Twitter Dataset Analysis

Group 17

Name	**Student Number (SN)**	**Contribution**
Karan Goel	7836685	14.28%
Alvin Jose	8066358	14.28%
Ashutosh Bhosale	7795786	14.28%
Banin Sensha Shreshta	8447196	14.28%
Gaurav Adarsh Santosh	7032663	14.28%
Lino Thankachan	7926017	14.28%
Rishab Manokaran	7863974	14.28%

CSCI946 Big Data Analytics Assignment 2

September 20, 2024

1 Introduction

In this assignment, the primary objective is to detect misinformation on social networks by identifying profiles that are incorrectly classified as human or non-human. The focus of the analysis will be on the Twitter user dataset, where we aim to explore methods to distinguish between genuine and artificially generated accounts. To achieve this, we will follow a structured approach outlined in Tasks 1-4.

Task 1 involves designing a comprehensive big data analytics project, adhering to the principles of the Big Data Analytics Lifecycle.

In Task 2, the dataset will be processed by taking into account the various data types and properties. Core models and algorithms will be applied, including regression, association rules, clustering, classification, and text processing methods.

Task 3 focuses on visualizing the dataset and utilizing visual representations to evaluate the analysis results. This step will provide valuable insights into the data and help validate the findings.

Finally, Task 4 entails a detailed study of various profile factors, such as text, color, and tweet content. Based on this analysis, recommendations will be made to adjust the classification of human and non-human profiles.

2 Task 1: Data Analysis Design

2.1 Learn Business Domain

Objective: The aim of assignment this is to identify profiles on social networks that are mistakenly recorded as human or non-human profiles (e.g., bots).
Specific Task: You need to identify if the user's gender is either human (male, female) or non-human(brand) based on their Twitter profile data and associated tweets.
Dataset: The dataset for this classification task can be accessed via Kaggle at the Twitter User Gender Classification dataset.

2.2 Define Resources & Goals

Data Source: Twitter user profiles including tweets, descriptions, link color, sidebar color, and other meta-data.
Primary Task: Classify gender into two categories: human or non-human.
Key Questions:

- How well do the words in tweets and profiles predict the user as human or non-human?
- What are the specific words that strongly predict human or non-human?
- How well do other factors (like link color, tweet count, retweet count, favourite number) predict user as human or non-human?

2.3 Frame the Problem & Develop Initial Hypotheses

Problem Type: This is a supervised learning problem where the task is to predict a categorical variable (gender: male, female, or brand).
Hypotheses:

- **Null Hypothesis (H0)**: Words in tweets and profiles do not have a significant effect on predicting user gender (i.e., the predictive power is random or weak).
- **Alternative Hypothesis (H1)**: Words in tweets and profiles have a significant effect on predicting user gender (i.e., they provide strong predictive power).
- **Additional Hypotheses**:
  - **H0**: Stylistic factors (link color, sidebar color) are not good predictors of gender.
  - **H1**: Stylistic factors can strongly predict gender.

2.4 Data Preparation

Preprocessing Steps:

- **Stemming & Lemmatization**: To reduce words to their base or root forms.
- **Lowercasing**: To eliminate discrepancies between uppercase and lowercase letters.
- **Tokenization**: Split text into individual words or tokens.
- **Embedding**: Use word embeddings (e.g., Word2Vec, GloVe, or TF-IDF) to represent text data numerically.
- **Bag of Words**: A simple and effective method for text representation.
- **Word2Vec**: A simple model to generate world embeddings.

2.5 Visualize Data

Visualization Techniques:

- **Word Cloud**: To display frequent words by gender category (male/female/brand).
- **Word Distance/Similarity**: Use techniques like cosine similarity to understand which words or phrases are closer to each gender category.

2.6 Model Building

Classification Models:

- **K-Nearest Neighbors (KNN)**, **Support Vector Machines (SVM)**, **Decision Trees**, **Random Forest**.

Clustering Models:

- **K-Means**, **DBSCAN**, **Self-Organizing Maps (SOM)**.

Regression Models:

- **Linear Regression**, **Logistic Regression**.

Neural Networks:

- Build deep learning models to classify gender using **Neural Networks**.

2.7 Training and Testing

Cross-Validation: Use **k-fold cross-validation** to prevent overfitting and assess model generalization.
Grid Search: Optimize hyperparameters by performing grid search to select the best model configuration.

2.8 Deliverables

Evaluation Metrics:

- **Confusion Matrix**: To visualize the performance of each classification model.
- **Accuracy**: The percentage of correctly classified instances.
- **Precision**: The proportion of true positives out of all positive predictions.
- **Recall**: The proportion of true positives out of all actual positives.
- **Loss Curves**: For neural networks, to visualize the training process and detect overfitting or underfitting.

3 Task 2: Data Processing & Model Application

The dataset was loaded using the `load_data` method from the `Task1` class, which reads a CSV file containing Twitter user data. After loading, the data was preprocessed to make it suitable for the classification task.

3.1 Data Filtering

- Rows with a `gender:confidence` score greater than 0.9 were selected to ensure data quality.
- The `gender` labels were restricted to three categories: `male`, `female`, and `brand`.
- The `gender` column was relabeled: `male` and `female` were grouped into a `human` class, while `brand` was grouped into the `non-human` class.

3.2 Handling Missing Values

Missing values were filled with empty strings (`''`) to avoid issues during text processing and model training.

3.3 Feature Selection

The key features selected for the classification task were `gender`, `fav_number`, `retweet_count`, `tweet_count`, and `text`.

3.4 Feature Scaling and Label Encoding

Two important steps were applied to further process the numerical and categorical data:

3.4.1 Scaling Features

We used `StandardScaler` to normalize numerical features such as `fav_number`, `retweet_count`, and `tweet_count`. This step ensures that all features are on a similar scale, improving the performance of machine learning algorithms.

3.4.2 Label Encoding

The `gender` column was encoded using `LabelEncoder` to convert the `human` and `non-human` categories into numerical labels.

3.5 Exploratory Data Analysis

To better understand the relationships between features, we performed the following analyses:

3.5.1 Correlation Matrix

We computed the correlation matrix for the numerical features to observe the strength of linear relationships between variables. P-values were calculated for each pair of features to identify significant correlations. A heatmap was generated to visualize only significant correlations (with p-values less than 0.05). Figure 1

3.5.2 Pairplot

A scatterplot matrix (pairplot) was created to visualize relationships between the numerical features. This helped in detecting patterns and potential outliers in the dataset. Figure 2

3.6 Text Processing

Since the core of this classification task involves analyzing text data (tweets), we employed several text preprocessing techniques:

3.6.1 Denoising

The text was cleaned by removing unwanted characters such as URLs, HTML tags, punctuation, hashtags, and special characters using various utility methods such as `remove_html`, `remove_url`, and `remove_punctuation`.

3.6.2 Standardization

All text was converted to lowercase using the `standardize_text` method to avoid case-sensitive discrepancies.

3.6.3 Lemmatization

The `WordNetLemmatizer` was used to reduce words to their base forms (lemmas), making the text more uniform for the model while preserving meaning.

3.6.4 Stopword Removal

Common stopwords (e.g., “and”, “the”, “is”) and irrelevant words like “RT”, “like”, and “follow” were removed to focus on meaningful words.

3.6.5 Word Cloud Visualization

A word cloud was generated to visualize the most frequent words across all tweets, providing insight into the dominant terms associated with the gender categories. Figure 3

3.7 Feature Extraction

To convert the processed text data into a numerical format suitable for machine learning models, we employed two feature extraction techniques:

3.7.1 Bag of Words (BoW)

The text data was vectorized using `CountVectorizer`, transforming the text into a sparse matrix of token counts. This simple and effective method allowed us to represent the frequency of words in each tweet. The resulting dense array was stored in the dataset as the `bow_feature`. This Bow can be seen in added in this Figure 4.

3.7.2 Word2Vec Embeddings

Word2Vec embeddings were computed using the `Word2Vec` model from the Gensim library. This method captured semantic information by learning distributed vector representations of words. The average embedding for each tweet was calculated and stored as `word2vec_embeddings` in the dataset.

4 Task 3: Visualization

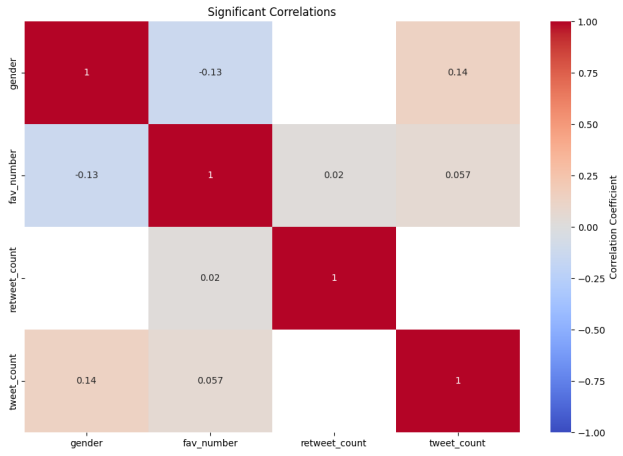

Figure 1: Correlation Matrix

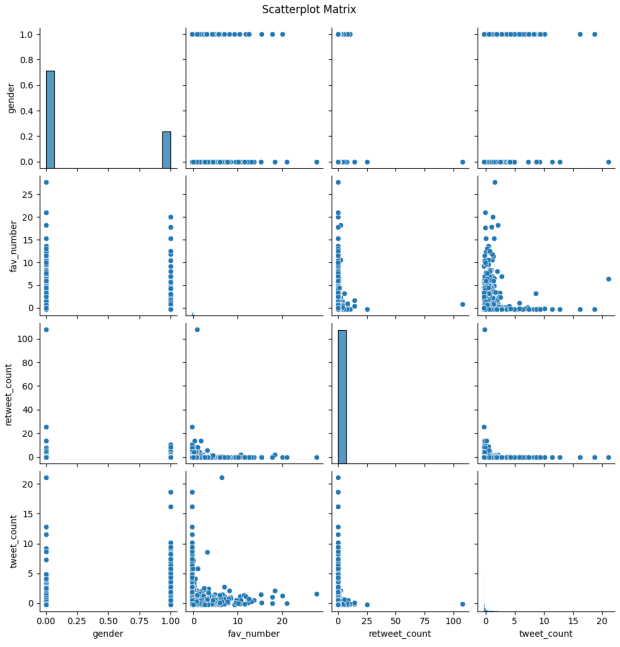

Figure 2: Pair Plot

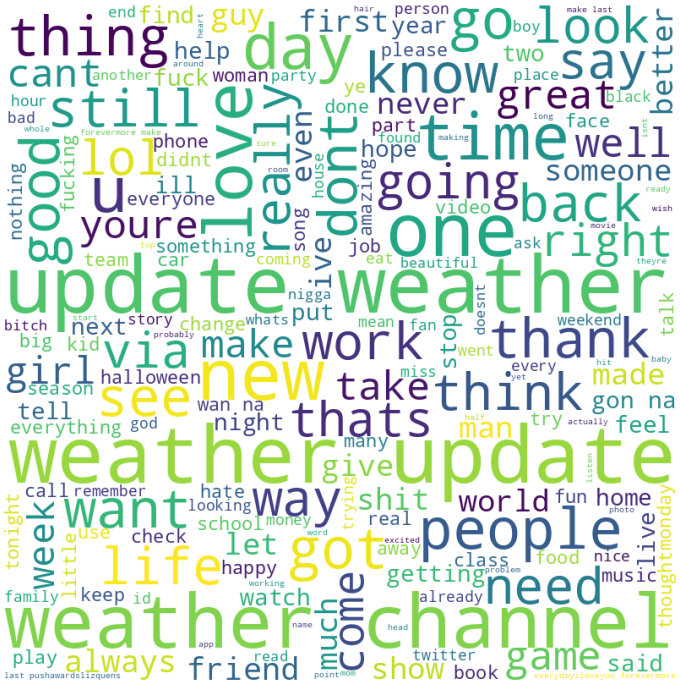

Figure 3: Word Cloud

	gender	fav_number	retweet_count	text	tweet_count	bow_feature	word2vec_embeddings
0	0	-0.348345	-0.047969	robbie e responds critic eddie edward worldit...	0.582729	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0.04739282, -0.010395167, 0.015648082, -0.004...
1	0	-0.342754	-0.047969	felt friend living story retired ian1	-0.244501	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0.2457285, -0.064767668, 0.07366451, -0.03446...
3	0	-0.331736	-0.047969	hi looking url use dont typically see advanced...	-0.290685	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0.4261804, -0.12715583, 0.11448994, -0.063192...
4	0	2.719920	-0.047969	watching neighbour sky catching neighbs xxx xxx	-0.052738	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0.07816511, -0.020627912, 0.027190495, -0.005...
5	0	-0.027607	-0.047969	ive seen people train lamp chair tv etc	-0.144067	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0.36296287, -0.10343567, 0.10271529, -0.05848...

Figure 4: Data Frame

References

Bishop, C. M. (2006). *Pattern recognition and machine learning.* Springer. Retrieved from `https://www.springer.com/gp/book/9780387310732`

Developers, N. (2024). *Numpy: The fundamental package for scientific computing with python.* Retrieved from `https://numpy.org`

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow.* O'Reilly Media. Retrieved from `https://www.oreilly.com/library/view/hands-on-machine/9781492032632/`

learn Developers, S. (2024). Scikit-learn user guide. *Scikit-learn Documentation*. Retrieved from `https://scikit-learn.org/stable/user_guide.html`

Team, P. D. (2024). *Pandas: Python data analysis library.* Retrieved from `https://pandas.pydata.org`

VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data.* O'Reilly Media. Retrieved from `https://jakevdp.github.io/PythonDataScienceHandbook/`