CSC2515 Fall 2022: Introduction to Machine Learning Homework 1

Collaborators: Chen Dan and Hongbo Zhou

Student Name: Yulin Wang   ID Number: 1003942326

2022-10-13

Q1

(a)

Since $X \sim U(0,1)$ and $Y \sim U(0,1)$, so we have:
$E(X) = E(Y) = \frac{0+1}{2} = \frac{1}{2}$
$Var(X) = Var(Y) = \frac{(1-0)^2}{12} = \frac{1}{12}$
The probability density function (PDF) for X and Y:
$f(x) = 1$ for $x \in [0,1]$, $f(y) = 1$ for $y \in [0,1]$.

Then the expectation of Z

$$\begin{aligned} E(Z) = E(|X-Y|^2) &= E(X^2 + Y^2 - 2XY) \\ &= E(X^2) + E(Y^2) - 2E(XY) \\ &= E(X^2) + E(Y^2) - 2E(X)E(Y) \quad \text{; since X and Y are independent} \\ &= Var(X) + [E(X)]^2 + Var(Y) + [E(Y)]^2 - 2E(X)E(Y) \\ &= \frac{1}{12} + (\frac{1}{2})^2 + \frac{1}{12} + (\frac{1}{2})^2 - 2 \cdot \frac{1}{2} \cdot \frac{1}{2} \\ &= \frac{1}{3} + \frac{1}{3} - \frac{1}{2} \\ &= \frac{1}{6} \end{aligned}$$

and the variance of Z

$$\begin{aligned} Var(Z) = E(Z^2) - [E(Z)]^2 &= E(|X-Y|^4) - (\frac{1}{6})^2 \\ &= E\{X^4 - 4X^3Y + 6X^2Y^2 - 4XY^3 + Y^4\} - \frac{1}{36} \\ &= E(X^4) - 4E(X^3)E(Y) + 6E(X^2)E(Y^2) - 4E(X)E(Y^3) + E(Y^4) - \frac{1}{36} \\ &\text{; since X and Y are independent} \end{aligned}$$

By the moments properties of the uniform distribution, in general the n-th moment of the uniform distribution is:

$$E(X^n) = \frac{b^{n+1} - a^{n+1}}{(n+1)(b-a)} \quad \text{for} \quad X \sim U(a,b)$$

Then we have:

$$\begin{aligned}
E(X^4) = E(Y^4) &= \frac{1}{(4+1)\cdot 1} = \frac{1}{5} \\
E(X^3) = E(Y^3) &= \frac{1}{(3+1)\cdot 1} = \frac{1}{4} \\
E(X^2) = E(Y^2) &= \frac{1}{(2+1)\cdot 1} = \frac{1}{3}
\end{aligned}$$

Thus,

$$\begin{aligned}
Var(Z) &= \frac{1}{5} - 4\cdot\frac{1}{4}\cdot\frac{1}{2} + 6\cdot\frac{1}{3}\cdot\frac{1}{3} - 4\cdot\frac{1}{2}\cdot\frac{1}{4} + \frac{1}{5} - \frac{1}{36} \\
&= \frac{1}{5} - \frac{1}{2} + \frac{2}{3} - \frac{1}{2} + \frac{1}{5} - \frac{1}{36} \\
&= \frac{1}{15} - \frac{1}{36} \\
&= \frac{7}{180}
\end{aligned}$$

(b)

$$\begin{aligned}
E(||X-Y||^2) = E(R) &= E\{|X_1 - Y_1|^2 + ... + |X_d - Y_d|^2\} \\
&= E(|X_1 - Y_1|^2) + ... + E(|X_d - Y_d|^2) \\
&= \sum_{i=1}^{d} E(|X_i - Y_i|^2) \\
&= \sum_{i=1}^{d} E(Z) \\
&= d\cdot\frac{1}{6} \\
&= \frac{d}{6}
\end{aligned}$$

Since $Z_1, ..Z_d$ are from d different dimensions, which means Z_i and Z_j are independent for all $0 < i, j \leq d$ and $i \neq j$. And we know that the variance of sum of the independent variables is the sum of variances of each independent variable. Thus, we have the variance

$$\begin{aligned}
Var(||X-Y||^2) = Var(R) &= Var\{Z_1 + ... + Z_d\} \\
&= Var[\sum_{i=1}^{d} Z_i] \\
&= \sum_{i=1}^{d} Var(Z_i) \\
&= d\cdot\frac{7}{180} \\
&= \frac{7d}{180}
\end{aligned}$$

(c)

The maximum possible squared Euclidean distance between two points within the d-dimensional unit cube would be

$$(\sqrt{\sum_{i=1}^{d}(1-0)^2})^2 = (\sqrt{d})^2 = d$$

As the dimension d increases, $lim_{d\to\infty}E(||X-Y||^2) = lim_{d\to\infty}\frac{d}{6} = \infty$. That is, "most points are far away".

For the standard deviation $SD(||X-Y||^2) = \sqrt{Var(||X-Y||^2)} = \sqrt{\frac{7d}{180}}$, we can compare it and the mean by the fraction

$$\frac{SD(||X-Y||^2)}{E(||X-Y||^2)} = \frac{\sqrt{\frac{7d}{180}}}{\frac{d}{6}} = \sqrt{\frac{7d}{180}\cdot\frac{36}{d^2}} = \sqrt{\frac{7}{5d}}$$

As the dimension d increases, the fraction $lim_{d\to\infty}\sqrt{\frac{7}{5d}} = 0$, which means the standard deviation is very small relatively to the mean, indicating the variation among distances are approaching to 0. That is, "approximately have the same distance".

Therefore, I've proved the claim that in high dimensions, "most points are far away, and approximately have the same distance".

Q2

(a) Show that the random variable Z_i is uniformly distributed between the unit interval $[0,1]$.

We know that $Z_i = |X_i - y| = |X_i - 1|$ and $X_i \sim U(0,2)$.
The CDF of X_i is $F_{X_i}(x_i) = P(X_i \le x_i) = \frac{x_i - 0}{2-0} = \frac{x_i}{2}$.
Then,

$$\begin{aligned}
P(Z_i > z_i) &= P(|X_i - 1| > z_i) \\
&= P(X_i - 1 > z_i) + P(X_i - 1 < -z_i) \\
&= P(X_i > z_i + 1) + P(X_i < 1 - z_i) \\
&= 1 - P(X_i \le z_i + 1) + P(X_i < 1 - z_i) \\
&= 1 - F_{X_i}(z_i + 1) + F_{X_i}(1 - z_i) \\
&= 1 - \frac{z_i + 1}{2} + \frac{1 - z_i}{2} \\
&= 1 - z_i
\end{aligned}$$

Thus, we have the CDF of Z_i as

$$\begin{aligned}
F_{Z_i}(z_i) = P(Z_i \le z_i) &= 1 - P(Z_i > z_i) \\
&= 1 - (1 - z_i) \\
&= z_i \\
&= \frac{z_i - 0}{1 - 0} \quad \text{for} \quad z_i \in [0,1]
\end{aligned}$$

Therefore, $Z_i \sim U(0,1)$, that is, the random variable Z_i is uniformly distributed between the unit interval $[0,1]$.

(b) Show that $E[Z_{(1)}] = \frac{1}{n+1}$.

$$\begin{aligned}
P\{Z_{(1)} > t\} &= P\{min_{i=1,...,n} Z_i > t\} \\
&= P\{Z_1 > t, Z_2 > t, ..., Z_{n-1} > t, Z_n > t\} \\
&= \Pi_{i=1}^{n} P(Z_i > t) \qquad \text{; since } Z_i's \text{ are independent} \\
&= \Pi_{i=1}^{n} \{1 - P(Z_i \le t)\} \\
&= \Pi_{i=1}^{n} \{1 - F_{Z_i}(t)\} \\
&= \{1 - F_{Z_i}(t)\}^n \\
&= (1-t)^n \qquad\qquad \text{; since } F_{Z_i}(t) = t \text{ for } t \in [0,1]
\end{aligned}$$

Then the CDF for $Z_{(1)}$ is

$$F_{Z_{(1)}}(t) = P\{Z_{(1)} \le t\} = 1 - P\{Z_{(1)} > t\} = 1 - (1-t)^n$$

So the PDF for $Z_{(1)}$ is

$$f_{Z_{(1)}}(t) = \frac{dF_{Z_{(1)}}(t)}{dt} = n(1-t)^{n-1}$$

Thus, we have

$$E[Z_{(1)}] = \int_0^1 t f_{Z_{(1)}}(t) dt = \int_0^1 nt(1-t)^{n-1} dt = n \int_0^1 t(1-t)^{n-1} dt$$

By using integral by parts, set $u = t, du = dt$ and $dv = (1-t)^{n-1}dt, v = -\frac{(1-t)^n}{n}$. Then,

$$\begin{aligned}
\int_0^1 t(1-t)^{n-1} dt &= \{-\frac{t(1-t)^n}{n}|_{t=0}^{t=1}\} - \int_0^1 -\frac{(1-t)^n}{n} dt \\
&= \{-\frac{0^n}{n} + \frac{0}{n}\} + \frac{1}{n}\int_0^1 (1-t)^n dt \\
&= \frac{1}{n}\int_0^1 (1-t)^n dt \\
&= \frac{1}{n}\{-\frac{(1-t)^{n+1}}{n+1}|_{t=0}^{t=1}\} \\
&= \frac{1}{n}\{-\frac{0}{n+1} + \frac{1}{n+1}\} \\
&= \frac{1}{n} \cdot \frac{1}{n+1}
\end{aligned}$$

Therefore,

$$E[Z_{(1)}] = n\int_0^1 t(1-t)^{n-1} dt = n \cdot \frac{1}{n} \cdot \frac{1}{n+1} = \frac{1}{n+1}$$

(c) Determine the expected value of the random variable $Z_{(k)}$.

Use the fact that the probability density function of $Z_{(k)}$ is

$$f_{Z_{(k)}}(t) = \frac{n!}{(k-1)!(n-k)!} t^{k-1}(1-t)^{n-k}, \qquad t \in [0,1]$$

Then the expected value of $Z_{(k)}$

$$\begin{aligned}
E[Z_{(k)}] &= \int_0^1 t f_{Z_{(k)}}(t) dt \\
&= \int_0^1 \frac{n!}{(k-1)!(n-k)!} t^k (1-t)^{n-k} dt \\
&= \frac{n!}{(k-1)!(n-k)!} \int_0^1 t^k (1-t)^{n-k} dt \\
&= \frac{n!}{(k-1)!(n-k)!} \int_0^1 t^{(k+1)-1} (1-t)^{(n-k+1)-1} dt \\
&= \frac{n!}{(k-1)!(n-k)!} \cdot B(k+1, n-k+1) \qquad \text{; by definition of beta function} \\
&= \frac{n!}{(k-1)!(n-k)!} \cdot \frac{\Gamma(k+1) \cdot \Gamma(n-k+1)}{\Gamma(k+1+n-k+1)}
\end{aligned}$$

; by the property of beta function with the relationship to gamma function

$$\begin{aligned}
&= \frac{n!}{(k-1)!(n-k)!} \cdot \frac{\Gamma(k+1) \cdot \Gamma(n-k+1)}{\Gamma(n+2)} \\
&= \frac{n!}{(k-1)!(n-k)!} \cdot \frac{(k+1-1)! \cdot (n-k+1-1)!}{(n+2-1)!} \qquad \text{; by definition of gamma function} \\
&= \frac{n!}{(k-1)!(n-k)!} \cdot \frac{k! \cdot (n-k)!}{(n+1)!} \\
&= \frac{k}{n+1}
\end{aligned}$$

(d)

$$lim_{n \to \infty} E[Z_{(k)}] = lim_{n \to \infty} \frac{k}{n+1} = 0$$

Thus, as $n \to \infty$ and $\frac{k}{n} \to 0$, the k-th nearest closest point gets closer and closer to the query point.

Q3

(a) Prove that the entropy $H(X)$ is non-negative.

$$H(X) = \sum_x p(x)log_2(\frac{1}{p(x)}) = -\sum_x p(x)log_2(p(x))$$

Since $p(x) \in [0,1]$, so $log_2(p(x))$ is non-positive, then $\sum_x p(x)log_2(p(x))$ is non-positive. Thus, $H(X) = -\sum_x p(x)log_2(p(x))$ must be non-negative.

(b) If X and Y are independent random variables, show that $H(X,Y) = H(X) + H(Y)$.

$$\begin{aligned}
H(X,Y) &= -\sum_x \sum_y p(x,y)log_2(p(x,y)) \\
&= -\sum_x \sum_y p(x)p(y)log_2(p(x)p(y)) \quad ; \; p(x,y) = p(x)p(y) \text{ since X and Y are independent} \\
&= -\sum_x \sum_y p(x)p(y)[log_2(p(x)) + log_2(p(y))] \\
&= -\sum_x \sum_y p(x)p(y)log_2(p(x)) - \sum_x \sum_y p(x)p(y)log_2(p(y)) \\
&= -\sum_x p(x)log_2(p(x)) - \sum_y p(y)log_2(p(y)) \quad ; \text{since} \; \sum_y p(y) = \sum_x p(x) = 1 \\
&= H(X) + H(Y)
\end{aligned}$$

(c) Prove the chain rule for entropy: $H(X,Y) = H(X) + H(Y|X)$.

$$\begin{aligned}
H(X,Y) &= -\sum_x \sum_y p(x,y)log_2(p(x,y)) \\
&= -\sum_x \sum_y p(y|x)p(x)log_2(p(y|x)p(x)) \\
&= -\sum_x \sum_y p(y|x)p(x)[log_2(p(y|x)) + log_2(p(x))] \\
&= -\sum_x \sum_y p(y|x)p(x)log_2(p(y|x)) - \sum_x \sum_y p(y|x)p(x)log_2(p(x)) \\
&= -\sum_y p(y|x)log_2(p(y|x)) - \sum_x p(x)log_2(p(x)) \\
&= H(Y|X) + H(X)
\end{aligned}$$

(d) Prove that $KL(p||q)$ is non-negative.

$$
\begin{aligned}
KL(p||q) &= \sum_x p(x) log_2 \frac{p(x)}{q(x)} \\
-KL(p||q) &= -\sum_x p(x) log_2 \frac{p(x)}{q(x)} \\
-KL(p||q) &= \sum_x p(x) log_2 \frac{q(x)}{p(x)} \\
-KL(p||q) &= E_{p(x)}[log_2 \frac{q(x)}{p(x)}] \\
&\leq log_2 E_{p(x)}[\frac{q(x)}{p(x)}] \qquad \text{; by Jensen's inequality}
\end{aligned}
$$

Then

$$
\begin{aligned}
KL(p||q) &\geq -log_2 E_{p(x)}[\frac{q(x)}{p(x)}] \\
&= -log_2 \int p(x) \frac{q(x)}{p(x)} dx \\
&= -log_2 \int q(x) dx \\
&= -log_2 1 \\
&= 0
\end{aligned}
$$

Thus, $KL(p||q)$ is non-negative.

(e) The Information Gain or Mutual Information between X and Y is $I(Y;X) = H(Y) - H(Y|X)$. Show that $I(Y;X) = KL(p(x,y)||p(x)p(y))$

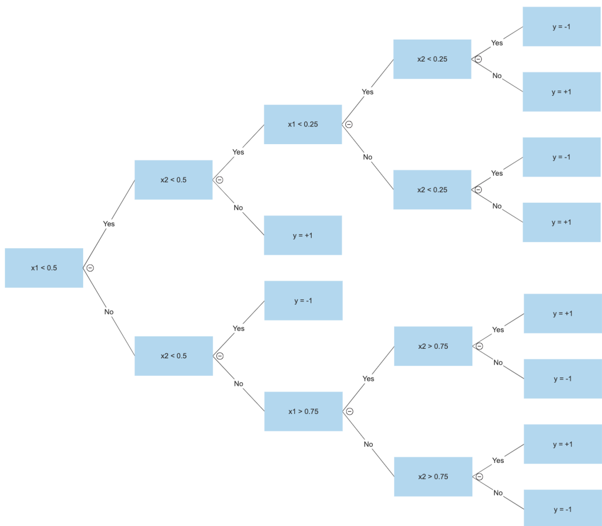
$$
\begin{aligned}
KL(p(x,y)||p(x)p(y)) &= \sum_x \sum_y p(x,y) log_2 \frac{p(x,y)}{p(x)p(y)} \\
&= \sum_x \sum_y p(x,y) log_2 \frac{p(y|x)p(x)}{p(x)p(y)} \\
&= \sum_x \sum_y p(x,y) log_2 \frac{p(y|x)}{p(y)} \\
&= \sum_x \sum_y p(x,y)[log_2(p(y|x)) - log_2(p(y))] \\
&= \sum_x \sum_y p(x,y) log_2(p(y|x)) - \sum_x \sum_y p(x,y) log_2(p(y)) \\
&= -(-\sum_x \sum_y p(y|x)p(x) log_2(p(y|x))) - \sum_y p(y) log_2(p(y)) \\
&\text{; since } p(y) = \sum_x p(x,y), \text{marginal distribution of Y} \\
&= -(-\sum_y p(y|x) log_2(p(y|x))) - \sum_y p(y) log_2(p(y)) \\
&= -H(Y|X) + H(Y) \\
&= I(Y;X)
\end{aligned}
$$

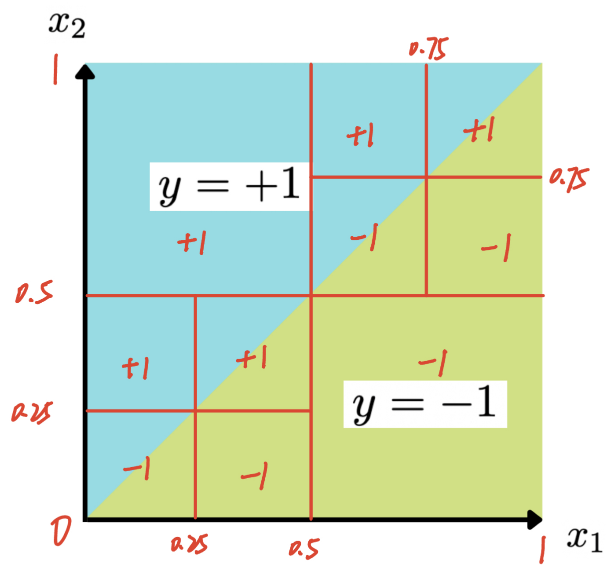
Q4

(a) Explain why f_d with a finite d cannot represent f^* exactly.

Since function f^* splits the region $[0,1]^2$ diagonally, which means every vertical split on x_1 or horizontal split on x_2 will generate two sub-regions with at least one sub-region containing both $y = +1$ and $y = -1$. Thus, we cannot partition the region $[0,1]^2$ into sub-regions that each of them only represents one value (either $y = +1$ or $y = -1$), with a finite number of partitions. Therefore, a finite d cannot represent f^* exactly.

(b) Show what f_4 is.





(c) What is the value of e_2 and e_4?

$$e_2 = 2 \cdot (\frac{1}{2} \cdot 0.5 \cdot 0.5) = 0.25$$

$$e_4 = 4 \cdot (\frac{1}{2} \cdot 0.25 \cdot 0.25) = 0.125$$

(d) Provide the formula for e_d for any even d, that is, $d = 2k$ with $k \in N$.

$$e_d = (\frac{1}{2})^{k+1}, \qquad \text{where } d = 2k \text{ with } k \in N$$

For every even split $d = 2k$ with $k \in N$, the region will generate 2^{k+1} smallest squares, each of whom are with length of $(\frac{1}{2})^k$ and area of $(\frac{1}{2})^{2k}$. Since we know that half of these smallest squares will generate correct y values and wrong y values evenly, resulting the error of $\frac{1}{2} \cdot \frac{1}{2} \cdot$ areas of all smallest squares. Thus,

$$e_d = 2^{k+1} \cdot (\frac{1}{2})^{2k} \cdot \frac{1}{2} \cdot \frac{1}{2} = (\frac{1}{2})^{k+1} \quad \text{with} \quad k \in N$$

, where 2^{k+1} is the number of smallest squares and $(\frac{1}{2})^{2k}$ is the area of each smallest square.

(e) What does this tell us about the quality of approximation as a function of depth d?

As d increases, k increases, then the error $e_d = (\frac{1}{2})^{k+1}$ decreases. That is, the quality of approximation as a function of depth d increases as d increases. Additionally, as the depth d increases, each unit of increase on d will still make improvement for the quality of approximation, but with less improvement effect.

Q5

(a)

Please refer to my uploaded file CSC2515_HW1_Q5.ipynb on MarkUs.

(b)

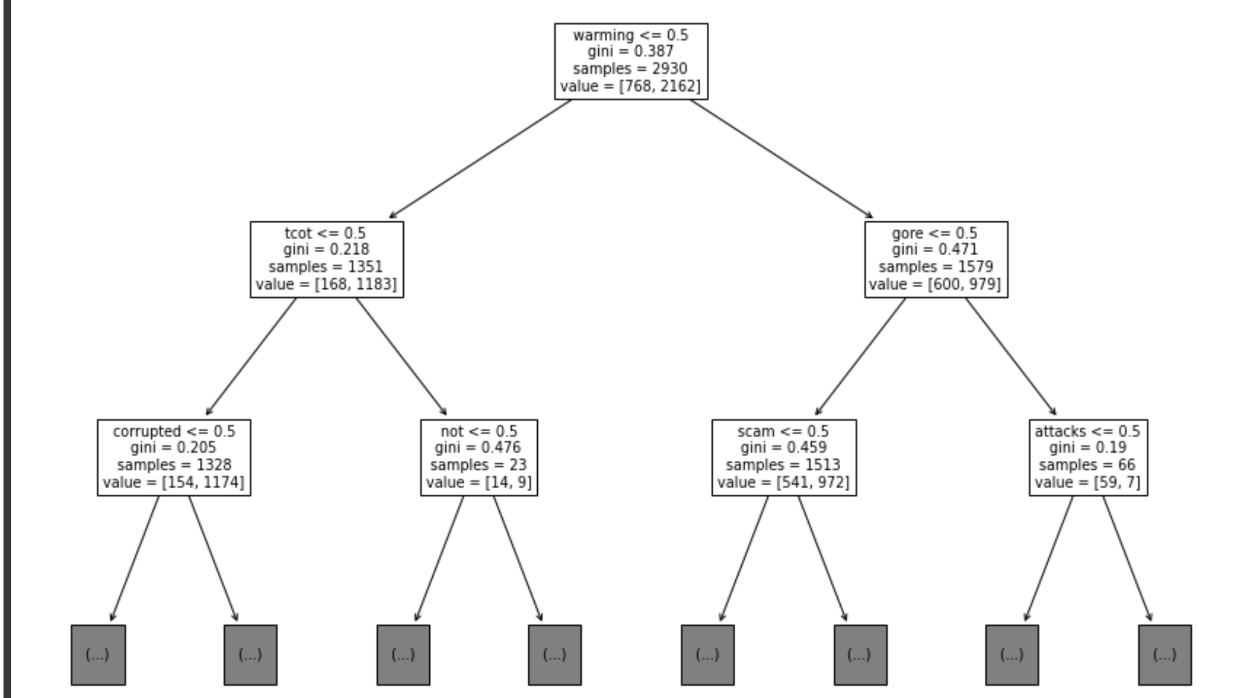
```
[ ] # 5(b) Evaluate the performance of each model on the validation set, and prints the resulting accuracies.
    train_input, train_target, val_input, val_target, test_input, test_target, vectorizer = load_data()
    best_val_model = select_tree_model(train_input, train_target, val_input, val_target)

    Validation accuracy with max_depth 2 and criterion entropy: 0.7611464968152867
    Validation accuracy with max_depth 3 and criterion entropy: 0.7611464968152867
    Validation accuracy with max_depth 4 and criterion entropy: 0.767515923566879
    Validation accuracy with max_depth 5 and criterion entropy: 0.7722929936305732
    Validation accuracy with max_depth 6 and criterion entropy: 0.7770700636942676
    Validation accuracy with max_depth 7 and criterion entropy: 0.7834394904458599
    Validation accuracy with max_depth 8 and criterion entropy: 0.785031847133758
    Validation accuracy with max_depth 9 and criterion entropy: 0.7834394904458599
    Validation accuracy with max_depth 10 and criterion entropy: 0.785031847133758
    Validation accuracy with max_depth 11 and criterion entropy: 0.7882165605095541
    Validation accuracy with max_depth 12 and criterion entropy: 0.7961783439490446
    Validation accuracy with max_depth 13 and criterion entropy: 0.7977707006369427
    Validation accuracy with max_depth 14 and criterion entropy: 0.802547770700637
    Validation accuracy with max_depth 15 and criterion entropy: 0.8057324840764332
    Validation accuracy with max_depth 16 and criterion entropy: 0.802547770700637
    Validation accuracy with max_depth 17 and criterion entropy: 0.8009554140127388
    Validation accuracy with max_depth 18 and criterion entropy: 0.8073248407643312
    Validation accuracy with max_depth 19 and criterion entropy: 0.8073248407643312
    Validation accuracy with max_depth 20 and criterion entropy: 0.8089171974522293
    Validation accuracy with max_depth 2 and criterion gini: 0.7611464968152867
    Validation accuracy with max_depth 3 and criterion gini: 0.767515923566879
    Validation accuracy with max_depth 4 and criterion gini: 0.767515923566879
    Validation accuracy with max_depth 5 and criterion gini: 0.7818471337579618
    Validation accuracy with max_depth 6 and criterion gini: 0.785031847133758
    Validation accuracy with max_depth 7 and criterion gini: 0.7882165605095541
    Validation accuracy with max_depth 8 and criterion gini: 0.7898089171974523
    Validation accuracy with max_depth 9 and criterion gini: 0.7882165605095541
    Validation accuracy with max_depth 10 and criterion gini: 0.7929936305732485
    Validation accuracy with max_depth 11 and criterion gini: 0.7993630573248408
    Validation accuracy with max_depth 12 and criterion gini: 0.8073248407643312
    Validation accuracy with max_depth 13 and criterion gini: 0.8089171974522293
    Validation accuracy with max_depth 14 and criterion gini: 0.8121019108280255
    Validation accuracy with max_depth 15 and criterion gini: 0.8136942675159236
    Validation accuracy with max_depth 16 and criterion gini: 0.8121019108280255
    Validation accuracy with max_depth 17 and criterion gini: 0.8073248407643312
    Validation accuracy with max_depth 18 and criterion gini: 0.8057324840764332
    Validation accuracy with max_depth 19 and criterion gini: 0.8089171974522293
    Validation accuracy with max_depth 20 and criterion gini: 0.8057324840764332
    The best model is the decision tree with max_depth 15, criterion gini and accuracy: 0.8136942675159236
```

(c)

```
# 5(c) Test accuracy of the best model and visualize its first two layers.
test_and_visualize(best_val_model, test_input, test_target, vectorizer)
```

```
The test accuracy of the best model: 0.78060413354531
```



(d)

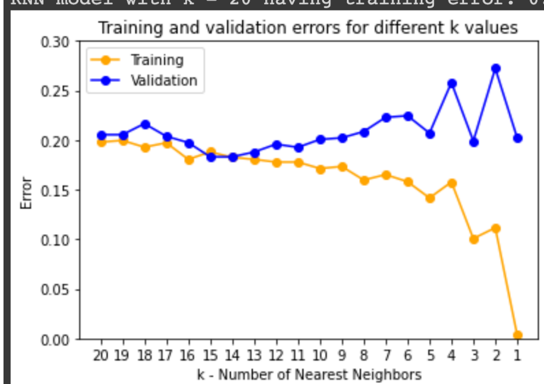
```
# 5(d)
words =  ['warming','tcot','gore', 'scam', 'attacks', 'snow','fraud']
compute_information_gain(train_input, train_target, vectorizer, words)
```

```
('attacks', 0.008625979091950953)
('fraud', 0.015055100099519380)
('gore', 0.02813186170258075)
('scam', 0.012411796856010016)
('snow', 0.012660404534423852)
('tcot', 0.005028414631297329)
('warming', 0.04899454770722378)
```

(e)

```
# 5(e)
select_knn_model(train_input, train_target, val_input, val_target, test_input, test_target)
```

```
KNN model with k = 1 having training error: 0.0034129692832764505 and validation error: 0.20222929936305734
KNN model with k = 2 having training error: 0.11160409556313994 and validation error: 0.27229299363057324
KNN model with k = 3 having training error: 0.10068259385665529 and validation error: 0.19904458598726116
KNN model with k = 4 having training error: 0.157679180887372 and validation error: 0.25796178343949044
KNN model with k = 5 having training error: 0.1416382252559727 and validation error: 0.207006369426751 6
KNN model with k = 6 having training error: 0.15802047781569967 and validation error: 0.22452229299363058
KNN model with k = 7 having training error: 0.16518771331058021 and validation error: 0.2229299363057325
KNN model with k = 8 having training error: 0.1597269624573379 and validation error: 0.2085987261146497
KNN model with k = 9 having training error: 0.17337883959044367 and validation error: 0.20222929936305734
KNN model with k = 10 having training error: 0.17133105802047782 and validation error: 0.20063694267515925
KNN model with k = 11 having training error: 0.17781569965870306 and validation error: 0.1926751592356688
KNN model with k = 12 having training error: 0.17781569965870306 and validation error: 0.19585987261146498
KNN model with k = 13 having training error: 0.18054607508532422 and validation error: 0.18789808917197454
KNN model with k = 14 having training error: 0.1825938566552901 and validation error: 0.18312101910828024
KNN model with k = 15 having training error: 0.18805460750853242 and validation error: 0.18312101910828024
KNN model with k = 16 having training error: 0.18054607508532422 and validation error: 0.19745222929936307
KNN model with k = 17 having training error: 0.19726962457337885 and validation error: 0.20382165605095542
KNN model with k = 18 having training error: 0.19283276450511946 and validation error: 0.21656050955414013
KNN model with k = 19 having training error: 0.19965870307167236 and validation error: 0.2054140127388535
KNN model with k = 20 having training error: 0.19795221843003413 and validation error: 0.2054140127388535
```




```
The best KNN model with k = 14 having validation accuracy: 0.8168789808917197
The best KNN model with k = 14 having test accuracy: 0.739268680445151
```