Assignment - 12

Comparing Algorithms Performances

HARSH SIDDHAPURA
1230169813

03/24/2024

PART - 1

	Decision Tree Classifier	K-Nearest Neighbor Classifier
Noise	Because decision trees may split data into smaller, more homogeneous groupings, they are very robust when processing noisy data. It is possible that noisy data points are anomalies with minimal impact on the tree's overall structure. Decision trees can utilize pruning approaches to reduce overfitting, which is exacerbated by noisy data. However, it is vital to note that decision trees are vulnerable to small-scale variations in the dataset. This sensitivity can lead to overfitting, reducing their utility, particularly when high levels of noise are present.	Noise may have an impact on the K-NN approach, which is based on computing distances between data points. The presence of noisy data points may cause incorrect classifications. Nonetheless, selecting an appropriate value for the parameter "k" can increase the algorithm's ability to control noise. As 'k' grows, the influence of nearby noisy data points reduces, making the classification process less susceptible to isolated noisy data occurrences. Because noisy data points can impact the classification of nearby points, K-NN classifiers with lower 'k' values may be more sensitive to noise. Larger "k" values, on the other hand, can reduce the impact of noise by considering more neighbors when making judgements.
Missing Values	The decision tree uses another splitting rule to accommodate missing elements. Tags can be defined to control the structure of the tree when certain attributes are not important. Decision trees can handle missing data because this will help them overcome critical problems. Decision trees can handle missing values by classifying data without considering missing features. Even without character, one can decide what is good. Interpolation techniques can be used to replace missing data with	The K-nearest neighbor (K-NN) method requires a large amount of data for distance estimation, so missing values are a problem. To resolve this issue, use imputation techniques or remove items with missing values. The K-NN algorithm can be applied to data sets with missing values using various imputation methods such as mean, median and regression-based imputation. K-NN classifiers cannot handle missing values, but they can be used with imputation techniques to fill in missing values before calculating distances, allowing for nice

	predictions before building the tree. Therefore, decision trees can resolve missing values by using the interaction method to replace missing values with predicted values or ignore missing factor features during data segmentation.	classification. It is important to note that missing values can affect the calculation of the distance between data points, skewing the distribution of results. Therefore, it is important to use the interpolation technique to replace missing values with estimated values before using the K-NN algorithm.
Redundant Attributes	Because the decision tree model selects the most common points at each node, balance elements are generally less affected. Segmentation methods (such as Gini impurity or data gain) drive the selection process. Repeated features can be ignored and features that provide more variance are selected when building the tree. But adding unique features can make the tree more complex, which can lead to over intrusion and make interpretation difficult. Therefore, it is better to use reduction or feature selection method to remove irregular features before training the tree algorithm. As the tree grows, duplicate features become less likely to be selected for splitting because they add less new information. The natural handling of redundant attributes by decision trees can be further improved by selecting algorithms that help identify and remove redundant attributes, such as information gain and Gini impurity.	Discontinuous features can negatively affect the performance of the K-nearest neighbor (K-NN) method with skew distance calculation. This is because connection or reconnection has a negative impact on the distance measurement, resulting in biased results. To solve this problem, use feature selection or dimension reduction such as principal component analysis (PCA) before using K-NN. This reduces the size of the dataset, improves the performance of the classifier, and reduces the effects of redundancy. It should be noted that the balance of features will lead to overfitting and increased computational cost. Therefore, limiting the number of features analyzed when distributed through exclusive selection or size reduction can help solve these problems. More importantly, to improve the performance of K-NN, such strategies generally need to reduce detection behavior during classification.

PART - 2

1. This tool evaluates the connectivity of dataset attributes and predicts class names. When there is a good relationship between these attributes and the class list, the prediction model becomes better because it indicates that these attributes provide important information for the prediction to be made. For example, if there is little correlation it will be difficult for the model to be accurate. More importantly, the strength of the relationship between dataset attributes and class labels can affect the model's ability to distinguish different classes and thus its accuracy in distribution.

2. The model's ability to accurately predict the relationship between classroom writing and meaningful behavior is critical to its effectiveness. This depends on the model's ability to capture and explain this relationship. If the model can adequately represent and represent this relationship, its predictive power can be increased. Conversely, if the model cannot describe this relationship, it will pose a significant challenge in making accurate predictions. This tool evaluates the model's ability to understand and capture relationships between labels and features.

   A well-designed model should be able to describe and represent the relationship between groups and attributes. If the model can capture these relationships, it can predict more uncertain data.

3. The performance and generalization ability of the model is greatly affected by training methods, which vary depending on the type of model used. These parameters include training cost and number of iterations for support vector machine (SVM) and neural networks (NN), impurity measure (DT) for decision tree, distance measure and nearest neighbors (KNN) to K neighbors. These hyperparameters are important because they can affect the performance of the model.

   For example, a learning curve that is too high will affect the convergence of the model and lead to deviations from the optimal solution. Similarly, using inappropriate impurity metrics in decision trees or inappropriate distance metrics in KNNs can impact the model's ability to identify patterns in the data, resulting in decreased performance. Therefore, careful selection and tuning of these hyperparameters is important to ensure model validity and ability to produce reliable classification results. Insufficient selection can lead to problems such as slow connections, inefficiencies, or poor performance, ultimately hindering the model's ability to identify new situations.