CM146, Winter 2021
Problem Set 1: Decision Trees, Nearest Neighbors
Due Jan 29, 2021

Name: Joseph Picchi
UID: 605-124-511

1 Problem 1: Splitting Heuristic for Decision Trees

(a) Problem 1a

Solution:

By definition of the ID3 algorithm, the one leaf will contain the guess that is equivalent to the most frequent label in the data.

Therefore, the ID3 model with 1 leaf will make the same guess for every instance it receives.

Since the training data consists of all 2^n possible examples, and because $Y = 0$ only when $X_1 = 0 \wedge X_2 = 0 \wedge X_3 = 0$, there are 2^{n-3} instances where $Y = 0$. All other instance have label $Y = 1$.

$$\frac{2^{n-3}}{2^n} = 2^{-3} = \frac{1}{8} < 0.5 = 50\,\%$$

Therefore, the instances with label $Y = 0$ are the minority in the data set.
$Y = 1$ is the most frequent label.
The 1 leaf will contain guess $Y = 1$.
The model will guess $Y = 1$ for every training example.
The number of mistakes is equivalent is equivalent to the number of training examples with label $Y = 0$, since the model will incorrectly guess $Y = 1$.

$Y = 0$ only when $X_1 = 0 \wedge X_2 = 0 \wedge X_3 = 0$.
The other $n - 3$ attributes $X_4, \ldots, X_n$ can have any value in $\{0,1\}$.

∴ the number of mistakes is 2^{n-3}

(b) Problem 1b

Solution:

No, there exists no split that reduces the number of mistakes by at least one.

Explanation:

1. Because $X_i \in \{0,1\}$ for each attribute X_i, the single split will result in 2 leaf nodes, no matter which attribute we choose to split on.
2. We denote the attribute we choose to split on as $X_S \in \{X_1, \ldots, X_n\}$. X_S can be any attribute.
3. Since the data set consists of all 2^n examples, half of the examples have $X_S = 0$ and half of the examples have $X_S = 1$.
4. By (3), $\dfrac{2^n}{2} = 2^{n-1}$ examples have $X_S = 0$, and the other 2^{n-1} have $X_S = 1$.
5. From the solution to problem 1a, there exist 2^{n-3} examples with $Y = 0$.
6. Each of the 2 leaves can have a maximum of 2^{n-3} examples with $Y = 0$.
7. By (6) and (4), the maximum percentage of examples with $Y = 0$ for either leaf is:

$$\frac{2^{n-3}}{2^{n-1}} = 2^{n-3-n+1} = 2^{-2} = \frac{1}{4} = 0.25 = 25\,\%$$

8. For both leaves, the majority of examples have label $Y = 1$.
9. By the definition of the ID3 algorithm and (8), both leaves receive the label $Y = 1$.
10. The algorithm guesses that $Y = 1$ for every example.
11. From problem 1a, there are 2^{n-3} instances where $Y = 0$.
12. By (10) and (11), the algorithm makes 2^{n-3} mistakes.
13. Since X_S was any arbitrary attribute, the algorithm makes 2^{n-3} mistakes, no matter which attribute we split on.
14. In my answer to part 1a, the algorithm made 2^{n-3} mistakes.
15. ∴ By (13) and (14), there exists no split that reduces the number of mistakes by at least 1.

(c) Problem 1c

Solution:

$$\begin{aligned} H[Y] &= -\left[P(Y=0)\log P(Y=0) + P(Y=1)\log P(Y=1)\right] \\ &= -\left[P(Y=0)\log P(Y=0) + \left(1-P(Y=0)\right)\log\left(1-P(Y=0)\right)\right] \\ &= -\left[\frac{2^{n-3}}{2^n}\cdot\log\left(\frac{2^{n-3}}{2^n}\right) + \left(1-\left(\frac{2^{n-3}}{2^n}\right)\right)\log\left(1-\left(\frac{2^{n-3}}{2^n}\right)\right)\right] \\ &= -\left[2^{-3}\cdot\log 2^{-3} + (1-2^{-3})\log(1-2^{-3})\right] \\ &= 0.544 \end{aligned}$$

(d) Problem 1d

Solution:

Yes, we can reduce the entropy of Y if we split on either attribute X_1, X_2, or X_3.

Splitting on X_1, we know that all examples with $Y=0$ have $X_1=0$, and all examples with $Y=1$ have $X_1=1$. We get the following calculations:

$$\begin{aligned} H[Y \mid X_1=0] &= -\,[P(Y=0 \mid X_1=0)\log P(Y=0 \mid X_1=0) \\ &\quad + P(Y=1 \mid X_1=0)\log P(Y=1 \mid X_1=0)] \\ &= -\,[P(Y=0 \mid X_1=0)\log P(Y=0 \mid X_1=0) \\ &\quad + \left(1-P(Y=0 \mid X_1=0)\right)\log\left(1-P(Y=0 \mid X_1=0)\right)] \\ &= -\left[\frac{2^{n-3}}{2^{n-1}}\log\frac{2^{n-3}}{2^{n-1}} + \left(1-\frac{2^{n-3}}{2^{n-1}}\right)\log\left(1-\frac{2^{n-3}}{2^{n-1}}\right)\right] \\ &= -\left[\frac{1}{4}\log\frac{1}{4} + \left(1-\frac{1}{4}\right)\log\left(1-\frac{1}{4}\right)\right] \\ &= 0.811 \end{aligned}$$

$$
\begin{aligned}
H[Y\,|\,X_1 = 1] &= -\,[P(Y=0\,|\,X_1=1)\log P(Y=0\,|\,X_1=1) \\
&\quad + P(Y=1\,|\,X_1=1)\log P(Y=1\,|\,X_1=1)] \\
&= -\left[\frac{0}{2^{n-1}}\log\frac{0}{2^{n-1}} + \frac{2^{n-1}}{2^{n-1}}\log\frac{2^{n-1}}{2^{n-1}}\right] \\
&= -\left[0+0\right] \\
&= 0
\end{aligned}
$$

$$
\begin{aligned}
H[Y\,|\,X_1] &= P(X_1=0)H(Y\,|\,X_1=0) + P(X_1=1)H(Y\,|\,X_1=1) \\
&= \frac{1}{2}\log H(Y\,|\,X_1=0) + \frac{1}{2}H(Y\,|\,X_1=1) \\
&= 0.406
\end{aligned}
$$

From problem 1c and our result $H[Y\,|\,X_1]$, splitting on X_1 reduces the entropy from $H[Y] = 0.544$ to $H[Y\,|\,X_1] = 0.406$, which corresponds to a reduction of $H[Y] - H[Y\,|\,X_1] = 0.138 > 0$.

2 Problem 2: Entropy and Information

(a) Problem 2a

Solution:

Show that $0 \leq H(S) \leq 1$:

1. By definition, $H(S) = B(q)$ such that $q = \dfrac{p}{p+n}$.
2. By definition, $0 \leq q = \dfrac{p}{p+n} \leq 1$, since q is a probability value.
3. Find the critical points of $H(S)$:

$$\frac{d}{dq}H(S) = \frac{d}{dq}B(q) = \frac{d}{dq}\left[-q\log q - (1-q)\log(1-q)\right] = 0$$

$$-\log q - \frac{q}{q} + \log(1-q) - \frac{1-q}{1-q}(-1) = 0$$

$$-\log q - 1 + \log(1-q) + 1 = 0$$

$$\log(q^{-1}) + \log(1-q) = 0$$

$$\log\left(\frac{1-q}{q}\right) = 0$$

$$\frac{1-q}{q} = 2^0$$

$$1-q = q$$

$$q = \frac{1}{2}$$

We have a critical point at $q_1 = \dfrac{1}{2}$.

$B(q)$ is continuous in $(0,1)$, but it is discontinuous at the endpoints $q = 0$ and $q = 1$, so our critical points are:

$q_1 = 1/2$, $q_2 = 0$, and $q_3 = 1$.

4. Check to see if the point $q_1 = \dfrac{1}{2}$ is a relative max or min:

$$\frac{d^2}{dq^2}H(S) = \frac{d^2}{dq^2}B(q) = \frac{d}{dq}B'(q) = B''(q)$$

$$= \frac{d}{dq}\left[-\log q + \log(1-q)\right]$$

$$= -\frac{1}{q} - \frac{1}{1-q}$$

$$B''(1/2) = -2-2$$

$$= -4 < 0$$

Since $B''(q_1) < 0$, critical point q_1 is a relative maximum.

5. Evaluate the function at each critical point:

$$\begin{aligned} H(S)\big|_{q_1} = B(q_1) = B(1/2) &= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} \\ &= \frac{1}{2} + \frac{1}{2} \\ &= 1 \end{aligned}$$

$$\begin{aligned} \lim_{q\to q_2} H(S) = \lim_{q\to 0^+} B(q) &= \lim_{q\to 0^+} [-q\log q - (1-q)\log(1-q)] \\ &= \lim_{q\to 0^+} [-q\log q] + \lim_{q\to 0^+} [-(1-q)\log(1-q)] \\ &= \lim_{\frac{1}{q}=t\to\infty} \left[-\frac{\log(\frac{1}{t})}{t}\right] + \lim_{q\to 0^+} [-1\log 1] \\ &= \lim_{t\to\infty} \left[-\frac{1}{1/t}\cdot -t^{-2}\right] + 0 \\ &= \lim_{t\to\infty} \left[\frac{1}{t}\right] \\ &= 0 \end{aligned}$$

$$\begin{aligned} \lim_{q\to q_3} H(S) = \lim_{q\to 1^-} B(q) &= \lim_{q\to 1^-} [-q\log q - (1-q)\log(1-q)] \\ &= \lim_{q\to 1^-} [-q\log q] + \lim_{q\to 1^-} [-(1-q)\log(1-q)] \\ &= [1\log 1] + \lim_{q\to 1^-} \left[-\frac{\log(1-q)}{\frac{1}{1-q}}\right] \\ &= 0 + \lim_{q\to 1^-} \left[\frac{-\frac{1}{1-q}(-1)}{-(1-q)^{-2}(-1)}\right] \\ &= \lim_{q\to 1^-} \left[\frac{(1-q)^2}{1-q}\right] \\ &= \lim_{q\to 1^-} [1-q] \\ &= 0 \end{aligned}$$

6. $H(S)$ approaches value 0 at the endpoints $q = 0$ and $q = 1$, and $H(S)$ has value 0 at critical point $q_2 = 1/2$.

7. $H(S)$ is continuous in $0 < q < 1$
8. $\therefore$ By (6) and (7), $0 \leq H(S) \leq 1$

Show that $H(S) = 1$ when $p = n$

$$
\begin{aligned}
H(S) &= B\left(\frac{p}{p+n}\right) \\
&= B\left(\frac{p}{2p}\right) \\
&= B\left(\frac{1}{2}\right) \\
&= -\frac{1}{2}\log\left(\frac{1}{2}\right) - \left(1-\frac{1}{2}\right)\log\left(1-\frac{1}{2}\right) \\
&= \frac{1}{2} + \frac{1}{2} \\
&= 1
\end{aligned}
$$

(b) Problem 2b

Solution:

1. We denote the k disjoint subsets as $S_1, \ldots, S_k$.
2. We denote the set of all points as $S = S_1 \cup S_2 \cup \ldots \cup S_k$
3. The number of positive and negative examples in each subset S_i are p_i and n_i, respectively.
4. We denote the number of positive and negative examples in S as p and n, respectively.
5. By definition, the ratio of positive examples to total examples is the same for each subset S_i, so $\dfrac{p_i}{n_i + p_i} = \dfrac{p_k}{p_k + n_k}$ for all $i = 1,2,\ldots,k$

6. We show that $\frac{p}{p+n} = \frac{p_k}{p_k+n_k}$, as follows:

$$
\begin{aligned}
p &= p_1 + p_2 + \ldots + p_k \\
(p+n)\left(\frac{p}{p+n}\right) &= (p_1+n_1)\left(\frac{p_1}{p_1+n_1}\right) + \ldots + (p_k+n_k)\left(\frac{p_k}{p_k+n_k}\right) \\
|S|\left(\frac{p}{p+n}\right) &= |S_1|\left(\frac{p_1}{p_1+n_1}\right) + \ldots + |S_k|\left(\frac{p_k}{p_k+n_k}\right) \\
|S|\left(\frac{p}{p+n}\right) &= |S_1|\left(\frac{p_k}{p_k+n_k}\right) + \ldots + |S_k|\left(\frac{p_k}{p_k+n_k}\right) \\
|S|\left(\frac{p}{p+n}\right) &= \left(\frac{p_k}{p_k+n_k}\right)(|S_1| + |S_2| + \ldots + |S_k|) \\
|S|\left(\frac{p}{p+n}\right) &= \left(\frac{p_k}{p_k+n_k}\right)|S| \\
\left(\frac{p}{p+n}\right) &= \left(\frac{p_k}{p_k+n_k}\right)
\end{aligned}
$$

7. Information gain:

$$
\begin{aligned}
GAIN &= H[Y] - H[Y \mid attribute] \\
&= B\left(\frac{p}{p+n}\right) - \sum_{i=1}^{k}\left[\frac{|S_i|}{|S|}H\left(S_i\right)\right] \\
&= B\left(\frac{p_k}{p_k+n_k}\right) - \sum_{i=1}^{k}\left[\frac{|S_i|}{|S|}B\left(\frac{p_i}{p_i+n_i}\right)\right] \\
&= B\left(\frac{p_k}{p_k+n_k}\right) - \sum_{i=1}^{k}\left[\frac{|S_i|}{|S|}B\left(\frac{p_k}{p_k+n_k}\right)\right] \\
&= B\left(\frac{p_k}{p_k+n_k}\right) - B\left(\frac{p_k}{p_k+n_k}\right)\sum_{i=1}^{k}\frac{|S_i|}{|S|} \\
&= B\left(\frac{p_k}{p_k+n_k}\right) - B\left(\frac{p_k}{p_k+n_k}\right)\left(\frac{|S_1|+|S_2|+\ldots+|S_k|}{|S|}\right) \\
&= B\left(\frac{p_k}{p_k+n_k}\right) - B\left(\frac{p_k}{p_k+n_k}\right)\left(\frac{|S|}{|S|}\right) \\
&= B\left(\frac{p_k}{p_k+n_k}\right) - B\left(\frac{p_k}{p_k+n_k}\right) \\
&= 0
\end{aligned}
$$

3 Problem 3: k-Nearest Neighbor

(a) Problem 3a

Solution:

1. $k = 1$ minimizes the training set error because each data point is its own nearest neighbor.
2. The result training error is 0 because…
    1. Every time we classify a point in the training set, we are using the training set to label that point.

2. Therefore, the point that we are classifying is in the same location as its equivalent entry in the training set.
3. So the point being classified has its equivalent entry in the training set as its nearest neighbor.
4. Since $k = 1$, the point being classified receives the same label as its nearest neighbor.
5. By (3) and (4), the point being classified receives the same label as its equivalent entry in the training set. i.e. it receives its own label.
6. By (5), every point receives its own label, so every point is correctly classified.
7. $\therefore$ the training error is 0.

3. Training set error is not a reasonable estimate of test set error because the training points themselves are the means by which new points are classified. Therefore, when calculating training error, the nearest neighbor to the point being classified is always itself, since that point is in the training set that is used to classify points.

   This is especially bad for $k = 1$ because, as shown in (2), every point is classified only according to itself.

   This is still bad for $k > 1$ because one of the points voting on the label for the point being classified is always itself.
   Since any given point in the training set is unlikely to appear again in the test set, this phenomenon is unlike to occur when calculating test set error.
   Thus, training set error is not a reasonable estimate of test set error.

(b) Problem 3b

Solution:

$k = 5$ and $k = 7$ minimize the LOOCV error.

The resulting error is $\frac{4}{14}$ because the only points that are incorrectly classified at cross validation time are (2,7), (3,8), (7,2), and (8,3).

Cross validation is a better measure of test set performance because the training and validation sets are disjoint. Thus, when performance is evaluated on the validation set, the point being tested cannot use itself as a nearest neighbor to vote on its label. Therefore, we avoid the issue described in problem 3a and generate performance metrics that are more representative of those generated from the test set (since the test set points also cannot use themselves as one of the nearest neighbors).

(c) Problem 3c

Solution:

For $k = 1$, the LOOCV error is $\frac{10}{14} = 0.714 = 71.4\,\%$

For $k = 13$, the LOOCV error is $\frac{14}{14} = 1 = 100\,\%$.

Using a value of k that is too small may cause overfitting. The decision boundary will be very sensitive to each individual point in the training set, since we assign labels based on only a small number of nearest neighbors. This means that outliers or "noise" points in the test set have a greater effect on label predictions, likely increasing the test set error.

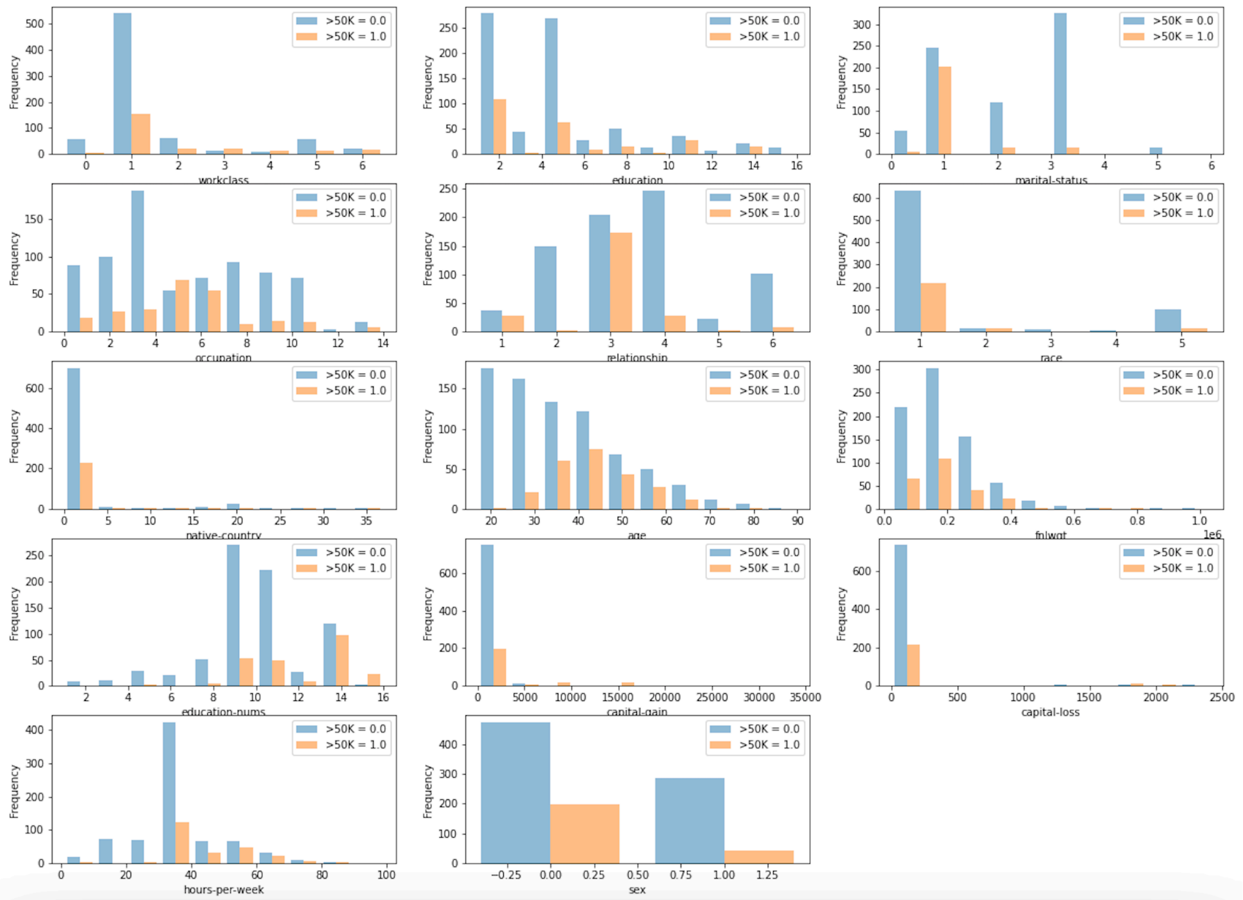
Using a value of k that is too large may cause underfitting. High k values cause the model to consider points that are longer distances away from the given instance when predicting that instance's label, thus leading increasing misclassifications because points that are farther away may be more likely to have different labels from the local neighborhood. This could increase both training set and test set error, since in both cases, high k values consider more points outside of an instance's local neighborhood.

4 Problem 4: k-Nearest Neighbor

(a) Problem 4a

Solution:

(Histograms and analysis shown on next page)



NOTE: I denote the class >50K as “high-earners” and the class <= 50K as “low earners.

For workclass, the feature value with the highest frequency for both classes is self-employed people in non-corporate settings. The distribution of both classes across the feature space for workclass is relatively equivalent.

For education, low earners are much more likely to have lower-level degrees, belonging to feature value 5 or less. High earners had the greatest frequency at feature value 5 or less, but their likelihood of having a degree higher than feature value 5 is greater that of low earners.

For marital status, high earners have a much higher frequency of belonging to feature value 1 compared to any other feature value. Low earners have the highest frequency of belonging to feature value 3, with comparably high likelihoods of belonging to feature values 1 and 2, meaning that low earners are more spread out across the feature space than are high earners.

For occupation, higher earners have the highest frequency of belonging to the middle range of feature values, with a relatively normal distribution around feature value 5. low earners have the highest likelihood of belonging to feature value 3, and they are much more spread out across the feature space (ie have higher frequencies around high and low feature values) compared with high earners.

For relationship, high earners are much more likely to belong to feature value 3 compared to any other value. Low earners have a mean feature value close to 3, but they are much more spread out across the feature space than are high earners.

For race, both low and high earners have a much higher frequency of belonging to feature value 1 compared to any other value.

For native country, both low and high earners have a much higher frequency of belonging to feature value 1 compared to any other value.

For age, high earners have a relatively normal distribution across the feature space with a mean value around 40. Low earners have the highest frequency of being in low feature values, and frequency of low earners consistently decreases as age increases.

For fnlwgt, both low and high earners have fairly equivalent distributions that are approximately normalized around feature value 0.2.

For education-nums, low earners have the highest frequencies around feature values 9-11. High earners have the highest frequencies centered around feature value 14, with half as much frequency of belonging to feature values 9-11 compared to 14.

For capital-gain, both low and high earners have the highest frequencies near feature value 0, though high earners are slightly more likely to belong to higher feature values compared to low earners.

For capital-loss, both low and high earners have the highest frequencies near feature value 0, with extremely low frequencies at higher feature values.

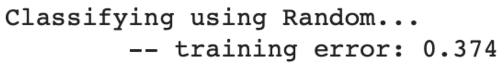
For hours-per-week, low earners have a much higher frequency of belonging to the feature value of approximately 35, with much lower frequencies of belonging to either extreme. High earners are more spread out across the feature space, though their maximum frequency is still centered around feature space 35.

For sex, both low and high earners have relatively equal distributions, with greater frequency of belonging to feature value 0 compared to 1.

(b) Problem 4b

Solution:

Result:



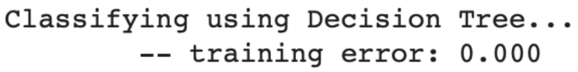
```
Classifying using Random...
        -- training error: 0.374
```

RandomClassifier causes a training error of 0.374.

(c) Problem 4c

Solution:

Result:



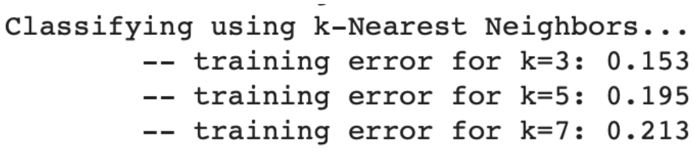
```
Classifying using Decision Tree...
        -- training error: 0.000
```

This classifier produces a training error of 0.

(d) Problem 4d

Solution:

Output:

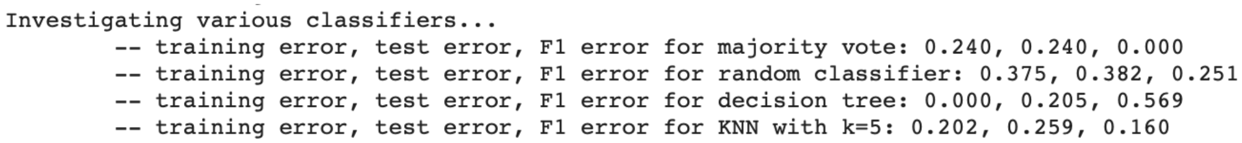


```
Classifying using k-Nearest Neighbors...
        -- training error for k=3: 0.153
        -- training error for k=5: 0.195
        -- training error for k=7: 0.213
```

(e) Problem 4e

Solution:

Result:



```
Investigating various classifiers...
        -- training error, test error, F1 error for majority vote: 0.240, 0.240, 0.000
        -- training error, test error, F1 error for random classifier: 0.375, 0.382, 0.251
        -- training error, test error, F1 error for decision tree: 0.000, 0.205, 0.569
        -- training error, test error, F1 error for KNN with k=5: 0.202, 0.259, 0.160
```

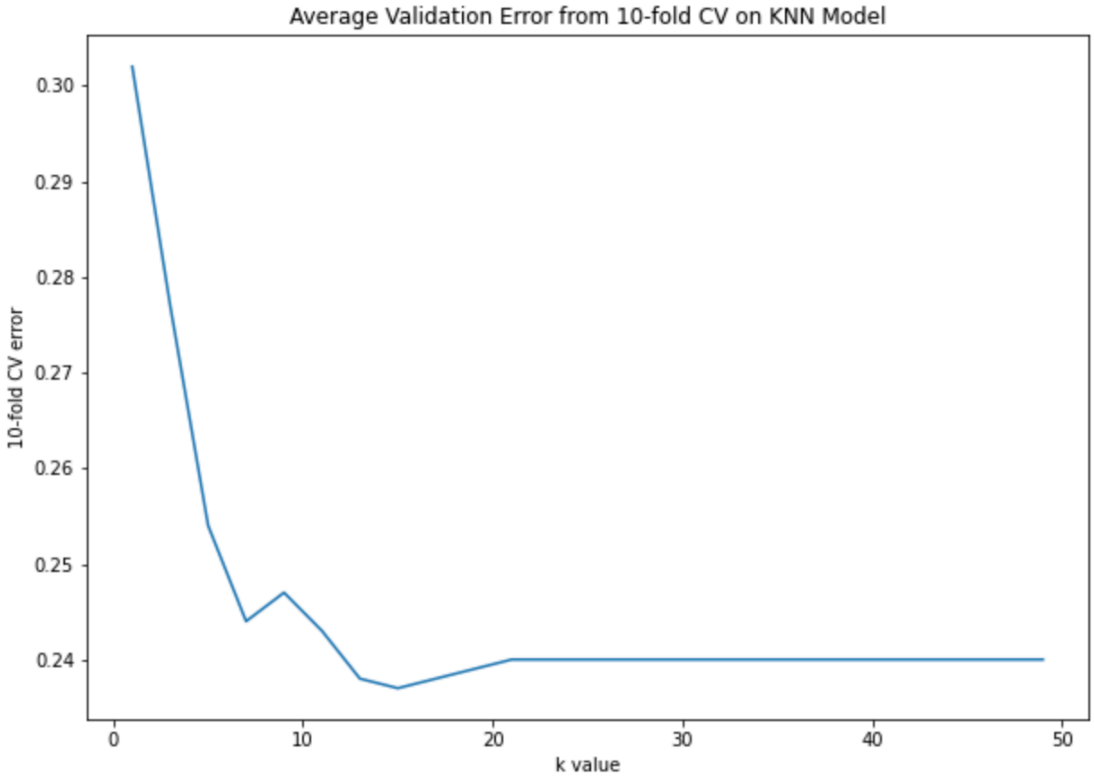
As seen above, the values are:

- Majority Vote Classifier:
  - Training error: 0.240
  - Test error: 0.240

  - F1 score: 0.000
- Random Classifier:
  - Training error: 0.375
  - Test error: 0.382
  - F1 score: 0.251
- Decision Tree Classifier:
  - Training error: 0.000
  - Test error: 0.205
  - F1 score: 0.569
- KNN with K=5
  - Training error: 0.202
  - Test error: 0.259
  - F1 score: 0.160

(f) Problem 4f

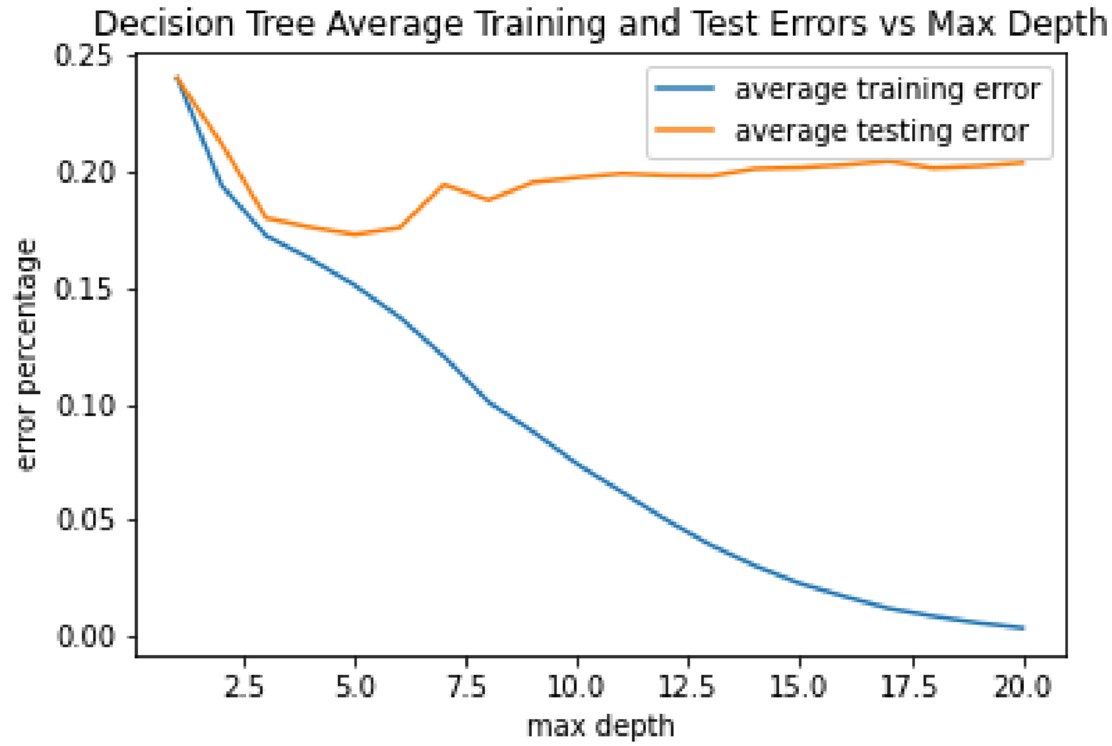
Plot of validation error against number of neighbors k:



The validation error starts off very high for small values of K because the model is overfit to the training data, so the model does not generalize well to new test instances and resultantly causes a high validation error rate. As K continues to grow, the validation error rate decreases to a minimum at K=15 and increases again for $K > 15$ because the model is underfit to the training data, causing it to consider points far outside the local neighborhood of common labels for a given test instance.

The best value of K is $K = 15$ with a validation error of 0.237.

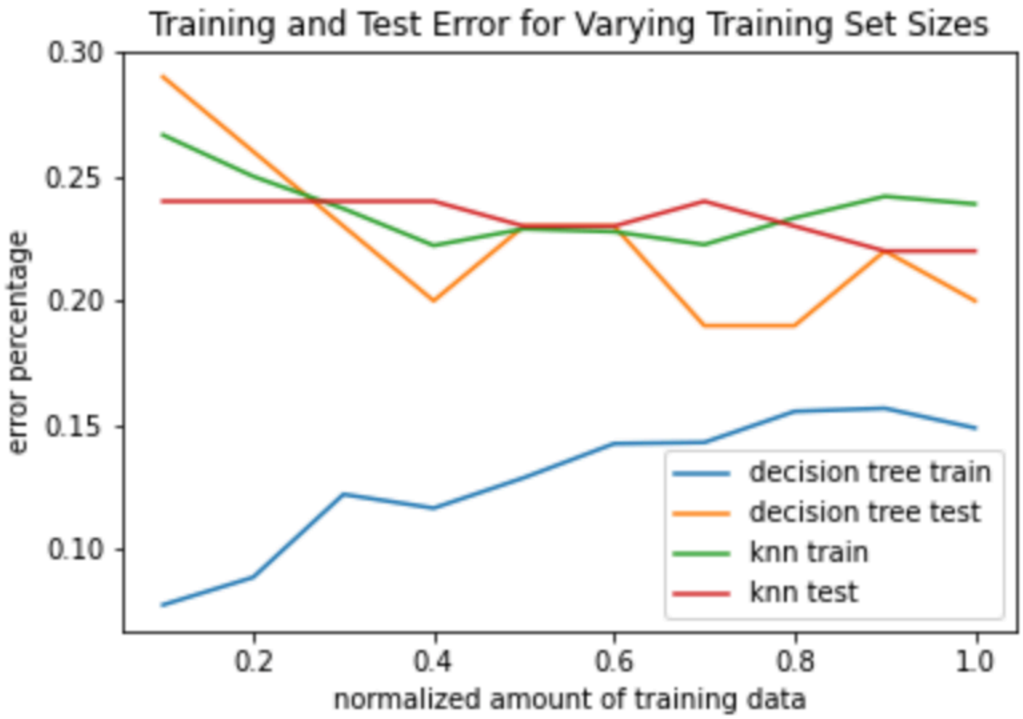
(g) Problem 4g



The best depth limit for this data is 5. We can see this in the plot because a max depth of 5 on the x-axis corresponds to the global minimum for the average test error curve. Since we seek to minimize test error in the final model, we choose this max depth as our optimal value.

When the depth limit is too large (i.e. the value for max depth is too large), the model overfits. This is evident because the training error decreases, then reaches a minimum, then increases again when the depth limit becomes large. All the while, training error continues to decrease as max depth increases. This signifies that the model is fitting more and more to the training data (yielding smaller and smaller training error) and consequently becomes worse at generalizing to new test instances (yielding increasing test errors). This trend is characteristic of overfitting.

(h) Problem 4h



As the amount of training data increases, the decision tree training error increases because the max depth of 5 cuts it short and the increased examples cause it to misclassify more and more examples. KNN training error initially decreases because more training examples mean that more of the 15 neighbors are in the same neighborhood as the training point being analyzed, and it later stabilizes because more data points increase the probability that close neighbors have equivalent labels.

Decision tree test error decreases because increased data points make it harder for the model to overfit to the training data, thus allowing it to generalize better to new test examples. KNN test error is relatively stable with a slight decrease because more examples increase the likelihood that the 15 neighbors of the given test point are in the same neighborhood, thus increasing the chances of correct classification.

(i) Problem 4i

Part b results:

RandomClassifier still has an error of 0.374.

Part c results:

The decision tree classifier still produces a training error of 0.

Part d results:

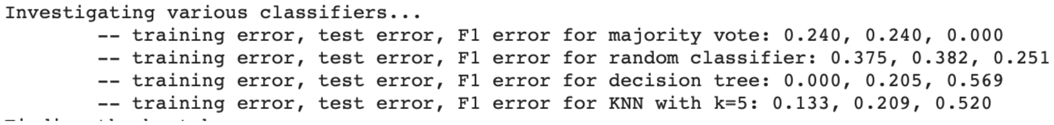
The training errors for KNeighborsClassifier changed to:

0.114 for k=3
0.129 for k=5
0.152 for k=7

Compared to the un-normalized data, the training errors for all 3 values of k decreased.

Part e results:

The training, test, and F1 values changed to the following:



```
Investigating various classifiers...
        -- training error, test error, F1 error for majority vote: 0.240, 0.240, 0.000
        -- training error, test error, F1 error for random classifier: 0.375, 0.382, 0.251
        -- training error, test error, F1 error for decision tree: 0.000, 0.205, 0.569
        -- training error, test error, F1 error for KNN with k=5: 0.133, 0.209, 0.520
```

Written out, the new values are:

- Majority Vote Classifier:
    - Training error: 0.240
    - Test error: 0.240
    - F1 score: 0.000
- Random Classifier:
    - Training error: 0.375
    - Test error: 0.382
    - F1 score: 0.251
- Decision Tree Classifier:
    - Training error: 0.000
    - Test error: 0.205
    - F1 score: 0.569
- KNN with K=5
    - Training error: 0.113
    - Test error: 0.209
    - F1 score: 0.520

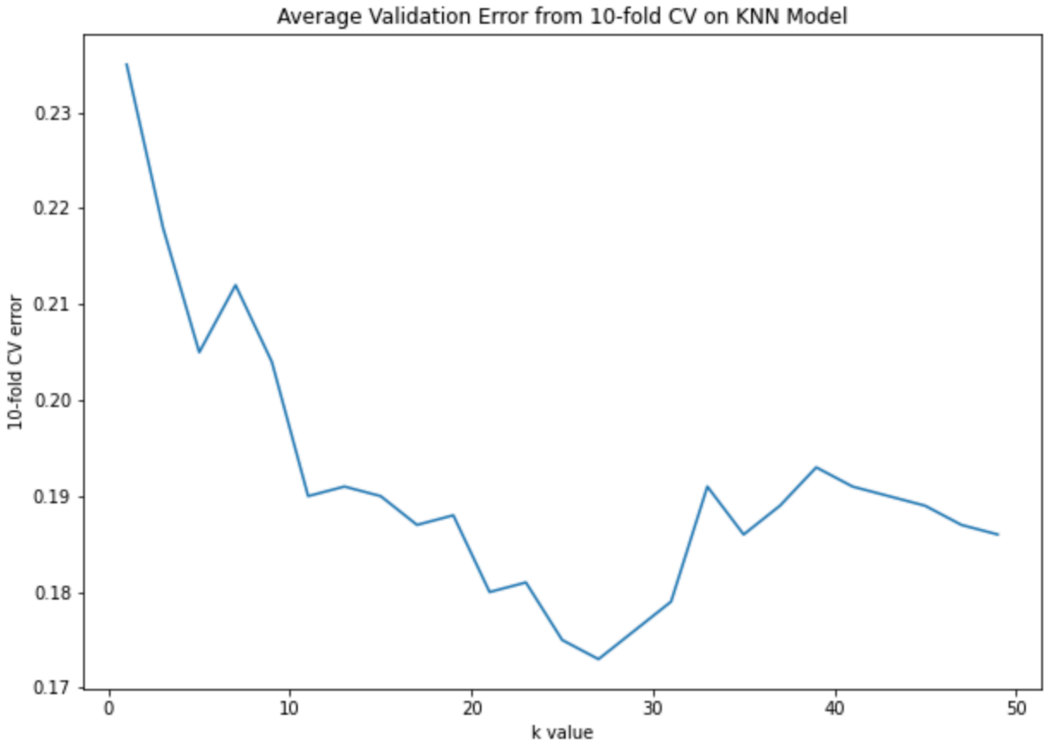
Comparing these new values with the old values, all of them remain unchanged for the first 3 models (ie Majority Vote Classifier, Random Classifier, and Decision Tree Classifier).

All 3 values changed for KNN with K=5. Training error decreased from its value of 0.202 for un-normalized data, test error decreased from its value of 0.259 for un-normalized data, and F1 score increased from its value of 0.160 for un-normalized data.

This makes sense because normalizing the data for KNN ensures that no one feature dominates the consideration of which neighbors are closest, thus allowing features with higher label correlations to drive more accurate predictions.

Part f results;

The new graph for part f appears as follows:



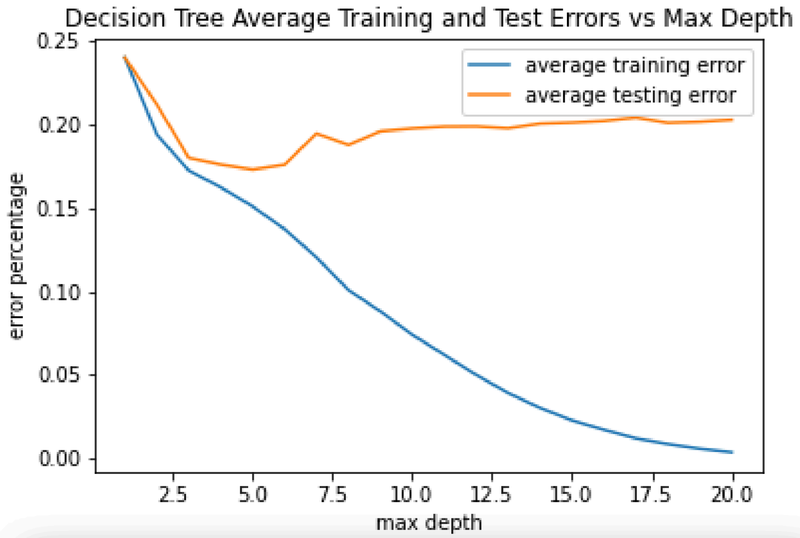
The new best value of k is $k = 27$, which has an average validation score of 0.173 (lower than the validation score of the minimum k for un-normalized data).

The dip in the graph is more defined than it was for un-normalized data because features with better label correlation are no longer dominated by other features simply because of their units, thus allowing for a more optimal validation score and clearer delineations of k values with overfitting and under-fitting.

The optimal k value increased because data points with similar labels are now more likely to be in the same local neighborhood as each other due to normalization. This means that a higher k value captures more of the local points with similar labels to the validation point we are classifying.

Part g results:

The new graph appears as follows:



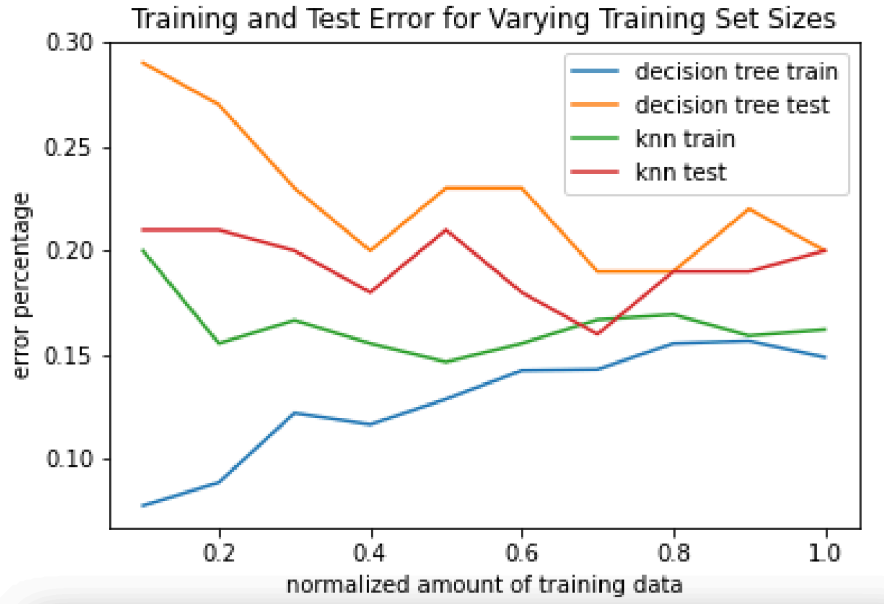
The new graph is pretty much the same as the old graph.

The optimal max depth is still 5, which makes sense because edge values in the decision tree can be selected from any of the feature values for a given node. Thus, normalization does not effect our results for decision trees as much as it does for KNN models.

We still see overfitting large values of max depth for the same reasons previously stated in the solution to problem 4g.

Part h results:

The new graph appears as follows:



Decision tree train and decision tree test still have the same general trends for the reasons stated in problem 4h solution.

KNN train and test errors still have the same general trends as before, but they are sloped down slightly more compared to the graph in part h. Both curves also have lower y-values at all points along the x-axis. This makes sense because features with better label correlation are no longer dominated by other features simply because of their units, thus allowing them to drive more accurate classifications from KNN and further reduce overfitting (which explains the more downward slope) as the amount of training data increases.