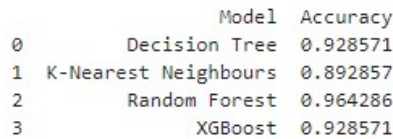
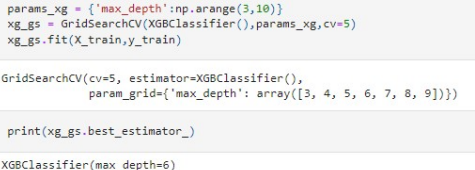


## Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID53287
Project Name	Project – Smart Lender
Maximum Marks	10 Marks

### Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	Accuracy Scores- Decision tree: 93% KNN :89% Random Forest:96% XGBoost:92%	 <pre> Model Accuracy 0 Decision Tree 0.928571 1 K-Nearest Neighbours 0.892857 2 Random Forest 0.964286 3 XGBoost 0.928571 </pre>
2.	Tune the Model	Hyperparameter Tuning - GridSearchCV Validation Method – Ensembling and Voting	 <pre> params_xg = {'max_depth':np.arange(3,10)} xg_gs = GridSearchCV(XGBClassifier(),params_xg,cv=5) xg_gs.fit(X_train,y_train)  GridSearchCV(cv=5, estimator=XGBClassifier(),              param_grid={'max_depth': array([3, 4, 5, 6, 7, 8, 9])})  print(xg_gs.best_estimator_)  XGBClassifier(max_depth=6) </pre>

```

estimators=[('knn', knn_model), ('rf', rf_model), ('dt', dt_model), ('xg', xg_model)]
ensemble = VotingClassifier(estimators, voting='hard')

ensemble.fit(X_train, y_train)

/opt/conda/envs/Python-3.10/lib/python3.10/site-packages/sklearn/ensemble/_forest.py:427: FutureWarning: 'max_features='auto'' has been deprecated in 1.1 and will
t' or remove this parameter as it is also the default value for RandomForestClassifiers and ExtraTreesClassifiers.
warn(
*
      knn      rf      dt      xg
> KNeighborsClassifier > RandomForestClassifier > DecisionTreeClassifier > XGBClassifier

ens_acc = ensemble.score(X_test, y_test)

comp_acc = pd.DataFrame({'Model':['Decision Tree','K-Nearest Neighbours','Random Forest','XGBoost','Ensemble'],'Accuracy':[dt_acc,knn_acc,rf_acc,xg_acc,ens_acc]})
print(comp_acc)

Model Accuracy
0 Decision Tree 0.928571
1 K-Nearest Neighbours 0.892857
2 Random Forest 0.964286
3 XGBoost 0.928571
4 Ensemble 0.928571

```