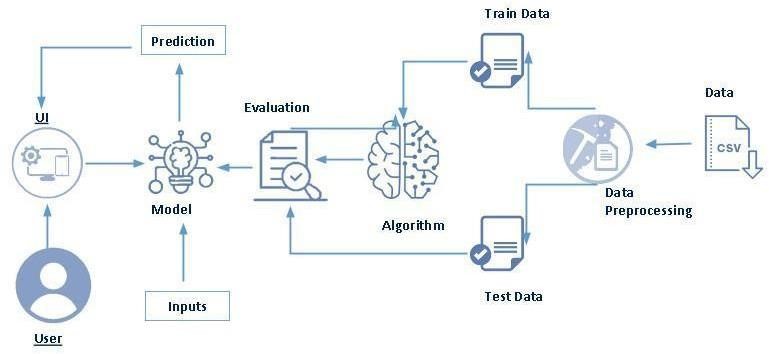
**Food Delivery Estimation using Machine Learning**

In the fast-paced world of modern living, the demand for convenience is at an all-time high. Nowhere is this more evident than in the realm of food delivery. Whether it's a busy professional craving a gourmet meal after a long day at the office or a family looking to enjoy a cozy dinner at home, food delivery services have become an integral part of our daily lives. However, one of the key challenges in ensuring a satisfying food delivery experience lies in accurately estimating the delivery time.

Food delivery estimation using machine learning is a cutting-edge approach to tackle this challenge head-on. By leveraging vast amounts of data, advanced algorithms, and predictive models, we can provide customers with accurate and reliable delivery time estimates. This not only enhances customer satisfaction but also streamlines operations for delivery platforms and restaurants. From data collection and preprocessing to the intricacies of predictive modeling and real-time updates, we will uncover how machine learning is revolutionizing the way we perceive, interact with, and experience food delivery services.

**Technical Architecture:**



**Project Flow:**

* User interacts with the UI to enter the input.
* Entered input is analysed by the model which is integrated.
* Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

* Define Problem / Problem Understanding
  1. Specify the business problem

○ Business requirements

○ Literature Survey

○ Social or Business Impact.

* Data Collection & Preparation
  1. Collect the dataset

○ Data Preparation

* Exploratory Data Analysis
  1. Descriptive statistical

○ Visual Analysis

* Model Building
  1. Training the model in multiple algorithms

○ Testing the model

* Performance Testing & Hyperparameter Tuning
  1. Testing model with multiple evaluation metrics

○ Comparing model accuracy before & after applying hyperparameter tuning

* Model Deployment
  1. Save the best model

○ Integrate with Web Framework

* Project Demonstration & Documentation
  1. Record explanation Video for project end to end solution

○ Project Documentation-Step by step project development procedure

**Prior Knowledge:**

You must have prior knowledge of following topics to complete this project.

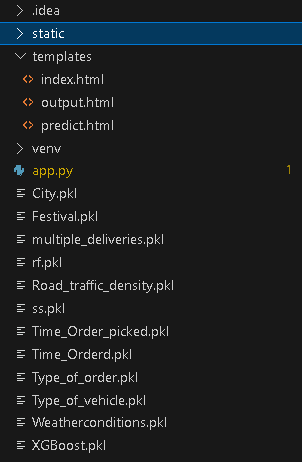
* ML Concepts

o Supervised learning: <https://www.javatpoint.com/supervised-machine-learning>o Unsupervised learning: <https://www.javatpoint.com/unsupervised-machine-learning>

* Decision tree: [https://www.javatpoint.com/machine-learning-decision-tree-classificationalgorithm](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm)
* Random forest: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>
* KNN: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
* Xgboost: [https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-tounderstand-the-math-behind-xgboost/](https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/)
* Evaluation metrics: [https://www.analyticsvidhya.com/blog/2019/08/11-important-modelevaluation-error-metrics/](https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/)
* Flask Basics : <https://www.youtube.com/watch?v=lj4I_CvBnt0>

**Project Structure:**

Create the Project folder which contains files as shown below.



* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
* rf.pkl is our saved model. Further we will use this model for flask integration.
* Train.csv is the Dataset used
* The Notebook file contains procedure for building th model.

### Milestone 1: Define Problem / Problem Understanding

**Activity 1: Specify the business problem**

Refer Project Description

**Activity 2: Business Requirements**

A Food Delivery Estimation project can have a variety of business requirements, depending on the specific goals and objectives of the project. Some potential requirements may include:

**Real-Time Updates:**

Implement mechanisms to continuously update delivery time estimates in real-time, considering factors like traffic congestion, driver speed, and order preparation time.

