Fake Job Predictor

Anshupriya Srivastava

Project Overview

This project is a part of the final submission for Udacity's Nanodegree in ML Engineering. The folder data contains all relevant data used for the project and the folder code contains EDA and modelling files. The five stages adopted for this project are –

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

Problem Definition

Employment scams are on the rise. According to CNBC, the number of employment scams doubled in 2018 as compared to 2017. The current market situation has led to high unemployment. Economic stress and the impact of the coronavirus have significantly reduced job availability and the loss of jobs for many individuals. A case like this presents an appropriate opportunity for scammers. Many people are falling prey to these scammers using the desperation that is caused by an unprecedented incident. Most scammer do this to get personal information from the person they are scamming. Personal information can contain address, bank account details, social security number etc. I am a university student, and I have received several such scam emails. The scammers provide users with a very lucrative job opportunity and later ask for money in return. Or they require investment from the job seeker with the promise of a job. This is a dangerous problem that can be addressed through Machine Learning techniques and Natural Language Processing (NLP).

Data Collection

This project uses data from Kaggle to determine if a job posting is real or fake. The link for this dataset is - https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction.

Data Exploration

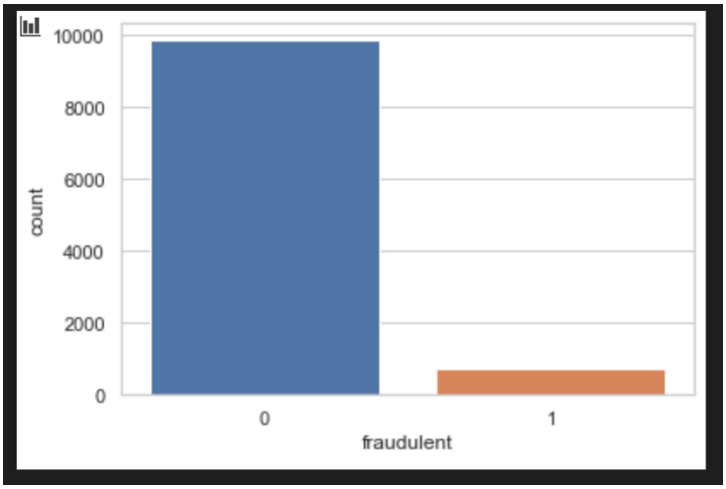
I started the data exploration with the entire dataset that consisted of 17880 observations and 18 features. These are features are -

#	Column	Non-Null Count	Dtype
0	job_id	17880 non-null	int64
1	title	17880 non-null	object
2	location	17534 non-null	object
3	department	6333 non-null	object
4	salary_range	2868 non-null	object
5	company_profile	14572 non-null	object
6	description	17879 non-null	object
7	requirements	15185 non-null	object
8	benefits	10670 non-null	object
9	telecommuting	17880 non-null	int64
10	has_company_logo	17880 non-null	int64
11	has_questions	17880 non-null	int64
12	employment_type	14409 non-null	object
13	required_experience	10830 non-null	object
14	required_education	9775 non-null	object
15	industry	12977 non-null	object
16	function	11425 non-null	object
17	fraudulent	17880 non-null	int64

I decided to use only data from US based locations that account for nearly 60% of the dataset. I did so to ensure that I have data in English. Also, the location is split into state and city for further analysis. The final dataset has 10593 observations and 20 features.

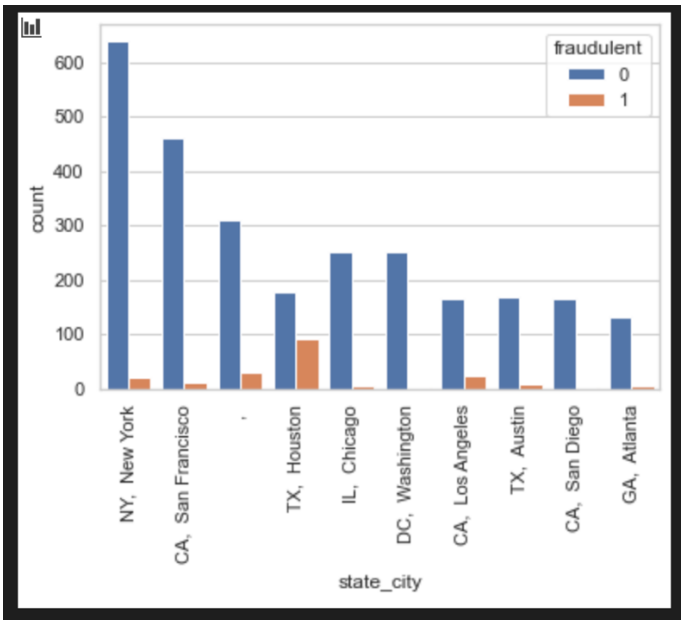
After determining the final dataset, I moved on to exploring the dataset further based on other features.

Real vs fake job distribution:
Distribution of fake and real jobs



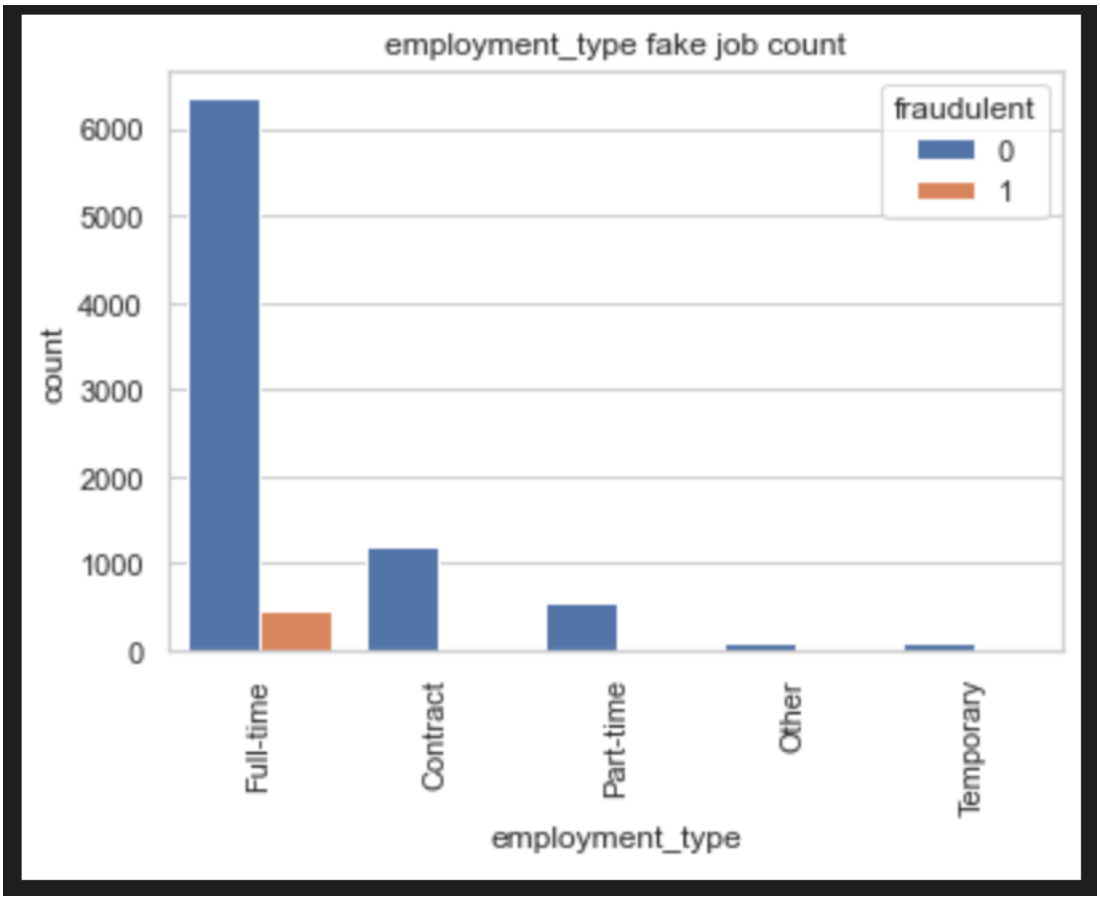
It is visible from the plot that the fake jobs are a very small fraction of the real ones.

Location based real vs fake job distribution:
Distribution of fake and real jobs based on location



From the plot it can be seen that Houston, TX has the very high fake to real job ratio.

Employment_type based real vs fake job distribution:
Distribution of fake and real jobs based on employment type



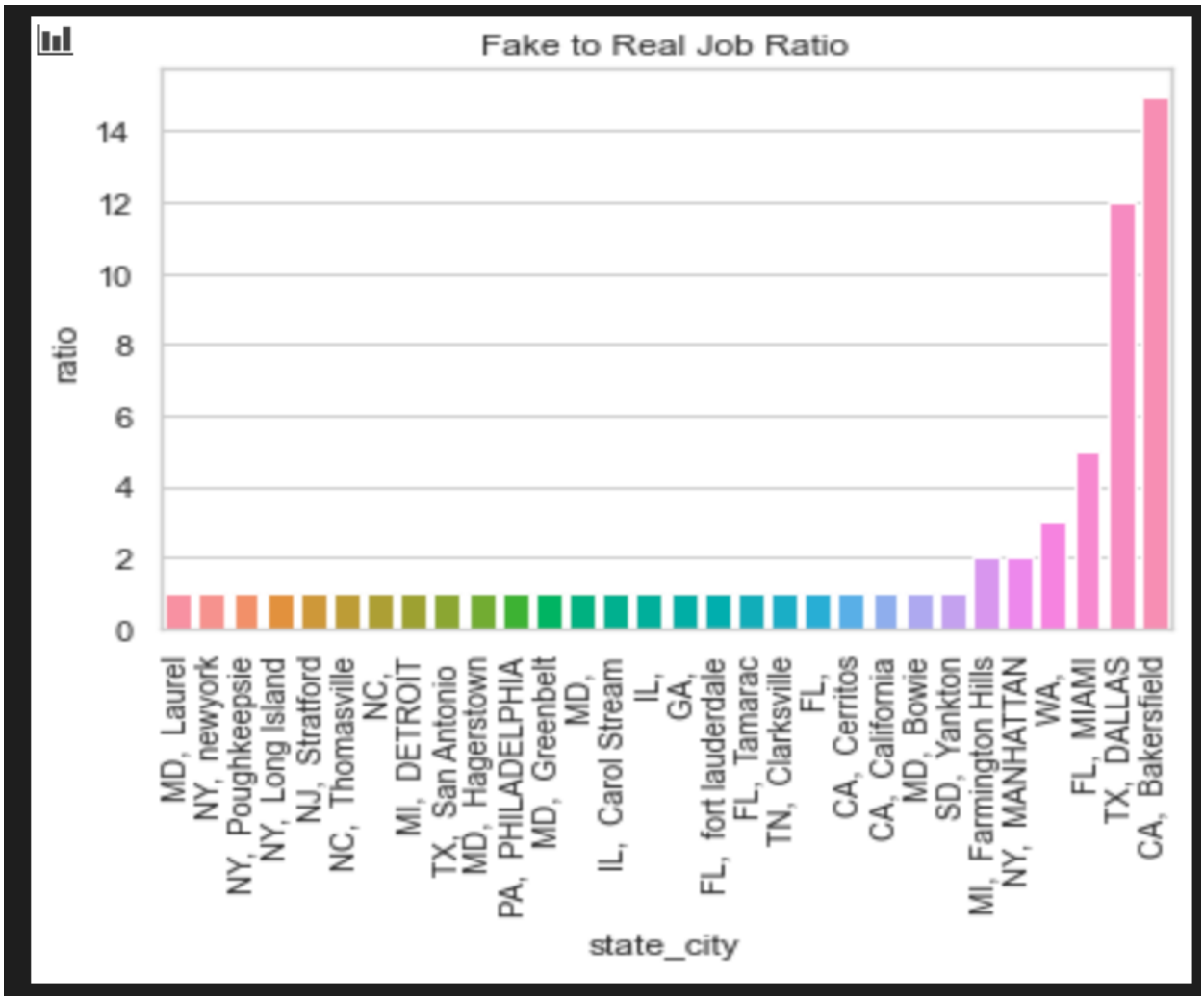
Most fraudulent offers arise in a situation when the type to job offered is full time.

Other analysis:

Most entry level jobs have fraudulent jobs
Jobs requiring only a bachelor's degree or high school education have a higher chance of being fraudulent.
To enhance the results of this analysis, I created a fraud to real ratio for different locations. Based on the ratio a countplot is designed to view places with very high ratios.

Fake to real ratio

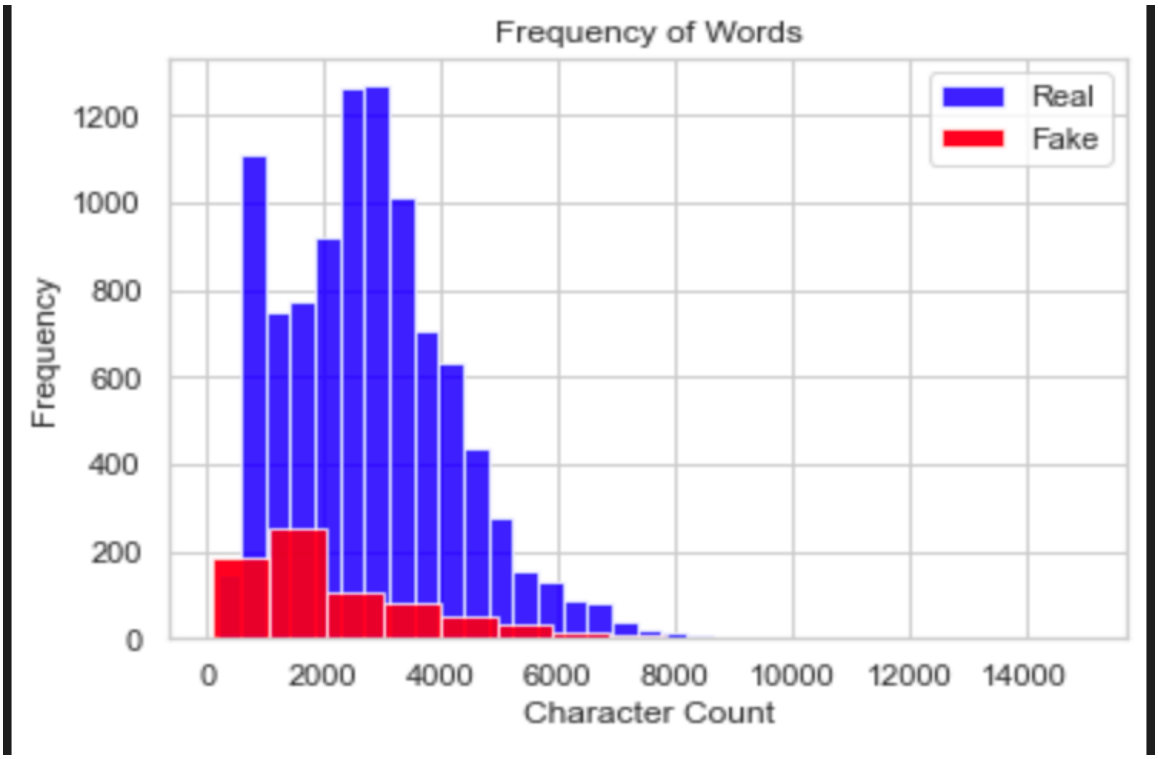
Some places like Bakersfield, CA and Dallas, Texas have a very high fake to really job ratio. Location seems to be a very important factor in evaluating fake jobs.



Two other features - telecommuting and has_company_logo had some kind of relationship with fake jobs as well. 58.6% of fraudulent jobs have both these feature values equal to 0.

After performing EDA and determined important features a new dataframe is created. This dataset has the following feilds-

- telecommuting
- has_company_logo
- has_questions
- fraudulent
- ratio: fake to real job ratio based on location
- text: combination of title, location, company_profile, description, requirements, benefits, required_experience, required_education, industry and function
- Word count: Count of words in the textual data Word count histogram



This final dataset is used further for text preprocessing before using a final model.

Data Preprocessing

Using python's natural language processing package - nltk the next steps are performed. The text data is tokenized, stopwords are removed and the text is lemmatized. A new clean dataset is used for the final step of this project - modeling.

Modeling

The new and clean dataset is used for creating the final models. The. X and y variables are separated.

X: telecommuting, has_company_logo, has_questions, ratio, text, character_count

y: fraudulent

This dataset is split into test-train datasets using sci-kit learn's train_test_split method. The test set is one-third of the entire dataset. Using CountVectorizer from sci-kit learn the text data from X is converted to a count matrix which will be used in the baseline model - Naive Bayes. The accuracy of this classifier is 0.971.

Another model - SGDClassifier is used and it produces an accuracy of 0.974. Based on accuracy scores, SGDClassifier is chosen. Another SDGClassifier is implemented on the numeric part of

the dataset. The accuracy of the test counterpart of this is 0.934. The final results of both these models are combined and a final output is generated. Both models need to say that a particular job is real in order to give the final output as real. If either or both say that the job is fraudulent, the job is put in the fraudulent category.

Evaluating

The final model is evaluated using F1 score. This has been chosen as the final metric because it is the right balance between recall and precision. We need this model to identify fake and real jobs, both effectively.
The f1 score for the final model is 0.79. The f1-score for the baseline model is 0.743.

Conclusion

This model is quite effective in degerming if a job is real. However, due to a small fraction of data belonging to the fraudulent class, the ML algorithm is in general favorable to the dominant class. Further work requires using techniques like SMOTE to create a class balance.