

Text  Description automatically generated with low confidence

**Waste-Whiz: Data Science-enabled handle Market Efficiency Enhancement**

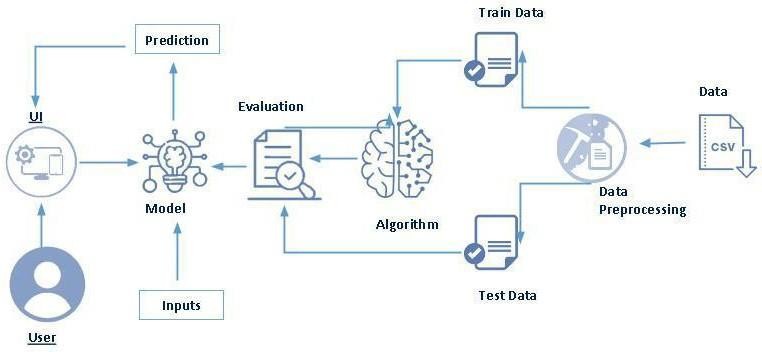
SmartInternz [www.smartinternz.com](http://www.smartinternz.com/)

# WASTE-WHIZE: DATA SCIENCE ENABLE HANDLE MARKET EFFICIENCY ENHANCEMENT

**Project Description:**

In this project, the goal is to leverage machine learning techniques to optimize the yearly marketing spend of an organization that offers a hiring assessment platform. The objective is to build a sophisticated machine learning model that can predict the most effective marketing channels, allocate resources efficiently, and ultimately reduce overall marketing costs while maximizing the acquisition of qualified leads and customers. The dataset used in this project will be collected from various sources such as online surveys, social media platforms, and other publicly available data sources. The data will be pre-processed and cleaned to ensure its quality and eliminate any noise or missing values. once the data is cleaned, it will be split into training and testing sets. Several machine learning models will be built and evaluated on the training data to determine the best-performing model. The models to be explored include linear regression, random forests, and boosting algorithms. After selecting the best-performing model, it will be used to predict the satisfaction level of the passengers in the testing set. The model's performance will be evaluated based on various metrics such as mse, mae, and mape.

# Technical Architecture:



**Project Flow:**

* User interacts with the UI to enter the input.
* Entered input is analyzed 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
  + Specify the business problem
  + Business requirements
  + Literature Survey
  + Social or Business Impact.
* Data Collection & Preparation
  + Collect the dataset
  + Data Preparation
* Exploratory Data Analysis
  + Descriptive statistical
  + Visual Analysis
* Model Building
  + Training the model in multiple algorithms
  + Testing the model
* Performance Testing & Hyperparameter Tuning
  + Testing model with multiple evaluation metrics
  + Comparing model accuracy before & after applying hyperparameter tuning
* Model Deployment
  + Save the best model
  + Integrate with Web Framework
* Project Demonstration & Documentation
  + 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 the following topics to complete this project.

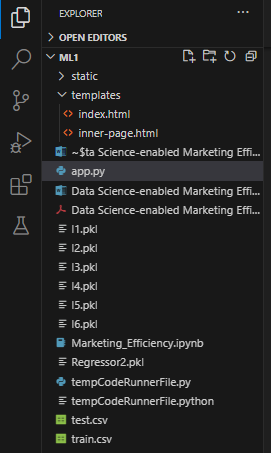
* ML Concepts

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

* + Decision tree Regression: https://medium.com/analytics-vidhya/regression-trees- decision-tree-for-regression-machine-learning
  + Random forest Regression https://towardsdatascience.com/random-forest-regression
  + XGBoost Regression https://towardsdatascience.com/xgboost-regression-explain-it-to- me-like-im-10-2cf324b0bbdb
  + Evaluation metrics: https://towardsdatascience.com/3-evaluation-metrics-for-regression- 80cb34cee0e8
* 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 that needs HTML pages stored in the templates folder and a Python script app.py for scripting.
* model.pkl will be our saved model. Further, we will use this model for flask integration.
* The training folder contains a model training file.

# Milestone 1: Define Problem / Problem Understanding

**Activity 1: Specify the business problem**

Refer Project Description

# Activity 2: Business requirements

Address the challenge of high marketing expenditures in SaaS organizations.

Propose a solution to target a qualified customer set for improved revenue, deal closure rates, and profit margins. Realize cost savings by strategically allocating marketing resources to leads with a higher predicted probability of conversion, resulting in enhanced revenue generation, increased deal closure rates, and improved profit margins.

# Activity 3: Literature Survey (Student Will Write)

The challenge of optimizing marketing expenditures is a common concern for Software as a Service (SaaS) organizations, as highlighted in the problem statement. Existing literature underscores the significance of targeted customer acquisition in maximizing revenue and profit margins. Machine Learning (ML) techniques have been increasingly employed to address such challenges. Regression, a fundamental ML concept, is frequently utilized in predicting customer behavior, and its application in marketing lead conversion aligns with industry practices. This literature survey emphasizes the relevance of ML models in optimizing marketing spending and underscores the importance of regression techniques for predicting marketing lead conversion probabilities.

# Activity 4: Social or Business Impact.

Social Impact: Enhancing marketing budget efficiency through machine learning promotes sustainable business practices, minimizing unnecessary expenditures and contributing to a more responsible allocation of resources.

Business Impact: t Implementing the sophisticated ML model for predicting marketing lead conversion probabilities results in increased revenue, higher deal closure rates, and improved profit margins, thereby positively impacting the organization's financial performance and competitiveness.

# 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.

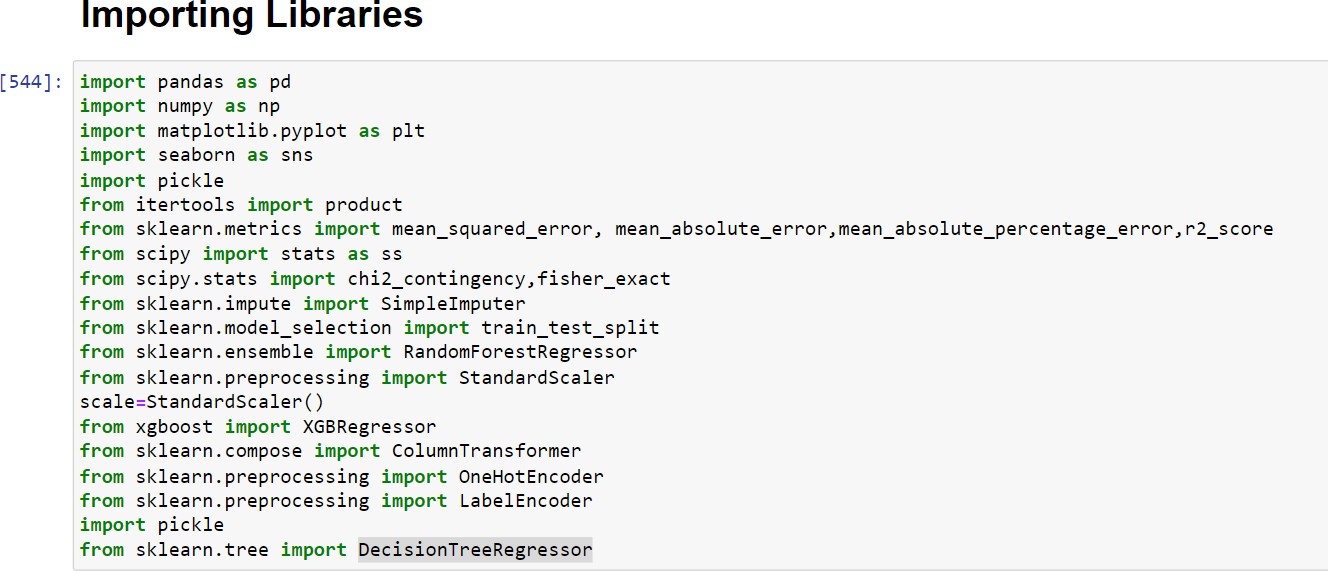
Link: [https://www.kaggle.com/datasets/bhavikjain/reduce-marketing-waste-hackerearth-ml-](https://www.kaggle.com/datasets/bhavikjain/reduce-marketing-waste-hackerearth-ml-challenge?select=train.csv) [challenge?select=train.csv](https://www.kaggle.com/datasets/bhavikjain/reduce-marketing-waste-hackerearth-ml-challenge?select=train.csv)

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 a number of 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

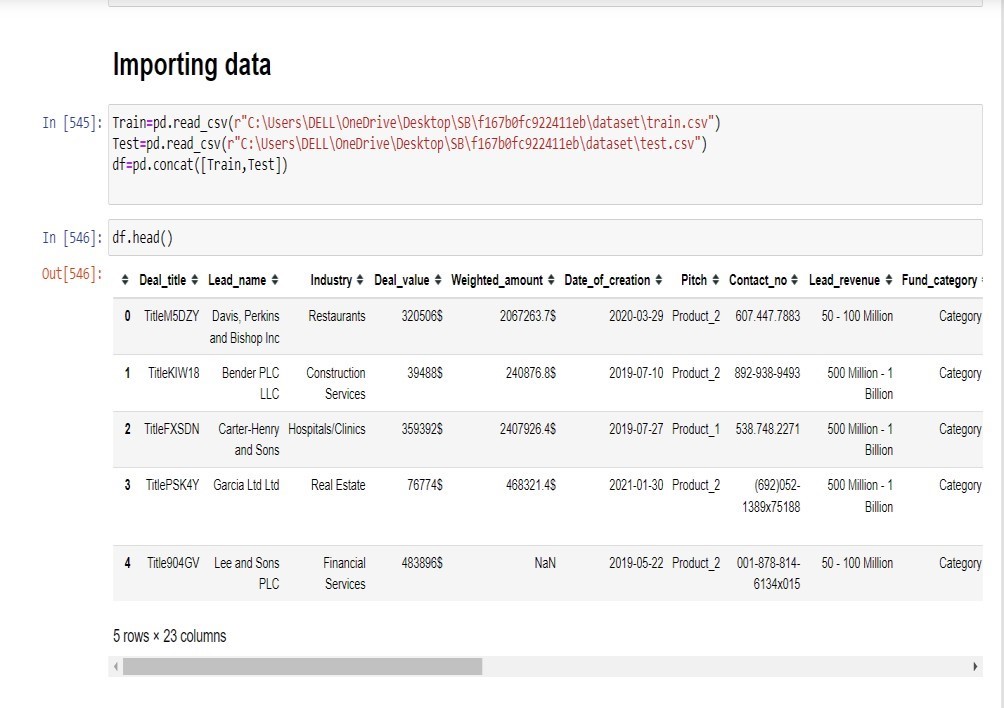
Import the necessary libraries as shown in the image.



# Activity 1.2: Read the Dataset

Our dataset format might be in .csv, excel files 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.



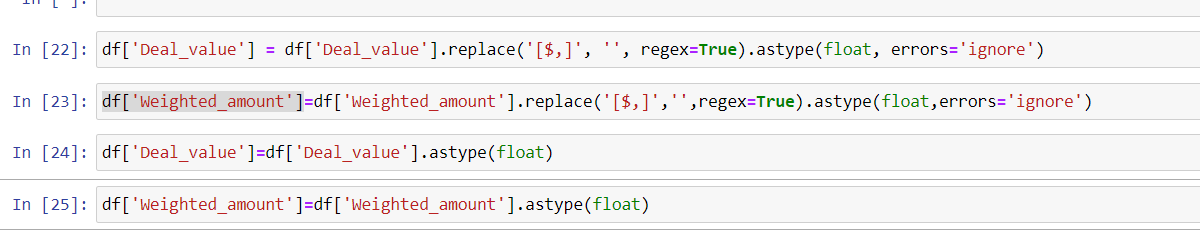
# Activity 2: Data Preparation

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 to fetch good results. This activity includes the following steps.

* + Data Cleaning
  + Handling missing values
  + Handling categorical data
  + 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.

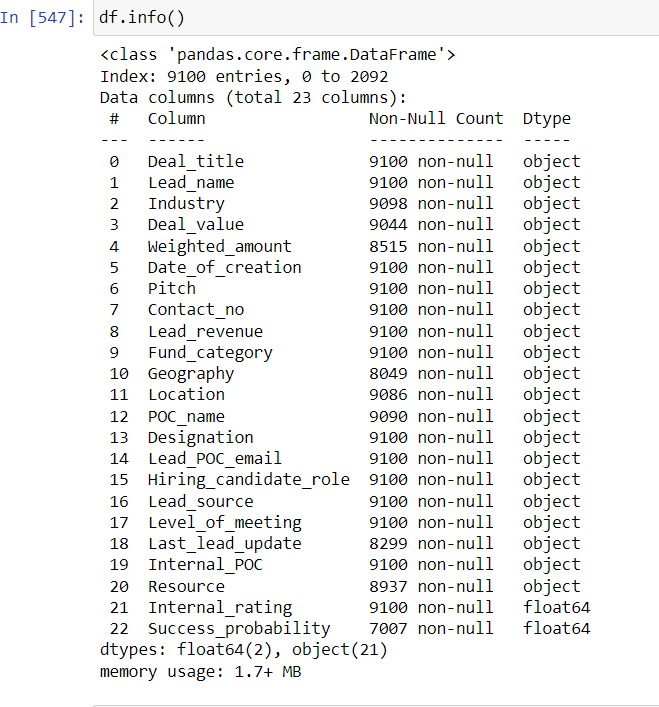
# Activity 2.1 Data Cleaning.



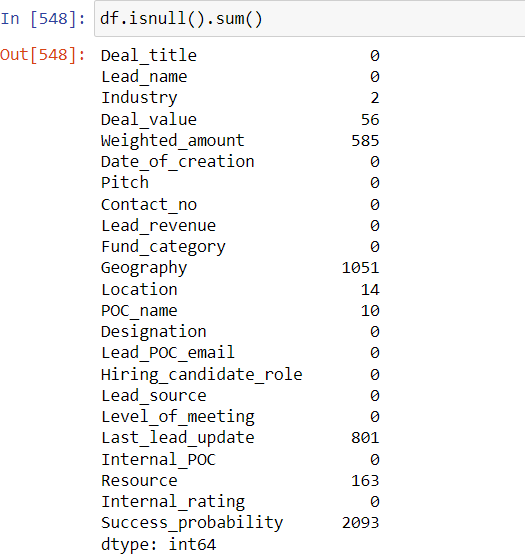
* + df1=df.drop(['Deal\_title','Date\_of\_creation','Contact\_no','Lead\_name','Lead\_POC\_email','P OC\_name'],axis=1)

# Activity 2.2: Handling missing values

* + Let’s find the shape of our dataset first. To find the data type, the df.info() function is used.

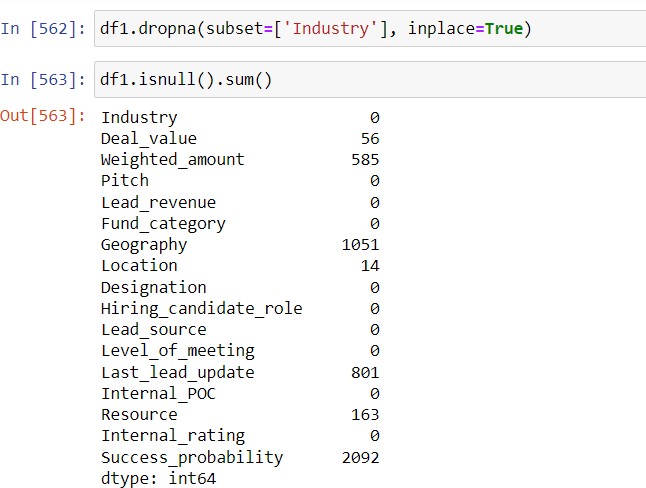


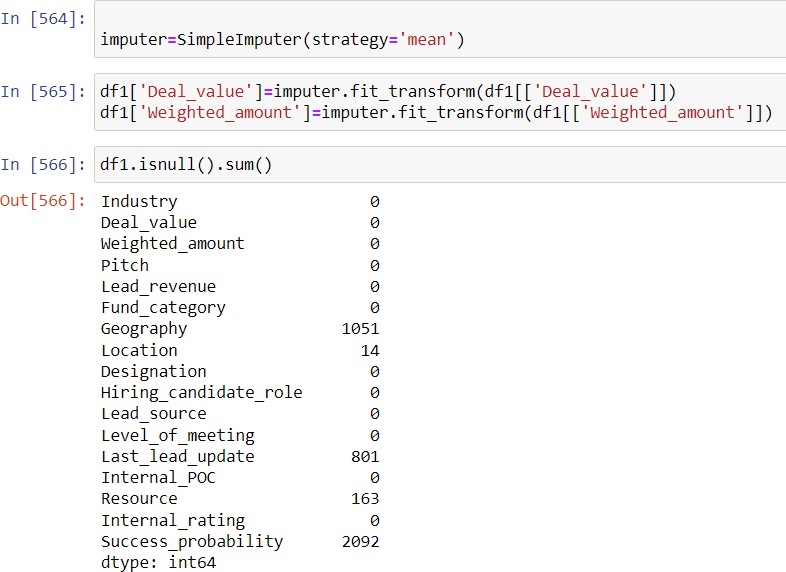
* + - For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function. From the below image we found that there are no null values present in our dataset. So we can skip handling the missing values step.

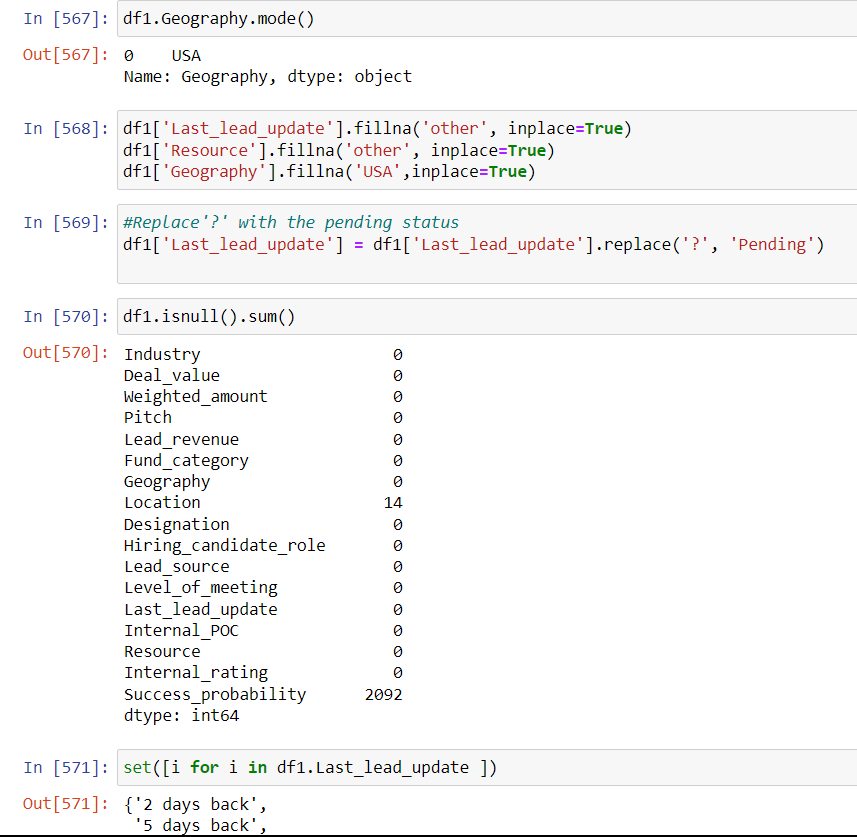


Some of the missing values we will try to replace with different methods as per the requirement.

* droping
* Replace with mean/median/mode
* Consider it as an ‘other’ category

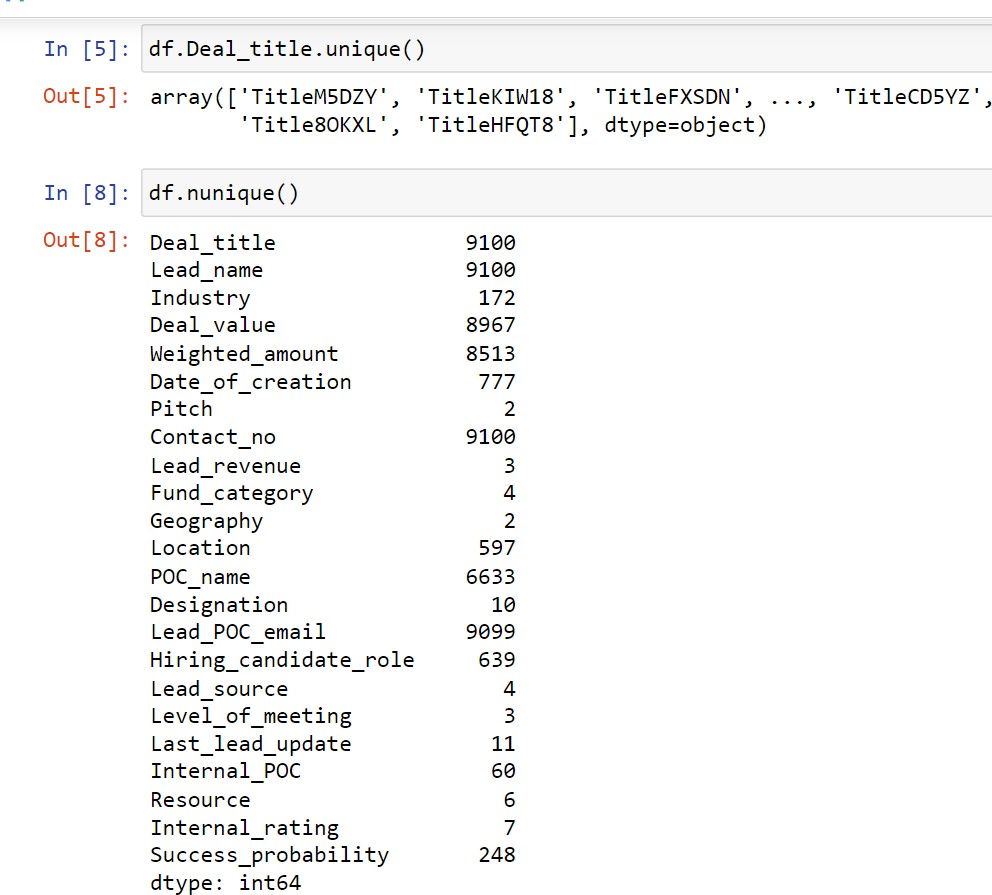






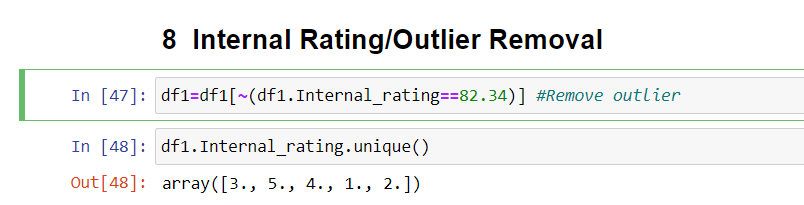
# Activity 2.3: Handling Categorical Values

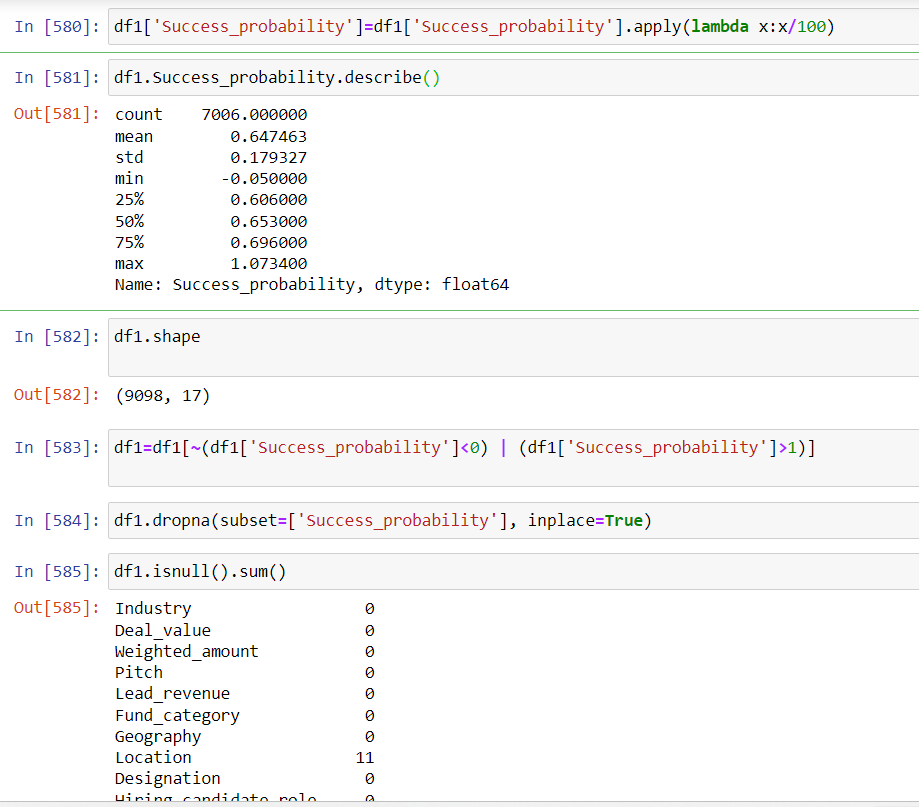
As per our dataset, we will convert the categorical to numerical at the time of model buildng



# Activity 2.4: Handling Outliers

Rating is in range [-1,5] so that’s why we will remove the outlier which exceeds the range. Also our target variable is Success\_probability it is in percentage but percentage first we convert it to proportion which exceeds the range [0,1] we consider as outlier and remove it.

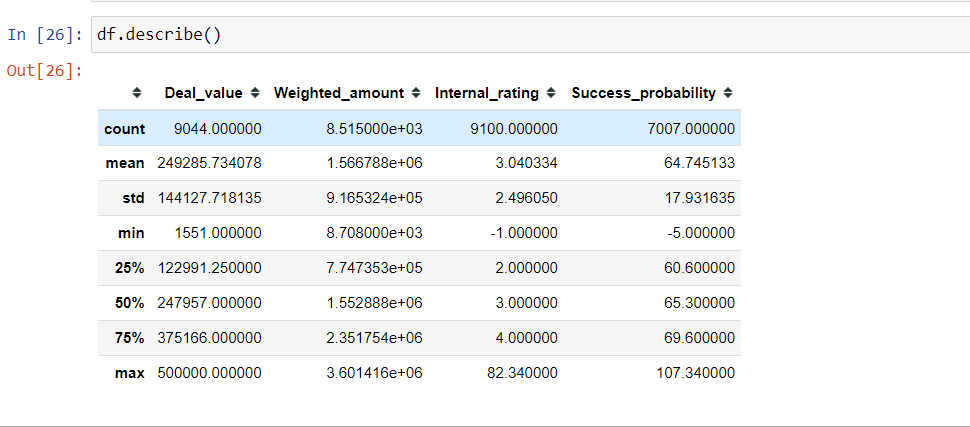




# Milestone 3: Exploratory Data Analysis

**Activity 1: Descriptive statistics**

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.

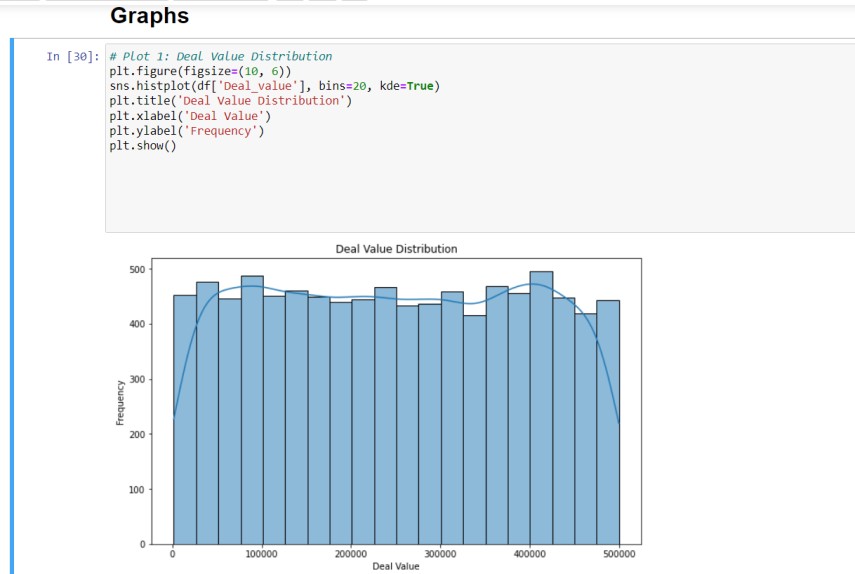


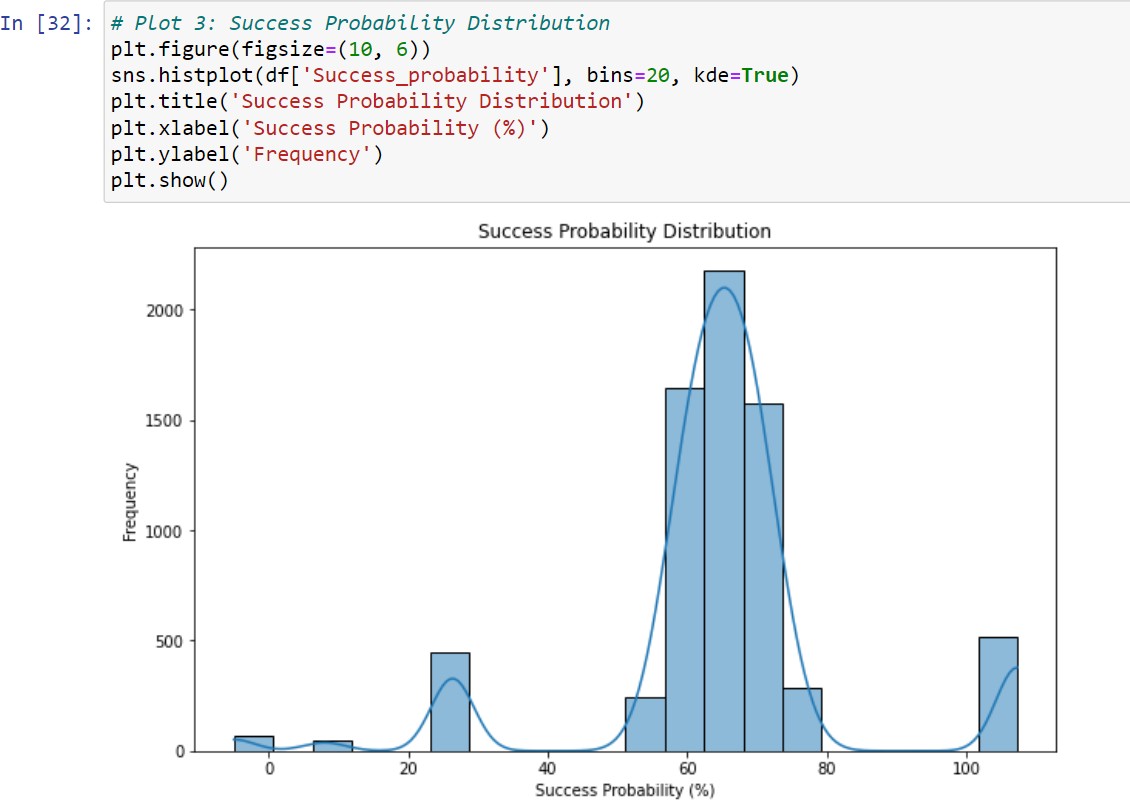
# Activity 2: Visual analysis

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 distplot and countplot.

Seaborn package provides a wonderful function distplot. With the help of distplot, we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot. 

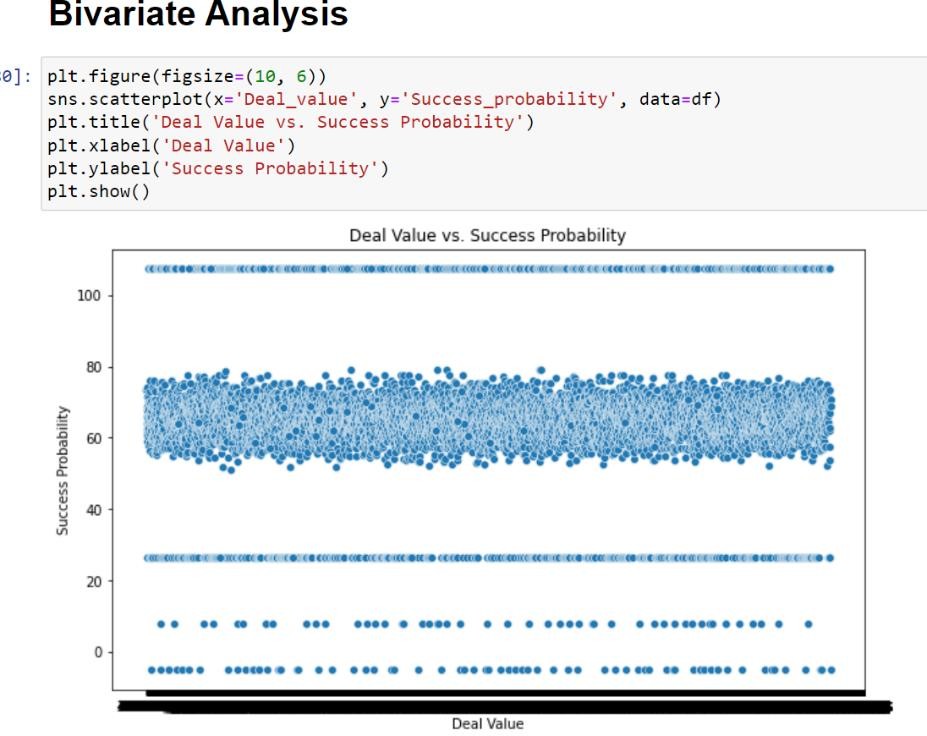


In our dataset, we have some categorical features. With the countplot function, we are going to count the unique category in those features. We have created a dummy data frame with categorical features. With for loop and subplot, we have plotted the below graph.



# Activity 2.2: Bivariate Analysis

To find the relation between two features we use bivariate analysis. Here we are visualizing the relationship between probability of success and deal\_value

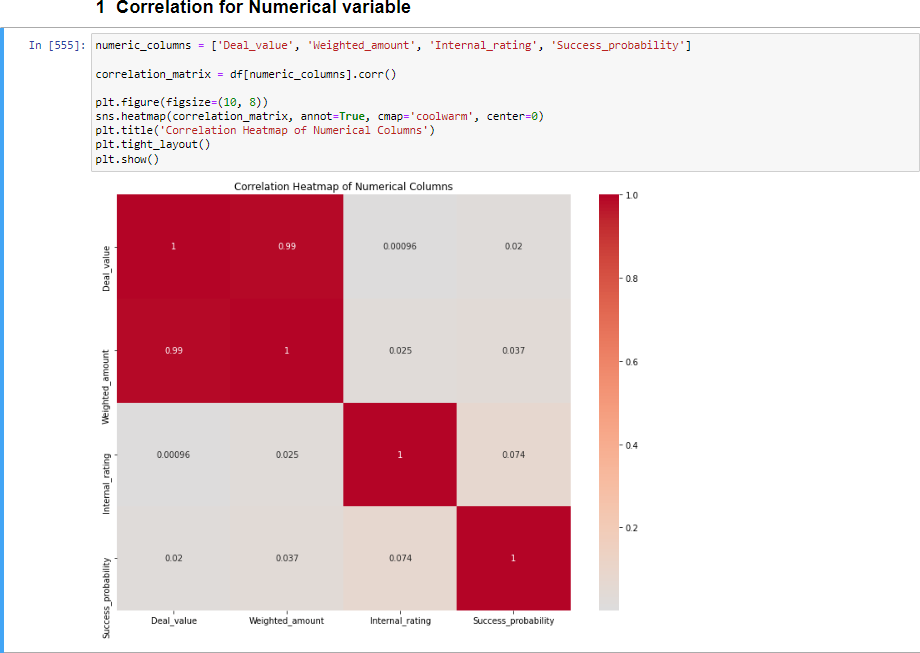


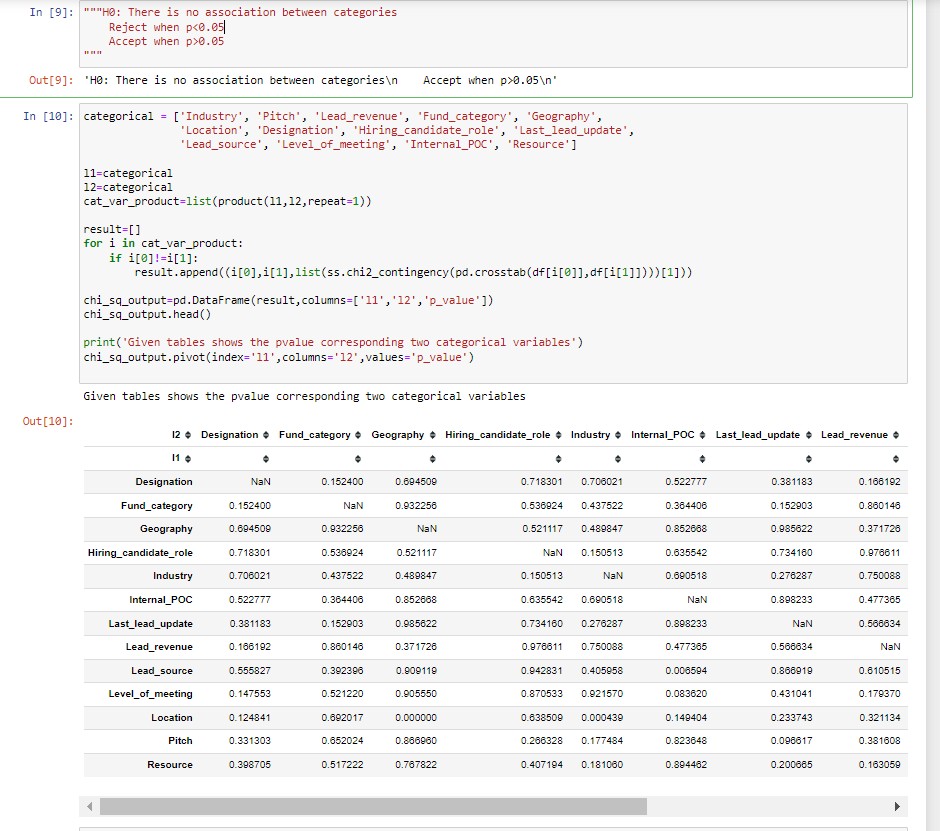
# Activity 2.3: Multivariate analysis

In simple words, multivariate analysis is to find the relation between multiple features. Here we have used a heatmap for continuous variables and the Statistical ChiSquare Test.

Null Hypothesis There is no association between the two categories We will reject it if the p-value is less than 0.05

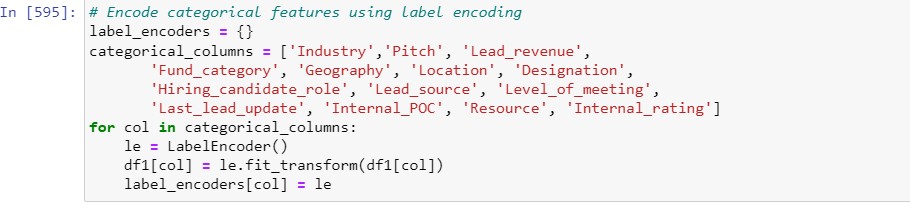
Association test in one table for Categorical variables.

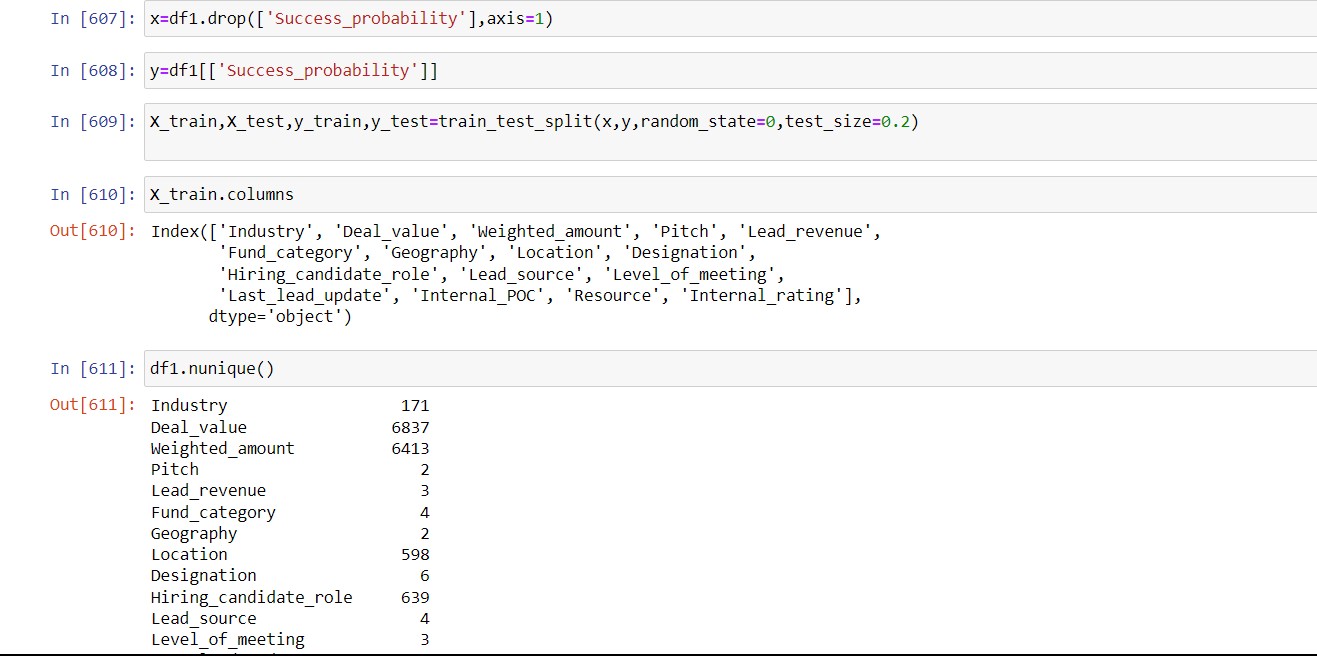




# Splitting data into train and test/ (data Preparation Encoding to category features)

First we encoding the features so that we can use them for Machine Learning 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 the x variable, df is passed by dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using the train\_test\_split() function from sklearn. As parameters, we are passing x, y, test\_size, random\_state.



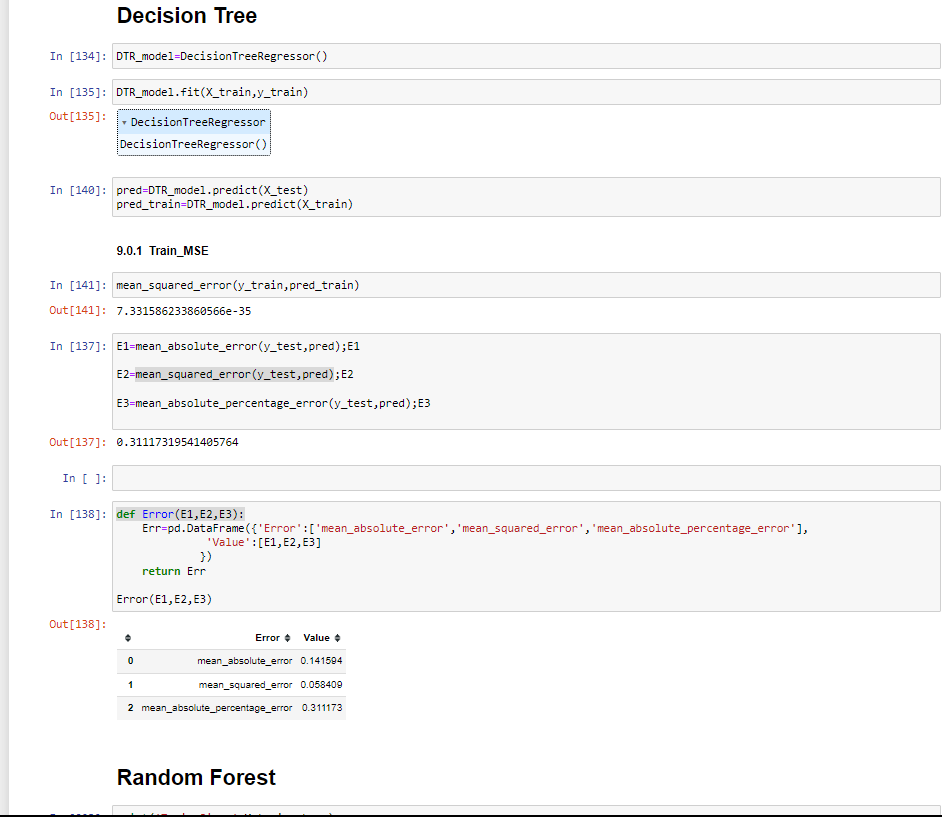
# Milestone 4: Model Building

**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 Regression algorithms. The best model is saved based on its performance.

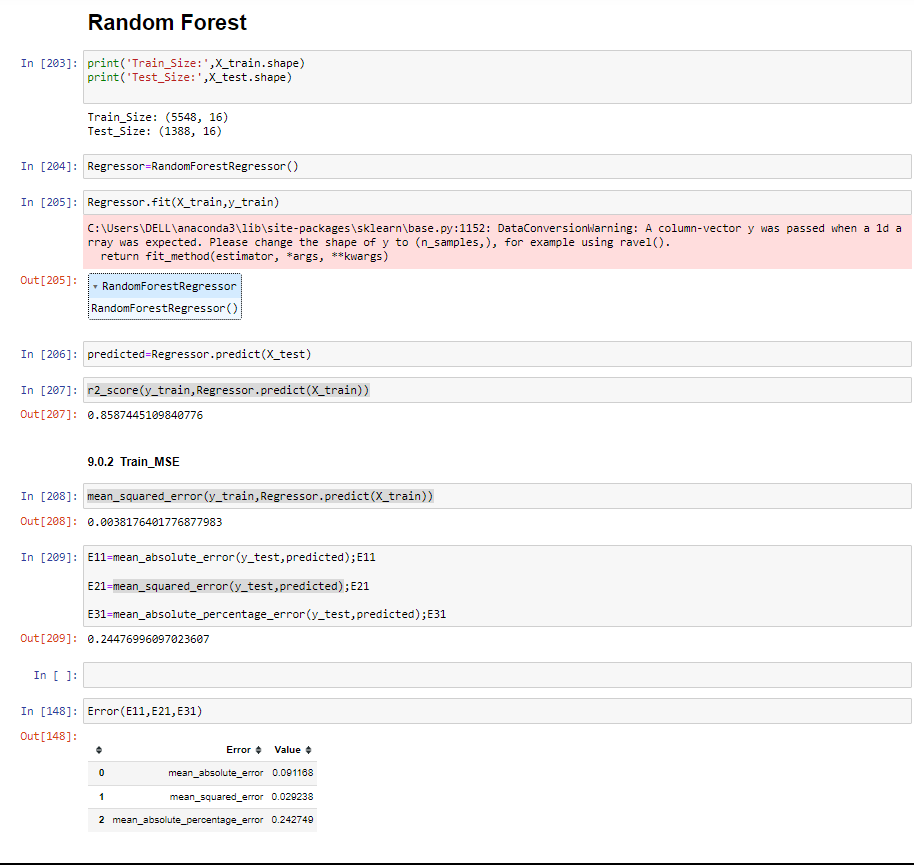
# Activity 1.1: Decision tree model

A function named decisionTree is created and train and test data are passed as the parameters. Inside the function, the DecisionTreeRegressor algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with the

.predict() function and saved in a new variable. For evaluating the model, MSE, MAE, MAPE, R\_Square.

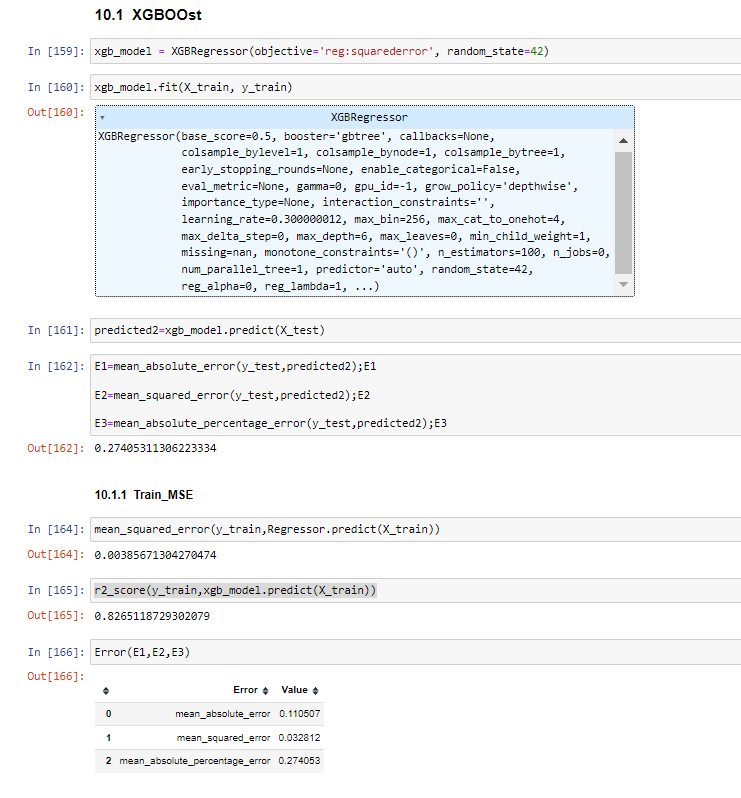
# Activity 1.2: Random forest model

A function named randomForest is created and train and test data are passed as the parameters. Inside the function, RandomForestRegressor 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. For evaluating the model, , MSE, MAE, MAPE R\_Square.



# Activity 1.3: XGBoost Regression

A function named XGBoost is created and train and test data are passed as the parameters. Inside the function, XGBoostRegressor algorithm is initialized 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, , MSE, MAE, MAPE R\_Square.



# Milestone 5: Performance Testing & Hyperparameter Tuning

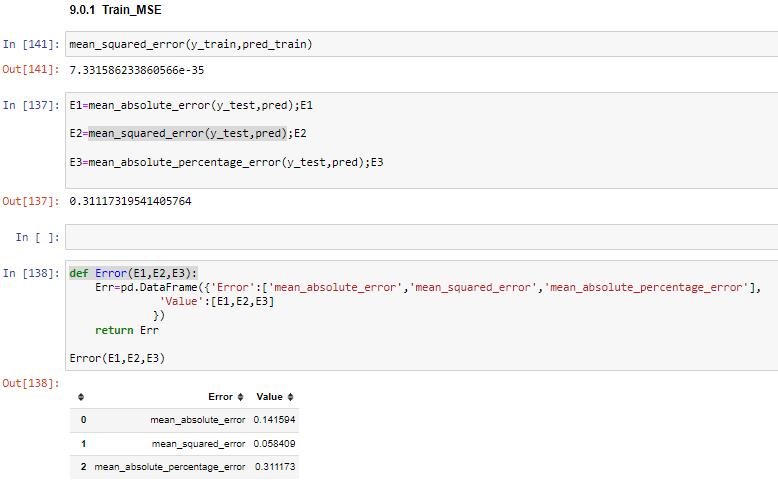
**Activity 1: Testing model with multiple evaluation metrics**

Multiple evaluation metrics means evaluating the model's performance on a test set using different performance measures. This can provide a more comprehensive understanding of the model's strengths and weaknesses. We are using evaluation metrics for Regression tasks including MSE, MAE, MAPE, R\_Square

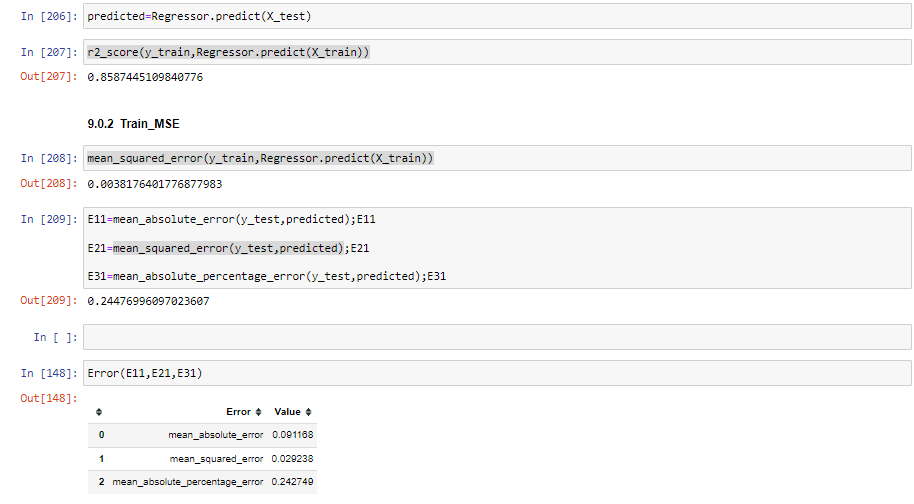
# Activity 1.1: Compare the model

From the above model, the random forest Regressor is performing well. From the below image, we can see the accuracy of the model here random forest is selected and evaluated with cross-validation. Additionally, we can tune the model with hyperparameter tuning techniques for the best model Random Forest.

**Decision Tree:**



**Random Forest:**

****

**XGBoost Regression:**

****

# Activity 1.2 parameter Tunning.

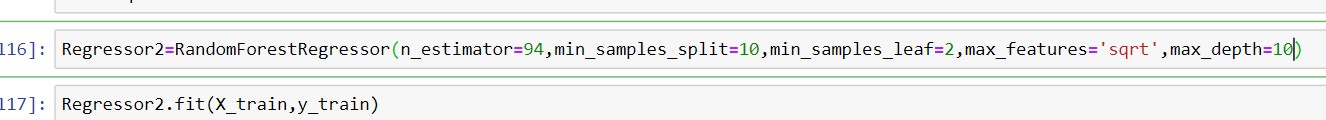
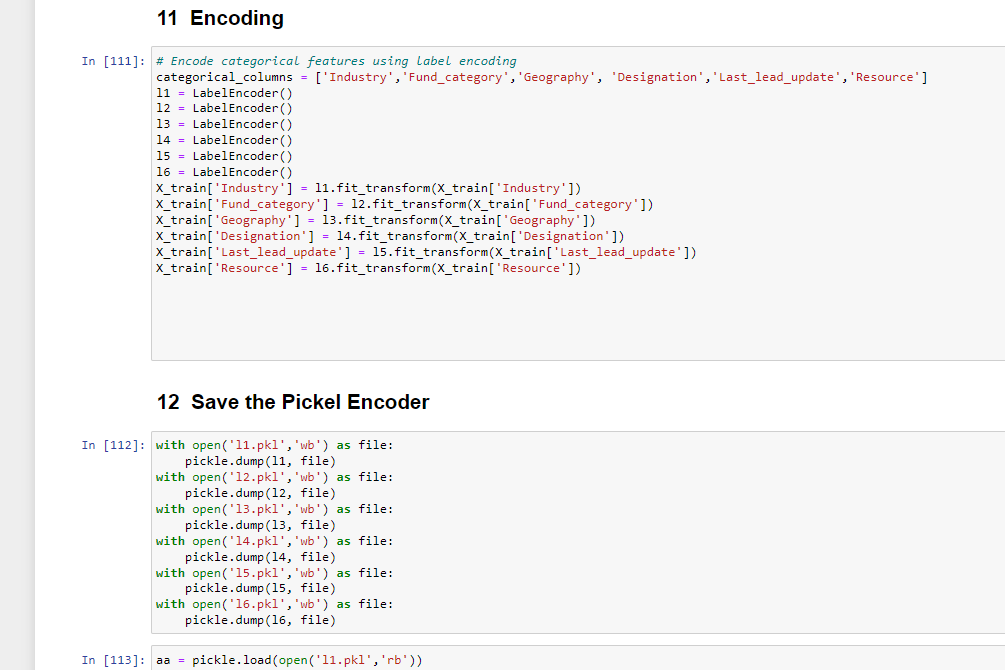


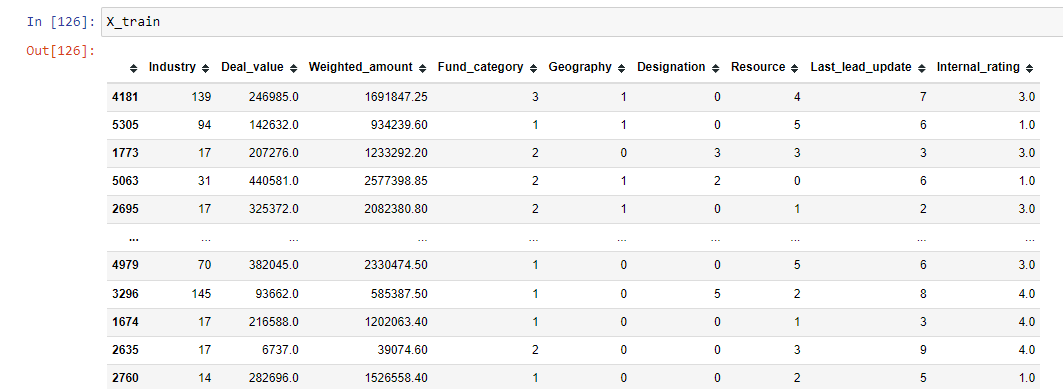
After calling the function, the results of models are displayed as output. From the all models, random forest is performing well

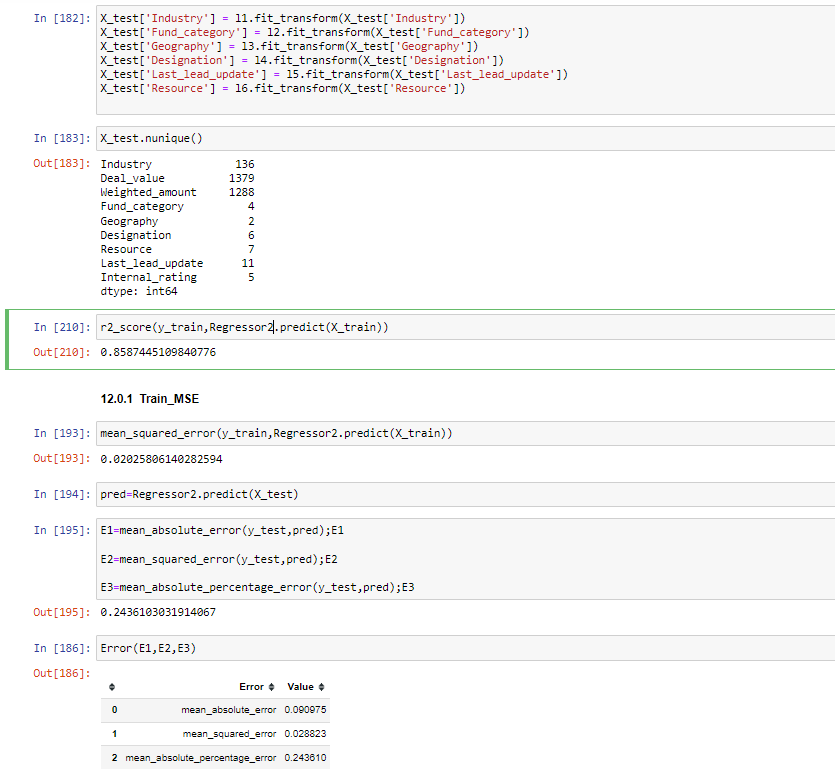
# Activity 2: Best Model.

We can not take all featues for the deployement so we select best features by hyperparameter tunnning









# 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 and saving its weights and configuration. 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.



# 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 Pages:

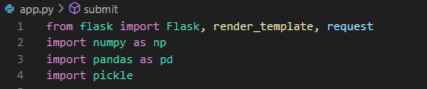
For this project create HTML files namely

* + index.html
  + inner-page.html

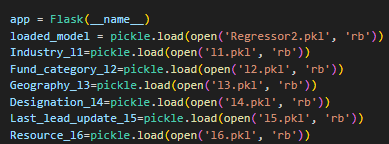
and save them in the templates folder. Refer to this [link](https://drive.google.com/drive/folders/1K-C2uvRstV8x6bwsOY_BWBvXdzjbbfPF?usp=share_link) for templates.

# Activity 2.2: Build Python code:

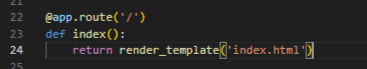
Import the libraries



Load the saved model. Importing the flask module for prediction and Encoding 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.



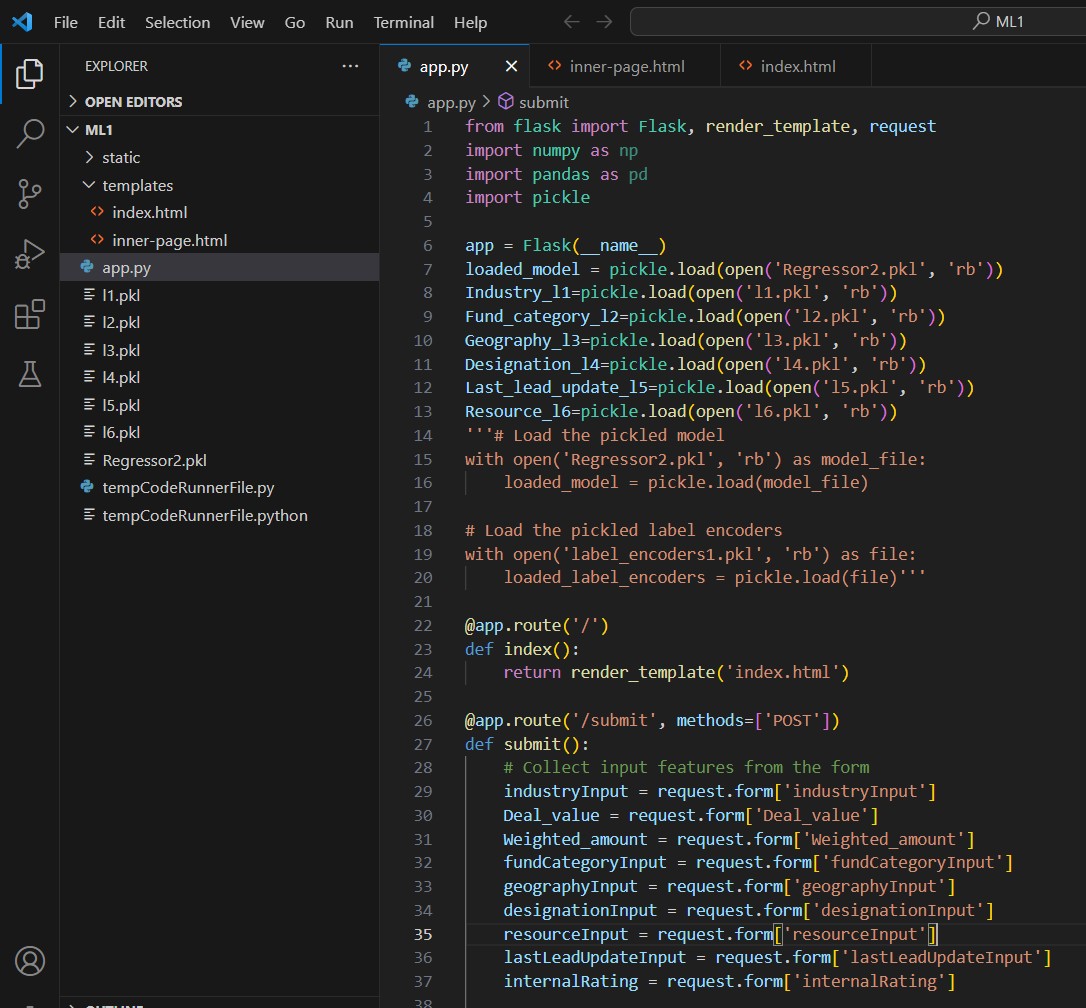
Render HTML page:

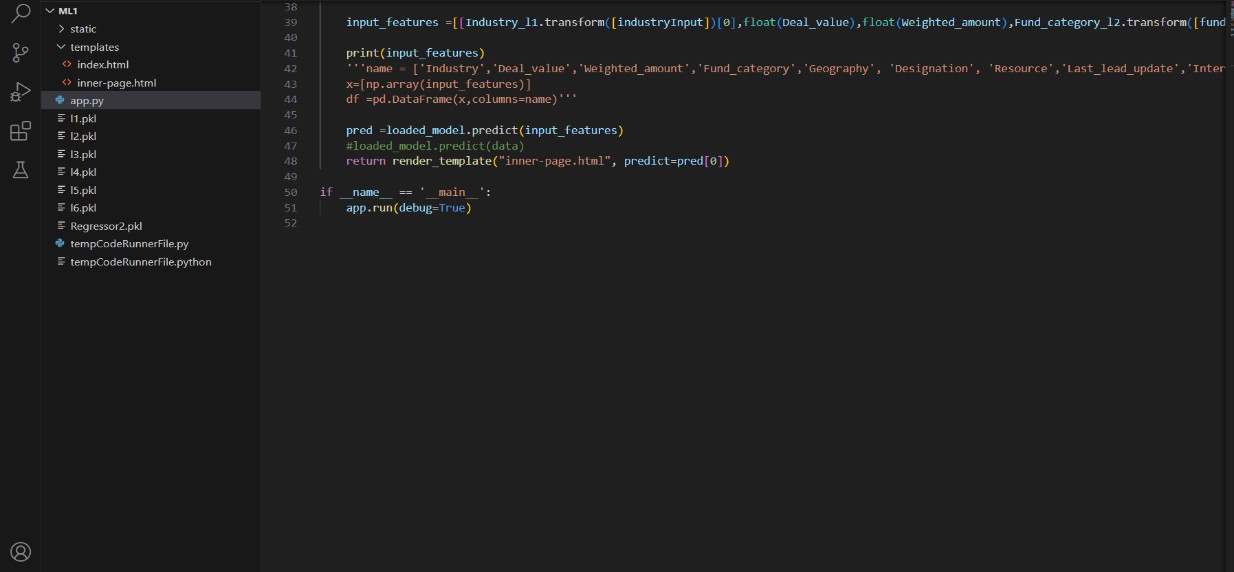


Here we will be using a declared constructor to route to the HTML page which we have created earlier.

In the above example, ‘/’ URL is bound with the home.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.

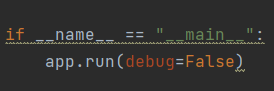
Retrieves the value from UI:





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:

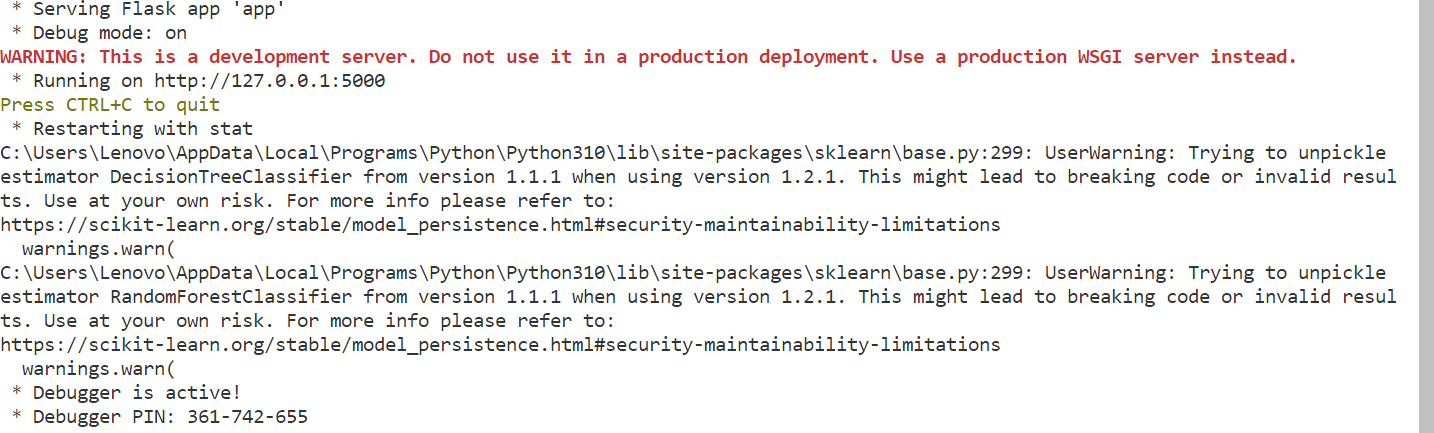


# Activity 2.3: Run the web application

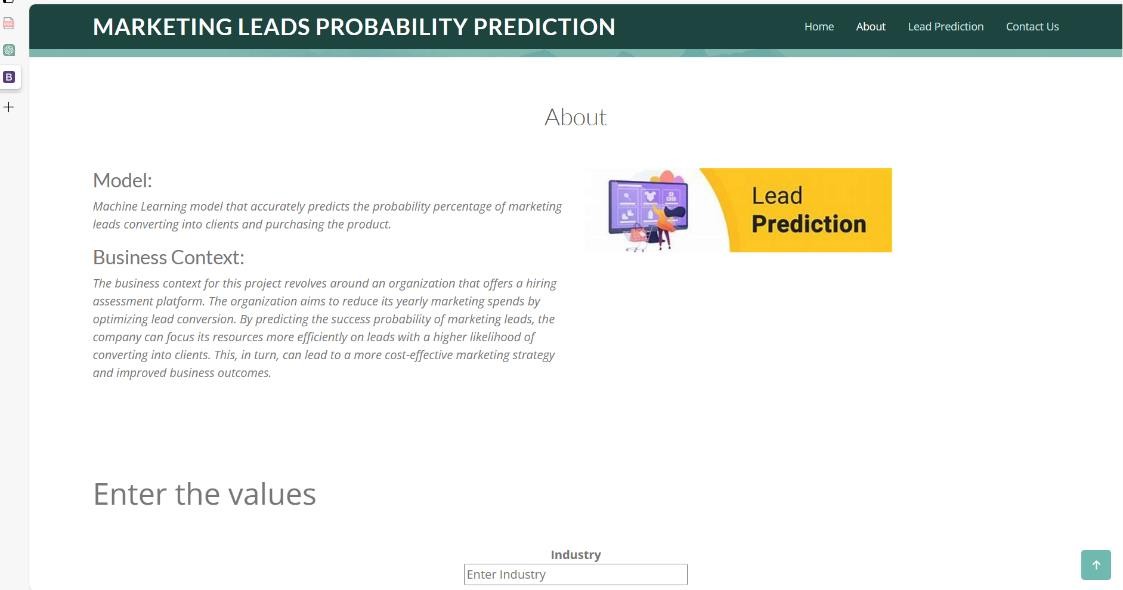
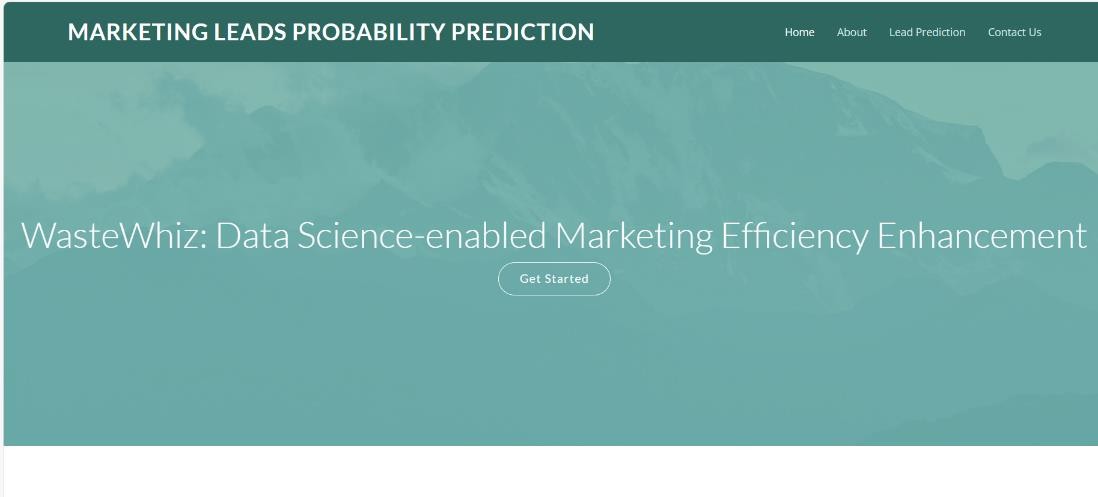
Open anaconda prompt from the start menu Navigate to the folder where your python script is. Now type “python app.py” command

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.

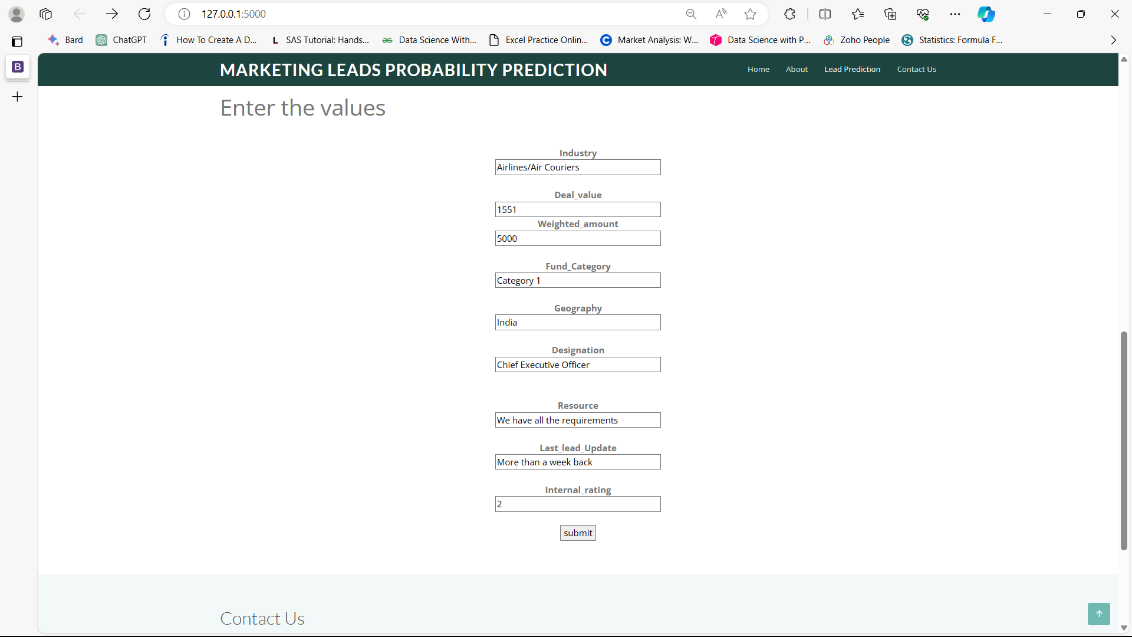


Now, Go to the web browser and write the localhost url (http://127.0.0.1:5000) to get the below result

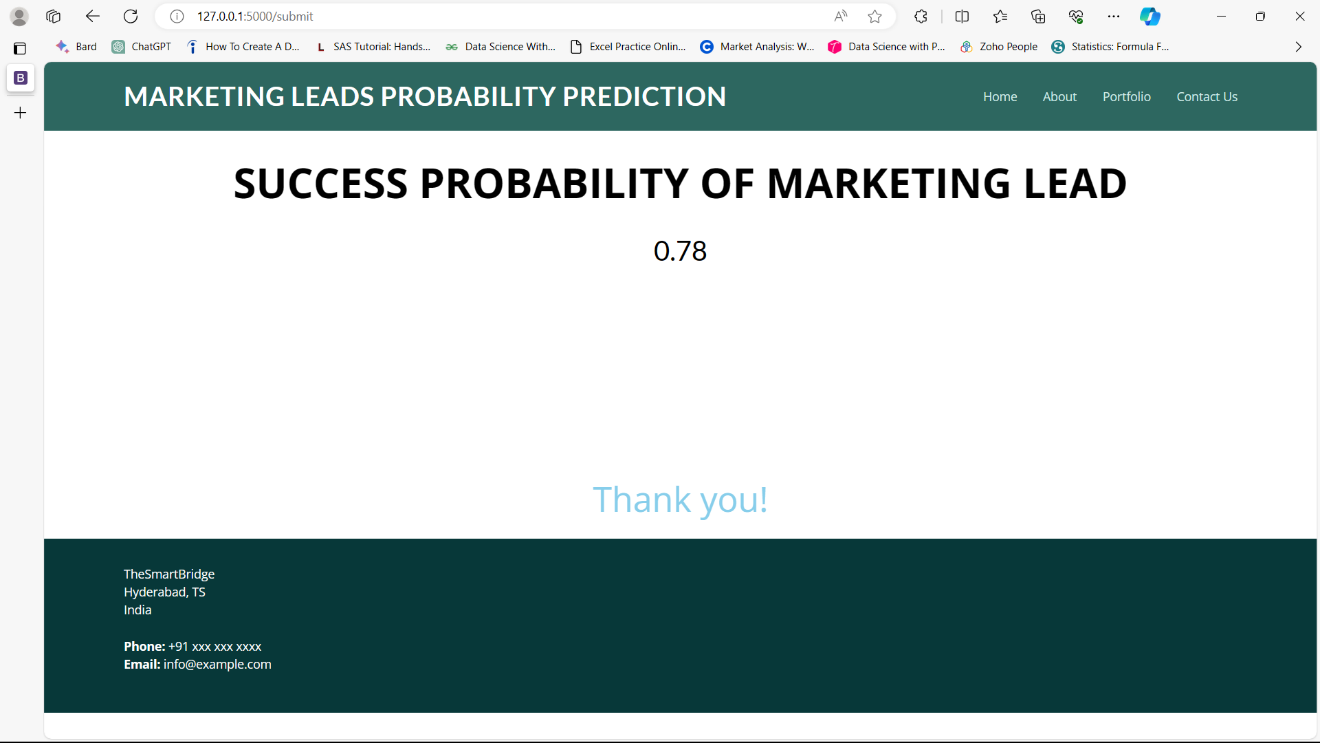


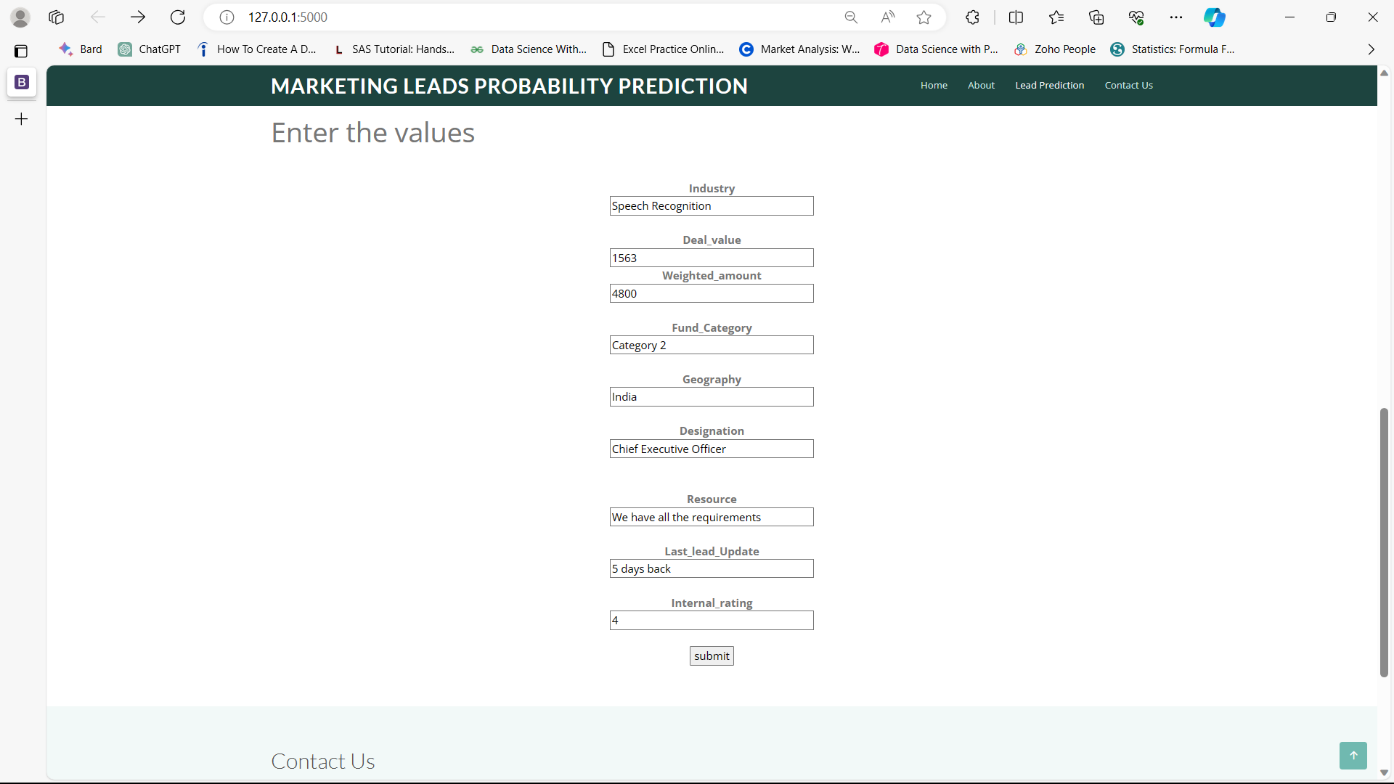
Now when you click on the ‘Lead Prediction’ button from the top right corner you will get redirected

