Movie Review Documentation

**Step #1.**

For the initial processing of the data,

Stop-word list was set

*from sklearn.feature\_extraction.text import CountVectorizer*

*movie\_data = pd.read\_csv('moviereviews\_train.tsv', sep='\t')*

*texts = movie\_data['review']*

*vec = CountVectorizer().fit(texts)*

*bag\_of\_words = vec.transform(texts)*

*sum\_words = bag\_of\_words.sum(axis=0)*

*words\_freq = [(word, sum\_words[0, idx]) for word, idx in vec.vocabulary\_.items()]*

*sorted(words\_freq, key = lambda x: x[1], reverse=True)*

After this, top 10 words were chosen for stop-word selection.

freqwords=sorted(words\_freq, key = lambda x: x[1], reverse=True)

most\_freqwords=freqwords[:10]

most\_freqwords

hv = HashingVectorizer(n\_features=2 \*\* 17, alternate\_sign=False, stop\_words=most\_freqwords)

X\_hv = hv.fit\_transform(movie\_data.review)

print("Shape of HashingVectorizer X:")

print(X\_hv.shape)

**Step#2.**

Ridge Regression / Random Forest showed the best performances for the first test round,

Two models were chosen to be improved mainly.

* **Ridge Regression**

Initial Lambda (Alpha) is 1.0.

To decrease the variance in the prediction, the value was increased up to 10.0

For the test set the prediction value is 0.4962

from sklearn import linear\_model

rdg = linear\_model.RidgeClassifier(alpha=10)

rdg.fit(X\_train, y\_train)

rdg\_performance\_train = BinaryClassificationPerformance(rdg.predict(X\_train), y\_train, 'rdg\_train')

rdg\_performance\_train.compute\_measures()

print(rdg\_performance\_train.performance\_measures)

* **Random Forest**

Default n\_estimator value is 100.

However, for the sake of clearer distinction,

150 is set for this trial.

Max\_features : by reducing the number of features selected, I tried to give more

Varieties among trees randomly generated in bootstrapping way.

Default value is ‘auto’ which is sqrt of feature size.

Now the number of features is 131075. Then, the auto value is roughly around 362.

For this trial, ‘300’ was selected.

For the test set the prediction value is 0.4824

from sklearn.ensemble import RandomForestClassifier

rdf = RandomForestClassifier(n\_estimators=80,max\_depth=2,random\_state=0, max\_features =300)

rdf.fit(X\_train, y\_train)

rdf\_performance\_train = BinaryClassificationPerformance(rdf.predict(X\_train), y\_train, 'rdf\_train')

rdf\_performance\_train.compute\_measures()

print(rdf\_performance\_train.performance\_measures)

Since between the two regression models, ridge regression showed slightly better performance,

This time Ridge Regression was selected to do the first test set.