Text Classification for Fake Job Posting

Jiaqi Jiang, Jingyu Zou, Yewon Kim

Abstract

Our project focuses on using classification models such as Logistic Regression, Random Forest, XGBoost, and LSTM (Long short-term memory) to identify fake job posts based on textual information and meta-information of the job postings. We want to find an appropriate resampling technique for the imbalanced dataset, and test whether bringing in external reference data would improve the overall performance of our model. We have obtained relevant datasets, completed data cleaning, and built a preliminary model using Random Forest and LSTM with a resulting f1 score of 0.78.

1 Goal

For this project, we would like to build an effective classification model using advanced neural network models and bring in additional predictors to identify fake job postings. Due to the imbalanced nature of our dataset, we want to test our resampling techniques, and specifically augment our data by generating additional data samples for minority classes. In addition, we are going to implement a semi-supervised model to test whether bringing in additional unlabeled data would boost the overall performance of the model. We would like to compare models' performance between methods that use neural networks versus non-neural network models. For the neural network models, we plan on experimenting with added layers, and finetune the hyperparameters for optimal performance.

2 Progress Made

2.1 Data

Our project uses the "Real or Fake: Fake Job Description Prediction" dataset from Kaggle, which contains 17,880 job descriptions across 18 features with disk size of 50.1 MB. [1] In addition, we used the government census data, which contains the average employment and salary across different states, as the ground truth for location and salary range. [2] This file has 1,380 rows with a disk size of 190KB for nationwide data and 36,900 rows with a disk size of 5.73MB for statewide data. Refer to **Table 1** for the summary statistics for the dataset.

Category	Name	Missing		Unique
		#	%	
Index	job_id	0	0.00%	
Response Variable	fraudulent	0	0.00%	
Contextual	company_profile	3308	18.50%	
	description	1	0.01%	
	location	346	1.94%	
	department	11547	64.58%	
	title	0	0.00%	
	benefits	7210	40.32%	
	requirements	2695	15.07%	
	salary_range	15012	83.96%	
Binary	telecommuting	0	0.00%	2
	has_company_logo	0	0.00%	2
	has_questions	0	0.00%	2
Categorical	required_education	8105	45.33%	6
	required_experience	7050	39.43%	7
	function	6455	36.10%	38
	industry	4903	27.42%	132
	employment_type	3471	19.41%	6

Table 1: Summary Statistics

Examples of text columns for both real and fake job postings are shown below in **Table 2**. In order to clean the text columns in our data (i.e. company profile and job description), we removed special characters, converted each word to lowercase, and removed stopwords. We then tokenized the texts and converted each attribute into a vector form, padded the vectors for further analysis. For missing data in text columns, we filled the data with string "not provided". For the location column, we split the comma separated text into three columns (Country, State and City). We eliminated the City column because most of them are unique values.

The department column has some data with special characters (i.e. "@ ecgstudio — process improvement specialists"). The initial EDA shows the

Attribute	Example
Company Profile-Real job	If working in a cubical seems like your idea of hell then joining our awesome startup team might be the opportunity youÄôve been waiting for.Come join the TradeGecko team, weÄôre a Singapore head-quartered company, weÄôre ventured backed and weÄôre growing fast...
Description-Real job	We are currently expanding our Marketing Department and we are looking to hire a Content Marketeer with a flair for developing relations with media, partners and customers.The main areas of responsibility will be:- Content creation...
Company Profile-Fake job	A Resource Hub for Top Recruiters, Client Companies amp; Career Opportunities
Description-Fake job	Director of Engineering HMA Securities Products — San Jose, CAReporting to the VP of Service Provider Engineering, the Director of Engineering will have responsibility for managing the successful development and deployment of the companyÄôs Security products and solutions...

Table 2: Example data points

job postings are more likely to be fake if it contains special characters (15.0% vs 4.6%), shown in **Figure 1**. So we use a separate binary column to indicate if there are special characters in the department column. In addition, jobs are more likely to be fake if the postings have no company logos (15.9% vs 2.0%), shown in **Figure 2**. The benefit column was given in text, but the text did not provide benefit information well enough for text analysis, as many of data have been parsed incorrectly (i.e. some job description parsed in the benefit columns). So we evaluated this column by checking if it mentioned about 3 major benefits, health insurance, life insurance, and retirement plan, and then assigned binary columns for each. We assigned another category to indicate the missing data.

