Christopher Nyi
DISC 1C
1/28/19

CS M146 Problem Set 1

① Maximum Likelihood Estimation

$P(X_i = 1) = \theta$ and $P(X_i = 0) = 1 - \theta$

$$\hat{\theta}_{MLE} = \arg\max_{\theta} L(\theta)$$

A) $L(\theta) = P(X_1, \dots X_n; \theta)$

Bernoulli Dist
↳ $P_\theta(x) = \theta^x (1-\theta)^{1-x}$

$$L(\hat{\theta}) = P_{\hat{\theta}}(X_1, \dots, X_n)$$

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

$$L(\hat{\theta}) = \prod_{i=1}^{n} \hat{\theta}^{X_i} (1-\hat{\theta})^{1-X_i}$$

→ Let a_1 = # of times $X_i = 1$
a_0 = # of times $X_i = 0$

$$\boxed{L(\hat{\theta}) = \hat{\theta}^{a_1} (1-\hat{\theta})^{a_0}}$$

No, the likelihood function does not depend on the order of the random variables because the likelihood function is just a product of a sequence of values, and since multiplication is commutative, the order does not matter.

B) log likelihood $\ell(\theta) = \log(L(\theta))$

$$\ell(\theta) = \log(L(\theta)) \qquad L(\hat{\theta}) = \hat{\theta}^{a_1} (1-\theta)^{a_0}$$

$$= \log(\theta^{a_1} (1-\theta)^{a_0})$$

$$= \log(\theta^{a_1}) + \log((1-\theta)^{a_0})$$

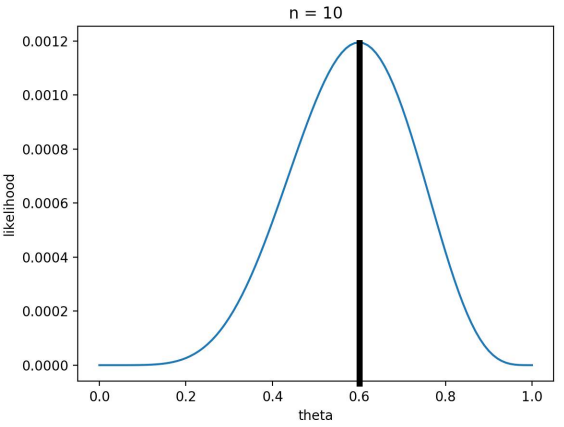
$$\boxed{\ell(\theta) = a_1 \log(\theta) + a_0 \log(1-\theta)}$$

$$\boxed{\ell'(\theta) = \frac{a_1}{\theta} + \frac{a_0}{1-\theta}}$$

$$\ell''(\theta) = \frac{-a_1}{\theta^2} - \frac{a_0}{(1-\theta)^2}$$

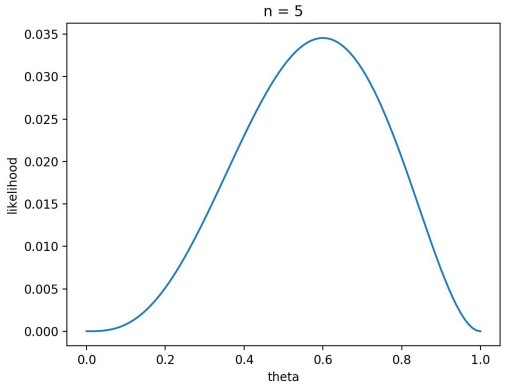
$$\boxed{\ell''(\theta) = -\left(\frac{a_1}{\theta^2} + \frac{a_0}{(1-\theta)^2}\right)}$$

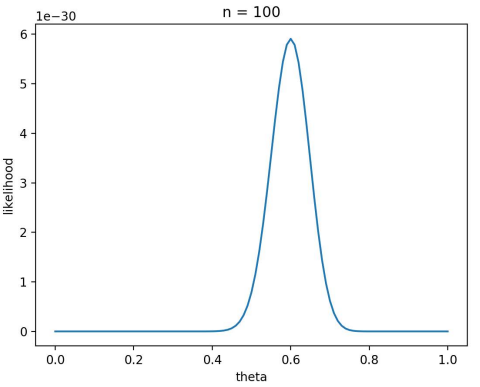
1C)

