EasyVisa

Ensemble Techniques

Mona Desai
Date : 01/13/2023

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Model Performance Summary
- Appendix

Executive Summary

- **Conclusions, Actionable insights**
    - The number of Visa denials seems high at ~33%.
    - education_of_employee with just high school gets the denied more than other qualifications
    - has_job_experience get certified ~19% more than those who don't have
    - Below list of fields have higher impact on Visa approval process:
        - education_of_employee, has_job_experience, region_of_employment, prevailing_wage, unit_of_wage
    - Most of the data did not give any significant impact to show the reason for cancellation
    - The best regression Model seems to be providing us prediction with 0.84 F1 score
    - **Stacking Classifier** and **Tuned Gradient Boost**, **Tuned XGBoost** and **Tuned Random** forest are the best fit model

- **Recommendations**
    - OFLC should consider applications which has job_experience, has yearly and higher prevailing_wage
    - Adding a field to generalize the value of the unit_of_wages
    - OFLC processed 775,979 applications, whereas only 25480 data is given to analyze. OFLC could provide more data in order to get more accurate prediction.

Business Problem Overview and Solution Approach

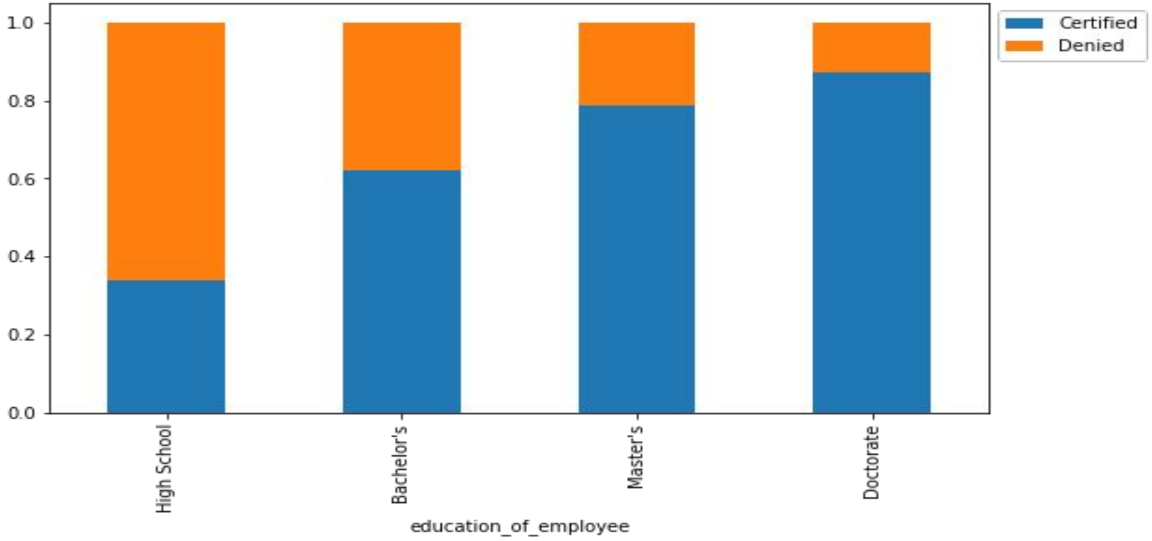
- **Problem**
  - Solve high rate of visa denials; Predict probability of visa approvals
  - Finding the best fit model to predict visa certification with high accuracy and repeatability
- **Solution approach / methodology**
  - Exploratory Data Analysis
    - Overview of dataset, Univariate analysis, Bivariate analysis
  - Model Building - Decision Tree
    - Model building and Hyperparameter Tuning
    - Bagging classifier
    - Random Forest
    - Boosting classifier - AdaBoost, Gradient Boosting , XGBoost
    - Stacking Classifier

EDA Results

- **Key results from EDA**
    - Data contains information on Visa approval status
    - 25480 rows and 12 columns
    - 3 numerical value fields and nine categorical fields in the dataset
    - No duplicate values found
    - no_of_employee field has negative value. Treat this to make it an absolute value
    - Average prevailing_wage is 74445 and the max wage is 319210 with 50% wage is ~ 70,000, assuming in USD.
- **Observation from Univariate Analysis**
    - 176 rows have unit_of_wage='Hour' and have <100 wage.
    - Average prevailing_wage of an applicant is ~75000
    - The most applicants are from the continent “Asia” ~66%
    - ~80% applicants holds Bachelor's and Master’s degrees
    - ~90% applicants seems to requires_job_training
    - Most of the applicant’s unit_of_wage is on yearly base
    - ~67% applicants visa seems to get certified, 33% get rejected which is high.

