Market Segmentation Analysis Using ML

Project Description:

- In this report, we are going through the Steps of Market segmentation and covering the basic

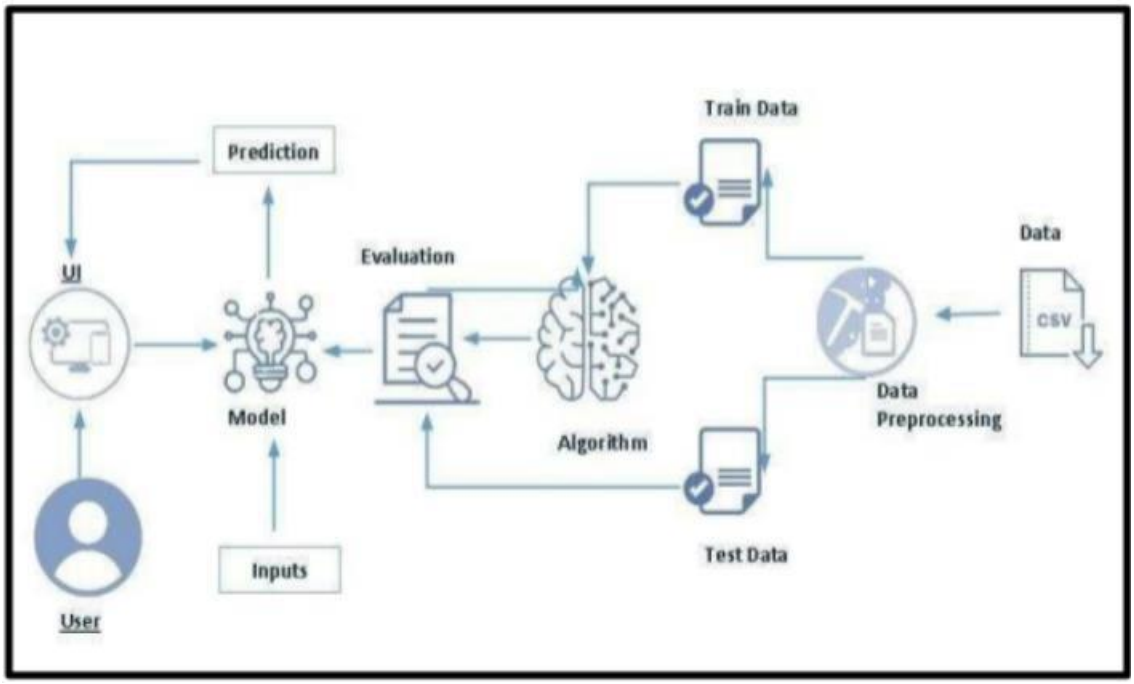
idea of market segmentation. We will be covering all the instruction or procedure you have to keep in mind during the market segmentation.

• The purpose of marketing is to match the genuine needs and desires of consumers with the

offers of suppliers particularly suited to satisfy those needs and desires. This matching process

benefits consumers and suppliers, and drives an organization's marketing planning process.

Technical Architecture:



Milestone 1 : Define Problem/Problem Understanding

Activity 1: Specify the business problem

• The Market Segmentation Analysis Using ML aims to analyze the spending behaviour of

customers and identify opportunities for growth. The data set consists of spending

('yummy','convenient','spicy','fattening','greasy','fast','cheap','tasty','expensive', 'healthy',

'disgusting', 'Like','Age','VisitFrequency','Gender',) of the customer.

• 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 behaviours, the project aims to provide insights

on how w businesses can tailor their marketing strategies and product offerings to better

serve each customer segment. The project also aims to identify opportunities for growth,

such as which products or product categories are underrepresented among customers, and

which segments may be receptive to new product offerings.

• Overall, the project seeks to provide valuable insights for wholesale businesses on how to

optimize their operations and increase customer satisfaction and retention.

Activity 2: Business requirements

Here are some potential business requirements for Market Segmentation Analysis

Using ML.

Accurate forecasting: The predictor must be able to accurately forecast the spending

behaviour of customers.

User-friendly interface: The predictor must have a user-friendly interface that is easy

to navigate and understand. The interface should present the results of the predictor in a clear and

concise manner to provide valuable insights for businesses on how to optimize their operations

and increase customer satisfaction and retention.

Activity 3: Literature Survey

Market Segmentation is the process of dividing customers into groups based on their

shared characteristics, such as spending habits, location, or industry. This can be a valuable tool for

wholesale businesses to better understand their customers and tailor their marketing and sales

strategies accordingly.

There is a growing body of literature on wholesale customer segmentation. A 2019

study by the Aberdeen Group found that businesses that use customer segmentation are more likely

to achieve their revenue and profit goals than those that do not. The study also found that

businesses that use customer segmentation are better able to:

Target their marketing campaigns more effectively Develop products and services that

meet the needs of their customers Increase customer satisfaction and retention

There are a number of different ways to segmentcustomers. Some common methods

include:

• **Geographic segmentation**: This involves dividing customers into groups based on their

location. This can be a useful way to target customers with local marketing campaigns or to

tailor product offerings to meet the needs of customers in different regions.

• **Demographic segmentation**: This involves dividing customers into groups based on their

age, gender, income, or other demographic characteristics. This can be a useful way to

target customers with specific products or services.

• **Behavioral segmentation**: This involves dividing customers into groups based on their

buying habits, such as the products they purchase, the frequency of their purchases, or the

amount they spend. This can be a useful way to identify customers who are most likely to

respond to a particular marketing campaign or to develop new products or services that

meet the needs of these customers.

Wholesale customer segmentation can be a valuable tool for businesses of all sizes. By

understanding their customers and their needs, businesses can better tailor their marketing and

sales strategies to achieve their goals.

Activity 4: Social or Business Impact.

The social and business impact of the Wholesale Customer Segmentation project are as

follows:

• Increased customer satisfaction and retention: By understanding the spending behavior

of their customers, wholesale businesses can tailor their marketing strategies and product

offerings to better meet the needs of each customer segment. This can lead to increased

customer satisfaction and retention, as customers are more likely to do business with

companies that understand their needs and preferences.

• Improved operational efficiency: By identifying opportunities for growth, such as which

products or product categories are underrepresented among customers, and which segments

may be receptive to new product offerings, wholesale businesses can improve their

operational efficiency. This can be done by streamlining their supply chain, optimizing

their inventory management, and allocating resources more effectively.

• Increased profitability: By improving customer satisfaction and retention, and by

improving operational efficiency, wholesale businesses can increase their profitability. This

can be done by generating more revenue from existing customers, by acquiring new

customers, and by reducing costs.

Milestone 2: Data Collection

ML depends heavily on data, It is most crucial aspect that makes algorithm training

possible. So this section allows you to download the required dataset.

Activity 1: Download the dataset

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

repository, etc.

In this project we have used The Market Segmentation Analysis Using ML Mcdonald's

data.

Milestone 3: Visualizing and analysing the data

As the dataset is downloaded. Let us read and understand the data properly with the

help of some visualization 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

```
[ ] import pandas as pd
    import numpy as np
    import os
    import matplotlib.pyplot as plt
    from sklearn.decomposition import PCA
    import seaborn as sns
```

Activity 2: Read the Dataset:

```
[ ] dataset=pd.read_csv("Dataset.csv")
```

```
dataset
```

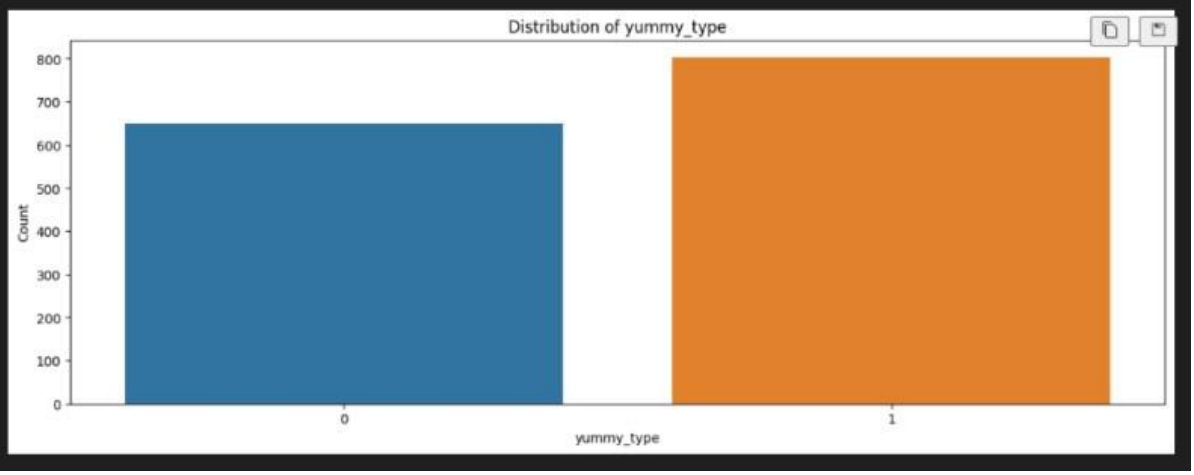
	yummy	convenient	spicy	fattening	greasy	fast	cheap	tasty	expensive	healthy	disgusting	Like	Age	VisitFrequency	Gender
0	No	Yes	No	Yes	No	Yes	Yes	No	Yes	No	No	-3	61	Every three months	Female
1	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	+2	51	Every three months	Female
2	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	+1	62	Every three months	Female
3	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No	Yes	+4	69	Once a week	Female
4	No	Yes	No	Yes	Yes	Yes	Yes	No	No	Yes	No	+2	49	Once a month	Male
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1448	No	Yes	No	Yes	Yes	No	No	No	Yes	No	Yes	I hate it!	-5	47	Once a year	Male
1449	Yes	Yes	No	Yes	No	No	Yes	Yes	No	Yes	No	+2	36	Once a week	Female
1450	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes	No	No	+3	52	Once a month	Female
1451	Yes	Yes	No	No	No	Yes	Yes	Yes	No	Yes	No	+4	41	Every three months	Male
1452	No	Yes	No	Yes	Yes	No	No	No	Yes	No	Yes	-3	30	Every three months	Male

1453 rows × 15 columns

Activity 3: Univariate analysis:

```
import seaborn as sns
plt.figure(figsize=(15,5))
plt.title('Distribution of yummy_type')
sns.barplot(x='yummy_type',y='Count',data=yummy_df)
```

```
<Axes: title={'center': 'Distribution of yummy_type'}, xlabel='yummy_type', ylabel='Count'>
```



Activity 4: Descriptive analysis:

```
dataset.describe()
```

	Age
count	1453.000000
mean	44.604955
std	14.221178
min	18.000000
25%	33.000000
50%	45.000000
75%	57.000000
max	71.000000

```
dataset.describe(include="all")
```

	yummy	convenient	spicy	fattening	greasy	fast	cheap	tasty	expensive	healthy	disgusting	Like	Age	VisitFrequency	Gender
count	1453	1453	1453	1453	1453	1453	1453	1453	1453	1453	1453	1453	1453.000000	1453	1453
unique	2	2	2	2	2	2	2	2	2	2	2	11	NaN	6	2
top	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No	No	+3	NaN	Once a month	Female
freq	803	1319	1317	1260	765	1308	870	936	933	1164	1100	229	NaN	439	788
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	44.604955	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	14.221178	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	18.000000	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	33.000000	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	45.000000	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	57.000000	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	71.000000	NaN	NaN

Milestone 4: Data Pre-processing

Activity 1: Checking for null values:

```
[ ] dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1453 entries, 0 to 1452
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   yummy           1453 non-null   object
 1   convenient      1453 non-null   object
 2   spicy           1453 non-null   object
 3   fattening       1453 non-null   object
 4   greasy          1453 non-null   object
 5   fast            1453 non-null   object
 6   cheap           1453 non-null   object
 7   tasty           1453 non-null   object
 8   expensive       1453 non-null   object
 9   healthy         1453 non-null   object
 10  disgusting      1453 non-null   object
 11  Like            1453 non-null   object
 12  Age             1453 non-null   int64
 13  VisitFrequency  1453 non-null   object
 14  Gender          1453 non-null   object
dtypes: int64(1), object(14)
memory usage: 170.4+ KB
```

```
dataset.isna().sum()
```

```
yummy            0
convenient       0
spicy            0
fattening        0
greasy           0
fast             0
cheap            0
tasty            0
expensive        0
healthy          0
disgusting       0
Like             0
Age              0
VisitFrequency   0
Gender           0
dtype: int64
```

Activity 2: Handling outliers:

```python
segment_df.Age.describe([.75,.90,.95,.99])
```

```
count    1453.000000
mean       44.604955
std        14.221178
min        18.000000
50%        45.000000
75%        57.000000
90%        63.000000
95%        66.000000
99%        70.000000
max        71.000000
Name: Age, dtype: float64
```

```
sns.distplot(df["Age"])
```

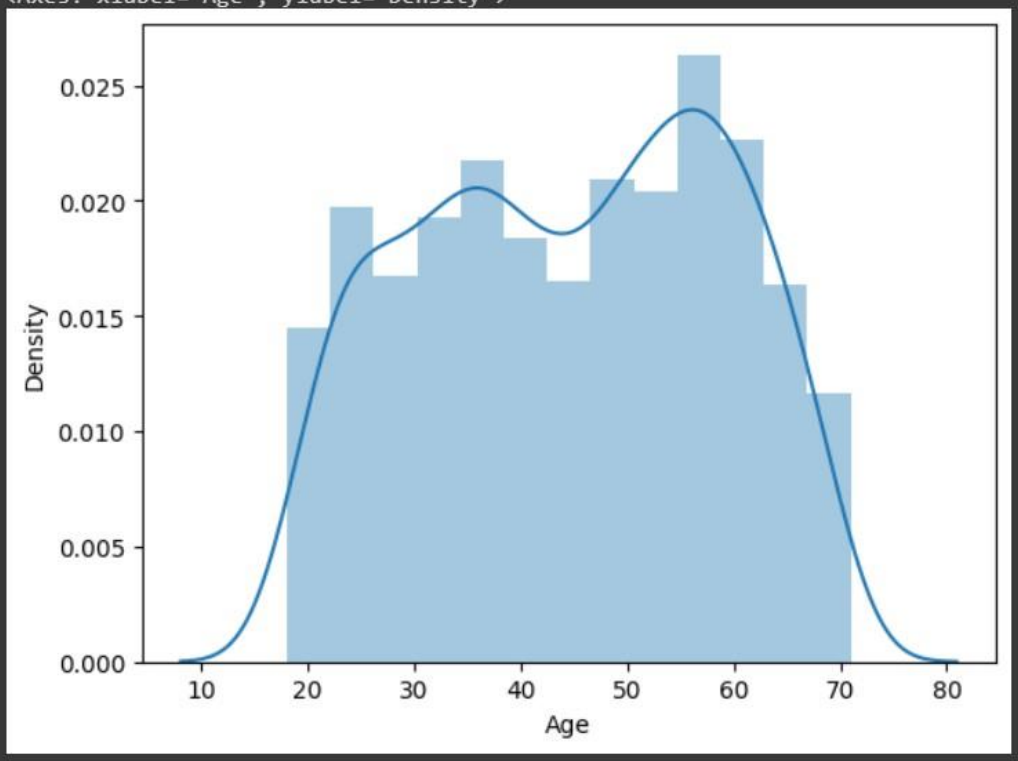
```
<ipython-input-23-cf0334540b62>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df["Age"])
<Axes: xlabel='Age', ylabel='Density'>
```



