Universidade Federal do Paraná

Cauê Oliveira Gobbo Rodrigues

Análise de classificadores de texto baseados em aprendizado de máquina aplicada ao contexto de chamados de suporte

Curitiba PR
2018

Cauê Oliveira Gobbo Rodrigues

Análise de classificadores de texto baseados em aprendizado de máquina aplicada ao contexto de chamados de suporte

Trabalho de Graduação apresentado como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação, no curso de Graduação em Bacharelado em Ciência da Computação, Setor de Informática, da Universidade Federal do Paraná.

Orientador: Fabiano Silva.

Curitiba PR
2018

Resumo

Este trabalho aborda o problema existente em empresas que prestam suporte ou outra atividade que exija delegação de atividades de acordo com as necessidades apresentadas por seus clientes, ou seja, a tratativa do problema ou da questão levantada inicialmente deve ser encaminha a uma pessoa ou equipe responsável sobre o tema. Tendo em vista a necessidade da resolução do problema apresentado por seus clientes em tempo hábil, o tempo perdido na delegação da solicitação pode impactar negativamente tanto na qualidade do serviço prestado quanto na imagem da empresa que o fornece. Para fornecer uma possível solução para este problema, foram utilizados métodos de aprendizado de máquina para efetuar a classificação automática de chamados de suporte e sua atribuição às equipes inicialmente pré-estabelecidas. Apresentaremos então alguns algoritmos devidamente configurados para a aplicação e, por fim, os resultados por estes obtidos, demonstrando quais modelos são adequados para realizar a atividade de classificação de chamados de suporte.

Palavras-chave: Classificação de textos, Aprendizado de máquina, Chamados de suporte, Categorização.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

SVM	*Support Vector Machines*
TF	*Term Frequency*
IDF	*Inverse Document Frequency*
SAP	*Systeme, Anwendungen und Produkte in der datenverarbeitung*
CO	*Controlling*
FI	*Financial Accounting*
SD	*Sales and Distribution*
MM	*Materials Management*
SGD	*Stochastic Gradient Descent*
LSVC	*Linear Support Vector Classifier*
BNB	*Bernoulli Naive Bayes*
MNB	*Multinomial Naive Bayes*

Capítulo 1

Introdução

Não é difícil notar o crescimento da computação nas últimas décadas em todos os setores da sociedade, desde simples atividades cotidianas até processos de produção inteiros. Essa popularização da informática, provavelmente se dê, devido a agilidade, segurança e baixo custo que ela proporciona.

Isso fez com que a computação abrangesse uma quantidade enorme de setores na sociedade, gerando, por sua vez, a necessidade de cada vez mais profissionais especializados dedicados a cada uma dessas áreas, uma vez que cada uma dessas tecnologias necessita de um suporte específico, com conhecimentos relacionados à área na qual se aplicam.

Em vista do crescimento das áreas de atuação da informática, assim como das diferentes especialidades de cada profissional da área de computação, um dos primeiros problemas a serem resolvidos por empresas que trabalham com suporte ou consultoria voltadas à informática é, atribuir um problema ao devido responsável, que possua os conhecimentos necessários para resolução do problema.

No entanto, direcionar um tema ao órgão solucionador correto pode ser uma tarefa complexa dependendo da amplitude do escopo da atuação da empresa, principalmente baseando-se apenas em relatos fornecidos pelos usuários finais da tecnologia, que utilizam a computação apenas como uma ferramenta de trabalho.

Felizmente, esse problema pode ser resolvido utilizando-se a própria computação, o uso do aprendizado de máquinas para a realização de categorização de textos tem se mostrado muito eficaz em diversas aplicações, Sebastiani (2002), tanto em modelos de aplicações na vida real, como identificação de spam, Hovold (2005), quanto na literatura, Joachins (1998).

Esse trabalho irá abordar algumas das técnicas já existentes de classificação de texto com o intuito de identificar qual delas melhor se aplica à classificação de chamados de suporte. O próximo capítulo fará uma revisão dos conceitos fundamentais de cada técnica utilizada no experimento. O capítulo três irá relatar como foi feita a experiência e como foram parametrizados os algoritmos. E por fim, no último capítulo, serão expostos os resultados e as conclusões do estudo.

Capítulo 2

Conceitos

A classificação de documentos é uma subárea de aprendizado de máquinas, o aprendizado de máquinas por sua vez, é o campo da computação que estuda máquinas que podem fazer inferência sobre algum conteúdo e, a partir disso, melhorar a performance da atividade que realizam, Mitchell (1997). Então, a classificação é a capacidade de um algoritmo, a partir de uma informação externa, de ser capaz de reconhecer diferenças em um conjunto de documentos e efetuar a correta distinção entre eles. Podemos separar ainda a classificação em dois tipos, a não supervisionada, onde a máquina não tem acesso à modelos previamente classificados, o algoritmo deve realizar a atividade apenas nos modelos de entrada, e a supervisionada, que consistem em fornecer documentos já classificados ao algoritmo a fim de que ele possa criar regras de classificação a partir dos dados de treino.

A categorização de textos tem sido aplicada a diversas áreas, como detecção de *spam*, Hovold (2005), e identificação de sentimentos, Bo Pang (2002), e tem se mostrado muito versátil. A capacidade de tratar uma grande quantidade de dados com um bom desempenho, faz com que essa atividade seja praticamente indispensável, tendo em vista o crescimento acelerado da geração de dados digitais que temos atualmente.

No entanto, a tratativa de um texto, considerando-se toda informação nele contida, não é facilmente reconhecida pela máquina, pois não existe uma regra que defina quais palavras devam ser utilizadas na formação de uma frase ou um único modelo possível para transmitir uma mensagem. O texto então, antes de ser analisado por uma máquina, deve ser reescrito de uma maneira mais inteligível à mesma.

Para realizar a tarefa de adequar os documentos textuais aos algoritmos de categorização automática foram criadas representações baseadas em modelos estatísticos, que fornecem informações de um modo diferente como utilizamos usualmente, mas extremamente útil para ser tratado computacionalmente.

Foram escolhidas representações textuais e modelos de classificadores que já são conhecidamente bons na realização de classificação de texto, conforme demonstrado por Joachins (1998) e Hovold (2005), as representações escolhidas foram *Bag of Words* e TF-IDF, os classificadores escolhidos foram baseados em *Naive Bayes* e SVM.

2.1 Representações Textuais

Existem representações textuais específicas, responsáveis por traduzir o texto legível a um humano a um modelo estatístico compreensível aos métodos computacionais utilizados, na realização deste trabalho foram utilizadas dois modelos diferentes dessas representações, o *Bag*

of Words e o TF-IDF (*Term Frequency - Inverse Document Frequency*), mais profundamente analisados por Anne Kao (2007).

2.1.1 *Bag of Words*

O modelo *Bag of Words* é uma das mais populares representações textuais voltadas à computação, a ideia por trás dela, no entanto, é extremamente simples. Esse método descarta qualquer informação quanto à ordenação das palavras, realizando apenas uma contagem dos termos.

A partir da contagem dos termos presentes no documento é realizada a criação de um dicionário, esse dicionário contem o termo e a quantidade de vezes em que ele aparece no texto. Existem palavras, porém, encontradas em grande quantidade na grande maioria dos textos, mas que não carregam nenhum sentido quanto a classificação do mesmo, pois fazem parte apenas da representação textual, como verbos de ligação, pronomes, advérbios, entre outros, esses itens são conhecidos como *stopwords*.

As *stopwords* podem ser tratadas de diversas maneira, como a criação de um dicionário contendo-as ou contabilizando a frequência em que aparecem em diferentes documentos, sua identificação é importante para que um documento não seja relacionado a uma classe baseando-se nesse conjunto de palavras.

A metodologia da contagem de palavras contem a informação de quais termos possuem maior relevância no documento, essa informação é necessária para a realização de algumas técnicas de categorização demonstradas mais adiante.

2.1.2 *Term Frequency - Inverse Document Frequency*

O outro modelos de representação textual foi o TF-IDF, diferentemente do *Bag of Words*, esse modelo não carrega valores absolutos, como a contagem dos termos, mas sim valores relativos, o que ajuda a escalar a representação de maneira homogênea, independentemente do tamanho dos documentos.

A primeira etapa deste método, o *term frequency*, opera de maneira similar ao modelo anterior, realizando a contagem dos termos, porém, ao invés de armazenar a quantidade em que o termo aparece no texto, ele retorna a proporção entre essa quantidade e o total de palavras contidas no documento.

Apesar da diferença nessa primeira etapa, o problema com as *stopwords* que ocorria anteriormente se mantém nesse caso, pois a aplicação do método acima irá gerar uma frequência consideravelmente alta de palavras utilizadas na estruturação do texto, para sanar este problema é utilizado o IDF.

O IDF, *inverse document frequency*, tem o papel de atenuar a significância de palavras que não tem importância na classificação dos textos, para isso é calculada o *log* da proporção entre a quantidade total de documentos sobre a quantidade de documentos que possuem o termo em questão.

Ambos, TF e IDF são multiplicados a fim de ter-se um balanceamento nos pesos dos termos, configurando uma representação que valorize os termos de maior relevância no documento, Jones (1972).

2.2 Classificadores de Texto

Como já mencionado, os classificadores são os responsáveis por, dado um documento, discernir a qual classe pré-estabelecida o mesmo pertence. Para realizar esta atividade é necessário que tais classificadores tenham um conhecimento prévio sobre as classes, a fim de serem capazes de decidir qual delas será escolhida. A realização desta tarefa é dada por um conjunto de dados de treinos, já categorizados, que servirão de base para a classificação dos demais dados.

A realização desta tarefa pode ser realizada utilizando-se vários métodos estatísticos existentes, neste trabalho, baseamo-nos em dois modelos, o *Naive Bayes* e o SVM.

2.2.1 *Support Vector Machine*

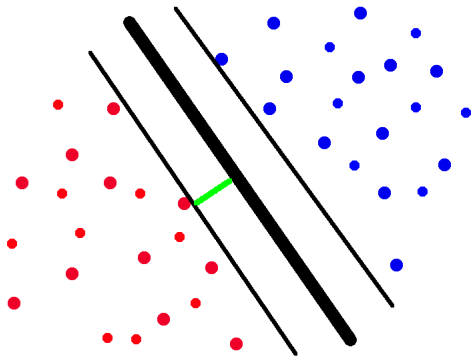
Os *Support Vector Machines* são um conjunto de métodos de algoritmos que utilizam aprendizagem supervisionada e realizam reconhecimento de padrões nos documentos fornecidos durante a fase de treinamento com o intuito de encontrar as diferenças entre as classes existentes.

A partir dos modelos obtidos na etapa de treino este método gera um hiperplano que tenta dividir os dados da melhor maneira possível, em outras palavras, criar uma linha de divisão entre as classes, afastando-a dos itens a serem divididos, maximizando a margem entre a linha e o conjunto de dados.

A criação do hiperplano trata-se de um problema de otimização, que consiste em maximizar as distâncias entre ele e a localização dos documentos no espaço de busca, ou, como é mais conhecido na literatura, minimizar os resíduos.

Esse modelo utiliza a técnica "um contra um"que consiste em realizar a comparação sempre entre duas classes distintas, uma vez que a comparação é binária, e escolher aquela com a qual o documento melhor se enquadra. Essa característica evita que seja necessário tratar uma quantidade muito grande de classes, limitando-se sempre à duas, Joachins (2002).

Figura 2.1: Representação do hiperplano



A figura 2.1 demonstra o posicionamento do hiperplano entre os dados, os círculos azuis e vermelhos representam textos de categorias distintas, as linhas pretas finas representam os vetores de suporte, a linha grossa demonstra o posicionamento do hiperplano e a linha verde mostra a margem de um dos lados.

2.2.2 *Naive Bayes*

Este conceito baseia-se no teorema de *Bayes*, equação 2.1, este teorema utiliza um método probabilístico onde assume que cada termo é independente dos demais, sendo possível

calcular a probabilidade de, dado um termo, calcular a probabilidade do documento pertencer a determinada classe.

Esse teorema trabalha com os conceitos de probabilidade a priori e probabilidade a posteriori, a primeira, indica as chances determinado documento pertencer a uma classe antes que sejam verificadas quaisquer características deste documento. A probabilidade a posteriori por sua vez, julga as informações obtidas para realizar a classificação do documento. É possível dizer que, enquanto a probabilidade a priori esta relacionada com a especificidade, a probabilidade a priori se relaciona com a sensibilidade. O conceito formal de probabilidade a priori e posteriori é demonstrado por Murphy (2006).

No teorema de *Bayes* $P(A)$ denota a probabilidade de A e $P(A|B)$ denota a probabilidade de A condicional a B.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

Capítulo 3

Desenvolvimento

Para a realização deste trabalho serão utilizados quatro modelos de classificadores diferentes, dois algoritmos baseados em SVM (*Support Vector Machines*) e dois em *Naive Bayes*. A base de dados para a realização dos treinos e dos testes dos algoritmos são provenientes de uma empresa privada de consultoria SAP.

Para a realização dos testes nos modelos escolhidos foi utilizada a biblioteca *Scikit Learn*, que dispõe tanto as funções necessárias para a criação das representações textuais estatísticas quanto para a modelagem dos classificadores utilizados, além de possuir métodos eficientes para a parametrização dos mesmos, efetuando várias execuções paralelas do mesmo algoritmo com diferentes configurações.

3.1 Dados

A SAP (*Systeme, Anwendungen und Produkte in der datenverarbeitung*, ou em português, Sistemas, Aplicativos e Produtos para processamento de dados), é uma empresa alemã proprietária de um software de gestão de empresas homônimo, o qual oferece um conjunto de módulos integrados, os quais possuem as funcionalidades essenciais para o funcionamento de empresas de grande porte.

Os módulos presentes no sistema, também conhecidos por módulos SAP, são divididos de acordo com o escopo aos quais estão responsáveis dentro de uma empresa, por exemplo o módulo FI (*Financial Accounting*), diz respeito à parte financeira da empresa, como impostos e inventário, enquanto o módulo CO (*Controlling*) trata do controle da empresa, como ordens internas e centro de custo.

Existem dezenas de módulos SAP, para a realização deste trabalho foram selecionados quatro deles, FI, CO, SD (*Sales and Distribution*), relacionado às vendas da organização, e MM (*Materials Management*), responsável pelas compras e inventário. A escolha destes módulos se deu pelos seguintes motivos:

- São módulos críticos, que impactam diretamente o faturamento da empresa;
- Estão intimamente relacionados, fazendo com que muitas vezes seja difícil distingui-los;
- Representam uma grande parcela dos chamados abertos à empresa de consultoria, nos fornecendo assim uma grande quantidade de dados;
- Os chamados são normalmente abertos por pessoas da área financeira ou administrativa da empresa, que desconhecem o funcionamento do sistema de suporte.

Para a realização deste trabalho, foram coletados mais de 33 mil chamados abertos à uma empresa de consultoria SAP, feito isso, foram separados os chamados pertencentes aos módulos supracitados, obtendo a distribuição dos documentos entre os quatro módulos conforme mostra a tabela 3.1. Todos os dados coletados são privados.

Tabela 3.1: Distribuição dos Documentos

Módulo	Quantidade de Documentos
CO	529
FI	1117
MM	1721
SD	1856

O conteúdo dos chamados que foram utilizados é composto majoritariamente de texto e alguns marcadores, que são ignorados automaticamente no momento em que a representação é organizada, nenhum dos chamados foi editado ou alterado. Apenas foi realizada a alteração da codificação para UTF-8 para remover possíveis problemas em caracteres especiais.

3.2 Algoritmos

3.2.1 *Stochastic Gradient Descent*

Um dos modelos utilizados para a realização dos experimentos foi o SGD (*Stochastic Gradient Descent*), este modelo, baseado em SVM, utiliza regressão para encontrar um hiperplano linear que separe de maneira satisfatória os chamados que estão sendo analisados. O SGD utiliza o método dos mínimos quadrados como função de otimização, que consiste em calcular a diferença entre os valores estimados e os dados observados e somar os resultados obtidos, esse número é conhecido como resíduo, e, para um bom resultado final, deve ser reduzido ao menor valor possível.

O funcionamento deste algoritmo ocorre da seguinte maneira, primeiramente o algoritmo escolhe de maneira estocástica, ou aleatória, um ponto da função a ser minimizada, e então, utilizando os pontos vizinhos, descobre se o segmento da função é ascendente ou descendente, feito isso, ele caminha na direção que retorna os menores valores - ou menor resíduo - até que encontre um mínimo local na função.

Esse passos são realizados 1000 vezes, independentemente do valor final do resíduo nas iterações, e o menor dos mínimos locais é definido como mínimo global. Vale ressaltar que resíduos muito pequenos - menores que $1e-3$ - são ignorados durante a soma para obtenção do resíduo geral, sendo considerados nulos para facilitar os cálculos e a convergência do algoritmo.

A representação textual escolhida para ser aplicada a este algoritmo foi a TF-IDF, o fato dela calcular as frequências de cada termo faz com que essa representação forneça informações mais relevantes do que a simples contagem realizada pela *bag of words*. Além das operações básicas para montagem dessa estrutura, também foram removidos os termos com incidência maior do que 95% na etapa de treino.

3.2.2 *Linear Support Vector Classifier*

O outro modelo baseado em SVM escolhido foi o LSVC (*Linear Support Vector Classifier*), assim como o modelo anterior, utiliza um modelo linear para realizar a categorização

dos documentos e também utiliza o método dos mínimos quadrados para criação do hiperplano, pode-se dizer que este modelo é uma das mais simples e básicas opções que o SVM oferece.

A diferença deste algoritmo para com o anterior se dá ao fato de que o mesmo não utiliza regressão, devido a isto, ele não é limitado por um número máximo de iterações mas sim por um valor de tolerância a ser alcançado pelo resíduo, o limite de tolerância que retornou os melhores resultados foi de $1e-5$.

Ademais, todos os outros fatores se assemelham aos do modelo SGD, incluindo a modelagem da representação textual e seus parâmetros, TF-IDF com aceitação de todas as palavras com incidência menor do que 95% durante a fase de treinamento.

3.2.3 *Bernoulli Naive Bayes*

Além de algoritmos baseados em SVM, também foram utilizados dois modelos baseados em *Naive Bayes*, que utilizam o teorema probabilístico de *Bayes* para realizar a classificação dos documentos, esses métodos, por serem baseados em um cálculo estatístico já consolidado, são conhecidos por serem práticos, eficientes mesmo com uma pequena base de treino e utilizarem poucos recursos computacionais.

Um dos algoritmos utilizados foi o BNB (*Bernoulli Naive Bayes*), este algoritmo trabalha com a existência ou inexistência dos termos dentro de um documento, ou seja, é irrelevante se uma palavra aparece uma ou cem vezes em um documento, pois a única informação armazenada é de que a palavra aparece no documento. Isso faz com que esse algoritmo tenha bons resultados quando a base de dados é composta por textos curtos.

Para a execução deste método as probabilidades a priori das classes são ignoradas, ou seja, é considerado que os documentos são distribuídos de maneira uniforme em todas as classes. A probabilidade a posteriori é a proporção do número de documentos que contém o termo sobre a quantidade total de documentos na base de treino. A constante utilizada na técnica de *Additive Smoothing* - ou constante de suavização Laplace - que retornou os melhores resultados foi $1e-2$, essa constante é um estimador da probabilidade a posteriori utilizado por *Bayes*.

Assim como os algoritmos baseado em SVM, o BNB apresentou resultados superiores utilizado a representação TF-IDF, os parâmetros utilizados são equivalentes aos utilizados anteriormente, descartando apenas termos com incidência superior a 95% na base de testes.

3.2.4 *Multinomial Naive Bayes*

Outra técnica também derivada da *Naive Bayes* que foi utilizada nos experimentos foi a MNB (*Multinomial Naive Bayes*), seu embasamento estatístico em nada difere do modelo exposto anteriormente, a diferença entre ambos os algoritmos é como são utilizados os dados e como é feita a análise das probabilidades.

A primeira diferença é que este modelo aprende as probabilidades a priori das classes, essa probabilidade é a relação do número de documentos pertencentes à classe em questão com o número total de documentos presentes na base de treino, ou seja, uma classe que possui mais representantes durante a etapa de treino tem, automaticamente, maior probabilidade de ser escolhida durante os testes, levando em consideração a probabilidade a priori apenas.

A probabilidade a posteriori desse modelo também difere do anterior, enquanto o BNB armazena a informação da presença ou não de um termo em dado documento, o MNB realiza a contagem da quantidade de vezes que os termos aparecem nos documentos, devido a isto, este algoritmo que utiliza a representação textual *Bag of Words*. A única restrição dessa representação

é considerar apenas as palavras que aparecem com uma frequência inferior a 97% durante os testes.

3.3 Métricas

A análise dos resultados foi baseada nos conceitos estatísticos de sensitividade e especificidade, ou, taxa de verdadeiros positivos e taxa de verdadeiros negativos, ou ainda, precisão e *recall*, respectivamente.

Esses valores podem ser melhor entendidos observando uma matriz de confusão, como demonstra a tabela 3.2

Tabela 3.2: Matriz de confusão

	CO	FI	MM	SD
CO	152	5	4	3
FI	6	323	13	2
MM	0	4	533	9
SD	0	12	11	490

A matriz acima nos fornece os valores de precisão e *recall* de uma execução de um dos classificadores utilizados neste trabalho, as linhas desta matriz representam a quantidade de chamados que pertencem ao módulo, enquanto suas colunas evidenciam a qual classe eles foram distribuídos.

A precisão do algoritmo é obtida considerando-se a as linhas da matriz, tomemos por observação o módulo CO, em um total de 164 chamados deste módulo, 152 foram categorizados corretamente, logo sua precisão é de $152/164 = 0,93$ - Tabela 3.3.

Tabela 3.3: Matriz de confusão evidenciando a precisão

	CO	FI	MM	SD
CO	152	5	4	3
FI	6	323	13	2
MM	0	4	533	9
SD	0	12	11	490

O *recall* por sua vez, é relatado pelas colunas desta matriz, 158 chamados foram classificados como CO, dos quais 152 realmente pertenciam ao módulo, retornando um *recall* de $152/158 = 0,96$ - Tabela 3.4.

Tabela 3.4: Matriz de confusão evidenciando o *recall*

	CO	FI	MM	SD
CO	152	5	4	3
FI	6	323	13	2
MM	0	4	533	9
SD	0	12	11	490

A partir da precisão e do *recall*, é possível obter o F-score, que foi a métrica de comparação utilizada para comparar os algoritmos testados neste trabalho. O F-score é obtido

através da média harmônica dos resultados obtidos anteriormente, utilizando a fórmula 3.1. Logo o valor de F-score para o exemplo acima é de aproximadamente 0,94

$$Fscore = 2 \times \frac{precisao \times recall}{precisao + recall} \tag{3.1}$$

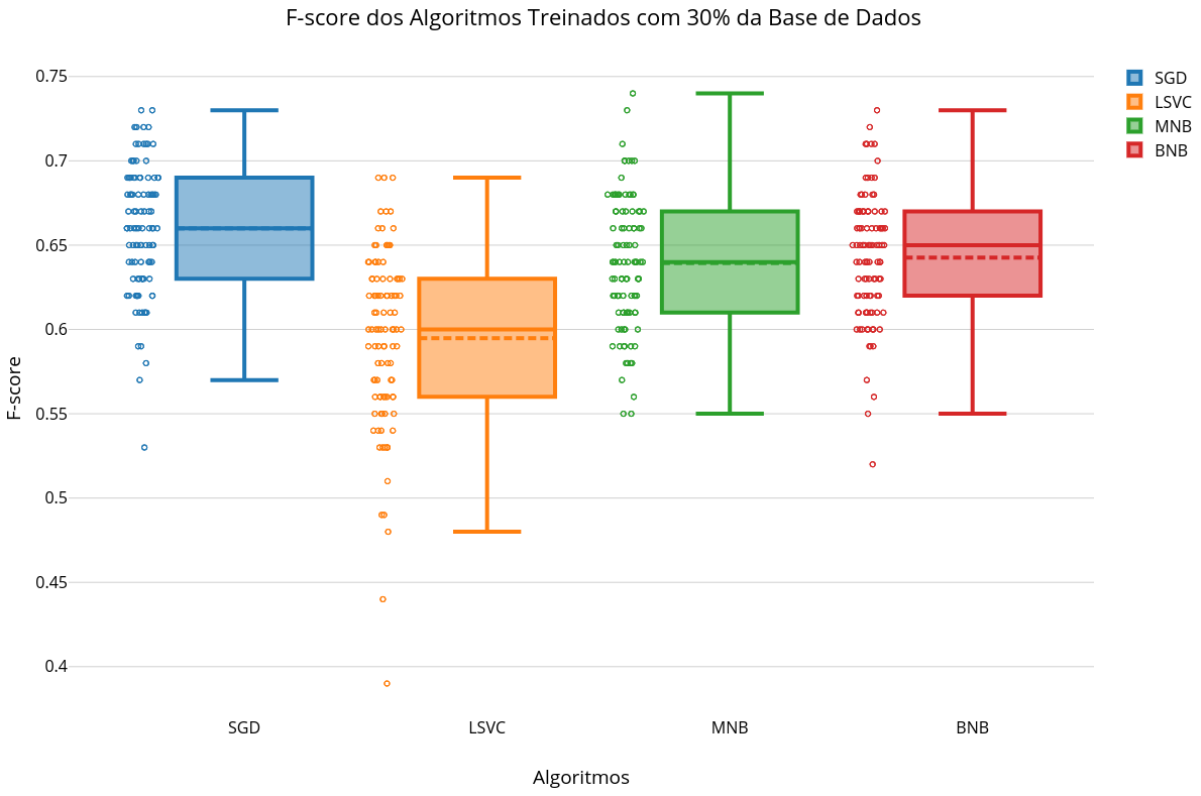
Capítulo 4

Resultados

Utilizando o método de cálculo do F-score para gerar a pontuação de cada execução, foram realizados os seguintes testes para a avaliação de desempenho dos quatro algoritmos, primeiramente foi realizada uma bateria de 100 execuções para cada algoritmo utilizando 30% da base de dados como conjunto de treino - cerca de 1567 chamados - e o restante para a realização de testes.

O objetivo deste primeiro teste foi visualizar o comportamento das diferentes metodologias com uma base de treinos relativamente pequena, quais deles possuem bons resultados, quais são bem comportado, ou seja, retornam pontuações semelhantes em várias iterações, ou ainda evidenciar um processo de aprendizagem muito lenta de algum algoritmo. Os resultados obtidos podem ser visualizados na figura 4.1 que demonstra o gráfico *boxplot* com uma escala que varia entre 0,4 e 0,75, a linha contínua dentro da caixa representa a mediana e o tracejado representa o valor médio, cada círculo representa uma execução do algoritmo.

Figura 4.1: F-score dos algoritmos treinados com 30% da base de dados



Apesar dos resultados obtidos não serem considerados satisfatórios devido ao tamanho reduzido da base de treino, é possível identificar algumas características nos algoritmos utilizados.

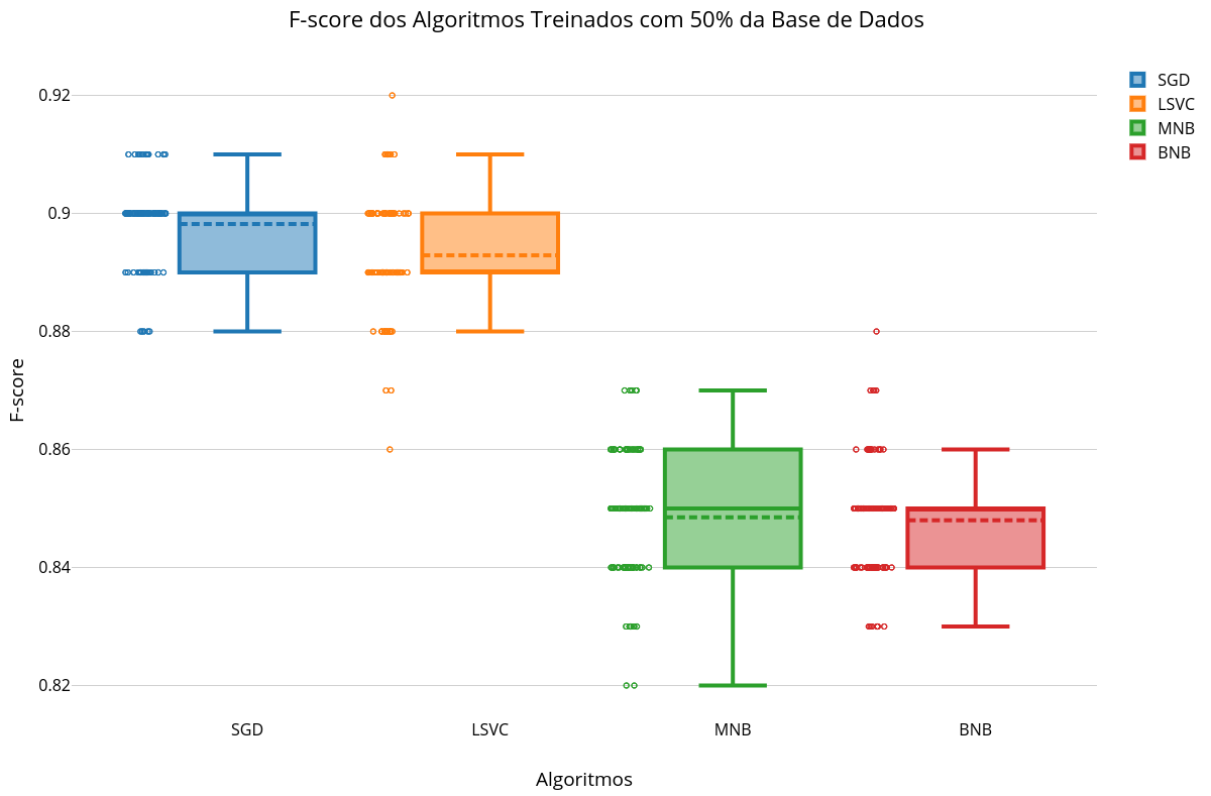
Os resultados dos métodos baseados em *Naive Bayes* obtiveram basicamente os mesmos resultados divididos em uma proporção muito semelhante, ambos retornaram praticamente o mesmo valor de média, cerca de 0,63, o mesmo pior valor, 0,55, desconsiderando-se o *outlier* do BNB e também tiveram uma diferença mínima no maior valor obtido, 0,73 para o BNB e 0,74 para o MNB.

