Yu, Zhang (25238743)

Q1: new document X_* A:1 B:0 C:0 D:2

(a): Multinominal:

E^+: A:5 B:3 C:9 D:6 All: 23

E^-: A:11 B:3 C:0 D:2. All: 16

$\therefore p\theta^+ = (\frac{5}{23}, \frac{3}{23}, \frac{9}{23}, \frac{6}{23})$

$p\theta^- = (\frac{11}{16}, \frac{3}{16}, \frac{0}{16}, \frac{2}{16})$

$\therefore p(x_*^+|\theta^+) = \frac{(\frac{5}{23})^1}{1!} \cdot \frac{(\frac{3}{23})^0}{0!} \cdot \frac{(\frac{9}{23})^0}{0!} \cdot \frac{(\frac{6}{23})^2}{2!} \times 3! = 0.0443$

$\therefore p(x_*^-|\theta^-) = \frac{(\frac{11}{16})^1}{1!} \cdot \frac{(\frac{3}{16})^0}{0!} \cdot \frac{(\frac{0}{16})^0}{0!} \cdot \frac{(\frac{2}{16})^2}{2!} \cdot 3! = 0.0322$

$\therefore p(x_*^+|\theta^+) > p(x_*^-|\theta^-)$

$\therefore X_*$ should be classified as positive

$p(x_*) = p(x_*|\theta^+) \cdot p(\theta^+) + p(x_*|\theta^-) \cdot p(\theta^-)$

$= 0.0443 \times \frac{4}{8} + 0.0322 \times \frac{4}{8} = 0.0382$

$p(+|x_*) = \frac{p(x_*|\theta^+) \cdot p(\theta^+)}{p(x_*)} = \frac{0.0443 \times \frac{4}{8}}{0.0382} = 0.579$

(b). (add-1) use smooth probabilities.

$p\theta^+ = (\frac{6}{27}, \frac{4}{27}, \frac{10}{27}, \frac{7}{27})$

$p\theta^- = (\frac{12}{20}, \frac{4}{20}, \frac{1}{20}, \frac{3}{20})$

$$\therefore P(x_*|\theta^+) = \frac{\left(\frac{6}{27}\right)^1}{1!} \cdot \frac{\left(\frac{4}{27}\right)^0}{0!} \cdot \frac{\left(\frac{10}{27}\right)^0}{0!} \cdot \frac{\left(\frac{7}{27}\right)^2}{2!} \times 3! = 0.0448$$

$$\therefore P(x_*|\theta^-) = \frac{\left(\frac{12}{20}\right)^1}{1!} \cdot \frac{\left(\frac{4}{20}\right)^0}{0!} \cdot \frac{\left(\frac{1}{20}\right)^0}{0!} \cdot \frac{\left(\frac{3}{20}\right)^2}{2!} \cdot 3! = 0.0405$$

$$P(x_*) = P(x_*|\theta^+) \cdot P(\theta^+) + P(x_*|\theta^-) \cdot P(\theta^-)$$
$$= 0.0448 \times \frac{4}{8} + 0.0405 \times \frac{4}{8} = 0.0426$$

$$\therefore P(-|x_*) = \frac{P(x_*|\theta^-) \cdot P(\theta^-)}{P(x_*)} = \frac{0.0405 \times \frac{4}{8}}{0.0426} = 0.4748$$

(c) Multivariate Bernolli:

Doucument	A	B	C	D	class
1	1	0	1	1	+
2	0	1	1	0	+
3	1	0	0	1	+
4	0	0	1	0	+
5	0	0	0	1	-
6	1	0	0	0	-
7	1	1	0	0	-
8	1	0	0	1	-

E^+: A: 2 B: 1 C: 3 D: 2 $P(\theta^+) = (\frac{2}{4}, \frac{1}{4}, \frac{3}{4}, \frac{2}{4})$

E^-: A: 3 B: 1 C: 0 D: 2 $P(\theta^-) = (\frac{3}{4}, \frac{1}{4}, \frac{0}{4}, \frac{2}{4})$

new document x_* A: 1 B: 0 C: 0 D: 2 $\Rightarrow (1, 0, 0, 1)$

$\therefore P(x_*|\theta^+) : \frac{2}{4} \times (1-\frac{1}{4}) \times (1-\frac{3}{4}) \times \frac{2}{4} = 0.046875$

$P(x_*|\theta^-) : \frac{3}{4} \times (1-\frac{1}{4}) \times (1-\frac{0}{4}) \times \frac{2}{4} = 0.28125$

$P(x_*) = P(x_*|\theta^+) \cdot P(\theta^+) + P(x_*|\theta^-) \cdot P(\theta^-)$

$= 0.046875 \times \frac{4}{8} + 0.28125 \times \frac{4}{8} = 0.164062$

$\therefore P(+|x_*) = \frac{P(x_*|\theta^+) P(\theta^+)}{P(x_*)} = \frac{0.046875 \times \frac{4}{8}}{0.164062} = 0.142$

cd).

Document	A	B	C	D	class
1	1	0	1	1	+
2	0	1	1	0	+
3	1	0	0	1	+
4	0	0	1	0	+
5	0	0	0	1	−
6	1	0	0	0	−
7	1	1	0	0	−
8	1	0	0	1	−
9	1	1	1	1	+
10	0	0	0	0	+
11	1	1	1	1	−
12	0	0	0	0	−

E^+: A: 2 B: 1 C: 3 D: 2 $P(\theta^+) = (\frac{3}{8}, \frac{2}{8}, \frac{4}{8}, \frac{3}{8})$

E^-: A: 3 B: 1 C: 0 D: 2 $P(\theta^-) = (\frac{4}{8}, \frac{2}{8}, \frac{1}{8}, \frac{3}{8})$

new document x_* A: 1 B: 0 C: 0 D: 2 $\Rightarrow (1, 0, 0, 1)$

$P(x_*|\theta^+) = \frac{3}{8} \times (1 - \frac{2}{8}) \times (1 - \frac{4}{8}) \times \frac{3}{8} = 0.0527$

$P(x_*|\theta^-) = \frac{4}{8} \times (1 - \frac{2}{8}) \times (1 - \frac{1}{8}) \times \frac{3}{8} = 0.1230$

$P(x_*) = P(x_*|\theta^+) \cdot P(\theta^+) + P(x_*|\theta^-) \cdot P(\theta^-)$

$= 0.0527 \times \frac{4}{8} + 0.1230 \times \frac{4}{8} = 0.0879$

$\therefore P(-|x_*) = \frac{P(x_*|\theta^-) P(\theta^-)}{P(x_*)} = \frac{0.1230 \times \frac{4}{8}}{0.0879} = 0.7692$

Q2:

(a). $L_c(y, \hat{y}) = \sum_{i=1}^{n} L_c(y_i, \hat{y}_i) = \sum_{i=1}^{n} \left[\sqrt{\frac{1}{C^2}(y_i - w^T x_i)^2 + 1} - 1 \right]$

$\because \hat{y}_i = w^T x_i = w_0 + w_1 x_i$ for $i = 1, \ldots, n$.

$\therefore L_c(y, \hat{y}) = \sum_{i=1}^{n} L_c(y_i, \hat{y}_i) = \sum_{i=1}^{n} \left\{ \left\{ \frac{1}{C^2}[y_i - (w_0 + w_1 x_i)]^2 + 1 \right\}^{\frac{1}{2}} - 1 \right\}$

$$\frac{\partial L_c(y, \hat{y})}{\partial w_0} = \sum_{i=1}^{n} \frac{1}{2}\left(\frac{1}{C^2}[y_i - (w_0 + w_1 x_i)]^2 + 1\right)^{-\frac{1}{2}} \cdot 2[y_i - (w_0 + w_1 x_i)] \cdot (-1)$$

$$= \sum_{i=1}^{n} \frac{1}{C^2}(y_i - w_0 - w_1 x_i)^2 + 1)^{-\frac{1}{2}} (y_i - w_0 - w_1 x_i) \cdot (-1)$$

