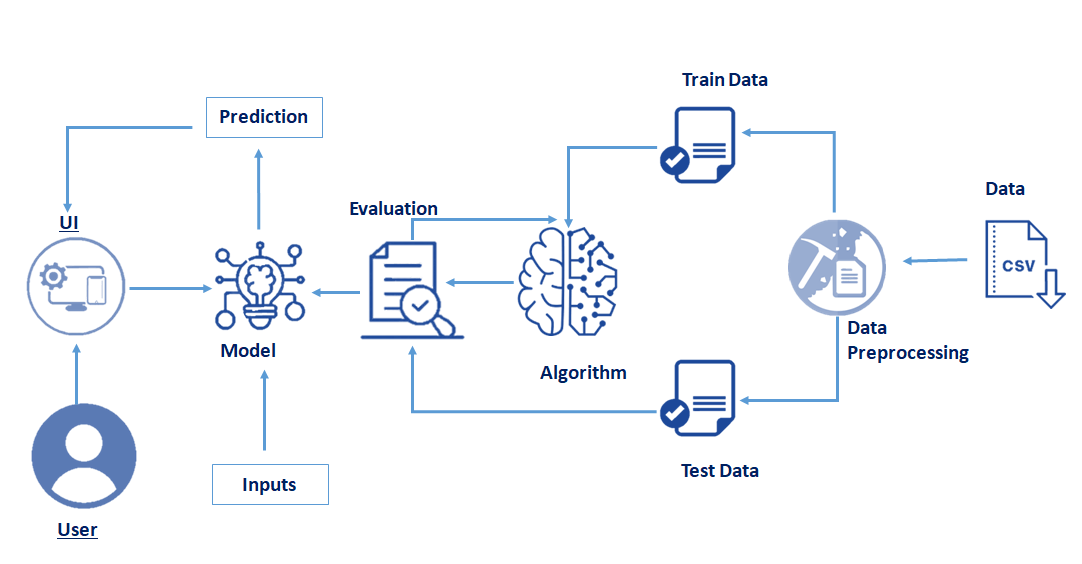
**GDP Analysis**

**Project Description:**

Gross domestic product (GDP) is the total monetary or market value of all the finished goods and services produced within a country’s borders in a specific time period. As a broad measure of overall domestic production, it functions as a comprehensive scorecard of a given country’s economic health. The calculation of a country’s GDP encompasses all private and public consumption, government outlays, investments, additions to private inventories, paid-in construction costs, and the foreign [balance of trade](https://www.investopedia.com/terms/b/bot.asp). (Exports are added to the value and imports are subtracted).

**Technical Architecture:**

****

**Prerequisites:**

**To complete this project, you must require the following software’s, concepts, and packages**

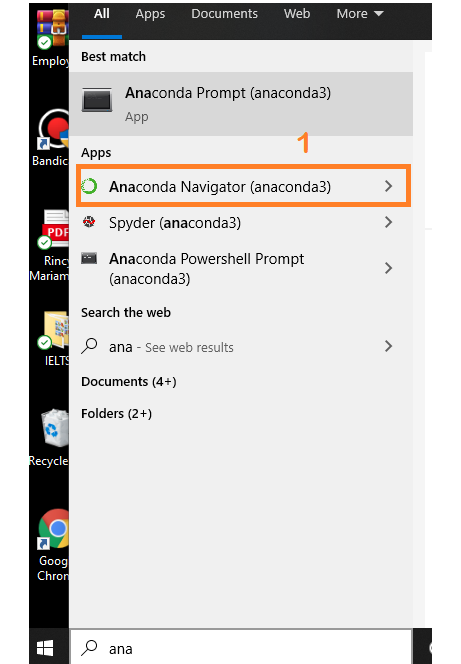
* **Anaconda navigator:**
  + Refer to the link below to download anaconda navigator
  + **Link: https://www.youtube.com/watch?v=5mDYijMfSzs**
* **Python packages:**

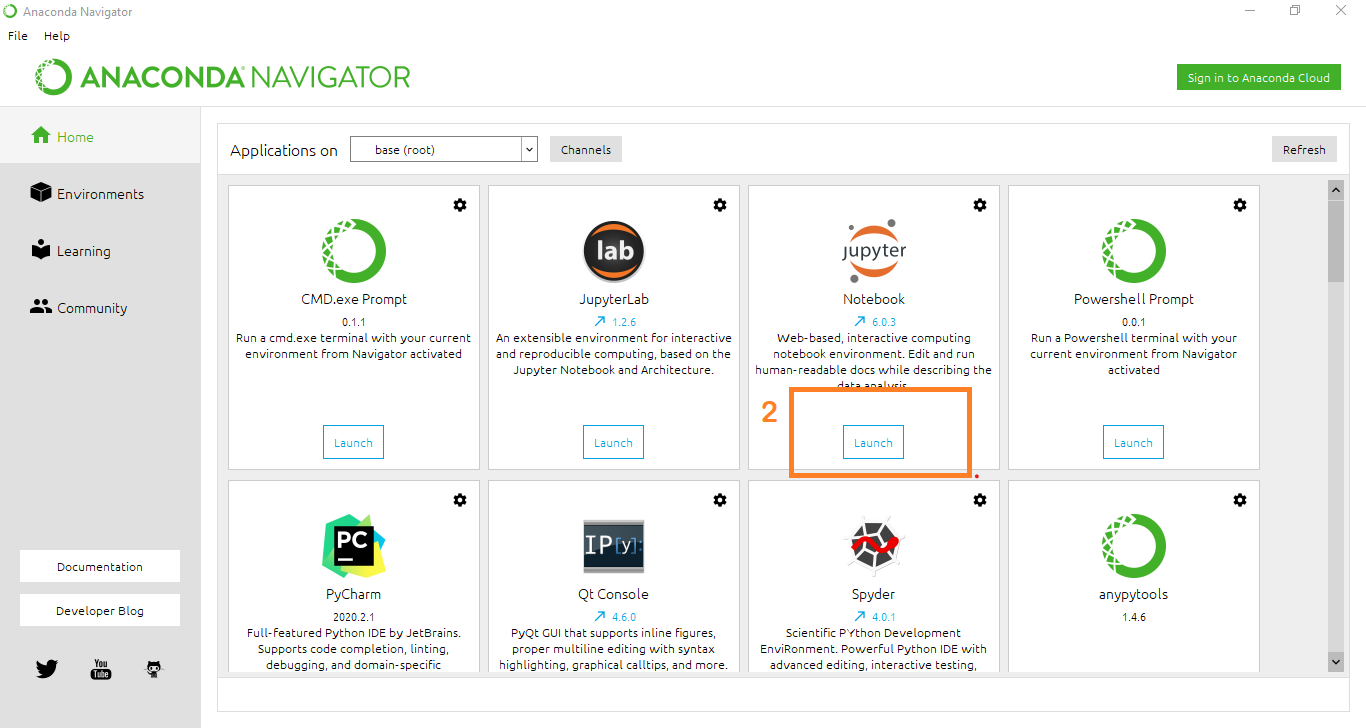
Open anaconda prompt as administrator.

* Type “pip install numpy” and click enter.
* Type “pip install pandas” and click enter.
* Type “pip install matplotlib” and click enter.
* Type “pip install sci-kit-learn” and click enter.
* Type “pip install Flask” and click enter.

