CLASSIFICATION OF FAKE JOB POSTINGS

AIDI 1002 Final Project – Statement of Work (V2)

Tejas Vyas
Supervised by Marcos Bittencourt

Contents

Executive Summary

The project's topic is "Classification of Fake Job Postings". This project's importance arises from recently growing scam of fake job postings, which results in an increase in identity theft cases throughout the globe.

The purpose of this project is to develop an intuitive solution by creating a supervised machine learning model that can classify a given job posting into fake or real.

Problem Statement

Globalization and Internet has yielded an amazing job market today which allows businesses and individuals to connect to each other through various non-traditional means. With the rise of freelancers and remote positions, today companies can hire employees with minimal in-person interviews, or in some cases waiving in-person interviews altogether resulting in very swift recruitment.

While this has significantly decreased costs associated with recruitment and increased accessibility of job positions for job seekers, the number of fake job postings intended to scam the applicant have grown tremendously due of the anonymization and minimal infrastructure required.

Machine Learning can help in classify these job postings, assisting in decreasing the scam success rate, similar to detection of fraud emails and other security features.

Rationale Statement

Machine Learning is widely used today to detect fraudulent and spam emails as well as minimize phishing attempts throughout the big companies holding significant user base, generally targeted by scammers. The Job Posting boards and services are a similar target for scammers where most of the labor force ends up going through the posted positions and provide their personal details, including most personal data and even highly sensitive data such as social security or social insurance numbers which make these boards a target for identity theft scammers.

During my research, I found very few details about security measures taken by these job boards in classification of genuine and fake positions, which results in users generally having to perform manual analysis and review of the posts in order to distinguish between real and fake job postings. Most blogs suggest tips for manual analysis such as - https://www.inc.com/jt-odonnell/no-1-sign-a-job-posting-is-fake.html, however, because of the scale of this problem, an automated solution is necessary, which can decrease costs endured by job seekers as well as businesses in mitigating security risk posed by scam job positions.

This project aims to assist this strategy with AI, using machine learning algorithms to classify job postings and allow both businesses and job-seekers to perform classification of phishing or fake job postings at scale in real-time and yield significant financial and security savings.

Data Requirements

This minimum data requirement for this project includes a dataset which contains:

- Job postings
    - This includes description of a posting and additional text features such as requirements and company profile to be used to determine authenticity of a job posting
- Result Classification labels

These two are primary set of data to be used for training supervised models. Additionally, we would like some categorical features to allow the classification to not be biased or limited to language. For example, some of these features may include:

- Location
- Remote Position

Data

During my research, I found an existing dataset – "Employment Scam Aegean Dataset" curated by researchers at University of the Aegean, in Greece containing 17,014 legitimate and 866 fraudulent job ads published between 2012 to 2014. More details can be reviewed at the university data set website: http://emscad.samos.aegean.gr/

It contains features extracted actual job postings and has been annotated manually by university researchers to assist in classification of Scam Job postings. The data can be accessed in a csv file at the following link: https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction

The data contains the following features, which satisfy our requirements for text data for NLP analysis as well as categorical data for further correlation analysis:

Name	Description
Title	The title of the job ad entry.
Location	Geographical location of the job ad.
Department	Corporate department (e.g. sales).
Salary range	Indicative salary range (e.g. $50,000-$60,000)
Company profile	A brief company description.
Description	The details description of the job ad.
Requirements	Enlisted requirements for the job opening.
Benefits	Enlisted offered benefits by the employer.
Telecommuting	True for telecommuting positions.
Company logo	True if company logo is present.
Questions	True if screening questions are present.
Fraudulent	Classification attribute.
In balanced	Selected for the balanced dataset
Employment type	Full-type, Part-time, Contract, etc.
Required experience	Executive, Entry level, Intern, etc.
Required education	Doctorate, Master's Degree, Bachelor, etc.
Industry	Automotive, IT, Health care, Real estate, etc.
Function	Consulting, Engineering, Research, Sales etc.

Data Assumptions

Assumptions for this dataset includes:

- The dataset acquired is assumed to contain valid data, including actual features and accurate labeling from the data curators

- We assume the data for this class contains enough variation in text to perform NLP techniques as well as statistical analysis.

Data Constraints

Constraints for the data set include:

- Analysis is being performed on the specific features available in this dataset, and therefore may contain bias originating from the location, or by curators.
- The data heavily contains records that are genuine, which would provide limited variance to the model and may result in decreased accuracy or overfitting towards certain keywords available in the limited records

Development Environment

We will be developing this project using Python, taking advantage of numerous available libraries for machine learning and data analysis. We will be performing the analysis tasks in iPython (Jupyter) notebooks, to allow easy review for reference.

Additionally, this project will be deployed on a cloud system as an API endpoint, possible deployment platforms will be reviewed before development is completed and a chosen cloud platform (Azure/Google/AWS) would be chosen to host the API.

Model/Architecture Approach

The project aims to use most popular Supervised Learning algorithms and NLP techniques and analyze the model results in order perform classification accurately:

Feature Extraction: NLP Techniques

- TF-IDF (Term frequency–inverse document frequency)

We will be using this technique to gather relevant features from the text data available on the job postings. The features gathered from this technique would allow us to easily prepare the text rows for the machine learning models to be ran on.

Oversampling Handling:

- SMOTE (Synthetic Minority Oversampling Technique)

We will be using SMOTE to handle the dataset being oversampled. Handling oversampling will allow us to decrease the bias caused by an unbalanced dataset.

Training Models: Machine Learning Techniques

- Naïve Bayes

One of the fastest and easiest to implement conditional probability technique – Naïve Bayes should allow us to prepare a baseline model to be compared against other statistical techniques.

- Stochastic Gradient Descent using Support Vector Machines RBF Sampler

Since the data is being used to create a supervised classifier on a non-linear data, SVMs would provide excellent statistical base and therefore should provide a good accuracy, with improvements possible by hyper parameter updates. However, since the text features generated would require enormous memory, we will try to approximate things using SGD classifier in addition to SVM RBF Sampler.

- Logistic Regression

Logistic Regression is known to provide a good result on word classification problems, we will be analyzing the performance of this technique on the data to gather effect of a regression model on the results.

Testing and Validation Approach

To get the best results, we would be dividing the data set into 2 groups, for testing and training, respectively. Testing samples would be primarily used in analysis, allowing us to perform statistical validation as well as tuning.

We will evaluate the model primarily using:

- Accuracy Scores
- K-fold cross validation
- Confusion Matrices
- F1 Scores

Based on available time additional techniques may be included to get further improvement on the model.

Project Plan

This project tasks include checkpoints for smooth and efficient progress. It includes a task, an effort associated with it and a deadline.
The project has a hard deadline of December 18, allowing us a set a time limited budget of under 200 work hours. Details of categories and tasks are as follows:

Business Understanding and Problem Discovery

Deliverable: Statement of Work V1

Task	Effort (hrs)	Deadline
Perform Problem Analysis	10	20/10/2020
Collect Information about dataset	10	25/10/2020
Set up Repository and required infrastructure	10	28/10/2020
Create Proposal	10	03/11/2020

Data Acquisition and Understanding

Deliverable: Statement of Work V2

Clean and Analyze Dataset (EDA)	20	08/11/2020
Create Data Preprocessing Pipeline	5	09/11/2020
Perform Feature Extraction using TFIDF	20	12/11/2020

ML Modeling and Evaluation

Deliverable: Dataset, Results of evaluation (Included in deliverable above)

Prototype Model – Naïve-Bayes Algorithm	10	14/11/2020
Prototype Model – Logistic Regression	10	16/11/2020
Prototype Model – SGD + SVM	10	18/11/2020
Evaluate Prototypes	10	21/11/2020
Analyze best approach	10	22/11/2020
Refine Model/Architecture	5	23/11/2020

Delivery and Acceptance

Deliverable: Working API in Cloud, Presentation

Develop Software Pipeline	10	24/11/2020

Develop API Endpoint for Result Aggregation	10	30/11/2020
Testing Endpoints and Model	10	05/12/2020
Deploy Model/API to Production	10	12/12/2020
Presentation Preparation and Demo	5	18/12/2020
Total	185	