uOttawa

Faculté de génie
Faculty of Engineering

Bayesian Decision Assignment 2

Applied Machine Learning ELG5255[EG]

Group name: Group 6

Ahmed Fares Saad Eldin Khalifa

Abdallah Medhat Mohamed Rashed

Contents

1. Part 1

1.1 Class 0

number	Sepal length	Sepal width	petal length	petal width
0	5.1	3.4	1.5	0.2
1	5.0	3.4	1.5	0.2
2	4.8	3.0	1.4	0.1
3	5.0	3.3	1.4	0.2
mean	4.975	3.275	1.45	0.175
variance	**0.011875**	**0.026875**	**0.0025**	**0.001875**

1. Mean (μ) =$x1$+$x2$+………. +xn/n

Mean (sepal length) =5.1+5.0+4.8+5.0/4=4.975

Mean (sepal width) =3.4+3.4+3.0+3.3/4=3.275

Mean (petal length) =1.5+1.5+1.4+1.4/4=1.45

Mean (petal width) =0.2+0.2+0.1+0.2/4=0.175

2. variance (σ2) =∑ (xi− μ)2/n

variance (sepal length) = (5.1 - 4.975)2+ (5.0 - 4.975)2+ (4.8 - 4.975)2 + (5.0 - 4.975)2/4=0.011875

variance (sepal width) = (3.4 - 3.275)2+ (3.4 - 3.275)2+ (3.0 - 3.275)2+ (3.3 - 3.275)2/4=0.026875

variance (petal length) = (1.5 - 1.45)2+ (1.5 - 1.45)2+ (1.4 - 1.45)2+ (1.4 - 1.45)2/4=0.0025

variance (petal width) = (0.2- 0.175)2+ (0.1 - 0.175)2+ (0.2 - 0.175)2+ (0.2 - 0.175)2/4=0.001875

3. likelihood

sepal length	sepal width	**petal length**	**petal width**
5.7	2.8	4.5	1.3
5.4	3.9	1.3	0.4

- Likelihood for first row

P($\boldsymbol{x}$) = (1/ ($\boldsymbol{2\pi}$ σ2)1/2) × exp [-(x- μ)2/2 ×σ2]

P(x) sepal length = (1/ ($\mathbf{2}$× $\boldsymbol{\pi}$ × 0.011875)1/2) × exp [-(5.7- 4.975)2/2 × 0.011875] =8.953 ×10-10

P(x) sepal width = (1/($\mathbf{2}$× $\boldsymbol{\pi}$ × 0.026875)1/2) × exp[-(2.8- 3.275)2/2 × 0.026875] =0.037

P(x) petal length = (1/ (2× $\boldsymbol{\pi}$ × 0.0025)1/2) × exp [-(4.5- 1.45)2/2 × 0.0025] =0

P(x) petal width = (1/ ($\mathbf{2}$× $\boldsymbol{\pi}$ × 0.001875)1/2) × exp [-(1.3- 0.175)2/2 × 0.001875] =0

Total likelihood for first row =

P(x) sepal length × P(x) sepal width × P(x) petal length × P(x) petal width =

8.953 ×10-10× 0.037×0 ×0 = 0

- Likelihood for second row

P($\boldsymbol{x}$) = (1/ ($\mathbf{2}\boldsymbol{\pi}$ σ2)1/2) × exp[-(x- μ)2/2 ×σ2]

P(x) sepal length = (1/ ($\mathbf{2}$× $\boldsymbol{\pi}$ × 0.011875)1/2) × exp [-(5.4- 4.975)2/2 × 0.011875] =1.823 ×10-3

P(x) sepal width = (1/ ($\mathbf{2}$× $\boldsymbol{\pi}$ × 0.026875)1/2) × exp [-(3.9- 3.275)2/2 × 0.026875] =1.698 × 10-3

P(x) petal length = (1/ (2× $\boldsymbol{\pi}$ × 0.0025)1/2) × exp [-(1.3- 1.45)2/2 × 0.0025] =0.089

P(x) petal width = (1/ ($\mathbf{2}$× $\boldsymbol{\pi}$ × 0.001875)1/2) × exp [-(0.4- 0.175)2/2 × 0.001875] =1.263× 10-5

Total likelihood for second row =

P(x) sepal length × P(x) sepal width × P(x) petal length × P(x) petal width =

1.823 ×10-3× 1.698 × 10-3×0.089 ×1.263× 10-5 = 3.479× 10-12

1.2 Class 1

number	Sepal length	Sepal width	petal length	petal width
4	4.9	2.4	3.3	1.0
5	5.7	3.0	4.2	1.2
6	5.4	3.0	4.5	1.5
7	5.6	2.5	3.9	1.1
mean	5.4	2.725	3.975	1.2
variance	0.095	0.076875	0.196875	0.035

1. Mean (μ) =x1+x2+…………. +xn/n

Mean (sepal length) =4.9+5.7+5.4+5.6/4=5.4

Mean (sepal width) =2.4+3.0+3.0+2.5/4=2.725

Mean (petal length) =3.3+4.2+4.5+3.9/4=3.975

Mean (petal width) =1.0+1.2+1.5+1.1/4=1.2

2. variance (σ2) =∑ (xi− μ)2/n

variance (sepal length) = (4.9 - 5.4)2+ (5.7 - 5.4)2+ (5.4 - 5.4)2 + (5.6 - 5.4)2/4 = 0.095

variance (sepal width) = (2.4 - 2.725)2+ (3.0 - 2.725)2+ (3.0 - 2.725)2+ (2.5 - 2.725)2/4 = 0.076875

variance (petal length) = (3.3 - 3.975)2+ (4.2 - 3.975)2+ (4.5 - 3.975)2+ (3.9 - 3.975)2/4 = 0.196875

variance (petal width) = (1.0- 1.2)2+ (1.2 - 1.2)2+ (1.5 - 1.2)2+ (1.1 - 1.2)2/4 = 0.035

3. Likelihood

sepal length	sepal width	**petal length**	**petal width**
5.7	2.8	4.5	1.3
5.4	3.9	1.3	0.4

- Likelihood for first row
  P(***x***) = (1/(**2*π*** σ2)1/2) × exp[-(x- μ)2/2 ×σ2]

  P(x) sepal length = (1/ (**2**× ***π*** × 0.095)1/2) × exp [-(5.7- 5.4)2/2 × 0.095] =0.806

  P(x) sepal width = (1/ (**2**× ***π*** × 0.076875)1/2) × exp [-(2.8- 2.725)2/2 × 0.076875] =1.387

  P(x) petal length = (1/(2× ***π*** × 0.196875)1/2) × exp[-(4.5- 3.975)2/2 × 0.196875] =0.446

  P(x) petal width = (1/ (**2**× ***π*** × 0.035)1/2) × exp [-(1.3- 1.2)2/2 × 0.035] =1.849

  Total likelihood for first row =

  P(x) sepal length × P(x) sepal width × P(x) petal length × P(x) petal width =

  0.806× 1.387×0.446 ×1.849 = 0.922

- Likelihood for second row
  P(***x***) = (1/(**2*π*** σ2)1/2) × exp[-(x- μ)2/2 ×σ2]

  P(x) sepal length = (1/ (**2**× ***π*** × 0.095)1/2) × exp [-(5.4- 5.4)2/2 × 0.095] =1.294

  P(x) sepal width = (1/ (**2**× ***π*** × 0.076875)1/2) × exp [-(3.9- 2.725)2/2 × 0.076875] = 1.812 × 10-4

  P(x) petal length = (1/ (2× ***π*** × 0.196875)1/2) × exp [-(1.3- 3.975)2/2 × 0.196875] = 1.152× 10-8

P(x) petal width = (1/ (**2**× $\boldsymbol{\pi}$ × 0.035)1/2) × exp [-(0.4- 1.2)2/2 × 0.035] =2.281× 10-4

Total likelihood for second row =

P(x) sepal length × P(x) sepal width × P(x) petal length × P(x) petal width =

1.294× 1.812 × 10-4×1.152× 10-8×2.281× 10-4= 6.161× 10-16

1.3 Class 2

number	Sepal length	Sepal width	petal length	petal width
8	6.5	3.0	5.8	2.2
9	7.7	2.6	6.9	2.3
mean	7.1	2.8	6.35	2.25
variance	0.36	0.04	0.3025	0.0025

1. Mean (μ) =x1+x2+………… +xn/n

   Mean (sepal length) =6.5+7.7/2=7.1

   Mean (sepal width) =3.0+2.6/2=2.8

   Mean (petal length) =5.8+6.9/2=6.35

   Mean (petal width) =2.2+2.3/2=2.25

2. variance (σ2) =∑ (xi− μ)2/n

   variance (sepal length) = (6.5 – 7.1)2+ (7.7 – 7.1)2/2=0.36

   variance (sepal width) = (3.0 – 2.8)2+ (2.6 - 2.8)2/2=0.04

   variance (petal length) = (5.8 – 6.35)2+ (6.9 – 6.35)2/2=0.3025

   variance (petal width) = (2.2- 2.25)2+ (2.3 -2.25)2/2=0.0025

3. Likelihood

sepal length	sepal width	**petal length**	**petal width**
5.7	2.8	4.5	1.3
5.4	3.9	1.3	0.4

- Likelihood for first row

  P($\boldsymbol{x}$) = (1/ (**2**$\boldsymbol{\pi}$ σ2)1/2) × exp [-(x- μ)2/2 ×σ2]

  P(x) sepal length = (1/ (**2**× $\boldsymbol{\pi}$ × 0.36)1/2) × exp [-(5.7- 7.1)2/2 × 0.36] =0.044

  P(x) sepal width = (1/ (**2**× $\boldsymbol{\pi}$ × 0.04)1/2) × exp [-(2.8- 2.8)2/2 × 0.04] =1.994

P(x) petal length = (1/(2× $\boldsymbol{\pi}$ × 0.3025)1/2) × exp[-(4.5- 6.35)2/2 × 0.3025] =2.534×10-3

P(x) petal width = (1/(**2**× $\boldsymbol{\pi}$ × 0.0025)1/2) × exp[-(1.3- 2.25)2/2 × 0.0025] =3.249×10-78

Total likelihood for first row =

P(x) sepal length × P(x) sepal width × P(x) petal length × P(x) petal width =

0.044× 1.994×2.534×10-3×3.249×10-78=7.223 ×10-82

- Likelihood for second row
  P($\boldsymbol{x}$) = (1/ (**2**$\boldsymbol{\pi}$ σ2)1/2) × exp [-(x- μ)2/2 ×σ2]

  P(x) sepal length = (1/ (**2**× $\boldsymbol{\pi}$ × 0.36)1/2) × exp [-(5.4- 7.1)2/2 × 0.36] =0.012

  P(x) sepal width = (1/ (**2**× $\boldsymbol{\pi}$ × 0.04)1/2) × exp [-(3.9- 2.8)2/2 × 0.04] =5.385 × 10-7

  P(x) petal length = (1/ (2× $\boldsymbol{\pi}$ × 0.3025)1/2) × exp [-(1.3- 6.35)2/2 × 0.3025] =3.579× 10-19

  P(x) petal width = (1/ (**2**× $\boldsymbol{\pi}$ × 0.0025)1/2) × exp [-(0.4- 2.25)2/2 × 0.0025] =0

  Total likelihood for second row =

  P(x) sepal length × P(x) sepal width × P(x) petal length × P(x) petal width =

  0.012× 5.385 × 10-7×3.579× 10-19×0= 0

After calculate total likelihood for every row in every class, Calculate prior for every class.

4. Prior

   Prior (class0) = 4/10 =0.4

   Prior (class1) = 4/10 =0.4

   Prior (class2) = 2/10 =0.2

5. $posterior$ for every class in every row

   Posterior ($class$) = $Piror \times likelihood$

- Posterior for row 1
  Posterior (class 0) = 0.4 × 0=0

  Posterior (class 1) = 0.4 × 0.922 =0.3688

  Posterior (class 2) = 0.2 × 7.223 ×10-82=1.444 × 10-82

After calculate posterior for every class in row 1, Obtain first row equal class 1 because of posterior of class 1 is bigger than posterior of class 0 and class 2.

- Posterior for row 2
  Posterior (class 0) = 0.4 × 3.479× 10-12 = 1.3916 × 10-12

  Posterior (class 1) = 0.4 × 6.161× 10-16 = 2.464 × 10-16

  Posterior (class 2) = 0.2 × 0 = 0

After calculate posterior for every class in row 2, Obtain first row equal class 0 because of posterior of class 0 is bigger than posterior of class 1 and class 2.

sepal length	sepal width	**petal length**	**petal width**	**label**
5.7	2.8	4.5	1.3	1
5.4	3.9	1.3	0.4	0

In this part of the assignment, we didn't use the evidence in posterior equation because of comparing the classes.

2. Part 2

2.1 Load Iris Dataset

```
# Load Iris Dataset
iris = datasets.load_iris()
X, y = iris.data, iris.target
trX, teX, trY, teY = train_test_split(X, y, random_state=0)
```

Figure 1: loading the dataset

2.2 Drop the petal length and petal width features to form a 2D Iris dataset

```
# Drop the petal length and petal width features to form a 2D Iris dataset
x=X[:,0:2]
```

Figure 2:drop features from data

We drop the petal length and width features from the data which are in the second and third columns in our data. So we will use data of the special length and width.

2.3 Apply Naïve Bayes Classifier to get training and testing accuracy

```
#Apply Naïve Bayes Classifier to get training and testing accuracy (after droping the petal length and petal width)
clf_nai_drop=GaussianNB()
plot_accuracy(clf_nai_drop, x_train, x_test, y_train, y_test)

predict Values:
 [1 1 0 2 0 2 0 2 2 1 1 2 1 2 1 0 1 1 0 0 1 1 0 0 1 0 0 1 1 0]

Accuracy for model in training data:  82.5
Accuracy for model in testing data:  73.33333333333333
```

Figure 3apply Naive Bayes Classifier

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	0.69	0.69	0.69	13
2	0.33	0.33	0.33	6
accuracy			0.73	30
macro avg	0.68	0.68	0.68	30
weighted avg	0.73	0.73	0.73	30

Figure 4 report on Naive Bayes classifier

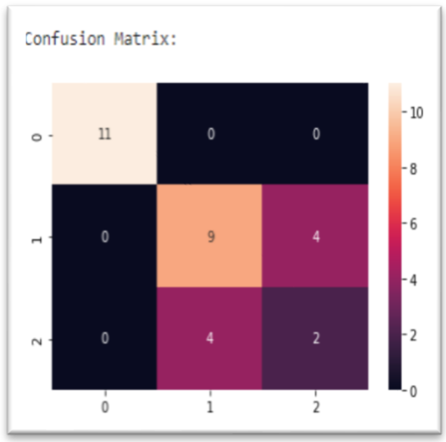


Figure 5:confusion Matrix on Naive Bayes classifier

2.4 Tune hyperparameters of Naive Bayes Classifier (i.e., var_smoothing). Try var_smoothing as 1e-9, 1e-8, 1e-7. Plot accuracy vs var_smoothing curve for training and testing set.

```
acc_train=[]
acc_test=[]
var_smoothing=[1e-9, 1e-8, 1e-7,2e-7,3e-7]
for i in range(len(var_smoothing)):
    clf_nai_tune=GaussianNB(var_smoothing=var_smoothing[i])
    clf_nai_tune.fit(x_train, y_train)
    y_pred=clf_nai_tune.predict(x_test)
    acc_train.append(clf_nai_tune.score(x_train, y_train)*100)
    acc_test.append(clf_nai_tune.score(x_test, y_test)*100)
#Plot accuracy vs var_smoothing curve for training set
plot_accVSvarsmooth(var_smoothing, acc_train)
```

Figure 6 tuning hyperparameters

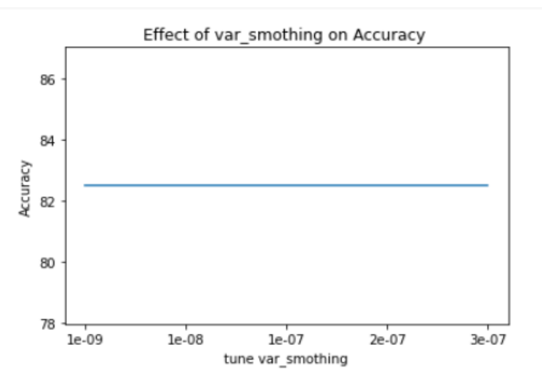


Figure 7: var_smothing with acc train

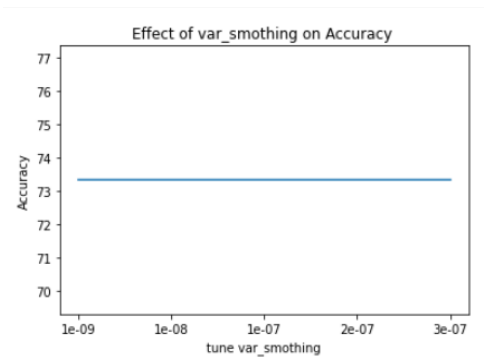


Figure 8: var_smothing with acc test

We applied tuning hyperparameters on var_smothing to see it's effect on the accuracy and we found that when changing the parameter values, it didn't affect on the accuracy of naïve bayes classifier.

2.5 Develop Risk-based Bayesian Decision Theory Classifier (RBDTC)

```
class BayesianDecisionTheoryClassifier(BaseEstimator, ClassifierMixin):
    def __init__(self, estimator, utilityMat):
        self.estimator = estimator
        self.utilityMat = utilityMat

    def fit(self, X, y):
        check_X_y(X,y)
        self.classes_ = np.unique(y)
        self.estimator_ = clone(self.estimator).fit(X, y)
        return self

    def predict_proba(self, X):
        check_is_fitted(self)
        prob = self.estimator_.predict_proba(X)
        probList = [(prob * self.utilityMat[index]).sum(axis=1).reshape((-1, 1))
                    for index, c in enumerate(self.classes_)]
        prob = np.hstack(probList)
        return prob

    def predict(self, X):
        pred = self.predict_proba(X).argmin(axis=1)
        return self.classes_[pred]
```

Figure 9:Bayesian Decision Theory Classifier

2.6 Apply Risk-based Bayesian Decision Theory Classifier which takes Naïve Bayes Classifier as base estimator and uses Table 3 as risk matrix

```
utilityMat = np.array([
    [-10, -5, -5],
    [-5, -10, -5],
    [-5, -5, -100],
])
```

```
bdtc = BayesianDecisionTheoryClassifier(clf_nai_drop, utilityMat)
```

```
plot_accuracy(bdtc, x_train, x_test, y_train, y_test)
```

```
predict Values:
  [2 2 0 2 0 2 0 2 2 2 2 2 2 2 2 0 2 2 0 0 2 2 0 0 2 0 0 2 1 0]
```

Figure 10: Utility Matrix

```
bdtc_text = BayesianDecisionTheoryClassifier(clf_nai_drop, utilityMat)
plot_accuracy(bdtc_text, x_train, x_test, y_train_names,y_test_names)


predict Values:
  ['virginica' 'virginica' 'setosa' 'virginica' 'setosa' 'virginica'
 'setosa' 'virginica' 'virginica' 'virginica' 'virginica' 'virginica'
 'virginica' 'virginica' 'virginica' 'setosa' 'virginica' 'virginica'
 'setosa' 'setosa' 'virginica' 'virginica' 'setosa' 'setosa' 'virginica'
 'setosa' 'setosa' 'virginica' 'versicolor' 'setosa']
```

Figure 11predicted value with class names

As mentioned in the previous figures we applied Bayesian Decision Theory Classifier by passing the naïve bayes model to it and the utility matrix, and we check the input by "utils. Check_X_Y" to check the input validation, and "check_is_fitted" to check the fit of model. The two objects of class were applied one for producing the number of classes to use them for plotting the decision boundary and the other for producing the classes name by using function (predict_values) in the next figures. We didn't Plot decision boundary for classes name that because the decision boundary needs numbers instead of string for plotting, but the accuracy semantically the same. Plot decision boundary and calculate precision, recall and accuracy for training and testing set.

```
def predict_values(y):
    pre=[]
    for i in range(len(y)):
        if y[i]==0:
            pre.append('setosa')
        elif y[i]==1:
            pre.append('versicolor')
        elif y[i]==2:
            pre.append('virginica')
    return pre
```

Figure 12: predict function convert the target data

```
y_train_names = predict_values(y_train)
y_test_names = predict_values(y_test)
```

Figure 13: use the predicted fun

2.7 Plot decision boundary and calculate precision, recall and accuracy for training and testing set

```
Accuracy for model in training data:  65.83333333333333
Accuracy for model in testing data:  60.0

Classification Report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       1.00      0.08      0.14        13
           2       0.33      1.00      0.50         6

    accuracy                           0.60        30
   macro avg       0.78      0.69      0.55        30
weighted avg       0.87      0.60      0.53        30
```

Figure 14: classification report for RBDTC

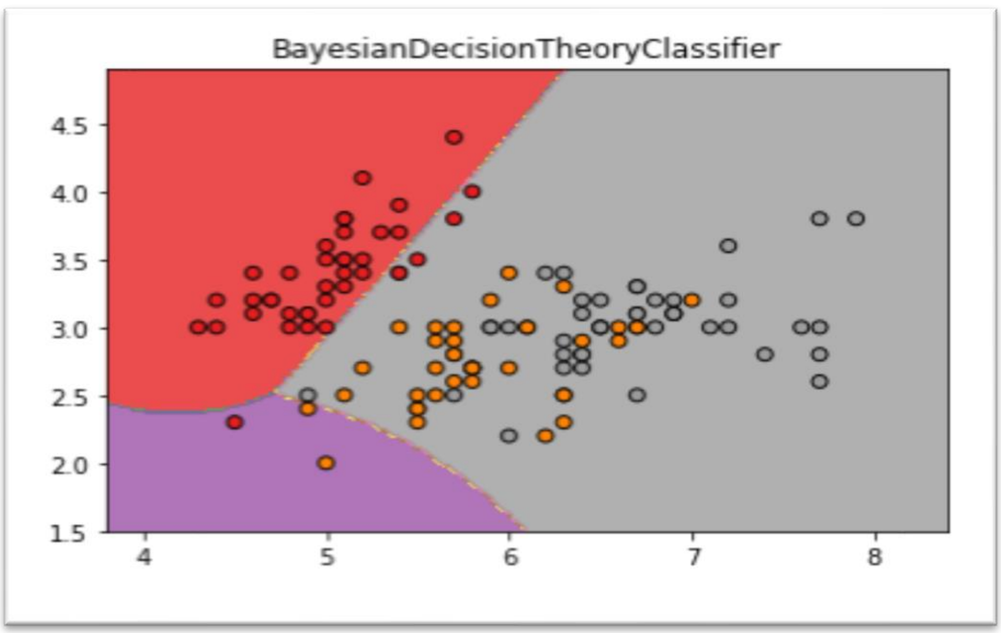
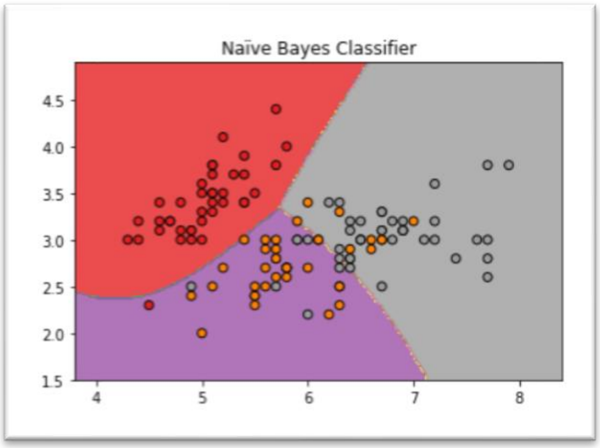


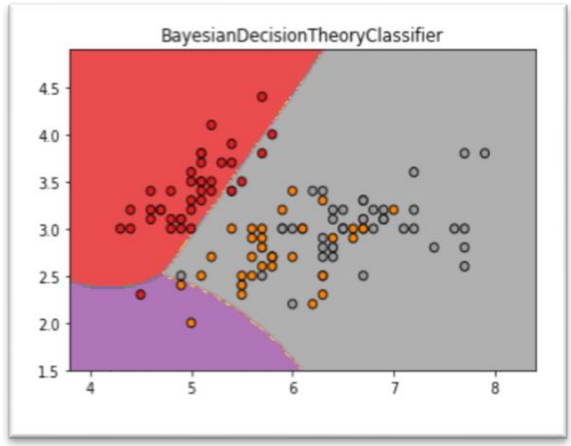
Figure 15: Decision boundary for RBDTC

We calculate the recall, precision, and accuracy of RBDTC from the classification report:
accuracy = 60 %
precision = 87%

recall= 60%

2.8 Compare and analysis the performance between NB and RBDTC regarding to their decision boundary, precision, recall, and accuracy






```
Accuracy for model in training data:  82.5
Accuracy for model in testing data:  73.33333333333333

Classification Report:
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	0.69	0.69	0.69	13
2	0.33	0.33	0.33	6
accuracy			0.73	30
macro avg	0.68	0.68	0.68	30
weighted avg	0.73	0.73	0.73	30

```
Accuracy for model in training data:  65.83333333333333
Accuracy for model in testing data:  60.0

Classification Report:
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	1.00	0.08	0.14	13
2	0.33	1.00	0.50	6
accuracy			0.60	30
macro avg	0.78	0.69	0.55	30
weighted avg	0.87	0.60	0.53	30

After applying Naïve Bayes Classifier from sklearn we obtain that the accuracy and recall are higher than using Risk-based Bayesian Decision Theory Classifier as mentioned in the previous figures.

3. Conclusion

In conclusion, this report could be summarized by loading the data from Iris dataset, after we dropped the two features and split the data, then applied Naïve Bayes Classifier and tuning hyperparameters of var smoothing in it. After applied the tuning hyperparameters for the var_smothing on Naïve Bayes, we notice that the var_smothing didn't affect the accuracy of Naïve Bayes. then we applied Risk-based Bayesian Decision Theory Classifier and comparing the plot of decision boundary between NB Classifier and RBDTC. And we get higher performance by using the NB classifier with accuracy (73.33%). Finally, we gain experience to reduce error in prediction for specific class" virginica" such as when we use that model to classify cancer in medical field and predict classes name such as "Setosa, Versicolour, Virginica" and classes number such as "0, 1, 2".