Task_3_solution

September 11, 2021

0.0.1 Task 3

Assignment task

0.1 1. Build a feature Matrix using TF-IDF method

0.0.1 1a. Preprocessing

1a.1 Reading the first dataset

[ ]:
```
import numpy as np
import math
import pandas as pd

df_A = pd.read_csv('../../datasets/Tweets.csv')

df_A
```

[ ]:
```
           tweet_id airline_sentiment  \
0      5.703060e+17           neutral
1      5.703010e+17          positive
2      5.703010e+17           neutral
3      5.703010e+17          negative
4      5.703010e+17          negative
…               …                …
14635  5.695880e+17          positive
14636  5.695870e+17          negative
14637  5.695870e+17           neutral
14638  5.695870e+17          negative
14639  5.695870e+17           neutral

                                                    text     tweet_created
0                    @VirginAmerica What @dhepburn said.  24-02-2015 11:35
1      @VirginAmerica plus you've added commercials t…  24-02-2015 11:15
2      @VirginAmerica I didn't today… Must mean I n…  24-02-2015 11:15
3      @VirginAmerica it's really aggressive to blast…  24-02-2015 11:15
4      @VirginAmerica and it's a really big bad thing…  24-02-2015 11:14
…                                                    …                …
14635  @AmericanAir thank you we got on a different f…  22-02-2015 12:01
14636  @AmericanAir leaving over 20 minutes Late Flig…  22-02-2015 11:59
```

```
14637  @AmericanAir Please bring American Airlines to…  22-02-2015 11:59
14638  @AmericanAir you have my money, you change my …  22-02-2015 11:59
14639  @AmericanAir we have 8 ppl so we need 2 know h…  22-02-2015 11:58

[14640 rows x 4 columns]
```

1a.2. Using Regex for preprocessing the text

[ ]:
```
df_A['text'].replace(regex=True, inplace=True, to_replace=r'[^A-Za-z0-9 ]+',␣
↪value=r'')
df_A['text'].replace(regex=True, inplace=True, to_replace=r'\d+\s*', value=r'')
df_A["text"] = df_A["text"].apply(lambda x: x.lower())

df_A['text']
```

[ ]:
```
0                          virginamerica what dhepburn said
1         virginamerica plus youve added commercials to …
2         virginamerica i didnt today must mean i need t…
3         virginamerica its really aggressive to blast o…
4         virginamerica and its a really big bad thing a…
                               …
14635     americanair thank you we got on a different fl…
14636     americanair leaving over minutes late flight n…
14637     americanair please bring american airlines to …
14638     americanair you have my money you change my fl…
14639     americanair we have ppl so we need know how ma…
Name: text, Length: 14640, dtype: object
```

1a.3 Generating the TF-IDF feature matrix

[ ]:
```
processed_reviews = df_A['text']

unique_words=[]
unique_words_count = []

for review in processed_reviews:
    for word in review.split():
        if word not in unique_words:
            unique_words.append(word)
            unique_words_count.append(1)
        else:
            index = unique_words.index(word)
            unique_words_count[index] = unique_words_count[index] + 1

tfidf_feature_matrix = np.zeros((len(processed_reviews),len(unique_words)))

for n,review in enumerate(processed_reviews):
```

```
    for word in review.split():
        index = unique_words.index(word)
        tfidf_feature_matrix[n][index] = review.split().count(word) * math.
↪log(len(processed_reviews) / unique_words_count[index])

tfidf_feature_matrix
```

[ ]:
```
array([[3.33768398, 3.10078925, 9.59151279, …, 0.        , 0.        ,
        0.        ],
       [3.33768398, 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       [3.33768398, 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       …,
       [0.        , 0.        , 0.        , …, 0.        , 0.        ,
        9.59151279],
       [0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.        ]])
```

0.2 2. Fit the Log Regression Classifier and get Accuracy, Precision, Recall and AUC Score

0.2.1 2a. Mapping the airline_sentiment data

[ ]:
```
df = df_A

# Mapping values from -1 to 1
# Limiting values between 0 and 1 for binary logistic regression

df['airline_sentiment'] = df['airline_sentiment'].map({'negative': 0, 'neutral':
↪ 1, 'positive': 1})
df['airline_sentiment'] = df['airline_sentiment'].astype('category')
df.head(10)
```

[ ]:

	tweet_id	airline_sentiment	\
0	5.703060e+17	1	
1	5.703010e+17	1	
2	5.703010e+17	1	
3	5.703010e+17	0	
4	5.703010e+17	0	
5	5.703010e+17	0	
6	5.703010e+17	1	
7	5.703000e+17	1	
8	5.703000e+17	1	
9	5.702950e+17	1	

```
                                                text     tweet_created
0                    virginamerica what dhepburn said  24-02-2015 11:35
1  virginamerica plus youve added commercials to …  24-02-2015 11:15
2  virginamerica i didnt today must mean i need t…  24-02-2015 11:15
3  virginamerica its really aggressive to blast o…  24-02-2015 11:15
4  virginamerica and its a really big bad thing a…  24-02-2015 11:14
5  virginamerica seriously would pay a flight for…  24-02-2015 11:14
6  virginamerica yes nearly every time i fly vx t…  24-02-2015 11:13
7  virginamerica really missed a prime opportunit…  24-02-2015 11:12
8           virginamerica well i didntbut now i do d  24-02-2015 11:11
9  virginamerica it was amazing and arrived an ho…  24-02-2015 10:53
```

0.2.2 2b. Splitting the data into training and test sets and fitting the model

[ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_auc_score,␣
↪average_precision_score

X = tfidf_feature_matrix

y = df_A['airline_sentiment']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,␣
↪random_state=1)

lg = LogisticRegression(C=0.5, solver='liblinear', penalty='l1')
lg.fit(X_train, y_train)
```

[ ]:
```
LogisticRegression(C=0.5, penalty='l1', solver='liblinear')
```

[ ]:
```
lg_y_pred = lg.predict(X_test)

print("Accuracy:", lg.score(X_test, y_test))
print("Average precision-recall score", average_precision_score(y_test,␣
↪lg_y_pred))
print("AUC score:", roc_auc_score(y_test, lg_y_pred))
```

```
Accuracy: 0.8306010928961749
Average precision-recall score 0.6829619580696455
AUC score: 0.8202985034653019
```

0.3 3. Compare the result with Naive Bayes

[ ]:
```
from sklearn.naive_bayes import BernoulliNB

bb = BernoulliNB()
bb.fit(X_train, y_train)
bb_y_pred = bb.predict(X_test)

print("Accuracy:", bb.score(X_test, y_test))
print("Average precision-recall score", average_precision_score(y_test,␣
 ↪bb_y_pred))
print("AUC score:", roc_auc_score(y_test, bb_y_pred))
```

```
Accuracy: 0.833879781420765
Average precision-recall score 0.6892610138445917
AUC score: 0.8171850404558888
```

0.4 4. Observations

- The dataset is imbalanced, as the reviews are largely negative.
- The data was split between two, considering negative reviews as `0`, whereas neutral and positive reviews as `1` for binary classification.
- The training and test data have been split into training and test sets using the `train_test_split` method.(ratio 3:1)
- The model has been fit using Logistic Regression, and the accuracy is reported as `0.830`
- The model has been fit using BernoulliNB, and the accuracy is reported as `0.833`
- The TF-IDF feature matrix performs better than the BoW matrix by approximately 9.8% accuracy.
- BernoulliNB outperforms Logistic Regression in Accuracy as it is better at smaller amounts of data compared to other popular models.