Predicting Airbnb Listing Price Across NewYork

Technical Report · February 2020

DOI: 10.13140/RG.2.2.28089.70246

Kala Brahmaih
Letterkenny Institute of Technology

Predicting Airbnb Listing Price Across NewYork

Brahamaiah Kala (L00151250)

***Abstract*— Airbnb is an online marketplace where accommodation, mainly homes and tourism activities can be arranged or offered.No real estate websites are open, nor does the company hold events; this organization operates as a dealer and collects fees from each reservation. Among the worldwide holiday destinations Airbnb has become increasingly popular. Consequently, large data sets with rich characteristics are obtained from the Airbnb listings. In this mission, forecasting Airbnb prices with different machine learning approaches in three cities New York City (NYC) by finding the dependent variables and independent variables inorder to calculate the correlations between them and predicting the model.**

***Index Terms*— Prediction, Regression analysis.**

I. Introduction

THE Pricing a rental property on Airbnb is a difficult task for the owner because the number of customers on the premises is determined. On the other hand, customers must assess the offered price with minimal knowledge of the property's optimum value. In order to help the landlord and consumers with price estimation, this paper seeks to build a robust pricing model utilizing computer intelligence, deep learning and natural language processing techniques, despite the minimum accessible description of the house. Characteristics of contracts, owner property and customer reviews will include the predictors and a variety of approaches from linear regression to tree models.rates are usually determined empirically by the owners, unlike hotels with their own pricing system. The new hosts as well as established hosts of updated lists face challenges when it comes to setting rates that are fairly high yet do not lose popularity. On the consumer side it is still important for him to know whether the current price is reasonable and whether it is a good time to book the rooms, even though they can compare the price between other related listings.

dataset chosen given the description information about the customer, host, availability, nearer places, geographical areas etc. which interns contains the require to attain certain to predictions with conclusions.

Airbnb rentals are subject to several criteria each night and separate the form of data into 4 groups, including constant, categorical, date features, and text. 16 features have been extracted from the dataset for further analysis and prediction.

Only few attributes have been classified and categorized for further tasks room type private room, shared room, entire room/apt locality neighbourhood, latitude, longitude, neighbourhood group ,reviewsnumber of reviews, last review, reviews per month booking related calculated host listings, availability .The ground truth mark is the actual price so we use a variety of methods like regression.

II. Related Work

Since wide datasets are accessible in many cities, Airbnb price prediction is common.

In the 2015 Multi-Scale Affinity Propagation Li et al. [1] used the price recommendation to show that the price prediction improves significantly.

Wang et al. [2]have been dealing with Airbnb datasets in 33 cities in 2017, using an ordinary least compressed and quantile analysis, finding 25 quality determinants in a 180,533 array of lodging rentals.

A related paper by Teubner et al. [3] explores the characteristics attributed to credibility and its impact on linear regression pricing.

Throughout 2019, in the NYC dataset, Kalehbasti and other collaborators [4] used numerous learning processes and nostalgic analyzes for the estimation of Airbnb rates, hitting 0.6901 R2 on research datasets.

Lewis [5] has recently predicted that in London Airbnb prices will be higher than other Kaggle-leaving machine models by using machine learning and in-depth learning and that XGBoost will provide the best accuracy (R2 = 07274) [6].

Although several initiatives on market listings have been carried out, none of them have been carried out in various cities.

III. Implementation

To order to achieve strategic goals and objectives, management is the mechanism that turns policies and proposals into behavior.We will consider our data and analysis prior to deployment, which includes cleaning and data processing, and modeling and prediction.

Implementation undergoes following steps by first understanding the dataset, choosing the platform and language to do the further analysis in achieving the goal, analysis the data and finding the correlation,apply the algorithms to predict the accuracy.

A. Understanding the data

Datasets: Airbnb is a company which serves the travellers to lodge, rent, homestay a place as per their requirements to

Under the supervision of Dr. Shagufta Henna, Letterkenny Institute of Technology, Letterkenny, CO. Donegal

experience their travelling. This company allows the people to rent the place online, in this stage the requirements varies from guest to guest and the suggestions, neighbourhood amenities, the filters are not as expected to every customer/guest this interns leads to either compromise or quit the site. Finding the relevant and most filters is the is challenging as it varies to the different geographical areas.

Regarding Airbnb listings in New York, Kaggle datasets have been used. [7, 8, 9]. The three databases include a detailed summary table of 96 columns / features of raw content. A total of 44317 lists are found in the NYC dataset, 59881 are in the Paris databases and 22552 in the Berline dataset. For each dataset, we partition the dataset into a train: validation: test ratio of 7:2:1.

Features:

Features are only chosen where they are insightful and can be connected to the price label. We have therefore disabled features such as I d and host I d, which tend to be a static, industry and country code apps, where most of the details have the same meaning, and features such as licenses and authority terms, where the remainder is negative.

- **Label:** Based on the abnormally high prices of datasets, we have tested different power transformations as well as logarithmic transformations for label change and found that square-root transformation and logarithmic transformation function well with the price estimation and have used two methods, data threshold and label transformation. And to find that logarithmic transformation of the square root works well for the prediction of prices.
- **Continous Features:** We have defined and removed highly correlated features and received 28 continuous features from our data. For the price linked features, we have filled the null with 0.
- **Categorical features:** We implemented one-hot encoding directly for most categorical items, while we encoded them into vectors using dictionary building and mapping for a small portion of the list features, such as the facilities and house name. Total 16 encoded features have been identified.
- **Date features:** We have 3 date characteristics (host, first revision, last revision) that we convert into constant values by entering the null value with the mean time period and extracting the earliest time value from all date values.

B. Apparatus used

Azure Databricks

Azure Databricks is a Microsoft Azure Cloud services platform based on Apache Spark.Databricks has been built with the creators of Apache Spark and is compatible with Azure to provide one-click set-up, simplified workflows and an open environment for collaborations between data scientists.Azure Databricks is a quick, easy and collaborative analytics service based on Apache Spark. With large data repositories, the data is processed in batches (raw or structured), or is transmitted almost in real time via Kafka, event center, or IoT platform, in Azure Data Factory.

Databricks allows developers and data scientists in a wide range of advanced analytical algorithms in SQL, Python, Scala, Java and R.

The workspace is a place where all the Data Bricks resources are available. The workspace organizes objects (notebooks, databases, dashboards and experiments) into directories, providing access to computing tools and data artifacts.Few important process and features to be used:

1) **A Data Bricks cluster** consists of a series of computer resources and configurations for workloads such as data technology, data science and analytics, for example, ETL pipelines for production, streaming analytics, ad hoc analytics and machine education.You run the workloads on a notebook or as an automated task as a set of commands.Databricks distinguishes between interactive and automated clusters.In interactive notebooks you use interactive clusters for collaborative data analysis.In interactive notebooks you use interactive clusters for collaborative data analysis.

   Two sets are shown in the Clusters page: Interactive and Automated Clusters. Every list includes: Cluster name, State number, driver sort and operating nodes Databricks version Runtime, Cluster creator or job owner.
2) **Notebooks** Notebook is a web-based document interface which contains running code, text and visualizations such as dashboards, notebook configurations and dashboards with plugins, creating dynamic networks for workflows of notebooks.
3) **data** Build tables from imported data directly. In the Databricks internal metastore, the table schema is saved by default and you can configure and use external metastores as well.The distributed data system mounted on Databricks working surfaces and available on the Databricks clusters can be imported into Databricks's file system (DBFS), DBFS API, Databricks file system utilities, Spark APIs and local file APIs.
4) **Job and libraries** A job is an immediate or scheduled manner of running a JAR or notebook. The other way to operate a notebook is through the notebook interactively.it is an imminent or planned way of running a notebook.

   Remote libraries can be built with the Program utility directly in a notebook session in Databricks Server Python libraries.Databricks advises that you use this approach wherever possible as libraries installed on a notebook will not impact library installed on other notebooks, although all notebooks are operating on the same cluster.

PySpark in databricks, has been released in order to support the collaboration of Apache Spark and Python, it actually is a Python API for Spark. In addition, PySpark, helps you interface with Resilient Distributed Datasets (RDDs) in Apache Spark and Python programming language.

Features of pyspark: PySpark has a number of libraries to write programs with efficiency.

1) **PySparkSQL:** A PySpark library which uses a large number of structured or semistructured data to apply

sql analysis.PySparkSQL may also allow us to use SQL queries.The connection to Apache Hive can also be made.The framework DataFrame, which is like that of a chart of the relational database management system, was developed by PySparkSQL.

2) **MLlib:** The data parallel technique is used by this library to store and work with data.The MLlib library's machine-learning Interface is very easy to use.For the classification, regression, clustering, collaborative filtering and size reduction and underlying primitive optimization MLlib supports numerous machine learning algorithms.

C. Data Analysis

The cluster is created, required dataset is added in the data in the tablular form, start coding on notebook by choosing workspace in dashboard. A file storage path and details of the datasets such as the path and table name is given automatically, which makes works easier. Now, the data given is to be analysed, cleaned and find the correlation, apply the machine learning algorithms to get prediction.The following steps:

1) **Loading the data:** import the libraries needed and load the dataset. to know better about the data,code must be number of columns , heads,datatypes to be known.
2) **Data Cleaning:** In AirBnb datasets multiple categories from the head of the airbnb dataset, allowing for detailed analysis of the data.There are also some details groups, though, where it is not required and therefore we have to clean up these records.Many missing values are later converted to zero or completely removed from the background and the utility of the knowledge.
   a) Deleting redundant columns.
   b) Dropping duplicates.
   c) Cleaning individual columns.
   d) Remove the NaN values from the dataset.

   Such data are clean without any procedure, but some columns are unrelated to the review and should therefore be removed.There is a lot of missing details about 'last review' and the columns 'review per month'. But 'last review' feature is deleted due to the background of each category and substitute for zero for all missed values in 'Reviews per month'.
3) **Exploring and Visualizing Data** The data is evaluated and the values of features and interactions between features are visualized. Data analysis gives details in graphs and charts that are easily understood in order to learn about various hosts and areas.By eexploring the data, the following results have been achieved and visualised such as
   a) Rooms having most number reviews,expensive rooms in newyork is visualised in heatmap.
   b) Map of Neighbourhood group and neighbourhood groups are visualised in scatterplot.
   c) Number of reviews is visualised in bargraph.
   d) Get Correlation between different variables all Neighbourhood Group
   e) Neighbourhood
   f) Room Type
   g) Relation between neighbourgroup
   h) Availability of Room
   i) Map of Neighbourhood group
   j) Map of Neighbourhood
   k) Availabity of rooom

After visualising and finding the correlations, Machine learning algorithms are used to find the predictions.

IV. Methodology

After the preparation, analysis and finidng the correclation for which the prediction to be calculated, Now the algorithm to predict must be chosen.

There are some factors that help us to pick the algorithm and choose the algorithm to be used.If the dataset is small, we must choose models with a low complexity (models with a limited number of parameters).The model we need is a regression model and we don't just want great precision for quick training time. We need reasonable precision.We have no linear correlations in the relationship maps, but they also reduce precision if we choose linear models.The amount of features is 16. For this case we would need svm / svr if we had very few features.

Based on the above conditions, the problem can be solved by many algorithms as Linear Regression, Gradient Boosted Regression and xgboost.

