codebasics / py

Code Issues 30 Pull requests 32 Actions Projects Wiki Sec

master

py / ML / 14_naive_bayes / 14_naive_bayes_2_email_spam_filter.ipynb

dhavalsays Naive bayes History

1 contributor

475 lines (475 sloc) | 12.2 KB

In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv("spam.csv")
df.head()
```

Out[2]:

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

In [3]:
```
df.groupby('Category').describe()
```

Out[3]:

	Message			
	count	**unique**	**top**	**freq**
Category				
ham	4825	4516	Sorry, I'll call later	30
spam	747	641	Please call our customer service representativ...	4

In [4]:
```
df['spam']=df['Category'].apply(lambda x: 1 if x=='spam' else 0)
df.head()
```

Out[4]:

	Category	Message	spam
0	ham	Go until jurong point, crazy.. Available only ...	0
1	ham	Ok lar... Joking wif u oni...	0
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	1
3	ham	U dun say so early hor... U c already then say...	0
4	ham	Nah I don't think he goes to usf, he lives aro...	0

In [7]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df.Message,df.spam)
```

In [31]:
```
from sklearn.feature_extraction.text import CountVectorizer
v = CountVectorizer()
X_train_count = v.fit_transform(X_train.values)
X_train_count.toarray()[:2]
```

Out[31]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

In [23]:
```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(X_train_count,y_train)
```

Out[23]:
```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

In [37]:
```
emails = [
    'Hey mohan, can we get together to watch footbal game tomorrow?',
    'Upto 20% discount on parking, exclusive offer just for you. Dont m
iss this reward!'
]
emails_count = v.transform(emails)
model.predict(emails_count)
```

Out[37]:
```
array([0, 1], dtype=int64)
```

In [38]:
```
X_test_count = v.transform(X_test)
model.score(X_test_count, y_test)
```

Out[38]:
```
0.9827709978463748
```

Sklearn Pipeline

In [39]:
```
from sklearn.pipeline import Pipeline
clf = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('nb', MultinomialNB())
])
```

In [40]:
```
clf.fit(X_train, y_train)
```

Out[40]:
```
Pipeline(memory=None,
     steps=[('vectorizer', CountVectorizer(analyzer='word', binary=Fals
e, decode_error='strict',
        dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
        lowercase=True, max_df=1.0, max_features=None, min_df=1,
        ngram_range=(1, 1), preprocessor=None, stop_words=None,
        strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
        tokenizer=None, vocabulary=None)), ('nb', MultinomialNB(alpha=
1.0, class_prior=None, fit_prior=True))])
```

In [41]:
```
clf.score(X_test,y_test)
```

Out[41]:
```
0.9827709978463748
```