**Decision Tree**

Dataset: MNIST

Params: DecisionTreeClassifier(criterion='entropy', max\_depth=60, max\_features='auto',min\_samples\_split=7)

**Accuracy:** 0.769

**Bagging**

Dataset with Decision Tree(unstable): MNIST

Params: BaggingClassifier(base\_estimator=DecisionTreeClassifier(), bootstrap=False,max\_features=10, max\_samples=25, n\_estimators=100, random\_state=0)

**Accuracy:** 0.624

**Random** **Forest**

Dataset: MNIST

Params: RandomForestClassifier(bootstrap=False, max\_depth=20, max\_features='sqrt',min\_samples\_leaf=2, n\_estimators=500, n\_jobs=-1,

random\_state=0)

**Accuracy:** 0.963

**Gradient** **Boosting**

Dataset: MNIST

Params: GradientBoostingClassifier(criterion='mse', max\_depth=10, max\_features='log2',n\_estimators=50, random\_state=0)

**Accuracy:** 0.944

MNIST dataset gets highest generalization with **Random Forest classifier**. It is because Random Forest classifiers introduces a randomness into the iterations, by choosing a best feature out of randomly selected m features instead of trying to find best feature out of all the features. This randomness helps avoid overfitting and provides the ability to generalize better.