1)

a) The scheduled meeting for our group project is at 4:30 pm on Tuesday, the meeting agenda includes going over data we found about the income gpt around different counties in Indiana, as well as planning the next steps in out project on how we will represent the data.

b) Naive Bayes Classifier is more of a probability classifier that applies Bayes theorem with an independent assumption between features while the Nearest Neighbor Classifier is more of an instance-based learning algorithm. Naive Bayes Classifier also builds a model based on training data and computes conditional probabilities from the training data and uses it for classification, while the Nearest Neighbor Classifier does not build any models but instead stores the training dataset, finally Naive Bayes Classifier assumes features are independent given the class while Nearest Neighbor Classifier makes no assumption about the feature being independent and relies on distance. Another difference between the two are that Nayes Bayes Classifier is faster than Nearest Neighbor Classifier when applied to big data, sources (https://www.datasciencecentral.com/comparing-classifiers-decision-trees-knn-naive-bayes/).

c) To train a Naive Bayes classifier on the iris dataset, we should first load the dataset into training and testing sets, then divide each data into features and make sure each feature contributes equally to the model, then calculate the conditional probabilities and the prior probabilities then classify the data points, then you should train the naive Bayes classifier and finally evaluate the performance of the model on the testing set.

d) For distribution (1) (⅓, ⅕, ⅕):

$$H(p) = [(⅓)\log_2(1/(1/3)) + (⅕)\log_2(1/(⅕)) + (⅕)\log_2(1/(1/5))]$$
$$[0.5283, 0.4643, 0.4643]$$
$$= 1.4571$$

For distribution (2) (½, ½, ½)

$$H(p) = [(½)\log_2(1/(½)) + (½)\log_2(1/(½)) + (½)\log_2(1/(½))]$$
$$[0.5, 0.5, 0.5]$$
$$= 1.5$$

2)

a) Mileage:
Min: 12458
Max: 37394
Standardized Mileage:
12458 -> (12458 - 12458)/(37394 - 12458) = 0
37394 -> (37394 - 12458)/(37394 - 12458) = 1
23570 -> (23570 - 12458)/(37394 - 12458) ≈ 0.445
20000 -> (20000 - 12458)/(37394 - 12458) ≈ 0.302

Doors:
Min: 2
Max: 4
Standardized doors:
4 -> (4 - 2)/(4 - 2) = 1
2 -> (2 - 2)/(4 - 2) = 0
4 -> (4 - 2)/(4 - 2) = 1
3 -> (3 - 2)/(4 - 2) = 0.5

Age: Min: 6
Max: 13
Standardized Age:
13 -> (13 - 6)/(13 - 6) = 1
7 -> (7 - 6)/(13 - 6) = 0.143
8 -> (8 - 6)/(13 - 6) = 0.286
6 -> (6 - 6)/(13 - 6) = 0

d(0,1,1) = √(0.302-0)^2 + (0.5-1)^2 + (0-1)^2 ≈ 1.158
d(1,0,0.143) = √(0.302-1)^2 + (0.5-0)^2 + (0-0.143)^2 ≈ 0.870
d(0.445, 1, 0.286) = √(0.302-0.445)^2 + (0.5-1)^2 + (0-0.286)^2 ≈ 0.594
The smallest distance is the third instance so the estimated class using Euclidean distance is Y

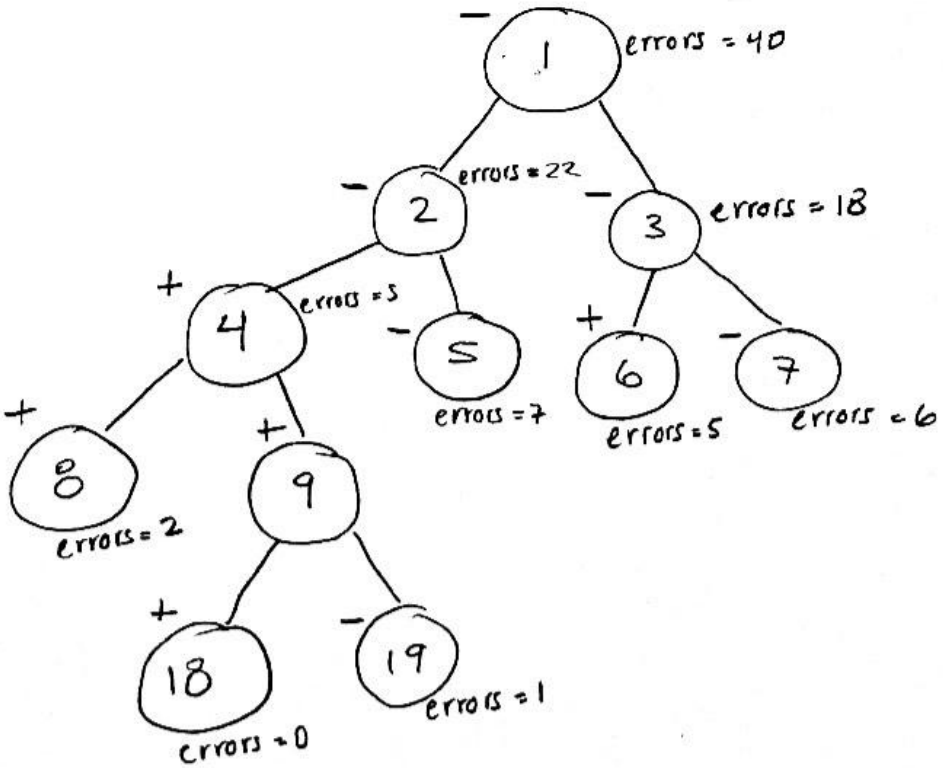
b) d(0, 1, 1): $|0.302-0| + |0.5-1| + |0-1| = 1.802$
d(1, 0, 0.143): $|0.302-1| + |0.5-0| + |0-0.143| = 1.341$
d(0.445, 1, 0.286): $|0.302-0.445| + |0.5-1| + |0-0.286| = 0.929$

The smallest distance is the third instance and so the estimated class using Manhattan distance is Y

c) Standardizing data helps features have equal importance in distance computations and helps make sure their are more accurate classifications.

3) c) My error rate is not that much of an accurate estimation of the error rate when testing on the previously-unseen data because computing an error rate by training and testing on the same dataset can provide optimistic estimates

4) a)



---

b) Node 1 R(T) = 40/100
Node 6 R(T) = 5/100
Node 19 R(T) = 1/100

c) this is not an accurate representation of the trees performance on new data because it is based on training set only and the tree might not perform well on a new data set.