Detection of Fake Spammer and Genuine Accounts on Instagram

Team Member: Ellen Dong / Mark Du / Winci Liang / Laurie Ye

TABLE OF CONTENTS

Problem & Significance

01

Problem: As artificial intelligence and technology continue to advance, the growth of social media platforms like Instagram has brought an increasing concern regarding the prevalence of fake and spam accounts.

- Distort engagement metrics
- Spread misinformation
- Degrade the overall user experience

Goal: Aim to develop a robust machine learning model to classify Instagram accounts as either authentic or spam based on various account features.

Outcome: Enhance the reliability of online interactions, the integrity of social media analytics, and mitigating the negative consequence of false online behavior. Foster a safe online community for users.

Domains: Social media platforms, marketing & advertising, and others

Dataset Selection

02

Dataset: Instagram Fake Spammer and Genuine Accounts

- Downloaded from Kaggle
- Data collected using a crawler from 15-19, March 2019
- Contains 12 variables, 696 observations (half spammer and half non-spammer)

Independent Variable	**Description**
Profile Pic	Indicates whether the user has a profile picture (1 = Yes, 0 = No)
Nums/Length Username	Ratio of the number of numerical characters in the username to its length
Fullname Words	Full name represented in word tokens
Nums/Length Fullname	Ratio of the number of numerical characters in the full name to its length
Name == Username	Indicates whether the username and full name are the same (1 = Yes, 0 = No)
Description Length	Length of the bio in characters
External URL	Indicates whether the account has an external URL (1 = Yes, 0 = No)
Private	Indicates whether the account is private (1 = Yes, 0 = No)
#Posts	Total number of posts made by the account
#Followers	Number of followers the account has
#Follows	Number of accounts the user follows
Dependent Variable	
Fake	Class indicating whether the account is genuine (0) or spam (1)

Methodology 03

Machine Learning Model Development

01

Regression-based

Logistic Regression:
- Baseline model for binary classification
- Linear and high interpretability

02

Tree-based

Decision Trees:
- Non-linear method that splits data based on feature value
- Easy interpretability but can overfit

Random Forest:
- Ensemble method, reducing overfitting, improving generalizability

03

Boosting Method

Gradient Boosting Machines (GBM), XGBoost and AdaBoost:
- More powerful ensemble method, capturing more nuances

04

Support Vector Method

Support Vector Machines (SVM):
- More powerful, non-linear model, effective in high-dimensional data
- Find the optimal hyperplane for classification

05

Neural Network

Sequential Neural Network:
- Most complex, captures highly intricate patterns, least interpretable

Evaluation Strategies

Feature Importance Analysis

Regularization Techniques

Lasso (L1) Regularization:

- Help prevent overfitting
- Improves model interpretability

Ridge (L2) Regularization:

- Reduce model complexity

Elastic Net:

- Combined Lasso and Ridge

Cross-Validation & Hyperparameter Tuning

Use Cross-Validation for:

- Hyperparameter tuning
- Further feature engineering
- Employ bootstrap methods

-> identify the best prediction model

Model Evaluation

Assess model performance using:

- Accuracy
- Precision
- Recall
- F1-Score
- ROC-AUC

Limitations

04

Reasons & Limitation

Reasonable Approaches to Consider

- **Diverse Methodologies:** combination of regression, tree-based methods, boosting techniques, and neural networks allows for capturing a wide range of patterns in the data -> improving classification accuracy.
- **Flexibility and Robustness:**
  - Regularization: prevent overfitting and enhances interpretability.
  - Cross-Validation: validates model on unseen data, to reduce overfitting and improving generalization.
  - Ensemble Methods: Techniques in gradient boosting improve performance by combining multiple weak learners for error correction.

Limitations

- **Small Dataset Size:**
  With only around 700 instances, the model may struggle to generalize effectively to unseen data.
- **Risk of Overfitting:**
  Even with regularization, complex models (like deep neural networks) may still overfit the training data if not managed properly.
- **Feature Sensitivity**:
  Algorithms like Decision Trees are sensitive to irrelevant features, necessitating thorough feature selection. Tree-based methods may misinterpret correlations between features, affecting accuracy.

Timeline

Time: Approximate 6 weeks until full presentation

Task	Duration	Team Member
Data Preprocessing and EDA	**2 weeks**	**4**
Model Development	**2 weeks**	**2**
Model Evaluation and Optimization	**1 week**	**2 / 3**
Report Writing and Presentation	**1 week**	**4**

Thank you!