Apartment Rental Price Prediction

Abtin Mahyar
Department of Computer Science
Shahid Beheshti University
email:abtinmahyar[at]gmail[dot]com

Abstract

This report gives an overview of the various machine learning algorithms implemented to predict rental price of an apartment based on different features that a rental offer has. Exploratory data analysis was performed on the dataset in order to take insights from the data. Feature selection and feature extraction was performed using different machine learning algorithms with the help of Scikit-learn package, and various machine learning models was used to build supervised learning regression, that provided a root mean squared error of 149.33 on the test dataset, which is an acceptable amount of error according to the rental price range. The dataset was obtained from the popular data science competition portal, Kaggle.

1 Introduction

Making predictions is something a business or system depends on, which is indirectly dependant on data analytics. Data analytics is the science of analyzing raw data in order to make conclusions about that information.This analysis can then be used to optimize processes to increase the overall efficiency of a business or system. The dataset here, that I have obtained from Kaggle, is a sample of various types of apartment rental offers. This project aims to develop a price prediction model with comparing different popular models and choose the best one that can recognize the patterns in the data.

2 Methodology

The performed methodology involved exploratory data analysis, preprocessing (which contains outlier detection, feature extraction, feature selection, handling null values, scailing, and encoding), regression, and model evaluation which will be further elaborated on, in the following subsections.

2.1 Exploratory Data Analysis

The data was presented in the form of a csv file on the Kaggle data science competition portal. The data set contains most of the important properties, such as living area size, the rent, both base rent as well as total rent (if applicable), the location (street and house number, if available, ZIP code and state), type of energy etc. This dataset has 268,850 records with various types of fields which each record refers to a particular mobile device. Each record has 48 attributes and a total rent price which were described completely on the dedicated dataset page.As it can be suspected based on the attributes, some of the features are useless, illegible, or duplicated, which means they must be discarded during the preprocessing section. There are lots of null values in some of the entries in the dataset which should be filled or dropped during preprocessing section. Distribution of the numeric features in the dataset, after prepocessing is shown in Figure 1.

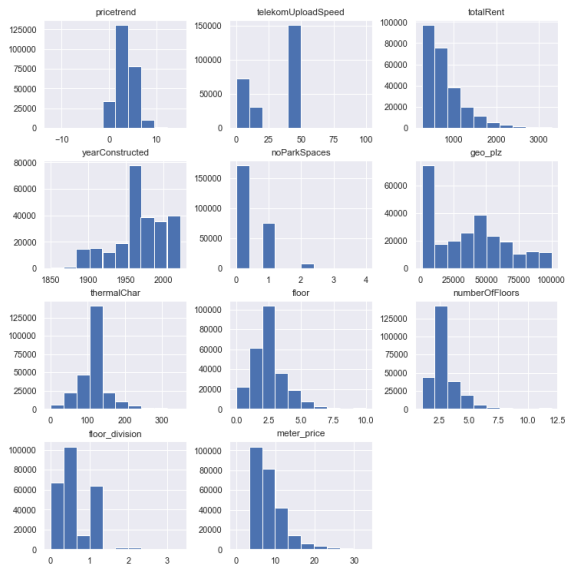

Figure 1: Distribution of each feature in the dataset

As it can be seen from the Figure 1, all of the

distributions of numeric features in the dataset are normal and have acceptable skewness. Correlation analysis between each feature was performed on the data and the results can be seen in Figure 2.Top correlated features to the target variable (total rent) is shown in Table 1.

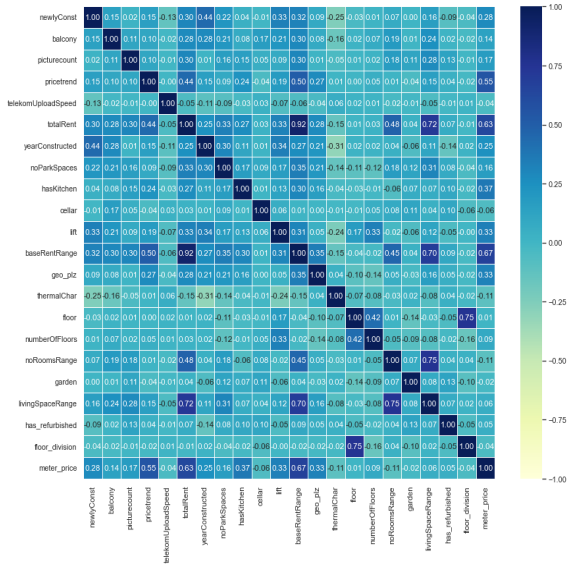

Figure 2: Correlation between each feature in the dataset

Feature	Value
Base Rent Range	0.91
Living Space Range	0.72
Price for each Meter	0.62
Number of Rooms Range	0.47
Price Trend	0.44

Table 1: Top correlated features to the target variable

There are some other visualisation work which will be represented in the following in order to give meaningful insight of data and can help us for encoding the ordinal categorical variables in preprocessing section.(All the violin plots are sorted based on median of each category). Figure 3 represents the distribution of apartment rental offers based on their city. As it can be seen from the pie chart, the majority of the offers are from Nordrhein Westfalen and Sachsen, and the minority of the offers are from Saarland, Bermen, and Hamburg.

Figure 4 demonstrates total rental price in different cities in Germany. According to the violin plot, Hamburg, Berlin, and Bayern are the most expensive cities in the Germany; whilst,

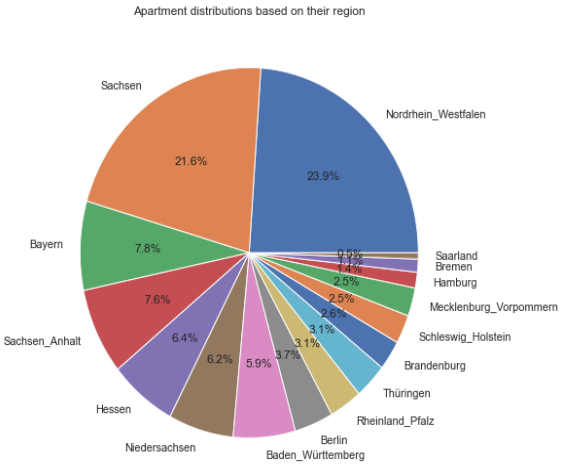

Figure 3: Rental offer distributions based on their region

Sachsen Anhalt, Thuringen, and Sachsen are the least expensive ones. Moreover, according to this figure and 3, cities which have less total rent price formed the majority of the total offers, and the minority of the offers are form expensive cities.

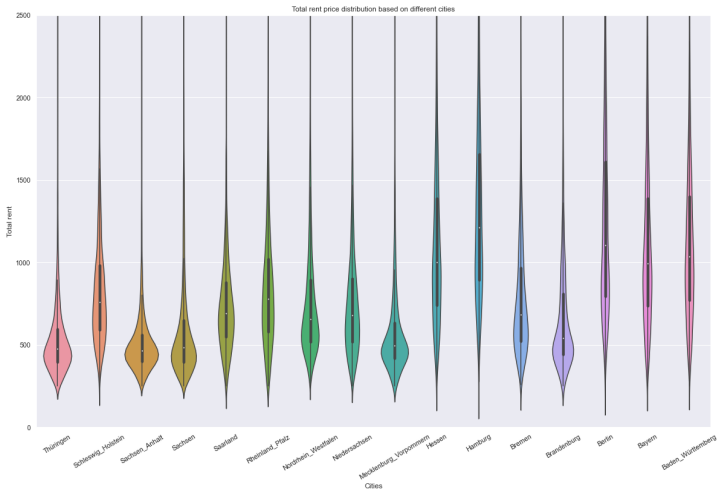

Figure 4: Total rental price offer distributions based on their region

Figure 5 demonstrates apartment distributions based on their condition in Germany. According to the pie chart, most of the apartments has well kept condition; whilst, the minority have bad condition.

Figure 6 illustrates total rent price distributions based on their interior quality in Germany. According to the violin plot, luxury, and sophisticated offers are more expensive than the other categories in the dataset. However, number of these offers are less than the others.

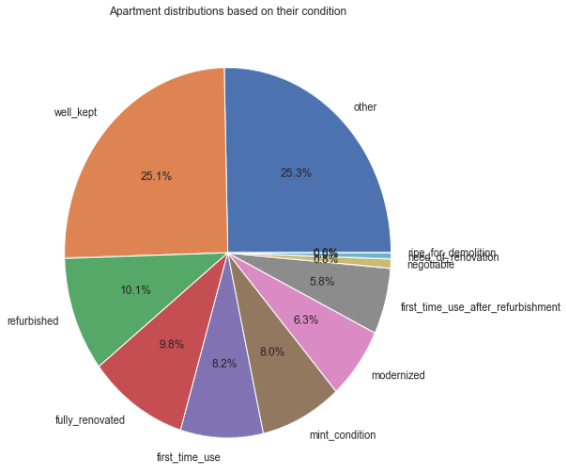

Figure 5: Apartment distributions based on their condition

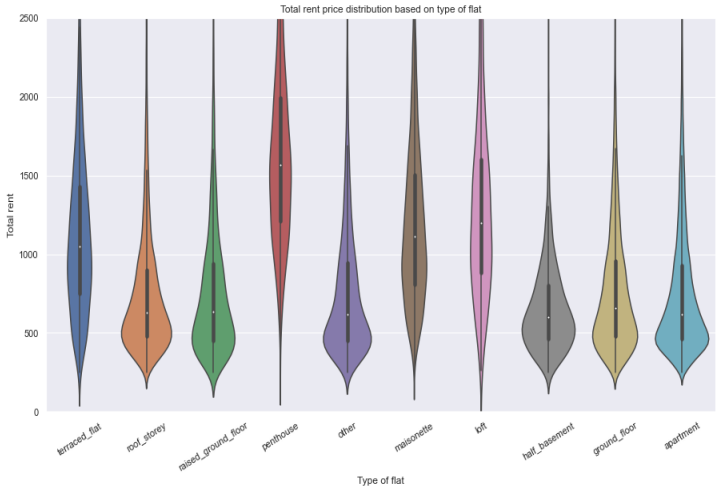

Figure 7: Total rent price distributions based on their type of flat

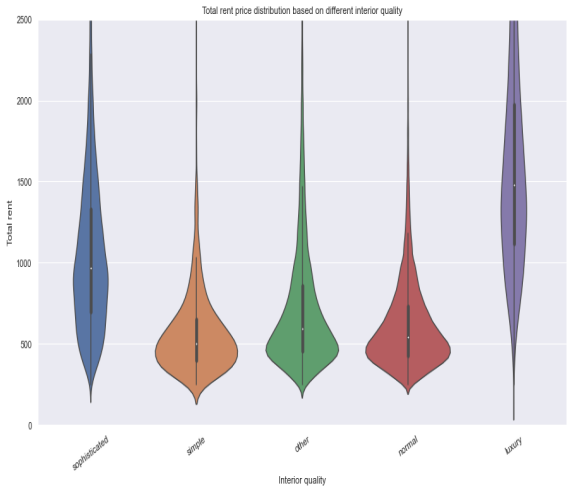

Figure 6: Total rent price distributions based on their interior quality

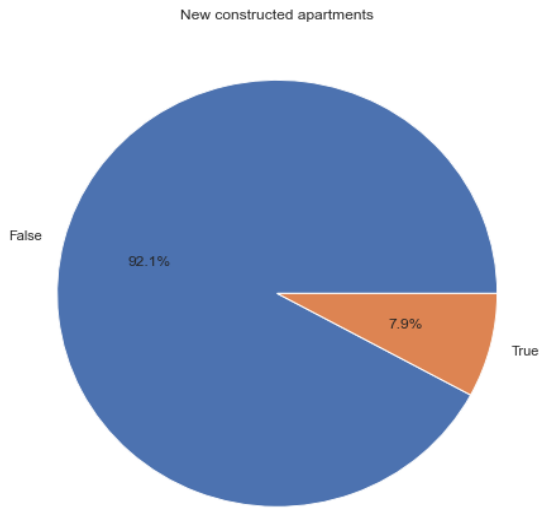

Figure 8: Newly constructed apartment offers distribution

Figure 7 illustrates total rent price distributions based on their type of flat in Germany. According to the violin plot, penthouse offers are much more expensive than the other categories in the dataset. However, number of these offer are less than the others.

Figure 8 illustrates newly constructed apartment offers distribution in Germany. According to the pie chart, most of the offers are not constructed recently and just 7.9% of the whole dataset were constructed newly.

2.2 Preprocessing

The preprocessing involved handling null entries using different techniques. First, with using an insight which has achieved form exploratory data analysis, I drop the columns which were duplicated, or have same information with another column, or has so many null entries and that was not possible to fill them with a logical and appropriate value. Next, null values in numeric features filled with an appropriate value with respect to each column with zero, or aggregating other values with mean or mod functions. Also, total rent price which is our target variable in this project, has null values in some of it's entries, which filled with the sum of base rent price, service charge, and heating costs (based on the documentation on the Kaggle), After that, these values were dropped since they are directly correlated to the target variable to prevent the model from outperforming. Null values in object variables filled with

new sampled value in order to prevent dropping most of the columns and loss information. Outlier detection was performed on every feature in the dataset, and entries with irrational values were drop from the data. There were 3 nominal categorical variables in the dataset: city, heating type, and type of flat. After reducing categories of these variables by aggregating similar categories or variables that formed the small part of the data, these variables encoded to vectors representing their state using one hot encoding. There were 4 ordinal categorical variables in the dataset, which were encoded using custom mapping with respect to their value. Also, duplicated entries were dropped. Other, object variables which need NLP in order to extract feature from them, were dropped.At the end, the final dataset has 254,498 entries with 55 columns and a target variable (5.34% of the records detected as outliers and dropped). More over, categorical and numerical variables converted to their appropriate data types and memory usage of the dataset reduced about 72.5% by using a custom function, in order to have faster model training and better performance. Scailing was performed using standard scaler which transforms the data with following formula:

$$x_{new} = \frac{x - mean(X)}{stdev(X)}$$

2.3 Feature Extraction

In this step two different extracted from the previous features based on their correlation and value to the target variable. First one is price for each meter of the apartment which is the division of base rent price and living space in squared meters. The second one is the floor division which is the division of the number of floor of rental offer to the total number of floors that the apartment has.

2.4 Regression

Several different regressors were applied on the dataset which were generated by carrying out the preprocessing phase, but in the following only the top regressors with the highest accuracy will be discussed for the evaluation of performance on test dataset, These regressors are listed in Table 2. Also, for evaluating model performances root mean squared error were calculated. In addition, k-fold cross validation was performed on the training set before evaluating the test data set, and in this project, for all regressors, the default k is set to 5. Advanatages of performing k-fold cross validation included that it prevents overfitting of the regressor model and provides generality to the model that could later better predict an independent data set, such as the test data set. The original dataset has split to train and test set, The size of test data set here is 12,725 instances (5% of the data, since we have a large dataset) and train data set is 241,773 instances. Also, since dataset was too large to fit in machine learning models, PCA was performed on the dataset in order to reduce dimensionality of the data with saving much information that is possible. I used 0.95 proportion of variance in order to reduce number of features that has the least variance, the resulted dataset has 42 column.

Model	Train Accuracy	Test Accuracy
Extra Trees	1.23	154.31
Random Forest	59.57	158.36
Bagging	73.22	169.18
Gradient Boosting	171.61	174.58

Table 2: Top models based on their performances(RMSE)

2.4.1 Extera Trees

Extra Trees is an ensemble machine learning algorithm that combines the predictions from many decision trees. This algorithm works by creating a large number of unpruned decision trees from the training dataset. Predictions are made by averaging the prediction of the decision trees in the case of regression or using majority voting in the case of classification. There are some hyperparameters to tune in this model, but since this model takes about 10 minutes for training an evaluating by test set on average, and due to the hardware limitation that I had, I decided to use default hayperparameters for this model which has an acceptable error on the test set (some possible solutions to tackle this problem is provided in the future work section). This model achieved RMSE of 2.0 on the training set and RMSE of 149.33 on test set.

