동국대학교
dongguk university

Probabilistic Reasoning

Practice4

실습 1. Basic Probability

- Seeing Theory
  - https://students.brown.edu/seeing-theory/index.html

- Basic Probability

Example of Random Sampling with Replacement

```
# Probability of drawing spade
printProb(event_probability(deck['spades'], deck['cards']))
printProb(event_probability(deck['spades'], deck['cards']))
printProb(event_probability(deck['spades'], deck['cards']))
# Probability of drawing spade
printProb(event_probability(deck['hearts'], deck['cards']))
printProb(event_probability(deck['hearts'], deck['cards']))
printProb(event_probability(deck['hearts'], deck['cards']))
```

```
25.0%
25.0%
25.0%
25.0%
25.0%
25.0%
```

Example of Random Sampling without Replacement

```
# returns the probability of drawing the card then draw from the deck
def draw(card, deck):
    #fill the (1)
    (1)
    return prob
```

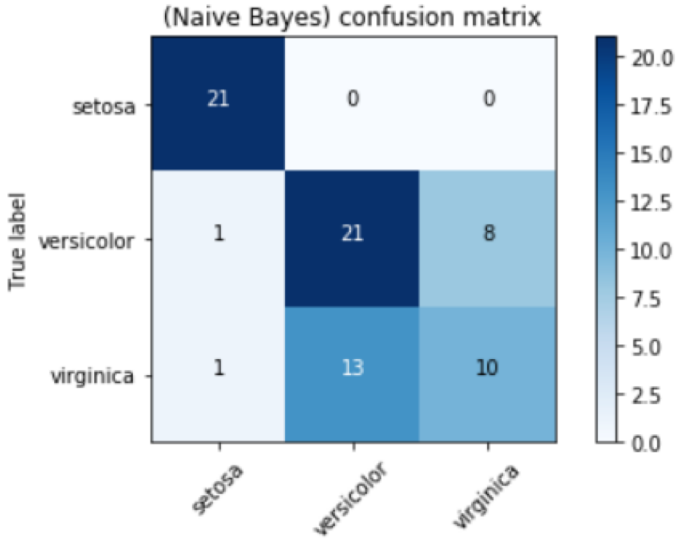
```
# Print probabilities
printProb(draw('spades', deck))
printProb(draw('spades', deck))
printProb(draw('spades', deck))
printProb(draw('hearts', deck))
printProb(draw('hearts', deck))
printProb(draw('hearts', deck))
```

```
25.0%
23.53%
22.0%
26.53%
25.0%
23.4%
```

실습 2. Naïve Bayes (scikit-learn)

- Iris data set classification

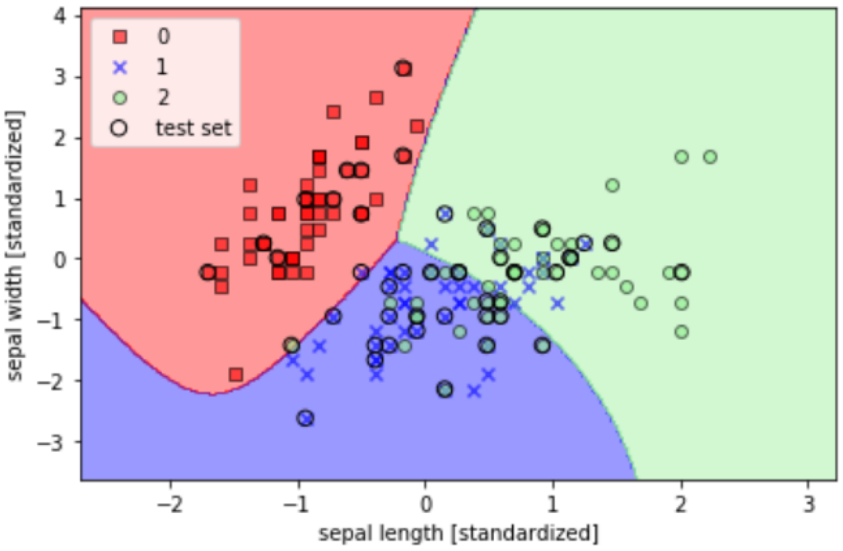
```
Classification Report
             precision    recall  f1-score   support

     setosa       0.91      1.00      0.95        21
 versicolor       0.62      0.70      0.66        30
  virginica       0.56      0.42      0.48        24

avg / total       0.68      0.69      0.68        75


 Accuracy
0.693333333333
```



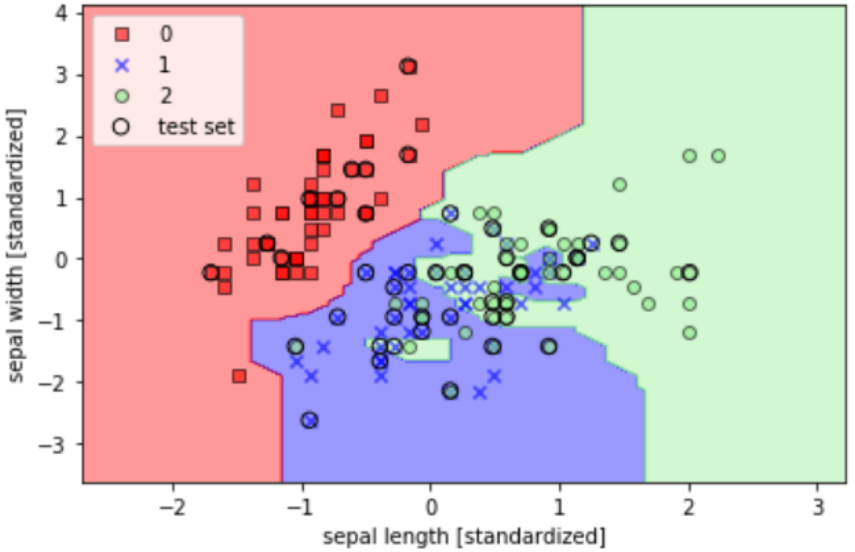
실습 2. Naïve Bayes (scikit-learn)

- Plot Decision Regions

Decision Region of Naive Bayes



Decision Region of KNN



실습 3. Custom Naïve Bayes Implementation

Bayesian Classifier Example

No.	age	income	student	credit_rating	buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31…40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31…40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31…40	medium	no	excellent	yes
13	31…40	high	yes	fair	yes
14	>40	medium	no	excellent	no

실습 3. Custom Naïve Bayes Implementation

```python
def getPredictionBayes(dataSet, X):
    classProb = {
        'yes': 1,
        'no': 1
    }
    numberOfData = len(dataSet)

    for classValue in classProb.keys():
        probability = 1

        naiveBayesPrint(classValue)

        # (3) = P(Yes) or P(No)
        classCount = countClass(classValue, dataSet)
        p = ()
        probability *= p
        print("%.5f" % p, end='')

        for i in range(len(X)):
            value = X[i]
            # ex) P(<30|Yes)
            # Fill (4) using count function
            p = (4)
            print(" * %.5f" % p, end='')
            probability *= p
        print("\n = %.5f" % probability)

        classProb[classValue] = probability

    bestClass, bestProb = None, -1

    for classValue, probability in classProb.items():
        if (5):
            bestProb = probability
            bestClass = classValue

    return bestClass, bestProb
```

```python
# get number of class in the dataSet
def countClass(classValue, dataSet):
    n = 0
    (1)

    return n

# get number of value in i^th column
def count(classValue, i, value, dataSet):
    n = 0
    (2)

    return n
```

Classification Results

```
X:  ['<=30', 'medium', 'yes', 'fair']


P(yes|X) = P(yes) * P(X|yes) =
P(yes) * P(<=30|yes) * P(medium|yes) * P(yes|yes) * P(fair|yes)
0.64286 * 0.22222 * 0.44444 * 0.66667 * 0.66667
 = 0.02822


P(no|X) = P(no) * P(X|no) =
P(no) * P(<=30|no) * P(medium|no) * P(yes|no) * P(fair|no)
0.35714 * 0.60000 * 0.40000 * 0.20000 * 0.40000
 = 0.00686


###################################
X is classified to "yes"
probability = 0.02822
###################################
```

실습 4. Understanding Generative Model