Lab 4 - Naive Bayes Classifier Titanic Dataset

Author: Krishna Swaroop

181CO125, NITK Surathkal

Introduction

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

Dataset

We are using the Titanic Disaster Dataset. It gathers personal information about the passengers onboard the Titanic Ship which met a iceberg crash in the ocean in 1912.

Our target is to predict whether or not a person will survive with the given features

Features:

> | Variable | Definition | Key |
> |---|---|---|
> | survival | Survival | 0 = No, 1 = Yes |
> | pclass | Ticket class | 1 = 1st, 2 = 2nd, 3 = 3rd |
> | sex | Sex | |
> | Age | Age in years | |
> | sibsp | # of siblings / spouses aboard the Titanic | |
> | parch | # of parents / children aboard the Titanic | |
> | ticket | Ticket number | |
> | fare | Passenger fare | |
> | cabin | Cabin number | |
> | embarked | Port of Embarkation | C = Cherbourg, Q = Queenstown, S = Southampton> |

Survival is numbered as 1/0 respectively and is our target variable

Naive Bayes Classifier

1) Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import time
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, BernoulliNB, MultinomialNB
```

2) Load data

```
data = pd.read_csv("/content/train.csv")
```

Print head of dataset

```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 714 entries, 0 to 890
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Survived          714 non-null    int64
 1   Pclass            714 non-null    int64
 2   Sex_cleaned       714 non-null    int64
 3   Age               714 non-null    float64
 4   SibSp             714 non-null    int64
 5   Parch             714 non-null    int64
 6   Fare              714 non-null    float64
 7   Embarked_cleaned  714 non-null    int64
dtypes: float64(2), int64(6)
memory usage: 50.2 KB
None
```

3) Clean data

Here, Male is replaced by 0 and female is replaced by 1

```
# Convert categorical variable to numeric
data["Sex_cleaned"]=np.where(data["Sex"]=="male",0,1)
data["Embarked_cleaned"]=np.where(data["Embarked"]=="S",0, np.where(data["Embarked"
```

Print cleaned data values

```
print(data.head())
```

```
       PassengerId  Survived  Pclass  ... Embarked Sex_cleaned  Embarked_cleaned
```

```
0           1         0       3  ...        S          0                 0
1           2         1       1  ...        C          1                 1
2           3         1       3  ...        S          1                 0
3           4         1       1  ...        S          1                 0
4           5         0       3  ...        S          0                 0

[5 rows x 14 columns]
```

Select features

```
data=data[[
    "Survived",
    "Pclass",
    "Sex_cleaned",
    "Age",
    "SibSp",
    "Parch",
    "Fare",
    "Embarked_cleaned"
]].dropna(axis=0, how='any')
```

4) Split data

Use `train_test_split()` to split the data to training and testing dataset. Here, 20% of the dataset is reserved to test our algorithm

```
X_train, X_test = train_test_split(data, test_size=0.2, random_state=10)
```

5) Fit model

```
gnb = GaussianNB()
```

```
used_features =[
    "Pclass",
    "Sex_cleaned",
    "Age",
    "SibSp",
    "Parch",
    "Fare",
    "Embarked_cleaned"
]

# Train classifier
gnb.fit(X_train[used_features].values,X_train["Survived"])
```

```
    GaussianNB(priors=None, var_smoothing=1e-09)
```

6) Predict

```
y_pred = gnb.predict(X_test[used_features])

# Print results
print("Number of mislabeled points out of a total {} points : {}, performance {:05.
      .format(
          X_test.shape[0],
          (X_test["Survived"] != y_pred).sum(),
          100*(1-(X_test["Survived"] != y_pred).sum()/X_test.shape[0])
))
```

```
    Number of mislabeled points out of a total 143 points : 31, performance 78.32%
```

7) Analysis

We get a performance of 78.32% on the test set containing 143 points. Hence there are a total of 31 mislabelled points in our test dataset