COMS 4771 HW 2

Griffin Klett

TOTAL POINTS

70 / 115

QUESTION 1

Cost-sensitive Classification 20 pts

1.1 (i) **6 / 10**

- **0 pts** Correct
- ✓ **- 1 pts** **Small arithmetic/notational error**
- **- 2 pts** Used wrong Gaussian $$\sigma^2$$
- **- 3 pts** Mostly correct
- ✓ **- 3 pts** **Show more work**
- **- 6 pts** Show more work
- **- 7 pts** Partially correct
- **- 10 pts** Incorrect or missing
- **- 0 pts** Click here to replace this description.

1 review the notation for N(mean, var)

2 need an expression of the subset in terms of c

1.2 (ii) **10 / 10**

- ✓ **- 0 pts** **Correct**
- **- 1 pts** Small Mistake
- **- 3 pts** Medium Mistake
- **- 5 pts** Large Mistake
- **- 10 pts** Missing/Incomplete

QUESTION 2

Making data linearly separable by feature space mapping 20 pts

2.1 (i) **9 / 10**

- **0 pts** Correct
- ✓ **- 2 pts** **Minor mistake**
- **- 3 pts** __Incorrect bounds for when $$\Phi$$ mapping is nonzero (correct indices $$\alpha \in (x-\sigma, x + \sigma)$$)
- **- 2 pts** $$\sigma$$ does not separate nonzero bounds for every sample/did not provide a concrete $$\sigma$$
- **- 3 pts** Weight vector does not perfectly separate data/no proof of perfect linear separation given/concrete weight vector not provided
- **- 10 pts** No solution
- **+ 1 Point adjustment**

3 You used $$x_{k-1}$$ earlier, and no guarantees that $$x_{k+1} > x_k$$

2.2 (ii) **8 / 10**

- **0 pts** Correct
- ✓ **- 2 pts** **Mostly correct with minor error**
- **- 5 pts** Somewhat correct with one or more significant errors
- **- 8 pts** Attempted / incomplete
- **- 10 pts** Missing

4 But this dot product, naively computed, is intractable. I think you mean instead multiplying the two expressions from above?

QUESTION 3

Learning DNFs with kernel perceptron 35 pts

3.1 (i) **0 / 5**

- **+ 5 pts** Correct
- **+ 5 pts** No Justification but correct
- **+ 3 pts** Very minor error
- ✓ **+ 0 pts** **Incorrect**

3.2 (ii) **0 / 7**

- **0 pts** Correct or close enough with minor mistakes and justified sufficiently.
- **- 2 pts** right answer but reached by examples alone

and no logical or mathematical arguments present

- **- 3 pts** right idea but not quite there
- **- 4 pts** incorrect but with a good attempt
- **- 6 pts** Absolutely no justification for correctness

✓ **- 7 pts** **incorrect or missing**

3.3 (iii) 0 / 8

- **- 0 pts** Correct
- **- 3 pts** Didn't show/explain w* linearly separates $$\phi(S)$$
- **- 5 pts** Incorrect approach for finding $$\gamma$$

✓ **- 8 pts** **Missing/Incorrect**

- **- 2 pts** Minor issue

3.4 (iv) 0 / 8

- **- 0 pts** Correct
- **- 2 pts** Correct attempt with minor issue/slightly incorrect conclusion
- **- 4 pts** Partially correct attempt but incorrect conclusion
- **- 6 pts** Correct initial idea, but incorrect approach post that

✓ **- 8 pts** **Incorrect attempt**

3.5 (v) 0 / 7

- **- 0 pts** Correct
- **- 3 pts** Incorrect proof that each iteration takes O(nd)
- **- 4 pts** Incorrect proof that mistake bound is O(s2^d)
- **- 3 pts** Major mistake / incomplete proof
- **- 2 pts** Minor mistake

✓ **- 7 pts** **Missing / Incorrect solution**

QUESTION 4

Understanding model complexity and overfitting 40 pts

4.1 (i) 28 / 28

✓ **- 0 pts** **Complete Classifier**

- **- 6 pts** Minor Error in Classifier Implementation (e.g trends substantially differ from expectations)
- **- 14 pts** Major Error in Classifier Implementation (e.g >40% training error on all depths)
- **- 28 pts** Missing Classifier

4.2 (ii) 3 / 3

✓ **- 0 pts** **Correct**

- **- 2 pts** Plot Incorrect
- **- 3 pts** Plot missing

4.3 (iii) 3 / 3

✓ **+ 3 pts** **Correct**

- **- 1 pts** Only single plot provided with complete analysis
- **- 1 pts** Misinterpreted random split for different random seeds
- **- 1 pts** Only plots provided with no analysis
- **- 3 pts** Missing
- **- 2 pts** No use of any code (sklearn or personal) but attempt at analysis

4.4 (iv) 3 / 3

✓ **+ 3 pts** **Correct**

- **- 1 pts** Some significant explanation with no reference to plateau/overfit/generalization gap/model complexity
- **- 1.5 pts** Some minor explanation such as description of trend without reasoning, or generic remark on decision trees
- **- 3 pts** Missing/Incorrect
- **- 1 pts** Reference to generic term such as "overfit/underfit" without graph/explanation support

4.5 (v) 3 / 3

✓ **+ 3 pts** **Correct. Return full points if correct in light of analysis despite wrong code**

- **- 0.5 pts** Unreasonable value that is inadequately justified
- **- 3 pts** Missing
- **- 1 pts** splits instead of depth

QUESTION 5

5 Adjustments -3 / 0

- **- 0 pts** Correct

✓ **- 3 pts** **Pages not selected / selected incorrectly**

- 115 pts HOLD (please contact the instructor directly to release the hold)

COMS 4771 Machine Learning (Spring 2021)
Problem Set #2

Griffin Klett - gk2591@columbia.edu

2/26/21

Problem 1: Cost Sensitive Classification

i.
Typical cost function can be defined as the total number of mistakes on $y_1 + y_2$. Here we multiply the cost of the mistakes of false positives by the value c.

$$f^*(x) = \arg\max_{y \in Y} f(x) = Pr(x \mid y = y_i) \cdot Pr(y_i) \cdot \underbrace{(\frac{1}{c})}_{\text{if } y = y_1} \tag{1}$$

To find the decision boundarie(s), set the probability of choosing $y_0 = y_1$

$$Pr(x \mid \pi_0) \cdot Pr(\pi_0) = Pr(x \mid \pi_1) \cdot Pr(\pi_1) \cdot \frac{1}{c} \tag{2}$$

And substitute class conditionals with Guassian densities

$$\frac{1}{\sigma_{\pi_0}\sqrt{2\pi}} e^{-\left(x-\mu_{\pi_0}\right)^2 \Big/ 2\sigma_{\pi_0}^2} \cdot \pi_0 = \frac{1}{\sigma_{\pi_1}\sqrt{2\pi}} e^{-\left(x-\mu_{\pi_1}\right)^2 \Big/ 2\sigma_{\pi_1}^2} \cdot \pi_1 \cdot \frac{1}{c} \tag{3}$$

Note: $\sigma_{\pi_0} = 1, \sigma_{\pi_1} = \frac{1}{4}, \mu_{\pi_0} = 0, \mu_{\pi_1} = 2, \pi_0 = \frac{1}{3}, \pi_1 = \frac{2}{3}$ 1

And when solved, yield the decision boundary for f^* chooses 1 as a function of x and c: When we set c = 1, the range I get for when the classifier will choose 1 is :

$$(4\ ,\ \infty) \tag{4}$$

and when c = 14, the range I get for when the classifier will choose 1 is: 2

$$(2.531\ ,\ 2.802) \tag{5}$$

ii.
Solving the same equation above with c = 15 yields only non-real solutions. Thus there is no real x for which the classifier chooses 1.

1.1 (i) **6 / 10**

- **0 pts** Correct

✓ - **1 pts** **Small arithmetic/notational error**

- **2 pts** Used wrong Gaussian $$\sigma^2$$
- **3 pts** Mostly correct

✓ - **3 pts** **Show more work**

- **6 pts** Show more work
- **7 pts** Partially correct
- **10 pts** Incorrect or missing
- **0 pts** Click here to replace this description.

1. review the notation for N(mean, var)
2. need an expression of the subset in terms of c

COMS 4771 Machine Learning (Spring 2021)
Problem Set #2

Griffin Klett - gk2591@columbia.edu

2/26/21

Problem 1: Cost Sensitive Classification

i.
Typical cost function can be defined as the total number of mistakes on $y_1 + y_2$. Here we multiply the cost of the mistakes of false positives by the value c.

$$f^*(x) = \arg\max_{y \in Y} f(x) = Pr(x \mid y = y_i) \cdot Pr(y_i) \cdot \underbrace{(\frac{1}{c})}_{\text{if } y = y_1} \tag{1}$$

To find the decision boundarie(s), set the probability of choosing $y_0 = y_1$

$$Pr(x \mid \pi_0) \cdot Pr(\pi_0) = Pr(x \mid \pi_1) \cdot Pr(\pi_1) \cdot \frac{1}{c} \tag{2}$$

And substitute class conditionals with Guassian densities

$$\frac{1}{\sigma_{\pi_0}\sqrt{2\pi}} e^{-\left(x-\mu_{\pi_0}\right)^2 \Big/ 2\sigma_{\pi_0}^2} \cdot \pi_0 = \frac{1}{\sigma_{\pi_1}\sqrt{2\pi}} e^{-\left(x-\mu_{\pi_1}\right)^2 \Big/ 2\sigma_{\pi_1}^2} \cdot \pi_1 \cdot \frac{1}{c} \tag{3}$$

Note: $\sigma_{\pi_0} = 1, \sigma_{\pi_1} = \frac{1}{4}, \mu_{\pi_0} = 0, \mu_{\pi_1} = 2, \pi_0 = \frac{1}{3}, \pi_1 = \frac{2}{3}$ 1

And when solved, yield the decision boundary for f^* chooses 1 as a function of x and c: When we set c = 1, the range I get for when the classifier will choose 1 is :

$$(4\ ,\ \infty) \tag{4}$$

and when c = 14, the range I get for when the classifier will choose 1 is: 2

$$(2.531\ ,\ 2.802) \tag{5}$$

ii.
Solving the same equation above with c = 15 yields only non-real solutions. Thus there is no real x for which the classifier chooses 1.

1.2 (ii) **10 / 10**

✓ **- 0 pts Correct**

- 1 pts Small Mistake

- 3 pts Medium Mistake

- 5 pts Large Mistake

- 10 pts Missing/Incomplete

2. Making data linearly separable by feature space mapping

i.

In order to show that the mapping Φ_σ can linearly separate any binary labeling, it is sufficient to show that there exists a hyperplane with normal vector $\vec{w}$ when dotted with $\Phi_\sigma(x_i)$ will split the data by the sign of outputs.

$$sign\left(\vec{w} \cdot \Phi_\sigma(x_i)\right) = sign(y_i) \qquad \text{for all } i = 1, 2, \dots, n$$

Where the mapping $\Phi_\sigma(x) = (\max\{0, 1 - |\frac{\alpha - x}{\sigma}|\})_{\alpha \epsilon \mathbb{R}}$

$$\text{And } \Phi_\sigma(x) = 1 \ when \ 1 - \left|\frac{\alpha - x}{\sigma}\right| > 0$$

So we can rewrite the function to be = 1 when

$$\left|\frac{\alpha - x}{\sigma}\right| < 1$$

Remove the absolute values, multiply out sigma, and add x to both sides to get

$$\Phi_\sigma(x) = 1 \ when \ x - \sigma < \alpha < \ x + \sigma$$

$$\Phi_\sigma(x) = \left(1 - \left|\frac{\alpha - x}{\sigma}\right| \ if \ x - \sigma < \alpha < \ x + \sigma\right)$$

$$\Phi_\sigma(x) = (0 \ if \ not \ x - \sigma < \alpha < \ x + \sigma)$$

Where the non-negative components range from $x - \sigma \ to \ x + \sigma$ for all datapoints.

So the integral can be simplified from:

$$\vec{w} \cdot \Phi_\sigma(x) = \int_{i=0}^{i=\infty} w_i * \left(\Phi_\sigma(x)\right)_i \, di = \int_{x-\sigma}^{x+\sigma} w_i * \left(\Phi_\sigma(x)\right)_i \, di$$

Since we know from the equation of $\Phi_\sigma(x)$ that all values will be non-negative and in the range of [0, 1], the sign is determined by the values of w_i in the range from $x \pm \sigma$

The idea then is to be able to create sufficiently small ranges over w such that they never overlap, and within each range of w we can set $w_i = sign(y_i)$. More specifically, over the range of $i \epsilon \ x \pm \sigma$ we set $w_i = sign\ (y_i)$ for each x.

The next key step is to understand that it is the distance between the two closest/nearest/most similar points which determines how small we set σ to.

Assume that the two closest points are $x_k and\ x_{k-1}$. Their bounds would be:

$$(x_k - \sigma, x_k + \sigma) \quad and \quad (x_{k+1} - \sigma, x_{k+1} + \sigma)$$

Apply the non-overlapping condition $x_k + \sigma \leq x_{k+1} - \sigma$ and solve for sigma:

$$\sigma \leq \frac{x_{k+1} - x_k}{2} \quad (3)$$

Since $x_{k+1} - x_k > 0$ as it is the distance between the two nearest points, we have found a sigma which will be able to linearly separate all the points – and that such sigma is positive and non-zero.

Where our $\vec{w}$ contains $\vec{w}_i \in \left(w_{x_k - \sigma}, w_{x_k + \sigma} \right) = y_k$ for all k

Which when plugged into the original classifier function yields:

$$\vec{w} \cdot \Phi_\sigma(x_k) = \int_{x_k-\sigma}^{x_k+\sigma} w_i * \left(\Phi_\sigma(x_k)\right)_i \, di = \int_{x_k-\sigma}^{x_k+\sigma} y_k * \left(\Phi_\sigma(x_k)\right)_i \, di = y_k * \int_{x_k-\sigma}^{x_k+\sigma} \left(\Phi_\sigma(x_k)\right)_i \, di$$

And since we know the integral $\int_{x_k-\sigma}^{x_k+\sigma} \left(\Phi_\sigma(x_k)\right)_i \, di > 0$ the sign is determined by y_k, which is exactly what we want.

Then we can say that for any n distinct points there exists a $\sigma > 0$ which allows the mapping to linearly separate the binary labeling of the points.

ii.

Given two arbitrary points x and x' the dot product is equal to:

$$\Phi_\sigma(x) \cdot \Phi_\sigma(x') = \int_{i \in \mathbb{R}} \left(\Phi_\sigma(x)\right)_i * \left(\Phi_\sigma(x')\right)_i \, di$$

$$\left(\Phi_\sigma(x)\right)_i > 0 \; in \; range \; [x - \sigma, x + \sigma] \; and \; zero \; otherwise$$

$$\left(\Phi_\sigma(x')\right)_i > 0 \; in \; range \; [x' - \sigma, x' + \sigma] \; and \; zero \; otherwise$$

So this leaves us with two cases:

2.1 (i) **9 / 10**

- **0 pts** Correct

✓ **- 2 pts Minor mistake**

- **3 pts** __Incorrect bounds for when $$\Phi$$ mapping is nonzero (correct indices $$\alpha \in (x-\sigma, x + \sigma)$$)

- **2 pts** $$\sigma$$ does not separate nonzero bounds for every sample/did not provide a concrete $$\sigma$$

- **3 pts** Weight vector does not perfectly separate data/no proof of perfect linear separation given/concrete weight vector not provided

- **10 pts** No solution

+ 1 Point adjustment

3 You used $$x_{k-1}$$ earlier, and no guarantees that $$x_{k+1} > x_k$$

The next key step is to understand that it is the distance between the two closest/nearest/most similar points which determines how small we set σ to.

Assume that the two closest points are $x_k and\ x_{k-1}$. Their bounds would be:

$$(x_k - \sigma, x_k + \sigma) \quad and \quad (x_{k+1} - \sigma, x_{k+1} + \sigma)$$

Apply the non-overlapping condition $x_k + \sigma \leq x_{k+1} - \sigma$ and solve for sigma:

$$\sigma \leq \frac{x_{k+1} - x_k}{2} \quad (3)$$

Since $x_{k+1} - x_k > 0$ as it is the distance between the two nearest points, we have found a sigma which will be able to linearly separate all the points – and that such sigma is positive and non-zero.

Where our $\vec{w}$ contains $\vec{w}_i \in \left(w_{x_k - \sigma}, w_{x_k + \sigma} \right) = y_k$ for all k

Which when plugged into the original classifier function yields:

$$\vec{w} \cdot \Phi_\sigma(x_k) = \int_{x_k-\sigma}^{x_k+\sigma} w_i * \left(\Phi_\sigma(x_k)\right)_i \, di = \int_{x_k-\sigma}^{x_k+\sigma} y_k * \left(\Phi_\sigma(x_k)\right)_i \, di = y_k * \int_{x_k-\sigma}^{x_k+\sigma} \left(\Phi_\sigma(x_k)\right)_i \, di$$

And since we know the integral $\int_{x_k-\sigma}^{x_k+\sigma} \left(\Phi_\sigma(x_k)\right)_i \, di > 0$ the sign is determined by y_k, which is exactly what we want.

Then we can say that for any n distinct points there exists a $\sigma > 0$ which allows the mapping to linearly separate the binary labeling of the points.

ii.

Given two arbitrary points x and x' the dot product is equal to:

$$\Phi_\sigma(x) \cdot \Phi_\sigma(x') = \int_{i \in \mathbb{R}} \left(\Phi_\sigma(x)\right)_i * \left(\Phi_\sigma(x')\right)_i \, di$$

$$\left(\Phi_\sigma(x)\right)_i > 0 \; in \; range \; [x - \sigma, x + \sigma] \; and \; zero \; otherwise$$

$$\left(\Phi_\sigma(x')\right)_i > 0 \; in \; range \; [x' - \sigma, x' + \sigma] \; and \; zero \; otherwise$$

So this leaves us with two cases:

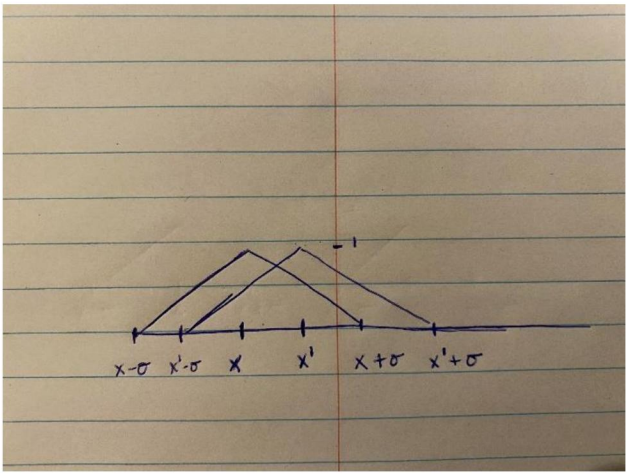
The first and simpler case is when the ranges do not overlap:

$$\Phi_\sigma(x) \cdot \Phi_\sigma(x') = 0$$

Since for all i , either i will be $i \leq x - \sigma$ and $i > x + \sigma$ or $i \leq x' - \sigma$ and $i > x' + \sigma$, and this anything multiplied by zero gives zero, which is all cases of i.

The second more complex case is when the ranges do overlap. In that case, if we arbitrarily assume

$x' > x$, then we calculate the dot product by taking the integral over the overlapping range. Below is a figure of the overlap situation. This occurs in the case where $|x - x'| < 2\sigma$



We can find the interval of intersection as:

$$(\max\{x - \sigma, x' - \sigma\}, \min\{x + \sigma, x' + \sigma\})$$

Outside the range of overlap, the value of the dot product will be zero, by the same logic as the non-overlapping case.

For the overlapping part, we can set definite bounds on the integral as the value of the dot product:

$$\Phi_\sigma(x) \cdot \Phi_\sigma(x') = \int_{\max\{x-\sigma, x'-\sigma\}}^{\min\{x+\sigma, x'+\sigma\}} \left(\Phi_\sigma(x)\right)_i * \left(\Phi_\sigma(x')\right)_i \, di$$

Combining the first and second cases into a piecewise function give us the analytical formula:

$$\Phi_\sigma(x) \cdot \Phi_\sigma(x') = \begin{cases} \int_{\max\{x-\sigma, x'-\sigma\}}^{\min\{x+\sigma, x'+\sigma\}} \left(\Phi_\sigma(x)\right)_i \left(\Phi_\sigma(x')\right)_i \, di & if \ |x - x'| < 2\sigma \\ 0 & otherwise \end{cases}$$

3. Learning DNFS with kernel perceptron

i.

ii.

2.2 (ii) **8 / 10**

- **0 pts** Correct

✓ **- 2 pts** **Mostly correct with minor error**

- **5 pts** Somewhat correct with one or more significant errors
- **8 pts** Attempted / incomplete
- **10 pts** Missing

4 But this dot product, naively computed, is intractable. I think you mean instead multiplying the two expressions from above?

3.1 (i) **0 / 5**

+ **5 pts** Correct
+ **5 pts** No Justification but correct
+ **3 pts** Very minor error
✓ + **0 pts** **Incorrect**

3.2 (ii) **0 / 7**

- **0 pts** Correct or close enough with minor mistakes and justified sufficiently.
- **2 pts** right answer but reached by examples alone and no logical or mathematical arguments present
- **3 pts** right idea but not quite there
- **4 pts** incorrect but with a good attempt
- **6 pts** Absolutely no justification for correctness

✓ **- 7 pts incorrect or missing**

3.3 (iii) **0 / 8**

- **0 pts** Correct
- **3 pts** Didn't show/explain w* linearly separates $$\phi(S)$$
- **5 pts** Incorrect approach for finding $$\gamma$$

✓ - **8 pts** **Missing/Incorrect**

- **2 pts** Minor issue

3.4 (iv) **0 / 8**

- **0 pts** Correct
- **2 pts** Correct attempt with minor issue/slightly incorrect conclusion
- **4 pts** Partially correct attempt but incorrect conclusion
- **6 pts** Correct initial idea, but incorrect approach post that

✓ **- 8 pts** **Incorrect attempt**

3.5 (v) **0 / 7**

- **0 pts** Correct
- **3 pts** Incorrect proof that each iteration takes O(nd)
- **4 pts** Incorrect proof that mistake bound is O(s2^d)
- **3 pts** Major mistake / incomplete proof
- **2 pts** Minor mistake

✓ **- 7 pts** **Missing / Incorrect solution**

iii.

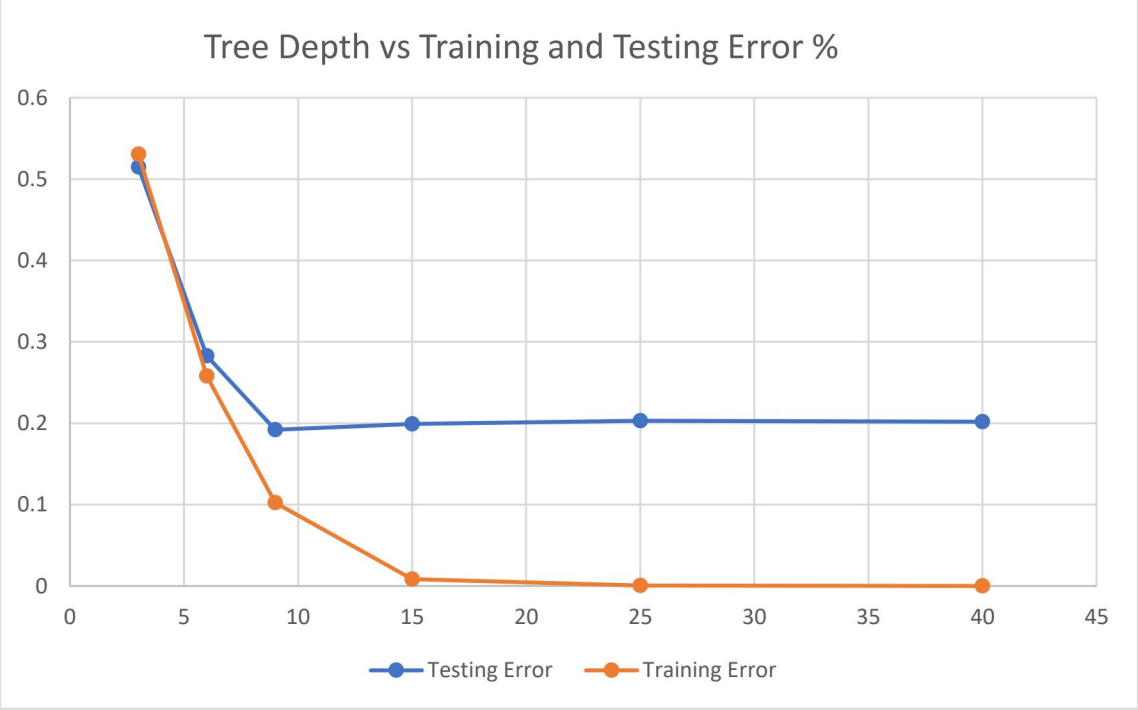
iv.

v.

4. Understanding model complexity and overfitting

i. Code submitted separately

ii.



Depth (K)	Testing Error	Training Error
3	0.515	0.530667
6	0.283	0.258222
9	0.192	0.102556
15	0.199	0.008444
25	0.203	0.000556
40	0.202	0

This dataset is from the 90% training – 10% testing data split. Full data results attached at end.

iii. The trends do not change with different random splits of tests of test/training data. As an example, here is the same graph but using a 60 – 40 % split:

4.1 (i) **28 / 28**

✓ **- 0 pts Complete Classifier**

- 6 pts Minor Error in Classifier Implementation (e.g trends substantially differ from expectations)

- 14 pts Major Error in Classifier Implementation (e.g >40% training error on all depths)

- 28 pts Missing Classifier

iii.

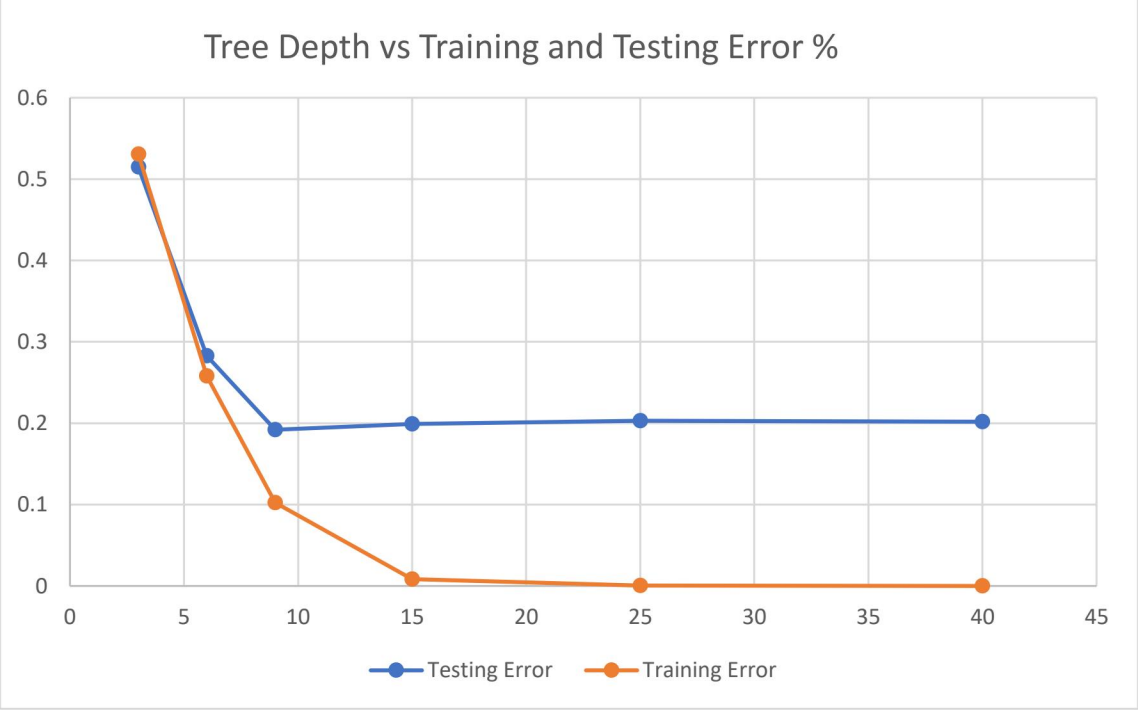
iv.

v.

4. Understanding model complexity and overfitting

i. Code submitted separately

ii.



Depth (K)	Testing Error	Training Error
3	0.515	0.530667
6	0.283	0.258222
9	0.192	0.102556
15	0.199	0.008444
25	0.203	0.000556
40	0.202	0

This dataset is from the 90% training – 10% testing data split. Full data results attached at end.

iii. The trends do not change with different random splits of tests of test/training data. As an example, here is the same graph but using a 60 – 40 % split:

4.2 (ii) **3 / 3**

✓ **- 0 pts** **Correct**

- 2 pts Plot Incorrect

- 3 pts Plot missing

iii.

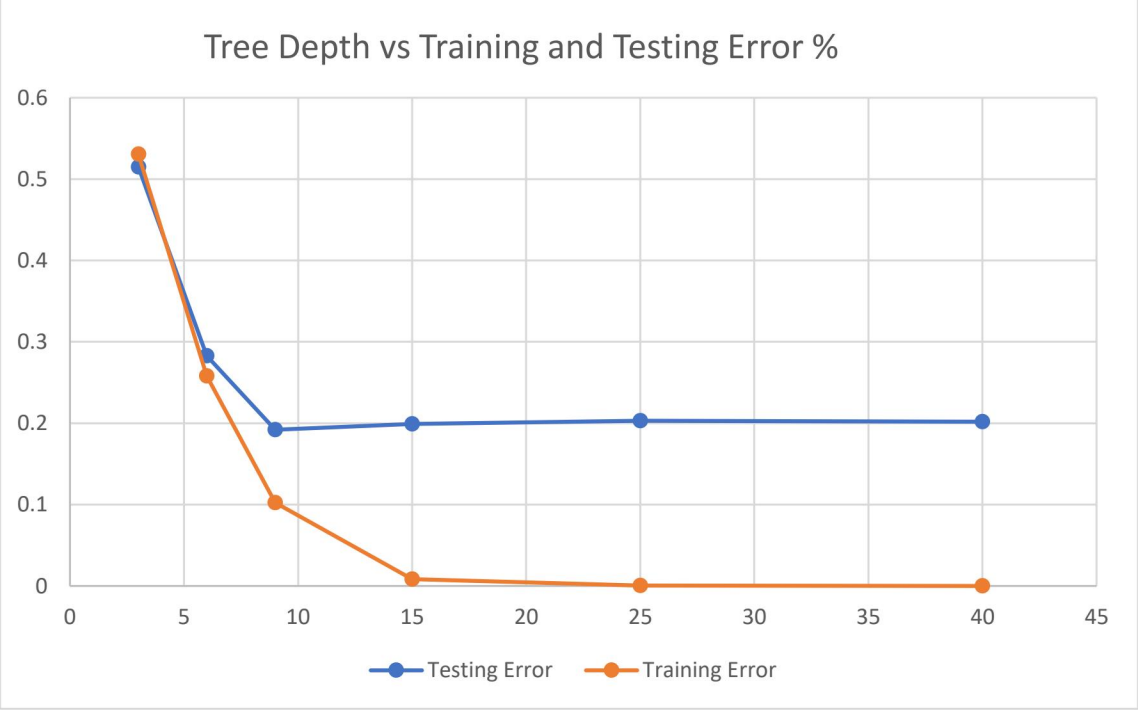
iv.

v.

4. Understanding model complexity and overfitting

i. Code submitted separately

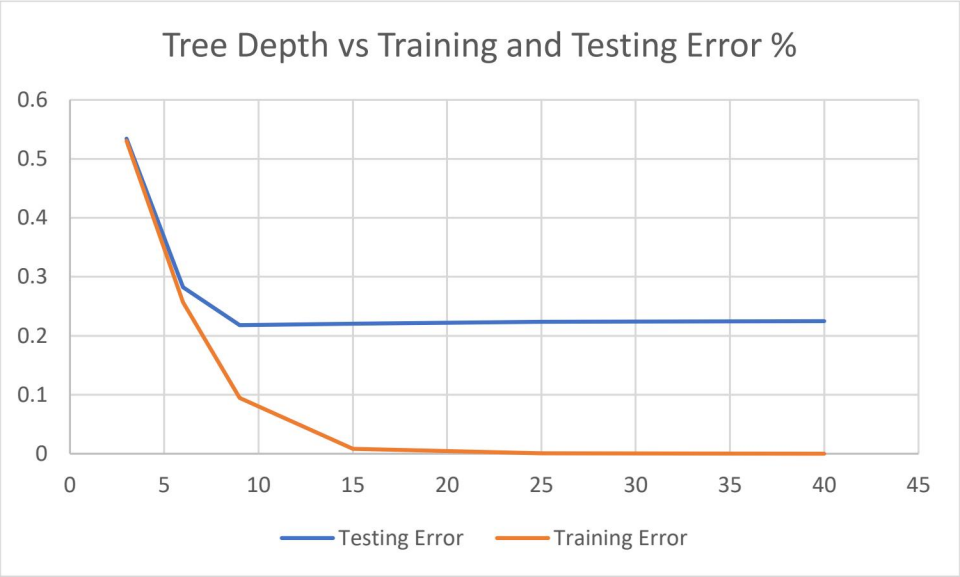
ii.



Depth (K)	Testing Error	Training Error
3	0.515	0.530667
6	0.283	0.258222
9	0.192	0.102556
15	0.199	0.008444
25	0.203	0.000556
40	0.202	0

This dataset is from the 90% training – 10% testing data split. Full data results attached at end.

iii. The trends do not change with different random splits of tests of test/training data. As an example, here is the same graph but using a 60 – 40 % split:



Depth (K)	Testing Error	Training Error
3	0.53425	0.5305
6	0.2825	0.256833
9	0.218	0.0945
15	0.2205	0.008333
25	0.22375	0.0005
40	0.22475	0

We still see the lowest testing error around a tree depth of 9 (in the midde of ranges for tree depth), then a continued decrease in training error and increase in test error.

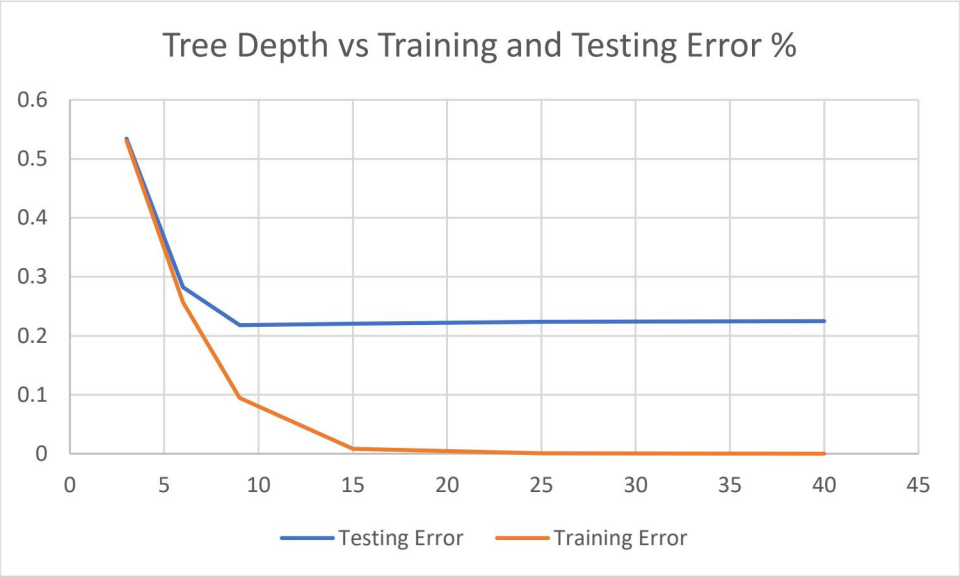
iv.

We can explain the difference in behavior from training and testing error in terms of overfitting. As K increases, the classifier does better and better on the training data, as specific nuances in the data are accounted for and classified correctly – however these small branches are likely overfitting the data. On the test dataset, they perform worse. There exists a “sweetspot” where the tree learns from the training data, but does not allow the branching groups to be too small, so that each branch still makes the best decision for the generalized data and not for the training dataset.

4.3 (iii) 3 / 3

✓ **+ 3 pts** **Correct**

- **1 pts** Only single plot provided with complete analysis
- **1 pts** Misinterpreted random split for different random seeds
- **1 pts** Only plots provided with no analysis
- **3 pts** Missing
- **2 pts** No use of any code (sklearn or personal) but attempt at analysis



Depth (K)	Testing Error	Training Error
3	0.53425	0.5305
6	0.2825	0.256833
9	0.218	0.0945
15	0.2205	0.008333
25	0.22375	0.0005
40	0.22475	0

We still see the lowest testing error around a tree depth of 9 (in the midde of ranges for tree depth), then a continued decrease in training error and increase in test error.

iv.

We can explain the difference in behavior from training and testing error in terms of overfitting. As K increases, the classifier does better and better on the training data, as specific nuances in the data are accounted for and classified correctly – however these small branches are likely overfitting the data. On the test dataset, they perform worse. There exists a “sweetspot” where the tree learns from the training data, but does not allow the branching groups to be too small, so that each branch still makes the best decision for the generalized data and not for the training dataset.

4.4 (iv) 3 / 3

✓ **+ 3 pts Correct**

- 1 pts Some significant explanation with no reference to plateau/overfit/generalization gap/model complexity

- 1.5 pts Some minor explanation such as description of trend without reasoning, or generic remark on decision trees

- 3 pts Missing/Incorrect

- 1 pts Reference to generic term such as "overfit/underfit" without graph/explanation support

v.

We want to set the tree depth so that test error is lowest, based on the different K values I tested, a depth of 9 performed best. To further hone this number, I could re-run the data now will a smaller depth window (say 7-11) and potentially further specialize.

4.5 (v) **3 / 3**

✓ **+ 3 pts Correct. Return full points if correct in light of analysis despite wrong code**

- 0.5 pts Unreasonable value that is inadequately justified

- 3 pts Missing

- 1 pts splits instead of depth

5 Adjustments **-3 / 0**

- **0 pts** Correct

✓ **- 3 pts** **Pages not selected / selected incorrectly**

- **115 pts** HOLD (please contact the instructor directly to release the hold)