Business understanding

Determine Business Objectives

Background

With the help of Airbnb, people may let out their houses, flats, or even individual rooms to tourists. Offering an alternate kind of lodging for visitors, business travelers, and locals alike, Airbnb has revolutionized the conventional hotel industry with more than 7 million listings globally.

But given the volume of listings on the platform, it can be challenging for hosts to stand out and draw visitors to their properties. Data mining is then used in this situation. Hosts can learn a lot about what makes their listing appealing to potential guests and how to manage their home to increase their profit by examining the data accessible on Airbnb's platform.

Business Objectives

The major objective of this project is to employ data mining techniques to help hosts boost their profit by giving them useful information about the success of their listings. This will be accomplished by examining the listing dataset, which includes all of the website-visible information about a listing, and the reviews dataset, which includes information on customer reviews for each listing.

By this approach, we hope to provide concrete responses to questions like: Can we forecast the potential profit from each listing based on its features? What are the main characteristics that attract guests to a listing? How can hosts make their listing more appealing to travelers so they can make more money?

The target audience for this project is Airbnb hosts who are trying to find ways to make more money from their listings. We want to assist hosts in achieving their business objectives and prospering in the highly competitive Airbnb market by offering them valuable advice and practical suggestions.

Business success criteria

In this project, the main goal is to increase the profit of Airbnb hosts based on information about their listings. To achieve this, we have a listing dataset that contains all the information that the customer can see on the website about that listing, and a reviews dataset that contains date and reviews_id and listing_id.

The specific questions we want to answer with the data are to see if we can predict the profit each listing could make based on the listing features. The target audience for the project is the hosts who are looking to optimize their profits on Airbnb.

Measuring the success of the project is not just about an increase in profit but also giving a satisfied profit for every listing based on their features, which will lead to activate the market equally. To ensure the data used for the project is accurate and relevant, we have performed data understanding and filtering to make the dataset more accurate.

Data privacy concerns are also a major consideration, and we will ensure that the data is handled securely by following industry-standard data security and privacy policies.

For analyzing the data and deriving insights, we will use various tools and techniques such as Python, Colab GPU and CPU, Pandas, Sklearn.preprocessing, NumPy, Matplotlib.pyplot, and itertools.

The timeline for the project is till the end of the semester, and we will ensure that it stays on track and delivers results on time by following a well-defined project plan (**just joking**)

Communication of the results of the project to stakeholders is also important, and we plan to publish the project as a research paper that can be validated for better reasons. We will also seek feedback and input from stakeholders to incorporate into future iterations of the project.

In summary, we aim to create a valuable tool that can provide advanced studies to the market and help to improve the supply and demand of the Airbnb market, ultimately leading to better profits for hosts.

Assess Situation

Inventory of Resources

we have a good amount of data to work with, consisting of 256,864 listings with related data about reviews. We also have calender data, which will help us to determine the profit of each listing by crossing the listing data with calender data based on listing ID.

Requirements, Assumptions, and Constraints

However, there are some constraints and limitations to consider. The calender data only contains dates of one year and for Paris city only, limiting us to only 30,000 rows. This could potentially affect the accuracy of our predictions if we do not take these limitations into account.

We have made some assumptions regarding the project. For example, we assume that having more calender data, including other cities, could potentially increase the accuracy of our predictions. Additionally, we assume that we could predict the yearly profit of each listing based on its features if we had enough calender datasets.

Risks and Contingencies

One potential risk we need to consider is predicting very low profits for a listing, causing the host to not list their property on the website. This could potentially decrease the market supply and accuracy of future trained models.

Determine Data Mining Goals

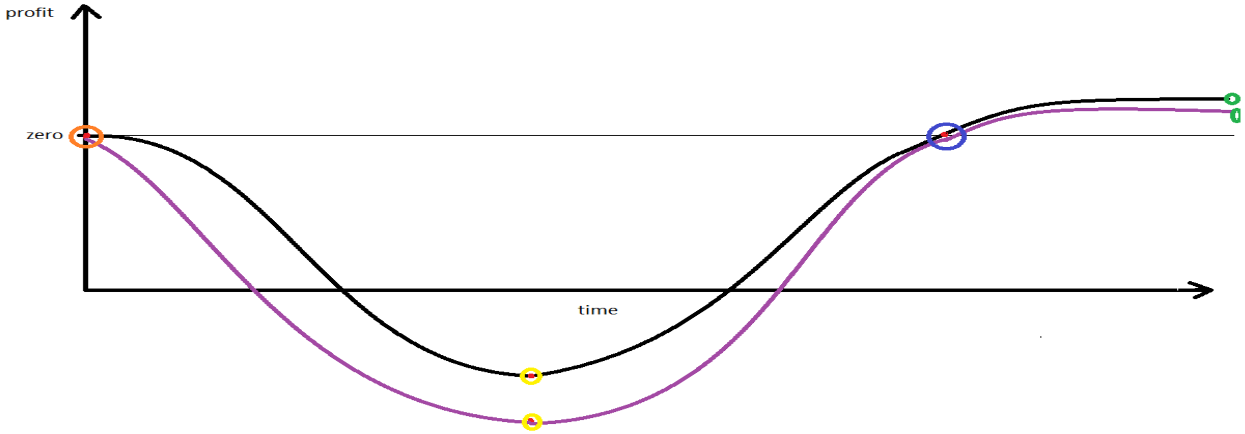
The objective of this project is to predict the profit of each listing based on its features and amenities.

Project Plan

Fact 1: Most of hosts prefer to put high price by thinking that with that they will get the maximum profits.
Fact 2: listing supply is bounded by time and constructions
Hypothesis1: By predicting what price they should put they will get higher profits.
Hypothesis2: profit can be calculated by (number_of_days_full x price)
Hypothesis3: Airbnb users are tourists or travelers

1. **Because The listing prices is in it’s local currency, first we should convert all currencies to dolla.**
2. **Get the final dataset size and that we can get by appling the following steps:**
   a. **Get all listing_ids which are common between the Listing dataset and calendar dataset (lets call them Paris_Listing_ids).**
   b. **Make sure to exclude all the listing_ids that are located outside Paris.**

3. Count all nights that full for every listing_id of the Paris_Listing_ids
4. Filter listing_ids dataset by excluding all ids which are not inside Paris_Listing_ids
5. Assigning a new column to listing_ids with profit column which will be calculated based on Hypothesis2.
6. Preprocessing the dataset with adding new columns can be produced by apply one hot encoding amenities and property_type and room_type
7.Apply PCA to reduce the number of columns
8.Apply the right model (train, test, validate)
9.Evaluate



By predicting the profit based on the features including prices we will see that, if we changed the price the profit will be we could get higher or lower profit, so we will evaluate our project based on determining the profit based on the price factor. Additional to that as we see in the diagram in the yellow section when hosts see the predicted profit if he emprove his amenities, here we can see that will make the change and the increasment in the profits when the host do the improvements.

To conclude: by making predicted simulation to the profit based on amenities and price, we encourage the hosts to emprove his amenities, and that what will make the real change in the listings community.

Data understanding:

Data Collection:

The data sources used in this project consist of three datasets: Listing.csv, Reviews.csv, and Paris_calendar.csv. These datasets were obtained from kaggle.com, a popular platform for machine learning and data science projects.

The first two datasets (Listing.csv and Reviews.csv) were provided by the data mining course. The Listing.csv dataset contains information about 256,864 listings, including features such as the number of bedrooms and bathrooms, the property type, and the location. The Reviews.csv dataset contains data about the reviews left by customers for each listing.

The third dataset, Paris_calendar.csv, was obtained online. It contains information about the availability and pricing of Paris listings for the year 2021. However, the dataset is limited to Paris city only, and contains information for one year only, which might limit the scope of the project.

In addition to the three datasets, we manually retrieved data about the approximate currency exchange rate between the local currencies of each city and the US dollar. This data will be used to help standardize the profit estimates for each listing, as the datasets use different currencies.

There may be limitations and biases in the data sources used. For example, the Reviews.csv dataset does not contain any information about the reviews themselves, and the Paris_calendar.csv dataset only contains information about Paris listings for one year. These limitations may affect the accuracy of the project's predictions.

Data Description:

The dataset used in this project consists of three separate datasets: Listings.csv, Reviews.csv, and Paris_calendar.csv.

Listings.csv contains 279,712 records and 33 fields. The fields in this dataset include: Listing ID, Listing Name, Host ID, Date the Host joined Airbnb, Location where the Host is based, Estimate of how long the Host takes to respond, Percentage of times the Host responds, Percentage of times the Host accepts a booking request, Binary field to determine if the Host is a Superhost, Total listings the Host has in Airbnb, Binary field to determine if the Host has a profile picture, Binary field to determine if the Host has a verified identity, Neighborhood the Listing is in, District the Listing is in, City the Listing is in, Listing's latitude, Listing's longitude, Type of property for the Listing, Type of room type in Airbnb for the Listing, Guests the Listing accommodates, Bedrooms in the Listing, Amenities the Listing includes, Listing price (in each country's currency), Minimum nights per booking, Maximum nights per booking, Listing's overall rating (out of 100), Listing's accuracy score based on what's promoted in Airbnb (out of 10), Listing's cleanliness score (out of 10), Listing's check-in experience score (out of 10), Listing's communication with the Host score (out of 10), Listing's location score within the city (out of 10), Listing's value score relative to its price (out of 10), and Binary field to determine if the Listing can be booked instantly.

Reviews.csv contains 5,373,143 records and 4 fields. The fields in this dataset include Listing ID, Review ID, Review date, and Reviewer ID.

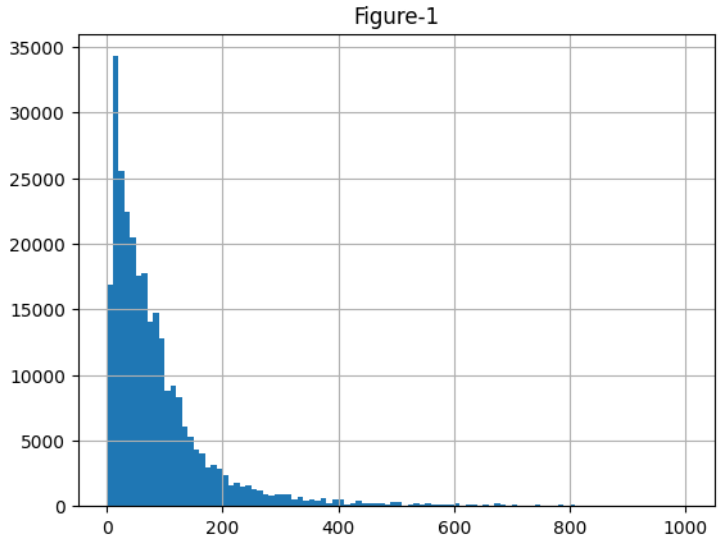
Paris_calendar.csv contains 23,610,091 records and 7 fields. The fields in this dataset include Listing ID, Date, Available, Price, Adjusted price, Minimum nights, and Maximum nights.

The data types and formats for each field are as follows:
Listings.csv columns types: integer, object, integer, object, object, object, float, float, object, float, object, object, object, object, object, float, float, object, object, integer, float, object, float, integer, integer, float, float, float, float, float, float, object.
Reviews.csv columns types: integer, integer, object, integer.
Paris_calendar-2021.csv columns types: integer, object, object, object, object, float, float.
One limitation of the data is that some fields contain missing or incomplete data, such as the 'Host_since' column in Listings.csv, which has NaN values that were replaced with random dates.

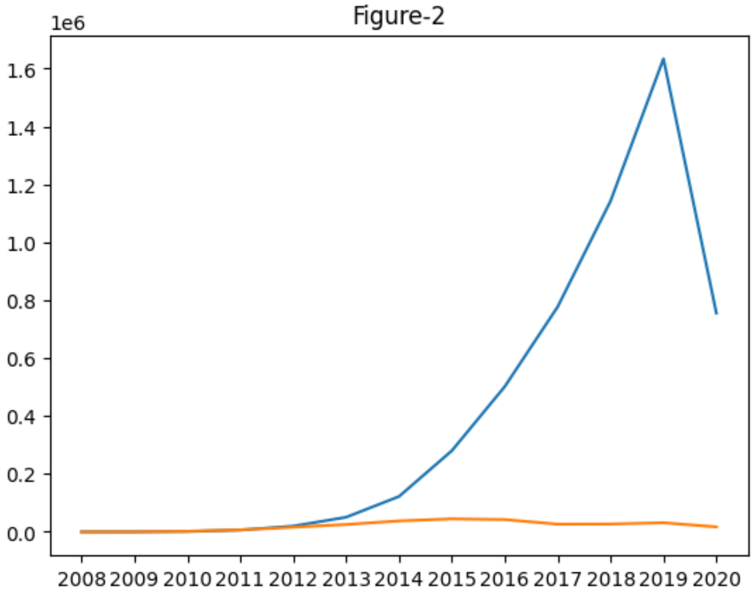
In addition, there are some outliers in the data, such as very high prices, that could potentially distort the distribution histograms. These issues will be addressed in the data preprocessing section of the project.

Data Exploration:

The data suggests that 75% of the prices of the listings are near to $112 while some listings cost $122,542. This raises the question of whether to choose the high price or the high demand, which brings us back to **Fact 1**.
shows us that most of the prices are between $0 and $200, which suggests that people are serious about prices to gain high demand.

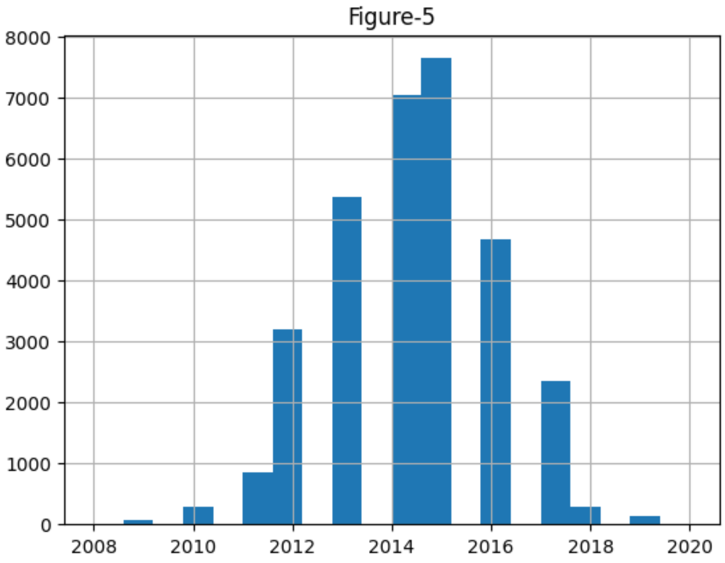


One of the trends worth mentioning is that the number of reviews has decreased dramatically in 2021, which is due to COVID-19 and quarantine protocols. **Figure-2** clearly shows this trend, which supports Hypothesis3, indicating that more than half of the users are travelers.

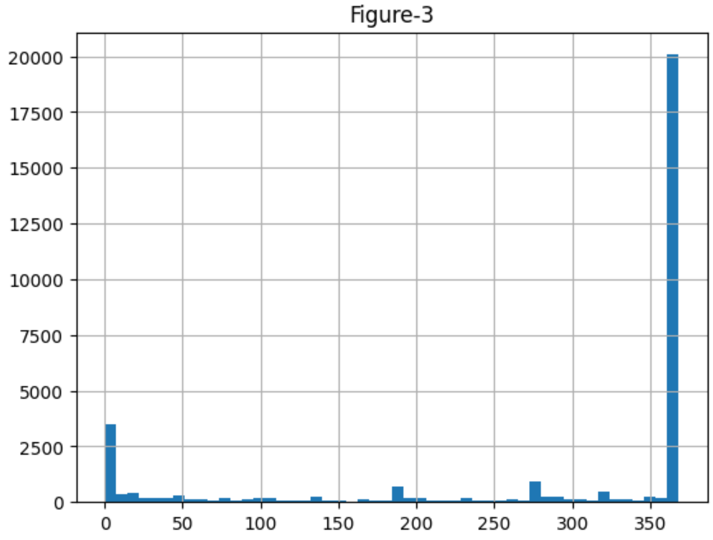


Issues or concerns that may impact the analysis include the fact that in the Paris-calendar-2021.csv, the prices are not different from the Listings.csv prices for the common listings. This could be a mistake from one of the sides because the Listings.csv price should be in euro, whereas in the Paris-calendar-2021.csv, the prices are mentioned in dollars.

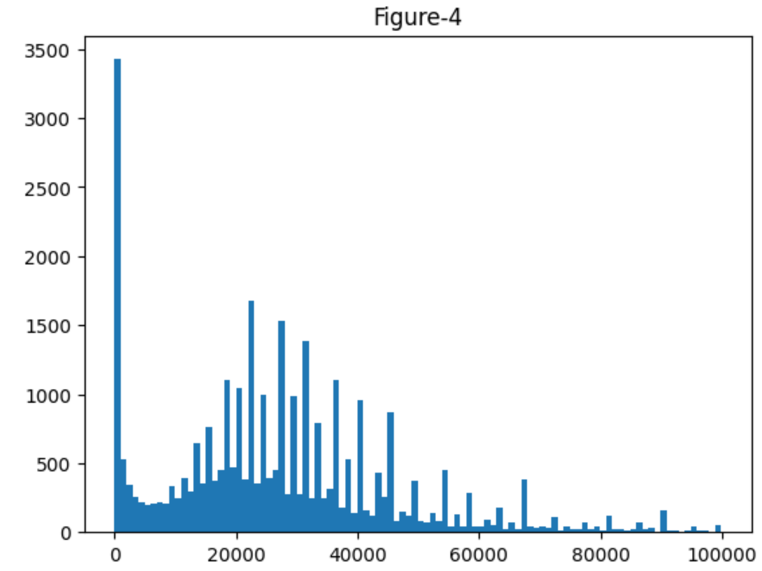
In **Firgure-5** It is worth mentioning that, that the supply is decreasing after 2015 which support **Fact 2**, so we should keep in constraints, the demand to not exceed specific point which will lead to overflow.



Data Quality:
To assess the quality of the data, we analyzed 31942 listings after filtering and joining. **Figure-3** displays the distribution of the number of days each listing is reserved for.



We were fortunate to find that the dataset contains over 20000 listings reserved for more than 350 days. However, during our analysis, we identified a data quality issue that needs to be addressed. **Figure-4** shows the distribution of profits on the listings, and we observed that over 3000 listings had zero profit. This can lead to inaccurate modeling and analysis, as it may indicate fake data. We plan to address this issue by removing listings with zero profit after we have tried modeling and validating the data.



Data Preperation:

Data Cleaning:

We filled in the missing values in the 'host_since' column of the Listings dataset with a random date ('2014-07-31') for less than 200 rows that had NaN values.

Data Reduction:

We removed several columns from the Listings.csv dataset that had no logical correlation with the target variable (profit) or the business goals, including 'host_id', 'district', 'host_location', 'host_response_time', 'host_response_rate', 'host_acceptance_rate', 'host_has_profile_pic', 'host_identity_verified', and 'neighbourhood'. We did not use the Reviews.csv dataset as it did not contain the required information.

Data Transformation, Integration, and Sampling:

We transformed the Paris-Calendar.csv dataset into a dictionary containing the number of days each Paris listing was reserved, and then concatenated this dictionary with the Listings dictionary to create new columns called 'nights_reserved' and 'profit'. We converted the 'host_since' column from the format XXXX:XX:X to the format XXXX, which allowed us to use it as a feature in the final dataset.

We used ordinal encoder to convert string-typed columns like 'host_is_superhost' and 'instant_bookable' into [0 or 1] instead of [f or t]. We also applied One Hot Encoding to the 'property_type' and 'room_type' columns, which increased the number of columns by 52 and 3 respectively.

To encode the amenities column, which was saved as a JSON string, we split the string into parts and used regular expressions to remove unnecessary characters. Then, we encoded amenities as dummy variables, which resulted in a total of 559 columns.

Finally, we dropped the 'nights_reserved' column as it had a negative effect on the accuracy of the model.

What is one Hot encoding?

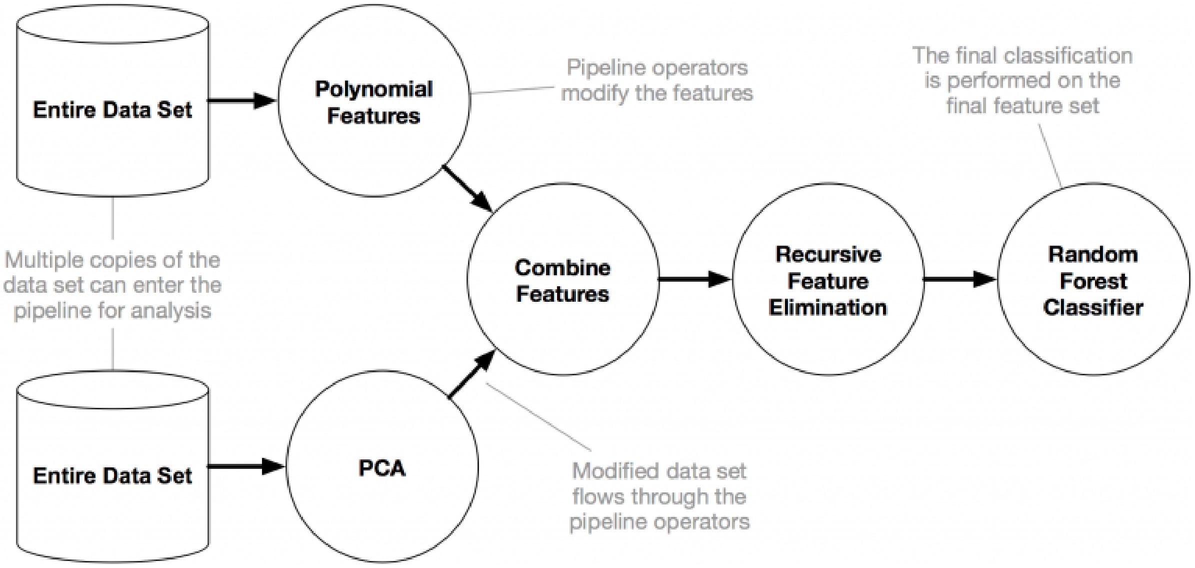
Categorical data can be transformed into numerical data via One Hot Encoding, which can then be employed in machine learning models. With this method, every distinct category in a categorical variable is represented as a binary column, with each column denoting whether the corresponding category was present in the original variable or not. Consider a categorical variable called "color" that has the following three categories: "red," "green," and "blue." Three binary columns, "color red," "color green," and "color blue," will be used to represent the variable "color" after One Hot Encoding; the values in each column will depend on whether the relevant category was included in the original variable or not.

Ordinal encoding:

By giving each category a different number, ordinal encoding preserves the order of the categories while converting categorical data to numerical data. The categories are given a numerical value in ordinal encoding based on their rank or order. For instance, "Small" could be given a value of 1, "Medium" a value of 2, and "Large" a value of 3 in a dataset of garment sizes. When the categories are naturally arranged or hierarchical, as in the case of clothing sizes or educational levels, ordinal encoding is advantageous.

Modeling:

What is Tpot? Quick review


Entire Data Set
Polynomial Features
Pipeline operators modify the features
The final classification is performed on the final feature set
Multiple copies of the data set can enter the pipeline for analysis
Combine Features
Recursive Feature Elimination
Random Forest Classifier
Entire Data Set
PCA
Modified data set flows through the pipeline operators

Tree-based pipeline from TPOT: credits: http://automl.info/tpot/

TPOT, short for Tree-based Pipeline Optimization Tool, is an automated machine learning tool developed by the Data Science and Automation team at DataRobot. It uses genetic programming to optimize a pipeline of machine learning models, feature preprocessors, and data transformations to find the best model for a given dataset.

With TPOT, users only need to provide their dataset and specify the target variable, and TPOT takes care of the rest. It evaluates a wide range of algorithms and preprocessors, including feature selection, dimensionality reduction, and scaling, to identify the most effective combination for the given dataset.

TPOT uses a genetic algorithm to search through the space of possible pipelines. It starts with an initial population of randomly generated pipelines and evolves them through successive generations, with each generation consisting of a new population of pipelines generated through genetic operations such as mutation, crossover, and selection. TPOT uses a fitness function to evaluate each pipeline, which is usually the cross-validated accuracy score of the pipeline on the training data.

One of the benefits of TPOT is that it can handle a wide range of data types, including numerical, categorical, and text data. It can also handle missing values and automatically preprocesses the data before training the model.

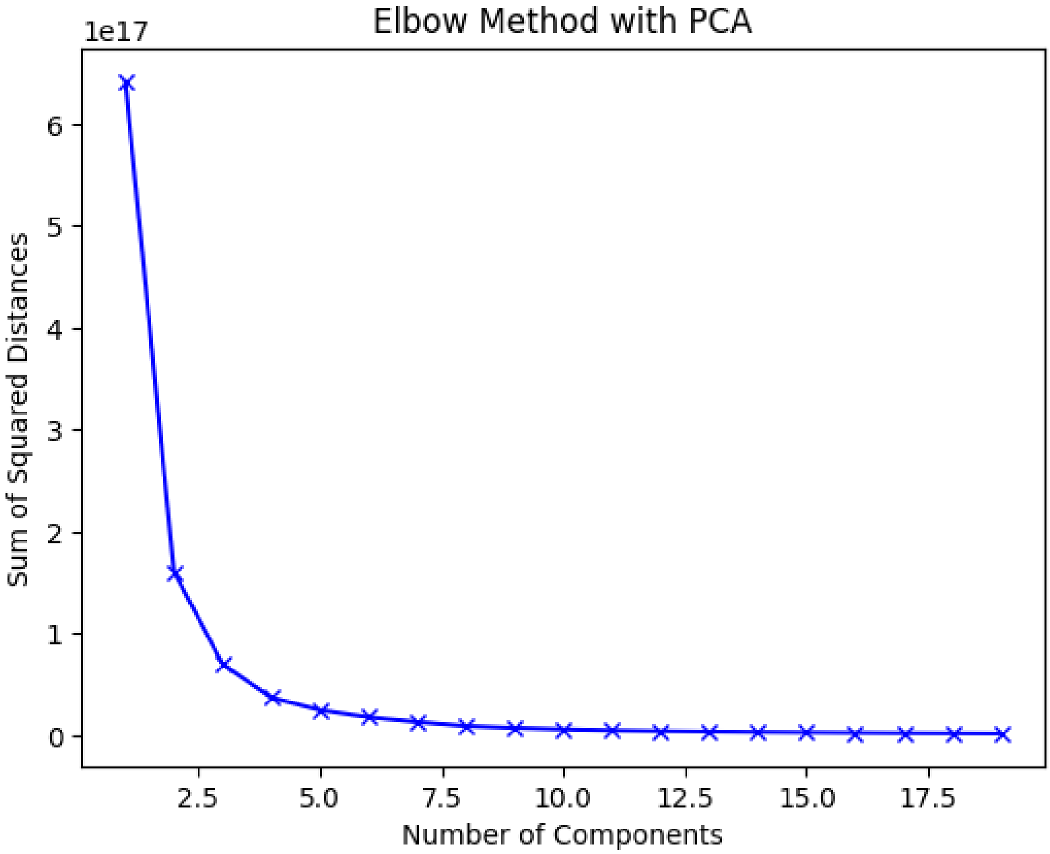
In addition to its automation capabilities, TPOT also allows for manual intervention in the pipeline optimization process. Users can specify the range of algorithms and preprocessors to be considered in the optimization process and can also include custom transformers and models.

To use TPOT, you will need to install it via pip or conda, and then import it into your Python script or notebook. You can then create an instance of the TPOT class and call its fit method to run the pipeline optimization process. Once the optimization process is complete, you can call the TPOT instance's score method to evaluate the best pipeline on a held-out test set.

In conclusion, TPOT is a powerful tool for automating the machine learning pipeline optimization process. Its ability to handle a wide range of data types and missing values, along with its flexibility to allow for manual intervention, make it a valuable tool for data scientists and machine learning practitioners. With TPOT, you can save time and effort in the pipeline optimization process and focus on other aspects of your machine learning project.

To ensure that we used the best modeling technique with optimal parameters, we employed TPOT.

Before applying TPOT, we split our dataset into training and testing sets, with a ratio of 80:20. To handle missing values in the dataset, we applied the SimpleImputer technique to fill the null values with the most mean values in the dataset.



We also applied PCA on the data to reduce the number of columns and avoid overfitting. To determine the optimal number of components for PCA, we utilized the Elbow Method with PCA. The Elbow Method is a technique used to identify the optimal number of components by plotting the explained variance against the number of components and identifying the "elbow point" on the graph.

After determining the optimal number of components for PCA, However, fitting the data inside the TPOT can take a long time, so we utilized the max_time_mins parameter = 60 to specify the maximum time for the model to run. Usually it will take double the time without applying PCA on the data

Finally, the TPOT algorithm

```
TPOTRegressor(generations=5, max_time_mins=60, population_size=20, verbosity=2)
```

provided us with the best model with an R squared score, which we can use for predictive analysis. (I did not use PCA before fitting inside TPOT).

The best model is:

RandomForestRegressor(bootstrap=False, max_features=0.75, min_samples_leaf=16, min_samples_split=19, n_estimators=100)

Evaluation:

In the evaluation step, we analyzed the performance of the model built using TPOT to predict profit for a given company. We calculated the mean absolute error between the y_test and y_pred data, which was found to be 2331.72. We also applied PCA to reduce the number of components and avoid overfitting, which slightly increased the mean absolute error but did not exceed 2600.

To determine the optimal number of components for PCA, we utilized the Elbow Method and selected 20 components, which resulted in an R squared score of 0.675. We also filtered the data to include only those with profits between $10k and $40k to avoid noisy profits.

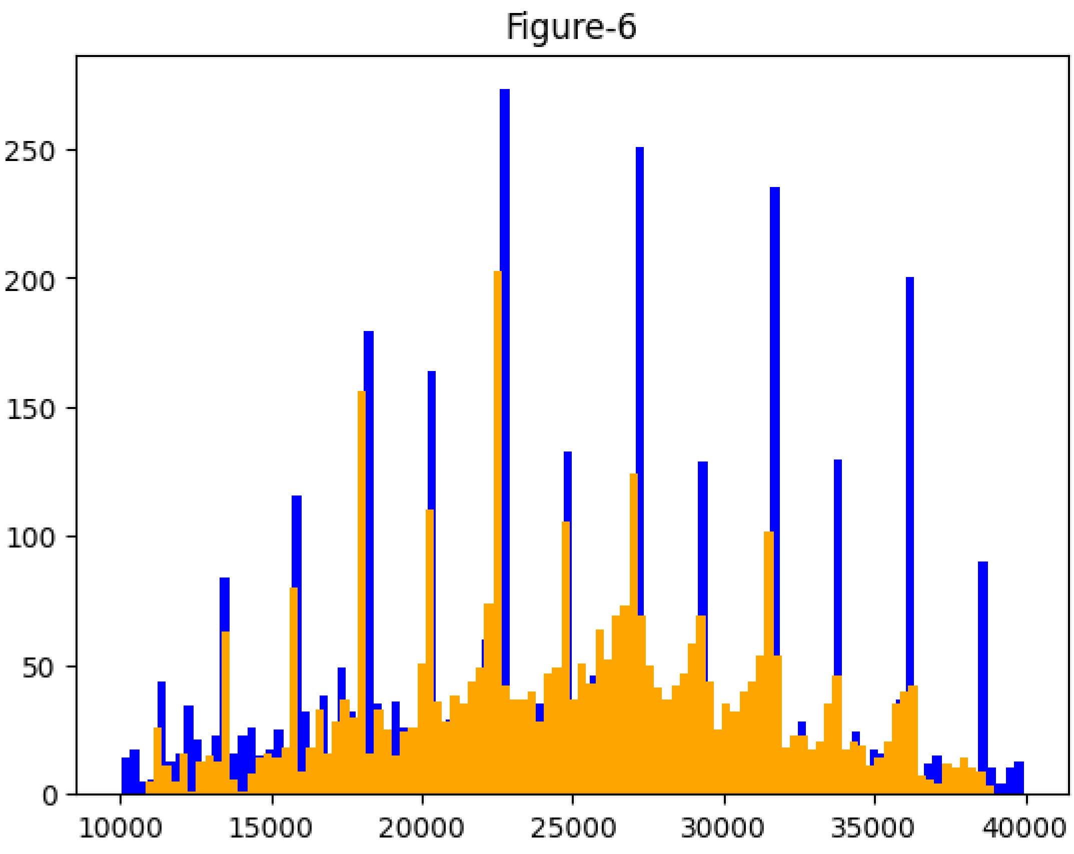
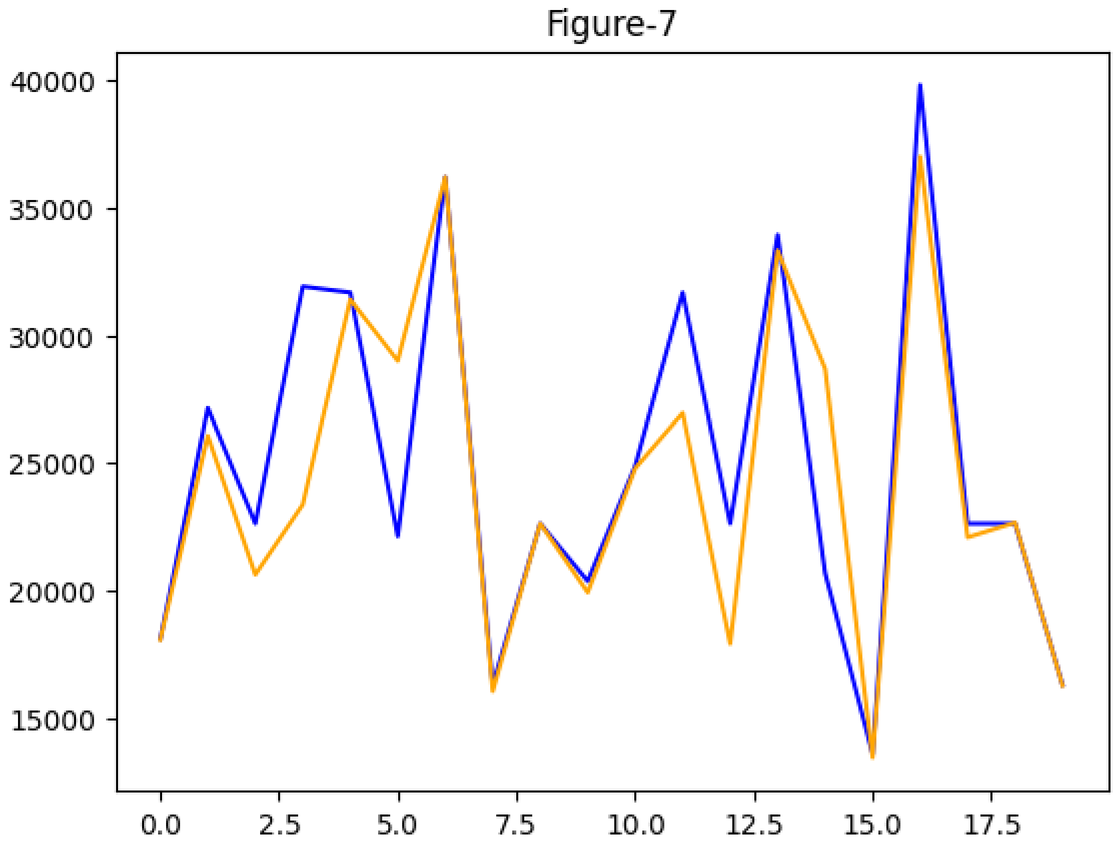


Figure-6 shows that the distribution of the predicted profit (Orange) and the true profit (Blue) are highly correlated, indicating the correctness of the predictions. Overall, the model built using RandomForestRegressor showed promising results in predicting the profit for a given company.

This is an example of the predicted and true profit values:

True	23535	17732	24893	16764	19009	17446	11606	18104	28061	22630
Pred	23238	23412	29194	26932	18749	28307	26485	18009	27829	22444

In Figure-7 we took 20 random samples and got the true profit (blue) and predicted profit (orange).



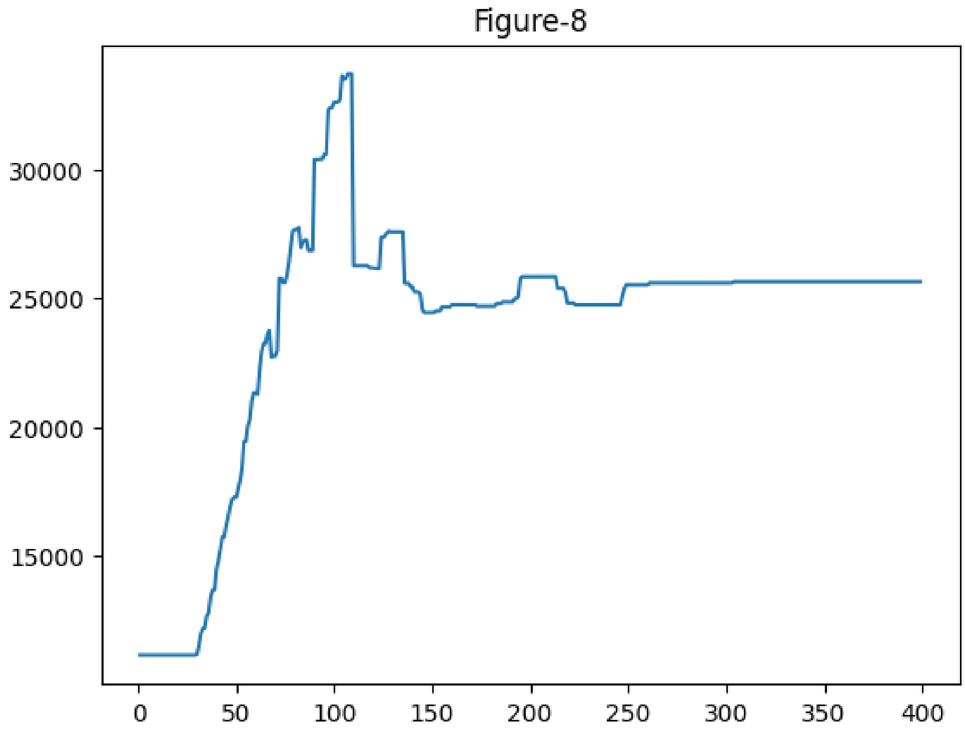
Example1:

We took one listing which has very low prediction True MAE.

predicted profit for this price: 25598 > Real Profit: 23535.0

After Calculating the profit with changing the price Highest Profit is :33717

As Figure-8 showsThe profit on Y_axis and price on X_axis:



```
Best price for best profit is : 106 And Profit will be: 33717.87816192128
```

Details of this Sample:

listing_id	since	listings count	latitude	longitude	accommodates	bedrooms	price
13764816	2014	1	48.88916	2.31737	4	1	291.0
Minimum nights	Maximum nights	Entire place,Iron, Hangers	Smoke alarm	Hair dryer	Washer	Dedicated workspace	Wifi
4	11	1	Elevator	Kitchen	TV	Bathtub	Essentials

Example2:

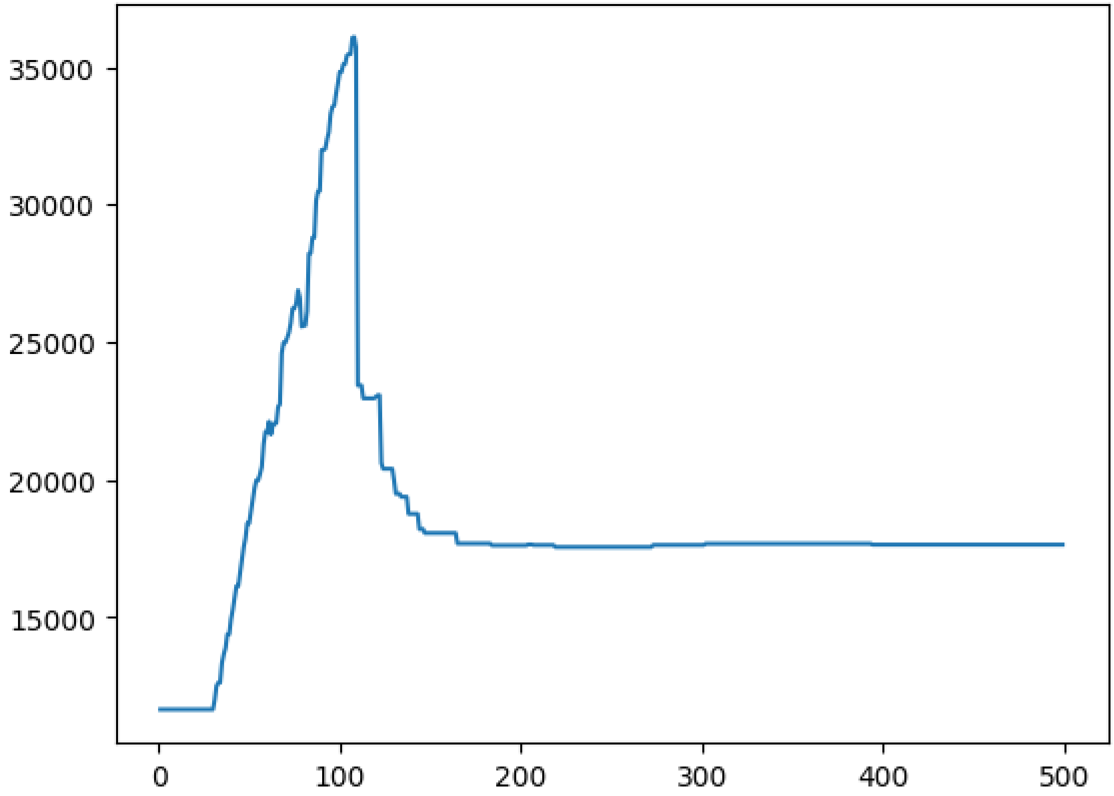
A Sample of A listing has high price and the owner should decrease the price to increase his profit. We took one Sample with very hight price and we want to reduce this price to get better

Real Profit: 16368.0 , Predicted Profit: 17648

After Calculating the profit with changing the price Highest Profit is :36095

As Figure-9 showsThe profit on Y_axis and price on X_axis:

Figure-9



```
Best price for best profit is : 107 profit will be: 36095
```

Details of this Sample:

listing_id	since	listings count	latitude	longitude	accommodates	bedrooms	price
20471163	2018	15	48.8642	2.28641	4	2	7999
Minimum nights	Maximum nights	Entire villa,Entire place, Heating	Air conditioning	Hair dryer	Washer	Dryer	Wifi
1	1125	1	Elevator	Kitchen	TV	Pool	

In conclusion here are the updated statistics for the project's performance:

The actual total profit is 91498387, while the predicted total profit is 91808513, resulting in a difference of 3676 listings from the X_test dataset.
The maximum predicted profit is 124012677, with a difference of 32204164 from the original profit.
The percentage increase in profit is 35%.
The model's total error, based on mean absolute error (MAE), is approximately 2300, resulting in a margin of error of +-7352000.
The business goal of achieving a profit between 27% to 43% has been met.
Figure-10 displays the maximum profit on the Y-axis and the listing index on the X-axis.
Figure-11 illustrates the distribution of the actual profit (blue) compared to the maximum profit with the best price (orange).

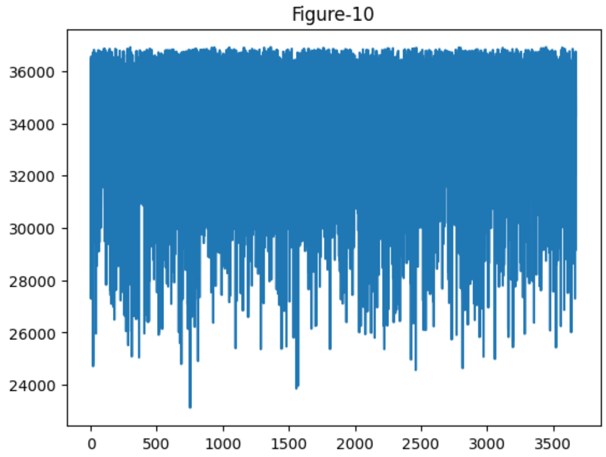


Figure-10

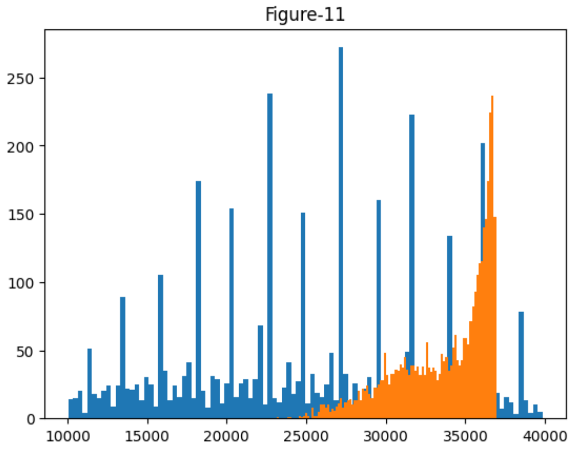


Figure-11