

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA

GUILHERME JUNIO SILVA PASSOS

**APLICANDO APRENDIZADO DE MÁQUINA NA PREDIÇÃO DOS
PREÇOS DE APARTAMENTOS DA CIDADE DE BELO HORIZONTE**

Belo Horizonte
Dezembro de 2018

GUILHERME JUNIO SILVA PASSOS

**APLICANDO APRENDIZADO DE MÁQUINA NA PREDIÇÃO DOS
PREÇOS DE APARTAMENTOS DA CIDADE DE BELO HORIZONTE**

Monografia apresentada durante o seminário dos trabalhos de conclusão do curso de graduação em Engenharia Elétrica da UFMG, como parte dos requisitos necessários à obtenção do título de Engenheiro Eletricista.

Orientador: **Prof. Dr. Cristiano Leite de Castro**
Universidade Federal de Minas Gerais

Belo Horizonte
Dezembro de 2018

Dedico este trabalho a Deus, à causalidade dos eventos do universo que nos trazem ao momento presente e todas as pessoas que amei; em especial Mãe, Pai e Nina.

AGRADECIMENTOS

Os agradecimentos principais são direcionados para:

- A minha família, em especial minha mãe e irmã, pelo enorme suporte dado à mim ao longo do meu curso de graduação.
- O Professor Dr. Cristiano Leite de Castro pela disposição em transferir parte do seu conhecimento de maneira paciente e clara.
- A Universidade Federal de Minas Gerais por ter me proporcionados ao longo de sete anos conhecer pessoas incríveis e vivenciar experiências intensas, tanto positivas quanto negativas.

"Ninguém pode se sentir satisfeito enquanto ainda houver crianças, milhões de crianças, que não recebem uma educação que lhes ofereça dignidade e o direito de viver suas vidas completamente. (Nelson Mandela, 2005)"

RESUMO

Este trabalho tem como objetivo realizar uma análise de dados dos preços de apartamentos da cidade de Belo Horizonte e desenvolver um modelo computacional inteligente que seja de prever tais preços. Para tal, obteve-se um conjunto de dados composto por mais de 60 mil anúncios de imóveis entre os meses de agosto e outubro de 2018. Assim, testou-se a eficiência de modelos de aprendizado de máquina baseados em métodos meramente estatísticos, aproximações biológicas e árvores de decisão. A avaliação do desempenho dos modelos escolhidos será apresentada no Capítulo 5. Ressalta-se que a análise de dados realizada neste trabalho é atemporal e desconsidera a variação dos preços dentro da janela de tempo observada. Por fim, os resultados finais obtidos foram satisfatoriamente validados e demonstraram que a modelagem desenvolvida neste trabalho é útil ao setor imobiliário de qualquer cidade brasileira.

Palavras-chave: Aprendizado de Máquina. Regressão. Precificação Imobiliária.

ABSTRACT

This present project aims at making a data analysis of Belo Horizonte apartment prices and creating a computer model that is capable of predicting these prices. To do so, over 60 thousand housing ads were collected from August to October of 2018. Then, supervised training was performed on regression algorithms of machine learning that are based on statistic methods, biological systems and decision trees. The assessment of these algorithms is presented in chapter 5. Note that the data analysis done in this project is timeless and ignores the variation of housing prices within time window previously cited. The final results were successfully validated and they prove that this approach is able to generate useful predictions for the housing market of any Brazilian city.

Keywords: Machine Learning. Regression. Predicting Housing Prices.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Curvas de demanda e oferta da função de preços hedônicos. Fonte: Follain e Jimenez (1985) - página 79 | 6 |
| Figura 2 – Gráfico de predição dos preços futuros de imóveis. Fonte: (NG, 2015) - página 36 | 14 |
| Figura 3 – Identificação de áreas da cidade Londres que estão dentro do orçamento desejado. Fonte: (NG, 2015) | 15 |
| Figura 4 – Resultado da pesquisa das áreas que se enquadram dentro do orçamento desejado. Fonte: (NG, 2015) | 15 |
| Figura 5 – Representação de um neurônio artificial | 21 |
| Figura 6 – Função linear | 22 |
| Figura 7 – Função tangente hiperbólica | 22 |
| Figura 8 – Função sigmoidal | 22 |
| Figura 9 – Função unidade linear retificada | 22 |
| Figura 10 – Exemplo de previsão com um neurônio artificial | 23 |
| Figura 11 – Uma rede neural com uma camada escondida | 25 |
| Figura 12 – Aplicação de uma rede neural com uma camada escondida em um problema de previsão | 25 |
| Figura 13 – Partição do Espaço do Amostral do Problema de Análise de Óleo Mineral. Fonte: (LADEIRA et al., 2017) | 27 |
| Figura 14 – Obtenção do objeto JSON do site da Rede Net Imóveis | 31 |
| Figura 15 – Objeto JSON contendo as informações dos imóveis | 32 |
| Figura 16 – Script usado para se extrair os dados do portal da Rede NetImóveis automaticamente | 32 |
| Figura 17 – Distribuição dos preços dos apartamentos por área e região | 34 |
| Figura 18 – Alguns <i>outliers</i> mostrados na Figura 17 | 35 |
| Figura 19 – Matriz de correlação das variáveis do conjunto de dados obtido | 36 |
| Figura 20 – Quantidade de amostras que possuem valor do IPTU ou valor do condomínio igual a zero | 36 |
| Figura 21 – Distribuição geográfica dos dados no território da cidade divididos por região | 37 |
| Figura 22 – Distribuição geográfica dos dados divididos em 50 regiões | 38 |

| | |
|---|----|
| Figura 23 – Matriz de correlação | 38 |
| Figura 24 – Distribuição amostral dos preços dos apartamentos | 39 |
| Figura 25 – Distribuição amostral dos preços normalizada | 40 |
| Figura 26 – Conceito de distribuições assimétricas | 40 |
| Figura 27 – Avaliação do coeficiente de assimetria das distribuições das variáveis inde- pendentes | 41 |
| Figura 28 – Coeficiente de assimetria das distribuições das variáveis dependentes após a transformação Box-Cox | 42 |
| Figura 29 – Ilustração do procedimento <i>k-fold Cross Validation</i> | 43 |
| Figura 30 – Parametrização do modelo de regressão linear | 45 |
| Figura 31 – Parametrização do modelo lasso | 45 |
| Figura 32 – Parametrização do modelo regressão ridge | 45 |
| Figura 33 – Parametrização da rede de perceptrons de multicamadas | 45 |
| Figura 34 – Parametrização do modelo <i>gradient tree boosting</i> | 45 |
| Figura 35 – Parametrização do modelo <i>xtreme gradient tree boosting</i> | 46 |
| Figura 36 – Parametrização do modelo <i>light gradient tree boosting</i> | 46 |
| Figura 37 – Distribuição do erro médio percentual absoluto por experimento | 47 |
| Figura 38 – Ilustração do procedimento de empilhamento de modelos | 49 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Variáveis utilizadas por NG para treinamento do modelo | 13 |
| Tabela 2 – Comparação da quantidade percentual de amostras que foram precificadas dentro da margem de $\pm 10\%$ e $\pm 20\%$ em relação ao preço real. Fonte: AaronNg - Página 41 | 16 |
| Tabela 3 – Exemplo de um conjunto de dados para aprendizado supervisionado de um neurônio | 23 |
| Tabela 4 – Relação dos dados obtidos por extração | 33 |
| Tabela 5 – Resumo dos resultados do experimentos | 48 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|-------------|--|
| ABNT | Associação Brasileira de Normas Técnicas |
| NBR | Norma Brasileira |
| RNN | Redes Neurais Recorrentes (do inglês: <i>Recurrent Neural Network</i>) |
| FPH | Função de Preço Hedônico |
| MMQ | Método dos Mínimos Quadrados |
| SQE | Soma dos Quadrados dos Resíduos |
| MLP | Rede de Perceptrons de Multicamadas (do inglês: <i>Multi Layer Perceptron</i>) |
| GBoost | <i>Gradient Tree Boosting Machine</i> |
| XGBoost | <i>Xtreme Gradient Tree Boosting Machine</i> |
| LightGBoost | <i>Light Gradient Tree Boosting Machine</i> |
| JSON | <i>JavaScript Object Notation</i> |
| API | <i>Application Programming Interface</i> |
| IPTU | Imposto sobre a Propriedade Predial e Territorial Urbana |
| MSE | Erro Quadrático Médio (do inglês: <i>Mean Squared Error</i>) |
| MAPE | Erro Médio Percentual Absoluto (do inglês: <i>Mean Absolute Percentage Error</i>) |
| FIPE | Fundação Instituto de Pesquisas Econômicas |

LISTA DE SÍMBOLOS

| | |
|---------------|---|
| ϕ | Função de base do modelo de regressão |
| β | Parâmetros de ajuste do modelo de regressão |
| ε | Erro residual |
| λ | Parâmetro de ajuste dos modelos de regressão ridge, lasso e <i>gradient tree boosting</i> |
| ∇ | Operador diferencial |
| σ | Desvio padrão |
| μ | Média ou valor esperado |
| γ | Coefficiente de assimetria |
| ρ | Coefficiente de transformação Box-Cox |

SUMÁRIO

| | |
|---|-----------|
| 1 – INTRODUÇÃO | 1 |
| 1.1 Organização do trabalho | 3 |
| 2 – REVISÃO BIBLIOGRÁFICA: MANEIRAS DE SE PRECIFICAR IMÓVEIS | 4 |
| 2.1 Modelo de precificação hedônica de Rosen | 4 |
| 2.1.1 Formalização do método | 5 |
| 2.1.2 Críticas ao modelo | 6 |
| 2.1.3 Considerações do autor | 7 |
| 2.2 Métodos de avaliação de imóveis urbanos da ABNT NBR 14653 | 7 |
| 2.2.1 Métodos para identificar o valor de um bem | 8 |
| 2.2.1.1 Método comparativo de dados de mercado | 8 |
| 2.2.1.2 Método involutivo | 9 |
| 2.2.1.3 Método evolutivo | 9 |
| 2.2.1.4 Método da capitalização da renda | 10 |
| 2.2.2 Métodos para identificar o custo de um bem | 10 |
| 2.2.2.1 Método da quantificação do custo | 10 |
| 2.2.2.2 Método comparativo direto de custo | 12 |
| 2.2.3 Considerações do autor | 12 |
| 2.3 Monografia DE NG - <i>Machine Learning For A London House Price Prediction Mobile Application</i> | 12 |
| 2.3.1 Objetivos | 12 |
| 2.3.2 Tratamentos dos dados | 13 |
| 2.3.3 Treinamento do modelo | 13 |
| 2.3.4 Resultados | 14 |
| 2.3.5 Considerações do autor | 16 |
| 3 – FUNDAMENTAÇÃO TEÓRICA: APRENDIZADO DE MÁQUINA | 17 |
| 3.1 Introdução ao aprendizado de máquina | 17 |
| 3.2 Regressão linear | 18 |
| 3.2.1 Ajuste do modelo de regressão linear | 19 |

| | | |
|----------|---|-----------|
| 3.3 | Regressão ridge | 19 |
| 3.4 | Lasso | 20 |
| 3.5 | Redes neurais artificiais | 21 |
| 3.5.1 | Conceito básico de um neurônio artificial | 21 |
| 3.5.2 | Processo de aprendizado de um neurônio artificial | 23 |
| 3.5.3 | Redes de perceptrons de múltiplas camadas | 24 |
| 3.6 | Modelos baseados em árvores de decisão | 26 |
| 3.6.1 | Gradient tree boosting machine | 27 |
| 3.6.2 | Análise crítica do método gradient tree boosting | 29 |
| 4 | METODOLOGIA | 31 |
| 4.1 | Obtenção dos dados | 31 |
| 4.2 | Extrações | 33 |
| 4.3 | Análise e transformação dos dados | 34 |
| 4.3.1 | Outliers | 34 |
| 4.3.2 | Correlação entre as variáveis do problema | 35 |
| 4.3.3 | Agregando um significado quantitativo à distribuição geográfica dos imóveis | 37 |
| 4.3.4 | Normalização das distribuições dos dados | 39 |
| 4.4 | Definindo uma estratégia de <i>cross-validation</i> | 42 |
| 4.4.1 | Metodologia de avaliação dos modelos | 44 |
| 4.5 | Parametrização dos modelos | 44 |
| 5 | RESULTADOS E DISCUSSÃO | 47 |
| 5.1 | Resultados dos experimentos | 47 |
| 5.2 | Sugestão de implementações que podem obter melhores resultados | 48 |
| 6 | CONCLUSÕES | 50 |
| 6.1 | Possíveis aplicações e futuros trabalhos | 50 |
| | Referências | 51 |

1 INTRODUÇÃO

O mercado imobiliário é um dos segmentos que mais influenciam a economia de uma nação (HARVEY; JOWSEY, 2004). Contudo, esse setor possui uma dinâmica própria e complexa devido ao seu produto, o imóvel, ser composto de uma quantidade significativa de características meramente subjetivas, como por exemplo a posição das janelas de um apartamento em relação ao pôr do sol ou o tipo de atividade e comportamento da vizinhança do imóvel. Além disso, soma-se a esses atributos abstratos as características espaciais (localização) e estruturais do imóvel, bem como as depreciações e valorizações que esse pode sofrer ao longo do tempo por fatores próprios ou externos. Deste modo, a avaliação imobiliária tanto micro quanto macro-espacial é uma tarefa socialmente importante, engenhosa e que atrai há anos a atenção de muitos pesquisadores no mundo inteiro. Neste trabalho ir-se-á examinar detalhadamente o problema da precificação imobiliária e apresentar o processo de desenvolvimento de uma ferramenta computacional capaz de resolvê-lo.

De acordo com (XIAO, 2016), os métodos mais comuns de avaliação de preços imóveis podem ser divididos em dois grupos: métodos tradicionais e métodos avançados. O grupo dos tradicionais é formado pelos seguintes métodos: método comparativo (comparação), método do contratante (método de custo), método residual (método de desenvolvimento), método dos lucros e método de investimento. O estudo desses cinco métodos, porém, foge do escopo deste presente trabalho. Os métodos avançados de precificação imobiliária segundo XIAO incluem as técnicas de modelagem hedônica, redes neurais recorrentes (RNN), raciocínio baseado em casos (do inglês: *case-based reasoning*) e métodos de análise espacial. Contudo, a modelagem hedônica é o método mais utilizado desses considerados avançados.

A teoria da modelagem de precificação hedônica, formalizada inicialmente por Rosen (1974), busca estimar o valor de um bem baseando-se na primícia de que o preço desse é determinado pela combinação do valor de suas características internas e os fatores externos que o afetam. Assim, a modelagem de precificação hedônica é frequentemente utilizada para estimar quantitativamente valores para o ecossistema e os serviços do meio ambiente que impactam diretamente nos preços de imóveis. Por exemplo, o preço de um apartamento pode ser determinado em função de suas características estruturais (*ex: detalhes arquitetônicos, área construída, estado de conservação, quantidade de cômodos e acabamento físico do imóvel*), bem como pelas características do ambiente a sua volta (*ex: índice de violência e criminalidade do*

bairro, acessos à região central da cidade, proximidade de escolas e hospitais, nível de poluição sonora da região e preços dos imóveis adjacentes). Rothenberg et al. (1991) destaca em seu livro duas vantagens significativas dessa modelagem em relação aos métodos alternativos de precificação. Primeiramente, ele diz que a modelagem hedônica possui a capacidade de mensurar de maneira homogênea várias características do imóvel em uma única dimensão e, em segundo lugar, ROTHENBERG et al. cita que a modelagem hedônica é capaz de mensurar as margens de perda e ganho que vendedores e compradores fazem a respeito das características do mercado no momento de uma transação. Entretanto, a modelagem hedônica possui algumas limitações que serão discutidas no capítulo Capítulo 2 deste trabalho. Por exemplo, essa abordagem quantifica apenas a vontade que compradores tem de pagar sobre aquilo que eles enxergam como qualidade do ambiente em torno do imóvel. Em outras palavras, modelos hedônicos partem do pressuposto que as partes envolvidas numa transação imobiliária estão cientes de todos os aspectos positivos e negativos relacionados ao bem transicionado, o que na prática é impossível. Ou seja, se um comprador em potencial não estiver ciente do nível de poluição atmosférica na região do seu futuro imóvel, ou se um vendedor desconhecer que há uma possibilidade iminente de se construir um shopping nas adjacências de seu imóvel, a abordagem hedônica falha. Além disso, modelos hedônicos nem sempre incorporam no seu processo de regressão fatores externos como regulações governamentais, valorizações nos preços por escassez de demanda e impostos que afetam o imóvel.

Partindo desse ponto, este trabalho pretende desenvolver um modelo computacional inteligente capaz de aprender por indução as diversas particularidades e vieses que compõem os preços de apartamentos da cidade de Belo Horizonte. Assim, espera-se que as características internas e externas dos apartamentos, bem como os fatores intrínsecos do mercado imobiliário de belo horizonte, sejam devidamente quantificadas ao se treinar modelos de aprendizado de máquina com uma base de dados contendo os preços de vários apartamentos e algumas dezenas de suas características. O termo **aprendizado de máquina** é recente, mas seu desenvolvimento teórico se deu entre as décadas de 60 e 90 do século passado. Durante esse período, diversos trabalhos científicos desenvolveram algoritmos eficientes capazes de aprender a reconhecer padrões e fazer previsões a partir de dados. Destaca-se o algoritmo de *back propagation* para ajuste de pesos da rede neural de multicamadas de Rumelhart, Hinton e Williams (1986) e a formalização matemática das *support vector machines* de Vapnik (1995). Porém, a aplicação de aprendizado de máquina em problemas do mundo real se intensificou apenas na segunda metade da década de 2000 devido à evolução da capacidade computacional e o aumento da quantidade

de dados (informações) na forma digital.

Este trabalho conta com uma quantidade razoável de dados extraídos de portais de anúncios de imóveis da cidade de Belo Horizonte. Em dois meses de coleta, foi possível obter mais de 60 mil dados de imóveis contendo informações como preço, área interna, endereço, quantidade de quartos, suítes, vagas de garagens, valor do condomínio entre outros. O conjunto desses dados descreve com muita precisão as características do mercado imobiliário belo horizontino. Assim, pretende-se conciliar essas informações com o que as ferramentas disponíveis de aprendizado de máquina e *statistical learning* para se desenvolver um modelo preditivo capaz de estimar preços de apartamentos de Belo Horizonte.

1.1 Organização do trabalho

Este trabalho está dividido em 6 capítulos e este primeiro capítulo introduz o assunto a ser abordado.

No Capítulo 2 analisa-se trabalhos acadêmicos e normas técnicas que se destinam a fornecer métodos, ferramentas e soluções para o problema de precificação imobiliária.

No Capítulo 3 é realizada uma revisão do princípio teórico dos seguintes modelos de aprendizado de máquina: *ridge regression*, *lasso*, *redes de perceptrons de múltiplas camadas* e *gradient tree boosting*.

No Capítulo 4 apresenta-se detalhadamente a metodologia do processo de obtenção, análise e transformação dos dados utilizados no trabalho.

No Capítulo 5 os modelos descritos no Capítulo 2 são testados e seus resultados são devidamente avaliados e comparados com outros trabalhos.

Por fim, as conclusões são feitas no Capítulo 6.

2 REVISÃO BIBLIOGRÁFICA: MANEIRAS DE SE PRECIFICAR IMÓVEIS

Neste capítulo serão apresentados alguns modelos e métodos de precificação imobiliária que são utilizados na prática com menor ou maior frequência, a fim de mostrar as diferentes maneiras já implementadas de se resolver o problema aqui proposto. Ir-se-á apresentar de maneira detalhada o modelo de precificação hedônica desenvolvido por Rosen (1974); os métodos de avaliação de imóveis urbanos apresentado na ABNT NBR 14653-1 e a monografia de Ng (2015).

2.1 Modelo de precificação hedônica de Rosen

O termo hedônico advém do hedonismo, uma doutrina filosófico-moral que afirma ser o prazer o bem supremo da vida humana. A fundamentação teórica da modelagem hedônica em econometria é atribuída à Lancaster (1966) e está apresentada no seu artigo *A New Approach to Consumer Theory*. LANCASTER supõe que consumidores adquirem um produto baseando-se na quantidade de características deste bem e no valor individual de cada uma dessas características. Em outras palavras, um bem pode ser interpretado como um "pacote de atributos" no qual o preço deste é determinado pela forma que esses atributos são valorados. Por exemplo, de acordo com LANCASTER, as pessoas ao escolherem um carro avaliam na verdade o conjunto de características do carro, como aceleração, itens de segurança, design, prestígio, preço de revenda, entre outras, e determinam o quanto estão dispostas a pagar por essas características. Como consequência da modelagem hedônica, espera-se que consumidores com maiores rendas adquirirão uma maior quantidade de características e vice-versa; assim, gera-se uma natural segmentação de mercado, no qual pessoas com capacidade de compra semelhantes adquirem produtos com características parecidas.

Embora LANCASTER foi o primeiro a discutir a utilidade hedônica dos produtos, foi ROSEN que formolou matematicamente um método de precificação imobiliária baseado na abordagem hedônica. ROSEN argumentou que um item pode ser avaliado em função de suas características e o preço total deste item é a soma de cada um dos seus atributos homogêneos. Como cada atributo possui um preço implícito único no equilíbrio do mercado, o preço de um item pode ser avaliado por meio da regressão de suas características e, assim, determina-se o quanto cada característica contribui para o preço total do produto. Para a formalização do seu método, ROSEN assume que é possível determinar uma função $p(z)$ que descreve o **equilíbrio**

do mercado entre as quantidades demanda e ofertada de um produto. Ou seja, no equilíbrio de mercado o preço de cada atributo coincide perfeitamente com o valor que um comprador deseja pagar por ele, não havendo assim inter-relação entre o preço de cada uma das variáveis explicativas que correspondem aos atributos. Devido à esse limite teórico, pode-se afirmar que o método de precificação hedônica de ROSEN constitui um modelo de economia neoclássica ¹.

2.1.1 Formalização do método

ROSEN propõe em seu trabalho um procedimento constituído de duas etapas para determinação do preços dos atributos que compõe um imóvel. Inicialmente, estima-se de maneira empírica a função de preço hedônico (FPH) para uma unidade imobiliária i , utilizando o preço de vários imóveis nas adjacências desta, conforme a Equação (1) abaixo:

$$P_{hi} = P_h(S_{i1}, \dots, S_{ij}, \dots, N_{i1}, \dots, N_{ik}, \dots, Q_{i1}, \dots, Q_{im}) \quad (1)$$

onde S_j , N_k e Q_m representam, respectivamente, o vetor contendo o preço das características do lote, bairro e ambiente do imóvel. O processo para estimar a Equação (1) envolve a aplicação de uma série de técnicas de modelagem estatística. Por motivos de simplificação, representa-se por Z o conjunto de atributos dos vetores (S_j , N_k e Q_m) utilizados no modelo empírico. Assim, a representação empírica do preço do i -ésimo imóvel é dada pela Equação (2).

$$p_i(z) = p(Z_i, \omega, \varepsilon) \quad (2)$$

onde ω é o vetor de parâmetros a ser estimado, ε é um termo residual estocástico e p_i é o preço implícito do imóvel com relação àquelas características. Uma das consequências do equilíbrio econômico de mercado entre compradores e vendedores, discutido na Seção 2.1, é a possibilidade de expressar a Equação (2) por meio das funções que descrevem a quantidade ofertada $Q^s(z)$ e a quantidade demandada $Q^d(z)$ de produtos. Assim, na situação de equilíbrio temos:

$$p_i(z_i) = Q_i^s(z_i) = Q_i^d(z_i) \quad (3)$$

Uma vez estimada a FPH, a determinação do preço implícito da k -ésima característica, z_k para cada um dos compradores e vendedores do imóvel é realizada computando-se a derivada parcial da FPH em relação à essa k -ésima característica, ou seja,

$$p_k = \frac{\partial p_i(z_i)}{\partial z_{ik}} \quad (4)$$

¹https://pt.wikipedia.org/wiki/Economia_neoclássica

A segunda etapa do método de ROSEN consiste em utilizar os preços p_k marginais estimados na Equação (5) como variáveis dependentes para a estimação das equações do modelo. Assim, a função $p(z)$ de ROSEN é uma curva envoltória das curvas de demandas e oferta que pode assumir a forma côncava ou convexa conforme está mostrado na Figura 1. A forma côncava representa um problema na formalização do fenômeno baseado na maximização dos consumidores, dado que o budget set é côncavo. Em estudos empíricos, a função hedônica assume majoritariamente a forma convexa, porém nada impede que seja linear, já que isto ocorre quando o lambda da transformação de Box-Cox apresenta valor igual a 1 (FAVERO; BELFIORE; LIMA, 2008).

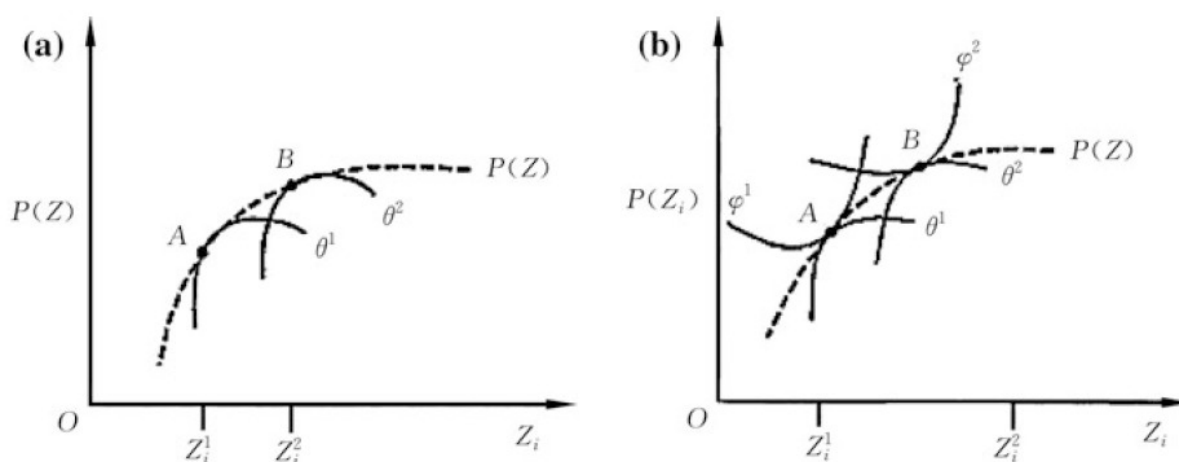


Figura 1 – Curvas de demanda e oferta da função de preços hedônicos. Fonte: Follain e Jimenez (1985) - página 79

2.1.2 Críticas ao modelo

É necessário ressaltar que o modelo de precificação hedônica proposto por ROSEN recebeu críticas de diversos pesquisadores da área de econometria ao longo dos anos. A maior crítica ao modelo de precificação hedônica é a suposição principal do equilíbrio de mercado. Para essa suposição ser verdadeira é necessário que haja equidade de informações, entre compradores e vendedores, a respeito do imóvel e que os custos de transação sejam nulos (MADDISON, 2001). Se a situação de equilíbrio não ocorre, a determinação implícita dos preços marginais de cada atributo se torna enviesada. Ademais, Freeman (1993) alega que uma das consequências do desequilíbrio de mercado é o provável aumento na variância dos resultados ao invés de um sistemático enviesamento do modelo.

De acordo com Bitros e Panas (1988), a estimação hedônica não é fruto de uma interação entre oferta e demanda. Por ser de natureza hedônica, Bitros e Panas (1988) afirmam que a

decisão de um consumidor individualmente não afeta a FPH, ou seja, a decisão de um comprador não pode influenciar a precificação realizada pelos vendedores.

Segundo Follain e Jimenez (1985), o preço marginal de uma característica derivado da FPH não mede de fato a vontade de um comprador em específico pagar por aquele atributo; pelo contrário, a precificação dos atributos é resultado das interações de oferta e demanda no mercado como um todo.

Uma outra crítica levantada por Freeman (1979) ao modelo de precificação hedônica de ROSEN é a velocidade de ajuste do mercado para mudanças ocorridas nas condições de oferta e demanda. Se o ajuste não for completo, o preço implícito de uma característica não irá medir precisamente o quanto compradores estão dispostos a pagar por aquele atributo. Quando a demanda por um atributo específico do imóvel aumenta, o preço marginal deste atributo irá subestimar o quanto realmente se deseja pagar por ele. Pois, a disposição dos compradores em pagar por esse atributo não será traduzida nas transações do mercado que afetam o preço implícito marginal deste enquanto o potencial de ganho da propriedade não ultrapassar o limiar de transações e custos móveis.

2.1.3 Considerações do autor

Pode-se concluir, portanto, que o modelo de precificação hedônica baseia-se em técnicas de regressão lineares que, por sua vez, receberam críticas de vários autores devido à uma série de problemas de econometria tais como: heterogeneidade espacial, autocorrelação espacial, mudanças na qualidade do imóvel, multicolinearidade e heteroscedasticidade. Xiao (2016) analisa esses problemas com mais profundidade no capítulo 2 de seu livro.

2.2 Métodos de avaliação de imóveis urbanos da ABNT NBR 14653

A Associação Brasileira de Normas Técnicas (ABNT) apresenta na Norma Brasileira 14653 (NBR 14653) seis métodos para se avaliar imóveis urbanos divididos em dois grupos principais. Eles são:

- Métodos para identificar o valor de um bem
 - Método comparativo de dados de mercado;
 - Método involutivo;
 - Método evolutivo;
 - Método da capitalização da renda;

- Métodos para identificar o custo de um bem
 - Método da quantificação do custo;
 - Método comparativo direto de custo;

A escolha do método é realizada considerando-se a natureza do bem avaliado, da finalidade da avaliação e da disponibilidade, qualidade e quantidade de informações colhidas no mercado.

2.2.1 Métodos para identificar o valor de um bem

Os métodos descritos nesta subsecção não se aplicam aos empreendimentos de base imobiliária, tais como hotéis, *shopping centers*, entre outros.

2.2.1.1 Método comparativo de dados de mercado

Este método consiste-se, inicialmente, em reunir a maior quantidade de informações representativas de imóveis com características, tanto quanto possível, semelhantes às do avaliado, usando-se toda a evidência disponível. Em uma segunda etapa, classifica-se as informações obtidas em variáveis dependentes e variáveis independentes. As variáveis dependentes referem-se aqueles dados que são influenciados pela conduta e às formas de expressão dos preços no mercado (por exemplo, informações a respeito de negociações realizadas e ofertas contemporâneas à data de referência da avaliação unitário, moeda de referência, formas de pagamento, entre outros). As variáveis independentes referem-se às características físicas do imóvel (por exemplo: área, frente), de localização (como bairro, logradouro, distância ao polo de influência, entre outros) e econômicas (como oferta ou transação, época e condição do negócio – à vista ou a prazo). Ressalta-se que as variáveis dependentes devem ser escolhidas com base em teorias existentes, conhecimentos adquiridos, senso comum e outros atributos que se revelem importantes no decorrer dos trabalhos, pois algumas variáveis consideradas no planejamento da pesquisa podem se mostrar pouco relevantes e vice-versa.

A etapa final do método baseia-se em organizar o conjunto de informações obtidas sob a forma de gráficos que mostrem as distribuições de frequência para cada uma das variáveis, bem como as relações entre elas. Nesta etapa, identificam-se as variáveis que mais influenciam o preço do imóvel, a forma de variação, possíveis dependências entre elas e *outliers*. Por fim, o(a) engenheiro(a) de vendas escolhe o método científico a ser utilizado para inferir o comportamento do mercado e formação de valores. As ferramentas analíticas para a indução do comportamento do mercado sugeridas pela NBR 14653 são: redes neurais artificiais, regressão

espacial e análise envoltória de dados.

2.2.1.2 Método involutivo

O método involutivo tem como objetivo identificar o valor de mercado de um empreendimento hipotético e fornecer informações para o estudo de viabilidade técnico-econômica deste. Esse método alicerçar-se no aproveitamento eficiente do bem, nas características da propriedade e as as condições do mercado no qual este está inserido, considerando-se cenários viáveis para execução e comercialização do produto (ABNT NBR 14653-2). A determinação das seguintes variáveis e parâmetros é fundamental para a aplicação desse método:

- Levantamento do custo de produção do projeto hipotético que corresponde à apuração dos custos diretos e indiretos, inclusive de elaboração a aprovação de projetos, necessários à transformação do imóvel para as condições do projeto hipotético.
- Previsão de despesas adicionais como compra do imóvel, administração do empreendimento, impostos e taxas, publicidade e comercialização de unidade.
- Margem de lucro do incorporador.
- Prazos para a execução do projeto hipotético e para a venda das unidades.
- Taxas de valorização imobiliária, de evolução de custos e despesas, de juros do capital investido e a mínima de atratividade.
- Modelo de avaliação: fluxos de caixa específicos, aplicação de modelos simplificados dinâmicos ou aplicação de modelos estáticos.

2.2.1.3 Método evolutivo

O método evolutivo identifica o valor do bem pelo somatório dos valores de seus componentes. A composição do valor total do imóvel avaliado pode ser obtida através da conjugação de métodos, a partir do valor do terreno, considerados o custo de reprodução das benfeitorias devidamente depreciado e o fator de comercialização, ou seja:

$$VI = (VT + VB).FC \quad (5)$$

onde VI é o valor do imóvel, VT é o valor do terreno, VB é o valor da benfeitoria, FC é o fator de comercialização.

A aplicação do método evolutivo exige que o valor do terreno seja determinado pelo método comparativo de dados de mercado (Subseção 2.2.1.1) ou, na impossibilidade deste,

pelo método involutivo (Subseção 2.2.1.2); as benfeitorias sejam apropriadas pelo método comparativo direto de custo ou pelo método da quantificação de custo; e fator de comercialização seja levado em conta, admitindo-se que pode ser maior ou menor do que a unidade, em função da conjuntura do mercado na época da avaliação (ABNT NBR 14653-2). A norma ressalta que quando o imóvel estiver situado em zona de alta densidade urbana, onde o aproveitamento eficiente é preponderante, o engenheiro de avaliações deve analisar a adequação das benfeitorias, ressaltar o sub-aproveitamento ou o super-aproveitamento do terreno e explicitar os cálculos correspondentes.

2.2.1.4 Método da capitalização da renda

O método da capitalização da renda identifica o valor do bem, com base na capitalização presente da sua renda líquida prevista, considerando-se cenários viáveis (ABNT NBR 14653-2). Quando a avaliação não se tratar de empreendimentos de base imobiliária tais como hotéis, *shopping centers* e outros; deve se observar os seguintes aspectos:

- Estimação das despesas necessárias à sua manutenção e operação, impostos, entre outros e receitas provenientes da sua exploração.
- Montagem do fluxo de caixa com base nas despesas e receitas previstas para o imóvel e suas respectivas épocas.
- Estabelecimento da taxa mínima de atratividade em função das oportunidades de investimentos alternativos existentes no mercado de capitais e, também, dos riscos do negócio.
- Estimação do valor máximo estimado para o imóvel é representado pelo valor atual do fluxo de caixa, descontado pela taxa mínima de atratividade.

2.2.2 Métodos para identificar o custo de um bem

Os métodos descritos nesta subseção são recomendados para a identificação do custo de todos os tipos de imóveis, incluindo os empreendimentos de base imobiliária tais como hotéis, *shopping centers* e outros.

2.2.2.1 Método da quantificação do custo

Esse método identifica o custo do bem ou de suas partes por meio de orçamentos sintéticos ou analíticos, a partir das quantidades de serviços e respectivos custos diretos e indiretos. O método da quantificação do custo é utilizado para identificar o custo de reedição de

benfeitorias e seu procedimento baseia-se no cálculo no custo unitário básico de construção ou na identificação do custo pelo orçamento detalhado. As etapas para se calcular o custo unitário básico são as seguintes:

- Determinação da área construída de acordo com a Equação (6).

$$S = A_p + \sum_i^n (A_{qi} \cdot P_i) \quad (6)$$

onde S representa a área construída efetiva do imóvel, A_p é a área construída padrão, A_{qi} é a área construída de padrão diferente e P_i é o percentual correspondente à razão entre o custo estimado da área de padrão diferente e a área padrão, de acordo com os limites estabelecidos na ABNT NBR 12721.

- Estimação do custo unitário básico de construção segundo a Equação (7)

$$C = [CUB + \frac{OE + OI + (OF_e - OF_d)}{S}] (1 + A)(1 + F)(1 + L) \quad (7)$$

onde C é o custo unitário de construção por metro quadrado de área equivalente de construção, CUB é o custo unitário básico, OE é o orçamento de elevadores, OI é o orçamento de instalações especiais e outras, tais como geradores, sistemas de proteção contra incêndio, centrais de gás, interfones, antenas, coletivas, urbanização, projetos etc., OF_e é o orçamento de fundações especiais, OF_d é o orçamento de fundações diretas, S é a área construída efetiva do imóvel, A é a taxa de administração da obra, é o percentual relativo aos custos financeiros durante o período da construção e L é o percentual correspondente ao lucro ou remuneração da construtora.

As etapas para se identificar o custo pelo orçamento detalhado consiste dos seguintes procedimentos:

- Levantamento dos quantitativos de materiais e serviços aplicados na obra.
- Pesquisa de custos de acordo com as especificações dos materiais e serviços utilizados para execução da benfeitoria, coletam-se os seus respectivos custos em fontes de consulta especializadas.
- Preenchimento da planilha orçamentária onde são discriminados todos os serviços, indicando-se a unidade de medida, a quantidade, o custo unitário, o custo total e a fonte de consulta.
- O cálculo da depreciação física que pode ser realizado de forma analítica – por meio de orçamento necessário à recomposição do imóvel na condição de novo – ou por meio da aplicação de coeficiente de depreciação, que leve em conta a idade e o estado de conservação. Esse coeficiente deve ser aplicado sobre o valor depreciável.

- O custo de reedição da benfeitoria que é o resultado da subtração do custo de reprodução da parcela relativa à depreciação.

2.2.2.2 Método comparativo direto de custo

De acordo com a ABNT NBR 14653-2, a utilização do método comparativo direto para a avaliação de custos deve considerar uma amostra composta por imóveis de projetos semelhantes, a partir da qual são elaborados modelos que seguem os procedimentos usuais do método comparativo direto de dados de mercado (Subsubseção 2.2.1.1).

2.2.3 Considerações do autor

A ABNT NBR 14653-2 apresenta uma razoável quantidade de métodos para se precificar imóveis que podem ser aplicados à unidades imobiliárias com diferentes características, situações diversas e informações de natureza distinta. Contudo, a maioria dos métodos apresentados não definem modelos ou equações matemáticas de precificação; e sim diretrizes gerais que servem como uma espécie de manual de "boas práticas" para engenheiros de avaliações de imóveis. Desde modo, o modelo computacional implementado neste trabalho pode ser utilizado como uma ferramenta suplementar pelo engenheiro de avaliação de imóveis.

