Рубежный контроль №2 по курсу "Методы машинного обучения"

Вариант: 11

Выполнил: Саврасов П.А. группа ИУ5-24М

Задание

Необходимо решить задачу классификации текстов на основе любого выбранного Вами датасета. Классификация может быть бинарной или многоклассовой. Целевой признак из выбранного Вами датасета может иметь любой физический смысл, примером является задача анализа тональности текста.

Необходимо сформировать два варианта векторизации признаков - на основе CountVectorizer и на основе TfidfVectorizer. В качестве классификаторов необходимо использовать два классификатора по варианту для Вашей группы: Классификатор №1: KNeighborsClassifier Классификатор №2: Complement Naive Bayes (CNB)

Для каждого метода необходимо оценить качество классификации. Сделайте вывод о том, какой вариант векторизации признаков в паре с каким классификатором показал лучшее качество.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import ComplementNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

```
data = pd.read_csv('Youtube04.csv', sep = ',')
data.head()
```

	COMMENT_ID	AUTHOR	DATE	CONTENT	CLASS
0	z12rwfnyyrbsefonb232i5ehdxzkjzjs2	Lisa Wellas	NaN	+447935454150 lovely girl talk to me xxx	1
1	z130wpnwwnyuetxcn23xf5k5ynmkdpjrj04	jason graham	2015-05-29T02:26:10.652000	I always end up coming back to this song 	0
2	z13vsfqirtavjvu0t22ezrgzyorwxhpf3	Ajkal Khan	NaN	my sister just received over 6,500 new <a rel=...	1
3	z12wjzc4eprnvja4304cgbbizuved35wxcs	Dakota Taylor	2015-05-29T02:13:07.810000	Cool	0
4	z13xjfr42z3uxdz2223gx5rrzs3dt5hna	Jihad Naser	NaN	Hello I'am from Palastine	1

```
msgContent = data['CONTENT']
msgClass = data['CLASS']
```

```
TrainX,TestX,TrainY,TestY = train_test_split(msgContent, msgClass, test_size=0.3, random_state = 1)
report = []
```

```
def ModelPredictReport(vectoriser, classifier, modelName, vectName):
    model = Pipeline(
        [("vectorizer", vectoriser),
         ("classifier", classifier)])
    model.fit(TrainX,TrainY)
    prediction = model.predict(TestX)
    report = [modelName, vectName]
    report.append(accuracy_score(TestY,prediction))
    report.append(recall_score(TestY,prediction))
    report.append(f1_score(TestY,prediction))
    return report
```

```
report.append(ModelPredictReport(CountVectorizer(),KNeighborsClassifier(),'KNN','CountVectorizer'))
report.append(ModelPredictReport(TfidfVectorizer(),KNeighborsClassifier(),'KNN','TfidfVectorizer'))
```

```
report.append(ModelPredictReport(CountVectorizer(),ComplementNB(),'CNB','CountVectorizer'))
report.append(ModelPredictReport(TfidfVectorizer(),ComplementNB(),'CNB','TfidfVectorizer'))
```

```
pd.DataFrame(report, columns = ['Model','Vectorizer','Accuracy','Recall','F1'])
```

	Model	Vectorizer	Accuracy	Recall	F1
0	KNN	CountVectorizer	0.859259	0.724638	0.840336
1	KNN	TfidfVectorizer	0.585185	0.188406	0.317073
2	CNB	CountVectorizer	0.866667	0.927536	0.876712
3	CNB	TfidfVectorizer	0.851852	0.927536	0.864865