In [1]:
```
import numpy as np
import pandas as pd
import scipy
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
from sklearn import cross_validation
from sklearn.cross_validation import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

#np.random.seed(36)
```

In [2]:
```
pd.set_option('precision',8)
```

In [ ]:
```
trainData = pd.read_csv("data/feature_train.csv")
testData = pd.read_csv("data/feature_test.csv")
```

In [ ]:
```
trainData
```

In [ ]:
```
X_train = trainData.drop(['Topic'], axis=1)
Y_train = trainData['Topic']
```

In [ ]:
```
X_test = testData.drop(['Topic'], axis=1)
Y_test = testData['Topic']
```

In [ ]:
```
model = MultinomialNB(alpha=0.01)
model.fit(X_train, Y_train)
```

In [ ]:
```
pred = model.predict(X_test)
print "F1 Accuracy = %.4f" %metrics.f1_score(Y_test, pred, average='weighted')
print "Accuracy = %.4f" %metrics.accuracy_score(Y_test, pred)
print "Classification Report: \n" + metrics.classification_report(Y_test, pred)
```

In [ ]:
```
lr_model = LogisticRegression(multi_class='multinomial',solver='newton-cg')
lr_model.fit(X_train, Y_train)
```

In [ ]:
```
pred = lr_model.predict(X_test)
print "F1 Accuracy = %.4f" %metrics.f1_score(Y_test, pred, average='weighted')
print "Accuracy = %.4f" %metrics.accuracy_score(Y_test, pred)
print "Classification Report: \n" + metrics.classification_report(Y_test, pred)
```

In [ ]:
```
knn_model = KNeighborsClassifier()
knn_model.fit(X_train, Y_train)
```

In [ ]:
```
pred = knn_model.predict(X_test)
print "F1 Accuracy = %.4f" %metrics.f1_score(Y_test, pred, average='weighted')
print "Accuracy = %.4f" %metrics.accuracy_score(Y_test, pred)
print "Classification Report: \n" + metrics.classification_report(Y_test, pred)
```

In [ ]:
```

```