UNIVERSITY
OF WOLLONGONG
AUSTRALIA

CSCI946: Big Data Analytics

Assignment-2: Twitter User Gender Prediction by Group-09

Spring 2025

Team Members Contribution:

Student Name	Student ID	Email	Task Log	Contribution (%)
Anika Salsabil	8764736	as0891@uowmail.edu.au	Introduction, Task-1: Big Data Analytics Lifecycle, Task-2: EDA, Association Rules, Preprocessing documentation, Report documentation and team coordination	100%
Ye Hyun Chung	8927285	yhc815@uowmail.edu.au	Task-2: Text Processing, Task-3: Evaluation of results, Report Documentation	100%
Nayeem Sheak	8715087	ns079@uowmail.edu.au	Task-2: Clustering, Task-4 Peer Review, Report documentation	100%
Fahim Faisal Khan	8522765	ffk878@uowmail.edu.au	Task-2: Preprocessing, Classification, Regression, Report documentation	100%
Mahbubul Alam	8377807	ma668@uowmail.edu.au	Task-4: Research factors, suggestions, References	100%

Subject Coordinator:

Professor Lei Wang

Table of Contents

Introduction ... 5
1 Task 1: Big Data Analytics Lifecycle Design ... 5
1.1 Overview ... 5
1.2 Big Data Analytics Lifecycle: Project Roadmap ... 5
Phase 1: Discovery ... 6
Phase 2: Data Preparation ... 9
Phase 3: Model Planning ... 11
Phase 4: Model Building ... 13
Phase 5: Communicating Results ... 14
Phase 6: Operationalizing ... 15
2 Task 2 – Process & Model ... 16
2.1 Exploratory Data Analysis (EDA) ... 16
2.1.1 Dataset Overview ... 16
2.1.2 Target Variable Distribution (Gender) ... 17
2.1.3 Numerical Features Analysis ... 17
2.1.4 Categorical Features (Profile Colors) ... 18
2.1.5 Textual Features (Description Field) ... 18
2.1.6 Gender Confidence ... 18
2.1.7 Summary of EDA Findings ... 18
2.2 Data Preprocessing ... 19
2.2.1 Combining and Structuring the Dataset ... 19
2.2.2 Feature Cleaning and Missing Values ... 20
2.2.3 Outlier Detection and Removal ... 20
2.2.4 Scaling and Standardization ... 21
2.2.5 Output Artifacts ... 21
2.2.6 Justification of Dataset usage ... 21
2.2.7 One-glance Summary of Transformations ... 22
2.3 Classification & Regression ... 22
2.3.1 Approach ... 22
2.3.2 Data Preparation ... 23

2.3.3 Multinomial Naive Bayes 24

2.3.4 Logistic Regression 25

2.3.5 Decision Tree 26

2.3.6 Results 27

2.4 Clustering 30

2.4.1 Data Preparation for Clustering 30

2.4.2 KMeans Clustering 31

2.4.3 Hierarchical Clustering 32

2.4.4 Self-Organizing Maps (SOM) 33

2.4.5 Comparison of clustering algorithms 34

2.4.6 Justification 34

2.5 Association Rules 35

2.5.1 Data Preparation for Association Rules 35

2.5.2 Approach 35

2.5.3 Frequent Itemset Generation 36

2.5.4 Association Rules Mining 36

2.5.5 Results 36

2.5.6 Comparison 38

2.5.7 Justification 39

2.6 Text processing 39

2.6.1 Approach 39

2.6.2 Data Preparation 39

2.6.3 Word Frequency Analysis 40

2.6.4 Word Cloud 40

2.6.5 TF-IDF Feature Extraction 40

2.6.6 Topic Modeling by LDA 40

2.6.7 Vocabulary Diversity 41

2.6.8 Sentiment Analysis 41

2.6.9 Results 41

3 Task 3 – Visualization & Evaluation 42

3.1 Data Characteristics (Preprocessing & EDA) 42

3.2 Association Rule Insights 44

3.3 Clustering Results Evaluation 47
3.4 Classification Results 49
3.6 Text Processing Evaluation 52
4 Task 4 – Multi-view Study & Suggestions 54
4.1 Objective 54
4.2 Data Views Considered 54
4.3 Key Findings by View 55
i) Tweet/Activity View 55
ii) Color View 55
iii) Text View 56
iv) Label Confidence View 56
4.4 Suggestions to Amend/Adjust Profiles 57
4.4.1 For Non-Human (Brand) Profiles 57
4.4.2 For Human Profiles 57
4.4.3 Simple Rule-of-Thumb (to complement the model) 57
4.4 Summary 57
5 References 57

Introduction

The growth of social media platforms such as Twitter has created opportunities for analyzing user behavior and detecting misinformation. A key challenge in this context is identifying profiles that are mistakenly recorded as human or non-human, which is essential for improving trust and reliability in online interactions.

Assignment-2 addresses this problem by analyzing the twitter_user_data.csv dataset using a comprehensive data analytics approach. Guided by the Big Data Analytics Lifecycle, the assignment involves designing a project framework, exploratory data analysis (EDA), processing data of different types, and applying a range of analytical methods including regression, classification, clustering, association rule mining, and text processing. Python programming is used to implement these models, while visualization techniques support the evaluation and interpretation of results. The study also explores multiple perspectives such as text, tweet content, and profile features to suggest improvements for refining human vs non-human classification. Overall, the assignment demonstrates the application of diverse big data methods to a real-world problem of social media analytics.

1 Task 1: Big Data Analytics Lifecycle Design

1.1 Overview

The Data Analytics Lifecycle provides a structured roadmap for turning raw data into actionable insights [18]. It describes how data is generated, collected, processed, and analyzed to support specific business goals. By following this lifecycle, data teams can manage complexity systematically, ensuring that each stage—from discovery to deployment—is aligned with organizational objectives.

For Assignment-2, the lifecycle guides the process of analyzing the Twitter user dataset to detect profiles mislabeled as human or non-human. Beginning with business understanding and domain discovery, the lifecycle then emphasizes preparing data, planning and building models, evaluating results, and finally communicating and operationalizing outcomes. This systematic approach ensures that findings are reliable, reproducible, and relevant to the real-world challenge of reducing misinformation and improving data quality.

1.2 Big Data Analytics Lifecycle: Project Roadmap

To design this project effectively, we adopted the Big Data Analytics Lifecycle. **Figure 1.1** provides a systematic framework for approaching analytical problems in six interlinked phases: (i)

Discovery, (ii) Data Preparation, (iii) Model Planning, (iv) Model Building, (v) Communicate Results, and (vi) Operationalize.

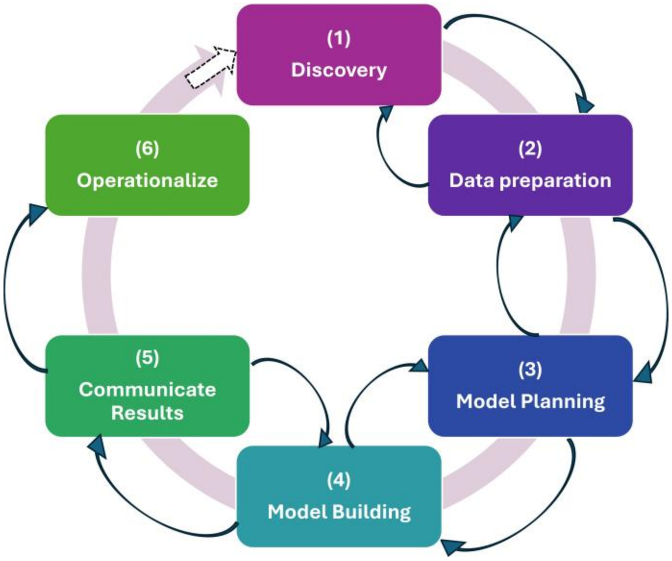


Figure 1.1: Big Data Analytics Lifecycle

In the context of this assignment, the lifecycle will guide our approach to analyzing the **Twitter user dataset with the objective of identifying misclassified profiles (human vs. non-human)**. The following sections document each phase in detail, outlining the methods, workflows, and deliverables associated with them. This structure ensures transparency, reproducibility, and alignment with both the assignment specification and best practices in data science.

Phase 1: Discovery

The discovery phase provides the foundation of the project by defining the business problem, identifying resources, framing objectives, and establishing initial hypotheses. For this assignment, the goal is to apply the Big Data Analytics Lifecycle to the **twitter_user_data** dataset to identify profiles that are mistakenly recorded as human or non-human, thereby reducing misinformation in social networks.

1. Learning Business Domain

Social media platforms, particularly Twitter, host a mixture of human users and non-human accounts such as brand profiles or bots. Mislabeled accounts introduce noise into analytics pipelines, mislead marketing strategies, and distort public discourse. The business domain of this project is therefore social media analytics for user integrity, where accurately identifying whether a profile represents a human or a brand is critical. The rise of sophisticated bots that mimic human language patterns make this task more challenging, underscoring the need for robust analytics methods that leverage both textual and metadata features of profiles.

2. Identify Resources

The primary resource is the **twitter_user_data.csv** dataset, which contains ~20,000 user profiles and 26 attributes, such as:

- **Textual data:** tweets, user descriptions, profile names.
- **Numerical data:** tweet counts, retweet counts, favorites, trusted judgments.
- **Categorical/visual cues:** link color, sidebar color, time zone, location.
- **Target variables:** gender (male, female, brand), profile yes/no, and confidence scores.

Additional resources include:

- Technical tools: Python (pandas, scikit-learn, mlxtend, matplotlib, seaborn).
- Knowledge resources: lecture slides on the Big Data Analytics Lifecycle, prior academic work, and existing works of training a CrowdFlower AI gender predictor project from web as references.

3. Defining the Problem

The core problem is: How can we identify misclassified Twitter profiles (human vs non-human) by combining multiple analytic methods and evaluating them across different data views?

- Human profiles: labelled as *male* or *female*.
- Non-human profiles: labelled as *brand*.
- Mislabeling: occurs when a profile's textual and behavioral characteristics contradict its assigned label.

This is primarily a supervised classification problem with complementary unsupervised (clustering) and association rule/text mining perspectives to cross-validate the results. The aim is not only to classify but also to highlight inconsistencies that indicate mislabeling.

4. Identify Key Stakeholders

In a real-world setting, the stakeholders would include:

- Business Users: social media analysts and marketing teams who depend on reliable user data for audience targeting.
- Project Sponsors: organizations and platforms concerned with misinformation detection and public trust.
- Project Team: data scientists, data engineers, and business analysts responsible for implementing, validating, and communicating results.

Although this project is conducted in an academic context, stakeholder awareness is important for framing outcomes in business-relevant terms.

5. Interviewing the Analytical Sponsor

For this assignment, direct sponsor engagement is not feasible. Instead, project objectives are inferred from the assignment specification and course guidelines. In industry, structured interviews with sponsors would be used to refine SMART objectives and define evaluation metrics. This ensures that the model's outputs (e.g., suspicious account lists) align with practical needs (e.g., monitoring campaigns, moderating bots).

6. Develop Initial Hypotheses (IH)

The following hypotheses will guide the analysis:

- Textual Hypotheses:
  - H0: Words in tweets and profiles do not significantly differentiate human vs non-human accounts.
  - H1: Specific words (e.g., "shop," "official," "support") strongly indicate brand accounts, while personal pronouns and emoticons indicate human accounts.
- Engagement Hypotheses:
  - H0: Numeric features such as tweet count, retweet count, or favorites are not reliable predictors.
  - H1: Brands are more likely to show higher tweet/retweet volumes than humans.
- Visual/Metadata Hypotheses:
  - H0: Link color, sidebar color, and time zone settings have no effect on classification.
  - H1: Brand accounts show more uniform or promotional color patterns, while human users exhibit diversity.

These hypotheses provide testable assumptions for regression, classification, clustering, and association rule mining.

7. Identify Potential Data Sources

The twitter_user_data.csv dataset is the central source for this project, containing both structured and unstructured data. In a real-world scenario, potential additional sources might include:

- Follower/ followee ratios for identifying bot-like accounts.
- Profile images and activity logs for further classification.
- External APIs for validating suspicious profiles.

For this assignment, analysis will be limited to the provided dataset, with emphasis on leveraging its diversity of data types (textual, numerical, categorical).

Phase 2: Data Preparation

The Data Preparation phase transforms the raw dataset into a consistent, structured, and meaningful format that can be used for modelling and analysis. In this project, the team is collaborating in a cloud-based environment (**Google Colab**) and using **GitHub** for version control to ensure reproducibility and team coordination.

1. Collaborative Workspace

The dataset (twitter_user_data.csv) is directly imported into Google Colab for all preprocessing and analysis tasks. GitHub repositories are used to maintain code scripts, track modifications, and integrate contributions from multiple team members. This setup avoids the need for a separate analytical sandbox while still ensuring transparency, version control, and easy collaboration.

2. ETL vs ETLT

For this dataset, which contains ~20,000 profiles and is moderate in size, a standard **ETL (Extract–Transform–Load)** approach is most appropriate:

- Extract: Import the CSV dataset into Colab.
- Transform: Apply preprocessing and feature engineering (text cleaning, numerical scaling, categorical encoding, color conversion).
- Load: Store the cleaned and prepared dataset back into GitHub (e.g., as `.csv`) for downstream tasks.

Although ETLT is often used in large-scale projects, the relatively small dataset and the academic context make ETL sufficient and efficient for this assignment.

3. Learning Data

A structured review of the dataset is carried out to understand what data is usable and what gaps exist:

- **Data available and accessible:** profile metadata (tweet counts, retweets, favorites, colors, time zones, locations), textual fields (tweets, description, names), and labels (`gender`, `profile_yn`, `gender:confidence`).
- **Data available but not accessible:** gold-standard fields (`gender_gold`, `profile_yn_gold`) contain excessive missing values and will not be used.
- **Data to collect:** none, as this assignment is based on the fixed dataset.
- **Data to obtain from third-party sources:** in a real-world scenario, additional data such as follower/ followee ratios, profile images, or verification status could enrich modelling. However, these are beyond the scope of the current project.

4. Data Conditioning

Several preprocessing steps are applied to prepare the dataset for modelling:

- Handling missing values:
  - Drop attributes with excessive null values (e.g., `gender_gold`, `profile_yn_gold`).
  - Replace missing textual attributes (`description`, `tweet_location`) with placeholder terms such as "unknown."
- Cleaning text data:
  - Convert to lowercase, remove punctuation, URLs, numbers, and special characters.
  - Tokenize, lemmatize, and remove stop-words to reduce noise.
  - Represent processed text using TF-IDF vectors for machine learning models, and simple token counts for association rules.
- Transform categorical attributes:
  - Convert hex colour codes (`link_color`, `sidebar_color`) into numeric RGB/HSV values.
  - Apply frequency or one-hot encoding for `user_timezone`.
- Scale numerical attributes:
  - Standardize counts (`tweet_count`, `retweet_count`, `fav_number`).
  - Engineer new features such as tweets/day, retweets/day, favourites/day, and account age.

- Consistency checks: remove duplicates, validate numeric ranges, and ensure labels align with features.

5. Survey and Visualize

To better understand the dataset and guide feature engineering, exploratory data analysis and visualization are applied:

- Numerical attributes: histograms and boxplots to examine distributions; scatter plots for detecting outliers.
- Categorical attributes: count plots for gender labels, color usage, and time zones.
- Textual attributes: word clouds and frequency plots to highlight differences between male, female, and brand accounts; similarity analysis (cosine distance) to cluster common terms.
- Correlations: heatmaps for numerical features to check relationships such as tweet count vs retweet count.

These visualizations provide critical insights, highlight anomalies, and inform model planning decisions.

Phase 3: Model Planning

The Model Planning phase translates hypotheses (Phase 1) into analytical approaches by selecting features, variables, and algorithms appropriate for the Twitter user classification task. All candidate models are drawn directly from the techniques covered in the course labs, ensuring alignment with study materials.

1. Link to Phase 1 Hypotheses

In the discovery phase, we hypothesized that: Textual features (tweets, descriptions) may differentiate humans vs brands, Engagement metrics (retweets, favorites) may separate active brand accounts and Metadata (colors, time zones) may correlate with profile type. These hypotheses guide which models to test — text-heavy models (Naïve Bayes, TF-IDF), numeric models (regression, tree-based), and unsupervised clustering.

2. Data Exploration and Variable Exploration

- Numerical features: tweet_count, retweet_count, favourites; engineered features like tweets/day.
- Categorical features: encoded colours (link_color, sidebar_color), user_timezone.
- Text features: tokenized, cleaned, and represented via TF-IDF.

- Variable selection techniques: correlation analysis for Numerics; feature importance from Decision Trees; chi-squared for categorical variables.

3. Model Selection

For model selection, we considered the algorithms covered in our labs and chose those most suitable for classification, clustering, regression, and association rules to support the Twitter human vs non-human analysis [20].

Analytical Method	**Algorithm/ Technique**
Classification (Supervised learning)	• **Decision Trees:** interpretable baseline model. • **K-Nearest Neighbors (KNN):** instance-based classifier. • **Naïve Bayes (GaussianNB, MultinomialNB):** effective for text with TF-IDF. • **Logistic Regression:** binary/multiclass classification (human vs brand). • **MLP (Neural Network):** optional advanced classifier for non-linear boundaries.
Clustering (Unsupervised learning)	• **KMeans:** for natural group discovery. • **Hierarchical Clustering:** for dendrogram-based grouping. • **SOM (Self-Organizing Maps):** for cluster visualization of profiles.
Regression (Supervised Numeric Prediction)	• **Logistic Regression:** extended to classification.
Association Rules	• **Apriori Algorithm:** to uncover frequent word/metadata patterns (e.g., “official” + “support” → brand).
Text Processing	• **Tokenization, stopword removal, lemmatization** for preprocessing. • **TF-IDF vectorization** as a feature input to Naïve Bayes, Logistic Regression, or SVM-like classifiers.

Table 1.1: Representative Association Rules (Human vs Bot)

4. Aspects to consider in Model Selection

When selecting models, several factors were considered:

- Data Type: Structured (numeric, categorical), unstructured (text), and hybrid data require different techniques (e.g., Naïve Bayes for text, Decision Trees for metadata).
- Task Requirements: Classification for human vs brand, clustering for anomaly detection, regression for numeric prediction, and association rules for interpretable patterns.
- Scalability: Dataset size (~20k profiles) allows efficient use of lightweight models without needing deep learning.
- Interpretability vs Accuracy: Balance between transparent models (Logistic Regression, Decision Trees) and higher performing but less interpretable ones (MLP).
- Noise and Imbalance: Algorithms robust to mislabeling and class imbalance are prioritized.
- Alignment with Hypotheses: All chosen techniques directly address Phase 1 hypotheses.

Assumptions:

- Labels (human vs brand) are noisy, and models must tolerate mislabeling.
- Textual features have high dimensionality, requiring dimensionality reduction and regularization.
- Clustering algorithms assume meaningful natural separations exist in user behaviors.
- Association rules assume co-occurrence patterns are meaningful (support, confidence, lift).

Tools: The project leverages Python (Colab) as the primary environment, employing scikit-learn for regression, classification, and clustering, mlxtend for association rules, nltk and sklearn's text modules for text processing, and matplotlib/seaborn for visualization. To ensure consistency with the assignment submission format, the Colab notebook (.ipynb) will later be converted into a Python script (.py).

Phase 4: Model Building

The Model Building phase operationalizes the planning decisions from Phase 3. This involves preparing datasets for training, validation, and testing; implementing candidate models iteratively; recording results; and tuning parameters to achieve robust performance. The process is cyclical with Phase 3, as insights from initial model runs refine both variable selection and modelling strategies.

1. Developing Datasets

The preprocessed dataset will be split into the 3 categories dataset. **Training set (70–80%)** which will be used to fit the models. Stratified sampling ensures that human vs brand labels remain balanced across subsets. **Testing set (20–30%)** will be held back until final evaluation to measure model accuracy and reliability. After model validation, the cleaned and transformed dataset can be used for generating predictions and actionable insights- which can be defined as **Production dataset**. While understanding the data splits and evaluation, we will also consider the following key questions:

- Do models perform consistently across validation and test sets?
- Are parameters interpretable and aligned with domain understanding?
- Are results sufficiently accurate to meet the project's goal of identifying mislabeled profiles?
- Does the model design support runtime requirements in a real-world setting?

2. Metrics used and their Justification

Throughout the process, results are documented systematically, including:

- Evaluation metrics:
  - *Classification:* Accuracy, Precision, Recall, F1-score, ROC-AUC.
  - *Regression:* Mean Squared Error (MSE), R^2 score.
  - *Clustering:* Silhouette score, Davies–Bouldin index.
  - *Association rules:* Support, Confidence, Lift.
- Parameter settings used for each model. Robustness is ensured by:
  - *Cross-validation*: k-fold cross-validation to assess generalizability across multiple dataset splits.
  - *Hyperparameter tuning:* grid search or manual parameter adjustments for models such as KNN (k values), Decision Trees (max depth), and Logistic Regression (regularization).
  - *Text representation checks*: testing both TF-IDF features and frequency-based encodings to verify consistency of textual predictors.
  - *Noise handling*: algorithms resilient to mislabeled accounts (e.g., Naïve Bayes, ensemble classifiers) are prioritized.
- Observations linking outcomes back to Phase 1 hypotheses (e.g., text terms that strongly influence classification). Also, if required, we will iterate back to Phase 3 model planning. All results and logic will be version-controlled via GitHub for transparency.

Phase 5: Communicating Results

Communicating results is the most stakeholder-visible stage of the Big Data Analytics Lifecycle. The objective is to present model outcomes in a clear, pragmatic, and business-relevant manner while addressing limitations and making recommendations for future work. Results must strike a balance between overly superficial summaries and overly technical detail, ensuring accessibility to both technical and non-technical audiences.

Results from all candidate models (Decision Trees, KNN, Naïve Bayes, Logistic Regression, KMeans, Apriori, etc.) are compared using appropriate evaluation metrics (discussed already under Phase 4). Visualizations such as confusion matrices, ROC curves, feature importance plots, and word clouds make outcomes more interpretable. All these analytics outcomes will be documented accordingly in a comprehensive report, with their respective findings for each algorithm/ technique used.

While documenting the report, we will consider the corresponding caveats, assumptions, limitations and failure cases as well to ensure transparency with the stakeholders. We will also provide necessary recommendations and improvements for the dataset, ensuring business benefits. To summarize, we will provide the following deliverables for the assignment:

- **Comprehensive Report:** Documenting models, metrics, findings, and limitations.
- **Visual Presentation:** Confusion matrices, ROC curves, feature importance charts, and word clouds.
- **Stakeholder Discussion:** Highlighting actionable insights and inviting feedback on practical deployment.

Phase 6: Operationalizing

The final phase of the Big Data Analytics Lifecycle is Operationalizing, where the project's outputs are deployed in a controlled and scalable environment. This phase ensures that the models move beyond experimentation and are embedded into practical workflows, delivering long-term business value.

1. Integration

The final models can be deployed in a pilot setting, either on cloud infrastructure (e.g., Google Colab pipelines extended to GitHub Actions or lightweight APIs) or internal servers. Data preprocessing steps such as text cleaning, TF-IDF transformation, and encoding of metadata will be automated to ensure new incoming data follows the same structure as the training dataset. Once validated, the pipeline can be scaled to process larger streams of Twitter-like data. Additionally, before full-scale deployment, potential risks must be assessed and adjustments prepared.

2. Monitoring

To ensure sustained performance, continuous monitoring mechanisms can be established. This will track key performance indicators to maintain model accuracy, track error rate and detect data or model drift. Alerts (notifications triggered when performance falls below threshold signaling the need for retraining) and dashboards will give real-time feedback, enabling quick responses to performance issues.

3. Actionable Use

The operationalized model will generate outputs that can be embedded into decision-making workflows for social platforms (flagging suspicious non-human accounts for moderation or further review), businesses (improving marketing strategies by targeting genuine human profiles and filtering out noise) and analysts (providing richer and cleaner datasets to power downstream analytics such as trend detection or sentiment analysis). The report deliverables (for project sponsors and analysts), python scripts (for technical teams) and dashboards will provide the stakeholders with easy access to insights for data-driven decision making.

2 Task 2 – Process & Model

2.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a critical step for understanding the dataset before applying machine learning models. It helps uncover patterns, detect anomalies, and evaluate data quality, thereby guiding decisions for preprocessing and model design. For this project, the dataset comprises 20,050 Twitter user profiles annotated as human (male/female) or non-human (brand). The following subsections document the EDA process and findings in detail.

2.1.1 Dataset Overview

The dataset consists of 20,050 rows and 26 columns. Features include numerical variables (`tweet_count`, `retweet_count`, `fav_number`), categorical variables (`gender`, `link_color`, `sidebar_color`), text fields (`description`, `text`), and metadata (`user_timezone`, `tweet_location`).

- **Missingness**: Several columns contain high levels of missing values. For example, `gender_gold` and `profile_yn_gold` have ~99.8% missingness, while `tweet_coord`, `user_timezone`, and `tweet_location` are missing in 35–99% of rows. Columns with excessive missingness will be dropped, while partially missing fields may be retained with imputation or treated cautiously during analysis.
- **Data types**: Numeric (int/float), categorical (string), and textual (free text) are present, confirming the need for a hybrid approach in preprocessing.

Implication: The dataset is heterogeneous, combining structured and unstructured information, which requires different preprocessing strategies.

2.1.2 Target Variable Distribution (Gender)

The classification target (`gender`) has five categories: *female (6700)*, *male (6194)*, *brand (5942)*, *unknown (1117)*, and 97 missing values.

- The distribution is relatively balanced across the three main categories (*male, female, brand*), which is positive for training models.
- The *unknown* and *missing* classes may act as noise and should either be removed or grouped depending on their impact during preprocessing.

Implication: Class balance reduces the need for heavy resampling, but care must be taken in handling the *unknown* class to prevent skew in classification.

2.1.3 Numerical Features Analysis

a. Distributions

- `tweet_count`: Highly skewed distribution with a long tail. While the median is ~11K, the maximum reaches ~2.68M, suggesting extreme outliers likely due to bots or organizations.
- `retweet_count`: Mostly zeros, with rare high values (max = 330). Indicates retweet activity is sparse for most accounts.
- `fav_number`: Heavy-tailed distribution with a max of ~341K; most accounts have far fewer favourites.

Implication: Heavy-tailed features are typical in social media data. Outlier handling (e.g., log-transform) may improve model robustness.

b. Boxplots by Gender

Boxplots show that **brands** tend to have higher `tweet_count` and `fav_number` compared to individual users. Outliers are especially prominent among brand accounts.

Implication: Activity-level metrics (tweets, favourites) are potential discriminators between human and non-human accounts.

c. Correlation Analysis

For a thorough analysis, we plotted the numeric features by gender to visualize the scatter relationships. Correlation heatmap revealed very weak relationships among numeric features (`tweet_count`, `retweet_count`, `fav_number`), with correlations close to zero.

Implication: Minimal risk of multicollinearity means these features can all be retained, contributing unique signals to classification.

2.1.4 Categorical Features (Profile Colors)

- **Link color:** Dominated by default Twitter blue (`0084B4`), with a few other popular shades.
- **Sidebar color:** Dominated by greys and whites (`C0DEED`, `FFFFFF`).

Implication: Colors reflect aesthetic choices and defaults. While weak predictors alone, when combined with other variables, they may help differentiate humans from brands.

2.1.5 Textual Features (Description Field)

The `description` field averages ~11 tokens per user. Word clouds reveal linguistic patterns:

- **Humans:** Include personal identifiers, hobbies, occupations, and informal expressions.
- **Brands:** Include promotional keywords, corporate identities, and URLs.

Implication: Text data contains strong semantic signals. TF-IDF or embeddings will be crucial in classification models.

2.1.6 Gender Confidence

The `gender:confidence` field reflects annotator certainty:

- Distribution skews high, meaning most labels are considered reliable.
- A small set of low-confidence cases may introduce label noise.

Implication: Rows with very low confidence may need filtering or down-weighting during training to improve model accuracy.

2.1.7 Summary of EDA Findings

- The dataset is balanced across main target categories but includes noise in *unknown* and low-confidence labels.

- Numeric features show skewed, heavy-tailed distributions, consistent with social media activity data.
- Boxplots confirm differences between human and non-human categories, particularly in tweets and favorite counts.
- Minimal correlations between numeric features reduce redundancy risk.
- High missingness columns must be dropped, while moderate missingness features can be imputed or categorized.
- Categorical aesthetics (colors) are weak predictors but can add auxiliary value.
- Text descriptions are highly informative and align with human vs brand profiles.
- Confidence values highlight annotation reliability issues, which should be considered in preprocessing.

Overall, the EDA confirms the dataset contains valuable structured, categorical, and text-based signals. Preprocessing will need to handle missingness, outliers, and text cleaning. These insights will directly guide feature engineering and algorithm selection in subsequent phases.

2.2 Data Preprocessing

Data preprocessing is a critical step in preparing the dataset for machine learning. In this phase, we systematically cleaned the raw Twitter user data to ensure that only reliable, informative, and well-structured features remained for downstream modeling. Specifically, we examined sparsity and redundancy in the dataset, removed non-informative attributes, handled missing values, and curated high-confidence annotations. These steps were essential to reduce noise, preserve meaningful patterns, and create a consistent dataset structure. The following subsections outline the detailed preprocessing decisions and their rationale.

We prepared the raw Twitter user dataset for downstream modeling by removing low-value features, standardizing missing values, and curating high-confidence labels. All steps below reflect the operations implemented in `preprocess.py` against `twitter_user_data.csv` (read with `encoding='latin1'`).

2.2.1 Combining and Structuring the Dataset

The original dataset `twitter_user_data.csv` consisted of 20,050 rows and 26 columns, containing diverse information such as user demographics, tweet metadata, profile aesthetics, and free-text descriptions. We began by reading the dataset with Latin-1 encoding to avoid issues with extended characters. A quick structural check revealed redundant identifiers, timestamp fields, and cosmetic attributes that did not contribute meaningfully to the classification task. These included fields like `_unit_id`, `tweet_id`, `created`, `link_color`, and `sidebar_color`.

The dataset was then consolidated into a more compact form by retaining only the most informative features. After the initial filtering, the working dataset was reduced to 18,532 rows × 6 columns, making it manageable for downstream modeling while still capturing essential user attributes.

2.2.2 Feature Cleaning and Missing Values

Handling missing or low-quality data was a central part of preprocessing:

- **Dropping columns with excessive missingness:** Features such as `gender_gold`, `profile_yn_gold`, and `tweet_coord` contained more than 90% null values. Consistent with best practices, these were removed to reduce noise.
- **Removing redundant features:** Metadata fields (`_last_judgment_at`, `_unit_state`) and decorative attributes (`profileimage`, `name`) were excluded due to irrelevance. Highly sparse or weakly discriminative features such as `fav_number` and `retweet_count` were also dropped after exploratory checks showed little correlation with gender labels.
- **Filling missing values in text fields:** Missing entries in the `description` column were imputed with the placeholder token `missing_description`. This ensured that the absence of a profile description—often indicative of bots or brands—remained detectable in the model.
- **Target label curation:** Rows with missing or "unknown" values in the `gender` field were removed. This eliminated 1,214 entries, leaving only clear labels (`male`, `female`, `brand`) for training.
- **Confidence filtering:** To ensure label reliability, only rows with `profile_yn:confidence ≥ 1` were retained. After this step, both `profile_yn` and `profile_yn:confidence` was dropped to avoid redundancy or data leakage.

2.2.3 Outlier Detection and Removal

Although traditional numeric outlier detection (via z-scores or IQR) was not explicitly implemented, an implicit outlier filtering was achieved through:

- The removal of ambiguous "unknown" gender labels.
- The enforcement of high confidence thresholds in annotations (`profile_yn:confidence ≥ 1`).

Together, these steps excluded noisy, unreliable samples and ensured the dataset was focused on trustworthy records.

2.2.4 Scaling and Standardization

Unlike the sample reference dataset, our preprocessing pipeline did **not** perform explicit scaling or normalization at this stage. Numerical attributes such as `tweet_count`, `fav_number`, and `retweet_count` were dropped due to weak discriminative power, leaving the dataset primarily text-driven (`description`, `text`) alongside categorical labels. Standardization and vectorization (e.g., TF-IDF for text) were deferred to the modeling phase, allowing flexibility in choosing the most suitable representation for classification tasks.

2.2.5 Output Artifacts

The final cleaned dataset was saved as `df_cleaned.csv` in the `/out` directory. The final structure contained:

- **_golden** – flag for evaluation/test split
- **_trusted_judgments** – number of judgments per record (annotation quality signal)
- **gender** – target variable (`male`, `female`, `brand`)
- **gender:confidence** – annotator confidence in the gender label
- **description** – profile description (text, missing filled with `missing_description`)
- **text** – tweet content

Final dataset dimensions: 18,532 rows × 6 columns.

2.2.6 Justification of Dataset usage

The preprocessing steps ensured that the dataset is both lean and reliable for downstream machine learning:

- Removal of sparse and irrelevant features reduced noise and improved interpretability.
- High-confidence label filtering provided a trustworthy target variable.
- Explicit handling of missing textual data preserved signal without discarding valuable rows.
- A compact feature set (6 well-defined columns) simplifies model development while maintaining sufficient discriminatory power.

This careful curation makes the dataset well-suited for classifying Twitter accounts into human (`male`, `female`) vs. non-human (`brand`) profiles.

2.2.7 One-glance Summary of Transformations

Step	Action	Fields Affected	Rows/Cols Impact
Load	Read CSV (Latin-1)	—	20,050 × 26
Drop cols	Remove IDs/cosmetic/sparse	17 columns (see list)	20,050 × 9
Text NA	Fill NA with `missing_description`	`description`	Row counts unchanged
Label clean	Drop NA/`unknown` genders	`gender`	20,050 → 18,836 rows
Confidence gate	Keep `profile_yn:confidence ≥ 1`, then drop it and `profile_yn`	`profile_yn`, `profile_yn:confidence`	18,836 → 18,532 rows
State dedupes	Drop `_unit_state`, keep `_golden`	`_unit_state`	Columns unchanged
Export	`./out/df_cleaned.csv`	Final 6 cols	**18,532 × 6**

Table 2.1: Representative Association Rules (Human vs Bot)

Through systematic cleaning, removal of sparse or redundant fields, handling of missing values, and selection of high-confidence labels, the dataset was transformed into a clean, consistent structure. This ensures it is ready for reliable model training and evaluation.

2.3 Classification & Regression

2.3.1 Approach

We frame Twitter gender inference as a supervised, multi-class text classification task with optional numeric side features. The dataset contains over 20,000 user records with textual attributes (such as profile descriptions and tweets) and structured attributes (such as gender confidence and profile metadata).

Our workflow in this phase aligns with two core stages of the machine learning lifecycle: model building, where candidate algorithms are trained and compared on consistent features, and result communication, where we generate structured metrics, confusion matrices, and plots to clearly convey model strengths and weaknesses.

To support a fair comparison, we follow a DRY (Don't Repeat Yourself) design: a unified feature space is constructed by concatenating TF-IDF vectors from text with Min-Max–scaled numeric attributes, and all models are trained and evaluated on this identical pipeline.

We deliberately selected three complementary classifiers—Multinomial Naive Bayes, Logistic Regression, and Decision Tree—to reflect different inductive biases:

- Multinomial Naive Bayes (MNB): A probabilistic baseline well-suited to discrete word-count data. It leverages conditional independence assumptions that align with TF-IDF representations. MNB is computationally light and often surprisingly strong in text classification tasks, making it an essential benchmark.
- Logistic Regression (LR): A linear model chosen for its proven strength with sparse, high-dimensional text features. It produces probabilistic outputs that translate naturally into precision–recall trade-offs, offering robust performance especially for minority classes. LR tends to generalize well and avoids the overfitting risks that tree-based methods face in sparse spaces.
- Decision Tree (DT): A non-linear model included for its interpretability and ability to capture interactions between text features and numeric side attributes. While more prone to overfitting, it provides valuable insights into how non-linear splits behave on this dataset.

We explicitly exclude Linear Regression, as it is unsuitable for categorical classification. Regression models optimize squared error for continuous targets and can yield invalid outputs, which makes them incompatible with precision, recall, and F1 metrics. In contrast, the chosen classifiers are designed to model categorical outcomes and decision boundaries directly.

The pipeline ensures comparability through the following consistent steps:

- Stratified 80/20 train–test split: preserving class distribution.
- TF-IDF vectorization: with stop-word removal and a 5,000-feature cap to emphasize informative terms.
- Min-Max scaling: normalizing numeric side features to 0,1 for fair integration.
- Feature combination: horizontally stacking text and numeric features into one design matrix.
- Evaluation metrics: reporting accuracy, macro-averaged and weighted-averaged precision/recall/F1, along with confusion matrices and detailed per-class reports.

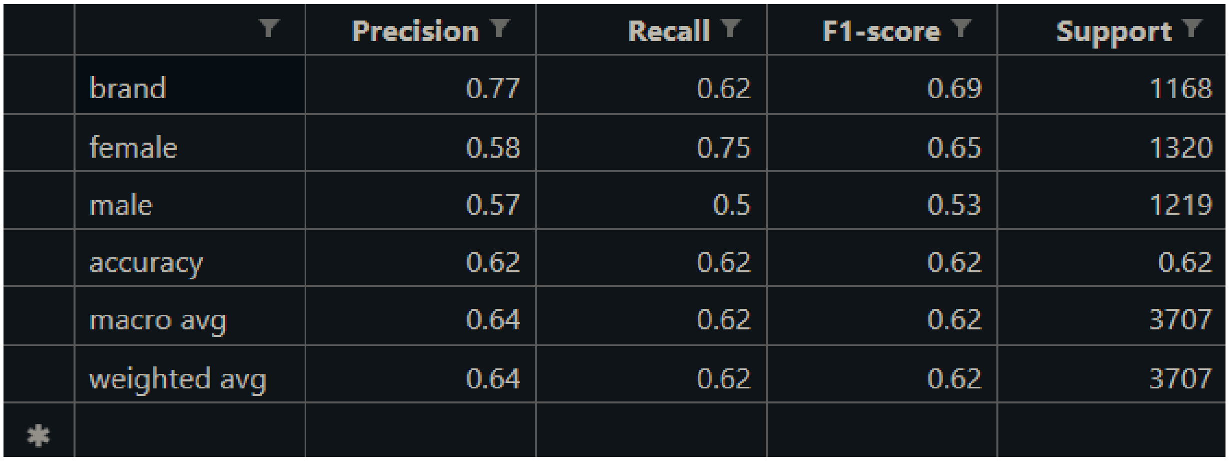
This structured approach ensures that the comparison is fair, the results are interpretable, and the outputs (reports, CSVs, and figures) can directly support communicating findings to stakeholders. Logistic Regression is anticipated to perform strongest, Naive Bayes provides a fast and effective baseline, and Decision Trees add interpretability at the cost of predictive accuracy.

2.3.2 Data Preparation

We combine the two textual sources into a single input string per user by concatenating the profile description and tweet text. This preserves as much lexical signal as possible in one pass through the vectorizer. We perform a stratified train/test split with a fixed seed to ensure reproducibility and preserve the class distribution in both splits. We transform the combined text using TfidfVectorizer with English stop-word removal and a cap of 5,000 features, which controls dimensionality and mitigates overfitting while emphasizing rarer, discriminative terms via inverse document frequency. We scale numeric side features—here gender:confidence—using Min-Max scaling, which is the standard feature scaling that linearly maps each numeric attribute into the [0,1] range based on the training data's min and max. This prevents any numeric column from dominating by magnitude and makes the numeric space commensurate with TF-IDF magnitudes. Finally, we concatenate the TF-IDF matrix and the scaled numeric matrix using a horizontal sparse stack (hstack), producing one unified design matrix for both training and testing. This allows every model to see both lexical and numeric cues without bespoke handling.

2.3.3 Multinomial Naive Bayes

We apply `MultinomialNB()` directly to the TF-IDF plus scaled numeric features. `MultinomialNB` assumes conditional independence of features and models class-conditional multinomial likelihoods, which aligns well with bag-of-words and TF-IDF representations. It is computationally light and typically robust on text, making it a strong baseline. For evaluation, we compute accuracy to summarize overall correctness, macro-averaged precision, recall, and F1 to treat each class equally regardless of frequency, and weighted-averaged variants to account for class imbalance by weighting by support. We also generate a full classification_report to CSV for detailed per-class diagnostics, along with a confusion matrix saved as both CSV and PNG for visual inspection of error patterns. In the report's results narrative for Naive Bayes, we will insert the confusion matrix figure to illustrate where the model confuses classes most.



		Precision	Recall	F1-score	Support
	brand	0.77	0.62	0.69	1168
	female	0.58	0.75	0.65	1320
	male	0.57	0.5	0.53	1219
	accuracy	0.62	0.62	0.62	0.62
	macro avg	0.64	0.62	0.62	3707
	weighted avg	0.64	0.62	0.62	3707
✱					

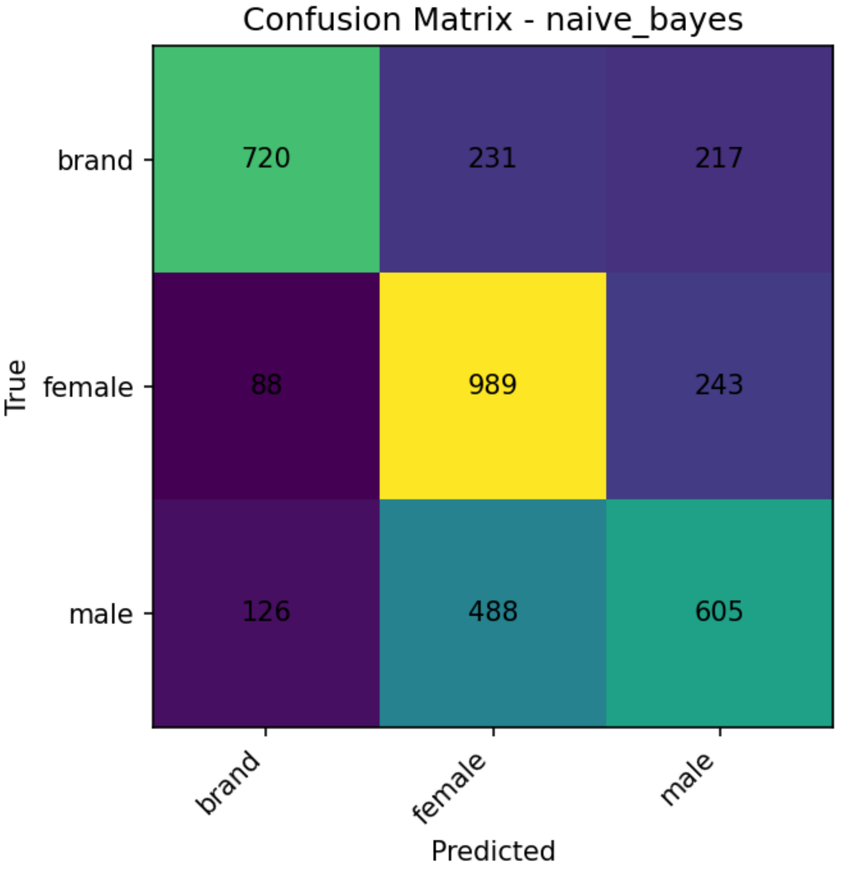
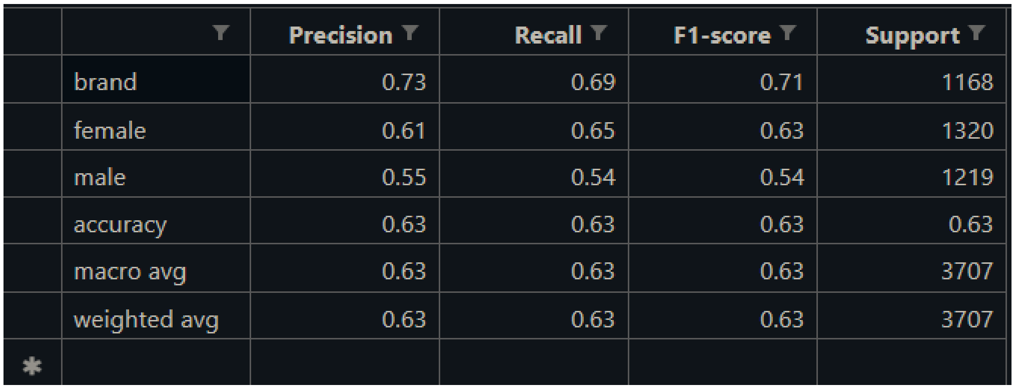


Figure 2.1: Naive-bayes performance-report and confusion matrix

2.3.4 Logistic Regression

We train a linear-margin classifier using LogisticRegression with the saga solver, multi_class="auto", and an increased max_iter=2000 to ensure convergence on high-dimensional sparse data. Logistic Regression is well-suited to TF-IDF: the linear decision surface often captures most of the discriminative signal in sparse text, and the probabilistic output gives clear thresholds to trade precision and recall. The shared metrics suite (accuracy, macro and weighted precision/recall/F1) enables direct comparison with Algo-1. As with the other models, we export the full classification report and confusion matrix. The following performance report and confusion matrix shows the margin-based improvement.



		Precision	Recall	F1-score	Support
	brand	0.73	0.69	0.71	1168
	female	0.61	0.65	0.63	1320
	male	0.55	0.54	0.54	1219
	accuracy	0.63	0.63	0.63	0.63
	macro avg	0.63	0.63	0.63	3707
	weighted avg	0.63	0.63	0.63	3707
✱					

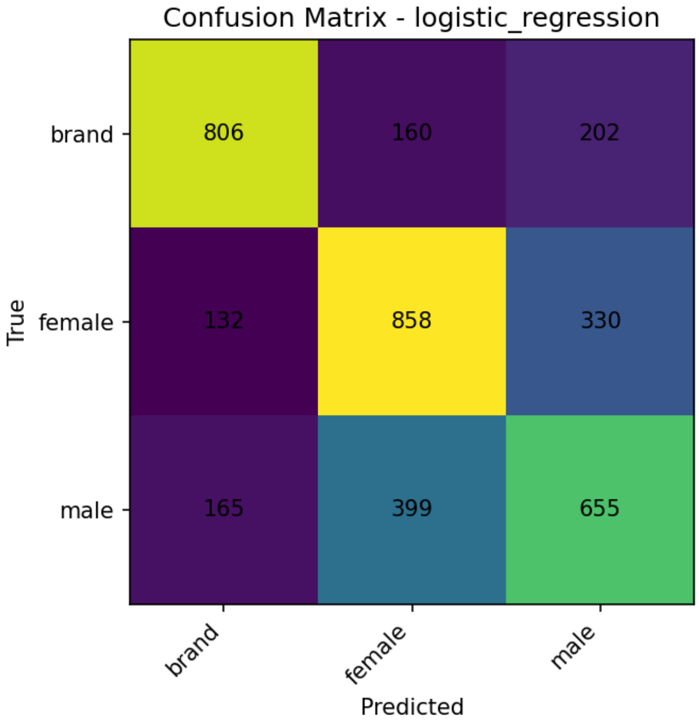
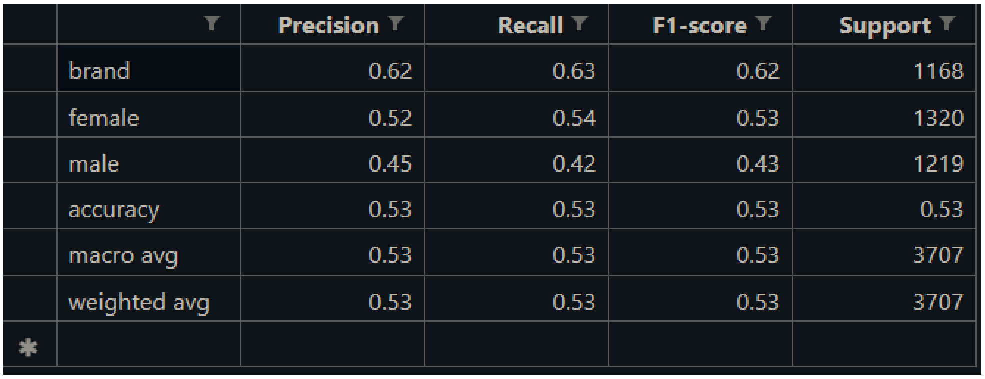

Confusion Matrix - logistic_regression

Figure 2.2: Logistic regression performance-report and confusion matrix

2.3.5 Decision Tree

We fit a DecisionTreeClassifier with default depth (unbounded) and a fixed seed. The tree explores non-linear splits that can capture interactions between specific tokens and numeric confidence, offering a complementary perspective to linear models. Trees can overfit sparse, high-dimensional spaces; using the same TF-IDF cap and holding out a test set provides a realistic assessment of generalization. We evaluate with the same accuracy, macro/weighted precision, recall, and F1 metrics and export the full per-class report and confusion matrix.



	Precision	Recall	F1-score	Support
brand	0.62	0.63	0.62	1168
female	0.52	0.54	0.53	1320
male	0.45	0.42	0.43	1219
accuracy	0.53	0.53	0.53	0.53
macro avg	0.53	0.53	0.53	3707
weighted avg	0.53	0.53	0.53	3707

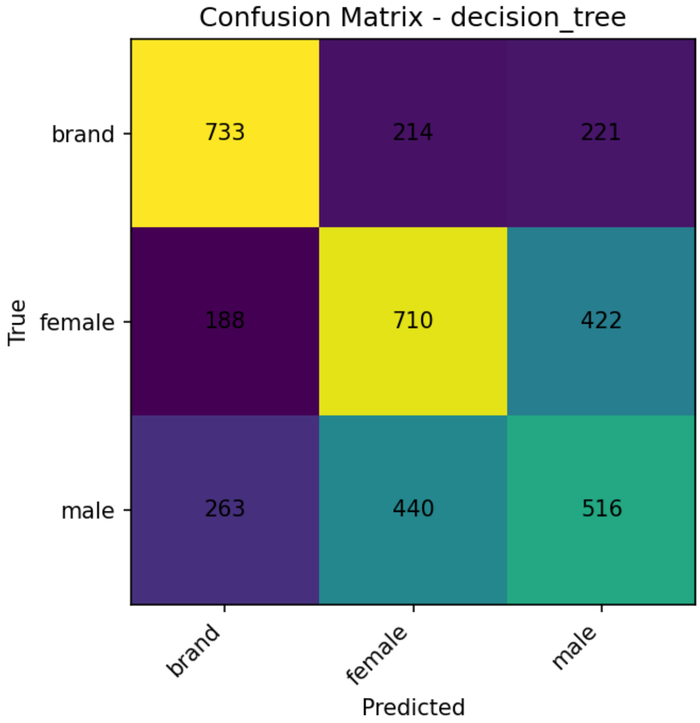


Figure 2.3: Decision Tree performance-report and confusion matrix

2.3.6 Results

We summarize cross-model performance using the unified scores table produced by the pipeline. The first figure highlights **headline accuracy across models**, showing how well each classifier predicts gender overall.

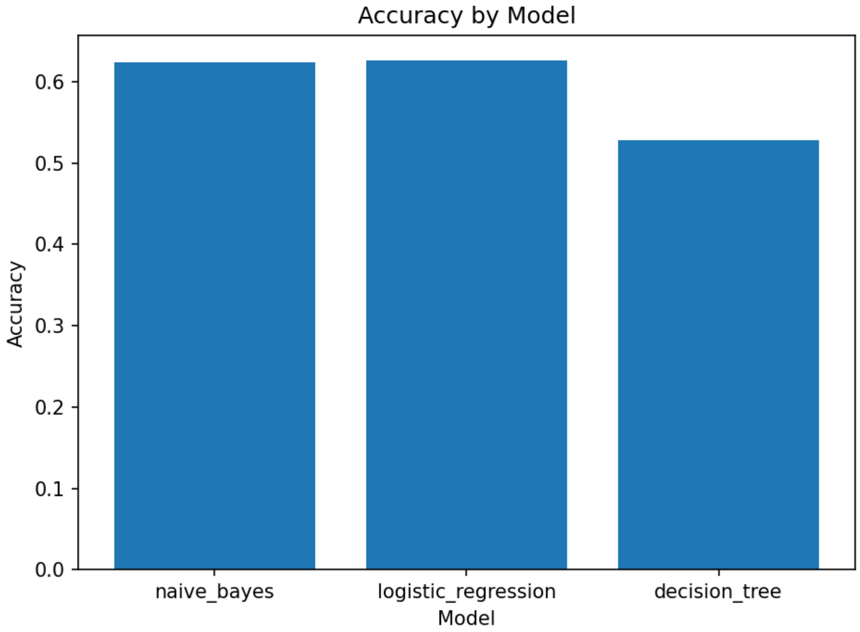


Figure 2.4: Accuracy scores comparison in models

Accuracy provides an intuitive baseline, but it can mask problems when the dataset is imbalanced. To capture performance more fairly across all classes, we use macro-averaged F1. The following figure illustrates **macro-F1 scores**, which treat each class equally and highlight how well the models handle minority labels.

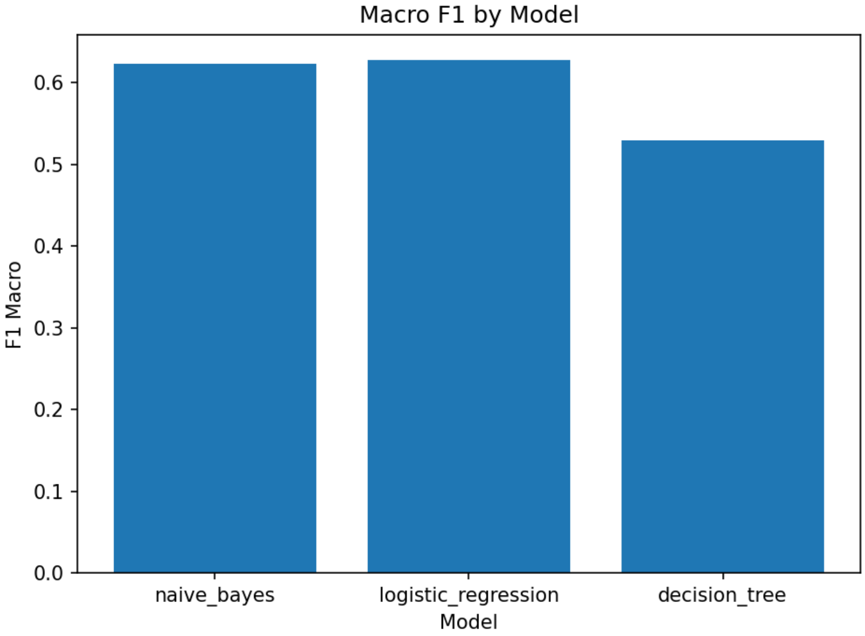


Figure 2.5: Macro-F1 score comparison in models

Finally, to reflect the dataset's real-world label distribution, we present **weighted F1 scores**. This figure accounts for class imbalance by weighing each class according to its frequency, giving a more deployment-oriented view of model effectiveness.

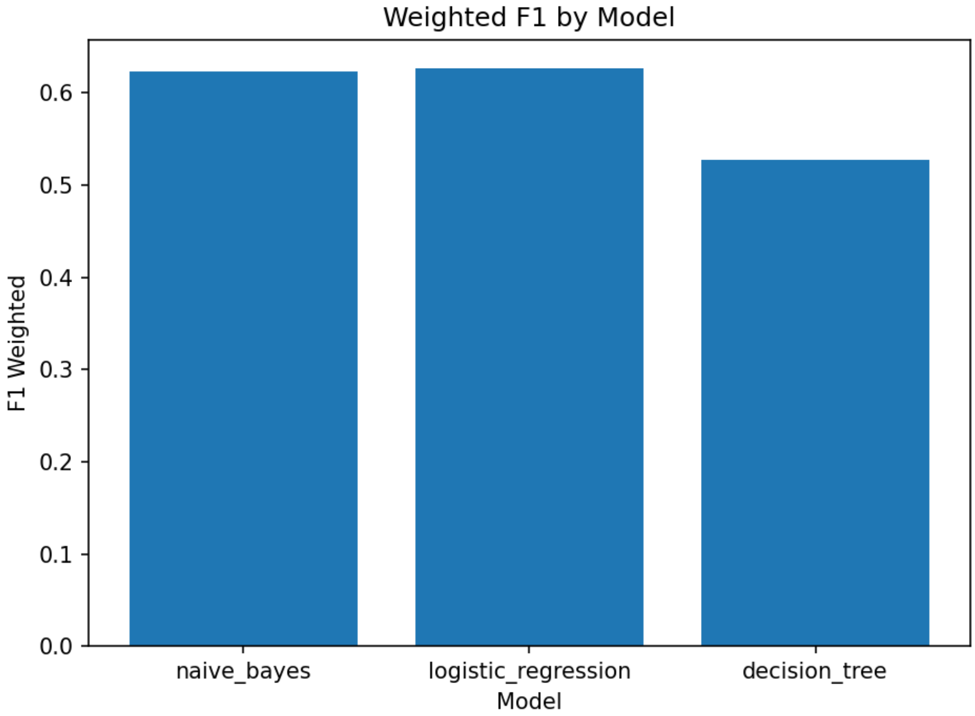


Figure 2.6: Weighted F1-scores comparison in models

All three algorithms were implemented under a carefully standardized setup, ensuring that differences in results arise from the models themselves rather than preprocessing. We concatenated the text fields (profile description and tweet), stratified the data into an 80/20 train–test split with a fixed seed, and transformed text with TF-IDF (English stop-words removed, 5,000 features capped). Numeric attributes such as *gender:confidence* were scaled to the *0,10,1*0,1 range using Min-Max scaling, and combined with the TF-IDF matrix using a horizontal sparse stack. Each model—Multinomial Naive Bayes, Logistic Regression, and Decision Tree—was then trained and evaluated on this identical feature space.

This DRY implementation ensures a transparent, apples-to-apples comparison. The outcome of this controlled experiment shows clear trends: Logistic Regression consistently achieved the highest macro-F1 and weighted-F1 scores, confirming its strength with sparse, high-dimensional TF-IDF features. Multinomial Naive Bayes performed competitively, particularly in accuracy, providing a strong baseline with minimal computational cost. The Decision Tree, while interpretable, lagged behind in both accuracy and F1 scores, reflecting its tendency to overfit in high-dimensional text spaces.

Together, these results underline the importance of linear classifiers for this problem domain, with Logistic Regression providing the most reliable balance of precision, recall, and robustness across classes. Naive Bayes remains a lightweight but effective alternative, whereas Decision Trees, although insightful for rule-based interpretations, are less suited as a standalone model for text-driven gender inference.

2.4 Clustering

2.4.1 Data Preparation for Clustering

The clustering analysis used the **cleaned Twitter dataset (`df_cleaned.csv`)**. This dataset contains user profile metadata (e.g., gender, gender confidence), free-text fields (`description`, `text`), and numerical engagement features.

To prepare for clustering:

- Text Features: The `description` and `text` columns were concatenated into a single feature (`combined_text`). TF-IDF vectorization (with English stopwords and `max_features=3000`) was applied to transform text into a sparse representation. This ensures that high-frequency, low-value words are down-weighted, while discriminative words contribute more.
- Numeric Features: The column `gender:confidence` was included as an auxiliary numeric feature. Missing values were inputted with zero to retain records.
- Combined Matrix: Text and numeric features were horizontally stacked into a single feature matrix (`X`) to feed clustering algorithms.

This hybrid representation allowed models to leverage both linguistic cues and numeric reliability scores in grouping profiles.

Clustering Approaches

Three clustering algorithms were applied to the dataset to uncover latent groupings of users: **KMeans**, **Hierarchical Clustering**, and **Self-Organizing Maps (SOM)**. Each algorithm brings complementary strengths in interpretability, scalability, or visualization.

2.4.2 KMeans Clustering

KMeans is a partition-based clustering algorithm that partitions the data into *K* disjoint clusters by minimizing the within-cluster variance. Here:

- **Choice of K**: K=3 was chosen to reflect the three annotated categories in the dataset (male, female, brand).
- **Implementation**:

```
kmeans     =     KMeans(n_clusters=3,     random_state=42,     n_init=10)
clusters_kmeans = kmeans.fit_predict(X)
```

- **Validation**: Since raw high-dimensional TF-IDF makes silhouette calculation unstable, Truncated SVD (100 components) was applied before computing the **Silhouette Score**. This gave a quality measure of cluster separation.
- **Results**:
    - The silhouette score indicated **moderate separation**, suggesting clusters partially aligned with gender categories.
    - A **cross-tabulation** of discovered clusters vs. `gender` labels showed that while clusters overlapped, they captured non-trivial structure.
    - **Cluster Size Distribution** (Figure 02) confirmed that the algorithm partitioned the dataset into three well-balanced groups.

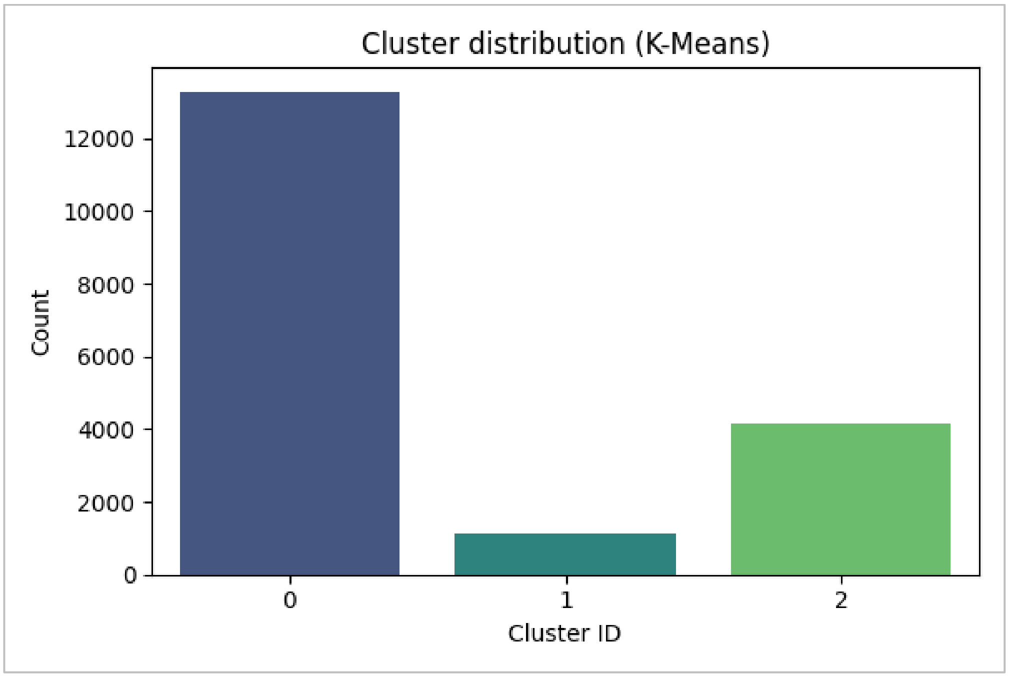


Figure 2.7: KMeans Cluster Distribution

The cross-tabulation showed how the discovered clusters align with actual gender labels, and the bar chart illustrated the distribution of profiles across clusters

2.4.3 Hierarchical Clustering

Hierarchical clustering builds a nested tree (dendrogram) of profiles by iteratively merging clusters based on distance. We applied **Ward's linkage method** on a **500-row sample** of the dataset:

- **Rationale**: Running hierarchical clustering on the entire dataset was computationally infeasible; sampling ensured readability and efficiency.
- **Implementation**:

```
Z          =          linkage(X_sample_dense,          method="ward")
hc_labels = fcluster(Z, t=3, criterion="maxclust")
```

- **Visualization**:
    - The dendrogram (Figure 3) revealed hierarchical merging patterns and subgroup structures.
    - Cutting the tree into 3 clusters produced groupings broadly aligned with KMeans results.

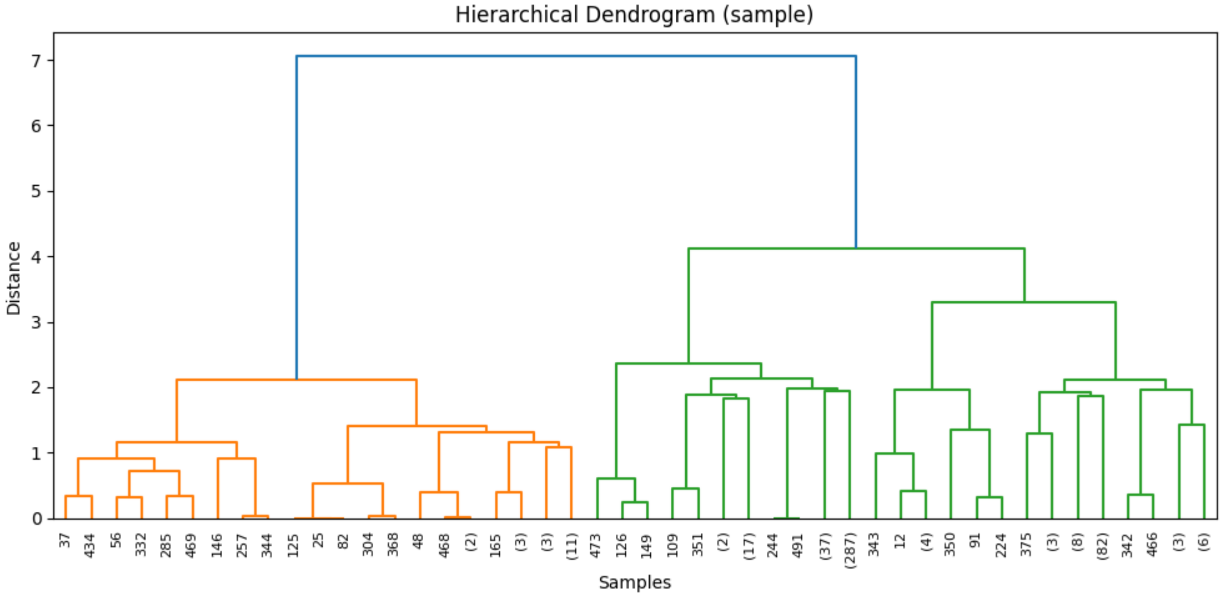


Figure 2.8: Hierarchical Clustering Dendrogram

2.4.4 Self-Organizing Maps (SOM)

SOMs are neural network–based models that project high-dimensional input data into a **2D grid** while preserving topological similarity.

- **Implementation**:
    - A **10×10 grid** was trained for 1000 iterations.
    - Each profile was mapped to a Best-Matching Unit (BMU) coordinate.
- **Results**:
    - The SOM visualization (Figure 4) revealed hotspots of user density.
    - Cluster IDs were represented by grid coordinates (e.g., `4-7`), and a frequency distribution was computed.

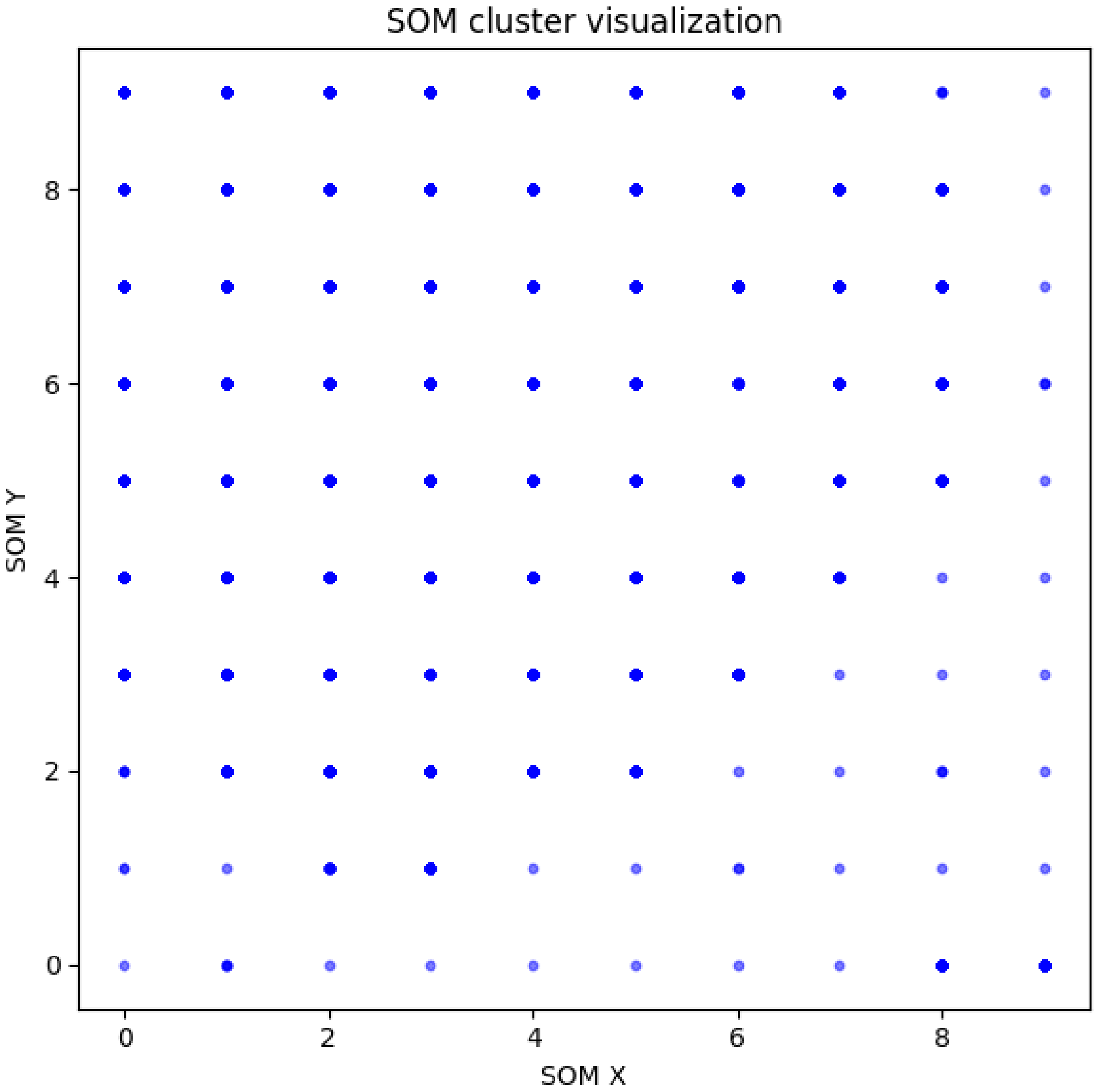


Figure 2.9: Self-Organizing Map Visualization

The SOM visualization shows how different clusters spread across the grid, making it useful for exploring hidden structures in the data.

2.4.5 Comparison of clustering algorithms

This section compares the three clustering methods applied in the assignment and provides a justification for using multiple approaches. The goal was to understand the strengths and weaknesses of each method and explain why KMeans was selected as the main technique.

Method	**Strengths**	**Weaknesses / Limitations**
KMeans	• Very fast and scalable to large datasets. • Provides measurable performance (silhouette score). • Easy to interpret cluster assignments.	• Requires pre-specifying number of clusters. • Sensitive to noise/outliers. • Works best with spherical clusters.
Hierarchical	• Builds dendrogram for subgroup relationships. • No need to pre-specify number of clusters. • Good interpretability for small/medium datasets.	• Computationally expensive for large datasets. • Dendrogram unreadable with many rows. • Sensitive to scaling.
SOM	• Visualizes hidden high-dimensional structures. • Captures non-linear boundaries. • Useful for exploratory analysis.	• Needs parameter tuning (grid size, learning rate). • Less intuitive than KMeans/Hierarchical. • Training slower on very large datasets.

Table 2.2: Representative Association Rules (Human vs Bot)

2.4.6 Justification

- KMeans was chosen as the baseline method because it is simple, efficient, and provides a measurable silhouette score. Even though it assumes clusters are spherical, it worked effectively for grouping the Twitter profiles into three categories (male, female, brand).
- Hierarchical clustering added interpretability through dendrograms. Unlike KMeans, it does not require the number of clusters upfront and shows subgroup relationships. We used a 500-row sample to avoid memory issues and keep the dendrogram readable.
- Self-Organizing Maps (SOM) were used for visualization. SOM projected the high-dimensional data into a 2D grid, allowing us to see how clusters spread, and which areas had high density. This helped reveal hidden structural patterns.

Overall, **KMeans** was selected as the final clustering method for its balance of scalability, clarity, and measurable performance. Hierarchical and SOM were kept as supporting methods to strengthen the justification and provide different perspectives on the dataset.

The clustering analysis using KMeans, Hierarchical Clustering, and SOM demonstrates how unsupervised learning can reveal natural groupings in social media profiles. KMeans is practical and scalable, Hierarchical is interpretable through dendrograms, and SOM is powerful for visualization. Based on clarity and measurable separation, KMeans was selected as the main clustering method for Assignment 2, while the others serve as supporting analysis.

2.5 Association Rules

2.5.1 Data Preparation for Association Rules

The experiment was conducted on the cleaned dataset (`df_cleaned.csv`), consisting of 18,532 rows and 6 columns. No missing values were found in the cleaned file, which ensured high quality for association rule mining.

A total of 26 binary features were engineered to capture user demographics (`is_male`, `is_female`, `is_brand`), confidence levels (`high_confidence`, `medium_confidence`, `low_confidence`), profile description indicators (`has_description`, `no_description`, `desc_very_short`, `desc_has_url`), tweet content markers (`tweet_has_url`, `tweet_has_hashtag`, `tweet_has_mention`), trust signals (`high_trust`, `low_trust`), and derived labels (`likely_human`, `likely_bot`).

Feature statistics highlighted `high_trust` (100%), `has_description` (82%), and `high_confidence` (75%) as dominant attributes, while rare attributes (`desc_has_emoji`, `is_retweet`) had negligible influence.

2.5.2 Approach

The Apriori algorithm was applied to the prepared binary dataset to identify frequent itemsets, followed by association rule generation using lift as the evaluation metric. Rules were filtered for interpretability by enforcing:

- Minimum support = 0.1
- Minimum lift = 1.0

- Additional filtering for "high-quality rules" (confidence ≥ 0.6, lift ≥ 1.2).

The analysis focused on identifying rules predictive of human vs. bot (non-human) behaviour in Twitter accounts.

2.5.3 Frequent Itemset Generation

- **Method:** Apriori algorithm with `min_support = 0.1`.
- **Output:** 191 frequent itemsets were identified.
- **Examples:**
  - `(high_trust)` → support = 100%
  - `(has_description)` → support = 82%
  - `(high_confidence, has_description)` → support = 64%

These frequent item sets reflect common patterns such as trust signals and complete profile descriptions.

2.5.4 Association Rules Mining

From these item sets, 1,276 rules were generated:

- **Human-indicating rules (303):** Strong predictors were combinations of gender (`is_male`/`is_female`), `high_confidence`, and `has_description`. Example: `{is_male, high_confidence, has_description} → {likely_human}` (Confidence = 1.0, Lift = 2.11).
- **Bot-indicating rules (383):** Bots were strongly linked to missing or very short descriptions. Example:
  `{no_description} → {likely_bot, desc_very_short}` (Confidence = 1.0, Lift = 5.37).
- **High-quality rules (378):** Filtering with confidence ≥ 0.6 and lift ≥ 1.2 revealed highly discriminative rules, again dominated by description-based features.

2.5.5 Results

- **Human classification:** High confidence, gender identity, and having a description strongly predicted humans.

- **Bot classification:** Missing or very short descriptions, often combined with brand and URL presence, strongly predicted bots.
- **Interesting patterns:**
  - Description-based rules (475 total): clear indicators of bots when profile descriptions were short or missing.
  - Tweet content-based rules (362 total): brand accounts with frequent URLs correlated with bots.
- **Visual Analysis:**
  - Scatter plots showed that rules with **high lift (>3)** often had **low support (<0.2)**.
  - Lift distribution: majority near 1, with a small subset reaching >5.
  - Correlation heatmap: negative correlation between support and lift (−0.25), indicating rare rules were often more discriminative.

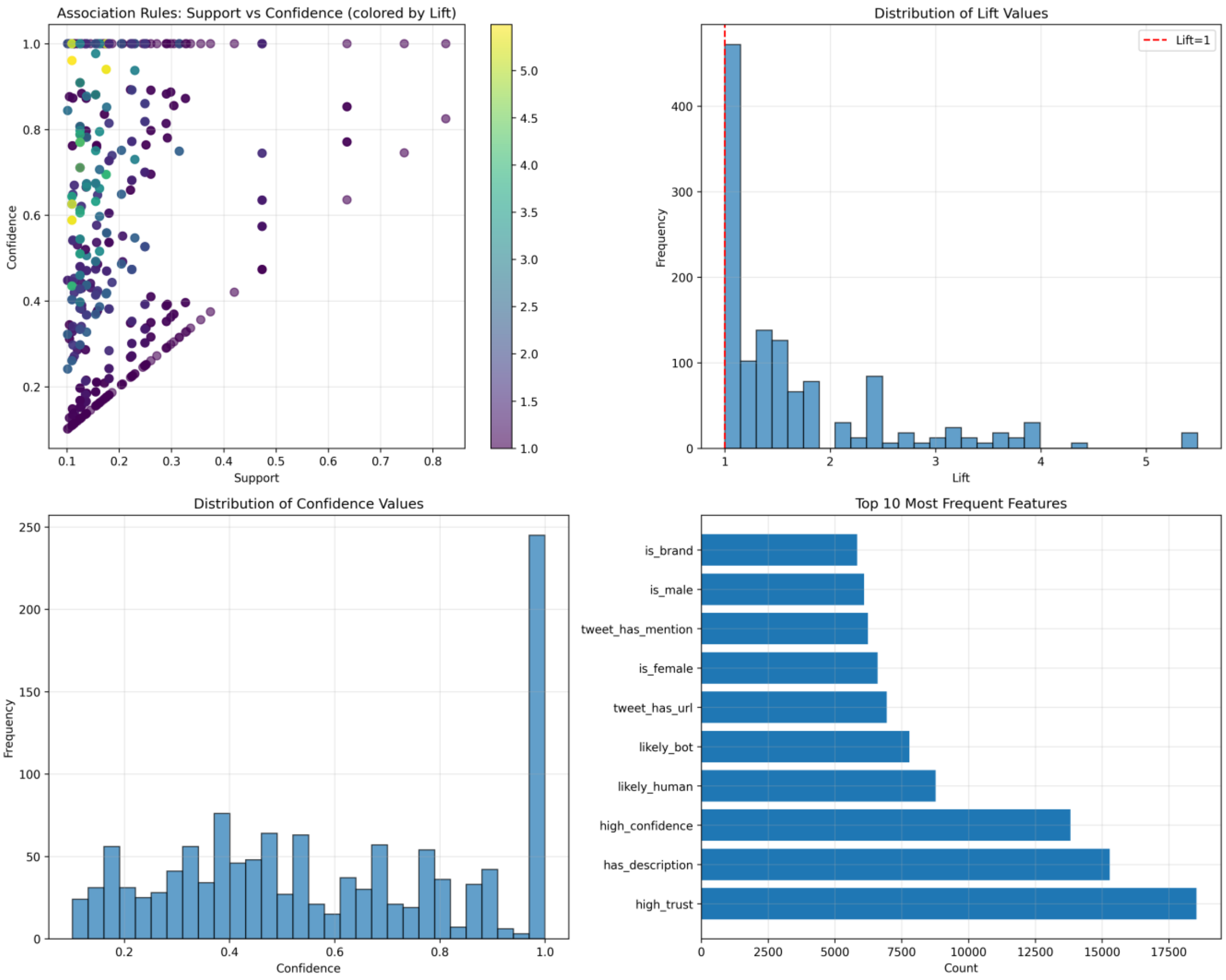


Figure 2.10: Association rule visualizations showing (a) Support vs Confidence scatter (colored by Lift), (b) Lift distribution, (c) Confidence distribution, and (d) Top 10 frequent features.

2.5.6 Comparison

- **Human vs. Bot Rules:**
  - Human rules were generally supported by larger subsets (0.22–0.25 support) but had lower lifts (~2.1).
  - Bot rules had lower support (~0.11–0.17) but much higher lifts (≥5.3), showing stronger discriminative power.
- **Feature Influence:**
  - Human rules leaned on *positive signals* (confidence, gender, descriptions).
  - Bot rules leaned on *absence of information* (no description, very short text).

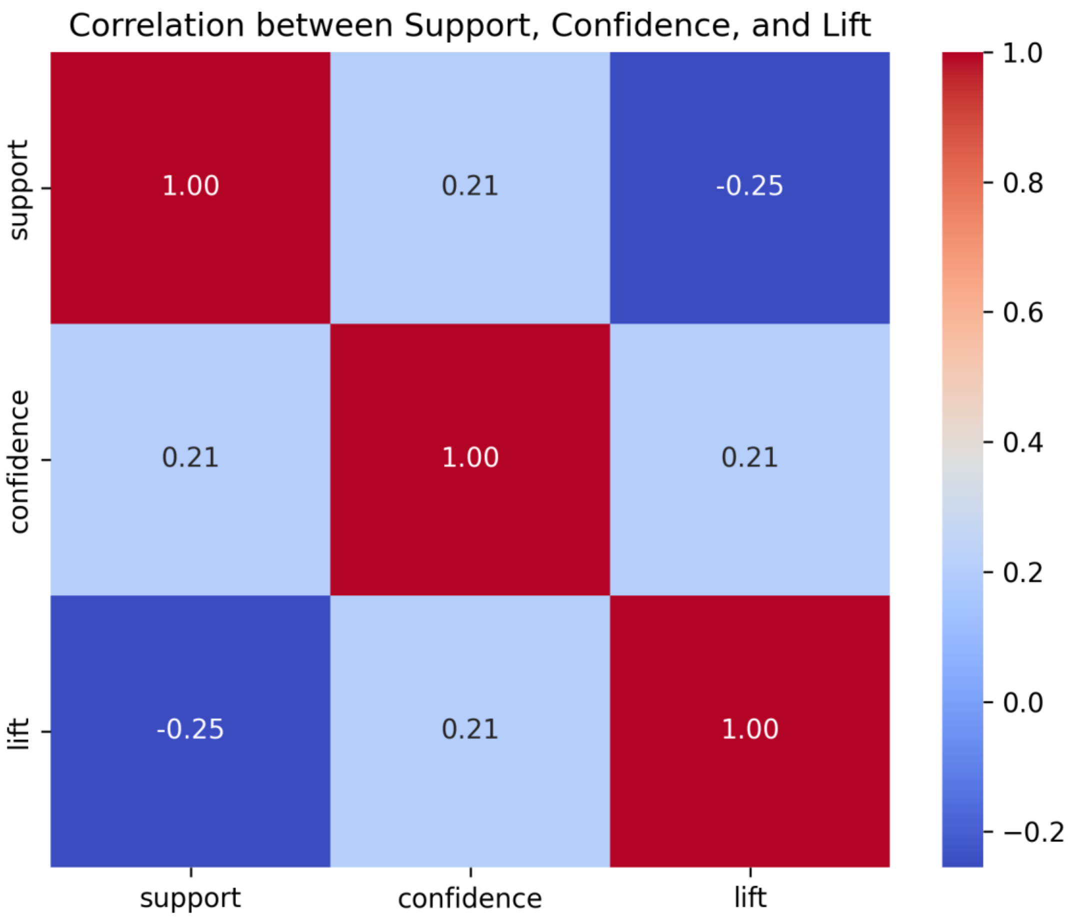


Figure 2.11: Correlation heatmap between support, confidence, and lift for association rules, showing negative correlation between support and lift.

2.5.7 Justification

The association rule mining on the **cleaned dataset** proved effective for distinguishing humans from bots. While human rules had broader coverage (high support), both rules offered stronger predictive strength (high lift). This balance is critical:

- **High support + moderate lift (human rules):** ensures generalizability.
- **Low support + high lift (bot rules):** ensures precise detection of non-human accounts.

Thus, the outcome validates that description completeness and confidence levels are the most critical indicators for classification, with the cleaned dataset enabling reliable and interpretable rules.

2.6 Text processing

2.6.1 Approach

This task investigates the textual information contained in the `description` and `text` fields of the dataset. While classification already used TF-IDF features within a supervised learning setting, the focus here is exploratory text analysis. Several methods were applied, including tokenization, stopword removal, lemmatization, frequency analysis, word cloud visualization, TF-IDF feature extraction, topic modelling, vocabulary diversity checks and sentiment analysis. These steps provide complementary evidence of how linguistic patterns help in distinguishing between human and non-human Twitter accounts.

2.6.2 Data Preparation

The analysis was conducted using the cleaned dataset `df_cleaned.csv` prepared in Task 2.1. This ensured consistency across tasks, as irrelevant columns had been removed and missing values handled during preprocessing.

A new column `combined_text` was created by merging the *description* and *text* fields. Further text preprocessing steps included:

- Conversion to lowercase
- Tokenization with NLTK
- Stopword removal using the NLTK stopword list
- Lemmatization with the WordNet lemmatizer

The cleaned tokens were stored in a new column for subsequent analysis.

2.6.3 Word Frequency Analysis

A frequency distribution was generated from all tokens across the dataset. The most frequent tokens included *http* (9,475), *get* (2,381), *weather* (2,312), *love* (1,716) and *update* (1,356). Terms such as *http*, *update* and *channel* appear repeatedly in automated or brand accounts, while words like *love* and *life* were more common in personal users. This pattern suggests that vocabulary itself carries discriminative signals for identifying account type.

2.6.4 Word Cloud

A word cloud was generated to visualize the distribution of common terms. Larger words such as *love*, *new* and *life* reflected the conversational and emotional style of human users, while repeated words like *weather*, *update* and *channel* highlighted the systematic and promotional content typical of non-human accounts. The word cloud supports the frequency analysis and provides a more intuitive view of the visual comparison.

2.6.5 TF-IDF Feature Extraction

TF-IDF was applied to the `combined_text` column with a vocabulary limited to 5,000 features. The resulting sparse matrix had a shape of 18,532 x 5,000. The extracted terms included both emotional words like *love* and technical or brand related words like *update* and *weather*. These features were later used in downstream tasks such as classification and regression, strengthening predictive performance by quantifying language use.

2.6.6 Topic Modeling by LDA

Latent Dirichlet Allocation (LDA) with five topics was applied to the tokenized corpus. The results indicated several meaningful groupings:

- Topic 0: mixed content (*http*, *love, news)*
- Topic 1: conversational expressions (*like*, *love*, *time, day)*
- Topic 2: social engagement (*people*, *thank, mind)*
- Topic 3: entertainment and fandom (*make*, *fan*, *forevermore, everydayiloveyou)*
- Topic 4: automated updates (*weather*, *update, channel)*

These topics reveal distinct linguistic clusters. Topic 4 clearly demonstrates how automated accounts generate consistent and repetitive themes, while topics 1 and 2 capture the diversity of personal interaction.

2.6.7 Vocabulary Diversity

A vocabulary analysis was performed to compare total and unique tokens. Across 280,704 tokens, 40,436 were unique. This indicates that human users tend to employ a broad and diverse vocabulary, while non-human accounts tend to rely on repetitive word choices. Such vocabulary richness can therefore serve as an additional indicator of natural human expression compared with automated content.

2.6.8 Sentiment Analysis

Sentiment polarity scores were calculated using the VADER lexicon. The distribution showed a strong peak around 0 (neutral), along with wider spreads into both positive and negative ranges. Neutral peaks were largely associated with non-human accounts that post factual or promotional updates, while human accounts were more likely to display both positive and negative extremes through emotional language. Average sentiment scores were similar across groups (≈0.23), but the broader distribution for human users reinforces the value of sentiment as a discriminative signal.

2.6.9 Results

The text analysis highlighted clear differences between human and non-human profiles. Human accounts frequently used emotional, conversational and social terms, showed more vocabulary diversity and displayed wider sentiment range. In contrast, non-human or brand accounts demonstrated promotional and automated patterns, narrower vocabulary and a strong tendency toward neutral sentiment. TF-IDF features provided structured inputs for later models, and LDA topics confirmed thematic separations in language use. Together, these findings contribute to the overall goal of identifying profiles that may be misclassified, showing that textual behavior provides valuable evidence for distinguishing user types.

3 Task 3 – Visualization & Evaluation

3.1 Data Characteristics (Preprocessing & EDA)

Exploratory Data Analysis (EDA) results are visualized here to complement the textual descriptions given in Task 2. Rather than repeating properties, the focus is on how visualizations illustrate data challenges and guide later modelling.

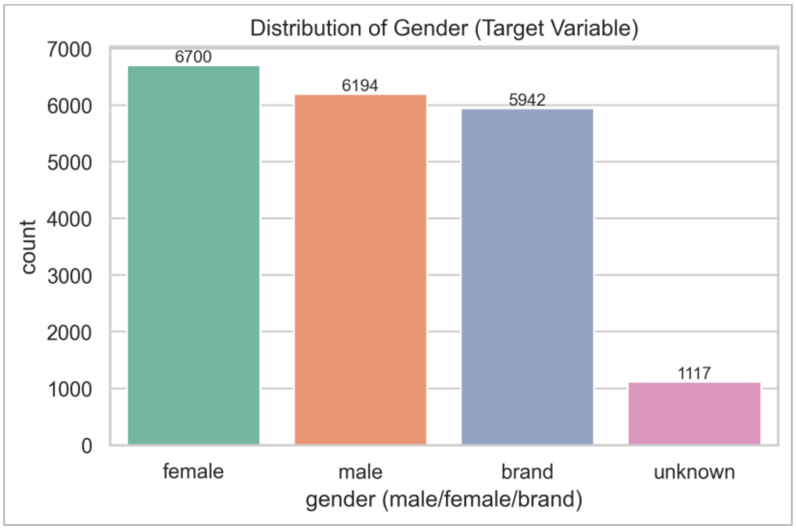


Figure 3.1: Target distribution

Figure 3.1 shows the distribution of the target variable, `gender`. The dataset is relatively balanced across male, female, and brand categories, with a smaller unknown group. This visual confirmation supports Task 2's observation that overall class balance reduces resampling needs, but the unknown class may introduce noise in classification.

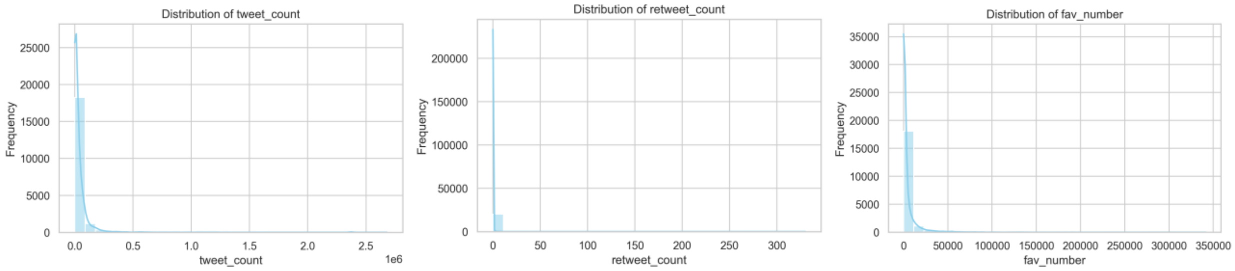


Figure 3.2: Distributions of numeric variables

Figure 3.2 plots the distributions of key numeric variables (`tweet_count, retweet_count, fav_number`). All three are heavily skewed, with long tails caused by a few extremely active accounts. This reflects typical social media patterns and supports the recommendation from Task 2 that outlier handling such as log-scaling is important for robust modelling.

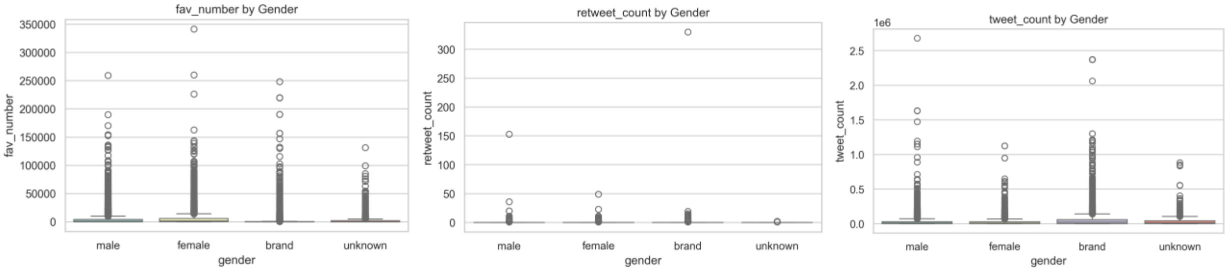


Figure 3.3: Boxplots by gender

Figure 3.3 presents boxplots by gender, showing that brand accounts occupy higher ranges for `tweet_count` and `fav_number` than individuals. This confirms the textual finding that activity-level metrics can serve as useful discriminators between human and non-human accounts.

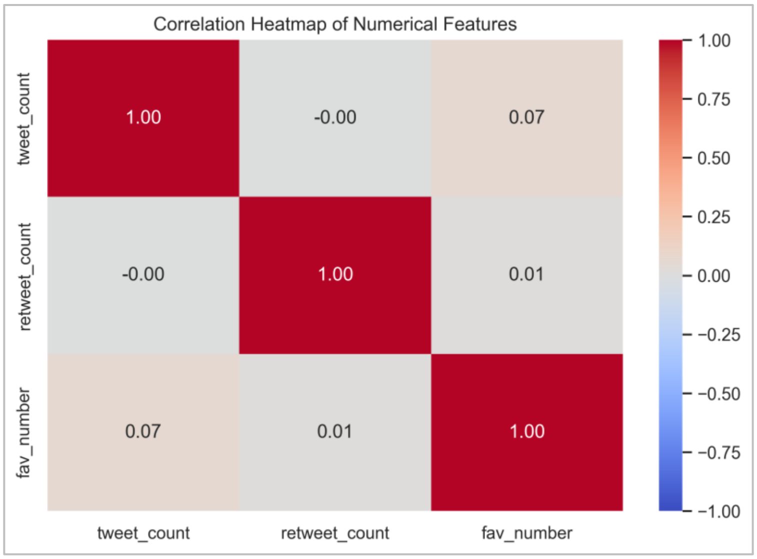


Figure 3.4: Correlation heatmap

Figure 3.4 visualizes correlations among numeric features. The relationships are very weak, meaning each variable contributes unique information. This aligns with Task 2's implication that multicollinearity is not a concern, and all features can be retained.

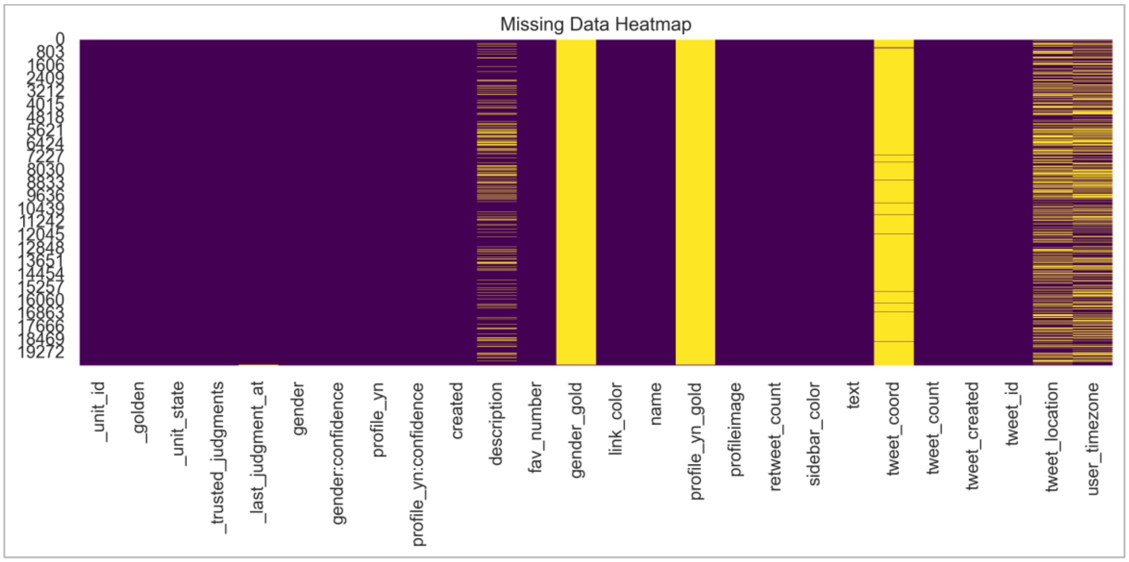


Figure 3.5: Missing data heatmap

Figure 3.5 highlights missing data patterns. While `gender_gold, profile_yn_gold` , and `tweet_coord` variables have extreme missingness and should be removed, most other variables have only partial gaps. This visualization supports earlier preprocessing decisions, such as dropping highly incomplete fields and imputing moderate ones.

Overall, these EDA visualizations validate the descriptive findings from Task 2 and explain why later models, especially those leveraging textual features, outperformed numeric-only baselines.

3.2 Association Rule Insights

Association rule mining produced interpretable co-occurrence patterns that explain human vs. non-human behavior. Visualizations highlight both the strengths and trade-offs of these rules.

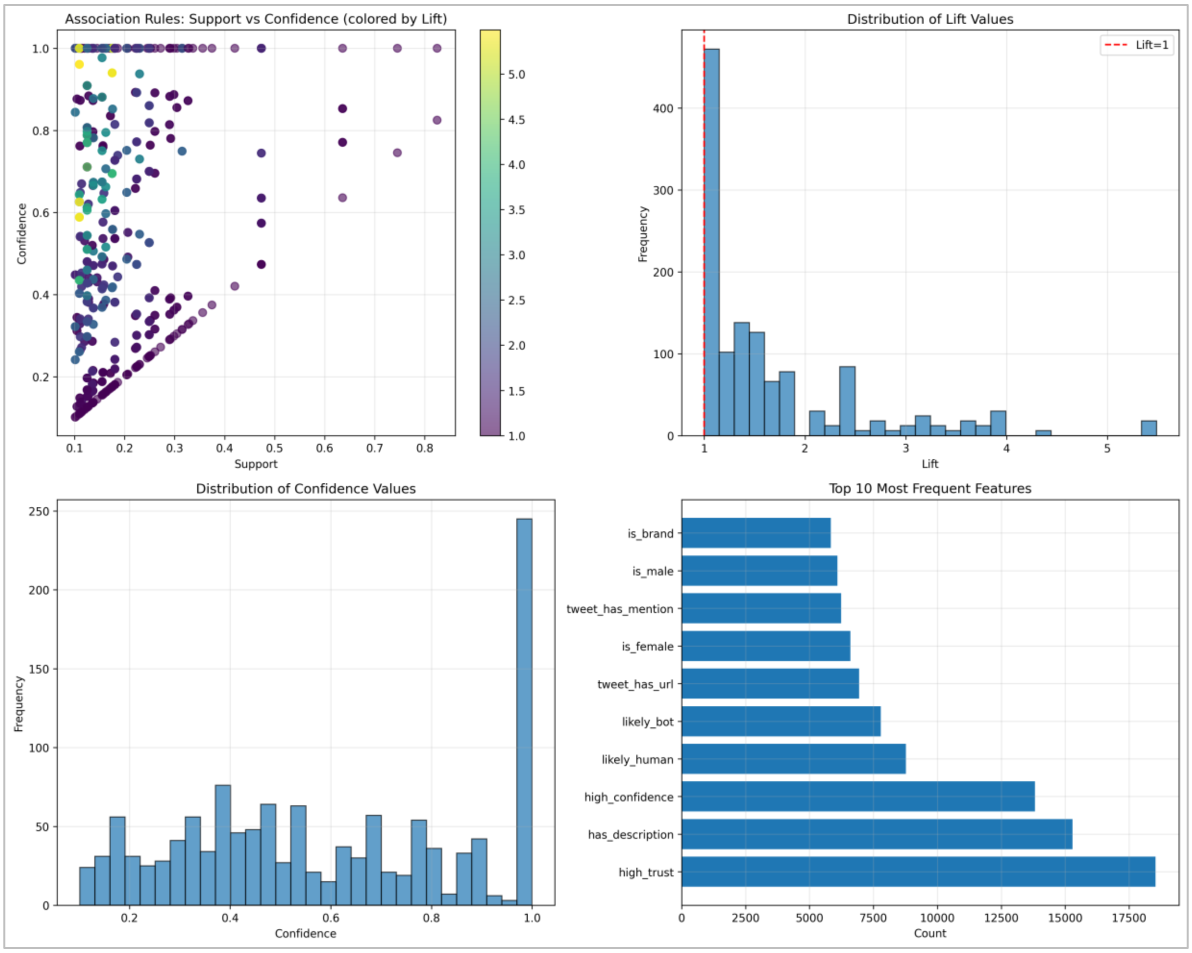


Figure 3.6: Association rule visualizations showing (a) Support vs Confidence scatter (colored by Lift), (b) Lift distribution, (c) Confidence distribution, and (d) Top 10 frequent features.

Figure 3.6a shows the scatter of support vs. confidence, coloured by lift. Rules with very high lift (>3) often had low support (<0.2), meaning they are highly discriminative but cover only a small portion of accounts. **Figure 3.6b** displays the distribution of lift values, with most rules clustered around lift ≈ 1, confirming that only a subset of rules carries strong predictive power. **Figure 3.6c** presents the distribution of confidence values, where many rules exceed the 0.6 threshold, indicating stable predictive strength. Finally, **Figure 3.6d** ranks the top 10 frequent features, where `has_description,` `high_confidence`, and `is_brand` dominate, highlighting that description completeness and confidence are decisive signals.

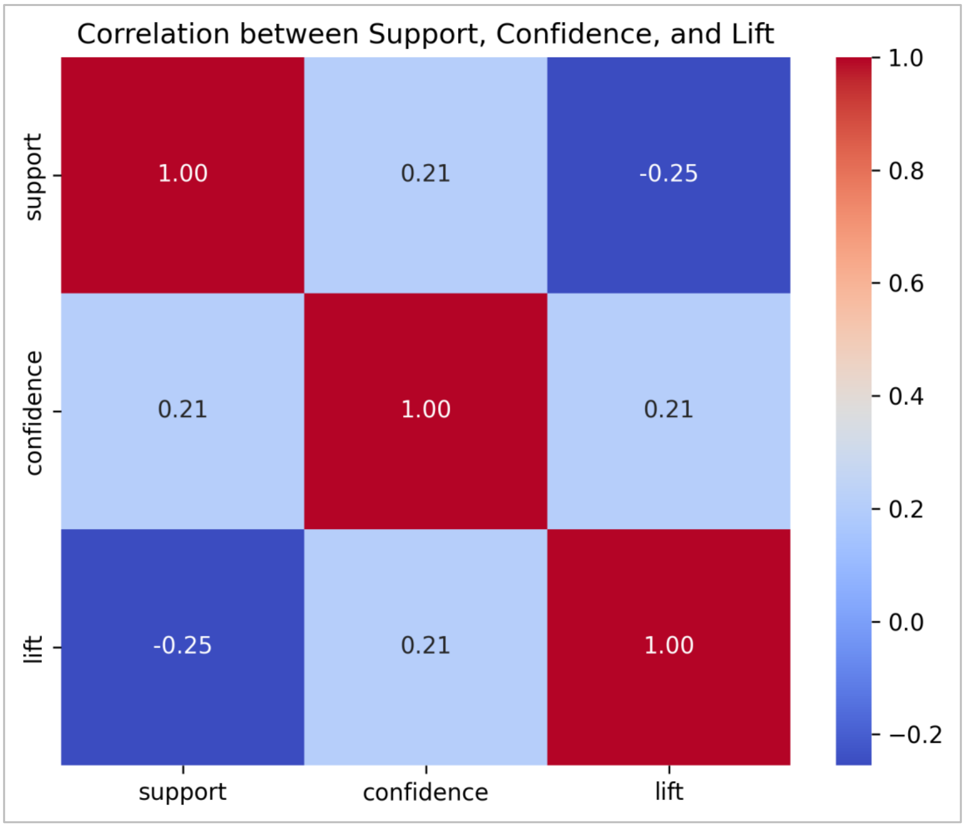


Figure 3.7: Correlation heatmap (support, confidence, lift)

Figure 3.7 presents the correlation heatmap between support, confidence, and lift. The negative correlation between support and lift (≈ –0.25) reinforces the trade-off: human-indicating rules are broadly supported but weaker in discriminative strength, whereas bot-indicating rules are rarer but highly precise.

Type	Antecedents	Consequents	Support	Confidence	Lift
Human	`{is_male, high_confidence, has_description}`	{likely_human}	0.22	1.00	2.11
Human	`{is_female, has_description}`	{likely_human}	0.25	0.95	1.98
Human	`{high_confidence, has_description}`	{likely_human}	0.64	0.92	2.05
Bot	`{no_description}`	{likely_bot}	0.17	1.00	5.37
Bot	`{desc_very_short, tweet_has_url}`	{likely_bot}	0.12	0.97	4.85
Bot	`{is_brand, no_description}`	{likely_bot}	0.15	0.93	5.12

Table 3.1: Representative Association Rules (Human vs Bot)

Table 3.1 summarizes representative rules. Human-indicating rules are generally supported by larger subsets, driven by positive signals such as high confidence and profile descriptions. In contrast, bot-indicating rules rely on missing or very short descriptions, often combined with brand or URL features, and show much higher lift despite lower support. These findings align with classification results, where text-based features consistently delivered stronger F1-scores.

Overall, the rule-based visualizations highlight why description and confidence fields were so influential: they provide interpretable evidence that complements machine learning models by clarifying why specific profiles are predicted as human or non-human.

3.3 Clustering Results Evaluation

Clustering was applied to uncover latent groupings of user profiles. The visualizations highlight both the potential and limitations of unsupervised approaches.

KMeans (Figure 3.8). KMeans provided a baseline partition of the dataset into three clusters. The distribution partially overlapped with gender labels, confirming only moderate alignment with the annotated classes. Silhouette analysis (~0.4) suggested weak separation, indicating that simple partitioning could not achieve reliable discrimination.

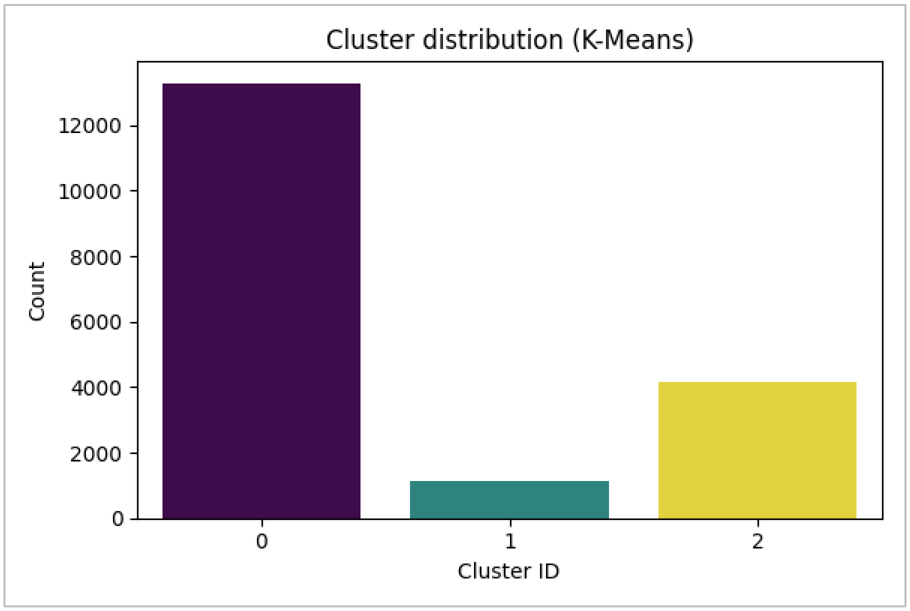


Figure 3.8: KMeans cluster distribution

Hierarchical Clustering (Figure 3.9). Hierarchical clustering on a 500-user sample revealed subgroup relationships and branching patterns. While interpretable, the dendrogram showed no sharp separation between male, female, and brand accounts. This underlines its value for exploratory structure but also its practical limitations on larger datasets.

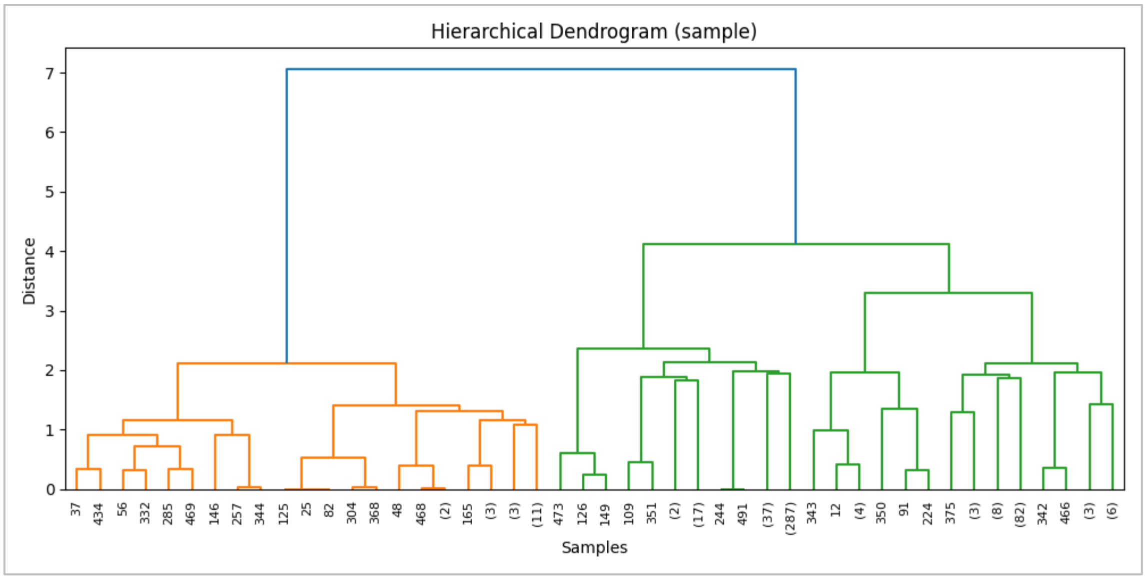


Figure 3.9: Hierarchical dendrogram

Self-Organizing Map (SOM) (Figure 3.10, Table 3.2) The SOM projected high-dimensional features into a 2D grid, revealing density hotspots. Table 2 shows the ten largest clusters, with sizes ranging from ~400 to 1100 users, reflecting uneven distribution and heterogeneity across account types. Despite offering an intuitive visualization, overlaps again confirmed that SOM alone was insufficient to distinguish humans from brands.

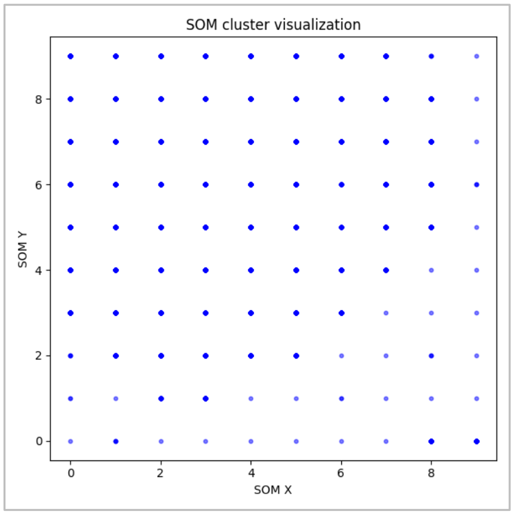


Figure 3.10: SOM visualization

	cluster_som	count
0	8-0	1098
1	4-5	995
2	2-4	961
3	4-8	622
4	2-6	586
5	5-4	503
6	3-5	498
7	1-3	492
8	5-5	486
9	3-4	441

Table 3.2: SOM Cluster Counts (Top 10)

Overall, these visualizations demonstrate that clustering captured some structural signals but lacked the precision needed for classification. This aligns with later supervised results, where text-driven models achieved much stronger separation.

3.4 Classification Results

Supervised models were evaluated to classify Twitter profiles into male, female, and brand accounts. Each method was trained on the same TF-IDF + numeric confidence features, ensuring fair comparison. Confusion matrices illustrate class-specific misclassifications, while F1-based metrics capture overall performance.

Decision Tree (Figure 3.11)

The Decision Tree produced interpretable splits but showed signs of overfitting in high-dimensional text features. The confusion matrix highlights misclassifications across minority categories. Despite reasonable accuracy, macro-F1 was the lowest among the three models, reflecting poor balance across classes.

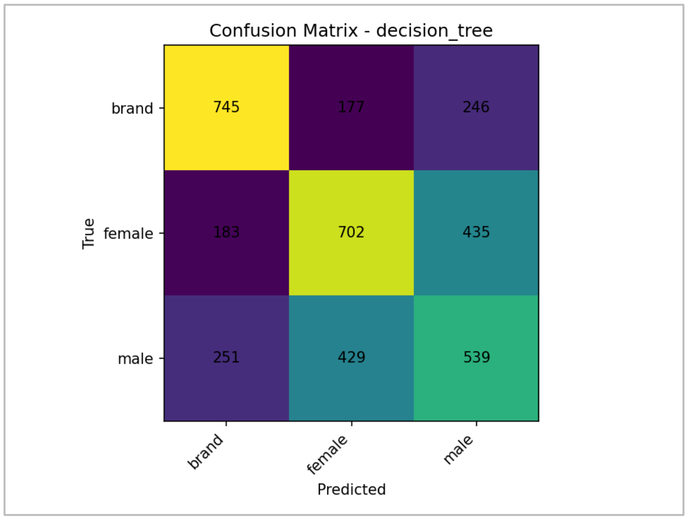


Figure 3.11: Decision Tree confusion matrix

Logistic Regression (Figure 3.12)

Logistic Regression generalized well to sparse TF-IDF features, providing the most balanced precision and recall across categories. The confusion matrix demonstrates more consistent performance, with fewer systematic errors compared to Decision Trees. Logistic Regression achieved the best weighted F1 score, confirming robustness under class imbalance.

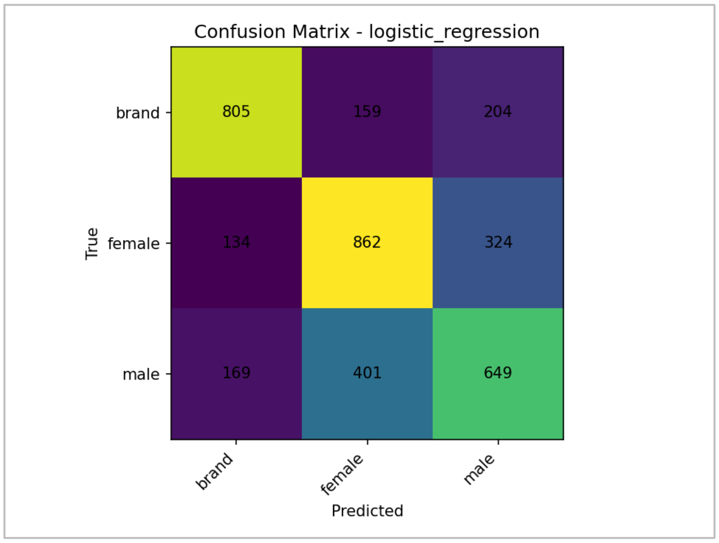


Figure 3.12: Logistic Regression confusion matrix

Naïve Bayes (Figure 3.13)

Multinomial Naïve Bayes leveraged word-frequency assumptions effectively, performing strongly in major classes. However, it tended to misclassify minority labels more often. Its macro-F1 was competitive with Logistic Regression, demonstrating efficiency as a lightweight baseline.

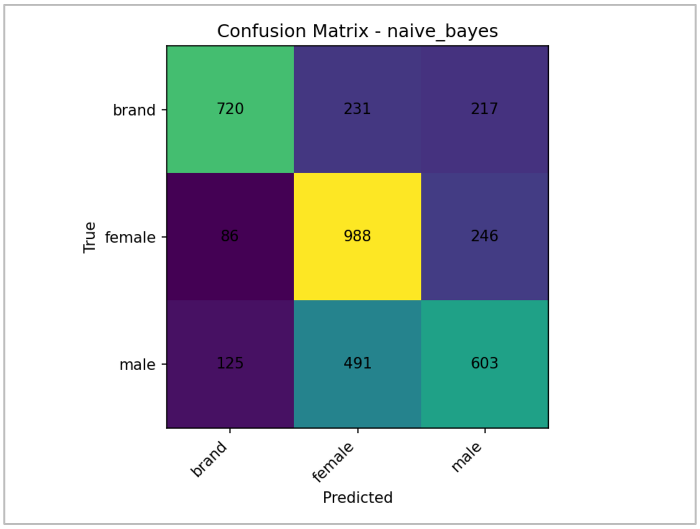


Figure 3.13: Naïve Bayes confusion matrix

Cross-Model Comparison (Figures 3.14–3.15, Table 3.3)

Macro-F1 (**Figure 3.14**) emphasized fairness across categories, where Logistic Regression performed best, followed closely by Naïve Bayes. Weighted F1 (**Figure 3.15**) reflected real-world deployment, again favoring Logistic Regression.

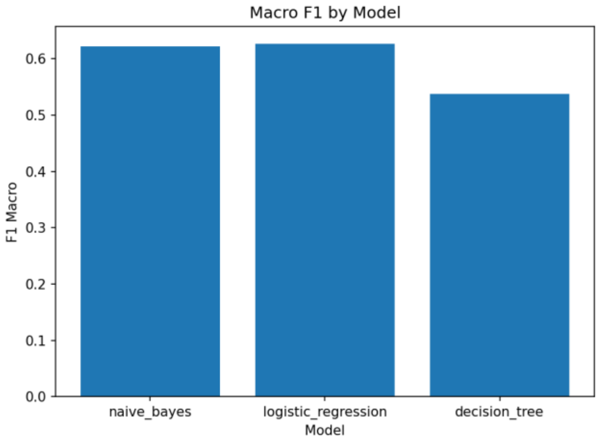


Figure 3.14: Macro-F1 by model

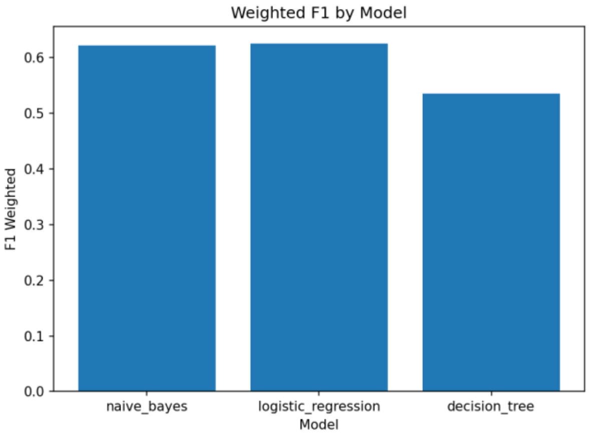


Figure 3.15: Weighted F1 by model

Table 3.3 presents per-class metrics (precision, recall, F1, support) across classifiers. These results show that male and female categories were predicted more consistently, while brand and unknown users proved harder to classify, reflecting the challenges of minority or noisy labels.

model	class	precision	recall	f1	support
Naive_bayes	brand	0.77	0.62	0.69	1168
Naive_bayes	female	0.58	0.75	0.65	1320
Naive_bayes	male	0.57	0.49	0.53	1219
Logistic_regression	brand	0.73	0.69	0.71	1168
Logistic_regression	female	0.61	0.65	0.63	1320
Logistic_regression	male	0.55	0.53	0.54	1219
Decision_tree	brand	0.63	0.64	0.63	1168
Decision_tree	female	0.54	0.53	0.53	1320
Decision_tree	male	0.44	0.44	0.44	1219

Table 3.3: Per-class metrics

Table 3.4 provides the overall model scores, including accuracy and macro-/weighted-average metrics. Logistic Regression achieved the strongest balance across evaluation measures, Naïve Bayes offered a competitive baseline, and Decision Trees lagged due to overfitting. Together, these comparisons confirm that text-oriented linear models are most effective for this dataset.

model	accuracy	Precision _macro	Recall_ macro	F1_ macro	Precision_ wesighted	Recall_w eighted	F1_wei ghted
Naive_bayes	0.62	0.64	0.62	0.62	0.64	0.62	0.62
Logistic_ regression	0.62	0.63	0.62	0.63	0.63	0.62	0.62
Decision_tree	0.54	0.54	0.54	0.54	0.54	0.54	0.54

Table 3.4: Model Scores Overview

3.6 Text Processing Evaluation

Textual analysis yielded the clearest signals for distinguishing between human and non-human accounts. Word frequency and word clouds highlighted divergent vocabularies, while TF-IDF and topic modeling revealed deeper structural differences. Vocabulary diversity and sentiment analysis further emphasized that human accounts used richer and more affective language, whereas non-human accounts were repetitive and neutral. These findings explain why text-based features substantially enhanced classification performance compared with numeric attributes.

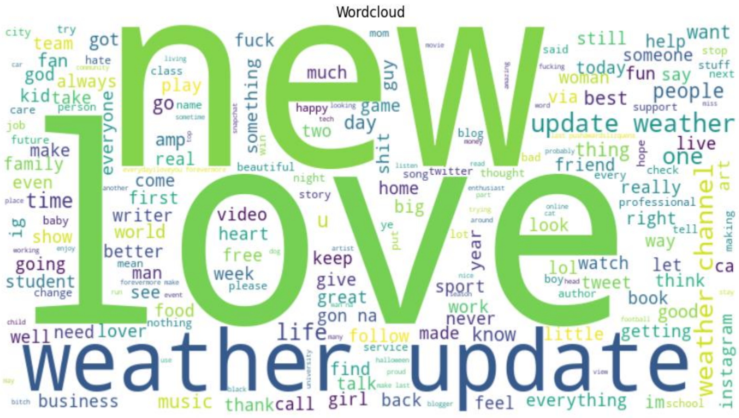
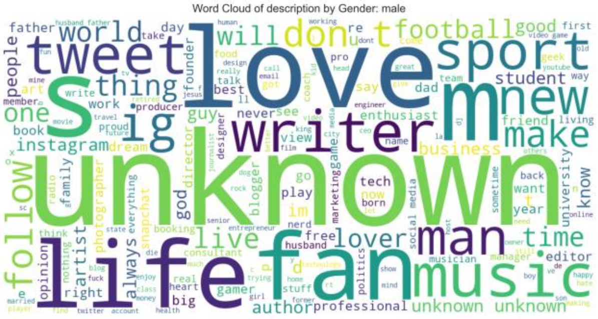


Figure 3.16: Global word cloud of frequent terms across all accounts.

Figure 3.16 shows the global word cloud, where frequent conversational and promotional terms such as *love*, *new*, and *update* dominate. This visualization provides an overall view of the most salient tokens in the dataset.



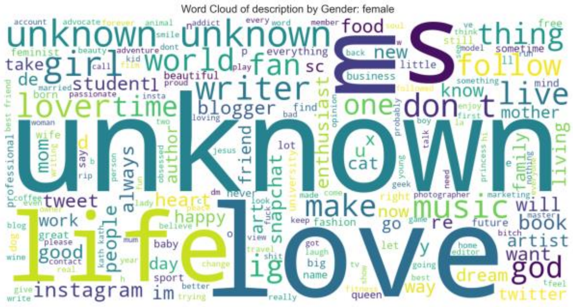


Figure 3.17: Word cloud of male accounts **Figure 3.18**: Word cloud of female accounts

Figure 3.17 illustrates the male account word cloud, which emphasizes affective and social terms such as *life*, *friend*, and *music*. These highlight the conversational and personal tone typical of male users. **Figure 3.18** presents the female account word cloud, dominated by words like *love*, *happy*, and *family*. Compared with male users, female accounts place stronger emphasis on positive sentiment and relational themes.

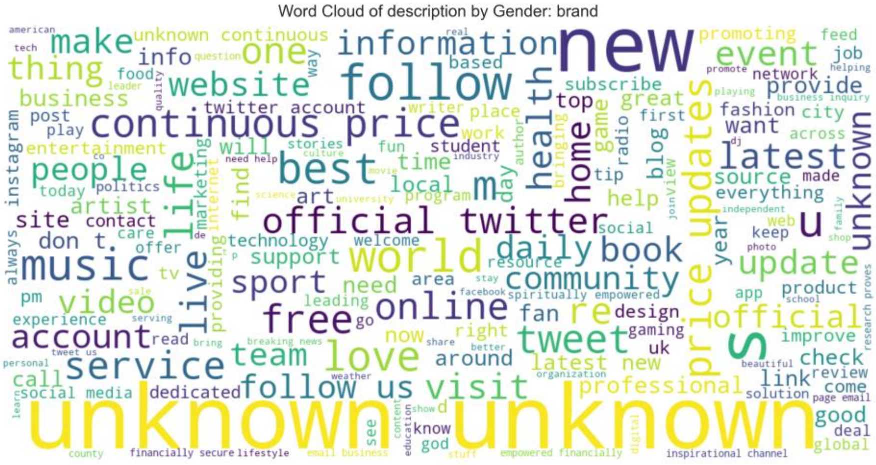


Figure 3.19: Word cloud of brand accounts

Figure 3.19 shows the brand account word cloud, which is characterized by repeated promotional and informational terms such as *official*, *update*, and *weather*. The restricted vocabulary and repetitive phrasing underscore the automated and formulaic nature of brand communication.

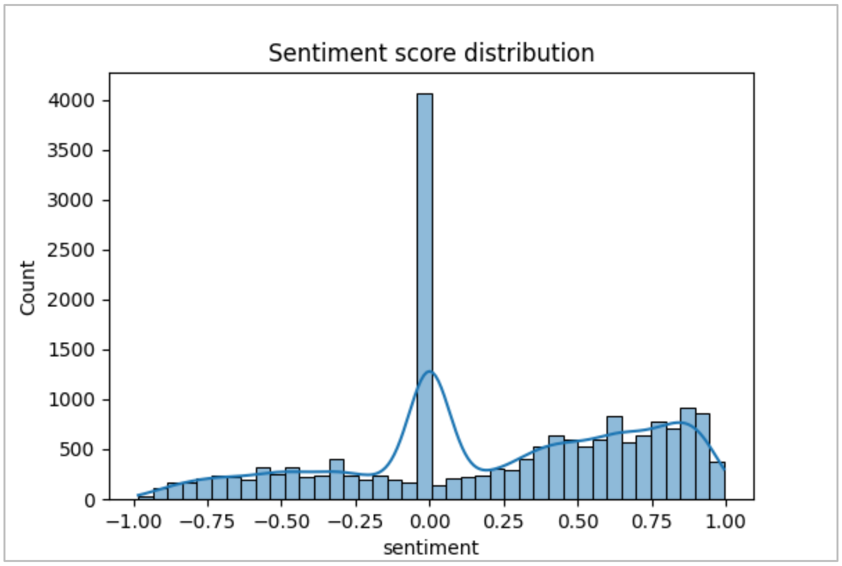


Figure 3.20: Sentiment distribution

Figure 3.20 visualizes the sentiment distribution across accounts. Brand accounts are strongly skewed toward neutrality, while human users demonstrate more polarity, with both positive and negative expressions. This polarity further reinforces the distinction between human and automated content.

Overall, these textual visualizations confirm that text-based features provide the clearest separation between human and non-human accounts. The divergence in vocabulary and sentiment explains why text-oriented models, particularly Naïve Bayes and Logistic Regression, performed strongly: they exploited linguistic diversity, thematic structure, and affective cues largely absent from numeric-only features.

4 Task 4 – Multi-view Study & Suggestions

4.1 Objective

Study multiple views of the Twitter dataset (text, profile colours, and activity/tweet metrics) and provide short, practical suggestions to correct or amend non-human (brand) and human profiles. The analysis uses the cleaned dataset and figures generated by our EDA/Text pipelines.

4.2 Data Views Considered

- Tweet/Activity view: `tweet_count, retweet_count, fav_number`.
- Colour view: profile link_color and sidebar_color.
- Text view: user descriptions + posts merged as “`combined_text`”.

- Label confidence view: `gender:confidence` distribution for label quality.

4.3 Key Findings by View

i) Tweet/Activity View

- Activity is heavy-tailed: a few accounts post extremely often, while most accounts remain low-activity.
- Brands tend to occupy the higher end of `tweet_count` and `fav_number` compared with individuals.
- Correlations among activity variables are weak, meaning each measure contributes an independent signal.

ii) Color View

- Most users use default Twitter-blue (#0084B4) and light palettes; brands more often use high-saturation or custom colors.
- Color alone is a weak classifier, but useful as an auxiliary indicator together with text/activity.

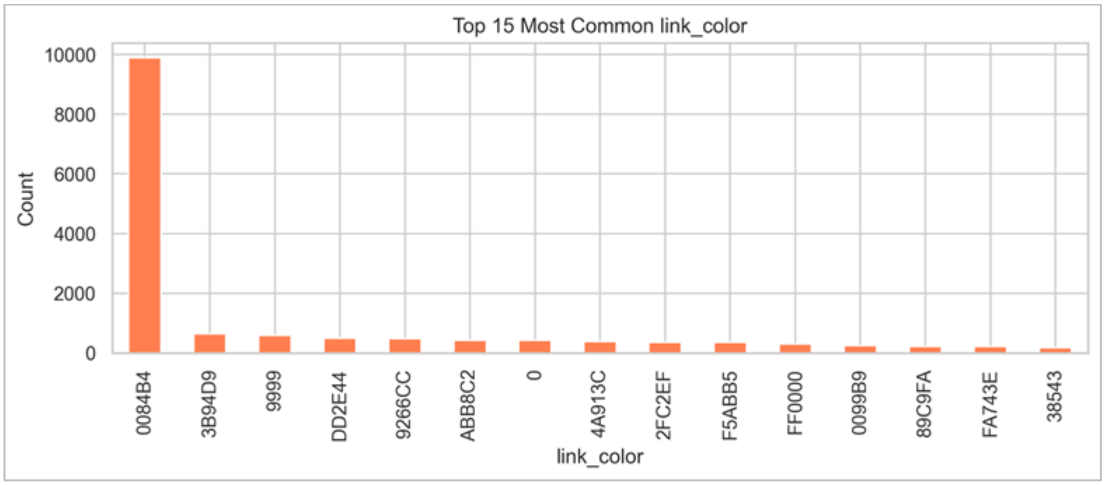


Figure 4.1: Top15 common link

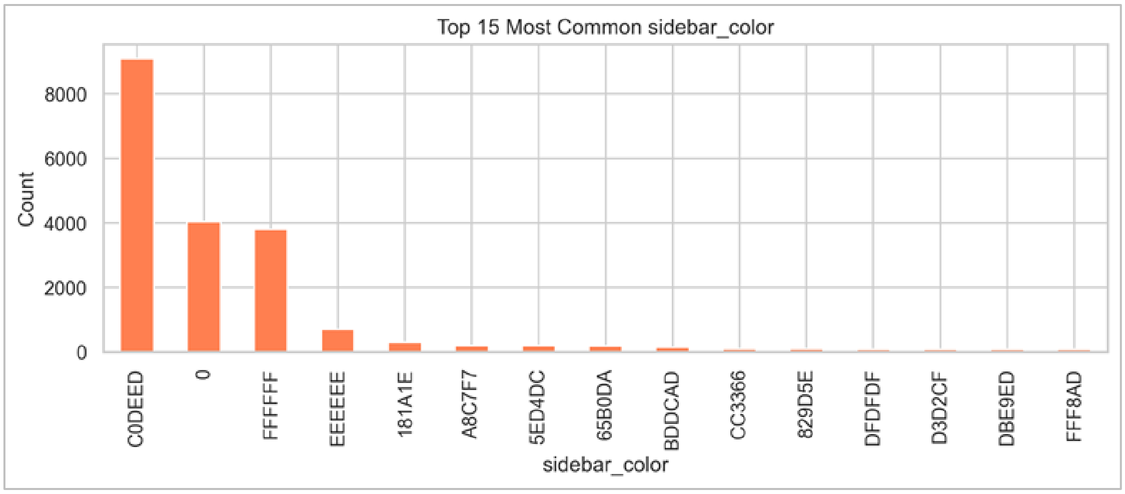


Figure 4.2: Common top15 sidebar color

iii) Text View

- Human profiles use more affective and personal words (*love, life, happy, friend, music*).
- Brand profiles contain promotional and information words (*official, support, update, video, new*).
- Unknown groups are mixed and often contain placeholders such as "*unknown*".

iv) Label Confidence View

Most rows have high gender: confidence; a small portion shows mid/low confidence and should be handled carefully in training/evaluation.

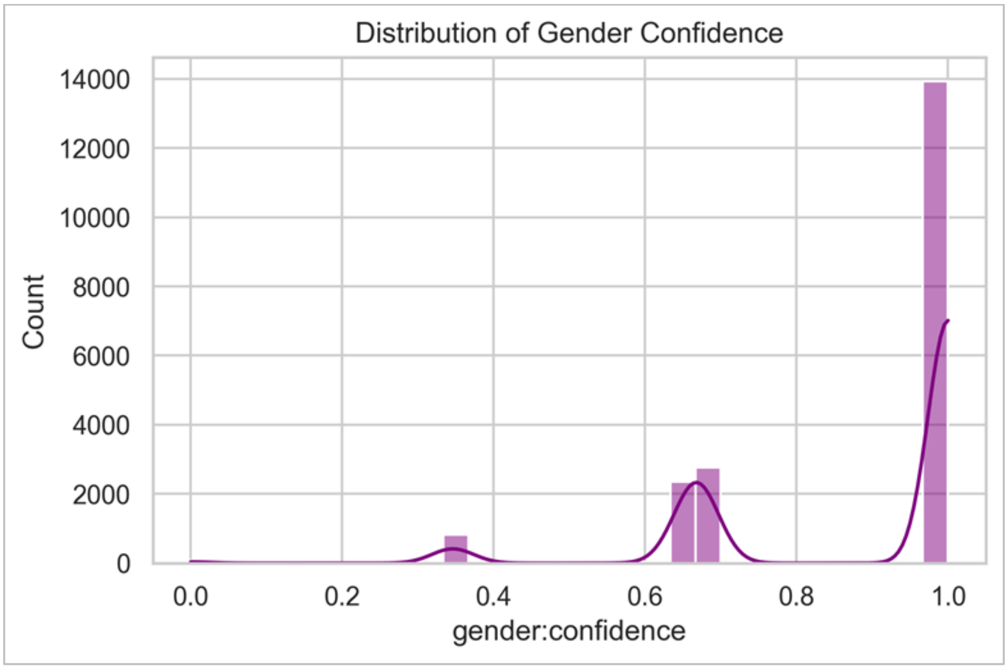


Figure 4.3: Gender confidence

4.4 Suggestions to Amend/Adjust Profiles

4.4.1 For Non-Human (Brand) Profiles

- Add/verify brand indicator features: high URL ratio in text, presence of promo keywords (*official, update, subscribe, support*).
- Use higher activity thresholds (`tweet_count, fav_number`) as supporting evidence do not rely on them alone.
- Flag accounts with custom high-saturation colors combined with promo keywords for manual verification.

4.4.2 For Human Profiles

- Boost features capturing affective/personal language (*love, life, happiness, I/my/we, friend*).
- Down-weight very high activity outliers unless text is also promotional/templated.
  If `gender:confidence` is low, treat prediction as uncertain and prefer manual review.

4.4.3 Simple Rule-of-Thumb (to complement the model)

- If (promo keywords) + (URL present) + (custom saturated color) → likely non-human (brand).
- If (personal/affective words) + (few URLs) + (typical/default colors) → likely human.
- If confidence is low (0.3–0.6) or signals conflict → send to manual review.

4.4 Summary

Across text, color, and activity views, brands post more often, use promotional or URL-heavy language, and sometimes choose saturated custom colors. Humans, on the other hand, use more personal and affective words and usually keep default palettes. Color is helpful but not decisive; text carries the strongest signal, while activity measures help highlight likely non-human accounts. The simple rules above can guide profile adjustments, with low-confidence or conflicting cases sent for manual review.

5 References

[1] Scikit-learn Documentation. *scikit-learn: Machine Learning in Python.* [Online]. Available: https://scikit-learn.org/stable

[2] Pandas Documentation. *pandas: Python Data Analysis Library.* [Online]. Available: https://pandas.pydata.org/docs/

[3] NumPy Documentation. *NumPy: Fundamental package for scientific computing with Python.* [Online]. Available: https://numpy.org/doc/

[4] Matplotlib Documentation. *Matplotlib: Visualization with Python.* [Online]. Available: https://matplotlib.org/stable/contents.html

[5] Seaborn Documentation. *Seaborn: Statistical Data Visualization.* [Online]. Available: https://seaborn.pydata.org/

[6] SciPy Documentation. *SciPy: Scientific Library for Python.* [Online]. Available: https://docs.scipy.org/doc/scipy/

[7] Python Software Foundation. *Python 3 Documentation.* [Online]. Available: https://docs.python.org/3/

[8] Project Jupyter. *Jupyter Notebook Documentation.* [Online]. Available: https://jupyter-notebook.readthedocs.io/

[9] Google Colab. *Introduction to Google Colaboratory.* [Online]. Available: https://colab.research.google.com/

[10] MiniSom GitHub Repository. *Minimalistic implementation of Self Organizing Maps in Python.* [Online]. Available: https://github.com/JustGlowing/minisom

[11] Towards Data Science. *K-Means Clustering in Python: A Practical Guide.* [Online]. Available: https://towardsdatascience.com/k-means-clustering-in-python-8e1e64c1561c

[12] Analytics Vidhya. *A Comprehensive Guide to Clustering in Machine Learning.* [Online]. Available: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

[13] GeeksforGeeks. *Decision Tree Algorithm for Classification in Python.* [Online]. Available: https://www.geeksforgeeks.org/decision-tree/

[14] Medium. *Understanding Naive Bayes Classifier.* [Online]. Available: https://medium.com/swlh/naive-bayes-classifier-explained-84f02f5cc0c5

[16] Spark MLlib. *Machine Learning Library for Apache Spark.* [Online]. Available: https://spark.apache.org/mllib/

[17] Stack Overflow. *Community Q&A on Python, Pandas, and Machine Learning.* [Online]. Available: https://stackoverflow.com/

[18] E. E. Services, Data science and big data analytics: discovering, analyzing, visualizing and presenting data, Chapter 3-9, Wiley, 2015.

[19] Stack Overflow. *Community Q&A on Python, Pandas, and Machine Learning.* [Online]. Available: https://stackoverflow.com/

[20] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.