**Projeto: Programa de Venda de Imóveis**

**Equipe:**

* Andre Gonçalves Borges - 172317511
* Felipe Doro Warchelski - 172316514
* Felipe Mariano – 172317392
* Igor Ricardo Oliveira Lourenço 172317394

**Unidades Curriculares:**

* Estruturas Matemáticas
* Inteligência Artificial

**Objetivo Geral:**

Desenvolver um modelo de IA para prever o valor de imóveis utilizando técnicas de regressão linear, com um foco aprofundado nas estruturas matemáticas que fundamentam o funcionamento do modelo, como álgebra linear e estatística.

**Objetivo Específico 1 (IA):**

Aplicar algoritmos de IA, como regressão linear ou regressão de árvore de decisão, para realizar previsões precisas de preços de imóveis com base em variáveis como localização, metragem e número de quartos. Enfatizar o ajuste do modelo e otimização de parâmetros para melhorar a precisão do sistema.

**Objetivo Específico 2 (Estruturas Matemáticas - problema mais complexo):**

Explorar e aplicar conceitos de álgebra linear e estatística, como decomposição de matrizes, análise de variância, e métodos de otimização, para garantir uma base matemática sólida na implementação do modelo de IA. Demonstrar como esses conceitos podem melhorar a precisão, eficiência e interpretação do modelo, tornando a análise mais robusta.

**Justificativa**

O tema foi escolhido devido à crescente demanda no mercado imobiliário e à necessidade de prever aluguéis com valores mais justos. Uma ferramenta que ajude proprietários e inquilinos a entender variações de preços com base nas características dos imóveis é muito útil para ambos. O projeto também permite pôr em prática os conceitos das disciplinas de IA e Estruturas Matemáticas. Além disso, modelos preditivos no mercado imobiliário ajudam nas decisões de investimento, compra, venda e aluguel.

**Metodologia**

- Pesquisa Bibliográfica

Será realizada uma revisão de literatura em fontes como livros, artigos e publicações sobre IA, regressão linear e o mercado imobiliário. Fontes confiáveis incluem Google Scholar e bases de dados acadêmicos como Scielo.

- Fontes de Dados

Bases de dados de imóveis, como aquelas fornecidas por sites de anúncios (ex.: OLX, ZAP Imóveis, etc), podem ser utilizadas para coleta de dados. Alternativamente, datasets disponíveis em plataformas como Kaggle podem ser aproveitados.

- Brainstorm

Discussões em grupo sobre possíveis variáveis a serem incluídas no modelo, como proximidade a transporte público, segurança do bairro, entre outras.

- Simulações e Modelagem Computacional

Simulações e modelagem serão feitas usando ferramentas de IA, para implementar e testar a regressão linear.

- Principais Ferramentas

IDE: Vscode para o desenvolvimento do código.

Linguagem: Python, com as bibliotecas scikit-learn (para machine learning) e Pandas (para manipulação de dados).

Bibliotecas de IA: scikit-learn para construção e avaliação do modelo de regressão linear.

Ferramentas Matemáticas: NumPy para cálculos matriciais e operações matemáticas.

**DESENVOLVIMENTO**

Foram utilizadas as seguintes ferramentas para prosseguir com o desenvolvimento do projeto:

Visual Studio Code para desenvolvimento, Python, Bibliotecas de IA: scikit-learn para Machine Learning, Pandas para tratamento de dados, Biblioteca para cálculos matemáticos: NumPy e Bibliotecas para visualização gráfica: Matplotlib e Seaborn.  
  
Procedimentos para a realização do projeto:

Levantamento de dados: Coleta de informações sobre imóveis a partir de um dataset disponível na plataforma Kaggle, o dataset tem sua origem em Web Scraping do site “https://www.dfimoveis.com.br/”.

Preparação dos dados: Limpeza e transformação dos dados em um formato utilizável, alteração dos nomes dos bairros conforme tabela numérica para facilitar codificação.

Divisão do conjunto de dados: Separação dos dados em conjuntos de treino e teste, conforme ficará mais claro no código em Python.

Criação do modelo: Implementação do modelo de regressão linear utilizando scikit-learn.

Treinamento do modelo: Ajuste do modelo aos dados de treino.

Avaliação do modelo: Teste do modelo com dados não vistos e cálculo de métricas de desempenho.

**Código Python**

**import pandas as pd**

**import numpy as np**

**from sklearn.model\_selection import train\_test\_split, GridSearchCV, cross\_val\_score**

**from sklearn.linear\_model import LinearRegression**

**from sklearn.ensemble import RandomForestRegressor**

**from sklearn.metrics import mean\_squared\_error, r2\_score, mean\_absolute\_error**

**from sklearn.preprocessing import StandardScaler**

**from sklearn.pipeline import Pipeline**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**sns.set(style="whitegrid")**

**plt.rc("figure", figsize=(10, 6))**

**# Carregamento dos dados**

**data = pd.read\_csv('dataset.csv', header=None, names=['preco', 'tipo', 'area', 'quartos', 'bairro'], skiprows=1)**

**print(data.head())**

**X = data[['tipo', 'area', 'quartos', 'bairro']]**

**y = data['preco']**

**# Divisão dos dados**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)**

**# Criação do pipeline**

**pipeline = Pipeline(steps=[**

**('scaler', StandardScaler()),**

**('regressor', RandomForestRegressor()) # Da pra tentar com RandomForestRegressor() ou LinearRegression()**

**])**

**# Validação cruzada para avaliação do modelo**

**cv\_scores = cross\_val\_score(pipeline, X\_train, y\_train, cv=5, scoring='neg\_mean\_squared\_error')**

**print(f"MSE Médio com Validação Cruzada: {-np.mean(cv\_scores)}")**

**# Treinamento do modelo**

**pipeline.fit(X\_train, y\_train)**

**y\_pred = pipeline.predict(X\_test)**

**# Cálculo das métricas**

**rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred))**

**mae = mean\_absolute\_error(y\_test, y\_pred)**

**r2 = r2\_score(y\_test, y\_pred)**

**print(f"Erro Quadrático Médio: {rmse}")**

**print(f"Erro Absoluto Médio: {mae}")**

**print(f"Coeficiente de Determinação: {r2}")**

**# Gráfico**

**plt.scatter(y\_test, y\_pred, alpha=0.6)**

**plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)**

**plt.xlabel("Preço Real")**

**plt.ylabel("Preço Predito")**

**plt.title("Comparação entre Preço Real e Predito Com Os Dados Testes")**

**plt.xlim(0, 10000)**

**plt.ylim(0, 10000)**

**plt.show()**

**# Informações**

**print("\nPor favor, insira as informações do imóvel para previsão do preço conforme tabelas:")**

**print("\nApartamento: 1, \nKitnet: 2, \nHotel-Flat: 3, \nCasa: 4 \nComercial: 5")**

**print(**

**"'Asa Norte': 1, 'Sudoeste': 2, 'Asa Sul': 3, 'Águas Claras': 4, 'Taguatinga': 5, "**

**"'Lago Norte': 6, 'Lago Sul': 7, 'Guará II': 8, 'Samambaia': 9, 'Ceilândia': 10, "**

**"'Núcleo Bandeirante': 11, 'Jardim Botânico': 12, 'Noroeste': 13, 'Guará I': 14, "**

**"'Riacho Fundo': 15, 'Cruzeiro': 16, 'Park Sul': 17, 'Sobradinho': 18, 'Areal': 19, "**

**"'Vicente Pires': 20, 'Mangueiral': 21, 'Park Way': 22, 'Gama': 23, 'Octogonal': 24, "**

**"'Santa Maria': 25, 'Lúcio Costa': 26, 'Paranoá': 27, 'Recanto das Emas': 28, "**

**"'Candangolândia': 29"**

**)**

**# Dados do cliente**

**tipo = int(input("Tipo do imóvel (exemplo: 1 = Apartamento, 2 = Casa, etc.): "))**

**area = float(input("Área do imóvel em m²: "))**

**quartos = int(input("Número de quartos: "))**

**bairro = int(input("Código do bairro (conforme tabela): "))**

**dados\_usuario = pd.DataFrame([[tipo, area, quartos, bairro]], columns=['tipo', 'area', 'quartos', 'bairro'])**

**# Realiza a previsão com o modelo treinado**

**preco\_previsto = pipeline.predict(dados\_usuario)[0]**

**print(f"\nPreço estimado para o imóvel: R${preco\_previsto:.2f}")**

Testes realizados:

Validação Cruzada: Realizada para avaliar a robustez do modelo.

Testes com diferentes conjuntos de dados: Avaliação do modelo com diferentes amostras de dados para verificar a consistência nas previsões.

Análise de erros: Identificação de erros sistemáticos nas previsões do modelo.

Validação do modelo/código:

O modelo foi validado utilizando métricas como RMSE (Root Mean Squared Error) e R² (coeficiente de determinação). O desempenho foi comparado com benchmarks do setor e com resultados de outros modelos de regressão.

Coleta dos resultados:

Os resultados foram coletados a partir das previsões do modelo. As previsões foram comparadas com os preços reais dos imóveis para avaliar a precisão do modelo. Gráficos de dispersão foram gerados para visualizar a relação entre os preços reais e os previstos.