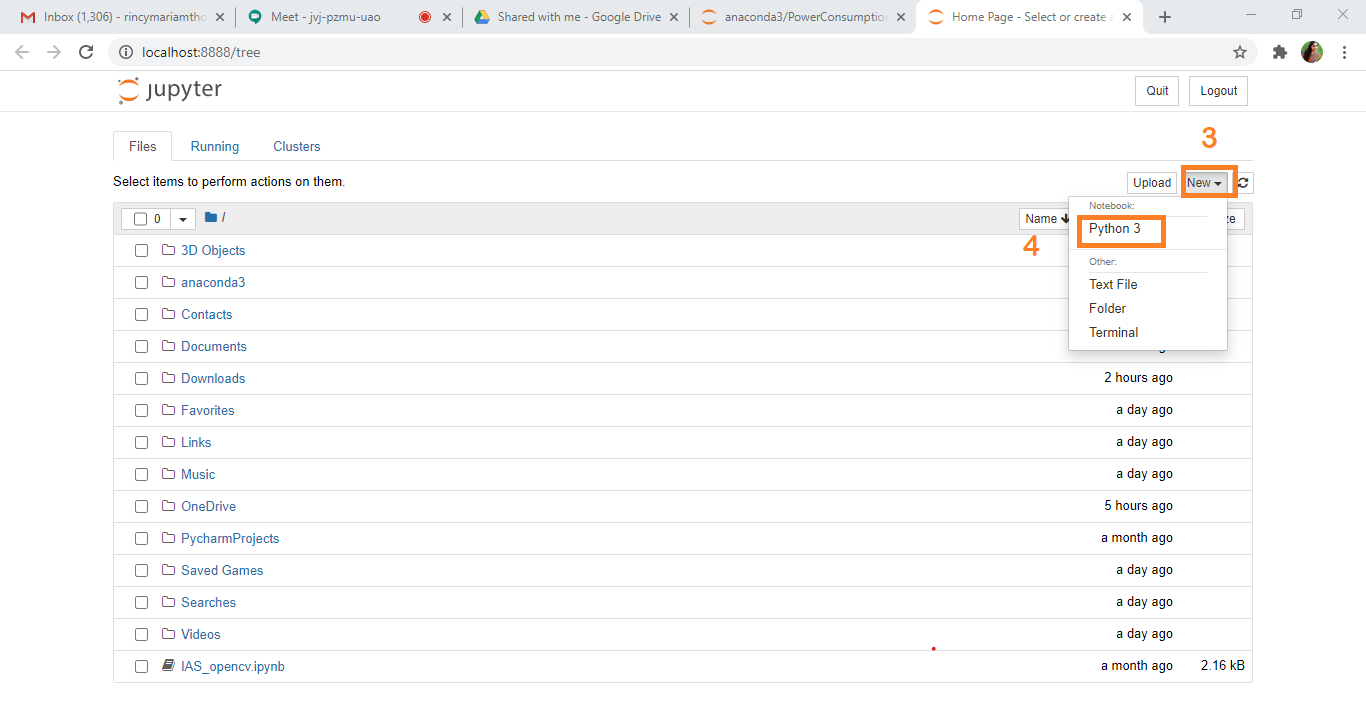
The above steps allow you to install the packages in the anaconda environment

* **Launch Jupyter**
  + Search for Anaconda Navigator and open Launch Jupyter notebook.





* Then you will be able to see that the jupyter notebook runs on local host:8888.
* To Create a new file Go to New 🡪Python3. The file in the jupyter notebook is saved with the .ipynb extension.



* + Flask Basics: <https://www.youtube.com/watch?v=lj4I_CvBnt0>

**Project Objectives:**

By the end of this project:

* You’ll be able to understand the problem to classify if it is a regression or a classification kind of problem.
* You will be able to know how to pre-process/clean the data using different data pre-processing techniques.
* You will be able to analyze or get insights into data through visualization.
* Applying different algorithms according to the dataset and based on visualization.
* You will be able to know how to find the accuracy of the model.
* You will be able to know how to build a web application using the Flask framework.

**Project Flow:**

* User interacts with the UI (User Interface) to enter the input values
* Entered input values are analyzed by the model which is integrated
* Once the model analyses the input the prediction is showcased on the UI

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

* Data Collection.
* Collect the dataset or Create the dataset
* Data Pre-processing.
  + Import the Libraries.
  + Importing the dataset.
  + Checking for Null Values.
  + Data Visualization.
  + Taking care of Missing Data.
  + Splitting Data into Train and Test.
* Model Building
  + Import the model building Libraries
  + Initializing the model
  + Training and testing the model
  + Evaluation of Model
  + Save the Model
* Application Building
  + Create an HTML file
  + Build a Python Code

**Project Structure:**

Create a Project folder that­­ contains files as shown below

Graphical user interface, text, application

Description automatically generated

* A python file called app.py for server-side scripting.
* We need the model which is saved and the saved model in this content is **GDP.pkl**
* Templates folder which contains index.html file, home.html file, gdp\_pred.html file.

**Milestone 1: Data Collection:**

ML depends heavily on data, without data, an “AI” can’t learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training **data set.**It is the actual **data set** used to train the model for performing various actions.

**Activity1: Download the dataset**

You can collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository, etc.

Please refer to the link given below to download the data set and to know about the dataset

[**https://drive.google.com/file/d/1tR-ykruA8s6vbA7EZv\_8xPVN9r3C3TH7/view?usp=sharing**](https://drive.google.com/file/d/1tR-ykruA8s6vbA7EZv_8xPVN9r3C3TH7/view?usp=sharing)

**Milestone 2: Data Pre-processing**

Data Pre-processing includes the following main tasks

* + Import the Libraries.
  + Importing the dataset.
  + Checking for Null Values.
  + Data Visualization.
  + Splitting Data into Train and Test.

**Activity 1: Import Necessary Libraries**

* + It is important to import all the necessary libraries such as pandas, NumPy, matplotlib.
  + **Numpy** - It is an open-source numerical Python library. It contains a multi-dimensional array and matrix data structures. It can be used to perform mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.
  + **Pandas**- It is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language.
  + **Seaborn**- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
  + **Matplotlib**- Visualisation with python. It is a comprehensive library for creating static, animated, and interactive visualizations in Python
  + **Sklearn** – which contains all the modules required for model building

Text

Description automatically generated

**Activity 2: Importing the Dataset**

* You might have your data in .csv files, .excel files
* Let’s load a .csv data file into pandas using **read\_csv() function.**We will need to locate the directory of the CSV file at first (it’s more efficient to keep the dataset in the same directory as your program).
* If your dataset is in some other location, Then

**Data=pd.read\_csv(r”File\_location/datasetname.csv”)**

A screenshot of a computer

Description automatically generated with medium confidence

**Note:**r stands for "raw" and will cause backslashes in the string to be interpreted as actual backslashes rather than special characters.

* If the dataset is in the same directory of your program, you can directly read it, without giving raw as r.
* Our Dataset world.csv contains the following Columns:-
* Country - different-different country.
* Population – Population-based on the country.
* Area(sq. mi.) – Area in a square meter.
* Pop. Density (per sq. mi) = Population density per square meter.
* Coastline(coast/area ratio), net migration etc.
* Traffic volume – GDP ($ per capita)

The output column to be predicted is **GDP**. Based on the input variables we predict the volume of the traffic. The predicted output gives them a fair idea about the GDP of the country.

**Activity 3: Analyse the data**

* head() method is used to return top n (5 by default) rows of a DataFrame or series.

Graphical user interface, text, application

Description automatically generated

From the data, we infer that there are only decimal values or categorical values

* info() gives information about the data

A screenshot of a computer

Description automatically generated with medium confidence

**Activity 4: Handling Missing Values**

1. After loading it is important to check the complete information of data as it can indicate many of the hidden information such as null values in a column or a row

2. Check whether any null values are there or not. if it is present then the following can be done,

Text

Description automatically generated

3.Imputing data using Imputation method in pandas i.e median(), mode(), max().

Filling NaN values with mean, median, and mode using the above method.

**Activity 5: Data Visualisation**

* Data visualization is where a given data set is presented in a graphical format. It helps the detection of patterns, trends, and correlations that might go undetected in text-based data.
* Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. Even the most sophisticated machine learning models will perform poorly on data that wasn’t visualized and understood properly.
* To visualize the dataset we need libraries called Matplotlib and Seaborn.
* The Matplotlib library is a Python 2D plotting library that allows you to generate plots, scatter plots, histograms, bar charts, etc.

Let’s visualize our data using Matplotlib and the seaborn library.

Before diving into the code, let's look at some of the basic properties we will be using when plotting.

**xlabel:** Set the label for the x-axis.

**ylabel:** Set the label for the y-axis.

**title:** Set a title for the axes.

**Legend:** Place a legend on the axes.

1. Heat map with corr() gives the correlation between the columns with an interactive visual representation.

Chart

Description automatically generated

* Correlation strength varies based on color, lighter the color between two variables, more the strength between the variables, darker the color displays the weaker correlation
* We can see the correlation scale values on the left side of the above image

**2. Barplot :- sns.barplot()**

It can help us to plot categorical variables with numeric variables. You can see on the y-axis numeric values and the x-axis categorical values.

The output is as shown below

A picture containing text

Description automatically generated

3. Correlation ranges is -1 to +1, where -1 means negatively correlated and +1 means positively correlated.

* Inside the below code we are just converting negatively correlated to positive.
* Then we are dropping some highly correlated features.

Text

Description automatically generated

4. We have just checked the correlation concerning the GDP column and finally we are plotting in scatter plot.

Text

Description automatically generated

5. Applying filter.

Graphical user interface

Description automatically generated

**6. Encoding:- We have some categorical features for that we need to apply some encoding techniques. We are using label encoder here.**

**Label Encoder refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.**

Text

Description automatically generated

**7. Editing features the name, we need to give the features name as per python variable declaration rule.**

Text

Description automatically generated

**Activity 6: Splitting the data into Train and Test**

* When you are working on a model and you want to train it, you have a dataset. But after training, we have to test the model on some test datasets. For this, you will a dataset which is different from the training set you used earlier. But it might not always be possible to have so much data during the development phase. In such cases, the solution is to split the dataset into two sets, one for training and the other for testing.
* But the question is, how do you split the data? You can’t possibly manually split the dataset into two sets. And you also have to make sure you randomly split the data. To help us with this task, the Scikit library provides a tool, called the Model Selection library. There is a class in the library which is, **‘**[**train\_test\_split**](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html)**.’** Using this we can easily split the dataset into the training and the testing datasets in various proportions.
* The train-test split is a technique for evaluating the performance of a machine learning algorithm.
* **Train Dataset**: Used to fit the machine learning model.
* **Test Dataset**: Used to evaluate the fit machine learning model.
* In general you can allocate 70% of the dataset to the training set and the remaining 30% to the test set. We will create 4 sets— X\_train (training part of the matrix of features), X\_test (test part of the matrix of features), Y\_train (training part of the dependent variables associated with the X train sets, and therefore also the same indices), Y\_test (test part of the dependent variables associated with the X test sets, and therefore also the same indices.
* There are a few other parameters that we need to understand before we use the class:
* **test\_size** — this parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset
* **train\_size** — you have to specify this parameter only if you’re not specifying the test\_size. This is the same as test\_size, but instead, you tell the class what percent of the dataset you want to split as the training set.
* **random\_state** — here you pass an integer, which will act as the seed for the random number generator during the split. Or, you can also pass an instance of the Random\_state class, which will become the number generator. If you don’t pass anything, the Random\_state instance is used by np. random will be used instead.
* Now split our dataset into train set and test using train\_test\_split class from sci-kit learn library.

**from sklearn import model\_selection**

Text

Description automatically generated

Text

Description automatically generated

**Milestone 3: Model Building:**

The model building includes the following main tasks

* + Import the model building Libraries
  + Initializing the model
  + Training and testing the model
  + Evaluation of Model
  + Save the Model

**Activity 1: Training and Testing the Model**

* Once after splitting the data into train and test, the data should be fed to an algorithm to build a model.
* There are several Machine learning algorithms to be used depending on the data you will process, such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have maybe Classification algorithms are Regression algorithms.

1. Linear Regression

2. Decision Tree Regressor

3. Random Forest Regressor etc.

**Steps in Building the model:-**

* **Initialize the model**
* **Fit the models with x\_train and y\_train**
* **Predict the y\_train values and calculate the accuracy**
* **After predicting we will find the r-square value of each model**
* **Predict the y\_test values and calculate the r-square value**

*Multiple* **Linear Regression:-** *Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable.*

Text

Description automatically generated

**After considering both r squared values of test and train we concluded that random forest regressor is giving the better value, it can explain the 97% of the data in train values**

We’re going to use x\_train and y\_train obtained above in the train\_test\_split section to train our Random forest regression model. We’re using the fit method and passing the parameters as shown below.

We are using the algorithm from Scikit learn library to build the model as shown below,

Once the model is trained, it’s ready to make predictions. We can use the **predict** method on the model and pass **x\_test** as a parameter to get the output as **y\_pred.**

Notice that the prediction output is an array of real numbers corresponding to the input array.

