資工三 許雰茹 108321015

NLP with Disaster Tweets

Topic – NLP with Disaster Tweets

- Twitter has become an important communication channel.
- Many people announce an emergency through smartphone.

-> More agencies are interested in programmatically monitoring Twitter. Ex disaster relief organization ,news agencies.

Goal

- To predict which Tweets are about real disasters and which one's aren't. (1 for real ,0 for not a disaster.)
- Submissions are evaluated using F1 between the predicted and expected answers.

Figure Eight

- An open source training data which the industry needs for benchmarking and advancing machine learning deployments.
- The dataset we used in this task is :

Multilingual Disaster Response Messages

A set of messages related to disaster response, covering multiple languages, suitable for text categorization and related natural language processing tasks.

Source:
https://www.prnewswire.com/news-releases/figure-eight-announces-datasets-video-object-tracking-and-smart-bounding-box-annotation-to-accelerate-the-adoption-of-ai-300646558.html

Dataset

資料集	size	key(features)
Train	7613	id, text , location , keyword , target
Test	3263	id, text, location, keyword

columns	
Id	每個tweet的編號
Text	Tweets內容
Location	Tweet發送的地點(不一定每個都有)
Keyword	災難分類(不一定每個都有)
Target	只有train.csv有，為每則tweet的label , 1或0

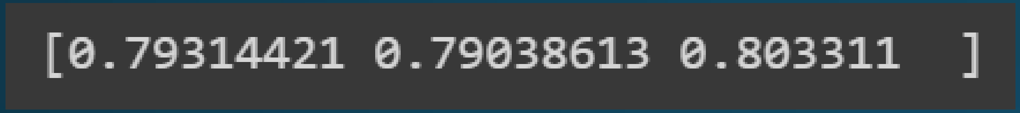
最後test.csv的測試結果存至sample_submission.csv，再下載下來提交給kaggle。

Data Preprocess

- 將train set的資料shuffle後再訓練，以避免資料間的相依性。
- 針對text 使用 CountVectorizer建立字典，把所有的字收入字典，且根據每則推文出現的字產生word vector (length=dictionary size)。

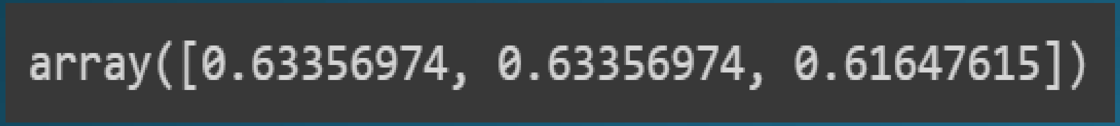
dimensionality reduction test

- 利用pca降低text的維度，配合LogisticRegression(c=0.25)
- pca(n_components = 0.99) scoring=accuracy



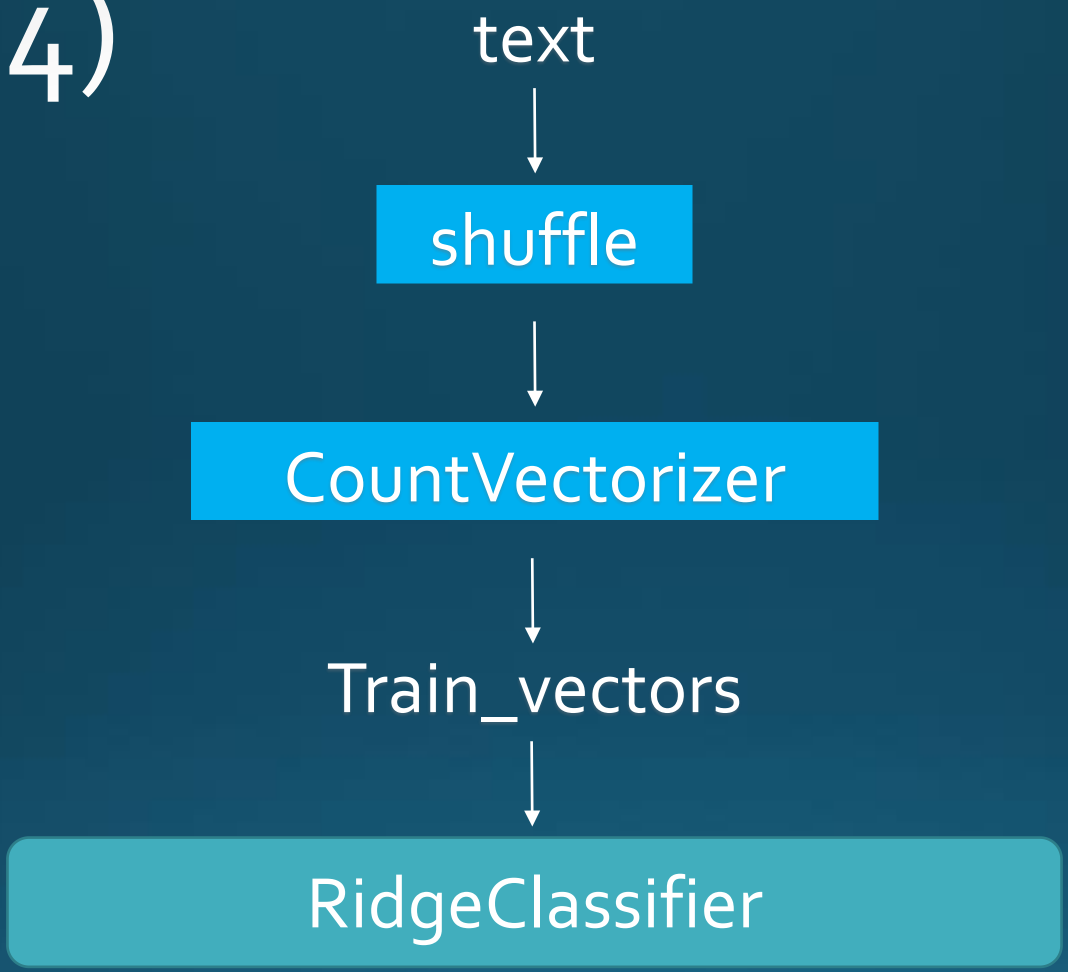
```
[0.79314421 0.79038613 0.803311  ]
```

- Kpca(n_components = 2, kernel = rbf, gamma=0.05) accuracy 約為63%



```
array([0.63356974, 0.63356974, 0.61647615])
```

Model (1/4)



text

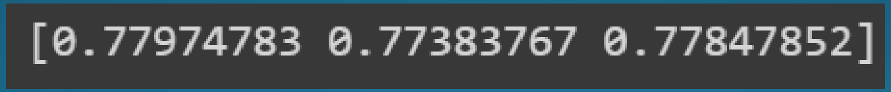
shuffle

CountVectorizer

Train_vectors

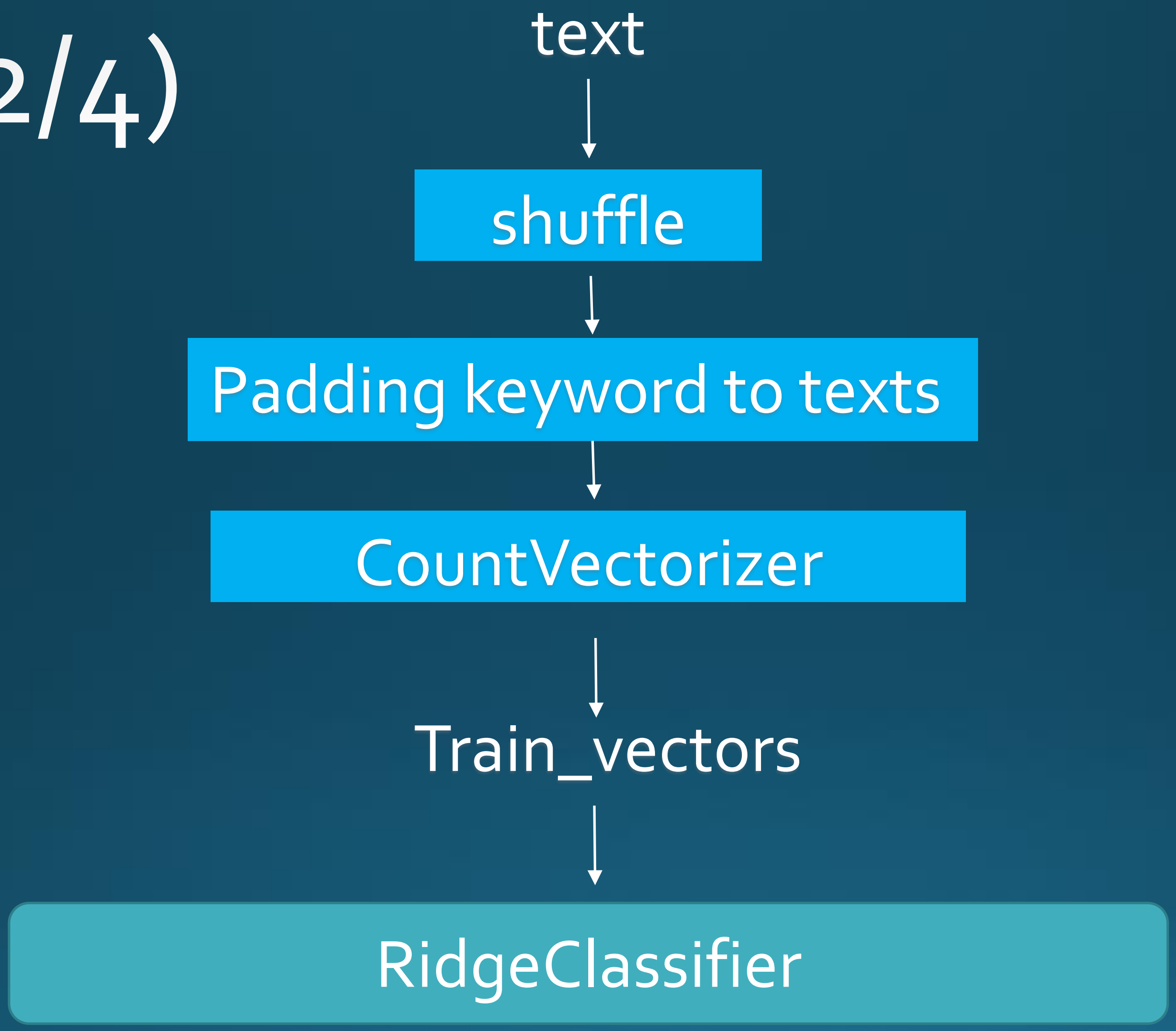
RidgeClassifier

Cv=3 accuracy:



```
[0.77974783 0.77383767 0.77847852]
```

Model(2/4)



text

shuffle

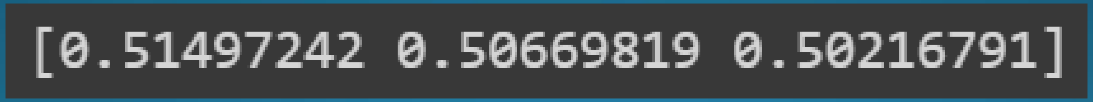
Padding keyword to texts

CountVectorizer

Train_vectors

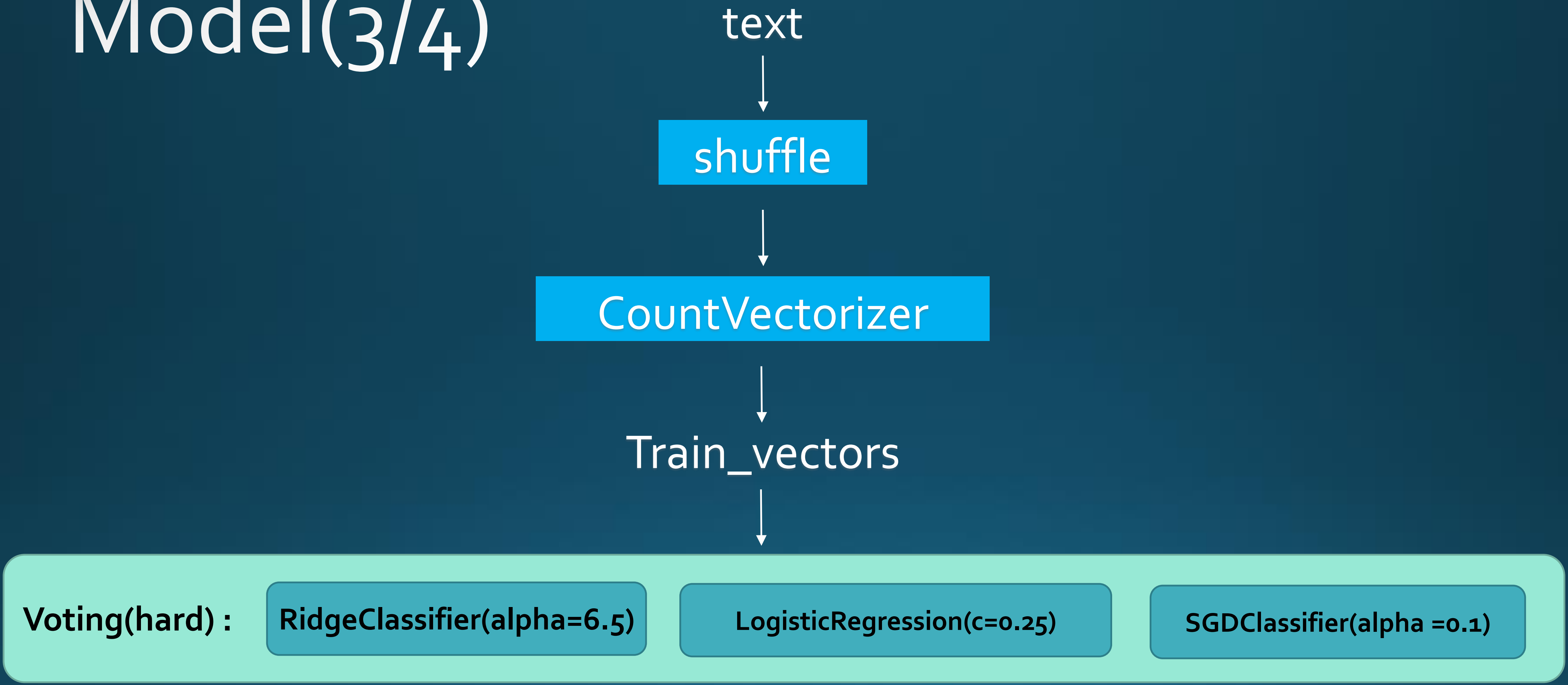
RidgeClassifier

Cv=3 accuracy:



```
[0.51497242 0.50669819 0.50216791]
```

Model(3/4)



text

shuffle

CountVectorizer

Train_vectors

Voting(hard) :

RidgeClassifier(alpha=6.5)

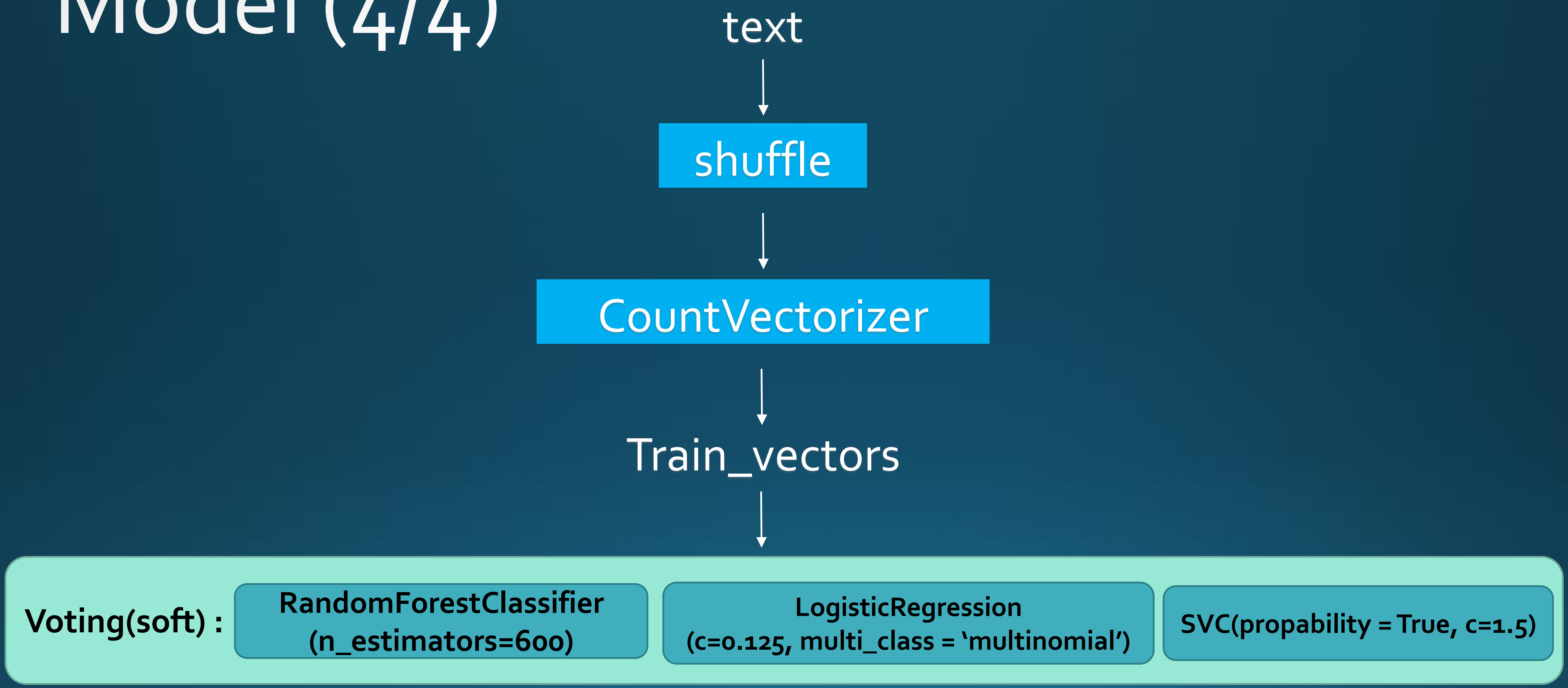
LogisticRegression(c=0.25)

SGDClassifier(alpha =0.1)

Cv=3 accuracy:

```
[0.7998424  0.79235619 0.798581  ]
```

Model (4/4)



text

shuffle

CountVectorizer

Train_vectors

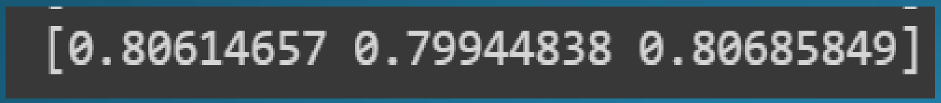
Voting(soft) :

RandomForestClassifier
(n_estimators=600)

LogisticRegression
(c=0.125, multi_class = ‘multinomial’)

SVC(propability = True, c=1.5)

Cv=3 accuracy:



```
[0.80614657 0.79944838 0.80685849]
```

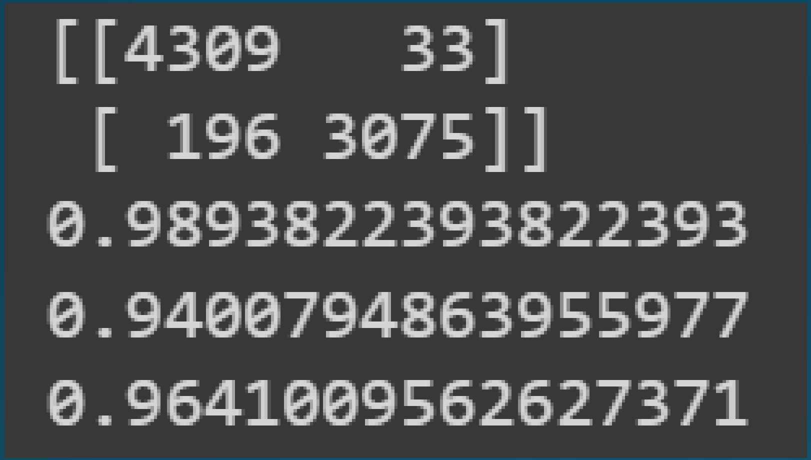
Confusion matrix

Confusion matrix:

Precision:

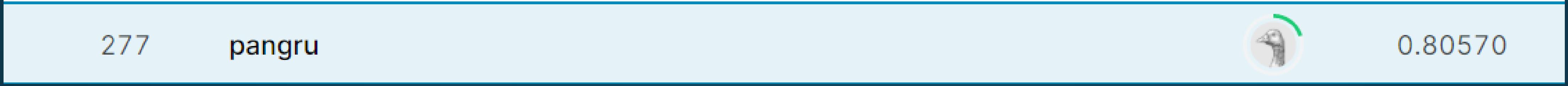
Recall:

F1 score:



```
[[4309   33]
 [ 196 3075]]
0.9893822393822393
0.9400794863955977
0.9641009562627371
```

Ranking and Score



277 pangru 0.80570

Thank you for listening !