CS 210 – Course Project

Mehmet Serdar ÖZCAN - 26994

Instructor:

Onur Varol

Date: 18.01.2024

Sabancı Üniversitesi

SECTIONS

1. **Data Parsing and Preprocessing**
    a. Read and parse HTML content using BeautifulSoup.
    b. Create a DataFrame containing user information and the date of liked posts.

2. **Exploratory Data Analysis (EDA)**
    c. Visualize monthly, daily, and hourly likes to identify patterns.
    d. Analyze likes based on user, season, and day of the week.
    e. Identify top users and visualize their engagement.

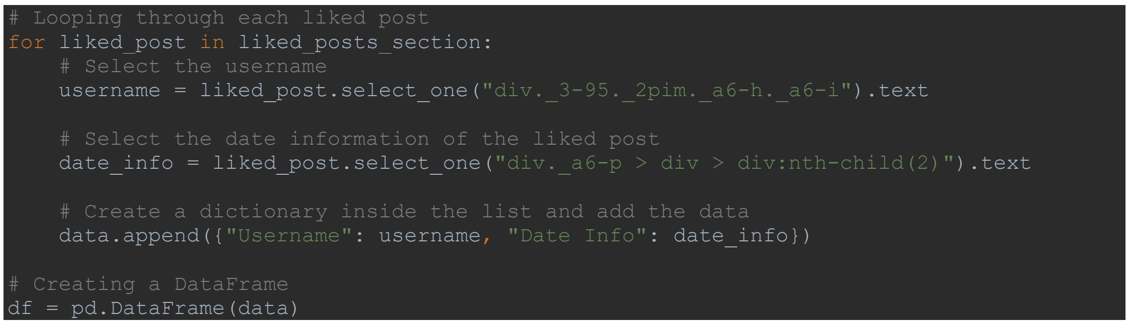
3. **Machine Learning Predictive Modeling**
    f. Use RandomForestRegressor to predict the number of likes based on features like day of the week and hour of the day.
    g. Evaluate the model's performance and visualize actual vs. predicted values.

4. **Time-Series Analysis**
    h. Apply time-series forecasting models like ARIMA to predict future likes.

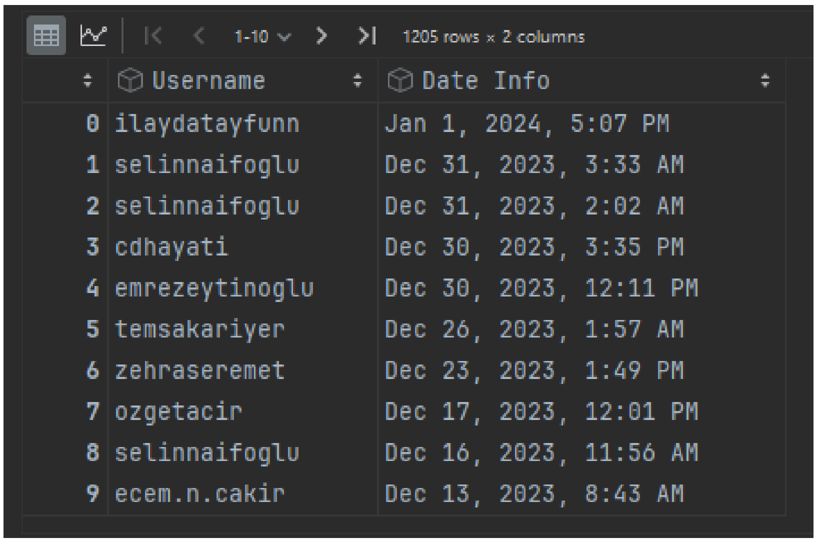
5. **Conclusion and Insights**
    i. Summarize key findings and insights from the analysis.
    j. Provide recommendations for improving user engagement.

1. Data Parsing and Preprocessing

Firstly, the data parsing and preprocessing stage involves reading and parsing HTML content using BeautifulSoup. This process is crucial for extracting relevant information from the HTML file. Subsequently, a DataFrame is created to store user information along with the date of liked posts. This structured representation of data sets the foundation for further exploration and analysis.



```python
# Looping through each liked post
for liked_post in liked_posts_section:
    # Select the username
    username = liked_post.select_one("div._3-95._2pim._a6-h._a6-i").text

    # Select the date information of the liked post
    date_info = liked_post.select_one("div._a6-p > div > div:nth-child(2)").text

    # Create a dictionary inside the list and add the data
    data.append({"Username": username, "Date Info": date_info})

# Creating a DataFrame
df = pd.DataFrame(data)
```



```
   Username          Date Info
0  ilaydatayfunn     Jan 1, 2024, 5:07 PM
1  selinnaifoglu     Dec 31, 2023, 3:33 AM
2  selinnaifoglu     Dec 31, 2023, 2:02 AM
3  cdhayati          Dec 30, 2023, 3:35 PM
4  emrezeytinoglu    Dec 30, 2023, 12:11 PM
5  temsakariyer      Dec 26, 2023, 1:57 AM
6  zehraseremet      Dec 23, 2023, 1:49 PM
7  ozgetacir         Dec 17, 2023, 12:01 PM
8  selinnaifoglu     Dec 16, 2023, 11:56 AM
9  ecem.n.cakir      Dec 13, 2023, 8:43 AM
```

Figure 1: First state of dataframe

2. Exploratory Data Analysis (EDA)

Moving on to the exploratory data analysis (EDA) phase, the second step (c) involves visualizing monthly, daily, and hourly likes to identify patterns in user engagement over time. This step provides valuable insights into the temporal distribution of likes, allowing for the detection of any recurring trends or notable variations.

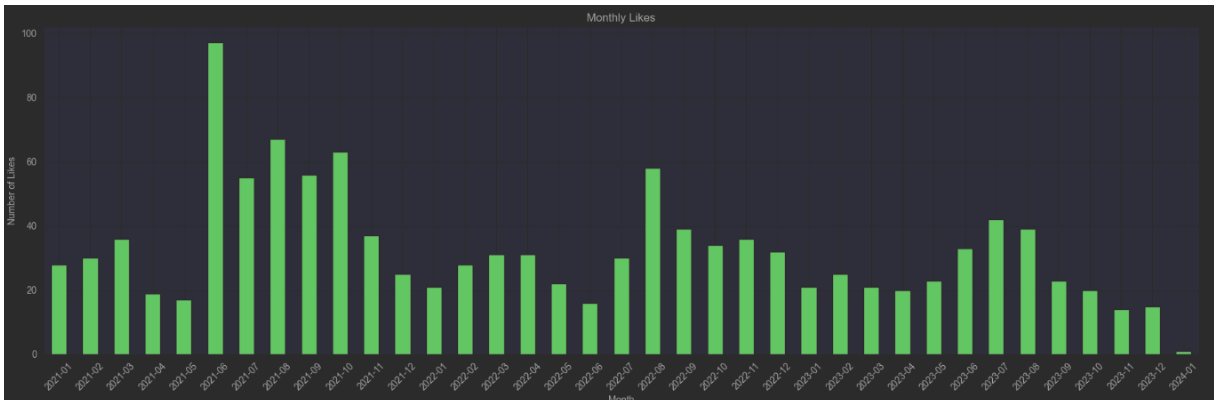


Figure 2: Monthly total likes (per year)

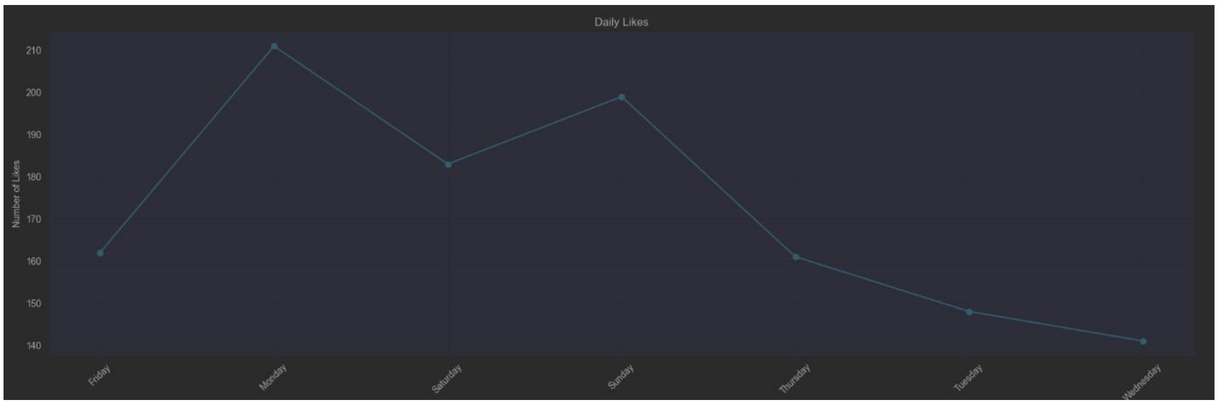


Figure 3: Daily total likes

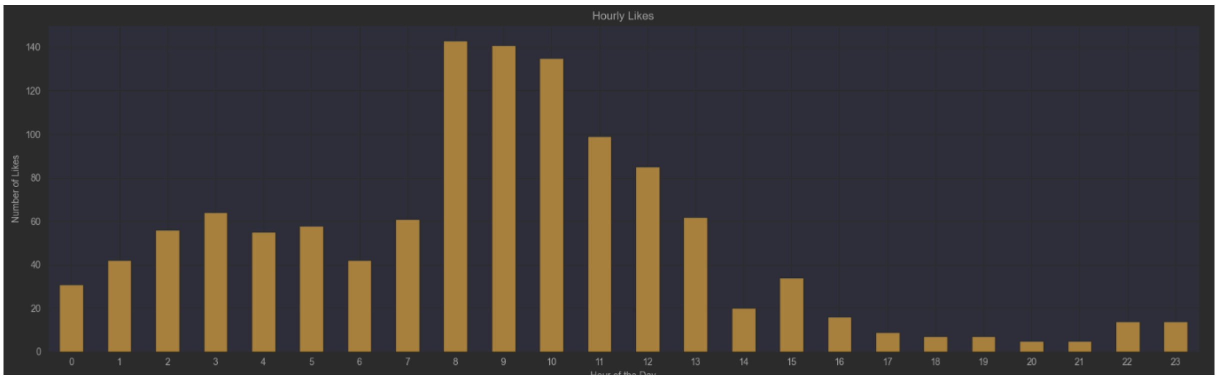
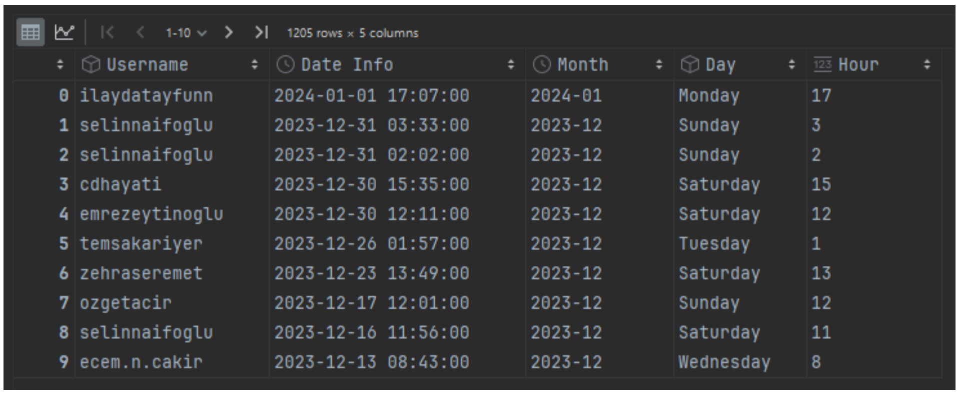


Figure 4: Hourly total likes



	Username	Date Info	Month	Day	Hour
0	ilaydatayfunn	2024-01-01 17:07:00	2024-01	Monday	17
1	selinnaifoglu	2023-12-31 03:33:00	2023-12	Sunday	3
2	selinnaifoglu	2023-12-31 02:02:00	2023-12	Sunday	2
3	cdhayati	2023-12-30 15:35:00	2023-12	Saturday	15
4	emrezeytinoglu	2023-12-30 12:11:00	2023-12	Saturday	12
5	temsakariyer	2023-12-26 01:57:00	2023-12	Tuesday	1
6	zehraseremet	2023-12-23 13:49:00	2023-12	Saturday	13
7	ozgetacir	2023-12-17 12:01:00	2023-12	Sunday	12
8	selinnaifoglu	2023-12-16 11:56:00	2023-12	Saturday	11
9	ecem.n.cakir	2023-12-13 08:43:00	2023-12	Wednesday	8

Figure 5: New state of dataframe with new columns

Continuing with the EDA, the third step (d) focuses on analyzing likes based on user, season, and day of the week. This granular analysis helps uncover patterns related to user engagement preferences, seasonal variations, and day-specific trends, contributing to a comprehensive understanding of user behavior.

