KNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ROC Curve | Precision | Recall | F1 Score |
| Before Optimization | 0.851 | 0.74 | 0.83 | 0.78 |
| After Optimization | 0.955 | 0.83 | 0.97 | 0.89 |

SVC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ROC Curve | Precision | Recall | F1 Score |
| Before Optimization | 0.851 | 0.74 | 0.83 | 0.78 |
| After Optimization | 0.942 | 0.85 | 0.93 | 0.89 |

LR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ROC Curve | Precision | Recall | F1 Score |
| Before Optimization | 0.892 | 0.8 | 0.83 | 0.81 |
| After Optimization | 0.892 | 0.8 | 0.83 | 0.81 |

Random Forest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ROC Curve | Precision | Recall | F1 Score |
| Before Optimization | 0.952 | 0.78 | 0.92 | 0.84 |
| After Optimization | 0.947 | 0.83 | 0.93 | 0.87 |

XGB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ROC Curve | Precision | Recall | F1 Score |
| Before Optimization | 0.989 | 0.94 | 0.95 | 0.94 |
| After Optimization | 0.989 | 0.94 | 0.95 | 0.94 |

LGBM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ROC Curve | Precision | Recall | F1 Score |
| Before Optimization | 0.973 | 0.89 | 0.95 | 0.91 |
| After Optimization | 0.973 | 0.89 | 0.95 | 0.91 |

Ada Boost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ROC Curve | Precision | Recall | F1 Score |
| Before Optimization | 0.964 | 0.89 | 0.9 | 0.9 |
| After Optimization | 0.964 | 0.89 | 0.9 | 0.9 |

By considering the F1 Score, we can select the model that achieves the highest F1 Score value. This ensures that the model performs well in terms of both precision and recall and provides a good balance between correctly identifying positive cases (stroke) and minimizing false positives.

After Optimizing KNN, SVC, Random Forest these algorithms F1 score changed. Others are unchanged. XGB produced the highest F1 score. We can choose the XGB model.

For optimization I have Used a grid search for hyperparameter tuning: Instead of using the default parameters for the algorithms, I have used grid search to find the optimal hyperparameters for the models. Grid search will try different combinations of hyperparameters and evaluate their performance using cross-validation. Then I compared the before and after result.