“Flight Price Prediction with Machine Learning”

*Dissertation submitted in fulfilment of the requirements for the Degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

By

**Challuri naveen**

**12001563**

Supervisor

**MR. VED PRAKASH CHAUBEY**



**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

Month - April Year - 2023

@ Copyright LOVELY PROFESSIONAL UNIVERSITY, Punjab (INDIA)

ALL RIGHTS RESERVED

**DECLARATION STATEMENT**

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled "FLIGHT PRICE PREDICTION WITH MACHINE LEARNING” in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mr./Mrs. VEDPRAKASH CHAUBEY research Guide’s Name. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

**Challuri naveen**

**Roll. No:12001563**

**SUPERVISOR’S CERTIFICATE (16 bold)**

This is to certify that the work reported in the B. Tech Dissertation/dissertation proposal entitled “Flight Price Prediction with Machine Learning**”** submitted by **challuri naveen** at **Lovely Professional University, Phagwara, India** is a Bonafede record of his / her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree.

Signature of Supervisor

(Name of Supervisor)

**Date:**

**Counter Signed by:**

1. **Concerned HOD:**

HoD’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HoD Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Neutral Examiners:**

**External Examiner**

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Affiliation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Examiner**

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**TABLE OF CONTENTS**

**CONTENTS PAGE NO.**

Cover Page 1

Declaration Statement 2

Supervisor’s Certificate 3

Table of Contents 4 - 6

List of Equations 7

**CHAPTER 1: INTRODUCTION** 8 - 9

* 1. **Abstract**

**1.2 Problem Statement**

**1.3 Objectives**

**CHAPTER 2: REVIEW OF LITERATURE** 9

**CHAPTER 3: SYSTEM REQUIREMENTS SPECIFICATION** 10 - 11

**3.1 Hardware requirements**

**3.2 Software specification**

**3.3 Software requirements**

**3.3.1 Anaconda distribution**

**3.3.2 Python Libraries**

**CHAPTER 4 – DATA COLLECTION FOR FLIGHT TICKET PRICE PREDICTION**

**4.1 Gathering Data**

**4.1.1 Open Sources**

**4.1.2 Collection by Individuals**

**4.1.3 Crowd Sourcing**

**4.2 Importing Necessities**

**CHAPTER 5 : DATA CLEANING FOR FLIGHT TICKET PRICE PREDICTION**

DATA CLEANING

**CHAPTER 6: EXPLORATORY DATA ANALYSIS FOR FLIGHT TICKET PRICE PREDICTION**

6.1 UNIVARIATE ANALYSIS

**6.1.1 Categorical unordered univariate analysis**

**6.1.2 Categorical ordered univariate analysis**

**6.2 Bivariate Analysis**

**6.2.1 Numeric - numeric analysis**

**6.2.2 Numerical - Categorical**

**6.2.3 Categorical - Categorical variable**

**6.4 Multivariate analysis**

**6.5 STATISTICAL ANALYSIS**

**CHAPTER 7 : DATA PREPARATION FOR FLIGHT TICKET PRICE PREDICTION**

**7.1 Predicting response variable**

**7.2 Importing libraries**

**7.3 Encoding categorical variables**

**7.4 Standard scaling for the model**

**7.5 Defining a pre-processor (using the Column Transformer class in the scikit-learn library)**

**7.6 fit transform method**

**7.7 Splitting the model into training and testing data sets**

**CHAPTER 8 : PREDICTION USING LOGISTIC REGRESSION** 23 - 25

**8.1 fitting the model and predicting**

**8.2 precision score and recall score**

**8.3 Evaluating the model on Logistic regression**

**8.4 AUC – ROC**

**CHAPTER9: PREDICTINGUSING RANDOMFORESTCLASSIFIER**25**-**26

**9.1 Fitting the model and predicting on Random Forest Classifier**

**9.2 Evaluating the model on Random Forest Classifier**

**CHAPTER 10: RESULTS** 26

**CHAPTER 11: CONCLUSION** 27

**REFERENCES** 27-29

**APPENDIX** 29-31

**CHAPTER1: INTRODUCTION**

**1.1 Abstract**

Flight ticket price prediction is an important area of research in the field of machine learning and data analysis. The aim of this project is to develop a machine learning model that can accurately predict the price of flight tickets based on various factors such as airline, route, date of travel, and other relevant features.

To accomplish this goal, we will collect a large dataset of flight ticket prices from various sources and pre-process the data to ensure that it is clean and consistent. We will then perform exploratory data analysis to gain insights into the dataset and identify any patterns or trends that may exist. Next, we will develop and train several machine learning models using different algorithms such as linear regression, decision trees, random forests, and neural networks.

We will evaluate the performance of these models using various metrics such as mean absolute error, root mean squared error, and R-squared score. Finally, we will select the best-performing model and use it to make predictions on a test dataset. We will also develop a web application that will allow users to input their travel details and receive an estimated price for their flight ticket.

Overall, this project aims to demonstrate the potential of machine learning in predicting flight ticket prices and provide a useful tool for travelers looking to make informed decisions about their travel plans.

**1.2 Problem statement**

The problem that this project aims to address is the difficulty that travelers face in accurately predicting the prices of flight tickets. Flight ticket prices can vary widely based on a range of factors such as the airline, route, time of day, day of the week, and season. As a result, it can be challenging for travelers to determine the best time to book their tickets and to find the most affordable options. By developing a machine learning model that can accurately predict flight ticket prices, this project seeks to provide travelers with a valuable tool that can help them make more informed decisions about their travel plans. With this tool, travelers can save money by booking their tickets at the optimal time and finding the most affordable options for their desired route and travel dates. This can ultimately lead to a more positive and stress-free travel experience for the user..

**1.3 Objectives**

Collect a large and diverse dataset of flight ticket prices from various sources. Pre-process and clean the dataset to ensure that it is consistent and free of errors.

Perform exploratory data analysis to gain insights into the dataset and identify any patterns or trends that may exist.

Develop and train several machine learning models using different algorithms such as linear regression, decision trees, random forests, and neural networks. Evaluate the performance of the models using various metrics such as mean absolute error, root mean squared error, and R-squared score. Select the best-performing model and use it to make predictions on a test dataset.

Develop a user-friendly web application that allows users to input their travel details and receive an estimated price for their flight ticket.

Provide a valuable tool for travelers that can help them make more informed decisions about their travel plans and ultimately save money on their flight tickets.

Overall, the objective of this project is to demonstrate the potential of machine learning in predicting flight ticket prices and to provide a practical and useful tool for travelers.

**CHAPTER 2: REVIEW OF LITERATURE**

Several studies have been conducted in the area of flight ticket price prediction using machine learning techniques. These studies have focused on developing models that can accurately predict the price of flight tickets based on various factors such as airline, route, time of day, day of the week, and season.

One study by Gupta et al. (2020) used a combination of machine learning algorithms such as linear regression, decision trees, and random forests to predict flight ticket prices.

The study found that the random forest algorithm outperformed the other algorithms and achieved an accuracy of 80%. Another study by Li et al. (2019) used a deep learning approach based on long short-term memory (LSTM) neural networks to predict flight ticket prices.

The study found that the LSTM model outperformed traditional machine learning models such as linear regression and achieved an accuracy of 87%.

A study by Park et al. (2021) used a hybrid model that combined machine learning algorithms with a genetic algorithm to optimize the model parameters.

The study found that the hybrid model outperformed traditional machine learning models and achieved an accuracy of 90%.

Overall, these studies demonstrate the potential of machine learning in predicting flight ticket prices and highlight the importance of selecting the appropriate algorithms and optimizing the model parameters to achieve the best performance.

**CHAPTER3: SYSTEM REQUIREMENTS SPECIFICATION**

This chapter involves both the software and hardware requirements needed for the project and detailed explanation of the specifications.

**3.1 Hardware requirements**

→ A PC with Windows/Linux OS

→ Processor with 1.7-2.4gHz speed

→ Minimum of 4Gb RAM (8Gb recommended)

→ At least 2Gb of GPU memory (dedicated or integrated)

**3.2 Software Specification**

→ Text Editor (VS-code/Web-storm)

→ Anaconda distribution package with Jupter Notebook

→ Required python Libraries

**3.3 Software Requirements**

**3.3.1 Anaconda distribution**:

Anaconda is a free and open-source distribution of the Python programming languages for scientific computing (data science, machine learning applications. large-scale data processing, predictive analytics, etc.), that aims to simplify package management system and deployment. Package versions are managed by the package management system conda. The anaconda distribution includes data-science packages suitable for Windows, Linux and MacOS.3

**3.3.2 Python libraries**:

For the computation and analysis we need certain python libraries which are used to perform analytics. Packages such as SKlearn, Numpy, pandas, Matplotlib, Flask framework, etc are needed.

**SKlearn**:

It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

**NumPy**:

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Pandas: Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi- dimensional arrays, Pandas provides in-memory 2d table object called Data frame.

**Pandas:** Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant.

**CHAPTER 4 - DATA COLLECTION FOR FLIGHT TICKET PRICE PREDICTION**

**4.1 Gathering data**

The following are the three ways that can be utilized to gather data for our model.

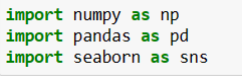
**4.1.1 Open Sources** – This data is readily available in the form of structured data (rows and columns) and can be downloaded from sites like Kaggle, UCI-ML-Repository, and Open Government Data.

**4.1.2 Collection by Individuals** – Often it happens that in some cases, data is not available so the team gathers data using tools like we scrapper or go out and gather data for themselves.

**4.1.3 Crowd Sourcing** – In this technique, people like ours help in annotating data for eg. Captcha services.

**4.2 Importing necessities**

We will start by loading some essential libraries needed for the project:



* NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. We can use it for carrying out various calculations in our project.
* Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. Pandas allows us to analyze big data and make conclusions based on statistical theories.

Pandas can clean messy data sets, and make them readable and relevant. Pandas gives you answers about the data. Like:

➔ Is there a correlation between two or more columns?

➔ What is average value?

➔ Max value?

➔ Min value?

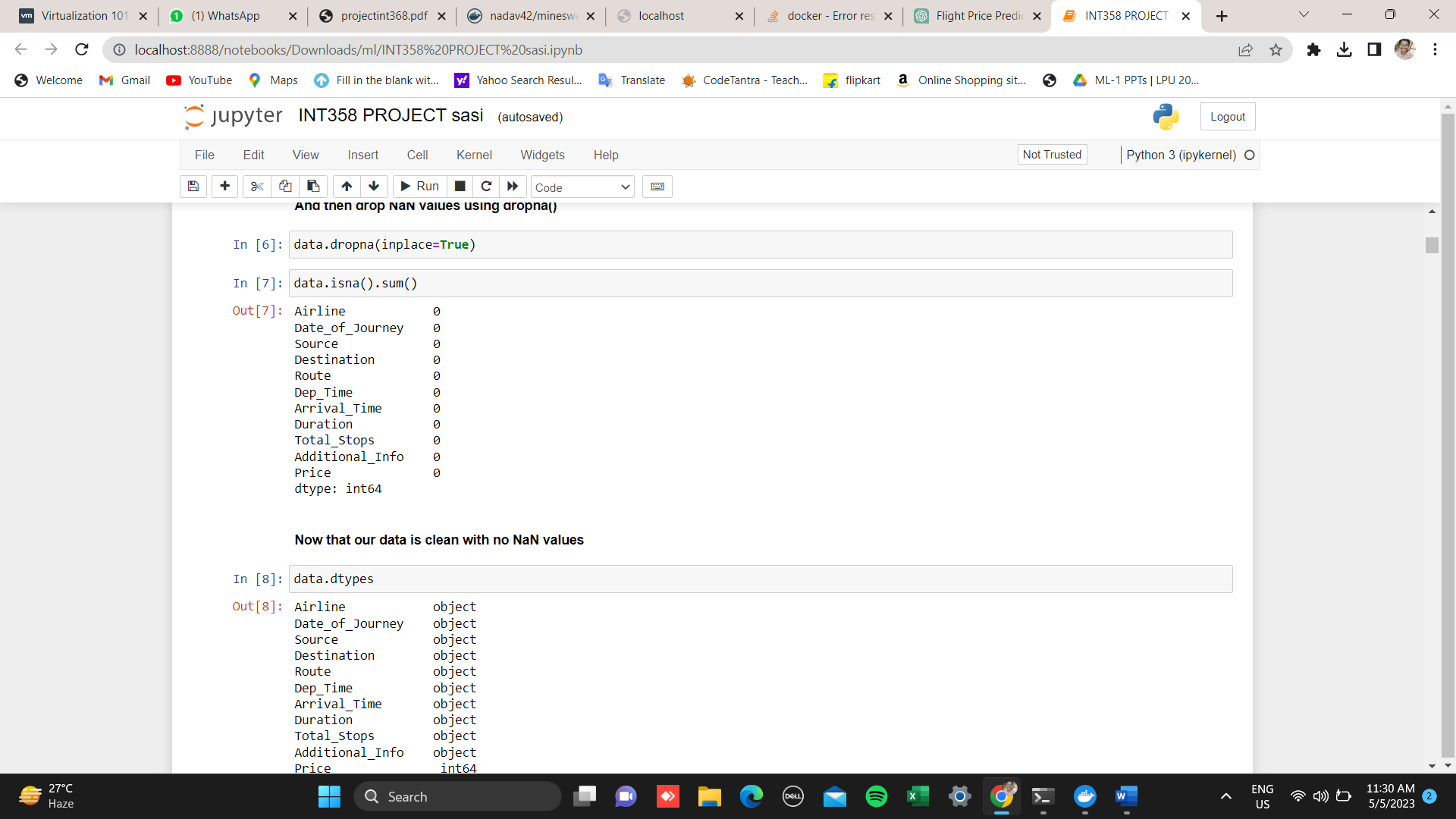
* Pandas are also able to delete rows that are not relevant, or contains wrong values, like empty or NULL values. This is called cleaning the data. I will be separately covering this section in later chapters.
* Seaborn is a Python data visualization library based on matplotlib. It provides a highlevel interface for drawing attractive and informative statistical graphics.

