Machine Learning

Lab : 3

Naive Bayes Classifier

December 2024

Perform Date: December 16-20, 2024

1 Objective

To implement a simple Naive Bayes Classifier in Python.

2 Description

The example data set we use to perform the classification is as shown below:

ID :1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
Weather : 'Sunny', 'Sunny', 'Overcast', 'Rainy', 'Rainy','Rainy', 'Overcast', 'Sunny', 'Sunny', 'Rainy', 'Sunny', 'Overcast', 'Overcast', 'Rainy'
Temperature : 'Hot','Hot','Hot','Mild','Cool','Cool','Cool','Mild', 'Cool','Mild','Mild','Mild','Hot','Mild'

Play='No','No','Yes','Yes','Yes','No','Yes','No','Yes', 'Yes','Yes','Yes','Yes','No'

As show above the example data set has 14 examples. Each examples has two features [Weather, Temperature] and corresponding class label [Play]. Weather and Temperature are nominal attributes. Values for 'weather' are [Sunny, Overcast, Rainy] and values for 'Temperature' are [Hot, Mild, Cold]. Class label 'Play' divides examples into two classes [Yes, No]. e.g. the first example is ID1 {Sunny, Hot, No}, meaning "if the 'Weather' is 'Sunny' & 'Temperature' is 'Hot' , then 'Play' is 'No'.

The model parameters are : P(play), P(temperature/play) and P(weather/play)

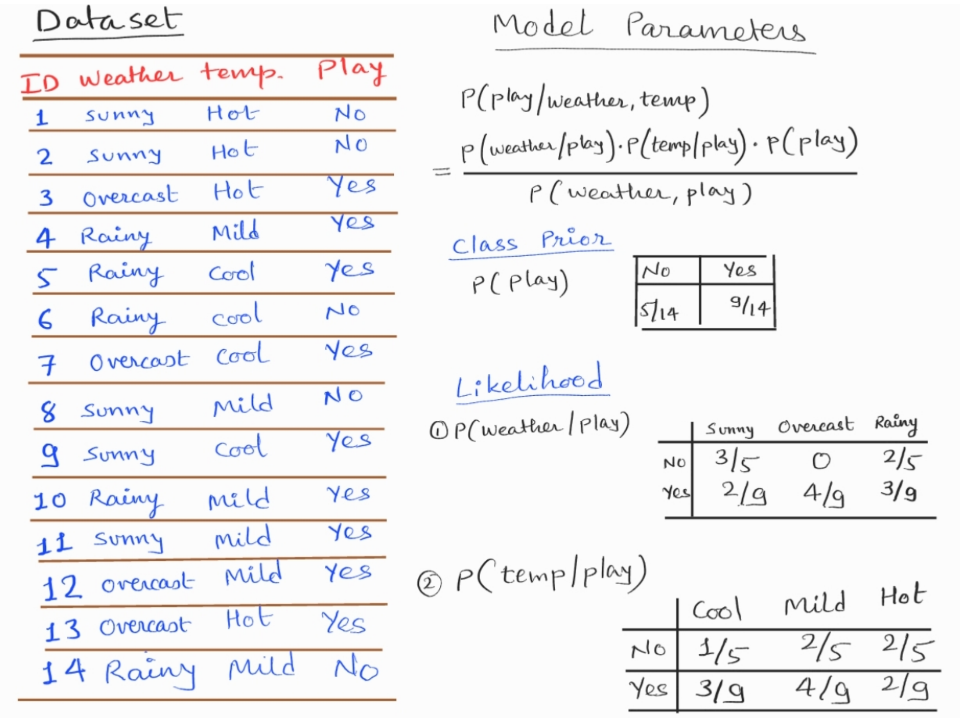
[6]:
```python
from IPython.display import Image
from IPython.core.display import HTML
PATH = "/Users/brijeshbhatt/Downloads/"
```

2.1 Manual Calculation

To understand the working of Naive Bayes classifier let's first manually calculate the model parameters and an example of inference.

Following figures shows the model parameters calculated from the data set.

[7]:



Lets take an example inferencing question as, "Will you go out to play if the weather is sunny & Temperature is mild?

[8]:

Inferencing :

$$P(\text{Play}=\text{Yes} / \text{weather}=\text{sunny}, \text{temp}=\text{mild})$$

$$= \frac{\overset{(A)}{P(\text{weather}=\text{sunny}/\text{play}=\text{Yes})} \cdot \overset{(B)}{P(\text{temp}=\text{mild}/\text{play}=\text{yes})} \cdot \overset{(C)}{P(\text{play}=\text{yes})}}{\underset{(D)}{P(\text{weather}=\text{sunny}, \text{temp}=\text{mild})}}$$

(A) $P(\text{weather}=\text{Sunny} / \text{play}=\text{yes})$

$= 2/9$ [From CPT of likelihood $P(\text{weather}/\text{play})$]

(B) $P(\text{temp}=\text{mild} / \text{play}=\text{yes})$

$= 4/9$ [From CPT of likelihood $P(\text{temp}/\text{play})$]

(C) $P(\text{play}=\text{yes})$

$= 9/14$ [From CPT of prior $P(\text{play})$]

The calculations for A, B, & C can be easily done from the conditional probability tables. The denominator can be calculated as shown in the figure below.

[9]:

(D) $P(\text{weather}=\text{Sunny}, \text{temp}=\text{mild})$

$= P(\text{weather}=\text{sunny}, \text{temp}=\text{mild}, \text{play}=\text{yes}) + P(\text{weather}=\text{Sunny}, \text{temp}=\text{mild}, \text{play}=\text{No})$ [margin-alize 'play']

$= P(\text{weather}=\text{Sunny}, \text{temp}=\text{mild} / \text{play}=\text{yes})\, P(\text{play}=\text{yes}) + P(\text{weather}=\text{Sunny}, \text{temp}=\text{mild} / \text{play}=\text{No})\, P(\text{play}=\text{No})$ [Bayes Model]

$= P(\text{weather}=\text{Sunny} / \text{play}=\text{yes}) \cdot P(\text{temp}=\text{mild} / \text{play}=\text{yes})\, P(\text{play}=\text{yes}) + P(\text{weather}=\text{Sunny} / \text{play}=\text{No})\, P(\text{temp}=\text{mild} / \text{play}=\text{No}) \cdot P(\text{play}=\text{No})$

[Naive Assumption]

$= (2/9)(4/9)(9/14) + (3/5)(2/5)(5/14)$

$= \frac{4}{63} + \frac{3}{35}$

The final calculation is as follows,

[10]:

$$P(\text{Play} = \text{Yes} \,/\, \text{weather} = \text{Sunny}, \text{temp} = \text{mild})$$
$$= \frac{A \times B \times C}{D}$$
$$= \frac{(2/9)(4/9)(9/14)}{\left(\frac{4}{63} + \frac{3}{35}\right)}$$
$$= \frac{(4/63)}{(4/63 + 3/35)}$$
$$= \frac{0.063}{0.063 + 0.0857}$$
$$= \frac{0.063}{0.1487}$$
$$= 0.4236$$

3 Implementation Guidelines

3.0.1 Step 1: Import necessary libraries.

We will use preprocessing and naive bayes libraries of sklearn

[1]:
```
from sklearn import preprocessing
from sklearn.naive_bayes import GaussianNB, MultinomialNB
```

3.0.2 Step 2: Prepare dataset.

Create feature set for weather and temperature, and classlabel play.

[2]:
```
weather = ['Sunny', 'Sunny', 'Overcast', 'Rainy', 'Rainy','Rainy', 'Overcast',
           'Sunny', 'Sunny', 'Rainy', 'Sunny', 'Overcast', 'Overcast', 'Rainy']


temp = ['Hot','Hot','Hot','Mild','Cool','Cool','Cool','Mild',
        'Cool','Mild','Mild','Mild','Hot','Mild']
```

```
play=['No','No','Yes','Yes','Yes','No','Yes','No','Yes',
      'Yes','Yes','Yes','Yes','No']
```

3.0.3 Step 3: Digitize the data set using encoding

[3]:
```
#creating labelEncoder
le = preprocessing.LabelEncoder()

# Converting string labels into numbers.
weather_encoded=le.fit_transform(weather)
print("Weather:" ,weather_encoded)
```

```
Weather: [2 2 0 1 1 1 0 2 2 1 2 0 0 1]
```

[4]:
```
temp_encoded=le.fit_transform(temp)
label=le.fit_transform(play)

print("Temp:",temp_encoded)
print("Play:",label)
```

```
Temp: [1 1 1 2 0 0 0 2 0 2 2 2 1 2]
Play: [0 0 1 1 1 0 1 0 1 1 1 1 1 0]
```

3.0.4 Step 4: Merge different features to prepare dataset

[5]:
```
#Combinig weather and temp into single listof tuples
features=tuple(zip(weather_encoded,temp_encoded))
print("Features:",features)
```

```
Features: ((2, 1), (2, 1), (0, 1), (1, 2), (1, 0), (1, 0),␣
↪(0, 0), (2, 2), (2,
0), (1, 2), (2, 2), (0, 2), (0, 1), (1, 2))
```

3.0.5 Step 5: Train 'Naive Bayes Classifier'

[24]:
```
#Create a Classifier
model=MultinomialNB(alpha=0, fit_prior=True)
#model=MultinomialNB(alpha=0, force_alpha=True)
# Train the model using the training sets
model.fit(features,label)
model.get_params()
```

```
/Users/brijeshbhatt/miniforge3/envs/ml/lib/python3.9/site-
packages/sklearn/naive_bayes.py:591: UserWarning: alpha too␣
↪small will result in
numeric errors, setting alpha = 1.0e-10
  warnings.warn(
```

[24]:
```
{'alpha': 0, 'class_prior': None, 'fit_prior': True}
```

3.0.6 Step 6: Predict Output for new data

[25]:
```
#Predict Output
predicted= model.predict([[0,2]]) # 0:Overcast, 2:Mild
#prob = model.predict_proba([0,2])
print("Predicted Value:", predicted)#, prob)
```

```
Predicted Value: [1]
```

[26]:
```
predicted= model.predict([[0,1]]) # 0:Overcast, 1:Hot
print("Predicted Value:", predicted)
```

```
Predicted Value: [1]
```

[27]:
```
predicted= model.predict([[2,2]]) # 2:Sunny, 2:Mild


print("Predicted Value:", predicted)
```

```
Predicted Value: [1]
```

4 Exercise

4.1 Manually calculate output for the following cases and compare it with system's output.

4.1.1 (1) Will you play if the temperature is 'Hot' and weather is 'overcast'?

4.1.2 (2) Will you play if the temperature is 'Mild' and weather is 'Sunny'?

5 Exercise 1.1

1. **1** $<=$ Rollnumber $<=$ 25: #Task 1: Try the algo on Dataset1 - OneHotEncoding of features: and Train test Division 70%-30%

   #Task 2: Apply algorithm on digits dataset - LabelEncoding of features: and Train test Division $80\% - 20\%$

2. **26** $<=$ Rollnumber $<=$ **50** : #Task 1: Try the algo on Dataset1 - LabelEncoding of features: and Train test Division 80%-20%

   #Task 2: Apply algorithm on breast cancer wisconsin dataset - One Hot Encoding of features: and Train test Division $60\% - 40\%$

3. **51** $<=$ Rollnumber $<=$ **75** : #Task 1: Try the algo on Dataset2 - LabelEncoding of features: and Train test Division $90\% - 10\%$

   #Task 2: Apply algorithm on digits dataset - One Hot Encoding of features: and Train test Division $65\% - 35\%$

4. **76** $<=$ Rollnumber $<=$ 100: #Task 1: Try the algo on Dataset2 - OneHotEncoding of features: and Train test Division 75%-25% #Task 2: Apply algorithm on wine dataset - LabelEncoding of features: and Train test Division $80\% - 20\%$

5. **101** $<=$ Rollnumber $<=$ 125: #Task 1: Try the algo on Dataset3 - OneHotEncoding of features:and Train test Division 85%-15% #Task 2: Apply algorithm on wine dataset - LabelEncoding of features: and Train test Division $66\% - 34\%$

6. **126** $<=$ Rollnumber: #Task 1: Try the algo on Dataset3 LabelEncoding of features:and Train test Division 95%-5%

#Task 2: Apply algorithm on breast cancer wisconsin dataset - One Hot Encoding of features: and Train test Division 50% – 50%

5.1 Instruction for Task-1 & 2:

i) Set Random state of model equals to your roll number (or last 2 digit of your id)

5.2 Questions: For Task - 1

(1) What will be the value of Play, if Outlook is 'Rainy', Temperature is 'Mild', Humidity ='Normal', and Wind = 'False'?

(2) What will be the value of Play, if Outlook is 'Sunny', Temeprature is 'Cool', Humidity ='High', and Wind = 'True'?

5.3 Questions: For Task - 1 and Task 2

(3) Accuracy , precision and recall of both Models?

(4) For question 1 and 2 mention the probability estimates of the trained model.

(5) Give insights into the model trained.

(6) Describe the datasets used in Task 1 and Task 2.

6 References

1. https://scikit-learn.org/stable/modules/naive_bayes.html
2. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html