doi:

Previsão de preço de imóveis no Município do Rio de Janeiro usando Redes Neurais

Frederico Thiers Dutra de Oliveira da Silva [**Universidade Federal do Estado do Rio de Janeiro** ***frederico.thiers@edu.unirio.br***]
Paulo Rodrigo Teixeira [**Universidade Federal do Estado do Rio de Janeiro** ***paulo.teixeira@edu.unirio.br***]
Thiago de Moura Parracho [**Universidade Federal do Estado do Rio de Janeiro** ***thiago.parracho@edu.unirio.br***]

Resumo

O presente estudo tem como objetivo desenvolver uma aplicação que permita a interação entre humano e máquina para prever o preço de imóveis no município do Rio de Janeiro. Para isso, os dados foram coletados em sites de classificados de imóveis, resultando em um conjunto com 12.029 entradas únicas, organizadas em 39 atributos. Diferentes modelos foram testados para a previsão dos preços, incluindo regressão linear simples, XGBoost, LightGBM e Rede Neural. A Rede Neural, implementada com o pacote PyTorch, apresenta o melhor desempenho, com um valor de R2 de 0,91. O modelo é otimizado utilizando o pacote Optuna, que ajusta os hiperparâmetros para alcançar a melhor performance. Os resultados indicam que a aplicação de Redes Neurais é eficaz para estimar preços no mercado imobiliário, com precisão considerável, especialmente para imóveis com valores entre R$ 1 milhão e R$ 3 milhões, apresentando um erro médio de 21,4% no conjunto de validação. Para facilitar o uso da ferramenta por corretores de imóveis, desenvolve-se uma interface gráfica baseada no pacote PySide. Os testes de usabilidade indicam que a aplicação é bem recebida pelos usuários, destacando a facilidade de uso e a precisão das previsões. Conclui-se que as Redes Neurais possuem um potencial significativo para a previsão de preços de imóveis no Rio de Janeiro, proporcionando uma ferramenta valiosa para profissionais do mercado imobiliário. No entanto, reconhece-se a necessidade de aprimoramento para lidar com imóveis de alto valor (acima de R$ 3 milhões) e da incorporação de variáveis externas para captar a dinâmica do mercado imobiliário.

Keywords: Inteligência Artificial. Redes Neurais.

1 Introdução

O mercado imobiliário possui um papel significativo na economia brasileira. Apenas no Rio de Janeiro, as transações de imóveis nos três primeiros trimestres de 2024 movimentaram R$28,8 bilhões, tendo um crescimento de 15,7% em relação ao ano anterior [Ademi, 2024]. Compreender as dinâmicas do mercado imobiliário pode contribuir para que imobiliárias e corretores otimizem suas estratégias de venda, facilitando transações e atendendo às demandas de consumidores. Um dos fatores determinantes na velocidade de venda de imóveis é o preço.

Determinar de forma errada o preço de um imóvel pode impactar sua venda. Preços muito abaixo do valor de mercado podem acelerar a venda, mas reduzem a receita, tanto para os corretores quanto para os proprietários. Por outro lado, imóveis com preços acima da média de mercado frequentemente enfrentam maior tempo de permanência à venda, o que pode gerar custos e impactar negativamente a atratividade do bem.

Com a evolução da Inteligência artificial (IA) e do aprendizado de máquina (ML, do inglês *Machine Learning*), surgiram novas abordagens para a previsão de preços de imóveis de forma mais eficiente e precisa. No entanto, a previsibilidade de preços depende de dados relevantes e disponíveis para alimentar tais modelos de ML.

Neste meio surgem as redes neurais. As redes neurais (NNs, do inglês *Neural Networks*) são conjuntos de funções aninhadas que processam dados de entrada para gerar saídas. Essas funções são definidas por parâmetros, compostos por pesos e vieses, que no PyTorch são armazenados em tensores. O treinamento de uma rede neural ocorre em duas etapas principais: 1. **Propagação Direta (*Forward Propagation*)**: Nesta etapa, a rede neural realiza sua melhor estimativa do resultado correto. Os dados de entrada são processados sequencialmente por todas as funções da rede, gerando uma predição. 2. **Propagação Reversa (*Backward Propagation*)**: Nesta etapa, a rede ajusta seus parâmetros de acordo com o erro da predição. O processo consiste em retroceder desde a saída, calculando as derivadas do erro em relação aos parâmetros de cada função (gradientes). Em seguida, os parâmetros são otimizados utilizando o método de descida do gradiente (*gradient descent*) [Pytorch, 2024].

As redes neurais artificiais (RNAs) tornaram-se uma tecnologia transformadora no estudo dos mercados financeiros. Os preços das ações são frequentemente considerados sequências temporais aleatórias e ruidosas. As RNAs, como sistemas de processamento não linear em larga escala e paralelos, utilizam dados de conexões intrínsecas para oferecer métodos e técnicas capazes de aproximar qualquer função contínua não linear. Notavelmente, esses sistemas operam sem a necessidade de pressuposições prévias sobre a natureza do processo subjacente que gera os dados [Pino *et al*., 2008].

É evidente que diversos fatores influenciam o preço dos imóveis, e a principal limitação das análises anteriores está em considerar apenas um número restrito de variáveis, além de, frequentemente, utilizarem métodos lineares. Nesse contexto, embora estudos anteriores tenham abordado parcialmente o problema, nenhum deles apresentou um modelo abrangente e preciso para estimar os preços de imóveis. O desenvolvimento de uma estimativa confiável, por meio de um modelo capaz de reduzir significativamente as incertezas,

pode oferecer suporte essencial à tomada de decisão no mercado imobiliário, promovendo maior eficiência e transparência. Assim, a realização de pesquisas científicas que visem à construção de um modelo adequado e eficaz para a predição de preços de imóveis é uma tarefa de grande relevância [Ahangar *et al*., 2010].

O presente estudo tem por objetivo desenvolver aplicação que permita interação entre humano e máquina para predição do preço de imóvel situado no município do Rio de Janeiro. Sua relevância paira na exata medida em que realizao o desenvolvimento de uma interface visual para o usuário-final, associado à modelo de predição de preços de imóveis, permitindo uma interação mais amigável, sendo este, inclusive, a contribuição para o campo de Sistemas de Informação.

O trabalho é organizado da seguinte forma: (1) introdução; (2) trabalhos relacionados; (3) background teórico; (4) método de pesquisa e construção da base de dados; (5) resultados obtidos; (6) interface humano-computador (IHC); (7) discussão; (8) conclusão.

2 Trabalhos relacionados

A seguir, são apresentados alguns estudos relacionados ao tema proposto, que exploraram a predição de preço de apartamentos. Chaphalkar e Sandbhor analisaram e resumiram diversas técnicas de inteligência artificial (IA) que podem ser utilizadas para previsão e avaliação de imóveis [Chaphalkar e Sandbhor, 2013]. Antunes desenvolveu uma aplicação móvel que utiliza uma rede neural artificial para avaliar apartamentos [Antunes, 2022]. A rede neural gerada pelo modelo, sendo do tipo perceptron multicamadas de regressão, resultou em R^2 de 81,14% e erro médio absoluto de R$ 63.861,71.

Pinter et al. propuseram um modelo de aprendizado de máquina para lidar com a complexidade da avaliação imobiliária. Fatores essenciais de mobilidade foram considerados como variáveis de entrada, como distância do trabalho dos moradores, distância da casa dos trabalhadores, giro do morador, entre outros [Pinter *et al*., 2020]. O modelo foi gerado utilizando o tipo perceptron multicamadas, treinado com o algoritmo evolucionário de otimização de enxame de partículas. O modelo apresentou resultados interessantes, indicando que a entropia dos trabalhadores e as distâncias de trabalho dos moradores influenciam diretamente o preço do imóvel.

Por fim, Kabaivanov e Markovska avaliaram como a IA pode ser usada para melhorar a compreensão das pessoas acerca das mudanças do mercado imobiliário [Kabaivanov e Markovska, 2021]. Os autores testaram um modelo de três estágios em suporte à avaliação imobiliária e previsão de mercado, que é capaz de levar em conta fatores econômicos globais, bem como características individuais que influenciam os preços dos imóveis. Cada estágio prevê o uso de diferentes métodos de IA e aprendizado de máquina para automatizar o processamento de dados de mercado e avaliar como os fatores qualitativos afetam a avaliação. Além disso, conduziram uma pesquisa sobre a precisão dos dados do modelo NAREIT e do índice BGREIT.

3 Método de pesquisa

Inicialmente, é fundamental destacar que o presente estudo enfrentou um desafio prévio à análise da questão central, qual seja, a predição do valor do imóvel. Não havia um conjunto de dados (*dataset*) adequado disponível para a realização da pesquisa, o que exigiu a criação de um novo banco de dados a partir de sites de classificados de imóveis.

Essa limitação demandou o desenvolvimento de *crawlers* e *spiders*, ferramentas que permitiram a navegação pelos classificados e a extração das informações necessárias para a construção do modelo.

3.1 Extração e tratamento dos atributos e entradas

Com o objetivo de construir a base de dados para o modelo, foram escolhidos os classificados do Grupo OLX, composto pelo OLX Imóveis, Viva Real e Zap Imóveis. A escolha se justificou pelo fato de esse grupo ser o maior no segmento de venda de imóveis e por sua colaboração com a Fundação Instituto de Pesquisas Econômicas (FIPE) no desenvolvimento do `fipezap`[1], o ”primeiro índice de preços de imóveis residenciais e comerciais com abrangência nacional”.

Dada sua relevância, os três sites foram utilizados como fonte para a criação do *dataset* empregado nesta pesquisa.

A extração dos dados foi realizada por meio de *crawler* e *spider*, desenvolvidos com o uso do pacote Selenium, em sua versão estável 4.27.1, na versão Chromium. Para evitar bloqueios de acesso ao site, foi utilizado o serviço oficial do Cloudflare em sua versão de teste gratuito.

A coleta ocorreu entre os dias 20 de novembro e 20 de dezembro, resultando na captura de 12.145 entradas únicas. A limitação geográfica foi restrita ao município do Rio de Janeiro, considerando apenas as consultas realizadas na página principal da pesquisa, sem detalhamento específico por bairros. Essa escolha metodológica baseia-se no fato de que buscas detalhadas poderiam aumentar o risco de erros no preenchimento dos dados, demandando um esforço adicional para a exclusão de registros inadequados.

3.2 Construção dos atributos do *dataset*

Foram incluídos inicialmente a totalidade dos imóveis capturados pelos *crawlers* e *spiders*, totalizando monde de 12.145 entradas. A partir deste resultado preliminar foi realizada exclusão de *outliers* que pudessem comprometer o treinamento do modelo e tratamento das informações a serem utilizadas. Após realização desta etapa foram excluídos 116 entradas que apresentavam a variável *target* (`PREÇO`) anormal. Foram incluídos para manipulação e possível treinamento do modelo 12.029 entradas únicas.

