Assignment 4: Classification

This assignment needs assign4_train.csv and assign4_test.csv. assign4_train.csv is for training and assign4_test.csv is for test. Both of them have samples in the following format:

label	text
1	I must admit that I'm addicted to "Version 2.0...
0	I think it's such a shame that an enormous tal...
1	The Sunsout No Room at The Inn Puzzle has oddl...
...	...

Q1 Classification

Write a function **classify** to conduct a classification experiement as follows:

1. Take the training and testing file names (strings) as inputs, i.e. classify (training_file, testing_file).
2. Classify text samples in the training file using **Linear SVM** as follows:
   - a. First apply grid search with **6-fold cross validation** to find the best values for parameters **min_df**, **stop_words**, and **C** of Linear SVM that are used the modeling pipeline. Use f1-macro as the scoring metric to select the best parameter values. Potential values for these parameters are:
     - min_df' : [1,2,3]
     - stop_words' : [None,"english"]
     - C: [0.5,1,5]
   - b. Using the best parameter values, **train a Linear SVM with all samples** in the training file
3. Test the classifier created in Step 2.b using the test file. Report the testing performance as:
   - Precision, recall, and f1-score of each label
   - Treat label 1 as the positive class, plot precision-recall curve and ROC curve, and calculate AUC, Average Precision (sklearn.metrics.average_precision_score). Note, LinearSVC does not ouput probability, but you can use **decision_function** to calculate AUC (https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC.decision_function (https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC.decision_function))
4. Your function "classify" has no return. However, when this function is called, the best parameter values from grid search is printed and the testing performance from Step 3 is printed.

Q2. How to determine K in k-fold cross validation?

This question will use assign4_train.csv dataset. Use this experiment to find the best **k** for k-fold cross validation.

Write a function "K_fold_CV" as follows:

- Take the full file name path string for a dataset as an input, e.g. K_fold_CV(dataset_file).
- Create tf-idf matrix using TfidfVectorizer
- Conduct k-fold cross validation for different k values varying from 2 to 20. For each k, do the following:
  1. train a classifier using **multinomial Naive Bayes** model with k-fold cross validation
  2. train a classifier using **linear support vector machine** model with k-fold cross validation
  3. for each classifier, collect the average AUC across k folds (treat label 1 as the positive class). Hint, for binary classification, you can set "roc_auc" as the value of "metric" parameter of function "cross_validate".
- Plot a line chart to show **the relationship between sample size and AUC**
- Write your analysis in a **separate pdf file (not in code)** on the following:
  - How does k affect model performance on evaluation sets?
  - By varying k, you also change sample size for training. How can the sample size affect model performance?
- There is no return for this function, but the charts should be plotted.

Q3 (Bonus): Ensemble Models by Stacking

An emsemble model combines decisions from multiple models to improve the overall performance. This question askes you to implement an emsemble model by stacking. The details of this technique and sample code can be found at https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/ (https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/).

Define a function **stacking** to achieve the following:

- Take the training and testing file names (strings) as inputs, i.e. stacking(training_file, testing_file).
- Train Naive Bayes and Linear SVM using 6-fold cross validation as two base models
- Following the procedure for stacking, train a decision tree or random forest as the top model using the predictions from the base models
- Test the ensemble model performance using the testing dataset and print out precision, recall and F-1 score.

This function has not return. Note, this ensemble model may not give you a better performance than base models. Just take this chance to learn how to create ensemble models by stacking. This is a very useful technique.

In [162]:
```
import pandas as pd
# add import statements
```

In [179]:
```
# Q1

def classify(train_file, test_file):

    return None
```

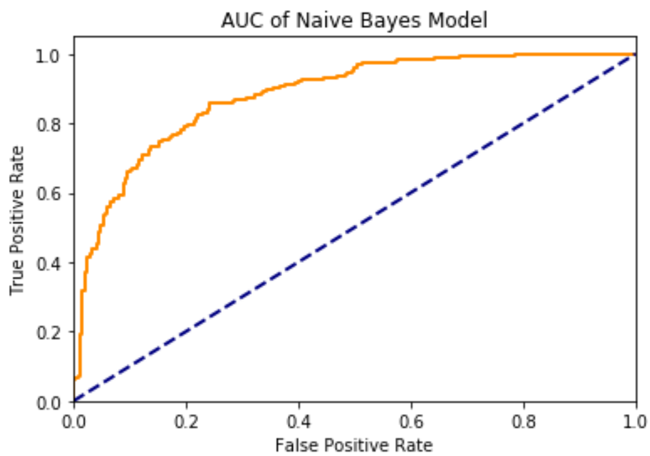
In [180]:
```
# Q2

def K_fold_CV(train_file):

    return None
```

In [186]:

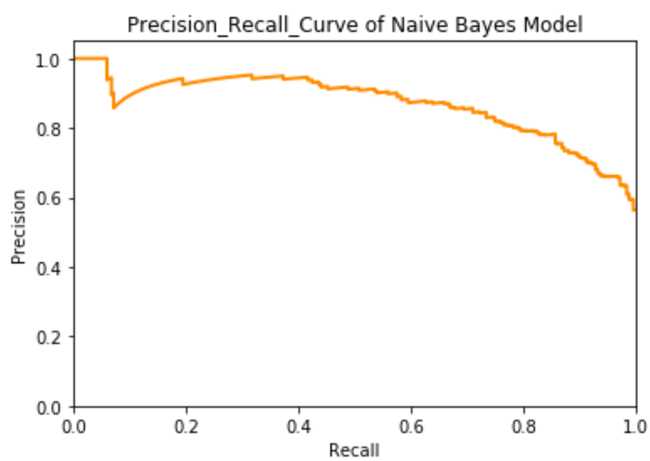
```
def stacking(train_file, test_file):

    return None
```

In [187]:
```
if __name__ == "__main__":
    # Question 1
    print("Q1")
    classify("../../dataset/assign4_train.csv",\
         "../../dataset/assign4_test.csv")

    # Test Q2
    print("\nQ2")
    K_fold_CV("../../dataset/assign4_train.csv")

    # Test Q3
    print("\nQ3")
    stacking("../../dataset/assign4_train.csv",\
         "../../dataset/assign4_test.csv")
```

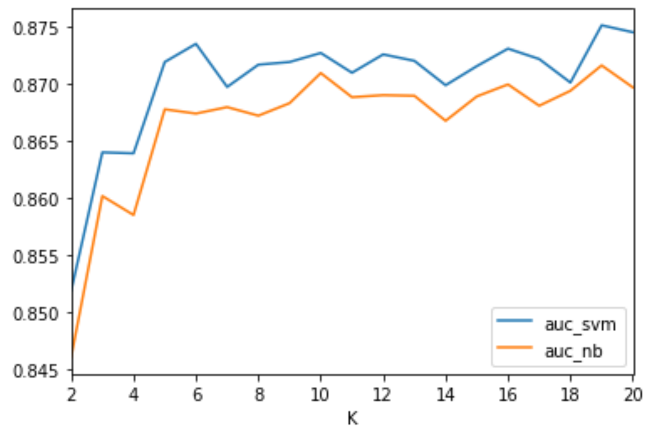
```
Q1
clf__C: 0.5
tfidf__min_df: 1
tfidf__stop_words: None
best f1_macro: 0.8045970021016776
              precision    recall  f1-score   support

           0       0.77      0.83      0.80       248
           1       0.82      0.75      0.78       252

   micro avg       0.79      0.79      0.79       500
   macro avg       0.79      0.79      0.79       500
weighted avg       0.79      0.79      0.79       500

AUC: 0.881
```




```
Average Precision: 0.871
```




```
Q2
```



Q3

	precision	recall	f1-score	support
0	0.79	0.77	0.78	1488
1	0.78	0.80	0.79	1512
micro avg	0.78	0.78	0.78	3000
macro avg	0.78	0.78	0.78	3000
weighted avg	0.78	0.78	0.78	3000

In [ ]: