Data Preparation and Analysis of Fraudulent Job Postings

By Alisha Khan, Anette Volkova, Tiffany Wong

Executive Summary

Key Research Issue

The issue we want to research is fraudulent job postings. With the job market shrinking and expanding, a lot of applicants are facing emails from recruiters as well as scammers. These fraudulent job listings have the intent of gathering information from their applicants and possibly resulting in identity theft.

Using a variety of models, we aim to find any indicators of a fraudulent job posting from a Kaggle dataset through comparing models' accuracies and variables of importance.

Key Findings

Our job postings dataset consists of 95% non-fraudulent postings and 5% fraudulent postings. With such a positive case leaning dataset, we tried out 7 types of models to seek the highest accuracy.

From our stochastic gradient boosting and classification tree models, we found that two attributes of high importance include 1) the number of characters in a company's profile and 2) whether the job is in the oil energy industry or not.

The best performing model was Stochastic Gradient Boosting. The model that performed the worst was Naive Bayes, because of its oversimplified assumptions.

Future Work

Some future work we can continue to do is to incorporate hyper-parameter tuning. With our dataset being very unbalanced, we could introduce weights to the attributes that hold higher importance and try for a slower learning model.

We can further increase our accuracy by further mining into our text columns. Columns such as requirements and job description could provide lots of information to help determine if a job posting is fraudulent. We could look into sentiment analysis features to determine the tone of the job posting.

Project Plan

Methodology

- EDA of the dataset (11/1 - 11/15)
- Feature Engineer Text Columns using NLP Methods (11/15 - 11/30)
- Train models and validate on test data (12/1-12/3)
- Evaluate models based on accuracy while taking into account specificity and sensitivity (12/1-12/3)
- Accumulate relevant and important models and plots and write project report of insights, conclusions, and other important information. (12/3-12/4)

Tasks - Alisha

EDA

- Looked at telecommuting, title, salary_range, requirements, required_experience, function

Feature Engineering

- created binary variables for all predictors that had NA's, example: has_salary, etc
- added 20+ features for each text column regarding counts such as number of words, number of caps characters, number of first-person words, etc.
- Did further NLP by looking at unigrams and bigrams for title and grouped job titles into categories, created binary variables for each category
- Merged all group members EDA and feature engineering into one R file.

Modeling

- Naive Bayes, SVM, KNN

Tasks - Tiffany

EDA

- Looked at department, has_question, description, employment_type, industry

Feature Engineering

- created word count attributes for text attributes
- created dummy attributes for the six different employment types
- created attributes for whether a job posting is in top 25 industries and departments ranked by frequency

Modeling

- Classification Tree
- Random Forest
- Stochastic Gradient Boosting

Tasks - Anette

EDA

- Looked at has_company_logo, location, company_profile, benefits, and required_education

Feature Engineering

- Created word count for my text attributes
- Split location into country, state, city
- Created new column, region, from location
- Created counts for required_education/dummy variables

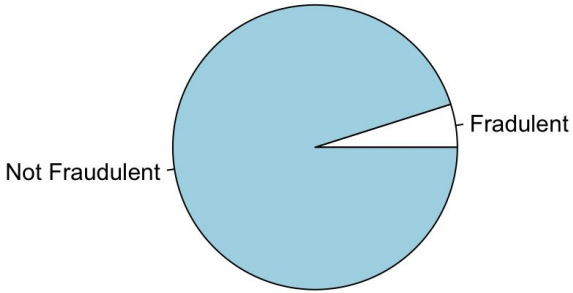
Modeling

- Logistic regression
- (Attempted: ridge regression, lasso regression)

EDA Findings

Fraudulent

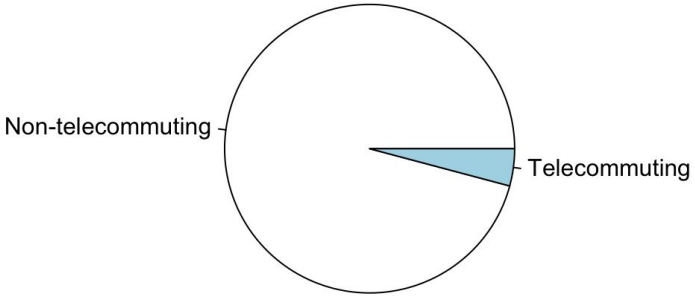
Our response variable. Whether a job posting is fraudulent or not. No NA's.



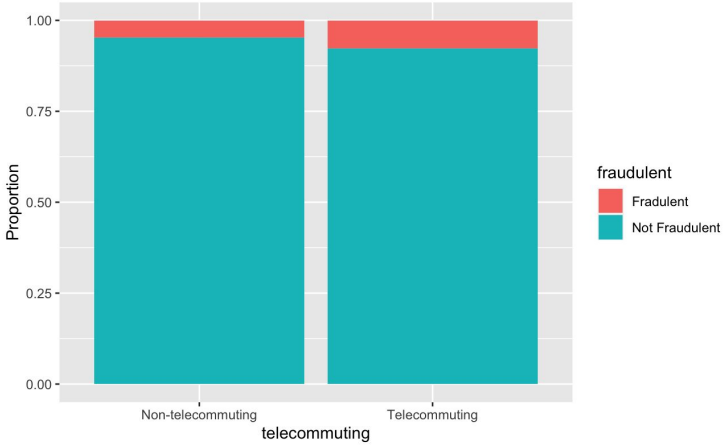
	Fraudulent	Not Fraudulent
Counts	700	13604
Fraction	0.049	0.951

Telecommuting

Binary. Whether a job is work-from-home or not. No NA's



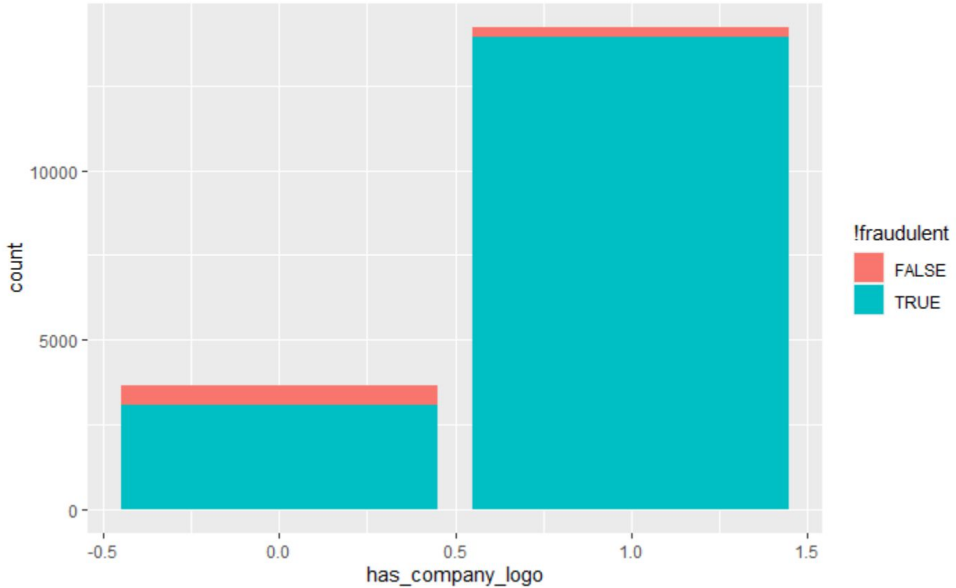
	Non-telecommuting	Telecommuting
Counts	13709	595
Fraction	0.958	0.042



telecommuting <chr>	fraudulent <chr>	n <int>	pct <dbl>	lbl <chr>
Non-telecommuting	Fradulent	654	0.04770589	5%
Non-telecommuting	Not Fraudulent	13055	0.95229411	95%
Telecommuting	Fradulent	46	0.07731092	8%
Telecommuting	Not Fraudulent	549	0.92268908	92%

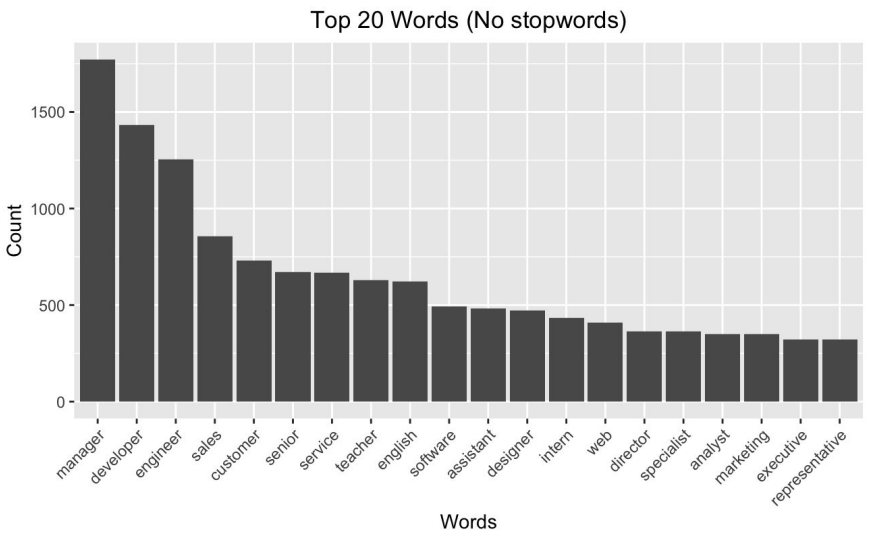
Has_company_logo

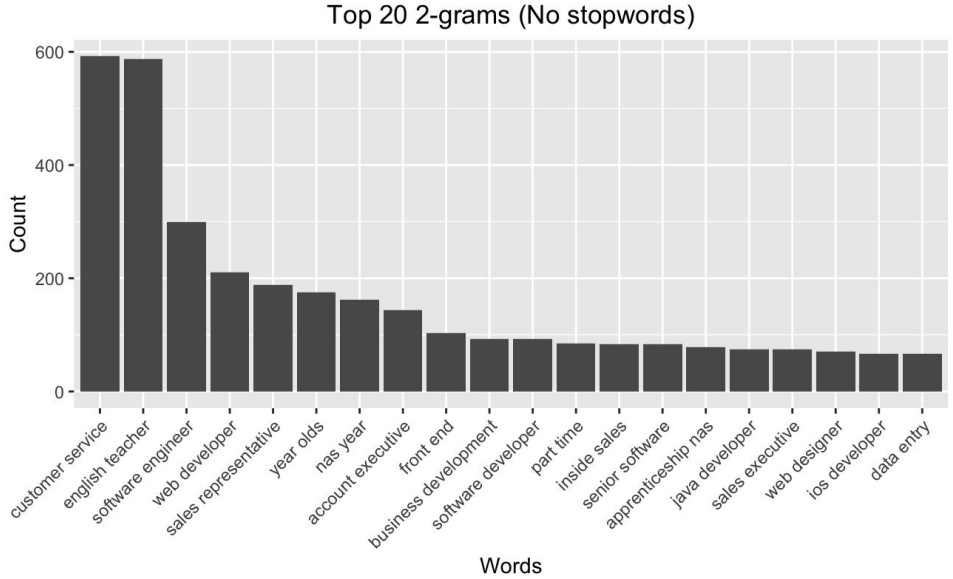
This feature is a binary variable with no NA's. It described if the job posting had the company logo or not.



Title

Text. Title of the job posting. No NA's. Created dummy variables based on unigrams and bigrams

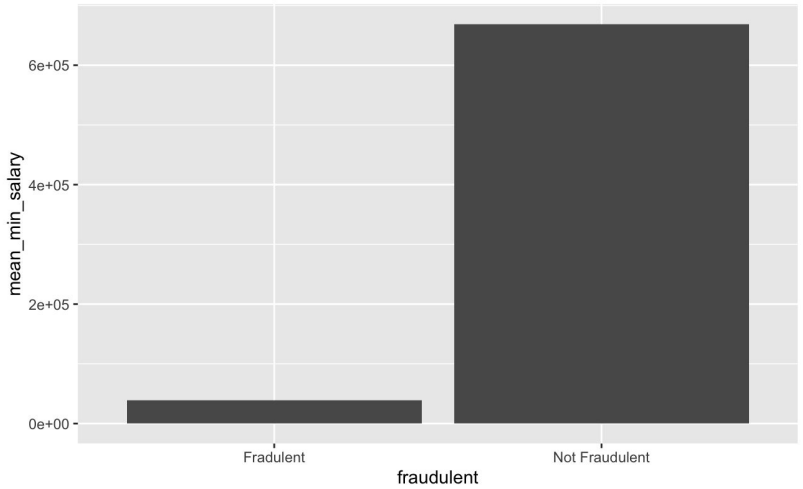




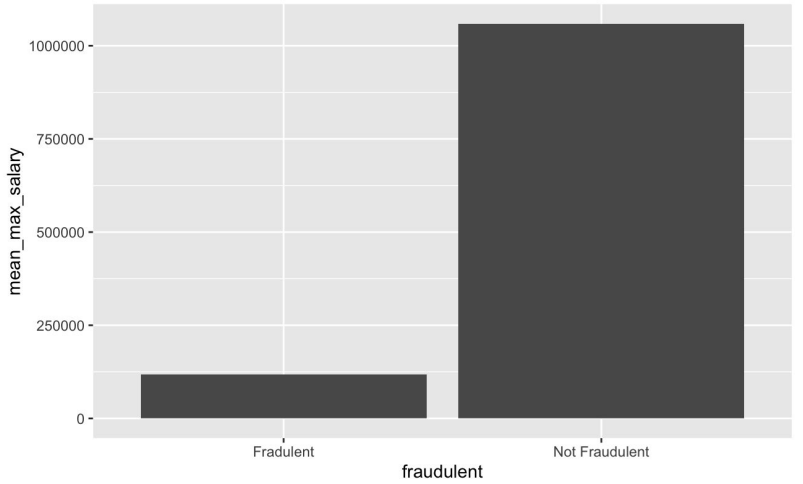
Salary

String. Contains the salary range for the position. Contains NA's. Extracted the min and max salary for each position from it. We can observe a large difference in the salaries for fraudulent and not fraudulent jobs.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0	18000	35000	621178	60000	800000000	12014

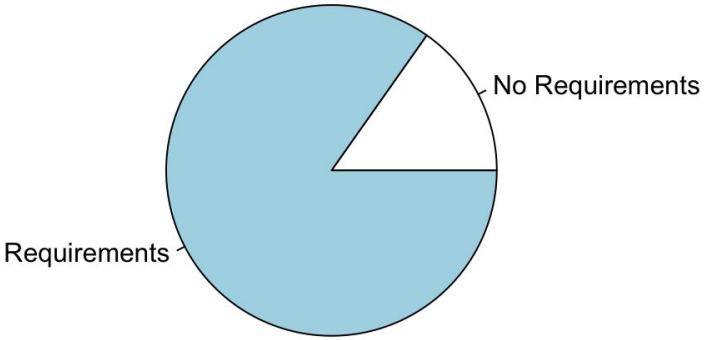


Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.000e+00	2.500e+04	5.000e+04	9.874e+05	9.000e+04	1.200e+09	12028



Requirements

Text. Contains the requirements from the job postings. Contains NA's. Created text features from it.



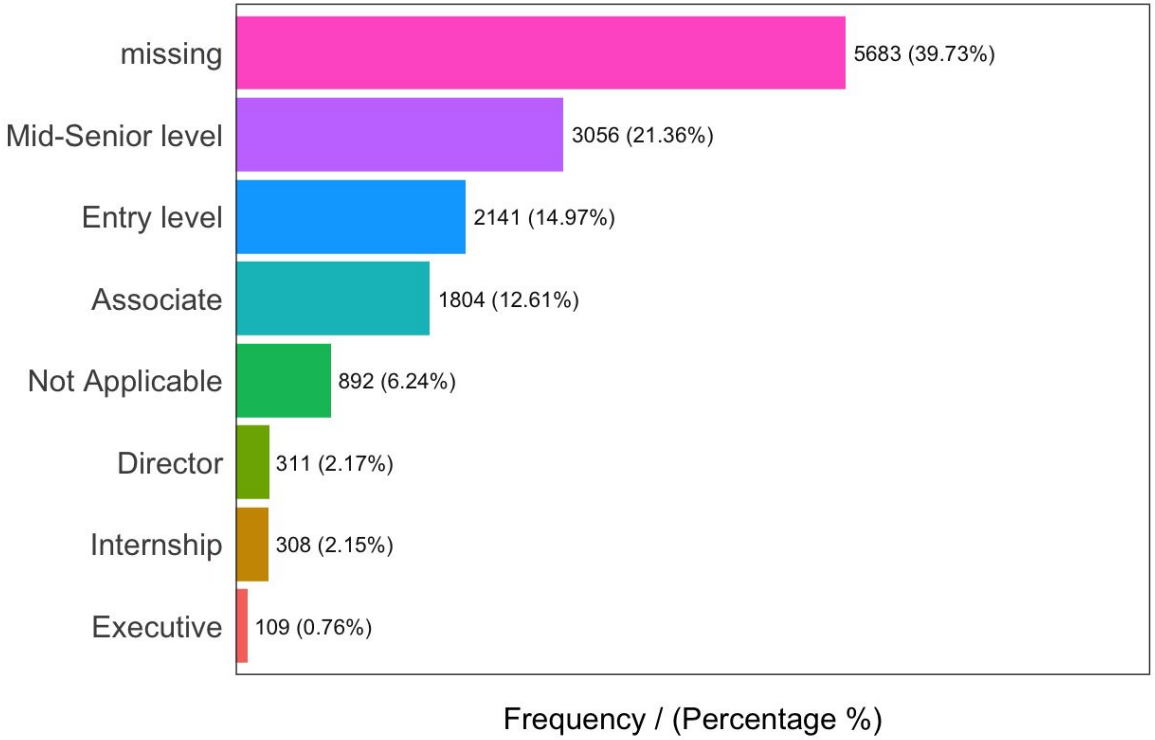
	No requirements	Requirements
Counts	2178	12126
Fraction	0.152	0.848

has_requirements <chr>	fraudulent <fctr>	n <int>	freq <dbl>
No Requirements	0	2058	0.94490358
No Requirements	1	120	0.05509642
Requirements	0	11546	0.95216889
Requirements	1	580	0.04783111

We can see that there is a small difference in fraud rate between postings that have requirements and postings that don't have requirements.

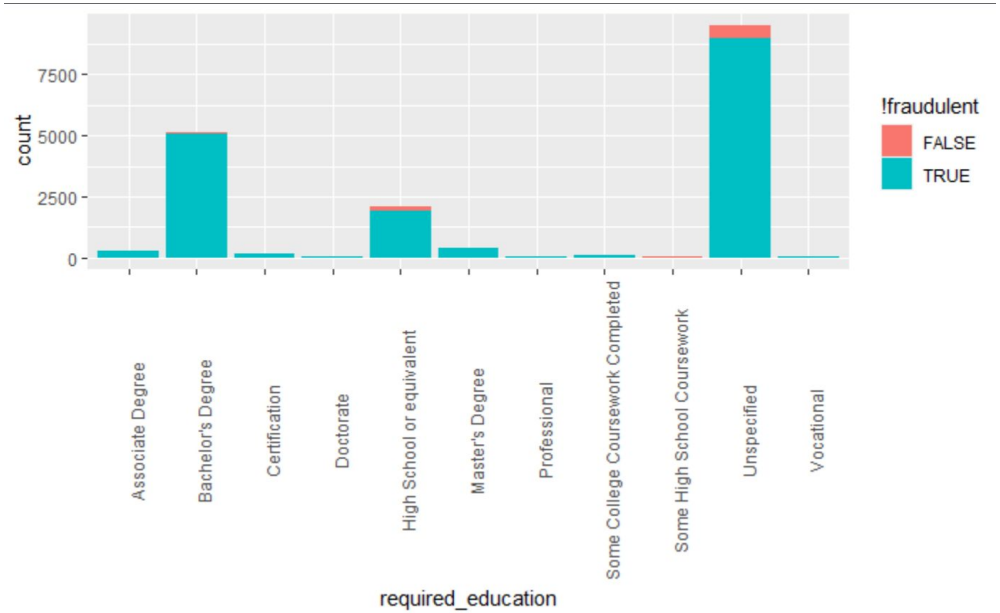
Required Experience

Categorical Variable. Contains the experience from the job postings. Contains NA's. Created dummy variables



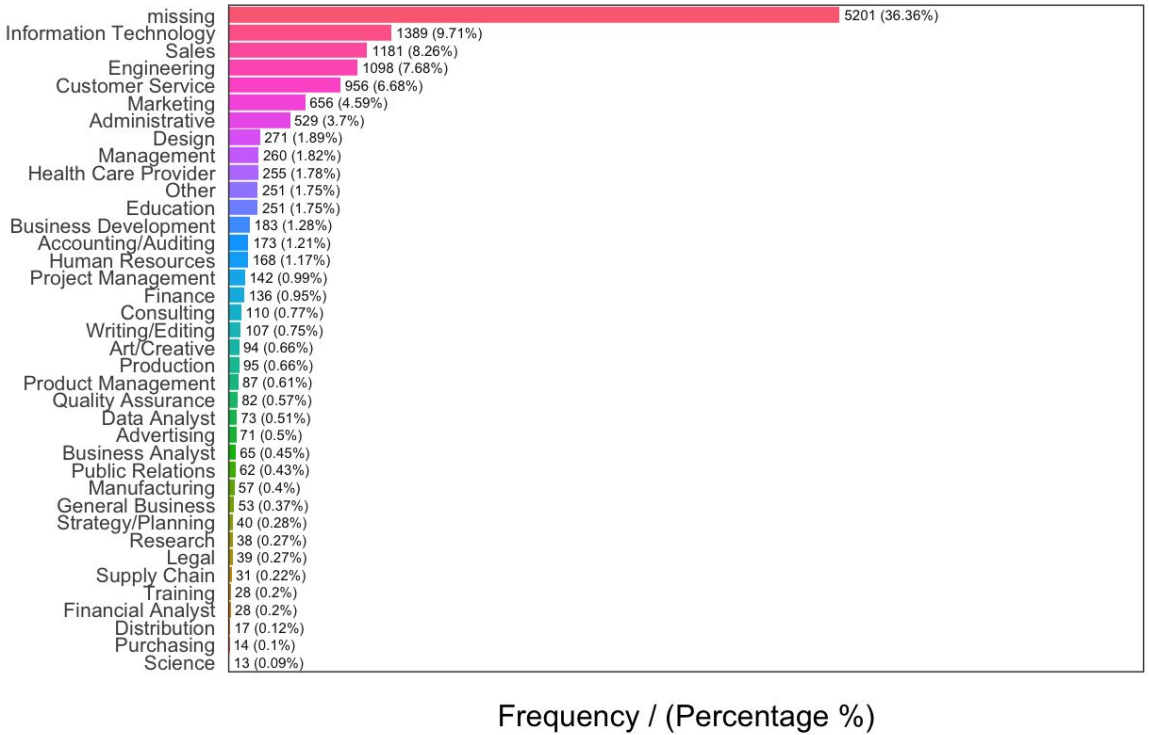
Required Education

This is a categorical feature which has 14 unique values. It had NA's and needed to be cleaned due to unnecessary separation between variables.



Function

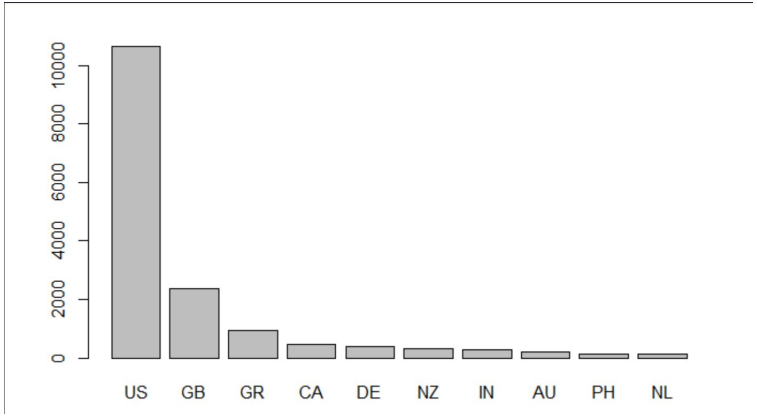
Text. The job functional area (science, consulting, administrative), etc. There were 38 distinct values. Contains NAs. Created dummy variables



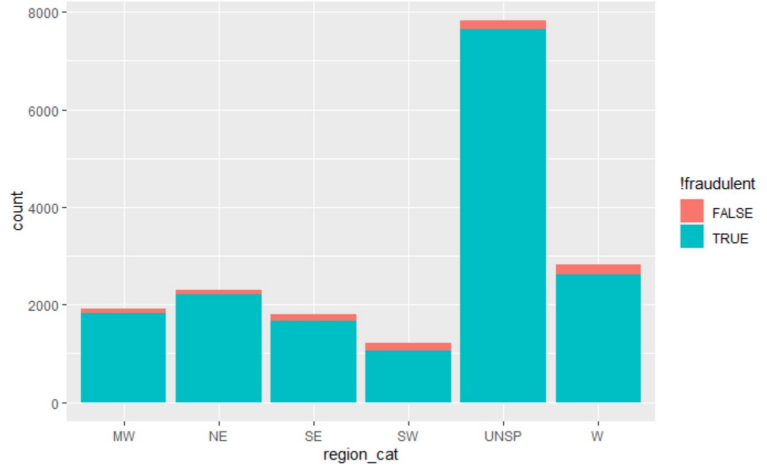
Location

Originally a text column which was split into four categorical ones. Split into country, state, city, and region. Each contained NA's and were labeled as unspecified. Country has 91 unique attributes, state has 325 unique attributes, and region has 6.

country	state	city	sum_fraud
US	TX	Houston	92
US	*NA*	*NA*	33
AU	NSW	Sydney	31
US	CA	Bakersfield	24
US	CA	Los Angeles	23
US	CA	San Mateo	22
US	NY	New York	20
NA	*NA*	*NA*	19
US	CA	San Jose	14
US	TX	AUSTIN	14