Para a captura e saturação das informações, foram inicialmente desenvolvidos 214 atributos únicos, os quais foram expandidos para 237 após a coleta de dados nos classificados imobiliários. Esse conjunto de dados não inclui três campos de controle: a *identificação* do anúncio, o *site* pesquisado e

[1] `https://www.fipe.org.br/pt-br/indices/fipezap/`

a data de consulta. Assim, ao final, foram criados 240 atributos únicos para cada imóvel.

Os atributos, com exceção daqueles que representavam o número de banheiros, quartos e a metragem do imóvel, bem como a zona de localização, o bairro e o endereço, foram tratados como variáveis `booleanas`, onde ”1”indicava verdadeiro e ”0”indicava falso. Essa abordagem simplificou o tratamento e a manipulação dos dados nas etapas subsequentes, permitindo maior eficiência no processamento.

A análise de correlações foi realizada utilizando um mapa de calor (*heatmap*), que revelou que algumas categorias apresentavam baixa ou nenhuma relevância para a precificação dos imóveis. Além disso, essas categorias não puderam ser agrupadas em outras existentes. Consequentemente, os seguintes atributos foram excluídos:

```
"SLAB", "PLATED_GAS", "FULL_FLOOR",
    "BLINDEX_BOX", "WOOD_FLOOR",
    "BEDROOM_WARDROBE", "GAS_SHOWER",
    "ALUMINUM_WINDOW", "INTEGRATED_ENVIRONMENTS",
    "PORCELAIN",
"PAY_PER_USE_SERVICES", "SERVICE_ROOM", "SANCA",
    "GRASS", "HIGH_CEILING_HEIGHT",
    "SIDE_ENTRANCE", "PLANNED_FURNITURE",
    "PANTRY", "SMALL_ROOM", "REVERSIBLE_ROOM",
"COPA", "COOKER", "WATER_TANK", "DIVIDERS",
    "FREEZER", "FULL_CABLING", "CARETAKER",
    "DECK", "HALF_FLOOR", "STAIR", "WELL",
    "ARTESIAN_WELL", "LAMINATED_FLOOR",
"GLASS_WALL", "CORNER_PROPERTY", "WALLS_GRIDS",
    "GEMINADA", "ADMINISTRATION", "VINYL_FLOOR",
    "LAND", "DRYWALL", "BURNT_CEMENT",
    "MEZZANINE", "PASTURE",
"RAISED_FLOOR", "EDICULE", "HEADQUARTERS",
    "BACKGROUND_HOUSE", "NEAR_SHOPPING_CENTER_2",
    "DRESS_ROOM2", "DINNER_ROOM", "CABLE_TV",
    "NUMBER_OF_FLOORS",
"INTERNET_ACCESS"
```

Os atributos remanescentes foram reorganizados em 23 categorias únicas, que traduzem de forma mais clara as características de cada imóvel. Além disso, 17 colunas originais foram mantidas, resultando em um total de **40 atributos únicos** para cada unidade habitacional.

Oportuno destacar que a criação de categorias não visa manipular os resultados obtidos, mas sim agrupar elementos que, devido à carência de uniformização de nomenclatura, acabam sendo representados como itens distintos, quando, na realidade, referem-se ao mesmo elemento. Por exemplo, o termo `PISCINA` apresentou variantes como `PISCINA AQUECIDA`, `PISCINA INFANTIL`, `PISCINA ADULTO`, entre outros correlatos.

Adicionalmente, para garantir que o modelo fosse adequadamente treinado, os atributos `ZONA` e `BAIRRO` foram convertidos em variáveis numéricas. O atributo `RUA` foi transformado em um índice ponderado, variando de 0 (baixa relevância) a 1 (alta relevância), para refletir sua importância dentro de um determinado bairro. Esses ajustes foram imprescindíveis para que a rede neural processasse exclusivamente dados numéricos, otimizando a acurácia dos resultados obtidos.

Após a realização dos agrupamentos, foi efetuada uma redução de dimensionalidade com o objetivo de eliminar possíveis multicolinearidades presentes no conjunto de dados. Nesse processo, foi identificada apenas uma ocorrência de multicolinearidade, a qual foi descartada no modelo final, resultando em um total de **39 atributos** únicos, incluindo a variável *target*.

É relevante destacar que esse procedimento também foi conduzido no início do processo de tratamento dos dados. Contudo, devido à natureza esparsa das variáveis e à recorrência de informações apresentadas de maneira distinta, as multicolinearidades não foram identificadas na primeira verificação, sendo evidenciadas apenas na segunda análise.

A fim de garantir melhores resultados na construção e avaliação do modelo, o conjunto de dados, composto por 12.029 entradas, foi estrategicamente dividido em duas partes distintas. A primeira parte, destinada ao treinamento e teste do modelo de predição, corresponde a 90% do total de dados disponíveis ($n = 10.826$). Essa parcela foi utilizada para ajustar os parâmetros do modelo e realizar a avaliação inicial de sua performance.

A segunda parte, composta pelos 10% restantes do conjunto de dados ($n = 1.203$ entradas únicas), foi mantida intacta como um *dataset* de validação final. Esse *dataset* intocado desempenha um papel crucial no processo de validação, permitindo que o desempenho do modelo seja testado com informações que o sistema não havia encontrado previamente.

Essa abordagem metodológica possibilita uma avaliação mais realista e precisa da capacidade preditiva do modelo, minimizando o risco de superajuste (*overfitting*) aos dados de treinamento e garantindo maior generalização dos resultados.

