written3

October 29, 2024

1 STP598 Machine Learning & Deep Learning

1.1 Written Assignment 3

1.1.1 Due 11:59pm Sunday Nov. 10, 2024 on Canvas

1.1.2 name, id

[1]:
```
import warnings
import numpy as np
import matplotlib.pyplot as plt
```

1.2 Question 1 Naive Bayes

Recall that the optimal decision rule for naive Bayes is

$$\arg\max_k \Pr[Y = k|X = x] = \arg\max_k \pi_k f_k(x) \tag{1}$$

where π_k is the prior class probability $\Pr[Y = k]$ and $f_k(x)$ is the conditional probability $\Pr[X = x|Y = k]$ that is approximated by $f_k(x) = \prod_{j=1}^{p} f_{kj}(x_j)$.

The following dataset contains loan information and can be used to try to predict whether a borrower will default (the last column is the classification). Use the naive Bayes method to determine whether a loan $X = (HomeOwner = No, MaritalStatus = Married, Income = High)$ should be classified as a Defaulted Borrower or not. So, determine which is larger, $\Pr[Yes|X]$ or $\Pr[No|X]$.

Hint: Both π_k and $f_{kj}(x_j)$ can be computed empirically based on the following table. For example, $\Pr[Yes] = 0.3$ and $\Pr[HomeOwner = No|No] = 4/7$.

Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	High	No
2	No	Married	High	No
3	No	Single	Low	No
4	Yes	Married	High	No
5	No	Divorced	Low	Yes
6	No	Married	Low	No
7	Yes	Divorced	High	No
8	No	Single	Low	Yes
9	No	Married	Low	No

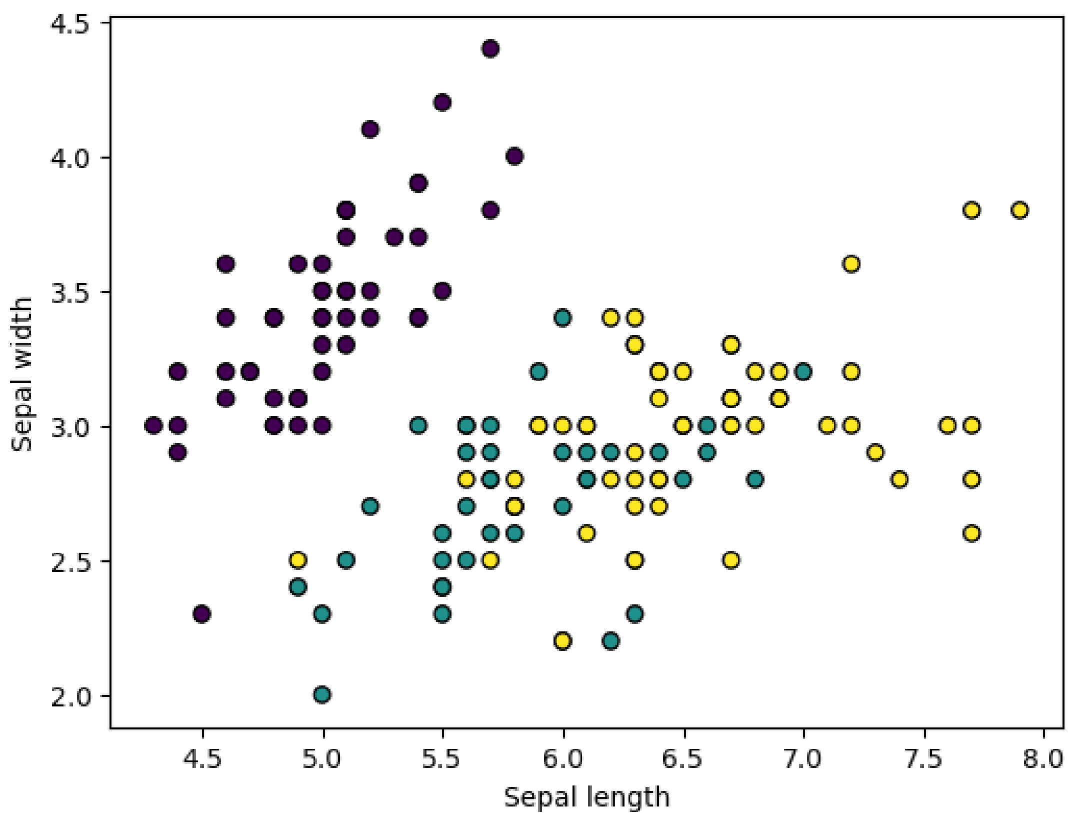
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
10	No	Single	Low	Yes

1.3 Question 2 Classification on iris data

Consider iris data and classify them with the first two features using different methods. Let's load the data.

[3]:
```
from sklearn import datasets

iris = datasets.load_iris()
X = iris.data[:,:2]
y = iris.target
K=len(np.unique(y))
plt.scatter(X[:, 0], X[:, 1], c=plt.get_cmap('viridis',K).colors[y],␣
  ↪edgecolor='k',cmap=plt.get_cmap('viridis'))
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.show()
```



Fit RandomForest and GaussianProcess models to the iris data. For RandomForest, use **50 ran-**

dom trees (`n_estimators`). For GaussianProcess, use anisotropic **Radial-basis function** kernel (aka squared-exponential kernel, `RBF`). Save the fitted models as **rfc** and **gpc** respectively.

Note! Plesae set `random_state=2024` in RandomForest

[ ]:
```
# Fit RandomForest and GaussianProcess
```

Now compare their decision boudaries in the following plot.

[ ]:
```
# plot decision boundairs of RandomForest and GaussianProcess
```

1.4 Question 3 Backpropagation

Assume we have the following neural network for classification:

$$\begin{aligned} z &= Wx^{(i)} + b \\ \hat{y}^{(i)} &= \sigma(z) := \frac{1}{1+e^{-z}} \\ L^{(i)} &= y^{(i)} * \log(\hat{y}^{(i)}) + (1-y^{(i)}) * \log(1-\hat{y}^{(i)}) \\ J &= -\frac{1}{m}\sum_{i=1}^{m} L^{(i)} \end{aligned} \tag{2}$$

- What is $\frac{\partial J}{\partial \hat{y}^{(i)}}$?
- What is $\frac{\partial J}{\partial b}$?
- (bonus) What is $\frac{\partial J}{\partial W}$?