**MobileNudge: Personalized Suggestions for Mobile Users**

**PROJECT DESCRIPTION**

The MobileNudge project aims to develop a sophisticated mobile recommendation system that leverages cutting-edge artificial intelligence and machine learning algorithms to provide personalized suggestions to mobile users. By analyzing user behavior, preferences, and contextual data, MobileNudge delivers relevant and timely recommendations to enhance the user experience across various mobile applications.

**Scenario-1:**

Sarah, a frequent online shopper, uses a mobile shopping app that integrates the MobileNudge recommendation system. Based on her browsing history, past purchases, and search queries, the app offers personalized product suggestions. When Sarah opens the app, she sees a curated list of products she might be interested in, including a notification for a flash sale on items similar to her previous purchases.

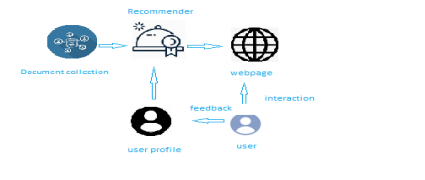
**Scenario-2:**

John is planning a trip to Paris and uses a travel app that incorporates MobileNudge. The app suggests nearby attractions, restaurants, and activities based on his current location, the weather, and his past travel interests. As he explores the city, the app updates in real-time, providing recommendations that fit his preferences and the time of day.

**Scenario-3:**

Emma enjoys reading the news and watching videos on her mobile device. Her news app, powered by MobileNudge, tracks her reading habits and preferences. When she opens the app, it presents her with articles and videos on topics she frequently follows, such as technology and health. The app also suggests new content based on trending topics and her interests.

**ARCHITECTURE**

****

**Learning Outcomes:**

By the end of this project:

● You’ll be able to perform one of the techniques to build you recommendation system

● You’ll be able to know the recommendation system using Content-Based Filtering.

● You will be able to know how to pre-process / clean the data using different data pre-processing techniques.

● You will able to analyse or get insights of data through visualization.

● Applying algorithms according to dataset and based on visualization.

● You will able to know how to find accuracy of the model.

● You will be able to know how to build a web application using Flask framework.

**PRE-REQUISITES:**

To complete the project successfully, you need to install following software & packages:

**Activity 1: Install Anaconda IDE / Anaconda Navigator.**

**●** In order develop a solution to this problem statement, we need an environment to write and test the code.

**●** We use Anaconda IDE (Integrated Developing Environment).

● Refer to the below link to download & install Anaconda Navigator.

**Link:** https://www.youtube.com/watch?v=5mDYijMfSzs

**1. Activity 2: To build Machine learning models you must require the following packages**

● **Numpy**:

It is an open-source numerical Python library. It contains a multidimensional array and matrix data structures and can be used to perform mathematical operations

● **Numpy**:

It is a free machine learning library for Python. It features various algorithms like

support vector machine, random forests, and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy

● **Matplotlib and Seaborn**

Matplotlib is mainly deployed for basic plotting. Visualization using matplotlib generally consists of bars, pies, lines, scatter plots and so on. Seaborn: Seaborn, on the other hand, provides a variety of visualization patterns. It uses fewer syntax and has easily interesting default themes.

**● Flask:**

Web framework used for building Web applications.

If you are using **anaconda navigator**, follow below steps to download required packages:

● Open anaconda prompt.

● Type “pip install pandas” and click enter.

● Type “pip install matplotlib” and click enter.

● Type “pip install seaborn” and click enter.

● Type “pip install plotly” and click enter.

● Type “pip install numpy” and click enter.

● Type “pip install scikit-image” and click enter.

● Type “pip install scikit-learn” and click enter.

● Type “pip install Flask” and click enter.

**Link:** Introduction to Scikit-Learn (sklearn) in Python • datagy

**PRIOR KNOWLEDGE**

One should have knowledge on the following Concepts:

**Link:** Supervised and Unsupervised Learning

Watch the below video to know about the types of machine learning **Link:** Regression, Classification and Clustering

**Link:** ML - Content Based Recommender System - GeeksforGeeks **Link:** NLTK :: Natural Language Toolkit

**Link:** Flask:

**Link:** Recommendation System

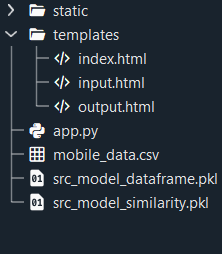
It is recommended to watch above video’s to understand the concepts before you start your project.

**PROJECT WORK FLOW**

● User interacts with the UI (User Interface) to enter the input features.

● Entered features/input is analysed by the model which is integrated

● Once model analyses the entered inputs, the prediction is showcased on the UI.



**TASKS**

1. Data Collection.

● Collect the dataset or Create the dataset

2. Data Pre- processing.

Import the Libraries.

Importing the dataset.

Exploratory Data Analysis

Data Visualization.

3. Content Based Filtering

Merging datasets

Creating the recommender system

Predicting the results

4. Application Building

Create an HTML file

Build a Python Code

**Milestone 1: Data Collection**

we will use the mobile phones data for our analysis to draw conclusions using the content filtering method.

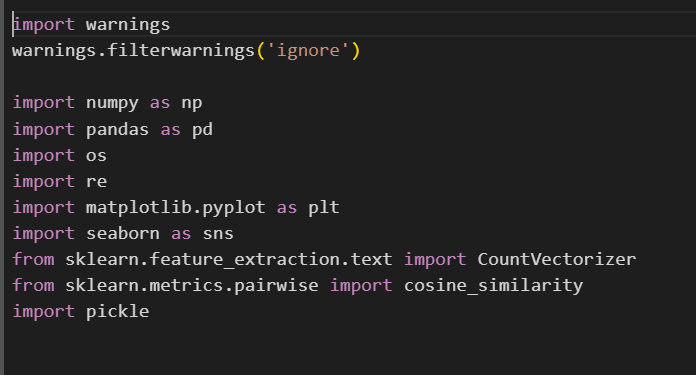
Here is the dataset link:[**LINK**](https://www.kaggle.com/datasets/gyanprakashkushwaha/mobile-recommendation-system-dataset)

**Milestone 2: Data Pre-processing**

In this milestone, you need to complete all the below activities to build the model.

