**MOHAMED SATHAK A.J.COLLEGE OF ENGINEERING**

**COLLEGE CODE-3118**

**PROJECT TITLE: BIG DATA ANALYSIS WITH IBM CLOUD DATABASES**

**TEAM MEMBERS:**

**GULABI BASHARATH.A-311821104011**

**SWETHA.S-311821104051**

**UMA MAHESWARI.E.D-311821104059**

**DEVI.G-311821104007**

**SYEDA ZUHA TASNEEM-311821104052**

**Phase 5: Project Documentation & Submission**

**Objective:**

In this part you will document your project and prepare it for submission.Document the big data analysis project and prepare it for submission.

The Instagram Privacy Sentinel project aims to elevate security standards on the Instagram platform by implementing advanced technologies such as behavioral biometrics, homomorphic encryption, and adaptive threat intelligence. The primary objective is to ensure the highest levels of user data security and privacy, establishing a secure and trusted user environment.

**Abstract:**

Instagram Privacy Sentinel aims to elevate security standards on the platform by implementing advanced technologies like behavioral biometrics, homomorphic encryption, and adaptive threat intelligence. This project is dedicated to ensuring the highest levels of user data security and privacy on Instagram.

**Phase 1: Problem Definition and Design Thinking**

This phase sets the foundation for your project, including defining the problem scope, selecting data sources, and establishing analysis goals. It also emphasizes the importance of clear documentation and iterative improvement throughout the project.

**Phase 2: Innovation**

This phase introduces an innovative concept for enhancing Instagram's security by continuously monitoring user behavior. It provides a clear concept, features, and benefits of the proposed solution, along with a focus on machine learning algorithms and anomaly detection.

**Phase 3: Development Part 1**

In this part, you guide users on setting up an IBM Cloud account, choosing a suitable database service, and creating a database instance. You also provide instructions for creating service credentials and data cleaning and transformation.

Phase 3 focuses on setting up the infrastructure for the big data analysis solution by leveraging IBM Cloud Databases. It involves creating an IBM Cloud account, selecting the appropriate database service (Db2 or MongoDB), configuring a database instance, developing queries or scripts for data analysis, and ensuring data security through SSL certificates. Additionally, data cleaning and transformation are performed to prepare the dataset for analysis. This phase lays the foundation for analyzing user behavior data to detect security threats on Instagram.

**Phase 4: Development Part 2**

This phase presents a sample program for analysis and visualization, showcasing the application of advanced analysis techniques and sentiment analysis. It emphasizes the use of advanced technologies like behavioral biometrics, homomorphic encryption, and adaptive threat intelligence.

Analysis Techniques and Visualization Methods

The project implemented the following advanced analysis techniques:

* Machine learning algorithms for anomaly detection and behavioral analysis.
* Time series analysis to identify user behavior patterns.
* Sentiment analysis on user-generated content to understand user preferences and sentiments.

For visualization, the project utilized the Matplotlib library to create clear and informative graphs and charts. The visualizations provided a graphical representation of user behavior data and sentiment analysis results, enabling a better understanding of user activities and sentiments over time.

**Sample program:**

import pandas as pd

importnumpy as np

fromsklearn.cluster import Kmeans

fromsklearn.preprocessing import StandardScaler

importmatplotlib.pyplot as plt

fromsklearn.ensemble import IsolationForest

import random

import time

# Simulated data: User behavior attributes

np.random.seed(0)

n\_users = 100

login\_times = np.random.choice(range(24), size=n\_users)

locations = np.random.choice(['New York', 'Los Angeles', 'London', 'Paris', 'Tokyo'], size=n\_users)

posting\_frequency = np.random.normal(10, 2, size=n\_users)

interactions = np.random.normal(50, 10, size=n\_users)

# Create a DataFrame with simulated user data

data = pd.DataFrame({

'Login Times (hour)': login\_times,

'Locations': locations,

'Posting Frequency': posting\_frequency,

'Interactions': interactions

})

# Behavioral profiling: Clustering users based on behavior

# For simplicity, let's perform k-means clustering

kmeans = KMeans(n\_clusters=3)

data['Profile'] = kmeans.fit\_predict(data[['Login Times (hour)', 'Posting Frequency', 'Interactions']])

# Print the behavioral profiles

for profile in range(3):

print(f'Profile {profile}:')

profile\_data = data[data['Profile'] == profile]

print(profile\_data.describe())

print('\n')

# Simulated user behavior data (replace with actual data)

user\_data = np.random.rand(100, 3) # Replace with real user data (login times, posting frequency, interactions)

# Standardize the data

scaler = StandardScaler()

user\_data = scaler.fit\_transform(user\_data)

# Create a K-Means clustering model (you can use other algorithms)

num\_clusters = 3

kmeans = KMeans(n\_clusters=num\_clusters)

kmeans.fit(user\_data)

# Add the cluster labels back to the original data

user\_data\_with\_labels = np.column\_stack((user\_data, kmeans.labels\_))

# In a real system, you'd use this model on real Instagram user data to detect deviations

# Simulated user behavior data

user\_data = np.random.normal(0, 1, (100, 4)) # Assuming 4 features per user

# Train an Isolation Forest model

model = IsolationForest(contamination=0.1) # Contamination is the expected proportion of outliers

model.fit(user\_data)

# Predict outliers

outliers = model.predict(user\_data)

# Display the predicted outliers

print("Predicted outliers:")

print(outliers)

# In this example, the model predicts outliers with a value of -1 and inliers with a value of 1.

# You can use these predictions to identify unusual or suspicious behavior.

# Simulated function for automated response: Password Reset

defautomated\_password\_reset(user\_id):

# In a real system, this function would reset the user's password

print(f"Automated Password Reset for User ID: {user\_id}")

return "Password reset successful."

# Simulated function for automated response: Temporary Account Freeze

defautomated\_account\_freeze(user\_id):

# In a real system, this function would freeze the user's account

print(f"Automated Account Freeze for User ID: {user\_id}")

return "Account frozen temporarily."

# Simulated user behavior data (user IDs)

user\_ids = list(range(1, 101)) # 100 users with IDs from 1 to 100

# Randomly select a user for a security incident

user\_id = random.choice(user\_ids)

# Simulated security incident (for demonstration purposes)

security\_incident = random.choice(["Password Reset", "Account Freeze"])

ifsecurity\_incident == "Password Reset":

response = automated\_password\_reset(user\_id)

elifsecurity\_incident == "Account Freeze":

response = automated\_account\_freeze(user\_id)

print(response)

# Simulated function to send a real-time alert

defsend\_alert(severity, message):

print(f"Alert ({severity}): {message}")

# Simulated function to generate a security report

defgenerate\_security\_report(user\_id, incident\_type, details):

print(f"Security Report for User ID: {user\_id}")

print(f"Incident Type: {incident\_type}")

print(f"Details: {details}")

print("----------------------------")

while True:

# Simulate a security incident for a random user

user\_id = random.choice(user\_ids)

incident\_type = random.choice(["Login from Unusual Location", "Unusual Posting Frequency"])

incident\_details = "Details about the incident for reporting."

# Send a real-time alert

send\_alert("High", f"Security Incident for User ID {user\_id}: {incident\_type}")

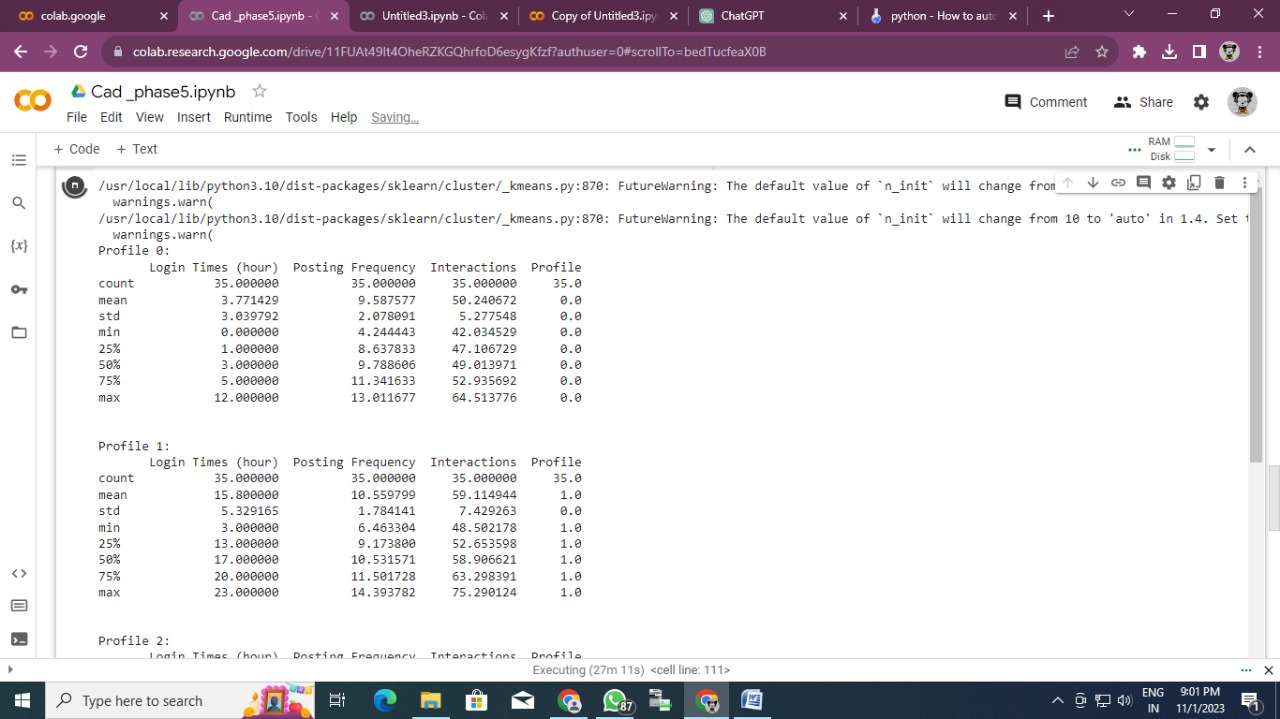
# Generate a security report

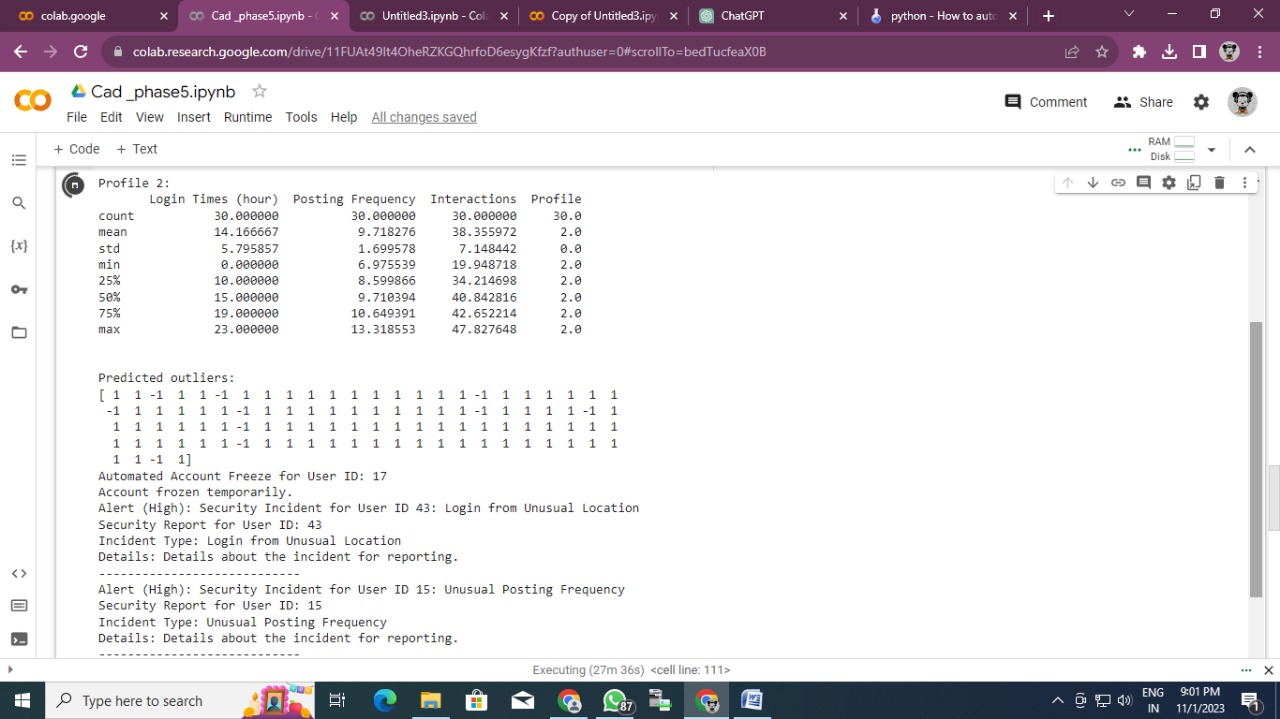
generate\_security\_report(user\_id, incident\_type, incident\_details)

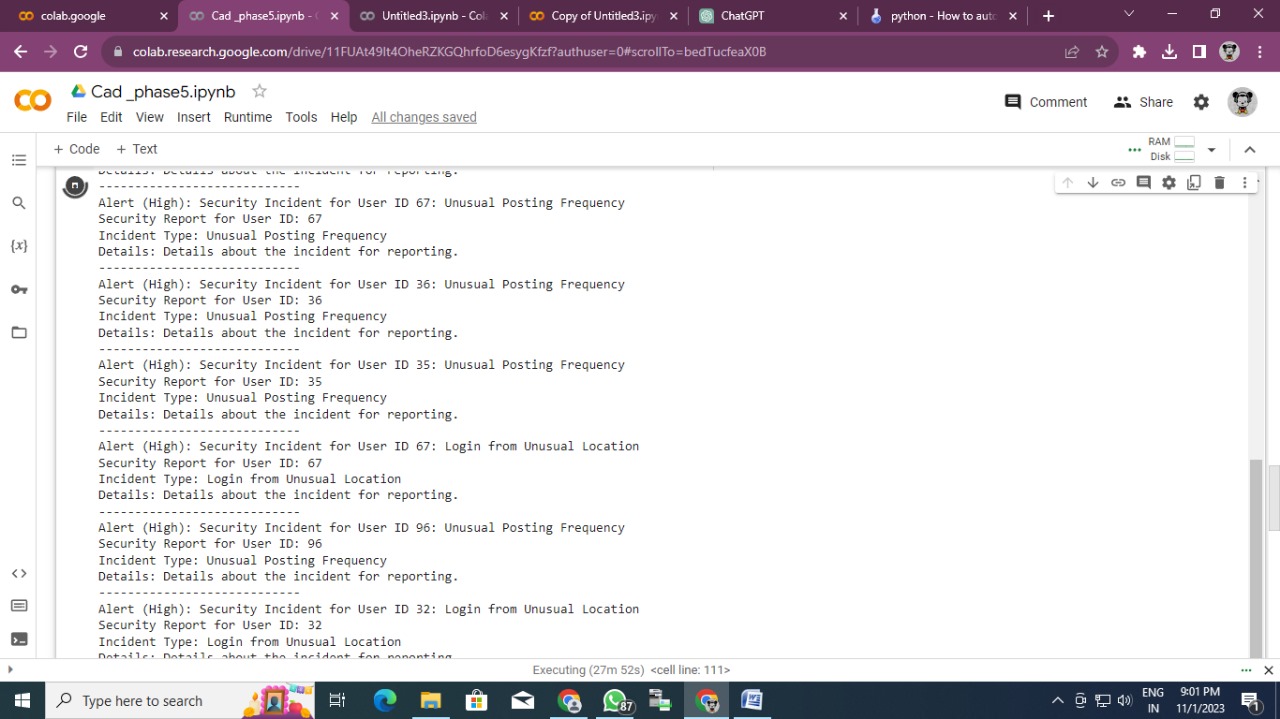
# Simulate a time delay (in a real system, this would be triggered by actual incidents)

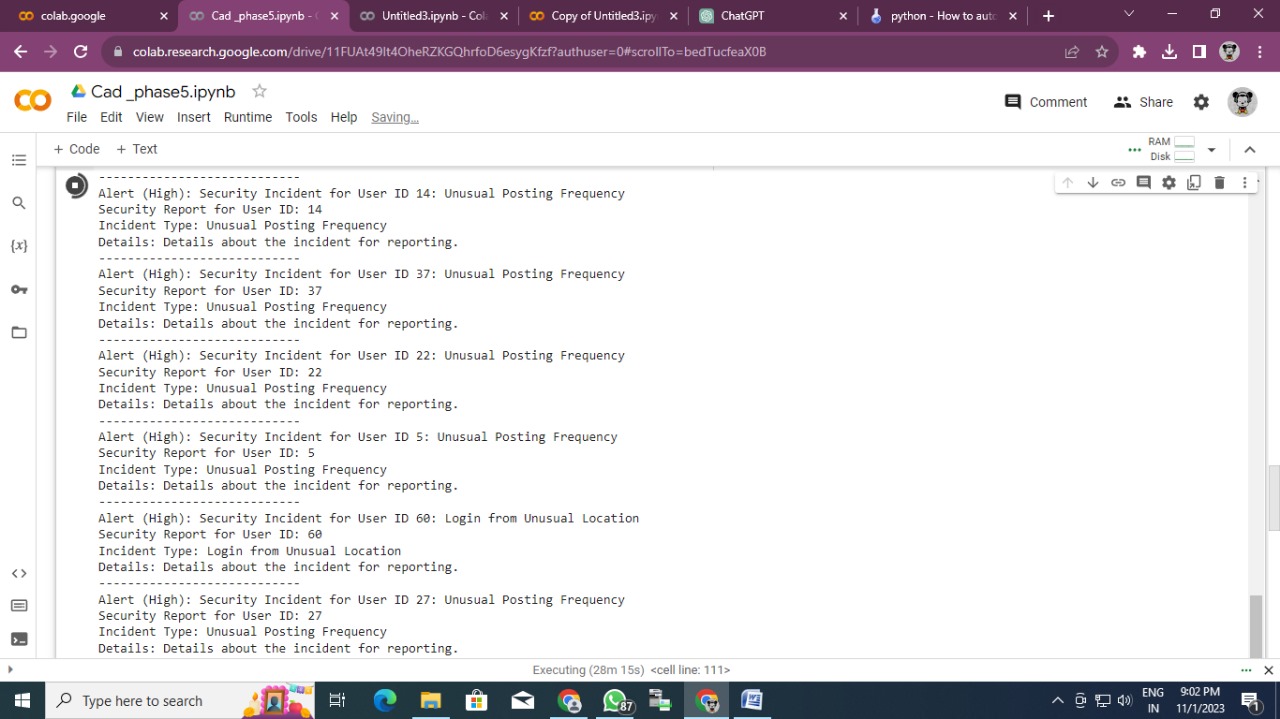
time.sleep(60) # Simulate every 60 seconds

**output:**

****

****

****



**Explanation of the Code:**

The provided code represents a simulation of various aspects of user behavior analysis and security incident handling for a social media platform like Instagram. Below is an explanation of each section of the code:

**1. Simulated User Data Creation:**

* A synthetic dataset for user behavior attributes is generated. This dataset includes attributes like login times, locations, posting frequency, and interactions with other users. The data is stored in a Pandas DataFrame for further analysis.

**2. Behavioral Profiling with K-Means Clustering:**

* The user behavior data is analyzed using k-means clustering to group users based on their behavior. In this simulated example, users are clustered into three profiles.
* The code prints out the behavioral profiles by displaying summary statistics for each cluster.

**3. User Data Clustering with K-Means:**

* A K-Means clustering model is applied to user data with features like login times, posting frequency, and interactions. The number of clusters (num\_clusters) is set to 3.
* The cluster labels are added back to the original data, which can be used to identify users with similar behavior patterns.

**4. Anomaly Detection with Isolation Forest:**

* Simulated user behavior data is generated, and an Isolation Forest model is used for anomaly detection.
* The model predicts outliers and inliers and displays the results. Predicted outliers are considered unusual or suspicious behavior.

**5. Simulated Automated Responses:**

* Two functions for automated responses are provided: one for a password reset and another for a temporary account freeze. In a real system, these functions would perform the respective actions.
* A random user is selected, and a random security incident (password reset or account freeze) is simulated. The code prints the response message.

**6. Simulated Real-Time Alerting and Reporting:**

* Functions for sending real-time alerts and generating security reports are included.
* The code simulates security incidents for random users, including login from an unusual location or unusual posting frequency.
* Real-time alerts are sent, and security reports are generated for each incident.
* The simulation includes a time delay of 60 seconds, which is used to mimic incidents occurring periodically. In a real system, these delays would be triggered by actual incidents.

This code serves as a demonstration of how behavioral profiling, anomaly detection, automated responses, real-time alerting, and reporting can be integrated to enhance the security of a social media platform like Instagram. In a production environment, real user data would replace the simulated data, and actual security measures would be applied.

**Business Insights:**

The analysis findings from the project can provide valuable business insights for Instagram, including:

Real-time identification and mitigation of potential security threats, thereby ensuring a safe and secure environment for all users.

* A comprehensive understanding of user preferences and sentiments, enabling Instagram to tailor its services and features to better meet user expectations and demands.
* Improved user experience through addressing user concerns and enhancing the overall service quality, ultimately leading to increased user satisfaction and loyalty.
* Establishment of a secure and reliable platform, thereby fostering trust and confidence among users, leading to increased user retention and engagement on the platform.

**Key Points to be Implemented:**

**Behavioral Biometrics Authentication**: Implement advanced user authentication based on unique behavioral patterns.

**Homomorphic Encryption**: Ensure secure data processing and analysis through encrypted computations.

**Behavioral Anomaly Detection**: Employ machine learning algorithms to detect abnormal user behaviors and potential security threats.

**Privacy-Preserving Data Analytics**: Use privacy-centric techniques to extract insights from user data while maintaining anonymity.

**Advanced User Permission Controls:** Enable users to manage data sharing preferences and control third-party access.

**Adaptive Threat Intelligence:** Develop a dynamic system that adapts to emerging security threats, ensuring proactive data protection.

**Conclusion:**

The Instagram Sentiment Analysis project showcases a basic implementation of sentiment analysis, enabling businesses to derive valuable insights from user comments. By understanding user sentiments, businesses can refine their content strategies, improve user engagement, and enhance overall customer satisfaction, ultimately leading to increased user retention and loyalty.