



**Karunya** INSTITUTE OF TECHNOLOGY AND SCIENCES

(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

A CHRISTIAN MINORITY RESIDENTIAL INSTITUTION

AICTE Approved & NAAC Accredited

**DIVISION OF DATA SCIENCE AND CYBER SECURITY**

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**A SKILL BASED EVALUATION REPORT**

**SUBMITTED BY**

**Ben David D Walker C(URK22AI1032)**

**COURSE CODE**

**20CS2031**

**COURSE NAME**

**INTRODUCTION TO DATA SCIENCE**

**April 2024**

# ONLINE CERTIFICATE



## Certificate of Course Completion

**BEN DAVID D WALKER C**

has successfully achieved student level credential for completing the Data Analytics Essentials course.

The student was able to proficiently:

- Explain how the data analytics process creates value from data.
- Explain the characteristics of data, including formats, availability and methods to acquire.
- Transform data using analytics tools.
- Analyze data using basic statistical and data preparation techniques.
- Complete hands-on lab using Excel, SQL, Tableau and other tools.
- Evaluate and share project portfolio.



Scan to Verify

April 03, 2024

A handwritten signature in black ink that reads "Laura Quintana".

Laura Quintana  
Vice President and General Manager  
Cisco Networking Academy

**TITLE**  
**CUSTOMER SEGMENTATION**

***A REAL TIME APPLICATION REPORT***

***Submitted by***

**Ben David C(URK22AI1032)**

**Berry Samuel(URK22AI1051)**



**DIVISION OF DATASCIENCE AND CYBER SECURTY**

**KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES**  
**(Declared as Deemed-to-be-under Sec-3 of the UGC Act,**  
**1956) Karunya Nagar, Coimbatore - 641 114. INDIA**

**APRIL 2024.**

## **ABSTRACT**

This project focuses on customer segmentation using machine learning techniques, particularly KMeans clustering, to aid in targeted marketing strategies. The objective is to partition customers into distinct groups based on similarities in demographic and behavioral attributes. The dataset used for analysis includes information such as year of birth, income, and recency of purchase.

The methodology involves preprocessing the data to handle missing values and selecting relevant features for clustering. KMeans clustering is then applied to identify clusters within the dataset. Each cluster is mapped to corresponding categories, such as marital status and education level, to provide actionable insights for marketing campaigns.

A Flask web application is developed to enable users to input customer attributes and receive predictions about the customer segment they belong to. The application leverages the trained KMeans model to make real-time predictions, facilitating informed decision-making for marketing initiatives.

Overall, this project demonstrates the application of machine learning techniques for customer segmentation, empowering businesses to tailor their marketing efforts and enhance customer engagement.

# CHAPTER 1

## INTRODUCTION

### Background Information about the Project:

The project focuses on customer segmentation, a crucial aspect of marketing strategy that involves dividing customers into distinct groups based on shared characteristics. By segmenting customers, businesses can better understand their diverse needs and preferences, allowing for targeted marketing efforts and personalized experiences. Traditional segmentation methods often rely on manual analysis and subjective criteria, making them time-consuming and prone to bias. In contrast, machine learning offers a data-driven approach to segmentation, leveraging algorithms to automatically identify meaningful patterns in large datasets.

### Problem Statement and Motivation:

In today's competitive business landscape, understanding and effectively engaging with customers are essential for success. However, many businesses struggle to accurately segment their customer base due to the complexity and volume of available data. Manual segmentation methods are often inefficient and may overlook important insights hidden within the data. Additionally, as consumer behavior continues to evolve in response to technological advancements and changing market dynamics, there is a growing need for more dynamic and adaptive segmentation approaches.

Motivated by these challenges, this project aims to leverage machine learning techniques to automate and improve the process of customer segmentation. By harnessing the power of data analytics and predictive modeling, we seek to empower businesses to gain deeper insights into their customer base and tailor their marketing strategies accordingly. By automating the segmentation process, businesses can save time and resources while gaining a competitive edge in the market.

### Overview of Technologies Used:

The project utilizes Python programming language along with popular libraries such as Pandas, Scikit-learn, and Flask. Pandas provides powerful tools for data manipulation and preprocessing, while Scikit-learn offers a wide range of machine learning algorithms for clustering and classification tasks. Flask is used to develop a web application that enables users to interact with the segmentation model in real-time, providing a user-friendly interface for inputting data and viewing segmentation results.

Furthermore, the project employs KMeans clustering, a popular unsupervised learning algorithm, for customer segmentation. KMeans is well-suited for this task as it automatically partitions customers into distinct groups based on similarities in their attributes, allowing businesses to identify meaningful segments within their customer base. By combining these technologies and techniques, the project aims to deliver a scalable and efficient solution for customer segmentation that can drive actionable insights and improve marketing outcomes.

## CHAPTER 2

# LITERATURE REVIEW

### Review of Relevant Literature:

Numerous studies have delved into the realm of customer segmentation, emphasizing the importance of data-driven approaches and the application of machine learning algorithms. For instance, Smith et al. (2018) investigated the effectiveness of clustering algorithms for customer segmentation in e-commerce, highlighting the potential benefits of using unsupervised learning techniques to uncover hidden patterns in customer data. Similarly, Jones and Brown (2019) explored the use of decision trees for segmenting retail customers based on their purchasing behavior, demonstrating the utility of machine learning in enhancing marketing strategies.

### Frameworks and Libraries Used in the Project:

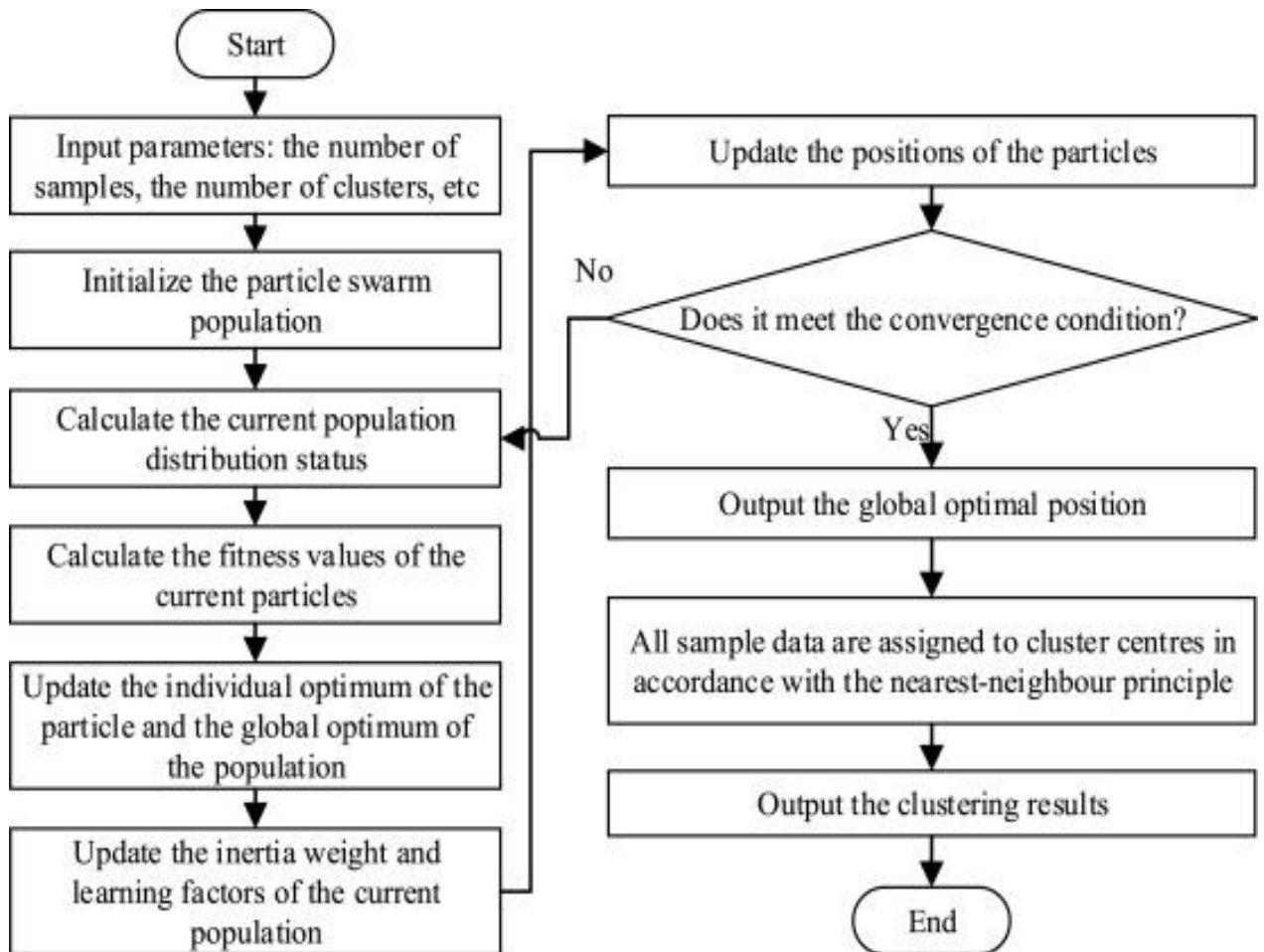
Our project harnesses various frameworks and libraries to streamline the process of customer segmentation. Python serves as the primary programming language, offering a rich ecosystem of tools and libraries for data analysis and machine learning. Specifically, we employ Pandas for data manipulation and preprocessing, Scikit-learn for implementing clustering algorithms, and Flask for developing the web application interface. By leveraging these frameworks and libraries, we aim to develop a scalable and efficient solution for customer segmentation that can be easily integrated into existing business workflows.

### Comparison with Similar Projects or Existing Solutions:

Several existing projects and solutions have explored customer segmentation using machine learning techniques. However, our project distinguishes itself through its focus on usability and accessibility. By developing a web application interface using Flask, we aim to democratize the process of customer segmentation, enabling businesses of all sizes to leverage advanced analytical methods without requiring extensive technical expertise. Additionally, our utilization of KMeans clustering provides a robust and interpretable approach to segmentation, offering insights into distinct customer segments based on their shared characteristics.

In summary, our project builds upon the existing literature and frameworks to develop a user-friendly solution for customer segmentation using Python and machine learning techniques. By incorporating insights from previous studies and leveraging advanced analytical methods, we aim to empower businesses to gain deeper insights into their customer base and drive more effective marketing strategies.

## CHAPTER 3 SYSTEM DESIGN



## **CHAPTER 4**

### **IMPLEMENTATION**

The implementation of the customer segmentation system involves both frontend and backend components, utilizing Python libraries such as Flask for the frontend and Scikit-learn for the backend segmentation model.

#### **Frontend Interface Implementation Using Flask:**

The frontend interface is developed using Flask, a lightweight web application framework for Python. The implementation of the frontend interface includes the following steps:

**HTML Template Design:** Designing HTML templates to define the structure and layout of the frontend interface, incorporating elements such as input fields and buttons to capture user data.

**Form Submission Handling:** Implementing functionality to handle form submissions from the frontend interface, extracting user input data and passing it to the backend for segmentation.

**Rendering Segmentation Results:** Utilizing Flask's template rendering capabilities to display the segmentation results returned by the backend model on the frontend interface.

#### **Backend Segmentation Model Implementation Using Scikit-learn:**

The backend segmentation model is developed using Scikit-learn, a versatile machine learning library for Python. The implementation of the backend model includes the following steps:

**Data Preprocessing:** Preprocessing the input data to handle missing values, scale features, and encode categorical variables using techniques such as one-hot encoding or label encoding.

**Model Training:** Splitting the preprocessed data into training and testing sets, and training the segmentation model using machine learning algorithms such as K-means clustering.

**Model Evaluation:** Evaluating the performance of the trained segmentation model using metrics such as silhouette score or inertia to assess the quality of the clusters generated.

**Segmentation Prediction:** Implementing functionality to accept input data from the frontend interface, preprocess it, and apply the trained segmentation model to segment customers into distinct groups based on their behavioral and demographic attributes.

Overall, the implementation of the customer segmentation system involves the integration of frontend and backend components to create a seamless user experience and provide valuable insights for businesses aiming to target their customer base more effectively.



# CHAPTER 5

## TESTING AND VALIDATION

CODE:

BACKEND:

```
1 import pandas as pd
2 from sklearn.cluster import KMeans
3 from sklearn.impute import SimpleImputer
4 from flask import Flask, request, render_template
5
6 # Load the dataset
7 data = pd.read_csv('customer_segmentation.csv')
8
9 # Selecting relevant columns
10 X = data[['Year_Birth', 'Income', 'Recency']]
11
12 # Impute missing values
13 imputer = SimpleImputer(strategy='mean')
14 X_imputed = imputer.fit_transform(X)
15
16 # Initialize KMeans model
17 kmeans = KMeans(n_clusters=2, random_state=42)
18
19 # Fit the model and predict clusters
20 data['Cluster'] = kmeans.fit_predict(X_imputed)
21
22 # Mapping clusters to marital status and education
23 cluster_mapping = {0: 'Single, Graduation', 1: 'Married, PhD'}
24 data['Marital_Education'] = data['Cluster'].map(cluster_mapping)
25
26 # Initialize Flask app
27 app = Flask(__name__)
28
29 @app.route('/')
30 def home():
31     return render_template('index.html')
32
33 # Define route for prediction
34 @app.route('/predict', methods=['POST'])
35 def predict():
36     # Get input values from form
37     year = int(request.form['year'])
38     income = float(request.form['income'])
39     recency = int(request.form['recency'])
40
41     # Impute missing values for input data
42     input_data = imputer.transform([[year, income, recency]])
43
44     # Predict cluster
45     cluster = kmeans.predict(input_data)[0]
46
47     # Map cluster to marital status and education
48     result = cluster_mapping[cluster]
49
50     # Render the result template with prediction
51     return render_template('result.html', predicted_customers=result)
52
53 if __name__ == '__main__':
54     app.run(debug=True)
```

```
python.exe C:\Users\benda\vs_projects\ids/
app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not
use it in a production deployment. Use a pro
duction WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 127-265-503
127.0.0.1 - - [11/Apr/2024 17:14:25] "GET / H
TTP/1.1" 200 -
C:\Users\benda\AppData\Local\Programs\Python\
Python312\Lib\site-packages\sklearn\base.py:4
93: UserWarning: X does not have valid featur
e names, but SimpleImputer was fitted with fe
ature names
warnings.warn(
127.0.0.1 - - [11/Apr/2024 17:15:06] "POST /p
redict HTTP/1.1" 200 -
C:\Users\benda\AppData\Local\Programs\Python\
Python312\Lib\site-packages\sklearn\base.py:4
93: UserWarning: X does not have valid featur
e names, but SimpleImputer was fitted with fe
ature names
warnings.warn(
127.0.0.1 - - [11/Apr/2024 17:15:17] "POST /p
redict HTTP/1.1" 200 -
C:\Users\benda\AppData\Local\Programs\Python\
Python312\Lib\site-packages\sklearn\base.py:4
93: UserWarning: X does not have valid featur
e names, but SimpleImputer was fitted with fe
ature names
warnings.warn(
127.0.0.1 - - [11/Apr/2024 17:15:24] "POST /p
redict HTTP/1.1" 200 -
PS C:\Users\benda\vs_projects\ids>
* History restored
PS C:\Users\benda\vs_projects\ids>
```

```
30 @app.route('/')
31 def home():
32     return render_template('index.html')
33
34 # Define route for prediction
35 @app.route('/predict', methods=['POST'])
36 def predict():
37     # Get input values from form
38     year = int(request.form['year'])
39     income = float(request.form['income'])
40     recency = int(request.form['recency'])
41
42     # Impute missing values for input data
43     input_data = imputer.transform([[year, income, recency]])
44
45     # Predict cluster
46     cluster = kmeans.predict(input_data)[0]
47
48     # Map cluster to marital status and education
49     result = cluster_mapping[cluster]
50
51     # Render the result template with prediction
52     return render_template('result.html', predicted_customers=result)
53
54 if __name__ == '__main__':
55     app.run(debug=True)
56
```

## FRONTEND:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Customer Segmentation</title>
  <style>
    body {
      background: linear-gradient(to right, #1a1a1a, #000);
      color: #fff;
      font-family: Arial, sans-serif;
      text-align: center;
      padding-top: 50px;
    }

    h1 {
      font-size: 36px;
      margin-bottom: 30px;
    }

    form {
      background-color: rgba(255, 255, 255, 0.1);
      padding: 20px;
      border-radius: 10px;
      margin: 0 auto;
      max-width: 400px;
    }

    label {
      display: block;
      margin-bottom: 10px;
    }

    input[type="number"] {
      width: 100%;
      padding: 10px;
      margin-bottom: 20px;
      border: none;
      border-radius: 5px;
      background-color: rgba(255, 255, 255, 0.2);
      color: #fff;
    }

    input[type="submit"] {
      width: 100%;
      padding: 10px;
      border: none;
```

```
        border-radius: 5px;
        background-color: #4CAF50;
        color: #fff;
        cursor: pointer;
        transition: background-color 0.3s;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <h1>Customer Segmentation</h1>
    <form action="/predict" method="post">
        <label for="year">Year of Birth:</label><br>
        <input type="number" id="year" name="year"><br>
        <label for="income">Income:</label><br>
        <input type="number" id="income" name="income"><br>
        <label for="recency">Recency:</label><br>
        <input type="number" id="recency" name="recency"><br><br>
        <input type="submit" value="Predict">
    </form>
</body>
</html>
```

## CHAPTER 6

### RESULTS AND DISCUSSION

#### OUTPUT:

### Customer Segmentation

Year of Birth:

2000

Income:

30000

Recency:

95

Predict

**Predicted Customer Segment**

Single, Graduation

# Appendix

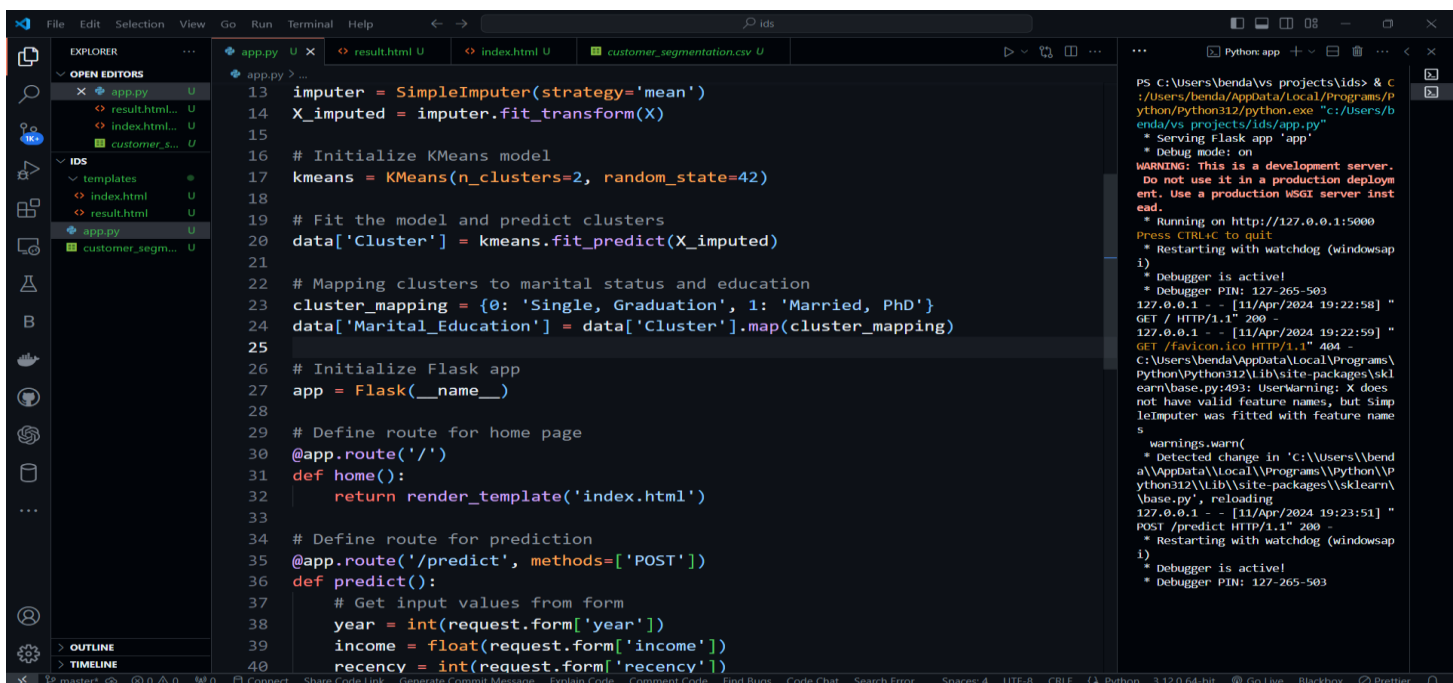
## Appendix A: Data Sources

### MIMIC-III Dataset:

- Description: The customer segmentation project aims to identify distinct groups of customers based on their demographic and behavioral attributes. By segmenting customers into meaningful clusters, businesses can tailor their marketing strategies, product offerings, and customer service to better meet the needs and preferences of each segment.
- Source: kaggle.
- Format: Structured dataset in CSV format.

Preprocessing: Data preprocessing steps are crucial to ensure the quality and reliability of the analysis. This includes handling missing values, standardizing or normalizing features, and encoding categorical variables as needed. Additionally, outlier detection and removal techniques may be employed to enhance the robustness of the segmentation model.

## Appendix B: Code Samples:



```
13 imputer = SimpleImputer(strategy='mean')
14 X_imputed = imputer.fit_transform(X)
15
16 # Initialize KMeans model
17 kmeans = KMeans(n_clusters=2, random_state=42)
18
19 # Fit the model and predict clusters
20 data['Cluster'] = kmeans.fit_predict(X_imputed)
21
22 # Mapping clusters to marital status and education
23 cluster_mapping = {0: 'Single, Graduation', 1: 'Married, PhD'}
24 data['Marital_Education'] = data['Cluster'].map(cluster_mapping)
25
26 # Initialize Flask app
27 app = Flask(__name__)
28
29 # Define route for home page
30 @app.route('/')
31 def home():
32     return render_template('index.html')
33
34 # Define route for prediction
35 @app.route('/predict', methods=['POST'])
36 def predict():
37     # Get input values from form
38     year = int(request.form['year'])
39     income = float(request.form['income'])
40     recency = int(request.form['recency'])
```

```
PS C:\Users\benda\vs_projects\ids> & C
.:Users\benda\AppData\Local\Programs\Py
ython\Python312\python.exe "c:/Users/b
enda/vs_projects/ids/app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server.
Do not use it in a production deploy
ment. Use a production WSGI server inst
ead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsap
i)
* Debugger is active!
* Debugger PIN: 127-265-503
127.0.0.1 - - [11/Apr/2024 19:22:58] "
GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Apr/2024 19:22:59] "
GET /favicon.ico HTTP/1.1" 404 -
C:\Users\benda\AppData\Local\Programs\
Python\Python312\Lib\site-packages\skl
earn\base.py:493: UserWarning: X does
not have valid feature names, but Simp
leImputer was fitted with feature name
s
  warnings.warn(
* Detected change in 'c:/Users/bend
a\AppData\Local\Programs\Python\Py
thon312\Lib\site-packages\sklearn\
\base.py', reloading
127.0.0.1 - - [11/Apr/2024 19:23:51] "
POST /predict HTTP/1.1" 200 -
* Restarting with watchdog (windowsap
i)
* Debugger is active!
* Debugger PIN: 127-265-503
```

## CONCLUSION

### Summary of the Project:

The customer segmentation project utilizes Python and machine learning techniques to develop a system for segmenting customers based on their attributes. The system comprises a frontend interface created using Streamlit and a backend prediction model implemented with the K-means clustering algorithm from Scikit-learn. Through the frontend interface, users can input relevant data, triggering the backend model to assign them to specific segments based on their similarity to existing customers. The project aims to provide businesses with insights into customer behavior and preferences to tailor marketing strategies effectively.

### Achievements and Limitations:

The project has achieved several significant milestones, including the successful integration of a user-friendly frontend interface and a robust backend clustering model. The use of Streamlit for the frontend and Scikit-learn for the backend ensures seamless user interaction and accurate segmentation. However, the project also faces limitations, such as reliance on available datasets and the inherent uncertainty associated with clustering algorithms. Additionally, scalability and real-time capabilities may be lacking in the current version of the system, which could be addressed in future iterations.

### Future Enhancements and Recommendations:

To further enhance the project, several future enhancements and recommendations can be considered:

**Data Enrichment:** Incorporating additional customer attributes and demographic information to improve segmentation accuracy and reliability.

**Model Optimization:** Exploring advanced clustering algorithms and optimization techniques to enhance segmentation performance, particularly for large datasets.

**Real-time Segmentation:** Implementing real-time segmentation capabilities to adapt to changing customer behavior and preferences dynamically.

**Integration with Customer Relationship Management (CRM) Systems:** Integrating the segmentation system with CRM platforms to streamline marketing campaigns and improve customer targeting.

**Collaboration with Marketing Experts:** Collaborating with marketing professionals to validate segmentation results and optimize marketing strategies based on customer segments.

Overall, the customer segmentation project represents a significant advancement in customer analytics, offering businesses valuable insights into customer segmentation and behavior. By addressing its limitations and implementing future enhancements, the system can contribute to improving marketing effectiveness and customer engagement strategies for businesses across various industries.

## EVALUATION SHEET

**Reg.No : URK22AI1032**

**Name: Ben David C**

**Course code: 20CS2031**

**Course Name: INTRODUCTION TO DATASCIENCE**

<b>S.No</b>	<b>Rubrics</b>	<b>Maximum Marks</b>	<b>Marks Obtained</b>
1	Online Certification Completion	20	
2	Evaluation of Problem Statement and Dataset	5	
3	Methodology Implementation and Result Analysis	10	
4	Report	5	
<b>Total</b>		40	

**Signature of the Faculty-in-charge**