É importante ressaltar que os métodos descritos na ABNT NBR 14653-2 se aplicam a situações normais e típicas do mercado. Em situações atípicas, onde ficar comprovada a impossibilidade de utilizar as metodologias previstas nesta parte da NBR 14653, é facultado ao engenheiro de avaliações o emprego de outro procedimento, desde que devidamente justificado.

2.3 Monografia DE NG - *Machine Learning For A London House Price Prediction Mobile Application*

Nesta seção, os objetivos e resultados da monografia de NG são apresentados e analisados a fim de orientar a metodologia que será apresentada no capítulo 4 deste trabalho.

2.3.1 Objetivos

Os objetivos de NG se assemelham bastante com aqueles aqui propostos. Em seu trabalho de monografia, NG utilizou uma base de dados contendo mais de 2 milhões de transações imobiliárias da cidade de Londres, no período compreendido entre 1995 e 2015, para treinar um modelo de aprendizado de máquina que seja capaz de fornecer previsões e tendências do

mercado imobiliário da cidade. NG foi além e incorporou o modelo desenvolvido num aplicativo para dispositivos móveis a fim de disponibilizar aos cidadãos londrinos as predições geradas pelo algoritmo.

2.3.2 Tratamentos dos dados

A Tabela 2 mostra como NG organizou os dados que possuía para treinar o modelo de aprendizado de máquina.

| Variáveis dependentes | Variáveis independentes |
|--|-------------------------|
| Data da transação (Quantidade de meses desde 1995) Tipo do imóvel (000/100/010/001) Tipo de construção (0/1) Possessão (0/1) Endereço (Latitude/Longitude) | Preço do imóvel |

Tabela 1 – Variáveis utilizadas por NG para treinamento do modelo

- A data da transação foi representada pela diferença de meses entre janeiro de 1995 e o mês mais recente presente na base de dados. Essa alteração simplificou a representação do tempo que estava medido em meses e anos.
- O tipo do imóvel foi representado por quatro códigos binários que representam:
 - 000 - Casa não geminada
 - 001 - Pequeno apartamento (*flat*)
 - 010 - Casa com terraço
 - 100 - Casa parcialmente geminada
- O tipo de construção foi representado pelo códigos 0 e 1 que significam, respectivamente, imóvel velho e imóvel novo.
- A posseção também foi representada pelo códigos 0 e 1 que significam, respectivamente, imóvel financiado e imóvel próprio.
- Para capturar a variação geográfica dos endereços nos preços dos imóveis com maior precisão, NG representou a localidade das propriedades por meio da latitude e longitude do endereços, ao invés de usar o código postal dos imóveis.

2.3.3 Treinamento do modelo

Devido à grande quantidade de dados disponíveis (mais de 2,4 milhões de transações), o treinamento do modelo se tornou computacionalmente impraticável. Para contornar esse

problema, NG fragmentou a base de dados em vários subconjuntos menores separados por localização. A quantidade de dados de cada subconjunto foi determinada empiricamente em função da relevância da micro região para o mercado imobiliário da cidade. Por exemplo, áreas da região central de Londres tiveram subconjuntos contendo mais de 10.000 dados, enquanto regiões mais afastadas do centro da cidade tiveram subconjuntos contendo pouco menos de 100 dados. Cada subconjunto foi treinado separadamente utilizando modelos de aprendizado de máquina baseados no processo gaussiano ². NG implementou o modelo em MATLAB e usou a ferramenta *GPML toolbox* ³ que é compatível com o software e possui uma vasta biblioteca de funções de média e covariância.

2.3.4 Resultados

As previsões obtidas pelo modelo de aprendizado de máquina são de característica temporal e espacial, uma vez que o aplicativo desenvolvido por NG é capaz de prever preços futuros de imóveis e apontar regiões da cidade de Londres que estão dentro de um orçamento (*budget*) determinado pelo usuário. A Figura 2 mostra o gráfico de tendência da evolução dos preços de imóveis da cidade de Londres gerado pelo modelo.

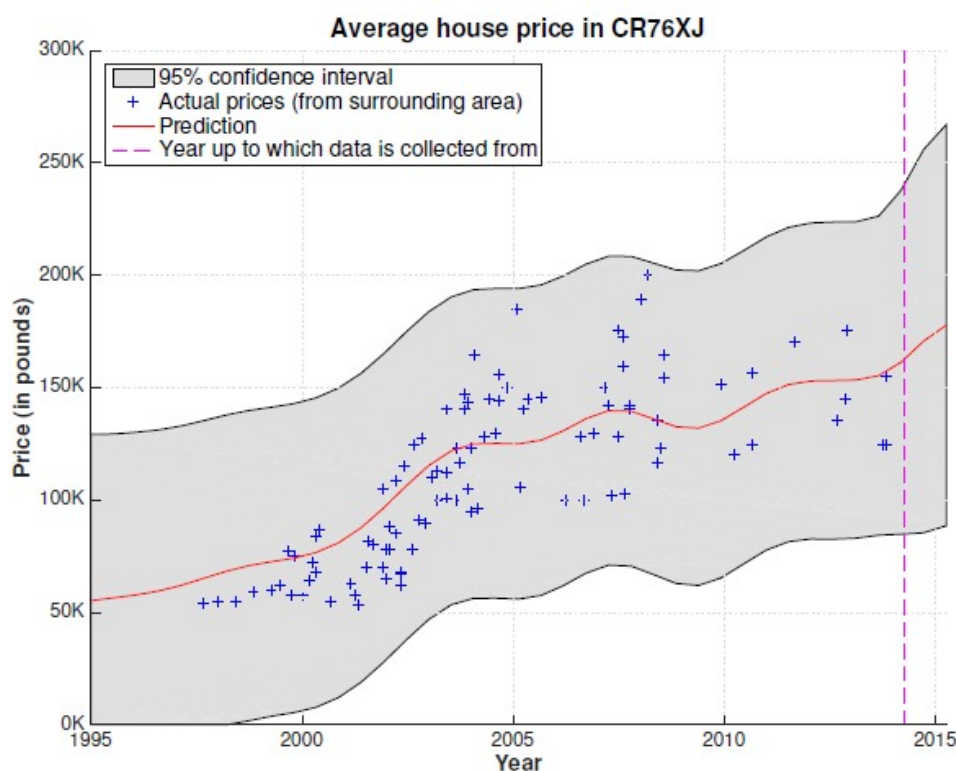
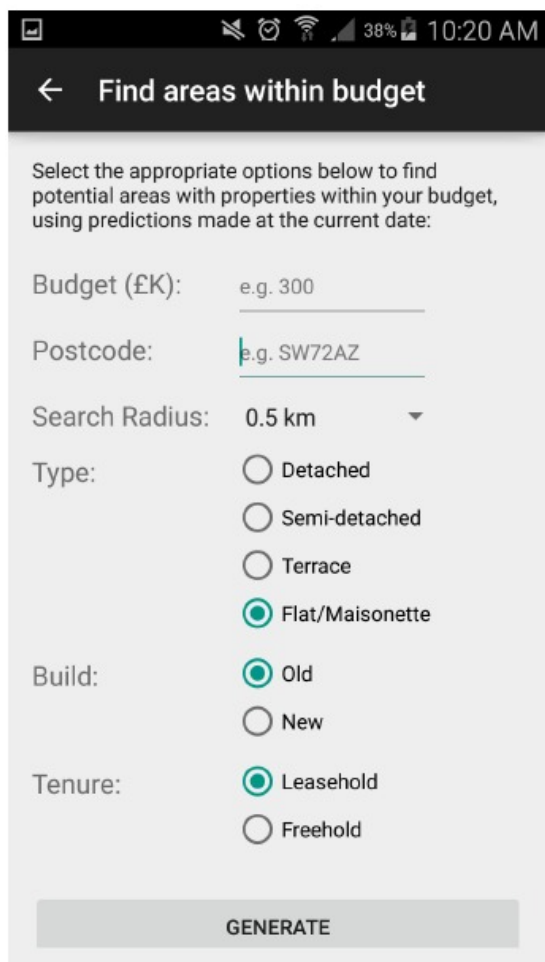


Figura 2 – Gráfico de previsão dos preços futuros de imóveis. Fonte: (NG, 2015) - página 36

²https://en.wikipedia.org/wiki/Gaussian_process

³<http://www.gaussianprocess.org/gpml/code/matlab/doc/index.html>

As figuras 3 e 4 mostram os resultados para as predições espaciais do aplicativo desenvolvido pelo autor da monografia. Observa-se que na Figura 4 estão destacadas as áreas que potencialmente possuem imóveis abaixo ou equivalente ao *budget* estipulado na Figura 3.



Find areas within budget

Select the appropriate options below to find potential areas with properties within your budget, using predictions made at the current date:

Budget (£K): e.g. 300

Postcode: e.g. SW72AZ

Search Radius: 0.5 km

Type:

- ☐ Detached
- ☐ Semi-detached
- ☐ Terrace
- ☒ Flat/Maisonette

Build:

- ☒ Old
- ☐ New

Tenure:

- ☒ Leasehold
- ☐ Freehold

GENERATE

Figura 3 – Identificação de áreas da cidade Londres que estão dentro do orçamento desejado. Fonte: (NG, 2015)

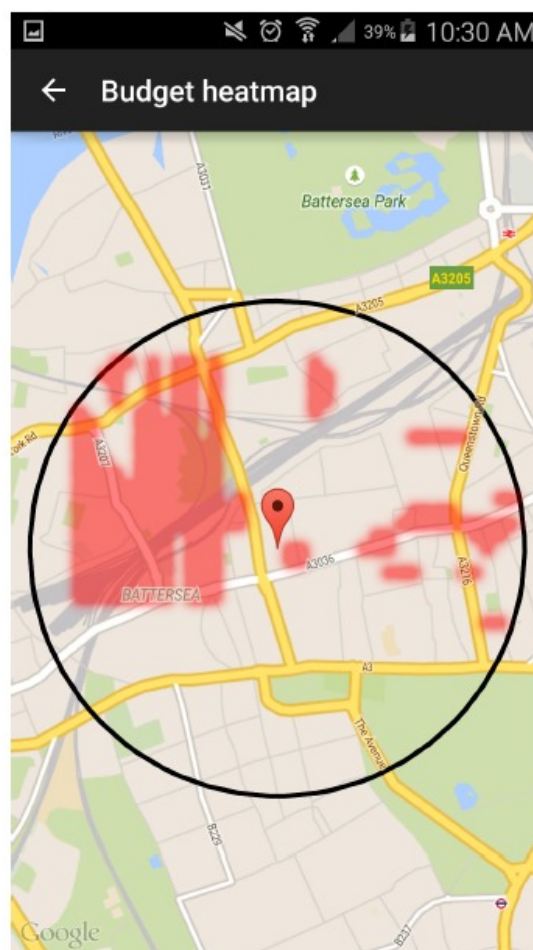


Figura 4 – Resultado da pesquisa das áreas que se enquadram dentro do orçamento desejado. Fonte: (NG, 2015)

A fim de validar os resultados de suas predições, NG realizou um experimento comparativo entre o modelo desenvolvido por ele e o modelo de estimativa de preços de um portal inglês de avaliação imobiliária e anúncios de imóveis chamado Zoopla⁴. A Tabela 2 apresenta os resultados desse experimento; pode-se concluir ao avaliá-la que os resultados de NG são significativamente piores do que aqueles precificados pelo portal. A justificativa apresentada por NG é que seu modelo utiliza menos variáveis independentes em relação ao modelo do portal Zoopla e, por isso, a melhora de seus resultados dependeria de uma investigação mais profunda

⁴Zoopla é um portal de anúncios e precificação de imóveis da Inglaterra. - <<https://www.zoopla.co.uk/>>

do mercado imobiliário londrino a fim de se obter mais fatores (variáveis) que afetam os preços dos imóveis.

| Sistema de Predição | Percentual de amostras | |
|---------------------|------------------------|----------------------|
| | Dentro de $\pm 10\%$ | Dentro de $\pm 20\%$ |
| Portal Zoopla | 47,6% | 75,0% |
| Monografia NG | 39,0% | 62,0% |

Tabela 2 – Comparação da quantidade percentual de amostras que foram precificadas dentro da margem de $\pm 10\%$ e $\pm 20\%$ em relação ao preço real. Fonte: AaronNg - Página 41

2.3.5 Considerações do autor

Apesar do trabalho de NG ter a mesma finalidade deste, a base de dados utilizada por ele permite realizar a predição futura dos preços e fornecer a tendência do mercado imobiliário; diferenciando-se assim do presente do trabalho que tem como objetivo predizer os preços dos imóveis desconsiderando, dentro do período analisado, a variação desses no tempo.

Ressalta-se também que o trabalho de NG foi publicado em Junho de 2015. Antes dessa data, modelos de aprendizado de máquina baseados em *gradient tree boosting*, que hoje são considerados o estado da arte de problemas de *machine learning*, não eram tão populares quanto atualmente. Desse modo, espera-se obter neste trabalho resultados melhores do que os NG por meio da utilização de modelos baseados em *gradient tree boosting*. O próximo capítulo descreve detalhadamente esse modelos.

3 FUNDAMENTAÇÃO TEÓRICA: APRENDIZADO DE MÁQUINA

Este capítulo introduz o conceito fundamental de aprendizado de máquina, apresenta o princípio teórico básico regressão e formula matematicamente os modelos de maior destaque em competições de aprendizado de máquina da plataforma Kaggle ¹; eles são: *ridge regression*, *lasso*, *redes de perceptrons de múltiplas camadas* e *gradient tree boosting*. Esses modelos serão utilizados e avaliados nos experimentos de predição de preços dos apartamentos no Capítulo 4.

3.1 Introdução ao aprendizado de máquina

Aprendizado de máquina (em inglês: *machine learning*) é a área de estudo da ciência da computação destinada ao desenvolvimento de algoritmos que otimizam um critério de desempenho por meio de dados e experiências passadas (ALPAYDIN, 2010). Os programas de computadores com capacidade de aprendizado são necessários quando o processo de se determinar um conjunto de regras para resolver um dado problema é extremamente complexo, mas é possível resolvê-lo com exemplos e dados anteriores. Como exemplo de problemas dessa natureza, cita-se a predição do volume de vendas de uma loja, classificação de experimentos físicos de alta energia, classificação de conteúdo textual, detecção de movimento, classificação de *softwares* maliciosos, sugestão de anúncios de internet baseada em cliques, precificação imobiliária, entre outros. O grande diferencial desses algoritmos é que eles não seguem um conjunto de instruções pré-definidas por seres humanos para produzir uma resposta e são classificados em três grupos: algoritmos de aprendizado supervisionado, algoritmos de aprendizado não supervisionado e algoritmos de aprendizado por reforço.

Durante o processo de aprendizado supervisionado, o modelo recebe diversos exemplos de entradas e saídas na tentativa de construir o melhor sistema que reproduz as saídas, de modo que futuras entradas sejam mapeadas de maneira automática. Problemas de regressão e classificação são exemplos da aplicação de algoritmos de aprendizado supervisionado (BISHOP, 2006). O aprendizado não supervisionado consiste em deixar o modelo encontrar o melhor sistema que descreve um determinado conjunto de dados a partir apenas da entrada do problema. Aplicações dessa modalidade de aprendizado são realizadas em problemas *clustering* (JAIN; MURTY; FLYNN, 1999) e estimativa de densidade (SILVERMAN, 1986). Finalmente, o aprendizado por

¹Kaggle é uma comunidade online de cientistas de dados e entusiastas de *machine learning* - <<https://www.kaggle.com/>>

reforço é área de estudo que visa compreender como modelos podem aprender a otimizar suas ações durante sua utilização a fim de melhorar seu desempenho, como por exemplo o treinamento de robôs para sair de labirintos.

Regressão é uma subárea de aprendizado supervisionado no qual a resposta dos problemas pode ser descrita por variáveis contínuas. Por sua vez, a classificação é também uma subárea de aprendizado de supervisionado em que a resposta dos problemas consiste em atribuir um grupo à amostra de entrada. A predição de preços imóveis é um problema de típico de regressão.

3.2 Regressão linear

O modelo mais simples de regressão consiste na combinação linear de variáveis independentes para se estimar uma variável dependente contínua (BISHOP, 2006). Embora a regressão linear seja uma aproximação muito simples para se modelar o complexo mercado de imóveis, existem vários conceitos fundamentais de regressão linear que são importantes para compressão de outras técnicas de regressão mais avançadas.

Considere um problema no qual $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$ representa o vetor de variáveis dependentes a serem determinadas e $\mathbf{X} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_n^T)^T$ é o vetor contendo as variáveis independentes, um modelo de regressão linear pode ser formalizado para se determinar cada uma das variáveis de \mathbf{y} da seguinte maneira:

$$y_i = \varepsilon_i + \sum_{j=0}^{p-1} \phi_j(\beta_j, x_{ij}) \quad i = 1, \dots, n \quad (8)$$

onde $\phi_j(\beta_j, x_{ij})$ é a função de base com os parâmetros $\beta = (\beta_0, \beta_1, \dots, \beta_j)^T$ e ε_i é o termo residual do processo estocástico e p é o número de variáveis dependentes (*predictors*) presentes em \mathbf{X} . Resumidamente, o termo $\sum_{j=0}^{p-1} \phi_j(\beta_j, x_{ij})$ representa a predição \hat{y}_i de y_i com erro ε_i , ou seja:

$$\varepsilon_i = y_i - \sum_{j=0}^{p-1} \phi_j(\beta_j, x_{ij}) = y_i - \hat{y}_i \quad i = 1, \dots, n \quad (9)$$

As funções de bases ϕ podem ser tanto funções lineares da variável independente no simples caso onde $\phi_j(x_{ij}) = x_{ij}$ quanto funções não lineares, tais como $\phi_j(x_{ij}) = x_{ij}^2, x_{ij}^3$ ou $\sqrt{x_{ij}}$. Na maioria dos casos, $\phi_0(x_{i0})$ é considerado igual à 1 tal que β_0 pode se comportar como parâmetro de ajuste do modelo a possíveis *outliers* que podem estar presentes no conjunto de dados (MONTGOMERY; PECK; VINING, 2013). Embora a Equação (8) pode ser uma função não linear através do uso de funções de base não lineares, o modelo continua sendo linear com

relação à β . De forma alternativa, a Equação (8) pode ser expressa na forma vetorial conforme mostrado abaixo:

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon \quad (10)$$

3.2.1 Ajuste do modelo de regressão linear

O processo de ajuste do modelo de regressão linear aos dados de entrada \mathbf{X} e \mathbf{y} consiste em determinar os parâmetros β que minimizam o erro residual ε . O Método dos Mínimos Quadrados (MMQ)² é uma técnica de otimização que minimiza ε por meio da função da soma dos quadrados dos resíduos (SQE).

$$SQE = \varepsilon_0^2 + \varepsilon_1^2 + \dots + \varepsilon_i^2 = \sum_i^n \varepsilon_i^2 = \varepsilon^T \varepsilon \quad (11)$$

Substituindo a Equação (10) na Equação (11), obtém-se:

$$SQE(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) = \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{X}\beta - \beta^T \mathbf{X}^T \mathbf{y} + \beta^T \mathbf{X}^T \mathbf{X} \beta \quad (12)$$

O processo de minimização é realizado derivando-se $SQE(\beta)$ em relação a β e igualando a equação a zero. Assim, tem-se:

$$\begin{aligned} \frac{\partial SQE(\beta)}{\partial \beta} &= -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X} \beta = 0 \\ \mathbf{X}^T \mathbf{X} \beta &= \mathbf{X}^T \mathbf{y} \\ \beta &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \end{aligned} \quad (13)$$

3.3 Regressão ridge

A *Ridge Regression* formulada por HOERL; KENNARD é um método de otimização do modelo apresentado na Equação (10) bem similar ao MMQ. Contudo, os coeficientes de Ridge β^R são estimados por meio de um processo que busca minimizar a equação descrita a seguir:

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \phi_j(\beta_j, x_{ij}) \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = SQE + \lambda \sum_{j=1}^p \beta_j^2 \quad (14)$$

onde $\lambda \geq 0$ é o parâmetro de ajuste (*tuning parameter*) a ser determinado de separadamente. Resolvendo a Equação (14) tendo em vista os coeficientes β^R que minimizam o erro residual da

²O MMQ é uma técnica de otimização matemática que procura encontrar o melhor ajuste para um conjunto de dados tentando minimizar a soma dos quadrados das diferenças entre o valor estimado e os dados observados (tais diferenças são chamadas resíduos. - <https://eml.berkeley.edu/sst/regression.html>)

regressão Ridge, obtém-se:

$$\beta^R = (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{y} \quad (15)$$

A Equação (14) busca estabelecer uma relação ideal entre a redução da soma dos quadrados dos resíduos e do fator de penalização λ , também conhecido como *shrinkage penalty*. A Equação (15) mostra que esse parâmetro de ajuste λ serve para controlar a magnitude dos coeficientes β^R e, consequentemente, a capacidade de regularização do modelo. Quando $\lambda = 0$, o efeito do fator de penalização é anulado e a *ridge regression* produz o mesmo resultado do MMQ. Entretanto, quando $\lambda \rightarrow \infty$, o impacto do fator de penalização aumenta e a estimativa dos coeficientes β^R tenderá a zero. Diferentemente do método dos mínimos quadrados que gera apenas um conjunto de parâmetros β , a regressão Ridge produz conjuntos diferentes de β^R para cada valor de λ . Portanto, a seleção de um bom valor para λ é crucial para performance do modelo (JAMES et al., 2014).

3.4 Lasso

O modelo Lasso é uma alternativa recente ao modelo de regressão Ridge que tem a capacidade de anular o efeito das variáveis dependentes de \mathbf{X} quando $\lambda \rightarrow \infty$. A formulação deste modelo é a seguinte:

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \phi_j(\beta_j, x_{ij}) \right)^2 + \lambda \sum_{j=1}^p |\beta_j| = SQE + \lambda \sum_{j=1}^p |\beta_j| \quad (16)$$

Observa-se que a única diferença entre os dois modelos é a substituição do termo β_j^2 por $|\beta_j|$. Enquanto na regressão Ridge o fator de penalização $\lambda \sum_{j=1}^p \beta_j^2$ aproxima os coeficientes β^R de zero quando $\lambda \rightarrow \infty$, o modelo Lasso vai forçar os coeficientes β_λ^L serem iguais a zero. Esta sutil modificação pode não trazer melhorias significativas para a acurácia do modelo, mas a manutenção do termo β_j^2 pode criar um desafio na interpretação dos resultados quando o número de variáveis dependentes de \mathbf{X} for muito grande. Em outras palavras, a regressão Ridge sempre irá produzir modelos incluindo todos os p *predictors* de \mathbf{X} por mais que se aumente o valor de λ (JAMES et al., 2014).

Desde modo, conclui-se que o método Lasso possui uma vantagem em relação ao método de regressão Ridge que é a capacidade de selecionar as variáveis dependentes de \mathbf{X} que possuem maior relevância para o modelo.

3.5 Redes neurais artificiais

3.5.1 Conceito básico de um neurônio artificial

Um neurônio artificial é uma unidade computacional capaz de mapear um vetor de variáveis de entrada $\mathbf{X} = (x_1, x_2, \dots, x_n)^T$ a uma variável de saída \hat{y} , através de uma função $f(x, w)^3$, cujos parâmetros w são obtidos através de um processo de otimização que visa aproximar $f(x, w)$ à função geradora dos dados, f^* . Uma representação de um neurônio artificial pode ser visto na Figura 5.

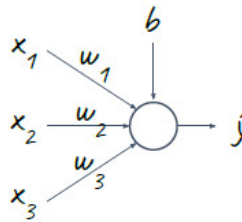


Figura 5 – Representação de um neurônio artificial

A função $f(x, w)$ computada por um neurônio tem sua forma genérica representada pela Equação (17).

$$\hat{y} = f(x, w) = g(w_1x_1 + w_2x_2 + \dots + w_mx_m + b) = g(w^T x) \quad (17)$$

onde os vetores de pesos e de entrada são dados por:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

e b é um peso de polarização, m é o número de características de x e $g(\cdot)$ é uma função de ativação.

As funções de ativação podem ter diversas formas e a escolha da mais apropriada depende da função que se deseja aproximar. As funções de ativação mais populares podem ser vistas nas figuras de 6 a 9.

Quando a função que se deseja aproximar é a função geradora de uma série temporal univariada, a Equação (17) toma uma forma auto-regressiva, onde a saída é o valor da série no

³A notação $f(x, w)$ pode ser lida como “função de x e parametrizada por w ”

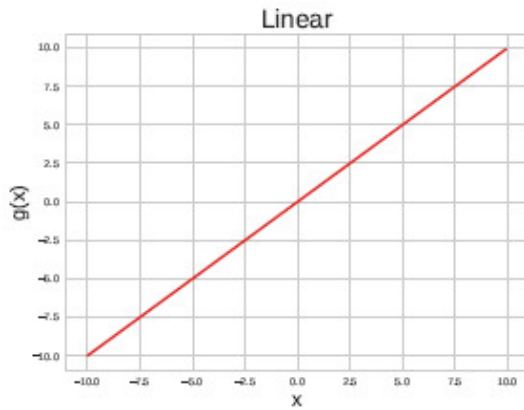


Figura 6 – Função linear

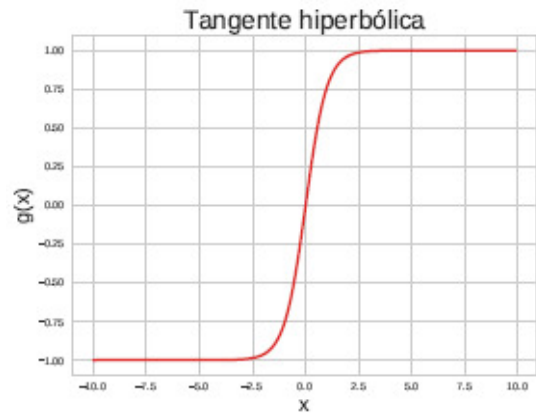


Figura 7 – Função tangente hiperbólica

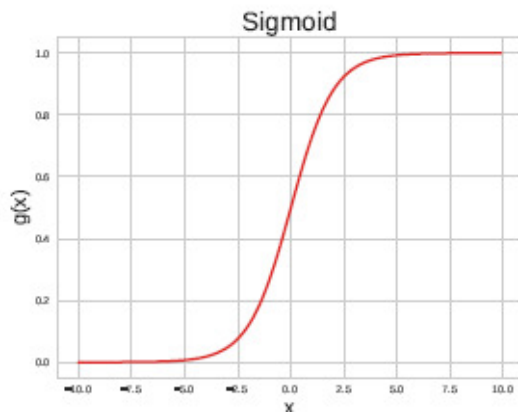


Figura 8 – Função sigmoideal

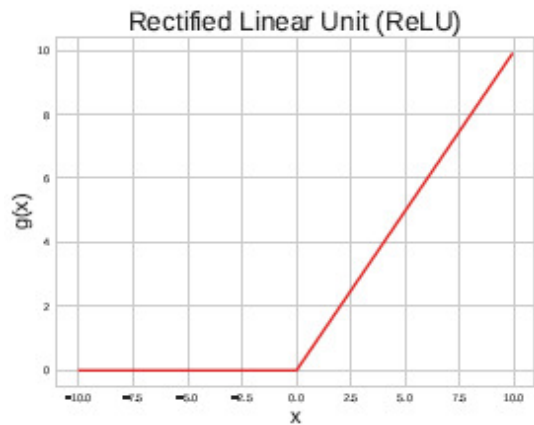


Figura 9 – Função unidade linear retificada

instante futuro t (\hat{y}_t) e o vetor de entradas é composto por amostras passadas da série (como y_{t-1} , y_{t-2} , etc.).

Essas amostras de entrada não necessariamente têm que ser consecutivas. Por exemplo, para uma série amostrada de hora em hora, pode ser interessante usar as amostras da hora anterior (y_{t-1}), do dia anterior (y_{t-24}) e do dia anterior a esse (y_{t-48}). Pelo fato de ser um problema de regressão – onde a saída esperada é um valor real –, a função de ativação utilizada é a linear ($g(x) = x$). Assim, a Equação (17) é adaptada para o exemplo em questão através da Equação (18).

$$\hat{y}_t = w_1 y_{t-1} + w_2 y_{t-24} + w_3 y_{t-48} + b \quad (18)$$

A Figura 10 ilustra a previsão \hat{y}_t da amostra y_t de uma série temporal, utilizando um neurônio alimentado pelos atrasos y_{t-1} , y_{t-2} , y_{t-3} e y_{t-4} .

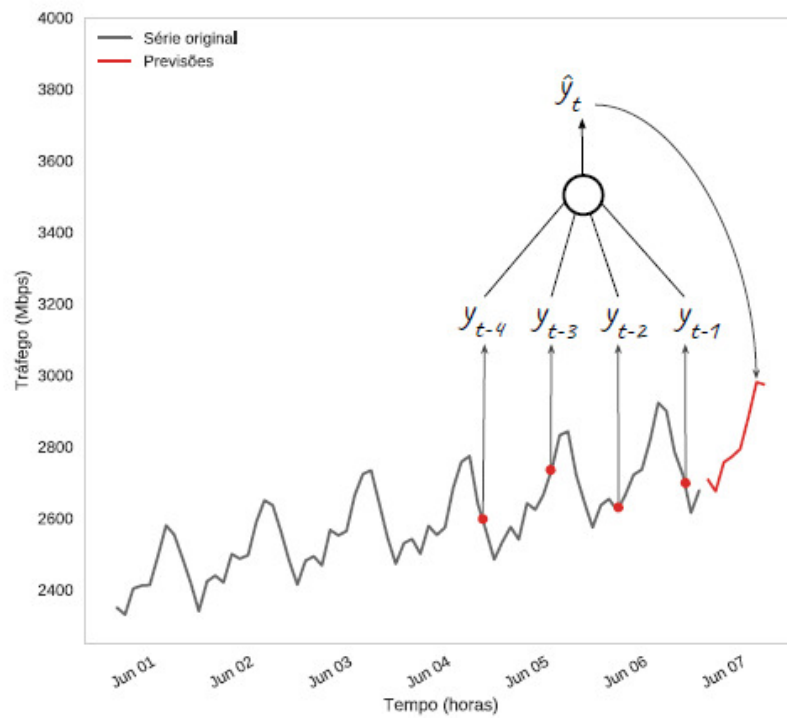


Figura 10 – Exemplo de previsão com um neurônio artificial

3.5.2 Processo de aprendizado de um neurônio artificial

Para que a função $f(x, w)$ seja encontrada, é necessário rearranjar a série temporal como um conjunto de dados de um problema de aprendizado supervisionado, como exemplificado na Tabela 3.

| Entradas | | | Saídas |
|-----------|-----------|-----------|----------|
| y_1 | y_2 | y_3 | y_4 |
| y_2 | y_3 | y_4 | y_5 |
| y_3 | y_4 | y_5 | y_6 |
| \vdots | \vdots | \vdots | \vdots |
| y_{N-3} | y_{N-2} | y_{N-1} | y_N |

Tabela 3 – Exemplo de um conjunto de dados para aprendizado supervisionado de um neurônio

De posse do conjunto de dados, uma parcela deles (os primeiros 80%, por exemplo) é utilizada para resolver o problema de otimização definido na Equação (19), procedimento que resulta na obtenção do vetor de pesos w .

$$w^* = \underset{w}{\operatorname{argmin}} J \quad (19)$$

Nessa equação, J é a função de custo dada pela Equação (20)

$$J = \frac{1}{2}(y - \hat{y})^T \cdot (y - \hat{y}) \quad (20)$$

onde \hat{y} e y são os vetores de previsão e de saída esperada do conjunto de treinamento, respectivamente. Especificamente, o problema de otimização definido em 19 é resolvido pelo algoritmo **gradiente descendente**, que está apresentado a seguir:

Algoritmo 1: Gradiente descendente

```

while critério de parada não alcançado do
    calcula previsões  $\hat{y}$  a partir das entradas  $X$  do conjunto de treino;
    calcula função de custo  $J$ ;
    calcula vetor gradiente  $\nabla J$  em relação aos pesos  $w$ ;
    atualiza os pesos:  $w \leftarrow w - \eta \nabla J$ ;
end

```

onde o vetor gradiente é dado por

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial w_1} \\ \frac{\partial J}{\partial w_2} \\ \vdots \\ \frac{\partial J}{\partial w_n} \end{bmatrix}$$

e η é a taxa de aprendizado, que controla quão rápido serão os ajustes em w . A intuição por trás da atualização de w no Algoritmo 1 reside no fato de ∇J possuir o sentido de máximo *crescimento* de J e, por consequência, $-\nabla J$ possuir o de máximo *decréscimo*. Como w está sendo atualizado para esse sentido, J está caminhando para um ponto de mínimo.

3.5.3 Redes de perceptrons de múltiplas camadas

Os modelos baseado em um único neurônio - como o Perceptron - têm a desvantagem de mapear somente relações lineares entre entradas e saídas. Pelo fato de no mundo real isso raramente ocorrer, tais modelos não são tão interessantes.

Por outro lado, quando neurônios com função de ativação não-lineares (exceto o da camada de saída) são conectados em camadas sucessivas, forma-se uma rede neural de múltiplas camadas denominada *Multi Layer Perceptron* (MLP), que tem capacidade de aproximar qualquer função não-linear (CYBENKO, 1989; DUDA; HART; STORK, 2001). A Figura 11 mostra uma

rede com uma camada escondida e a Figura 12 uma aplicação dessa rede para num problema de previsão de séries temporais.

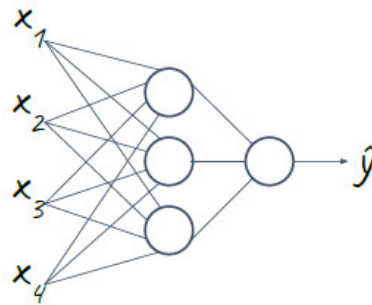


Figura 11 – Uma rede neural com uma camada escondida

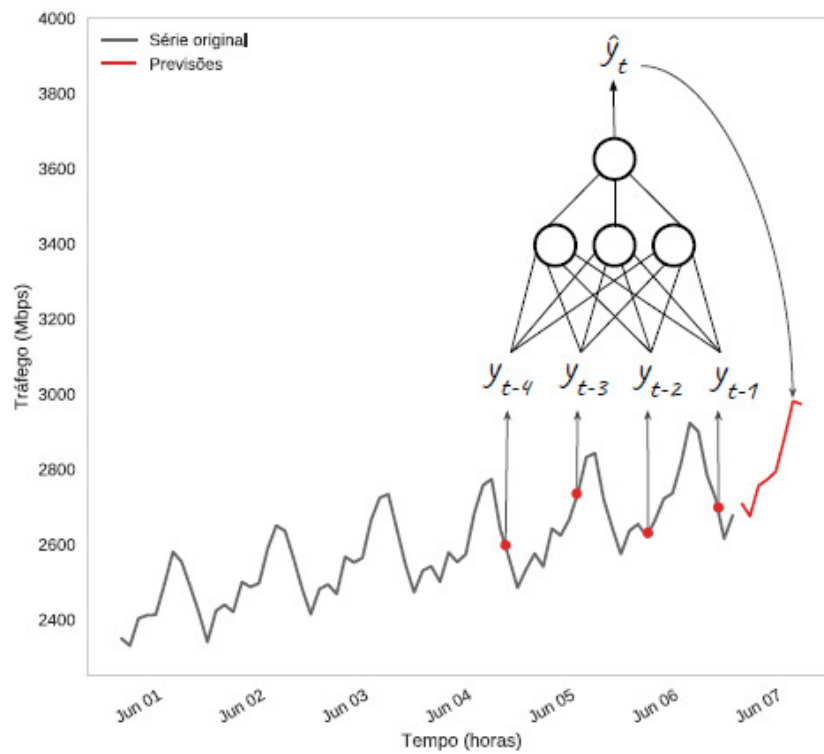


Figura 12 – Aplicação de uma rede neural com uma camada escondida em um problema de previsão

3.6 Modelos baseados em árvores de decisão

Seguindo uma abordagem diferente dos métodos apresentados anteriormente, os modelos de aprendizado de máquina baseados em árvores de decisão, propostos por BREIMAN et al., particionam o subespaço dos *predictors* \mathbf{X} em R regiões a fim de se estimar os valores das variáveis dependentes de \mathbf{Y} . Esse processo de estimativa envolve tipicamente o cálculo da média ou da moda das amostras de treino contida numa dada região R_i . Uma vez que as regras usadas para dividir o subespaço de \mathbf{X} podem ser organizadas num formato de árvore, esse tipo de abordagem foi denominado árvore de decisão (JAMES et al., 2014).

A Figura 13 ilustra o processo de decisão em árvore para o problema de avaliação do óleo de fluido de equipamentos de mineração abordado por LADEIRA et al. em seu artigo. O processo de decisão mostrado na Figura 13 é baseado num conjunto de regras que se iniciam no topo da árvore. Inicialmente, as amostras que possuem horímetro ⁴ superior a 40 dias são posicionadas no galho esquerdo da árvore. As amostras do lado direito são atribuídas à região que possui a maior média das amostras deste galho, no caso a região azul inferior do gráfico da esquerda. Observe que o galho esquerdo realiza mais duas subdivisões nas amostras, criando outras três regiões no subespaço de \mathbf{X} ; cada região R é denominada nó terminal ou folha da árvore. O particionamento do subespaço das variáveis independentes de \mathbf{X} conforme está mostrado na Figura 13 pode ser resumido no seguinte algoritmo:

Algoritmo 2: Árvore de decisão

Divida o subespaço \mathbf{X} em J regiões distintas e não sobrepostas, R_1, R_2, \dots, R_J .

for Para cada amostra do conjunto de teste que for classificada numa dada região

R_j **do**

Atribua à essa amostra a média das observações (dados do conjunto de treino)
presente em R_j .

end

O processo de construção da árvore descrito no Algoritmo 2 é realizado recursivamente. Primeiramente seleciona-se a variável X_j e o ponto de corte s que divide o subespaço de \mathbf{X} em duas regiões $\{X|X_j < s\}$ e $\{X|X_j \geq s\}$ que geram a maior redução da SQE possível. Em seguida, avalia-se todos os *predictors* X_1, \dots, X_p , assim como todos os possíveis valores de corte s de cada *predictor*, e escolhe-se a variável X_j e o valor de corte que resultam no menor valor da SQE (JAMES et al., 2014). Para uma abordagem geral desse processo que englobe qualquer valor de

⁴<http://blog.ceabs.com.br/odometro-e-horimetro-voce-sabe-diferenca-entre-eles/>

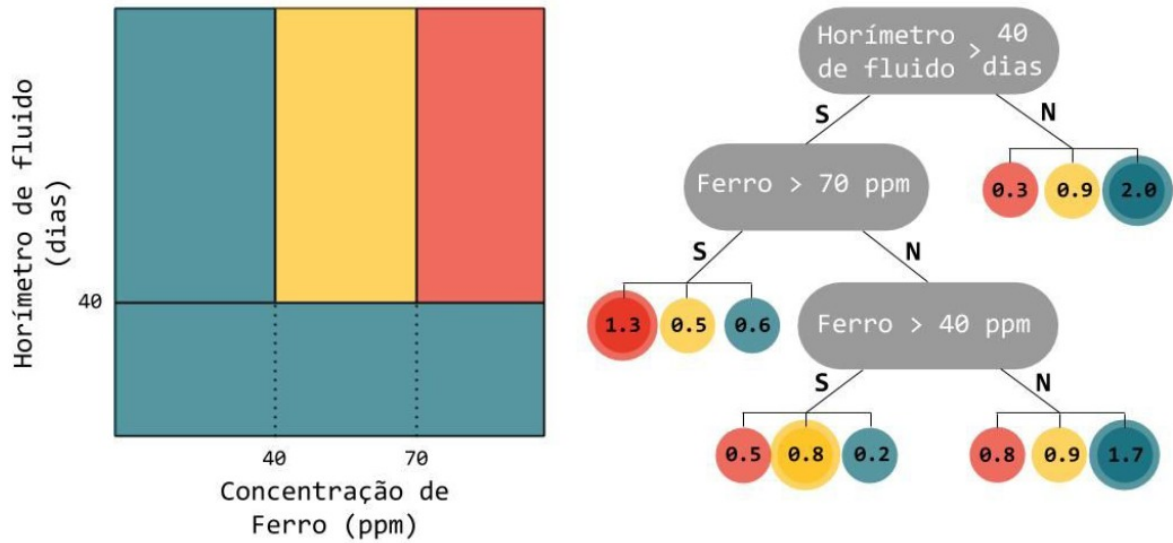


Figura 13 – Partição do Espaço do Amostral do Problema de Análise de Óleo Mineral. Fonte: (LADEIRA et al., 2017)

j e s , define-se o seguinte par de sub-planos

$$R_1(j,s) = \{X|X_j < s\} \quad e \quad R_2(j,s) = \{X|X_j \geq s\} \quad (21)$$

e determina-se o valor de j e s que minimizam a equação

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \quad (22)$$

onde $\hat{y}_{R_1}^2$ é o valor médio das amostras de treino presente em $R_1(j,s)$ e analogamente $\hat{y}_{R_2}^2$ representa o valor médio das amostras de treino presente em $R_2(j,s)$. A busca dos valores j e s que minimizam a equação Equação (22) pode ser realizada rapidamente se o número de variáveis \mathbf{X} do problema não for tão grande.

