Flight fare prediction project

Submitted by:

Leena Chatterjee

ACKNOWLEDGMENT

I wish to express my sincere gratitude to Data trained Academy and FlipRobo Technologies who gave me the opportunity to do this Project. It helped me to do a lot of research and I have grasped many new things. The data source is internet, we fetched data through scrapping. The training I have taken from Data trained, Kaggle and GitHub helped me to complete this project

INTRODUCTION

Business Problem Framing:

Airline companies use complex algorithms to calculate flight prices given various conditions present at that particular time. These methods take financial, marketing, and various social factors into account to predict flight prices.

Nowadays, the number of people using flights has increased significantly. It is difficult for airlines to maintain prices since prices change dynamically due to different conditions. That's why we will try to use machine learning to solve this problem. This can help airlines by predicting what prices they can maintain. It can also help customers to predict future flight prices and plan their journey accordingly.

Conceptual Background of the Domain Problem:

Data science comes as a very important tool to solve problems. In this project we tried to predict ticket fared based on some features. Usually as per the scenario we buy tickets at the last moment before travelling it cost more and vice versa. This data will help us to predict Fare, where we will see the drastic of fare based on purchase date

Goals of the Study:

The main objectives of this study are as follows:

- To apply data pre-processing and preparation techniques in order to obtain clean data
- To build machine learning models able to car price based on features
- To analyse and compare models performance in order to choose the best model

Motivation for the Problem Undertaken:

Learning the theoretical background for data science or machine learning are often a frightening experience, because it involves multiple fields of mathematics and an extended list of online resources. By proper practical research and practice I can become better in this field. These suggestions are derived from my mentors/SME's and my own experience in the beginner projects.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

- In the project, we need to predict fare of flight using the provided features in the dataset.
- In the project we have used classifications target is object type data ,
- There are no outliers in the dataset although all the column being object so we did not remove skew or outlier
- Describe method is used to display all the summary statistics of the dataset.
- In Model building we split the dataset into train and test data and is scaled. The scaled data is passed into the model for training and results are predicted.

Data Sources and their formats:

The Data source refers to Flight prediction project. The dataset is in a csv(comma separated values) format. The whole dataset contains 1250 rows and 7 columns. The following are the features with their description in detail:

df.columns:

```
Index(['FlightNAME', 'Date', 'Arr_timr', 'dep_time', 'Fare', 'Stop', 'Time'],
 dtype='object')
```

1.Flight name -Some customer having personal choice while travelling so we have taken this column as input features , some cases it plays vital role , while users buy ticket

2.Date This is the Major features variable . As mentioned earlier as well If date is near to the booking date the fare will increase

3.Arr_time – some users avoid late arrival fight to destination city , on that prospective it is very much important

4.Dep_time-Usally late night or ear morning departure time flight cheap. It is usually avoid by Family travelling

5.Fare –This is the target variable we need to predict Fare based on the features available

Stop- Usually Non-stop flight is preferable by all users

Data Pre-processing Done:

Data Pre-processing is defined as the process of preparing the data and making it suitable for a machine learning model. It's the primary and crucial step while creating a machine learning model. When creating a machine learning project, it's not always a case that we encounter the clean and formatted data. And while doing any operation with data, it's mandatory to clean it and put during a formatted way. So for this, we use data pre-processing task.

In simple terms, Data pre-processing is a data mining technique which is used to transform the raw data in a useful and efficient format. In the present project, data pre-processing steps are as follows: \

- **Collection of Data**: The data collected is in the form of a csv file.

- **Importing the required Libraries**: Initially we import the libraries like NumPy, pandas and matplotlib. These libraries are well explained in the below section.

- **Importing the Dataset**: This is also known as loading/reading the data which can be done by using Pandas. Dataset is read by giving the right path where it is located.

- **Identifying the missing values and Handling them**: In the given dataset, we do not have missing values in most of the columns.

- **Encoding the categorical data**: This section is of converting Categorical columns to Numerical columns since, model takes only numerical values. There are varieties of encoding techniques. Here, we use Label Encoding to convert the categorical columns. Before using Label Encoding, we import label encoder from sklearn.preprocessing library. In the given dataset, we have nearly most of the features as categorical (object type) columns.

```
for i in df1.columns:
    df[i]=le.fit_transform(df[i])
```

Data Inputs- Logic- Output Relationships:

In the given dataset, label is the output feature and all the remaining features can be taken as input. The relationship between the input and output variables are found by correlation. In general, the correlation can be found using heatmap in which all the relationship between the feature variables can be observed clearly. We can also use different plots by giving the target variable against the remaining columns.

Hardware and Software Requirements and Tools Used:

Hardware:HP Laptop, Intel Core i5 Processor, 8th Generation

Software: The complete project is done using Jupyter Notebook.

Libraries and Packages used:

Initially we load the basic libraries such as pandas, seaborn and matplotlib

NumPy: NumPy is the library used for scientific computing in python. It is also termed as numerical python.

