CUSTOMER SEGMENTATION USING MACHINE LEARNING

Mini Project documentation submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

Submitted By

DIVYA SRI KUSURI	**(21UK1A6799)**
LIKHITHA RAGAM	**(21UK1A6783)**
SAIRAM SANKOJI	**(21UK1A6785)**
PURNACHAND KONGALA	**(22UK5A6709)**

Under the guidance of

K. SOUMYA

Asst. Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

VAAGDEVI ENGINEERING COLLEGE WARANGAL

CERTIFICATE

This is to certify that the Mini project report entitled "**CUSTOMER SEGMENTATION USING MACHINE LEARNING**" is being submitted by **DIVYA SRI KUSURI (21UK1A6799), LIKHITHA RAGAM(21UK1A6783), SAIRAM SANKOJI(21UK1A6785), PURNACHAND KONGALA(21UK5A6702)**, partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024-2025.

Project Guide	**Head of the Department**
K. SOUMYA	**Dr. K. SHARMILAREDDY**
(Asst. professor)	**(Professor)**

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. Prasad Rao**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this mini project in the institute.

We extend our heartfelt thanks to **Dr. K. SHARMILAREDDY,** Head of the Department of CSD, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the mini project.

We express heartfelt thanks to the guide **K. SOUMYA**, Asst. Professor, Department of CSD for her constant support and giving necessary guidance for completion of this mini project.

Finally, we express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experiencing throughout thesis.

DIVYA SRI KUSURI **(21UK1A6799)**

LIKHITHA RAGAM **(21UK1A6783)**

SAIRAM SANKOJI **(21UK1A6785)**

PURNACHAND KONGALA **(22UK5A6709)**

ABSTRACT

Nowadays Customer segmentation became very popular method for dividing company's customers for retaining customers and making profit out of them, in the following study customers of different of organizations are classified on the basis of their behavioural characteristics such as spending and income; by taking behavioural aspects into consideration makes these methods an efficient one as compares to others.

For this classification a machine algorithm named as k-means clustering algorithm is used and based on the behavioural characteristic's customers are classified. Formed clusters help the company to target individual customer and advertise the content to them through marketing campaign and social media sites which they are really interested in.

This enables the learner to understand the business use case of how and why to segment the customers and we'll able to understand the unsupervised learning methods such as H-clustering and k-means clustering.

We can Apply different algorithms according to a dataset and based on visualization.

CUSTOMER SEGMENTATION USING MACHINE LEARNING

1.INTRODUCTION

1.1 OVERVIEW

Customer segmentation is the process of dividing a customer base into groups of individuals that are similar in certain ways relevant to marketing, such as age, gender, interests, and spending habits. It enables companies to target specific groups with tailored promotions, products, or services that are most likely to resonate with them. Machine learning has become a popular tool for automating the process of customer segmentation, providing a more efficient and effective way to identify patterns and relationships within customer data.

- This project aims to analyze the behavior of customers based on the similarities and identify opportunities for growth. The data set consists of sex, marital status, education, occupation, income and settlement size.
- It's a way for organizations to understand their customers. Knowing the differences between customer groups, it's easier to make strategic decisions regarding product growth and marketing.
- The outcomes of this project are endless and depend mainly on how much customer data you have at your use.

1.2 PURPOSE

The purpose of this project is to help businesses identify new opportunities for growth. By analysing customer data, businesses can identify gaps in the markets or areas where there is untapped potential. This can lead to the development of new products or services that better meet the needs of specific customer segments.

2. LITERATURE SURVEY

2.1. EXISTING PROBLEM

The poor customer segmentation can significantly impact sales performances. Ineffective segmentation variables and tools can lead to negative financial consequences. Proper market segmentation is crucial for meeting diverse consumer demands and improving sales volume.

- Poor quality or incomplete data can lead to inaccurate segmentation.

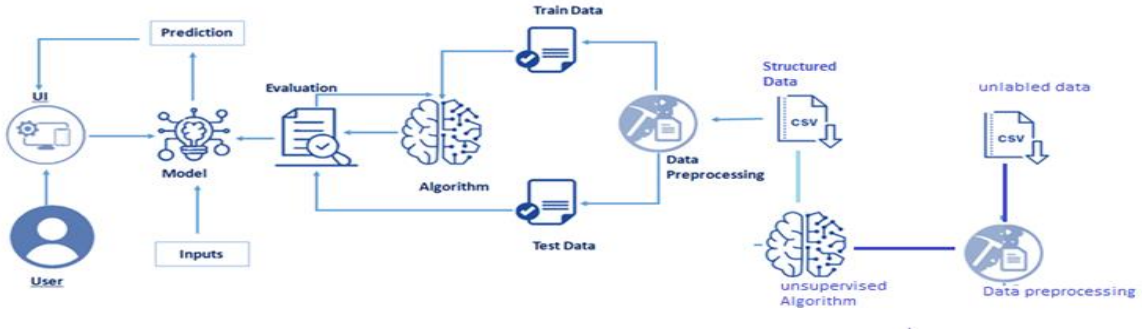
- Choosing the wrong features or not enough relevant features can result in ineffective segmentation.
- The choice of clustering algorithm significantly affects the segmentation results, and the wrong choice can lead to suboptimal segments.
- Target their marketing campaigns more effectively Develop products and services that meet theneeds of their customers Increase customer satisfaction and retention

2.2 PROPOSED SOLUTION

- The Customer Segmentation project aims to analyze the spending behavior of customers and identify opportunities for growth.
- Using unsupervised machine learning techniques, specifically clustering algorithms, the project seeks to group customers with similar spending patterns together. By identifying customer segments with distinct spending behavior's, the project aims to provide insights on how businesses can tailor their marketing strategies and product offerings to better serve each customer segment.
- Overall, the Customer Segmentation project seeks to provide valuable insights for wholesale businesses on how to optimize their operations and increase customer satisfaction and retention.

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2 HARDWARE / SOFTWARE DESIGNING

To Complete this project, you must require the following software's, concepts and packages:

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 scikit-learn**" and click enter.
- Type "**pip install Flask**" and click enter.

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

1.Programming language: python is widely used for machine learning projects due to its extensive libraries and community support.

2.Development environment: Jupyter notebook is an open-source web application for creating and sharing documents that contains live code, equations, visualizations, and narrative text.

3.Data collection and management: The dataset in CSV format is essential for training and testing your predictive model. It should include historical air quality data, weather information, pollutant levels, and other relevant features.

4.Data processing tools: Python libraries like NumPy, Pandas, and Scikit-learn will be used to preprocess the dataset. This includes handling missing data, feature scaling, and data cleaning.

- Pandas: For data manipulation and analysis.

- NumPy: For numerical computing and array operations.

5.Data visualisation:

- Matplotlib: Basic plotting library in Python.

- Seaborn: Statistical data visualization built on top of Matplotlib.

6.Feature selection: Feature selection or dropping unnecessary features from the dataset can be done using Scikit-learn or custom Python code to enhance the model's efficiency.

7.Machine learning libraries:

- scikit-learn: Comprehensive library for traditional machine learning algorithms, used for scaling features.

- TensorFlow: Open-source library for deep learning.

8.Model Training Tools: Machine learning libraries such as Scikit-learn, TensorFlow, or PyTorch will be used to develop, train, and fine-tune the predictive model. Regression or classification models can be considered, depending on the nature of the AQI prediction task.

9.Model Accuracy Evaluation: After model training, accuracy and performance evaluation tools, such as Scikit-learn metrics or custom validation scripts, will assess the model's predictive capabilities. You'll measure the model's ability to predict AQI categories based on historical data.

10.UI Based on Flask Environment: Flask, a Python web framework, will be used to develop the user interface (UI) for the system. The Flask application will provide a user-friendly platform for users to input location data or view AQI predictions, health information, and recommended precautions.

By using these tools and frameworks, you can efficiently build, evaluate, and deploy a customer segmentation project using machine learning. These software solutions provide a comprehensive ecosystem to handle the entire machine learning pipeline, from data collection to deployment and monitoring.

4.EXPERIMENTAL INVESTIGATIONS

Customer Segmentation using Machine Learning is a strategic approach to dividing a customer base into distinct groups based on shared characteristics, behaviours, and preferences. By leveraging machine learning algorithms and customer data, this project aims to uncover meaningful insights and create targeted marketing strategies, personalized offerings, and improved customer experience.

By the end of this project:

- This project enables the learner to understand the business use case of how and why to segment the customers.
- You'll able to understand the unsupervised learning methods such as H-clustering and k-means clustering.
- 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 analyse or get insights into data through visualization.
- Applying different algorithms according to a 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 frameworks.

Project flow:

· User interacts with the UI (User Interface) to enter the input values

· Entered input values are analysed 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.

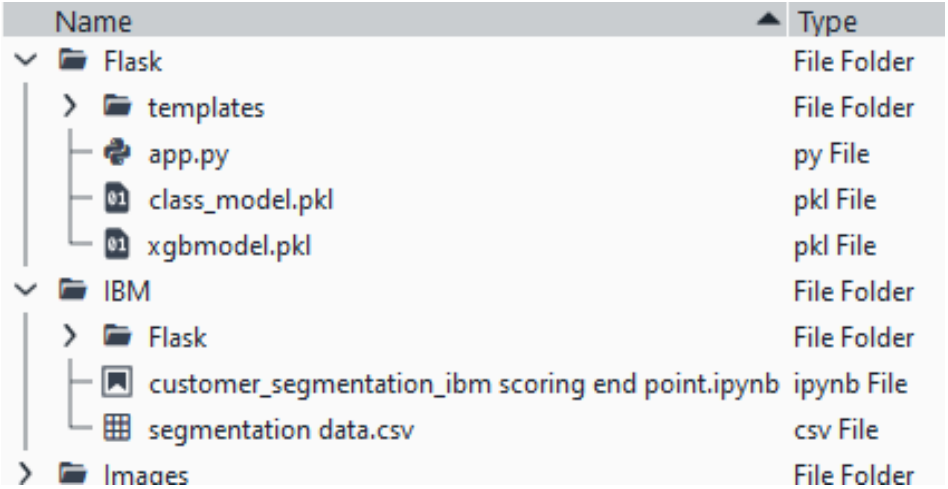
  - Collect the dataset or create the dataset

· Data Pre-processing.

  - Import the Libraries.
  - Importing the dataset.
  - Checking for Null Values.
  - Data Visualization.
  - Taking care of Missing Data.
  - Feature Scaling.

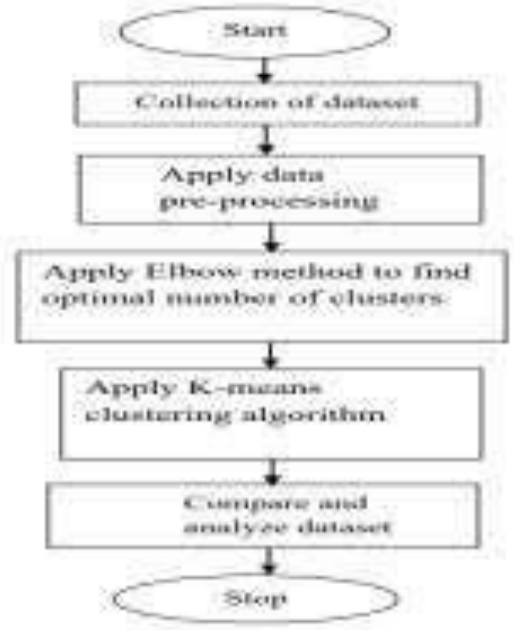
- Unsupervised Model Building
    - Import the model building Libraries
    - Initializing the model
    - Fit and predict the clusters
    - Add the classes to the main data set and save the dataset
    - Splitting x and y
    - Splitting train and test data
- Supervised Model Building
    - Import the model building Libraries
    - Initializing the model
    - Model Training
    - Evaluating the Model
    - Save the Model
- Application Building
    - Create an HTML file
    - Build a Python Code

Project Structure:



- We are building a flask application which needs HTML pages stored in the templatesfolder and a python script app.py for scripting.
- Model.pkl is our saved model. Further we will use this model for flask integration.

5. FLOW CHART



6. RESULT

Data Collection:

ML depends heavily on data, without data, it is impossible for an “AI” to 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.

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

ID	Sex	Marital status	Age	Education	Income	Occupation	Settlement size
100000001	0	0	67	2	124670	1	2
100000002	1	1	22	1	150773	1	2
100000003	0	0	49	1	89210	0	0
100000004	0	0	45	1	171565	1	1
100000005	0	0	53	1	149031	1	1
100000006	0	0	35	1	144848	0	0
100000007	0	0	53	1	156495	1	1
100000008	0	0	35	1	193621	2	1
100000009	0	1	61	2	151591	0	0
100000010	0	1	28	1	174646	2	0
100000011	1	1	25	1	108469	1	0
100000012	1	1	24	1	127596	1	0
100000013	1	1	22	1	108687	1	2
100000014	0	0	60	2	89374	0	0
100000015	1	1	28	1	102899	1	1
100000016	1	1	32	1	88428	0	0
100000017	0	0	53	1	125550	1	0
100000018	0	0	25	0	157434	1	2
100000019	1	1	44	2	261952	2	2
100000020	0	0	31	0	144657	1	1
100000021	0	0	48	1	118777	1	1
100000022	0	0	44	1	147511	1	1
100000023	0	0	48	1	89804	0	0

segmentation data - Google Sheets.

https://docs.google.com/spreadsheets/d/1NnUMX3sjJgRRerkJTAXemlfdyo2GiUhgE_m4w-fAhvs/edit#gid=1219451115

Data Pre-Processing

Data Pre-processing includes the following main tasks

- Import the Libraries.
- Importing the dataset
- Checking for Null Values.
- Data Visualization.
- Feature Scaling.

Visualizing and analyzing the data

- As the dataset is downloaded. Let us read and understand the data properly with the help of somevisualization techniques and some analysing techniques.
- **Note:** There is n 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: Importing the 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
- **SciPy** – which contains all the modules required for scientific and computing functions.

```
import os
import pandas as pd
import seaborn as sns
import sklearn
import scipy
import matplotlib
```

Activity2: Read the Dataset

- You might have your data in .csv files, excel files
- Let's load a .csv data file into pandas using reads () 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")

```
os.chdir('G:\AI&ML\ML projects\cluster analysis')

# Reading the dataset
data = pd.read_csv('segmentation data.csv',header='infer')
```

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 segmentation.csv contains the following Columns

- ID - Unique id of the customer

- Sex – Gender of the customer
- Marital status – whether the person is married or not
- Age = Age of the person
- Education – Education of the person
- Income – income of the person
- Occupation – indicates the profession of a person, employed or unemployed or business
- Settlement size –Represents the no. of persons in a family

Analyzing the data

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

	ID	Sex	Marital status	Age	Education	Income	Occupation	Settlement size
0	100000001	0	0	67	2	124670	1	2
1	100000002	1	1	22	1	150773	1	2
2	100000003	0	0	49	1	89210	0	0
3	100000004	0	0	45	1	171565	1	1
4	100000005	0	0	53	1	149031	1	1

describe () method computes a summary of statistics like count, mean, standard deviation, min, max and quartile values.

```
data.describe()
```

```
data.describe()
```

	ID	Sex	Marital status	Age	Education	Income	Occupation	Settlement size
count	2.000000e+03	2000.000000	2000.000000	2000.000000	2000.00000	2000.000000	2000.000000	2000.000000
mean	1.000010e+08	0.457000	0.496500	35.909000	1.03800	120954.419000	0.810500	0.739000
std	5.774946e+02	0.498272	0.500113	11.719402	0.59978	38108.824679	0.638587	0.812533
min	1.000000e+08	0.000000	0.000000	18.000000	0.00000	35832.000000	0.000000	0.000000
25%	1.000005e+08	0.000000	0.000000	27.000000	1.00000	97663.250000	0.000000	0.000000
50%	1.000010e+08	0.000000	0.000000	33.000000	1.00000	115548.500000	1.000000	1.000000
75%	1.000015e+08	1.000000	1.000000	42.000000	1.00000	138072.250000	1.000000	1.000000
max	1.000020e+08	1.000000	1.000000	76.000000	3.00000	309364.000000	2.000000	2.000000

From the data we infer that there are only decimal values and no categorical ss

info () gives information about the data -

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   ID               2000 non-null   int64
 1   Sex              2000 non-null   int64
 2   Marital status   2000 non-null   int64
 3   Age              2000 non-null   int64
 4   Education        2000 non-null   int64
 5   Income           2000 non-null   int64
 6   Occupation       2000 non-null   int64
 7   Settlement size  2000 non-null   int64
dtypes: int64(8)
memory usage: 125.1 KB
```

Activity 3: Data Pre-processing

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

- Handling missing values
- Handling categorical data
- Handling outliers
- Scaling Techniques
- Splitting dataset into training and test set

Note: These are the general steps of pre-processing the data before using it for machine learning.

1. The Most crucial step in data pre-processing is dealing with missing data, the presence of missing data in the dataset can lead to low accuracy.

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

```
data.isna().sum()
```

```
Sex              0
Marital status   0
Age              0
Education        0
Income           0
Occupation       0
Settlement size  0
dtype: int64
```

3. There are no null values in the dataset, if there are any null/missing values in the columns of the data, we need to fill it. if there are no null values we can skip this step.

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 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 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. **data. Corr ()** gives the correlation between the columns.

Correlation is a statistical term describing the degree to which two variables move in coordination with one another. If the two variables move in the same direction, then those variables are said to have a positive correlation. If they move in opposite directions, then they have a negative correlation

2. A heatmap is a graphical representation of data that uses a system of color-coding to represent different values.it is used to identify the correlation between the columns using a visual manner.

	ID	Sex	Marital status	Age	Education	Income	Occupation	Settlement size
ID	1.000000	0.328262	0.074403	-0.085246	0.012543	-0.303217	-0.291958	-0.378445
Sex	0.328262	1.000000	0.566511	-0.182885	0.244838	-0.195146	-0.202491	-0.300803
Marital status	0.074403	0.566511	1.000000	-0.213178	0.374017	-0.073528	-0.029490	-0.097041
Age	-0.085246	-0.182885	-0.213178	1.000000	0.654605	0.340610	0.108388	0.119751
Education	0.012543	0.244838	0.374017	0.654605	1.000000	0.233459	0.064524	0.034732
Income	-0.303217	-0.195146	-0.073528	0.340610	0.233459	1.000000	0.680357	0.490881
Occupation	-0.291958	-0.202491	-0.029490	0.108388	0.064524	0.680357	1.000000	0.571795
Settlement size	-0.378445	-0.300803	-0.097041	0.119751	0.034732	0.490881	0.571795	1.000000

```
sns.heatmap(cor,annot=True)
```

```
<AxesSubplot:>
```

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

3.Pair Plot: Plot pairwise relationships in a dataset.

- By default, this function will create a grid of Axes such that each numeric variable in data will by shared across the y-axes across a single row and the x-axes across a single column. The diagonal plots are treated differently: a univariate distribution plot is drawn to show the marginal distribution of the data in each column.
- We implement this using the below code

Code: - sns.pairplot (data)

The output is as shown below -

Pair plot usually gives pair wise relationships of the columns in the dataset

From the above pair plot, we infer that

1. from the above plot we can draw inferences such as linearity and strength between the variables

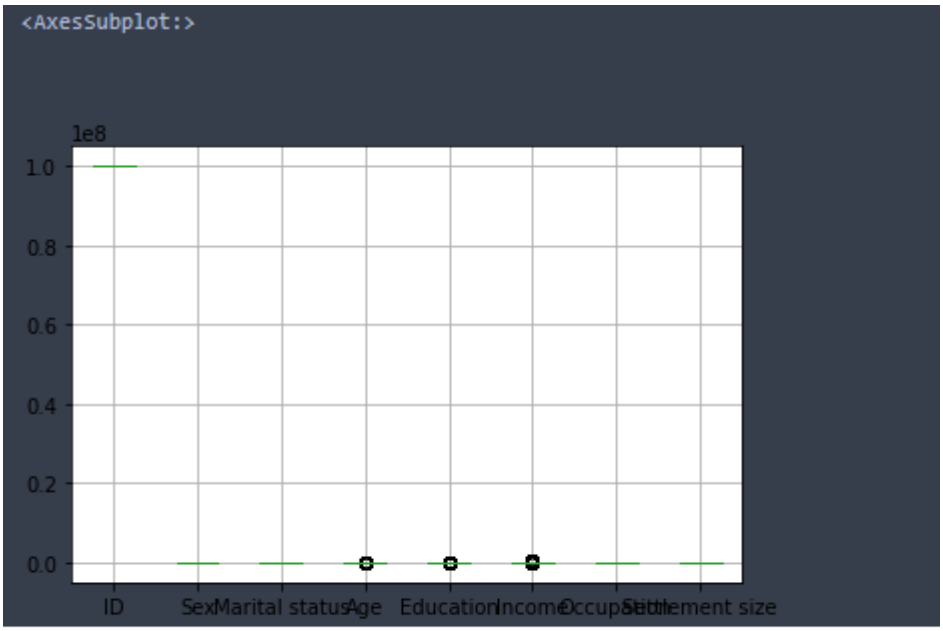
2. how features are correlated (positive, neutral and negative)

4. Box Plot:

Box-plot is a type of chart often used in explanatory data analysis. Box plots visually show the distribution of numerical data and skewness through displaying the data quartiles (or percentiles) and averages.

Box plots are useful as they show the average score of a data set. The median is the average value from a set of data and is shown by the line that divides the box into two parts. Half the scores are greater than or equal to this value and half are less.

Jupiter has a built-in function to create a boxplot called boxplot (). A boxplot plot is a type of plot that shows the spread of data in all the quartiles



From the above box plot we infer how the data points are spread and the existence of the outliers

Feature Scaling

There is a huge disparity between the x values so let us use feature scaling.

Feature scaling is a method used to normalize the range of independent variables or features of data.

