Job Recommender with Fraudulent Job Filter

Garima Garima	Rajit Bharambe	Samruddhi Khandale	Tanwi Kale
ggarima@ncsu.edu	rbharam@ncsu.edu	sskhanda@ncsu.edu	tkale@ncsu.edu

Project Group Id - P05

1 Background & Introduction

In these uncertain times, employment rate has gone down, thus job seekers are applying to every job posting they see. They can easily fall prey to fraudulent job postings and submit sensitive information or click on virus infected URLs. The major issue job seekers are facing today is finding real and relevant job postings. Currently, there are not many job recommendation systems which filter the fraudulent job postings and recommend the jobs relevant to the user's resume.

Our system filters out the fraudulent job postings before recommending the jobs based on the details provided by the user.

1.1 Literature Survey

We researched a few papers that were relevant to the problem at hand and to the approach that we were considering. In [1], the authors analyse and compare different filtering algorithms and similarity calculation techniques, but the dataset used is very small and restricted in terms of features. This literature also does not touch the topic of fraudulent job detection or classification. In [2], the authors dive into the analysis of a few single and ensemble classifier models. They further evaluate them using various metrics and build a fraudulent job classifier to filter out the fake jobs, but do not utilise the filter in recommending jobs to a user.

2 Proposed Method

2.1 Approach

The system has two phases. In Phase one we will do exploratory data analysis, data pre-processing and create a fraudulent job postings filter. We will train a Decision Tree algorithm for classifying the jobs. Then we will use the filtered job postings from this classifier in phase two to create a job recommendation system. We will take user resume and parse it to pass into the system where we will match it with the non-fraudulent postings and outputs top 10 relevant job postings to the user. The relevant job postings are will be extracted using cosine similarity.

We carried out the following steps in order:

2.1.1 Phase One - Creating a Fraudulent Job Filter

- Conducted Exploratory Data Analysis (EDA) to find the difference in fraudulent and non fraudulent job postings, identifying the correlation between different features and understanding the data.
- Performed feature selection
  - Domain Knowledge - Dropping the features that did not provide any useful information. For example the job_id
  - Informative Value - Identifying the information value of all features and then combining or dropping them accordingly. For example combined company_profile, benefits and requirements with the description

  - Missing Value - Handling the missing by using the backward filling.
- Performed data cleaning on textual features like removing stopwords, punctuation and converting categorical features into one hot encoding. Converted the textual features into vectors using the Count Vectorizer.
- Implemented Decision Tree algorithm, trained on fake and real job postings dataset and used it as a filter to remove fake job postings.

2.1.2 Phase Two - Creating a Job Recommendation System

We performed the following steps:

- The input to this system is the filtered non-fraudulent job postings from the phase one and user's resume
- Parsed and tokenized the user's resume for important keywords
- Converted text into feature vectors using TF-IDF vectorizer
- Matched the keywords from user's resume to non-fraudulent job postings to recommend top 10 job postings with the highest cosine similarity

2.2 Rationale

2.2.1 Phase One - Creating a Fraudulent Job Filter

- Conducted Exploratory Data Analysis (EDA) as expected to understand the patterns and data distribution.
- Feature selection process was based on many factors like
  - Domain Knowledge - Dropping the features that did not provide any useful information. For example the job_id.
  - Informative Value - The informative value we dropped the columns that provided very less information For example, the salary_range attribute had more than 80% of missing values and hence it was dropped. We also combined some textual features company_profile, benefits and requirements were combined with the description as they provided similar information about the company so combining them can give better results in text processing.
  - Missing Value - Missing values in critical attributes are handled by analyzing approaches like forward or backward filling and filling them with the median value. In our scenario, we used backward filling as it proved to be optimal.
- We trained all the machine learning algorithms proven for classification for example Logistic Regression, K-Nearest Neighbors, Support Vector Classifier, Random Forest Classifier and Decision Tree Classifier. As the data was highly skewed towards non-fraudulent job postings, we couldn't use the confusion matrix and so decided to use the AUC-ROC score. Decision Tree had the highest AUC-ROC value and hence was used as a fraudulent job filter classifier.