**Activity 1: Import Libraries**

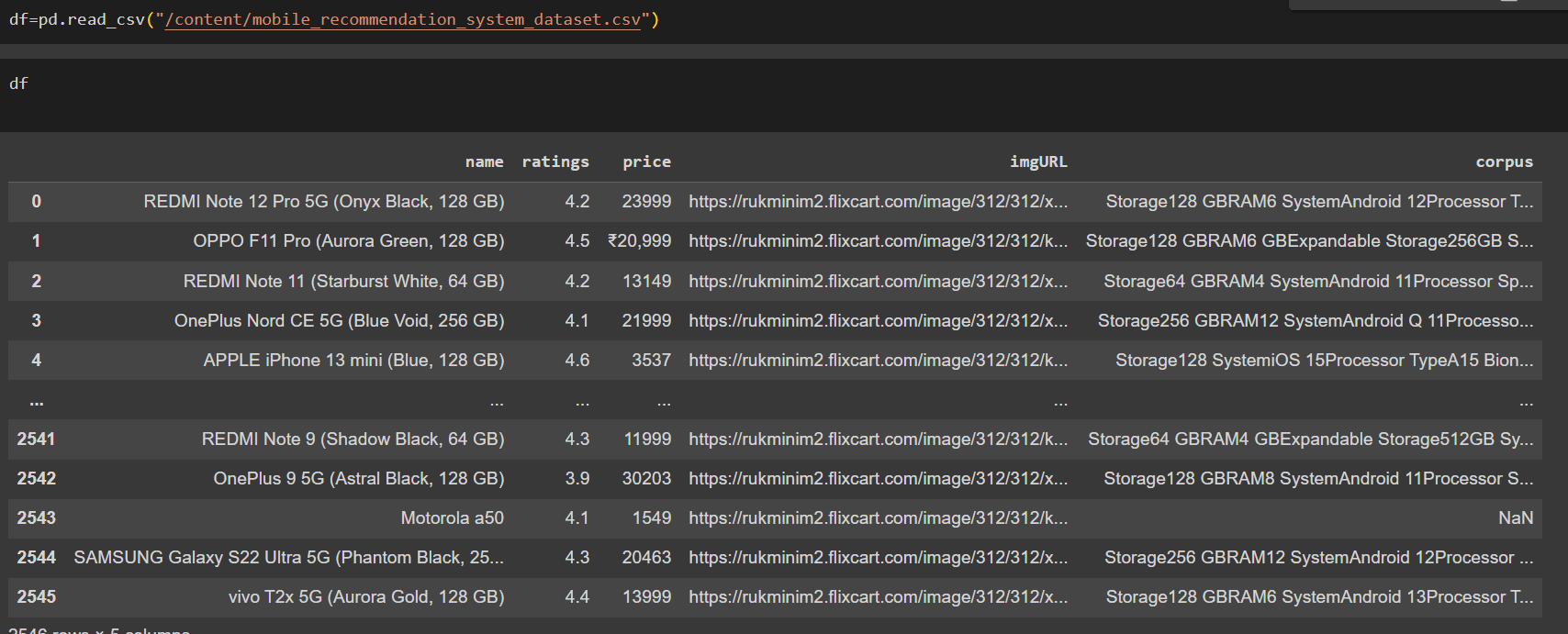
Import the below essential libraries for data pre-processing and creating recommendation system. Pandas and NumPy are used for data pre-processing and cleaning. Seaborn, Plotly and Matplotlib helped in creating visual graphics and bar plots for the dataset. Also, since there would be cleaning of text data (reviews) as well, therefore for that we will use nltk and sklearn library.



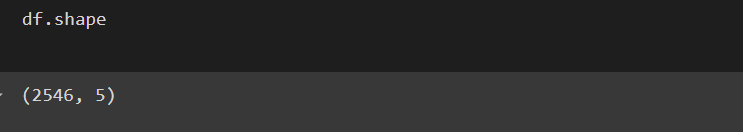
**Activity 2: Read the Dataset:**

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 csv file.

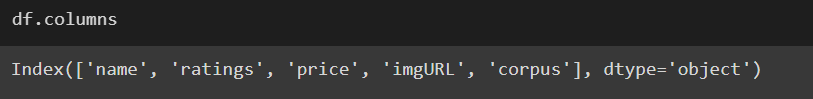
****

**Activity 3: Analyse the Dataset:**

****

The dataset contains 2546 records with 5 features.

**Checking the columns in the dataset.**

****

**Columns description**

**name:** The name or model of the mobile device.

**ratings**: The user rating of the mobile device, often based on reviews from customers.

**price**: The cost of the mobile device, usually in a specific currency.

**imgURL**: The URL link to an image of the mobile device.

**corpus**: A textual description or collection of information about the mobile device. This could include specifications, features, and other relevant details.

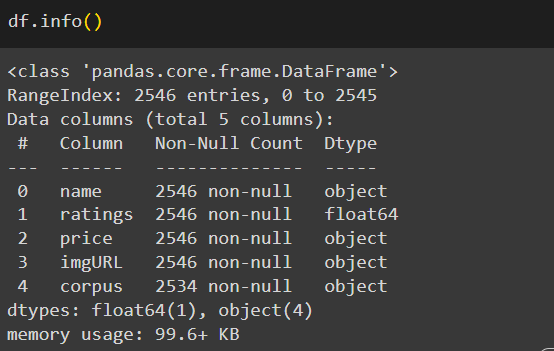
**Understanding Overview of features**

● How the information is stored in a DataFrame or Python object affects what we can do with it and the outputs of calculations as well. There are two main types of data those are numeric and text data types.

● Numeric data types include integers and floats.

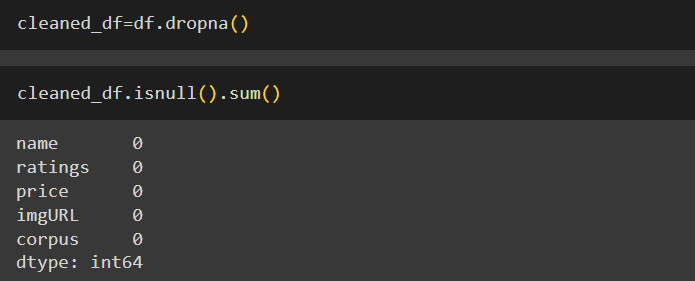
● Text data type is known as Strings in Python, or Objects in Pandas. Strings can contain numbers and / or characters.

● For example, a string might be a word, a sentence, or several sentences. ● Will see how our dataset is, by using **info ()** method.



● As you can see in our dataset, except ‘votes’, all other features are categorical data, but it is not necessary that all the continuous data which we are seeing has to be continuous in nature. There may be a case that some categorical data is in the form of numbers but when we perform info () operation we will get numerical output. So, we need to take care of those type of data also.

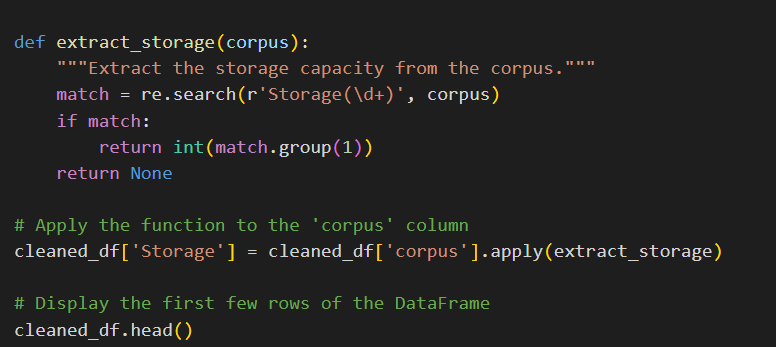
**Checking for null values in the dataset**

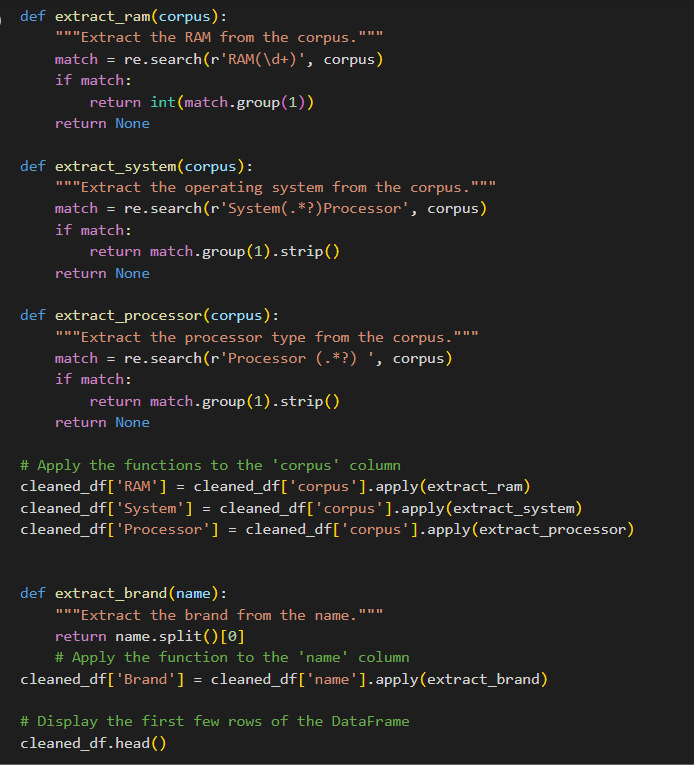
****

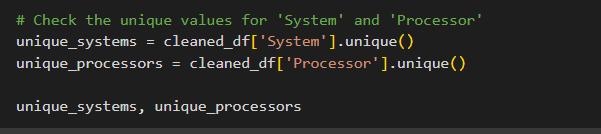
**Feature Engineering:**

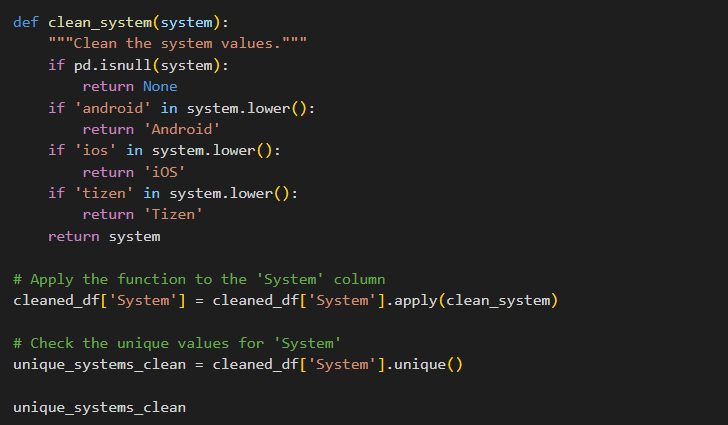
We are adding the new columns to the data to check the distribution of the data.so we are

Going to extract the column corpus into system, storage, processor and system.





****

****

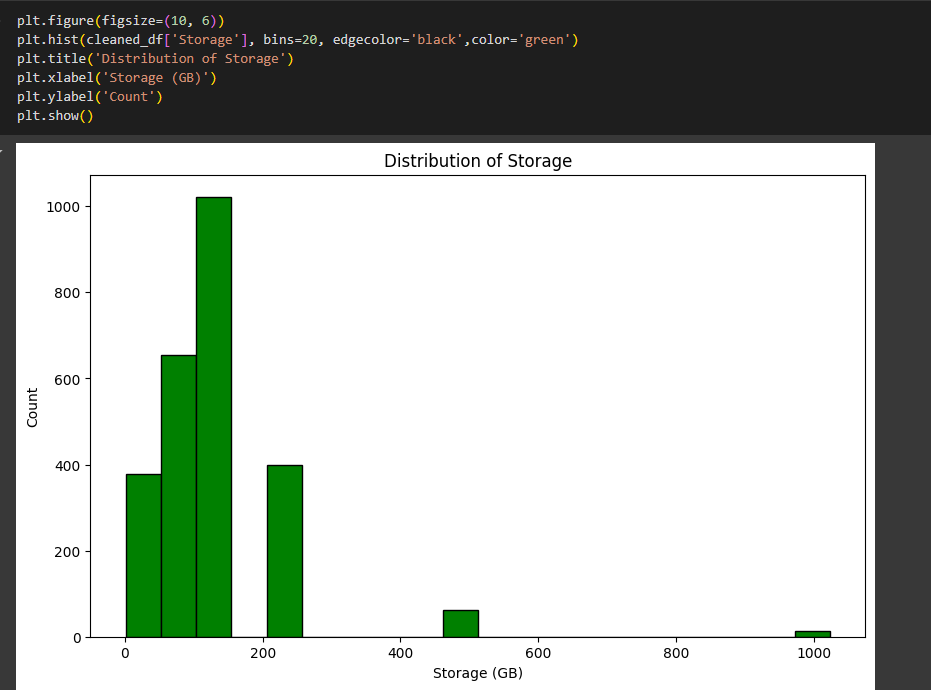
**Milestone 3: Data Visualization**

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. In fact, even the most sophisticated machine learning models will perform poorly on data that wasn’t visualized and understood properly.

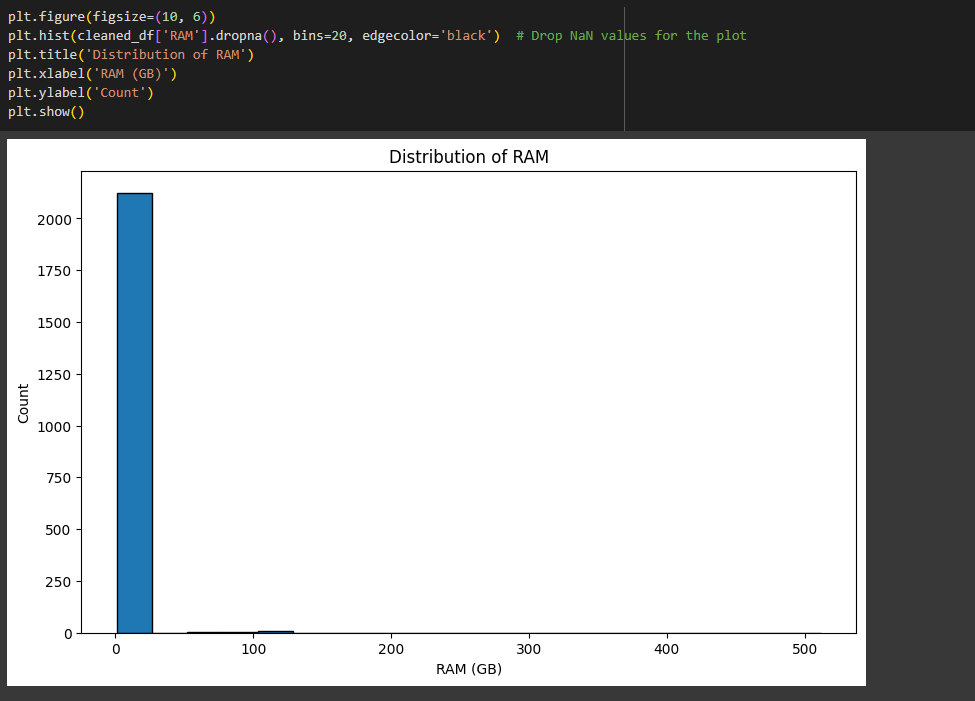
To visualize the dataset, we need libraries called Matplotlib, Seaborn. The Matplotlib library is a Python 2D plotting library which allows you to generate plots, scatter plots, histograms, bar charts etc.

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

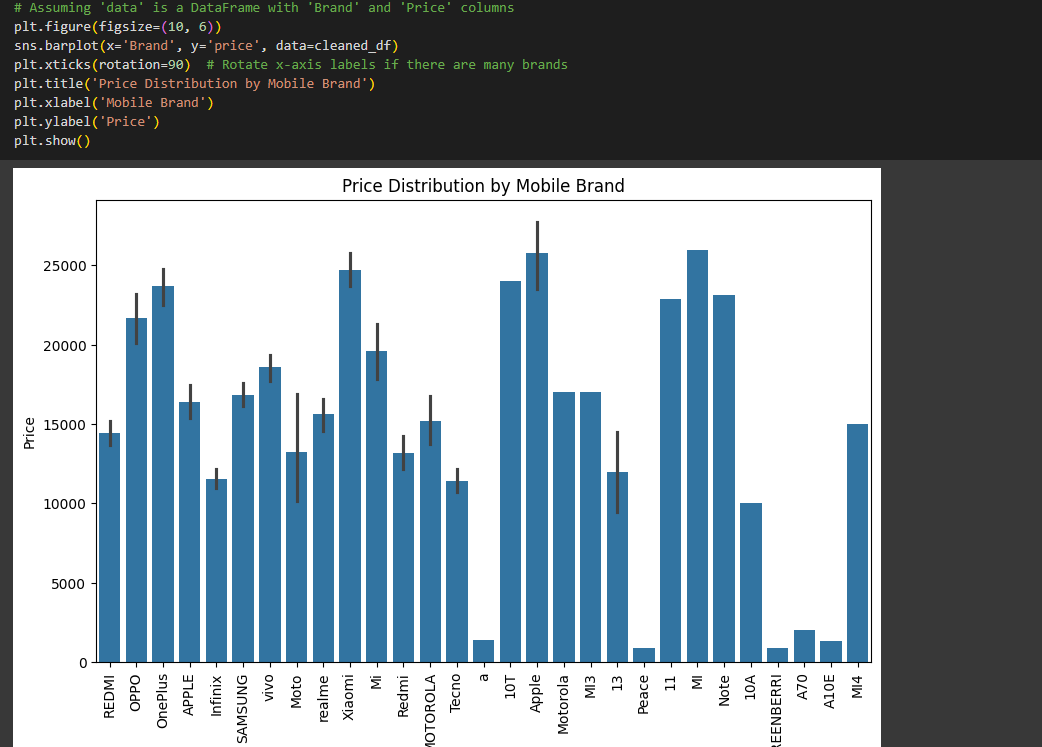
At first, we will be plotting a hist plot using matplotlib for showing the storage rate.



Checking the distribution of ram, for that we are using histplot from matplotlib library.



Using the barplot seeing the brand wise prices for the mobile phones.

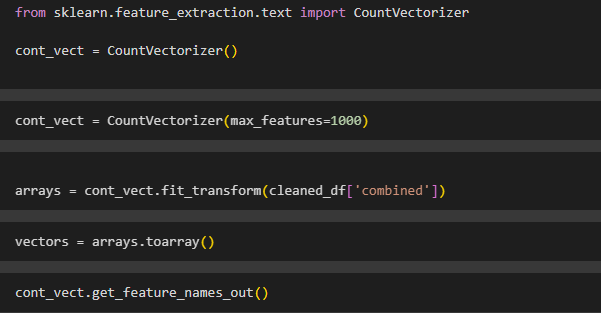


**Milestone 4: CONTENT-BASE RECOMMENDER SYSTEM**

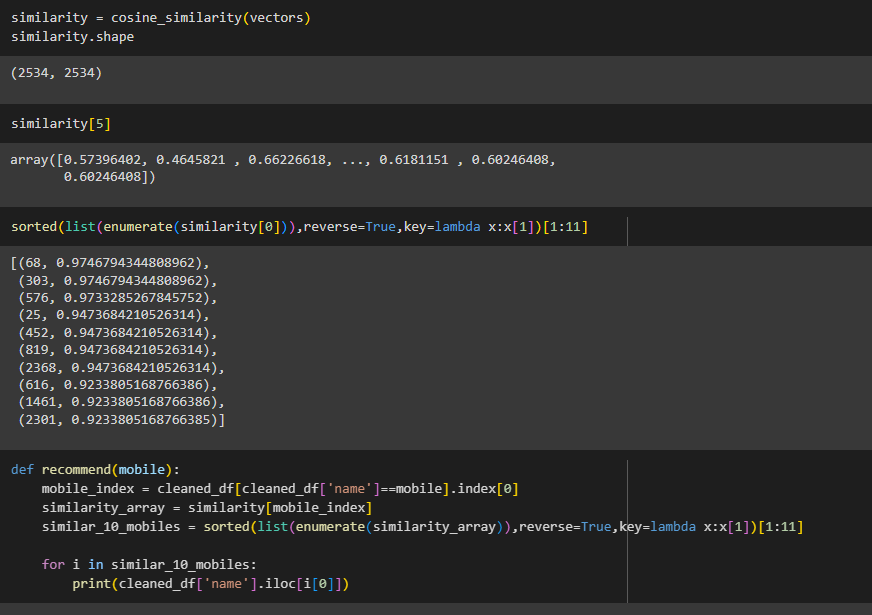
**Activity:1 Count Vectorization**

Count vectorization is a statistical method of assessing the meaning of a word in a given document. Now we use Count vectorization on the dataset.

Cosine similarity is a metric used to determine how similar the documents are irrespective of their size. Calculating the cosine similarity of each item with every other item in the dataset.

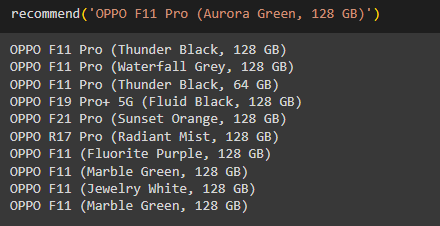


**Calculating Cosine Similarity**

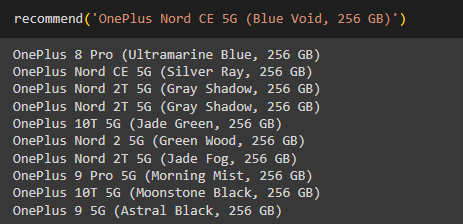
****

**Querying recommendation for 3 users:**

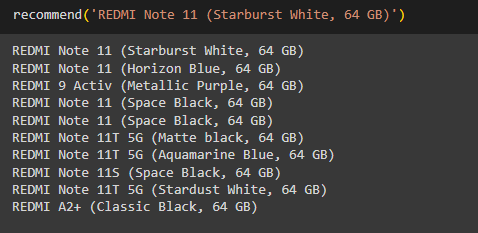
**Recommend for Oppo :**

****

**Recommend for oneplus :**

****

**Recommend for Redmi:**

****

**Milestone 5: Application Building**

**Activity 1: Create an HTML File**

We use HTML to create the front end part of the web page.

Here, we created 2 html pages- index.html, web.html.

index.html displays home page.

web.html accepts the values from the input and displays the prediction. For more information regarding HTML refer the link below

**https://www.w3schools.com/bootstrap/bootstrap\_forms\_inputs.asp**

● We also use JavaScript-main.js and CSS-main.css to enhance our functionality and view of HTML pages.

**○ Link :https://www.w3schools.com/css/**

○ **https://www.w3schools.com/js/DEFAULT.asp**

**Activity 2: Build python code**

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

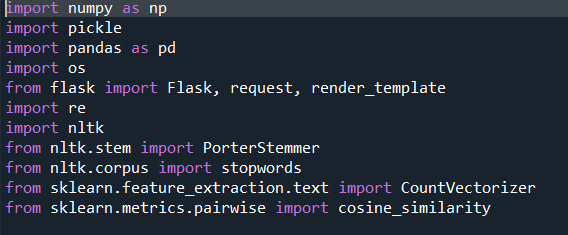
**●** App starts running when “\_\_name\_\_” constructor is called in main.

**●** render template is used to return html file.

**●** “GET” method is used to take input from the user.

**●** “POST” method is used to display the output to the user.

**Importing libraries**

****

Libraries required for the app to run are to be imported.

**Creating our flask app and loading the newly created dataset**

Now after all the libraries are import we will be creating our flask app with the updated dataset

**Routing to the html Page:**

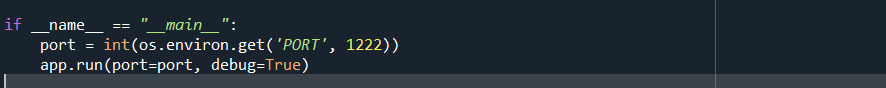
Basically, we give routes of our html pages in order to show case the UI. By giving the routes the built code in the html page is connected to our flask app. This is how a UI can be built and showcased.



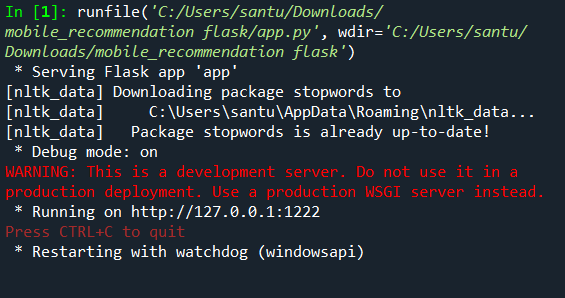
We are routing the app to the html templates which we want to render.

Firstly, we are rendering the home.html template and from there we are navigating to our prediction page that is indexnew.html





Lastly, we run our app on the local host.



**Activity:3 Run The app in local browser**

● Open anaconda prompt from the start menu

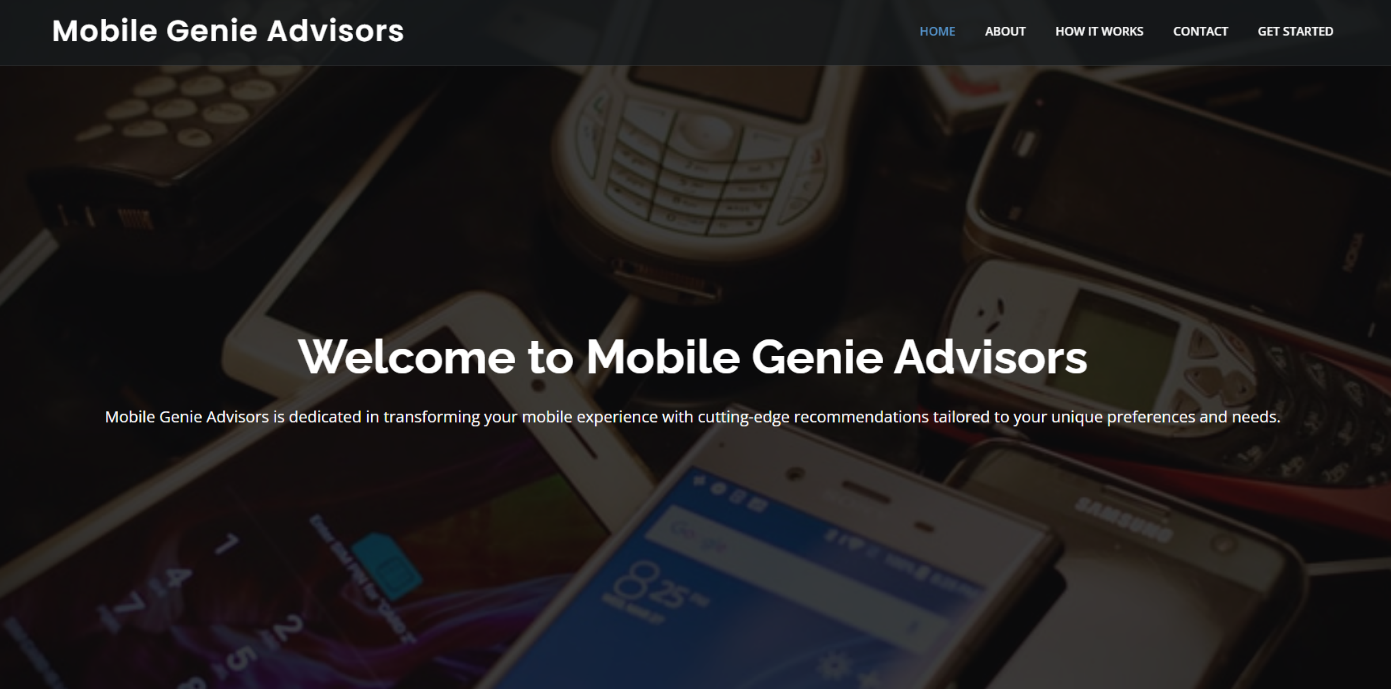
● Navigate to the folder where your python script is.

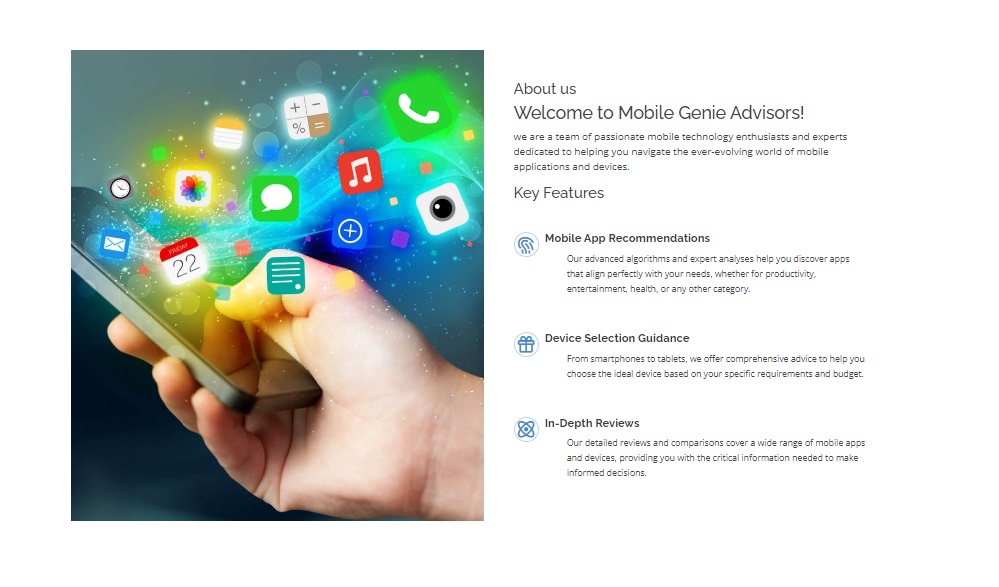
● Now type “python app.py” command

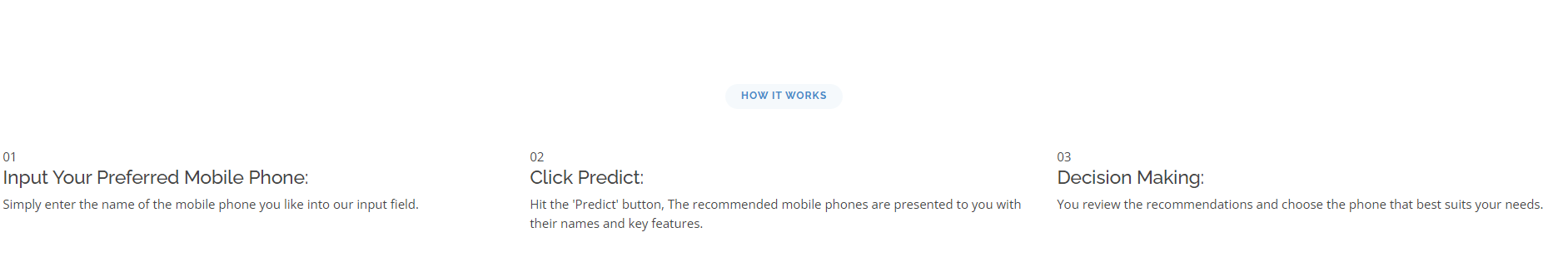
● Navigate to the localhost where you can view your web page

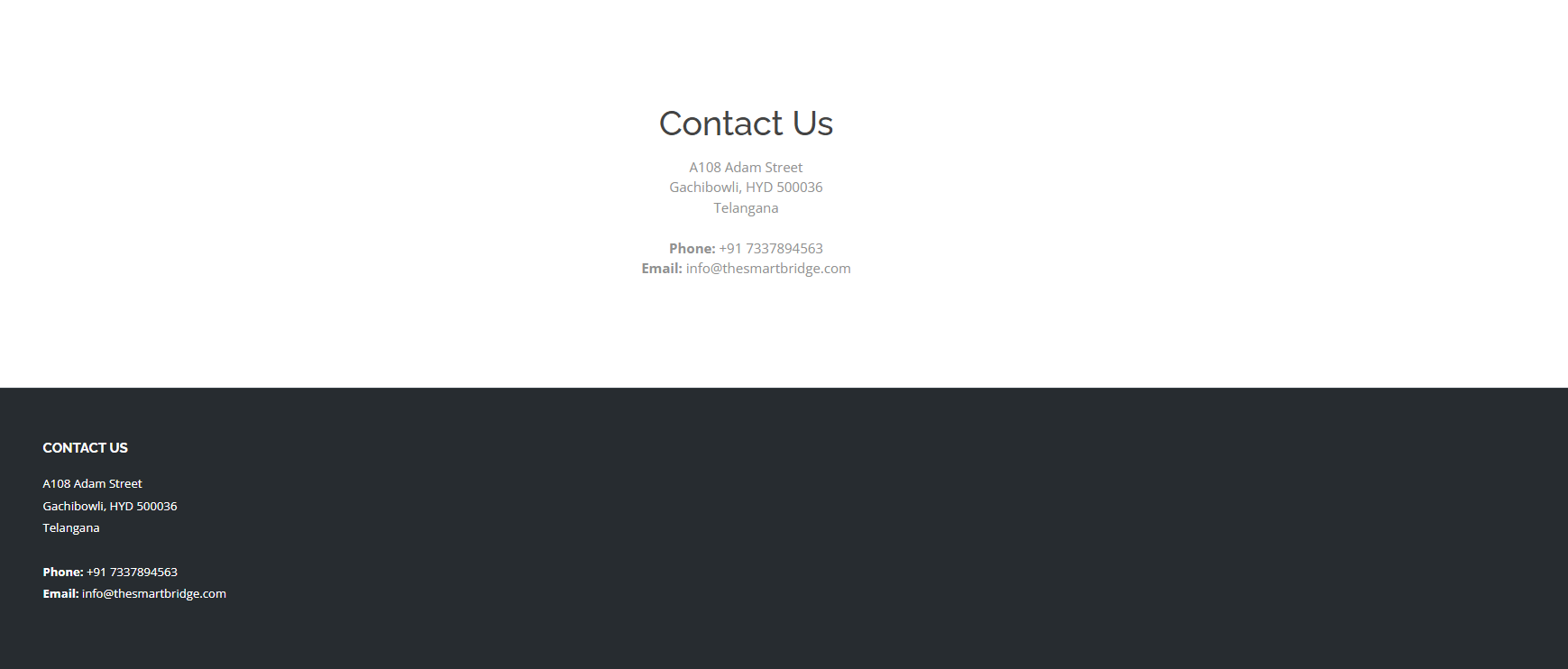
**Showcasing The UI**

This is home main page that describes the project summarizes it.



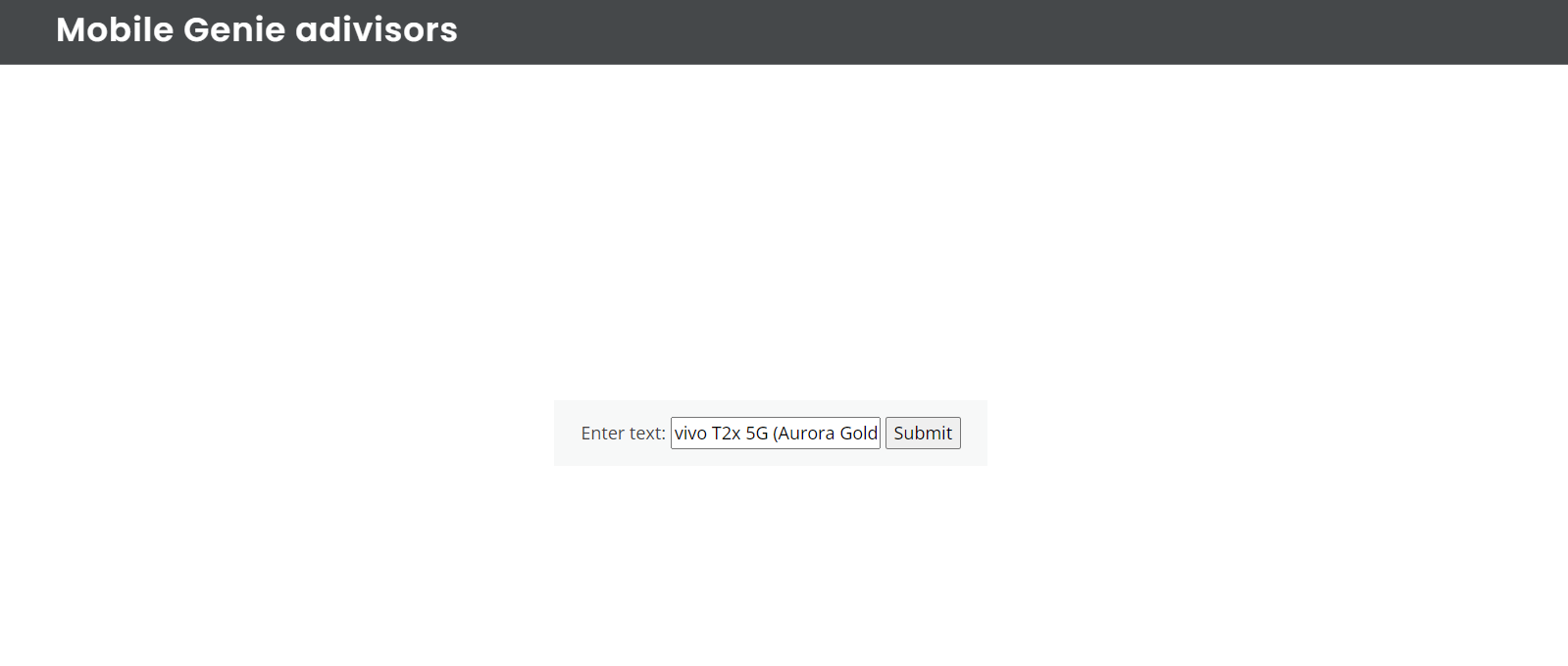
****

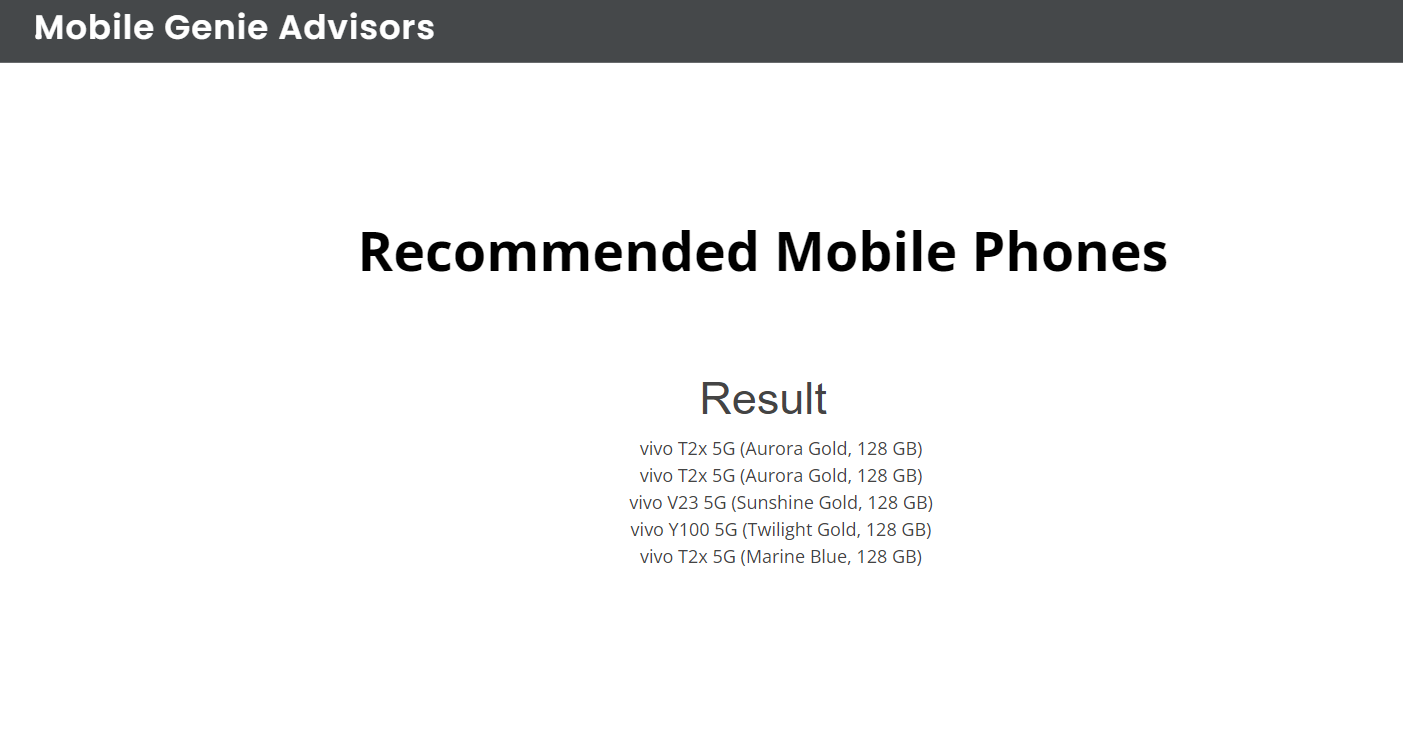
****

****

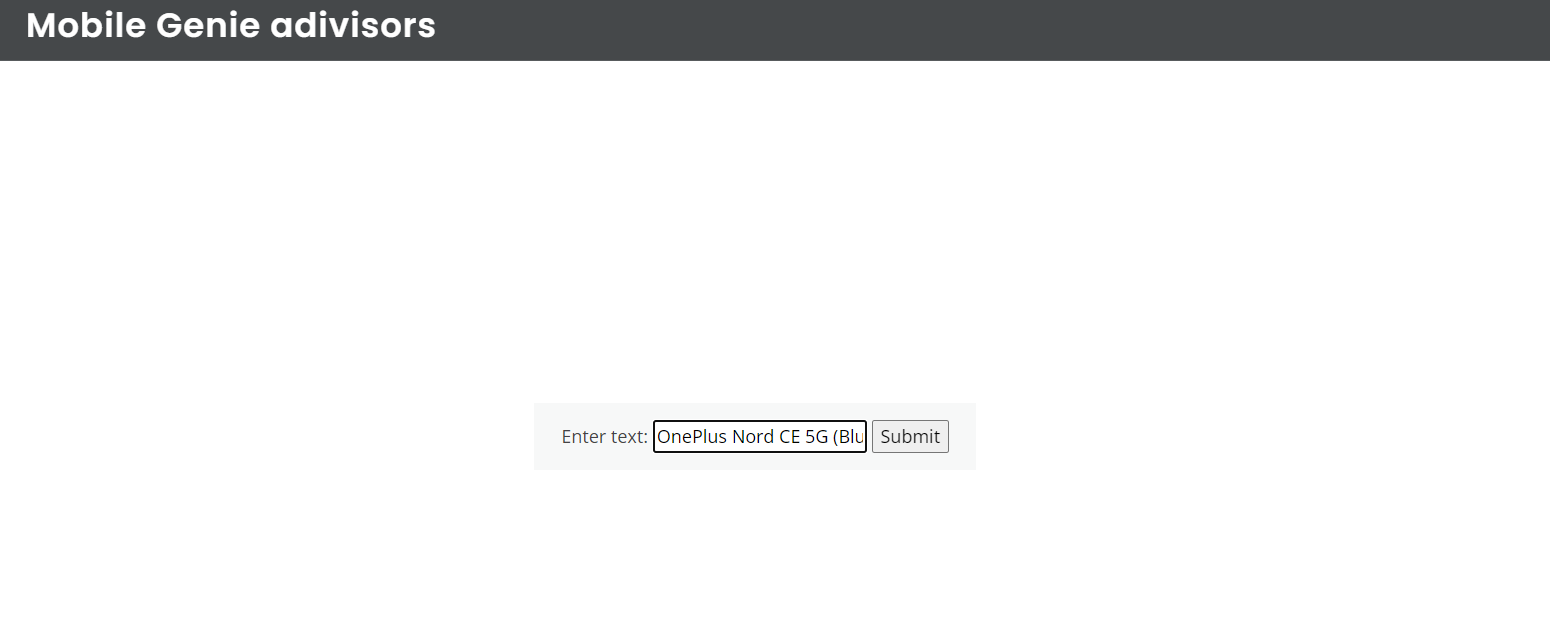
**User input for the mobile:**

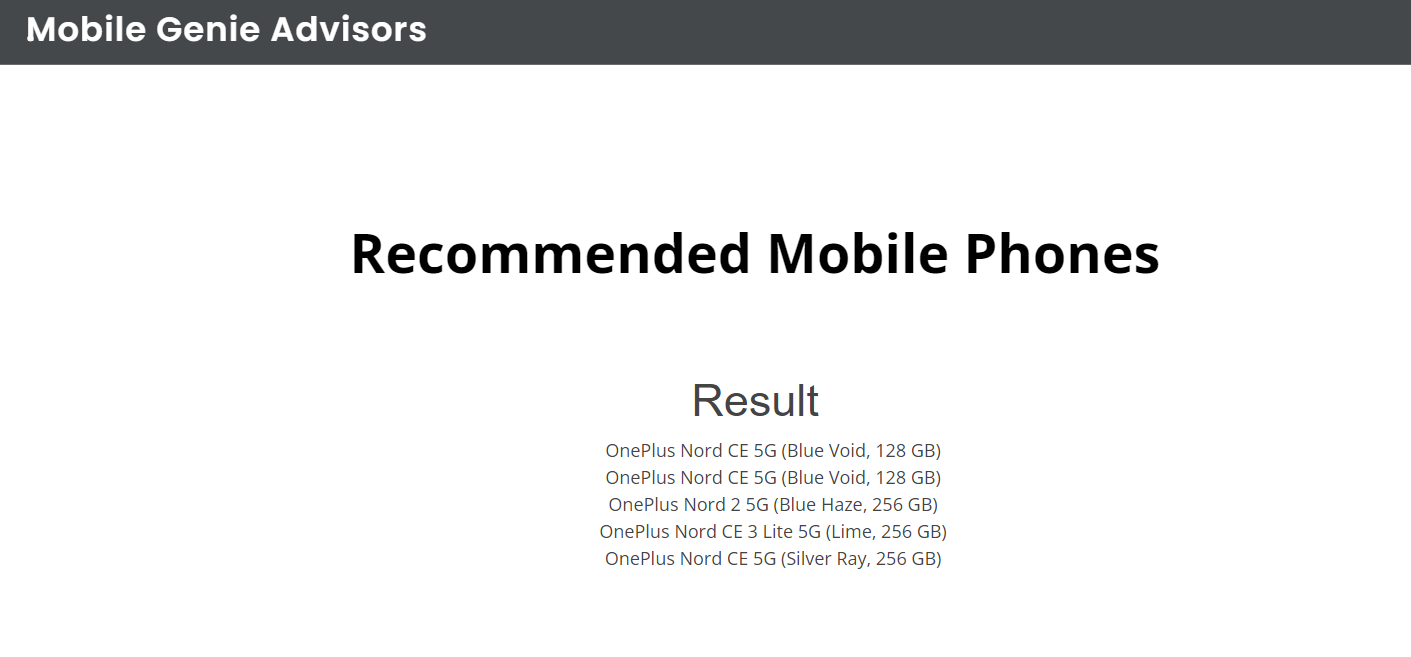
