**Natural Language Processing with Disaster Tweets**

107321057 黃琇筠 108321015 許雱茹

**Goal:** predict which Tweets are about real disasters(1) and which ones are not(0).

**Dataset**:

使用Kaggle 提供的train.csv , test.csv作為資料集

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | | |
| Keyword | 災難分類 | | |
| 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。

**DataPreprocess:**

1. 將train set的資料shuffle後再訓練，以避免資料間的相依性。
2. 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)。

Ex; train\_vector[0]=[0,0,0,…….,1,0,1] length(train\_vector[0])=dic.length()

**Model and dimensionality reduction:**

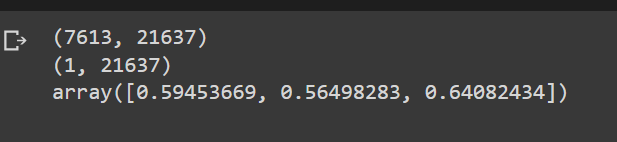
我們把上面產生的word vectors，作為train set丟入classifier訓練

測試的方式均使用cross validation 分三堆（cv =3 accuracy = f1)。

我們使用f1 score 的原因是資料集性質為binary targets。

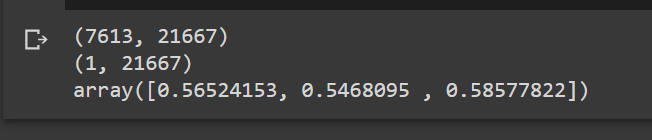
**First test:**

Model : RidgeClassifier with cross validation(cv=3) scoring by f1

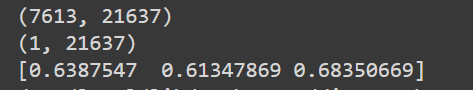


**Second test:**

把每個text句首/句尾加上其keyword之後，效果反而降低6%

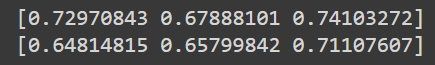


Model : LogisticRegression，f1 score比RidgeClassifier提升4%

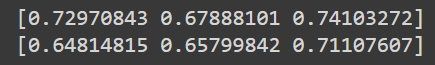


**=>需大量的regularization(因為text維度大，資料量大) :**

Model : RidgeClassifier(alpha=6.5 , default = 1.0)，f1 score 提升10%



Model : SGDClassifier(alpha=0.1 , default = 0.0001)，f1 score 提升7%



**Third test :**

將資料shuffle過後

Model : RidgeClassifier(alpha =6.5)，f1 score 提升6%



Model： SGDClassifier(alpha =0.1) ，f1 score 提升0.5%



**PCA TEST :**

利用**pca**降低text的維度，配合LogisticRegression(c=0.25)

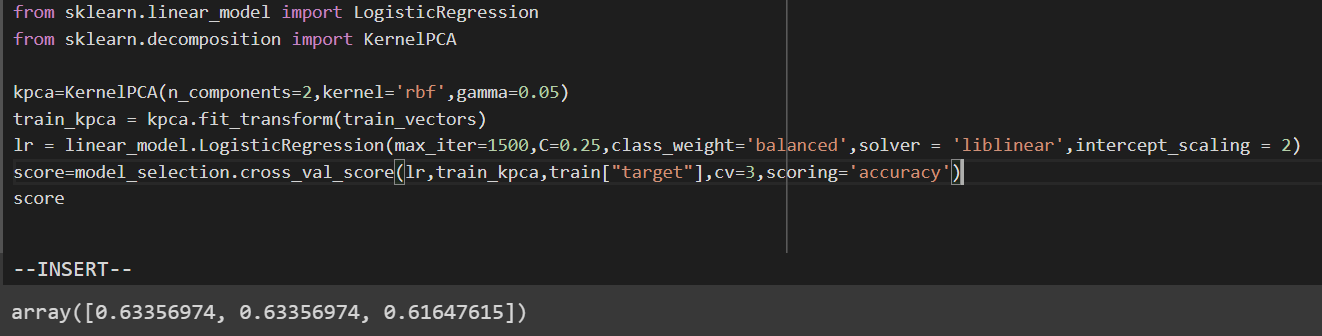
pca(n\_components = 0.99) scoring=accuracy



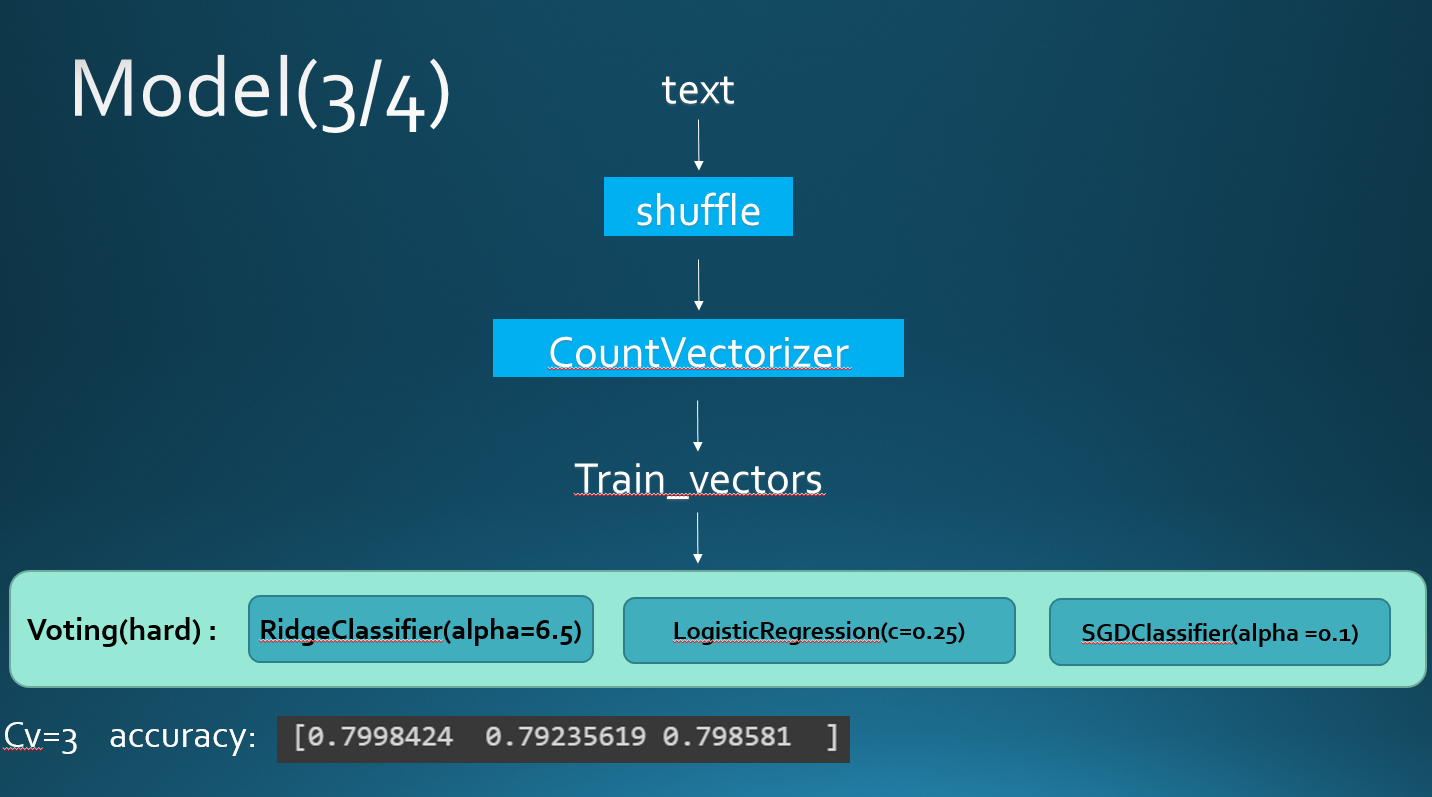
**Kpca**(n\_components = 2, kernel = rbf, gamma=0.05) accuracy 約為63%

降到二維，保留的資訊應該已經剩很少，但是他依然有63％正確率。

效果不理想原因: 降維過多，且最後沒有preimage回原來的維度就丟入scoring



**Final tuning :**

****

**Model :**

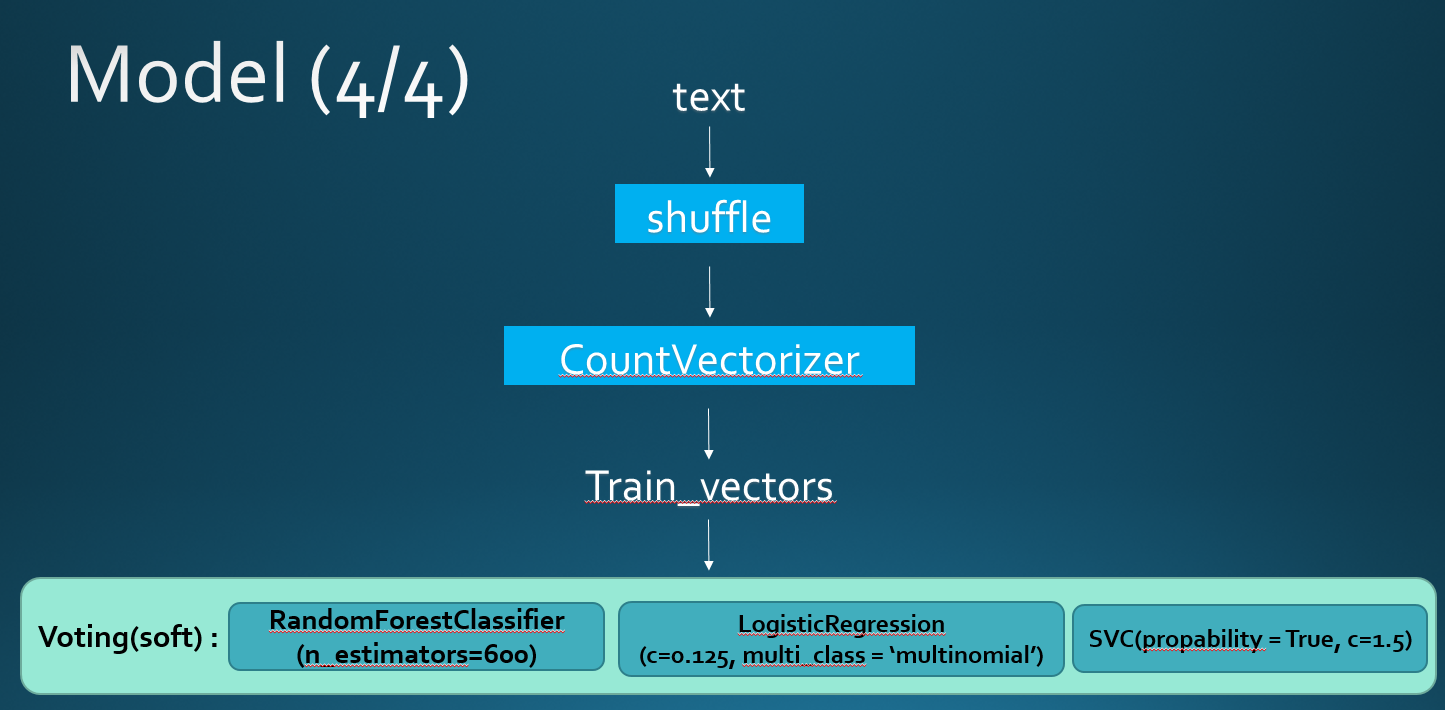
Voting (hard) => ridgeclassifier (alpha=6.5)

LogisticRegression(c=0.25)

SGDClassifier(alpha =0.1)

**Accuracy:** 0.7998424

Hard voting效果沒有比表現最好的model的效果好，我們認為是因為voting的model太少，沒辦法真正凸顯出hard voting 的優點。因此決定改用soft voting。



**Model :**

Voting (soft) => RandomForestClassifier(n\_estimators=600)

LogisticRegression(c=0.125, multi\_class = ‘multinomial’)

SVC(propability = True, c=1.5)

**Accuracy:** 0.80685849

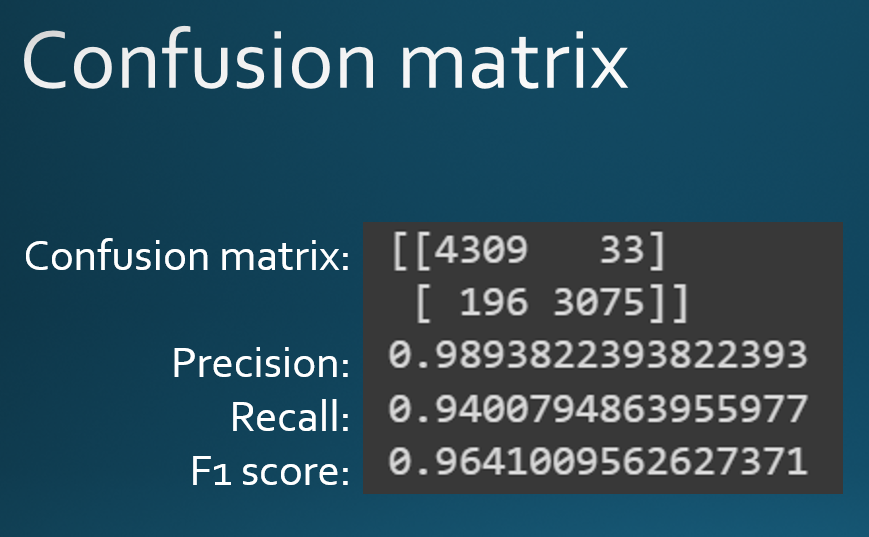
雖說效果只比hard voting 好一點，但soft voting 確實是所有classifier 中表現最好的，我們認為，只要再提升每個classifier的效能，少數的model也能用soft voting 發揮最大的優勢。

lr = LogisticRegression(C=0.125,multi\_class='multinomial',random\_state=1)

svm = SVC(probability=True,C=1.5)

rnd\_clf = RandomForestClassifier(n\_estimators=600,random\_state=1)

voting\_clf=VotingClassifier(estimators=[('lr',lr),('rnd',rnd\_clf('svm',svm)],voting='soft')



recall

Precision

整體上只看個別數值都表現不錯，日後如要再改進model，可以朝降低紅圈中的數值進步，因為此task為災難預測，FN的數值偏高，代表有較多災難沒有被預測出來，這點是有待加強的。

提交上kaggle後的accuracy是使用soft Voting Classifier： **0.80570 排名 ：277/867**

