## IBM\_Attrition\_prediction:

<https://www.kaggle.com/code/karthikeyanscs21m028/ibm-attrition-prediction-with-grid-search/edit>

Testing model **Logistic Regression**

Cross\_validation mean:0.8493946275642911,Training Accuracy:0.8852040816326531,Test Accuracy:0.8435374149659864

              precision    recall  f1-score   support

Not Resigned       0.85      0.98      0.91       236

    Resigned       0.77      0.29      0.42        58

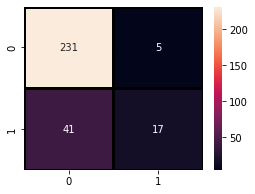
    accuracy                           0.84       294

   macro avg       0.81      0.64      0.67       294

weighted avg       0.83      0.84      0.81       294

[[231   5]

 [ 41  17]]



Testing model **Random Forest Classifier**

Cross\_validation mean:0.8121606558787546,Training Accuracy:1.0,Test Accuracy:0.8197278911564626

              precision    recall  f1-score   support

Not Resigned       0.82      0.99      0.90       236

    Resigned       0.73      0.14      0.23        58

    accuracy                           0.82       294

   macro avg       0.78      0.56      0.56       294

weighted avg       0.80      0.82      0.77       294

[[233   3]

 [ 50   8]]

Testing model **Gradient Boosting Classifier**

Cross\_validation mean:0.8509214057600646,Training Accuracy:0.9591836734693877,Test Accuracy:0.8333333333333334

              precision    recall  f1-score   support

Not Resigned       0.85      0.96      0.90       236

    Resigned       0.67      0.31      0.42        58

    accuracy                           0.83       294

   macro avg       0.76      0.64      0.66       294

weighted avg       0.81      0.83      0.81       294

[[227   9]

 [ 40  18]]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F1 Score | MCC |
| Logistic regression | 0.8435 | 0.9788 | 0.8493 | 0.9094 |  |
| Random Forest | 0.8197 | 0.9873 | 0.8233 | 0.8979 |  |
| Gradient Boosting Classifier | 0.8333 | 0.9619 | 0.8502 | 0.9026 |  |
|  |  |  |  |  |  |

Logistic regression:

param\_grid={'C': (0.2, 0.3, 0.35, 0.45, 0.55),

                         'fit\_intercept': ('True', 'False')})

Best parameters: {'C': 0.3, 'fit\_intercept': 'True'}

Best score: 0.870739271547061

Random forest:

Best Parameters {'criterion': 'gini', 'max\_depth': 5, 'n\_estimators': 1000}

Best score 0.8554453660295709

Overall

LogisticRegression() Best Parameters {'C': 0.3, 'fit\_intercept': 'True', 'solver': 'liblinear'}

LogisticRegression() Best score 0.870739271547061

https://www.kaggle.com/code/karthikeyanscs21m028/predict-employee-retention-using-random-forest/edit

Standard RF

|  |  |  |  |
| --- | --- | --- | --- |
| **accuracy** | **precision** | **recall** | **f1** |

|  |  |  |  |
| --- | --- | --- | --- |
| 0.880435 | 0.75 | 0.125 | 0.214286 |

RF with SMOTE:

0.877717 0.571429 0.25 0.347826

accuracy precision recall f1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Standard RFC** | 0.880435 | 0.750000 | 0.125000 | 0.214286 |
| **RFC with SMOTE** | 0.877717 | 0.571429 | 0.250000 | 0.347826 |
| **RFC with SMOTE & Hyperparameter Tuning** | 0.872283 | 0.520000 | 0.270833 | 0.356164 |

* Random Forest Classifier (rfc) model with SMOTE has relatively better score in accuracy, recall and f1 compared to standard rfc and rfc with SMOTE & Hyperparameter tuning
* rfc model will perform better with SMOTE without adding any hyperparameter tuning
* Still, the model performance score wasn't good enough. We will trying use another methods to improve model performance in the futur

XGboost:

XGBoost Classifier 0.8299319727891157

              precision    recall  f1-score   support

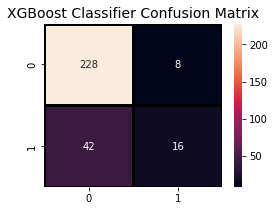
           0       0.84      0.97      0.90       236

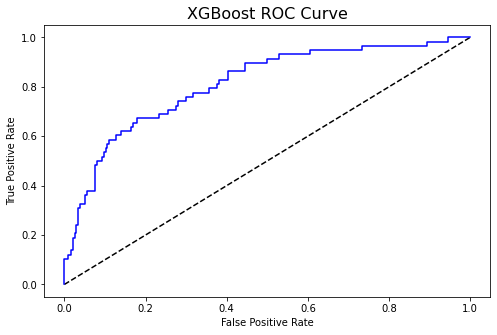
           1       0.67      0.28      0.39        58

    accuracy                           0.83       294

   macro avg       0.76      0.62      0.65       294

weighted avg       0.81      0.83      0.80       294





Light gbm:

Light GBM Classifier 0.8333333333333334

              precision    recall  f1-score   support

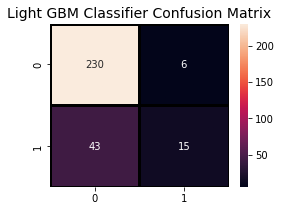
           0       0.84      0.97      0.90       236

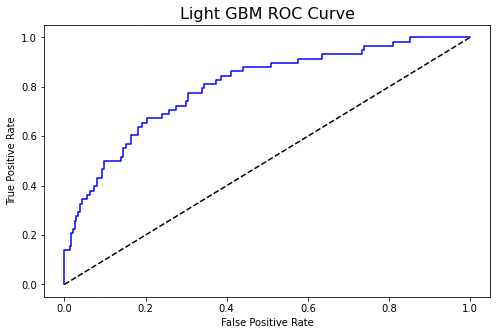
           1       0.71      0.26      0.38        58

    accuracy                           0.83       294

   macro avg       0.78      0.62      0.64       294

weighted avg       0.82      0.83      0.80       294





KNN:

K-Nearest Neighbors Classifier 0.8095238095238095

              precision    recall  f1-score   support

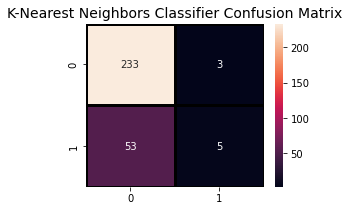
           0       0.81      0.99      0.89       236

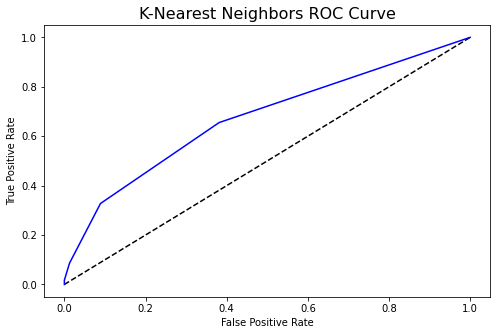
           1       0.62      0.09      0.15        58

    accuracy                           0.81       294

   macro avg       0.72      0.54      0.52       294

weighted avg       0.78      0.81      0.75       294





SVC:

Support Vector Classifier 0.8231292517006803

              precision    recall  f1-score   support

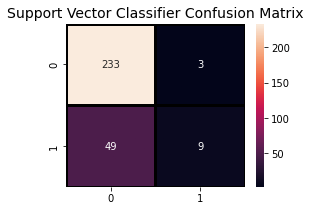
           0       0.83      0.99      0.90       236

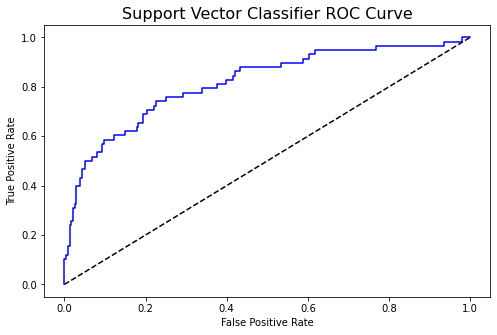
           1       0.75      0.16      0.26        58

    accuracy                           0.82       294

   macro avg       0.79      0.57      0.58       294

weighted avg       0.81      0.82      0.77       294





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithms | Accuracy | Precision | Recall | F1 Score | MCC |
| Logistic regression | 0.8435 | 0.9788 | 0.8493 | 0.9094 | 0.31 |
| Random forest | 0.8197 | 0.9873 | 0.8233 | 0.8979 | 0.37 |
| Gradient Boosting Classifier | 0.8333 | 0.9619 | 0.8502 | 0.9026 | 0.41 |
| XGboost | 0.82 | 0.96 | 0.84 | 0.90 | 0.35 |
| LightGBM | 0.83 | 0.97 | 0.84 | 0.90 | 0.36 |
| KNN | 0.80 | 0.98 | 0.81 | 0.89 | 0.17 |
| SVC | 0.82 | 0.98 | 0.82 | 0.89 | 0.28 |