

ANÁLISE DE DADOS APLICADA AO MERCADO IMOBILIÁRIO: ESTRATÉGIAS DE PRECIFICAÇÃO E PREVISÃO DE PREÇOS

DATA ANALYSIS APPLIED TO THE REAL ESTATE MARKET: PRICING STRATEGIES AND PRICE FORECASTING

Seity, Eduardo. Pedroso, Matheus.

eduardoseity@hotmail.com, matheusbarbosa039@gmail.com

Professor orientador: Adriano Valério Santos da Silva

Centro Universitário Facens - Sorocaba, SP, Brasil

Fevereiro / 2024.

RESUMO

Este projeto de pesquisa se concentra na aplicação da análise de dados no mercado imobiliário, especialmente em estratégias de precificação e previsão de preços. Utilizando técnicas de programação em Python e web scraping, dados são extraídos de plataformas online de imóveis. As bibliotecas Python, como Pandas, Matplotlib, e Sklearn, são fundamentais para a manipulação, visualização e aplicação de algoritmos de aprendizado de máquina, incluindo Regressão Linear, Árvores de Decisão e Redes Neurais Artificiais. Metodologias de pré-processamento são empregadas para padronização de dados e avaliação de modelos. Este estudo reconhece a limitação geográfica ao mercado imobiliário de Sorocaba, focalizando em propriedades residenciais populares para capturar padrões influenciadores nos valores imobiliários nesse ambiente específico.

Palavras-chave: Ciência de Dados, Mercado Imobiliário, Python.

ABSTRACT

This research project focuses on applying data analysis in the real estate market, particularly in pricing strategies and price forecasting. Utilizing Python programming and web scraping techniques, data is extracted from online real estate platforms. Key Python libraries such as Pandas, Matplotlib, and Sklearn play a crucial role in data manipulation, visualization, and the application of machine learning algorithms including Linear Regression, Decision Trees, and Artificial Neural Networks. Pre-processing methodologies are employed for data standardization and model evaluation. This study acknowledges the geographic limitation to the Sorocaba real estate market, concentrating on popular residential properties to capture influential patterns in property values within this specific setting.

Keywords: Data Science, Real State Market, Python.

1 INTRODUÇÃO

O mercado imobiliário é um setor complexo, influenciado por uma variedade de fatores que afetam os preços dos imóveis. Este projeto de pesquisa, intitulado "Análise de Dados Aplicada ao Mercado Imobiliário: Estratégias de Precificação e Previsão de Preços", busca compreender as principais características que determinam os valores dos imóveis e explorar a viabilidade de prever esses preços com base em dados disponíveis.

Utilizando técnicas avançadas de análise de dados e aprendizado de máquina, este estudo visa não apenas desvendar os fatores determinantes por trás da precificação imobiliária, mas também desenvolver modelos preditivos precisos. Ao empregar métodos de web scraping para coletar dados de diversas plataformas online de imóveis, este projeto extrairá uma ampla gama de informações, desde atributos físicos dos imóveis até dados socioeconômicos e geográficos relevantes.

A linguagem de programação Python, juntamente com bibliotecas especializadas como Pandas, Matplotlib, NumPy e Scikit-learn, servirá como a espinha dorsal técnica deste estudo, possibilitando a manipulação eficiente dos dados, visualização de padrões e construção de modelos de aprendizado de máquina robustos.

Além disso, este projeto reconhece a importância do contexto geográfico, concentrando-se na cidade de Sorocaba e suas áreas circundantes. Ao se restringir a essa região específica, pretende-se capturar nuances locais e características únicas que influenciam os preços dos imóveis, especialmente em bairros e condomínios populares.

Ao longo deste artigo, exploraremos em detalhes as técnicas utilizadas, os resultados obtidos e as implicações práticas dessa análise para os stakeholders do mercado imobiliário, desde corretores e investidores até compradores e vendedores.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste projeto de pesquisa mergulha em conceitos fundamentais relacionados à análise de dados dentro do contexto do mercado imobiliário. Começamos com uma compreensão das técnicas essenciais de análise de dados, cruciais para extrair insights significativos de conjuntos de dados complexos (PENG; MATSUI, 2015).

A análise de dados no mercado imobiliário demanda proficiência em estatística, matemática e programação. Python destaca-se como uma linguagem de programação central, permitindo a manipulação eficiente, processamento e visualização de dados por meio de bibliotecas como Pandas (MCKINNEY, 2017), NumPy (OLIPHANT, 2006), Matplotlib (HUNTER, 2007) e Plotly (SIEVERT; PARMER, 2020).

O uso de técnicas de Web Scraping é indispensável para a coleta de dados provenientes de plataformas online de imóveis. A manipulação desses dados requer habilidades em técnicas de pré-processamento, tais como normalização e padronização, empregando ferramentas como Standard Scaler e bibliotecas Python específicas.

No contexto de aprendizado de máquina (machine learning), exploramos uma variedade de algoritmos, como Regressão Linear, Árvores de Decisão, Random Forest, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Redes Neurais Artificiais (MLP) e K-Means para análise de clusters. Cada algoritmo possui características distintas relevantes para a previsão de preços e estratégias de precificação no mercado imobiliário (HASTIE et al., 2009).

Este projeto também incorpora conceitos de avaliação de modelos, incluindo métricas como Erro Quadrático Médio, Erro Médio Absoluto, R2 Score e Erro Percentual Absoluto Médio, fundamentais para avaliar o desempenho dos modelos de machine learning (BROWNLEE, 2018).

Por fim, o projeto reconhece as limitações inerentes à sua abordagem, principalmente a limitação geográfica à cidade de Sorocaba e o foco em propriedades de caráter popular. Essa delimitação permite uma análise mais aprofundada e específica das dinâmicas de mercado em uma determinada região, buscando capturar padrões e características exclusivas que influenciam os valores dos imóveis nesse contexto.

