Title : Fake Job Posting Prediction

Team members : Sanket Badhe, Isha Raju, Akshay Sitlani

Abstract: Fake job postings are a nuisance and very common. These postings are put up for various reasons. For example, fake jobs are posted by cyber criminals to trick job seekers, hiring managers to assess available talent pool, random people to just add people to their network, unfair hiring practices and the list goes on. The job market is already very tough for job seekers due to Covid-19 situation. Fake job postings add to their frustration and result in loss of precious time and energy which could have been used on more productive applications.

In today's times of big data it is very difficult to manually assess each and every job that is posted. A faster and more efficient method to identify these would significantly help job-related search engine companies and their users. This project aims to automatic identification of fake job postings from job portals to prevent all the above mentioned problems. In this project, we used text mining techniques in conjunction with manually generated features for input to our data mining algorithms. We generated the features based on observed different patterns in fake and genuine job postings. Finally, we build models based on our curated feature to classify given job posting as fake or genuine. We used popular data mining techniques such as Multinomial Naive Bayes, Logistic regression, Decision tree, Random forest and Gradient boosting for model building. Our best performing model on unseen data was Random forest.

Objective Statement: Classification of the given set of job postings into "Real" or "Fake" based on text and numeric data using machine learning algorithms.

Data: In this project, we used Employment Scam Aegean Dataset(University of the Aegean) dataset that consists of job descriptions and their meta-information for 18k job postings. A small proportion of these descriptions are fake or scam which can be identified by the column "fraudulent".
An interesting part of the project is that the data we are working on is a highly imbalanced dataset, the dataset contains 18k job postings out of which only 800 are fake job postings. I have shown that in the Figure1.

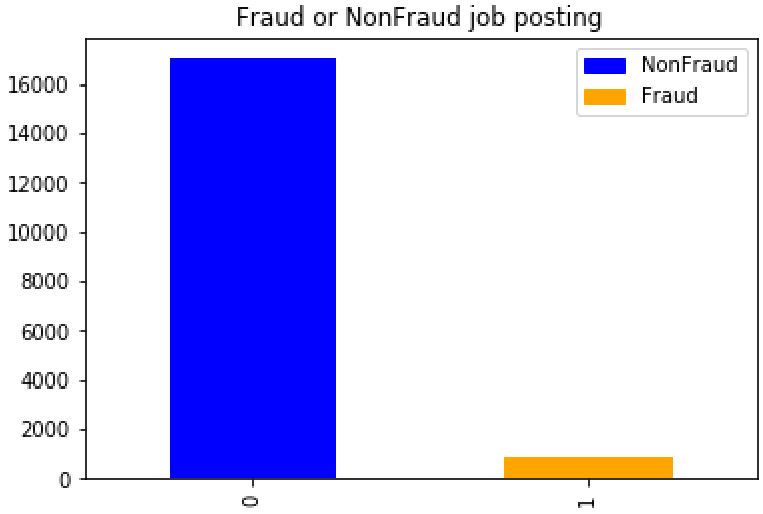


Figure1, Imbalanced data

The data consists of the following features: Unique job ID, title of the job ad entry, geographical location of the job ad, corporate department (e.g. sales), salary range (e.g. $50,000-$60,000), a brief company description, detailed description of the job ad, enlisted requirements for the job opening, enlisted offered benefits by the employer, position to telecommute, presence of company logo, 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.), fraudulent or not.

Feature Engineering, Importance, Exploration and Experimentation:

In order to perform feature engineering and data exploration, first, we removed all the punctuation and stopwords using sklearn learn library. stopwords are the words like ["you","me", "and", "the", "him", etc] which are mostly uninformative in representing the content of a text and which may be removed to avoid them creating unnecessary signals for prediction. Most of the variables in the data were categorical text related hence we replace Null values by "Not Available" or "Other" depending on context.

In the next steps, We generated the feature based on two techniques: Bag of Words Modelling and Creation of ruleset by different criteria. Let's look at it one by one.