2.2.2 Phase Two - Creating a Job Recommendation System

- We parsed and tokenized the user's resume for important keywords because we want to find the important keyword from user resume and we want to bring it in the form suitable for TF-IDF vectorizer.
- There are many other feature vectorizers like TF, word embeddings, etc., but our project didn't involve any complex and huge number of features and so, the primitive method of TF-IDF vectorizer proved to be efficient in creating the feature vectors.
- To find the relevant job postings we had the option to choose between Jaccard index and cosine similarity. Cosine similarity performs better with vectorized data and the machine learning algorithms we are using for classification require the same. Therefore, we chose cosine similarity for closely matching the job postings with the user's resume.

3 Plan and Experiment

3.1 Dataset

`Fake and Real Job Postings`, `User resume`

3.2 Hypotheses

Our hypotheses in Phase one before creating a fraudulent classifier was as follows -

- Number of words used in real and fake jobs are significantly different
- Fake jobs are independent of career levels
- There is no recruitment process for fake jobs
- There is one model whose AUC-ROC score is close to 1 amongst all the different models that are evaluated for creating the fraudulent classifier

Our hypotheses in Phase two before creating a Job recommendation system was as follows -

- There cannot be cross industry recommended job postings
- There would be atleast 80% closely matching job postings recommendation for the user's resume.

3.3 Experiments

3.3.1 Experimental Design

We will run a series of experimental steps as explained below in order to answer each of the research questions.

- Number of words used in real and fake jobs are significantly different
  - Plot the graph for number of words used in description for the fraudulent and non-fraudulent jobs postings
- Fake jobs are independent of career levels
  - Analyze distribution of fraudulent jobs on required_experience feature
- There is no recruitment process for fake jobs
  - Analyze distribution of fraudulent jobs on has_questions features
- There is one model whose AUC-ROC score is close to 1 amongst all the different models that are evaluated for creating the fraudulent classifier
  - Train and test job postings dataset using 0.1 split
  - Create machine learning models - Logistic Regression, K Nearest Neighbors, Support Vector Classifier, Random Forest Classifier and Decision Tree Classifier using GridSearch CV
  - Evaluate models using AUC-ROC score
- There cannot be cross industry recommended job postings
  - Use software engineer resume and analyze the recommended jobs by the industry. If there are jobs from non-IT companies looking for software engineer then our hypothesis will be incorrect.
- There would be atleast 80% closely matching job postings recommendation for the user's resume
  - We will test the job recommender system by analyzing the recommended jobs. If there are relevant job positions with less than 80% match, then our hypothesis will be incorrect.

3.3.2 Experimental Design challenges

- The dataset is skewed and dominant with non-fraudulent job postings, so, it was challenging to deal with the empty feature values
- Choosing a method to find an efficient machine learning algorithm for training a model was a difficult job as none of the parameters in the confusion matrix give the desired result
- Deciding upon the features to be dropped and ensuring that their absence does not affect our expected results

3.3.3 Experiment Setup

Programming Language - Python 3.8.0

Platform for collaboration - Google Colab

Python packages
1. Data pre-processing - Pandas, Numpy, re, string, base64, collections, en_core_web_lg, pycountry
2. Data transformation - spacy, sklearn, nltk, scipy
3. Data visualization - matplotlib, seaborn, wordcloud
4. Machine learning models - sklearn
5. Metrics - sklearn

3.3.4 Experiments in Phase one

We performed Exploratory Data Analysis (EDA) on the dataset and observed that the dataset has 18 attributes and 17880 job postings.The distribution of the target as fraudulent or non-fraudulent job postings can be observed from the graph in Figure 1.

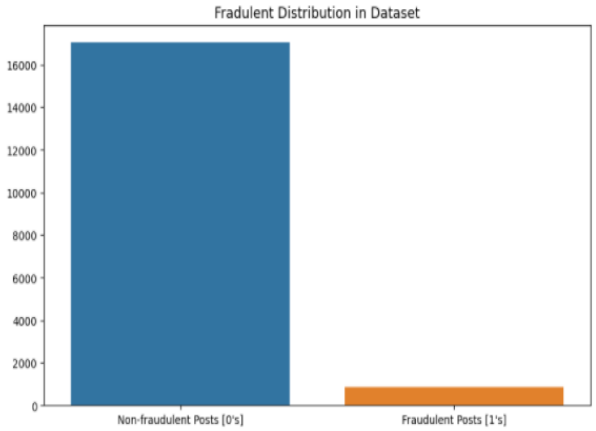

Figure 1: Fraudulent Distribution in the dataset

We did analysis to identify number of words been used fake and real job postings. We had an understanding that it would be different but after plotting it was surprising to find that both the job positing used almost same number of words reaching approximately 1000 words. Then to analyze the data, we checked for missing values in the dataset. Figure 2 shows the outcome as normalized distribution of missing values in all the features. The value ranges from 0 to 1 where 0 indicates all data objects are present and 1 indicates all data objects are missing.

On the basis of missing values we have dropped the columns of salary_range as more than 80% of the data is missing. We also removed as job_id because it does not provide any useful information about the job posting.

We further processed the data by removing duplicate rows and only keeping the first entry amongst the duplicate entries. In this process, 250 entries were dropped.

We combined the columns of company_profile, description and requirements into the description column as all of them provide information about the particular job posting as a whole. This also

Attribute name	Normalized distibution of missing values
job_id	0.000000
title	0.000000
location	0.019351
department	0.645805
salary_range	0.839597
company_profile	0.185011
description	0.000056
requirements	0.150727
benefits	0.403244
telecommuting	0.000000
has_company_logo	0.000000
has_questions	0.000000
employment_type	0.194128
required_experience	0.394295
required_education	0.453300
industry	0.274217
function	0.361018
fraudulent	0.000000

Figure 2: Missing Values per feature in raw dataset

helps us in text processing and keyword matching. Since we combined the company_profile and the requirements columns into description and they were dropped.

We split the attribute of location into two attributes country_name and cities using the pycountry package for better analysis and keyword matching.

We handled missing values by using backward filling on the attributes - employment_type, required_experience, required_education, industry, function. The very few (<5) records that still had NaN values were dropped after this step.

We then performed data cleaning on textual features of description by converting the text to lowercase, removing stop words, special characters, links and words containing numbers. In-order to apply the machine learning algorithms on the dataset, we require attribute values to be float. Thus, we tokenized the description feature as it contained text and applied CountVectorizer. Also, we transformed the categorical features like required_experience, required_education, title, function, industry, country_code, city, department and employment_type using One Hot Encoding.

The final dataset which we would use for model evaluation contains 11272 records and 857 features. We have selected 90% of data for training and 10% of data for testing using train_test_split.

The Machine Learning models we evaluated are Logistic Regression, KNearestNeighbour, Support Vector Classifier, Random Forest Classifier and Decision Tree. For all these algorithms we have created their specific parameter grid. Then we used GridSearchCV for each model, their respective parameter grid, cross validation as 10 and scoring as auc_roc curve to identify the best parameters. Models and their parameter grid and output is defined below:

- For **Logistic Regression**, we created a parameter grid of L1 and L2 norm penalties and regularization ranging from 0.00001 to 10000. The best values for each parameter came out to be as follows - norm penalty = L2, regularizer = 0.1 and auc_roc score = 0.779
- For **KNearestNeighbour**, we created a parameter grid of nearest neighbour hyperparameter ranging from 2 to 23. The best values for each parameter came out to be as follows - Nearest neighbour = 22 and auc_roc score = 0.6
- For **SVC**, we created a parameter grid of kernel hyperparameter having the values 'linear' and 'rbf'. The best values for each parameter came out to be as follows - kernel = 'linear' and auc_roc score = 0.819
- For **Random Forest Classifier**, we created a parameter grid of n_estimators hyperparameter having the values[1, 2, 4, 8, 16, 32, 64, 100, 200]. The best values for each parameter came out to be as follows - n_estimator = 200 and auc_roc score = 0.74
- For **Decision Tree**, we created a parameter grid of criteria hyperparameter having the values 'Entropy' and 'Grid'. The best values for each parameter came out to be as follows - criteria = 'Entropy' and score = 0.878

3.3.5 Experiments in Phase two

In phase two of the system, the user resume is taken as input in the form of a .docx file. This file is parsed to remove stopwords, punctuation and special characters. We then tokenized the parsed input to extract keywords and then converted them into feature vectors using the TF-IDF vectorizer.

The input from Phase one (non-fraudulent job postings) is already in the form of TF-IDF vectors. The job postings vectors obtained from Phase one and the user resume vectors obtained parsed in this phase are compared using cosine similarity. The 10 job postings that give the highest cosine similarity are recommended to the user. These job postings denote the closest and the most relevant job postings to the input user resume.

4 Results

For phase one of the system, we evaluated Logistic Regression, KNearestNeighbour, SVC, Random Forest Classifier and Decision Tree models using AUC-ROC score as mentioned in experiment section. We observed that Decision Tree gave the best performance as it has the highest AUC-ROC score and KNN gave the worst performance. Refer to Figure 3 to see the performance plot. Hence, we created the Fraudulent job filter using Decision Tree model in phase one of the system.

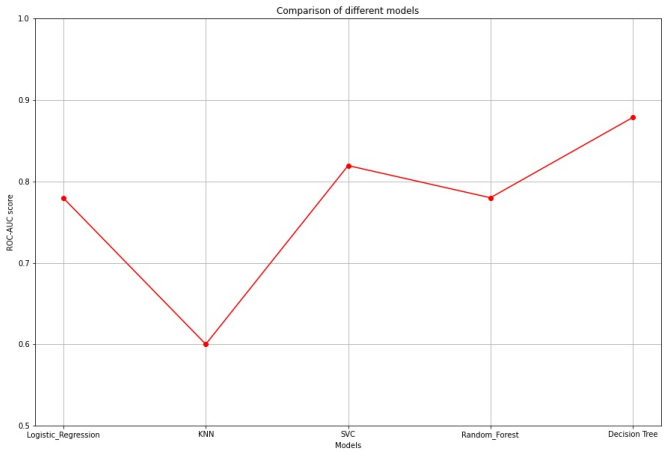

Figure 3: ROC curve for different machine learning algorithms

The number of words used in real and fake jobs are not significantly different as shown in figure 4. Thus, fake jobs description is not lengthy as compared to the real ones.

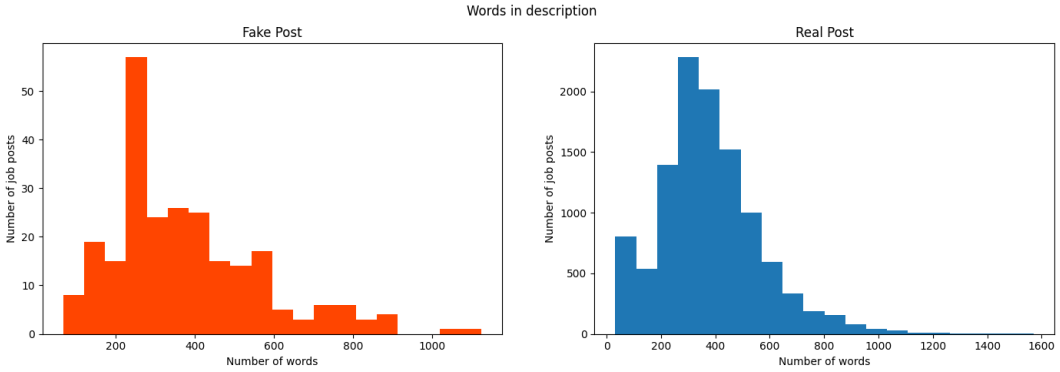

Figure 4: Number of words in job Postings

As mentioned in the hypothesis, fraudulent job postings are independent of career levels as they are not targeting just one level but spans across all possible levels as plotted in figure 5. Also, according to figure 6, fraudulent job postings consists of recruitment process for 40% of the fake posts. For Phase two of the system, the job recommender took the user resume [5] and the non-

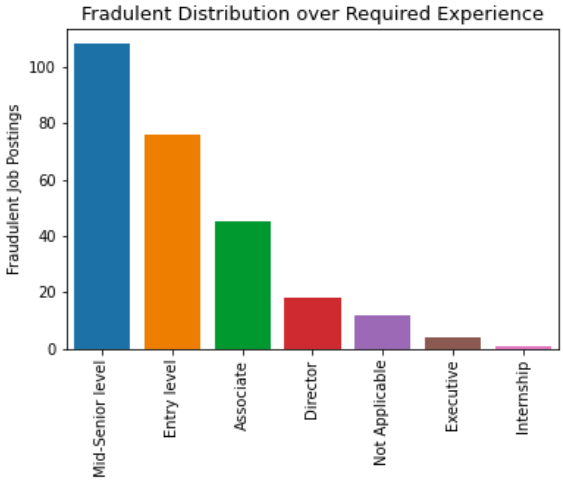

Figure 5: Distribution of fraudulent job Postings using Career levels

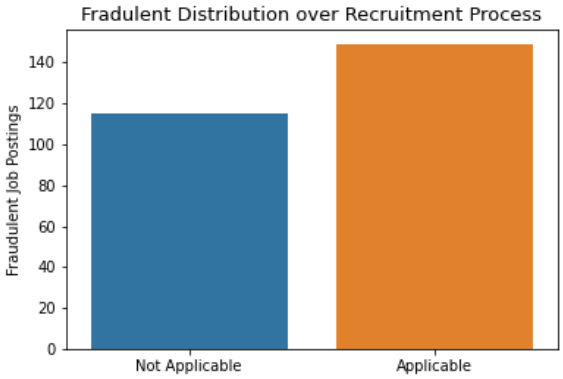

Figure 6: Distribution of fraudulent job Postings using Career levels

fraudulent dataset (obtained from Phase one) as the input. Cosine similarity was used for finding the job postings that were most similar to the user resume and out of all the postings, top 10 job postings with the highest cosine similarity were recommended as shown in Figure 7. 40% closely matching job postings were recommended for the user's resume which also include cross-industry job postings.

	Job_ID	Title	Location	Departme	Company	Descriptio	Requirement	Employme	Experienc	Education	Industry	Function	Fraudulen	score
0	14508	Business Development M	US, MA, Boston		We Provic	(We have	more than 1500	Full-time	Mid-Senic	Bachelor's	Computer	Project M	0	0.422643
1	10489	Digital Producer	US, ID, Boi	Digital	Since 197	We are loc	Solid technical	Full-time	Not Appli	Bachelor's	Marketing	Project M	0	0.4125
2	15447	Business Development M	DE, NW, DÃ¼sseldor		34 OFFICE	We are loc	Strong passion	Full-time	Not Appli	Unspecifi	Informati	Sales	0	0.308312
3	8733	Development Manager	NZ, , Wellington		SilverStrip	We're look	ing for an energetic, gregarious person to be our Development Man						0	0.381054
4	17302	Care Worker (Personal As	GB, MLN, Edinburgh		Social Car	We are loc	You are require	Full-time	Associate	Vocationa	Hospital &	Health Ca	0	0.36155
5	15743	Data Solutions Consultant	US, OR, Portland		Can data b	About Sea	5+ years of tech	Contract	Mid-Senic	Bachelor's	Computer	Consultin	0	0.359535
6	13669	Customer Experience Cha	AU, VIC, Ca	Customer	Brosa is a	Company (	Brosa Customer	Experience	Champions	play a	pivotal	role in our suc	0	0.355879
7	8464	Direct Sales Opportunity	US, VA, Arlington			In the Dire	About Us	Full-time	Entry level		Internet	Sales	0	0.352478
8	15925	Customer Service Represe	US, IL, Libertyville		Handi Ran	Handi Ram	Skills and Requ	Full-time	Entry leve	Bachelor's	Consumer	Customer	0	0.350078
9	7728	Customer Service Associat	US, MA, Boston		Novitex Ei	The Custo	Minimum Requ	Full-time	Entry leve	High Scho	Facilities	Customer	0	0.349831

Figure 7: Recommended job postings for the input user resume [5]

4.1 Critical Evaluation

For phase one of the system, as mentioned in the hypothesis, we observed one model - Decision tree whose AUC-ROC score is close to one (0.878). Our classifier is able to correctly distinguish between positive and negative classes for 87% of the times. However, we did not get a model which has a perfect AUC-ROC score of one because the dataset is skewed and is dominated by non-fraudulent job postings. Also, decision tree performed better than other models because it handles a large set of categorical values in training data well, supports non-linearity and is more flexible.

The number of words used in real and fake job postings were hypothesised to be significantly different. However, they are approximately equal and it answers the question that we can not differentiate between fake and real postings by only considering the count of words in the description as the fake job postings do not add more words and talk more about the posting to make it feel genuine.

For phase two of the system, as mentioned in the hypotheses, there should have been 80% match in between the recommended job postings and the user's resume. However, we observed only a 40% match largely because of the fact that the chosen dataset consists of postings from a variety of industries and not just the one in which the user had experience in.

We observed that the user's resume mainly consists of a Business Analyst profile. However, our system recommended non-Business Analyst positions such as Care Worker and Digital Producer. After comparing the postings with the resume, we identified that the keywords in the description of the postings matched with the resume and were majorly responsible for recommending cross-industry jobs. A follow-up strategy for this would be increasing the size and quality of the dataset by gathering more job instances in order to adequately represent all the industries.

5 Conclusion

The project gave us insights into the importance of feature extraction, conversion of text features into vectors for improving the model's performance and choosing the right method to handle missing values. Also, we learnt how critical proper analysis and comparison of different models is as we can not rely on a single model due to data bias. We learnt how important it is to understand the dataset in order to find an efficient solution to the set concrete hypotheses.

Future scope of the project can include taking in user preferences such as location and salary range, weighing how important a feature is for the user and then recommending the results based on that.

6 References

[1] Y. Zhang, C. Yang and Z. Niu, "A Research of Job Recommendation System Based on Collaborative Filtering," *2014 Seventh International Symposium on Computational Intelligence and Design*, Hangzhou, 2014, pp. 533-538, doi: 10.1109/ISCID.2014.228.

[2] Shawni Dutta, Prof.Samir Kumar Bandyopadhyay *"Fake Job Recruitment Detection Using Machine Learning Approach"* International Journal of Engineering Trends and Technology 68.4(2020):48-53.

[3] Sharad Jain, *https://towardsdatascience.com/predicting-fake-job-postings-part-1*

[4] Sharad Jain, *https://towardsdatascience.com/predicting-fake-job-postings-part-2*

[5] *https://docs.google.com/document/user-resume*

7 Code

Github link - `https://github.ncsu.edu/engr-csc522-fall2020/P5`