CS361: Enhancing Fraud Detection on Social Media - A Machine Learning Approach

Lavanya Koleti - 210101057 Vanitha Nagapuri - 210101071 Harshavardhani Thota - 210101105
Nikhitha Vanga - 210101107

Abstract

With the exponential growth of online marketplaces, the need for effective fraud detection mechanisms has become paramount. This project focuses on enhancing the security and trustworthiness of online transactions by employing the Random Forest, KNN, and Naive Bayes algorithms to detect fraudulent sellers. In this comprehensive study, we will meticulously compare the performance of these models, assessing their effectiveness in identifying fraudulent activities within online marketplaces. Through rigorous evaluation of performance metrics such as accuracy, precision, recall, and F1-score, we aim to determine the most robust and reliable model or combination of models for fraud detection. By conducting extensive experimentation and analysis, we endeavor to unveil valuable insights into the strengths and weaknesses of each algorithm, thereby guiding the development of more effective fraud detection mechanisms. This comparative study promises to contribute significantly to enhancing the security and trustworthiness of online transactions, safeguarding the interests of consumers and businesses alike in the rapidly evolving digital marketplace landscape.

1. Introduction

In the dynamic landscape of digital commerce, Instagram has emerged as a pivotal platform for businesses, fostering a diverse marketplace. However, it faces a pressing challenge: identifying and mitigating fraudulent practices, particularly among sellers. The prevalence of fraudulent sellers jeopardizes consumer interests and undermines platform integrity.

Motivated by concerns surrounding fraudulent activities, this project aims to enhance Instagram's security and user trust. The primary objective is to deploy a robust fraud detection system capable of accurately identifying fraudulent sellers. Leveraging advanced machine learning techniques, the project contributes to creating a safer online marketplace.

Through a systematic approach, this endeavor addresses evolving challenges posed by fraudulent practices. By strategically applying machine learning methodologies, the project aims to fortify the online environment against fraudulent activities, enhancing the foundation of digital commerce.

2. Problem Formulation

The proliferation of online marketplaces has significantly transformed consumer transactions, yet the escalating presence of fraudulent sellers on platforms like Instagram presents a formidable challenge to user trust and security. This problem endeavors to devise robust methodologies tailored for the detection and mitigation of fraudulent activities perpetrated by sellers within Instagram's social marketplace.

Objectives encompass the formulation of algorithms adept at discerning aberrant seller behaviors, harnessing user-generated content to evaluate seller reliability, and delving into identity verification protocols. Leveraging historical transaction data, markers of potential fraudulence such as extended periods of inactivity, sparse follower counts, or novice account status are scrutinized. Machine learning models, including RandomForest, Naive Bayes Classification, and KNN, are employed to analyze these parameters and predict the legitimacy of Instagram accounts posing as sellers.

Despite the myriad approaches available for detecting fraudulent transactions or websites, the scarcity of mechanisms tailored to identifying deceptive Instagram accounts coupled with limited data availability poses a formidable obstacle to effective intervention in this domain.

3. Data Analysis

In this study, the dataset that we have utilized for data analysis is sourced from Kaggle.(Source)

3.1. Data Exploration

- Comprehensive exploratory data analysis (EDA) was conducted to gain insights into the distribution and characteristics of the dataset. Visualizations such as Count plots, Box plots, and Histograms over various features in dataset were utilized to uncover patterns, identify outliers, and understand the underlying structure of the data.
- A correlation matrix was constructed to examine the relationships between different variables in the dataset. This analysis provided valuable insights into the strength of relationship between variables aiding in feature selection and model interpretation.

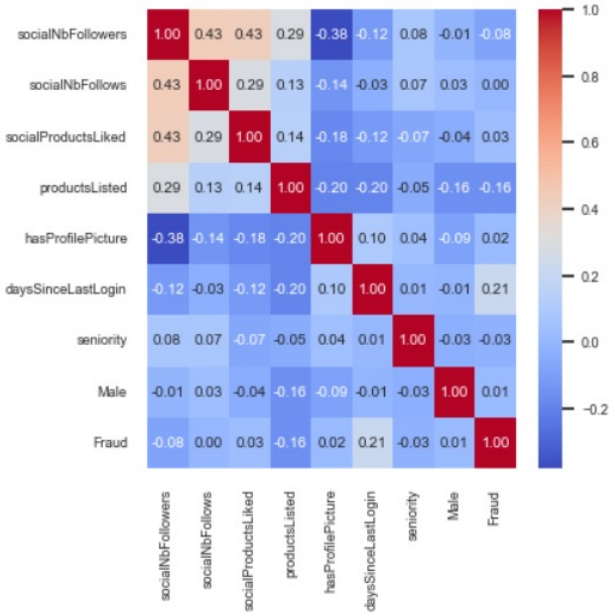


Figure 1. Correlation Matrix

3.2. Data Preprocessing

- Handling Null Values:

  Null values within the dataset were identified and effectively managed. Strategies such as imputation or removal were employed to mitigate the impact of missing data on subsequent analyses, ensuring robustness and reliability.
- Label Encoding:

  Label encoding techniques were applied to transform categorical variables into numerical representations suitable for machine learning algorithms. This process involved strategies such as one-hot encoding in which categorical variables are changed to integers ensuring that categorical data could be effectively utilized in predictive modeling tasks.
- Handling Outliers:

  Outliers that are observed during graphical analysis were removed form the dataset using strategies like z-score calculation to ensure that they donot cause any imbalance while applying algorithms.

3.3. Feature Engineering

Using feature engineering we transformed raw data into a format that is suitable for machine learning algorithms. We mainly used two techniques

3.3.1. Interaction of Features:

Interaction features involve combining two or more existing features to create new features that capture interactions between them. This helped us capture relationships that individual features may not capture on their own, potentially improving the predictive power of the model.

3.3.2. Polynomial Featuring:

Polynomial featuring involves creating new features by raising existing features to higher powers. This can help capture non-linear relationships between features and the target variable, allowing the model to better fit complex patterns in the data.

4. Methodology

4.1. Models

We chose four models: Naive Bayes, K-Nearest Neighbors (KNN), Random Forest, and Logistic Regression, and implemented custom code for each model to tailor them to our needs.

4.1.1. Naive Bayes

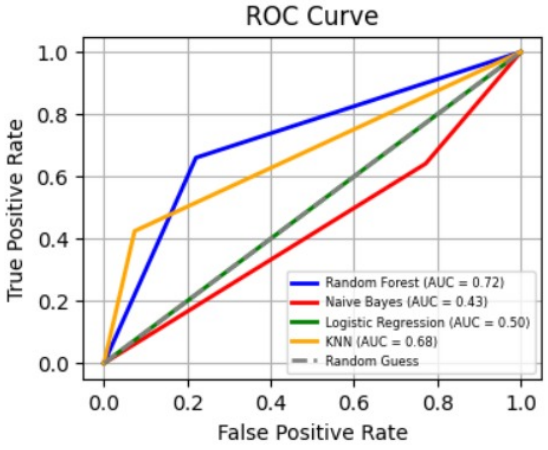
Naive Bayes is a probabilistic classification model based on Bayes' Theorem. It assumes that the features are conditionally independent given the class label, which simplifies the probability estimation and allows for efficient computation.

4.1.2. Logistic Regression

Logistic Regression is a linear model that estimates probabilities for classification tasks. It is widely used for binary classification, provides interpretability of model coefficients, and is suitable for datasets with linear relationships between features and the target variable.

4.1.3. Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees and combines their predictions to make a final decision. It offers high accuracy and



robustness by reducing overfitting and improving generalization.

4.1.4. K-Nearest Neighbours

KNN is a non-parametric classification algorithm that classifies new instances based on the majority vote of their k nearest neighbors. It is easy to understand and implement, making it a versatile choice for various classification tasks.

4.2. Hyperparameter Tuning

Optimizing hyperparameters is crucial for improving the performance of machine learning models. We used three common hyperparameter tuning methods:

4.2.1. Grid Search

Grid search is a brute-force approach where a predefined grid of hyperparameter values is explored exhaustively. The model is trained and evaluated for every combination of hyperparameters, and the best combination is selected based on performance metrics such as accuracy.

4.2.2. Random Search

Random Search works by randomly sampling combinations of hyperparameters from a defined search space and evaluating the model's performance for each combination. This approach is efficient for exploring large search spaces and can quickly find good parameter settings.

4.2.3. Bayesian Optimization

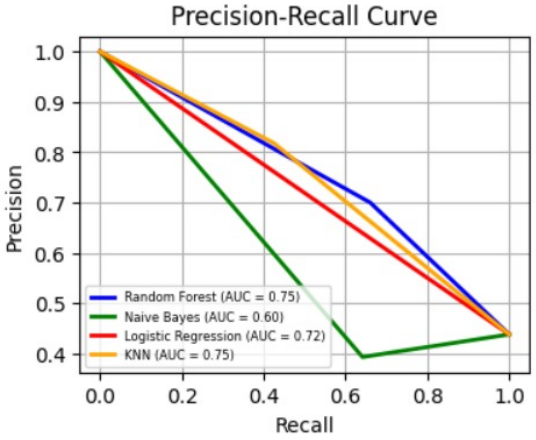
Bayesian optimization is an advanced method that builds a probabilistic model (usually Gaussian process) of the hyperparameter space. It uses this model to predict which combination of hyperparameters is likely to yield the best results, then iteratively selects the most promising combinations to evaluate.

4.3. Cross-Validation

Cross-validation is a technique used in machine learning to evaluate the performance and generalization of a model by testing it on different subsets of the data.

4.3.1. K-Fold Cross-Validation

The dataset is divided into k subsets (or folds). The model is trained on k-1 folds and tested on the remaining fold. This process is repeated k times, with each fold serving as the test set once. The final performance metric is calculated as the average of all test fold scores. It helps select an appropriate model and avoid overfitting, ensuring the model's robustness and accuracy on unseen data.



5. Model Improvement

To enhance model performance, we focus on three key strategies: feature selection, regularization, and dimensionality reduction.

5.1. Feature Selection

Feature selection is a crucial step in the data preprocessing and machine learning workflow. It involves selecting a subset of relevant features (variables, predictors) for use in model construction.

5.1.1. Filter Methods

These methods evaluate the relevance of features based on statistical measures and filter out irrelevant features before model training. Common measures include correlation, mutual information, and chi-squared tests. Filter methods are fast and simple but do not consider the interaction between features.

- Correlation:

  Measures the linear relationship between features and target variable. Features with high correlation with the

target are selected.

- Mutual Information:

  Measures the dependency between features and the target variable. Features with high mutual information are selected.

5.1.2. Wrapper Methods

These methods use a predictive model to evaluate the performance of different feature subsets. Features are selected based on the model's performance (e.g., accuracy, precision). Wrapper methods can be computationally expensive due to multiple model trainings.

- Recursive Feature Elimination:

  Starts with all features and recursively removes the least important features based on model performance.

- Forward/Backward Selection:

  Forward selection starts with an empty set and adds features incrementally. Backward selection starts with all features and removes the least important ones.

5.1.3. Embedded Methods:

These methods perform feature selection as part of the model training process. The model has a built-in mechanism to penalize the inclusion of irrelevant features.

- Lasso (L1 regularization):

  A linear model that penalizes non-zero coefficients, effectively setting some of them to zero, thus performing feature selection.

5.2. Regularization

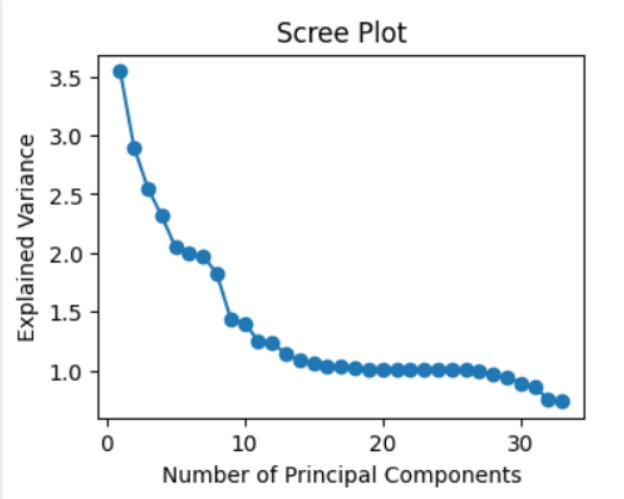
Regularization is a technique used in machine learning to prevent overfitting and improve the generalization of a model. By introducing a penalty term to the loss function, regularization discourages overly complex models that may perform well on the training data but poorly on unseen data. There are two main types of regularization:

5.2.1. Lasso (L1) Regularization

Lasso regularization adds a penalty proportional to the absolute value of the model coefficients. This technique can lead to sparsity in the model by setting some coefficients to zero, effectively performing feature selection.

5.2.2. Ridge (L2) Regularization

Ridge regularization adds a penalty proportional to the square of the model coefficients. This technique does not set coefficients to zero but instead shrinks them, allowing for better handling of multicollinearity and improving the stability of the model.



5.3. Dimensionality Reduction

Dimensionality reduction aims to reduce the number of features in a dataset while preserving as much information as possible.

5.3.1. Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique used for transforming high-dimensional data into a lower-dimensional space while retaining as much of the original data's variance as possible. It works by identifying and projecting the data onto the directions (principal components) where the variance is maximized. This helps simplify the data and reduce noise, making it easier to visualize and analyze. PCA is useful for preprocessing data before applying machine learning models, as it can lead to improved model performance and reduced computation time.

6. Comparitive Study

6.1. Hyperparameters

The best hyperparameters were determined using Bayesian optimization, yielding an optimal set of values for our models. The best hyperparameters identified during this process:

1. Logistic Regression: $C = 3.746$
2. GaussianNB: priors $= 0.375$
3. Random Forest: n_estimators $= 144$, max_depth $= 19$
4. KNN: n_neighbors $= 2$, weights $=$ distance

6.2. PCA

PCA resulted in a significant improvement in model accuracy for both Naive Bayes and Logistic Regression. PCA does not significantly impact the performance of KNN and Random Forest models. KNN relies on distances between

Model	Condition	Accuracy
Naive Bayes	Without PCA	0.4091
Naive Bayes	With PCA	0.5620
Logistic Regression	Without PCA	0.5868
Logistic Regression	With PCA	0.5950

data points, and Random Forest's ensemble learning handles high-dimensional data well without the need for PCA.

6.3. Regularization

Regularization is significantly improving model performance by preventing overfitting and enhancing generalization capabilities. It helps the models achieve better accuracy on both training and testing data.

Table 1. Logistic Regression

PCA	Regularization	Accuracy
Yes	L1	0.5992
Yes	No	0.5950
No	L2	0.5909
No	No	0.5868

Regularization may have limited impact on the remaining models (Naive Bayes, KNN, and Random Forest) due to their inherent characteristics and built-in mechanisms such as probabilistic estimations in Naive Bayes and feature subsets and tree depth control in Random Forest. These models may naturally exhibit resilience to overfitting, thus minimizing the effect of additional regularization.

6.4. Feature Selection

In the feature selection section, filter methods improve the accuracy of the Naive Bayes model by selecting relevant features, while Recursive Feature Elimination (RFE) increases accuracy in the Logistic Regression model by iteratively retaining the best features.

Model	Feature Selection Method	Accuracy
Naive Bayes	Without Filter Methods	0.4091
Naive Bayes	With Filter Methods	0.4256
Logistic Regression	Without RFE	0.5868
Logistic Regression	With RFE	0.5909

Feature selection techniques did not significantly impact the performance of K-Nearest Neighbors (KNN) and Random Forest models. These models handle feature selection internally: KNN uses distance metrics considering all features, while Random Forest leverages its intrinsic feature importance measure. Thus, external feature selection methods offer little performance improvement.

7. Results

7.1. Cross Validation

Model	**Cross-validated mean accuracy**
Naive Bayes Model	0.5303
KNN Model	0.4981
Random Forest Model	0.4186
Logistic Regression Model	0.5519

Based on the cross-validation values, we can state that logistic regression and Naive Bayes models perform better compared to KNN and Random Forest models in terms of mean accuracy.

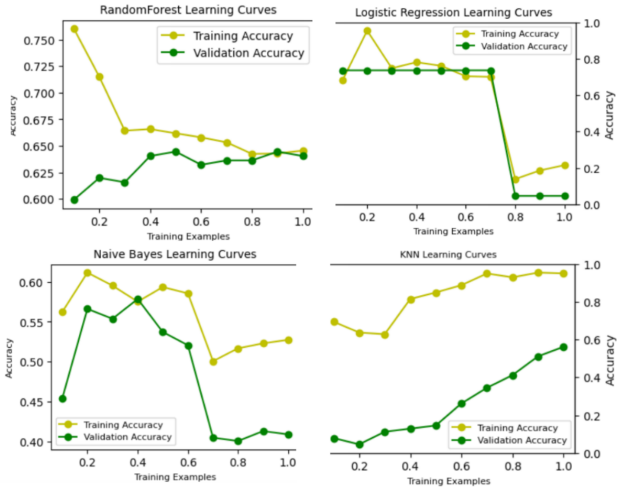


Figure 2. Learning curves

7.2. Model Learning Curves Analysis

The following are the observations from the learning curves shown in Figure 2:

7.2.1. K-Nearest Neighbors

The learning curves show a large gap between training and validation accuracy, indicating overfitting. While both accuracies increase, the model is memorizing the training data but struggles to generalize well to new data.

7.2.2. Random Forest

The curves show low training and validation accuracy that converge when all training examples are used. This suggests underfitting, as the model may not be capturing enough complexity. The small gap between accuracies is positive, but overall levels suggest room for improvement.

7.2.3. NAIVE BAYES

The model achieves above 50% accuracy for both sets, which is promising. However, the large gap between training and validation accuracies after 70% of the training examples suggests overfitting or issues with generalization.

7.2.4. LOGISTIC REGRESSION

The learning curves initially show high accuracy for both training and validation sets, indicating good performance and generalization. However, the decrease in both accuracies after 80% of the training examples suggests potential overfitting or a need for model tuning.

8. Major Challenges

8.1. Feature Selection

Initially, our dataset consisted of 21 variables . However, it became apparent that many of these variables were irrelevant or redundant for our predictive model. Consequently, we faced the challenge of determining the most informative and significant features among the initial set.

8.2. Feature Engineering Complexity

While feature engineering is complex, particularly in the context of social media data. Extracting meaningful features from user-generated content and account behaviors requires careful consideration and domain expertise.

8.3. Handling Imbalanced Data

Fraudulent activities often represent a minority class within the dataset, leading to class imbalance.

9. Future Scope

9.1. Dynamic Data Integration

Integrating real-time data can boost fraud detection, adapting to evolving tactics and market dynamics.

9.2. Advanced Model Architectures

Exploring advanced natural language processing (NLP) methods and investigating more complex deep learning architectures, such as recurrent neural networks (RNNs) or transformers may better capture sequential and contextual information present in social media data, thus improving fraud detection accuracy.

References

Source Codes.

Data Reference collected from Kaggle.

A comparative study on fake job post prediction using different machine learning techniques. Technical report, Sathyabama Institute of Science and Technology. (link to reference).

Alpaydın, E. *Introduction to Machine Learning*. MIT Press, Cambridge, MA, fourth edition, 2019.

Dr.Padmalatha. E-commerce frauds and the role of fraud detection tools in managing therisks associated with the frauds. Technical report. (link to reference).

Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*. Wiley, 2001.