Bag of Words Modelling: In this technique each individual word token occurrence frequency is treated as a feature. Vector of all the token frequencies for a given posting is considered a multivariate sample. We tried two types of bag of words modelling.

- CountVectorizer: CountVectorizer works on Terms Frequency, i.e. counting the occurrences of tokens and building a sparse matrix of documents x tokens.
- TF-IDF: term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

First we created a new feature with the name ‘text’ by concatenating all the columns with text columns namely ‘title’, ‘company_profile’, ‘description’, ‘requirement’, ‘benefits’, ‘fraudulent’. Then we perform Counting word occurrence for each job post. Next, we calculated the tf-idf score based on output of the countvectorizer. The reason behind this approach is that keywords and important signals will occur again and again in each class. We used sklean’s CountVectorizer(), TfidfTransformer() in-built function for this task.

We didn't find any improvement in our test auc score due to tf-idf technique, hence we just used the CountVectorizer bag of word technique for the final modelling part.

Binary Rule based Features: After Bag of Words Modelling, we created features based on developing ruleset by different experiments. In this section we assume some hypotheses and perform experiments to check that hypotheses are correct or not. Some of our hypothesis were:

1) Scammers are very unlikely to compose a meticulous and lengthy job description, profile, requirements etc.
2) Scammers use certain phrases or spam words to attract people of a certain category. These words are “Easy Money”, “Work from home”, “No Experience”, “Only weekdays”, “Extra Money”, “Extra Income”, “flexible timings”, etc.
3) Spammer gives external application prompts links on their postings using keywords such as “apply at”, “send resume”, “visit here”, etc.
4) Most of the fake jobs have very little education requirements to attract many people. Hence education requirements will be 'High School or equivalent','Some High School Coursework','Some College Coursework Completed' etc.
5) Most of the fake job postings are from the US.

In the table1, we have given a detailed description of all the features we have created and our findings.

Note: we used balanced data for the Table1 and Figure2 so that comparison should be appropriate. Hence we randomly sampled 866 Non Fraud(genuine) job posts.

Feature Name	**Feature description**	**Findings**
is_company_profile	Does the posting consist of a company profile?Flag postings with a company profile as 1 and others as 0	It is observed that most (700+) of the genuine postings have their company profile mentioned. Whereas only(~200) fraudulent postings mention it.
is_company_profile_short	Flag postings with a company profile with <= 10 characters as 1 and others as 0	It was observed that many fraudulent(~ 600) posts had short company profiles and very few genuine posts had short company profiles(~ 200)
is_company_profile_long	Flag postings with a company profile with > 100 characters as 1 and others as 0	It was observed that many genuine(~ 300) posts had long company profiles and very few fraudulent posts had long company profiles(~ 100)
is_company_description_short"	Flag postings with a job description with >40 characters (removing punctuations) as 1 and others as 0	It was observed that more fraudulent posts had shorter descriptions than genuine posts.
is_company_description_long"	Flag postings with a job description with > 100 characters (removing punctuations) as 01and others as 0	It was observed that more fraudulent posts have shorter job descriptions than the genuine posts

is_company_requirements_short	Flag postings with a requirements/ qualifications with <= 10 characters (removing punctuations) as 1 and others as 0	It was observed that more number of fraudulent posts(~300) had shorter requirements than genuine posts(~150)
money_in_description	If the title or job description consist of the word “Money” then flag it as 1	It was observed that there are more farudulent posts with “money” in its title or description than genuine posts.
telecommuting(work from home)	Whether it is a telecommuting(work from home) position: if yes it takes value : True	It was observed that telecommuting positions were more common in fraudulent posts than genuine posts
has_company_logo	True, if the company logo is present in the post	It was observed that only few fraudulent postshave a company logo(~200) whereas most genuine posts had the company logo in the posting(~700).
has_questions	True, if screening questions are present in the post	It was observed that few fraudulent posts listed out screening questions whereas most genuine posts had screening questions.
spamword_list	The following keywords were considered as likely to be spam : 'free', 'earn', 'fun', 'guarantee', 'weekend', 'home', 'easy', 'incentives', 'week', 'incentive', 'commission', 'flexible', 'cash', 'income', 'no experience', 'online'. If the posting consisted any of these it was flagged as 1	These spam words were more frequent in fraudulent posts than genuine posts
prompts_external_application	The following prompts to external application were considered : 'visit here', 'apply here', 'apply at', 'send resume', 'call', 'email', 'contact', 'url', 'apply online','click here','send','apply' to create a flag with value 1	These prompts to external applications were more frequent in fraudulent posts than genuine posts
lower_education	If the posting requires only 'High School or equivalent','Some High School Coursework','Some College Coursework Completed','Unspecified','Vocational' and doesn't require higher education to qualify for the job then flag = 1	It was observed that there were more fraudulent job posts than genuine posts that did not require higher education .
US_or_not	If the country of job posting is US then flag it as 1 otherwise 0	It was observed that there were more fraudulent job posts (approx 745) than genuine posts(approx 500) in the US

Table1, Feature Engineering

In the figure2, we have plotted few of the generated binary features for fraud and non-fraud postings.

