Report

Word Count: 1200

Introduction

Most investment decisions in the hospitality sector are strategic. This is where machine learning predictive models, data manipulation and analysis come into play. Henceforth, hotel chains are actively using business analytics to maximise profits. This can be used for forecasting analytics, better expense management, greater customer satisfaction and profitability of a hotel. This report focuses on how the study of geographical and socio-economic data of a hotel's location and its neighbourhood impact a hotel's turnover.

Objective

To build a predictive model that analyses whether a new hotel opened in a given location will be profitable or not (i.e., whether it will make a profit or a loss, regardless of the amount).

Methodology

Comparative Study

To perform the proposed study of binary classification problem in determining the *profit* or *loss* of a hotel, we perform a comparative study of the following machine learning procedures:-

a) Decision Tree Classification:- Decision trees are a type of supervised learning that can solve classification problems by creating rules that are represented as a tree structure. We use the Gini index criterion to calculate the information gain required to split the nodes.

   Advantages:

   i. Effective in building classifier models on non-linear data.
   ii. It has less tendency of underfitting.

   Disadvantages:

   i. With the increase in several attributes, the decision tree's complexity also increases.
   ii. The tree structures are sensitive to minor variations in the dataset.

b) Logistic Regression:- Logistic Regression is a classification algorithm that gives the probabilistic output of the dependent variable, which is binary in nature.

   Advantages:

   i. It is easy to implement, interpret and train.
   ii. It makes no assumption of the distribution of classes.

   Disadvantages:

   i. It can cause overfitting in high-dimensional datasets.
   ii. Non-linear problems cannot be solved with logistic Regression.

c) SVM:- Support vector machines learn from support vectors, that is, extreme data points rather than learning from correct and incorrect data.

Advantages:

i. It can be used as both a linear and non-linear classifier.
ii. It is not biased by the presence of outliers.

Disadvantages:

i. It does not perform best for a large number of features.
ii. There is no probabilistic explanation for SVM classification.

Additional Comparative Study

To perform the regression problem of building a machine learning system in determining the annual profit of a business, we perform a comparative study of the following machine learning procedures:-

a) Linear Regression:- Linear regression attempts to build a linear model between the outcome and the predictor variable.

Advantages:

i. It is easy to implement and interpret the findings.
ii. It is less complex compared to other algorithms.

Disadvantages:

i. It assumes a linear relationship between independent and dependent variables.
ii. It is susceptible to overfitting.

b) Gradient Boosting:- Gradient boosting or GBR or additive model combines multiple simple models into a more robust single composite model.

Advantages:

i. It has efficient prediction capability as it uses clone methods.
ii. It is flexible with the type of input variable.

Disadvantages:

i. It is sensitive to outliers.
ii. Tuning can be complex as it has many parameters to tune.

c) Random Forest:- Random decision forest is an ensemble learning method that constructs multitudes of decision trees and returns the mean or average of prediction of the individual trees.

Advantages:

i. It automates missing values present in the data.
ii. It helps to reduce overfitting and increase accuracy in decision trees.

Disadvantages:

i. It requires more training time compared to other machine learning algorithms.
ii. It fails to determine the significance of each variable.

Output and Interpretation

<u>Comparative Study</u>

- *Data Pre-processing*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 22 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   F1      1000 non-null   float64
 1   F2      1000 non-null   float64
 2   F3      1000 non-null   float64
 3   F4      1000 non-null   float64
 4   F5      1000 non-null   float64
 5   F6      1000 non-null   float64
 6   F7      1000 non-null   float64
 7   F8      1000 non-null   float64
 8   F9      1000 non-null   float64
 9   F10     1000 non-null   float64
 10  F11     1000 non-null   int64  
 11  F12     1000 non-null   float64
 12  F13     1000 non-null   float64
 13  F14     1000 non-null   int64  
 14  F15     1000 non-null   float64
 15  F16     1000 non-null   float64
 16  F17     1000 non-null   float64
 17  F18     1000 non-null   float64
 18  F19     1000 non-null   float64
 19  F20     1000 non-null   float64
 20  F21     500 non-null    float64
 21  Class   1000 non-null   bool   
dtypes: bool(1), float64(19), int64(2)
memory usage: 165.2 KB
```

Fig. 1

Firstly, we print the information about the data frame and look for null-values, datatypes, etc. In this dataset, we choose to drop the variable 'F21' because half of its values are missing. Further, we assign binary values of *1* and *0* to the variable ‘Class’ in place of the values *profit* and *loss*, respectively.

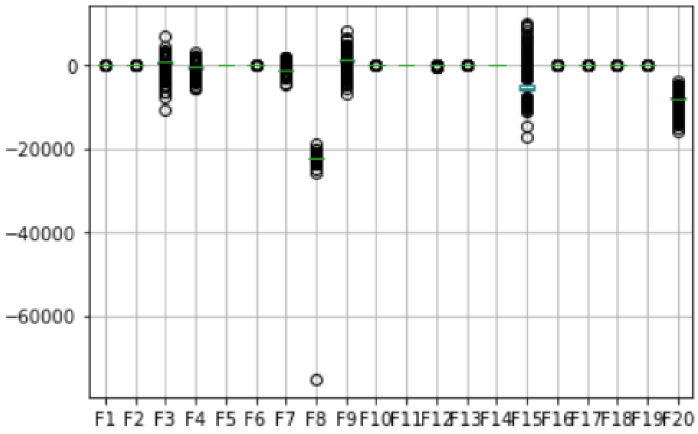

Fig. 2

We look for potential outliers in the data set to limit the variability in the data.

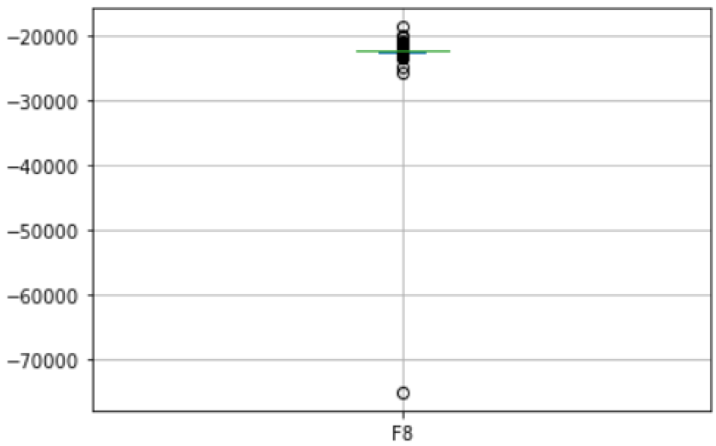

Fig. 3

```
Out[12]: (999, 21)
```

Fig. 3

In the variable 'F8', we come across one extreme value. We drop that entire row consisting of the outlier and are left with 999 data entries.

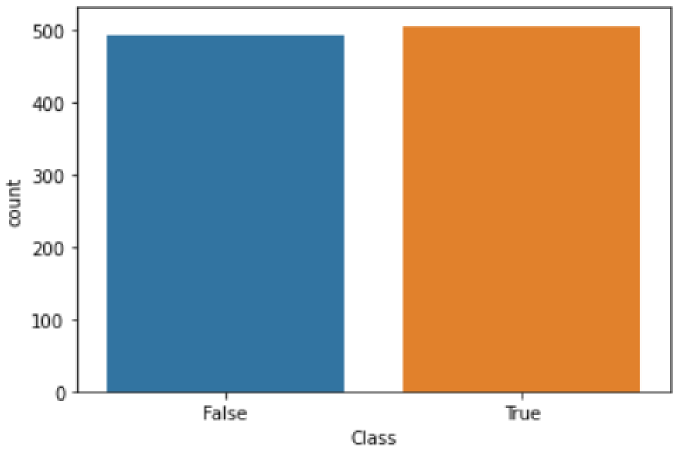

Fig. 4

Next, we check if the data is well-balanced with respect to the response variable ‘Class’. By plotting a histogram, we can verify that the dataset is almost equally balanced and proceed to split it into train and test data.

- *Data Modelling*

Decision Tree

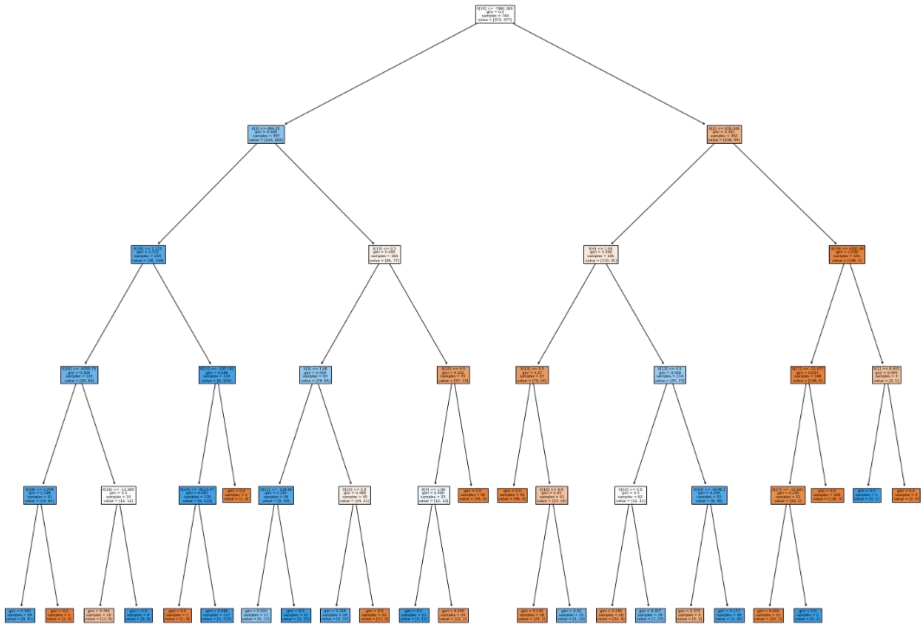

Fig. 5

A decision tree classifier is built using the “gini” criterion with a maximum depth of 5.

```
Confusion matrix:
 [[339  33]
 [ 16 361]]
AUC for the training dataset: 97.1771%
Accuracy of the training dataset: 93.4579%
```

Fig. 6

```
Confusion matrix:
 [[ 98  23]
 [ 10 119]]
AUC for the testing dataset: 85.9760 %
Accuracy of the testing dataset: 86.8000 %
```

Fig. 7

We plot a confusion matrix for the decision tree classifier and get an accuracy of 93% and 86% for training and testing data, respectively.

Logistic Regression

```
Confusion matrix:
 [[275  97]
 [127 250]]
AUC for the training dataset: 75.1119%
Accuracy of the training dataset: 70.0935%
```

Fig. 8

```
Confusion matrix:
 [[79 42]
 [40 89]]
AUC for the testing dataset: 74.9760%
Accuracy of the testing dataset: 67.2000%
```

Fig. 9

We scale the values into a common range using StandardScaler and fit a logistic regression model to the dataset. In plotting the confusion matrix for the logistic regression model, we get 70% and 67% accuracy for training and testing data, respectively.

SVM

```
{'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'}
```

Fig. 10

We create and train an SVM classifier. Then, we look for the best hyperparameters.

```
Confusion matrix:
 [[  7 114]
 [  2 127]]
```

Fig. 11

```
Accuracy for the training dataset with tuning is : 51.53%
```

Fig. 12

```
Confusion matrix:
 [[  7 114]
 [  2 127]]
```

Fig. 13

```
Accuracy for the testing dataset with tuning is : 53.60%
```

Fig. 14

In tuning the hyperparameters, we score an accuracy of 51% and 53% for training and testing data, respectively.

- *Model Comparison*

Model	**Accuracy (on test data)**
Decision Tree	86.80%
Logistic Regression	67.20%
SVM	53.60%

Fig. 15

Thus, in building a classification model for an estimate of profit or loss, the Decision Tree classifier performed best with a prediction accuracy of 86.80% on test data.

Additional Comparative Study

- *Data Pre-processing*

```
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   F1      1500 non-null   float64
 1   F2      1500 non-null   float64
 2   F3      1500 non-null   float64
 3   F4      1500 non-null   float64
 4   F5      1500 non-null   float64
 5   F6      1500 non-null   float64
 6   F7      1500 non-null   float64
 7   F8      1500 non-null   float64
 8   F9      1500 non-null   float64
 9   F10     1500 non-null   float64
 10  F11     1500 non-null   int64
 11  F12     1500 non-null   float64
 12  F13     1500 non-null   float64
 13  F14     1500 non-null   float64
 14  F15     1500 non-null   float64
 15  F16     1500 non-null   float64
 16  F17     1500 non-null   float64
 17  F18     1500 non-null   float64
 18  F19     1500 non-null   float64
 19  F20     1500 non-null   object
 20  F21     1500 non-null   float64
 21  F22     1500 non-null   float64
 22  F23     1500 non-null   float64
 23  F24     1500 non-null   float64
 24  F25     1500 non-null   float64
 25  F26     1500 non-null   float64
 26  F27     1500 non-null   object
 27  F28     1500 non-null   float64
 28  F29     1500 non-null   float64
 29  F30     1500 non-null   int64
 30  F31     1500 non-null   float64
 31  F32     1500 non-null   float64
 32  F33     1500 non-null   float64
 33  F34     1500 non-null   float64
 34  F35     1500 non-null   float64
 35  F36     1500 non-null   float64
 36  Target  1500 non-null   float64
```

Fig. 16

Firstly, we print the information about the data frame and look for null-values, datatypes, etc. We assign numerical values to categorical variables depending upon irrespective of any ordinal relationship.

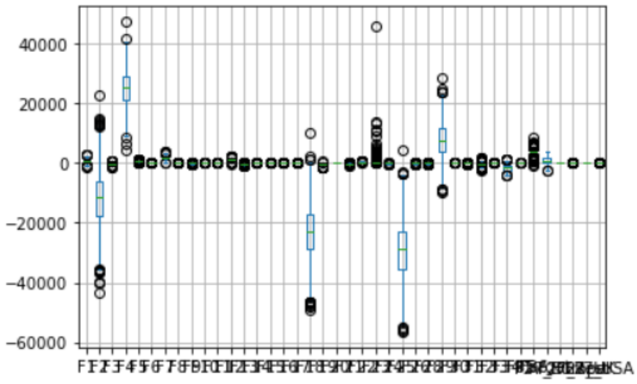

Fig. 17

Out[10]: (1155, 40)

Fig. 18

We detect and remove the outliers from the dataset using z-scores and leave 1155 data entries. Then, we check for multicollinearity as it may weaken the statistical significance of independent variables in building a regression model. There seems to be an insignificant correlation among the independent variables, and we proceed to split the dataset into train and test data.

- *Data Modelling*

Linear Regression

The score of the model on train data is: 71.0995%

Fig. 19

The score of the model on test data is: 71.5175%

Fig. 20

We fit a linear regression model to the data and achieved an accuracy of 71% for both training and testing data.

Gradient Boosting

The score of the model on train data is: 94.7161%

Fig. 21

The score of the model on test data is: 83.8603%

Fig. 22

By fitting gradient boosting regressor to the data, we get an accuracy of 94% and 83% for training and testing data, respectively.

Random Forest

```
The score of the model on train data is: 95.2091%
```

Fig. 23

```
The score of the model on test data is: 66.4840%
```

Fig. 24

For random forest regressors, an accuracy of 95% and 66% is obtained for training and testing data, respectively.

- *Model Comparison*

Model	**Accuracy (on test data)**
Linear Regression	71.51%
Gradient Boosting	83.86%
Random Forest	66.48%

Thus, in building a regression model for evaluating the profitability of a business, the Gradient Boosting classifier performed best with a prediction accuracy of 83.86% on test data.

Conclusion

Data insights can give a great advantage to hotels and companies with the right set of data at our disposal. The success of hotels depends on the characteristics of each hotel. Some factors play a crucial role in determining the profitability of a hotel. Hence, there is a need to find out more about these factors. In conclusion, identifying the factors that affect the profitability of a hotel company can predict future growth and turnover. A research focus on identifying and isolating the impact of various quantitative and qualitative variables on profitability can be conducted for further accuracy and precision.