NLP LAB – 9

Anveshak Rathore
181060012
Final Year B.Tech Electronics

AIM: To design and implement an algorithm for computing precision ,recall and accuracy on Text data.

SOFTWARE: Python 3.8.6, Visual Studio Code

THEORY:

Classi cation models are used in classi cation problems to predict the target class of the data sample. The classi cation model predicts the probability that each instance belongs to one class or another. It is important to evaluate the performance of the classi cations model in order to reliably use these models in production for solving real-world problems. Performance measures in machine learning classi cation models are used to assess how well machine learning classi cation models perform in a given context. These performance metrics include **accuracy**, **precision** and **recall**.

Prediction Score --

The model precision score measures the proportion of positively predicted labels that are actually correct. Precision is also known as the positive predictive value. Precision is mainly used when we need to predict the positive class and there is a greater cost associated with false positives than with false negatives such as in medical diagnosis or spam ltering. It is used in conjunction with the recall to trade-o false positives and false negatives. Precision is a ected by the class distribution. If there are more samples in the minority class, then precision will be lower. Precision can be thought of as a measure of exactness or quality.

Precision Score = TP / (FP + TP)

Recall Score --

Model recall score represents the model's ability to correctly predict the positives out of actual positives. This is unlike precision which measures how many predictions made by models are actually positive out of all positive predictions made.In other words, it measures how good our machine learning model is at identifying all actual positives out of all positives that exist within a dataset. The higher the recall score, the better the machine learning model is at identifying both positive and negative examples. Recall is also known as sensitivity or the true positive rate. A high recall score indicates that the model is good at identifying positive examples.

Recall Score = TP / (FN + TP)

Accuracy Score --

Model accuracy is a machine learning model performance metric that is de ned as the ratio of true positives and true negatives to all positive and negative observations. In other words, accuracy tells us how often we can expect our machine learning model will correctly predict an outcome out of the total number of times it made predictions. Mathematically, it represents the ratio of the sum of true positives and true negatives out of all the predictions.

Accuracy Score = (TP + TN)/ (TP + FN + TN + FP)

PROGRAM and OUTPUT:

```python
import nltk
import numpy as np
import pandas as pd
import random
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
import pprint, time
# nltk.download('treebank')
# nltk.download('universal_tagset')
```

✓ 1m 3.3s Python

```python
nltk_data = list(nltk.corpus.treebank.tagged_sents(tagset='universal'))
train_set,test_set = train_test_split(nltk_data,train_size=0.90,test_size=0.01,random_state = 101)
```

✓ 7.8s Python

```python
#printing a random sentence
random_sentence = [random.randint(1,len(test_set)) for x in range(5)]
test_run = [test_set[i] for i in random_sentence]
test_run[0]
```

✓ 0.1s Python

```
[('And', 'CONJ'),
 ('an', 'DET'),
 ('FT-SE', 'NOUN'),
 ('futures', 'NOUN'),
 ('contract', 'NOUN'),
 ('is', 'VERB'),
 ('traded', 'VERB'),
 ('*-1', 'X'),
 ('on', 'ADP'),
 ('the', 'DET'),
 ('London', 'NOUN'),
 ('International', 'NOUN'),
 ('Financial', 'NOUN'),
 ('Futures', 'NOUN'),
 ('Exchange', 'NOUN'),
 ('.', '.')]
```

```python
def find_word(w):
    counts={}
    for i in train_set:
        for j in i:
            if j[0].lower()==w.lower():
                if j[1] not in counts.keys():
                    counts[j[1]]=1
                else:
                    counts[j[1]]+=1
    max_c=-1
    ans=''
    for i in counts.keys():
        if counts[i]>max_c:
            max_c=counts[i]
            ans=i
    return ans
```

✓ 0.7s Python

```python
for i in range(1):  #len(test_run)
    correct_answer=[]
    predicted_answer=[]
    print("Sentence is:")
    print(test_run[i])
    for j in test_run[i]:
        correct_answer.append(j[1])
        predicted_answer.append(find_word(j[0]))
    print("Correct answer:")
    print(correct_answer)
    print("Predicted answer: ")
    print(predicted_answer)
    print("Confusion Matrix")
    print(confusion_matrix(correct_answer,predicted_answer ))
    print(classification_report(correct_answer, predicted_answer, digits=3))
    print("The accuracy is {}".format(accuracy_score(correct_answer, predicted_answer)))
```

✓ 1.2s Python

```
Output exceeds the size limit. Open the full output data in a text editor
Sentence is:
[('And', 'CONJ'), ('an', 'DET'), ('FT-SE', 'NOUN'), ('futures', 'NOUN'), ('contract', 'NOUN'), ('is', 'VERB'), ('traded', 'VERB'), ('*-1', 'X'),
('on', 'ADP'), ('the', 'DET'), ('London', 'NOUN'), ('International', 'NOUN'), ('Financial', 'NOUN'), ('Futures', 'NOUN'), ('Exchange', 'NOUN'), ('.',
'.')]
Correct answer:
['CONJ', 'DET', 'NOUN', 'NOUN', 'NOUN', 'VERB', 'VERB', 'X', 'ADP', 'DET', 'NOUN', 'NOUN', 'NOUN', 'NOUN', 'NOUN', '.']
```

```
['CONJ', 'DET', 'NOUN', 'NOUN', 'NOUN', 'VERB', 'VERB', 'X', 'ADP', 'DET', 'NOUN', 'NOUN', 'ADJ', 'NOUN', 'NOUN', '.']
Confusion Matrix
[[1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 1 0 0 0 0 0]
 [0 0 0 1 0 0 0 0]
 [0 0 0 0 2 0 0 0]
 [0 1 0 0 0 7 0 0]
 [0 0 0 0 0 0 2 0]
 [0 0 0 0 0 0 0 1]]
              precision    recall  f1-score   support

           .      1.000     1.000     1.000         1
         ADJ      0.000     0.000     0.000         0
         ADP      1.000     1.000     1.000         1
        CONJ      1.000     1.000     1.000         1
         DET      1.000     1.000     1.000         2
        NOUN      1.000     0.875     0.933         8
        VERB      1.000     1.000     1.000         2
           X      1.000     1.000     1.000         1
   ...
   macro avg      0.875     0.859     0.867        16
weighted avg      1.000     0.938     0.967        16

The accuracy is 0.9375
```

CONCLUSION:

We have thus successfully obtained the precision, recall and accuracy scores for a random sentence in the corpus. The confusion matrix presents the state of true positives, true negatives, false positives and false negatives. These parameters prove to be useful when assessing the quality of our model.