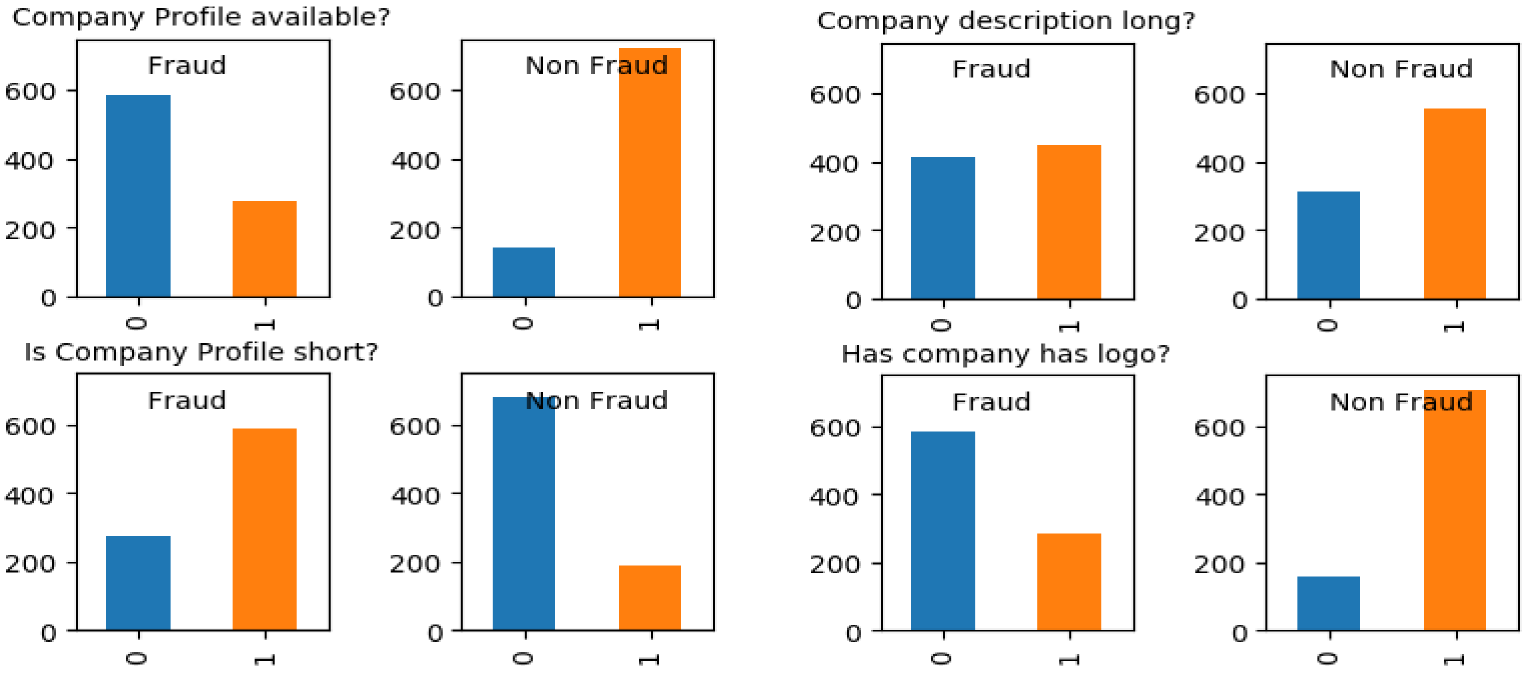
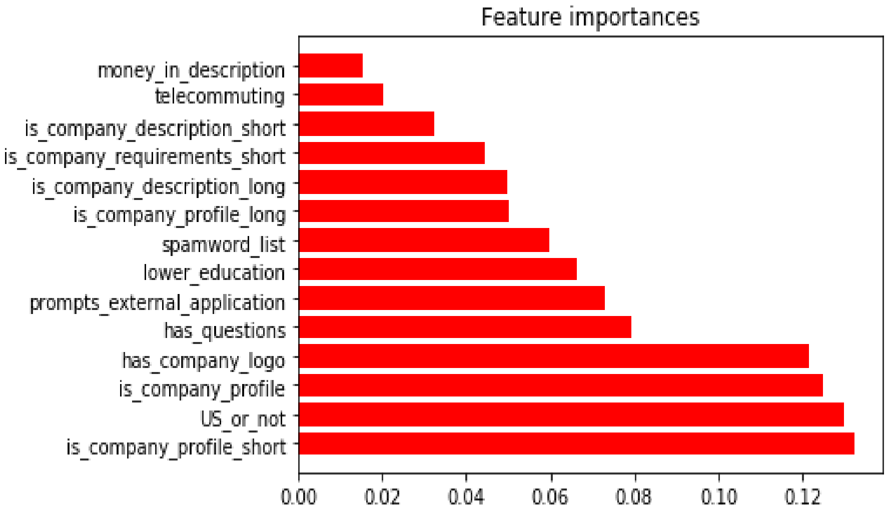
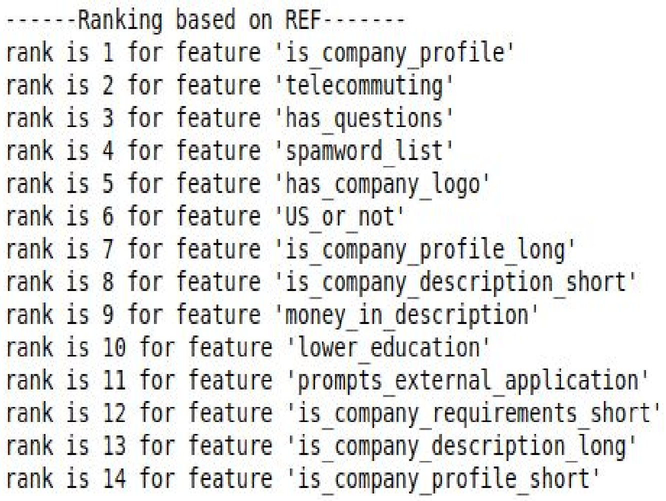


Figure 2, Generated feature comparison

Feature Importance: We tried two feature importance methods based on algorithm techniques. In tree based method quality of split at any given node is quantified by *gini impurity.* As shown in figure 3, we calculated feature importance based on averaging the decrease in impurity over trees for Random Forest. Next, for calculating feature importance for logistic Regression based model, we used backward subset selection model based on gini index. As we just have only 14 features in our data which are not hard to interpret hence we used all the features for model building. Also, Removing the least importance features was affecting our test auc score by 1-2%.





```
------Ranking based on REF-------
rank is 1 for feature 'is_company_profile'
rank is 2 for feature 'telecommuting'
rank is 3 for feature 'has_questions'
rank is 4 for feature 'spamword_list'
rank is 5 for feature 'has_company_logo'
rank is 6 for feature 'US_or_not'
rank is 7 for feature 'is_company_profile_long'
rank is 8 for feature 'is_company_description_short'
rank is 9 for feature 'money_in_description'
rank is 10 for feature 'lower_education'
rank is 11 for feature 'prompts_external_application'
rank is 12 for feature 'is_company_requirements_short'
rank is 13 for feature 'is_company_description_long'
rank is 14 for feature 'is_company_profile_short'
```

Figure 3, Feature importance

Modelling, Comparison, Validation and Results:

Due to an imbalance dataset, we train our model using sampling as well as without sampling of the train data. We didn't find any improvement in our test results due to sampling techniques(Check table2 and table3 for comparison). We tried the downsampling of the non-fraud posting to balance our dataset. Most of the columns have text data, hence applying SMOTE was not an easy task. It's ongoing research in the industry to generate synthetic data for text based features. Hence, we just focused on downsampling methods.

We performed the stratified 5-fold cross validation for model selection. Stratification ensures that each fold represents the original distribution of the data. I.e. If your original data have 100 observations, 75 for class A and 25 for class B, Then each fold has 15 of class A and 5 of class B for each fold. We used StratifiedKFold in sklearn for this purpose.

We selected the optimal hyper parameter for tree based algorithms using Grid Search using cross validation. Where we provided different parameters and grid search found the best hyperparameter by trying all permutation and combination. Sklearn provides GridSearchCV for this purpose. We were not able to run Grid Search for gradient boositng due to computational limitations.

We are assuming that our system will be fully automatic and we did not use any human intervention. Our main objective in this project was to reduce false positives, because this system will automatically remove all positives(fraud) from the job portal and we do not want to remove the false positive(i.e. True job posting) from our job portal. Also, if we predict some of the fake jobs as genuine it's not a serious issue. Hence, our main focus was on improving the precision score. We used default 0.5 probability as the threshold for assigning the classes for training on imbalanced data and 0.98 for training on balanced data. We calculated AUC score by passing input parameters to the function as the predicted probability and true class.

In table2, you can see the test accuracy of different algorithms. All the accuracy is tested on imbalanced data i.e. original format.

Technique	AUC score	F1 Score	Precision Score	Recall Score	Accuracy Score
Multinomial Naive Bayes	0.94	0.61	0.97	0.46	0.97
Logistic Regression	0.97	0.75	0.99	0.58	0.98
Decision Tree	0.87	0.80	0.85	0.73	0.98
Random Forest	0.98	0.81	0.99	0.69	0.98
Gradient Boosting	0.97	0.80	0.95	0.70	0.98

Table2, Trained without sampling and tested on imbalance

Technique	AUC score	F1 Score	Precision Score	Recall Score	Accuracy Score
Multinomial Naive Bayes	0.96	0.5	0.36	0.83	0.92
Logistic Regression	0.96	0.50	0.87	0.32	0.96
Decision Tree	0.87	0.37	0.24	0.74	0.87
Random Forest	0.98	0.46	0.98	0.30	0.97
Gradient Boosting	0.96	0.54	0.89	0.39	0.97

Table3, Trained with sampling(downsampling) and tested on imbalance

From the above table you can see that Random Forest provided the best auc score as well as precision score. Hence, we selected the random forest as our final modelling technique.

Conclusion: We can conclude that we can correctly classify the job posting as fraud or genuine using data mining and text mining techniques. We generated different features using text mining as well as rule based features. We performed downsampling to tackle the imbalanced data, however the results were better for training on imbalanced data. Other Upsampling methods like SMOTE can be explored. Currently SMOTE for free text data is ongoing research topic. We tried the countvectorizer and TF IDF bag of words modelling technique. However, we can explore Glove - word2vec for representing text in vectors and see if it can improve the model.

References:

1) Data: http://emscad.samos.aegean.gr/
2) https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction.text
3) https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html
4) https://machinelearningmastery.com/feature-selection-in-python-with-scikit-learn/
5) http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.323.8516&rep=rep1&type=pdf