```
data = preprocessing.minmax_scale(data,feature_range=(0,1))  #scali

data

                                                              ...

data = pd.DataFrame(data,columns=names) #scaled data will convert t
```

- After scaling the data will be converted into an array form
- Loading the feature names before scaling and converting them back to data frame after standard scaling is applied

- After scaling the data will be converted into an array form

- Loading the feature names before scaling and converting them back to data frame after standard scaling is applied

Unsupervised Model Building

- o Import the model building Libraries
- o Initializing the model
- o Fit and predict the clusters
- o Add the classes to the main data set and save the dataset
- o Splitting x and y
- o Splitting train and test data

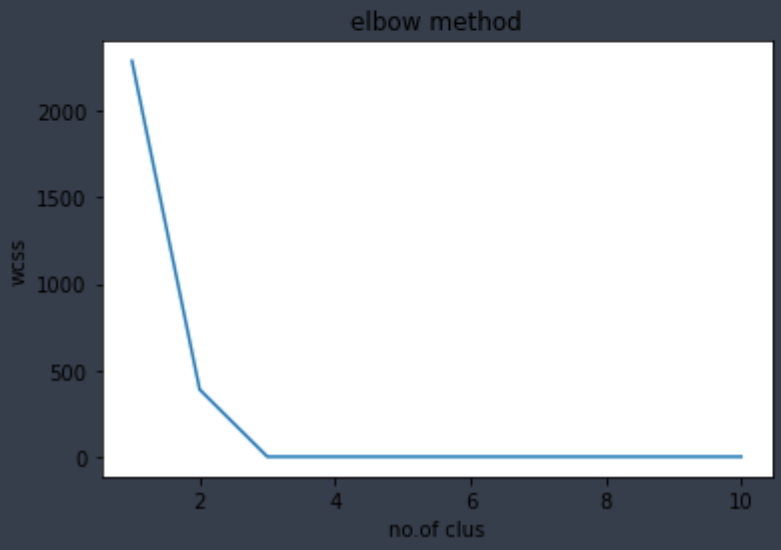
Importing And Initializing the Model

- From sklearn.clusters import K-means
- from SciPy import spatial

For selecting no of clusters, it is essential to plot an elbow curve, from that we can identify how many no. of clusters can be taken

```
wcss = []
for i in range(1,11):
    kmeans = cluster.KMeans(n_clusters=i,init='k-means++',random_state=0)
    kmeans.fit(data)
    wcss.append(kmeans.inertia_)
```

```
plt.pyplot.plot(range(1,11),wcss)
plt.pyplot.title('elbow method')
plt.pyplot.xlabel('no.of clus')
plt.pyplot.ylabel('wcss')
plt.pyplot.show()
```



From the above graph, it can be inferred that the curve has 3 bends (I.e., 0-2 ,2-3, and 3-10, making it 3 clusters.

```
km_model = cluster.KMeans(n_clusters=3,init='k-means++',random_state=0)
```

```
ykmeans = km_model.fit_predict(data)
```

```
data.head()
```

	Sex	Marital status	Age	Education	Income	Occupation	Settlement size
0	0.0	0.0	0.844828	0.666667	0.324781	0.5	1.0
1	1.0	1.0	0.068966	0.333333	0.420210	0.5	1.0
2	0.0	0.0	0.534483	0.333333	0.195144	0.0	0.0
3	0.0	0.0	0.465517	0.333333	0.496223	0.5	0.5
4	0.0	0.0	0.603448	0.333333	0.413842	0.5	0.5

```
data['kclus'] = pd.Series(ykmeans)
```

```
data.head()
```

	Sex	Marital status	Age	Education	Income	Occupation	Settlement size	kclus
0	0.0	0.0	0.844828	0.666667	0.324781	0.5	1.0	2
1	1.0	1.0	0.068966	0.333333	0.420210	0.5	1.0	1
2	0.0	0.0	0.534483	0.333333	0.195144	0.0	0.0	0
3	0.0	0.0	0.465517	0.333333	0.496223	0.5	0.5	2
4	0.0	0.0	0.603448	0.333333	0.413842	0.5	0.5	2

Splitting The Dataset into Dependent and Independent Variable

· In machine learning, the concept of the dependent variable (y) and independent variables(x) is important to understand. Here, the Dependent variable is nothing but output in the dataset and the independent variable is all inputs in the dataset.

· With this in mind, we need to split our dataset into the matrix of independent variables and the vector or dependent variable. Mathematically, Vector is defined as a matrix that has just one column.

To read the columns, we will use iloc of pandas (used to fix the indexes for selection) which takes two parameters — [row selection, column selection].

Let's split our dataset into independent and dependent variables.

```
y = data['kclus']
x = data.drop(columns=['kclus'],axis=1)
```

Activity 4: Splitting data into train and test

When you are working on a model and you want to train it, you obviously 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.

- 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 80% of the dataset to the training set and the remaining 20% to test.
- Now split our dataset into train set and test using train_test_split class from sci-kit learn library.

From scikit=learn import model selection
x_train,x_test,y_train,y_test=model_selection.train_test_split(x,y,test_size=0.2,random_state =0)

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

Activity 5: Model Building

Supervised Model Building

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

Training And Testing the Model

K-Means Clustering:

K-means is an unsupervised learning algorithm used for customer segmentation. It clusters customers into groups based on their spending patterns. Each cluster represents a different customer segment. You can then make predictions about which segment a new customer belongs to.

- 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 are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it may be Classification algorithms are classification algorithms.
    a. Decision Tree classifier
    b. Random Forest classifier
    c. xgboost

Steps in Building the model: -

- **Initialize the model**

Applying supervised learning on the data

```
from sklearn.ensemble import RandomForestClassifier
from sklearn import tree
import xgboost
```

```
rand_model = RandomForestClassifier()
tree_model =  tree.DecisionTreeClassifier()
xgb_model = xgboost.XGBClassifier()
```

fit the initialized models with x_train and y _train data, it means that we are training the models using train data

```
rand_model.fit(x_train,y_train)
tree_model.fit(x_train,y_train)
xgb_model.fit(x_train,y_train)
```

```
C:\Users\SmartbridgePC\anaconda3\lib\site-packages\xgboost\
```

Model Evaluation

- Accuracy testing using the train data

```
pred = rand_model.predict(x_train)
pred1 = tree_model.predict(x_train)
pred2 = xgb_model.predict(x_train)
```

```
from sklearn import metrics
```

```
print(metrics.accuracy_score(pred,y_train))
print(metrics.accuracy_score(pred1,y_train))
print(metrics.accuracy_score(pred2,y_train))
```

```
1.0
1.0
1.0
```

ata

- Predict the y_test values and calculate the accuracy.

```
pred = rand_model.predict(x_test)
pred1 = tree_model.predict(x_test)
pred2 = xgb_model.predict(x_test)
```

```
print(metrics.accuracy_score(pred,y_test))
print(metrics.accuracy_score(pred1,y_test))
print(metrics.accuracy_score(pred2,y_test))
```

- After predicting we will find the accuracy value of each model.

```
0.995
0.9983333333333333
0.9883333333333333
```

- From the above metrics we can conclude that model xgboost gives the best accuracy, other models fall under the category of over-fitting, when measured with train data, so omitting other models and considering the xgboost model for deployment

Activity 5: Save the best 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.

```
import pickle
```

```
pickle.dump(xgb_model,open("xgbmodel.pkl",'wb'))
```

Activity 6: 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

Building HTML pages

- In this HTML page, we will create the front-end part of the web page. In this page we will accept input from the user and predict the values.

For more information regarding HTML link

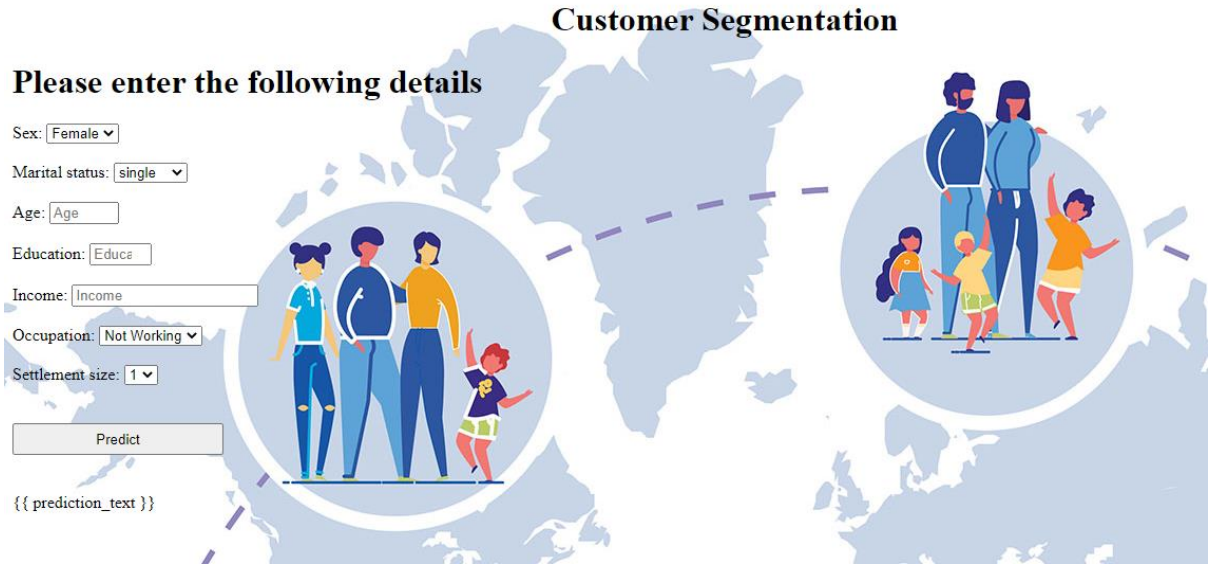
- In our project we have HTML files, they are

1.index.html

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>Customer Segmentation</title>
</head>


<body background="https://www.imf.org/external/pubs/ft/fandd/2020/03/images/032020/picture-1600.jpg" text="black">

 <div class="login">
    <center><h1>Customer Segmentation</h1></center>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
<h1>Please enter the following details</h1>


</style></head>


  <label for="Sex">Sex:</label>
        <select id="Sex" name="Sex">
            <option value=0>Female</option>
            <option value=1>Male</option>

        </select>   <br>


<br><label for="Marital status">Marital status:</label>
        <select id="Marital status" name="Marital status">
            <option value=0>single</option>
            <option value=1>Married</option>

        </select>   <br>

<br>   <label>Age:</label>
     <input type="number" min="20" max="80"   name="Age" placeholder="Age  " required="required" /><br>

 <br>
      <label>Education:</label>
     <input type="number"  min="0" max="3"  name="Education" placeholder="Education" required="required" /><br>
```

```
       <label>Income:</label>
      <input type="number" min="5000"  name="Income " placeholder="Income" required="required" /><br>

<br><label for="Occupation">Occupation:</label>
        <select id="Occupation" name="Occupation">
            <option value=0>Not Working</option>
            <option value=1>Working</option>
           <option value=1>Business</option>
        </select>   <br>

<br>


      <label for="Settlement size">Settlement size:</label>
        <select id="Settlement size" name="Settlement size">
            <option value=1>1</option>
            <option value=0>0</option>
            <option value=2>2</option>


        </select>   <br>
<br><br>


<button type="submit" class="btn btn-primary btn-block btn-large" style="height:30px;width:200px">Predict</button>

    </form>
<br>

<br>

   {{ prediction_text }}


  <br>
  <br>
  <img src="{{url_3}}" alt="Submit Form" height="180" width="233" onerror="this.style.display='none'"/>
  <img src="{{url_1}}" alt="Submit Form" height="180" width="233" onerror="this.style.display='none'"/>
  <img src="{{url_4}}" alt="Submit Form" height="180" width="233" onerror="this.style.display='none'"/>
  <br>
  <br>
  <img src="{{url_2}}" alt="Submit Form" height="150" width="711" onerror="this.style.display='none'"/>
```

The HTML page looks like



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

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. In order to develop web API with respect to our model, we basically use the Flask framework which is written in python.

Importing the necessary libraries for building a flask application and integrating model and HTML pages

Initializing the flask app

- Calling the pkl models and saving into a variable
- Routing and rending to the HTML page
- Calling the inputs from the HTML page and saving into the variable
- Creating the data labels
- Forming the data frame with labels and the data
- Scaling the data
- Predicting the values, by passing the data into the model
- Rendering the results onto the HTML pages based on the output

If the output is class-0, it means a page that displays non-potential customers will be rendered, if the output is 1, a page with the potential customers will be displayed and the output is 2 a page with highly potential customers will be rendered.

- 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

```
import numpy as np
import pickle
import joblib
import matplotlib
import matplotlib.pyplot as plt
import time
import pandas
import os
from flask import Flask, request, jsonify, render_template


app = Flask(__name__)
model = pickle.load(open('G:/AI&ML/ML projects/cluster analysis/xgbmodel.pkl', 'rb'))
scale = pickle.load(open('C:/Users/SmartbridgePC/Desktop/AIML/Guided projects/rainfall_prediction/IBM flask push/Rainfall IBM deploy/scale.pkl','rb'))

@app.route('/')# route to display the home page
def home():
    return render_template('index.html') #rendering the home page

@app.route('/predict',methods=["POST","GET"])# route to show the predictions in a web UI
def predict():
    #  reading the inputs given by the user
    input_feature=[float(x) for x in request.form.values() ]
    features_values=[np.array(input_feature)]
    names = [['Sex', 'Marital status', 'Age', 'Education', 'Income', 'Occupation',
       'Settlement size']]
    data = pandas.DataFrame(features_values,columns=names)
    data = scale.fit_transform(features_values)

     # predictions using the loaded model file
    prediction=model.predict(data)
    print(prediction)

    if (prediction == 0):
       return render_template("index.html",prediction_text ="Not a potential customer")
    elif (prediction == 1):
       return render_template("index.html",prediction_text = "Potential customer")
    else:
       return render_template("index.html",prediction_text = "Highly potential customer")
     # showing the prediction results in a UI
if __name__=="__main__":

    # app.run(host='0.0.0.0', port=8000,debug=True)    # running the app
    port=int(os.environ.get('PORT',5000))
    app.run(port=port,debug=True,use_reloader=False)
```

Run The App

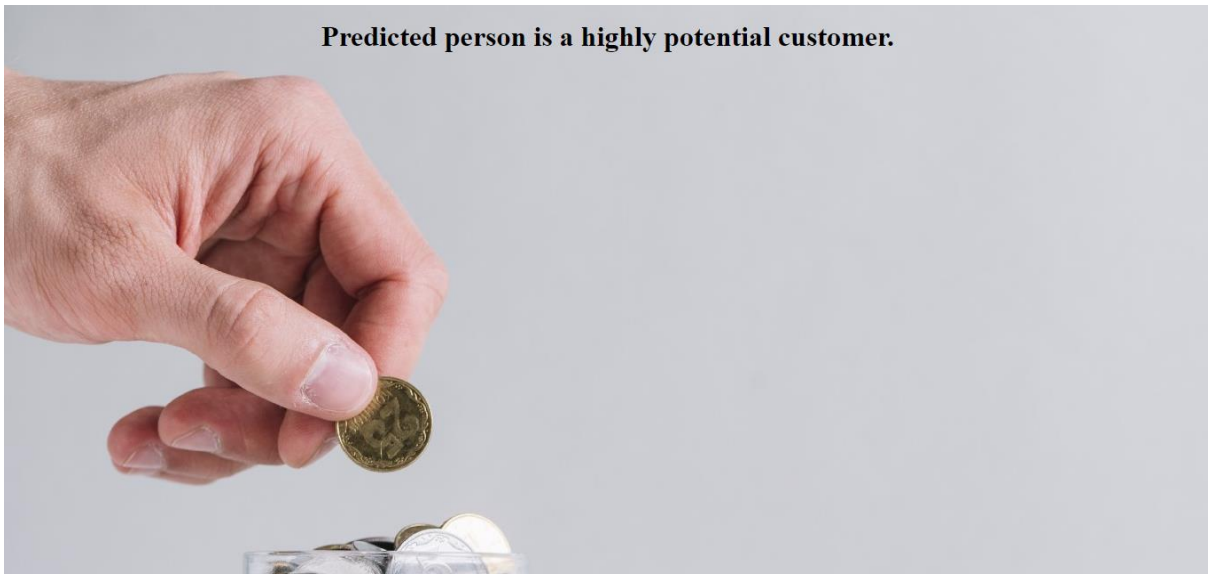
- Open anaconda prompt from the Start menu
- Navigate to the folder where your python script is.
- Now type the “python app.py” command

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

```
Serving Flask app "app" (lazy loading)
Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
Debug mode: on
\Users\SmartbridgePC\anaconda3\lib\site-packages\sklearn\base.py:324:
erWarning: Trying to unpickle estimator StandardScaler from version 0.23.2 when
ing version 1.0.1. This might lead to breaking code or invalid results. Use at
ur own risk. For more info please refer to:
tps://scikit-learn.org/stable/modules/model_persistence.html#security-
intainability-limitations
arnings.warn(
Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

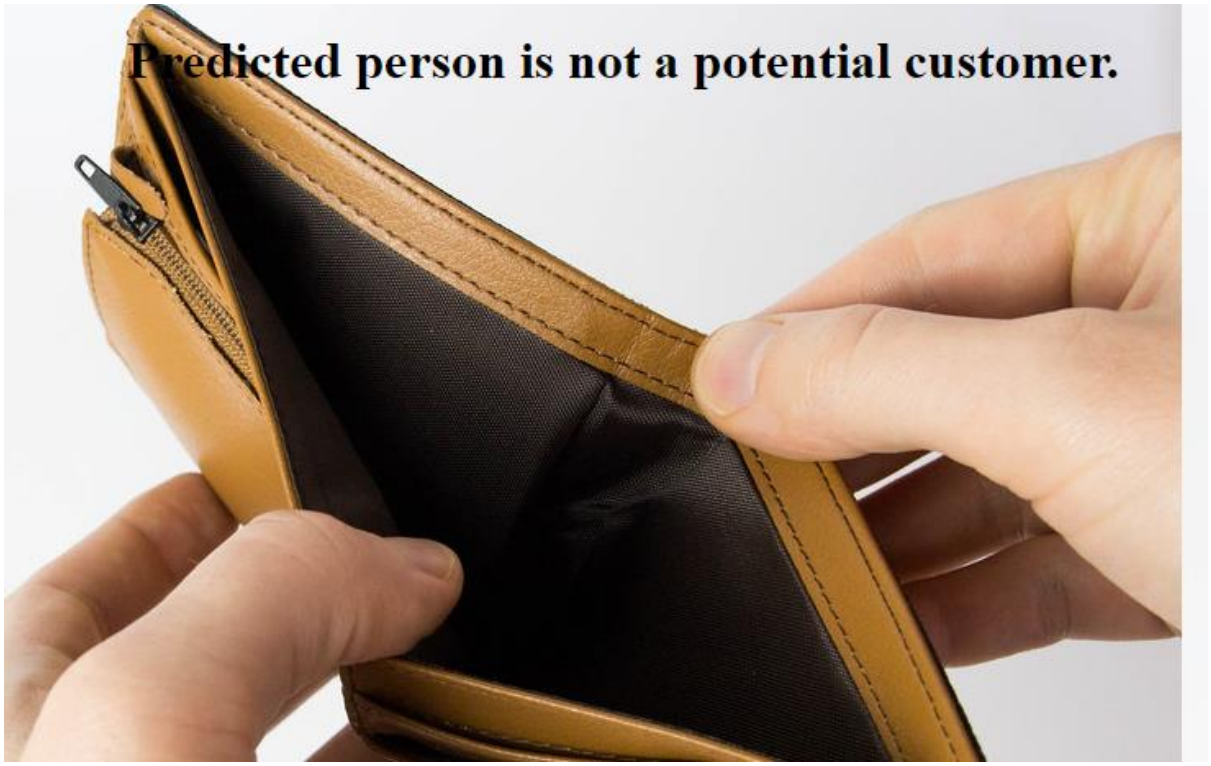
Output:

- Copy the HTTP link and paste it in 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
- If the prediction belongs to class-2, it means that the customer is highly potential



If the prediction belongs to class-1, it means that the customer is potential

If the prediction belongs to class-0, it means the customer is a not potential



7.Advantages

- Improved Customer Retention
- Stay Competitive with Big Retailers
- Turn Shoppers into Loyal Brand Enthusiasts
- Find the Right Price and Offers
- Stop Wasting Time Trying to Reach Cold Audiences
- Discover How Your Audience Wants to Be Engaged
- Riches are in the Niches: Identify Your Big Spenders
- Better Marketing Messaging that Makes an Impression
- Improve Inventory Management and Merchandising
- Gain Audience Insights to Guide Product Development

8. CONCLUSION

Implementing customer segmentation using machine learning involves a comprehensive process that includes data collection, preprocessing, model selection, and continuous monitoring. While it can be challenging, the benefits it brings in terms of business intelligence and customer-centric strategies make it a worthwhile investment.

By continually updating segmentation models and adapting to evolving customer behaviors, businesses can maintain their relevance and competitiveness in the market. Customer segmentation through machine learning transforms data into valuable insights, empowering companies to deliver more personalized and effective interactions, ultimately driving sustainable growth and success.

In conclusion, machine learning-driven customer segmentation is a transformative approach that equips businesses with the tools to better understand their customers, optimize their strategies, and achieve their strategic goals in an increasingly competitive landscape.

9.REFERENCES

Dataset Reference:

Dataset Name: Customer segmentation using Machine Learning

Data Source: Kaggle

URL:https://docs.google.com/spreadsheets/d/1NnUMX3sjJgRRerkJTAXemlfdyo2GiUhgE_m4w-fAhvs/edit?gid=1219451115#gid=1219451115

Description: This dataset was obtained from Kaggle and served as a valuable resource for training and testing the machine learning models in our project.

Machine Learning Libraries Scikit-learn is an open-source machine learning library for Python that was used to implement and evaluate the machine learning models in our project.

Web framework:

Flask

URL: http://127.0.0.1:5000

Description: Flask is a lightweight and powerful web framework used to create the web application interface for our project.

SOURCE CODE OF FLASK:
My git repository
link:https://github.com/DivyasriKusuri/SmartBridge/tree/main/Customer%20Segmentation