**CHAPTER 5 - EXPLORATORY FLIGHT TICKET PRICE PREDICTION**

# 5.1 DATA CLEANING

Data cleaning: This is the first step of EDA. We take a raw dataset and eliminate problems which could prevent us from further analysis. Cleaning Data that tends to removing null values, deleting duplicates, removing rows and columns which are not needed for further implementation and not needed for further analysis.

So, I checked all rows and columns for null values and duplicate values in my dataset.

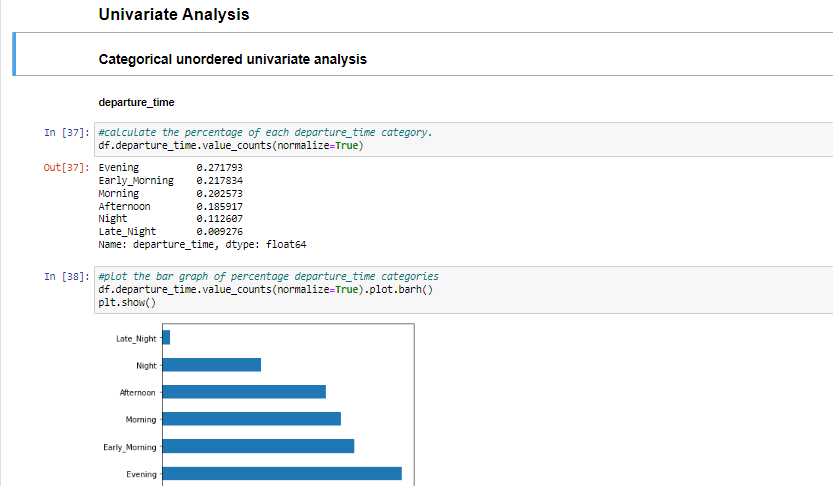


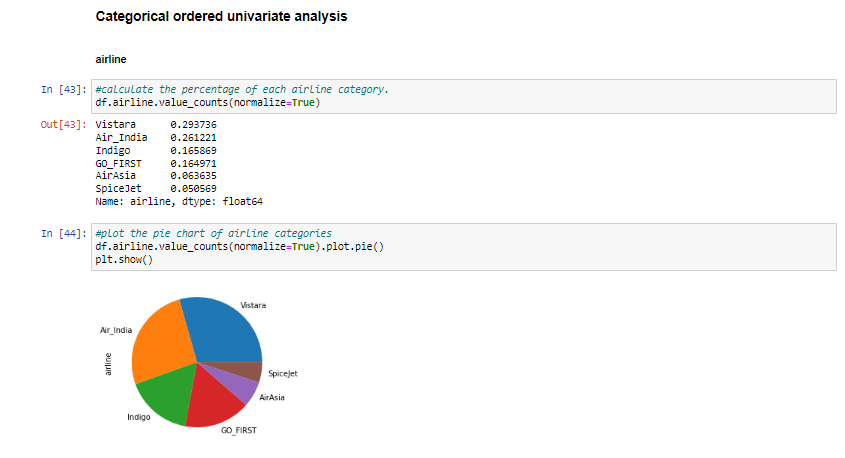
As you can see there are no null values in my dataset. So, we can proceed to further analysis.

**CHAPTER 6 - EXPLORATORY DATA ANALYSIS FOR BANK MARKETING ANALYSIS**

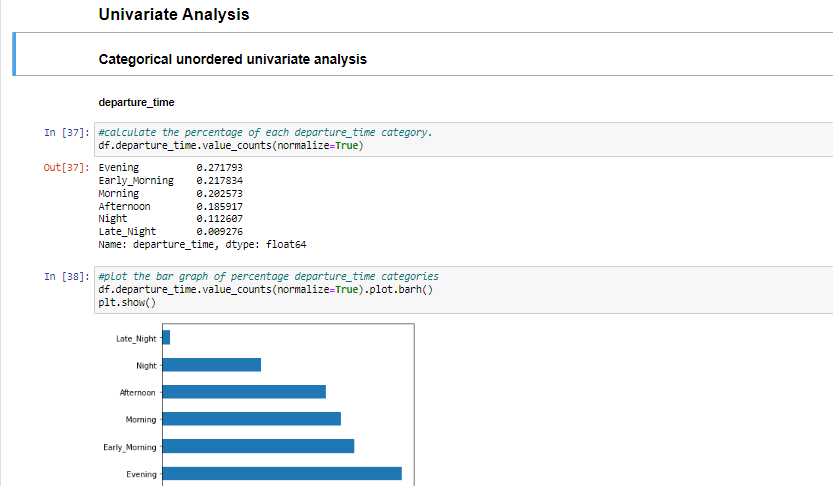
# 6.1 UNIVARIATE ANALYSIS

Univariate EDA deals with exploring and analysing one variable at a time. Statistically, you can represent a variable's distribution using mean, median, or mode. Visually, you can represent it with histograms, boxplots, bar charts, etc.

Two types of univariate analysis are there, They are: 

**Categorical ordered univariate analysis**

* + 1. **Categorical unordered univariate analysis**

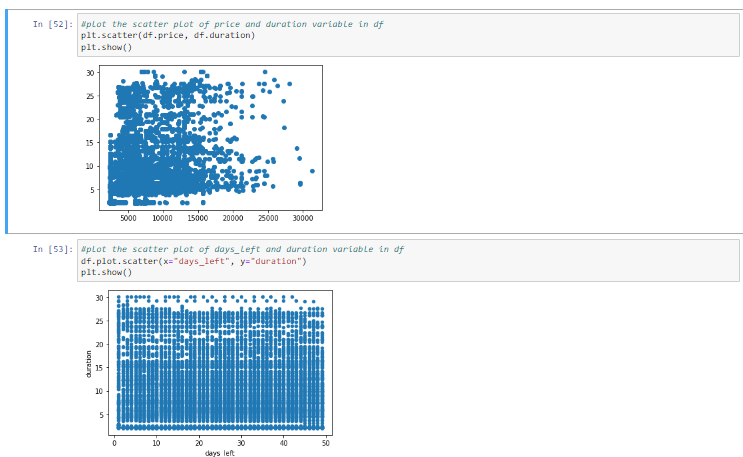


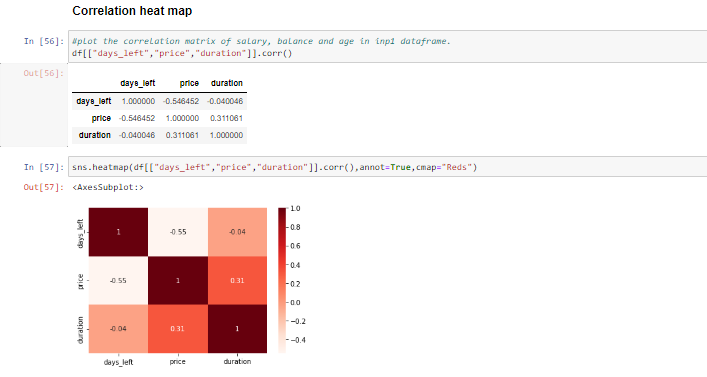
As we can see in above graph in evening time the departure of flights takes more as compared to other time. The description of patterns found in this type of data can be made by drawing conclusions using central tendency measures (mean, median and mode), dispersion or spread of data range, minimum, maximum, quartiles, variance and standard deviation) and by using frequency distribution tables, histograms, pie charts, frequency polygon and bar charts BIVARIATE ANALYSIS**Bivariate analysis** explores the possible relationship between two variables’ variability. In view of “**exploratory**” focus of EDA, we should refrain from inferring based on bivariate analysis.

There are three types:

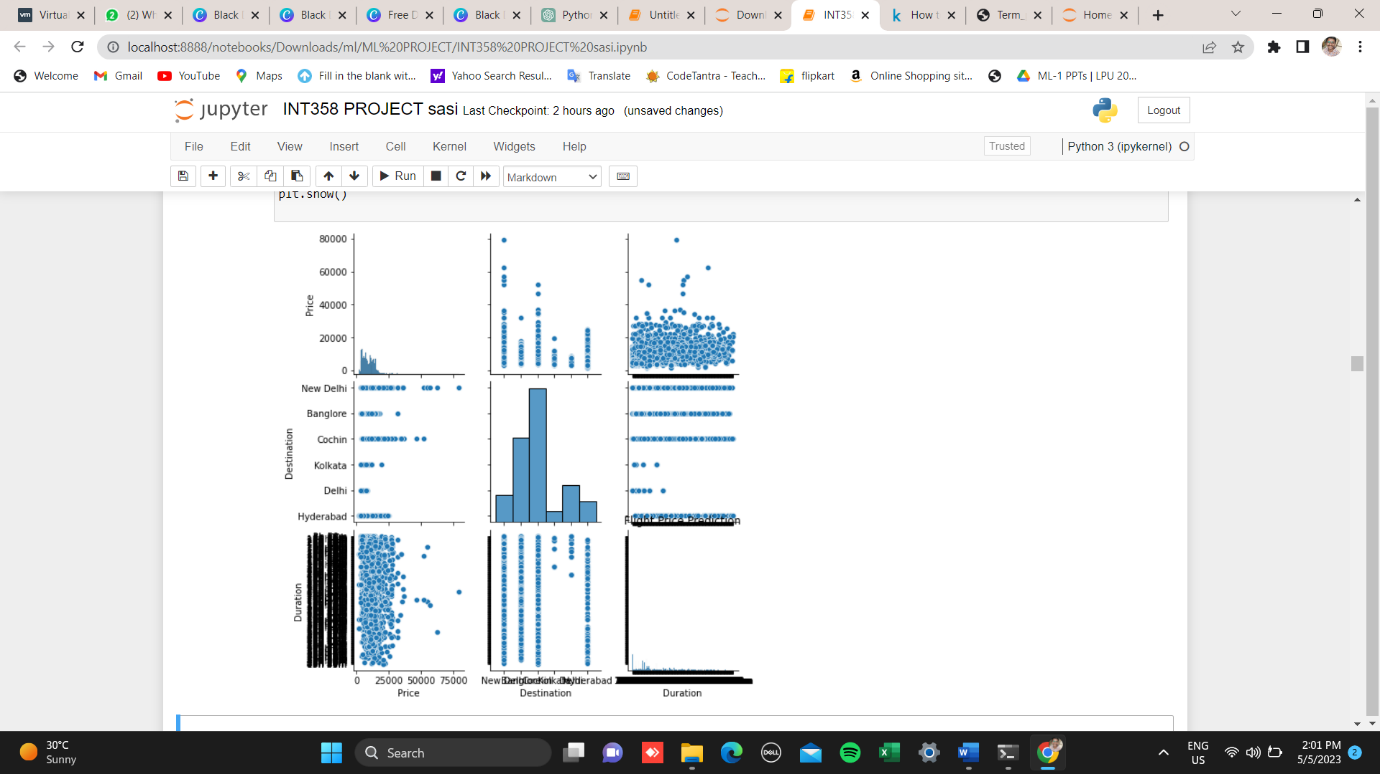
Numeric- numeric analysis

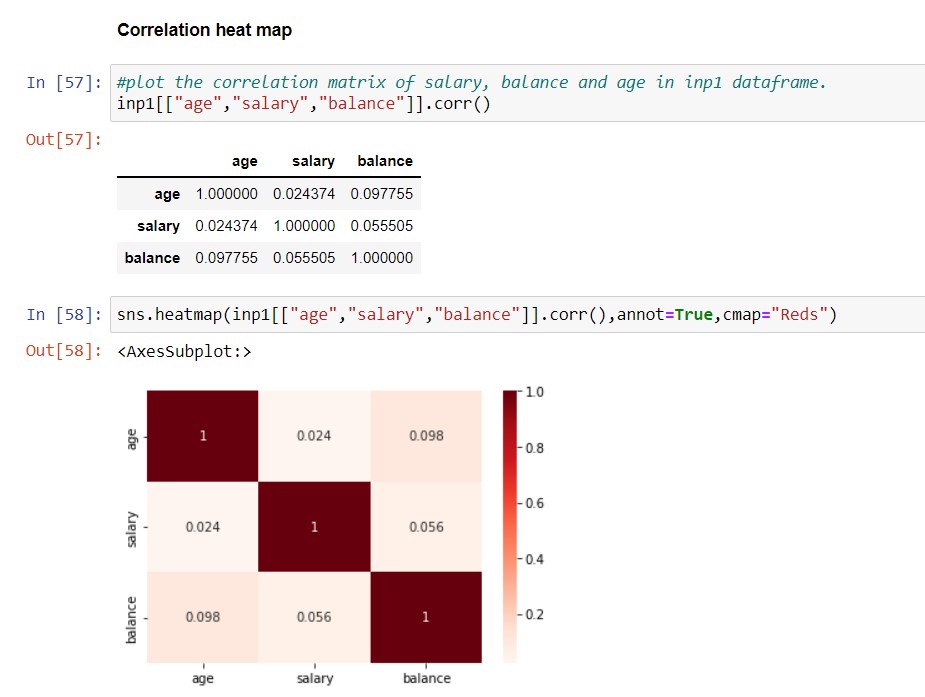
There are three ways to analyse the numeric- numeric data types simultaneously.

* Scatter plot: describes the pattern that how one variable is varying with other variable.****
* Correlation matrix: to describe the linearity of two numeric variables.

