LIX

WICKHAM, H. The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, v. 40, n. 1, p. 1–29, 2011. Disponível em: <http://www.jstatsoft.org/v40/i01/>. LIX

WICKHAM, H.; FRANCOIS, R. *dplyr: A Grammar of Data Manipulation*. [S.l.], 2014. R package version 0.3.0.2. Disponível em: <http://CRAN.R-project.org/package=dplyr>. LIX

WIKIPEDIA. *API*. 2013. http://pt.wikipedia.org/wiki/API. Acessado em 28/08/2013. XXI

ZHU, X. *Basic Text Process*. 2014. http://pages.cs.wisc.edu/~jerryzhu/cs769/text_preprocessing.pdf. Acessado em 10/02/2014. XVII