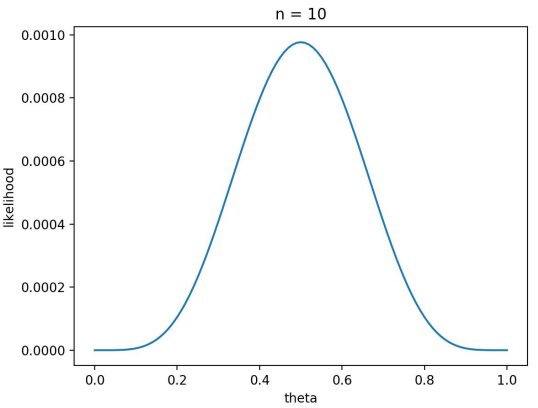


Yes, the answer agrees with the closed form answer.

1D)







The maximum likelihood estimates are equal to the proportion of 1's out of all the data points. Thus, for the first two functions (n=5 & n=100) the MLE's are both 0.6 and are asymmetric. On the other hand, the last function (n=10) the MLE is 0.5 and is symmetric. As n increases, the function becomes sharper at the peak and the range of values under the curve decreases.

(2) Splitting Heuristic for Decision Trees

A) Given that x_1, x_2, and x_3 all have to be equal to 0, there would be $n-3$ features that do not matter whether $x_i = 1$ or $x_i = 0$. Since there are 2 possible options for what these features can be, this tells us that there are 2^{n-3} samples in which $y = 0$. The best 1-leaf decision tree would always predict $y = 1$ since $y=1$ in $2^n - 2^{n-3}$ samples whereas $y=0$ in only 2^{n-3} samples. Thus the best 1-leaf decision tree will make 2^{n-3} mistakes over 2^n training examples.

$$\frac{2^{n-3}}{2^n} = 2^{n-3-n} = 2^{-3} = \frac{1}{8}$$

The best 1-leaf decision tree will make a mistake once out of every 8 times.

B) No, there is no split that would reduce the number of mistakes by at least one. It does not matter what variable you put at the root because ultimately the proportion of ones in the leaves will be 7/8, still producing a 1/8 error rate.

c) $H[x] = -\sum_{k=1}^{k} P(x=a_k) \log P(x=a_k)$

$= -\left(\frac{1}{8}\log_2\left(\frac{1}{8}\right) + \frac{7}{8}\log_2\left(\frac{7}{8}\right)\right)$

$\boxed{H[x] = 0.544}$

D) Yes, there is a split that reduces entropy of the output y by a non-zero amount. If you split with either x_1, x_2, or x_3, then you will get the entropy to be:

$H[x] = \frac{1}{2}(0) - \frac{1}{2}\left(\frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{3}{4}\log_2\left(\frac{3}{4}\right)\right)$

$= 0 - \frac{1}{2}(-0.811)$

$\boxed{H[x] = 0.406}$

(3) Entropy and Information

$$B(q) = -q \log q - (1-q)\log(1-q) \qquad P(X=1) = q$$

$$H(s) = B\left(\frac{p}{p+n}\right)$$

A) $0 \le H(s) \le 1$, $H(s) = 1$ when $p = n$

$$H(s) = B\left(\frac{p}{p+n}\right)$$

$B_{max}(s)$: global max of $B(s)$

$$B(s) = -s \log s - (1-s)\log(1-s)$$

$$B'(s) = -\log s - \frac{s}{s} + \log(1-s) + \frac{1-s}{1-s}$$

$$B'(s) = -\log s + \log(1-s) = 0$$

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

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

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

$$1 - s = s$$

$$1 = 2s$$

$$s = \frac{1}{2} \Rightarrow$$ The maximum value of s in $H(s)$ is $\frac{1}{2}$

Thus $H_{max}(S) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \left(1-\frac{1}{2}\right)\log\left(1-\frac{1}{2}\right)$

$H_{max}(s) = 1$

On the other hand, $H_{min}(s) = 0$ because entropy is always nonnegative.

This implies that $H_{min} \le H(s) \le H_{max}$

↳ $0 \le H(s) \le 1$ ✓

A) $H(S) = 1$ when $p = n$

$$H(S) = B\left(\frac{p}{p+n}\right) = B\left(\frac{p}{p+p}\right) = B\left(\frac{p}{2p}\right) = B\left(\frac{1}{2}\right)$$

In this case, if $p = n$, then $S = \frac{1}{2}$. Previously, we found that we get our max entropy when $S = \frac{1}{2}$. Thus:

$$H(S) = B\left(\frac{1}{2}\right) = H_{max} \Rightarrow \boxed{H(S) = 1} \checkmark$$

B) k disjoint subsets S_k, with p_k positive
n_k negative examples

$$\frac{p_k}{p_k + n_k}$$

$$Gain(S, X_j) = H(S) - \sum_{k=1}^{k} \frac{|S_k|}{|S|} H(S_k)$$

$$= B\left(\frac{p}{p+n}\right) - \sum_{k=1}^{k} \frac{(p_k + n_k)}{(p+n)} B\left(\frac{p_k}{p_k + n_k}\right)$$

$p = \sum_k p_k$, $n = \sum_k n_k$ } if ratio $\frac{p_k}{p_k+n_k}$ is same for all $k \Rightarrow \frac{p_k}{p_k+n_k} = \frac{p}{p+n}$ $\forall k$

$$Gain(S, X_j) = B\left(\frac{p}{p+n}\right) - \sum_{k=1}^{k} \frac{(p_k + n_k)}{p+n} B\left(\frac{p_k}{p_k+n_k}\right)$$

$$= B\left(\frac{p}{p+n}\right) - \frac{(p+n)}{p+n} B\left(\frac{p}{p+n}\right)$$

$$= B\left(\frac{p}{p+n}\right) - B\left(\frac{p}{p+n}\right)$$

$$\boxed{Gain(S, X_j) = 0}$$

(4) K-Nearest Neighbor

A) $k = 1$ would minimize the training set error for this data set, assuming that each instance can be considered its own neighbor. If each instance of the training set is its own neighbor, then every instance will be classified correctly as its own label, thus producing a training set error of $\emptyset$. Training set error is not a reasonable estimate of test set error, especially given that $k = 1$, because training set instances are being classified based on training data that also includes its own instance, whereas test data instances can be completely random and are classified based on training data that does not include that particular instance.

B) $k = 5$ would minimize the leave-one-out cross-validation error for this data set because in each distinct grouping of instances, there are 7 instances. Thus, we would need a k value greater than half of the instances in that grouping in order for the instance to be classified by a more reasonable majority vote. Thus $k \geq 4$. Since we typically use odd k values, we take $k = 5$. Using a value of $k = 5$ would give us an error of 4/14 because the 2 asterisks near the 5 circles and the 2 circles near the 5 asterisks will be classified incorrectly. Cross-validation is a better measure of test set performance because you are essentially able to partition training data into test data and training data and get a more reasonable estimate of the test set error since the test instance isn't in the training data, and you're performing cross-validation multiple times for a more accurate measurement.

c) The LOOCV error for k = **1** is **10** and the LOOCV error for k = 13 is 14.
Using too small a value of k is bad because it **may be too specific, causing incorrect classification.** On the other hand, using too large a value of k is bad because it is too general, causing underfitting and causing every data instance to be incorrectly classified.

5) Programming Exercise: Applying Decision Trees

a) For the 'Parch' (Parent/Children Aboard) feature, the histograms show a trend indicating that, amongst this demographic, the majority of people who did not survive are people with 0 children. This is likely because parents with children had higher priority to escape on the lifeboats than those who didn't. As the ratio of parent to children increases, the number of those who both survived and didn't survive exponentially decreases. I also noticed a trend that for parents with 1-3 children had higher survival rate than death rate.

   For the 'Age' feature, I noticed that the highest death rates were amongst passengers aged 20-40. However, their survival rate is also relatively higher than those outside that demographic most likely because there were a larger amount of passengers in between the age range of 20-40. There's also a trend showing that for almost all age ranges except 0-10 years there were more people who died than survived.

   For the 'SibSp' (Siblings/Spouses) feature, I noticed an exponentially decreasing trend for both death and survival frequencies, with the highest death rates being those with 0 siblings/spouses and the lowest death rates being those with 5 siblings/spouses. This plot shows that there were many more passengers with 0-1 siblings/spouses than those with 2+ siblings/spouses.

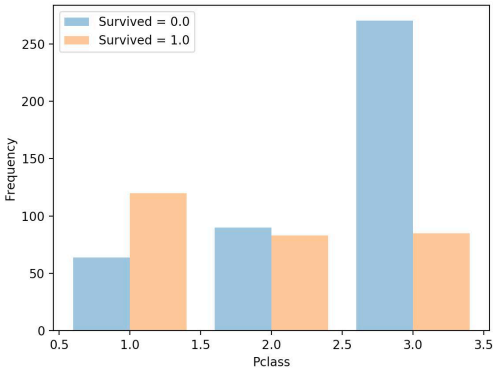
   For the 'Sex' feature, I noticed that the females had a higher survival rate than death rate. On the other hand, males had a higher death rate than survival rate. This is because women were prioritized over the men to get on the lifeboats.

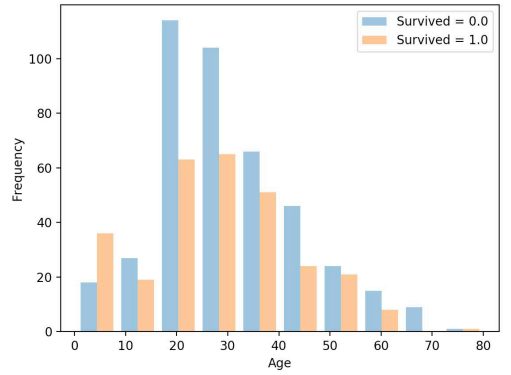
   For the 'Embarked' (Port of Embarkation) feature, assuming that 0 = Cherbourg, 1 = Queenstown, and 2 = Southampton, I noticed that there was a substantially greater number of deaths coming from those who embarked from Southampton. In general, there were the most amount of passengers who embarked from Southampton. Additionally, another interesting trend is that those who embarked from Cherbourg were the only demographic who had a higher survival rate than death rate. This may be due to socioeconomic class and prioritizing those of higher class over those of lower class.

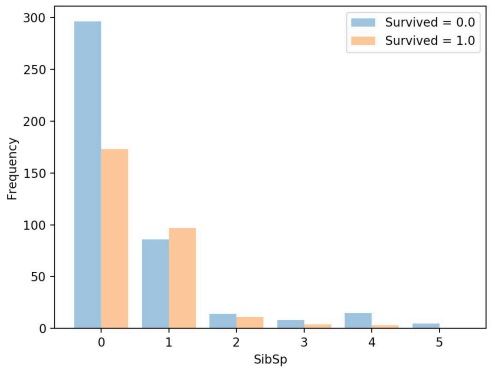
   For the 'Fare' feature, I noticed that those who paid the least amount of fare expenses had the highest death rates. Those who paid between 0-50 were amongst the only demographic who had a higher death rate compared to survival rate. Those who paid 50+ all had a higher survival rate than death rate. This again might be caused by a socioeconomic division between the passengers, preserving those with higher class and leaving lower class members to die.

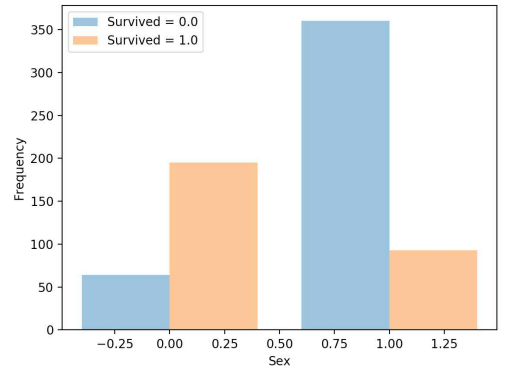
   For the 'Pclass' feature, I noticed that those of the upper class (1) were the only group to have a higher survival rate than death rate. The middle class (2) had a relatively even survival/death ratio but there were still a couple more deaths than survivals. On the contrary, the lower class (3) had an extremely high death rate compared to survival rate, strengthening the idea that the upper class had a higher priority to the lifeboats than the lower class

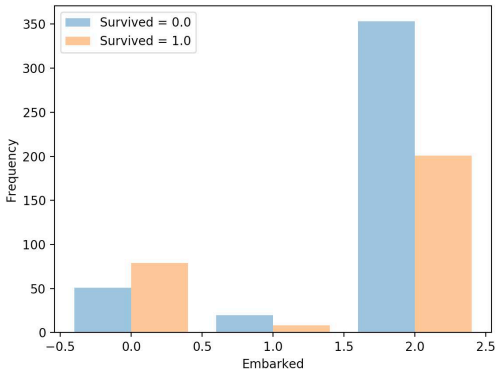
a) Histograms

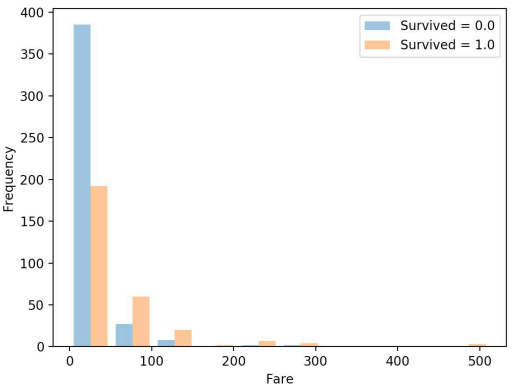


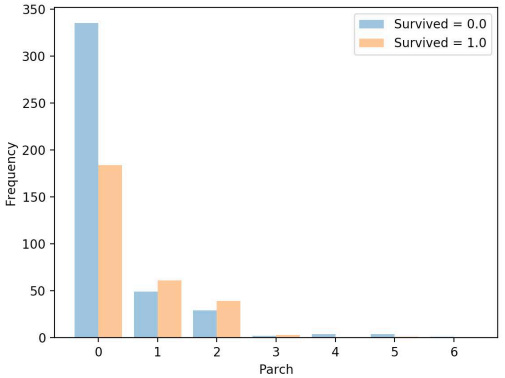








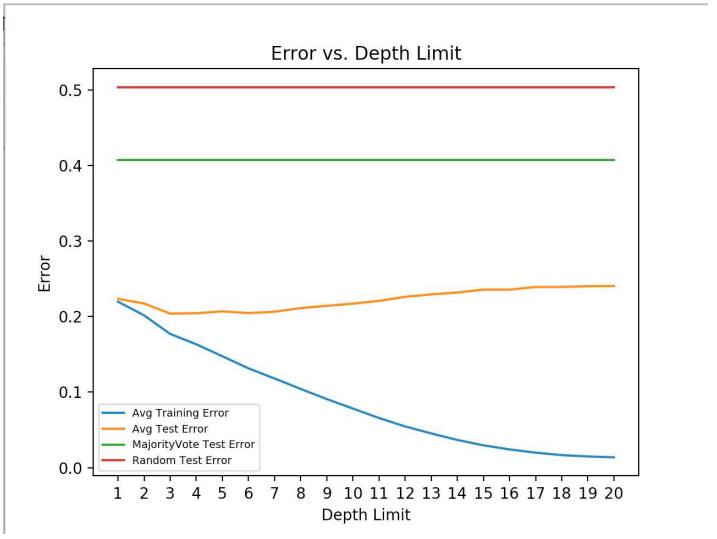




c) The training error of the DecisionTreeClassifier is 0.014.

d) For the MajorityVoteClassifier, the training error is 0.404 and the test error is also 0.404. For the RandomClassifier, the training error is 0.503 and the test error is also 0.503. For the DecisionTreeClassifier, the training error is 0.012 and the test error is 0.266.

e) Given the plot, I think that the best depth limit is 3 because this is when the test error is at its lowest and the training error is starting to decrease as well, but not to the point where it's caused by overfitting. I noticed that for the training error, there is overfitting as the depth limit increases because the training error gets closer and closer to 0.0, indicating that a depth tree with a greater depth limit will be built more specifically to satisfy the training data.



f) The plot shows that, for the DecisionTreeClassifier, as the training data split size increased, the training error gradually increases and the test error gradually decreases. The training error has a sharp increase around a split size of about 0.2 because the training data is essentially being tested against itself so as the split size increases, so does its error. The training error and test error also level out around an error of 0.2 indicating that the average error is around 0.2. Finally, the training data split sizes don't have much of an effect on the baseline classifiers.