Detecting Fraudulent Job Postings with Machine Learning

Authors:

1. Sai Pavan Padamata
2. Madhu Sree Kalluri
3. Venkata Sai Chandra

1. Introduction

1.1 Project Type:

Application Flavor

1.2 Problem Statement:

The expansion of online job portals has enabled convenience of access to employment opportunities for job seekers. However, this convenience has been exploited by scammers who post fake job listings, fake advertising which results in financial and emotional losses for job seekers. The problem we wish to solve is distinguishing between fake and real job postings using machine learning models.

1.3 Project Motivation:

The goal for this project is to develop a robust machine learning model that will be able to successfully identify fraudulent job postings.

The motivation behind this project is two-fold:

First, to create a tool that will be able to assist job seekers in avoiding scams and remaining safe when utilizing online job boards.

Secondly, to explore and experiment with cutting-edge machine learning models, gaining first-hand exposure to their efficiency and performance on binary classification issues.

2. Project Scope and Objectives:

2.1 Key Goals:

- Develop a machine learning model that is able to detect fraudulent job posts.
- Compare various state-of-the-art ML techniques to determine the best approach.
- Experiment with various datasets to ensure model generalization.
- Employ a rigorous methodology to achieve high accuracy and reliability.

3. Methodology:

3.1 Dataset Selection:

- **Primary Dataset:** The Kaggle dataset "Fake Job Posting Prediction" Contains 17,880 records with job descriptions and labels indicating fraudulent postings will serve as the primary dataset for this project. It includes features relevant for classification and a labeled "fraudulent" target variable."

- **Data Augmentation Strategy:**
    - If the given dataset is not sufficient enough then we are planning for data augmentation techniques to generate additional synthetic samples from our existing dataset.
    - Methods include textual data augmentation (synonym replacement, back-translation) and generative models like GPT-based data synthesis**.**
    - This approach ensures that we have a sufficient dataset size without introducing potential noise from external sources.

3.2 Data Preprocessing:

- Handling missing values
- Text preprocessing (tokenization, lemmatization, removing stop words)
- Feature extraction (TF-IDF, word embeddings)
- Encoding categorical features
- Normalization of numerical attributes
- Augmenting text data to improve model generalization

3.3 Model Selection:

To ensure the effectiveness of our fraud detection model, we have selected the following advanced models:

1. BERT (Bidirectional Encoder Representations from Transformers):

- **Reason for Selection:**
    - BERT is highly effective for natural language processing tasks and can capture contextual information in job descriptions which is very important aspect of our project.
    - As it also supports transfer learning, which removes the need for a large, labeled dataset if required.
- **Implementation Approach:**
    - We will utilize a pre-trained BERT model and fine-tune it on the job posting dataset
    - Compare with LSTMs and other deep learning techniques and focus on areas of improvement.

2. LSTM (Long Short-Term Memory):

- **Reason for Selection:**
    - LSTM is well-suited for sequential text data. So we are planning to make it effective for detecting fraud patterns in job descriptions.
    - As, it captures long-range dependencies better than traditional RNNs algorithms we have.

- **Implementation Approach:**
    - Preprocess job descriptions and convert them into word embeddings (using Word2Vec, GloVe, or BERT embeddings).
    - Then train an LSTM model to classify fraudulent vs. legitimate job postings.
    - Compare performance with CNN and BiLSTM models and we will look focus on areas of improvement.

3. BiLSTM (Bidirectional Long Short-Term Memory):

- **Reason for Selection:**
    - Unlike standard LSTM, BiLSTM can processes text in both forward and backward directions.
    - It also enhances fraud detection by understanding contextual relationships from both sides of a sentence.
- **Implementation Approach:**
    - We will start with tokenization and preprocess job postings from our datset.
    - Train a BiLSTM model, then we are planning to experiment of different dropout and attention layers for improved feature learning.
    - Compare its classification performance with LSTM and CNN.

4. CNN (Convolutional Neural Networks for Text Classification):

- **Reason for Selection:**
    - CNNs are faster than LSTMs and effective at detecting key textual patterns in short job descriptions.
    - Which helps to capture local dependencies in text, making them ideal for identifying fraudulent keywords.
- **Implementation Approach:**
    - Convert job descriptions into word embeddings and pass them through a 1D convolutional network.

- Train and optimize CNN architecture with batch normalization and dropout for regularization.

3.4 Experimental Setup:

- **Hyperparameter tuning:** BERT (batch size, and fine-tuning transformer layers), LSTM and BiLSTM (dropout rates, recurrent dropout, and sequence length), CNN (number of filters, pooling mechanism, and activation function).
- **Model comparison based on:** Accuracy, Precision, Recall, F1-score, Cohen's Kappa.
- **Experiment with dataset sizes:** To observe performance variations.
- **Cross-validation:** real-world validation using augmented datasets.
- **Compare different neural network architectures:** LSTM, BiLSTM, CNN, and BERT.

3.5 Evaluation Metrics:

- Accuracy, Precision, Recall, F1-score
- Cohen's Kappa for imbalance handling
- ROC-AUC curve analysis

4. Required Resources:

- **Programming Language:** Python
- **Libraries:** scikit-learn, TensorFlow/PyTorch, XGBoost, CatBoost, NLTK

5. Workload Distribution:

- As a collaborative team, we have worked together to research and identify suitable datasets, discuss various algorithms, and plan the implementation strategy for the next phase.
- **Madhu:** Data collection, preprocessing, and exploratory analysis. Responsible for data augmentation techniques and ensuring dataset sufficiency.
- **Sai Pavan:** Model selection, implementation, and hyperparameter tuning. Ensures comparative analysis of models and validation using real-world patterns.
- **Chandra:** Evaluation, comparison, and report documentation. Focuses on experimental design, metric analysis, and final result interpretation. Our collaboration offers a comprehensive approach to solving the problem, leveraging each member's strengths in providing the best possible outcomes.