UNIT 3 ASSIGNMENT

Understanding the Mechanics of   
ML Algorithms

## Instructions

The questions below will prepare you for future interviews as they relate to concepts discussed throughout the unit. You’ve practiced these concepts in the coding activities, exercises, and coding portion of the assignment; let’s now formulate your programming into well-reasoned responses.

Except as indicated, use this document to record all your assignment work and responses to any questions. At a minimum, you will need to turn in a digital copy of this document to your facilitator   
as part of your assignment completion. You may also have additional supporting documents that   
you will need to submit. Your facilitator will provide feedback to help you work through your findings.

**Note:** Though your work will only be seen by those grading the course and will not be used or   
shared outside the course, you should take care to obscure any information you feel might be   
of a sensitive or confidential nature.

*Begin your assignment by completing the questions below. Directions to submit your work can be found on the assignment page. Information about the grading rubric is available on any of the course assignment pages online. Do not hesitate to contact your facilitator if you have any questions about the assignment.*

Unit 3 Written Portion

# Building and Evaluating a Model

Answer the questions below about building and evaluating your models using algorithms such as decision trees and k-nearest neighbors.

## Questions:

1. What are the advantages and disadvantages of decision trees?

| Advantages of Decision Trees:  1. Visualization: visually understandable structure, enabling the explanation of prediction rationale.  2. Simplified Data Preparation: require less effort in data preprocessing before training.  3. Feature Importance: provides a metric for gauging the significance of features, aiding in feature selection and comprehension.  4. Efficiency: quicker training times and can process a certain degree of scalability.  5. Outlier Robustness: as they make decisions based on feature space splits, they limit the impact of outliers on the overall tree structure.  Disadvantages  1. Overfitting: prone to overfitting, especially when the tree becomes too complex or the data has noisy or irrelevant features. Overfitting occurs when the tree captures the training data's specific details too well, resulting in poor generalization to unseen data.  2. Sensitive: Decision trees are sensitive to small changes in the data, which can lead to different tree structures. |
| --- |

1. What are the advantages and disadvantages of k-nearest neighbors?

| 1. Easy Implementation**-** KNN is very easy to implement as the only thing to be calculated is the distance between different points on the basis of data of different features and this distance can easily be calculated using distance formula such as- Euclidian or Manhattan  2. New data can be added at anytime as long as it doesn’t overfit the data  Disadvantages:  1. Does not work well with large dataset as calculating distances between each data instance would be very costly.  2. Does not work well with high dimensionality as this will complicate the distance calculating process to calculate distance for each dimension.  3. Sensitive to noisy and missing data  4. Feature Scaling- Data in all the dimension should be scaled (normalized and standardized) properly |
| --- |

1. Explain the difference between k-nearest neighbors and decision trees. When would you decide to use one over the other?

| KNN makes predictions based on the similarity of new unseen data based off the training set. It determines the class of a new data point by considering the class labels of its K nearest neighbors. KNN training only involves storing the training data. It also computes the distance between the neighbors which could be less efficent.  Decision Trees builds a tree-like model of decisions and their possible consequences. It learns a series of if-else conditions based on the features to classify the data. Theyrepresent a series of logical conditions that lead to a particular outcome. Trees have an efficient training process as they recursively split the data based on the features.  They can automatically select the most informative features for classification or regression tasks.  Some instances when you should use KNN over D-Trees:  KNN: when the data has a well-defined distance metric and local patterns are important. D-Tree: both numerical and categorical data, easier interpretable vizualization.  D-Tree: Faster and cheaper |
| --- |

1. What are hyperparameters? List some hyperparameters in k-nearest neighbors and decision trees.

| Hyperparameters are algorithms’ specific inputs that control how the model is built. They’re set prior to training the model. It declares the mechanics of the model and determines how the model is trained. Some examples are: Size of neighborhood in KNN, Depth of tree in decision tree, Learning rate, or step size, in gradient descent |
| --- |

1. What is overfitting? How can you avoid overfitting? Give examples using a model discussed   
   so far.

| Overfitting when the model trains to so much data that it follows the specific features of that singular data set and then the model cannot generalize to new data. When training the model, you can avoid overfitting by splitting the data into a training set and a test set for a K nearest Neighbor model. You can also pick a k value thats not too small or too big. If it’s too big it will catch a lot of the specifics and overfit. |
| --- |

1. What is the purpose of splitting data into different sets?

| The purpose of splitting data into different sets when using the K-nearest neighbors (KNN) algorithm is to evaluate the performance of the model and assess its ability to generalize to unseen data.  There are 3 sets: training set, validation set, and test set. The training set is the largest portion. It trains the Model and establishes the basis for making predictions. The validation set is also used during the training process to fine-tune the model's hyperparameters. Then, test set is used to assess the final performance. The model has never seen the data in the test set before. |
| --- |

*To submit this assignment, please refer to the instructions in the course*. 

Since we don’t want to add many dimensions to a model, how are we able to tell if a column should be encoded or not if it adds many binary features?