UNIVERSIDADE FEDERAL DO CEARÁ

IUDE KILDARE, EDICLEITON ALVES

TRABALHO FINAL

CRATEÚS

2025

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos, o setor imobiliário tem passado por transformações significativas, impulsionadas por fatores econômicos, demográficos e sociais. Essa dinâmica gera a necessidade de ferramentas preditivas capazes de oferecer uma estimativa precisa dos valores dos imóveis, contribuindo para tomadas de decisão mais informadas por investidores, corretores e gestores públicos.

O presente trabalho propõe uma investigação baseada em técnicas de aprendizado de máquina, com o objetivo de prever o preço mediano dos imóveis na Malásia para o ano de 2025. Para isso, utilizamos um conjunto de dados composto por informações relevantes, como a localização (com variáveis *Township*, *Area* e *State*), características dos imóveis (como *Tenure* e *Type*), além de indicadores do mercado imobiliário, como o *Median_Price*, o *Median_PSF* (preço por pé quadrado) e o número de *Transactions*.

A metodologia empregada envolve a aplicação de diferentes modelos de regressão, entre eles a Regressão Linear, Árvore de Decisão, Random Forest e Gradient Boosting. Cada um desses algoritmos foi implementado considerando as especificidades do problema. Por exemplo, o pré-processamento incluiu a transformação de variáveis categóricas por meio do One-Hot Encoding e a aplicação de escalonamento (StandardScaler) especificamente para a Regressão Linear, que é sensível à magnitude dos dados. Além disso, a validação dos modelos foi realizada utilizando a técnica de K-Fold Cross-Validation (com 10 partições) associada à otimização de hiperparâmetros por meio de Grid Search, permitindo uma análise comparativa robusta dos desempenhos obtidos.

A motivação desta investigação reside na crescente demanda por previsões acuradas no mercado imobiliário, que é um dos setores mais dinâmicos e complexos da economia. Ao explorar e comparar diferentes abordagens de regressão, o estudo busca identificar as variáveis mais influentes na determinação do valor dos imóveis e aprimorar os métodos de previsão, contribuindo para o avanço das aplicações de inteligência artificial e aprendizado de máquina em contextos reais.

2 FUNDAMENTAÇÃO TEÓRICA

O problema em estudo consiste na previsão do preço mediano dos imóveis na Malásia, caracterizando-se como uma tarefa de regressão. Para isso, utilizamos variáveis relacionadas à localização, características dos imóveis e indicadores do mercado imobiliário. O objetivo é identificar relações entre essas variáveis para estimar o preço das propriedades.

2.1 Modelos de aprendizado de máquina utilizados

Para a previsão do preço mediano dos imóveis, foram aplicados modelos de aprendizado de máquina, incluindo Regressão Linear, Árvore de Decisão, Random Forest e Gradient Boosting. Cada abordagem foi preparada com pré-processamento adequado, como One-Hot Encoding para variáveis categóricas e escalonamento para modelos sensíveis à magnitude dos dados. Além disso, utilizamos otimização de hiperparâmetros por meio de Grid Search e validação por K-Fold Cross-Validation, garantindo uma análise robusta e comparativa do desempenho dos modelos.

2.2 Conjunto de dados e variáveis utilizadas

O conjunto de dados utilizado neste estudo contém informações detalhadas sobre o mercado imobiliário na Malásia, focando em propriedades avaliadas em 2025. Esses dados foram organizados para fornecer um panorama abrangente do setor, permitindo a aplicação de técnicas de aprendizado de máquina para a previsão do preço dos imóveis. A seguir, apresenta-se uma descrição das principais variáveis:

- **Township:** Nome da localidade específica ou do conjunto habitacional onde o imóvel está localizado. Essa variável pode influenciar diretamente os preços.
- **Area:** Bairro ou região dentro do estado onde o imóvel se encontra, permitindo uma análise geográfica detalhada.
- **State:** Estado da Malásia onde o imóvel está situado, como Penang, Johor e Perak. Essa informação é crucial para entender variações regionais no mercado.
- **Tenure:** Tipo de posse do imóvel, podendo ser *Freehold* (propriedade permanente) ou *Leasehold* (com prazo limitado, geralmente 99 anos), o que pode impactar seu valor.
- **Type:** Classificação do tipo de imóvel, como *Terrace House* ou *Cluster House*, refletindo características construtivas e funcionais.

- **Median_Price:** Preço mediano dos imóveis vendidos na localidade. Essa variável é a saída (target) dos modelos de regressão.
- **Median_PSF:** Preço mediano por pé quadrado, indicando o custo por unidade de área.
- **Transactions:** Número total de transações registradas em 2025, refletindo a atividade do mercado e a demanda por imóveis em diferentes regiões.

Essa estrutura de dados permite uma análise aprofundada dos fatores que influenciam o valor dos imóveis e fornece uma base sólida para a aplicação e comparação de diversos modelos de aprendizado de máquina, visando obter previsões mais precisas e confiáveis.

2.3 Revisão de Trabalhos Similares

Estudos anteriores exploraram diversas abordagens de aprendizado de máquina para prever preços de imóveis, utilizando variáveis socioeconômicas, características estruturais e localização geográfica. Trabalhos como os de Antipov e Pokryshevskaya (2012) aplicaram Random Forest para prever preços de apartamentos, destacando a precisão do modelo ao capturar relações não lineares entre variáveis. Já Zhang (2019) utilizaram uma abordagem de empilhamento (stacking) combinando modelos de árvores de decisão e regressão linear, obtendo melhorias significativas na previsão dos valores imobiliários. Esses estudos demonstram a eficiência dos métodos baseados em árvores na modelagem de preços de propriedades.

Além disso, pesquisas como as de Li (2020) e Chen e Hao (2020) exploraram o uso de Gradient Boosting para prever preços imobiliários, mostrando que essa técnica supera abordagens tradicionais ao otimizar a seleção de variáveis e minimizar erros de previsão. Métodos estatísticos mais simples, como a Regressão Linear, também foram investigados por Fan (2018) e Park e Kwon (2017), que analisaram a influência de fatores como localização e infraestrutura urbana no valor das propriedades. Finalmente, a comparação entre diferentes algoritmos foi realizada por Kumar (2021), evidenciando que modelos baseados em aprendizado de máquina, especialmente aqueles com técnicas de boosting e ensemble, oferecem maior precisão e generalização para a previsão de preços imobiliários.

Esses estudos fornecem uma base sólida para o presente trabalho, que busca avaliar o desempenho de diferentes técnicas de aprendizado de máquina na previsão do preço mediano dos imóveis na Malásia. A revisão dessas pesquisas permite identificar abordagens eficazes, bem como desafios comuns, como a necessidade de pré-processamento adequado dos dados e a escolha de modelos que melhor capturam padrões complexos do mercado imobiliário.

3 TRABALHOS RELACIONADOS

Diversos estudos na literatura abordam a previsão de preços de imóveis por meio de técnicas de aprendizado de máquina, empregando diferentes abordagens e conjuntos de dados. A seguir, são apresentados sete trabalhos relevantes para este estudo:

3.1 Modelos Baseados em Árvores de Decisão

O estudo de Antipov e Pokryshevskaya (2012) aplicou a técnica de Random Forest para avaliação em massa de apartamentos residenciais, explorando também o uso da Árvore de Decisão para diagnóstico dos modelos. Os autores demonstraram que o Random Forest oferece alta precisão na previsão de preços, enquanto a interpretação dos fatores determinantes foi aprimorada com a análise das divisões geradas pela árvore. Esse estudo reforça a eficácia dos modelos baseados em árvores para problemas de regressão no mercado imobiliário.

Outro trabalho relevante é o de Zhang (2019), que propôs um modelo de empilhamento (stacking) de dois níveis para avaliação de preços de imóveis na China. A abordagem combinou modelos de árvores de decisão com regressão linear para capturar padrões complexos dos preços residenciais. Os resultados mostraram que essa combinação melhora a precisão da previsão em relação a abordagens tradicionais, destacando a importância de métodos híbridos no setor imobiliário.

3.2 Aplicação de Gradient Boosting para Previsão Imobiliária

O trabalho de Li (2020) explorou o método Gradient Boosting como uma solução eficaz para prever preços de imóveis em mercados complexos. Os autores testaram diversas configurações do algoritmo para otimizar o desempenho do modelo e demonstraram que a técnica supera abordagens tradicionais, como Regressão Linear e Árvores de Decisão simples. A pesquisa concluiu que o Gradient Boosting é particularmente útil para capturar relações não lineares entre variáveis do setor imobiliário.

Outro estudo que utilizou Gradient Boosting foi o de Chen e Hao (2020), que previram índices de preços de imóveis na China. A pesquisa comparou diferentes técnicas de aprendizado de máquina e constatou que o Gradient Boosting apresentou o melhor desempenho na modelagem de preços residenciais, especialmente quando combinado com um pré-processamento eficiente dos dados. Esse estudo valida o uso de algoritmos baseados em

boosting para problemas de regressão no mercado imobiliário.

3.3 Regressão Linear e Modelagem Estatística

O estudo de Fan (2018) analisou os fatores que influenciam os preços dos imóveis em Pequim, utilizando um modelo de Regressão Linear. Os autores identificaram que variáveis como localização, área construída e tempo de posse do imóvel são determinantes no valor de venda. O trabalho destaca a importância da regressão linear como uma abordagem simples, mas ainda relevante, para modelar preços de imóveis.

Além disso, Park e Kwon (2017) realizaram uma pesquisa semelhante, aplicando a regressão linear para prever preços de casas em Pequim. No entanto, diferentemente de Fan (2018), os autores incluíram variáveis socioeconômicas, como renda média da população e infraestrutura urbana, aprimorando a capacidade preditiva do modelo. Os resultados mostraram que a incorporação desses fatores melhora significativamente a precisão das previsões.

3.4 Comparação de Algoritmos de Aprendizado de Máquina

O estudo de Kumar (2021) comparou diversos algoritmos de aprendizado de máquina, incluindo Random Forest, Gradient Boosting e Redes Neurais Artificiais, para prever preços de imóveis. Os autores avaliaram a precisão e eficiência de cada modelo e concluíram que os métodos baseados em árvores de decisão apresentaram melhor equilíbrio entre desempenho e interpretabilidade.

A pesquisa contribui para a escolha de modelos mais adequados a diferentes cenários do mercado imobiliário. Além disso, os autores destacaram a importância do pré-processamento dos dados e da escolha de hiperparâmetros para maximizar a precisão dos modelos, enfatizando a necessidade de uma abordagem robusta para lidar com a complexidade do setor.

4 METODOLOGIA

Este capítulo apresenta a metodologia adotada para aplicar modelos de aprendizado de máquina na previsão do preço mediano dos imóveis na Malásia, detalhando as etapas de preparação dos dados, treinamento e otimização dos modelos, e avaliação dos resultados. O código-fonte completo referente à implementação pode ser encontrado no Apêndice A, bem como no repositório disponível no GitHub.

4.1 Preparação e Pré-processamento dos Dados

Os dados foram carregados com a biblioteca `pandas` e submetidos a uma análise exploratória, a qual confirmou a integridade do conjunto. As variáveis categóricas (*Township*, *Area*, *State*, *Tenure* e *Type*) foram convertidas em variáveis dummy por meio de One-Hot Encoding, enquanto a variável numérica *Transactions* foi mantida em seu formato original. Para a Regressão Linear — modelo sensível à escala dos dados —, aplicou-se o `StandardScaler` aos atributos numéricos.

4.2 Treinamento e Otimização dos Modelos

Foram empregados quatro modelos de aprendizado de máquina para a previsão do preço mediano dos imóveis: **Regressão Linear, Árvore de Decisão, Random Forest e Gradient Boosting**. Cada um desses modelos possui características distintas, influenciando sua capacidade preditiva e adequação ao problema.

A **Regressão Linear** é um modelo estatístico clássico que assume uma relação linear entre as variáveis preditoras e a variável-alvo. Seu modelo matemático é expresso como:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon. \tag{4.1}$$

Onde y é o preço previsto do imóvel, x_i são as variáveis explicativas e ε representa o erro aleatório. Esse modelo se destaca por sua simplicidade e interpretabilidade dos coeficientes, além de sua baixa complexidade computacional. No entanto, apresenta limitações, como sensibilidade a outliers e pressuposições de independência e ausência de multicolinearidade entre as variáveis preditoras. Para mitigar esses problemas, as variáveis numéricas foram normalizadas utilizando o `StandardScaler`, garantindo que os coeficientes não fossem influenciados por diferenças de escala entre as variáveis.

A **Árvore de Decisão** é um modelo baseado na divisão recursiva dos dados em subconjuntos menores. A escolha das divisões é feita minimizando métricas como o erro quadrático médio (MSE) ou a impureza de Gini. Sua principal vantagem é a capacidade de modelar relações não lineares e lidar com variáveis categóricas sem necessidade de transformações complexas. Além disso, seu modelo é interpretável, permitindo análise direta das regras de decisão. Entretanto, Árvores de Decisão isoladas podem sofrer de sobreajuste (*overfitting*), especialmente quando muito profundas, e são sensíveis a pequenas variações nos dados. Para mitigar esse problema, utilizou-se o ajuste de hiperparâmetros via *Grid Search*, testando diferentes valores para a profundidade máxima da árvore (`max_depth`) e o número mínimo de amostras por divisão (`min_samples_split`).

O **Random Forest** é um modelo de *ensemble* que combina múltiplas Árvores de Decisão para obter previsões mais robustas. Ele utiliza o método de *bagging* (*bootstrap aggregation*), onde cada árvore é treinada em um subconjunto aleatório dos dados, reduzindo a variância do modelo. Essa abordagem melhora a generalização e reduz o sobreajuste em relação a uma única árvore. Além disso, é eficaz para lidar com dados de alta dimensionalidade. Entretanto, possui maior custo computacional e sua interpretabilidade é reduzida devido ao grande número de árvores no modelo. A otimização do *Random Forest* envolveu a busca pelos melhores valores de *número de árvores* (`n_estimators`) e *profundidade máxima* (`max_depth`) usando *Grid Search*.

O **Gradient Boosting** também é um modelo de *ensemble* baseado em Árvores de Decisão, mas ao contrário do *Random Forest*, as árvores são construídas sequencialmente. A cada iteração, o modelo tenta corrigir os erros das previsões anteriores, tornando-o altamente eficiente para tarefas preditivas. Essa abordagem proporciona melhor ajuste aos dados em comparação com métodos tradicionais, mas pode levar ao sobreajuste caso não seja corretamente ajustado. Seu alto custo computacional também pode ser um desafio em conjuntos de dados grandes. Para otimizar esse modelo, foram ajustados hiperparâmetros como a *taxa de aprendizado* (`learning_rate`), o *número de árvores* (`n_estimators`) e a *profundidade das árvores* (`max_depth`).

4.3 Otimização dos Modelos

Cada modelo foi treinado utilizando *pipelines* que integraram pré-processamento e modelagem. A otimização dos hiperparâmetros foi realizada por meio de *Grid Search*, enquanto a validação foi conduzida utilizando *K-Fold Cross-Validation* (10 *folds*), garantindo que os

resultados fossem generalizáveis.

A combinação dessas técnicas permitiu uma análise comparativa eficiente entre os modelos, identificando aquele que melhor se adequa à previsão do preço mediano dos imóveis na Malásia.

4.4 Avaliação dos Modelos

A performance dos modelos foi avaliada utilizando as seguintes métricas:

- **Mean Squared Error (MSE):** Média dos quadrados dos erros entre as previsões e os valores reais.
- **Root Mean Squared Error (RMSE):** Raiz do MSE, expressando os erros na mesma escala dos dados originais.
- **Coeficiente de Determinação (R^2):** Proporção da variância dos dados explicada pelo modelo.

Essas métricas permitiram uma comparação objetiva entre os modelos, identificando aquele que melhor se adapta ao problema. A implementação dos experimentos foi realizada em Python, utilizando as bibliotecas `scikit-learn` para modelagem, `pandas` e `numpy` para manipulação de dados, e `matplotlib` e `seaborn` para visualizações.

4.5 Fluxograma do Processo Metodológico

Para facilitar a compreensão, apresenta-se a seguir um fluxograma resumido do processo metodológico:

1. **Coleta e Carregamento dos Dados:** Obtenção e leitura dos dados com `pandas` contido no Apêndice A.1.
2. **Análise Exploratória:** Verificação da integridade e análise preliminar dos dados contido no Apêndice A.2.
3. **Pré-processamento:**
    - Conversão de variáveis categóricas via One-Hot Encoding contido no Apêndice A.3.
    - Escalonamento dos atributos numéricos (para modelos sensíveis à escala).
4. **Construção dos Pipelines:** Integração das etapas de pré-processamento e modelagem para cada algoritmo contido no Apêndice A.4.
5. **Otimização de Hiperparâmetros:** Ajuste dos parâmetros via Grid Search contido no

Apêndice A.5.

6. **Validação:** Aplicação do K-Fold Cross-Validation (10 folds) para avaliar a generalização dos modelos contido no Apêndice A.5.
7. **Avaliação de Desempenho:** Cálculo das métricas MSE, RMSE e R^2 contido no Apêndice A.5.

Essa metodologia estruturada possibilitou uma análise comparativa dos diferentes modelos, garantindo resultados robustos e a identificação da técnica mais adequada para a previsão do preço mediano dos imóveis na Malásia.

5 RESULTADOS

Neste capítulo, são apresentados os resultados obtidos a partir da aplicação dos modelos de aprendizado de máquina, bem como a discussão sobre o desempenho de cada abordagem.

5.1 Apresentação dos Resultados Quantitativos

A Tabela 1 resume as principais métricas de desempenho para os modelos testados, considerando o Mean Squared Error (MSE), o Root Mean Squared Error (RMSE) e o Coeficiente de Determinação (R^2).

Tabela 1 – Desempenho dos Modelos

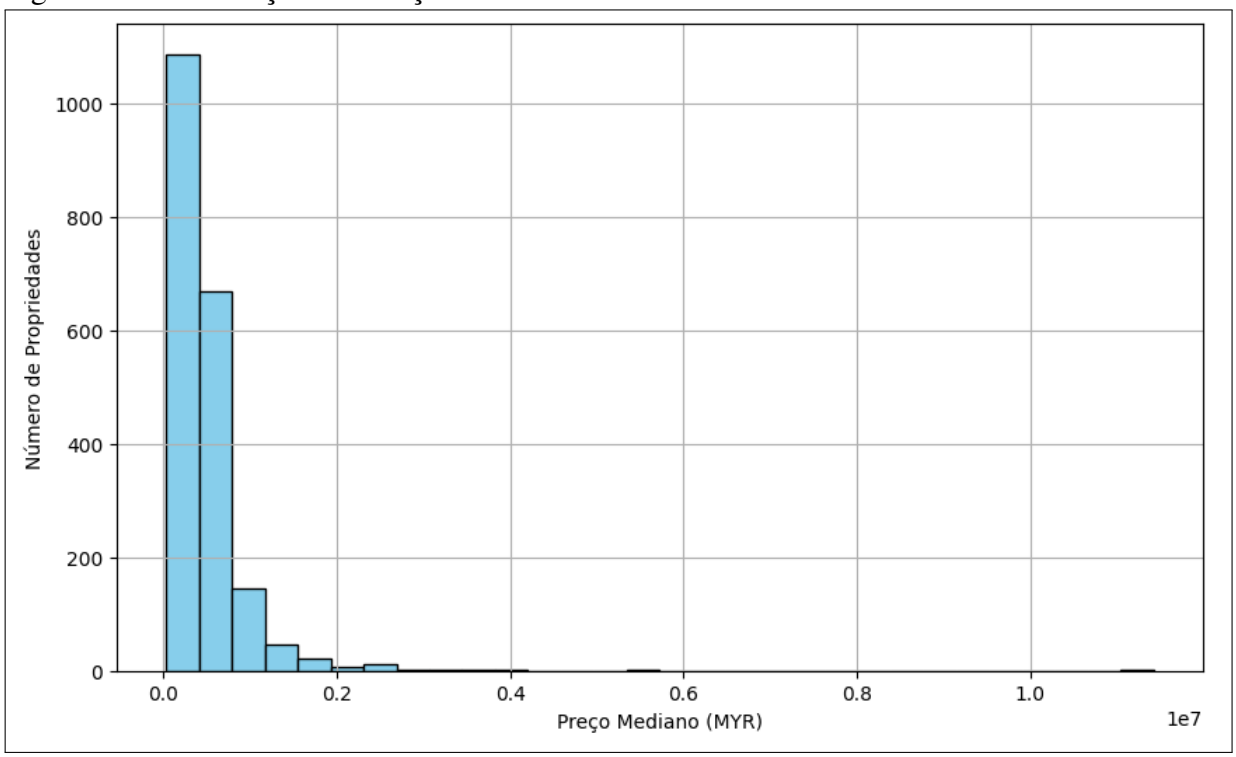
Modelo	**MSE**	**RMSE**	R^2
Regressão Linear	1.78e+11	398716.84	0.1933
Árvore de Decisão	1.93e+11	412551.56	0.1464
Random Forest	1.83e+11	400877.87	0.1961
Gradient Boosting	1.65e+11	379168.50	0.2810

Observa-se que o modelo de Gradient Boosting apresentou o melhor desempenho, com o menor RMSE e o maior valor de R^2, sugerindo uma maior capacidade de capturar a variabilidade dos dados. Os modelos baseados em árvores (Árvore de Decisão e Random Forest) apresentaram desempenho similar, enquanto a Regressão Linear ficou atrás em termos de acurácia.

5.2 Visualização dos Resultados

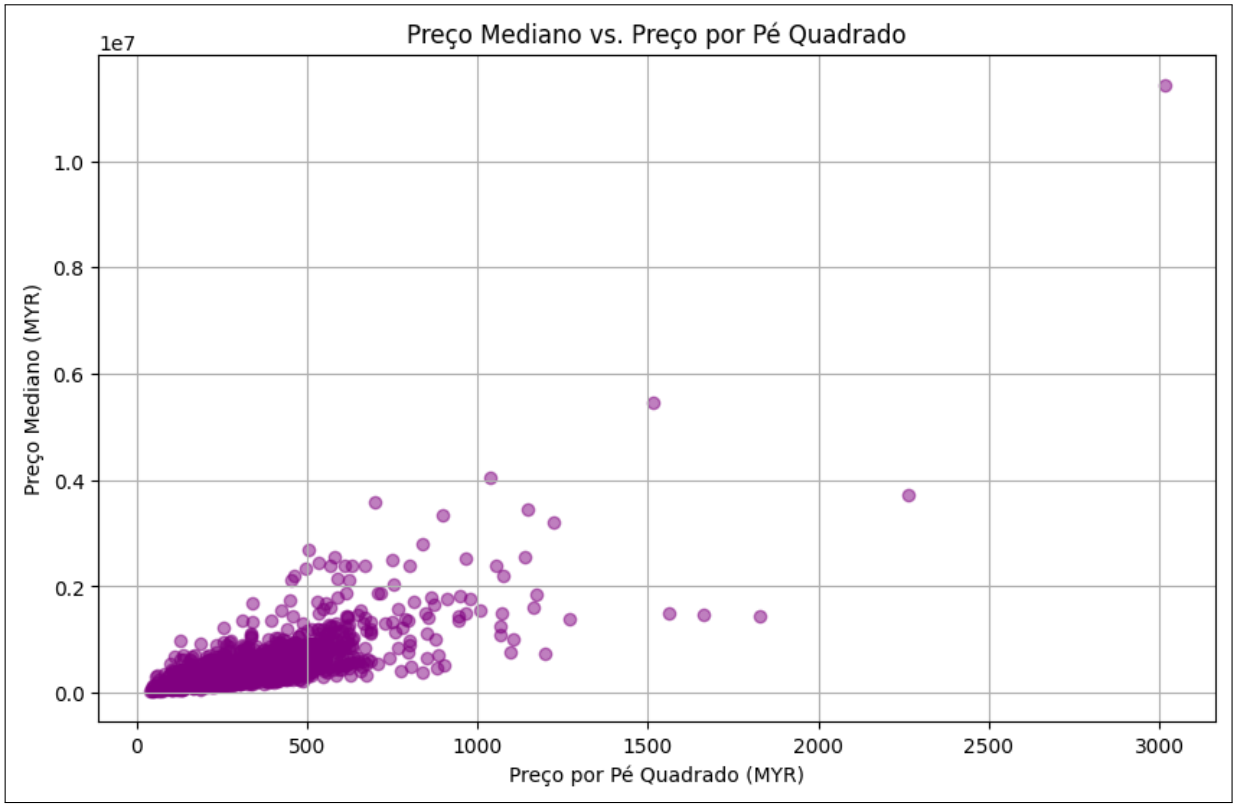
Para complementar a análise quantitativa, foram geradas diversas figuras. A Figura 1 ilustra a distribuição dos preços medianos dos imóveis, evidenciando a presença de dispersão nos dados. A Figura 2 apresenta a relação entre o preço mediano e o preço por pé quadrado, permitindo identificar tendências e possíveis correlações entre essas variáveis. Adicionalmente, a Figura 3 mostra a relação entre o preço mediano e o número de transações, o que pode refletir a atividade do mercado em diferentes regiões.

Figura 1 – Distribuição dos Preços Medianos dos Imóveis



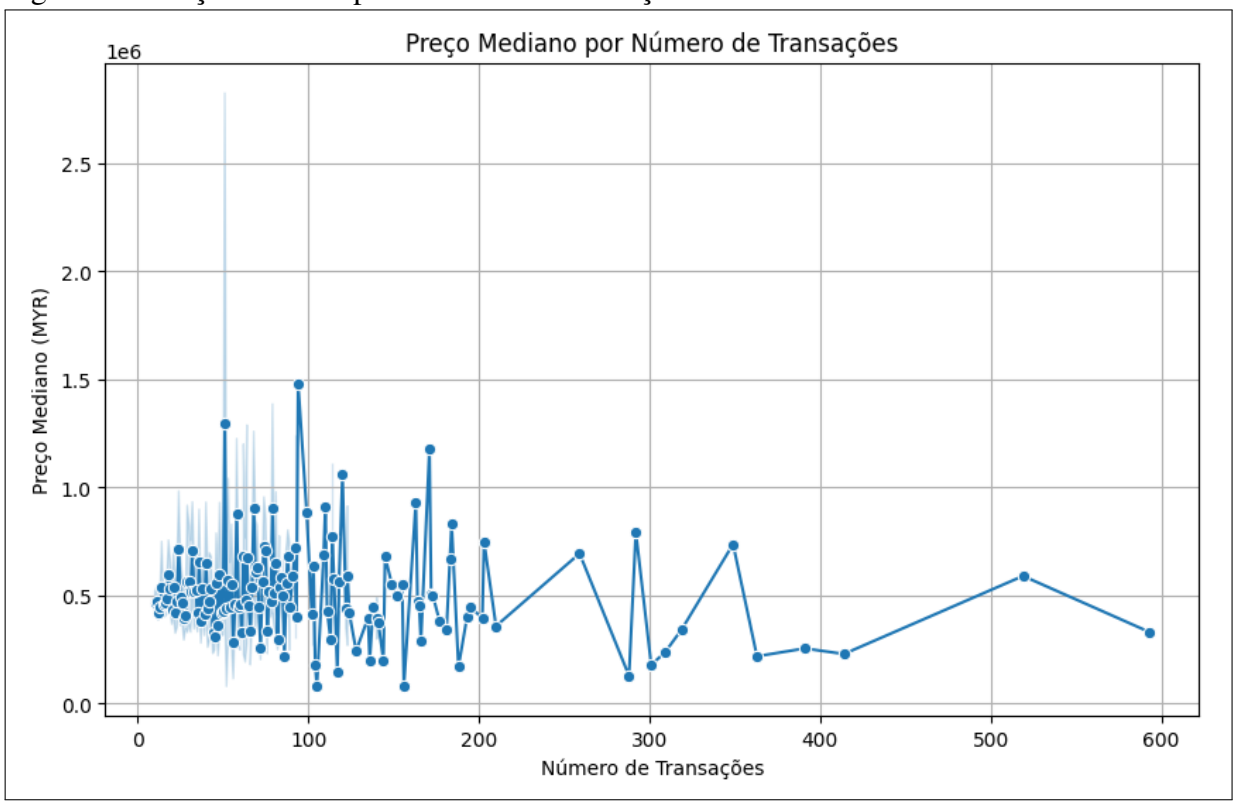
Fonte: Elaborado pelos autores (2025).

Figura 2 – Relação entre Preço Mediano e Preço por Pé Quadrado



Fonte: Elaborado pelos autores (2025).

Figura 3 – Preço Mediano por Número de Transações



Fonte: Elaborado pelos autores (2025).

5.3 Discussão dos Resultados

A análise dos resultados revela que o modelo de Gradient Boosting superou os demais, alcançando um R^2 de 0.2810, o que indica que aproximadamente 28,1% da variabilidade dos preços pode ser explicada pelo modelo. Esse desempenho superior pode ser atribuído à capacidade desse método de capturar relações não lineares entre as variáveis. Em contrapartida, a Regressão Linear apresentou limitações, evidenciando a necessidade de modelos mais sofisticados para lidar com a heterogeneidade dos dados do mercado imobiliário.

Além das relações observadas entre preço mediano, preço por pé quadrado e a distribuição dos preços, a Figura 3 ilustra a relação entre o preço mediano e o número de transações. Observa-se que regiões com maior atividade de transações tendem a apresentar variações nos preços, o que pode indicar que, embora um maior número de transações sugira uma demanda ativa, essa relação não é linear e pode ser influenciada por outros fatores, como a localização e o tipo de imóvel. Essa complexidade reforça a necessidade de modelos de ensemble, capazes de integrar múltiplas variáveis e capturar interações sutis entre elas.

Os resultados obtidos demonstram a importância do pré-processamento adequado, da otimização dos hiperparâmetros e da validação cruzada para a construção de modelos robustos.

A utilização de técnicas de ensemble, como Gradient Boosting e Random Forest, mostrou-se promissora para a previsão de preços em mercados complexos. Com base nesses achados, recomenda-se a exploração de abordagens mais avançadas e a inclusão de variáveis adicionais em estudos futuros, a fim de aprimorar a capacidade preditiva dos modelos.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho teve como objetivo desenvolver e comparar modelos de aprendizado de máquina para a previsão do preço mediano dos imóveis na Malásia. Foram aplicados algoritmos como Regressão Linear, Árvore de Decisão, Random Forest e Gradient Boosting, os quais foram avaliados por meio de métricas como MSE, RMSE e R^2. Os resultados demonstraram que o modelo de Gradient Boosting apresentou o melhor desempenho, evidenciando sua capacidade de capturar relações não lineares e a variabilidade dos dados do mercado imobiliário.

Embora os objetivos propostos tenham sido alcançados, o estudo aponta para diversas oportunidades de aprimoramento. Em futuras investigações, recomenda-se a incorporação de novas variáveis, como indicadores socioeconômicos, dados temporais e características específicas do imóvel, que podem contribuir para uma modelagem mais robusta. Além disso, a exploração de técnicas avançadas, como redes neurais profundas e métodos de aprendizado não supervisionado para segmentação de mercado, pode proporcionar insights adicionais e aumentar a precisão das previsões.

Em síntese, o presente trabalho contribui para o avanço das técnicas de previsão de preços imobiliários e serve como base para pesquisas futuras que visem melhorar os modelos de aprendizado de máquina, oferecendo subsídios para uma tomada de decisão mais informada no setor.

REFERÊNCIAS

ANTIPOV, E.; POKRYSHEVSKAYA, E. **Mass appraisal of residential apartments: An application of Random Forest for valuation and a CART-based approach for model diagnostics**. 2012. Disponível em: <https://core.ac.uk/outputs/6531251/>.

CHEN; HAO. Predictions of residential property price indices for china via machine learning approaches. 2020. Disponível em: <https://link.springer.com/article/10.1007/s11135-025-02080-3>.

FAN, e. a. **Research on the Influencing Factors Affecting Beijing House Prices Based on Linear Regression Model**. 2018. Disponível em: <https://link.springer.com/chapter/10.1007/978-981-99-6441-3_37>.

KUMAR, e. a. Housing price prediction using machine learning techniques. 2021. Disponível em: <https://ieeexplore.ieee.org/document/10629723>.

LI, e. a. A gradient boosting method for effective prediction of housing prices in complex real estate systems. 2020. Disponível em: <https://ieeexplore.ieee.org/document/9382480>.

PARK; KWON. **House Price Prediction Based on Machine Learning –Taking Beijing House Prices as an Example**. 2017. Disponível em: <https://eudl.eu/pdf/10.4108/eai.17-11-2023.2342672>.

ZHANG, e. a. A new appraisal model of second-hand housing prices in china's first-tier cities based on a two-tier stacking framework. 2019. Disponível em: <https://link.springer.com/article/10.1007/s10614-020-09973-5>.

APÊNDICE A – CÓDIGO FONTE UTILIZADO

A.1 Coleta e Carregamento dos dados

Código-fonte 1 – Importação das bibliotecas e dados

```
import pandas as pd
import numpy as np
from sklearn.model_selection import KFold, GridSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor,
   GradientBoostingRegressor
from sklearn.preprocessing import StandardScaler,
   OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns

file_path = "./malaysia_house_price_data_2025.csv"
df = pd.read_csv(file_path)
print(df.head())
```

A.2 Análise Exploratória

Código-fonte 2 – Verificação da integridade e análise preliminar dos dados.

```

missing_data = df.isnull().sum()

missing_data = missing_data[missing_data > 0]

```

```
6  if missing_data.empty:
7      print("Nao ha dados faltantes no dataset.")
8  else:
9      print("Dados faltantes encontrados:\n", missing_data)
10
11 # Distribuicao dos precos medianos
12 plt.figure(figsize=(10, 6))
13 plt.hist(df["Median_Price"].dropna(), bins=30, color='
       skyblue', edgecolor='black')
14 plt.title("Distribui  o de Pre os Medianos")
15 plt.xlabel("Pre o Mediano (MYR)")
16 plt.ylabel("N mero de Propriedades")
17 plt.grid(True)
18 plt.show()
19
20 # Relacao entre preco mediano e preco por pe quadrado
21 plt.figure(figsize=(10, 6))
22 plt.scatter(df["Median_PSF"], df["Median_Price"], alpha
       =0.5, color='purple')
23 plt.title("Pre o Mediano vs. Pre o por P  Quadrado")
24 plt.xlabel("Pre o por P  Quadrado (MYR)")
25 plt.ylabel("Pre o Mediano (MYR)")
26 plt.grid(True)
27 plt.show()
28
29 # Relacao entre transacoes e preco
30 plt.figure(figsize=(10, 6))
31 sns.lineplot(x="Transactions", y="Median_Price", data=df,
       marker="o")
32 plt.title("Pre o Mediano por N mero de Transa  es")
33 plt.xlabel("N mero de Transa  es")
34 plt.ylabel("Pre o Mediano (MYR)")
```

```
35 plt.grid(True)
36 plt.show()
```

A.3 Pré-processamento

Código-fonte 3 – Conversão de variáveis categóricas via One-Hot Encoding

```
1
2 y = df['Median_Price']
3
4 # Definindo as colunas categoricas
5 categorical_columns = ["Township", "Area", "State", "Tenure
     ", "Type"]
6 encoder = OneHotEncoder(sparse_output=False, drop="first")
7 encoded_cats = encoder.fit_transform(df[categorical_columns
     ])
8
9 # Converter para DataFrame
10 encoded_cats_df = pd.DataFrame(
11     encoded_cats,
12     columns=encoder.get_feature_names_out(
          categorical_columns)
13 )
14 encoded_cats_df.reset_index(drop=True, inplace=True)
15 df.reset_index(drop=True, inplace=True)
16
17 # Definindo as colunas num ricas
18 numeric_columns = ["Transactions"]
19
20 # Criar o DataFrame final (sem aplicar escalonamento ainda)
21 X = pd.concat([encoded_cats_df, df[numeric_columns].
     reset_index(drop=True)], axis=1)
```

```
22
23 print(X.head(),y.head())
```

A.4 Construção dos Pipelines

Código-fonte 4 – Integração das etapas de pré-processamento e Escalonamento

```
1
2 # Definindo os modelos com pre-processamento ideal
3 models = {
4     'Linear Regression': Pipeline([
5         ('scaler', StandardScaler()),  # Escalonamento
              apenas para regressao linear
6         ('model', LinearRegression())
7     ]),
8     'Decision Tree': Pipeline([
9         ('model', DecisionTreeRegressor())
10    ]),
11    'Random Forest': Pipeline([
12        ('model', RandomForestRegressor())
13    ]),
14    'Gradient Boosting': Pipeline([
15        ('model', GradientBoostingRegressor())
16    ])
17 }
```

A.5 Otimização, Validação e Avaliação de Desempenho

Código-fonte 5 – Integração das etapas de otimização de hiperparâmetros, validação e avaliação de desempenho

```
1
```

```
# Definindo os par metros para o GridSearch
param_grids = {
    'Linear Regression': {},
    'Decision Tree': {
        'model__max_depth': [3, 5, 10],
        'model__min_samples_split': [2, 5, 10]
    },
    'Random Forest': {
        'model__n_estimators': [50, 100, 200],
        'model__max_depth': [3, 5, 10]
    },
    'Gradient Boosting': {
        'model__n_estimators': [50, 100, 200],
        'model__learning_rate': [0.01, 0.1, 0.2],
        'model__max_depth': [3, 5, 10]
    }
}

# Definir o KFold
kf = KFold(n_splits=10, shuffle=True, random_state=42)

# Avaliar os modelos com GridSearchCV e KFold Cross-
   Validation
for model_name, pipeline in models.items():
    print(f"Model: {model_name}")

    # Inicializar o GridSearchCV
    grid_search = GridSearchCV(
        estimator=pipeline,
        param_grid=param_grids[model_name],
        cv=kf,
        scoring='neg_mean_squared_error'
```

```
    )

    # Variaveis para armazenar resultados
    mse_scores = []
    rmse_scores = []
    r2_scores = []

    # Loop KFold para treinamento e validacao
    for train_index, test_index in kf.split(X):
        X_train, X_test = X.iloc[train_index], X.iloc[
            test_index]
        y_train, y_test = y.iloc[train_index], y.iloc[
            test_index]

        # Treinar o modelo com GridSearchCV
        grid_search.fit(X_train, y_train)

        # Obter o melhor modelo
        best_model = grid_search.best_estimator_

        # Fazer previsoes
        y_pred = best_model.predict(X_test)

        # Calcular as metricas
        mse = mean_squared_error(y_test, y_pred)
        rmse = np.sqrt(mse)
        r2 = r2_score(y_test, y_pred)

        # Armazenar as metricas
        mse_scores.append(mse)
        rmse_scores.append(rmse)
        r2_scores.append(r2)
```

```
63
64      # Exibir os melhores parametros e as m tricas
65      # print(f"Best Parameters for {model_name}: {
            grid_search.best_params_}")
66      # print(f"Mean Mean Squared Error (MSE): {np.mean(
            mse_scores)}")
67      # print(f"Mean Root Mean Squared Error (RMSE): {np.mean
            (rmse_scores)}")
68      # print(f"Mean  R  : {np.mean(r2_scores)}\n")
```