NumPy is imported as "import numpy".

Pandas: Pandas is mainly used for data analysis. It allows importing data from various file formats.

The Pandas library is imported as "import pandas"

Seaborn: Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Matplotlib: Matplotlib is a visualization library in Python for 2D plots of arrays.

sklearn.preprocessing:The sklearn.preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators. In general, learning algorithms benefit from standardization of the data set.

LabelEncoder: LabelEncoder library is used to convert categorical columns to numerical columns. It is imported from sklearn.preprocessing package.LabelEncoder is imported as

from sklearn.preprocessing import LabelEncoder

StandardScaler: StandardScaler is used to standardize the data to a single scale and it is imported as

from sklearn.preprocessing import StandardScaler

SciPy: SciPylibrary includes modules for linear algebra, integration, optimization, and statistics. Skew is used to treat the skewness in the dataset. It is imported from scipy.stats library in the following way:

from scipy.stats import skew

StandardScaler: It is used to scale all the values of each feature such that its distribution will have a mean value 0 and standard deviation of 1. It is imported from sklearn. pre-processing library.

from sklearn.preprocessing import train_test_split: Used in breaking our input and target variable into train and test data. In the project we have split train and test into 80:20 respectively. It is imported from sklearn.model_selectionlibrary.

from sklearn.model_selection import train_test_split

Performance metrics:

All the classification model's performance is found by Accuracy score , classification matrix

As some of these libraries are frequently used, we write these in short form as following:

import numpy as np import
pandas as pd import matplotlib.pyplot
as plt import seaborn as sns

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods) The whole problem-solving approach includes the following steps:

- **Problem Framing:** It includes understanding the problem that is whether the problem is of regression or classification. The present project is of regression type. As mentioned Earlier price cannot be categorial

- **Data Understanding:** Data understanding means having an intimate grasp of both the distributions of variables and the relationships between variables. It also includes summary statistics and data visualization.

- **Data Cleaning:** The process of identifying and repairing issues with the data is termed as data cleaning. Statistical methods are used for data cleaning. Some of them are outlier detection and imputation. There are many outliers in the present dataset, but as being categorial column we did not remove those

- **Data Selection:** The process of reducing the scope of data to those elements that are most useful for making predictions is called data selection.

- **Data Preparation:** It includes the data to identify the features to be selected and removed. We converted obj data to numeric.

- **Model Evaluation:** Model evaluation consists of identifying input and output variables and splitting the dataset into train and test datasets. The output variable is “Fare” and the remaining features are input variables. In this project, we have split into 80:20 train and test respectively.
- **Model Configuration:** Hyperparameter tuning the models will get the best fit parameters of each and every model. In this project we use GridSearchCV for knowing the best fit parameters.
- **Model Selection:** The process of selecting one method as the solution is called model selection. It includes the regression model performance metrics.

Testing of Identified Approaches (Algorithms):

The Machine Learning Algorithms used in this project for training and testing to predict the prices of the houses are namely:

Logistic Regression
Decision Tree classification

Random Forest Classification
KMN Classification

Linear Regression:

The logistic classification model (or logit model) is a **binary classification model** in which the conditional probability of one of the two possible realizations of the output variable is assumed to be equal to a linear combination of the input variables, transformed by the logistic function.

Decision Tree Classifier :

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

Random Forest Classifier:

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

trees.

Run and Evaluate selected models:

The above-mentioned algorithms have been run in the jupyter notebook and the performance metrics is found .

To check whether the model is overfitting/underfitting GridSearchCV is used

We have chosen the best acc_score by running the random state of 42 to 100

Observations from modelling:

- Target variable is selected as “Fare”
- Split the dataset to 80:20 for train and test respectively.
- Used standard scaler to scale the values
- All the data obtained from above steps given to model and made the predictions.
- Observed the best fit parameters from GridSearchCV

CONCLUSION:

Key Findings and Conclusions of the Study:

The goal is to achieve the system which will reduce the human effort to find a car price. The proposed system Flight price prediction model approximately tries to achieve the same one. We have managed how to prepare a model that gives users for a best approach with future lodging value predictions.

Limitations of this work and Scope for Future Work:

- **Limitations:** Datas are limited

We found after all analysis RandomForest Classifier is the best model and 'criterion': 'Entropy', 'max_depth': 15, 'n_estimators': 8 we got the final model which accuracy is 99.9540419237838

```
Finalmodel=RandomForestClassifier(criterion= 'entropy', max_depth= 15, n_estimators = 8, max_leaf_nodes= 6)
Finalmodel.fit(x_train, y_train)

pred=Finalmodel.predict(x_test)
acc=accuracy_score(y_test,pred)

print(acc*100)
```

```
87.17171717171716
```

```
import joblib
joblib.dump(Finalmodel,"Flight.pkl")
```

```
['Flight.pkl']
```