spam classification using naive bayes

RAMANATHAN 2BAI1723

```
import pandas as pd
d=pd.read_csv('/content/spam.csv',encoding="latin-1")

d.head()
```

```
{"summary":"{\n  \"name\": \"d\",\n  \"rows\": 5572,\n  \"fields\": [\
n    {\n      \"column\": \"v1\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"spam\",\n          \"ham\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"v2\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5169,\n
\"samples\": [\n          \"Did u download the fring app?\",\n
\"Pass dis to all ur contacts n see wat u get! Red;i'm in luv wid u.
Blue;u put a smile on my face. Purple;u r realy hot. Pink;u r so swt.
Orange;i thnk i lyk u. Green;i realy wana go out wid u. Yelow;i wnt u
bck. Black;i'm jealous of u. Brown;i miss you Nw plz giv me one
color\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Unnamed: 2\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 43,\n
\"samples\": [\n          \" GOD said\",\n          \" SHE SHUDVETOLD
U. DID URGRAN KNOW?NEWAY\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Unnamed: 3\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 10,\n
\"samples\": [\n          \" \\\\\\\"OH No! COMPETITION\\\\\\\". Who
knew\",\n          \" why to miss them\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Unnamed: 4\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"GNT:-)\\\"\",\n          \" one day these two will become FREINDS
FOREVER!\\\"\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"d"}
```

```
pd.set_option('display.max_colwidth',0)
d=d[['v1','v2']]
d.head()
```

```
{"summary":"{\n  \"name\": \"d\",\n  \"rows\": 5572,\n  \"fields\": [\
n    {\n      \"column\": \"v1\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"spam\",\n          \"ham\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"v2\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5169,\n
\"samples\": [\n          \"Did u download the fring app?\",\n
\"Pass dis to all ur contacts n see wat u get! Red;i'm in luv wid u.
Blue;u put a smile on my face. Purple;u r realy hot. Pink;u r so swt.
Orange;i thnk i lyk u. Green;i realy wana go out wid u. Yelow;i wnt u
bck. Black;i'm jealous of u. Brown;i miss you Nw plz giv me one
color\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"d"}
```

```
import string
string.punctuation

def rem_pun(text):
  pun="".join([i for i in text if i not in string.punctuation])
  return pun

d['a']=d['v2'].apply(lambda x : rem_pun(x))

d.head()
```

```
{"summary":"{\n  \"name\": \"d\",\n  \"rows\": 5572,\n  \"fields\": [\
n    {\n      \"column\": \"v1\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"spam\",\n          \"ham\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"v2\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5169,\n
\"samples\": [\n          \"Did u download the fring app?\",\n
\"Pass dis to all ur contacts n see wat u get! Red;i'm in luv wid u.
Blue;u put a smile on my face. Purple;u r realy hot. Pink;u r so swt.
Orange;i thnk i lyk u. Green;i realy wana go out wid u. Yelow;i wnt u
bck. Black;i'm jealous of u. Brown;i miss you Nw plz giv me one
color\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"a\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5144,\n        \"samples\": [\n          \"This
is ur face test  1 2 3 4 5 6 7 8 9  ltgt   select any number i will
tell ur face astrology am waiting quick reply\",\n          \"HEY BABE
FAR 2 SPUNOUT 2 SPK AT DA MO DEAD 2 DA WRLD BEEN SLEEPING ON DA SOFA
ALL DAY\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"d"}
```

```
d['l']=d['a'].apply(lambda x:x.lower())

d.head()
```

```
{"summary":"{\n  \"name\": \"d\",\n  \"rows\": 5572,\n  \"fields\": [\
n    {\n      \"column\": \"v1\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"spam\",\n          \"ham\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"v2\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5169,\n
\"samples\": [\n          \"Did u download the fring app?\",\n
\"Pass dis to all ur contacts n see wat u get! Red;i'm in luv wid u.
Blue;u put a smile on my face. Purple;u r realy hot. Pink;u r so swt.
Orange;i thnk i lyk u. Green;i realy wana go out wid u. Yelow;i wnt u
bck. Black;i'm jealous of u. Brown;i miss you Nw plz giv me one
color\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"a\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5144,\n        \"samples\": [\n          \"This
is ur face test  1 2 3 4 5 6 7 8 9  ltgt   select any number i will
tell ur face astrology am waiting quick reply\",\n          \"HEY BABE
FAR 2 SPUNOUT 2 SPK AT DA MO DEAD 2 DA WRLD BEEN SLEEPING ON DA SOFA
ALL DAY\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"l\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5142,\n        \"samples\": [\n          \"oh
only 4 outside players allowed to play know\",\n          \"aight
should i just plan to come up later tonight\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"d"}
```

```
import re
def tok(text):
  t=re.split('\W+',text)
  return t

d['token']=d['l'].apply(lambda x:tok(x))

d.head()
```

```
{"summary":"{\n  \"name\": \"d\",\n  \"rows\": 5572,\n  \"fields\": [\
n    {\n      \"column\": \"v1\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"spam\",\n          \"ham\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"v2\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5169,\n
\"samples\": [\n          \"Did u download the fring app?\",\n
\"Pass dis to all ur contacts n see wat u get! Red;i'm in luv wid u.
Blue;u put a smile on my face. Purple;u r realy hot. Pink;u r so swt.
Orange;i thnk i lyk u. Green;i realy wana go out wid u. Yelow;i wnt u
bck. Black;i'm jealous of u. Brown;i miss you Nw plz giv me one
color\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"a\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5144,\n        \"samples\": [\n          \"This
is ur face test  1 2 3 4 5 6 7 8 9  ltgt   select any number i will
tell ur face astrology am waiting quick reply\",\n          \"HEY BABE
FAR 2 SPUNOUT 2 SPK AT DA MO DEAD 2 DA WRLD BEEN SLEEPING ON DA SOFA
ALL DAY\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"l\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5142,\n        \"samples\": [\n          \"oh
only 4 outside players allowed to play know\",\n          \"aight
should i just plan to come up later tonight\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"token\",\n      \"properties\": {\
n        \"dtype\": \"object\",\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"d"}
```

```
import nltk
nltk.download('stopwords')
sw=nltk.corpus.stopwords.words('english')
print(sw[0:10])
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',
"you're"]
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
def rem_sw(text):
  swr=[i for i in text if i not in sw]
  return swr
```

```
d['s']=d['token'].apply(rem_sw)
```

```
d.head()
```

```
{"summary":"{\n  \"name\": \"d\",\n  \"rows\": 5572,\n  \"fields\": [\
n    {\n      \"column\": \"v1\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"spam\",\n          \"ham\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"v2\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5169,\n
\"samples\": [\n          \"Did u download the fring app?\",\n
\"Pass dis to all ur contacts n see wat u get! Red;i'm in luv wid u.
Blue;u put a smile on my face. Purple;u r realy hot. Pink;u r so swt.
Orange;i thnk i lyk u. Green;i realy wana go out wid u. Yelow;i wnt u
bck. Black;i'm jealous of u. Brown;i miss you Nw plz giv me one
color\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"a\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5144,\n        \"samples\": [\n          \"This
is ur face test  1 2 3 4 5 6 7 8 9  ltgt   select any number i will
tell ur face astrology am waiting quick reply\",\n          \"HEY BABE
FAR 2 SPUNOUT 2 SPK AT DA MO DEAD 2 DA WRLD BEEN SLEEPING ON DA SOFA
ALL DAY\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"l\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5142,\n        \"samples\": [\n          \"oh
only 4 outside players allowed to play know\",\n          \"aight
should i just plan to come up later tonight\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"token\",\n      \"properties\": {\
n        \"dtype\": \"object\",\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"s\",\n      \"properties\": {\n        \"dtype\": \"object\",\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"d"}
```

STEMMING

```
# prompt: from nltk.stem.porterps

from nltk.stem import PorterStemmer
ps=PorterStemmer()

def stem(text):
  st=[ps.stem(i) for i in text]
  return st

d['st']=d['s'].apply(stem)

d.head()
```

```
{"summary":"{\n  \"name\": \"d\",\n  \"rows\": 5572,\n  \"fields\": [\
n    {\n      \"column\": \"v1\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"spam\",\n          \"ham\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"v2\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5169,\n
\"samples\": [\n          \"Did u download the fring app?\",\n
\"Pass dis to all ur contacts n see wat u get! Red;i'm in luv wid u.
Blue;u put a smile on my face. Purple;u r realy hot. Pink;u r so swt.
Orange;i thnk i lyk u. Green;i realy wana go out wid u. Yelow;i wnt u
bck. Black;i'm jealous of u. Brown;i miss you Nw plz giv me one
color\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"a\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5144,\n        \"samples\": [\n          \"This
is ur face test  1 2 3 4 5 6 7 8 9  ltgt   select any number i will
tell ur face astrology am waiting quick reply\",\n          \"HEY BABE
FAR 2 SPUNOUT 2 SPK AT DA MO DEAD 2 DA WRLD BEEN SLEEPING ON DA SOFA
ALL DAY\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"l\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5142,\n        \"samples\": [\n          \"oh
only 4 outside players allowed to play know\",\n          \"aight
should i just plan to come up later tonight\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"token\",\n      \"properties\": {\
n        \"dtype\": \"object\",\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"s\",\n      \"properties\": {\n        \"dtype\": \"object\",\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"st\",\n      \"properties\": {\n
\"dtype\": \"object\",\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"d"}
```

NAIVE BAYES using count vectorizer

```
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer()
x=cv.fit_transform(d['st'].astype(str))

y=d['v1']

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)

from sklearn.naive_bayes import MultinomialNB
model=MultinomialNB()
model.fit(x_train,y_train)

y_pred=model.predict(x_test)



from sklearn.metrics import
accuracy_score,confusion_matrix,classification_report
print("Accuracy is :",accuracy_score(y_test,y_pred))
print("Classification Report is :\
n",classification_report(y_test,y_pred))
```

```
Accuracy is : 0.9811659192825112
Classification Report is :
               precision    recall  f1-score   support

         ham       0.99      0.99      0.99       967
        spam       0.92      0.94      0.93       148
```

```
    accuracy                           0.98      1115
   macro avg       0.96      0.96      0.96      1115
weighted avg       0.98      0.98      0.98      1115
```

USING TFID

```
from sklearn.feature_extraction.text import TfidfVectorizer


tfidf = TfidfVectorizer()
x_tfidf = tfidf.fit_transform(d['st'].astype(str))


x_train_tfidf, x_test_tfidf, y_train, y_test =
train_test_split(x_tfidf, y, test_size=0.2)
model_tfidf = MultinomialNB()
model_tfidf.fit(x_train_tfidf, y_train)


y_pred_tfidf = model_tfidf.predict(x_test_tfidf)


print("Accuracy (TF-IDF):", accuracy_score(y_test, y_pred_tfidf))
print("Classification Report (TF-IDF):\n",
classification_report(y_test, y_pred_tfidf))
```

```
Accuracy (TF-IDF): 0.9623318385650225
Classification Report (TF-IDF):
               precision    recall  f1-score   support

         ham       0.96      1.00      0.98       963
        spam       1.00      0.72      0.84       152

    accuracy                           0.96      1115
   macro avg       0.98      0.86      0.91      1115
weighted avg       0.96      0.96      0.96      1115
```

CONCLUSION

```
COUNT VECTORIZER HAS HIGHER ACCURACY(98%) THAN TFID (96%)
```