Fake Job Detection

Authors: Congkai Sun, Jie Kuai, Maojie Xia

Summary

With the development of online business, many companies are taking advantage of it including their hiring process. However, the new employment situation has also brought many employment scams. Some scammers post fake jobs on websites to gain money from job seekers. In order to analyze the differences between real jobs and fake jobs, the goal of this project is to select features that have a strong relationship with job prediction and apply these features to appropriate classification models in order to predict the fake job accurately in the future.

This project uses data provided by Kaggle. This data contains features that define a job posting. These job postings are categorized as either real or fake. This project follows four stages -

1. Data Collection
2. Data cleaning, exploring, and data pre-processing
3. Modeling
4. Evaluating

In this project, the models are evaluated by accuracy and F1_score. The models include logistic regression, random forest, Naive Bayes, KNN, and XGBoost. Finally, XGBoost and random forest have a good performance on fake job prediction.

Methods

1. Data Sample

The dataset contains 17880 job descriptions out of which about 800 are fakes and 18 features. The data is a combination of integer, binary, and textual data types.

2. Data Processing

- Missing value processing: First, determine the proportion of missing values for each variable. If the proportion of missing values exceeds 50%, it is considered that there is a problem with the variable, and the variable can be directly eliminated. The ‘department’ and ‘salary_range’ contain 11547 and 15012 missing values which are about 80% of the total. Thus, we drop these two columns. When the proportion of missing values of a variable does not exceed 50%, it is necessary to analyze the distribution of the variable. If the variable follows a normal distribution, it can be replaced by the mean value. If the variable does not follow the normal distribution, fill it directly with ‘Not Applicable’. Meanwhile, the values with ‘Unspecified’ are replaced by ‘Not Applicable’. The table of the sum of the null values is shown in the appendix.

- Eliminate invalid jobs: According to the filtered special text of location, find out the countries where the fake jobs are concentrated. From the table, more than 90% of the fake jobs are in the United States, so choose the jobs posted in the United States as the sample. The table of locations of fake jobs is shown in the appendix.

- Feature selection: For the numeric data, use the correlation matrix to visualize the relationship between each feature and fraudulent and select the feature with a correlation greater than positive 0.1 or less than negative 0.1. Considering the correlation and distribution of each numeric feature, the ‘has_company_logo’ can be selected as the feature.

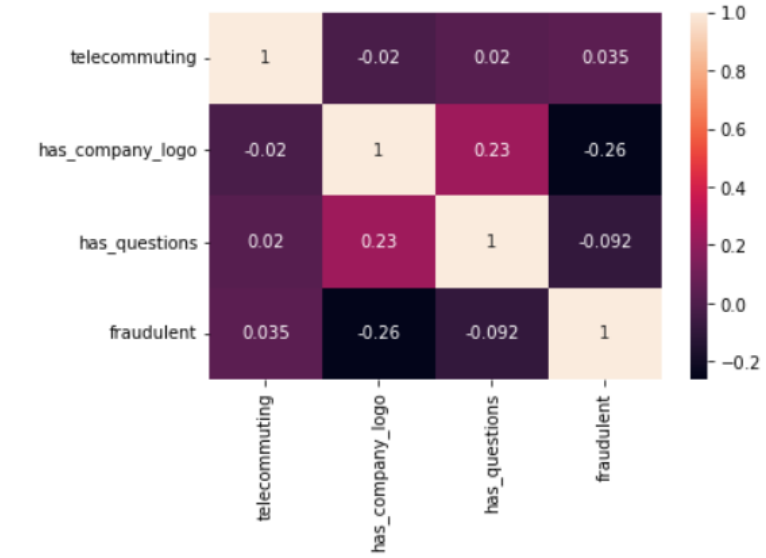

Figure 1: Correlation matrix for numeric features

The location is split into state and city and combined into a new column. To explore fraudulent distributions, two figures are made: distribution in each state and distribution in each city of the state. According to the distribution in these two figures, the state distribution of fraud is more concentrated which is mainly located in California, New York, and Texas. Therefore, the state can be selected as the feature for the model.

