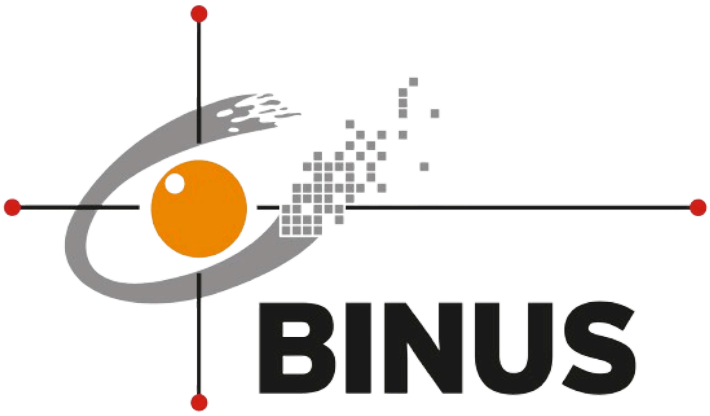


Fundamentals of Data Science Final Project Report

Instagram Post Engagement Rate Prediction

Lecturer: Ms. Nunung Nurul Qomariyah

Class: L3AC

Computer Science Program
School of Computing and Creative Arts

Bina Nusantara International University
Jakarta, 2024

Name: Gabriel Anderson
NIM: 2702256315

Name: Rafael Anderson
NIM: 2702255981

Name: Albertus Santoso
NIM: 2702334885

Table Of Contents

I. Problem Analysis

In the current digital world, the number of social media accounts are increasing at a rapid rate [1]. In 2024, the number of social media users increased from 4.72 billion to 5.02 billion users across all social media platforms from the previous year, which is an astounding increase of 320 million new users [2]. Instagram is one of the more common social media platforms, who has the 4th most active users which is over 2 billion users [3]. Instagram growth is high with a 50 million growth rate from 2023 to 2024 [4]. Instagram users spend around 31.4 hours daily with a daily users of around 500 million [5]. The continuously increasing number of Instagram users has increased the competition between brands and influencers to capture as much engagement rate from the users.

Instagram being the 4th most popular social media platform, it is a platform often used by influencers to promote products, where diverse sizes of businesses utilize this platform, since it is extremely helpful in increasing their brand awareness [6]. Instagram has a steady increase in ad reach with a 12.2 percent increase year to year [7]. However, finding which instagram accounts that receive the overall most rate of engagement from the users is a challenge faced by most brands trying to find a suitable promoter for their brand. The engagement rate is calculated by getting the average of the total likes and total comments, which is then divided by the amount of followers, and then multiplied by 100. This means that the average engagement rate doesn't necessarily represent the number of users, as it is proportional to the number of followers.

It is crucial to find the amount of active users by making it proportional to the number of followers. As brands would prefer quality over quantity, brands would rather find Instagram accounts with audiences interested in the content of that account than an account with more likes but a lower engagement rate. Interested audiences will talk and share the product, providing additional exposure to the promoted product [8]. This is also important for companies or people trying to find a suitable influencer to promote their product or service. The size of an influencer is

strongly correlated with the cost of asking them to promote a product or service. For example, the cost of sponsoring an influencer with a larger following will generally be more expensive than an influencer with a smaller following. Thus, by finding smaller influencers with a decent or larger engagement rate, it may give brands higher quality audiences and be charged less amount for the sponsor.

II. Related Works

Based on a study conducted by Trunfio and Rossi [9], it stated that social media engagement has a multidimensional and polysemic nature, requiring complex models to analyze it. The study suggests the usage of the COBRA model (Consumer Online Brand Related Activities) as a tool to analyze the engagement on social media. COBRA model analyzes all 3 dimensions of actions from the behavioral perspective, which are consumptions, contributions, and creation. Basically, it applies different weights for different social media platforms, as some social media platforms can remove or hide the exact number of social media metrics, including likes, comments, and subscribers, which can result in inaccuracies [10].

A research by [11] explored the aspects of influencers, and how those aspects affect their engagement rate. The engagement of the follower is determined by three critical factors, including content, source, and psychological characteristics, where source and psychological characteristics receive limited attention during engagement analysis, compared to content characteristics. To resolve this issue, the research proposes a model that is supported by both Elaboration Likelihood Model (ELM), and Dual Process Theory. The ELM is able to highlight the content's quality, and the attractiveness of the source. On the other hand, Dual Process Theory focuses more on the psychological aspects. The research explains how consumers switch between automatic and reflective processing when encountering influencer content.

Another study by [12] utilizes past studies, which circulate around engagement rates on social media, finding patterns from these studies. One of the findings states that brand awareness has a significant impact on the impression rate and reach. The study suggests understanding of social media metrics and analytics based on one's business. Additionally, providing consumers with content they want is crucial, and less of the brand's promotion. Lastly, it states that analyzing the optimal way of calculating social media metrics to measure a business' brand awareness, and observing how social media is adopted by different industries are important, and must be considered. Lastly, a study by [13] revolves around the studying of accounts of MSME actors, where it was found that most MSME weren't able to optimize their accounts in attracting engagement. It stated that the engagement level is crucial, as it tells how much influence an account has on its followers.

Our study aims to approach these issues by using a unique approach, machine learning. Unlike some of the studies, this study solely focuses on the engagement rate on the Instagram platform. By performing analysis on the quantitative aspects, new patterns can be discovered. Additionally, implementing various machine learning models with different levels of complexities can result in a better engagement rate analysis. Hopefully, the insights discovered from this study can be helpful, providing marketers with a reliable method in increasing the engagement rate of their Instagram account. Machine learning models like gradient boost, K-nearest neighbors, random forest, extreme gradient boost, and categorial boosting, are utilized.

III. Dataset and Preprocessing

The size of the data after cleaning is (922, 9), with 922 records and 9 attributes. Based on the data, it can be separated into several aspects. Which includes the origin of the account, which is either Indonesian or Global, and by the following size of the account, including mega (More than 1M), macro (100k - 1M), micro (10K - 100K), and nano (1K - 10K).

Dataset Snippet:

	Followers	Average_Likes	Average_Comments	Average_Engagement_Rate	Engagement_Rate_(60_Days)	Posting_frequency_(last_60_days)	Posts_(image)	Posts_(videos)	Posts_(carousel)
2	47352198	458723.76	4422.62	0.9781	0.4896	132.0	9.0	34.0	89.0
3	39581466	118171.13	692.59	0.3003	0.0018	4.0	0.0	0.0	4.0
4	38776943	139324.51	763.73	0.3613	0.1303	100.0	42.0	44.0	14.0
5	36318320	152328.63	2451.69	0.4262	0.22	83.0	8.0	46.0	29.0
6	34486065	145525.51	948.45	0.4247	0.2491	98.0	5.0	56.0	37.0
7	1245315	164240.73	1246.38	13.2888	0.2431	1.0	0.0	0.0	1.0
8	1808490	70429.07	1255.41	3.9638	0.2944	5.0	1.0	4.0	0.0
9	80883913	3674629.51	16979.32	4.5641	5.2609	26.0	7.0	9.0	10.0
10	33412540	272957.06	1596.6	0.8217	0.692	47.0	1.0	11.0	35.0
11	2047559	185356.19	630.77	9.0834	4.2291	23.0	0.0	13.0	10.0
12	1811551	26212.84	145.84	1.455	0.7947	13.0	0.0	2.0	11.0
13	27521485	731834.07	3964.1	2.6735	4.405	7.0	0.0	2.0	5.0
14	1180207	154382.52	1755.34	13.2297	0.5307	1.0	0.0	0.0	1.0
15	7994281	201394.06	628.37	2.5271	1.5076	29.0	4.0	8.0	17.0
16	35137412	291341.05	2557.12	0.8364	0.4342	30.0	4.0	26.0	0.0
17	27421149	150723.02	3042.12	0.5608	0.1551	35.0	2.0	24.0	9.0
18	7146979	700613.67	5303.04	9.8771	5.8659	4.0	0.0	0.0	4.0
19	7900796	94032.52	642.67	1.1983	0.6597	48.0	16.0	16.0	16.0
20	8277231	266408.27	1143.38	3.2324	3.1181	2.0	0.0	0.0	2.0
21	8660319	55255.08	150.44	0.6398	0.0008	12.0	1.0	4.0	7.0

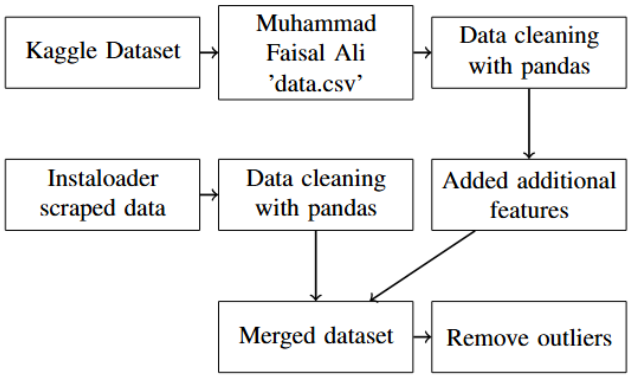
Attributes Table:

The description of all attributes and sample data are present in the following table:

Attribute	Description	Sample Data
Username	Username of the account	cristiano
Followers	Follower count of the account	47352198
Average Likes	The average likes of the account	458723.76
Average Comments	The average comments of the account	4422.62
Average Engagement Rate	Value calculated from the average likes, comments, and followers	0.9781
Engagement Rate (60 Days)	Same as the average engagement rate, but only containing data from posts posted the last 60 days	0.4896
Posting Frequency (60 Days)	Number of posts posted (60 Days)	132
Posts (Image)	Number of posts of the image type (60 Days)	9
Posts (Video)	Number of posts of the video type (60 Days)	34
Posts (Carousel)	Number of posts of the carousel type (60 Days)	89
Average Hashtags / Post (60 Days)	Average number of hashtags used for posts posted last 60 days	1.0

Kaggle and Instaloader scraping are the main sources of our datasets. The dataset from kaggle was manually scraped by the author from the socialbook website [14]. To process our data to be readable, we first need to format the CSV files. Since the accounts scraped and the data we got from Kaggle have their respective CSV files, the datasets need to be formatted individually. The data from Kaggle is cleaned by removing unneeded columns that are not used in the analysis. It is also added with additional attributes, to match the structure of the scraped data. For example, data from the last 60 days are added to the Kaggle data. Then, both csv files are merged, and the outliers are removed from the data by using the Z-score method.

