MIS-637
FINAL PROJECT

PREDICTING PRICES FOR SMART STAY RESERVATION - AIRBNB

PREPARED BY: **ARUN KRISHNASAMY**

STUDENT ID: **10455045**

PROFESSOR: **DANESHMAND MAHMOUD**

CONTENTS

PROBLEM STATEMENT – BUSINESS UNDERSTANDING

- A major issue in planning a holiday trip to other city/country is **finding a cheap accommodation** online and save prices on reservation. **Airbnb** is a peer-to-peer online marketplace for arranging homestay solutions which allows users to rent and book residential properties. This project provides insights to users on what are the principal factors majorly **influences the price** of listings in Airbnb and how to effectively use the insights to their own situation.

PROBLEM STATEMENT – BUSINESS UNDERSTANDING

- This project focus on providing insights to following contexts

I. To provide **insights to customers** who wishes to rent listings from Airbnb on how to choose best listing for cheap price and save money

II. To provide **insights to renters** who post their place on Airbnb for rent on how much renting price to fix for their listings and yearn more profits.

RELEVANT DATA – DATA UNDERSTANDING

- The Dataset details the listing activity of homestays in Washington state's largest metropolitan city "**Seattle**" which is obtained from http://insideairbnb.com/get-the-data.html.
- The following data files are used to determine insights.
- **Calendar -** contains date availability, listing price and ID.
- **Listing -** contains all details of listing such as ID, transit, address, location, availability, reviews.
- **Reviews -** contains user id, reviews and comments.

RELEVANT DATA – DATA UNDERSTANDING

Listing:

number_c	first_revie	last_revie	review_sc	review_sc	review_sc	review_sc	review_sc	rev
207	########	1/2/2016	95	10	10	10	10	
43	########	########	96	10	10	10	10	
20	########	9/3/2015	97	10	10	10	10	
0								
38	########	########	92	9	9	10	10	

Reviews:

listing_id	id	date	reviewer_	reviewer_	comments
7202016	38917982	########	28943674	Bianca	Cute and coz
7202016	39087409	########	32440555	Frank	Kelly has
7202016	39820030	########	37722850	Ian	Very
7202016	40813543	8/2/2015	33671805	George	Close to Seat
7202016	41986501	########	34959538	Ming	Kelly was
7202016	43979139	########	1154501	Barent	Kelly was

Calendar:

listing_id	date	available	price
241032	1/4/2016	t	$85.00
241032	1/5/2016	t	$85.00
241032	1/6/2016	f	
241032	1/7/2016	f	
241032	1/8/2016	f	

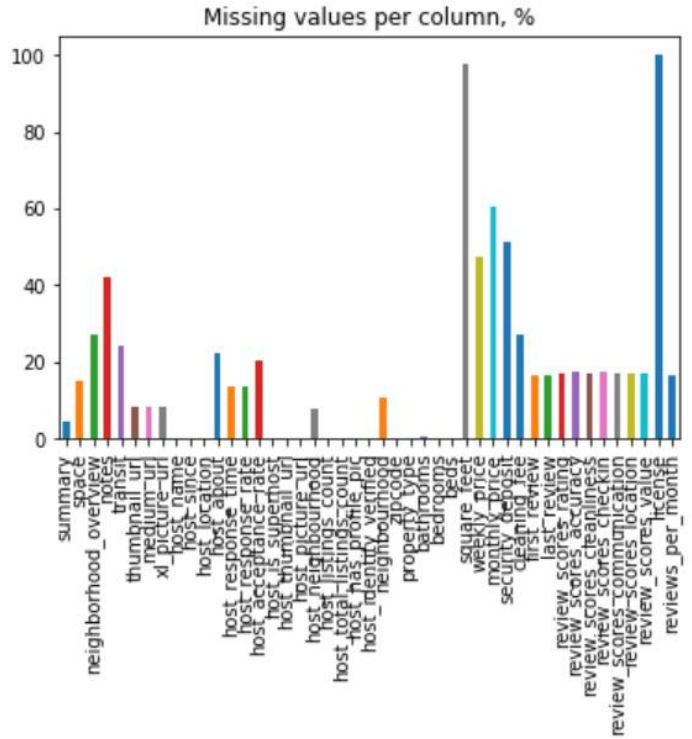
- The current process in airbnb.com determines its listing price based on factors like number of nights, cleaning fee, number of guests, extra guest and VAT-taxes and local taxes as mentioned in their website - "Airbnb price determination".
- This analysis will rather focus on data's like reservation season (time), amenities provided, reviews and location of the listing (based on its neighborhood) and understand its significance on how it will affect the listing price and provide insights to the customers.

DATA PREPROCESSING & DATA QUALITY - DATA PREPARATION

Listing data set analysis:

- Perform missing data analysis on listing dataset to figure out the percentage of missing data
- Based on the missing data analysis on we find out that various fields in the listing dataset like weekly price, Monthly price, square feet, security deposit, license have more than 50% of the data missing in the dataset
- These missing data has least significance in our analysis, hence these datasets are ignored/removed in further processing during data cleaning to avoid overfitting the training model.

```
#find percentage of missing data
listings_missing_df = listings_df.isnull().mean()*100

#filter out columns which have missing values only
listings_columns_with_nan = listings_missing_df[listings_missing_df > 0]

#plot the final results in below graph
listings_columns_with_nan.plot.bar(title='Missing values per column,%')
```



DATA PREPROCESSING & DATA QUALITY - DATA PREPARATION

Calendar data set analysis:

- Perform missing data analysis on calendar dataset to figure out the percentage of missing data
- Based on the missing data analysis on we find out that price data is missing for around 30% of the listing data provided in the dataset

```
[ ]  #get percentage of missing values for each column in dataset
     calendar_missing_df = pd.DataFrame([calendar_df.isnull().mean()*100])

     #plot the results
     calendar_missing_df.plot.bar(title='Missing values per column, %')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fde9fdefd68>
```

DATA PREPROCESSING & DATA QUALITY - DATA PREPARATION

Data Cleaning & pre-processing:

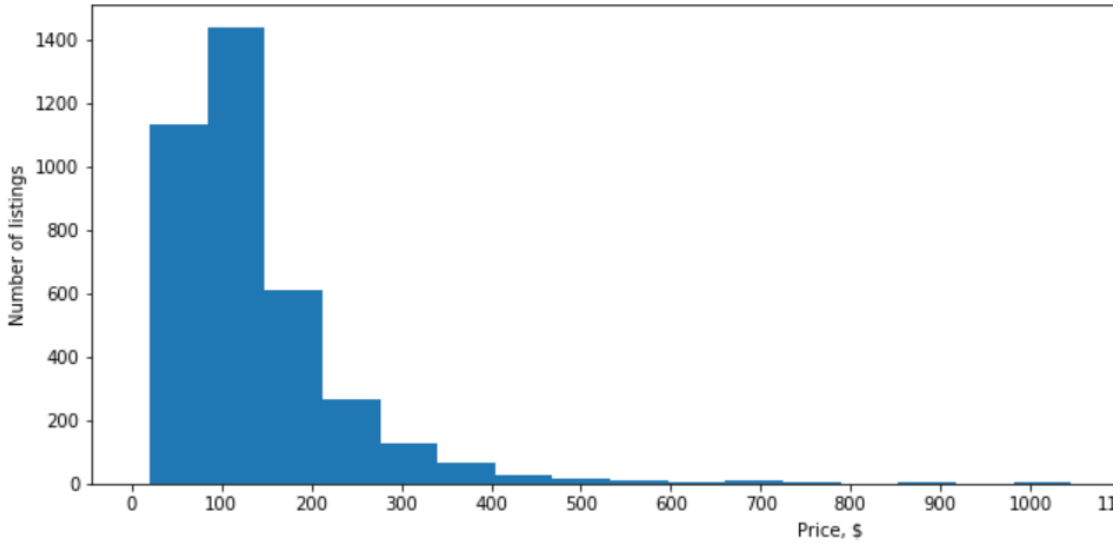
- Various data preprocessing and data cleaning techniques are used, starting with **merging dataset** into single using listing identifier
- And **remove extraneous data's** which are not required
- Convert date and price value into number value.
- Split columns containing list into different related fields
- **Replace missing values** with other values like **mean** value, **mode** value or **dummy values like 0's and 1's**

```
#merge datasets
listings_df = listings_df.rename(index=str, columns={"id": "listing_id"})
df = pd.merge(calendar_df, listings_df, on = 'listing_id')

#drop the irrelevant columns
columns_to_drop = ['available', 'host_id','host_location','host_acceptance_rate','host_neighbourhood',
                'host_total_listings_count', 'weekly_price', 'monthly_price',
                'security_deposit', 'cleaning_fee', 'calendar_updated',
                'listing_url','last_scraped' ,'scrape_id', 'name', 'summary', 'space', 'description',
                'experiences_offered', 'street', 'neighbourhood', 'neighbourhood_cleansed', 'zipcode',
                'neighborhood_overview', 'notes', 'transit', 'thumbnail_url', 'medium_url', 'picture_url',
                'xl_picture_url', 'host_url', 'host_name', 'host_about', 'host_thumbnail_url', 'host_picture_url',
                'city', 'state', 'market', 'smart_location', 'country_code', 'country', 'latitude', 'longitude',
                'is_location_exact', 'square_feet', 'has_availability', 'availability_30',
                'availability_60', 'availability_90', 'availability_365', 'calendar_last_scraped',
               'first_review', 'last_review', 'requires_license', 'license', 'jurisdiction_names', 'price_y',
               'reviews_per_month']
df = df.drop(columns = columns_to_drop)
```