A título de exemplo, a estrutura de atributos de um imóvel é apresentada abaixo:

```
"FURNISHED": 1(bool),
"GOURMET_SPACE": 1(bool),
"KITCHEN": 1(bool),
"DRESS_ROOM": 0(bool),
"BALCONY": 1(bool),
"SPORTS_COURT": 0(bool),
"FOOD_COURT": 1(bool),
"RECREATION_AREA": 1(bool),
"REST_AREAS": 0(bool),
"POOL": 1(bool),
"SAFETY": 1(bool),
"VIEW": 0(bool),
"ACCESSIBILITY": 0(bool),
"WORK_AREAS": 0(bool),
"GARAGE": 0(bool),
"SUSTAINABILITY": 0(bool),
"OUTDOOR_AREAS": 0(bool),
"PET_FRIENDLY": 1(bool),
"MID_END_APARTMENT_AMENITIES": 0(bool),
"HIGH_END_APARTMENT_AMENITIES": 0(bool),
"HIGH_END_CONDO_AMENITIES": 0(bool),
"ZONE_CODE": 2(int),
"NEIGHBORHOOD_CODE": 94(int),
-------------------------------------------------
"PRICE": 400000(int),
-------------------------------------------------
"USABLEAREAS": 57(int),
"TOTALAREAS": 57(int),
"PARKINGSPACES": 1(bool),
```

```
"SUITES": 1(int),
"BEDROOMS": 2(int),
"ELEVATOR": 0(bool),
"SERVICE_AREA": 0(bool),
"LAUNDRY": 0(bool),
"LUNCH_ROOM": 0(bool),
"ESSENTIAL_PUBLIC_SERVICES": 0(bool),
"NEAR_HOSPITAL": 0(bool),
"NEAR_SHOPPING_CENTER": 0(bool),
"COVERAGE": 0(bool),
"YEARLY_EXPENSES": 0(bool),
"STREET_RELEVANCE": 0.06072318007376945 (float)
```

A estrutura convertida para o formato `.csv` foi utilizada no modelo de treinamento, sendo eliminada a variável *target*, resultando no treinamento do modelo com as 38 variáveis residuais.

3.3 Construção do modelo

Inicialmente, foi aplicado um modelo de regressão simples, o qual apresentou uma métrica de R^2 equivalente a 0.57. Após esta análise inicial, foi implementada uma regressão utilizando o *gradient-boosting* LightGBM[2], que obteve uma métrica de R^2 igual a 0.72.

Com o objetivo de melhorar ainda mais os resultados, foi desenvolvida uma regressão por meio de rede neural, a qual alcançou um valor de R^2 de 0.91, tornando-se o modelo selecionado para realizar a regressão e a predição final do preço dos imóveis (Table 1).

Método	R^2
Regressão linar simples	0.57
XGBoost	0.70
LightGBM	0.72
Rede Neural	0.91

Tabela 1. Comparação entre os métodos e valor de R^2.

O modelo de rede neural foi construído com o auxílio do pacote PyTorch[3], em sua versão 2.5.1. Os hiperparâmetros foram otimizados utilizando o pacote Optuna[4], em sua versão 3.4.0.

```
def objective(trial):
 hidden_units = trial.suggest_int('hidden_units',
     32, 512, step=32)
    num_layers = trial.suggest_int('num_layers',
        2, 5)
 dropout_rate =
     trial.suggest_float('dropout_rate', 0.1, 0.5)
 learning_rate =
     trial.suggest_float('learning_rate', 1e-5,
     1e-2)
 best_mae = train_model(model, criterion,
     optimizer, train_loader, test_loader,
     num_epochs=50)
```

[2]https://lightgbm.readthedocs.io
[3]https://pytorch.org/
[4]https://optuna.org/

```
def train_model(model, criterion, optimizer,
    train_loader, val_loader, num_epochs=50):
 train_loss = running_loss / len(train_loader)
 train_mae = running_mae / len(train_loader)
 val_loss = val_loss / len(val_loader)
 val_mae = val_mae / len(val_loader)
```

Cada ensaio (*trial*) foi configurado para executar a operação um total de 50 vezes. Assim, considerando o total de 10 ensaios realizados, os hiperparâmetros foram definidos com base em um conjunto de 500 execuções. Após o término das execuções, foram encontrados os seguintes valores:

- `hidden units` = 480
- `num layers` = 2
- `dropout rate` = 0.25193649347473823
- `learning rate` = 0.002918542020635498

Os parâmetros foram utilizados para parametrizar o modelo inicial, que executou o treinamento por 100 épocas iniciais. Após essa execução, o modelo foi submetido a um treinamento adicional (*fine-tuning*) em duas etapas distintas, cada uma com 5.000 épocas, utilizando pequenas variações nos parâmetros de entrada. Essa abordagem permitiu um melhor ajuste do modelo aos dados disponíveis.

No entanto, o modelo apresentou interrupção automática na época 4.287 durante o primeiro *fine-tuning*, devido ao surgimento de comportamento de *overfitting* indesejado. O modelo salvo corresponde ao estado anterior ao início do *overfitting*, sendo considerado mais adequado para a tarefa de previsão dos preços. A segunda etapa de *fine-tuning* não foi realizada em virtude da interrupção do *script*.

