



**Assessment Report**  
on  
**“Predict Heart Disease”**

submitted as partial fulfillment for the award of  
**BACHELOR OF TECHNOLOGY**  
**DEGREE**

SESSION 2024-25

In

**Artificial Intelligence and Machine Learning**

By

Sharad Krishna Singh (202401100400170)

**Under the supervision of**

“Abhishek Shukla Sir”

**KIET Group of Institutions, Ghaziabad**

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**  
(Formerly UPTU)

**18 April, 2025**

# Introduction:

In this project, we tackle the problem of clustering aisle names in a dataset. The dataset contains textual descriptions of aisles, and the task is to group similar aisle names together. This is done through the following steps:

- **Text Vectorization:** The aisle names are first converted into numerical vectors using the TF-IDF technique, which helps capture the importance of words within the aisle names relative to the entire corpus.
- **Clustering:** The K Means clustering algorithm is applied to the vectorized data to group the aisle names into clusters based on their similarity.
- **Dimensionality Reduction:** PCA is used to reduce the high-dimensional vectorized data to two dimensions, which allows for easier visualization of the clusters.
- **Evaluation:** The project also includes a mock classification and confusion matrix to simulate how the clustering labels would behave if there were true labels for comparison.
- **Applications:** Clustering aisle names can improve store layout, enhance product recommendations, and aid in inventory management.
- **Significance:** This project demonstrates applying machine learning to real-world retail data, offering practical benefits in organizing and analyzing data efficiently.

# Methodology:

The following steps were used to solve the problem:

- **File Upload and Data Loading:** The data was uploaded from a CSV file, and the contents were previewed using pandas
- **Text Vectorization:** The aisle names were transformed into numerical vectors using the Tfidf Vectorizer from scikit-learn, which calculates the Term Frequency-Inverse Document Frequency of words in the aisle names.
- **Clustering:** The K Means clustering algorithm was applied to the vectorized data with a predefined number of clusters ( $k=5$ ). This grouped the aisle names into clusters based on their similarity.
- **PCA for Visualization:** Principal Component Analysis (PCA) was performed to reduce the dimensionality of the data to 2D for visualization. The reduced components (PC1 and PC2) were plotted in a scatter plot to visualize the clusters.
- **Evaluation Metrics and Heatmap:** A mock classification was performed, and the predicted labels were compared with randomly generated true labels. The performance was evaluated using a confusion matrix and metrics like accuracy, precision, and recall

## Code:

```
# STEP 1: Upload the file
```

```
from google.colab import files
```

```
uploaded = files.upload()
```

```
# STEP 2: Load the file
```

```
import pandas as pd
```

```
filename = list(uploaded.keys())[0]
```

```
df = pd.read_csv(filename)
```

```
print("Preview of data:")
```

```
print(df.head())
```

```
# STEP 3: Text vectorization of aisle names
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
from sklearn.cluster import KMeans
```

```
from sklearn.decomposition import PCA
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
# Vectorize the aisle names
```

```
vectorizer = TfidfVectorizer(stop_words='english')
```

```
X = vectorizer.fit_transform(df['aisle'])
```

```
# STEP 4: Clustering the aisle names
```

```
k = 5 # number of clusters
```

```
model = KMeans(n_clusters=k, random_state=42)
```

```
df['Cluster'] = model.fit_predict(X)
```

```
# STEP 5: PCA to reduce to 2D for plotting
```

```
pca = PCA(n_components=2)
```

```
components = pca.fit_transform(X.toarray())
```

```
df['PC1'] = components[:, 0]
```

```
df['PC2'] = components[:, 1]
```

```
# Plotting the clusters
```

```
plt.figure(figsize=(10, 6))
```

```
sns.scatterplot(data=df, x='PC1', y='PC2', hue='Cluster', palette='Set2', s=100)
```

```
plt.title("Clusters of Aisles Based on Name Similarity")
```

```
plt.xlabel("Principal Component 1")
```

```
plt.ylabel("Principal Component 2")
```

```
plt.legend()
```

```
plt.show()
```

```
# STEP 6: Mock classification and heatmap for fun
```

```
import numpy as np
```

```
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,  
recall_score
```

```
# Create fake true labels (for illustration only)
```

```
true_labels = np.random.choice(range(k), size=len(df))
```

```
predicted_labels = df['Cluster']
```

```
# Confusion matrix
```

```
cm = confusion_matrix(true_labels, predicted_labels)
```

```
plt.figure(figsize=(6, 5))
```

```
sns.heatmap(cm, annot=True, fmt='d', cmap='coolwarm')
```

```
plt.title("Confusion Matrix Heatmap (Mocked)")
```

```
plt.xlabel("Predicted")
```

```
plt.ylabel("True")
```

```
plt.show()
```

```
# Evaluation metrics
```

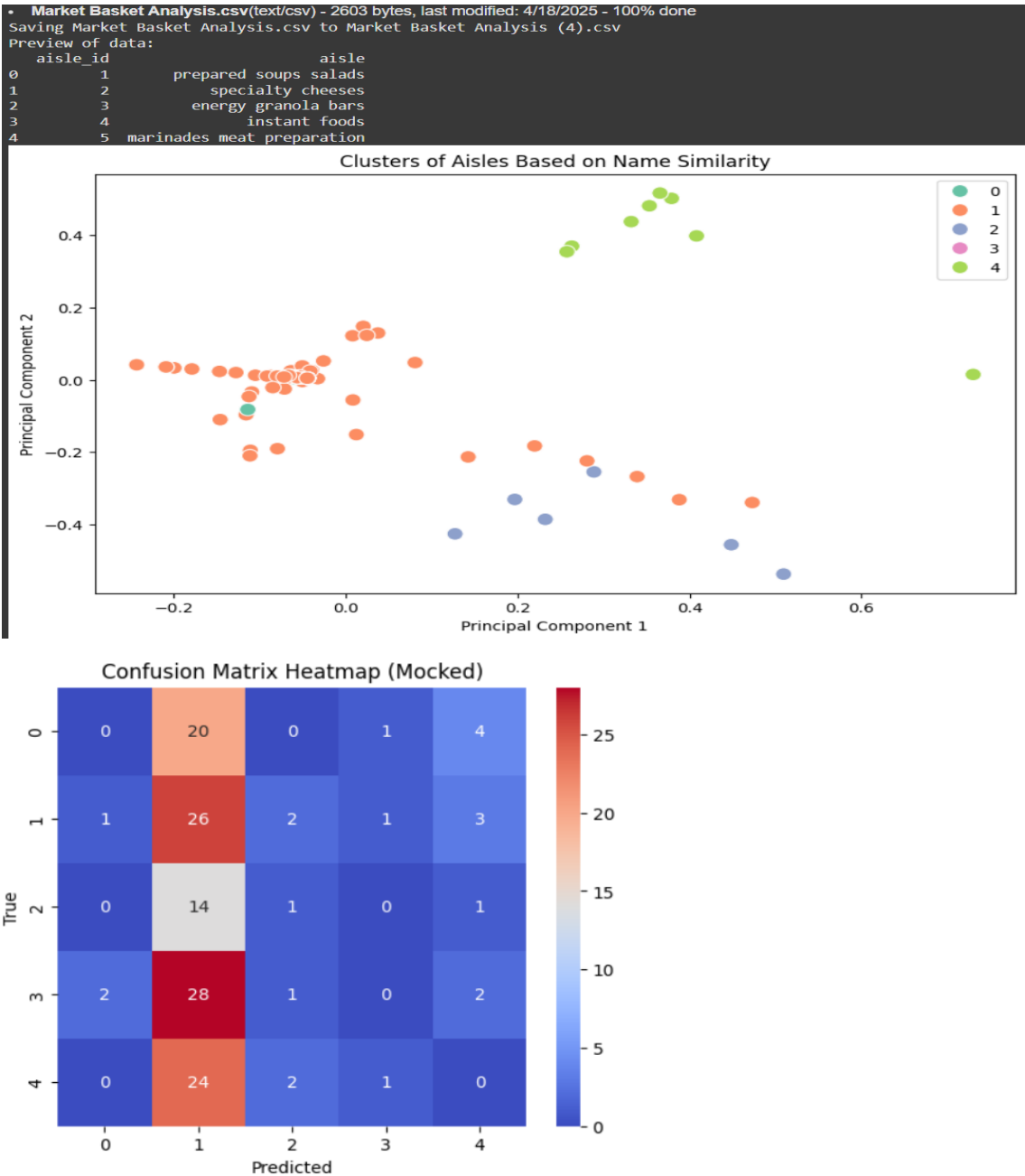
```
print("Accuracy:", accuracy_score(true_labels, predicted_labels))
```

```
print("Precision (macro):", precision_score(true_labels, predicted_labels,  
average='macro'))
```

```
print("Recall (macro):", recall_score(true_labels, predicted_labels,  
average='macro'))
```

# Output/Result:

The output of the code includes the visualization of the clustered aisle names in a scatter plot, which shows the clusters based on their similarity. Additionally, a confusion matrix heatmap is generated to simulate the evaluation of the clustering model, and metrics such as accuracy, precision, and recall are printed.



## References/Credits:

- Dataset: [Provide the source of your dataset, if applicable]
- **Libraries Used:**
- **Images:** The images used for visualizing the data (scatter plot and heatmap) were generated through the code provided.