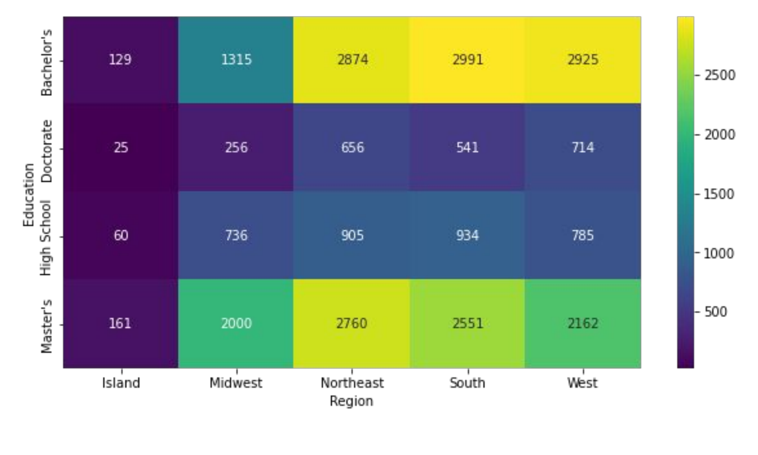
EDA Results continued..

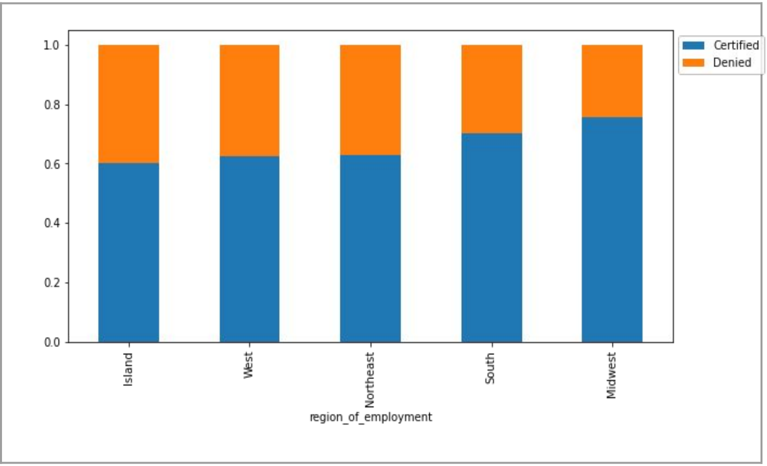
- **Insights and Observation from Bivariate Analysis**
    - There is no strong correlations between no_of_employees, yr_of_estab and prevailing_wage
    - ~65% certified applicants has_job_experience Vs. 35% don't.
    - ~87% applicants having "Doctorate" education get Certified Vs. ~33% applicants with the High School degree
    - ~21% Master's degree get Denied Vs. ~40% Bachelor's degree applicants



EDA Results continued..

- **Insights and Observation from Bivariate Analysis continued..**
  - Midwest and Island requires more Master’s education than Bachlore’s Vs. Northest, South and West regions has more demand of Bachelor's
  - Applicants applying for Midwest Certified the most.

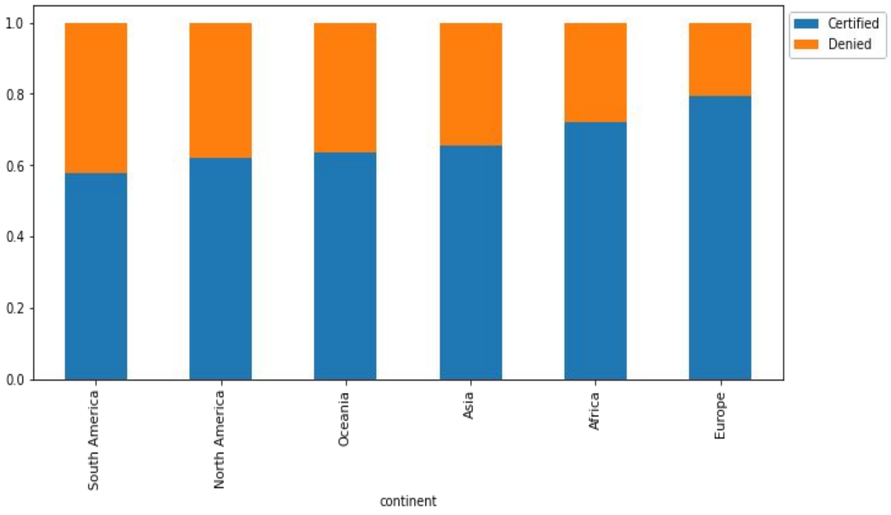




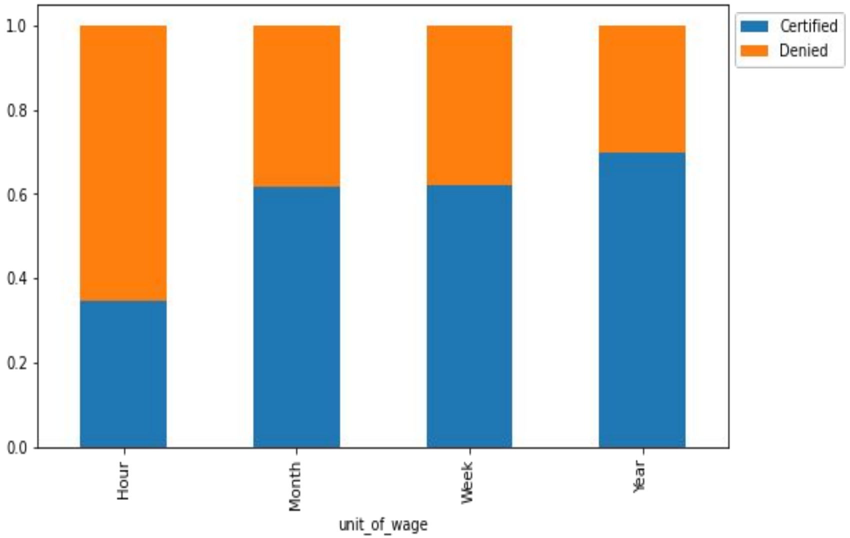
EDA Results continued..

- **Insights and Observation from Bivariate Analysis continued..**

Applicants from Europe certified the most and South America the least.
Africa is the second most to get certified

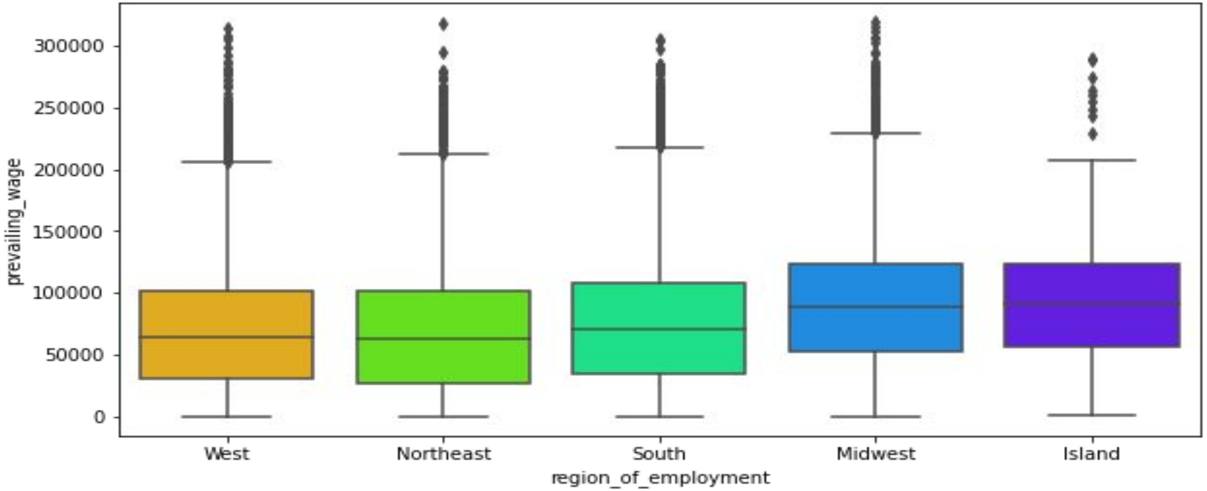


Applications mentioning yearly wage are certified more than hourly
Weekly and Monthly seems to get certified equally



EDA Results continued..

- **Insights and Observation from Bivariate Analysis continued..**
  - West and Northeast regions has an average 60000 prevailing_wage Vs. Midwest and Island has an average 90000



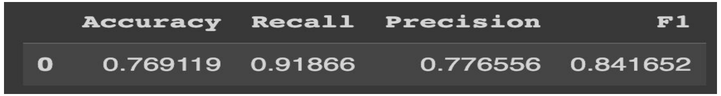
  - There is not much impact of prevailing_wage on case_status
    - Denied has average of ~63000 prevailing_wage Vs. ~70000 for certified

Data Preprocessing

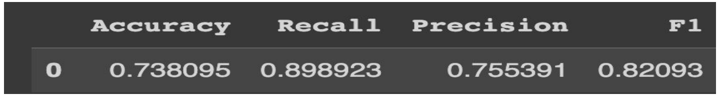
- **Duplicate value check & Missing value treatment**
  - No missing data or any duplicate values
- **Outlier check (treatment if needed)**
  - All three numerical columns has outliers, however we will not treat them as they are proper values
- **Feature engineering**
  - We drop the column with the unique “case_id” because it will not have impact on the dependent variable
- **Data preparation for modeling**
  - “case_status” is the dependent categorical variable.
  - We encoded certified status to “1” and denial to “0” under the column “case_status”
  - encode “categorical” data
  - To build the model on the train set, split the data into train and test sets in 70:30 part
  - Build functions to calculate different metrics and confusion matrix to use the same code for each model

Model Performance Summary

- **Overview of final ML model and its parameters**
    - **Stacking Classifier** and **Tuned Gradient Boost**, **Tuned XGBoost** and **Tuned Random forest** are the best fit model
    - However **Tuned Random Forest** can be chosen as a final Model
    - Scores for train and tests sets



	Accuracy	Recall	Precision	F1
0	0.769119	0.91866	0.776556	0.841652



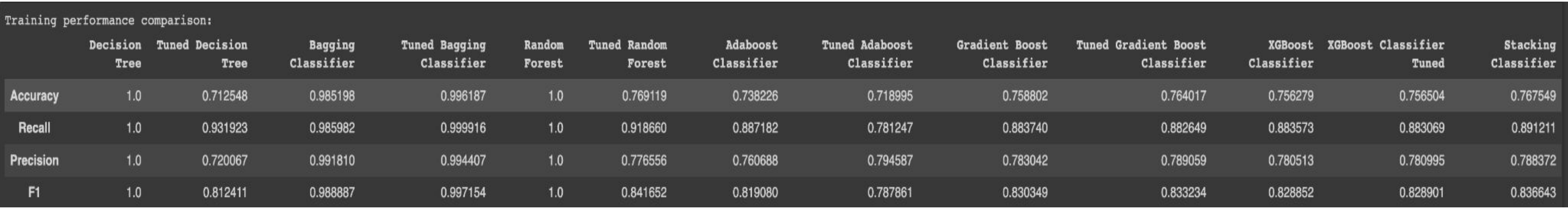
	Accuracy	Recall	Precision	F1
0	0.738095	0.898923	0.755391	0.82093

    - F1 score for training set is 0.84 which is the heist amongst all models
    - The Recall score is very high as well

- **Summary of most important factors used by the ML model for prediction**
    - Education_of_the employee is the most important features in order to identify if the application should be certified or denied
    - Followed by has_job_experience and the prevailing_wage.
    - Employees applying for the job in the midwest shows some importance too
    - Application from the Europe continent is an important factor

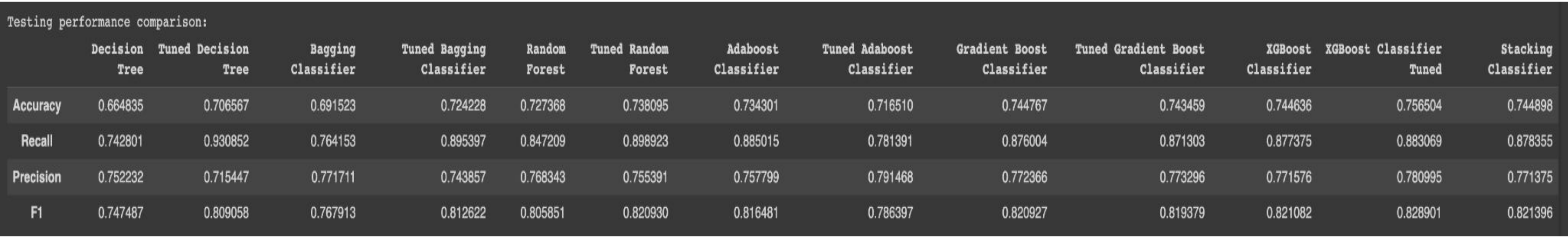
Model Performance Summary continued..

- **Summary of key performance metrics for training and test data in tabular format for comparison**



Training performance comparison:

	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	0.712548	0.985198	0.996187	1.0	0.769119	0.738226	0.718995	0.758802	0.764017	0.756279	0.756504	0.767549
Recall	1.0	0.931923	0.985982	0.999916	1.0	0.918660	0.887182	0.781247	0.883740	0.882649	0.883573	0.883069	0.891211
Precision	1.0	0.720067	0.991810	0.994407	1.0	0.776556	0.760688	0.794587	0.783042	0.789059	0.780513	0.780995	0.788372
F1	1.0	0.812411	0.988887	0.997154	1.0	0.841652	0.819080	0.787861	0.830349	0.833234	0.828852	0.828901	0.836643



Testing performance comparison:

	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	0.706567	0.691523	0.724228	0.727368	0.738095	0.734301	0.716510	0.744767	0.743459	0.744636	0.756504	0.744898
Recall	0.742801	0.930852	0.764153	0.895397	0.847209	0.898923	0.885015	0.781391	0.876004	0.871303	0.877375	0.883069	0.878355
Precision	0.752232	0.715447	0.771711	0.743857	0.768343	0.755391	0.757799	0.791468	0.772366	0.773296	0.771576	0.780995	0.771375
F1	0.747487	0.809058	0.767913	0.812622	0.805851	0.820930	0.816481	0.786397	0.820927	0.819379	0.821082	0.828901	0.821396

- Stacking Classifier, Tuned Gradient Boost, Tuned XGBoost and Tuned Random forest are the best fit model
- Tuned Decision Tree and Gradient Boost Classifier model fits well too

Model Building - Bagging and Boosting

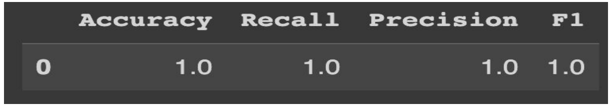
- **Model building steps**
  - Bagging models- Decision Tree, Bagging Classifier and Random Forest
  - Boosting model - AdaBoost, Gradient Boosting, XGBoost and
  - Stacking model
    - Both cases “FP” and “FN” are important
      - Model predicts that the visa application will get certified but in reality, the visa application should get denied
      - Model predicts that the visa application will not get certified but in reality, the visa application should get certified.
  - Use F1 score to evaluate the model, greater the F1 score higher are the chances of minimizing False Negatives and False Positives.
  - model_performance_classification_sklearn function will be used to check the model performance of each model for train and test data
  - confusion_matrix_sklearn function will be used to plot the confusion matrix for each model for train and test data
  - Check all the model performance after defining the classifiers and setting the Hyperparameters for each different models

Model Building - Bagging

- Decision Tree and Bagging Model performance comparison
  - Decision Tree model performance on Training and testing set before and after Hyperparameter Tuning
  - Hyperparameters: max_depth, min_sample_leaf,Max_leaf_nodes,min_impurity_decrease

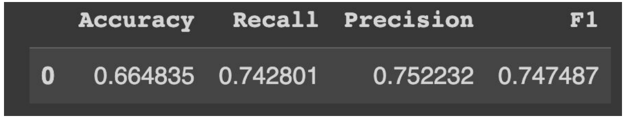
Decision Tree

Train set



	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0

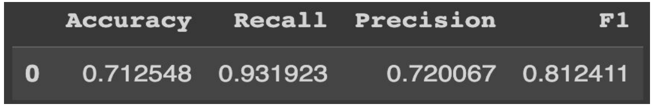
Test Set



	Accuracy	Recall	Precision	F1
0	0.664835	0.742801	0.752232	0.747487

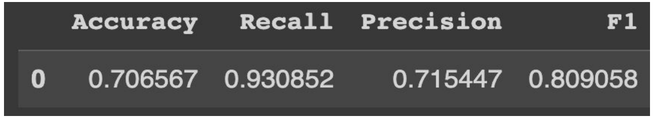
Hyperparameter Tuning - Decision Tree

Train Set



	Accuracy	Recall	Precision	F1
0	0.712548	0.931923	0.720067	0.812411

Test Set



	Accuracy	Recall	Precision	F1
0	0.706567	0.930852	0.715447	0.809058

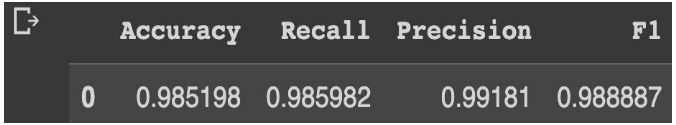
- The Decision Tree is overfitting the data
- After hyperparameter tuning the performance has improved
- f1 score has increased for test data and is very comparable for both train and test sets

Model Building - Bagging continued...

- Bagging Classifier performance on Training and Testing sets before and after Hyperparameter Tuning
- Hyperparameters:max_samples,max_features,n_estimators

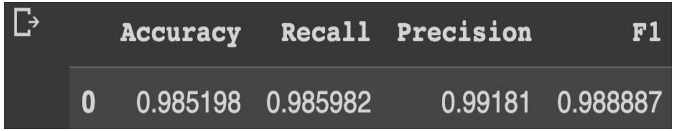
<u>Bagging Classifier</u>

Train Set



	Accuracy	Recall	Precision	F1
0	0.985198	0.985982	0.99181	0.988887

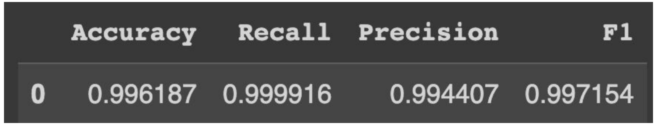
Test Set



	Accuracy	Recall	Precision	F1
0	0.985198	0.985982	0.99181	0.988887

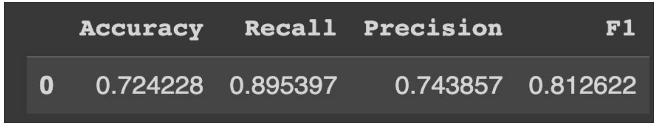
<u>Hyperparameter Tuning- Bagging Classifier</u>

Train Set



	Accuracy	Recall	Precision	F1
0	0.996187	0.999916	0.994407	0.997154

Test Set



	Accuracy	Recall	Precision	F1
0	0.724228	0.895397	0.743857	0.812622

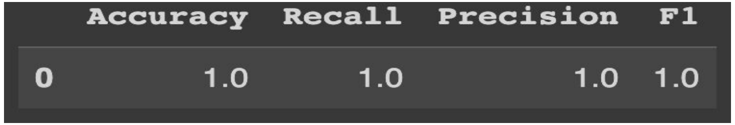
- Bagging Classifier is overfitting both training and testing data
- After hyperparameter tuning f1 score for testing data has reduced a lot. surprisingly, the model performance as decreased

Model Building - Bagging continued...

- Random Forest performance on Training and Testing sets before and after Hyperparameter Tuning
- Hyperparameters : max_depth, max_fetures, min_sample_splits, n_estimators

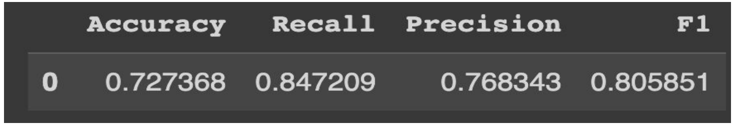
Random Forest classifier

Train set



	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0

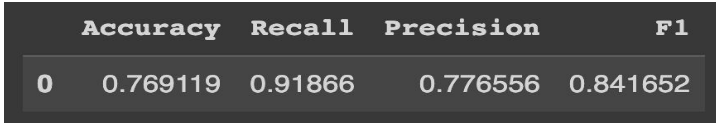
Test Set



	Accuracy	Recall	Precision	F1
0	0.727368	0.847209	0.768343	0.805851

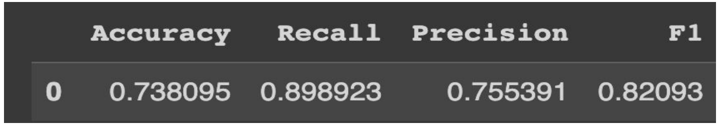
Hyperparameter Tuning-Random Forest

Train Set



	Accuracy	Recall	Precision	F1
0	0.769119	0.91866	0.776556	0.841652

Test Set



	Accuracy	Recall	Precision	F1
0	0.738095	0.898923	0.755391	0.82093

- Clearly, the random forest classifier is overfitting the training set
- Overfitting has reduced significantly and overall performance has improved after hyperparameter tuning, Recall has improved too on test data

Model Improvement - Bagging

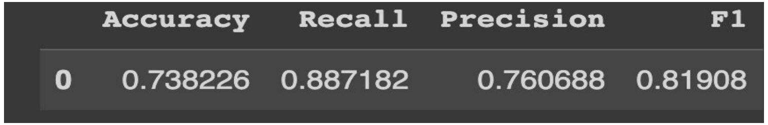
- Improvement in the model performance by hyperparameter tuning
  - Decision Tree
    - The Decision Tree is overfitting the data.After hyperparameter tuning the performance has improved
    - max_depth, min_sample_leaf,Max_leaf_nodes,min_impurity_decrease hyperparameters are used
    - f1 score reduced for train set -0.81 and
    - has increased for test data - 0.81 and is very comparable for both train and test sets
  - Bagging
    - Bagging Classifier is overfitting both training and testing data
    - After hyperparameter tuning f1 score for testing data has reduced a lot. surprisingly, the model performance as decreased
    - Max_samples,max_features,n_estimators hyperparameters are used
    - F1 score has reduced from 0.98 to 0.81 after tuning on test data
  - Random forest
    - Random forest classifier is overfitting the training set
    - Overfitting has reduced significantly and overall performance has improved after hyperparameter tuning, Recall has improved too on test data
    - max_depth, max_fetures, min_sample_splits, n_estimatorshyperparameters are used
    - F1 score are 0.84 and 0.82 for train and test sets respectively after hypertuning
  - f1 score **0.81 and 0.81** for train and test **improved** for Decision Tree after hyperparameter tuning
  - f1 score **0.99 and 0.81** for train and test, test score has **decreased** after hyperparameter tuning in Bagging
  - f1 score **0,84 and 0.82** for train and test, overfitting has reduced and performance has improved after hyperparameter in Random Forest

Model Building - Boosting

- Boosting Models performance comparison
  - Adaboost Classifier and Hypertuning performance on training and testing sets
  - Hyperparameters: base_estimator,n_estimators, learning_rate

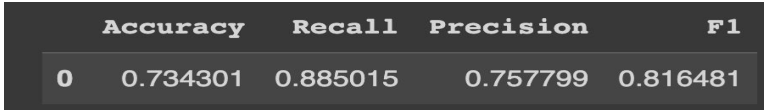
Adaboost Classifier

Train set



	Accuracy	Recall	Precision	F1
0	0.738226	0.887182	0.760688	0.81908

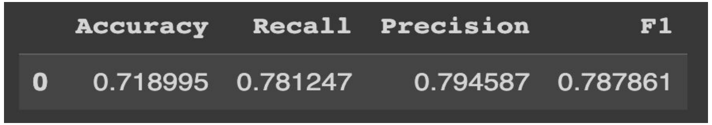
Test Set



	Accuracy	Recall	Precision	F1
0	0.734301	0.885015	0.757799	0.816481

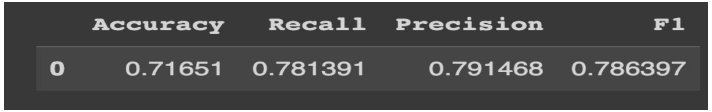
AdaBoost - Hypertuning

Train set



	Accuracy	Recall	Precision	F1
0	0.718995	0.781247	0.794587	0.787861

Test Set



	Accuracy	Recall	Precision	F1
0	0.71651	0.781391	0.791468	0.786397

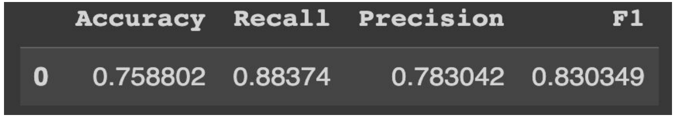
- AdaBoost seems to give good performance on both train and test sets.
- However the performance has decreased after the Hypertuning, yet the scores are very comparable on both test and train sets

Model Building - Boosting continued..

- Gradient Boosting Classifiers and Hypertuning performance on train and test sets
- Hyperparameters:n_estimators, subsamples,max_features

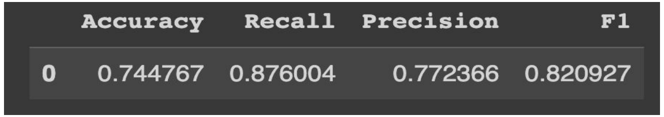
Gradient Boosting Classifiers

Train set



	Accuracy	Recall	Precision	F1
0	0.758802	0.88374	0.783042	0.830349

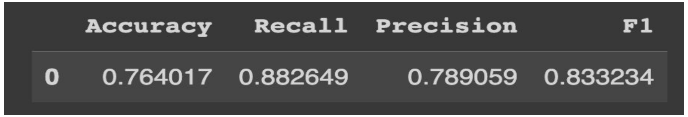
Test set



	Accuracy	Recall	Precision	F1
0	0.744767	0.876004	0.772366	0.820927

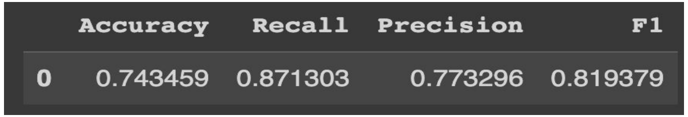
Gradient Boosting-Hyperparameter Tuning

Train set



	Accuracy	Recall	Precision	F1
0	0.764017	0.882649	0.789059	0.833234

Test set



	Accuracy	Recall	Precision	F1
0	0.743459	0.871303	0.773296	0.819379

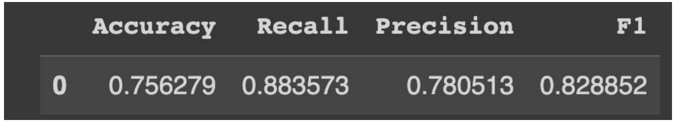
- Gradient boosting classifier performs well on both train and test data
- The performance is almost same after the Hyperparameter Tuning

Model Building - Boosting continued..