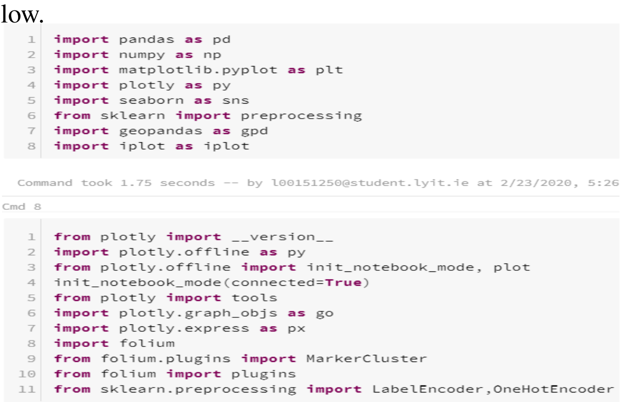
1) **Linear Regression:** Linear regression model means the estimation of the values of the data representation coefficients.Linear regression conducts the function of predicting a vector dependent value (y) based on a specified equation (x). Therefore, this technique of regression considers a linear relation between x (input) and y(output). The definition is therefore Linear Regression.
2) **Gradient Boosted Regression:**This method groups all the weak predictive models and built the accurate model, which typically are decision trees. This combines both the decision tree and the boosting algorithms by considering the previously built decisions trees which ts better to improve the accuracy where boosting automatically detects the best fit.
3) **XGBoost:** XGBoost is a decision tree assembly boost tree algorithm.It varies in two ways from the random trees.Random forests first construct a tree on their own while XGBoost produces a tree on a single basis.Furthermore, random forests merge findings from all trees at the top, and XGBoost puts results together

Algorithm:

a) Load the data.
b) Call DataFrame display()to show a preview of data in Databricks.
c) Preprocess the data.To see the type of each column, print a scheme for our dataset.
d) Change the string datatypes into integer. usually in databricks, by default it considers the datatypes , hence it is necessary to change the datatypes to int.
e) Define as lm,GBoost and xgb.to be used.
f) Define the necessary features i.e,host id, neighbourhoodgroup, neighbourhood, latitude, longitude, room type, minimum nights, number of reviews, reviews per month, calculated host listings count, availability.
g) Split data into training and test sets, predict the test of chosen algorithm.training and testing has been assigned in 80:20 ratio.
h) Using METRICES, This function measures the consistency of the subset in the Multi-label Classification: the expected label set will exactly match the corresponding collection of labels in y true.
i) Calculate R square and RMSE value. R Square: This scale varies from 0 to one. It reveals that the proposed model does not improve forecast over the middle form, and one implies a perfect estimate. R-squared is intuitive.Improved the regression analysis contributes to relative R-squared changes. RMSE:The standard deviation of residuals (foresight errors) is Root Mean Square Error (RMSE).Residuals are an indicator of the variance from the regression data points; RMSE is an example of how these residuals are allocated.In other terms, it describes how the details was clustered around the best line.
j) Predict and evaluate outcomes.
k) Compare the actual and predicted values. plot a chosen model prediction graph.

V. Software package description

In this section, the source code for the entire process of this project i.e, importing the necessary libraries, data cleaning, visualisation, finding the RSquare and predictions is given below.



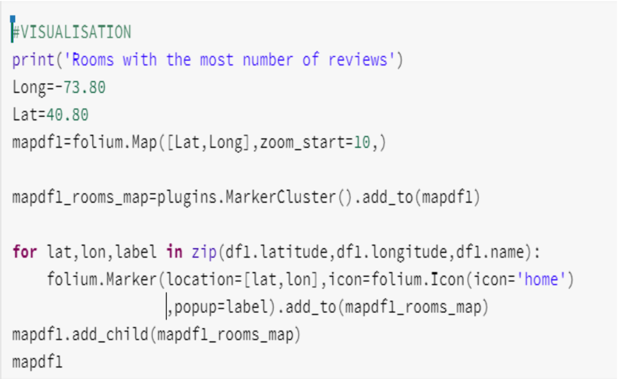
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly as py
import seaborn as sns
from sklearn import preprocessing
import geopandas as gpd
import iplot as iplot
```

```
Command took 1.75 seconds -- by l00151250@student.lyit.ie at 2/23/2020, 5:26
```

```
Cmd 8
```

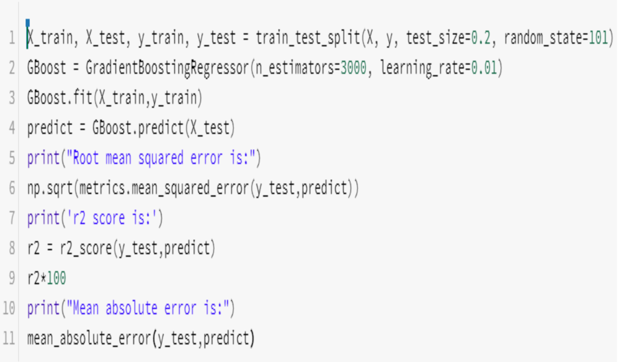
```
from plotly import __version__
import plotly.offline as py
from plotly.offline import init_notebook_mode, plot
init_notebook_mode(connected=True)
from plotly import tools
import plotly.graph_objs as go
import plotly.express as px
import folium
from folium.plugins import MarkerCluster
from folium import plugins
from sklearn.preprocessing import LabelEncoder,OneHotEncoder
```

Source code 1. importing libraries



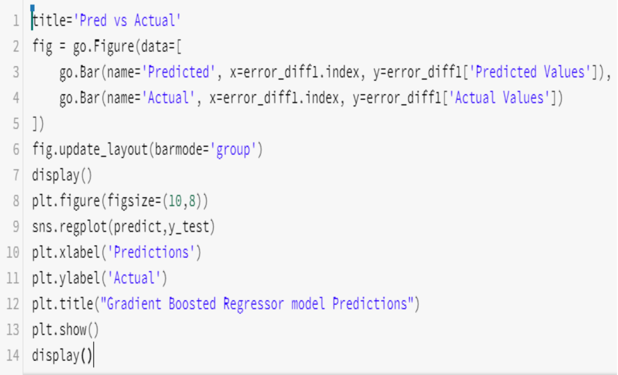
```
#VISUALISATION
print('Rooms with the most number of reviews')
Long=-73.80
Lat=40.80
mapdf1=folium.Map([Lat,Long],zoom_start=10,)

mapdf1_rooms_map=plugins.MarkerCluster().add_to(mapdf1)

for lat,lon,label in zip(df1.latitude,df1.longitude,df1.name):
    folium.Marker(location=[lat,lon],icon=folium.Icon(icon='home')
                  ,popup=label).add_to(mapdf1_rooms_map)
mapdf1.add_child(mapdf1_rooms_map)
mapdf1
```

Source code 2. visualisation



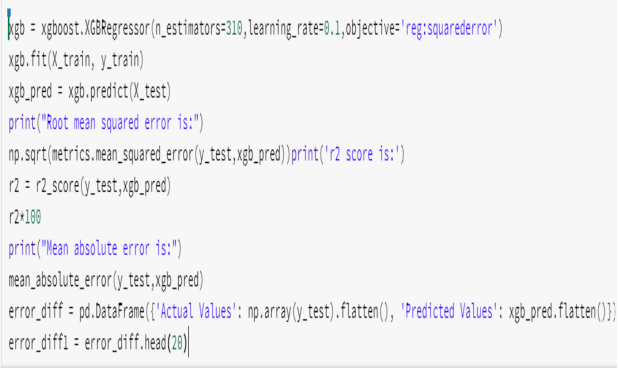
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=101)
GBoost = GradientBoostingRegressor(n_estimators=3000, learning_rate=0.01)
GBoost.fit(X_train,y_train)
predict = GBoost.predict(X_test)
print("Root mean squared error is:")
np.sqrt(metrics.mean_squared_error(y_test,predict))
print('r2 score is:')
r2 = r2_score(y_test,predict)
r2*100
print("Mean absolute error is:")
mean_absolute_error(y_test,predict)
```

