

# Predição de preços do mercado imobiliário russo

## Trabalho de IA

Lucas Gonzalez de Queiroz<sup>1</sup>, Victor Hugo da S. Ota<sup>1</sup>

<sup>1</sup>Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)  
Campo Grande – MS – Brasil

**Abstract.** *Although the housing market is relatively stable in Russia, the country's volatile economy makes forecasting prices as a function of apartment characteristics a unique challenge. Therefore, Sberbank, Russia's largest bank needs an algorithm that predicts realty prices. In this article, it is specified, in detail, how it was done and what was used to build an algorithm for realty prices.*

**Resumo.** *Embora a Rússia possua um mercado imobiliário relativamente estável, a economia volátil do país torna os preços das previsões em função das características dos apartamentos um desafio único. Portanto, O Sberbank, maior banco da Rússia precisa de um algoritmo que prevê os preços dos imóveis. Neste artigo está especificado detalhadamente como foi feito e o que foi utilizado para construção de um algoritmo de previsão de imóveis.*

### 1. Introdução

Nesse trabalho, desenvolvemos um algoritmo que usa um amplo espectro de recursos para prever os preços dos imóveis. Foi utilizado de um conjunto de dados rico que incluía dados de habitação e padrões macroeconômicos para prever os preços de imóveis financiados pelo Sberbank.

#### 1.1. Descrição do problema

Os custos da habitação exigem um investimento significativo de consumidores e desenvolvedores. E quando se trata de planejar um orçamento - seja pessoal ou corporativo -, a última coisa que alguém precisa é a incerteza sobre uma das suas maiores despesas. Sberbank, o maior e mais antigo banco da Rússia, ajuda seus clientes a fazerem previsões sobre os preços dos imóveis, de modo que os inquilinos, desenvolvedores e credores estejam mais confiantes quando assinam um contrato de arrendamento ou adquirem um edifício.

Embora o mercado imobiliário seja relativamente estável na Rússia, a economia volátil do país torna os preços de previsão em função das características dos apartamentos um desafio único. Interações complexas entre recursos habitacionais, como número de quartos e localização, são suficientes para tornar as previsões de preços complicadas. Adicionar uma economia instável à mistura significa que o Sberbank e seus clientes precisam de mais do que modelos de regressão simples em seu arsenal.

O Sberbank desafia a desenvolver algoritmos que utilizam um amplo espectro de recursos para prever os preços dos imóveis. Um modelo de previsão preciso permitirá que o Sberbank forneça mais certeza aos seus clientes em uma economia incerta.

## 2. Biblioteca Utilizada

O XGBoost é uma biblioteca otimizada de gradiente distribuído, desenvolvida para ser altamente eficiente, flexível e portátil. Ele implementa algoritmos de aprendizado de máquina sob o framework Gradient Boosting. O XGBoost fornece um aumento de árvore paralela (também conhecido como GBDT, GBM) que resolve muitos problemas de ciência de dados de forma rápida e precisa. O mesmo código é executado no principal ambiente distribuído (Hadoop, SGE, MPI) e pode resolver problemas além de bilhões de exemplos.

Foi utilizada principalmente para realizar os treinos e testes em cima dos dados.

### 2.1. Outras bibliotecas:

- Pandas: Leitura do arquivo csv.
- Matplotlib: Plotar o resultado.
- Sklearn: Realizar o fit dos dados.
- Lightgbm: Configuração dos parâmetros.
- Datetime: Classe para manutenção de data e tempo.
- Seaborn: Biblioteca utilizada para visualização de dados, porém não utilizada no resultado final;

## 3. Leitura e Fit dos dados

Foi feito a leitura dos dados "train.csv", "test.csv" e "macro.csv" utilizando a biblioteca pandas e armazenando-os nas variáveis 'treino', 'teste' e 'macro' respectivamente. Então foi feito uma variável 'xtreino' e 'xteste' para separar os dados. No fit do dados foi utilizado o "preprocessing" do sklearn.

## 4. Configurando Classificadores

```
# Config de parametros do XGB
xgb_params = {
    'eta': 0.05,
    'max_depth': 5,
    'subsample': 0.7,
    'colsample_bytree': 0.7,
    'objective': 'reg:linear',
    'eval_metric': 'rmse',
    'silent': 1
}

# Config de parametros do LGB
lgb_params = {'objective': 'regression', 'metric': 'rmse'}
```

Figure 1. Parametros do XGBOOST

Como visto na Figura 1, eta: 0.05 foi aproximada de zero, pois foi a estimativa mais conservadora para evitar overfit; max-depth:5 numero da profundidade da arvore do xgboost; métrica: rmse; silent 1: modo silencioso; objective: reg linear= utilizamos regressão linear;

## 5. Realizando Treino, Testes

Para realizar os testes, referentes a Figura 2, os dados foram adicionados em uma matriz, assim, simplesmente, aplicamos XGBoost aos dados e o resultado foi plotado em um CSV, procuramos realizar os testes da maneira mais simples possível para avaliar o XGBoost de forma limpa.

```

58
59 # realiza treino e teste aplicando XGB nos dados
60 dtreino = xgb.DMatrix(x_treino, y_treino)
61 dteste = xgb.DMatrix(x_teste)
62
63 #aplica os resultados no cv final
64 cv_saida = xgb.cv(xgb_params, dtreino, num_boost_round=1000, early_stopping_rounds=20,
65                 verbose_eval=50, show_stdv=False)
66
67 num_boost_rounds = len(cv_saida)
68 model = xgb.train(dict(xgb_params, silent=0), dtreino, num_boost_round= num_boost_rounds)
69 #faz a prediccao
70 y_predict = model.predict(dteste)
71 # formata a saida, prediccao em price_doc
72 saida = pd.DataFrame({'id': id_teste, 'price_doc': y_predict})
73 saida.head()
74
75 # geraçao saida
76 saida.to_csv('xgb.csv', index=False)

```

**Figure 2. Código referente a aplicação do XGBOOST**

## 6. Conclusão

No fim, foi implementado um algoritmo de previsão de imóveis bastante simples. Após os testes, podemos concluir que os resultados foram satisfatórios. Tentamos utilizar outros classificadores, porém, com o uso de apenas o XGBoost, foi possível realizar uma boa previsão, mantendo o código bem simples e enxuto, assim, mostrando que nem sempre é necessário fazer um código complexo para realizar manipulação dos dados para obter bons resultados.

## References

DMLC. Xgboost. Disponível em: <https://github.com/dmlc/xgboost> . Acesso em: 04 Ago. 2017.

Kaggle. Lgbm. Disponível em: <https://www.kaggle.com/schoolpal/lgbm-lb-0-3093-0-3094> . Acesso em: 04 Ago. 2017.

Kaggle. Sberbank russian housing market. Disponível em: <https://www.kaggle.com/c/sberbank-russian-housing-market> . Acesso em: 04 Ago. 2017.

[DMLC ] [Kaggle a] [Kaggle b]