Assignment Precision-Recall

June 14, 2019

In [1]:
```
#For data frame related and matrix related operations
import pandas as pd
import numpy as np
```

In [2]:
```
#For plotting and charting
import matplotlib.pyplot as plt
```

In [3]:
```
import os
```

In [4]:
```
#Machine Learning Algorithms and Evaluation Metrics Library
from sklearn.linear_model import LogisticRegression as LR
from sklearn.metrics import accuracy_score
```

In [5]:
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
```

In [6]:
```
import string
```

In [7]:
```
#Import the Libraries related to ML Algorithms
from sklearn.linear_model import LogisticRegression
```

In [8]:
```
#Import Classification Metrics
from sklearn.metrics import confusion_matrix, roc_auc_score
```

In [9]:
```
from collections import Counter
```

In [10]:
```
baby = pd.read_csv('/home/shrikrishna/Desktop/Coursera Machine Learning/A Case Study
```

In [11]:
```
baby.head()
```

Out[11]:

	name	review	rating
0	Planetwise Flannel Wipes	These flannel wipes are OK, but in my opinion ...	3
1	Planetwise Wipe Pouch	it came early and was not disappointed. i love...	5
2	Annas Dream Full Quilt with 2 Shams	Very soft and comfortable and warmer than it l...	5
3	Stop Pacifier Sucking without tears with Thumb...	This is a product well worth the purchase. I ...	5
4	Stop Pacifier Sucking without tears with Thumb...	All of my kids have cried non-stop when I trie...	5

0.1 Perform Text Cleaning

In [12]:
```
#Removing Punctuation strips punctuation from line of text
def remove_punctuation(text):
    #print(type(text))
    exclude = set(string.punctuation)
    #print(("".join(ch for ch in text if(ch not in exclude))))
    return ("".join(ch for ch in text if(ch not in exclude)))
```

In [13]:
```
#Fill NA
baby.review = baby.fillna({'review': ''})['review']
```

In [14]:
```
#Remove Punctuation from the reviews
baby['review_clean'] = list(map(remove_punctuation, baby['review'].tolist()))
```

0.2 Extract Sentiments

0.2.1 3. We will ignore all reviews with rating = 3, since they tend to have a neutral sentimentű

In [15]:
```
baby = baby.loc[baby['review']!=3, :]
```

0.2.2 4. Label the ratings as positive class(+1) and negative class(-1)

In [16]:
```
baby['sentiment'] = list(map(lambda x:-1 if(x<3) else 1,
                             baby['rating'].tolist()))
```

In [17]:
```
#Check if any sentiment having rating>3 and sentiment = -1 is present
baby.loc[(baby['sentiment']==-1) & (baby['rating']>3), :]
```

Out[17]:
```
Empty DataFrame
Columns: [name, review, rating, review_clean, sentiment]
Index: []
```

In [18]:
```
baby.loc[(baby['sentiment']==-1) & (baby['rating']<3), :].shape
```

Out[18]:
```
(26493, 5)
```

0.3 Read the Training and Test Files

In [19]:
```
os.getcwd()
```

Out[19]:
```
'/home/shrikrishna/Desktop/Coursera Machine Learning/Machine Learning: Classification/
```

In [20]:
```
base_path = '/home/shrikrishna/Desktop/Coursera Machine Learning/Machine Learning: Cla
train_path = base_path + '/train-idx.json/train-idx.json'
test_path = base_path + '/test-idx.json/test-idx.json'
```

In [21]:
```
train_data_index, test_data_index = [pd.read_json(path) for path in
                                     [train_path, test_path]]
train_data = baby.iloc[train_data_index[0], :]
test_data = baby.iloc[test_data_index[0], :]
```

In [22]: train_data.head()

Out[22]:

	name	review	rating	review_clean	sentiment
0	Planetwise Flannel Wipes	These flannel wipes are OK, but in my opinion ...	3	These flannel wipes are OK but in my opinion n...	1
1	Planetwise Wipe Pouch	it came early and was not disappointed. i love...	5	it came early and was not disappointed i love ...	1
2	Annas Dream Full Quilt with 2 Shams	Very soft and comfortable and warmer than it l...	5	Very soft and comfortable and warmer than it l...	1
3	Stop Pacifier Sucking without tears with Thumb...	This is a product well worth the purchase. I ...	5	This is a product well worth the purchase I h...	1
4	Stop Pacifier Sucking without tears with Thumb...	All of my kids have cried non-stop when I trie...	5	All of my kids have cried nonstop when I tried...	1

In [23]: test_data.head()

Out[23]:

	name	review	rating	review_clean	sentiment
8	Baby Tracker® - Daily Childcare Journal, S...	A friend of mine pinned this product on Pinter...	5	A friend of mine pinned this product on Pinter...	1
9	Baby Tracker® - Daily Childcare Journal, S...	This has been an easy way for my nanny to reco...	4	This has been an easy way for my nanny to reco...	1
14	Nature\'s Lullabies First Year Sticker Calendar	Space for monthly photos, info and a lot of us...	5	Space for monthly photos info and a lot of use...	1
18	Nature\'s Lullabies Second Year Sticker Calendar	I completed a calendar for my son\'s first yea...	4	I completed a calendar for my sons first year ...	1
24	Nature\'s Lullabies Second Year Sticker Calendar	Wife loves this calender. Comes with a lot of ...	5	Wife loves this calender Comes with a lot of s...	1

0.4 Build the word count vector for each review

```
In [24]: vectorizer = CountVectorizer(token_pattern=r'\b\w+\b')
         # Use this token pattern to keep single-letter words
         # First, learn vocabulary from the training data and assign columns to words

         # Then convert the training data into a sparse matrix
         train_matrix = vectorizer.fit_transform(train_data['review_clean'])

         # Second, convert the test data into a sparse matrix, using the same word-column mapp
         test_matrix = vectorizer.transform(test_data['review_clean'])
```

0.5 7. Learn a logistic regression classifier using the training data.

```
In [25]: model = LR()

         #Fit the model on training data with sentiment as the target
         model.fit(train_matrix, train_data['sentiment'])

Out[25]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                   verbose=0, warm_start=False)
```

0.6 Model Evaluation

```
In [26]: accuracy_model_test = accuracy_score(test_data['sentiment'],
                                               model.predict(test_matrix))
         print("Accuracy Test Data:- %.3f" %(accuracy_model_test))

Accuracy Test Data:- 0.913


In [27]: prediction_test, prediction_train = [model.predict(data) for data in
                                               [test_matrix, train_matrix]]
```

0.7 Baseline: Majority class prediction

Typically, a good model should beat the majority class classifier. Since the majority class in this dataset is the positive class (i.e., there are more positive reviews than negative reviews), the accuracy of the majority class classifier is simply the fraction of positive reviews in the test set:

```
In [28]: baseline = len(test_data[test_data['sentiment'] == 1])/len(test_data)
         print ("Baseline accuracy (majority class classifier): %s" % baseline)

Baseline accuracy (majority class classifier): 0.8561915046796257
```

0.7.1 Quiz question: Using accuracy as the evaluation metric, was our logistic regression model better than the baseline (majority class classifier)?

In [29]:
```
ans = "Yes"
print(ans)
```

```
Yes
```

In [30]:
```
actual_vs_predicted_test = pd.DataFrame({'Actual':test_data['sentiment'],
                                         'Predicted': prediction_test},
                                         columns = ['Actual', 'Predicted'])
actual_vs_predicted_test.head()
```

Out[30]:

	Actual	Predicted
8	1	1
9	1	1
14	1	1
18	1	1
24	1	1

0.8 Confusion Matrix

In [31]:
```
#Confusion Matrix for test data

CM = confusion_matrix(test_data['sentiment'], prediction_test)
CM
```

Out[31]:
```
array([[ 2934,  1860],
       [ 1031, 27511]])
```

In [32]:
```
model.classes_
```

Out[32]:
```
array([-1,  1])
```

In [33]:
```
print("Actual Label\tPredicted Label\t Count")
print("-------------------------------------------")

for i in range(len(model.classes_)):
    for j in range(len(model.classes_)):
        print('{0:^13} | {1:^15} | {2:5d}'.format(
            model.classes_[i], model.classes_[j], CM[i,j]))
```

```
Actual Label        Predicted Label         Count
-------------------------------------------
     -1       |       -1        |  2934
     -1       |        1        |  1860
      1       |       -1        |  1031
      1       |        1        | 27511
```

```
In [34]: help(confusion_matrix)

Help on function confusion_matrix in module sklearn.metrics.classification:

confusion_matrix(y_true, y_pred, labels=None, sample_weight=None)
    Compute confusion matrix to evaluate the accuracy of a classification

    By definition a confusion matrix :math:`C` is such that :math:`C_{i, j}`
    is equal to the number of observations known to be in group :math:`i` but
    predicted to be in group :math:`j`.

    Thus in binary classification, the count of true negatives is
    :math:`C_{0,0}`, false negatives is :math:`C_{1,0}`, true positives is
    :math:`C_{1,1}` and false positives is :math:`C_{0,1}`.

    Read more in the :ref:`User Guide <confusion_matrix>`.

    Parameters
    ----------
    y_true : array, shape = [n_samples]
        Ground truth (correct) target values.

    y_pred : array, shape = [n_samples]
        Estimated targets as returned by a classifier.

    labels : array, shape = [n_classes], optional
        List of labels to index the matrix. This may be used to reorder
        or select a subset of labels.
        If none is given, those that appear at least once
        in ``y_true`` or ``y_pred`` are used in sorted order.

    sample_weight : array-like of shape = [n_samples], optional
        Sample weights.

    Returns
    -------
    C : array, shape = [n_classes, n_classes]
        Confusion matrix

    References
    ----------
    .. [1] `Wikipedia entry for the Confusion matrix
           <https://en.wikipedia.org/wiki/Confusion_matrix>`_

    Examples
    --------
    >>> from sklearn.metrics import confusion_matrix
    >>> y_true = [2, 0, 2, 2, 0, 1]
```