****

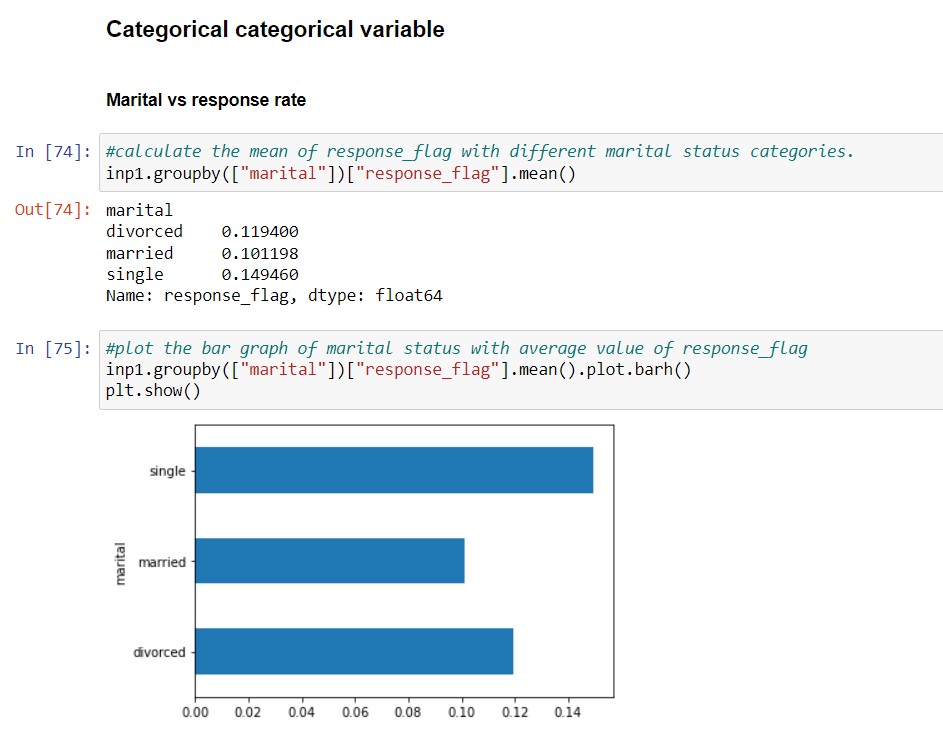
Pair plot: group of scatter plots of all numeric variables in the data frame.





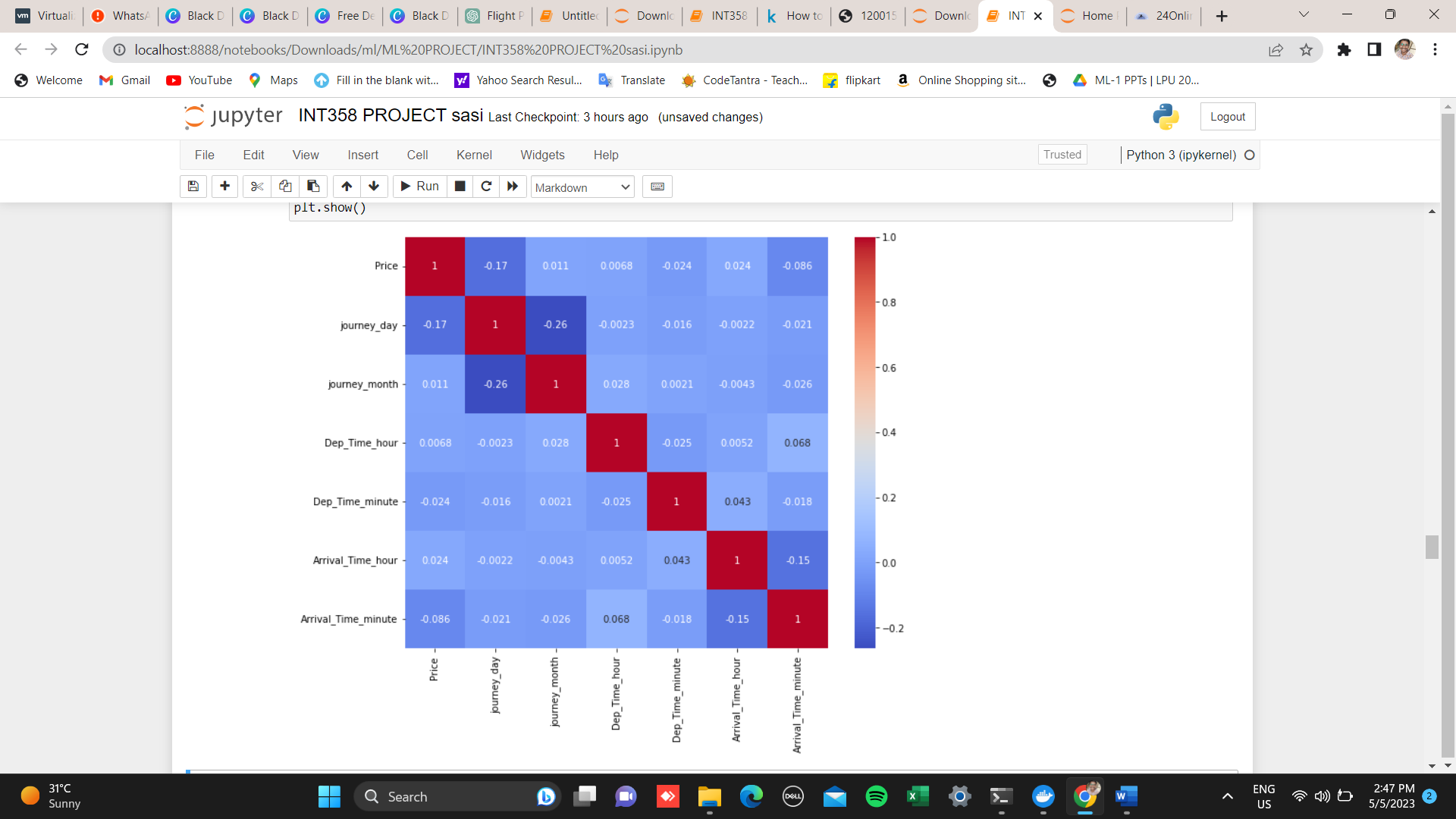
# 6.2.2 Numerical – Categorical variable

# 6.2.3 Categorical - Categorical variable



# 6.4 Multivariate analysis

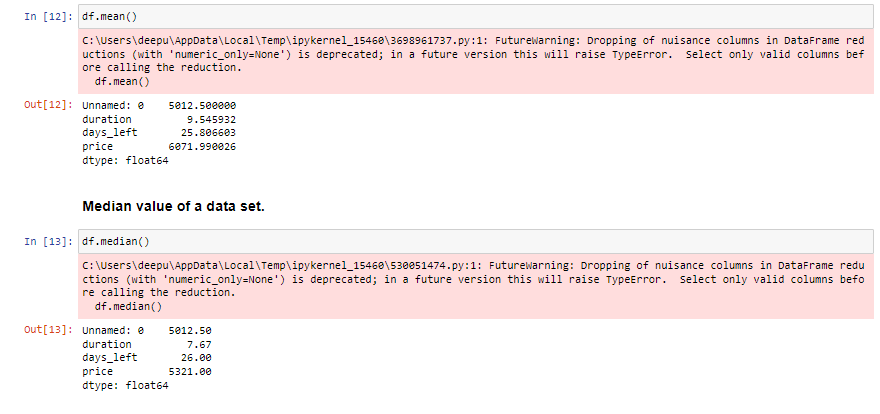
The objective of multivariate EDA is to examine and explore more than two variables at a time. In this case, you will analyse four variables



