Pattern Recognition and Machine LearningAssignment 3

Course Instructor : Arun Rajkumar.

Release Date : October 2, 2022

Submission Date:On or before 11:59 PM on November 17, 2022

SCORING: There are 2 questions in this assignment. The contribution of points scored in this assignment towards your final grades will be 10 points

The points will be decided based on the clarity and rigour of the report provided andthe correctness of the code submitted.

DATASETS The data-sets are in the corresponding google drive folder shared in Moodle.

WHAT SHOULD YOU SUBMIT? You should submit a zip file titled 'Solutions _ rollnumber.zip' where rollnumber is your institute roll number. Your assignment will NOTbe graded if it does not contain all of the following:

- A text file titled 'Details.txt' with your name and roll number.
- A PDF file which includes explanations regarding each of the solution as required inthe question. Title this file as 'Report.pdf'
- Clearly named source code for all the programs that you write for the assignment .

CODE LIBRARY: You are expected to code all algorithms from scratch. You cannot use standard inbuilt libraries for **algorithms** taught in class. You are free to use inbuilt libraries for plots. You can code using either Python or Matlab or C.

GUIDELINES: Keep the below points in mind before submission.

- Plagiarism of any kind is unacceptable. These include copying text or code from anyonline sources. These will lead to disciplinary actions according to institute guidelines.
- Any graph that you plot is unacceptable for grading unless it labels the x-axis andy-axis clearly.
- Don't be vague in your explanations. The clearer your answer is, the more chance itwill be scored higher.

LATE SUBMISSION POLICY You are expected to submit your assignment on or before the deadline to avoid any penalty. Late submission incurs a penalty in points equal to the number of days your submission is late by. Any late submission post three days of the deadline would not be graded and will fetch 0 points.

SPAM or HAM?

In this assignment, you will build a spam classifer from scratch. No training data will be provided. You are free to use whatever training data that is publicly available/does not have any copyright restrictions (You can build your own training data as well if you think that is useful). You are free to extract features as you think will be appropriate for this problem. The final code you submit should have a function/procedure which when invoked will be able to automatically read a set a emails from a folder titled test in the current directory. Each _le in this folder will be a test email and will be named 'email#.txt' ('email1.txt', 'email2.txt', etc). For each of these emails, the classifer should predict +1 (spam) or 0 (non Spam). You are free to use whichever algorithm learnt in the course to build a classifer (or even use more than one). The algorithms (except SVM) need to be coded from scratch. Your report should clearly detail information relating to the data-set chosen, the features extracted and the exact algorithm/procedure used for training including hyperparameter tuning/kernel selection if any. The performance of the algorithm will be based on the accuracy on the test set.

Ans. I've implemented spam classifier using two algorithms that is **Logistic Regression** and **Support Vector Machine.**
The image given below is dataset that is used:

```
df=pd.read_csv('spam.csv', encoding='latin-1')
df.head()
```

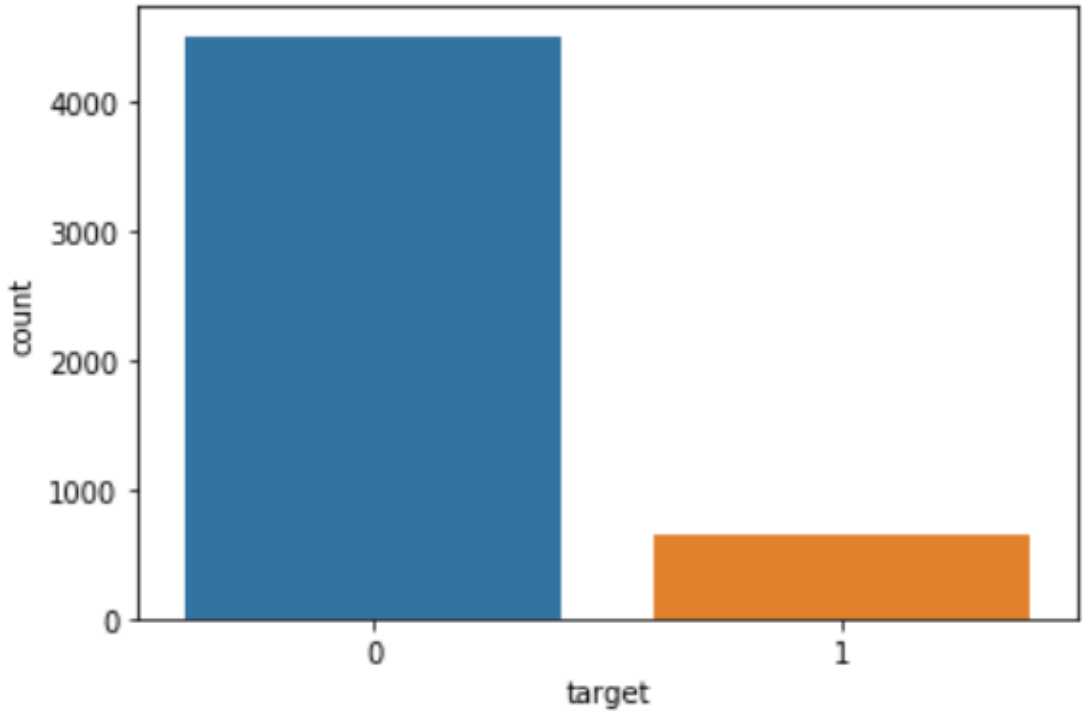
	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

It has 5572 rows containing a collection of spam and ham emails both. As we can see that it contains some empty columns and also column names are not that much descriptive so we apply some of data cleaning to it at first. After removing duplicate rows and **assigning ham=0 and spam=1** our dataset looks like:

Out[23]:

	target	emails
0	0	go jurong point avail bugi n great world la e ...
1	0	ok lar joke wif u oni
2	1	free entri 2 wkli comp win fa cup final tkt 21...
3	0	u dun say earli hor u c alreadi say
4	0	nah think goe usf live around though

To see how much spam or ham data it contains we visualize it and got to know that our **data is imbalanced.**



After that I've applied Data preprocessing step which includes converting entire data into lowercase, tokenizing it, removing stop words and after that stemming ,which is done using transformed_text function.

```
def transform_text(text):
    #lower case
    text=text.lower()

    #tokenization

    text=nltk.word_tokenize(text)#text is converted to a list
    #removing special char
    y=[]
    for i in text:
        if i.isalnum():
            y.append(i)
    #removing stop words
    text=y[:]
    y.clear()
    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)
    text=y[:]
    y.clear()
    for i in text:
        y.append(ps.stem(i))

    return " ".join(y)
```

After this step our data looks like this:

Out[10]:

	target	emails
0	0	go jurong point avail bugi n great world la e ...
1	0	ok lar joke wif u oni
2	1	free entri 2 wkli comp win fa cup final tkt 21...
3	0	u dun say earli hor u c alreadi say
4	0	nah think goe usf live around though

Then using CountVectorizer I've vectorize the data that is collecting each word and its frequency in each email. Now this matrix can be used for train test split and further model building.

Logistic Regression Implementation:

The logistic regression is built upon a logistic function which is sigmoid function. The logistic function gives a value between 0 and 1 for the input variables, which enables the algorithm to decide if its spam or not. The expression of sigmoid function is:

$$\sigma(x) = 1/(1+\exp(-\theta^T X))$$

where $\theta^T X$ represents line that separates two classes that is spam and ham and θ is weight vector that we need to find.

Like in linear regression, I find the optimal value of weight vector using Gradient descent update step:

$$\Theta^{t+1} = \theta^t + lr * (x^T [\, h_\theta(x) - y])$$

h_θ is our sigmoid function.
Here, learning rate that I've taken is 0.5 and I'm running my gradient descent for 4000 epochs to get a optimal value of weight vector.
At each step , I've calculated error using cost function:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

$$J(\theta) = \frac{1}{m} [\sum_{i=1}^{m} -y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))]$$

$$m = number\ of\ samples$$

```
def gradient(X,y,theta):
    '''
    return grad_vect-(n+1,1)
    '''
    hi=hypothesis(X,theta)
    grad= -np.dot(X.T,(y-hi))
    m=X.shape[0]
    return grad/m
def gradient_descent(X,y,lr=0.5,max_itr=4000):

    n=X.shape[1]
    theta= np.zeros((n,1))
    #print(theta.shape)
    error_list=[]
    for i in range(max_itr):
        err=error(X,y,theta)
        error_list.append(err)

        grad=gradient(X,y,theta)

        #update
        theta=theta - (lr*grad)

    return theta,error_list
```

The error vs iteration graph is given as follows:

Now our model is build so I test it on the remaining 20% data. The results that I've got is as follows:

For 20% test data:

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      0.98      0.99       909
           1       0.89      0.98      0.94       125

    accuracy                           0.98      1034
   macro avg       0.94      0.98      0.96      1034
weighted avg       0.98      0.98      0.98      1034

Accuracy 0.9835589941972921
Precision 0.8913043478260869
Recall 0.984
F1 Score 0.9353612167300379
```

Here I'm getting 98% accuracy.

For Sample text file:

(From test folder)

```
file = os.listdir('test/')
test_data = []

for mail in file:
    with open('test/'+mail) as infile:
        content = infile.read()
        text = rtf_to_text(content)
    test_data.append(text)
# Testing On Test Data
test_data1= [transform_text(x) for x in test_data]
test_data1 = cv.transform(test_data).toarray()
new_test=np.hstack((np.ones((test_data1.shape[0],1)),test_data1))
pred = predict(new_test,theta)
for i in range(len(pred)):
    if pred[i] == 1:
        res = "spam\n"
    else:
        res = 'ham\n'
    print(res,'  ------ ',test_data[i])
```

```
spam
   ------  Dear Beneficiary,The United Nations Compensation Commission (UNCC) has approved to pay you a compensation amount of US$1,500,000 (One Million,
ham
   ------  Dear Sir,PRML quiz was the first exam we have taken among all the other courses here at IIT Madras. We were not really sure about what kind of
```

It is classifying **first email as spam** and the **second one as ham**.

SVM Implementation:

Here I've done SVM using sklearn library. The kernel here I've used is rbf (radial basis kernel) as it is giving the best possible accuracy than the other kernels. The objective of SVM is to select a hyperplane with the maximum possible margin between support vectors in the given dataset.
I've done this to compare my implemented algorithm with SVM , which is done as follows:

For 20% test data:

SVM Implementation

```
[40] from sklearn.svm import SVC
     classifier = SVC(kernel ='rbf', random_state = 0)
     classifier.fit(X_train, Y_train)

     print("Accuracy",classifier.score(X_test,Y_test))
```

42s

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/va
  y = column_or_1d(y, warn=True)
Accuracy 0.9709864603481625
```

Here the accuracy I'm getting is 97% which is less than the logistic regression implementation that I've done.

For Sample text file:
(From test folder)

```
# Testing On Test Data
test_data1= [transform_text(x) for x in test_data]
test_data1 = cv.transform(test_data).toarray()
Y1=[1,0]
pred = classifier.predict(test_data1)
for i in range(len(pred)):
    if pred[i] == 1:
        res = "spam"
    else:
        res = 'ham'
    print(res,'  ------ ',test_data[i])
```

```
ham   ------  Dear Beneficiary,The United Nations Compensation Commission (UNCC) has approved to pay you a compensation amount of US$1,500,000 (One Millio
ham   ------  Dear Sir,PRML quiz was the first exam we have taken among all the other courses here at IIT Madras. We were not really sure about what kind
```

It is classifying both the data as ham while my model is classifying the first mail as spam email.