The full process can be seen in the following image:



Kaggle dataset link: https://www.kaggle.com/datasets/syedjaferk/top-200-instagrammers-data-cleaned

Codes for scraping:

```python
import instaloader
from datetime import datetime, timedelta
import time
import random

L = instaloader.Instaloader()

def login_and_refresh_session():
    try:
        L.load_session_from_file(@username)
        time.sleep(random.uniform(60, 100))
    except FileNotFoundError:
        L.login(@username, @password)
        L.save_session_to_file()
        time.sleep(random.uniform(60, 100))

def random_delay2():
    time.sleep(random.uniform(60, 80))

def random_delay():
    time.sleep(random.uniform(30, 40))

# Get Hashtag Dictionary, average hashtags, follower count, post count (60 Days), and average likes from the last 60 days

sixty_days = timedelta(days=60)
now = datetime.now()
sixty_days_ago = now - sixty_days

usernamesx = [@target_usernames]

login_and_refresh_session()

for username in usernamesx:

    hashtags_dict = {}
    hashtags_count = 0
    max_pin_count = 0
    posts_count = 0
    total_likes = 0

    profile = instaloader.Profile.from_username(L.context, username)

    for post in profile.get_posts():

        if posts_count % 20 == 0 and posts_count != 0:
            random_delay2()

        if max_pin_count > 3:
            break
        if post.date_utc < sixty_days_ago:
            max_pin_count += 1
            continue

        total_likes += post.likes
        posts_count += 1
        hashtags = post.caption_hashtags
        hashtags_count += len(hashtags)
        for hashtag in hashtags:
            if hashtag not in hashtags_dict.keys():
                hashtags_dict.update({hashtag : 1})
            else:
                hashtags_dict[hashtag] += 1

    try:
        print (f"Username: {username}")
        print(f"Followers: {profile.followers}")
        average_likes = total_likes / posts_count
        print(f"Average Likes: {average_likes}")
        print(f"Posts Count Last 60 Days: {posts_count}")
        average_hashtags = hashtags_count/posts_count
        print(f"Average Hastags : {average_hashtags}, Hashtags Dictionary: {hashtags_dict}")
    except:
        print("Num of hashtags, or posts invalid")
```

Code to get Hashtag Dictionary, average hashtags, Follower count, post count (60 Days), and average likes from the last 60 days

```python
import instaloader
import time
import random

L = instaloader.Instaloader()

def login_and_refresh_session():
    try:
        L.load_session_from_file(#username)
        time.sleep(random.uniform(60, 100))
    except FileNotFoundError:
        L.login(#username, #password)
        L.save_session_to_file()
        time.sleep(random.uniform(60, 100))

def random_delay2():
    time.sleep(random.uniform(60, 80))

def random_delay():
    time.sleep(random.uniform(38, 40))


# Get Average Likes

usernamess = [#Target_Usernames]

for username in usernamess:

    login_and_refresh_session()

    total_likes = 0
    average_likes = 0
    posts_count = 0
    profile = instaloader.Profile.from_username(L.context, username)

    time.sleep(random.uniform(10, 30))

    for post in profile.get_posts():

        total_likes += post.likes
        posts_count += 1

        if posts_count % 10  == 0:
            random_delay2()

        if posts_count % 1000 == 0 and posts_count != 0:
            time.sleep(random.uniform(100, 200))
            continue

        if posts_count %  500 == 0 and posts_count != 0:
            time.sleep(random.uniform(40, 60))
            continue

    average_likes = total_likes / posts_count

    print(f"Username: {username}, Average likes: {average_likes}, Total likes: {total_likes}, Posts Count: {posts_count}")
```

Code to get average likes of all time and total post count

```python
import instaloader
import time
import random  # Import the random module
from itertools import islice

# Initialize Instaloader
L = instaloader.Instaloader()

# Attempt to load the session from a file
try:
    L.load_session_from_file("@username")  # Ensure you've saved a session previously
except FileNotFoundError:
    # If the session file doesn't exist, log in to create one
    L.login("@username", "#password")  # Replace with your credentials
    L.save_session_to_file()  # Save the session for future use

# Define the username of the target profile
username = [""]#target account

try:
    # Load the profile
    profile = instaloader.Profile.from_username(L.context, username)

    # Initialize a counter for post numbers
    post_number = 1

    # Initialize a dictionary to accumulate post type counts
    post_type_counts = {
        "Image": 0,
        "Video": 0,
        "Carousel": 0,
        "Reel": 0
    }

    # Loop through the last 50 posts
    for post in islice(profile.get_posts(), 274):
        try:
            print(f"Post Number: {post_number}")  # Print post number
            print(f"Post ID: {post.shortcode}")

            # Determine the post type
            if post.typename == "GraphImage":
                post_type_counts["Image"] += 1
            elif post.typename == "GraphVideo":
                post_type_counts["Video"] += 1
            elif post.typename == "GraphSidecar":
                post_type_counts["Carousel"] += 1
            elif post.is_video and post.video_duration <= 90:
                post_type_counts["Reel"] += 1

            # Introduce a random delay between 2 to 5 seconds
            delay = random.uniform(2, 5)  # Change the range as needed
            print(f"Waiting for {delay:.2f} seconds...")
            time.sleep(delay)

            post_number += 1  # Increment the post number

        except instaloader.exceptions.QueryReturnedBadRequestException as e:
            print(f"Error occurred for post {post.shortcode}: {e}")
            continue  # Skip this post and continue with the next one

    # Print the accumulated counts of each post type
    print("\nPost Type Counts:")
    for post_type, count in post_type_counts.items():
        print(f"{post_type}: {count}")

except instaloader.exceptions.ProfileNotExistsException:
    print("Profile does not exist.")
except instaloader.exceptions.ConnectionException as e:
    print(f"Connection error: {e}")
except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

Code to get post types (image, video, or carousel)

The scraping codes can be accessed through Github:
https://github.com/gamakagami/FoDS-FinalProject/tree/main/data%20scraping

IV. Models and Techniques

We used five supervised machine learning techniques to analyze the dataset:

- Gradient Boosting Regressor constructs a series of decision trees with the aim of reducing the errors made by the previous ones and sum up the outputs from all the trees [15]. It was chosen for its high efficiency in dealing with complex relationships and high accuracy.
- XGBoost is an optimized distributed gradient boosting model that works by building decision trees sequentially [16]. It was chosen for its efficiency and ability to handle complex relationships.
- Random Forest constructs several decision trees using a random subset of data and averages their predictions to reduce overfitting [17]. It was chosen for its ability to handle both linear and non-linear relationships.
- CatBoost is a variant of gradient boosting that scales the data, has built in cross validation, and is robust to overfitting [18]. It was chosen for its high efficiency and ease of use
- K-Nearest Neighbor (KNN) finds the distance between a query and all examples in the data, selecting the specified number of examples closest to the query and averages their labels [19]. It was chosen for its effectiveness in capturing local patterns in the data.

The label used for this dataset is 'Average Engagement Rate', which translates to the average engagement rate for a post. All the features used to predict the label are continuous variables, we used 'Followers', 'Average Likes', 'Average Comments', 'Engagement Rate (60 Days)', 'Posting Frequency (60 Days)', 'Posts (Image)', 'Posts (Video)', 'Posts (Carousel)'.

Libraries used

Instaloader- Tool to scrape and download instagram data

Numpy- Numerical computing library to fill missing values

Sklearn- Library to apply machine learning models

Pandas- Library to clean, transform, and analyze datasets

Matplotlib- Library used for data visualization (graphs)

Seaborn- Library used for data visualization (correlation matrix)

For our features, we tried to use all of them but couldn't effectively implement the hashtags for our final dataset used, so we decided to not use the hashtags. Aside from this, we decided to remove the average hashtags / post in the last 60 days feature as it worsens the results compared to not using it.

Other than the final five models we used, we tried linear regression, decision tree, ridge regression models. We ended up not using these models because the result wasn't maximal. This may be due to the compatibility of the model with our dataset as our dataset doesn't have a linear relationship which is incompatible with linear regression and ridge regression. Additionally, decision trees are prone to overfitting on this dataset as it may be imbalanced.

V. Evaluation Method

To evaluate which machine learning produces the best results, several evaluating techniques are used. The evaluation technique includes Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R2 score.

1) Mean Squared Error (MSE): Mean squared error represents the average squared difference between the predicted values and the actual values [20].

2) Root Mean Squared Error (RMSE): Root mean squared error is the square root of the variance of the residuals, where the residuals represent the difference in the predicted values and the actual values of a model [21].

3) R2 score: R2 score tells how well independent variables in a statistical model explain the variation in the dependent variable, in a regression model. It is calculated by conducting a regression analysis to find the best fit line (predicted values),which is subtracted by the actual values, and squared [22].

These evaluation metrics provide a balanced evaluation and insight of model performance, capturing the error values and the proportion of the data explained.

Splitting dataset

We split our dataset using the train_test_split function from the sklearn library, where we split the data in an 80:20 ratio, where 80% of the data are used for training, while 20% are used for testing.

Evaluation Result

Before having our final dataset (the merged one), we did some experimentation with the data by splitting it into categories (mega, macro, etc.) and evaluating each of them.

Global (accounts from over the world):

Data	Linear regression	KNN	Xgboost	Random forest	Decision Tree	Gradient Boosting
Mega (195 rows)	R^2: 0.92 MSE:4.342	R^2: 0.77 MSE:13.29	R^2: 0.97 MSE:1.672	R^2: 0.83 MSE:9.53	R^2: 0.96 MSE:2.258	R^2: 0.85 MSE:8.317
Macro (139 rows)	R^2: 0.69 MSE:33.44	R^2: 0.80 MSE:21.72	R^2: 0.82 MSE:19.57	R^2: 0.79 MSE:22.71	R^2: 0.75 MSE:26.91	R^2: 0.83 MSE:19.08
Micro (99 rows)	R^2: 0.91 MSE:9.178	R^2: 0.84 MSE:15.55	R^2: 0.92 MSE:7.930	R^2: 0.81 MSE:18.89	R^2: 0.60 MSE:38.49	R^2: 0.90 MSE:9.763
Nano (102 rows)	R^2: 0.82 MSE:2.506	R^2: 0.50 MSE:6.833	R^2: 0.96 MSE:2.258	R^2: 0.87 MSE:1.789	R^2: 0.96 MSE:0.500	R^2: 0.88 MSE:1.665
Global (merged) (532 rows)	R^2: 0.26 MSE:36.97	R^2: 0.81 MSE:9.257	R^2: 0.59 MSE:20.41	R^2: 0.62 MSE:18.86	R^2: 0.63 MSE:18.32	R^2: 0.70 MSE:15.07
Mega (with hashtags) (195 rows)	R^2: 0.92 MSE:4.347	R^2: 0.77 MSE:13.29	R^2: 0.97 MSE:1.678	R^2: 0.83 MSE:9.541	R^2: 0.96 MSE:2.297	R^2: 0.81 MSE:10.89
Macro (with hashtags) (139 rows)	R^2: 0.72 MSE:31.18	R^2: 0.80 MSE:21.72	R^2: 0.83 MSE:18.68	R^2: 0.74 MSE:28.01	R^2: 0.93 MSE:7.782	R^2: 0.83 MSE:19.16
Micro (with hashtags) (99 rows)	R^2: 0.91 MSE:8.828	R^2: 0.84 MSE:15.55	R^2: 0.92 MSE:8.077	R^2: 0.84 MSE:15.10	R^2: 0.49 MSE:49.79	R^2: 0.82 MSE:17.51
Nano (with hashtags) (102 rows)	R^2: 0.76 MSE:3.380	R^2: 0.73 MSE:3.754	R^2: 0.88 MSE:1.675	R^2: 0.88 MSE:1.685	R^2: 0.78 MSE:3.084	R^2: 0.96 MSE:0.560
Global (merged) (with hashtags) (532 rows)	R^2: 0.26 MSE:36.79	R^2: 0.55 MSE:22.54	R^2: 0.51 MSE:24.30	R^2: 0.62 MSE:18.77	R^2: 0.63 MSE:18.54	R^2: 0.68 MSE:15.98

Indonesia (accounts from Indonesia):

Data	Linear regression	KNN	Xgboost	Random forest	Decision Tree	Gradient Boostin
Mega (95 rows)	R^2: -0.20 MSE:11.79	R^2: 0.77 MSE:2.245	R^2: 0.66 MSE:3.363	R^2: 0.55 MSE:4.395	R^2: 0.96 MSE:2.258	R^2: 0.78 MSE:2.193
Macro (105 rows)	R^2: 0.84 MSE: 2.523	R^2: 0.73 MSE:4.382	R^2: 0.79 MSE:3.376	R^2: 0.95 MSE:0.802	R^2: 0.38 MSE:10.03	R^2: 0.96 MSE:0.609
Micro (103 rows)	R^2: 0.70 MSE:1.594	R^2: 0.72 MSE:1.475	R^2: 0.61 MSE:2.063	R^2: 0.92 MSE:0.423	R^2: 0.54 MSE:2.412	R^2: 0.90 MSE:0.527
Nano (100 rows)	R^2: 0.60 MSE:165.5	R^2: 0.16 MSE:346.2	R^2: 0.98 MSE:7.133	R^2: 0.96 MSE:17.62	R^2: 0.10 MSE:372.9	R^2: 0.93 MSE:28.18
Indo (merged) (400 rows)	R^2: 0.07 MSE:112.5	R^2: 0.15 MSE:103.1	R^2: 0.22 MSE:93.78	R^2: 0.18 MSE:98.91	R^2: 0.14 MSE:103.9	R^2: 0.37 MSE:75.91
Mega (with hashtags) (95 rows)	R^2: -0.19 MSE:11.67	R^2: 0.77 MSE:2.245	R^2: 0.70 MSE:2.995	R^2: 0.55 MSE:4.436	R^2: 0.51 MSE:4.771	R^2: 0.79 MSE:2.057
Macro (with hashtags) (104 rows)	R^2: 0.76 MSE:3.948	R^2: 0.64 MSE:5.935	R^2: 0.50 MSE:8.266	R^2: 0.86 MSE:2.399	R^2: 0.34 MSE:11.02	R^2: 0.78 MSE:3.643
Micro (with hashtags) (103 rows)	R^2: 0.64 MSE:1.898	R^2: 0.32 MSE:3.561	R^2: 0.68 MSE:1.692	R^2: 0.89 MSE:0.599	R^2: 0.58 MSE:2.228	R^2: 0.84 MSE:0.843
Nano (with hashtags) (100 rows)	R^2: 0.60 MSE:166.2	R^2: 0.23 MSE:317.8	R^2: 0.45 MSE:226.5	R^2: 0.94 MSE:24.89	R^2: 0.10 MSE:372.9	R^2: 0.56 MSE:180.2
Indo (merged) (with hashtags) (399 rows)	R^2: 0.07 MSE:112.5	R^2: 0.14 MSE:103.2	R^2: 0.21 MSE:95.37	R^2: 0.19 MSE:97.76	R^2: 0.17 MSE:100.3	R^2: 0.35 MSE:78.07

The results from these experiments were mostly not good and have small data. After the experimentation, we decided to merge the data immediately instead of continuing in experiment with the data.

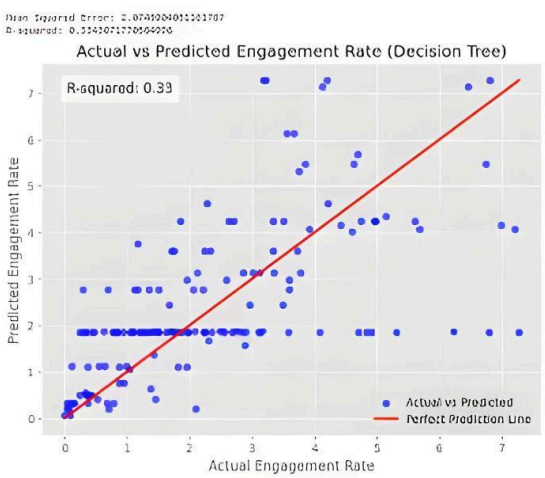
Initially, we used 10 features instead of the final 9 which is 'Followers', 'Average Likes', 'Average Comments', 'Engagement Rate (60 Days)', 'Posting Frequency (60 Days)', 'Posts (Image)', 'Posts (Video)', 'Posts (Carousel)', and 'average hashtags/ Post'. In addition to this, we only tried 6 models initially and experimented with several more models based on our result as the model won't improve no matter what we tried.

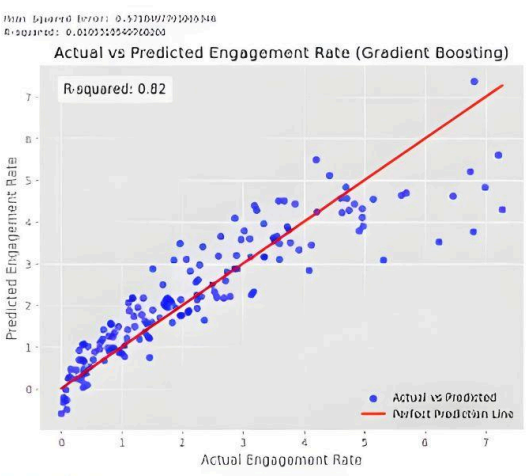
After merging our dataset, we tried removing outliers using the iqr and z-score method on different features. Here are the results:

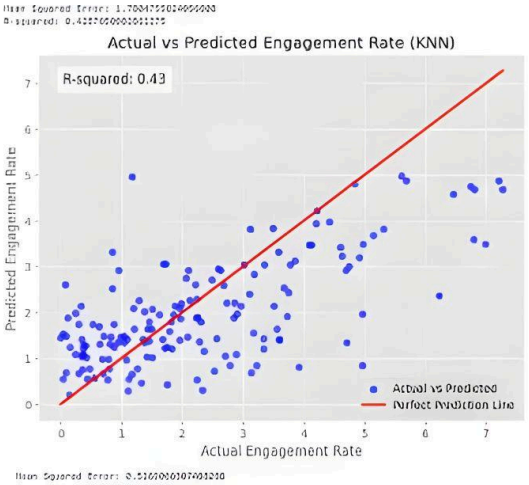
IQR on average engagement rate (793 rows):

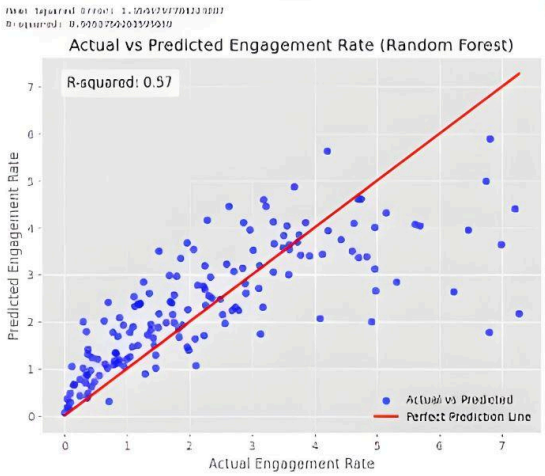
Visualizations: [Decision Trees | Gradient Boosting | KNN]

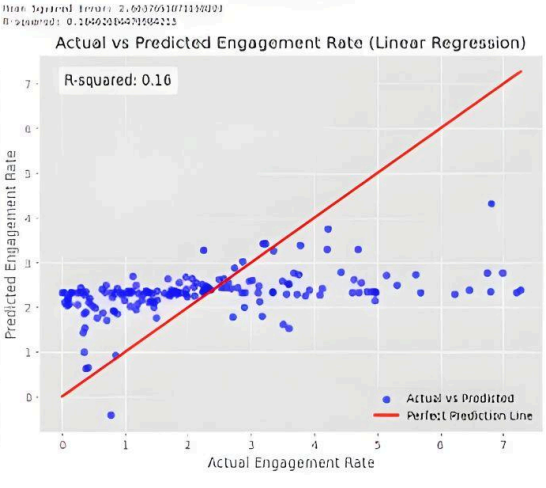
[Random Forests | Linear Regression | XGBoost]

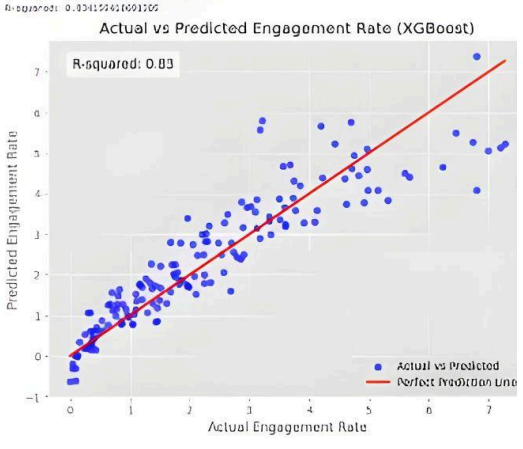








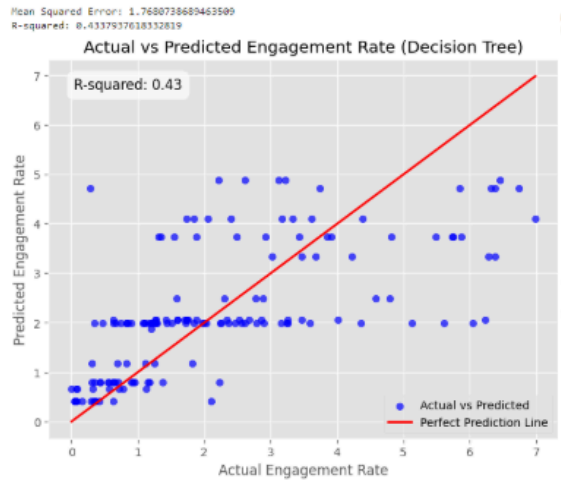


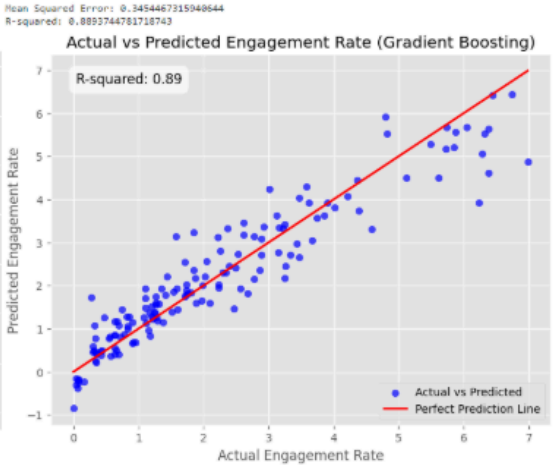


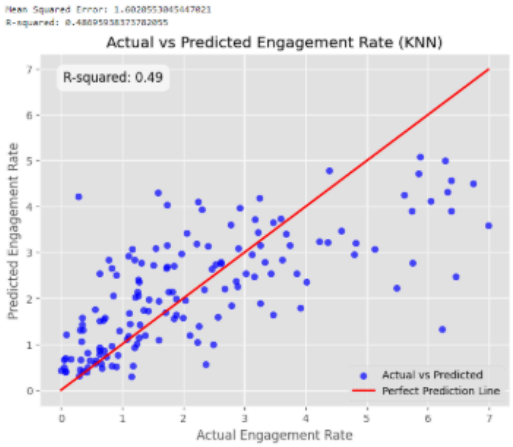
IQR on average engagement rate x2 (755 rows):

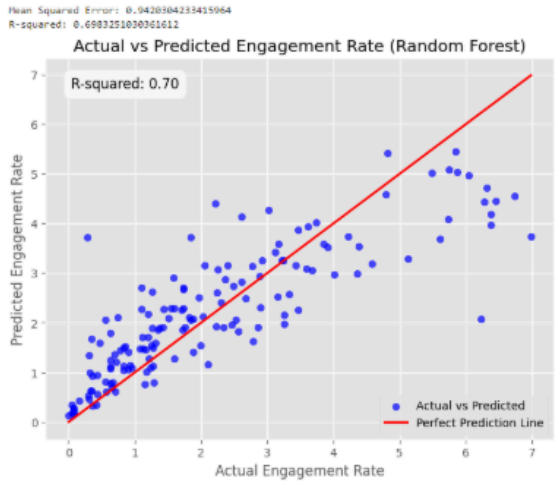
Visualizations: [Decision Trees | Gradient Boosting | KNN]

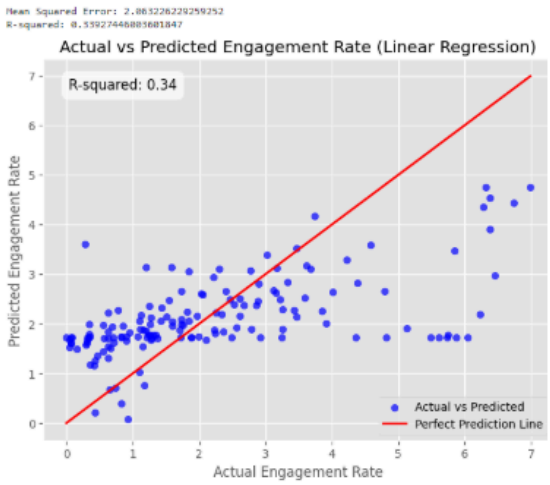
[Random Forest | Linear Regression | XGBoost]

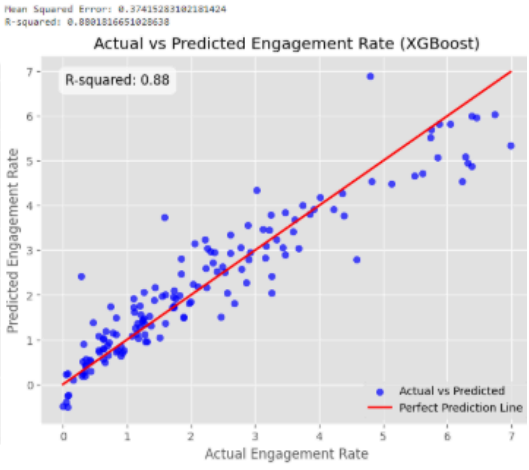








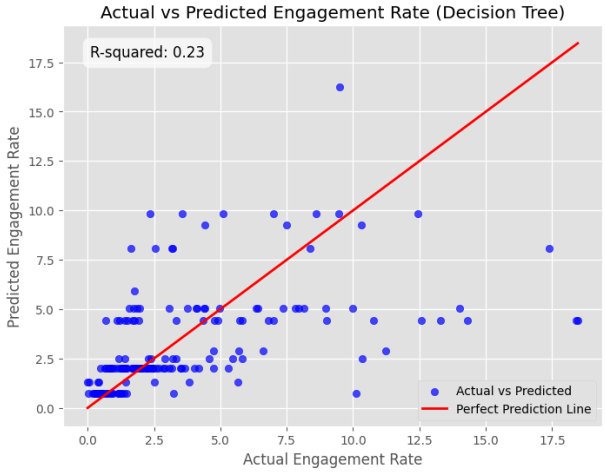




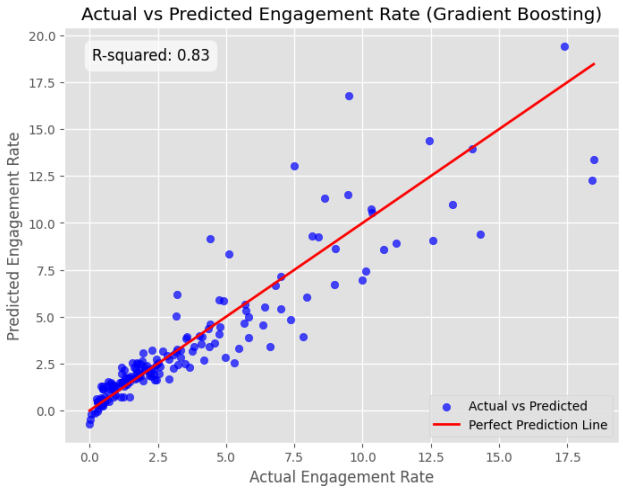
Z - Score on all features (827 rows):

Visualizations: [Decision Trees | Gradient Boosting]
[KNN | Linear Regression]
[Random Forest | XGBoost]

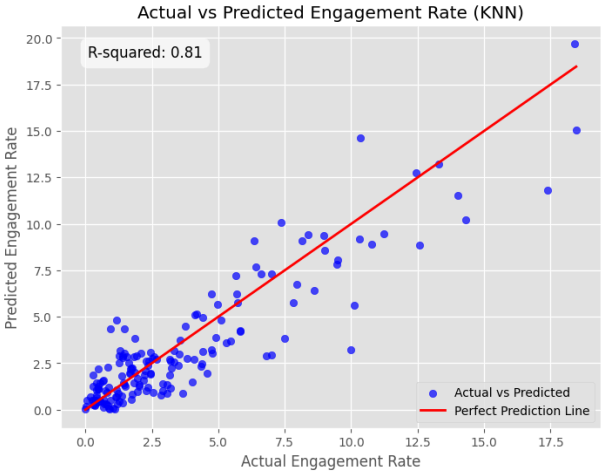
Mean Squared Error: 10.840901022824822
R-squared: 0.22606533480856272



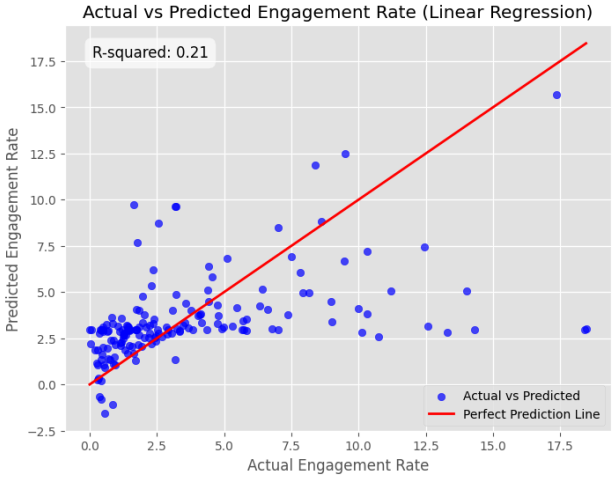
Mean Squared Error: 2.3341168131715984
R-squared: 0.8333668105154459



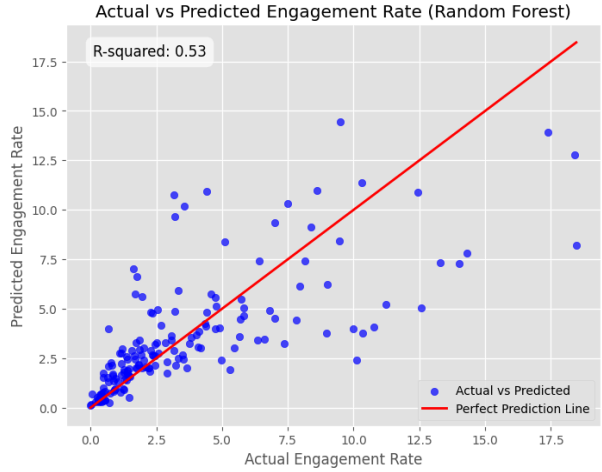
Mean Squared Error: 2.6017567369277104
R-squared: 0.8142599286845043



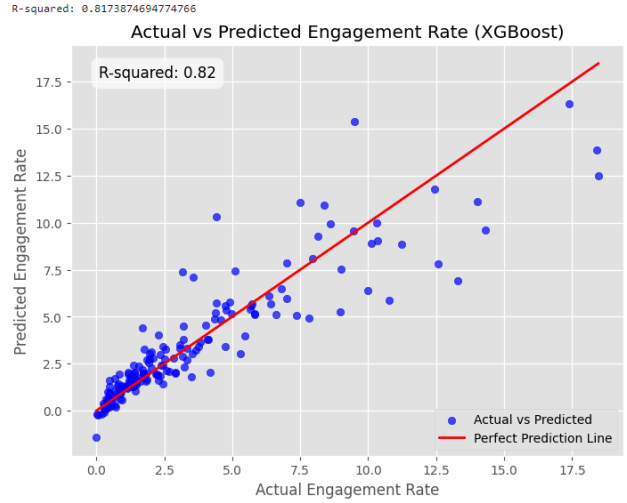
Mean Squared Error: 11.089449615019733
R-squared: 0.20832138796510669



Mean Squared Error: 6.636152427072916
R-squared: 0.5262434002494247



Mean Squared Error: 2.557947664009294
R-squared: 0.8173874694774766



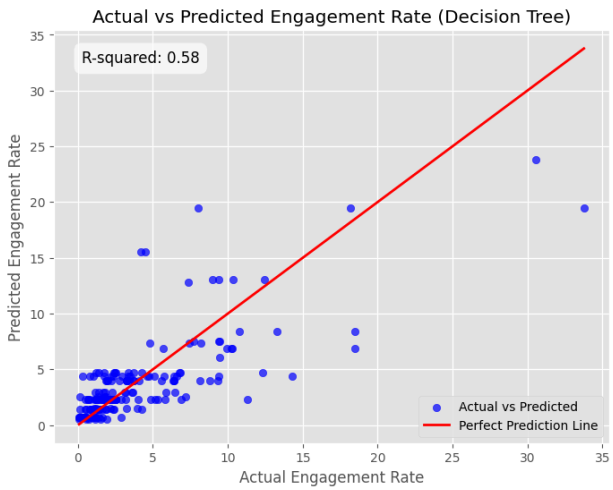
Z - Score on average engagement rate (923 rows):

Visualizations: [Decision Trees | Gradient Boosting]

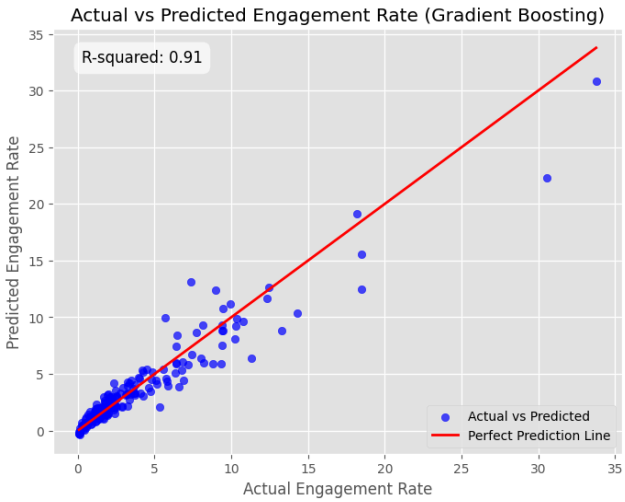
[Random Forests | KNN]

[Linear Regression | XGBoost]

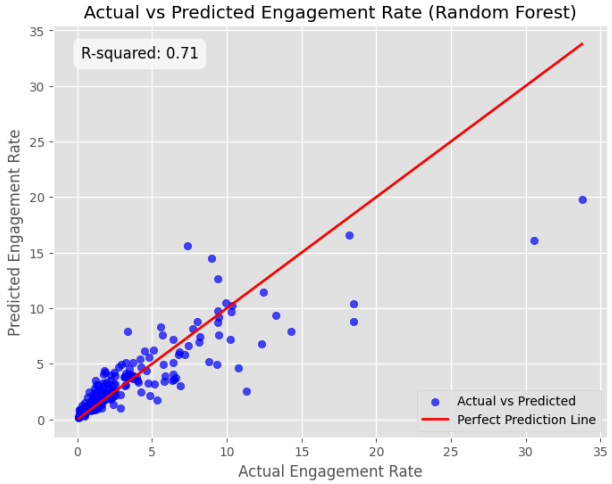
Mean Squared Error: 9.264110339609887
R-squared: 0.5754865544212162



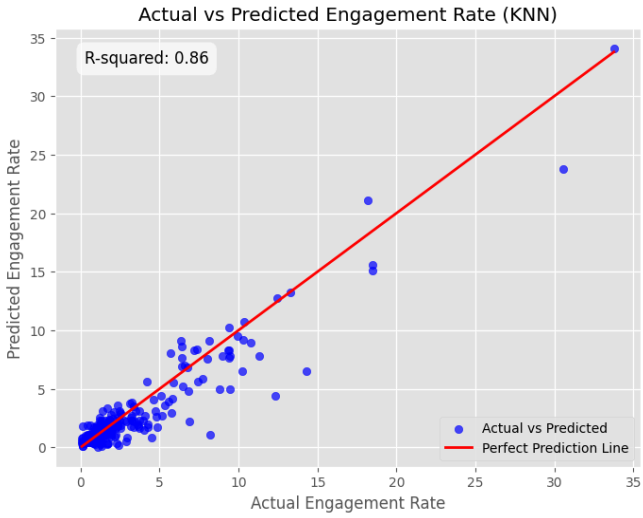
Mean Squared Error: 2.0386818895875463
R-squared: 0.9065805736696009



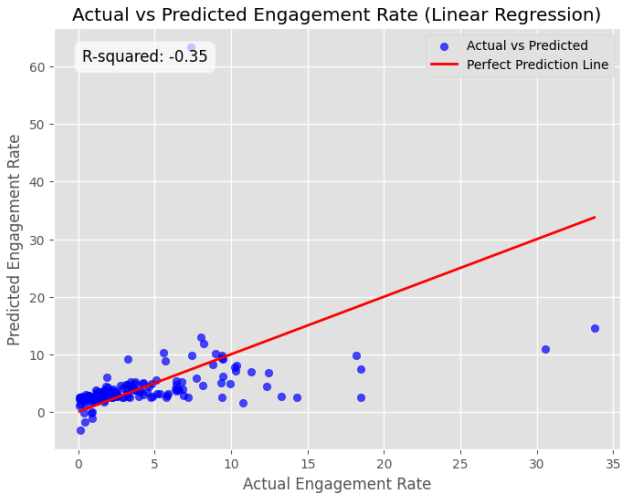
Mean Squared Error: 6.253321026602529
R-squared: 0.7134512912736871



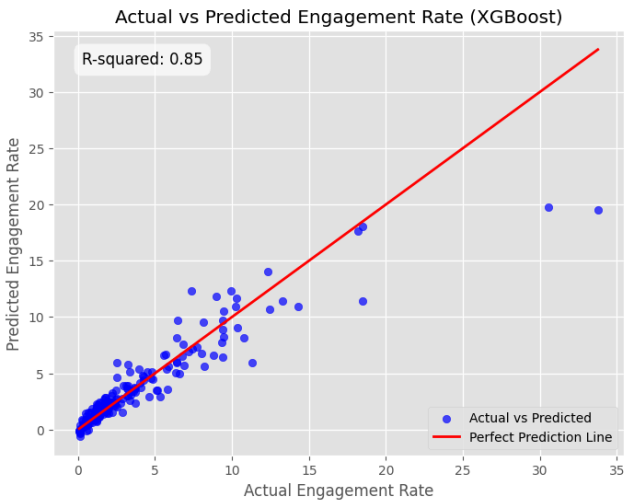
Mean Squared Error: 3.0051785906486486
R-squared: 0.86229236577189



Mean Squared Error: 29.497535473700776
R-squared: -0.35167867835978583



Mean Squared Error: 3.253039388278185
R-squared: 0.8509345302789622

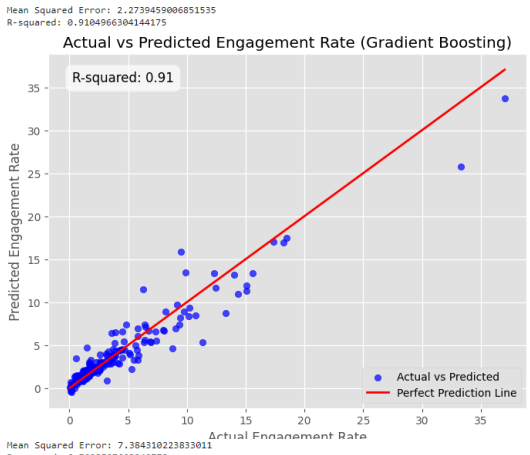


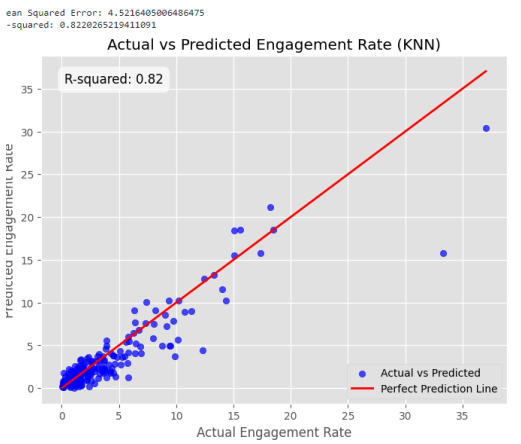
Z - Score on average engagement rate and engagement rate (60 days) (922 rows):

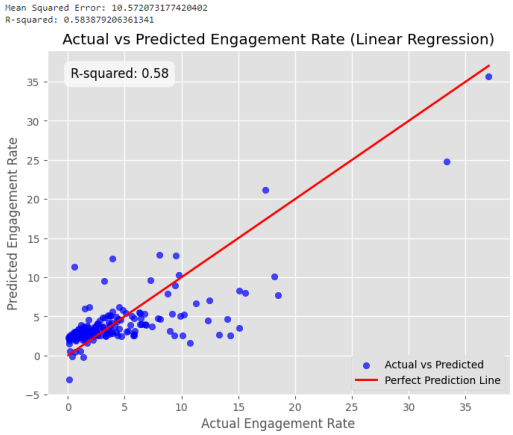
Visualizations: [Gradient Boosting | KNN | Linear Regression]

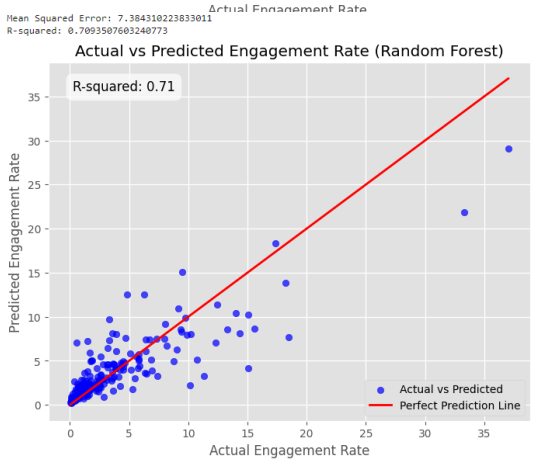
[Random Forest | XGBoost | Ridge Regression]

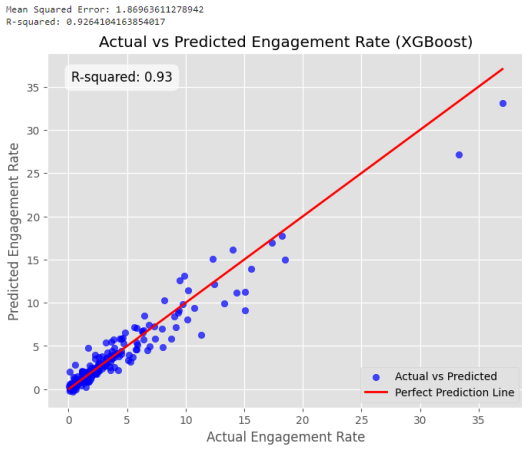
[CatBoost | Decision Trees]

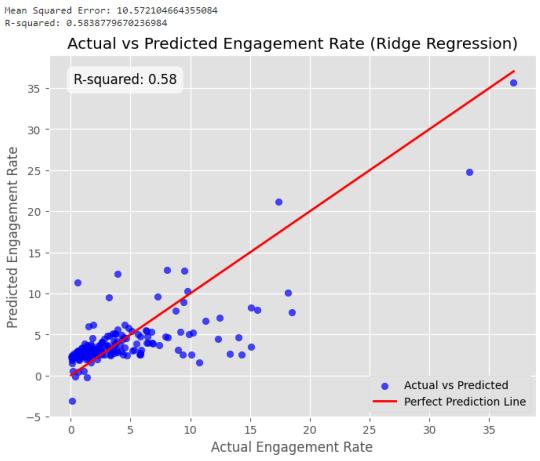


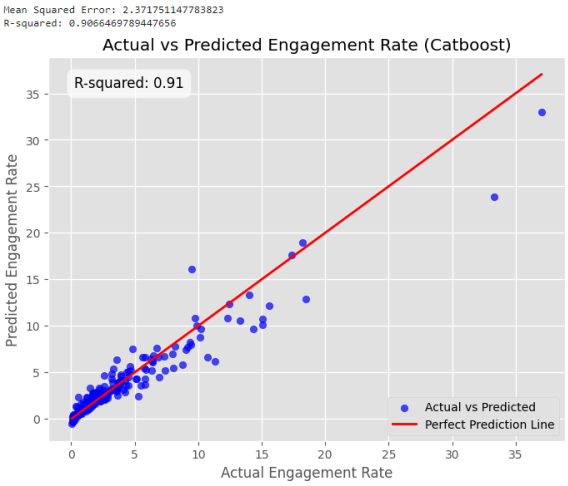


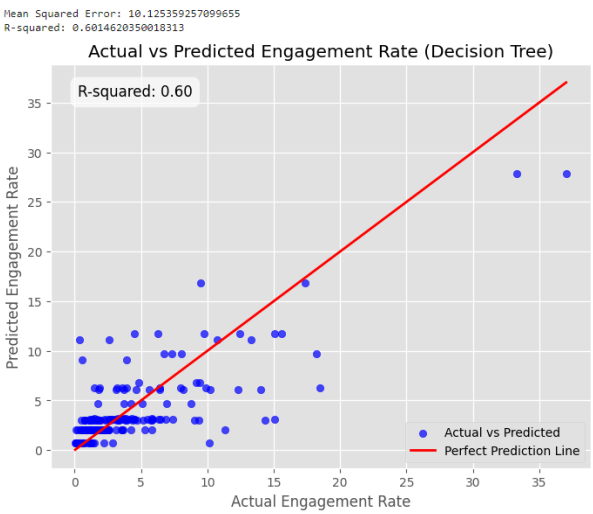










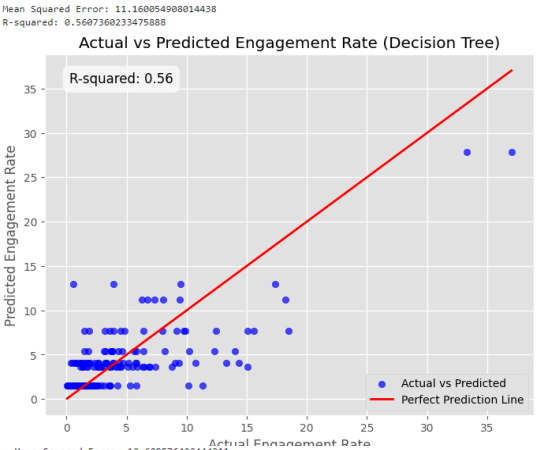


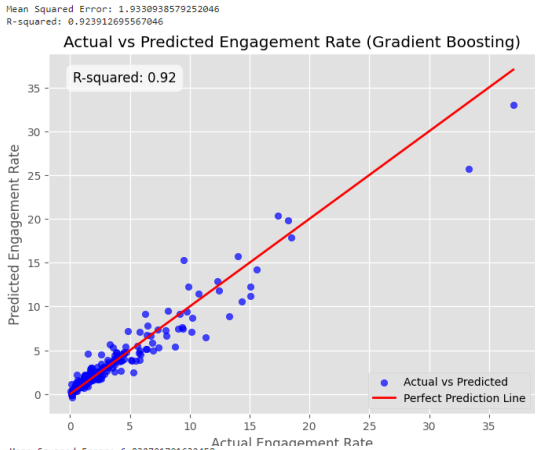
Z - Score on average engagement rate and engagement rate (60 days) with average hashtags/post dropped (Final):

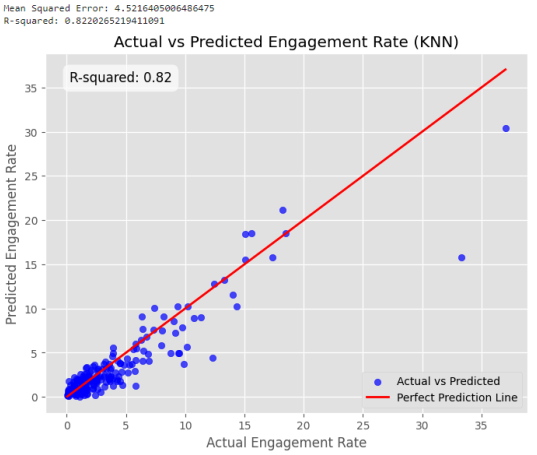
Visualizations: [Decision Trees | Gradient Boosting | KNN]

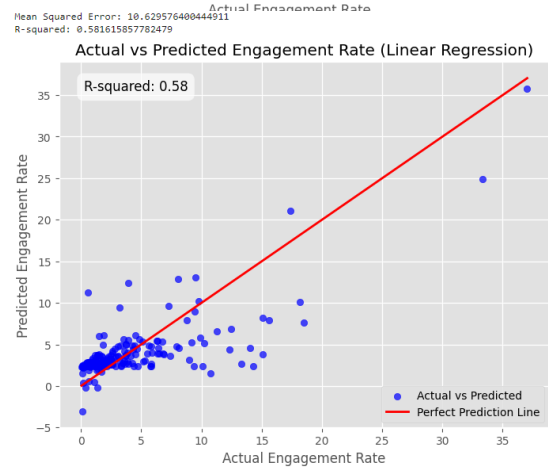
[Linear Regression | Random Forest | Ridge Regression]

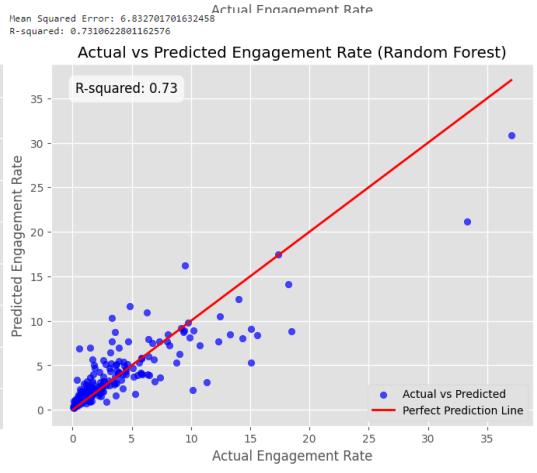
[XGBoost | Catboost]

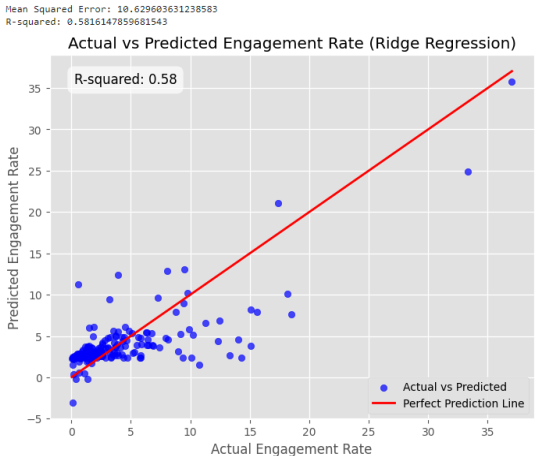


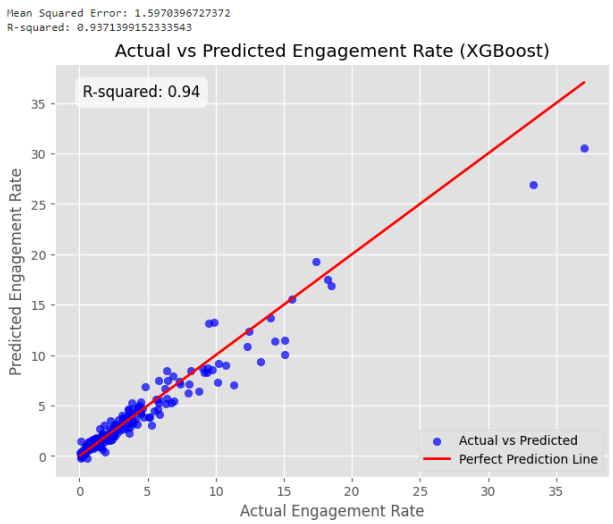


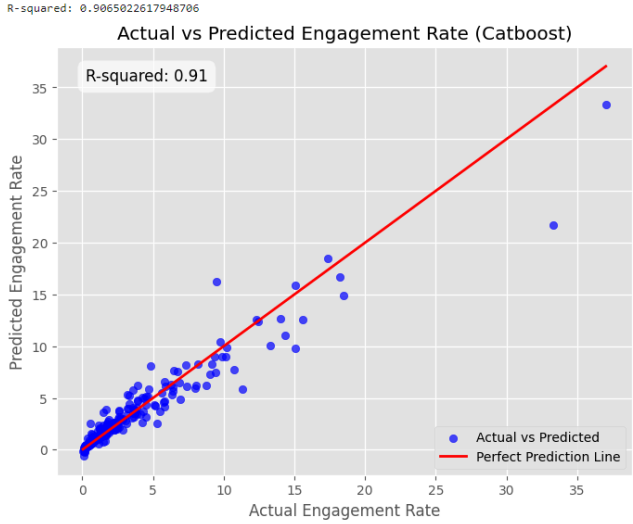












VI. Results and Discussions

From our experimentation results, most of them are not consistent across various models, thus the result is not satisfactory. Separating the dataset into smaller datasets based on each category such as (Mega, Macro, etc.) is not effective as it has less data and reliability compared to merging them, thus we decided to continue the refinement on the merged data only.

To improve the result of our merged data set, we decided to try cleaning it with the iqr and z - score method on several features. After trying to clean the dataset several times and tuning the models, using the z-score method on the average engagement rate and engagement rate (60 days) features produces the best result for several models. Even after this, the results of some models were not improving, we decided to add more models such as CatBoost and Ridge Regression due to our suspicion that the models we used are not compatible with our dataset. The CatBoost model results with a satisfactory result which we suspected as it is a variant of gradient boosting and the gradient boosting regressor as well as the XGBoost model performs well.

From the final result, we can conclude that the **decision tree** model is likely overfitting because decision trees often work by capturing patterns which may not generalize well in this dataset. There is no overfitting or underfitting for the **gradient boosting regressor** model. The **KNN** model is likely underfitting because the model cannot capture complex patterns across the dataset. The **linear regression** model is clearly underfitting as it oversimplifies the relationship between the variables and cannot perform well in a complex dataset. The **random forest** model is likely underfitting as it averages the result from various decision trees and may fail to capture subtle patterns. Like linear regression, the **ridge regression model** is underfitting due to the oversimplification imposed by regularization and the dataset's complexity. There is no overfitting or underfitting for the **XGBoost** model. There is no overfitting or underfitting for the **CatBoost** model.

Final Result used:

MODELS PERFORMANCE

Models	MSE	RMSE	R2
XGBoost	1.60	1.26	0.94
Gradient Boosting	1.93	1.39	0.92
CatBoost	2.38	1.54	0.91
KNN	4.52	2.13	0.82
Random Forest	6.83	2.61	0.73

Compared to the five models, XGBoost performs the best and most accurate in all evaluation aspects compared to other models with an MSE of 1.60, RMSE of 1.26, and R2 of

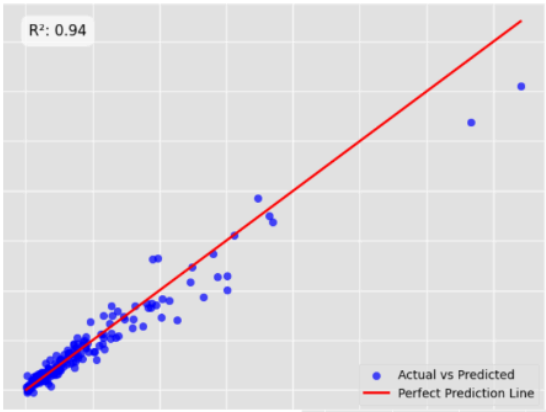
0.94. However, the Random Forest model performed poorly compared to the other models with an MSE of 6.83, an RMSE of 2.61, and an R2 of 0.73.

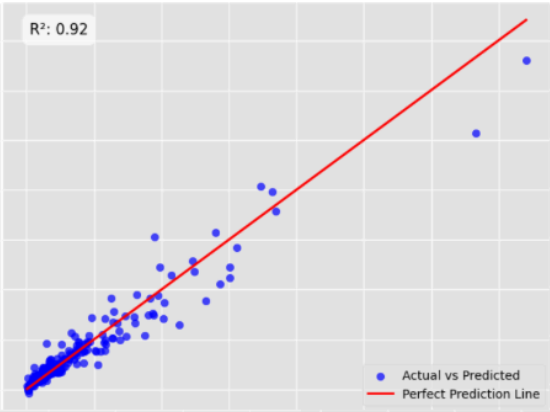
An interesting insight from the result is that the gradient boosting models (Gradient Boosting Regressor, XGBoost, and CatBoost) performed significantly better compared to the other models, indicating that the gradient boosting models are suitable for this task. From this insight, we can conclude that Gradient Boosting models performed superior mainly due to their ability to capture complex, non-linear relationships, and feature interactions.

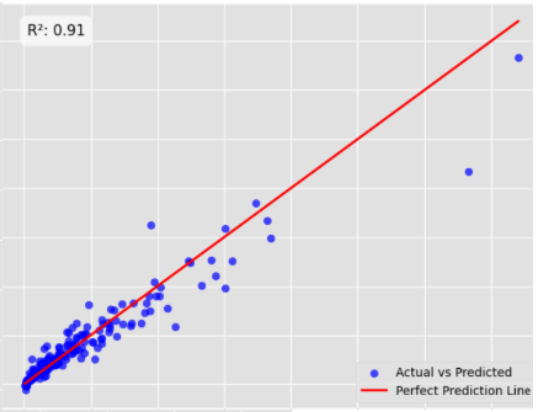
The following visualization visualizes the predicted and the actual values, where the blue points are the actual values, while the red line represents the predicted values.

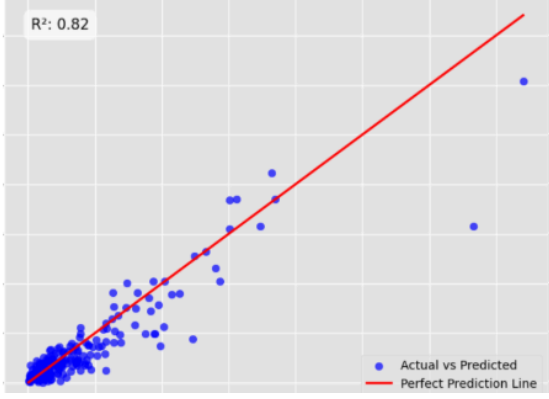
Visualizations: [XGBoost | Gradient Boosting | CatBoost]

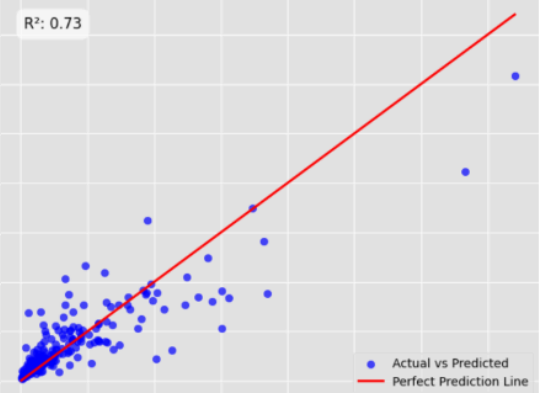
[KNN | Random Forest]

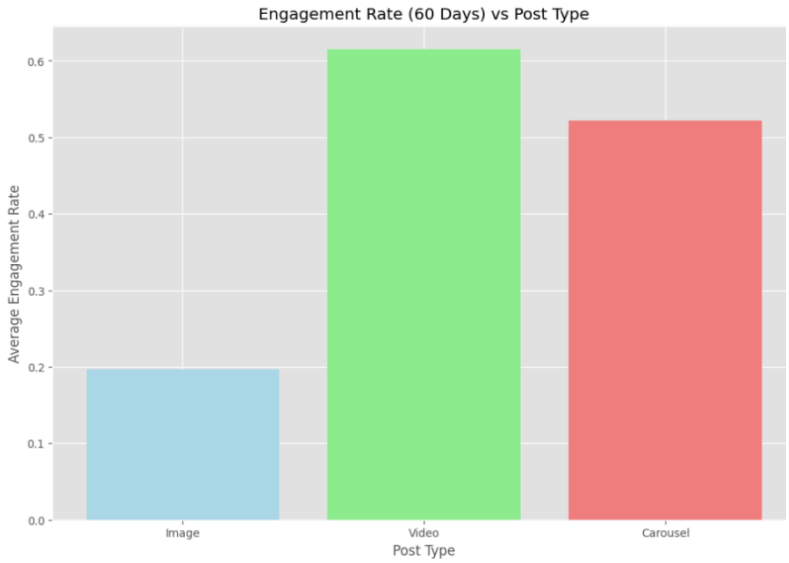




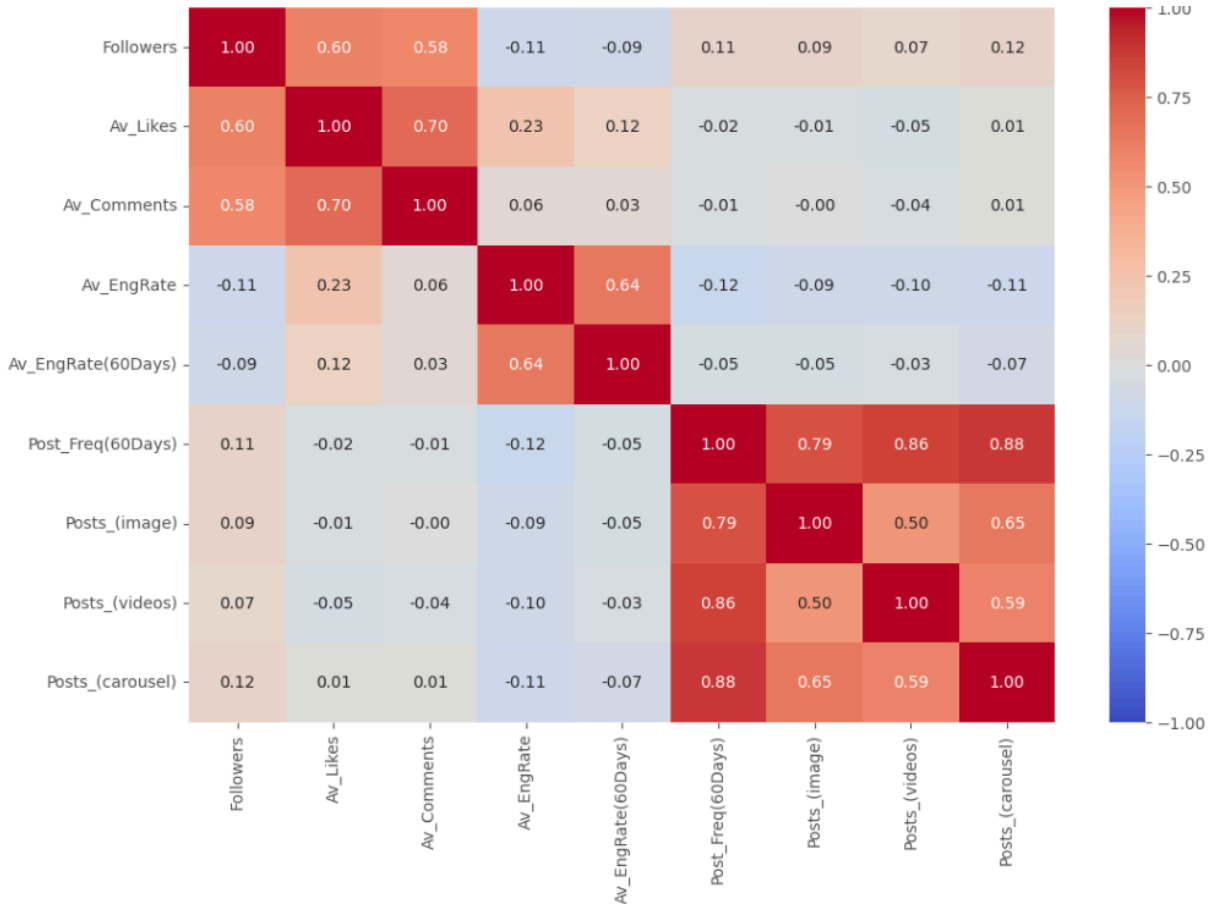




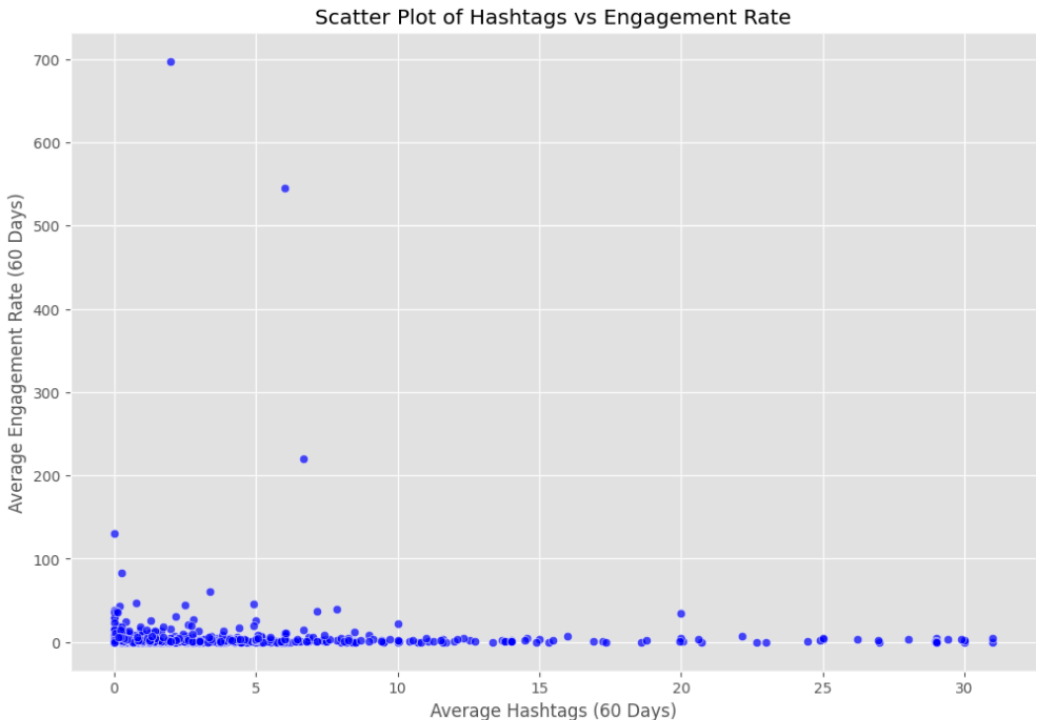




There is also an interesting insight where on median, the video post type received more engagement than other post types. It is calculated by creating new data frames where the post type is not equal to 0, then it is divided by the posting frequency and multiplied to the average engagement rate. Then, the median is calculated, as it is less sensitive to outliers.



From our analysis after cleaning the dataset as shown in the correlation matrix, there is a strong positive correlation between average likes and average comments, indicating that posts with higher likes tend to have more comments. Additionally, there is a moderate positive correlation between engagement rate in the last 60 days and average engagement rate, indicating consistency between the short-term and long-term trends.



This scatterplot visualizes the relationship between the average hashtags used per post with the average engagement rate of the 60 days. It shows that the number of hashtags used have little to no correlation with the engagement rate. Actually, many posts with few hashtags used outperformed the posts with many hashtags.

VII. Conclusion and Recommendation

To conclude, this study aims to predict the engagement rate of a post in instagram using five models: XGBoost, CatBoost, Gradient Boosting Regressor, K-Nearest Neighbor (KNN), and Random Forest.

From our experimentation, we observed inconsistencies in several model performance throughout several approaches. Splitting the dataset into smaller subsets based on categories (Mega, Macro, etc.) proved to be ineffective due to small data which lacks the reliability and size. Thus, we decided to focus the refinement solely on the merged dataset. To improve the result, we tried the iqr and z-score method on several features as well as tuning the models.

Through experimentation, we observed that the decision tree model is likely overfitting as it captures patterns that do not generalize well, while the KNN, Linear Regression, Ridge Regression, and Random Forest models underfit the dataset due to their inability to handle complex relationships effectively.

The XGBoost model performs better compared to the other models with a Root Mean Squared Error (RMSE) of 1.26, indicating that the predicted value is close to the actual value. On the other hand, the Random Forest model performed relatively poorly compared to the other models with a Root Mean Squared Error (RMSE) of 2.61, showing a significant gap compared to the XGBoost model. This might be caused by the model failing to identify subtle patterns. The gradient boosting models (XGBoost, CatBoost, and Gradient Boosting Regressor) the other models, making them the most suitable type for this research. This is likely due to their ability to capture complex, non-linear relationships and feature interactions. Gradient Boosting models iteratively correct errors through boosting, offering higher accuracy and better generalization.

These findings are particularly relevant for brands and influencers who want to optimize their marketing strategies on Instagram. Smaller influencers with a high engagement rate can be cost effective while having a good engagement rate, making them a choice for brands with limited budgets. There is also an interesting insight, where a video post type receives the most engagement, when compared to other post types.

For our future work, we would like to first add more data, as for this research we have only used the data from one social media platform which is Instagram. We would like to add other social media platforms such as Tiktok or X. Next, we would like to add the data volume as our data after cleaning is 922 accounts and as we all know the more the data the better the outcome. We would also like to find a better and more reliable scraper as Instaloader is unstable and costs

us time that we could have used to add more data. Predicting the engagement rate in real time is needed for our research to have better accuracy as followers, likes, and comments change from time to time. We would implement the hashtag dictionary for machine learning as for the research as we did not implement it.

VIII. References

[1] A. A. Arman and A. P. Sidik, "Measurement of engagement rate in instagram (case study: Instagram indonesian government ministry and institutions)," in 2019 International Conference on ICT for Smart Society (ICISS), vol. 7, pp. 1–6, IEEE, 2019.

[2] A. Radu, "Social media statistics you should know in 2024," 2024. Available: https://socialbee.com/blog/social-media-statistics/.

[3] S. Larson, "Social media users 2024 (global data & statistics)," 2024. Available: https://prioridata.com/data/social-media-usage/.

[4] Backlinko, "Instagram statistics: Key demographic and user numbers," 2024. Available: https://backlinko.com/instagram-users.

[5] N. Kumar, "How many people use instagram 2024 [new data]," 2024. Available: https://www.demandsage.com/instagram-statistics/.

[6] K. Kuligowski, "12 reasons to use instagram for your business," 2024. Available: https://www.business.com/articles/10-reasons-to-use-instagram-for-business/.

[7] J. Zote, "Instagram statistics you need to know for 2024 [updated],"] 2024. Available: https://sproutsocial.com/insights/instagram-stats/users.

[8] I. Media, "Quality vs. quantity in social media," 2018. Available: https://inklingmedia.net/quality-vs-quantity-in-social-media/.

[9] M. Trunfio and S. Rossi, "Conceptualising and measuring social media engagement: A systematic literature review," 2021. Published in the Italian Journal of Marketing, Volume 2021, Pages 267–292, DOI: 10.1007/s43039-021-00035-8

[10] B. Schivinski, G. Christodoulides, and D. Dabrowski, "Measuring consumers' engagement with brand-related social-media content: Development and validation of a scale that identifies levels of social-media engagement with brands," Journal of Advertising Research, vol. 56, no. 1, pp. 1–18, 2016.

[11] A. Shahzad, H. Rashid, A. Nadeem, M. Bilal, and W. Ahmad, "Social media influencer marketing: Exploring the dynamics of follower engagement," Journal of Policy Research, no. 4, pp. 1–8, 2023.

[12] W. B. Tan and T. Lim, "A critical review on engagement rate and pattern on social media sites," in Proceedings of the International Conference on Digital Transformation and Applications (ICDXA 2020), TARUMT, TARUC, January 2020.

[13] H. A. Putranto, D. P. S. Setyohadi, T. Rizaldi, E. S. J. Atmadji, H. Y. Riskiawan, and I. H. Nuryanto, "Measurement of engagement rate on instagram for business marketing (case study msme of dowry in jember)," 2022. Available: https://www.researchgate.net/publication/366171806 Measurement of Engagement Rate on Instagram for Business Marketing Case Study MSME of Dowry in Jember.

[14] Socialbook, "Top 200 instagrammers," n.d. Available: https://socialbook.io/instagram-channel-rank/top-200-instagrammers.

[15] S. Baladram, "Gradient boosting — towards data science," November 2024. Available: https://towardsdatascience.com/gradient-boosting-regressor-explained-a-visual-guide-with-code-examples-c098d1ae425c

[16] NVIDIA, "Xgboost." https://www.nvidia.com/en-us/glossary/xgboost/,n.d.

[17] GeeksforGeeks, “Random forest algorithm in machine learning,” 2024. Available: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/.
[18] GeeksforGeeks, “Catboost in machine learning,” 2024. Available: https://www.geeksforgeeks.org/catboost-ml/.
[19] O. Harrison, “Machine learning basics with the k-nearest neighbors algorithm,” 2019. Available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
[20] GeeksforGeeks, “Mean squared error,” August 2024. Available: https://www.geeksforgeeks.org/mean-squared-error/.
[21] Z. Bobbitt, “How to interpret root mean square error (rmse),” May 2021. Available: https://www.statology.org/how-to-interpret-rmse/.
[22] J. Fernando, “R-squared: Definition, calculation, and interpretation,” November 2024. Available: https://www.investopedia.com/terms/r/r-squared.asp.

IX. Source Code and contact

Github: https://github.com/gamakagami/FoDS-FinalProject

Albertus Santoso - albertus.santoso@binus.ac.id

Gabriel Anderson - gabriel.anderson@binus.ac.id

Rafael Anderson - rafael.anderson@binus.ac.id

X. Question and Answer

Q: Are there any new insights?

A: Based on the types of posts, the video post type receives the most engagement compared to other posts. Furthermore, it is found that the number of hashtags used have little to no impact towards the engagement rate. Lastly, average engagement rate is non-linear, which means that larger accounts don’t guarantee larger engagement rate.