Easy Visa

(Business Report)

Report By:
Naman Srivastava
05 Oct, 2025

TABLE OF CONTENT

LIST OF FIGURES

CONTEXT

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 permit 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 a higher chance of VISA approval. OFLC has hired the firm EasyVisa for data-driven solutions. You, as a data scientist at EasyVisa, have to analyze the data provided and, with the help of a classification model:

1. Facilitate the process of visa approvals.
2. 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 DESCRIPTION

The data contains the different factors to analyse for the content. The detailed data dictionary is given below.

Variable	*Description*
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 have 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

EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis was done on the provided data set (EasyVisa.csv) using Python tools on google colab. The objective of entire process was done to make data more understandable, reliable for meaningful decision making.

Data Import and Cleaning

After successfully loading the data on the google colab notebook, and importing all the required libraries, we found out that on initial checking, that the data consists of **25480 entries, and 12 features**.

- Loading head and tail of the data

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_of_employees	yr_of_estab	region_of_employment	prevailing_wage	unit_of_wage	full_tim
0	EZYV01	Asia	High School	N	N	14513	2007	West	592.2029	Hour	
1	EZYV02	Asia	Master's	Y	N	2412	2002	Northeast	83425.6500	Year	
2	EZYV03	Asia	Bachelor's	N	Y	44444	2008	West	122996.8600	Year	
3	EZYV04	Asia	Bachelor's	N	N	98	1897	West	83434.0300	Year	
4	EZYV05	Africa	Master's	Y	N	1082	2005	South	149907.3900	Year	

fig 33: Head

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_of_employees	yr_of_estab	region_of_employment	prevailing_wage	unit_of_wage	f
25475	EZYV25476	Asia	Bachelor's	Y	Y	2601	2008	South	77092.57	Year	
25476	EZYV25477	Asia	High School	Y	N	3274	2006	Northeast	279174.79	Year	
25477	EZYV25478	Asia	Master's	Y	N	1121	1910	South	146298.85	Year	
25478	EZYV25479	Asia	Master's	Y	Y	1918	1887	West	86154.77	Year	
25479	EZYV25480	Asia	Bachelor's	Y	N	3195	1960	Midwest	70876.91	Year	

fig 34: Tail

- Getting info of the data

```
Data columns (total 12 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   case_id                25480 non-null  object
 1   continent              25480 non-null  object
 2   education_of_employee  25480 non-null  object
 3   has_job_experience     25480 non-null  object
 4   requires_job_training  25480 non-null  object
 5   no_of_employees        25480 non-null  int64
 6   yr_of_estab            25480 non-null  int64
 7   region_of_employment   25480 non-null  object
 8   prevailing_wage        25480 non-null  float64
 9   unit_of_wage           25480 non-null  object
 10  full_time_position     25480 non-null  object
 11  case_status            25480 non-null  object
dtypes: float64(1), int64(2), object(9)
```

fig 35: Info

- Checking for duplicated values

```
np.int64(0)
```

fig 36: Duplicated values

- Investigating null values

```
case_id                  0
continent                0
education_of_employee    0
has_job_experience       0
requires_job_training    0
no_of_employees          0
yr_of_estab              0
region_of_employment     0
prevailing_wage          0
unit_of_wage             0
full_time_position       0
case_status              0
dtype: int64
```

fig 37: Null valuesv

- Getting description of the data

	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

fig 38: Description of numerical data

	count	unique	top	freq
case_id	25480	25480	EZYV25480	1
continent	25480	6	Asia	16861
education_of_employee	25480	4	Bachelor's	10234
has_job_experience	25480	2	Y	14802
requires_job_training	25480	2	N	22525
region_of_employment	25480	5	Northeast	7195
unit_of_wage	25480	4	Year	22962
full_time_position	25480	2	Y	22773
case_status	25480	2	Certified	17018

fig 39: Description of Categorical data

- Dropping Cases ID as that feature is not of much use to us. Also, we have object data to category data.

```
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   continent             25480 non-null  category
 1   education_of_employee 25480 non-null  category
 2   has_job_experience    25480 non-null  category
 3   requires_job_training 25480 non-null  category
 4   no_of_employees       25480 non-null  int64
 5   yr_of_estab           25480 non-null  int64
 6   region_of_employment  25480 non-null  category
 7   prevailing_wage       25480 non-null  float64
 8   unit_of_wage          25480 non-null  category
 9   full_time_position    25480 non-null  category
 10  case_status           25480 non-null  category
```

fig 40: Case ID dropped

- We found some negative values in number of employees column, so, we ignored the negative sign and treated them as positive number as employees can't be negative in number.
- Below is the value count of the categorical data:

```
continent
Asia             16861
Europe            3732
North America     3292
South America      852
Africa             551
Oceania            192
Name: count, dtype: int64
.....................................
education_of_employee
Bachelor's     10234
Master's        9634
High School     3420
Doctorate       2192
Name: count, dtype: int64
.....................................
has_job_experience
Y    14802
N    10678
Name: count, dtype: int64
.....................................
requires_job_training
N    22525
Y     2955
Name: count, dtype: int64
.....................................
region_of_employment
Northeast    7195
South        7017
West         6586
Midwest      4307
Island        375
Name: count, dtype: int64
.....................................
unit_of_wage
Year     22962
Hour      2157
Week       272
Month       89
Name: count, dtype: int64
.....................................
full_time_position
Y    22773
N     2707
Name: count, dtype: int64
.....................................
case_status
Certified    17018
Denied        8462
Name: count, dtype: int64
.....................................
```

fig 41: Categorical data value count

Now we have sufficient idea about our data and we can proceed towards further analysis.

Univariate Analysis

In univariate analysis we analyse single variable individually.

- Number of Employees:

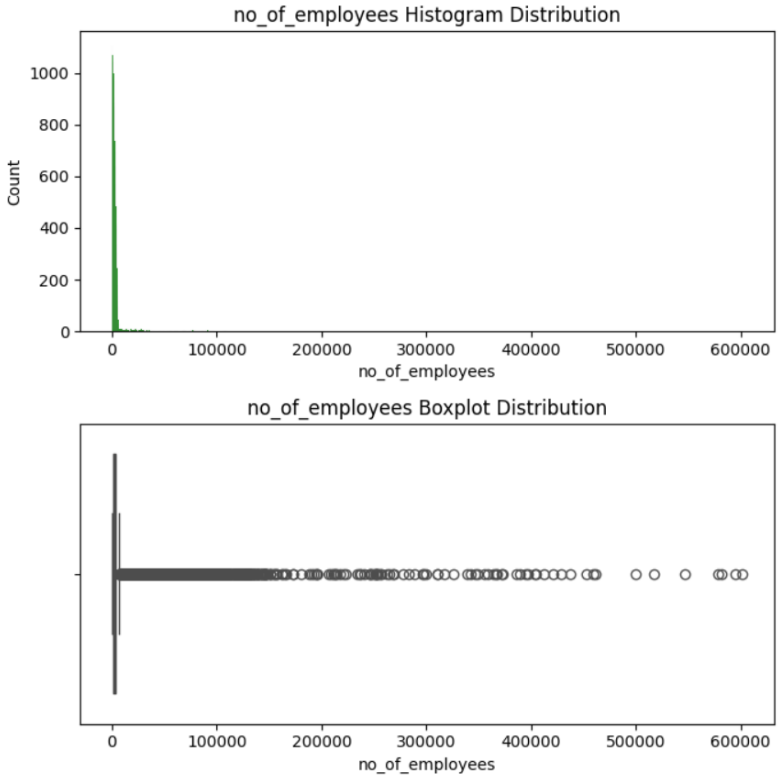

fig 42: Histogram and Boxplot for Number of employees

- Prevailing wages:

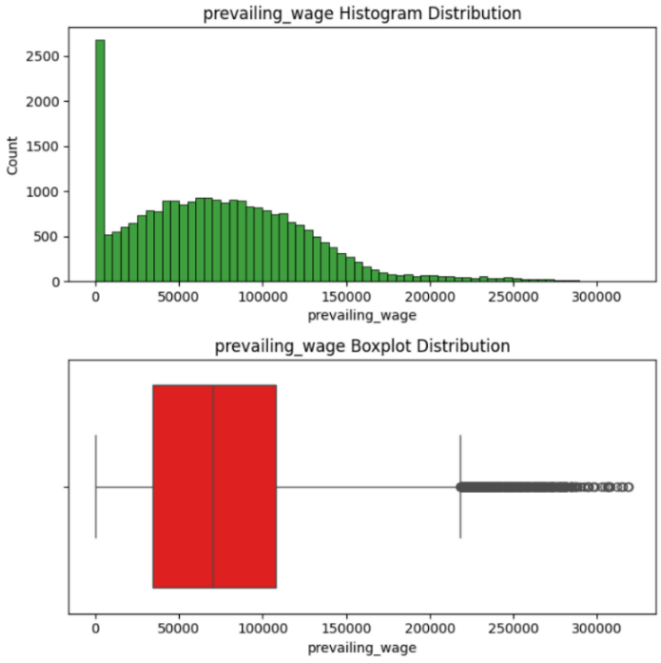

fig 43: Histogram and boxplot for prevailing wages

- Year of establishment of the companies:
  In the data we've companies ranging from the year which are established in 1800 to the year of 2016.

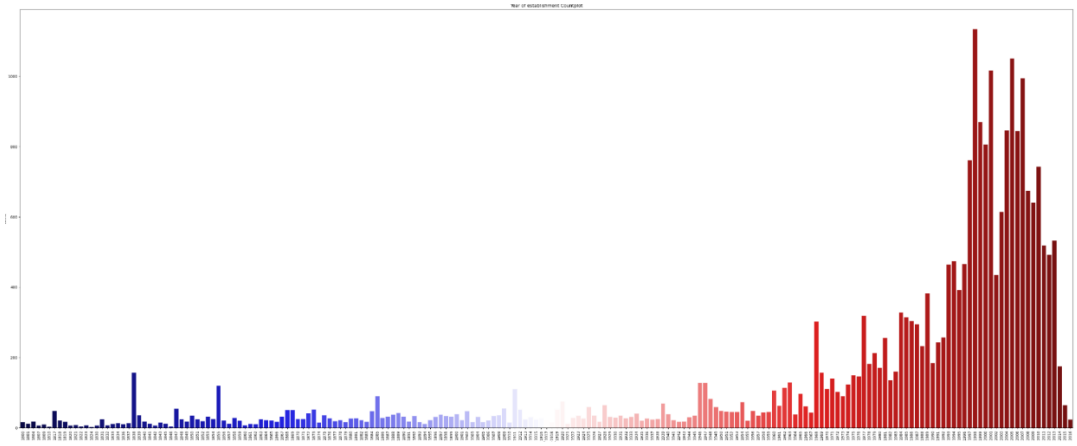

fig 44: Count plot of year of establishment of companies

- Continent:

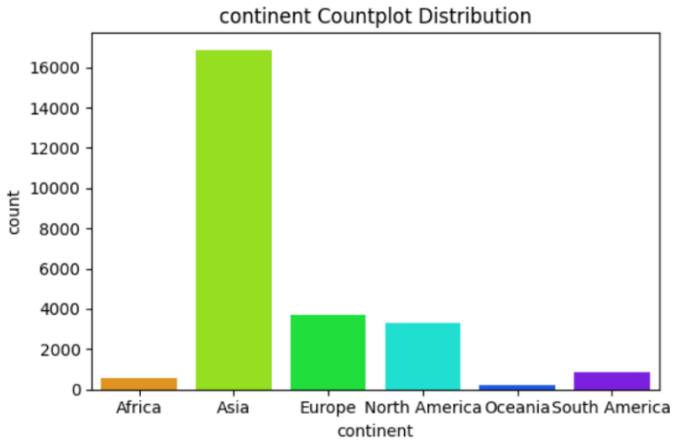

fig 45: Count plot for Continent

- Education of employee:

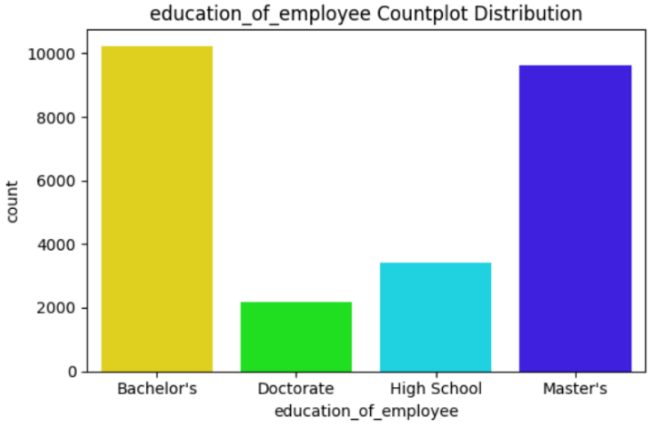

fig 46: Count plot for Education of employee

- Job experience:

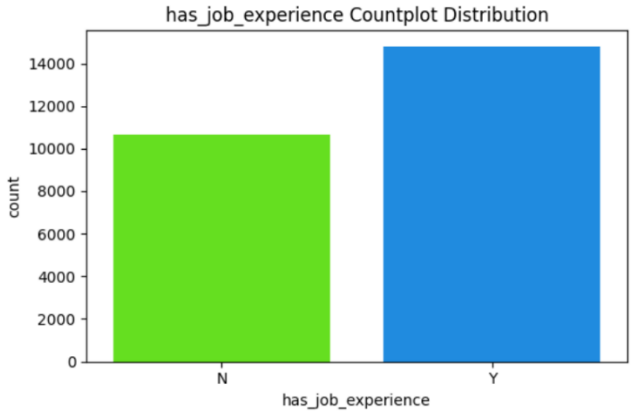

fig 47: Count Plot for Job experience

- Job training requirement:

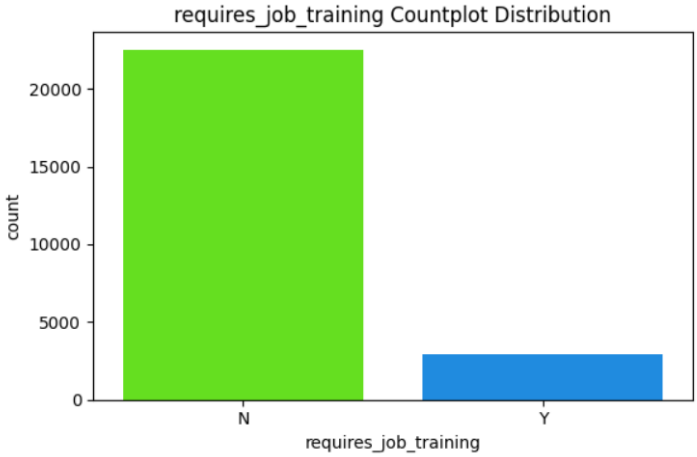

fig 48: Count Plot of Job training requirement

- Region of employment:

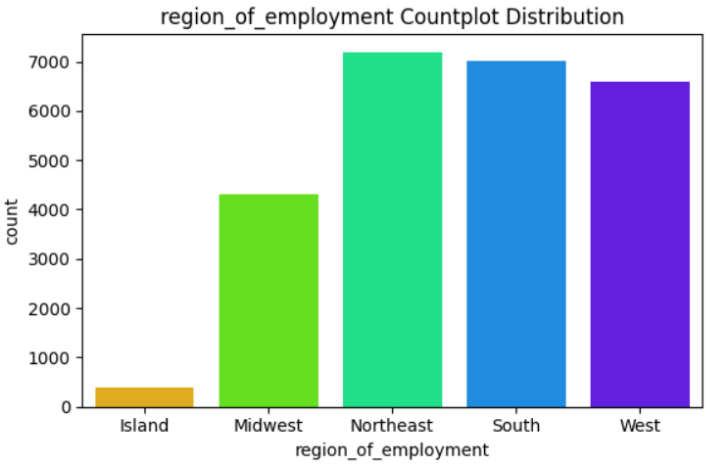

fig 49: Count Plot of Region of employment

- Unit of wage:

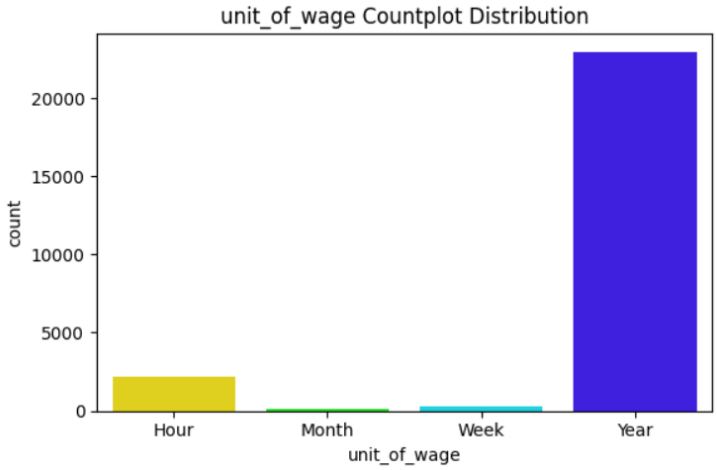

fig 50: Count Plot of unit of wage

- Full time position:

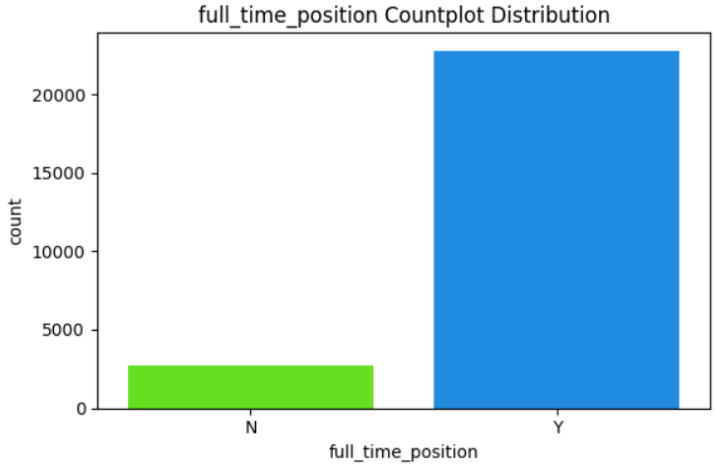

fig 51: Count Plot of Full-time position

- Case Status:

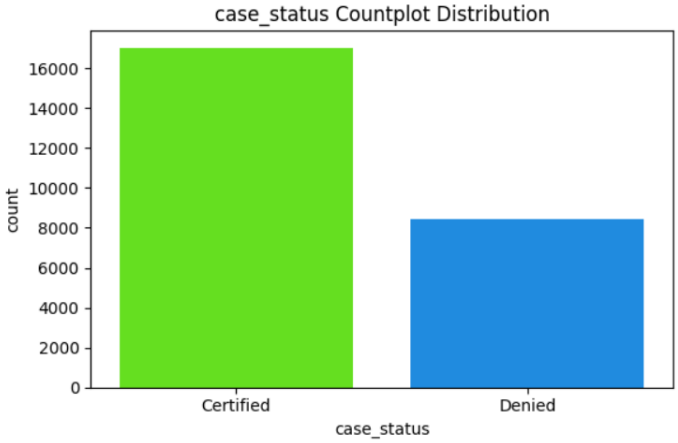

fig 52: Count Plot of Case Status

Bivariate and Multivariate Analysis

Here, we analyse the relationship between 2 or more variables.

- Continent wrt case status:

continent	case_status	
Africa	Certified	72.050817
	Denied	27.949183
Asia	Certified	65.310480
	Denied	34.689520
Europe	Certified	79.233655
	Denied	20.766345
North America	Certified	61.877278
	Denied	38.122722
Oceania	Certified	63.541667
	Denied	36.458333
South America	Certified	57.863850
	Denied	42.136150

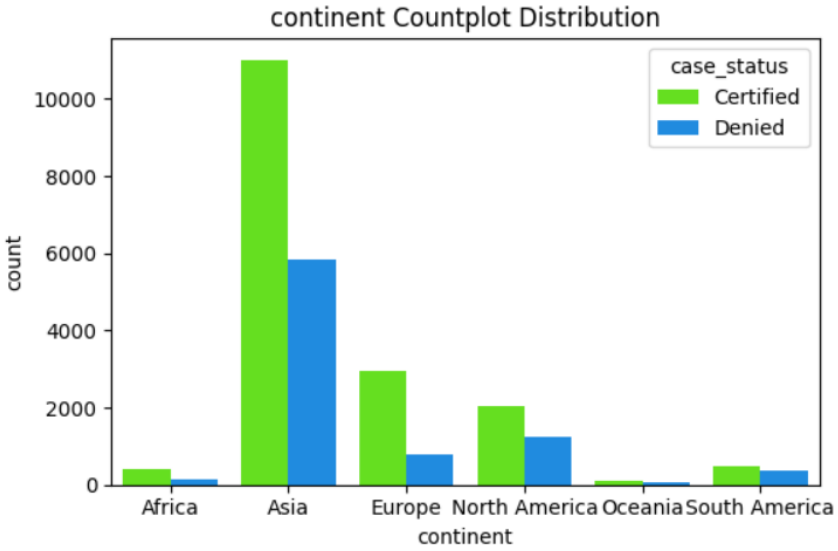

fig 53: Continent wrt case status

- Education of employee wrt case status:

education_of_employee	case_status	
Bachelor's	Certified	62.214188
	Denied	37.785812
Doctorate	Certified	87.226277
	Denied	12.773723
High School	Denied	65.964912
	Certified	34.035088
Master's	Certified	78.627777
	Denied	21.372223

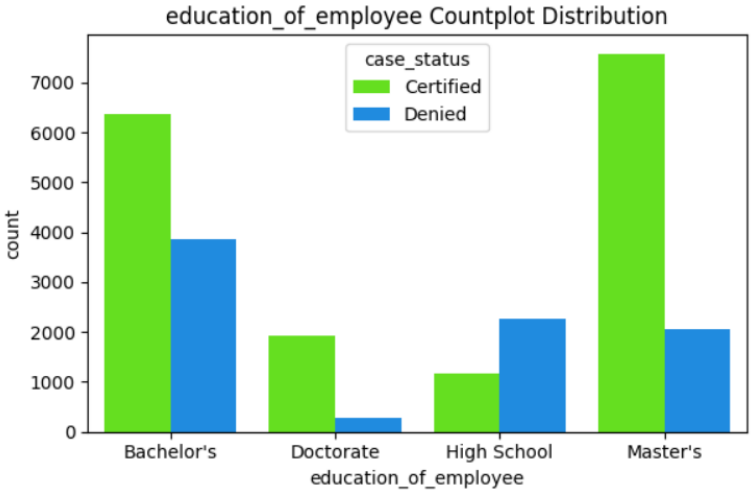

fig 54: Education of employee wrt case status

- Job experience wrt case status:

has_job_experience	case_status	
N	Certified	56.134108
	Denied	43.865892
Y	Certified	74.476422
	Denied	25.523578

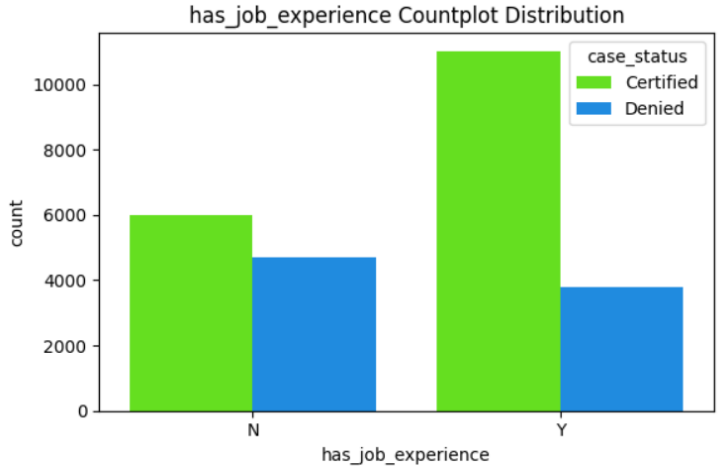

fig 55: Job experience wrt case status

- Requirement of job training wrt case status:

requires_job_training	case_status	
N	Certified	66.645949
	Denied	33.354051
Y	Certified	67.884941
	Denied	32.115059

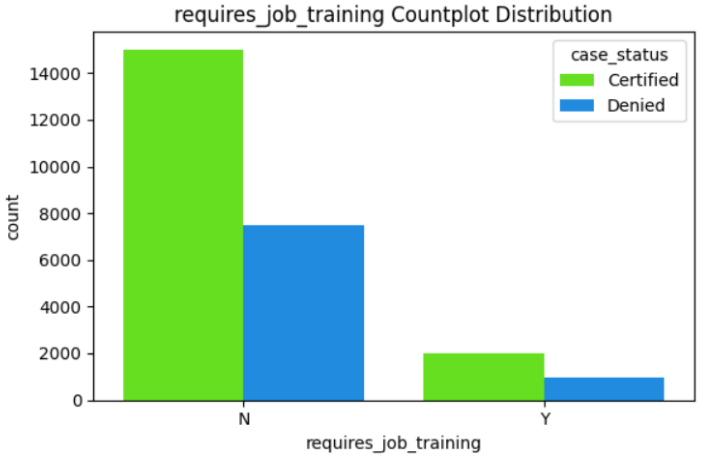

fig 56: Requirement of job training wrt case status

- Region of employment wrt case status:

region_of_employment	case_status	
Island	Certified	60.266667
	Denied	39.733333
Midwest	Certified	75.528210
	Denied	24.471790
Northeast	Certified	62.904795
	Denied	37.095205
South	Certified	70.015676
	Denied	29.984324
West	Certified	62.253265
	Denied	37.746735

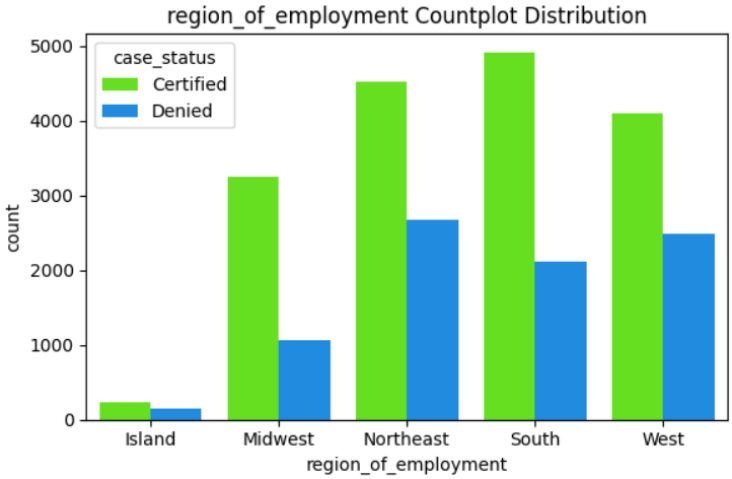

fig 57: Region of employment wrt case status

- Unit of wage wrt case status:

unit_of_wage	case_status	
Hour	Denied	65.368567
	Certified	34.631433
Month	Certified	61.797753
	Denied	38.202247
Week	Certified	62.132353
	Denied	37.867647
Year	Certified	69.885027
	Denied	30.114973

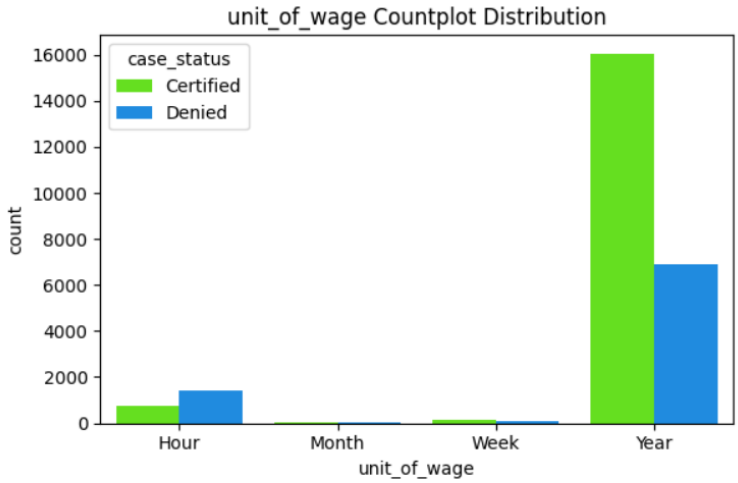

fig 58: Unit of wage wrt case status

- Full time position wrt case status:

full_time_position	case_status	
N	Certified	68.526044
	Denied	31.473956
Y	Certified	66.583235
	Denied	33.416765

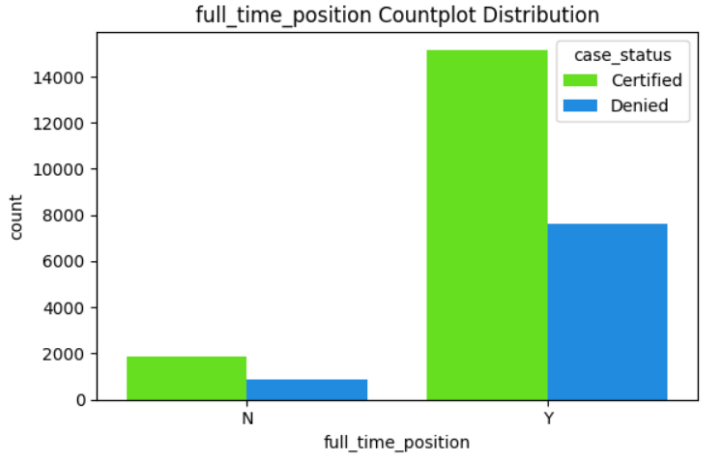

fig 59: Full time position wrt case status

- Unit of wage vs prevailing wage:

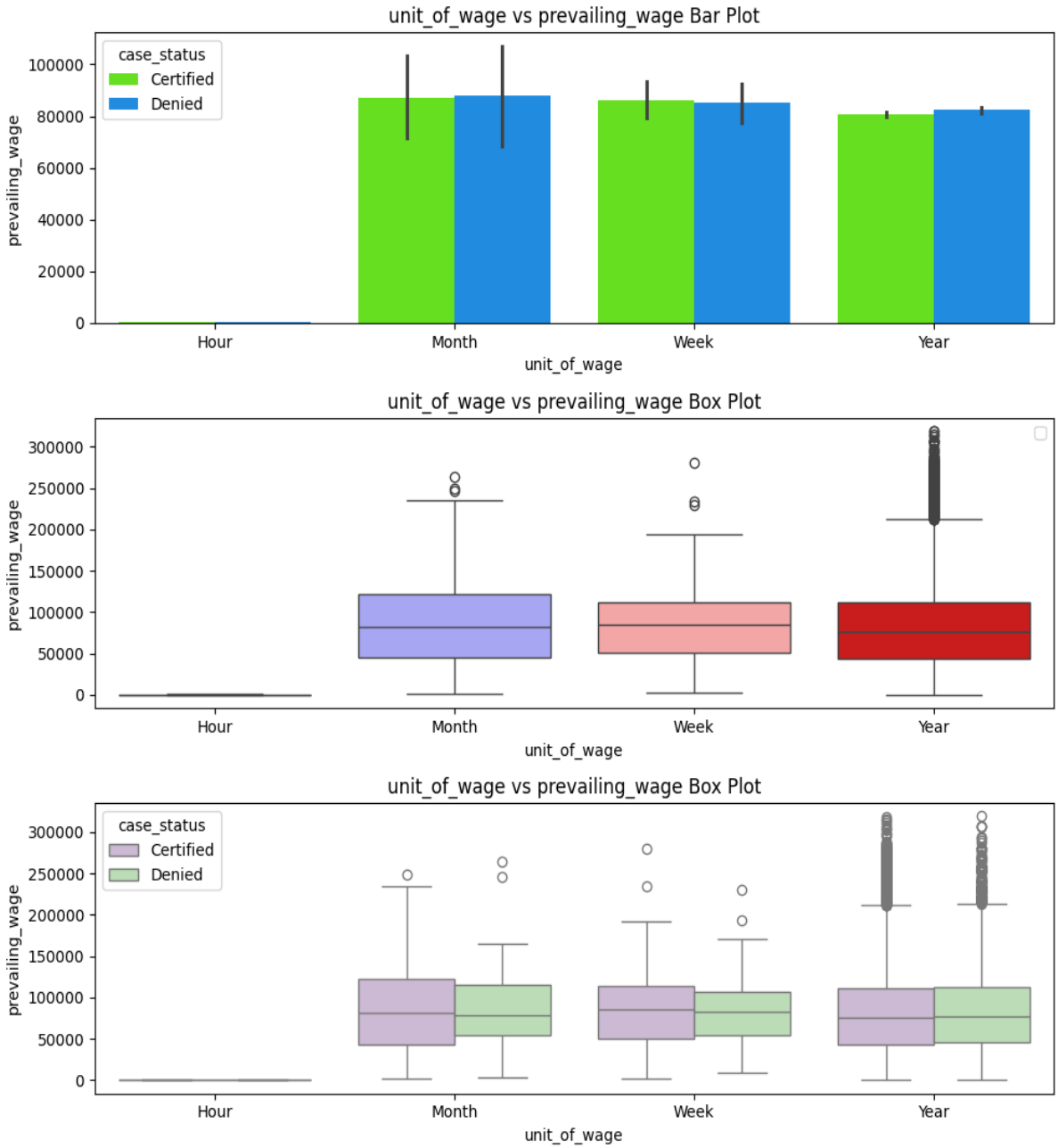

fig 60: Unit of wage vs prevailing wage

- Continent vs Prevailing wage

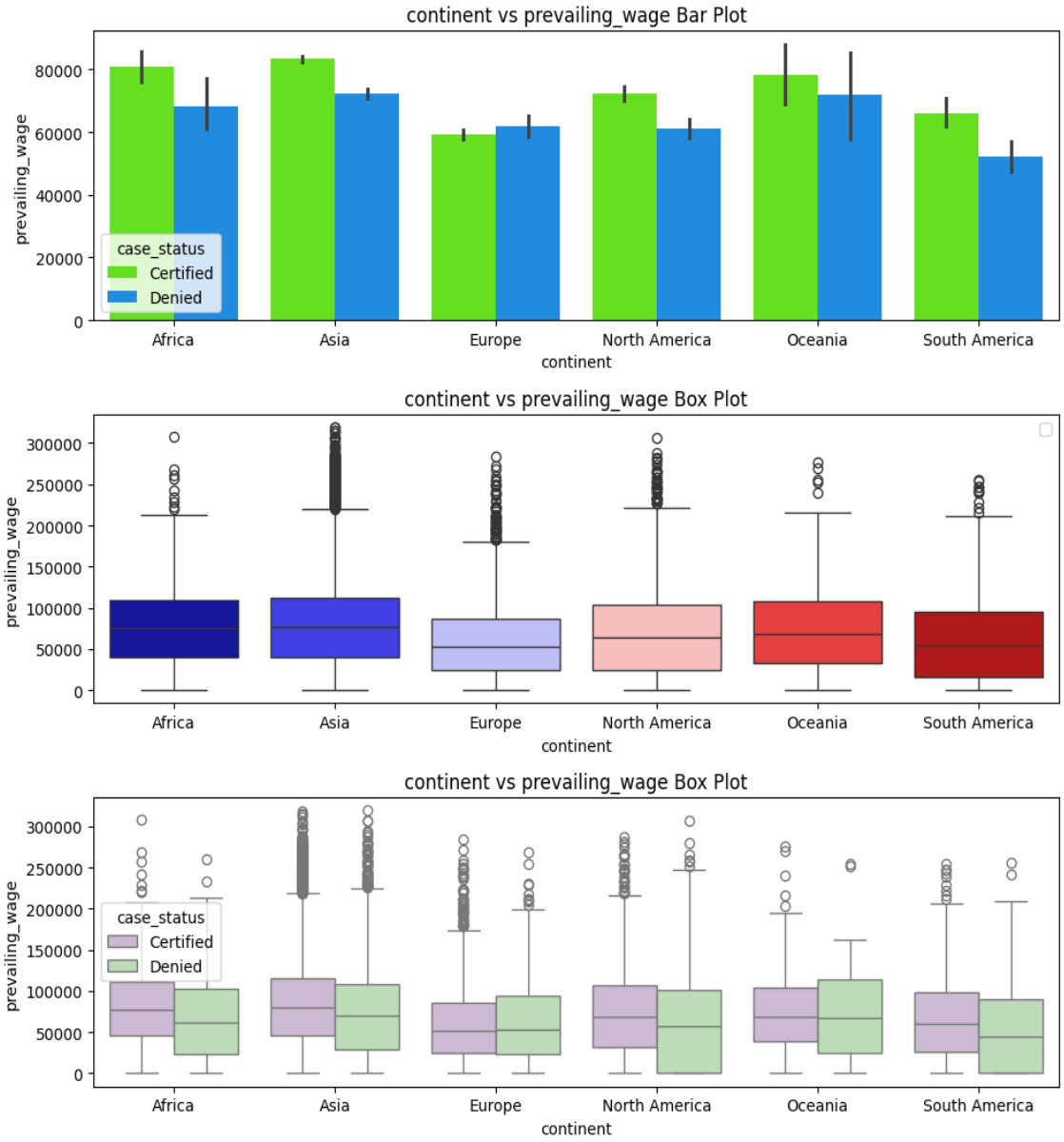

fig 61: Continent vs prevailing wage

- Education of employee vs Prevailing wage

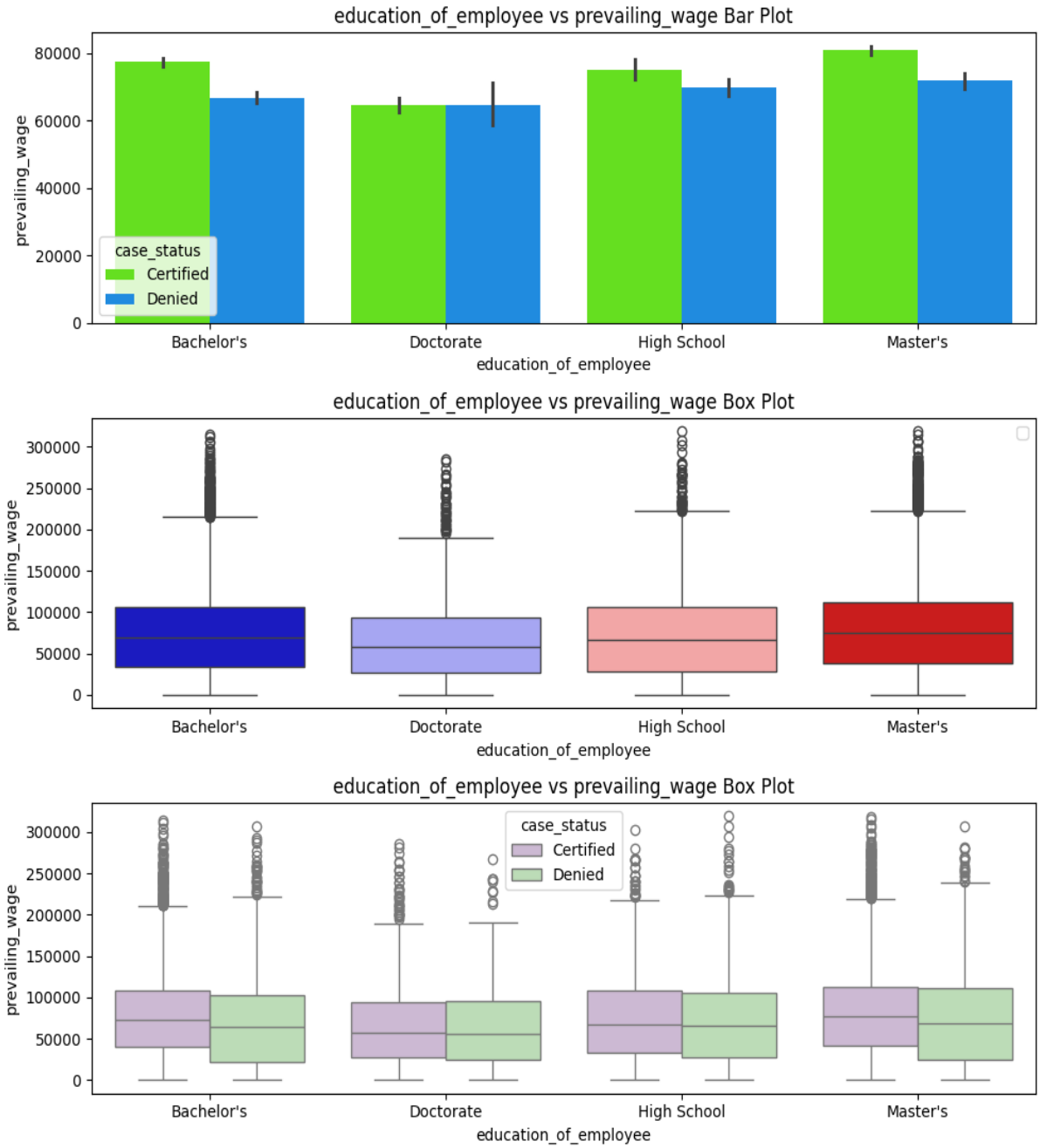

fig 62: Education of employee vs prevailing wage

- Region of employment vs Prevailing wage

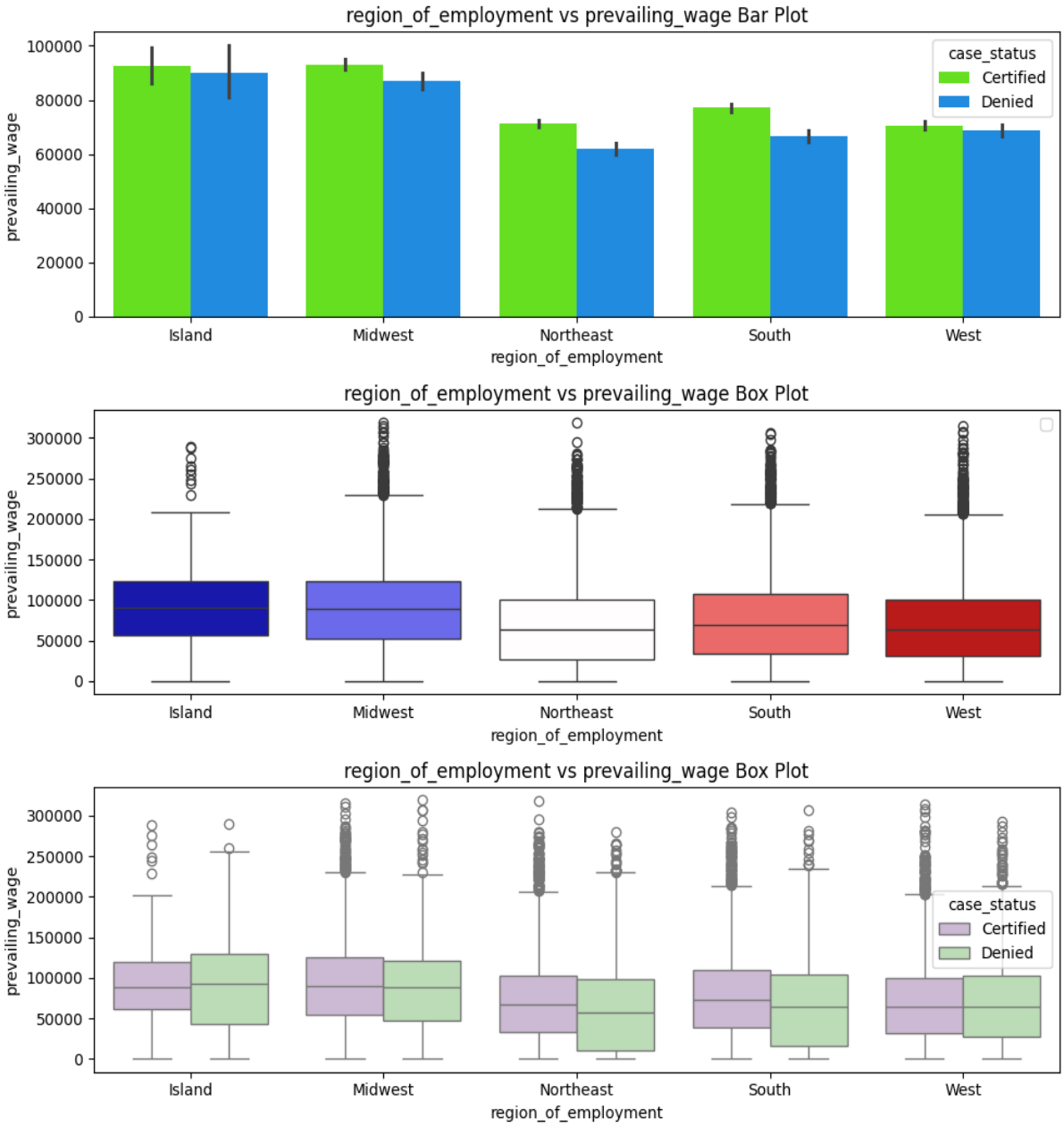

fig 63: Region of Employment vs prevailing wage

Heatmap and Pair Plot

For better understanding of the numerical vs numerical data, we created Heatmap.

In the heatmap we can see the corelation between different numerical data. The heatmap shows almost negligible corelation among the features of the data, which is suitable to us.

fig 64: Heatmap

Below, is the pair plot created among all the numerical data.

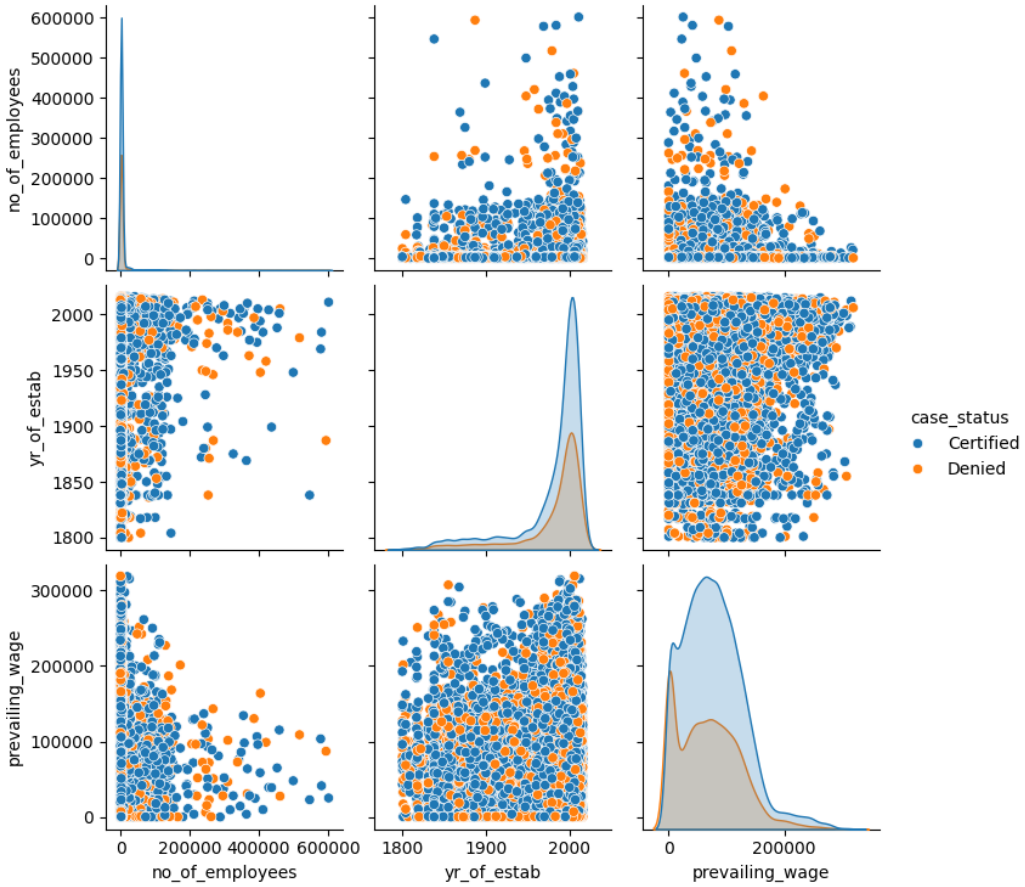

fig 65: Pairplot

KEY INSIGHTS AS PER EDA

- South region has the highest number of jobs.
- Island region has the lowest number of employment but at the same time offers highest wages.
- Large number of companies have established in the recent years.
- Asians are the highest in number to apply for the visa.
- Europe has the lower rejection rate of 20.7%, whereas South America has the highest rejection rate of 42.13%.
- People with unit of wages as hour have been rejected the most i.e. 65.36% and those with unit of wages as year have been accepted the most i.e. 69.88%.

OUTLIERS TREATMENT

We have detected outliers in the data set, but those outliers seem to be genuine, therefore, we are not going to treat the outliers. We will keep the outliers in the data and will move ahead with the same.

ONE-HOT ENCODING AND FEATURE ENGINEERING

We need to apply One-Hot Encoding in the categorical data, i.e. for "Continent, education of employee, job experience, job training requirement, region of employment, unit of wage and full-time position. This process converts each category into a new binary column. For example, the job experience becomes multiple new columns, like job experience Y and job experience N., with a value of 1 if the data point belongs to that category and 0 otherwise.

	0	1	2	3	4
no_of_employees	14513.0000	2412.00	44444.00	98.00	1082.00
yr_of_estab	2007.0000	2002.00	2008.00	1897.00	2005.00
prevailing_wage	592.2029	83425.65	122996.86	83434.03	149907.39
continent_Asia	1.0000	1.00	1.00	1.00	0.00
continent_Europe	0.0000	0.00	0.00	0.00	0.00
continent_North America	0.0000	0.00	0.00	0.00	0.00
continent_Oceania	0.0000	0.00	0.00	0.00	0.00
continent_South America	0.0000	0.00	0.00	0.00	0.00
education_of_employee_Doctorate	0.0000	0.00	0.00	0.00	0.00
education_of_employee_High School	1.0000	0.00	0.00	0.00	0.00
education_of_employee_Master's	0.0000	1.00	0.00	0.00	1.00
has_job_experience_Y	0.0000	1.00	0.00	0.00	1.00
requires_job_training_Y	0.0000	0.00	1.00	0.00	0.00
region_of_employment_Midwest	0.0000	0.00	0.00	0.00	0.00
region_of_employment_Northeast	0.0000	1.00	0.00	0.00	0.00
region_of_employment_South	0.0000	0.00	0.00	0.00	1.00
region_of_employment_West	1.0000	0.00	1.00	1.00	0.00
unit_of_wage_Month	0.0000	0.00	0.00	0.00	0.00
unit_of_wage_Week	0.0000	0.00	0.00	0.00	0.00
unit_of_wage_Year	0.0000	1.00	1.00	1.00	1.00
full_time_position_Y	1.0000	1.00	1.00	1.00	1.00

fig 34: One Hot Encoding

Target value is a categorical data; therefore, it has to be converted to numerical data, such that, visa certified will be called as 1 and visa denied will be called as 0.

	case_status
0	0.0
1	1.0
2	0.0
3	0.0
4	1.0

fig 35: Target value converted to numerical data

SPLITTING DATA

Data has been split into 80:20 ratio, where 80% data is used for training and 20% data will be used for testing.

```
Shape of total train data: (20384, 21)
Shape of test data: (5096, 21)
Percentage of classes in total training data: case_status
1.0    0.667877
0.0    0.332123
Name: proportion, dtype: float64
Percentage of classes in testing data: case_status
1.0    0.667975
0.0    0.332025
```

fig 36: Train Test Data Split

This training data is further split into 2 parts train data and validation data. Validation data is 25% of the training data.

```
Shape of train data: (15288, 21)
Shape of validation data: (5096, 21)
Percentage of classes in training data: case_status
1.0    0.667909
0.0    0.332091
Name: proportion, dtype: float64
Percentage of classes in validation data: case_status
1.0    0.667779
0.0    0.332221
```

fig 37: Train Validation Data Split

We'll be using test data **at the last** after deciding the final model. We'll be using validation data **after hyper tuning the models**, and compare the results of the training data with validation data. This is being done to avoid the **overfitting to validation set.**

BUILDING MODEL- ORIGINAL DATA

We'll be focusing on F1 score as both visa certified and visa denied, both the cases are important to use

Decision Tree Classification: This is an overfit model

```
DecisionTreeClassifier
DecisionTreeClassifier(class_weight='balanced', random_state=42)
```

fig 38: Decision Tree

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	5077
1.0	1.00	1.00	1.00	10211
accuracy			1.00	15288
macro avg	1.00	1.00	1.00	15288
weighted avg	1.00	1.00	1.00	15288

fig 39: Classification Report Decision Tree

	Accuracy	Recall	Precision	F1_score
0	1.0	1.0	1.0	1.0

fig 40: Model Performance Decision Tree

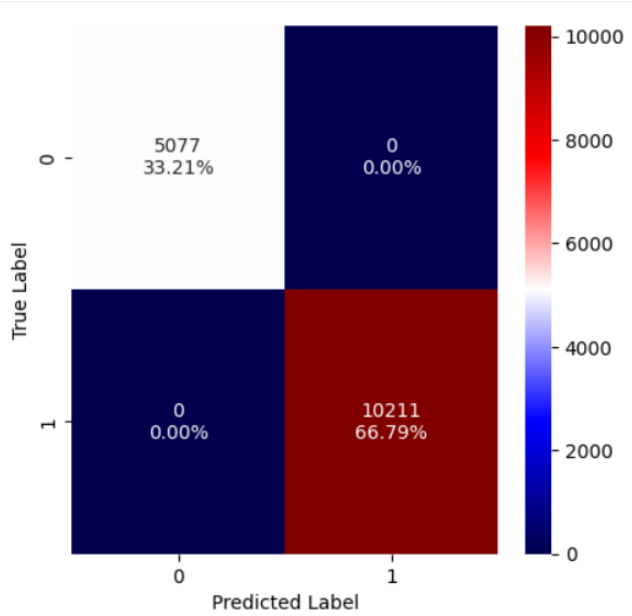

fig 41: Confusion Matrix Decision Tree

Random Forest Classification: This is an overfit model

```
RandomForestClassifier(class_weight='balanced', random_state=42)
```

fig 42: Random Forest

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	5077
1.0	1.00	1.00	1.00	10211
accuracy			1.00	15288
macro avg	1.00	1.00	1.00	15288
weighted avg	1.00	1.00	1.00	15288

fig 43: Classification Report Random Forest

	Accuracy	Recall	Precision	F1_score
0	0.999935	1.0	0.999902	0.999951

fig 44: Model Performance Random Forest

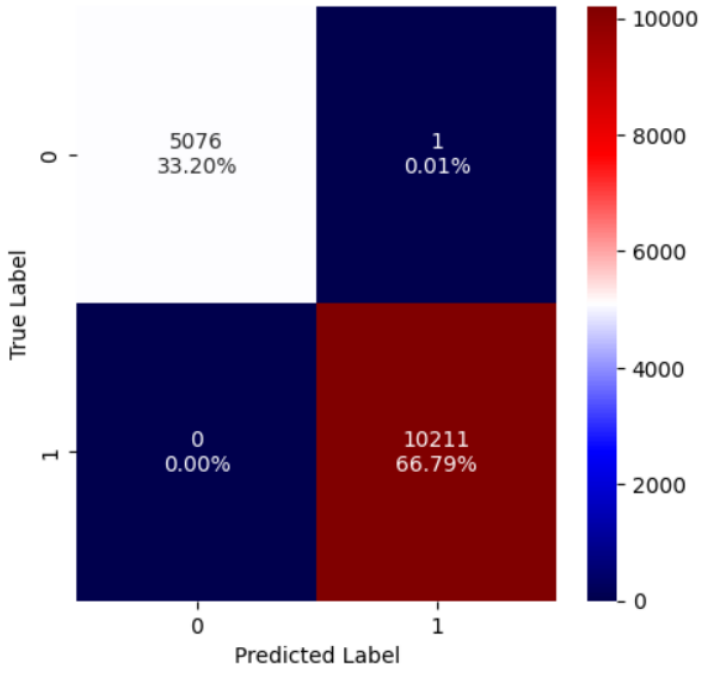

fig 45: Confusion Matrix Random Forest

Bagging: This is an overfit model

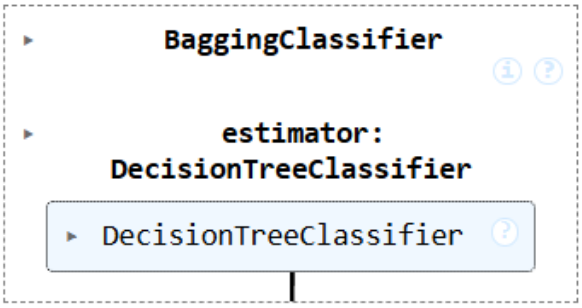

fig 46: Bagging Classification

	precision	recall	f1-score	support
0.0	0.97	0.98	0.98	5077
1.0	0.99	0.98	0.99	10211
accuracy			0.98	15288
macro avg	0.98	0.98	0.98	15288
weighted avg	0.98	0.98	0.98	15288

fig 47: Classification Report Bagging

	Accuracy	Recall	Precision	F1_score
0	0.984236	0.98482	0.99152	0.988159

fig 48: Model Performance Bagging

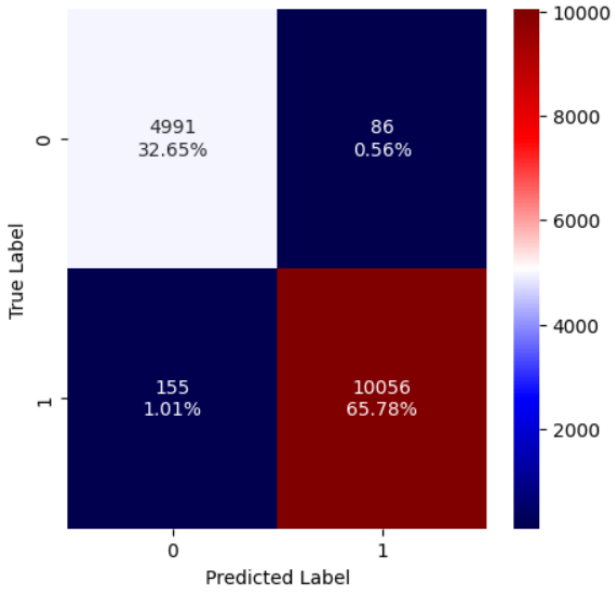

fig 49: Confusion Matrix Bagging

Ada Boosting: This looks like a sensible model

```
AdaBoostClassifier
AdaBoostClassifier(random_state=42)
```

fig 50: Ada Boost Classifier

	precision	recall	f1-score	support
0.0	0.66	0.45	0.54	5077
1.0	0.77	0.88	0.82	10211
accuracy			0.74	15288
macro avg	0.71	0.67	0.68	15288
weighted avg	0.73	0.74	0.73	15288

fig 51: Classification Report Ada Boosting

	Accuracy	Recall	Precision	F1_score
0	0.741497	0.884145	0.765279	0.820429

fig 52: Model Performance Ada Boosting

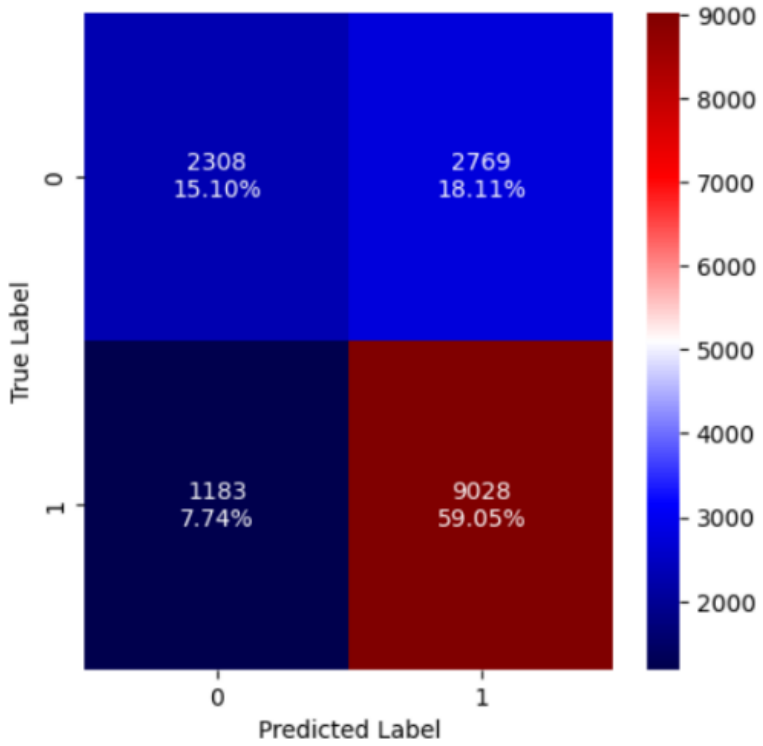

fig 53: Confusion Matrix Ada Boosting

XG Boosting: This looks like a sensible model

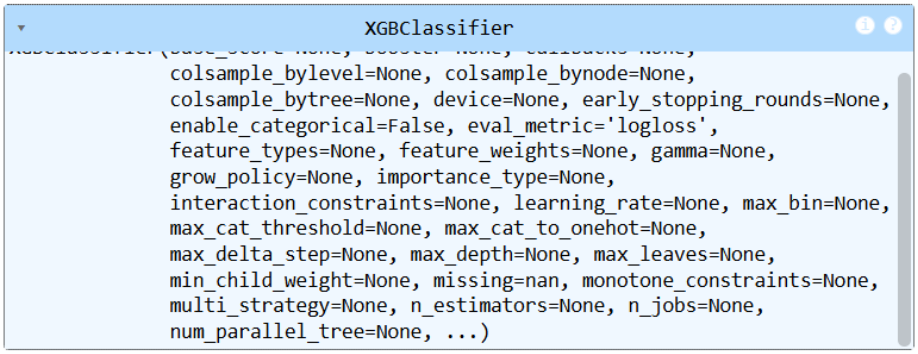

fig 54: XG Boosting Classification

	precision	recall	f1-score	support
0.0	0.85	0.69	0.76	5077
1.0	0.86	0.94	0.90	10211
accuracy			0.85	15288
macro avg	0.85	0.81	0.83	15288
weighted avg	0.85	0.85	0.85	15288

fig 55: Classification Report XG Boosting

	Accuracy	Recall	Precision	F1_score
0	0.854199	0.937812	0.857296	0.895749

fig 56: Model Performance XG Boosting

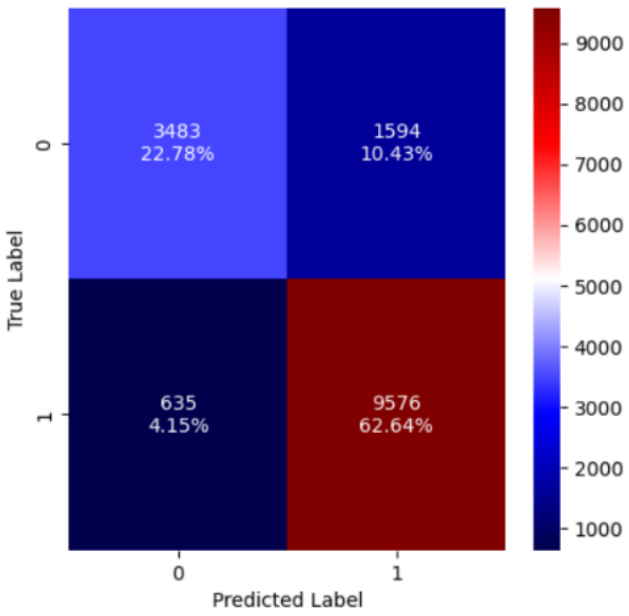

fig 57: Confusion Matrix XG Boosting

BUILDING MODEL- OVERSAMPLED DATA

One of the methods to eliminate or reduce the biasness in the data, which is because of the imbalance present in the data, i.e., number of visa certified are more than number of visa denied, so, this creates a possibility that our model will be biased towards number of visa certified. There are 2 methods to treat this issue; (a) Over-sampling the data (b) Under-sampling data.

Now we'll be doing over-sampling of the data.

```
Before Oversampling, counts of label 'Certified': 10211
Before Oversampling, counts of label 'Denied': 5077

After Oversampling, counts of label 'Certified': 10211
After Oversampling, counts of label 'Denied': 10211

After Oversampling, the shape of train_X: (20422, 21)
After Oversampling, the shape of train_y: (20422,)
```

fig 58: Oversampling Data

Decision Tree Oversampled: This is an overfit model

```
DecisionTreeClassifier
DecisionTreeClassifier(class_weight='balanced', random_state=42)
```

fig 59: Decision Tree Oversampled

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	10211
1.0	1.00	1.00	1.00	10211
accuracy			1.00	20422
macro avg	1.00	1.00	1.00	20422
weighted avg	1.00	1.00	1.00	20422

fig 60: Classification report Decision Tree Oversampled

	Accuracy	Recall	Precision	F1_score
0	1.0	1.0	1.0	1.0

fig 61: Model Performance Decision Tree Oversampled

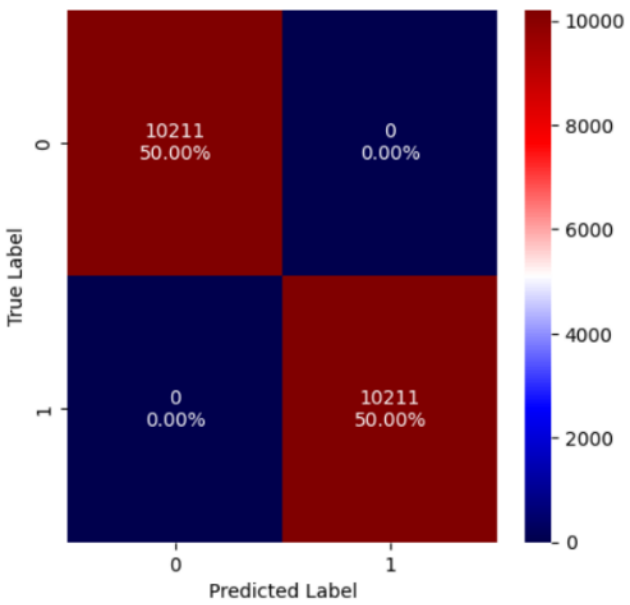

fig 62: Confusion Matrix Decision Tree Oversampled

Random Forest Oversampled: This is an overfit model

```
RandomForestClassifier(class_weight='balanced', random_state=42)
```

fig 63: Random Forest Oversampled

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	10211
1.0	1.00	1.00	1.00	10211
accuracy			1.00	20422
macro avg	1.00	1.00	1.00	20422
weighted avg	1.00	1.00	1.00	20422

fig 64: Classification Report Random Forest Oversampled

	Accuracy	Recall	Precision	F1_score
0	1.0	1.0	1.0	1.0

fig 65: Model Performance Random Forest Oversampled

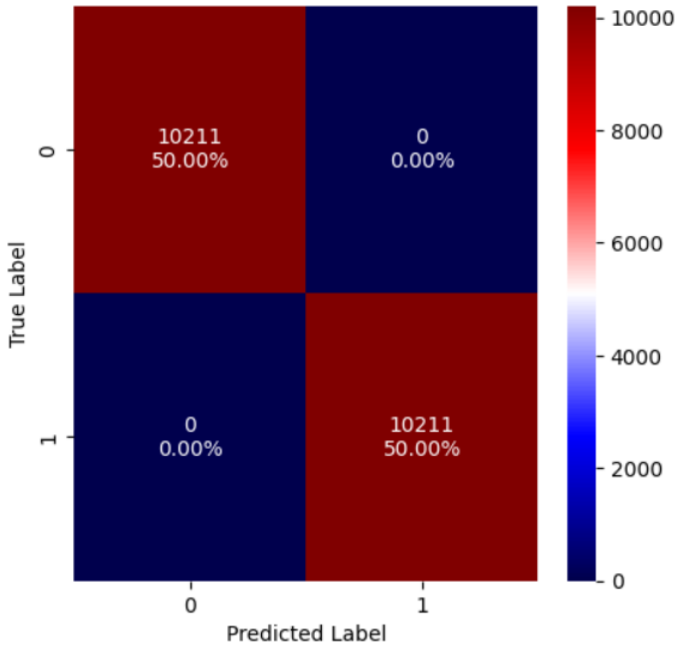

fig 66: Confusion Matrix Random Forest Oversampled

Bagging Oversampled: This is an overfit model

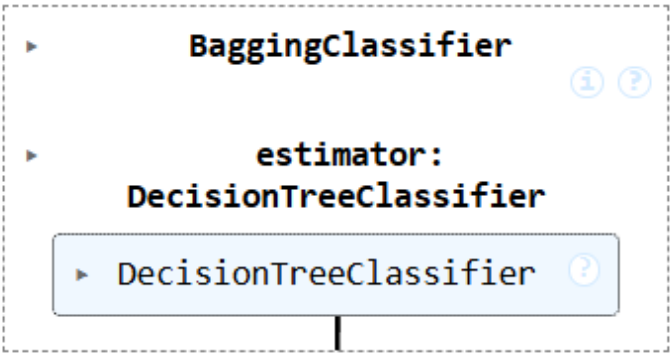

fig 67: Bagging Oversampled

	precision	recall	f1-score	support
0.0	0.98	0.99	0.99	10211
1.0	0.99	0.98	0.99	10211
accuracy			0.99	20422
macro avg	0.99	0.99	0.99	20422
weighted avg	0.99	0.99	0.99	20422

fig 68: Classification Report Bagging Oversampled

	Accuracy	Recall	Precision	F1_score
0	0.988248	0.984526	0.991909	0.988204

fig 69: Model Performance Bagging Oversampled

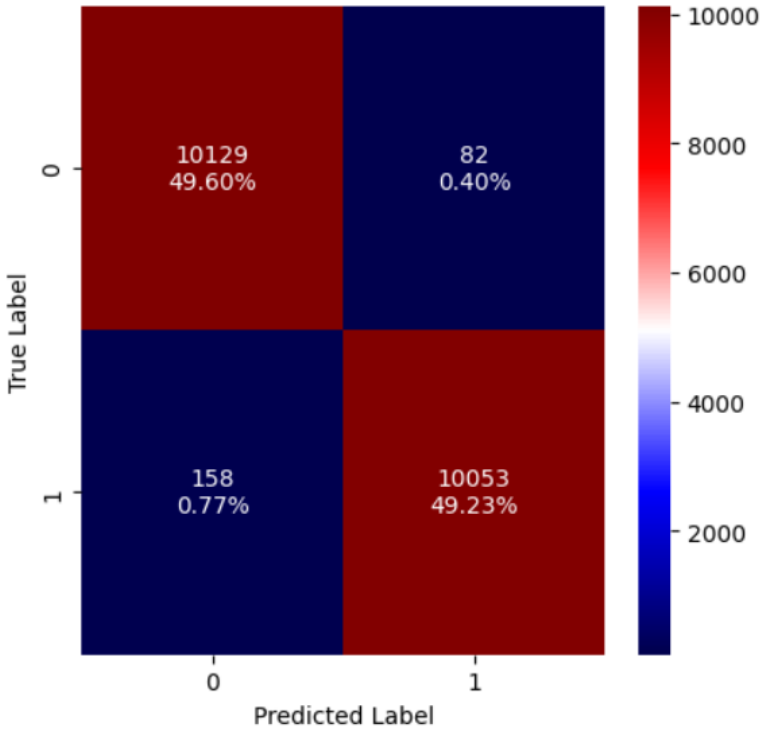

fig 70: Confusion Matrix Bagging Oversampled

Ada Boost Oversampled: This looks like a sensible model

```
AdaBoostClassifier
AdaBoostClassifier(random_state=42)
```

fig 71: Ada Boost Oversampled

	precision	recall	f1-score	support
0.0	0.82	0.76	0.79	10211
1.0	0.78	0.83	0.80	10211
accuracy			0.80	20422
macro avg	0.80	0.80	0.80	20422
weighted avg	0.80	0.80	0.80	20422

fig 72: Classification Report Ada Boost Oversampled

	Accuracy	Recall	Precision	F1_score
0	0.796445	0.83175	0.776894	0.803386

fig 73: Model Performance Ada Boost Oversampled

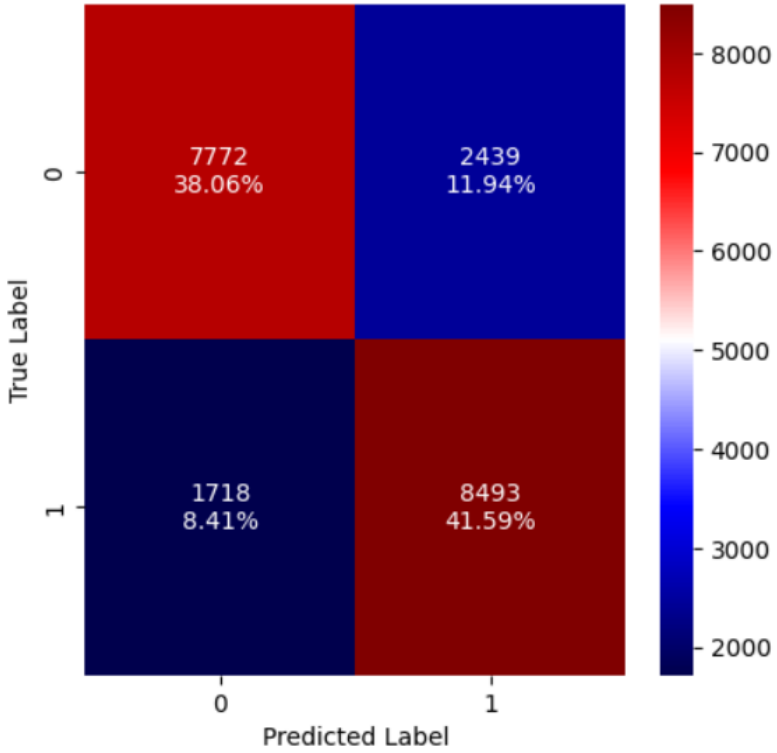

fig 74: Confusion Matrix Ada Boost Oversampled

XG Boost Oversampled: This looks like a sensible model

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              feature_weights=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None,
              n_jobs=None, num_parallel_tree=None, ...)
```

fig 75: XG Boosting Oversampled

	precision	recall	f1-score	support
0.0	0.92	0.84	0.88	10211
1.0	0.85	0.93	0.89	10211
accuracy			0.88	20422
macro avg	0.89	0.88	0.88	20422
weighted avg	0.89	0.88	0.88	20422

fig 76: Classification Report XG Boosting Oversampled

	Accuracy	Recall	Precision	F1_score
0	0.88292	0.925375	0.852952	0.887688

fig 77: Model Performance XG Boosting Oversampled

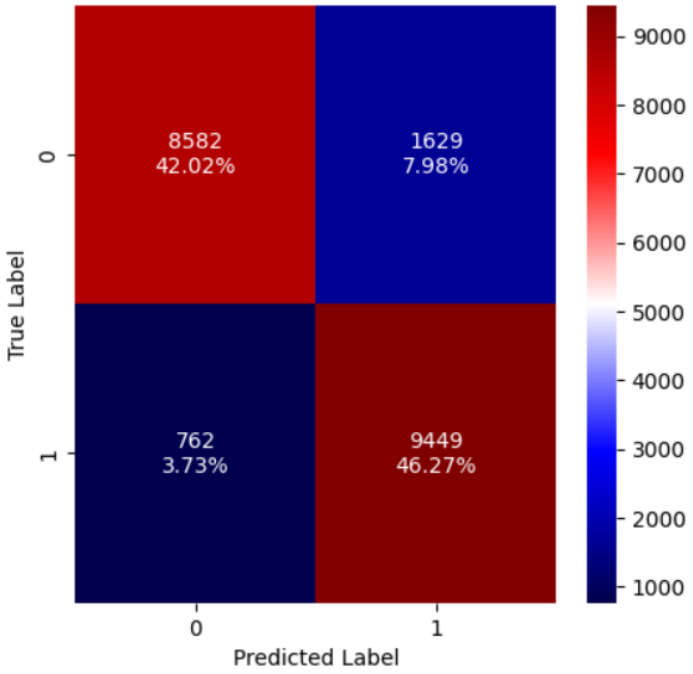

fig 78: Confusion Matrix XG Boosting Oversampled

BUILDING MODEL- UNDER SAMPLED DATA

We did the oversampling of the data trained and evaluated their model performances, now, similarly, we'll be doing the under-sampling of the data, that would be another way of reducing the biasness in our models.

```
Before Undersampling, counts of label 'Certified': 10211
Before Undersampling, counts of label 'Denied': 5077

After Undersampling, counts of label 'Certified': 5077
After Undersampling, counts of label 'Denied': 5077

After Undersampling, the shape of train_X: (10154, 21)
After Undersampling, the shape of train_y: (10154,)
```

fig 79: Under-sampling of Data

Decision Tree Under Sampled: This is an overfit model

```
DecisionTreeClassifier
DecisionTreeClassifier(class_weight='balanced', random_state=42)
```

fig 80: Decision Tree Under Sampled

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	5077
1.0	1.00	1.00	1.00	5077
accuracy			1.00	10154
macro avg	1.00	1.00	1.00	10154
weighted avg	1.00	1.00	1.00	10154

fig 81: Classification Report Decision Tree Under Sampled

	Accuracy	Recall	Precision	F1_score
0	1.0	1.0	1.0	1.0

fig 82: Model Performance Decision Tree Under Sampled

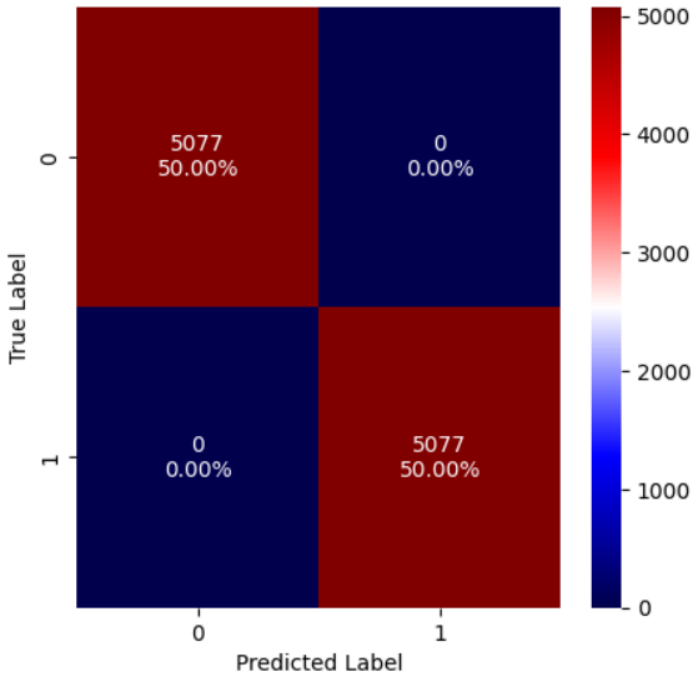

fig 83: Confusion Matrix Decision Tree Under Sampled

Random Forest Under Sampled: This is an overfit model

```
RandomForestClassifier
RandomForestClassifier(class_weight='balanced', random_state=42)
```

fig 84: Random Forest Under Sampled

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	5077
1.0	1.00	1.00	1.00	5077
accuracy			1.00	10154
macro avg	1.00	1.00	1.00	10154
weighted avg	1.00	1.00	1.00	10154

fig 85: Classification Report Random Forest Under Sampled

	Accuracy	Recall	Precision	F1_score
0	1.0	1.0	1.0	1.0

fig 86: Model Performance Random Forest Under Sampled

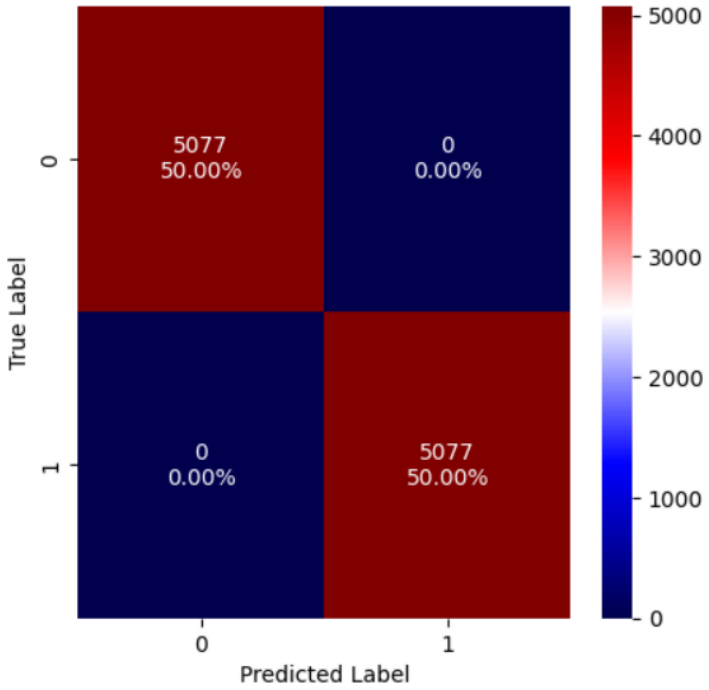

fig 87: Confusion Matrix Random Forest Under Sampled

Bagging Under Sampled: This is an overfit model

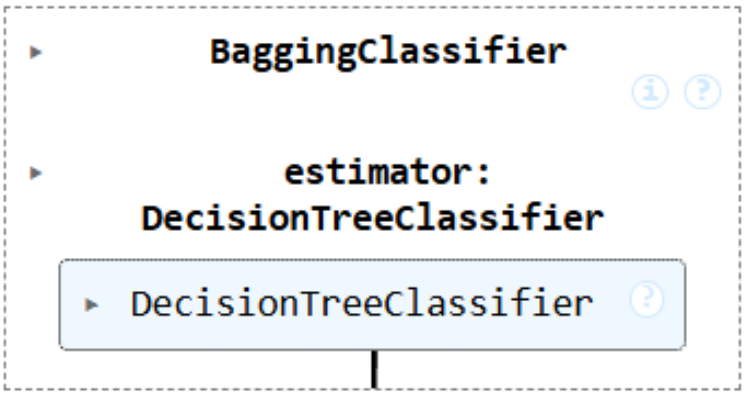

fig 88: Bagging Under Sampled

	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	5077
1.0	0.99	0.97	0.98	5077
accuracy			0.98	10154
macro avg	0.98	0.98	0.98	10154
weighted avg	0.98	0.98	0.98	10154

fig 89: Classification Report Bagging Under Sampled

	Accuracy	Recall	Precision	F1_score
0	0.979515	0.967697	0.991124	0.97927

fig 90: Model Performance Bagging Under Sampled

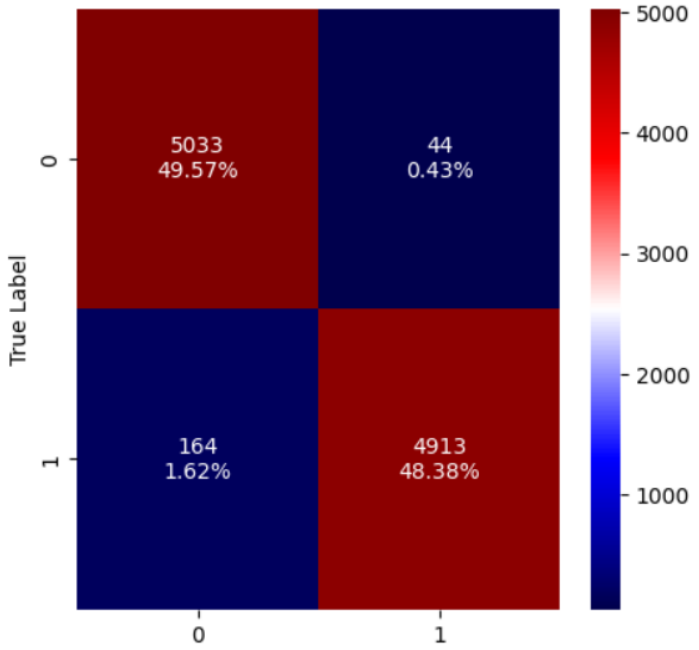

fig 91: Confusion Matix Bagging Under-sampled

Ada Boost Under Sampled: This looks like a sensible model

```
AdaBoostClassifier
AdaBoostClassifier(random_state=42)
```

fig 92: Ada Boost Under Sampled

	precision	recall	f1-score	support
0.0	0.71	0.64	0.68	5077
1.0	0.67	0.74	0.71	5077
accuracy			0.69	10154
macro avg	0.69	0.69	0.69	10154
weighted avg	0.69	0.69	0.69	10154

fig 93: Classification Report Ada Boost Under Sampled

	Accuracy	Recall	Precision	F1_score
0	0.691058	0.739807	0.674085	0.705418

fig 94: Model Performance Ada Boost Under Sampled

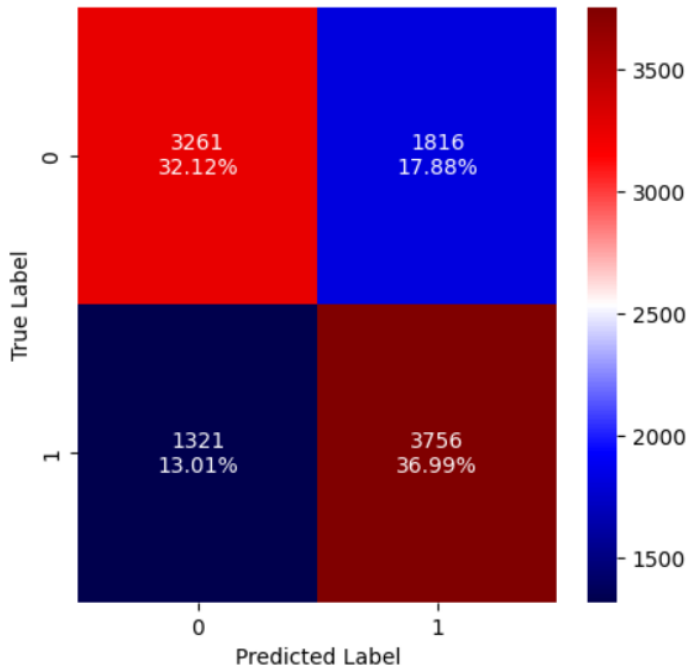

fig 95: Confusion Matrix Ada Boost Under Sampled

XG Boost Under Sampled: This looks like a sensible model

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              feature_weights=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None,
              n_jobs=None, num_parallel_tree=None, ...)
```

fig 96: XG Boost Under Sampled

	precision	recall	f1-score	support
0.0	0.87	0.87	0.87	5077
1.0	0.87	0.87	0.87	5077
accuracy			0.87	10154
macro avg	0.87	0.87	0.87	10154
weighted avg	0.87	0.87	0.87	10154

fig 97: Classification Report XG Boost Under Sampled

	Accuracy	Recall	Precision	F1_score
0	0.867835	0.866457	0.868852	0.867653

fig 98: Model Performance XG Boost Under Sampled

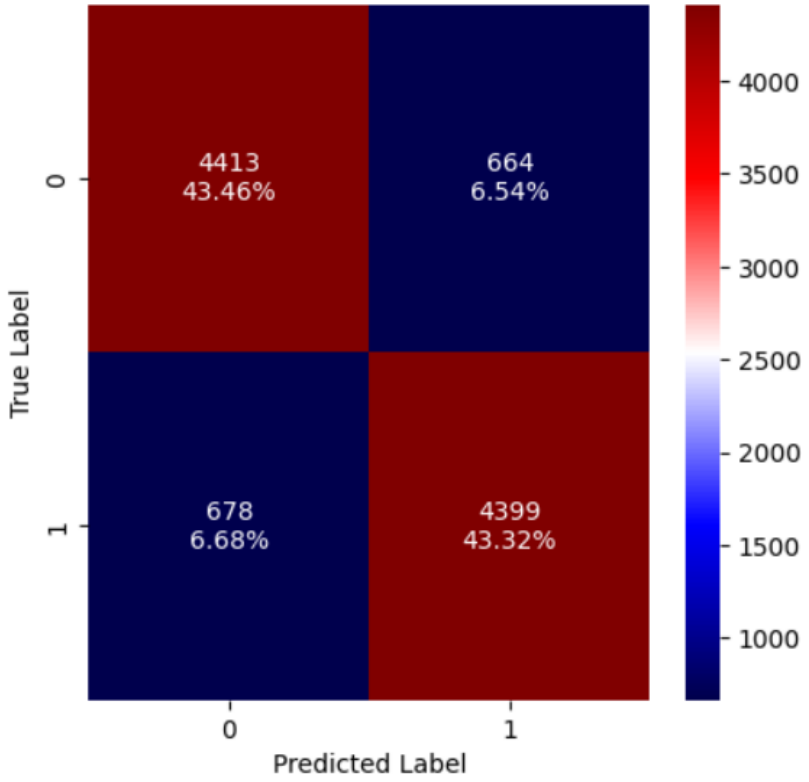

fig 99: Confusion Matrix XG Boost Under Sampled

BUILDING MODEL- HYPERPARAMETER TUNING

We'll be now tuning the models, for this we've chosen following few models:

1. Random Forest Classification undersampled
2. Decision Tree Classification undersampled
3. XG Boosting Classification undersampled
4. XG Boosting Classification oversampled

For hyperparameter tuning, we'll be modifying some parameters and finding the best parameters for the models using Random Search CV. Further, we'll be analysing the results of training data and comparing them using validation data.

Random Forest Classification Under Sampled: Model looks good but it should be noted that f1 score for validation data is more than training data.

```
RandomForestClassifier
RandomForestClassifier(class_weight='balanced', max_depth=3, max_features=0.2,
                       max_samples=0.3000000000000001,
                       min_impurity_decrease=0.003, min_samples_leaf=7,
                       n_estimators=125, random_state=42)
```

fig 100: Random Forest Under Sampled Tuned

	precision	recall	f1-score	support
0.0	0.74	0.60	0.66	5077
1.0	0.66	0.79	0.72	5077
accuracy			0.69	10154
macro avg	0.70	0.69	0.69	10154
weighted avg	0.70	0.69	0.69	10154

fig 101: Classification Report Random Forest Under Sampled Tuned (Training data)

	precision	recall	f1-score	support
0.0	0.58	0.59	0.58	1693
1.0	0.79	0.79	0.79	3403
accuracy			0.72	5096
macro avg	0.69	0.69	0.69	5096
weighted avg	0.72	0.72	0.72	5096

fig 102: Classification Report Random Forest Under Sampled Tuned (Validation data)

	Accuracy	Recall	Precision	F1_score
0	0.694406	0.786488	0.664172	0.720173

fig 103: Model Performance Random Forest Under Sampled Tuned (Training data)

	Accuracy	Recall	Precision	F1_score
0	0.723116	0.791654	0.793286	0.792469

fig 104: Model Performance Random Forest Under Sampled Tuned (Validation data)

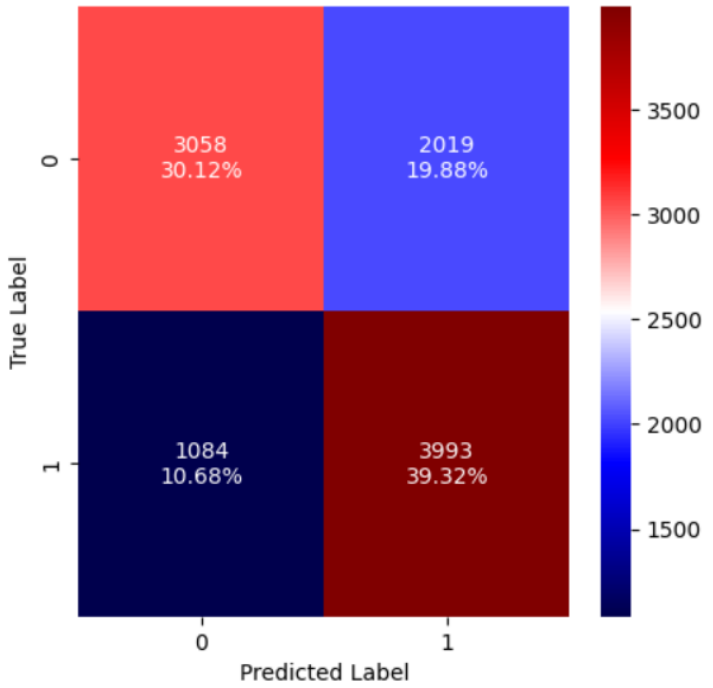

fig 105: Confusion Matrix Random Forest Under Sampled Tuned (Training data)

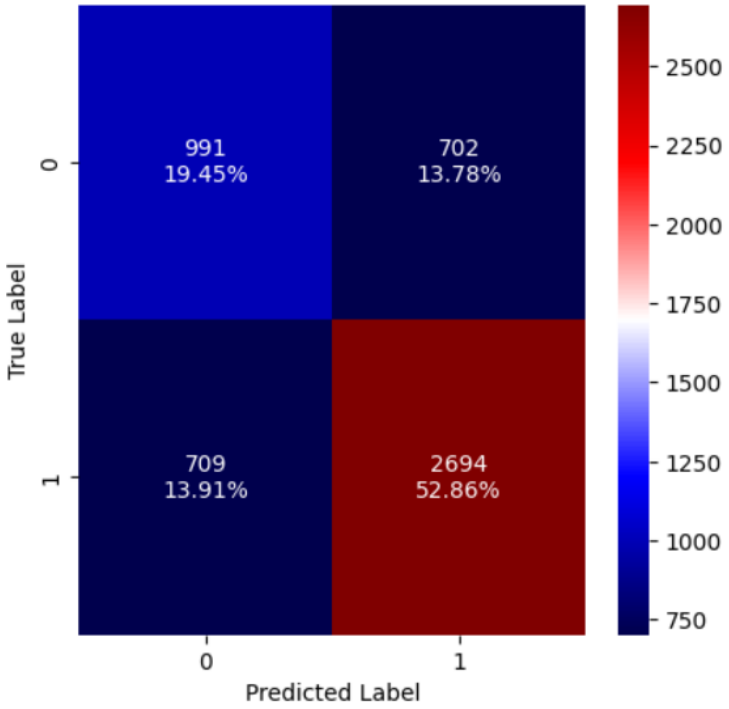

fig 106: Confusion Matrix Random Forest Under Sampled Tuned (Validation data)

Decision Tree Classification Under Sampled: Model looks good but it should be noted that f1 score for validation data is more than training data.

```
DecisionTreeClassifier
DecisionTreeClassifier(class_weight='balanced', max_depth=40, max_leaf_nodes=15,
                       min_samples_split=8, random_state=42)
```

fig 107: Decision Tree Under Sampled Tuned

	precision	recall	f1-score	support
0.0	0.75	0.61	0.68	5077
1.0	0.67	0.80	0.73	5077
accuracy			0.71	10154
macro avg	0.71	0.71	0.70	10154
weighted avg	0.71	0.71	0.70	10154

fig 108: Classification Report Decision Tree Under Sampled Tuned (Training data)

	precision	recall	f1-score	support
0.0	0.59	0.59	0.59	1693
1.0	0.80	0.80	0.80	3403
accuracy			0.73	5096
macro avg	0.69	0.70	0.69	5096
weighted avg	0.73	0.73	0.73	5096

fig 109: Classification Report Decision Tree Under Sampled Tuned (Validation data)

	Accuracy	Recall	Precision	F1_score
0	0.705929	0.799094	0.673585	0.730991

fig 110: Model Performance Decision Tree Under Sampled Tuned (Training data)

	Accuracy	Recall	Precision	F1_score
0	0.723116	0.791654	0.793286	0.792469

fig 111: Model Performance Decision Tree Under Sampled Tuned (Validation data)

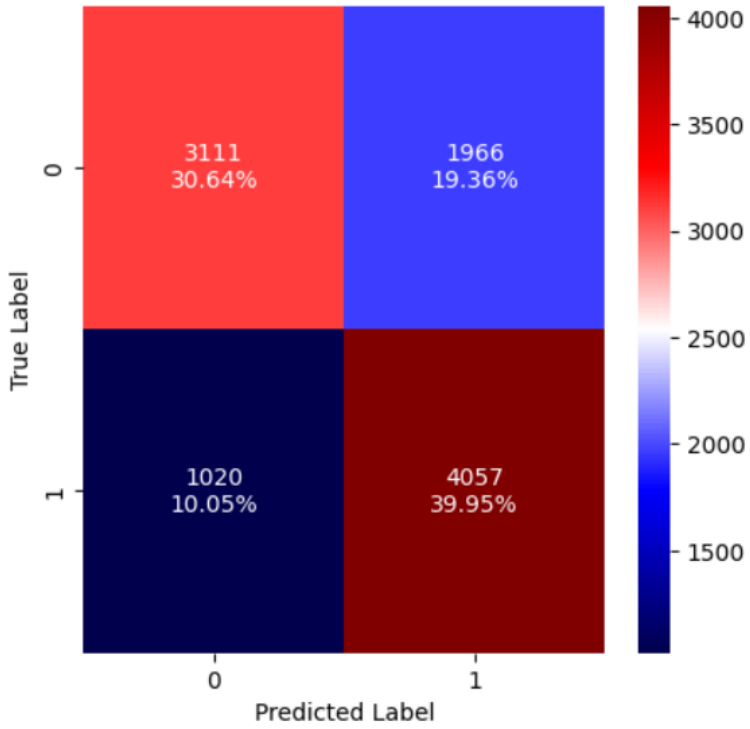

fig 112: Confusion Matrix Decision Tree Under Sampled Tuned (Training data)

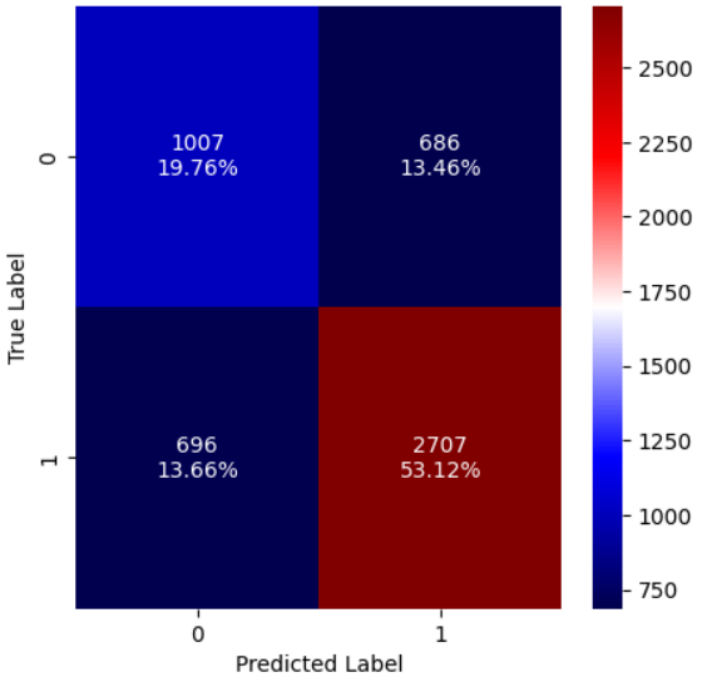

fig 113: Confusion Matrix Decision Tree Under Sampled Tuned (Validation data)

XG Boost Under Sampled: Model looks good but it should be noted that f1 score for validation data is more than training data.

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              feature_weights=None, gamma=5, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=0.1, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=15,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=75,
              n_jobs=None, num_parallel_tree=None, ...)
```

fig 114: XG Boost Under Sampled Tuned

	precision	recall	f1-score	support
0.0	0.75	0.69	0.72	5077
1.0	0.71	0.77	0.74	5077
accuracy			0.73	10154
macro avg	0.73	0.73	0.73	10154
weighted avg	0.73	0.73	0.73	10154

fig 115: Classification Report XG Boosting Under Sampled Tuned (Training data)

	precision	recall	f1-score	support
0.0	0.57	0.65	0.61	1693
1.0	0.81	0.76	0.79	3403
accuracy			0.72	5096
macro avg	0.69	0.71	0.70	5096
weighted avg	0.73	0.72	0.73	5096

fig 116: Classification Report XG Boosting Under Sampled Tuned (Validation data)

	Accuracy	Recall	Precision	F1_score
0	0.72789	0.768761	0.71067	0.738575

fig 117: Model Performance XG Boosting Under Sampled Tuned (Training data)

	Accuracy	Recall	Precision	F1_score
0	0.723901	0.759624	0.81443	0.786073

fig 118: Model Performance XG Boosting Under Sampled Tuned (Validation data)

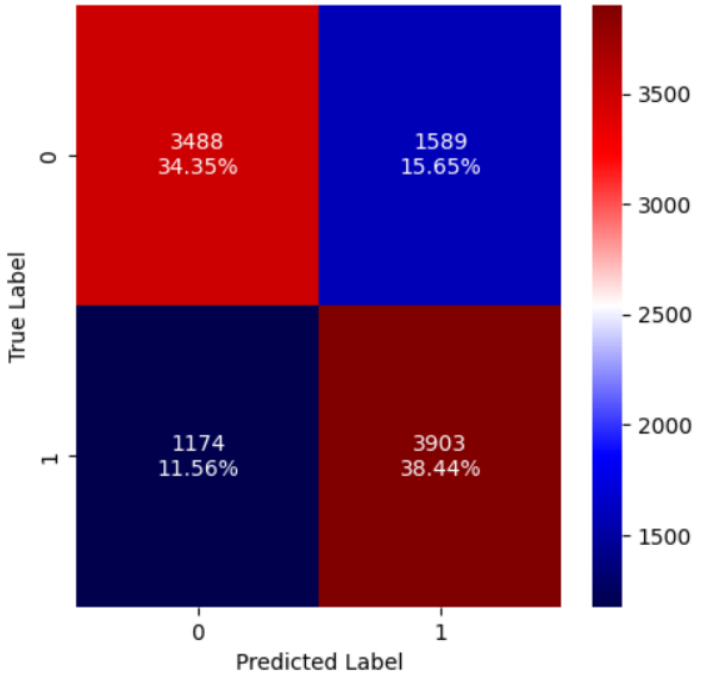

fig 119: Confusion Matrix XG Boosting Under Sampled Tuned (Training data)

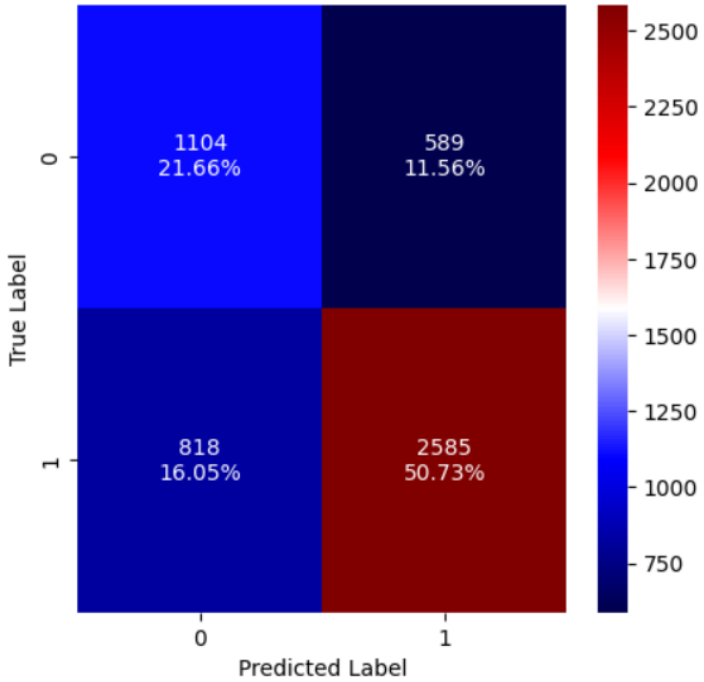

fig 120: Confusion Matrix XG Boosting Under Sampled Tuned (Validation data)

XG Boost Over Sampled: Model looks good but it should be noted that f1 score for validation data and training data seems to be sensible.

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              feature_weights=None, gamma=5, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=0.1, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=15,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=125,
              n_jobs=None, num_parallel_tree=None, ...)
```

fig 121: XG Boost Over Sampled Tuned

	precision	recall	f1-score	support
0.0	0.86	0.77	0.81	10211
1.0	0.79	0.87	0.83	10211
accuracy			0.82	20422
macro avg	0.83	0.82	0.82	20422
weighted avg	0.83	0.82	0.82	20422

fig 122: Classification Report XG Boosting Over Sampled Tuned (Training data)

	precision	recall	f1-score	support
0.0	0.65	0.52	0.58	1693
1.0	0.78	0.86	0.82	3403
accuracy			0.75	5096
macro avg	0.72	0.69	0.70	5096
weighted avg	0.74	0.75	0.74	5096

fig 123: Classification Report XG Boosting Over Sampled Tuned (Validation data)

	Accuracy	Recall	Precision	F1_score
0	0.822495	0.872099	0.793389	0.830884

fig 124: Model Performance XG Boosting Over Sampled Tuned (Training data)

	Accuracy	Recall	Precision	F1_score
0	0.748626	0.864531	0.782031	0.821214

fig 125: Model Performance XG Boosting Over Sampled Tuned (Validation data)

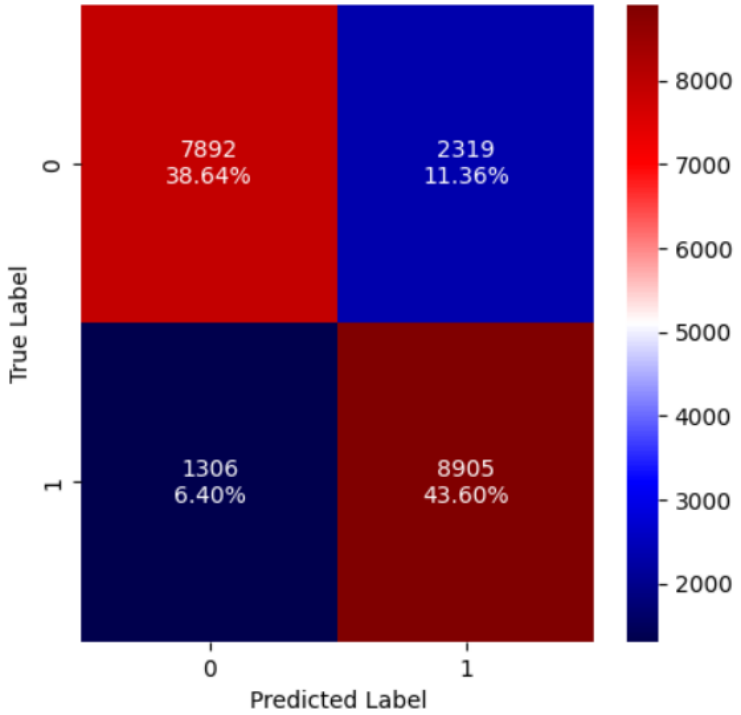

fig 126: Confusion Matrix XG Boosting Over Sampled Tuned (Training data)

RESULT

We'll be using XG Boosting Over Sampled Tuned Model, for our final test result. This because the difference between F1 Score of Training data and Validation data is least in the **XG Boosting Over Sampled Tuned Model**.

	Accuracy	Recall	Precision	F1_score
0	0.742151	0.861046	0.777041	0.81689

fig 127: Model Performance XG Boosting Over Sampled Tuned (Testing data)

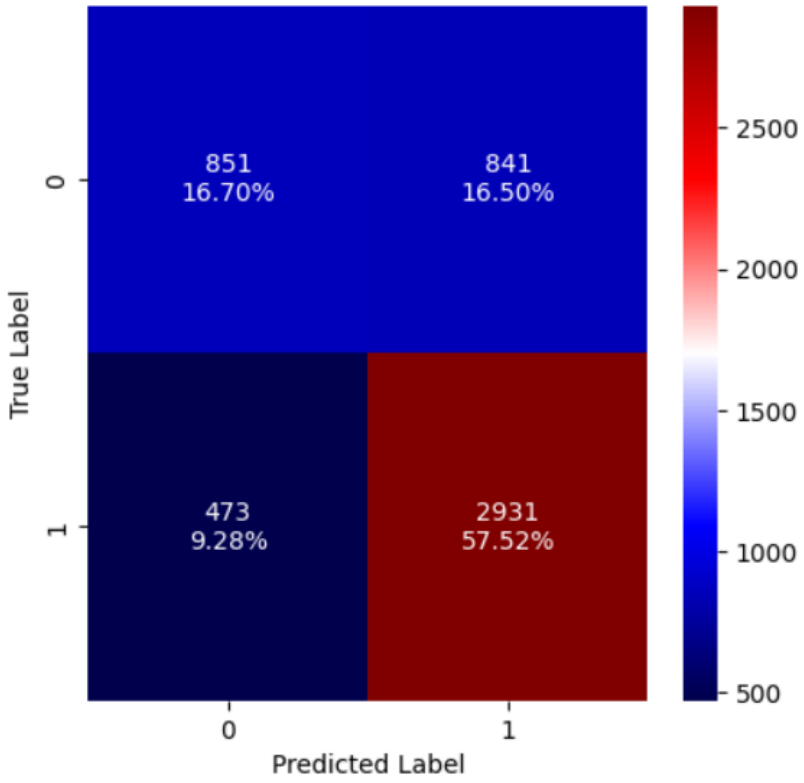

fig 128: Confusion Matrix XG Boosting Over Sampled Tuned (Testing data)

Comparison of Data

Metric	Training Data	Validation Data	Testing Data
Accuracy	0.82	0.74	0.74
Recall	0.87	0.86	0.86
Precision	0.79	0.78	0.77
F1 Score	0.83	0.82	0.81

Important Features

Following is the list of important features after the final test result as per the model tuned.

	Imp
education_of_employee_High School	0.316520
has_job_experience_Y	0.240818
education_of_employee_Master's	0.054647
unit_of_wage_Year	0.053600
region_of_employment_South	0.048859
continent_Europe	0.043927
region_of_employment_Midwest	0.033994
education_of_employee_Doctorate	0.026426
region_of_employment_Northeast	0.022760
continent_Asia	0.020160
region_of_employment_West	0.019990
full_time_position_Y	0.018255
continent_North America	0.015860
prevailing_wage	0.012524
requires_job_training_Y	0.012223
continent_South America	0.010573
unit_of_wage_Month	0.010398
no_of_employees	0.010268
unit_of_wage_Week	0.009955
yr_of_estab	0.009935
continent_Oceania	0.008310

fig 129: List of Important Features

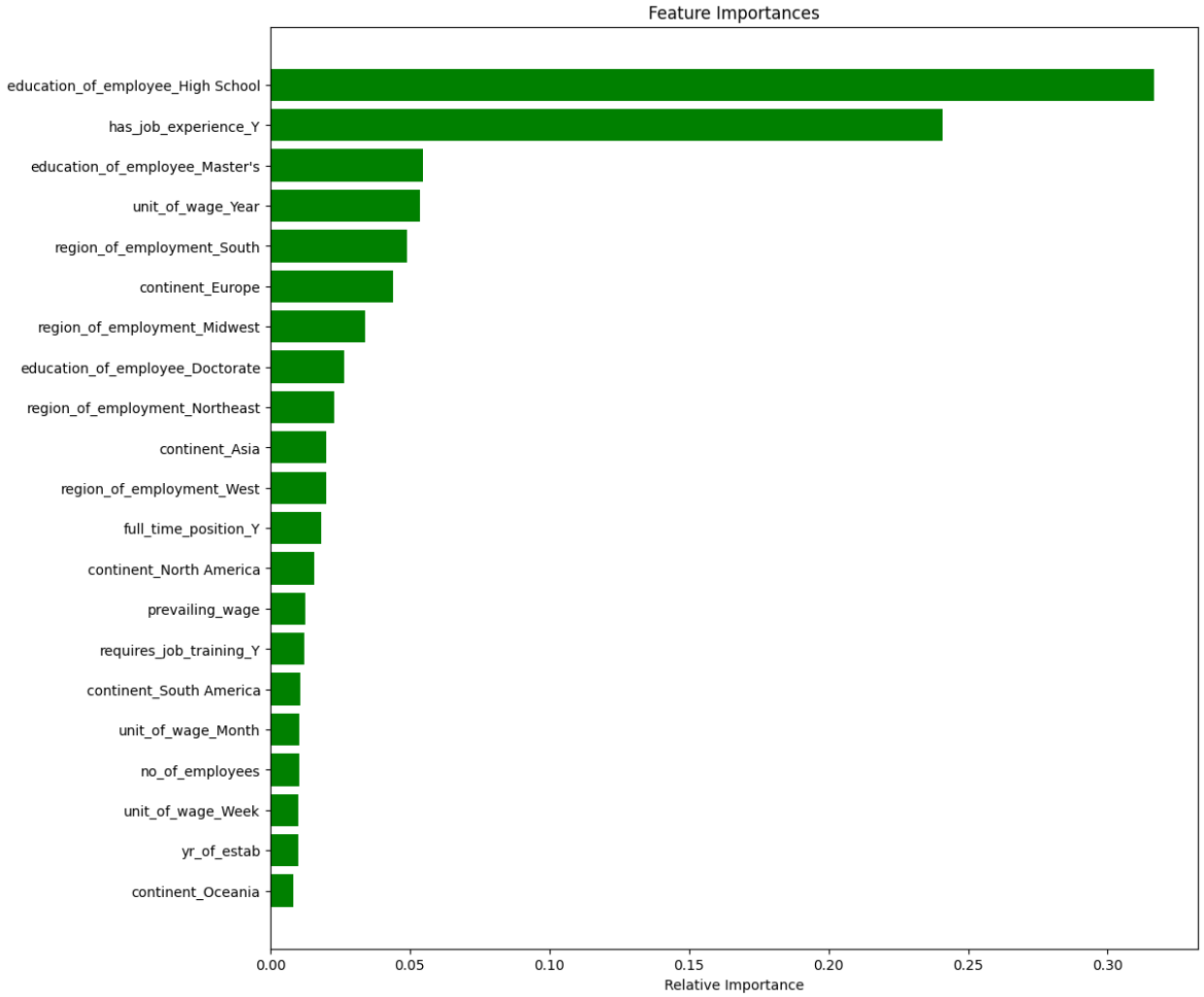

fig 130: Graphical Representation of List of Important Features

KEY TAKEAWAYS FOR THE BUSINESS

- Education of employee and job experience are the most important factors in the decision making of the visa status.
- People with job experience have higher chances of visa getting **certified.**
- People with higher educational background have higher chances of visa getting **certified.**
- People with lower educational background have higher chances of visa getting **denied.**
- People getting wages in units of year have higher chances of visa getting **certified.**
- People getting wages in units of hour have higher chances of visa getting **denied.**
- People from Africa, Asia and Europe are most likely to get their visa **certified.**
- People from South America are most likely to get their visa **denied.**

- Prevailing wages, year of establishment of the company and requirement of job training are such features, which do not play much important role in the decision making of the visa getting **certified or denied.**