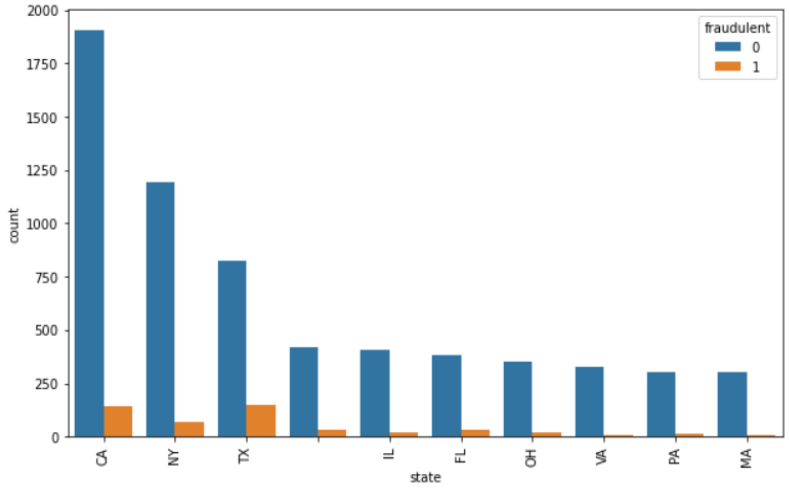

Figure 2: Distribution of fraudulent and real jobs in each state

For text features, 'required_education', 'required_experience', 'employment_type', 'function' and 'industry' these features’ categories are short. Therefore, with fake probability and distribution analysis, select the appropriate features.

