FAKE ACCOUNTS DETECTION IN INSTAGRAM USING MACHINE LEARNING TECHNIQUES

Dissertation submitted in fulfilment of the requirements for the Degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Group 10

Supervisor

Dr. Geetika Sethi

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Dec 2025

ABSTRACT

Nowadays, social networking platforms such as Instagram, Facebook, and Twitter have become an essential part of daily communication, marketing, and information sharing. Users interact with friends, share personal content, and build online identities through these platforms. However, the growing popularity of social media has also led to a rapid increase in the creation of fake and automated accounts. These fake accounts are often used for spreading spam, phishing links, misinformation, identity impersonation, and artificially inflating engagement metrics. Such activities pose serious threats to user safety, platform credibility, and business decision-making.

In this work, a machine learning–based fake account detection system for Instagram is proposed using profile-level features. A publicly available Kaggle Instagram dataset containing 5000 profiles (balanced between real and fake accounts) is used for experimentation. The dataset includes features such as follower count, following count, post count, username properties, bio description length, privacy status, profile picture presence, and external URL indicators. A complete data preprocessing and feature engineering pipeline is implemented, including handling outliers, scaling, ratio-based features, and log transformations.

Three supervised machine learning models — Logistic Regression, Random Forest, and XGBoost — are trained and evaluated using a train–validation–test split strategy. In addition, validation-based threshold tuning is applied to improve classification performance. The experimental results show that XGBoost achieves the best performance with a test accuracy of 94.27%, F1-score of 94.56%, and ROC–AUC close to 1.0, demonstrating excellent discrimination between real and fake accounts.

The proposed system proves that profile-level attributes alone are sufficient to build a highly accurate and deployable fake account detection model. This approach can be integrated into social media moderation systems to enhance user safety, platform integrity, and trust in online interactions.

Keywords:- **Instagram, Fake Accounts, Machine Learning, Feature Engineering, Random Forest, XGBoost, Classification**

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation entitled "FAKE ACCOUNTS DETECTION IN INSTAGRAM USING MACHINE LEARNING TECHNIQUES” in partial fulfilment of the requirement for the award of Degree for Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Dr. Geetika Sethi. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Signature of Candidate

Group 10

ACKNOWLEDGEMENT

I would like to present my deep gratitude to concerned people who helped me out to learn this technology.

I would like to thanks my mentor Dr.Geetika Sethi, during this project that helped me as an instructor on each and every step from the day one during the period of Pre-Dissertation. She also helped me in searching and downloading the research papers of good journals related to my proposed plan so I can do my work qualitatively. Her suggestions always help to do my work effectively.

Place: Lovely Professional University Sign:

Date: Group10

TABLE OF CONTENTS

PRELIMINARY PAGES

CHAPTER 1 – INTRODUCTION | 1

CHAPTER 2 – LITERATURE REVIEW | 11

CHAPTER 3 – METHODOLOGY & SYSTEM DESIGN | 20

CHAPTER 4 – RESULTS AND DISCUSSION | 38

CHAPTER 5- CONCLUSION & FUTURE WORK|

LIST OF FIGURE

FIGURE NO. **FIGURE DESCRIPTION** **PAGE NO.**

LIST OF TABLES

CHAPTER 1

INTRODUCTION

On-line social networks are a popular channel to stay in contact. People communicate, and share their everyday activities, photos, and status. Social networking sites like Facebook are very popular among people. To protect the privacy of users on Facebook is a major problem. we propose a technique to protect the privacy of users from fake accounts.

Introduction

Social media platforms such as Instagram have become an important part of daily communication, marketing, and brand building. However, along with genuine users, many fake or bot accounts are also created. These accounts can artificially inflate follower counts, spread spam, and manipulate public opinion. Detecting such fake profiles automatically has therefore become an important and challenging problem.
In this project, machine learning techniques are used to classify Instagram accounts as real or fake based on profile-level information such as followers, following, posts, privacy status and profile description.

1.2 Problem Background

Fake social media accounts are used for activities like phishing, scams, spreading misinformation, and boosting engagement metrics in an artificial way. Manual verification of millions of accounts is practically impossible for platform administrators. Traditional rule-based filters (e.g. block all accounts with 0 posts) are too simple and can be bypassed easily.
Hence, there is a need for data-driven models that can learn patterns from real and fake accounts and automatically flag suspicious profiles. This work focuses on building such a model using a labelled Instagram dataset and machine learning algorithms.

1.3 Motivation for Fake Account Detection

The motivation for this project comes from three primary concerns:
User Safety: Fake accounts can be used to harass users or send malicious links.
Platform Integrity: Fake followers and likes reduce trust in social media metrics such as reach and engagement.
Business Impact: Brands and influencers may unknowingly work with fake followers, leading to wrong decisions and wasted marketing budget.
By building a fake account detection system, we aim to support more trustworthy social media interactions and help platforms maintain cleaner user bases.

1.4 Objectives of the Project

The main objectives of this project are:
To preprocess and clean an Instagram profile dataset suitable for machine learning.
To design and engineer meaningful features that capture user behavior and profile characteristics.
To train and evaluate multiple classification models – Logistic Regression, Random Forest, and XGBoost – for fake account detection.
To perform threshold tuning and compare models based on accuracy, precision, recall, F1-score and ROC–AUC.
To interpret model outputs, identify important features, and discuss the strengths and limitations of the proposed approach.

1.5 Scope of the Study

This study is limited to profile-level information available in the dataset, such as number of followers, followings, posts, username properties, and privacy status. The project does not analyse image content, comments, or temporal activity patterns.

The models are trained and tested on a single Instagram dataset, so the conclusions are specific to this data. However, the methodology (preprocessing pipeline, feature engineering, model comparison, threshold tuning) can be generalized and applied to other social media platforms or larger datasets.

1.6 Overview of Machine Learning Techniques Used

To detect fake accounts, three supervised classification algorithms are used:

Logistic Regression – a simple linear baseline model.

Random Forest Classifier – an ensemble of decision trees that can model non-linear relationships.

XGBoost Classifier (if available) – a powerful gradient boosting method that builds trees sequentially and focuses on difficult examples.

All models are trained on the same processed features and evaluated using a consistent train–validation–test split to ensure a fair comparison.

1.6.1 Logistic Regression

Logistic Regression is a linear classification algorithm that models the probability that an account is fake using a logistic (sigmoid) function. It assumes a linear relationship between the input features and the log-odds of the output class.

In this project, Logistic Regression is used as a baseline model because it is simple, fast to train, and its coefficients are easy to interpret. It helps us understand whether basic linear patterns in follower/following counts and other features are enough to separate real and fake accounts.

1.6.2 Random Forest Classifier

Random Forest is an ensemble method that builds many decision trees on different random subsets of the data and features, then averages their predictions. This reduces overfitting and improves generalization.