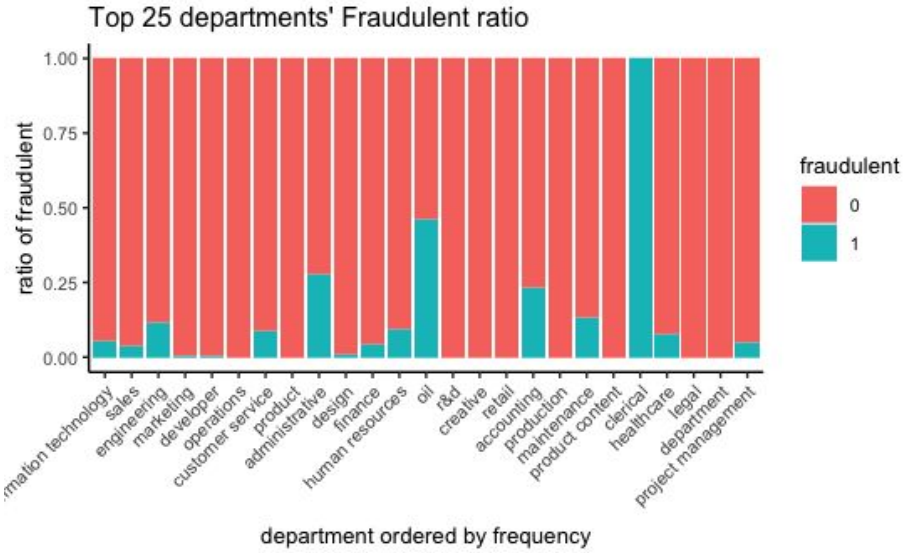


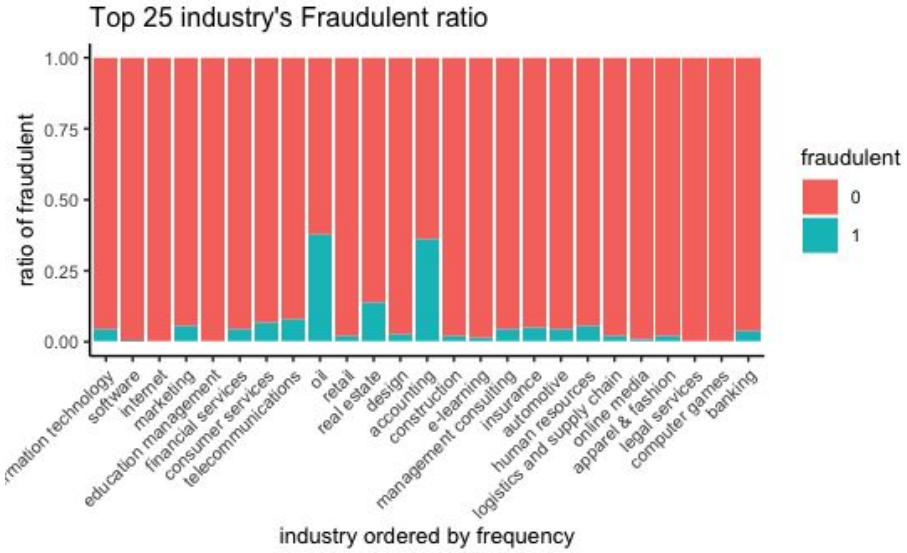
	Var1	Freq
1	CA	2051
2	NY	1259
3	LND	992
4	TX	975
5	I	688
6	IL	424
7	FL	415
8	OH	372
9	VA	332
10	MA	321



Department and Industry

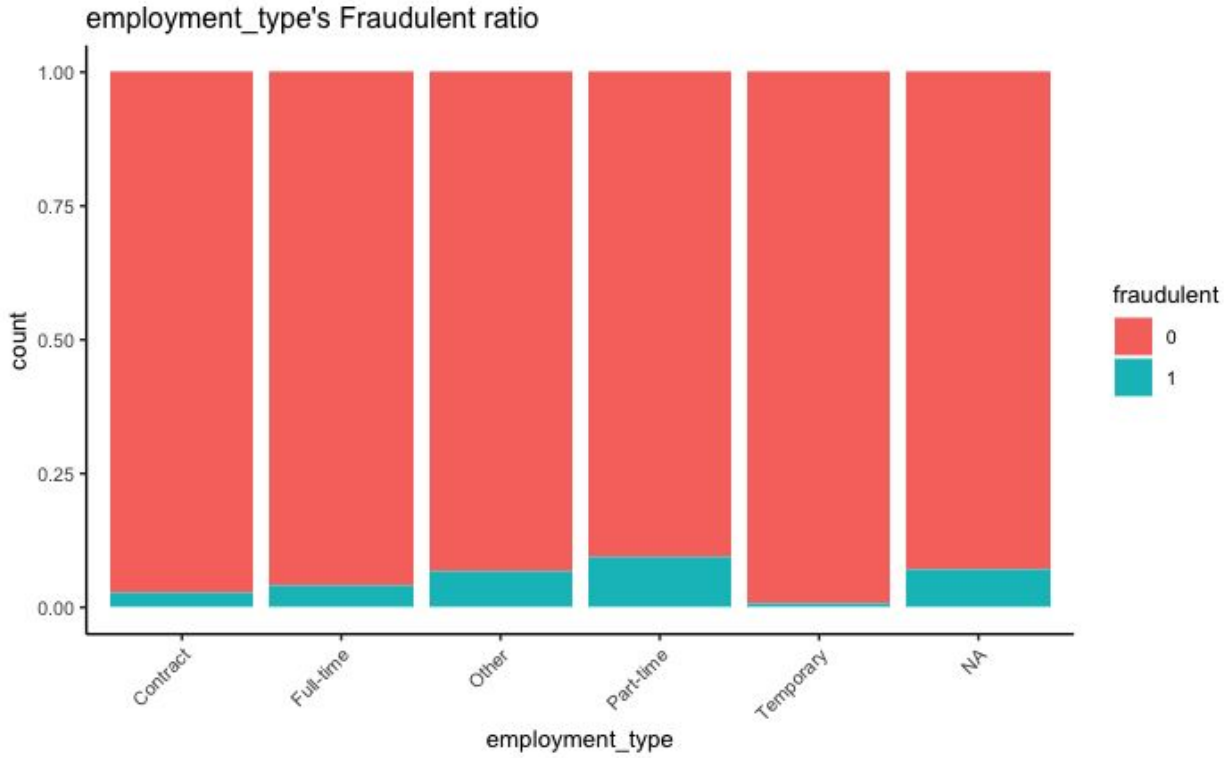
- An observation of importance is the two departments and industries that had the highest fraudulent to non-fraudulent ratio is Information Technology and Oil Energy.





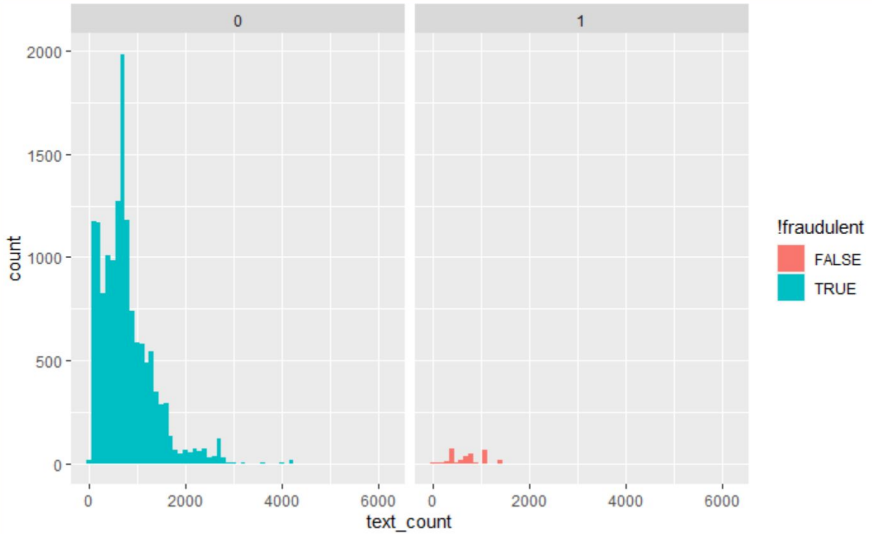
Employment_type

The highest fraudulent to non-fraudulent ratio are the listings with employment type of “part-time”.



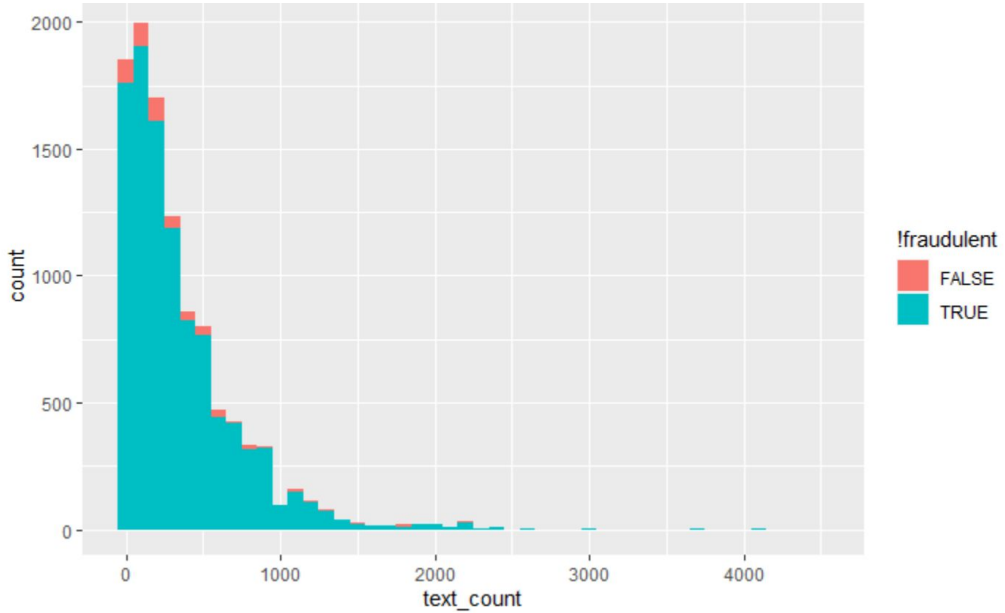
Company_profile

Company_profile is a text feature which did have NA's. First created a text_count based off of this. Then after creating a corpus and dtm it was tokenized to be used as multiple numerical columns.



Benefits

Benefits is a text feature, this described the benefits of employees. A text count of this was created as well as tokenization for numerical data.



Feature Engineering

Transformations

In our dataset, we have three types of attributes: binary, text, and categorical.

For the text attributes, we converted them into binary attributes to indicate its existence in the job posting. We also created columns such as the number of words, characters per word, etc for each of the text columns.

Split location into country, state, and city. Split salary into min/max salary.

Looked at most common values and made dummy variables for those values in the title, department, industry, employment_type, required_experience, and fn columns.

Model Processing

Models

In this project, we tried out 7 types of models:

1. Naive Bayes
2. Logistic Regression
3. Support Vector Machine
4. Classification Tree
5. Random Forest
6. Stochastic Gradient Boosting
7. k-Nearest Neighbors

Naive Bayes

Classifier that assumes independence between predictor variables.

	Observed Label (Fraudulent)	
Prediction Label (Fraudulent)	0	1
0	715	8
1	2695	158

Accuracy: 24.41%, Sensitivity: 99%, Specificity: 5.5%

Logistic Regression

Accuracy: 96.07%

Sensitivity: 99.37%

Null deviance: 5589.3 on 14303 degrees of freedom

Residual deviance: 3173.1 on 14260 degrees of freedom

AIC: 3261.1

Number of iterations: 8

```
Call:
glm(formula = fraudulent ~ ., family = binomial, data = df_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8420  -0.1919  -0.0800  -0.0268   3.8695

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -2.348e+00  2.411e-01  -9.736  < 2e-16 ***
has_company_logo            -9.867e-01  1.709e-01  -5.773 7.77e-09 ***
has_questions               -4.420e-01  1.231e-01  -3.590 0.000331 ***
has_department              -1.139e+00  4.608e-01  -2.471 0.013478 *
has_salary_range             3.326e-01  1.395e-01   2.384 0.017146 *
has_company_profile         -1.345e+00  2.972e-01  -4.525 6.05e-06 ***
has_requirements             2.275e+00  3.458e-01   6.579 4.73e-11 ***
has_benefits                 7.630e-01  2.644e-01   2.885 0.003910 **
```

Support Vector Machine

Used linear approach for SVM. Less vulnerable to overfitting compared to logistic regression.

	Observed Label (Fraudulent)	
Prediction Label (Fraudulent)	0	1
0	3389	110
1	21	56

Accuracy: 96.34%, Sensitivity: 96.86%, Specificity: 72.73%

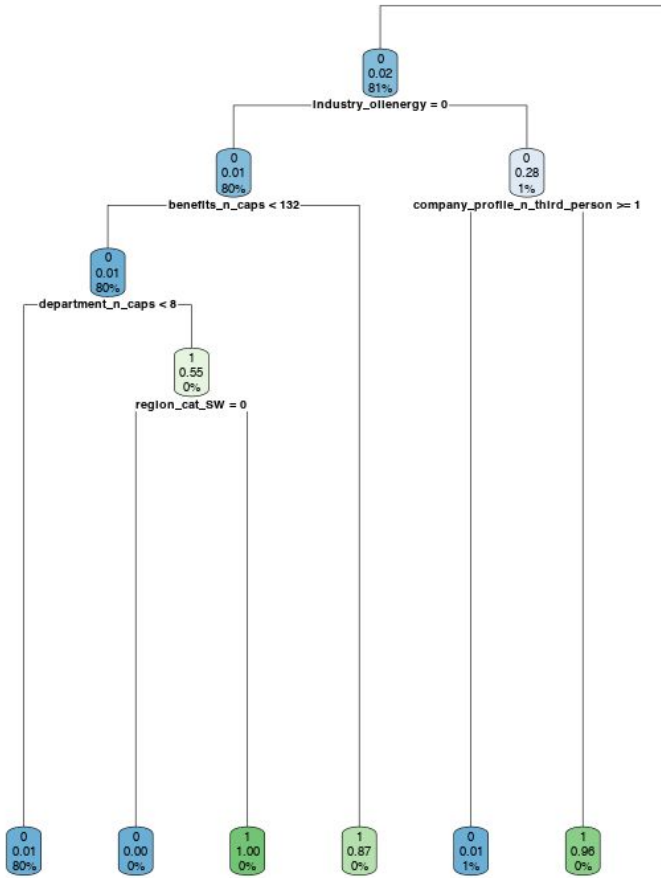
Classification Tree

Accuracy: 96.03%

Sensitivity: 99.56%

Variables of Importance:

- company_profile_n_chars
- industry_oilenergy
- requirements_n_chars



(zoomed into left branch of classification tree)

Random Forest

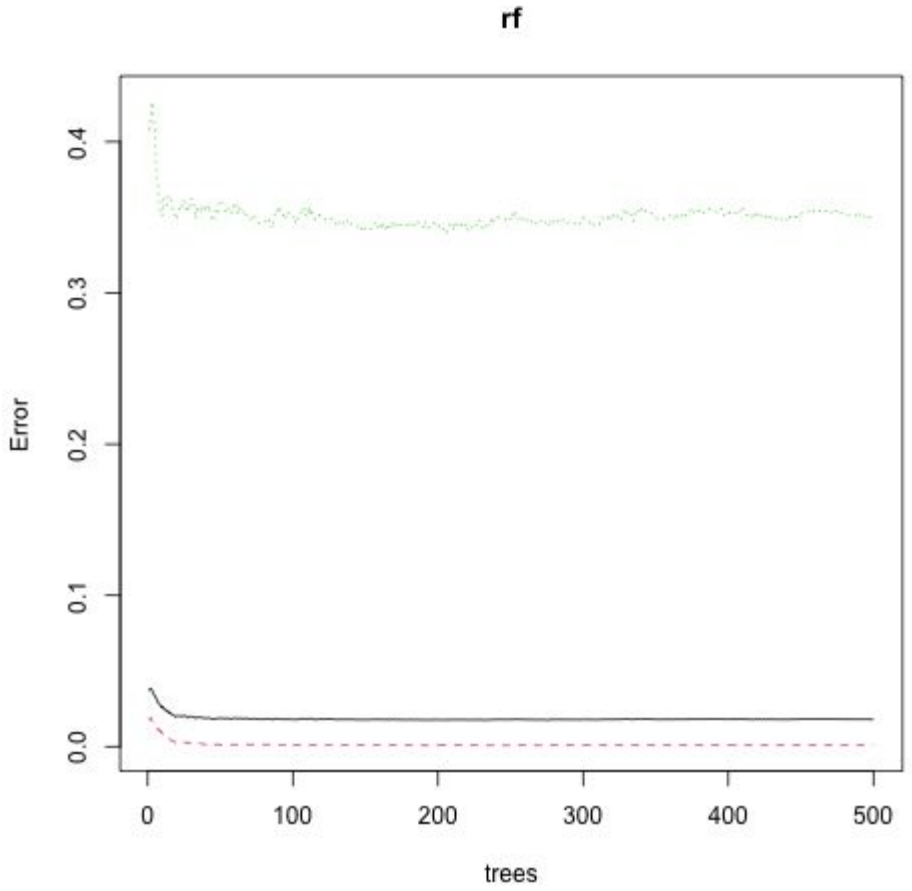
Accuracy: 97.99%

Sensitivity: 100%

Number of trees: 500

No. of variables tried at each split: 8

OOB estimate of error rate: 1.94%



Random Forest Classification error vs number of trees

Stochastic Gradient Boosting

Accuracy: 97.82%

Sensitivity: 100%

Variables of Importance:

- company_profile_n_chars
- cleandescription_length
- has_company_profile
- industry_oilenergy

	Observed Label (Fraudulent)	
Prediction Label (Fraudulent)	0	1
0	3396	50
1	14	116

Confusion Matrix	Percentage Rate
Accuracy	98.21%
Sensitivity (Recall)	99.59%
Specificity	69.88%

K- Nearest Neighbors

Looks at the K most similar observations in the train set and uses the mode of their response to predict the response

	Observed Label (Fraudulent)	
Prediction Label (Fraudulent)	0	1
0	3375	48
1	35	118

Accuracy: 97.68%, Sensitivity: 98.6%, Specificity: 77.12%

Performance Results

Classification Results

- Accuracy

Stochastic Gradient Boosting has 98.21%, meaning it correctly classified the most fraudulent cases.

- Sensitivity

Random Forest had 100%, meaning that all fraudulent job postings were identified as fraudulent.

- Specificity

KNN got a value of 77.12%, but there's a range from 5.5% to 77.12%.