$$= \sum_{i=1}^{n} \frac{1}{C^2} \cdot \frac{-(y_i - w_0 - w_1 x_i)}{\sqrt{(y_i - w_0 - w_1 x_i)^2 + 1}}.$$

$$\frac{\partial L_c(y, \hat{y})}{\partial w_1} = \sum_{i=1}^{n} \frac{1}{2}\left(\frac{1}{C^2}[y_i - (w_0 + w_1 x_i)]^2 + 1\right)^{-\frac{1}{2}} \cdot 2(y_i - w_0 + w_1 x_i)) \cdot (-x_i)$$

$$= \sum_{i=1}^{n} \frac{1}{C^2}[y_i - w_0 - w_1 x_i]^2 + 1)^{-\frac{1}{2}} \cdot (y_i - w_0 - w_1 x_i) \cdot (-x_i)$$

$$= \sum_{i=1}^{n} \frac{1}{C^2} \cdot \frac{(y_i - w_0 - w_1 x_i) \cdot (-x_i)}{\sqrt{(y_i - w_0 - w_1 x_i)^2 + 1}}$$

(b) Initialise $t = 0$

$w = (w_0^t, w_1^t)$

$t = 1, 2, 3, 4, \cdots\cdots n,$

while True:

$$w_0^{(t+1)} = w_0^{(t)} - \alpha \cdot \frac{\partial L_c(y, \hat{y})}{\partial w_0}$$

$$w_1^{(t+1)} = w_1^{(t)} - \alpha \cdot \frac{\partial L_c(y, \hat{y})}{\partial w_1}$$

$$w_0^{t} = w_0^{(t+1)}$$

$$w_1^{t} = w_1^{(t+1)}$$

if change in $L(w^{(t)})$ is negligible:

break

(c). The result is:

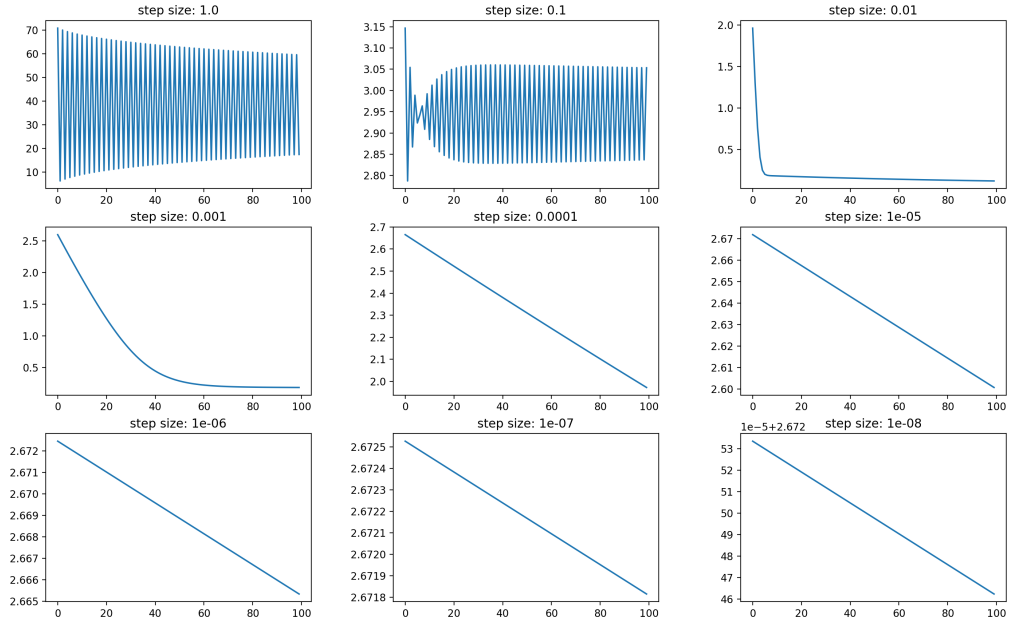


figure 2.1

```
## Question 2

import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)       # make sure you run this line for consistency
x = np.random.uniform(1, 2, 100)
y = 1.2 + 2.9 * x + 1.8 * x**2 + np.random.normal(0, 0.9, 100)
plt.scatter(x,y)
plt.show()

## (c)
c = 2
interation = 100
import math
def loss_function(c,x,y):
    result = math.sqrt((pow((x-y),2)/c/c+1))-1
    return result

def w0_cost_function(c,x,y,w0,w1):
    result =0
    i= 0
    while i < interation:
        cy = x[i]*w1 +w0
        ty = y[i]
        result = result + (-(1/pow(c,2))*(ty-cy)/(math.sqrt((1/pow(c,2))*pow((ty-cy),2)+1)))
        i = i +1
    return result
```

```
def w1_cost_function(c,x,y,w0,w1):
    result =0
    i = 0
    while i < interation:
        cy = x[i]*w1 + w0
        ty =y [i]
        result =result + (-(1/pow(c,2))*x[i]*(ty -cy)/(math.sqrt((1/pow(c,2))*pow((ty-cy),2)+1)))
        i = i +1
    return result

def update_weight_function(w0,w1,c,x,y,alphy):
    nw0 = w0-alphy *w0_cost_function(c,x,y,w0,w1)
    nw1 = w1-alphy *w1_cost_function(c,x,y,w0,w1)
    return nw0,nw1

```

```
def run_function():
    lista =[]
    i = 0
    while i < 9:
        lista.append(1*pow(10,-i))
        i = i+1

    temploss =[]
    losses =[]
    for i in lista:
        temploss =[]
        w0 =1
        w1 =1
        j = 0
        while(j<interation):
            w0,w1 = update_weight_function(w0,w1,c,x,y,i)
            two_temploss = []
            k =0
            while(k<interation):
                cy = x[k]*w1 +w0
                two_temploss.append(loss_function(c,cy,y[k]))
                k = k + 1
            temploss.append(np.mean(two_temploss))
            j = j +1
        losses.append(temploss)
    return losses

losses = run_function()
```

```


## plotting help
fig, ax = plt.subplots(3,3, figsize=(10,10))
alphas = [10e-1, 10e-2, 10e-3,10e-4,10e-5,10e-6,10e-7, 10e-8, 10e-9]
for i, ax in enumerate(ax.flat):
    # losses is a list of 9 elements. Each element is an array of length 100 storing the loss at each iteration for
    ax.plot(losses[i])
    ax.set_title(f"step size: {alphas[i]}")  # plot titles
plt.tight_layout()      # plot formatting
plt.show()
```

(d). the first two step size can not find the correct weights until the 3 step size that can be train. If step size is large, the classifier may miss the optimal solution. If step size is too small, the classifier trains will becaams slow and require more iteration

(e).

Q4:

(a): According to figure 1, the data is linearly separable.

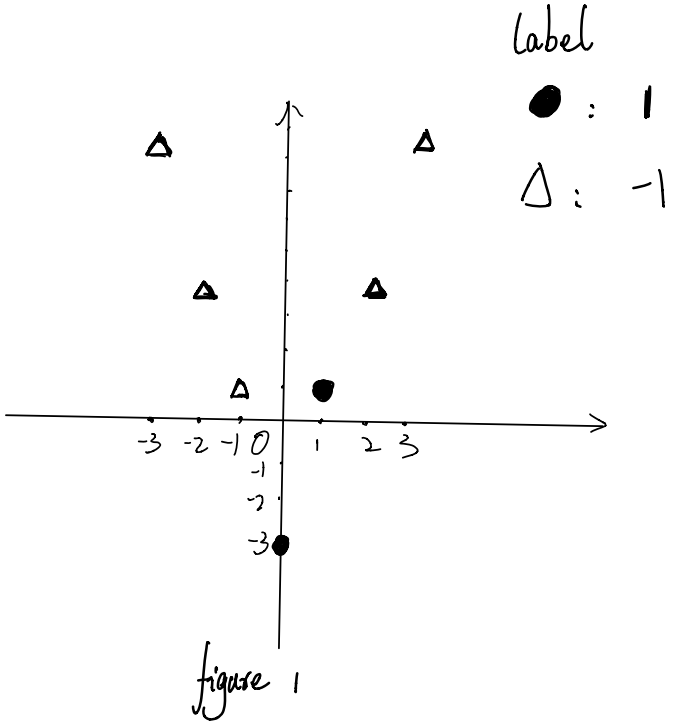


figure 1

(b): according to (a), figure 1, the max margin is close to (-1,1) (1,1) (2,4)
the classifier only near to (-1,1), (1,1), (2,4)

$$\because X=\begin{pmatrix}-1 & 1\\ 1 & 1\\ 2 & 4\end{pmatrix} \quad Y=\begin{pmatrix}-1\\ 1\\ -1\end{pmatrix} \quad X'=XY=\begin{pmatrix}1, & -1\\ 1, & 1\\ -2, & -4\end{pmatrix}$$

$$X'\cdot X'^{T}=\begin{pmatrix}2 & 0 & 2\\ 0 & 2 & -6\\ 2 & -6 & 20\end{pmatrix}$$

$$\arg\max_{a_1,\dots a_n} \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j x_i \cdot x_j \quad \text{subject to } a_i \geq 0, \text{ for all } i,\ \sum_{i=1}^{n} a_i y_i = 0$$

$$= \arg\max_{a_1,\dots a_3} (a_1+a_2+a_3) - \frac{1}{2}(2a_1^2+2a_2^2+20a_3^2+0a_1a_2+2a_1a_3+0a_2a_1 - 6a_2a_3+2a_3a_1-6a_3a_2)$$

$$= a_1+a_2+a_3-a_1^2-a_2^2-10a_3^2-2a_1a_3+6a_2a_3$$

$\because \sum_{i=1}^{n} a_i y_i = 0$

$\therefore -a_1 + a_2 - a_3 = 0$

$a_2 - a_1 = a_3$

$\therefore a_1 + a_2 + (a_2 - a_1) - a_1^2 - a_2^2 - 10(a_2 - a_1)^2 - 2a_1 \cdot (a_2 - a_1) + 6a_2 \cdot (a_2 - a_1)$

$= 2a_2 - a_1^2 - a_2^2 - 10a_2^2 + 20a_2 a_1 - 10a_1^2 - 2a_1 a_2 + 2a_1^2 + 6a_2^2 - 6a_2 a_1$

Assume: $\frac{\partial L}{\partial a_1} = \frac{\partial L}{\partial a_2} = 0$

$\frac{\partial L}{\partial a_1} = -2a_1 + 20a_2 - 20a_1 - 2a_2 + 4a_1 - 6a_2$

$\frac{\partial L}{\partial a_2} = 2 - 2a_2 - 20a_2 + 20a_1 - 2a_1 + 12a_2 - 6a_2$

$\therefore \begin{cases} 12a_2 - 18a_1 = 0 \\ 2 - 4a_2 + 18a_1 = 0 \end{cases} \Rightarrow \begin{cases} a_1 = \frac{2}{3} \\ a_2 = 1 \\ a_3 = \frac{1}{3} \end{cases}$

(c) $w = \sum_{i=1}^{m} a_i y_i x_i$

$= a_1 y_1 x_1 + a_2 y_2 x_2 + a_3 y_3 x_3$

$= -\frac{2}{3} \times \begin{pmatrix} -1 \\ 1 \end{pmatrix} + 1 \times \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \frac{1}{3} \times \begin{pmatrix} 2 \\ 4 \end{pmatrix}$

$= \begin{pmatrix} \frac{2}{3} + 1 - \frac{2}{3} \\ -\frac{2}{3} + 1 - \frac{4}{3} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

Thus, resulting in a margin is $\frac{1}{\|w\|} = \frac{1}{\sqrt{1^2 + (-1)^2}} = \frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2}$

$\because y_i (w^T x_i - t) = 1$

$\because y_1 = -1 \quad x_1 = (-1, 1)$

$\therefore -1(-2 - t) = 1$

$t = 1$

Thus, bias $t = 1$.

(d) Linear classifers are unable to represent non-linear functions.
linear classifer is a line that discriminate two class, and the two class are called linearly - separable. And in some condition, It is clear that drawing one straight line cannot identify all the points of the class correctly.
For example:

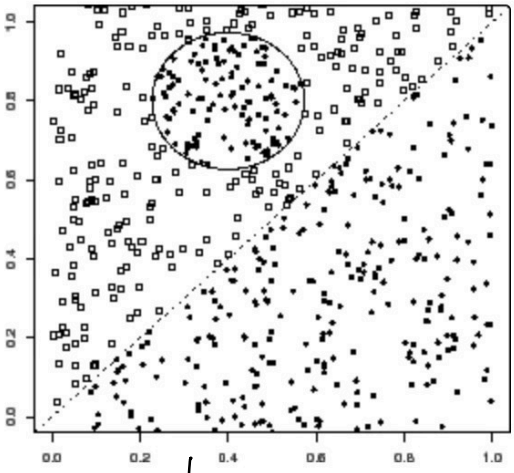


figure 1

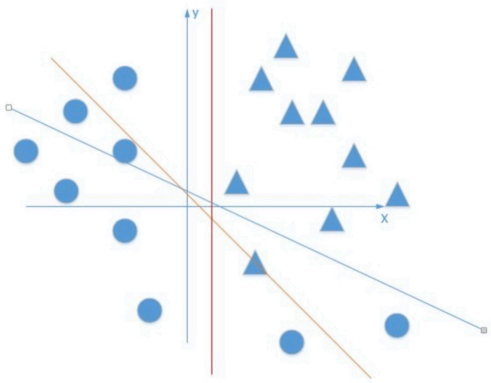


figure 2

Although 'kernel trick' will decrease dimension, it will not decrease to 1 dimension.

(e). the kernel trick is general task of pattern analysis is to find and study types of relations in dataset. for many algorithm that solve these tasks, the data in raw representation have to be explicitly transformed into feature vector via a user-specify feature map. In contrast, kernel methods require only a user-specified kernel and can operate in a high-dimensional

Thus, this is such an important technique for machine learning.

Q4:

(a).

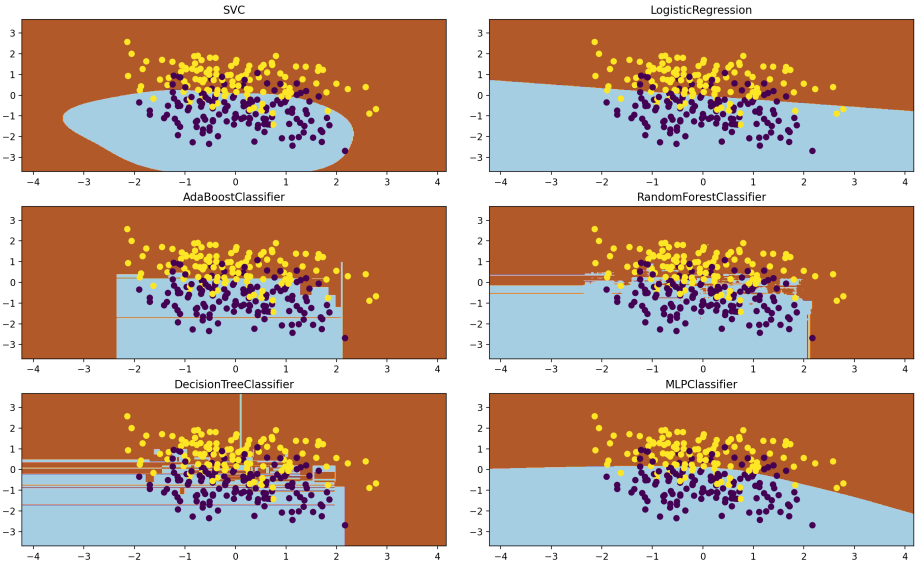
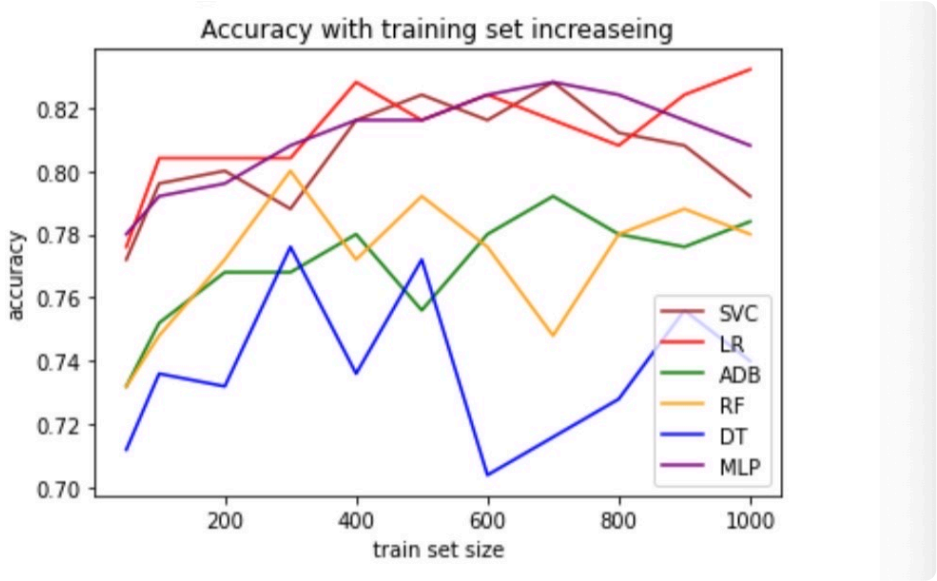


figure 1

(b).



with the increase of train set size. The accurcy of DT are not stability especially in 600 train set size, it have the lowest accuracy. In contrast, with the increase of train set size. ADB has the increasing accuracy, SVC

and LR and MLP have high accuracy all the time.

(c)

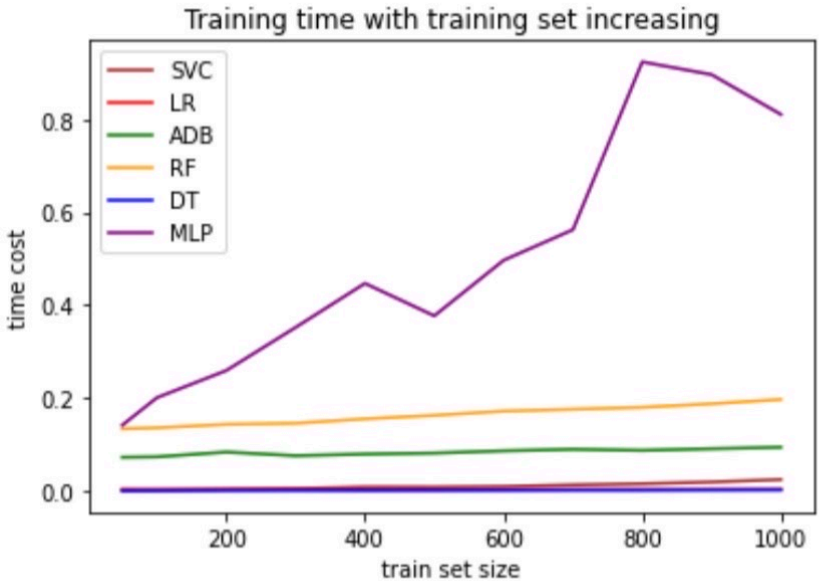


figure 3

with the increasing train set size, the time cost of MLP are still increase. Because MLP in dealing with data, it have a lot of layer so it will cost too much time