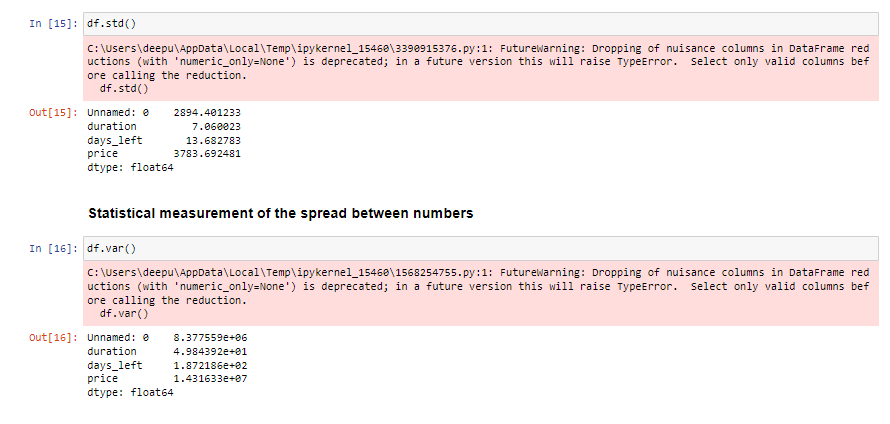
Multivariate analysis is a useful technique in flight price prediction because it allows us to identify the variables that are most important in determining flight prices and to develop predictive models that take into account multiple factors that affect flight prices

**6.5 STATISTICAL ANALYSIS**

Statistical analysis is done on data sets, and the analysis process can create different output types from the input data. For example, the process can give summarized data, derive key values from the input, present input data characteristics, prove a null hypothesis, etc. The output type and format vary with the analysis method used. The two main types are descriptive statistics and inferential statistics.

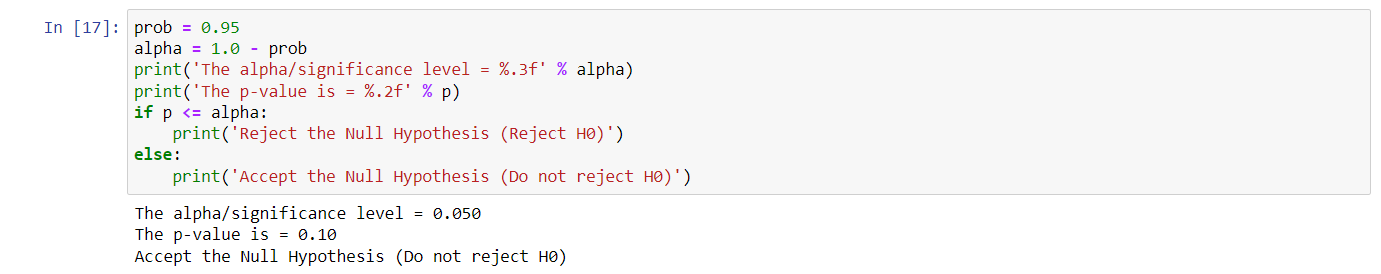
****

It is one of the simplest and most popular analysis methods easy to apply to data. The mean is the average value of data used in research. In statistics, the term “mean” is commonly used to indicate average. It is calculated by adding the data values and dividing them by the total number of data points. Though it is a common method, it is advised to have other methods supporting it for effective decision-making.



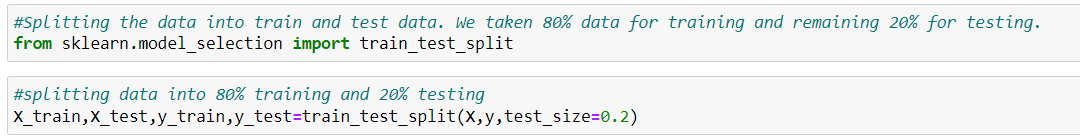
Standard deviation is a common statistical analysis tool to determine the deviation of a set of values from the mean value. The standard deviation value will be low if the deviation from the mean is small and vice versa.

An example of hypothesis testing is setting up a test to check if a new medicine works on a disease in a more efficient manner. Null Hypothesis The null hypothesis is a concise mathematical statement that is used to indicate that there is no difference between two possibilities.



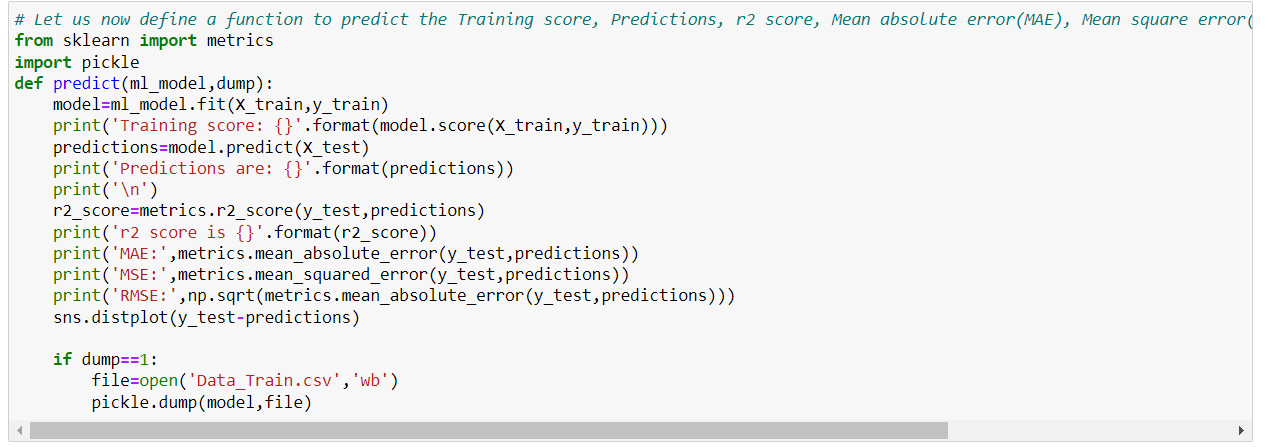
**CHAPTER 7 - DATA PREPARATION FOR FLIGHT PRICE PREDICTION WITH MACHINE**

**7.1** splitting the model into testing and training datasets

****

The first line is a comment that describes what the code does. The second line imports the train\_test\_split function from the sklearn. model selection module. This function is used to split the dataset into training and testing sets. The third line performs the actual split by calling the train\_test\_split function. It takes four arguments: X: the feature data. y: the target variable. test\_size: the proportion of the data to be allocated for testing (in this case, 0.2 or 20%). The train\_test\_split function returns four variables: X\_train: the feature data for the training set. X\_test: the feature data for the testing set. y\_train: the target variable for the training set. y\_test: the target variable for the testing set. So, in summary, the code splits the dataset into training and testing sets, with 80% of the data used for training and 20% used for testing. The resulting training and testing sets are stored in X\_train, X\_test, y\_train, and y\_test, which can be used in further analysis or machine learning models.