Quanto aos resultados baseados em SVM podemos observar uma grande diferença, enquanto o modelo SGD se saiu melhor do que os modelos *Naive Bayes*, com uma média de 0,66, o método se LSVC se mostrou o mais ineficaz até agora, tanto na média dos resultados, 0,59, quanto em amplitude, sendo o único a possuir resultados inferiores a 0,5.

O segundo teste realizado seguiu os mesmo padrões que o teste anterior, sendo a única diferença a proporção de tamanho das bases de treino e teste, que nesta versão dividiu ambos na metade, ou seja, cerca de 2612 chamados na etapa de treino.

Esse teste foi realizado, novamente, para obter-se um comparativo entre os modelos utilizados e verificar se os padrões identificados no primeiro testes se acentuam ou se atenuam. É possível verificar no gráfico apresentado pela figura 4.2, que possui uma escala entre 0,82 e 0,92, uma grande diferença entre os métodos SVM e *Naive Bayes*.

Figura 4.2: F-score dos algoritmos treinados com 50% da base de dados



Nessa etapa já foi possível evidenciar com clareza a diferença entre os algoritmos baseados em SVM e *Naive Bayes*, se no teste anterior temos resultados muito parecidos, neste, temos os valores dos piores resultados dos modelos SVM obtendo um F-score semelhante aos melhores resultados dos modelos *Naive Bayes*.

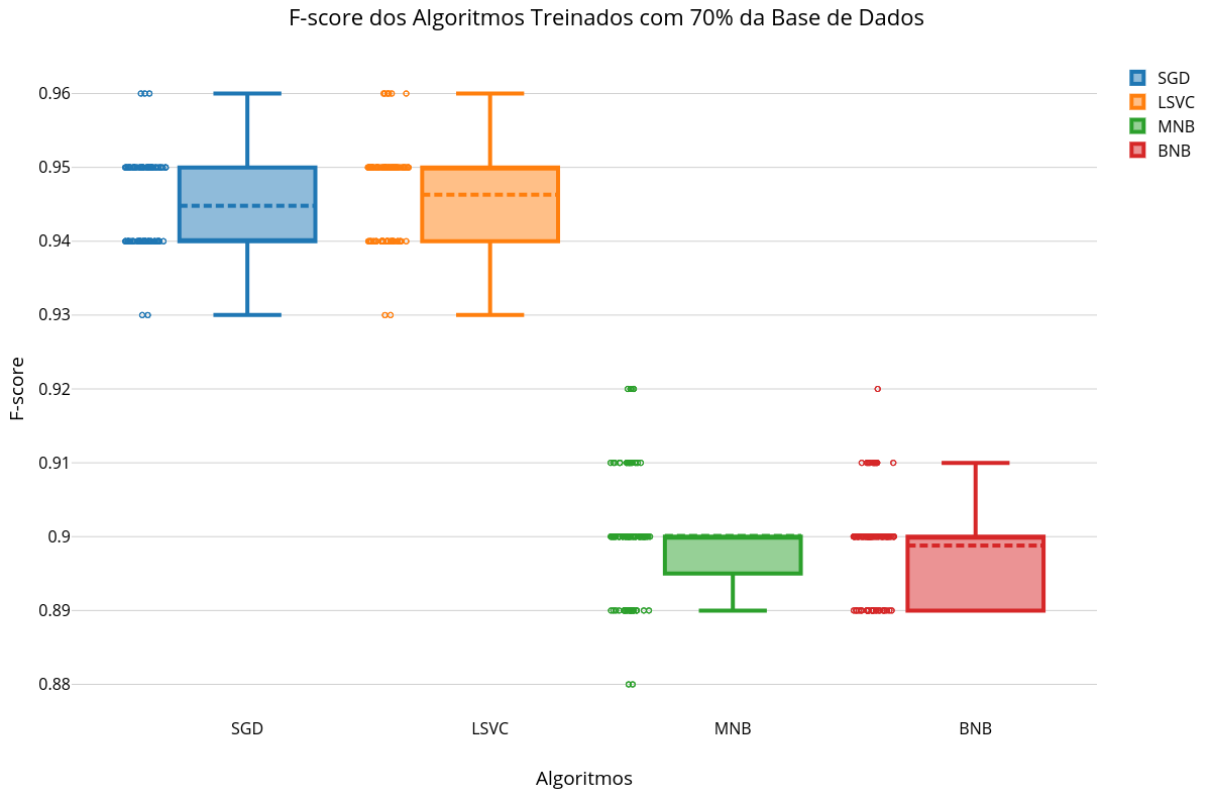
Considerando os algoritmos MNB e BNB notamos que ambos obtiveram novamente resultados parecidos, com uma média quase idêntica, no entanto, é possível verificar também,

que o modelo BNB se mostrou mais bem comportado, aparentando um desvio padrão semelhante aos do modelo SVM.

Nos métodos embasados em SVM temos uma surpresa, o algoritmo LSVC que retornou os piores resultados no primeiro teste obtém uma melhora incrível com o aumento da base de treinos, quase igualando-se ao SGD, obtendo uma média ligeiramente inferior e alguns *outliers* apenas.

O último teste realizado foi feito utilizando-se 70% da base de dados para a realização dos treinos dos algoritmos, cerca de 3656 chamados, neste teste final é possível visualizar de melhor forma o comportamento dos quatro modelos e identificar sua capacidade. A escala utilizada no gráfico apresentado pela figura 4.3 varia entre 0,88 e 0,96.

Figura 4.3: F-score dos algoritmos treinados com 70% da base de dados



Nos modelos *Naive Bayes* continuamos com valores muito próximos, uma média próxima a 0,9, um desvio padrão menor em relação aos modelos SVM, porém com resultados muito inferiores aos mesmo.

Em relação ao SGD e ao LSVC é possível dizer que ambos retornaram os mesmos resultados, valendo-se apenas do fator estatístico para justificar as discretas diferenças. Podemos observar também que, o modelo baseado em regressão necessidade de uma maior base de treinamento para atingir resultados semelhantes ao classificador linear.

Após as realizações dos testes e da obtenção dos resultados, foi evidenciado que, dada a aplicação proposta de categorização de chamados de suporte, os modelos baseados em SVM apresentaram resultados em média 3% superiores aos baseados em *Naive Bayes*, considerando-se uma base de treino com 70% da base de dados, essa diferença já havia sido demonstrada por Yiming Yang (1999).

Comparando os modelos *Naive Bayes* entrei si, não foi notada uma grande diferença, ambos possuem uma média de 90% de acerto, possivelmente seja possível visualizar um maior

desvio entre ambos com uma base de dados mais ampla, como demonstra Andrew McCallum (1998).

Dentre os métodos SGD e LSVC, que obtiveram os melhores resultados, o LSVC demonstrou necessitar de uma base de treinos maior para obter resultados equivalentes aos do SGD, com uma base de treino de 70% ambos obtiveram um *F-score* de 93%, no entanto, utilizando uma base de treino com apenas 30% da nossa base de dados, a diferença da média da pontuação de ambos é de 6%.

Dado isso, podemos definir que, dentre os algoritmos utilizados e com as configurações escolhidas, o modelo que obteve os melhores resultados foi o SGD, que utiliza um método de classificação baseado em SVM.

Capítulo 5

Conclusão

Este trabalho foi realizado utilizando-se uma coleção de 5223 chamados de suporte enviados à uma empresa de consultoria SAP, foram aplicados quatro algoritmos de classificação automática de texto, cada um com a representação textual que melhor se adequou a cada um deles a fim de obter-se uma resposta sobre qual metodologia seria a mais adequada para a proposta.

Os resultados demonstraram que os modelos baseados em SVM são superiores aos baseados em *Naive Bayes* para esta finalidade, obtendo uma eficácia superior em até 3% em média. Além disso, dentre o modelo com melhor resultado foi possível notar que a técnica de classificação (SGD) converge para melhores resultados mais rapidamente do que a técnica de regressão (LSVC).

Futuramente, novos testes podem ser realizados escalando-se a quantidade de dimensões do problema proposto, aumentando o número de classes para categorização. Outro ponto interessante a ser abordado é a inclusão de análise de documentos anexos, textos ou imagens, uma vez que em alguns casos é onde está contida toda a informação para a classificação e resolução do chamado. Dessa forma, é possível encarar um desafio mais próximo à realidade e obter maiores informações para a realização da classificação, aumentando a eficácia dos resultados.

Referências Bibliográficas

Andrew McCallum, K. N. (1998). A comparison of event models for naive bayes text classification. Em *AAAI-98 Workshop on Learning for Text Categorization*, Pittsburgh, PA.

Anne Kao, S. P. (2007). *Natural Language Processing and Text Mining*. Springer Science & Business Media.

Bo Pang, Lillian Lee, S. V. (2002). Thumbs up?: sentiment classification using machine learning techniques. Em *EMNLP '02 Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, páginas 79–86, Stroudsburg, PA.

Hovold, J. (2005). Naive bayes spam filtering using word-position-based attributes. Em *Proc. of the 2nd Conference on Email and Anti-Spam (CEAS 2005)*, Palo Alto, CA.

Joachins, T. (1998). Text categorization with support vector machines: Learning with many relevant features. Em *Nédellec C., Rouveirol C. (eds) Machine Learning*, Berlin, HD.

Joachins, T. (2002). *Support Vector Machines. In: Learning to Classify Text Using Support Vector Machines*, páginas 4–5. The Springer International Series in Engineering and Computer Science.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Mitchell, T. (1997). *Machine Learning*.

Murphy, K. (2006). Naive bayes classifiers. Em *Technical Report October 2006*.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47.

Yiming Yang, X. L. (1999). A re-examination of text categorization methods. *SIGIR '99 Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, páginas 42–49.