Universidade Federal de Pernambuco
Centro de informática

Graduação em ciência da computação

Análise e implantação de um ambiente para classificação de potenciais compradores de imóveis

Daniel Ferreira Maida

Trabalho de Graduação

Recife
28 de junho de 2018

Universidade Federal de Pernambuco
Centro de informática

Daniel Ferreira Maida

Análise e implantação de um ambiente para classificação de potenciais compradores de imóveis

Trabalho apresentado ao Programa de Graduação em ciência da computação do Centro de informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Luciano de Andrade Barbosa*

Recife
28 de junho de 2018

Para minha mãe, a cientista que eu mais admiro nesse mundo.

Agradecimentos

Agradeço à minha família por ter sempre me incentivado a seguir caminho da ciência e da curiosidade e por ter me apoiado nas decisões que tomei durante a vida. A minha mãe por ter me dado os livros de planetas e insetos que eu gostava tanto de ler na infância, também como as aventuras do Asterix e Obelix que me distraiam nas horas vagas. Ao meu pai por ter me introduzido à informática, ao uso de computador e a mexer com componentes eletrônicos que certamente me guiaram até o caminho que estou hoje.

Também agradeço ao meu orientador, Luciano, que sempre me apoiou e ensinou coisas que certamente eu não saberia hoje se não fosse pela paciência e pelo cuidado dele, digo com confiança que tive muita sorte de tê-lo como orientador. A Leonardo Andrade e Sandrelly Coutinho por terem me recebido na You Digital e disponibilizado o ambiente para que eu pudesse executar o estudo desse trabalho, assim como todos meus amigos da You Digital que passaram comigo bons momentos de diversão e de trabalho.

Aos meus amigos que sempre me apoiaram, me deram conselhos e que foram a minha luz em tempos escuros. Também gostaria de agradecer a Alê, que me ajudou a fazer as figuras desse trabalho enquanto eu estava preocupado em como ajeitar as tabelas e referências. Tudo que eu tenho hoje em dia, eu dedico e agradeço às pessoas que eu citei acima.

"We all make choices, but in the end our choices make us."
—ANDREW RYAN (Bioshock, 2007)

Resumo

Com a disseminação do uso da internet e sua adoção no dia a dia da população, as empresas tiveram que adotar o marketing digital como meio de impactar seus clientes em meio de tamanha concorrência. Para que as vendas aumentem e as companhias consigam competir pela atenção de seus clientes, métodos cada vez mais sofisticados de direcionamento de anúncios e classificação de possíveis compradores vem sido estudados e executados através de abordagens que tem como base o campo de aprendizagem de máquina e análise de dados. O setor imobiliário é um domínio que pode usufruir desses novos métodos, pois a concorrência é grande, os processos de venda são trabalhosos e demorados, além de o valor dos produtos ser alto e de um público específico. O objetivo deste trabalho é criar um ambiente propício para a execução de um sistema de classificação de possíveis compradores de imóveis, assim como sua implementação e análise de resultados.

Palavras-chave: ciência de dados, classificação, pré-processamento de dados, aprendizagem de máquina, setor imobiliário

Abstract

With the dissemination of the daily internet usage among the people, companies had to adopt the digital marketing as a tool to impact their clients to keep up with the increasing competition. To increase the sales and to compete for the attention of the clients, enterprises are using and studying sophisticated methods of focused advertisement and classification of potential buyers through approaches that rely on the machine learning and data analysis fields. The real estate industry is a domain that can take advantage of these new methods, because of the high competition in this field, the laborious and time-consuming process of property sale and the high value of the products, besides this the target audience of this domain is very specific. The goal of this project is to create a environment that is friendly to the execution of a system that classifies possible property buyers, as well as its implementation and result analysis.

Keywords: data science, machine learning, data pre-processing, real estate industry

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Com o aumento do fluxo de usuários na *world wide web*, também aumentaram o número de empresas que estão usando esse meio para divulgação e exposição dos seus produtos e serviços. Isso levou à uma grande competição por atenção dos usuários que navegam pela internet, pois quanto mais atenção dos clientes a respeito de um produto, maior vantagem ele tem dentre os outros. Várias plataformas de anúncio surgiram na última década com o intuito de potencializar a capacidade de venda de seus clientes, no entanto, atualmente, só o simples ato de anunciar em uma destas plataformas não é mais suficiente para aumentar as vendas de uma compania [17]. Tendo isso em vista, as técnicas de análise de comportamento de usuários estão ficando cada vez mais elaboradas para aumentar a taxa de conversão dos anúncios digitais [10].

A área de ciência dos dados vem ganhando bastante popularidade no meio do marketing direcionado pois engloba técnicas usadas para o estudo e a análise de comportamento de usuários, o uso de algoritmos de aprendizagem de máquina que classificam e dizem a probabilidade de um determinado usuário ser ou não um comprador em potencial e a possibilidade de aprender com o resultados de campanhas de marketing e melhorá-las através desse conhecimento. No setor imobiliário, a atenção dos compradores é um item essencial para o sucesso de uma empresa, pois a concorrência é muito grande, os produtos possuem um valor alto e os compradores são muito específicos. Corretores de imóvel possuem uma lista extensa de clientes para entrar em contato e a taxa de sucesso de conversão destes contatos em vendas é muito baixa para uma grande quantidade de tempo e esforço gastos. O objetivo deste trabalho é criar, através do estudo e implementação de uma situação real de mercado, um ambiente propício para a implantação de um sistema de classificação de usuários com base em seu comportamento de navegação coletado através de anúncios imobiliários, assim como a análise dos processos e métodos necessários para que este sistema seja implementado.

Os principais desafios deste trabalho são: executar todas as etapas do complexo trabalho de um cientista de dados e lidar com as características peculiares que fazem parte da predição de possíveis clientes com base em seu comportamento [20]. Este *pipeline* possui quatro grandes etapas: Preparação do ambiente para que seja possível a execução da tarefa, análise dos conjunto de dados em questão, o pré-processamento de dados para que a tarefa de classificação seja possível e a execução de algoritmos de classificação de usuários em clientes em potencial própriamente dita.

CAPÍTULO 2

Fundamentos

A preparação de um ambiente propício para a realização de tarefas de classificação é um processo trabalhoso e que tem que ser feito com muito cuidado para que a qualidade do modelo gerado pelos algoritmos seja a melhor possível. Para que isso seja feito de maneira correta, uma série de fatores tem que ser levados em conta. Neste capítulo (2) iremos introduzir alguns fundamentos que são bastante úteis para a preparação deste ambiente e também para a construção dos modelos de classificação.

2.1 Pré-processamento de dados

Quando se deseja utilizar uma base de dados para a construção de algum modelo estatístico, alimentação de algoritmos de aprendizagem supervisionada ou até para realizar uma análise de dados, é necessário que haja um tratamento prévio desses dados, pois eles podem conter características que impactam negativamente na realização destas tarefas. Estes dados inválidos ou ruidosos podem ter sido gerados através de ruídos em sensores, perturbações no processo de coleta de dados, degradação de instrumentos e também por erro humano [13]. O pré-processamento de dados é uma etapa essencial para a construção do modelo de aprendizagem e sem ela a precisão e qualidade do modelo final pode ser bastante prejudicada por dados faltantes, outliers e duplicação de dados. Além de ser muito importante, esta fase é a que leva mais tempo para ser concluída no trabalho de um cientista de dados [24]. Sabendo disso, podemos afirmar que o pré-processamento de dados, apesar de não ser a tarefa mais agradável de ser feita, é a pedra angular do processo de construção de modelos de aprendizagem e de análise de dados. Nesta seção (2.1) iremos introduzir alguns problemas encontrados na utilização de bases de dados e os principais métodos usados para solucioná-los.

2.1.1 Detecção e limpeza de outliers

Outliers são observações no conjunto de dados que estão à uma distância anormal dos outros valores presentes, ou seja, são dados que forgem da normalidade ou do padrão do conjunto [14]. A classificação de uma observação como anormal ou não é geralmente feita por quem está analisando os dados. Os *outliers* podem ser gerados por erros na coleta dos dados e interferem drasticamente no resultado da construção de modelos estatísticos e, consequentemente, na análise dos dados em questão. Enviesamentos e problemas na acurácia de modelos de classificação são o principais problemas causados pela presença de *outliers* no conjunto de dados. Estes dados anormais podem ser úteis para a análise dos dados quando se deseja detectar ou

estudar casos que se distanciam do padrão no conjunto de dados, ou seja, a presença de *outliers* no conjunto de dados nem sempre é um problema. Nesta seção (2.1.1) iremos apresentar algumas possíveis soluções para a remoção e tratamento de outliers.

2.1.1.1 Análise de dados brutos

A detecção de outliers pode ser feita através de uma análise dos dados brutos, ou seja, através de uma simples visualização dos dados pertencentes à base. Dados muito discrepantes da maioria ou com valores inválidos como, por exemplo, uma idade igual a 400 podem ser facilmente identificados por meio de uma análise manual. O grande problema deste método é que ele pode ser muito trabalhoso em conjuntos que possuam uma grande quantidade de dados.

2.1.1.2 Análise de gráficos

Outra maneira de se detectar outliers é por meio de gráficos gerados a partir dos dados da base, assim pode-se visualizar os pontos que desviam da normalidade mais facilmente do que ao analisar os dados brutos. *Boxplots*, histogramas e graficos de dispersão são alguns dos possíveis gráficos que podem ser usados para se detectar *outliers*.

2.1.1.3 *Z-Score* modificado

O *Z-Score* ou *standard score* representa quantas vezes em relação ao desvio padrão que um ponto está distante da média de um determinado conjunto de dados. O *Z-Score* pode possuir valores positivos e negativos e o seu sinal representa se o ponto está acima ou abaixo do valor da média. O *Z-Score* é comumente usado para a detecção de *outliers*, onde se estabelece um valor arbitrário e pontos que possuem um score que ultrapassa esse valor são eliminados do conjunto. O valor do *Z-score* pode ser obtido através da fórmula abaixo:

$$Z_i = \frac{Y_i - \bar{Y}}{s} \tag{2.1}$$

Onde o $\bar{Y}$ e s representam respectivamente a média e o desvio padrão do conjunto. Apesar do poder deste método, por seu valor ser ao máximo $[(n-1)/\sqrt{n}]$, ele pode levar a conclusões enganosas, principalmente em conjuntos com poucos dados. Para contornar este problema, foi sugerido por Iglewicz e Hoaglin [16] o uso do método de *Z-Score* modificado que pode ser representado pela seguinte equação:

$$M_i = \frac{0.6745(x_i - \bar{x})}{MAD} \tag{2.2}$$

Nela o *MAD* corresponde à *Median absolute deviation* [23] e $\bar{x}$ é a mediana do conjunto. O valor do *modified Z-Score* consegue levar a melhores resultados para a eliminação de outliers quando comparado ao método tradicional [16]. Apesar de ambos serem bons métodos para eliminação de *outliers*, eles funcionam melhor quando aplicados a conjuntos de dados univariados e que seguem uma distribuição aproximadamente normal. Existem diversos métodos de detecção de *outliers* para estas outras condições, mas não iremos falar sobre eles neste trabalho.

2.1.2 Lacunas nos dados

Outro problema encontrado quando se deseja fazer um modelo estatístico ou construir um conjunto de treinamento para aprendizagem de máquina é que na maioria das vezes em conjuntos de dados reais há muitos dados faltantes. Estas lacunas de dados pode ser causada por diversos motivos: erro humano, falha em sensores, corrompimento de dados, entre outros. Essa falta de dados afeta o modelo a ser gerado de forma negativa [26], pois para que a geração do modelo não seja prejudicada é necessário que todas as features estejam presente no seu treinamento. Há diversas técnicas para lidar com esses casos, desde o simples preenchimento de dados vazios com valores que não interferem no resultado do modelo quanto técnicas mais elaboradas para recuperação desses dados com base nos valores de outras *features*. Essas e outras técnicas serão abordadas nessa subseção.

2.1.2.1 Preenchimento *naive* de lacunas de dados

Muitas vezes, a solução para lacunas nos dados pode ser bem simples e feita de forma que não afeta o modelo gerado. Valores faltantes nos dados geralmente são representados de duas formas por convenção: usando uma máscara que identifica globalmente os valores vazios ou escolhendo um valor pré-determinado que representa uma entrada nula. Na primeira abordagem, um *array* de booleanos pode ser usado para representar as lacunas nos dados ou até mesmo a apropriação de um bit na representação do dado para indicar localmente que o seu valor é nulo. Já na segunda técnica, o valor do sentinela pode ser um valor escolhido arbitrariamente de acordo com o tipo do dado faltante e usado como convenção para representar valores vazios, como por exemplo usar um valor negativo para representar um valor nulo de um inteiro. Essa técnica é simples e não interfere na construção de diversos modelos, porém existem técnicas mais sofisticadas que podem até recuperar dados vazios baseados em outras características daquele conjunto de dados [26].

2.1.2.2 Recuperação de dados faltantes

Diferentemente do preenchimento simples de dados, a recuperação pode ser feita através de técnicas mais elaboradas e que levam em consideração outras entradas do conjunto de dados que estão completas para a realização do preenchimento. Algumas técnicas populares para recuperação de dados se baseiam no uso de abordagens estatísticas, como a aplicação da moda, mediana e média dos valores globais ou específicos de um determinado grupo de dados [27]. Na tabela 2.1, que possui dados-exemplo do setor automobilístico, temos uma pequena parte de um conjunto de dados para representar o caso de dados faltantes.

Neste caso podemos usar a média para recuperar o dado que está faltando já que não há *outliers* nesse pequeno conjunto de dados, caso houvesse a mediana poderia ser uma alternativa interessante. Apesar de usarmos a média, não é adequado escolher a média global para preencher o valor nulo pois os valores dentre os tipos de carro são consideravelmente discrepantes, então podemos usar a média dos valores da classe para obter um resultado mais relevante. Aplicando esta técnica temos o resultado presente na tabela 2.2

Esta técnica é simples e bem efetiva para determinados casos, mas como qualquer outra ela depende de uma análise prévia do conjunto de dados antes de ser aplicada para que pro-

Tipo do carro	Autonomia (km/l)	Peso(kg)
SUV	7	1.393
Sedan	10	1.237
Sedan	-	1.123
SUV	6.76	1.430
Sedan	11	1.133

Tabela 2.1 Exemplo de conjunto de dados com lacunas

Tipo do carro	Autonomia (km/l)	Peso(kg)
SUV	7	1.393
Sedan	10	1.237
Sedan	10.5	1.123
SUV	6.76	1.430
Sedan	11	1.133

Tabela 2.2 Conjunto de dados após o preenchimento das lacunas

duza bons resultados. Existem diversas técnicas para recuperação dos dados, porém as vezes a melhor opção pode ser deletar as linhas que possuem lacunas nos dados para que não haja intereferência na construção do modelo de dados.

2.1.3 Normalização e correção de dados

Conjuntos de dados reais possuem muitos dados duplicados ou ruidosos, necessitando que eles passem por uma espécie de tratamento. Esses erros no conjunto de dados muitas vezes ocorrem por falhas na sua captação, no entanto, em alguns casos a solução para isso pode ser bem simples. A tabela 2.3 representa um exemplo de duplicação de elementos em um conjunto de dados.

Email	Num. de ligações	Num. de mensagens	Num. de videochamadas
allanpoe@raven.com	6	15	2
lovecraft@mythos.com	10	5	1
Allanpoe@raven.com	10	5	1

Tabela 2.3 Exemplo de um conjunto com dados duplicados

Podemos ver que as duas linhas da tabela representam a mesma pessoa, só que no momento de captação elas foram tratadas como duas entradas diferentes por não ignorar a caixa do texto, causando um erro de consistência nos dados. Apesar de atrapalhar a geração do modelo de aprendizagem, este é um problema fácil de ser resolvido com uma simples ordenação dos dados, conversão de todos os valores de email para letras minúsculas e depois a junção das linhas que possuem o mesmo email, produzindo o resultado exibido na tabela 2.4.

Este e outros casos podem ser resolvidos trivialmente, porém existem outros casos que

Email	Num. de ligações	Num. de mensagens	Num. de videochamadas
allanpoe@raven.com	16	20	3
lovecraft@mythos.com	10	5	1

Tabela 2.4 Conjunto após a correção de duplicação de dados

não podem ser resolvidos com um simples agregamento de valores. Por exemplo, duas linhas que representam uma média de um valor não podem ser simplesmente agregadas. Como foi dito na seção de recuperação de dados, quando os dados forem simplesmente ruídos que não contribuem para o modelo, a melhor opção pode ser apenas deletá-los. Nesta seção vimos como o pré-processamento de dados é importante para o processo de construção de um modelo e alguns dos principais métodos de pré-processamento assim como o seu fluxo. Na próxima seção iremos falar sobre aprendizado de máquina supervisionado e algoritmos de classificação.

2.2 Aprendizado supervisionado

Segundo Mitchell: "Aprendizado é um fenômeno de multifacetado. O processo de aprendizado inclui a arquisição de novos conhecimentos declarativos, o desenvolvimento de habilidades motoras e cognitivas através de instrução ou prática, a organização do novo conhecimento em representações gerais e efetivas e a descoberta de novos fatos e teorias através de observação e experimentação"[22]. O autor apresenta uma definição clara e concisa sobre o processo de aprendizado, mais especificamente sobre o supervisionado quando ele aponta o desenvolvimento de habilidades e desenvolvimento do conhecimento através de experiência e observação. A aquisição de conhecimento através de exemplos e observação é a principal característica do aprendizado supervisionado, nele um conjunto de treinamento e de teste é dado ao observador (normalmente um programa de computador) para que ele aprenda através de exemplos rotulados a predizer a classe de um exemplo dado. Os modelos de classificação são construidos a partir do conjunto de treinamento onde cada instância possui diversas características ou *features* e um valor que representa a classe que ela se encaixa (rótulo). Essas instâncias que servem como entradas para o treinamento do modelo são chamados de dados rotulados. A partir do momento que o modelo é treinado e uma função de indução para a predição de um elemento é gerada, ele é validado através do conjunto de testes [18]. Quando o modelo tenta prever um valor numérico, isso se trata de um problema de regressão, já quando o ele tenta prever um valor categórico, um problema de classificação. No processo de validação, o modelo tenta prever a classe das entradas do conjunto de testes e compara o resultado com o valor real que está presente no rótulo dos dados. Com os resultados do teste, é possível avaliar o modelo construído através de diversas métricas que apontam a qualidade do resultado do treinamento deste modelo e comparar com outros modelos para que se tenha conhecimento de qual é mais adequado para o conjunto de dados em questão [12]. Nesta seção iremos focar em problemas de classificação e apresentaremos algumas métricas para avaliá-los, também faremos um breve resumo sobre alguns algoritmos de aprendizado supervisionado que foram usados neste trabalho.

2.2.1 Métricas de avaliação de um modelo de classificação

Métricas são uma peça chave para a escolha de um modelo adequado para determinado conjunto de dados [11], com o seu uso é possível saber o desempenho de um algoritmo de classificação e compará-lo com outros para a escolha do mais eficiente. Nesta sub-seção iremos apresentar algumas das principais métricas usadas para a avaliação de modelos. Nas equações abaixo, TP são positivos verdadeiros (*true positives*), FP são falsos positivos (*false positives*), FN são falsos negativos (*false negatives*) e TN são negativos verdadeiros (*True Negatives*).

2.2.1.1 *Precision*

Precision ou precisão é a métrica que mede a proporção de casos que foram indicados como positivos pelo classificador e que realmente estão corretos. O cálculo da precisão pode ser representado pela seguinte fórmula:

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

2.2.1.2 *Recall*

Recall ou revocação mede a proporção de casos positivos que foram apontados corretamente pelo classificador em relação ao conjunto total de positivos. O cálculo da revocação pode ser representado pela seguinte fórmula:

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

Normalmente os valores de precisão e revocação de um modelo se comportam de maneira inversamente proporcional, ou seja, quando a precisão aumenta a revocação diminui e vice-versa. Por conta deste comportamento, quando se vai avaliar a qualidade de um classificador, os valores tanto de precisão quanto de revocação tem que ser observados para que o resultado da análise seja relevante.

2.2.1.3 *F-measure*

F-measure ou Teste-F é uma métrica que calcula o *score* de um modelo estatístico baseado nos seus valores de precisão e revocação. Em outras palavras, o valor do *F-measure* corresponde à média harmônica dos valores de precisão e revocação, onde 0 é o pior valor e 1 é o melhor valor que pode ser assumido. O cálculo desta métrica pode ser feito usando a seguinte fórmula:

$$F - measure = 2 * \frac{precision * recall}{precision + recall} \tag{2.5}$$

2.2.1.4 *Accuracy*

Accuracy ou acurácia é a medida de quantos casos foram previstos corretamente por um modelo estatístico. É uma das métricas mais simples e mais usadas para avaliar a qualidade de um modelo. O cálculo da acurácia pode ser representado pela seguinte fórmula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

2.2.2 Algoritmos de aprendizado supervisionado

Nesta sub-seção iremos falar brevemente sobre alguns dos principais algoritmos de aprendizado supervisionado, todos estes algoritmos foram utilizados e comparados [12] neste trabalho e iremos falar sobre os resultados de seus treinamentos no capítulo 4.

2.2.2.1 *Naive* Bayes

Este algoritmo de aprendizado supervisionado é baseado no teorema de bayes e se encaixa na classe dos algoritmos probabilísticos. O teorema de bayes calcula a probabilidade de um evento a partir de um conhecimento a priori de fatores que levam o evento a ocorrer. O teorema de bayes pode ser representado pela seguinte fórmula:

$$P(A|B) = \frac{P(B|A) * (P|A)}{P(B)} \tag{2.7}$$

Nesta equação, $P(A)$ representa a probabilidade a priori de A ocorrer, $P(B|A)$ a probabilidade de B ocorrer dado que A é verdade e $P(B)$ a probabilidade a priori de que o próprio evento em questão ocorra. Este método de aprendizado é chamado de *naive* pois assume "ingênuamente"que as *features* dos dados são independentes entre si dada a classe. Este algoritmo de classificação possui um pequeno tempo de treinamento, é simples de ser implementado e funciona bem com conjuntos de dados multidimensionais. Apesar destas vantagens, por assumir a independência entre as *features*, este método não funciona bem para casos onde elas não são independentes.

2.2.2.2 Regressão logística

Como o método de *Naive Bayes*, a regressão logística é um método que tem suas origens na estatística e foi agregado ao campo de aprendizado de máquina. Ele tem como base o uso da função logística, que foi usada inicialmente para prever o crescimento de populações no estudo da ecologia. Essa é uma função sigmóide e recebe como entrada um valor que é mapeado dentro de um intervalo entre 0 e 1. Esse mapeamento é utilizado para prever a probabilidade de uma determinada entrada ser de uma classe ou não, essa probabilidade pode ser transformada em uma classificação binária e por isso pode ser usada para resolver problemas de classificação. O método de regressão logística possui robustez quanto à presença de *outliers* e baixa variância e, por este último motivo, pode sofrer de um alto viés se não usado com cuidado.

2.2.2.3 Árvore de decisão

O algoritmo de árvore de decisão ou *decision tree* é um método de classificação bastante utilizado no campo de aprendizagem de máquina. Ele possui esse nome pois sua estrutura de decisão se assemelha com a de uma árvore, onde os ramos são os caminhos que podem ser to-

mados e as folhas são as decisões finais sobre qual classe uma determinada entrada se encaixa. O algoritmo de construção da árvore usa a entropia do conjunto para saber a sua homogeneidade e leva esse valor em conta para fazer a sua divisão em sub-conjuntos. Essa divisão gera os caminhos de decisão da árvore que serão usados para fazer a classificação de entradas futuras, esses caminhos são escolhidos de acordo com o valor das *features* da entrada que se deseja classificar. Uma grande vantagem deste modelo é sua fácil visualização por ter essa estrutura de indução baseada em uma árvore, além de lidar bem com dados qualitativos e com conjuntos de treinamento que possuam poucas *features*.

2.2.2.4 *Support Vector Machine*

Support Vector Machine ou SVM é um algoritmo de classificação que se destaca dos outros por possuir uma abordagem diferente quando se trata do seu modo de geração do modelo. A construção do classificador SVM é baseada na divisão do espaço de dados por vetores de suporte, a diferença deste algoritmo está na forma de como esta divisão é feita. Enquanto outros algoritmos utilizam a similaridade entre pontos da mesma classe para a construção de seu modelo de classificação, o SVM usa os pontos das classe de dados que são mais diferentes da maioria como guia para a construção de seu modelo de classificação. Esses vetores dividem o espaço do conjunto de dados e a classificação de novas entradas de dados será feita com base na posição em que elas se encontram no espaço em relação aos vetores de suporte. De uma forma mais clara, os vetores de suporte são usados para dividir o espaço de dados em regiões que correspondem as classes do modelo, por este motivo o SVM é um ótimo modelo para classificar conjuntos de dados que possuem *features* esparsas e em grande quantidade, como é o caso da classificação de textos.

2.2.2.5 *Random forest*

Esse algoritmo faz parte de uma classe de algoritmos de aprendizagem de máquina chamados de *ensemble learning*, que pode ser traduzido livremente como aprendizado de conjunto, justamente por combinar algoritmos de aprendizado de máquina para produzir um resultado final de classificação. O *Random forest* se faz do uso de um número, escolhido pelo desenvolvedor, de instâncias de algoritmos de árvores de decisão aplicados a sub-conjuntos aleatórios do conjunto de dados. Esses sub-conjuntos servem como conjunto de treinamento das instâncias das árvores de decisão são treinadas, após esta etapa, quando surgem novas entradas para serem classificadas, elas são fornecidas para todas as instâncias do algoritmo e seu resultado é escolhido através do voto majoritário realizado entre elas. O *Random Forest* possui uma variância menor em relação as *Decision Trees* porém ele não é tão fácil de ser interpretado visualmente.

2.3 Trabalhos relacionados

Nesta seção iremos falar brevemente sobre alguns trabalhos relacionados a este, tanto da indústria quanto da academia. A idéia desta seção é explicar soluções atuais que existem no mercado e que de certa forma serviram de motivação para a execução deste estudo. Quando se trata de

geração de *leads*, grande parte das empresas trata esse assunto como apenas a captação das informações de contato de um usuário de um site, ou seja, uma grande quantidade de usuários que se identificaram em uma determinada plataforma. A definição de *lead* apenas como um contato já é ultrapassada, uma lista de *leads* é muito mais que apenas uma lista de e-mails aleatórios que os encarregados de vendas deve entrar em contato [21]. Com a grande quantidade de geração de dados comportamentais a todo instante nas plataformas, é necessário uma análise, seleção e ranqueamento destes comportamentos para se chegar a conclusão de quais usuários possuem um comportamento que pode ser caracterizado como um comportamento de um cliente em potencial. Em seguida, iremos expor algumas empresas e estudos que seguem essa linha de pensamento.

2.3.1 InGaia Ads

A InGaia é uma empresa do ramo imobiliário que possui um CRM (sistema de gestão de relacionamento com o cliente) amplamente usado em território nacional e recentemente realizou uma parceria com a WebCompany e com o Google para lançar uma plataforma de geração de *leads* e campanhas de marketing inteligente chamada InGaia Ads [2]. O foco dessa plataforma é na administração e geração de campanhas inteligentes no Google Adwords. Os divulgadores dessa plataforma também falam de geração de *leads* qualificados, mas não detalham o processo de qualificação de *leads* e nem falam se eles são gerados através de plataformas inteligentes de aprendizagem de máquina.

2.3.2 Smart Leads

A Smart Leads é um produto de uma empresa americana chamada *The AdTrack Corporation* que tem seu foco na geração e gerenciamento de *leads*, assim como criação de campanhas [9]. Eles possuem um serviço genérico de gestão de *leads* que não depende de um domínio específico. Eles possuem serviços chamados *lead scoring* onde o usuário escolhe as features que possuem mais relevância na hora do ranqueamento dos *leads* e também *lead qualification* que é uma ferramenta de qualificação de *leads* baseado no *lead scoring*. Eles também promovem a nutrição desses *leads* através de sua ferramenta de gerenciamento, o que é algo interessante e que aumenta a probabilidade de fechamento de negócio.

2.3.3 *Data Mining for Direct Marketing: Problems and Solutions*

Este trabalho descreve os problemas e soluções do processo de mineração de dados para marketing direcionado. Os autores discutem o processo de identificação de usuários em potencial para um determinado produto e da tarefa de impactação destes usuários através de anúncios [20]. Os autores também discutem os desafios econtrados nesse processo, o problema da distribuição desbalanceada das classes (geralmente os exemplos positivos são bem mais escassos que os negativos) o que leva a diminuir a confiança na métrica de acurácia, pois exemplos falsos negativos se fazem muito mais presentes por este motivo (classificar não-compradores como possíveis compradores). O trabalho também sugere o uso da métrica de *lift* [15] ao invés da acurácia como critério de avaliação do modelo.

CAPÍTULO 3

Solução

Como foi dito no capítulo introdutório, o objetivo deste trabalho é a construção de um ambiente propício para a geração de *leads* qualificadas através de algoritmos de aprendizagem de máquina. Como caso de estudo, foi escolhido o a ambiente da empresa You Digital, que é proprietária de dois grandes sistemas do ramo imobiliário: o Expoimóvel [1], um dois maiores portais imobiliários do estado de Pernambuco e o Smart Imobiliário [5], um sistema de *CRM* (*Customer relationship Manager*) usado por grande parte das maiores imobiliárias do estado, além de diversos sites de imobiliárias coordenados por ela. Este capítulo irá relatar a preparação do ambiente para que fosse possível a aplicação do estudo em questão, assim como as peculiaridades e processos usados no caminho para a sua construção. A figura 3.1 mostra o *pipeline* de execução do estudo desse trabalho.

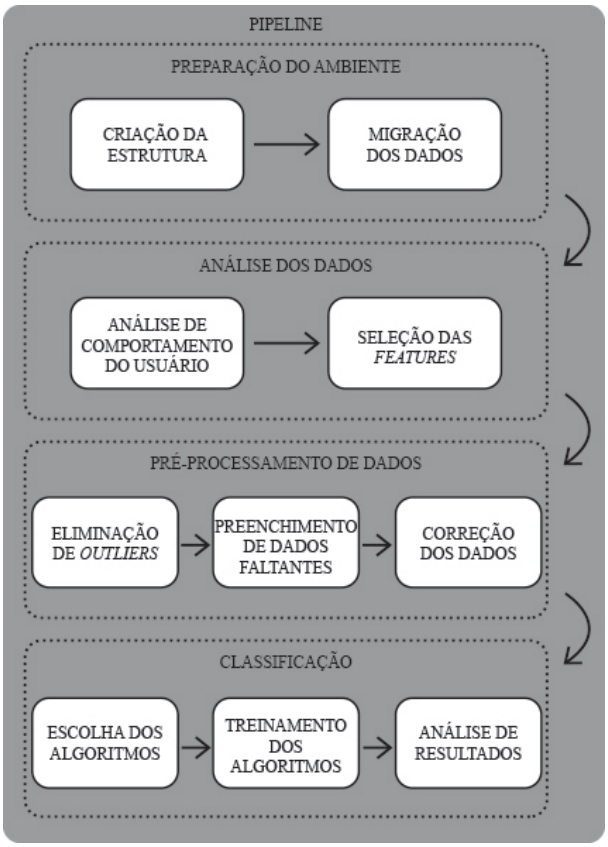

Figura 3.1 *Pipeline* do estudo

3.1 Preparação do ambiente

Após a análise da estrutura de comunicação entre as aplicações da empresa, do sistema de banco de dados e do funcionamento de certos processos dentro dela, foi concluído que o ambiente não era propício para a implementação do sistema de qualificação de leads. A empresa já guardava algumas informações de navegação dos usuários antes do início do estudo e a figura 3.2 ilustra o sistema de fluxo de dados dela:

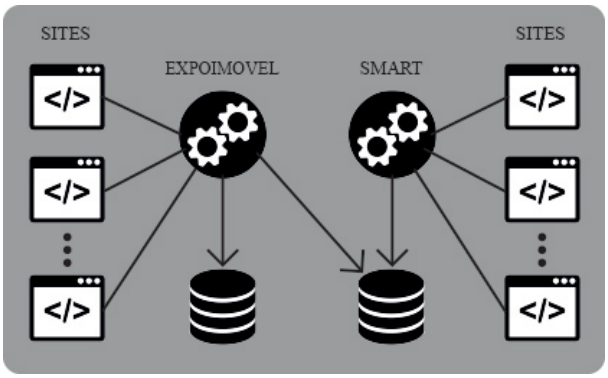

Figura 3.2 Estrutura inicial do ecossistema da You Digital

Na figura 3.2, podemos ver que os dados de navegação das duas plataformas eram salvos em seus respectivos bancos de dados e, além disso, os sites das imobiliárias também nutriam os bancos das duas plataformas. O que se nota de diferente é o fluxo de dados saindo do Expoimóvel e alimentando o banco do Smart. Este fluxo tinha a ideia inicial de prover uma certa centralização dos dados e desafogamento do banco de dados do Expoimóvel (que possui um fluxo bem maior do que o do CRM). Esta medida levou a uma grande duplicação de dados, pois os sites das imobiliárias e alguns dados de navegação do portal imobiliário estavam sendo salvos tanto no seu banco quanto no banco do CRM. Outro fator preocupante e que impedia uma análise, tratamento e classificação dos dados, era a estrutura extremamente discrepante dos bancos, onde os seus elementos em comum possuiam muitas vezes até a tipagem de dados diferente, além de erros estruturais que também dificultavam este processo. Tendo esses fatores em vista, foi decidido que a melhor opção, tanto para aliviar o banco de dados das duas aplicações quanto para criar um ecossistema sustentável para a implantação do sistema de qualificação de leads era criar um novo banco de dados que comportaria as informações necessárias de forma clara, genérica e concisa, assim como uma API que serviria de interface de comunicação para os dois sistemas. De acordo com essas conclusões, foi implementada a estrutura ilustrada na figura 3.3.

Para que a esturura do novo banco de dados ficasse genérica o suficiente para comportar os dois sistemas, foi necessário um estudo profundo do domínio e das regras de negócio, o que levou uma quantia de tempo considerável. Após a criação desta estrutura, foi realizada uma importação dos dados antigos que serviriam como entrada para os algoritmos de classificação que serão abordados futuramente neste trabalho. Esta importação também necessitou de uma grande quantia de esforço e de tempo, pois como foi dito anteriormente, as estruturas dos

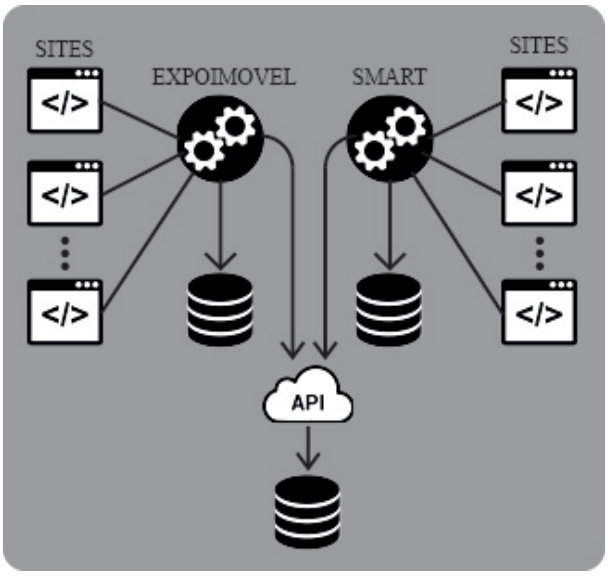

Figura 3.3 Estrutura modificada para implantação do sistema de qualificação de leads

bancos possuiam diversos erros e inconsistências de tipo que tiveram que tratadas para que os dados pudessem ser importados para o novo banco. Após esta etapa, foi consolidada uma estrutura sólida para a geração de leads no mercado imobiliário que será discutida na seção seguinte.

3.2 Análise do fluxo de comportamento do usuário

O processo de procura e compra de um imóvel é um processo bem definido e que serviu de guia para a construção da nova estrutura e também para a seleção de features do conjunto de dados para o treinamento dos algoritmos de aprendizagem de máquina. O processo (digital) pode ser divido em duas etapas, a etapa do portal imobiliário e a etapa do CRM. Primeiramente, quando um usuário entra em um portal imobiliário ele realiza uma busca com alguns filtros que correspondem as características de um imóvel alvo, logo em seguida ele clica nos imóveis que lhe chamaram atenção, gerando comportamento de visualização. Caso ele esteja realmente interessado ele pode entrar em contato com o responsável pela venda do imóvel, gerando um comportamento de contato. Normalmente esta é a etapa em que o usuário se identifica na plataforma. A partir deste ponto, a coleta de dados não é feita mais pelo portal imobiliário, pois ele só administra o processo de compra de um imóvel até a realização do contato do cliente com o vendedor, os passos seguintes são feitos através do CRM. Geralmente quando um cliente entra em contato com um corretor ou com uma imobiliária este contato é salvo pelo corretor responsável em uma plataforma CRM, e todos seus acompanhamentos a partir deste ponto (agendamento de visita, visita realizada, proposta de negócio e fechamento de negócio) são cadastrados no sistema por seres humanos responsáveis pelos acompanhamentos do clientes. Na figura 3.4 podemos ver este fluxo que foi descrito com mais facilidade.

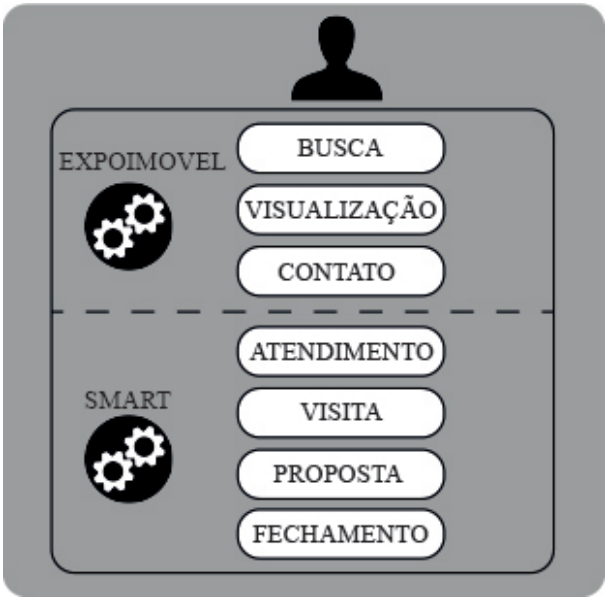

Figura 3.4 Fluxo da compra de um imóvel por um usuário interessado

3.3 Análise e seleção das *features* para treinamento dos modelos

De acordo com o fluxo de comportamento dos usuários, análise dos dados e estudo sobre o domínio, foram selecionadas algumas *features* para compor o conjunto de dados que no futuro serviu como conjunto de treinamento e teste. Todas estas *features* foram agrupadas por usuário, são elas:

- Quantidade de visualizações do usuário
- Quantidade de buscas do usuário
- Mediana do intervalo entre visualizações
- Mediana do intevalo entre buscas
- Mediana de preços de imóveis de interesse
- Mediana do número de quartos do imóvel
- Mediana da área do imóvel de interesse
- Quantidade de contatos feitos pelo usuário

A partir da análise e escolha dessas features foi possível ter conhecimento das respostas que poderiam ser obtidas a partir delas, por questão de quantidade de dados, foi decido que a pergunta que iríamos responder para a qualificação de *leads* é a seguinte:

"Dado um usuário que navega no portal imobiliário, qual é a probabilidade dele entrar em contato a respeito de um imóvel?"

Foi decidido descer somente até o nível de contato pois a quantidade de dados após este passo não era suficiente para construir um modelo minimamente robusto. Isto é justificavel pois, como o e-mail do usuário foi escolhido como sua *footprint* digital e muitos usuários de CRM sequer cadastram o e-mail dos clientes em acompanhamento na plataforma, a ligação entre o contato realizado no portal imobiliário e os próximos passos no sistema CRM muitas vezes não é possível. Portanto, se o usuário entrou em contato, ele é um exemplo rotulado positivamente no conjunto de dados, caso contrário ele é rotulado negativamente no conjunto de treinamento. Na figura 3.5, podemos ver a distribuição de frequência de uma das features principais usadas para a classificação de usuários que possivelmente irão entrar em contato.

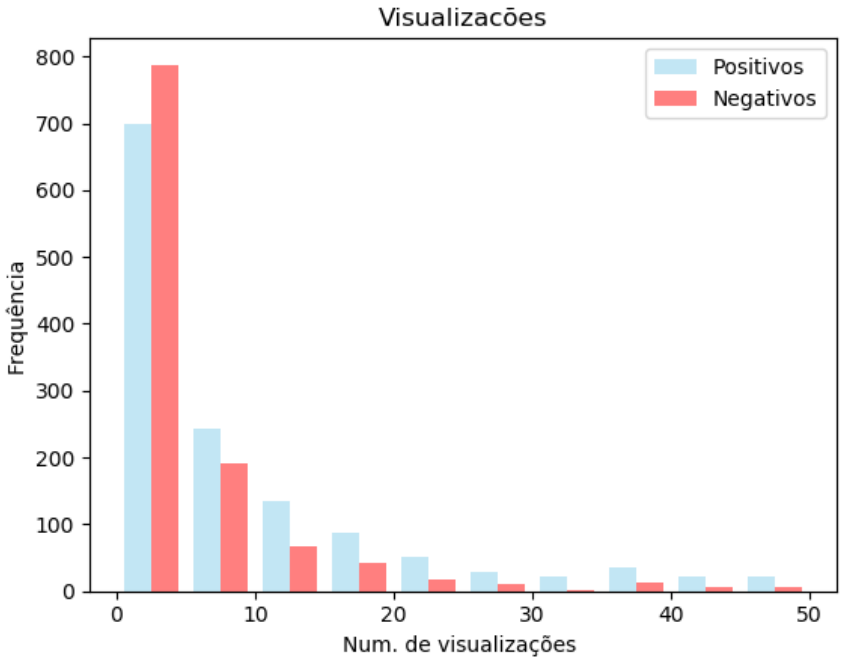

Figura 3.5 Distribuição de quantidade de visualizações por usuário (sem *outliers*)

3.4 Pré-processamento dos dados

Mesmo fazendo uma estrutura organizada e genérica o suficiente para comportar os dois sistemas, a desorganização inicial dos dados impactou na sua importação. Alguns dados duplicados vieram nesse processo, assim como dados que possuíam lacunas, ruídos e outros impedimentos que poderiam vir a prejudicar a construção do modelo aprendizado. Nesta seção iremos discutir alguns ajustes que foram necessários para assegurar a qualidade do conjunto de dados.

3.4.1 Detecção e limpeza de *outliers*

Para a detecção de outliers neste trabalho, foram plotados gráficos do tipo *boxplot* e histograma (Figura 3.5) para que a distribuição dos dados fosse analisada e os casos discrepantes fossem eliminados. Inicialmente, tentamos utilizar o método do *Z-Score* modificado (ver seção 2.1.1.3) para a detecção do outliers, mas como a distribuição dos dados não é normal este método não foi efetivo. O conjunto de dados apresentou uma distribuição de *heavy-tail* ou cauda longa, possuindo muitos pontos com valores próximos a zero. Os *outliers* eram em sua maioria causados por *bots* que ficavam navegando no site e gerando dados comportamentais e por corretores de imóveis que fazem muitas pesquisas na plataforma, não correspondendo ao comportamento médio dos clientes. A partir destas observações, os *outliers* foram detectados e eliminados do conjunto de dados.

3.4.2 Preenchimento de lacunas de dados

Quando os usuários realizam buscas no portal imobiliário, nem sempre eles colocam todos os filtros de dados (como por exemplo a área do imóvel), isso provoca algumas lacunas nos dados coletados. Além disso, lacunas nos dados podem ter sido causadas por erros no sistema ou por problemas de implementação, como em certos casos que foram registrados usuários que fizeram um contato mas não uma visualização, o que não faz muito sentido quando retornamos para analisar o fluxo comportamental desses usuários na plataforma. Por estes motivos houve a necessidade de preencher essas lacunas nos dados para que não houvesse prejuízo na qualidade do modelo. Como os valores das features escolhidas são puramente numéricas, foi optado o preenchimento das lacunas dos dados com valores iguais a zero para simbolizar casos em que o usuário não preencheu aquele campo. A escolha de preenchimento das lacunas com zero pôde ser feita por ela não afetar o valor dos demais casos, isso pode ser visto no campo de preço, pois não há apartamentos cujo preço seja zero.

3.4.3 Correção dos dados

Um grande problema com o *dataset* foi a grande quantidade de dados duplicados (seção 3.1), essa duplicação foi causada por dois motivos principais: o repasse dos dados do Expoimovel para o Smart e a falta de um tratamento prévio na hora de salvar o e-mail dos usuários na base de dados. Por este tratamento não ter sido feito, emails da mesma pessoa só que com diferença de caixa de texto foram salvos como se fossem pessoas diferentes. A solução para este problema foi a transformação de todos os emails para caixa baixa e o agrupamento dos emails iguais (ver seção 2.1.3). Essa falta de tratamento dos e-mails dos usuários causou outros problemas. Por não limitar os caracteres que poderiam ser inseridos no campo de email, por erro de digitação houveram usuários que colocaram caracteres inválidos nesse campo, quebrando o modelo de identificação. Para estes casos foi escolhida uma deleção simples por estes endereços eletrônicos não serem válidos.

3.4.4 Geração do conjunto de treinamento e de teste

Após todo o pré-processamento de dados foi possível gerar o conjunto de treinamento e de testes que possuía uma estrutura propícia para a construção de um modelo de aprendizagem. Foi escolhido o formato csv por sua simplicidade e facilidade de ser entendido e manipulado. No próximo capítulo iremos expor e discutir os resultados dos algoritmos de classificação que usaram como entrada o conjunto de treinamento e processado.

CAPÍTULO 4

Experimento

Após o processamento dos dados, tornou-se possível a execução do experimento para tentar responder a pergunta que foi escolhida (ver seção 3.3), para isto rodamos e comparamos diversos classificadores para avaliar qual deles é o mais adequado para o nosso propósito. Neste capítulo iremos discutir a execução e os resultados do experimento proposto no trabalho, assim como seus aspectos técnicos e peculiaridades.

4.1 Características do conjunto de dados de entrada

O conjunto de dados de entrada possui informações sobre o comportamento dos usuários no portal imobiliário Expoimóvel e nos sites de construtoras e imobiliárias. As *features* selecionadas foram discutidas na seção 3.3 e servirão para a construção do modelo de previsão de possíveis usuários que podem entrar em contato a respeito do interesse em um imóvel. A tabela abaixo mostra as características desse conjunto de dados, lembrando que os casos rotulados como positivos são aqueles em que o usuário entrou em contato e os negativos que ele não entrou em contato.

Positivos	**Negativos**
1694	6252
Total: 7946	

Tabela 4.1 Características do conjunto de treinamento

A tabela 4.1 mostra que a quantidade de casos positivos é bem inferior a de negativos, podendo causar o enfraquecimento na confiabilidade da métrica de acurácia apontada por Li [20]. Na tabela 4.2 pode ser vista a divisão do conjunto de dados entre treinamento e teste.

Treinamento	**Teste**
6356	1590
Total: 7946	

Tabela 4.2 Proporção entre os conjuntos de treinamento e teste

Esse valor de oitenta por cento para conjunto de treinamento e vinte por cento para teste foi escolhido por ser recomendado pela maioria dos autores [22] e serviu adequadamente para o nosso caso, como iremos ver nos resultados exibidos mais adiante.

4.2 Características de implementação dos classificadores

Os algoritmos de aprendizagem de máquina foram implementados na distribuição *anaconda* [6] que é baseada na linguagem de programação *python*, que é uma distribuição *open-source* amplamente usada em aplicações do campo de ciência dos dados. Ela tem como principal vantagem a facilidade de administração e *deployment* de bibliotecas. Outra grande vantagem é que esta distribuição possui as principais bibliotecas que são usadas para a criação de aplicações de classificação e análise de dados como *sklearn* [4], *pandas* [7] e *numpy* [3].

4.3 Algoritmos de classificação usados

Para este experimento, foram escolhidos os principais algoritmos que são geralmente usados para classificação de comportamento de usuários, assim como alguns mais simples que são usados para diversas tarefas de classificação supervisionada. São eles:

- Naive Bayes Gaussiano com os parâmetros padrão do *sklearn*[4]
- Logistic Regression com os parâmetros padrão do *sklearn*
- Decision Tree com *max-depth* = 5
- Linear SVC com os parâmetros padrão do *sklearn*
- Random forest com *max-depth* = 2 e *gini criterion*

4.4 Análise dos classificadores

Nessa seção iremos mostrar os resultados obtidos através da comparação dos classificadores que usaram o conjunto de treinamento e teste citado anteriormente.

4.4.1 *Naive* Bayes Gaussiano

	Precision	Recall	F-measure
Positivos	0.79	0.98	0.88
Negativos	0.56	0.09	0.16
Média/Total	0.74	0.78	0.72
Score do Treinamento: 79.5%			
Tempo de treinamento: 0.01s			
Acurácia: 78%			

Tabela 4.3 Métricas para o algoritmo *Naive* Bayes

O que mais se destaca nesse classificador é sua rapidez de treinamento, porém ele não bons resultados de *recall* para os exemplos negativos e consequentemente isso afetou o valor

do *f-measure*. Estes valores podem ter sido causados pela correlação entre algumas *features* e pela distribuição dos dados, que são fatores que normalmente impactam este algoritmo de classificação.

4.4.2 Regressão Logística

	Precision	Recall	F-measure
Positivos	0.83	0.97	0.89
Negativos	0.72	0.25	0.37
Média/Total	0.80	0.82	0.78
Score do Treinamento: 82.8%			
Tempo de treinamento: 0.02s			
Acurácia: 82%			

Tabela 4.4 Métricas para o algoritmo de regressão logística

O tempo de treinamento desse classificador também foi bem curto, porém como no caso do *naive bayes* os valores das métricas para os casos negativos foi baixo, isso também pode ter sido causado pela distribuição dos dados.

4.4.3 Linear SVC

	Precision	Recall	F-measure
Positivos	0.84	0.97	0.90
Negativos	0.73	0.31	0.44
Média/Total	0.81	0.83	0.80
Score do Treinamento: 82.8%			
Tempo de treinamento: 0.45s			
Acurácia: 79%			

Tabela 4.5 Métricas do algoritmo *Linear SVC*

O tempo de treinamento desse algoritmo foi o maior e seus resultados não foram muito bons, mas ainda sim melhores que os dos algoritmos anteriores. Isso pode ter sido causado pela característica peculiar da construção dos *support vectors*, do tamanho do conjunto de dados e da baixa esparsidade das *features*.

4.4.4 *Decision Tree* ou árvore de decisão

Este classificador foi o que de longe obteve os melhores resultados, menor tempo de treinamento e métricas com valores mais balanceados tanto nos exemplos positivos quantos pros negativos. O principal motivo foi que este algoritmo tem bom desempenho em conjuntos de dados com poucas features e com limites de decisão paralelos ao eixo das *features*.

	Precision	Recall	F-measure
Positivos	0.90	0.94	0.92
Negativos	0.71	0.60	0.65
Média/Total	0.86	0.86	0.86
Score do Treinamento: 87.2%			
Tempo de treinamento: 0.008s			
Acurácia: 86%			

Tabela 4.6 Métricas do algoritmo de árvore de decisão

4.4.5 Random forest

	Precision	Recall	F-measure
Positivos	0.84	0.91	0.88
Negativos	0.57	0.40	0.47
Média/Total	0.78	0.80	0.79
Score do Treinamento: 80.5%			
Tempo de treinamento: 0.04s			
Acurácia: 79%			

Tabela 4.7 Métricas do algoritmo *Random Forest*

Esse algoritmo obteve um resultado semelhante por ser baseado no *Decision Tree*, porém teve um tempo de treinamento maior por ter que separar o conjunto de dados em subconjuntos e treinar várias instâncias do algoritmo. O resultado das métricas pode ter sido pior pela aleatoridade da escolha dos subconjuntos, apesar disso o resultado desse algoritmo foi melhor que o da maioria.

4.5 Conclusões a respeito da análise dos algoritmos

Classificador	*F-measure*
Árvore de decisão	0.92
Linear SVC	0.90
Regressão logística	0.89
Random Forest	0.88
Naive bayes	0.88

Tabela 4.8 Valores de *f-measure* dos classificadores relativos à classe positiva

De acordo com os resultados mostrados na tabela 4.8 e na análise dos classificadores (seção 4.4), chegamos a conclusão que o algoritmo *Decision Tree* é o mais adequado para a realização da qualificação de leads por apresentar os melhores resultados dentre os outros algoritmos.

Com ele podemos ranquear possíveis compradores de acordo com a probabilidade de uma entrada ser um possível comprador ou não, esta probabilidade pode ser facilmente obtida através de funções do *sklearn*. A execução dos algoritmos foi rápida e simples, porém a maior parte do esforço deste trabalho, cerca de 85% do tempo de estudo e implementação, veio da etapa de organização e pré-processamento de dados. Confirmando o que foi dito no capítulo 2 sobre a proporção de tempo gasta no pré-processamento em comparação com as outras tarefas de um cientista de dados.

CAPÍTULO 5

Conclusão

Neste trabalho foi apresentado o processo de construção de um ambiente propício para a execução da tarefa de classificação de possíveis compradores de imóveis. O estudo realizado contemplou a preparação básica desse ambiente, a análise dos dados e do domínio, o pré-processamento desses dados e por fim, a execução de algoritmos de classificação. Após executados os algoritmos, percebemos que, para esta tarefa, o que obteve o melhor desempenho foi o algoritmo de árvore de decisão. Este algoritmo apresentou uma acurácia de 86% e um valor de 0.92 para o *f-measure* dos casos positivos, assim como um baixo tempo de treinamento.

Neste processo, percebemos que a organização e o processamento dos dados foi a etapa mais trabalhosa do *pipeline*, consumindo mais de 80% do tempo de trabalho em que foi feito este estudo. Além de ser o processo mais custoso, ele também é a tarefa que garante a qualidade do resultando final, assim como o auxilia no entendimento do conjunto de dados.

O desbalanceamento das classes foi um fator que prejudicou a confiabilidade da métrica de precisão, porém este foi um ponto interessante que foi ressaltado no estudo e que pode servir de auxílio para estudos futuros. Outro fator que dificultou o processamento e a análise dos dados foi a sua característica não-normal de distribuição, pois impactou no processo de eliminação de *outliers*. Para estudos futuros, poderia se estudar e aplicar métodos mais sofisticados de remoção de *outliers* para distribuições não normais, assim como métodos baseados em aprendizagem de máquina para preenchimento de lacunas nos dados.

Também como sugestão, a possível implementação de mais algoritmos do tipo *ensemble* usando a tecnologia do projeto *auto-ml* [8] que poderiam melhorar os resultados de classificação. Outro ponto interessante seria a análise a longo prazo do impacto que estas técnicas causam nas vendas de uma empresa, formas de tirar proveito de seus benefícios e calibrá-las através do acompanhamento e estudo dos seus resultados.

Referências Bibliográficas

[1] Expoimóvel. http://www.expoimovel.com/.

[2] Ingaia ads: Plataforma de geração de leads para o mercado imobiliário. https://webcompany.com.br/news/ingaia-ads-plataforma-de-geracao-de-leads-para-o-mercado-imobiliario.

[3] Numpy. http://www.numpy.org/.

[4] Sci-kit learn. http://scikit-learn.org/stable/index.html.

[5] Smart imobiliário. http://www.smartimobiliario.com.br/.

[6] *Anaconda Cloud*. https://anaconda.org/anaconda/python.

[7] *Pandas - Python Data Analysis Library*. https://pandas.pydata.org/.

[8] *Python auto-ml*. https://pypi.org/project/automl/.

[9] *Smart Lead*. http://www.smartlead.com/.

[10] How digital marketing will change in 2018: 15 top trends. https://www.forbes.com/sites/forbesagencycouncil/2017/12/18/how-digital-marketing-will-change-in-2018-15-top-trends/#7a2dd92d2d9a, 2018.

[11] WO Bussab and PA MORETIN. Estatística básica. 5ª edição, editora sariva, s, 2004.

[12] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

[13] Carroll Croarkin, Paul Tobias, JJ Filliben, Barry Hembree, and Will Guthrie. *NIST/SEMATECH e-Handbook of Statistical Methods*, chapter 7.1.6, pages 1–1. NIST/SEMATECH, https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm, 2006.

[14] Carroll Croarkin, Paul Tobias, JJ Filliben, Barry Hembree, and Will Guthrie. *NIST/SEMATECH e-Handbook of Statistical Methods*, chapter 1.3.5.17, pages 1–1. NIST/SEMATECH, https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm, 2012.

[15] Arthur Middleton Hughes. *The complete database marketer: second-generation strategies and techniques for tapping the power of your customer database*. McGraw-Hill, 1996.

[16] B. Iglewicz and D.C. Hoaglin. *How to Detect and Handle Outliers*. ASQC basic references in quality control. ASQC Quality Press, 1993.

[17] Rob Jackson. Google adwords and digital marketing – bricks for clicks. `https://www.theguardian.com/media-network/media-network-blog/2013/mar/28/google-adwords-digital-marketing-retail-seo`, 2013.

[18] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

[19] Erik G Learned-Miller. Introduction to supervised learning. *I: Department of Computer Science, University of Massachusetts*, 2014.

[20] Charles X Ling and Chenghui Li. Data mining for direct marketing: Problems and solutions. In *KDD*, volume 98, pages 73–79, 1998.

[21] Dan MacDade. *Sales Tips, What is a Lead?* `https://www.salesforce.com/blog/2014/03/what-is-a-lead-gp.html`, 2014.

[22] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[23] T Pham-Gia and TL Hung. The mean and median absolute deviations. *Mathematical and Computer Modelling*, 34(7-8):921–936, 2001.

[24] Gil Press. Cleaning big data: Most time-consuming, least enjoyable data science task, survey says. `https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-tas#119cf7ee6f63`, 2016.

[25] D. Kanellopoulos "S. B. Kotsiantis and P. E. Pintelas". Data Preprocessing for Supervised Leaning . *International Journal of Computer Science*, 1(2):111–117, 2006.

[26] Jake VanderPlas. *Python data science handbook: Essential tools for working with data*. "O'Reilly Media, Inc.", 2016.

[27] Luis E Zarate, Bruno M Nogueira, Tadeu RA Santos, and Mark AJ Song. Techniques for missing value recovering in imbalanced databases: Application in a marketing database with massive missing data. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, volume 3, pages 2658–2664. IEEE, 2006.

Este volume foi tipografado em LaTeX na classe UFPEThesis (www.cin.ufpe.br/~paguso/ufpethesis).