

PROJECT REPORT FORMAT

1.INTRODUCTION

1.1 Project Overview: Market segmentation is a critical aspect of marketing strategy.

- Utilizing Machine Learning (ML) for segmentation enhances precision and efficiency.

1.2 Purpose:

Enhance marketing strategy precision and efficiency by employing Machine Learning for market segmentation, enabling targeted campaigns and personalized approaches, ultimately improving overall return on investment (ROI).

2.LITERATURE SURVEY: [2]

2.1 Existing Problem: Current market segmentation approaches lack precision and efficiency, often relying on traditional methods that struggle to handle diverse and dynamic datasets. This leads to suboptimal targeting and personalization in marketing strategies, hindering overall effectiveness and ROI.

2.2 References:

[1] Blanchard, Tommy. Bhatnagar, Pranshu. Behera,

Trash. (2019). Marketing Analytics Scientific

Data: Achieve your marketing objectives with

Python's data analytics capabilities. S.l: Packt

printing is limited

[2]Griva, A., Bardaki, C., Pramadari, K.,

Papakiriakopoulos, D. (2018). Sales business

analysis: Customer categories use market basket

data. Systems Expert Systems, 100, 1-16.

[3] Hong, T., Kim, E. (2011). It separates consumers

from online stores based on factors that affect the

customer's intention to purchase. Expert System

Applications, 39 (2), 2127-2131.

[4] Hwang, Y. H. (2019). Hands-on Advertising

Science Data: Develop your machine learning

marketing strategies... using python and r. S.l:

Packt printing is limited

[5] Puwanenthiren Premkanth, - Market

Classification and Its Impact on Customer

Satisfaction and Special Reference to the

Commercial Bank of Ceylon PLC. Global

Journal of Management and Business Publisher

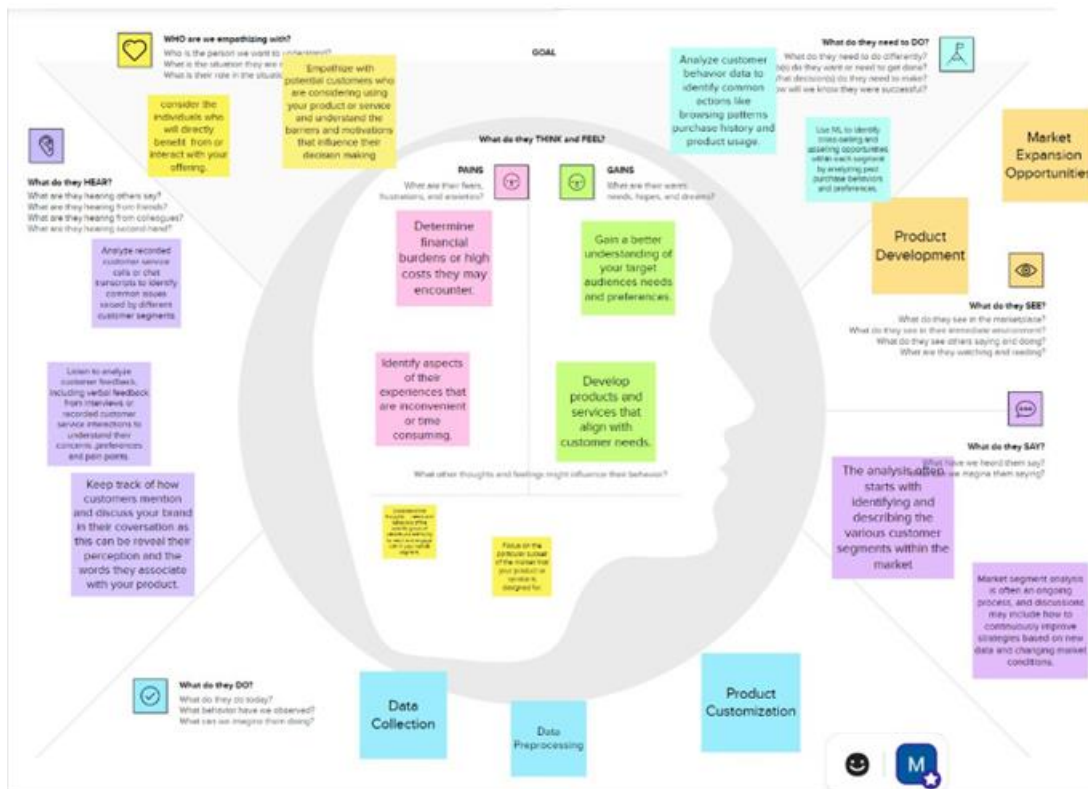
Research: Global Magazenals Inc. (USA). 2012.

Print ISSN: 0975-5853. Volume 12 Issue 1.

2.3 Problem Statement Definition: Develop an efficient and precise market segmentation strategy using Machine Learning to overcome the limitations of traditional methods. Address the challenge of suboptimal targeting and personalization in marketing due to the inability to handle diverse and dynamic datasets effectively.

3.IDEATION&PROPOSED SOLUTION:

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:

Step-1: Team Gathering, Collaboration and Select the Problem Statement:

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- 10 minutes to prepare
- 1 hour to collaborate
- 2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

- Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
- Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.
- Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#)

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

How might we improve communication and collaboration among remote team members?

PROBLEM

How might we streamline our supply chain for cost savings?

PROBLEM

How might we reduce employee turnover in our organization?

PROBLEM

How might we increase the sustainability of our packaging for our products?

Step-2: Brainstorm, Idea Listing and Grouping:

Brainstorm

Write down any ideas that come to mind that address your problem statement.

15 minutes

TIP
You can select a sticky note and/or the pencil (which is sticky) icon to start drawing!

Swathi

Brindha

By implementing better inventory management

By transitioning to eco-friendly packaging

By streamlining the interface

By implementing better portion control

By implementing a customer loyalty program

By conducting user research

By implementing a customer rewards

By transitioning to optimizing packaging designs

Eswar Sripadh

Abhinaya

By improving app performance

by enhancing control over portion sizes.

By conducting user research, streamlining the interface, and improving app performance

By enhancing the supervision of inventory.

Switch to environmentally friendly packaging materials

Conduct a life cycle analysis to identify areas for improvement in sustainability

by optimizing packaging designs.

By implementing a customer loyalty program with rewards and personalized offers

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

TIP
Add customizable tags to sticky notes to make it easier to find, remove, organize, and categorize important ideas as they're written your mind.

By conducting user research, streamlining the interface, and improving app performance

By enhancing the supervision of inventory.

Switch to environmentally friendly packaging materials

By transitioning to eco-friendly packaging materials and optimizing packaging designs

Step-3: Idea Prioritization:

4

Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Tip
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the W key on the keyboard.

After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
[Open the template ->](#)

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
[Open the template ->](#)

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
[Open the template ->](#)

4. Requirement Analysis:

4.1 Functional Requirements:

Functional Requirements for Market Segmentation Analysis Using ML:

Data Collection:

- The system should collect diverse data sources, including demographic, behavioral, and transactional data.
- Data gathering should support seamless integration with ML algorithms.

Preprocessing:

- The system must clean and preprocess data to handle missing values, outliers, and inconsistencies.
- It should support feature selection and transformation for optimal model performance.

Machine Learning Models:

- a. Implementation of clustering algorithms (e.g., K-means, hierarchical clustering) for segmentation.
- b. The system should facilitate model training, validation, and fine-tuning.

Interpretation and Insights:

- a. Provide tools for interpreting results and developing actionable insights from segment profiles.
- b. Ensure user-friendly interfaces for effective analysis.

Integration with Marketing Strategy:

- a. Enable seamless integration with marketing tools and platforms.
- b. Support customization of marketing campaigns based on segment characteristics.