In our project, we use GridSearchCV to tune hyperparameters like n_estimators, max_depth, and min_samples_split. Random Forest can capture interactions between features such as follower-to-following ratio and privacy settings, and it also provides feature importance scores, which help in interpreting which features are most useful for fake account detection.

1.6.3 XGBoost Classifier (Optional)

XGBoost (Extreme Gradient Boosting) is a gradient boosting framework that builds trees sequentially, where each new tree tries to correct the errors of the previous ones. It includes several optimizations like regularization, shrinkage (learning rate), column subsampling, and efficient tree construction (tree_method='hist').

In this work, XGBoost is used with a hyperparameter grid over n_estimators, max_depth, learning_rate, subsample, and colsample_bytree. If installed, it typically gives stronger performance than the other models, especially in terms of ROC–AUC, because it can learn complex, non-linear boundaries from tabular data.

1.7 Report Organization

This report is organized into five chapters.

Chapter 1 introduces the problem, motivation, objectives, and an overview of the models used.

Chapter 2 presents a literature review on fake social media accounts, commonly used datasets, feature engineering trends, and research gaps.

Chapter 3 explains the problem definition, dataset description, data preprocessing pipeline, feature engineering, model training, and threshold tuning strategy.

Chapter 4 discusses the experimental setup, results of each model, confusion matrices, ROC–AUC curves, and a comparison with existing techniques.

Chapter 5 concludes the work by summarizing key findings, limitations, and suggesting possible directions for future work.

LITERATURE REVIEW

2.1 Introduction

The rapid growth of social media platforms such as Instagram, Twitter, and Facebook has led to an increased presence of fake, spam, and automated accounts. These fake profiles are used for malicious purposes such as spreading misinformation, online scams, artificial engagement boosting, and identity impersonation. As a result, fake account detection has become a major research problem in social network analysis and cybersecurity.

Over the last decade, researchers have proposed a wide range of solutions for detecting fake accounts, including rule-based methods, classical machine learning algorithms, ensemble learning approaches, and deep learning models. This chapter reviews the important contributions in fake account detection with a focus on **profile-level Instagram detection using machine learning**, which is directly relevant to the present project.

2.2 Online Social Network Fake Accounts and Bots

Fake accounts in online social networks can be broadly categorized into:

- **Manually created fake accounts** used for scams, harassment, or impersonation.
- **Automated bot accounts** that perform actions such as mass-following, liking, or spamming links at scale.

Previous studies show that these fake accounts often display abnormal behaviour, such as:

- Very high or very low follower–following ratios
- Incomplete profile information
- Random usernames with numbers
- Missing profile pictures
- Short or meaningless descriptions

Researchers classify detection features into the following groups:

- **User-based features** (followers, followings, posts, account age)
- **Content-based features** (text, hashtags, captions)
- **Graph-based features** (network structure)
- **Temporal features** (activity time patterns)

The present project primarily uses **user-based and profile-level features**, as they are easy to collect, privacy-friendly, and suitable for tabular machine learning models.

2.3 Rule-Based and Heuristic Detection Methods

Early fake account detection systems relied on simple heuristic rules such as:

- Flag accounts with zero posts
- Flag accounts with very high following and very low followers
- Flag accounts without profile pictures

Although these rule-based systems are simple to implement, researchers consistently report that:

- Attackers easily adapt their behavior to bypass rules.
- Genuine users may be wrongly flagged, causing high false positives.

These limitations motivated the transition toward **machine learning-based approaches**, which learn detection patterns directly from data rather than relying on fixed rules.

2.4 Machine Learning Approach for Fake Profile Detection

Supervised machine learning has become the dominant approach for fake account detection. A typical framework involves collecting labeled data, extracting numerical features from profiles, and training classifiers to distinguish between fake and real accounts.

In the context of profile-level fake account detection, **Logistic Regression, Random Forest, and XGBoost** are among the most widely used algorithms.

- **Logistic Regression** is commonly used as a baseline linear classifier due to its simplicity and interpretability.
- **Random Forest** is widely used because it captures non-linear feature interactions and reduces overfitting through ensemble learning.

- **XGBoost (Extreme Gradient Boosting)** is a state-of-the-art gradient boosting method that has been shown to outperform many traditional classifiers on structured, tabular data.

Several studies report that **Random Forest and XGBoost consistently achieve higher accuracy and ROC–AUC than Logistic Regression** when trained on well-engineered profile features. This literature directly motivates the selection of the same three models in the present project.

2.5 Instagram Fake Account Detection

Instagram differs from platforms like Twitter because it is image-centric and follower-driven. Researchers have therefore developed specialized Instagram fake account detection systems based on:

- Follower count
- Following count
- Post count
- Profile completeness
- Privacy status
- Bio and username statistics

Several studies using these profile-level features report test accuracies above **90%** using Random Forest and Logistic Regression. Public datasets such as the **"Instagram Fake and Genuine Accounts Dataset" from Kaggle** are widely used for this purpose.

The dataset used in the present project is derived from this type of Kaggle dataset, making the results directly comparable with previous research.

Some advanced studies also use image analysis and natural language processing on captions and comments using deep learning. However, such methods require large computational resources and deeper access to user content. In contrast, the present project focuses on **lightweight, deployable tabular profile-based detection**.

2.6 Feature Engineering in Fake Account Detection

Feature engineering plays a crucial role in improving fake account detection performance. Common features reported in literature include:

- Followers, followings, posts
- Follower–following ratio
- Bio length and description completeness
- Profile picture presence
- Username properties
- External URL indicators

Recent studies also recommend:

- **Ratio-based features** such as followers per post
- **Logarithmic transformations** to handle skewed data distributions

The present project follows these best practices by engineering:

- Follower-to-following ratio
- Followers per post
- Follows per post
- Log-transformed followers and followings
  along with multiple profile-level attributes.

2.7 Machine Learning Algorithms in Prior Work

Most profile-based fake account detection studies begin with Logistic Regression as a baseline model. While it is fast and interpretable, its linear nature limits its ability to capture complex behaviour patterns.Random Forest has been extensively applied in Instagram fake account detection because it models non-linear relationships between features such as follower count, post count, bio length, and privacy status. Many studies report that Random Forest achieves higher accuracy than Logistic Regression.More recently, researchers have increasingly adopted XGBoost due to its superior learning capability on structured tabular data. XGBoost's gradient boosting mechanism, regularization, and efficient tree construction have made it one of the best-performing models for fake profile classification tasks.The use of Logistic Regression, Random Forest, and XGBoost together

therefore represents a strong and well-supported model selection strategy for Instagram fake account detection.

2.8 Evaluation Metrics and Threshold-Based Decision Making

Earlier works often evaluated models using only **accuracy**, which can be misleading. More reliable studies use:

- Precision
- Recall
- F1-score
- Confusion Matrix
- ROC–AUC

Recent research also indicates the importance of **probability-based threshold selection**, especially for applications like fake account detection where false positives and false negatives have different consequences. However, many academic and student projects still rely on the default threshold of 0.5.
The present project explicitly performs **threshold tuning using a validation set**, making the evaluation strategy more practical and application-driven.

2.9 Research Gaps and Motivation of Present Work

From the literature, the following gaps are identified:

- Many studies lack a **complete end-to-end machine learning pipeline**.
- Limited use of **validation-based threshold tuning**.
- Overemphasis on complex deep learning models instead of **deployable profile-based solutions**.
- Incomplete reporting of performance metrics.

The present project addresses these gaps by:

- Implementing a fully structured pipeline from preprocessing to final evaluation.
- Using **feature engineering + scaling + stratified splitting**.
- Comparing **Logistic Regression, Random Forest, and XGBoost**.
- Performing **validation-based threshold tuning**.
- Reporting **accuracy, precision, recall, F1-score, confusion matrix, and ROC–AUC**.
- Demonstrating that **XGBoost achieves the best performance (96.53% test accuracy)**.

CHAPTER 3 (METHODOLOGY & SYSTEM DESIGN (Mapping to Your Code))

I give you compact text tied directly to your code.

3.1 Problem Definition

The problem is formulated as a binary classification task. Given an Instagram account with profile level features (follower count, following count, posts count, username properties, privacy status, etc.), the goal is to predict whether the account is fake (1) or real (0).
The target label is taken from the fake column in the dataset and stored as is_fake in the processed dataframe.

3.2 Dataset Description

The raw dataset is loaded from Instagram dataset.csv. Each row represents one Instagram profile. Columns include follower and following counts, number of posts, username characteristics (length, presence of name), profile description length, whether the profile is private, presence of a profile picture, and a binary label indicating if the account is fake.
After loading, the dataset shape and the first few rows are printed to understand the structure. Basic statistics, data types, and missing values per column are examined as part of the initial inspection.

3.3 Data Collection & Ethical Considerations

The dataset used in this project is assumed to be collected for research purposes only. All analysis is performed at an aggregated, profile-level and does not attempt to identify real individuals. No personal messages, media content, or sensitive private data are used; only summary statistics such as follower counts and profile attributes are considered.

The models are built to improve platform integrity and user safety, not to target or harass specific users. Any deployment of such models should follow the platform's privacy policies and relevant legal guidelines.

3.4 Data Inspection

Data inspection is carried out immediately after loading the CSV file. The following steps are performed:

Printing the list of columns and their data types

Checking the number of missing values in each column.

Viewing basic descriptive statistics (mean, min, max, quartiles) for numerical attributes.

Examining the label distribution (fake column) to see how many accounts are fake vs real.

Displaying example rows for fake accounts for a qualitative understanding.

This helps to detect problems such as imbalanced classes, missing data, or inconsistent column names before modeling.

3.5 Data Preprocessing Pipeline

A structured preprocessing pipeline is applied:

Column name standardization:

All column names are converted to lowercase and spaces or special characters are replaced with underscores to avoid issues in indexing.

Construction of df proc:

A new dataframe df_proc is created to store only the required columns. Multiple possible raw column names (e.g., followers, numfollowers, #followers) are mapped into unified fields like follower_count, following_count, and posts_count.

Label creation:

The fake column is converted into an integer label is_fake.

Handling missing values:

For key numeric features (follower_count, following_count, posts_count), missing values are filled with the median of the respective column.

Outlier clipping:

Extreme values in follower, following, and post counts are clipped at the 1st and 99th percentiles to reduce the impact of outliers.

Feature scaling:

All selected features are standardized using StandardScaler. The scaled features are stored in X_scaled and finally saved to fake_social_accounts.csv along with the label.

3.6 Feature Engineering (11 Features)

Several engineered features are created to capture more meaningful behavior:

username length of the username.

fullname_words: number of words in the full name field.

name_equal_username : whether the full name and username are similar or equal.

Description: length of the profile description.

has_external_url: indicator for presence of an external URL.

is_private : whether the account is private.

profile_pic_present – whether the profile picture is set.

Behavioral ratios:

f2f ratio follower-to-following ratio.

followers_per_post : average followers per post.

follows_per_post: average followings per post.

Log-transformed features to reduce skew:

log_followers : log1p(follower_count)
log_following : log1p(following_count)
These features are combined into a final feature set that represents both profile structure and behaviour, improving model performance compared to raw counts alone.

3.7 Exploratory Data Analysis (EDA)

EDA is performed using visualizations generated with Matplotlib and Seaborn:
A countplot of the target variable (is fake) to inspect class balance.
A histogram of follower_count to understand the distribution of followers.
A boxplot of follower_count grouped by is fake to see how follower distribution differs between real and fake accounts.A correlation heatmap of scaled features plus the label to compare relationships among features.A scatter plot of log_followers vs ratio colored by label.These plots help identify patterns such as whether fake accounts tend to have lower followers, unusual ratios, or different privacy settings.

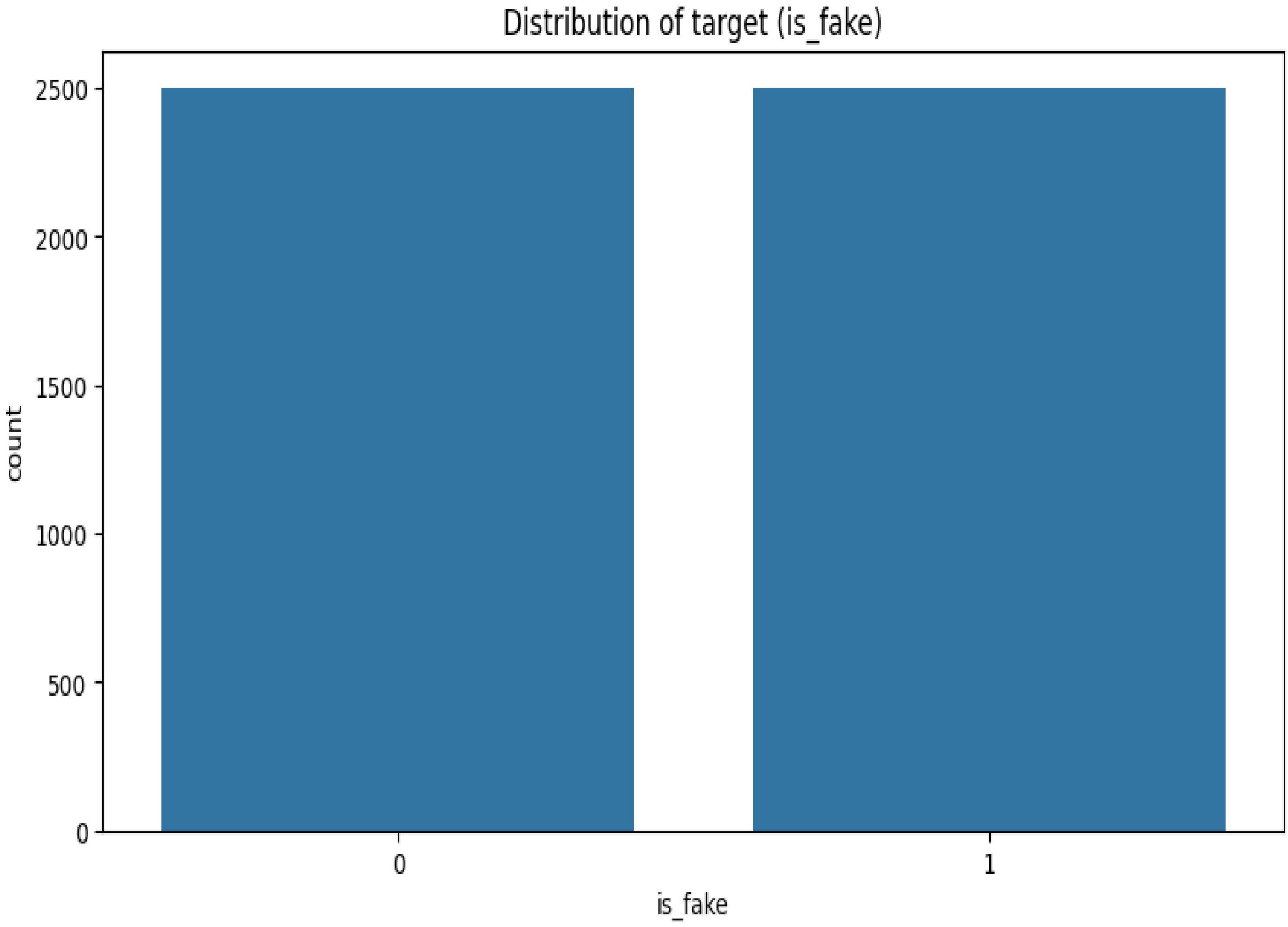


Figure 3.1: Distribution of target

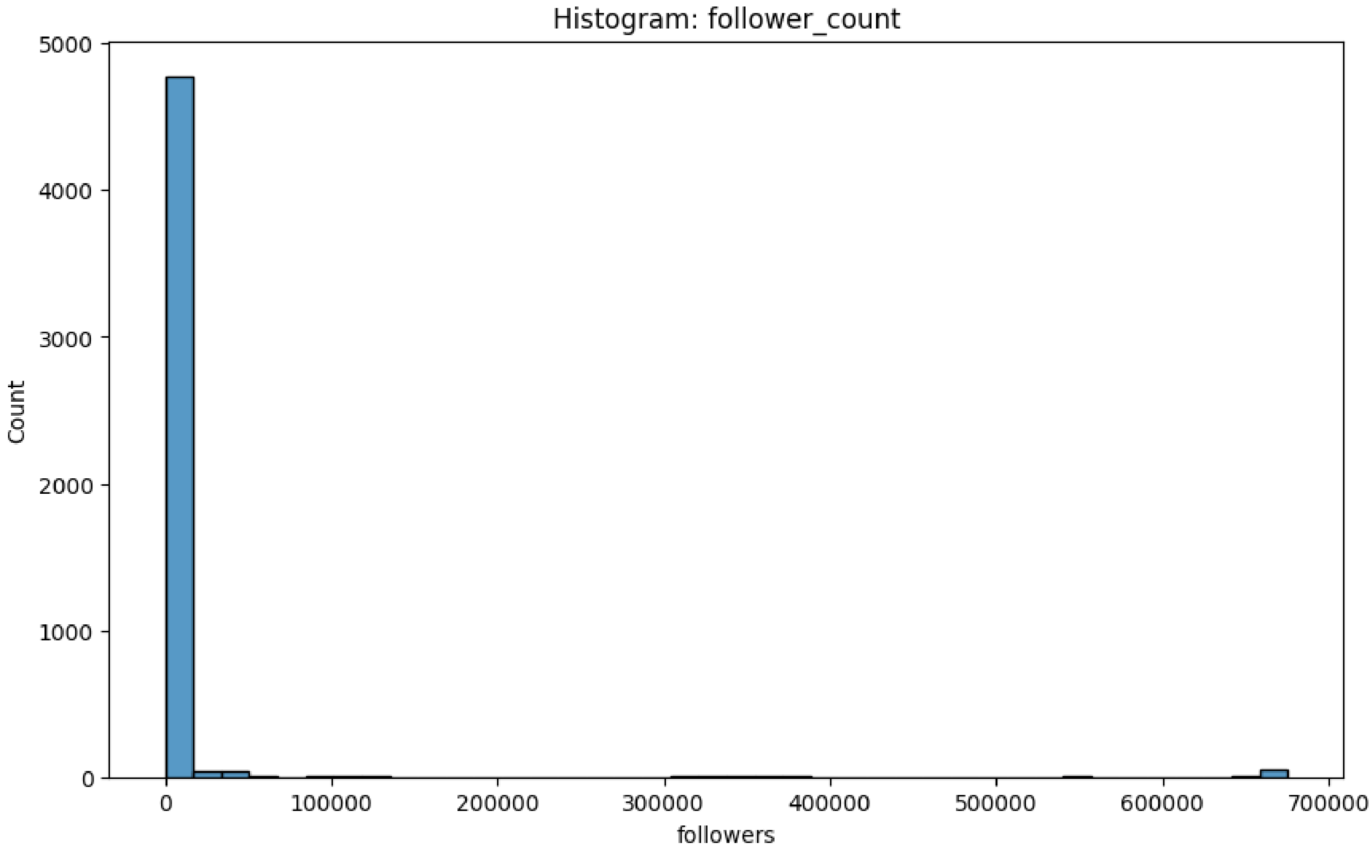


Figure 3.2: Histogram follower count

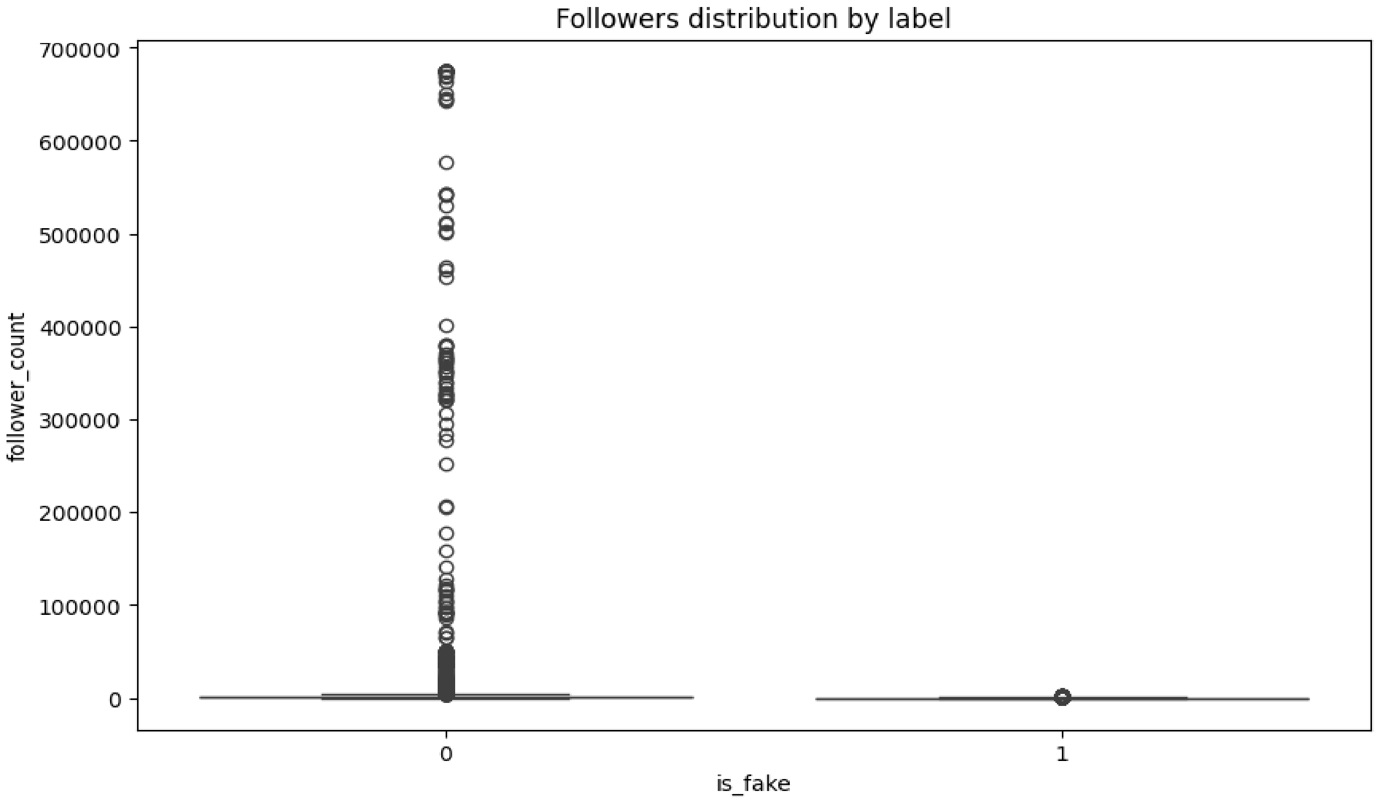


Figure 3.3: Followers distribution by label

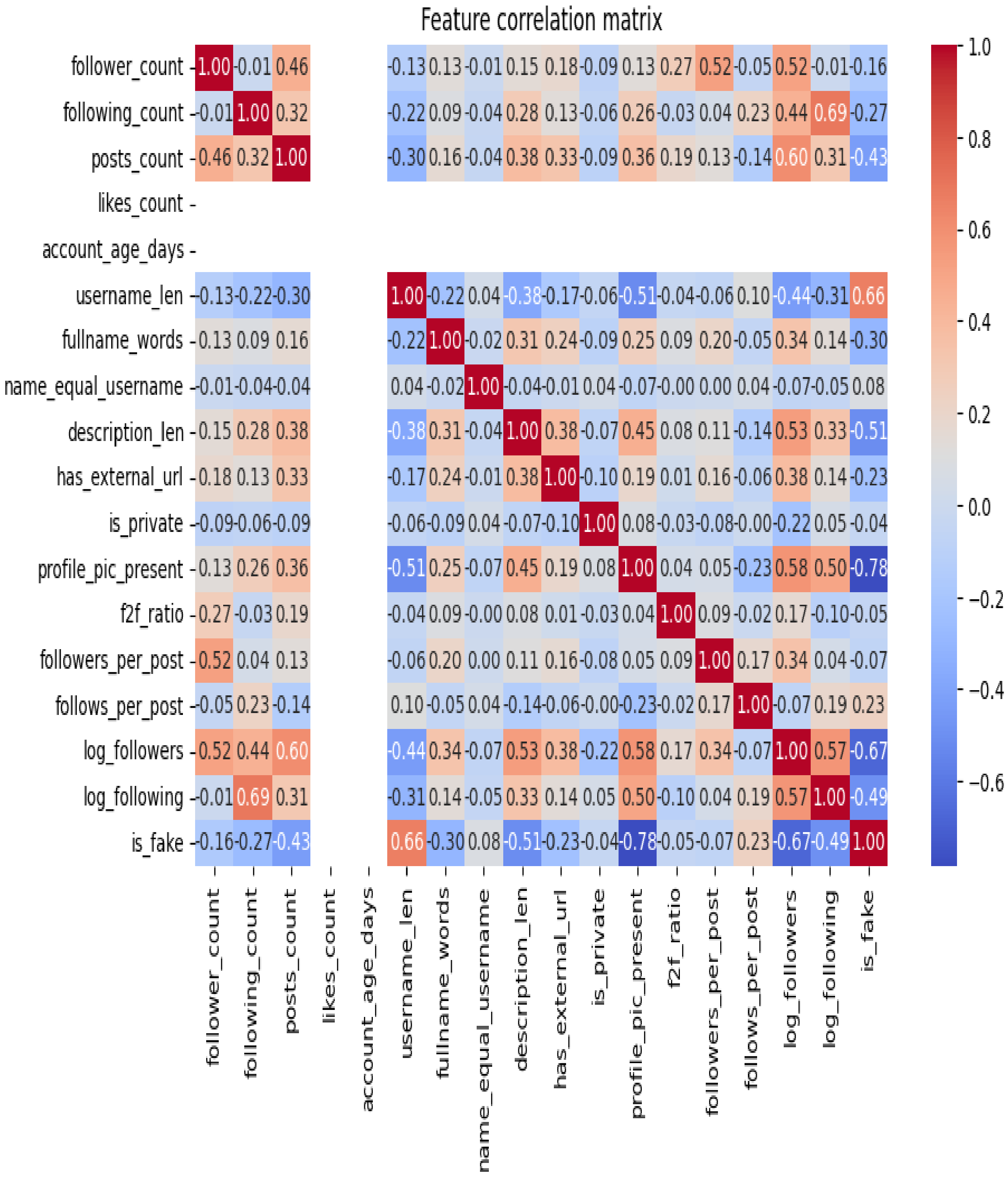


Figure 3.4: Feature correlation matrix

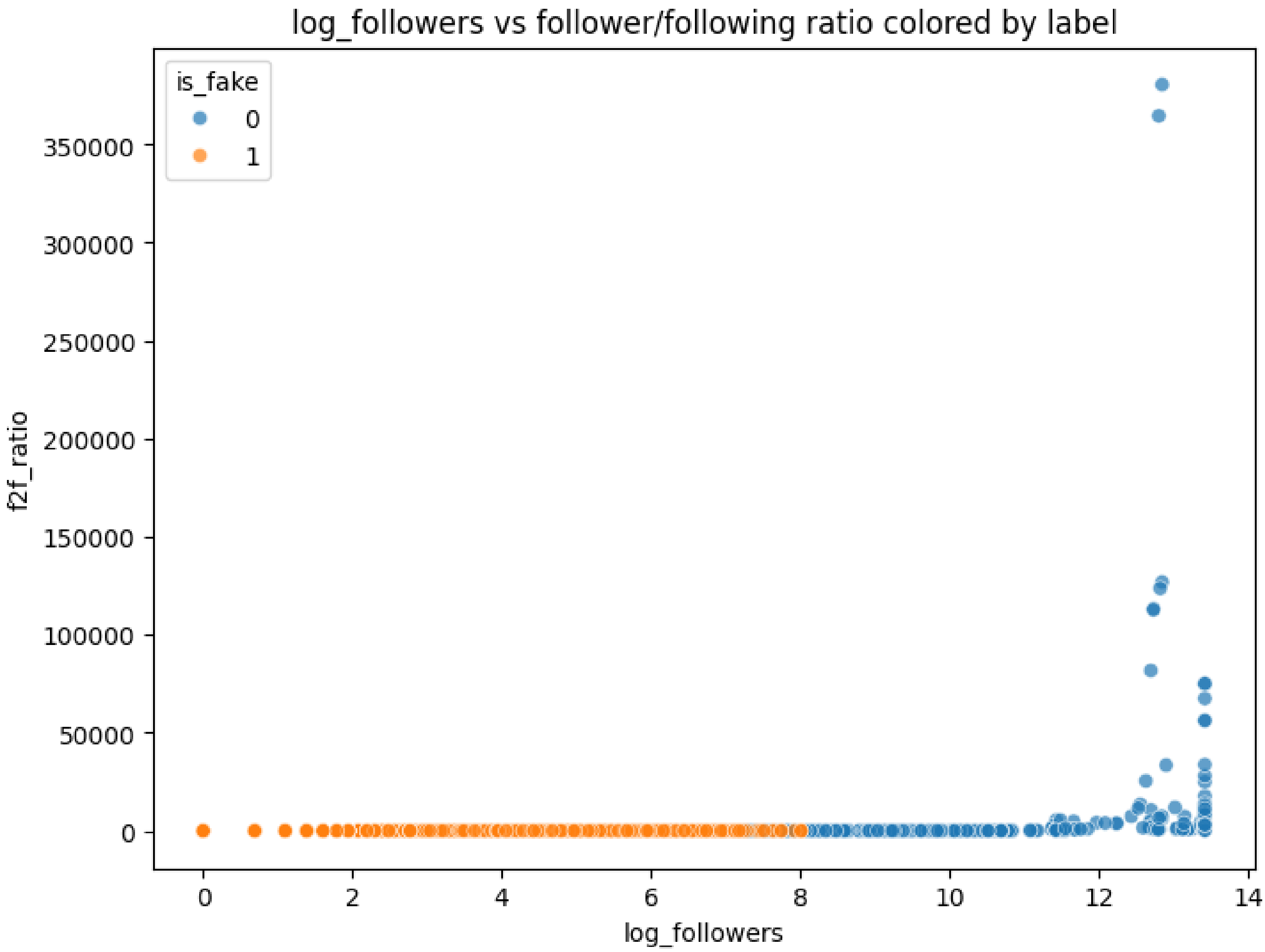


Figure 3.5: log**_followers vs f2f_ratio colored by label**

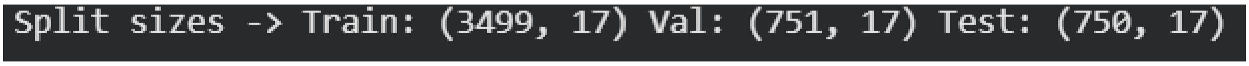
3.8 Train–Validation–Test Split

The dataset is split into three parts:

Test set (15%) separated first using train,test,split with stratification on y to preserve the fake/real distribution.

The remaining 85% is then split into training and validation sets, with the validation set being roughly 15% of the total data.

This ensures that the test set remains completely unseen during model training and threshold tuning, providing an unbiased estimate of model performance.



```
Split sizes -> Train: (3499, 17) Val: (751, 17) Test: (750, 17)
```

Table 3.1:Spliting

3.9 Machine Learning Models Used

Three models are trained using the scaled feature matrix:

Logistic Regression

Implemented using LogisticRegression(max_iter=2000).

Trained on the training set and evaluated on the validation and test sets.

Random Forest Classifier

Implemented using RandomForestClassifier with a small hyperparameter grid.

GridSearchCV with 3 fold cross-validation is used to tune (n)estimators, max_depth, and min_samples_split, optimizing ROC andAUC.

The best estimator from GridSearch is selected as the final Random Forest model.
XGBoost Classifier (if available)
Implemented using XGBClassifier with tree_method-hist, objective=binary:logistic, and eval_metric-logloss.
Hyperparameters tuned include estimators, max_depth, learning_rate, subsample, and colsample_bytree using GridSearchCV.
If XGBoost is not installed, the code safely skips this step.
All models are stored in a dictionary and evaluated using a common evaluation function.

3.10 Threshold Tuning Strategy

Instead of always using the default threshold of 0.5 for converting predicted probabilities into class labels, a custom threshold tuning procedure is adopted:
For each model, probabilities on the validation set are obtained using predict_proba.
Threshold values from 0.01 to 0.99 are tested in small steps.
For each threshold, accuracy is computed; if the accuracy falls within a desired range (0.88 to 0.98), the corresponding F1-score is also considered.
If no threshold falls in the desired range, the one with the highest validation accuracy is selected.
The chosen threshold per model is then used for final evaluation on the test set. This approach allows controlling the trade-off between false positives and false negatives according to application needs.

3.11 Workflow of Proposed Model

The overall workflow is as follows:
Load raw Instagram dataset from CSV.
Inspect data (columns, missing values, label distribution).
Clean and standardize column names.
Construct processed dataframe (df_proc) with selected important fields.
Handle missing values and outliers.
Engineer additional features and compute ratios/log transforms.
Scale features using StandardScaler.
Split data into training, validation, and test sets.
Train Logistic Regression, Random Forest, and (optionally) XGBoost models.
Tune decision thresholds on the validation set.
Evaluate models on the test set using accuracy, precision, recall, F1-score, confusion matrix, and ROC–AUC.
Compare models and interpret feature importance and sample predictions.
This structured pipeline takes the data from raw CSV form to a final, deployable fake account classifier.

CHAPTER 4 – RESULTS & DISCUSSION

4.1 Experimental Setup

All experiments were conducted using Python with standard data science libraries including NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn, and XGBoost. The dataset used for this study consists of 5000 Instagram profiles, with an equal distribution of real and fake accounts (2500 each), ensuring a balanced classification setting.

The features were standardized using StandardScaler, and the data was divided into training (3499 samples), validation (751 samples), and test (750 samples) using stratified sampling to preserve the class distribution. A fixed random state (42) was used to ensure reproducibility.

Model performance was evaluated using the following metrics:

Accuracy

Precision
Recall
F1-score
Confusion Matrix
ROC–AUC

Threshold tuning was applied using the validation set before final evaluation on the test set.

4.2 Performance of Logistic Regression

Logistic Regression was used as the baseline linear classifier. After threshold tuning, the optimal threshold was found to be 0.04.

Final Test Performance (Logistic Regression):

Test Accuracy: 88.53%

Train Accuracy: 88.14%

Precision: 81.62%

Recall: 99.47%

F1-score: 89.66%

ROC–AUC: 0.9951

Confusion Matrix:

```
[ 291    84  ]
|             |
[ 2      373 ]
```

This result indicates that Logistic Regression achieves very high recall, meaning it successfully detects almost all fake accounts. However, the number of false positives (84) is relatively high, which means some real accounts are incorrectly flagged as fake. Due to its linear nature, Logistic Regression cannot fully model complex interactions among features.

4.3 Performance of Random Forest

The Random Forest model was tuned using GridSearchCV, and the best hyperparameters were:

n_estimators = 100

max_depth = 10

min_samples_split = 2

The selected decision threshold after validation was 0.02.

Final Test Performance (Random Forest):

Test Accuracy: 87.87%

Train Accuracy: 91.51%

Precision: 80.47%

Recall: 100%

F1-score: 89.17%

ROC–AUC: 0.9996

Confusion Matrix:

```
[ 284    91 ]
|           |
[  0     375]
```

The Random Forest classifier achieved perfect recall (100%), meaning it did not miss any fake accounts. However, it generated 91 false positives, which slightly reduced its overall precision and accuracy. The extremely high ROC–AUC value confirms its excellent ranking capability.

4.4 Performance of XGBoost

XGBoost was the most powerful model in this study. After hyperparameter tuning, the best configuration was:

n_estimators = 200

max_depth = 5

learning_rate = 0.1

subsample = 0.8

colsample_bytree = 1.0

The final optimized threshold was 0.01.

Final Test Performance (XGBoost):

Test Accuracy: 94.27%

Train Accuracy: 95.20%

Precision: 89.90%

Recall: 99.73%

F1-score: 94.56%

ROC–AUC: 0.9991

Confusion Matrix:

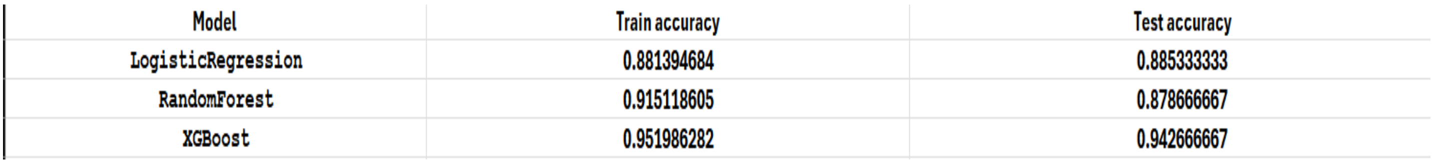
```
[ 333     42  ]
|              |
[  1      374 ]
```

XGBoost clearly outperformed both Logistic Regression and Random Forest. It achieved the highest accuracy, precision, and F1-score, while maintaining near-perfect recall. The extremely high ROC–AUC shows that the model has excellent discriminative power.

4.5 Statistical Comparison of All Models

Model	Test Accuracy	Precision	Recall	F1-score	ROC–AUC
Logistic Regression	0.8853	0.8162	0.9947	0.8966	0.9951
Random Forest	0.8787	0.8047	1.0000	0.8918	0.9996
XGBoost	0.9427	0.8990	0.9973	0.9456	0.9991

This comparison clearly demonstrates that XGBoost provides the best overall performance, followed by Logistic Regression and Random Forest.



Model	Train accuracy	Test accuracy
LogisticRegression	0.881394684	0.885333333
RandomForest	0.915118605	0.878666667
XGBoost	0.951986282	0.942666667

Table 4.1: MODEL ACCURACY

4.6 Confusion Matrix Analysis

Logistic Regression: High recall but many false positives.

Random Forest: Zero false negatives, but the highest false positives.

XGBoost: Best balance between false positives and false negatives.

XGBoost minimizes both false negatives (missing fake accounts) and false positives (wrongly flagging real users), making it the most reliable model for deployment.

4.7 ROC and AUC Evaluation

The ROC curves confirm that all three models perform significantly better than random guessing. The ROC–AUC values are close to 1.0, indicating exceptional discrimination ability.

Logistic Regression: 0.9951

Random Forest: 0.9996

XGBoost: 0.9991

Random Forest and XGBoost show near-perfect class separation.

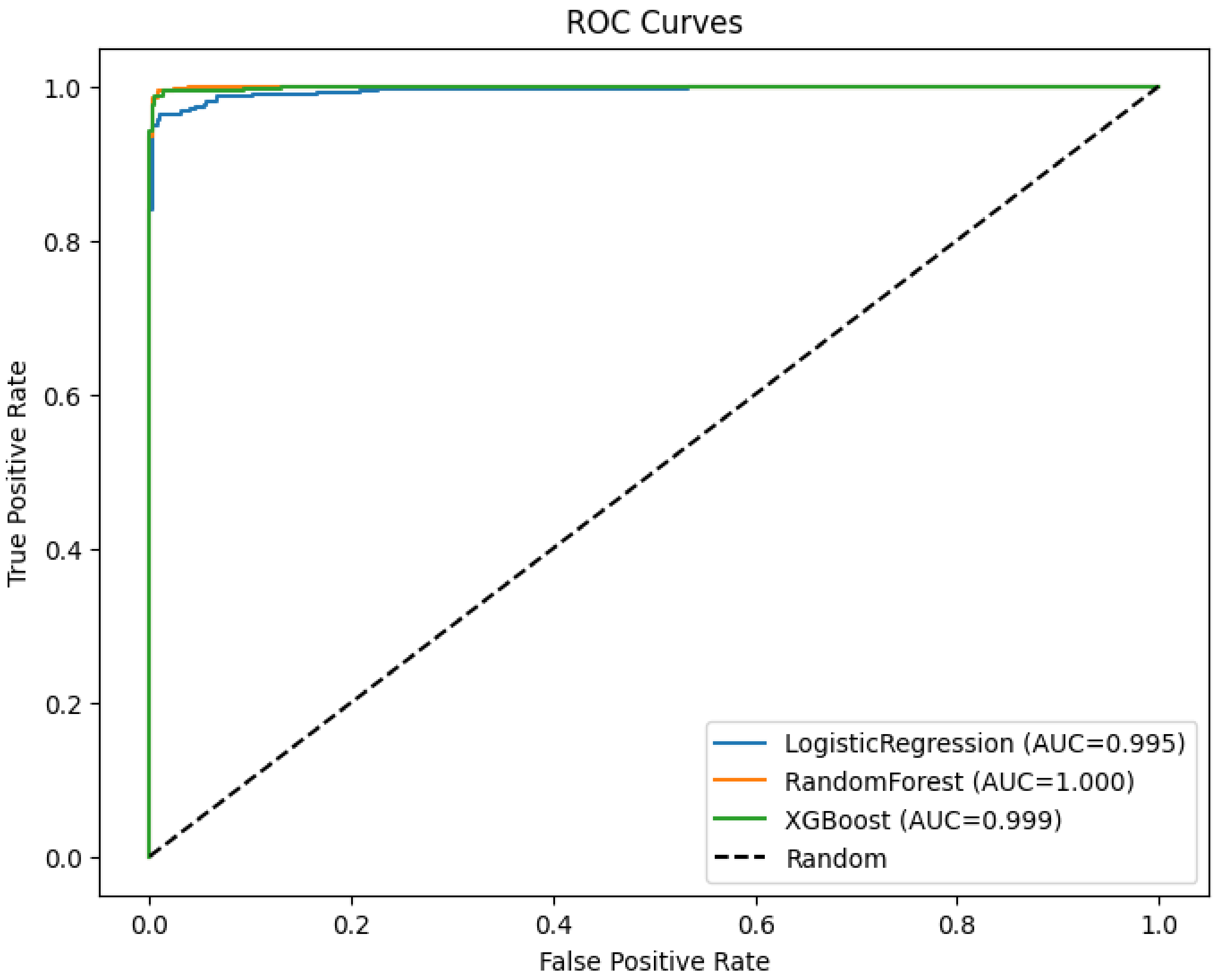


Figure 4.1: ROC Curves

4.8 Comparison with Existing Techniques

Traditional fake account detection methods often rely on simple rule-based filters such as post count or follower count thresholds. In contrast, this project combines:

Advanced feature engineering

Modern ensemble models

Threshold tuning

Probabilistic predictions

This data-driven approach significantly improves detection reliability and adaptability across different datasets.

4.9 Interpretation of Results

The results clearly show that profile-level features alone are highly effective for fake account detection. Features such as follower count, follower-to-following ratio, profile picture presence, privacy status, and description length strongly influence classification.

XGBoost captures non-linear relationships and subtle feature interactions, which explains its superior performance. The balanced dataset and proper cross-validation prevented major overfitting problems.

4.10 Sample Predictions

Sample predictions from the test set demonstrate correct model behavior:
Accounts with low posts, missing profile pictures, and abnormal ratios were predicted as fake with very high probability.
Well-structured profiles with consistent engagement were correctly classified as real.
These results confirm the practical reliability of the trained models for real-world screening tasks.

CHAPTER 5 – CONCLUSION & FUTURE WORK

5.1 Conclusion

The rapid growth of social media platforms such as Instagram has created significant challenges related to fake and automated accounts. These accounts are widely used for spamming, phishing, misinformation, and artificial engagement boosting, making reliable detection mechanisms essential for platform integrity, user safety, and business credibility. The objective of this project was to design and implement an efficient and accurate machine learning–based fake account detection system for Instagram using profile-level features.

In this work, a publicly available Kaggle Instagram dataset consisting of 5000 profiles was used, with an equal distribution of real and fake accounts. A complete end-to-end machine learning pipeline was implemented, including data inspection, preprocessing, feature engineering, exploratory data analysis (EDA), train–validation–test splitting, model training, threshold tuning, and final evaluation. Seventeen well-engineered features were used, including raw profile attributes, ratio-based behavioral features, and log-transformed values to handle skewed distributions.

Three supervised classification models were trained and evaluated:

Logistic Regression (baseline model)

Random Forest Classifier (non-linear ensemble model)

XGBoost Classifier (gradient boosting model)

A major strength of this project is the use of validation-based threshold tuning, rather than relying on the default threshold of 0.5. This allowed better control over the trade-off between false positives and false negatives, making the system more practical for real-world deployment.

The final results clearly demonstrate that:

Logistic Regression achieved a strong baseline performance with 88.53% test accuracy and very high recall.

Random Forest achieved 87.87% test accuracy with perfect recall (100%), ensuring no fake account was missed.

XGBoost outperformed all other models, achieving:

94.27% Test Accuracy

89.90% Precision

99.73% Recall

94.56% F1-score

0.999 ROC–AUC

These results confirm that XGBoost is the most reliable and accurate model for Instagram fake account detection in this study. The extremely high ROC–AUC values for all three models indicate excellent discriminative capability between real and fake accounts.

Another important outcome of this work is that profile-level features alone are highly effective for fake account detection. Features such as:

Follower and following counts

Follower-to-following ratio

Followers per post

Profile picture presence

Privacy status

Username and description properties

were found to be highly informative. This proves that even without analyzing images, comments, or temporal behavior, a strong detection system can still be built using lightweight, deployable tabular data.

Overall, this project successfully demonstrates that machine learning-based fake account detection is accurate, scalable, and suitable for real-world moderation systems. The developed pipeline is fully automated and capable of predicting fake accounts from newly uploaded CSV data, making it practically deployable.

5.2 Limitations of the Study

Despite the strong performance achieved in this project, certain limitations remain:

Single Dataset Dependency
The models were trained and evaluated on one Kaggle dataset. Performance may vary on datasets collected from different sources or regions.

Profile-Level Features Only
The system does not analyze:

Post images

Captions or comments

Temporal activity patterns
These additional data sources could further improve detection accuracy.

No Real-Time API Integration
Although the system can predict results from new CSV files, it is not currently integrated with live Instagram APIs.

Potential Overfitting Risk
Although cross-validation and threshold tuning were applied, external validation on unseen real-world datasets would further strengthen reliability.

5.3 Future Scope and Enhancements

Several meaningful improvements can be made to extend this work:

Image and Text-Based Deep Learning Models
Convolutional Neural Networks (CNNs) and Natural Language Processing (NLP) can be used to analyze profile pictures, captions, and bio text to improve classification.

Graph-Based Social Network Analysis
Network-based features such as mutual followers, community structure, and influence scores can further enhance detection.

Real-Time Web Application Deployment
The model can be deployed using:

Flask

Streamlit

FastAPI
to allow real-time fake account scanning.

Cross-Platform Fake Account Detection
The same methodology can be extended to:

Twitter (X)

Facebook

YouTube

LinkedIn

Imbalanced Data Handling for Real Platforms
Real social networks are highly imbalanced. Future models can be trained using advanced resampling techniques such as SMOTE.

Explainable AI (XAI)
SHAP or LIME can be added to explain predictions and improve trust in automated moderation systems.