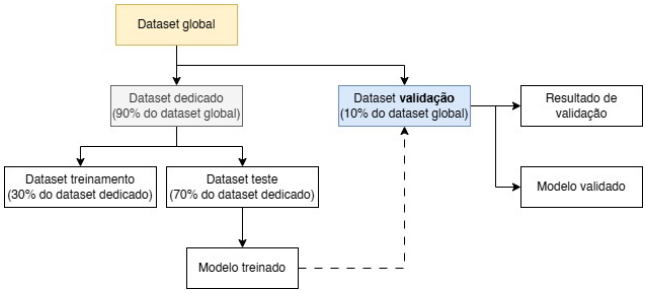

Figura 1. Treinamento do modelo de IA - Rede Neural.

A média de erro obtida no *dataset* de validação — que, conforme mencionado anteriormente, corresponde a 10% do total de entradas do *dataset* original — foi de 21,4%. Embora a margem de erro seja relativamente ampla, ela é considerada razoável, dada a complexidade de prever os preços de imóveis. Esse desafio é agravado pela presença de valores que não refletem, de maneira precisa, as características reais dos bens avaliados.

4 Resultados

Os resultados foram obtidos a partir de duas métricas. A primeira dela consistiu na realização de predição do valor de imóvel das entradas do *dataset* de validação. Os resultados obtidos (Figure 2) indicam que o modelo conseguiu prever o valor do imóvel de forma relativamente adequada.

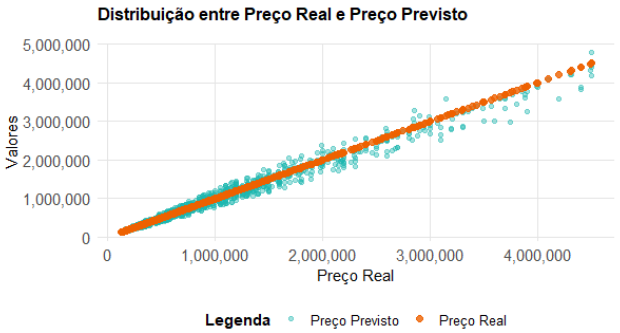

Figura 2. Preço efetivo *x* Preço projetado.

O modelo apresentou dois comportamentos que se destacam. O primeiro deles é que, em imóveis de até R$ 1.000.000,00 (um milhão de reais), a diferença entre o valor projetado e o valor efetivo do imóvel foi de 8.82% (Figure 3).

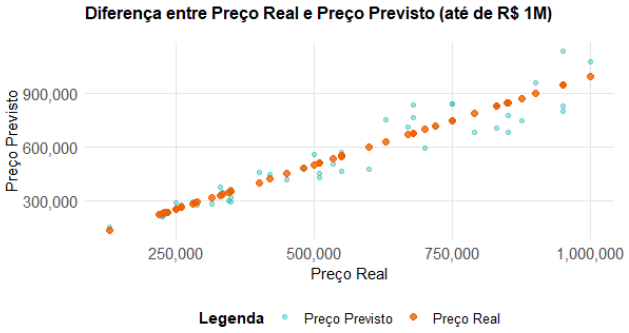

Figura 3. Preço efetivo *x* Preço projetado até R$ 1M.

Entre os valores de R$ 1.000.000,00 (um milhão de reais) e R$ 2.500.000,00 (dois milhões e quinhentos mil reais), o modelo apresentou flutuação ligeiramente menor, com variação de 8.37% (Figure 4). A média simples deste grupo foi de -2.85%, indicando que o modelo teve tendência de apresentar preços ligeiramente menores do que o efetivo do mercado, transparecendo o comportamento da predição.

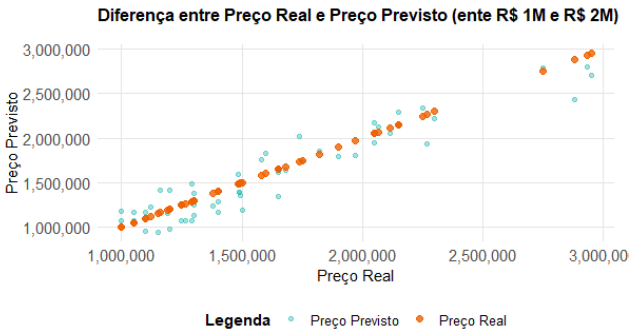

Figura 4. Preço efetivo *x* Preço projetado entre R$ 1M e 2M.

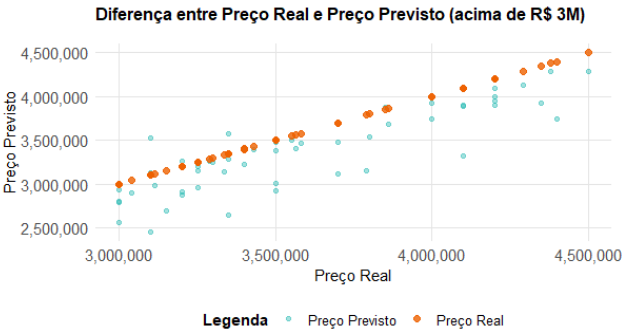

Figura 5. Preço efetivo *x* Preço projetado acima de R$ 3M.

A partir da marca de R$ 3.000.000,00 (três milhões de reais), o modelo passou a entregar valores inferiores ao valor efetivo do imóvel, de modo que a flutuação superior foi praticamente inexistente (Figure 5), com média de erro de 6.83%, e média simples de -5.96%, indicando que o modelo apresenta maior erro na predição de imóveis nesta faixa de preço, com subestimação do preço do imóvel.

O comportamento do modelo é esperado, uma vez que grande parte dos dados do *dataset* de validação possuiam preço entre R$ 1M e R$ 3M, sendo o grupo que o modelo precisava considerar maior número de variáveis para alcançar o resultado esperado. Em que pese este comportamento, é possível observar que os valores alcançados pelo modelo (Figures 4 and 5) apresentavam ajustamento próximo aos valores efetivos dos imóveis, indicando que o modelo conseguia, efetivamente, entregar valores próximos aos determinado pelo mercado.

5 Interface e interação humano-computador

Com o objetivo de desenvolver um interfaciamento gráfico para que o usuário-final — corretor de imóvel — pudesse realizar a avaliação preliminar do imóvel a ser comercializado, foi desenvolvida aplicação visual baseada no pacote PySide, em sua versão 6 — este baseado na biblioteca QtPy6 —, também desenvolvida diretamente em pýthon fig. 6.

Figura 6. Tela inicial da aplicação.

Uma vez apresentado ao sistema, o usuário pode iniciar uma aplicação nova atravez do menu superior ou do atalho

de teclado `CTRL + N`, ao que o levará para a tela de preenchimento dos dados (fig. 7). A tela é dividida em sete abas, separadas de forma temática, que permitem com que o usuário preencha de forma lógica as informações necessárias para o cálculo do valor do imóvel.

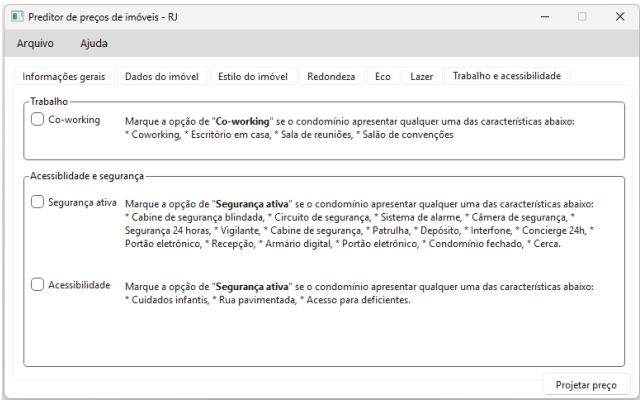

Figura 7. Tela de preenchimento dos dados.

Ao longo do preenchimento o usuário é possibilitado de salvar o projeto, através do atalho `CTRL + S`, ou projetar preço, que consiste na efetiva previsão do valor do imóvel. O projeto salvo poderá ser acessado posteriormente pelo acervo de projetos, permitindo com que o usuário ajuste projetos já existitentes sem a necessidade de realizar novo preenchimento de informações já constantes no imóvel.

5.1 Validação do modelo e testes de uso

Com o objetivo de validar o modelo e testar a usabilidade da aplicação, foram convidados cinco corretores de imóvel ou profissionais que necessitavam mensurar o preço médio de imóveis. Destes, quatro eram advogados e um corretor de imóvel, apresentando faixa etária entre 31 a 54 anos.

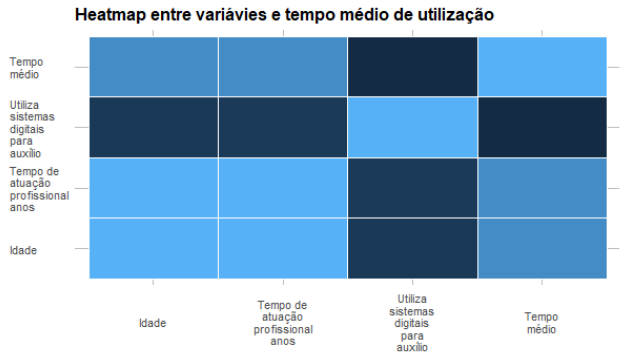

Figura 8. *Heatmap* entre variávies e tempo médio de utilização.

O tempo total de preenchimento das informações não se mostrou uniforme, tendo como tempo médio total de 14min7seg. Embora o tempo seja relativamente alto, este se justifica pela necessidade de haver preenchimento de informações detalhadas do imóvel, bem como necessidade de compreender as categorias que são apresentadas aos usuários.

Ao longo dos testes de uso foi possível perceber que as duas primeiras interações apresentaram maior demora, com

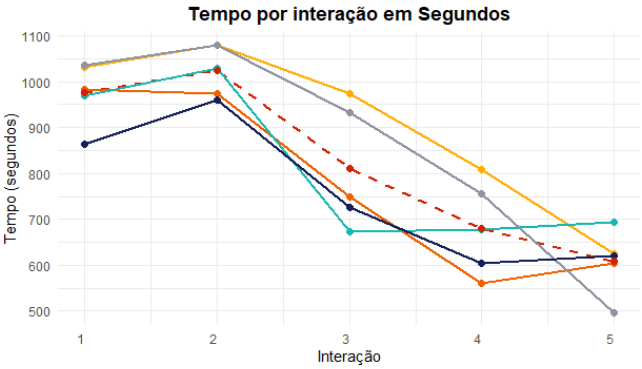

Figura 9. Tempo de preenchimento por interação.

tempo médio de 16min28seg para a **primeira** e 17min08seg para a **segunda**. com média absoluta de 17min02seg, e as três demais apresentaram média absoluta de 12min15seg, indicando que os usuários estavam mais habituados ao preenchimento dos dados tendem a apresentar melhores tempos na criação de novos projetos.

Observa-se que imóveis com maior valor apresentavam maior tempo de preenchimento, quando comparado com imóveis de menor valor (fig. 10), em iguais observações, e corroborando os achados descritos na seção de resultados, quanto maior o número de atributos preenchido, maior o tempo gasto pelo usuário na aplicação (fig. 11).