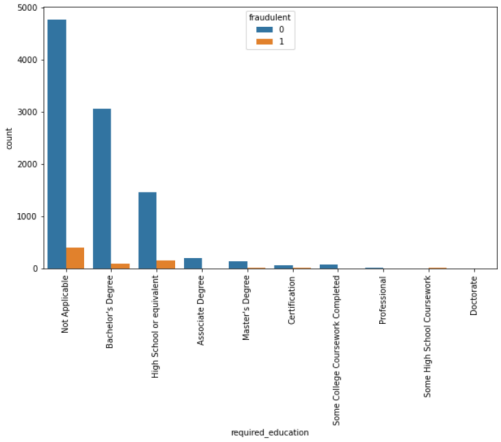

Figure 3: Distribution of fraudulent and real jobs in each degree

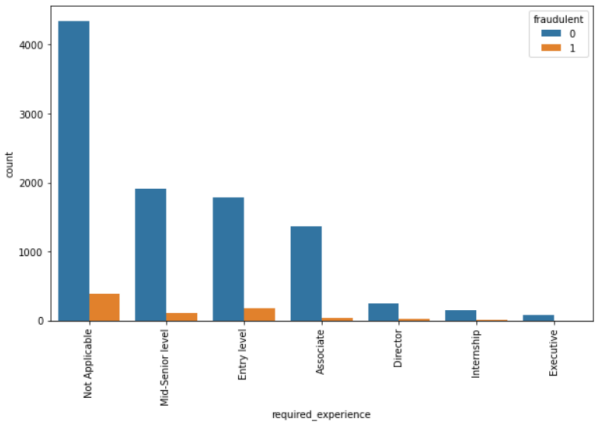

Figure 4: Distribution of fraudulent and real jobs in each experience

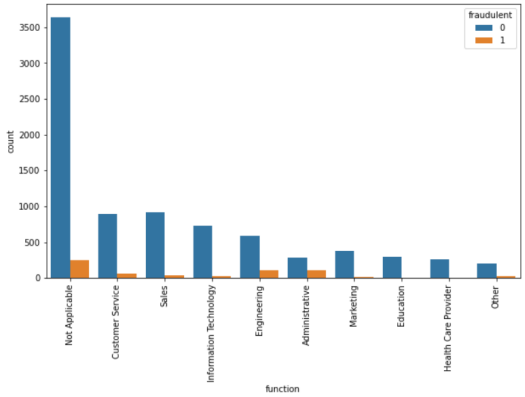

Figure 5: Distribution of fraudulent and real jobs in each function

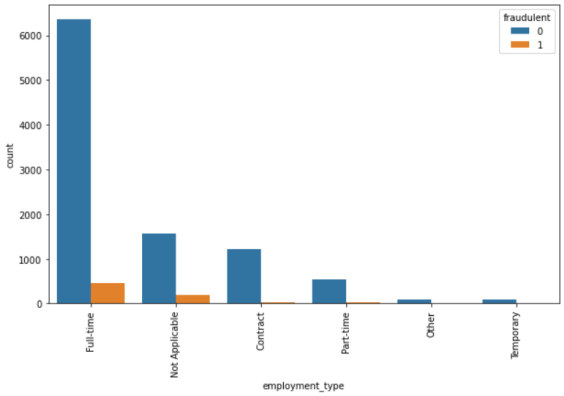

Figure 6: Distribution of fraudulent and real jobs in each employment type

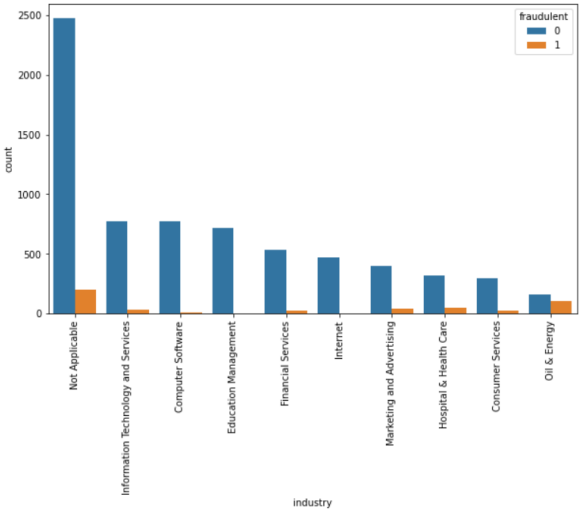

Figure 7: Distribution of fraudulent and real jobs in each industry

The remaining text-based features are combined into one field called text to extend the analysis on text-related fields further. These features have a lot of text descriptions, so use the stopwords package to remove common words in the text to extract key information. The combined fields are — title, company_profile, description, requirements, benefits. A histogram describing a character count is explored to visualize the difference between real and fake jobs. It can be seen that in

different word count intervals, the frequency of words in the real job is higher than that in the fake job. Thus, the labels for each word's count intervals can be created.

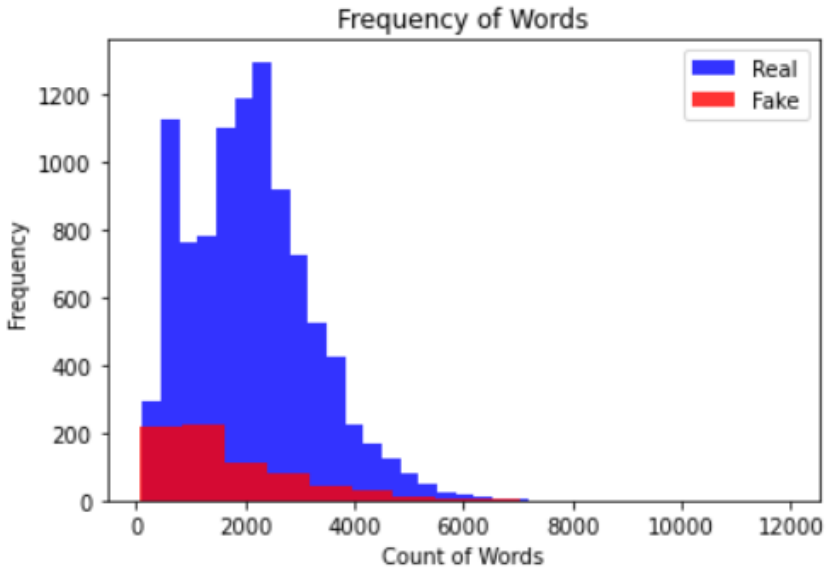

Figure 8: The frequency of words count range

- Feature encoding: Use Labelencode to encode the features, including 'employment_type', 'required_experience', 'require_education', 'industry', function, 'state', and 'words_mapped'.

3. Model Implementation

- Extract 67% of the data for model training and 33% of the data as the test set.
- Algorithms and techniques:
    1. Logistic Regression: The model can be used for data that fits into named categories, in this project, most selected features have named categories.
    2. Random Forest: The forest chooses the classification with the majority of the 'votes'. The forest picks the average of the output of all trees. In this project, there is a low correlation between each individual feature.
    3. Naive Bayes: This is the baseline model. It is used because it can compute two events' conditional probabilities based on the probabilities of occurrence of each event; encoding those probabilities is extremely useful.
    4. KNN: Make the plot of accuracy with different k values and choose the n_neighbors with the best performance. Here we use n=2.
    5. XGBoost: This model is suitable for data with a large number of observations and many categorical variables.