Activity 3: Splitting data into train and test

```
[ ] from sklearn.model_selection import train_test_split
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_sta
```

```
[ ] x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((1017, 10), (436, 10), (1017, 1), (436, 1))
```

```
[ ] a=[1,2,3,4,5,6]    # 4 values for training and 2 for testing
    b=[1,0,1,5,6,3]

    for i in range(5):
        a_train,a_test,b_train,b_test=train_test_split(a,b,test_size=0.3)
        print("with random state",a_train)
```

```
with random state [3, 5, 4, 6]
with random state [1, 5, 3, 2]
with random state [6, 1, 5, 3]
with random state [4, 2, 1, 6]
with random state [6, 4, 1, 3]
```

```
[ ] a=[1,2,3,4,5,6]    # 4 values for training and 2 for testing
    b=[1,0,1,5,6,3]

    for i in range(5):
        a_train,a_test,b_train,b_test=train_test_split(a,b,test_size=0.3)
        print("without random state",a_train)
```

```
without random state [6, 1, 4, 2]
without random state [6, 3, 1, 2]
without random state [1, 5, 2, 4]
without random state [3, 1, 5, 6]
without random state [1, 6, 4, 5]
```

```
[ ] from sklearn.preprocessing import StandardScaler
    sc=StandardScaler()
```

```
x_train
```

	fattening	greasy	fast	cheap	expensive	healthy	disgusting	Like	Age	Yes
1382	Yes	Yes	Yes	No	Yes	No	Yes	I hate it!-5	42	0
418	Yes	No	Yes	Yes	No	No	No	+2	59	1
891	Yes	No	Yes	Yes	Yes	No	No	I love it!+5	55	1
759	Yes	Yes	Yes	No	Yes	No	Yes	-4	60	0
443	No	No	No	Yes	No	Yes	No	+3	67	1
...	...	...	...	...	...	...	...	...	...	...
763	Yes	Yes	Yes	Yes	Yes	No	No	+1	49	1
835	Yes	Yes	Yes	Yes	No	No	No	-1	53	0
1216	Yes	Yes	Yes	No	No	No	No	+1	27	1
559	Yes	No	Yes	Yes	No	No	No	0	59	1
684	Yes	No	Yes	No	No	No	No	+3	37	0

1017 rows × 10 columns

```
y_train
```

	VisitFrequency
1382	Never
418	Once a week
891	Once a week
759	Once a year
443	Every three months
...	...
763	Once a year
835	Once a year
1216	Once a year
559	Never
684	Once a month

1017 rows × 1 columns

Milestone 5: Model Building

```
[ ] from sklearn.linear_model import LogisticRegression
    model=LogisticRegression()

[ ] model.fit(x_train,y_train)

    ▾ LogisticRegression
    LogisticRegression()

[ ] model.fit(x_train,y_train)

[ ] pred

    array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
           0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
           0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1])
```

```
y_test
```

	VisitFrequency
1352	Every three months
482	Once a month
1309	Once a month
270	Once a year
278	Once a week
...	...
432	Once a month
1220	Never
517	Once a week
1392	Never
649	Once a month

436 rows × 1 columns

```
df
```

	yummy	convenient	spicy	fattening	greasy	fast	cheap	tasty	expensive	healthy	disgusting	Like	Age	VisitFrequency	Gender
0	No	Yes	No	Yes	No	Yes	Yes	No	Yes	No	No	-3	61	Every three months	Female
1	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	+2	51	Every three months	Female
2	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	+1	62	Every three months	Female
3	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No	Yes	+4	69	Once a week	Female
4	No	Yes	No	Yes	Yes	Yes	Yes	No	No	Yes	No	+2	49	Once a month	Male
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1448	No	Yes	No	Yes	Yes	No	No	No	Yes	No	Yes	I hate it! -5	47	Once a year	Male
1449	Yes	Yes	No	Yes	No	No	Yes	Yes	No	Yes	No	+2	36	Once a week	Female
1450	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes	No	No	+3	52	Once a month	Female
1451	Yes	Yes	No	No	No	Yes	Yes	Yes	No	Yes	No	+4	41	Every three months	Male
1452	No	Yes	No	Yes	Yes	No	No	No	Yes	No	Yes	-3	30	Every three months	Male

1453 rows × 15 columns

Milestone 6: Application Building

Activity1: Building Html Pages:

index.html × app.py Market_Segmentation_Analysis.ipynb Market_Segmentation_ML_Models.ipynb

templates > index.html > html > body > main#main > section#features.features > div.container > div.row > div.image.col-lg-6

```
    </div>
  </section><!-- End Hero -->

  <main id="main">

    <!-- ======= Features Section ======= -->
    <section id="features" class="features">
      <div class="container" data-aos="fade-up">
<div class="row">
          <div class="image col-lg-6" style='background-image: url("https://www.start.io/wp-content/uploads/2022
          <div class="col-lg-6" data-aos="fade-left" data-aos-delay="100">
            <div class="icon-box mt-5 mt-lg-0" data-aos="zoom-in" data-aos-delay="150">
              <i class="bx bx-receipt"></i>
              <h4>Data Collection and Preprocessing</h4>
              <p>The first step involves collecting customers data and preprocessing it to handle missing values
            </div>
            <div class="icon-box mt-5
        " data-aos="zoom-in" data-aos-delay="150">
              <i class="bx bx-cube-alt"></i>
              <h4>Feature Engineering and Model Selection</h4>
              <p>The second step involves selecting relevant features and transforming them into a format suitab
            </div>
            <div class="icon-box mt-5" data-aos="zoom-in" data-aos-delay="150">
              <i class="bx bx-images"></i>
              <h4>Model Training and Evaluation</h4>
              <p>The third step involves training the selected model using the preprocessed data and evaluating
            </div>
            <div class="icon-box mt-5" data-aos="zoom-in" data-aos-delay="150">
              <i class="bx bx-shield"></i>
              <h4>Model Deployment</h4>
              <p>The final step involves deploying the model in a real-world scenario To predict the customer be
            </div>
          </div>
```

Activity 2: Build Python code:

Import the libraries:

```
from flask import Flask, render_template, url_for,request
import pickle as p
import pickle
from flask import Flask,request,jsonify,render_template
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
```

Importing flask module:

```
modelfile = 'models/final_prediction.pickle'
model = p.load(open(modelfile, 'rb'))
scaler= pickle.load(open('models/scaler.pickle','rb'))
app = Flask(__name__)
```

Render HTML page:

```
@app.route('/')
def welcome():
    return render_template('index.html')
```

Retrieves the value from UI:

```
@app.route('/predict',methods =['GET','POST'])
def predict():
    yummy = float(request.form["yummy"])
    convenient =float(request.form['convenient'])
    spicy = float(request.form['spicy'])
    fattening=float(request.form['fattening'])
    greasy = float(request.form['greasy'])
    fast  = float(request.form['fast'])
    cheap= float(request.form['cheap'])
    tasty =float(request.form['tasty'])
    expensive = float(request.form['expensive'])
    healthy=float(request.form['healthy'])
    disgusting = float(request.form['disgusting'])
    Age  = float(request.form['Age'])
    Gender= float(request.form['Gender'])


    total = [[yummy, convenient, spicy, fattening, greasy, fast, cheap,
       tasty, expensive, healthy, disgusting, Age, Gender]]
    prediction = model.predict(scaler.transform(total))
    prediction = int(prediction[0])

    if prediction==0:
        return render_template('index.html',predict="Predicts Customer belong to cluster 0"

    if prediction==1:
        return render_template('index.html',predict="Predicts Customer belong to cluster 1"
    if prediction==2:
        return render_template('index.html',predict="Predicts Customer belong to cluster 2"

    else:
        return render_template('index.html',predict="Predicts Customer belong to cluster 3"
```

Main Function:

```
if __name__ == "__main__":
    app.run(debug = True)
```

Activity 3: Run the application

Final Output :

yummy

convenient

spicy

fattening

greasy

fast

cheap

tasty

expensive