

1.02-CK-Analyse_NaiveBayes

November 15, 2024

```
[1]: import os
import sys
sys.path.append(os.path.dirname(os.getcwd()))
from src.load_covid19 import load_clean_covid19
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
import scipy
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score, precision_score,
↳ recall_score
```

```
[2]: df = load_clean_covid19()
```

Dataset already exists at C:\Users\peter\covid-19-risiko-erkennung\src\..\data\raw\covid19-dataset. Skipping download.
Saving clean dataset to: C:\Users\peter\covid-19-risiko-erkennung\data\interim\covid-data-clean.csv
Saved
Loading clean dataset from: C:\Users\peter\covid-19-risiko-erkennung\data\interim\covid-data-clean.csv

```
[3]: #Dropped USMER, MEDICAL_UNIT, PATIENT_TYPE, INTUBED, .CLASIFFICATION_FINAL, ICU,
↳ DIED
columns_to_keep=['SEX', 'INTUBED', 'PNEUMONIA', 'AGE', 'PREGNANT', 'DIABETES',
↳ 'COPD', 'ASTHMA', 'INMSUPR',
'HIPERTENSION', 'OTHER_DISEASE', 'CARDIOVASCULAR',
↳ 'OBESITY', 'RENAL_CHRONIC', 'TOBACCO']

df=df[[col for col in columns_to_keep]]
df = df.dropna()
for col in df.select_dtypes(include=bool).columns:
    df[col] = df[col].apply(lambda x: 1 if x == True else 0)

df['SEX']=df['SEX'].apply(lambda x: 1 if x=='female' else 0)
```

```
[11]: # train(90%), test(5%)
train, test = train_test_split(df, test_size=0.05, shuffle=True)

train_y = train.INTUBED
train_x = train.drop(columns = ['INTUBED'])

test_y = test.INTUBED
test_x = test.drop(columns = ['INTUBED'])

def get_scores(y_test, y_pred):
    acc = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    prec = precision_score(y_test, y_pred)
    rec = recall_score(y_test, y_pred)
    return acc, f1, prec, rec
```

```
[15]: from sklearn.naive_bayes import CategoricalNB
model_cat = CategoricalNB()
model_cat.fit(train_x, train_y)
```

```
[15]: CategoricalNB()
```

```
[16]: y_pred = model_cat.predict(test_x)
```

```
[17]: acc, f1, prec, rec = get_scores(test_y, y_pred)
print(f'Accuracy: {acc}')
print(f'F1: {f1}')
print(f'Precision: {prec}')
print(f'Recall: {rec}')
```

```
Accuracy: 0.8494148244473342
F1: 0.0034423407917383822
Precision: 0.3333333333333333
Recall: 0.0017301038062283738
```

```
[18]: from sklearn.naive_bayes import GaussianNB
model_gauss = GaussianNB()
model_gauss.fit(train_x, train_y)
```

```
[18]: GaussianNB()
```

```
[19]: y_pred_gauss = model_gauss.predict(test_x)
```

```
[20]: acc, f1, prec, rec = get_scores(test_y, y_pred_gauss)
print(f'Accuracy: {acc}')
print(f'F1: {f1}')
print(f'Precision: {prec}')
print(f'Recall: {rec}')
```

Accuracy: 0.764369310793238
F1: 0.19823008849557522
Precision: 0.2028985507246377
Recall: 0.19377162629757785

[]: