

Trabalho 2: Previsão de Preços Imobiliários

Descrição

Sistema de previsão de valores imobiliários (venda e aluguel) para a cidade de São Carlos/SP. O projeto implementa um pipeline completo de Machine Learning, desde a coleta e limpeza de dados até a disponibilização dos modelos através de uma interface web interativa. Este trabalho foi desenvolvido em colaboração com a Roca Imóveis, que proveu os dados necessários para o projeto.

1 Identificação do Problema

Objetivo: Desenvolver um sistema de previsão de preços de imóveis para auxiliar na tomada de decisão em transações imobiliárias na cidade de São Carlos/SP.

Problemática:

- O mercado imobiliário apresenta grande variabilidade de preços baseada em múltiplos fatores
- Necessidade de estimativas precisas tanto para vendas quanto para aluguéis
- Falta de transparência e ferramentas acessíveis para consulta de valores de mercado
- Dificuldade em avaliar o valor justo de um imóvel baseado em suas características

Solução Proposta:

- Sistema preditivo baseado em múltiplos modelos de machine learning
- Interface web interativa para consulta de estimativas em tempo real
- Análise integrada de características físicas e amenidades dos imóveis
- Fornecimento de intervalos de confiança para as estimativas

2 Pré-processamento

Fonte de Dados: Arquivos XML de listagens da imobiliária Roca Imóveis

Etapas de Limpeza (`clean_data.py`):

1. **Filtragem Geográfica:** Seleção exclusiva de imóveis da cidade de São Carlos
2. **Tratamento de Outliers:** Remoção de registros com `area_util > 10` (valores inconsistentes que representam erros de digitação)
3. **Tratamento de Valores Ausentes:**
 - `dropna()`: Remoção de linhas com dados faltantes em campos críticos
 - `fillna()`: Preenchimento de valores ausentes em features opcionais
4. **Separação de Datasets:**
 - `clean_data_sell.csv`: Dados específicos de venda
 - `clean_data_rent.csv`: Dados específicos de aluguel

Parsing de Dados (`parser/parse_roca.py`):

- Implementação de parser XML eficiente.
- Extração estruturada de todas as características dos imóveis

Resultado: Datasets limpos e estruturados prontos para análise exploratória e modelagem.

3 Extração de Padrões (Modelagem)

Análise Exploratória (Notebooks):

- `eda.ipynb`: Análise exploratória detalhada dos dados
- `modelling.ipynb`: Experimentação e treinamento dos modelos
- `data_cleaning.ipynb`: Processo iterativo de limpeza de dados

Modelos Implementados:

1. **OLS (Ordinary Least Squares)** - Regressão Linear Clássica
 - Modelo baseline para comparação de desempenho
 - Alta interpretabilidade dos coeficientes
 - Intervalos de confiança padrão
 - Arquivo: `OLS.pickle`
2. **GLM Gamma com Link Identidade** - Modelo Linear Generalizado
 - Adequado para variáveis resposta estritamente positivas (preços)
 - Distribuição Gamma captura a assimetria natural dos preços imobiliários
 - Intervalos de confiança estimados via simulação Monte Carlo (1000 iterações)
 - Arquivo: `gamma_identity.pickle`
3. **XGBoost** - Ensemble Gradient Boosting
 - Modelo de alta performance para previsão de venda
 - Modelo específico e otimizado para previsão de aluguel
 - Regularização L1/L2 para evitar overfitting
 - Suporte nativo a variáveis categóricas
 - Arquivos: `xgb_model.json`, `xgb_model_rent.json`

Features Utilizadas:

- **Características físicas:** número de quartos, banheiros, área útil, vagas de garagem
- **Comodidades:** piscina, churrasqueira, academia, área gourmet, playground, etc.
- **Tipo de imóvel:** Casa, Apartamento, Cobertura

4 Pós-processamento

Formatação de Saídas:

- Conversão automática para formato monetário brasileiro (R\$)
- Exibição simultânea de três previsões (OLS, GLM, XGBoost)
- Apresentação de médias e desvios padrão quando aplicável
- Interface clara e organizada para facilitar interpretação

5 Utilização do Conhecimento

Interface Web Streamlit (`gui.py`):

- Formulário interativo para entrada de características do imóvel
- Seleção de tipo de imóvel via dropdown
- Controles deslizantes e campos numéricos para características
- Checkboxes para amenidades disponíveis
- Previsão instantânea ao clicar no botão “Prever”
- Exibição de múltiplas estimativas com intervalos de confiança
- Visualização clara em formato de tabela responsiva

Pipeline Completo de Predição:

Entrada do Usuário → Padronização → Modelo(s) → Pós-processamento → Exibição

Casos de Uso Práticos:

1. Para Compradores:

- Avaliar se o preço pedido está compatível com o mercado
- Comparar ofertas de diferentes imóveis
- Fundamentar negociações com base em dados

2. Para Vendedores:

- Definir preço competitivo baseado em características reais
- Entender quais features agregam mais valor
- Evitar sobre ou subvalorização do imóvel

3. Para Imobiliária:

- Fundamentar propostas com estimativas técnicas
- Fornecer análises baseadas em dados aos clientes
- Acelerar processo de avaliação de imóveis

Implementação:

- Aplicação local via Streamlit
- Requisitos especificados em `requirements.txt` e `pyproject.toml`
- Instruções detalhadas de instalação e execução

Modelos Implementados

Para Venda:

1. **OLS (Ordinary Least Squares)** - Regressão Linear com intervalos de confiança
2. **GLM Gamma** - Modelo Linear Generalizado com distribuição Gamma e link identity
3. **XGBoost** - Gradient Boosting com suporte a variáveis categóricas

Para Aluguel:

1. **XGBoost** - Otimizado para predição de valores de aluguel

Funcionalidades

- Predição de valores de venda com três modelos diferentes
- Predição de valores de aluguel
- Intervalos de confiança para estimativas
- Suporte a múltiplos tipos de imóveis (Casa, Apartamento, Cobertura)
- Análise de amenidades e características do imóvel
- Interface web intuitiva e responsiva

Características dos Imóveis Analisadas

Características Básicas:

- Tipo de imóvel (Casa, Apartamento, Cobertura)
- Número de quartos
- Número de banheiros
- Vagas de estacionamento
- Área útil (m²)
- Área total (m²)
- Taxa de condomínio

Amenidades:

- Piscina
- Área de churrasco
- Playground
- Sauna
- Salão de festas
- Quadra esportiva
- Segurança 24h
- Lavanderia
- Closet
- Escritório
- Despensa