# Analysis of Hotel Bookings & Cancellation patterns

**Data Intensive Computing 587 PHASE 2**
Manikanta Kalyan Gokavarapu - mgokavar - 50465129
Rakesh Kumar Gavara - rgavara - 50483851

## 1. Problem Statement

The aim of this project is to detect patterns and tendencies in customer actions that can be utilized to improve revenue management, enhance resource allocation, optimize customer service, and shape marketing and promotional strategies. Through an analysis of historical booking and cancellation data, **the goal is to develop precise forecasts about future cancellations**/demand and make necessary operational adjustments to meet customer requirements and maximize profitability.

### 1.1 Pre-Processing the data again for modeling:

- During modeling we got a need to again preprocess some of the attributes of the data. We have encoded the attributes country, reserved_room_type, assigned_room_type, customer_type, deposit type, arrival_year using label encoding and converted the string data into integers.
- Dropped some unwanted columns like reservation_status, reservation_status_date which are directly related to the target variable (is_canceled).
- Using the relevant domain knowledge and correlation matrix output in the figure [1] for target variable and the input features, We dropped some low correlated features like waiting_days, country, booking_changes, arrival_week_number, arrival_month, arrival_date_day_of_month. To get a better performing model.

```
arrival_week_number             0.001089
arrival_month                   0.003332
arrival_date_day_of_month       0.005327
waiting_days                    0.015067
babies                          0.021676
customer_type                   0.031001
meal                            0.045317
reserved_room_type              0.046926
weekend_night_stays             0.059610
kids                            0.059810
assigned_room_type              0.060118
children                        0.066557
hotel                           0.069510
week_night_stays                0.083903
adults                          0.084302
arrival_year                    0.087723
is_repeated_guest               0.089300
country                         0.097190
guests                          0.097243
booking_changes                 0.099017
previous_uncancelled_bookings   0.101116
special_requests                0.122146
adr                             0.125695
prev_cancel                     0.126658
deposit_type                    0.137384
market_segment                  0.180605
lead_time                       0.183321
required_car_parking_spaces     0.185317
is_canceled                     1.000000
Name: is_canceled, dtype: float64
```

**Figure [1]**

## 2. Algorithms/Visualizations, Explanation and Analysis:

Below are the Algorithms we have used to predict/classify whether a customer makes a cancellation or not for his booking.

- KNN (Classroom Algorithm)
- Naive Bayes (Classroom Algorithm)
- Random Forest
- Logistic Regression (Classroom Algorithm)
- SVM.

## 2.1 K-Nearest Neighbors Algorithm (KNN)

### 2.1.1 Intro and working of algorithm:

Generally we use the KNN to classify or label objects/data points. We start with already labeled data points and use proximity to make classification. The basic intuition for KNN is for a new unlabeled element, we look at k most similar elements in the labeled dataset based on various attributes, and choose the label that most of the elements have.

### 2.1.2 Feature Selection:

Target Variable : Is_canceled - A binary column that indicates whether the reservation was canceled (1) or not (0).
Input Features : Based on the correlation output above in the figure 1, we have removed all the unwanted columns and choose all the features as inputs except the target variable. As all features have some amount of correlation with the target variable.

### 2.1.3 Preparing Train and Test Data:

We have used 70% of the data to train the model and the remaining 30% we used to test the model with a random state equal to 0.

### 2.1.4 Choosing Optimal K-value:

- For the KNN algorithm choosing the optimal **K** value is an important step. Here K represents the number of nearest neighboring data points we need to consider to classify a new datapoint.
- For a model to perform well we need to minimize the error rate. So we need to find a K value where the error rate is minimum so that will be our optimal K-value.

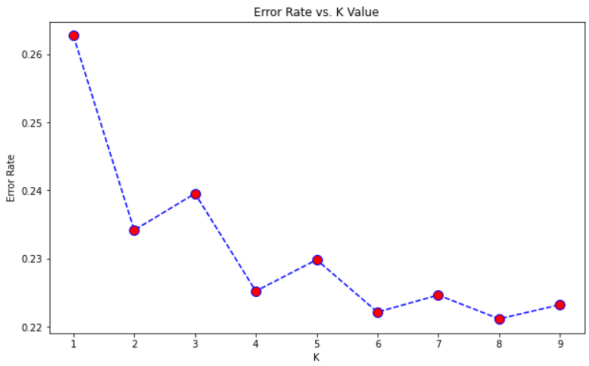


**Figure [2]**

- From the figure 2 we can see that at K=8 the error rate is minimum. So the optimal K-value is 8.

### 2.1.5 KNN model.

I have used an inbuilt KNeighborsClassifier in sklearns and passed on the optimal K value as n_neighbors parameter in KNeighborsClassifier. The results are given below.

### 2.1.6 Result Visualizations:

- The Accuracy of the model is given in the below figure [3] which is around 78 percent.

```
Accuracy Score of KNN is : 0.7788749711826635
```

**Figure [3]**

- The Accuracy of the model is the highest at the Optimal K value where there is minimum error rate and that can be found from the below figure [4].

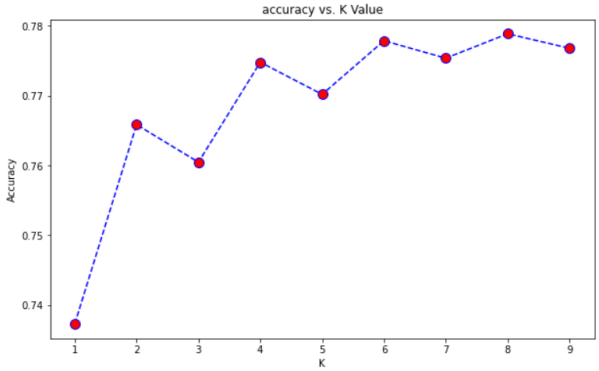


**Figure [4]**

- Generally if we have two-dimensional input data and a binary classification problem, we can plot the data points as a scatter plot and color-code them by their class. But as we have more than 2 input features it will become difficult to visualize the data so we can create a confusion matrix to see how well the KNN model predicts each class. A confusion matrix shows the number of true positives, false positives, true negatives, and false negatives for each class as show in the below **figure [5]**

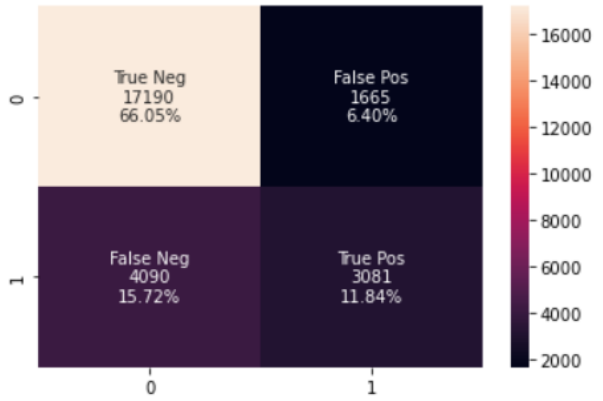


**Figure [5]**

- The performance of a machine learning model used for classification tasks can be assessed using a classification report. It offers a thorough breakdown of numerous metrics that can aid in your understanding of how well your model is doing.
- These parameters, as well as the overall metrics for the model, are included in the classification report for each class in the dataset. It can be a useful tool for analyzing your model's advantages and disadvantages as well as for pinpointing potential areas for performance enhancement. The classification report for my model is given in the below **figure [6]**

Classification Report :

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.81 | 0.91 | 0.86 | 18855 |
| 1 | 0.65 | 0.43 | 0.52 | 7171 |
| accuracy | | | 0.78 | 26026 |
| macro avg | 0.73 | 0.67 | 0.69 | 26026 |
| weighted avg | 0.76 | 0.78 | 0.76 | 26026 |

**Figure [6]**

- The performance of a binary classifier as the discrimination threshold is changed is depicted graphically by a ROC (Receiver Operating Characteristic) curve. The False Positive Rate (FPR) is plotted against the True Positive Rate (TPR) at various threshold values. It is shown in the below **figure [7]**

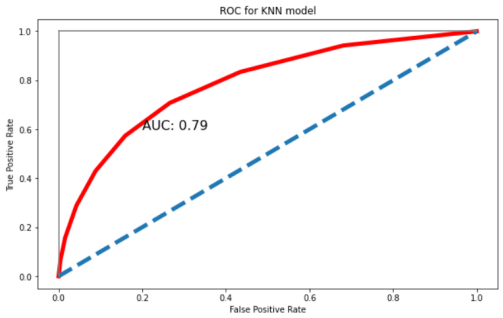


**Figure [7]**

### 2.1.7 Explanation and Analysis.

**Why choose the KNN algorithm?**

- The first reason for choosing KNN for cancellations predictions data is it is very easy to implement and requires very less Hyper parameters i.e. we only require an optimal K value.
- Secondly, as my data consists of both linear and non linear data KNN will be a good fit because it doesn't make any assumptions about the distribution of the data.
- For KNN we don't need to train the model on entire data because it will store the training data and will make the necessary classifications/predictions.

**Work we had to do to tune and train the model:**

- As Explained in the sections 1.1, 2.1.2, 2.1.3, 2.1.4 we had to preprocess the data for KNN model and we had to split the data and we had to find the optimal K value to get the best results.

**Analysis of Effectiveness of the Algorithm:**

- Accuracy is a good method to judge how effective an algorithm is working. The KNN model gave an accuracy of nearly 78 % at the optimal K-value of 8 for the given cancellation prediction dataset implying the model is performing well.
- From the confusion matrix in figure 5 we could see that True Negatives implying no.of records actually not canceled and predicted not canceled bookings, True Positives implying no.of actually canceled and predicted canceled bookings, False Positives implying no.of not canceled records given as canceled, False Negative implying no.of canceled bookings given as not canceled. As the amount

of true positives and negatives are far greater than false positives and negatives we can say the model is effective.

- From the classification report in figure 6 we could see the precision, recall, f1-score and support of both the classes (0 - not canceled, 1- canceled). For the class 0 the model is performing well as precision, recall, f1-score are close to 1 and has high support with good number of instances for the class. But if we see the class 1 metrics precision, recall they are not so good especially recall. Recall is generally the fraction of true positive instances that are correctly identified by the model. So we can say the model is a little bit behind on identifying true positive instances as there is less support for the class 1 implying less number of instances are there. To improve we need to have more data on cancelation data.
- As the ROC curve in figure 7 is close to the upper left corner of the plot, indicating a high TPR and low FPR for a wide range of threshold values and also has a good AUC score of 0.79 which is close to 1 indicating the model is performing well in predicting/classifying cancellations.

## 2.2 Naive Bayes Algorithm (NB)

### 2.2.1 Intro and working of algorithm:

Naive Bayes is a statistical model generally used for classification. There are three types of Naive Bayes models: Multinomial, Gaussian, Bernoulli. I have used **Multinomial Naive Bayes** for the bookings data. The basis of Naive Bayes is it finds the probability of an event, based on prior knowledge of conditions that might be related to the event. It mainly uses Bayes's law to determine the probabilities of each class for given input features and compares those probabilities for classification.

### 2.2.2 Feature Selection:

Target Variable : Is_canceled - A binary column that indicates whether the reservation was canceled (1) or not (0).

Input Features : Based on the correlation output above in the figure 1, we have removed all the unwanted columns and choose all the features as inputs except the target variable. As all features have some amount of correlation with the target variable.

### 2.2.3 Preparing Train and Test Data:

We have used 70% of the data to train the model and the remaining 30% we used to test the model with a random state equal to 0.

### 2.2.4 Naive Bayes Model.

I have used an inbuilt MultinomialNB model in sklearns and passed on the required hyperparameters needed for the model like alpha and prior probabilities. The results are given below.

### 2.2.5 Result Visualizations:

- The Accuracy of the model is given in the below figure [8] which is around 76 percent.

```
Accuracy Score of Multinomial naive bayes is : 0.7510182125566741
```

**Figure [8]**

- The prior probabilities of both the classes and posterior probabilities (feature probabilities : log probabilities of 22 input features for each class) for multinomial Naive Bayes are given in the figure 9 below. These probabilities are used for finding the final probability for each class for the given input features.

```
Class priors:
[0.72333558 0.27666442]
Feature probabilities:
[[ -4.24089117  -2.15011337  -3.18585644  -3.37783102  -2.42299241
   -2.72935091  -5.45708454  -7.75760371  -3.9843469   -1.74926083
   -6.35134119  -8.11186934  -6.33628273  -3.15433026  -2.79762336
   -9.08736525  -2.62075737  -1.85322997  -5.50977401  -3.61849746
   -5.36175727  -2.65981008]
 [ -4.51983869  -1.99169863  -3.15358291  -3.32194329  -2.36612482
   -2.75400062  -5.10580396  -8.36197557  -3.86988076  -1.71934753
   -7.94546062  -6.45986804  -8.44898694  -3.08343583  -3.03981961
   -6.56467592  -2.71493982  -1.88235364 -13.14946731  -4.04426525
   -5.06830153  -2.65124506]]
```

**Figure [9]**

- As we have more than 2 input features it will become difficult to visualize the data so we can create a confusion matrix to see how well the Naive Bayes model predicts each class. A confusion matrix shows the number of true positives, false positives, true negatives, and false negatives for each class as show in the below **figure [10]**

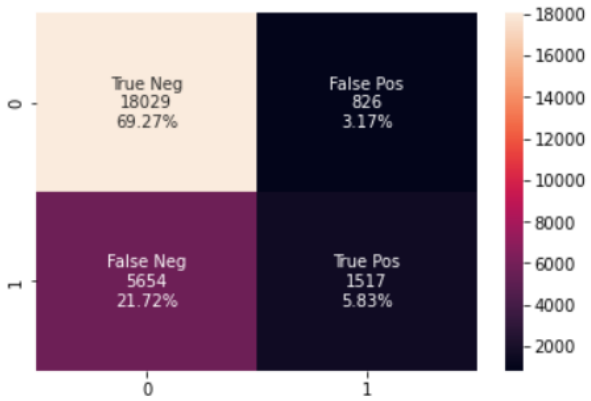


**Figure [10]**

- The performance of a machine learning model used for classification tasks can be assessed using a classification report. It offers a thorough breakdown of numerous metrics that can aid in your understanding of how well your model is doing.
- These parameters, as well as the overall metrics for the model, are included in the classification report for each class in the dataset. It can be a useful tool for analyzing your model's advantages and disadvantages as well as for pinpointing potential areas for performance enhancement. The classification report for my model is given in the below **figure [11]**

Classification Report :

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.76 | 0.96 | 0.85 | 18855 |
| 1 | 0.65 | 0.21 | 0.32 | 7171 |
| accuracy | | | 0.75 | 26026 |
| macro avg | 0.70 | 0.58 | 0.58 | 26026 |
| weighted avg | 0.73 | 0.75 | 0.70 | 26026 |

**Figure [11]**

- The performance of a binary classifier as the discrimination threshold is changed is depicted graphically by a ROC (Receiver Operating Characteristic) curve. The False Positive Rate (FPR) is plotted against the True Positive Rate (TPR) at various threshold values for the Multinomial Naive Bayes Model. It is shown in the below **figure [12]**

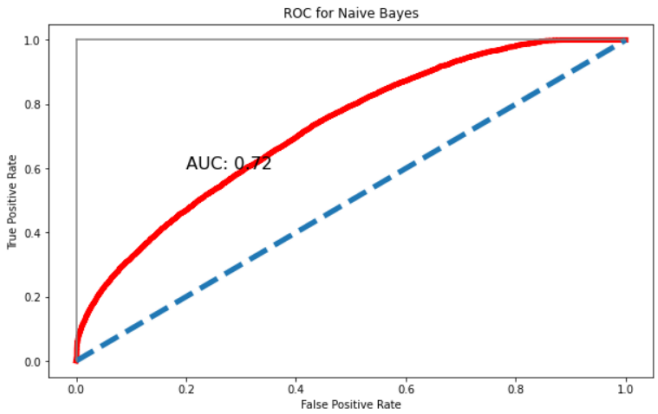


**Figure [12]**

### 2.2.6 Explanation and Analysis.

**Why choose the Naive Bayes algorithm?**

- The first reason for choosing Naive Bayes for Booking cancellations predictions data is Naive Bayes works well with High Dimensional Data as we have 22 input features the processing time will be more. But in naive bayes as it is based on probabilities it is faster even when we have high dimensional data.
- Secondly, Naive Bayes doesn't get affected by irrelevant data. It assumes that all the features are independent. So even if some features don't have a relation between them the classification results are not affected.
- Specifically Multinomial Naive Bayes because it handles discrete data very well. So as my data consists of large amounts of discrete/categorical data, multinomial naive bayes will be very useful.

**Work we had to do to tune and train the model:**

- As Explained in the sections 1.1, 2.2.2, 2.2.3 we need to tune the data based on the model and also we need to include some hyper parameters like alpha set 1.0 which is used to avoid zero probabilities, basically it corresponds to laplace smoothing. And we use fit_prior and set it to true so the model will automatically estimate the prior probabilities of each class from the training data. So it will result in better performance of the model.

**Analysis of Effectiveness of the Algorithm:**

- Accuracy is a good method to judge how effective an algorithm is working. The model gave an accuracy of nearly 75 % for the given cancellation prediction dataset implying the model is doing good classification.
- From the confusion matrix in figure 10 we could see that True Negatives implying no.of bookings actually not canceled and predicted not canceled bookings - 18029, True Positives implying no.of actually canceled and predicted canceled bookings - 1517, False Positives implying no.of not canceled bookings given as

canceled - 826, False Negative implying no.of canceled bookings predicted as not canceled - 5654 . As the amount of true positives and negatives are far greater than false positives and negatives we can say the model is effective.

- From the classification report in figure 11 we could see the precision, recall, f1-score and support of both the classes (0 - not canceled, 1- canceled). For the class 0 the model is performing well as precision, recall, f1-score are close to 1 and has high support with good number of instances for the class. But if we see the class 1 metrics f1-score, recall they are not so good. Recall is generally the fraction of true positive instances that are correctly identified by the model. So we can say the naive bayes model is also a little bit behind on identifying true positive instances as there is less support for the class 1 implying less number of instances are there. To improve we need to have more data on cancelation data. As precision and recall is less for class 1 obviously the F1 score will be less as it is the harmonic mean of precision and recall.
- The prior and posterior probabilities in figure 9 are used to calculate the final individual class probabilities of each class. For class 1 we multiply the prior probability of class 1 with all the 22 input features probabilities of class 1 and find the final class 1 probability. In the similar fashion for class 0 we multiply the prior probability of class 0 with all the 22 input feature probabilities of class 0 and find the final class 0 probability. For the new data we compare the final class 1 probability and class 2 probability and assign the class of the highest probability to the new data.
- As the ROC curve in figure 12 is close to the upper left corner of the plot, indicating a high TPR and low FPR for a wide range of threshold values and also has a good AUC score of 0.72 which is close to 1 indicating the model is performing well in predicting/classifying cancellations.

## 2.3 Random Forest Algorithm (RF)

### 2.3.1 Intro and working of algorithm:

The working of Random Forest Algorithm mainly depends on the Decision tree.

**Decision Tree:** A Decision tree is a binary tree that recursively splits the dataset until we are left with pure leaf nodes that is the data with only one type of class. The basic working of decision tree is as follows, the data is feed into root node of a decision tree and based on a specific splitting condition the data is split into two parts like for example data points satisfying a specific split condition goes into one node and which doesn't satisfy the condition goes to other node and if we get a node with a mix of both types of classes data we iterate further down the decision tree until we get a leaf node with only one type of class data. But if we have more complex data, we cannot achieve 100% pure leaf nodes. In those cases, we opt for "majority voting" and we will assign the majority class of the

data points to the test point. And we choose the optimal split condition based on maximum information gain and this uses entropy.

**Random Forest:** Random Forest is one of the Ensemble techniques. Ensemble means combining various models and these models are utilized to train the data and get a needed output. In ensemble techniques we have two types one is bagging and other one is boosting. Random forest is one of the techniques that uses the bagging concept which is also known as Bootstrap aggregation. The Basic working of Random Forest is as follows, suppose if we have a dataset with 'd' records or rows and 'm' number of features or columns. Now we select some rows using row sampling with replacement (RS) and we select some columns or features known as feature sampling (FS) from the original dataset and we feed this data into the first decision tree. Similarly, we create multiple bootstrapped datasets using RS and FS and we will feed this data into different decision trees. The decision trees get trained using the input data and produce an output. Now in random forests we take the majority voting we consider the output that has the maximum number of votes.

### 2.3.2 Feature Selection:

Target Variable : Is_canceled - A binary column that indicates whether the reservation was canceled (1) or not (0).
Input Features : Based on the correlation output above in the figure 1, we have removed all the unwanted columns and choose all the features as inputs except the target variable. As all features have some amount of correlation with the target variable.

### 2.3.3 Preparing Train and Test Data:

We have used 70% of the data for training the model and the rest 30% we used to test the model with a random state equal to 0.

### 2.3.4 Random Forest model.

I have used an inbuilt RandomForestClassifier in sk-learns and passed on hyper parameters 'n-estimators' as 112 'minimum sample splits' as 4. The results are given below.

### 2.3.5 Result Visualizations:

- The Accuracy of the model is given in the below figure [13] which is around 81 percent.

```
Accuracy Score of Random Forest is : 0.8085376162299239
```

**Figure [13]**

- But as we have more than 2 input features it will become difficult to visualize the data so we can create a confusion matrix to see how well the Random forest model classifies each class. A confusion matrix shows the number of true positives, false positives, true negatives, and false negatives for each class as show in the below **figure [14]**

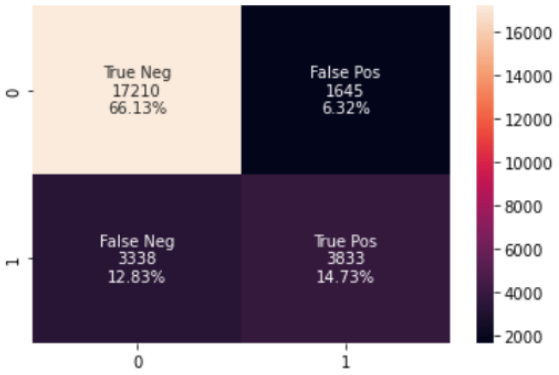


**Figure [14]**

- The performance of a machine learning model used for classification tasks can be assessed using a classification report. It offers a thorough breakdown of numerous metrics that can aid in your understanding of how well your model is doing.
- These parameters, as well as the overall metrics for the model, are included in the classification report for each class in the dataset. It can be a useful tool for analyzing your model's advantages and disadvantages as well as for pinpointing potential areas for performance enhancement. The classification report for my model is given in the below **figure [15]**

```
Accuracy Score of Random Forest is : 0.8085376162299239
Confusion Matrix :
[[17210  1645]
 [ 3338  3833]]
Classification Report :
              precision    recall  f1-score   support

           0       0.84      0.91      0.87     18855
           1       0.70      0.53      0.61      7171

    accuracy                           0.81     26026
   macro avg       0.77      0.72      0.74     26026
weighted avg       0.80      0.81      0.80     26026
```

**Figure [15]**

- The performance of a binary classifier as the discrimination threshold is changed is depicted graphically by a ROC (Receiver Operating Characteristic) curve. The False Positive Rate (FPR) is plotted against the True Positive Rate (TPR) at various threshold values. It is shown in the below **figure [16]**

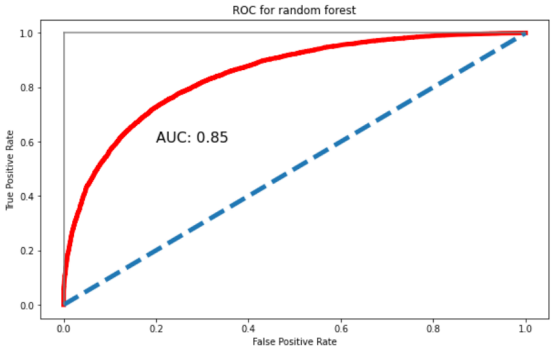


**Figure [16]**

### 2.3.6 Explanation and Analysis.

**Why choose the Random Forest algorithm?**

- The first reason for choosing Random Forest is it is not affected by noisy data. Because it follows an approach of majority voting so it is the result of multiple decision trees so the noise is significantly reduced.
- Secondly, Unlike the decision tree classification, random forest is less prone to overfitting. Actually as it creates multiple trees it improves the accuracy of the model.
- Another reason for choosing this algorithm is it works very well on large datasets even with a large number of input features.
-

**Work we had to do to tune and train the model:**

- As Explained in the sections 1.1, 2.3.2, 2.3.3, 2.3.4 we had to preprocess the data for Random Forest model and we had to split the data and we had to find optimal n_estimators value to get the best results. Here n_estimators is nothing but the number of trees in the forest. In figure 17 we can see that at 112 trees we are getting the best accuracy and in the similar fashion tried different values of min sample splits - samples required to split an internal node and found the optimal value which is 4 for my dataset.

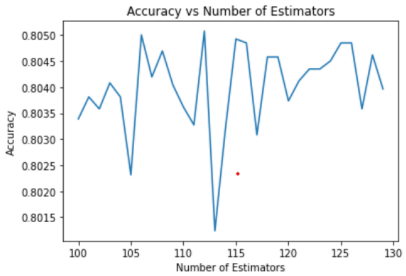


**Figure [17]**

**Analysis of Effectiveness of the Algorithm:**

- Accuracy is a good method to judge how effective an algorithm is working. The Random Forest model gave an accuracy of nearly 81 % at the optimal n -estimatiors-value of 112 and min splits value of 4 for the given bookings cancellation dataset. The accuracy of the random forest is higher than KNN and naive bayes algorithms because it is an ensemble technique which uses multiple small classifiers to get the output data. So, this algorithm is more effective.
- From the confusion matrix in figure 14 we could see that True Negatives implying no.of records actually not canceled and predicted not canceled bookings - 17210, True Positives implying no.of actually canceled and predicted canceled bookings -3833, False Positives implying no.of not canceled records given as canceled - 1645, False Negative implying no.of canceled bookings given as not canceled - 3338. As the amount of true positives and negatives are far greater than false positives and negatives we can say the model is effective.
- From the classification report in figure 15 we could see the precision, recall, f1-score and support of both the classes (0 - not canceled, 1- canceled). For the class 0 the model is performing well as precision, recall, f1-score are close to 1 and has high support with good number of instances for the class. In the random forest the class 1 metrics precision and recall also improved. Recall is generally the fraction of true positive instances that are correctly identified by the model. But this needs to be further improved. We need more instances of true positives in the data to improve it further.
- As the ROC curve in figure 16 is close to the upper left corner of the plot, indicating a high TPR and low FPR for a wide range of threshold values and also has a good AUC score of 0.85 which is close to 1 indicating the model is performing well in predicting/classifying cancellations.
- As it is difficult to visualize all the decision trees in a random forest classification I have plotted a single decision tree based on the decision tree classifier. As the input features are large it is difficult to visualize the tree but we can see in the below snippet decision tree in figure 15 where the input features are used as split conditions and constantly reducing the entropy of the tree to achieve pure leaf node (entropy = 0) to classify the data.

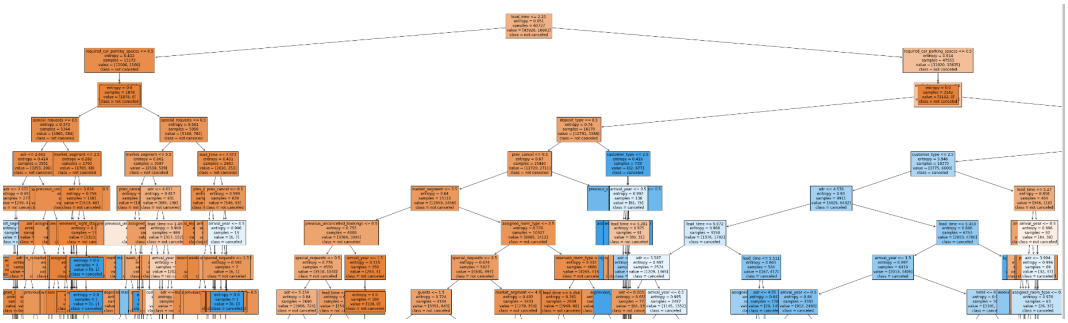

**Figure [18]**

## 2.4 Logistic regression Algorithm

### 2.4.1 Intro and working of algorithm:

Logistic regression is a supervised classification algorithm to classify the data into classes(label 1, label 2).In logistic regression, the dependent variable is modeled as a function of the independent variables using a logistic function which is generally if form of a S-shaped curve, which transforms the continuous values of the independent variables into probabilities of the dependent variable. The logistic function estimates the probability of an event occurring, given the values of the independent variables.

The optimization objective of logistic regression is typically to maximize the log-likelihood of the training data, which measures how well the algorithm fits the observed data.

Below **Figure [19]** depicts the logistic regression curve:

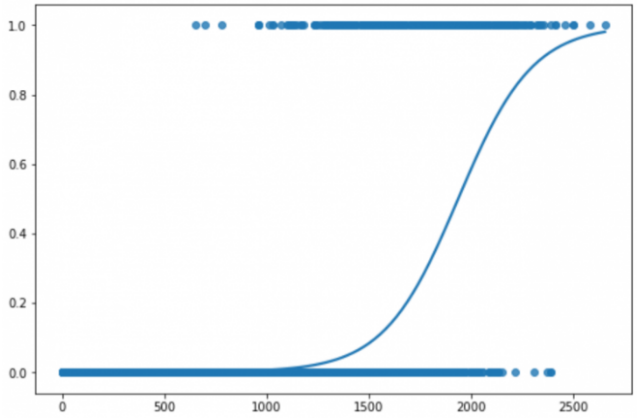


**Figure [19]**

### 2.4.2 Feature Selection:

Target Variable : Is_canceled - A binary column that indicates whether the reservation was canceled (1) or not (0).

Input Features : Based on the correlation output above in the figure 1, we have removed all the unwanted columns and choose all the features as inputs except the target variable. As all features have some amount of correlation with the target variable.

### 2.4.3 Preparing Train and Test Data:

We have used 70% of the data for training the model and the rest 30% for testing the model with a random state equal to 0.

### 2.4.4 Logistic regression model & hyper parameters used.

Since we have used Binary logistic regression for model prediction. Binary logistic regression is a type of logistic regression which is used when we have only two possible outcomes in the target/dependent variable('is_canceled'). Hyper parameters used here are max_iter: '3000', solver: 'sag' and penalty:'l2' respectively. The parameter max_iter specifies the upper limit on the number of iterations or epochs during which the optimization algorithm will attempt to discover the best weights for the model. When working with large datasets that have a substantial number of features and samples, the 'sag' solver outperforms other solvers in terms of speed. Since our data is of high dimensionality logistic regression may lead to overfitting. In order to avoid it we have used penalty parameters 'l2' as a regularization technique.

In the below **Figure 20** we can observe that for the solver sag the accuracy is maximum

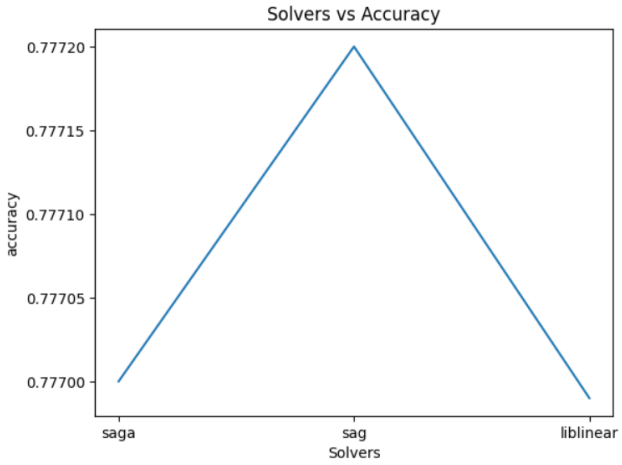


**Figure [20]**

### 2.4.5 Result Visualizations:

- The Accuracy of the model is given in the below **Figure [21]** which is around 77.7 percent.

```
Accuracy Score of Logistic Regression is : 0.7772612003381234
```

**Figure [21]**

- Generally if we have two-dimensional input data and a binary classification problem, we can plot the data points as a scatter plot and color-code them by their class. But as we have more than 2 input features it will become

difficult to visualize the data so we can create a confusion matrix to see how well the logistic regression model predicts each class. A confusion matrix shows the number of true positives, false positives, true negatives, and false negatives for each class as show in the below **Figure [22]**

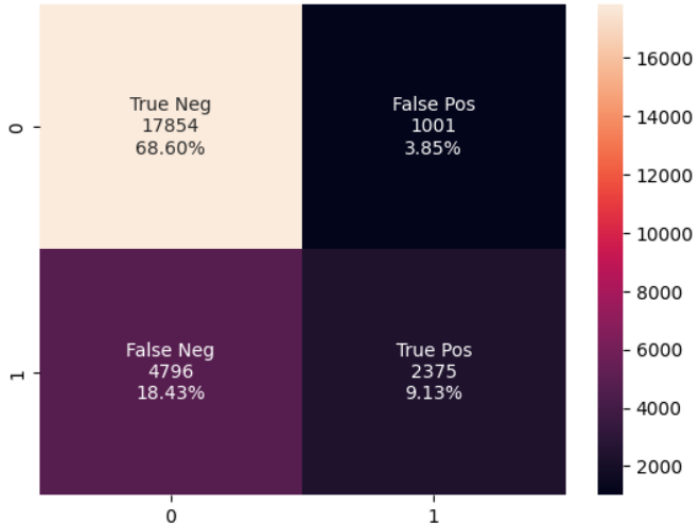


**Figure [22]**

- The performance of a machine learning model used for classification tasks can be assessed using a classification report. It offers a thorough breakdown of numerous metrics that can aid in your understanding of how well your model is doing.
- These parameters, as well as the overall metrics for the model, are included in the classification report for each class in the dataset. It can be a useful tool for analyzing your model's advantages and disadvantages as well as for pinpointing potential areas for performance enhancement. The classification report for my model is given in the below **Figure [23]**

```
Confusion Matrix :
[[17854  1001]
 [ 4796  2375]]
Classification Report :
              precision    recall  f1-score   support

           0       0.79      0.95      0.86     18855
           1       0.70      0.33      0.45      7171

    accuracy                           0.78     26026
   macro avg       0.75      0.64      0.66     26026
weighted avg       0.76      0.78      0.75     26026
```

**Figure [23]**

- The performance of a binary classifier as the discrimination threshold is changed is depicted graphically by a ROC (Receiver Operating Characteristic) curve. The False Positive Rate (FPR) is plotted against the True Positive Rate (TPR) at various threshold values. It is shown in the below **Figure [24]**

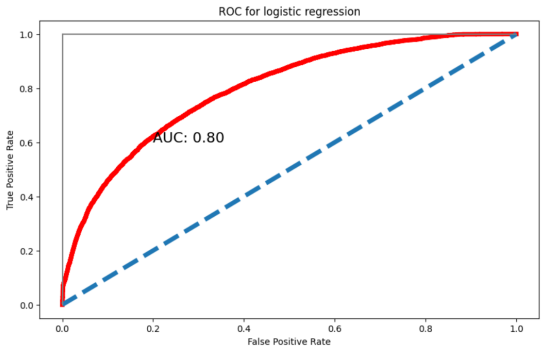


**Figure [24]**

### 2.4.6 Explanation and Analysis.

**Why choose the Logistic regression algorithm?**

- We chose Logistic regression as its effective over a large data set. It is also easier to understand, implement and comprehend .
- Logistic regression can quickly train huge data sets like ours, eventually saving computation time.

**Work we had to do to tune and train the model:**

- As Explained in the sections 1.1, 2.4.2, 2.4.3 and 2.4.4 we had to preprocess the data for the logistic regression model, find an ideal set of hyper parameters so as to improvise the computation of logistic regression.

**Analysis of Effectiveness of the Algorithm:**

- Accuracy is a good method to judge how effective an algorithm is working. The logistic model gave an accuracy of nearly 77 % given cancellation prediction dataset implying the model is performing well.
- From the confusion matrix in figure 22 we could see that the sum of (True positives and True negatives) is 2375 + 17854 = 20229. As the amount of true positives and negatives are far greater than false positives and negatives which is (1001 + 4796 = 5797) 5797 we can say the model is effective.
- From the classification report in Figure 23 we could see the precision, recall, f1-score and support of both the classes (0 - not canceled, 1- canceled). For the class 0 the model is performing well as precision, recall, f1-score are close to 1

and has high support with good number of instances for the class. We have a f1 score of 0.86 for class 0 and 0.45 for class 1 which indicates that the model performance is better for class 0 rather than class 1. But if we see the class 1 metrics precision, recall they are not so good especially recall. Recall is generally the fraction of true positive instances that are correctly identified by the model. So we can say the model is a little bit behind on identifying true positive instances as there is less support for the class 1 implying less number of instances are there. To improve we need to have more data on cancelation data.

From ROC we can observe the trend of True positive rate and false positive rate (TPR & FPR). From the figure 25we can observe that the true positive rate is higher for small values of false positive rate till FPR reaches 0.18. As the ROC curve in figure 25 is close to the upper left corner of the plot, indicating a high TPR and low FPR for a wide range of threshold values and also has a good AUC score of 0.80 which is close to 1 indicating the model is performing well in predicting/classifying cancellations.

## 2.5 Support vector machine Algorithm

### 2.5.1 Intro and working of algorithm:

SVM is a supervised machine learning algorithm used for classification and regression analysis. It is a powerful and popular algorithm in the field of machine learning because of its ability to handle both linear and non-linear data, as well as high-dimensional data.

For binary classification, the SVM algorithm determines the hyperplane that maximizes the margin between the data points of the two classes. The data points that are nearest to the hyperplane are referred to as support vectors and are instrumental in determining the position of the hyperplane. By maximizing the distance between the support vectors and the hyperplane, the SVM algorithm enhances the model's ability to generalize to new data. The SVM algorithm can handle high-dimensional data effectively and has applications in various domains such as text classification, image classification, and bioinformatics.

Below **Figure 25** depicts the SVM classification

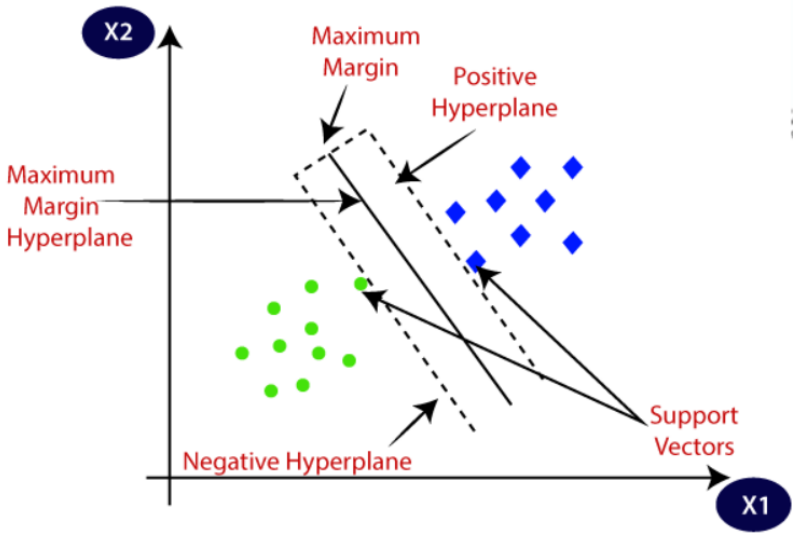


**Figure [25]**

**2.5.2 Feature Selection:**

Target Variable : Is_canceled - A binary column that indicates whether the reservation was canceled (1) or not (0).

Input Features : Based on the correlation output above in the figure 1, we have removed all the unwanted columns and choose all the features as inputs except the target variable. As all features have some amount of correlation with the target variable.

**2.5.3 Preparing Train and Test Data:**

We have used 70% of the data for training the model and the rest 30% for testing the model with a random state equal to 0.

**2.5.4 Hyper parameters tuning.**

One of the most difficult tasks in SVM is to select proper hyper parameters for tuning compared to other algorithms. Instead of trying permutations & combinations we have an efficient way to identify ideal hyper parameters. We used ‘gridSearchCV’ for this. Hyper parameters used here are kernel = 'rbf', C = 1, gamma = 0.1 respectively. SVM with kernel ‘rbf’(Radial basis function) is a type of SVM which models non-linear decision boundaries. It achieves this by first mapping each data point into a high-dimensional space and then utilizing the RBF kernel to calculate the similarity between them. The kernel computes the distance between each pair of data points using the Euclidean distance metric and generates a similarity value that serves to create a decision boundary that divides the data points into distinct classes.’C’ is a hypermeter in SVM to control error.Gamma is a hyperparameter which we have to set before training model.

Gamma decides how much curvature we want in a decision boundary.Gamma is used with the 'rbf' kernel.

Below **Figure 26** is just a screenshot showing combinations used by gridSearchCV to find the optimal Hyper parameter values for SVM

```
Fitting 5 folds for each of 4 candidates, totalling 20 fits
[CV 1/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.776 total time=   7.4s
[CV 2/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.781 total time=   6.9s
[CV 3/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.778 total time=   6.9s
[CV 4/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.778 total time=   6.9s
[CV 5/5] END ......C=0.1, gamma=0.1, kernel=rbf;, score=0.778 total time=   6.8s
[CV 1/5] END ..C=0.1, gamma=0.1, kernel=sigmoid;, score=0.722 total time=   6.3s
[CV 2/5] END ..C=0.1, gamma=0.1, kernel=sigmoid;, score=0.723 total time=   6.3s
[CV 3/5] END ..C=0.1, gamma=0.1, kernel=sigmoid;, score=0.723 total time=   6.3s
[CV 4/5] END ..C=0.1, gamma=0.1, kernel=sigmoid;, score=0.723 total time=   6.2s
[CV 5/5] END ..C=0.1, gamma=0.1, kernel=sigmoid;, score=0.723 total time=   6.2s
[CV 1/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.802 total time=  14.6s
[CV 2/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.808 total time=  17.3s
[CV 3/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.803 total time=  20.0s
[CV 4/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.806 total time=  16.7s
[CV 5/5] END ........C=1, gamma=0.1, kernel=rbf;, score=0.809 total time=  15.4s
[CV 1/5] END ....C=1, gamma=0.1, kernel=sigmoid;, score=0.711 total time=   6.4s
[CV 2/5] END ....C=1, gamma=0.1, kernel=sigmoid;, score=0.710 total time=   6.4s
[CV 3/5] END ....C=1, gamma=0.1, kernel=sigmoid;, score=0.710 total time=   6.4s
[CV 4/5] END ....C=1, gamma=0.1, kernel=sigmoid;, score=0.710 total time=   6.3s
```

**Figure [26]**

### 2.5.5 Result Visualizations:

- The Accuracy of the model is given in the below figure [19] which is around 80 percent.

```
Accuracy Score of SVM model is : 0.8071159609621148
```

**Figure [27]**

- Generally if we have two-dimensional input data and a binary classification problem, we can plot the data points as a scatter plot and color-code them by their class. But as we have more than 2 input features it will become difficult to visualize the data so we can create a confusion matrix to see how well the logistic regression model predicts each class. A confusion matrix shows the number of true positives, false positives, true negatives, and false negatives for each class as show in the below **Figure [28]**

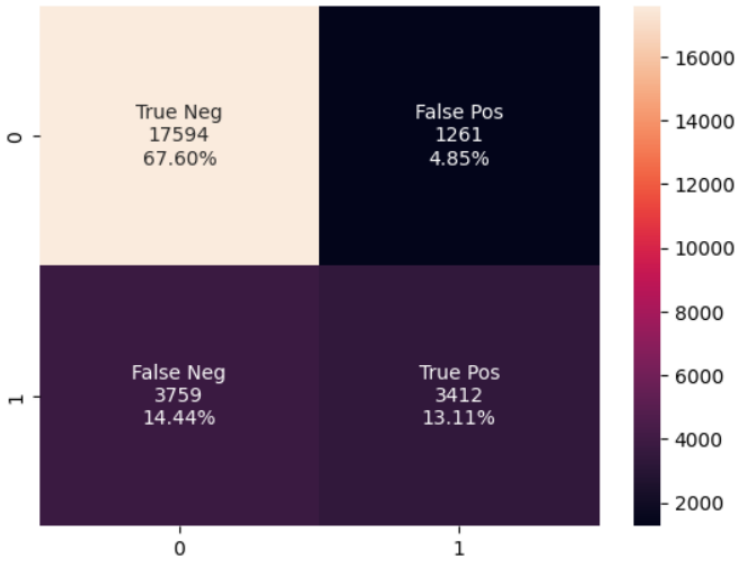


**Figure [28]**

- The performance of a machine learning model used for classification tasks can be assessed using a classification report. It offers a thorough breakdown of numerous metrics that can aid in your understanding of how well your model is doing.
- These parameters, as well as the overall metrics for the model, are included in the classification report for each class in the dataset. It can be a useful tool for analyzing your model's advantages and disadvantages as well as for pinpointing potential areas for performance enhancement. The classification report for my model is given in the below **Figure [29]**

```
Confusion Matrix :
[[17594  1261]
 [ 3759  3412]]
Classification Report :
              precision    recall  f1-score   support

           0       0.82      0.93      0.88     18855
           1       0.73      0.48      0.58      7171

    accuracy                           0.81     26026
   macro avg       0.78      0.70      0.73     26026
weighted avg       0.80      0.81      0.79     26026
```

**Figure [29]**

- The performance of a binary classifier as the discrimination threshold is changed is depicted graphically by a ROC (Receiver Operating Characteristic) curve. The

False Positive Rate (FPR) is plotted against the True Positive Rate (TPR) at various threshold values. It is shown in the below **Figure [30]**

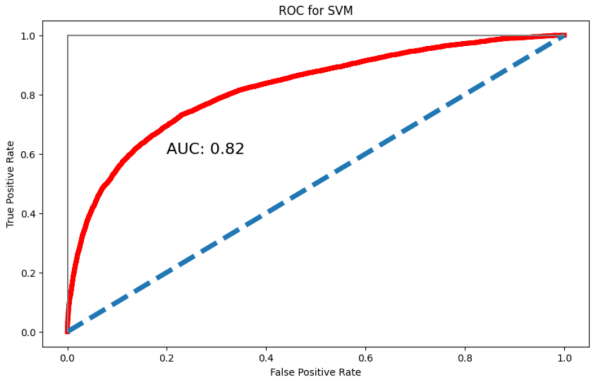


**Figure [30]**

### 2.5.6 Explanation and Analysis.

**Why choose the SVM regression algorithm?**

- We chose SVM as it is effective in handling high-dimensional datasets with many features. They are also robust to noise in the data, making them useful in real-world applications where the data may be incomplete or contain errors.
- Though SVMs are used mostly for multi classification problems they are specifically designed for binary classification problems such as our use case.

**Work we had to do to tune and train the model:**

- As Explained in the sections 1.1, 2.5.2, 2.5.3 and 2.5.4 we had to preprocess the data for the logistic regression model, find an ideal set of hyper parameters so as to improvise the computation of logistic regression.

**Analysis of Effectiveness of the Algorithm:**

- Accuracy is a good method to judge how effective an algorithm is working. The logistic model gave an accuracy of nearly 80 % given cancellation prediction dataset implying the model is performing well.
- From the confusion matrix in figure 28 we could see that True Negatives: 17594 , True Positives: 3412, False Positives: 1261 and False negative: 3759. Since TP + TN = 21006 is much greater than FP + FN = 5020 implying the model is effective.
- From the classification report in Figure 29 we could see the precision, recall, f1-score and support of both the classes (0 - not canceled, 1- canceled). For the class 0 the model is performing well as precision, recall, f1-score are close to 1 and has high support with good number of instances for the class. We have a f1 score of 0.88 for class 0 and 0.58 for class 1 which indicates that the model performance is better for class 0 rather than class 1. But if we see the class 1

metrics precision, recall they are not so good especially recall. Recall is generally the fraction of true positive instances that are correctly identified by the model. So we can say the model is a little bit behind on identifying true positive instances as there is less support for the class 1 implying less number of instances are there. To improve we need to have more data on cancelation data.

From ROC we can observe the trend of True positive rate and false positive rate (TPR & FPR). From the figure 30 we can observe that the true positive rate is higher for small values of false positive rate till FPR reaches 0.2. As the ROC curve in figure 30 is close to the upper left corner of the plot, indicating a high TPR and low FPR for a wide range of threshold values and also has a good AUC score of 0.82 which is close to 1 indicating the model is performing well in predicting/classifying cancellations.

**Conclusion:**

Out of the 5 algorithms used Random Forest got the highest accuracy with 81% because the algorithm's result is achieved by majority voting of multiple minor models (decision trees) results so the effect of noisy data in the algorithms is very less. So we are getting good accuracy.

**References:**


KNN:

1. https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb4
2. Lecture slides by Erik Mikida.

Naive Bayes:

1. Lecture slides by Erik Mikida.
2. https://www.kaggle.com/code/gautigadu091/categorical-naive-bayes-from-scratch-in-python/notebook

Decision Tree :

1. Devanshi Srivastava, Programs Buzz, Decision Tree: Introduction, 06/04/2021

Random Forest :

1. Chengyou Chen, Ensemble Learning, Lecture Slides, 10/11/2022.
2. James Thorn, A summary of the Basic Machine Learning models, 15/02/2021

SVM:

1. https://medium.com/@myselfaman12345/c-and-gamma-in-svm-e6cee48626be
2. https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

Linear regression:

1. https://www.datacamp.com/tutorial/understanding-logistic-regression-python
2. https://www.statology.org/plot-logistic-regression-in-python/


Data source link:
https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-02-11/hotels.csv