CS M146, Winter 2019
Problem Set 1: Decision Trees
Due Jan 28, 2019

Tian Ye
Collaborator: Derek Chu

January 26, 2019

1 Maximum Likelihood Estimation

(a) **Solution:** The likelihood estimation is given by the following:

$$L(\theta) = \prod_{i=1}^{n} P_\theta(X_i) \tag{1}$$

$$= \prod_{i=1}^{n} \theta^{X_i}(1-\theta)^{1-X_i} \tag{2}$$

$$= \theta^{x_1}(1-\theta)^{x_0} \tag{3}$$

Where x_1 counts the number of cases which $X_i = 1$ and x_0 counts the number of cases which $X_i = 0$.
The order of the individual random variables X_i do not matter as they are independent from one another.

(b) **Solution:** Taking the log likelihood of the previous expression:

$$\ell(\theta) = \log(\theta^{x_1}(1-\theta)^{x_0}) \tag{4}$$

$$= x_1 \log(\theta) + x_0 \log(1-\theta) \tag{5}$$

Taking the first and second derivatives of $\ell(\theta)$ with respect to θ:

$$\ell'(\theta) = \frac{x_1}{\theta} - \frac{x_0}{1-\theta} \tag{6}$$

$$\ell''(\theta) = -\frac{x_1}{\theta^2} - \frac{x_0}{(1-\theta)^2} \tag{7}$$

$$\tag{8}$$

Since $\ell''(\theta) < 0$, the function is always concave down.
We can therefore set $\ell'(\theta) = 0$ to solve for the MLE:

$$\theta_{MLE} = \frac{x_1}{x_1 + x_0} \tag{9}$$

(c) **Solution:**

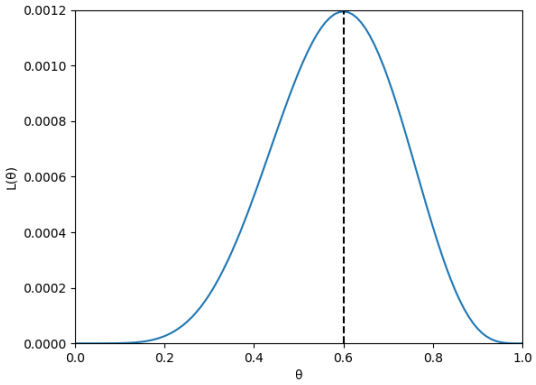

Figure 1: The figure above does agree with Equation 9 given in the previous section; we can see this as the maximum is at $\theta = 0.6$, which corresponds with $\frac{6}{4+6}$.

(d) **Solution:**

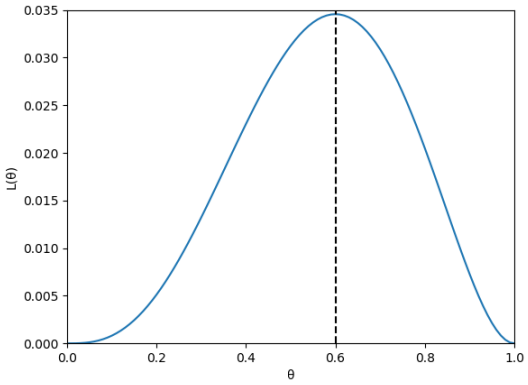

Figure 2: By decreasing the number of data points while maintaining the ratio of 1s to 0s, the likelihood plot keeps the same MLE while having a wider spread and a lower likelihood. This is because we have less data; therefore we are less confident.

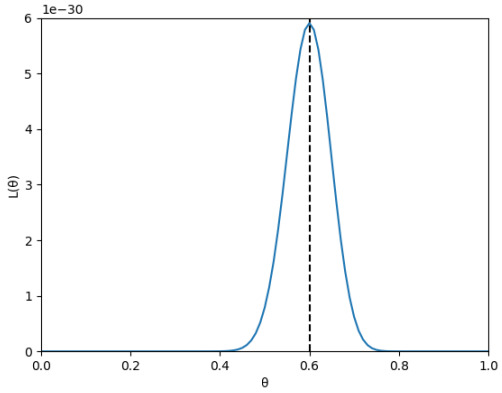

Figure 3: By increasing the number of data points while maintaining the ratio of 1s to 0s, the likelihood plot keeps the same MLE while having a narrower spread and a higher likelihood. This is because we have more data; therefore we are more confident.

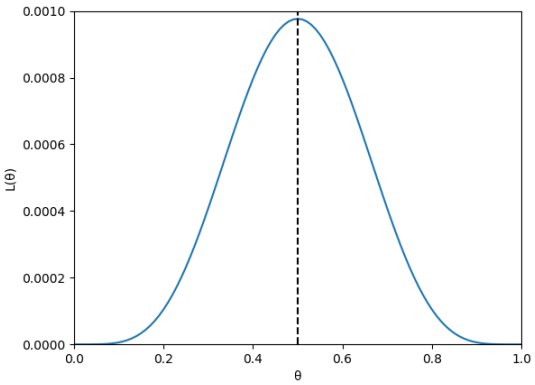

Figure 4: By maintaining the same number of data points while changing the ratio of 1s to 0s, the likelihood plot shifts the MLE while maintaining the spread and likelihood. Intuitively, this is because we have the same number of data points.

2 Splitting Heuristic for Decision Trees

(a) **Solution:** The best 1-leaf decision tree makes an error $\frac{1}{8}$ of the time. The decision tree is as follows: $Y = 0$ if and only if $X_1 = X_2 = X_3 = 0$. This leaves us with 2^{n-3} remaining binary vectors; hence the error is $\frac{2^{n-3}}{2^n} = \frac{1}{8}$.

(b) **Solution:** There does not exist a split that reduces the number of mistakes. If we split on X_1, X_2, or X_3, it will create a tree in which one leaf only contains 1s and the other leaf contains 1s with a proportion of $\frac{3}{4}$. Splitting on a value of $n \geq 4$ will create two leaves where the proportion of 1s is $\frac{7}{8}$. In both cases the tree will always predict 1, leaving an error rate of $\frac{1}{8}$.

(c) **Solution:** $\frac{1}{8}\log(8) + \frac{7}{8}\log(\frac{8}{7}) = 0.543$

(d) **Solution:** By splitting on X_1, X_2, or X_3, we can reduce the entropy on the output Y. The new entropy following the split is as follows:
$\frac{1}{2}[\frac{1}{4}\log(4) + \frac{3}{4}\log(\frac{4}{3})] = 0.406$
This reduces the entropy on Y by 0.137.

3 Entropy and Information

(a) **Solution:** We know that the entropy must fall within the range of $0 \leq H(S) \leq 1$ using the following:

$$B(q) = -q\log(q) - (1-q)\log(1-q) \tag{10}$$

For a set containing p positive examples and n negative examples, $H(S) = B(\frac{p}{n+p})$. We will show that $0 \leq H(S)$ by setting $p = 0$ and n to some value greater than 0:

$$B(\tfrac{0}{n+0}) = -q\log(q) - (1-q)\log(1-q) \tag{11}$$
$$B(0) = 0\log(0) - (1-0)\log(1-0) \tag{12}$$
$$H(S) = 0 \tag{13}$$

For any value of p such that $0 \leq p \leq n$, $H(S)$ increases and approaches 1 as q increases towards $\frac{1}{2}$. We know this from the first and second derivatives of $B(q)$:

$$B(q)' = \log(1-q) - \log(q) \tag{14}$$
$$B(q)'' = -\frac{1}{q} - \frac{1}{1-q} \tag{15}$$

Since q is given by $(\frac{p}{n+p})$, q is bound between 0 and 1. Furthermore, we know that the function maximizes at $q = \frac{1}{2}$ as the first derivative at that value is 0 and the second derivative is strictly negative for the interval $0 \leq q \leq 1$. Once $p = n$, we yield the following:

$$B(\tfrac{n}{2n}) = -q\log(q) - (1-q)\log(1-q) \tag{16}$$
$$B(\tfrac{1}{2}) = -\tfrac{1}{2}\log(\tfrac{1}{2}) - (1-\tfrac{1}{2})\log(1-\tfrac{1}{2}) \tag{17}$$
$$H(S) = 1 \tag{18}$$

As p increases past n, the value q once again begins to decrease towards 0, and as such, $H(S)$ decreases back towards 0. Hence, $0 \leq H(S) \leq 1$.

(b) **Solution:** Given that $S_1, S_2, S_3, ..., S_k$ are disjoint subsets of X_j and that the ratio $\frac{p_k}{p_k+n_k}$ is the same for all k, we can state the following:

$$H(S_k) = B(\tfrac{p_k}{p_k+n_k}) \tag{19}$$

$$H(S|X_j) = H(S_1)\tfrac{p_1+n_1}{p+n} + ... + H(S_k)\tfrac{p_k+n_k}{p+n} \tag{20}$$

$$H(S|X_j) = H(S)\tfrac{p_1+n_1+p_2+n_2+...+p_k+n_k}{p+n} \tag{21}$$

$$H(S|X_j) = H(S)\tfrac{p+n}{p+n} \tag{22}$$

$$H(S|X_j) = H(S) \tag{23}$$

$$H(S|X_j) = B(\tfrac{p}{n+p}) \tag{24}$$

Therefore we can see that splitting the set S by the attribute X_j gains 0 information:

$$H(S|X_j) - H(S) = B(\tfrac{p}{n+p}) - B(\tfrac{p_k}{p_k+n_k}) = 0 \tag{25}$$

4 k-Nearest Neighbor

(a) **Solution:** Setting a value of $k = 1$ minimizes the training set error, achieving a set error of 0. This is because the closest training set data point to a given point is itself, causing it to achieve perfect accuracy. This training set is not a reasonable estimate of test set error, as this value of k essentially causes the algorithm to "memorize" the training set.
Achieving an error of 0 is impossible if the test set is distinct from the training set, as the program would not have "learned" anything from the test set due to overfitting. It also makes it very susceptible to noise.
Consequently, given a distinct testing set from the training set, the error would be very high with a value of $k = 1$.

(b) **Solution:** $k = 5$ minimizes the LOOCV error as it guarantees that the two groups of five circles and five astericks will be correctly classified. As a result, only the two groupings of two circles and two astericks will fail.
This leads to an error rate of $\frac{2}{7}$. Setting $k = 5$ is a better measure of test set performance as it is nonzero and therefore realistic.

(c) **Solution:** The LOOCV error for $k = 1$ is $\frac{5}{7}$, while the LOOCV error for $k = 13$ is 1.
The disadvantage of using a large k is that the predicted value for a given data point depends on the remaining data points; for example, in this instance, setting $k = 13$ will cause an error rate of 100% as there will be 6 remaining data points of the selected point and 7 remaining data points for the other selected point.
The disadvantage of using too small a data point is that the classification of a given point will be based solely on the nearest data value. This will cause issues in patterns such as a checkerboard, where related points lie along diagonals but the closest and unrelated points lie along the rows and columns; consequently, this will cause high amounts of error.

5 Programming Exercises: Applying Decision Trees

(a) **Solution:**

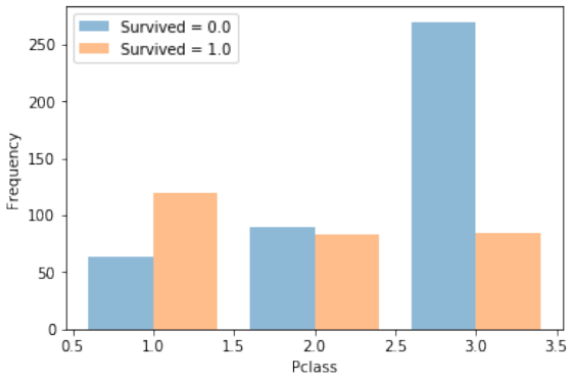

Figure 5: We can note that for ticket class 1, the number of survivors was approximately double that of those that did not survive. For ticket class 2, the number of survivors compared to fatalities was approximately the same. For ticket class 3, the number that did not survive was nearly three times that did survive. Thus it can be concluded that the wealthier the passenger, the higher the likelihood of survival.

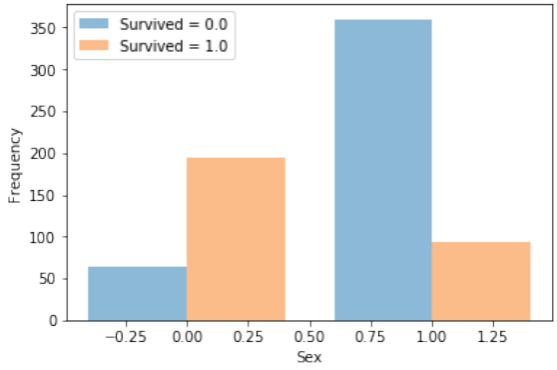

Figure 6: We can note that for males, the survival rate was significantly lower than that of females. Namely, approximately 200 females survived while 50 did not, while for males 350 did not survive while only about 100 did.

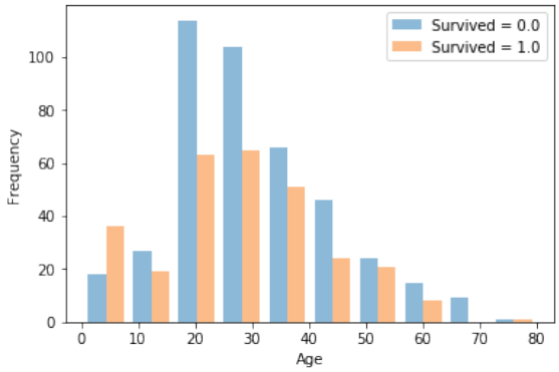

Figure 7: We can note that with the exception of the youngest age group, those younger than 10 years old, more passengers perished than those that survived. The data set as a whole is right skewed, and the highest fatality rates were generally found for those in the age range of 20 - 40 years old.

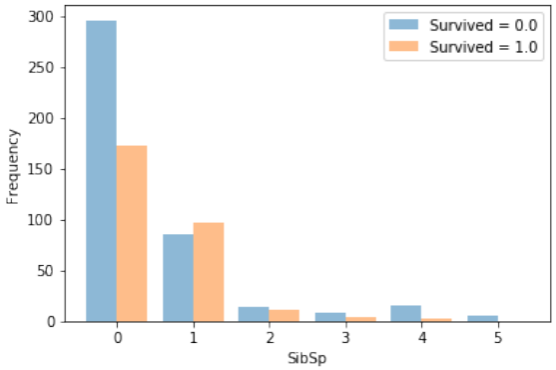

Figure 8: We can note that the data as a whole is heavily skewed right, with those having 1 other sibling or spouse having a higher survival rate than the other categories. The other sibling-spouse categories had the similar surivival rates as the 0 group, albeit with much lower frequencies.

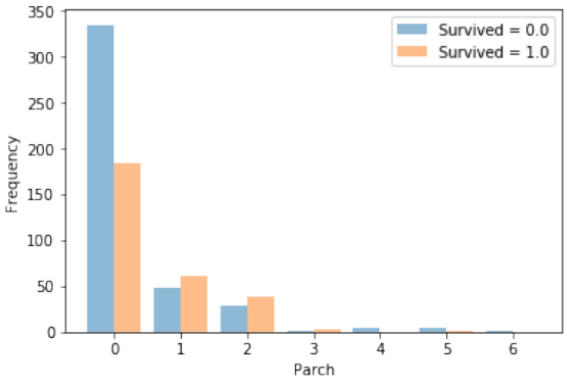

Figure 9: We can note that the data as a whole is heavily skewed right, with those having 1 or 2 other parents or children having a higher survival rate than those with none. Those with none suffered approximately a 66% mortality rate, whereas those with 1 or 2 had more surviving than perishing. The other parent-children categories have nearly negligible frequencies.

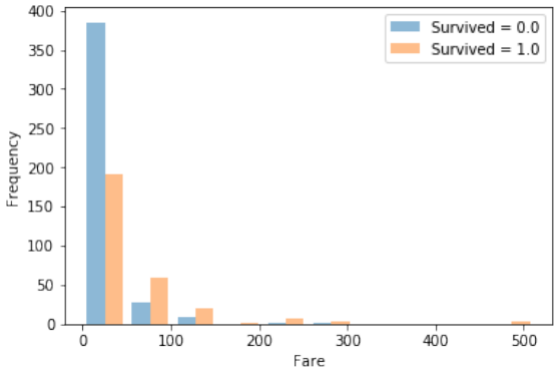

Figure 10: We can note that the data as a whole is heavily skewed right, and that those that payed less than $100 as a whole had a much lower survival rate than those who did not. Notably, those in the $50 category suffered a 66% mortality rate while those above suffered a 33% mortality rate.

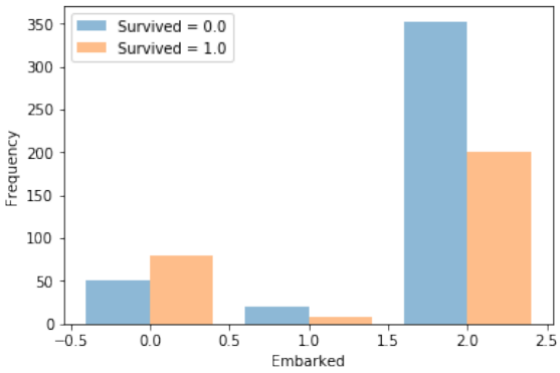

Figure 11: In this graph, the three x-axis values are Cherbourg, Queenstown, and Southampton. Passengers from Cherbourg enjoyed the highest survival rate, while those from Queenstown and Southampton had more fatalities than survivals. However, the number of survivals from Southampton was higher than that of Cherbourg, owing to the much higher number of passengers departing from Southampton.

(b) **Solution:** After implementing RandomClassifier, I achieved the training error value of 0.485.

(c) **Solution:** Using the entropy criterion for information gain, the training error of the DecisionTreeClassifier is 0.014.

(d) **Solution:**

Table 1: Investigating Various Classifiers

Model Type	train_err	test_err
Majority	0.404	0.407
Random	0.489	0.487
Decision Tree	0.012	0.241

(e) **Solution:**

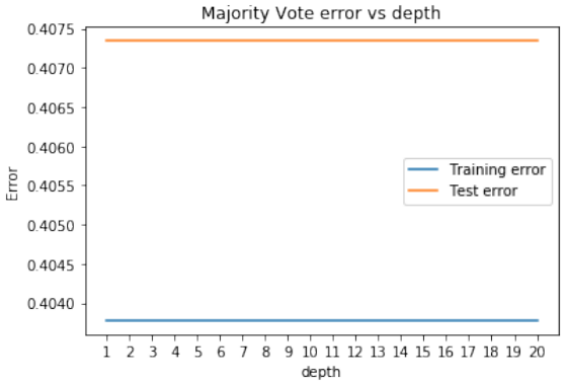

Figure 12: We can tell by the figure above that the Majority Vote Classifier does not depend on depth; therefore there is no overfitting and no best value for depth.

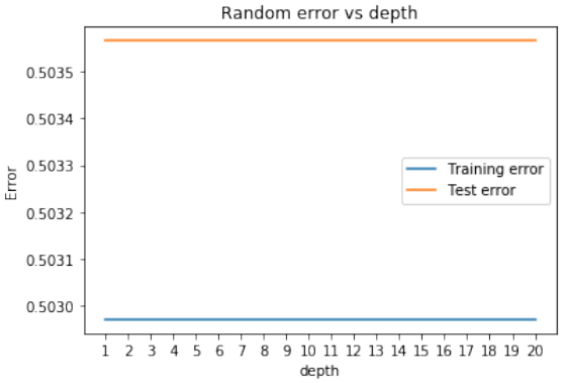

Figure 13: We can tell by the figure above that the Random Classifier does not depend on depth; therefore there is no overfitting and no best value for depth.

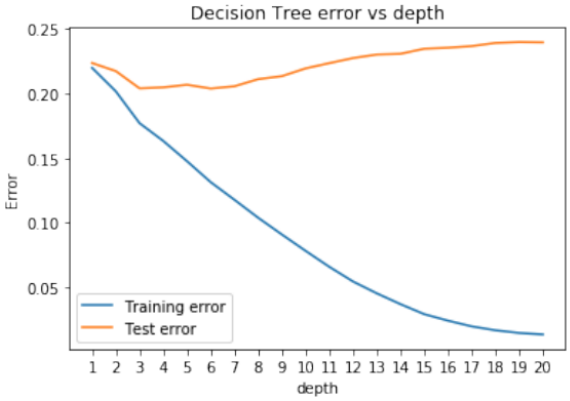

Figure 14: Viewing the figure above, we can see clear signs of overfitting: as the depth increases, the training error steadily decreases while the test error first decreases before increasing again. This shows that as the decision tree becomes more complex as the depth increases, it becomes less adapatable and is unable to generalize new data and examples in the testing set, leading to higher error rates. The ideal depth for the Decision Tree Classifier is 6. At this value, the error is minimized at a value of 0.203776223776.

(f) **Solution:**

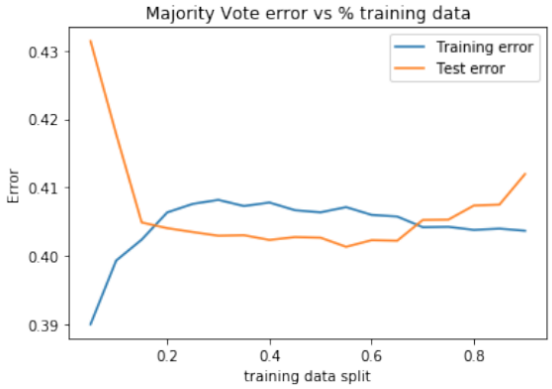

Figure 15: Viewing the figure above, we can see that until 20%, each increase in the training data split leads to a lower test_err and a higher train_err until they center around 40% error for both. Afterwards, they remain relatively constant. This is because as the training set increases in size, it becomes more representative of the actual data, leading to higher accuracy in relation to the test error.

Figure 16: Viewing the figure above, we can see that as the training set increases up to 20% error, the training error increases steadily until it reaches around 50%. As the training set increases up to 95%, both the training error and the test error fluctuate between 50% and 51%. This can be attributed to the fact that Random Classifier will classify at random examples based on a percentage of the entire training set: the random nature of the data assignment to classes will keep random error fluctuating across a center error value.

Figure 17: Viewing the figure above we can see that as the training set increases, the training error decreases while the test error decreases, with nearly 10% change for both test and training error. This is attributed to the fact that as the training set increases, the decision tree will increase in depth. This creates more options for the training set examples to be matched with, increasing error. At the same time, the tree improves at generalizing, decreasing test error.