**Machine Learning- EX1**

**Abd el Rahman Mustafa- 322664939**

**Noor Edres -207031055**

**Results:**

**B)**

**Decision Tree Classification**

Grid Search Result: maxDipth=8 minSamples=2

My tree accurcy is: 0.9060509554140127

**Decision Tree Regression**

Grid Search Result: maxDipth=6 minSamples=2

My tree MSE is: 65230071096394.2

**AdaBoost Classification**

Grid Search Result: NEstimators=50

Adaboost accurcy is : 0.8586783439490446

**C)**

**Decision Tree Classification**

DecisionTreeClassifier(max\_depth=9, min\_samples\_split=5, random\_state=50)

Accuracy: 0.9144108280254777

**Decision Tree Regression**

DecisionTreeRegressor(max\_depth=6, min\_samples\_split=4, random\_state=50)

MSE: 57647020122153.3

**AdaBoost Classification**

AdaBoostClassifier(base\_estimator=DecisionTreeClassifier(max\_depth=1,

random\_state=50), n\_estimators=200)

Accuracy: 0.9080414012738853

**Comparison:**

We can see from the result that the built-in models is more efficient in terms of metrics and runtime the explanation of this is the built-in models use more efficient algorithm to split the data (smart cut) and use the n-jobs(multiprocessing) to run the program, in terms of metrics in the built-in we use Cross Validation and that’s why we get better results and the built-in

The way that we built the Grid Search is worse than the built-in one and that’s effect the results

**Bonus**

**DecisionTree Sensitivity: 0.9514338575393154**

**DecisionTree Specificity: 0.6257142857142857**

**AdaBoost Sensitivity: 0.9875115633672525**

**AdaBoost Specificity: 0.06285714285714286**

the Sensitivity is the true positive rate (tp/tp+fn) and we check in this score if the model predict all the positives right we can see that the rate is high and it says that from all the 1 the model predict 95% of them

The Specificity is the true negative rate (tn/tn+fp) and we check in this score if the model predict the negatives right and it says that from all the 0 the model predict 63% of them

We can optimize these scores by checking them in grid search as a measure

Or make a weight score that consist of the avg of the specificity and sensitivity

So instead of calculating the accuracy in the grid search we can calculate the AVG of them and get the best estimators