sa_binary_training_naive_bayes_m

May 29, 2016

0.1 Load already prepared data

```
In [1]: TEST_SIZE=0.2

In [2]: import pandas as pd

In [3]: 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 [4]: 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 [5]: 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 [6]:
```
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:
[[  511     9    10     0  1285]
 [  214    13    20     1  1021]
 [  148     8    20     4  1646]
 [   76     4    21     3  3508]
 [  153     7    25     7 11286]]
Classification report:
             precision    recall  f1-score   support

          1       0.46      0.28      0.35      1815
          2       0.32      0.01      0.02      1269
          3       0.21      0.01      0.02      1826
          4       0.20      0.00      0.00      3612
          5       0.60      0.98      0.75     11478

avg / total       0.46      0.59      0.46     20000
```