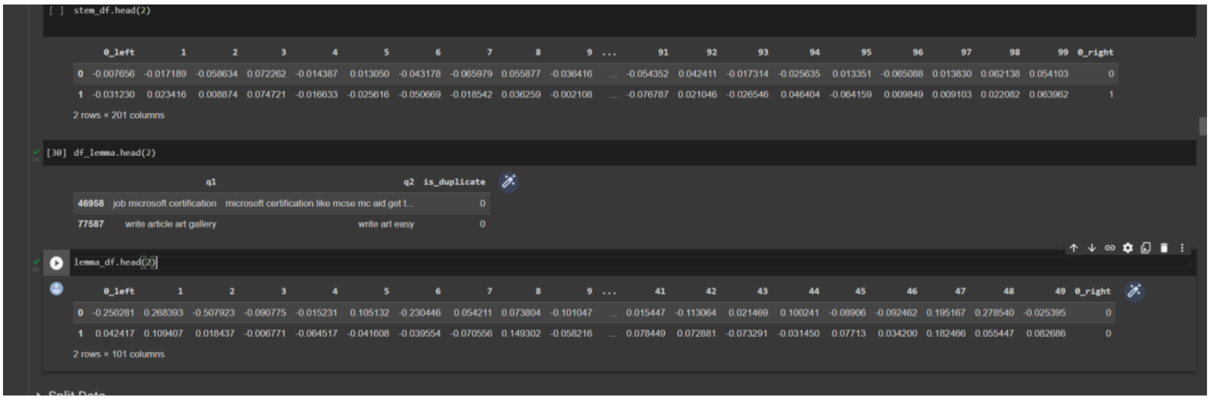
DOC2VEC



```
[ ] stem_df.head(2)
```

	0_left	1	2	3	4	5	6	7	8	9	...	91	92	93	94	95	96	97	98	99	0_right
0	-0.007656	-0.017189	-0.058634	0.072262	-0.014387	0.013050	-0.043178	-0.065979	0.055877	-0.036416	...	-0.054352	0.042411	-0.017314	-0.025635	0.013351	-0.065088	0.013830	0.062138	0.054103	0
1	-0.031230	0.023416	0.008874	0.074721	-0.016633	-0.025616	-0.050669	-0.018542	0.036259	-0.002108	...	-0.076787	0.021046	-0.026546	0.046404	-0.064159	0.009849	0.009103	0.022082	0.063962	1

2 rows × 201 columns

```
[30] df_lemma.head(2)
```

	q1	q2	is_duplicate
46958	job microsoft certification	microsoft certification like mcse mc aid get t...	0
77587	write article art gallery	write art easy	0

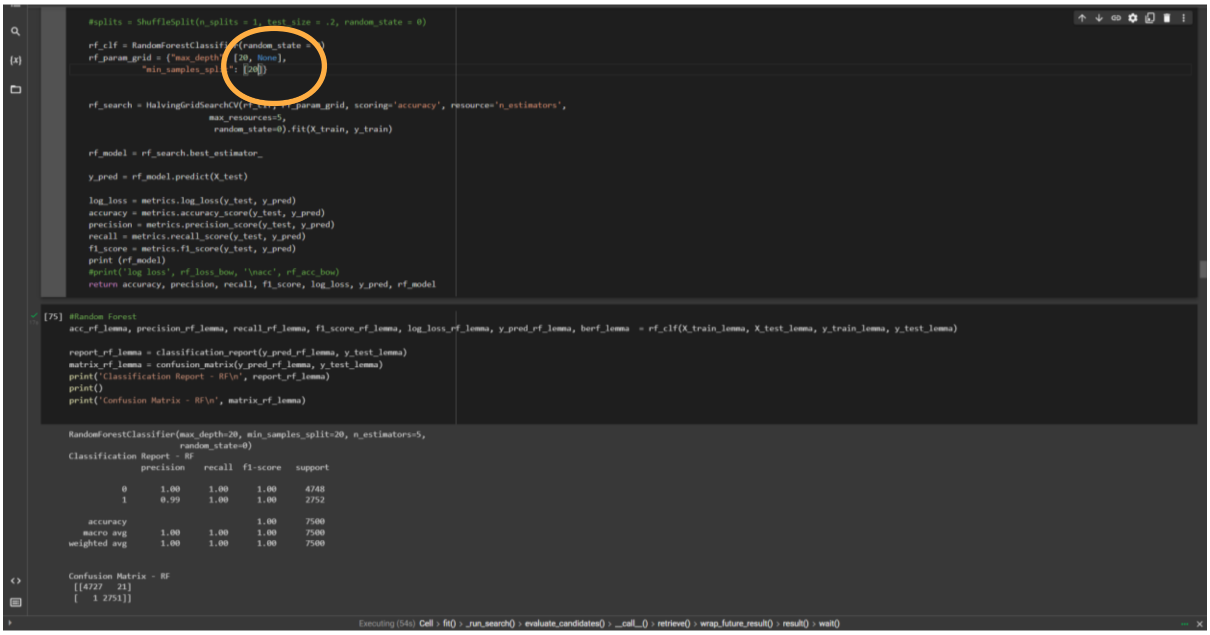
```
lemma_df.head(2)
```

	0_left	1	2	3	4	5	6	7	8	9	...	41	42	43	44	45	46	47	48	49	0_right
0	-0.250281	0.268393	-0.507923	-0.090775	-0.015231	0.105132	-0.230446	0.054211	0.073804	-0.101047	...	0.015447	-0.113064	0.021469	0.100241	-0.08906	-0.092462	0.195167	0.278540	-0.025395	0
1	0.042417	0.109407	0.018437	-0.006771	-0.064517	-0.041608	-0.039554	-0.070556	0.149302	-0.058216	...	0.078449	0.072881	-0.073291	-0.031450	0.07713	0.034200	0.182466	0.055447	0.082686	0

2 rows × 101 columns

Split Data

RANDOM FOREST

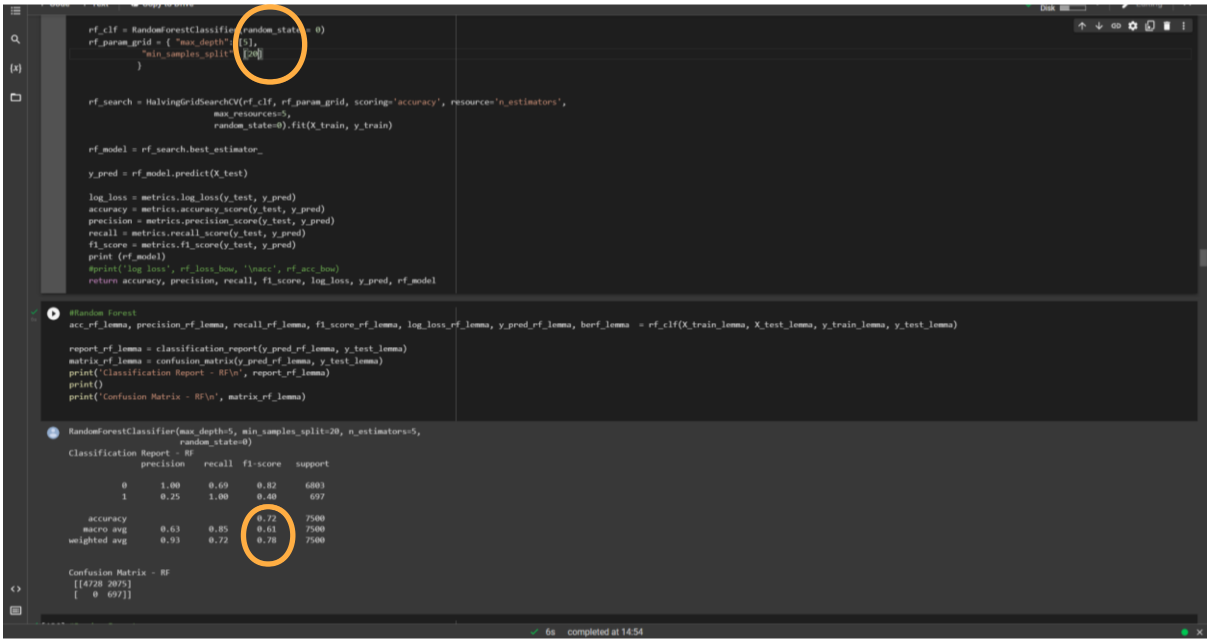


```
    #splits = ShuffleSplit(n_splits = 1, test_size = .2, random_state = 0)

    rf_clf = RandomForestClassifier(random_state = 0)
    rf_param_grid = {"max_depth": [20, None],
                "min_samples_split": [20]}


    rf_search = HalvingGridSearchCV(rf_clf, rf_param_grid, scoring='accuracy', resource='n_estimators',
                               max_resources=5,
                                random_state=0).fit(X_train, y_train)

    rf_model = rf_search.best_estimator_

    y_pred = rf_model.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred)
    recall = metrics.recall_score(y_test, y_pred)
    f1_score = metrics.f1_score(y_test, y_pred)
    print (rf_model)
    #print('log loss', rf_loss_bow, '\nacc', rf_acc_bow)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, rf_model
```

```
[75] #Random Forest
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(max_depth=20, min_samples_split=20, n_estimators=5,
                       random_state=0)
Classification Report - RF
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      4748
           1       0.99      1.00      1.00      2752

    accuracy                           1.00      7500
   macro avg       1.00      1.00      1.00      7500
weighted avg       1.00      1.00      1.00      7500


Confusion Matrix - RF
 [[4727   21]
 [   1 2751]]
```

Executing (54s) Cell > fit() > _run_search() > evaluate_candidates() > __call__() > retrieve() > wrap_future_result() > result() > wait()

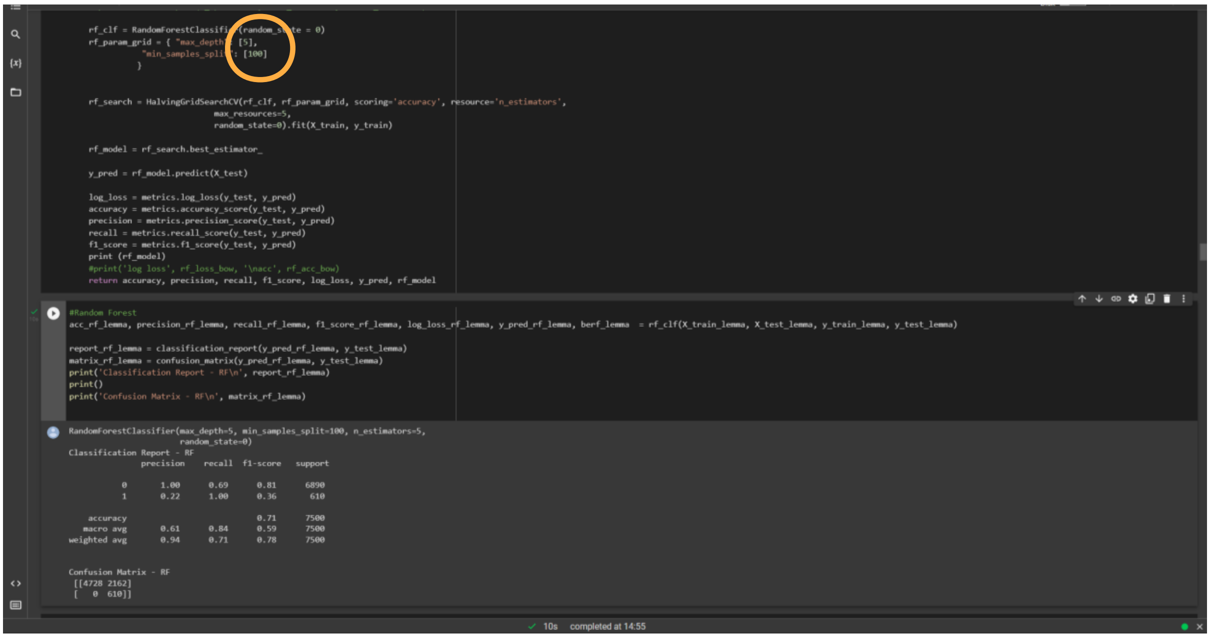


```
    rf_clf = RandomForestClassifier(random_state = 0)
    rf_param_grid = { "max_depth": [5],
                "min_samples_split": [20]
               }


    rf_search = HalvingGridSearchCV(rf_clf, rf_param_grid, scoring='accuracy', resource='n_estimators',
                                max_resources=5,
                                random_state=0).fit(X_train, y_train)

    rf_model = rf_search.best_estimator_

    y_pred = rf_model.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred)
    recall = metrics.recall_score(y_test, y_pred)
    f1_score = metrics.f1_score(y_test, y_pred)
    print (rf_model)
    #print('log loss', rf_loss_bow, '\nacc', rf_acc_bow)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, rf_model
```

```
#Random Forest
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(max_depth=5, min_samples_split=20, n_estimators=5,
                       random_state=0)
Classification Report - RF
               precision    recall  f1-score   support

           0       1.00      0.69      0.82      6803
           1       0.25      1.00      0.40       697

    accuracy                           0.72      7500
   macro avg       0.63      0.85      0.61      7500
weighted avg       0.93      0.72      0.78      7500


Confusion Matrix - RF
 [[4728 2075]
 [   0  697]]
```

