**CHAPTER 4: RESULTS AND DISCUSSION**

**4.1 Experimentation setup**

In this project, I have done research on the hotel reservation cancellation dataset and developed machine learning algorithms on oversampled and under-sampled data separately to answer the research questions and hypothesis as well. All this work is done in Python Jupyter Notebook. Below are the details of the experimentation setup for this project.

|  |  |
| --- | --- |
| **System Specifications** | **Configuration Details** |
| OS | 64-bit Windows 11 OS |
| RAM | 8 GB |
| Processor | 11th Gen Intel(R) Core(TM) i9-11900H |
| Clock Speed | up to 3.90 GHz |
| Python | 3.10 |
| Python Libraries | NumPy, Pandas, Matplotlib, Librosa, sklearn, imlearn |

Table 2: Experimentation Setup for this Research

**4.2 Checking for missing data**

As mentioned in the previous chapter, *DataFrame.Columns.isnull()* function and *DataFrame.info()* will give a clear picture of the null values available in the data. In this research, after running the code line ***d.info()*** where ***d*** is the data frame with stored data of the hotel reservation cancellation dataset.All the columns in the data frame consist of 36275 rows without any missing values as shown in Fig 12.

Whereas to verify again for null values in every column, ***d.columns.isnull()*** is executed with the following results shown in Fig 11.

A screenshot of a computer

Description automatically generated

Fig 11: Missing values check using *d.columns.isnull()*

A screenshot of a computer

Description automatically generated

Fig 12: Missing values check using *d.info()*

There are no missing values in the dataset.

**4.3 Data Separation Results**

After data pre-processing and one-hot encoding, the data frame consists of 36275 rows and 27 columns. This dataset is split two times. Initially, data is separated into 85% train data with (30833, 27) and 15% into validation data with (5442, 27). Secondly, validation is split into two parts i.e., validation (80% of (5442, 27)) and test data (20% of (5442, 27)). Overall, below is the structure of the data before applying sampling techniques.

***Train data:*** (30833, 26)

***Train target:*** (30833,)

***Validation data:*** (4353, 26)

***Validation target:*** (4353,)

***Test data:*** (1089, 26)

***Test target:*** (1089,)

**4.4 Data Visualization Results**

Data visualization is important for this project to observe the data behavior. Firstly, the bar graphs are plotted for the object-type columns which are categorical as shown in the below figure.

A graph of a number of people

Description automatically generated with medium confidence

Fig 13: Count of categories in *room\_type\_reserved, market\_segement\_type* columns.

In Fig 13, it is observed that room type 1 is the most chosen room 25000 people followed by room type 4 as the second preferred by nearly 6000 people. The remaining room types are the very least chosen.

In the second graph, the market segment type through which the customer has made the booking. It is observed that online and offline booking are more with online bookings over 25000 and 10000 offline bookings followed by bookings through the corporate segment with nearly 2000.

Overall, it is identified that most customers have chosen to book online and offline with room type 1 as most preferred.

It is important to know, how many people have canceled and not canceled the reservation with respect to each room type. Fig 14 has shown that, in room\_type\_1 total of 9072 bookings were cancelled and 19058 bookings were not cancelled. Whereas 2069 people preferred to cancel the reservation who booked room type 4 and 3988 people decided to stay.

A graph of a number of people

Description automatically generated with medium confidence

Fig 14: Cancellation Status with respect to room type reserved.