**Clear Objectives and Goals:**

Define the specific objectives and goals of the food delivery estimation project, such as reducing delivery times, improving accuracy, or enhancing customer satisfaction.

**User Interface:**

Develop user-friendly interfaces for customers and delivery drivers to view estimated delivery times and receive notifications.

**Activity 3: Literature Survey (Student Will Write):**

A literature survey for a drug classification project would involve researching and reviewing existing studies, articles, and other publications on the topic of drug classification. The survey would aim to gather information on current classification systems, their strengths and weaknesses, and any gaps in knowledge that the project could address. The literature survey would also look at the methods and techniques used in previous drug classification projects, and any relevant data or findings that could inform the design and implementation of the current project.

**Activity 4: Social or Business Impact.**

**Social Impact:**

Improved Convenience: Food delivery estimation using machine learning enhances the convenience for customers by providing more accurate delivery time predictions. This reduces the frustration associated with long wait times and ensures that customers can plan their meals more effectively**.**

**Business Impact:**

Customer Retention: Accurate delivery estimations contribute to higher customer satisfaction and can lead to increased customer loyalty. Satisfied customers are more likely to continue using the service.

### Milestone 2: Data Collection & Preparation

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

#### **Activity 1: Collect the dataset**

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

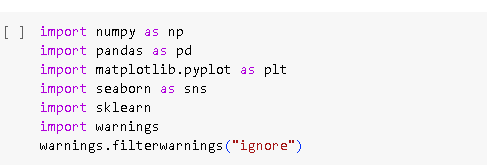
Link: <https://www.kaggle.com/datasets/gauravmalik26/food-delivery-dataset?select=train.csv>[-insurance-claims-data](https://www.kaggle.com/datasets/buntyshah/auto-insurance-claims-data)

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analyzing techniques.

**Note:** There are several techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

#### **Activity 1.1: Importing the libraries**

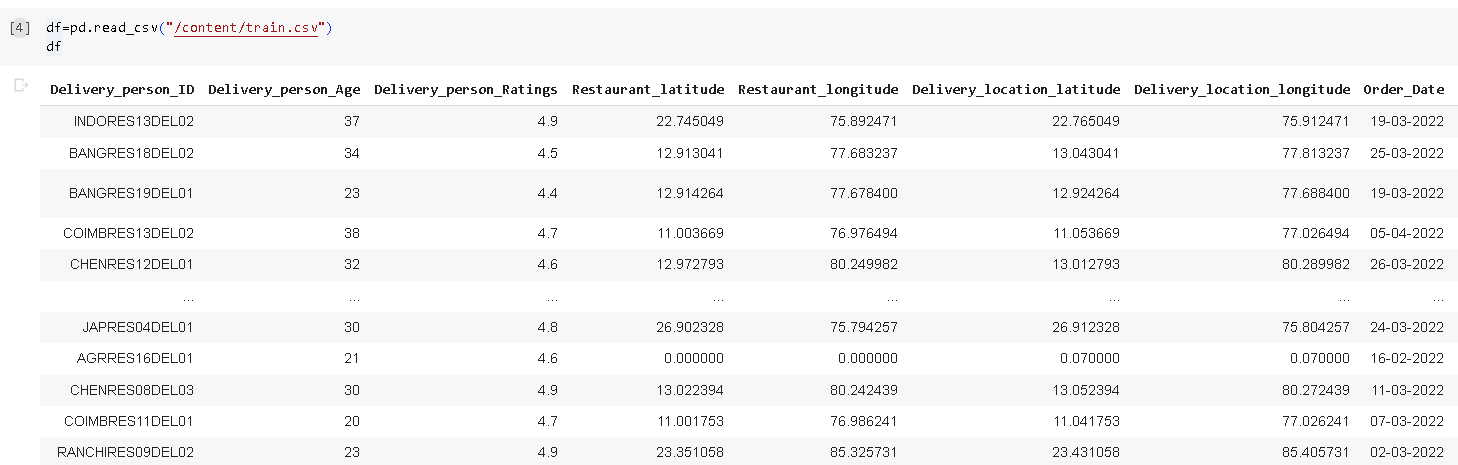
Import the necessary libraries as shown in the image. (optional) Here we have used visualization style as FiveThirtyEight.



#### **Activity 1.2: Read the Dataset**

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of the csv file.



#### **Activity 2: Data Preparation**

As we have understood how the data is, let's pre-process the collected data.

The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

* Handling missing values
* Handling Outliers

Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

#### **Activity 2.1: Handling missing values**

● For checking the null values, df.isna().any( ) function is used. To sum those null values we use .sum() function. From the below image we found that there are no null values present in our dataset. So we can skip handling the missing values step.



#### **Activity 2.2: Handling Outliers**

With the help of boxplot, outliers are visualized. And here we are going to find upper bound and lower bound of Delivery Person Age feature with some mathematical formula.

* From the below diagram, we could visualize that Delivery Person Age feature has outliers. Boxplot from seaborn library is used here.

A screenshot of a computer screen

Description automatically generated

#### 

#### **Activity 2.3: Dropping Unwanted Columns:**

Dropping unwanted columns is a common data preprocessing step when working with datasets, especially in data analysis and machine learning projects. This process involves removing specific columns or features from your dataset that are not relevant or necessary for your analysis or modeling. Below is a description of how to drop unwanted columns in a dataset:

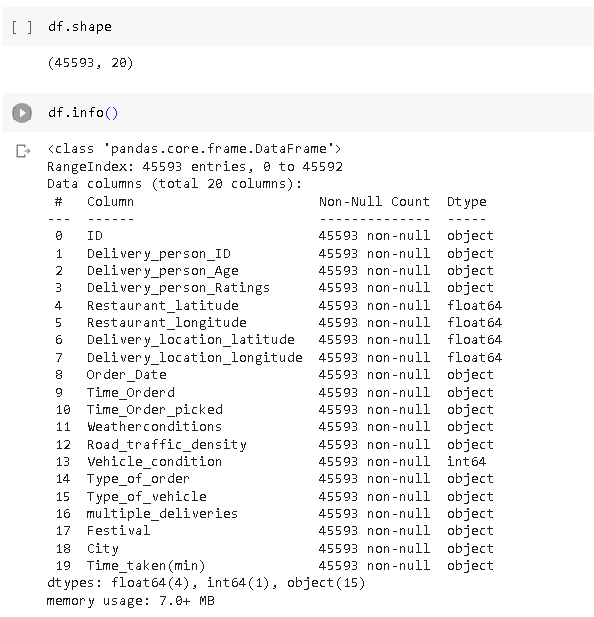
A screenshot of a computer code

Description automatically generated

### Milestone 3: Exploratory Data Analysis

#### **Activity 1: Descriptive statistical**

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.



A screenshot of a computer

Description automatically generated

#### **Activity 2: Visual analysis**

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

#### **Activity 2.1: Univariate analysis**

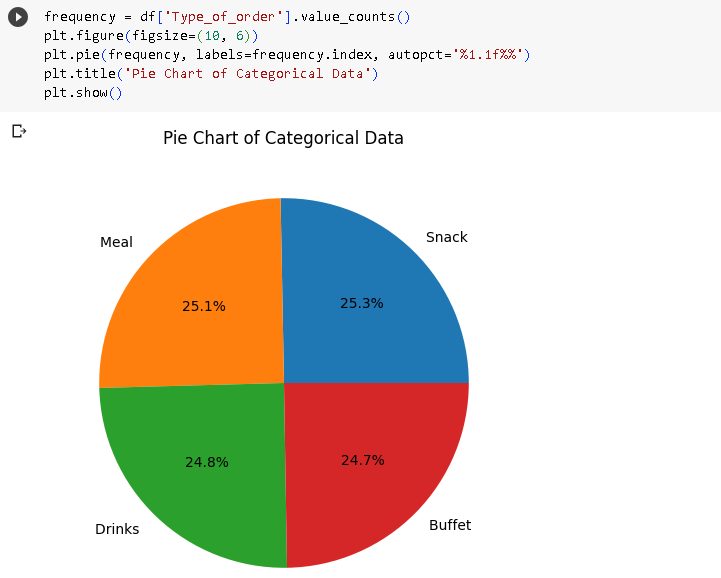
In simple words, univariate analysis is understanding the data with single feature. Here we have displayed two different graphs such as Piechart and countplot.

A "histplot" typically refers to a histogram plot in the context of data visualization. A histogram is a graphical representation of the distribution of a dataset. It shows the frequency or count of data points falling into specific intervals or "bins" along the numerical range of the data. Histograms are particularly useful for understanding the underlying data distribution, identifying patterns.



From The Histogram, Most of the People are in the Age group of 20 to 40.

A pie chart is a circular data visualization that is used to represent data in a way that illustrates the relative proportions or percentages of different categories within a whole. Each category is represented as a "slice" of the pie, with the size of each slice proportional to the value or percentage it represents. Pie charts are particularly useful for showing the composition of a whole and making it easy to compare the parts to the whole.



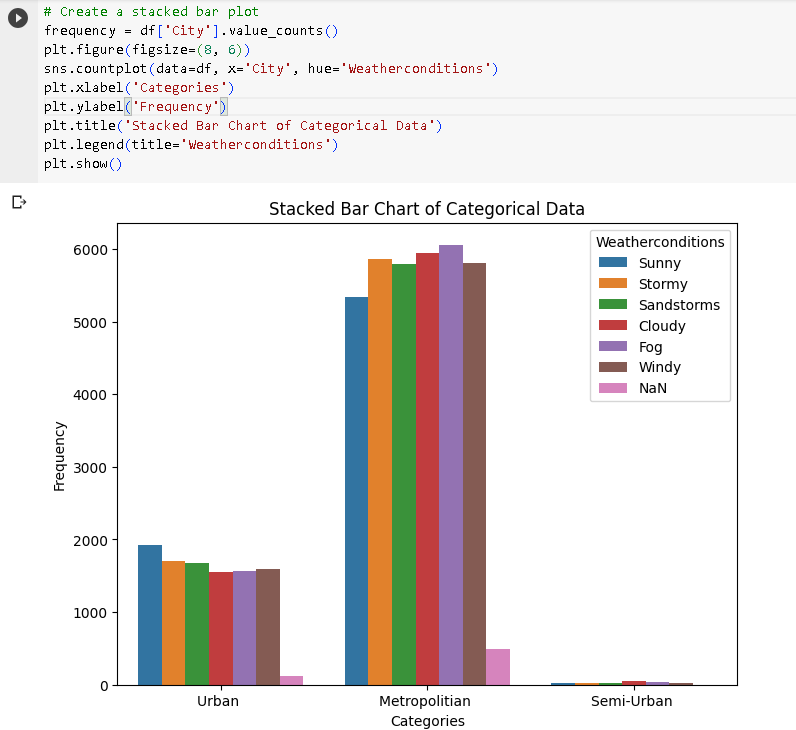
The Meal, Snack, Buffet, Drinks share almost equal quarters.

#### **Activity 2.2: Bivariate analysis**

To find the relation between two features we use bivariate analysis. Here we use Stacked Barchart, Scatterplot.

**A stacked bar chart**, also known as a stacked bar graph, is a data visualization that represents data in the form of rectangular bars.

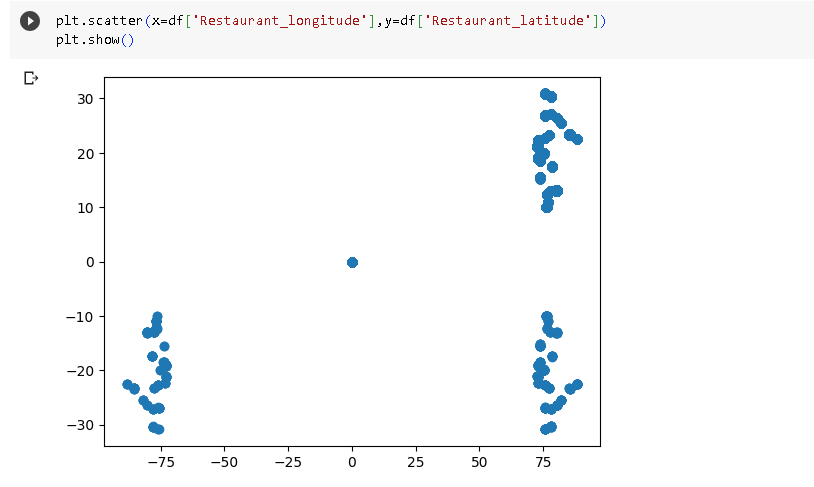
Unlike a regular bar chart where the bars are side by side, in a stacked bar chart, the bars are stacked on top of each other, and each bar is divided into segments or subcategories.



Stacked Barchart Indicates the categories of Weather conditions in Different Cities.

**Scatter Plot:**

A scatterplot is a type of data visualization used to display the relationship between two continuous variables or to visualize individual data points within a dataset. It is particularly useful for identifying patterns, trends, clusters, and outliers in data. Scatterplots consist of points, each representing a single data observation, plotted on a Cartesian plane with one variable on the x-axis and another on the y-axis.



The Scatterplot Depicts the Location of Different Restaurants.

**HeatMap:**

A heatmap is a data visualization technique used to represent the data values in a matrix or a table using colors. It is particularly useful for displaying the magnitude or intensity of a variable in a two-dimensional format.

A screenshot of a computer

Description automatically generated

**MILESTONE 3: Feature Engineering:**

Feature engineering is a crucial process in data science and machine learning that involves creating new, relevant, and informative features (variables) from the existing data to improve the performance of a machine learning model. Effective feature engineering can have a significant impact on model accuracy and predictive power.

**Activity 1: Column Transformation**

In the Data set we have Restaurant Latitude, Restaurant Longitude, Delivery Location Latitude, Delivery Location Longitude. So We can calculate the distance between The Restaurant and Delivery Location given their latitudes and longitudes, using the Haversine formula. The Haversine formula calculates the great-circle distance between two points on the surface of a sphere (like the Earth).

A white background with black and red text

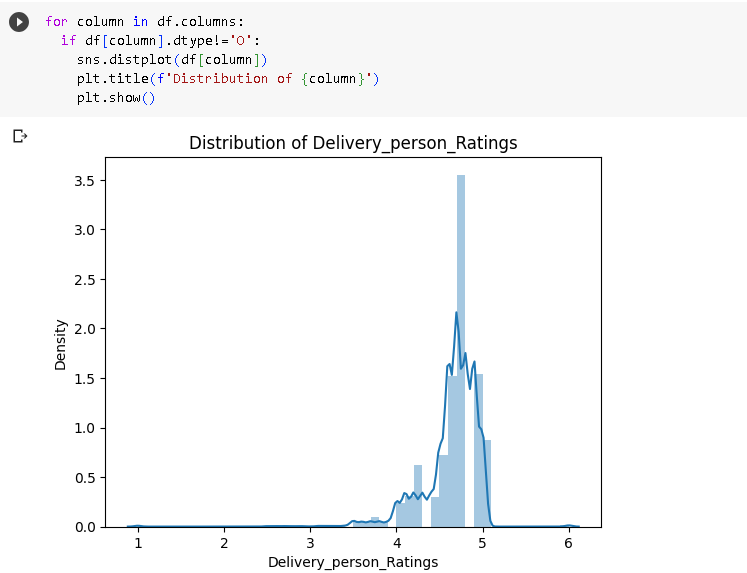
Description automatically generated

Here I have Created a New Column Named Distance\_Km which represents the distance between the delivery Location and Restaurant and dropped the Restaurant Latitude, Restaurant Longitude, Delivery Location Latitude, Delivery Location Longitude

**Activity2: Data Transformation:**

Data transformation is a fundamental step in data preprocessing and preparation for various data analysis, machine learning, and modeling tasks. It involves modifying the raw data to make it suitable for analysis, model training, or visualization. Data transformation encompasses several techniques and processes, depending on the characteristics of the data and the specific goals of the analysis.

**Distplot:** A distplot, short for distribution plot, is a data visualization tool that displays the distribution of a univariate dataset. It combines a histogram with a kernel density estimation (KDE) plot to provide insights into the data's underlying probability distribution. Distplots are particularly useful for understanding the data's central tendency, spread, and shape.

****

So Here Apply the Square root Transformation on The columns.

A screenshot of a computer code

Description automatically generated

**Activity 3: Encoding**

* The categorical Features are can’t be passed directly to the Machine Learning Model. So we convert them into Numerical data based on their order. This Technique is called Encoding.
* Here we are importing Label Encoder from the Sklearn Library.
* Here we are applying fit\_transform to transform the categorical features to numerical features.

**A screenshot of a computer program

Description automatically generated**

##### **Activity 4: Splitting data into train and test**

Now let’s split the Dataset into train and test sets. First split the dataset into x and y and then split the data set

Here x and y variables are created. On x variable, df is passed with dropping the target variable. And on y target variable is passed.

A screenshot of a computer

Description automatically generated

###### **Activity 5: Scaling**

* Scaling is a technique used to transform the values of a dataset to a similar scale to improve the performance of machine learning algorithms. Scaling is important because many machine learning algorithms are sensitive to the scale of the input features.
* Here we are using Standard Scaler.
* This scales the data to have a mean of 0 and a standard deviation of 1. The formula is given by:

X\_scaled = (X - X\_mean) / X\_std

A screenshot of a computer program

Description automatically generated

**Activity 6: Performing Train Test Split:**

train\_test\_split is a commonly used function in machine learning for splitting a dataset into two subsets: a training set and a testing (or validation) set. This split allows you to train a machine learning model on one portion of the data and evaluate its performance on another, which helps assess the model's ability to generalize to new, unseen data.

A black text on a white background

Description automatically generated

### Milestone 4: Model Building

#### **Activity 1: Training the model in multiple algorithms:**

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying three classification algorithms. The best model is saved based on its performance.

Metrics for regression are used to evaluate the performance of machine learning models that are designed for predicting continuous numerical values. These metrics help assess how well a regression model's predictions align with the actual target values. Here are some commonly used regression metrics:

**R-squared (R2) Score:**

R-squared measures the proportion of the variance in the target variable that is explained by the model. It ranges from 0 to 1, with higher values indicating a better fit.

Formula: R2 = 1 - (Σ(y\_i - ŷ\_i)^2) / (Σ(y\_i - ȳ)^2)

**Mean Absolute Error (MAE):**

MAE measures the average absolute difference between the predicted values and the actual target values. It gives equal weight to all errors, making it less sensitive to outliers.

Formula: MAE = (1/n) \* Σ|y\_i - ŷ\_i|

**Mean Squared Error (MSE):**

MSE measures the average squared difference between the predicted values and the actual target values. Squaring the errors gives more weight to larger errors, making it sensitive to outliers.

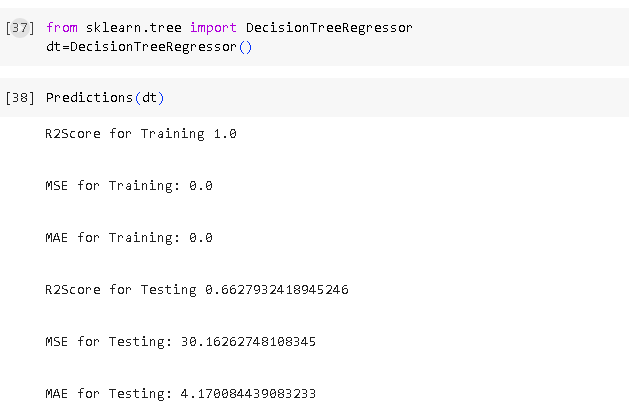
Formula: MSE = (1/n) \* Σ(y\_i - ŷ\_i)^2

A screenshot of a computer program

Description automatically generated

#### **Activity 1.1: Decision tree model**

First Decision Tree is imported from sklearn Library then DecisionTreeClassifier algorithm is initialised and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. We can find the Train and Test accuracy by X\_train and X\_test.



#### **Activity 1.2: Random forest model:**

First Random Forest Model is imported from sklearn Library then RandomForestClassifier algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in a new variable. We can find the Train and Test accuracy by X\_train and X\_test.

A screenshot of a computer

Description automatically generated

#### **Activity 1.3: KNN model**

KNN Model is imported from sklearn Library then KNeighborsClassifier algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.

A screenshot of a computer

Description automatically generated

**Activity 1.4: XG Boost Model**

XGBoost (Extreme Gradient Boosting) is a popular and powerful machine learning algorithm that belongs to the ensemble learning family. It is widely used for both classification and regression tasks and has gained popularity in data science competitions and real-world applications due to its high predictive accuracy and efficiency.

A screenshot of a computer

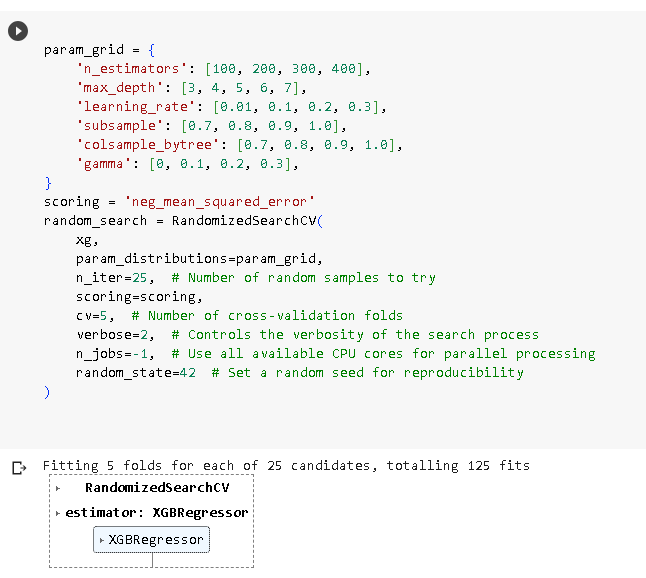
Description automatically generated

**Activity 2: Hyper Parameter Tuning:**

Hyperparameter tuning is a crucial step in optimizing the performance of machine learning models. Hyperparameters are settings or configurations that are not learned from the data but are set prior to training a model. These parameters can significantly impact a model's performance, and finding the best combination of hyperparameters is often done through a process called hyperparameter tuning.

One popular method for hyperparameter tuning is Randomized Search Cross-Validation (RandomizedSearchCV). RandomizedSearchCV is a more efficient alternative to Grid Search, which exhaustively explores all possible combinations of hyperparameters. Instead, Randomized Search samples a specified number of hyperparameter combinations randomly. This approach can save a lot of time and computational resources while still finding good hyperparameters.

**Activity 2.1 Implementing RandomSearchCV for XGBoost:**



A screenshot of a computer

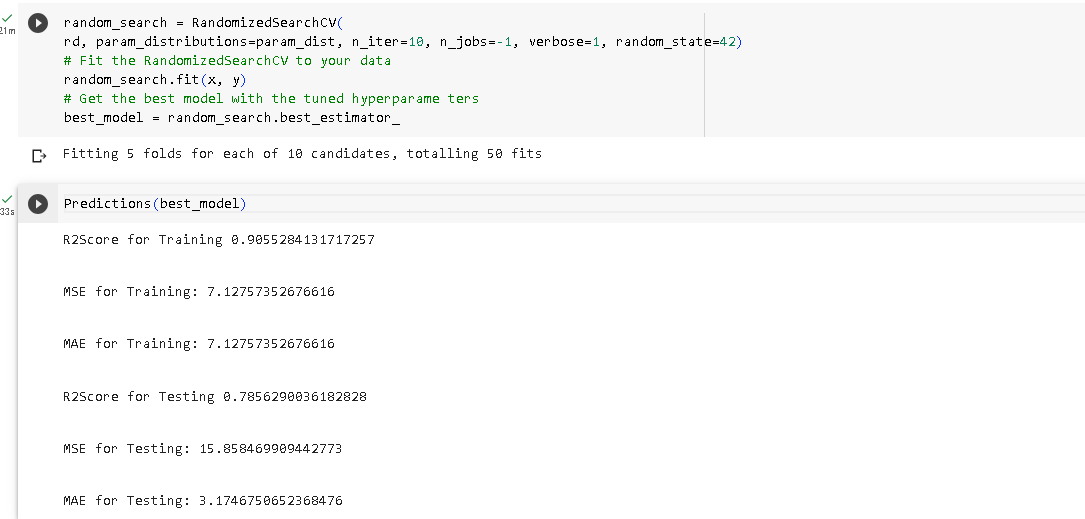
Description automatically generated

**Activity 2.2: RandomSearchCv for Decision Tree:**

A screenshot of a computer program

Description automatically generated

**Activity 2.3: RandomSearchCv for RandomForest:**



### Milestone 6: Model Deployment

#### **Activity 1: Save the best model**

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance.This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.

A computer screen shot of a computer code

Description automatically generated

#### **Activity 2: Integrate with Web Framework**

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server-side script
* Run the web application

**Activity 2.1: Building Html Page:**

For this project create HTML file namely

index.html

Predict.html

Output.html

and save them in the templates folder. Refer this [link](https://drive.google.com/file/d/1kgOB42PCwDzr4KYB7euFZWluyarjq2CH/view?usp=share_link) for templates.

**Activity 2.2: Build Python code:**

Import the libraries

A screenshot of a computer program

Description automatically generated

Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (name ) as argument.

A screen shot of a computer program

Description automatically generated

Render HTML page:

A screenshot of a computer program

Description automatically generated

Here we will be using a declared constructor to route to the HTML page which we have created earlier.

In the above example, ‘/’ URL is bound with the index.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered.

Whenever you enter the values from the html page the values can be retrieved using POST Method.

Retrieves the value from UI:

A computer screen shot of text

Description automatically generated

Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the submit.html page earlier.

Main Function:

A computer code with text

Description automatically generated

#### **Activity 2.3: Run the web application**

* Open anaconda prompt from the start menu
* Navigate to the folder where your python script is.
* Now type “python app.py” command
* Navigate to the localhost where you can view your web page.
* Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

A screen shot of a computer

Description automatically generated

Now,Go the web browser and write the localhost url (http://127.0.0.1:5000) to get the below result

A person riding a red scooter

Description automatically generated

A person holding a bag of food

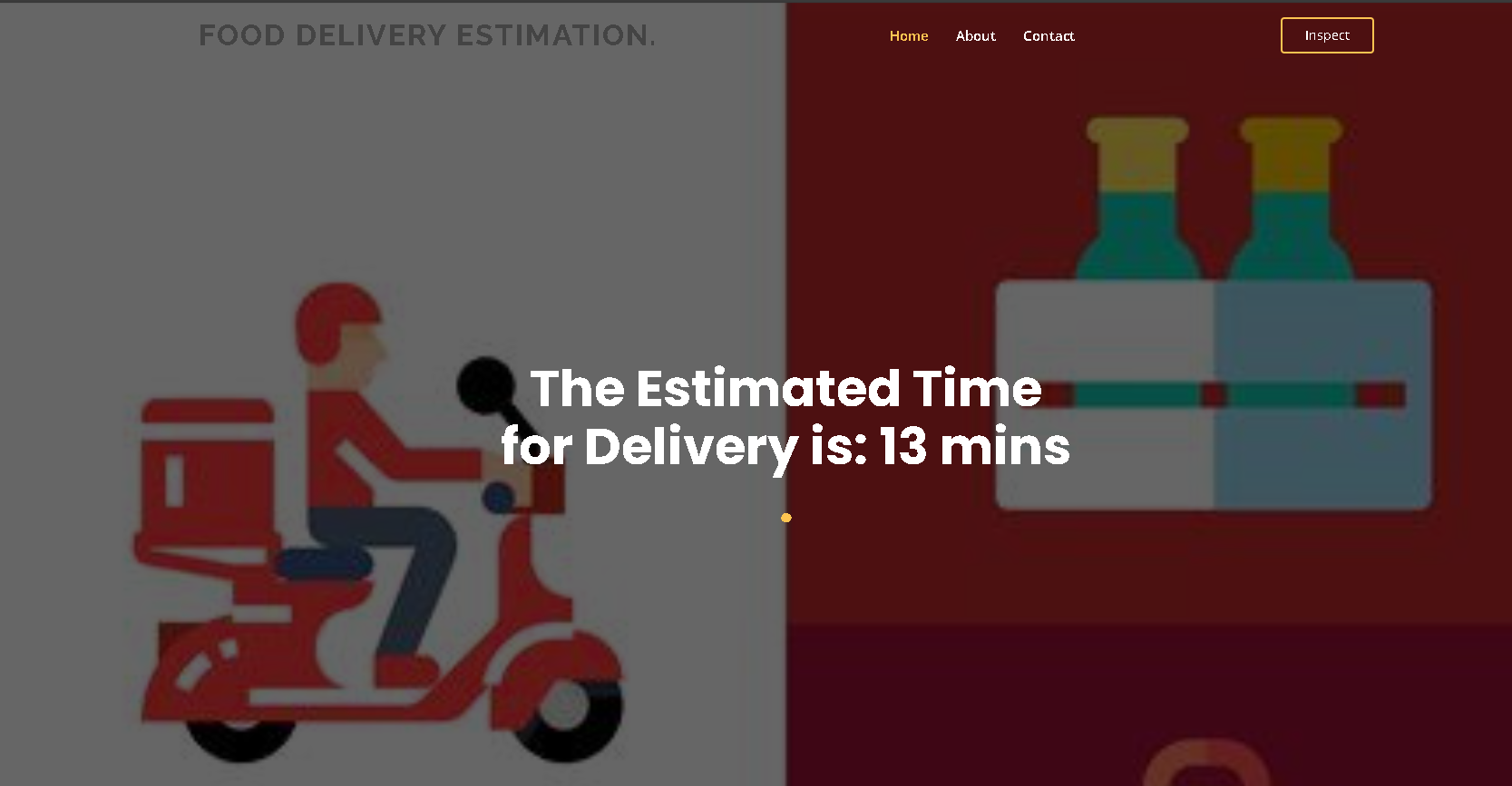
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



##### **Milestone 7: Project Demonstration & Documentation**

Below mentioned deliverables to be submitted along with other deliverables

**Activity 1:- Record explanation Video for project end to end solution**

#### **Activity 2:- Project Documentation-Step by step project development procedure**

Create document as per the template provided