| **Algorithm** | **Dataset Representation** | **Training Phase** | **Classification Phase** |
| --- | --- | --- | --- |
| **Logistic Regression** | Tabular format with | Adjusts the weights and biases of the logistic | Calculates the probability of an instance |
|  | feature columns | regression model based on the training data | belonging to a particular class |
|  | and a target column |  |  |
| **Decision Trees** | Tabular format with | Builds a decision tree by recursively splitting | Assigns a class label to instances based on |
|  | feature columns | the dataset based on feature conditions | traversing the decision tree |
|  | and a target column |  |  |
| **Random Forest** | Tabular format with | Constructs multiple decision trees using | Uses the ensemble of decision trees to |
|  | feature columns | random subsets of the dataset and features | make predictions by aggregating their |
|  | and a target column |  | individual predictions |
| **Support Vector** | Tabular format with | Finds the hyperplane that maximizes the margin | Determines the class of an instance based |
| **Machines (SVM)** | feature columns | between instances of different classes | on which side of the hyperplane it falls |
|  | and a target column |  |  |
| **k-Nearest Neighbors** | Tabular format with | Stores the feature vectors and class labels | Finds the k nearest neighbors of an |
| **(KNN)** | feature columns | of the training instances | instance and assigns the majority class |
|  | and a target column |  | label among its neighbors |
| **Naive Bayes** | Tabular format with | Estimates the class priors and conditional | Calculates the posterior probability of an |
|  | feature columns | probabilities based on the training data | instance belonging to each class and |
|  | and a target column |  | assigns the class label with the highest |
|  |  |  | probability |
| **Neural Networks** | Tabular format with | Learns the optimal weights and biases through | Applies the learned weights and biases to |
|  | feature columns | backpropagation algorithm using the training | compute the outputs of the neural network |
|  | and a target column | data | and assigns the class label with the |
|  |  |  | highest output activation |

In each case, the dataset is typically represented in a tabular format, with the feature columns containing the input variables or attributes, and the target column containing the class labels or target variable. During the training phase, the algorithms utilize the dataset to learn patterns and relationships between the features and the target labels. In the classification phase, the trained model or algorithm is used to make predictions or assign class labels to new, unseen instances based on their feature values.

It's important to note that the specific implementation and usage of each algorithm may vary, and additional preprocessing steps such as data normalization, feature engineering, or hyperparameter tuning might be involved. This table provides a general overview of how the dataset interacts with each classification algorithm.

Top of Form