Enhancing Accuracy with AdaBoost Ensemble: A Comparison of Decision Tree and Fused Model Approaches"

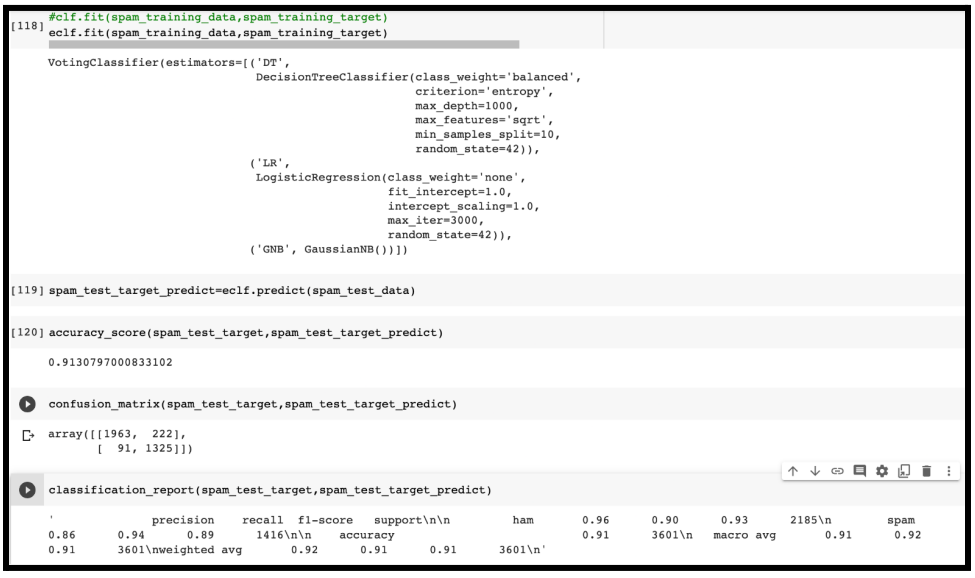
Report:

TASK 1 -

The task is to compare the accuracies of two classifiers: a fused model and an AdaBoost ensemble with Decision Tree as the base learner. The classifiers should be trained using the first 1000 instances and tested using the remaining 3601 instances. The results of the comparison should be reported along with observations and conclusions, supported by evidence.

To complete this task, first, the training and testing data files should be created by splitting the data into two parts, with 1000 instances used for training and 3601 instances used for testing. Then, the two classifiers should be trained and tested on the respective data sets. Finally, the accuracy of each classifier should be calculated and compared, and observations and conclusions should be drawn based on the results.

Below is the accuracy of the fused model.



```
[118] #clf.fit(spam_training_data,spam_training_target)
      eclf.fit(spam_training_data,spam_training_target)

      VotingClassifier(estimators=[('DT',
                                    DecisionTreeClassifier(class_weight='balanced',
                                                           criterion='entropy',
                                                           max_depth=1000,
                                                           max_features='sqrt',
                                                           min_samples_split=10,
                                                           random_state=42)),
                                   ('LR',
                                    LogisticRegression(class_weight='none',
                                                       fit_intercept=1.0,
                                                       intercept_scaling=1.0,
                                                       max_iter=3000,
                                                       random_state=42)),
                                   ('GNB', GaussianNB())])

[119] spam_test_target_predict=eclf.predict(spam_test_data)

[120] accuracy_score(spam_test_target,spam_test_target_predict)

      0.9130797000833102

      confusion_matrix(spam_test_target,spam_test_target_predict)

      array([[1963,  222],
             [  91, 1325]])

      classification_report(spam_test_target,spam_test_target_predict)

      '              precision    recall  f1-score   support\n\n         ham       0.96      0.90      0.93      2185\n        spam
      0.86      0.94      0.89      1416\n\n    accuracy                           0.91      3601\n   macro avg       0.91      0.92
      0.91      3601\nweighted avg       0.92      0.91      0.91      3601\n'
```

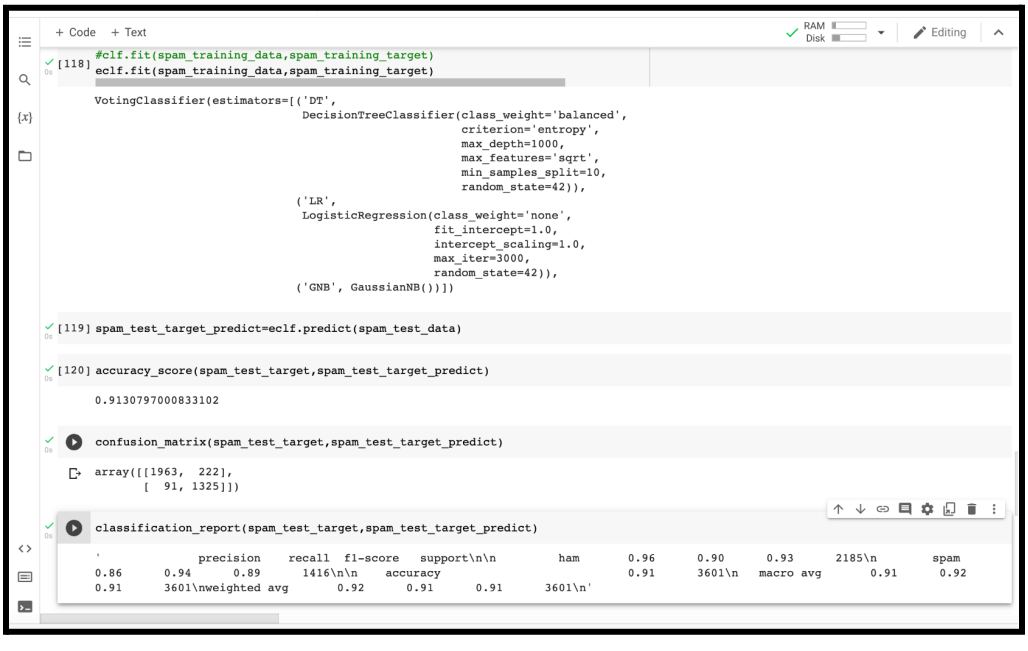
Below is the accuracy of the adaboost ensemble with decision tree as the base learner(Note: To use decision tree as the base learner, the base estimator should be set as none, If `None`, then the base estimator is DecisionTreeClassifier)

```
[109] from sklearn.naive_bayes import GaussianNB
      from sklearn.ensemble import AdaBoostClassifier, VotingClassifier
      from sklearn.linear_model import LogisticRegression
      from sklearn.neighbors import KNeighborsClassifier
      from sklearn.metrics import confusion_matrix
      from sklearn.metrics import classification_report
      from sklearn.metrics import accuracy_score
      #clf = RandomForestClassifier(n_estimators=1000)
      clf = DecisionTreeClassifier(criterion = "entropy")
      clf_dt = DecisionTreeClassifier(min_samples_split=10, max_depth = 1000, criterion = "entropy", splitter = "best", random_state=42, c
      #splitter = "best", max_depth = 10, min_samples_split=100, min_samples_leaf=1, random_state=42, class_weight = "balanced", min_weigh
      #clf = KNeighborsClassifier(n_neighbors=7)
      clf = AdaBoostClassifier(n_estimators = 200, base_estimator=None)
      #LRI = LogisticRegression()
      #clf_lr = LogisticRegression(solver='lbfgs', penalty = "l2", dual = False, tol = 1e-4, C = 1.0, fit_intercept = 1.0, intercept_scal
      #clf_gnb = GaussianNB()
      #clf = AdaBoostClassifier(n_estimators = 200,base_estimator=LRI)
      #eclf = VotingClassifier(estimators = [('DT', clf_dt),('LR',clf_lr),('GNB',clf_gnb)], voting ='hard')
      clf.fit(spam_training_data,spam_training_target)
      #eclf.fit(spam_training_data,spam_training_target)

      AdaBoostClassifier(n_estimators=200)

[110] spam_test_target_predict=clf.predict(spam_test_data)

[111] accuracy_score(spam_test_target,spam_test_target_predict)

      0.9208553179672313

      confusion_matrix(spam_test_target,spam_test_target_predict)

      array([[2007,  178],
             [ 107, 1309]])
```

TASK 2 -

The task is to compare the accuracy of two classifiers: a fused model and a Random Forest classifier with 1000 base learners.
The classifiers will be trained using the first 1000 instances and tested using the remaining 3601 instances.

Below is the accuracy of the fused model.



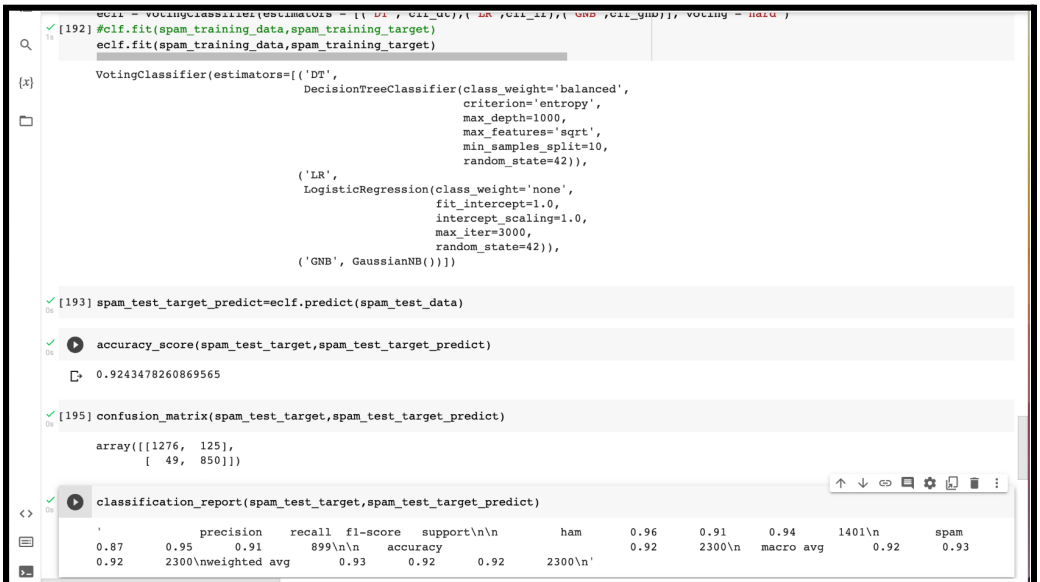
```
+ Code  + Text                                                    RAM  Disk   Editing

[118] #clf.fit(spam_training_data,spam_training_target)
      eclf.fit(spam_training_data,spam_training_target)

      VotingClassifier(estimators=[('DT',
                                    DecisionTreeClassifier(class_weight='balanced',
                                                           criterion='entropy',
                                                           max_depth=1000,
                                                           max_features='sqrt',
                                                           min_samples_split=10,
                                                           random_state=42)),
                                   ('LR',
                                    LogisticRegression(class_weight='none',
                                                       fit_intercept=1.0,
                                                       intercept_scaling=1.0,
                                                       max_iter=3000,
                                                       random_state=42)),
                                   ('GNB', GaussianNB())])

[119] spam_test_target_predict=eclf.predict(spam_test_data)

[120] accuracy_score(spam_test_target,spam_test_target_predict)

      0.9130797000833102

      confusion_matrix(spam_test_target,spam_test_target_predict)

      array([[1963,  222],
             [  91, 1325]])

      classification_report(spam_test_target,spam_test_target_predict)

      '              precision    recall  f1-score   support\n\n         ham       0.96      0.90      0.93      2185\n        spam
      0.86      0.94      0.89      1416\n\n    accuracy                           0.91      3601\n   macro avg       0.91      0.92
      0.91      3601\nweighted avg       0.92      0.91      0.91      3601\n'
```

Below is the accuracy of the random forest classifier with 1000 base learners:

TASK 3-

The task is to study the impact of training sample size on the accuracy of two classifiers: a fused classifier and an AdaBoost ensemble with Decision Tree as the base learner. The study will involve comparing their accuracy with four different training-test splits: 50%-50%, 60%-40%, 70%-30%, and 80%-20%.

Below is the accuracy of the fused classifier with the 50%-50% training test split



Below is the accuracy of the Ada boost Ensemble with decision tree as the base learner with the <u>50%-50% training test split</u>

```
from sklearn.metrics import accuracy_score
#clf = RandomForestClassifier(n_estimators=1000)
clf = DecisionTreeClassifier(criterion = "entropy")
clf_dt = DecisionTreeClassifier(min_samples_split=10, max_depth = 1000, criterion = "entropy", splitter = "best", random_state=42, (
#splitter = "best", max_depth = 10, min_samples_split=100, min_samples_leaf=1, random_state=42, class_weight = "balanced", min_weigl
#clf = KNeighborsClassifier(n_neighbors=7)
clf = AdaBoostClassifier(n_estimators = 200, base_estimator=None)
#LRI = LogisticRegression()
#clf_lr = LogisticRegression(solver='lbfgs', penalty = "l2", dual = False, tol = 1e-4, C = 1.0, fit_intercept = 1.0, intercept_scal:
#clf_gnb = GaussianNB()
#clf = AdaBoostClassifier(n_estimators = 200,base_estimator=LRI)
#eclf = VotingClassifier(estimators = [('DT', clf_dt),('LR',clf_lr),('GNB',clf_gnb)], voting ='hard')
clf.fit(spam_training_data,spam_training_target)
#eclf.fit(spam_training_data,spam_training_target)
```

```
AdaBoostClassifier(n_estimators=200)
```

```
[144] spam_test_target_predict=clf.predict(spam_test_data)
```

```
accuracy_score(spam_test_target,spam_test_target_predict)
```

```
0.9304347826086956
```

```
[146] confusion_matrix(spam_test_target,spam_test_target_predict)
```

```
array([[1303,   98],
       [  62,  837]])
```

```
[147] classification_report(spam_test_target,spam_test_target_predict)
```

```
'              precision    recall  f1-score   support\n\n         ham       0.95      0.93      0.94      1401\n        spam
0.90      0.93      0.91       899\n\n    accuracy                           0.93      2300\n   macro avg       0.92      0.93
0.93      2300\nweighted avg       0.93      0.93      0.93      2300\n'
```

Below is the accuracy of the fused classifier with the <u>60%-40% training test split</u>

```
eclf = VotingClassifier(estimators = [('DT', clf_dt),('LR',clf_lr),('GNB',clf_gnb)], voting ='hard')
#clf.fit(spam_training_data,spam_training_target)
eclf.fit(spam_training_data,spam_training_target)
```

```
VotingClassifier(estimators=[('DT',
                              DecisionTreeClassifier(class_weight='balanced',
                                                     criterion='entropy',
                                                     max_depth=1000,
                                                     max_features='sqrt',
                                                     min_samples_split=10,
                                                     random_state=42)),
                             ('LR',
                              LogisticRegression(class_weight='none',
                                                 fit_intercept=1.0,
                                                 intercept_scaling=1.0,
                                                 max_iter=3000,
                                                 random_state=42)),
                             ('GNB', GaussianNB())])
```

```
[165] spam_test_target_predict=eclf.predict(spam_test_data)
```

```
[166] accuracy_score(spam_test_target,spam_test_target_predict)
```

```
0.9266702878870179
```

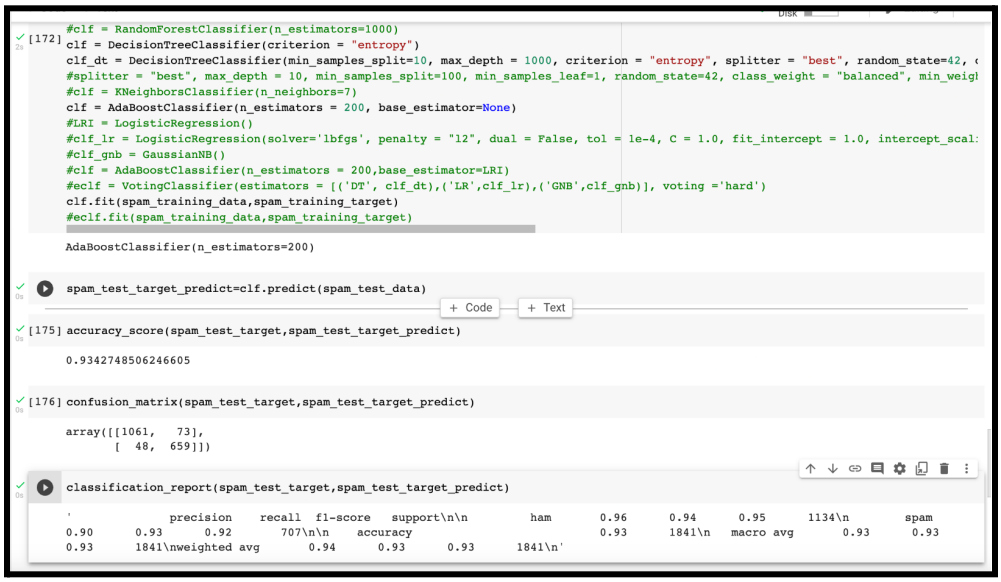
```
confusion_matrix(spam_test_target,spam_test_target_predict)
```

```
array([[1037,   97],
       [  38,  669]])
```

```
classification_report(spam_test_target,spam_test_target_predict)
```

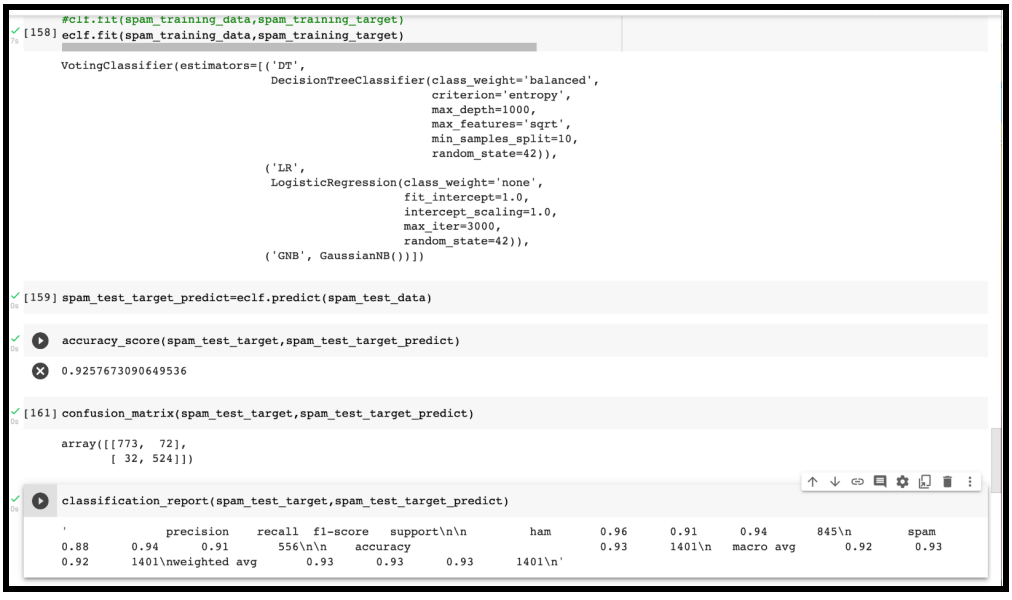
```
'              precision    recall  f1-score   support\n\n         ham       0.96      0.91      0.94      1134\n        spam
0.87      0.95      0.91       707\n\n    accuracy                           0.93      1841\n   macro avg       0.92      0.93
0.92      1841\nweighted avg       0.93      0.93      0.93      1841\n'
```

Below is the accuracy of the Ada boost Ensemble with decision tree as the base learner with the 60%-40% training test split



```
[172] #clf = RandomForestClassifier(n_estimators=1000)
      clf = DecisionTreeClassifier(criterion = "entropy")
      clf_dt = DecisionTreeClassifier(min_samples_split=10, max_depth = 1000, criterion = "entropy", splitter = "best", random_state=42, c
      #splitter = "best", max_depth = 10, min_samples_split=100, min_samples_leaf=1, random_state=42, class_weight = "balanced", min_weigh
      #clf = KNeighborsClassifier(n_neighbors=7)
      clf = AdaBoostClassifier(n_estimators = 200, base_estimator=None)
      #LRI = LogisticRegression()
      #clf_lr = LogisticRegression(solver='lbfgs', penalty = "l2", dual = False, tol = 1e-4, C = 1.0, fit_intercept = 1.0, intercept_scal
      #clf_gnb = GaussianNB()
      #clf = AdaBoostClassifier(n_estimators = 200,base_estimator=LRI)
      #eclf = VotingClassifier(estimators = [('DT', clf_dt),('LR',clf_lr),('GNB',clf_gnb)], voting ='hard')
      clf.fit(spam_training_data,spam_training_target)
      #eclf.fit(spam_training_data,spam_training_target)

      AdaBoostClassifier(n_estimators=200)

      spam_test_target_predict=clf.predict(spam_test_data)

      + Code  + Text

[175] accuracy_score(spam_test_target,spam_test_target_predict)

      0.9342748506246605

[176] confusion_matrix(spam_test_target,spam_test_target_predict)

      array([[1061,   73],
             [  48,  659]])

      classification_report(spam_test_target,spam_test_target_predict)

      '              precision    recall  f1-score   support\n\n         ham       0.96      0.94      0.95      1134\n        spam
      0.90      0.93      0.92       707\n\n    accuracy                           0.93      1841\n   macro avg       0.93      0.93
      0.93      1841\nweighted avg       0.94      0.93      0.93      1841\n'
```

Below is the accuracy of the fused classifier with the 70%-30% training test split

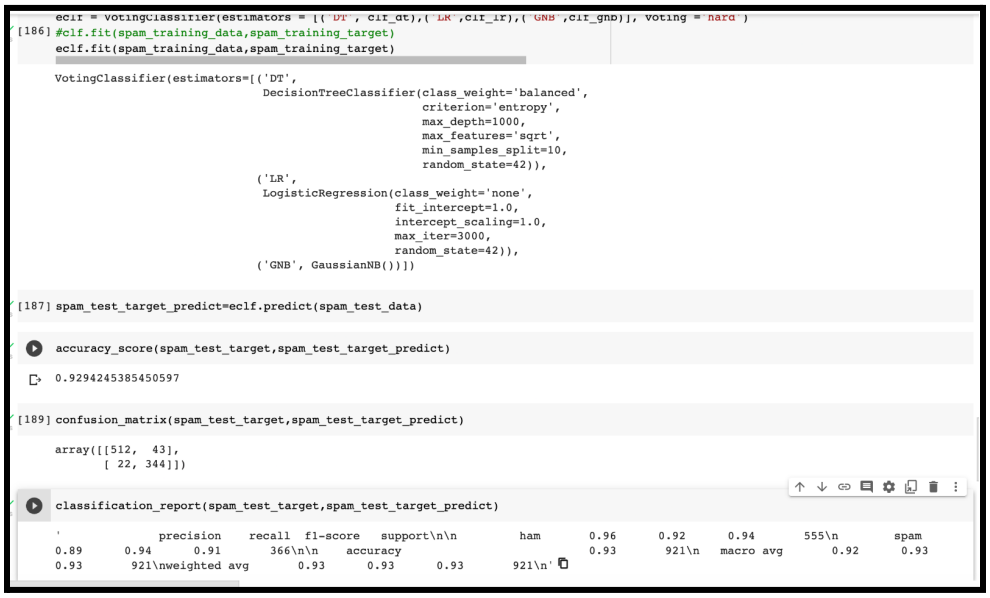


```
      #clf.fit(spam_training_data,spam_training_target)
[158] eclf.fit(spam_training_data,spam_training_target)

      VotingClassifier(estimators=[('DT',
                                    DecisionTreeClassifier(class_weight='balanced',
                                                           criterion='entropy',
                                                           max_depth=1000,
                                                           max_features='sqrt',
                                                           min_samples_split=10,
                                                           random_state=42)),
                                   ('LR',
                                    LogisticRegression(class_weight='none',
                                                       fit_intercept=1.0,
                                                       intercept_scaling=1.0,
                                                       max_iter=3000,
                                                       random_state=42)),
                                   ('GNB', GaussianNB())])

[159] spam_test_target_predict=eclf.predict(spam_test_data)

      accuracy_score(spam_test_target,spam_test_target_predict)

      0.9257673090649536

[161] confusion_matrix(spam_test_target,spam_test_target_predict)

      array([[773,  72],
             [ 32, 524]])

      classification_report(spam_test_target,spam_test_target_predict)

      '              precision    recall  f1-score   support\n\n         ham       0.96      0.91      0.94       845\n        spam
      0.88      0.94      0.91       556\n\n    accuracy                           0.93      1401\n   macro avg       0.92      0.93
      0.92      1401\nweighted avg       0.93      0.93      0.93      1401\n'
```

Below is the accuracy of the Ada boost Ensemble with decision tree as the base learner with the 70%-30% training test split

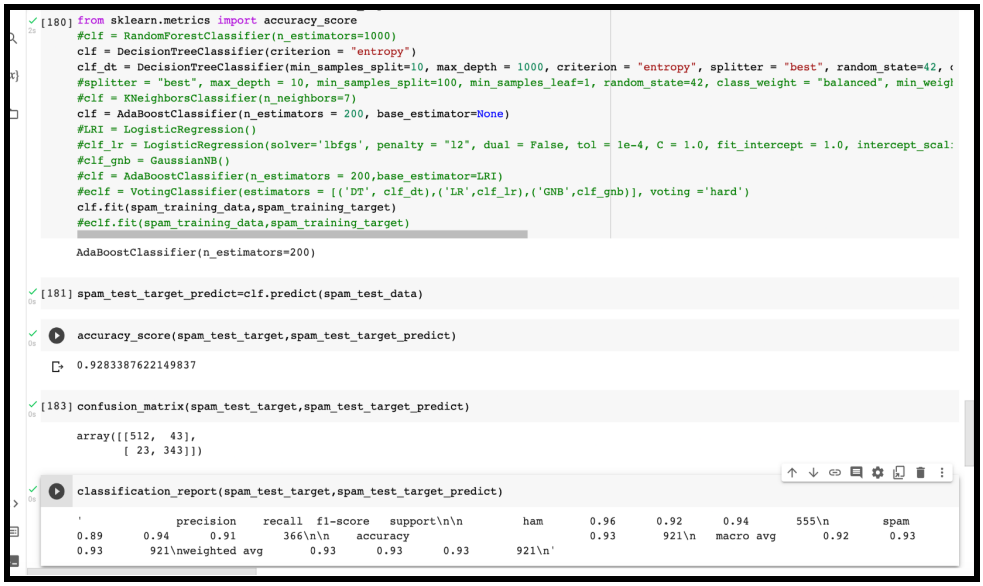
```
+ Code + Text                                                    RAM Disk   Editing

[153] from sklearn.metrics import accuracy_score
      #clf = RandomForestClassifier(n_estimators=1000)
      clf = DecisionTreeClassifier(criterion = "entropy")
      clf_dt = DecisionTreeClassifier(min_samples_split=10, max_depth = 1000, criterion = "entropy", splitter = "best", random_state=42, c
      #splitter = "best", max_depth = 10, min_samples_split=100, min_samples_leaf=1, random_state=42, class_weight = "balanced", min_weigh
      #clf = KNeighborsClassifier(n_neighbors=7)
      clf = AdaBoostClassifier(n_estimators = 200, base_estimator=None)
      #LRI = LogisticRegression()
      #clf_lr = LogisticRegression(solver='lbfgs', penalty = "l2", dual = False, tol = 1e-4, C = 1.0, fit_intercept = 1.0, intercept_scal
      #clf_gnb = GaussianNB()
      #clf = AdaBoostClassifier(n_estimators = 200,base_estimator=LRI)
      #eclf = VotingClassifier(estimators = [('DT', clf_dt),('LR',clf_lr),('GNB',clf_gnb)], voting ='hard')
      clf.fit(spam_training_data,spam_training_target)
      #eclf.fit(spam_training_data,spam_training_target)

      AdaBoostClassifier(n_estimators=200)

[154] spam_test_target_predict=clf.predict(spam_test_data)

[155] accuracy_score(spam_test_target,spam_test_target_predict)

      0.9307637401855817

      confusion_matrix(spam_test_target,spam_test_target_predict)

      array([[783,  62],
             [ 35, 521]])

      classification_report(spam_test_target,spam_test_target_predict)

      '              precision    recall  f1-score   support\n\n         ham       0.96      0.93      0.94       845\n        spam
      0.89      0.94      0.91       556\n\n    accuracy                           0.93      1401\n   macro avg       0.93      0.93
      0.93      1401\nweighted avg       0.93      0.93      0.93      1401\n'
```

Below is the accuracy of the fused classifier with the 80%-20% training test split



```
      eclf = VotingClassifier(estimators = [('DT', clf_dt),('LR',clf_lr),('GNB',clf_gnb)], voting ='hard')
[186] #clf.fit(spam_training_data,spam_training_target)
      eclf.fit(spam_training_data,spam_training_target)

      VotingClassifier(estimators=[('DT',
                                    DecisionTreeClassifier(class_weight='balanced',
                                                           criterion='entropy',
                                                           max_depth=1000,
                                                           max_features='sqrt',
                                                           min_samples_split=10,
                                                           random_state=42)),
                                   ('LR',
                                    LogisticRegression(class_weight='none',
                                                       fit_intercept=1.0,
                                                       intercept_scaling=1.0,
                                                       max_iter=3000,
                                                       random_state=42)),
                                   ('GNB', GaussianNB())])

[187] spam_test_target_predict=eclf.predict(spam_test_data)

      accuracy_score(spam_test_target,spam_test_target_predict)

      0.9294245385450597

[189] confusion_matrix(spam_test_target,spam_test_target_predict)

      array([[512,  43],
             [ 22, 344]])

      classification_report(spam_test_target,spam_test_target_predict)

      '              precision    recall  f1-score   support\n\n         ham       0.96      0.92      0.94       555\n        spam
      0.89      0.94      0.91       366\n\n    accuracy                           0.93      921\n   macro avg       0.92      0.93
      0.93       921\nweighted avg       0.93      0.93      0.93       921\n'
```

Below is the accuracy of the Ada boost Ensemble with decision tree as the base learner with the 80%-20% training test split



```
[180] from sklearn.metrics import accuracy_score
      #clf = RandomForestClassifier(n_estimators=1000)
      clf = DecisionTreeClassifier(criterion = "entropy")
      clf_dt = DecisionTreeClassifier(min_samples_split=10, max_depth = 1000, criterion = "entropy", splitter = "best", random_state=42, c
      #splitter = "best", max_depth = 10, min_samples_split=100, min_samples_leaf=1, random_state=42, class_weight = "balanced", min_weigl
      #clf = KNeighborsClassifier(n_neighbors=7)
      clf = AdaBoostClassifier(n_estimators = 200, base_estimator=None)
      #LRI = LogisticRegression()
      #clf_lr = LogisticRegression(solver='lbfgs', penalty = "l2", dual = False, tol = 1e-4, C = 1.0, fit_intercept = 1.0, intercept_scal:
      #clf_gnb = GaussianNB()
      #clf = AdaBoostClassifier(n_estimators = 200,base_estimator=LRI)
      #eclf = VotingClassifier(estimators = [('DT', clf_dt),('LR',clf_lr),('GNB',clf_gnb)], voting ='hard')
      clf.fit(spam_training_data,spam_training_target)
      #eclf.fit(spam_training_data,spam_training_target)

      AdaBoostClassifier(n_estimators=200)

[181] spam_test_target_predict=clf.predict(spam_test_data)

      accuracy_score(spam_test_target,spam_test_target_predict)

      0.9283387622149837

[183] confusion_matrix(spam_test_target,spam_test_target_predict)

      array([[512,  43],
             [ 23, 343]])

      classification_report(spam_test_target,spam_test_target_predict)

      '              precision    recall  f1-score   support\n\n         ham       0.96      0.92      0.94       555\n        spam
      0.89      0.94      0.91       366\n\n    accuracy                           0.93       921\n   macro avg       0.92      0.93
      0.93       921\nweighted avg       0.93      0.93      0.93       921\n'
```