```
import spacy
import pandas as pd


nlp = spacy.load('en_core_web_sm')


df = pd.read_csv('train.csv')


df.head()
```

1 to 5 of 5 entries Filter

index	sms	label
0	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...	0
1	Ok lar... Joking wif u oni...	0
2	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's	1
3	U dun say so early hor... U c already then say...	0
4	Nah I don't think he goes to usf, he lives around here though	0

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: Generate code with df View recommended plots

```
def lemmatization(text):
  doc = nlp(text)
  lemmalist = [token.lemma_ for token in doc]
  return ' '.join(lemmalist)


df['lemma']=df['sms'].apply(lemmatization)


df.head()
```

	sms	label	lemma
0	Go until jurong point, crazy.. Available only ...	0	go until jurong point , crazy .. available onl...
1	Ok lar... Joking wif u oni...\n	0	ok lar ... joke wif u oni ... \n
2	Free entry in 2 a wkly comp to win FA Cup fina...	1	free entry in 2 a wkly comp to win FA Cup fina...
3	U dun say so early hor... U c already then say...	0	u dun say so early hor ... u c already then sa...

Next steps: Generate code with df View recommended plots

```
def remove_stopwords(text):
  doc = nlp(text)
  no_stopwords = [token.text for token in doc if not token.is_stop and not token.is_punct]
  return ' '.join(no_stopwords)


df['preprocessed'] = df['lemma'].apply(remove_stopwords)


df.head()
```

	sms	label	lemma	preprocessed
0	Go until jurong point, crazy.. Available only ...	0	go until jurong point , crazy .. available onl...	jurong point crazy available bugis n great wor...
1	Ok lar... Joking wif u oni...\n	0	ok lar ... joke wif u oni ... \n	ok lar joke wif u oni \n
2	Free entry in 2 a wkly comp to win FA Cup fina...	1	free entry in 2 a wkly comp to win FA Cup fina...	free entry 2 wkly comp win FA Cup final tkts 2...

Next steps: Generate code with df View recommended plots

```
X = df['preprocessed']
y = df['label']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2, random_state = 42)
```

```
X_train.shape, X_test.shape
```

```
((4459,), (1115,))
```

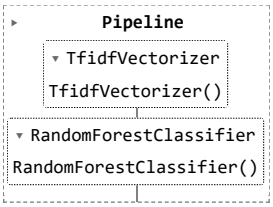
```
!pip install --upgrade scikit-learn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Collecting scikit-learn
  Downloading scikit_learn-1.5.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.4/13.4 MB 45.5 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.25.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Installing collected packages: scikit-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 1.2.2
    Uninstalling scikit-learn-1.2.2:
      Successfully uninstalled scikit-learn-1.2.2
Successfully installed scikit-learn-1.5.1
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
model = Pipeline([
    ('cvectorizer_tfidf', TfidfVectorizer()),
    ('Random Forest', RandomForestClassifier())
])

model.fit(X_train, y_train)
```




```
model.score(X_test, y_test) * 100
```

```
97.75784753363229
```

```
pred = model.predict(X_test)
```

```
y_test[:5]
```

```
3690    0
3527    0
724     0
3370    0
468     0
Name: label, dtype: int64
```

```
pred[:5]
```

```
array([0, 0, 0, 0, 0])
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       954
           1       1.00      0.84      0.92       161

    accuracy                           0.98      1115
   macro avg       0.99      0.92      0.95      1115
weighted avg       0.98      0.98      0.98      1115
```

```
import seaborn as sns
sns.set_style('darkgrid')
```

```
cf = confusion_matrix(y_test, pred, normalize = 'true')
sns.heatmap(cf, annot=True, cmap = 'Greens')
```

```
<Axes: >
```