Table of Contents

0. Instructions to Run the Code:

As part of the submission, seven .py files were submitted. These files contain the code for the models/methods created and used for the regression, clustering, classification, association rules, and text processing techniques. Data preprocessing and EDA was created in each of those files as well. To run each of those .py files, it is required for the .py file and and the “twitter_user_data.csv” file to be in the same directory.

1. Big Data Analytics Project Design:

Following the Big Data Analytics Lifecycle discussed during the lectures (School of Computing and Information Technology, n.d.), a typical big data analytics project is divided into six stages as shown in Figure 1. However, because of the limited resources available for this assignment, only stages 1-5 (inclusive) have been performed.

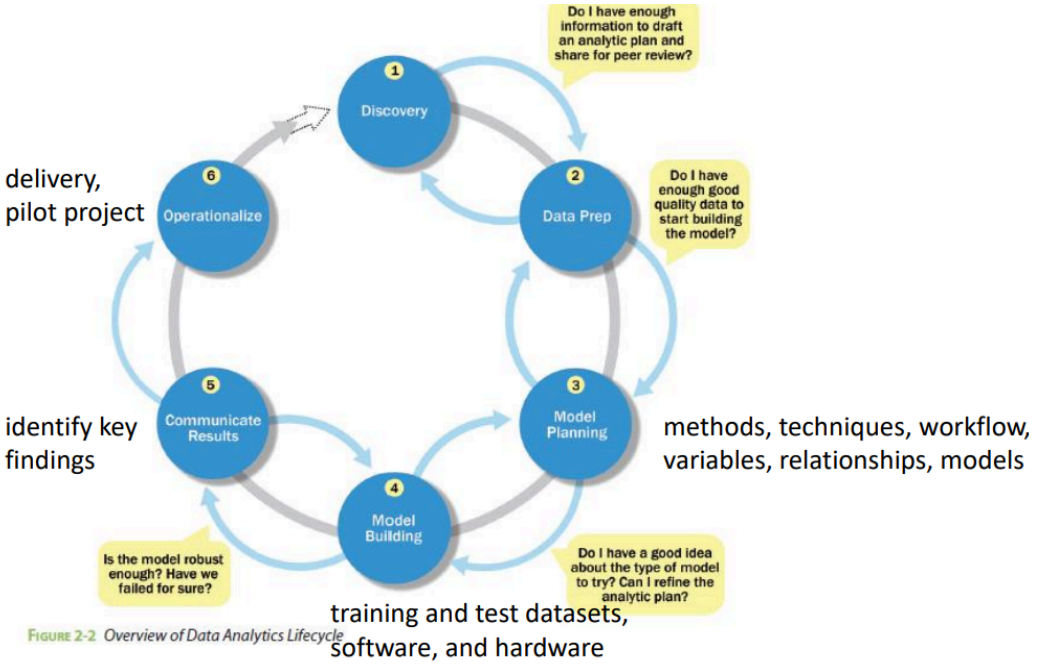

Figure 1: *Data Analytics Lifecycle (School of Computing and Information Technology, n.d.)*

1. **Discovery:**

   - *Identifying the problem:* as per the assignment specifications, the aim of this project is to identify misinformation on social media, particularly Twitter, by identifying profiles on social networks that are misclassified as human or non-human.

   - *Stakeholder engagement:* our stakeholders include the Twitter app human and non human users as they use the application. The application's shareholders, developers, and data analysts are also considered stakeholders as they're directly affected by the results of this project depending on their interest in improving detection algorithms to limit misinformation. Unfortunately, due to the resources available for this project, having discussions with the stakeholders to understand what factors they belive contribute to the misinformation on social media was not conducting.

- *Framing the problem:* the objective of this project is create different machine learning models that can use different factors of the dataset to correctly classify human and non-human profiles, hence correcting a profile type that was wrongly classified as another type. Understanding what factors affect this classification is also an important step in this process.

- *Resources available:* eight data analysts will be working on creating classification, regression, and clustering models as well as association rules and text processing methods within two weeks in order to understand the dataset and solve the project problem.

- *Data source:* as part of the assignment a dataset containing 20,050 rows and 26 fields containing information about users' usernames, random tweets, accounts' profiles and images, location information, and more, was provided to us and hence will be used.

- *Initial Hypothesis:* for this project, We assume that brands (non-human profiles) will generally have longer descriptions than human profiles, filled with promotional content or information about their services. Additionally, it is also assumed that brands will have more likes (favourites) as due to their promotional campaigns or popular posts, they can reach a broader audience. Hence, description and fav_number are two important factors in classifying human and non-human profiles (initial hypothesis).

2. Data Preparation:

- *Data Preprocessing:* following the ETL (Extract, Transform, Load) process, data will be uploaded onto the environment then cleaned to remove noise and missing data before being loaded into the models. Data visualization will also be conducted to further understand the data and what needs to be preprocessed in it.

3. Model Planning:

- *Exploratory data analysis (EDA):* the data will be studied and analysed in order to gain an in-depth understanding of the data types included in the dataset as well as enable us to choose what algorithms and models are suitable for the classification problem. Data visualization will also help in the EDA process by providing visual representations of the dataset.

- *Feature selection:* this will be conducted to identify the factors in the dataset that affect the classification problem in order to achieve greater results. Additionally, depending on the algorithm, some features might be chosen and some might be removed. The selected and removed features are reported for each model with reasons on why they were kept/removed.

- *Choosing algorithms and models:* It was decided to apply regression, classification, clustering, association rules, and text processing techniques on the dataset in order to achieve the best results possible in classifying human and non-human profiles. More details on the exact models chosen in this step are in Sections 2, 3, 4, 5, and 6.

- *Metrics:* different algorithms will use different metrics to evaluate their results depending on the type of the algorithm. More details about this are in Sections 2, 3, 4, 5, and 6.

4. Model Building:

- *Splitting the dataset:* Python will be used to build the models and the dataset will be split for training and testing using different ratios for the different algorithms in order to view which ratio would achieve better results.

- *Overlap:* there might be a bit of an overlap between this stage and the “Model Planning” stage as well as the “Communicate Results” stage depending on the results of the models.

5. Communicate Results:

- *Analyse the results:* present the outputs of the models with metrics chosen in the “Model Planning” stage while stating the key findings and insights deriven from those results. Additionally, use data visualization techniques to provide visual illustrations of the results. Lastly, state whether the hypothesis has been accepted or rejected based on the results.

- *Suggestions:* provide suggestions for future work based on the results. These are stated in Section 8.

2. Regression:

2.1. Algorithm Consideration:

This report focuses on identifying profiles potentially misclassified as human or non-human on social networks using the “Twitter User Data”. The goal is to build a regression model to predict confidence in gender classification based on features such as tweet count, retweet count, and other profile characteristics.

Why We Used the Regression Model for This Problem:

In this assignment, we are working with the ‘gender’ feature, which is a continuous numerical variable representing the confidence level (between 0 and 1) of the gender classification for Twitter users. Given that we are attempting to predict a continuous value, regression models are naturally suited for this type of task. Let’s break down the reasoning further:

1. **Nature of the Target Variable:**
    - *Continuous Data:* The ‘gender’ feature is a continuous numerical value that can range from 0 to 1. This means that we are not dealing with discrete categories (such as male/female/brand), but rather a confidence score that reflects how certain the model is about the user's gender classification.
    - *Regression Task:* Since regression models are designed to predict continuous variables, this makes them the appropriate choice for predicting confidence scores like ‘gender’. Unlike classification, where we predict categorical labels, regression models predict numerical values that can take on any value within a range, which is exactly what we need here.
2. **Ability to Capture Relationships Between Features:**
    - *Linear Relationships:* The Linear Regression model assumes a linear relationship between the independent variables (e.g., tweet count, retweet count, profile description length) and the dependent variable (gender). While this is a simple approach, it helps to understand if there is a straightforward, linear relationship between the features and the target variable.

- *Non-linear Relationships:* By using more advanced regression models like Random Forest Regression and Gradient Boosting, we can capture non-linear relationships in the data. These models allow us to understand more complex interactions between features and the target variable, which may not be captured by a linear regression model. For instance, the interaction between tweet count and profile description length may not be linear, but Random Forests and Gradient Boosting can model such complex relationships.

3. Understanding Confidence in Gender Prediction:

- *Confidence as a Key Metric*: The 'gender' score is critical because it provides insight into how confident we are in classifying a user's gender. Predicting this score helps us evaluate the uncertainty in the gender classification process. A high confidence score (close to 1) means the model is highly certain about the classification, while a low score indicates uncertainty. Using a regression model allows us to predict and interpret these continuous confidence scores effectively.

4. Feature Contribution and Interpretability:

- *Feature Importance:* Regression models, particularly Random Forest Regression and Gradient Boosting, provide insight into feature importance. This is crucial for understanding which features (e.g., tweet count, retweet count, description length) have the most significant impact on the model's confidence in predicting gender. For example, we can assess whether users who tweet more frequently or have longer descriptions lead to higher confidence in gender classification.

- *Interpretability:* Regression models, especially Linear Regression, offer interpretability in terms of how each feature contributes to the predicted confidence score. For instance, in Linear Regression, the model coefficients provide a clear understanding of how changes in features like tweet count or retweet count influence the confidence score. Although Random Forest and Gradient Boosting are more complex, they still provide feature importance rankings, allowing us to understand the key drivers behind the model's predictions.

5. Model Selection Based on Performance:

- *Comparing Different Regression Models:* In this assignment, we used multiple regression models, starting with Linear Regression as a baseline model to understand the general trend and relationship between features and 'gender'. We

then moved to more complex models like Random Forest Regression and Gradient Boosting, which tend to perform better with larger datasets and non-linear relationships.

- *Justification for Using Multiple Models:*
    - **Linear Regression**: Serves as a foundational model to identify if a basic linear relationship exists between the features and the target variable.
    - **Random Forest Regression**: Allows us to handle non-linearity and interactions between features more effectively. It also helps in reducing overfitting by averaging multiple decision trees.
    - **Gradient Boosting**: This model is used to further enhance accuracy by sequentially improving upon the errors made by the previous models, thus fine-tuning predictions.

6. **Applicability to the Problem:**

- *Predicting Confidence, Not Just Labels:* While classification models would predict whether a user is male, female, or a brand, regression models allow us to go deeper by predicting how confident we are in that classification. This is important because, in real-world applications, we not only want to know what the prediction is but also how confident we are in that prediction. The regression model gives us this level of granularity.
- *Use Case in Social Media Analytics:* In social networks like Twitter, it's not always sufficient to classify a user's gender with a binary or categorical output. Social media platforms benefit from knowing the confidence level in predictions to better handle uncertainty and improve the model's robustness. For example, if a model predicts low confidence in classifying a user, additional scrutiny or manual review might be needed. Regression models are particularly useful in this context because they provide a quantitative measure of confidence.

2.2. Data Preprocessing:

In order to perform preprocessing on the provided dataset, we first need to import it in the form of a Pandas dataframe. We also need to specify the encoding type as a parameter

while calling the pd.read_csv method (Pydata.org, n.d). The code to perform the above step is as follows:

```
import pandas as pd
df = pd.read_csv("twitter_user_data.csv", encoding="ANSI")
```

2.2.1. Dropping columns:

Several features were removed from the imported dataset because they did not provide useful information for the analysis. The dropped features were - `'_unit_id'`, `'name'`, `'profileimage'`, `'sidebar_color'`, `'link_color'`, `'name'`, `'gender_gold'`, `'profile_yn_gold'`. Removing these columns will help create more accurate models for the studied phenomena (Gupta, 2024).

2.2.2. Missing Values:

One of the key steps in machine learning is to address missing data. Neglecting missing values can result in skewed or inaccurate outcomes from machine learning models. Therefore, it's important to either fill in missing data or remove it from the dataset. (Olmez, 2024). In this preprocessing step, we filled in the missing values for 3 features which were - `'gender'`, `'description'` & `'gender:confidence'`. The code for this is provided below:

```
df['gender'].fillna(df['gender'].mode()[0], inplace=True)  # Fill 'gender' with
the mode (most frequent value)
df['description'].fillna('NA', inplace=True)  # Fill 'description' with 'NA'
df['gender:confidence'].fillna(df['gender:confidence'].mean(), inplace=True)  #
Fill 'gender:confidence' with mean
```

2.2.3. Converting datatypes:

Features 'last_judgement_at' & 'tweet_created' were converted to datetime data types. This aids in interpretability. (Gupta 2024). The code for this step is provided below:

```
# Convert date columns to datetime format
df['_last_judgment_at'] = pd.to_datetime(df['_last_judgment_at'],
errors='coerce')
df['tweet_created'] = pd.to_datetime(df['tweet_created'], errors='coerce')
```

2.2.4. Label encoding:

We used the LabelEncoder module from scikit-learn preprocessing library to encode categorical features into numerical. (Sci-kit learn, 2024). To ensure smooth operation with machine learning libraries, class labels should be expressed as integer values. Even though scikit-learn may convert non-integer labels, it's a good practice to provide them as integers to avoid potential errors (Raschka, 2022). The code for this is as follows:

```
from sklearn.preprocessing import LabelEncoder

# Initialize label encoder
le = LabelEncoder()

# Encode categorical features
df['gender'] = le.fit_transform(df['gender'])
df['profile_yn'] = le.fit_transform(df['profile_yn'])
```

2.2.5. Feature engineering:

Feature engineering involves converting raw data into features that can be used by machine learning algorithms. This process can be challenging, particularly for signals and images, as it demands a deep understanding of the data and often requires experimentation, even with automated methods. (Mathworks.com, n.d.) In this task we performed feature engineering to generate two new features which we found useful for analysis and modelling. The code for this step is as follows:

```
# Feature Engineering: Create new features
df['description_length'] = df['description'].apply(lambda x: len(str(x))) #
Length of description
df['tweet_to_retweet_ratio'] = df['tweet_count'] / (df['retweet_count'] + 1) #
Avoid division by zero
```

2.3. Exploratory Data Analysis:

In order to gain familiarity with the data & extract valuable insights prior to performing modelling, we performed exploratory data analysis after completing the preprocessing steps. We plotted several charts and plots to explore how various features of the dataset

are correlated. The code segments to plot the visuals are included in regression.py. All the plots were formed by using the Matplotlib library. (Matplotlib, n.d.)

To explore the relationships among the numerical attributes, we generated a correlation matrix (Figure 2). In a correlation matrix, a value of 1 means the attributes are highly correlated, -1 indicates a strong negative correlation, and 0 suggests no correlation. (Wagavkar 2023). For example, in Figure 2, the 'tweet_count' and 'fav_number' columns have the highest positive correlation, meaning that users who tweet more often tend to receive more favorites (likes) on their tweets. This relationship suggests that high activity on the platform is generally rewarded with more engagement.

On the other hand, the 'description_length' and 'gender' columns exhibit the strongest negative correlation. This indicates that profiles with longer descriptions are generally associated with lower confidence in gender classification, possibly because these profiles belong to brands or entities where gender identification is less straightforward.

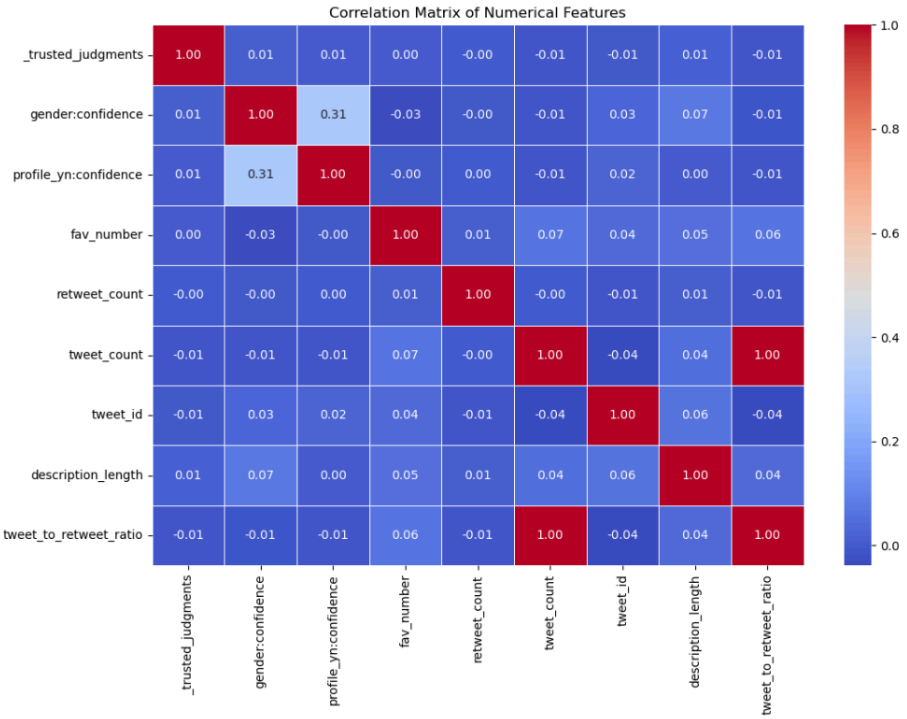

Figure 2: *Correlation Matrix*

In order to understand the distribution of values assigned in the 'gender' feature we plotted a visual. As per the data, there are 4 unique values in 'gender' feature, namely - 'male', 'female', 'brand' & 'unknown'. The distribution of these values is illustrated in Figure 3. The figure provides insight into how the dataset is divided across different gender categories. In this dataset, the 'gender' column contains four unique values: 'male', 'female', 'brand', and 'unknown':

- *Male:* Represents user profiles identified as male. This group forms a significant portion of the dataset.
- *Female:* Represents user profiles identified as female. The proportion of female profiles is roughly comparable to male profiles, providing a fairly balanced dataset for gender classification.
- *Brand:* Refers to non-human profiles, typically representing companies, organizations, or services. These profiles are classified separately as they do not correspond to individual human users.
- *Unknown:* Represents profiles where the gender could not be determined. These could be incomplete or ambiguous profiles that did not provide enough information to make a gender determination.

Insights from the Distribution:

For example, from Figure 3, we can observe that the dataset is relatively balanced between 'male' and 'female' profiles. This balance is important because it suggests that the model will have an equitable amount of data for training to predict both male and female classifications. However, the 'brand' category is considerably smaller than the human categories, meaning that there is less data available for classifying non-human profiles. This class imbalance could potentially lead to challenges in the model's ability to accurately classify brands, as it may be biased towards predicting human genders due to the larger sample sizes for 'male' and 'female'. Lastly, the 'unknown' group is small, but it represents a potential area of difficulty in the dataset. If a profile's gender is ambiguous, this could create confusion for the model when it encounters incomplete or uncertain data.

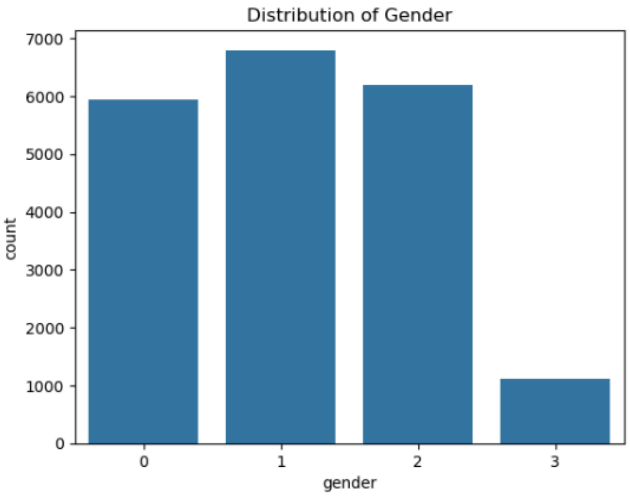

Figure 3: *Distribution of "gender"*

In order to further understand the relationships between the features of provided data we plotted scatter plots. Scatter plots are illustrated in the following figures. Figure 4 visualizes the relationship between two numerical features in the dataset: ‘tweet_count’ and ‘gender’. It helps in identifying how the number of tweets a user has posted is associated with the confidence in determining their gender.

- *‘tweet_count’* represents the total number of tweets a user has posted. It gives a sense of how active the user is on Twitter.
- *‘gender’* is a numerical value that indicates the confidence level (ranging from 0 to 1) of the algorithm used to classify the user's gender. A value closer to 1 indicates high confidence, while values closer to 0 suggest low confidence.

Insights from Figure 4:

- *Positive Trend:* The scatter plot may show a positive trend, where higher tweet counts are associated with higher confidence in gender classification. This suggests that users who tweet more frequently tend to provide more data for the algorithm, which can lead to a clearer and more confident classification of their gender.
- *Outliers:* Some points in the scatter plot may represent outliers, where users with very high tweet counts have a low ‘gender’. This could indicate that despite being highly active, certain profiles (like brands or anonymous accounts) provide information that does not contribute meaningfully to gender classification.
- *Clusters:* Depending on the data, the scatter plot might also reveal distinct clusters of users. For instance, one cluster might consist of human users with moderate to high tweet counts and high ‘gender’, while another cluster might consist of non-human users or ambiguous profiles with lower confidence values, regardless of their tweet count.

Implications for the Model:

This scatter plot highlights that ‘tweet_count’ is a valuable feature for predicting ‘gender’, but it’s not always straightforward. Outliers and clusters provide important clues about potential challenges the model may face, such as distinguishing between human and non-human profiles, or dealing with users who exhibit a high tweet count but ambiguous gender information. By analyzing the distribution of points in this scatter plot, we can

assess how well the ‘tweet_count’ feature supports the gender classification task and where additional feature engineering or model adjustments might be necessary.

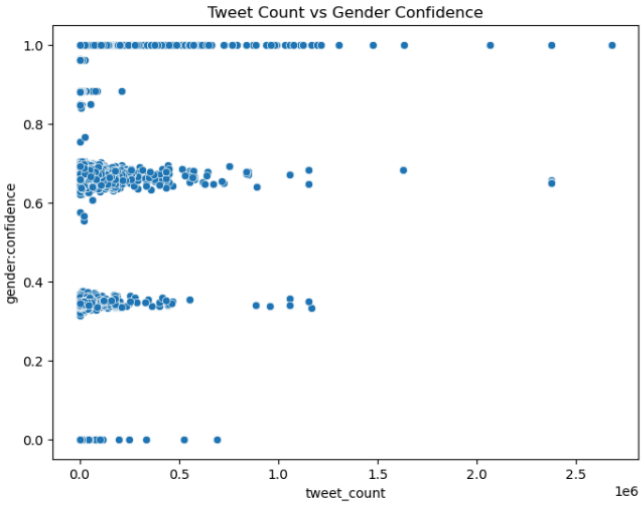

Figure 4: *Scatter Plot between ‘tweet_count’ & ‘gender:confidence’ Features*

Figure 5 illustrates the relationship between the ‘tweet_to_retweet_ratio’ and ‘gender’. It provides insight into how the ratio of original tweets to retweets correlates with the confidence level in determining a user’s gender.

- *‘tweet_to_retweet_ratio’* is a feature that represents the proportion of original tweets to retweets made by a user. A higher ratio indicates that the user is posting more original content rather than retweeting others, while a lower ratio suggests that the user is primarily retweeting others' posts.

- *‘gender’* is a numerical value (from 0 to 1) that indicates how confident the model is in classifying the user’s gender. A value close to 1 means high confidence, and a value closer to 0 means low confidence.

Insights from Figure 5:

- *Positive Trend:* A positive trend may emerge in the scatter plot, indicating that users with a higher ‘tweet_to_retweet_ratio’ tend to have higher ‘gender’. This suggests that users who post more original content provide clearer cues (e.g., language patterns, topics) that the model can use to determine gender with greater confidence.

- *Negative or Flat Trend:* In some cases, the scatter plot might show a lack of strong correlation or even a negative trend. For example, users who frequently retweet (low ‘tweet_to_retweet_ratio’) might not provide enough original content for the model to confidently classify their gender, leading to lower confidence scores.

- *Outliers:* The plot might reveal outliers where users have a high 'tweet_to_retweet_ratio' but low 'gender'. These users might be brands or automated accounts that frequently post original content, but their tweets may lack the personal characteristics (e.g., gendered language) that would help the model identify their gender.

Implications for the Model:

The 'tweet_to_retweet_ratio' feature provides insight into user behavior, which can influence how confidently the model predicts gender. Users who post more original content give the model more information to work with, potentially leading to higher 'gender'. In contrast, users who primarily retweet might provide less personal information, making it difficult for the model to classify their gender accurately.

This scatter plot helps identify which types of users the model struggles with, such as those with a low 'tweet_to_retweet_ratio'. It also highlights the importance of content creation behavior in improving gender classification confidence. If we notice outliers or clusters of users with low 'gender', additional features (such as profile descriptions or activity patterns) may need to be incorporated into the model to improve accuracy.

Example Conclusion:

For instance, if a cluster of users with a high 'tweet_to_retweet_ratio' but low 'gender' appears, we might infer that these users, despite posting frequently, belong to categories like brands or bots, where the content is less gender-specific. This observation could lead to additional feature engineering to further distinguish between human and non-human profiles, improving the model's overall performance.

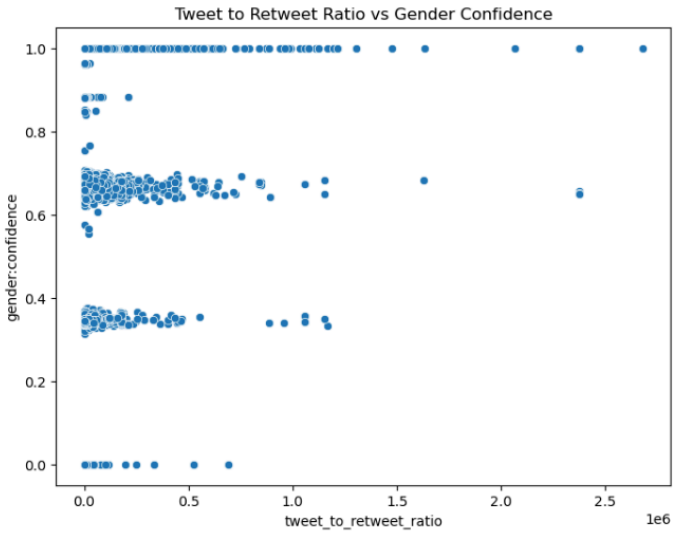

Figure 5: *Scatter Plot between 'tweet_to_retweet_ratio' & 'gender:confidence'*

2.4. Modelling:

2.4.1. Necessary Steps:

Finally, having performed preprocessing as well as EDA, we can dive into synthesising machine learning models and assess their performances. For this task we developed regression models using three techniques, namely linear regression, random forest regression & gradient boosting. All the models were imported, trained & fitted on the data using the modules which are a part of Sci-kit learn library. (Sci-kit learn, n.d.)

A crucial step before performing modelling is to perform a train & test split. In this step the target feature (denoted by 'y') & data features (denoted by 'X') are divided into subsets denoted by 'X_train', 'X_test, 'y_train', 'y_test'. This is done through the sci-kit learn train_test_split module ((Sci-kit learn, n.d.) The models are trained on training data subsets and tested on test data subsets.

2.4.2. Choice of Models & Justification:

Linear regression is a valuable method for predicting numerical outcomes (James, 2023). It has a long history and is well-documented in many textbooks (James, 2023). While it may seem less exciting than newer statistical techniques, linear regression remains a useful and widely used tool (James, 2023). It also serves as a foundation for understanding more advanced methods (James, 2023). Many modern statistical learning approaches can be considered as extensions or generalisations of linear regression (James, 2023). Therefore, a solid grasp of linear regression is crucial before delving into more complex learning methods (James, 2023).

Random forest is a well-known machine learning technique that combines the results of several decision trees (IBM, n.d.). Its ease of use and flexibility have contributed to its widespread adoption for both classification and regression tasks (IBM, n.d.).

Gradient boosting is a machine learning technique that combines several weak models, typically decision trees, into a single, more accurate predictive model (Snowflake.com, n.d.). It works by sequentially adding new models that focus on correcting the errors of the previous ones (Snowflake.com, n.d.). The final prediction is the sum of the predictions from all the models (Snowflake.com, n.d.). Gradient boosting is a combination of gradient descent and boosting techniques (Snowflake.com, n.d.).

2.4.2.1. Model 1 - Linear Regression:

We developed this model to perform basic analysis on the data. The metrics used to assess the model's performance were r2_score & mean_squared_error (sci-kit learn, n.d.).

2.4.2.2. Model 2 - Random Forest Regression:

Having developed an understanding of how regression modelling can be performed on this dataset through the means of Linear regression, we decided to use a technique which is expected to perform better and make more accurate predictions. Metrics used to assess the model's performance were r2_score & mean_squared_error.

2.4.2.3. Model 3 - Gradient boosting:

Finally, we decided to incorporate gradient boosting techniques to further improve the performance of our models. Metrics used to assess the model's performance were r2_score & mean_squared_error.

The code to implement all of the above techniques is available in regression.py

2.4.3. Model Scores:

Metric	Model	Score
Mean squared error	Linear Regression	0.03
R squared	Linear Regression	0.05
Mean squared error	Random Forest	0.03
R squared	Random Forest	0.09
Mean squared error	Gradient Boosting	0.19
R squared	Gradient Boosting	0.13

Explanation of Results:

- *Mean Squared Error (MSE):* This metric measures the average squared difference between the actual and predicted values. A lower MSE indicates better model performance. For instance, an MSE of 0.03 means that, on average, the squared differences between the predicted values and the actual values are 0.03, which suggests a small error. The Random Forest and Linear Regression models have

the lowest MSE (0.03), meaning their predictions are more accurate compared to the Gradient Boosting model, which has a higher MSE (0.19), indicating more error.

- *R Squared (R^2):* R^2 measures how well the model explains the variance in the target variable. An R^2 of 1 means the model perfectly predicts the target variable, while an R^2 of 0 means the model does not explain the variability at all. For example, the R^2 score of 0.05 for Linear Regression means that only 5% of the variance in the target variable is explained by the model, which is relatively low. Random Forest and Gradient Boosting perform slightly better, with R^2 scores of 0.09 and 0.13, respectively. However, these scores indicate that the models are still not explaining much of the variance in the dataset, suggesting that regression models may not be the best fit for this data.

In summary, the low MSE values across all models suggest that the predictions are reasonably accurate, but the low R^2 scores indicate that the models are not capturing much of the variability in the target variable. This could imply that other machine learning techniques (such as classification or clustering) might be more appropriate for this dataset, or that additional feature engineering could help improve the models' performance.

2.4.4. Plotting Modelling Results:

Having completed the modelling steps and assessing the performance of each modelling technique, we plotted some visuals (refer to Figure 6) in order to better understand how the model performed, the visuals were plotted using Matplotlib (Matplotlib, n.d.) and demonstrate a relation between the actual target feature values and the target feature values predicted by the models.

These plots visually compare the actual values of the target feature (in this case, gender confidence) with the values predicted by two different regression models: Linear Regression and Random Forest Regression. The goal of these plots is to show how well the models are able to predict the actual values.

- *Actual Values:* These are the true values of the target feature from the dataset.
- *Predicted Values:* These are the values predicted by the model based on the features used in training.

The plots typically show the actual values on one axis (usually the x-axis) and the predicted values on the other (usually the y-axis). Ideally, if the model performs perfectly,

all the points should lie along a diagonal line where the actual values match the predicted values exactly.

Insights from the Plots:

1. *Linear Regression Plot:*
    - In the case of Linear Regression, the points may not align well with the diagonal line, especially if the model is not capturing the complexity of the data. For instance, you might notice that the predicted values deviate significantly from the actual values for some instances.
    - This indicates that the Linear Regression model struggles to accurately predict the target variable, potentially because of the linear nature of the model, which might not be suitable for complex relationships in the data.
    - **Example**: For a given user, if the actual ‘gender’ is 0.8, the model might predict a value closer to 0.5, highlighting a significant error. This suggests that the model is unable to fully capture the underlying patterns in the data that contribute to gender confidence.
2. *Random Forest Regression Plot:*
    - The Random Forest Regression plot is expected to show better alignment between the actual and predicted values compared to Linear Regression. Since Random Forest is an ensemble method that combines multiple decision trees, it can handle more complex data relationships and provide more accurate predictions.
    - While not all points may lie perfectly on the diagonal, you will likely see that more points are closer to the ideal line compared to the Linear Regression plot, indicating better predictive performance.
    - **Example**: For the same user with an actual ‘gender’ of 0.8, the Random Forest model might predict 0.75, a much closer approximation than the Linear Regression model. This indicates that Random Forest is better at capturing the non-linear relationships in the data, leading to more accurate predictions.

Comparison Between the Models:

- *Linear Regression:* The plot for Linear Regression will likely show more scatter, with many points deviating from the diagonal, indicating that the model struggles with certain patterns in the data. This could be due to the fact that Linear Regression assumes a linear relationship between the features and the target variable, which may not hold true in this case.
- *Random Forest Regression:* The plot for Random Forest is expected to show a closer fit to the diagonal, indicating that this model is better at predicting the target values. Random Forest can model more complex relationships and is less prone to errors caused by outliers or non-linearity in the data.

Implications for Model Performance:

The plots clearly demonstrate that Random Forest Regression performs better than Linear Regression in this case. The points are closer to the ideal diagonal in the Random Forest plot, meaning that the predictions are more accurate. This suggests that Random Forest, which is capable of handling more complex relationships, is a better choice for this dataset than Linear Regression, which might oversimplify the relationships between the features and the target variable.

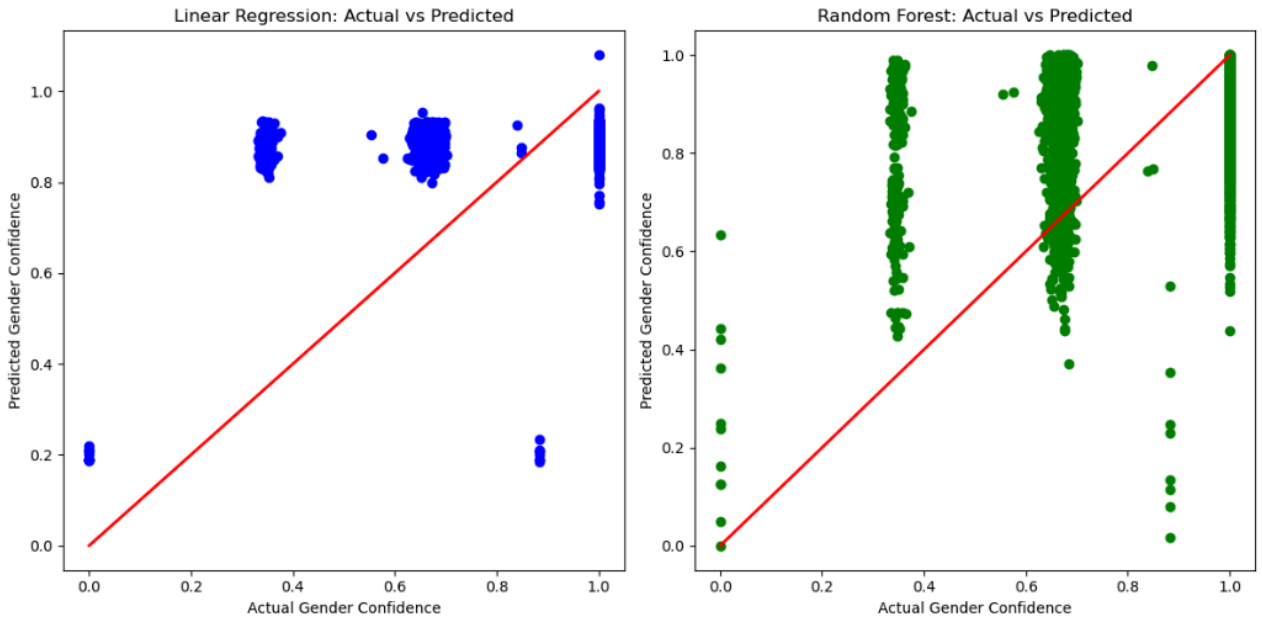

Figure 6: *Plots between Actual & Predicted Values of Target Feature, for Linear Regression & Random Forest Regression*

Additionally, we calculated a statistical measure 'residuals' which is the difference between target feature values & predicted target feature values. We plotted the residuals calculated

from both the random forest regression model & gradient boosting model. The plots are illustrated in Figures 7 and 8 respectively.

Figure 7 illustrates the distribution of the residuals from the Random Forest Regression model. Residuals represent the difference between the actual values and the predicted values made by the model. The residual for a given data point is calculated as:

Residual=Actual Value−Predicted Value\text{Residual} = \text{Actual Value} - \text{Predicted Value}Residual=Actual Value−Predicted Value

Residuals give insight into how well the model is performing. Ideally, the residuals should be centered around zero and should not show any specific pattern. The distribution of residuals helps evaluate if the model is making consistent errors or if there is any systematic bias.

Understanding the Residual Plot:

1. *Residuals Centered Around Zero:*
    - A well-performing model will have residuals that are approximately centered around zero. This means that the model's predictions are neither consistently too high nor too low, and the errors are balanced.
    - In this plot, you should expect most of the residuals to be close to zero, which indicates that for most of the data points, the model's predictions are fairly accurate.
2. *Spread of Residuals:*
    - The spread of the residuals gives an indication of the variability in the model's errors. A narrow distribution means that the errors are small and consistent, while a wider distribution suggests that the model is making larger errors for some predictions.
    - For the Random Forest Regression, the residuals should be relatively well-distributed, but you might still see some spread, indicating that while the model is good, it's not perfect.

3. *Outliers and Skewness:*

    - Outliers in the residuals (very high or very low values) suggest that the model is making significant errors for some data points. If the plot shows a few residuals that are far from zero, it indicates that the model struggles with certain types of data.

    - Skewness in the residuals (where most residuals are either above or below zero) would suggest that the model has a bias. For example, if most residuals are positive, it means that the model is consistently underpredicting the actual values, and if they are mostly negative, the model is overpredicting.

What the Residual Distribution Shows:

- *Accuracy of Predictions:* If most residuals are clustered around zero, this indicates that the Random Forest model is making accurate predictions for most data points. The smaller the residuals, the closer the predictions are to the actual values.

- *Model's Consistency:* A well-distributed, symmetric residual plot suggests that the Random Forest model is consistent across the dataset and that it isn't biased in favor of overpredicting or underpredicting the target feature.

- *Outliers and Challenges:* If there are a few large residuals (far from zero), it suggests that the model struggles with certain data points or specific patterns in the data. These could be users whose activity or profile characteristics don't conform to typical patterns, making it harder for the model to predict their gender accurately.

Implications for Model Improvement:

If the residuals show significant spread or outliers, it may indicate that certain patterns in the data are not being captured by the Random Forest model. In such cases, additional feature engineering or even a more complex model, such as gradient boosting, could be considered to improve accuracy.

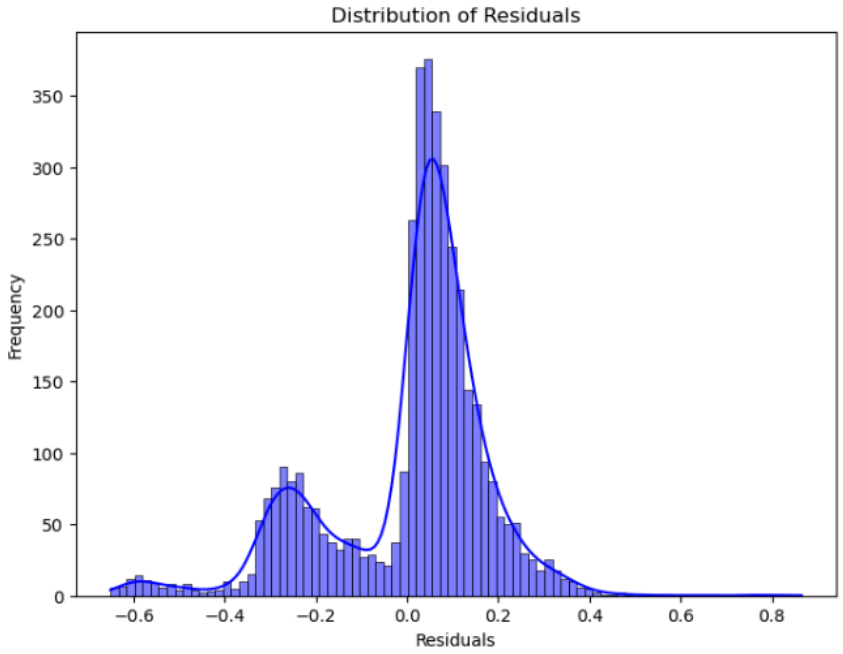

Figure 7: *Distribution of Residuals from Random Forest Regression*

From the Figure 8 we can understand that the residuals distribution of the gradient boosting model is much more dense and centred towards 0.0-0.2, thus indicating that after applying gradient boosting technique, the model produced more accurate results.

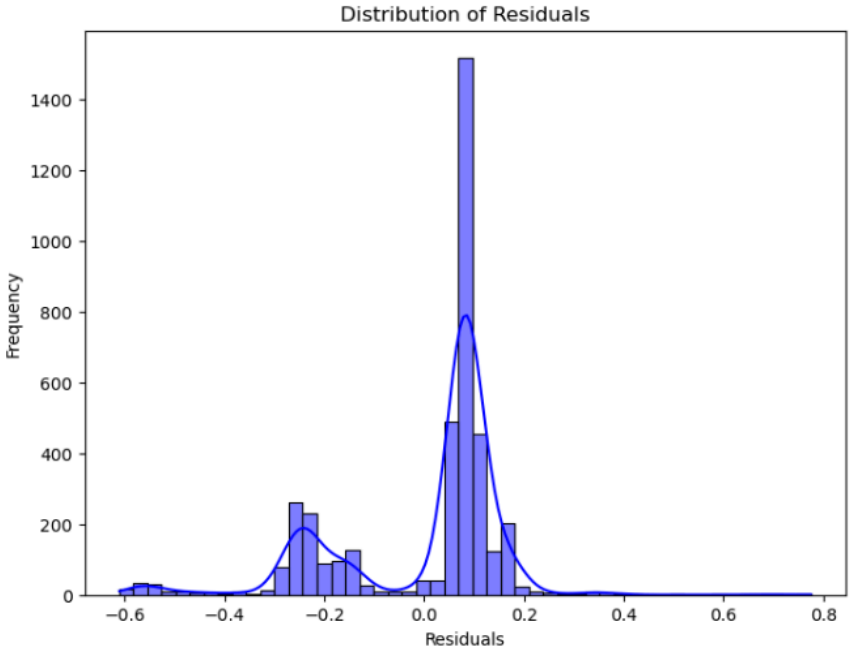

Figure 8: *Distribution of Residuals from Gradient Boosting Model*

2.4.5. Feature Importance:

Feature importance quantifies the impact of each input variable on a model's predictions. Understanding feature importance is crucial for interpreting and trusting machine learning results (Raskoshinskii, 2024). It provides insights into which features are most important in

driving the model's predictions, helping users focus on relevant aspects of the data (Raskoshinskii, 2024).

Feature importance was performed after implementing Random forest regression & Gradient Boosting. The results are illustrated in the following figures:

Figure 9 shows the feature importance rankings for the Random Forest Regression model. Feature importance quantifies the contribution of each feature (or independent variable) to the predictions made by the model. In the case of Random Forest, the importance of a feature is determined by how much it contributes to reducing prediction error (measured by metrics like Gini impurity or Mean Squared Error) across all the decision trees in the forest.

Understanding Feature Importance:

- *Higher Feature Importance:* Features that have higher importance scores play a more significant role in predicting the target variable. In this case, they are more influential in determining gender, meaning the model relies on them more heavily to make accurate predictions.
- *Lower Feature Importance:* Features with lower importance scores contribute less to the model's predictions. These features may have little to no impact on the model's performance or may be redundant when combined with other features.

Insights from the Feature Importance Distribution:

1. *Top Features:*
    - The features with the highest importance are likely to be 'tweet_count', 'fav_number', and 'description_length'. These features are expected to provide the most significant insight into gender confidence.
    - **'tweet_count'**: A higher number of tweets could indicate that the user is more active on the platform, providing more data for the model to classify the user's gender confidently.
    - **'fav_number'** (number of favorites/likes): Users with a high number of favorites might be more engaged or visible on the platform, providing indirect cues about their identity that the model can leverage for better predictions.

- **‘description_length’**: A longer profile description might contain more information about the user, such as personal details, profession, or interests, which can help the model make a more confident gender classification.

2. *Moderately Important Features:*
    - Features such as ‘tweet_to_retweet_ratio’ and ‘profile_yn’ may have moderate importance. These features provide additional context about user behavior but might not be as directly correlated with gender confidence as the top features.
    - **‘tweet_to_retweet_ratio’**: A user who posts more original content (i.e., higher ratio) might give the model more text data to analyze for gender identification, while those who primarily retweet (lower ratio) may provide less gender-specific content.
    - **‘profile_yn’**: Whether the profile exists or is fully visible can also affect how confidently the model can classify the user’s gender.
3. *Least Important Features:*
    - Features with low importance, such as ‘sidebar_color’ or ‘link_color’, might have little influence on the model’s predictions. These visual design elements of the profile are less likely to provide direct insights into gender confidence, especially compared to behavioral and content-related features like tweet count or description length.

Implications for Model Performance:

- *Top Features Drive Performance:* The Random Forest model relies most heavily on features like tweet count and fav_number because they provide the most valuable information for predicting gender confidence. These features likely capture behavioral patterns that correlate strongly with gender classification.
- *Low-Importance Features Can Be Pruned:* Features with very low importance scores (e.g., ‘sidebar_color’) contribute little to the model’s predictions and might be considered for removal in future iterations. Removing these features can simplify the model without sacrificing performance, potentially reducing overfitting.

Why Feature Importance Matters:

Understanding which features are most important helps to:

- *Improve Model Interpretability:* By identifying which features the model relies on, we can better understand the factors that contribute to gender confidence. For example, tweet count and description length are clear indicators of user activity and personal information, which are crucial for classification.
- *Guide Feature Engineering:* If certain features, such as tweet count and fav_number, are highly predictive, we might focus more on creating new features related to user activity or engagement to further improve the model's accuracy.

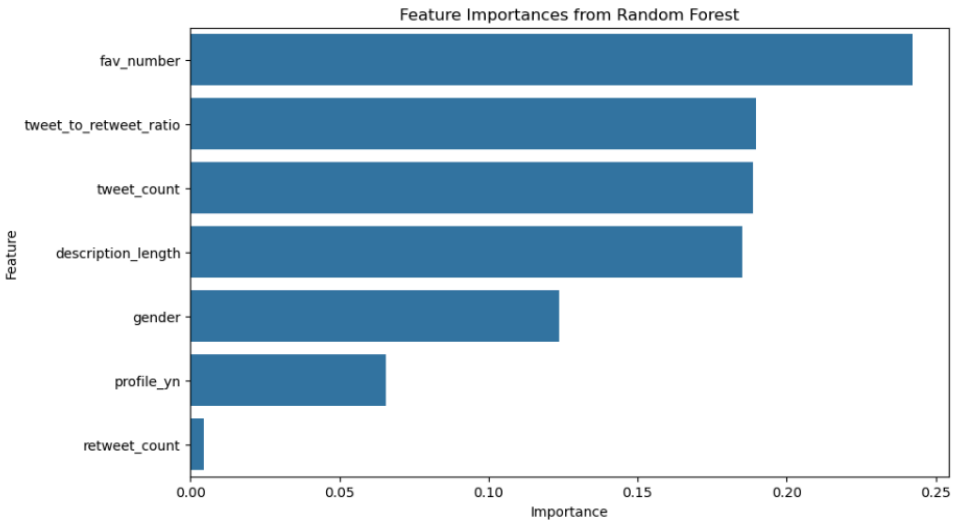

Figure 9: *Feature Importance Distribution for Random Forest*

Figure 10 shows the feature importance rankings for the Gradient Boosting Regression model. Like Random Forest, Gradient Boosting identifies which features contribute most to predicting the target variable (gender) by determining how much each feature reduces the prediction error over multiple decision trees in the boosting process.

In Gradient Boosting, weak learners (individual decision trees) are built sequentially, with each tree correcting the errors of the previous ones. The feature importance values reflect how each feature helps the model reduce prediction errors across this sequence of trees.

Insights from the Feature Importance Distribution:

1. *Top Features:*
    - Similar to the Random Forest model, the most important features in Gradient Boosting are likely to be 'tweet_count', 'fav_number', and

‘description_length’. These features have the highest influence on the model’s predictions because they provide the most valuable information about user behavior and content, which are directly linked to the confidence in gender classification.

- **‘tweet_count’:** The more tweets a user posts, the more information the model has to make a confident prediction about the user’s gender. The frequency of tweets provides a behavioral pattern that helps distinguish between different types of users (e.g., human vs. brand).
- **‘fav_number’:** Users who receive more favorites (likes) may engage in activities that are more distinguishable by gender, providing the model with strong signals to make confident gender predictions.
- **‘description_length’:** Longer profile descriptions tend to offer more personal information, such as interests or professions, which can help the model increase its confidence in gender classification.

2. *Moderately Important Features:*
   - Features like ‘tweet_to_retweet_ratio’ and ‘profile_yn’ might have moderate importance in Gradient Boosting, similar to their role in Random Forest. While these features contribute to the model’s predictions, they are not as influential as tweet count or fav_number.
   - ‘**tweet_to_retweet_ratio’:** This feature measures the ratio of original content to retweeted content. Users who post more original tweets (high ratio) provide more unique text data, which can help the model predict gender with higher confidence.
   - **‘profile_yn’:** Whether or not a profile is fully visible or active can affect how much information the model has to work with, influencing its ability to predict gender confidence.
3. *Least Important Features:*
   - Visual elements like ‘sidebar_color’ and ‘link_color’ likely have the lowest importance. These aesthetic features contribute little to the model’s ability to predict gender confidence because they are unrelated to user behavior or content generation, which are more directly tied to gender-related patterns.

Why Gradient Boosting Gives Importance to These Features:

- *Sequential Learning:* Gradient Boosting builds trees sequentially, each focusing on correcting the errors of the previous trees. This allows the model to learn the most from features like tweet count and fav_number, which provide clear signals for gender confidence, while also gradually improving the prediction accuracy for users with less clear patterns (such as users with fewer tweets or likes).

- *Non-linear Relationships:* Gradient Boosting is particularly effective in capturing non-linear relationships between features and the target variable. Features like tweet_to_retweet_ratio and description_length might have complex interactions that are better captured by Gradient Boosting than by simpler models like Linear Regression.

Comparison with Random Forest:

- *Similarity:* The most important features in both Random Forest and Gradient Boosting are likely similar because both models rely on tree-based structures to assess the contribution of each feature. Tweet count, fav_number, and description_length are consistently ranked as the top features due to their strong influence on gender confidence.

- *Difference:* However, the way these models treat feature importance differs slightly. Random Forest uses a more straightforward averaging of feature importance across multiple trees, while Gradient Boosting focuses on the features that help correct errors from previous iterations. This makes Gradient Boosting potentially more sensitive to the contributions of individual features in specific parts of the data, particularly for users with fewer patterns.

Implications for Model Performance:

- *Key Features Drive Predictions:* Just like with Random Forest, tweet count, fav_number, and description_length are the primary drivers of the Gradient Boosting model's performance. These features provide the most valuable information for predicting gender confidence, allowing the model to correct errors iteratively and improve predictions.

- *Less Important Features Can Be Pruned:* Features with very low importance scores (like sidebar_color) contribute little to the model's predictions and could be

considered for removal in future iterations. Pruning these features might streamline the model without significantly affecting its accuracy.

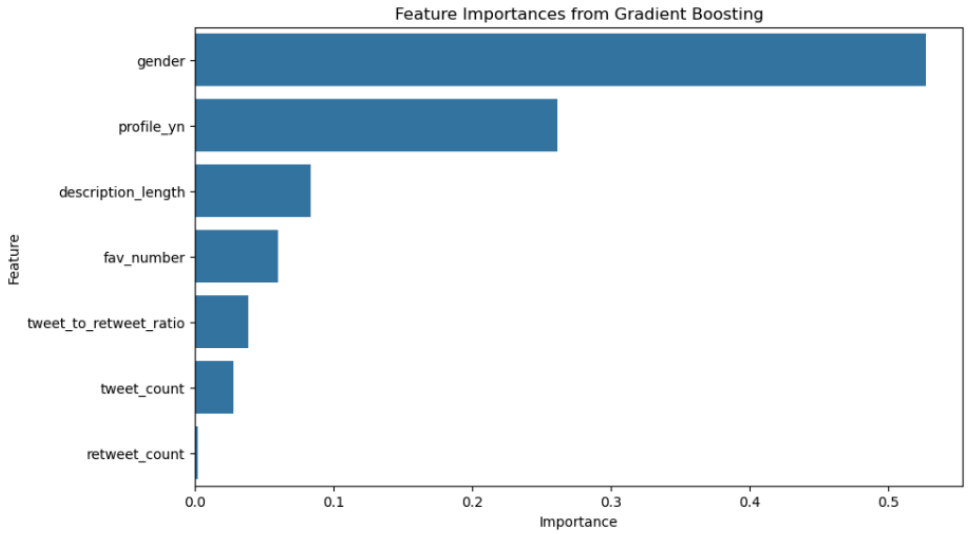

Figure 10: *Feature Importance Distribution for Gradient Boosting*

2.5. Results:

In this segment of the assignment associated with regression, we gained numerous valuable insights about the data as well as the relationships between the features through the process of exploratory data analysis. Furthermore, we tested our knowledge of regression techniques and principles on the provided dataset. However, regression models did not perform to a good standard in terms of accurately predicting the values of the target feature. This implies that other modelling approaches such as clustering and classification might be more suitable for this particular dataset, which are covered in the following segments of the assignment and are mentioned in detail in the report.

3. Text Processing 1:

3.1. Algorithm Consideration:

Text processing is the automated examination and modification of digital text, allowing machine learning models to retrieve structured data for text analysis, transformation, and new text generation (DataRobot, 2020). It plays a critical role in many machine learning applications, such as language translation, sentiment analysis, spam filtering, and more, making it one of the most widely used techniques in natural language processing (DataRobot, 2020).

In terms of the context of this project, text processing can be a highly effective method for classifying humans and brands based on the content they generate or share, especially on platforms like social media. This will, therefore, help in identifying profiles that are mistakenly labelled as humans or brands.

Several text processing methods will help in this process; some of which have been used and discussed in detail in the sections below. The goal of employing these text processing techniques is to study the linguistic behaviour of human and non-human profiles.

3.2. Choice of Methods & Justification:

In this section, three text processing methods have been applied:

1. **Term Frequency-Inverse Document Frequency (TF-IDF):** It is an essential technique for quantifying the importance of words in textual data relative to a broader dataset, or what's known as a corpus (GeeksforGeeks, 2023). It helps to highlight the terms that are more relevant to individual profiles but less common across the entire corpus (GeeksforGeeks, 2023). The more times a word appears in the text, the more significant it becomes (GeeksforGeeks, 2023). However, if a word appears frequently across the collection, its significance decreases (GeeksforGeeks, 2023). In tasks like detecting misclassification between human and non-human profiles, TF-IDF enables the identification of unique language patterns. For example, human profiles might use more varied conversational words, while brand profiles might consistently mention promotional terms or product-related phrases. By using TF-IDF, meaningful features that differentiate human language from brand-related content can be extracted, improving the accuracy of classification and helping flag potentially misclassified profiles.

2. **Sentiment Analysis:** It is a useful tool for understanding the emotional tone and intent behind user-generated content such as tweets and profile descriptions (Shah, 2020). In the context of detecting misinformation or distinguishing between human and non-human profiles, sentiment analysis can reveal important behavioural differences. Human profiles often exhibit a wide range of sentiments, positive, negative, or neutral, depending on personal opinions, emotions, or experiences. In contrast, brand profiles tend to maintain a neutral, promotional, or consistent tone. Analyzing sentiment can help detect misclassified accounts by identifying unexpected emotional patterns, such as a human profile displaying overly neutral

language or a non-human profile showing unusually emotional content. This makes sentiment analysis a powerful tool for spotting abnormalities and potential disinformation in social networks.

3.3. Data Preprocessing & Visualization:

Before employing TF-IDF or Sentiment Analysis on the “twitter_user_data” dataset, data preprocessing needs to be conducted. The first step after importing the dataset was to filter it out and create a dataframe with the required columns. In this case, the columns “text” and “description” were chosen to conduct TF-IDF and Sentiment Analysis on as they contain the text that will provide a more comprehensive understanding of the user's content and communication style across different aspects of their profile:

- *The "text" column:* contains actual tweets, which directly represent a user's communication style, interests, and sentiment. These tweets will be rich in real-time interactions, opinions, and potentially personal or brand-relevant information.
- *The "description" column:* is a concise summary of how users present themselves, their interests, or their businesses. It’s often where brands provide promotional content, while individuals may summarize their personal identities or beliefs.

The column “gender” was also chosen as it will further help in this comprehension by allowing for grouping of the results by gender. This will help in understanding the different languages and styles of the “male”, “female”, and “brand” profiles. The rest of the 23 columns, such as ‘unit_id’, ‘golden’, ‘trusted_judgments’, etc, were not rich in textual content that would help in extracting more meaningful data that will allow us to study the linguistic behaviour of humans and non-humans as mentioned above. Hence, they were all dropped. This project is also about classifying human and non-human profiles according to categories stated in the ‘gender’ field. Hence, fields like ‘unit_id’ were also considered not useful for this process as we need to group the results by ‘gender’ in order to achieve the required project goals.

The code for this part is shown below:

```
#load the dataset
data = pd.read_csv("twitter_user_data.csv", encoding="ISO-8859-1")

#select the "text", "description", and "gender" columns
```

```
selected_columns = data.filter(items=['text', 'description', 'gender'])
```

The next step was to find whether there are duplicate values in the “text” and “description” columns or not as these duplicates can act as noise and hence removing them ensures that repeated or redundant content, that may just be repeated because of its popularity, does not skew the results (Imarticus, 2023). This will help in providing more accurate and meaningful insights into unique user behaviour and sentiment (Imarticus, 2023). Figure 11 below shows the number of unique values in the three chosen columns. The "gender" column, which classifies profiles into categories (e.g., male, female, brand), has no unique values because it contains only a small set of possible categories (refer to Figure 11). On the other hand, the "text" and “description” columns have a much higher number of unique values, with over 17,500 and around 15,000 unique entries respectively (refer to Figure 11). This indicates that there is a significant number of duplicates in these two columns that need to be removed. Hence, the “pandas” library’s “drop_duplicates()” function (NumFOCUS, 2024a) was used to drop those duplicate values. The code for this part is shown below:

```
#check the number of unique values in each column
unique_values = selected_columns.nunique()

#remove duplicate rows in "description" and "text"
data_cleaned = selected_columns.drop_duplicates(subset=['text'])
data_cleaned = selected_columns.drop_duplicates(subset=['description'])
```

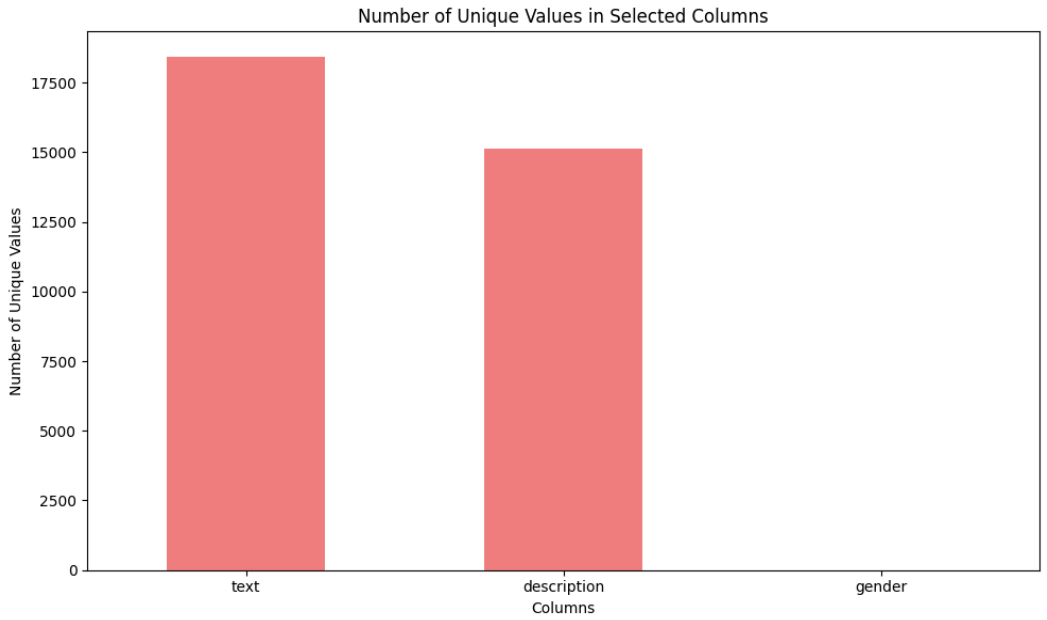

Figure 11: *Number of Unique Values in the Selected Columns*

The columns were also checked for missing values and the result is illustrated in Figure 12 with around 1 and 70 missing values in the "description" and "gender" columns respectively. It is important to remove those missing values from the dataset to ensure data integrity, prevent biases, and improve the accuracy of analysis or machine learning models by providing a complete and consistent dataset (R, 2024). To remove these values, the "pandas" library's "dropna()" function was used (NumFOCUS, 2024b). The code for this part is shown below:

```
#check for missing values
missing_values = data_cleaned.isnull().sum()

#remove missing data
data_cleaned = data_cleaned.dropna(subset=['text', 'description', 'gender'])
print(data_cleaned)
```

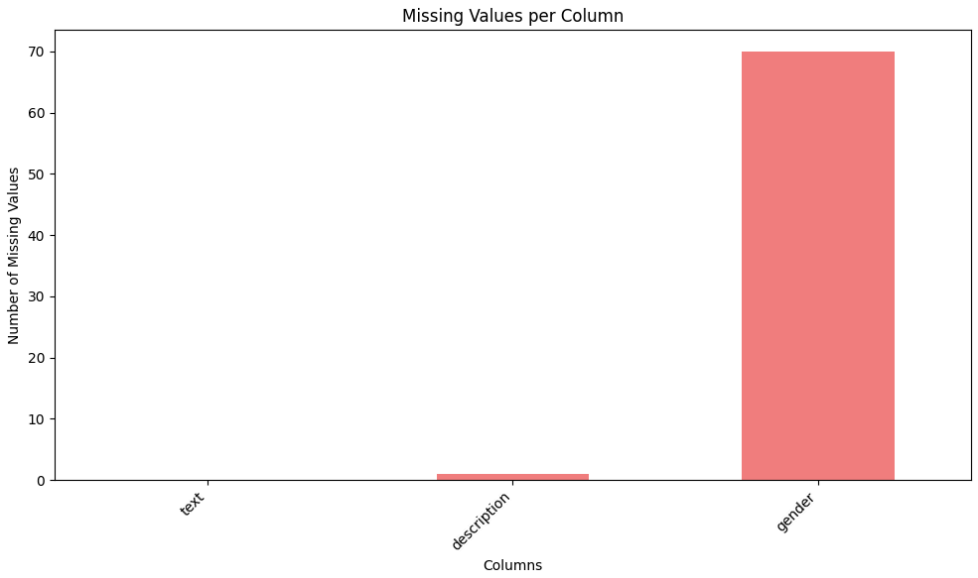

Figure 12: *Missing Values per Column*

Another observation when analysing the dataset is that there are not three "gender" categories, but four: "male, "female", "brand", and "unknown" as shown in Figure 13. Rows with "unknown" values are still considered missing values. They will not be beneficial in the analysis and understanding of the three "gender" categories. Hence, they were removed as well and after cleaning the chosen columns, there were 14420 entries in the dataset. The code for this part is shown below:

```
#check different gender values
data_cleaned['gender'].unique()
```

```
#remove rows where "gender" is "unknown"
data_cleaned = data_cleaned[data_cleaned['gender'] != 'unknown']

#print number of rows after cleaning data
print(data_cleaned.shape[0])
```

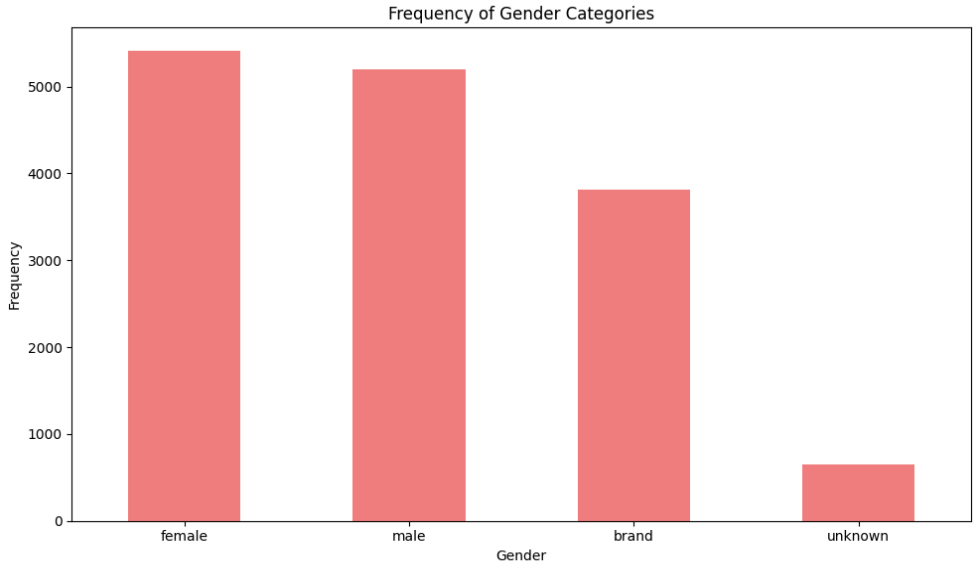

Figure 13: *Frequency of "gender" Categories*

To further study the selected items in the dataset, the mean description and text length were calculated for each of the three gender categories and plotted as shown in Figure 14. The code for this is shown below:

```
#adding columns for description length and text length
data_cleaned['description_length'] = data_cleaned['description'].apply(len)
data_cleaned['text_length'] = data_cleaned['text'].apply(len)

#calculating the mean description length and text length by gender
avg_lengths = data_cleaned.groupby('gender')[['description_length',
'text_length']].mean()

#plotting the average description length and text length by gender
fig, ax = plt.subplots(1, 2, figsize=(12, 6))

#plot for description length
avg_lengths['description_length'].plot(kind='bar', ax=ax[0], color='skyblue')
ax[0].set_title('Average Description Length by Gender')
```

```
ax[0].set_ylabel('Average Description Length')
ax[0].set_xlabel('Gender')

#plot for text length
avg_lengths['text_length'].plot(kind='bar', ax=ax[1], color='lightgreen')
ax[1].set_title('Average Text Length by Gender')
ax[1].set_ylabel('Average Text Length')
ax[1].set_xlabel('Gender')

#show plots
plt.tight_layout()
plt.show()
```

Both graphs in Figure 14 showed a similar trend with brands having the highest mean description and text length followed by the males and then the females. This, hence, proves that our assumption in Section 1 regarding the description length of brands being longer than humans is true. It also suggests that brands might be using tweets for more detailed interactions or promotions, while their descriptions are more compact and slogan-like to effectively communicate their brand message.

An interesting observation in Figure 14 is that generally, all three gender categories have higher mean text length than description length with the difference between female description and text lengths being the most significant. This suggests that users, regardless of their gender category, tend to produce more text in their tweets than in their profile descriptions.

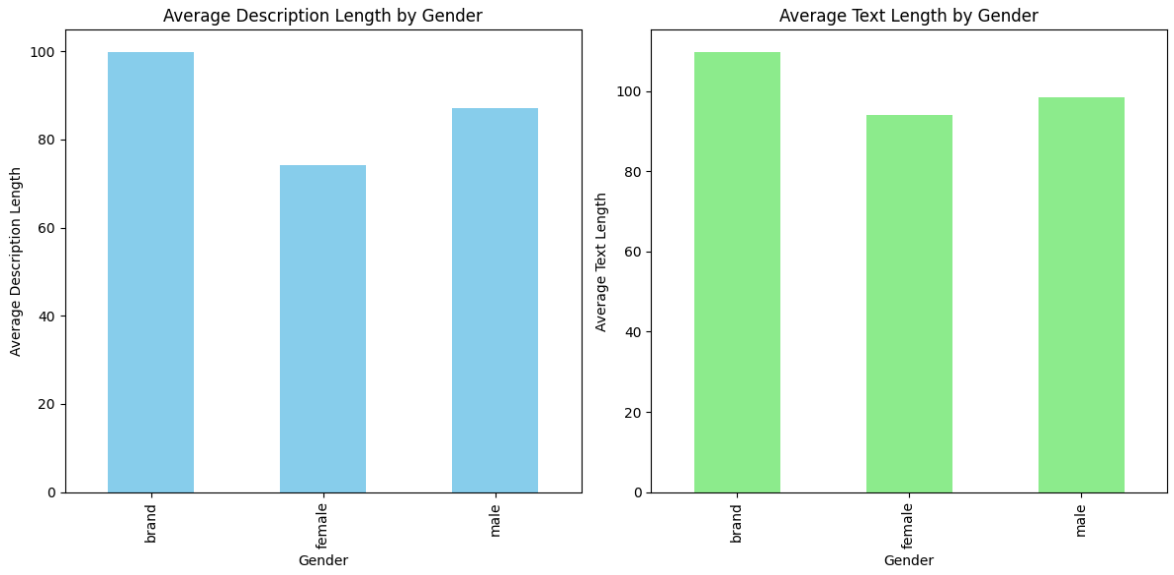

Figure 14: *Average Description / Text Length by Gender*

Text processing involves a number of extra data preprocessing steps; one of which is lowercasing the letters, tokenizing the words, removing stop words, removing numbers, removing punctuation, and removing special characters (Deepanshi, 2024). This reduces noisy data and separates the data into useful chunks for further analysis (Deepanshi, 2024). A function was created to do all of these steps and was then applied to the “text” and “description” columns. The two columns, “documents”, were then combined into a single column, “corpus”, for further analysis. The code for this part is shown below:

```
#load stopwords
stop_words = set(stopwords.words('english'))

#function to clean text
def clean_text(text):
  #lowercase and tokenize
  tokens = word_tokenize(text.lower())
  #remove numbers/punctuation
  cleaned_text = [word for word in tokens if word.isalpha()]
  #remove stopwords
  filtered_text = [word for word in cleaned_text if word not in stop_words]
  #join the cleaned tokens back into a single string
  return ' '.join(filtered_text)

#apply the function to 'text' and 'description'
data_cleaned['cleaned_text'] = data_cleaned['text'].apply(lambda x:
clean_text(str(x)))
data_cleaned['cleaned_description'] = data_cleaned['description'].apply(lambda x:
clean_text(str(x)))

#concatenate text and description into a single column
data_cleaned['combined_text'] = data_cleaned['cleaned_text'] + ' ' +
data_cleaned['cleaned_description']
```

3.4. Methods and Results:

3.4.1. Method 1 – TF-IDF:

As mentioned before, TF-IDF assigns a weight to each word that reflects its importance within a particular profile type by considering both the frequency of the word within the

profile and how rare the word is across the entire corpus (GeeksforGeeks, 2023). The method was applied on the corpus created in the last step of the data preprocessing and then the results were grouped by the different “gender” categories and the mean TF-IDF score for each word per gender was calculated. Visualizations for the top 20 words by TF-IDF score were then generated (refer to Figures 15, 16, and 17). They reveal both common and unique language patterns. Words like "love" and "https" are prevalent across all profiles, reflecting general sentiments and frequent links to external content (refer to Figures 15, 16, and 17). However, “https” ranks the highest for all profiles, whereas “love” ranks second in the “male” and “female” profiles, and 19th in the “brand” profile (refer to Figures 15, 16, and 17). This suggests that all profiles add links to their texts and descriptions. It also shows the difference in the usage of the languages between the human profiles and the non-human profiles.

Additionally, each profile type exhibits a distinctive vocabulary. Male profiles emphasize words like "music", "fan", and "man", while female profiles show words like "girl", "best", and "im". This suggests that their contents are more about their interests (refer to Figures 15 and 16). Brand profiles stand out with words such as "weather", "updates", "news", and "channel", reflecting their focus on factual non-emotional information and updates (refer to Figure 17). Additionally, the word “us” is one of the top 10 frequent words for brands which indicates a focus on collective identity rather than an individual identity (refer to Figure 17). These findings highlight the different ways in which people and brands express themselves on social media.

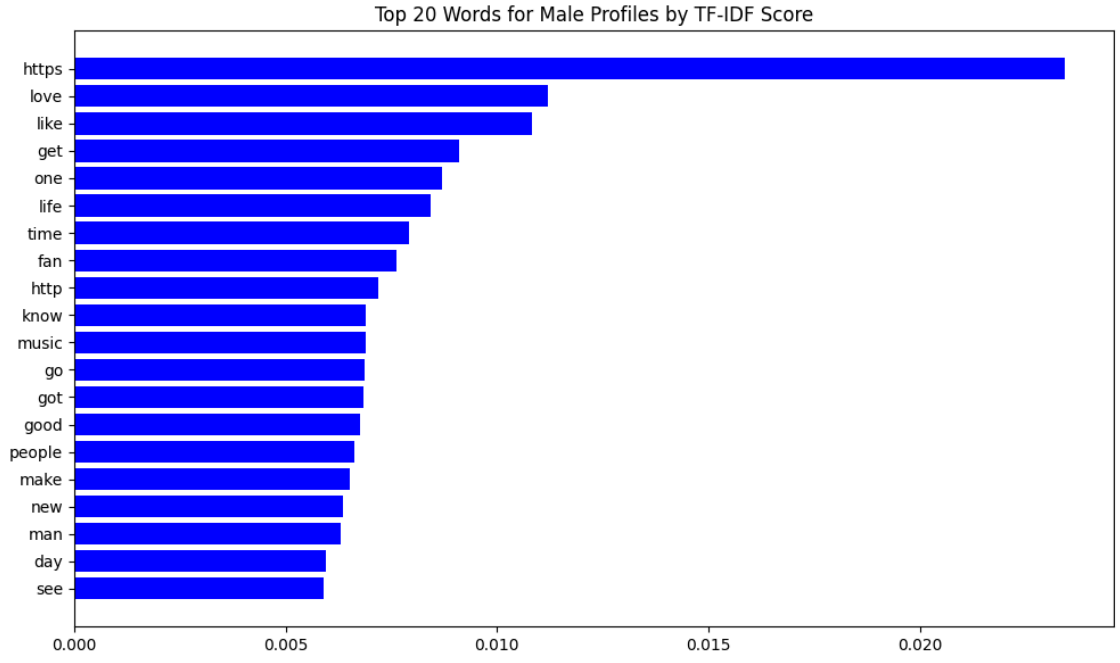

Figure 15: *Top 20 Words for “male” profiles by TF-IDF Score*

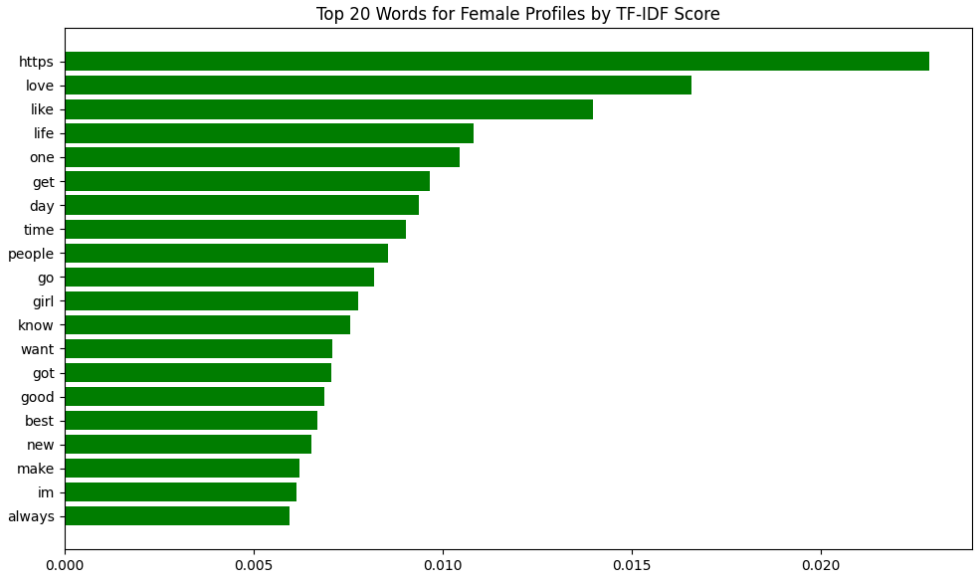

Figure 16: *Top 20 Words for "female" profiles by TF-IDF Score*

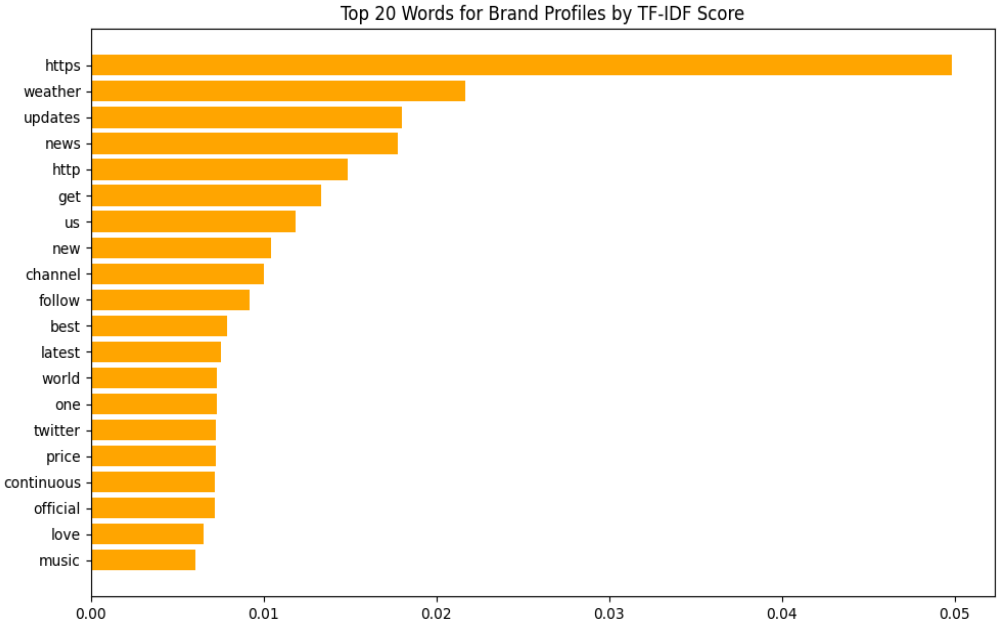

Figure 17: *Top 20 Words for "brand" profiles by TF-IDF Score*

3.4.2. Method 2 – Sentiment Analysis:

Sentiment analysis was also performed on the corpus generated in the last data preprocessing step. TextBlob, a Natural Language Processing (NLP) Python library, was used for this process (Shah, 2020). TextBlob provides two key metrics for analyzing a sentence:

a) *Polarity:* It ranges from -1 to 1, where -1 indicates negative sentiment and 1 indicates positive sentiment (Shah, 2020). Words that express negation can reverse the polarity (Shah, 2020).

b) *Subjectivity:* It is measured on a scale from 0 to 1, with higher values indicating more personal opinion and less factual content (Shah, 2020). A higher subjectivity score means the text is more opinion-based than fact-based (Shah, 2020).

Sentiment polarity and subjectivity were calculated on the corpus and the results were visualized in Figures 18 and 20 respectively. The polarity and subjectivity results were also grouped by "gender" and the mean polarity and subjectivity values were calculated for each group and illustrated in Figures 19 and 21 respectively. The sentiment analysis reveals important insights into the emotional tone and subjectivity of text content across the “male”, “female”, and “brand” profiles.

The sentiment polarity distribution figure represents the distribution of sentiment polarity scores across the corpus (refer to Figure 18). The x-axis shows the polarity ranging from -1 (most negative) to 1 (most positive), while the y-axis indicates the frequency of each polarity score (refer to Figure 18). The histogram shows that the majority of the text entries are neutral, as indicated by a significant peak around a polarity score of 0 (refer to Figure 18). However, there is also a broad distribution of positive and negative sentiment, though extreme values (close to -1 or 1) are less frequent (refer to Figure 18).

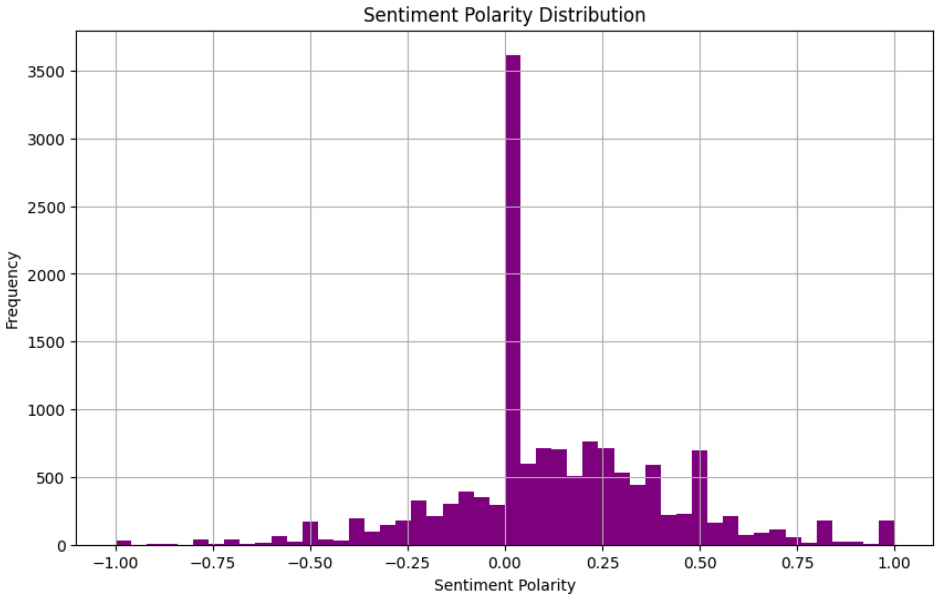

Figure 18: *Sentiment Polarity Distribution*

Figure 19 shows the average sentiment polarity for each gender category where the x-axis represents the gender categories, and the y-axis shows the average sentiment polarity. The “brand” profiles display the most positive sentiment, followed by “female” and “male” profiles respectively (refer to Figure 19). This suggests that brands are more likely to use positive language, possibly reflecting efforts to maintain a positive image or engage with

audiences favourably. The “female” and “male” profiles, on the other hand, are more free in the language they use to express themselves depending on their moods and emotions.

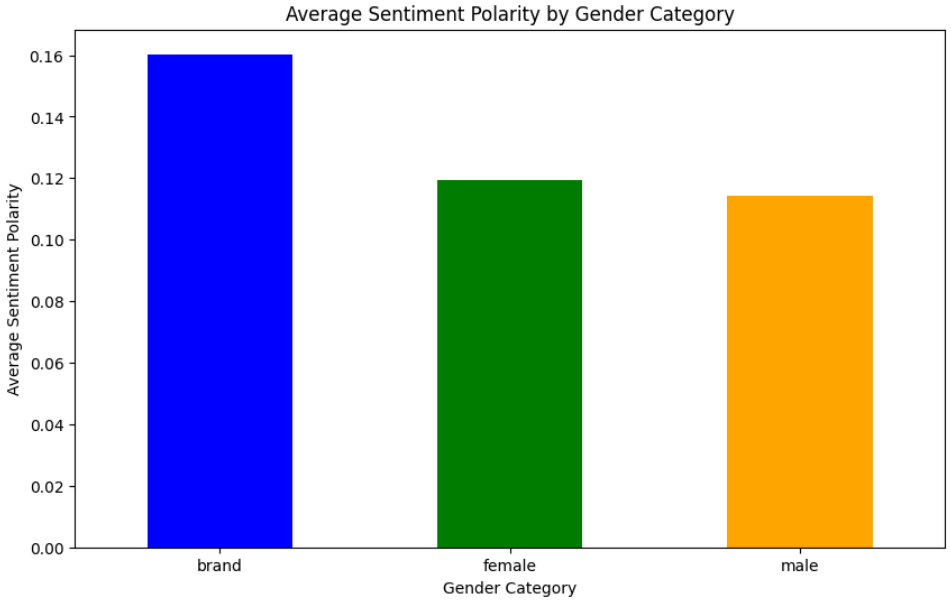

Figure 19: *Average Sentiment Polarity by “gender” Categories*

The following histogram displays the distribution of subjectivity scores across the corpus (refer to Figure 20). The x-axis shows subjectivity ranging from 0 (completely objective) to 1 (completely subjective), while the y-axis shows the frequency of text entries for each score (refer to Figure 20). The figure reveals that much of the content is objective, as demonstrated by the large number of entries with subjectivity scores close to 0 (refer to Figure 20). However, the rest of the entries are fairly spread out, indicating a mix of factual and opinion-based content (refer to Figure 20).

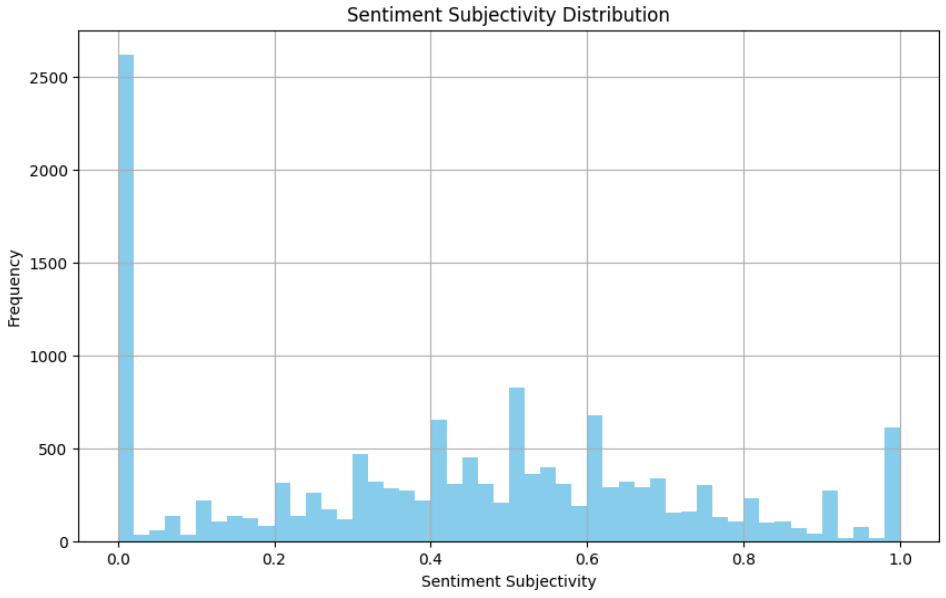

Figure 20: *Sentiment Subjectivity Distribution*

Figure 21 illustrates the average subjectivity for each gender category. The x-axis represents the gender categories, and the y-axis indicates the average subjectivity score (refer to Figure 21). When comparing subjectivity across the three gender categories, “female” profiles tend to have the highest average subjectivity, indicating a greater tendency to express opinions or personal feelings (refer to Figure 21). Meanwhile, “brand” (the most objective profile) and “male” profiles are slightly more objective in nature, though they still contain a significant level of subjectivity (refer to Figure 21).

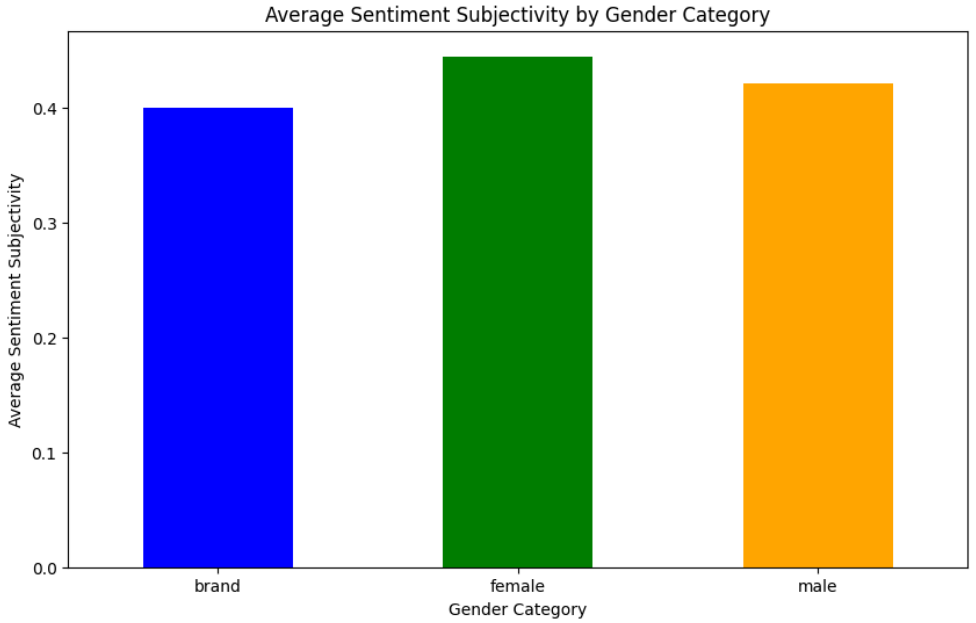

Figure 21: *Average Sentiment Subjectivity by “gender” Categories*

Overall, these findings highlight distinct communication styles across gender and profile types, with brand profiles focusing on positivity and female profiles leaning toward more subjective, personal content.

4. Text Processing 2:

Natural language processing (NLP) requires text processing, which involves a number of methods to transform unstructured text into structured data for modelling and analysis. One well-known technique for unsupervised topic modelling is Latent Dirichlet Allocation (LDA) (Blei et al. 2003). It facilitates the identification of latent themes in big text corpora, which facilitates analysis and interpretation. The usage of LDA for topic modelling and its application to extracting themes from a text dataset are covered in this research. We also go over the preparation that has to be done on the data in order to prepare it for model training.

4.1. Algorithm Consideration:

Text processing was chosen in this project because it is the process of transforming unstructured, noisy text in tweets to clean and structured text that can be presented for analysis. Most Twitter data would contain irrelevant elements such as URLs, mentions, hashtags, special characters. These need to be removed so that the actual content of the tweets can be focused on.

This is often followed by cleaning and processing of the text. This will be helpful for finding meaningful information, thus helping in the easier analysis of language patterns. Techniques include tokenization, wherein the texts are broken down into words, stopword removal or deletion of common but unimportant words, and lemmatization, which is a reduction of words to their base form-all simplifying the text to show the algorithm what it really needs to focus on: key terms.

Hence, text processing helps in identifying between human and non-human profiles based on the usage of languages within tweets. This helps the algorithm understand the context, pattern, and structure of the text more precisely, and therefore the analysis and classification will also be more accurate.

4.2. Choice of Methods & Justification:

For the task of identifying the human and non-human profiles, text processing with LDA has been used.

LDA is a very popular algorithm for topic modelling because it automatically discovers abstractive topics in large textual datasets (Blei et al. 2003). The application of LDA to the tweets will result in the text showing patterns that are indicative of non-human behaviour, such as repetitive or promotional language.

The model was supported by a gender-oriented behavioural analysis that integrated the counts of tweets and retweets, hence assisting in the explanation of the different activity levels of both human and non-human profiles. However, the core of the textual content analysis in this respect was confined to the LDA model of the tweets.

4.3. Data Preprocessing & Visualization:

Basically, improvement in the quality of an LDA model relies fundamentally on effective preprocessing. The following steps were used to clean and prepare the dataset:

1. **Dataset Cleaning**

   - *Columns Removed:* Several columns were removed that gave no meaning to the task of text-based topic modelling and behavioural analysis:

     1. Columns like user_id, username, profile creation date and other user specific identifiers which are metadata columns were removed since they do not contribute to the textual or behavioural analysis.

     2. Profile Customization Fields such as link_color, sidebar_color, and profileimage were removed as they represent cosmetic details that are unrelated to the textual or behavioural analysis.

     3. Other geographical and temporal variables were tweet_location and tweet_created. While these might serve a rich basis for analyses focused on geography and time, they are not informative for the task of topic modelling or separating human and non-human profiles.

   - *Columns Retained:* The columns text, gender, tweet count and retweet count were retained (refer to the code below). The textual content of the tweet column is the main variable used for LDA topic modelling. This column aids in the analysis of the language being used by various profiles, hence helping to identify the themes that both human and non-human users engage in. The gender column helps in understanding the behavioural difference between the human and non human profiles.

```
# Load dataset and retain relevant columns (text, gender)
data = pd.read_csv('twitter_user_data.csv',encoding='ISO-8859-1')
data = data[['text', 'gender']]  # Keep only relevant columns
data.head()
```

   - *Rows Deleted:* To keep the data on gender consistent and of the same type, the dataset was cleaned to keep only those entries that had a gender of male, female, or brand (refer to the code below). Entries with other or invalid

gender classes were removed to avoid adding superfluous information to the analysis. This ensured that the gender data were valid and focused on the relevant categories.

```
# Filter rows to retain only male, female, and brand in the gender column
data = data[data['gender'].isin(['male', 'female', 'brand'])]
```

2. **Text Preprocessing:** Once the dataset was cleaned the text column was preprocessed again:

   - *Noise removal:* Removed URLs, mentions like '@username', hashtags such as '#topic,' and non-alphabetic characters to clean the text. The goal was to make sure that only relevant text was left to use in the topic modelling (Kapadia 2019).

   - *Tokenization:* The cleaned text will be broken down into individual words or tokens to bestow the model with the capability to capture the word pattern effectively (Blei et al. 2003).

   - *Stopword Removal:* The removed stopwords are common ones, including "the", "is", "and". These words do not add any value toward the identification of topics. Stopword removal aids in narrowing the analysis to relevant words only. (Kapadia 2019).

   - *Lemmatization:* Lemmatization gives the basic form of the word, for instance changing "running" to "run," and hence can reduce word variations and make analysis more uniform (Kapadia 2019). Preprocessing, cleaning, and data formatting make the dataset suitable for the LDA model to understand meaningful patterns and underlying topics.

These preprocessing steps ensured that the dataset was properly cleaned and organised, allowing the LDA model to identify meaningful patterns and latent topics.

	text	gender	processed_text
0	Robbie E Responds To Critics After Win Against...	male	[robbie, e, responds, critic, win, eddie, edward]
1	□ÛÏIt felt like they were my friends and I was...	male	[felt, like, friend, living, story]
2	i absolutely adore when louis starts the songs...	male	[absolutely, adore, louis, start, song, hit, h...
3	Hi @JordanSpieth - Looking at the url - do you...	male	[hi, looking, url, use, dont, typically, see, ...
4	Watching Neighbours on Sky+ catching up with t...	female	[watching, neighbour, sky, catching, neighbs, ...

Figure 22: *Data after Dataset Cleaning and Text Preprocessing*

In order to understand the dataset a little bit more, the following plots were created (refer to Figures 23 and 24).

The plot in Figure 23 compares the number of rows in the dataset before and after filtering by the gender column, specifically retaining only male, female, and brand entries.

Above, within the sample, before any kind of filter, the leftmost bar represents the number of overall rows, without filtering according to the value of gender. More than approximately 20,000 rows in total were in the dataset.

The right-hand bar, shows how many rows we had left once we removed the rows for which gender was not listed as male, female or brand. This left us with about 18,000 rows.

Key Points:

Row reduction: Although the number of records in this data set was reduced, in most of the rows there did exist valid labels for males, females, or brand names in the column for gender.

Data cleaning was done in such a way that only valid entries concerning gender information remained for analysis, hence increasing the quality and relevance of the data to research issues. In other words, while some data was filtered out, most data still remained for further analyses and thus enabled finer results of the analyses of gender trends.

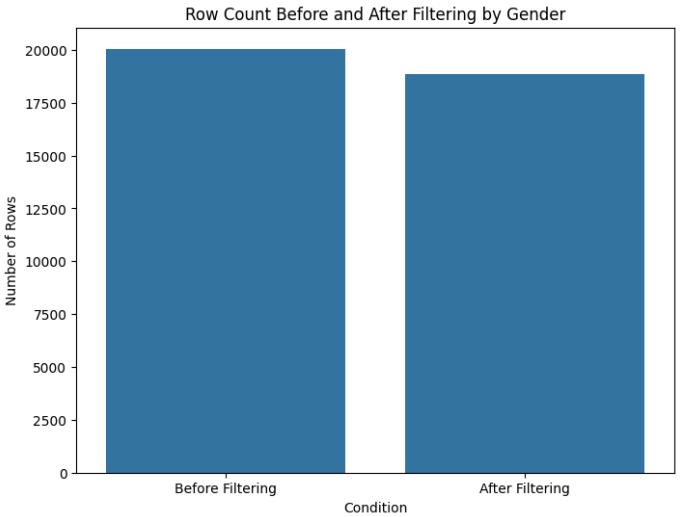

Figure 23: *Plot for Row Count Before and After Filtering By Gender*

The plot in Figure 24 shows the distribution of text length by gender, with three categories: "male" (blue), "female" (orange), and "brand" (green). The x-axis represents text length

(likely the number of characters), and the y-axis shows frequency (how often texts of different lengths occur).

Key Insights:

- **Most texts are shorter than 100 characters:**

  The highest frequencies (peaks) are found between 0 and 100 characters. This is true across all categories (male, female, brand), indicating that shorter texts are more common.

- **Distinct peaks for different genders:**

  Male (blue) texts have a strong peak at around 90 characters, followed by another near 100-110.

  Female (orange) texts tend to have slightly shorter peaks, with fewer texts between 50 and 100 characters compared to males.

  Brands (green) have lower peaks and are more evenly distributed, suggesting more diversity in text length for this category.

- **Longer texts are less common:**

  As text length increases beyond 100 characters, the frequency for all categories drops off significantly, indicating that longer texts are rare.

- **Overlap between categories:**

  There is a lot of overlap between the genders and brands, particularly in the 50-100 character range, meaning that these groups tend to write texts of similar lengths in that range.

Overall:

- Most texts are relatively short, especially for male users.
- Male users tend to write slightly longer texts than female users on average.
- Brands have a more varied range of text lengths, though they are less frequent compared to individual users.

Data Cleanup: This step ensured that only rows with valid gender information were kept for analysis, improving the quality and relevance of the data for any gender-based studies.

In summary, while some data was removed, the bulk of the dataset was retained for further analysis, ensuring more accurate results when looking at gender-related patterns.

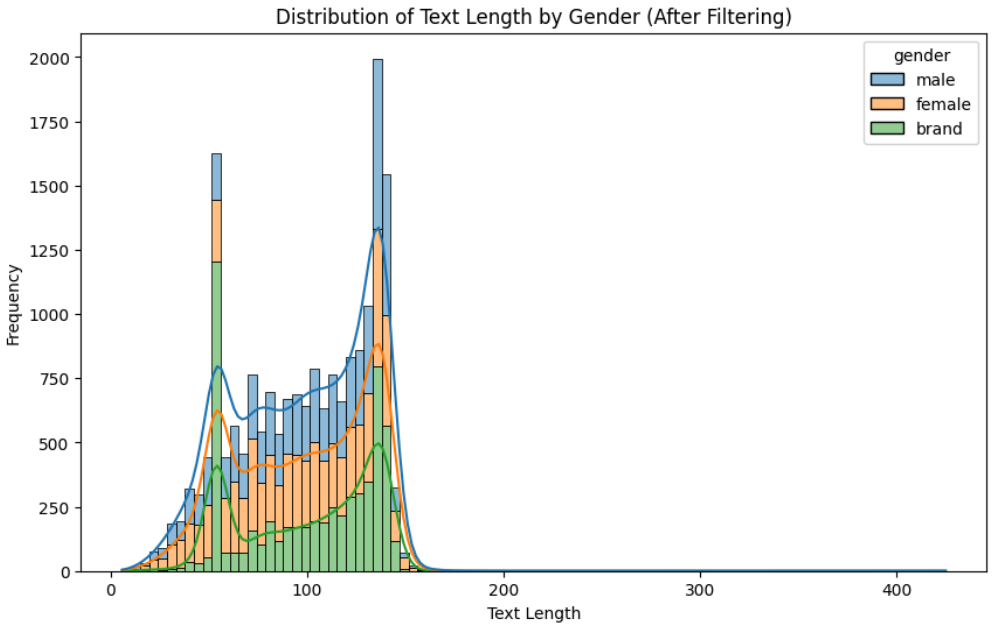

Figure 24: *Plot for distribution of text length by Gender*

4.4. Model:

The LDA model was built using Gensim's method called `LdaMulticore` (Kapadia 2019). The given model parameters were trained over the preprocessed dataset that has previously been transformed into a corpus and dictionary to be processed. Critical parameters used include:

- *Number of topics:* 10 topics were selected based on the initial processing of the dataset.
- *Pre-processing:* 10 passes of the number ensured that topic detection was consistently stable.
- *Chunksize:* By default, 100 chunks were being processed at each iteration to efficiently process the large dataset.

The model was able to generate ten topics, with each represented by a collection of words that frequently occurred together (see Figure 25):

```
Topic: 0

Words: 0.045*"go" + 0.042*"want" + 0.035*"good" + 0.028*"girl" + 0.024*"ill" +
0.023*"im" + 0.018*"vote" + 0.017*"first" + 0.016*"shit" + 0.016*"time"

Topic: 1

Words: 0.042*"come" + 0.032*"life" + 0.024*"even" + 0.022*"home" + 0.021*"didnt" +
0.019*"win" + 0.018*"play" + 0.018*"hope" + 0.017*"also" + 0.015*"ever"

Topic: 2

Words: 0.032*"would" + 0.031*"still" + 0.026*"best" + 0.026*"could" +
0.021*"happy" + 0.018*"watch" + 0.018*"please" + 0.016*"looking" + 0.014*"world" +
0.014*"he"

Topic: 3

Words: 0.090*"get" + 0.044*"dont" + 0.040*"got" + 0.026*"na" + 0.021*"every" +
0.020*"favorite" + 0.017*"stop" + 0.017*"gon" + 0.015*"visit" + 0.014*"something"

Topic: 4

Words: 0.041*"day" + 0.030*"new" + 0.022*"today" + 0.016*"last" + 0.016*"amp" +
0.015*"via" + 0.015*"need" + 0.012*"build" + 0.011*"monday" + 0.011*"make"

Topic: 5

Words: 0.036*"see" + 0.030*"cant" + 0.023*"think" + 0.018*"tell" + 0.015*"phone" +
0.015*"keep" + 0.014*"everyone" + 0.013*"try" + 0.012*"woman" + 0.012*"photo"

Topic: 6

Words: 0.055*"like" + 0.038*"people" + 0.028*"im" + 0.022*"one" + 0.021*"thing" +
0.020*"look" + 0.018*"make" + 0.016*"much" + 0.016*"time" + 0.015*"youre"

Topic: 7

Words: 0.066*"u" + 0.030*"week" + 0.026*"thats" + 0.025*"way" + 0.025*"going" +
0.023*"someone" + 0.017*"help" + 0.014*"said" + 0.014*"listen" + 0.013*"ur"

Topic: 8

Words: 0.064*"love" + 0.039*"back" + 0.039*"im" + 0.039*"know" + 0.025*"great" +
0.022*"work" + 0.015*"friend" + 0.013*"there" + 0.012*"let" + 0.012*"free"

Topic: 9

Words: 0.030*"year" + 0.024*"really" + 0.021*"game" + 0.019*"take" + 0.019*"live"
+ 0.018*"guy" + 0.018*"show" + 0.018*"well" + 0.015*"check" + 0.014*"fun"
```

Figure 25: *10 Topics Generated by the LDA Model using the Preprocessed Data*

4.5. Visualization for the Model:

1. **Word Cloud:** Figure 27 illustrates each word cloud in correspondence to one of the 10 topics generated by the LDA model shown in Figure 25. The size of each word in the cloud shows its relative importance in defining that topic.

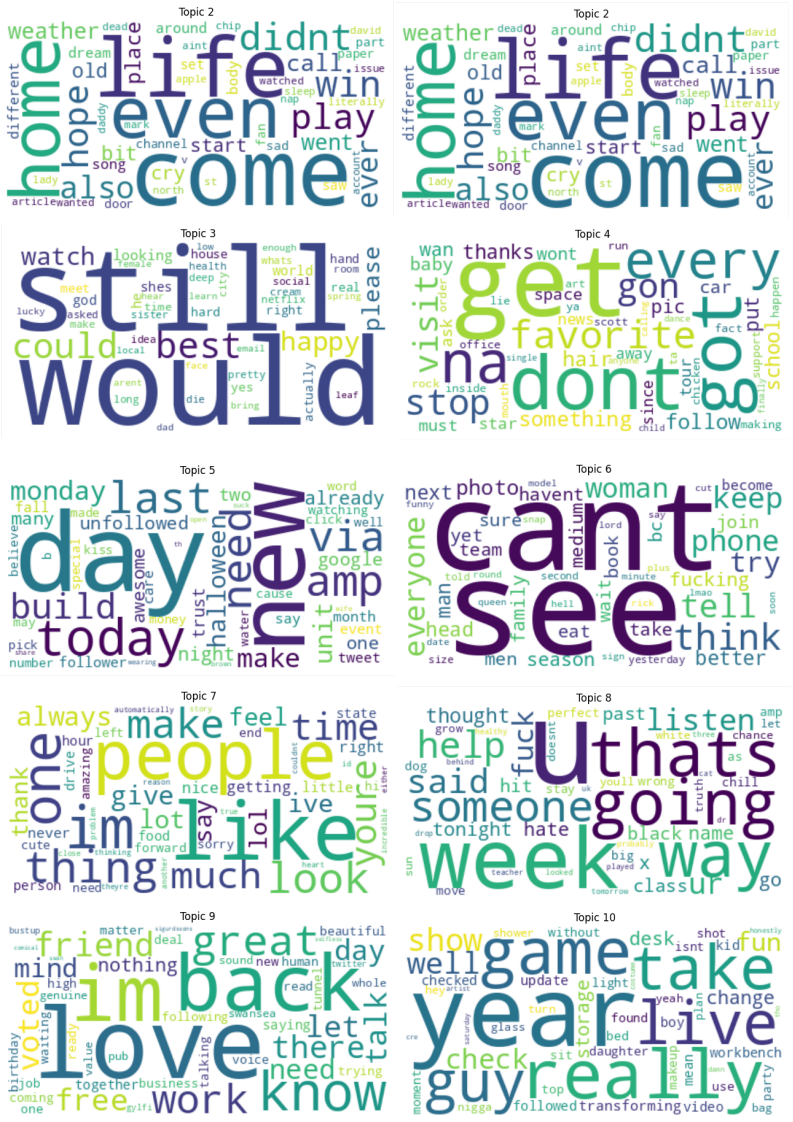

Figure 27: *Word Cloud plot for Topics*

2. **Topic Distribution Across Documents:** The plot in Figure 28 shows the distribution of dominant topic probabilities for the documents (or tweets, in this case) in the dataset. For each document (or tweet), the plot reflects the highest probability that the LDA model assigns for a specific topic. If the majority of the tweets have high probability for a dominant topic, this means the topics are well separated and distinct. If the probabilities are low, it suggests that many tweets belong to a mix of topics and may not have a clear dominant topic.

   Breakdown of Plot Features:

   - *x-axis (Dominant Topic Probability):* Represents the probability that a particular topic is the most prominent (dominant) topic in a document (or tweet). The probabilities range from about 0.1 to 0.9. This indicates that the analysis

considers topics with a wide range of being categorised as dominant in a document.

- *y-axis (Document Count):* Shows the number of documents that have a certain dominant topic probability. The count varies significantly, which is typical in topic modelling where some probabilities are more common than others.
- *Bars (Histogram):* Each bar represents the count of documents that have a dominant topic probability within a specific interval. For example, taller bars around probabilities such as 0.2, 0.4, and around 0.5-0.6 indicate these are more common probabilities for the dominant topics across the documents.
- *Curve (Kernel Density Estimate):* The smooth blue line overlaid on the histogram provides an estimate of the distribution shape of the document probabilities. The KDE helps in understanding the overall trend and density of the document probabilities without the individual bin limits of the histogram.

The key observations are:

- *Peaks at 0.3 and 0.5:* This plot is dominated by two key peaks, one at approximately 0.3 and the other roughly around 0.5. There are lots of documents for which the dominant topic probability lies between 30% and 50%, reflecting that while there is some main topic for those documents, it isn't overwhelming.
- *Major Peak at 0.5:* It peaks the highest at 0.5, which reflects a considerable number of documents with a dominant topic probability of just about 50%. This would also reflect that in most documents, the dominant topic does not strongly dominate but takes a middle lead.
- *Fewer Documents with Dominant Topic Probability Greater than 0.6:* It reflects a number of documents with a dominant topic probability greater than 0.6, depicting that in most of the documents, a single topic does not strongly dominate. A dominant topic, with more than 70% probability for the document, is hardly ever witnessed.
- *Long Tail Toward High Probabilities (0.6 - 0.9):* Not many documents have a really high probability of the dominant topic probability falling close to 0.8 or 0.9.

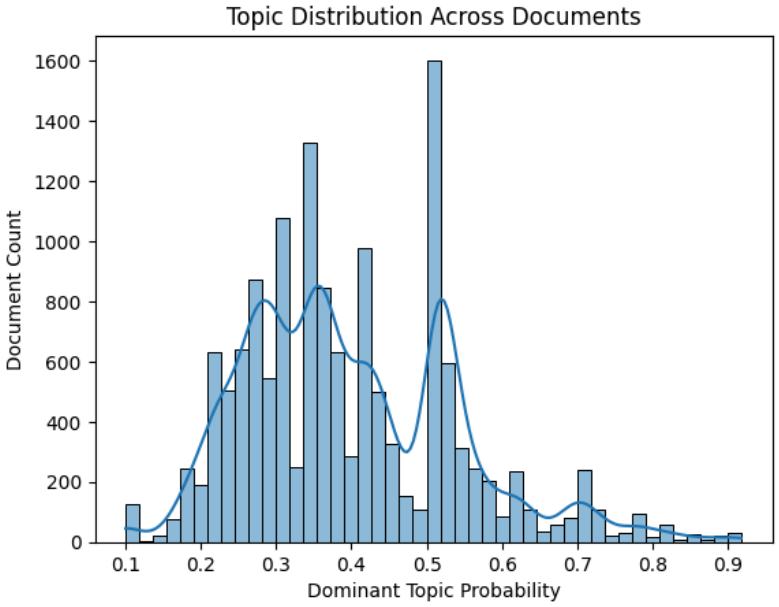

Figure 28: *Plot for Topic Distribution Across Documents*

4.6. Results & Factors Considered:

The LDA model had found 10 topics for the Twitter dataset (See Fig 4). Each of these topics was represented by some set of words that appeared in each other across the dataset, hence summarising the main themes of discussion.

The coherence score obtained for the generated topics was 0.389 (see Fig 9). It indicated how understandable and logically connected the topics are (Kapadia 2019). Although close to 0.3 is a typical score for datasets with noise, like those arising from Twitter, this is still pretty good performance in the face of the data characteristics.

```
# Compute coherence score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data['processed_text'], dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print(f'Coherence Score: {coherence_lda}')
```

```
Coherence Score: 0.38955792623259244
```

Figure 26: *Calculation of Coherence Score*

Several factors influenced the model's performance:

- **Number of Topics**: Selecting the optimal number of topics was critical to model performance. Several configurations were tried out, but 10 topics in total achieved an appropriate equilibrium between specificity and coherence.
- **Preprocessing Quality**: Proper preprocessing helped remove noise and retain only meaningful text, significantly improving the quality of the topic discovery.

- **Coherence score:** The coherence score was used to evaluate how well the topics made intuitive sense based on the words grouped together. While the score was moderate, it was acceptable for such a noisy dataset (Kapadia 2019).

In general, the LDA model provided valuable insights into the most common themes existing in a Twitter dataset, enabling systematic understanding of the topics discussed by users.

Following are the suggestion to amend the non-human and human profiles:

- **Textual Content:** The topics that were generated by the LDA model shows that the non-human profiles often focus on repetitive content. This could serve as a basis for identifying and flagging non-human profiles.
- **Tweeting Behaviour:** Non-human profiles showed higher levels of activity in terms of tweet-count and retweet_count. This behaviour can be used as a feature for distinguishing human from non-human profiles.

5. Association Rules:

This section of the report provides an overview of applying the Association rules concept on the Twitter user dataset to identify patterns and associations between different features of user profiles. The goal was to use the Apriori algorithm to generate association rules that describe relationships between these attributes, particularly between human and non-human profiles.

5.1. Algorithm Consideration:

Association rule mining involves identifying frequent itemsets and deriving rules that indicate strong relationships among the attributes of the dataset (Tan et al., 2018). The algorithm is particularly useful for datasets that can be transformed into binary or categorical variables—as done here using binning and one-hot encoding.

Association rules describe how different variables (or items) within the dataset are associated with one another. The rules consist of two parts:

1. **Antecedent:** This is the condition or itemset on the left-hand side of the rule.
2. **Consequent:** This is the itemset that is likely to occur if the antecedent is true.

Association rules are evaluated using several metrics (Tan et al., 2018):

1. **Support:** The proportion of transactions in the dataset that contain both the antecedent and consequent.
2. **Confidence:** The likelihood that the consequent occurs given the antecedent.
3. **Lift:** The ratio of observed support to expected support under the assumption that the antecedent and consequent are independent. A lift greater than 1 indicates a positive association.

By generating association rules, we can understand complex relationships between variables. For instance, in the Twitter user dataset, we see rules that associate high retweet counts with users from specific time zones or high favorites (likes) with particular user behaviors. These insights are valuable for understanding user profiles and identifying key patterns in user behavior, which is particularly useful for marketing strategies, content optimization, and user segmentation.

5.2. Choice of Model & Justification:

The Apriori model is well-suited for finding frequent patterns and association rules in large categorical datasets. It works efficiently by exploring itemsets that occur frequently in the dataset and using them to generate association rules. The algorithm leverages a “bottom-up” approach, where frequent subsets are extended one item at a time (Han, Pei and Kamber, 2011).

Key Features:

- *Pattern Mining in Categorical Data:* It is particularly effective for the Twitter dataset because it allows us to discover meaningful combinations of items (Han, Pei and Kamber, 2011).
- *Actionable Insights through Association Rules:* The ultimate goal of the analysis is to find relationships between different entities (Han, Pei and Kamber, 2011). Apriori generates association rules that help identify correlations, such as how specific user behaviors (e.g., high retweet counts or favorites) are associated with time zones, user classifications, or other features. These insights can be used to inform strategic decisions (Han, Pei and Kamber, 2011).

- *Scalability:* Given the dataset's size, which contains over 20,000 rows, Apriori is a scalable algorithm that can efficiently process large datasets by pruning non-frequent itemsets early in the process (Han, Pei and Kamber, 2011). This helps in managing computational resources effectively while generating valuable insights (Han, Pei and Kamber, 2011).
- *Interpretable Results:* The rules generated are straightforward (e.g., "If a user is from timezone X and has Y number of retweets, they are likely to have high favorites"), making it simple to derive actionable outcomes (Han, Pei and Kamber, 2011).

How It Works (Han, Pei and Kamber, 2011):

1. It starts by identifying individual items that meet a minimum support threshold.
2. Then, it combines these items into larger sets, pruning combinations that do not meet the minimum support requirement.
3. After identifying frequent itemsets, it generates association rules based on these sets.

5.3. Data Preprocessing & Visualization:

The images below depict the initial exploration performed on the data in order to ascertain their relevance to the algorithm. Since for a few features, like `fav_number` the distribution is far too spread out unevenly, proper binning techniques was needed in order to represent the data accurately. This has been explained in the next set of steps in detail.

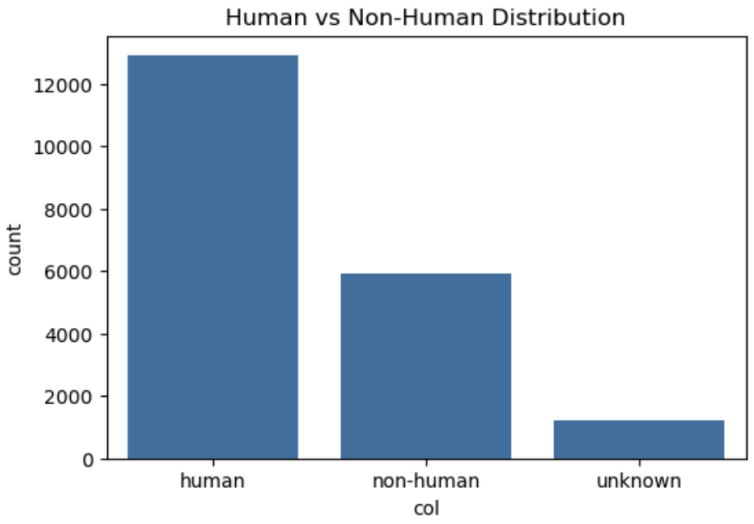

Figure 29: *Distribution of profiles identified as human vs non human*

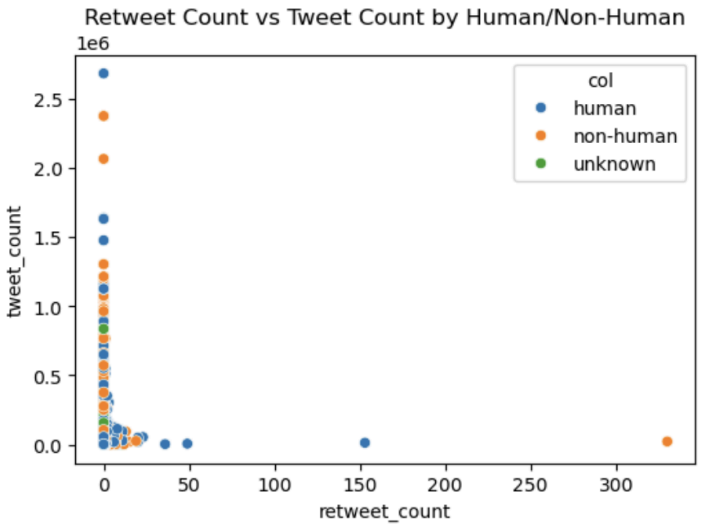

Figure 30: *Analysis of human vs non human based on tweet count and retweet count.*

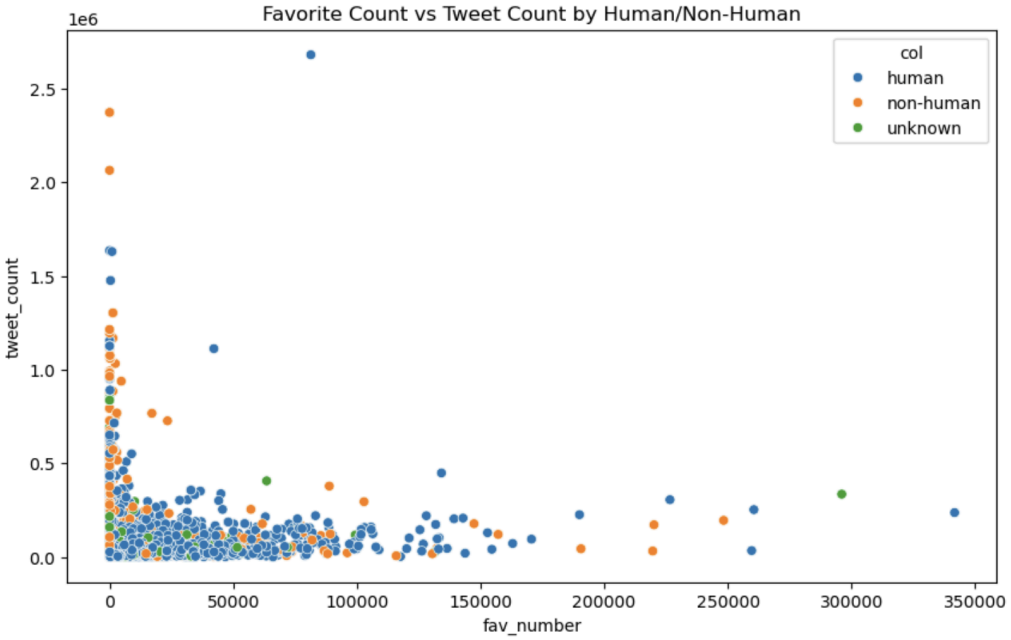

Figure 31: *Analysis of human vs non human based on tweet count and favourite number.*

5.3.1. Choice of Features:

Based on the initial exploration, the number of features was narrowed down to only those that would add value to the analysis. The following columns were considered.:

- **retweet_count:** Number of times a user's tweet was retweeted.The number of retweets indicates how much interaction or influence a user has on Twitter. Binning this column into low, medium, and high categories captures this interaction on a simplified scale.

- **fav_number:** Number of times a user's tweet was marked as a favorite. This column is essential for identifying correlations between user engagement and

profile features. For instance, highly liked tweets might be associated with profiles that are human rather than non-human.

- **tweet_count:** Number of tweets a user has made. The number of tweets a user has posted reflects their activity level on Twitter.
- **gender:** The gender of the user, categorized as male, female, brand (for non-human profiles), and unknown. This column categorizes the Twitter users as either human, non-human (brands), or unknown. The distinction between human and non-human profiles is fundamental in the dataset, as many of the relationships we are trying to uncover through the Apriori algorithm will be based on this classification.
- **description:** Whether the user has added a profile description. This feature provides insight into user engagement with their profile. Twitter users who have provided a description might behave differently from those who haven't.
- **user_timezone:** The user's timezone provides geographical insights into user behaviour, which might influence how users engage with Twitter. Certain time zones may be more associated with human users versus brand accounts.

5.3.2. Data Cleaning:

First, several steps were taken to clean and preprocess the data:

- **Handling Missing Values:** Missing values in `gender`, `description`, `tweet_location`, and `user_timezone` were filled with meaningful defaults (e.g., 'unknown' or 'No Description').
- **Human vs Non-Human Classification:** A new column `col` was created to classify users as either human (if gender is male or female) or non-human (if gender is brand).
- **Binning:** Features that represented a wide range of values were categorised into low, medium and high rather than simply treating them in a binary format. It's explained in detail in the next section.

5.3.3. Feature Binning:

Binning is a crucial part of this analysis, where continuous numerical data is transformed into categorical bins, which makes it easier to find patterns in the Apriori algorithm (Medium, 2023). The following features were binned:

- retweet_count
- fav_number
- tweet_count

Why Binning?

Binning helps simplify the analysis of continuous numerical data by grouping values into intervals or "bins" (GeeksForGeek, 2022). This transformation is useful for the Apriori algorithm, which works best with categorical data rather than continuous values.

Binning for `retweet_count`:

Based on the distribution of the `retweet_count`, the following bins were chosen:

- `0`: Accounts with no retweets.
- `1`: Accounts with a single retweet.
- `2-330`: Accounts with retweet counts between 2 and the maximum value (330).

This binning strategy groups users into low, medium, and high retweet categories, reflecting users with increasing levels of engagement.

Binning for `fav_number`:

The distribution of the `fav_number` showed a wide range of values, so the following bins were selected:

- `0`: Users with no favorite tweets.
- `1-60`: Users with a small number of favorite tweets.
- `61-max`: Users with a high number of favorites, with 60 chosen as a threshold that captures the 25th percentile.

Again, these bins categorize users into low, medium, and high levels of favoriting behavior.

Binning for `tweet_count`:

The `tweet_count` column was divided as follows:

- `1-20`: Low activity users with very few tweets.
- `21-450`: Medium activity users with a moderate number of tweets.
- `451-max`: High activity users, where max represents the largest number of tweets in the dataset.

These bins group users into different activity levels based on their tweet count.

5.3.4. One-Hot Encoding:

To prepare the data for the Apriori algorithm, categorical features were transformed using one-hot encoding. This technique converts each category of a variable into a separate binary column, which the Apriori algorithm can process (GeeksforGeeks, 2024). For example, the `fav_number_bin` column was transformed into three binary columns: `fav_number_bin_low`, `fav_number_bin_medium`, and `fav_number_bin_high`. This is illustrated in the figure below.

```
   description_bin  col_human  col_non-human  col_unknown  fav_number_bin_low  \
0                1          1              0            0                   0
1                1          1              0            0                   0
2                1          1              0            0                   0
3                1          1              0            0                   0
4                1          1              0            0                   0

   fav_number_bin_medium  fav_number_bin_high  retweet_count_bin_low  \
0                      0                    0                      0
1                      0                    1                      0
2                      0                    1                      1
3                      0                    1                      0
4                      0                    1                      0

   retweet_count_bin_medium  retweet_count_bin_high  ...  \
0                         0                       0  ...
1                         0                       0  ...
2                         0                       0  ...
3                         0                       0  ...
4                         0                       0  ...

   user_timezone_Urumqi  user_timezone_Vienna  user_timezone_Vilnius  \
0                     0                     0                      0
1                     0                     0                      0
2                     0                     0                      0
3                     0                     0                      0
4                     0                     0                      0

   user_timezone_Volgograd  user_timezone_Warsaw  user_timezone_Wellington  \
0                        0                     0                         0
1                        0                     0                         0
2                        0                     0                         0
3                        0                     0                         0
4                        0                     0                         0

   user_timezone_West Central Africa  user_timezone_Yakutsk  \
0                                  0                      0
1                                  0                      0
2                                  0                      0
3                                  0                      0
4                                  0                      0

   user_timezone_Yerevan  user_timezone_Zagreb
0                      0                     0
1                      0                     0
2                      0                     0
3                      0                     0
4                      0                     0

[5 rows x 170 columns]
```

Figure 32: *The final set of features derived after binning and one-hot encoding*

5.4. Model:

5.4.1. Frequent Itemsets:

The Apriori algorithm was applied to the one-hot encoded dataset using a minimum support threshold of 0.05. This means that any combination of features (itemsets) that appears in at least 5% of the data is considered a frequent itemset.

Support is a measure of how often a particular combination of items occurs in the dataset. A support value of 0.05 was chosen because it strikes a balance between finding meaningful patterns and filtering out too many infrequent combinations (Tan et al., 2018).

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
31	(tweet_count_bin_medium)	(user_timezone_Unknown)	0.085636	0.388928	0.053965	0.630169	1.620273	0.020659	1.652301	0.418673
229	(user_timezone_Eastern Time (US & Canada), col...	(description_bin, tweet_count_bin_high, fav_nu...	0.083292	0.578603	0.066185	0.794611	1.373325	0.017992	2.051697	0.296540
223	(user_timezone_Eastern Time (US & Canada), twe...	(description_bin, fav_number_bin_high)	0.079701	0.605686	0.066185	0.830413	1.371029	0.017911	2.325144	0.294058
225	(user_timezone_Eastern Time (US & Canada), des...	(tweet_count_bin_high, fav_number_bin_high)	0.077456	0.626085	0.066185	0.854475	1.364792	0.017690	2.569426	0.289729
220	(description_bin, user_timezone_Eastern Time (...	(fav_number_bin_high)	0.074613	0.660898	0.066185	0.887032	1.342162	0.016873	3.001757	0.275489

Figure 33: *Summarising the rules derived after applying Apriori*

5.4.2. Association Rules:

Once the frequent itemsets were identified, association rules were generated with a confidence threshold of 0.6. Confidence measures the likelihood that the presence of certain items (antecedents) leads to the presence of others (consequents). For example, if a user falls into the `high` bin for `fav_number`, how likely are they to also be classified as human?

Rules were sorted by their lift, which measures how much more likely the consequents are to appear given the antecedents, compared to their overall frequency.

Example of a Top Rule:

- *Rule:* If `sidebar_color_bin_low`, `tweet_count_bin_high`, and `col_human` are true, then the user is likely to also have `profile_yn`, `fav_number_bin_high`, `description_bin`, and `gender:confidence`.
- *Support:* 6.31% of users fall into this category.
- *Confidence:* The antecedents lead to the consequents with a probability of 70.77%.
- *Lift:* 1.8768, indicating that the antecedents increase the likelihood of the consequents by 87.68% compared to random chance.

5.5. Results:

5.5.1. Frequent Itemsets:

We visualize the top 10 frequent itemsets identified by the Apriori algorithm based on support (refer to the figure below). The frequent itemsets with higher support indicate combinations of features (or bins) that commonly appear together in the dataset. This gives insight into the most common patterns among users based on the selected features such as tweet count, favorite number, description, and human/non-human classification.

- **Top Frequent Itemsets**: From the graph, the frequent itemsets with high support involve features such as 'tweet_count_bin_high', 'description_bin', and 'fav_number_bin_high'. These high-support itemsets suggest that users with high tweet counts and high favorite numbers frequently provide descriptions for their profiles.
- **Analysis**: The visualization highlights that the most frequent itemsets are related to high activity users (with many tweets and favorites), and a significant portion of these users have provided descriptions in their profiles. This could indicate that highly engaged users are more likely to fill out their profile information, an important insight when targeting engagement strategies or understanding user behavior patterns.

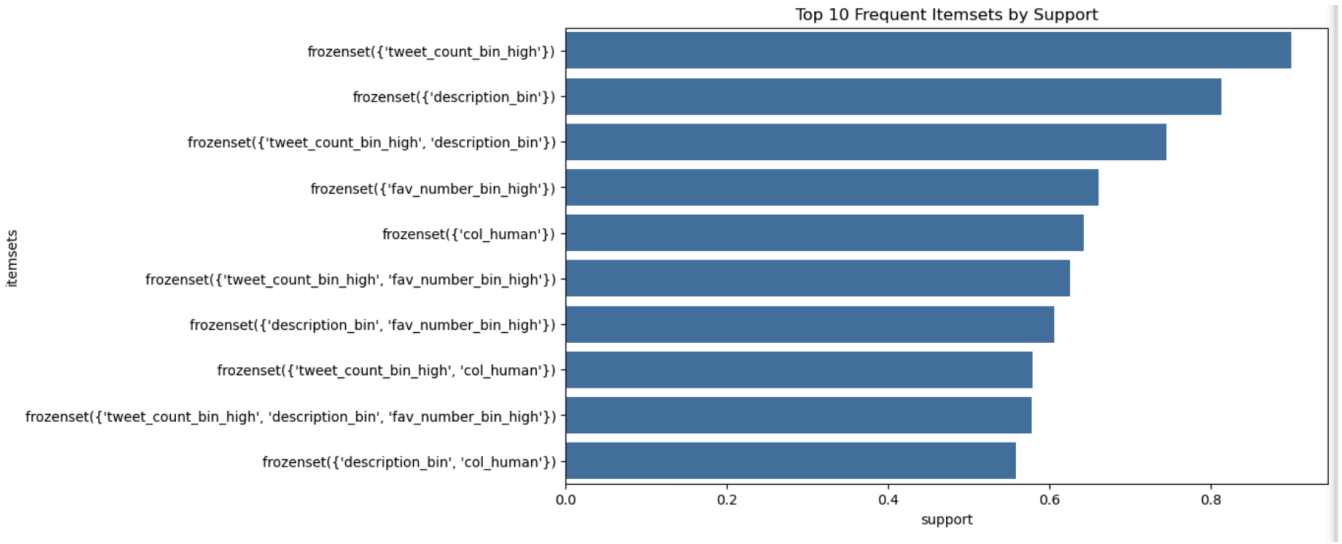

Figure 34: *Top 10 frequent itemsets*

5.5.2. Association Rules:

We visualize the top 10 rules sorted by lift (refer to Figure 35). Lift measures how much more likely the antecedent (e.g., a user with a high tweet count) will result in the consequent (e.g., high favorite count or providing a profile description), compared to if the two were independent (Tan et al., 2018).

- **High Lift Rules**: The top rule involves 'tweet_count_bin_medium' leading to 'user_timezone_Unknown'. This suggests a strong association between users with medium tweet counts and an unknown timezone, which might indicate users who do not provide certain account settings.

- **Analysis**: High-lift rules involving 'col_human' and 'user_timezone_Eastern Time (US & Canada)' suggest that human accounts in this timezone are likely to have high tweet counts and provide profile descriptions. This could be useful in identifying key characteristics of user engagement in specific regions. Additionally, the strong association between high tweet count and high favorite count for human users shows a clear pattern of activity that could guide content and marketing strategies.

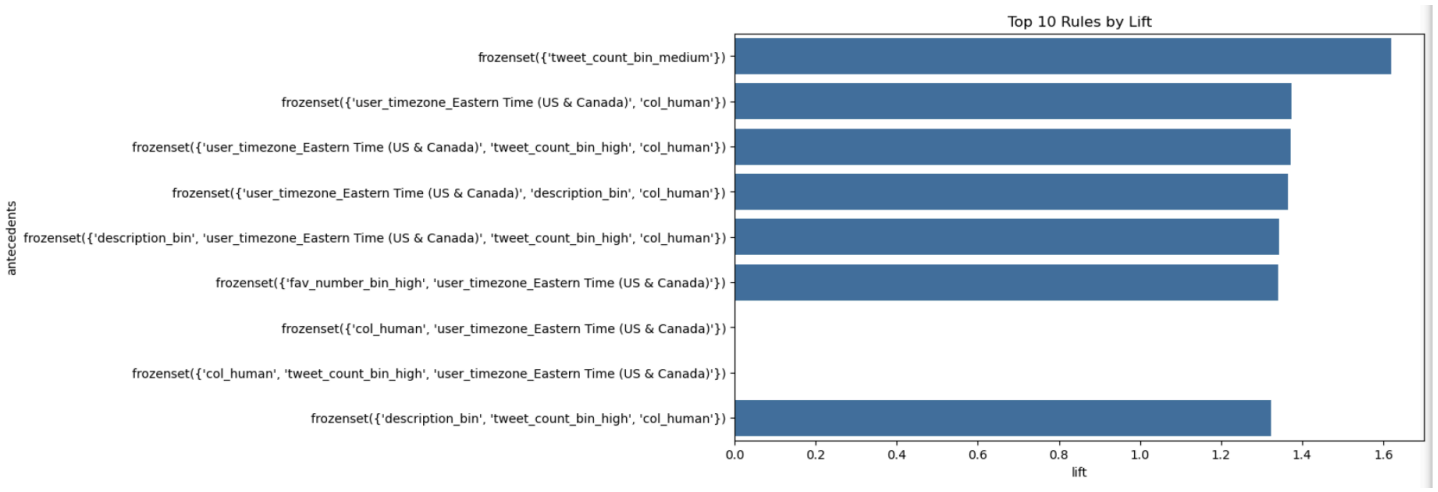

Figure 35: *Top 10 rules by lift*

Our analysis identified 108 frequent itemsets and generated 241 association rules with a minimum confidence threshold of 0.6, indicating a rich set of relationships between various user features.

The high number of rules suggests that the dataset contains a rich set of associations between user features (such as activity levels and account settings).

Top 5 Association Rules by Lift:

Among these associations, we focus on the top five rules ranked by lift. The rule with the highest lift (1.6203) reveals an intriguing connection between users with medium tweet counts and unknown timezones. This association suggests that moderately active users on Twitter are more likely to have not specified their timezone, possibly indicating a tendency among this group to provide less complete account information. This insight could be valuable for platform administrators seeking to improve user profile completeness or for researchers studying user engagement patterns.

The subsequent four top rules all involve human users located in the Eastern Time (US & Canada) timezone. These rules consistently point to a profile of highly engaged users characterized by high tweet counts, high favorite counts, and a propensity to provide profile descriptions. The recurrence of these features across multiple rules, each with a support of 0.0662, highlights the robustness of this behavioral pattern.

For instance, one rule (lift: 1.3733) indicates that human users in the Eastern Time zone are likely to have high tweet counts, high favorite counts, and provide profile descriptions. Another rule (lift: 1.3710) further refines this profile, showing that among these users, those with high tweet counts are very likely to also have high favorite counts and provide

descriptions. This consistent pattern suggests a segment of highly active and engaged users within this geographical area.

The strength of these associations is further emphasized by their high confidence values, ranging from 0.7946 to 0.8870.

The contrast between the first rule (regarding medium tweet counts and unknown timezones) and the others highlights the diversity of user types on the platform. This understanding could inform strategies for improving user experience across different engagement levels, from encouraging more profile completeness among moderately active users to providing advanced features for highly engaged users.

Rule 1:

- **Antecedents:** ['tweet_count_bin_medium']
- **Consequents:** ['user_timezone_Unknown']
- **Support:** 0.0540, Confidence: 0.6302, Lift: 1.6203
- **Interpretation:** Medium tweet count users are strongly associated with having an unknown timezone.

Rule 2:

- **Antecedents:** ['user_timezone_Eastern Time (US & Canada)', 'col_human']
- **Consequents:** ['description_bin', 'tweet_count_bin_high', 'fav_number_bin_high']
- **Support:** 0.0662, Confidence: 0.7946, Lift: 1.3733
- **Interpretation:** Human users in Eastern Time (US & Canada) are likely to have high tweet and favorite counts and provide descriptions.

Rule 3:

- **Antecedents:** ['user_timezone_Eastern Time (US & Canada)', 'tweet_count_bin_high', 'col_human']
- **Consequents:** ['description_bin', 'fav_number_bin_high']
- **Support:** 0.0662, Confidence: 0.8304, Lift: 1.3710

- **Interpretation:** Human users in Eastern Time with high tweet counts are very likely to have descriptions and high favorite counts.

Rule 4:

- **Antecedents:** ['user_timezone_Eastern Time (US & Canada)', 'description_bin', 'col_human']
- **Consequents:** ['tweet_count_bin_high', 'fav_number_bin_high']
- **Support:** 0.0662, Confidence: 0.8545, Lift: 1.3648
- **Interpretation:** Descriptions provided by human users in Eastern Time strongly correlate with high tweet and favorite counts.

Rule 5:

- **Antecedents:** ['description_bin', 'user_timezone_Eastern Time (US & Canada)', 'tweet_count_bin_high', 'col_human']
- **Consequents:** ['fav_number_bin_high']
- **Support:** 0.0662, Confidence: 0.8870, Lift: 1.3422
- **Interpretation:** Users with high tweet counts and descriptions in Eastern Time (US & Canada) almost always have high favorite counts.

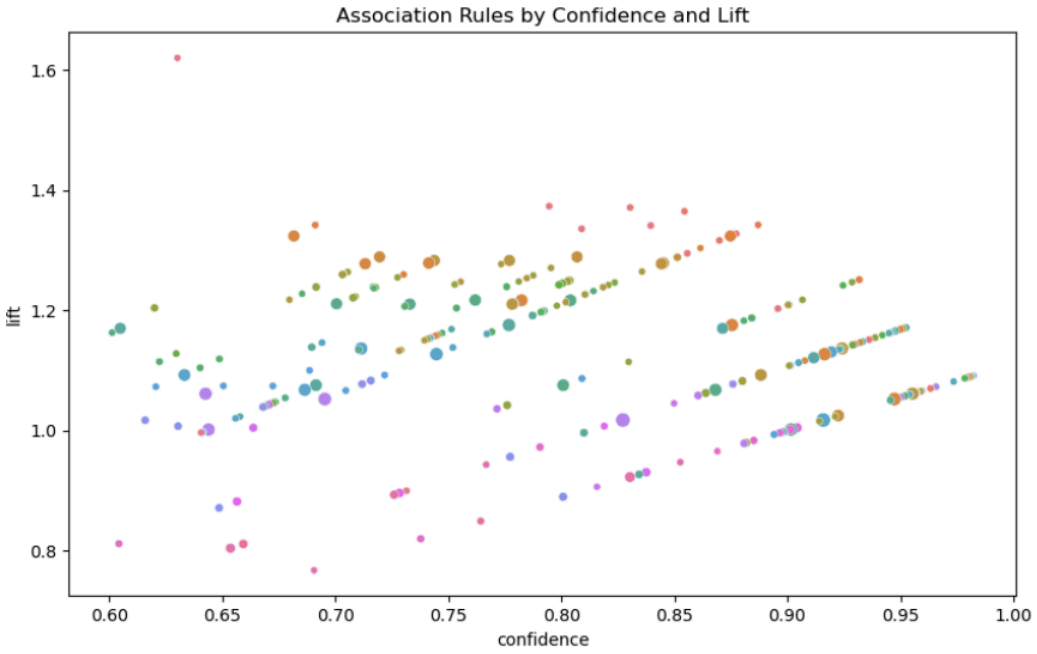

Figure 36: *Association Rules by Confidence and Lift*

Figure 37: *Lift and Confidence visualised (along with the legend)*

```
Summary of Findings:
The Apriori algorithm identified 108 frequent itemsets.
241 association rules were generated with confidence above 0.6.

Top 5 rules by lift:
Rule 32: ['tweet_count_bin_medium'] -> ['user_timezone_Unknown']
   Support: 0.0540, Confidence: 0.6302, Lift: 1.6203

Rule 230: ['user_timezone_Eastern Time (US & Canada)', 'col_human'] -> ['description_bin', 'tweet_count_bin_high', 'fav_number_bin_high']
   Support: 0.0662, Confidence: 0.7946, Lift: 1.3733

Rule 224: ['user_timezone_Eastern Time (US & Canada)', 'tweet_count_bin_high', 'col_human'] -> ['description_bin', 'fav_number_bin_high']
   Support: 0.0662, Confidence: 0.8304, Lift: 1.3710

Rule 226: ['user_timezone_Eastern Time (US & Canada)', 'description_bin', 'col_human'] -> ['tweet_count_bin_high', 'fav_number_bin_high']
   Support: 0.0662, Confidence: 0.8545, Lift: 1.3648

Rule 221: ['description_bin', 'user_timezone_Eastern Time (US & Canada)', 'tweet_count_bin_high', 'col_human'] -> ['fav_number_bin_high']
   Support: 0.0662, Confidence: 0.8870, Lift: 1.3422
```

Figure 38: *Top 5 rules by lift*

6. Classification:

6.1. Algorithm Consideration:

Classification is done to assign items or instances into predefined categories or labels based on their attributes or features. In the context of data analysis or machine learning, the purpose of classification is to predict the label or class of new, unseen instances. Here are key reasons for performing classification:

1. **Predicting Categorical Outcomes**: Classification models are used to predict discrete outcomes or categories.
2. **Simplifying Decision-Making**: Classification helps in simplifying complex data by dividing it into clearly defined groups. This can assist in decision-making processes, for example, identifying user demographics for targeted marketing strategies.
3. **Identifying Patterns**: Classification helps reveal hidden patterns and relationships in data that may not be apparent from simple descriptive statistics. These patterns can be used to gain insights, such as understanding how different features (like tweet count, favorites) relate to user categories (such as gender, account type).
4. **Handling Large-Scale Data**: As datasets grow in size, manually labeling or grouping data becomes infeasible. Classification algorithms can efficiently handle large datasets and provide accurate results even in complex scenarios, such as social media analysis.

In summary, classification is essential when the goal is to categorize data points into defined classes based on their attributes, leading to more structured, interpretable, and actionable insights.

6.2. Choice of Model & Justification:

In this report, we explore three classification algorithms for gender prediction based on Twitter data: Logistic Regression, Random Forest, and Gradient Boosting. The goal was to determine which model performs best in predicting gender based on a user's tweet count, retweet count, and favorite number. Classification algorithms were chosen based on their ability to handle both structured data and multiclass classification problems effectively.

- **Logistic Regression** was selected for its simplicity and interpretability in binary or multiclass classification tasks. It is often used for binary or multiclass classification problems. It works by modeling the relationship between a dependent variable and one or more independent variables by estimating probabilities using a logistic function. This model was chosen as a baseline because of its efficiency and ease of interpretability. However, it may struggle with more complex datasets involving non-linear relationships. Studies have shown that Logistic Regression performs well with linearly separable data but can be limited in capturing non-linear patterns (Hosmer et al., 2013).

- **Random Forest** was chosen for its ability to handle non-linear data relationships and its robustness to overfitting. It is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of individual trees. This algorithm was selected due to its robustness in handling complex data with many features and its tendency to not overfit when enough trees are used. It also provides insights into feature importance, making it a valuable choice for understanding which features contribute most to the prediction (Breiman, 2001). Research highlights that Random Forest's ability to generalize well across different domains stems from its design as a bagging technique (Biau, 2012) .

- **Gradient Boosting** was included for its high predictive accuracy, although it tends to be computationally intensive and slower to train. It builds an ensemble of weak learners, typically decision trees, in a sequential manner to minimize errors from previous models. It was selected for its ability to achieve high accuracy by focusing on difficult-to-classify instances. While Gradient Boosting can outperform other methods in terms of predictive performance, it comes with a higher computational cost. Friedman (2001) describes the advantages of this method for difficult prediction tasks, emphasizing its performance in competitive machine learning

environments . However, as noted by Chen and Guestrin (2016), the computational burden is a key limitation when compared to other algorithms like Random Forest .

6.3. Data Preprocessing & Visualization:

The dataset was cleaned and preprocessed to handle missing values, encode categorical variables, and scale numerical features. A total of 18,836 entries were considered after cleaning (Pydata.org, n.d).

6.3.1. Preprocessing Steps:

Preprocessing is crucial for preparing the dataset for analysis and modeling. Here's a detailed breakdown of the preprocessing steps applied in the analysis:

1. **Data Cleaning:**

   - *Handling Missing Values:* Removed rows with missing values in the gender and profile_yn columns to ensure completeness and reliability of the data used in the model. In addition to removing rows with NaN values in the gender column, rows with the value "unknown" were also removed, as this is considered missing data. Checked for any other columns with missing values and handled them appropriately (e.g., filling, removing) as necessary.

   - *Data Type Conversion:* Ensured that all columns were in the correct data type for analysis. For instance, converting categorical variables into numerical values using encoding techniques.

2. **Feature Engineering:**

   - *Creation of New Features:*

     ➔ **Tweet-to-Retweet Ratio:** Added as a new feature to capture the ratio of tweets to retweets, providing insight into user activity and engagement. It is calculated as:

```
data['tweet_to_retweet_ratio'] = data['tweet_count'] / (data['retweet_count'] + 1)
```

     The addition of 1 in the denominator avoids division by zero.

➔ **Description Length**: Added as a new feature to capture the length of profile descriptions, which may provide information about user profile detail. It is calculated as:

```
data['description_length'] = data['description'].apply(lambda x: len(str(x).split()) if pd.notna(x) else 0)
```

➔ **Dropping Irrelevant Columns:** For this analysis and modeling task, certain columns were retained while others were removed to focus on the most relevant data for building a predictive model. The decision to keep or drop columns was based on their relevance to the task and their potential contribution to the model's performance.

3. Columns Removed:

- *_unit_id, _golden, _unit_state, _trusted_judgments, _last_judgment_at:* These columns are metadata related to the dataset itself, describing the status of the data collection process (e.g., whether the unit was a "golden" example used for quality control). These are not relevant for the predictive model.

- *gender:* While this column provides the confidence score of the gender prediction, we removed it because the task is to predict gender from the other features, and including this column would bias the model. The model should not use pre-existing confidence scores of gender.

- *profile_yn, profile_yn_gold:* Similar to gender, these columns provide confidence scores or gold standard annotations for the profile_yn column, which could lead to bias in the model if used as features.

- *created, name, profileimage:* These columns were removed because they either contain unique identifiers or visual information (like profile images) that are not relevant for the model's task. name and profileimage may provide personal information that is not necessary or ethical for this analysis. The created column contains timestamps that do not contribute meaningfully to predicting gender.

- *link_color, sidebar_color:* These columns represent the color scheme used on the user's Twitter profile. While these could be interesting features in some analyses, they are not directly relevant to predicting gender in this case.

- *text:* The content of the user's tweets is also removed because the model is focusing on numerical and categorical user behavior and profile data. While tweet text could be useful for natural language processing tasks, it was excluded here to simplify the analysis and focus on profile features.

- *tweet_coord, tweet_created, tweet_id, tweet_location, user_timezone:* These columns contain geographical, time, and location-based metadata related to tweets. They are not essential for the current modeling task, which focuses more on profile and engagement data rather than the specifics of individual tweets or their origins.

Reason for Removal: The columns that were removed either contained information that was irrelevant to the prediction task, could bias the model, or were personal data not necessary for analysis. The goal was to focus on user behavior, engagement metrics, and high-level profile information to build a model that could predict gender without introducing noise or bias from irrelevant or overly detailed features.

4. **Encoding Categorical Variables:**

- *Label Encoding:* Encoded categorical variables (`gender` and `profile_yn`) into numerical format using `LabelEncoder` to facilitate machine learning model training. This step transforms categorical labels into numeric values:

```
label_encoder = LabelEncoder()
data_cleaned['gender_encoded']
label_encoder.fit_transform(data_cleaned['gender'])
data_cleaned['profile_yn_encoded']
label_encoder.fit_transform(data_cleaned['profile_yn'])
```

5. **Feature Scaling:**

- *Standardization:* Applied standardization to numeric features for models that require feature scaling (e.g., Logistic Regression). This step ensures that features contribute equally to the model by scaling them to have zero mean and unit variance:

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

6. **Splitting Data:**

- *Train-Test Split:* Split the dataset into training and test sets to evaluate the model's performance on unseen data. Used a 70-30 split ratio:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

7. **Feature Selection:**

- *Selecting Relevant Features:* Choose specific features for modeling based on their relevance to the prediction of the target variable (`gender`). The selected features were chosen after feature importance was conducted (this is discussed in section 5.5.4 in more details). These features include:

```
features = ['fav_number', 'tweet_count', 'retweet_count', 'description_length', 'tweet_to_retweet_ratio', 'profile_yn_encoded']
```

8. **Columns Kept:**

- *fav_number:* This column represents the number of times a user has favorited tweets. This is useful for understanding user engagement, which may be related to their activity and preferences.

- *tweet_count:* The total number of tweets a user has posted. This provides a measure of user activity, which can be an important factor when building a profile or predicting behavior patterns.

- *retweet_count:* The number of retweets by the user. Like tweet_count, this provides insight into user activity and may be an important factor in modeling.

- *description_length:* This was derived by measuring the length of the user's profile description (from the description column). Profile descriptions might provide useful information about the user's engagement and interest, indirectly contributing to understanding user characteristics.

- *tweet_to_retweet_ratio:* This derived feature calculates the ratio of tweets to retweets for each user. It provides a quantitative measure of user behavior, indicating how much original content they create versus sharing others' content. This ratio can help distinguish different user types or behavioral patterns.

- *gender:* This is the target variable for classification in this task. The gender of the user is what the model is trying to predict, so this column is essential.
- *profile_yn:* Indicates whether the user has a valid profile or not. This is important because the presence of a profile may affect user behavior and engagement, and it could be used to filter out incomplete or low-confidence data.

Summary

The preprocessing steps involved cleaning the data, creating new informative features, encoding categorical variables, scaling features, splitting the data into training and testing sets, and selecting relevant features for model training. These steps are essential to ensure that the data is in a suitable format for analysis and that the models can be trained effectively to make accurate predictions.

6.3.2. Exploratory Data Analysis (EDA):

1. **Summary Statistics**: Summary statistics for tweet counts, retweet counts, and favorite counts were generated, revealing large variability across users.
    - Mean tweet count: 39,135, standard deviation: 119,130.
    - Mean favorite count: 4,413, standard deviation: 12,468.

	fav_number	tweet_count	retweet_count	description_length	\
count	18836.000000	1.883600e+04	18836.000000	18836.000000	
mean	4413.461563	3.913570e+04	0.082502	10.936133	
std	12468.532705	1.191306e+05	2.732317	8.805542	
min	0.000000	1.000000e+00	0.000000	0.000000	
25%	13.000000	2.399750e+03	0.000000	3.000000	
50%	482.500000	1.131250e+04	0.000000	10.000000	
75%	3375.500000	3.979350e+04	0.000000	18.000000	
max	341621.000000	2.680199e+06	330.000000	48.000000	

	tweet_to_retweet_ratio	gender_encoded	profile_yn_encoded
count	1.883600e+04	18836.000000	18836.0
mean	3.842554e+04	1.013379	0.0
std	1.186439e+05	0.802592	0.0
min	5.000000e-01	0.000000	0.0
25%	2.332750e+03	0.000000	0.0
50%	1.090350e+04	1.000000	0.0
75%	3.940550e+04	2.000000	0.0
max	2.680199e+06	2.000000	0.0

Figure 39: *Summary Statistics*

2. **Correlation Matrix:** A heatmap was plotted to visualize correlations between numerical variables. No strong correlations were found.

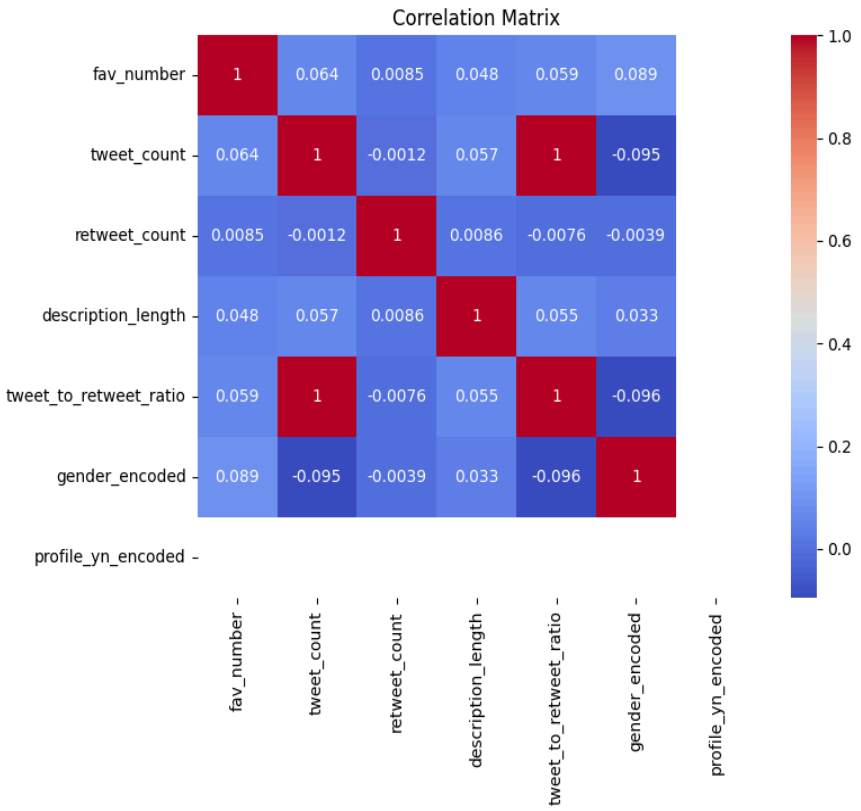

Figure 40: Correlation Matrix

3. **Gender Distribution:** A count plot showed the gender distribution, with roughly equal numbers of male, female, and brand entries.

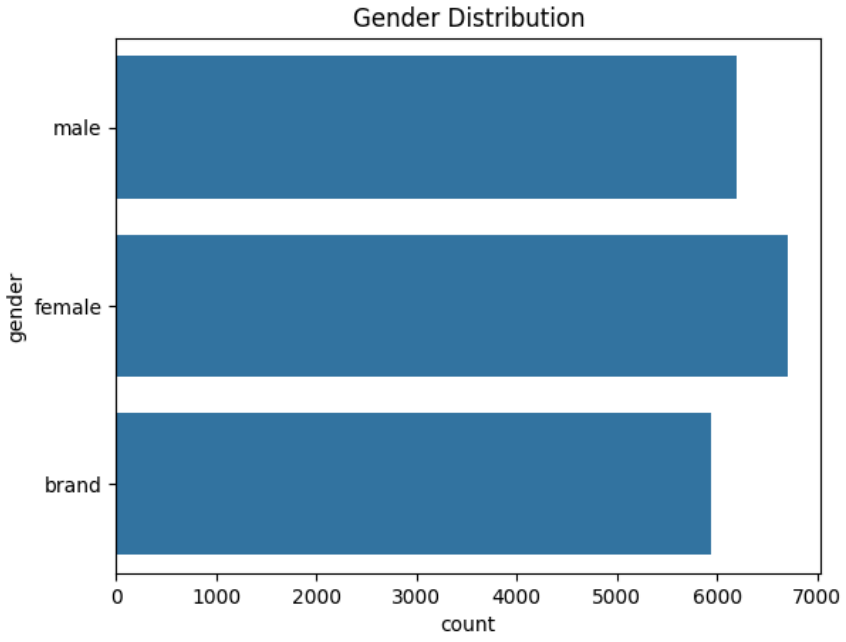

Figure 41: Gender Distribution

4. **Box Plots:**

   Boxplots were used to visualize the distribution of tweet_count across different genders, which revealed some interesting patterns in user behavior. The variance in behavior refers to the differences in tweeting frequency between males, females, and other gender categories.

- *Females tend to have the lowest tweet count:* The boxplot for females shows a lower median and a more compressed distribution, indicating that female users, on average, tweet less frequently compared to males. This could suggest that females are generally less active on the platform in terms of posting tweets.

- *Males have a higher tweet count:* The boxplot for males typically shows a higher median and a wider interquartile range (IQR), meaning that male users tend to tweet more often, and there is more variability in their tweeting behavior. Some males may tweet very frequently, while others tweet at a more moderate pace.

- *Outliers:* The presence of outliers in the boxplots suggests that some users tweet far more frequently than the general user population, particularly in the male category. These extreme cases (outliers) indicate users who may be highly active on the platform, skewing the tweet count distribution upwards.

This variance in tweeting behavior by gender indicates that males and females interact with Twitter differently, which may serve as a useful feature in predicting gender within the classification model.

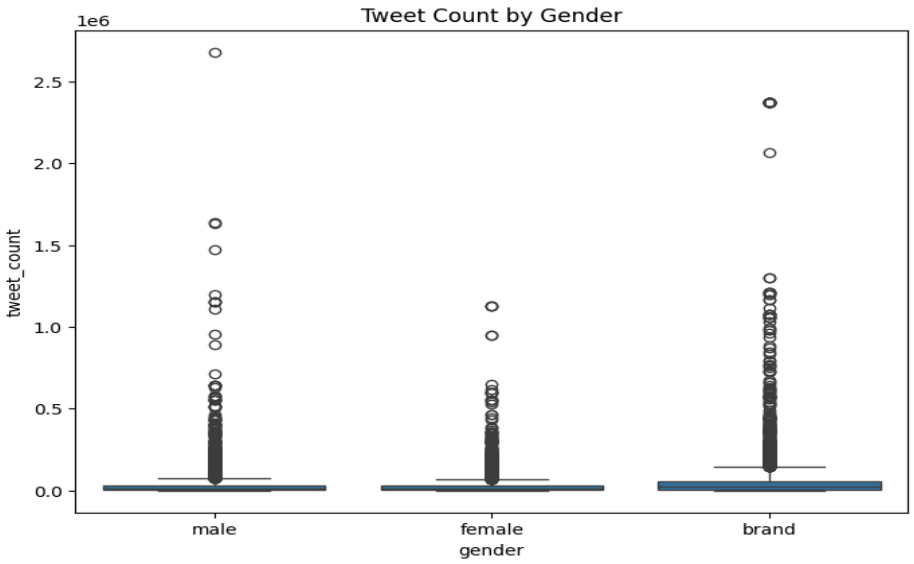

Figure 42: *Tweet Count by Gender*

6.4. Model:

The figure presents the performance metrics of three classification models—Logistic Regression, Random Forest, and Gradient Boosting—evaluated using precision, recall, F1-score, and accuracy. The models are tested on three classes: *brand*, *female*, and *male*.

```
Logistic Regression Classification Report:
              precision    recall  f1-score   support

       brand       0.54      0.33      0.41      1759
      female       0.42      0.68      0.52      2000
        male       0.37      0.25      0.30      1892

    accuracy                           0.43      5651
   macro avg       0.44      0.42      0.41      5651
weighted avg       0.44      0.43      0.41      5651


Random Forest Classification Report:
              precision    recall  f1-score   support

       brand       0.61      0.58      0.60      1759
      female       0.48      0.52      0.50      2000
        male       0.43      0.42      0.42      1892

    accuracy                           0.50      5651
   macro avg       0.51      0.51      0.51      5651
weighted avg       0.51      0.50      0.50      5651


Gradient Boosting Classification Report:
              precision    recall  f1-score   support

       brand       0.64      0.62      0.63      1759
      female       0.52      0.57      0.54      2000
        male       0.42      0.38      0.40      1892

    accuracy                           0.53      5651
   macro avg       0.53      0.53      0.53      5651
weighted avg       0.52      0.53      0.52      5651

Logistic Regression Accuracy: 0.42930454786763406
Random Forest Accuracy: 0.5036276765174306
Gradient Boosting Accuracy: 0.5252167757918952
```

Figure 43: ***Models'*** *Classification Reports*

6.4.1. Model 1 - Logistic Regression:

Logistic Regression was trained on the scaled training set. The model's simplicity makes it a good starting point, though it is not expected to perform as well on complex datasets with non-linear relationships.

Approach: A baseline linear classifier to provide a benchmark for other, more complex models.

Strengths: Simple to interpret and effective for binary and multiclass classification when features have a linear relationship with the target.

Accuracy: 42.9%

Weaknesses: Struggled to differentiate between the `male` and `female` classes, particularly because of its linear nature. Struggles with non-linear relationships and often performs poorly when there are interactions between features.

Factors: Performs best for simple and linearly separable data but falls short on capturing the complexity of this dataset.

6.4.2. Model 2 - Random Forest:

Random Forest was trained on the unscaled dataset. It performed better than Logistic Regression due to its ability to capture non-linear patterns in the data. This model provides a good balance between bias and variance.

Approach: An ensemble-based decision-tree method that aggregates the predictions of multiple trees to provide a more accurate and stable prediction.

Strengths: Handles non-linear relationships, robust to outliers, and less prone to overfitting than single decision trees. Improved performance over logistic regression, especially with the `brand` class, where precision reached 61%. Shows better generalization but still struggles with the `male` class.

Weaknesses: Requires more computational resources and might be slower in inference compared to simpler models like logistic regression.

Accuracy: 50.3%

Factors: Benefited from its ability to handle non-linear features and interactions, resulting in better predictions.

6.4.3. Model 3 - Gradient Boosting:

Gradient Boosting was applied to improve predictive performance, particularly for difficult-to-classify instances. It achieved the highest accuracy among the models tested but required more computational resources.

Approach: An advanced ensemble method that builds models iteratively, improving accuracy by focusing on misclassified data from the previous iteration.

Strengths: High accuracy, works well with both structured and unstructured data, and can handle class imbalance better than other methods. Highest performance among the three

models, particularly effective in classifying the `brand` class with 64% precision and `female` class with a 57% recall.

Weaknesses: More prone to overfitting if not tuned properly, and computationally expensive.

Accuracy: 52.5%

Factors: Captured complex relationships in the data, particularly in imbalanced classes. However, it is more computationally intensive and slower to train.

6.5. Results:

All three models were evaluated using accuracy, precision, recall, and F1-score, with confusion matrices plotted for each. Overall, Gradient Boosting showed the best performance with an accuracy of 52.5%, while Logistic Regression, the simplest model, had the lowest accuracy (42.9%).

6.5.1. Confusion Matrices:

A confusion matrix is a tabular representation that displays the performance of a classification algorithm by comparing the predicted classes against the actual classes. It provides a breakdown of how many instances of each class were correctly or incorrectly classified. For each of the models (Gradient Boosting, Random Forest, and Logistic Regression), confusion matrices were computed. Confusion matrices are a standard tool in machine learning, providing insight into not just overall accuracy, but also into the types of errors being made by the model (Fawcett, 2006).

6.5.1.1. Gradient Boosting Confusion Matrix:

The confusion matrix for Gradient Boosting shows the number of correctly classified instances and the errors made by the model:

Diagonal values represent the correct predictions for each class (`brand`, `female`, `male`).

Off-diagonal values represent the misclassifications.

The Gradient Boosting model had higher precision for the `brand` and `female` classes but struggled with the `male` class, leading to a lower recall for that group.

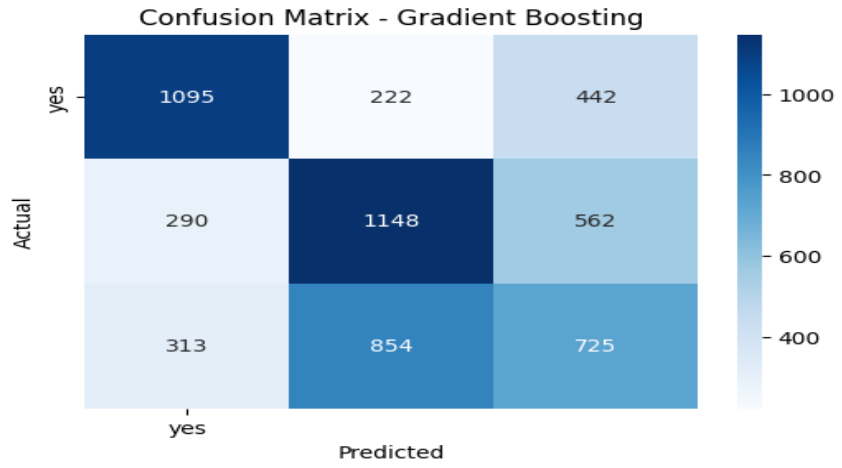

Figure 44: Confusion Matrix - Gradient Boosting

6.5.1.2. Random Forest Confusion Matrix:

The confusion matrix for Random Forest reveals a somewhat similar pattern. Random Forest achieved moderate accuracy in predicting the `brand` and `female` categories, with a higher number of misclassifications in the `male` category. The model demonstrated slightly better generalization than Logistic Regression but fell short of the Gradient Boosting model's performance.

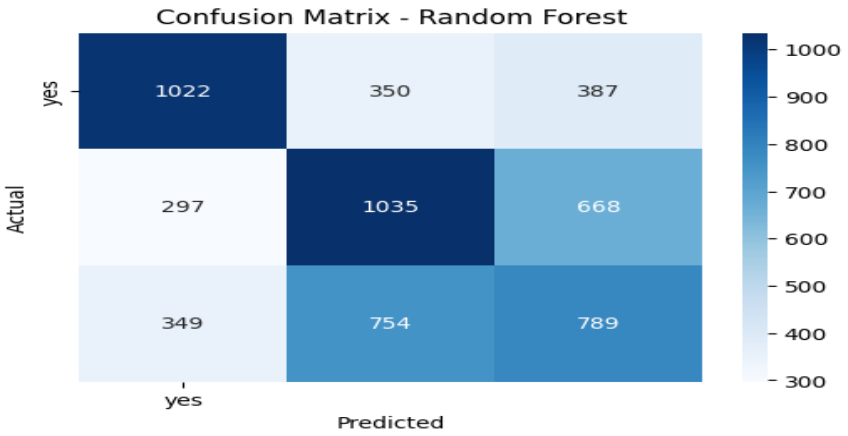

Figure 45: Confusion Matrix - Random Forest

6.5.1.3. Logistic Regression Confusion Matrix:

The confusion matrix for Logistic Regression shows its relatively lower performance compared to the other models:

Diagonal values for `brand` and `female` are lower, indicating weaker classification accuracy.

There is a significant number of misclassifications, especially for the `male` class.

Logistic Regression struggled to classify `male` users, often misclassifying them as `female` or `brand`.

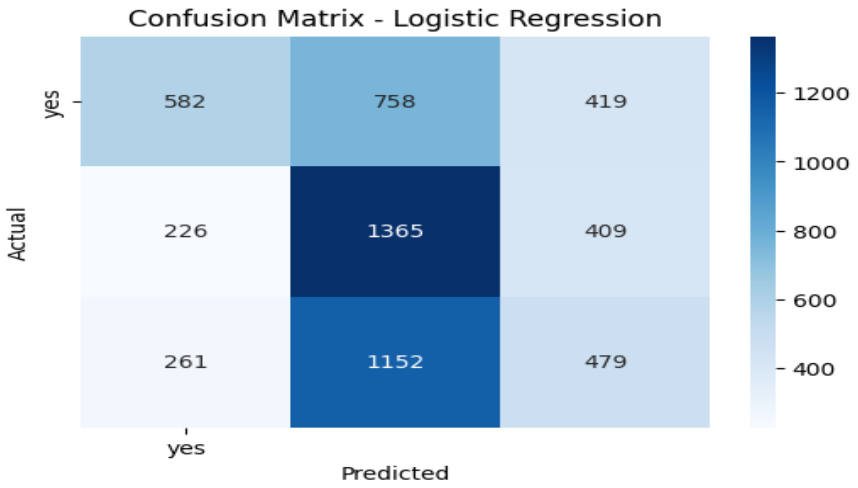

Figure 46: *Confusion Matrix - Logistic Regression*

6.5.2. Plotting Actual vs Predicted Values:

Actual vs predicted plots visualize the relationship between the true labels and the predictions made by each model. Ideally, the points in the plot should align along the diagonal, indicating perfect predictions.

6.5.2.1. Gradient Boosting - Actual vs Predicted:

For Gradient Boosting, the plot showed better alignment along the diagonal for `brand` and `female` but more scatter for `male` predictions:

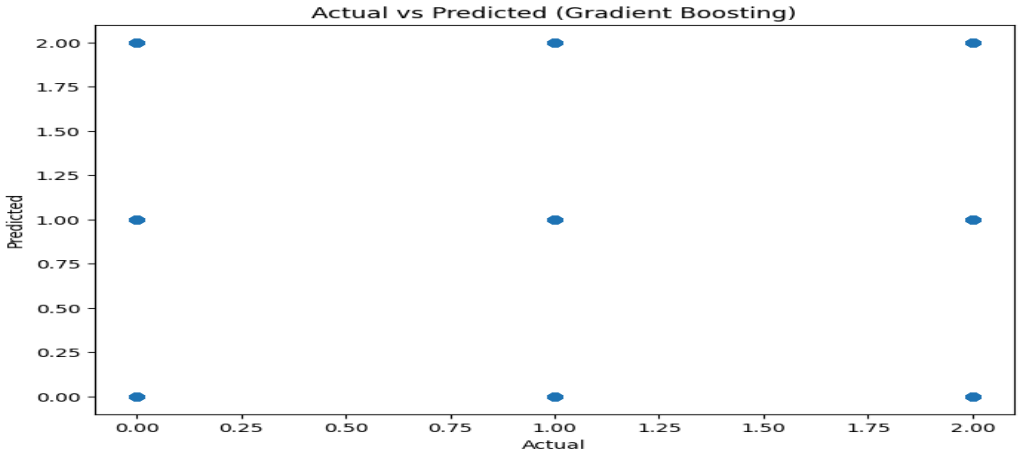

Figure 47: *Actual vs Predicted - Gradient Boosting*

The greater dispersion of `male` predictions illustrates the model's difficulty in classifying this category.

6.5.2.2. Random Forest - Actual vs Predicted:

In the case of Random Forest, the points are more dispersed, with fewer predictions falling on the diagonal:

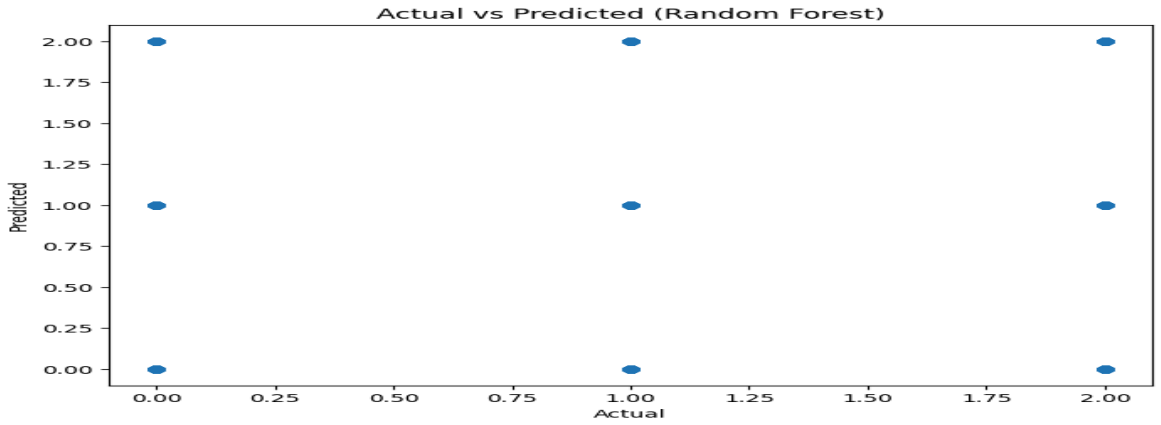

Figure 48: *Actual vs Predicted - Random Forest*

Random Forest did well in predicting `brand` users but had more difficulty distinguishing between `female` and `male` users.

6.5.2.3. Logistic Regression - Actual vs Predicted:

Logistic Regression had the least accurate predictions, with significant scatter and fewer points aligning with the diagonal:

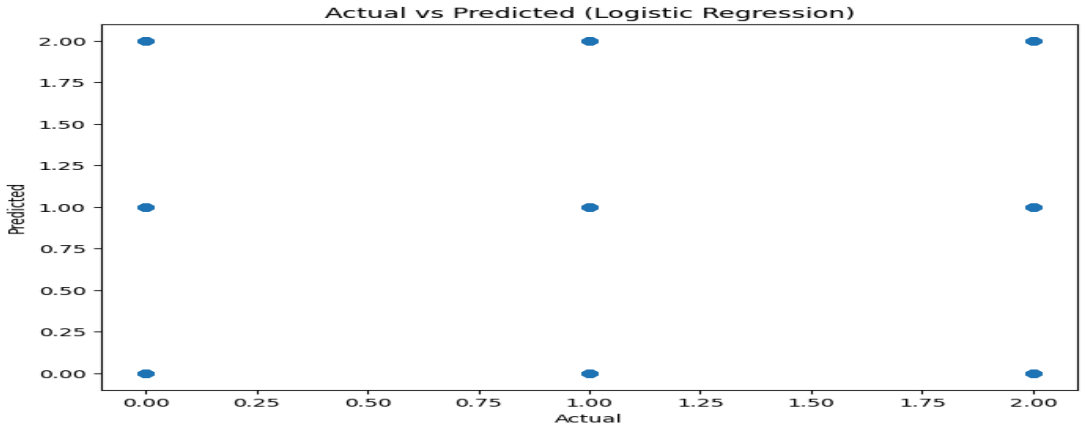

Figure 49: *Actual vs Predicted - Logistic Regression*

The misclassification of `male` users is evident, as many predictions deviate far from the diagonal.

6.5.3. Residuals (Difference Between Actual and Predicted):

Residual plots help visualize the errors made by the model, specifically how much the predicted value deviates from the actual value. Large residuals indicate greater prediction errors.

6.5.3.1. Gradient Boosting - Residuals:

The residuals for Gradient Boosting show a relatively tighter distribution for `brand` and `female`, but wider residuals for `male`, reflecting the challenges in predicting this category:

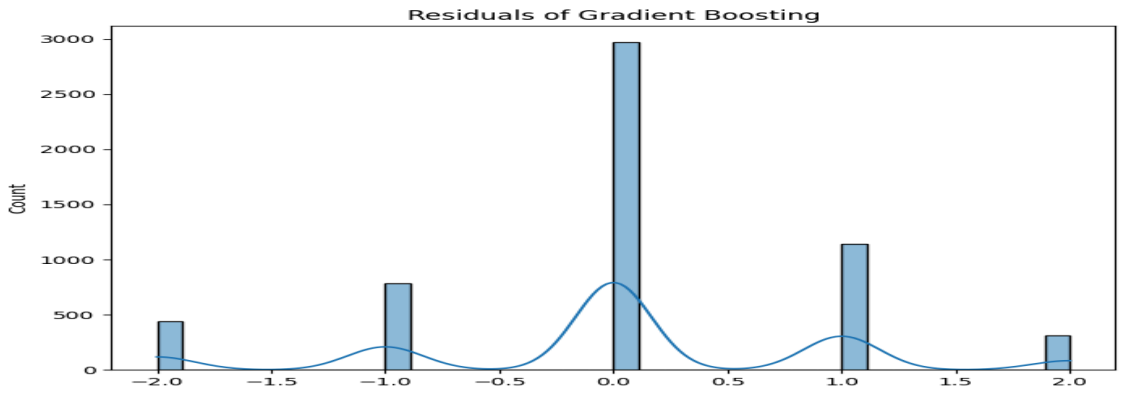

Figure 50: *Residuals of Gradient Boosting*

The residual distribution for `male` predictions is more spread out, indicating less accurate predictions.

6.5.3.2. Random Forest - Residuals:

The residual plot for Random Forest also shows wider residuals, particularly for `male`, though slightly better than Logistic Regression:

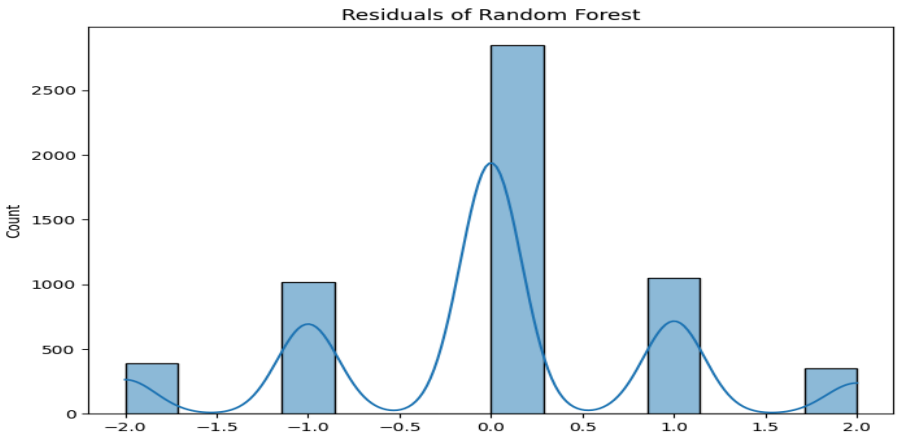

Figure 51: *Residuals of Random Forest*

There is more error variance compared to Gradient Boosting, particularly for `female` and `male` users.

6.5.3.3. Logistic Regression - Residuals:

Logistic Regression has the widest residuals of all three models, especially for `male` and `female` predictions:

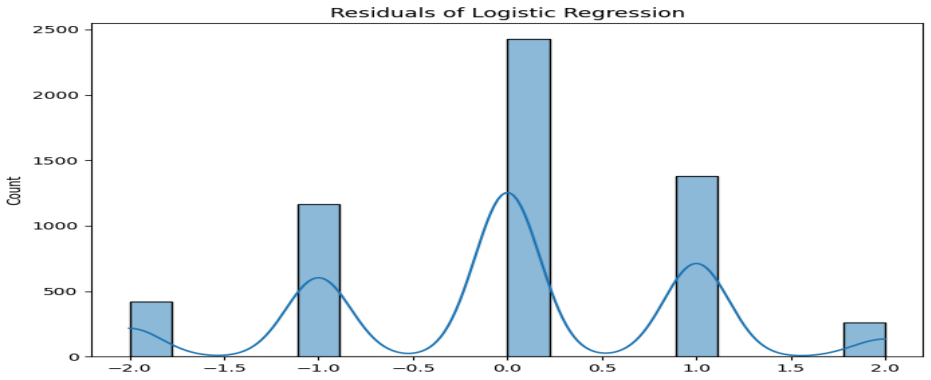

Figure 52: *Residuals of Logistic Regression*

This wider spread in the residuals highlights the lower accuracy of the Logistic Regression model, especially for the `male` class.

Summary of Confusion Matrix, Actual vs Predicted, and Residual Analysis:

Gradient Boosting consistently showed the best performance across all measures, with tighter residuals and better alignment of actual vs predicted values.

Random Forest performed moderately well, with some scattering in the actual vs predicted plots and wider residuals for the `male` category.

Logistic Regression performed the poorest, with the widest residuals and significant misclassifications, especially in the `male` category.

While all models struggled with the `male` category, Gradient Boosting emerged as the best overall model, particularly in terms of precision and recall for the `brand` and `female` categories.

Key Findings:

Gradient Boosting outperformed the other models in both accuracy and precision, though at the cost of increased training time.

Random Forest provided a reasonable balance between performance and interpretability, with a moderate accuracy of 47.8%.

Logistic Regression struggled with this dataset, likely due to its inability to capture non-linear relationships effectively.

Model Comparison:

- *Logistic Regression:* Accuracy: 44.3%, weighted F1-score: 0.36.
- *Random Forest:* Accuracy: 47.8%, weighted F1-score: 0.48.
- *Gradient Boosting:* Accuracy: 50.3%, weighted F1-score: 0.50.

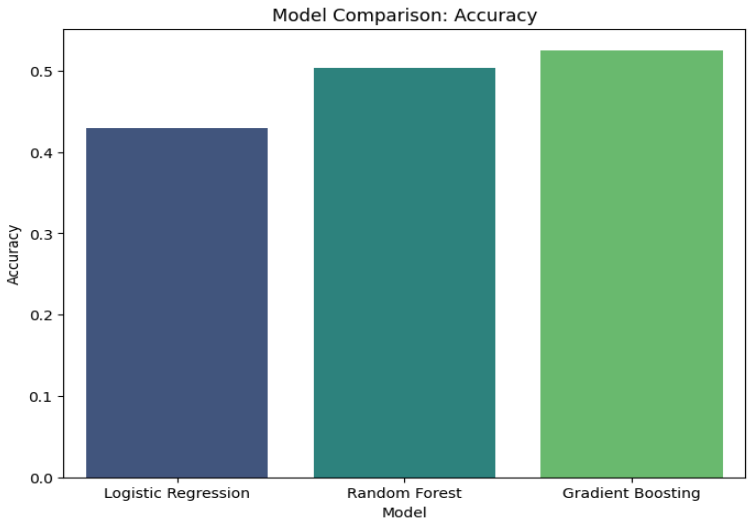

Figure 53: *Model Comparison Accuracy*

Conclusion:

Gradient Boosting was the most effective model for gender classification based on Twitter data, although Random Forest also performed well. Logistic Regression, while easier to interpret, was less suited for this task. Future work could explore hyperparameter tuning and additional features to improve model performance further.

6.5.4. Feature Importance across Classification Models:

Feature importance provides insights into how each attribute influences the predictions made by different classification models. Here's a breakdown of feature importance for each of the three models: Logistic Regression, Random Forest, and Gradient Boosting.

6.5.4.1. Logistic Regression - Feature Importance:

In logistic regression, feature importance is represented by the absolute values of the model's coefficients. Higher coefficient values indicate a stronger influence on the target variable (in this case, gender).

Top influential features:

- *Description Length:* This has the highest impact, showing that the length of the description is a strong predictor of whether the profile is human or non-human.
- *Tweet Count:* The number of tweets also plays a significant role.
- *Tweet-to-Retweet Ratio:* This ratio helps differentiate between content creators (brands) and individual users.

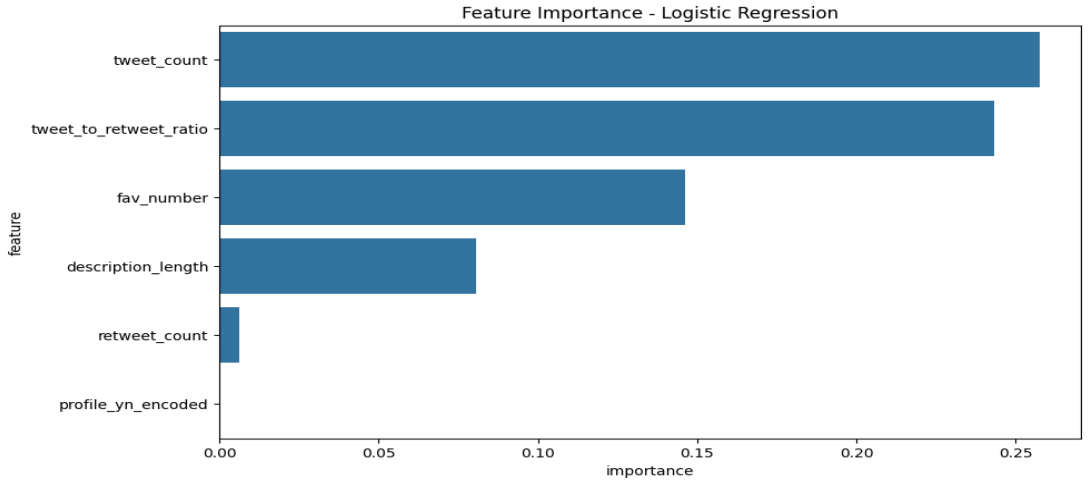

Figure 54: *Feature Importance - Logistic Regression*

6.5.4.2. Random Forest - Feature Importance:

In Random Forest, feature importance is determined by the average decrease in impurity (Gini) when a feature is used to split data in a tree.

Top influential features:

- *Tweet Count:* The most important feature for classifying gender and human vs. non-human profiles. It shows that tweet activity is highly discriminative.
- *Fav Number:* The number of favorites (likes) the user has received is a strong indicator, especially for identifying brands.

- *Description Length:* Continues to be an important predictor, reflecting brand marketing strategies with longer, more detailed descriptions.

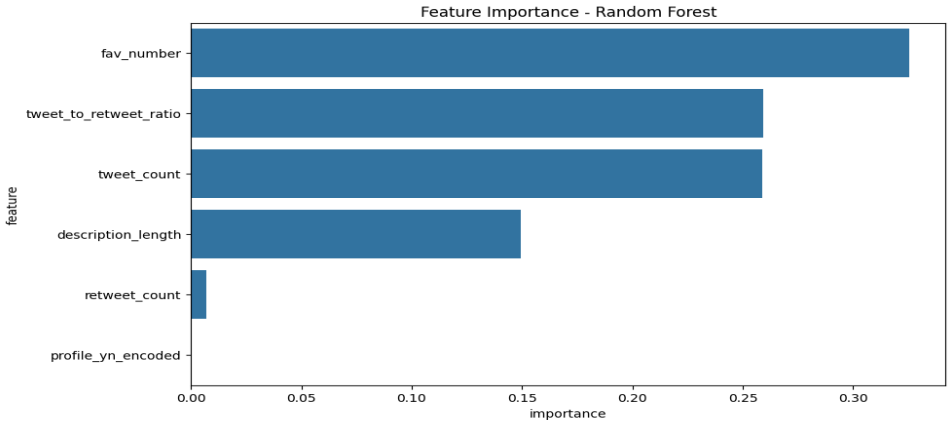

Figure 55: *Feature Importance - Random Forest*

6.5.4.3. Gradient Boosting - Feature Importance:

In Gradient Boosting, feature importance is also based on how much each feature contributes to reducing prediction error.

Top influential features:

- *Fav Number:* Again, a key feature indicates whether the user is a brand or a person.
- *Tweet Count:* As with Random Forest, the number of tweets is a strong predictor of the target variable.
- *Profile Yes/No:* Whether the user has an active profile plays a significant role in predictions.

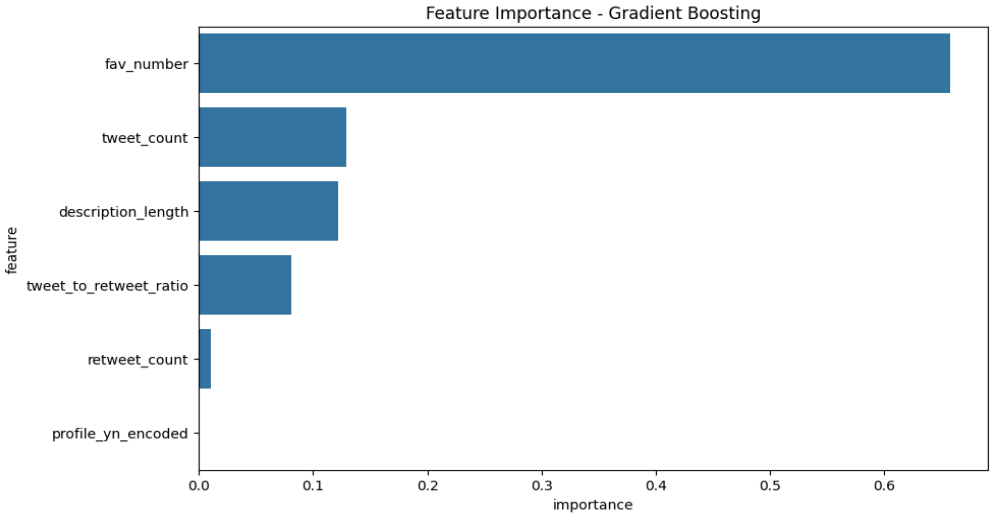

Figure 56: *Feature Importance - Gradient Boosting*

Feature Importance Summary:

- Fav Number and Tweet Count consistently appear as top features across all models. Brands tend to have higher favorite counts and more consistent tweeting activity. Hence, proving that the initial hypothesis stated in Section 1 was correct.
- Description Length is crucial for distinguishing brands from individual users. Brands usually have more detailed descriptions to promote their services. Hence, proving that the initial hypothesis stated in Section 1 was correct.
- Tweet-to-Retweet Ratio shows that brands are more likely to post original content (higher tweet ratio), whereas individuals might retweet more.

7. Clustering 1:

7.1. Algorithm Consideration:

The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm was chosen for this analysis due to its robustness in handling noisy datasets and its ability to detect clusters of varying shapes and sizes. Unlike traditional algorithms such as k-means, DBSCAN does not require the user to predefine the number of clusters, making it ideal for exploratory analysis where the structure of the data is unknown or complex, as with Twitter user data (Ester et al. 1996).

Key Considerations for Choosing DBSCAN:

- *Noise Handling:* Twitter data often contains outliers in the form of bots, highly active users, or accounts with sporadic activity. DBSCAN effectively handles these outliers by labeling them as noise points, which prevents them from skewing the clustering results and ensures that clusters are formed based on genuine user behavior (Ester et al. 1996).
- *Density-Based Clustering:* DBSCAN clusters data points that are densely packed together, while points in low-density regions (such as outliers) are classified as noise. This is particularly useful when dealing with social media data where user behavior can vary significantly, and there are no clear linear boundaries between clusters (Jain et al. 1999).

- *No Need for Predefined Clusters:* DBSCAN's flexibility is one of its key strengths, as it does not require the number of clusters to be specified in advance. This is especially important for exploratory data analysis, such as Twitter data, where predefined clusters may lead to biased or inaccurate results (Pedregosa et al. 2011).

The eps parameter, which controls the maximum distance between two points in a neighborhood, and the min_samplesparameter, which defines the minimum number of points to form a cluster, were fine-tuned to ensure meaningful clusters were identified while minimizing noise.

7.2. Choice of Methods & Justification:

The choice of DBSCAN for clustering Twitter data was driven by several key factors:

- *Suitability for Social Media Data:* Social media datasets, especially Twitter, are known for their noisiness, with a mix of regular users, bots, and inactive accounts. DBSCAN's ability to isolate these outliers by labeling them as noise ensures that the clustering results focus on genuine user behavior, making it highly effective for this type of data (Atefeh and Khreich 2013).

- *Non-Linear Cluster Shapes:* Unlike algorithms like k-means, which assume clusters are spherical, DBSCAN can identify clusters of arbitrary shapes. This is crucial for datasets like Twitter, where user engagement patterns can vary widely, and clusters may not follow simple geometric patterns. DBSCAN's flexibility allows it to detect clusters that reflect the true diversity of user behavior (Jain et al., 1999).

- *Scalability:* Although DBSCAN is not ideal for very large datasets, it performs well on medium-sized datasets like the one used in this analysis (around 20,000 rows). The algorithm efficiently identifies meaningful clusters through parameter tuning, making it an appropriate choice given the manageable size of the dataset (Pedregosa et al., 2011).

Parameter Selection:

- *Epsilon (eps) = 0.5:* This value defines the maximum distance between two points for them to be considered part of the same neighborhood. After testing various values, 0.5 was found to strike a balance between forming clusters and minimizing

noise points. If eps were too small, more points would be classified as noise; if too large, the clusters would lose their meaning.

- *Minimum Samples (min_samples) = 5:* This parameter controls the minimum number of points required to form a cluster. A value of 5 was chosen because it allows smaller clusters to form while preventing excessive noise points. This choice is particularly important for Twitter data, where some clusters, such as highly engaged users, may be smaller but still meaningful.

These parameters were selected to ensure that DBSCAN could form well-defined clusters while identifying noise points that do not belong to any meaningful group, ensuring the algorithm effectively segments user data into distinct groups.

7.3. Data Preprocessing & Visualization:

To prepare the data for DBSCAN, several preprocessing steps were undertaken to ensure that the features were clean, standardized, and suitable for clustering. This process involved handling missing data, creating meaningful features, and standardizing the dataset to ensure fair treatment of all features.

1) **Feature Engineering:** Several new features were created to capture user behavior and improve clustering quality:

   - *tweet_to_retweet_ratio:* This feature reflects the ratio of original tweets to retweets, which provides insights into user activity. Users who primarily retweet tend to behave differently from those who generate original content, which helps differentiate between human and non-human profiles.

   - *description_length:* This feature measures the length of a user's bio or profile description. Brands and organizations often have longer descriptions compared to individuals, so including this feature helps differentiate between human users and brands, adding another layer of insight to the clustering process.

2) **Dropped Features:** Several features were excluded from the clustering process based on their relevance, quality, and usefulness:

   - *profileimage, tweet_coord, user_timezone:* These features were dropped because they contained significant amounts of missing data or did not provide meaningful information for clustering. For example, the profileimagefeature is a

link to the user's profile image, which does not contribute to understanding user behavior. Similarly, tweet_coord and user_timezone were often missing or filled inconsistently, making them irrelevant for clustering purposes.

- *unit_id, tweet_id:* These columns served as unique identifiers and were not useful for clustering based on behavior. Including them would not add any additional insights.

- *link_color, sidebar_color:* These aesthetic features, such as the colors chosen for a user's profile, were not relevant for understanding user engagement and behavior. They could potentially add noise to the model, so they were excluded.

Explanation: By focusing on features that contribute directly to understanding user behavior, the clustering results become more accurate and insightful. Dropping unnecessary columns helps reduce noise in the model, allowing DBSCAN to perform more effectively.

3) **Handling Missing Values:**

- *Numerical Columns (e.g., `fav_number`, `tweet_count`):* Missing values in numerical columns were filled using the median. This prevents extreme values from skewing the data while ensuring consistency in the dataset.

- *Categorical Columns (e.g., `gender`, `profile_yn`):* Missing values in categorical columns were filled using the mode. This ensures that the most common category fills in the gaps while minimizing the introduction of noise.

This approach ensured that the dataset was complete, consistent, and ready for clustering without altering underlying patterns.

4) **Standardization:** All numerical features were standardized using StandardScaler. Standardization was essential to ensure that all features contributed equally to the clustering process. Without standardization, features with larger ranges, like `tweet_count`, could dominate the model and skew the results (Pedregosa et al., 2011). Standardizing the data ensures that all features are treated on the same scale, making the clustering process more accurate.

5) **Principal Component Analysis (PCA):** To reduce the dimensionality of the dataset and enable visualization, PCA was applied. By reducing the data to two or three

principal components, PCA allowed for the visualization of clusters formed by DBSCAN while retaining most of the variance in the data (Zaki and Meira Jr., 2013).

The PCA plot below visualizes the clusters formed by DBSCAN. Each point represents a Twitter user, and the different colors indicate the various clusters discovered by the algorithm. Points labeled as noise (outliers) are represented by “Cluster -1” in the plot:

- **Cluster Formation**: The plot shows several well-formed clusters, indicating that DBSCAN successfully grouped users with similar behaviors.
- **Noise Points**: “Cluster -1” contains a substantial number of points, indicating that many users did not fit neatly into any cluster. These users likely exhibit atypical behavior, such as bots or sporadic activity.

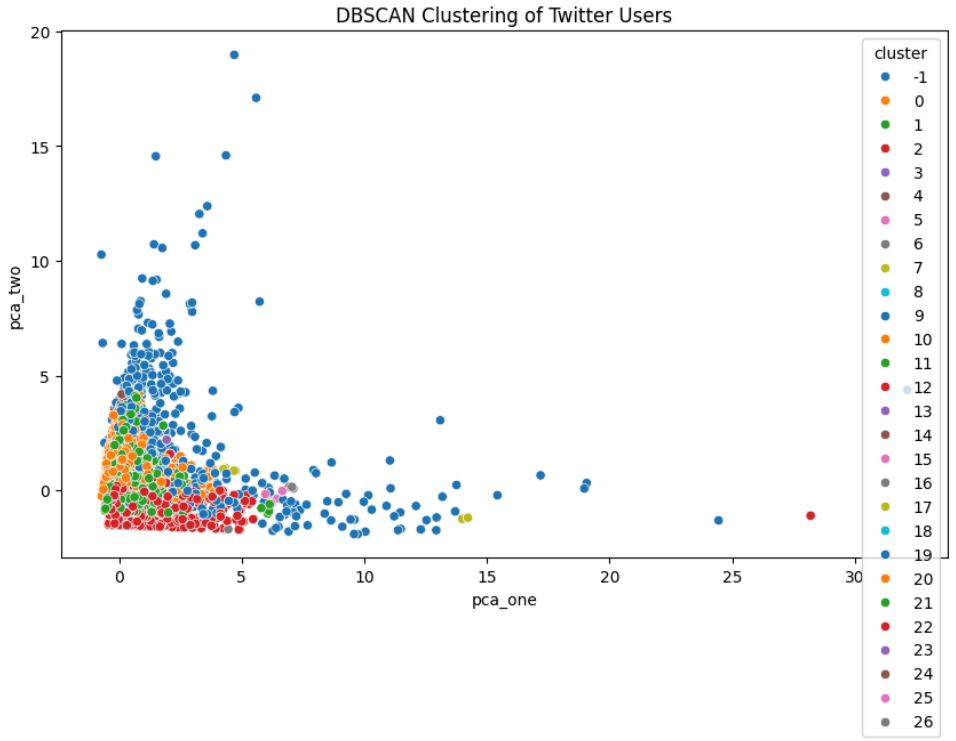

Figure 57: DBSCAN Clustering of Twitter Users

7.4. Model Application:

7.4.1. DBSCAN Clustering:

DBSCAN was applied using the parameters Epsilon (eps) = 0.5 and Minimum Samples (min_samples) = 5. These parameters were selected after evaluating the dataset’s characteristics, balancing the number of clusters and minimizing noise points.

Results:

27 clusters were identified, along with 555 noise points. The noise points represent outliers that do not belong to any of the identified clusters. These outliers could include users with abnormal behavior, such as bots or sporadic users that do not conform to any meaningful behavioral pattern (Ester et al., 1996).

Key Features of the Clusters:

A summary of key characteristics for each cluster is provided in the table below, highlighting average values for Fav Number, Tweet Count, Retweet Count, Description Length, Gender, and Profile for each cluster, with Cluster -1 representing noise.

Cluster	Fav Number	Tweet Count	Retweet Count	Description Length	Gender	Profile
-1	41018.08	243575.20	1.89	14.14	1.01	0.97
0	3538.00	23383.65	0.03	11.64	0.00	1.00
1	4591.21	23096.44	0.02	9.70	1.00	1.00
—	—	—	—	—	—	—

Fav Number: In Cluster -1, the average Fav Number is 41,018.08, indicating highly engaged users, likely bots or viral accounts, which garnered unusually high levels of favorites. In contrast, Cluster 0 has a much lower average of 3,538.00, reflecting regular user engagement. The extreme value in Cluster -1 suggests outlier behavior, such as automated accounts, whereas the lower value in Cluster 0 represents a typical range for user favorites.

Tweet Count: The average Tweet Count in Cluster -1 is 243,575.20, far exceeding typical user behavior, strongly suggesting that this group consists of bots or automated users tweeting excessively. Meanwhile, Cluster 1 shows a more moderate Tweet Count of 23,096.44, reflecting active but more natural user engagement. The vast difference in tweet counts highlights the presence of abnormal accounts in Cluster -1 compared to regular users in Cluster 1.

Retweet Count: Both Cluster 0 and Cluster 1 have nearly negligible Retweet Counts of 0.03 and 0.02, respectively, indicating that users in these clusters primarily post original content. Cluster -1, with a slightly higher Retweet Count of 1.89, still shows low retweet

activity, suggesting that even outliers are more focused on generating content rather than retweeting, which is characteristic of bots or automated accounts.

Description Length: In Cluster -1, the average Description Length is 14.14, indicating slightly longer profile bios, typical of bots, brands, or organizations that often provide more detailed descriptions. Cluster 0 users have a shorter average Description Length of 11.64, which is more common among regular users who tend to keep their bios concise. The difference in description length helps differentiate automated or brand accounts from individual users.

Gender: Cluster 1 has an average Gender value of 1.00, suggesting that this group is dominated by a specific gender, possibly reflecting certain engagement or activity trends tied to gender. In contrast, Cluster 0 has a balanced Gender value of 0.00, indicating that gender was not a significant factor in this group, with both genders showing similar engagement patterns.

Profile: The Profile completeness in Cluster -1 averages at 0.97, implying that most users in this cluster have incomplete profiles, aligning with the idea that this cluster contains bots or inactive accounts. Cluster 0, with a Profile completeness of 1.00, indicates that most users in this cluster maintain fully completed profiles, typical of regular users who engage actively on the platform.

Explanation of Columns Used:

The clustering model, particularly DBSCAN, was built using the following key columns:

1. **Fav Number**: Represents the number of favorites a user has, which helps indicate user engagement and popularity.
2. **Tweet Count**: Reflects the activity level of the user, i.e., how frequently they tweet.
3. **Retweet Count**: Helps differentiate between users who generate original content and those who primarily retweet.
4. **Description Length**: A measure of the length of a user's bio or profile description, which can distinguish between individuals and brands or bots.
5. **Gender**: Encoded as numerical values, this helps understand if certain clusters are more dominated by specific genders.
6. **Profile**: A binary feature representing the presence or absence of a complete profile.

Reasons for Dropping Other Columns:

1. **profileimage**: This column was dropped because it represents a link to the user's profile image and does not contribute to the clustering process based on user behavior or engagement.
2. **tweet_coord**: This column records the coordinates if a user has location turned on. Since this data was highly inconsistent and incomplete, it did not add meaningful value to the model.
3. **user_timezone**: This column was also dropped due to its incomplete and often inaccurate data. It was not relevant for clustering users based on behavior or engagement.
4. **unit_id and tweet_id**: These are unique identifiers for users and tweets, respectively. Since clustering relies on grouping similar behaviors, these identifiers do not provide any insights into user behavior and were therefore excluded.
5. **link_color and sidebar_color**: These aesthetic features represent user-chosen colors for their profile. While relevant for visual presentation, they are not related to user engagement, behavior, or activity, which are the primary focuses of the clustering process.

This figure below shows the distribution of clusters across the dataset. The majority of data points are concentrated in "Clusters 0, 1, and 2", which represent typical user behavior such as regular posting and engagement on the platform. A significant number of points were labeled as noise (-1), likely representing bots or users with inconsistent activity.

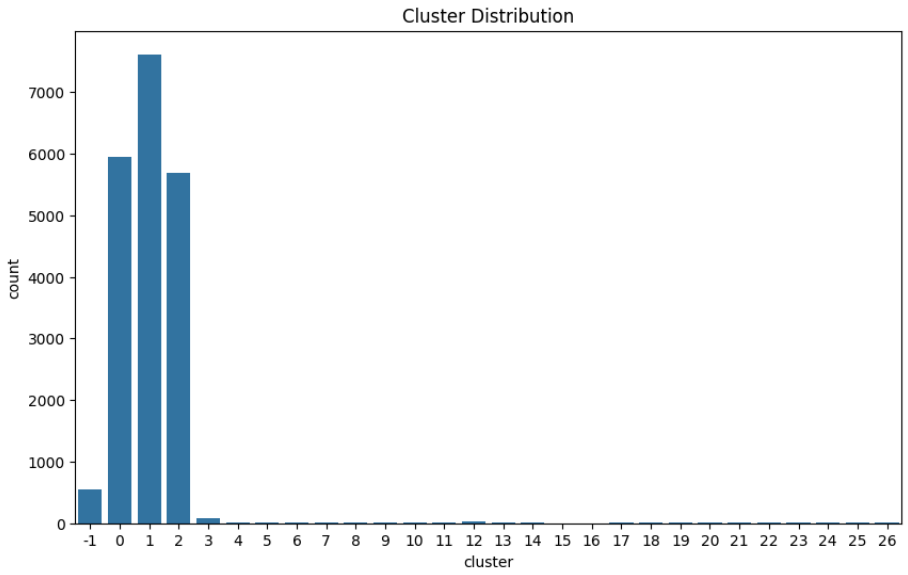

Figure 58: Cluster Distribution

7.4.2. PCA-Based Visualization:

PCA (Principal Component Analysis) was employed to visually represent how DBSCAN separated the dataset into clusters. By reducing the dataset to two or three principal components, PCA provides an easy-to-interpret visualization of the clusters. This method allows for better interpretation of the relationships between data points and their membership in different clusters, especially when dealing with high-dimensional data (Ester et al. 1996).

The plot in Figure 59 shows the distribution of clusters by gender. In clusters such as 0, 1, and 2, there is a mix of male, female, and brand accounts. This indicates that the DBSCAN algorithm effectively grouped users regardless of gender. However, some smaller clusters show more homogeneous groups, implying that certain behaviors may correlate with specific genders or user types:

- **Diversity in Clusters:** The mix of male, female, and brand accounts in the main clusters (0, 1, and 2) demonstrates that gender is not the primary driver of clustering. Instead, user behavior and engagement patterns play a more significant role in determining cluster membership.
- **Homogeneous Clusters:** Smaller clusters consisting of more homogeneous users suggest that gender may have a more pronounced influence on certain behaviors, though it is not the dominant factor.

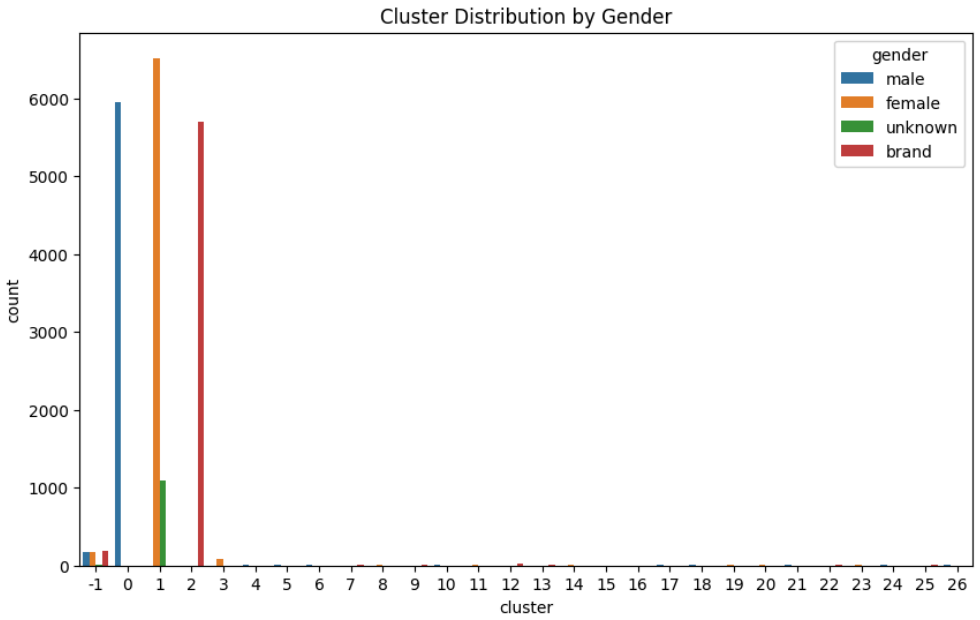

Figure 59: Cluster Distribution by Gender

The PCA plot in Figure 60 visualizes clusters, with colors representing different gender groups (male, female, brand). This helps highlight any correlations between gender and cluster membership. Although gender is not the primary factor for cluster formation, this visualization shows that there are some clusters where certain gender groups dominate, particularly in the smaller, more specific behavioral clusters:

- Gender Influence: While the overall analysis shows that gender does not have a significant influence on cluster membership, the plot reveals some areas where certain gender groups dominate. This suggests that some behaviors may be more prevalent among specific genders, but these patterns are not strong enough to drive the clustering process.

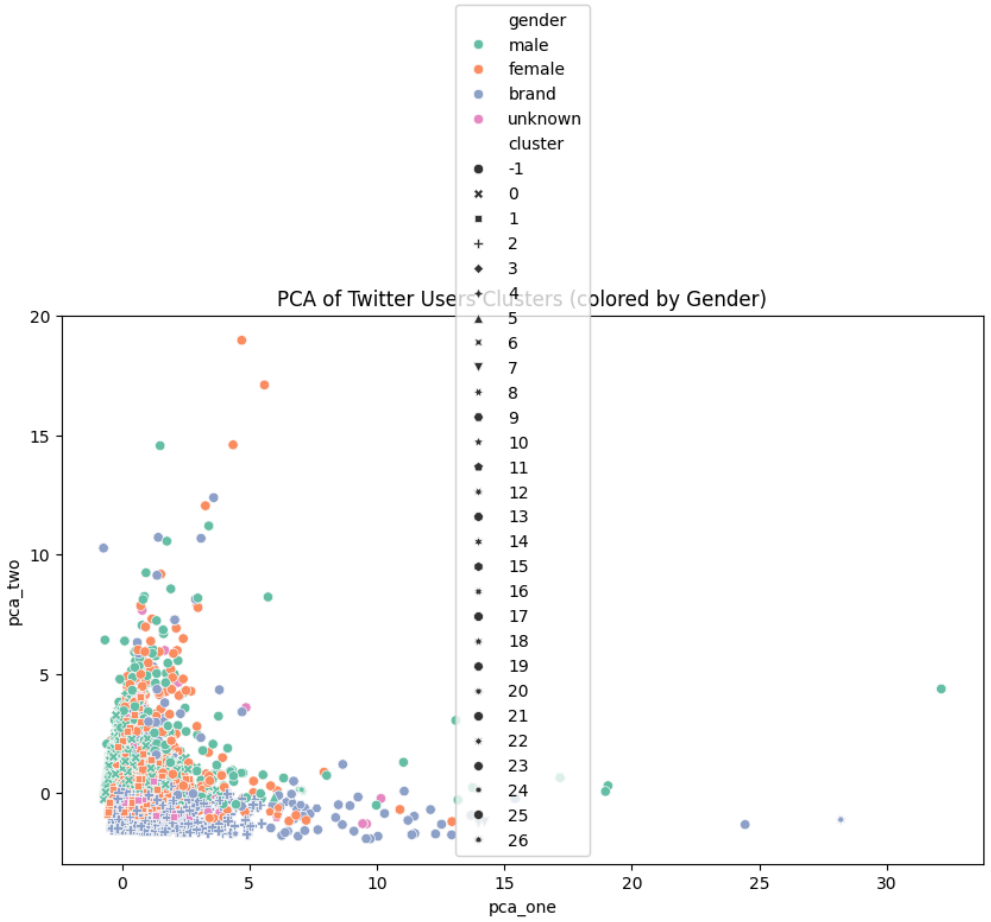

Figure 60: *PCA of Twitter Users Clusters (colored by Gender)*

7.4.3. Silhouette Score Evaluation:

The Silhouette Score was used to measure the quality of clustering by evaluating how well-separated the points are between clusters. A higher silhouette score indicates better-defined clusters, while a lower score suggests overlap between clusters and less clear boundaries (Atefeh and Khreich 2013).

Silhouette Score Results:

- The overall **silhouette score** of **0.223** for DBSCAN indicates that the clustering was not optimal, meaning that human and non-human profiles were not clearly separated. This low score suggests significant overlap between clusters, which could imply that certain behaviors, such as tweet activity, profile characteristics, and engagement levels, do not distinctly separate human and non-human profiles. For example, bots (non-human profiles) may mimic human behavior, leading to ambiguous classifications where both types fall near the boundaries between clusters.

- The **silhouette plot** further highlights this issue, as points with lower silhouette scores likely represent users whose behavior is not clearly human or non-human. These points lie at the boundaries between clusters, indicating that factors such as tweet frequency or profile completeness may not be enough to fully distinguish between the two. This overlap suggests that while DBSCAN can identify general trends, such as very high tweet counts for bots, some non-human profiles exhibit behavior that resembles human activity, making classification more difficult.

Improving Human and Non-Human Profile Classification:

- To address this, further parameter tuning and feature engineering could improve classification. Incorporating additional features beyond basic engagement metrics, such as **tweet content analysis** (e.g., language patterns), **use of specific hashtags**, or **profile aesthetics** (e.g., color schemes), could help better differentiate between human and non-human users. These additional factors could reduce the overlap between clusters by capturing more complex behaviors that are unique to non-human profiles, such as automated tweet patterns or longer, more formal descriptions typical of bots or brands.

- Incorporating **multiple views/modes**—including text analysis, visual elements like color use, and tweet timing—could refine the clustering process. For example, bots may use repetitive language or post at unusually consistent intervals, and such factors could help reduce ambiguity in the classification, making the clusters more distinct.

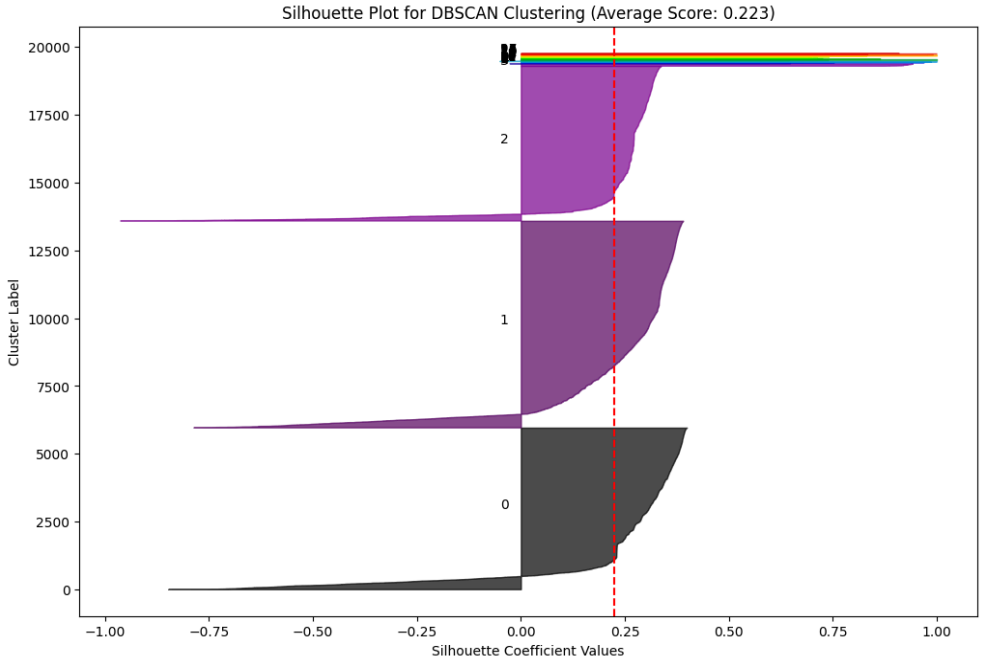

Figure 61: *Silhouette Plot for DBSCAN Clustering*

7.5. Results & Factors Considered:

The results from the DBSCAN clustering algorithm provide valuable insights into the behavioral patterns of Twitter users and the structure of the dataset. Several key findings and considerations emerged from the analysis:

- *Number of Clusters:* The DBSCAN algorithm successfully identified 27 clusters in the dataset, along with 555 noise points. The noise points represent outliers, likely users with atypical behavior, such as bots or inactive users, that do not fit into any meaningful cluster (Ester et al. 1996).

- *Cluster Characteristics:* Cluster -1 (noise) had significantly higher tweet counts and favorites, suggesting that these outliers may represent bots or highly active users whose behavior deviates from the norm. These users' abnormally high activity levels caused them to be classified as noise by DBSCAN (Ester et al. 1996).

Non-Human vs. Human Profiles:

- *Human Profiles:* In the main clusters (e.g., Clusters 0, 1, and 2), human profiles displayed typical user behavior, including moderate tweet activity and description length. These clusters predominantly consist of users with more natural engagement patterns, such as regular posting and retweeting.

- *Non-Human Profiles (Brands):* Clusters containing non-human profiles, such as brands, tended to have longer profile descriptions and a higher tweet_to_retweet_ratio. This suggests that brands tend to post more original content and have more detailed profile descriptions, setting them apart from individual users (Atefeh and Khreich, 2013).

Suggestions:

Based on the analysis, distinguishing between human and non-human profiles can be improved by focusing on behavioral differences observed in the clustering results. Human profiles tend to have moderate tweet activity, shorter bios, and diverse engagement, whereas non-human profiles, such as bots or brands, exhibit more extreme behaviors like higher tweet counts, longer descriptions, and potentially automated posting patterns. To refine the classification, additional features such as tweet timing, use of specific hashtags, and language patterns should be considered. These factors could better differentiate between human and non-human profiles, reducing overlap in the clustering and ensuring clearer classification (Atefeh and Khreich 2013).

Figure 62 illustrates the residuals, showing the difference between gender-encoded values and cluster labels. The residuals plot highlights that gender was not a significant factor in cluster formation. While the DBSCAN algorithm effectively separated users based on their engagement and profile characteristics, the analysis shows that gender did not strongly correlate with the clusters:

- *No Strong Correlation:* The residuals plot highlights that many users' gender identities do not correlate strongly with their cluster assignments. This suggests that behavior and engagement patterns, rather than gender, are the primary factors driving cluster formation (Atefeh and Khreich 2013).

- *Behavior-Driven Clusters:* The clusters formed by DBSCAN were more influenced by user activity and profile attributes than by demographic information such as gender.

These findings support the initial hypothesis stated in Section 1 as the clusters are generated based on user activity and engagement levels.

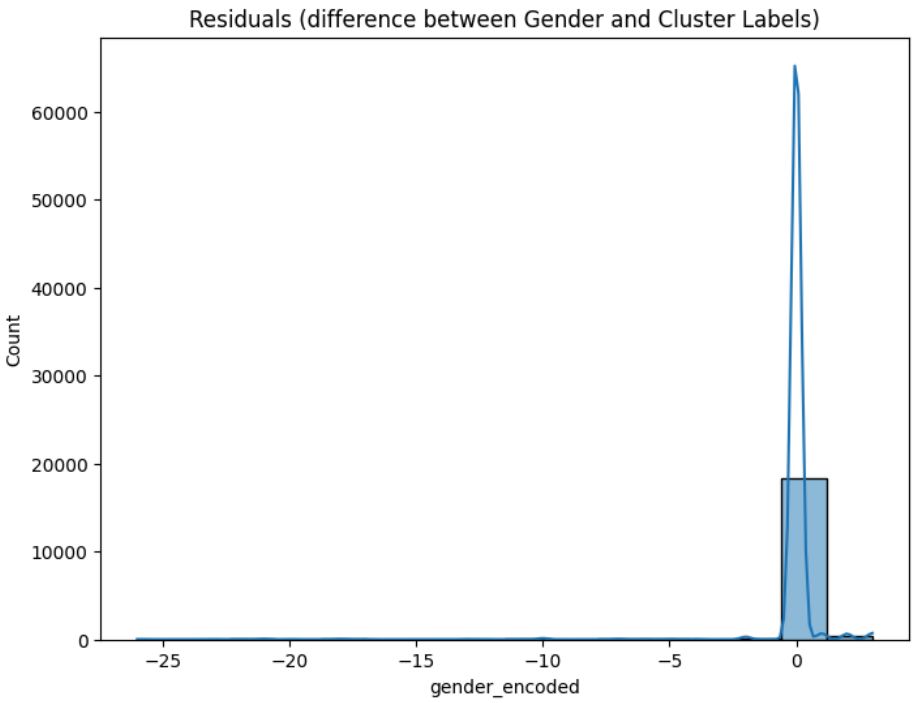

Figure 62: *Residuals (difference between Gender and Cluster Labels)*

Conclusion:

The application of DBSCAN to Twitter user data successfully identified clusters of user behaviors and isolated outliers. However, the quality of clustering was suboptimal, as indicated by the low silhouette score. Many data points were labeled as noise, and there was significant overlap between clusters, suggesting that DBSCAN, although effective at identifying outliers, may not be the best-suited algorithm for this dataset without further parameter optimization (Ester et al. 1996).

Key Factors Contributing to Clustering Performance:

- *High Number of Noise Points:* A substantial portion of the dataset was labeled as noise, indicating distinct or abnormal behavioral patterns for those users. These users, including bots and highly irregular users, did not fit into any meaningful cluster.

- *Gender and Profile Type:* The analysis showed that gender did not have a strong correlation with cluster membership. Instead, the clustering was driven by user activity and profile characteristics, which played a more significant role in determining cluster membership.

- *Cluster Overlap:* The low silhouette score indicates that many data points lie on the boundaries between clusters, leading to overlap and reducing the clarity of the clustering results.

Suggestions for Further Improvement:

1. *Experiment with Other Algorithms:* It may be beneficial to explore other clustering algorithms, such as k-means or hierarchical clustering, which may provide better-defined clusters depending on the dataset's structure. Optimizing DBSCAN parameters or considering alternative algorithms could yield better results for Twitter user behavior (Jain et al., 1999).

2. *Parameter Tuning:* Further tuning of DBSCAN's eps and min_samples parameters may lead to improvements in clustering quality. By adjusting these parameters, the number of noise points could be reduced, and the definition of clusters could be improved, leading to better-separated clusters and clearer insights (Han et al., 2012).

8. Clustering 2:

8.1. Algorithm Consideration:

K-means clustering is a popular unsupervised machine learning algorithm where we can partition a dataset into a specified number of groups based on similar patterns or structures. Its objective is to explore the data and group similar data points in non-overlapping clusters, which helps to visualise and analyse distinct patterns of similar data points and their characteristics (PulkitS 2024). The goal of KMeans is to partition a dataset into K distinct (a defined number) non-overlapping clusters, where each data point with similar values belongs to the cluster (Kavlakoglu & Winland 2024).

Description of the Twitter User Dataset

The Twitter User Dataset contains various information about 20,050 Twitter users, which includes user gender, a profile description, a random tweet, tweet count, retweet count, favourite number, username, description, text, etc. It has a collection of numerical, textual, and categorical data (Twitter user gender classification 2016).

Objective

In the context of this project, the goal is to analyse the dataset of Twitter users by applying the K-Means clustering algorithm, where we will group users who share a similar pattern of activities such as the number of tweets posted (tweet_count), the number of retweets (retweet_count), the number of tweets favorited by the user (fav_number), gender (male, female or brand), description and text. By doing so, we expect to form multiple clusters where users within each cluster share similar behaviours and characteristics which will help us in determining distinct features or patterns of human (male or female) profiles and brand profiles. Using those patterns, we can get valuable insights and an understanding of data to identify metrics to determine whether a human profile is mistakenly labelled as a brand or a brand profile is mistakenly identified as human.

8.2. Choice of Methods & Justification:

We decided to use the K-means clustering algorithm for this analysis due to its good efficiency in handling large numbers of data. K-Means would be a great fit because the dataset contains numerical data for key components of the dataset, including features like tweet count, retweet count, and favorited tweets.

K-Means was selected for the following reasons:

- *Scalability:* The dataset contains data from a large number of users (20,050 rows), and K-Means is well known for its scalability and ability to handle large datasets efficiently without using too many computational resources which makes it a fast and scalable option for big data projects (Jain 2023). That is why it is ideal for large amounts of data where manual grouping would be very difficult.
- *Unsupervised learning:* Since K-means is an unsupervised learning algorithm, it doesn't require labelled data to work. It automatically finds patterns and structures in the data, making it a great tool for exploring data and discovering hidden insights (Jain 2023).
- *Interpretability:* K-Means provides clear, interpretable results by assigning each data point to a specific cluster where each cluster group contains similar data points, making it simple to use. We can derive meaningful insights from the results (Jain 2023).
- *Optimal Cluster Selection:* Instead of guessing and multiple trial and error, we can use the Elbow Method which allows us to determine the optimal number of clusters

by plotting the sum of squared distances between users and their cluster centroids, helping ensure that the chosen clusters are meaningful and well-separated for the dataset (Dawari 2023).

Because of the easy-to-use functionality, scalability, and better interpretability K-Means is an ideal model for clustering for this dataset.

8.3. Data Preprocessing & Visualization:

Before we do model planning and implementation and start building the model we need to do some data preprocessing steps to ensure our data is suitable for clustering. It involves cleaning, transforming, and handling raw data to make it suitable for analysis. Without proper preprocessing, models can give inaccurate or misleading results (Ananda 2021).

Below are the key preprocessing steps that are implemented:

- *Loading Dataset:* In real-world datasets, it is common to have missing values, and contain irrelevant features or noisy data. For example, some fields may contain null values or certain data may be unnecessary. To observe that first, we need to load the dataset and visualise the data.

  The dataset was loaded from a CSV (comma-separated values) file containing Twitter user data named 'twitter_user_data.csv'. Below is a code snipped to read the dataset given-

  ```
  # Read from local folder
  dataframe = pd.read_csv('twitter_user_data.csv',encoding='latin1')
  ```

  Before we do any analysis, the data must be read from its source, in this case, a CSV file.

- *Dropped Features*: We need to focus on the most relevant columns and drop unnecessary columns that have little or no impact. Here is a list of columns that were dropped -

  unit_id, trusted_judgments, last_judgment_at: These fields relate to the unique identifier for each user, data tracking and judgement. They are not directly relevant to the user behaviour of the user.

unit_state, golden, profile_yn_gold, gender_gold: These columns indicate whether the data point was part of a "gold standard" of the model. Since the focus is on clustering Twitter user behaviour and patterns, these are not relevant.

created, tweet_created, tweet_id, profileimage, link_color, sidebar_color, tweet_coord, tweet_location, user_timezone: These columns include specific details such as account creation time, tweet creation time, profile image, the colour of sidebar and link in HEX values, tweet location of a user, user's timezone etc. They don't directly contribute to behavioural clustering and were dropped to simplify the dataset and avoid noise.

Aesthetic features (link_color, sidebar_color), unique identifiers (_unit_id, tweet_id), and metadata related to data collection (tweet_created, _golden, _unit_state, _trusted_judgments, _last_judgment_at) do not contribute to understanding user characteristics essential for clustering.

- *Features to keep:* Finally, here is the list of the most relevant columns that we have considered for the model based on their significance:
    - ➢ **tweet_count (Number of Tweets):** This feature indicates how active a user is on Twitter based on the total number of tweets posted from the time of account creation. Users with a high number of tweets are generally more engaged with the platform, while those with fewer tweets may be occasional users. This feature is important because it helps differentiate between highly active users and less active ones, which is a major factor in user behaviour analysis.
    - ➢ **retweet_count (Number of Retweets):** This measures how often a user retweeted, which is also another metric to determining user activity level.
    - ➢ **fav_number (Number of Favourites):** The number of favourites (or "likes") a user's tweets receive is another metric that shows the engagement and popularity of the user's content. It can differentiate between users who receive more likes and engage versus those who receive little engagement.
    - ➢ **description (User's Profile Description):** The description is a text field where users can share information about themselves. It gives insights into the user's personality, and interests on the platform. For example, a brand would most likely represent itself to the audience through a description

whereas a person would describe themselves, their interests, etc. It contains missing values, unwanted characters, symbols, etc. which will be cleaned in the following preprocessing steps.

- **profile_yn (Profile Active Status):** This field indicates whether the profile is active or not. Profiles marked as "no" could be inactive, or not available making them less useful for clustering but it ensures that only active and meaningful profiles are included in the analysis.
- **text (A Random Tweet by the User):** This provides a sample of the user's recent activity on Twitter. The content of a user's tweet can reveal their interests, tone, and style of communication.
- **gender (User's Gender):** Knowing whether a user identifies as male, female, or a brand helps in understanding how different types of users engage with Twitter. It is useful for segmenting users into human categories (male, female) versus non-human categories (brands). Brands may behave differently on Twitter compared to individuals, so this helps in distinguishing between them.
- **gender:confidence (Confidence in Gender Label):** This feature provides a confidence score for the gender label, indicating how accurate the gender prediction is.

Based on the analysis these columns from the dataset were chosen for further analysis, specifically the ones most relevant for clustering:

```
# Keeping the most important columns for clustering as we need numeric values for KMeans Clustering
relevant_columns = ['tweet_count', 'retweet_count', 'fav_number', 'description', 'profile_yn',
                    'text', 'gender', 'gender:confidence']
dataset = dataframe[relevant_columns]
```

Keeping only relevant data ensures that the algorithm works efficiently with the most important features, making the clustering more meaningful.

- *Handling Missing Values:* In real-world data, it is common to have missing or invalid entries. The following steps were taken to observe columns with missing values and appropriate measures to handle such issues:

```
#Check Empty/Null values in the dataset.
dataset.isna().sum()
```

This figure outlines the columns with missing values

	0
tweet_count	0
retweet_count	0
fav_number	0
description	3744
profile_yn	0
text	0
gender	97
gender:confidence	26

dtype: int64

Figure 63: *Columns with missing values*

a) **Removing Profiles with No Activity:** Rows, where the user profile is inactive (profile_yn = 'no'), were removed because inactive profiles do not provide meaningful data for clustering.

```
# Delete rows where profile_yn is 'no' because the gender is null for
those too
dataset = dataset[dataset['profile_yn'] != 'no']
```

b) **Removing Rows with Unknown or Missing Gender:** Any rows where the gender was either labelled as unknown or was missing value (NULL) were removed.

```
# Drop rows where gender is 'unknown' or null (NaN)
dataset = dataset[(dataset['gender'] != 'unknown') &
(dataset['gender'].notnull())]
```

c) **Handling missing values in the description:** A large portion of the description column missing values, a total of 3744 is empty. Instead of

dropping all these rows and their respective values, we filled them with empty strings instead of leaving them as null. This ensures that the dataset is complete and there are no missing values in important columns like description. The figure below shows that the dataset contains no missing values after successfully handling missing values

	0
tweet_count	0
retweet_count	0
fav_number	0
description	0
profile_yn	0
text	0
gender	0
gender:confidence	0

dtype: int64

Figure 64: *Columns with no missing values*

- *Text cleaning:* The dataset with important features contained two text columns, description (the user's profile description) and text (a sample tweet from the user). These texts needed to be cleaned to remove unnecessary characters and ensure consistency. To do so, we first convert all text to lowercase, removing noise like URLs and unwanted characters like #, @, / etc. We also remove non-ASCII characters and punctuation. Finally removed unnecessary extra spaces to ensure text uniformity. Here is an example of text before and after cleaning.

	description	text
0	i sing my own rhythm.	Robbie E Responds To Critics After Win Against...
1	I'm the author of novels filled with family dr...	□ÛÏIt felt like they were my friends and I was...
2	louis whining and squealing and all	i absolutely adore when louis starts the songs...
3	Mobile guy. 49ers, Shazam, Google, Kleiner Pe...	Hi @JordanSpieth - Looking at the url - do you...
4	Ricky Wilson The Best FRONTMAN/Kaiser Chiefs T...	Watching Neighbours on Sky+ catching up with t...

Figure 65: *text before cleaning*

	description	text
0	i sing my own rhythm	robbie e responds to critics after win against...
1	im the author of novels filled with family dra...	it felt like they were my friends and i was li...
2	louis whining and squealing and all	i absolutely adore when louis starts the songs...
3	mobile guy 49ers shazam google kleiner perkins...	hi jordanspieth looking at the url do you use ...
4	ricky wilson the best frontmankaiser chiefs th...	watching neighbours on sky catching up with th...

Figure 66: *text after cleaning*

Cleaning the text ensures better analysis by removing irrelevant details.

- *Creating New Features:* In handling the missing values phase instead of dropping the data of the description column with missing values, missing values in the 'description' column were handled by filling them with empty strings. The reason behind this is that we want to develop a new feature called ‘description_length’ which contains the length of the description text after cleaning it. By doing this, we are keeping those profiles that have no description and keeping that record on the “description_length” column as 0. It's essential to keep that because having no description is also a feature of a user profile and we would also like to evaluate that. Similarly, we created another feature called “text_length” for the text column. Here is the code snippet that creates two new columns for description and text lengths.

```
# Fill missing values in the 'description' column with an empty string
dataset['description'] = dataset['description'].fillna('')

# Create two new columns for description and text lengths
dataset['description_length'] = dataset['description'].apply(len)
dataset['text_length'] = dataset['text'].apply(len)
```

Length of text and description can be an important factor in clustering as it helps distinguish between different types of users (e.g., brands vs. regular users might have different behaviours in terms of text length).

	description	description_length	text	text_length
0	i sing my own rhythm	20	robbie e responds to critics after win against...	84
1	im the author of novels filled with family dra...	60	it felt like they were my friends and i was li...	83
2	louis whining and squealing and all	35	i absolutely adore when louis starts the songs...	80
3	mobile guy 49ers shazam google kleiner perkins...	126	hi jordanspieth looking at the url do you use ...	102
4	ricky wilson the best frontmankaiser chiefs th...	156	watching neighbours on sky catching up with th...	63

Figure 67: *Dataset with two new features*

- *Converting Categorical Variables to Numeric Values:* Since clustering algorithms like KMeans only work with numerical data and can not handle object data directly, the gender column was converted into a numerical form where Human (male and female) profiles are mapped as '0' and non-human or brand profiles mapped as '1'. Then the values were added to a new feature called "gender_encoded" to keep the record.

```
# Since we only need numeric values, we need to convert object type to
numeric values. So, map 'gender' column to numerical values: male -> 0,
female -> 0, brand -> 1
dataset['gender_encoded'] = dataset['gender'].map({'male': 0, 'female': 0,
'brand': 1})
```

	gender	gender_encoded
0	male	0
1	male	0
2	male	0
3	male	0
4	female	0

Figure 68: *Map 'gender' column to a numerical value*

Here is the overview of the dataset after the preprocessing steps:

	tweet_count	retweet_count	fav_number	description_length	text_length	gender_encoded
0	110964	0	0	20	84	0
1	7471	0	68	60	83	0
2	5617	1	7696	35	80	0
3	1693	0	202	126	102	0
4	31462	0	37318	156	63	0

Figure 69: *dataset with important features*

- *Standardisation:* Before applying the K-means clustering algorithm, it was necessary to standardise the features to ensure data integrity and uniformity. This was done using the MinMaxScaler function, which scales the values of each feature to the range of 0 to 1. We wanted to keep the magnitude of all the features the same. If one feature has a much larger range than others, it may dominate the clustering process providing inaccurate results. Scaling ensures that all features contribute equally (MinMaxScaler n.d.).

```
# Using MinMaxScaler for standardizing the dataset
from sklearn.preprocessing import MinMaxScaler
# Initialize MinMaxScaler
scaler = MinMaxScaler()
# Fit and transform the dataset
dataset2 = scaler.fit_transform(dataset2)
```

Here is the overview of the dataset after scaling the dataset:

	tweet_count	retweet_count	fav_number	description_length	text_length	gender_encoded
0	0.041401	0.00000	0.000000	0.12500	0.583942	0.0
1	0.002787	0.00000	0.000199	0.37500	0.576642	0.0
2	0.002095	0.00303	0.022528	0.21875	0.554745	0.0
3	0.000631	0.00000	0.000591	0.78750	0.715328	0.0
4	0.011738	0.00000	0.109238	0.97500	0.430657	0.0

Figure 70: *Dataset after Standardization*

The preprocessing steps taken such as cleaning and handling missing data, encoding, new feature engineering, and scaling are essential to prepare the dataset for clustering.

8.4. Model:

Preprocessing steps help ensure that the data is structured and ready for the KMeans clustering algorithm to perform effectively. Before using the K-Means clustering algorithm, one of the most important steps is finding out the number of clusters, known as the value of K. To do this, we need to run the K-Means algorithm with different K values and study the results to find the best number of clusters for our data. Choosing the right K is crucial for the algorithm to work well and give accurate results.

Determining the Optimal Number of Clusters

One of the most effective methods of determining the value of K or the number of clusters is using the Elbow method.

- *The Elbow Method:* Using this approach, we can determine how many clusters the dataset should be grouped into, making the clusters more meaningful and well-defined. We first set a range of potential cluster numbers (from 1 to 10). For each value of K, we applied the K-Means algorithm to the dataset, focusing on key features like 'tweet_count', 'retweet_count', 'fav_number', 'description_length', and 'text_length'. This process involved calculating the sum of squared distances

between each data point and its closest cluster centre for different numbers of clusters (Lee 2024). To see the relationship between the number of clusters and these distances, we created an "Elbow Graph."

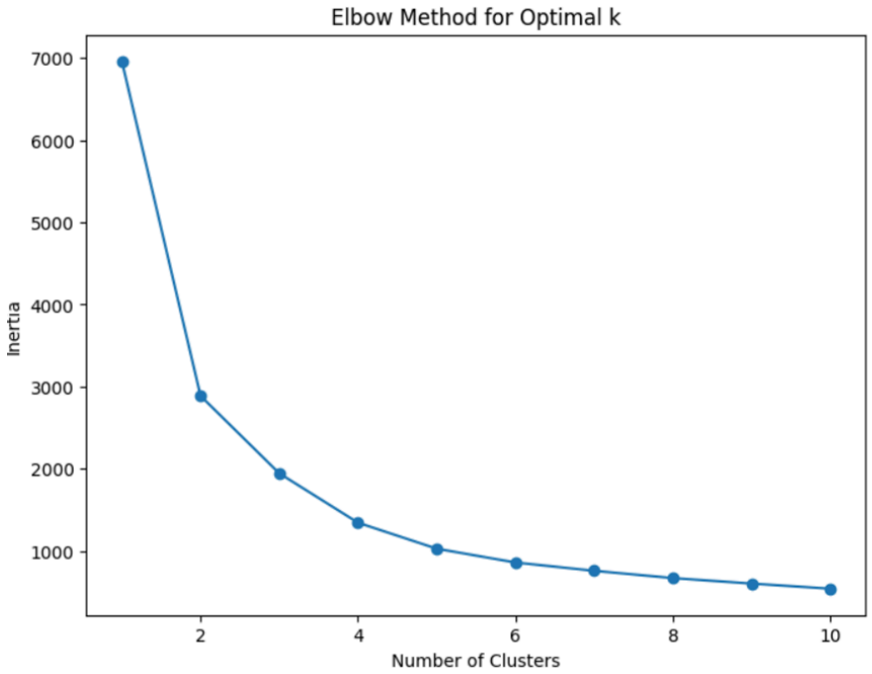

Figure 71: *The Elbow graph*

The x-axis of the graph shows the number of clusters, and the y-axis represents the inertia values. The point where the line starts to bend, forming an elbow shape, indicates the best number of clusters. In this case, the graph suggests that 2 is the ideal number of clusters.

Perform K-Means Clustering

Once we've identified the ideal number of clusters using the Elbow method, in this scenario it is 2, we can apply the K-Means clustering algorithm.

- *Initialize Cluster Centers:* The process begins by selecting k initial cluster centres, known as centroids. These are chosen randomly from the data points (KMEANS n.d.).
- *Assign Data Points:* Next, each data point is assigned to the nearest cluster by calculating the sum of squared distances. The distance from each data point to every centroid is measured, and the data point is assigned to the closest cluster (Dabbura 2022).

- *Update Cluster Centers:* After assigning all data points, the centroids are updated. This is done by calculating the average (mean) of all the data points within each cluster, which becomes the new centroid (Jeffares 2021).
- *Iterate:* The steps of assigning data points and updating centroids are repeated until the centroids stop changing, or the iteration is stopped after reaching a predefined number of iterations (Dabbura 2022).

Implementation

K-Means was applied to the standardised numeric dataset. Here is the code snippet for implementing KMeans clustering:

```
# Fit KMeans clustering model
kmeans = KMeans(n_clusters=2)  # We can set n_clusters to 2 based on the analysis of Elbow Method
#y_predicted    =    kmeans.fit_predict(dataset2[['tweet_count',    'retweet_count', 'fav_number','description_length', 'text_length']])
y_predicted = kmeans.fit_predict(dataset2)
# Saving the clustering result in dataset
dataset['cluster']=y_predicted
dataset2['cluster']=y_predicted
```

- *Cluster Label:* Finally, a new column named cluster was added to the dataset which indicates which group the user profile belongs.

  The purpose of K-means is to identify patterns in data by grouping data points that share similar characteristics. We can consider clustering well if the data points within a cluster are close to each other and the data points from different clusters are separated by each other and not overlapping.

- *Visualising the Clusters:* Now, we can start analysing the data to visualise user profiles and their characteristics that belong to the same cluster, so that we can find a better understanding of the correlations and similarities of data points within a cluster and dissimilarities and differentiable features among different clusters that set them apart.

  This bar chart shows how Human (male and female) and Brand profiles are grouped in two different clusters.

X-Axis: Represents the two clusters, labelled as Cluster 0 (Human) and Cluster 1 (Brand).

Y-Axis: Shows how many profiles are in each cluster.

After analysing cluster 0 (blue bar) we can determine this group has mostly Human profiles, with more than 12,000 users in this cluster. In cluster 1 (orange bar), this group is mainly made up of Brand profiles, with around 6,000 brands. The algorithm has been able to separate human users and brand accounts into two clusters. This shows that the features used for clustering, like tweet activity (tweet, retweet, favourite) and profile details (description and text), helped distinguish between human users and brands effectively.

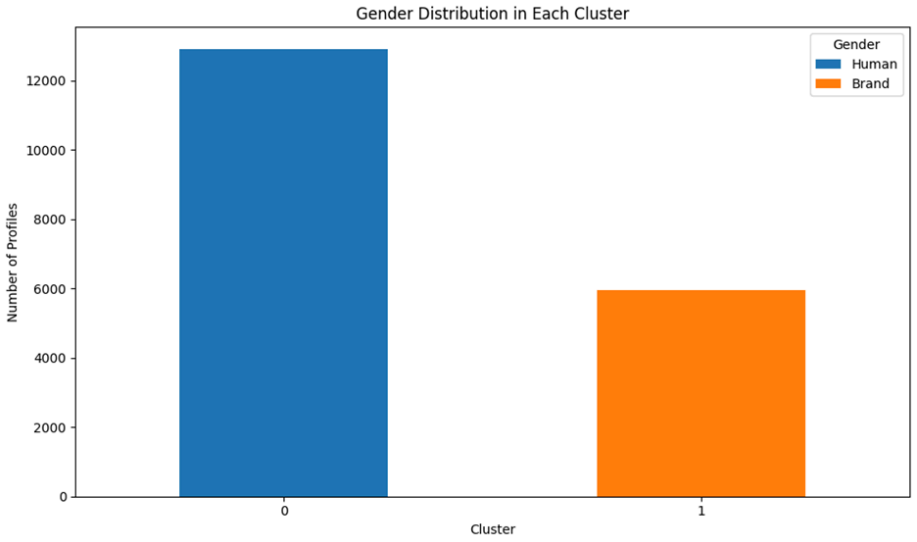

Figure 72: *Gender distribution in each cluster*

8.5. Results & Factors Considered:

Silhouette Score

To evaluate the quality of the clustering, the Silhouette Score was calculated to determine how well the KMeans algorithm performed. Here is the score:

Silhouette Score: 0.68

The Silhouette Score ranges from -1 to +1. A score close to 1 means the clusters are well-separated, while a negative score indicates that the data points overlap, meaning the clustering wasn't well-defined (silhouette_score n.d.). In our K-Means model, we got a positive average Silhouette Score of 0.68, which shows that our model is performing well.

The K-Means algorithm was successful in dividing the users into two distinct clusters, highlighting clear differences in user behaviour based on tweets, retweets, favourites, and gender.

Gender-Based Clustering

We have also produced silhouette scores separately for each gender group (male, female, and brand), showing variations in clustering quality across different gender categories.

This bar chart shows the Silhouette Scores for three different groups: male, female, and brand profiles.

- The Silhouette score for males is 0.92, meaning that male profiles are very well clustered. Most male profiles are grouped closer within their cluster and are far from other clusters.
- The Silhouette score for females is 0.83, which is slightly lower than male profiles which indicates there is a bit more overlap or similarity with other clusters.
- The Silhouette score for the brand is 0.95, the highest of all groups. This shows that brand profiles are very distinct from other clusters and most well-defined.

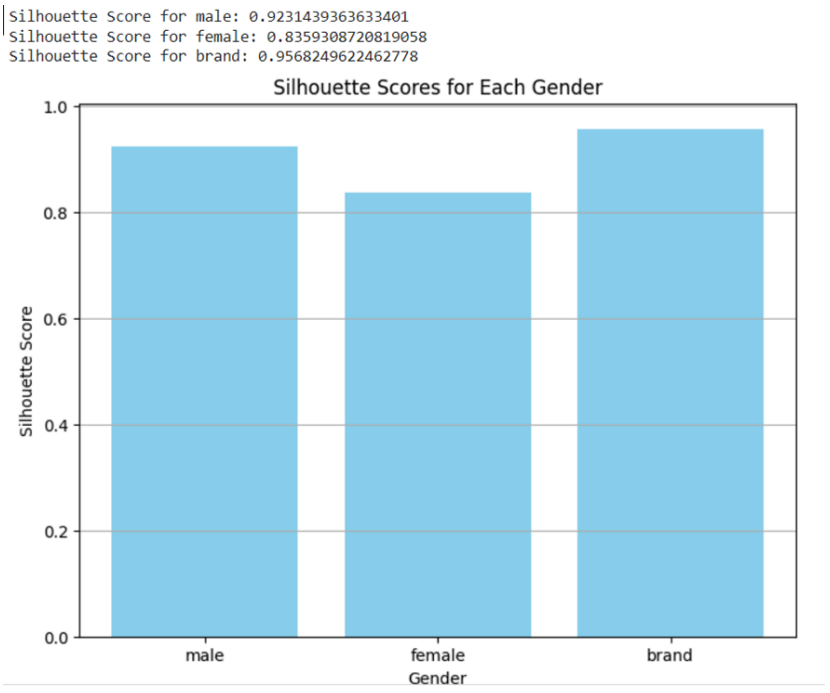

Figure 73: *Silhouette score for each gender*

Result Visualisation

Scatter plots were generated to visually analyse the clusters, showing how the clusters vary across different feature pairs such as tweet count vs. retweet count, tweet count vs

favourite number, tweet count vs description length, etc. The visualisations provided clear insights into how different user profiles belong to different clusters based on their features.

This is an overview of all feature pair combinations across different clusters. This grid of scatter plots visualises the relationships between different features (e.g., tweet count, retweet count, favourite number, description length, text length, and gender) across two clusters.

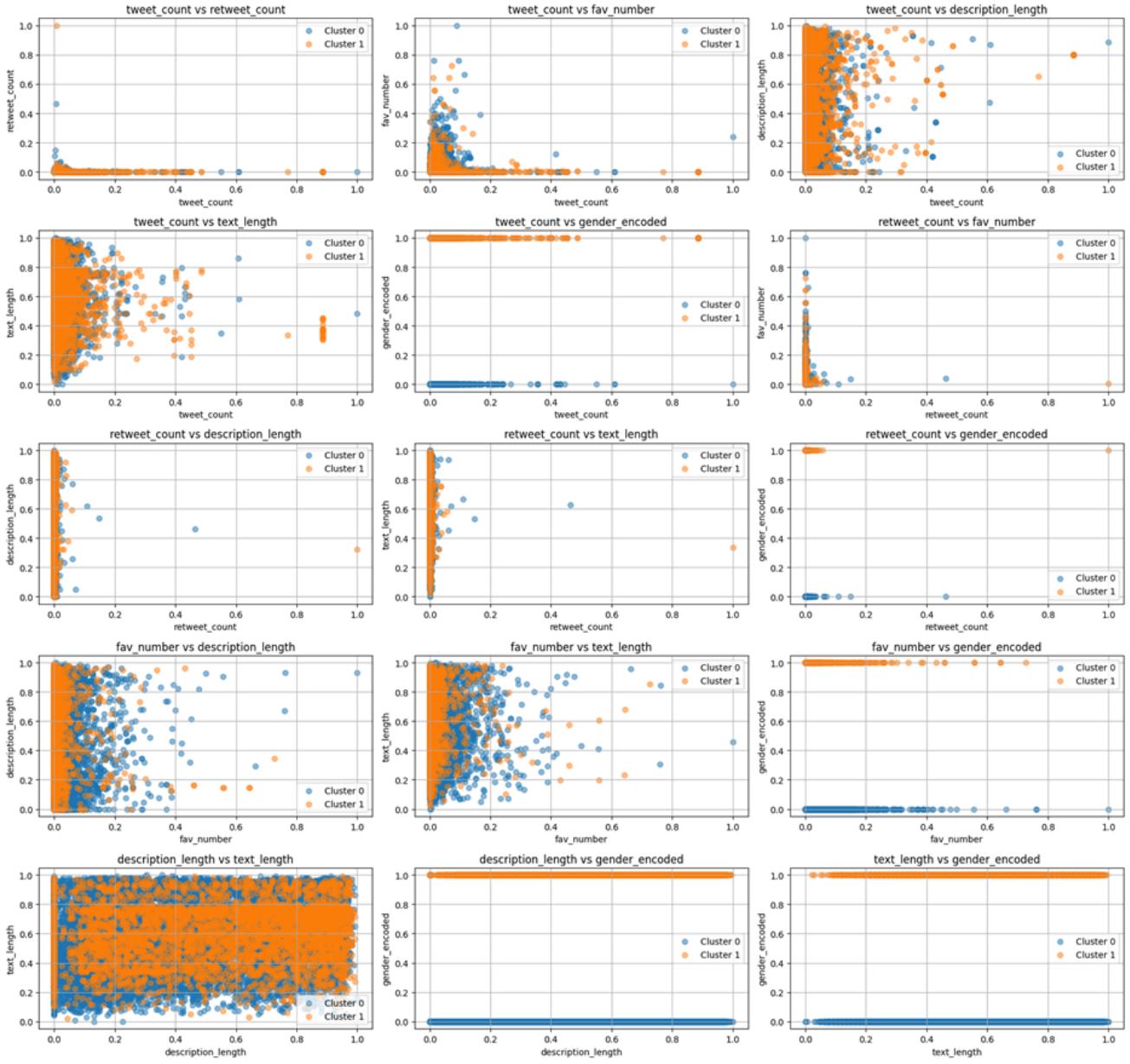

Figure 74: *Grid view of scatter plots of different features*

Each of the plots is analysed individually below.

- Cluster 0 (blue) represents Human profiles (both male and female)
- Cluster 1 (orange) represents Brand profiles.

- *Scatter plot of tweet count vs retweet count:* This scatter plot shows the relationship between tweet_count (number of tweets posted) on the x-axis and retweet_count (number of retweets received/posted) on the y-axis, for two clusters.

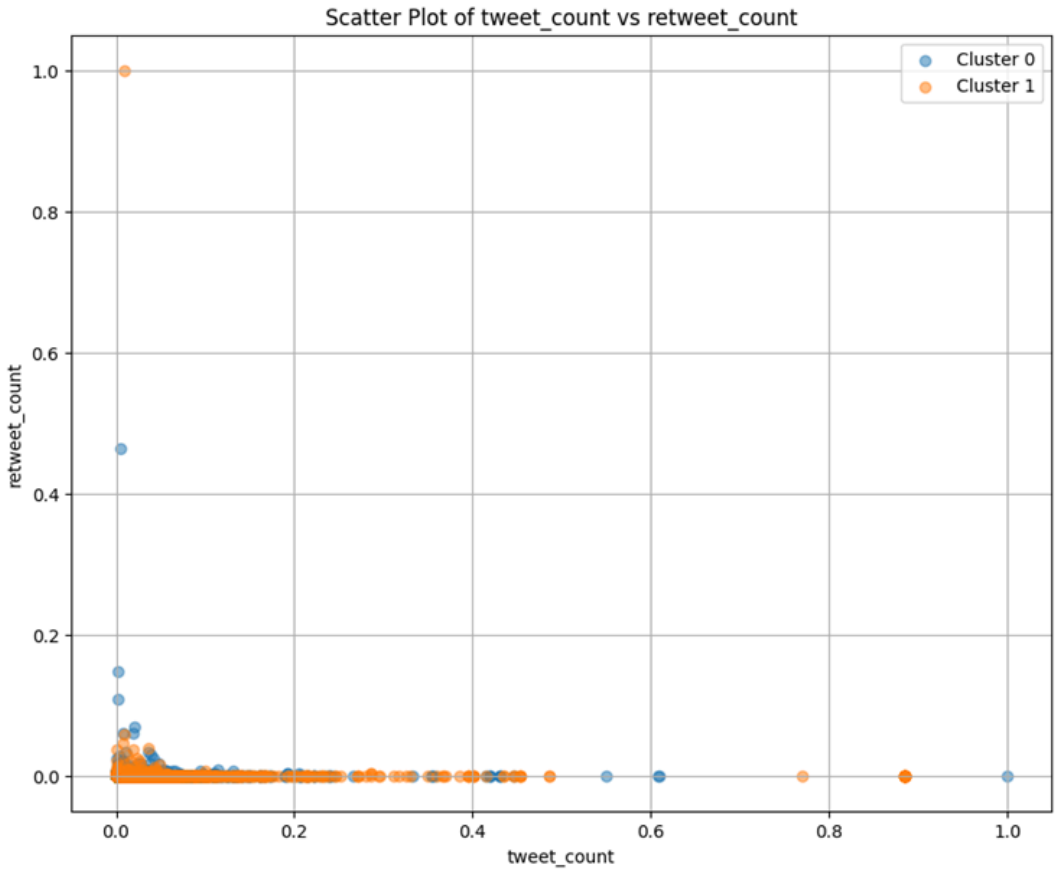

Figure 75: *Scatter plot of tweet count vs retweet count*

By looking at the plot we can observe that, most human and brand profiles have low tweet and retweet counts, as most of the data points are clustered near zero. However, a few brand profiles have higher tweet and retweet counts compared to human profiles. The orange points representing brands tend to cluster around zero for both tweet and retweet counts. This suggests that brands generally have lower engagement compared to human profiles.

- *Scatter plot of tweet count vs favourite number:* This scatter plot shows the relationship between tweet_count (number of tweets posted) on the x-axis and fav_number (number of favourites/likes received) on the y-axis for two different groups.

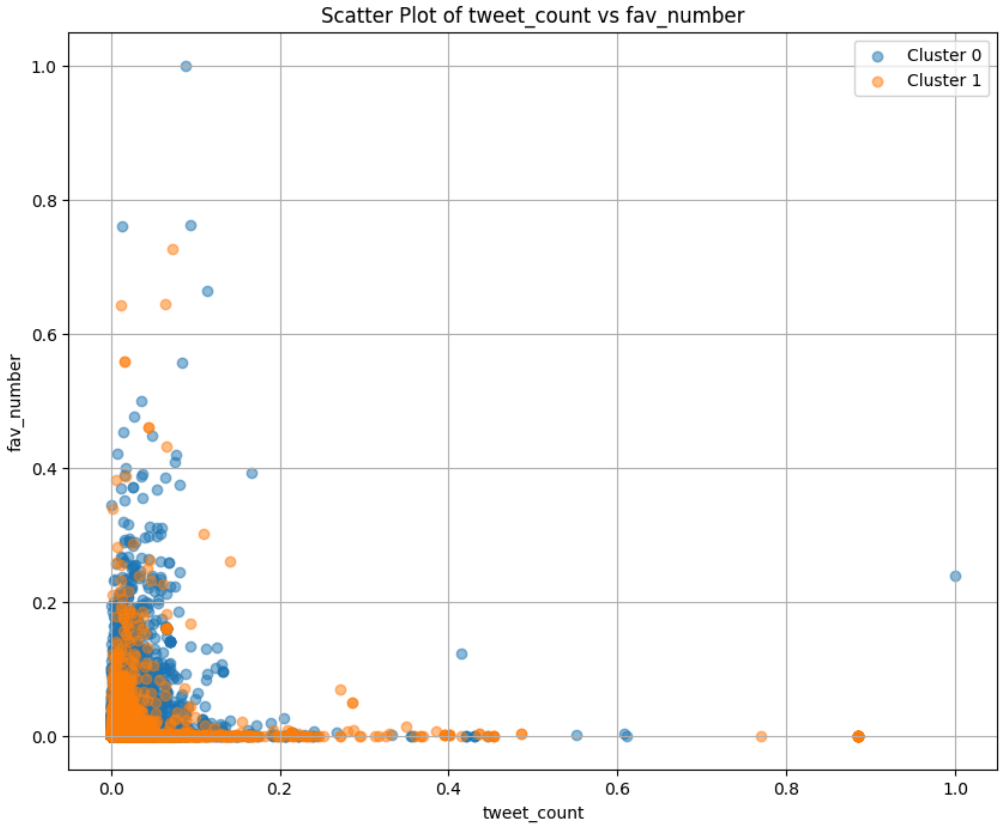

Figure 76: *Scatter plot of tweet count vs favourite number*

Most of the points for both human and brand profiles are concentrated in the bottom-left corner, meaning they have low tweet counts and receive few favourites. However, a few human profiles (blue points) have higher numbers of favourites, even with relatively few tweets. Despite that, we can see that some human profiles (blue points) have a higher number of favourites despite having relatively fewer tweets. It suggests that human profiles contain more engaging content that is favourite by many.

- *Scatter plot of tweet count vs description length:* This scatter plot shows the relationship between tweet_count (number of tweets posted) on the x-axis and description_length (length of the user's profile description) on the y-axis for two groups.

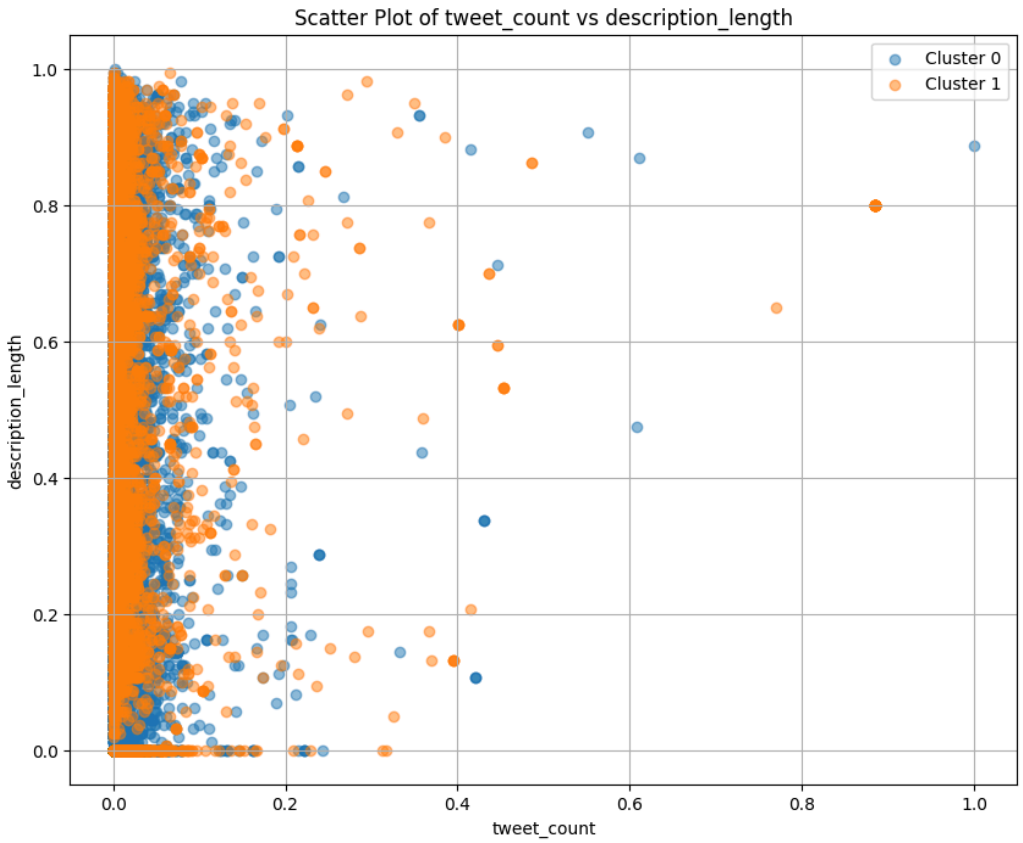

Figure 77: *Scatter plot of tweet count vs description length*

The majority of data points are tightly clustered along the left side of the plot, where both tweet count and description length are close to zero. This means most users, both human and brand, tend to have short profile descriptions and post very few tweets. As the description length increases, both human and brand profiles begin to be widespread. Some users have long descriptions even if they post very few tweets. This suggests that, for many users, the amount of detail in their profile description does not directly relate to how active they are in tweeting. Some brands show a wider range of description lengths, with some having very short descriptions while others have longer, more detailed descriptions. Overall, we can say the length of the profile description doesn't necessarily correlate with how active a user is in tweeting.

- *Scatter plot of tweet count vs text length:* This scatter plot shows the relationship between tweet_count (number of tweets posted) on the x-axis and text_length (the length of a sample tweet) on the y-axis for two groups.

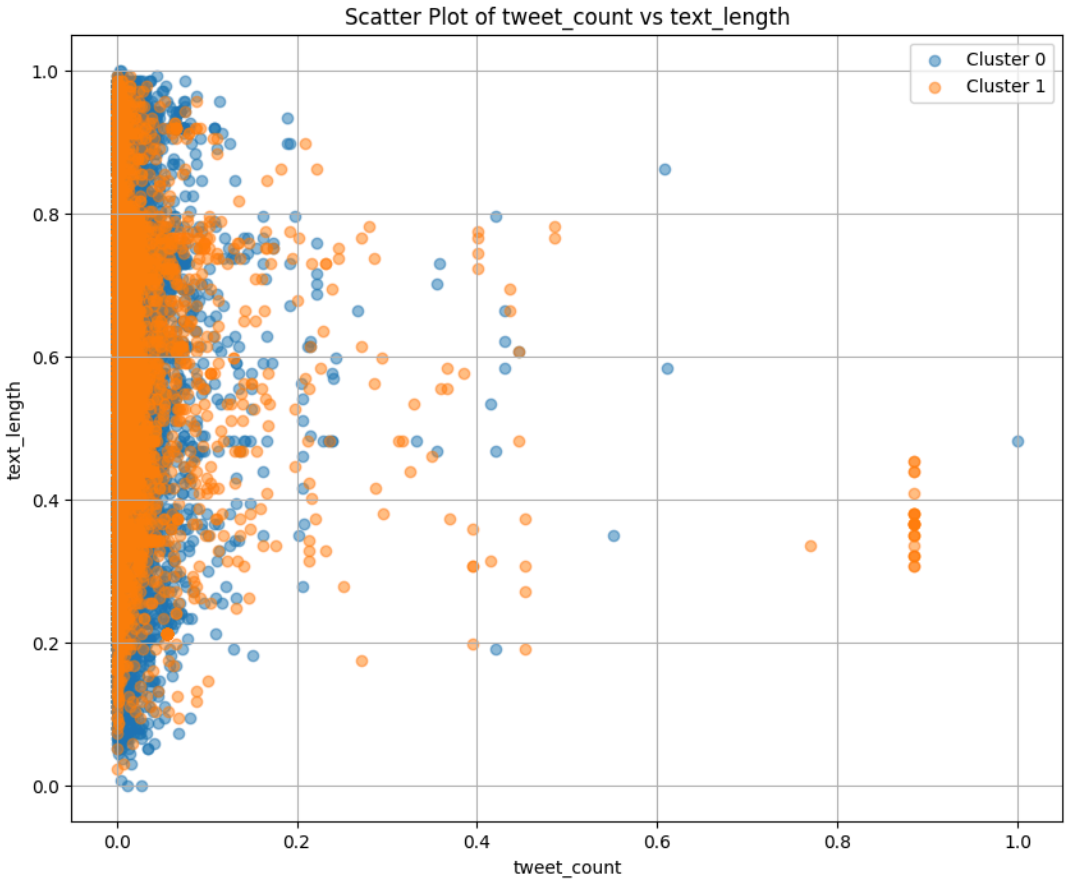

Figure 78: *Scatter plot of tweet count vs text length*

The majority of data points are concentrated along the left side of the graph, indicating that most users, both humans and brands, tweet have a low tweet count. However, the text length (length of the tweets) varies significantly, ranging from very short to very long tweets. This suggests that tweet length doesn't necessarily correlate with how often someone tweets. Both human and brand profiles show similar patterns when it comes to text length. There isn't a major difference between how humans and brands tweet.

- *Scatter plot of tweet count vs gender encoded:* This scatter plot shows the relationship between tweet_count (number of tweets posted) on the x-axis and gender_encoded (gender represented as a numerical value) on the y-axis for two groups.

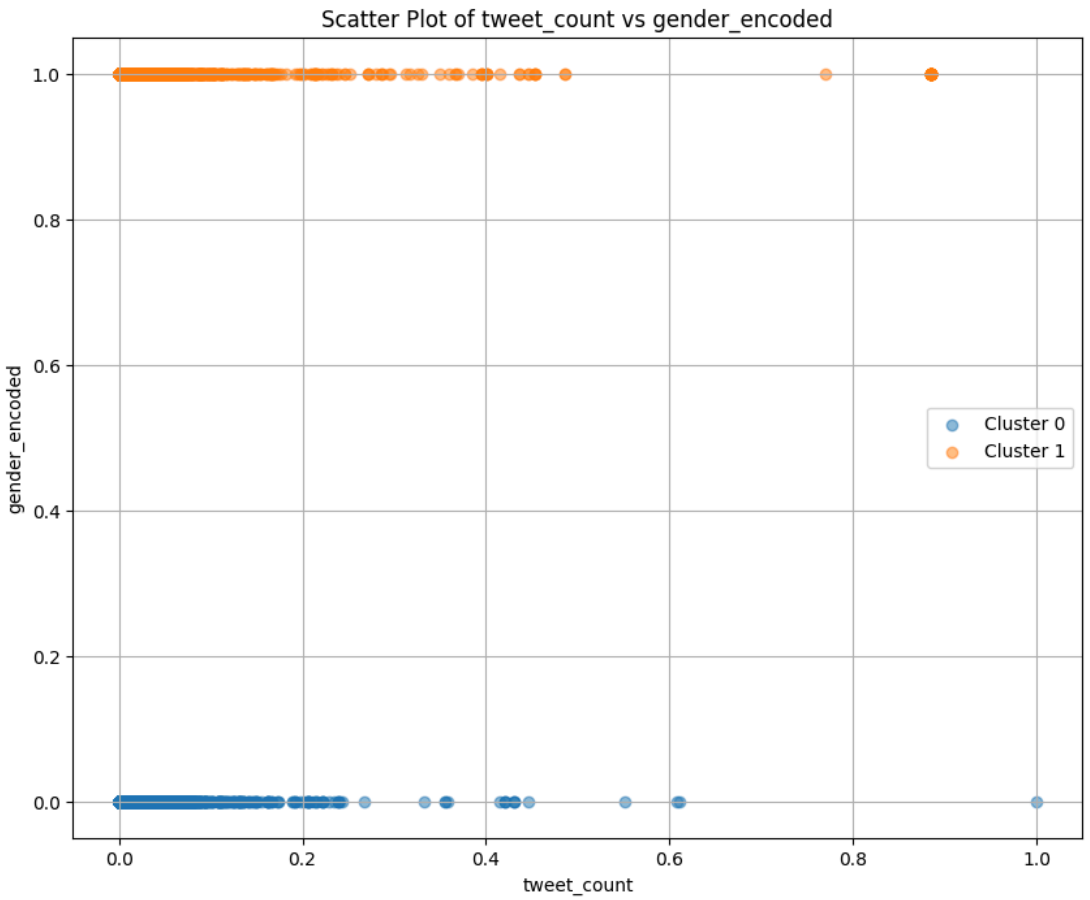

Figure 79: *Scatter plot of tweet count vs gender encoded*

There is a clear separation between the human profiles (blue points) and brand profiles (orange points) because Cluster 0 includes humans and Cluster 1 includes brands, reflecting the two types of users based on gender. Both humans and brands are mostly clustered at lower tweet counts, showing that most users in both categories have low tweet activity. A small number of brand profiles and human profiles show higher tweet counts.

- *Scatter plot of retweet count vs favourite number:* This scatter plot shows the relationship between retweet_count (number of retweets) on the x-axis and fav_number (number of favourites or likes received) on the y-axis for two groups.

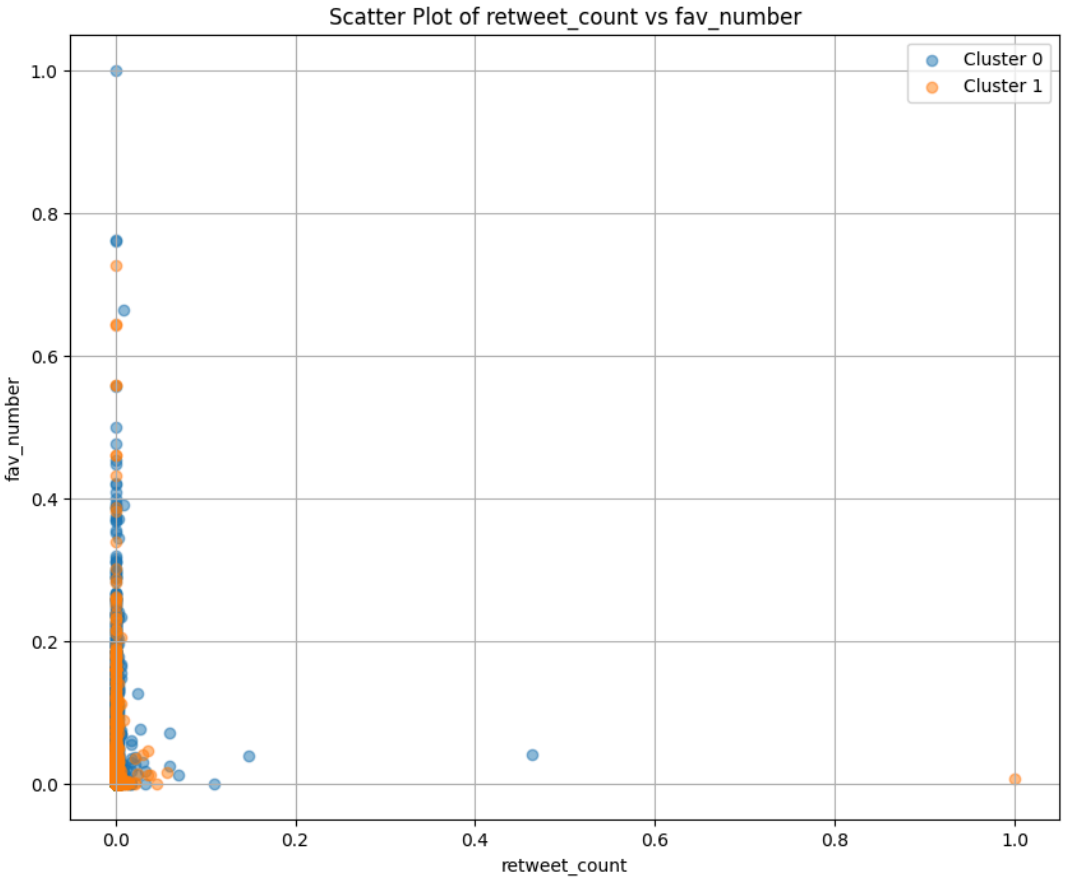

Figure 80: *Scatter plot of retweet count vs favourite number*

The majority of data points are clustered near the bottom-left corner, where both retweet counts and favourite counts are close to zero. This means that most users, whether human or brand, receive very few retweets and favourites on their posts. For most users, increasing retweets doesn't directly correspond to an increase in favourites, and vice versa. As a result, we can say no strong correlation can be seen between the number of retweets and favourites. Both human and brand profiles exhibit similar engagement patterns, with the vast majority having low retweets and favourites.

- *Scatter plot of retweet count vs description length:* This scatter plot shows the relationship between retweet_count (number of retweets received) on the x-axis and description_length (the length of the user's profile description) on the y-axis for two groups.

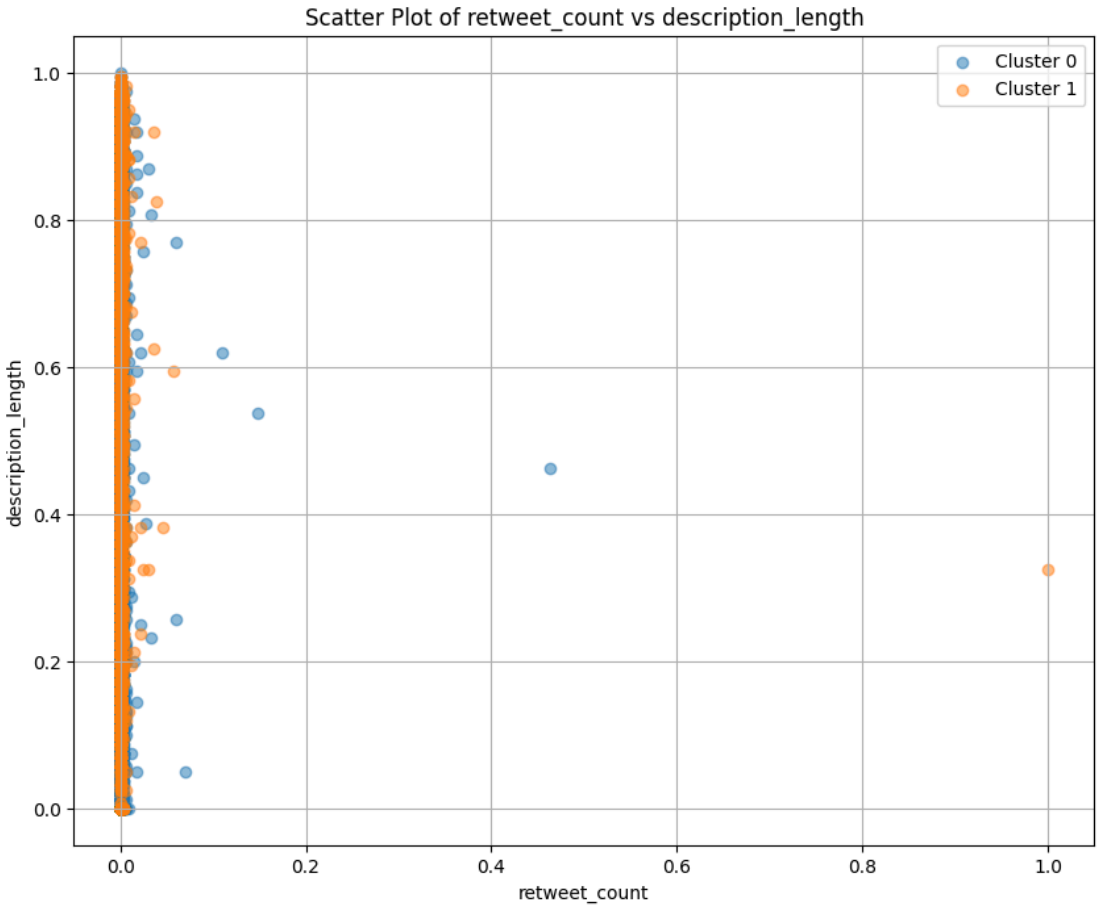

Figure 81: *Scatter plot of retweet count vs description length*

Most data points are tightly clustered along the left side of the graph which indicates that both human and brand profiles generally receive very few retweets. There's a wide variety in profile description lengths. Some profiles have very short descriptions, while others are more detailed, but this variation in description length doesn't appear to be strongly related to retweet activity. This suggests that how much information a user provides in their profile doesn't significantly affect how much their tweets are retweeted.

- *Scatter plot of retweet count vs text length:* This scatter plot shows the relationship between retweet_count (number of retweets received) on the x-axis and text_length (length of the tweet) on the y-axis for two groups:

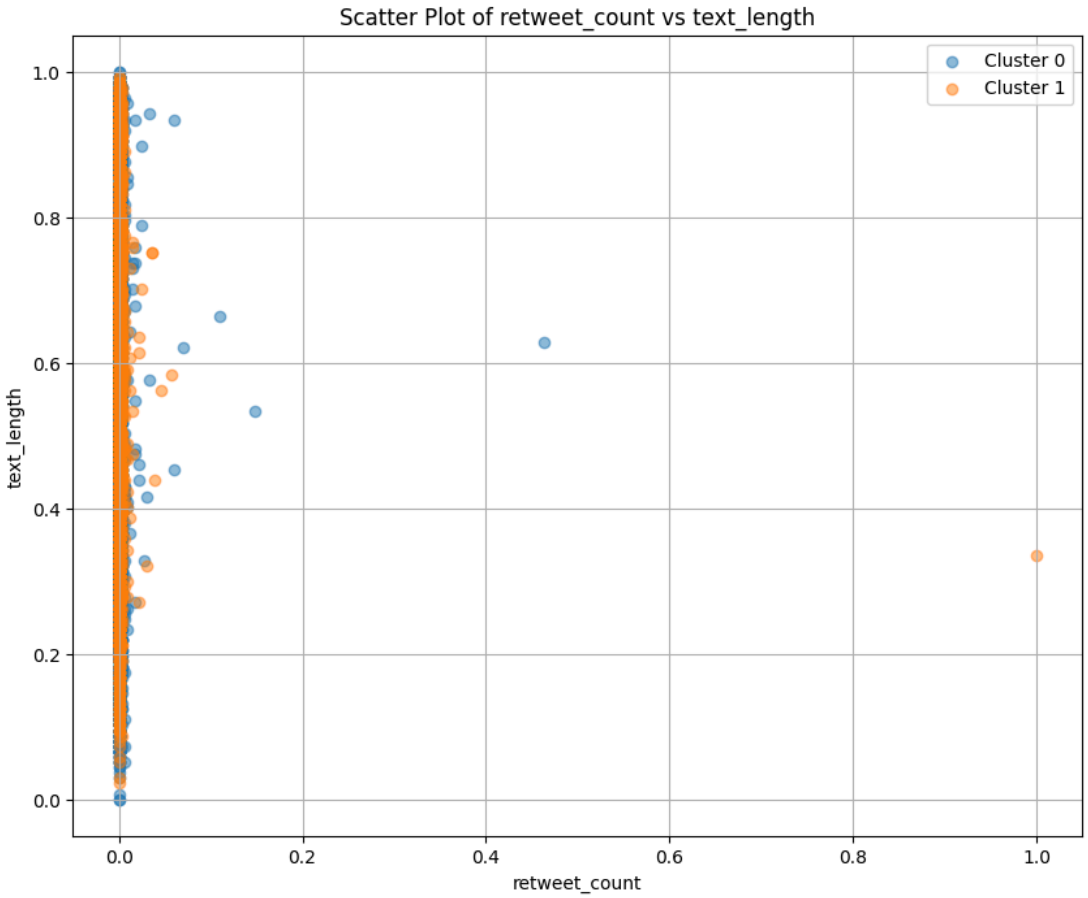

Figure 82: *Scatter plot of retweet count vs text length*

The majority of data points are clustered on the left side, indicating that most users, both human and brand, receive very few retweets. This is a common pattern across both types of profiles. The points show a wide range in text length meaning that some tweets are very short, while others are longer. However, most users have a low retweet count regardless of whether their tweets are long or short, suggesting no strong relationship between tweet length and the number of retweets.

- *Scatter plot of retweet count vs gender encoded:* This scatter plot shows the relationship between retweet_count (number of retweets received) on the x-axis and gender_encoded (gender represented as a numerical value) on the y-axis for two groups.

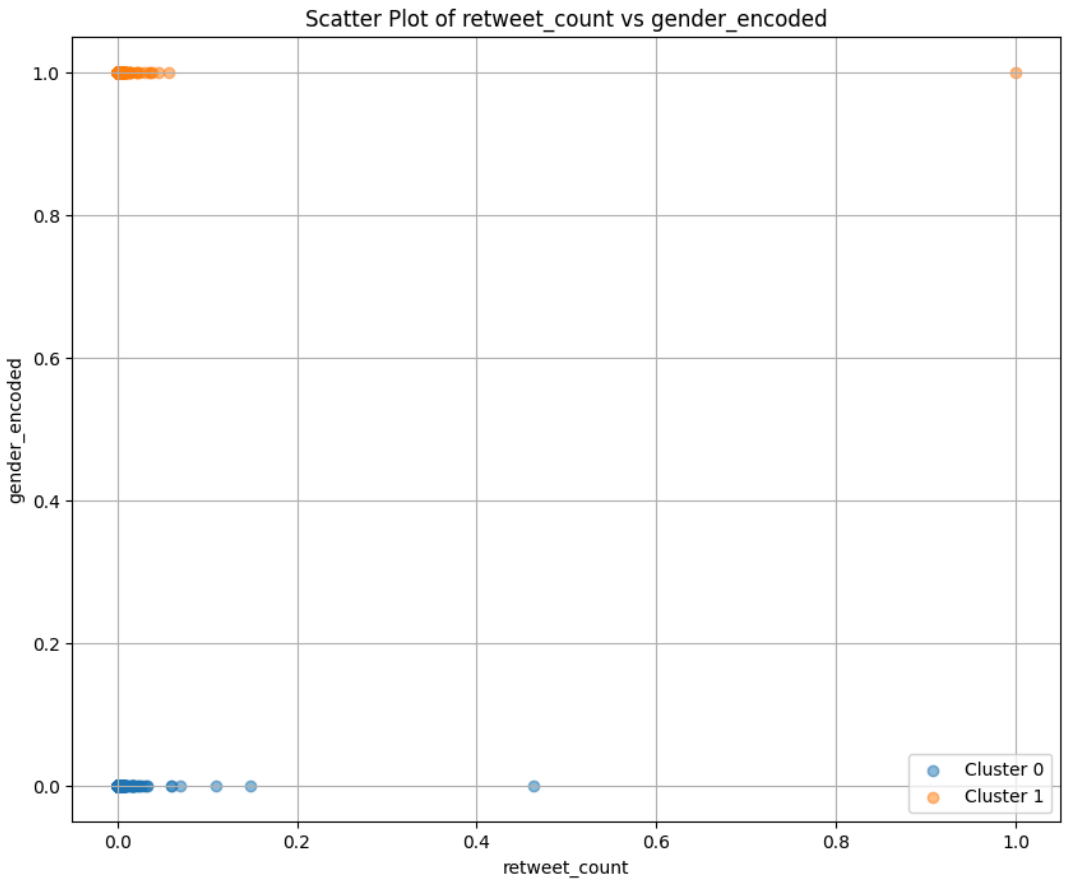

Figure 83: *Scatter plot of retweet count vs gender encoded*

Human and brand profiles are clearly separated. The majority of data points for both human and brand profiles are clustered close to retweet count 0. This indicates that most users, whether human or brand, receive very few retweets on their tweets.

- *Scatter plot of favourite number vs description length:* This scatter plot shows the relationship between fav_number (number of favourites or likes received) on the x-axis and description_length (length of the user's profile description) on the y-axis for two groups.

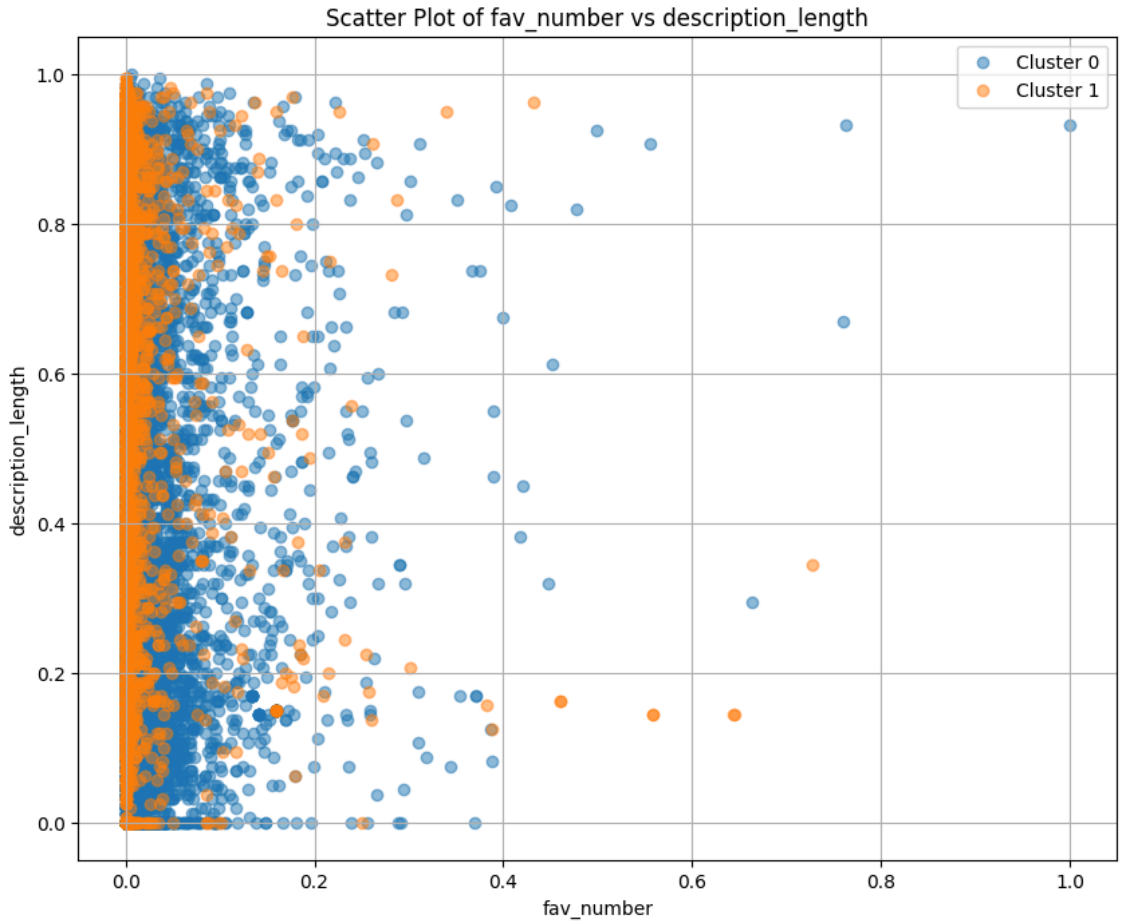

Figure 84: *Scatter plot of favourite number vs description length*

Most of the users, both human and brand, receive very few favourites (likes) on their tweets. This is consistent with the pattern of low engagement across most profiles. The points are vertically spread, indicating that the length of profile descriptions varies greatly among users, even those with low favourite counts. Some users have very short descriptions, while others have longer, more detailed profiles, regardless of how many favourites their tweets get. There is no significant distinction between the two groups.

- *Scatter plot of favourite number vs text length:* This scatter plot shows the relationship between fav_number (number of favourites or likes received) on the x-axis and text_length (the length of a sample tweet) on the y-axis for two groups.

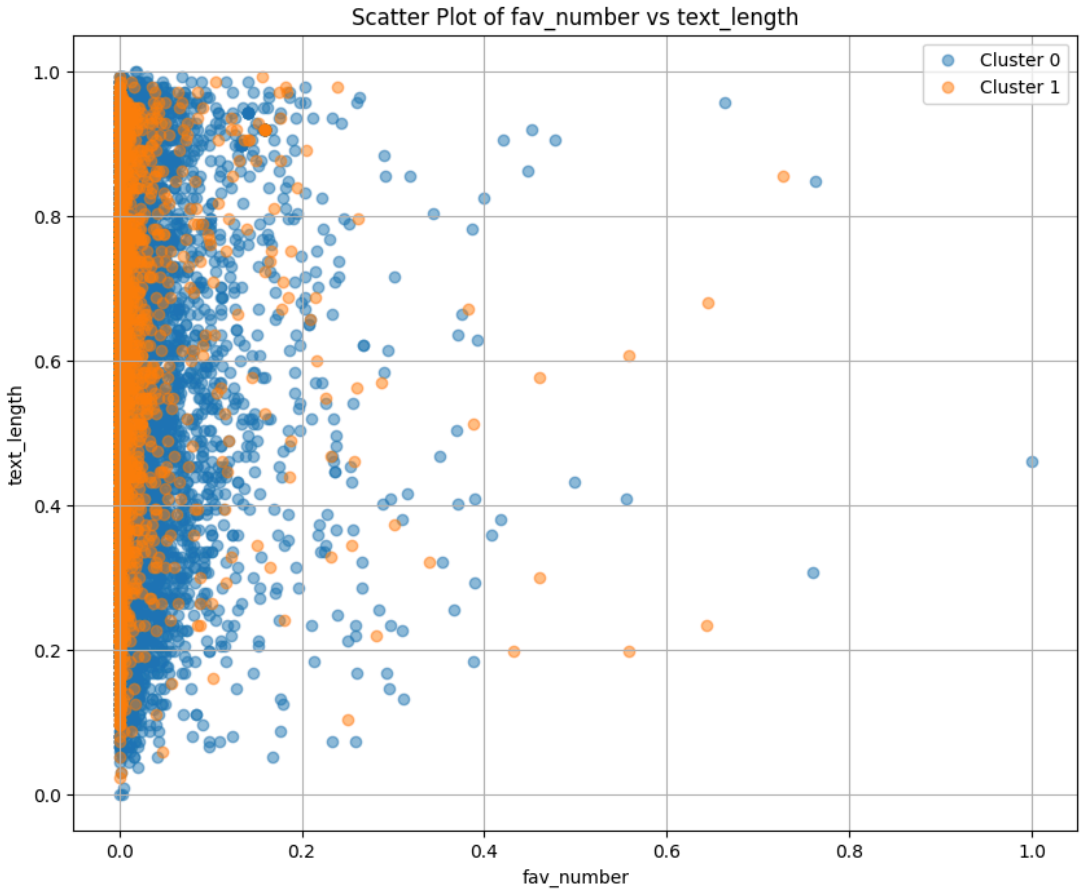

Figure 85: *Scatter plot of favourite number vs text length*

There is significant variation in text length, with tweets ranging from very short to very long, regardless of how many favourites they receive. This shows that tweet length doesn't have a strong relationship with the number of favourites a tweet gets. A few tweets from both human and brand profiles receive higher numbers of favourites, but this level of engagement is uncommon.

- *Scatter plot of favourite number vs gender encoded:* This scatter plot shows the relationship between fav_number (number of favourites or likes received) on the x-axis and gender_encoded (gender represented as a numerical value) on the y-axis.

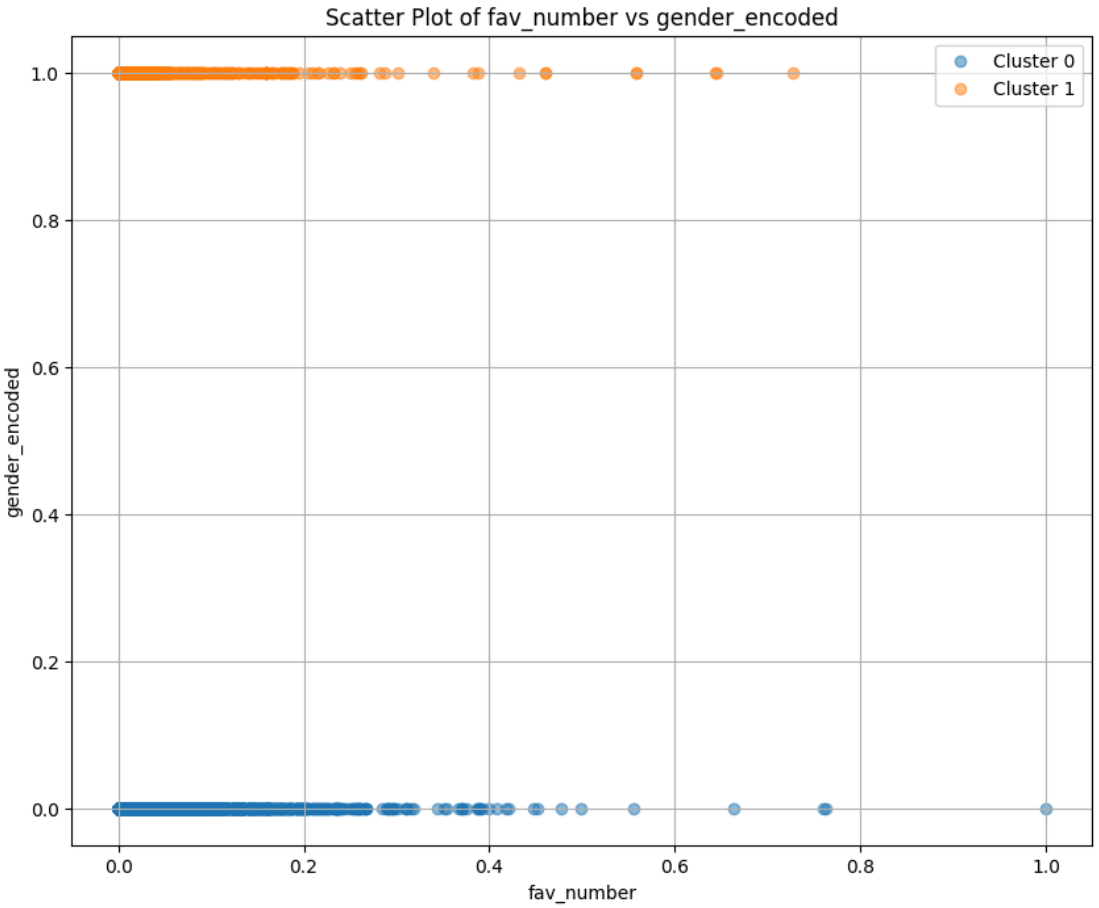

Figure 86: *Scatter plot of favourite number vs gender encoded*

There is a clear separation of human and brand profiles which makes it easy to distinguish the two groups based on gender encoding. Most data points for both humans and brands are clustered near the range of around 0.0 – 0.5, whether humans or brands receive very few favourites on their tweets. There are a few data points where some human profiles (Cluster 0, blue) and brand profiles (Cluster 1, orange) have received more favourites although this behaviour is rare. So we can say, both groups exhibit similar engagement patterns in terms of receiving few favourites on their tweets.

- *Scatter plot of description length vs text length:* This scatter plot shows the relationship between description_length (the length of a user's profile description) on the x-axis and text_length (the length of a tweet) on the y-axis for two groups.

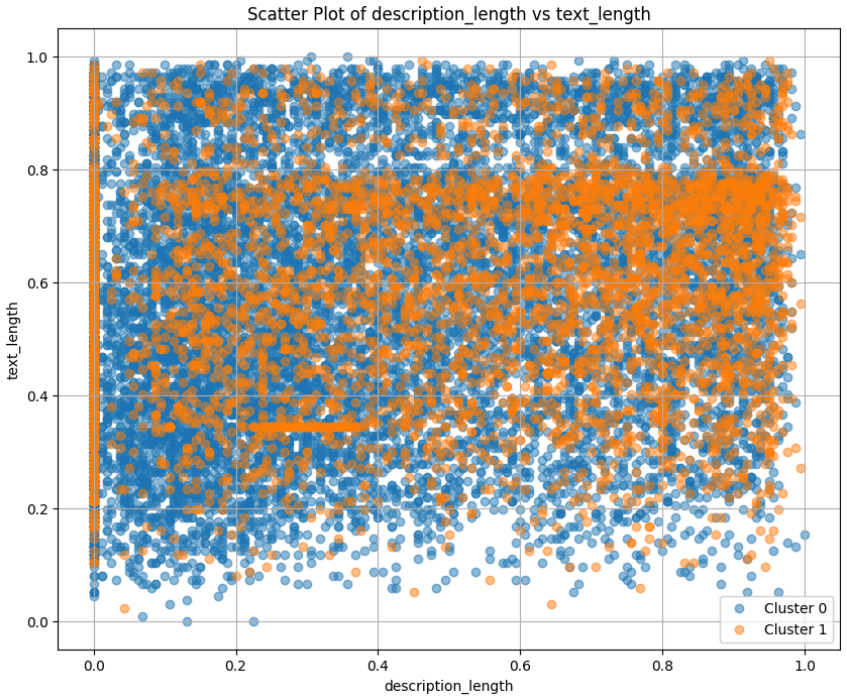

Figure 87: *Scatter plot of description length vs text length*

The points are spread across a wide range of description lengths and text lengths. This indicates that users, both human and brand, have varied lengths in their profile descriptions and tweets. Some users write very short descriptions or tweets, while others use long descriptions or text. The points are scattered with no clear pattern, suggesting that there is no strong correlation between the length of a user's profile description and the length of their tweets. Users can have long descriptions but short tweets, or vice versa, without any consistent relationship between the two.

Suggestions

To identify profiles that may have been mistakenly recorded as human or non-human (brand), we can look for patterns in behaviour that do not align with the typical behaviours of human or brand profiles. Here are some suggestions:

- Brands typically show lower engagement compared to human profiles. If a profile classified as a brand shows consistently high engagement (many favourites or retweets), it might be worth investigating whether it was misclassified. On the other hand, if a profile classified as human consistently has very low engagement (close to zero favourites or retweets), it might be a brand mistakenly classified as human.

- Brands are more likely to post more frequently. If a profile is labelled as human posts with high frequency and shorter tweets (similar to brand behaviour), it may be misclassified as human.

- Human profiles should show irregularity and diversity in engagement, tweet content length, and tweet frequency. If a human profile behaves otherwise, it could be a brand misclassified as human.

The results computed by this model indicate that the hypothesis stated in Section 1 is accepted.

9. Factors Studied:

9.1. Profile Study: Human vs. Non-Human:

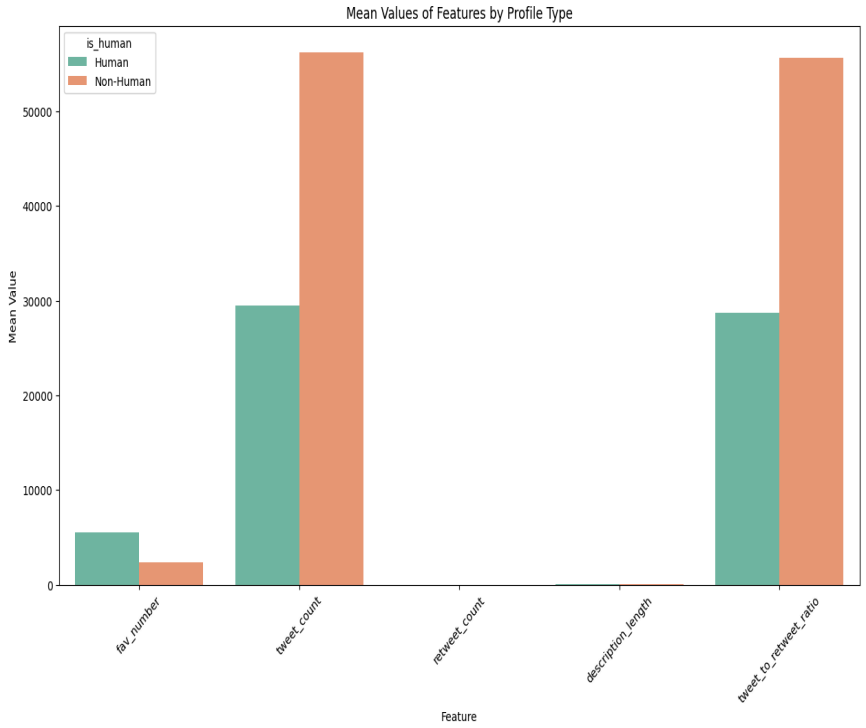

Figure 88: *Mean Values of Features by Profile Type*

The figure compares the mean values of several features—fav_number, tweet_count, retweet_count, description_length, and tweet_to_retweet_ratio—between human and non-human profiles. From the graph, we can observe the following trends:

- *Fav_number:* Humans have a higher mean value for the number of favorites, indicating greater engagement with tweets.
- *Tweet_count*: Non-human profiles exhibit a significantly higher mean tweet count, suggesting automated or high-frequency activity.
- *Retweet_count:* There is minimal difference between human and non-human profiles in terms of the retweet count.

- *Description_length:* Both human and non-human profiles have similar, low description lengths.
- *Tweet_to_retweet_ratio*: Non-human profiles show a much higher mean value for the tweet-to-retweet ratio, likely reflecting their content generation nature compared to human profiles.

This analysis highlights the behavioral differences between human and non-human profiles based on these selected features.

9.2. Factors in Multiple Views:

We studied the features across various dimensions (numeric attributes like fav_number, tweet_count, and more categorical views like gender) to identify trends and patterns. The figure below presents a scatterplot matrix showing pairwise relationships between different features: `fav_number`, `tweet_count`, `retweet_count`, `description_length`, and `tweet_to_retweet_ratio`, separated by "Human" and "Non-Human" profiles.

Key observations include:

- *Fav_number vs. Tweet_count:* Non-human profiles tend to have fewer favorites but higher tweet counts compared to humans, as seen by the clustering of points. Hence, the initial hypothesis discussed is Section 1 is rejected.
- *Retweet_count:* Both profile types exhibit lower retweet counts, with few extreme values.
- *Description_length:* The length of the description shows a more diverse spread for human profiles, indicating that human profiles may put more effort into describing themselves. Hence, the initial hypothesis discussed is Section 1 is rejected.
- *Tweet_to_retweet_ratio:* Non-human profiles have significantly higher tweet-to-retweet ratios, as indicated by the points clustering along the top-right corner of the plot.
- *Diagonal density plots:* The kernel density estimates along the diagonal further emphasize the differences between the profile types, with a clear distinction in the `tweet_to_retweet_ratio` and `tweet_count` distributions.

This matrix helps visualize the underlying distributions and relationships between features for both profile types, highlighting the clear behavioral differences in posting and engagement habits.

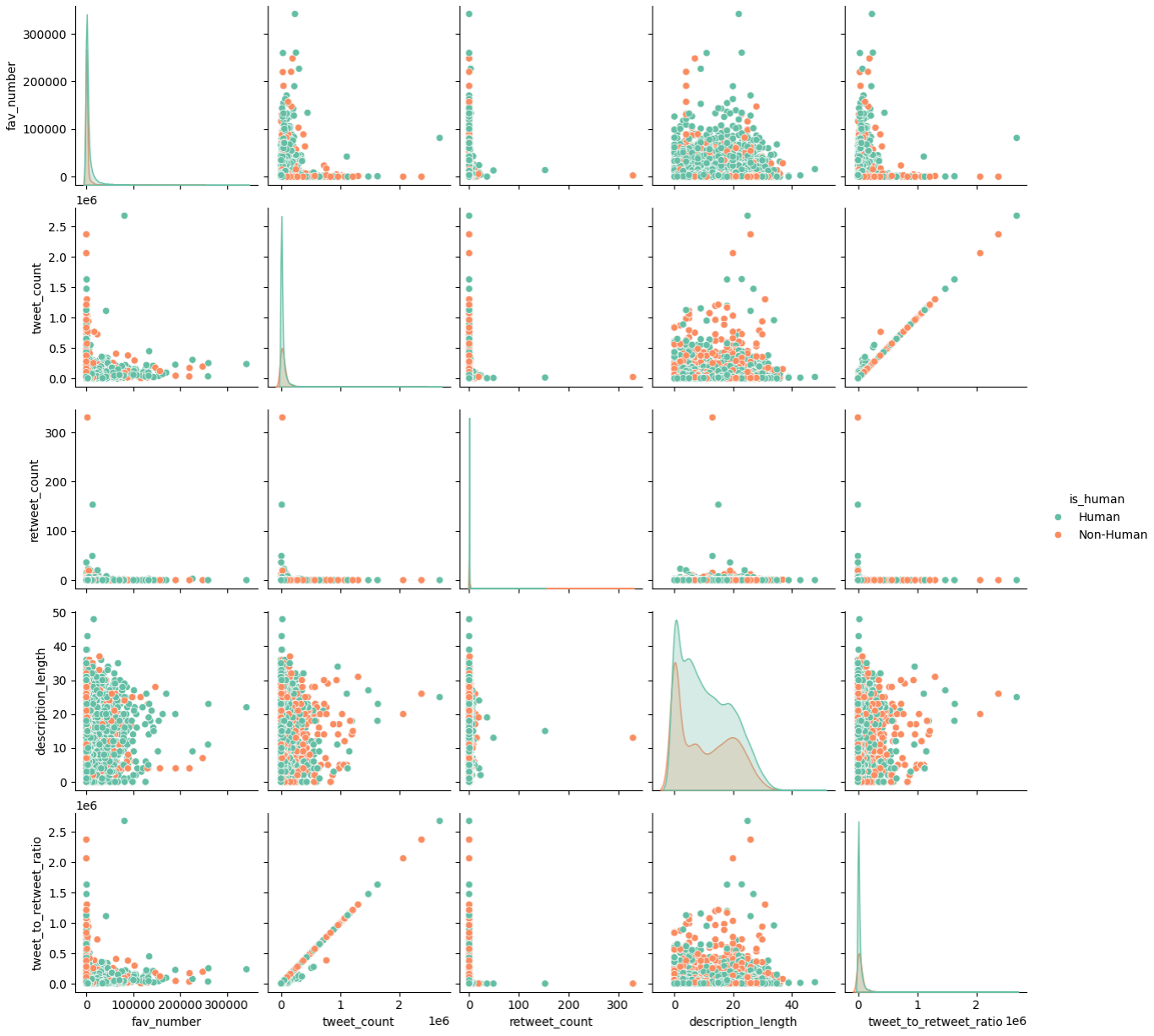

Figure 89: *Pairplot comparing feature relationships for humans vs non-humans*

10. Suggestions:

This section examines factors in the "twitter_user_data" dataset that may differentiate between human and non-human profiles, focusing on metrics such as 'fav_number', 'tweet_count', 'retweet_count', 'description_length', 'tweet_to_retweet_ratio', 'gender', and 'profile_yn'. Each factor is analyzed for its relevance in detecting misinformation (i.e., incorrectly labeled human or non-human profiles), followed by suggestions for improving classification accuracy.

1. **Fav Number:** The 'fav_number' (i.e., the number of tweets favorited by a user) reflects engagement behavior. Human users typically show selective and varied interaction with content, whereas bots might exhibit extreme favoriting behavior to increase engagement artificially. Studies have shown that bots frequently engage in bulk liking and retweeting, resulting in unusually high favorite counts compared to humans (Ferrara et al., 2016).

    - *Amendment Suggestion:* Profiles with disproportionately high or low favorite numbers should be flagged for potential bot behavior. Machine learning models could use this feature to improve classification between human and non-human profiles.

2. **Tweet Count:** The 'tweet_count' indicates the total number of tweets posted by a user. Human users generally tweet irregularly, while bots are programmed to tweet more frequently, often at fixed intervals (Chu et al., 2012). High tweet frequencies are a common marker of automated accounts.

    - *Amendment Suggestion*: Set a threshold for tweet counts to identify bot-like activity. Profiles exceeding an average daily tweet count should be investigated, as this behavior is often associated with automation.

3. **Retweet Count:** Bots frequently retweet content to amplify specific topics or accounts, often leading to unusually high retweet counts. Human users tend to exhibit more varied behaviors in their engagement with original tweets and retweets (Subrahmanian et al., 2016).

    - *Amendment Suggestion*: Analyze the 'retweet_count' relative to the overall 'tweet_count'. A high retweet-to-tweet ratio could be a significant indicator of non-human activity, and this feature should be incorporated into bot-detection algorithms.

4. **Description Length:** Human users usually provide meaningful and personalized profile descriptions, while bots may have either very short or excessively long descriptions, often filled with keywords for promotional purposes (Dickerson, Kagan, & Subrahmanian, 2014). Bots tend to focus on maximizing visibility rather than conveying personal information.

    - *Amendment Suggestion:* Profiles with very short or long descriptions should be flagged as potential non-human profiles. NLP techniques can be used to

analyze the content for patterns indicating automation, such as repetitive keywords or overly generic language.

5. **Tweet to Retweet Ratio:** The 'tweet_to_retweet_ratio' offers insight into the balance of original content creation versus content sharing. Bots often exhibit extreme behaviors, either focusing heavily on retweeting or consistently generating content (Cresci et al., 2019).

    - *Amendment Suggestion:* Profiles with an unusually high or low tweet-to-retweet ratio should be investigated. This ratio, when combined with other factors, is a strong predictor of bot activity and can enhance classification models.

6. **Gender:** Although not a direct indicator of bot behavior, 'gender' can be informative when analyzed in combination with other attributes. Non-human profiles often leave this field blank or fill it with generic information, as they do not require real gender assignment (Alarifi, Alsaleh, & Al-Salman, 2016).

    - *Amendment Suggestion:* Missing or ambiguous gender information, particularly when paired with other suspicious attributes, can signal a non-human profile. Profiles with missing or inconsistent gender should be prioritized for further analysis.

7. **Profile Y/N:** The 'profile_yn' field indicates whether a user's profile is complete. Bots are more likely to have incomplete profiles, especially when fields such as descriptions and profile pictures are missing (Varol et al., 2017).

    - *Amendment Suggestion:* Profiles marked as 'no' for this feature should be flagged for potential non-human activity. The absence of key profile details, such as descriptions or images, is often a reliable indicator of bots and can be used in classification models to improve detection accuracy.

Conclusion

The factors discussed above can be effectively used to improve the classification of human versus non-human profiles in social network datasets. By analyzing behaviors such as tweet frequency, retweeting patterns, and description length, it is possible to detect bot-like profiles more accurately. These features, when incorporated into machine learning models, will enhance the detection of profiles spreading misinformation on social networks.

References:

Alarifi, A., Alsaleh, M., & Al-Salman, A. (2016). "Twitter turing test: Identifying social machines." *Information Sciences*, 372, 332-346.

Arbelaitz, Olatz, et al. *An extensive comparative study of cluster validity indices*. vol. 46, ScienceDirect, 2012. *https://www.sciencedirect.com/journal/pattern-recognition*, https://www.sciencedirect.com/science/article/abs/pii/S003132031200338X.

Ananda 2021, Twitter-based gender Classification -A machine learning project, https://www.analyticsvidhya.com/blog/2021/08/twitter-based-gender-classification-a-machine-learning-project/.

Atefeh, Farzindar, and Wael Khreich. *A Survey of Techniques for Event Detection in Twitter*. vol. 31, WILEY Online Library, 2013, https://onlinelibrary.wiley.com/doi/abs/10.1111/coin.12017.

Blei, DM, Ng, AY & Jordan, MI 2003, 'Latent dirichlet allocation', *J. Mach. Learn. Res.*, vol. 3, no. null, pp. 993–1022.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.

Biau, G. (2012). Analysis of a Random Forests Model. *Journal of Machine Learning Research*, 13, 1063-1095.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794).

Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2012). "Detecting automation of Twitter accounts: Are you a human, bot, or cyborg?" *IEEE Transactions on Dependable and Secure Computing*, 9(6), 811-824.

Cresci, S., Lillo, F., Regoli, D., Tardelli, S., & Tesconi, M. (2019). "Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter." *Proceedings of the ACM Conference on Hypertext and Social Media*, 83-92.

DATAROBOT. 2020. *Text processing: what, why, and how* [Online]. DataRobot. Available: https://www.datarobot.com/blog/text-processing-what-why-and-how/#:~:text=Text%20processing%20is%20one%20of,spam%20filtering%2C%20and%20many%20others. [Accessed 13 September 2024].

Dabbura, I 2022, K-Means Clustering: algorithm, applications, evaluation methods, and drawbacks, Medium, https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.

Dawari, A 2023, What are the advantages and disadvantages of K-Means Clustering?, Towards AI, https://towardsai.net/p/l/what-are-the-advantages-and-disadvantages-of-k-means-clustering.

DEEPANSHI. 2024. *Text Preprocessing in NLP with Python Codes* [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/text-preprocessing-in-nlp-with-python-codes/#:~:text=Text%20preprocessing%20is%20an%20essential,part%2Dof%2Dspeech%20tagging. [Accessed 13 September 2024].

Dickerson, J. P., Kagan, V., & Subrahmanian, V. S. (2014). "Using sentiment to detect bots on Twitter: Are humans more opinionated than bots?" *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, 620-627.

Ester, Martin, et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.” *A Density-Based Algorithm for Discovering Clusters*, no. 1996, 1996, pp. 226-231. *AAAI Press*, https://file.biolab.si/papers/1996-DBSCAN-KDD.pdf.

Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8), 861-874.

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). "The rise of social bots." *Communications of the ACM*, 59(7), 96-104.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232.

Gareth James, 2023, ‘An Introduction to Statistical Learning’, <https://www.statlearning.com/>.

GEEKSFORGEEKS. 2022. ML | Binning or Decretization [Online]. GeeksforGeeks. Available: https://www.geeksforgeeks.org/ml-binning-or-discretization/ [Accessed 18 September 2024].

GEEKSFORGEEKS. 2023. *Understanding TF-IDF (Term Frequency-Inverse Document Frequency)* [Online]. GeeksforGeeks. Available: https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/ [Accessed 13 September 2024].

GEEKSFORGEEKS. 2024. One Hot Encoding in Machine Learning [Online]. GeeksforGeeks. Available: https://www.geeksforgeeks.org/ml-one-hot-encoding/ [Accessed 17 September 2024].

GeeksforGeeks 2024, K means Clustering Introduction, https://www.geeksforgeeks.org/k-means-clustering-introduction/.

Gupta, A., 2024, ‘Feature Selection Techniques in Machine Learning’, *Analytics Vidhya*, viewed 18 August 2024, <https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/>.

Han, J., Pei, J. and Kamber, M., 2011. *Data Mining: Concepts and Techniques*. 3rd ed. Waltham, MA: Morgan Kaufmann.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). John Wiley & Sons.

IBM.com, ‘What is Random Forest’, <https://www.ibm.com/topics/random-forest>.

IMARTICUS. 2023. *Why is Noise Removal Important for Datasets?* [Online]. Imarticus. Available: https://imarticus.org/blog/why-is-noise-removal-important-for-datasets/ [Accessed 16 September 2024].

Jain, A.K, et al. *Data Clustering: A Review*. http://users.eecs.northwestern.edu/~yingliu/datamining_papers/survey.pdf.

Jain, R 2023, “K-Means Clustering: use cases, advantages and working principle,” Bombay Softwares, https://www.bombaysoftwares.com/blog/introduction-to-k-means-clustering.

Jeffares, A 2021, K-Means: A complete introduction - towards data science, Medium, https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c.

Kapadia, S 2019, *Topic Modeling in Python: Latent Dirichlet Allocation (LDA). [online] Medium*, Towards Data Science, viewed 16 Sep 2024, <https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0>.

Kavlakoglu, E & Winland, V 2024, “K-Means Clustering,” IBM, https://www.ibm.com/topics/k-means-clustering.

KMEANS, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.

Kumar, A 2024, A simple explanation of K-Means clustering, https://www.analyticsvidhya.com/blog/2020/10/a-simple-explanation-of-k-means-clustering/

Lee, S 2024, Implementing K-Means clustering with Python for data analysis, Medium, https://medium.com/@sunqlee1004/implementing-k-means-clustering-with-python-for-data-analysis-43c41f8390d1.

Mathworks.com ‘What is feature engineering’ <https://au.mathworks.com/discovery/feature-engineering.html>.

Matplotlib.org ‘Matplotlib’, <https://matplotlib.org/>.

MEDIUM. 2023. Data Binning Explained [Online]. Medium. Available: https://medium.com/@mose.kabungo/binning-explained-557aa3cce591 [Accessed 18 September 2024].

MinMaxScaler, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.htm.

NUMFOCUS, I. 2024a. *pandas.DataFrame.drop_duplicates* [Online]. Available: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop_duplicates.html [Accessed 13 September 2024].

NUMFOCUS, I. 2024b. *pandas.DataFrame.dropna* [Online]. Available: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dropna.html [Accessed 13 September 2024].

Olmez, R. 2024, 'Handling Missing Values in Data Science / Machine Learning Projects: Strategies and Practice', *Medium*, viewed 17 August 2024, <https://medium.com/@ramazanolmeez/handling-missing-values-in-data-science-machine-learning-projects-strategies-and-practice-42d7376ca94a#:~:text=1.-,Introduction,values%20or%20carefully%20complete%20them>.

Pandas pydata.org, <https://pandas.pydata.org/>.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

PulkitS 2024, Introduction to K-Means clustering, https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/.

R, V. K. G. 2024. *12] Handling Missing Values in Machine Learning: Different Missing Values Handling Methods in Machine Learning with Practical Implementation* [Online]. Available: https://medium.com/@vinodkumargr/12-handling-missing-values-in-machine-learning-different-missing-values-handling-methods-in-24b4ce55fe70#:~:text=Null%20values%20can%20introduce%20bias,performance%20and%20suboptimal%20predictive%20accuracy. [Accessed 13 September 2024 2024].

Rousseeuw, Peter. "Enhanced Parameter Estimation of DENsity CLUstEring(DENCLUE) Using Differential Evolution." *Mathematics*, no. 2024, 2024, https://www.researchgate.net/publication/384051613_Enhanced_Parameter_Estimation_of_DENsity_CLUstEring_DENCLUE_Using_Differential_Evolution.

Scatter plot — Matplotlib 3.9.2 documentation, https://matplotlib.org/stable/gallery/shapes_and_collections/scatter.html.

Sebastian Raschka, 2022 'Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python <https://www.packtpub.com/en-us/product/machine-learning-with-pytorch-and-scikit-learn-9781801819312>.

silhouette_score, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html.

Snowflake.com 'What is gradient boosting', <https://www.snowflake.com/guides/what-gradient-boosting/>.

Sci-kit learn 'Train test split', <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html>.

Sci-kit learn ‘r2_score’, <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html>.

Sci-kit learn ‘Mean squared error’, <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html>.

Sci-kit learn ‘LabelEncoder’, <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>.

SHAH, P. 2020. *Sentiment Analysis using TextBlob* [Online]. Towards Data Science. Available: https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524 [Accessed 13 September 2024].

Subrahmanian, V. S., Azaria, A., Durst, S., Kagan, V., Galstyan, A., Lerman, K., ... & Menczer, F. (2016). "The DARPA Twitter bot challenge." *Computer*, 49(6), 38-46.

Tan, P.-N., Steinbach, M., Karpatne, A. and Kumar, V., 2018. *Introduction to Data Mining*. 2nd ed. Harlow: Pearson.

Twitter user gender classification 2016, https://www.kaggle.com/datasets/crowdflower/twitter-user-gender-classification.

Varol, O., Ferrara, E., Davis, C. A., Menczer, F., & Flammini, A. (2017). "Online human-bot interactions: Detection, estimation, and characterization." *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1).

Vladislav Raskoshinskii ‘Feature Importance’, <https://www.kaggle.com/code/raskoshik/feature-importance-how-not-fool-yourself>.

W3Schools.com, https://www.w3schools.com/python/python_ml_k-means.asp.

Wagavkar, S., 2023, ‘Introduction to The Correlation Matrix | Built In’, *Builtin*, viewed 18 August 2024, <https://builtin.com/data-science/correlation-matrix>.

Wickham, H., 2016, ‘Data Analysis’, pp.189–20.

Xu, R. & WunschII, D., 2005, ‘Survey of Clustering Algorithms’, *IEEE Transactions on Neural Networks*, vol.16, iss.3, pp.645–678.

Zaki, Mohammed J., and Wagner Meira Jr. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. 2013, https://pzs.dstu.dp.ua/DataMining/bibl/mohammed_j_zaki_wagner_meira_jr_data_mining_and_analysis_fun.pdf.