3.6.1 Gradient tree boosting machine

Os métodos *boosting* (do inglês: impulsionamento) visam melhorar o desempenho dos modelos baseados em árvores de decisão por meio da combinação não aleatória de diversas árvores simples, denominadas *weak learners*. Pode-se dizer, portanto, que ao invés de se criar uma única árvore ajustada aos dados de treino, os métodos *boosting* combinam a resposta de várias árvores e constroem um único modelo de predição (JAMES et al., 2014). Durante o processo de aprendizado, esses métodos criam novas árvores de maneira sequencial e utilizando as informações de árvores construídas anteriormente; assim cada nova árvore é ajustada a partir

de uma versão modificada do conjunto de dados original. Os métodos *boosting* possuem, no geral, uma gama de parâmetros a serem ajustados; para fins didáticos lista-se a seguir os três parâmetros mais relevantes desses métodos:

1. O número de árvores B , também conhecido como número de estimadores. Se B for muito grande, pode ocorrer um sobre-ajuste do modelo aos dados de treinamento levando-o a um desempenho ruim na predição dos dados de teste.
2. O parâmetro de redução (*shrinkage parameter*) λ , também conhecido como taxa de aprendizado. Esse parâmetro controla a taxa no qual o aprendizado de cada nova árvore contribui ao modelo de predição final. Os valores típicos de λ são 0.01 ou 0.001 e sua escolha correta depende de cada problema. Um valor pequeno de λ requer um grande valor de B para se atingir uma boa performance do modelo.
3. O número de divisão d ou nós terminais que cada árvore B pode criar. Esse parâmetro é também conhecido como *maxium depth*. Basicamente, d controla a complexidade do impulsionamento que se almeja atingir.

Uma vez definido os parâmetros principais, o algoritmo que sintetiza o funcionamento dos métodos *boosting* pode ser descrito conforme o Algoritmo 3. Esse algoritmo é denominado **Gradient Tree Boosting Machine** e foi formulado por FRIEDMAN. Ele recebe esse nome porque o processo de se criar um novo modelo $\hat{f}(x) + \lambda \hat{f}^b(x)$ a cada iteração, onde $\hat{f}^b(x)$ busca compensar o erro residual \mathbf{r}_{b-1} do ajuste do modelo anterior, é semelhante ao processo de se minimizar uma função por meio do deslocamento na direção oposta do seu gradiente; denominado gradiente descendente ⁵:

$$\theta_{i+1} = \theta_i - \phi \frac{\partial J}{\partial \theta_i}. \quad (23)$$

Matematicamente, o algoritmo do método *gradient tree boosting machine* tenta minimizar a equação:

$$J = \sum_b^B L(y, \lambda \hat{f}^b) \quad (24)$$

onde L representa a função de perda do modelo, ajustando $\hat{f}^1, \hat{f}^2, \dots, \hat{f}^B$ árvores de decisão. Ora, isto é:

$$\frac{\partial J}{\partial \hat{f}^b} = \frac{\partial \sum_b^B L(y, \lambda \hat{f}^b)}{\partial \hat{f}^b} = \frac{\partial L_b(r_{b-1}, \lambda \hat{f}^b)}{\partial \hat{f}^b} = \lambda \hat{f}^b - r_{b-1}. \quad (25)$$

⁵<https://en.wikipedia.org/wiki/Gradient_descent>

Algoritmo 3: Gradient Tree Boosting Machine

Ajuste a função do modelo de predição \hat{f} para $\hat{f}(x) = 0$.

Nomeie as variáveis independentes y como r_0 , onde r_b representa o erro residual do ajuste da b -ésima árvore.

for Para cada árvore $b = 1, 2, \dots, B$, **do**

Ajuste a árvore \hat{f}^b com d divisões e $d + 1$ partições com os dados de treinamento $(\mathbf{X}, \mathbf{r}_{b-1})$:

$$\hat{f}^b(x) \leftarrow r_{b-1} \quad (26)$$

Atualize a função do modelo de predição:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x) \quad (27)$$

Atualize o erro residual r :

$$r_b(x) \leftarrow r_{b-1} - \lambda \hat{f}^b(x) \quad (28)$$

end

return Retorne o modelo boosting final:

$$\hat{f}(x) = \sum_{b=1}^B \lambda \hat{f}^b(x). \quad (29)$$

A Equação (25) mostra que os erros residuais r podem ser interpretados como gradientes negativos. Portanto, a partir das equações 26, 28 e 25, a Equação (27) pode ser rescrita da seguinte maneira:

$$\hat{f}(x) = \hat{f}(x) - \lambda \frac{\partial J}{\partial \hat{f}^b(x)} \quad (30)$$

Observe que a Equação (30) é semelhante à Equação (23). Provando assim a capacidade intrínseca do método *gradient tree boosting machine* de minimizar o erro residual do ajuste ao dados de treino por meio do deslocamento na direção oposta do gradiente da função de perda L .

3.6.2 Análise crítica do método gradient tree boosting

Dentre os métodos de aprendizado de máquina usados em aplicações reais, o *gradient tree boosting* é a técnica que tem obtido resultados extremamente satisfatórios e liderado várias competições de *machine learning* na plataforma Kaggle. O *gradient tree boosting* não requer muita memória de processamento e é relativamente rápido, não necessita que os dados de

entrada estejam perfeitamente normalizados e suporta uma mistura de tipos de dados entrada como binários, categóricos e valores contínuos. Assim, esse método é considerado por muitos pesquisadores de *machine learning* o estado da arte da solução de problemas clássicos de classificação (GRUNWALD; SPIRITES, 2012). O algoritmo LambdaMART, uma variação do *gradient tree boosting* desenvolvido por BURGESS, alcança o estado da arte atualmente em resultados de problemas de ranqueamento. Além disso, métodos de *gradient tree boosting* são incorporados em aplicações reais para predição de cliques em anúncios online (HE et al., 2014), modelos que combinam dois ou mais modelos (*ensemble*) e em desafios como o *Netflix prize* (BENNETT; LANNING, 2007). Uma outra variação da proposição original do *gradient tree boosting*, denominada **XGBoost** e desenvolvida por CHEN; GUESTRIN, inclui funcionalidades como tratamento eficiente de dados esparsos e regularizações de diversos tipos. O XGBoost tem obtido bons desempenhos para o treinamento com o uso de paralelismo; dentre as 29 soluções vencedoras do site Kaggle publicadas em 2015, 17 usaram esse modelo.

Entretanto, a performance do *gradient tree boosting*, assim como a de outros métodos baseados em árvores de decisão, é extremamente difícil de ser interpretada por seres humanos. Ademais, esse método requer uma cuidadosa escolha de seus parâmetros principais e o processo de treinamento pode demandar um significativo poder computacional.

4 METODOLOGIA

Neste capítulo serão apresentados o processo de obtenção, análise e transformação dos dados, bem como a metodologia adotada nos experimentos e a parametrização dos modelos.

4.1 Obtenção dos dados

Os dados utilizados neste trabalho foram extraídos do portal Rede NetImóveis ¹ por meio de uma técnica simples de *web scraping* ². Essa técnica consiste em obter o objeto JSON ³ passado à API ⁴ do site durante uma requisição. Felizmente, a página da Rede NetImóveis possui uma baixa segurança dos seus dados e o JSON que é retornado em uma consulta pode ser facilmente obtido ao se monitorar as chamadas que a API realiza. A Figura 14 exibe o processo de se inspecionar as chamadas que a API de uma página web realiza ao servidor.

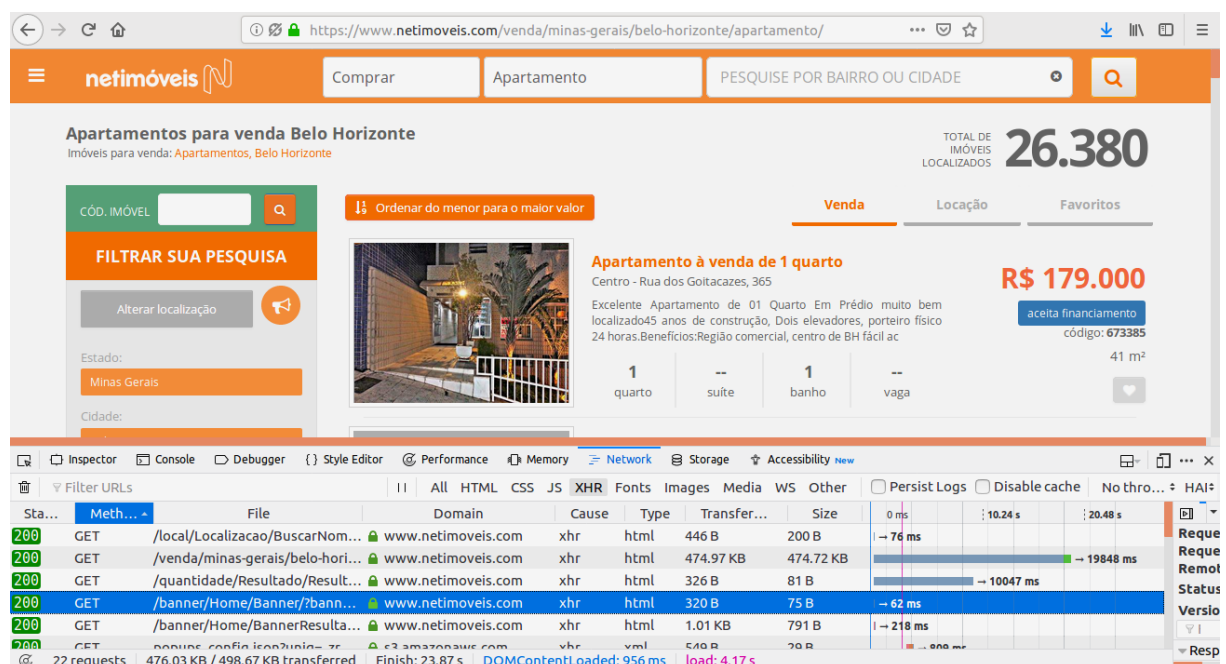


Figura 14 – Obtenção do objeto JSON do site da Rede Net Imóveis

Ao se abrir num navegador o terceiro link listado na parte inferior da Figura 14, obtém-se todos os dados referentes aos imóveis da consulta realizada, formatados em notação de objeto

¹ <<https://www.netimoveis.com/>>

² Ação de extrair informações de páginas web em massa e automaticamente <https://en.wikipedia.org/wiki/Web_scraping>

³ JavaScript Object Notation <<https://pt.wikipedia.org/wiki/JSON>>

⁴ Application Programming Interface <https://en.wikipedia.org/wiki/Web_API>

JavaScript. A Figura 15 exibe o objeto JSON que contém as informações dos imóveis.

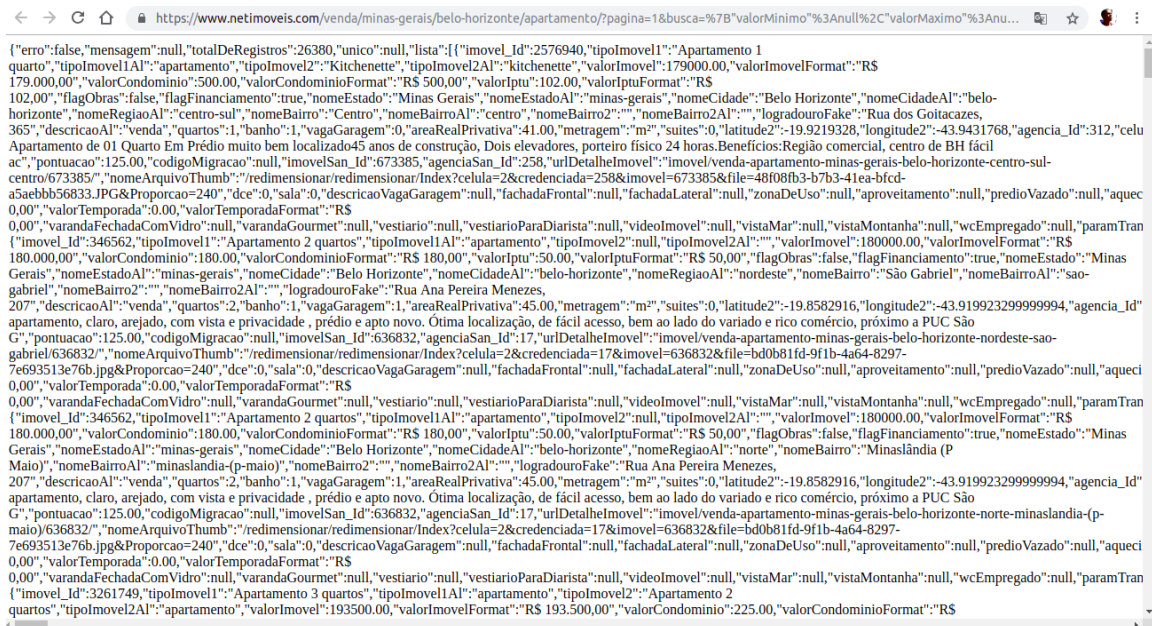


Figura 15 – Objeto JSON contendo as informações dos imóveis

Uma vez obtido o link que contém o objeto JSON, o script mostrado na Figura 16 e escrito na linguagem Python⁵ pode ser executado para se extrair os dados do site da Rede NetImóveis de forma automática.

```

1  getdata.py
2  import urllib.request
3
4  head_columns = ['Line', 'Imovel ID', 'Preço, Área', 'Latitude',
5                  'Longitude', 'Qtde. Quartos', 'Qtde. Banheiros',
6                  'Qtde. Suítes', 'Vagas. Garagem', 'Valor IPTU',
7                  'Valor Cond', 'Região', 'Bairro', 'Page',
8                  'DataExtração', 'Rua', 'Número']
9
10 print (''.join(head_columns))
11
12 page = 1
13 n = 1
14
15 link = 'https://www.netimoveis.com/venda/minas-gerais/belo-horizonte/apartamento/?pagina='
16
17 features = ['imovel_id', 'valorImovel', 'areaRealPrivativa',
18             'latitude2', 'longitude2', 'quartos',
19             'banho', 'suítes', 'vagaGaragem', 'valorCondominio',
20             'valorIptu', 'nomeCidadeAL', 'nomeRegiaoAL',
21             'nomeBairro']
22
23 while True:
24     r = urllib.request.urlopen(link + str(page))
25
26     import json
27     my_data = json.loads(r.read().decode('utf-8'))
28     if not my_data['lista']:
29         break
30     for i in range(len(my_data['lista'])):
31         json = my_data['lista'][i]
32         my_list = []
33         my_list.append(str(n))
34         for feature in features:
35             my_list.append(str(json[feature]))
36         my_list.append(str(page))
37         my_list.append(str(json['logradouroFake']))
38         n += 1
39     print (''.join(my_list))
40     page += 1
41     my_data.clear()

```

Figura 16 – Script usado para se extrair os dados do portal da Rede NetImóveis automaticamente

⁵Python é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte.

4.2 Extrações

Foram realizadas oito extrações no portal da Rede NetImóveis nos dias 05/08/2018, 12/08/2018, 21/08/2018, 25/08/2018, 29/08/2018, 21/09/2018 e 12/10/2018 a fim de se obter a maior quantidade de dados possível. No total foram obtidos **62.605** anúncios de imóveis que estão disponíveis no GitHub ⁶ do autor deste trabalho. Entretanto, a Tabela 4 abaixo mostra que apenas **21.234** ou 34% dos dados extraídos são úteis para os experimentos. Ademais, A Tabela 4 mostra a relação da quantidade de dados obtidos por extração, desconsiderando as amostras com informações inconsistentes; como por exemplo dados que possuem coordenadas geográficas fora do território de Belo Horizonte, quantidade de quartos e banheiros igual a zero e apartamentos com área inferior a $10m^2$.

A coluna **variação do preço** da Tabela 4 indica a variação a percentual que ocorreu no preço das amostras que se repetiram de uma extração para outra. A tendência de deflação dos preços dos apartamentos evidenciada na tabela pode ser verificada também no índice de preço FipeZap ⁷.

Por fim, a Tabela 4 confirma que a desconsideração da variação dos preços dos apartamentos no tempo, dentro do período analisado, é uma aproximação razoavelmente aceitável.

| Data | Quantidade de Dados | | Variação % média do preço dos dados repetidos |
|--------------|---------------------|---------------|---|
| | Extraídos | Novos | |
| 05/08 | 6.323 | 6.323 | 0% |
| 12/08 | 15.411 | 9.191 | -1.93% |
| 21/08 | 5.617 | 534 | -1.21% |
| 25/08 | 1.594 | 1 | 3.3% |
| 29/08 | 8.616 | 176 | -0.61% |
| 21/09 | 5.616 | 0 | -1.21% |
| 12/10 | 19.428 | 5.009 | -1.7% |
| TOTAL | 62.605 | 21.234 | -1.94% |

Tabela 4 – Relação dos dados obtidos por extração

No total, obteve-se 14 características de cada uma das 21.234 amostras. Elas são: **preço, área útil do imóvel, quantidade de quartos, quantidade de banheiros, quantidade de suítes, quantidades de vagas de garagem, valor do condomínio, valor IPTU, latitude, longitude, região, bairro, rua, número.**

⁶GitHub é uma plataforma de hospedagem de código-fonte online com controle de versão. - <https://github.com/gpass0s/Graduation_Project>

⁷FipeZap é um dos principais indicadores de preços de imóveis do Brasil. - <<http://fipezap.zapimoveis.com.br/>>

4.3 Análise e transformação dos dados

A análise e visualização dos dados são procedimentos que contribuem para a melhoria da performance dos algoritmos de *machine learning*, pois elas possibilitam que os cientistas de dados compreendam melhor as informações e as ajuste caso necessário. Nesta seção serão apresentadas algumas técnicas de análise, visualização e transformação de dados que foram realizadas, utilizando a linguagem Python, nos dados obtidos na Seção 4.2 antes de se avaliar a performance dos algoritmos de aprendizado de máquina.

4.3.1 Outliers

Em estatística, *outlier*, valor aberrante ou valor atípico, é uma observação que apresenta um grande afastamento dos demais valores da série ou que é inconsistente. A existência de *outlier* implica, tipicamente, em prejuízos a interpretação dos resultados dos testes estatísticos aplicados às amostras. Existem vários métodos de identificação de *outliers*. A Figura 17 ajuda na identificação de tais ocorrências.

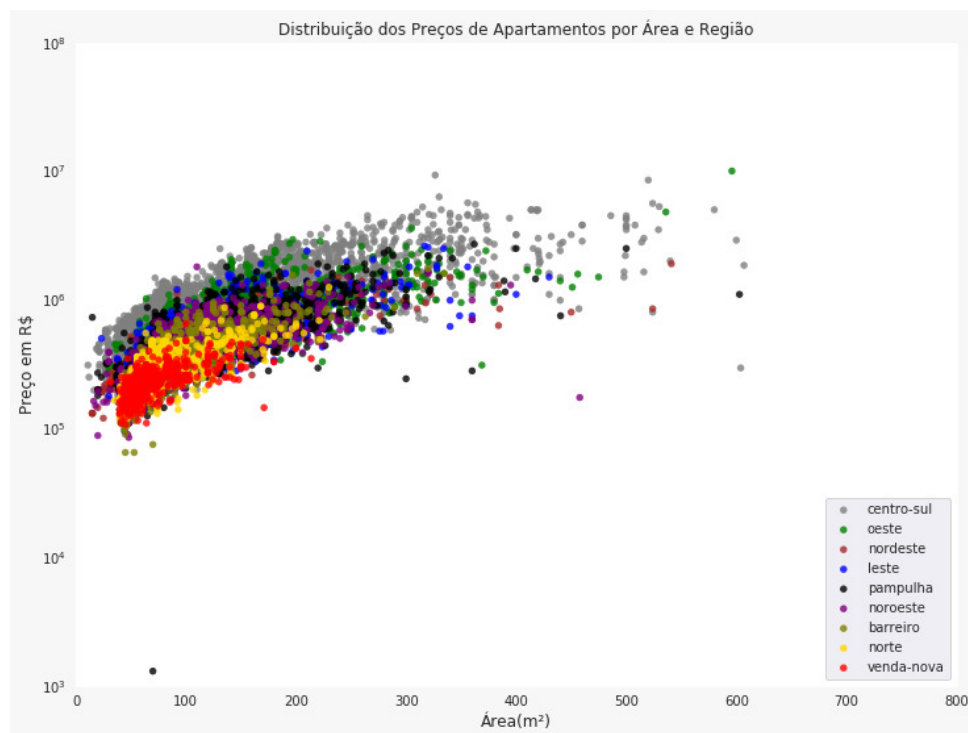


Figura 17 – Distribuição dos preços dos apartamentos por área e região

A Figura 17 permite identificar alguns *outliers* de maneira direta, como por exemplo o ponto roxo na parte inferior direita e o ponto preto na parte inferior esquerda da figura; outros

pontos, porém, merecem uma análise mais detalhada. A Figura 18 mostra detalhadamente as informações desses dois pontos mencionados e permite concluir que eles são amostras inconsistentes a serem removidas. Todavia a exclusão de todos os *outliers* pode afetar negativamente o desempenho dos modelos, uma vez que pode haver outros *outliers* nos dados de teste. Portanto, ao invés de removê-los por completo, ir-se-á parametrizar os modelos de aprendizado de máquina de modo a torná-los o mais robustos possíveis à dados dessa natureza.

```
In [63]: 1 # Outlier mostrado na parte inferior esquerda do gráfico:
2 train.loc[(train['Preço'] < 10000)][['Preço', 'Area', 'Regiao', 'Bairro', 'Qtde Quartos', 'Qtde Banheiros',
3                                         'Qtde Suites', 'Vagas Garagem', 'Valor IPTU']]

Out[63]:
```

| | Preço | Area | Regiao | Bairro | Qtde Quartos | Qtde Banheiros | Qtde Suites | Vagas Garagem | Valor IPTU |
|-------|-----------|---------|----------|-----------|--------------|----------------|-------------|---------------|------------|
| 15945 | 1300.0000 | 70.0000 | pampulha | Liberdade | 3 | 1 | 1 | 1 | 300.0000 |

```
In [64]: 1 # Outliers mostrado na parte inferior direita do gráfico:
2 train.loc[(train['Regiao'] == 'noroeste') &
3           (train['Area'] > 400)][['Preço', 'Area', 'Regiao', 'Bairro', 'Qtde Quartos', 'Qtde Banheiros',
4                                   'Qtde Suites', 'Vagas Garagem', 'Valor IPTU']]

Out[64]:
```

| | Preço | Area | Regiao | Bairro | Qtde Quartos | Qtde Banheiros | Qtde Suites | Vagas Garagem | Valor IPTU |
|-------|-------------|----------|----------|------------|--------------|----------------|-------------|---------------|------------|
| 13302 | 173900.0000 | 457.7700 | noroeste | Vila Oeste | 2 | 1 | 0 | 1 | 0.0000 |

Figura 18 – Alguns *outliers* mostrados na Figura 17

4.3.2 Correlação entre as variáveis do problema

A correlação indica o quão relacionadas duas variáveis X, Y são. Por exemplo, se essas duas variáveis crescem ou diminuem na mesma direção e proporção, elas são positivamente correlatas. Analogamente, se essas duas variáveis crescem ou diminuem em direções opostas, elas são negativamente correlatas. O coeficiente de correlação $\rho_{X,Y}$ entre duas variáveis X e Y , com médias μ_x e μ_y e desvios padrões σ_X e σ_Y pode ser calculado por meio da Equação (31)

$$\rho_{X,Y} = \text{corr}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_x)(Y - \mu_y)]}{\sigma_X \sigma_Y} \quad (31)$$

O colário do matemático de Cauchy–Schwarz ⁸ diz que o coeficiente de correlação não pode exceder $|1|$. Posto isso, uma correlação igual a $+1$ indica o caso em que duas variáveis são perfeitamente correlatas e -1 o caso de relação inversamente linear ou anticorrelação.

A Figura 19 exibe a matriz de correlação das variáveis do conjunto de dados obtido. Como as informações que compõe o endereço dos imóveis estão sintetizadas na coordenadas geográficas, essas serão desconsideradas daqui em diante.

A matriz de correlação da Figura 19 mostra que há uma baixa correlação das variáveis **valor do IPTU**, **valor do condomínio**, **latitude** e **longitude** com a variável de interesse, o **preço**

⁸Augustin-Louis Cauchy foi um matemático, engenheiro e físico francês do século 18 que contribuiu para áreas da matemática analítica e mecânica de meios contínuos. - <https://en.wikipedia.org/wiki/Cauchy%E2%80%9393Schwarz_inequality>

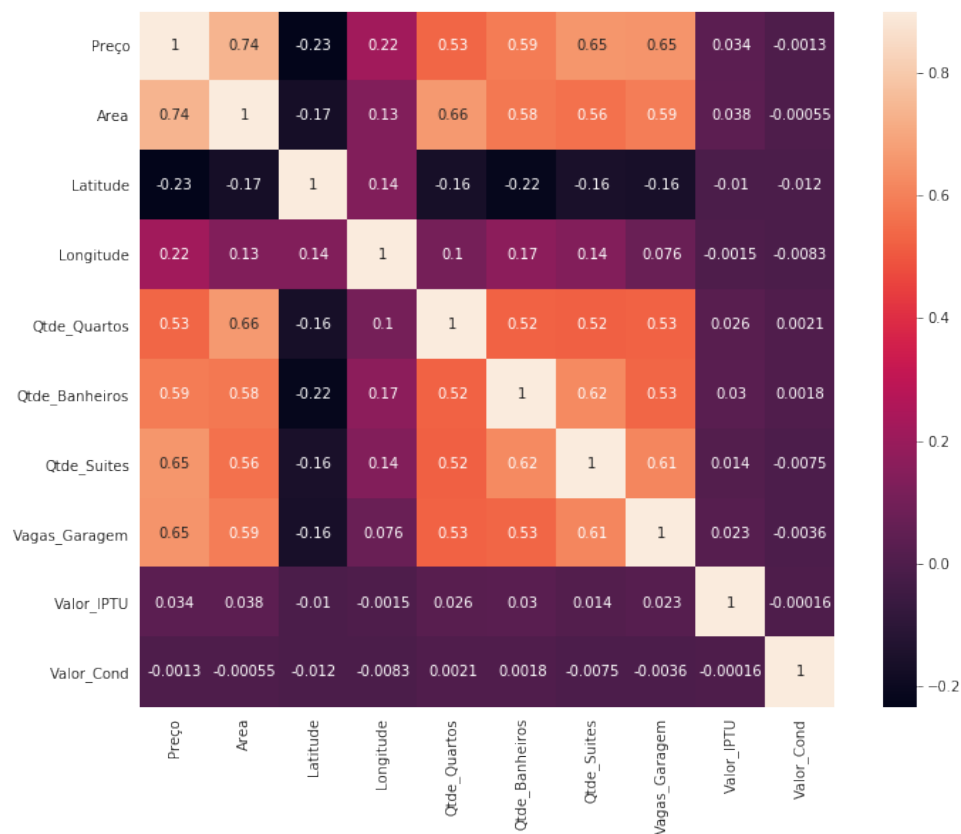


Figura 19 – Matriz de correlação das variáveis do conjunto de dados obtido

dos apartamentos. Uma possível explicação para a baixa correlação das duas primeiras variáveis citadas é o fato de vários imóveis serem isentos de IPTU, não terem taxa de condomínio, ou até mesmo não possuírem essas informações cadastradas no banco de dados da Rede NetImóveis. A Figura 20 revela que 7.634 ou aproximadamente 36% dos dados está em uma dessas situações mencionadas anteriormente, um valor bem expressivo.

```
In [76]: 1 # Quantidade de imóveis que são isentos de IPTU ou não possuem taxa de condomínio
          2 len(train.loc[(train['Valor_IPTU']==0) | (train['Valor_Cond']==0)])
Out[76]: 7634
```

Figura 20 – Quantidade de amostras que possuem valor do IPTU ou valor do condomínio igual a zero

A razão da baixa correlação entre os preços dos imóveis e as coordenadas geográficas se deve, muito provavelmente, ao fato da latitude e longitude não possuírem nenhuma propriedade ordinária, como a idade de uma pessoa. Por exemplo, os pontos (-19,9385; -43.8559) e (-19,9285; -43,9559) não possuem nenhum significado ou ordem de grandeza, eles são apenas dois pontos no espaço como qualquer outro. Essas duas variáveis, latitude e longitude, avaliadas de maneira

distinta, irão no máximo captar um gradiente leste-oeste ou norte-sul na distribuição geográfica dos preços, se houver. Como é sabido que a localidade carrega informações relevantes sobre os preços dos apartamentos, ir-se-á, na próxima sub seção, agregar um significado quantitativo às coordenadas geográficas.

4.3.3 Agregando um significado quantitativo à distribuição geográfica dos imóveis

Uma maneira simples de se agregar valor à localidade dos apartamentos é calcular a média e o desvio padrão dos preços da região que um determinado imóvel se encontra. A Figura 21 exibe os dados distribuídos no território de Belo Horizonte e divididos por região.

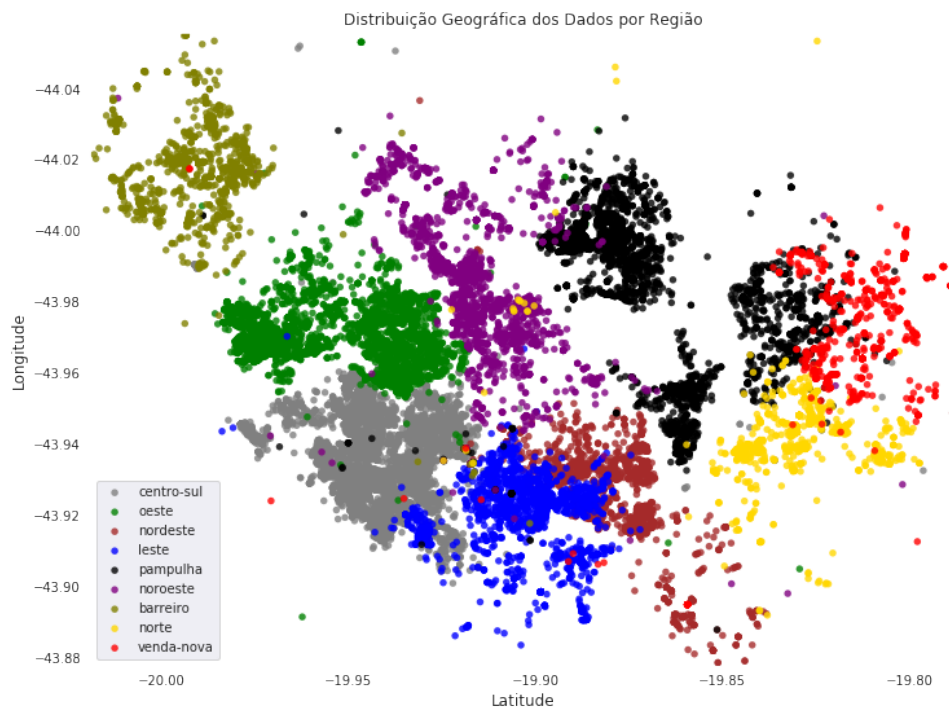


Figura 21 – Distribuição geográfica dos dados no território da cidade divididos por região

Contudo, sabe-se que dividir os apartamentos nas nove regiões da cidade é uma aproximação bastante grosseira da realidade. Ir-se-á, portanto, definir uma quantidade de regiões tal que permita cada agrupamento ter no mínimo 0,5% ou 107 amostras das 21.232 do conjunto de treino. Após uma série de tentativas, verificou-se que 50 sub regiões é a quantidade limiar da regra estabelecida. Acima desse valor, os dados ficam melhores subdivididos no espaço, mas algumas regiões apresentam quantidade ínfima de amostras. A Figura 22 apresenta o resultado obtido.

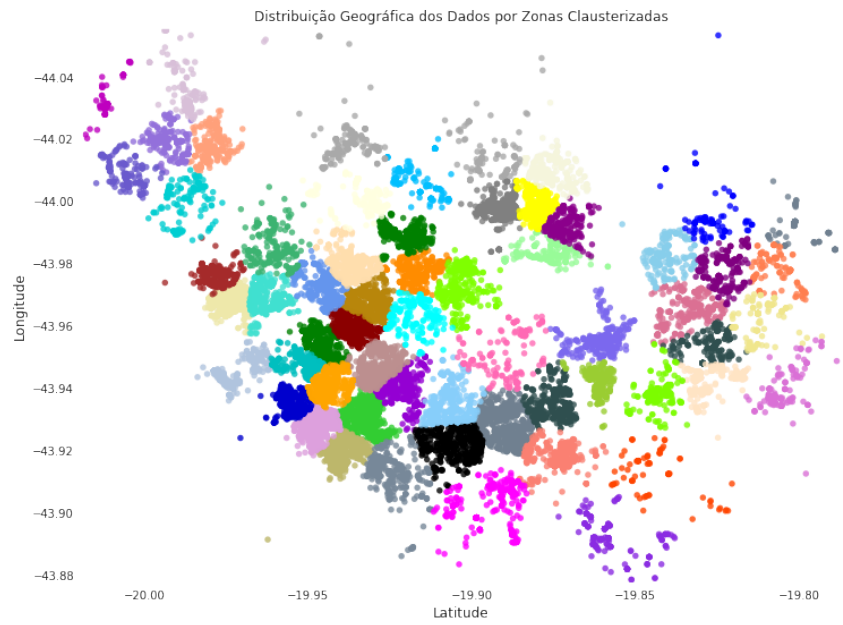


Figura 22 – Distribuição geográfica dos dados divididos em 50 regiões

Ao se computar a média e o desvio padrão de cada umas regiões mostradas na Figura 22, cria-se mais duas variáveis, **Mean_PreçoZone** e **Std_PreçoZone**, no conjunto de dados. A matriz de correlação mostrada na Figura 22 revela que essas duas novas variáveis possuem uma relação significativa com os preços e ajudam agregar valor quantitativo à localidade dos imóveis.

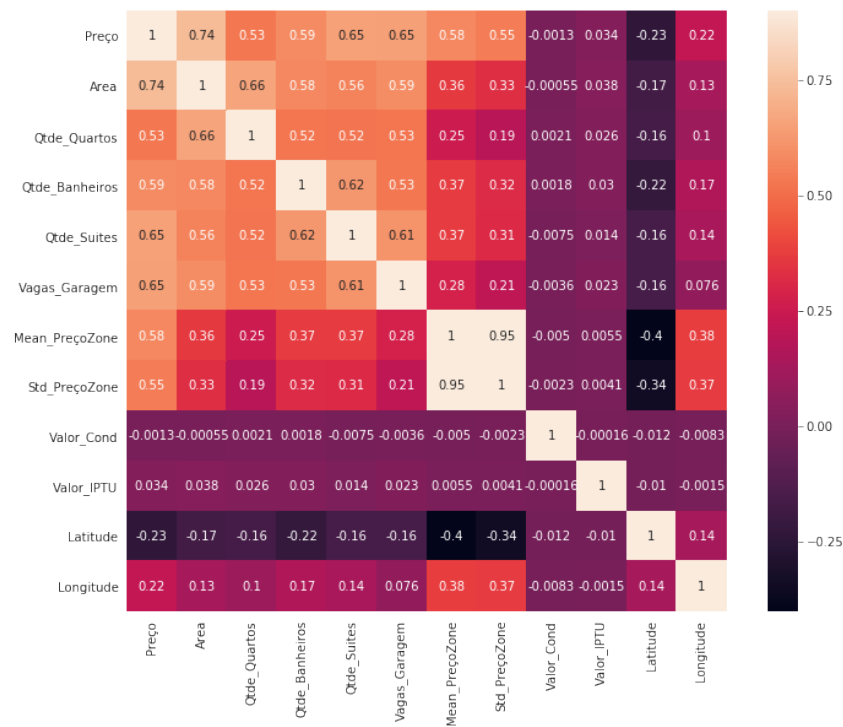


Figura 23 – Matriz de correlação

4.3.4 Normalização das distribuições dos dados

A distribuição normal, ou distribuição gaussiana, é a distribuição preferida dos estatísticos e dos cientistas de dados por serem capazes de modelar uma vasta quantidade de fenômenos naturais e por tornarem as operações matemáticas de análise relativamente simples, se comparadas à de outras distribuições. Ademais, o teorema central do limite ⁹ afirma que quando se aumenta a quantidade de amostras de um evento, a distribuição amostral da sua média aproxima-se cada vez mais de uma distribuição normal. Como foi visto no Capítulo 3, os algoritmos de aprendizado de máquina são fundamentalmente baseados em modelos matemáticos; portanto, esses algoritmos tendem a performar melhor quando os dados estão distribuídos conforme uma distribuição normal.

