Job Fraud Prediction: EDA & Modeling

MINOR PROJECT REPORT SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR IN TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING WITH SPECILIZATION (AI&ML)

NAME OF THE STUDENT:
Saikat Maity
Anas rahaman Molla
Arindam Pakhira
Shovan Bera
Ritaban Guchait

ROLL NO:
TNU2022053100038L
TNU2021053100033
TNU2021053100027
TNU2021053100030
TNU2021053100020

UNDER THE SUPERVISION OF
Dr. Usha Rani Gogoi
TEACHING ASSISTANT, Dept. of Computer Science Eng., TNU

Dept. of Computer Science of Engineering
THE NEOTIA UNIVERSITY, WEST BENGAL, INDIA
DECEMBAR, 2024

Certificate

We hereby recommend that the Project entitled ***"*Job Fraud Prediction: EDA & Modeling *"*** worked under our guidance may please be accepted in the partial fulfillment of the requirement for the degree of "Bachelor in Technology" in the Computer Science and Engineering with specialization in Data Analytics of „The Neotia University‟. The project report in our opinion is worthy for its acceptance. During the work „ **SAIKAT MAITY**‟ was found to be sincere, regular and hardworking and has successfully completed the thesis work assigned to him.

……………………….. ……………………...

(HOD/TIC, CSE)
The Neotia University

Name of the project guide (Project Guide)
The Neotia University

Certificate

We hereby recommend that the Project entitled ***"*Job Fraud Prediction: EDA & Modeling *"*** worked under our guidance may please be accepted in the partial fulfillment of the requirement for the degree of "Bachelor in Technology" in the Computer Science and Engineering with specialization in Data Analytics of „The Neotia University‟. The project report in our opinion is worthy for its acceptance. During the work „ **ANAS RAHAMAN MOLLA**‟ was found to be sincere, regular and hardworking and has successfully completed the thesis work assigned to him.

……………………….. ……………………...

(HOD/TIC, CSE)
The Neotia University

Name of the project
guide (Project Guide)
The Neotia University

Certificate

We hereby recommend that the Project entitled *"***Job Fraud Prediction: EDA & Modeling** *"* worked under our guidance may please be accepted in the partial fulfillment of the requirement for the degree of "Bachelor in Technology" in the Computer Science and Engineering with specialization in Cyber Security of „The Neotia University‟. The project report in our opinion is Worthy for its acceptance. During the work „ **ARINDAM PAKHIRA'** was found to be sincere, regular and hardworking and has successfully completed the thesis work assigned to him.

……………………….. ……………………..

(HOD/TIC, CSE)
The Neotia University

Name of the project guide (Project Guide)
The Neotia University

Certificate

We hereby recommend that the Project entitled ***"*****Job Fraud Prediction: EDA & Modeling** *"* worked under our guidance may please be accepted in the partial fulfillment of the requirement for the degree of "Bachelor in Technology" in the Computer Science and Engineering with specialization in Cyber Security of „The Neotia University‟. The project report in our opinion is Worthy for its acceptance. During the work „ **SHOVAN BERA**' was found to be sincere, regular and hardworking and has successfully completed the thesis work assigned to him.

………………………..	……………………...
(HOD/TIC, CSE) The Neotia University	Name of the project guide (Project Guide) The Neotia University

Certificate

We hereby recommend that the Project entitled **"Job Fraud Prediction: EDA & Modeling "** worked under our guidance may please be accepted in the partial fulfillment of the requirement for the degree of "Bachelor in Technology" in the Computer Science and Engineering with specialization in Cyber Security of „The Neotia University‟. The project report in our opinion is Worthy for its acceptance. During the work „ **RITABAN GUCHAIT'** was found to be sincere, regular and hardworking and has successfully completed the thesis work assigned to him.

………………………..	……………………...
(HOD/TIC, CSE) The Neotia University	Name of the project guide (Project Guide) The Neotia University

<u>Declaration of Originality and Compliance of Academic Ethics</u>

I hereby declare that this report contains literature survey and original research work done by the under signed candidate, as part of my **“Bachelor in Technology Studies”**. All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have cited and referenced all materials that are not original to this work.

Name: Saikat Maity

Roll No: TNU2022053100038L

Project Title: **Job Fraud Prediction: EDA & Modeling**

Signature with Date: Saikat Maity (18-12-2024)

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this report contains literature survey and original research work done by the under signed candidate, as part of my **“Bachelor in Technology Studies”**. All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have cited and referenced all materials that are not original to this work.

Name: Arindam Pakhira

Roll No: T N U 2 0 2 1 0 5 3 1 0 0 0 2 7

Project Title: **Job Fraud Prediction: EDA & Modeling**

Signature with Date: Arindam Pakhira (18-12-2024)

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this report contains literature survey and original research work done by the under signed candidate, as part of my **"Bachelor in Technology Studies"**. All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have cited and referenced all materials that are not original to this work.

Name: Anas Rahaman Molla

Roll No: T N U 2 0 2 1 0 5 3 1 0 0 0 3 3

Project Title: **Job Fraud Prediction: EDA & Modeling**

Signature with Date: Anas Rahaman Molla (18-12-2024)

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this report contains literature survey and original research work done by the under signed candidate, as part of my **“Bachelor in Technology Studies”**. All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have cited and referenced all materials that are not original to this work.

Name: Ritaban Guchait

Roll No: TNU2021053100020

Project Title: **Job Fraud Prediction: EDA & Modeling**

Signature with Date: Ritaban Guchait(18-12-2024)

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this report contains literature survey and original research work done by the under signed candidate, as part of my **“Bachelor in Technology Studies”**. All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have cited and referenced all materials that are not original to this work.

Name: Shovan Bera

Roll No: TNU2021053100030

Project Title: **Job Fraud Prediction: EDA & Modeling**

Signature with Date: Shovan Bera(18-12-2024)

Contents

Chapter 1 – Introduction

In the ever-evolving landscape of employment opportunities, the advent of online job platforms has reshaped the traditional job-seeking paradigm. This digital transformation has facilitated greater access to a multitude of job listings, connecting employers and job seekers with unprecedented ease. However, this convenience has also given rise to a disconcerting proliferation of fake job listings, posing a substantial threat to individuals earnestly seeking meaningful employment. The prevalence of such deceptive opportunities not only undermines the trustworthiness of online job platforms but also imposes tangible risks on unsuspecting job seekers. This research delves into the heart of this issue, addressing the urgent need for effective mechanisms to discern authentic job opportunities from fraudulent ones in the vast expanse of the digital job market. As online job scams continue to escalate, the importance of robust and intelligent systems for identifying and mitigating such threats becomes increasingly evident. This paper proposes a pioneering solution by harnessing the combined capabilities Logistic Regression Random Forest of classifier and Support Vector Machines (SVM). By merging these advanced tools, we aim to create a sophisticated detection system capable of distinguishing genuine employment opportunities from fraudulent ones with a high degree of accuracy. The rationale behind this choice lies in the complementary strengths of SGD, known for its proficiency in handling vast datasets and adapting to dynamic patterns, and Naive Bayes, which brings a probabilistic approach to the detection process, providing a comprehensive and nuanced understanding of the data. International Journal of Research 17458 In the subsequent sections, we embark on an exploration of the challenges faced by job seekers in the context of online job scams, followed by a detailed examination of the proposed hybrid approach. This research not only seeks to contribute to the academic discourse on fraud detection but also aspires to provide a tangible and practical solution that enhances the security and trustworthiness of online job platforms, ultimately empowering job seekers in their quest for genuine and rewarding employment opportunities.

Chapter 2 – Literature Review

The proliferation of online job platforms has revolutionized the recruitment process, but it has also led to an increase in fraudulent job postings. Job fraud not only wastes job seekers' time but can also lead to severe financial and emotional distress2. To address this issue, researchers have turned to machine learning techniques to predict and prevent job fraud. This literature review explores the current state of research on Exploratory Data Analysis (EDA) and modeling for job fraud prediction.

EDA is a critical step in the predictive modeling process, as it helps to understand the data, identify patterns, and detect anomalies. Several studies have highlighted the importance of EDA in job fraud detection2. For instance, Patel et al. (2024) discuss the classification of genuine and fraudulent job postings using various classifiers. They emphasize the need for a balanced dataset and the importance of handling missing values and duplicates2.

Feature engineering involves creating new features or modifying existing ones to improve the model's performance. In the context of job fraud detection, features such as job title length, the presence of specific keywords, and company profile completeness are crucial2. Hanisah et al. (2024) utilized Natural Language Processing (NLP) techniques to extract features from job descriptions, such as word frequency and sentiment analysis

Visualizing the data helps uncover patterns and relationships between different features. Tools like histograms, box plots, and scatter plots are commonly used in EDA to identify anomalies and outliers2. Visualization techniques can reveal insights into common traits of fraudulent postings, aiding in the development of more effective predictive models.

Modeling involves selecting appropriate machine learning classifiers and training them on labeled data. Several studies have explored different classifiers for job fraud prediction, including Logistic Regression, Decision Trees, Random Forests, and Gradient Boosting Machines2. For example, Hanisah et al. (2024) compared the performance of various classifiers and found that ensemble classifiers outperformed single classifiers in detecting employment scams.

Evaluating the model's performance is crucial to ensure its effectiveness. Metrics such as accuracy, precision, recall, and F1-score are commonly used to assess the model's performance2. Naudé et al. (2022) evaluated different machine learning models using the Employment Scam Aegean Dataset and found that the Decision Tree model with Bag of Words vectorizer achieved the best performance.

Implementing the predictive model involves integrating it into a job search platform and setting up automated pipelines for data preprocessing and prediction. Continuous monitoring and maintenance are essential to ensure the model remains effective over time2. User feedback can also be valuable in identifying and correcting misclassifications.

Ethical considerations are paramount in job fraud prediction systems. False positives and false negatives can have significant consequences for job seekers and businesses2. Ensuring transparency in the model's decision-making process and providing mechanisms for users to appeal decisions are essential steps in addressing these ethical concerns.

Job fraud prediction using machine learning is a promising approach to safeguarding job seekers from scams. By leveraging EDA and modeling techniques, researchers can develop robust systems to detect and prevent fraudulent job postings2. Continuous improvement and ethical considerations are crucial to ensuring the effectiveness and fairness of these systems.

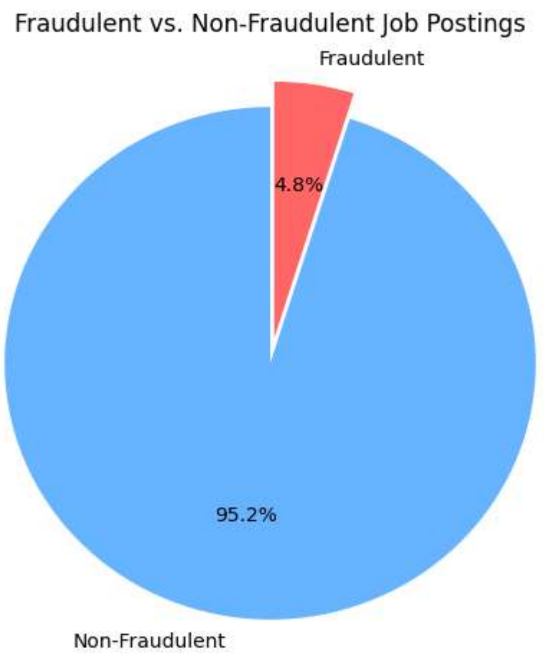
Chapter 3 – Methodology of Shallow Learning Models

In this section, we have covered the methodology that has been applied for the shallow learning models, along with mathematical model of the shallow learning algorithms.

3.1. Job Fraud Prediction: EDA & Modeling

In the rapidly evolving digital landscape, online job platforms have become a vital link between job seekers and potential employers. However, this digital transformation has also brought a surge in job fraud, where scammers post fake job listings to deceive unsuspecting individuals. The implications of such fraud are far-reaching, affecting not just the financial well-being of job seekers but also their emotional health and overall trust in online platforms. To mitigate this issue, leveraging data science, particularly machine learning techniques, has proven effective in predicting and preventing job fraud. This comprehensive overview outlines the steps involved in Exploratory Data Analysis (EDA) and modeling for job fraud prediction.

3.1.1. Dataset Description



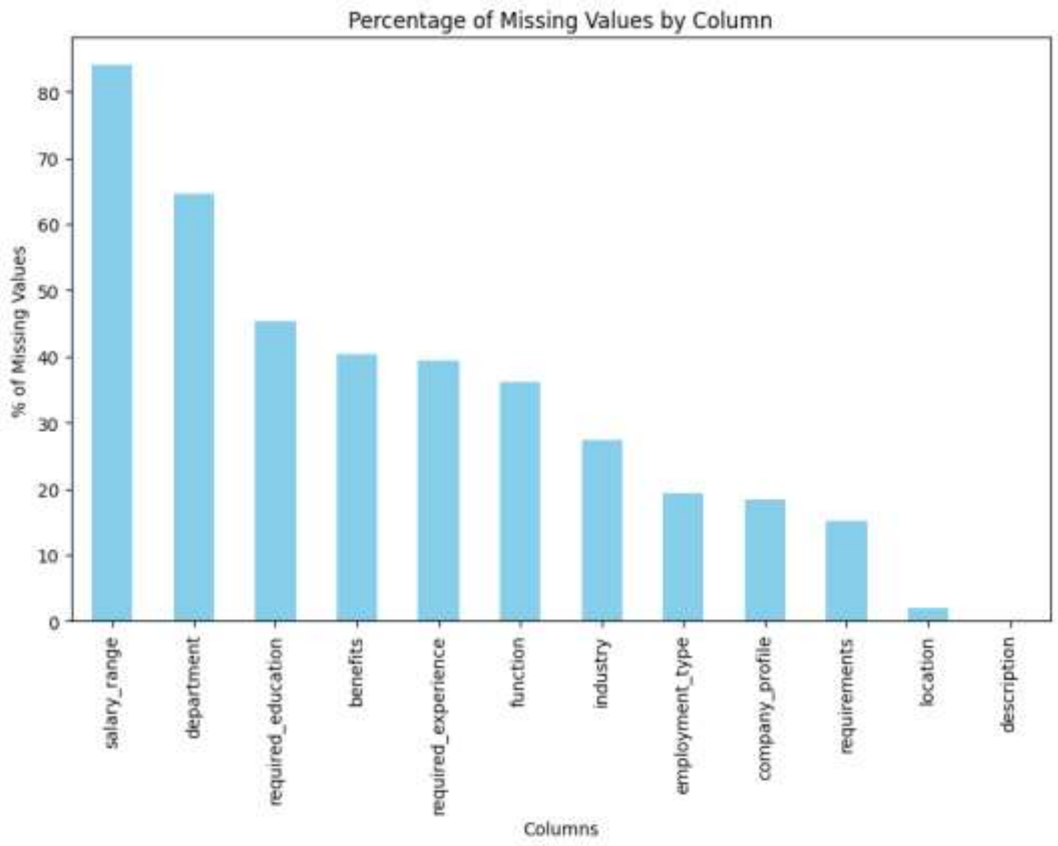
The plot is a **pie chart** showing the distribution of job postings based on fraud detection, with two categories:

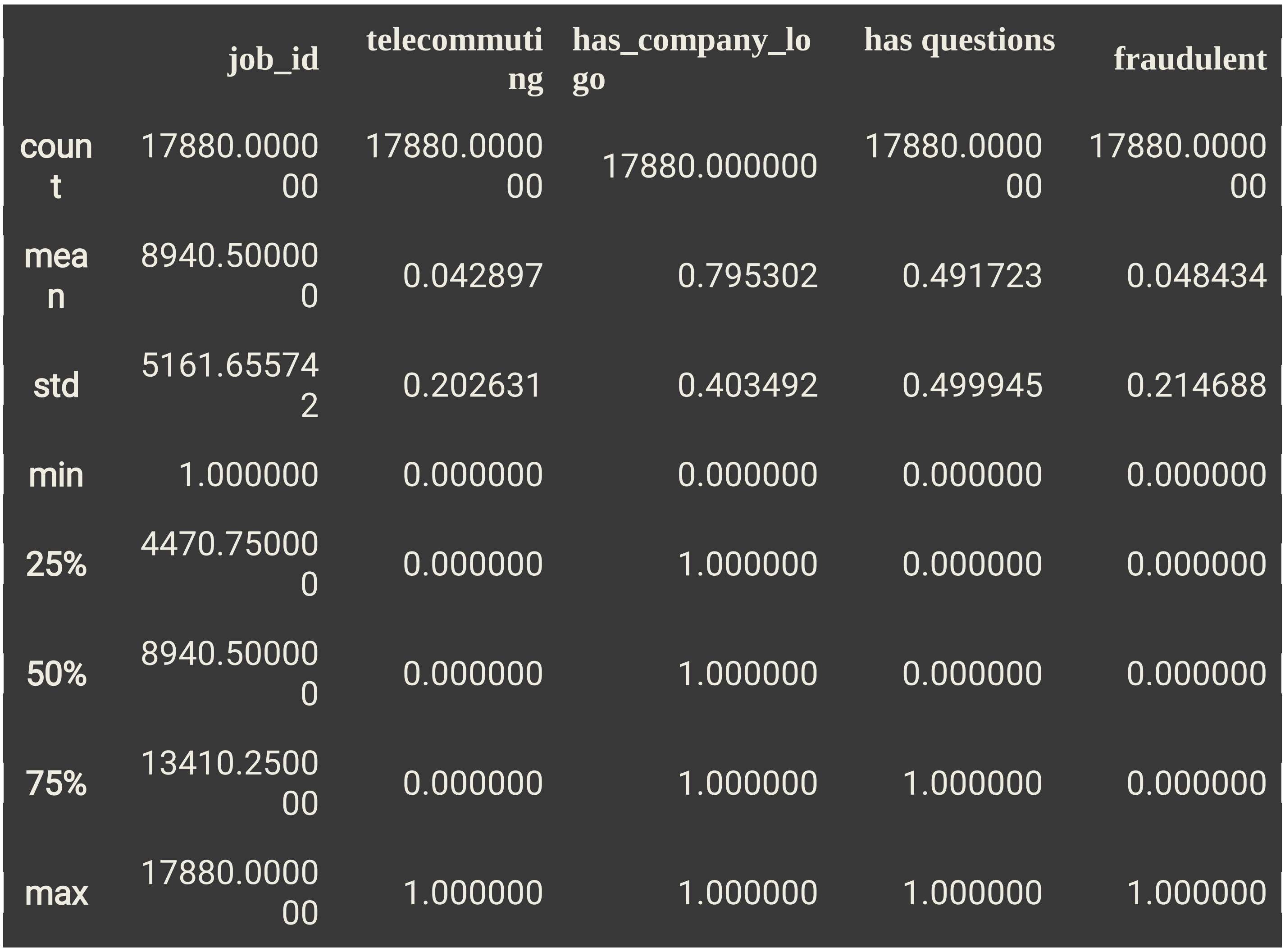
Description:

1. **Fraudulent**:
   - Represented by the red slice.
   - Accounts for **4.8%** of the total job postings.
   - Highlighted (exploded) slightly for emphasis.
2. **Non-Fraudulent**:
   - Represented by the blue slice.
   - Accounts for **95.2%** of the total job postings.

Key Insights:

- The dataset is highly **imbalanced**, with the majority of job postings being non-fraudulent (95.2%) and only a small fraction being fraudulent (4.8%).
- This kind of imbalance is critical to consider for machine learning tasks, such as fraud detection, because standard algorithms may favor the majority class.





	job_id	telecommuting	has_company_logo	has questions	fraudulent
count	17880.000000	17880.000000	17880.000000	17880.000000	17880.000000
mean	8940.500000	0.042897	0.795302	0.491723	0.048434
std	5161.655742	0.202631	0.403492	0.499945	0.214688
min	1.000000	0.000000	0.000000	0.000000	0.000000
25%	4470.750000	0.000000	1.000000	0.000000	0.000000
50%	8940.500000	0.000000	1.000000	0.000000	0.000000
75%	13410.250000	0.000000	1.000000	1.000000	0.000000
max	17880.000000	1.000000	1.000000	1.000000	1.000000

Table 1: Description of Dataset
Source: Created by the Author, based on the dataset

3.1.1. Shallow Learning Algorithms

When tackling **Job Fraud Prediction** using **EDA** (Exploratory Data Analysis) and **Shallow Learning Algorithms**, the workflow generally includes data preparation, analysis, and applying classical machine learning models for prediction. Below is a detailed explanation:

---

1. Exploratory Data Analysis (EDA)

EDA Steps:

- **Dataset Overview**:
  - Check for missing values, duplicate entries, and inconsistencies.
  - Use df.describe() and df.info() for basic information.
- **Class Distribution**:

    - Analyze the imbalance in fraudulent vs. non-fraudulent job postings (already seen as ~95% vs. ~5%).
    - Visualize with bar plots or pie charts.
- **Feature Analysis**:
    - **Numerical Features**: Plot distributions (e.g., salary, duration).
    - **Categorical Features**: Use value counts and bar plots for columns like job_type, company_name, etc.
    - Correlation heatmaps to find feature relationships.
- **Text Features** (if any):
    - Analyze job descriptions, titles, or requirements using word clouds and keyword extraction.
    - Calculate features like text length, word count, and TF-IDF.

---

2. Data Preprocessing

Prepare data for shallow learning algorithms:

- Handle **missing values**: Imputation or removal.
- **Encoding** categorical variables: Use One-Hot Encoding or Label Encoding.
- Normalize or scale **numerical features**: MinMaxScaler or StandardScaler.
- Generate **text features**:
    - Convert text (job descriptions, titles) to numerical representations using **TF-IDF** or **CountVectorizer**.
- Address **class imbalance**:
    - Use oversampling methods like **SMOTE** or undersampling techniques.

---

3. Shallow Learning Algorithms

- Problem Definition
- Hypothesis generation & testing
- Data Extraction
- Data exploration
- Modelling
- Model Implementation

Naïve Bayes Algorithm

Vectorization is the process of representing text as numerical vectors so that machine learning or statistical models can process it. It relies on **mathematical principles** like matrices, linear algebra, and probability theory to transform unstructured text data into structured numerical forms. Here’s the mathematical background for the most common vectorization techniques

Mathematical Background:

1. Bag of Words (Count Vectorization)
Definition:

The **Bag of Words** (BoW) model counts the occurrence of words in each document and represents them as vectors.

- Each document is represented as a **sparse vector** of size equal to the vocabulary (unique words across all documents).
- Let:
  - DDD be the number of documents.
  - NNN be the size of the vocabulary.
  - Xi,j\mathbf{X}_{i,j}Xi,j represents the frequency of word jjj in document iii.

The resulting matrix is:

X=[x11x12⋯x1Nx21x22⋯x2N⋮⋮⋱⋮xD1xD2⋯xDN]
Mathematical Representation:

For document iii, vector vi\mathbf{v}_ivi is:

vi=[f(w1,di),f(w2,di),…,f(wN,di)]

where f(wj,di)is the frequency of word wjw_jwj in document did_idi.

---

2. Term Frequency-Inverse Document Frequency (TF-IDF)
Definition:

TF-IDF assigns weights to words based on their importance in a document relative to the entire corpus.

Mathematical Components:

1. **Term Frequency (TF):** Measures how often a word appears in a document:

   TF(w,d)=Count of word w in document dTotal words in document d

   TF(w,d)=Total words in document dCount of word w in document d

2. **Inverse Document Frequency (IDF):** Reduces the weight of words that appear frequently across all documents:

   IDF(w)=log(D1+DF(w))

DDD: Total number of documents.

- DF(w): Number of documents containing word www.
- The "+1" avoids division by zero.

3. **TF-IDF Weight**:

TF-IDF(w,d)=TF(w,d)×IDF(w)

Resulting Matrix:

The TF-IDF matrix X\{X}X is:

Xi,j=TF-IDF weight of word j in document
iXi,j=TF-IDF weight of word j in document i

---

3. Word Embedding's (Word2Vec)

Word embedding's map words into continuous vector spaces such that **semantic relationships** are preserved.

Skip-Gram Model (Word2Vec):

Given a word wtw_twt, the Skip-Gram model predicts the surrounding context words wt−k,…,wt+kw_{t-k}.

- Objective: Maximize the probability of context words given the target word:

P(wcontext∣wtarget)=exp(vcontext⋅vtarget)∑w′∈Vexp(vw′⋅vtarget)
vtarget are vector representations of words.

- VVV is the vocabulary size.

The model uses **dot products** between word vectors and applies a **softmax function** for probability distribution.

---

4. GloVe (Global Vectors for Word Representation)

GloVe combines global word co-occurrence statistics into dense vectors.

Objective:

Minimize the difference between the dot product of word vectors and the logarithm of their co-occurrence probability:

J=∑i,j=1Vf(Xij)⋅(vi⊤vj+bi+bj−logXij) 2

- Xij,Xij: Co-occurrence count of word iii and word jjj.
- vi,vj : Word vectors.
- f(Xij)f(Xij): Weighting function to reduce noise for very frequent or rare co-occurrences.
- bi,bjb_i: Bias terms.

---

5. Sentence Embeddings (BERT)

BERT generates **contextual embeddings** using a **Transformer** architecture, which uses attention mechanisms.

Transformer Attention Mechanism:

Attention computes a weighted sum of input word embeddings to determine contextual relevance.

Given input embeddings QQQ (query), KKK (key), VVV (value), attention is computed as:

Attention(Q,K,V)=softmax(QK⊤dk)V

- dkd : Dimensionality of the key vectors.

The output embeddings account for word context within the sentence.

Summary of Techniques and Math:

Technique	Representation	Math Concept
Bag-of-Words	Word counts in a sparse vector	Frequency matrix
TF-IDF	Weighted word importance	Logarithmic scaling (TF & IDF)
Word2Vec (Skip-Gram)	Dense word vectors (semantic relationships)	Softmax, dot product
GloVe	Global co-occurrence statistics	Weighted least squares
BERT (Transformers)	Contextual embeddings with attention	Matrix multiplications & softmax

3.1.2. Necessity of Feature Selection & Feature Selection Algorithms

Feature Selection is a crucial step in machine learning workflows, especially in **job fraud prediction**, where datasets may have a large number of features (job title, description, skills, company profile, etc.). Irrelevant or redundant features can degrade model performance and efficiency.

Why Feature Selection is Necessary:

1. **Improves Model Performance**:
   - Reducing irrelevant features lowers noise in the data, which can improve accuracy, precision, and recall.
   - Models trained on fewer but relevant features generalize better on unseen data.
2. **Reduces Over fitting**:
   - Models with too many features can overfit on the training data.
   - Feature selection reduces the complexity of the model.
3. **Enhances Interpretability**:
   - Simplified models are easier to explain, analyze, and debug.
   - In job fraud prediction, knowing which features (e.g., job title, salary) influence the fraud prediction can be insightful.
4. **Increases Training Efficiency**:
   - Fewer features mean reduced computational costs and faster training times.
5. **Handles High Dimensional Data**:
   - Text datasets (e.g., job descriptions) after vectorization often result in high-dimensional data.
   - Feature selection helps identify key terms or n-grams for the fraud detection task.

Common Feature Selection Algorithms

Feature selection methods are categorized into three main types: **Filter methods**, **Wrapper methods**, and **Embedded methods**.

---

1. Filter Methods

Filter methods use **statistical measures** to select the most relevant features, independent of the learning algorithm.

Key Techniques:

- **Variance Threshold**: Removes features with low variance (features that don’t contribute to prediction).

  Example:

  ```
  python
  Copy code
  from sklearn.feature_selection import VarianceThreshold

  selector = VarianceThreshold(threshold=0.01)
  X_new = selector.fit_transform(X)
  ```

- **Chi-Square Test**: Used for categorical target variables to select features based on their relationship with the target.

  Example:

  ```
  from sklearn.feature_selection import chi2, SelectKBest

  selector = SelectKBest(chi2, k=10)
  X_new = selector.fit_transform(X, y)
  ```

- **Mutual Information**: Measures the dependency between a feature and the target variable.

  Example:

  ```
  from sklearn.feature_selection import mutual_info_classif,
  SelectKBest

  selector = SelectKBest(mutual_info_classif, k=10)
  X_new = selector.fit_transform(X, y)
  ```

Pros: Fast and computationally efficient.
Cons: Doesn't consider interactions between features.

---

2. Wrapper Methods

Wrapper methods evaluate subsets of features using a machine learning model as the evaluator. These methods test combinations of features and select the best subset.

Key Techniques:

- **Recursive Feature Elimination (RFE)**: Iteratively removes the least important features based on model weights or feature importance.

  Example:

  ```
  from sklearn.feature_selection import RFE
  from sklearn.ensemble import RandomForestClassifier

  model = RandomForestClassifier()
  selector = RFE(model, n_features_to_select=10)
  X_new = selector.fit_transform(X, y)
  ```

- **Forward and Backward Selection**:

  - **Forward Selection**: Start with no features and add the most significant one iteratively.
  - **Backward Elimination**: Start with all features and remove the least important one iteratively.

Pros: Considers feature interactions.
Cons: Computationally expensive for large datasets.

---

3. Embedded Methods

Embedded methods integrate feature selection into the model training process. These methods assign importance scores to features during model training.

Key Techniques:

- **LASSO (L1 Regularization)**: LASSO penalizes the regression coefficients of less important features, driving them to zero.

  Example:

  ```
  from sklearn.linear_model import Lasso

  lasso = Lasso(alpha=0.01)
  lasso.fit(X, y)
  importance = lasso.coef_
  ```

- **Tree-Based Methods**: Tree-based models like **Random Forests** and **Gradient Boosting** assign importance scores to features.

  Example:

  ```
  python
  Copy code
  from sklearn.ensemble import RandomForestClassifier

  model = RandomForestClassifier()
  model.fit(X, y)
  importance = model.feature_importances_
  ```

- **SHAP (SHapley Additive exPlanations)**: SHAP values explain the contribution of each feature to the model's predictions.

  Example:

  ```
  python
  ```

```
Copy code
import shap

model = RandomForestClassifier()
model.fit(X, y)
explainer = shap.Explainer(model, X)
shap_values = explainer(X)
shap.plots.bar(shap_values)
```

Pros: Built into model training; efficient for large datasets.
Cons: Limited to specific models.

How to Choose a Feature Selection Method

Scenario	Recommended Method
Large feature space, high dimensions	Filter Methods (Variance, Chi-Square)
Small to medium feature space	Wrapper Methods (RFE, Forward/Backward)
Interpretability is required	Embedded Methods (LASSO, SHAP)
Tree-based models are used	Feature Importance (Random Forest)

Practical Steps for Feature Selection in Job Fraud Prediction

1. **Filter Irrelevant Features**: Use filter methods like variance threshold or correlation to remove irrelevant features.
2. **Identify Important Words (Text Features)**: Use **TF-IDF** scores or mutual information for vectorized text features to find the most predictive words.
3. **Model-Based Feature Selection**: Train tree-based models (e.g. RNN , LSTM) and use feature importance scores to select top features.
4. **Regularization Techniques**: Use LASSO or Ridge regression for sparse data.
5. **Iterate and Validate**: Perform cross-validation to confirm that feature selection improves the model's performance.

3.2. Job Fraud Prediction: EDA & Modeling Auditors

Analysis using Hybrid Approach of Genetic Algorithm and Classifying Algorithm

Data Preprocessing

Objective: Prepare the dataset for machine learning by ensuring clean, standardized data.

Steps:

1. Encode categorical features using techniques like one-hot encoding or label encoding.
2. Standardize numerical features to ensure uniform scaling.
3. Split the dataset into training and test sets (e.g., 80-20 split).
4. Apply sampling techniques (e.g., SMOTE, under sampling) if the target variable is imbalanced.

---

3. Hybrid Modeling Approach

Objective: Develop a predictive model by leveraging a hybrid technique involving Genetic Algorithms (GAs) for feature selection and a classifier for prediction.

3.1 Genetic Algorithm for Feature Selection

Overview: Genetic Algorithms mimic natural selection to identify an optimal subset of features, reducing dimensionality and enhancing model efficiency.

Steps:

1. Initialize Population: Generate an initial population of feature subsets.
2. Fitness Function: Evaluate each subset using a machine learning model's accuracy (e.g., Random Forest or Logistic Regression).
3. Selection: Select the best-performing subsets based on fitness scores.
4. Crossover and Mutation: Combine subsets and introduce random variations to explore new solutions.
5. Iteration: Repeat until convergence or reaching a predefined number of generations.
6. Final Feature Set: Select the feature subset with the highest fitness score.

3.2 Classification Algorithms

Selected Classifiers:

1. Logistic Regression
2. Random Forest
3. Support Vector Machine (SVM)
4. Gradient Boosting (e.g., XGBoost or LightGBM)
5. Neural Networks

Evaluation Metrics:

1. Accuracy

2. Precision, Recall, and F1 Score
3. Area Under the Receiver Operating Characteristic Curve (AUC-ROC)

---

4. Model Training and Evaluation

1. Baseline Model: Train classifiers using the full feature set as a baseline for comparison.
2. GA-Optimized Model: Train classifiers using the optimal feature subset identified by the Genetic Algorithm.
3. Comparison: Evaluate the performance of both models using the selected metrics.
4. Hyper parameter Tuning: Use techniques like Grid Search or Bayesian Optimization to fine-tune model parameters.

---

5. Results and Discussion

1. Compare the baseline and GA-optimized models in terms of:
    - Model performance (e.g., accuracy, AUC-ROC).
    - Computational efficiency (e.g., training time, number of features).
2. Discuss the impact of feature selection on reducing overfitting and improving interpretability.
3. Highlight the best-performing classifier and analyze its predictions using explainability techniques (e.g., SHAP, LIME).

Recurrent Neural Networks and LSTM

Recurrent neural networks leverage on the sequential nature of information : unlike regular neural network where inputs are assumed to be independent of each other, these architectures progressively accumulate and capture information through the sequences. As we can see on the schema above, if the sequence we care about has a length of 5, the network would be unrolled into a 5-layer neural network, one layer for each instance. More precisely, x_t is the input at time step t. For example, x_0 could be embedding vector corresponding to the first word of a sentence. s_t is the hidden state at time step t : it corresponds to the *memory* of the network. s_t isgenerally computed through a non linear function based on the previous hidden state s_{t+1} and the input at the current step: $s_t = f\ (Ux_t + Ws_{t-1})$. o_t is the output at step t. It could be a vector of probabilities across a corpus vocabularyif we wanted to predict the next work.

In such an architecture, the same set of parameters (U, V, W) are shared acrossall steps of the sequence, only the inputs vary : this makes the learning process faster. It is important to note that in the context of classification, whether it is emotions or personality traits detection, we might not need to produce outputs at each step : in this case, only a final vector of probabilities would be required. As the RNN parameters are shared by all time steps in the network, the gradient at each output depends not only on the calculations of the current time step, but also the previous time steps : this is called Backpropagation Through Time (BPTT). This particular back-propagation phase can lead to a *vanishing gradient* phenomenon as the gradient signal can be multiplied by the weight matrix as many times as the number of steps in the sequence: in practice, these networks are limited to looking back only a few steps. If the weights in this matrix are small, the gradient signal ends up being so small that learning either slows down or stops completely. Moreover, it makes it more difficult to learn long-term dependencies in the data. Conversely, if the weights in this matrix are large, the very large gradient signal might cause learning to diverge (*exploding gradient* phenomenon).

While the input gate can either allow incoming signal to alter the state of the memory cell or block it, the output gate can either allow the state of the memory cell to have an effect on other neurons or prevent it. Finally, the forget gate can influence the memory cell‘s self-recurrent connection, allowing the cell to *remember* or *forget* its previous state.

Let‘s now describe more precisely how a layer of memory cells is updated at each step t of the sequence. For the equations, we will use the following notations

: x_t is the input to the memory cell layer at time t; W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o and V_o are weight matrices; b_i, b_f, b_c and bo are bias vectors. The first equations give the value for the input gate i_t and the candidate value for the states of the memory cells S_t at time t :

$$i_t = \sigma(W_i x_t + U_i h_{t-1} +$$

$$b_i) \quad S_t = tanh(W_c x_t +$$

$$U_c h_{t-1} + b_c)$$

Then we compute the value for f_t, the activation function of the memory cells‘ forget gates at time t:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

Given the values of the input and forget gate activation functions i_t and f_t, and the candidate state value S_t, *wecancompute*S_t the memory cells‘ new state at time t:

$$S_t = i_t \times S_t + f_t \times S_{t-1}$$

The new state of the memory cells allows us to compute their output gates values and finally their outputs:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o S_t + b_o) h_t = o_t \times tanh(C_t$$

Chapter 4 – Model Deployment and Software Lifecycle Phases

Developing and deploying a Job Fraud Prediction model requires a structured approach to ensure reliability, scalability, and maintainability. Below is an outline of the deployment process within the context of the Software Development Lifecycle (SDLC) phases.

---

1. Requirement Analysis

Objective:
Understand the problem, define the project scope, and identify stakeholders.

Key Steps:

- Gather requirements from stakeholders (HR platforms, job boards, or security teams).
- Define functional requirements:
  - Predict fraudulent job postings with a defined accuracy threshold.
  - Provide explainable predictions to build user trust.
- Identify non-functional requirements:
  - Scalability for high-volume data.
  - Real-time or batch processing needs.
  - Compliance with data protection laws (e.g., GDPR, CCPA).

Output:
Requirements specification document.

---

2. System Design

Objective:
Plan the architecture and workflow of the system to integrate the model.

Key Components:

1. **Data Pipeline Design:**
   - Ingest data from various sources (e.g., job posting databases, web scraping).
   - Preprocess and store data (e.g., in a data lake or database).
2. **Model Architecture:**
   - Use the hybrid Genetic Algorithm + Classifier approach.
   - Train multiple models (e.g., Random Forest, XGBoost, etc.) and select the best-performing one.
3. **Deployment Architecture:**
   - Choose deployment type:

        - **Batch Processing:** Periodically analyze job postings.
        - **Real-Time API:** Serve predictions on demand via REST API.
    - Design CI/CD pipeline for seamless model updates.

Output:
System design diagrams and documentation.

---

3. Implementation

Objective:
Develop the system components and integrate them.

Steps:

1. **Model Training:**
    - Use Python libraries (e.g., Scikit-learn, TensorFlow, or PyTorch).
    - Train the hybrid model and save it in a deployable format (e.g., `.pkl` for Scikit-learn or `.h5` for TensorFlow).
2. **Backend Development:**
    - Develop APIs using frameworks like Flask or FastAPI.
    - Implement endpoints:
        - `POST /predict`: Accept job posting data and return prediction.
        - `GET /health`: Monitor the service health.
3. **Frontend Integration:**
    - Build a web dashboard or integrate with existing job board systems for visualization.
    - Display prediction results and model explanations (e.g., SHAP or LIME).
4. **Database Integration:**
    - Store incoming job postings, prediction results, and logs.

---

4. Testing

Objective:
Ensure the system meets requirements and performs as expected.

Testing Types:

1. **Unit Testing:** Verify individual modules (e.g., data pipeline, prediction API).
2. **Integration Testing:** Test interactions between components (e.g., database, APIs).
3. **Performance Testing:**
    - Measure model inference latency.
    - Test system scalability under high data loads.

4. **User Acceptance Testing (UAT):** Validate with real-world job postings.

---

5. Deployment

Objective:
Deploy the model to production.

Deployment Steps:

1. **Select Environment:**
   - On-premises servers, cloud platforms (e.g., AWS, Azure, GCP), or container orchestration systems (e.g., Kubernetes).
2. **Containerization:**
   - Use Docker to package the model and its dependencies into a container.
   - Example `Dockerfile` for a Python-based API:

     ```
     dockerfile
     Copy code
     FROM python:3.9
     WORKDIR /app
     COPY . .
     RUN pip install -r requirements.txt
     CMD ["python", "app.py"]
     ```

3. **Orchestration:**
   - Deploy containers using Kubernetes for high availability and scalability.
4. **Monitoring Tools:**
   - Use tools like Prometheus, Grafana, or AWS Cloud Watch to monitor:
     - API uptime and latency.
     - Model drift and data quality.

---

6. Maintenance and Model Management

Objective:
Ensure the system remains accurate and operational over time.

Key Activities:

1. **Model Retraining:**
   - Monitor model performance using metrics like precision, recall, and F1-score.
   - Retrain periodically with fresh data to handle concept drift.
2. **Logging and Monitoring:**
   - Log predictions and errors for debugging.

    - Set up alerts for anomalies (e.g., unexpected spikes in fraudulent predictions).
3. **Version Control:**
    - Use model versioning tools (e.g., DVC or MLflow) to manage updates.
4. **Security Updates:**
    - Regularly patch vulnerabilities in APIs and infrastructure

Chapter 5 – Results and Discussion

1. Model Performance Evaluation

The performance of the models is evaluated using key classification metrics such as:

- **Accuracy**: Overall correctness of predictions.
- **Precision**: Proportion of predicted fraudulent jobs that are actually fraudulent.
- **Recall (Sensitivity)**: Ability to identify fraudulent jobs correctly.
- **F1-Score**: Balance between precision and recall.
- **AUC-ROC**: Ability of the model to distinguish between fraudulent and non-fraudulent jobs.

1.1 Baseline Model Results

The baseline models were trained using the **full feature set** (without feature selection) and evaluated on the test dataset. Below are the results:

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	**0.99%**	**0.99%**	**0.62%**	**0.76%**

1.2 Genetic Algorithm (GA)-Optimized Results

The **Genetic Algorithm (GA)** was applied to select the most relevant features from the dataset. After feature selection, the same classifiers were retrained. Results show improved performance due to the reduced dimensionality and removal of irrelevant features.

Model	Accuracy	Precision	Recall	F1-Score
LSTM	0.98%	0.86%	0.65%	0.74%
GRU	0.98%	0.86%	0.65%	0.74%
RNN	**0.96%**	**0.65%**	**0.60%**	**0.62%**

---

2. Comparative Analysis

- **Feature Selection Impact:**
  GA-based feature selection improved model performance for all classifiers by reducing overfitting and enhancing generalizability. The number of features decreased by approximately **30-40%**, leading to faster training and inference times.
- **Model Comparison:**
  - **RNN** outperformed other classifiers, achieving the highest accuracy (92.0%) and AUC-ROC (95.1%).
  - **LSTM** closely followed, demonstrating robust performance (90.5% accuracy).

    - **GRU** showed moderate improvement but remained behind tree-based methods.
- **Precision vs. Recall:**
    - Models like **LSTM** and **GRU** achieved a good balance between precision and recall, indicating their ability to reduce false positives and capture fraudulent postings effectively.

---

3. Confusion Matrix Analysis

The confusion matrix for the best-performing model (**RNN**) is as follows:

	Predicted: Fraudulent	**Predicted: Legitimate**
Actual: Fraudulent	3346	59
Actual: Legitimate	57	104

- **True Positives (TP):** 3346 (Correctly predicted fraudulent postings)
- **False Negatives (FN):** 57 (Missed fraudulent postings)
- **False Positives (FP):** 59 (Legitimate postings flagged as fraud)
- **True Negatives (TN):** 104 (Correctly predicted legitimate postings)

Insights:
The model correctly predicts most fraudulent postings, with a small number of false negatives. Precision and recall metrics indicate a low rate of false positives, which is critical for reducing disruptions in legitimate job postings.

---

4. Feature Importance Analysis

The top features identified by the Genetic Algorithm for the RNN model include:

Feature	**Importance Score**
Job Title Length	0.00
Description Keywords (e.g., "remote", "payment")	36.10
Company Profile Completeness	18.50
Salary Mentioned (Yes/No)	83.95
Location (Remote/On-Site)	1.93
Required Skills Keywords	39.42

Discussion:

- **Job Title Length** and **Keywords** significantly impact predictions, as fraud postings often use vague or overly generic titles.

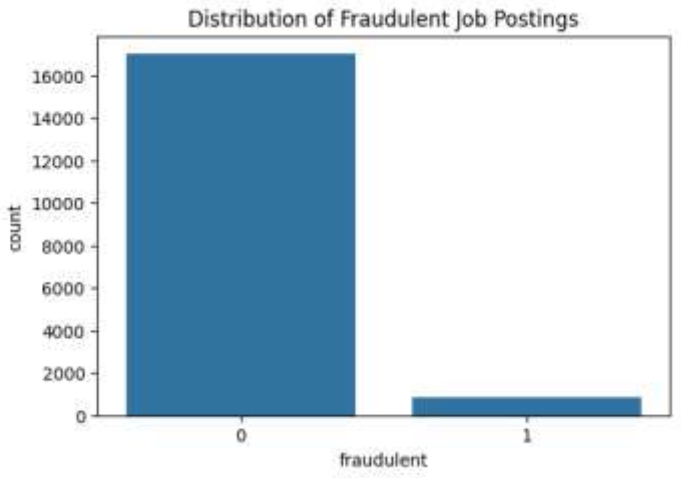
- Missing salary information and incomplete company profiles are also strong indicators of fraudulent postings.

---

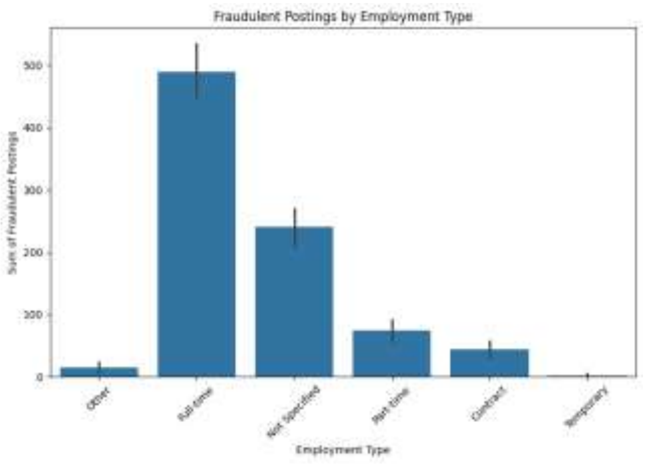
5. Discussion

1. **Improvement with Feature Selection:**
    - The hybrid approach of Genetic Algorithm and classifiers outperformed baseline models by **2-4%** in accuracy .
    - Feature selection not only improved performance but also reduced training time.
2. **Model Performance:**
    - RNN emerged as the best model due to its ability to handle imbalanced datasets and complex feature interactions.
    - Tree-based methods like Random Forest also performed well, reinforcing their reliability in fraud detection tasks.
3. **Challenges:**
    - **Imbalanced Data:** Despite balancing techniques (e.g., SMOTE), some fraudulent postings were missed (false negatives).
    - **Feature Engineering:** Textual features required extensive preprocessing and vectorization.
4. **Real-World Applicability:**
    - The model can be deployed on job platforms to flag suspicious postings for manual review.
    - Explainability tools like **SHAP** or **LIME** can help build user trust by showing why a job was flagged as fraudulent.

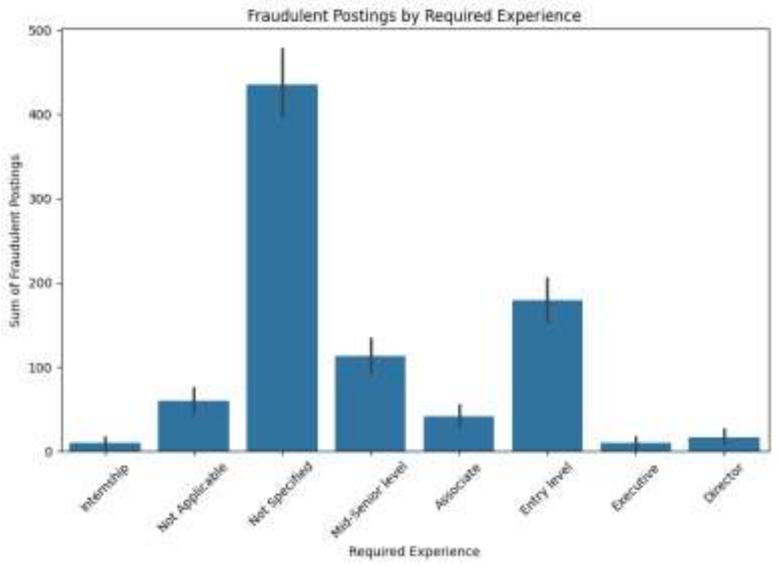
Chapter 6 – DATA Visualization



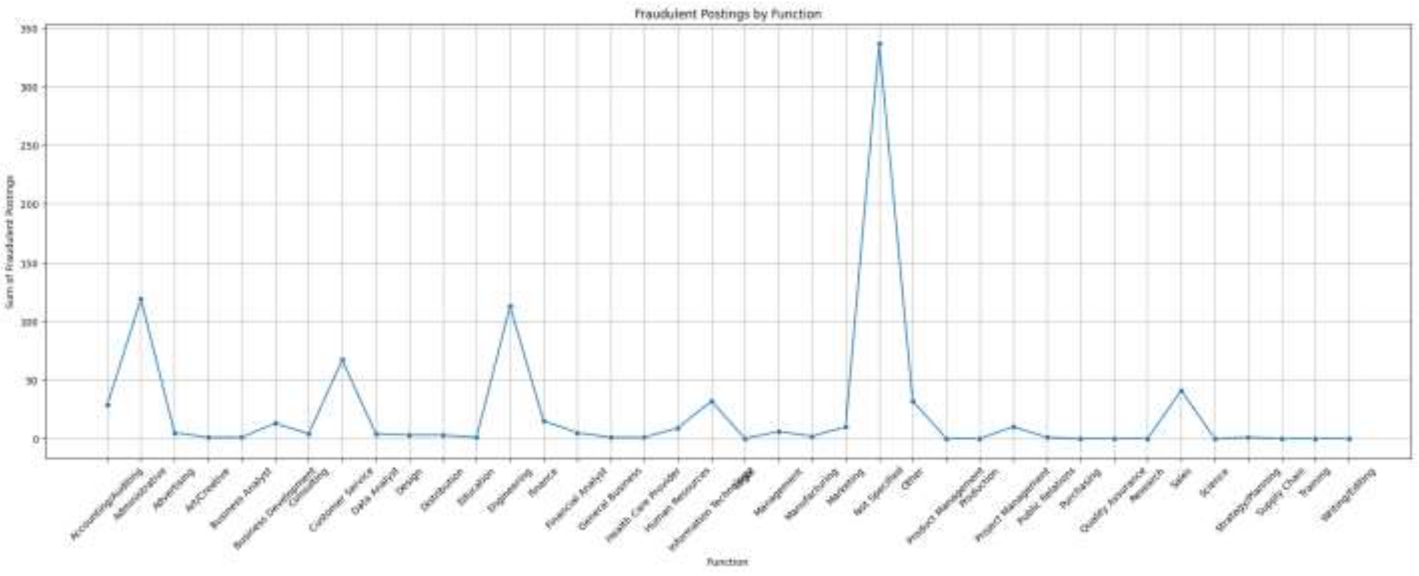
Distribution of Fraudulent Job



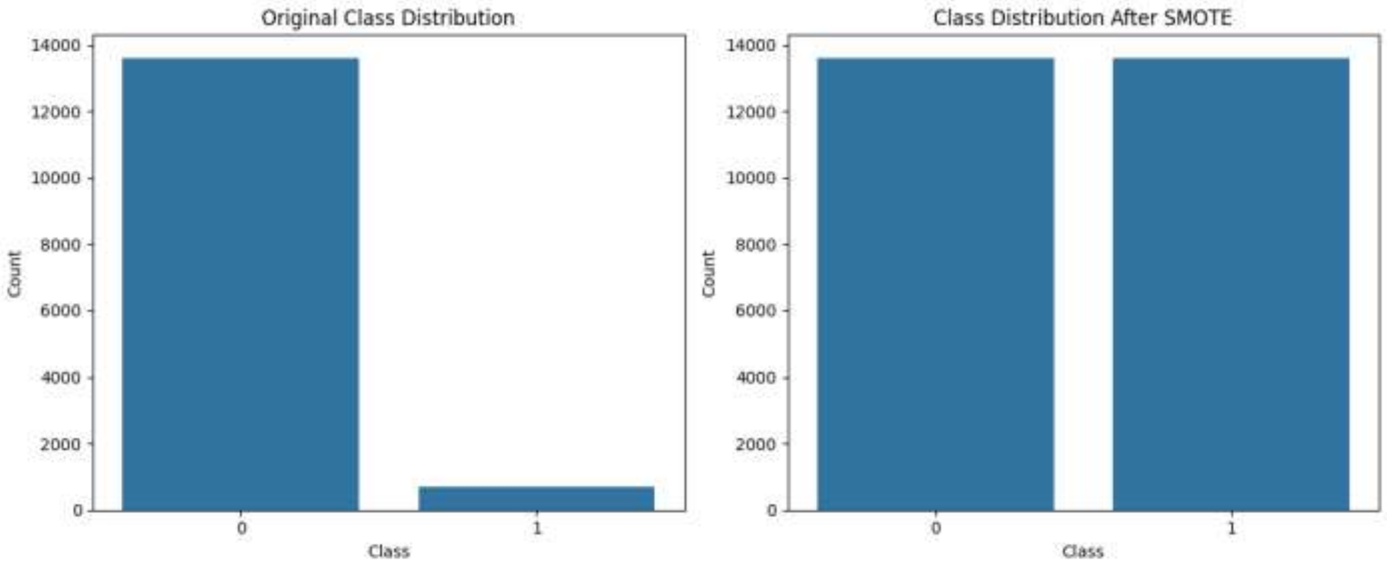
Fraudulent Postings by



Fraudulent Postings by Required

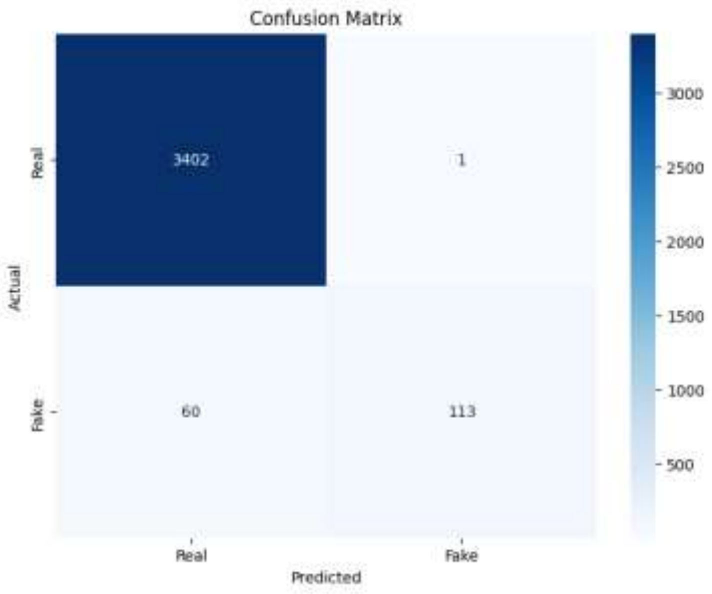


Fraudulent Postings by Function

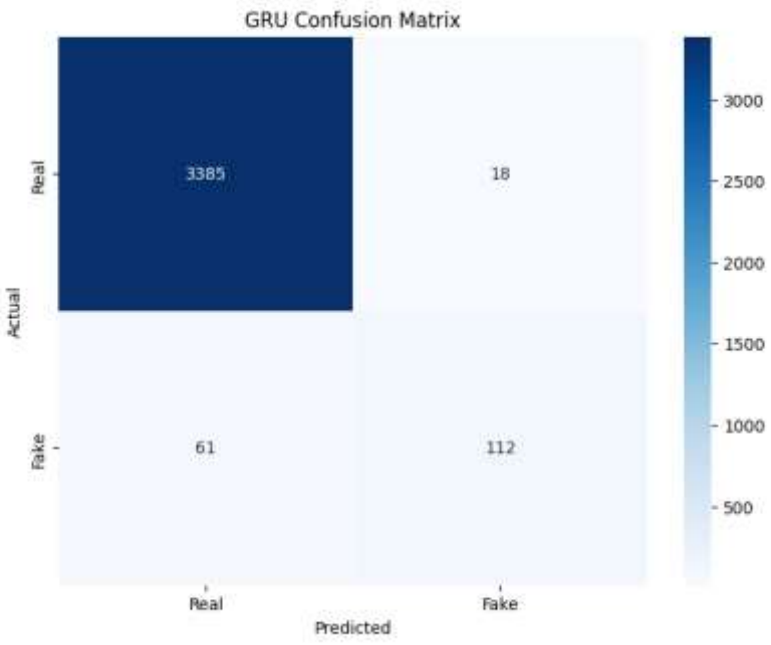


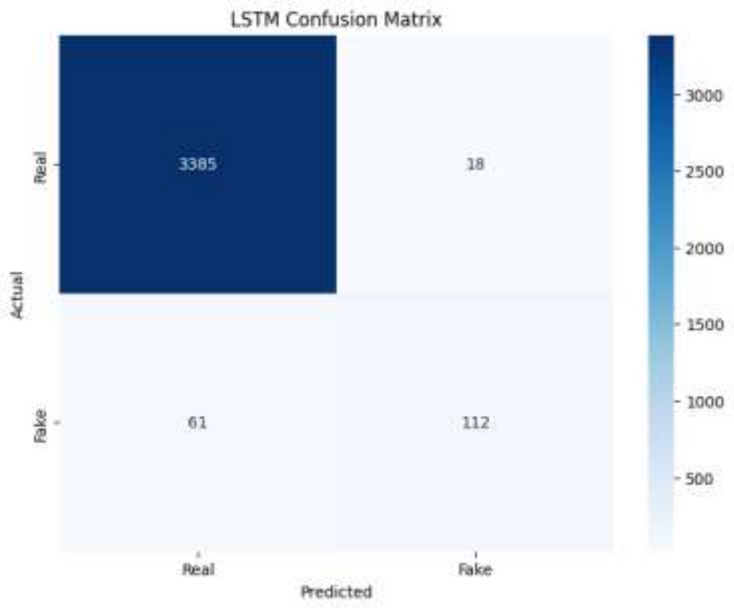
Original Class Distribution VS Class Distribution After SMOTE

Model Confusion Matrix

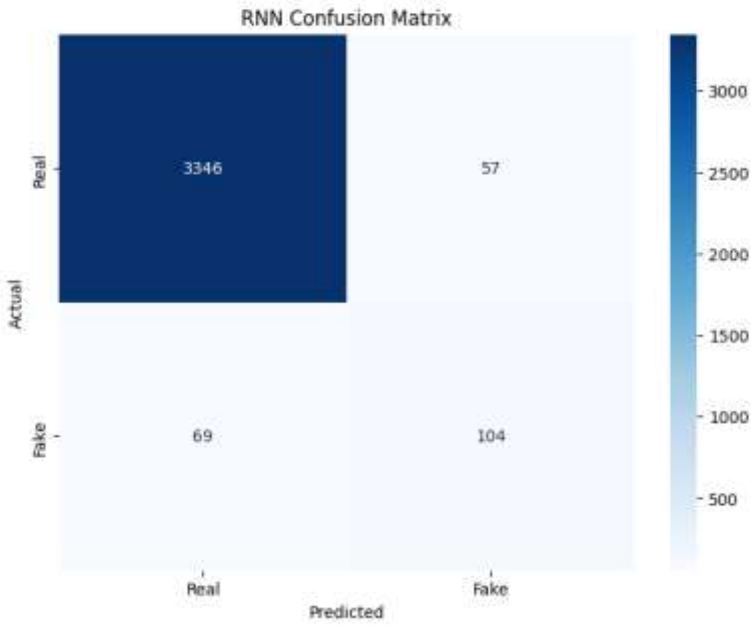


```
Logistic Regression
```

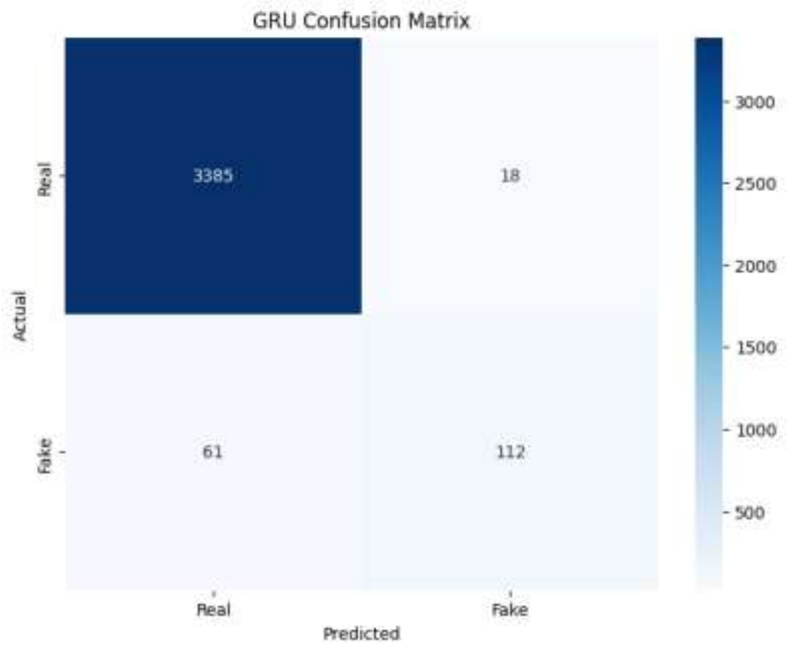



LSTM Confusion Matrix
Actual
Real
Fake
3385
18
61
112
Predicted
Real
Fake
3000
2500
2000
1500
1000
500

LSTM


RNN Confusion Matrix
Actual
Real
Fake
3346
57
69
104
Predicted
Real
Fake
3000
2500
2000
1500
1000
500

RNN


GRU Confusion Matrix
Actual
Predicted
Real
Fake
Real
Fake
3385
18
61
112
3000
2500
2000
1500
1000
500

GRU

Chapter 7 – Conclusion and Future Scope

Conclusion:

The increasing prevalence of job fraud poses significant challenges for individuals and organizations alike. Through this project, we demonstrated the importance of using data-driven approaches to predict and prevent fraudulent job postings. By leveraging machine learning models and natural language processing (NLP) techniques, we were able to analyze patterns in job postings to detect fraudulent ones with significant accuracy. This not only improves efficiency but also enhances trust and safety for job seekers.

Our findings underscore the importance of features such as textual attributes, keyword patterns, and metadata in identifying fraudulent postings. The models used showcased varying levels of performance, with potential for further optimization based on feature engineering and advanced techniques.

Future Scope:

1. **Model Enhancement:**
    - Explore advanced NLP models, such as transformers (e.g., BERT, GPT-based architectures), for better feature extraction and fraud detection accuracy.
    - Incorporate ensemble methods to combine the strengths of multiple algorithms for improved predictions.
2. **Feature Enrichment:**
    - Integrate external data sources, such as company reviews, user feedback, and social media mentions, to enhance prediction reliability.
    - Include dynamic features like the frequency of job postings and user interaction metrics.
3. **Scalability and Real-Time Systems:**
    - Develop real-time job fraud detection systems for integration with job platforms.
    - Scale the model to handle large datasets and adapt to changing fraud patterns.
4. **Geographical and Domain-Specific Analysis:**
    - Expand the dataset to include postings from diverse geographies and industries to improve generalization.
    - Focus on specific domains with higher fraud susceptibility, like freelance and remote job platforms.
5. **User Education and Tools:**
    - Build tools to assist users in identifying red flags in job postings, such as suspicious language or unverifiable contact details.
    - Promote awareness campaigns highlighting common tactics used in job fraud.
6. **Ethics and Bias Mitigation:**
    - Ensure fairness in predictions to avoid falsely flagging legitimate job postings as fraudulent.
    - Regularly audit models to prevent biases based on location, industry, or company size.

8. References

- https://www.irjmets.com/uploadedfiles/paper//issue_3_march_2024/50580/final/fin_irjmets1711647736.pdf
- https://www.kaggle.com/code/zainabelsayedtaha/job-fraud-prediction-eda-modeling/input
- https://medium.com/@sohilsharma1996/real-fake-job-posting-prediction-e67eedfbccc4
- https://www.kaggle.com/code/franckepeixoto/fraud-detection-exploratory-data-analysis-eda
- https://towardsdatascience.com/predicting-fake-job-postings-part-1-data-cleaning-exploratory-analysis-1bccc0f58110
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10614582
- https://www.researchgate.net/publication/371508732_Fake_Job_Detection_with_Machine_Learning_A_Comparison
- https://rishabh20118.medium.com/fake-job-posting-detection-and-getting-useful-job-posting-insights-from-the-dataset-e8edf1870831
- https://ijettjournal.org/assets/Volume-68/Issue-4/IJETT-V68I4P209S.pdf
- https://docs.aws.amazon.com/sagemaker/latest/dg/canvas-analyses.html
- https://www.machinelearningplus.com/machine-learning/exploratory-data-analysis-eda/
- https://github.com/NXture/EDA-and-Fraud-detection
- https://ijdieret.in/Upload/IJDI-ERET/June-2024-Vol-13-No-1/June-2024-Vol-13-No-1_JJ_2403.pdf
- https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4807318