DATA INSIGHTS – INSIGHT ANALYSIS & VISUALIZATION

- Based on the cleaned data we conduct various insight analysis on it to determine various factors directly and indirectly affects the price of the listing in airbnb.com

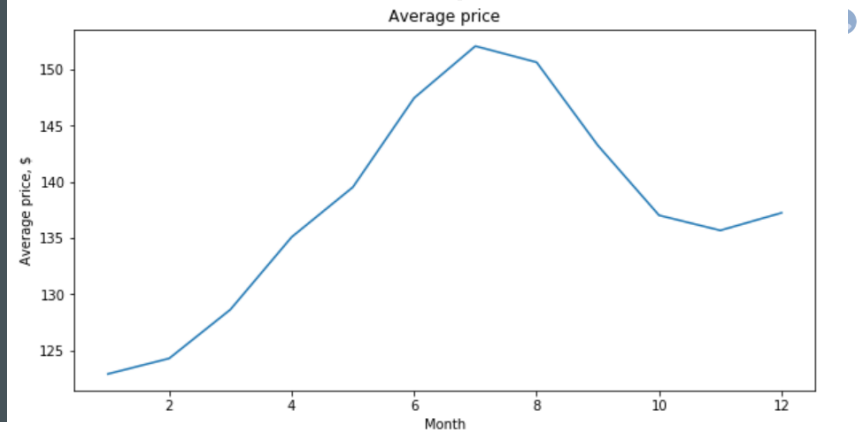
Listing/Price histogram:



DATA INSIGHTS – INSIGHT ANALYSIS & VISUALIZATION

- **Season analysis** is made to figure out which months in the year are busiest and which month in the year are not.
- Comparing season analysis with the **price analysis** chart it is also revealed that the average price of the listing is high in peak season like in the months 6 to 9
- Hence it can be concurred that **listing demand is directly proportional to setting listing price**.

```
#plot
plt.figure(figsize=(10,5))
plt.plot(average_price_by_month)
plt.ylabel('Average price, $')
plt.xlabel('Month')
plt.title('Average price')

plt.savefig('average price for month')

plt.show()
```



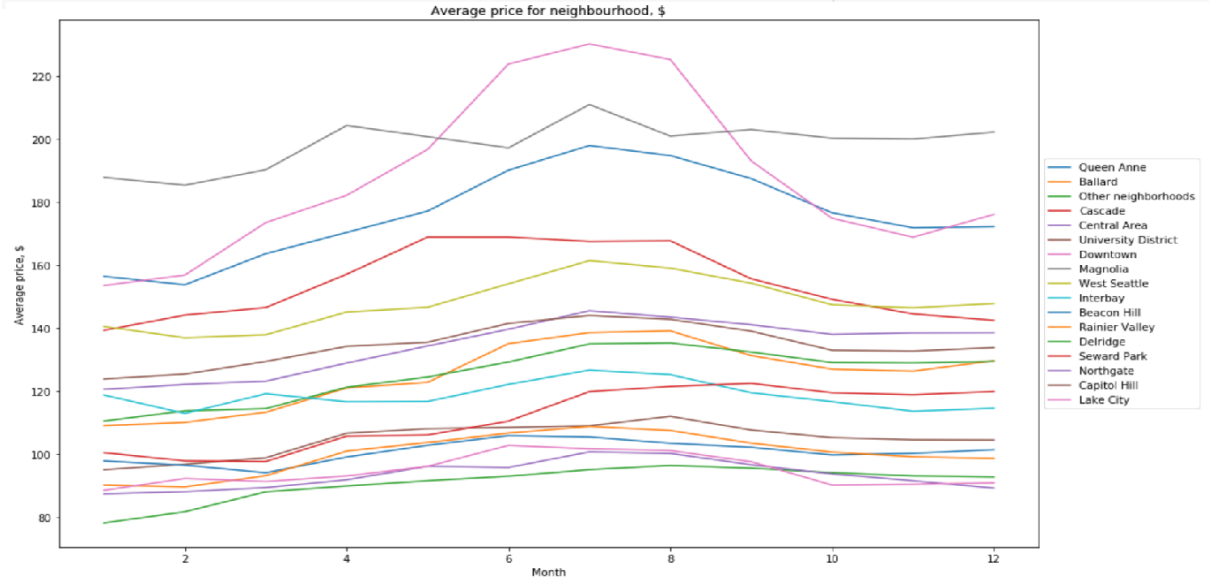

```
#plot
plt.figure(figsize=(10,5))
plt.plot(number_of_listings_by_month)
plt.xticks(np.arange(1, 13, step=1))
plt.ylabel('Number of listings')
plt.xlabel('Month')
plt.title('Number of listings per month, 2016')

plt.savefig('number of available listings.png')

plt.show()
```

DATA INSIGHTS – INSIGHT ANALYSIS & VISUALIZATION

Listing geography analysis:

- Analysis made on determining prices based on the listing location to figure out the cheap priced locations and the costliest one

```
for neighbourhood in neighbourhoods:
    ax.plot(price_by_month_neighbourhood[price_by_month_neighbourhood['neighbourhood_group_cleansed'] == neighbourhood]
            price_by_month_neighbourhood[price_by_month_neighbourhood['neighbourhood_group_cleansed'] == neighbourhood
            label = neighbourhood)

box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width * 0.8, box.height])
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))

plt.ylabel('Average price, $')
plt.xlabel('Month')
plt.title('Average price for neighbourhood, $')

plt.savefig('average price for neighbourhood')

plt.show()
```



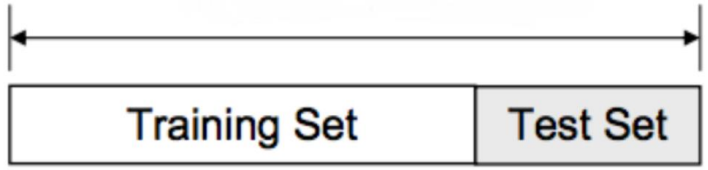
METHODOLOGIES/ALGORITHM'S – MODELLING & EXECUTION

Random Forest regressor:

- We are going to apply machine learning techniques test and train dataset to figure out which feature/field in the dataset highly influence the price value of the listing.
- RF-Random forest technique is used since its typically provide **high accuracy** because unlike simple decision tree algorithm, in random forest algorithm the process of finding root node and splitting feature node happens randomly and it is capable of handling missing values.
- The process involved in RF methods are
  - Use train_test_split() function to prepare train and test data sets
  - Train Random forest model using RandomForestRegressor() function
  - Post that fit the model on training data using fit() function
  - Calculate scores for the model
  - Plot the corresponding graph for the model predicted against its feature importance's along with their inter-tree variability.

METHODOLOGIES/ALGORITHM'S - MODELLING & EXECUTION

Random Forest regressor:



- **prepare train and test data sets**:

The **train_test_split()** function is used to make the split of dataset into two – training and testing dataset, the **value is represented from 0 to 1** which denotes corresponding split percentage. Ideally the training set percentage should be considerably higher than the testing data set. Here we have set the test size as **0.3 ~ 30%.** Once the Test size parameter is entered in the function it automatically sets the training set size (1 – test size), here its (1 – 0.3) = 0.7 (70%).

```
#prepare train and test datasets for modelling
TEST_SIZE = 0.3
RAND_STATE = 0

X = df.drop(columns = 'price')
y = df[['price']]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = TEST_SIZE, random_state=RAND_STATE)
```

METHODOLOGIES/ALGORITHM'S - MODELLING & EXECUTION

Random Forest regressor:

- **Train, fit and calculate score using Random forest:** The model is trained using randomforestregressor() function and the model is fitted into training data using fit function and "Mean Squared error" score and "R square" score is calculated for both training and testing data set