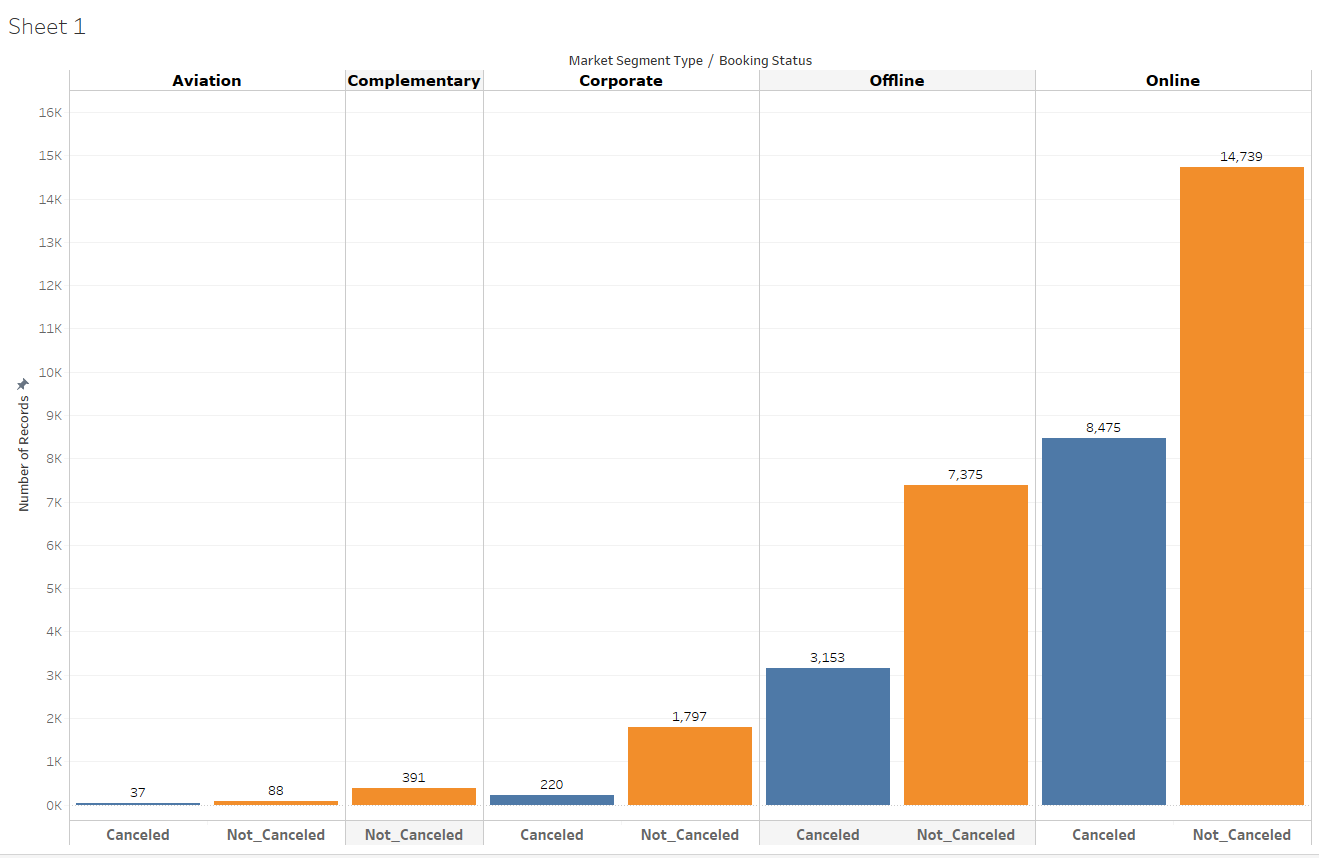


Fig 15: Cancellation Status with respect to market type segment.

Whereas with respect to the market type segment, 8475 people have canceled the online reservation which is nearly 50% compared to 14739 people who didn’t cancel the reservation made online. Offline cancellations are nearly 30% with 3153 and 70% of reservations made through the offline segment are not cancelled as shown in Fig 15.

It is observed that nearly 25% of people who booked room\_type\_1 online have preferred to cancel the reservation made in the hotel.

A group of blue and white bars

Description automatically generated

Fig 16: bar plots columns with int datatype – part1

From the above Fig 16, it is clearly shown that most people have opted for no car parking space required and many of them are adults with no children they also prefer to stay 1 to 3 weeknights on average, and some are preferred to stay 1 to 2 weekend nights as well.

A group of blue bars

Description automatically generated

Fig 17: bar plots columns with int datatype – part2

Many people have booked cancellations for almost all the months, but peak-time bookings are between August, September, and October. Almost 97% of the people are not repeat guests and don’t have any previous cancellations. And also nearly 15000 have made the special requests at the time of booking.

There are two more important columns that need to be analyzed as I personally feel, they might be the reason to cancel the reservation. i.e. Average price per room and lead time columns.

A graph with numbers and a blue bar

Description automatically generated

Fig 18: Histogram of Average price per room column

Fig 18 represents the price ranges on the x-axis and the number of records falling into those average price ranges. There are 20645 rows that are falling in the price range between 60 to 100 and 12570 people have booked the room with much higher prices ranging between 101 to 160. The least price range of a room is from 0-59.

A blue graph with numbers and a white background

Description automatically generated

Fig 19: Histogram of Lead Time column

There are only, 15803 rows which equivalent to 43% of the records are falling in the range of 0 to 50 days as lead time. Whereas 19% of the data accompanies to fall in the range of 50 to 90 days, this seems an acceptable range as people try to plan it before 3 months. But the remaining 38% of the data falls between the range 90 to 450 days which is very long and there is a high chance of cancellations due to a longer gap between the reservation time and arrival time.

