STATS 202A – HW 7

Name: Fuzail Mujahid Khan

UID: 405428622

My apologies for late submission! I hope it won't be negatively marked by more than 20% as you mentioned on the discussion forum.

Part 1)

Mean and Std Dev of the test accuracies :

In[14]:

```
accuracy = []
for train_index, test_index in kfold.split(X):
    X_train, Y_train = X[train_index], Y[train_index]
    X_test, Y_test = X[test_index], Y[test_index]

    xgbm = xgb.XGBClassifier(objective="binary:logistic", random_state=42)
    xgbm.fit(X_train, Y_train)

    y_pred = xgbm.predict(X_test)
    accu = y_pred == Y_test
    accuracy.append(np.mean(accu))

accuracy
print('The mean accuracy after 5-fold CV is ',np.mean(accuracy))
print('The standard deviation of the accuracy after 5-fold CV is ',np.std(accuracy))
```

```
The mean accuracy after 5-fold CV is  0.9648501785437045
The standard deviation of the accuracy after 5-fold CV is  0.009609970350036167
```

Part 2)

Mean test scores for each parameter combination:

In[21]:

```
nt('These are the 9 parameter combinations between max_depth and min_child_weight : \n',search.cv_results_['params'])
nt('These are the corresponding mean test scores for each parameter combination after 5 fold CV : \n',search.cv_results_['mean_test_score'])
```

```
These are the 9 parameter combinations between max_depth and min_child_weight :
 [{'max_depth': 3, 'min_child_weight': 0.1}, {'max_depth': 3, 'min_child_weight': 1}, {'max_depth': 3, 'min_child_weight': 5}, {'max_depth':
5, 'min_child_weight': 0.1}, {'max_depth': 5, 'min_child_weight': 1}, {'max_depth': 5, 'min_child_weight': 5}, {'max_depth': 7, 'min_child_w
eight': 0.1}, {'max_depth': 7, 'min_child_weight': 1}, {'max_depth': 7, 'min_child_weight': 5}]
These are the corresponding mean test scores for each parameter combination after 5 fold CV :
 [0.96309315 0.96309315 0.96485062 0.96133568 0.96660808 0.96309315
 0.96133568 0.96485062 0.96309315]
```

Not sure what you meant by plotting graphs like HW 6 since there we plotted training and testing error as no. of decision trees increased. While here we are only changing max_depth and min_child_weight. So I have plotted multiple graphs just in case.

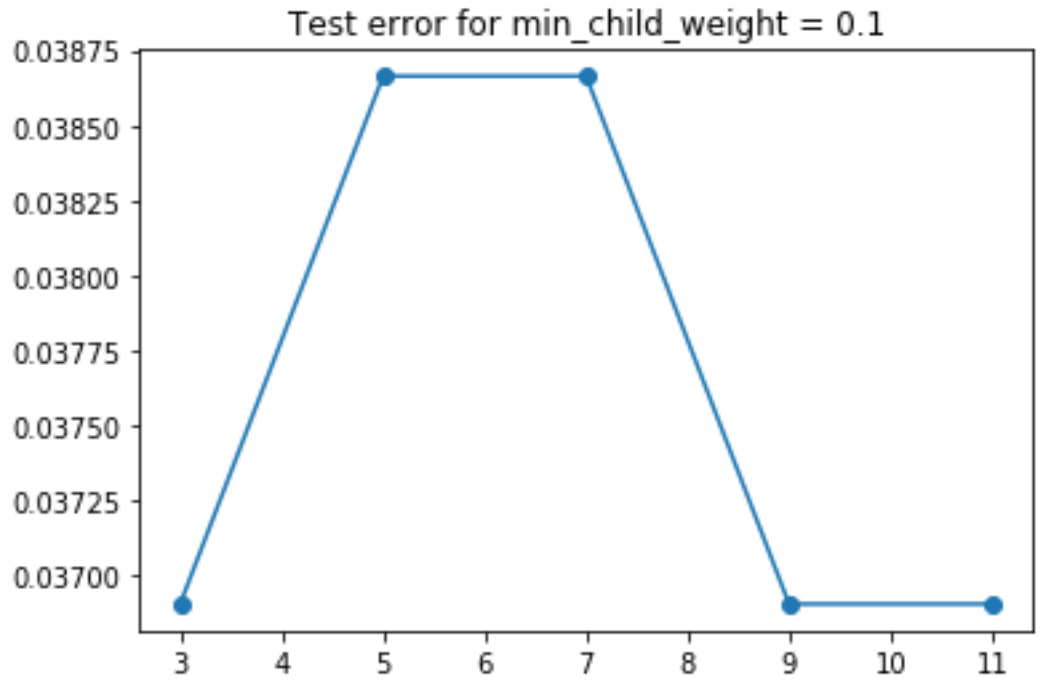
1) Training error is 0 for all the cross validation models. Training accuracy is 100%.

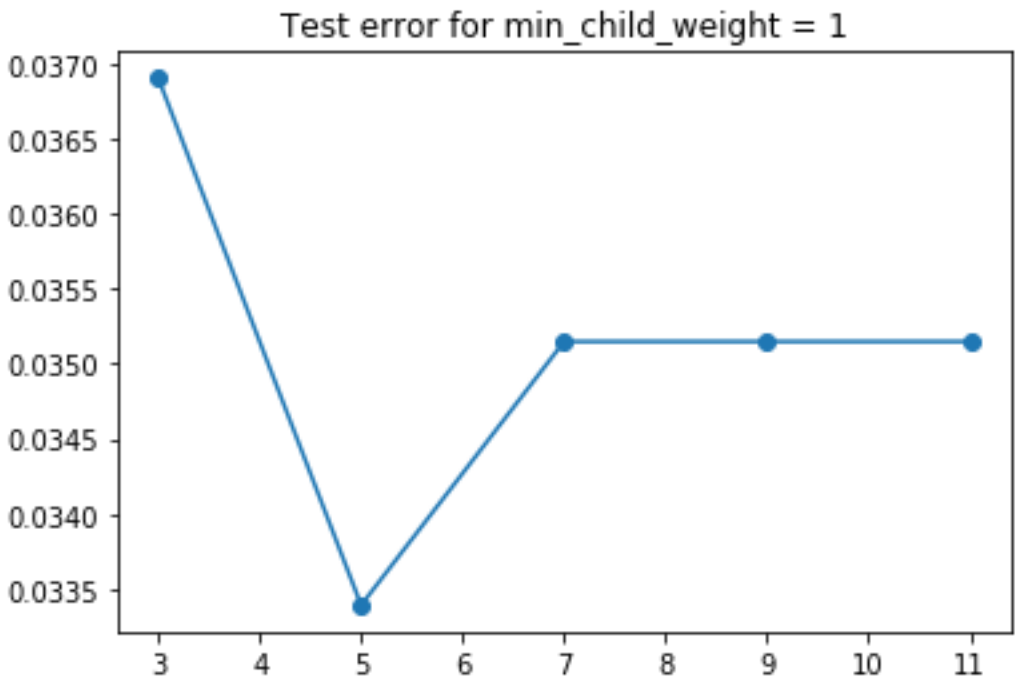
In[79]:

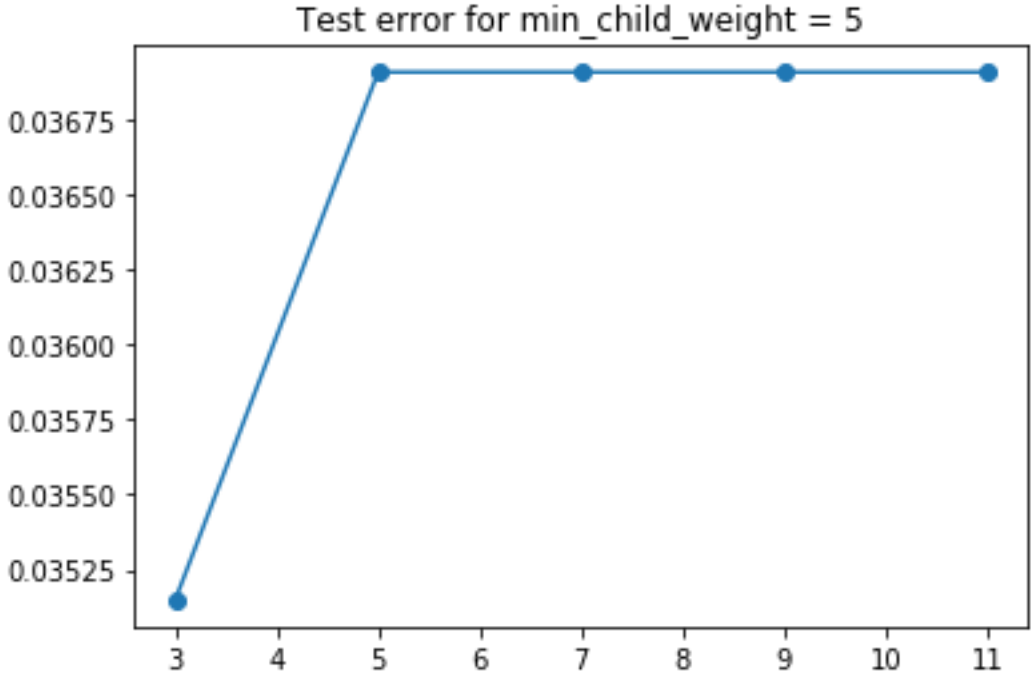
```
accuracy = []
for train_index, test_index in kfold.split(X):
    X_train, Y_train = X[train_index], Y[train_index]
    X_test, Y_test = X[test_index], Y[test_index]

    xgbm = xgb.XGBClassifier(objective="binary:logistic", random_state=42)
    xgbm.fit(X_train, Y_train)

    y_pred = xgbm.predict(X_train)
    accu = y_pred == Y_train
    accuracy.append(np.mean(accu))

print(accuracy)
print('The mean accuracy after 5-fold CV is ',np.mean(accuracy))
print('The standard deviation of the accuracy after 5-fold CV is ',np.std(accuracy))
```

```
[1.0, 1.0, 1.0, 1.0, 1.0]
The mean accuracy after 5-fold CV is  1.0
The standard deviation of the accuracy after 5-fold CV is  0.0
```

The below graphs are varying max_depth by [3,5,7,9,11] for each value of min_child_weight :

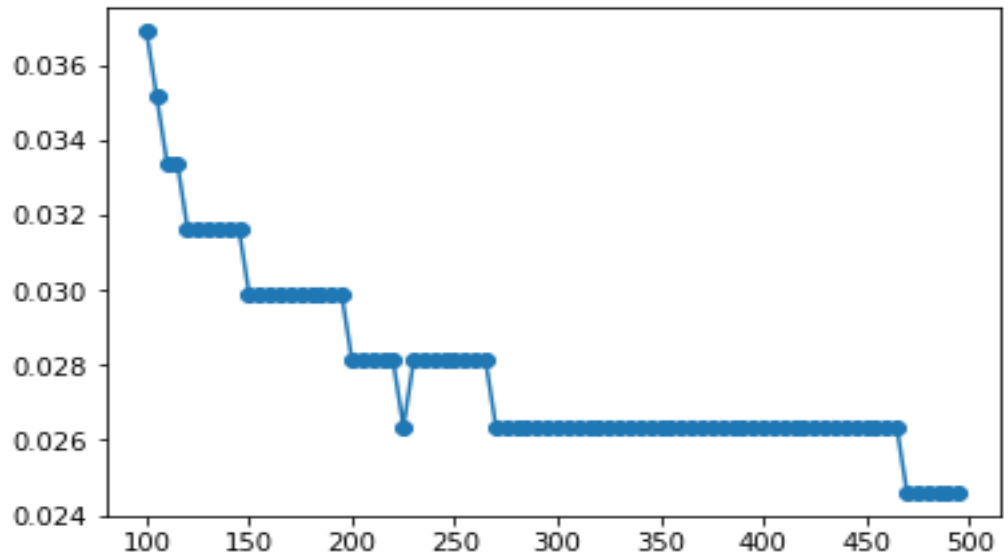



Test error for min_child_weight = 1
0.0370
0.0365
0.0360
0.0355
0.0350
0.0345
0.0340
0.0335
3
4
5
6
7
8
9
10
11


Test error for min_child_weight = 5
0.03675
0.03650
0.03625
0.03600
0.03575
0.03550
0.03525
3
4
5
6
7
8
9
10
11

Since you said to plot like HW6, I have made n_estimators from 100 to 500 in steps of 5, as a hyper parameters which will basically change the number of decision trees in the model.

The below graph is plotting testing error vs no. of decision trees :



Part 3 :

The feature importance from the best XGBoost model plotted: