JANUARY 2022 | DATE NO. 11

KAGGLE COMPETITION

pipeline + CountVectorizer+TfidfTransformer+SGDClassifier

Updates inside the campus:

TIPS FOR DATA MINING WORKS

How to improve accuracy

BY 110062466 孫槐駿

First and foremost, I import the data from tweets_DM.json, and make them to dataframe structure. In this structure, I took "emotion" as label, "text" as training data

```
tweets.head()
features = tweets['_source']
features.values
tweet = [_['tweet'] for _ in features.values]
collect_tweet = {_: list() for _ in tweet[0].keys()}
for r in tweet:
    for col_name, value in r.items():
        collect_tweet[col_name].append(value)
data_df = pd.DataFrame.from_dict(collect_tweet).set_index('tweet_id')
```

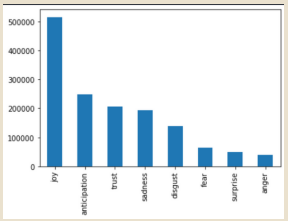
tweet_id	hashtags	text
0x376b20	[Snapchat]	People who post "add me on #Snapchat" must be ...
0x2d5350	[freepress, TrumpLegacy, CNN]	@brianklaas As we see, Trump is dangerous to #...
0x28b412	[bibleverse]	Confident of your obedience, I write to you, k...
0x1cd5b0	[]	Now ISSA is stalking Tasha 😂😂😂 <LH>
0x2de201	[]	"Trust is not the same as faith. A friend is s...
...	...	...

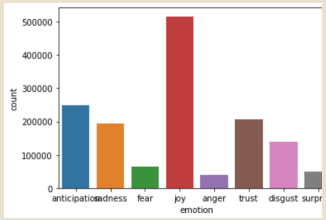
Then after that, I want to use Bag of Words to preprocessing the training data, and take BOW-500 to represent the feature of texts, and use this vector to train model.

However, the result of BOW was not good

0.29558	0.29427

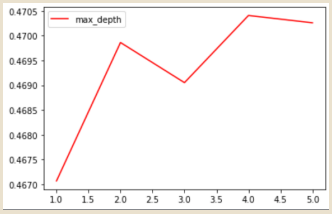
After getting bad feedback. I began to consider why and how to improve the accuracy.
So, I began to analyze the data





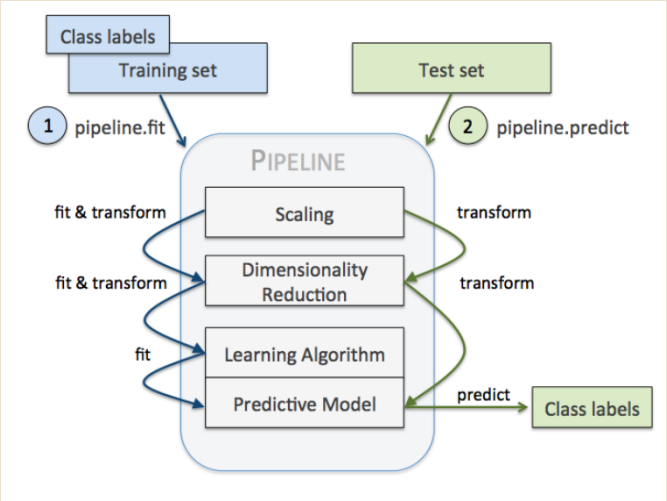
tweet_id	hashtags	text	identification	emotion
0x28b412	[bibleverse]	Confident of your obedience, I write to you, k...	test	anticipation
0x2de201	[]	"Trust is not the same as faith. A friend is s...	test	anticipation
0x218443	[materialism, money, possessions]	When do you have enough ? When are you satisfi...	test	joy
0x2939d5	[GodsPlan, GodsWork]	God woke you up, now chase the day #GodsPlan #...	test	joy
0x26289a	[]	In these tough times, who do YOU turn to as yo...	test	trust
...	...	...	...	...
0x2913b4	[]	"For this is the message that ye heard from th...	test	anticipation
0x2a980e	[]	"There is a lad here, which hath five barley l...	test	anticipation
0x316b80	[mixedfeeling, butimTHATperson]	When you buy the last 2 tickets remaining for ...	test	anticipation
0x29d0cb	[]	I swear all this hard work gone pay off one da...	test	anger
0x2a6a4f	[]	@Parcel2Go no card left when I wasn't in so I ...	test	sadness

Since I analyzed the text, I found out that we had so many emotions. So, I began to consider to use Decision Tree to finish this.
And that was result:



0.38305 0.38374

Well, It may be Better, but not that better.
At this time. I searched in the Internet to find better way to handle this.
And I found the pipeline



And I added CountVectorizer + TfidfTransformer + SGDClassifier in this pipeline

```
pipeline = Pipeline(
    [
        ('vect',CountVectorizer()),
        ('tfidf', TfidfTransformer()),
        ('clf',SGDClassifier()),
    ]
)
parameters = {
    'vect__max_df': (0.75,1.0),
    'vect__ngram_range' : ((1,1),(1,2)),
    'vect__decode_error':('strict','ignore','replace'),
    'vect__lowercase':(True,False),
    'clf__max_iter':(100,),
    'clf__alpha': (0.00001,0.000001),
}
```

The intermediate process of Pipeline is composed of transformers adapted to scikit-learn, and the last step is an estimator. For example, in the above code, CountVectorizer and tfidf transformer form intermediate steps, and SGDClassifier is used as the final estimator.
When we execute pipe_lr.fit(X_train, y_train), the fit and transform methods are first executed on the training set by the CountVectorizer, and the transformed data is passed to the next step of the Pipeline object, which is tfidf. Like CountVectorizer, tfidf also executes the fit and transform methods, and finally passes the transformed data to SGDClassifier.
After that, I just modified the parameters, and I could randomly combine different parameters data in different transformer, I could gradually found the best score.
Finally, I got the following score:

0.47848 0.47811

In conclusion

I've learned how to build Bag of words, decision tree, and pipeline model
Before that, I was wondering why I could train model in sklearn hundreds of and thousands of times?
But in this competition, I found out it was a good way to modify the parameters with the model in pipeline to increase the accuracy