Part B

Code : ./Hw3/bostonHousePricePrediction/code.py

Decision Tree Regressor

I experimented with GridSearchCV and got best parameters for following model.

```
 parameters =
{'max_depth':(None,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
,'max_features':(None,1,2,3,13),
'min_samples_split':(2,3,4,5,10,11,12,13,14,15,16,17,18,19,20,21,22,2
3,25,27,100),
'min_samples_leaf': (1, 2, 3)}

DecisionTreeRegressor(criterion='mse', max_depth=7, max_features=None,
           max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=2,
           min_samples_split=19, min_weight_fraction_leaf=0.0,
           presort=False, random_state=1, splitter='best')
```

The MSE for this model was `11.201909059110672`

Following are the results of different models I experimented with

Parameters used :

1) *Parameters 1 :* `max_depth=7, max_features=None, min_samples_split=2, min_samples_leaf=2, random_state=1`
   Mean Squared Error: `13.31046530868291`

2) *Parameters 2:* `DecisionTreeRegressor(max_depth=7, max_features=None, min_samples_split=19, min_samples_leaf=20, random_state=1)`
   Mean Squared Error: `17.48823286528156`

3) *Parameters 3: DecisionTreeRegressor(max_depth=1, max_features=None, min_samples_split=50, min_samples_leaf=50, random_state=1)*
   Mean Squared Error: `54.146473381756216`

The best model was following which gave MSE error of `11.201909059110672`

```
DecisionTreeRegressor(criterion='mse', max_depth=7, max_features=None,
           max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=2,
           min_samples_split=19, min_weight_fraction_leaf=0.0,
           presort=False, random_state=1, splitter='best')
```

ANALYSIS : *We can see as we increase the min_samples_split or min_samples_leaf high, decrease the max_depth of the tree low, the MSE increases. This is because of underfitting the train data. Whereas, when we decrease the min_samples_split or min_samples_leaf very low, increase the max_depth of the tree high the MSE again increases. This is due to overfitting the train data. We need to take parameters such that they neither underfit nor overfit the train data to get good results on test data.*

AdaBoost Regressor

I experimented with GridSearchCV and got best parameters for following model.

```
parameters = {'n_estimators':(5,10,20,50,80,95,100),
                  'learning_rate' : (0.001,0.01,0.1,1.0)}
```

AdaBoostRegressor(base_estimator=DecisionTreeRegressor(criterion='mse', max_depth=None, max_features=None,
max_leaf_nodes=None, min_impurity_decrease=0.0,
min_impurity_split=None, min_samples_leaf=1,
min_samples_split=2, min_weight_fraction_leaf=0.0,
presort=False, random_state=None, splitter='best'),
learning_rate=0.01, loss='linear', n_estimators=100,
random_state=1)

The MSE for this model was `8.542434210526315`

Following are the results of different models I experimented with
Parameters used:

1) Parameters 1 : `AdaBoostRegressor(n_estimators = 5, learning_rate = 1)`

Mean Squared Error: `20.525755855564444`

2) Parameters 2: `AdaBoostRegressor(n_estimators = 5, learning_rate = 0.01)`

Mean Squared Error: `14.734019872234997`

3) Parameters 3: `AdaBoostRegressor(n_estimators = 1000, learning_rate = 0.01)`

Mean Squared Error: `15.88643600961546`

ANALYSIS : *The results for adaboost improve over decisionTree regressor as it decreases the effect of overfitting done by decision trees using ensemble methods. From the experiments, we can see that increasing the learning rate decreases the accuracy. Also, if we decrease the n_estimators too low, the accuracy on test data decreases due to underfitting. The parameters need to be tuned to get accurate models.*