A Figura 24 revela que a distribuição amostral dos preços dos apartamentos está bem distante daquilo que seria uma distribuição normal, traçada em preto.

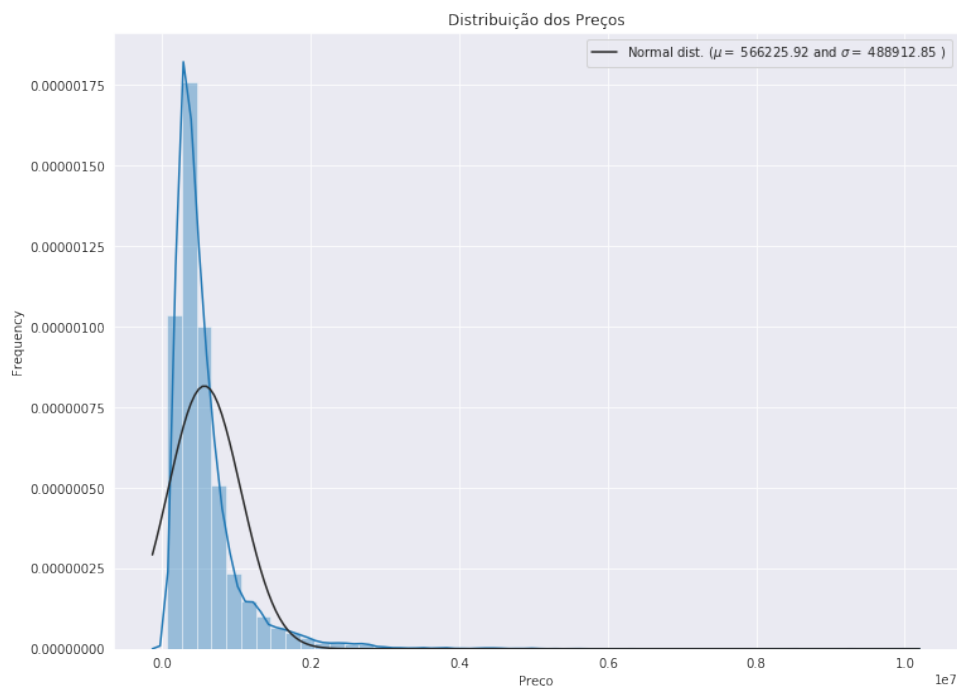


Figura 24 – Distribuição amostral dos preços dos apartamentos

Ao se aplicar uma transformação logarítmica, $\log(1 + x)$, nos preços, a distribuição amostral dos dados se toma a forma de uma distribuição normal, o que é desejado. A Figura 25 exhibe esse resultado.

⁹O Teorema do limite central é um importante resultado da estatística e a demonstração de muitos outros teoremas estatísticos dependem dele. - <https://pt.wikipedia.org/wiki/Teorema_central_do_limite>

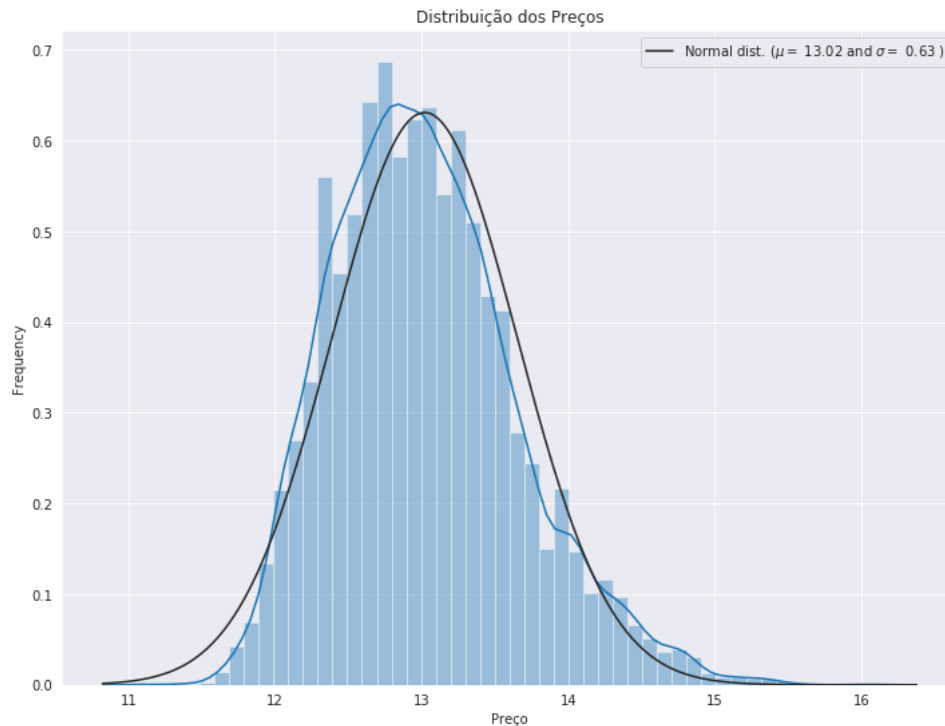


Figura 25 – Distribuição amostral dos preços normalizada

Uma outra análise a ser realizada é com relação a assimetria das distribuições das variáveis independentes. A assimetria (em inglês: *skewness*) mede o quanto a cauda lado esquerdo de uma distribuição é maior do que a cauda do lado direito ou vice-versa. Uma assimetria positiva indica que a cauda do lado direito é maior que a do lado esquerdo e uma assimetria negativa, obviamente, indica o contrário.

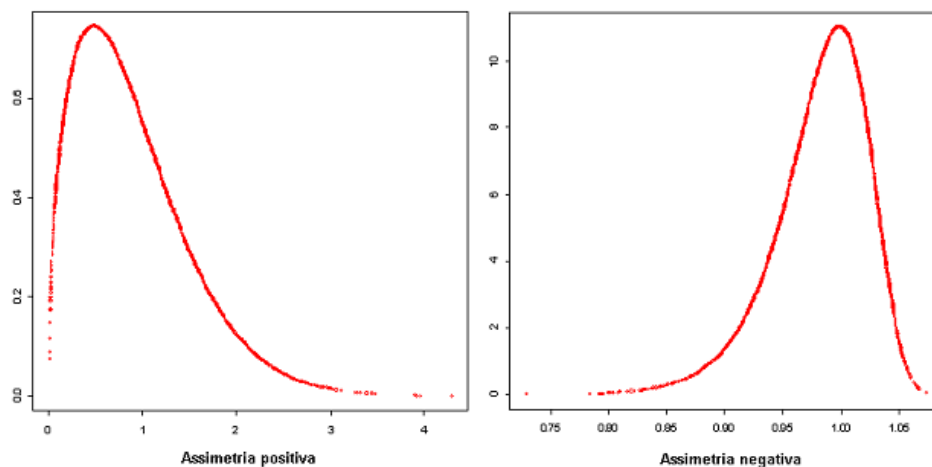


Figura 26 – Conceito de distribuições assimétricas

O coeficiente de assimetria γ de uma distribuição é definido conforme a Equação (32).

$$\gamma = E \left[\left(\frac{X - \mu}{\sigma} \right)^3 \right] = \frac{\mu_3}{\sigma^3} \quad (32)$$

onde μ é a média da distribuição, σ é o desvio padrão, E o operador valor esperado e μ_3 representa o terceiro momento central da distribuição. Felizmente, a biblioteca pandas¹⁰ da linguagem Python oferece uma função chamada *skew* que permite calcular o coeficiente de assimetria γ de uma distribuição de maneira simples e direta. A Figura 27 revela que as variáveis **valor do IPTU** e **valor do condomínio** estão bem assimétricas.

```
In [120]: 1 # Quantidade de variáveis do conjunto de treino com distribuição assimétrica
2 numeric_feats = train.iloc[:,2:14].dtypes[train.iloc[:,2:14].dtypes != "object"].index
3 # Verificando a assimetria das variáveis do problema
4 skewed_feats = train.iloc[:,2:14][numeric_feats].apply(lambda x: skew(x.dropna())).sort_values(ascending=False)
5 skewness_train = pd.DataFrame({'Assimetria':skewed_feats})
6 # Variáveis com distribuição assimétrica no conjunto de treino
7 skewness_train.head(11)
```

```
Out[120]:
```

| Assimetria | |
|----------------|----------|
| Valor_Cond | 143.6802 |
| Valor_IPTU | 81.6895 |
| Vagas_Garagem | 3.0675 |
| Area | 2.1436 |
| Qtde_Banheiros | 2.1179 |
| Std_PreçoZone | 1.3761 |
| Mean_PreçoZone | 1.3354 |
| Qtde_Suites | 1.1771 |
| Latitude | 0.4521 |
| Qtde_Quartos | 0.0391 |
| Longitude | -0.2153 |

Figura 27 – Avaliação do coeficiente de assimetria das distribuições das variáveis independentes

A transformação **Box-Cox**, formulada pelos estatísticos BOX; COX, é uma técnica de transformação da forma da distribuição amostral de dada variável X . Basicamente, essa técnica faz com que a distribuição dos dados de uma amostra seja normal no domínio do coeficiente ρ . A transformação Box-Cox de uma variável X é definida da seguinte maneira:

$$X(\rho) = \begin{cases} \frac{X^\rho - 1}{\rho} & \text{se } \rho \neq 0 \\ \log(X) & \text{se } \rho = 0 \end{cases} \quad (33)$$

Observe que $\rho = 0$ é o mesmo que se realizar uma transformação logarítmica. Desse modo, aplicar-se-á a transformação Box-Cox com $\rho = 0.5$ nas variáveis dependentes mostradas na Figura 27 que possuem coeficiente de assimetria maior do que $|1|$. A Figura 28 mostra que a

¹⁰Pandas é um software biblioteca escrito em Python para manipulação e análise de dados. <[https://en.wikipedia.org/wiki/Pandas_\(software\)](https://en.wikipedia.org/wiki/Pandas_(software))>

transformação Box-Cox reduziu significativamente a assimetria de todas as distribuições, embora algumas continuaram bastante assimétricas.

```
In [125]: 1 from scipy.special import boxcox1p
2 # Seleciona as variáveis que possuem coeficiente de assimetria maior do 1
3 skewness_train = skewness_train[abs(skewness_train.Assimetria)>1]
4 # Realiza a transformação Box Cox utilizando lambda = 0.5
5 skewed_features = skewness_train.index
6 lam = 0.5
7 for feat in skewed_features:
8     #all_data[feat] += 1
9     train[feat] = boxcox1p(train[feat], lam)
10
11 numeric_feats = train.iloc[:,2:14].dtypes[train.iloc[:,2:14].dtypes != "object"].index
12 # Verificando a assimetria das variáveis do problema
13 skewed_feats = train.iloc[:,2:14][numeric_feats].apply(lambda x: skew(x.dropna())).
14 sort_values(ascending=False)
15 skewness_train = pd.DataFrame({'Assimetria':skewed_feats})
16 # Variáveis com distribuição assimétrica no conjunto de treino
17 skewness_train.head(11)
```

```
Out[125]:
```

| | Assimetria |
|----------------|------------|
| Valor_Cond | 94.4535 |
| Valor_IPTU | 34.7510 |
| Area | 1.1602 |
| Qtde_Banheiros | 0.9230 |
| Std_PreçoZone | 0.8303 |
| Mean_PreçoZone | 0.7803 |
| Vagas_Garagem | 0.5369 |
| Latitude | 0.4521 |
| Qtde_Suites | 0.3196 |
| Qtde Quartos | 0.0391 |
| Longitude | -0.2153 |

Figura 28 – Coeficiente de assimetria das distribuições das variáveis dependentes após a transformação Box-Cox

4.4 Definindo uma estratégia de *cross-validation*

O principal objetivo de se utilizar algoritmos de aprendizado de máquina em problemas da vida real é obter um modelo computacional que seja capaz de produzir respostas eficazes para situações novas de um mesmo problema. Em outras palavras, a relevância dos resultados dos modelos está atrelada ao desempenho dos mesmos na predição das amostras do conjunto de teste, pois não faz sentido prático predizer os resultados das amostras de treino. Assim, faz-se necessário definir uma estratégia de validação da predição dos preços das 21.232 amostras de imóveis obtidas na Seção 4.2 e ao mesmo tempo usá-las como dados de treino dos modelos.

Uma maneira de se resolver tal problema é usar a técnica denominada *k-fold Cross Validation* (JAMES et al., 2014), popularmente conhecida entre os cientistas de dados. Basicamente, essa técnica consiste em dividir as amostras aleatoriamente em k grupos, ou dobras (em inglês: *folds*), de tamanho aproximado. Então, o primeiro grupo k_1 de amostras é usado como conjunto de validação e os $k - 1$ grupos restantes são usados como conjunto de treino. O erro

quadrático médio (EQM) (em inglês: *mean squared error* ou MSE) do experimento é computado e armazenado. Esse processo é repetido k vezes e, em cada vez, um grupo diferente de amostras k_i é tratado como conjunto de validação e os restantes $k_i - 1$ grupos são usados como conjunto de treino. Esse procedimento produz então k estimativas de erros de testes, $MSE_1, MSE_2, \dots, MSE_k$. Desse modo, a estimativa de erro do procedimento k -fold *Cross Validation* é definida pela média dos MSE_k erros computados em cada experimento, conforme a Equação (34). A Figura 31 ilustra o procedimento apresentado anteriormente.

$$MSE = \frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2$$

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^k MSE_i$$
(34)

Onde y_i representa o preço real do i -ésimo apartamento e \hat{y}_i representa o preço calculado pelo modelo.

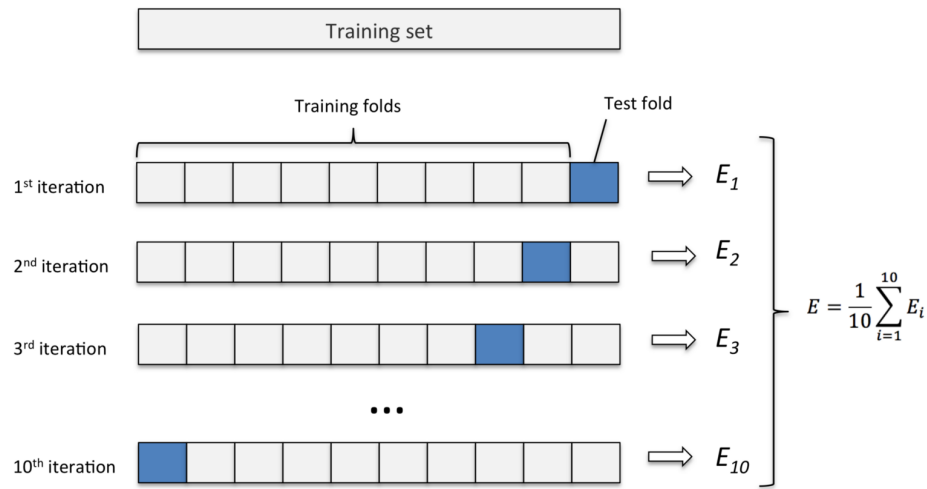


Figura 29 – Ilustração do procedimento k -fold *Cross Validation*

A determinação do número de k grupos a serem criados a partir do conjunto de treino depende de uma avaliação do custo computacional que se deseja "pagar". Na prática, usar $k = 5$ é uma estratégia razoavelmente boa.

4.4.1 Metodologia de avaliação dos modelos

Neste trabalho, os modelos de aprendizado de máquina citados no Capítulo 3 serão avaliados por meio do procedimento de *cross validation* descrito na Seção 4.4, com $k = 5$. Entretanto, ao invés de se computar o erro quadrático médio (MSE), ir-se-á calcular o erro médio percentual absoluto (EMPA) (em inglês: *mean absolute percentage error* ou MAPE) definido na Equação (35).

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (35)$$

O motivo de se utilizar o *MAPE* ao invés do *MSE* ou \sqrt{MSE} é o fato da ordem de grandeza dos preços variar de 10^4 a 10^7 . Assim, interpretar o erro de predição em R\$ ou em R\$² se torna uma tarefa meramente complicada.

Além disso, outros três parâmetros serão utilizados na avaliação dos algoritmos. Eles são: **percentual de amostras com *MAPE* inferior a 10%, percentual de amostras com *MAPE* inferior a 20% e tempo gasto pelo modelo para se ajustar e prever os dados.**

4.5 Parametrização dos modelos

Os modelos de aprendizado de máquina citados no Capítulo 3 foram acessados utilizando a biblioteca *scikit-learn*¹¹ da linguagem Python. A biblioteca possui vários algoritmos de classificação, regressão e agrupamento incluindo as máquinas de vetores de suporte (RVM), *random florest*, *gradient boosting*, *k-means*, *DBSCAN*, entre outros, e é projetada para interagir com as bibliotecas Pandas, NumPy e SciPy também da linguagem Python.

- **Regressão Linear:** Introduziu-se ao modelo a função *RobustScaler()* que realiza uma transformação nos dados usando a amplitude interquartil¹² da distribuição amostral, definida na Equação (36), a fim de tornar o modelo de regressão mais robusto a *outliers*.

$$x_i \leftarrow \frac{x_i - Q_1(x)}{Q_3(x)} \quad (36)$$

onde $Q_1(x)$ é a média das $\frac{n}{2}$ menores amostras, $Q_3(x)$ é a média das $\frac{n}{2}$ maiores amostras e n é a quantidade de dados no conjunto.

¹¹Artigo do Wikipédia com detalhes a respeito da biblioteca *scikit-learn*: <https://pt.wikipedia.org/wiki/Scikit-learn>

¹²O intervalo interquartil (IIQ) foi desenvolvido no âmbito da estatística a fim de avaliar o grau de espalhamento de dados (dispersão) em torno da medida de centralidade - https://pt.wikipedia.org/wiki/Amplitude_interquartil

```
In [153]: 1 from sklearn.linear_model import LinearRegression
2 linear_regression = make_pipeline(RobustScaler(),
3                                   LinearRegression(fit_intercept=True, normalize=False,
4                                                   copy_X=True, n_jobs=None))
```

Figura 30 – Parametrização do modelo de regressão linear

- **Lasso:** Após uma série de tentativas, observou-se que $\lambda = 0.01$, neste caso representado por α , produziu os melhores resultados. Assim como na regressão linear, utilizou-se a função *RobustScaler()* para tornar o modelo robusto a *outliers*.

```
In [156]: 1 lasso = make_pipeline(RobustScaler(), Lasso(alpha=0.01, random_state=1))
```

Figura 31 – Parametrização do modelo lasso

- **Regressão de ridge:** Do mesmo modo que no modelo lasso, parametrizou-se $\lambda = 0.01$, neste caso representado por α . A função de base ϕ utilizada é polinomial.

```
In [91]: 1 KRR = KernelRidge(alpha=0.01, kernel='polynomial', degree=2, coef0=2.5)
```

Figura 32 – Parametrização do modelo regressão ridge

- **Rede de perceptrons de multicamadas:** Definiu-se o número de camadas escondidas igual a 1000 e a função a unidade linear retificada como função de ativação dos neurônios. Ademais, definiu-se o método de otimização estocástico *gradient-based* (KINGMA; BA, 2014) para ajustes dos pesos w dos neurônios.

```
In [118]: 1 mlp = MLPRegressor(hidden_layer_sizes=(1000,), activation='relu', solver='adam', alpha=0.0001,
2                       batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5,
3                       max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False,
4                       warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False,
5                       validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10)
6
```

Figura 33 – Parametrização da rede de perceptrons de multicamadas

- **Gradient boosting machine:** Definiu-se o número de árvores igual a 3000, o parâmetro de redução (*shrinkage parameter*) $\lambda = 0.01$ e máxima quantidade de divisões por árvore igual $d = 6$.

```
In [137]: 1 GBoost = GradientBoostingRegressor(n_estimators=3000, learning_rate=0.1,
2                                           max_depth=6, max_features='sqrt',
3                                           min_samples_leaf=15, min_samples_split=10,
4                                           loss='huber', random_state=5)
```

Figura 34 – Parametrização do modelo *gradient tree boosting*

- **Xtreme gradient boosting machine:** Parametrizado conforme o *gradient boosting*. Ressalta-se que esse modelo incorpora o tratamento eficiente de dados esparsos e regularizações de diversos tipo.

```
In [103]: 1 model_xgb = xgb.XGBRegressor(colsample_bytree=0.4603, gamma=0.0468,  
2                                     learning_rate=0.1, max_depth=6,  
3                                     min_child_weight=1.7817, n_estimators=3000,  
4                                     reg_alpha=0.4640, reg_lambda=0.8571,  
5                                     subsample=0.5213, silent=1,  
6                                     random_state = 7, nthread = -1)
```

Figura 35 – Parametrização do modelo *xtreme gradient tree boosting*

- **Light gradient boosting machine:** Parametrizado conforme o *gradient boosting*. Esse modelo demanda menor custo computacional se comparado às outras variações do modelo *gradient boosting*. Além disso, o *LightGBM* (KE et al., 2017) é capaz de suportar grandes conjunto de dados.

```
In [104]: 1 model_lgb = lgb.LGBMRegressor(objective='regression', num_leaves=6,  
2                                     learning_rate=0.05, n_estimators=3000,  
3                                     max_bin = 55, bagging_fraction = 0.8,  
4                                     bagging_freq = 5, feature_fraction = 0.2319,  
5                                     feature_fraction_seed=9, bagging_seed=9,  
6                                     min_data_in_leaf = 6, min_sum_hessian_in_leaf = 11)
```

Figura 36 – Parametrização do modelo *light gradient tree boosting*

5 RESULTADOS E DISCUSSÃO

Ao término dos experimentos, obteve-se um modelo computacional de predição de preços de apartamentos para a cidade de Belo Horizonte que atende os objetivos iniciais do trabalho. Entretanto, alguns aspectos da modelagem realizada podem ser aperfeiçoados a fim de se obter melhores resultados. Neste capítulo serão apresentados os resultados obtidos e a análises dos mesmos.

5.1 Resultados dos experimentos

Foram realizados um total de seis experimentos usando os modelos descritos na Seção 4.5 e seguindo a metodologia abordada no capítulo anterior. Infelizmente, o modelo Regressão Ridge requiriu uma quantidade de memória computacional superior à disponível e seus resultados não puderam ser avaliados. A distribuição do erro absoluto percentual de cada experimento pode ser vista na Figura 37.

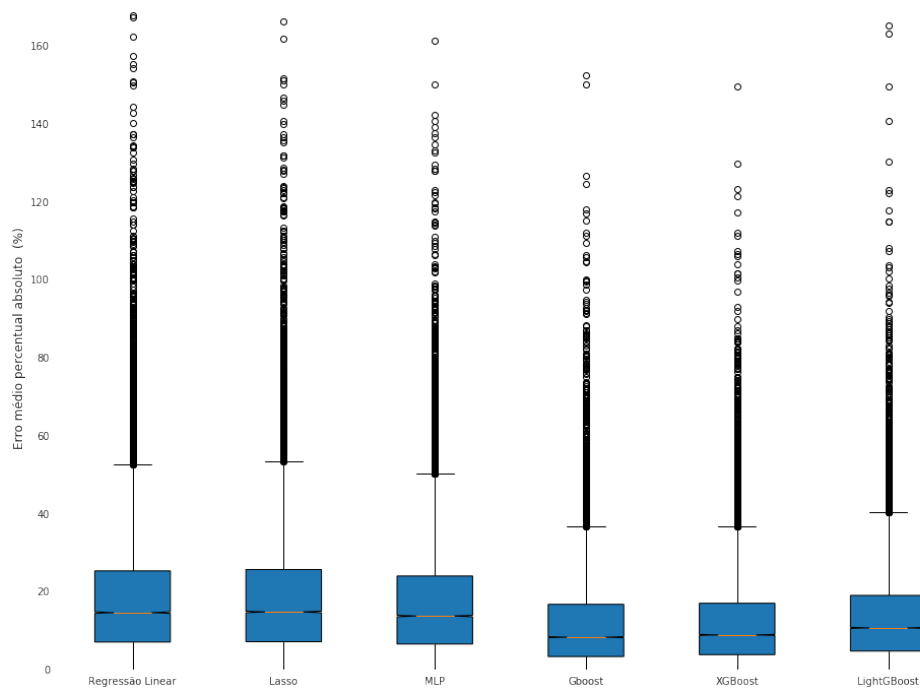


Figura 37 – Distribuição do erro médio percentual absoluto por experimento

A Tabela 5 mostra os resultados de cada experimento utilizando as métricas descritas na Subseção 4.4.1.

| Modelo | MAPE | Percentual de amostras | | Tempo de processamento |
|------------------|--------|------------------------|----------------------|------------------------|
| | | Dentro de $\pm 10\%$ | Dentro de $\pm 20\%$ | |
| Regressão Linear | 18,55% | 36,05% | 64,79% | < 0 segundos |
| Lasso | 18,60% | 36,05% | 64,30% | < 0 segundos |
| Regressão Ridge | - | - | - | - |
| MLP | 17,36% | 37,76% | 66,96% | 36 segundos |
| GBoost | 11,81% | 56,23% | 81,65% | 42 segundos |
| XGBoost | 12,15% | 54,53% | 81,17% | 13 segundos |
| LightGBost | 13,61% | 47,98% | 77,22% | 1 segundo |

Tabela 5 – Resumo dos resultados do experimentos

Pode-se concluir, portanto, que os modelos *gradient tree boosting* e *xtreme gradient tree boosting* obtiveram os melhores desempenhos; confirmando as afirmações feitas na Subseção 3.6.2 deste trabalho a respeito do sucesso que esses modelos têm obtido em aplicações reais e atuais que utilizam *machine learning*.

Ao se comparar os resultados mostrados na Tabela 5 com os resultados obtidos por NG em sua monografia e mostrados na Tabela 2, conclui-se que a abordagem realizada neste trabalho modelou significativamente melhor a dinâmica do mercado imobiliário da cidade em estudo. Ressalte-se, porém, que neste trabalho foram usadas mais variáveis para descrever cada amostra imobiliária, como por exemplo a quantidade de banheiros, suítes, vagas de garagem, valor do imposto e condomínio.

Por fim, destaca-se que o modelo *gradient tree boosting* obteve um erro médio absoluto percentual muito próximo de 10%, o que qualifica esse modelo para ser usado em uma aplicação real de geração de informação à profissionais do segmento imobiliário da cidade de Belo Horizonte.

5.2 Sugestão de implementações que podem obter melhores resultados

Como sugestão de abordagens, do problema apresentado neste trabalho, que podem atingir resultados melhores do àqueles aqui obtidos, cita-se:

1. A implementação de um modelo *gradient tree boosting* para cada sub-região mostrada na Figura 22.
2. A implementação de um modelo *gradient tree boosting* para cada segmento de apartamento, onde esses segmentos são determinados em função da área dos imóveis.
3. O empilhamento dos modelos de algoritmos de aprendizado de máquina para se criar um meta-modelo. Esse meta-modelo funcionaria, basicamente, em duas etapas. Na primeira

etapa o conjunto de dados é dividido em k folds, treinado e predito por k modelos. Numa segunda etapa um outro meta modelo é treinado a fim de mapear as previsões dos k modelos com os valores reais. A Figura 38 ilustra o procedimento descrito.

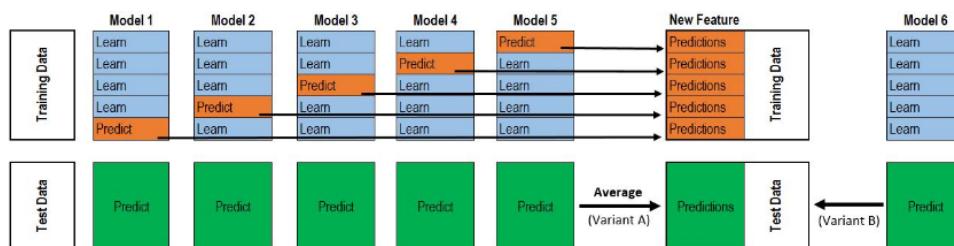


Figura 38 – Ilustração do procedimento de empilhamento de modelos

- O treinamento de modelos por raio de proximidade. Resumidamente, após fornecimento das coordenadas geográficas do imóvel que se deseja precificar, essa abordagem de predição treinaria um modelo com todos os dados que estão, por exemplo, dentro de um raio de 3km em relação ao ponto desejado.

6 CONCLUSÕES

A precificação imobiliária é um problema complexo que não possui um modelo majoritariamente aceito ou utilizado que o descreva razoavelmente. Na prática, os profissionais desse segmento precificam imóveis por inferência e utilizando o conhecimento adquirido por eles após um determinado tempo de trabalho na área. Os modelos teóricos existentes se baseiam, em sua grande maioria, no método de regressão linear, que foi também testado neste trabalho e apresentou um desempenho inferior comparado aos outros métodos testados. Desse modo, a abordagem realizada neste trabalho, ao utilizar os melhores algoritmos de *machine learning* disponíveis atualmente, consolida uma nova fronteira para solução desse problema. Pois ao término de todos os procedimentos, criou-se uma metodologia de precificação de apartamentos autônoma e válida, que pode ser expandida para outros tipos de imóveis, cidades brasileiras e ao segmento de locação.

6.1 Possíveis aplicações e futuros trabalhos

Após a análise da metodologia e dos resultados aqui apresentados, abre-se uma vasta possibilidade de aplicações e produtos que podem ser desenvolvidos a partir deste trabalho. A seguir, lista-se algumas dessas aplicações:

- **Aplicativo precificador de imóveis:** Uma aplicação simples da metodologia desenvolvida neste trabalho seria sua incorporação num aplicativo para dispositivos móveis a fim de se fornecer à sociedade em geral uma ferramenta de referência para precificação de imóveis.
- **Classificador de *outliers*:** Uma vez obtido um modelo inteligente e válido para precificação de imóveis, pode-se utilizá-lo na classificação automática de amostras que estão precificadas muito abaixo ou muito acima de um valor x esperado. Essa informação pode ser útil à investidores do setor.
- **Precificador de automóveis usados:** Automóveis, no geral, são ativos mais voláteis do que imóveis. Assim, a metodologia de extração de dados, treinamentos de modelos de aprendizado de máquina e predição de preços aqui desenvolvida; pode ser perfeitamente aplicada num dispositivo autônomo de precificação de carros. Esse dispositivo poderia ser usado, entre outras coisas, como alternativa à tabela FIPE ¹ de automóveis.

¹ A Fundação Instituto de Pesquisas Econômicas - FIPE é um órgão de apoio institucional ao Departamento de Economia da Faculdade de Economia, Administração e Contabilidade (FEA) da Universidade de São Paulo (USP).

Referências

- ABNT. **NORMA BRASILEIRA ABNT NBR 14653-2 - Avaliação de bens Parte 2: Imóveis urbanos**. Primeira edição. Av. Treze de Maio, 13 – 28º andar 20003-900 – Rio de Janeiro – RJ Tel.: + 5 21 3974-2300 Fax: + 5 21 2220-1762 abnt@abnt.org.br w.abnt.org.br, 2004. Citado 5 vezes nas páginas 4, 7, 9, 10 e 12.
- ALPAYDIN, E. **Introduction to Machine Learning**. 2nd. ed. [S.l.]: The MIT Press, 2010. ISBN 026201243X, 9780262012430. Citado na página 17.
- BENNETT, J.; LANNING, S. The netflix prize. In: **Proceedings of the KDD Cup Workshop 2007**. New York: ACM, 2007. p. 3–6. Disponível em: <<http://www.cs.uic.edu/~liub/KDD-cup-2007/NetflixPrize-description.pdf>>. Citado na página 30.
- BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387310738. Citado 2 vezes nas páginas 17 e 18.
- BITROS, G. C.; PANAS, E. E. Measuring product prices under conditions of quality change: The case of passenger cars in greece. **The Journal of Industrial Economics**, Wiley, v. 37, n. 2, p. 167–186, 1988. ISSN 00221821, 14676451. Disponível em: <<http://www.jstor.org/stable/2098563>>. Citado na página 6.
- BOX, G. E. P.; COX, D. R. An analysis of transformations. **Journal of the Royal Statistical Society. Series B (Methodological)**, p. 211–252, 1964. Citado na página 41.
- BREIMAN, L. et al. **Classification and Regression Trees**. [S.l.]: Wadsworth, 1984. ISBN 0-534-98053-8. Citado na página 26.
- BURGES, C. J. C. **From RankNet to LambdaRank to LambdaMART: An Overview**. [S.l.], 2010. Disponível em: <http://research.microsoft.com/en-us/um/people/cburges/tech_reports/MSR-TR-2010-82.pdf>. Citado na página 30.
- CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2016. (KDD '16), p. 785–794. ISBN 978-1-4503-4232-2. Disponível em: <<http://doi.acm.org/10.1145/2939672.2939785>>. Citado na página 30.
- CYBENKO, G. Approximation by superpositions of a sigmoidal function. **Mathematics of Control, Signals, and Systems (MCSS)**, Springer London, v. 2, n. 4, p. 303–314, dez. 1989. ISSN 0932-4194. Disponível em: <<http://dx.doi.org/10.1007/BF02551274>>. Citado na página 24.
- DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification**. 2. ed. New York: Wiley, 2001. ISBN 978-0-471-05669-0. Citado na página 24.
- FAVERO, L. P. L.; BELFIORE, P. P.; LIMA, G. A. S. F. Modelos de precificação hedônica de imóveis residenciais na região metropolitana de são paulo: uma abordagem sob as perspectivas da demanda e da oferta. **Estudos Econômicos (São Paulo)**, scielo, v. 38, p. 73 – 96, 03 2008. ISSN 0101-4161. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-41612008000100004&nrm=iso>. Citado na página 6.

FOLLAIN, J. R.; JIMENEZ, E. The demand for housing characteristics in developing countries. **Urban Studies**, v. 22, n. 5, p. 421–432, 1985. Disponível em: <<https://doi.org/10.1080/00420988520080731>>. Citado 3 vezes nas páginas , 6 e 7.

FREEMAN, A. **The Measurement of Environmental and Resource Values: Theory and Methods**. Resources for the Future, 1993. (An RFF Press book). ISBN 9781891853623. Disponível em: <<https://books.google.com.br/books?id=Ur3tBatIV4IC>>. Citado na página 6.

FREEMAN, A. M. Hedonic prices, property values and measuring environmental benefits: A survey of the issues. **The Scandinavian Journal of Economics**, [Wiley, Scandinavian Journal of Economics], v. 81, n. 2, p. 154–173, 1979. ISSN 03470520, 14679442. Disponível em: <<http://www.jstor.org/stable/3439957>>. Citado na página 7.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **Annals of Statistics**, v. 29, p. 1189–1232, 2000. Citado na página 28.

GRUNWALD, P.; SPIRITES, P. Proceedings of the twenty-sixth conference on uncertainty in artificial intelligence (2010). **CoRR**, abs/1205.2597, 2012. Disponível em: <<http://arxiv.org/abs/1205.2597>>. Citado na página 30.

HARVEY, J.; JOWSEY, E. **Urban Land Economics**. Macm, 2004. ISBN 9781403900012. Disponível em: <<https://books.google.com.br/books?id=SxpPAAAAMAAJ>>. Citado na página 1.

HE, X. et al. Practical lessons from predicting clicks on ads at facebook. In: **Proceedings of the Eighth International Workshop on Data Mining for Online Advertising**. New York, NY, USA: ACM, 2014. (ADKDD'14), p. 5:1–5:9. ISBN 978-1-4503-2999-6. Disponível em: <<http://doi.acm.org/10.1145/2648584.2648589>>. Citado na página 30.

HOERL, A. E.; KENNARD, R. W. Ridge regression: Biased estimation for nonorthogonal problems. **Technometrics**, v. 12, p. 55–67, 1970. Citado na página 19.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. **Data Clustering: A Review**. 1999. Citado na página 17.

JAMES, G. et al. **An Introduction to Statistical Learning: With Applications in R**. [S.l.]: Springer Publishing Company, Incorporated, 2014. ISBN 1461471370, 9781461471370. Citado 4 vezes nas páginas 20, 26, 27 e 42.

KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. December 2017. Disponível em: <<https://www.microsoft.com/en-us/research/publication/lightgbm-a-highly-efficient-gradient-boosting-decision-tree/>>. Citado na página 46.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **CoRR**, abs/1412.6980, 2014. Disponível em: <<http://arxiv.org/abs/1412.6980>>. Citado na página 45.

LADEIRA, R. W. S. et al. Automação de diagnóstico para manutenção preditiva baseada em análise de fluídos de equipamentos com machine learning. **ABM Week**, p. 268–279, 2017. Citado 3 vezes nas páginas , 26 e 27.

LANCASTER, K. J. A new approach to consumer theory. v. 74, p. 132–132, 04 1966. Citado na página 4.

MADDISON, D. **The Amenity Value of the Global Climate**. Earthscan Publications, 2001. ISBN 9781853836770. Disponível em: <<https://books.google.com.br/books?id=YUMtPAymtNEC>>. Citado na página 6.

MONTGOMERY, D. C.; PECK, E. A.; VINING, G. G. Introduction to linear regression analysis. **Biometrics**, v. 69, n. 4, p. 1087–1087, 2013. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/biom.12129>>. Citado na página 18.

NG, A. **Machine Learning for a London Housing Price Prediction Mobile Application**. Junho 2015. 48 f. Dissertação (Submitted in partial fulfilment of the requirements for BEng in Electronics and Information Engineering) — Imperial College London, London - UK, 2015. Citado 8 vezes nas páginas , 4, 12, 13, 14, 15, 16 e 48.

ROSEN, S. Hedonic prices and implicit markets: Product differentiation in pure competition. **Journal of Political Economy**, v. 82, n. 1, p. 34–55, 1974. Disponível em: <<https://doi.org/10.1086/260169>>. Citado 5 vezes nas páginas 1, 4, 5, 6 e 7.

ROTHENBERG, J. et al. **The Maze of Urban Housing Markets**. 1. ed. University of Chicago Press, 1991. Disponível em: <<https://EconPapers.repec.org/RePEc:ucp:bkecon:9780226729510>>. Citado na página 2.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation. In: RUMELHART, D. E.; MCCLELLAND, J. L. (Ed.). **Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations**. Cambridge, MA: MIT Press, 1986. p. 318–362. Citado na página 2.

SILVERMAN, B. **Density Estimation for Statistics and Data Analysis**. Taylor & Francis, 1986. (Chapman & Hall/CRC Monographs on Statistics & Applied Probability). ISBN 9780412246203. Disponível em: <<https://books.google.com.br/books?id=e-xsrjsL7WkC>>. Citado na página 17.

VAPNIK, V. N. **The Nature of Statistical Learning Theory**. Berlin, Heidelberg: Springer-Verlag, 1995. ISBN 0-387-94559-8. Citado na página 2.

XIAO, Y. **Urban Morphology and Housing Market**. Springer Singapore, 2016. (Springer Geography). ISBN 9789811027628. Disponível em: <<https://books.google.com.br/books?id=yRuRDQAAQBAJ>>. Citado 2 vezes nas páginas 1 e 7.