CIS 520, Machine Learning, Fall 2020
Homework 1
Due: Monday, September 21st, 11:59pm
Submit to Gradescope

Chun

1 Non-Normal Norms

1. For the given vectors, the point closest to x_1 under each of the following norms is

   $x_{21} = x_2 - x_1 = [-2.7, 0.3, 2.5, 0.5]$
   $x_{31} = x_3 - x_1 = [-3.8, 1., -2.1, -0.7]$
   $x_{41} = x_4 - x_1 = [-3.6, 2.7, 0., 1.2]$

   a) L_0:
   $|x_{21}|_0 = 4$
   $|x_{31}|_0 = 4$
   $closest : |x_{41}|_0 = 3$

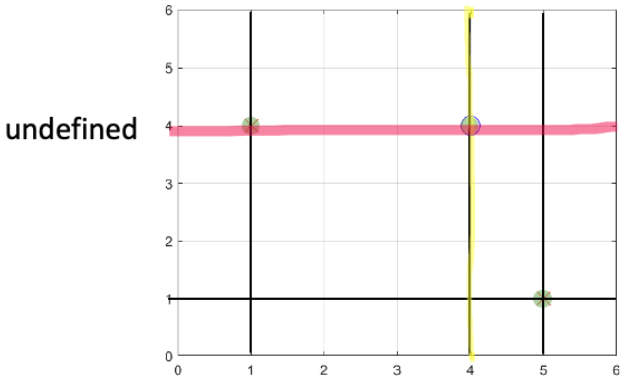
   b) L_1:
   $closest : |x_{21}|_1 = 6$
   $|x_{31}|_1 = 7.6$
   $|x_{41}|_1 = 7.5$

   c) L_2:
   $closest : |x_{21}|_2 = 3.7255872$
   $|x_{31}|_2 = 4.50998891$
   $|x_{41}|_2 = 4.65725241$

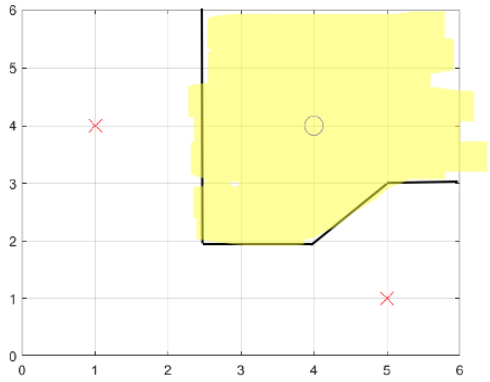
   d) L_{inf}:
   $closest : |x_{21}|_{inf} = 2.7$
   $|x_{31}|_{inf} = 3.8$
   $|x_{41}|_{inf} = 3.6$

2. Draw the 1-Nearest Neighbor decision boundaries with the given norms and lightly shade the o region:

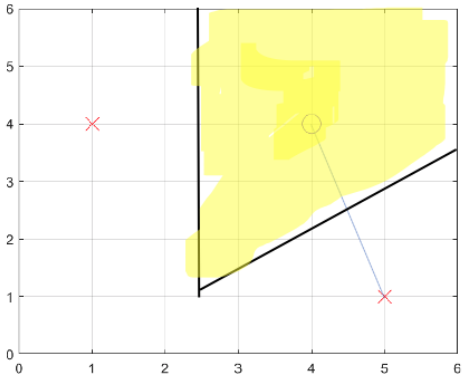
a) L_0



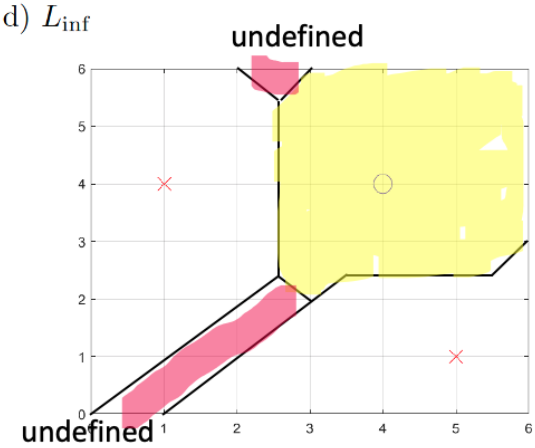
b) L_1



c) L_2



d) L_{inf}



2 Decision trees

1. Concrete sample training data.

   (a) The sample entropy $H(Y)$ is ....

$$H(Y) = \sum_{j=1}^{m} (P(y_j) * log_2(P(y_j)))$$
$$= -(\frac{22}{40} * \log_2(\frac{22}{40}) + \frac{18}{40} * \log_2(\frac{18}{40}))$$
$$= 0.9927744539878083$$

(b) The information gains $IG(X_1)$ and $IG(X_2)$ are

$$P(Y=+, X_1=T) = 9/40, P(Y=+, X_1=F) = 13/40$$
$$P(Y=-, X_1=T) = 10/40, P(Y=-, X_1=F) = 8/40$$
$$P(Y=-, X_2=T) = 9/40, P(Y=-, X_2=F) = 9/40$$
$$P(Y=+, X_2=T) = 7/40, P(Y=+, X_2=F) = 15/40$$

$$P(X_1=T) = 19/40, P(X_1=F) = 21/40$$

$$P(Y=+ \mid X_1=T) = \frac{P(X_1=T, Y=+)}{P(X_1=T)} = \frac{9}{19}$$
$$P(Y=- \mid X_1=F) = \frac{P(X_1=F, Y=+)}{P(X_1=F)} = \frac{8}{21}$$
$$P(Y=- \mid X_1=T) = \frac{P(X_1=T, Y=-)}{P(X_1=T)} = \frac{10}{19}$$
$$P(Y=+ \mid X_1=F) = \frac{P(X_1=F, Y=+)}{P(X_1=F)} = \frac{13}{21}$$

$$IG(X_1) = H(Y) - H(Y \mid X_1)$$
$$= 0.9927744539878083 - \sum_{m,n} P(X_{1m}, Y_n) * \log_2 P(Y_n \mid X_{1m})$$

$$\sum_{m,n} P(X_{1m}, Y_n) * \log_2 P(Y_n \mid X_{1m}) =$$
$$P(X_1=T, Y=+) * \log_2 P(Y=+ \mid X_1=T)$$
$$+P(X_1=T, Y=-) * \log_2 P(Y=- \mid X_1=T)$$
$$+P(X_1=F, Y=+) * \log_2 P(Y=+ \mid X_1=F)$$
$$+P(X_1=F, Y=-) * \log_2 P(Y=- \mid X_1=F)$$
$$= 9/40 * \log_2(9/19) + 10/40 * \log_2(10/19) + 13/40 * \log_2(13/21) + 8/40 * \log_2(8/21)$$
$$= 0.9773741584023364$$

$$H(Y) - H(Y \mid X_1) = 0.015400295585471846$$

$$P(Y = + \mid X_2 = T) = \frac{P(X_2 = T, Y = +)}{P(X_2 = T)} = \frac{7}{16}$$
$$P(Y = - \mid X_2 = F) = \frac{P(X_2 = F, Y = +)}{P(X_2 = F)} = \frac{9}{24}$$
$$P(Y = - \mid X_2 = T) = \frac{P(X_2 = T, Y = -)}{P(X_2 = T)} = \frac{9}{16}$$
$$P(Y = + \mid X_2 = F) = \frac{P(X_2 = F, Y = +)}{P(X_2 = F)} = \frac{15}{24}$$

$$P(X_2 = T) = 16/40, P(X_2 = F) = 24/40$$

$$IG(X_2) = H(Y) - H(Y \mid X_2)$$
$$= 0.9927744539878083 - \sum_{m,n} P(X_{2m}, Y_n) * \log_2 P(Y_n \mid X_{2m})$$

$$\sum_{m,n} P(X_{2m}, Y_n) * \log_2 P(Y_n \mid X_{2m}) =$$
$$P(X_2 = T, Y = +) * \log_2 P(Y = + \mid X_2 = T)$$
$$+P(X_2 = T, Y = -) * \log_2 P(Y = - \mid X_2 = T)$$
$$+P(X_2 = F, Y = +) * \log_2 P(Y = + \mid X_2 = F)$$
$$+P(X_2 = F, Y = -) * \log_2 P(Y = - \mid X_2 = F)$$
$$= 7/40 * \log_2(7/16) + 9/40 * \log_2(9/16) + 15/40 * \log_2(15/24) + 9/40 * \log_2(9/24)$$
$$= 0.968140165070378$$

$$H(Y) - H(Y \mid X_2) = 0.024634288917430247$$

(c) The decision tree that would be learned is shown in Figure 2.

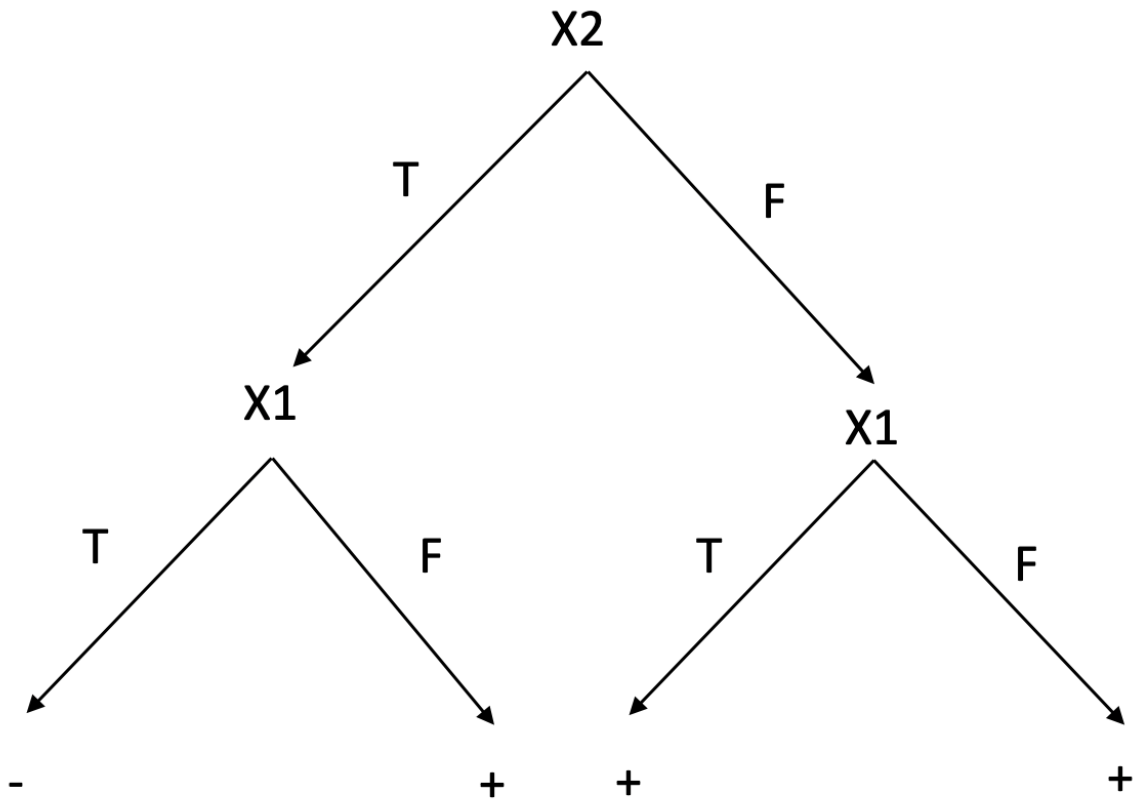


Figure 1: The decision tree that would be learned.

2. Information gain and KL-divergence.

   (a) If variables X and Y are independent, is $IG(x, y) = 0$? If yes, prove it. If no, give a counter example.

$$
\begin{aligned}
P(x, y) &= P(x) * P(y) \Leftrightarrow P(x) and P(y) are independent \\
IG(x, y) &= \sum_x \sum_y P(x, y) \log_2 (\frac{P(x) * P(y)}{P(x) * P(y)}) \\
&= \sum_x \sum_y P(x, y) * 0 \\
&= 0
\end{aligned}
$$

   (b) Prove that $IG(x, y) = H[x] - H[x \mid y] = H[y] - H[y \mid x]$, starting from the definition in terms of

KL-divergence:

$$
\begin{aligned}
IG(x,y) &= KL\left(p(x,y)||p(x)p(y)\right) \\
&= -\sum_x \sum_y P(x,y)\log_2(\frac{P(x)*P(y)}{P(x,y)}) \\
&= \sum_x \sum_y P(x,y)\log_2(\frac{P(x,y)}{P(x)*P(y))} \\
&= \sum_x \sum_y P(x,y)(\log_2(\frac{P(x,y)}{P(x)}) - \log_2(P(y))) \\
&= -\sum_x \sum_y P(x,y)\log_2(P(y)) + \sum_x \sum_y P(x,y)\log_2(\frac{P(x,y)}{P(x)}) \\
&= -\sum_y P(y)\log_2(P(y)) + \sum_x \sum_y P(x,y)\log_2(\frac{P(x,y)}{P(x)}) \\
&= H[y] - H[y \mid x] \\
&= -\sum_x \sum_y P(x,y)\log_2(\frac{P(x)*P(y)}{P(x,y)}) \\
&= \sum_x \sum_y P(x,y)\log_2(\frac{P(x,y)}{P(x)*P(y))} \\
&= \sum_x \sum_y P(x,y)(\log_2(\frac{P(x,y)}{P(y)}) - \log_2(P(x))) \\
&= -\sum_x \sum_y P(x,y)\log_2(P(x)) + \sum_x \sum_y P(x,y)\log_2(\frac{P(x,y)}{P(y)}) \\
&= -\sum_x P(x)\log_2(P(x)) + \sum_x \sum_y P(x,y)\log_2(\frac{P(x,y)}{P(y)}) \\
&= H[x] - H[x \mid y]
\end{aligned}
$$

3 High dimensional hi-jinx

1. Intra-class distance.

$$
\begin{aligned}
&\mathbf{E}[(X - X')^2] \\
&= \mathbf{E}[(X)^2 - 2*(X*X') + (X')^2] \\
&= \mathbf{E}[(X)^2] - 2*\mathbf{E}[(X*X')] + \mathbf{E}[(X'^2] \\
&= \mathbf{E}[(X)^2] - 2*\mathbf{E}[X]\mathbf{E}[X'] + \mathbf{E}[(X')^2] \\
&= \mu^2 + \sigma^2 - 2*\mu*\mu + \mu^2 + \sigma^2 = 2\sigma^2
\end{aligned}
$$

2. Inter-class distance.

$$\begin{aligned}
&\mathbf{E}[(X - X')^2] \\
&= \mathbf{E}[(X)^2 - 2 * (X * X') + (X')^2] \\
&= \mathbf{E}[(X)^2] - 2 * \mathbf{E}[(X * X')] + \mathbf{E}[(X'^2] \\
&= \mathbf{E}[(X)^2] - 2 * \mathbf{E}[X]\mathbf{E}[X'] + \mathbf{E}[(X')^2] \\
&= \mu_1^2 + \sigma^2 - 2 * \mu_1 * \mu_2 + \mu_2^2 + \sigma^2 \\
&= 2\sigma^2 + (\mu_1 - \mu_2)^2
\end{aligned}$$

3. Intra-class distance, m-dimensions.

$$\begin{aligned}
&\mathbf{E}[\sum_{j=1}^{m}(X_j - X'_j)^2] \\
&= \mathbf{E}[\sum_{j=1}^{m} X_j^2 - 2 * X_j * X'_j + X_j'^2] \\
&= \sum_{j=1}^{m}(\mathbf{E}[X_j^2] - 2 * \mathbf{E}[X_j * X'_j] + \mathbf{E}[X_j'^2]) \\
&= \sum_{j=1}^{m}(\mathbf{E}[X_j^2] - 2 * \mathbf{E}[X_j]\mathbf{E}[X'_j] + \mathbf{E}[X_j'^2]) \\
&= \sum_{j=1}^{m}(2\sigma^2 + (\mu_{1j} - \mu_{2j})^2) \\
&= 2m\sigma^2
\end{aligned}$$

4. Inter-class distance, m-dimensions.

$$\begin{aligned}
&\mathbf{E}[\sum_{j=1}^{m}(X_j - X'_j)^2] \\
&= \mathbf{E}[\sum_{j=1}^{m} X_j^2 - 2 * X_j * X'_j + X_j'^2] \\
&= \sum_{j=1}^{m}(\mathbf{E}[X_j^2] - 2 * \mathbf{E}[X_j * X'_j] + \mathbf{E}[X_j'^2]) \\
&= \sum_{j=1}^{m}(\mathbf{E}[X_j^2] - 2 * \mathbf{E}[X_j]\mathbf{E}[X'_j] + \mathbf{E}[X_j'^2]) \\
&= \sum_{j=1}^{m}(2\sigma^2 + (\mu_{1j} - \mu_{2j})^2) \\
&= \sum_{j=1}^{m}((\mu_{1j} - \mu_{2j})^2) + 2m\sigma^2
\end{aligned}$$

5. The ratio of expected intra-class distance to inter-class distance is: .... As m increases towards ∞,

this ratio approaches ....

$$
\begin{aligned}
&= \lim_{m\to\infty} \frac{2m\sigma^2}{\sum_{j=1}^{m}((\mu_{1j}-\mu_{2j})^2)+2m\sigma^2} \\
&= \lim_{m\to\infty} \frac{2m\sigma^2}{(\mu_{11}-\mu_{21})^2+2m\sigma^2} \\
&= 1
\end{aligned}
$$

The value approaches to 1. The phenomenon means that when the number of features is close to infinity and only one distribution is informative, the expected distance of inter-class samples and the expected distance of intra-class samples are the same. That means the multi-class problem downgrades to single class problem.

4 K-nearest neighbors Classification (Programming)

1. How does having a larger dataset might influence the performance of KNN?

   The curse of dimensionality might make the points in the dataset much more sparse

2. Tabulate your results in Table 2 for the **validation set**.

K	**Norm**	**Accuracy (%)**
3	L1	0.7217391304347827
3	L2	0.6956521739130435
3	L-inf	0.7304347826086957
5	L1	0.7565217391304347
5	L2	0.7652173913043478
5	L-inf	0.7304347826086957
7	L1	0.7304347826086957
7	L2	0.7739130434782608
7	L-inf	0.7304347826086957

Table 1: Accuracy for the KNN classification problem on the validation set

3. Finally, mention the best K and the norm combination you have settled upon from the above table and report the accuracy on the test set using that combination.

K	**Norm**	**Accuracy (%)**
7	L2	0.7142857142857143

Table 2: Accuracy for the KNN classification problem on the test set

5 Decision Trees (Programming)

5.1 Part 1: Effects of Dataset Size on Performance

1. Report the training, validation, and test accuracies on the full and partial datasets below. Note that this portion will be graded by the Autograder.

Accuracy Scores		
	Full Dataset	Small Dataset
Training Accuracy	1	1
Validation Accuracy	0.7043478260869566	0.7130434782608696
Test Accuracy	0.7532467532467533	0.6753246753246753

2. Which dataset had a higher difference between training and test accuracy? Briefly explain why.

   The small dataset has larger difference. Because the model was trained with the small dataset having less information

5.2 Part 2: Effects of Dataset Size on Performance

1. Report the chosen hyperparameters for the complete and partial set below. Note that this section will be graded by the Autograder.

Grid Search Chosen Hyperparameters		
	Full Dataset	Small Dataset
Tree Depth	3	1
Max Leaf Nodes	4	2

2. Did the small dataset have higher or lower chosen hyperparameter values than the full dataset? Briefly explain why.

   The small dataset model has lower chosen hyperparameter because the smaller dataset does not provide the information as much as the full dataset does. In other word, the smaller dataset might be easier to fit with simpler model than the full dataset is.

5.3 Part 3: Retrain Decision Tree and Plot Hyperparameter Search

1. Report the train, validation, and test accuracies after retraining the decision tree with the new hyperparameters. Also paste in the values for the training and validation scores lists when varying the max leaf node count hyperparameter.

Retrained Decision Tree Performance for Small Dataset	
	Score
Training Accuracy	0.8159722222222222
Validation Accuracy	0.7739130434782608
Test Accuracy	0.7142857142857143

Training and Validation List Values	
	List
Training	[0.7743055555555556, 0.7743055555555556, 0.7743055555555556, 0.7743055555555556, 0.7743055555555556, 0.7777777777777778, 0.7951388888888888, 0.8159722222222222, 0.8159722222222222]
Validation	[0.7478260869565218, 0.7478260869565218, 0.7478260869565218, 0.7478260869565218, 0.7478260869565218, 0.7478260869565218, 0.7130434782608696, 0.7739130434782608, 0.7739130434782608]

2. How did the training accuracy and testing accuracy change after tuning compared to before? Briefly explain why.

   The validation performance of model after tuning is better since we picked the hyperparameters as the validation scores went to its maximum

   The test performance is worse since we only cared about the validation scores and the optimal model didn't have sufficient information of the dataset as the model trained with full dataset did have

3. Paste the plot of training and validation scores with different leaf count values on the small dataset. Explain any trends or patterns with the plot within validation and training scores and briefly explain why.

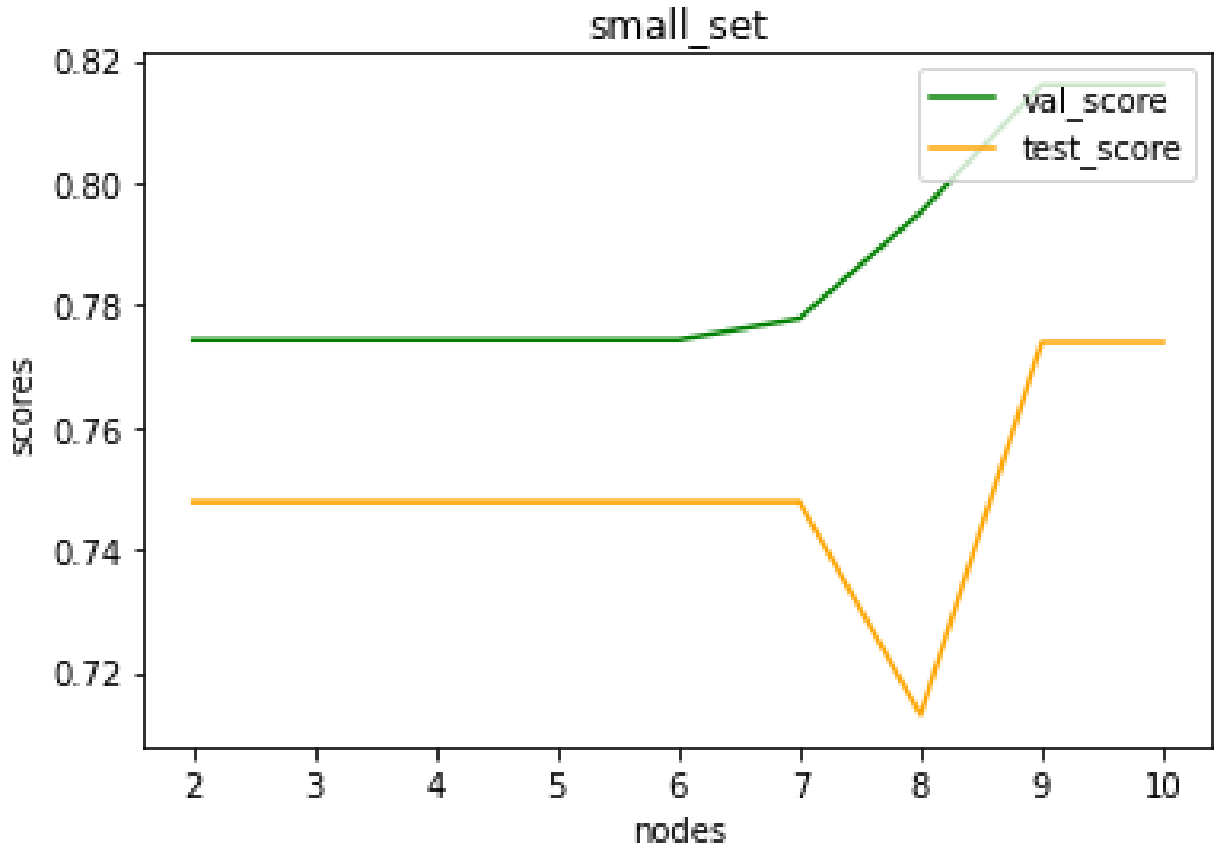


Figure 2: The decision tree that would be learned.

As the model gets more complex, the model is able to fit more perfectly to the small dataset and the validation scores go higher.

The test score could have gone down since the better the model fits to a specific dataset, the higher variance we might get from the test dataset

6 Feature Scaling Effects (Programming)

1. Report the training and testing accuracies for unstandardized and standardized data for both Decision Trees and KNNs using their default hyperparameter values.

Scores for Unstandardized and Standardized Data				
	KNN Unscaled	KNN Scaled	DT Unscaled	DT Scaled
Training Accuracy	0.7899305555555	0.8177083333333	1.0	1.0
Test Accuracy	0.7012987012987	0.8181818181818	0.7532467532467	0.7532467532467

2. What happens to performance when we use standardization for data with decision trees? What about KNN? Briefly explain why each happened.

The KNN has better performance since the standardization eliminated the dominance of some features with larger range.

The Decision Tree, which uses only if/else to split the tree, is not affected since the standardization does not change the relative sizes(probabilities) of the features. So the DT after standardization has the same result as it was.