**Activity 2: Model Evaluation**

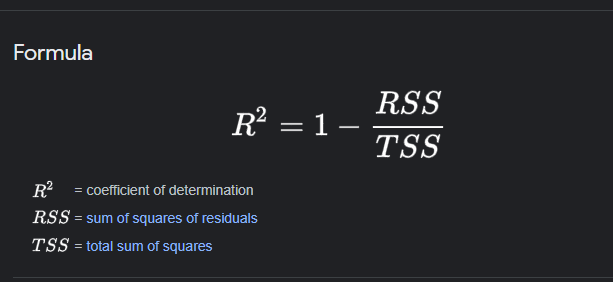
After training the model, the model should be tested by using the test data which is been separated while splitting the data for checking the functionality of the model.

**Regression Evaluation Metrics:**

These model evaluation techniques are used to find out the accuracy of models built in the Regression type of machine learning models. We have three types of evaluation methods.

* R-square\_score
* RMSE – root mean squared error

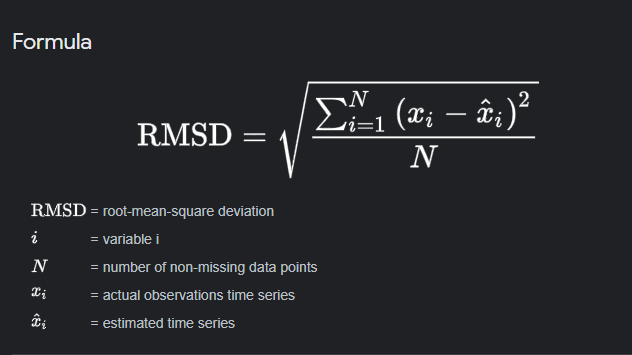
1. R-squared \_score



It is the ratio of several correct predictions to the total number of input samples.

Select the model, which gives the best accuracy of all, generates predictions and find the accuracy with training and testing data

2. RMSE –Root Mean Square Error



RMSE value for Random forest (**Random Forest Regression** is a supervised learning algorithm that uses **ensemble learning** method for regression. The ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model) is very less when compared with other models, so saving the Random forest model and deploying using the following process.

Text

Description automatically generated

Note:- Above one is a final tuned model with an r2 score and RMSE.

**Activity 3: Save the Model**

After building the model we have to save the model.

**Pickle** in **Python** is primarily **used** in serializing and deserializing a **Python** object structure. In other words, it's the process of converting a **Python** object into a byte stream to store it in a file/database, maintain program state across sessions or transport data over the network. wb indicates write method and rd indicates read method.

This is done by the below code

Logo

Description automatically generated with medium confidence

**Final Graphical Representation:-**

Text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

**Milestone 4: Application Building**

In this section, we will build a web application integrated into 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

**Activity 1: Build HTML Code**

* + We will create the front-end part of the web page on this HTML page. On this page, we will accept input from the user and Predict the values.

For more information regarding HTML

[**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)

In our project we have HTML files, they are

1.index.html

**home.html**

Text

Description automatically generated

**The HTML page looks like**

A picture containing text, indoor

Description automatically generated

It will display all the input parameters and the prediction text will display the output value of the data given by the user.

**Activity 2: Main Python Script**

Let us build an app.py flask file which is a web framework written in python for server-side scripting. Let’s see step by step procedure for building the backend application.

To develop web API concerning our model, we use the Flask framework which is written in python.

Line 1-5 We are importing necessary libraries like Flask to host our model request

Line 8 Initialise the Flask application

Line 10 Routes the API URL

Line 17 Rendering the template. This helps to redirect to the home page. On this home page, we give our input and ask the model to predict

Line 21 loading model

Line 25 Predicting the values given by the user

Line 27-30 Rendering to HTLM templates

Line 31 The value of \_\_name\_\_ is set to \_\_main\_\_ when the module run as the main program otherwise it is set to the name of the module

Text

Description automatically generated

**Activity 3: Run the App**

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

Navigate to the localhost where you can view your web page, Then it will run on local host:5000

**Text

Description automatically generated**

**Activity 4:**

* Copy the HTTP link and paste it in the google link tab, it will display the form page
* Enter the values as per the form and click on predict button
* It will redirect to the page based on prediction output