✓ 6s completed at 14:54

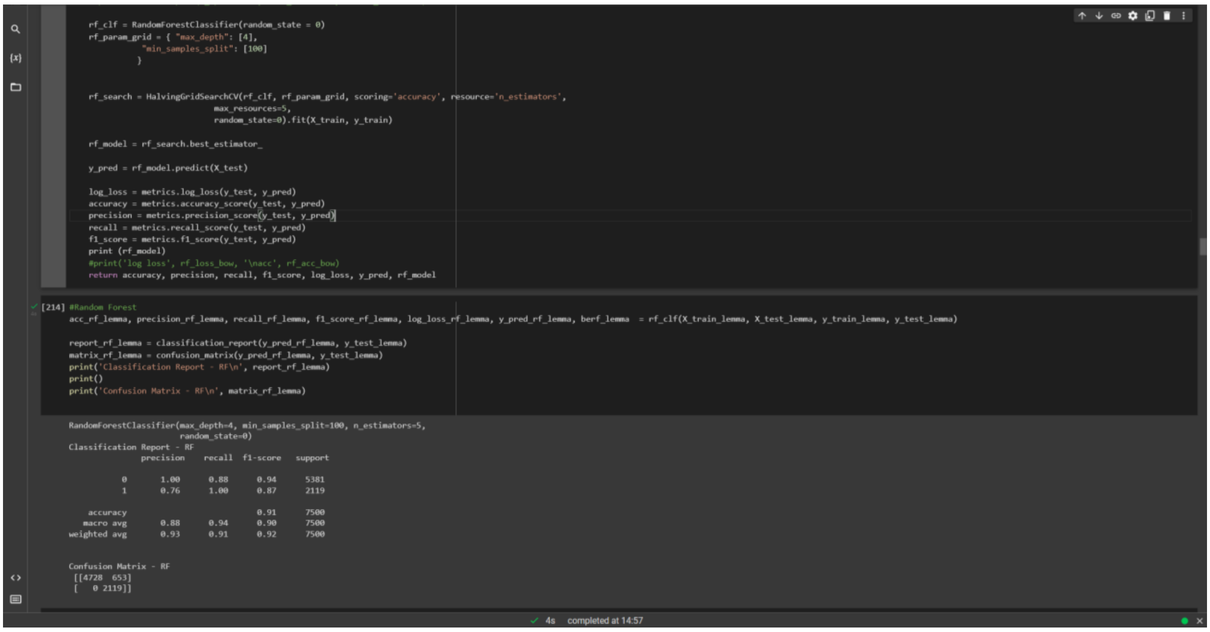


```
    rf_clf = RandomForestClassifier(random_state = 0)
    rf_param_grid = { "max_depth": [5],
                "min_samples_split": [100]
              }


    rf_search = HalvingGridSearchCV(rf_clf, rf_param_grid, scoring='accuracy', resource='n_estimators',
                          max_resources=5,
                          random_state=0).fit(X_train, y_train)

    rf_model = rf_search.best_estimator_

    y_pred = rf_model.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred)
    recall = metrics.recall_score(y_test, y_pred)
    f1_score = metrics.f1_score(y_test, y_pred)
    print (rf_model)
    #print('log loss', rf_loss_bow, '\nacc', rf_acc_bow)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, rf_model
```

```
#Random Forest
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(max_depth=5, min_samples_split=100, n_estimators=5,
                       random_state=0)
Classification Report - RF
               precision    recall  f1-score   support

           0       1.00      0.69      0.81      6890
           1       0.22      1.00      0.36       610

    accuracy                           0.71      7500
   macro avg       0.61      0.84      0.59      7500
weighted avg       0.94      0.71      0.78      7500


Confusion Matrix - RF
 [[4728 2162]
 [   0  610]]
```

✓ 10s completed at 14:55



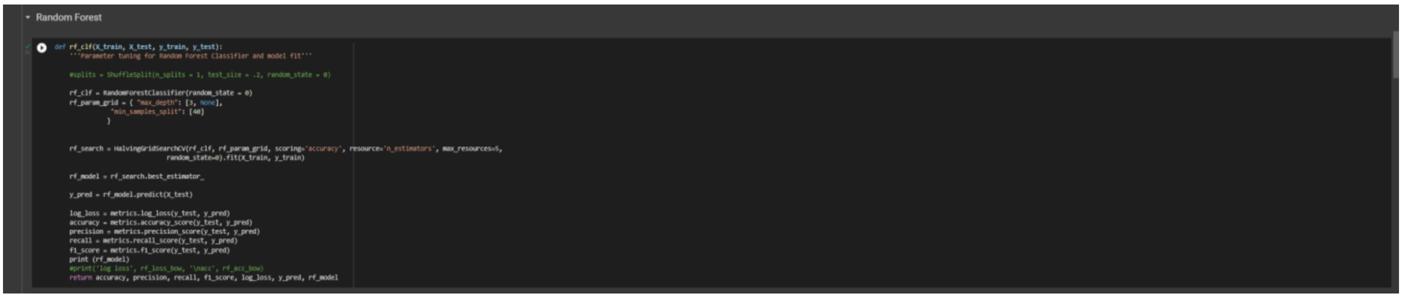
```
    rf_clf = RandomForestClassifier(random_state = 0)
    rf_param_grid = { "max_depth": [4],
                "min_samples_split": [100]
              }


    rf_search = HalvingGridSearchCV(rf_clf, rf_param_grid, scoring='accuracy', resource='n_estimators',
                          max_resources=5,
                          random_state=0).fit(X_train, y_train)

    rf_model = rf_search.best_estimator_

    y_pred = rf_model.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred)
    recall = metrics.recall_score(y_test, y_pred)
    f1_score = metrics.f1_score(y_test, y_pred)
    print (rf_model)
    #print('log loss', rf_loss_bow, '\nacc', rf_acc_bow)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, rf_model
```

```
[214] #Random Forest
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(max_depth=4, min_samples_split=100, n_estimators=5,
                       random_state=0)
Classification Report - RF
               precision    recall  f1-score   support

           0       1.00      0.88      0.94      5381
           1       0.76      1.00      0.87      2119

    accuracy                           0.91      7500
   macro avg       0.88      0.94      0.90      7500
weighted avg       0.93      0.91      0.92      7500


Confusion Matrix - RF
 [[4728  653]
 [   0 2119]]
```

✓ 4s completed at 14:57

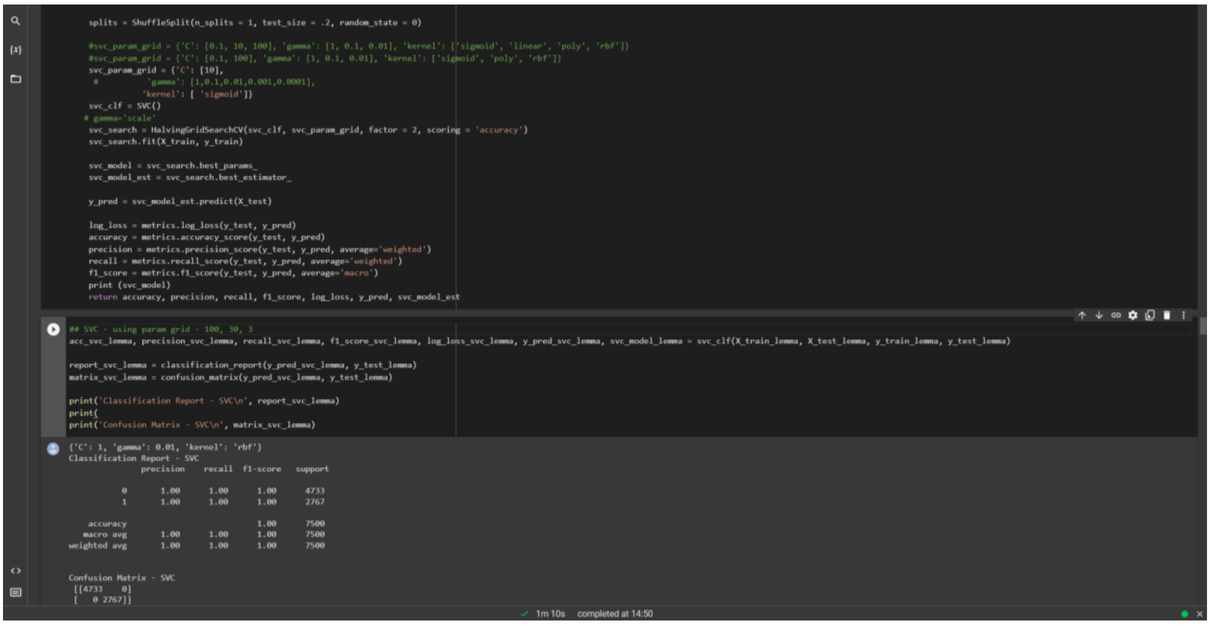
SELECTED HP



Random Forest

```
def rf_clf(X_train, X_test, y_train, y_test):
    '''Parameter tuning for Random Forest Classifier and model fit'''

    #splits = ShuffleSplit(n_splits = 1, test_size = .2, random_state = 0)

    rf_clf = RandomForestClassifier(random_state = 0)
    rf_param_grid = { "max_depth": [3, None],
                "min_samples_split": [40]
              }


    rf_search = HalvingGridSearchCV(rf_clf, rf_param_grid, scoring='accuracy', resource='n_estimators', max_resources=5,
                          random_state=0).fit(X_train, y_train)

    rf_model = rf_search.best_estimator_

    y_pred = rf_model.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred)
    recall = metrics.recall_score(y_test, y_pred)
    f1_score = metrics.f1_score(y_test, y_pred)
    print (rf_model)
    #print('log loss', rf_loss_bow, '\nacc', rf_acc_bow)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, rf_model
```

SVC

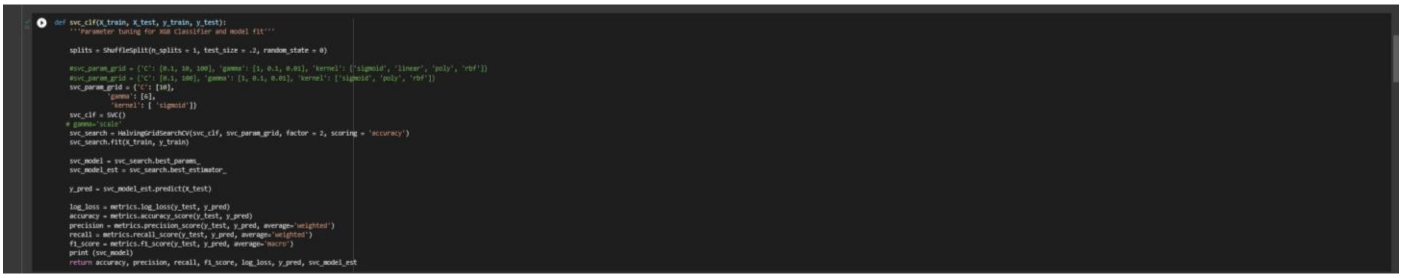


```
    splits = ShuffleSplit(n_splits = 1, test_size = .2, random_state = 0)

    #svc_param_grid = {'C': [0.1, 10, 100], 'gamma': [1, 0.1, 0.01], 'kernel': ['sigmoid', 'linear', 'poly', 'rbf']}
    #svc_param_grid = {'C': [0.1, 100], 'gamma': [1, 0.1, 0.01], 'kernel': ['sigmoid', 'poly', 'rbf']}
    svc_param_grid = {'C': [10],
     #         'gamma': [1,0.1,0.01,0.001,0.0001],
              'kernel': [ 'sigmoid']}
    svc_clf = SVC()
   # gamma='scale'
    svc_search = HalvingGridSearchCV(svc_clf, svc_param_grid, factor = 2, scoring = 'accuracy')
    svc_search.fit(X_train, y_train)

    svc_model = svc_search.best_params_
    svc_model_est = svc_search.best_estimator_

    y_pred = svc_model_est.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred, average='weighted')
    recall = metrics.recall_score(y_test, y_pred, average='weighted')
    f1_score = metrics.f1_score(y_test, y_pred, average='macro')
    print (svc_model)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, svc_model_est
```

```
## SVC - using param grid - 100, 30, 3
acc_svc_lemma, precision_svc_lemma, recall_svc_lemma, f1_score_svc_lemma, log_loss_svc_lemma, y_pred_svc_lemma, svc_model_lemma = svc_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_svc_lemma = classification_report(y_pred_svc_lemma, y_test_lemma)
matrix_svc_lemma = confusion_matrix(y_pred_svc_lemma, y_test_lemma)

print('Classification Report - SVC\n', report_svc_lemma)
print(
print('Confusion Matrix - SVC\n', matrix_svc_lemma)
```

```
{'C': 1, 'gamma': 0.01, 'kernel': 'rbf'}
Classification Report - SVC
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      4733
           1       1.00      1.00      1.00      2767

    accuracy                           1.00      7500
   macro avg       1.00      1.00      1.00      7500
weighted avg       1.00      1.00      1.00      7500


Confusion Matrix - SVC
 [[4733    0]
 [   0 2767]]
```

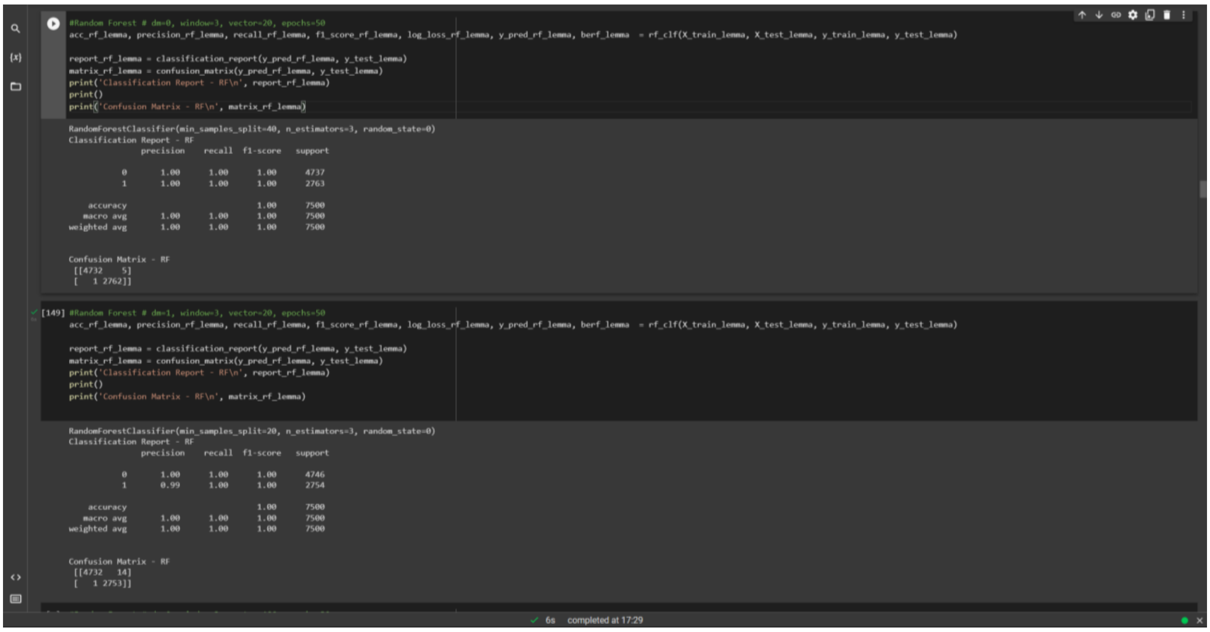
SELECTED



```
def svc_clf(X_train, X_test, y_train, y_test):
    '''Parameter tuning for XGB Classifier and model fit'''

    splits = ShuffleSplit(n_splits = 1, test_size = .2, random_state = 0)

    #svc_param_grid = {'C': [0.1, 10, 100], 'gamma': [1, 0.1, 0.01], 'kernel': ['sigmoid', 'linear', 'poly', 'rbf']}
    #svc_param_grid = {'C': [0.1, 100], 'gamma': [1, 0.1, 0.01], 'kernel': ['sigmoid', 'poly', 'rbf']}
    svc_param_grid = {'C': [10],
             'gamma': [6],
              'kernel': [ 'sigmoid']}
    svc_clf = SVC()
   # gamma='scale'
    svc_search = HalvingGridSearchCV(svc_clf, svc_param_grid, factor = 2, scoring = 'accuracy')
    svc_search.fit(X_train, y_train)

    svc_model = svc_search.best_params_
    svc_model_est = svc_search.best_estimator_

    y_pred = svc_model_est.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred, average='weighted')
    recall = metrics.recall_score(y_test, y_pred, average='weighted')
    f1_score = metrics.f1_score(y_test, y_pred, average='macro')
    print (svc_model)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, svc_model_est
```

DOC2VEC – RANDOM FOREST

# VECTOR 20



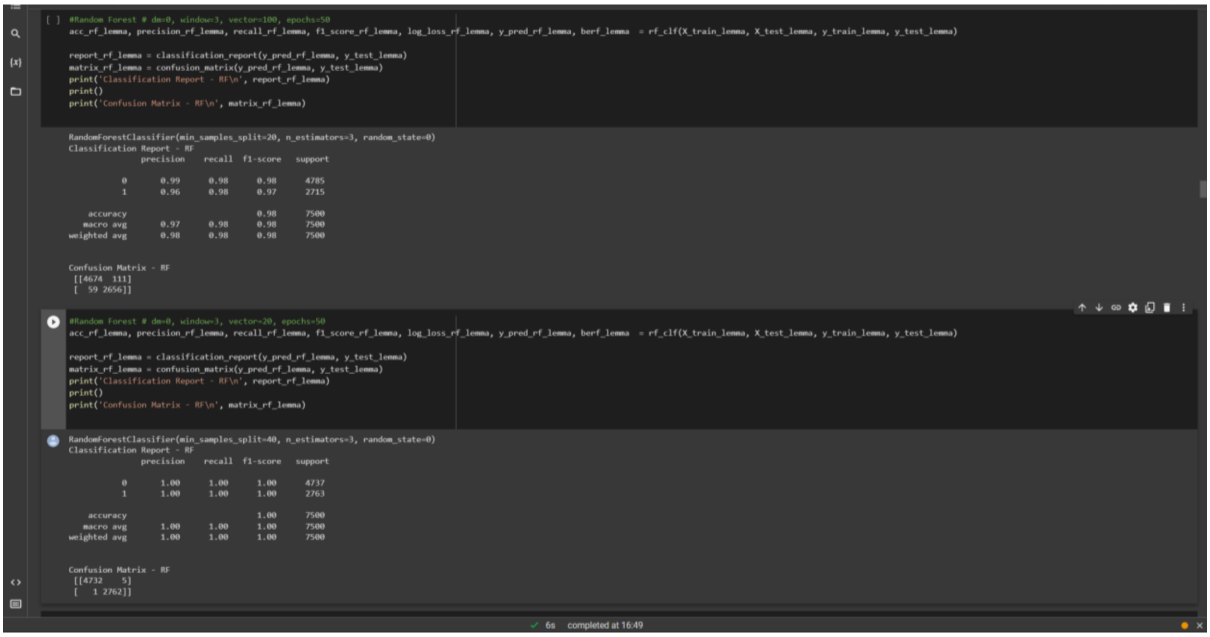
```
#Random Forest # dm=0, window=3, vector=20, epochs=50
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(min_samples_split=40, n_estimators=3, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4737
           1       1.00      1.00      1.00      2763

    accuracy                           1.00      7500
   macro avg       1.00      1.00      1.00      7500
weighted avg       1.00      1.00      1.00      7500


Confusion Matrix - RF
 [[4732    5]
 [   1 2762]]
```

```
[149] #Random Forest # dm=1, window=3, vector=20, epochs=50
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(min_samples_split=20, n_estimators=3, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4746
           1       0.99      1.00      1.00      2754

    accuracy                           1.00      7500
   macro avg       1.00      1.00      1.00      7500
weighted avg       1.00      1.00      1.00      7500


Confusion Matrix - RF
 [[4732   14]
 [   1 2753]]
```

6s completed at 17:29



```
[ ] #Random Forest # dm=0, window=3, vector=100, epochs=50
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

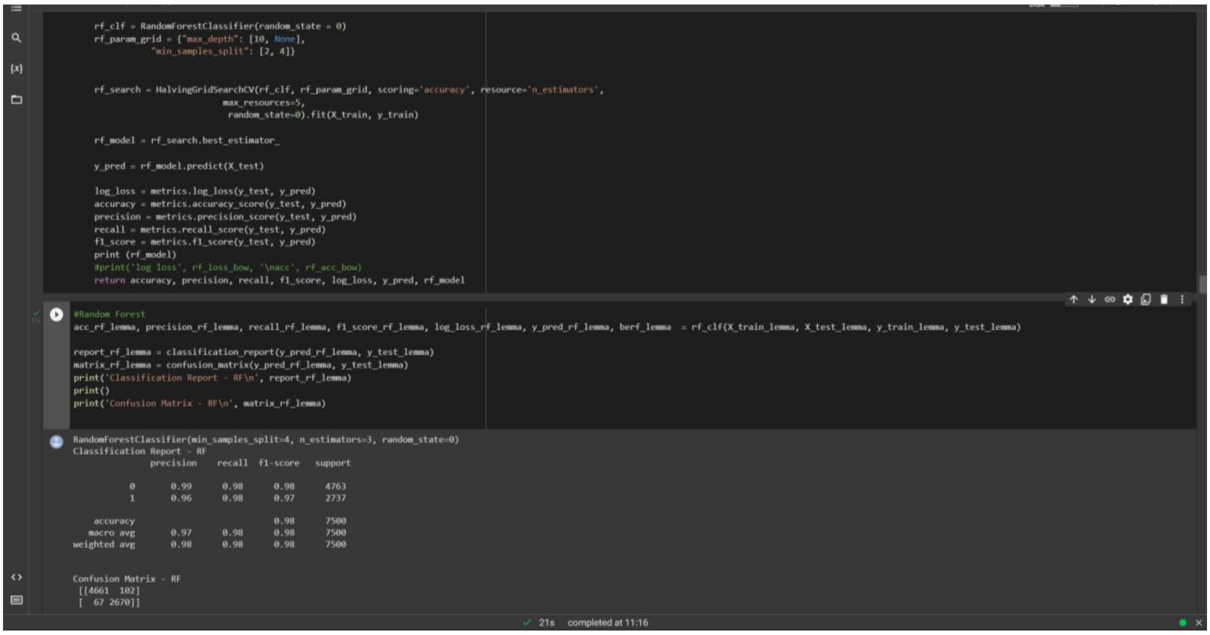
```
RandomForestClassifier(min_samples_split=20, n_estimators=3, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      4785
           1       0.96      0.98      0.97      2715

    accuracy                           0.98      7500
   macro avg       0.97      0.98      0.98      7500
weighted avg       0.98      0.98      0.98      7500


Confusion Matrix - RF
 [[4674  111]
 [  59 2656]]
```

```
#Random Forest # dm=0, window=3, vector=20, epochs=50
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(min_samples_split=40, n_estimators=3, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4737
           1       1.00      1.00      1.00      2763

    accuracy                           1.00      7500
   macro avg       1.00      1.00      1.00      7500
weighted avg       1.00      1.00      1.00      7500


Confusion Matrix - RF
 [[4732    5]
 [   1 2762]]
```

6s completed at 16:49

DM = 1 - VECTOR=50 – WINDOW= 3 - EPOCHS=30

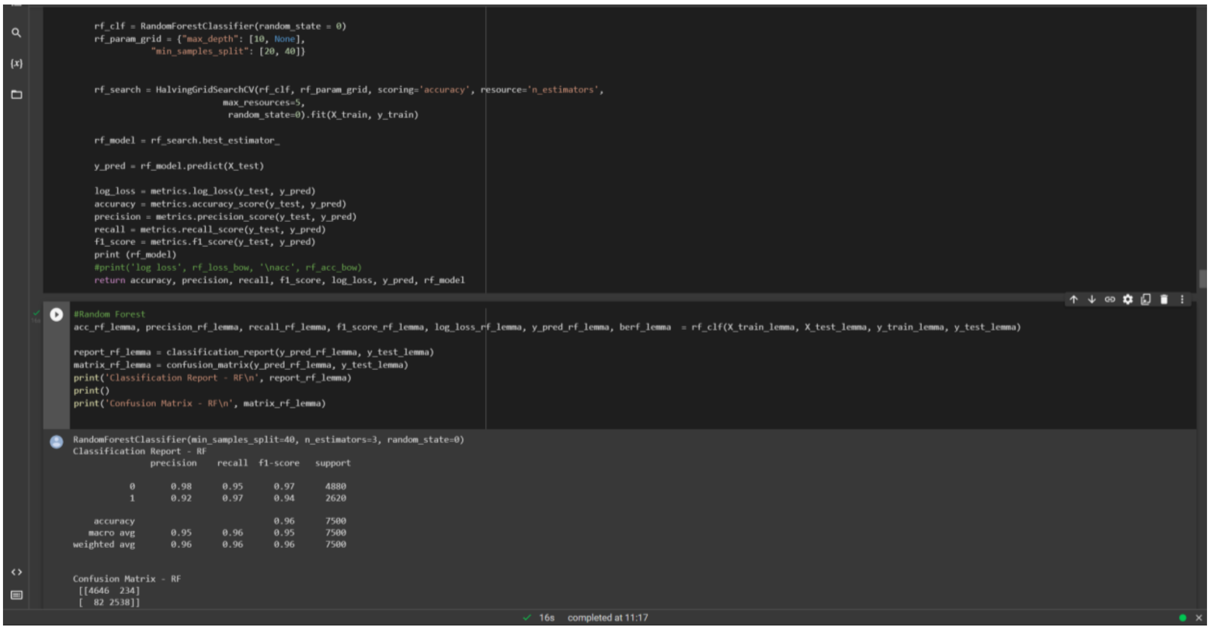


```
    rf_clf = RandomForestClassifier(random_state = 0)
    rf_param_grid = {"max_depth": [10, None],
                "min_samples_split": [2, 4]}


    rf_search = HalvingGridSearchCV(rf_clf, rf_param_grid, scoring='accuracy', resource='n_estimators',
                             max_resources=5,
                              random_state=0).fit(X_train, y_train)

    rf_model = rf_search.best_estimator_

    y_pred = rf_model.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred)
    recall = metrics.recall_score(y_test, y_pred)
    f1_score = metrics.f1_score(y_test, y_pred)
    print (rf_model)
    #print('log loss', rf_loss_bow, '\nacc', rf_acc_bow)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, rf_model
```

```
#Random Forest
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(min_samples_split=4, n_estimators=3, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      4763
           1       0.96      0.98      0.97      2737

    accuracy                           0.98      7500
   macro avg       0.97      0.98      0.98      7500
weighted avg       0.98      0.98      0.98      7500


Confusion Matrix - RF
 [[4661  102]
 [  67 2670]]
```

✓ 21s completed at 11:16

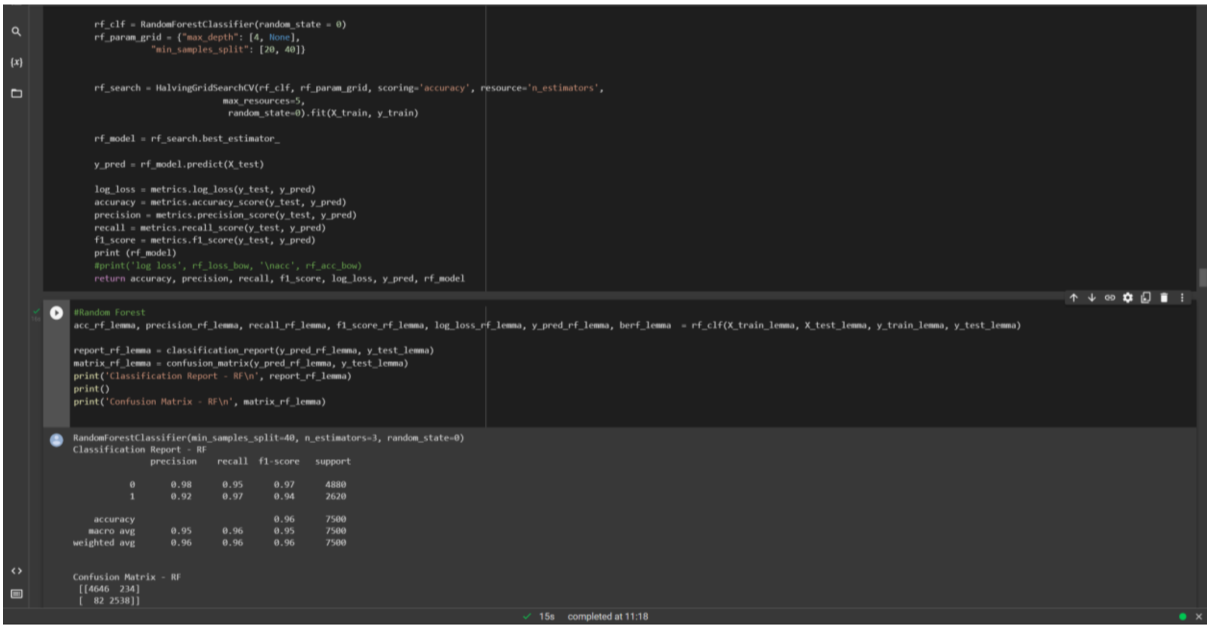


```
    rf_clf = RandomForestClassifier(random_state = 0)
    rf_param_grid = {"max_depth": [10, None],
                "min_samples_split": [20, 40]}


    rf_search = HalvingGridSearchCV(rf_clf, rf_param_grid, scoring='accuracy', resource='n_estimators',
                             max_resources=5,
                              random_state=0).fit(X_train, y_train)

    rf_model = rf_search.best_estimator_

    y_pred = rf_model.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred)
    recall = metrics.recall_score(y_test, y_pred)
    f1_score = metrics.f1_score(y_test, y_pred)
    print (rf_model)
    #print('log loss', rf_loss_bow, '\nacc', rf_acc_bow)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, rf_model
```

```
#Random Forest
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(min_samples_split=40, n_estimators=3, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.98      0.95      0.97      4880
           1       0.92      0.97      0.94      2620

    accuracy                           0.96      7500
   macro avg       0.95      0.96      0.95      7500
weighted avg       0.96      0.96      0.96      7500


Confusion Matrix - RF
 [[4646  234]
 [  82 2538]]
```

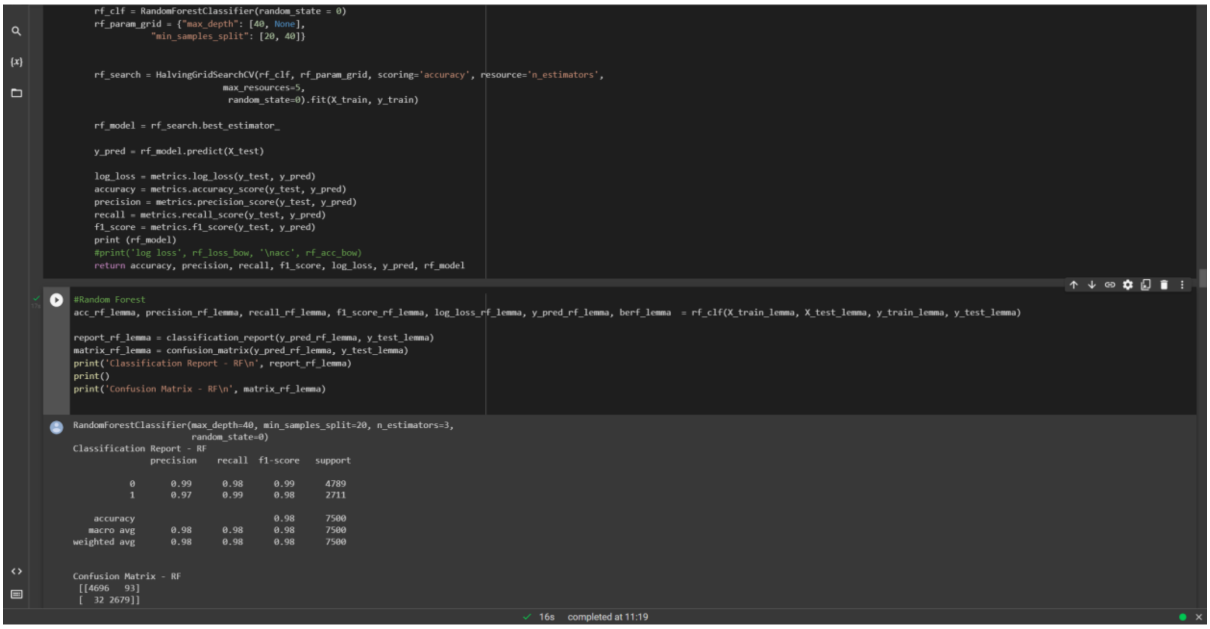
✓ 16s completed at 11:17



```
    rf_clf = RandomForestClassifier(random_state = 0)
    rf_param_grid = {"max_depth": [4, None],
              "min_samples_split": [20, 40]}


    rf_search = HalvingGridSearchCV(rf_clf, rf_param_grid, scoring='accuracy', resource='n_estimators',
                             max_resources=5,
                              random_state=0).fit(X_train, y_train)

    rf_model = rf_search.best_estimator_

    y_pred = rf_model.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred)
    recall = metrics.recall_score(y_test, y_pred)
    f1_score = metrics.f1_score(y_test, y_pred)
    print (rf_model)
    #print('log loss', rf_loss_bow, '\nacc', rf_acc_bow)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, rf_model
```

```
#Random Forest
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(min_samples_split=40, n_estimators=3, random_state=0)
Classification Report - RF
               precision    recall  f1-score   support

           0       0.98      0.95      0.97      4880
           1       0.92      0.97      0.94      2620

    accuracy                           0.96      7500
   macro avg       0.95      0.96      0.95      7500
weighted avg       0.96      0.96      0.96      7500


Confusion Matrix - RF
 [[4646  234]
 [  82 2538]]
```

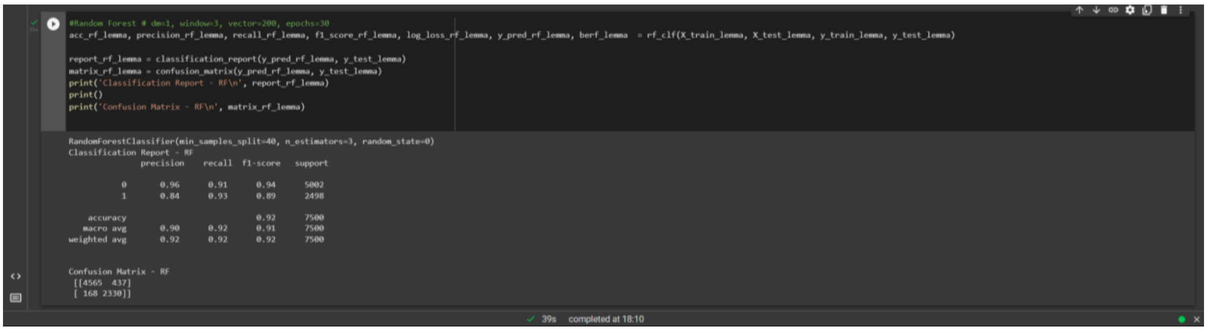


```
    rf_clf = RandomForestClassifier(random_state = 0)
    rf_param_grid = {"max_depth": [40, None],
              "min_samples_split": [20, 40]}


    rf_search = HalvingGridSearchCV(rf_clf, rf_param_grid, scoring='accuracy', resource='n_estimators',
                             max_resources=5,
                              random_state=0).fit(X_train, y_train)

    rf_model = rf_search.best_estimator_

    y_pred = rf_model.predict(X_test)

    log_loss = metrics.log_loss(y_test, y_pred)
    accuracy = metrics.accuracy_score(y_test, y_pred)
    precision = metrics.precision_score(y_test, y_pred)
    recall = metrics.recall_score(y_test, y_pred)
    f1_score = metrics.f1_score(y_test, y_pred)
    print (rf_model)
    #print('log loss', rf_loss_bow, '\nacc', rf_acc_bow)
    return accuracy, precision, recall, f1_score, log_loss, y_pred, rf_model
```

```
#Random Forest
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(max_depth=40, min_samples_split=20, n_estimators=3,
                       random_state=0)
Classification Report - RF
               precision    recall  f1-score   support

           0       0.99      0.98      0.99      4789
           1       0.97      0.99      0.98      2711

    accuracy                           0.98      7500
   macro avg       0.98      0.98      0.98      7500
weighted avg       0.98      0.98      0.98      7500


Confusion Matrix - RF
 [[4696   93]
 [  32 2679]]
```

VECTOR 100 VS 200 (DM = 1 – WINDOW = 3, EPOCHS = 30)

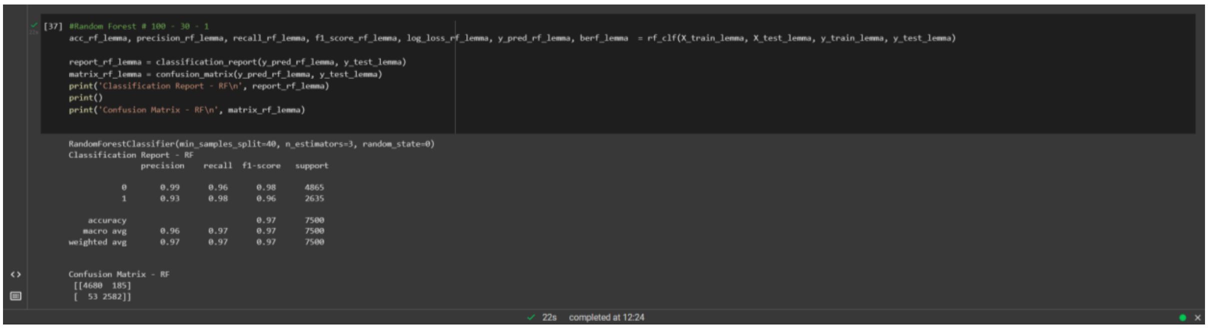


```
#Random Forest # dm=1, window=3, vector=200, epochs=30
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(min_samples_split=40, n_estimators=3, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.96      0.91      0.94      5002
           1       0.84      0.93      0.89      2498

    accuracy                           0.92      7500
   macro avg       0.90      0.92      0.91      7500
weighted avg       0.92      0.92      0.92      7500


Confusion Matrix - RF
 [[4565  437]
 [ 168 2330]]
```

✓ 39s completed at 18:10

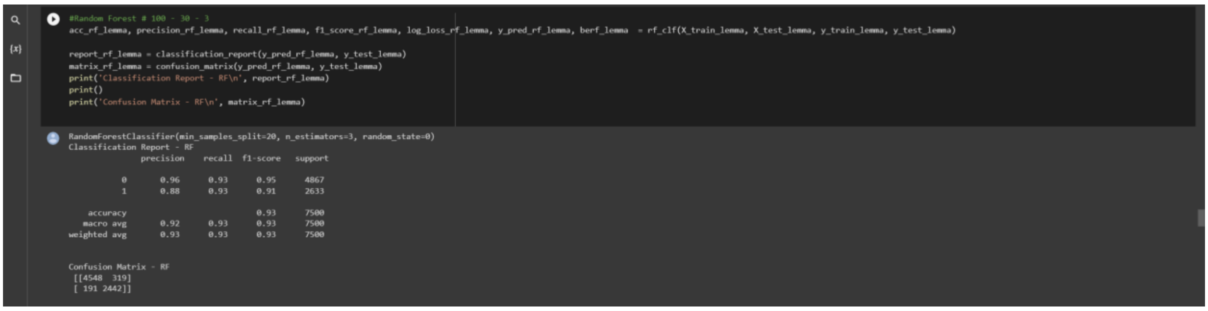
# CHANGING NUMBER OF WINDOW



```
[37] #Random Forest # 100 - 30 - 1
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

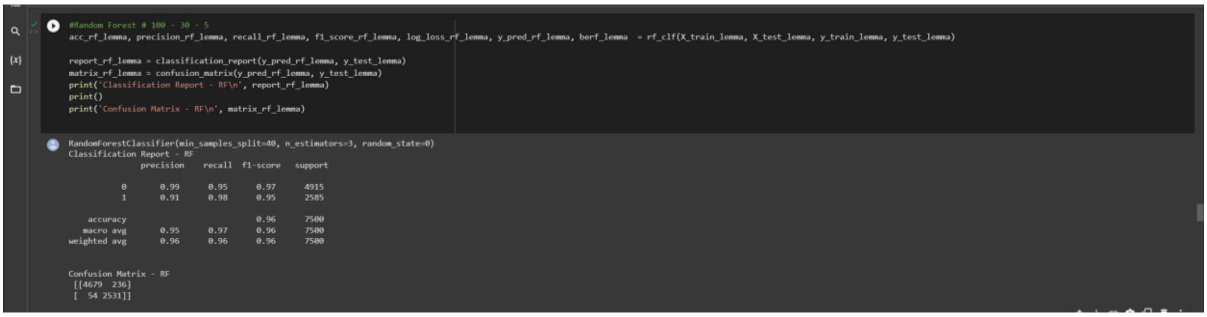
```
RandomForestClassifier(min_samples_split=40, n_estimators=3, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.99      0.96      0.98      4865
           1       0.93      0.98      0.96      2635

    accuracy                           0.97      7500
   macro avg       0.96      0.97      0.97      7500
weighted avg       0.97      0.97      0.97      7500


Confusion Matrix - RF
 [[4680  185]
 [  53 2582]]
```

✓ 22s completed at 12:24



```
#Random Forest # 100 - 30 - 3
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(min_samples_split=20, n_estimators=3, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.96      0.93      0.95      4867
           1       0.88      0.93      0.91      2633

    accuracy                           0.93      7500
   macro avg       0.92      0.93      0.93      7500
weighted avg       0.93      0.93      0.93      7500


Confusion Matrix - RF
 [[4548  319]
 [ 191 2442]]
```



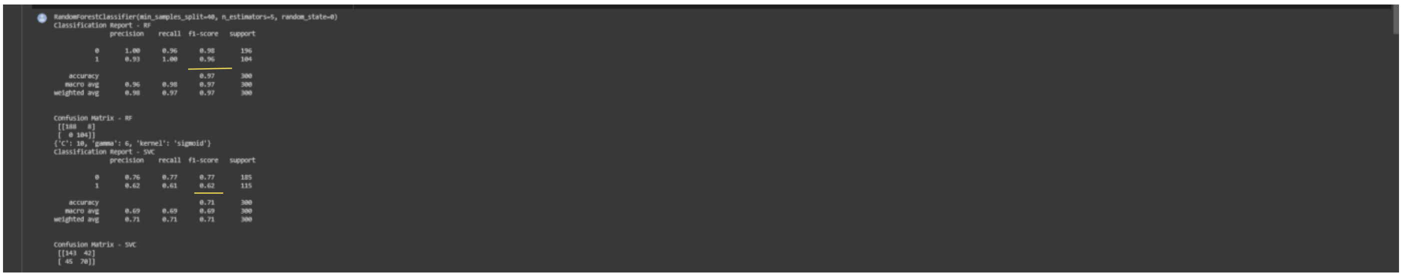
```
#Random Forest # 100 - 30 - 5
acc_rf_lemma, precision_rf_lemma, recall_rf_lemma, f1_score_rf_lemma, log_loss_rf_lemma, y_pred_rf_lemma, berf_lemma  = rf_clf(X_train_lemma, X_test_lemma, y_train_lemma, y_test_lemma)

report_rf_lemma = classification_report(y_pred_rf_lemma, y_test_lemma)
matrix_rf_lemma = confusion_matrix(y_pred_rf_lemma, y_test_lemma)
print('Classification Report - RF\n', report_rf_lemma)
print()
print('Confusion Matrix - RF\n', matrix_rf_lemma)
```

```
RandomForestClassifier(min_samples_split=40, n_estimators=3, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.99      0.95      0.97      4915
           1       0.91      0.98      0.95      2585

    accuracy                           0.96      7500
   macro avg       0.95      0.97      0.96      7500
weighted avg       0.96      0.96      0.96      7500


Confusion Matrix - RF
 [[4679  236]
 [  54 2531]]
```

RESULTS

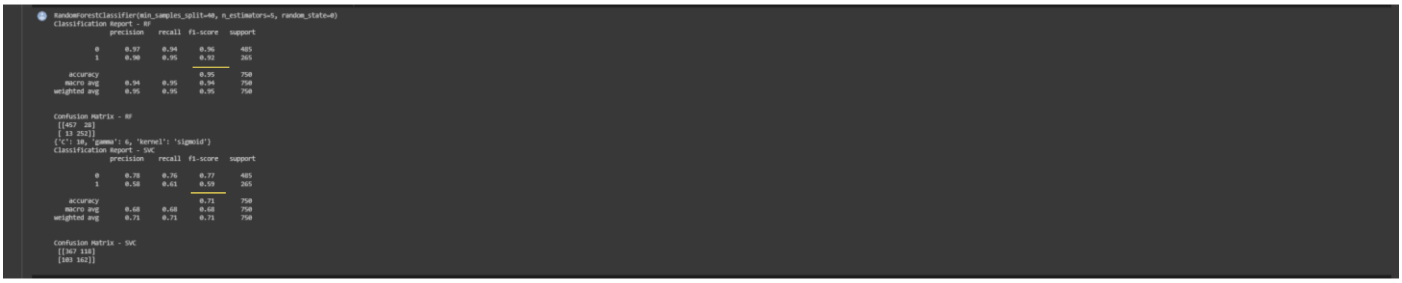
Lemmatized without stopwords

Not tokenized – 100 vectors



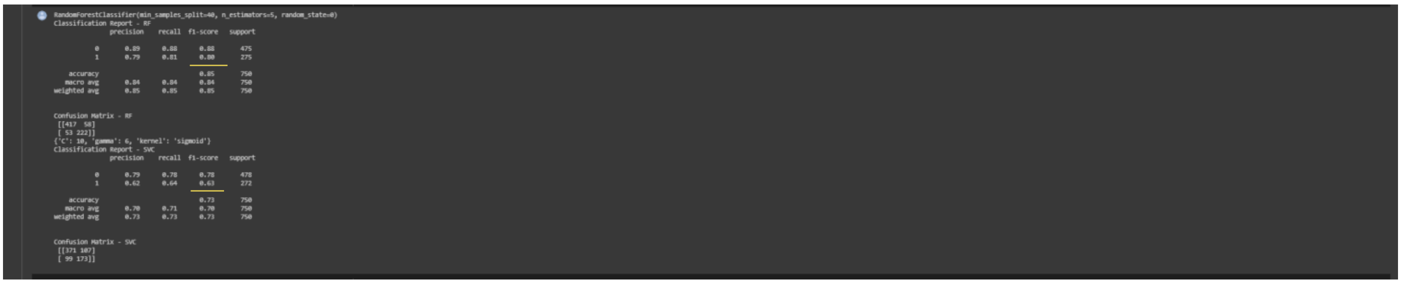
```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       1.00      0.96      0.98       196
           1       0.93      1.00      0.96       104

    accuracy                           0.97       300
   macro avg       0.96      0.98      0.97       300
weighted avg       0.98      0.97      0.97       300


Confusion Matrix - RF
 [[188   8]
 [  0 104]]
{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.76      0.77      0.77       185
           1       0.62      0.61      0.62       115

    accuracy                           0.71       300
   macro avg       0.69      0.69      0.69       300
weighted avg       0.71      0.71      0.71       300


Confusion Matrix - SVC
 [[143  42]
 [ 45  70]]
```

Tokenized – 100 VECTORS



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.97      0.94      0.96       485
           1       0.90      0.95      0.92       265

    accuracy                           0.95       750
   macro avg       0.94      0.95      0.94       750
weighted avg       0.95      0.95      0.95       750


Confusion Matrix - RF
 [[457  28]
 [ 13 252]]
{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.78      0.76      0.77       485
           1       0.58      0.61      0.59       265

    accuracy                           0.71       750
   macro avg       0.68      0.68      0.68       750
weighted avg       0.71      0.71      0.71       750


Confusion Matrix - SVC
 [[367 118]
 [103 162]]
```

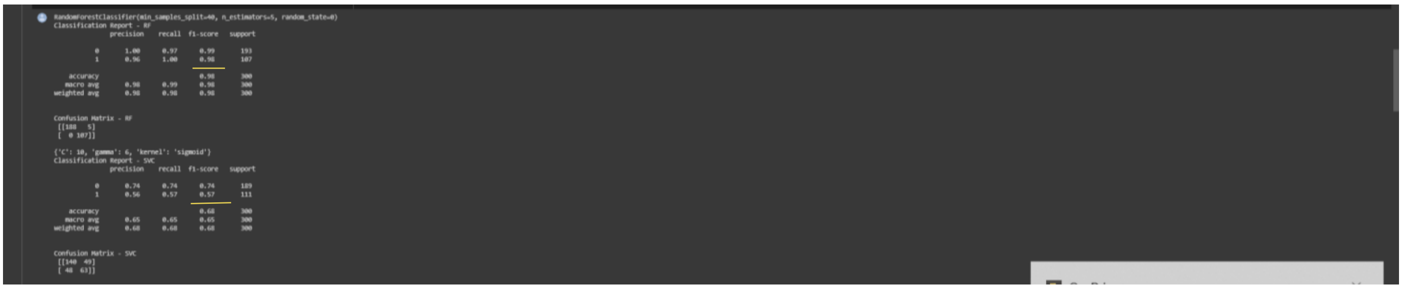
Tokenized – 200 vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.89      0.88      0.88       475
           1       0.79      0.81      0.80       275

    accuracy                           0.85       750
   macro avg       0.84      0.84      0.84       750
weighted avg       0.85      0.85      0.85       750


Confusion Matrix - RF
 [[417  58]
 [ 53 222]]
{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.79      0.78      0.78       478
           1       0.62      0.64      0.63       272

    accuracy                           0.73       750
   macro avg       0.70      0.71      0.70       750
weighted avg       0.73      0.73      0.73       750


Confusion Matrix - SVC
 [[371 107]
 [ 99 173]]
```

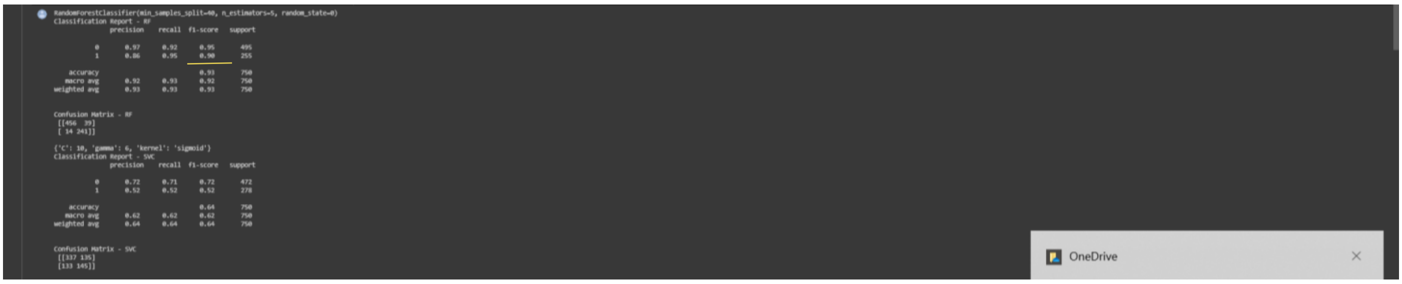
Lemmatized with stopwords

Not tokenized – 100 vectors



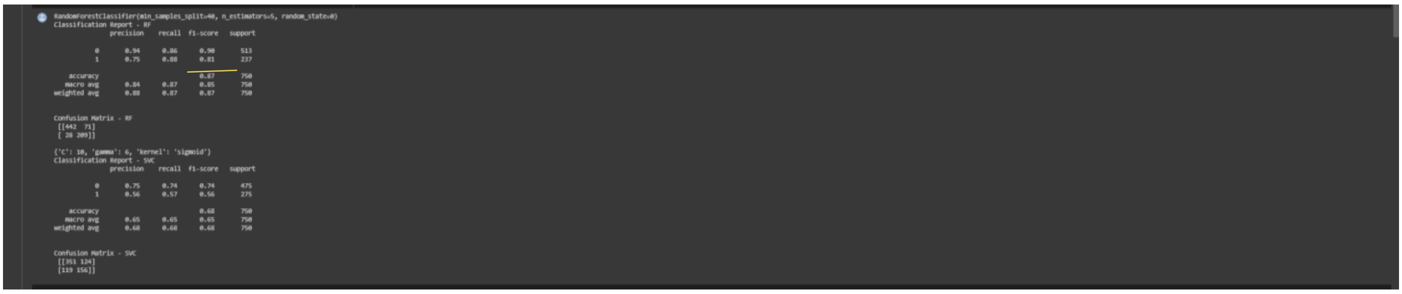
```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       1.00      0.97      0.99       193
           1       0.96      1.00      0.98       107

    accuracy                           0.98       300
   macro avg       0.98      0.99      0.98       300
weighted avg       0.98      0.98      0.98       300


Confusion Matrix - RF
 [[188   5]
 [  0 107]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.74      0.74      0.74       189
           1       0.56      0.57      0.57       111

    accuracy                           0.68       300
   macro avg       0.65      0.65      0.65       300
weighted avg       0.68      0.68      0.68       300


Confusion Matrix - SVC
 [[140  49]
 [ 48  63]]
```

Tokenized – 100 vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.97      0.92      0.95       495
           1       0.86      0.95      0.90       255

    accuracy                           0.93       750
   macro avg       0.92      0.93      0.92       750
weighted avg       0.93      0.93      0.93       750


Confusion Matrix - RF
 [[456  39]
 [ 14 241]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.72      0.71      0.72       472
           1       0.52      0.52      0.52       278

    accuracy                           0.64       750
   macro avg       0.62      0.62      0.62       750
weighted avg       0.64      0.64      0.64       750


Confusion Matrix - SVC
 [[337 135]
 [133 145]]
```

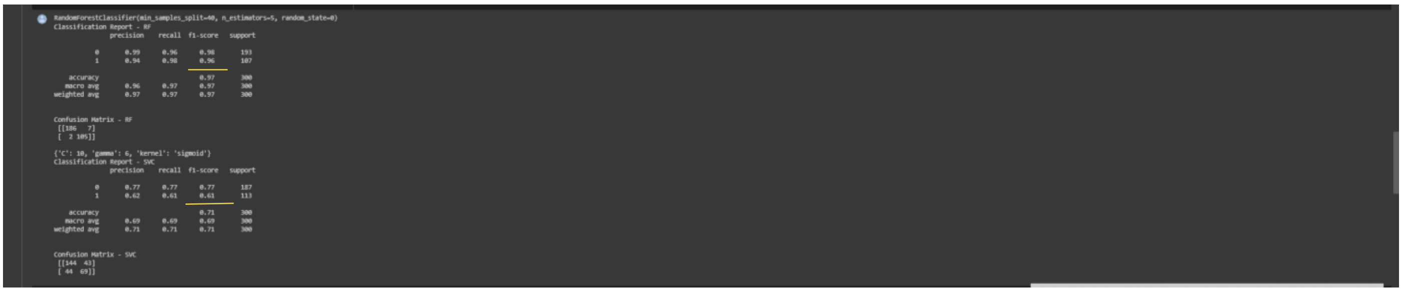
Tokenized – 200 vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.94      0.86      0.90       513
           1       0.75      0.88      0.81       237

    accuracy                           0.87       750
   macro avg       0.84      0.87      0.85       750
weighted avg       0.88      0.87      0.87       750


Confusion Matrix - RF
 [[442  71]
 [ 28 209]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.75      0.74      0.74       475
           1       0.56      0.57      0.56       275

    accuracy                           0.68       750
   macro avg       0.65      0.65      0.65       750
weighted avg       0.68      0.68      0.68       750


Confusion Matrix - SVC
 [[351 124]
 [119 156]]
```

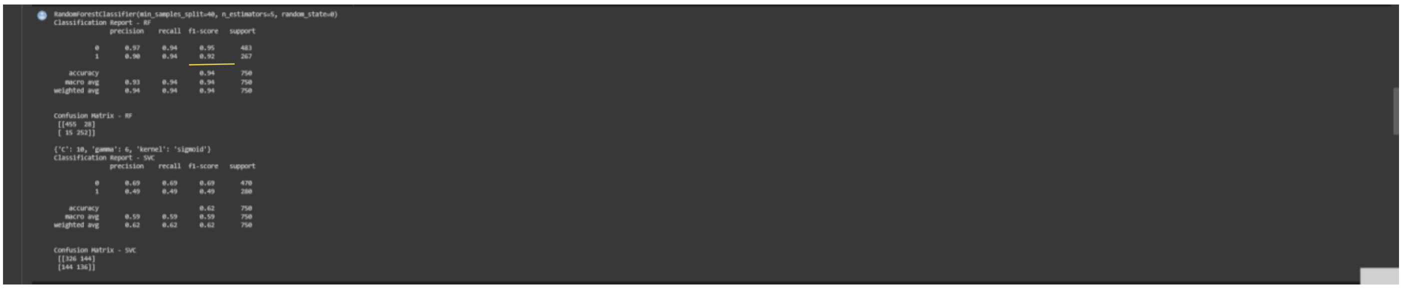
With stopwords, not lemmatized nor stemmed

Not tokenized – 100 vectors



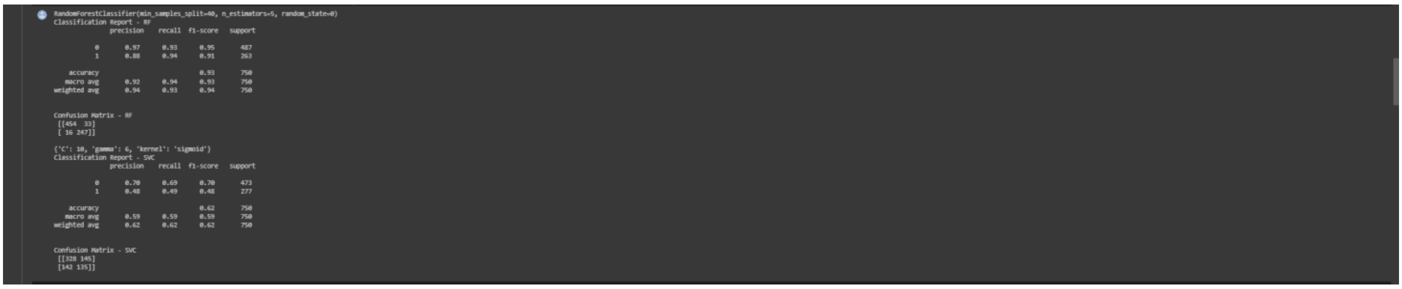
```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.99      0.96      0.98       193
           1       0.94      0.98      0.96       107

    accuracy                           0.97       300
   macro avg       0.96      0.97      0.97       300
weighted avg       0.97      0.97      0.97       300


Confusion Matrix - RF
 [[186   7]
 [  2 105]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.77      0.77      0.77       187
           1       0.62      0.61      0.61       113

    accuracy                           0.71       300
   macro avg       0.69      0.69      0.69       300
weighted avg       0.71      0.71      0.71       300


Confusion Matrix - SVC
 [[144  43]
 [ 44  69]]
```

Tokenized – 100 vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.97      0.94      0.95       483
           1       0.90      0.94      0.92       267

    accuracy                           0.94       750
   macro avg       0.93      0.94      0.94       750
weighted avg       0.94      0.94      0.94       750


Confusion Matrix - RF
 [[455  28]
 [ 15 252]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.69      0.69      0.69       470
           1       0.49      0.49      0.49       280

    accuracy                           0.62       750
   macro avg       0.59      0.59      0.59       750
weighted avg       0.62      0.62      0.62       750


Confusion Matrix - SVC
 [[326 144]
 [144 136]]
```

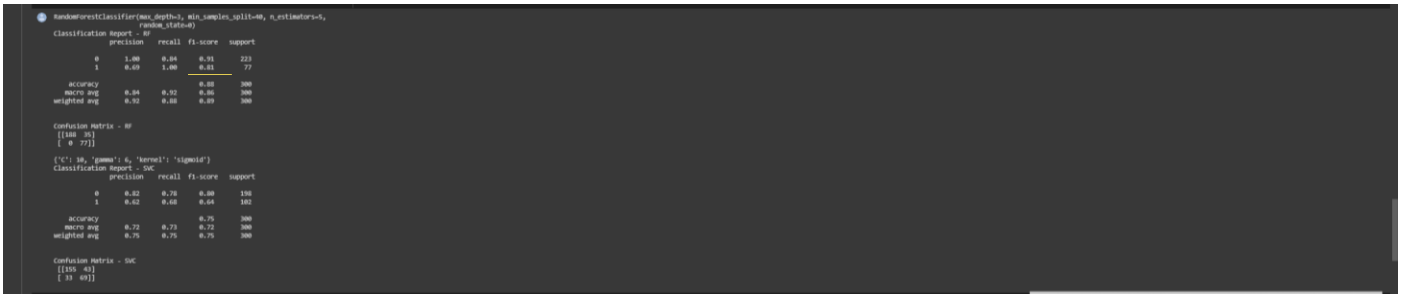
Tokenized – 200 vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.97      0.93      0.95       487
           1       0.88      0.94      0.91       263

    accuracy                           0.93       750
   macro avg       0.92      0.94      0.93       750
weighted avg       0.94      0.93      0.94       750


Confusion Matrix - RF
 [[454  33]
 [ 16 247]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.70      0.69      0.70       473
           1       0.48      0.49      0.48       277

    accuracy                           0.62       750
   macro avg       0.59      0.59      0.59       750
weighted avg       0.62      0.62      0.62       750


Confusion Matrix - SVC
 [[328 145]
 [142 135]]
```

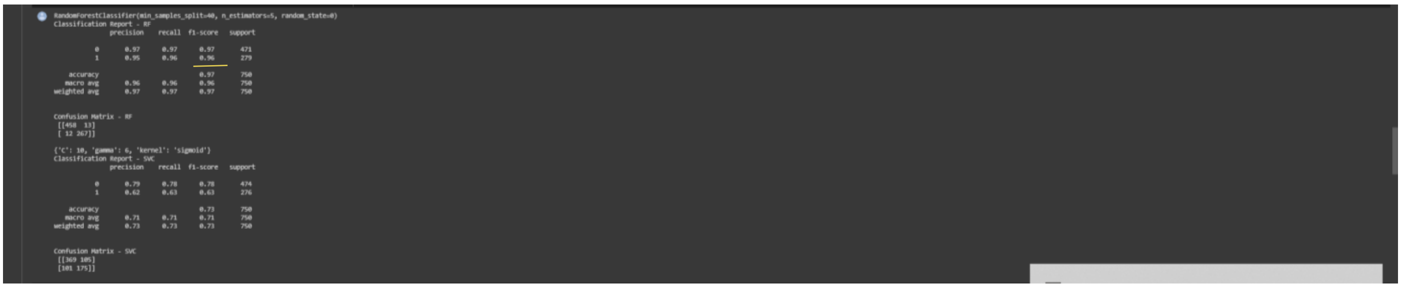
Without stopwords, not lemmatized nor stemmed

No tokenized – 100 vectors



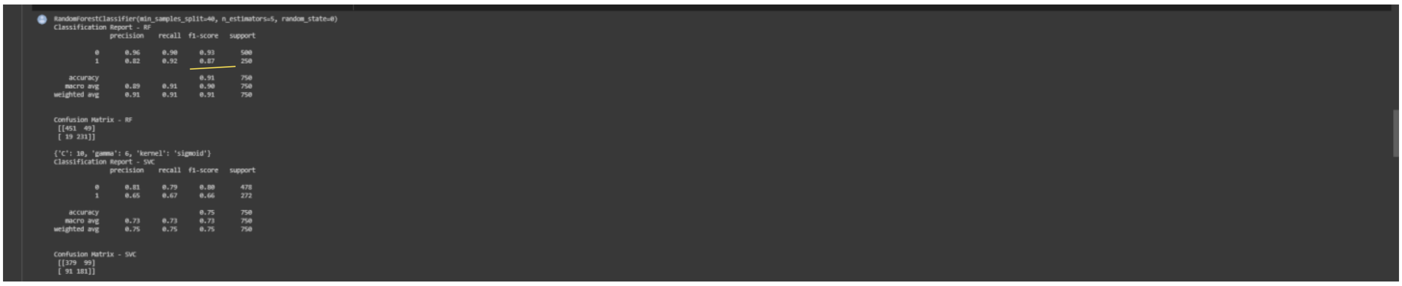
```
RandomForestClassifier(max_depth=3, min_samples_split=40, n_estimators=5,
                       random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       1.00      0.84      0.91       223
           1       0.69      1.00      0.81        77

    accuracy                           0.88       300
   macro avg       0.84      0.92      0.86       300
weighted avg       0.92      0.88      0.89       300


Confusion Matrix - RF
 [[188  35]
 [  0  77]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.82      0.78      0.80       198
           1       0.62      0.68      0.64       102

    accuracy                           0.75       300
   macro avg       0.72      0.73      0.72       300
weighted avg       0.75      0.75      0.75       300


Confusion Matrix - SVC
 [[155  43]
 [ 33  69]]
```

Tokenized - vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.97      0.97      0.97       471
           1       0.95      0.96      0.96       279

    accuracy                           0.97       750
   macro avg       0.96      0.96      0.96       750
weighted avg       0.97      0.97      0.97       750


Confusion Matrix - RF
 [[458  13]
 [ 12 267]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.79      0.78      0.78       474
           1       0.62      0.63      0.63       276

    accuracy                           0.73       750
   macro avg       0.71      0.71      0.71       750
weighted avg       0.73      0.73      0.73       750


Confusion Matrix - SVC
 [[369 105]
 [101 175]]
```

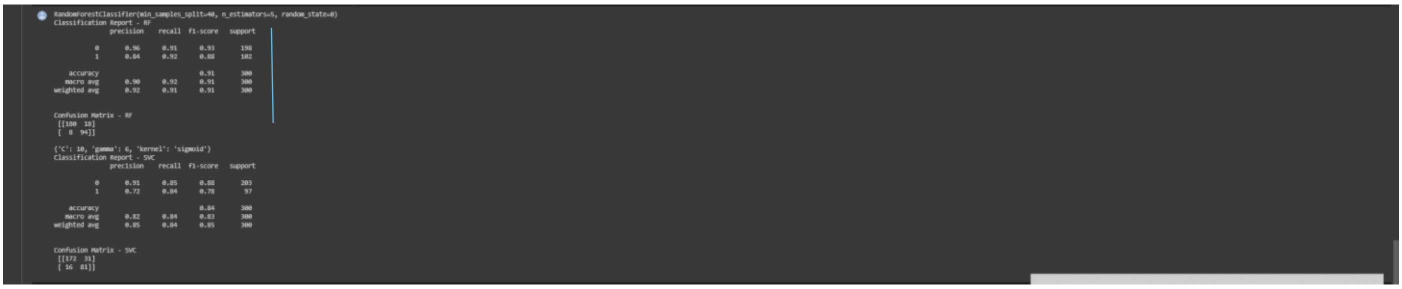
Tokenized – 200 vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.96      0.90      0.93       500
           1       0.82      0.92      0.87       250

    accuracy                           0.91       750
   macro avg       0.89      0.91      0.90       750
weighted avg       0.91      0.91      0.91       750


Confusion Matrix - RF
 [[451  49]
 [ 19 231]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.81      0.79      0.80       478
           1       0.65      0.67      0.66       272

    accuracy                           0.75       750
   macro avg       0.73      0.73      0.73       750
weighted avg       0.75      0.75      0.75       750


Confusion Matrix - SVC
 [[379  99]
 [ 91 181]]
```

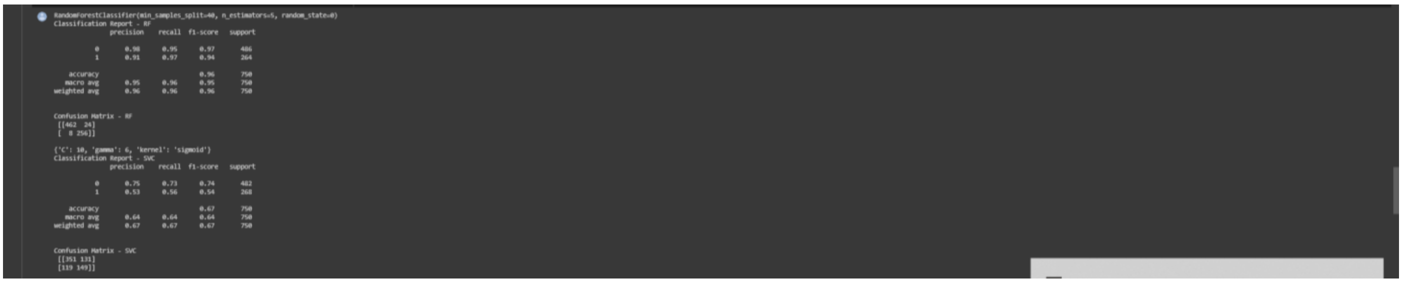
Stemmed with stopwords

Not tokenized – 100 vectors



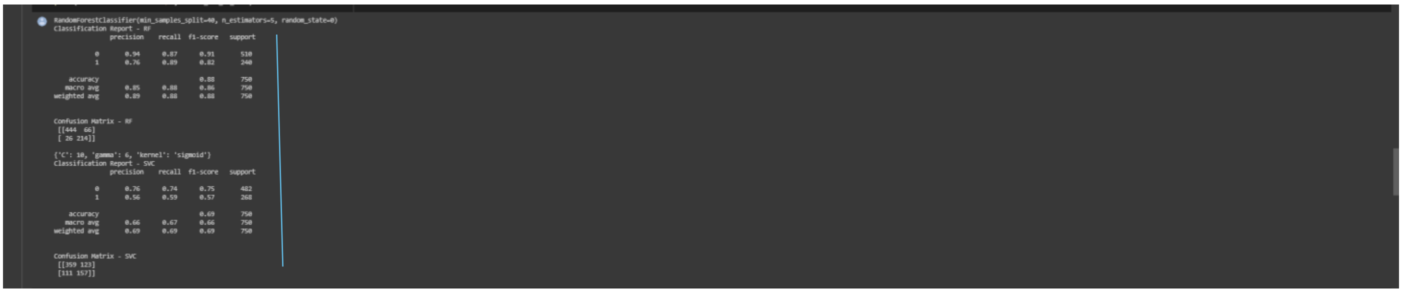
```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.96      0.91      0.93       198
           1       0.84      0.92      0.88       102

    accuracy                           0.91       300
   macro avg       0.90      0.92      0.91       300
weighted avg       0.92      0.91      0.91       300


Confusion Matrix - RF
 [[180  18]
 [  8  94]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.91      0.85      0.88       203
           1       0.72      0.84      0.78        97

    accuracy                           0.84       300
   macro avg       0.82      0.84      0.83       300
weighted avg       0.85      0.84      0.85       300


Confusion Matrix - SVC
 [[172  31]
 [ 16  81]]
```

Tokenized – 100 vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.98      0.95      0.97       486
           1       0.91      0.97      0.94       264

    accuracy                           0.96       750
   macro avg       0.95      0.96      0.95       750
weighted avg       0.96      0.96      0.96       750


Confusion Matrix - RF
 [[462  24]
 [  8 256]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.75      0.73      0.74       482
           1       0.53      0.56      0.54       268

    accuracy                           0.67       750
   macro avg       0.64      0.64      0.64       750
weighted avg       0.67      0.67      0.67       750


Confusion Matrix - SVC
 [[351 131]
 [119 149]]
```

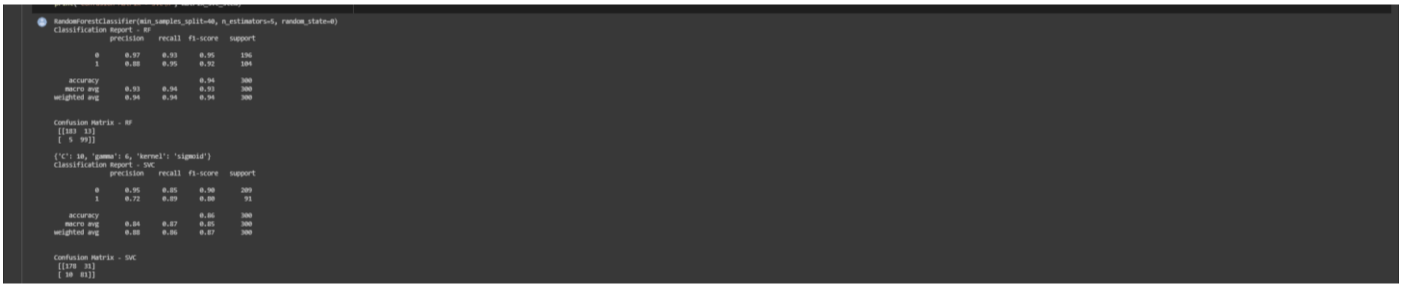
Tokenized – 200 vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.94      0.87      0.91       510
           1       0.76      0.89      0.82       240

    accuracy                           0.88       750
   macro avg       0.85      0.88      0.86       750
weighted avg       0.89      0.88      0.88       750


Confusion Matrix - RF
 [[444  66]
 [ 26 214]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.76      0.74      0.75       482
           1       0.56      0.59      0.57       268

    accuracy                           0.69       750
   macro avg       0.66      0.67      0.66       750
weighted avg       0.69      0.69      0.69       750


Confusion Matrix - SVC
 [[359 123]
 [111 157]]
```

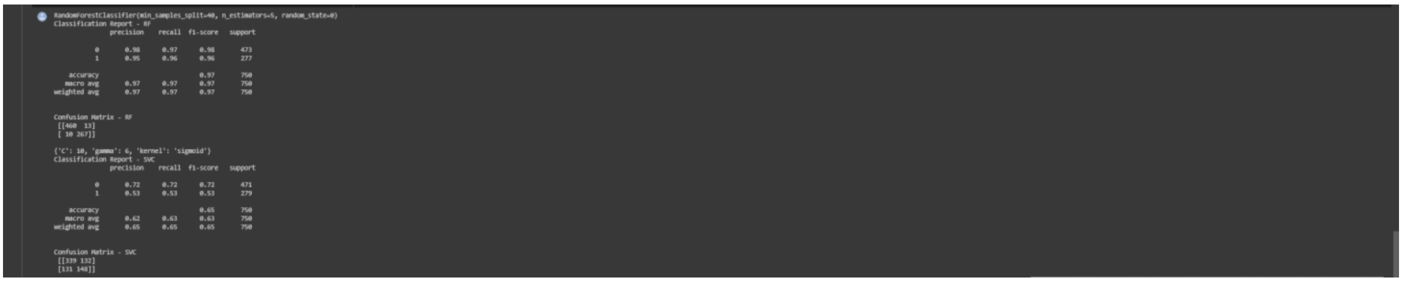
Stemmed without stopwords

No Tokenized – 100 vectors



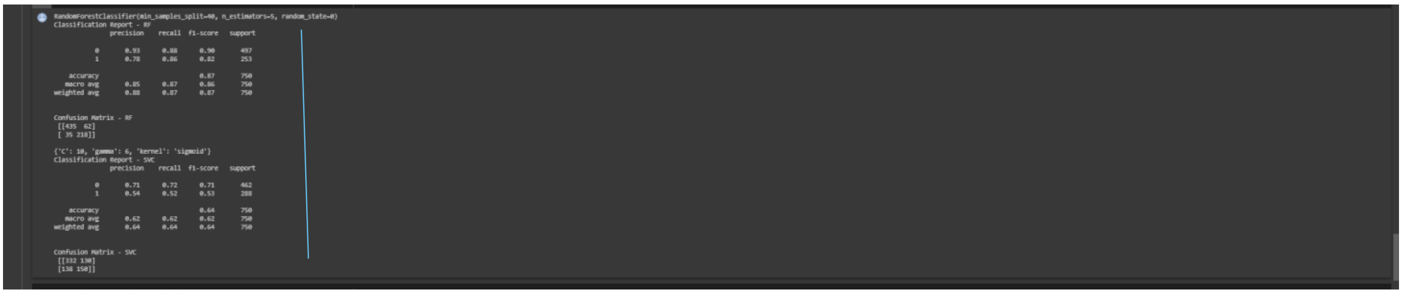
```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.97      0.93      0.95       196
           1       0.88      0.95      0.92       104

    accuracy                           0.94       300
   macro avg       0.93      0.94      0.93       300
weighted avg       0.94      0.94      0.94       300


Confusion Matrix - RF
 [[183  13]
 [  5  99]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.95      0.85      0.90       209
           1       0.72      0.89      0.80        91

    accuracy                           0.86       300
   macro avg       0.84      0.87      0.85       300
weighted avg       0.88      0.86      0.87       300


Confusion Matrix - SVC
 [[178  31]
 [ 10  81]]
```

Tokenized – 100 vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.98      0.97      0.98       473
           1       0.95      0.96      0.96       277

    accuracy                           0.97       750
   macro avg       0.97      0.97      0.97       750
weighted avg       0.97      0.97      0.97       750


Confusion Matrix - RF
 [[460  13]
 [ 10 267]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.72      0.72      0.72       471
           1       0.53      0.53      0.53       279

    accuracy                           0.65       750
   macro avg       0.62      0.63      0.63       750
weighted avg       0.65      0.65      0.65       750


Confusion Matrix - SVC
 [[339 132]
 [131 148]]
```

Tokenized – 200 vectors



```
RandomForestClassifier(min_samples_split=40, n_estimators=5, random_state=0)
Classification Report - RF
              precision    recall  f1-score   support

           0       0.93      0.88      0.90       497
           1       0.78      0.86      0.82       253

    accuracy                           0.87       750
   macro avg       0.85      0.87      0.86       750
weighted avg       0.88      0.87      0.87       750


Confusion Matrix - RF
 [[435  62]
 [ 35 218]]

{'C': 10, 'gamma': 6, 'kernel': 'sigmoid'}
Classification Report - SVC
              precision    recall  f1-score   support

           0       0.71      0.72      0.71       462
           1       0.54      0.52      0.53       288

    accuracy                           0.64       750
   macro avg       0.62      0.62      0.62       750
weighted avg       0.64      0.64      0.64       750


Confusion Matrix - SVC
 [[332 130]
 [138 150]]
```