5741 Project Final Report: H-1B Petition Analysis

Yueran Yang (yy595), Zongyuan Yuan (zy225)

1 Project Introduction

1.1 What is H1-B?

The H-1B is a visa in the United States that allows U.S. employers to temporarily employ foreign workers in specialty occupations. Labor Condition Application (LCA) for the employee that includes information such as wages and job titles should be filed by the employer to the U.S. Department of Labor Employment and Training Administration to show that the employer want to hire a foreign worker in a specific position for no more than 3 years. Once LCA is approved, the next stage of the H-1B is a random selection process often referred to as "H-1B Lottery". The lottery process will ultimately decide whether an applicant will get the H-1B visa. Effective 2022, a new policy will be applied to the lottery process, but the LCA stage remains unchanged. Due to the random nature of the lottery process, our project focuses only on the LCA stage.

1.2 Project Goal

The data set we are investigating in this project is the H1-B Visa Application data from Kaggle. Using the historical data, our goal is to predict the status of an application given the background information of an applicant. Our goal is not only to come up with a model that can predict the outcome of an Labor Condition Application with high accuracy, but also to identify important factors that play the biggest roles in the application. In other words, we want to make our model not as a black box but intepretable.

1.3 Project Significance

Hiring a foreign worker for a position brings additional cost to an employer because the employer has to pay for H-1B application fees and legal service fees. If an employer can know beforehand how likely their LCA for an employee is going to be approved, the employer can better decide whether they should hire a foreign candidate for a position. Our solution can not only make employers more informed about their risks in hiring a foreign but also save their cost by preventing filing an LCA that is unlikely going to be approved in the first place. This can save money and time for the companies to select their target employees for different positions. On the other hand, from the perspective of a job seeker, our solution can help him/her 1) compare between different job offers he/she has in terms of the chance of getting an H-1B 2) choose a job with more stable prospect 3) better plan their career path, which can make this two-way selection more efficient.

2 Data Analysis

2.1 About the Dataset

Our raw data, the H1b_EDA_data.csv file, has 2,448,729 rows and 36 columns with mixed data types containing numerical values, dates, categorical values, and texts. All features and counts of NA entry for each feature is shown in Table 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 Preliminary Data Processing

After conducting a preliminary investigation into each feature, we decided to discard the following features from our model:

[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/

feature	reason	feature	reason
Sector_data	Irrelevant information	WORKSITE_POSTAL_CODE	Duplicate information
EMP_STATE_full	Duplicate information	YEAR	Duplicate information
EMP_STATE_and_city	Simplicity	WORKSITE_COUNTY	Duplicate information
EMPLOYER_PHONE	Irrelevant information	WORKSITE_CITY	Duplicate information
AGENT_ATTORNEY_NAME	Irrelevant information	WAGE_RATE_OF_PAY	Duplicate information
AGENT_ATTORNEY_CITY	Irrelevant information	SOC_NAME	Duplicate information
AGENT_ATTORNEY_STATE	Irrelevant information	EMPLOYER_CITY	Duplicate information
JOB_TITLE	Duplicate information	CASE_NUMBER	Irrelevant information
EMPLOYER_POSTAL_CODE	Duplicate information		

These features are discarded because either they are 1. irrelevant, 2. contain essentially the same information as other features, or 3. for the sake of simplicity of our model. Specifically, EMPLOYER_PHONE apparently has nothing to do with the outcome of the application; SOC_CODE already contains the information SOC_NAME, JOB_TITLE can provide, and EMP_STATE_abb already encodes the location information of the employer, so there is no need to include EMP_STATE_full and EMPLOYER_POSTAL_CODE. For EMP_STATE_and_city, it contains too detailed information of the location of the employer. If we one-hot encode the feature, the dimension of the transformed dataset would be too large to handle. Also, there might be risk of over-fitting the data. Therefore, we are only interested in the state location of employers.

In the next step, we removed NA values for each remaining feature. From Table 1, we can see that remaining features do not have a lot of missing values compared to the size of the dataset. Thus, removing NA values has minimal impact.

Table 1: Counts of NA for remaining feature

Name	NACount	Name	NACount
CASE_STATUS	0	total_wage	0
CASE_SUBMITTED	1	NAIC_CODE	2
DECISION_DATE	0	EMP_STATE_abb	0
EMPLOYMENT_START_DATE	38	SOC_CODE	18
EMPLOYMENT_END_DATE	47	EMPLOYER_NAME	67
EMPLOYER_COUNTRY	0	VISA_CLASS	0
PREVAILING_WAGE	0	FULL_TIME_POSITION	7
PW_UNIT_OF_PAY	0	Worksite_STATE_abb	16
WAGE_UNIT_OF_PAY	19		

2.3 Exploratory Data Analysis and Further Data Processing

We first look at the distribution of CASE_STATUS. From Figure 1, we can see that most of the applications are certified. Therefore, the labels of our data set is very unbalanced. Hence, when we are tuning our models we have to make sure that we are setting our parameters to handle the imbalance.

Then we observed in Figure 9 that most applications are submitted by a small proportion of all companies. This means that the total demand for foreign workers concentrate in a small set of companies. If we simply apply one-hot encoding to the EMPLOYER_NAME, we will again have an extremely high-dimensional transformed dataset. Therefore, based on the observation we made in Figure 9, one approach to consider is to keep top 67 companies that submitted most applications which constitute around 80% of total applications and group the rest of companies as others. But in this way, we might overlook the diversified needs for foreign workers of companies with smaller demand. Instead, we counted the number of applications submitted by each company so that it can reflect the difference in need for foreign workers while keeping all companies in our dataset.

Wage is an important indicator of the value of the job. The violin plot in Figure 3 shows that average wages in certified applications are slightly higher than average wages in applications with other status. From the wage distribution plot below (Figure 4), we can see that most applications have wages from $50,000 to $100,000.

Another important factor is the location of the employer. From Figure 5, we can see that the applications from employers in some states are more likely to be certified than others. Additionally, as shown in Figure 6, the majority of the applications have SOC code that starts with 15 meaning that most of the applications are computer or software related.

Figure 1: Distribution of Labels

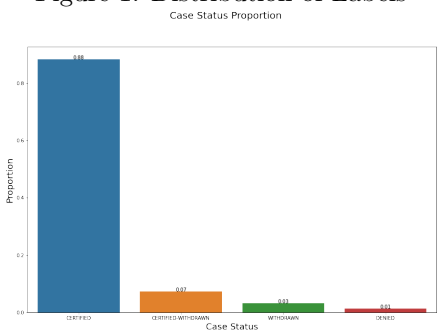


Figure 2: Cumulative Proportion of Applications by Top Companies

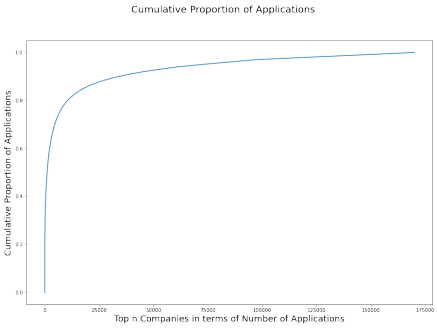


Figure 3: Wage Impact on Case Status

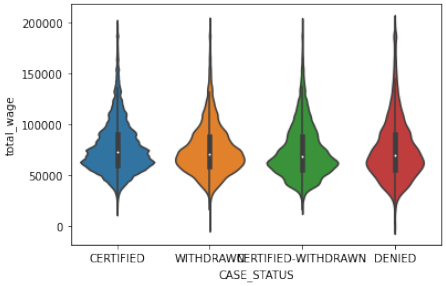


Figure 4: Wage Distribution

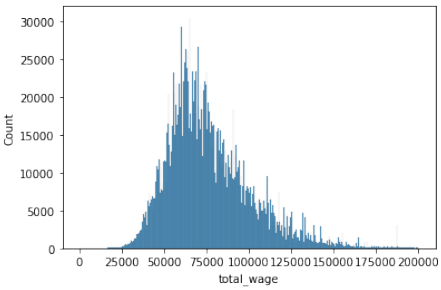


Figure 5: Certified Rate by States

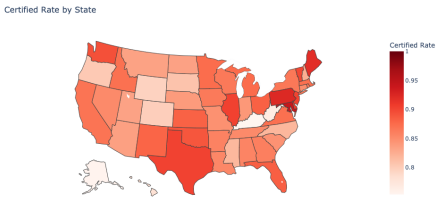
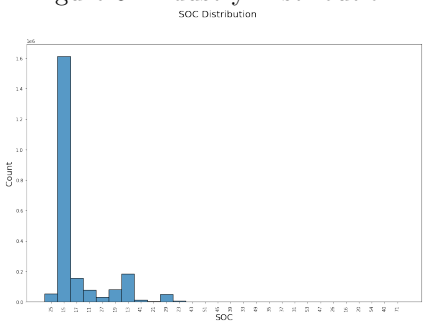


Figure 6: Industry Distribution



Finally, we have transformed date-related features in Table 1 into duration. Then we normalized all numerical features. We randomly chose 70% of the transformed dataset as our training set and used the rest as our test set.

3 Methods

In this section, all the models, sampling methods, metrics and how they are used are introduced in detail.

3.1 Models

3.1.1 Decision Tree

This is a preliminary model we are using in the beginning, thus training data for this model is not down-sampled. Because of the large size of our training, we are only setting the depth of the tree to 2. This is because setting it larger than 2 would make the run-time of the algorithm unreasonably long.

3.1.2 Logistic Regression & Linear SVM

Logistic regression can only used on binary classification. In multi-class problems, it simply generate 4 Logistic formula to calculate the possibility for each class to be positive and choose the most positive class as the final prediction. We have tried Logistic Regression with regularization on this data. We tried L1, L2 and elastic regularization, but no regularization outperforms. Thus the best model shown in the following evaluation is without regularization.

The linear SVM's main idea is to find a most robust boundary between the two classes. We use linear SVM on binary settings by setting the maximal iteration to be 100 and the regularization parameter to be 0.01 on binary settings.

We use pyspark to train the two models.

3.1.3 Linear Discriminant Analysis (LDA) & Quaduatic Discriminant Analysis (QDA)

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 prediction formula is

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{0.5}} \exp\left\{-\frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k)\right\}$$

where Σ is the correlation matrix for the features, μ_k is the mean and p is the dimension of the features. QDA, different from LDA, uses Σ_k instead of Σ for all the classes.

The two methods are expected to have more balanced performance on each label, but they do not work well in our problem.

3.1.4 Neural Network

Neural Network use interaction between neurons in different layers to simulate the human thinking process. The parameters between each layers can be a huge matrix to quantify the interaction. The Neural Network used for both kinds of classification contain a input layer, two hidden layer with 256 and 64 neurons respectively, a dropout layer with drop rate 0.5 to avoid over-fitting and an output layer. The activation for the two hidden layers is ReLU. The output activation is sigmoid. In a dropout layer, some number of layer outputs are randomly ignored with probability as the drop rate. We used tensorflow platform to train this model.

$$\text{Sigmoid: } S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = 1 - S(-x) \quad \text{ReLU: } f(x) = x^+ = \max(0, x) \tag{1}$$

3.1.5 AutoGluon[2]

AutoGluon[2] is an AutoML library, which can efficiently try many traditional or modern machine learning methods on the data and tune the models automatically. The output of AutoGluon can be a leaderboard of different methods with test score, shown in Table 2.

Table 2: Output of AutoGluon

model	score_test	score_val	pred_time_test
LightGBM[3]	0.980459	0.98111	9.922074
WeightedEnsemble_L2	0.980254	0.981521	219.2322
CatBoost[5]	0.979415	0.980113	1.61188
XGBoost [1]	0.97929	0.979878	11.71829
LightGBMXT	0.978001	0.978235	69.85277
NeuralNetFastAI	0.974666	0.974539	71.79455
KNeighborsDist	0.96803	0.968262	27.45305
KNeighborsUnif	0.967789	0.967441	26.48503
LightGBMLarge	0.881571	0.881556	5.466261

We use AutoGluon work as a very preliminary algorithm to help us to choose a model that can learn the mapping between labels and the features. Thus, the training data for AutoGluon in the first is non-down-sampled data which is the original training data, and the score is purely accuracy.

We can see from Table 2 that the AutoML chose lightGBM as the best algorithm on the data. Furthermore, the other models picked out by AutoGluon are mostly tree-based ensemble models. We can conclude that tree-models are more suitable on this problem and the mapping from features to labels are very complicated.

Remark: If we use balanced accuracy as score to evaluate each models, it turns out that the lightGBM is still the best model. The results is shown in figures/autogluon_balanced.png if you are interested.

3.1.6 Light Gradient Boosting Machine (LightGBM) [3]

Now we look into the detail of the best model picked by AutoGluon. LightGBM is a kind of gradient boosting decision tree algorithms, which improves the time efficiency and scalability on a large data set, compared to other gradient boosting decision trees. It is originally built by Microsoft. Gradient boosting decision tree is an ensemble of decision

trees (weak predictors). It use a loss function to measure the residual between output and the true labels. The algorithm use the formula (F_m is the current model and the h_m is the current tree)

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

to generate a ensemble model. And the γ_m is calculated by minimize the loss function, which is

$$\gamma_m = \arg\min \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

We used lightGBM package in python to train the model. In LightGBM, when it is dealing with binary classification problem, the model can handle imbalanced data by setting is_unbalanced to be True. This will add weights on the positive label to balance the classes, assuring that the model will not favor one specific class. Thus, we are trying two different objectives in LightGBM, one is to the original labels, which is a multi-class problem and the other is to use the one-vs-all (ova) method to deal with this multi-class classification. When using multi-class objective, we use our down-sampled data as training data. When using the ova method, we set the inner parameter is_unbalanced to be True.

3.2 Down-Sampling

The method is used to improve the performance of previous methods on each class. The original data is very imbalanced. About 90% of the labels are certified, and only approximately 1% of the labels are denied. In order to make our model more generalized to all the classes. We manually adjust the training data to make sure it contains the sample number of samples for each class.

There are totally 24,748 denied samples in our training data. Thus, we randomly sample 24,748 samples from the other 3 classes respectively, which makes the down-sampled training data to have 98,992 samples. The down-sampled training data will not be used in LDA and QDA because the prior possibility works as an essential part in both models. Also, the down-sampled method is not used on Decision Tree and LightGBM with the one-vs-all objective, as we mentioned before. The results of decision trees can be used to compare the binary classification and the 4-class classification.

For binary classification, we use the same method, but in this case, we only have two classes. The down-sampled data is used when there is a huge gap between accuracy on 1 and on 0. Thus, the down-sampled data is used on SVM, Neural Network and Logistic Regression.

3.3 Metrics

Since the data is very imbalanced, especially on denied status, instead of using pure accuracy to measure our models, we are using average accuracy (balanced accuracy) to make sure the final model we get will not favor any class. The balanced accuracy is the average of accuracy on each class. The balanced accuracy is only used in the last evaluation of each model, not in the training process as we already use down-sampled training data or other balanced methods to train the models. The average accuracy is defined as follows:

$$\frac{1}{4}\left(\text{accuracy}_{\text{certified}} + \text{accuracy}_{\text{certified-withdrawn}} + \text{accuracy}_{\text{withdrawn}} + \text{accuracy}_{\text{denied}}\right) \tag{2}$$

4 Fairness Analysis

Our features do not contain demographic information such as age, race and gender. Therefore, our models would not favor applicants from any demographic group. However, since our dataset is very imbalanced, if we do not add extra touch to our models, our predictions might favor certified cases more. Thus, we have taken the following two measures to ensure the fairness of our models.

1. Down sampling: Data with each label constitutes an equal proportion in our training set
2. Average accuracy: We care about the accuracy of the model for applications with different case status equally.

5 Results

5.1 Model Evaluation

5.1.1 Binary Classification

We can transform the certified status into 1 and the other three labels into 0 to make the original problem into a binary classification. The ratio of 1 and 0 would be 88:12, which is not as imbalanced as the original 4 classes. The results of each model are shown in Table 3

If we turn the models into binary classification, we can see that the accuracy of decision is high enough to separate the two classes. And the accuracy on the failed and the certified class is 0.91784763 and 0.98987939 respectively, which means that the decision tree can learn the data well.

Table 3: Evaluations of each model for binary classification

Models	Training Accuracy	Test Accuracy
Decision Tree	0.9820	0.9820
SVM	0.7885	0.8417
Neural Network	0.8933	0.9818
Logistic Regression	0.7908	0.8539
QDA	0.8275	0.8268
LDA	0.9303	0.9305

These are not the major results in our projects, so we are not going to dig into these results.

5.1.2 4-class classification

The performance of the models are shown in Table 4. According to the performance of each model and the output of the preliminary AutoGluon algorithm, the mapping from features to labels are so complicated that more flexible models (with more complexity) work better on this problem. Also the tree-based models generally work better than other kinds of model. Linear models are too simple to learn the data.

Table 4: Evaluations of each model for 4-class classification

Models	Training Average Accuracy	Test Average Accuracy	Test Accuracy			
			Certified	Certified-Withdrawn	Denied	Withdrawn
lightGBM (OVA)	0.8919	0.8804	0.9603	0.9650	0.7749	0.8215
lightGBM (multiclass)	0.8976	0.8860	0.9170	0.9532	0.8522	0.8214
Logistic Regression (best)	0.6070	0.6064	0.9708	0.8611	0.4034	0.1903
LDA (without downsampling)	0.5967	0.5980	0.9559	0.6993	0.4611	0.2757
Decision Tree (not down-sampled)	0.6659	0.6674	0.9900	0.9343	0.7452	0.0000
Neural Network	0.8760	0.8352	0.8733	0.9559	0.7689	0.7428

The reason that we are not using binary classification to analyze the certified rate in the following content is that the huge performance gap of Decision Tree on both problem shows that the binary classification may ignore some relations among features and labels. In order to use a more precise and reasonable model on each feature, we use lightGBM(ova) in the following analysis, since it is using model labels to train compared to other down-sampled models and having relatively higher average accuracy..

5.2 Analysis on lightGBM(ova)

From Figure 7, we can see that there is a huge gap between the importance of the top 9 features and the other features. Thus, we want to dig into the top 9 features. Also the Certified rate, certified-withdrawn rate, denied rate and withdrawn rate are highly related as their sum must be 1. In this case, we want to analyze certified chance. Thus, we will only specifically look into priori certified rate.

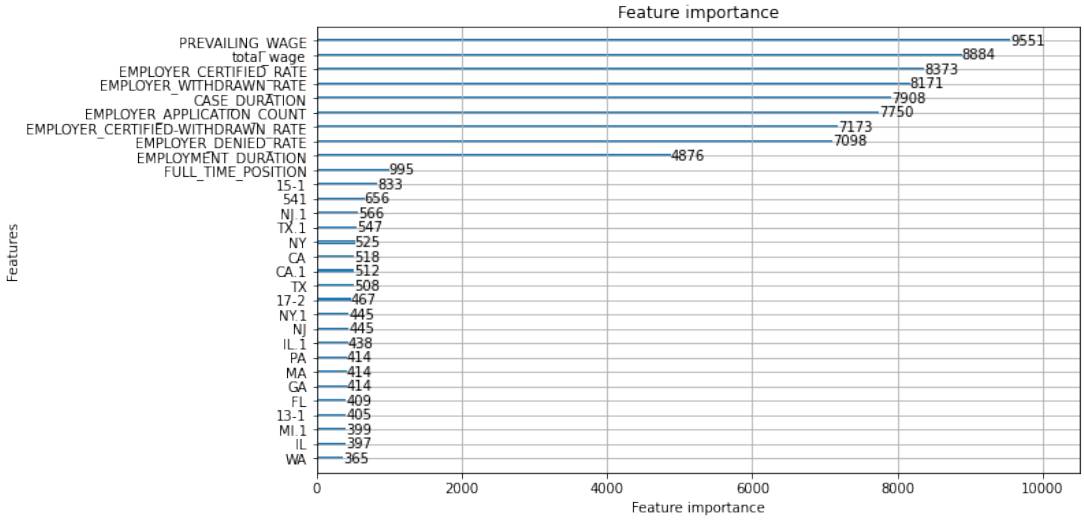


Figure 7: Top 30 important features

5.3 Feature Analysis

By analyzing the impact of these features in our model, we used artificial data, generated by randomly sampling 10 data from the test data and changing the value of the feature that we want to analyze while keeping the other features fixed. Then, we see the trend of the predicted possibility to be certified. The lines in the following figures are different samples from the test data.

Total Wage & Prevailing Wage

From Figure 8, if your wage is around average or slightly above average, there will be a slight increase in certified probability. But when it is too large, the predicted certified probability will decrease. The reason for the phenomena might be that there is some correlation between wage and other features, so when we adjust the wage manually, it will cause inconsistency to other features, which can be unreasonable to the U.S. Department of Labor Employment. And there might be some special position that cannot be approved to hire a foreign worker with a very high wage in our data.

Prevailing wage are showing the similar trend as the total wage.

Employer Certified Rate

This is a priori possibility for a employee in a company to be certified. To analyze the impact, we adjust the priori possibility from 0.5 to 1. In the mean time, to make the other three priori possibility work, we manually set the denied rate to be 1 - the certified rate we set and the other two to be 0. We can see from Figure 10 that the chance to be certified for a individual is highly related to the employer's total priori certified rate. The higher the rate, more possible for a employee to be certified.

Case Duration

From Figure 11, if you have a extreme case duration, the probability to be certified is very low. We can use the result to predict the final status, if you are waiting too long, you might be denied. If result comes out in a very short time, either you withdraw your application, or you might be denied.

Employer's application count

From Figure 12, application counts have a slight impact on the final status. We can still conclude that more applications before, there will be a higher chance to be certified for an employee.

Employment Duration

From Figure 13, there is no obvious impact on the certified probability. We can see slight decrease when the employment duration is very large. It can be because that the application happens in the very beginning of the employment (always in the beginning of OPT with longer employment duration), which would cause high denied rate, since being denied or withdrawn means that the person cannot continue to work in US.