The target variable is also available with two classes as shown in the figure below with a huge imbalance in the data.

A green and red bar chart

Description automatically generated

Fig 20: Counts of the Booking Status

In the booking status column, there are 24390 records that are Not Cancelled, and 11885 rows are canceled rows. The dataset is quite imbalanced, the sampled techniques have already been discussed to over this problem.

This is the overall interpretation of the visualizations obtained during the experiment.

**4.5 Machine Algorithms Results on Over-Sampled Data**

After separating the data, the training data is passed to the Oversampling technique to generate the data in between the records using the SMOTE function. The operation of SMOTE is already explained clearly in the previous chapter. After passing the training data with 30833 rows and 26 columns as independent variables and the target variable with 30833 rows, the SMOTE generated (41584,26) samples for independent data and (41584) samples for the target variable respectively for the minority class and balanced the data.

This is sent to standardization and the resulting shape is the same but with the values with mean 0 and standard deviation 1 for each column. This data is fed to the grid search and machine learning models.

**4.5.1 Logistic Regression**

After passing the standardized over-sampled data to fit the defined logistic regression model, 79.3% of the training accuracy was achieved and the validation accuracy was achieved at 78.15% with the confusion matrix below:

**Confusion Matrix:**

**[[1090 374]**

**[ 577 2312]]**

The classification report for the validation data is as follows:

|  |
| --- |
| **precision recall f1-score support** |
|  |
| Canceled 0.65 0.74 0.70 1464 |
| Not\_Canceled 0.86 0.80 0.83 2889 |
|  |
| accuracy 0.78 4353 |
| macro avg 0.76 0.77 0.76 4353 |
| weighted avg 0.79 0.78 0.78 4353 |

Table 3: Classification report for logistic regression with oversampled data

On the test data, this logistic regression model with oversampled data has achieved 77.68%.

**4.5.2 K-Nearest Neighbours**

The KNN is passed with the below parameters,

*param\_grid = {*

*'n\_neighbors': [3, 5, 7, 9, 11 ],*

*'weights': ['uniform', 'distance'],*

*'algorithm': ['ball\_tree', 'kd\_tree', 'brute']*

*}*

These parameters as passed to grid search with the KNN algorithm have identified the below parameters as the best.

*{'algorithm': 'brute', 'n\_neighbors': 11, 'weights': 'distance'}*

After passing the best hyper-parameters obtained from the grid search, the KNN model with

over-sampled data has achieved 99.38% training accuracy and 86.72% validation accuracy.

Below is the confusion matrix and classification report for the validation data of KNN.

**Confusion Matrix:**

**[[1202 262]**

**[ 300 2589]]**

|  |
| --- |
| **precision recall f1-score support** |
|  |
| Canceled 0.80 0.82 0.81 1464 |
| Not\_Canceled 0.91 0.90 0.90 2889 |
|  |
| accuracy 0.87 4353 |
| macro avg 0.85 0.86 0.86 4353 |
| weighted avg 0.87 0.87 0.87 4353 |
|  |

Table 4: Classification report for KNN with oversampled data

On testing the model on unseen data, it resulted in 87.87% accuracy.

**4.5.3 Decision Tree**

Below are the hyperparameters passed to the decision tree grid-search model with oversampled data.

*dt\_params = {*

*'criterion': ['gini', 'entropy'],*

*'max\_depth': [None, 5, 10, 15],*

*'min\_samples\_split': [2, 5, 10],*

*'min\_samples\_leaf': [1, 2, 5]*

*}*

The decision tree's best parameters are *{'criterion': 'entropy','max\_depth': None,*

*'min\_samples\_leaf': 1, 'min\_samples\_split': 2}*

After training with the best params, the decision tree model has achieved 99.46% of training accuracy and 87.08% of validation accuracy. Below is the confusion matrix and classification report for the Decision Tree model on over-sampled data.

**Confusion Matrix:**

**[[1202 262]**

**[ 300 2589]]**

|  |
| --- |
| **precision recall f1-score support** |
|  |
| Canceled 0.80 0.82 0.81 1464 |
| Not\_Canceled 0.91 0.90 0.90 2889 |
|  |
| accuracy 0.87 4353 |
| macro avg 0.85 0.86 0.86 4353 |
| weighted avg 0.87 0.87 0.87 4353 |

Table 5: Classification report for Decision Tree with oversampled data

The unseen test data accuracy achieved by this model is 88.33%

**4.5.4 Random Forest**

Random Forest is the most powerful algorithm in machine learning. The operation is mentioned in previous chapters. The hyper-parameters considered for the grid search is as follows:

*rf\_params = {*

*'n\_estimators': [100, 300, 500],*

*'criterion': ['gini', 'entropy'],*

*'max\_depth': [None, 5, 10, 15],*

*'min\_samples\_split': [2, 5, 10],*

*'min\_samples\_leaf': [1, 2, 5]*

*}*

Below are the results after passing the best parameters obtained to the random forest model using oversampled data are:

Training Accuracy: 99.46%

Validation Accuracy: 90.55%

Below is the confusion matrix and classification report.

Confusion Matrix:

[[1239 225]

[ 186 2703]]

|  |
| --- |
| **precision recall f1-score support** |
|  |
| Canceled 0.87 0.85 0.86 1464 |
| Not\_Canceled 0.92 0.94 0.93 2889 |
|  |
| accuracy 0.91 4353 |
| macro avg 0.90 0.89 0.89 4353 |
| Weighted avg 0.91 0.91 0.91 4353 |

Table 6: Classification report for Random Forest with oversampled data

On unseen test data, the random forest has achieved 89.99% accuracy.

**4.5.4.1 Feature Importance Results**

As mentioned in Chapter 2 under section 2.6.4.1, the feature importance will be obtained from the random forest. Below is the image generated by Random Forest for the feature importance of the columns based on the most used column for decision-making.

A bar graph with text

Description automatically generated

Fig 21: Random Forest Feature Importance’s on Oversampled Data.

As expected, lead time and average price per room columns are the top two columns that are very important reasons for the cancellation. Hence, research question 4 is answered with the feature importance.

**4.5.5 Combined Results of All Algorithms on Over-Sampled Data**

Below are the combined results of all the algorithms on the oversampled data with the best parameters obtained from the grid search.

A screenshot of a computer

Description automatically generated

Fig 22: Combined Results of algorithms on over-sampled data

**4.6 Machine Algorithms Results on Under-Sampled Data**

After separating the data, the training data is passed to the Under-sampling technique to reduce the data randomly in between the records of the majority class using the RUS function. After passing the training data with 30833 rows and 26 columns as independent variables and the target variable with 30833 rows, the RUS generated (20082,26) samples for independent data and (20082) samples for the target variable respectively.

This is sent to standardization and the resulting shape is the same This data is fed to the grid search and machine learning models.

**4.6.1 Logistic Regression**

The operation is the same as performed with oversampled data, but here only difference is that under-sampled data is passed to the logistic regression model and evaluated. Below are the performance results.

Training Accuracy: 77.79 %

Validation Accuracy: 77.92%

**Confusion Matrix:**

**[[1107 357]**

**[ 604 2285]]**

Classification Report:

|  |
| --- |
| **precision recall f1-score support** |
|  |
| Canceled 0.65 0.76 0.70 1464 |
| Not\_Canceled 0.86 0.79 0.83 2889 |
|  |
| accuracy 0.78 4353 |
| macro avg 0.76 0.77 0.76 4353 |
| weighted avg 0.79 0.78 0.78 4353 |

Table 7: Classification report for Logistic Regression with under-sampled data

The testing accuracy achieved on unseen data is 78.23%

**4.6.2 K-Nearest Neighbours**

The same parameters are passed to KNN as in the oversampled section, the only difference is that under-sampled data is passed this time. Below are the results of the KNN on under-sampled data.

Training Accuracy: 99.26%

Validation Accuracy: 84.58%

**Confusion Matrix:**

**[[1259 205]**

**[ 466 2423]]**

Below is the classification report for KNN with under-sampled data:

|  |
| --- |
| **precision recall f1-score support** |
|  |
| Canceled 0.73 0.86 0.79 1464 |
| Not\_Canceled 0.92 0.84 0.88 2889 |
|  |
| accuracy 0.85 4353 |
| macro avg 0.83 0.85 0.83 4353 |
| weighted avg 0.86 0.85 0.85 4353 |

Table 8: Classification report for KNN with under-sampled data

The Testing Accuracy is 84.38% on unseen data for the KNN model trained on under-sampled data.

**4.6.3 Decision Tree**

The decision tree is also passed with under-sampled data in this section, below are the performance results of the Decision Tree model.

Training Accuracy: 91.46 %

Validation Accuracy: 85.38%

**Confusion Matrix:**

**[[1275 189]**

**[ 447 2442]]**

|  |
| --- |
| **precision recall f1-score support** |
|  |
| Canceled 0.74 0.87 0.80 1464 |
| Not\_Canceled 0.93 0.85 0.88 2889 |
|  |
| accuracy 0.85 4353 |
| macro avg 0.83 0.86 0.84 4353 |
| weighted avg 0.87 0.85 0.86 4353 |

Table 9: Classification report for Decision Tree with under-sampled data

The test accuracy of the decision tree model is 85.21%.

**4.6.4 Random Forest**

The random forest achieved the below results on training with the under-sampled data.

Training Accuracy: 97.92%

Testing Accuracy: 89.34%

**Confusion Matrix:**

**[[1286 178]**

**[ 286 2603]]**

|  |
| --- |
| precision recall f1-score support |
|  |
| Canceled 0.82 0.88 0.85 1464 |
| Not\_Canceled 0.94 0.90 0.92 2889 |
|  |
| accuracy 0.89 4353 |
| macro avg 0.88 0.89 0.88 4353 |
| weighted avg 0.90 0.89 0.89 4353 |

Table 10: Classification report for random forest with under-sampled data

The trained random forest on under-sampled data has achieved 89.25% on test data.

**4.6.4.1 Feature Importance Results**

Below is the image of the feature importance generated by the random forest.

A graph of a number of people

Description automatically generated

Fig 23: Random Forest Feature Importance’s on under-sampled Data.

If we observe, the most important features are lead time and the average price per room same as in oversampled data. However, the remaining features have changed their order in under-sampled feature importance. The answer to research question 4 remains the same.

**4.6.5 Combined Results of All Algorithms on Under-Sampled Data**

Below are the combined results of the algorithms performed using under-sampled data.

A screenshot of a computer

Description automatically generated

Fig 24: Combined Results of algorithms on under-sampled data

**4.7 Comparison of Algorithms Results on Over-Sampled and Under-Sampled Data**

Below are the results tabulated for training, validation, and testing accuracy on oversampled and under-sampled data for all the machine learning algorithms used.

|  |  |  |
| --- | --- | --- |
|  | **Training Accuracy**  **(Over Sampled)** | **Training Accuracy**  **(Under Sampled)** |
| **Logistic Regression** | ***79.30*** | ***77.79*** |
| **K-Nearest Neighbours** | ***99.38*** | ***99.26*** |
| **Decision Trees** | ***99.46*** | ***91.46*** |
| **Random Forest** | ***99.46*** | ***97.92*** |

Table 11: Training accuracy comparison of algorithms on oversampled and under-sampled data

From the above table, it is observed that, on oversampled data, the decision tree model and random forest have trained equally with the highest accuracy compared to other algorithms. In under-sampled training, performances are very low comparatively.

|  |  |  |
| --- | --- | --- |
|  | **Validation Accuracy**  **(Over Sampled)** | **Validation Accuracy**  **(Under Sampled)** |
| **Logistic Regression** | ***78.15*** | ***77.92*** |
| **K-Nearest Neighbours** | ***86.72*** | ***84.58*** |
| **Decision Trees** | ***87.08*** | ***85.38*** |
| **Random Forest** | ***90.55*** | ***89.34*** |

Table 12: Validation accuracy comparison of algorithms on oversampled and under-sampled data

On validation data, clearly, the winner is the random forest with 90.55 % on oversampled and 89.34% on under-sampled data. Random forest algorithm trained on oversampled data has given the highest accuracy on the validation data.

|  |  |  |
| --- | --- | --- |
|  | **Testing Accuracy**  **(Over Sampled)** | **Testing Accuracy**  **(Under Sampled)** |
| **Logistic Regression** | ***77.68*** | ***78.23*** |
| **K-Nearest Neighbours** | ***87.87*** | ***84.38*** |
| **Decision Trees** | ***88.33*** | ***85.21*** |
| **Random Forest** | ***89.99*** | ***89.25*** |

Table 13: Testing accuracy comparison of algorithms on oversampled and under-sampled data

On comparing the performances of various machine learning algorithms on oversampled and under-sampled data, it is observed that Random Forest has performed the best in both cases. But Random Forest on Oversampled data has given the training accuracy of 99.46% Validation accuracy of 90.55% and Testing Accuracy is 89.99%.

All the algorithms trained on the oversampled data have achieved better performance on validation and test data compared to the algorithms trained on the under-sampled data. Hence, the “**Null Hypothesis has been rejected”.**