stat671_HW1

Dany Jabban

2022-09-12

Theory Solutions

1 Concepts of Learning

1. I would use a ranking algorithm because we want to rank the horses and and then select the first, second, and third rankings. The features would be the horses characteristics and and past race results and we would predict rankings.
2. I would use a classification algorithm with 4 labels for the different galaxies, and assign one of these labels to each image.
3. I would use a regression algorithm because we want the output of our algorithm to predict a continuous variable.
4. I would use a regression algorithm because we want our algorithm to predict pounds of vegetables, a continuous variable.
5. I would use clustering. We are trying to uncover hidden groupings and correlations between products and a clustering algorithm could find similarities between different clothing products.
6. I would use density estimation because we are trying to determine characteristics of an unknown distribution.
7. I would use ranking because we are trying order a list of customers based on the likelihood of agreeing to a sale.
8. I would use conditional probability estimation. We have a default likelihood model and now we can condition that model on the new data to improve the model with the updated information for this specific situation.
9. I would use a regression algorithm because we are trying to determine the malignant tumor likelihood which is a continuous variable. In this case something like logistic regression would be useful since that provides values between 0 and 1.
10. I would use a clustering algorithm because this is an unsupervised learning task. We are trying to determine groupings of species solely based on the similarities of their genetic material.

2 Information Theory

2.1)

$$KL(P,Q) = \sum_{a\in O} P(a)log\frac{P(a)}{Q(a)}$$

$$= -\sum_{a\in O} P(a)log\frac{Q(a)}{P(a)}$$

Since $-log$ is a convex function, Jensens Inequality $E(f(x)) > f(E(x))$ implies:

$$\leq -log\sum_{a\in O} P(a)\frac{Q(a)}{P(a)}$$

$$= -log\sum_{a\in O} Q(a)$$

$$= -log(1) = 0$$

2.2)

$$KL(P,Q) = \sum_{a\in O} P(a)log\frac{P(a)}{Q(a)}$$

if $P = Q$ then:

$$KL(P,Q) = KL(P,P) = \sum_{a\in O} P(a)log\frac{P(a)}{P(a)}$$

$$= \sum_{a\in O} P(a)log(1)$$

$$= \sum_{a\in O} P(a)0 = 0$$

2.3)

let: $P = Bernoulli(.6)$ and $Q = Bernoulli(.5)$

$$KL(P,Q) = .6log\frac{.6}{.5} + .4log\frac{.4}{.5} = 0.0087447$$

$$KL(Q,P) = .5log\frac{.5}{.6} + .5log\frac{.5}{.6} = 0.0088643$$

Thus KL is not symmetric

2.4)

Need to show $KL(J,PQ) = H(X) - H(X|Y)$.

$$KL(J,PQ) = \sum_{a,b\in O} J(a,b)log\frac{J(a,b)}{P(a)Q(b)}$$

$$= \sum_{a,b\in O} J(a,b)(log\frac{J(a,b)}{Q(b)} - logP(a))$$

$$= \sum_{a,b\in O} J(a,b)log\frac{J(a,b)}{Q(b)} - \sum_{a,b\in O} J(a,b)logP(a)$$

$$= \sum_{a,b\in O} J(a,b)log\frac{J(a,b)}{Q(b)} - \sum_{a\in X}\sum_{b\in Y} J(a,b)logP(a)$$

$$= \sum_{a,b\in O} J(a,b)log\frac{J(a,b)}{Q(b)} - \sum_{a\in X} P(a)logP(a)$$

$$= -H(X|Y) + H(X) = H(X) + H(X|Y)$$

2.5)

Need to show $I(X,Y) := H(Y) - H(Y|X) = KL(J,PQ)$.

$$KL(J,PQ) = \sum_{a,b\in O} J(a,b)log\frac{J(a,b)}{P(a)Q(b)}$$

$$= \sum_{a,b\in O} J(a,b)(log\frac{J(a,b)}{P(a)} - logQ(b))$$

$$= \sum_{a,b\in O} J(a,b)log\frac{J(a,b)}{P(a)} - \sum_{a,b\in O} J(a,b)logQ(b)$$

$$= \sum_{a,b\in O} J(a,b)log\frac{J(a,b)}{P(a)} - \sum_{b\in Y}\sum_{a\in X} J(a,b)logQ(b)$$

$$= \sum_{a,b\in O} J(a,b)log\frac{J(a,b)}{P(a)} - \sum_{b\in Y} Q(b)logQ(b)$$

$$= -H(Y|X) + H(Y) = H(Y) + H(Y|X) = I(X,Y)$$

2.6)

Since P is a proportion it takes on values on the interval $[0,1]$ but since we are only looking at $p < 0.5$, our interval for p is $[0,.5)$. Now since misclassification error is defined: $error = min(p, 1-p)$ and p is on the interval $[0,.5)$, then $error = min(p, 1-p) = p$. Entropy is defined: $H(p) = -plogp - (1-p)log(1-p)$. This means we need to show that as p decreases on the interval $[0,.5)$, $H(p)$ decreases as well. $\frac{dH}{dp} = log(1-p) - log(p)$ and since $-log(x) > 0$ and $-log(x) > |log(1-x)|$ on the interval $[0,.5)$ then $\frac{dH}{dp}$ is strictly positive on the interval $[0,.5)$ which implies $H(x)$ is a strictly monotonically increasing function on the interval $[0,.5)$. This means that as p decreases on the interval $[0,.5)$, $H(P)$ decreases as well.

3.1)

Look at Code Solutions for 3.1) for additional details

There were 3 threshold ranges that minimized misclassification error at 20%. They were (1.994, 0.564), (0.406, 0.278), (-0.316, -1.54). We can minimize misclassification error of g by using any threshold within these ranges so lets pick threshold = 0.3

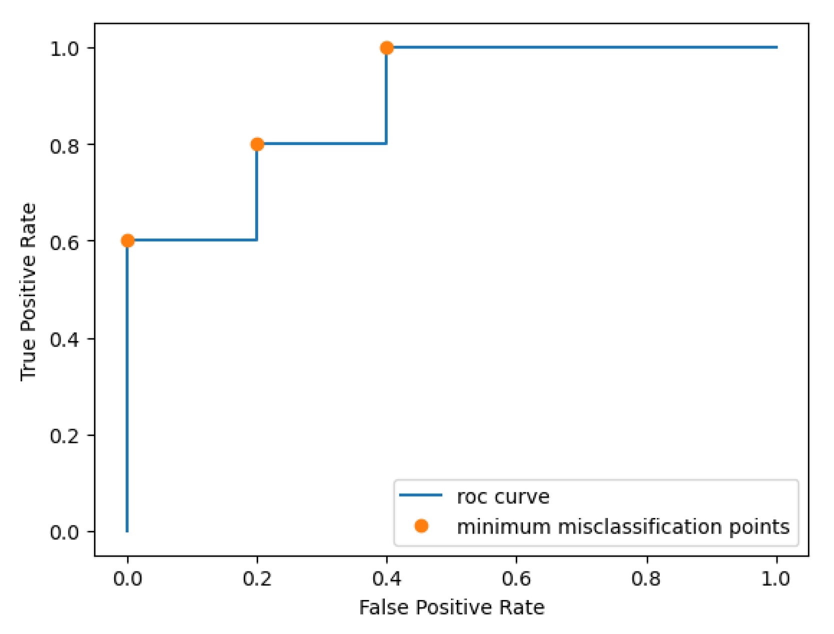
3.2)

Look at Code Solutions for 3.2) for additional details

There were 3 threshold ranges that minimized misclassification error at 20%. They were (0.96360121, 0.51093923), (0.38507106, 0.27105303), (-0.30588564, -0.91212037). We can minimize misclassification error of f by using any threshold within these ranges so lets pick threshold = 0.7. Choosing this threshold would give us a precision of 1.0, recall of 0.6 and an F1 score of 0.7499, and Confusion Matrix:

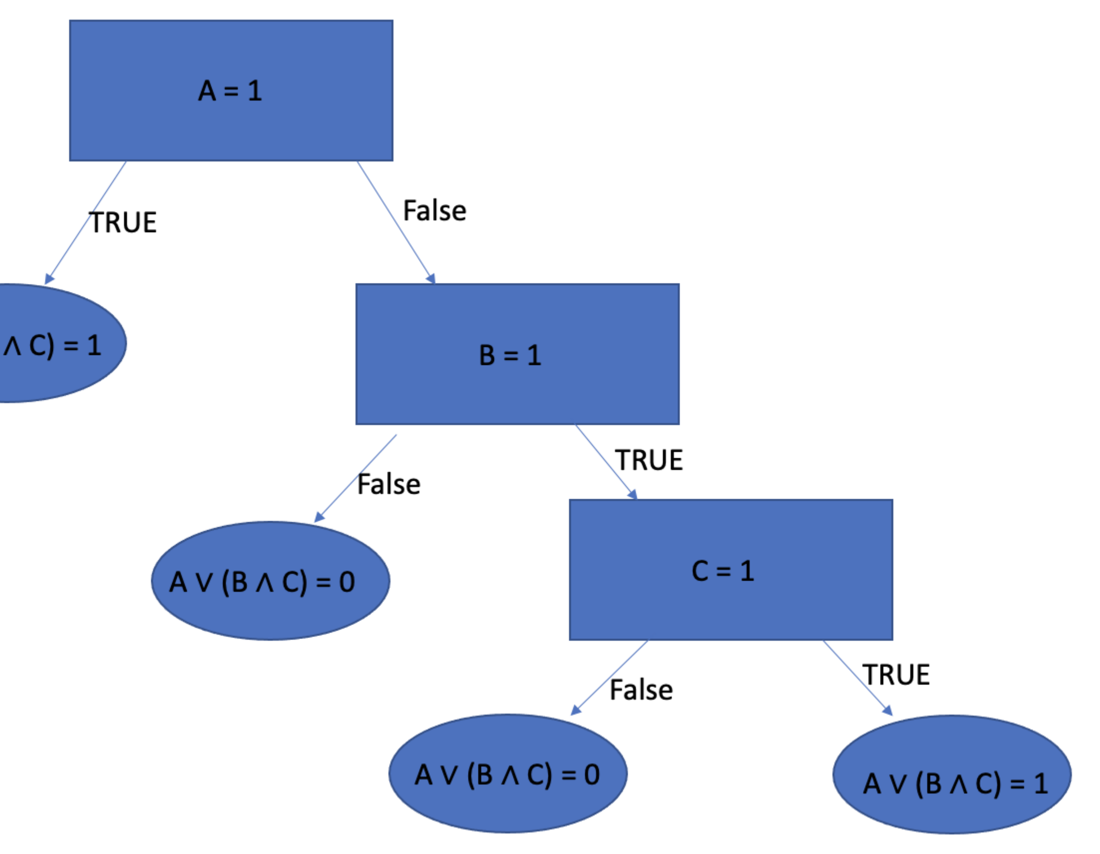
	Predicted_Positive	Predicted_Negative
Acutal_Positive	5	2
Actual_Negative	0	3

3.3)



ROC curve

4.1)



Tree split with smallest number of nodes

4.2)

Splitting on Gini index:

Gini Index when splitting on A first:

$p(A=1) = \frac{4}{8} = \frac{1}{2}$ $p(A=0) = \frac{4}{8} = \frac{1}{2}$

$p(A=1, A\vee(B\wedge C)=1) = \frac{4}{4} = 1$ $p(A=1, A\vee(B\wedge C)=0) = \frac{0}{4} = 0$

when $A = 1$ gini index $= 2p(p-1) = 2\cdot 1\cdot 0 = 0$

$p(A=0, A\vee(B\wedge C)=1) = \frac{1}{4}$ $p(A=0, A\vee(B\wedge C)=1) = \frac{3}{4}$

when $A = 0$ gini index $= 2p(p-1) = 2\frac{1}{4}\cdot\frac{3}{4} = \frac{3}{8}$

so gini index $= p(A=1)Gini(A=1) + p(A=0)Gini(A=0) = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot\frac{3}{8} = \frac{3}{16}$

Gini Index when splitting on B first:

$p(B=1) = \frac{4}{8} = \frac{1}{2}$ $p(B=0) = \frac{4}{8} = \frac{1}{2}$

$p(B=1, A\vee(B\wedge C)=1) = \frac{3}{4}$ $p(B=1, A\vee(B\wedge C)=0) = \frac{1}{4}$

when $B = 1$ gini index $= 2p(p-1) = 2\cdot\frac{3}{4}\cdot\frac{1}{4} = \frac{3}{8}$

$p(B=0, A\vee(B\wedge C)=1) = \frac{2}{4} = \frac{1}{2}$ $p(B=0, A\vee(B\wedge C)=1) = \frac{2}{4} = \frac{1}{2}$

when $B = 0$ gini index $= 2p(p-1) = 2\frac{1}{2}\cdot\frac{1}{2} = \frac{1}{2}$

so gini index $= p(V=1)Gini(V=1) + p(V=0)Gini(V=0) = \frac{1}{2}\cdot\frac{3}{8} + \frac{1}{2}\cdot\frac{1}{2} = \frac{7}{16}$

Since B and C's only involvement in the logical expression is in $(B\wedge C)$ and since $\wedge$ is commutative, B and C will have the same gini index. Further, A has a gini index value $\frac{3}{16}$ and B and C have gini index values of $\frac{7}{16}$ thus we will split on A first.

4.3)

Splitting on Information Gain:

Information Gain when splitting on A first:

$Gain(A) = H(\frac{5}{8},\frac{3}{8}) - [\frac{4}{8}\cdot H(1,0)] + \frac{4}{8}\cdot H(\frac{1}{4},\frac{3}{4})]$

$= [-\frac{5}{8}log_2\frac{5}{8} - \frac{3}{8}log_2\frac{3}{8}] - [\frac{4}{8}\cdot 0 + \frac{4}{8}(-\frac{1}{4}log_2\frac{1}{4} - \frac{3}{4}log_2\frac{3}{4})] = 0.548795$

$Gain(B) = H(\frac{5}{8},\frac{3}{8}) - [\frac{4}{8}\cdot H(\frac{3}{4},\frac{1}{4})] + \frac{4}{8}\cdot H(\frac{1}{2},\frac{1}{2})]$

$= [-\frac{5}{8}log_2\frac{5}{8} - \frac{3}{8}log_2\frac{3}{8}] - [\frac{4}{8}(-\frac{1}{4}log_2\frac{1}{4} - \frac{3}{4}log_2\frac{3}{4})] + \frac{4}{8}(-\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2})] = 0.048795$

Since B and C's only involvement in the logical expression is in $(B\wedge C)$ and since $\wedge$ is commutative, B and C will have the same information gain. Further, A has an information gain value 0.548795 and B and C have information gain values of 0.048795 thus we will split on A first.

5.1)

Not a question

5.2)

Look at Code Solutions for 5.2) for more details

After running the k fold cross validation for hyperparameter tuning of our CART algorithm, the results were: For criterion = gini, max_depth = 2, average F1 score was 0.458497 For criterion = gini, max_depth = 3, average F1 score was 0.498864 For criterion = entropy, max_depth = 2, average F1 score was 0.425164 For criterion = entropy, max_depth = 3, average F1 score was 0.511288

From these results we will choose hyperparameter combination criterion = entropy, max_depth = 3 since it had the highest F1 score

After running the k fold cross validation for hyperparameter tuning of our GOSDT algorithm, the results were: For regularization = 0.05, depth_budget = 2, average F1 score was 0.390638 For regularization = 0.05, depth_budget = 3, average F1 score was 0.390638 For regularization = 0.001, depth_budget = 2, average F1 score was 0.390638 For regularization = 0.001, depth_budget = 3, average F1 score was 0.556850

From these results we will choose hyperparameter combination regularization = 0.001, depth_budget = 3 since it had the highest F1 score

5.3)

Look at Code Solutions for 5.3) for all details

After running the k fold cross validation for hyperparameter tuning of our KNN algorithm, the results were: For Metric Distance = euclidean, k = 1, average F1 score was 0.420649 For Metric Distance = euclidean, k = 3, average F1 score was 0.318225 For Metric Distance = manhattan, k = 1, average F1 score was 0.464892 For Metric Distance = manhattan, k = 3, average F1 score was 0.359026

From these results we will choose hyperparameter combination Metric Distance = manhattan, k = 1 since it had the highest F1 score

5.4)

Look at Code Solutions for 5.4) for more details

After picking the best hyperparameters for each of the three algorithms, we configured the three models with the hyperparameters than then trained each of the models with the full carseat training data. After training the model we used the .predict method using the carseat test data. These were the following F1 scores for each of the models:

sktree_model f1 score: 0.7288135593220338 gosdt_model f1 score: 0.4576271186440678 knn_model f1 score: 0.5254237288135594

sklearn's CART model with hyperparameters Criterion = entropy and max_depth = 3 performed the best on the training set with an F1 score of 0.73.

5.5)

K-fold cross validation is a process to evaluate a model where you break up the data into k groups, of equal number of observations, iterating from 1 to k where for each iteration you use a different group as a test set and use the remaining data as the training set. after k iterations you average the scores on the test set to get model performance. LOOCV is a special case of k-fold CV where k = n, with n being the number of observations. One limitation of LOOCV is that for extremely large data sets it can take a long time because iterating over large n can be computationally expensive. Another drawback arises because we have a test set of 1 observation so there will be a high variance in our prediction scores.

5.6)

Accuracy score can be an issue when we have imbalanced data. If we have very few negatives relative to positives the significance of an accuracy score declines so in these cases F1 score is a better metric for model performance since target variable proportions are accounted for. In the case of the carseat data the training data is imbalanced while the test data is evenly balanced. This does raise questions regarding weather these observations came from the same distribution. Regardless since the overall data is imbalanced it would be useful to use F1 as our metric.

In the case of the carseat_test data the positives and negatives are evenly split so accuracy might be a more useful metric because it is more interpretable than F1 score.

6.1) Since each of the points x_i are i.i.d from the uniform distribution over the d-dimensional hypercube :

$$p(\rho(NN(x_{test}), x_{test}) > \epsilon) = p(\rho(x_1, x_{test}) > \epsilon, \ldots, \rho(x_n, x_{test}) > \epsilon) = \prod_{i=1}^{n} p(\rho(x_i, x_{test}) > \epsilon)$$

$$p(\rho(x_i, x_{test}) = 0) = \frac{1}{2^d}$$

therefore for some $\epsilon > 0$

$$p(\rho(x_i, x_{test}) \leq \epsilon) \geq \frac{1}{2^d}$$

$$1 - p(\rho(x_i, x_{test}) \leq \epsilon) \leq 1 - \frac{1}{2^d} < 1$$

$$p(\rho(x_i, x_{test}) > \epsilon) \leq 1 - \frac{1}{2^d} < 1$$

Thus

$$\lim_{x\to\infty} p(\rho(NN(x_{test}), x_{test}) > \epsilon) = \lim_{x\to\infty}\prod_{i=1}^{n} p(\rho(x_i, x_{test}) > \epsilon) = \lim_{x\to\infty}(p(\rho(x_i, x_{test}) > \epsilon))^n \leq \lim_{x\to\infty}(1 - \frac{1}{2^d})^n = 0$$

6.2) We are given that f satisfies the Lipschitz condition which implies that f is continious in the space. In 6.1 we showed that as n goes to infinity $NN(x_{test})$ converges in probability to x_{test}. Then by the continuous mapping theorem $f(NN(x_{test}))$ converges in probability to $f(x_{test})$ denoted: $\lim_{x\to\infty} p(d_y(f(NN(x_{test})), f(x_{test})) > \epsilon) = 0$. Therefore:

$$\lim_{x\to\infty} p(d_y(|f(NN(x_{test})) - f(x_{test})|, 0) > \epsilon) = \lim_{x\to\infty} p(|f(NN(x_{test})) - f(x_{test})| - 0 > \epsilon)$$

$$= \lim_{x\to\infty} p(|f(NN(x_{test})) - f(x_{test})| > \epsilon)$$

$$\lim_{x\to\infty} p(d_y(f(NN(x_{test})), f(x_{test})) > \epsilon) = 0$$

6.3) x_i is selected from a uniform distribution of a d-dimensional hypercube. probability of getting a point inside the hamming distance $r = 2$ is $\frac{\binom{d}{2}+\binom{d}{1}+1}{2^d}$ therefore the expected value of the number of points to lie within the distance $r = 2$ is $n\cdot(\frac{\frac{d^2-d}{2}+d+1}{2^d}) = n\cdot(\frac{\frac{d^2}{2}+\frac{d}{2}+1}{2^d})$

6.4) Since the numerator grows at a polynomial rate and the denominator grows at an exponential rate, the denominator will blow up much faster than the numerator thus as $d\to\infty$, K' goes to 0

6.5) From 6.4) we see that as the number of dimensions d increases the number of points within a given distance gets closer and closer to 0 for any arbitrary distance r. As the number of dimensions d increases, the number of observations needs to increase exponentially for the expected number of points within a given radius r to not be 0. As d gets relatively large the number of necessary observations quickly becomes impractical therefore KNN does not work well in high dimensions