Figure 8: Total Wage

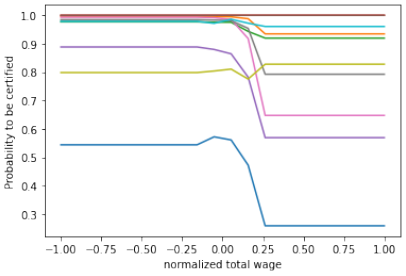


Figure 9: Prevailing Wage

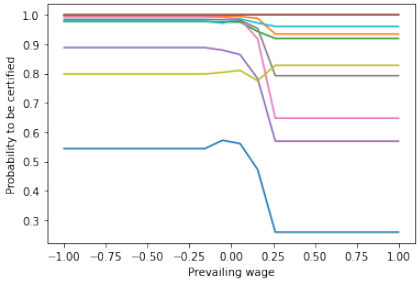


Figure 10: Employer certified rate

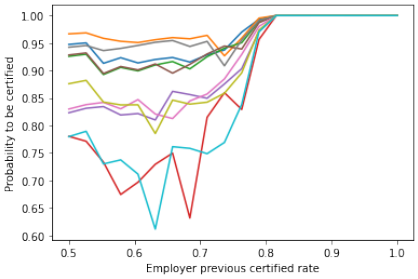


Figure 11: Case Duration

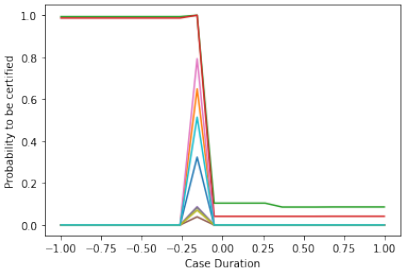


Figure 12: Employer's application count

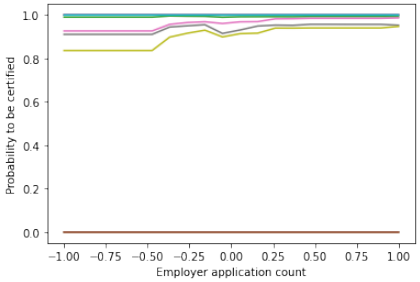
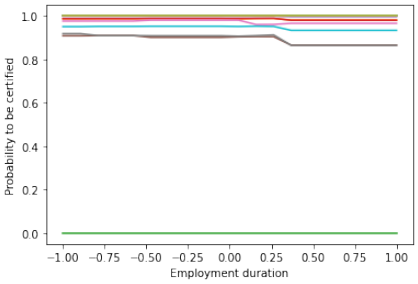


Figure 13: Employment Duration



6 Weapon of Math Destruction [4]

A model is considered a Weapon of Math Destruction (WMD) if it satisfies one of the following three criterion: 1. the outcome of a model is not easily measurable 2. can have negative consequences 3. has self-fulfilling feedback loop. Our model is not a WMD because it a has measurable outcome which is the status of an application. Secondly, our model has no negative consequence because most of the applications are accepted and our model focuses on helping those who might not be approved become aware of their risks. Furthermore, our model has no prominent self-fulfilling feedback loop. Whether an H1-B is going to be approved is determined entirely the competency of the employee and the industry demand. These two factors are unlikely to be affected by the prediction of the H1-B application status.

7 Conclusion

In this project, we explored the Labor Condition Application data and attempted various models to predict the case status of LCA application. The performance of each model is summarized in Table 4. The internal process of LCA application is very complicated. Based on our attempts, we found that tree-based models work better. More generally, linear models do not work well on our dataset while more complicated models have better performance. From a company's perspective, to increase the chance of being certified, it should offer a reasonable salary to its foreign employee but not too high. From the perspective of a foreign worker, to increase the chance of being certified, he/she should choose a big company that submitted more previous applications and/or choose a company with higher priori certified probability. But most importantly, one should increase his/her own competency to be accepted by a better company to increase his/her chance to be certified.

8 Future Work

In the analysis aspect, first of all, we trained a model that can predict the four class equivalently. The model can be further used in analyze the other two labels, which is withdrawn and certified withdrawn to understand specific pattern that may indicate the withdrawing. Secondly, we used synthetic data to analyze the feature impacts in our model. The model can be further evaluated and interpreted by the real data.

In the training aspect, more methods can be tried to deal with this imbalanced labels. Also, the correlation among features can be removed to make the model more accurate, but the detection of the correlation can be very difficult since most features are categorical in the original data.

References

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[2] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

[3] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[4] C. O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown, 2016.

[5] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features, 2019.