- XGBoost Classifier and Hypertuning performance on train and test sets
- Hyperparameters:n_estimators,scale_pos_weight,subsample,learning_rate,colsample_bytree,colsample_bylevel

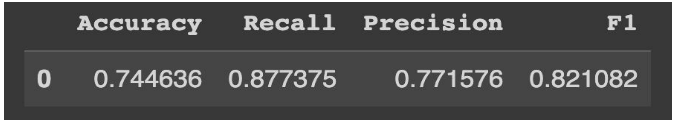
XGBoost Classifier

Train set



```
   Accuracy    Recall  Precision        F1
0  0.756279  0.883573   0.780513  0.828852
```

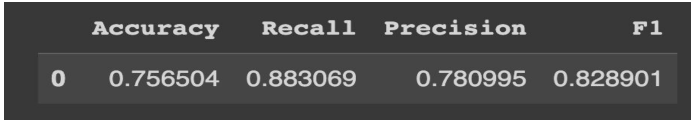
Test set



```
   Accuracy    Recall  Precision        F1
0  0.744636  0.877375   0.771576  0.821082
```

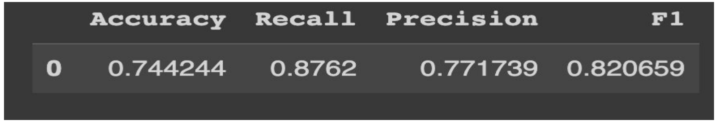
XGBoost-Hypertuning

Train set



```
   Accuracy    Recall  Precision        F1
0  0.756504  0.883069   0.780995  0.828901
```

Test set



```
   Accuracy  Recall  Precision        F1
0  0.744244  0.8762   0.771739  0.820659
```

- XGBoost classifier is also performing well on both the data sets
- The performance is almost unchanged after the Hyperparameter tuning

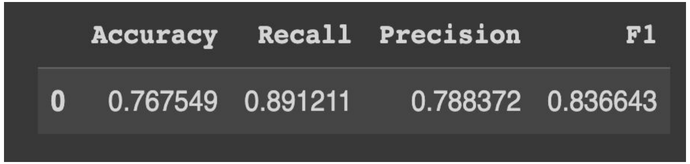
Model Improvement - Boosting

- Improvement in the model performance by hyperparameter tuning
  - AdaBoost
    - AdaBoost seems to give good performance on both train and test sets.
    - However the performance has decreased after the Hypertuning, yet the scores are very comparable on both test and train sets
    - base_estimator,n_estimators, learning_rate hyperparameters are used
  - Gradient Boosting
    - Gradient boosting classifier performs well on both train and test data.The performance is almost same after the Hyperparameter Tuning
    - n_estimators, subsamples,max_features hyperparameters are used
  - XGBoost
    - XGBoost classifier is also performing well on both the data sets.The performance is almost unchanged after the Hyperparameter tuning
    - N_estimators,scale_pos_weight,subsample,learning_rate,colsample_bytree,colsample_bylevel hyperparameter are used
  - f1 score is **0.78 and 0.78** for train and test, which has decreased after hyperparameter tuning in AdaBoost
  - f1 score is **0.83 and 0.82** for train and test, which is almost unchanged after hyperparameter tuning in Gradient Boosting
  - f1 score is **0.83 and 0.82** for train and test, which are same before and after hyperparameter tuning in XGBoost

Model Building-Stacking

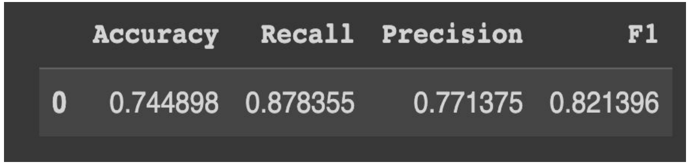
- Stacking Classifier
  - Define the Stacking classifier using AdaBoost, Gradient Boosting and Random Forest
  - Compare the performance on Train and test sets

Train set



	Accuracy	Recall	Precision	F1
0	0.767549	0.891211	0.788372	0.836643

Test Set



	Accuracy	Recall	Precision	F1
0	0.744898	0.878355	0.771375	0.821396

  - The Stacking Classifier is giving smiler performance
  - The confusion matrix shows that the model can predict whether to certified or denial both well.