5741 Project Midterm Report: H-1B Petition Analysis

Yueran Yang (yy595), Zongyuan Yuan (zy225)

1 Problem Statement

The Labor Condition Application (LCA) is an application for the employer to hire a foreign worker in a specific position for no more than 3 years. H-1B Lottery is followed by approval of LCA. This project is aimed at predicting the case status of an LCA application based on the relevant information of a foreign employee in a company such as wage, job type, etc.

2 Dataset

2.1 About the Dataset

The raw data contains 2,448,729 rows and 36 columns. All features and counts of NA entry for each feature is shown in Table 2 in Appendix 5.1.

SOC_CODE refers to Standard Occupation Classification, which is to classify workers into categories.[1] In this project, we use 2010 SOC System[2]. NAIC refers to North American Industry Classification. NAIC_CODE is used to classify the business of a company [3].

CASE_STATUS is the label in our project. It contains four status, CERTIFIED, WITHDRAWN, DENIED and CERTIFIED-WITHDRAWN. Thus, the goal of the project is to solve a multi-class classification problem.

2.2 Data Processing

After deleting all redundant features shown in Table 2. We first remove all the rows that contain NA. We do not consider fill in the values because the number of rows containing NA is quite small as shown in the table and it is hard to use any method to fill in those critical information. Similarly, if there are invalid values in the following cleaning process, we simply delete the samples.

For NAIC_CODE, we only keep 4 digits. The length of the code can be 2 to 6. The longer the code, the more detailed the classification. If we keep the original data, there will be 531 different values in this categorical feature, which can be too detailed for training and may cause over-fitting. Thus, we only consider to keep 3 digits with 190 different values.

For SOC_CODE, similar to NAIC_CODE, we only keep 3 digits, which decrease the number of different values from 1238 to 100.

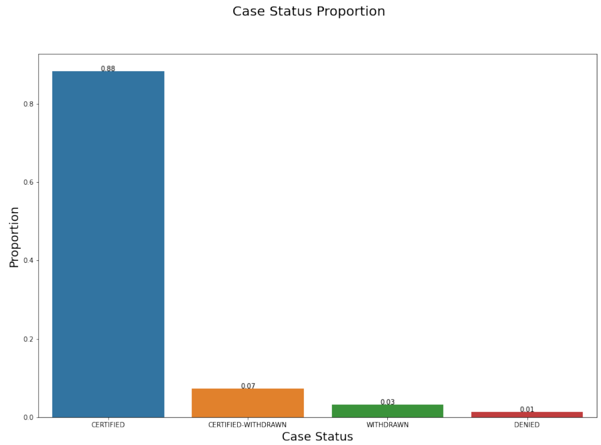
For Date features (EMPLOYER_START_DATE, EMPLOYER_END_DATE, CASE_SUBMITTED and DECISION_DATE), we use the duration as the new features and delete the original four features. For wages, we use the units of pay and values of wage to calculate the yearly wage as our feature.

For EMPLOYER_NAME, we count the frequency of a company in the data set and estimate the possibility of the four status for a specific company as the prior possibility.

For other categorical feature, we use one-hot vectors. The final data contains 422 columns and 2435102 rows.

2.3 EDA

We first look at the distribution of CASE_STATUS. From the bar plot below, we can see that most of the applications are certified.



Therefore, the labels of our data set is very unbalanced.

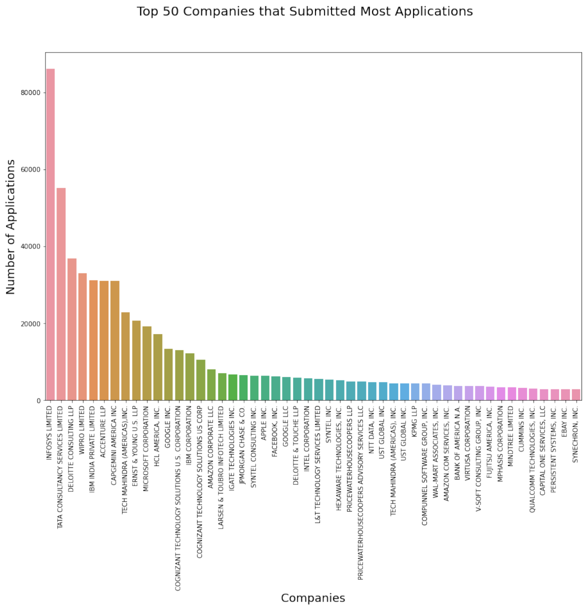
As we looked at the dataset, we found that there are 169,810 companies that submitted applications.

[1] See https://www.bls.gov/soc/

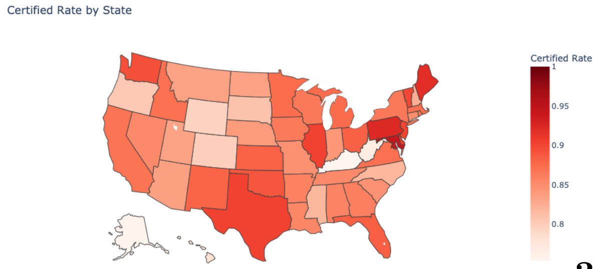
[2] See https://www.bls.gov/soc/2010/2010_major_groups.htm

[3] See https://www.census.gov/naics/

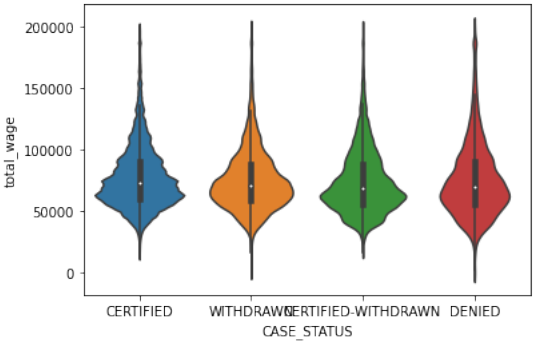
If we were to use one-hot encoding for the feature EMPLOYER_NAME, the dimension of our dataset would be too large to handle. Therefore, we want to keep only a subset of companies that that contributed a large proportion of total applications. The following histogram shows the top 50 companies that submitted most applications.



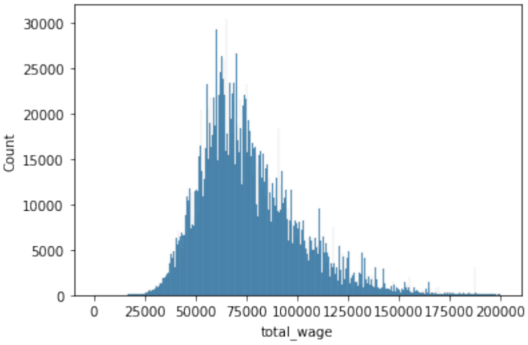
Another important factor is the location of the employer. From the plot below, we can see that the applications from employers in some states are more likely to be certified than others.



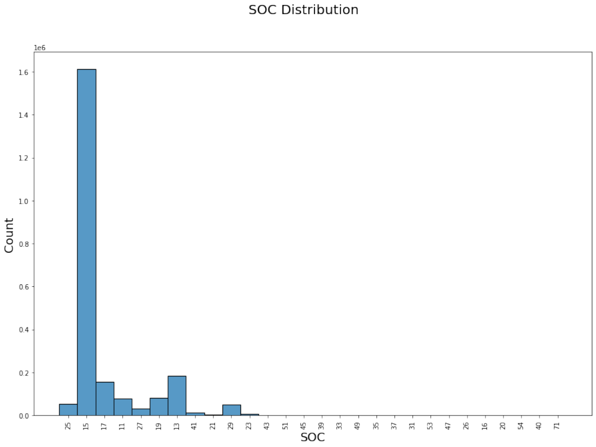
Wage is an important indicator of the value of the job. Empirically, the higher the wage, the more likely an application might be certified. The violin plot below shows that average wages in certified applications are slightly higher than average wages in applications with other status.



From the wage distribution below, we can see that most applications have wages from 50,000 to 100,000.



The majority of the applications have SOC code that starts with 15 meaning that most of the applications are computer or software related.



3 Preliminary Models

We developed three preliminary models: 1. Decision tree with two levels 2. Logistic Regression 3. Linear Discriminant Analysis (LDA)

3.1 Performance of the Three Models

Since the labels of the dataset is very unbalanced, using accuracy as our metric is not reflective of the

Table 1: F1-Score for Each Label

Model	**TEST F1-Score**			
	CERTIFIED	**CERTIFIED-WITHDRAWN**	**WITHDRWN**	**DENIED**
Decision Tree	0.9900	0.9343	0.7452	0.0000
Logistic Regression	0.9708	0.8611	0.1903	0.4034
LDA	0.9559	0.6993	0.2735	0.4635
Model	**TRAINING F1-Score**			
	CERTIFIED	**CERTIFIED-WITHDRAWN**	**WITHDRWN**	**DENIED**
Decision Tree	0.9898	0.9327	0.7412	0.0000
Logistic Regression	0.9707	0.8570	0.1897	0.4065
LDA	0.9560	0.6941	0.2757	0.4611

model performance. Therefore, we use F1 score instead. F1 is defined as follows:

$$\frac{2 \times (precision \times recall)}{precision + recall}$$

where precision is defined as

$$\frac{\text{TRUE POSITIVE}}{\text{TRUE POSITIVE} + \text{FALSE POSITIVE}}$$

and recall is defined as

$$\frac{\text{TRUE POSITIVE}}{\text{TRUE POSITIVE} + \text{FALSE NEGATIVE}}.$$

Table 1 summarizes the performance of the three models using F1 Score.

3.2 Model Performance Analysis

Decision Tree

We set the maximum depth of decision tree to 2 because our dataset is very large. Setting the max depth more than 2 is too computationally demanding.

Logistic Regression

Logistic Regression for muilt-class classification is to train as many models as as the distinct labels and output the label with the largest possibility. In our case, the model trained four binary logistic models for each label. In the preliminary model, we use logistic regression with no regularization and no weight on training. The training process using Logistic Regression in sklearn is quite slow (about 5 hours) on our training sets.

LDA

The main idea of LDA is to train distinct models using data with the same label and to calculate $P(Y = k|X = x)$ using Bayes Theorem. The output is $\arg\max_{k \in K} P(Y = k|X = x)$. The method is expected to have more balanced performance on each label, which can be inferred from Table 1.

Conclusion

All three models have bad performance on cases that are denied. This is because data with denied labels constitute only around 1 percent of the entire dataset.

4 Further Steps

There are several problems after pre-processing and preliminary model.

- The one-hot vectors for each categorical feature are too sparse and may contain redundant features, causing the data set to be too large to try some complicated models by using sklearn. Even reading the data into python would need 2.5 minutes. The jupyter notebook kernals always fail due to RAM limit when fitting a model.

- For this muilt-class classification problem, we need to find a reasonable way to measure the performance of models on the whole dataset instead of using the measurement for each class separately.

- Though the total accuracy is very high, the accuracy for each label is quite different, and the predictions on WITHDRAWN and DENIED are especially terrible. It is mainly because that the data set is imbalanced on each label. The CERTIFIED labels account for over 90% of the total labels.

Thus, our further steps would be around feature selections and further improve our models on imbalanced dataset. We may include weights in different classes or use other ways to boost our data in the training. For the feature selection, we are planning to use PCA for reducing dimension of the features while preserving as much important information as we can and use regularization for feature selections. Also, to solve the problem on RAM limit crash, we would apply spark (pyspark in python) to train those models.

5 Appendix

5.1 Data Intro

Table 2: Counts of NA for each feature

Name	NACount	Note
Sector_data	2	Delete
EMP_STATE_full	2358	Delete
EMP_State_and_city	0	Delete
Worksite_STATE_full	2453	Delete
Worksite_State_and_city	0	Delete
EMPLOYER_PHONE	1	Delete
AGENT_ATTORNEY_NAME	227034	Delete
AGENT_ATTORNEY_CITY	840452	Delete
AGENT_ATTORNEY_STATE	899957	Delete
JOB_TITLE	11	Delete
EMPLOYER_POSTAL_CODE	57	Delete
WORKSITE_POSTAL_CODE	70	Delete
YEAR	0	Delete
WORKSITE_COUNTY	4616	Delete
WORKSITE_CITY	41	Delete
WAGE_RATE_OF_PAY	0	Delete
SOC_NAME	20	Delete
EMPLOYER_CITY	29	Delete
CASE_NUMBER	0	Delete
CASE_STATUS	0	label
CASE_SUBMITTED	1	calculate duration
DECISION_DATE	0	calculate duration
EMPLOYMENT_START_DATE	38	calculate duration
EMPLOYMENT_END_DATE	47	calculate duration
EMPLOYER_COUNTRY	0	Only consider America.
PREVAILING_WAGE	0	Normalize it to yearly wage.
PW_UNIT_OF_PAY	0	Normalize it to yearly wage.
WAGE_UNIT_OF_PAY	19	Normalize it to yearly wage.
total_wage	0	Normalize it to yearly wage.
NAIC_CODE	2	Keep three digits.
EMP_STATE_abb	0	For geographical figures.
SOC_CODE	18	Keep three digits.
EMPLOYER_NAME	67	Count frequency
VISA_CLASS	0	
FULL_TIME_POSITION	7	
Worksite_STATE_abb	16	