Extension_with_Twitter_Data

May 13, 2019

1 Extension - Twitter Dataset

In [0]:
```
import pandas as pd
```

In [0]:
```
import numpy as np
```

In [0]:
```
!pip install vaderSentiment
```

```
Requirement already satisfied: vaderSentiment in c:\users\chandana priya\anaconda3\lib\site-pac
```

In [0]:
```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer, TfidfTransformer
import nltk
import gensim
from nltk.tokenize import sent_tokenize, word_tokenize
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from scipy.sparse import hstack
from sklearn.metrics import average_precision_score
from sklearn.metrics import roc_auc_score
from nltk import word_tokenize,sent_tokenize
from gensim import corpora
from sklearn.pipeline import Pipeline
from sklearn.model_selection import cross_val_predict
import matplotlib.pyplot as plt
```

```
C:\Users\chandana priya\Anaconda3\lib\site-packages\gensim\utils.py:1209: UserWarning: detected
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
```

In [0]:
```
df_train = pd.read_csv("all_tweets.csv")[["tweet","class"]]

df_train.loc[df_train['class'] == 2, 'class'] = 1
```

In [0]: `df_train.head()`

Out[0]:
```
                                               tweet  class
0  As of March 13th , 2014 , the booklet had been...      1
1  Thank you in advance. : ) Download the youtube...      1
2  In order to help increase the booklets downloa...      1
3  ( Simply copy and paste the following text int...      1
4  Click below for a FREE download of a colorfull...      0
```

In [0]:
```
X_train = df_train[df_train['class']==0]["tweet"]
X__train = df_train[df_train['class']==1]["tweet"][:1000]
X_train = X_train.append(X__train)
y_train = df_train[df_train['class']==0]["class"]
y_train = y_train.append(df_train[df_train['class']==1]["class"][:1000])

len(X_train)
```

Out[0]: 2196

In [0]: `len(y_train)`

Out[0]: 2196

1.1 Baseline model

In [0]:
```
pipe  =  make_pipeline(CountVectorizer(),LogisticRegression(solver="sag"))
print("Cross val score on baseline model")
pipe.fit(X_train,y_train)
print(np.mean(cross_val_score(pipe,X_train,y_train,cv=5,scoring="roc_auc")))
```

```
Cross val score on baseline model


C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergen
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergen
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergen
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergen
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergen
  "the coef_ did not converge", ConvergenceWarning)


0.762642730125523


C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergen
  "the coef_ did not converge", ConvergenceWarning)
```

1.2 Re-implementing the paper

1. Lowercase
2. Stem
3. bigram, unigram, trigram features, weighted by its tfidf
4. POS tag
5. FK Grade level
6. FK reading ease score
7. sentiment scores
8. binary indicators for: hashtags, mentions, retweets, urls
9. count indicators for :hashtags, mentions, retweets, urls
10. number of characters
11. numbers of words
12. number of syllables

In [0]:
```
nltk.download("stopwords")
from nltk.stem.porter import *
```

```
[nltk_data] Downloading package stopwords to C:\Users\chandana
[nltk_data]     priya\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [0]:
```
stopwords=stopwords = nltk.corpus.stopwords.words("english")

other_exclusions = ["#ff", "ff", "rt","RT"]
stopwords.extend(other_exclusions)

stemmer = PorterStemmer()

def preprocess(text_string):

    #Lowercase string
    text_string=text_string.lower()
    space_pattern = '\s+'
    giant_url_regex = ('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|'
        '[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+')
    mention_regex = '@[\w\-]+'
    hashtag_regex = '#[\w\-]+'
    parsed_text = re.sub(space_pattern, ' ', text_string)
    parsed_text = re.sub(giant_url_regex, 'URLHERE', parsed_text)
    parsed_text = re.sub(mention_regex, 'MENTIONHERE', parsed_text)
```

```
        parsed_text = re.sub(hashtag_regex, 'HASHTAGHERE', parsed_text)

        #Stem it
        #tweet = " ".join(re.split("[^a-zA-Z]*", parsed_text)).strip()
        tweet = parsed_text
        tokens = [stemmer.stem(t) for t in tweet.split()]
        return tokens


    def pos_tag_seq(tokens):
        tags = nltk.pos_tag(tokens)
        tag_list = [x[1] for x in tags]
        tag_str = " ".join(tag_list)
        return tag_str
```

In [0]:
```
def join_sent(l):
      return " ".join(l)
```

In [0]:
```
df_train=pd.DataFrame(X_train)
```

In [0]:
```
df_train.columns=["tweet"]
```

In [0]:
```
df_train.head()
```

Out[0]:
```
                                                tweet
4   Click below for a FREE download of a colorfull...
12  Click below for a FREE download of a colorfull...
19  She may or may not be a Jew but she 's certain...
21  God knows what would come out of Africa if onl...
24  I'm insulted by how stupid the Jews think we a...
```

In [0]:
```
s_train=df_train['tweet'].apply(preprocess)
```

In [0]:
```
s_tr=s_train.apply(join_sent)
```

In [0]:
```
nltk.download('averaged_perceptron_tagger')
t_tr=s_train.apply(pos_tag_seq)
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\chandana priya\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

In [0]:
```
vectorizer = TfidfVectorizer(
    preprocessor=None,
    lowercase=False,
```

```
            ngram_range=(1, 3),
            use_idf=True,
            smooth_idf=False,
            norm=None,
            stop_words=stopwords,
            decode_error='replace',
            max_features=10000,
            min_df=5,
            max_df=0.75
            )
```

```
In [0]: pos_vectorizer = TfidfVectorizer(
            tokenizer=None,
            lowercase=False,
            preprocessor=None,
            ngram_range=(1, 3),
            stop_words=None,
            use_idf=False,
            smooth_idf=False,
            norm=None,
            decode_error='replace',
            max_features=5000,
            min_df=5,
            max_df=0.75,
            )
```

```
In [0]: tfidf_tr = vectorizer.fit_transform(s_tr).toarray()

        vocab = {v:i for i, v in enumerate(vectorizer.get_feature_names())}
        idf_vals = vectorizer.idf_
        idf_dict = {i:idf_vals[i] for i in vocab.values()}
```

```
In [0]: pos_tr = pos_vectorizer.fit_transform(t_tr).toarray()

        pos_vocab = {v:i for i, v in enumerate(pos_vectorizer.get_feature_names())}
```

```
In [0]: from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer as VS
        sentiment_analyzer = VS()
```

```
In [0]: def get_sentiment(text):
          sentiment = sentiment_analyzer.polarity_scores(text)
          return sentiment

        #   return sentiment["neg"], sentiment["pos"], sentiment["neu"]
```

```
In [0]: df_train["sent"]=df_train["tweet"].apply(get_sentiment)
```

```
In [0]: df_train.head()
```

```
Out[0]:                                                  tweet  \
        4   Click below for a FREE download of a colorfull...
        12  Click below for a FREE download of a colorfull...
        19  She may or may not be a Jew but she 's certain...
        21  God knows what would come out of Africa if onl...
        24  I'm insulted by how stupid the Jews think we a...

                                                          sent
        4   {'neg': 0.15, 'neu': 0.687, 'pos': 0.163, 'com...
        12  {'neg': 0.15, 'neu': 0.687, 'pos': 0.163, 'com...
        19  {'neg': 0.182, 'neu': 0.647, 'pos': 0.171, 'co...
        21  {'neg': 0.0, 'neu': 0.884, 'pos': 0.116, 'comp...
        24  {'neg': 0.218, 'neu': 0.782, 'pos': 0.0, 'comp...
```

```
In [0]: foo_tr = lambda x: pd.Series([x["pos"],x["neg"],x["neu"]])
        rev_tr = df_train['sent'].apply(foo_tr)
```

```
In [0]: rev_tr.columns=["pos","neg","neu"]
```

```
In [0]: rev_tr.head()
```

```
Out[0]:       pos    neg    neu
        4   0.163  0.150  0.687
        12  0.163  0.150  0.687
        19  0.171  0.182  0.647
        21  0.116  0.000  0.884
        24  0.000  0.218  0.782
```

1.3 Binary count for URL https mentions etc

```
In [0]: def return_cont(parsed_text):
          return(parsed_text.count('urlher'),parsed_text.count('mentionher'),parsed_text.count
```

```
In [0]: df_train["counts"]=s_tr.apply(return_cont)
```

```
In [0]: df_train["counts"].head()
```

```
Out[0]: 4     (0, 0, 0)
        12    (0, 0, 0)
        19    (0, 0, 0)
        21    (0, 0, 0)
        24    (0, 0, 0)
        Name: counts, dtype: object
```

```
In [0]: foo = lambda x: pd.Series([x[0],x[1],x[2]])
        mention_counts_tr = df_train['counts'].apply(foo)
```

```
In [0]: mention_counts_tr.head()
```

Out[0]:
	0	1	2
4	0	0	0
12	0	0	0
19	0	0	0
21	0	0	0
24	0	0	0

1.4 FKRA and Flesch and number of syllables etc

In [0]:
```
!pip install textstat
from textstat.textstat import *
```

```
Requirement already satisfied: textstat in c:\users\chandana priya\anaconda3\lib\site-packages
Requirement already satisfied: repoze.lru in c:\users\chandana priya\anaconda3\lib\site-package
Requirement already satisfied: pyphen in c:\users\chandana priya\anaconda3\lib\site-packages (
```

In [0]:
```
def get_other_features(text):
    space_pattern = '\s+'
    giant_url_regex = ('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|'
        '[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+')
    mention_regex = '@[\w\-]+'
    parsed_text = re.sub(space_pattern, ' ', text)
    parsed_text = re.sub(giant_url_regex, '', parsed_text)
    words = re.sub(mention_regex, '', parsed_text)

    syllables = textstat.syllable_count(words)
    num_chars = sum(len(w) for w in words)
    num_chars_total = len(text)
    num_terms = len(text.split())
    num_words = len(words.split())
    avg_syl = round(float((syllables+0.001))/float(num_words+0.001),4)
    num_unique_terms = len(set(words.split()))

    ###Modified FK grade, where avg words per sentence is just num words/1
    FKRA = round(float(0.39 * float(num_words)/1.0) + float(11.8 * avg_syl) - 15.59,1)
    ##Modified FRE score, where sentence fixed to 1
    FRE = round(206.835 - 1.015*(float(num_words)/1.0) - (84.6*float(avg_syl)),2)

    features = [FKRA, FRE,syllables, avg_syl, num_chars, num_terms, num_words,
                num_unique_terms]
    return features
```

In [0]:
```
other_feats_tr=df_train["tweet"].apply(get_other_features)
```

In [0]:
```
other_feats_tr.head()
```

```
Out[0]: 4      [15.5, 23.0, 42, 1.909, 151, 22, 22, 20]
        12     [15.5, 23.0, 42, 1.909, 151, 22, 22, 20]
        19     [13.6, 68.95, 46, 1.15, 183, 40, 40, 36]
        21    [7.2, 75.77, 24, 1.3333, 100, 18, 18, 17]
        24    [10.0, 75.3, 35, 1.2069, 145, 29, 29, 26]
        Name: tweet, dtype: object
```

```
In [0]: other_features_names = ["FKRA", "FRE","num_syllables", "avg_syl_per_word", "num_chars"
```

```
In [0]: foo = lambda x: pd.Series(elem for elem in x)
        of_counts_tr = other_feats_tr.apply(foo)
```

```
In [0]: of_counts_tr.head()
```

```
Out[0]:       0      1     2       3      4     5     6     7
        4   15.5  23.00  42.0  1.9090  151.0  22.0  22.0  20.0
        12  15.5  23.00  42.0  1.9090  151.0  22.0  22.0  20.0
        19  13.6  68.95  46.0  1.1500  183.0  40.0  40.0  36.0
        21   7.2  75.77  24.0  1.3333  100.0  18.0  18.0  17.0
        24  10.0  75.30  35.0  1.2069  145.0  29.0  29.0  26.0
```

```
In [0]: of_counts_tr.columns=other_features_names
```

```
In [0]: of_counts_tr.head()
```

```
Out[0]:     FKRA    FRE  num_syllables  avg_syl_per_word  num_chars  num_terms  \
        4   15.5  23.00           42.0            1.9090      151.0       22.0
        12  15.5  23.00           42.0            1.9090      151.0       22.0
        19  13.6  68.95           46.0            1.1500      183.0       40.0
        21   7.2  75.77           24.0            1.3333      100.0       18.0
        24  10.0  75.30           35.0            1.2069      145.0       29.0

            num_words  num_unique_words
        4        22.0              20.0
        12       22.0              20.0
        19       40.0              36.0
        21       18.0              17.0
        24       29.0              26.0
```

```
In [0]: df_train.drop([ "sent","counts"], axis=1)
```

```
Out[0]:                                                   tweet
        4     Click below for a FREE download of a colorfull...
        12    Click below for a FREE download of a colorfull...
        19    She may or may not be a Jew but she 's certain...
        21    God knows what would come out of Africa if onl...
        24    I'm insulted by how stupid the Jews think we a...
        30    Does it surprise you that jews promote faggotry ?
        31    It 's another way to destroy Hungarians on top...
```

```
32    I watch fights on gay parade in budapest toget...
35    i have been talking about it for years. the fi...
40    We have a little piece of legislation called t...
66    I know one thing , no white should ever give o...
68    The white man has no future in Canada , that i...
96    Were Whites to leave in large numbers leaving ...
137   If you want to hear what a black `` intellectu...
147   I also , unfortuantely , spent 7-8th grade in ...
149   It wasnt really that dangerous perse , just st...
152   I was basically a babysitter , and the apes we...
158   Went outside for recess , and noticed two nigl...
165   I like to listen to this preacher , but why do...
169   I 've always considered teaching as one of the...
170   And the sad thing is the white students at tho...
173   I could n't imagine living in those conditions...
192   I feel so sorry for all you people living ther...
193   It is in the nature of Jews to desecrate tradi...
222   My wife had to spend time at the UP Hospital i...
224   Their gods often had pale skin and blue eyes b...
241   As usual the magic negro opens his mouth and l...
242   If lies were money he would be the richest neg...
249    Great image of the Jew as a poisonous mushroom .
255   The majority of the people in these photos are...
...                                                ...
1165  I think it has something to do with the type o...
1166   BTW - I do n't recall seeing a Mars Bar in S.A ?
1167  I never drink diet anything either because of ...
1168  I try to avoid the GMO stuff too , but since t...
1169  And then you need to avoid High Fructose Corn ...
1170             You also need to avoid anything Soya .
1171  And then there is the pesticides on your fresh...
1172  And the non-edible chemicals in the processed ...
1175  Well we must send money so you can arm yoursel...
1176  I saw on the news that one can buy an AK47 for...
1177  Many years ago maybe , there 's no ammo for th...
1180  YouTube - & #x202a ; Kenya - Hacked to death b...
1181                   This is black behavior as well .
1183  Seems none of these pinheads wanted to get dir...
1184  YouTube - & #x202a ; India ministers watch as ...
1185  MY father is American but I would never think ...
1186  I 'll go eat hagis and sip whiskey with the Sc...
1188  YouTube - & #x202a ; Unruly Teens wrecking a D...
1189  YouTube - & #x202a ; Police Brutality In Jamai...
1190  YouTube - Grandson attacks and rapes 87-Year-O...
1191  YouTube - BLACK WOMAN GOES BALLISTIC ON REPORT...
1192  And it works in our favor the other way too - ...
1193  This is the latest anti-white attack by a ' gr...
1195  Swinging On Both Females Guys Nobody Tries To ...
```

```
1197           It 's like that at most public boards .
1199  YouTube - Cops taser a violent drunk 12 year o...
1200  This thread makes me thankful that I will be h...
1202  YouTube - Black guy throws baby into traffic Y...
1203  Racist attack by blacks on a young white girl ...
1204  Negro holding the camera laughs with glee : Da...

[2196 rows x 1 columns]
```

In [0]:
```
for elem in [pd.DataFrame(tfidf_tr),pd.DataFrame(pos_tr),rev_tr,mention_counts_tr, of_c
    print(len(elem))
```

```
2196
2196
2196
2196
2196
```

In [0]:
```
# x_train=np.column_stack([tfidf,pos,rev,mention_counts, other_feats])
x_train=np.concatenate([pd.DataFrame(tfidf_tr),pd.DataFrame(pos_tr),rev_tr,mention_coun


x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.2)
```

In [0]:
```
print(len(x_train), len(y_train))
```

```
1756 1756
```

In [0]:
```
param_grid = {"logisticregression__C": [100,10,1,0.1,0.01],}
grid = GridSearchCV(make_pipeline(LogisticRegression(solver="sag"),memory="cache_folder
```

In [0]:
```
grid.fit(x_train, y_train)
```

```
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
```

```
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\linear_model\sag.py:326: Convergenc
  "the coef_ did not converge", ConvergenceWarning)
```

```
Out[0]: GridSearchCV(cv=5, error_score='raise',
              estimator=Pipeline(memory='cache_folder',
             steps=[('logisticregression', LogisticRegression(C=1.0, class_weight=None, dual=Fa
                  intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                  penalty='l2', random_state=None, solver='sag', tol=0.0001,
                  verbose=0, warm_start=False))]),
              fit_params=None, iid=True, n_jobs=1,
              param_grid={'logisticregression__C': [100, 10, 1, 0.1, 0.01]},
```

```
             pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
             scoring='roc_auc', verbose=0)
```

In [0]: grid.best_score_

Out[0]: 0.743899512313582

In [0]: grid.cv_results_

```
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:122: FutureWa
  warnings.warn(*warn_args, **warn_kwargs)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:122: FutureWa
  warnings.warn(*warn_args, **warn_kwargs)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:122: FutureWa
  warnings.warn(*warn_args, **warn_kwargs)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:122: FutureWa
  warnings.warn(*warn_args, **warn_kwargs)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:122: FutureWa
  warnings.warn(*warn_args, **warn_kwargs)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:122: FutureWa
  warnings.warn(*warn_args, **warn_kwargs)
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:122: FutureWa
  warnings.warn(*warn_args, **warn_kwargs)
```

Out[0]:
```
{'mean_fit_time': array([21.30317044, 21.89279237, 10.6701993 ,  9.66510706,  9.788574
 'std_fit_time': array([3.3566574 , 4.25104059, 1.45909786, 0.66716713, 0.85565639]),
 'mean_score_time': array([0.10899835, 0.01178164, 0.00558381, 0.00498772, 0.0051877 ]
 'std_score_time': array([0.17606636, 0.00313124, 0.00048935, 0.00063113, 0.00074662])
 'param_logisticregression__C': masked_array(data=[100, 10, 1, 0.1, 0.01],
              mask=[False, False, False, False, False],
        fill_value='?',
             dtype=object),
 'params': [{'logisticregression__C': 100},
  {'logisticregression__C': 10},
  {'logisticregression__C': 1},
  {'logisticregression__C': 0.1},
  {'logisticregression__C': 0.01}],
 'split0_test_score': array([0.72356932, 0.72356932, 0.72376408, 0.72360178, 0.7234070
 'split1_test_score': array([0.75561284, 0.75567811, 0.75561284, 0.75571074, 0.7552538
 'split2_test_score': array([0.80606318, 0.80599791, 0.80603054, 0.80599791, 0.8044968
 'split3_test_score': array([0.71746508, 0.71746508, 0.71746508, 0.71746508, 0.7171713
 'split4_test_score': array([0.71684506, 0.71661663, 0.7166819 , 0.71677979, 0.7163229
 'mean_test_score': array([0.74389951, 0.74385385, 0.74389942, 0.7438995 , 0.74331906]
 'std_test_score': array([0.03418599, 0.03420302, 0.03417692, 0.03417548, 0.03374025])
 'rank_test_score': array([1, 4, 3, 2, 5]),
 'split0_train_score': array([0.7513767 , 0.7512584 , 0.75135834, 0.75115643, 0.750819
 'split1_train_score': array([0.74238628, 0.74236388, 0.74237406, 0.74240665, 0.741860
 'split2_train_score': array([0.85271389, 0.85284221, 0.85267112, 0.85246947, 0.849748
```

```
'split3_train_score': array([0.75281492, 0.75287398, 0.75281899, 0.75280066, 0.752448
'split4_train_score': array([0.74873716, 0.74879012, 0.74874938, 0.74872494, 0.748217
'mean_train_score': array([0.76960579, 0.76962572, 0.76959438, 0.76951163, 0.76861899
'std_train_score': array([0.04170753, 0.04176193, 0.04169213, 0.04162917, 0.04072461]
```

In [0]:
```python
grid.best_params_
```

Out[0]:
```
{'logisticregression__C': 100}
```

In [0]:
```python
from sklearn.feature_selection import SelectFromModel
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import SelectFromModel
from sklearn.metrics import classification_report
from sklearn.svm import LinearSVC,SVC
from sklearn.model_selection import StratifiedKFold, GridSearchCV
from sklearn.pipeline import Pipeline
```

In [0]:
```python
tuned_parameters = [{'kernel': ['rbf'], 'gamma': [0.01, 10,100],
                    'C': [1, 10, 100, 1000]},
                   {'kernel': ['linear'], 'C': [0.1,0.01, 1, 10, 100]}]

scores = ['precision', 'recall']

for score in scores:
    print("# Tuning hyper-parameters for %s" % score)
    print()
    clf = GridSearchCV(SVC(), tuned_parameters, cv=5,
                       scoring='%s_macro' % score)
    clf.fit(x_train, y_train)

    print("Best parameters set found on development set:")
    print()
    print(clf.best_params_)
    print()
    print("Grid scores on development set:")
    print()
    means = clf.cv_results_['mean_test_score']
    stds = clf.cv_results_['std_test_score']
    for mean, std, params in zip(means, stds, clf.cv_results_['params']):
        print("%0.3f (+/-%0.03f) for %r"
              % (mean, std * 2, params))
    print()

    print("Detailed classification report:")
    print()
    print("The model is trained on the full development set.")
    print("The scores are computed on the full evaluation set.")
    print()
    y_true, y_pred = y_test, clf.predict(x_test)
```

```
        print(classification_report(y_true, y_pred))
        print()
```

```
In [0]: scores = ['precision', 'recall']
        penalty = ['l1', 'l2']

        logistic = LogisticRegression()
        # Create regularization hyperparameter space
        C = np.logspace(0, 4, 10)

        # Create hyperparameter options
        hyperparameters = dict(C=C, penalty=penalty)

        for score in scores:
            print("# Tuning hyper-parameters for %s" % score)
            print()
                                   scoring='%s_macro' % score)
            clf.fit(x_train, y_train)

            print("Best parameters set found on development set:")
            print()
            print(clf.best_params_)
            print()
            print("Grid scores on development set:")
            print()
            means = clf.cv_results_['mean_test_score']
            stds = clf.cv_results_['std_test_score']
            for mean, std, params in zip(means, stds, clf.cv_results_['params']):
                print("%0.3f (+/-%0.03f) for %r"
                      % (mean, std * 2, params))
            print()

            print("Detailed classification report:")
            print()
            print("The model is trained on the full development set.")
            print("The scores are computed on the full evaluation set.")
            print()
            y_true, y_pred = y_test, clf.predict(x_test)
            print(classification_report(y_true, y_pred))
            print()
```

```
# Tuning hyper-parameters for precision



/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
```

```
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
```

```
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
```

```
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
```

```
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
```

```
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)


Best parameters set found on development set:

{'C': 1.0, 'penalty': 'l1'}

Grid scores on development set:

0.801 (+/-0.021) for {'C': 1.0, 'penalty': 'l1'}
0.787 (+/-0.026) for {'C': 1.0, 'penalty': 'l2'}
0.788 (+/-0.020) for {'C': 2.7825594022071245, 'penalty': 'l1'}
0.781 (+/-0.039) for {'C': 2.7825594022071245, 'penalty': 'l2'}
0.772 (+/-0.036) for {'C': 7.742636826811269, 'penalty': 'l1'}
0.778 (+/-0.053) for {'C': 7.742636826811269, 'penalty': 'l2'}
0.759 (+/-0.038) for {'C': 21.544346900318832, 'penalty': 'l1'}
0.773 (+/-0.060) for {'C': 21.544346900318832, 'penalty': 'l2'}
0.759 (+/-0.029) for {'C': 59.94842503189409, 'penalty': 'l1'}
0.769 (+/-0.055) for {'C': 59.94842503189409, 'penalty': 'l2'}
0.752 (+/-0.036) for {'C': 166.81005372000593, 'penalty': 'l1'}
0.768 (+/-0.057) for {'C': 166.81005372000593, 'penalty': 'l2'}
0.753 (+/-0.035) for {'C': 464.15888336127773, 'penalty': 'l1'}
0.767 (+/-0.054) for {'C': 464.15888336127773, 'penalty': 'l2'}
0.740 (+/-0.054) for {'C': 1291.5496650148827, 'penalty': 'l1'}
0.767 (+/-0.051) for {'C': 1291.5496650148827, 'penalty': 'l2'}
0.728 (+/-0.036) for {'C': 3593.813663804626, 'penalty': 'l1'}
0.764 (+/-0.051) for {'C': 3593.813663804626, 'penalty': 'l2'}
0.716 (+/-0.057) for {'C': 10000.0, 'penalty': 'l1'}
0.766 (+/-0.051) for {'C': 10000.0, 'penalty': 'l2'}

Detailed classification report:

The model is trained on the full development set.
The scores are computed on the full evaluation set.

              precision    recall  f1-score   support

           0       0.86      0.85      0.85       288
           1       0.79      0.79      0.79       198

   micro avg       0.83      0.83      0.83       486
   macro avg       0.82      0.82      0.82       486
```

```
weighted avg       0.83      0.83      0.83       486


# Tuning hyper-parameters for recall



/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
```

```
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
```

```
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
```

```
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
```

```
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/logistic.py:433: FutureWarning: De
  FutureWarning)


Best parameters set found on development set:

{'C': 1.0, 'penalty': 'l1'}

Grid scores on development set:

0.799 (+/-0.024) for {'C': 1.0, 'penalty': 'l1'}
0.785 (+/-0.026) for {'C': 1.0, 'penalty': 'l2'}
0.786 (+/-0.022) for {'C': 2.7825594022071245, 'penalty': 'l1'}
0.779 (+/-0.035) for {'C': 2.7825594022071245, 'penalty': 'l2'}
0.771 (+/-0.029) for {'C': 7.742636826811269, 'penalty': 'l1'}
0.775 (+/-0.048) for {'C': 7.742636826811269, 'penalty': 'l2'}
0.758 (+/-0.031) for {'C': 21.544346900318832, 'penalty': 'l1'}
0.772 (+/-0.055) for {'C': 21.544346900318832, 'penalty': 'l2'}
0.758 (+/-0.024) for {'C': 59.94842503189409, 'penalty': 'l1'}
0.768 (+/-0.050) for {'C': 59.94842503189409, 'penalty': 'l2'}
0.752 (+/-0.028) for {'C': 166.81005372000593, 'penalty': 'l1'}
0.768 (+/-0.052) for {'C': 166.81005372000593, 'penalty': 'l2'}
0.744 (+/-0.049) for {'C': 464.15888336127773, 'penalty': 'l1'}
0.766 (+/-0.050) for {'C': 464.15888336127773, 'penalty': 'l2'}
0.739 (+/-0.056) for {'C': 1291.5496650148827, 'penalty': 'l1'}
0.767 (+/-0.047) for {'C': 1291.5496650148827, 'penalty': 'l2'}
0.731 (+/-0.064) for {'C': 3593.813663804626, 'penalty': 'l1'}
0.763 (+/-0.048) for {'C': 3593.813663804626, 'penalty': 'l2'}
0.718 (+/-0.052) for {'C': 10000.0, 'penalty': 'l1'}
0.766 (+/-0.048) for {'C': 10000.0, 'penalty': 'l2'}

Detailed classification report:
```

```
The model is trained on the full development set.
The scores are computed on the full evaluation set.

              precision    recall  f1-score   support

           0       0.86      0.85      0.85       288
           1       0.79      0.79      0.79       198

   micro avg       0.83      0.83      0.83       486
   macro avg       0.82      0.82      0.82       486
weighted avg       0.83      0.83      0.83       486
```

1.5 Chosing a model

From the given models, we choose using an SVC with the parameters 'C': 0.01, 'kernel': 'linear' This is because given the task of detecting hate speech requires that we maximize the amount of hate speech recognized from actual hate speech there exists. The cost of not recognizing hate speech is higher than the cost of recognizing false positives. Hence given a trade off, we chose to look for higher recall and hence settled on SVC model. The authors also did a similar thing.

```
In [0]: model=SVC(class_weight='balanced',C=0.01, kernel='linear',probability=True)
        model.fit(x_train,y_train)
        y_preds=model.predict(x_train)
        report = classification_report( y_preds, y_train )
```

1.6 Evaluating Model Performance

This section is divided into the following major parts : - In-sample predictive performance - Out-of-sample predictive performance - Effects of statistical significance on Predictive Power

Detailed explanations of the above parts follow.

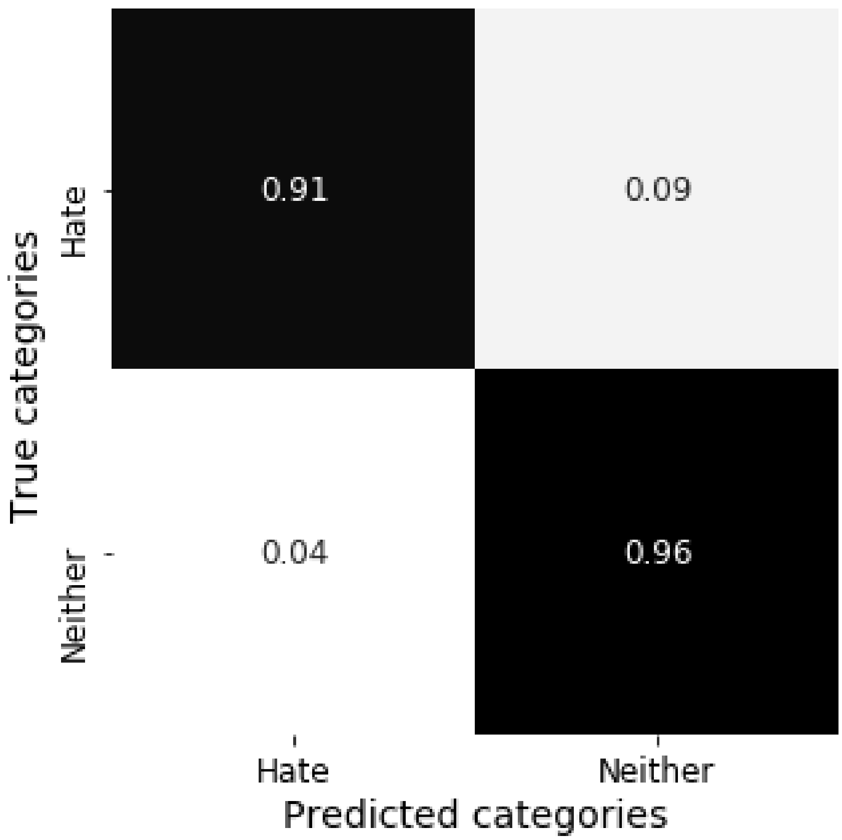
1.6.1 In-Sample Predictive Performance

We are interested in the training accuracy here, in other words the model is tested on data sampled from within the training set.

Logistic Regression is used here to obtain the best performing features using the "SelectFrom-Model" function and the Linear SVC model is trained and tested for performance.

```
In [0]: from sklearn.metrics import confusion_matrix
        import seaborn
        confusion_matrix = confusion_matrix(y_train,y_preds)
        matrix_proportions = np.zeros((2,2))
        for i in range(0,2):
            matrix_proportions[i,:] = confusion_matrix[i,:]/float(confusion_matrix[i,:].sum())
        names=['Hate','Neither']
        confusion_df = pd.DataFrame(matrix_proportions, index=names,columns=names)
```

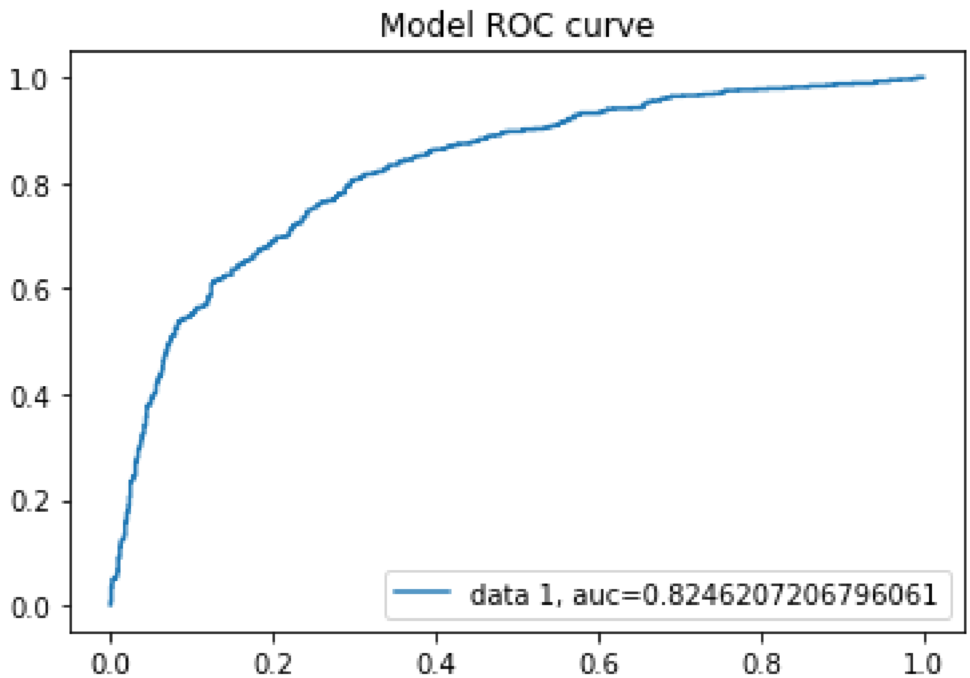
```
plt.figure(figsize=(5,5))
seaborn.heatmap(confusion_df,annot=True,annot_kws={"size": 12},cmap='gist_gray_r',cbar=
plt.ylabel(r'True categories',fontsize=14)
plt.xlabel(r'Predicted categories',fontsize=14)
plt.tick_params(labelsize=12)
```




```
In [0]: from pandas import read_csv
        from matplotlib import pyplot
        import pandas as pd
        # load results file
        results = pd.DataFrame()
        results['A'] = y_train
        results['B'] = y_preds
        # descriptive stats
        print(results.describe())
        # box and whisker plot
        results.boxplot()
        pyplot.show()
        # histogram
        results.hist()
        pyplot.show()
```

```
---------------------------------------------------------------------------

ValueError                                Traceback (most recent call last)

<ipython-input-122-61d09a516950> in <module>
      5 results = pd.DataFrame()
      6 results['A'] = y_train
----> 7 results['B'] = y_preds
      8 # descriptive stats
      9 print(results.describe())


~\Anaconda3\lib\site-packages\pandas\core\frame.py in __setitem__(self, key, value)
   3117         else:
   3118             # set column
-> 3119             self._set_item(key, value)
   3120
   3121     def _setitem_slice(self, key, value):


~\Anaconda3\lib\site-packages\pandas\core\frame.py in _set_item(self, key, value)
   3192
   3193         self._ensure_valid_index(value)
-> 3194         value = self._sanitize_column(key, value)
   3195         NDFrame._set_item(self, key, value)
   3196


~\Anaconda3\lib\site-packages\pandas\core\frame.py in _sanitize_column(self, key, valu
   3389
   3390             # turn me into an ndarray
-> 3391             value = _sanitize_index(value, self.index, copy=False)
   3392             if not isinstance(value, (np.ndarray, Index)):
   3393                 if isinstance(value, list) and len(value) > 0:


~\Anaconda3\lib\site-packages\pandas\core\series.py in _sanitize_index(data, index, co
   3999
   4000     if len(data) != len(index):
-> 4001         raise ValueError('Length of values does not match length of ' 'index')
   4002
   4003     if isinstance(data, ABCIndexClass) and not copy:


ValueError: Length of values does not match length of index
```

```
In [0]: clf = Pipeline(steps=[('classifier', SVC(class_weight='balanced',C=0.01, kernel='linea
        clf.fit(x_train,y_train)

        # {'C': 0.01, 'kernel': 'linear'}
        proba = cross_val_predict(clf, x_train,y_train, cv=5, method='predict_proba')
        from sklearn import metrics

        fpr, tpr, _ = metrics.roc_curve(y_train,  proba[::,1])
        auc = metrics.roc_auc_score(y_train, proba[::,1])
        plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
        plt.legend(loc=4)
        plt.title('Model ROC curve')
        plt.show()
```



The ROC curve shows promising results, and an AUC of 0.824 was obtained. To give context, a model that randomly guessed the class (50-50 chance) would give a straight line ROC curve with an AUC of 0.5. The model performs significantly better. This is close to the AUC score for the author's crowdsourced data.

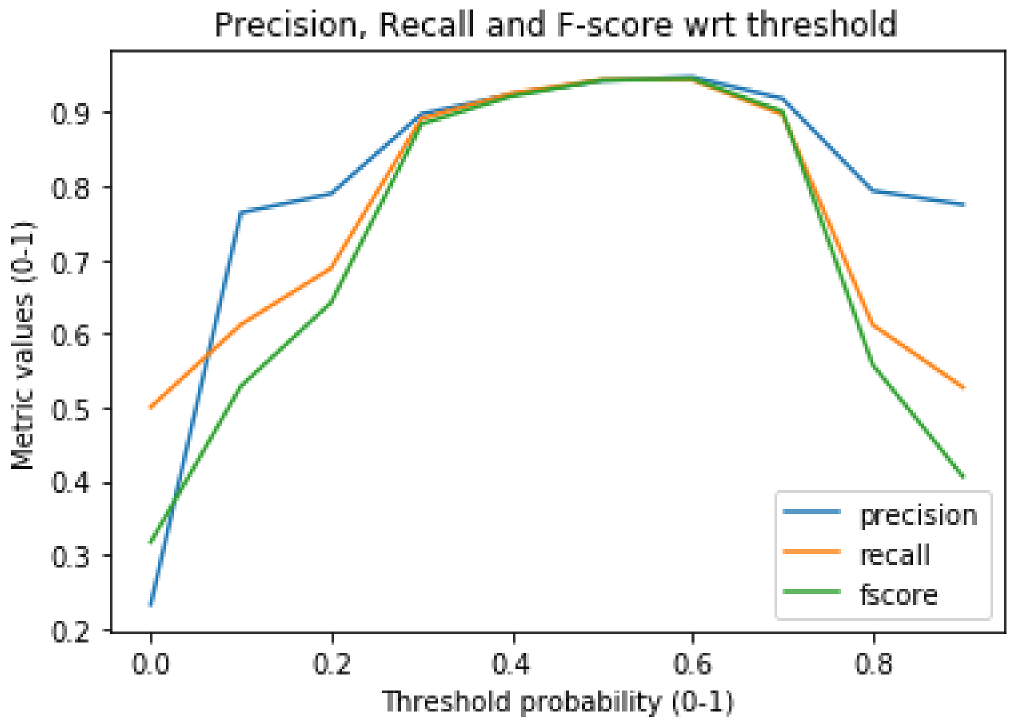
We will now investigate the these metrics (precision, recall and f1-scores) at the various threshold values ranging from 0 to 1.

```
In [0]: from sklearn.metrics import precision_recall_fscore_support as score

        thresh_range = list(np.arange(0,1,0.1))
        p_list = list()
        r_list = list()
```

```
f_list = list()
for threshold in thresh_range:
    y_preds = np.where(model.predict_proba(x_train)[:,1] > threshold, 1, 0)
    precision,recall,fscore,support=score(y_train,y_preds,average='macro')
    p_list.append(precision)
    r_list.append(recall)
    f_list.append(fscore)
plt.plot(thresh_range,p_list,label='precision')
plt.plot(thresh_range,r_list,label='recall')
plt.plot(thresh_range,f_list,label='fscore')
plt.xlabel('Threshold probability (0-1)')
plt.ylabel('Metric values (0-1)')
plt.title('Precision, Recall and F-score wrt threshold')
plt.legend()
```

```
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:1135: Und
  'precision', 'predicted', average, warn_for)
```

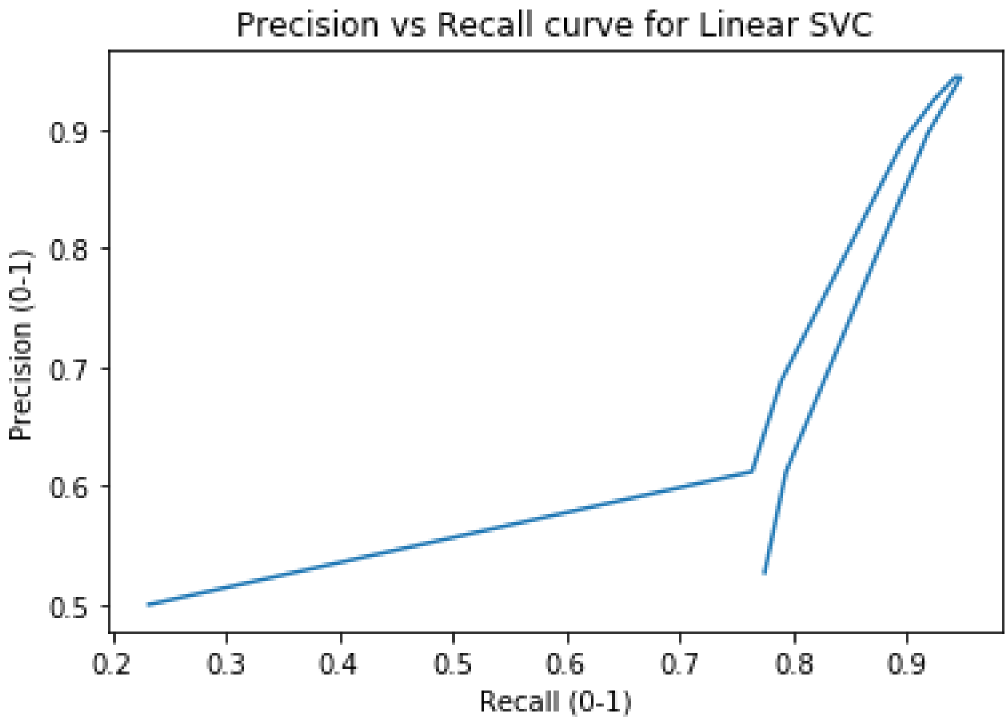
```
Out[0]: <matplotlib.legend.Legend at 0x20139c00320>
```



The above curves demonstrate the best threshold values for our metrics. For this model we obtain an optimum threshold in the (0.3,0.7) range as seen from the above plot. The precision and recall tend to follow very similar trends. Would be interesting to see the precision vs recall curve.

In [0]:
```
plt.plot(p_list,r_list)
plt.title('Precision vs Recall curve for Linear SVC')
plt.xlabel('Recall (0-1)')
plt.ylabel('Precision (0-1)')
```

Out[0]:
```
Text(0,0.5,'Precision (0-1)')
```



Interesting many-one, non-bijective curve. Both metrics peak at the same time, and at a little less than 1.

This concludes the In-sample performance evaluation. Now we will use a test dataset that isn't in-sample and see the difference in results.

1.6.2 Out-of-sample Predictive Performance

Here we use new unseen data to test our model. We expect a decline in performance, but this will also give us a peak into how generalizable the proposed Linear SVC model is.

In [0]:
```
# model2 = SVC(kernel='linear',class_weight='balanced',C=0.01,probability=True).fit(x_
# model2=SVC(class_weight='balanced',C=0.01, kernel='rbf',probability=True).fit(x_trai
y_preds = model.predict(x_test)
report = classification_report( y_test, y_preds )
```

In [0]:
```
print(report)
```

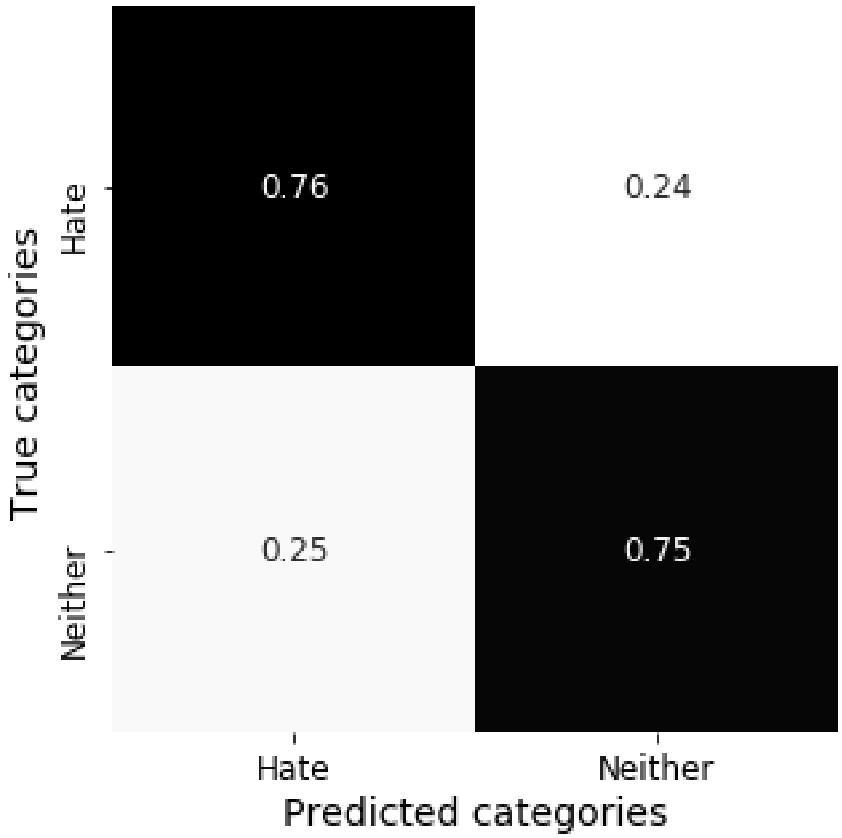
```
             precision    recall  f1-score   support
```

```
             0       0.81      0.76      0.79       255
             1       0.70      0.75      0.72       185

avg / total          0.76      0.76      0.76       440
```

The precision scores reduces by about ~10% compared to the original data and while recall reduces from 0.90 t0 0.76. This was expected, and the ~10% reduction in the scores on unseen data shows that the model is generalizable and wasn't overfitting the training data.

Let us now plot the roc curve for the out-sample case and compare it to our previous in-sample performance. One would expect a minor decline in AUC, which would be in line with the marginal performance metric decline we have seen above.

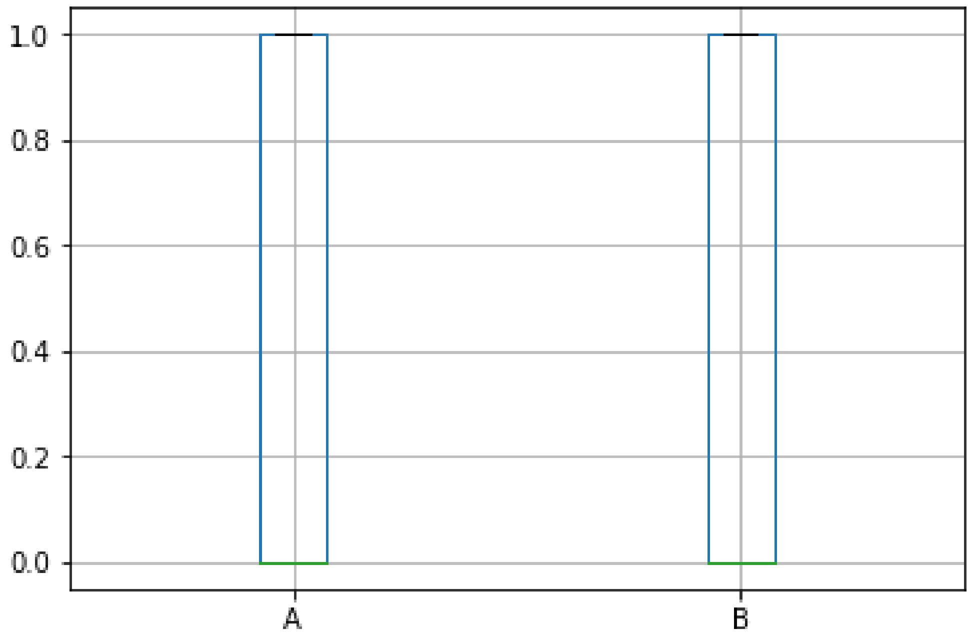
```
In [0]: from sklearn.metrics import confusion_matrix
        import seaborn
        confusion_matrix = confusion_matrix(y_test,y_preds)
        matrix_proportions = np.zeros((2,2))
        for i in range(0,2):
            matrix_proportions[i,:] = confusion_matrix[i,:]/float(confusion_matrix[i,:].sum())
        names=['Hate','Neither']
        confusion_df = pd.DataFrame(matrix_proportions, index=names,columns=names)
        plt.figure(figsize=(5,5))
        seaborn.heatmap(confusion_df,annot=True,annot_kws={"size": 12},cmap='gist_gray_r',cbar=
        plt.ylabel(r'True categories',fontsize=14)
        plt.xlabel(r'Predicted categories',fontsize=14)
        plt.tick_params(labelsize=12)
```

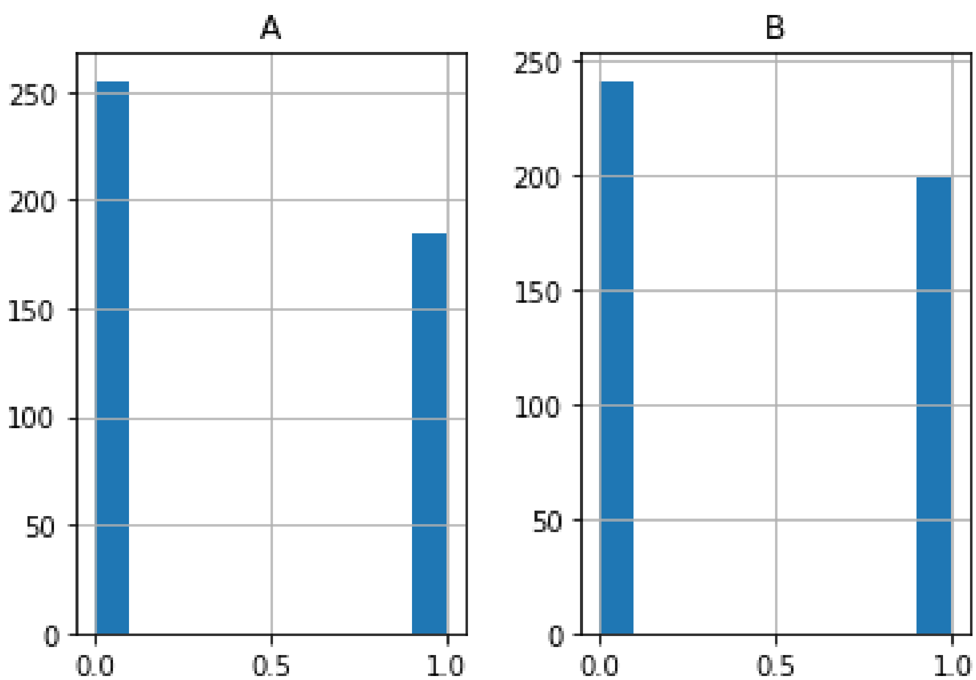



```
In [0]: from pandas import read_csv
        from matplotlib import pyplot
        import pandas as pd
        # load results file
        results = pd.DataFrame()
        results['A'] = y_test
        results['B'] = y_preds
        # descriptive stats
        print(results.describe())
        # box and whisker plot
        results.boxplot()
        pyplot.show()
        # histogram
        results.hist()
        pyplot.show()
```

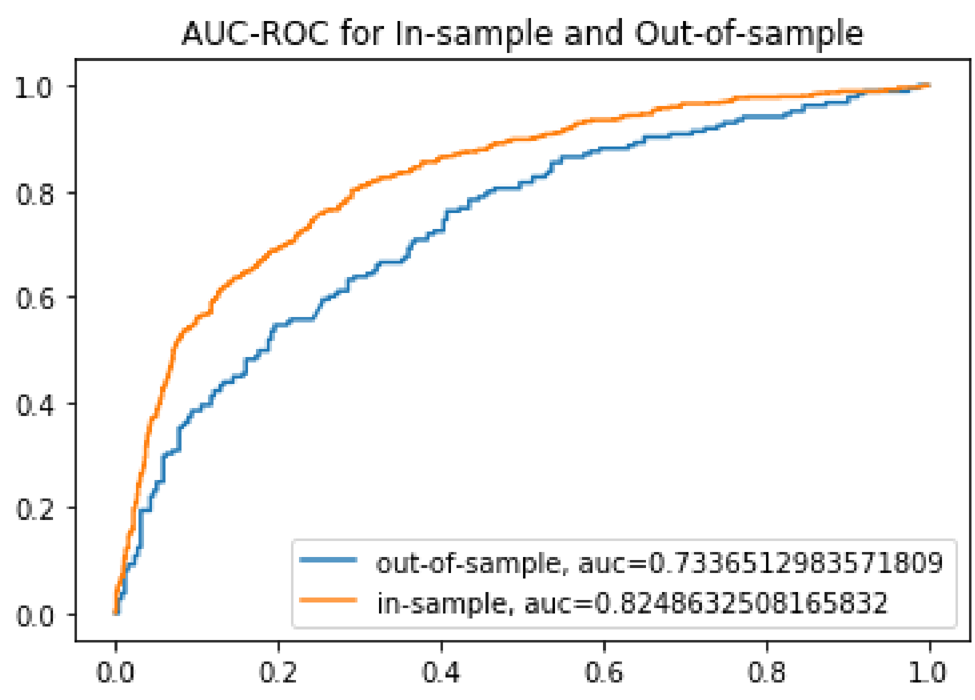
```
                A           B
count  440.000000  440.000000
mean     0.420455    0.452273
std      0.494194    0.498283
```

```
min       0.000000    0.000000
25%       0.000000    0.000000
50%       0.000000    0.000000
75%       1.000000    1.000000
max       1.000000    1.000000
```





```
In [0]: from sklearn.pipeline import Pipeline
        from sklearn.model_selection import cross_val_predict
        import matplotlib.pyplot as plt

        clf = Pipeline(steps=[('classifier', SVC(class_weight='balanced',C=0.01, kernel='linea
        clf.fit(x_train,y_train)

        proba = cross_val_predict(clf, x_test,y_test, cv=5, method='predict_proba')
        probb = cross_val_predict(clf, x_train,y_train, cv=5, method='predict_proba')
        from sklearn import metrics

        fpra, tpra, _ = metrics.roc_curve(y_test,  proba[::,1])
        auca = metrics.roc_auc_score(y_test, proba[::,1])
        fprb, tprb, _ = metrics.roc_curve(y_train,  probb[::,1])
        aucb = metrics.roc_auc_score(y_train, probb[::,1])

        plt.plot(fpra,tpra,label="out-of-sample, auc="+str(auca))
        plt.plot(fprb,tprb,label="in-sample, auc="+str(aucb))

        plt.legend(loc=4)
        plt.title('AUC-ROC for In-sample and Out-of-sample')
        plt.show()
```

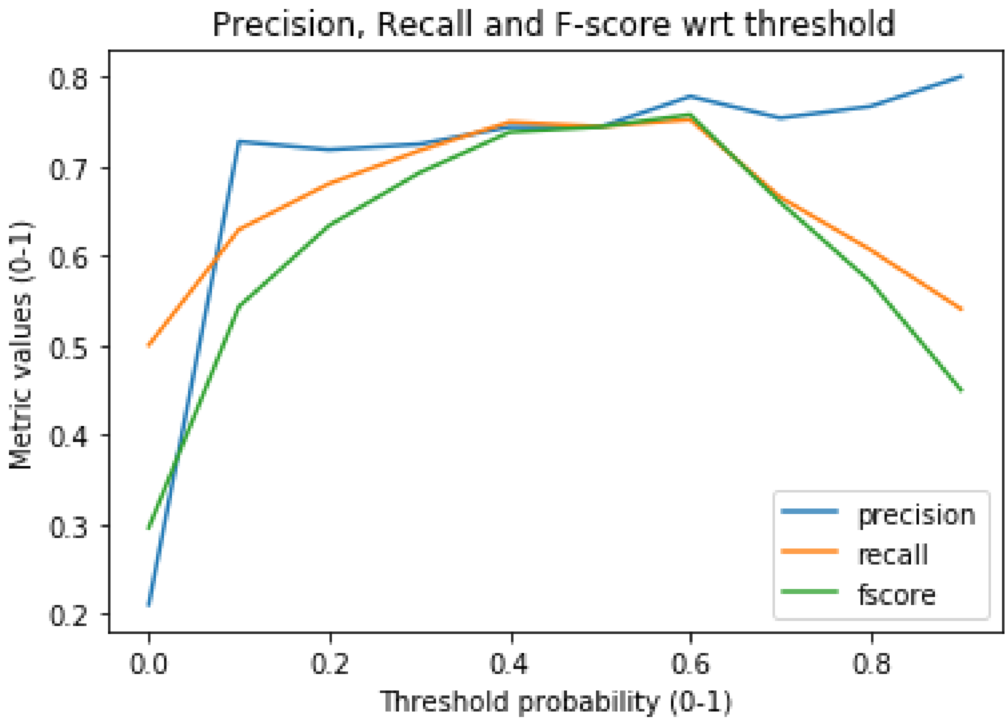


The AUC falls from 0.824 for in-sample to 0.73 for out-of-sample. The difference is clear from the ROC curves.

In [0]:
```
from sklearn.metrics import precision_recall_fscore_support as score

thresh_range = list(np.arange(0,1,0.1))
p_list = list()
r_list = list()
f_list = list()
for threshold in thresh_range:
    y_preds = np.where(model.predict_proba(x_test)[:,1] > threshold, 1, 0)
    precision,recall,fscore,support=score(y_test,y_preds,average='macro')
    p_list.append(precision)
    r_list.append(recall)
    f_list.append(fscore)
plt.plot(thresh_range,p_list,label='precision')
plt.plot(thresh_range,r_list,label='recall')
plt.plot(thresh_range,f_list,label='fscore')

plt.xlabel('Threshold probability (0-1)')
plt.ylabel('Metric values (0-1)')
plt.title('Precision, Recall and F-score wrt threshold')
plt.legend()
```

```
C:\Users\chandana priya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:1135: Unc
  'precision', 'predicted', average, warn_for)
```

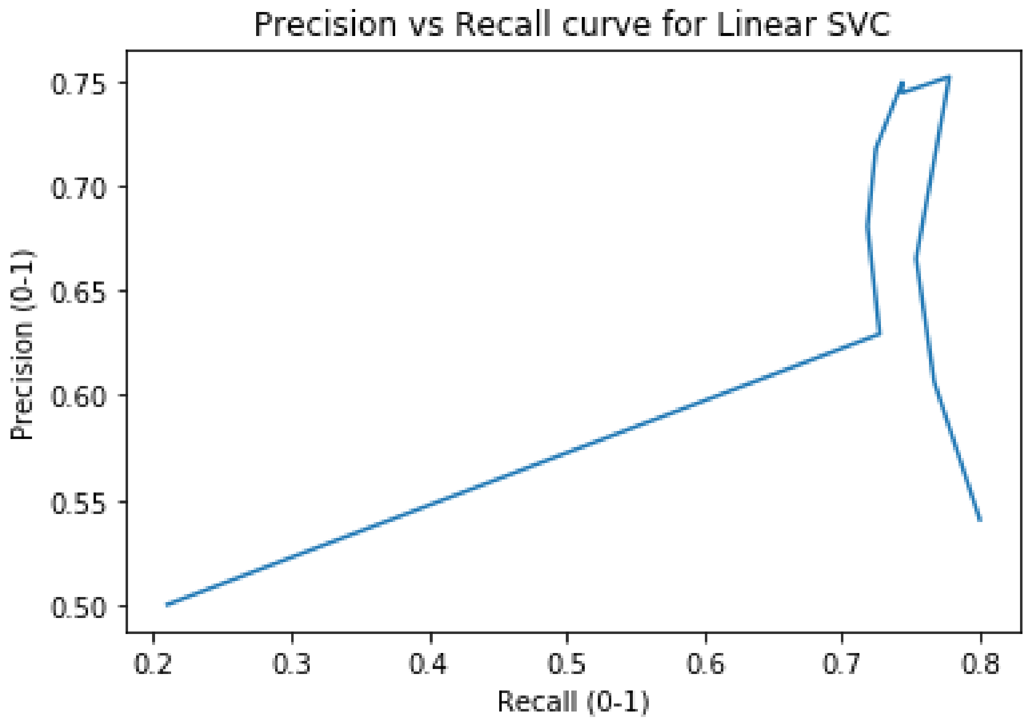
Out[0]:
```
<matplotlib.legend.Legend at 0x2013c1a2710>
```



We see a similar plot as in-sample, though the maxima seems to have shifted to a lower threshold.

The precision and recall tend to follow very similar trends. Would be interesting to see the precision vs recall curve.

```
In [0]: plt.plot(p_list,r_list)
        plt.title('Precision vs Recall curve for Linear SVC')
        plt.xlabel('Recall (0-1)')
        plt.ylabel('Precision (0-1)')

Out[0]: Text(0,0.5,'Precision (0-1)')
```



As before, many-one, non-bijective curve. Both metrics peak at around the same time, with two marked differences from the corresponding in-sample plot. - The maximum observed for out-sample is of lower value than in-sample.

- While they do peak around the same time, the spread is more (i.e both don't peak at exactly the same time as was the case with in-sample, rather the peak shows a greater spread as seen above).

This concludes the out-of-sample performance evaluation. Now we will use a test dataset that isn't in-sample and see the difference in results.

1.6.3 Effects of Statistical Significance on Predictive Power

The following section investigates the effect of statistical significance of a variable on its prediction power. On second thought, 'Effect' might not be the appropriate term here, since that implies causation. Let us investigate the correlation of statistical significance of features with their predictive powers.

This is achieved by implementing a 'backward elimination' function, that assumes all features are significant and the eliminates those that are found to have p-values higher than 5% Level of significance.

An issue encountered was the painfully slow runtime, so we take a short-cut here. Let us select 500 features at random (of the 2303 total features). Empricial evidence during the course of this project suggests that we get back less than 10% of the features we created on this dataset.

We apply the backward elimination function to obtained a reduced feature set. The the model constructed on the 500 features, and another one constructed on the stat-significant feature subset obtained are evaluated.

```
In [0]: import statsmodels.formula.api as sm
        def backwardElimination(x, Y, sl,columns):
            numVars = len(x[0])
            for i in range(0, numVars):
                regressor_OLS = sm.OLS(Y, x).fit()
                maxVar = max(regressor_OLS.pvalues)
                if maxVar > sl:
                    for j in range(0, numVars - i):
                        if (regressor_OLS.pvalues[j].astype(float) == maxVar):
                            x = np.delete(x, j, 1)
                            columns = np.delete(columns, j)

            regressor_OLS.summary()
            return x,columns
        SL = 0.05
```

```
In [0]: from random import sample

        # Prints list of random items of given length
        tot_features = list(range(2033))
        subset = sample(tot_features,500)
        dm,col = backwardElimination(x_train[:,subset], y_train, SL,np.arange(500))
```

```
In [0]: print('The number of statistically significant features from the 500 : ',len(col))
```

```
The number of statistically significant features from the 500 :  54
```

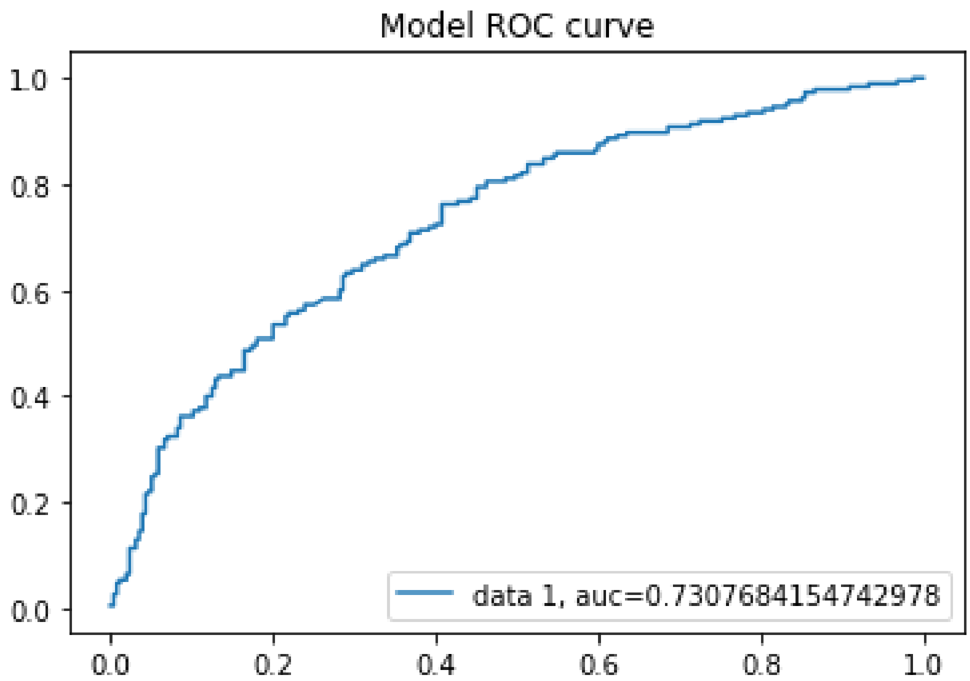
```
In [0]: model = SVC(kernel='linear',class_weight='balanced',C=0.01,probability=True).fit(x_tra

        y_preds = model.predict(x_test)
        report = classification_report( y_test, y_preds )
```

```
In [0]: print(report)
```

```
             precision    recall  f1-score   support

          0       0.81      0.76      0.79       255
          1       0.70      0.75      0.72       185

avg / total       0.76      0.76      0.76       440
```

```
In [0]: from sklearn.pipeline import Pipeline
        from sklearn.model_selection import cross_val_predict
        import matplotlib.pyplot as plt

        clf = Pipeline(steps=[('classifier', SVC(class_weight='balanced',C=0.01, kernel='linea
```

```
clf.fit(x_train,y_train)

proba = cross_val_predict(clf, x_test,y_test, cv=5, method='predict_proba')
from sklearn import metrics

fpr, tpr, _ = metrics.roc_curve(y_test,  proba[::,1])
auc = metrics.roc_auc_score(y_test, proba[::,1])
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.title('Model ROC curve')
plt.show()
```



The above analysis has been performed for the data with 500 randomly selected features.

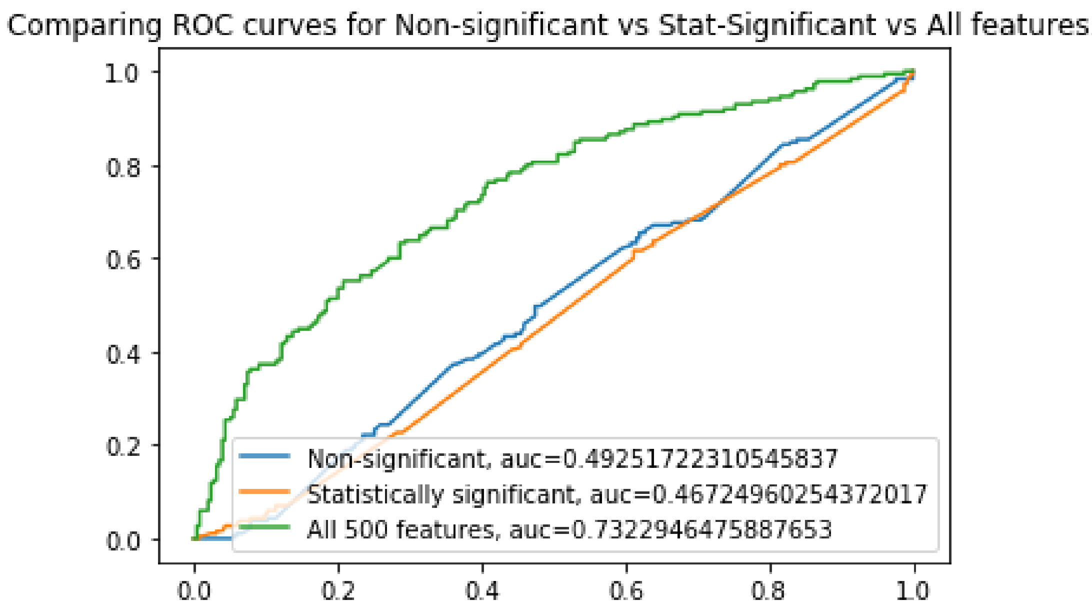
We can see that it gives us a reasonable 0.73 AUC value and the precision,recall and f1 are 0.76.

It would now be interesting to compare this to the results for a model that takes just the statistically significant features from these 500. We saw that 54 features qualified with p-values less than the 5% level of significance.

```
In [0]: model = SVC(kernel='linear',class_weight='balanced',C=0.01,probability=True).fit(x_tra

        y_preds = model.predict(x_test[:,col])
        report = classification_report( y_test, y_preds )

In [0]: print(report)

              precision    recall  f1-score   support
```

```
          0       0.70      0.12      0.21       255
          1       0.43      0.93      0.59       185

avg / total       0.59      0.46      0.37       440
```

In [0]:
```
from sklearn.pipeline import Pipeline
from sklearn.model_selection import cross_val_predict
import matplotlib.pyplot as plt

clf1 = Pipeline(steps=[('classifier', SVC(class_weight='balanced',C=0.01, kernel='linea
clf1.fit(x_train[:,~col],y_train)

clf2 = Pipeline(steps=[('classifier', SVC(class_weight='balanced',C=0.01, kernel='linea
clf2.fit(x_train[:,col],y_train)

clf3 = Pipeline(steps=[('classifier', SVC(class_weight='balanced',C=0.01, kernel='linea
clf3.fit(x_train,y_train)



proba = cross_val_predict(clf1, x_test[:,~col],y_test, cv=5, method='predict_proba')
probb = cross_val_predict(clf2, x_test[:,col],y_test, cv=5, method='predict_proba')
probc = cross_val_predict(clf3, x_test,y_test, cv=5, method='predict_proba')

from sklearn import metrics

fpra, tpra, _ = metrics.roc_curve(y_test,  proba[::,1])
auca = metrics.roc_auc_score(y_test, proba[::,1])

fprb, tprb, _ = metrics.roc_curve(y_test,  probb[::,1])
aucb = metrics.roc_auc_score(y_test, probb[::,1])

fprc, tprc, _ = metrics.roc_curve(y_test,  probc[::,1])
aucc = metrics.roc_auc_score(y_test, probc[::,1])

plt.plot(fpra,tpra,label="Non-significant, auc="+str(auca))
plt.plot(fprb,tprb,label="Statistically significant, auc="+str(aucb))
plt.plot(fprc,tprc,label="All 500 features, auc="+str(aucc))

plt.legend(loc=4)
plt.title('Comparing ROC curves for Non-significant vs Stat-Significant vs All feature
plt.show()
```



This is an interesting result, the performance for just the statistically significant features (54 in number) is 0.46, which is lesser than the 0.45 AUC result for the non-significant features (446 in number). Infact random guessing would give better results than the model trained on just the non-significant features or just the significant ones. Many statistically significant features trump the predictive power of a few significant ones.

The result taking all 500 features is the best performer by a margin. Model using the significant features alone is clearly overfitting.