UNIVERSIDAD POLITÉCNICA DE YUCATÁN

- Machine Learning -

Decision trees regression

By Pablo Iván Martin Enríquez

Date: 5/12/2023

Decision Trees is a machine learning algorithm used for both classification and regression tasks. It's a tree-like model where an internal node represents a feature or attribute, the branch represents a decision based on that feature, and each leaf node represents the outcome, or a decision taken.

Decision tree builds regression or classification breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

The core algorithm for building decision trees called ID3 by J. R. Quinlan employs a top-down search through the space of possible branches with no backtracking. The ID3 algorithm can be used to construct a decision tree for regression by replacing Information Gain with Standard Deviation Reduction.

Tree Structure: The decision tree is structured as a hierarchical tree. Each internal node tests a specific feature, and the branches represent the possible values or outcomes of that feature.

Decision Making: At each internal node, the decision tree algorithm chooses the feature that best splits the data based on certain criteria. This process is repeated recursively until a stopping criterion is met, creating a tree-like structure.

Regression Trees: In the case of regression, the target variable is continuous. The algorithm aims to predict a value at each leaf node, and the predicted value is often the average of the training samples that reach that leaf.

Splitting Criteria: Common criteria for splitting nodes in decision trees include measures like Mean Squared Error (MSE) for regression tasks. The algorithm chooses the split that minimizes the impurity or error.

Pruning: Decision trees are prone to overfitting, capturing noise in the training data. Pruning involves removing parts of the tree that do not provide significant predictive power, helping to generalize better to new, unseen data.

Advantages: Decision trees are easy to understand and interpret. They can handle both numerical and categorical data, and their visual representation is intuitive.

Disadvantages: Decision trees can be sensitive to small variations in the data and might overfit. Ensemble methods like Random Forests or Gradient Boosting are often used to overcome these limitations.

Example:

As for the example, while I was doing my research, I stumbled upon a perfect one, but funny enough, it had many errors, so I took the liberty to fix some of said errors, mainly, the tree would not even generate after running the code, there were also many minor errors that I focused on fixing, for I really wanted to make it work.