CSEN240 Machine Learning
Homework #4

(Level 1: high-level understanding)

1. (12 points) Which of the following are true? (Check all apply)
    a. KNN assumes that points close to each other in the space are in the same class.
    b. Feature scaling is important in KNN because the distance metric used may be sensitive to the scale of the features. Therefore, it is common to normalize or standardize the features before applying KNN.
    c. The K-value in KNN specifies the number of nearest neighbors to consider when making a prediction. This value is typically chosen based on cross-validation or other optimization techniques.
    d. KNN uses a majority voting scheme to classify new data points. That is, it assigns the class label that is most common among the k-nearest neighbors.
    e. It may not perform as well as other algorithms in high-dimensional spaces due to the "curse of dimensionality"
    f. KNN is not robust to irrelevant features.
2. (10 points) Which of the following are true? (Check all apply)
    a. Bayesian classifiers are probabilistic models that estimate the probability of each class label given the observed features using Bayesian theorem.
    b. Bayesian classifiers assume that the features are conditionally independent given the class label.
    c. While this assumption may not always hold in practice, Bayesian classifiers are still effective in many classification tasks.
    d. Training Bayesian Classifier for continuous attributes involves computing the Probability Density Function (PDF)
    e. Bayesian classifiers can produce probability features for each class label, which can be useful for decision-making and uncertainty analysis.

(Level 2: manual exercise)

3. (18 points) Given the following dataset of labeled points in two dimensions
    a. Positive samples: {(3,3), (4,4)}
    b. Negative samples: {(1,2), (2,1)}
    c. Please use the nearest neighbor classifier to predict the class of a new data point at (2,2.4)
4. (20 points) Given the examples below

Ex.	**Crust size**	**Shape**	**Filling size**	**Class**
e1	big	circle	small	**pos**
e2	small	circle	small	**pos**
e3	big	sq.	big	**pos**
e4	small	sq.	big	**pos**
e5	big	circle	big	**pos**
e6	big	sq.	small	**neg**
e7	big	sq.	small	**neg**

- Use Bayesian Classifier to predict
- $\mathbf{x}^T$= [Crust-size=small, shape=square, Filling-size=small]

(Level 3: computer-based exercise)

5. (20 points) KNN implementation (Using HW4-5.ipynb)
    - Implement a Nearest Neighbor Classifier (or, KNN algorithm when K=1) using Euclidean distance
    - First, assuming that the dataset is in 2-dimensional space.
    - Assuming $\sum_{i=1}^{2} x_i > 0$ is the ground truth of the decision boundary.
        - Create 160 unique samples (80 are positive and 80 are negative). Evenly split the 160 samples into 80 training samples and 80 testing samples. Using 100% of the training samples, and test the precision, recall, and F1 score of the unseen testing samples, using KNN. (Repeat 30 times for the average)
    - Second, assuming that the dataset is in 10-dimensional space.
        - Assuming $\sum_{i=1}^{10} x_i > 0$ is the ground truth of the decision boundary.
        - Create 800 unique samples (400 are positive and 400 are negative). Evenly split the 800 samples into 400 training samples and 400 testing samples. Using 100% of the training samples, and test the precision, recall, and F1 score of the unseen testing samples, using KNN. (Repeat 30 times for the average)
    - What do you observe? Can you explain?
6. (20 points) Bayesian Classifier implementation (Using HW4-6.ipynb)
    - Implement a Bayesian Classifier, using one Gaussian distribution to approximate the PDF for each class.
    - First, assuming that the dataset is in 2-dimensional space.
    - Assuming $\sum_{i=1}^{2} x_i > 0$ is the ground truth of the decision boundary.
        - Create 160 unique samples (80 are positive and 80 are negative). Evenly split the 80 samples into 80 training samples and 80 testing samples. Using 100% of the training samples, and test the precision, recall, and F1 score of the unseen testing samples, using Bayesian Classifier. (Repeat 30 times for the average)
    - Second, assuming that the dataset is in 10-dimensional space.
        - Assuming $\sum_{i=1}^{10} x_i > 0$ is the ground truth of the decision boundary.
        - Create 800 unique samples (400 are positive and 400 are negative). Evenly split the 800 samples into 400 training samples and 400 testing samples. Using 100% of the training samples, and test the precision, recall, and F1 score of the unseen testing samples, using Bayesian Classifier. (Repeat 30 times for the average)
    - Please compare the performance drop from 2-dimensional to 10-dimensional for Bayesian Classifier vs. that for KNN. What do you observe? Can you explain?