3 MATERIAIS E MÉTODOS (OU METODOLOGIA)

A fundamentação teórica deste projeto de pesquisa baseia-se em conceitos fundamentais relacionados à análise de dados no contexto do mercado imobiliário. Inicia-se com a compreensão das técnicas de análise de dados, que são essenciais para extrair informações valiosas de conjuntos de dados complexos.

A análise de dados aplicada ao mercado imobiliário exige conhecimento sólido em estatística, matemática e programação. Python emerge como uma linguagem de programação central, permitindo a manipulação eficiente, processamento e visualização de dados por meio de bibliotecas como Pandas, NumPy, Matplotlib e Plotly.

O projeto incorporou diversas etapas de preparação e processamento dos dados para garantir a robustez e confiabilidade das análises.

População dos Dados de Quantidade de Pisos:

Realizamos uma população dos dados na variável 'floors', que continha muitos valores vazios, com base na coluna 'unity types', que fornece descrições como 'SINGLE_STOREY_HOUSE' ou 'TWO_STOREY_HOUSE'. As Figuras 1 e 2 ilustram esse processo, sendo a "Figura 1 - População dos dados de quantidade de pisos" e a "Figura 2 - Antes e depois da população dos dados de número de pisos", respectivamente.

Remoção de Outliers

Outliers foram detectados e removidos com base em critérios estabelecidos para características-chave, como 'usablearea', 'totalarea', 'parkingSpaces', 'price', 'bathrooms' e 'bedrooms'. Este processo de limpeza de dados, incluindo a identificação e exclusão de outliers, foi fundamental para garantir a integridade dos dados. Uma abordagem eficaz para preservar a robustez dos dados envolveu o uso do Robust Scaling, uma técnica que leva em consideração a presença de outliers ao dimensionar as características.

A "Figura 3 - Classificação dos outliers" fornece uma representação visual dos dados excluídos durante esse processo. Essa visualização destaca a importância da identificação e remoção de outliers, contribuindo para a confiabilidade e precisão das análises subsequentes. O uso do Robust Scaling nesse contexto reforça a atenção dada à robustez dos dados, permitindo uma abordagem mais resiliente diante de valores extremos e garantindo resultados mais confiáveis.

Inclusão da Coluna 'Condomínio'

Uma coluna 'condomínio' foi incluída, preenchida com base em uma classificação padrão do site, além da busca pela palavra 'condominio' em diferentes formas nas descrições.

Transformação da Coluna 'Amenities'

No decorrer do nosso processo de pré-processamento, implementamos uma abordagem mais detalhada em relação à coluna de texto 'amenities'. Para melhorar a utilidade dessa informação nos modelos subsequentes, desenvolvemos uma função personalizada que realiza a transformação dessa coluna. Essa transformação consiste na criação de colunas individuais, cada uma representando a presença ou ausência de amenidades específicas, tais como piscina, ar condicionado e churrasqueira.

Essa estratégia não apenas simplifica a representação das amenidades, mas também permite que o modelo faça uso mais eficiente dessas informações durante o treinamento. Ao criar colunas individuais para cada amenidade, estamos promovendo uma abordagem mais granular e informativa, o que pode potencialmente melhorar a capacidade preditiva do modelo em relação às características específicas desejadas pelos usuários, como a presença de comodidades específicas em uma propriedade. Essa personalização na transformação dos dados contribui para aprimorar a qualidade e a relevância das características consideradas durante o treinamento dos modelos subsequentes.

Inclusão da Mediana de Casa por Bairro

Adicionalmente, incluímos uma coluna que representa a mediana do preço das casas por bairro, cuidadosamente calculada separadamente para os conjuntos de treino e teste. A "Figura 4 - Mediana de casa por bairro" evidencia

essa inclusão.

Por fim, a "Figura 5" representa o dataset inicial, oferecendo uma visão geral do estado original do conjunto de dados antes das intervenções e melhorias realizadas. Essas etapas de preparação são cruciais para garantir a confiabilidade e a validade das análises subsequentes.

Divisão entre Treino e Teste

A divisão do conjunto de dados entre treino e teste foi realizada para avaliação do desempenho dos modelos de machine learning. A "Figura 6 – Dataset de treino" e a "Figura 7 - Dataset de teste" apresentam visualmente os conjuntos de dados utilizados.

Balanceamento entre Teste e Treino

A "Figura 8" demonstra o balanceamento entre teste e treino, evidenciando visualmente que o equilíbrio foi alcançado em relação a cada uma das features.

O projeto também incorpora o uso do PyCaret, uma ferramenta poderosa para simplificar e automatizar o processo de experimentação com modelos de machine learning. O PyCaret proporciona uma abordagem eficiente para a seleção, treinamento e avaliação de modelos, permitindo uma análise mais abrangente e rápida.

Além disso, o projeto inclui conceitos de avaliação de modelos, como métricas de desempenho (Erro Quadrático Médio, Erro Médio Absoluto, R2 Score e Erro Percentual Absoluto Médio), fundamentais para avaliar o desempenho dos modelos de machine learning. O reconhecimento das limitações, como a geográfica à cidade de Sorocaba e o foco em propriedades populares, permite uma análise mais aprofundada e específica das dinâmicas de mercado.

Deploy do Modelo

Após a fase de treinamento e validação dos modelos de regressão, o próximo passo crucial é o deploy do modelo. O deploy refere-se à implementação prática do modelo treinado em um ambiente operacional para que ele possa ser acessado e utilizado em produção.

Neste trabalho, os modelos selecionados - Random Forest (rf), Light Gradient Boosting Machine (lightgbm) e Extra Trees Regressor (et) - foram salvos em arquivos específicos após o treinamento, utilizando a biblioteca Scikit-learn. Esses arquivos foram armazenados em um diretório adequado para facilitar o acesso e a recuperação posterior.

O processo de deploy foi facilitado pela criação de uma aplicação backend utilizando a biblioteca Flask do Python. Flask é uma estrutura leve e flexível que permite a construção rápida de aplicativos web. A aplicação backend foi

projetada para ser robusta e eficiente, garantindo uma comunicação suave entre o frontend e o backend.

Além disso, foi desenvolvida uma página HTML simples para servir como interface com o usuário. Esta página recebe os dados de entrada do usuário e os transmite para o backend, que utiliza o modelo treinado para fazer previsões. As previsões resultantes são então exibidas ao usuário na mesma página, proporcionando uma experiência fluida e intuitiva.

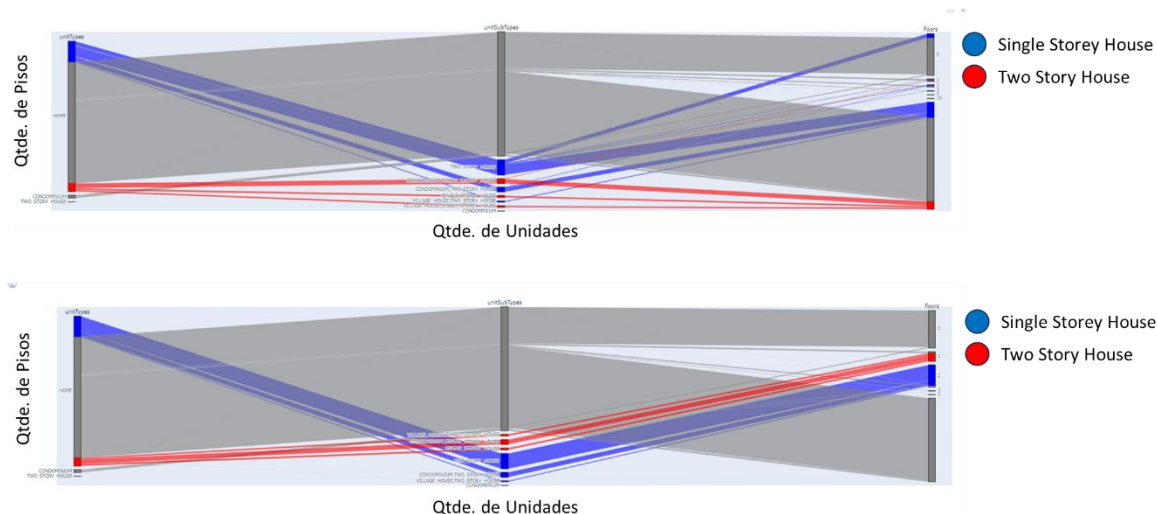
Essa abordagem modular permite uma fácil manutenção e escalabilidade da aplicação.

Em resumo, o deploy do modelo foi realizado de forma eficiente e acessível, garantindo que o modelo treinado pudesse ser facilmente utilizado em um ambiente de produção. Este processo foi fundamental para transformar o trabalho de modelagem em uma solução prática e aplicável no mundo real.

3.1 ILUSTRAÇÕES, QUADROS E TABELAS

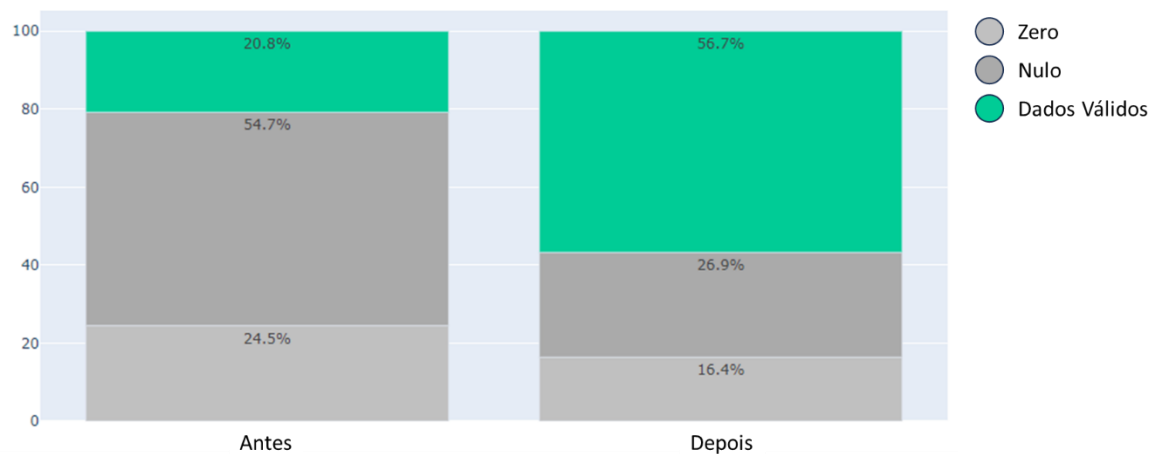
3.2 ILUSTRAÇÃO

Figura 1 - População dos dados de quantidade de pisos



Fonte: elaborado pelos autores

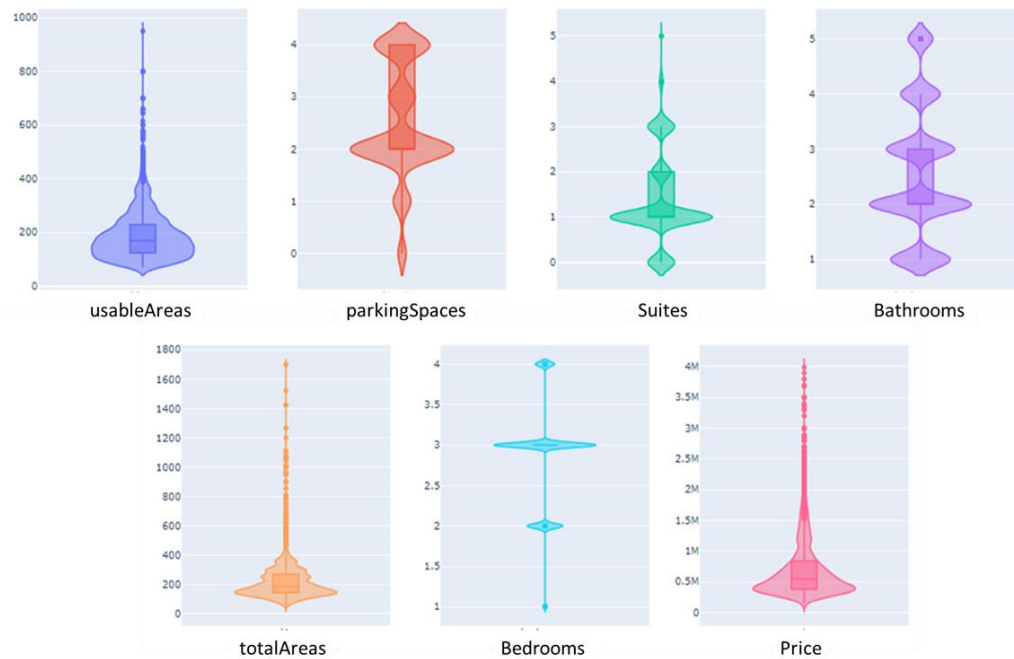
Figura 2 - Antes e depois da população dos dados de número de pisos



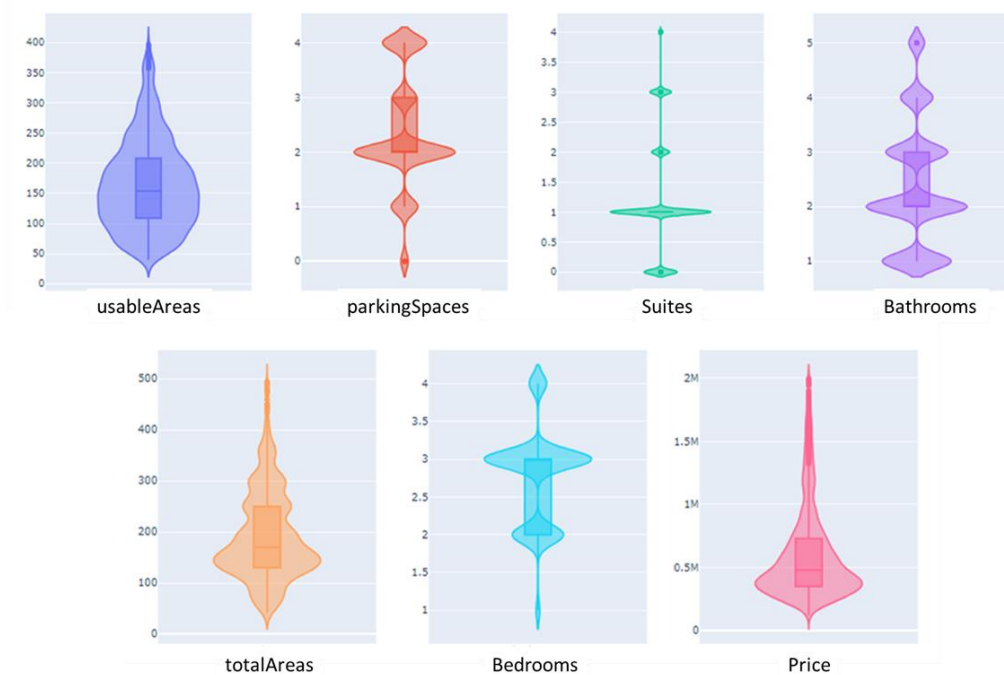
Fonte: elaborado pelos autores

Figura 3 - Classificação dos outliers

ANTES

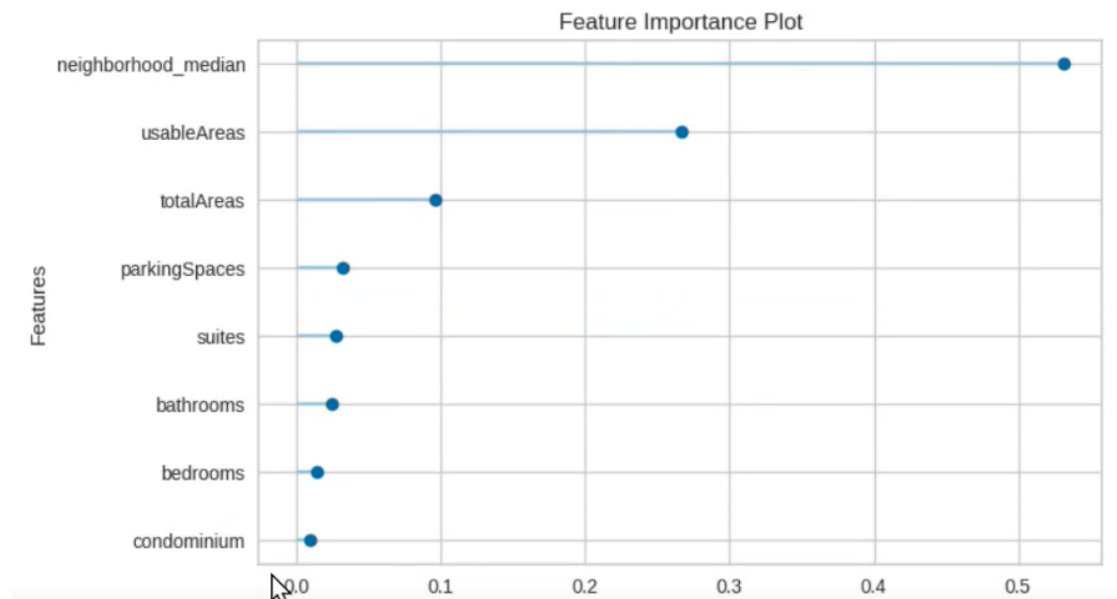


DEPOIS



Fonte: elaborado pelos autores

Figura 4 - Gráfico de importância de características



Fonte: elaborado pelos autores

Figura 5 - Dataset inicial

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16921 entries, 0 to 16920
Data columns (total 24 columns):
#   Column                Non-Null Count  Dtype
---  -
0   state                  16921 non-null  object
1   city                   16921 non-null  object
2   amenities              12846 non-null  object
3   usableAreas            16921 non-null  int64
4   constructionStatus     16921 non-null  object
5   description             16921 non-null  object
6   title                  16921 non-null  object
7   createdAt              16921 non-null  object
8   floors                 5173 non-null   float64
9   unitTypes              16921 non-null  object
10  propertyType           16921 non-null  object
11  unitSubTypes           3143 non-null   object
12  id                     16921 non-null  int64
13  parkingSpaces          16418 non-null  float64
14  street                 12395 non-null  object
15  neighborhood            16921 non-null  object
16  lon                    7416 non-null   float64
17  lat                    7416 non-null   float64
18  suites                 14128 non-null  float64
19  bathrooms              16921 non-null  int64
20  totalAreas             16268 non-null  float64
21  bedrooms               16878 non-null  float64
22  price                  16921 non-null  int64
23  businessType           16921 non-null  object
dtypes: float64(7), int64(4), object(13)
memory usage: 3.1+ MB
```

Fonte: elaborado pelos autores

Figura 6 – Dataset de treino

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3416 entries, 14894 to 3128
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   BARBECUE_GRILL        3416 non-null   int64
1   GYM                    3416 non-null   int64
2   POOL                   3416 non-null   int64
3   BACKYARD               3416 non-null   int64
4   condominium            3416 non-null   int64
5   usableAreas            3416 non-null   float64
6   parkingSpaces          3416 non-null   int64
7   neighborhood_median    3416 non-null   float64
8   suites                 3416 non-null   int64
9   bathrooms              3416 non-null   int64
10  totalAreas             3416 non-null   float64
11  bedrooms               3416 non-null   int64
12  price                  3416 non-null   float64
dtypes: float64(4), int64(9)
memory usage: 373.6 KB
```

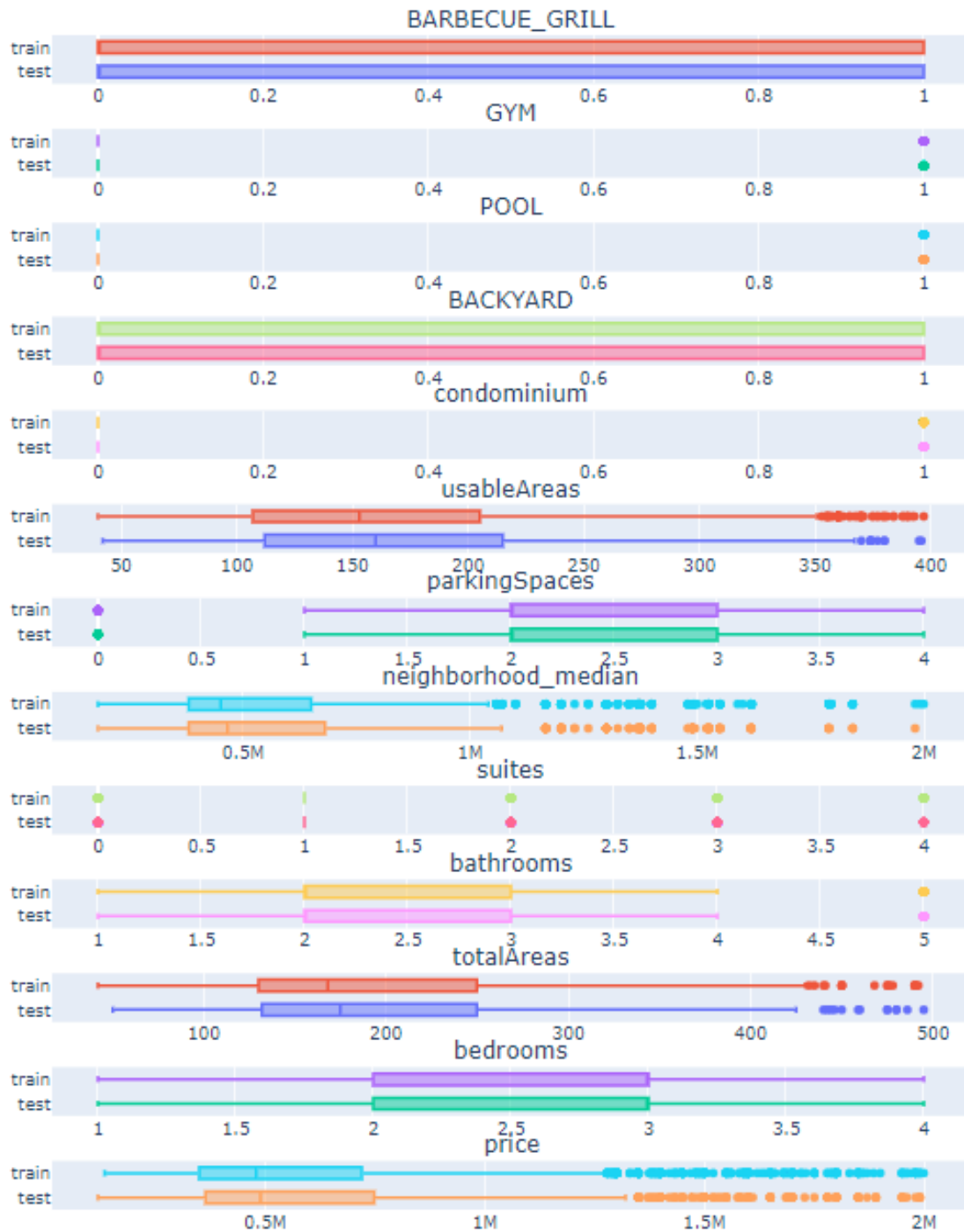
Fonte: elaborado pelos autores

Figura 7 - Dataset de teste

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1450 entries, 9703 to 10462
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   BARBECUE_GRILL        1450 non-null   int64
1   GYM                    1450 non-null   int64
2   POOL                   1450 non-null   int64
3   BACKYARD               1450 non-null   int64
4   condominium            1450 non-null   int64
5   usableAreas            1450 non-null   float64
6   parkingSpaces          1450 non-null   int64
7   neighborhood_median    1450 non-null   float64
8   suites                 1450 non-null   int64
9   bathrooms              1450 non-null   int64
10  totalAreas             1450 non-null   float64
11  bedrooms               1450 non-null   int64
12  price                  1450 non-null   float64
dtypes: float64(4), int64(9)
memory usage: 158.6 KB
```


Fonte: elaborado pelos autores

Figura 8 - Balanceamento do dataset (treino x teste)



Fonte: elaborado pelos autores

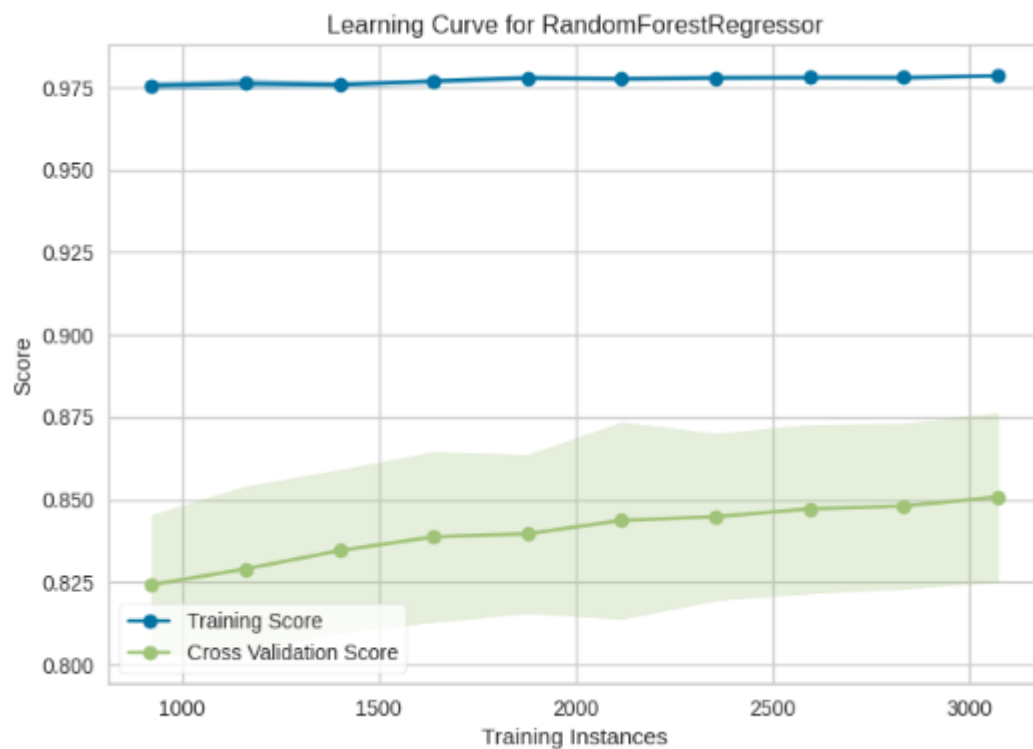
Figura 9 - Descrição das transformações do Pycaret



	Description	Value
0	Session id	42
1	Target	price
2	Target type	Regression
3	Original data shape	(4866, 13)
4	Transformed data shape	(4866, 13)
5	Transformed train set shape	(3416, 13)
6	Transformed test set shape	(1450, 13)
7	Numeric features	12
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Normalize	True
13	Normalize method	robust
14	Fold Generator	KFold
15	Fold Number	10
16	CPU Jobs	-1
17	Use GPU	False
18	Log Experiment	False
19	Experiment Name	reg-default-name
20	USI	92ba

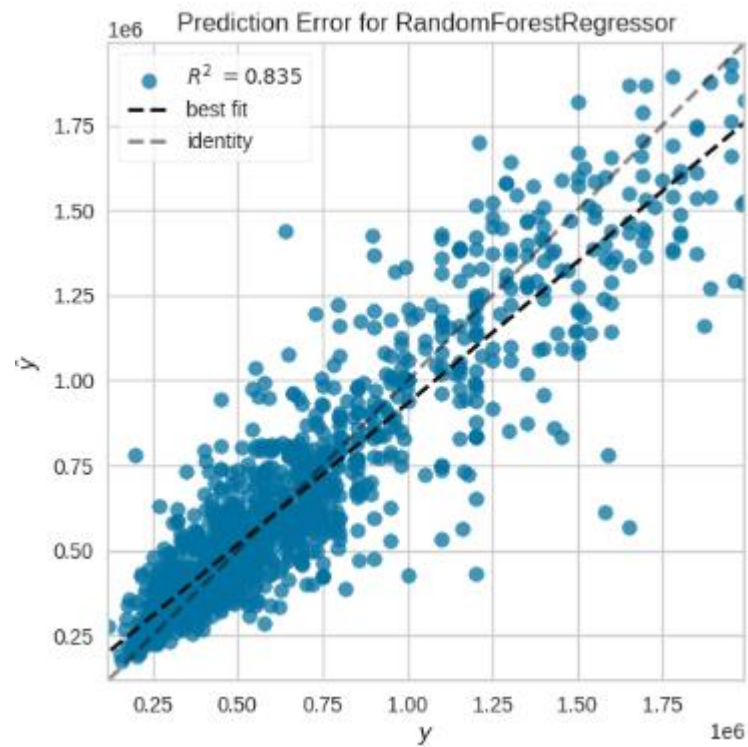
Fonte: elaborado pelos autores

Figura 10 - Curva de Aprendizado Random Forest



Fonte: elaborado pelos autores

Figura 101 – Gráfico de erro de previsão para Random Forest



Fonte: elaborado pelos autores

3.3 QUADROS

3.4 TABELAS

Tabela 11 – Resultados do Pycaret

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
rf	Random Forest	920.764.587	198.008.255.422.583	1.403.571.639	0.8511	0.2149	0.1621	10.290
lightgbm	Light Gradient	953.700.687	207.051.725.442.215	1.436.448.814	0.8445	0.2165	0.1665	0.8250
et	Extra Trees Regressor	939.274.208	209.256.910.024.200	1.444.128.712	0.8429	0.2177	0.1634	0.7730
gbr	Gradient Boosting	983.228.040	220.397.168.045.944	1.481.979.879	0.8345	0.2231	0.1732	0.3570
xgboost	Extreme Gradient	971.542.578	220.591.292.416.000	1.482.967.422	0.8342	0.2248	0.1702	0.1980
llar	Lasso Least Angle	1.107.305.633	253.960.331.264.000	1.590.289.359	0.8100	0.2634	0.2044	0.0320
br	Bayesian Ridge	1.107.017.773	253.923.946.496.000	1.590.157.016	0.8100	0.2633	0.2044	0.0400
lr	Linear Regression	1.107.310.000	253.960.361.984.000	1.590.289.547	0.8100	0.2634	0.2044	0.0340
lar	Least Angle Regression	1.107.310.109	253.960.368.128.000	1.590.289.578	0.8100	0.2634	0.2044	0.0330
ridge	Ridge Regression	1.107.249.570	253.951.543.296.000	1.590.258.328	0.8100	0.2634	0.2044	0.0320
lasso	Lasso Regression	1.107.305.555	253.960.321.024.000	1.590.289.328	0.8100	0.2634	0.2044	0.0340
knn	K Neighbors Regressor	1.073.870.602	261.714.522.112.000	1.614.746.781	0.8040	0.2432	0.1856	0.0460
