Lecture 10

K-Nearest Neighbor Methods for Density Estimation

F1 score

F1 score (combing Sensitivity and Precision via a *harmonic mean*)

$$F_1 = 2 \cdot \frac{\text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}} = 2 \cdot \frac{\frac{TP}{TP+FN} \cdot \frac{TP}{TP+FP}}{\frac{TP}{TP+FN} + \frac{TP}{TP+FP}}$$

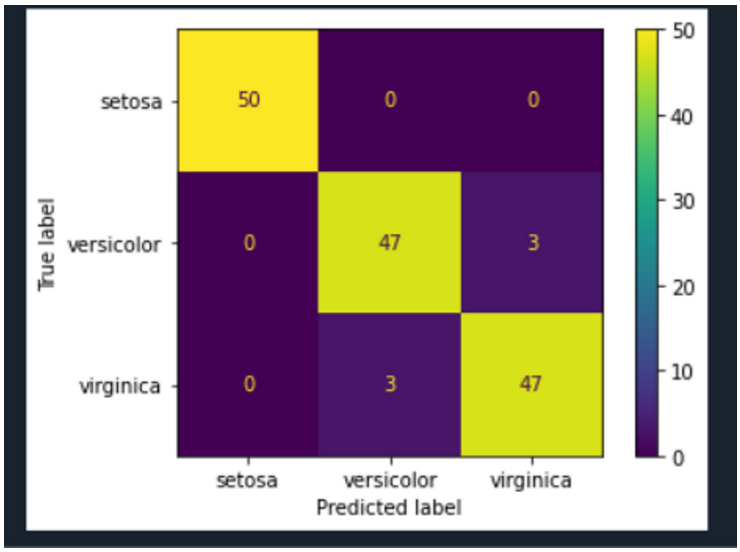
$$= \frac{2 \cdot \frac{TP^2}{(TP+FN)(TP+FP)}}{TP \cdot \frac{TP+FP+TP+FN}{(TP+FN)(TP+FP)}} = \frac{2TP}{2TP+FP+FN}$$

$$= \frac{TP}{TP + \frac{FP+FN}{2}}$$

Mean of the False Samples

"""
'naive_bayes_demo_**confusion_matrix**.py'
"""

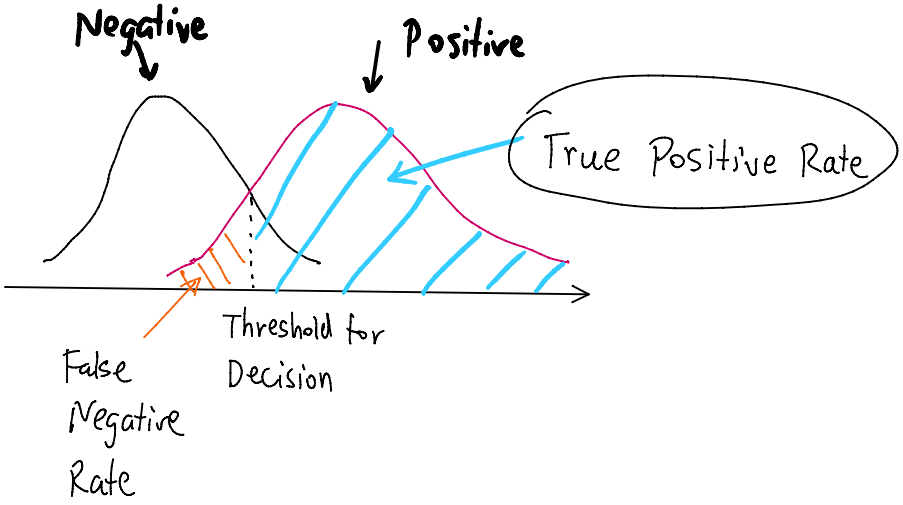
```
import numpy as np
from sklearn.naive_bayes import GaussianNB
clf = GaussianNB()

from sklearn.datasets import load_iris
train_samples = load_iris()
X = train_samples.data
Y = train_samples.target

clf.fit(X, Y)

clf.score(X, Y)

# Verify the classification accuracy
Y_pred = clf.predict(X)

num_correct = np.sum(Y == Y_pred)
num_sample = np.size(Y_pred)
num_correct/num_sample
```

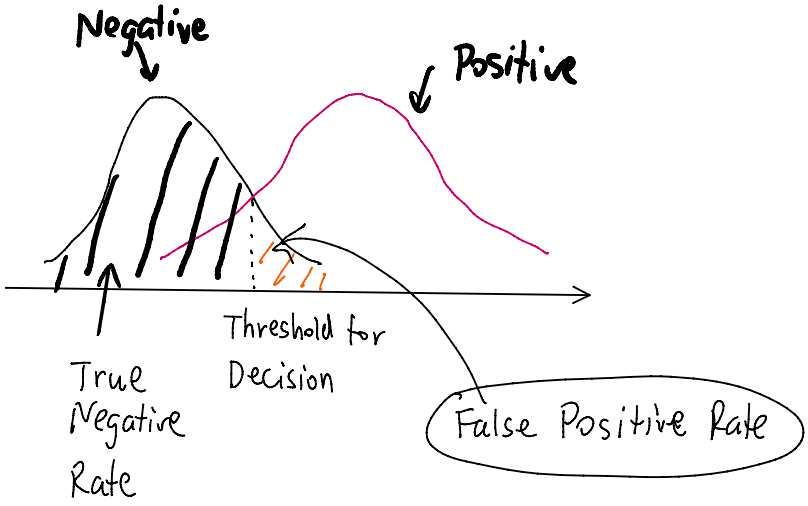


	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	50
versicolor	0.94	0.94	0.94	50
virginica	0.94	0.94	0.94	50
accuracy			0.96	150
macro avg	0.96	0.96	0.96	150
weighted avg	0.96	0.96	0.96	150

```
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
cm = confusion_matrix (Y,Y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix = cm,display_labels = train_samples.target_names)
import matplotlib.pyplot as plt
disp.plot()
plt.show()

from sklearn.metrics import classification_report
print(classification_report (Y, Y_pred, target_names = train_samples.target_names))
```

ROC curve:


Negative
Positive
True Positive Rate
Threshold for Decision
False Negative Rate


Negative
Positive
Threshold for Decision
True Negative Rate
False Positive Rate