1_NB_Classifier_Weather

December 20, 2021

1 Machine Learning Lab#3

1.1 Aim: Implement Naive Bayes classifier : Weather Example

1.2 Introduction

In this lab we will implement a simple Naive Bayes Classifier in Python. 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)

1.3 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]:

Dataset

ID	weather	temp.	Play
1	Sunny	Hot	No
2	Sunny	Hot	No
3	Overcast	Hot	Yes
4	Rainy	Mild	Yes
5	Rainy	Cool	Yes
6	Rainy	cool	No
7	Overcast	Cool	Yes
8	Sunny	Mild	No
9	Sunny	Cool	Yes
10	Rainy	Mild	Yes
11	Sunny	mild	Yes
12	overcast	Mild	Yes
13	Overcast	Hot	Yes
14	Rainy	Mild	No

Model Parameters

$$P(play/weather, temp) = \frac{P(weather/play) \cdot P(temp/play) \cdot P(play)}{P(weather, play)}$$

Class Prior

P(Play)

No	Yes
5/14	9/14

Likelihood

① P(weather/Play)

	Sunny	Overcast	Rainy
No	3/5	0	2/5
Yes	2/9	4/9	3/9

② P(temp/play)

	Cool	Mild	Hot
No	1/5	2/5	2/5
Yes	3/9	4/9	2/9

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

[8]:

Inferencing :

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

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

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

$$= P(\text{weather} = \text{sunny}, \text{temp} = \text{mild}, \text{play} = \text{yes}) + P(\text{weather} = \text{sunny}, \text{temp} = \text{mild}, \text{play} = \text{No})$$ [marginalize '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]:

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

$$= \frac{A \times B \times C}{D}$$

$$= \frac{(2/9)(4/9)(9/14)}{\left(\frac{4}{63} + \frac{3}{35}\right)}$$

$$= \frac{(4/63)}{(4/63 + 3/35)}$$

$$= \frac{0.063}{0.063 + 0.0857}$$

$$= \frac{0.063}{0.1487}$$

$$= 0.4236$$

1.4 Implementation

1.4.1 Step 1: Import necessary libraries.

We will use preprocessing and naive bayes libraries of sklearn

[11]:
```
from sklearn import preprocessing
from sklearn.naive_bayes import GaussianNB, MultinomialNB
```

1.4.2 Step 2: Prepare dataset.

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

[12]:
```
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']
```

1.4.3 Step 3: Digitize the data set using encoding

[13]:
```
#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]
```

[14]:
```
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]
```

1.4.4 Step 4: Merge different features to prepare dataset

[15]:
```
#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))
```

1.4.5 Step 5: Train 'Naive Bayes Classifier'

[16]:
```
#Create a Classifier
model=MultinomialNB()
# Train the model using the training sets
model.fit(features,label)
```

[16]:
```
MultinomialNB()
```

1.4.6 Step 6: Predict Output for new data

[17]:
```
#Predict Output
predicted= model.predict([[0,2]]) # 0:Overcast, 2:Mild
print("Predicted Value:", predicted)
```

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

[18]:
```python
predicted= model.predict([[0,1]]) # 0:Overcast, 1:Hot
print("Predicted Value:", predicted)
```

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

[19]:
```python
predicted= model.predict([[2,2]]) # 2:Sunny, 2:Mild

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

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

1.5 Exercise: Manually calculate output for the following cases and compare it with system's output.

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

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

[ ]:
```python

```