Submitted By **Junaid Asif / M.Khizar Baig**

Roll No **BSAI-144 / BSAI-156**

Class **BSAI - 3rd Sem**

Date **Dec 31, 2023**

Submitted To **Ms. Humaira Batool**

Final Project - Lab Report

Code:

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Load your dataset
file_path = r"E:\Sem3-GH\Sem3Lab\AI\final_project\AI-FP\Video_Games.csv"
df = pd.read_csv(file_path)
# Check the dataset and handle missing values if any
print(df.head()) # Check the structure of the dataset
print(df.isnull().sum()) # Check for missing values
# Handling missing values - replace 'tbd' in 'User_Score' with NaN
df['User_Score'] = pd.to_numeric(df['User_Score'], errors='coerce')
# Assuming 'Genre' is the target variable
x = df[['Critic_Score', 'Critic_Count', 'User_Score', 'User_Count']]
y = df['Genre']
```

```
# Drop rows with NaN values in the features
x.dropna(inplace=True)
y = y.iloc[x.index] # Update y based on the filtered x indices
# Applying KNN
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=42)
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train, y_train)
predictions = knn.predict(x_test)
accuracy = accuracy_score(y_test, predictions)
print(f"KNN on data accuracy: {accuracy}")
# Applying KMeans clustering
numeric_columns = ['Critic_Score', 'Critic_Count', 'User_Score', 'User_Count']
x = df[numeric_columns].dropna() # Drop rows with NaN values
# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(x)
# KMeans clustering
kmeans = KMeans(n_clusters=3, n_init=10, random_state=42)
kmeans.fit(X_scaled)
# Plotting clusters
cluster_labels = kmeans.labels_
cluster_centers = kmeans.cluster_centers_
plt.figure(figsize=(8, 6))
plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=cluster_labels, cmap='viridis', edgecolor='k',
label='Data Points')
plt.scatter(cluster_centers[:, 0], cluster_centers[:, 1], marker='*', s=100, c='red',
label='Cluster Centers')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
```

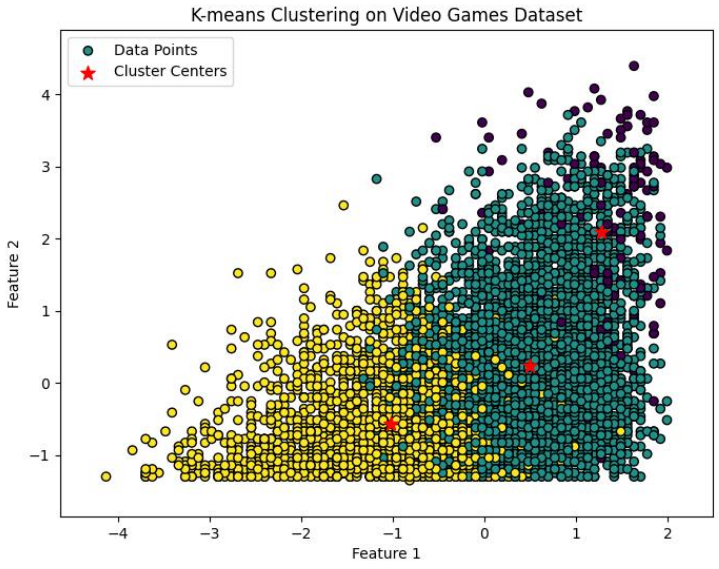
```
# Adjust x-axis and y-axis limits to show more data points
plt.xlim(X_scaled[:, 0].min() - 0.5, X_scaled[:, 0].max() + 0.5)
plt.ylim(X_scaled[:, 1].min() - 0.5, X_scaled[:, 1].max() + 0.5)
plt.title('K-means Clustering on Video Games Dataset')
plt.legend()
plt.show()
```

Output:

```
PS E:\Sem3-GH\Sem3Lab\AI\final_project> & C:/Users/ja138/AppData/Local/Microsoft/WindowsApps/python3.11.exe e:/Sem3-GH/Sem3Lab/AI/final_project/main.
py
   index                    Name Platform  Year_of_Release         Genre  ... Critic_Count  User_Score  User_Count  Developer  Rating
0      0              Wii Sports      Wii           2006.0        Sports  ...         51.0           8       322.0   Nintendo       E
1      1        Super Mario Bros.      NES           1985.0      Platform  ...          NaN         NaN         NaN        NaN     NaN
2      2           Mario Kart Wii      Wii           2008.0        Racing  ...         73.0         8.3       709.0   Nintendo       E
3      3        Wii Sports Resort      Wii           2009.0        Sports  ...         73.0           8       192.0   Nintendo       E
4      4  Pokemon Red/Pokemon Blue       GB           1996.0  Role-Playing  ...          NaN         NaN         NaN        NaN     NaN

[5 rows x 17 columns]
index                0
Name                 2
Platform             0
Year_of_Release    273
Genre                2
Publisher           55
NA_Sales             0
EU_Sales             0
JP_Sales             0
Other_Sales          0
Global_Sales         0
Critic_Score      8668
Critic_Count      8668
User_Score        6769
User_Count        9210
Developer         6688
Rating            6836
dtype: int64
e:\Sem3-GH\Sem3Lab\AI\final_project\main.py:20: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  x.dropna(inplace=True)
KNN on data accuracy: 0.2158374211632796
PS E:\Sem3-GH\Sem3Lab\AI\final_project>
```



Functionality:

This Python code performs a few machine learning tasks using the scikit-learn library on a dataset of video games.

Data Loading and Preprocessing:

- It starts by loading a dataset from a CSV file using Pandas.
- It inspects the dataset by printing the first few rows and checking for missing values (NaN) using isnull().sum().

Handling Missing Values:

- It handles missing values in the 'User_Score' column by converting 'tbd' values to NaN using pd.to_numeric().

Supervised Learning - K-Nearest Neighbors (KNN):

- It prepares the features (x) and the target variable (y) for KNN, using columns 'Critic_Score', 'Critic_Count', 'User_Score', and 'User_Count'.
- Rows with missing values in features are dropped, and the target variable is updated accordingly.
- Splits the data into training and testing sets (x_train, x_test, y_train, y_test) using train_test_split().
- Trains a KNN classifier with 3 neighbors (n_neighbors=3) using KNeighborsClassifier().
- Evaluates the model's accuracy on the test set using accuracy_score().

Unsupervised Learning - KMeans Clustering:

- It focuses on columns 'Critic_Score', 'Critic_Count', 'User_Score', and 'User_Count' for clustering.
- Drops rows with missing values in these columns and standardizes the data using StandardScaler().
- Performs KMeans clustering with 3 clusters (n_clusters=3) on the standardized data.

Visualization:

- Plots the clusters formed by KMeans on a 2D scatter plot.
- Data points are colored based on their assigned cluster, while cluster centers are marked with red asterisks.
- Axes labels, title, and legend are added for better understanding.
- The plot is displayed using Matplotlib.

The code essentially handles missing values, applies two different machine learning techniques—supervised (KNN for classification) and unsupervised (KMeans for clustering), and provides a visual representation of the clusters formed by KMeans in a 2D space.