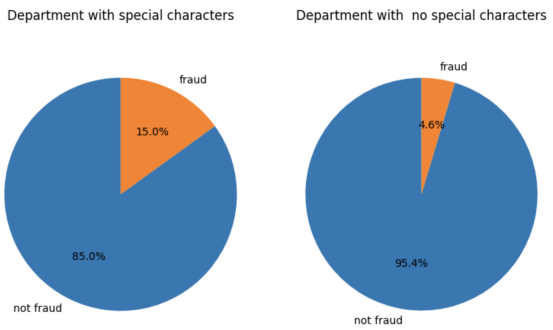

Figure 1: Department characteristics

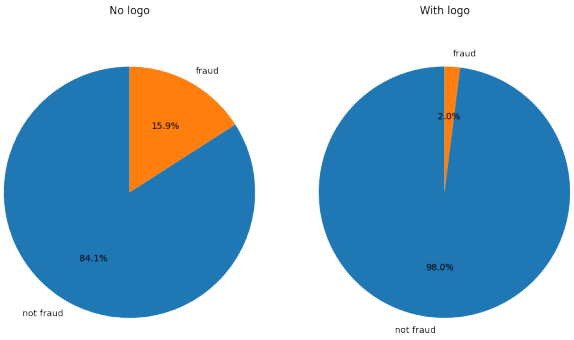

Figure 2: Logo characteristics

Our binary variable columns are already very clean and have no missing value. For each of the categorical features we break down the category into binary columns. Due to the high percentage of missing value, we use an additional binary column for each categorical feature to mark if the data is missing.

The salary was given in a range in our data. Instead of using the numerical value of salary as it is, we assigned some categorical labels to validate this salary range (e.g. ambiguous range given, salary off too far from the external data). We first converted the salary in range into the median value of its range. In this process, any hourly/weekly/monthly rate was also converted to annual rate for consistency of data. For those with unreasonable range (e.g. range between 0 to 1,000,000), we assigned a binary column to indicate it. Another binary column was used to indicate the missing salary.

2.2 Methods

Within our data, only 866 out of 17,880 job postings are fake (5% fake), making our dataset extremely imbalanced. Without using any resampling techniques, we've been getting extremely high accuracy scores, but low f1-scores, as the models are more likely to predict job postings to be real rather than fake. We experimented with both oversampling and undersampling techniques. While the undersampling method yielded relatively high f1-scores, we believe that only training on 1,600 data points could potentially lead to information loss and poor performance when tested with new data. Therefore, we decided to implement the oversampling technique by randomly resample examples in the minority class throughout our models.

Subsequent to data cleaning, we implemented

Logistic regression, Naive Bayes, XGBoost, Random Forest, as well as LSTM to evaluate the performance of our classification model based on the f1 score. The resulting f1 scores using each model are shown in **Table 3**. Prior research on fake job detection only utilized non-neural network models. [3] After comparing the results, we determined that the job description column is a stronger predictor when it comes to classifying real versus fake job postings, and LSTM seems to outperform the other models.

Model	**F1 Score**	
	Profile	**Job Desc**
Naive Bayes	0.38	0.29
Logistic Reg	0.39	0.62
Random Forest	0.39	0.64
XGBoost	0.39	0.30
LSTM	0.39	0.72

Table 3: Company Profile & Job Descriptions

For the salary column, prior research has mostly analyzed the salary within the given dataset as numerical values. Upon that, we decided to analyze the salary further by comparing with the external data and checking if the given salary was in a reasonable range. We used the government statistics of national/state occupation and salary published in 2019 to validate the salary for US job postings based on the national/state average. We matched two datasets by state and department, and then indicated if the difference is more than twice using a binary column. If state location is not provided, we used the national average to compare instead. We used Logistic Regression, Naive Bayes, XGBoost and Random Forest method to classify the job postings using the binary and categorical data. After cross validation and hyper-parameter tuning, we found Random Forest Models outperform other models, shown in **Table 4**.

Model	**F1 Score - Categorical**
Naive Bayes	0.14
Logistic Reg	0.38
Random Forest	0.78
XGBoost	0.37

Table 4: Categorical Data

2.3 Preliminary Results

Prior research done on recruitment fraud detection utilizes the same dataset as we used. [3] Researchers decided to create a balanced dataset by randomly selecting 450 real postings and 450 fake postings that contains the most information (with little to no missing values). After careful consideration, our team believes that 900 data points don't necessarily represent the extent of information contained in the large dataset. Hence, we didn't find the undersampling method appropriate. The model that performs the best set forth in the paper is Random forest. Therefore, we have decided to implement a random forest classifier as the baseline model. The f1-score obtained for this model is 0.63.

Our preliminary model takes into consideration contextual data, meta-information, as well as externally mapped reference from each data point. We implemented LSTM for text data, and Random Forest with GridSearch for externally mapped and categorical data. We obtained the probability output for both models, and we then use GridSearch on the validation set to tune the weight and threshold. The tuning result shows best performance on the validation set when the weight for the LSTM and weight for mapped and categorical data to be 0.89 and 0.11 respectively, with threshold to be 0.44.

We obtained a f1-score of 0.78 on our test set for the final prediction. Comparison of the preliminary model with the baseline model is shown in **Table 5** and **Table 6**.

Baseline			**Our Model**		
	0	1		0	1
0	2539	6	0	2518	27
1	72	65	1	33	104

Table 5: Confusion Matrix

	Precision	**Recall**	**F1-Score**
Baseline	0.92	0.47	0.63
Our Model	0.79	0.75	0.78

Table 6: Accuracy Score

Overall, given that our preliminary model has a higher f1-score, we believe that our current model outperforms the baseline model. We got less false negatives which caused higher recall, but slightly

more false positives which brought lower precision, compared to the baseline model. The increase in recall outweighed the decrease in precision, which consequently led to the higher f1-score for our preliminary model. Given this result, we believe that neural network models tend to perform better on text data than non-neural network models.

We believe that the reason why our model doesn't have a very high f1-score is largely caused by many missing values in the dataset, as well as the dataset being extremely imbalanced. Our current method uses neural networks on long contextual columns and random forest on other categorical columns. We then assign weights and threshold to combine probabilities obtained from both models to get the final prediction. We realize that this might not be the best method since weights and threshold require a lot of tuning, and the validation set is imbalanced due to the nature of our dataset, which might lead to overfitting.

Since neural network models tend to perform better, and are generally more efficient than random forest models, we think it might be better to perform dimension reduction on categorical data and combine these attributes with the textual data, and later feed them into the same neural network models.

2.4 Work Division

Team Member	Task
Jiaqi Jiang	Exploratory data analysis Data cleaning- binary Model implementation
Ye Won Kim	Data cleaning- external data Model implementation Hyperparameter tuning
Jingyu Zou	Data cleaning- text Model implementation Combine model result

3 Future Plan

In the short term, we plan to perform more feature engineering by extracting meaningful features from our current data in order to improve the algorithm. We already developed a code snippet to scrap additional job postings from Indeed, and plan on using those as additional unlabeled data points for semi-supervised learning and gain more understanding of the job posting population.

In the long term, we are going to further augment the current data by implementing a language model using only fake job posts, and artificially generating additional data samples in order to overcome the imbalance in the original data. We then plan on implementing BERT for classification, where words are represented based on their context due to the model's bidirectional nature. We expect our final model to be more complex with a higher f1-score. Afterwards, we will finetune the hyperparameters, changing the weights assigned to the contextual versus categorical variables to obtain a better performing model.

So far, our team believes that we are on a good track. In case additional questions pop up, we plan on utilizing office hours for guidance. The timeline for future task as well as work division is shown below.

Time	Task	Responsible
04.03 -04.09	Feature engineering, Implement semi supervised learning	Jiaqi Jiang Jingyu Zou
04.10 -04.16	Data augmentation, Implement BERT or ELMo for classification	Yewon Kim
04.17-04.20	Finetune hyperparameters, Set up code repository	All
04.20-04.23	Finish final report	All

References

[1] S Bansal. *[Real or Fake] Fake Job Posting Prediction*. 2020.

[2] U.S. Bureau of Labor Statistics. *May 2018 State Occupational Employment and Wage Estimates*. 2019.

[3] Kolias C. Kambourakis G. Vidros, S. and L. Akoglu. *Automatic Detection of Online Recruitment Frauds: Characteristics, Methods, and a Public Dataset*. Number 9(1). Future Internet, 2017.