1. **Linear Regression**:

Linear regression is used to predict a continuous target variable based on one or more input features. Let's consider a simple example of predicting house prices based on the size of the house.

Suppose we have a dataset with information about the size (in square feet) of several houses and their corresponding prices. We can use linear regression to model the relationship between the size of the house (feature) and its price (target variable).

For instance, the dataset might look like this:

```
markdown
Size (sq. ft.) | Price ($)
---------------------------
    1000       |   300,000
    1500       |   400,000
    2000       |   500,000
    2500       |   600,000
```

We can fit a linear regression model to this data to find a line that best fits the relationship between house size and price. Once the model is trained, we can use it to predict the price of a new house given its size.

2. **K-Nearest Neighbors (KNN)**:

K-nearest neighbors is a simple algorithm used for both classification and regression tasks. Let's consider a classification example of classifying fruits based on their color and weight.

Suppose we have a dataset with information about different fruits, including their color and weight, and their corresponding labels (e.g., "apple", "orange", "banana").

For instance, the dataset might look like this:

```
mathematica
Color   |  Weight (g)   |  Label
---------------------------------
Red     |     150       |  Apple
Orange  |     200       |  Orange
Yellow  |     120       |  Banana
```

Given a new fruit with unknown label but known color and weight, KNN classifies the fruit by looking at the k nearest fruits (based on color and weight) in the dataset and assigning the majority class among them to the new fruit.

3. **Decision Trees**:

Decision trees are versatile algorithms used for both classification and regression tasks. Let's consider a classification example of classifying whether a person will play tennis based on weather conditions.

Suppose we have a dataset with information about different days, including weather conditions (e.g., outlook, temperature, humidity, wind), and whether people played tennis on those days (label).

For instance, the dataset might look like this:

```
yaml
Outlook  |  Temperature  |  Humidity  |  Wind  |  Play Tennis
--------------------------------------------------------------
Sunny    |     85        |     85     |  False |    No
Overcast |     83        |     86     |  True  |    Yes
Rainy    |     70        |     96     |  False |    Yes
```

A decision tree can be trained on this data to learn rules that predict whether a person will play tennis based on the weather conditions. The decision tree might look like:

```
java
            Outlook = Sunny
            /           \
      Humidity <= 75    Play Tennis = No
         /       \
Temperature <= 80   Play Tennis = Yes
```

This tree can then be used to predict whether a person will play tennis on a new day based on its weather conditions.

1. **K-Means Clustering**:

K-Means Clustering is an unsupervised learning algorithm used to partition a dataset into K clusters. Let's consider a dataset with two features: annual income and spending score, and we want to segment customers into different groups based on their purchasing behavior.

Suppose we have the following dataset:

```
markdown
CustomerID | Annual Income (k$) | Spending Score (1-100)
---------------------------------------------------------
     1     |        15          |           39
     2     |        15          |           81
     3     |        16          |           6
     4     |        16          |           77
     5     |        17          |           40
```

To apply K-Means Clustering, we need to specify the number of clusters (K). Let's say we choose K = 3. The algorithm will then iteratively assign each data point to the nearest centroid and update the centroids until convergence.

After running K-Means Clustering, we might get clusters like this:

Cluster 1:

- Customers with low annual income and low spending score.

Cluster 2:

- Customers with moderate annual income and high spending score.

Cluster 3:

- Customers with high annual income and moderate spending score.

We can use these clusters to gain insights into customer segments and tailor marketing strategies accordingly.

2. **Logistic Regression**:

Logistic Regression is a supervised learning algorithm used for binary classification tasks. Let's consider an example of predicting whether a student will pass or fail an exam based on the number of hours they study.

Suppose we have the following dataset:

```
markdown
Hours Studied | Exam Result (Pass/Fail)
--------------------------------------
      2       |       Fail
      3       |       Fail
      4       |       Fail
      5       |       Pass
      6       |       Pass
```

To apply Logistic Regression, we'll model the relationship between the number of hours studied (feature) and the likelihood of passing the exam (target variable). The logistic regression model will output probabilities between 0 and 1, representing the probability of passing the exam.

After training the model, we can use it to predict whether a student will pass or fail based on the number of hours they study. For example, if a student studies for 4 hours, the model might predict a 60% chance of failing and a 40% chance of passing