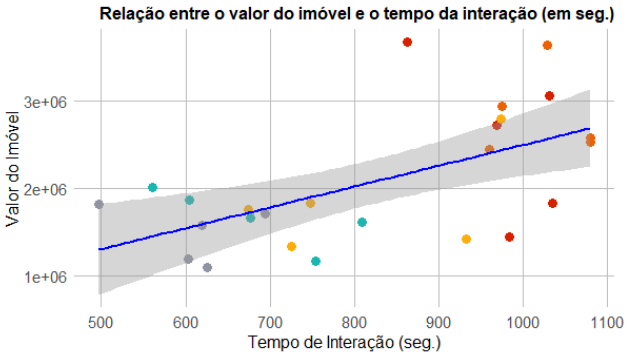

Figura 10. Relação entre o valor do imóvel e o tempo da interação (seg.).

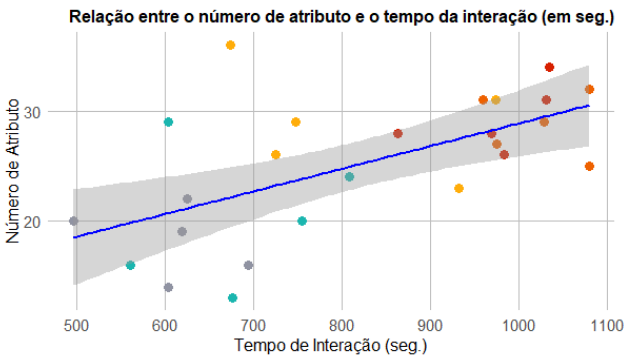

Figura 11. Relação entre o número de atributo e o tempo da interação (em seg.).

Dentre as facilidades relatou-se que a mais relevante foi a possibilidade recobrar projetos em andamento ou já concluídos, para posterior preenchimento. Dentre as dificuldades

relatadas pelos respondentes a dificuldade de precisar se imóvel apresenta elementos de imóvel de médio ou alto padrão foi a mais recorrente. Ainda no eixo das críticas, a necessidade de baixar aplicação *desktop* é vista de forma negativa, sendo referenciado que a aplicação seria melhor recebida se fosse baseada em *web* ou portátil (*mobile-first*).

Uma deficiência anotada majoritamente pelos usuários foi a carência de relatório PDF do valor obtido, bem como histórico de cada imóvel para acompanhar variação do preço em função dos atributos apresentados na aplicação. Outra deficiência apontada pelos participantes é a necessidade de incluir imagens do imóvel, considerando que, para estes, as imagens que compõe o imóvel podem apresentar elementos relevantes da qualidade do imóvel, afetando diretamente o preço do bem.

6 Discussão

Inicialmente, observa-se que a utilização de redes neurais apresentou vantagens significativas em relação à regressão simples, alcançando métricas superiores, como é mostrado em Chaphalkar e Sandbhor [2013] e Antunes [2022]. Entretanto, essa melhoria deve ser analisada com cautela, dado que o modelo de rede neural demonstrou maior suscetibilidade ao *overfitting* em comparação à regressão por LightGBM [Salman e Liu, 2019]. Esse comportamento pode ser atribuído tanto à maior quantidade de parâmetros utilizados (39 nas redes neurais contra 32 no LightGBM) quanto ao número de épocas de treinamento.

Para mitigar os efeitos do *overfitting*, foi aplicada a técnica de interrupção automática do treinamento (*early stopping*) [Bejani e Ghatee, 2021; Srivastava *et al*., 2014]. Essa abordagem permite treinar o modelo de maneira mais eficiente, reduzindo significativamente os riscos de *overfitting*.

O valor de R^2 igual a 0,92 demonstrou desempenho superior em comparação aos outros modelos treinados. Comparando com o trabalho relacionado de Antunes [2022], que teve o valor de R^2 de 81,14%, o nosso obteve melhor desempenho. No entanto, esse resultado deve ser interpretado dentro dos limites do *dataset* utilizado, não sendo possível garantir que o modelo apresente o mesmo desempenho em outros conjuntos de dados ou em contextos com maior heterogeneidade. No escopo do presente estudo, a rede neural desenvolvida mostrou-se mais precisa na previsão dos preços dos imóveis.

Quando aplicado ao *dataset* de validação, o modelo apresentou um comportamento consistente, oferecendo previsões relativamente confiáveis para os preços dos imóveis, conforme detalhado anteriormente.

Ao ser utilizado pelos usuários, o modelo foi recepcionado de forma positiva. Quando realizada a previsão de preços (Figure 12), o modelo apresentou erro médio de 18%, com tendência de apresentar preços ligeiramente abaixo quando alcançado o platô dos R$ 2M.

O comportamento do modelo se adequa àqueles observados no *dataset* de validação, de sorte que os testes realizados pelos usuários acabam por confirmar os achados inicialmente observados, isto é, em preços mais baixos (de até R$ 1.5M, o modelo apresenta preços ligeiramente mais altos, mas ainda assim próximos aos indicados como "corretos" pelos profissionais. Entre R$ 2M e R$ 3M passa-se observar distanciamento mais negativo entre os preços. Após o marco dos R$ 3M a diferença é mais evidente.

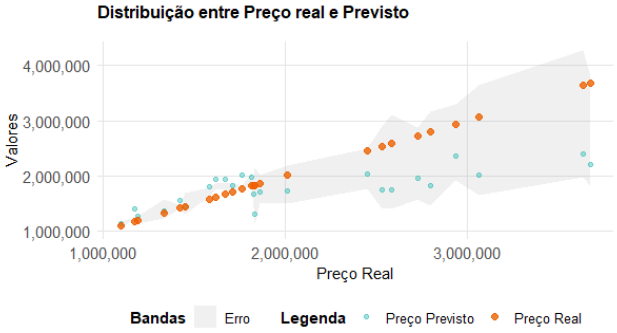

Figura 12. Distribuição de preços projetados os usuários e valor apontado como correto.

Embora a fig. 12 apresente diferença maior a partir do marco dos R$ 3M em relação àquele apresentado na fig. 2, este comportamento se justifica ao passo de que o preenchimento dos usuários apresentava dados mais específicos, e com ruídos não vistos nos dados de validação. A presença de preenchimentos errados — ruídos — e combinações específicas dos elementos do imóvel ocasionaram a presença de maior margem de erro.

Embora seja percebida esta variação, o preço predito se encontrava dentro da banda de *erro*, de modo que os valores que o modelo entregava — margem superior, valor predito, e margem inferior — compreendia o valor indicado como correto pelo usuário.

7 Conclusão

A pesquisa demonstrou o potencial significativo das Redes Neurais na previsão de preços de imóveis no Rio de Janeiro, alcançando um valor de R^2 de 0,91, superior aos modelos de regressão linear simples, XGBoost e LightGBM testados. O modelo foi treinado com um conjunto robusto de dados, composto por 12.029 entradas extraídas de sites de classificados imobiliários, que foram organizadas em 39 atributos para otimizar a predição. Os resultados indicam que a aplicação de Redes Neurais pode ser uma ferramenta eficaz para a estimativa de preços no mercado imobiliário, oferecendo uma precisão considerável mesmo em um ambiente complexo e volátil.

Além disso, a pesquisa contribui de forma relevante para o avanço do uso de Inteligência Artificial no setor imobiliário, mostrando que é possível alcançar um alto grau de precisão na previsão de preços com a utilização de dados estruturados. A utilização de modelos preditivos baseados em Redes Neurais pode não apenas facilitar as decisões de compra e venda, mas também fornecer insights valiosos para profissionais do mercado, como corretores e investidores, ao permitir uma melhor avaliação de imóveis em diversas faixas de preço.

No entanto, apesar dos resultados promissores, a pesquisa identificou algumas limitações, como a maior dificuldade na

previsão de imóveis com valores superiores a R$ 3 milhões, uma faixa menos representada no conjunto de dados de treinamento. A margem de erro de 21,4% no conjunto de validação, embora aceitável considerando a complexidade do mercado, aponta para a necessidade de aprimoramentos contínuos no modelo. Tais melhorias podem incluir a ampliação do conjunto de dados, especialmente para imóveis de alto valor, a incorporação de variáveis externas que capturam melhor a dinâmica do mercado imobiliário e o desenvolvimento de interfaces mais intuitivas, que aumentem a acessibilidade e aplicabilidade da ferramenta. Esses aprimoramentos são essenciais para consolidar a eficácia do modelo e expandir suas capacidades de previsão em um mercado em constante mudança.

Declarações

Contribuições dos autores

Todos os autores contribuiram igualmente para a elaboração do trabalho. Todos os autores leram e aprovaram o manuscrito final.

Conflitos de interesse

Os autores declaram não ter conflitos de interesse.

Disponibilidade de dados e materiais

Os conjuntos de dados (*datasets*) gerados e/ou analisados durante o estudo atual serão feitos mediante solicitação. O repositório do sistema encontra-se em domínio público localizado em: https://github.com/FTDutra/UNIRIO_IMO_V4. A URL será desativada no final do mês de fevereiro/2025, ou a critério dos autores.

Referências

Ademi (2024). Ademi parameters. `https://ademi.org.br/` [Accessed: (12-dez-2024)].

Ahangar, R. G., Yahyazadehfar, M., e Pournaghshband, H. (2010). The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in tehran stock exchange. *arXiv [cs.NE]*.

Antunes, A. K. (2022). *Aplicação de Rede Perceptron Multicamada para Predição do Valor de Venda de Apartamentos*. Monografia de especialização, Programa de Pós-Graduação em Inteligência Artificial Aplicada, Universidade Federal do Paraná, Curitiba, Brasil.

Bejani, M. M. e Ghatee, M. (2021). A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review*, 54(8):6391–6438. DOI: 10.1007/s10462-021-09975-1.

Chaphalkar, N. B. e Sandbhor, S. (2013). Use of artificial intelligence in real property valuation. *International Journal of Engineering and Technology*, 5(3).

Kabaivanov, S. e Markovska, V. (2021). Artificial intelligence in real estate market analysis. *AIP Conference Proceedings*, 2333(1):030001. DOI: 10.1063/5.0041806.

Pino, R., Parreno, J., Gomez, A., e Priore, P. (2008). Forecasting next-day price of electricity in the spanish energy market using artificial neural networks. *Eng. Appl. Artif. Intell.*, 21(1):53–62.

Pinter, G., Mosavi, A., e Felde, I. (2020). Artificial intelligence for modeling real estate price using call detail records and hybrid machine learning approach. *Entropy*, 22(12). DOI: 10.3390/e22121421.

Pytorch (2024). A gentle introduction to torch.autograd. `https://pytorch.org/tutorials/beginner/blitz/autograd_tutorial.html` [Accessed: (12-dez-2024)].

Salman, S. e Liu, X. (2019). Overfitting mechanism and avoidance in deep neural networks.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., e Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15.