**7COM1039-0109-2022**

Advanced Computer Science Masters Project

Prediction of hotel reservation cancellation of a customer with reservation details using Machine Learning

**Interim Progress Report (IPR)**

**Student Id :** 20067093

**Student Name :** Nikhil Reddy Marella

**Supervisor :** John Lones

**Contents**

1. Background research 2
2. Project Plan 5
   * 1. Gantt Chart for Project Plan 7
3. Summary of progress to date 8
   * 1. Idea Development, Data Gathering, and Understanding 8
     2. Literature Research 8
     3. Data Preparation and Data Visualization 9
     4. Model Building 9
4. Consideration of ethical, legal, professional, and social issues 10

**Appendices** 11

**References** 18

1. **Background Research**

This project is titled “Prediction of hotel reservation cancellation of a customer with reservation details using machine learning”. The dataset for this project is available on Kaggle i.e. <https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset>.

The dataset consists of 36275 rows and 19 columns.

The aim of this project is to predict the customer who is going to cancel the hotel reservation based on the details given at the time of booking in the respective hotel. The data is readily available, but the task here is to understand the columns and their interpretation to figure out which columns are important for the analysis. Later, observe the patterns and relations between the columns to identify the reasons behind the cancellations of hotel reservations and develop a machine learning algorithm that can predict the customer who might cancel the reservation based on the customer details provided. And also, there is some interest in knowing the solution for the hypothesis “Is the customer cancelling hotel reservation due to the average price change of the same room type in between the lead time?”

Starting with understanding the columns with the description as given in [1], and then gaining information about the reasons behind the cancellations because hotels aim to make a profit by providing the services. If the customers cancelling the prebooked hotel rooms, it might affect the hotel’s reputation among the customers and will lead to a loss for the hotel [2]. According to the estimate given in [2], Since 2014, cancellation costs across all platforms have increased by 6%, reaching an estimated 40% in 2018 [2]. To reduce the economic loss for the hotels, making use of the latest technologies to automate the process to find the probability of the customer cancelling the reservation based on the information given. So that hotels can decide whether the reservation should be given to the customer or not.

After understanding the complete description behind the dataset, the missing value from the data has to be cleaned after that data visualization has to be plotted using the libraries discussed in section B of [2]. But learning how to plot the histograms and bar plots has been learned from the sources [3][4].

Before the visualization part, there is one challenge to dealing with categorical values in two columns “room\_type\_reserved” and “market\_segment\_type”. These columns should be converted into number format for the computers to process the categorical data [5], it can be done by using a technique called one-hot encoding [5]. As discussed in [5], “One-hot Encoding is a type of vector representation in which all the elements in a vector are 0, except for one, which has 1 as its value, where 1 represents a boolean specifying a category of the element.”

Later the splitting the data is done using train\_test\_split function from the sklearn library, but code for this block has been referred from [7] which clearly explains the procedure to implement.

Standardization has been applied to all the columns in training data, testing data and unseen data.

One thing this project wants to make sure of is to play with the hyperparameters using cross-validation techniques to achieve the best parameters which give the best result. GridSearchCV [7] has been used to experiment with some selected parameters and fitted to machine learning algorithms selected as baseline models (Random Forest Classifier as first priority and Decision Tree Classifier as a second option).

**2)** **Project Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Prediction of hotel reservation cancellation of a customer with reservation details using Machine Learning** | | | | | |
|  |
|  | **University of Hertfordshire, Nikhil Reddy Marella, 20067093** | | | | |  |
|  |  |  |  |  |  |  |
|  | **Milestone Description** | **Progress** | **Start** | **End** | **Days** |  |
| **Project Idea** | Data Source and Project Idea | **100%** | 23/02/2023 | 24/02/2023 | 2 |  |
| Data Understanding | **100%** | 23/02/2023 | 24/02/2023 | 2 |  |
| Hypothesis Identification | **100%** | 25/02/2023 | 25/02/2023 | 1 |  |
| Detailed Project Proposal (DPP Report) | **100%** | 26/02/2023 | 02/03/2023 | 5 |  |
| DPP Submission | **100%** | 07/03/2023 | 07/03/2023 | 1 |  |
| **Literature Background** | Hotel reservation cancellation papers | **100%** | 10/03/2023 | 10/03/2023 | 1 |  |
| Best Classification Algorithms | **70%** | 11/03/2023 | 31/07/2023 | 143 |  |
| RF, DCS, KNN, K-Means, Naïve Bayes | **80%** | 11/03/2023 | 31/07/2023 | 143 |  |
| Cross Validation Techniques | **80%** | 11/03/2023 | 31/07/2023 | 143 |  |
| **Data Cleaning** | Deal with missed values in Columns | **100%** | 17/03/2023 | 17/03/2023 | 1 |  |
| Drop any null value rows | **100%** | 17/03/2023 | 17/03/2023 | 1 |  |
| **Data Preparation and Visualization** | Understanding visualization Plots | **100%** | 18/03/2023 | 18/03/2023 | 1 |  |
| learn code for plotting the charts | **70%** | 19/03/2023 | 19/03/2023 | 1 |  |
| Plot histograms for all columns | **100%** | 20/03/2023 | 21/03/2023 | 2 |  |
| Plot bar plot for categorical columns | **90%** | 20/03/2023 | 21/03/2023 | 2 |  |
| learn code for one hot encoding | **100%** | 22/03/2023 | 23/03/2023 | 2 |  |
| One hot encode on categorical columns | **100%** | 23/03/2023 | 23/03/2023 | 1 |  |
| **Model Building** | Split train and test | **100%** | 24/03/2023 | 24/03/2023 | 1 |  |
| Standardize the data | **100%** | 24/03/2023 | 24/03/2023 | 1 |  |
| Learn DCT and Random Forest theory and code | **100%** | 25/03/2023 | 28/03/2023 | 3 |  |
| GridSearch for Decision Tree Classifier | **80%** | 29/03/2023 | 29/03/2023 | 1 |  |
| Decision tree classifier (Best Params) | **80%** | 29/03/2023 | 29/03/2023 | 1 |  |
| GridSearch for Random Forest Classifier | **80%** | 30/03/2023 | 30/03/2023 | 1 |  |
| Random Forest (Best Params) | **80%** | 30/03/2023 | 30/03/2023 | 1 |  |
| Test unseen data on best model | **80%** | 31/03/2023 | 31/03/2023 | 1 |  |
| Interim Progress Report | **100%** | 15/04/2023 | 25/04/2023 | 11 |  |
| IPR Submission | **100%** | 01/05/2023 | 01/05/2023 | 1 |  |
| Hyperparameter Tuning on ML algorithms |  | 14/05/2023 | 21/05/2023 | 8 |  |
| Performance Comparision |  | 22/05/2023 | 22/05/2023 | 1 |  |
| Test unseen data on best model |  | 22/05/2023 | 22/05/2023 | 1 |  |
| **FPR Report** | Abstract |  | 24/05/2023 | 24/05/2023 | 1 |  |
| Introduction |  | 25/05/2023 | 27/05/2023 | 3 |  |
| Literature Review Search |  | 28/05/2023 | 10/06/2023 | 14 |  |
| Data Analysis and Preparation |  | 12/06/2023 | 20/06/2023 | 9 |  |
| Model Building |  | 21/06/2023 | 30/06/2023 | 10 |  |
| Results |  | 01/07/2023 | 03/07/2023 | 3 |  |
| Conclusion |  | 04/07/2023 | 04/07/2023 | 1 |  |
| Future Work |  | 05/07/2023 | 05/07/2023 | 1 |  |
| References |  | 05/07/2023 | 05/07/2023 | 1 |  |
| Appendices |  | 06/07/2023 | 06/07/2023 | 1 |  |
| Plagiarism Check |  | 07/07/2023 | 07/07/2023 | 1 |  |
| Final Edits and Review |  | 08/07/2023 | 15/07/2023 | 8 |  |
| Final Plagiarism Check |  | 16/07/2023 | 16/07/2023 | 1 |  |
| Meet with Supervisor |  | 17/07/2023 | 17/07/2023 | 1 |  |
| FPR Submission |  | 28/08/2023 | 28/08/2023 | 1 |  |

* + 1. **Gantt Chart for Project Plan**

**Chart

Description automatically generated**

**3) Summary of Progress to Date**

**A) Idea Development, Data Gathering, and Understanding**

From the Kaggle source, this project has been selected because of its unique and the latest updated version available with more rows and proper description. The idea of “Prediction of hotel reservation cancellation of a customer with reservation details using machine learning” has been developed with an interesting hypothesis as mentioned in the background section. The data Set has been downloaded from Kaggle and understood the columns description given and analyzed which columns are significant for further data analysis.

**B) Literature Research**

Literature research is the crucial part of the project as this is started from foundations and absolutely has no idea how to do things, with the help of lectures from research methods on how to perform a literature search. A literature search will go on till the end of the project as references are needed every time, so the timeline mentioned in above is followed. This project has progressed with a deep understanding of the dataset with the help of relevant articles to this project idea as mentioned in [2], followed by knowing machine learning classification algorithms and cross-validation techniques. Only 80% of the literature research is done, still might need their help of them while doing more experimentation on the remaining algorithms.

**C) Data Preparation and Data Visualization**

The available data consists of 36275 entries and 19 columns at the start. But there are a few columns that might not add value to the analysis, so the columns “Booking\_ID”, “type\_of\_meal\_plan” has been removed from the data and one-hot encoding has been applied for two categorical columns.

To understand the patterns of the data in every column, a histogram has been plotted. It is identified that lead time and average price room columns have great relationships and might be a reason for a reservation cancellation, but this is just the assumption made from the visualization detailed inspection is yet to be done. Output variable Booking Status has some data imbalance which needs to be rectified by finding some solution, its effect on the algorithm performance has to analyzed with the results and then act on it.

**D) Model Building**

Prepared data is split into train and test sets and later all of them are standardized before doing cross-validation for best parameters using GridSearchCV, which is performed on only selected values at random, but it needs to be experimented with a wider range of values after doing some extensive research about hyperparameter tuning. GridSearchCV is applied on both chosen baseline models i.e., Random Forest Classifier and Decision Tree Classifier. Performance results are overwhelming more than expected with the below results.

|  |  |  |
| --- | --- | --- |
|  | **Train Accuracy** | **Test Accuracy** |
| **Decision Tree Classifier** | 0.91308286074354 (**91.3%**) | 0.87531011669576 (**87.5%**) |
| **Random Forest Classifier** | 0.99428954001260 (**99.4%**) | 0.90710282091335 (**90.7%**) |

Table1: Baseline Models Performance Results

Still more experimentation has to be done on other classification algorithms as well along with hyperparameter tuning this will take time.

After the modeling part is done, Final Progress Report has to be started with all the necessary sections as mentioned in the project plan in section 2.

**4) Consideration of ethical, legal, professional, and social issues**

This project has not considered any surveys or any kind of interviews, simply to say primary research has not been considered for this project. The secondary data has been considered from the Kaggle with all the necessary information given. This project is ethically and professionally answerable to the hypothesis question which can help hotel management to predict the customer who can cancel the reservation within the lead time given.

Socially, if the customers cancelling the reservations and if that news spread across people, there might be damage to the hotel’s reputation, which will lead to economic loss. Hence, this project helps hotels prevent social, ethical, and professional issues.

**Appendices**

**Importing Necessary Libraries**

import pandas as pd

import numpy as np

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import accuracy\_score, confusion\_matrix

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

import matplotlib.pyplot as plt

from sklearn.preprocessing import OneHotEncoder

**# Loading the data**

d = pd.read\_csv("Hotel Reservations.csv")

d.head()

Graphical user interface, application, table

Description automatically generated

# Check the misssing data from the dataset in all the columns, if there are any null values, isnull() will show that as TRUE.

# If there are no null values, it will show false.

d.columns.isnull()

Since there are no null values in the dataset for 36275 rows across all the 19 columns, data is full.

**Remove Unwanted Columns**

d["type\_of\_meal\_plan"].unique()

d["room\_type\_reserved"].unique()

d["arrival\_year"].unique()

d["market\_segment\_type"].unique()

d = d.drop(columns = ["Booking\_ID","type\_of\_meal\_plan"],axis=1) #Booking\_ID is removed using drop function specifying axis = 1 which means from columns.

D

**One-hot encoding**

data\_encoded = pd.get\_dummies(d,columns=['room\_type\_reserved','market\_segment\_type'])

data\_encoded.info()

data\_encoded.head()

#poping target column inbetween the columns and adding it to the last

booking\_status\_col = data\_encoded.pop("booking\_status")

data\_encoded["booking\_status"] = booking\_status\_col

data\_encoded

data\_encoded["booking\_status"].value\_counts()

**Data Visualization**

# histograms for all columns

for col in data\_encoded.columns[:-1]:

plt.hist(data\_encoded[col])

plt.title(col)

plt.show()

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart, histogram

Description automatically generatedChart, bar chart

Description automatically generated

#Bar Plot for output Variable

output\_counts = data\_encoded[data\_encoded.columns[-1]].value\_counts()

plt.bar(output\_counts.index, output\_counts.values)

plt.title(data\_encoded.columns[-1])

plt.show()

Chart, bar chart

Description automatically generated

**Data Splitting**

train\_data, test\_data, train\_target, test\_target = train\_test\_split(

data\_encoded.iloc[:, :-1], data\_encoded.iloc[:, -1], test\_size=0.3, random\_state=42)

unseen\_data = data\_encoded.iloc[:, :-1].sample(frac=0.1, random\_state=42)

**Standardization**

scaler = StandardScaler()

train\_data = scaler.fit\_transform(train\_data)

test\_data = scaler.transform(test\_data)

unseen\_data = scaler.transform(unseen\_data)

**Cross Validation using Grid Search for Decision Tree Classifier**

dt\_params = {

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

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

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

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

}

dt\_clf = DecisionTreeClassifier(random\_state=42)

dt\_grid\_search = GridSearchCV(dt\_clf, dt\_params, cv=5, n\_jobs=-1, verbose=1)

dt\_grid\_search.fit(train\_data, train\_target)

dt\_best\_params = dt\_grid\_search.best\_params\_

**dt\_best\_params**

**Output:**

{'criterion': 'entropy',

'max\_depth': 15,

'min\_samples\_leaf': 1,

'min\_samples\_split': 2}

# Train decision tree classifier with best parameters and measure performance

dt\_clf = DecisionTreeClassifier(\*\*dt\_best\_params, random\_state=42)

dt\_clf.fit(train\_data, train\_target)

train\_pred = dt\_clf.predict(train\_data)

test\_pred = dt\_clf.predict(test\_data)

print('Decision Tree Classifier:')

print('Train Accuracy:', accuracy\_score(train\_target, train\_pred))

print('Test Accuracy:', accuracy\_score(test\_target, test\_pred))

print('Confusion Matrix:\n', confusion\_matrix(test\_target, test\_pred))

Output:

Text

Description automatically generated

**Cross Validation using Grid Search for Random Forest Classifier**

rf\_params = {

'n\_estimators': [100, 200, 500],

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

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

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

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

}

rf\_clf = RandomForestClassifier(random\_state=42)

rf\_grid\_search = GridSearchCV(rf\_clf, rf\_params, cv=5, n\_jobs=-1, verbose=1)

rf\_grid\_search.fit(train\_data, train\_target)

rf\_best\_params = rf\_grid\_search.best\_params\_

# Train random forest classifier with best parameters and measure performance

rf\_clf = RandomForestClassifier(\*\*rf\_best\_params, random\_state=42)

rf\_clf.fit(train\_data, train\_target)

train\_pred = rf\_clf.predict(train\_data)

test\_pred = rf\_clf.predict(test\_data)

print('Random Forest Classifier:')

print('Train Accuracy:', accuracy\_score(train\_target, train\_pred))

print('Test Accuracy:', accuracy\_score(test\_target, test\_pred))

print('Confusion Matrix:\n', confusion\_matrix(test\_target, test\_pred))

Output:

Text

Description automatically generated

# Use the trained model to make predictions on the unseen data

unseen\_pred = rf\_clf.predict(unseen\_data)

print('Unseen Data Accuracy:', accuracy\_score(data\_encoded.iloc[:, -1].sample(frac=0.1, random\_state=42), unseen\_pred))

# Plot feature importances for the random forest classifier

feat\_importances = pd.Series(rf\_clf.feature\_importances\_, index=data\_encoded.columns[:-1])

feat\_importances.nlargest(10).plot(kind='barh')

plt.title('Feature Importances')

plt.show()

Output:

Chart, bar chart

Description automatically generated

# As per the hypothesis, Lead time and Average price per room are the most important features and are responsible for reservation cancellation.

**REFERENCES**

[1] www.kaggle.com. (2023). *Hotel Reservations Dataset*. [online] Available at: <https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset>.

[2] R. Prabha, Senthil, G.A., Nisha, S Snega, L Keerthana and S Sharmitha (2022). Comparison of Machine Learning Algorithms for Hotel Booking Cancellation in Automated Method. *2022 International Conference on Computer, Power and Communications (ICCPC)*. doi:https://doi.org/10.1109/iccpc55978.2022.10072135.

[3] Gilbert Tanner. (2019). *Introduction to Data Visualization in Python*. [online] Available at: <https://gilberttanner.com/blog/introduction-to-data-visualization-inpython/>.

[4] GeeksforGeeks. (2021). *Data Visualization with Python*. [online] Available at: <https://www.geeksforgeeks.org/data-visualization-with-python/>.

[5] Stack Abuse. (2020). *One-Hot Encoding in Python with Pandas and Scikit-Learn*. [online] Available at: <https://stackabuse.com/one-hot-encoding-in-python-with-pandas-and-scikit-learn/>.

[6] Brownlee, J. (2020). *Train-Test Split for Evaluating Machine Learning Algorithms*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/>.

‌[7] Scikit-learn.org. (2019). *sklearn.model\_selection.GridSearchCV — scikit-learn 0.22 documentation*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html>.