

# Data Visualization Techniques Lab

Pulla Manoj Kumar

Assistant Professor, CSD, ACEEC Hyderabad

## Financial analysis using Clustering, Histogram and HeatMap

```
!pip install pywaffle==0.6.4
```

### STEP 1: Import Required Libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score

from yellowbrick.cluster import KElbowVisualizer
from pywaffle import Waffle

warnings.filterwarnings('ignore')
sns.set(style="whitegrid")
```

### STEP 2: Load the Dataset

```
df = pd.read_csv('financial_analysis.csv')
df.head()
```

## STEP 3: Feature Engineering

```
df['TOTAL_PAY'] = df[['PAY_AMT1','PAY_AMT2','PAY_AMT3',
                      'PAY_AMT4','PAY_AMT5','PAY_AMT6']].sum(axis=1)

df['TOTAL_BILL'] = df[['BILL_AMT1','BILL_AMT2','BILL_AMT3',
                      'BILL_AMT4','BILL_AMT5','BILL_AMT6']].sum(axis=1)

df.drop(columns=[

    'ID',
    'PAY_AMT1','PAY_AMT2','PAY_AMT3','PAY_AMT4','PAY_AMT5','PAY_AMT6',
    'BILL_AMT1','BILL_AMT2','BILL_AMT3','BILL_AMT4','BILL_AMT5','BILL_AMT6'
], inplace=True)
```

## STEP 4: Categorical Attribute Analysis

```
print("SEX Distribution:\n", df['SEX'].value_counts())
print("EDUCATION Distribution:\n", df['EDUCATION'].value_counts())
```

## STEP 5: Data Cleaning

```
df = df[~df['EDUCATION'].isin([5, 6])]
df.reset_index(drop=True, inplace=True)
```

## STEP 6: Correlation Heatmap

```
plt.figure(figsize=(10,8))
sns.heatmap(df.corr(), cmap='coolwarm')
plt.title("Correlation HeatMap of Financial Attributes")
plt.show()
```

## STEP 7: Data Scaling

```
num_df = df.select_dtypes(include=np.number)

scaler = StandardScaler()
scaled_data = scaler.fit_transform(num_df)
```

## STEP 8: Principal Component Analysis (PCA)

```
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)

X = pd.DataFrame(pca_data, columns=['PC1', 'PC2'])
print("PCA Data Shape:", X.shape)
```

## STEP 9: Histogram Analysis

```
df.hist(figsize=(15,15))
plt.suptitle("Histogram Analysis of Financial Data")
plt.show()
```

## STEP 10: Optimal Cluster Selection (Elbow Method)

```
elbow = KElbowVisualizer(KMeans(random_state=42), k=(2,10))
elbow.fit(X)
elbow.show()
```

## STEP 11: K-Means Clustering

```
kmeans = KMeans(n_clusters=4, random_state=42, max_iter=500)
y_kmeans = kmeans.fit_predict(X)
```

## STEP 12: PCA Clustered Data

```
pca_df = X.copy()
pca_df['Cluster'] = y_kmeans
print(pca_df.head())
```

## STEP 13: Cluster Distribution – Waffle Chart

```
unique, counts = np.unique(y_kmeans, return_counts=True)
total_customers = sum(counts)

percentages = (counts / total_customers) * 100

data_for_waffle = {
    f"Cluster {i+1}": round(p, 1)
    for i, p in zip(unique, percentages)
}

fig = plt.figure(
    FigureClass=Waffle,
    rows=10,
    columns=10,
    values=data_for_waffle,
    colors=['#FFBB00', '#3C096C', '#9D4EDD', '#FFE270'],
    icons='user',
    icon_size=14,
    legend={
        'labels': [
            f"Cluster {i+1} ({round(p,1)}%)"
            for i, p in zip(unique, percentages)
        ],
        'loc': 'upper left',
        'bbox_to_anchor': (1, 1),
        'frameon': False
    },
    figsize=(10, 5)
)

plt.title(
    "Financial Analysis: Customer Segmentation Distribution",
    loc='left',
    weight='bold'
)
plt.show()
```

## STEP 14: Cluster Evaluation Metrics

```
print("Silhouette Score:", silhouette_score(X, y_kmeans))
print("Davies Bouldin Score:", davies_bouldin_score(X, y_kmeans))
```

## STEP 15: Cluster-wise Analysis

```
cluster_df = df.copy()
cluster_df['Cluster'] = y_kmeans

print(cluster_df.groupby('Cluster').mean())
```

## STEP 16: Cluster Count Visualization

```
sns.countplot(x='Cluster', data=cluster_df)
plt.title("Cluster Count Distribution")
plt.show()
```

## STEP 17: Feature-wise Cluster Distribution

```
for col in cluster_df.drop('Cluster', axis=1).columns:
    g = sns.FacetGrid(cluster_df, col='Cluster')
    g.map_dataframe(sns.histplot, col)
    g.fig.suptitle(f"Distribution of {col}", y=1.05)
    plt.show()
```

## Understanding Clusters

- **Cluster 0:** Moderate credit limits and regular payment behavior.
- **Cluster 1:** High credit limits, high spending, low default risk.
- **Cluster 2:** Low credit limits, delayed payments, high default risk.
- **Cluster 3:** Controlled spending with consistent timely payments.