

20MSSL12 - MACHINE LEARNING LABORATORY - CAT 2 PROJECT

Date: 07-12-2023

Project title: Transaction Anomaly Detection.

Concept used: K-Means clustering.

Team members: Boopal.N – 71762131010

Aim:

The problem addressed by the code is transaction anomaly detection. The goal is to identify whether a given transaction is normal or an anomaly based on certain transaction details. This is a common problem in fraud detection, where anomalies could indicate potentially fraudulent activities.

Source code:

```
import pandas as pd
import streamlit as st

import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

data = pd.read_csv("transaction_anomalies_dataset.csv")
print(data)
relevant_features = ['Transaction_Amount', 'Average_Transaction_Amount',
'Frequency_of_Transactions']

st.title("Transaction Anomaly Detection ")
st.header("Enter Transaction Details:")

user_inputs = {}
for feature in relevant_features:
    user_input = st.number_input(f"{feature}: ", min_value=0.0)
    user_inputs[feature] = user_input

k = 2

scaler = StandardScaler()
X_scaled = scaler.fit_transform(data[relevant_features])
kmeans = KMeans(n_clusters=k, random_state=42)
data['Cluster'] = kmeans.fit_predict(X_scaled)

if st.button("Predict Anomaly"):
```

```

user_df = pd.DataFrame([user_inputs])
user_inputs_scaled = scaler.transform(user_df)
user_cluster = kmeans.predict(user_inputs_scaled)

st.write(f"Predicted Cluster: {user_cluster[0]}")

if user_cluster[0] == 1:
    st.warning("Anomaly detected: This transaction is flagged as an anomaly.")
else:
    st.success("No anomaly detected: This transaction is normal.")

fig, ax = plt.subplots()
for cluster_label in range(k):
    cluster_data = data[data['Cluster'] == cluster_label]
    ax.scatter(cluster_data['Transaction_Amount'],
               cluster_data['Average_Transaction_Amount'], label=f'Cluster {cluster_label}')
    centroids = scaler.inverse_transform(kmeans.cluster_centers_)
    ax.scatter(centroids[:, 0], centroids[:, 1], marker='X', color='red', s=200,
               label='Centroids')
ax.set_title('K-Means Clustering')
ax.set_xlabel('Transaction Amount')
ax.set_ylabel('Average Transaction Amount')
ax.legend()

st.pyplot(fig)
print(data)

```

Output:

PS D:\machinelearning\cat2> streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>

Network URL: <http://192.168.136.238:8501>

Transaction_ID Transaction_Amount Transaction_Volume Average_Transaction_Amount
Frequency_of_Transactions ... Time_of_Day Age Gender Income Account_Type

[1000 rows x 12 columns]

Transaction_ID Transaction_Amount Transaction_Volume Average_Transaction_Amount
Frequency_of_Transactions ... Age Gender Income Account_Type Cluster

Female 1101680 Current 0

[1000 rows x 13 columns]

Transaction Anomaly Detection App

Enter Transaction Details:

Transaction_Amount:

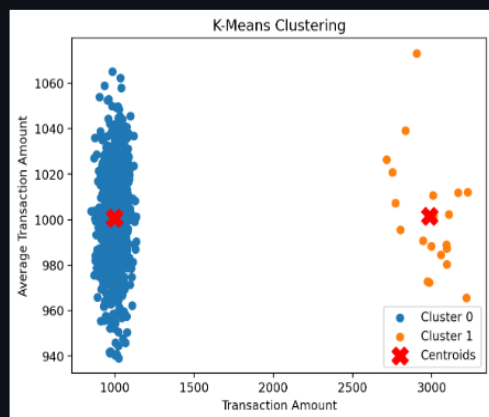
Average_Transaction_Amount:

Frequency_of_Transactions:

Predict Anomaly

Predicted Cluster: 0

No anomaly detected: This transaction is normal.



Transaction Anomaly Detection App

Enter Transaction Details:

Transaction_Amount:

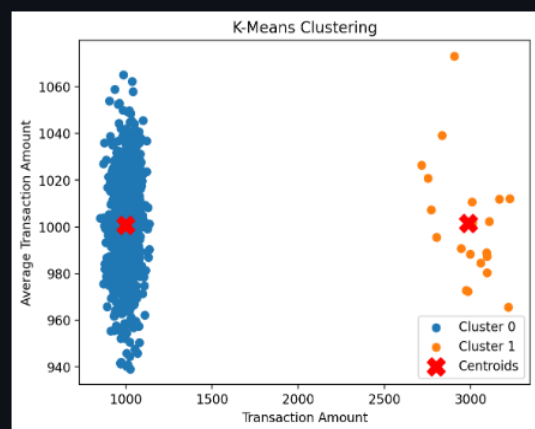
Average_Transaction_Amount:

Frequency_of_Transactions:

Predict Anomaly

Predicted Cluster: 1

Anomaly detected: This transaction is flagged as an anomaly.



Transaction Anomaly Detection App

Enter Transaction Details:

Transaction_Amount:

Average_Transaction_Amount:

Frequency_of_Transactions:

Predict Anomaly