Source code 3.Testing and training



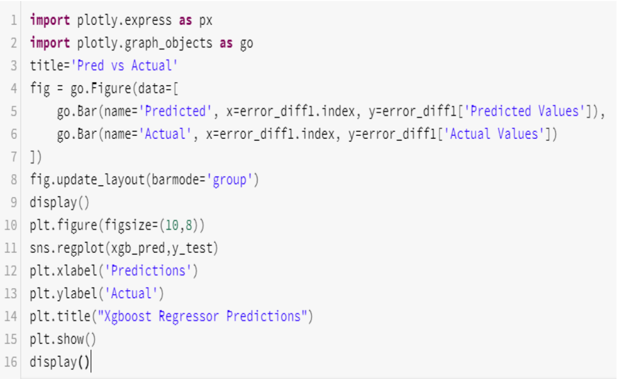
```
title='Pred vs Actual'
fig = go.Figure(data=[
    go.Bar(name='Predicted', x=error_diff1.index, y=error_diff1['Predicted Values']),
    go.Bar(name='Actual', x=error_diff1.index, y=error_diff1['Actual Values'])
])
fig.update_layout(barmode='group')
display()
plt.figure(figsize=(10,8))
sns.regplot(predict,y_test)
plt.xlabel('Predictions')
plt.ylabel('Actual')
plt.title("Gradient Boosted Regressor model Predictions")
plt.show()
display()
```

Source code 4. GBT predictions



```
xgb = xgboost.XGBRegressor(n_estimators=310,learning_rate=0.1,objective='reg:squarederror')
xgb.fit(X_train, y_train)
xgb_pred = xgb.predict(X_test)
print("Root mean squared error is:")
np.sqrt(metrics.mean_squared_error(y_test,xgb_pred))print('r2 score is:')
r2 = r2_score(y_test,xgb_pred)
r2*100
print("Mean absolute error is:")
mean_absolute_error(y_test,xgb_pred)
error_diff = pd.DataFrame({'Actual Values': np.array(y_test).flatten(), 'Predicted Values': xgb_pred.flatten()})
error_diff1 = error_diff.head(20)
```

Source code 5. XGB R2 square



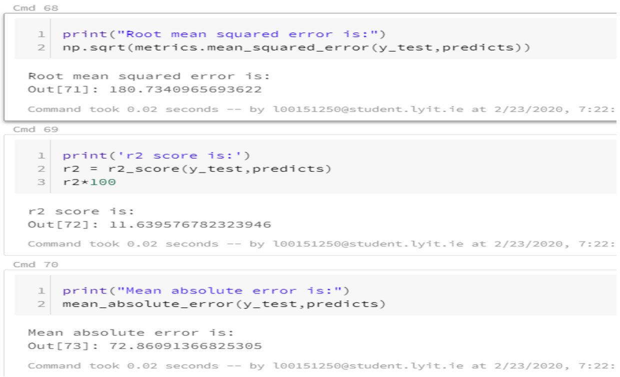
```
import plotly.express as px
import plotly.graph_objects as go
title='Pred vs Actual'
fig = go.Figure(data=[
    go.Bar(name='Predicted', x=error_diff1.index, y=error_diff1['Predicted Values']),
    go.Bar(name='Actual', x=error_diff1.index, y=error_diff1['Actual Values'])
])
fig.update_layout(barmode='group')
display()
plt.figure(figsize=(10,8))
sns.regplot(xgb_pred,y_test)
plt.xlabel('Predictions')
plt.ylabel('Actual')
plt.title("Xgboost Regressor Predictions")
plt.show()
display()
```

Source code 6. XGB predictions

VI. Experimental Results:

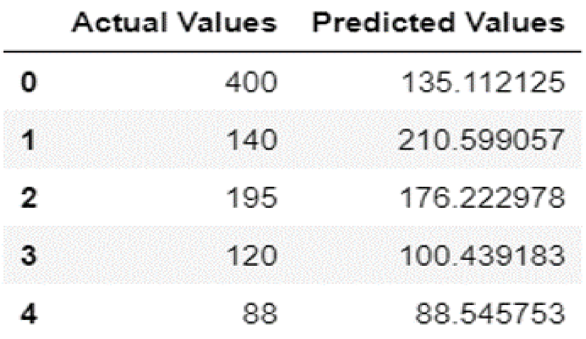
After the data analysis, visualisation and finding the correlations. Three different Machine Learning algorithms have been explained and implemented the training and testing algorithms, R Square and RMSE has been executed and predicted with following results.

To find the accuracy, Root Mean square, r2 Score(Regression score function.), Mean Absolute Error(is a difference measurement between two constant variables) is calculated and the respected model predictions is coded.



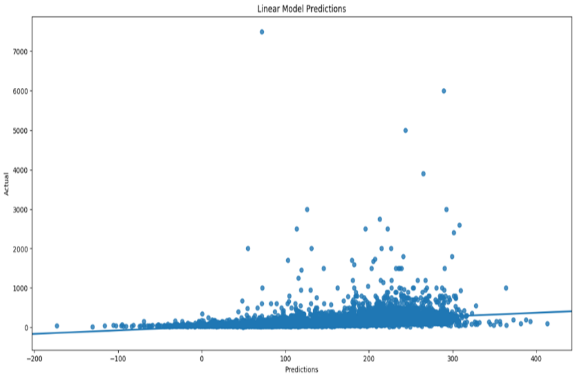
```
Cmd 68
print("Root mean squared error is:")
np.sqrt(metrics.mean_squared_error(y_test,predicts))

Root mean squared error is:
Out[71]: 180.7340965693622
Command took 0.02 seconds -- by l00151250@student.lyit.ie at 2/23/2020, 7:22:
Cmd 69
print('r2 score is:')
r2 = r2_score(y_test,predicts)
r2*100

r2 score is:
Out[72]: 11.639576782323946
Command took 0.02 seconds -- by l00151250@student.lyit.ie at 2/23/2020, 7:22:
Cmd 70
print("Mean absolute error is:")
mean_absolute_error(y_test,predicts)

Mean absolute error is:
Out[73]: 72.86091366825305
Command took 0.02 seconds -- by l00151250@student.lyit.ie at 2/23/2020, 7:22:
```

Result 1. linear regression accuracy

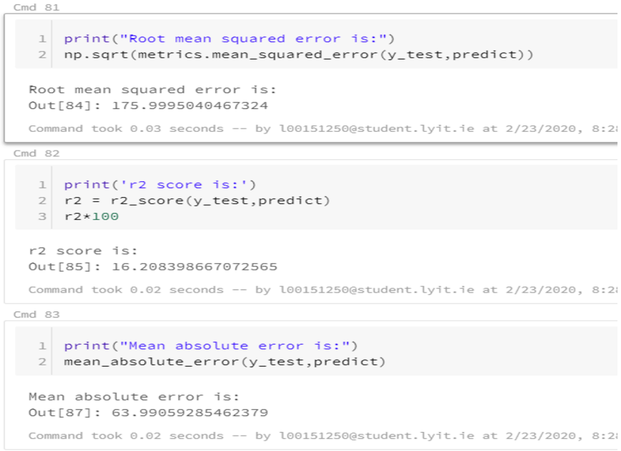


	Actual Values	Predicted Values
0	400	135.112125
1	140	210.599057
2	195	176.222978
3	120	100.439183
4	88	88.545753

Result 2. actual v/s predicted values.

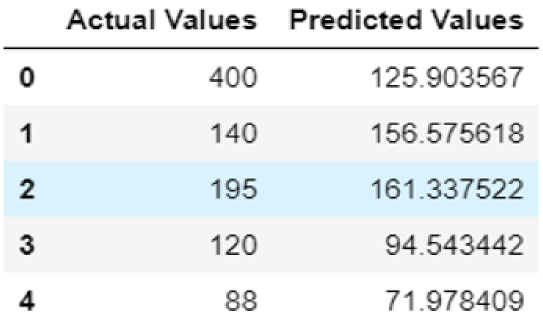


Result 3. linear regression predictions



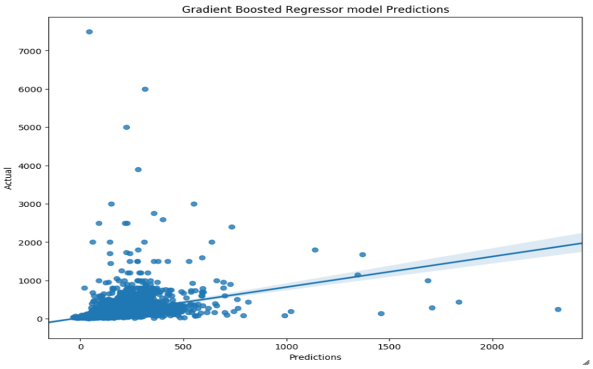
```
Cmd 81
print("Root mean squared error is:")
np.sqrt(metrics.mean_squared_error(y_test,predict))

Root mean squared error is:
Out[84]: 175.9995040467324
Command took 0.03 seconds -- by l00151250@student.lyit.ie at 2/23/2020, 8:2
Cmd 82
print('r2 score is:')
r2 = r2_score(y_test,predict)
r2*100

r2 score is:
Out[85]: 16.208398667072565
Command took 0.02 seconds -- by l00151250@student.lyit.ie at 2/23/2020, 8:2
Cmd 83
print("Mean absolute error is:")
mean_absolute_error(y_test,predict)

Mean absolute error is:
Out[87]: 63.99059285462379
Command took 0.02 seconds -- by l00151250@student.lyit.ie at 2/23/2020, 8:2
```

Result 4. gradient boost regress-or accuracy



	Actual Values	Predicted Values
0	400	125.903567
1	140	156.575618
2	195	161.337522
3	120	94.543442
4	88	71.978409

Result 5.actual v/s predicted values.



Result 6. gradient boost regressor predictions

VII. Conclusion

This paper aims to build the best-performing Airbnb market prediction model on the basis of a number of features, including property specifications, input from hosts and consumers on the listings. in the Regression Analysis, when comparing the r2 score between Linear and gradient Boost Regression , Gradient Boost Regression is having the highest r2Score hence it is good for predicting the results. The goal of predicting the price listings is achieved.

References

Github link of my project: Airbnb GithubYang Li, Quan Pan, Tao Yang, and Lantian Guo. Reasonable price recommendation on airbnb using multi-scale clustering. In 2016 35th Chinese Control Conference (CCC), pages 70387041. IEEE, 2016. Dan Wang and Juan L Nicolau. Price determinants of sharing economy based accommodation rental: A study of listings from 33 cities on airbnb. com. International Journal of Hospitality Management, 62:120131, 2017. Timm Teubner, Florian Hawlitschek, and David Dann. Price determinants on airbnb: How reputation pays o in the sharing economy. Journal of Self-Governance Management Economics, 5(4), 2017. Pouya Rezazadeh Kalehbasti, Liubov Nikolenko, and Hoormazd Rezaei. Airbnb price prediction using machine learning and sentiment analysis. arXiv preprint arXiv:1907.12665, 2019. Laura Lewis. Predicting airbnb prices with machine learning and deep learning, 2019. Kaggle. Airbnb price prediction, 2018..