huber	Huber Regressor	1.070.335.261	265.960.985.697.216	1.626.142.406	0.8013	0.2502	0.1904	0.0540
par	Passive Aggressive	1.069.109.937	271.383.958.665.744	1.642.240.608	0.7973	0.2507	0.1887	0.4160
en	Elastic Net	1.191.885.086	294.822.494.208.000	1.712.836.172	0.7802	0.2600	0.2172	0.0310
dt	Decision Tree	1.225.257.877	377.705.485.507.129	1.939.530.312	0.7169	0.2964	0.2135	0.0440
omp	Orthogonal Matching	1.334.461.430	435.766.116.352.000	2.082.066.047	0.6729	0.3194	0.2528	0.0320
ada	AdaBoost Regressor	2.104.282.553	580.716.727.477.223	2.407.877.953	0.5615	0.4543	0.5034	0.1910
dummy	Dummy Regressor	2.754.890.438	1.349.358.387.200.000	3.667.577.125	-0.0045	0.5533	0.5518	0.0270

4 RESULTADOS E DISCUSSÃO

Os resultados obtidos por meio do PyCaret fornecem uma visão abrangente do desempenho de diversos modelos de regressão na previsão de preços de imóveis. As métricas de avaliação foram cuidadosamente analisadas para entender a eficácia de cada algoritmo no contexto específico deste projeto.

Os modelos de regressão foram avaliados com base em métricas como Erro Médio Absoluto (MAE), Erro Quadrático Médio (MSE), Raiz do Erro Quadrático Médio (RMSE), Coeficiente de Determinação (R2), Erro Médio Percentual Absoluto (MAPE) e Tempo de Treinamento (TT). Os resultados detalhados podem ser consultados na “Tabela 1 – Resultados do Pycaret”.

Os modelos de regressão apresentaram desempenhos variados, destacando-se alguns algoritmos como mais promissores para a tarefa em questão. O Random Forest Regressor mostrou um desempenho robusto, com um coeficiente de determinação (R2) de 0.8511, indicando uma excelente capacidade de explicar a variação nos preços dos imóveis. O Extra Trees Regressor também demonstrou resultados sólidos, com métricas comparáveis às do Random Forest. A escolha entre esses dois modelos pode depender de considerações adicionais, como interpretabilidade e tempo de treinamento. O

Light Gradient Boosting Machine também apresentou um desempenho sólido, embora ligeiramente inferior aos modelos Random Forest e Extra Trees. No entanto, sua eficácia deve ser considerada em conjunto com outros fatores, como eficiência computacional e interpretabilidade.

Transformações do PyCaret e Análise Adicional:

No contexto do PyCaret, a "Figura 9 - Descrição das transformações do PyCaret" nos proporciona uma visão detalhada das etapas de pré-processamento realizadas pelo sistema, destacando as transformações aplicadas aos dados antes do treinamento dos modelos.

Ao analisarmos a "Tabela 1 - Resultados dos testes dos modelos PyCaret", deparamo-nos com uma representação visual dos resultados obtidos. Essa visualização permite uma comparação intuitiva entre os diferentes modelos e suas métricas associadas, oferecendo insights valiosos sobre o desempenho relativo de cada abordagem.

Além disso, a "Figura 10 - Curva de Aprendizado Random Forest" lança luz sobre o índice de acerto 'score' em relação ao número de instâncias de treino, proporcionando uma compreensão mais profunda do comportamento do modelo Random Forest ao longo do processo de aprendizado.

Dentro do ecossistema PyCaret, a "Figura 9 - Descrição das transformações do PyCaret" oferece uma visão minuciosa das etapas de pré-processamento executadas pelo sistema. Ela destaca as transformações aplicadas aos dados antes do treinamento dos modelos, proporcionando um entendimento detalhado das modificações realizadas para aprimorar a qualidade e a relevância das informações.

Ao explorarmos a "Tabela 1 - Resultados dos testes dos modelos PyCaret", nos deparamos com uma representação visual dos resultados obtidos. Essa visualização não é apenas um tópico isolado, mas sim uma ferramenta que facilita a comparação intuitiva entre os diferentes modelos e suas métricas associadas. As informações visuais fornecem insights valiosos sobre o desempenho relativo de cada abordagem, auxiliando na tomada de decisões informadas sobre a escolha do modelo.

Adicionalmente, a "Figura 10 - Curva de Aprendizado Random Forest" amplia nossa compreensão ao lançar luz sobre o índice de acerto ('score') em relação ao número de instâncias de treino. Este gráfico não é apenas um elemento isolado, mas sim uma ferramenta dinâmica que proporciona uma análise mais profunda do comportamento do modelo Random Forest ao longo do processo de aprendizado.

Além disso, a "Figura 11 - Gráfico de Erro de Previsão para o Random Forest" oferece uma representação visual do erro de previsão associado ao modelo Random Forest. Esta visualização fornece uma compreensão mais abrangente da performance do modelo, destacando áreas específicas onde as previsões podem divergir dos valores reais.

Resumidamente, os resultados advindos do PyCaret sugerem que modelos fundamentados em árvores de decisão, como Random Forest e Extra Trees, mostram-se promissores para a previsão de preços de imóveis neste projeto específico. Entretanto, é crucial considerar outros fatores, como interpretabilidade e eficiência computacional, para orientar a escolha do modelo final a ser adotado nas decisões práticas do mercado imobiliário.

5 CONSIDERAÇÕES FINAIS

Este projeto de pesquisa proporcionou uma imersão profunda na análise de dados aplicada ao mercado imobiliário, explorando técnicas avançadas de machine learning para a previsão de preços de imóveis. Ao longo do processo, diversas etapas foram realizadas, desde a coleta e preparação dos dados até a implementação e avaliação de modelos de regressão.

As transformações aplicadas aos dados, evidenciadas na "Figura 9 - Descrição das transformações do PyCaret", destacam a importância do pré-processamento na garantia da qualidade e confiabilidade dos resultados. A inclusão de variáveis, a remoção de outliers e a criação de novas características contribuíram significativamente para a robustez dos modelos desenvolvidos.

Os resultados obtidos pelo PyCaret, conforme apresentado na "Tabela 1 - Resultados dos testes dos modelos PyCaret", revelam insights valiosos sobre o desempenho dos diferentes algoritmos considerados. Modelos baseados em árvores de decisão, como Random Forest e Extra Trees, destacaram-se como opções sólidas para a previsão de preços de imóveis, evidenciando uma capacidade significativa de capturar padrões complexos nos dados.

No entanto, é crucial reconhecer as limitações inerentes a este estudo. A delimitação geográfica à cidade de Sorocaba e o foco em propriedades de caráter popular representam restrições que influenciam a generalização dos resultados para outros contextos. Além disso, o sucesso desses modelos está intrinsecamente ligado à qualidade e representatividade dos dados utilizados.

Como próximos passos, sugere-se a realização de validações adicionais, a expansão do escopo geográfico e a consideração de novas variáveis que possam enriquecer a capacidade preditiva dos modelos. Além disso, a interação contínua com profissionais do mercado imobiliário e especialistas na área pode fornecer insights valiosos para aprimorar ainda mais a aplicabilidade prática desses modelos.

Este estudo representa um passo significativo na direção da utilização de abordagens avançadas de análise de dados para informar e aprimorar as estratégias de precificação no mercado imobiliário. Com o comprometimento contínuo com a qualidade dos dados e a exploração de novas técnicas, espera-se que este trabalho contribua para avanços significativos no entendimento e na previsão de tendências no setor imobiliário.

REFERÊNCIAS

ALENCAR, Sérgio Ricardo Ribeiro. ***Precificação de Imóveis com Machine Learning***. 2022. Dissertação (Mestrado) - Universidade de São Paulo, São Carlos.

ZAGHI, Lucca Magri. ***Modelo de previsão de preços de imóveis na cidade de Florianópolis/SC a partir de técnicas de Machine Learning***. 2023. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Florianópolis.

DATA ZAP. **Índice de vendas no mercado imobiliário**, 2023. Disponível em: <<https://www.datazap.com.br/conteudos-fipezap>>. Acesso em: 25 de nov. 2023.

DATA ZAP. **Tendências de moradia**, 2023. Disponível em: <<https://www.datazap.com.br/tendencias-de-moradia-perfil-comprador>>. Acesso em: 25 de nov. 2023.

Yashmeet SINGH. **Robust Scaling: Why and How to Use It to Handle Outliers**, 2022. Disponível em: <<https://proclusacademy.com/blog/robust-scaler-outliers>>. Acesso em: 27 de nov. 2023.

Felipe AZANK. **Como avaliar seu modelo de regressão**, 2020. Disponível em: <<https://medium.com/turing-talks/como-avaliar-seu-modelo-de-regress%C3%A3o-c2c8d73dab96>>. Acesso em: 25 de nov. 2023.

PENG, Roger D.; MATSUI, Elizabeth. **The Art of Data Science: A Guide for Anyone Who Works with Data**. Springer, 2015. Acesso em: 12 de nov. 2023.

MCKINNEY, Wes. **Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython**. O'Reilly Media, 2017. Acesso em: 12 de nov. 2023.

OLIPHANT, Travis. **Guide to NumPy**. Travis Oliphant, 2006. Acesso em: 19 de nov. 2023.

HUNTER, John D. **Matplotlib: A 2D Graphics Environment**. CreateSpace Independent Publishing Platform, 2007. Acesso em: 27 de nov. 2023.

SIEVERT, Carson; PARMER, Chris. **Interactive Web-Based Data Visualization with R, plotly, and shiny**. CRC Press, 2020. Acesso em: 23 de nov. 2023.

SCIKIT-LEARN. **Documentation on preprocessing**. Disponível em: <https://scikit-learn.org/stable/modules/preprocessing.html>. Acesso em: [inserir data de acesso]. Acesso em: 10 de jan. 2024.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. Springer, 2009. Acesso em: 27 de nov. 2023.

BROWNLEE, Jason. **Better Evaluation Metrics for Classification, And Machine Learning in General**. Disponível em: <https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/>. Acesso em: 05 de jan. 2024.