**Test case-1:**

****

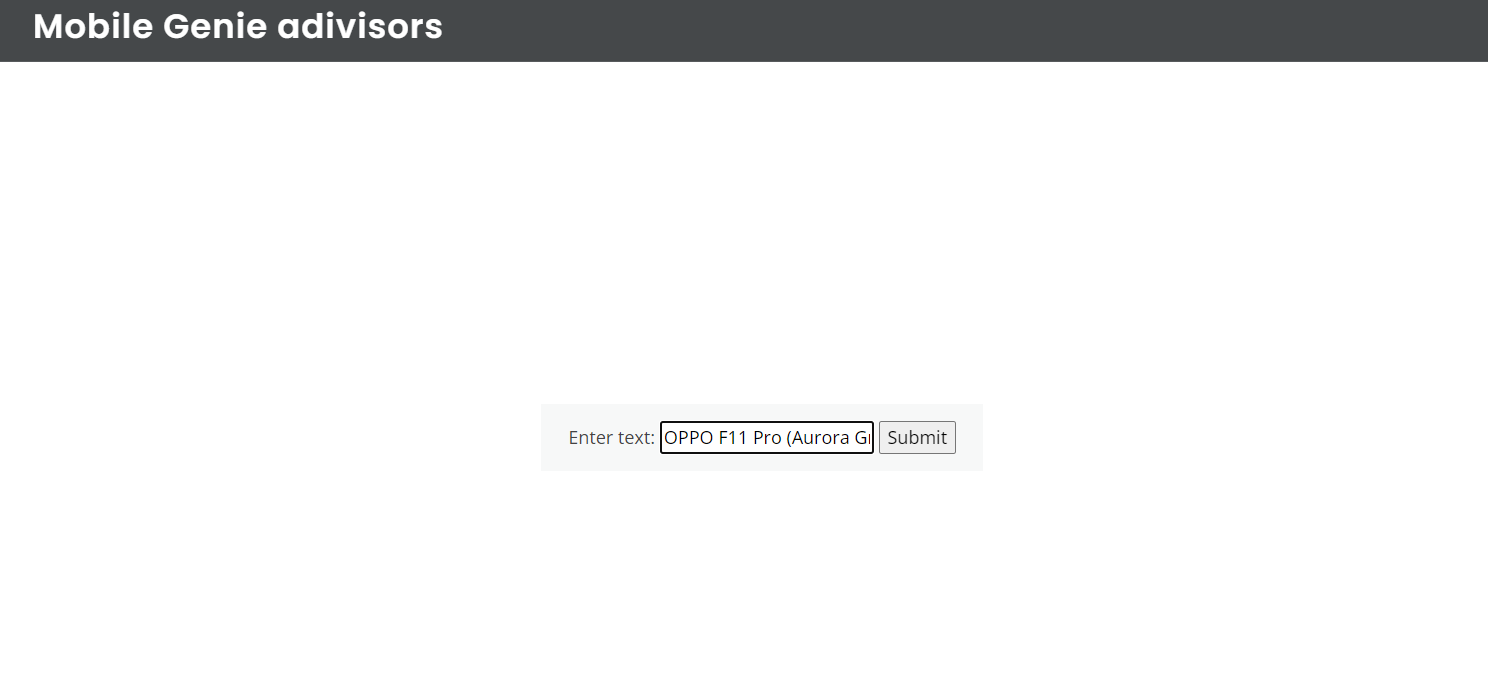
****

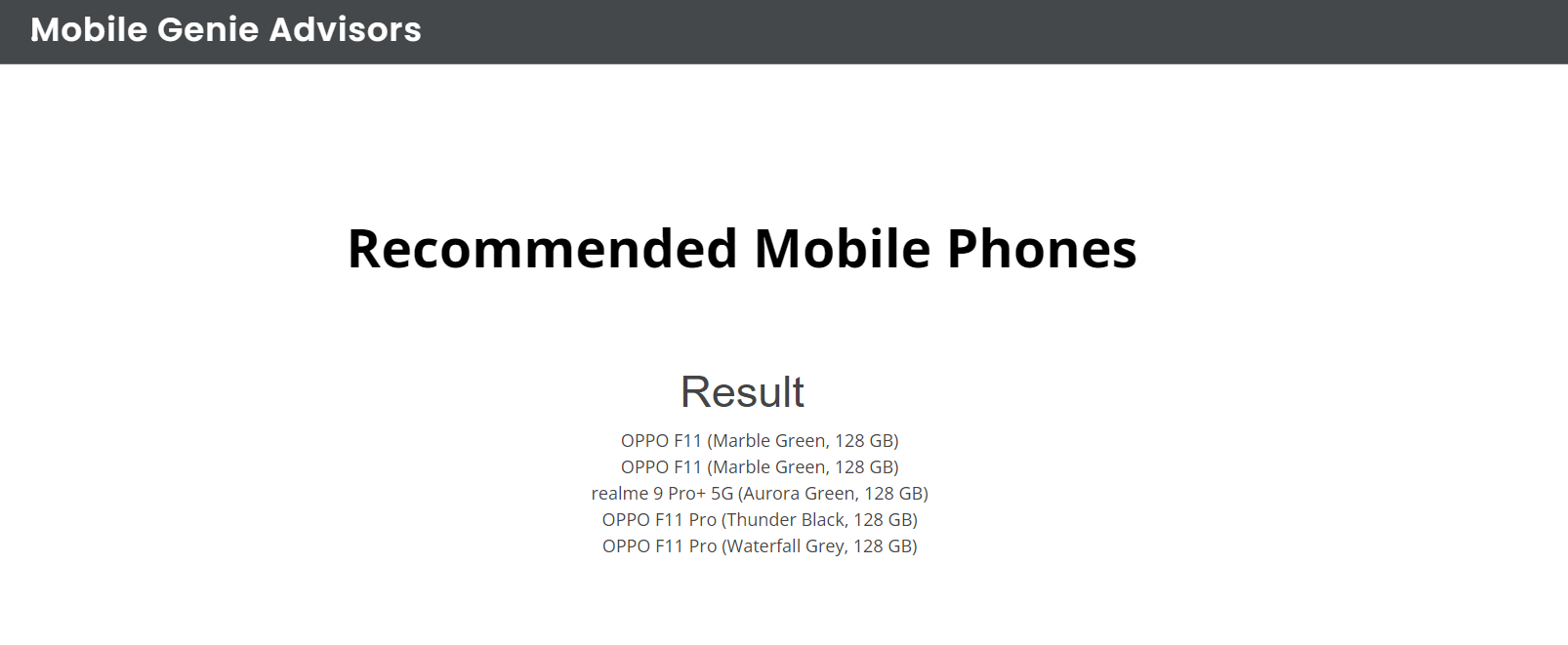
**Test case-2**

****

****

**Test case-3:**

****

****

Finally, the prediction for the given mobile phone inputs is shown.