4.2 Non-Functional Requirements:**Performance:**

- a. The system should process and analyze large datasets efficiently.
- b. Achieve low-latency response times for real-time decision-making.

Scalability:

- a. The solution must scale to handle growing datasets and user demands.

Reliability:

- a. Ensure the reliability of ML models for consistent and accurate segmentation.
- b. Implement mechanisms to handle and recover from system failures gracefully.

Security:

- a. Implement robust security measures to protect sensitive customer data.

Usability:

- a. Provide a user-friendly interface for users with varying levels of technical expertise.
- b. Include documentation and training materials for system users.

Interoperability:

- a. Ensure compatibility with various data sources, marketing tools, and platforms.
- b. Support industry-standard data formats for seamless integration.

Maintainability:

- a. Design the system with modular components to facilitate updates and maintenance.
- b. Provide tools for monitoring and troubleshooting.

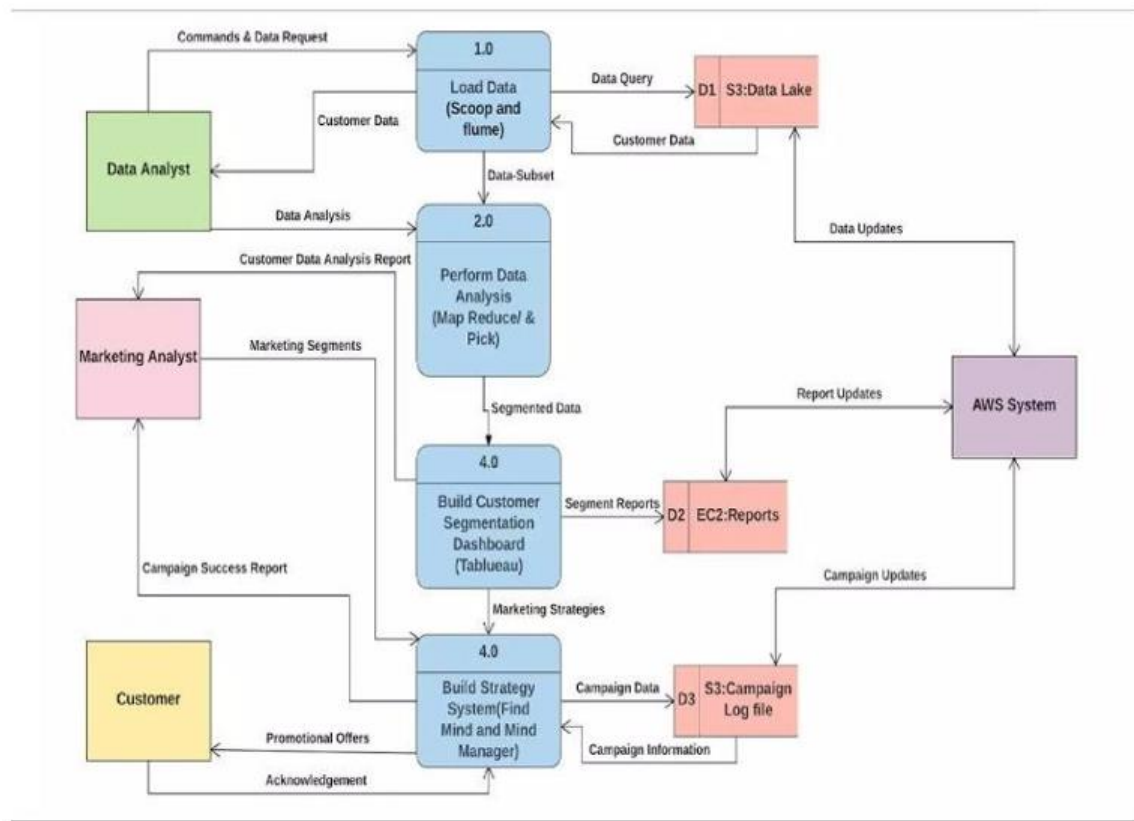
Compliance:

- a. Ensure compliance with data protection regulations and ethical considerations.

5. PROJECT DESIGN:

5.1 Data Flow Diagrams & User Stories:

Data Flow Diagrams:



User stories:

User Stories:

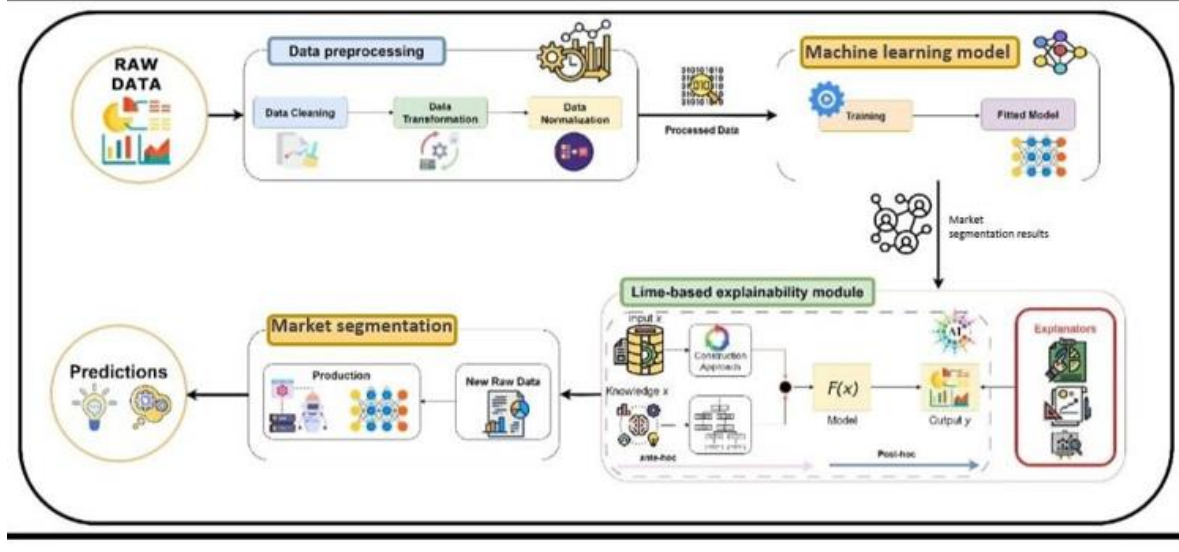
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a new user, I want to register with a valid email and password.	I can access and register with a valid email and password.	High	Sprint-1
		USN-2	As a user, I want to receive a confirmation email after successful registration.	I can access and receive a confirmation email after successful registration.	Medium	Sprint-1
		USN-3	As a user, I want the registration process to include basic profile information.	I can access basic profile information.	High	Sprint-1
	Login	USN-4	As a user, I want to see informative error	I can access credential securely.	Low	Sprint-2

2 / 3



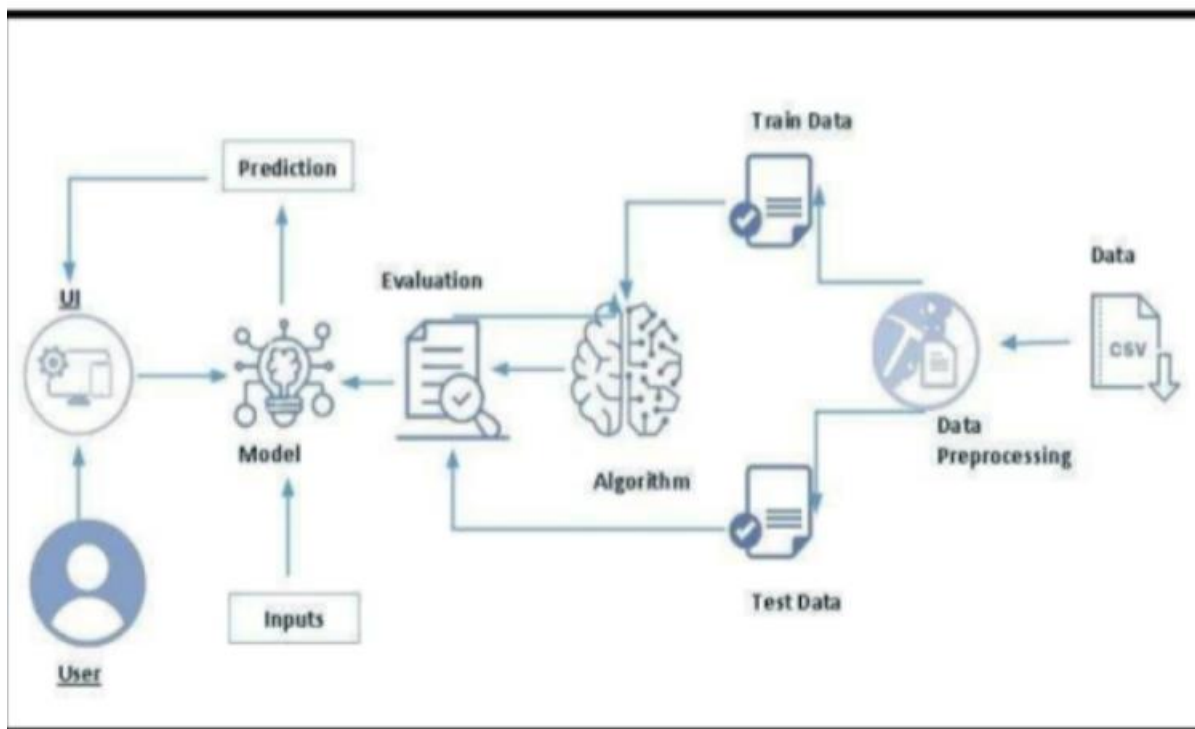
			messages if my login attempt fails.			
	Dashboard	USN-5	As a user, I want the option to export dashboard data for further analysis	I can access the option to export dashboard data for further analysis.	High	Sprint-1

5.2 Solution Architecture:



6. PROJECT PLANNING & SCHEDULING:

6.1 Technical Architecture:



6.2 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	MVL Swathi
Sprint-2		USN-2	As a user, I can register for the application through Facebook	2	Low	Eswar Sripadh
Sprint-1	Login	USN-3	As a user, I can register for the application through Gmail	2	Medium	Abhinaya
Sprint-1	Dashboard	USN-4	As a user, I can log into the application by entering email & password	1	High	Brindha Shri

6.3 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	12 Nov 2023	18 Nov 2023	20	18 Nov 2023

--	--	--	--	--	--	--

Sprint-2	20	6 Days	19 Nov 2023	22 Nov 2023		
----------	----	--------	-------------	-------------	--	--

7. CODING & SOLUTIONING:

7.1 Feature 1:

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:

```
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:
```

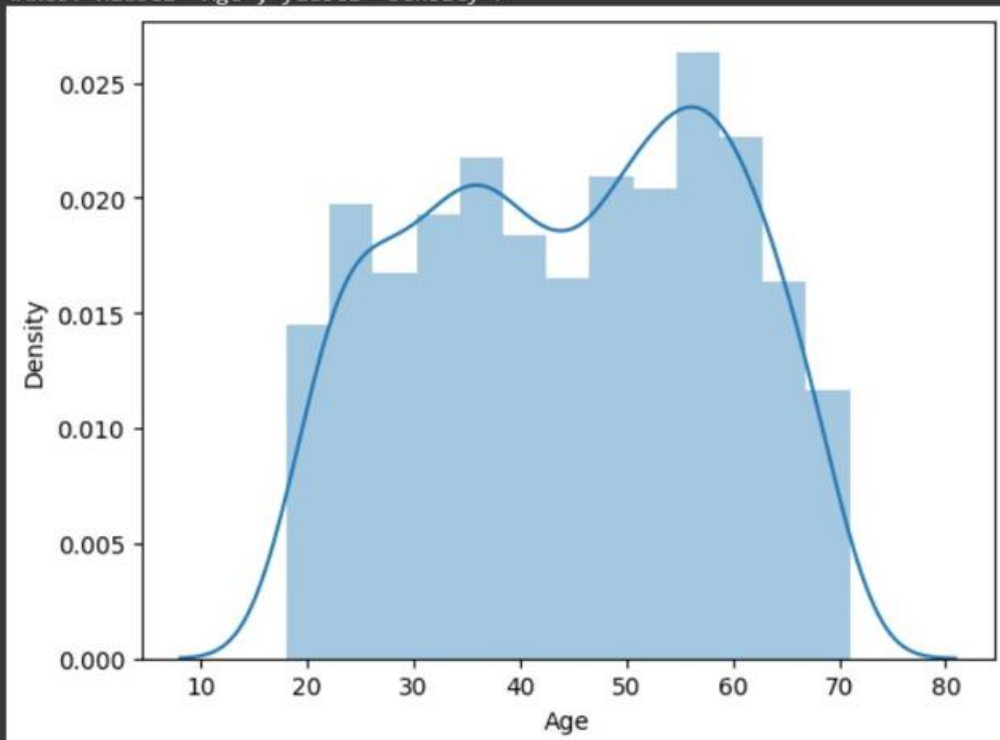
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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_state=1)
```

```
[ ] 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 x 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, 1, 0, 0, 1,
        0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 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, 1, 0, 0, 0, 1,
        0, 0, 0, 0, 1, 0, 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	
1460	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes	No	No	+3	52	Once a month	Female	
1461	Yes	Yes	No	No	No	Yes	Yes	Yes	No	Yes	No	+4	41	Every three months	Male	
1462	No	Yes	No	Yes	Yes	No	No	No	Yes	No	Yes	-3	30	Every three months	Male	

1453 rows × 15 columns

7.2 Feature 2:

Milestone 6: Application Building

Activity1: Building Html Pages:

```

templates > index.html > html > body > main#main > section#features.features > div.container > div.row > div.image.col-lg-6
76 </div>
77 </section><!-- End Hero -->
78
79 <main id="main">
80
81 <!-- ===== Features Section ===== -->
82 <section id="features" class="features">
83 <div class="container" data-aos="fade-up">
84 <div class="row">
85 <div class="image col-lg-6" style="background-image: url('https://www.start.io/wp-content/uploads/2022
86 <div class="col-lg-6" data-aos="fade-left" data-aos-delay="100">
87 <div class="icon-box mt-5 mt-lg-0" data-aos="zoom-in" data-aos-delay="150">
88 <i class="bx bx-receipt"></i>
89 <h4>Data Collection and Preprocessing</h4>
90 <p>The first step involves collecting customers data and preprocessing it to handle missing values
91 </div>
92 <div class="icon-box mt-5
93 " data-aos="zoom-in" data-aos-delay="150">
94 <i class="bx bx-cube-alt"></i>
95 <h4>Feature Engineering and Model Selection</h4>
96 <p>The second step involves selecting relevant features and transforming them into a format suitab
97 </div>
98 <div class="icon-box mt-5" data-aos="zoom-in" data-aos-delay="150">
99 <i class="bx bx-images"></i>
100 <h4>Model Training and Evaluation</h4>
101 <p>The third step involves training the selected model using the preprocessed data and evaluating
102 </div>
103 <div class="icon-box mt-5" data-aos="zoom-in" data-aos-delay="150">
104 <i class="bx bx-shield"></i>
105 <h4>Model Deployment</h4>
106 <p>The final step involves deploying the model in a real-world scenario to predict the customer be
107 </div>
108 </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

8. PERFORMANCE TESTING:

8.1 Performace Metrics:

Model Performance Testing:

1. **Metrics** **Regression Model:** MA E - , MSE - ,
RMSE - , R2 score -



Classification Model:

Confusion Matrix - , Accuray Score- & Classification Report

Screenshots: Regression Model MAE:

```
from sklearn.metrics import mean_absolute_error

mae = mean_absolute_error(y_actual, y_predicted)

print(f"Mean Absolute Error: {mae}")
```

Mean Absolute Error: 0.16

MSE:

```
[ ] #mean squared error
print(metrics.mean_squared_error(y_test,y_pred))
```

96015241.54707709

RMSE:

```
[ ] # RMSE (Root Mean Square Error)
    print(np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

```
9798.736732205693
```

R2 Score:

```
▶ from sklearn.metrics import r2_score
   r2_score(y_test,y_pred)
```

```
⇒ 0.988169515729126
```

Classification Model:

Confusion Matrix:

```
▶ confusion_matrix(y_test,pred)
```

```
⇒ array([[58,  0],
        [ 6, 16]])
```

Accuracy Score:

```
[ ] accuracy_score(y_test,pred)
```

```
0.925
```

Classification Report:

```
▶ print(classification_report(y_test,pred))
```

```
⇒
```

	precision	recall	f1-score	support
0	0.91	1.00	0.95	58
1	1.00	0.73	0.84	22
accuracy			0.93	80
macro avg	0.95	0.86	0.90	80
weighted avg	0.93	0.93	0.92	80

9. RESULTS:

9.1 Output Screenshots:

MARKET SEGMENTATION ANALYSIS USING ML.

Home About Contact Predict

Market Segmentation Analysis Using ML

yummy

convenient

spicy

fattening

greasy

fast

cheap

tasty

expensive

10. ADVANTAGES & DISADVANTAGES:

Precision: ML enhances segmentation accuracy by identifying nuanced patterns in data.

Efficiency: Automated processes reduce time and resources required for segmentation.

Personalization: Enables tailored marketing strategies for diverse customer segments.

ROI Improvement: Targeted campaigns lead to better returns on marketing investment.

Dynamic Adaptation: ML models can adapt to changing market dynamics.

Disadvantages:

Data Dependence: ML models rely heavily on the quality and diversity of input data.

Complexity: Implementing ML may require specialized skills and resources.

Interpretability: Some ML models lack transparency, making interpretation challenging.

Initial Costs: Setting up ML infrastructure may involve significant upfront investments.

Ethical Concerns: Bias in data or algorithms can lead to unintended consequences.

11. CONCLUSION:

We can conclude Machine Learning for market segmentation analysis offers unprecedented precision and efficiency. The enhanced accuracy in identifying customer segments, coupled

with the ability to adapt dynamically to market changes, positions the approach as a powerful tool for optimizing marketing strategies.

12. Future Scope:

The future scope for market segmentation analysis using Machine Learning holds promise for continuous evolution. Opportunities include exploring advanced algorithms for even finer segmentation, integrating real-time data for dynamic adaptation, and harnessing emerging technologies like artificial intelligence and predictive analytics.

13. APPENDIX

Source Code:

Import necessary libraries

from sklearn.cluster import KMeans

import pandas as pd

Load your dataset

data = pd.read_csv('your_dataset.csv')

Choose features for segmentation

features = data[['feature1', 'feature2', 'feature3']]

Specify the number of clusters (you can determine this based on analysis)

num_clusters = 3

Initialize the KMeans model

kmeans = KMeans(n_clusters=num_clusters, random_state=42)

Fit the model to the data

kmeans.fit(features)

Add the cluster labels to the original dataset


```
data['cluster_label'] = kmeans.labels_
```

```
# Print or visualize the results
```

```
print(data[['feature1', 'feature2', 'feature3', 'cluster_label']])
```

Project Demo Link

Demo link:

https://vitapacin.sharepoint.com/:f:/s/Internship849/EmOm8_JRp15FjawS5jX6p14BxKf3qgIc-xFgdFGJzKNHkw?e=zd777Z