In Class Assignment 5

Genesis Grant

CTEC 402

**PART 1:**

|  |  |  |
| --- | --- | --- |
| ML Algorithm | My Answer | ChatGPT Answer |
| Linear Regression | Linear regression is a mathematics concept applied to ml algorithms. It will find the line of best fit for related variables, and make predictions based on the trend of the line. | Linear Regression is a supervised learning algorithm used to predict continuous values by establishing a relationship between input features and an output variable. It works by fitting a straight line, known as the regression line, to the data points in such a way that the line minimizes the difference between the actual and predicted values. This line is often represented by the equation y=mx+b , where  m is the slope and  b is the y-intercept. Linear regression assumes a linear relationship between the features and the target variable, making it ideal for cases where changes in input variables produce proportional changes in the output. It is commonly used in cases like predicting prices, sales, or trends where the outcome is a numerical value. |
| Decision Tree | This algorithm makes a path of yes or no answer questions, then based on those path answers the result will be given. Each question you answer leads you to another, eventually reaching a final outcome. | A Decision Tree is a supervised learning algorithm used for both classification and regression tasks. It works by splitting the dataset into branches based on feature values, forming a tree-like structure where each node represents a decision based on one feature, and each leaf node represents an outcome or category. The tree recursively splits data based on the features that provide the most significant information gain, creating a model that resembles a series of "if-then" rules. This makes decision trees highly interpretable, as each path from root to leaf represents a decision path that leads to a prediction. They are useful for complex datasets with non-linear relationships, but they can easily overfit, which is often managed by limiting tree depth or using ensemble methods like Random Forest. |
| Naïve Bayes | Naive Bayes is a way to classify things by looking at features and using probabilities to predict categories. It assumes features don’t affect each other. It looks at individual features and calculates how likely it is for those features to be in each category, making its predictions based on that. It’s called "naive" because it assumes that all features are independent of each other, which isn’t usually the case, but it still works pretty well for many tasks, like classifying emails or text. | Naive Bayes is a classification algorithm based on Bayes' theorem, which calculates the probability of a class given certain features, assuming each feature contributes independently to the outcome. Despite this "naive" assumption of independence (which is rarely true in real-world data), the algorithm performs well for many tasks, especially in text classification and spam detection. Naive Bayes calculates the probability of each class and then chooses the most probable one for a given set of features. Its simplicity makes it fast and efficient, particularly for high-dimensional data, and it generally performs well even with relatively small training datasets. However, because it assumes feature independence, it may not work as well when features are highly correlated. |
| Random Forest | Random Forest is an expansion of the decision trees machine learning algorithm, that builds multiple decision trees and combines their predictions to improve accuracy. Each tree makes its own prediction, and the final decision is based on the majority vote from all the trees, making it more reliable than a single tree. | Random Forest is an ensemble learning algorithm that builds multiple decision trees and combines their results to improve accuracy and robustness. It creates each tree by using a random subset of the data and selecting random features at each split, which helps reduce overfitting and makes the model more generalized. During prediction, each tree in the forest casts a "vote," and the algorithm selects the class with the most votes (for classification) or averages the results (for regression). This approach makes Random Forest both powerful and versatile, as it can handle complex, non-linear relationships in data and is less prone to overfitting than individual decision trees. It’s commonly used in applications like fraud detection, medical diagnosis, and financial forecasting. |
| Dimensionality reduction algorithms | Dimensionality reduction is about making data simpler by reducing the number of features it has. Sometimes, data can have too many features, which makes it hard to analyze or visualize. These techniques help identify the most important features and cut out the rest, allowing you to keep the key information while making the dataset easier to work with. | Dimensionality reduction algorithms are techniques that reduce the number of features in a dataset while retaining as much important information as possible. By projecting data into a lower-dimensional space, these algorithms make it easier to visualize, process, and analyze data, especially when it has many features (or dimensions). Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are popular methods: PCA identifies the directions (principal components) where data varies the most, while LDA finds the axes that maximize class separation. Dimensionality reduction is particularly useful for simplifying models, speeding up computation, and addressing issues like overfitting in high-dimensional datasets. It is widely used in image processing, text analysis, and exploratory data analysis. |