Predicting the Price of Russia Real State

Abstract

The goal of this project was to use Regression models to Predict the Price of Real state in Russia. The real estate market has been in a growth phase for several years, which means that you can still find properties at very attractive prices, but with good chances of increasing their value in the future. I worked with data provided by Kaggle . I have used many visualizations along with many regression models to achieve promising results for this problem.

Design

This project was contemplated after the fall of the Soviet Union and after the great development that took place in the real estate market in Russia, especially the capital, Moscow. The project helps to find the right opportunities to get the best prices based on its features

Data

The dataset consists of lists of unique objects of popular portals for the sale of real estate in Russia. The dataset contains over 5 million real estate samples. The dataset has 13 fields, which are divided as follows: 12 of them as features and the remaining column is the price of the real state.

Algorithms

Data Cleaning

1. Remove outliers from Price, Area, and Kitchen Area columns.
2. Drop the real state with no. of rooms = -2 as it isn't logical.
3. Replace the no. of rooms with 0 instead of –1.
4. Drop unwanted columns like time.

Feature Engineering

Create a pivot table to add a new feature (Average Price) which depends on latitude and longitude.

Models

Linear regression, degree 2 polynomial regression and XGB regressor models were used before settling on XGB as the model with the highest accuracy.

Model Evaluation and Selection

The entire training dataset of over 5 million records and to train the model on that size was time and computation consuming. So, I only used a sample of 500K of the data. First, I divided the sample into 3 parts which are 60% Training, 20% Validation, and 20% Testing. And then I evaluated the 3 models using the 3 parts.

The best model is XGB and the Score was as follows:

- `0.962 on the training data`
- `0.941 on the validation data`
- `0.938 on the testing data`

Tools

- NumPy and Pandas for data manipulation
- Scikit-learn and xgboost for modeling
- Matplotlib and Seaborn for plotting