	Training Data	Test Data
Mean Squared Error	186.97	211.25
R Square	0.983	0.981

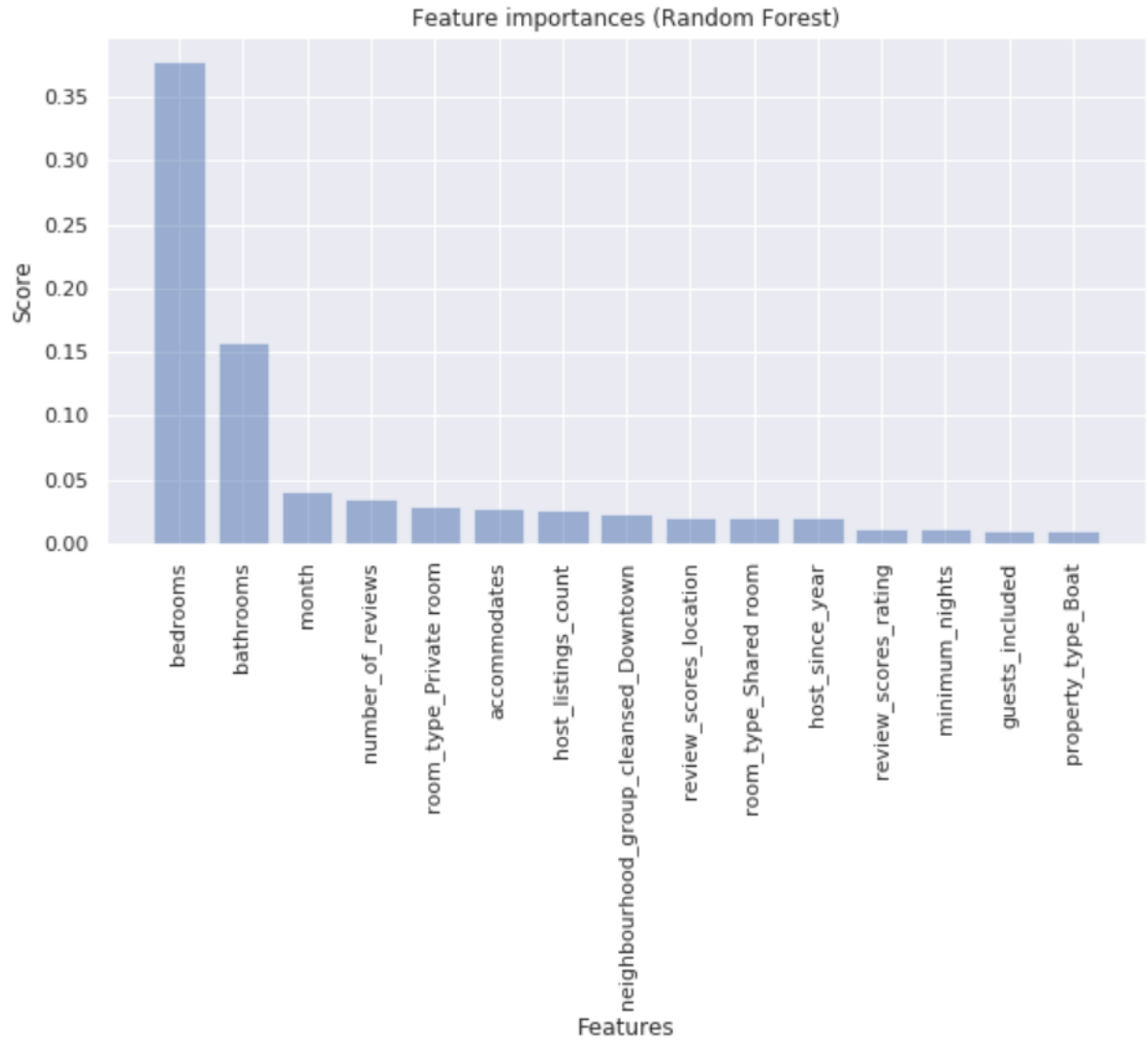
```
#train RF regressor model
forest = RandomForestRegressor(n_estimators=100,
                               criterion='mse',
                               random_state=RAND_STATE,
                               n_jobs=-1)
forest.fit(X_train, y_train.squeeze())

#calculate scores for the model
y_train_preds = forest.predict(X_train)
y_test_preds = forest.predict(X_test)

print('Random Forest MSE train: %.3f, test: %.3f' % (
        mean_squared_error(y_train, y_train_preds),
        mean_squared_error(y_test, y_test_preds)))
print('Random Forest R^2 train: %.3f, test: %.3f' % (
        r2_score(y_train, y_train_preds),
        r2_score(y_test, y_test_preds)))
```

```
Random Forest MSE train: 186.973, test: 211.258
Random Forest R^2 train: 0.983, test: 0.981
```

METHODOLOGIES/ALGORITHM'S - MODELLING & EXECUTION

Random Forest regressor:

- **Random forest feature importance:** The feature importance of the forest is calculated to evaluate the important features in the dataset which highly influence the result score which is directly influence the price of the listing
- Based on this analysis a conclusion can be made on which features directly influences the price

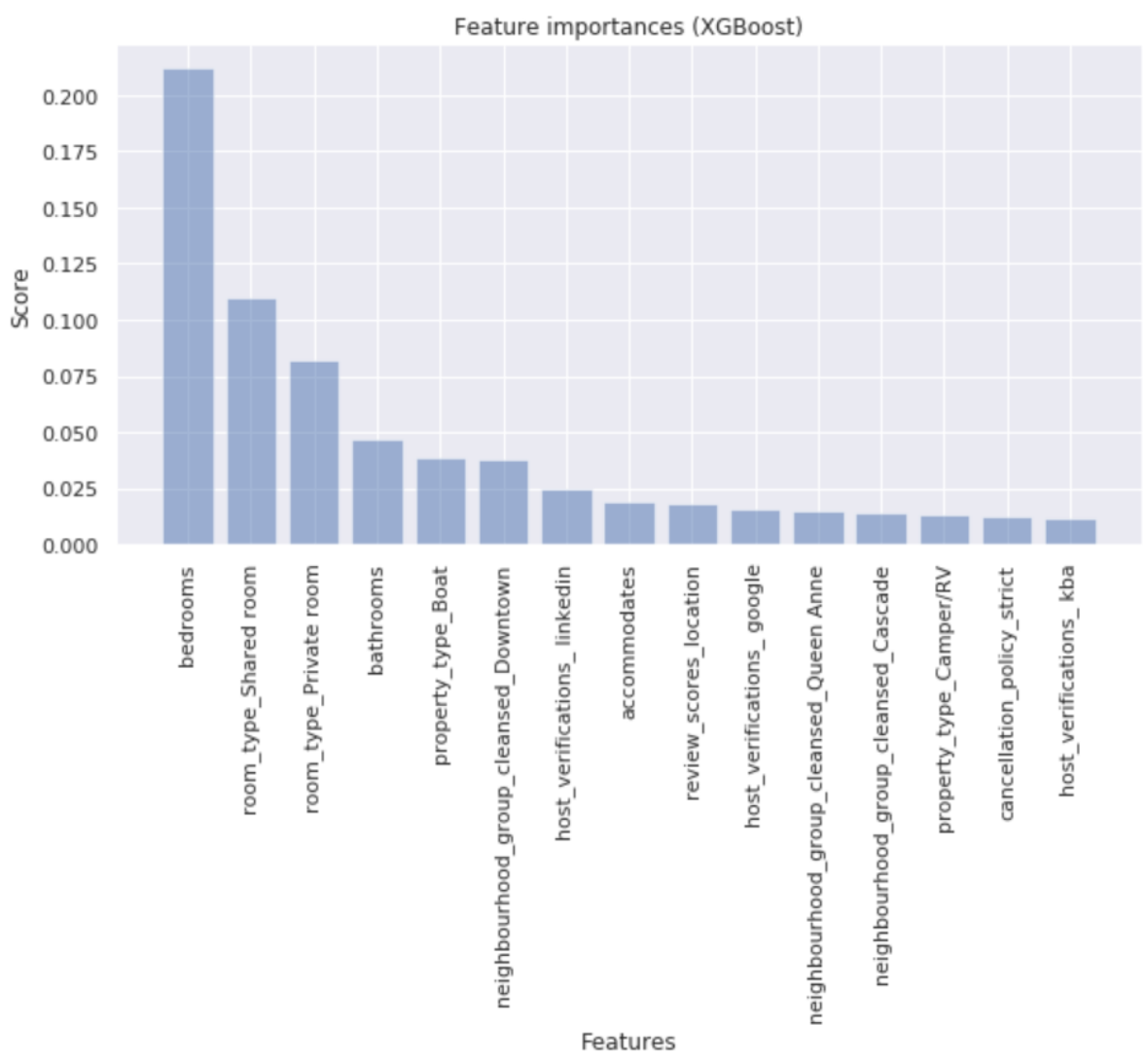


METHODOLOGIES/ALGORITHM'S - MODELLING & EXECUTION

Gradient Boost regressor:

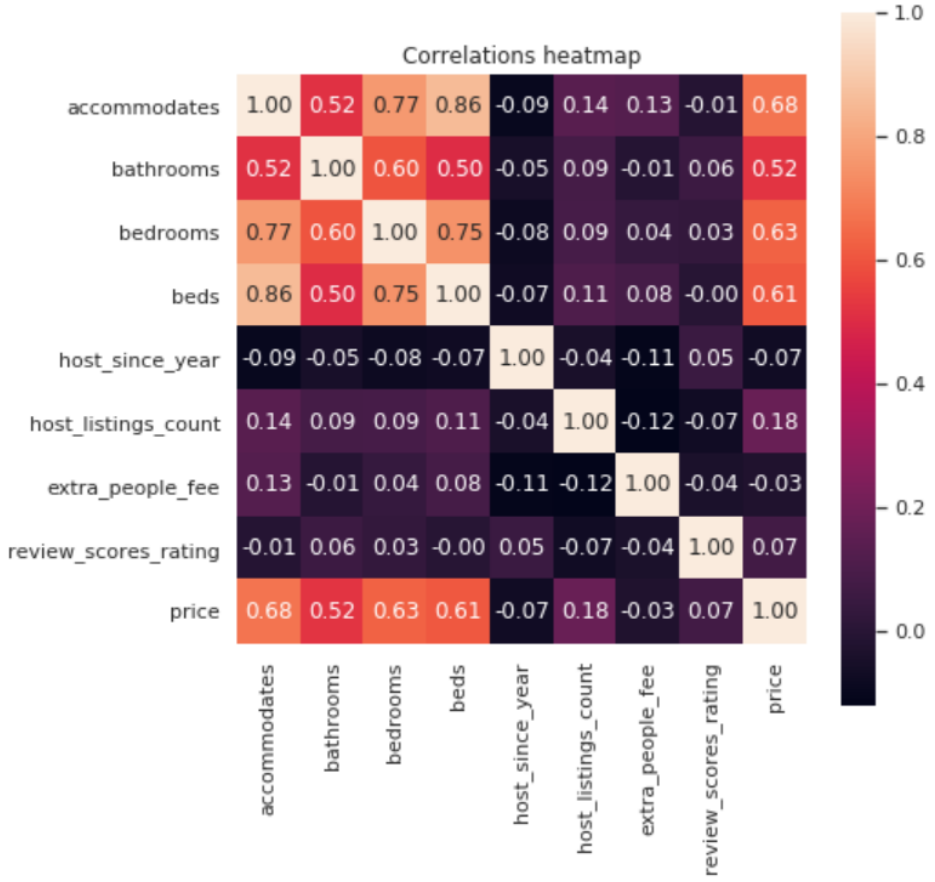
- Similar to random forest regressor method **XG Boost method** is implemented to analyze if the method can provide better results and provide better insights to the users based on the results from both the technique
- This technique if better tuned can provide better results than random forest

	Training Data	Test Data
Mean Squared Error	885.56	890.05
R Square	0.92	0.918



CONCLUSION AND ANALYSIS - EVALUATION & DEPLOYMENT

- While analyzing the results from both models we concur that certain factors have greater influence on the listing price. Features like bedroom, bathroom, room type bathroom type has greater influence on price factors
- From correlation heatmap we can also figure out that factors like reviews, beds provided, review count could also influence the price.
- Also we deduce that factors like host since year, extra people fee plays lesser role in determining the price



CONCLUSION AND ANALYSIS - EVALUATION & DEPLOYMENT

Procured Solutions - Insights To Customers:

- Based on analysis when planning for trip and to book cheap listings the customers should consider the month of travelling (season of travelling) and summer months (6to9) are costliest.
- The analysis provides insights to customers with neighborhood with cheap prices to rent listings
- Based on our machine learning model the customers are suggested to look out for features like number of bedrooms and bathrooms and extra people to determine cheap price.
  - For example. The customer can look for listings with less bedrooms and bathrooms for cheap price

CONCLUSION AND ANALYSIS - EVALUATION & DEPLOYMENT

Procured Solutions - Insights To Renters :

- Based on our analysis the renters can fix high prices in months in peak seasons to make use of high demand and earn more money
- Also, if the renters are provided with suggestion to tweak the prices based on their listing neighborhood to get maximum benefit
- Our machine learning insights suggests that if the listing has more bedrooms, bathrooms, and if it can accommodate more people in the house, then the listing can be posted with high markup prices in the website
- Also, renters should work hard to earn more reviews to increase their credibility and earn more money

REFERENCES

Data Source:

- http://insideairbnb.com/get-the-data.html

Airbnb price determination factors:

- https://www.airbnb.com/help/article/125/how-is-the-price-determined-for-my-reservation?locale=en

Development Reference:

- https://en.wikipedia.org/wiki/Random_forest
- https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6
- https://xgboost.readthedocs.io/en/latest/python/python_api.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

THANK YOU