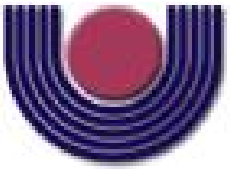


Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

Estimação de preços de imóveis na cidade de Cascavel assistida por técnicas de Aprendizagem de Máquina

Murilo Francisco Schaefer

CASCAVEL
2017

MURILO FRANCISCO SCHAEFER

Estimação de preços de imóveis na cidade de Cascavel assistida por técnicas de Aprendizagem de Máquina

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Prof. Dr. André Luiz Brun
Co-Orientador: Prof. Dr. Ivonei Freitas da Silva

CASCAVEL
2017

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

IA	Inteligência Artificial
KNN	K-Nearest Neighbors
RNA	Rede Neural Artificial
SVM	Support Vector Machine
SVR	Support Vector Regression
RF	Random Forest
MSE	Mean Square Error
RL	Regressão Linear
MAD	Mean Absolute Deviation

Lista de Símbolos

ρ Coeficiente de Spearman
ϵ Margem máxima para o SVR

Sumário

Resumo

O setor imobiliário tem grande importância perante a economia de determinada região, bem como em seus habitantes. O valor dos imóveis pode ser usado como estimador para a inflação, por exemplo. Para tanto, é necessário conseguir realizar com precisão a estimação dos valores dados seus atributos. Este trabalho tem como objetivo avaliar métodos de aprendizagem de máquina aplicada ao processo de estimação dos valores imobiliários na cidade de Cascavel-PR.

Para realizar a avaliação dos modelos, foram construídos quatro modelos: Regressão Linear, Rede Neural Artificial, Support Vector Regression e Random Forest. Embora a regressão linear não seja um método de aprendizagem de máquina, foi usado para comparação dos outros modelos.

Os dados para treinamento e teste foram obtidos a partir da demanda de uma entidade da cidade. Então estes dados foram tratados a fim de remover ruídos e distorções. Além disso, ainda foi realizada a inferência de novos atributos, como o valor médio da vizinhança e o resultado dos agrupamentos k-means e DBScan.

A fim de avaliar a precisão dos métodos foram realizadas 32 repetições dos experimentos. A comparação entre os modelos baseou-se nas métricas dos erros médios quadrados (MSE) e desvio absoluto médio (MAD). Como resultado os modelos de aprendizagem de máquina se mostraram mais precisos que o método de regressão linear. Dentre os métodos de aprendizagem de máquina, o mais preciso foi o Random Forest, apresentando um erro absoluto médio de 0.04 que representa um erro de 4%.

Além da precisão das abordagens propostas, avaliou-se o impacto da inferência de novos atributos sobre os modelos. Os quais mostraram uma melhoria significativa quando adotados.

Palavras-chave: Inteligência Artificial, imobiliário, estimação de preço, aprendizagem de máquina.

Capítulo 1

Introdução

O setor imobiliário tem grande influência na economia, onde a oscilação nos valores dos imóveis causam grande impacto na vida dos habitantes, principalmente para pessoas em que o bem de maior valor agregado é sua residência. A construção, manutenção e reforma de imóveis ainda fomentam o mercado gerando empregos e renda.

O setor, segundo Li, Leatham et al. (2011), pode ser utilizado como indicador para avaliação da inflação e produção da economia. Por estes fatores a estimação do valor de imóveis, de forma precisa e ágil, provê informações valiosas para economistas que, com base nos dados levantados, podem fazer uma estimação mais precisa da inflação.

Além de contribuir na análise econômica de uma sociedade, a estimação pode auxiliar na tomada de decisão por parte dos proprietários de imóveis (LI; LEATHAM et al., 2011). Além disso, uma estimação apurada é importante para proprietários, construtores, investidores, avaliadores e demais entidades envolvidas no mercado, tal como seguradoras e prefeituras (KHAMIS; KAMARUDIN, 2014); (KUŞAN; AYTEKIN; ÖZDEMIR, 2010); (PAGOURTZI et al., 2003).

O preço de um imóvel está relacionado a fatores macroeconômicos, como inflação, e fatores locais como diferenças espaciais, estrutura da comunidade e amenidades pontuais como características físicas e ambientais do imóvel (LI; LEATHAM et al., 2011); (KUŞAN; AYTEKIN; ÖZDEMIR, 2010).

O valor de mercado é estimado através da aplicação de métodos e procedimentos que refletem a natureza da propriedade e as circunstâncias às quais o imóvel está submetido. Uma forma para realizar esta estimação é quantificar a relação entre atributos do imóvel. Entretanto, a quantificação das relações entre as variáveis não é o suficiente para uma estimação precisa, pelo fato de haver muitos atributos reais e nominais envolvidos (LI; LEATHAM et al., 2011).

Segundo Pagourtzi et al. (2003), um dos métodos mais aplicados na estimação de valores é a utilização de alguma confrontação, como a comparação de capital direto e análise sobre os intervalos de observações. Estes métodos são conhecidos como métodos tradicionais.

Outros métodos que se utilizam de heurísticas para tentar analisar o mercado simulando seu comportamento, são ditos, segundo Pagourtzi et al. (2003), avançados. Neste escopo estão presentes os algoritmos de aprendizado de máquina, método de preços hedônicos e *lógica fuzzy* (PAGOURTZI et al., 2003).

Métodos hedônicos consistem de métodos que utilizam da estatística para tentar mensurar o valor de um bem de acordo com preferências individuais do ambiente em que este bem está inserido (NETO, 1976).

O campo da aprendizagem de máquina é uma área multidisciplinar que envolve conteúdos de inteligência artificial, probabilidade e estatística, neurobiologia, psicologia, filosofia entre outras disciplinas, onde se desenvolve e avalia a capacidade de uma máquina adquirir conhecimento dado uma experiência e validando este conhecimento de acordo com uma análise de desempenho (MITCHEL, 1997).

Com o uso da aprendizagem de máquina muitas aplicações de mercado com impacto significante começaram a ser desenvolvidas (MITCHEL, 1997). Esta influência é observada também no setor imobiliário onde foram desenvolvidas aplicações que se mostraram tão eficientes ou melhores que os métodos tradicionais.

Dada a importância atrelada às propriedades imobiliárias é proposto neste trabalho a análise de estratégias de aprendizagem de máquina com o objetivo de, dadas informações de imóveis reais, estimar o preço de um novo imóvel com base em suas características para o município de Cascavel - Paraná, avaliando também qual técnica é a mais adequada para realizar a estimação na cidade.

Para tanto, inicialmente, foi avaliado a qualidade dos atributos contidos na base de dados obtida sobre o nicho imobiliário da cidade, e então, realizado o tratamento dos dados tal como a remoção de inconsistências e *outliers*.

Então, utilizando técnicas de análise de relação entre variáveis, como o Coeficiente de Correlação de Pearson explicado com mais detalhes na seção 2.3, buscou-se descobrir quais atributos tem maior relação com o valor de um imóvel.

Para um melhor entendimento de como os dados estão dispostos foram utilizados algoritmos de agrupamentos, discutidos em detalhes na seção 2.4. A aplicação destes métodos mostra as divisões entre classes de imóveis, como a proporção entre casas de baixo padrão para casas de alto padrão.

Após a etapa de levantamento e preparação dos dados foram realizadas as etapas de treinamento e validação dos métodos de aprendizagem de máquina com aplicação na estimação do valor de mercado das propriedades. Neste intento, foram trabalhadas várias abordagens distintas, as quais tiveram seus desempenhos (em termos de precisão) comparados.

Buscou-se, com a realização desta pesquisa, levantar que tipo de informações são interessantes para o processo de estimação automatizada de valores de imóveis, bem como estudar quais estratégias são mais adequadas para o processo de estimação.

O Capítulo 2 apresenta a revisão bibliográfica sobre aprendizado de máquina e métodos de avaliação dos mesmos, apresentando os conceitos de cada método e suas características. No Capítulo seguinte, apresentamos em detalhes o processo operacional adotado no trabalho. Já o Capítulo 4 engloba a aplicação dos métodos estudados no contexto de Cascavel e, por fim, o Capítulo 5 apresenta as conclusões do trabalho.

Capítulo 2

Aprendizado de Máquina

Na área da Inteligência Artificial (IA) alguns problemas são difíceis para se representar algoritmicamente, como por exemplo, o reconhecimento da fala. A partir do momento que se assume que o agente[1] tem implementado todo o conhecimento de seu desenvolvedor e é aplicado a algum cenário com o objeto de resolver alguma tarefa, sempre realizará as melhores operações conforme foi programado para fazer. Porém esta nem sempre é a melhor abordagem. Caso o desenvolvedor do agente não tenha todo conhecimento necessário para a resolução do problema, o aprendizado é a única forma pela qual o agente compreenderá o que tem que saber e fazer (RUSSELL; NORVIG, 2003).

Ainda não se tem conhecimento de como fazer computadores aprenderem tão bem quanto humanos, mas diversos algoritmos mostraram-se eficientes para determinadas tarefas de aprendizado. Para campos de atuação como mineração de dados[2] o aprendizado de máquina mostrou-se bastante eficiente e útil, extraindo conhecimento, que antes não se tinha, de conjuntos de dados grandes (MITCHEL, 1997).

Para se aprofundar na área de aprendizado de máquina, é relevante haver uma definição do que o termo significa. Uma acepção é apresentada em Mitchel (1997), onde o autor define o aprendizado como um programa de computador capaz de aprender de uma experiencia E em respeito a uma classe de tarefas T e medindo a performance P. Melhora-se a performance P das tarefas T, com o ganho de experiência E.

Ou seja, dado um método de avaliação, enquanto houver um erro maior que o aceitável, o

[1] agente: qualquer objetivo que perceba o meio em que está através de sensores e age neste meio através de atuadores

[2] mineração de dados: utilização de técnicas para extração de conhecimento a partir de vastos conjuntos de dados

sistema continuará ganhando experiência a fim de aperfeiçoar a realização da tarefa e reduzir o erro.

2.1 Funcionamento do aprendizado de máquina

Um agente capaz de aprender pode ser dividido em quatro componentes conceituais: Análise, Treinamento, Performance e Gerador. Uma ilustração destes agentes é apresentada na Figura 2.1.

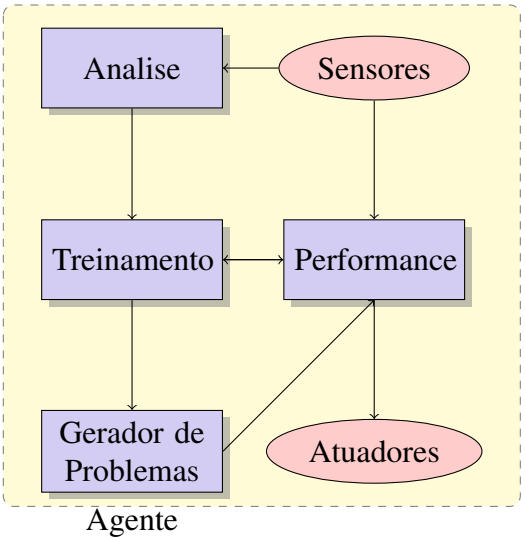


Figura 2.1: Modelo geral de agentes. Adaptado de Russell e Norvig (2003)

A distinção mais importante ocorre entre o elementos de treinamento e performance. O primeiro é responsável por realizar melhorias no agente. Para tanto, utiliza técnicas como a média dos erros quadrados para avaliar o erro, comparando os resultados obtidos com os esperados. Já o elemento de performance é responsável pelas ações externas. Ele recebe informações sobre o que deve ser realizado, como por exemplo, alteração dos coeficientes de algum método de classificação, para então o elemento de análise conseguir avaliar os novos resultados.

O elemento de análise tem como função determinar o quão bem um agente está se saindo em um determinado ambiente. Para a construção deste elemento é necessária uma performance padrão indicando o sucesso ou fracasso do agente.

Para conseguir sugerir ações que irão levar o agente a uma nova experiência e informação faz-se necessário o elemento Gerador de Problemas, o qual tem o papel de gerar possibilidades

ainda não testadas (RUSSELL; NORVIG, 2003).

Para realizar melhorias na performance, o responsável pelo treinamento utiliza dos *feedbacks* disponíveis pelo componente de análise. O elemento de treinamento pode conhecer o valor esperado de saída, denominado como aprendizagem supervisionada. Por outro lado, pode não se ter informação sobre o resultado desejado, caracterizando assim como aprendizagem não-supervisionada. Ainda temos o caso de se ter informações sobre como o agente está se saindo, mas não como melhorar, conhecido por aprendizagem por reforço (RUSSELL; NORVIG, 2003). O princípio de cada estratégia é discutido a seguir.

2.1.1 Aprendizagem Supervisionada

Quando a entrada apresenta uma saída conhecida, onde se é possível fazer uma comparação entre o resultado obtido e o desejado para se determinar o quão eficiente o agente está agindo, chamamos de aprendizagem supervisionada. Um exemplo de aprendizagem supervisionada é o reconhecimento de *Spam* em *e-mail* (SHWARTZ; DAVID, 2014).

Para saber classificar um e-mail como sendo um *Spam* ou não *Spam*, utiliza-se para o treinamento do agente os e-mails anteriores e se estes foram classificados como sendo ou não um Spam, ou seja, a entrada para treinamento já contém uma resposta se aquele e-mail é um *Spam* ou não (SHWARTZ; DAVID, 2014).

2.1.2 Aprendizagem Não-supervisionada

Em contraste a aprendizagem supervisionada temos a aprendizagem não-supervisionada, onde não há um valor para comparação no conjunto de treinamento, ou seja, quando não se sabe o resultado esperado. Um exemplo disto é o agrupamento de entradas em conjuntos, onde as instâncias de cada grupo é o mais similar possível, como discutido na seção 2.4. Uma aplicação de métodos não-supervisionados é na área de negócios, onde as empresas coletam imensas quantidades de informações sobre clientes e informações sobre o mercado atualmente e, então, utilizam o agrupamento para segmentar os possíveis clientes em um número menor, conseguindo assim, analisá-los e tomar medidas para marketing (TAN et al., 2006).

2.1.3 Aprendizagem por reforço

Na aprendizagem por reforço são avaliadas as ações do agente através de uma pontuação, calculada pelo próprio agente, onde a partir desta pontuação decide-se o que fazer. Por exemplo, considere um agente que deve aprender a se deslocar, a pontuação pode ser dada através do cálculo da distância do ponto de origem até o ponto de destino, e com isso avaliar se o agente melhorou ou piorou dada as tentativas anteriores (RUSSELL; NORVIG, 2003).

Para o treinamento e validação de cada algoritmo, separa-se o conjunto de dados disponíveis em dados de treinamento e de teste (ou, em alguns casos envolve também um conjunto de validação), onde o primeiro é responsável por servir de base de conhecimento para o classificador, o segundo (validação) é adotado para ajustar os parâmetros de cada método e o teste serve para verificar se os resultados atingiram um resultado satisfatório.

Após realizado o treinamento, o agente tem como objetivo a classificação, regressão ou agrupamento de uma determinada entrada.

Todos os tipos de agentes precisam passar pela fase de treinamento, que é onde se ajusta os parâmetros necessários.

2.2 Classificação

Métodos de classificação têm como objetivo categorizar as instâncias de entrada em classes. A ideia consiste em analisar os características do novo padrão e então decidir a qual grupo o elemento pertente. Por exemplo, determinar por inferência se um câncer é benigno ou maligno dada a aparência do tumor e a estrutura de suas células (RUSSELL; NORVIG, 2003).

Nas subseções seguintes são apresentados os métodos de classificação mais comuns na literatura.

2.2.1 *K-Nearest Neighbor*

O método conhecido como *K-Nearest Neighbor*, ou KNN, é um algoritmo para reconhecimento de padrões não paramétricos[3], que tenta encontrar os vizinhos mais próximos de uma

[3]regressão não-paramétrica: conjunto de técnicas utilizadas para aproximar uma curva, quando não se tem muito conhecimento sobre sua forma

observação, analisando seus atributos (ALTMAN, 1992) no espaço de características. Uma ilustração da técnica é apresentada na Figura 2.2.

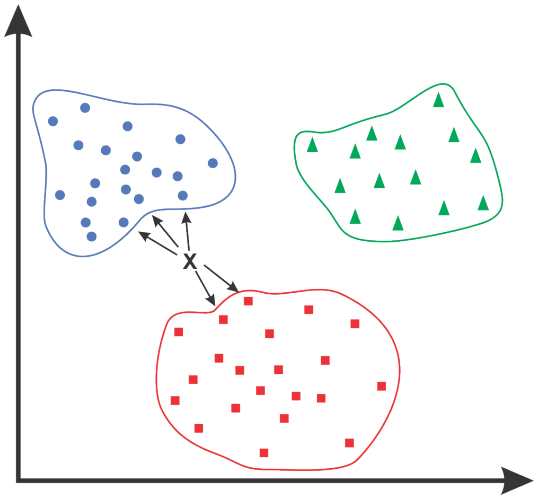


Figura 2.2: Conceito do algoritmo KNN. Adaptado de (RUSSELL; NORVIG, 2003)

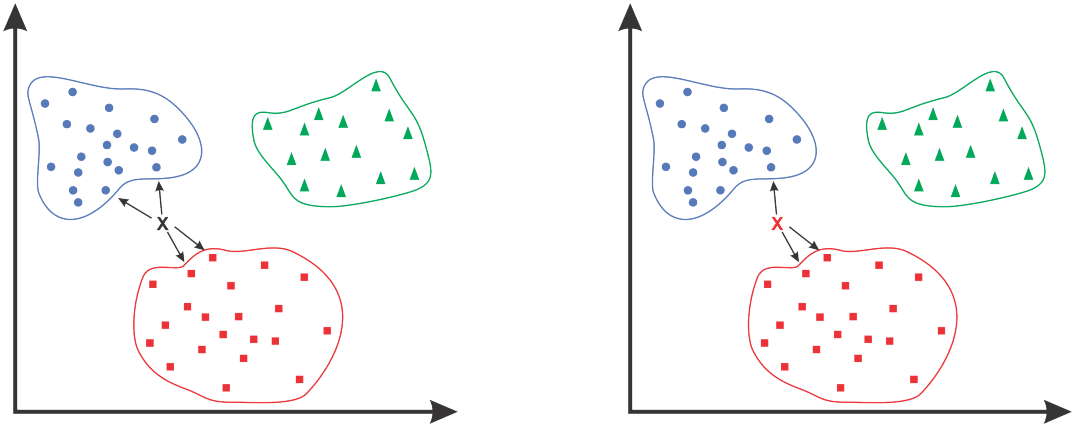
O *K-Nearest Keighbor* é um algoritmo de classificação supervisionado, onde para a etapa de treinamento o mesmo precisa de um conjunto de instâncias com suas características e uma classe de destino para cada entrada. E na fase de teste o algoritmo testa os valores tentando encontrar a classe ao qual aquela entrada pertence.

O KNN encontra os k elementos mais próximos do elemento desejado e, a partir destes vizinhos define sua classe. O processo é caracterizado pelo voto majoritário onde cada vizinho fornece "sua opinião"sobre a classe do novo padrão.

O valor de k é preferivelmente ímpar para tentar evitar empates na quantidade de vizinhos entre determinadas classes (PETERSON, 2009). A Figura 2.3 ilustra este caso, onde em 2.3(a) temos um k igual a 4, resultado em 2 vizinhos da classe A e dois da classe B. Isto gera uma ambiguidade da classe resultante do algoritmo, enquanto que em 2.3(b) k é equivalente à 3 onde há 2 vizinhos da classe A e 1 da classe B.

2.2.2 Árvore de decisão

O classificador por árvore e decisão, utiliza de uma estrutura de árvore para decidir a qual classe pertence o elemento em análise. Embora este método represente uma função booleana, pode-se também criar modelos com mais saídas, como por exemplo, classificar um animal entre mamíferos, répteis ou aves.



(a) k=4 (b) k=3

Figura 2.3: Exemplos para o valor de k. Adaptado de (RUSSELL; NORVIG, 2003)

Cada nodo interno da árvore de decisão corresponde a um teste sobre valores das entradas do padrão de teste, e as arestas de saída representam os possíveis valores de resposta. As folhas da árvore, representam as classes de resposta do algoritmo (RUSSELL; NORVIG, 2003). Uma representação pode ser vista na Figura 2.4, onde o agente tenta reconhecer se um dia é bom ou não para se jogar tênis, dado os atributos de clima daquele dia.

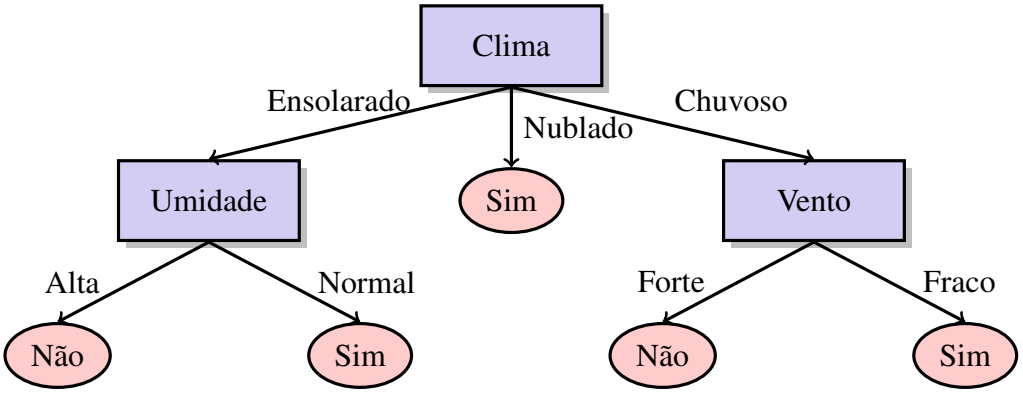


Figura 2.4: Exemplo da estrutura de um Árvore de Regressão. Adaptado de Mitchel (1997)

Em geral, uma árvore de decisão representa uma disjunção de conjuntos de acordo com os valores de seus atributos, onde cada caminho da raiz até as suas folhas levam a um conjunto (MITCHEL, 1997). Porém, este método não consegue representar conjuntos com mais de um objeto, por exemplo, o restaurante próximo mais barato, isto decorre do fato que a linguagem de decisão de uma árvore é essencialmente proposicional, sendo cada atributo testado por uma proposição. Mesmo que podendo se adicionar um atributo que informe qual restaurante é mais

barato, isto não é trivial quando o número de relacionamentos entre objetos aumenta (RUSSELL; NORVIG, 2003).

Árvores de decisão mostraram-se com resultados satisfatórios quando as instâncias são representadas por um par de atributo-valor, por exemplo, temperatura: 10^{o}C e também onde o conjunto de treinamento contenha erros e dados faltantes (MITCHEL, 1997).

Para o treinamento do método, pode-se utilizar uma estratégia trivial, onde a partir do nodo, dado uma amostra de entrada, busca-se um caminho para uma folha (onde há a classificação daquela amostra) seguindo seus atributos, caso não seja achado um caminho, cria-se um. Com isso quando uma outra entrada igual for fornecida, já haverá um caminho e a árvore saberá classificar a entrada. Porém caso seja um outra entrada nada pode ser dito quanto a classificação. Este é um problema já que apenas são memorizadas as entradas e não são extraídas informações para conseguir mapear uma entrada diferente de alguma já vista anteriormente.

A extração de padrões permite descrever um conjunto de casos grande de forma concisa. No caso de uma árvore de decisão, quanto menor a árvore mais consistente ela será e abrangerá mais observações. Para encontrar a menor árvore de decisão há algumas heurísticas, como por exemplo testar o atributo que causa maior impacto na classificação de uma entrada antes dos demais (RUSSELL; NORVIG, 2003).

2.2.3 Redes Neurais Artificiais

As redes neurais artificiais, ou RNA, baseiam-se na ideia do aprendizado biológico, onde uma rede de neurônios com ligações entre eles e cada neurônio é excitado ou inibido dada uma entrada recebida de outro neurônio ou elemento externo. As redes neurais artificiais fazem uso da mesma estratégia, onde cada nodo, análogo aos neurônios, tem uma função de ativação que fornece uma saída de acordo com uma entrada dada.

As RNA's podem ser aplicadas tanto para a classificação de instâncias quanto à regressão de valores, onde a rede pode tentar estimar um valor dado um conjunto de entrada quanto tentar classificar a instância de entrada em uma das classes de saída.

A rede neural é composta por nodos contendo sua função de ativação, conectadas através de *links*, onde cada um destes tem seu próprio peso associado. O aprendizado da rede dá-se pela atualização de tais pesos onde, a cada dado de entrada, valida-se a saída e utilizando um método

chamado *backpropagation* atualiza-se os pesos. A figura 2.5 demonstra a estrutura de um nodo.

Várias funções de ativação podem ser utilizadas nos nodos, como a binária, sinalizada ou sigmóide. Na abordagem binária, a partir de um limiar a saída tornar-se-á 1, caso contrário 0. Quando empregada a solução Sinalizada a saída será -1 se o peso for menor que 0 e 1 se o peso for acima de 0. E por último a Sigmóide, que dará como saída um valor entre 0 e 1, dependendo do valor de entrada. (RUSSELL; NORVIG, 2003)

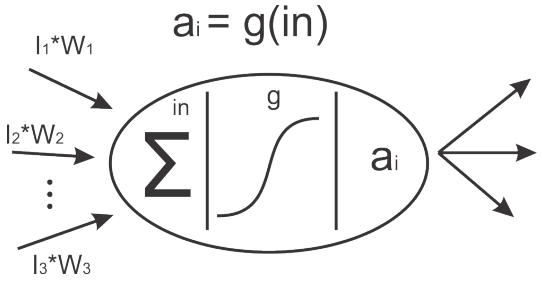


Figura 2.5: Representação de um nodo de uma RNA. Adaptado de (RUSSELL; NORVIG, 2003)

A RNA mais simples é o perceptron, por se tratar de uma rede *feed-forward*, ou seja, os *links* são unidirecionais, não contendo ciclos. Os nodos mais a esquerda são chamados de entradas ou camada de entrada e os nodos mais a direita são chamados de saída ou camada de saída, como apresentado na Figura 2.6. Porém, pelo fato de não conter camadas ocultas, ou seja, camadas intermediarias que conectam a entrada com a saída, a capacidade de representação destas redes é limitada em problemas lineares (RUSSELL; NORVIG, 2003).

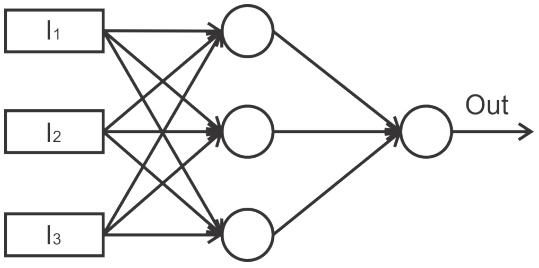


Figura 2.6: Estrutura de um perceptron. Adaptado de (RUSSELL; NORVIG, 2003)

Para resolver o problema de representação do Perceptron, adiciona camadas ocultas ao perceptron, camadas que se localizam entre a entrada e saída, com isso conseguem representar problemas não lineares também, também chamados de RNA multicamadas. Redes *feed-foward* tem como característica uma estrutura parametrizada, o que a torna uma estrutura que o estatísticos chamam de regressão não linear (RUSSELL; NORVIG, 2003).

2.3 Correlação e Regressão

O principal objetivo da análise de regressão é, através de modelos matemáticos, descrever ou explicar a relação entre variáveis. O modelo mais simples de regressão consiste em uma reta, onde esta demonstra a relação entre apenas duas variáveis. Caso haja mais de duas variáveis, tal reta tornar-se-á em um plano ou hiper-plano.

O vetor de entrada é conhecido como variáveis exploratórias, já a resposta obtida é chamada de variável de resposta. Por exemplo, o risco de um ataque cardíaco considerando a idade, peso, pressão sanguínea e taxa de glicose no sangue, neste caso o risco de ataque é a variável de saída o restante faz parte do vetor de entrada. (SEBER; LEE, 2012).

Para se determinar o fator de relacionamento entre as variáveis, utiliza-se um fator de correlação, em geral o fator de Pearson, em que o valor da correlação pode variar entre -1 e 1 sendo que, caso esteja em -1 é uma relação negativa, caso o valor seja 0 não há relação e caso o valor seja 1 é uma relação positiva (FILHO; JUNIOR, 2010). A idade tem uma relação positiva ao risco de ataque cardíaco, ou seja, quanto maior a idade maior o risco. Quanto mais tempo uma pessoa toca um instrumento, menos erros ela irá cometer ao tocá-lo, este é um exemplo de uma correlação negativa entre o tempo de uso do instrumento e os erros cometidos.

O cálculo do coeficiente de Pearson dá-se através da normalização dos atributos realizando o somatório do produto entre os atributos, o resultado obtido através deste cálculo é dividido por $n - 1$ tal que n é o número de amostras (FILHO; JUNIOR, 2010). A fórmula de pearson pode ser conferida na equação 2.1.

$$p = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{(\sum_{i=1}^{n}(x_i - \overline{x}))^2 \cdot (\sum_{i=1}^{n}(y_i - \overline{y}))^2}} \tag{2.1}$$

O coeficiente de Spearman, desenvolvido por Spearman (1904), avalia a intensidade de correlação entre duas variáveis, tal como o coeficiente de Pearson, porém para relações monótonas, ou seja, as variáveis tendem a alterar juntas, porém não em uma taxa constante, enquanto o coeficiente de Pearson diz respeito às relações lineares (SPEARMAN, 1904).

Para se realizar o cálculo de Spearman, primeiro é realizado um ranqueamento[4], e então é obtida a diferença entre cada *rank* e elevado ao quadrado. Após isto, é realizado o somatório do

[4]ranqueamento: uma relação entre dois itens, onde o primeiro é ranqueado como "acima de", "abaixo de"ou "igual a"ao segundo elemento

resultado obtido e multiplicado por 6 e dividido por $n^2 - n$ tal que n é o número de observações. Por fim subtrai-se de 1 o valor obtido, resultando no Coeficiente de Spearman. A Fórmula 2.2 representa o coeficiente, tal que $d_i = rank_x - rank_y$ (SPEARMAN, 1904).

$$\rho = 1 - \frac{6 \sum d_i^2}{n^2 - n} \tag{2.2}$$

2.3.1 Regressão Linear

Regressão linear (RL) é uma ferramenta comum na estatística para estimar o relacionamento entre duas variáveis, ou seja, o quanto uma influência a outra. A regressão linear simples trabalha com apenas 2 variáveis, estimando uma a partir do valor da outra. O modelo é definido assim pela equação $y = \alpha * x + \beta$ onde o resultado consiste de uma reta. Quando trabalhamos com mais de duas variáveis, também conhecido como regressão linear múltipla, o resultado deixa de ser representado por uma reta, passando a ser caracterizado por um hiperplano (SHWARTZ; DAVID, 2014).

2.3.2 Support Vector Machine

Máquina de vetores de suporte ou SVM é um método de aprendizado de máquina útil para o reconhecimento de padrões em conjuntos de dados complexos. Pois trabalha bem com dados de alta dimensionalidade. Uma característica do método é que este representa o limite de decisão, conjunto de possíveis separadores, usando um subconjunto dos exemplos de treinamento, chamado de vetores de suporte.

Porém, pode haver mais de um limite de decisão e, neste caso, o método precisa realizar a escolha de qual limite adotar, para isso utiliza do que é conhecido como lógica da margem máxima. É escolhido o vetor que há a maior margem entre as instâncias de ambas as classes, para calcular o tamanho de uma margem são criados dois vetores paralelos ao limite, um para cada lado, tal que estes vetores insidam nas instâncias mais próximas do limite. A Figura 2.7 ilustra a escolha entre os dois limites de decisão, B1 e B2. O limite escolhido é B2, pois tem uma margem maior. (TAN et al., 2006).

A utilização da lógica da margem máxima, produz um menor erro generalizado, erro computado por todo o conjunto de treinamento e não por instâncias isoladas, obtendo assim um

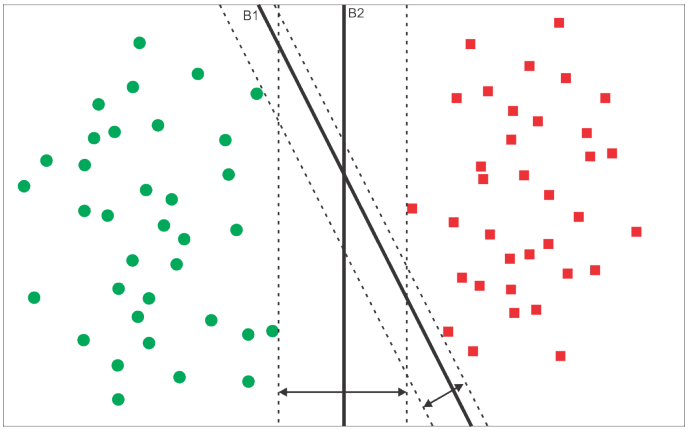


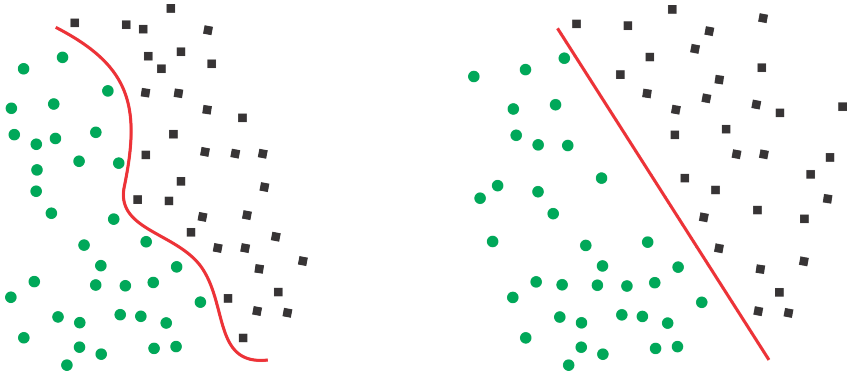
Figura 2.7: Conceito do SVM. Adaptado de (TAN et al., 2006)

melhor desempenho generalizado. Existem duas derivações do SVM, uma com o objetivo de classificação de entradas, conhecido como *Support Vector Classification* ou SVC e outro com o objeto de realizar a regressão de valores, conhecido como *Support Vector Regression* ou SVR. (BASAK; PAL; PATRANABIS, 2007)

Para conseguir realizar uma regressão, o SVR tem como meta encontrar uma função $f(x)$ não linear. Como a estrutura da função não é conhecida, considera-se que esta é um polinômio de ordem infinita. Porém, isto pode ser impraticável, já que o espaço de busca pode ser infinito. Então para resolver este problema, faz-se a utilização do *Reproducing kernel Hilbert Space* [5], onde a função é reproduzida por meio de um *kernel trick*, sem a necessidade de se representar todo o espaço (SMOLA; SCHÖLKOPF, 2004), encontrando uma função. A Figura 2.8 apresenta a utilização do *kernel trick*, onde na imagem 2.8(a) pode ser vista a entrada original, que tem uma separação não linear. Já na imagem 2.8(b) podemos ver o espaço mapeado para uma dimensão maior, onde agora as instâncias podem ser divididas por um hiperplano.

Outra caracteristica do SVR dá-se ao fato de que é desconsiderado erros pequenos, onde os erros apenas são computados caso sejam maiores que a margem, como apresentado na Figura 2.9, que os erros das instâncias dentro da margem não são considerados, apenas erros maiores que o definido são computados. Isto faz com que o método evite o *overfitting*, quando um modelo se ajusta demais ao conjunto de treino, porém não representa a realidade em casos ainda não conhecidos (SMOLA; SCHÖLKOPF, 2004).

[5]Reproducing kernel Hilbert Space: é um *Hilbert Space* onde cada ponto de validação pertence a uma função linear contínua



(a) Entrada original

(b) Entrada mapeada para uma dimensão maior

Figura 2.8: Utilização de Kernel Trick. Adaptado de (ALBUQUERQUE, 2016)

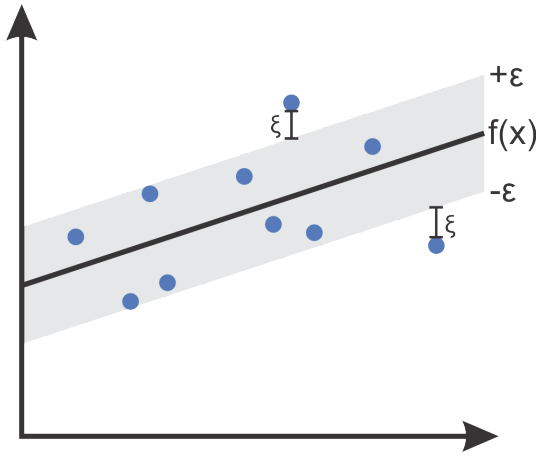


Figura 2.9: Conceito do SVR. Adaptado de (SMOLA; SCHÖLKOPF, 2004)

2.4 Agrupamento

Segundo Fraley e Raftery (1998) as técnicas de agrupamento tem como objetivo separar as entradas em grupos onde são mais semelhantes entre si, considerando os atributos das entradas. Separar as entradas em subconjuntos é útil quando se quer extrair outras informações ou padrões sobre os dados.

Entre os algoritmos de agrupamento os mais comuns são o k-means descrito por Hartigan e Wong (1979), DBScan e Hierárquico. Tais estratégias são apresentadas nas seções a seguir.

2.4.1 k-means

O K-Means é um algoritmo não supervisionado, ou seja, não necessita de um *cluster*[6] de destino na entrada para avaliar o erro. Este algoritmo tenta separar as entradas em K grupos, onde o valor de K é definido arbitrariamente para cada problema.

Para realizar a separação dos grupos, inicialmente o algoritmo define K centróides espalhados aleatoriamente no espaço de características. Então, para cada instância, verifica-se a qual dos K grupos esta pertence, analisando sua distância para cada um dos centroides. A entrada pertencerá ao *cluster* mais próximo. As métricas de avaliação de distância mais comuns são os cálculos da distância euclidiana (distância em linha reta até o ponto) e manhattan (distância horizontal mais a vertical).

Após analisar o *cluster* de cada entrada, é realizado a média dos pontos de cada grupo, e então o centróide passa a ser o ponto resultado do cálculo. Esse processo é então repetido até que não haja mais alterações nos centróides (HARTIGAN; WONG, 1979).

Quando há atributos qualitativos, ou seja, não numéricos, é necessário a quantificação dos valores para estas variáveis.

2.4.2 DBScan

DBScan que é uma sigla para *Density-Based Spatial Clustering of Applications with Noise*, é um algoritmo que realiza a clusterização a partir da densidade de pontos em uma determinada área. A densidade em um ponto é obtida através da contagem do número de elementos em determinado perímetro ao redor do ponto em questão (BIRANT; KUT, 2007).

O funcionamento do método parte da ideia de que se há um aglomerado de pontos, seus vizinhos estariam próximos o suficiente para serem encontrados em uma busca dado um perímetro máximo de busca. O Algoritmo faz uso de três tipos de pontos: os pontos núcleo que são pontos que contém pelo menos $minPontos$ próximos a eles; pontos não núcleos que tem pelo menos um vizinho, porém não tem $minPontos$ próximos; e ainda $outliers$ que são pontos isolados, que não tem nenhum vizinho próximo.

A Figura 2.10 ilustra o método, onde considerando $minPontos = 4$ o elemento A é um ponto núcleo já que dentro de seu perímetro são alcançados 4 pontos, contando com ele. Já

[6] cluster: um aglomerado ou grupo

o ponto B e C são pontos não-núcleos pois mesmo fazendo parte do *cluster*, não alcançam $minPontos$ em seus perímetros. Enquanto o Ponto N é um *outlier* considerando que este não alcança nenhum outro ponto (ESTER et al., 1996).

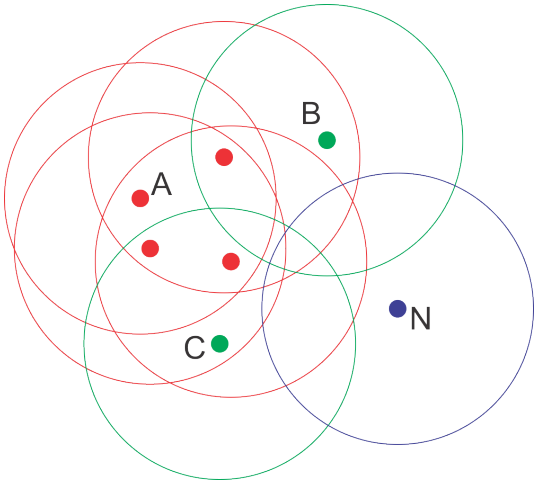


Figura 2.10: Conceito do DBScan. Adaptado de (ESTER et al., 1996)

Para a execução do método não é necessário ter conhecimento de quantos *clusters* pretende-se obter, diferente do k-means apresentado na seção 2.4.1. O DBScan também consegue obter *clusters* em diversas formas, como cluster ovais e lineares, como apresentado na Figura 2.11. Há ainda o tratamento de *outliers* ou *noise*, evitando que entradas fora do padrão acabem interferindo nos clusters (BIRANT; KUT, 2007).

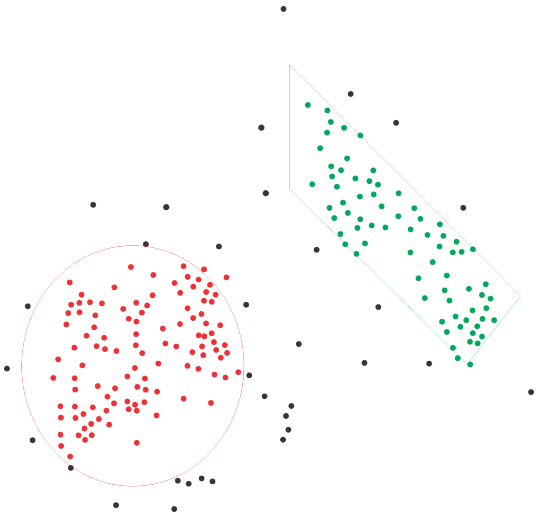


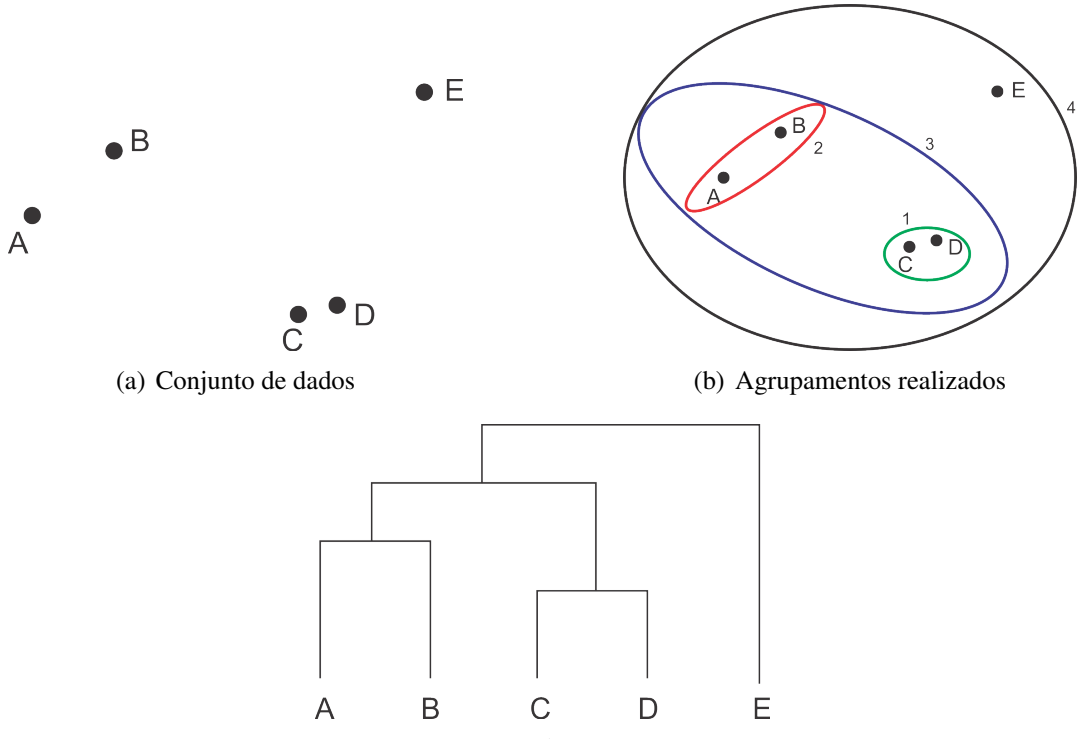
Figura 2.11: Formas do DBScan. Adaptado de (ESTER et al., 1996)

Porém, segundo Ester et al. (1996) o algoritmo não consegue detectar alguns ruídos quando

existem grupos com densidades diferentes. Outro fator, é que em pontos localizados na fronteira de dois grupos será alocado a um dos grupos dependendo da ordem de processamento, o que torna o algoritmo não determinístico.

2.4.3 Hierárquico

Uma estratégia com esta característica utiliza-se de uma hierarquia entre os elementos para conseguir formar grupos. O método considera que inicialmente cada elemento é um grupo constituído de apenas 1 individuo e então vai agrupando os *clusters* mais próximos até que se tenha apenas um grupo envolvendo todos os elementos. Uma ilustração do método pode ser visto na Figura 2.12. Os dados 2.12(a) foram agrupados conforme a Figura 2.12(b), onde os números dos conjuntos mostram a ordem de agrupamento, ou seja, o conjunto CD foi o primeiro a ser criado, seguido do conjunto AB, até haver apenas um grupo. A Figura 2.12(c) apresenta o dendograma das divisões, a altura das linhas condiz com a ordem de agrupamento (LINDEN, 2009).



(a) Conjunto de dados

(b) Agrupamentos realizados

(c) Dendograma

Figura 2.12: Conceito do agrupamento hierárquico. Adaptado de (LINDEN, 2009)

O método não necessita da informação de quantos grupos serão formados, como ocorre no *k-means*, e a visualização dos grupos é realizada através de um dendograma, o que torna a interpretação intuitiva e simples de avaliar quais grupos se uniram e o grau de semelhança entre eles.

Caso seja buscado um número fixo de grupos, a divisão dos grupos pode ser feita após a geração do dendograma. A Figura 2.13 apresenta esta divisão, onde cada corte horizontal, dependendo de sua altura, gera um número diferentes de grupos.

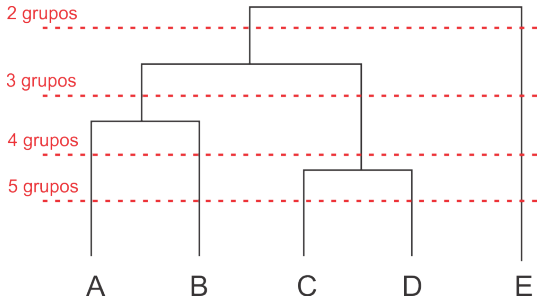


Figura 2.13: Exemplos da divisão dos grupos. Adaptado de (LINDEN, 2009)

Para realizar os passos de agrupamento entre dois *clusters* o método se baseia em alguma métrica de distância, como a distância euclidiana. Então, após estimar a distância entre cada um dos grupos, o método agrupa os elementos mais próximos e repete o processo.

Segundo Linden (2009) a principal vantagem do algoritmo é que ele não fornece apenas os grupos obtidos, mas uma estrutura de dados sobre todos os agrupamentos feitos incluindo também o período em que ocorreram, isso torna a compreensão de como os grupos e subgrupos se relacionam.

2.5 Regressão imobiliária

O setor imobiliário necessita predizer o valor de uma casa considerando atributos como localização, tamanho, quantidade de quartos e vagas de garagens. Isto é importante para conseguir estimar a valorização do mercado em uma região, conforme já discutido na Seção 1.

2.5.1 Trabalhos relacionados

Algumas aplicações desenvolvidas já demonstraram resultados positivos quando a tarefa é a estimação de preço no setor imobiliário, como em Khamis e Kamarudin (2014) que desen-

volveram um estudo para comparar modelos de aprendizagem de máquina (Rede Neural) com modelos estatísticos (Regressão Multilinear). Os resultados indicaram que a primeira estratégia obteve um aumento de 26.47% na precisão da regressão do valor de um imóvel quando comparado com a Regressão Multilinear.

Pow, Janulewicz e Liu (2014) compararam o uso de diferentes algoritmos de aprendizagem de máquina e estatísticos como Regressão Linear, Regressão de Vetores de Suporte (SVR), k-Nearest Neighbours (kNN) e Árvore de regressão. Os autores verificaram que com o uso de kNN em conjunto com Árvore de regressão foi possível predizer o preço dos imóveis em Montreal com uma taxa de erro de apenas 0.0985.

Dada a existência de uma demanda em nossa cidade, pretende-se, com o uso de técnicas de aprendizagem de máquina, realizar um estudo de modelos para regressão nos valores imobiliários de Cascavel-PR utilizando dados sobre imóveis locais como, tamanho da propriedade, número de quartos e garagens, latitude e longitude, bairro, considerando ainda a utilização de fatores externos locais, como mercados e escolas.

Capítulo 3

Metodologia

Conforme descrito no Capítulo 1, para realizar a comparação entre o desempenho dos métodos, inicialmente foram obtidos os dados referentes aos imóveis de Cascavel e, em seguida, foi realizado um tratamento neste conjunto de dados. Após tratados os dados, os modelos de regressão e agrupamento foram treinados utilizando uma fração do conjunto.

Com os modelos treinados, foram então realizados os testes com o conjunto de dados restantes. Nesta fase de teste o objetivo foi obter a acurácia do modelo para uma posterior avaliação entre os métodos.

A Figura 3.1 apresenta a estrutura do trabalho desenvolvido. Conforme a ilustração foram obtidos os dados submetidos a uma série de tratamentos na etapa de pré-processamento. Uma vez tratado o conjunto de dados, foi realizado a inferência de novos dados a partir das informações atuais. Então, estes dados foram divididos em dois conjuntos, um para treino e outro para teste.

O conjunto de treino serviu de base para os modelos, para que estes pudessem aprender com o conjunto. A partir do modelo treinado, foi utilizado o conjunto de teste na etapa de avaliação para validar a acurácia do método. Cada etapa do protocolo será melhor explorada nas seções seguintes.

Para a realização de cada etapa, foi utilizado a linguagem de programação Python em conjunto com as bibliotecas *sklearn* e *pandas*. A biblioteca *pandas* foi utilizada para a obtenção de estatísticas do conjunto de entrada, e ainda para o tratamento dos dados (AUGSPURGER CHRIS BARTAK, 2017).

A ferramenta *sklearn* foi responsável pela implementação dos modelos de regressão, bem como os métodos de avaliação da acurácia de cada método (PEDREGOSA et al., 2011).

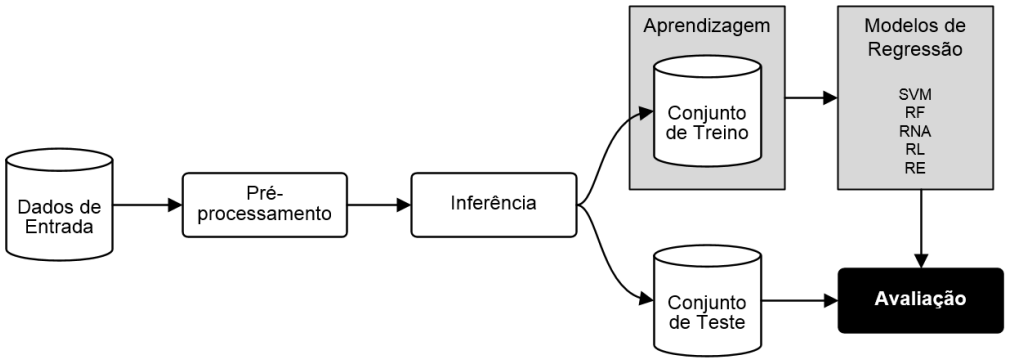


Figura 3.1: Estrutura do protocolo desenvolvido

3.1 Dados

A primeira etapa foi a coleta de dados. A obtenção destes deu-se através de uma empresa da cidade de Cascavel, a qual já realiza a coleta sobre vendas imobiliária na cidade desde 2015. Esta empresa tem como objetivo realizar uma busca em todas as imobiliárias da cidade e disponibilizar todas as ofertas de forma conjunta em um único aplicativo, evitando que o usuário precise visitar várias imobiliárias.

Foi realizado um tratamento sobre o conjunto de entrada, para que este representasse melhor a realidade dos imóveis presente na cidade. Então, após tratado, foi avaliada a existência de correlação dos atributos disponíveis com o valor do imóvel, para isto foram utilizados os coeficientes de Pearson e Spearman, ambos discutidos na seção 2.3.

3.1.1 Pré-processamento

Os dados poderiam conter inconsistências ou ainda valores faltantes. Devido a estes fatores, o conjunto de dados foi tratado de forma a eliminar ou diminuir alguns erros que tornavam estas entradas não condizentes com a realidade ou que pudessem gerar tendências irreais na interpretações dos valores.

A inconsistência dos dados dá-se quando há um imóvel com valores cadastrados errados, como por exemplo, o tamanho de um apartamento ser $2m^2$. Para remover tais inconsistências, qualquer entrada com um atributo menor do que meio desvio padrão acima do valor mínimo ou maior que dois desvios padrão acima da média foi excluída.

O problema de dados faltantes ocorre quando uma entrada não contém todos os atributos

necessários. Um método para contornar tal situação foi: caso o valor do atributo seja um inteiro, atribuir o valor mais próximo da média do conjunto, e caso seja um valor fracionário, atribuir o valor da média.

Devido a busca de informações em mais de uma imobiliária, há entradas duplicadas no conjunto de dados, já que o mesmo imóvel pode ser ofertado por várias imobiliárias. Porém, há ainda imóveis idênticos e que não são os mesmos, como no caso de prédios, onde os apartamentos tem os mesmos atributos, e ainda assim, são registros distintos. Neste caso, foram removidas todas as entradas duplicadas, bem como diferentes apartamentos de um mesmo edifício.

Outro problema tratado foi a existência de *outliers*, imóveis com valores muito acima ou muito abaixo dos demais, pois, como acontecem com baixa frequência, acabam por causar algumas anomalias no treinamento e teste dos modelos. Para solucionar tal problema, estas entradas foram removidas usando a mesma estratégia dos dados inconsistentes.

No passo seguinte os dados ainda foram submetidos a uma tarefa de inferência de novos atributos, para que pudesse obter mais informações sobre cada entrada.

3.1.2 Inferência de novos atributos

Uma vez tratados os dados, a próxima etapa é a inferência de novos atributos, conforme apresentado na Figura 3.1. Nesta etapa buscou-se inferir informações sobre a região onde cada imóvel está localizado.

Isto pois, no setor imobiliário há a dependência espacial, ou seja, o valor de um bem imobiliário depende dos atributos dos imóveis ao seu redor (BOURASSA; CANTONI; HOESLI, 2007). Para se obter informações sobre a região onde está localizado o imóvel, foi realizado o cálculo da média de valor dos apartamentos em um raio de 400 metros.

Outra inferência empregada, foi a utilização de algoritmos de agrupamento, onde foram agrupados entradas com atributos similares, e então, no momento da regressão foi levado em consideração o índice do grupo ao qual o imóvel pertence. Após a inferência, os dados foram utilizados para o treinamento e teste de cada modelo.

3.2 Métodos Utilizados

Para o treinamento e teste dos modelos, o conjunto de dados foi separado aleatoriamente em dois sub-conjuntos, onde o primeiro, treinamento, corresponde à 66% do total de registros e o segundo, teste, os 34% restantes, esta técnica é conhecida como *hold-out* (RUSSELL; NORVIG, 2003). Esta tarefa foi realizada utilizando o *sklearn*. Todos os modelos de regressão e agrupamento baseiam-se nos mesmos conjuntos resultantes, não sendo realizada esta tarefa a cada novo método. Isto nos permitiu realizar a comparação de desempenho dos métodos.

Com o intuito de avaliar o desempenho dos métodos, cada modelo foi executado 32 vezes. O valor utilizado para comparação das estratégias foi a média obtida ao longo das repetições. Um vez que em cada repetição são formados novos conjuntos de treino e teste o protocolo desenvolvido tornou-se bastante robusto.

3.2.1 Aprendizado de máquina

Para avaliar o uso do aprendizado de máquina, foram selecionados os métodos: Rede Neural, SVM e Random Forest (RF), uma variação do método da árvore de decisão, uma vez que estes modelos, segundo os trabalhos relacionados, obtiveram melhores resultados.

Então, o conjunto de treino foi empregado no aprendizado de cada uma das estratégias supracitadas (RNA, SVM e RF). Assim, após o treinamento, foi utilizado o conjunto de teste para validar a acurácia de cada método.

3.2.2 Regressão Estatística

Além das abordagens de Aprendizagem de Máquina, como parâmetro de avaliação foi também empregado um método estatístico para a regressão. Foi utilizado o modelo de regressão linear, sendo esta implementada utilizando a ferramenta *sklearn*.

3.2.3 Agrupamento

Com o objetivo de melhorar a visualização e entendimento de como os dados estão dispostos, foram utilizados algoritmos de agrupamento, auxiliando na tomada de decisão dos métodos como também no entendimento dos resultados. Para realizar esta função foram utilizados os

algoritmos *k-means* e o *DBScan*, já que ambos são não supervisionados e, em nosso caso, ofereceu um tempo de execução baixo.

Ao término da execução da etapa de agrupamento pudemos obter o grupo ao qual cada imóvel pertence, e o mesmo é utilizado como um novo atributo do registro para o treinamento do modelo.

3.3 Avaliação dos métodos

Para avaliar cada um dos métodos, uma vez treinado o modelo, foi utilizado um conjunto de testes, para avaliar o quão acurado era o método. Na avaliação foi utilizado o cálculo do *Root Mean Square Error*(MSE) e o desvio médio absoluto (MAD ou MAE). Os desvios foram obtidos através da distância do valor esperado para o valor encontrado.

A métrica RMSE calcula a média dos desvios ao quadrado, sendo o desvio a diferença entre o valor estimado e o valor desejado, ou seja, calcula a diferença média entre o valor estimado e o desejado, e então extrai a raiz do valor obtido. Quanto mais próximo de zero esta métrica estiver, melhor acurado é o método. A Fórmula 3.1 apresenta a equação do RMSE, onde n é o número de registros, $\hat{Y}_i$ é o valor obtido através do modelo e Y_i é o valor desejado.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2} \tag{3.1}$$

Já a métrica MAD ou MAE, utiliza a média dos desvios absolutos, tem um conceito próximo ao RMSE diferenciando apenas que não eleva os erros ao quadrado(ALLEN, 1971). Para ambas as métricas o resultado obtido é correspondente a porcentagem de erro do modelo que esta sendo avaliado.

Segundo o trabalho de Allen (1971) a métrica RMSE mostra-se mais apropriada quando espera que o modelo tenha uma distribuição gaussiana dos erros, em outros casos é preferível a utilização da média absoluta dos desvios.

A escolha destes métodos de avaliação da-se devido a utilização dos mesmos em trabalhos relacionados, como em Khamis e Kamarudin (2014) que se utiliza das métricas R^2 e MSE para avaliação, embora eles utilizem a métrica MSE para o cálculo, esta métrica calcula a distância quadrada entre o erro e o valor desejado, por isso a escolha do RMSE, onde a distância não é

mais quadrática. Pow, Janulewicz e Liu (2014) avaliam os métodos através do desvio médio absoluto e sua variância.

Capítulo 4

Aplicação

Este Capítulo apresenta a aplicação dos modelos de regressão, bem como os parâmetros utilizados para o funcionamento de cada método. Também trata a avaliação de cada modelo. Como citado no Capítulo 3, foi utilizado o Python em conjunto com as bibliotecas *Pandas* e *Sklearn* para o tratamento dos dados e construção dos modelos, respectivamente.

4.1 Análise dos dados

Nesta seção é discutida a obtenção e tratamento dos dados, avaliando a correlação de cada atributo com o valor da propriedade, para que então estes dados pudessem ser utilizados pelos métodos de aprendizagem de máquina.

4.1.1 Obtenção

Os dados utilizados foram obtidos através de uma empresa da cidade de Cascavel que utiliza um *crawler*[1] que armazena as informações dos imóveis ofertados em todas as imobiliárias da cidade desde 2015.

Porém, foram utilizados apenas os dados dos anos 2016 e 2017, pois estes descrevem melhor o cenário atual do mercado, buscando melhores resultados. Além disso, trabalhou-se apenas com dados sobre apartamentos, devido a maior oferta dos mesmos na cidade e maior consistência nos atributos.

As Tabelas 4.1 e 4.2 apresentam as descrições estatísticas dos conjuntos de dados levantados de 2016 e 2017, respectivamente. Pode ser observado que nem todas as entradas apresentam o

[1] crawler: trata-se de uma ferramenta para realizar a varredura e extração de informações de páginas web

número de garagens ou coordenadas, já que a contagem destes atributos é menor que o número total de entradas do conjunto.

Também puderam ser percebidas inconsistências nas linhas referentes à área do imóvel e à quantidade de dormitórios, em que observou-se valores mínimos são 2 e 0 respectivamente. Como não há imóveis com apenas $2m^2$ ou sem dormitórios (uma vez que estamos trabalhando apenas com apartamentos) optou-se pela remoção dos registros.

É possível visualizar a presença de *outliers* onde o valor médio de um imóvel é de aproximadamente 406 mil reais, com um desvio padrão de 290 mil, há a existência de imóvel com valor de 3.7 milhões, ou seja, uma entrada que está muito acima da média, sendo considerado um *outlier*. Este fato pode ocorrer devido à uma entrada inconsistente ou à real existência do imóvel, representação uma propriedade de altíssimo padrão.

Tabela 4.1: Descrição dos registros obtidos do ano de 2016

Atributo	**tipo**	**Contagem**	**Média**	**Desvio**	**Mínimo**	**Máximo**
Valor	Real	4905	$40.52 * 10^4$	$29 * 10^4$	$7 * 10^4$	$370 * 10^4$
Área	Real	4905	129.4680	77.04	2	945
Dormitórios	Inteiro	4905	2.4903	0.7718	0	32
Garagens	Inteiro	4883	1.5328	0.6469	0	5
Latitude	Real	4826	-24.95393	0.0108	-25.0666	-24.9140
Longitude	Real	4826	-53.46552	0.0173	-53.5138	-53.4075

Tabela 4.2: Descrição dos atributos dos registros de 2017

Atributo	**tipo**	**Contagem**	**Média**	**Desvio**	**Mínimo**	**Máximo**
Valor	Real	16054	$3.91765 * 10^5$	$3.97 * 10^5$	0.00	$33 * 10^5$
Área	Real	16046	145.3171	2936	0	$3.7190 * 10^5$
Dormitórios	Inteiro	15930	2.3970	0.7767	0	32
Garagens	Inteiro	15778	1.6441	7.3368	0	460
Latitude	Real	9747	-24.95625	0.2285	-28.4776	-4.1239
Longitude	Real	9747	-53.42084	0.4429	-54.2345	-3.8248

4.1.2 Tratamento

Como apresentado na Seção 4.1.1, o vetor de entrada necessita ser submetido a um tratamento antes de ser realizado a regressão. Foram mantidas apenas as entradas com uma área acima do mínimo do atributo mais meio desvio padrão (aproximadamente $40m^2$), o mesmo

foi realizado com o valor, onde qualquer instância abaixo de um desvio padrão da média foi removido (aproximadamente 116 mil reais).

Então, para as entradas em que não havia o número de garagens, atribuiu-se como valor o inteiro mais próximo da média do conjunto (geralmente 1 ou 2). O passo seguinte foi remover qualquer entrada onde ainda houvessem campos não preenchidos.

Durante os experimentos, testou-se a possibilidade de excluir os registros que apresentassem dados faltantes ao invés de empregar um valor médio. No entanto, observou-se que esta estratégia alcançou resultados similares, dessa forma, decidiu-se por manter a estratégia do valor médio.

Posteriormente foram removidas quaisquer entradas duplicadas, tal que para duas instâncias serem consideradas iguais é necessário que todos os atributos sejam os mesmos, com exceção do valor do imóvel.

Após, foi realizado a inferência da média do valor das residências em um raio de 400 metros. O resultado obtido da aplicação destes processos é apresentado nas Tabelas 4.3 e 4.4, onde possível verificar que mais de 1900 instâncias foram removidas para o ano de 2016 e 1975 para 2017.

Tabela 4.3: Descrição dos registros de 2016 após os tratamentos

Atributo	**Tipo**	**Contagem**	**Média**	**Desvio**	**Mínimo**	**Máximo**
Valor	Real	3068	$3.9459 * 10^5$	$2.4467 * 10^5$	$1.23 * 10^5$	$15.5 * 10^5$
Área	Real	3068	137.80	71.58	40.65	551.07
Dormitórios	Inteiro	3068	2.52	0.82	1	32
Garagens	Inteiro	3068	1.53	0.64	0	5
Latitude	Real	3068	-24.95393	0.01	-24.99809	-24.9140
Longitude	Real	3068	-53.46552	0.02	-53.512949	-53.4099
Valor médio da vizinhança	Real	3068	$3.9614 * 10^5$	$1.3942 * 10^5$	$1.1850 * 10^5$	$8.2 * 10^5$

Para que fossem realizados os testes de correlação entre as variáveis fez-se necessária a normalização dos atributos, de forma que todos estivessem na mesma escala. Para tanto, empregou-se a estratégia minmax conforme apresentada na equação 4.1.

$$f_i' = \frac{f_i - f_{i\_min}}{f_{i\_max} - f_{i\_min}} \tag{4.1}$$

Em que f_i corresponde ao valor original do atributo do registro, enquanto f_{i_min} e f_{i_max}

Tabela 4.4: Descrição dos registros de 2017 após os tratamentos

Atributo	Tipo	Contagem	Média	Desvio	Mínimo	Máximo
Valor	Real	4079	$3.991 * 10^5$	$2.24 * 10^5$	143000	$14.1 * 10^5$
Área	Real	4079	120.26	59.56	41	392
Dormitórios	Inteiro	4079	2.434175	0.6449	1	5
Garagens	Inteiro	4079	1.294435	0.8035	0	5
Latitude	Real	4079	-24.951875	0.0600	-25.4956	-23.3193
Longitude	Real	4079	-53.457411	0.1549	-53.5130	-49.2651
Valor médio da vizinhança	Real	4079	$4.0059 * 10^5$	$1.2455 * 10^5$	$1.4833 * 10^5$	$7.5504 * 10^5$

referem-se aos valores mínimos e máximos do respectivo atributo. Já f_i' consiste no valor normalizado.

4.1.3 Coeficientes de Correlação

O resultado dos coeficientes de correlação Pearson e Spearman, quando comparado cada atributo existente com o valor do imóvel, é apresentado na Tabela 4.5 que mostra que a área e quantidade de garagens tem um alto grau de correlação com o valor do imóvel. Também mostram que o valor médio na vizinhança tem mais influência que alguns atributos do imóvel, como por exemplo, o número de quartos.

Tabela 4.5: Coeficientes de relação dos atributos com o valor do imóvel

Ano	Método	Área	Dormitórios	Garagens	Lat	Lon	Valor médio vizinhança
2016	Pearson	0.89	0.36	0.67	0.06	0.23	0.57
	Spearman	0.90	0.47	0.66	0.05	0.26	0.59
2017	Pearson	0.80	0.39	0.45	-0.01	0.01	0.55
	Spearman	0.74	0.33	0.43	0.02	0.24	0.61

Embora o coeficiente de correlação da Latitude e Longitude estejam baixos na Tabela 4.5, estes valores não refletem a realidade, uma vez que é necessária a avaliação da correlação dos dois atributos como uma variável composta. Isto ocorre pois se avaliado separadamente, a Latitude abrange várias regiões da cidade em que a diferença de valores é muito grande entre as regiões. O mesmo ocorre para a Longitude.

4.2 Agrupamento

Visando um melhor entendimento de como as entradas estão distribuídas, foi realizado o agrupamento utilizando os algoritmos *k-means* e DBScan, ambos discutidos na Seção 2.4.

Os resultados obtidos pela aplicação do *k-means*, utilizando $k = 4$, ou seja, 4 grupos, podem ser vistos na Figura 4.1 e 4.2, para os anos de 2016 e 2017, respectivamente . Os grupos ficaram bem divididos. A exceção do ano de 2016 deu-se com o grupo 2, que envolve 7.2% das entradas. Este refere-se aos imóveis de alto padrão, o mesmo ocorre no ano de 2017, porém, estes registros ficaram divididos entre os grupos 1 e 2. Na figura, as barras correspondem ao número de registros presentes em cada grupo.

Foram testados diferentes valores para k em um intervalo entre 3 e 7. No entanto, verificou-se que, utilizando um k = 4 se alcançou os melhores resultados, ou seja, os grupos ficam melhores distribuídos.

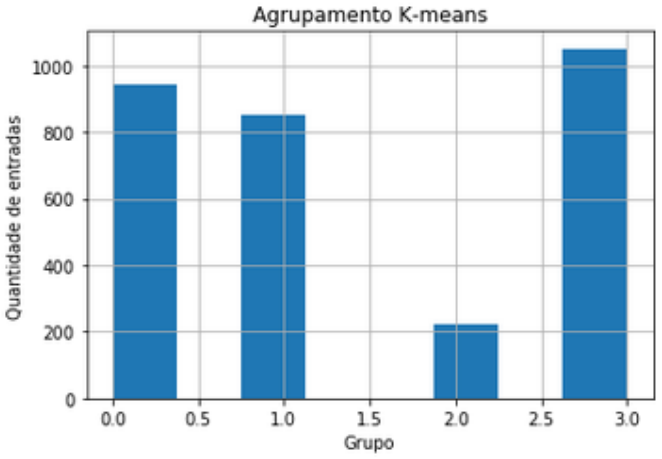


Figura 4.1: Agrupamento dos registros de 2016 utilizando *k-means*.

A Figura 4.3 apresenta a posição geográfica de cada elemento dos grupos obtidos pelo *k-means* no ano de 2016, porém, o ano de 2017 tem o mesmo comportamento. É possível verificar que o grupo dois (que correspondem aos imóveis de alto padrão), em azul, está localizado em sua maioria no centro da cidade. Enquanto os outros 3 grupos estão dispostos por todo o mapa.

Um mapa de calor da cidade, apresentado na Figura 4.4, utiliza como peso o valor do imóvel, onde é visível que a menor quantidade de imóveis (imóveis de alto padrão) é o que representa a maior parcela da renda no setor. Outras regiões, apesar de terem maior número de ofertas tem valores mais baixos, implicando em temperaturas mais baixas no mapa.

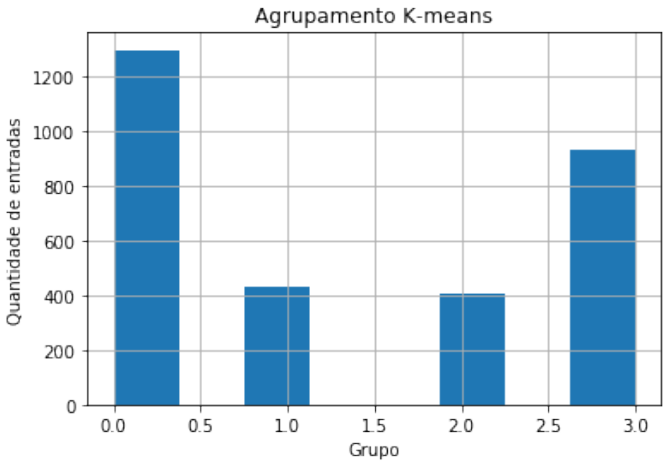


Figura 4.2: Agrupamento dos registros de 2017 utilizando *k-means*.

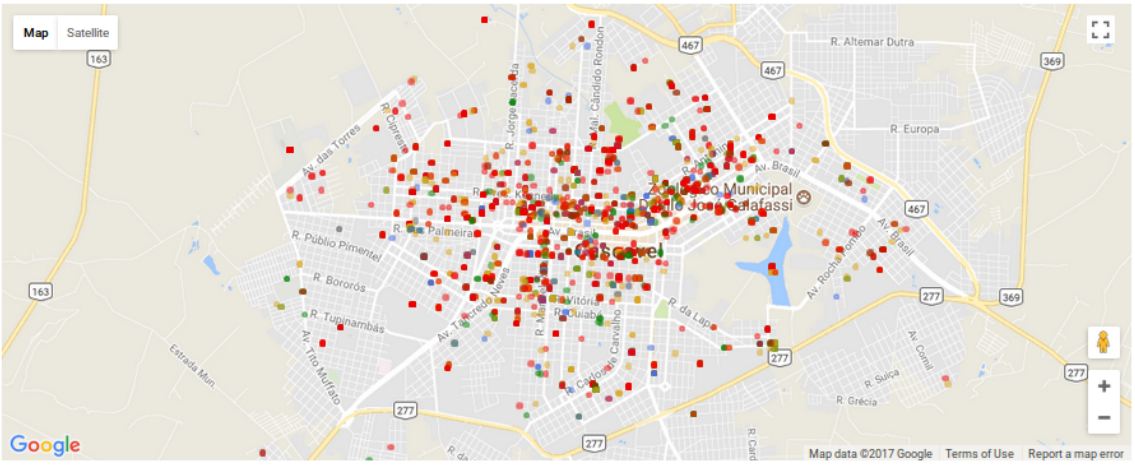


Figura 4.3: Agrupamento dos registros de 2016 utilizando *k-means*.

Para a execução do DBScan, varias combinações de parâmetros (raio e mínimo de vizinhos) foram testados. Testou-se, para o raio, valores entre 0 e 1 e para o mínimo de vizinhos foi utilizado o intervalo entre 0 e 80. Os parâmetros que obtiveram os melhores resultados foi a combinação de: $raio = 0.3$ e $minimo\ de\ vizinhos = 50$. Outras combinações não apresentaram ganho no momento de gerar os grupos, pois formavam muitos grupos pequenos enquanto vários imóveis permaneciam desagrupados ou, em alguns casos, havia a formação de um único grupo contendo a maioria dos elementos.

Os resultados obtidos no método, utilizando os melhores parâmetros, são apresentados na Figura 4.5, onde é possível verificar que 71% das entradas estão agrupadas no *cluster* 1. O grupo -1 diz respeito a todas as entradas que não se encaixaram em nenhum grupo, ou seja,

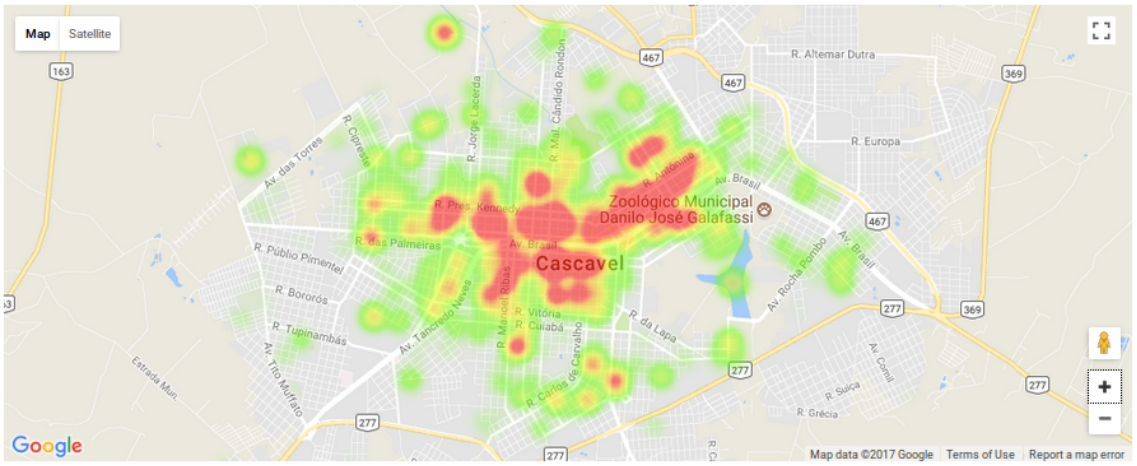


Figura 4.4: *Heatmap* do setor imobiliário na cidade de Cascavel

foram considerados *outliers* pelo algoritmo.

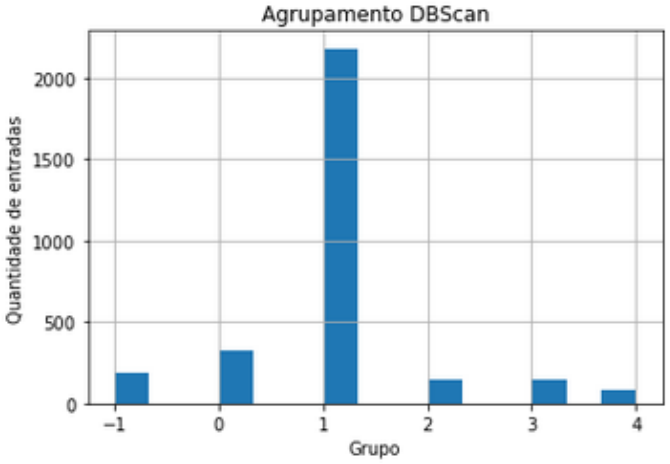


Figura 4.5: Agrupamento utilizando DBScan.

4.3 Parâmetros utilizados nos modelos

A execução da Regressão Linear não necessitou da configuração de nenhum parâmetro já que o mesmo adequa os valores dos coeficientes de acordo com os valores encontrados. Para os métodos de aprendizagem de máquina cada modelo necessita da configuração de seus parâmetros, nesta seção será discutido as configurações utilizadas para a execução dos testes. Os parâmetros utilizados para cada modelo é descrito na tabela 4.6.

Para o modelo SVR o parâmetro kernel corresponde a um kernel radial. O parâmetro gamma representa o quanto um registro na base de dados irá influenciar o treinamento, quanto mais

Tabela 4.6: Parâmetros utilizados

Modelo	Parâmetro	Valor
RNA	Camadas Ocultas	1
	Neurônios	10
	Taxa aprendizado	0.0001
SVR	kernel	rbf
	gamma	0.05
	C	10000
RF	número de estimadores	10
	critério para split	mse

baixo este valor, mais influência terá. E o atributo C do modelo corresponde a penalidade de erro.

O parâmetro número de estimadores do modelo RF é a quantidade de árvores de decisão utilizadas, já o critério para avaliar a qualidade de um split foi o MSE (mean square error).

4.4 Resultados

As médias de todas as execuções das métricas RMSE e MAD, discutidas na seção 3.3, são apresentadas na tabela 4.7. Estes valores correspondem à porcentagem de erro de cada método. Comparando-se os valores alcançados percebeu-se que os métodos de aprendizagem de máquina se mostraram mais precisos (menor taxa de erro) do que o modelo estatístico (Regressão Linear). Quando comparou-se o erro absoluto, o modelo de Random Forest obteve uma taxa de erro foi duas vezes menor que a regressão linear.

Tabela 4.7: Resultados Métricas

Ano	Métrica	RL	RNA	SVR	RF
2016	RMSE	0.1037	0.0815	0.0792	0.0638
	MAD	0.0730	0.0591	0.0584	0.0357
2017	RMSE	0.1198	0.0885	0.0934	0.0712
	MAD	0.0903	0.0645	0.0696	0.0392

De forma a avaliar o desempenho dos métodos no ano de 2016 aplicou-se o teste estatístico de Kruskal-Wallis com um grau de confiança de 95% e grau de liberdade igual a três sobre o erro quadrático médio. Durante a execução do teste, comparou-se o comportamento médio das 32 repetições de cada um dos 4 métodos. Observou-se a rejeição da hipótese nula, ou seja,

existe diferença significativa entre os métodos. Para tanto aplicamos o teste de Mann-Whitney com grau de significância de 5% para comparar os métodos par-a-par.

A comparação dos desempenhos em pares mostrou que todas as quatro estratégias são diferentes entre si. Onde o Random Forest apresentou a melhor precisão, seguido da RNA, SVR e por fim, a regressão linear simples.

Para o ano de 2017 foi executada a mesma sequência de testes estatísticos adotando os mesmos parâmetros. Os resultados obtidos foram similares ao ano de 2016.

4.5 Análise do impacto da inferência

Visando avaliar se a adoção da inferência dos novos dados (informação dos agrupamentos e valor médio da região circunvizinha) nos registros dos imóveis acarretou em ganho de precisão na estimação dos valores, comparou-se a eficiência dos métodos antes e depois da aplicação das informações inferidas. Assim sendo, se confrontou os registros básicos (sem inferência) com os registros utilizando apenas as informações de agrupamentos, apenas a informação do valor médio da vizinhança e as duas informações combinadas. Os valores do erro médio ao longo das 32 repetições são apresentados nas tabelas 4.8 e 4.9, referente aos anos de 2016 e 2017, respectivamente.

Tabela 4.8: Valores médios dos erros refentes ao ano de 2016

	RL	RNA	SVR	RF
Sem inferência	0.1202	0.1164	0.1107	0.0891
Média da vizinhança	0.1072	0.1071	0.1013	0.0785
Agrupamento	0.1103	0.0716	0.0720	0.0587
Ambos	0.1037	0.0815	0.0792	0.0638

Tabela 4.9: Valores médios dos erros refentes ao ano de 2017

	RL	RNA	SVR	RF
Sem inferência	0.1545	0.1329	0.1356	0.0929
Média da vizinhança	0.1271	0.1176	0.1088	0.0848
Agrupamento	0.1254	0.0902	0.0948	0.0698
Ambos	0.1198	0.0885	0.0934	0.0712

Observando-se os resultados apresentados, notou-se que o emprego das informações de agrupamento e valor médio da vizinhança impactou em ganho de precisão no momento de

estimar o valor do imóvel. Tal fato é retratado nas tabelas 4.8 e 4.9 uma vez que estas estratégias apresentaram um menor erro médio nas execuções. No ano de 2016, houve uma melhora de 2% entre a aplicação sem inferência e a aplicação com inferência de ambos os atributos. Já no ano de 2017, nota-se um ganho de 4%.

A inferência do agrupamento mostrou um ganho superior em relação à estimação do valor médio da vizinhança quando utilizando modelos de aprendizagem de máquina, representando uma melhora de 3% para o ano de 2016 e 2% para 2017.

De forma a obter validade estatística aplicou-se o teste de Mann-Whitney com 95% de confiança para os anos de 2016 e 2017 onde comparou-se o valor médio dos erros de cada uma das estratégias adotadas. Os resultados das comparações são apresentadas nas tabelas 4.10 e 4.11, correspondendo aos anos de 2016 e 2017 respectivamente. Nos casos onde observou-se diferença significativa representou-se pela letra "S".

Tabela 4.10: Análise de significância da adoção de inferência no ano de 2016

	RL	RNA	SVR	RF
Agrupamento	S	S	S	S
Média Redondezas	S	S	S	S
Ambos	S	S	S	S

Tabela 4.11: Análise de significância da adoção de inferência no ano de 2017

	RL	RNA	SVR	RF
Agrupamento	S	S	S	S
Média Redondezas	S	S	S	S
Ambos	S	S	S	S

Os valores apresentados nas tabelas 4.8 e 4.9 mostraram que os dados inferidos permitiram uma melhora no processo de estimação. Já as tabelas 4.10 e 4.11 indicam que todos os casos em que se empregou os novos atributos inferidos a melhora foi significativa a 5%.

Capítulo 5

Conclusões

Devido a importância do setor imobiliário na vida dos habitantes e economia de uma região, conseguir estimar o valor de imóveis com agilidade e precisão se faz necessário, servindo de indicador para a inflação e produção da economia. Neste trabalho foram desenvolvidas quatro estratégias para estimação dos valores imobiliários: regressão linear simples, Support Vector Regression, Redes Neurais Artificiais e Random Forest.

Os modelos construídos foram submetidos a um protocolo robusto de forma que os resultados pudessem ser comparados com base estatística.

A análise da média dos erros no processo estimativo mostrou que houve diferença significativa entre as estratégias propostas. Ao comparar-se duas a duas, verificou-se que as estratégias de aprendizagem de máquina puderam alcançar uma precisão superior à regressão linear, e, dentre estas, a abordagem de Random Forest foi a que mostrou-se mais acurada.

Durante os experimentos foram inferidas novas informações aos registros dos imóveis de forma a tentar melhorar a precisão do processo estimativo. Visando avaliar se tais informações trouxeram contribuição fez-se a comparação com os resultados "puros"iniciais. Verificou-se que para todas as estratégias foi vantajoso utilizar as informações de agrupamento e preço médio da vizinhança na estimação dos valores dos imóveis.

As técnicas de aprendizado de máquina se apresentaram melhores devido ao fato de serem mais robustas, então como trabalho futuro terá a análise com modelos de regressão mais avançados.

As técnicas empregadas no trabalho não envolvem características espaciais, como ocorre nos métodos SAR e CAR. Acredita-se, no entanto, que tais abordagens possam contribuir no aumento da precisão da estimação. Para tanto, pretende-se, futuramente, avaliar a adoção do

SAR e CAR. Além disso, pode ser interessante adicionar mais informações aos registros dos imóveis, tais como a distância para colégios, escolas, hospitais, taxa de criminalidade nas redondezas e etc.

Referências Bibliográficas

ALBUQUERQUE, P. *Prorum*. 2016. Consultado na INTERNET: http://prorum.com/index.php/2051/como-funcionam-os-modelos-de-support-vector-regression, 2017.

ALLEN, D. M. Mean square error of prediction as a criterion for selecting variables. *Technometrics*, Taylor & Francis Group, v. 13, n. 3, p. 469–475, 1971.

ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, v. 46, n. 3, p. 175–185, May 1992.

AUGSPURGER CHRIS BARTAK, P. C. A. H. T. *Pandas*. 2017. Consultado na INTERNET: http://pandas.pydata.org/, 2017.

BASAK, D.; PAL, S.; PATRANABIS, D. C. Support vector regression. *Neural Information Processing-Letters and Reviews*, v. 11, n. 10, p. 203–224, October 2007.

BIRANT, D.; KUT, A. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, Elsevier, turkey, v. 60, n. 1, p. 208–221, January 2007.

BOURASSA, S. C.; CANTONI, E.; HOESLI, M. Spatial dependence, housing submarkets, and house price prediction. *The Journal of Real Estate Finance and Economics*, Springer, Geneva, v. 35, n. 2, p. 143–160, June 2007.

ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Germany: AAAI Press, 1996. p. 226–231.

FILHO, D. B. F.; JUNIOR, J. A. S. Desvendando os mistérios do coeficiente de correlação de pearson (r). *Revista Política Hoje-ISSN: 0104-7094*, Pernambuco, v. 18, n. 1, p. 115–146, January 2010.

FRALEY, C.; RAFTERY, A. E. How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, Oxford University Press, Seattle, WA, v. 41, n. 8, p. 578–588, January 1998.

HARTIGAN, J. A.; WONG, M. A. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, JSTOR, New Haven, USA, v. 28, n. 1, p. 100–108, 1979.

KHAMIS, A. B.; KAMARUDIN, N. K. K. B. Comparative study on estimate house price using statistical and neural network model. *International Journal of Scientific & Technology Research*, v. 3, n. 12, p. 126–131, December 2014.

KUŞAN, H.; AYTEKIN, O.; ÖZDEMIR, İ. The use of fuzzy logic in predicting house selling price. *Expert systems with Applications*, Elsevier, Turkey, v. 37, n. 3, p. 1808–1813, March 2010.

LI, Y.; LEATHAM, D. J. et al. Forecasting housing prices: dynamic factor model versus lbvar model. In: *Paper for presentation at the Agricultural & Applied Economics Association's 2011 AAEA & NAREA Joint Annual Meeting, Pittsburgh, Pennsylvania*. Pittsburgh, Pennsylvania: [s.n.], 2011.

LINDEN, R. Técnicas de agrupamento. *Revista de Sistemas de Informação da FSMA*, Rio de Janeiro, v. 4, n. 4, p. 18–36, Dezembro 2009.

MITCHEL, T. M. *Machine Learning: An artificial intelligence approach*. 1. ed. New York: McGraw-Hill Science/Engineering/Math, 1997.

NETO, A. N. Preços hedônicos. *Economics*, São Paulo, v. 58, n. 3, p. 504–510, Dezembro 1976.

PAGOURTZI, E. et al. Real estate appraisal: a review of valuation methods. *Journal of Property Investment & Finance*, MCB UP Ltd, v. 21, n. 4, p. 383–401, 2003.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, n. 12, p. 2825–2830, October 2011.

PETERSON, L. E. K-nearest neighbor. *Scholarpedia*, v. 4, n. 2, p. 1883, 2009.

POW, N.; JANULEWICZ, E.; LIU, L. *Applied Machine Learning Project 4 Prediction of real estate property prices in Montréal*. 2014. Consultadona INTERNET: http://rl.cs.mcgill.ca/comp598/fall2014/comp598_submission_99.pdf, 2017.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 1. ed. Englewood Cliffs, New Jersey 07632: Prentice Hall, 2003.

SEBER, G. A.; LEE, A. J. *Linear regression analysis*. 2. ed. New Jersey: John Wiley & Sons, 2012.

SHWARTZ, S.; DAVID, S. *Understanding Machine Learning From Theory to Algorithms*. 1. ed. New York: Cambrigde University Press, 2014.

SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and computing*, Springer, Netherlands, v. 14, n. 3, p. 199–222, November 2004.

SPEARMAN, C. Spearman's rank correlation coefficient. *Amer J Psychol*, v. 15, n. 1, p. 72–101, January 1904.

TAN, P.-N. et al. *Introduction to data mining*. [S.l.]: Pearson Education India, 2006.