COMP6200 - Mock Final Exam

Question 1
Please briefly describe the learning process of an Artificial Neural Network, including the feedforward neural network, back propagation algorithm and the iterative learning of edge weights through gradient descent. (repetitively)

Question 2
Please briefly describe the iterative learning process of the K-means algorithm.
Discuss the overfitting problem in K-means, and how we can avoid overfitting of K-Means.

The iterative learning process of the K-means algorithm can be described as follows:

1. Initialization: The algorithm starts by randomly initializing K cluster centroids (means) in the data space.
2. Assignment step: Each data point is assigned to the nearest cluster centroid based on a distance metric (usually Euclidean distance).
3. Update step: After all data points have been assigned to clusters, the centroids are updated by calculating the mean of all data points belonging to each cluster.
4. Iteration: Steps 2 and 3 are repeated iteratively until convergence, which occurs when the assignments of data points to clusters no longer change or when a maximum number of iterations is reached.

The algorithm aims to minimize the sum of squared distances between data points and their assigned cluster centroids, effectively partitioning the data into K clusters.

Overfitting problem in K-means: Overfitting in K-means can occur when the number of clusters (K) is set too high relative to the dataset's inherent structure. In such cases, the algorithm may divide the data into an excessive number of clusters, leading to overfitting. This means that the clusters may capture noise or random fluctuations in the data rather than representing meaningful patterns.

Avoiding overfitting in K-means: There are several approaches to avoid overfitting in K-means:

1. Choosing an appropriate value of K: One of the most important steps is to select an appropriate value for K, the number of clusters. This can be done using various techniques such as the elbow method, silhouette analysis, or domain knowledge about the expected number of clusters.
2. Cross-validation: Cross-validation techniques can be used to evaluate the performance of different values of K on a held-out validation set. The value of K that generalizes well to the validation set can be chosen to avoid overfitting.

3. Regularization: Regularization techniques, such as adding a penalty term to the objective function, can help prevent overfitting by discouraging the formation of very small or tightly clustered groups.
4. Dimensionality reduction: Reducing the dimensionality of the data before applying K-means can help mitigate overfitting by removing noise or irrelevant features.
5. Ensemble methods: Using ensemble methods, such as running K-means multiple times with different initializations and combining the results, can help reduce the impact of random initialization and provide more robust clustering results.
6. Domain knowledge: Incorporating domain knowledge about the expected cluster structures or feature importances can guide the algorithm and help avoid overfitting.

It's important to note that while overfitting can be a concern in K-means, it is generally less problematic than in other machine learning models like neural networks or decision trees. The K-means algorithm is relatively simple and does not have as many parameters to overfit as more complex models.

Question 3

In the training dataset, there are 14 instances. Each instance has 4 attributes. The label is "Yes" or "No" indicating whether the student will play Tennis or not. Based on the provided training dataset, your task is to establish the Naive Bayes model to predict the label of an instance with attributes x=(Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong).

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

To establish the Naive Bayes model from the given training dataset and predict the label for the instance x = (Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong), we need to follow these steps:

1. Calculate the prior probabilities for each class label (PlayTennis = Yes or No) from the training data.
2. Calculate the likelihood probabilities for each attribute value given each class label, using the frequency counts from the training data and applying Laplace smoothing (adding 1 to each count) to avoid zero probabilities.
3. Apply Bayes' theorem to calculate the posterior probability of each class label given the instance x, using the prior probabilities and likelihood probabilities calculated in steps 1 and 2.
4. Assign the class label with the highest posterior probability as the prediction for the instance x.

Let's go through these steps:

Prior probabilities:

Number of instances with PlayTennis = Yes: 9
Number of instances with PlayTennis = No: 5
Total number of instances: 14

P(PlayTennis = Yes) = 9 / 14 ≈ 0.643
P(PlayTennis = No) = 5 / 14 ≈ 0.357

Likelihood probabilities:

In the training dataset, there are 14 instances. Each instance has 4 attributes. The label is "Yes" or "No" indicating the student will play Tennis or not. Based on provided training dataset, your task is to establish the Naive Bayes model to predict the label of an instance with attributes x= (Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong).

1. Calculate Prior Probabilities: P(yes) = 9/14 & p(no) = 5/14
2. Calculate Likelihoods:

Outlook	Play=Yes	Play=No
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=Yes	Play=No
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=Yes	Play=No
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=Yes	Play=No
Strong	3/9	3/5
Weak	6/9	2/5

3. Calculate Posterior Probabiliti

- For "Yes":

$$P(\text{Yes}|x) \propto \frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14} = \frac{2 \cdot 3 \cdot 3 \cdot 3 \cdot 9}{9^4 \cdot 14}$$

- For "No":

$$P(\text{No}|x) \propto \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14} = \frac{3 \cdot 1 \cdot 4 \cdot 3 \cdot 5}{5^4 \cdot 14}$$

Posterior probabilities for the instance x = (Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong):

P(Yes | x) = P(PlayTennis = Yes) * P(Outlook=Sunny | Yes) * P(Temperature=Cool | Yes) * P(Humidity=High | Yes) * P(Wind=Strong | Yes) ≈ 0.009

P(No | x) = P(PlayTennis = No) * P(Outlook=Sunny | No) * P(Temperature=Cool | No) * P(Humidity=High | No) * P(Wind=Strong | No) ≈ 0.007

Prediction:

Since P(Yes | x) > P(No | x), the Naive Bayes model predicts that for instance x = (Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong), the label is PlayTennis = Yes.

Therefore, according to the Naive Bayes model trained on the given dataset, the predicted label for the instance x = (Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong) is PlayTennis = Yes.

Question 4

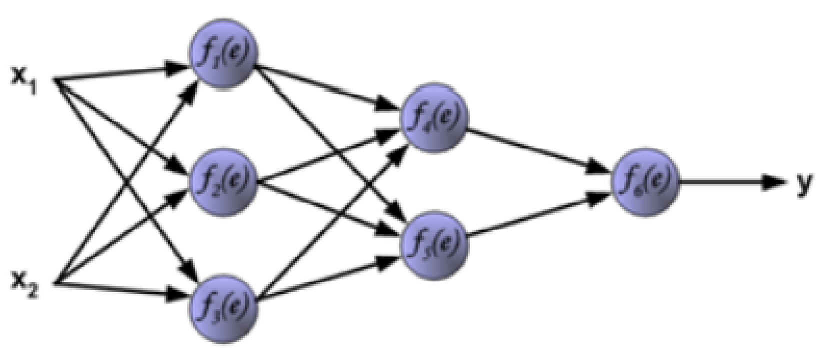
Here are the steps of the K-Means clustering algorithm, put them in order

Step 1:	Select K random points as initial centroids
Step 2:	Loop over all of the data points
Step 3:	Calculate the distance from the point to each cluster centroid
Step 4:	Allocate the point to the closest cluster centroid
Step 5:	Once all points have been allocated, compute the new cluster centroid
Step 6:	if the centroid positions have changed significantly, go back to step 2

1. Select K random points as initial centroids (cluster centers).
2. Loop over all of the data points.
3. Calculate the distance from the point to each cluster centroid.
4. Allocate the point to the closest cluster centroid.
5. Once all points have been allocated, compute the new cluster centroid (mean of all points in the cluster).
6. If the centroid positions have changed significantly, go back to step 2 and repeat the process. Otherwise, stop the iteration.

The algorithm continues iterating between steps 2-6 until the centroid positions converge and no longer change significantly. This iterative process aims to minimize the sum of squared distances between data points and their assigned cluster centroids, ultimately finding an optimal clustering of the data based on the specified number of clusters K.

Question 5
For an Artificial Neural Network, please briefly describe its training process, including the feedforward neural network, back propagation algorithm and iterative learning of edge weights by gradient descent.



1. **Feedforward Neural Network.**

- The neural network consists of an input layer, one or more hidden layers, and an output layer.
- During the forward propagation phase, the input data is fed into the input layer, and the activations are propagated through the network by applying weights and activation functions at each layer.
- The output from the final layer represents the network's prediction or output.

2. **Back-Propagation Algorithm:**
   - After the forward propagation, the network's output is compared to the expected or target output using a loss function (e.g., mean squared error, cross-entropy).
   - The back-propagation algorithm calculates **the gradients of the loss function** with respect to the **weights and biases** in the network, starting from the output layer and propagating the errors backward through the network.
   - The gradients provide information about how the weights and biases should be adjusted to **minimize the loss function**.

3. **Iterative Learning of Edge Weights by Gradient Descent:**
   - Gradient descent is an optimization algorithm used to update the weights and biases in the neural network based on the calculated gradients.
   - The weights and biases are updated by subtracting a fraction (determined by the learning rate) of the gradients from their current values.
   - This process is repeated iteratively over multiple epochs (complete passes through the entire training dataset).
   - During each iteration, the network's predictions are evaluated, the gradients are calculated using back-propagation, and the weights and biases are updated in the direction that minimizes the loss function.

SUMMARIZED VER.
The training process typically involves the following steps:

1. Initialize the weights and biases of the neural network with small random values.
2. Forward propagate the input data through the network to obtain the output predictions.
3. Calculate the loss between the predicted output and the expected output using a loss function.
4. Back-propagate the errors from the output layer to the input layer, computing the gradients of the loss function with respect to the weights and biases.
5. Update the weights and biases using gradient descent, adjusting them in the direction that minimizes the loss function.

6. Repeat steps 2-5 for multiple epochs, iterating over the entire training dataset until the loss converges or a desired performance is achieved.

During training, techniques like regularization (e.g., L1, L2), dropout, and early stopping may be employed to prevent overfitting and improve generalization. Additionally, techniques like batch normalization and adaptive learning rate optimization algorithms can be used to improve the training process.

The iterative nature of the training process, combined with the back-propagation algorithm and gradient descent optimization, allows the neural network to learn the appropriate weights and biases that map the input data to the desired output, capturing the underlying patterns and relationships in the data.

Question 6

Clustering is an example of:

Select one:

- ○ a. **Unsupervised learning** because we can just let the algorithm run without having to check it.
- ○ b. **Supervised learning** because we have to supervise the steps of the algorithm for it to work correctly
- ◉ c. **Unsupervised learning** because we don't feed the answer (clusters) into the algorithm.
- ○ d. **Supervised learning** because the algorithm needs training data to find the clusters

Clear my choice