Easy Visa Project

Supervised Learning - Ensemble

Yair Brama – October 2024

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Performance Summary of the following models – Decision tree, Bagging Classifier, Random Forest, Adaboost classifier, Gradient boost classifier, Xgboost classifier and Stacking classifier

Executive Summary - Business Context

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. In this project we will analyze the data provided and, with the help of a classification model we will 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.

Supervised Machine Learning Models

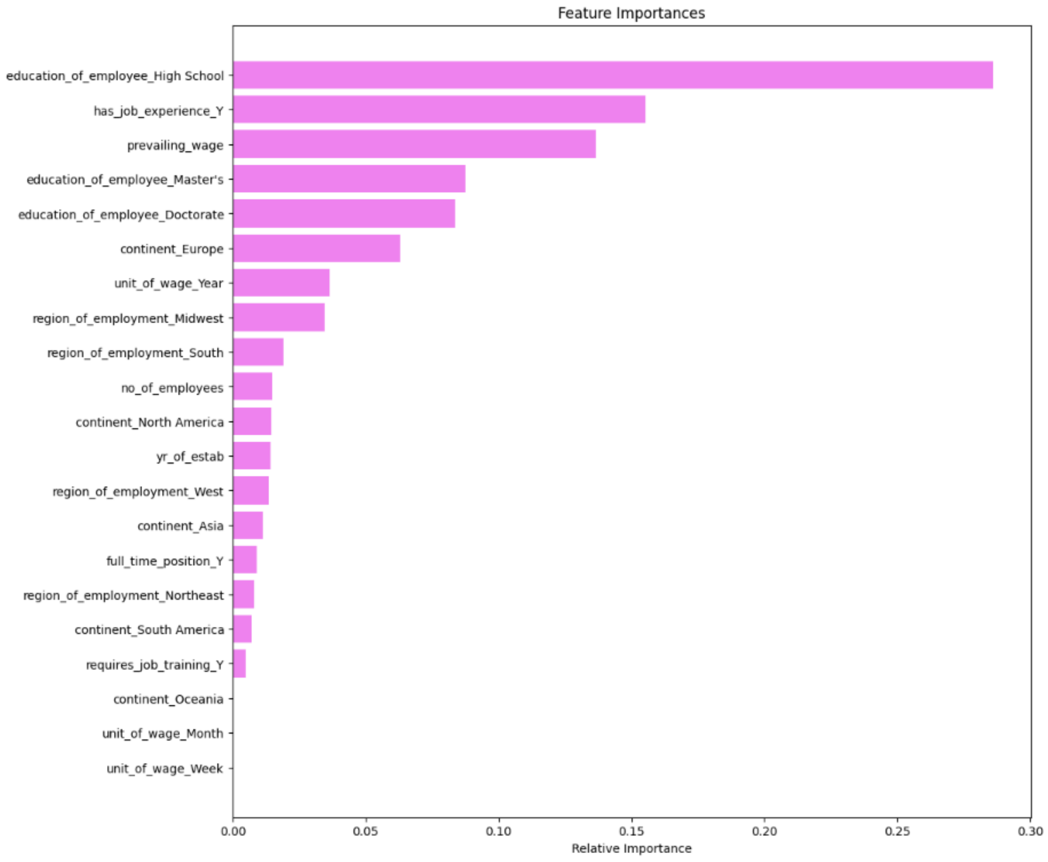
In this project we will review and compare several models, including decision tree, random forest, bagging model (based on decision tree) and 3 boosting models – Adaboosting, Gradient Boosting and XG Boost (Extreme Gradient Boosting). We will also run a Stacking model that uses 4 models one on top of the other – Adaboost, Gradient Boost, Random Forest and XG Boost as the final estimator.

Executive Summary - Features Importance and Evaluation Criterion

Feature Importance – Based on the models and the EDA, we see that the important features for getting a visa are:

- **Employee education**
- **Previous job experience**
- **Salary**
- **Original continent**

Model Evaluation – It is important for us to find the balance between false positives (precision score) and the false negatives (recall score), since for our purposes they are both equally bad.
Therefore, we will look at the F1 score and see which model can maximize this score while keeping the overfitting level low as possible.



Executive Summary – Model Recommendations

<u>The Overall Best</u> – The best results considering the highest F1 score and the lowest overfitting on training set is the **tuned XG Boost classifier**. The F1 score is 82.13% and 1.18% overfitting.

<u>The Least Overfitting</u>– The least overfitting model is the **Adaboost classifier** (with default parameters). The difference between the training F1 score and the testing F1 score is 0.2% (F1 score is 81.65%)

<u>The Most Overfitting</u> – The best training results were achieved with the **default decision tree**, that gave us a perfect score for the training set (100%). However, when running it on the test set, we see a difference of more than 25%, which makes it the most overfitting model.

Executive Summary – Final Recommendations

Based on the results of our models, we recommend to use the **Tuned XG Boost Classifier** model, with the following parameters setup:

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='logloss', feature_types=None, gamma=3, grow_policy=None, importance_type=None, interaction_constraints=None, learning_rate=0.05, 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=50, n_jobs=None, num_parallel_tree=None, random_state=1)

Testing performance:

	Accuracy	Recall	Precision	F1
0	0.744898	0.877767	0.771655	0.821298

Solution Approach – All models

Data initial review

- Check for null values
- Check for duplicates
- Check data types

EDA

- Univariate Analysis – Box plots, Histograms
- Multivariate Analysis - Check for correlation (Heatmap), visualize the correlated variables

Data Preprocessing

- Check outliers
- Change case status values to 1 and 0

Model Building

- Split the dataset between training and test data
- Set the case status as the predicted variable

Run Model with default params

- Run the model with default parameters
- Check scores
- Compare training and test results

Run Model after tuning

- Perform GridSearch to find the optimal values of the hyperparameters
- Check scores
- Compare the confusion matrix of the training and test sets

Final Model Summary

- Compare the results of all iterations and recommend the best option

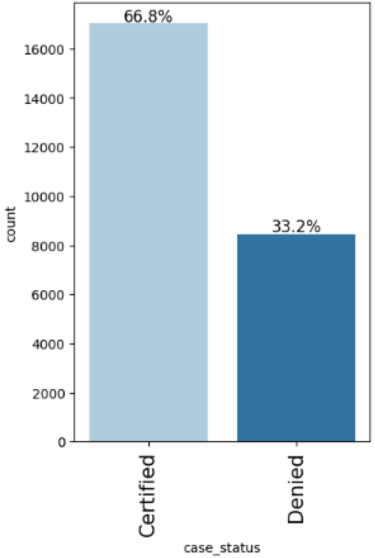
Data Description

- 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 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

EDA Results - Data Overview

The data includes 25480 rows and 12 columns. There are no missing values and no duplicates. 1/3 of the visa requests in our set are denied, and 2/3 were certified.

Visa Status – Dependent feature



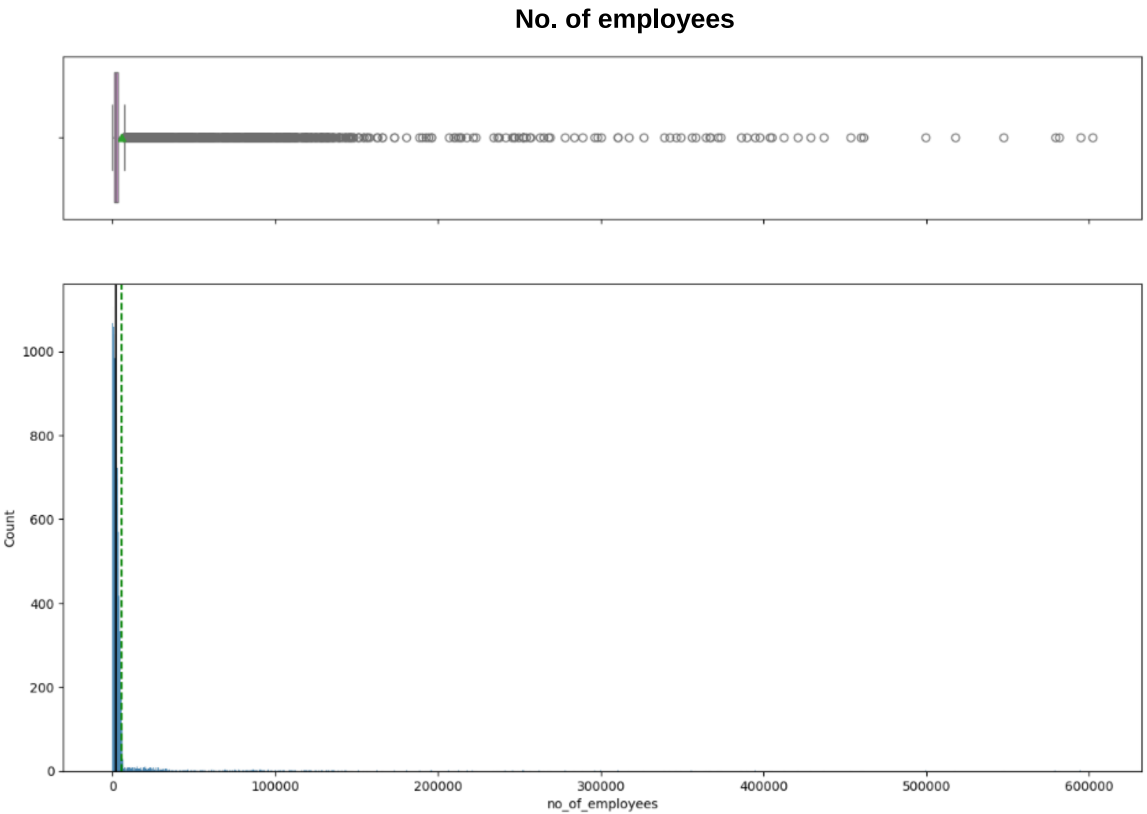

```
Certified    17018
Denied        8462
```

Statistical summary of the numeric fields:

	no_of_employees	yr_of_estab	prevailing_wage
count	25480.000000	25480.000000	25480.000000
mean	5667.043210	1979.409929	74455.814592
std	22877.928848	42.366929	52815.942327
min	-26.000000	1800.000000	2.136700
25%	1022.000000	1976.000000	34015.480000
50%	2109.000000	1997.000000	70308.210000
75%	3504.000000	2005.000000	107735.512500
max	602069.000000	2016.000000	319210.270000

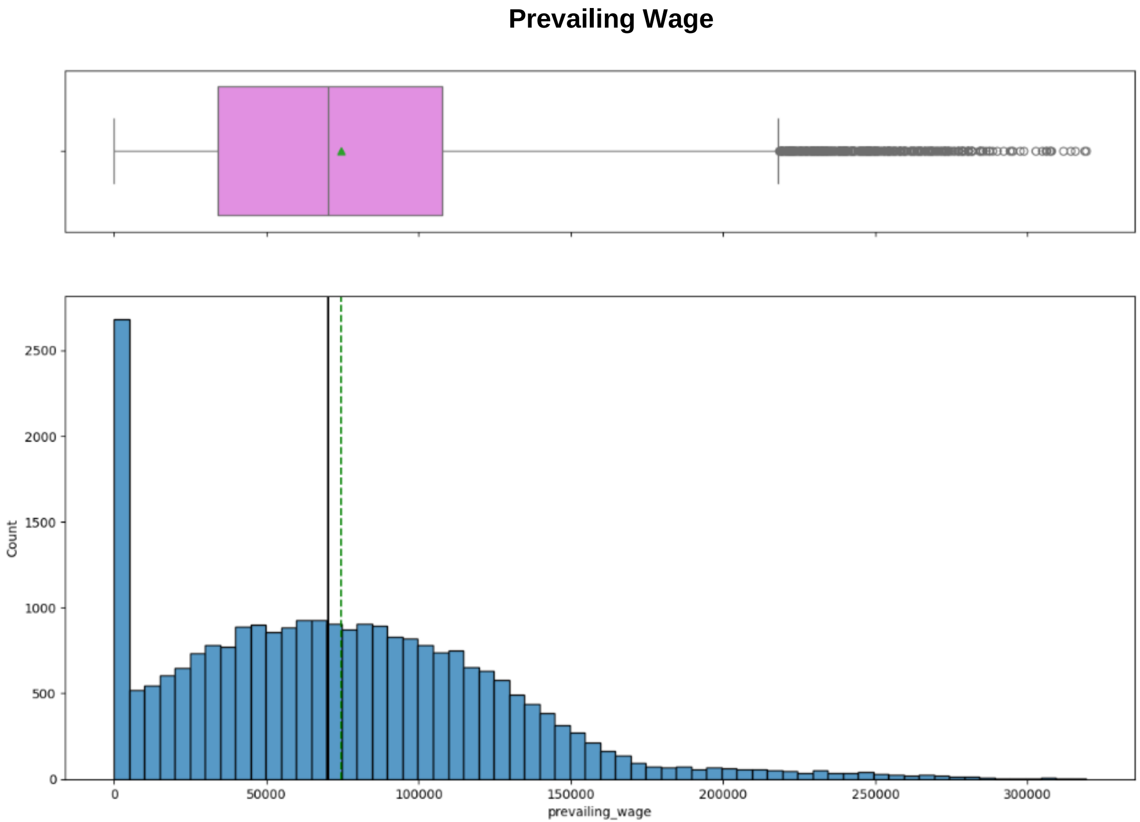
EDA Results - Univariate Analysis

Observation – The companies' size varies, where 50% of the companies have < 2000 workers, and 25% of the companies have more than 3000 workers.



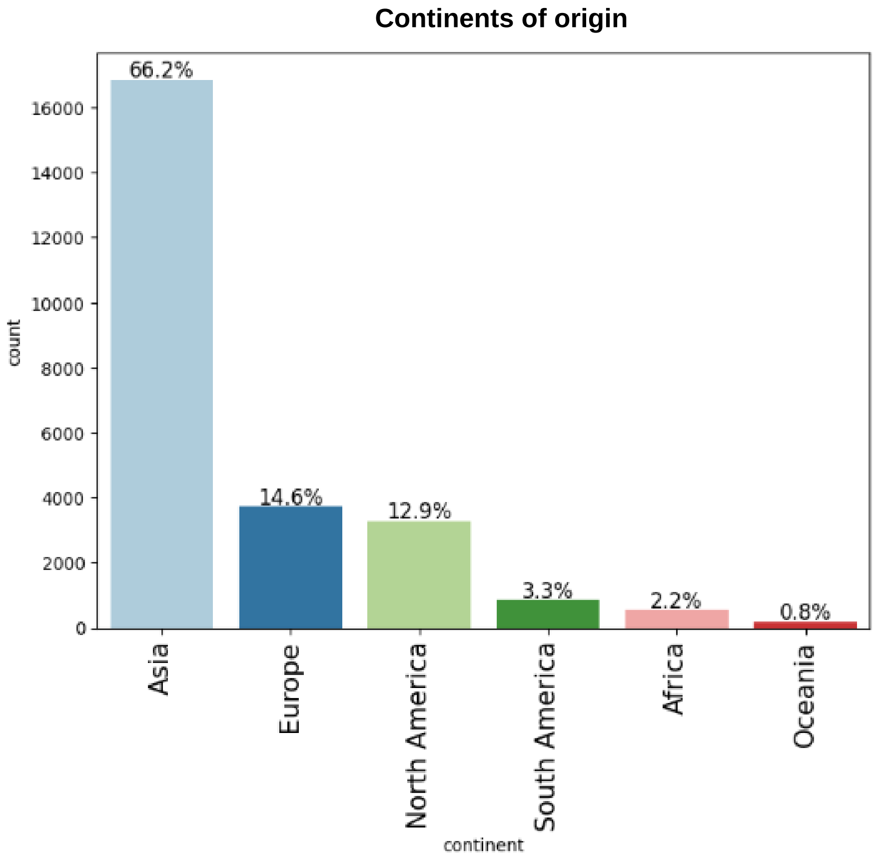
EDA Results - Univariate Analysis

Observation – The median salary is around $70,000, and most of the records are indicated as ‘annual’. Around 2500 records are hourly or monthly, and we can see that on the left side of the histogram. If we look at the annual salaries, we see a right skew, with outliers that have very high salaries (>$150,000)



EDA Results - Univariate Analysis

Asia	16861
Europe	3732
North America	3292
South America	852
Africa	551
Oceania	192

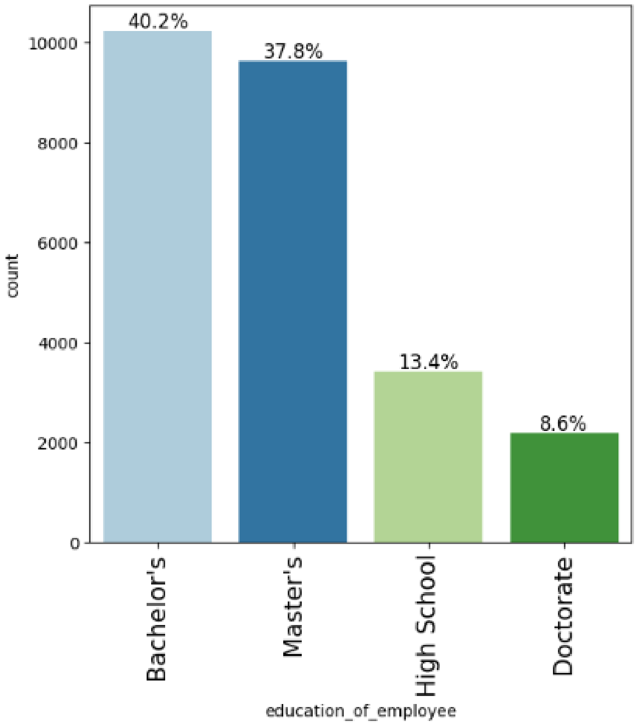


Observation – 66% of the visas requests in our data set come from Asia. Further analysis will show if the minority of visas coming from Europe and North America have higher chance to be granted.

EDA Results - Univariate Analysis

Bachelor's	10234
Master's	9634
High School	3420
Doctorate	2192

Education of Employees

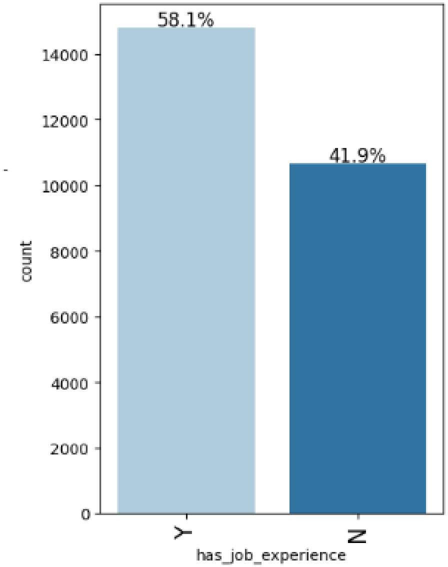


Observation – Unsurprisingly, almost 87% of the requests come from workers who have higher degrees.

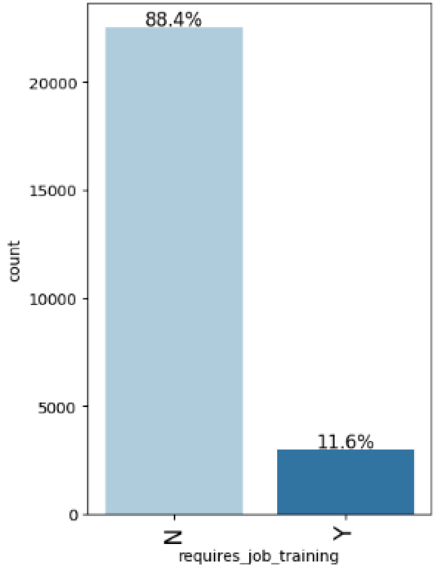
EDA Results - Univariate Analysis

Has Job Experience

Y	14802
N	10678



Requires Job Training



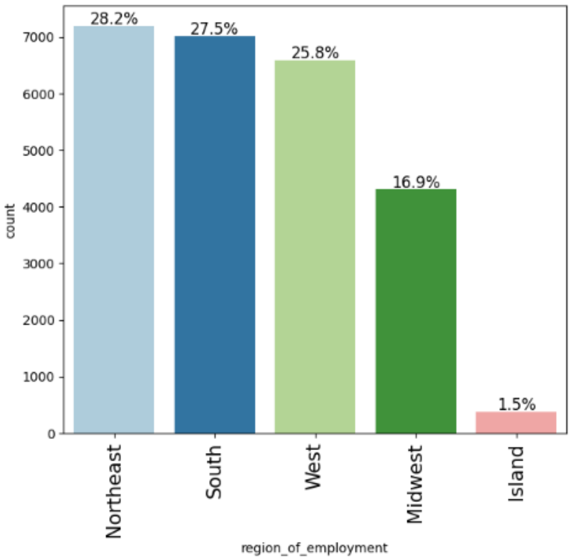
N	22525
Y	2955

Observation – ~90% of the jobs require training, which indicates that these visa requests are for people who are willing to be trained as part of their move to the US. We can see that ~40% of the employees have no experience, indicating younger people who are willing to take more risks.

EDA Results - Univariate Analysis

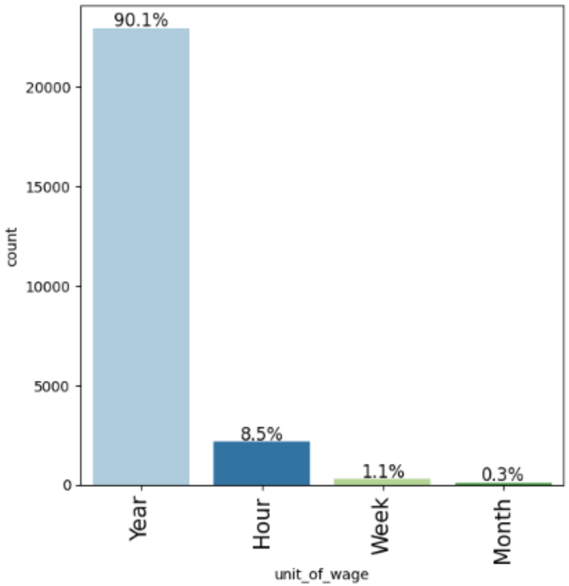
Observation – Most of the jobs are in regions that have urban centers with a lot of opportunities and demand for educated workers. The Midwest which has fewer of these centers, has less opportunities.
90% of the wages are annual, which means they are more stable and profitable.

Regions of employment



Northeast	7195
South	7017
West	6586
Midwest	4307
Island	375

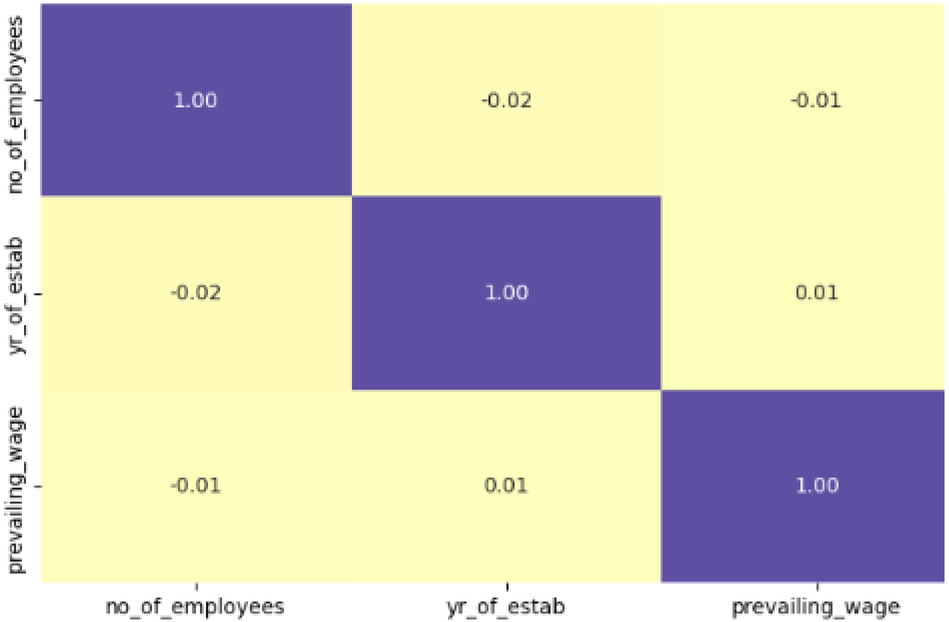
Unit of wage



Year	22962
Hour	2157
Week	272
Month	89

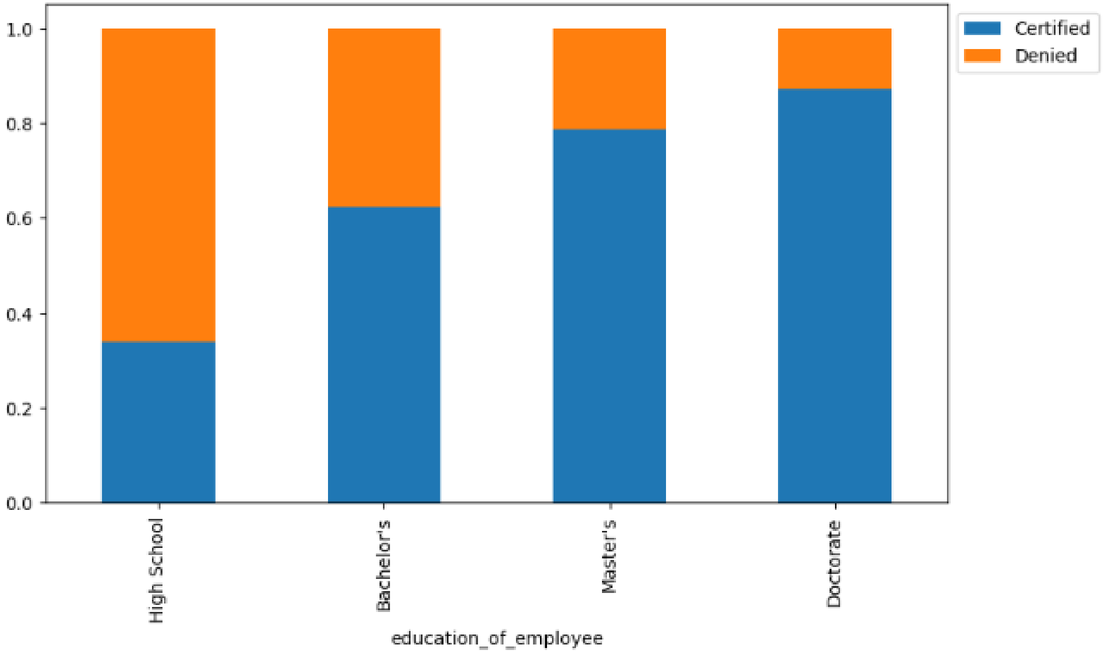
Multivariate Analysis – Heatmap/Correlation

Observation – Correlation check between the numerical fields show no correlation



Multivariate Analysis – Education vs. Visa Status

case_status education_of_employee	Certified	Denied	All
All	17018	8462	25480
Bachelor's	6367	3867	10234
High School	1164	2256	3420
Master's	7575	2059	9634
Doctorate	1912	280	2192



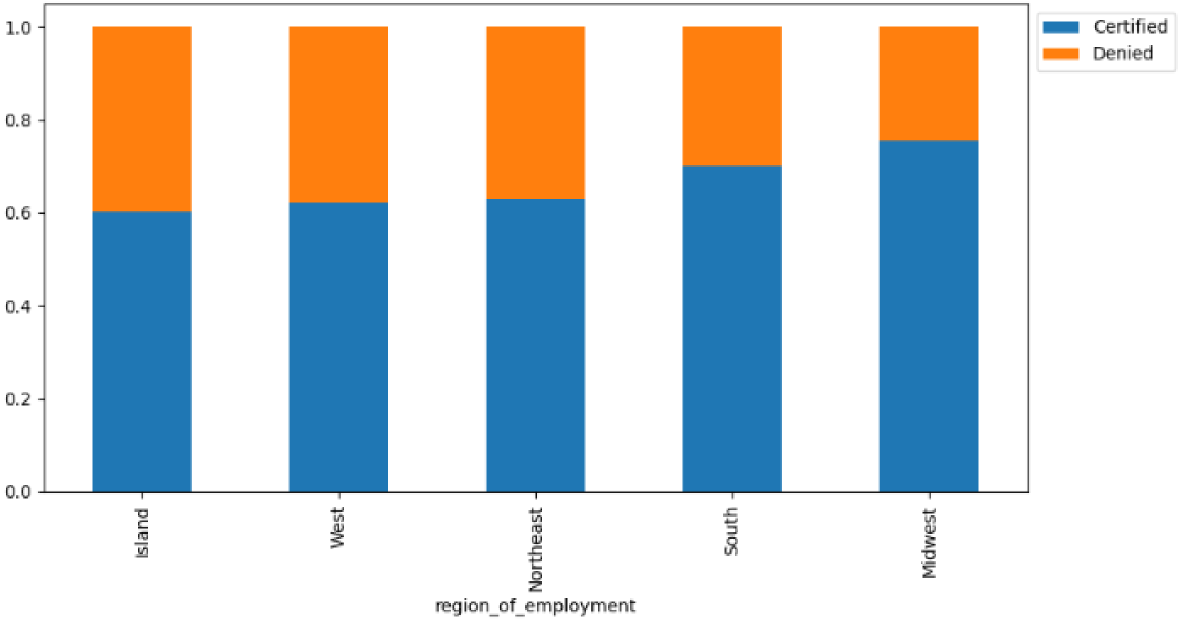
Observation – We can see how the higher the education, the higher the chance to get a visa

Multivariate Analysis – Education in Regions

Observation – In all regions, the demand for higher education is consistent.

Multivariate Analysis – Visa Status across the Regions

case_status	Certified	Denied	All
region_of_employment			
All	17018	8462	25480
Northeast	4526	2669	7195
West	4100	2486	6586
South	4913	2104	7017
Midwest	3253	1054	4307
Island	226	149	375

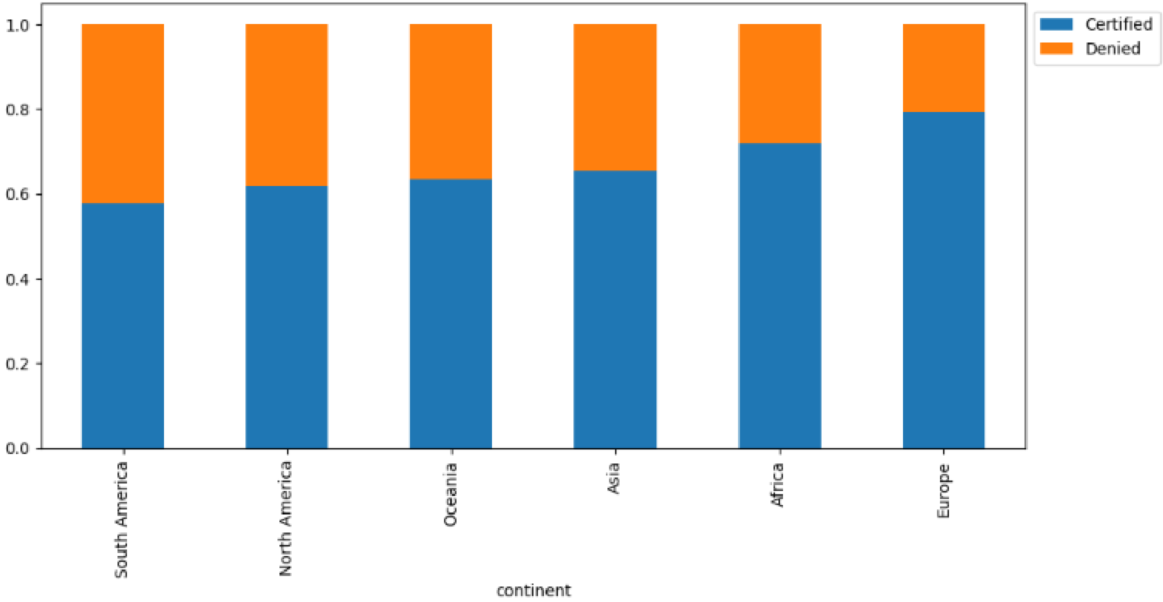


Observation – The ratio of denied vs. certified requests is pretty much the same across all regions. The slightly higher chance for certified visa is in the Midwest, where there are fewer opportunities.

Multivariate Analysis – Visa Status and Origin Continent

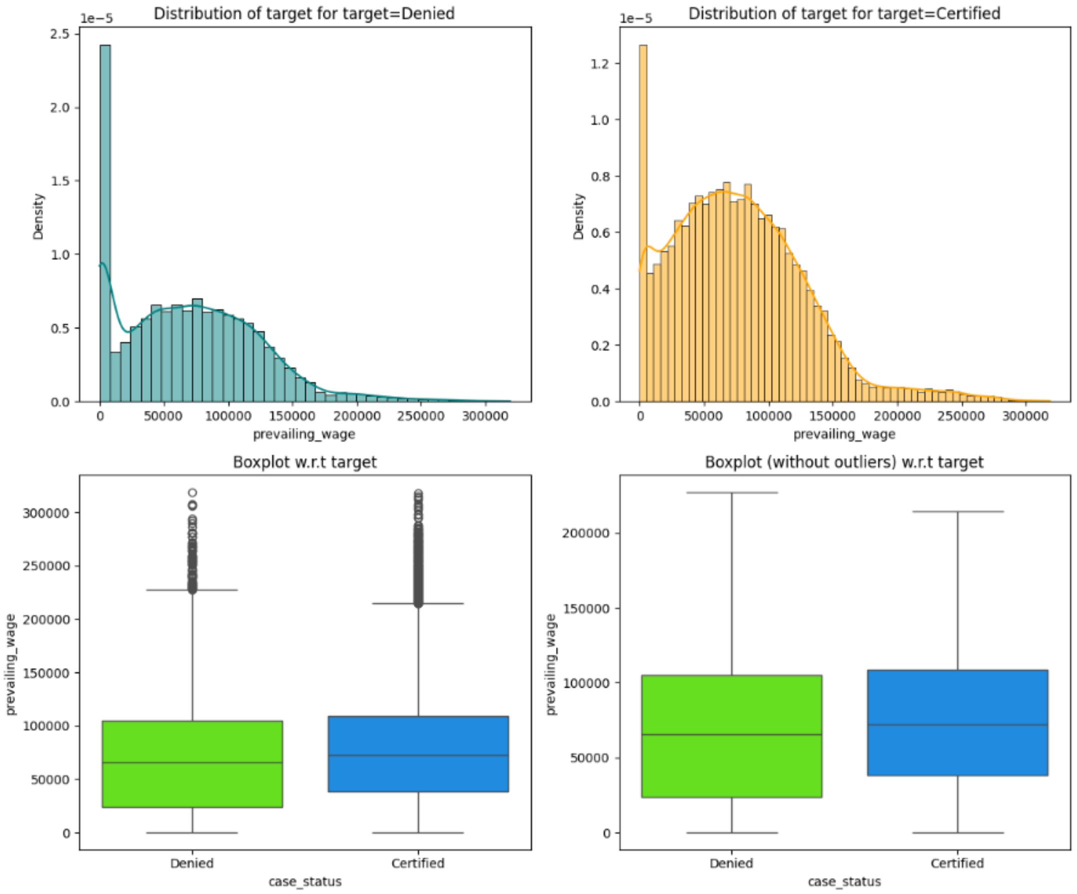
Observation – The highest acceptance rate is for European applicants, but not by far. Requests from Asia (66% of the requests), are in the middle, behind Europe and Africa, with around 65% acceptance rate

case_status	Certified	Denied	All
continent			
All	17018	8462	25480
Asia	11012	5849	16861
North America	2037	1255	3292
Europe	2957	775	3732
South America	493	359	852
Africa	397	154	551
Oceania	122	70	192



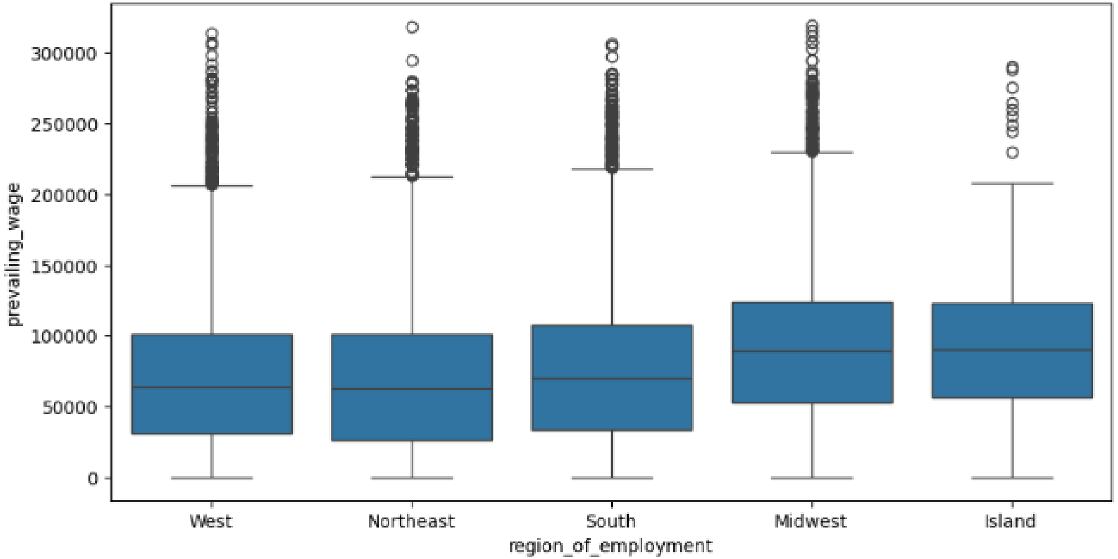
Multivariate Analysis – Wages vs. Visa Status

Observation – Whether with or without the outliers of the super higher salaries, there is no much difference in the acceptance rate of the visas. The only difference is in the lower salaries, where we see more requests are denied.



Multivariate Analysis – Wages in the Regions

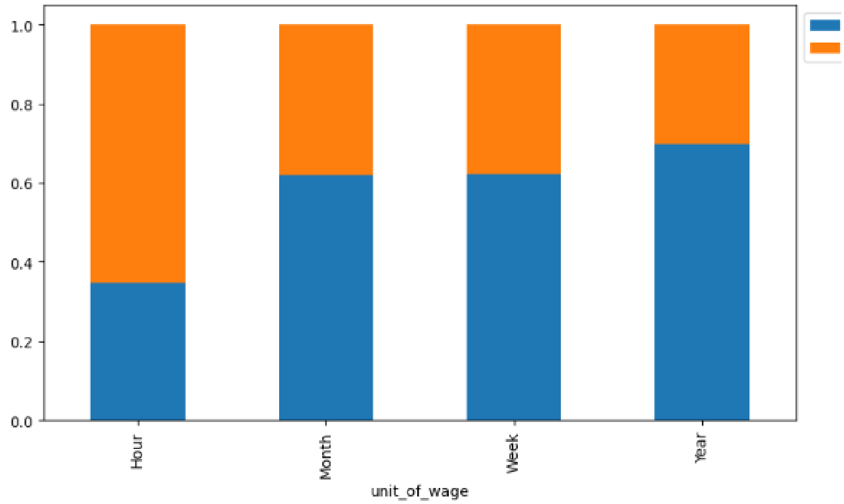
Observation – In the more desired regions (west, northeast and south), we see very similar wages. A slightly higher range of salaries is in the Midwest and Hawaii, maybe because they are less desirable for visa workers, so higher salary is offered.



Multivariate Analysis – Unit of Wage vs. Visa Status

Observation – As we saw above, there is a smaller acceptance rate for hourly rate visa workers. We can see that the more stable and long term the wage unit is, the higher the demand for visa workers.

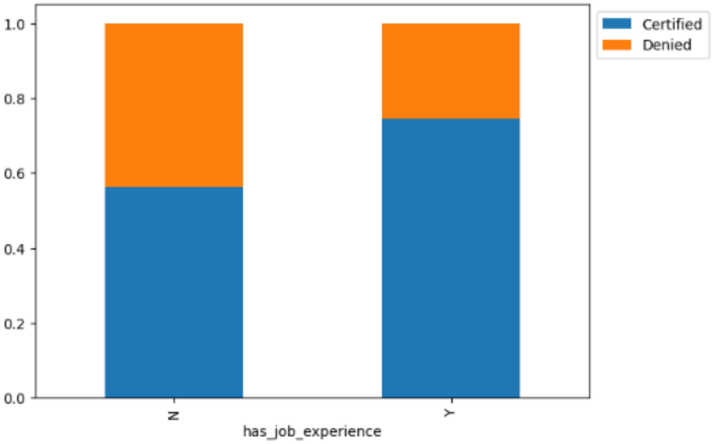
case_status unit_of_wage	Certified	Denied	All
All	17018	8462	25480
Year	16047	6915	22962
Hour	747	1410	2157
Week	169	103	272
Month	55	34	89



Multivariate Analysis – Job Experience vs. training and status

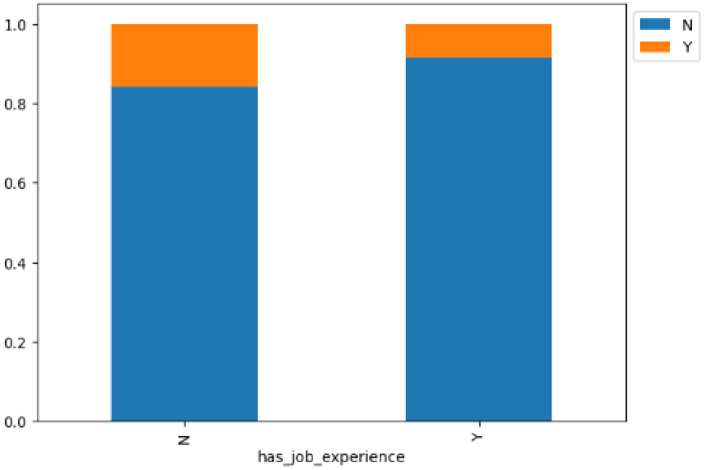
Job experience and acceptance rate

case_status	Certified	Denied	All
has_job_experience			
All	17018	8462	25480
N	5994	4684	10678
Y	11024	3778	14802



Job experience and training

requires_job_training	N	Y	All
has_job_experience			
All	22525	2955	25480
N	8988	1690	10678
Y	13537	1265	14802

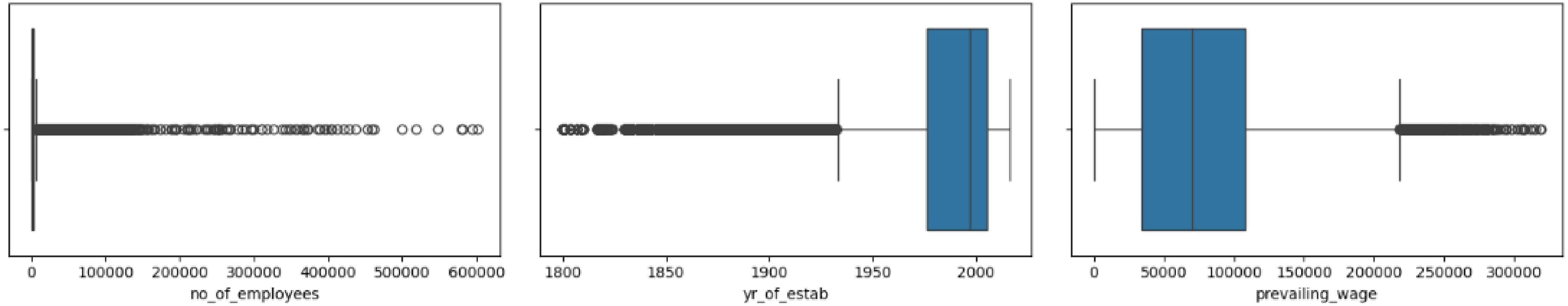


Observation – Having a job experience rases the chance to get a certified visa. Also, we can see that the jobs mostly require training, regardless if the workers have experience or not.

Data Pre-processing – Feature Engineering and Outlier Check

- **Outliers Check** – The numeric features have outliers, but we will not remove or normalize them at this point, before running the models.

- **Data Types** – The case_status is set to be numerical (certified = 1, denied = 0)



Training and Test Sets

- **30/70 Spilt** – The data set is split to training set (70% of the rows) and test set (30%).
- **Strata** – By setting the parameter stratify=Y, we make sure that the ratio of the classes (visa certified/denied) is kept in both sets, training and test.

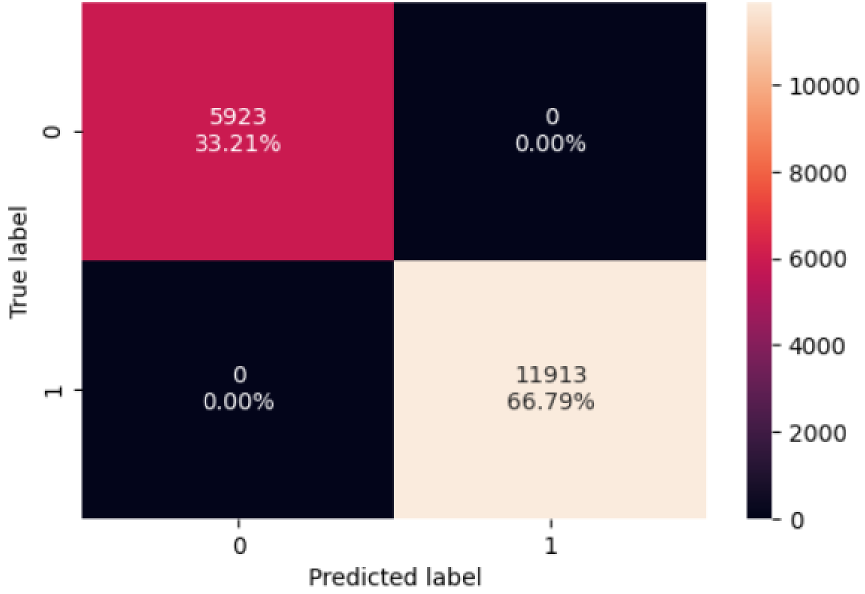
Train and test sets

```
Shape of Training set :  (17836, 21)
Shape of test set :  (7644, 21)
Percentage of classes in training set:
1    0.667919
0    0.332081
Name: case_status, dtype: float64
Percentage of classes in test set:
1    0.667844
0    0.332156
Name: case_status, dtype: float64
```

Decision Tree Model

Decision Tree Performance Summary – Training Set

Default Setting (random_state=1)

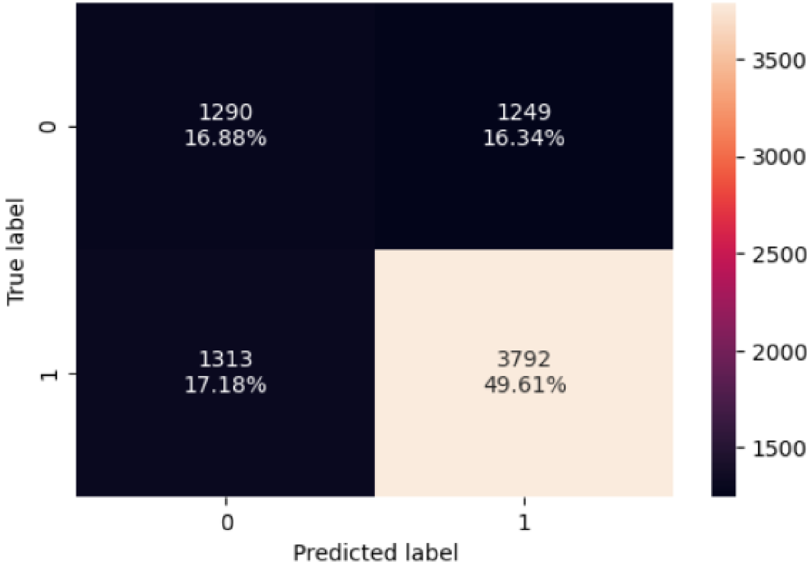


	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0

Results – As expected from a decision tree model, the results are very good for the training set, as we didn't limit (prune) the tree in any way.

Decision Tree Performance Summary – Test Set

Default Setting



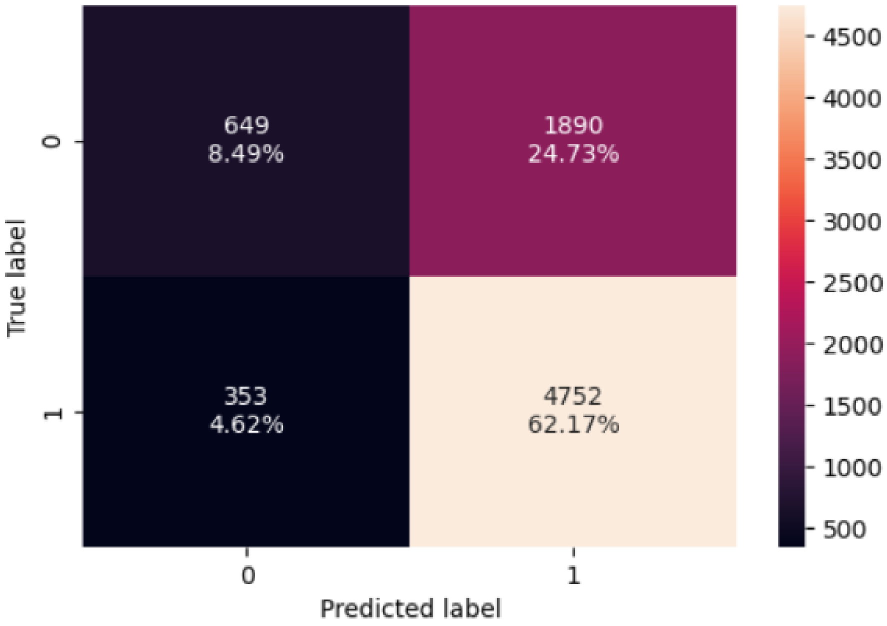
	Accuracy	Recall	Precision	F1
0	0.664835	0.742801	0.752232	0.747487

Results – In the test set we see that this model is overfitting – ~25% difference in the results of the scores.

Decision Tree – Hyperparameters Tuning

Decision Tree Performance Summary – Hyperparameter Tuned

(class_weight='balanced', max_depth=10, max_leaf_nodes=2, min_impurity_decrease=0.0001, min_samples_leaf=3, random_state=1)



Training performance:

	Accuracy	Recall	Precision	F1
0	0.712548	0.931923	0.720067	0.812411

Testing performance:

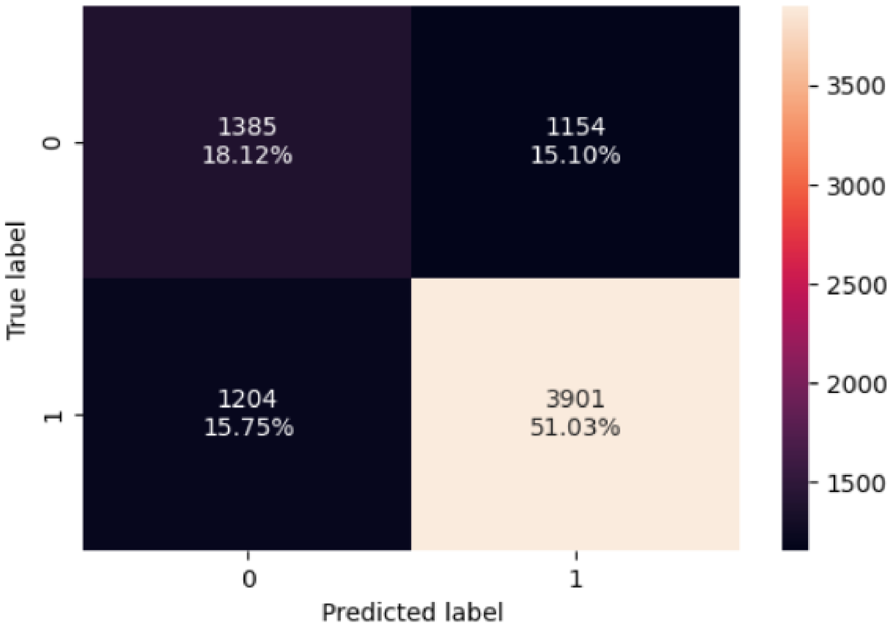
	Accuracy	Recall	Precision	F1
0	0.706567	0.930852	0.715447	0.809058

Results – By hyperparameters tuning, we can reduce the overfitting dramatically, and get a F1 score of 80%

Bagging Classifier Model

Bagging Classifier Performance Summary – Default Parameters

(random_state=1)



	Accuracy	Recall	Precision	F1
0	0.985198	0.985982	0.99181	0.988887

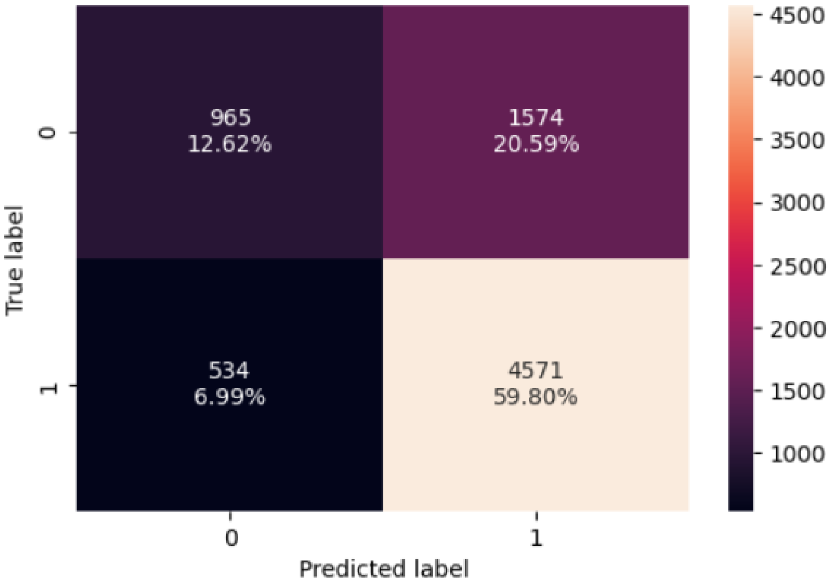
	Accuracy	Recall	Precision	F1
0	0.691523	0.764153	0.771711	0.767913

Results – This model shows strong overfitting, with almost 100% success in training, and ~25% reduction in test

Bagging Classifier – Hyperparameter Tuned

Bagging Classifier Performance Summary – Hyperparameters Tuned

(max_features=0.7, max_samples=0.7, n_estimators=100, random_state=1)



	Accuracy	Recall	Precision	F1
0	0.996187	0.999916	0.994407	0.997154

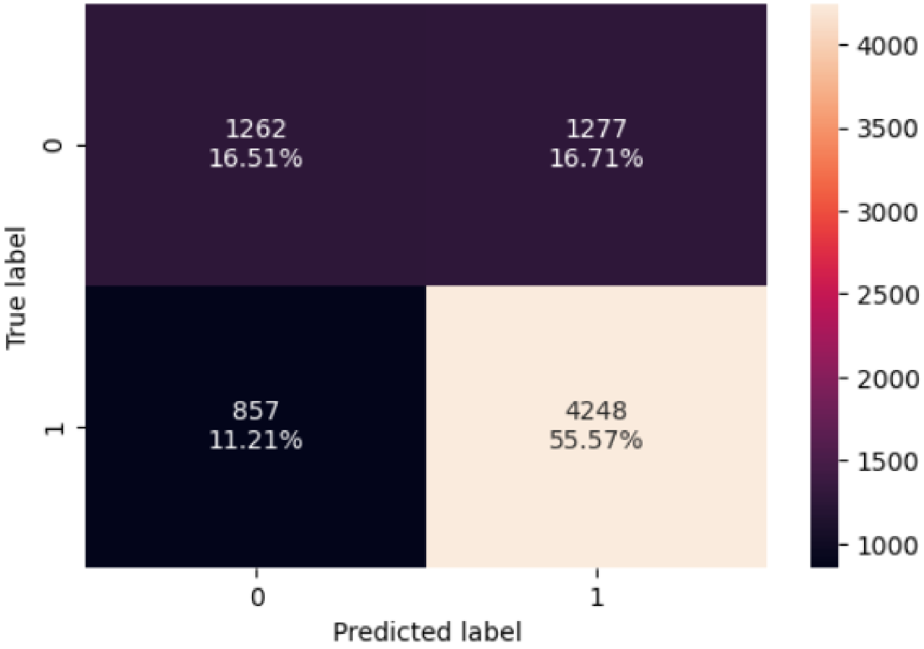
	Accuracy	Recall	Precision	F1
0	0.724228	0.895397	0.743857	0.812622

Results – When running the tuned bagging classifier, we see better results in the F1 score of the test set, but we also see strong overfitting, with ~20% difference between training and test sets.

Random Forest Model

Random Forest Performance Summary – Default Parameters

(random_state=1)



Training performance:

	Accuracy	Recall	Precision	F1
0	0.999944	0.999916	1.0	0.999958

Testing performance:

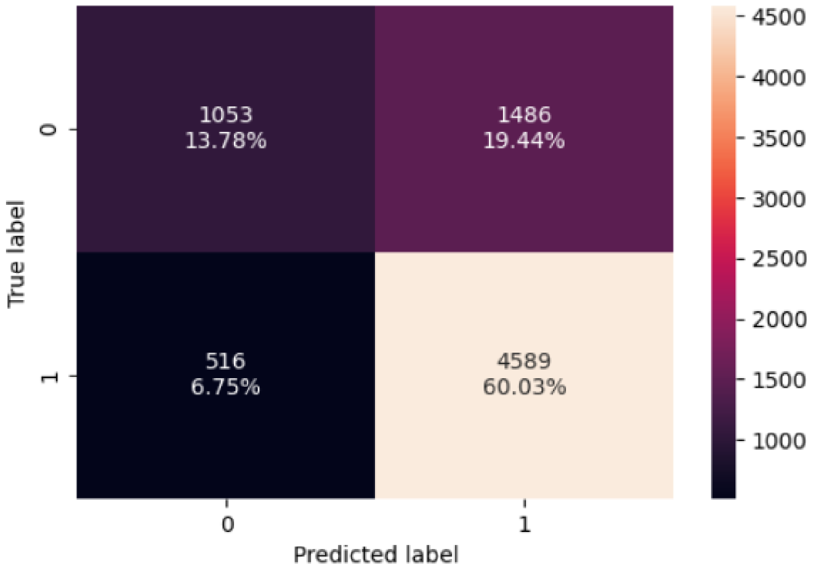
	Accuracy	Recall	Precision	F1
0	0.720827	0.832125	0.768869	0.799247

Results – Random Forest in default mode shows strong overfitting in training, we will try it again with tuning.

Random Forest – Hyperparameter Tuned

Random Forest Performance Summary – Hyperparameters Tuned

(max_depth=10, min_samples_split=7, n_estimators=20, oob_score=True, random_state=1)



Training performance:

	Accuracy	Recall	Precision	F1
0	0.769119	0.91866	0.776556	0.841652

Testing performance:

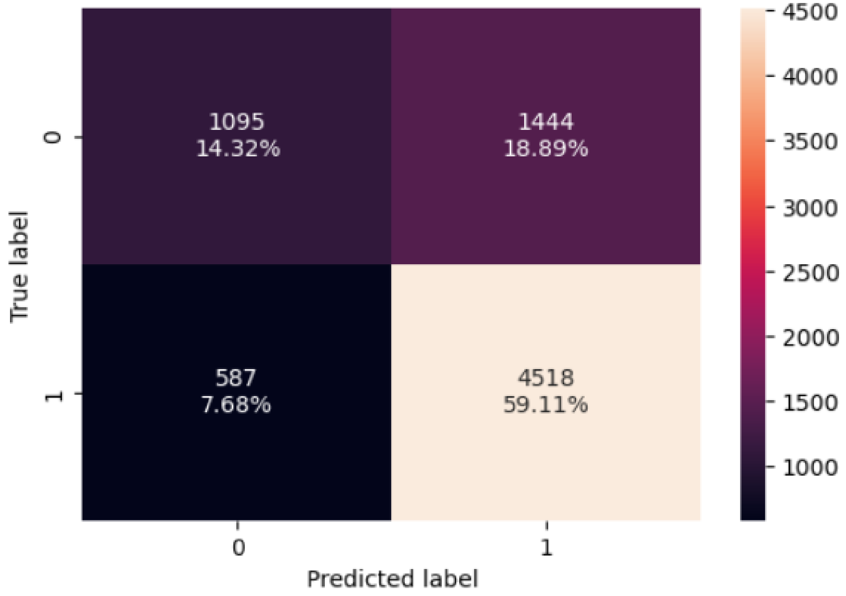
	Accuracy	Recall	Precision	F1
0	0.738095	0.898923	0.755391	0.82093

Results – The best results so far, with very small overfitting and F1 score of 82%

Adaboost Classifier Model

Adaboost Classifier Performance Summary – Default Parameters

(random_state=1)



	Accuracy	Recall	Precision	F1
0	0.738226	0.887182	0.760688	0.81908

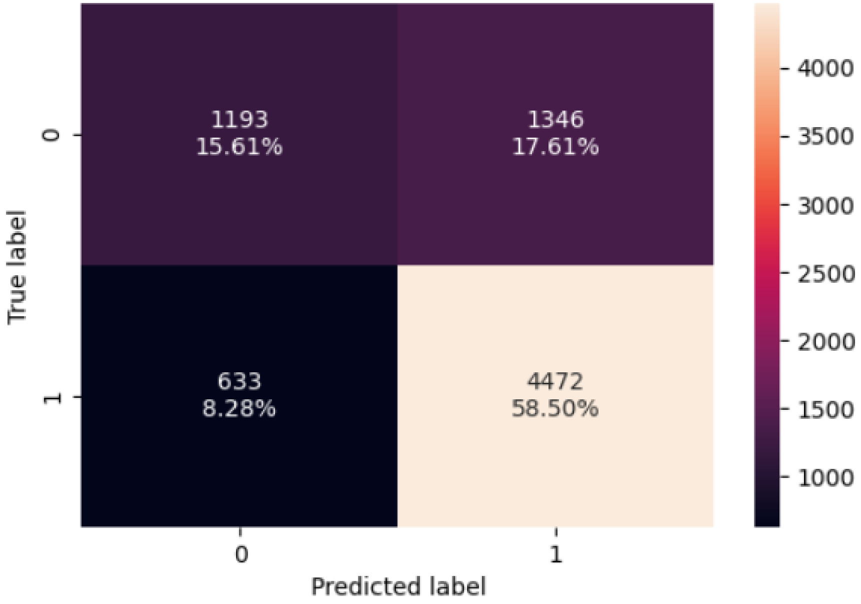
	Accuracy	Recall	Precision	F1
0	0.734301	0.885015	0.757799	0.816481

Results – In this model, we see almost no overfitting (the smallest so far), and good result in the F1 score.

Adaboost Classifier– Hyperparameter Tuned

Adaboost Performance Summary – Hyperparameters Tuned

(max_depth=3, random_state=1)



	Accuracy	Recall	Precision	F1
0	0.754429	0.883908	0.778443	0.82783

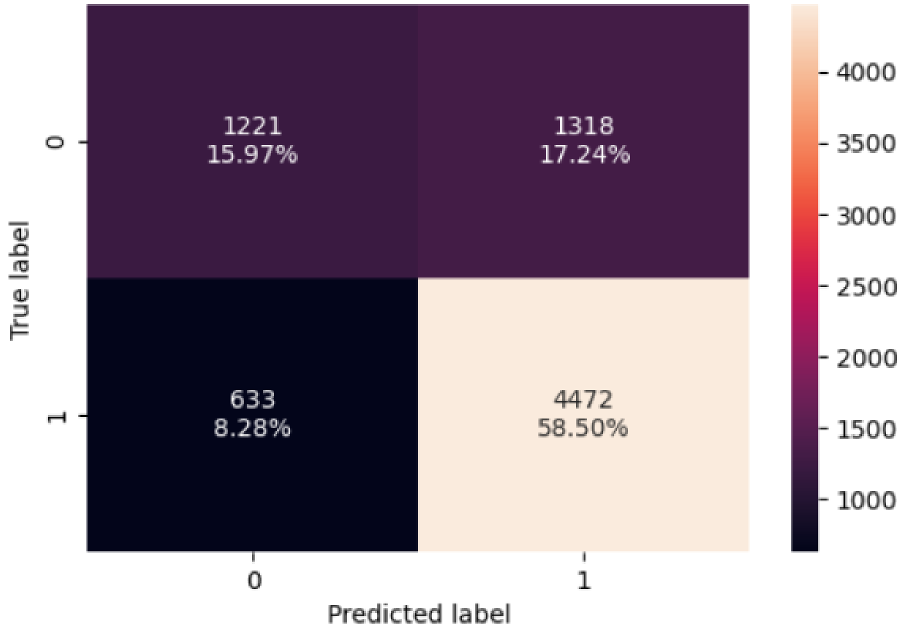
	Accuracy	Recall	Precision	F1
0	0.741104	0.876004	0.768649	0.818823

Results – Hyperparameters tuning gives us slightly better results. Small overfitting and insignificant improvement in the testing F1 score

Gradient Boosting Classifier Model

Gradient Boost Classifier Performance Summary – Default Parameters

(random_state=1)



Training performance:

	Accuracy	Recall	Precision	F1
0	0.758802	0.88374	0.783042	0.830349

Testing performance:

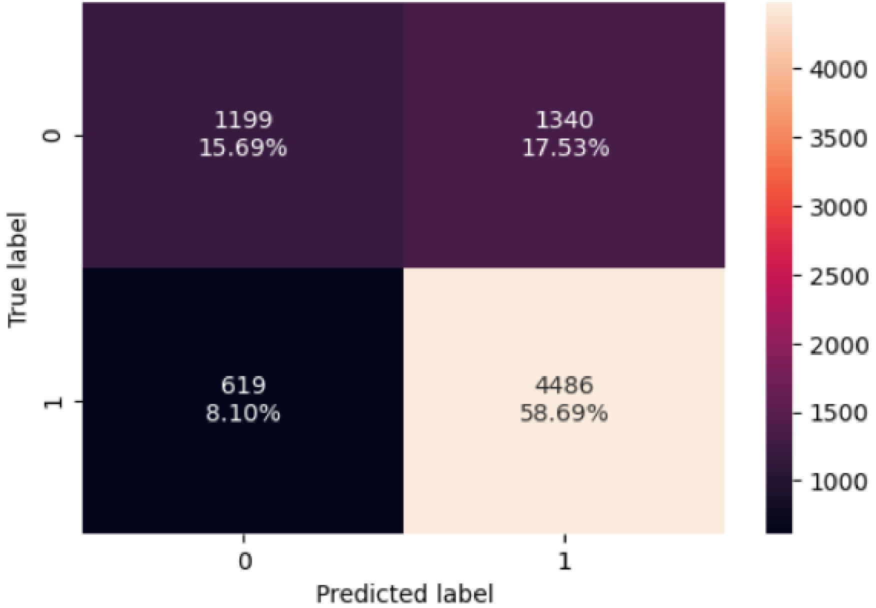
	Accuracy	Recall	Precision	F1
0	0.744767	0.876004	0.772366	0.820927

Results – The Gradient Boost model show strong results, with very low overfitting between training and testing sets.

Gradient Boosting Classifier–Hyperparameter Tuned

Gradient Boosting Performance Summary – Hyperparameters Tuned

(init=AdaBoostClassifier(random_state=1), random_state=1)



Training performance:

	Accuracy	Recall	Precision	F1
0	0.756167	0.885251	0.779568	0.829055

Testing performance:

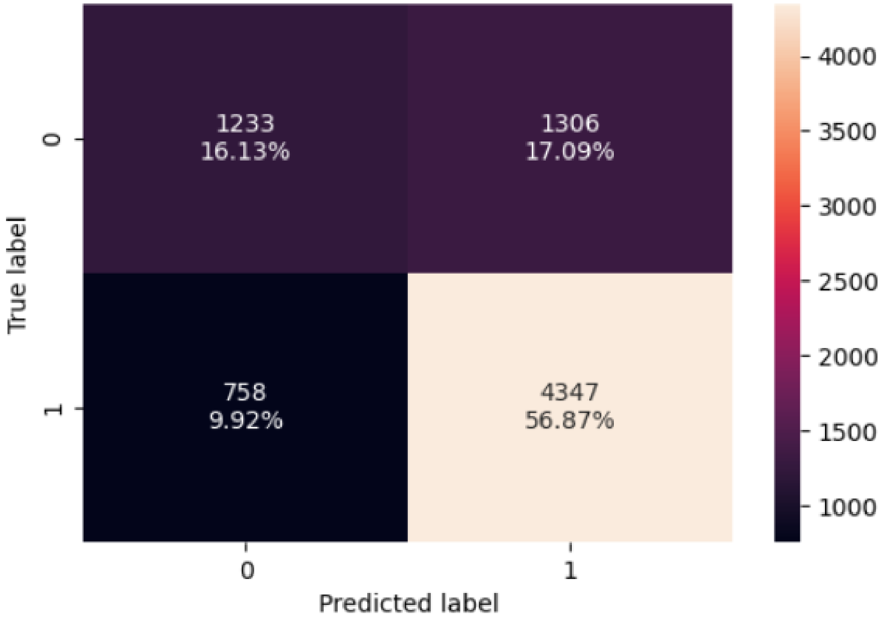
	Accuracy	Recall	Precision	F1
0	0.743721	0.878746	0.769997	0.820785

Results – After tuning the hyper parameters, we see even less overfitting (in the F1 score), but also a small (insignificant) reduction in the score (less than 0.1%)

XG Boosting Classifier Model

XG Boosting Classifier Performance Summary – Default Parameters

(random_state=1, eval_metric='logloss')



Training performance:

	Accuracy	Recall	Precision	F1
0	0.850807	0.935952	0.854537	0.893394

Testing performance:

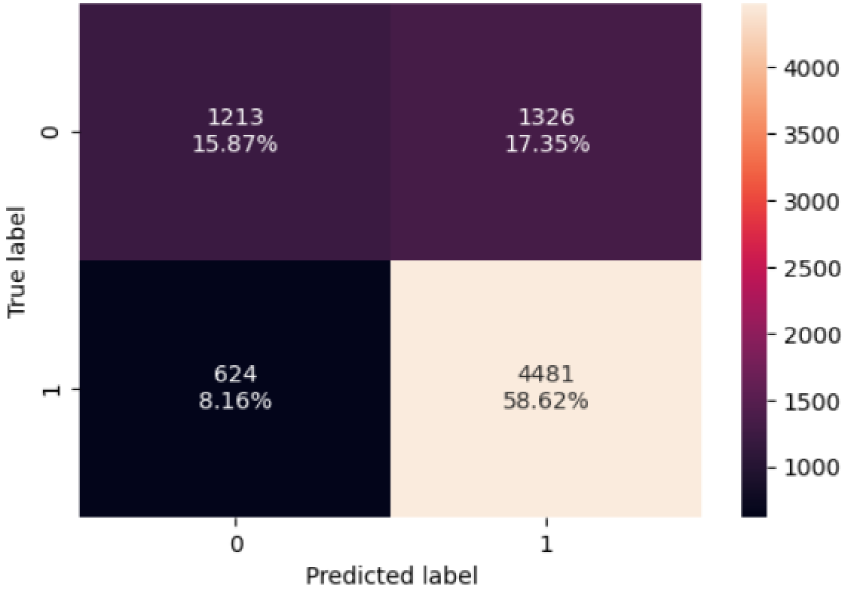
	Accuracy	Recall	Precision	F1
0	0.729984	0.851518	0.768972	0.808143

Results – The default run of the XGBoost model shows higher overfitting than the other boosting models, and the F1 scores are the lowest.

XG Boosting Classifier– Hyperparameter Tuned

Gradient Boosting Performance Summary – Hyperparameters Tuned

(eval_metric='logloss', gamma=3, learning_rate=0.05, n_estimators=50, random_state=1)



Training performance:

	Accuracy	Recall	Precision	F1
0	0.76211	0.888189	0.784243	0.832986

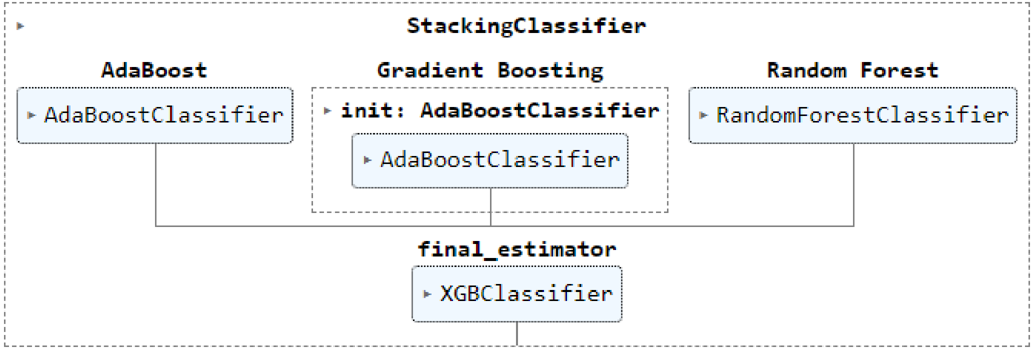
Testing performance:

	Accuracy	Recall	Precision	F1
0	0.744898	0.877767	0.771655	0.821298

Results – After tunning, the F1 score results are the best of all the models, with low overfitting.

Stacking Classifier Model

Stacking Classifier Performance Summary – Default Parameters




```
Training performance:
   Accuracy    Recall  Precision        F1
0  0.764745  0.887182   0.787497  0.834373
Testing performance:
   Accuracy    Recall  Precision      F1
0  0.742282  0.873653   0.770959  0.8191
```

Results – The stacking classifier shows good results, with a low overfitting. However, despite all the different classifiers involved, the F1 score is not as high as when we ran the XG Boost model (by 0.1%).

Models Performance Summary

Great Learning
POWER AHEAD

Training performance

	Decision Tree	Tuned Decision Tree	Bagging Classifier	Tuned Bagging Classifier	Random Forest	Tuned Random Forest	Adaboost Classifier
Accuracy	1.0	0.712548	0.985198	0.996187	0.999944	0.769119	0.738226
Recall	1.0	0.931923	0.985982	0.999916	0.999916	0.918660	0.887182
Precision	1.0	0.720067	0.991810	0.994407	1.000000	0.776556	0.760688
F1	1.0	0.812411	0.988887	0.997154	0.999958	0.841652	0.819080

Testing performance

	Decision Tree	Tuned Decision Tree	Bagging Classifier	Tuned Bagging Classifier	Random Forest	Tuned Random Forest	Adaboost Classifier
Accuracy	0.664835	0.706567	0.691523	0.724228	0.720827	0.738095	0.734301
Recall	0.742801	0.930852	0.764153	0.895397	0.832125	0.898923	0.885015
Precision	0.752232	0.715447	0.771711	0.743857	0.768869	0.755391	0.757799
F1	0.747487	0.809058	0.767913	0.812622	0.799247	0.820930	0.816481

Final Results: The best performance was received by the default decision tree, with a perfect score. However, it comes with the price of overfitting. The best performance regarding overfitting is the Adaboost classifier.

Models Performance Summary – Cont.

Training performance

	Tuned Adaboost Classifier	Gradient Boost Classifier	Tuned Gradient Boost Classifier	XGBoost Classifier	XGBoost Classifier Tuned	Stacking Classifier
Accuracy	0.754429	0.758802	0.756167	0.850807	0.762110	0.764745
Recall	0.883908	0.883740	0.885251	0.935952	0.888189	0.887182
Precision	0.778443	0.783042	0.779568	0.854537	0.784243	0.787497
F1	0.827830	0.830349	0.829055	0.893394	0.832986	0.834373

Testing performance

	Tuned Adaboost Classifier	Gradient Boost Classifier	Tuned Gradient Boost Classifier	XGBoost Classifier	XGBoost Classifier Tuned	Stacking Classifier
Accuracy	0.741104	0.744767	0.743721	0.729984	0.744898	0.742282
Recall	0.876004	0.876004	0.878746	0.851518	0.877767	0.873653
Precision	0.768649	0.772366	0.769997	0.768972	0.771655	0.770959
F1	0.818823	0.820927	0.820785	0.808143	0.821298	0.819100

Final Results: The overall best performance was received by the tuned XGBoost model, with the highest F1 score on the test set, with a small amount of overfitting.

Great Learning

Happy Learning !