In [36]:

```
%load_ext autoreload
%autoreload 2
%matplotlib inline
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [37]:

```
import sys
from pathlib import Path
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = (12, 6)
plt.rcParams['figure.dpi'] = 600

if 'twitter' not in sys.path:
    sys.path.append('twitter')
```

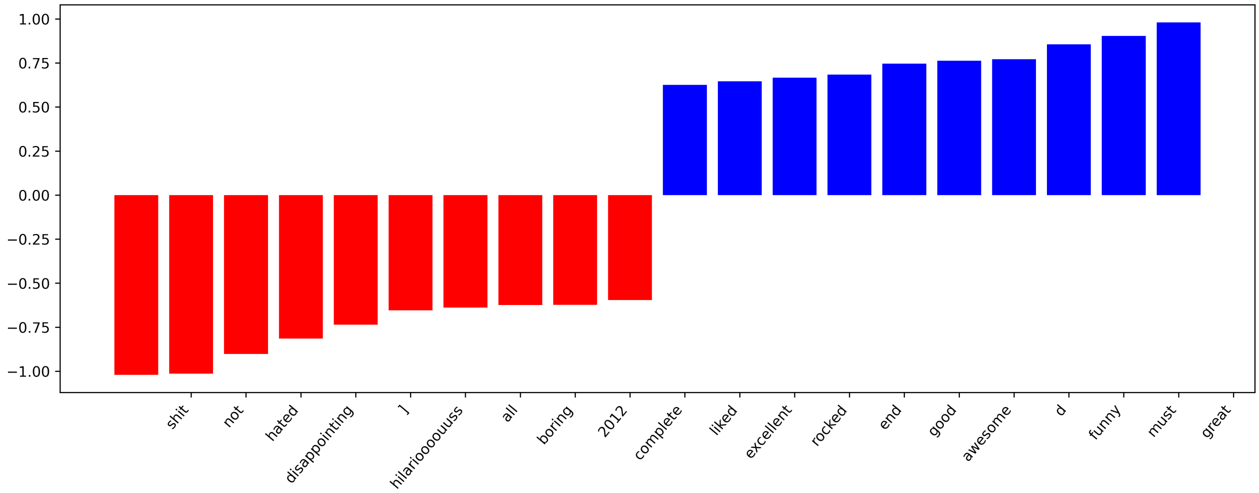
In [38]:

```
import twitter
from twitter import main
```

In [39]:

```
main()
```

```
Linear SVM Hyperparameter Selection
--------------------------------------------------
C_values : [1.e-03 1.e-02 1.e-01 1.e+00 1.e+01 1.e+02]
-----------------------------------------------------
accuracy: [0.655, 0.775, 0.823, 0.85, 0.846, 0.846]
f1_score: [0.792, 0.845, 0.871, 0.887, 0.885, 0.885]
auroc: [0.5, 0.699, 0.786, 0.828, 0.822, 0.822]
precision: [0.655, 0.767, 0.837, 0.876, 0.869, 0.869]
sensitivity: [1.0, 0.943, 0.907, 0.899, 0.902, 0.902]
specificity: [0.0, 0.456, 0.664, 0.757, 0.741, 0.741]
-------------------------------------------------------------
{'accuracy': 1.0, 'f1_score': 1.0, 'auroc': 1.0, 'precision': 1.0, 'sensitivity': 0.001,
'specificity': 1.0}
-------------------------------------------------------------
RBF SVM Hyperparameter Selection based on accuracy:
Maximum score for accuracy  :  0.848
----------------------------------------------------------------------------
RBF SVM Hyperparameter Selection based on f1_score:
Maximum score for f1_score  :  0.887
----------------------------------------------------------------------------
RBF SVM Hyperparameter Selection based on auroc:
Maximum score for auroc  :  0.822
----------------------------------------------------------------------------
RBF SVM Hyperparameter Selection based on precision:
Maximum score for precision  :  0.868
----------------------------------------------------------------------------
RBF SVM Hyperparameter Selection based on sensitivity:
Maximum score for sensitivity  :  1.0
----------------------------------------------------------------------------
RBF SVM Hyperparameter Selection based on specificity:
Maximum score for specificity  :  0.736
----------------------------------------------------------------------------
{'accuracy': (100.0, 0.01), 'f1_score': (100.0, 0.01), 'auroc': (100.0, 0.01), 'precisio
n': (100.0, 0.01), 'sensitivity': (0.001, 0.001), 'specificity': (100.0, 0.01)}
--------------------------------------------------------------------
Best C and gamma 100 0.01
For Linear kernal SVM
---------------------------------------------------------------------------
accuracy  :  0.7540142857142857 0.6571428571428571 0.8571428571428571
f1_score  :  0.824757246390667 0.7391304347826086 0.9
auroc  :  0.7241947416226766 0.5982142857142857 0.8493589743589743
precision  :  0.8696129218165481 0.7719298245614035 0.9574468085106383
sensitivity  :  0.7840628929651736 0.6666666666666666 0.8958333333333334
specificity  :  0.6640161560625207 0.4375 0.875
---------------------------------------------------------------------------
For RBF kernal SVM
---------------------------------------------------------------------------
accuracy  :  0.7554714285714287 0.6428571428571429 0.8428571428571429
f1_score  :  0.8289306148771275 0.7446808510638298 0.9038461538461539
auroc  :  0.7045459172268324 0.5696969696969697 0.8219696969696969
precision  :  0.853977205136828 0.75 0.9423076923076923
sensitivity  :  0.8052831354234621 0.6923076923076923 0.9130434782608695
specificity  :  0.6120251371567256 0.38461538461538464 0.8260869565217391
---------------------------------------------------------------------------
[128 221 507 847  24 169 107 583 236  61]
```



Hyperparameter selection for a linear kerner

1. Because, stratified kfold cross validation is useful when there is imbalanced dataset (may be more -ve tweets or more +ve tweets). Inoder to over come this over or under sample our data to deal with class imbalance. when we want to preserve the class ratio of our target.

1. Almost all the metric scores increases with increse in C, except for sensitivity, which has its best at lowest c. For smaller values of C, the optimization will choose a larger-margin hyperplane, and hence larger margin is choosen and hence most of the values of True positive as recalled. where as in specificity its zero, as it was not able to predict the Flase negatives.

Beat c given was 1.0

Hyperparameter selection for an RBF-Kernal

1. For RBF kernal most of the metrics are giving its best at C = 100 and gamma 0.01, except for sensitivity which gives its best with lowest C and gamma with lowest. h

Test Set Performace:

1. Has most of the metrics prefer C = 100 abd gamma = 0.01. we shall consider that to be the best Hyperparameters. and for linear kernal it is c with 1.0, as most of the metrics gives its best at C = 1.0

2. Mean of Accuracy, f1-score and precision seems to be same for both,Rbf and linear kernal with very littel differnce. RBF seems to be more sensitive. Linear kernal takes lead in the rest with slight difference. Even the bootstrapping lower and upper bound does not seem to vary a lot. But since accuracy anf F1 (which is commanly used) scaore is slighly better for RBF kernal, it seems that RBf kernal performs better.

Feature Importance

1. I have used Coef_ to find the top/important features
2. The graph shown above to indicate the top 10 negative and top 10 positive features negative words : shit, not, hated, dissapointing,] , hilariooouss, all, boring, 2012, complete positve words : liked, excellent, rocked, end, awesome, d, funny, must, great The words provide good insight but the few words like d,] seeems to be not reasonable.

Explaination

1. Brief explaination on sentiment analysis
2. let me use the graph above, for explaination. given the list of tweets below, if we use the feature of coef from the model gives us a weight, and this weight has a direction and this direction gives us the predicting class. from this we can find the important words that are affecting the model.