sa_binary_training_naive_bayes_m

May 29, 2016

0.1 Load already prepared data

```
In [31]: TEST_SIZE=0.2

In [32]: import pandas as pd

In [33]: X = pd.read_csv('../valt_sa_data/x_m.csv')
         y = pd.read_csv('../valt_sa_data/y_m.csv', header=None)[0]
```

0.2 Split data into training and test sets

Let's perform a train/test split with 80% of the data in the training set and 20% of the data in the test set. We use `random_state=0` so that every execution yields the same result.

```
In [34]: from sklearn.cross_validation import train_test_split

         X_train, X_test, y_train, y_test = train_test_split(X.as_matrix(),
                                                             y.as_matrix(),
                                                             test_size=TEST_SIZE,
                                                             random_state=0)
```

1 Train a sentiment classifier with logistic regression

We will now use logistic regression to create a sentiment classifier on the training data.

Note: This line may take a few minutes.

```
In [35]: from sklearn.naive_bayes import MultinomialNB

         clf = MultinomialNB(alpha=0.1)
         model = clf.fit(X_train, y_train)
```

2 Evaluate the trained model

We will now use the cross-validation set to evaluate our model.

In [36]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, model.predict(X_test))

print 'Confusion matrix:'
print cm

from sklearn.metrics import classification_report

print 'Classification report:'
print classification_report(y_test, model.predict(X_test))
```

```
Confusion matrix:
[[1085  177  155   67  331]
 [ 382  165  245  142  335]
 [ 243  137  449  401  596]
 [ 176   64  318 1166 1888]
 [ 325   87  168  927 9971]]
Classification report:
             precision    recall  f1-score   support

          1       0.49      0.60      0.54      1815
          2       0.26      0.13      0.17      1269
          3       0.34      0.25      0.28      1826
          4       0.43      0.32      0.37      3612
          5       0.76      0.87      0.81     11478

avg / total       0.61      0.64      0.62     20000
```