Configuration

In [1]:

```
# Parameters
ENABLE_COLAB = False

PROJECT_NAME = 'ML1010-Group-Project'
EXPERIMENT_NAME = 'ReviewText_Lemma_Bert2_Full (Logistic Regression)'
FILE_NAME = '01_ML1010_GP_LR_Bert2_Full'
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 10:34
Hello sir. Extra caffeine may help.
```

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
```

```
2022-01-25 10:34:57.539620: W tensorflow/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: l
ibcudart.so.11.0: cannot open shared object file: No such file or directory
2022-01-25 10:34:57.539644: I tensorflow/stream_executor/cuda/cudart_stub.cc:
29] Ignore above cudart dlerror if you do not have a GPU set up on your machi
ne.
```

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/utilit
y_files/DataExperimentSupport.py'>
```

Load Data

In [6]:

```
from sklearn.linear_model import LogisticRegression

#axis_labels=[1,2,3,4,5]
axis_labels=[0,1]
classifier = LogisticRegression(max_iter=200, verbose=0)
ANALSYSIS_COL = 'reviewText_lemma_bert'
UNIQUE_COL = 'uuid'
TARGET_COL = 'overall_posneg'
```

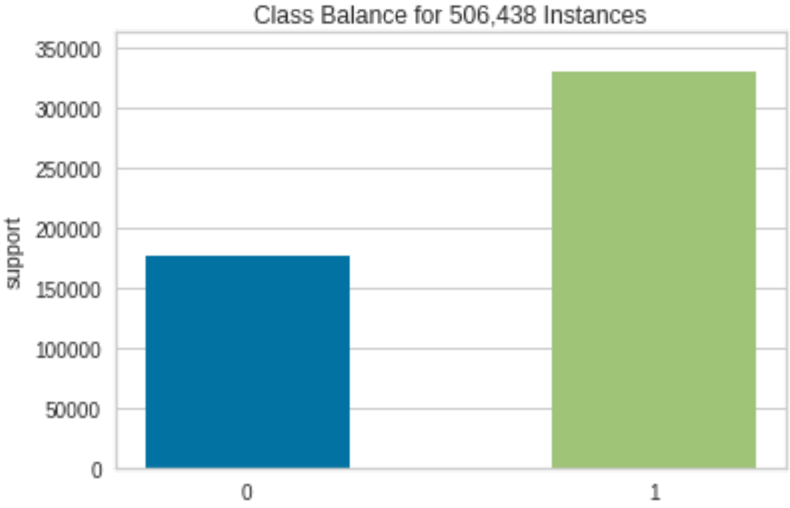
In [7]:

```
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_Cellphone_full.pkl.gz

    testDfBert = cwutils.getBertEncodeFrame(df=testDf,
                                            bertColumn=ANALSYSIS_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)
```

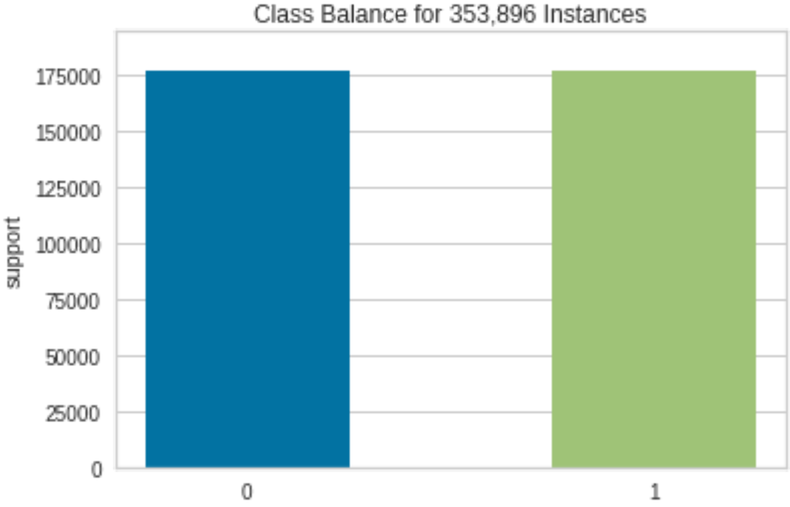
```
DataExperiment summary:
---> projectName: ML1010-Group-Project
---> experimentName: ReviewText_Lemma_Bert2_Full (Logistic Regression)
---> isDataPackageLoaded: True
---> isBaseModelLoaded: False
---> isBaseModelPredicted: False
---> isBaseModelLearningCurveCreated: False
---> isFinalModelLoaded: False
---> isFinalModelPredicted: False
---> isFinalModelLearningCurveCreated: False
---> isClassifierLoaded: True
LogisticRegression(max_iter=200)

    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: 176948
```



	overall_posneg	ttlCol
0	0	176948
1	1	176948

```
Completed train/test split (train_size = 0.8):
---> Original data size: 353896
---> Training data size: 283116
---> Testing data size: 70780
---> Stratified on column: overall_posneg
```

In [9]:

```
%%time
myExp.createBaseModel()
```

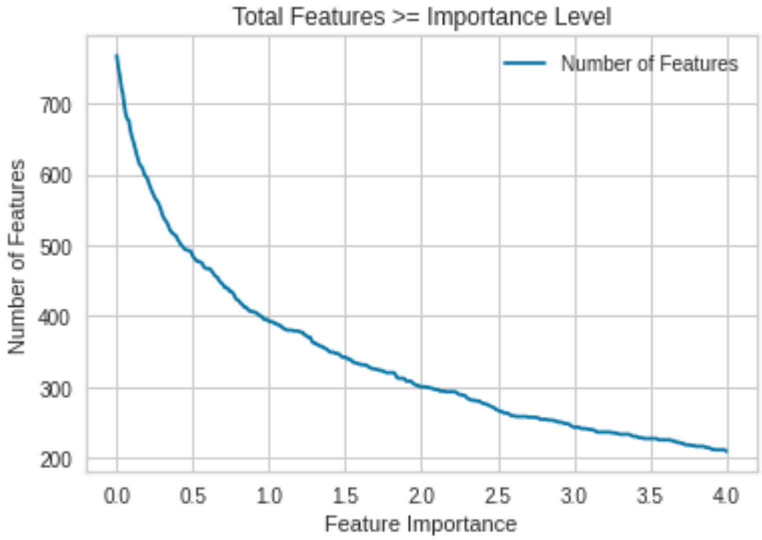
```
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/line
ar_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (stat
us=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
```

```
sion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG
Base Model Stats:
Accuracy: 0.84
Precision: 0.84
Recalll: 0.84
F1 Score: 0.84
Cohen kappa:: 0.68
CPU times: user 7min 6s, sys: 1min 46s, total: 8min 52s
Wall time: 38 s
```

In [11]:

```
_ = myExp.analyzeBaseModelFeatureImportance(startValue=0,
                                            increment=0.01,
                                            upperValue=4,
                                            returnAbove=1.5,
                                           showSummary=True)
#myExp.showBaseLimeGlobalImportance()
```

```
  0%|          | 0/402 [00:00<?, ?it/s]
Feature Importance Summary:
---> Original feature count: 768
---> Returned feature count: 342
---> Removed feature count: 426
---> Return items above (including): 1.5
```



In [12]:

```
%%time
myExp.createFinalModel(featureImportanceThreshold=1.5)
```

```
  0%|          | 0/101 [00:00<?, ?it/s]
  0%|          | 0/101 [00:00<?, ?it/s]
Final Model Stats:
Accuracy: 0.83
Precision: 0.83
Recalll: 0.83
F1 Score: 0.83
Cohen kappa:: 0.66
CPU times: user 3min 4s, sys: 45.9 s, total: 3min 49s
Wall time: 15.7 s
```

In [10]:

```
%%time
myExp.createBaseModelLearningCurve(n_jobs=1)
```

```
[learning_curve] Training set sizes: [ 22649  45298 113246 226492]
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent worke
rs.
[CV] END ..................., score=(train=0.834, test=0.827) total time=
1.5s
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    1.6s remaining:    0.
0s
[CV] END ..................., score=(train=0.834, test=0.832) total time=
3.3s
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    4.9s remaining:    0.
0s
[CV] END ..................., score=(train=0.838, test=0.835) total time=  1
1.5s
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:   16.7s remaining:    0.
0s
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/line
ar_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (stat
us=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
[CV] END ..................., score=(train=0.840, test=0.837) total time=  3
2.3s
[CV] END ..................., score=(train=0.836, test=0.830) total time=
1.4s
[CV] END ..................., score=(train=0.835, test=0.834) total time=
2.2s
[CV] END ..................., score=(train=0.838, test=0.839) total time=  1
2.7s
[CV] END ..................., score=(train=0.839, test=0.840) total time=  2
9.8s
[CV] END ..................., score=(train=0.836, test=0.826) total time=
1.5s
[CV] END ..................., score=(train=0.835, test=0.831) total time=
4.0s
[CV] END ..................., score=(train=0.838, test=0.836) total time=  1
3.2s
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/line
ar_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (stat
us=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
[CV] END .................., score=(train=0.840, test=0.839) total time=  2
8.9s
[CV] END .................., score=(train=0.835, test=0.828) total time=
1.8s
[CV] END .................., score=(train=0.836, test=0.831) total time=
3.0s
[CV] END .................., score=(train=0.838, test=0.836) total time=  1
3.3s
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/line
ar_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (stat
us=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
[CV] END .................., score=(train=0.839, test=0.838) total time=  2
8.7s
[CV] END .................., score=(train=0.839, test=0.825) total time=
1.5s
[CV] END .................., score=(train=0.836, test=0.830) total time=
2.9s
[CV] END .................., score=(train=0.839, test=0.835) total time=  1
2.1s
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/line
ar_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (stat
us=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
[CV] END .................., score=(train=0.840, test=0.837) total time=  3
0.0s
CPU times: user 46min 14s, sys: 11min 5s, total: 57min 19s
Wall time: 3min 59s
[Parallel(n_jobs=1)]: Done  20 out of  20 | elapsed:  4.0min finished
```

In [13]:

```
%%time
myExp.createFinalModelLearningCurve()
```

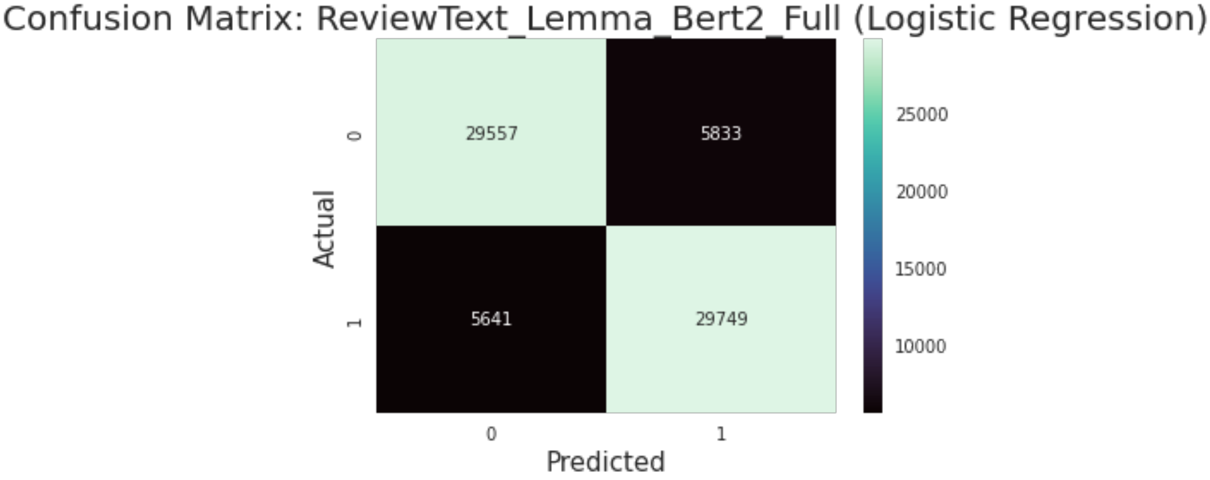
```
[learning_curve] Training set sizes: [ 22649  45298 113246 226492]
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent worke
rs.
[CV] END .................., score=(train=0.826, test=0.820) total time=
0.8s
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.8s remaining:    0.
0s
[CV] END .................., score=(train=0.825, test=0.824) total time=
1.9s
```

```
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    2.7s remaining:    0.
0s
[CV] END ..................., score=(train=0.830, test=0.829) total time=
4.6s
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:    7.5s remaining:    0.
0s
[CV] END ..................., score=(train=0.832, test=0.831) total time=
9.7s
[CV] END ..................., score=(train=0.827, test=0.824) total time=
0.8s
[CV] END ..................., score=(train=0.827, test=0.828) total time=
1.4s
[CV] END ..................., score=(train=0.830, test=0.832) total time=
4.3s
[CV] END ..................., score=(train=0.832, test=0.834) total time=  1
1.0s
[CV] END ..................., score=(train=0.827, test=0.819) total time=
0.7s
[CV] END ..................., score=(train=0.826, test=0.824) total time=
1.9s
[CV] END ..................., score=(train=0.830, test=0.829) total time=
5.3s
[CV] END ..................., score=(train=0.832, test=0.831) total time=  1
3.2s
[CV] END ..................., score=(train=0.827, test=0.819) total time=
0.9s
[CV] END ..................., score=(train=0.826, test=0.824) total time=
2.0s
[CV] END ..................., score=(train=0.830, test=0.830) total time=
4.9s
[CV] END ..................., score=(train=0.832, test=0.832) total time=  1
1.8s
[CV] END ..................., score=(train=0.828, test=0.819) total time=
0.7s
[CV] END ..................., score=(train=0.827, test=0.823) total time=
1.4s
[CV] END ..................., score=(train=0.832, test=0.827) total time=
4.5s
[CV] END ..................., score=(train=0.833, test=0.829) total time=  1
1.2s
CPU times: user 18min 14s, sys: 5min 21s, total: 23min 36s
Wall time: 1min 34s
```

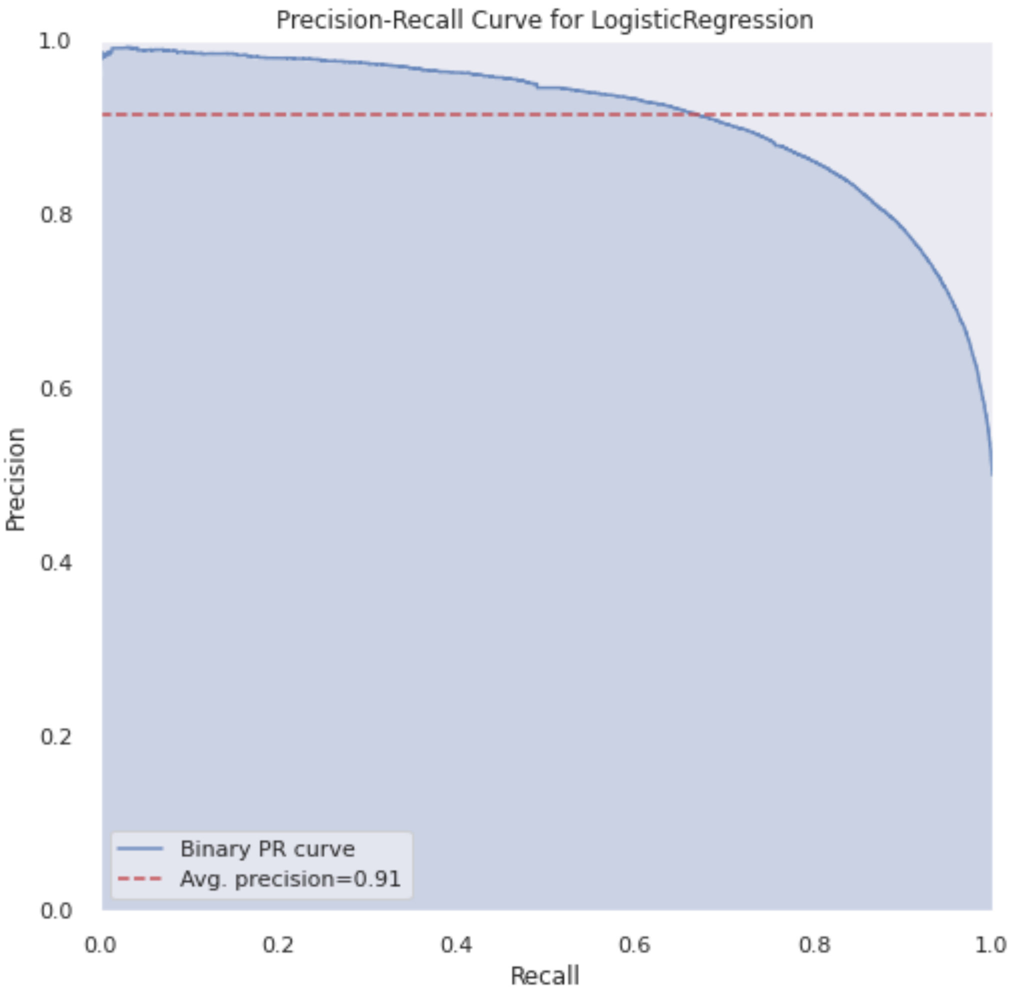
In [14]:

```
myExp.showBaseModelReport(axisLabels=axis_labels,
                          startValue=0,
                          increment=0.01,
                          upperValue=4)
```

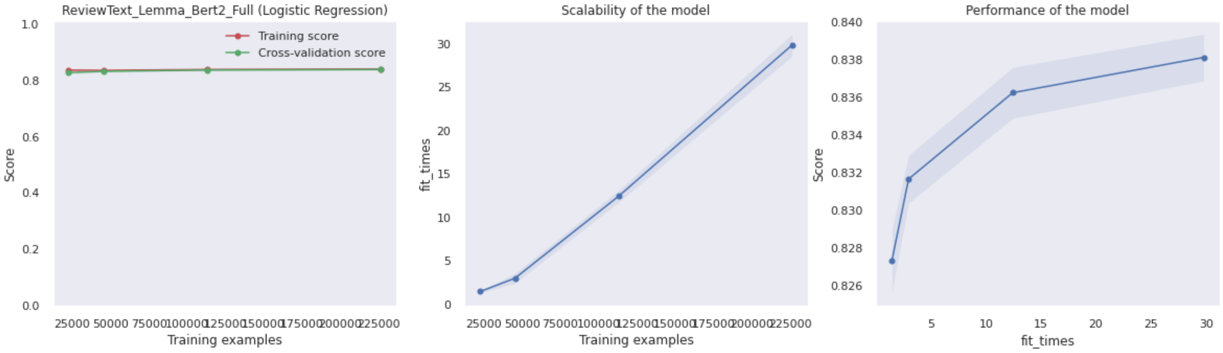
```
Base Model Stats:
Accuracy: 0.84
Precision: 0.84
Recalll: 0.84
F1 Score: 0.84
Cohen kappa:: 0.68
              precision    recall  f1-score   support

           0       0.84      0.84      0.84     35390
           1       0.84      0.84      0.84     35390

    accuracy                           0.84     70780
   macro avg       0.84      0.84      0.84     70780
weighted avg       0.84      0.84      0.84     70780
```



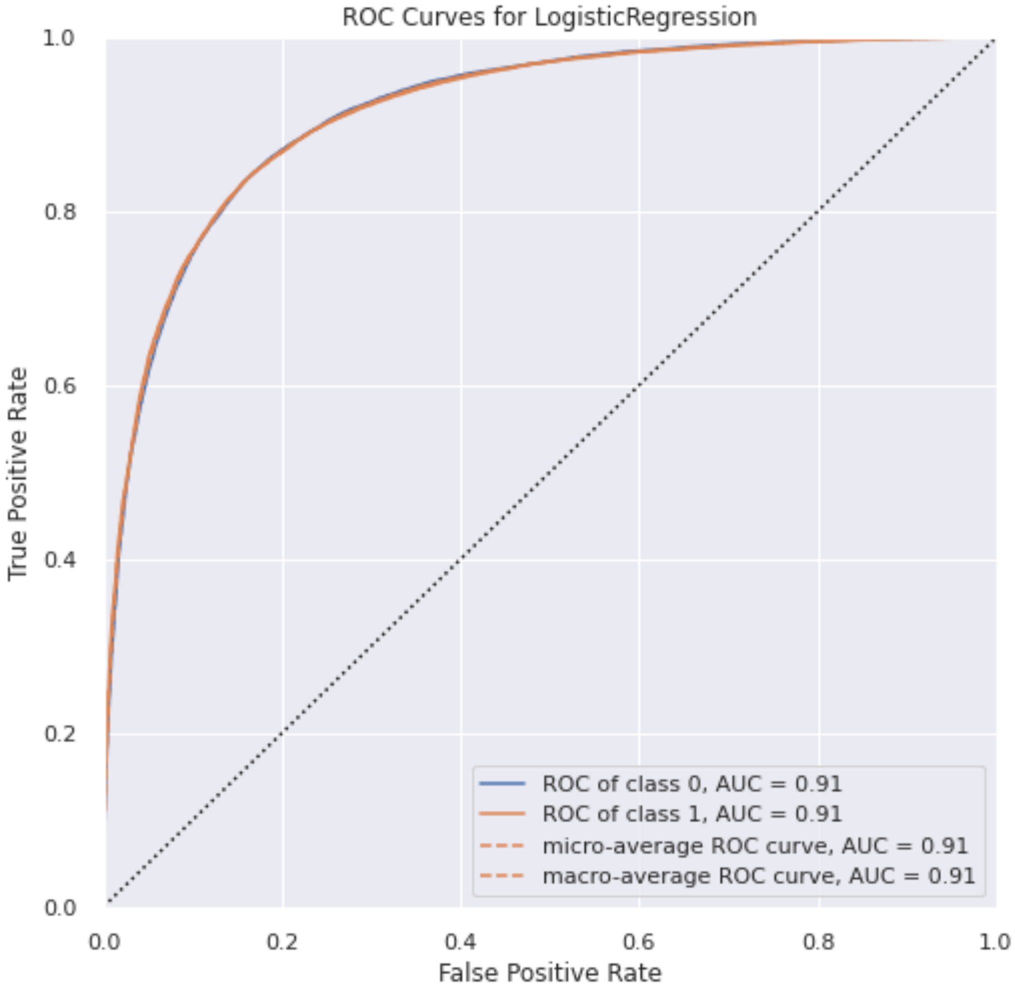
```
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/bas
e.py:444: UserWarning: X has feature names, but LogisticRegression was fitted
without feature names
  f"X has feature names, but {self.__class__.__name__} was fitted without"
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/bas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



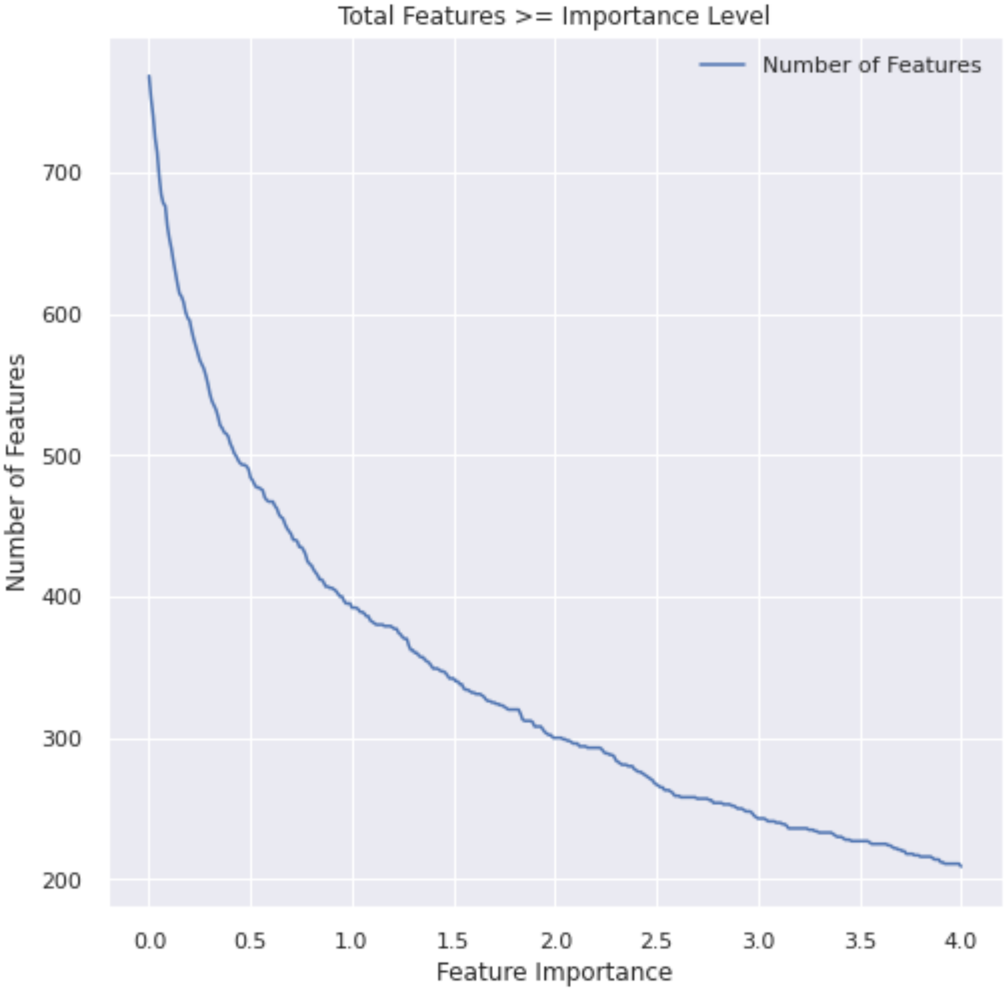
```
<Figure size 576x576 with 0 Axes>
```

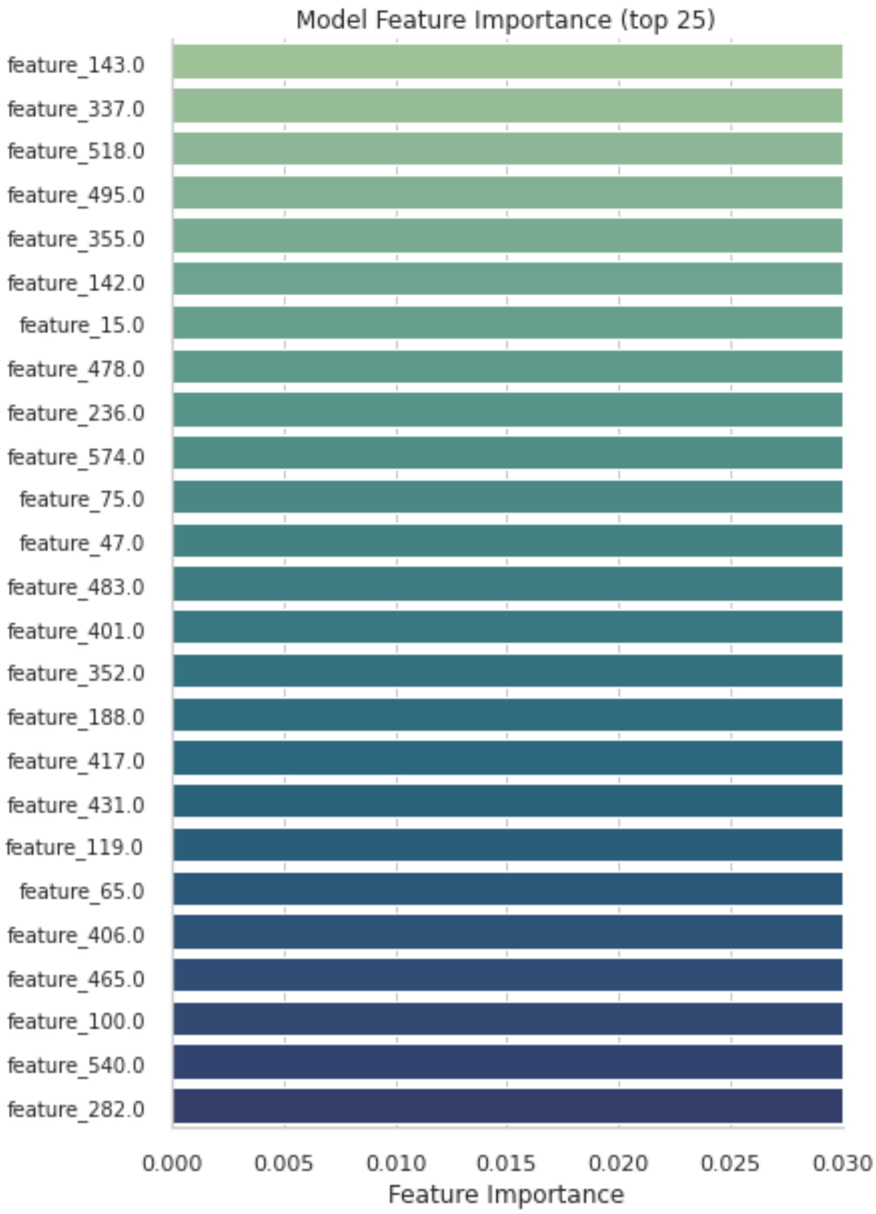


```
Base model ROCAUC not calculated. Starting now
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/line
ar_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (stat
us=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
```

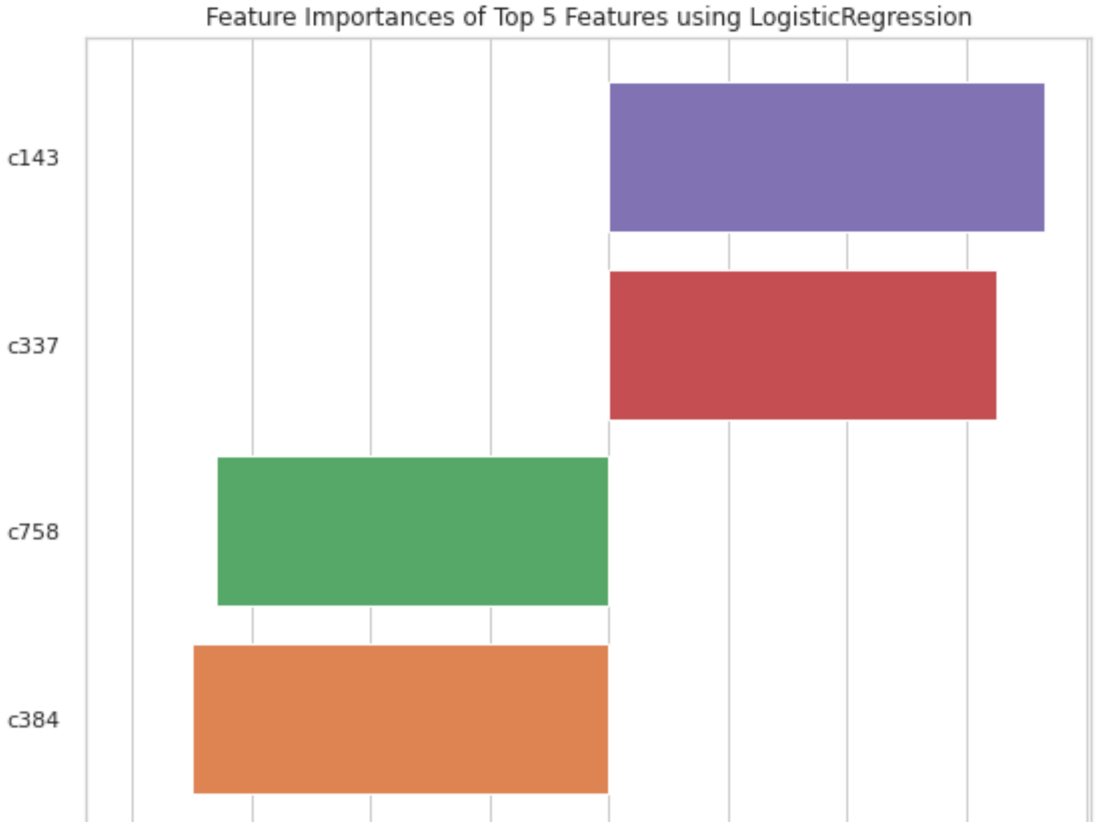


```
0%|          | 0/402 [00:00<?, ?it/s]
```


Total Features >= Importance Level
Number of Features
Number of Features
700
600
500
400
300
200
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Feature Importance



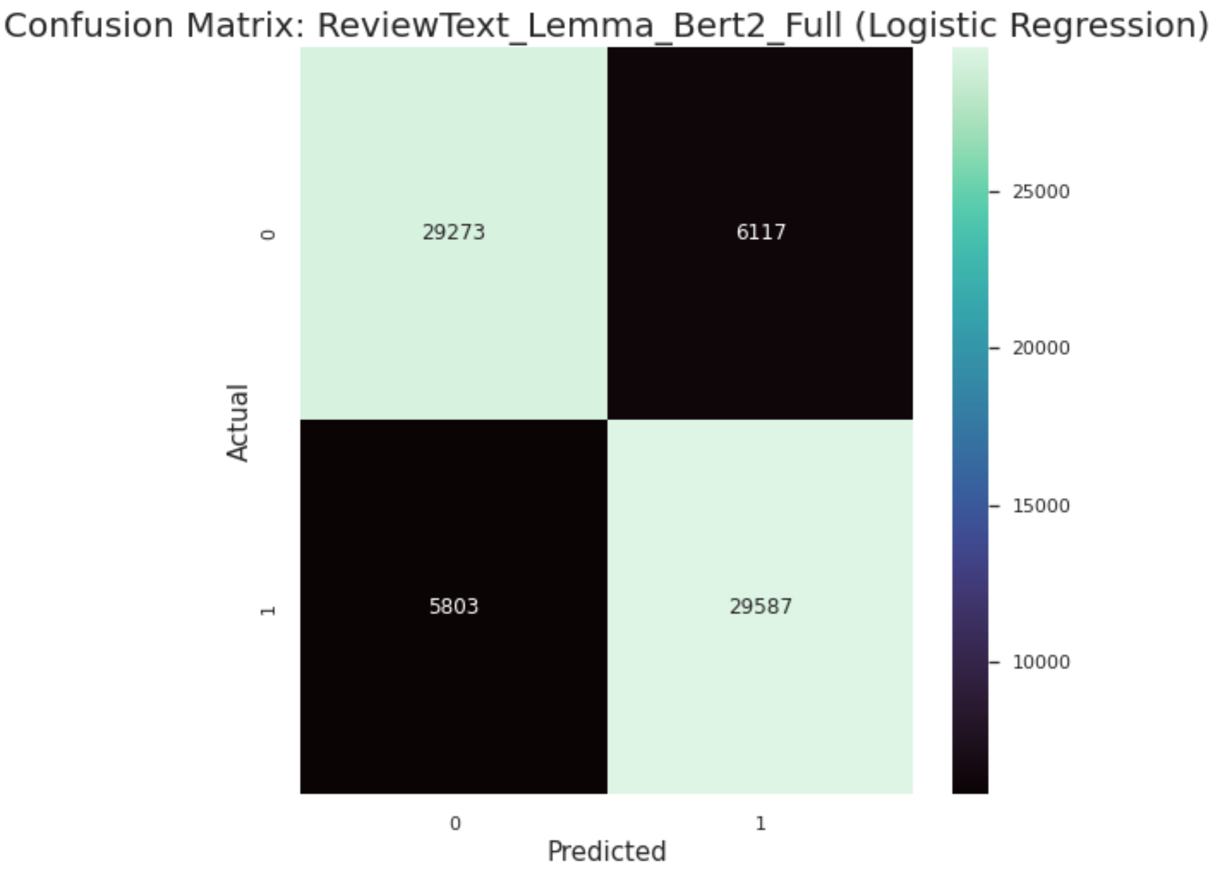
```
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/yellowbrick/
model_selection/importances.py:199: YellowbrickWarning: detected multi-dimens
ional feature importances but stack=False, using mean to aggregate them.
  YellowbrickWarning,
```



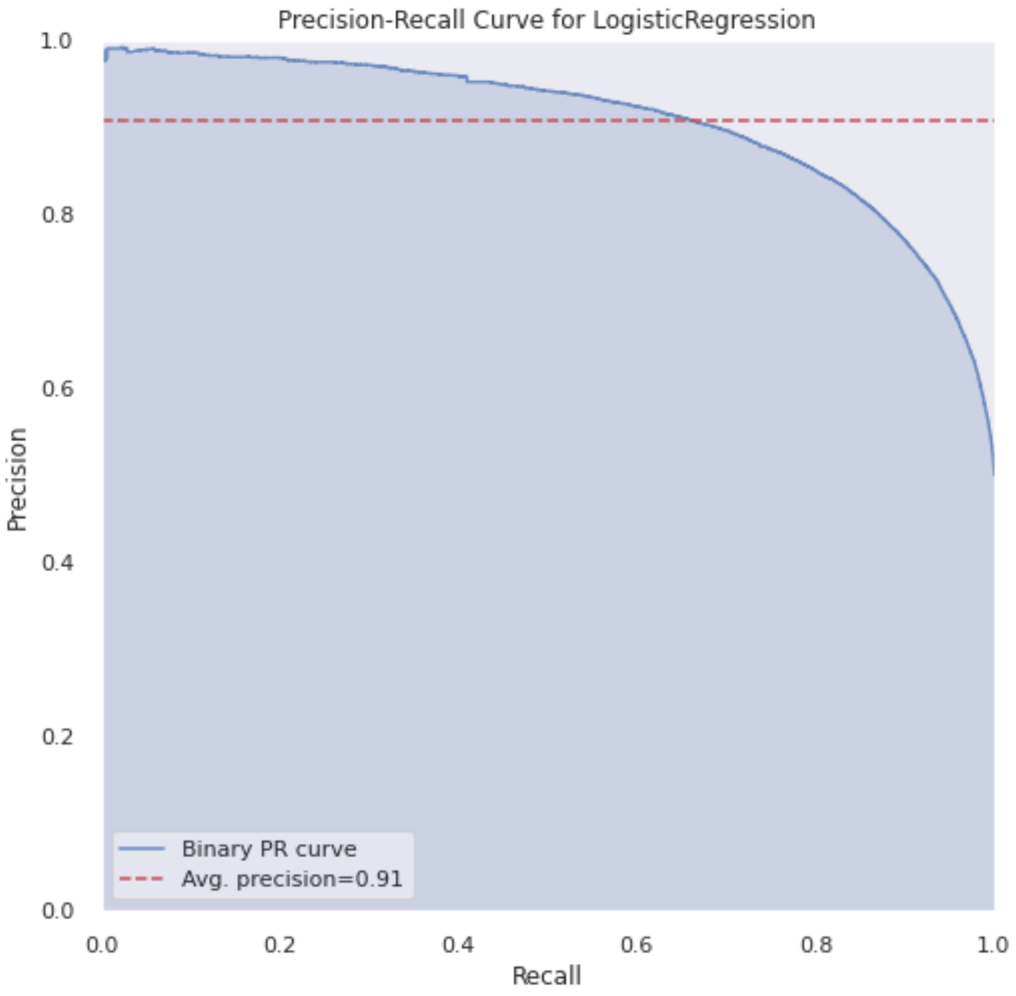
In [15]:

```
myExp.showFinalModelReport(axisLabels=axis_labels,
                           startValue=0,
                                        increment=0.01,
                                        upperValue=4)
```

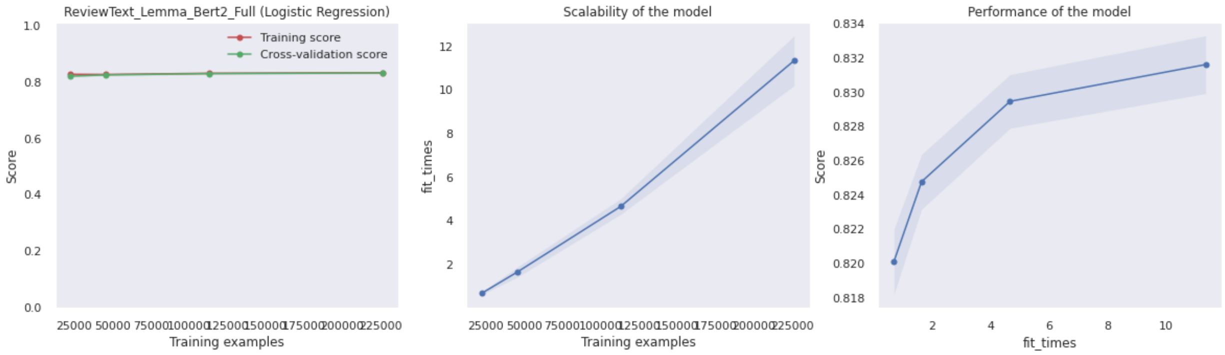
```
Final Model Stats:
Accuracy: 0.83
Precision: 0.83
Recalll: 0.83
F1 Score: 0.83
Cohen kappa:: 0.66
              precision    recall  f1-score   support

           0       0.83      0.83      0.83     35390
           1       0.83      0.84      0.83     35390

    accuracy                           0.83     70780
   macro avg       0.83      0.83      0.83     70780
weighted avg       0.83      0.83      0.83     70780
```



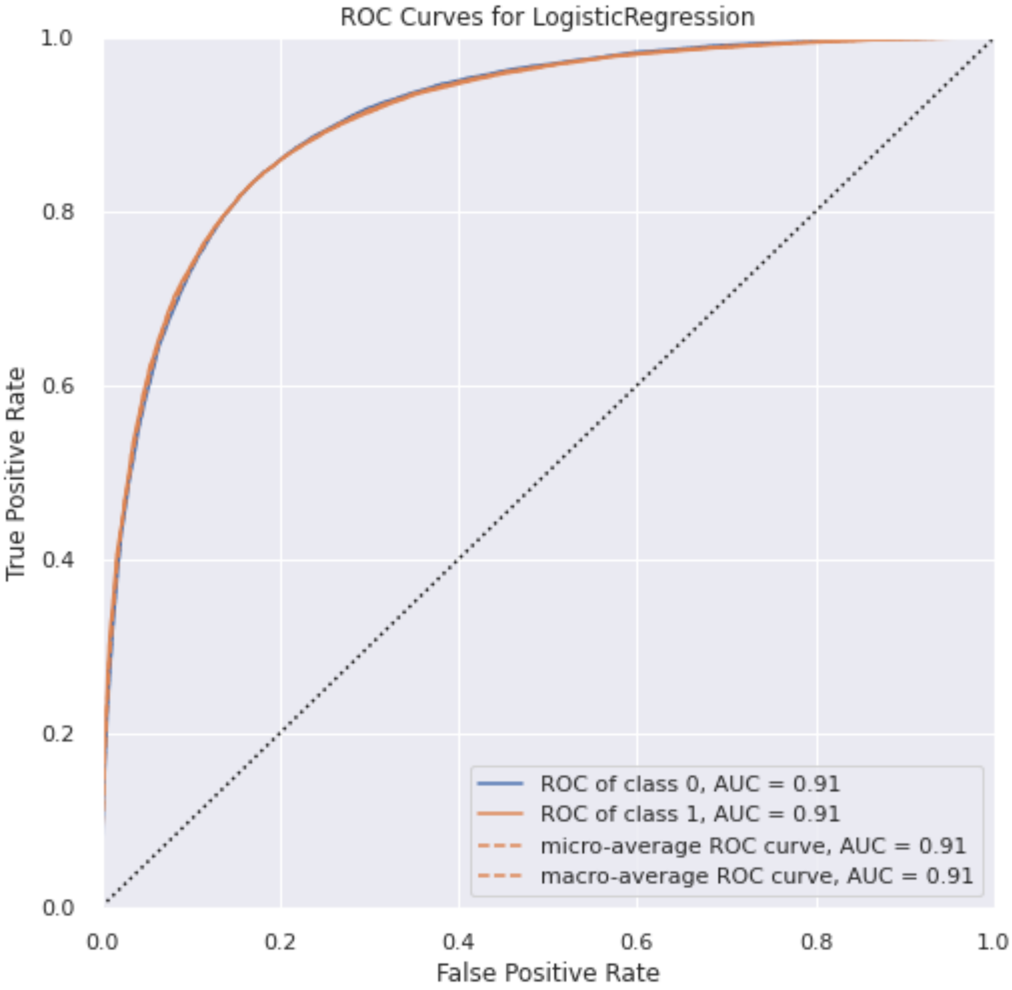
```
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/bas
e.py:444: UserWarning: X has feature names, but LogisticRegression was fitted
without feature names
  f"X has feature names, but {self.__class__.__name__} was fitted without"
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/sklearn/bas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



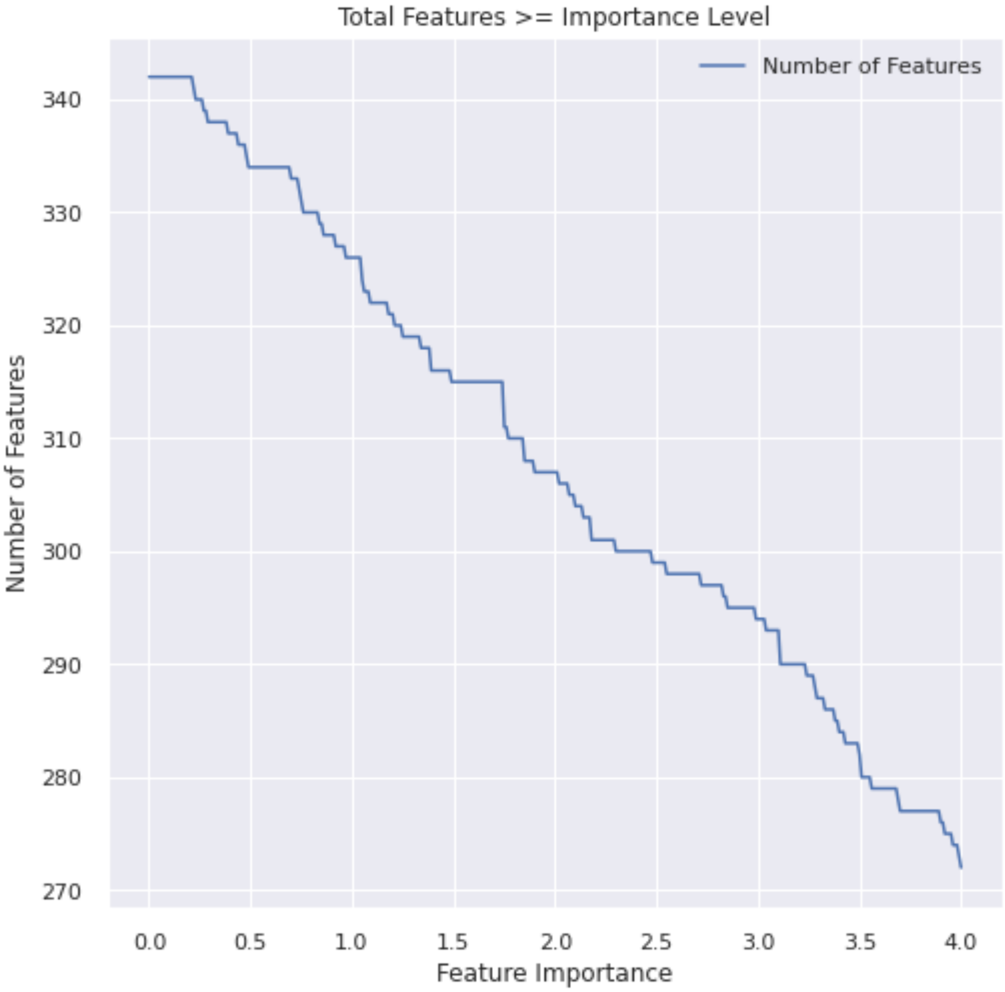
```
<Figure size 576x576 with 0 Axes>
```

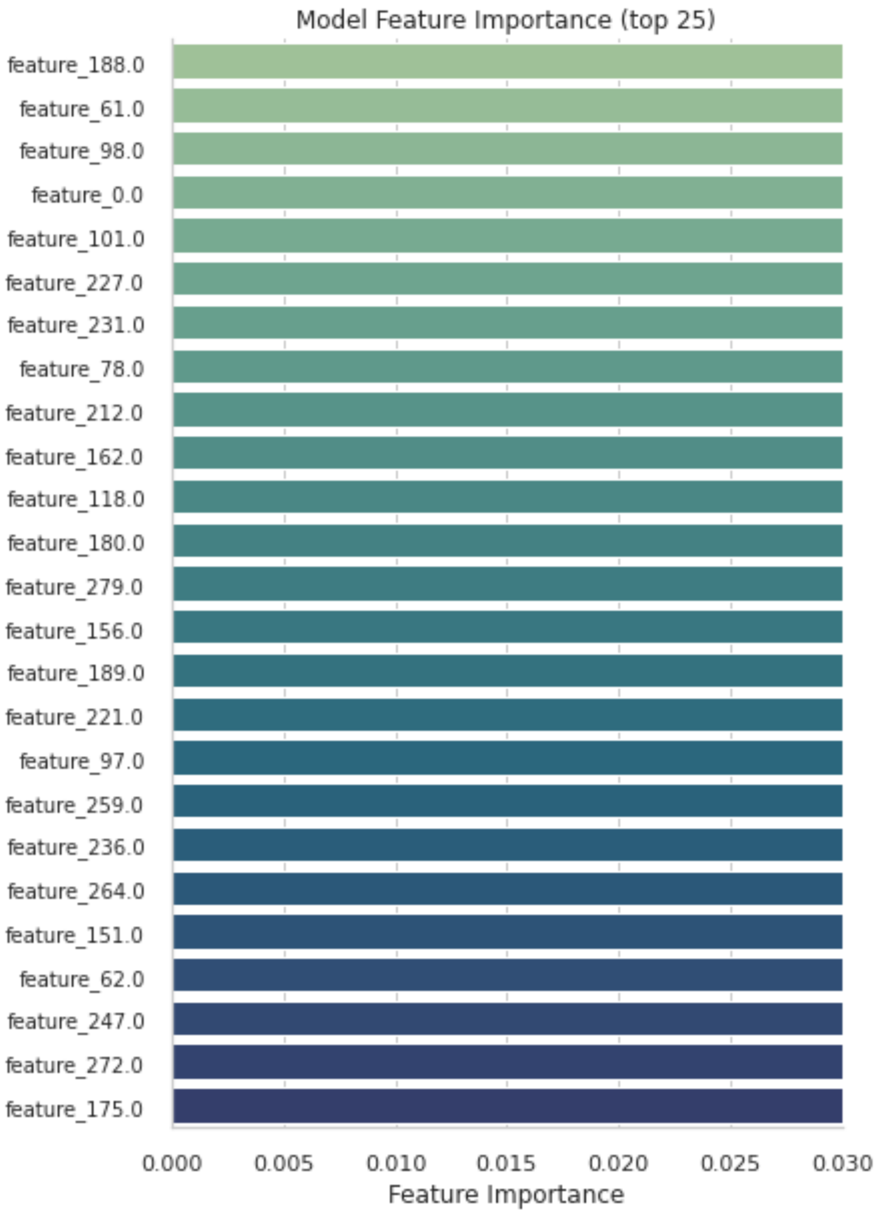


```
Final model ROCAUC not calculated. Starting now
```

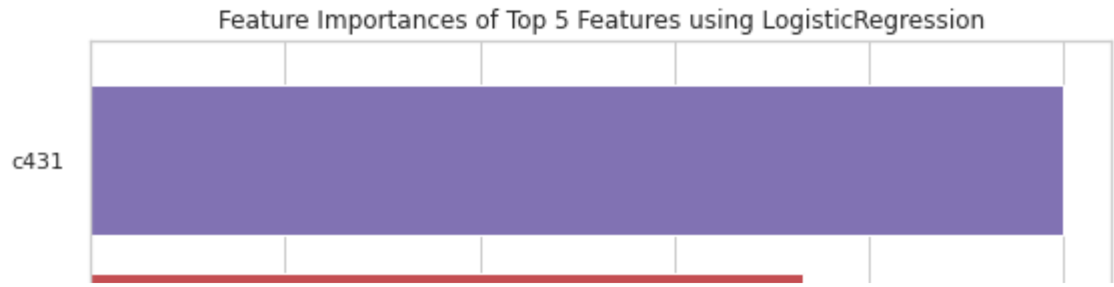


```
0%|          | 0/402 [00:00<?, ?it/s]
```


Total Features >= Importance Level
Number of Features
340
330
320
310
300
290
280
270
Number of Features
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Feature Importance



```
/home/magni/python_env/ML1010_env2/lib64/python3.7/site-packages/yellowbrick/
model_selection/importances.py:199: YellowbrickWarning: detected multi-dimens
ional feature importances but stack=False, using mean to aggregate them.
  YellowbrickWarning,
```



In [16]:

```
myExp.display()
```

```
DataExperiment summary:
---> projectName: ML1010-Group-Project
---> experimentName: ReviewText_Lemma_Bert2_Full (Logistic Regression)
---> isDataPackageLoaded: True
---> isBaseModelLoaded: True
---> isBaseModelPredicted: True
---> isBaseModelLearningCurveCreated: True
---> isFinalModelLoaded: True
---> isFinalModelPredicted: True
---> isFinalModelLearningCurveCreated: True
---> isClassifierLoaded: True
LogisticRegression(max_iter=200)

    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)
```

Scratchpad

In [19]:

```
model = myExp.getClassifier()
# define the datasets to evaluate each iteration
X_train=myExp.dataPackage.getXTrainData()
Y_train=myExp.dataPackage.getYTrainData()
X_test=myExp.dataPackage.getXTestData()
Y_test=myExp.dataPackage.getYTestData()
evalset = [(X_train, Y_train), (X_test,Y_test)]
# fit the model
model.fit(X_train, Y_train, eval_set=evalset)
# evaluate performance
yhat = model.predict(X_test)
score = accuracy_score(y_test, yhat)
print('Accuracy: %.3f' % score)
# retrieve performance metrics
results = model.evals_result()
# plot learning curves
plt.plot(results['validation_0']['logloss'], label='train')
plt.plot(results['validation_1']['logloss'], label='test')
# show the legend
plt.legend()
# show the plot
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipykernel_298457/798865827.py in <module>
      7 evalset = [(X_train, Y_train), (X_test,Y_test)]
      8 # fit the model
----> 9 model.fit(X_train, Y_train, eval_set=evalset)
     10 # evaluate performance
     11 yhat = model.predict(X_test)

TypeError: fit() got an unexpected keyword argument 'eval_set'
```

In [ ]: