Instagram Analysis Project Code Documentation

# 1. Installation of Dependencies

This line installs the `instagrapi` package, essential for interacting with Instagram's private API.

## Code Explanation

pip install instagrapi

Installs the `instagrapi` package.

## Code Explanation

import pandas as pd  
from instagrapi import Client  
  
def login\_to\_instagram(username, password):  
 client = Client()  
 client.login(username, password)  
 return client  
  
def fetch\_brand\_data(client, username):  
 account = client.user\_info\_by\_username(username)  
 followers = account.follower\_count  
 following = account.following\_count  
 total\_posts = account.media\_count  
 print(f"Brand: {username}, Followers: {followers}, Following: {following}, Total Posts: {total\_posts}")  
  
 medias = client.user\_medias(account.pk, amount=1)  
 top\_post = medias[0]  
 print(f"Top Post: {top\_post}, Likes: {top\_post.like\_count}, Comments: {top\_post.comment\_count}")  
  
 comments = client.media\_comments(top\_post.id, amount=20)  
 comment\_texts = [comment.text for comment in comments]  
 print(f"First 20 Comments on Top Post: {comment\_texts}")  
  
 return {  
 "username": username,  
 "followers": followers,  
 "following": following,  
 "total\_posts": total\_posts,  
 "top\_post\_likes": top\_post.like\_count,  
 "top\_post\_comments": top\_post.comment\_count,  
 "comments": comment\_texts  
 }  
  
def save\_data\_to\_excel(data):  
 df = pd.DataFrame.from\_dict(data, orient='index').T  
 df.to\_excel(r"C:\Users\A\Desktop\Hiral Project\Social media analysis\instagram\_data.xlsx", index=False)  
  
def main():  
 username = 'instadatascraper'  
 password = 'rizulvaidya'  
 client = login\_to\_instagram(username, password)  
  
 brands = ['apple', 'samsung', 'google']  
 data = {}  
 for brand in brands:  
 data[brand] = fetch\_brand\_data(client, brand)  
   
 save\_data\_to\_excel(data)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Defines functions to log in to Instagram, fetch data for specific brands, and save the data to an Excel file.

## Code Explanation

def calculate\_engagement(likes, comments, followers):  
 Calculates engagement score based on likes, comments, and followers.  
 return (likes + comments) / followers \* 100  
...  
data\_list = [apple\_data, samsung\_data, google\_data]  
labels = ['Apple', 'Samsung', 'Google']  
plot\_summary\_statistics(data\_list, labels)  
  
from IPython.display import Markdown, display  
  
def display\_conclusion():  
 summary\_text =   
 ...  
 display(Markdown(summary\_text))  
  
display\_conclusion()

Functions to calculate engagement, display engagement statistics, summarize findings, and provide recommendations.

# 2. Main Analysis Functions

This section sets up the main functions to log into Instagram, fetch user data, and save it.

## Code Explanation

pip install instagrapi

Installs the `instagrapi` package.

## Code Explanation

import pandas as pd  
from instagrapi import Client  
  
def login\_to\_instagram(username, password):  
 client = Client()  
 client.login(username, password)  
 return client  
  
def fetch\_brand\_data(client, username):  
 account = client.user\_info\_by\_username(username)  
 followers = account.follower\_count  
 following = account.following\_count  
 total\_posts = account.media\_count  
 print(f"Brand: {username}, Followers: {followers}, Following: {following}, Total Posts: {total\_posts}")  
  
 medias = client.user\_medias(account.pk, amount=1)  
 top\_post = medias[0]  
 print(f"Top Post: {top\_post}, Likes: {top\_post.like\_count}, Comments: {top\_post.comment\_count}")  
  
 comments = client.media\_comments(top\_post.id, amount=20)  
 comment\_texts = [comment.text for comment in comments]  
 print(f"First 20 Comments on Top Post: {comment\_texts}")  
  
 return {  
 "username": username,  
 "followers": followers,  
 "following": following,  
 "total\_posts": total\_posts,  
 "top\_post\_likes": top\_post.like\_count,  
 "top\_post\_comments": top\_post.comment\_count,  
 "comments": comment\_texts  
 }  
  
def save\_data\_to\_excel(data):  
 df = pd.DataFrame.from\_dict(data, orient='index').T  
 df.to\_excel(r"C:\Users\A\Desktop\Hiral Project\Social media analysis\instagram\_data.xlsx", index=False)  
  
def main():  
 username = 'instadatascraper'  
 password = 'rizulvaidya'  
 client = login\_to\_instagram(username, password)  
  
 brands = ['apple', 'samsung', 'google']  
 data = {}  
 for brand in brands:  
 data[brand] = fetch\_brand\_data(client, brand)  
   
 save\_data\_to\_excel(data)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Defines functions to log in to Instagram, fetch data for specific brands, and save the data to an Excel file.

## Code Explanation

def calculate\_engagement(likes, comments, followers):  
 Calculates engagement score based on likes, comments, and followers.  
 return (likes + comments) / followers \* 100  
...  
data\_list = [apple\_data, samsung\_data, google\_data]  
labels = ['Apple', 'Samsung', 'Google']  
plot\_summary\_statistics(data\_list, labels)  
  
from IPython.display import Markdown, display  
  
def display\_conclusion():  
 summary\_text =   
 ...  
 display(Markdown(summary\_text))  
  
display\_conclusion()

Functions to calculate engagement, display engagement statistics, summarize findings, and provide recommendations.

# 3. Visualization and Summary

This section includes functions to calculate and display engagement scores, and a summary of findings and recommendations.

## Code Explanation

pip install instagrapi

Installs the `instagrapi` package.

## Code Explanation

import pandas as pd  
from instagrapi import Client  
  
def login\_to\_instagram(username, password):  
 client = Client()  
 client.login(username, password)  
 return client  
  
def fetch\_brand\_data(client, username):  
 account = client.user\_info\_by\_username(username)  
 followers = account.follower\_count  
 following = account.following\_count  
 total\_posts = account.media\_count  
 print(f"Brand: {username}, Followers: {followers}, Following: {following}, Total Posts: {total\_posts}")  
  
 medias = client.user\_medias(account.pk, amount=1)  
 top\_post = medias[0]  
 print(f"Top Post: {top\_post}, Likes: {top\_post.like\_count}, Comments: {top\_post.comment\_count}")  
  
 comments = client.media\_comments(top\_post.id, amount=20)  
 comment\_texts = [comment.text for comment in comments]  
 print(f"First 20 Comments on Top Post: {comment\_texts}")  
  
 return {  
 "username": username,  
 "followers": followers,  
 "following": following,  
 "total\_posts": total\_posts,  
 "top\_post\_likes": top\_post.like\_count,  
 "top\_post\_comments": top\_post.comment\_count,  
 "comments": comment\_texts  
 }  
  
def save\_data\_to\_excel(data):  
 df = pd.DataFrame.from\_dict(data, orient='index').T  
 df.to\_excel(r"C:\Users\A\Desktop\Hiral Project\Social media analysis\instagram\_data.xlsx", index=False)  
  
def main():  
 username = 'instadatascraper'  
 password = 'rizulvaidya'  
 client = login\_to\_instagram(username, password)  
  
 brands = ['apple', 'samsung', 'google']  
 data = {}  
 for brand in brands:  
 data[brand] = fetch\_brand\_data(client, brand)  
   
 save\_data\_to\_excel(data)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Defines functions to log in to Instagram, fetch data for specific brands, and save the data to an Excel file.

## Code Explanation

def calculate\_engagement(likes, comments, followers):  
 Calculates engagement score based on likes, comments, and followers.  
 return (likes + comments) / followers \* 100  
...  
data\_list = [apple\_data, samsung\_data, google\_data]  
labels = ['Apple', 'Samsung', 'Google']  
plot\_summary\_statistics(data\_list, labels)  
  
from IPython.display import Markdown, display  
  
def display\_conclusion():  
 summary\_text =   
 ...  
 display(Markdown(summary\_text))  
  
display\_conclusion()

Functions to calculate engagement, display engagement statistics, summarize findings, and provide recommendations.