3 Related Work

There are number of techniques that can speed up the preprocessing and model training phases. Using python's multiprocessing which can split a process to multiple processes and Dask package which split the dataset in partitions we can speed up our preprocessing phase. Result of applying these technique to the dataset are listed in Table 3.

Technique	Time(s)
Raw Pandas data frame	3.11
Multiprocessing(8 processes)	3.43
Dask(3 partitions)	3.39

Table 3: Consumed time using different techniques in preprocessing

Figure 9 demonstrates consumed time by different number of processes using pythons's multiprocessing. According to this figure, with splitting the preprocessing to 8 parallelized processes, we have the least consumed time.

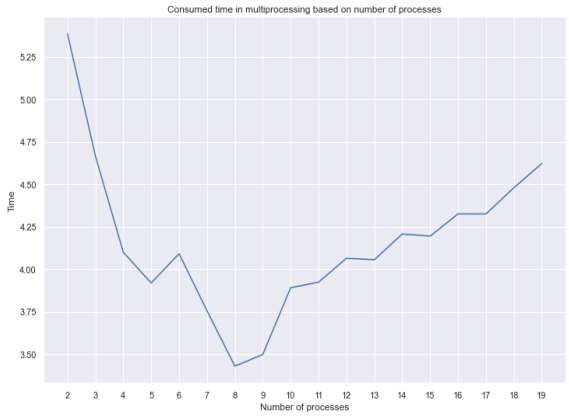

Figure 9: Consumed time by different number of processes

Figure 10 illustrates consumed time by different number of partitions using Dask package. According to this figure, with splitting the the dataset to 3 partitions, we have the least consumed time.

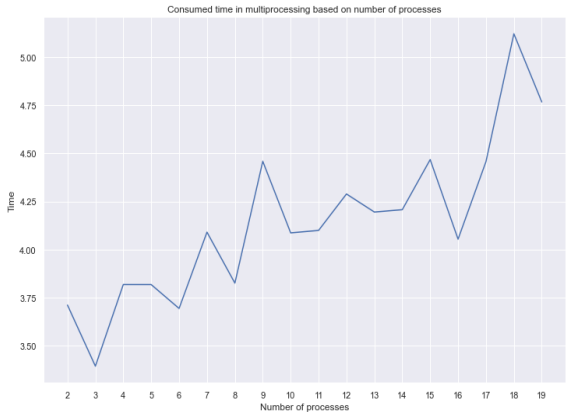

Figure 10: Consumed time by different number of partitions

4 Future Work

There are number of things that can help the models to achive better performance in this task. First of all, we can use different types of scailing in perprocessing section instead of just using standard scaler and fit the data with different number of samples which will be created with various scailing method, and evaluate the model and choose the best sample.
Second, we can make different sets form the original dataset before doing the regression task with using PCA with different values of porportion of variance and evaluate each set and choose the one with the lowest amount of error (maybe the model which fits with raw data, has the better performance than the others.)
Third, by doing a NLP task, we can extract some useful features form objects variables and add them to the original dataset in order to increase the performance of the model instead of just dropping them.
Finally, with using more powerful hardware we can tune the hyperparameters of the final model in order to increase the performance of the model and reduce overfitting. Also, we can use a small subsample of the data, instead of using all the large dataset in order to find best hyperparameters which could probably decrease the amount of error.

5 Conclusions

As it can be concluded from above sub sections, the best model that can predict the data is Extera Trees Regressor with the default hyperparameters. The scatter plot of predicted and actual values of total rent price is illustrated in Figure 11. As it can be seen from the plot, data points in the plot are making a line on $y = x$, which means that the model predict the price precisely.

Figure 11: Scatter plot of actual and predicted values

References