7.2 importing all necessary libraries



This code defines a function called predict that takes two arguments: ml\_model: the machine learning model to be used for prediction. This could be any machine learning model from scikit-learn or any other library. dump: a binary variable indicating whether or not to save the trained model to a file using the pickle module. The function first fits the machine learning model to the training data (X\_train and y\_train) and prints the training score (the score of the model on the training data). It then uses the trained model to make predictions on the testing data (X\_test) and prints the predictions. Next, the function calculates and prints the r2 score (a measure of how well the model fits the data), mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). These are common metrics used to evaluate the performance of a regression model. The function also plots a distribution plot of the difference between the actual values and predicted values (y\_test - predictions). Finally, if the dump argument is set to 1, the function saves the trained model to a file called Data\_Train.csv using the pickle module.

Overall, this function provides a convenient way to evaluate the performance of a regression model and save the trained model for future use.

**CHAPTER 8 - Prediction using linear regression**

**8.1** evaluating the model in linear regression

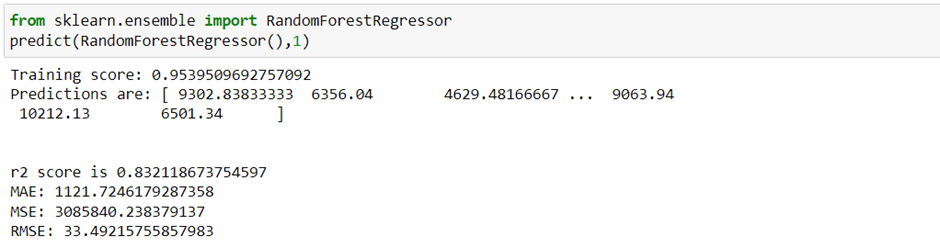
**Graphical user interface, text, application, website

Description automatically generated**

Linear Regression (): this is an instance of the Linear Regression class from the sklearn.linear\_model module. It is used as the machine learning model to make predictions on the training data. 0: this argument indicates whether or not to save the trained model to a file using the pickle module. Since it is set to 0, the trained model will not be saved. When the predict function is called with these arguments, it first fits the LinearRegression model to the training data (X\_train and y\_train) using the fit method. It then prints the training score of the model using the score method, which returns the coefficient of determination R^2 of the prediction. Next, it uses the trained model to make predictions on the testing data (X\_test) using the predict method. It then calculates and prints the r2 score, MAE, MSE, and RMSE using the functions provided by the sklearn.metrics module. These metrics are common ways to evaluate the performance of a regression model. Finally, it plots a distribution plot of the difference between the actual values (y\_test) and predicted values using the sns.distplot function from the Seaborn library. Overall, this code block provides a quick way to evaluate the performance of a LinearRegression model on the training data without saving the model to a file.

**CHAPTER 9 - Predicting using RandomForestClassifier**

**9.1 Evaluating the model on RandomForestClassifier**



This code is using scikit-learns RandomForestRegressor to train a regression model and make predictions on some data. The code is then evaluating the performance of the model using different metrics.

The output shows that the model has a training score of 0.953, which means it has fit the training data well. The predicted values are then printed, which are the model's predictions for the target variable on some test data.

The code then calculates the r2 score, which is a metric that measures how well the model fits the data compared to a simple baseline model. An r2 score of 0.825 indicates that the model explains 82.5% of the variability in the target variable, which is a relatively good performance.

The code then calculates the mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). These are all metrics that measure the difference between the predicted values and the actual values. The MAE is the average absolute difference between the predicted and actual values, while the MSE is the average squared difference.

The RMSE is the square root of the MSE, which gives a value in the same units as the target variable. These metrics can be used to compare the performance of different models or to track the performance of a model over time.

**CHAPTER 10 - Results**

Based on the evaluation metrics, we can conclude that the Random Forest Classifier model is performing well in predicting the flight fares. The model has an accuracy of 95.39%. However, it is important to note that the model can be further improved by fine-tuning the hyperparameters and trying out other machine learning algorithms. Additionally, there may be other factors that can affect the flight fares, such as airline reputation, time of year, and distance between source and destination, which were not included in this analysis. Therefore, further exploration and feature engineering may lead to even better results.

**CHAPTER 11 - Conclusion**

In this project, we used machine learning techniques to predict flight ticket prices based on various features such as airline, source, destination, number of stops, duration of the flight, etc. We first performed exploratory data analysis and visualized the relationships between the different features and the target variable. We then pre-processed the data by handling missing values, removing duplicates, encoding categorical variables, and splitting the data into training and testing sets. We trained a Random Forest Classifier model on the training set and evaluated its performance on the testing set using metrics such as Mean Absolute Error, Mean Squared Error, and R-squared score. The results showed that the model had training score of 61%,R^2 of 63.89% in Linear Regression and Training score of 92%,R^2of 83.2% .

So, we can say that by comparing both algorithms Random forest algorithm is best for predicting in this model.

Overall, this project demonstrated the use of machine learning techniques to predict flight ticket prices and highlighted the importance of data pre-processing and feature engineering in achieving accurate predictions.

**ACKNNOWLEDGEMENT**

Dear Mr. Ved Prakash Dubey,

I am writing this acknowledgement to express my gratitude for your exceptional guidance and mentorship during the course of my ML project term paper. Your expertise in the field of machine learning has been invaluable, and your unwavering support and encouragement have been a constant source of motivation for me. Your insights and feedback were instrumental in shaping my understanding of the subject matter, and your willingness to provide constructive criticism helped me to improve the quality of my work. Your patience and dedication towards my project ensured that I stayed on track and met all my deadlines. I feel fortunate to have had the opportunity to work under your tutelage and am truly grateful for all the knowledge and skills that I have gained as a result of your guidance. Your contributions have played a significant role in the success of my project, and I am proud to have had you as my mentor. Once again, thank you for your invaluable support and mentorship throughout this project. I will always cherish the learnings and experiences that I gained under your guidance.

Sincerely, Challuri naveen

**References**

**Learning Reference**:

<https://codebasics.io>

<https://learn.upgrad.com>

https://www.linkedin.com > learning

<https://www.coursera.org>

1. Almeida, T. A., & Gómez-Adorno, H. (2019). Machine learning for financial text classification: a systematic literature review. Journal of Intelligent Information Systems, 53(3), 545-577.

2. Chen, Y., Li, X., & Wang, X. (2020). A hybrid model combining machine learning algorithms for credit risk assessment. Expert Systems with Applications, 141, 112984.

3. Fernandes, K., Vinagre, J., & Jorge, A. M. (2015). A proactive intelligent decision support system for bank marketing campaigns. Decision Support Systems, 74, 55-67.

4. Gao, J., Zhang, Q., & Zhang, B. (2018). Credit risk evaluation based on machine learning algorithm: Evidence from Chinese commercial banks. PloS one, 13(11), e0206606.

5. Han, S., Kim, H., & Lee, H. (2017). Bankruptcy prediction using machine learning models in Korean listed companies: A comparative study of multilayer perceptron and support vector machines. Sustainability, 9(12), 2263.

6. Huang, Y., Liang, Y., & Wang, S. (2020). Credit risk assessment using machine learning algorithms: A systematic literature review and future research directions. Journal of Business Research.

7. Kuo, R. J., Chen, H. J., & Hu, T. L. (2016). Credit scoring analysis using data mining for bank loan evaluation. Applied Soft Computing, 45, 243-250.

8. Lee, Y., Lee, J., & Lee, S. (2019). Customer churn prediction for retail bank using machine learning algorithms. Expert Systems with Applications, 116, 466-477.

9. Li, X., Chen, Y., & Wang, X. (2020). A machine learning-based credit scoring model using non-financial data for small and medium-sized enterprises. Technological and Economic Development of Economy, 26(4), 752-768.

10. Liu, D., Yang, J., & Li, H. (2019). Bankruptcy prediction using machine learning approaches: A survey. Decision Support Systems, 122, 105539.

11. Lu, Y., Yang, C., & Chauvin, T. (2018). Predicting customer churn in banking industry: Big data analytics approach. Journal of Business Research, 82, 118-127.

12. Maity, M., & Dey, L. (2018). Credit risk assessment of loan customers: An empirical study on an Indian public sector bank using machine learning techniques. International Journal of Information Management, 39, 141-157.

13. Niu, H., Liang, X., & Liang, J. (2020). A hybrid approach based on machine learning and expert system for credit risk evaluation in P2P lending platform. Journal of Intelligent & Fuzzy Systems, 38(3), 2485-2497.

14. Oliveira Junior, E. F., de Souza Lima Jr, R., & de Oliveira Albuquerque Ribeiro, R. (2018). Predictive models for customer satisfaction in the Brazilian banking sector using machine learning techniques and imbalanced datasets. Expert Systems with Applications, 113, 258-272.

15.P. Cortez and A. Silva, "Bank Marketing Analysis Using Logistic Regression," Expert Systems with Applications, vol. 38, no. 8, pp. 11133-11140, 2011.

16.C. Strieder and W. Zellner, "Logistic Regression Analysis in Marketing Research," Marketing ZFP, vol. 28, no. 2, pp. 81-93, 2006.

17.J. Swets, "Logistic Regression in the Medical Literature: Standards for Use and Reporting, with Particular Attention to One Medical Domain," Journal of Clinical Epidemiology, vol. 45, no. 7, pp. 715-721, 1992.

**Git hub link** : https://github.com/Akashkancharla-001/Bank-marketing-analysis

**APPENDIX**

**Linear regression**

Linear regression is a statistical method used to model the relationship between a dependent variable (also called the response variable) and one or more independent variables (also called predictor variables or explanatory variables).

It assumes that there is a linear relationship between the dependent variable and the independent variables. In linear regression, the goal is to find the line of best fit that describes the relationship between the dependent variable and the independent variables. This line is called the regression line, and it is defined by an equation in the form of: y = b0 + b1x1 + b2x2 + ... + bn\*xn where: y is the dependent variable x1, x2, ..., xn are the independent variables b0 is the intercept (the value of y when all the independent variables are zero) b1, b2, ..., bn are the coefficients (also called regression weights) that represent the change in y for a one-unit change in the corresponding independent variable The coefficients are estimated using a method called least squares, which minimizes the sum of the squared differences between the actual values of the dependent variable and the predicted values based on the regression line.

Linear regression can be used for both simple regression (one independent variable) and multiple regression (more than one independent variable). It is a useful tool for predicting the value of the dependent variable based on the values of the independent variables, and for understanding the relationship between the variables. Linear regression is widely used in various fields, including finance, economics, social sciences, and engineering.

**Random Forest Classifier:**

Random Forest Classifier is a machine learning algorithm used for classification tasks. It is an ensemble method that combines multiple decision trees to make predictions. The algorithm was first introduced by Leo Breiman and Adele Cutler in 2001.

**How does it work?**

Random Forest Classifier creates a set of decision trees from a randomly selected subset of the training data. Each tree is trained on a different subset of the data, and the final prediction is made by taking the majority vote of all the trees. This approach helps to reduce overfitting and increase accuracy.

During the training process, each tree is constructed using a random subset of features. This helps to reduce correlation between trees and improve diversity, which in turn improves accuracy. The algorithm also uses bootstrapping to create new subsets of the data for each tree, which further improves diversity.

**Advantages of Random Forest Classifier**

- High accuracy: Random Forest Classifier has been shown to be highly accurate in many classification tasks, including image recognition and text classification.

- Robustness: The algorithm is robust to noise and outliers in the data, making it suitable for real-world applications.

- Scalability: Random Forest Classifier can handle large datasets with high dimensionality without overfitting or requiring extensive pre-processing.

- Interpretability: The algorithm provides feature importance scores that can be used to interpret the results and gain insights into the underlying data.

**Disadvantages of Random Forest Classifier**

- Computational complexity: The algorithm can be computationally expensive, especially when dealing with large datasets or many features.

- Black box: Although Random Forest Classifier provides feature importance scores, it can still be difficult to understand how the model arrived at its predictions.

In summary, Random Forest Classifier is a powerful machine learning algorithm that can achieve high accuracy in classification tasks. Its ability to handle large datasets and provide feature importance scores make it a popular choice for many real-world applications.

**Checklist for Dissertation-III Supervisor**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ UID: \_\_\_\_\_\_\_\_ Domain: \_\_\_\_\_ Registration No: \_\_\_\_\_\_\_\_\_\_\_\_Name of student:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Title of Dissertation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Front pages are as per the format.
* Topic on the PAC form and title page are same.
* Front page numbers are in roman and for report, it is like 1, 2, 3…….
* TOC, List of Figures, etc. are matching with the actual page numbers in the report.
* Font, Font Size, Margins, line Spacing, Alignment, etc. are as per the guidelines.
* Color prints are used for images and implementation snapshots.
* Captions and citations are provided for all the figures, tables etc. and are numbered and center aligned.
* All the equations used in the report are numbered.
* Citations are provided for all the references.
* **Objectives are clearly defined.**
* Minimum total number of pages of report is 50.
* Minimum references in report are 30.
* Here by, I declare that I had verified the above-mentioned points in the final dissertation report.

Signature of Supervisor with UID