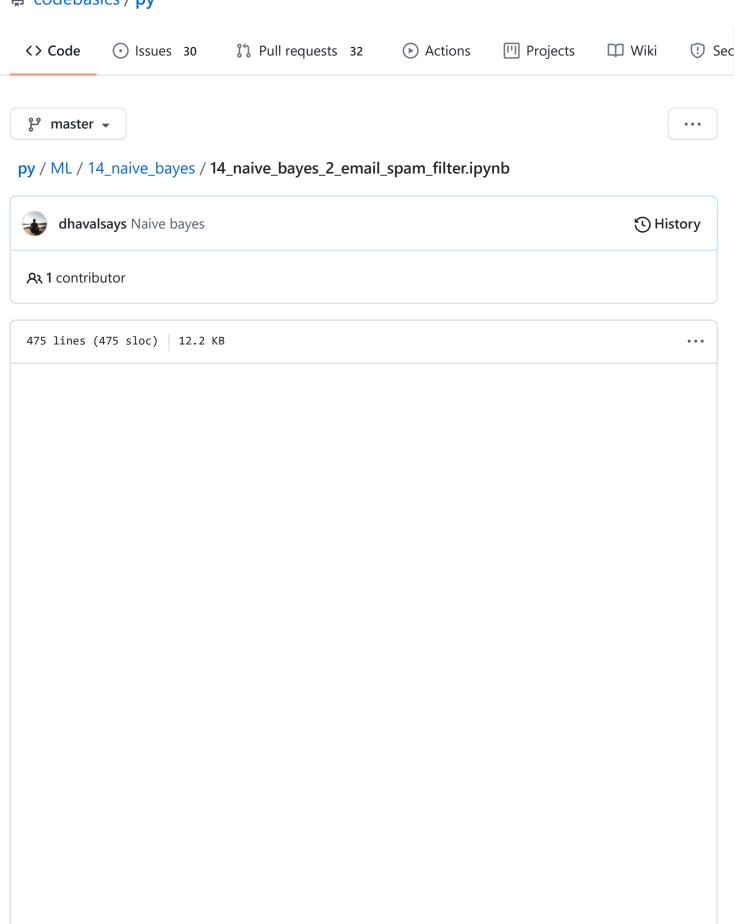
☐ codebasics / py



In [1]: import pandas as pd

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	

Out[4]:

	Category	y Message	
0	ham	Go until jurong point, crazy Available only	
1	ham	Ok lar Joking wif u oni	
2	spam	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	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[11] · 0 9827709978163718
```