# sa_binary_training_log_reg_m

May 29, 2016

### 0.1 Load already prepared data

```
In [1]: TEST_SIZE=0.4
```

```
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 60% of the data in the training set and 40% 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 import linear_model

        #logreg = linear_model.LogisticRegression(C=1e5)
        logreg = linear_model.LogisticRegression(C=1e5, solver='lbfgs', multi_class

        model = logreg.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:
[[ 2047   378   265    74   789]
 [  806   357   430   196   758]
 [  423   325   819   619  1456]
 [  165   162   503  1914  4507]
 [  310   129   258  1088 21222]]
Classification report:
             precision    recall  f1-score   support

          1       0.55      0.58      0.56      3553
          2       0.26      0.14      0.18      2547
          3       0.36      0.22      0.28      3642
          4       0.49      0.26      0.34      7251
          5       0.74      0.92      0.82     23007

avg / total       0.61      0.66      0.62     40000
```