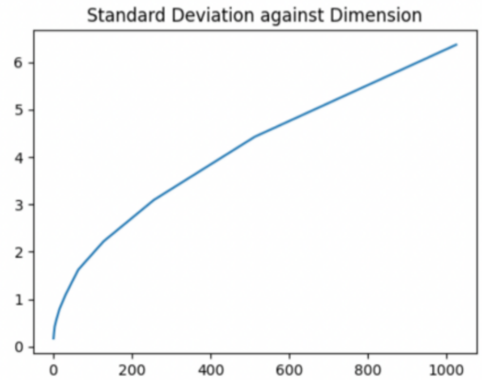
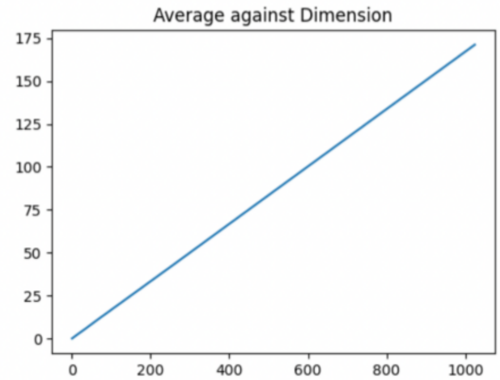
Question 1 a)





Question 1 b)

let $R = Z_1 + Z_2 + \cdots + Z_n$ where $Z_i = (X_i - Y_i)^2$

notice that for Z_i, Z_j for $1 \le i, j \le n$
Z_i and Z_j are independent, because

the difference of distance on one dimension

is independent on another dimension

we know $E[Z_i] = \frac{1}{6}$, $Var[Z_i] = \frac{7}{180}$

by Linearity: $E[R] = E[Z_1 + \cdots + Z_d] = E[Z_1] + \cdots + E[Z_d]$

$= 6\, E[Z_i] = \frac{d}{6}$

by independence: $Var[R] = Var[Z_1 + \cdots + Z_d]$

$= Var[Z_1] + \cdots + Var[Z_d]$

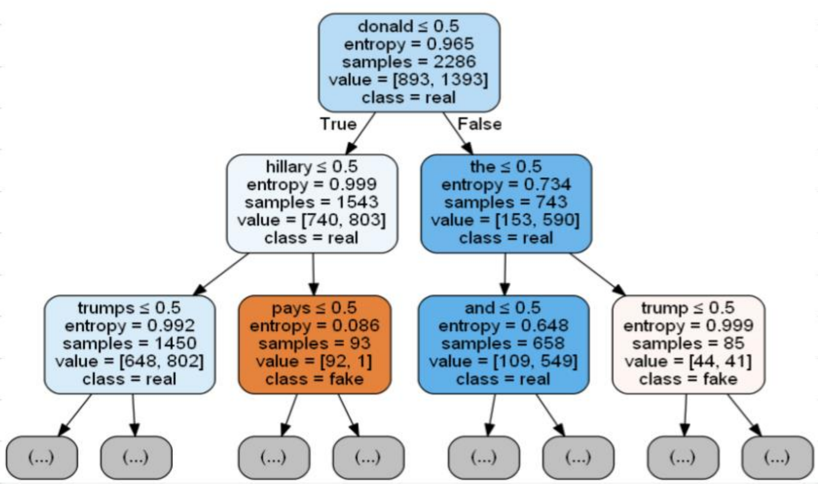
$= d\, Var[Z_i]$

$= \frac{7}{180} d$

(Question 2b)

```
decision tree classifier using max_depth 5 and split criteria information gain has accuracy 0.7
decision tree classifier using max_depth 5 and split criteria Gini coefficient has accuracy 0.7061224489795919
decision tree classifier using max_depth 10 and split criteria information gain has accuracy 0.7081632653061225
decision tree classifier using max_depth 10 and split criteria Gini coefficient has accuracy 0.7183673469387755
decision tree classifier using max_depth 50 and split criteria information gain has accuracy 0.7591836734693878
decision tree classifier using max_depth 50 and split criteria Gini coefficient has accuracy 0.7489795918367347
decision tree classifier using max_depth 100 and split criteria information gain has accuracy 0.7612244897959184
decision tree classifier using max_depth 100 and split criteria Gini coefficient has accuracy 0.7428571428571429
decision tree classifier using max_depth 150 and split criteria information gain has accuracy 0.7755102040816326
decision tree classifier using max_depth 150 and split criteria Gini coefficient has accuracy 0.7673469387755102
```

2c)



2d)

```python
feature_names, data_train, label_train, data_valid, label_valid, data_test, label_test = load_data()
select_model(feature_names, data_train, label_train, data_valid, label_valid, data_test, label_test)
print(
    f'information gain at root split donald is {compute_information_gain(data_train, label_train, "donald", feature_names)}')
print(
    f'information gain at other keyword the is {compute_information_gain(data_train, label_train, "the", feature_names)}')
print(
    f'information gain at other keyword hillary is {compute_information_gain(data_train, label_train, "hillary", feature_names)}')
print(
    f'information gain at other keyword trumps is {compute_information_gain(data_train, label_train, "trumps", feature_names)}')
```

```
C:\Users\wangy\anaconda3\python.exe "C:/uoft-academic/2021 fall/CSC311/hw1/a1.py"
information gain at root split donald is 0.052603317747226375
information gain at other keyword the is 0.047175200175596954
information gain at other keyword hillary is 0.04127665563376459
information gain at other keyword trumps is 0.04340669920550855
```

Question 3

a) Know that

$$J_{reg}^{\beta}(w) = \frac{1}{2N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)})^2 + \frac{1}{2}\sum_{j=1}^{D}\beta_j w_j^2$$

and $y^{(i)} = \sum_{j=1}^{D} w_j x_j^{(i)} + b$ -- ①

substitue $y^{(i)}$ in $J_{reg}^{\beta}(w)$ with ①

$$J_{reg}^{\beta}(w) = \frac{1}{2N}\sum_{i=1}^{N}\left(\left(\sum_{j=1}^{D} w_j x_j^{(i)} + b\right) - t^{(i)}\right)^2 + \frac{1}{2}\sum_{j=1}^{D}\beta_j w_j^2$$

break $J_{reg}^{\beta}(w) = J(w) + R(w)$,

then, $\frac{d}{dw_j} J_{reg}^{\beta}(w) = \frac{d}{dw_j} J(w) + \frac{d}{dw_j} R(w)$

calculate the partial derivative w.r.p w_j for $1 \le j \le D$ for $J(w)$

$$\frac{d}{dw_j} J = \frac{d}{dw_j}\left(\frac{1}{2N}\sum_{i=1}^{N}\left(\left(\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)} + b\right) - t^{(i)}\right)^2\right)$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[\left(\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)} + b - t^{(i)}\right)\left(\frac{d}{dw_j}\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)} + \frac{d}{dw_j} b - \frac{d}{dw_j} t^{(i)}\right)\right]$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[\left(\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)} + b - t^{(i)}\right) \cdot \frac{d}{dw_j}\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)}\right]$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[\left(\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)} + b - t^{(i)}\right) \cdot x_j^{(i)}\right]$$

substitue $y^{(i)}$ back:

$$= \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)}) \cdot x_j^{(i)}$$

calculate the partial derivative w.r.p w_j for $1 \le j \le D$ for $R(w)$

$$\frac{d}{dw_j} R(w) = \frac{d}{dw_j}\left(\frac{1}{2}\sum_{j=1}^{D}\beta_j w_j^2\right) = \beta_j w_j$$

then,

$$\frac{dJ_{reg}^{\beta}}{dw_j} = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)}) x_j^{(i)} + \beta_j w_j$$

According the gradient descent

$$w_j \leftarrow w_j - \alpha\frac{\partial J_{reg}^{\beta}}{\partial w_j}$$

$$w_j \leftarrow w_j - \frac{\alpha}{N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)}) x_j^{(i)} - \alpha\beta_j w_j$$

$$w_j \leftarrow (1 - \alpha\beta_j) w_j - \frac{\alpha}{N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)}) x_j^{(i)}$$

calculate $\frac{dJ_{reg}^{\beta}}{db}$

$$\frac{dJ_{reg}^{\beta}}{db} = \frac{d}{db}\left(\frac{1}{2N}\sum_{i=1}^{N}\left(\sum_{j=1}^{D} w_j' x_j'^{(i)} + b - t^{(i)}\right)^2\right) + \frac{d}{db}\left(\frac{1}{2}\sum_{j=1}^{D}\beta_j w_j^2\right)$$

$$= \frac{d}{db}\left(\frac{1}{2N}\sum_{i=1}^{N}\left(\sum_{j=1}^{D} w_j' x_j'^{(i)} + b - t^{(i)}\right)^2\right)$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[\left(\sum_{j=1}^{D} w_j' x_j'^{(i)} + b - t^{(i)}\right)\left(\frac{d}{db}\sum_{j=1}^{D} w_j' x_j'^{(i)} + \frac{d}{db}b - \frac{d}{db}(t^{(i)})\right)\right]$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[\left(\sum_{j=1}^{D} w_j' x_j'^{(i)} + b - t^{(i)}\right)\right.$$

substitue $y(i)$ back $$= \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)$$

According the gradient descent

$$b \Leftarrow b - \alpha \frac{dJ_{reg}^{\beta}}{db}$$

$$b \Leftarrow b - \frac{\alpha}{N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)$$

In conclusion: $$w_j \Leftarrow (1 - \alpha\beta_j) w_j - \frac{\alpha}{N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right) x_j^{(i)}$$

$$b \Leftarrow b - \frac{\alpha}{N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)$$

Weight decay: According to weight update form, we rescale the weights w_j by $(1-\alpha\beta_j)w_j$ on each gradient descent. it refers to weight decay as it makes weights smaller

b) from part (a), we know $\frac{dJ_{reg}^{\beta}}{dw_j} = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)})x_j^{(i)} + \beta_j w_j$

substitue $y^{(i)} = \sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)}$

$$\frac{dJ_{reg}^{\beta}}{dw_j} = \frac{1}{N}\sum_{i=1}^{N}((\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)}) - t^{(i)})x_j^{(i)} + \beta_j w_j$$

multiply both side by N:

$$\sum_{i=1}^{N}[((\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)}) - t^{(i)})x_j^{(i)} + \beta_j w_j] = N\frac{dJ_{reg}^{\beta}}{dw_j}$$

$$\sum_{i=1}^{N}[\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)} x_j^{(i)} + \beta_j w_j] - \sum_{i=1}^{N} t^{(i)} x_j^{(i)} = N\frac{dJ_{reg}^{\beta}}{dw_j}$$

Expressing $\beta_j w_j$ as $\sum_{j'=1}^{D} w_{j'}\beta_{j'} I_{j'=j}$. Here $I_{j'=j}$ is the indication function

$$I_{j'=j} \begin{cases} 1 & j'=j \\ 0 & j'\neq j \end{cases}$$

$$\sum_{i=1}^{N}[\sum_{j'=1}^{D}(x_{j'}^{(i)} x_j^{(i)} + \beta_{j'} I_{j'=j})w_{j'}] - \sum_{i=1}^{N} t^{(i)} x_j^{(i)} = N\frac{dJ_{reg}^{\beta}}{dw_j}$$

exchange order of $\sum_{i=1}^{N}$ and $\sum_{j'=1}^{D}$:

$$\sum_{j'=1}^{D}[\sum_{i=1}^{N}(x_{j'}^{(i)} x_j^{(i)} + \beta_{j'} I_{j'=j})\, w_{j'}] - \sum_{i=1}^{N} t^{(i)} x_j^{(i)} = N\frac{dJ_{reg}^{\beta}}{dw_j}$$

According the form $\sum_{j'=1}^{D} A_{jj'} w_{j'} - c_j = \frac{dJ_{reg}^{\beta}}{dw_j}$

we can see that

$$\begin{cases} A_{jj'} = \frac{1}{N}\sum_{i=1}^{N}(x_{j'}^{(i)} x_j^{(i)} + \beta_{j'} I_{j'=j}) \\ c_j = \frac{1}{N}\sum_{i=1}^{N} t^{(i)} x_j^{(i)} \end{cases}$$

c) since $c = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N} t^{(i)} x_1^{(i)} \\ \vdots \\ \frac{1}{N}\sum_{i=1}^{N} t^{(i)} x_j^{(i)} \\ \vdots \\ \frac{1}{N}\sum_{i=1}^{N} t^{(i)} x_D^{(i)} \end{bmatrix}$ $X = \begin{bmatrix} x_1^{(1)} & \cdots & x_j^{(1)} & \cdots & x_D^{(1)} \\ x_1^{(i)} & & \ddots & & \\ x_1^{(N)} & & & & x_D^{(N)} \end{bmatrix}$

$t = \begin{bmatrix} t^{(1)} \\ t^{(i)} \\ t^{(N)} \end{bmatrix}$

$c = \frac{1}{N} X^T t$ (verified by matrix product)

similarly, first break $A_{jj'} = \frac{1}{N}\sum_{i=1}^{N} (x_{j'}^{(i)} x_j^{(i)})$
$+ \frac{1}{N}\sum_{i=1}^{N} \beta_{j'} I_{j'=j}$

First notice that $\frac{1}{N}\sum_{i=1}^{N}(x_{j'}^{(i)} x_j^{(i)})$ is the dot product of j^{th} and j'^{th} col
by aggregate all the j and j' which equal to $X^T \cdot X$

for the second part. $\frac{1}{N}\sum_{i=1}^{N} \beta_{j'} I_{j'=j} = \beta_{j'} I_{j'=j}$ iff $j'=j$
by aggreagte all j and j' which equal to diag(β)
i.e a diagonal matrix on the j^{th} position in the
diagonal is β_j.
add the two parts.
$A = \frac{1}{N} X^T X + \text{diag}(\beta)$ and $c = \frac{1}{N} X^T t$

Accordly the form $\sum_{j'=1}^{d} A_{jj'} w_{j'} - C_j = \frac{dJreg^{\beta}}{dw_j} = 0$

weknow $AW = C$

$$W = A^{-1} C$$

$$w = (X^T X + N \operatorname{diag}(\beta))^{-1} X^T \cdot t$$