4. Model Evaluation

The models are evaluated based on accuracy and F1- score. The accuracy produces a ratio of all correctly categorized data points to all data points. This metric helps us to evaluate the accuracy of models which identify both real and fake jobs. However, since our data are highly unbalanced, both false negatives and false positives are important. Therefore, an F1- score is useful in our evaluation.

Results

1. Model Input

Through the characteristic analysis of data, 8 features are selected as the input of the model which are shown below.

Column	Explanation	Type
has_company_logo	Has company logo or not	int64
employment_type	The type of employment	int64
required_experience	Required experience for the job	int 64
required_education	Required educational degree for the job	int 64
industry	Field of the job	int64
function	Specific role for the job	int64
state	Location of the job	int64
words_mapped	Count of words range of job description	int64

Table 1: The features of the model

2. Model Output

The class of the job as the output of the model

Column	Explanation	Type
fraudulent	Fake job or not	int64

Table 2: The target of the model

3. Model Evaluation

From the table below, all models have good accuracy in the prediction, this may be due to the highly unbalanced dataset. So, F1- score is an important factor to judge whether the fake jobs are correctly predicted or not. Logistics regression and Naive Bayes have low F1-score which means most fake jobs fail to be predicted. Based on the accuracy and F1-score, XGBoost and random forest have good performance. Thus, these two models can be chosen for this project.

Model	Accuracy	F1-Score
Logistic Regression	0.940	0.117
Random Forest	0.964	0.663
Naive Bayes	0.940	0.192
K-Nearest Neighbors	0.956	0.563
XGBoost	0.965	0.670

Table 3: The evaluation of the models

A confusion matrix can be used to evaluate the quality of the project. The project aims to identify real and fake jobs.

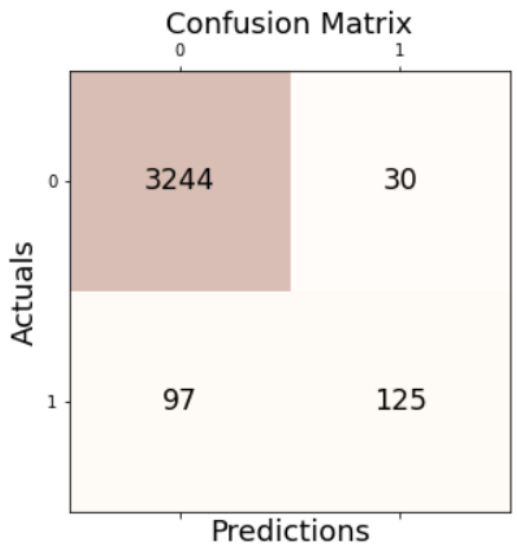

Figure 9: The confusion matrix of random forest

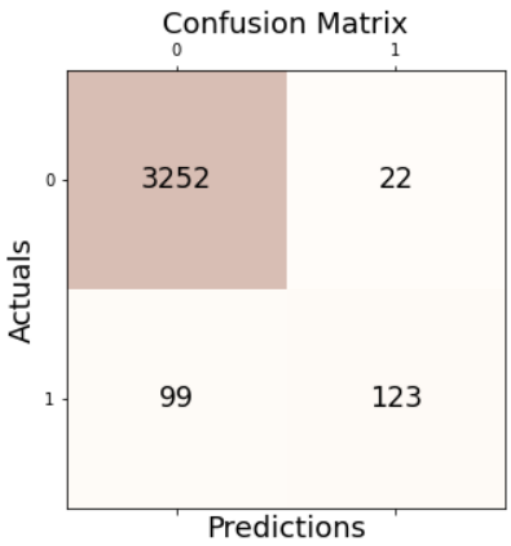

Figure 10: The confusion matrix of XGBoost

The test data has 3265 real jobs and 231 fake jobs. Based on the confusion matrix, the random forest can identify 99.36% real jobs and XGBoost identifies 99.60% real jobs. However, fraudulent jobs are identified only 54.11% by using random forest and 53.25% by using XGBoost. Only 3.63% of the time has the random forest not identified the jobs correctly and XGBoost fails to identify 3.46% jobs.

Discussion

It can be seen from the results, the selected feature and the type of classification model have a great influence on the prediction results. This project provides a potential solution to the fake job posting problem. The textual data is pre-processed to generate optimal format and relevant numerical data are chosen for the model. And comparing multiple models' performance to produce the best possible results. The feature selection is important for the model's performance. For example, location is a crucial feature of this project. California, Texas, and New York have more than 90% fake jobs. Places like this require some extra monitoring. Also, most fake jobs need a bachelor's degree or high school diploma for full-time employment. For the model implementation, the tree model like random forest and XGBoost have good predictions on fake jobs. Thus, these models are suitable for this kind of classification problem.

Currently, we live in an unexpected time. Coronavirus not only brings health threats but also accelerates impending recession. Especially in tech companies, many people are being laid off every day. Meanwhile, available job posts in the labor market are much less than people's demand for jobs. Under this phenomenon, anxiety and chaos are utilized by scammers. However, this project provides a possible solution for hiring websites to filter out fake jobs. Avoiding fake job postings will dramatically improve job-seeking efficiency.

Improvement

The dataset in this project is highly unbalanced. About 93.3% are real jobs and few are fake jobs. Therefore, it is difficult to find the consistency for the features of the fake job, and the probability of

predicting the fake job accurately will be reduced. Therefore, a balanced dataset should produce better results.

Besides, the tree models which are implemented in this project do not set any parameters. So, for better prediction in the future, some parameters such as iteration, deep, and randomly selected sample proportion can be tested for different situations.

Contributions

Maojie Xia: Coding (Data Exploring), Report (Summary)
Jie Kuai: Coding (Data Cleaning, Data Exploring), Report(Method, Discussion)
Congkai Sun: Coding(Data Collection, Data Cleaning, Data Exploring, Data Pre-Processing, Modeling, Evaluating), Report (Summary, Methods, Results, Discussion, Improvement)

References

[1] Dataset of Real / Fake Job Posting Prediction,
https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction

Appendix

Table 4: Sum of the null values

job_id	0
title	0
location	346
department	11547
salary_range	15012
company_profile	3308
description	1
requirements	2695
benefits	7210
telecommuting	0
has_company_logo	0
has_questions	0
employment_type	3471
required_experience	7050
required_education	8105
industry	4903
function	6455
fraudulent	0

Table 5: Location of fake jobs

Location	Count	Probability
US	725	0.837182
CA	155	0.178984
TX	152	0.175520
Houston	92	0.106236
NY	68	0.078522
San	57	0.065820
AU	40	0.046189
MD	35	0.040416
NSW	32	0.036952
Sydney	31	0.035797
FL	30	0.034642
Bakersfield	24	0.027714
Mateo	24	0.027714
Los	23	0.026559
Angeles	23	0.026559
New	23	0.026559
York	22	0.025404
GB	21	0.024249
GA	20	0.023095

Figure 11: Heatmap of null values:

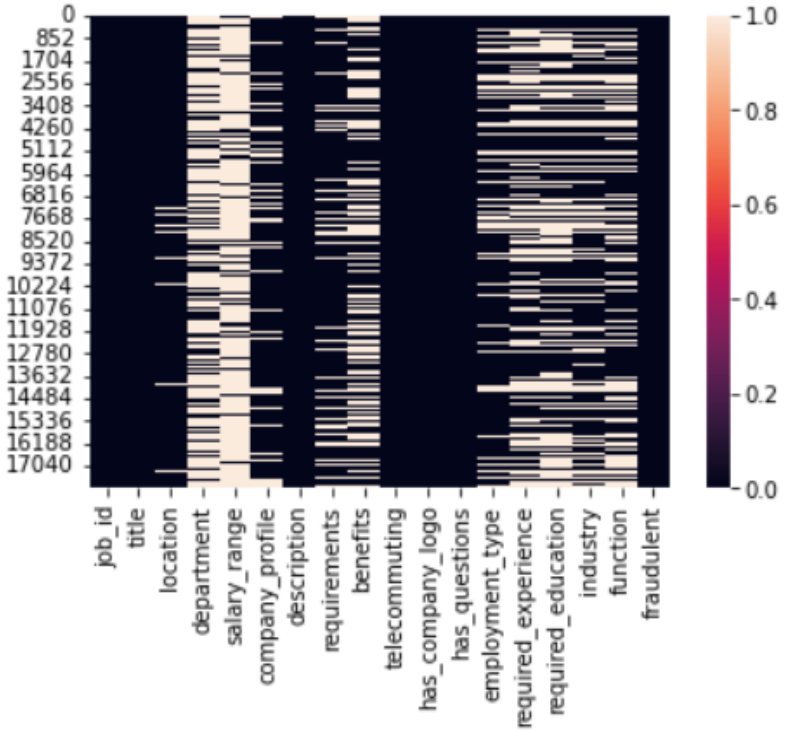

Figure 12: Proportion of fake jobs:

```
Real Job:  17014
Fake Job:  866
```

Figure 13: Distribution of telecommuting:

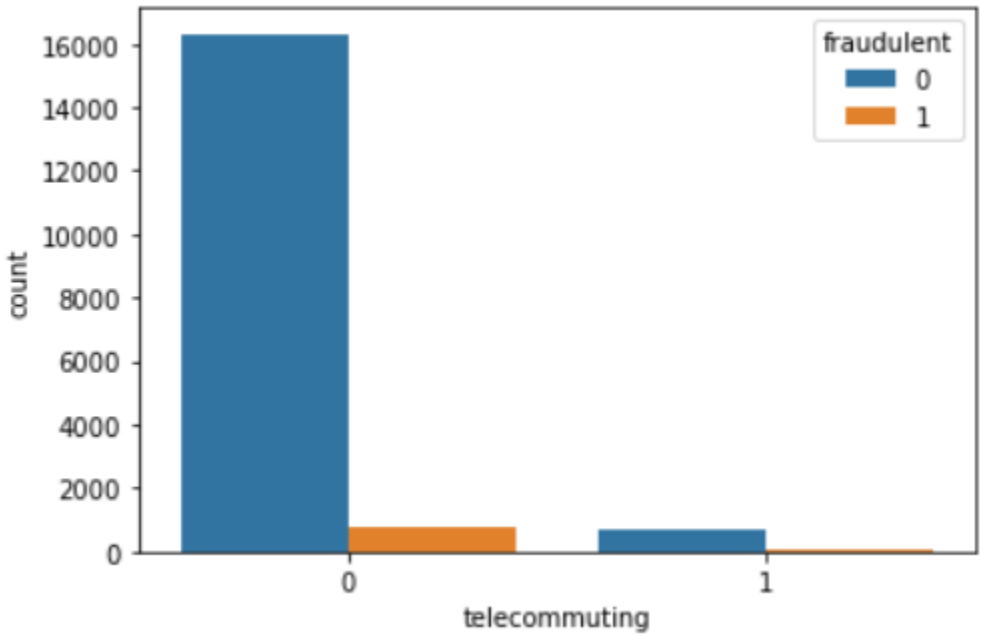

Figure 14: Distribution of has_company_logo:

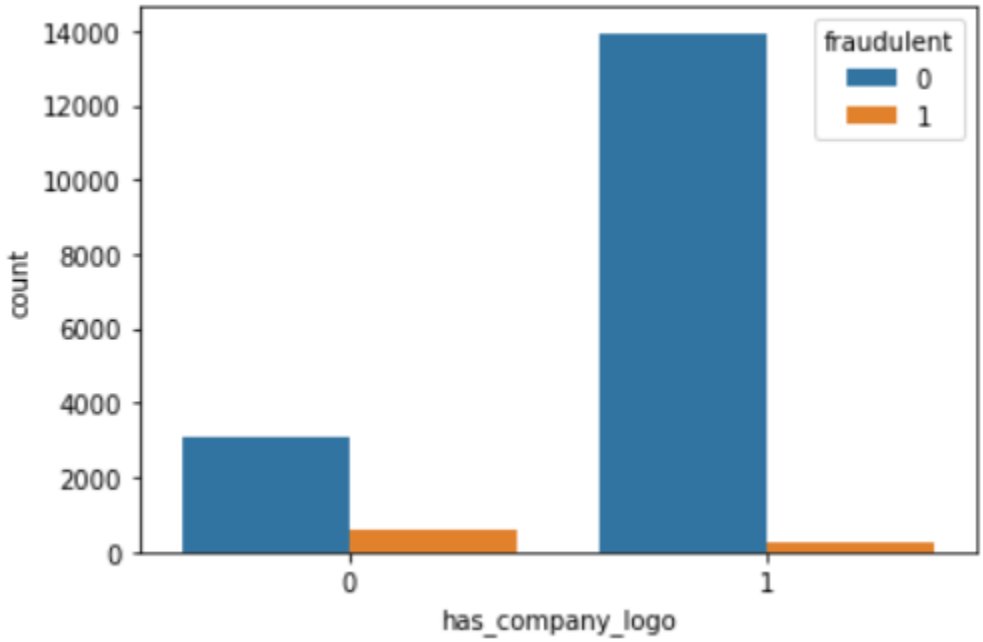

Figure 15: Distribution of has_questions:

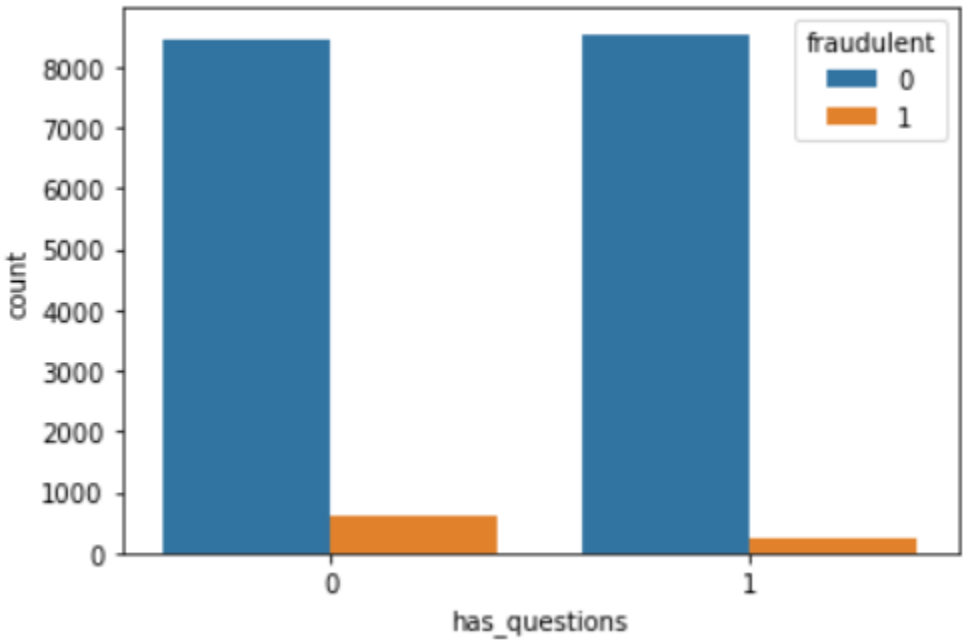

Figure 16: Distribution of jobs in each city:

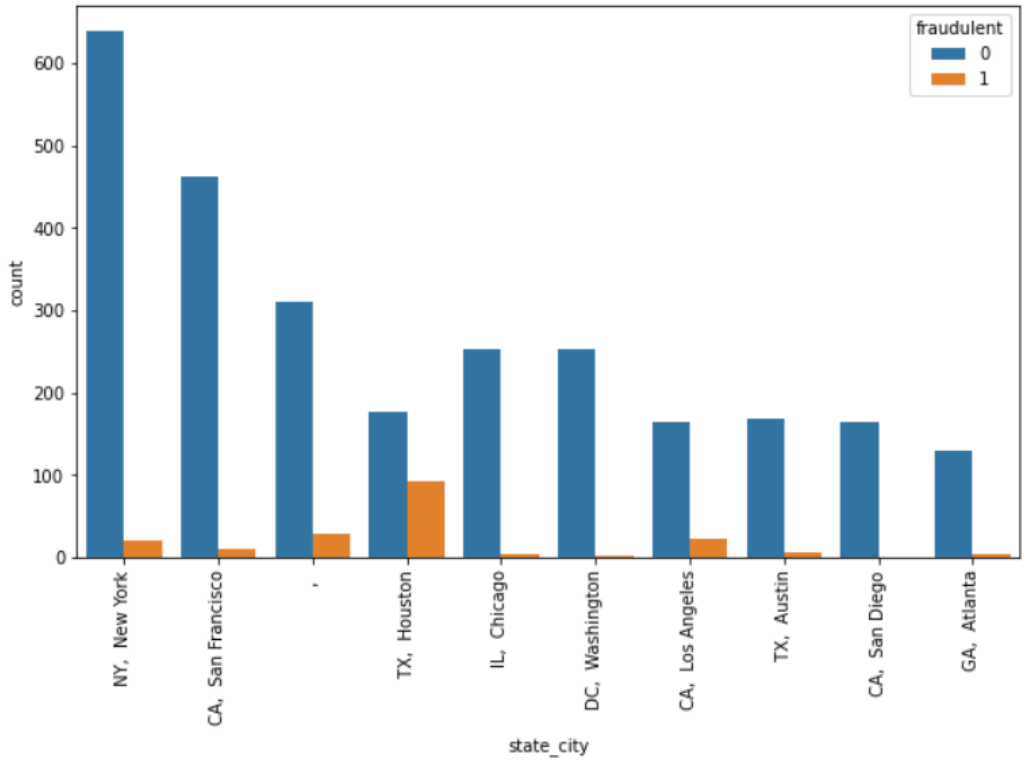

Figure 17: Confusion matrix of logistic regression:

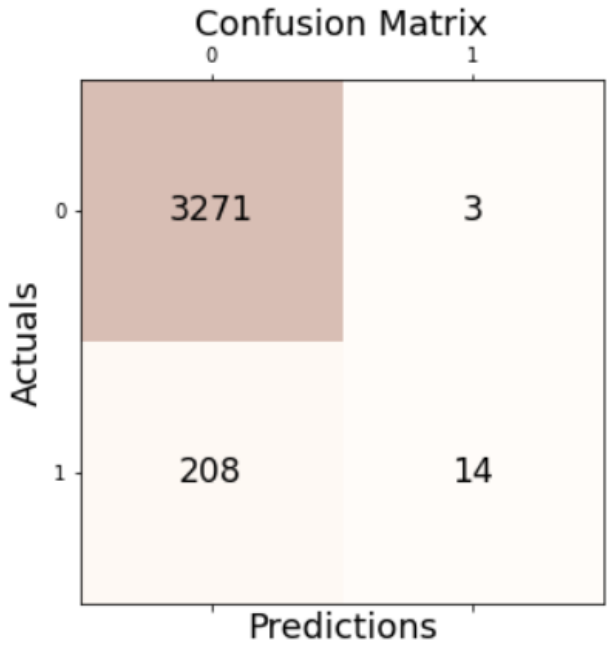

Figure 18: Confusion matrix of Naive Bayes:

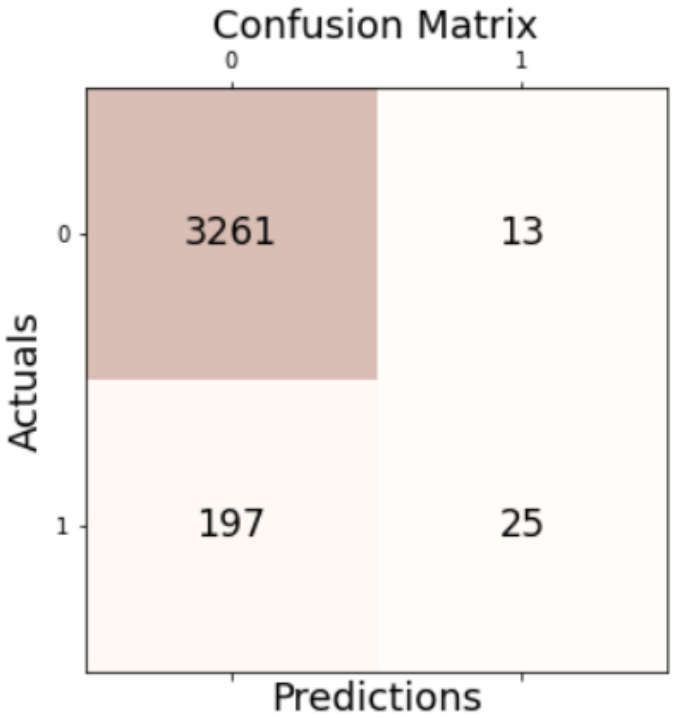

Figure 19: Confusion matrix of KNN: