HW5_Part2_Henry_Romero

April 6, 2025

[18]:

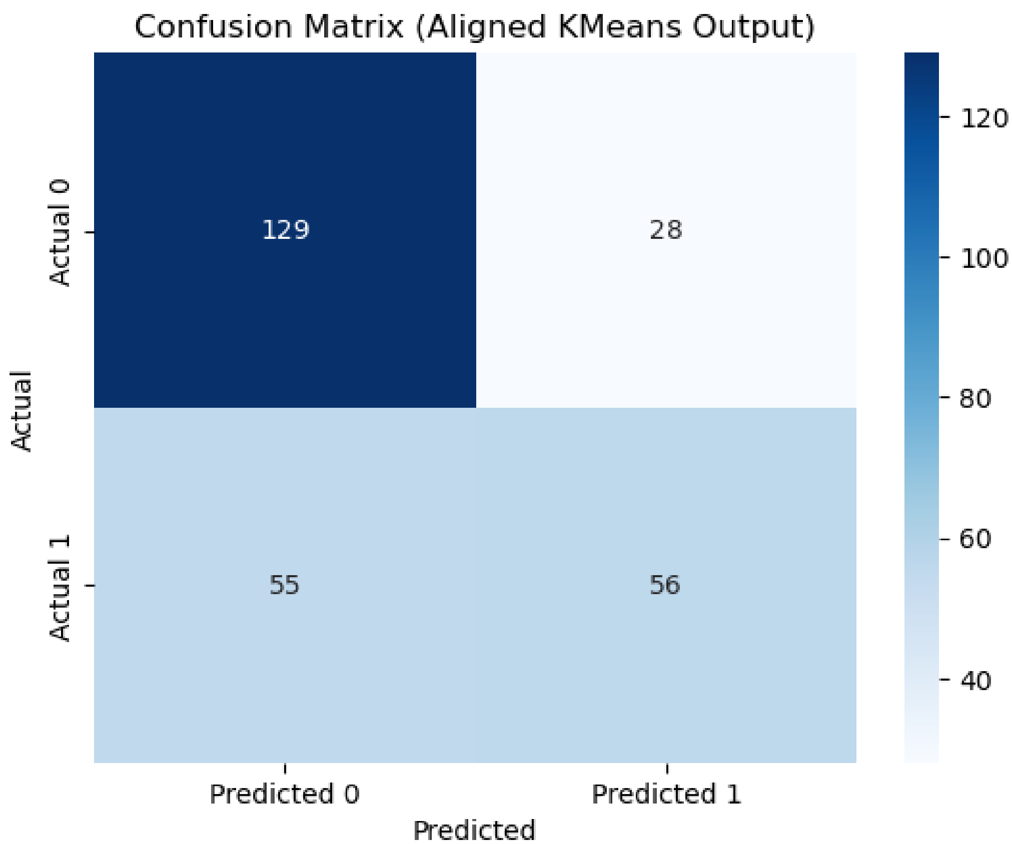
```
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import mode
# I continued the Titanic dataset route and selected the data from the cluster␣
↪dataset folder
# This accuracy will be dependent on PSA transformed data on an unsupervised␣
↪algorithm

# Load the dataset
df = pd.read_csv('Titanic-Dataset.csv')

# View column names and basic info
print("Columns:\n", df.columns)
print("\nMissing Values:\n", df.isnull().sum())
print("Rows",len(df))
print(df.describe())

# Feature selection
features = ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']
target = 'Survived' if 'Survived' in df.columns else 'survived'  # Just in case␣
↪casing differs in data

# preprocessing
# missing values
df['Age'] = SimpleImputer(strategy='mean').fit_transform(df[['Age']])
df['Embarked'] = SimpleImputer(strategy='most_frequent').
↪fit_transform(df[['Embarked']]).ravel()

# Encode categoricals
```

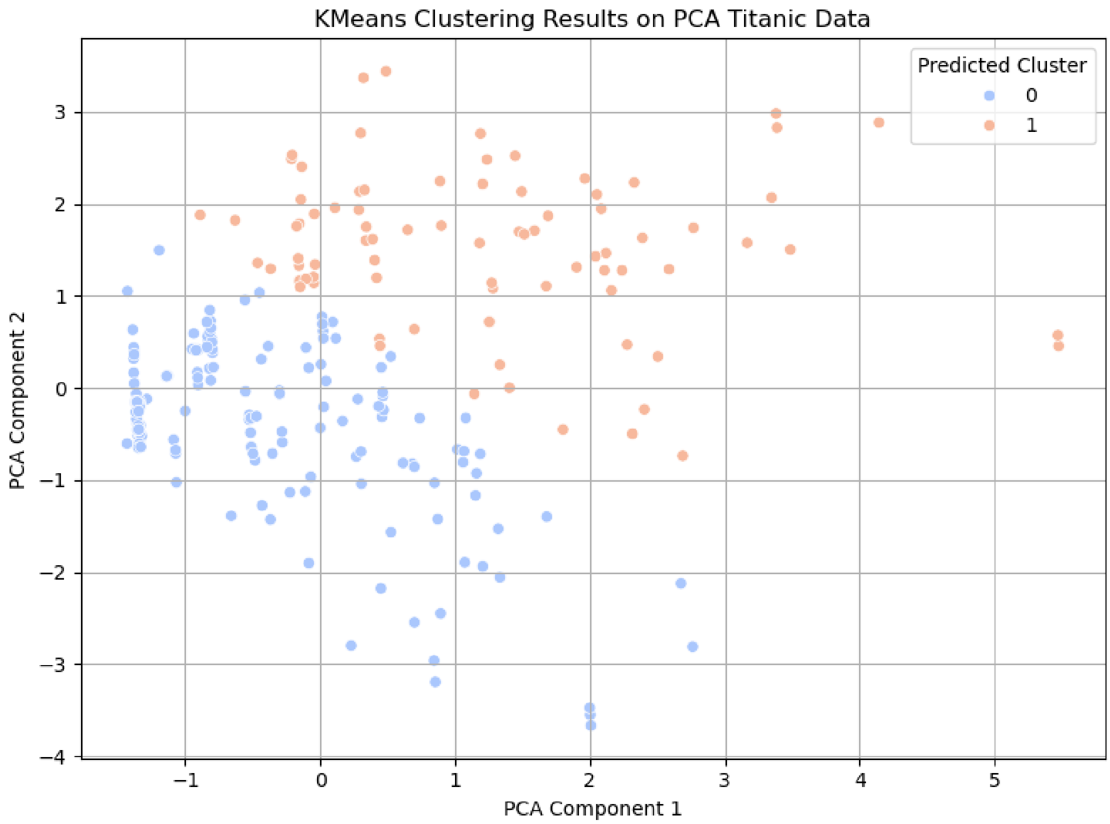
```
df['Sex'] = LabelEncoder().fit_transform(df['Sex'])
df['Embarked'] = LabelEncoder().fit_transform(df['Embarked'])

# Prepare features and labels
X = df[features]
y = df[target]

# Train/Test split with 30% on state 42
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
 ↪random_state=42)

# PCA on Training
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Initialize PCA
pca = PCA(n_components=2) #Used 2 components for PCA
X_train_pca = pca.fit_transform(X_train_scaled)
X_test_pca = pca.transform(X_test_scaled)

# KMeans clustering
kmeans = KMeans(n_clusters=2, random_state=42) # 2 clusters on state 42
kmeans.fit(X_train_pca)
y_train_pred = kmeans.predict(X_train_pca)
y_test_pred = kmeans.predict(X_test_pca)

# Flip cluster labels
# Align KMeans output with survived labels

def align_clusters(y_true, y_pred):
    labels = np.zeros_like(y_pred)
    for cluster in np.unique(y_pred):
        mask = (y_pred == cluster)
        labels[mask] = mode(y_true[mask])[0]
    return labels

y_test_aligned = align_clusters(y_test.values, y_test_pred)

# Evaluation and confusion matrix added with a plot
# High accuracy score of 69% for the PCA transformed data

accuracy = accuracy_score(y_test, y_test_aligned)
print("\nKMeans Accuracy after alignment:", accuracy)

cm = confusion_matrix(y_test, y_test_aligned)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
```

```python
            xticklabels=['Predicted 0', 'Predicted 1'],
            yticklabels=['Actual 0', 'Actual 1'])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix (Aligned KMeans Output)")
plt.show()

# Plotting Clusters with scatterplot
# Plotted graph has 2 distinct clusters near component 1 and component 2
plt.figure(figsize=(8, 6))
sns.scatterplot(x=X_test_pca[:, 0], y=X_test_pca[:, 1], hue=y_test_aligned,␣
 ↪palette='coolwarm')
plt.title('KMeans Clustering Results on PCA Titanic Data')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.legend(title='Predicted Cluster')
plt.grid(True)
plt.tight_layout()
plt.show()

print("The kmeans clustering on PCA data gives an accuracy of 69% with proper␣
 ↪class distribution as shown in the scatt")

# KMeans clusters to actual survival on the data, a comparison of the accuracy␣
 ↪shown before.
kmeans_vs_true = pd.DataFrame({
    'KMeans_Cluster': kmeans.labels_,
    'Actual_Survived': y_train.values
})
print("\nKMeans Cluster vs. Actual Survival:\n")
print(kmeans_vs_true.head(5))
print(pd.crosstab(kmeans_vs_true['KMeans_Cluster'],␣
 ↪kmeans_vs_true['Actual_Survived']))
```

```
Columns:
 Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')

Missing Values:
 PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
```

```
Parch             0
Ticket            0
Fare              0
Cabin           687
Embarked          2
dtype: int64
Rows 891
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

            Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200

KMeans Accuracy after alignment: 0.6902985074626866
```


Confusion Matrix (Aligned KMeans Output)
Actual 0
Actual 1
Actual
129
28
55
56
Predicted 0
Predicted 1
Predicted
120
100
80
60
40



The kmeans clustering on PCA data gives an accuracy of 69% with proper class distribution as shown in the scatt

```
KMeans Cluster vs. Actual Survival:

   KMeans_Cluster  Actual_Survived
0               1                1
1               0                0
2               0                1
3               0                0
4               0                0
Actual_Survived    0    1
KMeans_Cluster
0                330  134
1                 62   97
```