

Configuration

In [1]:

```
# Parameters
ENABLE_COLAB = False

PROJECT_NAME = 'ML1010-Group-Project'
EXPERIMENT_NAME = 'ReviewText_Lemma_Bert2 (Random Forest)'
FILE_NAME = '01_ML1010_GP_RF_Bert2'
LOAD_FROM_EXP = False

#Root Machine Learning Directory. Projects appear underneath
GOOGLE_DRIVE_MOUNT = '/content/gdrive'
COLAB_ROOT_DIR = GOOGLE_DRIVE_MOUNT + '/MyDrive/Colab Notebooks'
COLAB_INIT_DIR = COLAB_ROOT_DIR + '/utility_files'

LOCAL_ROOT_DIR = '/home/magni//ML_Root/project_root'
LOCAL_INIT_DIR = LOCAL_ROOT_DIR + '/utility_files'
```

Bootstrap Environment

In [2]:

```
#add in support for utility file directory and importing
import sys
import os

if ENABLE_COLAB:
    #Need access to drive
    from google.colab import drive
    drive.mount(GOOGLE_DRIVE_MOUNT, force_remount=True)

    #add in utility directory to syspath to import
    INIT_DIR = COLAB_INIT_DIR
    sys.path.append(os.path.abspath(INIT_DIR))

    #Config environment variables
    ROOT_DIR = COLAB_ROOT_DIR

else:
    #add in utility directory to syspath to import
    INIT_DIR = LOCAL_INIT_DIR
    sys.path.append(os.path.abspath(INIT_DIR))

    #Config environment variables
    ROOT_DIR = LOCAL_ROOT_DIR

#Import Utility Support
from jarvis import Jarvis
jarvis = Jarvis(ROOT_DIR, PROJECT_NAME)

import mv_python_utils as mvutils
```

Wha...where am I?

I am awake now.

I have set your current working directory to /home/magni/ML_Root/project_root
 /ML1010-Group-Project
 The current time is 16:04
 Hello sir. Reminder, no more coffee.

Setup Runtime Environment

In [3]:

```
if ENABLE_COLAB:
    #!pip install scipy -q
    #!pip install scikit-learn -q
    #!pip install pycaret -q
    #!pip install matplotlib -q
    #!pip install joblib -q
    #!pip install pandasql -q
    !pip install umap_learn -q
    !pip install sentence_transformers -q
    !pip install spacytextblob -q
    !pip install flair -q
    display('Google Colab enabled')
else:
    display('Google Colab not enabled')

#Common imports
import json
import pandas as pd
import numpy as np
import matplotlib
import re
import nltk
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn import metrics
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split as tts
#from yellowbrick.classifier import ConfusionMatrix
#from sklearn.linear_model import LogisticRegression
from yellowbrick.target import ClassBalance
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier

nltk.download('stopwords')
%matplotlib inline
```

'Google Colab not enabled'

[nltk_data] Downloading package stopwords to /home/magni/nltk_data...
 [nltk_data] Package stopwords is already up-to-date!

```
In [4]: import importlib
import cw_df_metric_utils as cwutils
import DataPackage as dp
import DataPackageSupport as dps
import DataExperiment
import DataExperimentSupport
```

```
2022-01-22 16:04:42.836268: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open shared object file: No such file or directory
2022-01-22 16:04:42.836299: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
```

```
In [5]: importlib.reload(dp)
importlib.reload(dps)
importlib.reload(DataExperiment)
importlib.reload(DataExperimentSupport)
```

```
Out[5]: <module 'DataExperimentSupport' from '/home/magni/ML_Root/project_root/utility_files/DataExperimentSupport.py'>
```

Load Data

```
In [6]: #axis_labels=[1,2,3,4,5]
axis_labels=[0,1]
classifier = RandomForestClassifier()
ANALYSIS_COL = 'reviewText_lemma_bert'
UNIQUE_COL = 'uuid'
TARGET_COL = 'overall_posneg'
```

```
In [7]: if LOAD_FROM_EXP:
    #start from saved state
    myExp = jarvis.loadExperiment(FILE_NAME)
    myExp.display()

else:
    #start from source file and regenerate
    testDf = pd.read_pickle(jarvis.DATA_DIR_WORK + "/01_NL_ReviewText_All(new

    testDfBert = cwutils.getBertEncodeFrame(df=testDf,
                                              bertColumn=ANALYSIS_COL,
                                              uniqueColumn=UNIQUE_COL,
                                              otherColumns=[TARGET_COL]
                                              )

    myExp = DataExperiment.DataExperiment(projectName=PROJECT_NAME,
                                          experimentName=EXPERIMENT_NAME,
                                          origData=testDfBert,
                                          uniqueColumn=UNIQUE_COL,
                                          targetColumn=TARGET_COL,
                                          classifier=classifier)
```

```
DataExperiment summary:
--> projectName: ML1010-Group-Project
--> experimentName: ReviewText_Lemma_Bert2 (Random Forest)
--> isDataPackageLoaded: True
--> isBaseModelLoaded: False
--> isBaseModelPredicted: False
--> isBaseModelLearningCurveCreated: False
--> isFinalModelLoaded: False
--> isFinalModelPredicted: False
--> isFinalModelLearningCurveCreated: False
--> isClassifierLoaded: True
RandomForestClassifier()
```

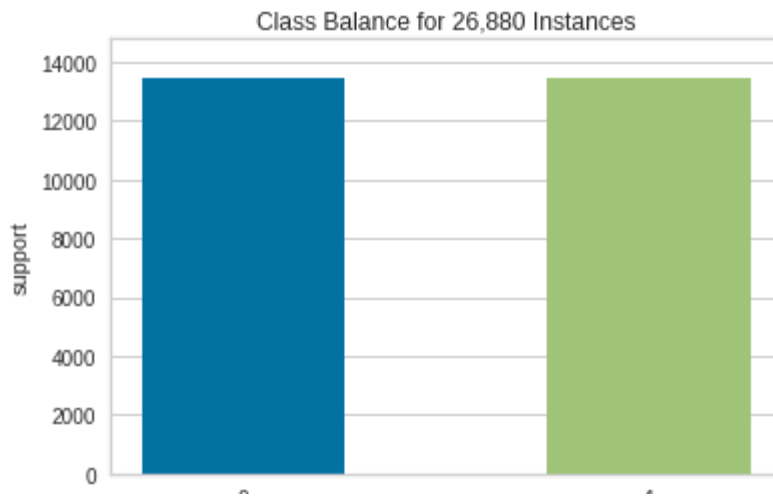
```
DataPackage summary:
Attributes:
--> uniqueColumn: uuid
--> targetColumn: overall_posneg
Process:
--> isBalanced: False
--> isTrainTestSplit: False
Data:
--> isOrigDataLoaded: True
--> isTrainDataLoaded: False
--> isTestDataLoaded: False
```

In [8]:

```
#myExp.processDataPackage()
myExp.dataPackage.classBalanceUndersample()
myExp.dataPackage.splitTrainTest()
```



Undersampling data to match min class: 0 of size: 13440



In [9]: `myExp.createBaseModel()`

Base Model Stats:
 Accuracy: 0.81
 Precision: 0.81
 Recall: 0.81
 F1 Score: 0.81
 Cohen kappa: 0.61

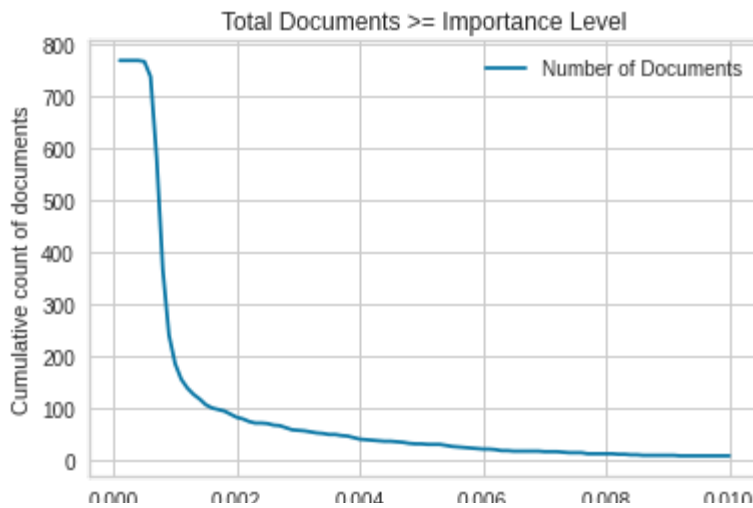
In [10]: `myExp.analyzeBaseModelFeatureImportance(returnAbove=0.002)`

0%| | 0/101 [00:00<?, ?it/s]
 Feature Importance Summary:
 ---> Original feature count: 768
 ---> Returned feature count: 82
 ---> Removed feature count: 686
 ---> Return items above (including): 0.002

Out[10]:

	feature_idx	importance
0	15	0.002212
1	20	0.004006
2	25	0.007208
3	38	0.002529
4	42	0.002185
...
77	698	0.002745
78	700	0.005529
79	707	0.004264
80	713	0.002237
81	726	0.005360

82 rows × 2 columns



```
In [11]: myExp.createFinalModel(featureImportanceThreshold=0.002)
```

```
0%|          | 0/101 [00:00<?, ?it/s]
0%|          | 0/101 [00:00<?, ?it/s]
Final Model Stats:
Accuracy: 0.81
Precision: 0.81
Recall: 0.81
F1 Score: 0.81
Cohen kappa: 0.61
```

```
In [12]: myExp.createBaseModelLearningCurve(n_jobs=10)
```

```
[learning_curve] Training set sizes: [ 1720  3440  8601 17203]
[Parallel(n_jobs=10)]: Using backend LokyBackend with 10 concurrent workers.
[Parallel(n_jobs=10)]: Done   3 out of  20 | elapsed:    2.5s remaining:  1
4.3s
[Parallel(n_jobs=10)]: Done   9 out of  20 | elapsed:    8.8s remaining:  1
0.8s
[Parallel(n_jobs=10)]: Done  15 out of  20 | elapsed:   22.2s remaining:
7.4s
[Parallel(n_jobs=10)]: Done  20 out of  20 | elapsed:   53.4s finished
```

```
In [13]: myExp.createFinalModelLearningCurve(n_jobs=10)
```

```
[Parallel(n_jobs=10)]: Using backend LokyBackend with 10 concurrent workers.
[learning_curve] Training set sizes: [ 1720  3440  8601 17203]
[Parallel(n_jobs=10)]: Done   3 out of  20 | elapsed:    0.9s remaining:
5.0s
[Parallel(n_jobs=10)]: Done   9 out of  20 | elapsed:    3.3s remaining:
4.0s
[Parallel(n_jobs=10)]: Done  15 out of  20 | elapsed:    8.1s remaining:
2.7s
[Parallel(n_jobs=10)]: Done  20 out of  20 | elapsed:   13.4s finished
```

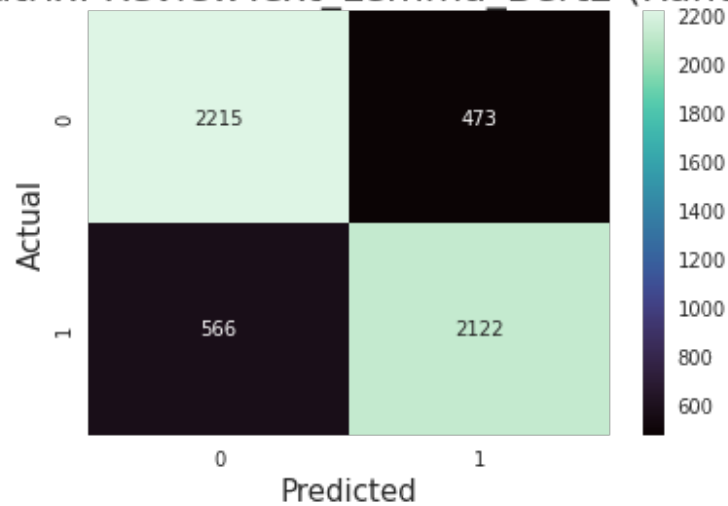
```
In [14]: myExp.showBaseModelReport(axisLabels=axis_labels,
                                     upperValue=0.025)
```

```
Base Model Stats:
```

Accuracy: 0.81
 Precision: 0.81
 Recall: 0.81
 F1 Score: 0.81
 Cohen kappa: 0.61

	precision	recall	f1-score	support
0	0.80	0.82	0.81	2688
1	0.82	0.79	0.80	2688
accuracy			0.81	5376
macro avg	0.81	0.81	0.81	5376
weighted avg	0.81	0.81	0.81	5376

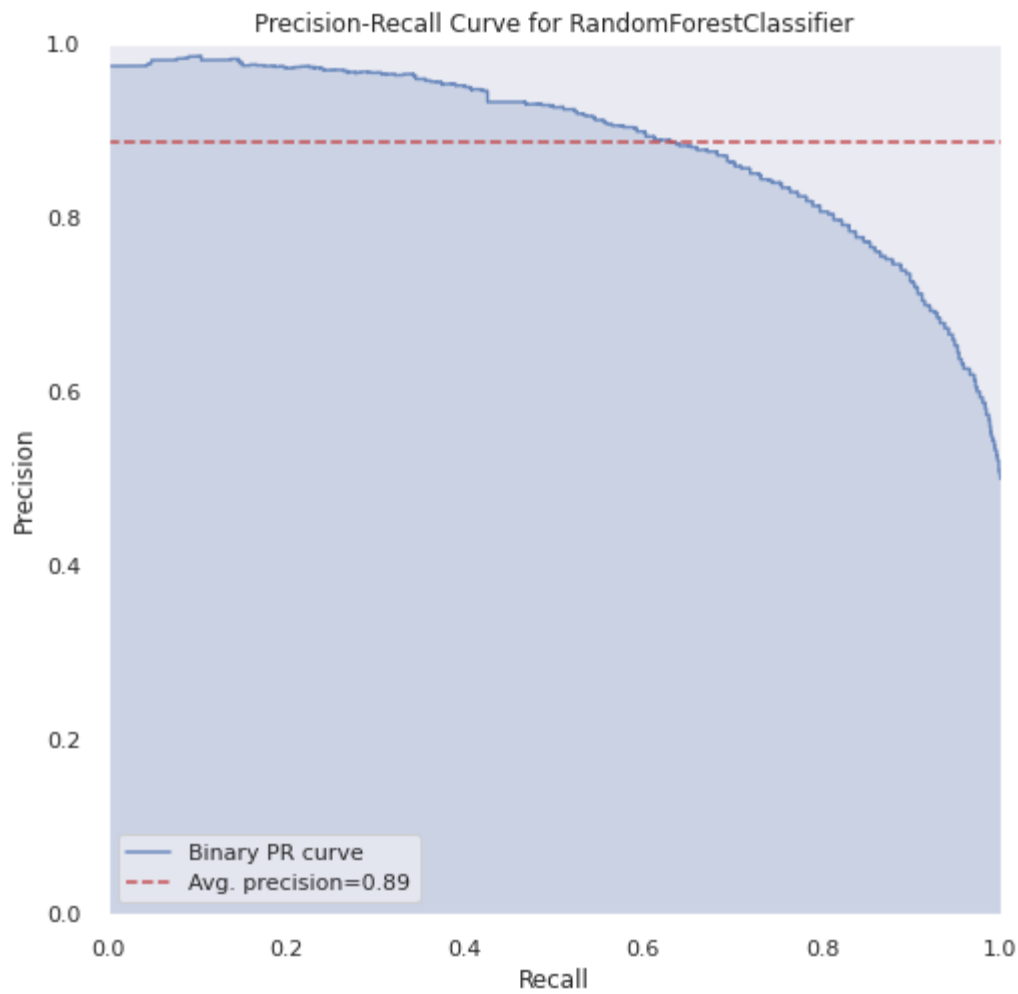
Confusion Matrix: ReviewText_Lemma_Bert2 (Random Forest)



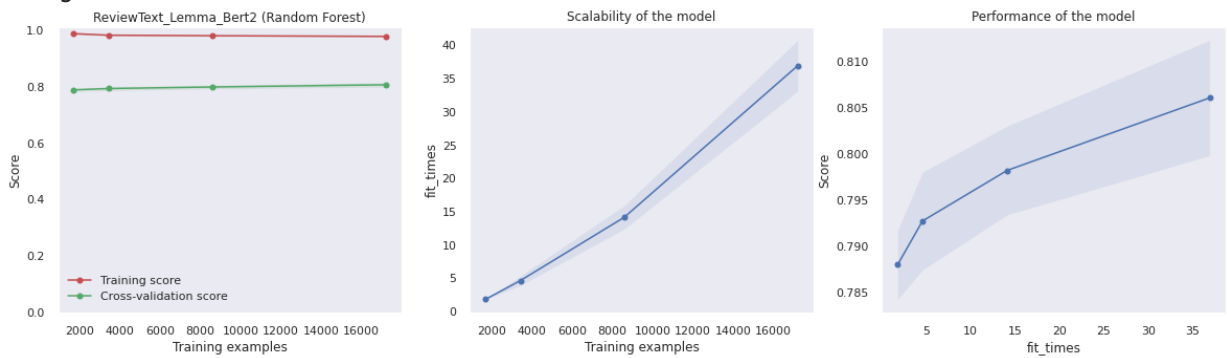
```
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/bas
e.py:444: UserWarning: X has feature names, but RandomForestClassifier was fi
tted without feature names
```

```
f"X has feature names, but {self.__class__.__name__} was fitted without"
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/bas
e.py:444: UserWarning: X has feature names, but RandomForestClassifier was fi
tted without feature names
```

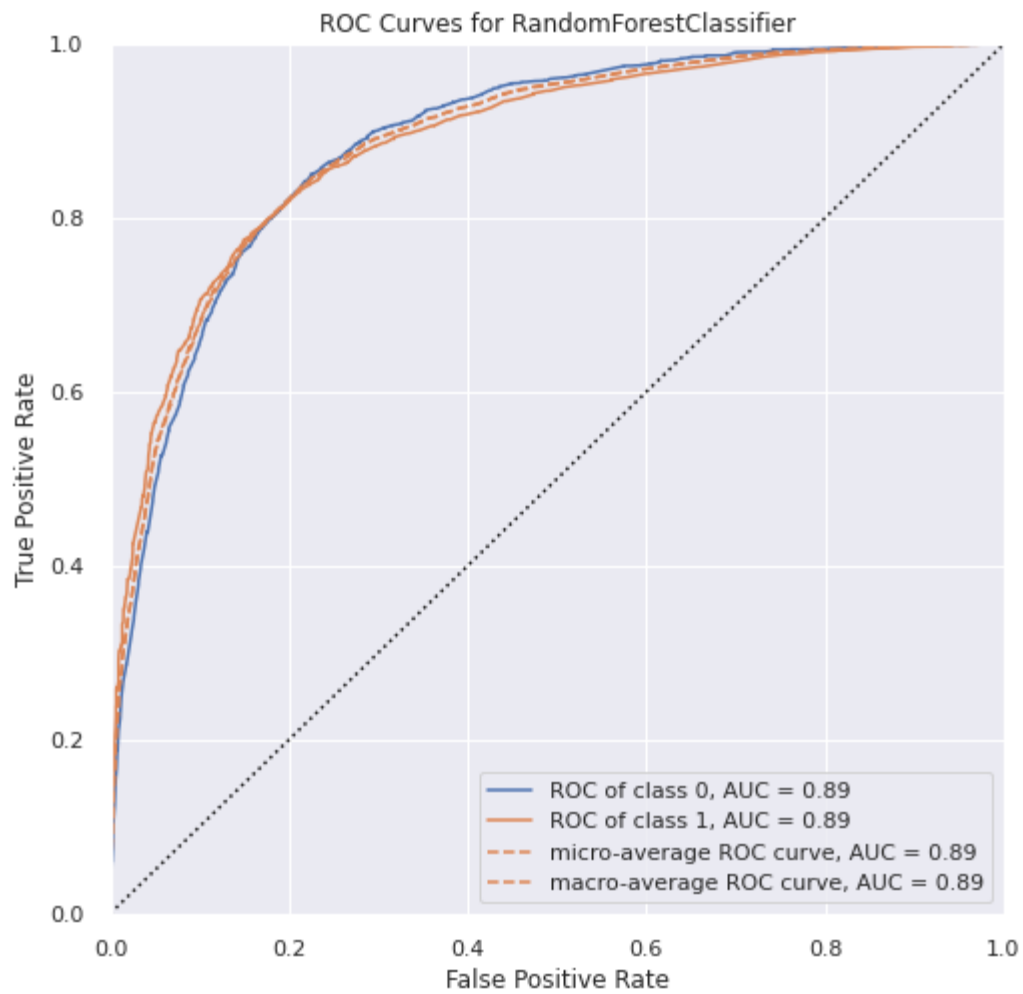
```
f"X has feature names, but {self.__class__.__name__} was fitted without"
```



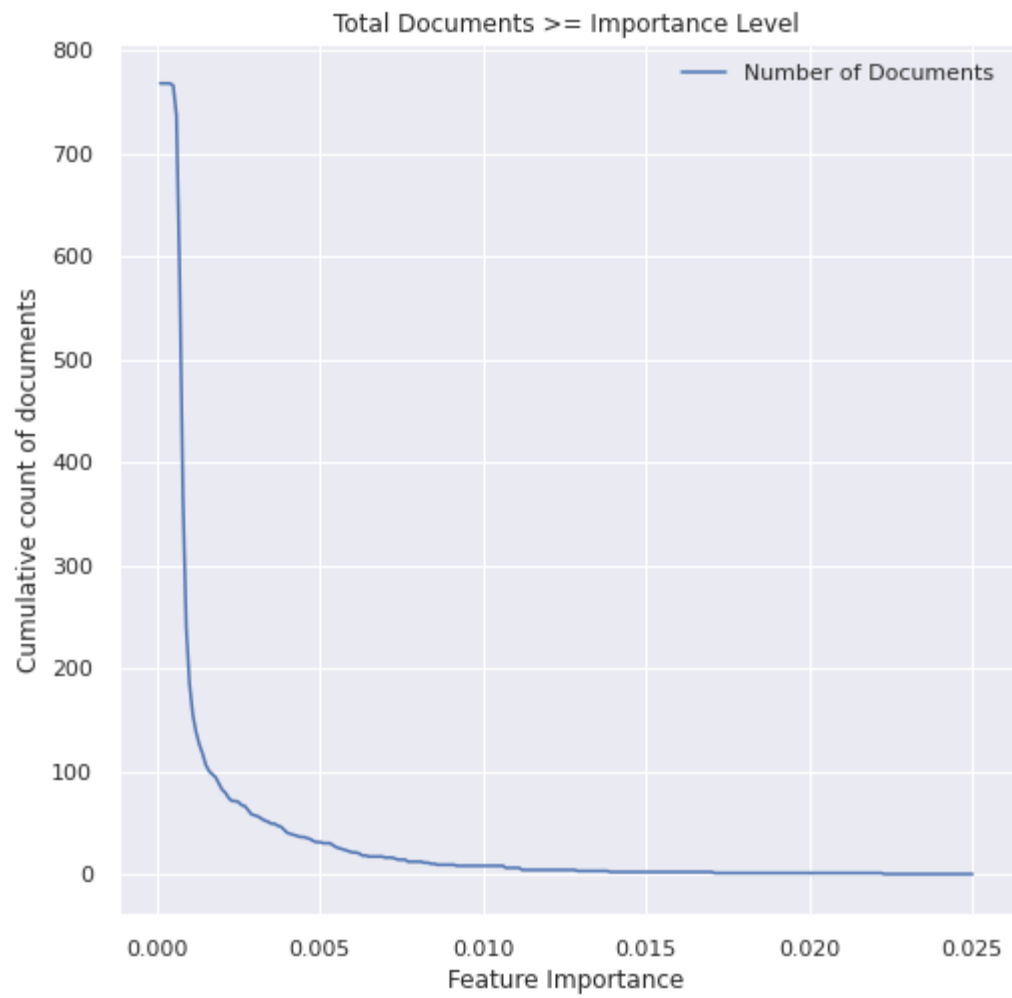
<Figure size 576x576 with 0 Axes>

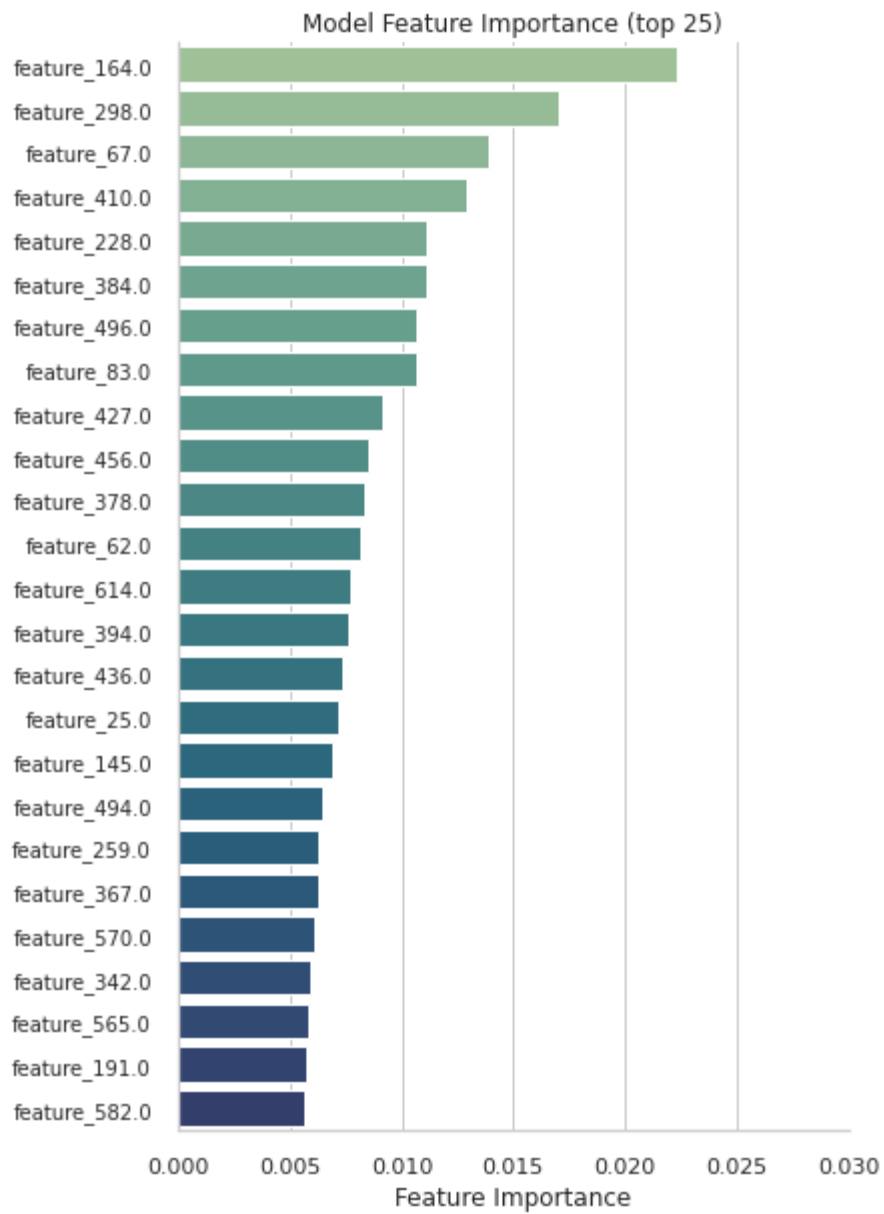


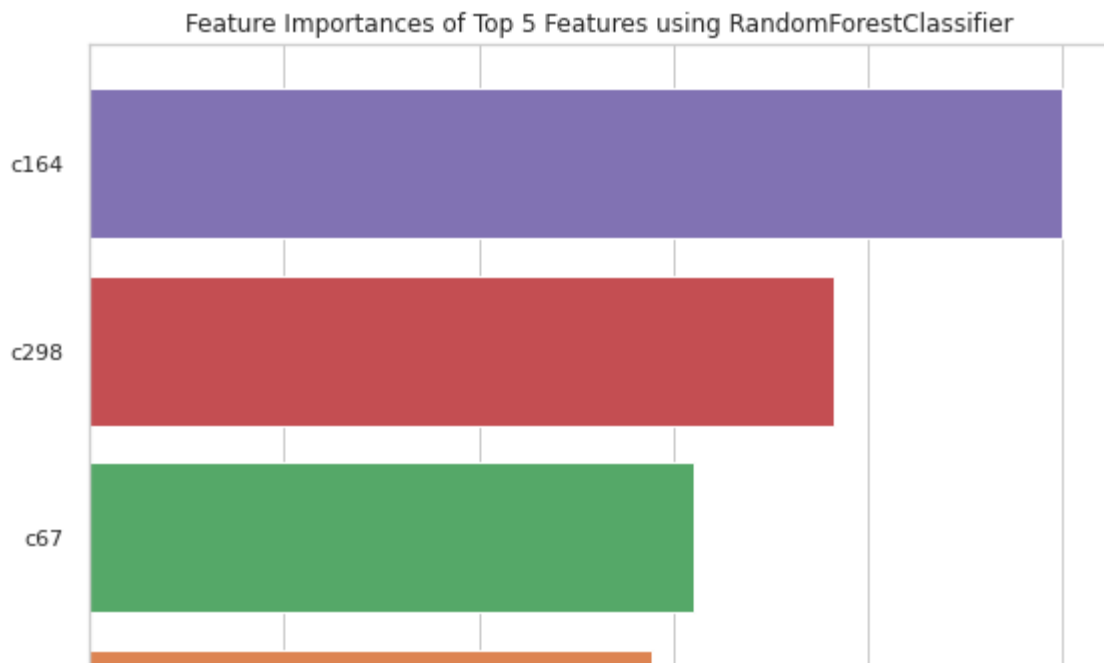
Base model ROCAUC not calculated. Starting now



0% | 0/251 [00:00<?, ?it/s]







```
In [15]: myExp.showFinalModelReport(axisLabels=axis_labels,
                                     startValue=0,
                                     increment=0.001,
                                     upperValue=0.03)
```

Final Model Stats:

Accuracy: 0.81

Precision: 0.81

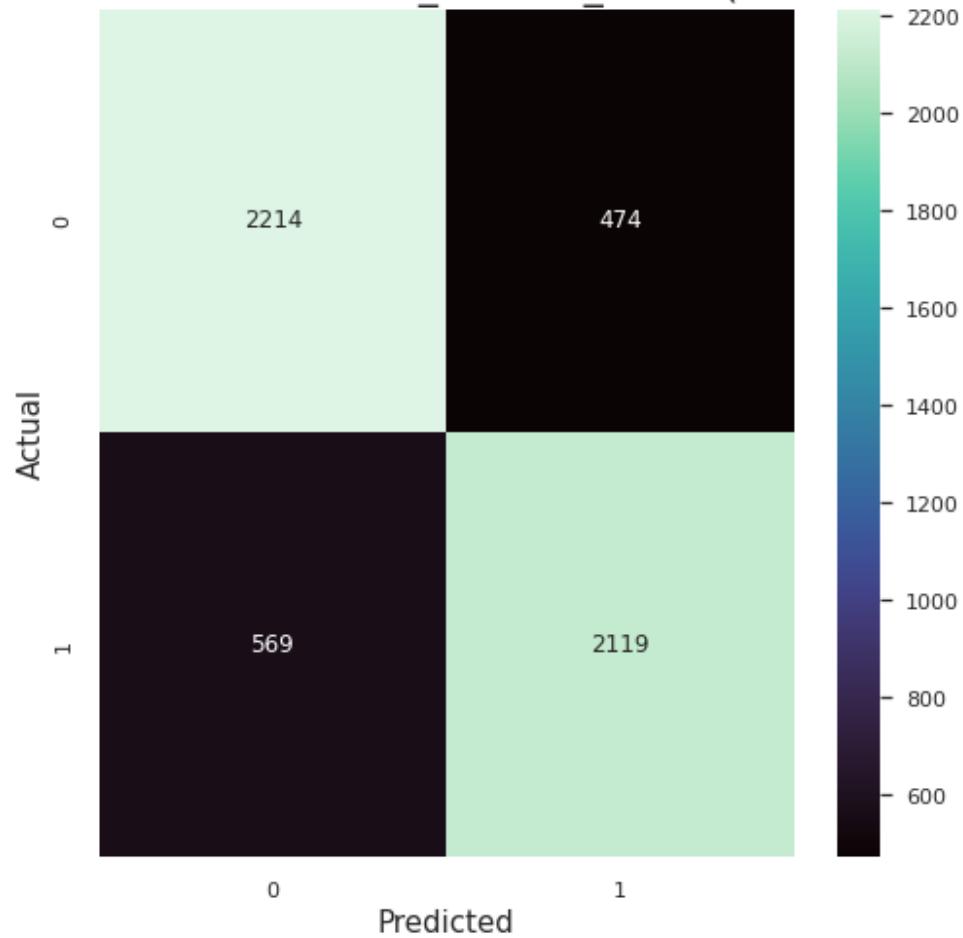
Recall: 0.81

F1 Score: 0.81

Cohen kappa: 0.61

	precision	recall	f1-score	support
0	0.80	0.82	0.81	2688
1	0.82	0.79	0.80	2688
accuracy			0.81	5376
macro avg	0.81	0.81	0.81	5376
weighted avg	0.81	0.81	0.81	5376

Confusion Matrix: ReviewText_Lemma_Bert2 (Random Forest)

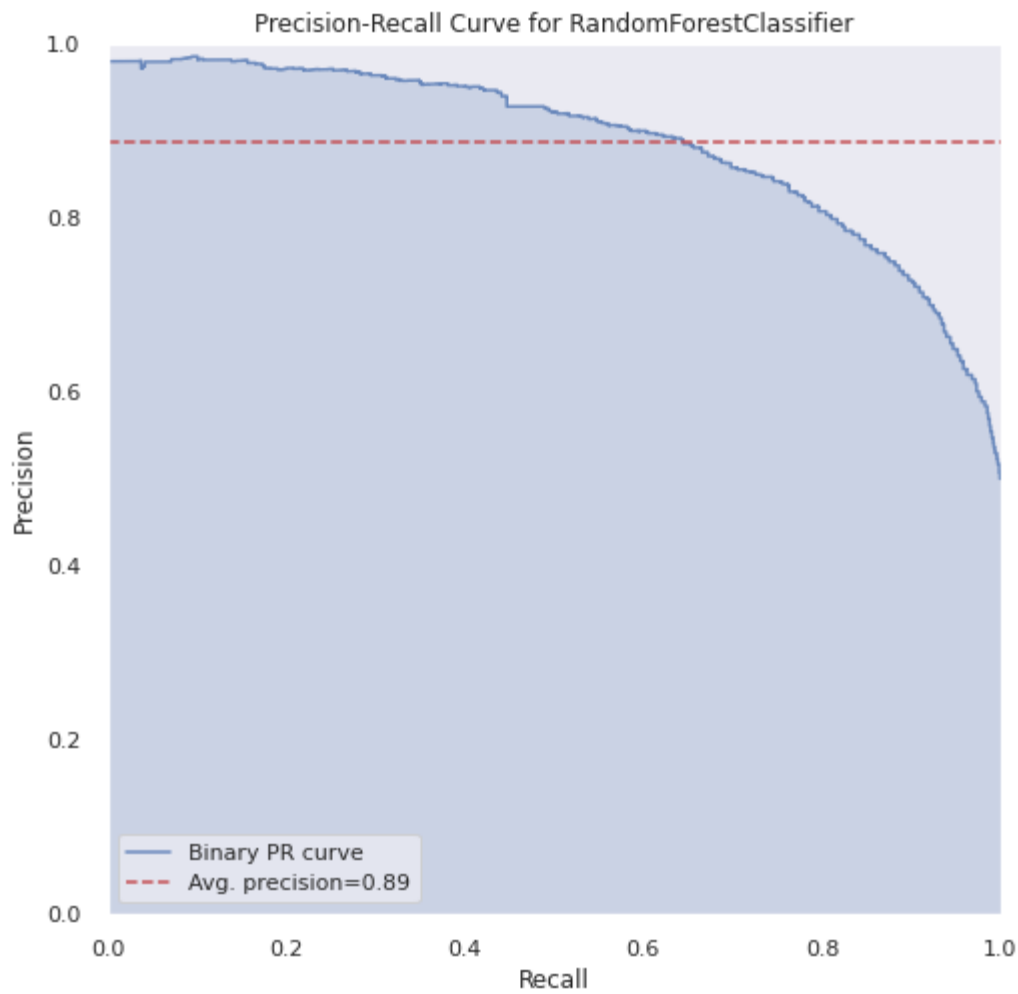


```
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/base.py:444: UserWarning: X has feature names, but RandomForestClassifier was fitted without feature names
```

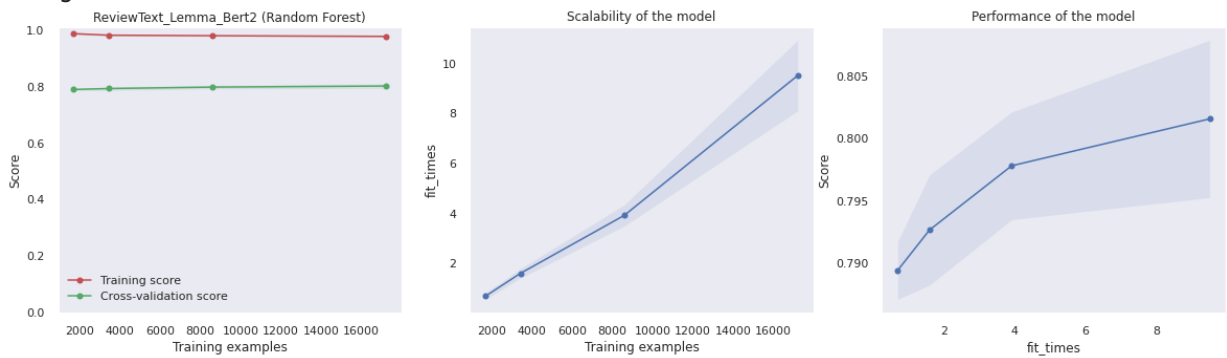
```
  f"X has feature names, but {self.__class__.__name__} was fitted without"
```

```
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/base.py:444: UserWarning: X has feature names, but RandomForestClassifier was fitted without feature names
```

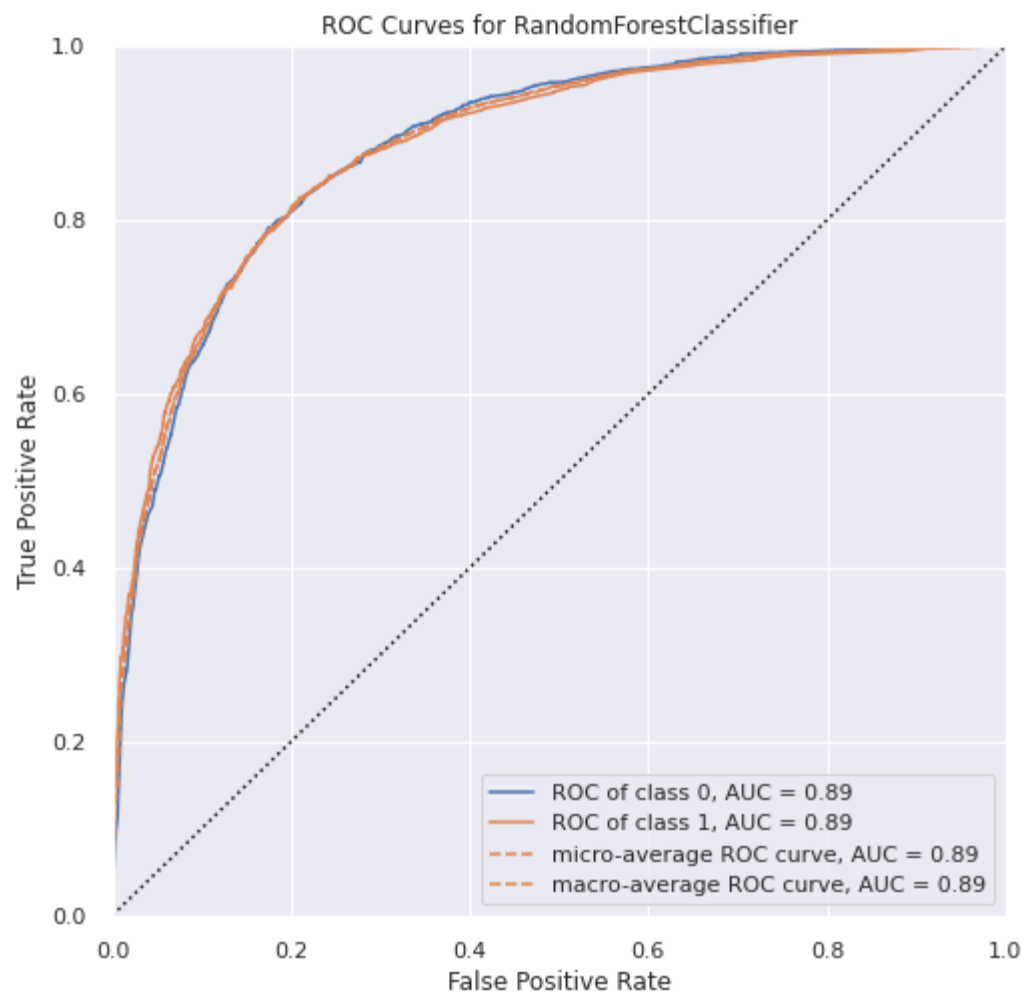
```
  f"X has feature names, but {self.__class__.__name__} was fitted without"
```



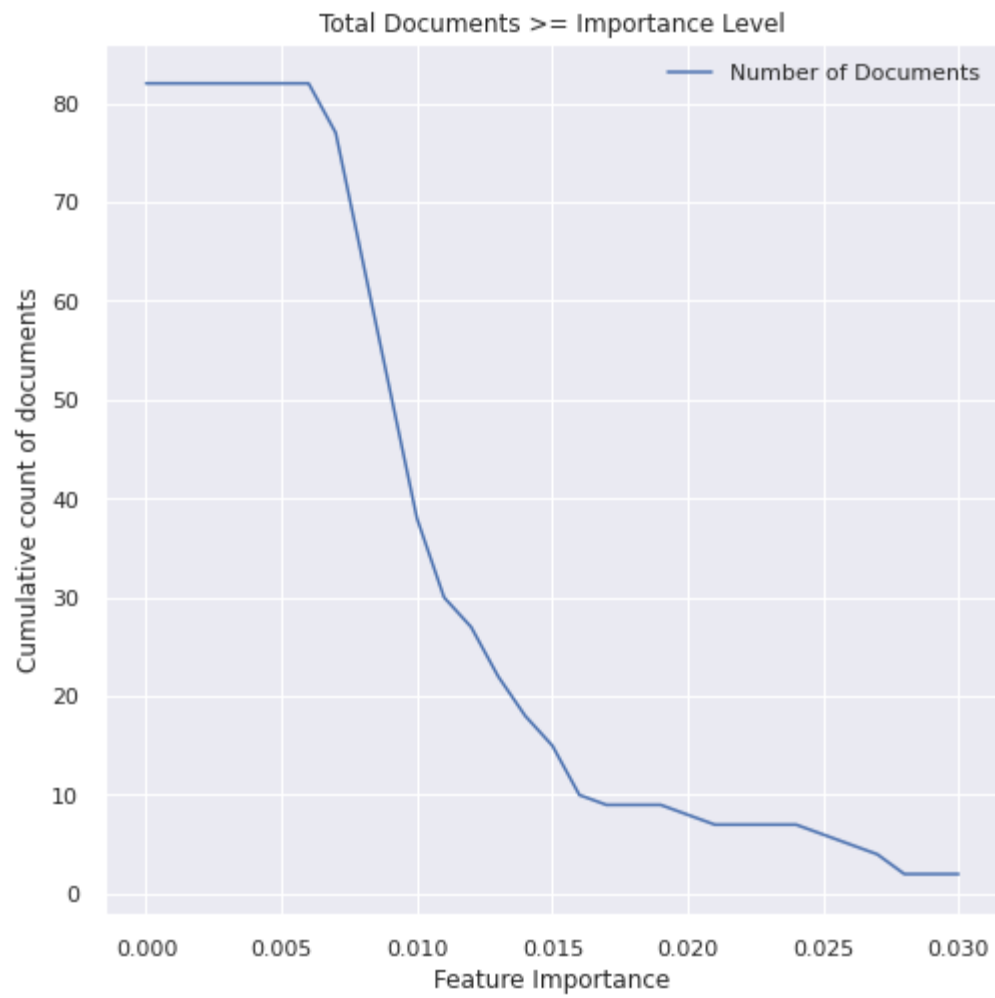
<Figure size 576x576 with 0 Axes>

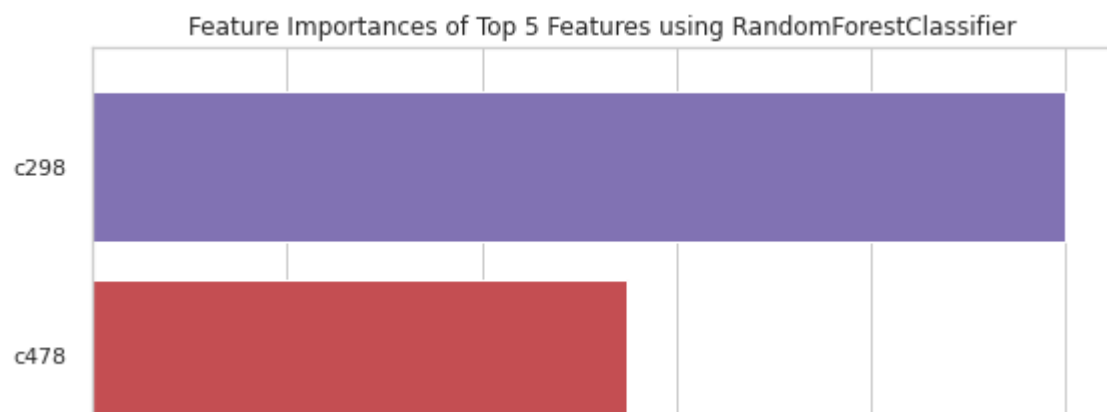
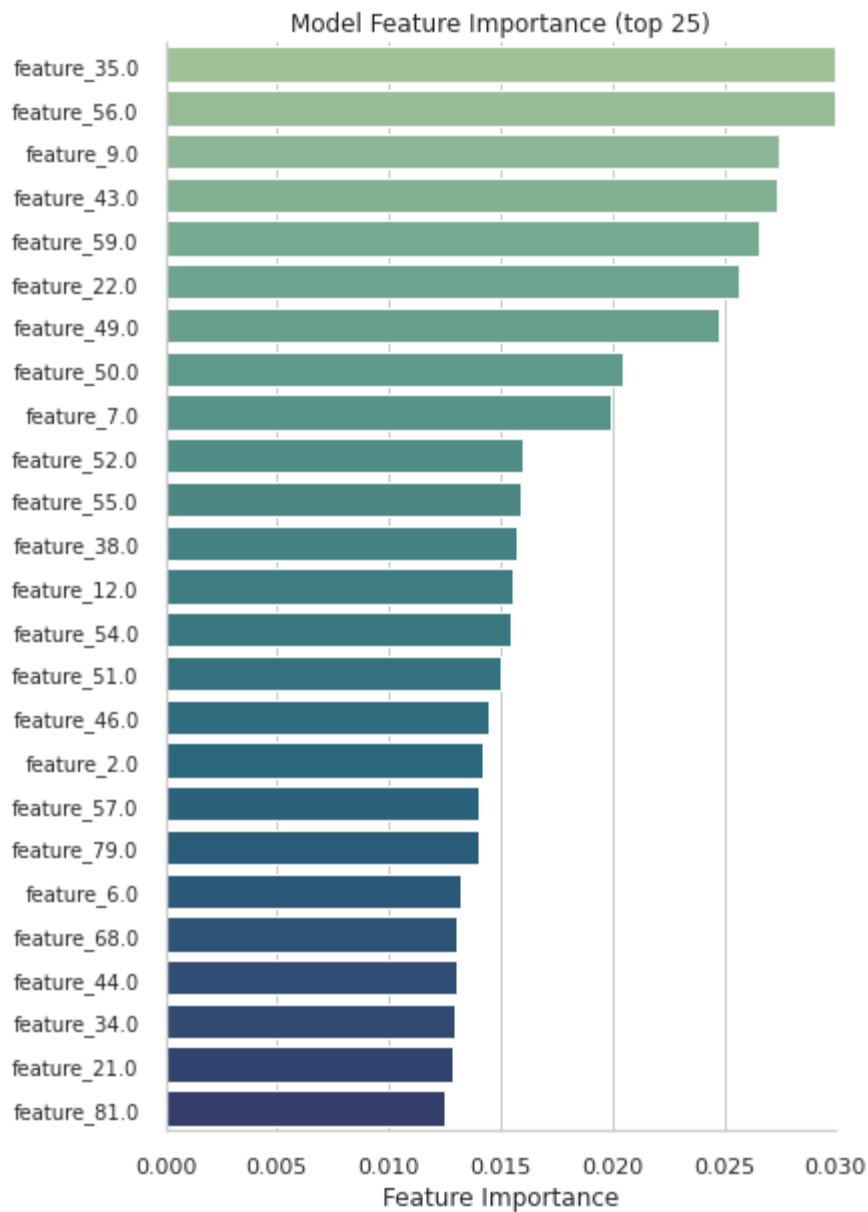


Final model ROCAUC not calculated. Starting now



0% | 0/32 [00:00<?, ?it/s]





```
In [16]: myExp.display()
```

DataExperiment summary:

---> projectName: ML1010-Group-Project

---> experimentName: ReviewText_Lemma_Bert2 (Random Forest)

---> isDataPackageLoaded: True

```

--> isBaseModelLoaded: True
--> isBaseModelPredicted: True
--> isBaseModelLearningCurveCreated: True
--> isFinalModelLoaded: True
--> isFinalModelPredicted: True
--> isFinalModelLearningCurveCreated: True
--> isClassifierLoaded: True
RandomForestClassifier()

```

```

DataPackage summary:
Attributes:
--> uniqueColumn: uuid
--> targetColumn: overall_posneg
Process:
--> isBalanced: True
--> isTrainTestSplit: True
Data:
--> isOrigDataLoaded: False
--> isTrainDataLoaded: True
--> isTestDataLoaded: True

```

Save Experiment

```
In [17]: jarvis.saveExperiment(myExp, FILE_NAME)
```

```

[CV] END ....., score=(train=0.985, test=0.784) total time=
1.8s
[CV] END ....., score=(train=0.988, test=0.793) total time=
1.9s
[CV] END ....., score=(train=0.981, test=0.802) total time=
4.3s
[CV] END ....., score=(train=0.980, test=0.801) total time= 1
3.2s
[CV] END ....., score=(train=0.985, test=0.787) total time=
0.6s
[CV] END ....., score=(train=0.981, test=0.799) total time=
3.5s
[CV] END ....., score=(train=0.983, test=0.793) total time=
4.3s
[CV] END ....., score=(train=0.980, test=0.804) total time= 1
3.4s
[CV] END ....., score=(train=0.980, test=0.793) total time=
4.1s
[CV] END ....., score=(train=0.988, test=0.791) total time=
1.9s
[CV] END ....., score=(train=0.981, test=0.791) total time= 1
3.4s
[CV] END ....., score=(train=0.981, test=0.790) total time=
1.8s
[CV] END ....., score=(train=0.987, test=0.789) total time=
0.9s
[CV] END ....., score=(train=0.982, test=0.790) total time=
1.7s
[CV] END ....., score=(train=0.977, test=0.802) total time= 3
5.2s
[CV] END ....., score=(train=0.981, test=0.788) total time=
1.6s

```

```

[CV] END ..... , score=(train=0.980, test=0.804) total time=
3.8s
[CV] END ..... , score=(train=0.981, test=0.787) total time=
6.2s
[CV] END ..... , score=(train=0.987, test=0.787) total time=
1.9s
[CV] END ..... , score=(train=0.982, test=0.794) total time=
4.4s
[CV] END ..... , score=(train=0.978, test=0.808) total time= 3
9.9s
[CV] END ..... , score=(train=0.983, test=0.796) total time=
1.3s
[CV] END ..... , score=(train=0.981, test=0.800) total time=
1.8s
[CV] END ..... , score=(train=0.980, test=0.800) total time=
4.7s
[CV] END ..... , score=(train=0.980, test=0.794) total time= 1
3.5s
[CV] END ..... , score=(train=0.987, test=0.787) total time=
0.6s
[CV] END ..... , score=(train=0.978, test=0.800) total time=
8.0s
[CV] END ..... , score=(train=0.978, test=0.800) total time= 1
7.8s
[CV] END ..... , score=(train=0.977, test=0.794) total time= 1
0.1s
[CV] END ..... , score=(train=0.987, test=0.784) total time=
1.9s
[CV] END ..... , score=(train=0.978, test=0.805) total time= 3
6.0s
[CV] END ..... , score=(train=0.978, test=0.797) total time= 1
0.6s
[CV] END ..... , score=(train=0.981, test=0.788) total time=
4.4s
[CV] END ..... , score=(train=0.977, test=0.817) total time= 3
1.5s
[CV] END ..... , score=(train=0.978, test=0.793) total time=
3.7s
[CV] END ..... , score=(train=0.978, test=0.805) total time=
7.8s
[CV] END ..... , score=(train=0.978, test=0.798) total time= 4
2.6s
[CV] END ..... , score=(train=0.988, test=0.792) total time=
0.8s
[CV] END ..... , score=(train=0.988, test=0.792) total time=
0.9s
[CV] END ..... , score=(train=0.977, test=0.812) total time= 1
1.4s

```

Scratchpad

In []: