EasyVISA

Project 5 – Ensemble Techniques

Joshua Willis, PGP-DSBA

CONTENTS

Business Problem:

Business communities in the United States are facing high demand for human resources, but one of the constant challenges is identifying and attracting the right talent, which is perhaps the most important element in remaining competitive. Companies in the United States look for hard-working, talented, and qualified individuals both locally as well as abroad.

The Immigration and Nationality Act (INA) of the US permits foreign workers to come to the United States to work on either a temporary or permanent basis. The act also protects US workers against adverse impacts on their wages or working conditions by ensuring US employers' compliance with statutory requirements when they hire foreign workers to fill workforce shortages. The immigration programs are administered by the Office of Foreign Labor Certification (OFLC).

OFLC processes job certification applications for employers seeking to bring foreign workers into the United States and grants certifications in those cases where employers can demonstrate that there are not sufficient US workers available to perform the work at wages that meet or exceed the wage paid for the occupation in the area of intended employment.

Objective:

In FY 2016, the OFLC processed 775,979 employer applications for 1,699,957 positions for temporary and permanent labor certifications. This was a nine percent increase in the overall number of processed applications from the previous year. The process of reviewing every case is becoming a tedious task as the number of applicants is increasing every year.

The increasing number of applicants every year calls for a Machine Learning based solution that can help in shortlisting the candidates having higher chances of VISA approval. This presentation outlines a data-driven solution utilizing several classification models.

This goal is to establish a process for visa approvals and recommend a suitable profile for the applicants for whom the visa should be certified or denied based on the drivers that significantly influence the case status.

Data Summary

Data Description

The data contains the different attributes of employee and the employer. The detailed data dictionary is given below.

- case_id: ID of each visa application
- continent: Information of continent the employee
- education_of_employee: Information of education of the employee
- has_job_experience: Does the employee has any job experience? Y= Yes; N = No
- requires_job_training: Does the employee require any job training? Y = Yes; N = No
- no_of_employees: Number of employees in the employer's company
- yr_of_estab: Year in which the employer's company was established
- region_of_employment: Information of foreign worker's intended region of employment in the US.
- prevailing_wage: Average wage paid to similarly employed workers in a specific occupation in the area of intended employment. The purpose of the prevailing wage is to ensure that the foreign worker is not underpaid compared to other workers offering the same or similar service in the same area of employment.
- unit_of_wage: Unit of prevailing wage. Values include Hourly, Weekly, Monthly, and Yearly.
- full_time_position: Is the position of work full-time? Y = Full Time Position; N = Part Time Position
- case_status: Flag indicating if the Visa was certified or denied

Data Summary

Dependent variable:

- case_status = certified or denied

Object variables:

- case_id, continent, education_of_employee, has_job_experience, requires_job_training,
- region_of_employment, unit_of_wage, full_time_position, case_status

Integer variables:

- no_of_employees, yr_of_estab

Float variables:

- prevailing_wage

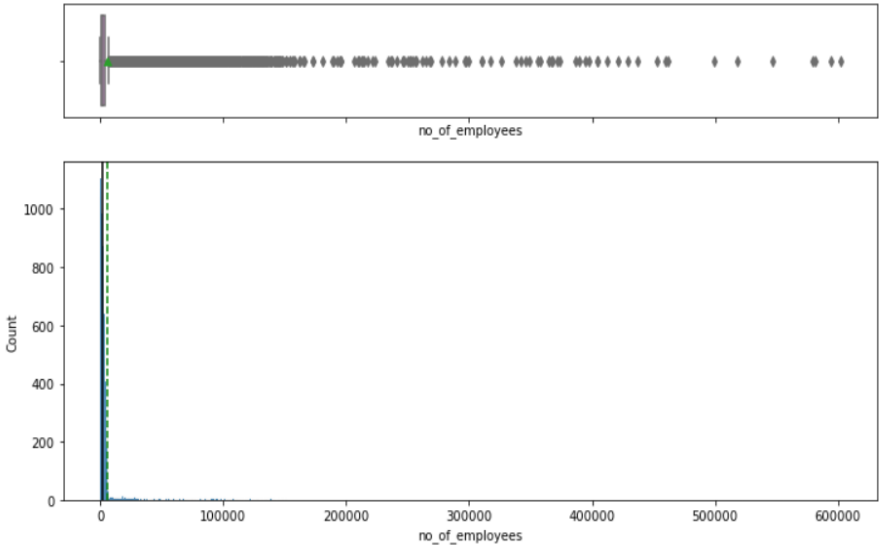
- Shape: 25,480 rows and 12 columns
- There are no missing values in the dataset
- There are no duplications in the dataset

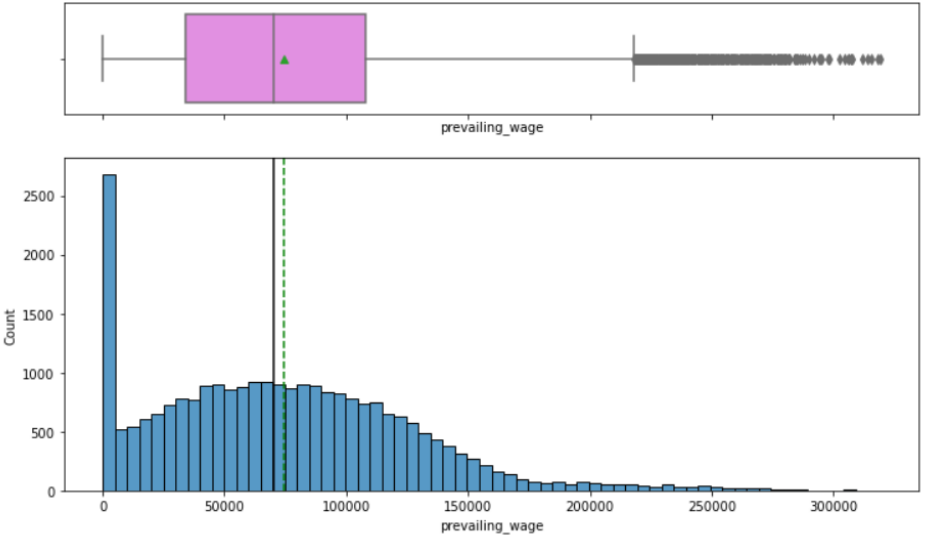
EDA Results

	count	mean	std	min	25%	50%	75%	max
no_of_employees	25480.0	5667.043210	22877.928848	-26.0000	1022.00	2109.00	3504.0000	602069.00
yr_of_estab	25480.0	1979.409929	42.366929	1800.0000	1976.00	1997.00	2005.0000	2016.00
prevailing_wage	25480.0	74455.814592	52815.942327	2.1367	34015.48	70308.21	107735.5125	319210.27

- Average number of employees is 5,667
- Average prevailing wage is $74,455
- There are outliers for all the above categories

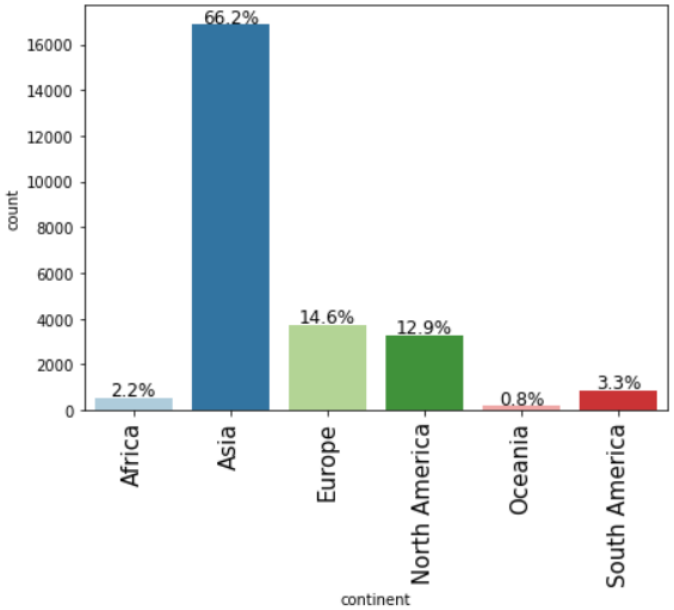
EDA Results

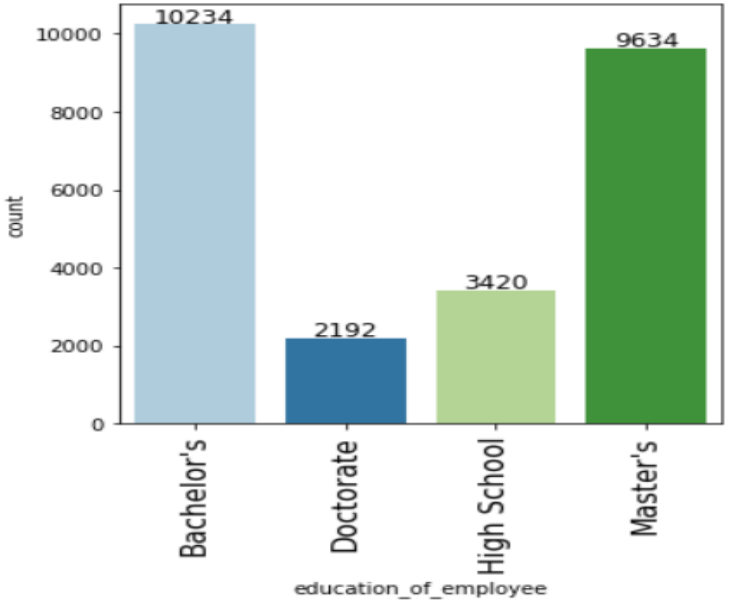




- Most companies had considerably less than 10,000 employees
- The distribution for prevailing wage is somewhat right skewed; a lot of people appear to have wages more than $74,455

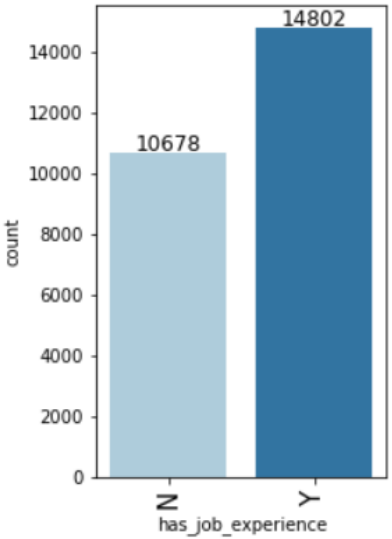
EDA Results

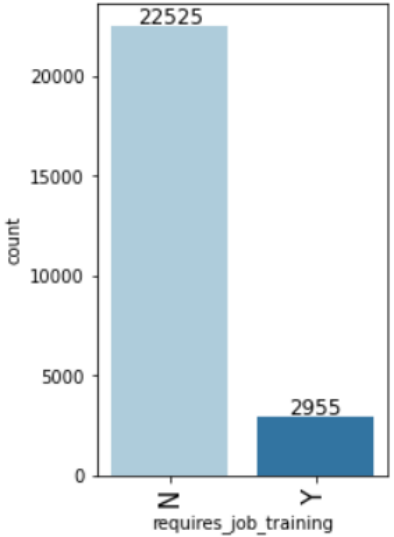


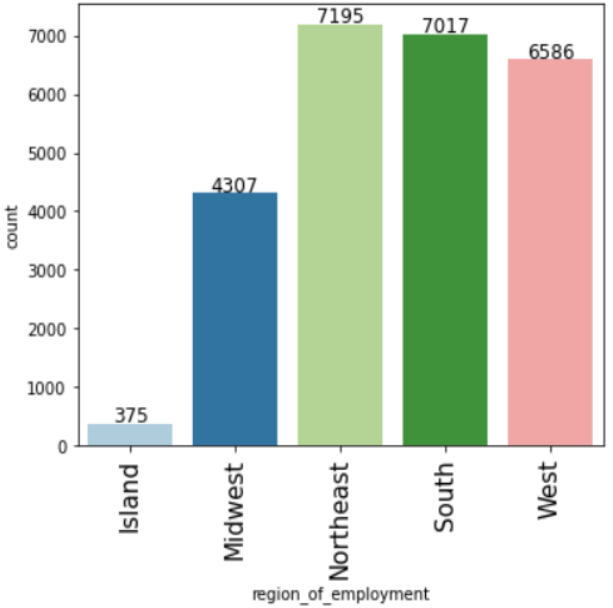


- Asia has the most applicants for visas followed by Europe, then North America
- Majority of applicants have a Bachelor's degree then followed by a Master's

EDA Results

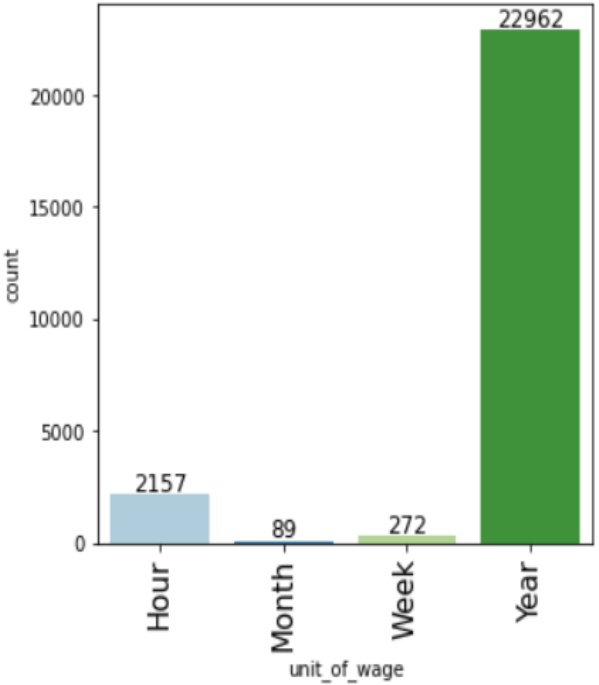




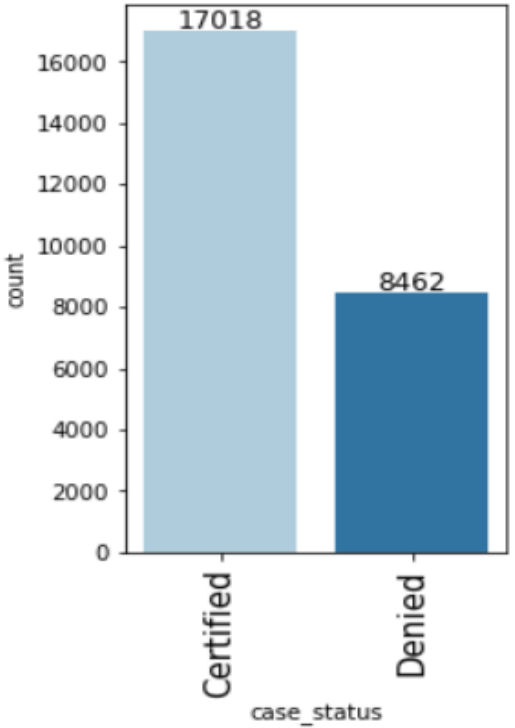


- Majority of applicants have job experience
- The majority of employers don’t require job training
- Majority of the intended regions for employment are for the Northeast, South, then West

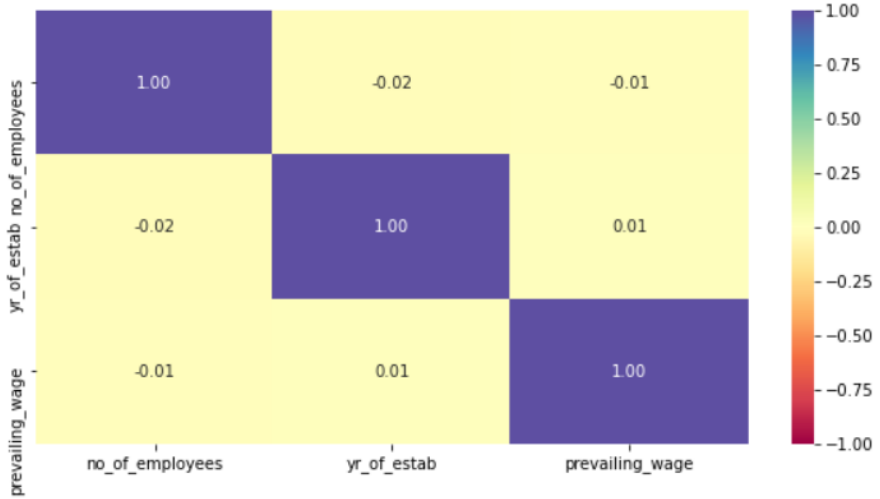
EDA Results

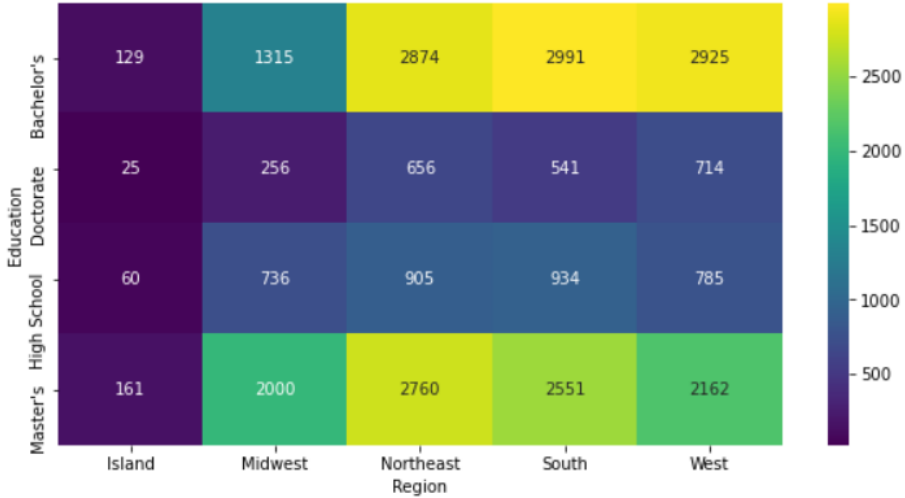


- Majority of employees unit of wage are reported on an annual basis
- Majority of cases are certified



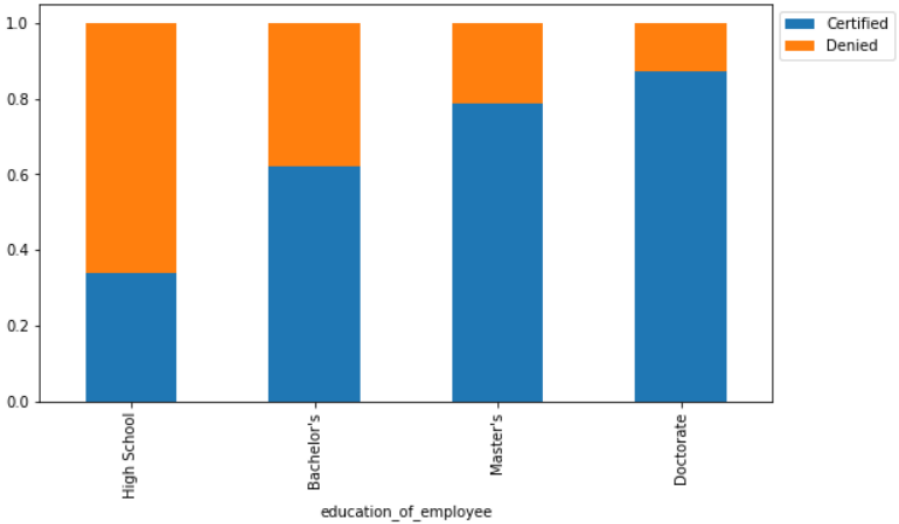
EDA Results

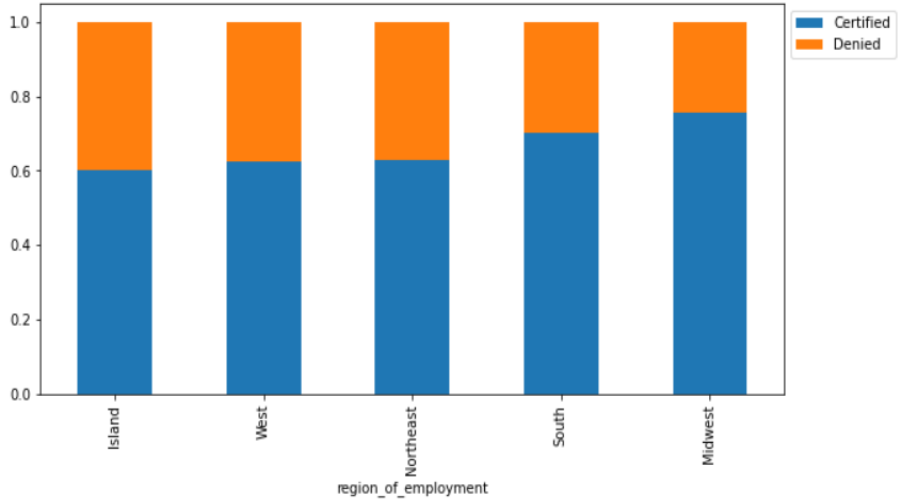




- No strong correlation between no_of_employees, yr_of_establishment, and prevailing_wage
- Very strong correlation with applicants for South, West, and Northeast regions having a Bachelor's degree
- Very strong correlation with applicants for Northeast region having a Master's degree
- Moderately strong correlation with applicants for Midwest, South, and West regions having a Master's degree

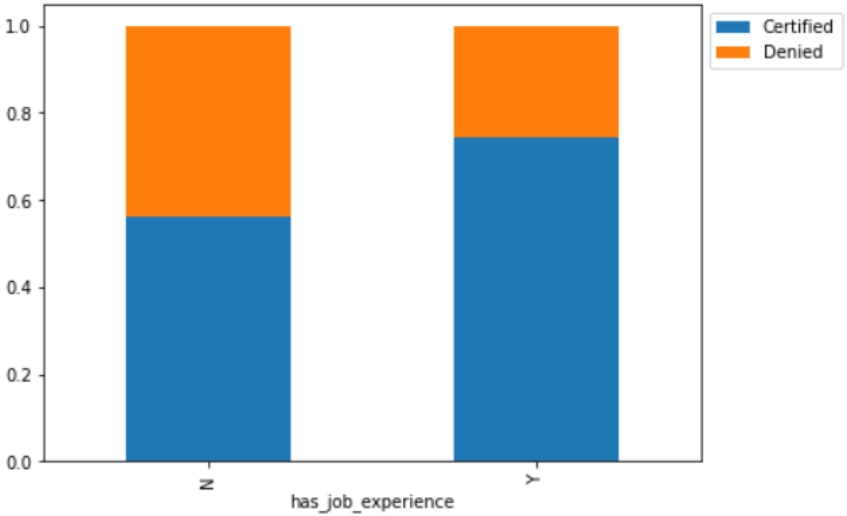
EDA Results

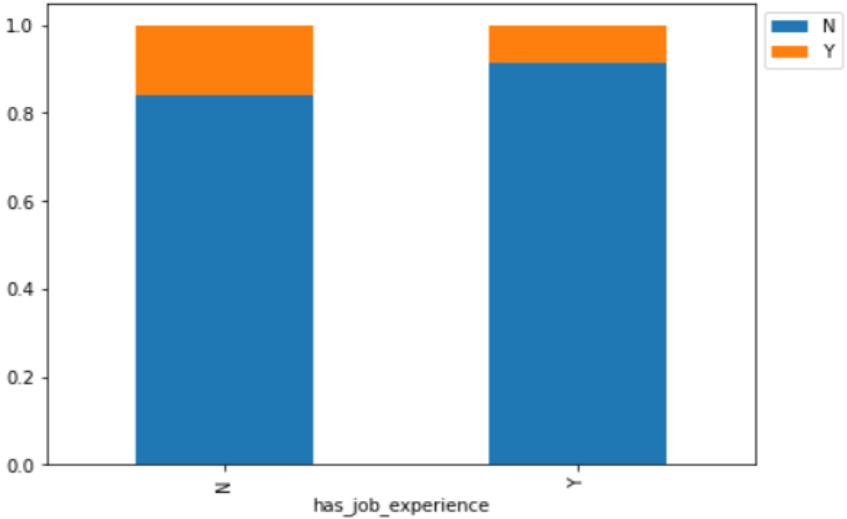




- Majority of the employees that were certified had a bachelor's degree and above
- More employees were certified in Midwest and South regions

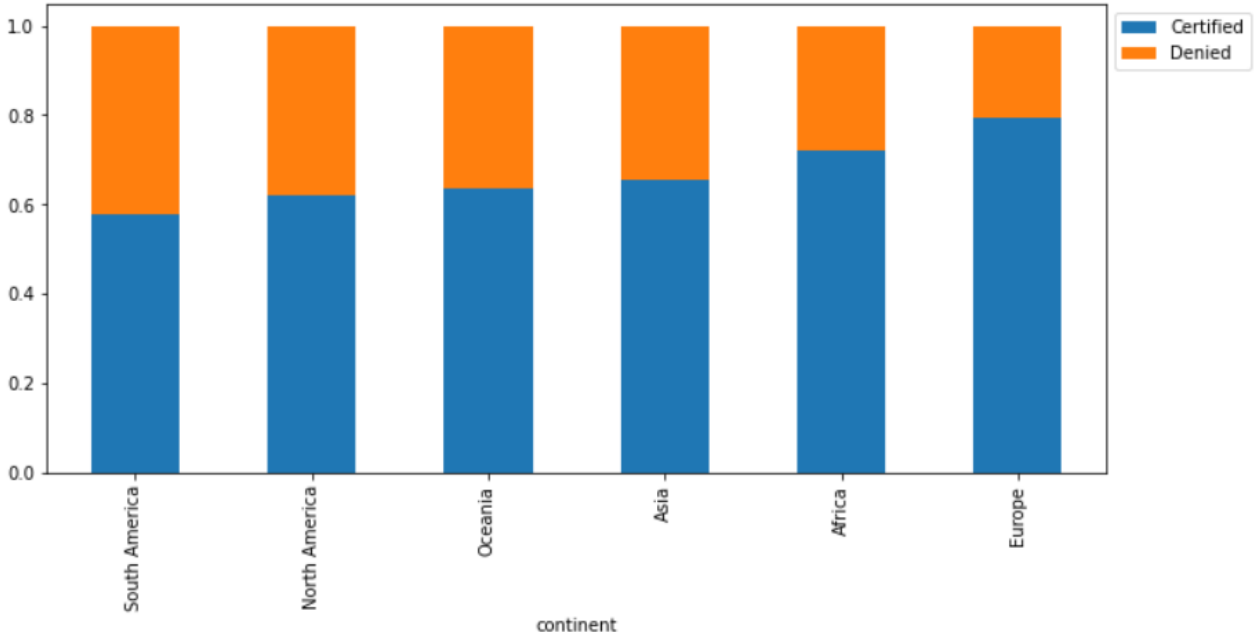
EDA Results





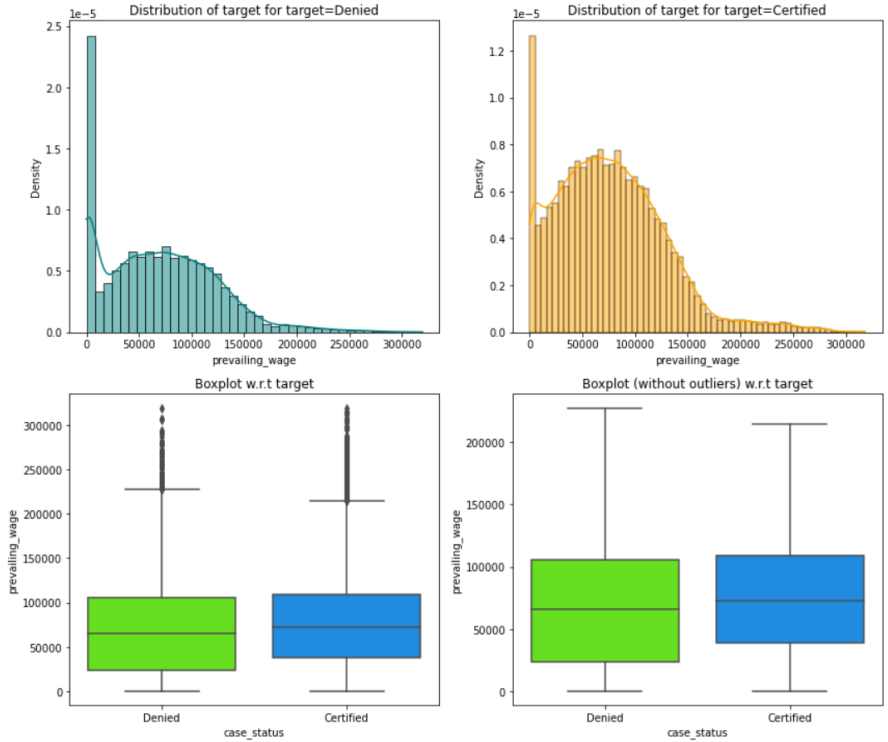
- Most employees who are certified have job experience
- Majority of the jobs don't require job training, but the employee has job experience

EDA Results



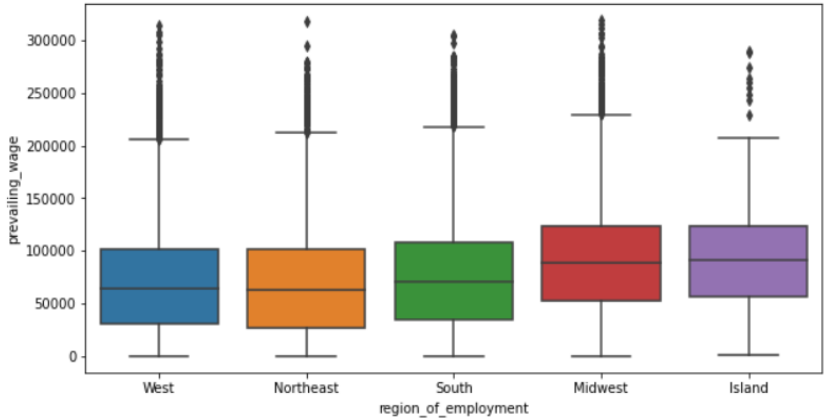
- Most cases certified were from European and African continents

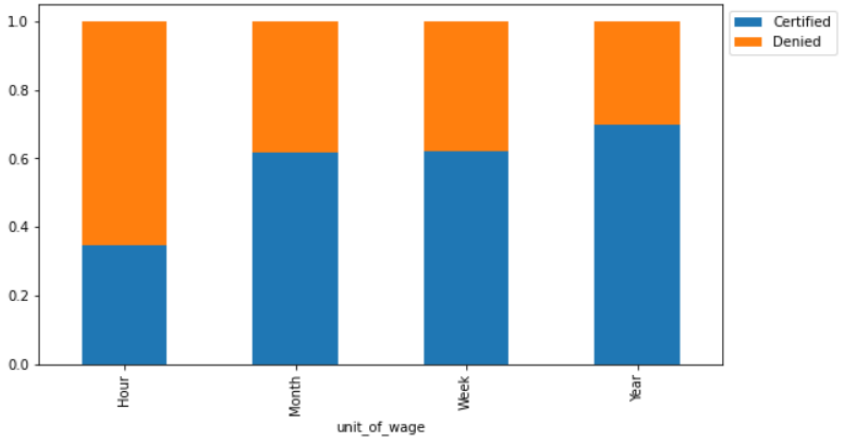
EDA Results



- Appears that prevailing wage is generally higher for those that are certified compared to applications that are denied.

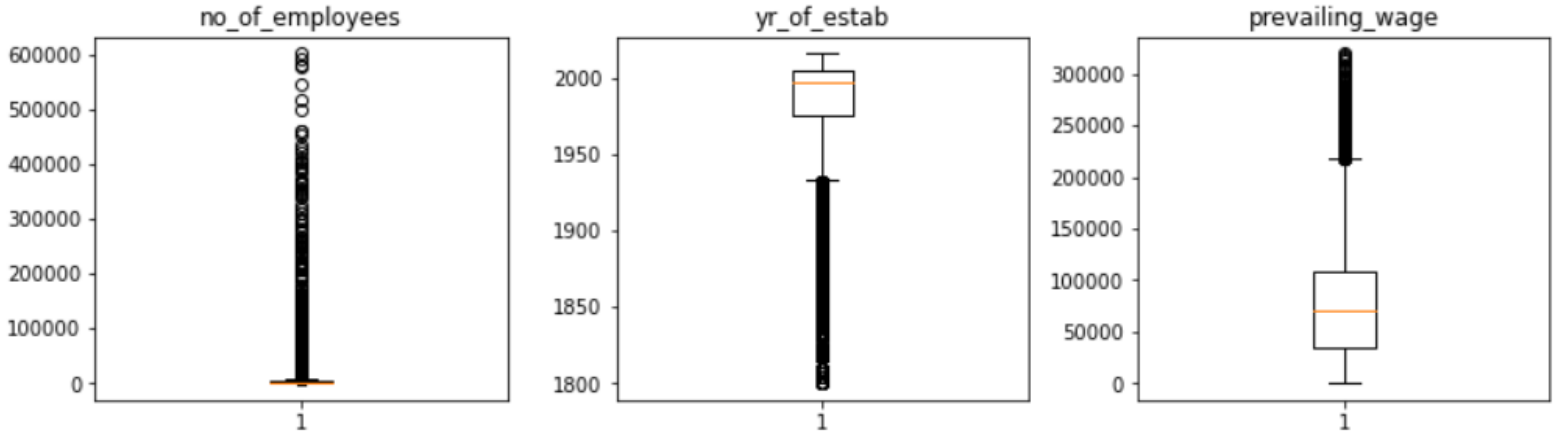
EDA Results





- Prevailing wages are higher in the Midwest and Island regions of the US
- Hourly wage employee applicants were denied significantly more than weekly, monthly, and yearly unit of wages

EDA Results



- Outliers for data doesn't need to be treated; negative number of employees have already been treated by converting negative value into positive ones.

Key EDA Observations

- Most of employee applicants were from Asia
- Majority of applicants had a Bachelor's or Master's degree
- Majority of applicants had previous job experience
- Majority of employers don't require job training
- Majority of visa applications received were certified
- Majority of applicants that were certified either had a Bachelor's or Master's degree
- Most of the applicants that were certified were from Europe
- Most of the applicants who certified has previous job experience

Data Preprocessing

- Dropped case_id feature due to no value added for analysis
- Converted negative no_of_employees value to absolute value (positive)
- Converted object data to category type to assist with processing speeds of models

Model Evalution: Why is it Important?

Model could make the following wrong predictions:

- Model predicts application will be certified but it should be denied
- Model predicts application will not be certified but it should be certified

Both cases are important:

- If a visa is certified and should be denied then the wrong employee will get the job while a US citizen will miss out on the opportunity
- If a visa is denied and should’ve been certified then the US will miss an opportunity for a foreign human resource that can contribute to the economy

How reduce losses:

- F1 scoring will be used for model evaluation
- Balanced class weights will be used so the model will focus on each classes equally

Model Building (Decision Tree):

- Performed model using Decision Tree Classifier
- Initial Training performance is 1
- Initial Testing performance is 0.75
- Will need to tune model; model is overfitting on training and testing is just average

Training

	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0

Testing

	Accuracy	Recall	Precision	F1
0	0.664835	0.742801	0.752232	0.747487

Model Tuning (Decision Tree):

- Hyperparameters used for tuning:

```
dtree_estimator = DecisionTreeClassifier(class_weight="balanced", random_state=1)

parameters = {
    "max_depth": np.arange(5, 16, 5),
    "min_samples_leaf": [3, 5, 7],
    "max_leaf_nodes": [2, 5],
    "min_impurity_decrease": [0.0001, 0.001],
}
```

Model is still not good after tuning

Results:

Training:

	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0

Testing:

	Accuracy	Recall	Precision	F1
0	0.664835	0.742801	0.752232	0.747487

Model Building (Bagging):

- Performed model using Bagging Classifier
- Initial Training performance is 0.98
- Initial Testing performance is 0.76
- Will need to tune model; model is overfitting on training and average on testing

Training

	Accuracy	Recall	Precision	F1
0	0.985198	0.985982	0.99181	0.988887

Testing

	Accuracy	Recall	Precision	F1
0	0.691523	0.764153	0.771711	0.767913

Model Tuning (Bagging):

- Hyperparameters used for tuning:

- bagging_estimator_tuned = BaggingClassifier(random_state=1)

- # Grid of parameters to choose from
- parameters = {
- "max_samples": [0.7, 0.9],
- "max_features": [0.7, 0.9],
- "n_estimators": np.arange(90, 111, 10),
- }

Model is still not good after tuning

Results:

Training:

	Accuracy	Recall	Precision	F1
0	0.996187	0.999916	0.994407	0.997154

Testing:

	Accuracy	Recall	Precision	F1
0	0.724228	0.895397	0.743857	0.812622

Model Building (Random Forest):

- Performed model using Random Forest Classifier
- Initial Training performance is 0.99
- Initial Testing performance is 0.80
- Will need to tune model; model is overfitting on training and average on testing

Training

	Accuracy	Recall	Precision	F1
0	0.985198	0.985982	0.99181	0.988887

Testing

	Accuracy	Recall	Precision	F1
0	0.691523	0.764153	0.771711	0.767913

Model Tuning (Random Forest):

- Hyperparameters used for tuning:

```
rf_tuned = RandomForestClassifier(random_state=1, oob_score=True, bootstrap=True)

parameters = {
    "max_depth": list(np.arange(5, 15, 5)),
    "max_features": ["sqrt", "log2"],
    "min_samples_split": [5, 7],
    "n_estimators": np.arange(15, 26, 5),
}
```

Results:

Model has gotten better from tuning; training and testing results are close to one another

Training:

	Accuracy	Recall	Precision	F1
0	0.769119	0.91866	0.776556	0.841652

Testing:

	Accuracy	Recall	Precision	F1
0	0.738095	0.898923	0.755391	0.82093

Model Building (AdaBoost):

- Performed model using AdaBoost Classifier
- Initial Training performance is 0.81
- Initial Testing performance is 0.81
- Model isn't great; training and testing scoring is similar
- Will tune model further

Training

	Accuracy	Recall	Precision	F1
0	0.738226	0.887182	0.760688	0.81908

Testing

	Accuracy	Recall	Precision	F1
0	0.734301	0.885015	0.757799	0.816481

Model Tuning (AdaBoost):

- Hyperparameters used for tuning:

```
abc_tuned = AdaBoostClassifier(random_state=1)

parameters = {
 "base_estimator": [
       DecisionTreeClassifier(max_depth=1, class_weight="balanced", random_state=1),
       DecisionTreeClassifier(max_depth=2, class_weight="balanced", random_state=1),
    ],
    "n_estimators": np.arange(80, 101, 10),
    "learning_rate": np.arange(0.1, 0.4, 0.1),
}
```

Results:

Model has gotten worse from tuning; training and testing are okay but not great

Training:

	Accuracy	Recall	Precision	F1
0	0.718995	0.781247	0.794587	0.787861

Testing:

	Accuracy	Recall	Precision	F1
0	0.71651	0.781391	0.791468	0.786397

Model Building (Gradient Boosting Classifier):

- Performed model using Gradient Boosting Classifier
- Initial Training performance is 0.83
- Initial Testing performance is 0.82
- Model is pretty good so far; training and testing are close to each other
- Will tune to see if better results can be achieved

Training

	Accuracy	Recall	Precision	F1
0	0.758802	0.88374	0.783042	0.830349

Testing

	Accuracy	Recall	Precision	F1
0	0.744767	0.876004	0.772366	0.820927

Model Tuning (Gradient Boosting):

- Hyperparameters used for tuning:

```
gbc_tuned = GradientBoostingClassifier(
    init=AdaBoostClassifier(random_state=1), random_state=1
)

parameters = {
    "n_estimators": [200, 250],
    "subsample": [0.9, 1],
    "max_features": [0.8, 0.9],
    "learning_rate": np.arange(0.1, 0.21, 0.1),
}
```

Results:

Model has gotten marginally worse from tuning

Training:

	Accuracy	Recall	Precision	F1
0	0.764017	0.882649	0.789059	0.833234

Testing:

	Accuracy	Recall	Precision	F1
0	0.743459	0.871303	0.773296	0.819379

Model Building (XGBoost Classifier):

- Performed model using XGBoost Classifier
- Initial Training performance is 0.88
- Initial Testing performance is 0.81
- Model is good; training and testing results are not close together
- Will tune the model to see if results can be achieved

Training

	Accuracy	Recall	Precision	F1
0	0.838753	0.931419	0.843482	0.885272

Testing

	Accuracy	Recall	Precision	F1
0	0.733255	0.860725	0.767913	0.811675

Model Tuning (XGBoost Classifier):

- Hyperparameters used for tuning:

```
xgb_tuned = XGBClassifier(random_state=1, eval_metric="logloss")

parameters = {
    "n_estimators": np.arange(150, 250, 50),
    "scale_pos_weight": [1, 2],
    "subsample": [0.9, 1],
    "learning_rate": np.arange(0.1, 0.21, 0.1),
    "gamma": [3, 5],
    "colsample_bytree": [0.8, 0.9],
    "colsample_bylevel": [0.9, 1],
}
```

Model has gotten better from tuning; training and testing results are closer to each other

Results:

Training:

	Accuracy	Recall	Precision	F1
0	0.765474	0.881642	0.791127	0.833935

Testing:

	Accuracy	Recall	Precision	F1
0	0.74516	0.86954	0.775913	0.820063

Model Building (Stacking Classifier):

- Performed model using Stacking Classifier
- Training performance is 0.83
- Testing performance is 0.82
- Model is decent; training and testing results are close to eachother
- Will not tune since this model uses predictions from AdaBoost, Gradient Boosting, and Random Forest

Training

	Accuracy	Recall	Precision	F1
0	0.765474	0.881642	0.791127	0.833935

Testing

	Accuracy	Recall	Precision	F1
0	0.74516	0.86954	0.775913	0.820063

Comparison of Models

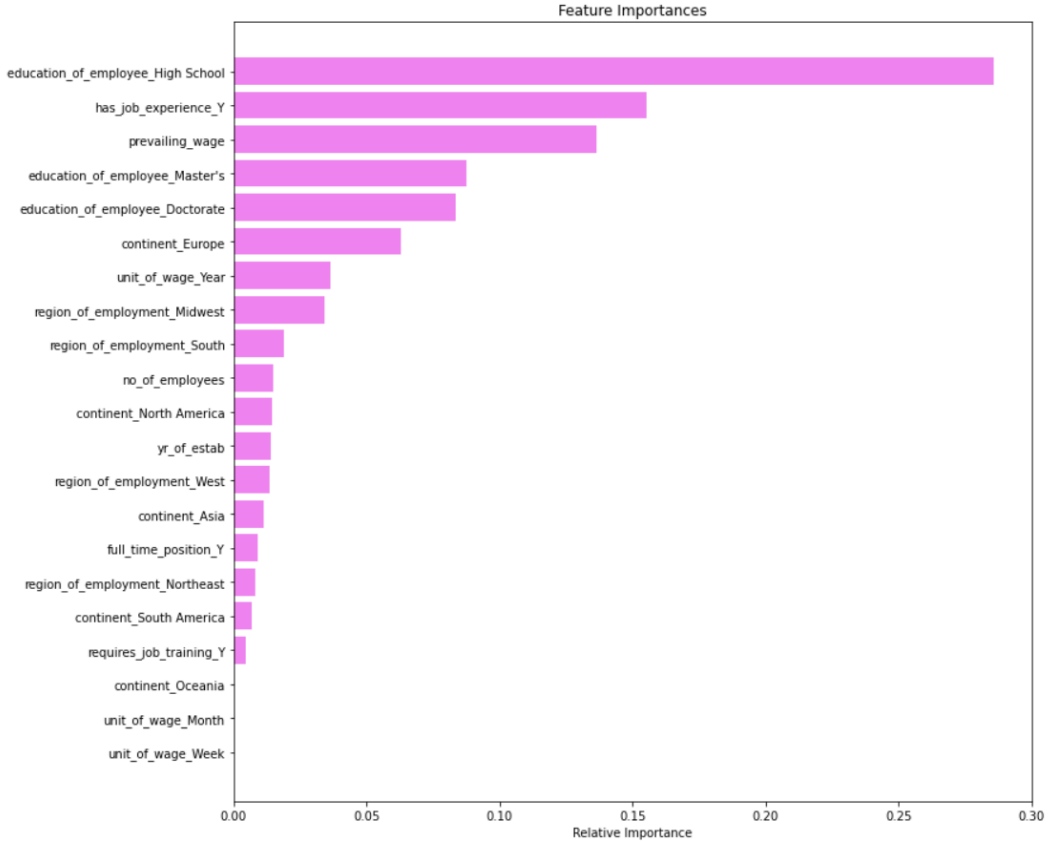
Training Models:

	Decision Tree	Tuned Decision Tree	Bagging Classifier	Tuned Bagging Classifier	Random Forest	Tuned Random Forest	Adaboost Classifier	Tuned Adaboost Classifier	Gradient Boost Classifier	Tuned Gradient Boost Classifier	XGBoost Classifier	XGBoost Classifier Tuned	Stacking Classifier
Accuracy	1.0	1.0	0.985198	0.996187	0.999944	0.769119	0.738226	0.718995	0.758802	0.764017	0.838753	0.765474	0.765474
Recall	1.0	1.0	0.985982	0.999916	0.999916	0.918660	0.887182	0.781247	0.883740	0.882649	0.931419	0.881642	0.881642
Precision	1.0	1.0	0.991810	0.994407	1.000000	0.776556	0.760688	0.794587	0.783042	0.789059	0.843482	0.791127	0.791127
F1	1.0	1.0	0.988887	0.997154	0.999958	0.841652	0.819080	0.787861	0.830349	0.833234	0.885272	0.833935	0.833935

Testing Models:

	Decision Tree	Tuned Decision Tree	Bagging Classifier	Tuned Bagging Classifier	Random Forest	Tuned Random Forest	Adaboost Classifier	Tuned Adaboost Classifier	Gradient Boost Classifier	Tuned Gradient Boost Classifier	XGBoost Classifier	XGBoost Classifier Tuned	Stacking Classifier
Accuracy	0.664835	0.664835	0.691523	0.724228	0.721088	0.738095	0.734301	0.716510	0.744767	0.743459	0.733255	0.745160	0.745160
Recall	0.742801	0.742801	0.764153	0.895397	0.840744	0.898923	0.885015	0.781391	0.876004	0.871303	0.860725	0.869540	0.869540
Precision	0.752232	0.752232	0.771711	0.743857	0.764926	0.755391	0.757799	0.791468	0.772366	0.773296	0.767913	0.775913	0.775913
F1	0.747487	0.747487	0.767913	0.812622	0.801045	0.820930	0.816481	0.786397	0.820927	0.819379	0.811675	0.820063	0.820063

Important Features of the Final Model



- When comparing all the models the Gradient Boost Classifier Model had the best results for training and testing data

- The important features of the model were:

- ❑ education_of_employee_High_School
- ❑ has_job_experience_Y
- ❑ prevailing_wage
- ❑ education_of_employee_Master's
- ❑ eduacation_of_employee_Doctorate

CONCLUSIONS

- We were able to create a profile of applicants that have better chances at being certified and can be shortlisted with F1 score of 0.83 and 0.82 for training and testing respectively
- Important profile features are high school education, job experience, prevailing wage, Master's and Doctorate education
- The applicants with education of bachelors and beyond and previous job experience should be shortlisted as these are candidates who will likely get certified
- Applicants with only high school education and no experience should not be shortlisted as they will more than likely be denied

RECOMMENDATIONS

- To reduce the number of applications, offer disclaimer to applicants that priority is given to applicants with post high school education and work experience
- Have internal controls that use education and work experience to automatically filter applications accordingly
- Instead of denying those who don't have higher education and experience immediately put them on a waitlist to where they can be given priority once degree is completed and job experience established
- Work with companies to provide more job training for applicants regardless of previous job experience
- Reduce time frame of application window and only accept predetermined number of applications on a first-come, first-serve basis. This will help to better plan resources to process visa applications.

THANKS!