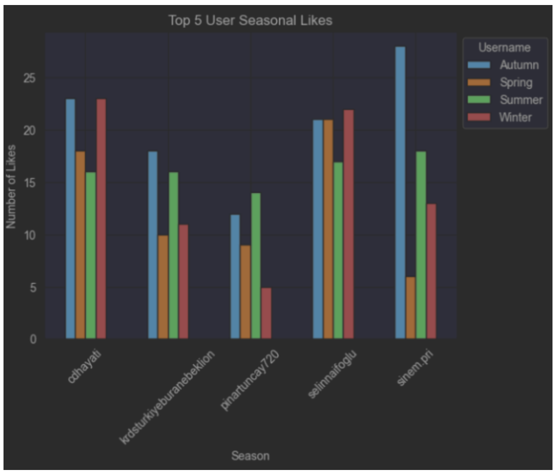


Figure 6: Season based Top 5 user I liked

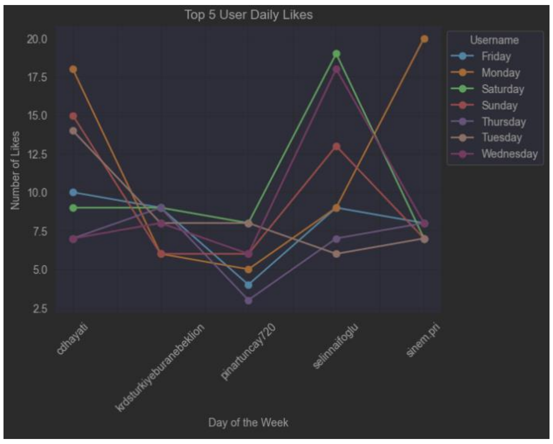


Figure 7: Day based Top 5 user I liked

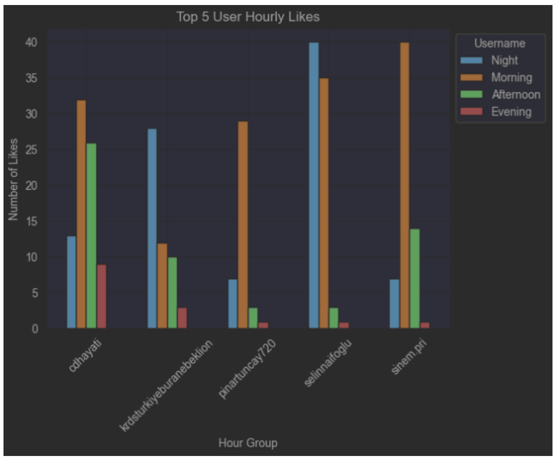


Figure 8: Hour based Top 5 user I liked

In step (e), the EDA process identifies and visualizes the engagement of top users. This involves recognizing users with the highest likes and presenting their engagement patterns. Understanding the behavior of top users is essential for gaining insights into the factors influencing higher engagement rates.

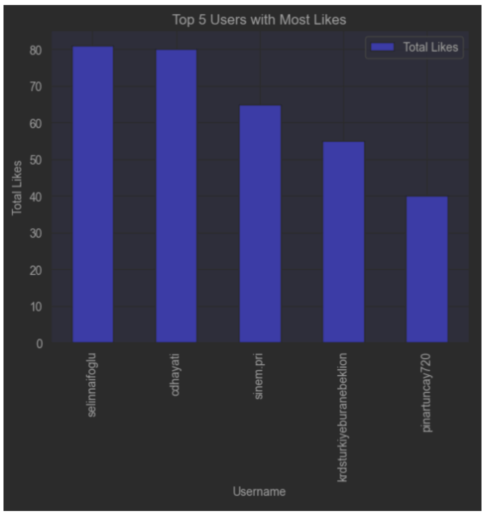


Figure 9: Top 5 users with most likes

```
# Setting the order of days of the week
day_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
df['Day of Week'] = pd.Categorical(df['Day of Week'], categories=day_order, ordered=True)

# Grouping by day of the week and weekday/weekend
likes_by_day = df.groupby(['Day of Week', 'Weekday/Weekend'], observed=True).size().unstack()
```

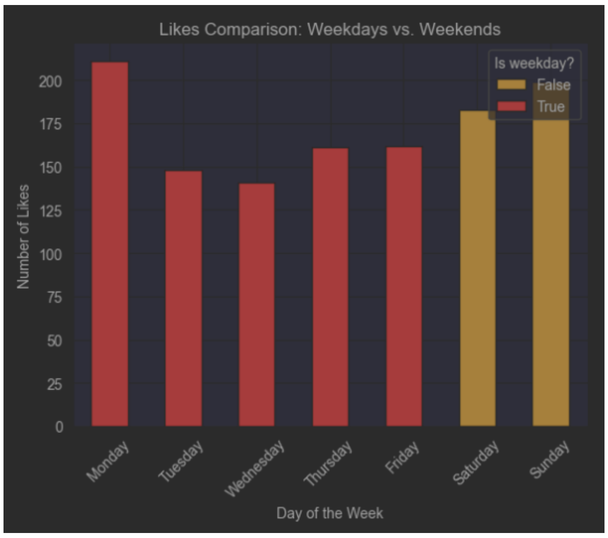


Figure 10: Likes comparison for weekdays between weekends

```python
# List of years to analyze
years_to_analyze = [2021, 2022, 2023]  # Add more years if needed

for year in years_to_analyze:
    # Filtering data for the current year
    df_year = df[df['Date Info'].dt.year == year]

    # Getting the top 5 users for the current year
    top_users = df_year['Username'].value_counts().nlargest(5).index
    df_top_users = df_year[df_year['Username'].isin(top_users)]

    # Grouping by username and month for the current year
    likes_by_user_and_month_yearly = df_top_users.groupby(['Username',
'Month']).size().unstack().fillna(0)
```

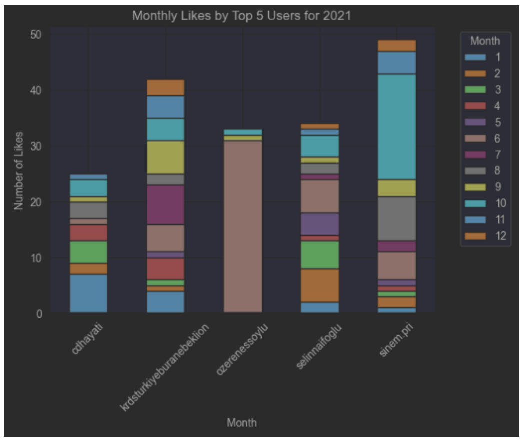


Figure 11: Monthly likes by top 5 users for 2021

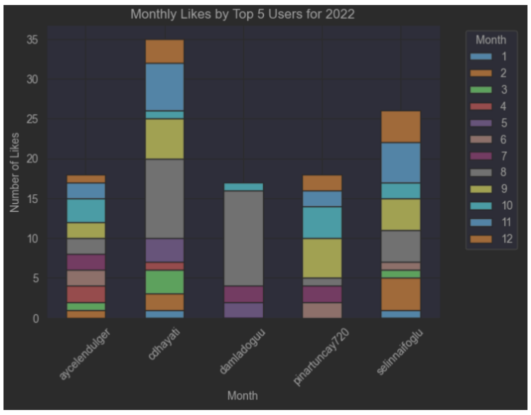


Figure 12: Monthly likes by top 5 users for 2022

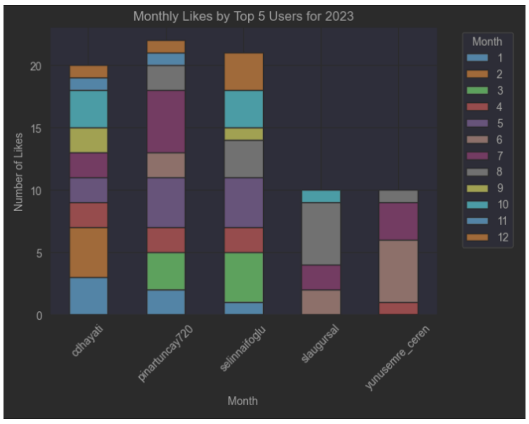


Figure 13: Monthly likes by top 5 users for 2023

```python
# List of years to analyze
years_to_analyze = [2021, 2022, 2023]

# Iterating over each year and plot the likes by usernames using assigned colors
for year in years_to_analyze:
    # Filtering data for the current year
    df_year = df[df['Date Info'].dt.year == year]

    # Grouping by username and count the likes
    likes_by_user = df_year['Username'].value_counts()

    # Using the colors assigned to usernames
    colors = [username_color_map[username] for username in likes_by_user.index]

    # Getting the top 5 users for the current year
    top5_users = likes_by_user.head(5)
```

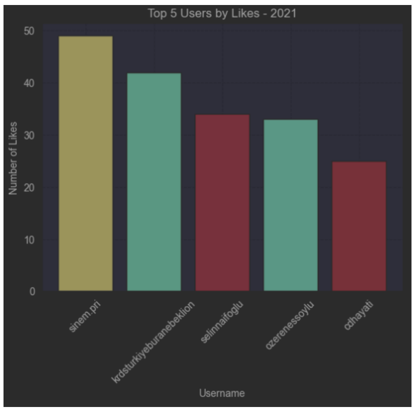


Figure 14: Top 5 Users By Likes for 2021

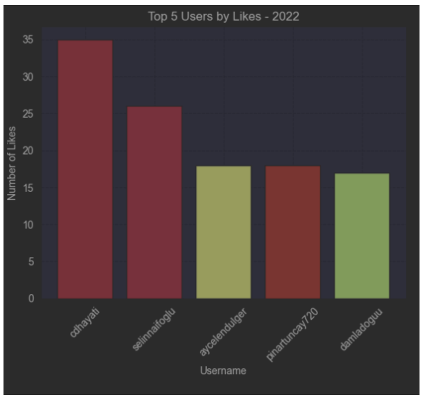


Figure 15: Top 5 Users By Likes for 2022

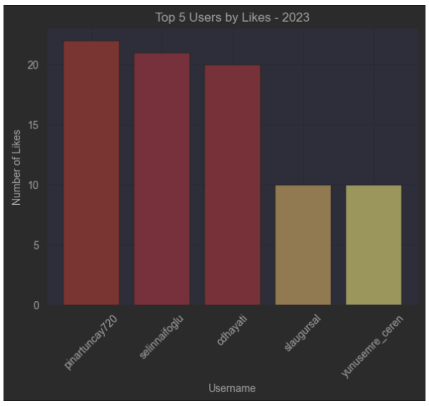
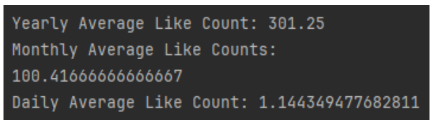


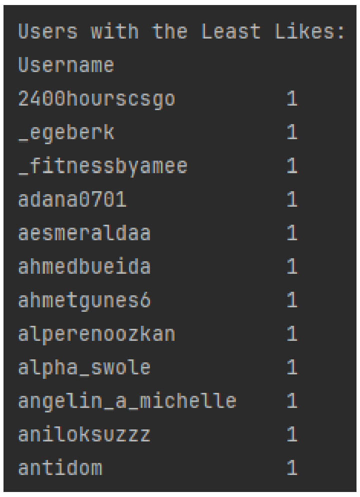
Figure 16: Top 5 Users By Likes for 2023

```python
# Extracting month and year information
df['Year'] = df['Date Info'].dt.year

# Calculating yearly average like counts
yearly_avg_likes = df.groupby('Year').size().mean()

# Calculating monthly average like counts
monthly_avg_likes = df.groupby('Month').size().mean()

# Calculating daily average like counts
daily_avg_likes = df.groupby('Date Info').size().mean()

print("Yearly Average Like Count:", yearly_avg_likes)
print("Monthly Average Like Counts:")
print(monthly_avg_likes)
print("Daily Average Like Count:", daily_avg_likes)
```

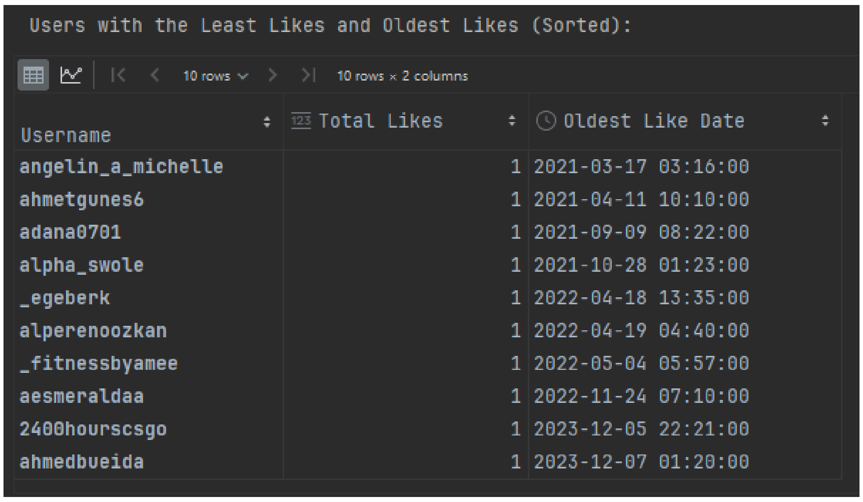


```
Yearly Average Like Count: 301.25
Monthly Average Like Counts:
100.41666666666667
Daily Average Like Count: 1.144349477682811
```

```python
# Finding the total like counts for each user
likes_count_by_user = df.groupby('Username').size()

# Identifying the users with the least likes
least_liked_users = likes_count_by_user.nsmallest(
    20)
# For example, we are taking the first 20 users with the least likes
print("Users with the Least Likes:")
print(least_liked_users)
```



```
Users with the Least Likes:
Username
2400hourscsgo        1
_egeberk             1
_fitnessbyamee       1
adana0701            1
aesmeraldaa          1
ahmedbueida          1
ahmetgunes6          1
alperenoozkan        1
alpha_swole          1
angelin_a_michelle   1
aniloksuzzz          1
antidom              1
```



Users with the Least Likes and Oldest Likes (Sorted):

Username	Total Likes	Oldest Like Date
angelin_a_michelle	1	2021-03-17 03:16:00
ahmetgunes6	1	2021-04-11 10:10:00
adana0701	1	2021-09-09 08:22:00
alpha_swole	1	2021-10-28 01:23:00
_egeberk	1	2022-04-18 13:35:00
alperenoozkan	1	2022-04-19 04:40:00
_fitnessbyamee	1	2022-05-04 05:57:00
aesmeraldaa	1	2022-11-24 07:10:00
2400hourscsgo	1	2023-12-05 22:21:00
ahmedbueida	1	2023-12-07 01:20:00

3. Machine Learning Predictive Modeling

Transitioning to machine learning predictive modeling (step 3), the fourth step (f) involves utilizing the RandomForestRegressor model. This model is applied to predict the number of likes based on specific features such as the day of the week and hour of the day. The predictive model serves as a valuable tool for forecasting user engagement.

```
# Encode categorical features
label_encoder = LabelEncoder()
df['Username'] = label_encoder.fit_transform(df['Username'])
df['Likes'] = total_likes_by_user[df['Username']].values

# Defining features and target variable
features = ['Username', 'DayOfWeek', 'HourOfDay']
X = df[features]
y = df['Likes']

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initializing the model (Random Forest Regressor)
model = RandomForestRegressor(n_estimators=100, random_state=42)

# Training the model
model.fit(X_train, y_train)

# Predicting on the test set
y_pred = model.predict(X_test)
```

In step (g), the evaluation of the model's performance is conducted, and the results are visualized to compare actual vs. predicted values. This step provides a quantitative assessment of the model's accuracy and reliability in predicting user likes.

```
Mean Squared Error: 4.705951867219917
Train Score: 0.9961210972989045
Test Score: 0.9944540670772736
```

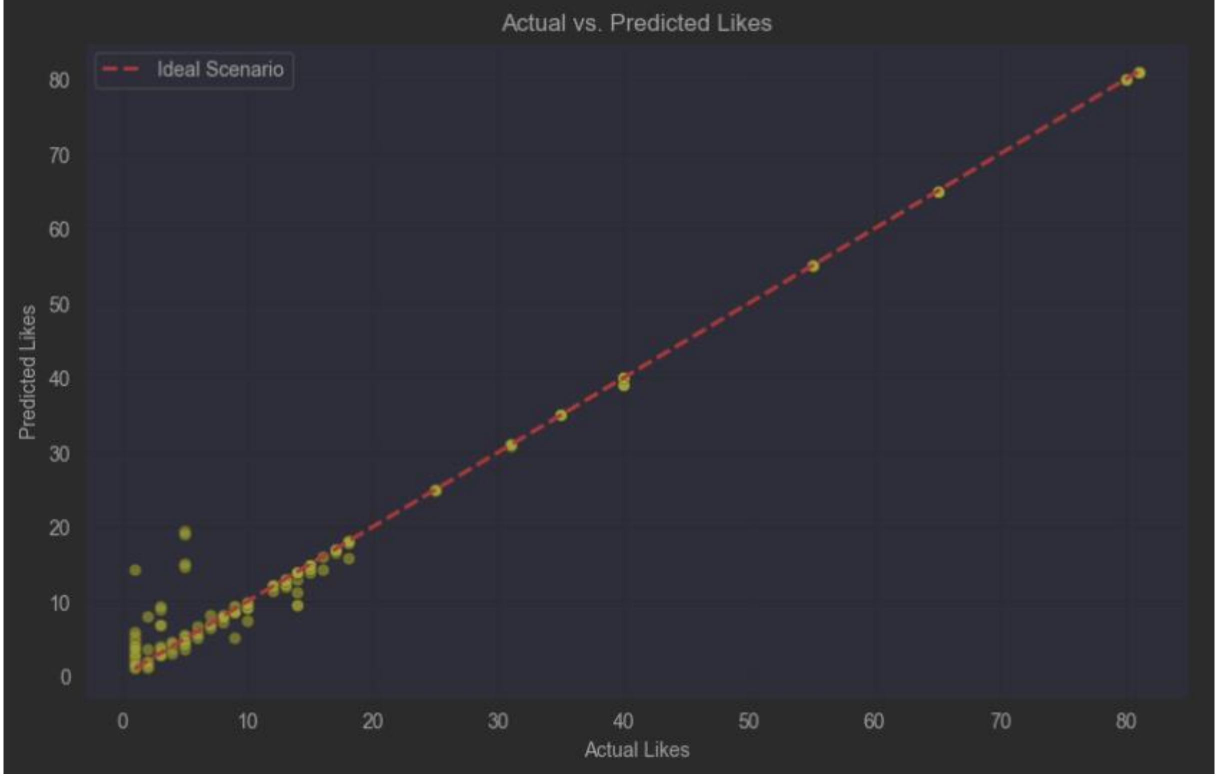
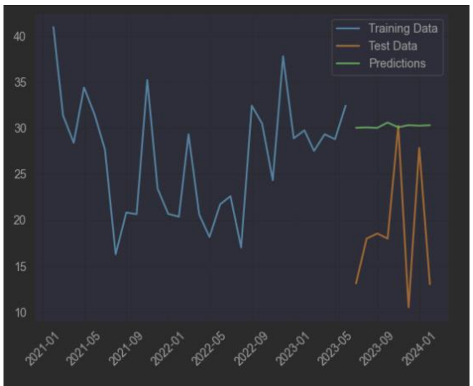


Figure 17: Actual vs. Predicted Total likes for users (with Random Forest Regressor)

4. Time-Series Analysis

Moving to time-series analysis (step 4), the fifth step (h) introduces the application of time-series forecasting models such as ARIMA. These models are employed to predict future likes, considering the historical patterns and trends identified in the previous steps.

```
# Splitting the time series into training and test sets
train_size = int(len(y) * 0.8)
train, test = y[:train_size], y[train_size:]

# Creating and training the ARIMA model
model = ARIMA(train, order=(5, 1, 0))  # Model parameters are just examples, can be adjusted based on needs
fit_model = model.fit()

# Making predictions on the test set
y_pred = fit_model.predict(start=len(train), end=len(train) + len(test) - 1, typ='levels')
```



```
Mean Squared Error: 177.1480713436416
```

5. Conclusion and Insights

- I displayed the total likes by month in each month of the year. I liked the most in June 2021.
- Divided into days, I found that I liked the most on Mondays.
- I saw that I liked the most between 8 and 10 in the morning.

- In addition, I displayed the distribution of likes for the photos of the 5 users whose posts I liked the most, according to seasons. Additionally, I displayed the distributions broken down by 7 days and hours of a day.

- I displayed the total number of likes of the 5 users whose posts I liked most.

- I analyzed my total number of likes according to weekdays and weekends. I saw that there was more on weekend days and Monday than other days.

- For each of the years 2021, 2022 and 2023, I displayed the 5 users whose posts I liked the most and in which month in that year I liked their posts more.

- At the same time, I identified the 5 users I liked the most, separately for these 3 years, according to those years.

- I determined the average number of likes on a yearly, monthly and daily basis.

- I identified the names of the 20 users I liked the least, and also identified the oldest likes among the people I liked the least.

- Finally, I got simple outputs by using random forest regression for classification and ARIMA models for time series analysis.