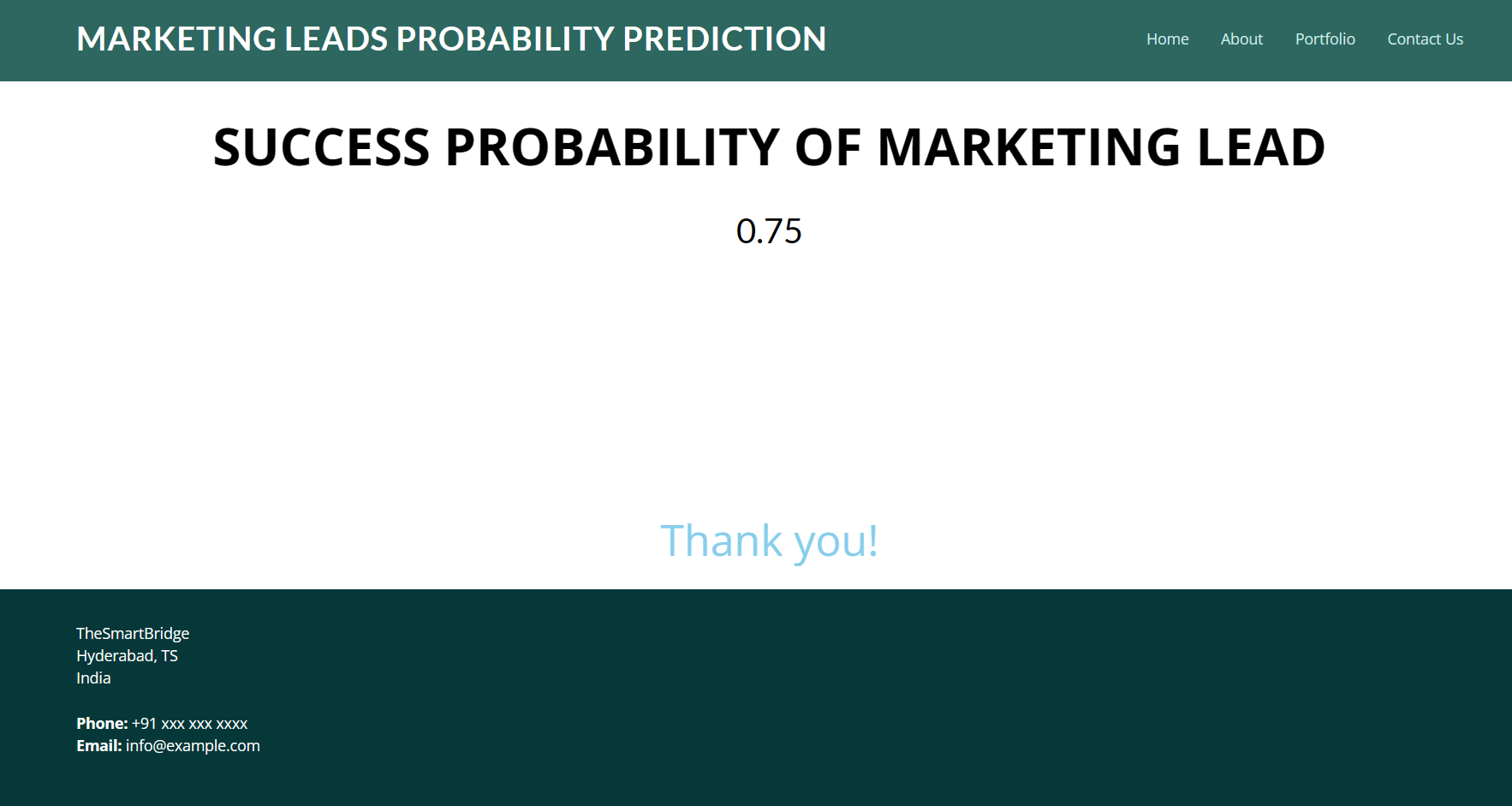
to predict.html

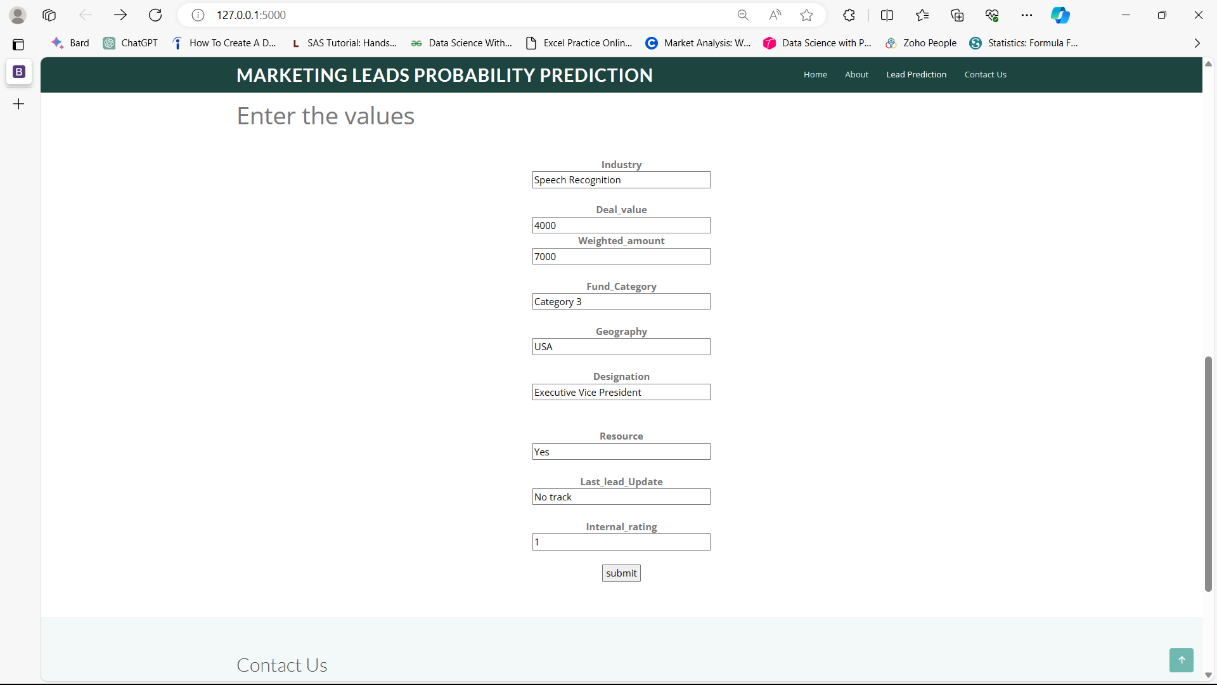


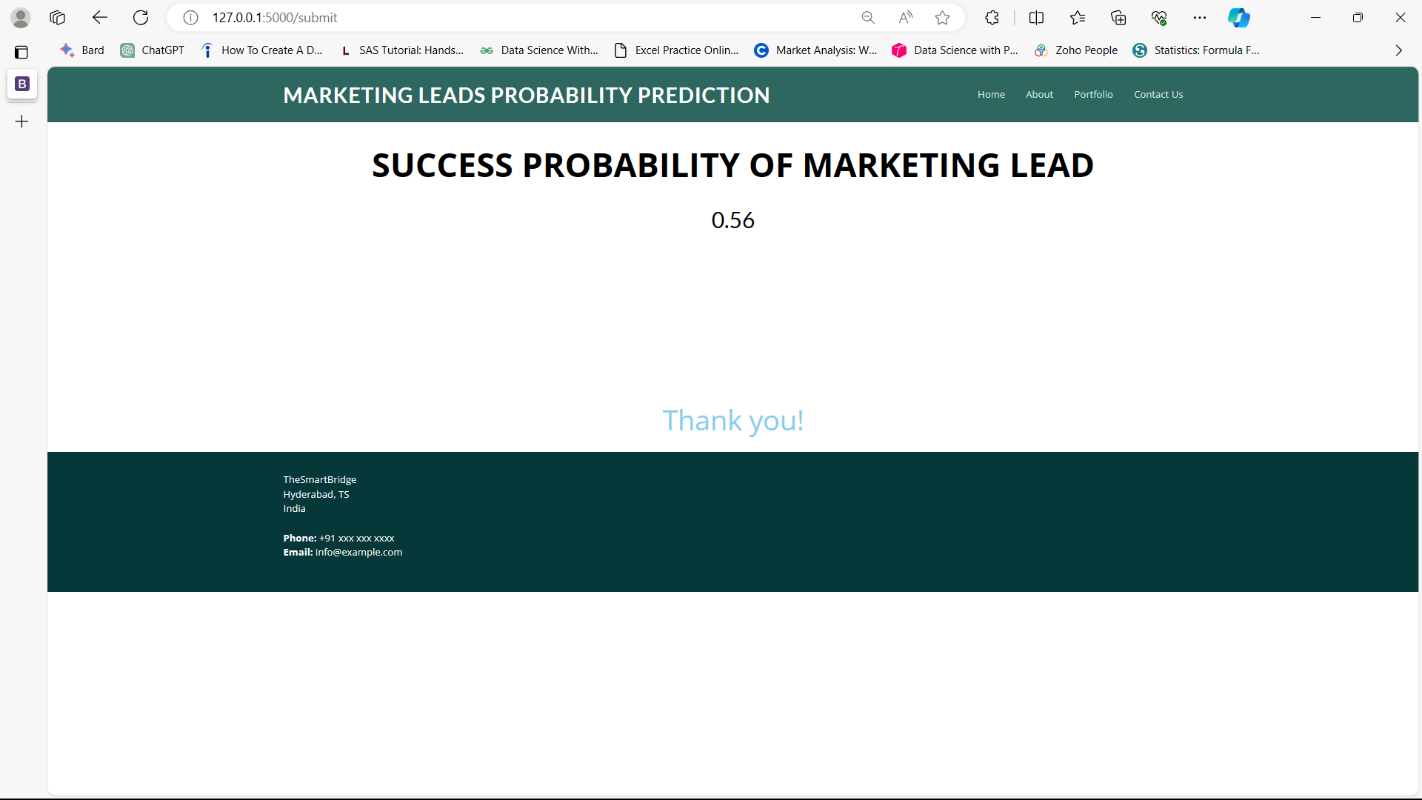
Lets look at how our predict.html file looks like when you click on the prediction button from the lower right below you will get redirected to the submit.html page with output.











# 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