← Go Back to Machine Learning

Course Content

# Supervised Learning - Classification

### Classification

Classification is a type of **supervised learning** problem in which the dependent variable is categorical in nature.

For example, predicting whether an email is **spam or not**, a credit card customer is **fraudulent or not**, a customer will **churn or not**, etc. We can have more than two categories in the target variable as well.

There are many classification techniques. We will learn about two of the most common classification algorithms: **Logistic Regression and K-Nearest Neighbors (K-NN)**

### Logistic Regression

As we saw in Linear Regression, we fit a line $Y = \theta_0^* + \theta_1^* X + W$ to our data. However, linear regression is not ideal for the situation in which we only need to output binary values like 0 and 1, as seen in the image below.

To overcome this problem, we add **non-linearity** to our regression equation. This is done using a **Sigmoid function** which is given as:

$$P = \frac{1}{(1 + exp\{-(\theta_0^* + \theta_1^* X)\})}$$

Where, $X$ is the set of input features and $P$ is the probability.

The values of a **Sigmoid function** range from **0 to 1** which makes it suitable to **represent probability**. The curve on the right side in the below figure represents the sigmoid function.

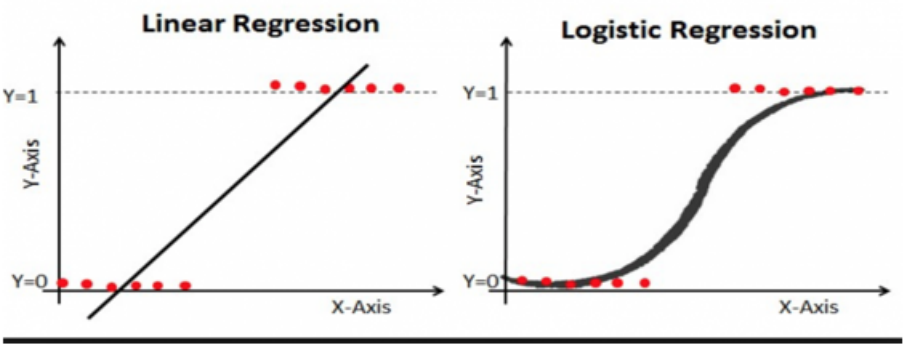



Image Source

**In easy terms:** Logistic regression uses the Sigmoid function to calculate the probability of the output feature $Y$ given some input features $X$.

**K-Nearest Neighbors**

K-NN can be used for both classification and regression problems. However, it is more widely used for classification problems.

**How does the K-NN algorithm work?**

We follow the below steps for K-NN:

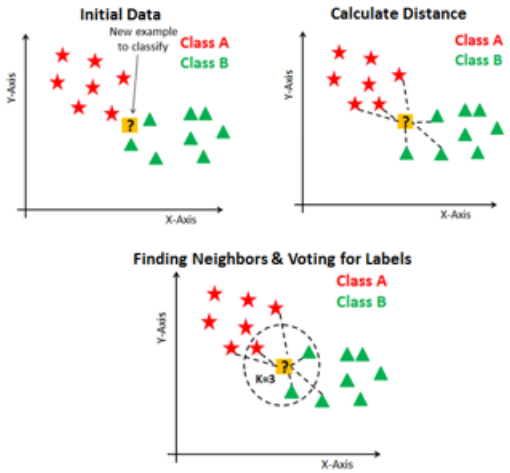
1. **Select K:** The first step is to select the value of K. It indicates **how many points** have to be **considered from the training data** while **classifying an unseen record**. The 'K' in the K-NN algorithm is the count of the nearest neighbors we wish to take the vote from. Generally, 'K' is taken to be odd when the number of classes is even to avoid a tie of votes for all classes.

2. **Calculate distance (Euclidean, Manhattan, etc.):** To estimate the closeness of a test data point with data points in the training data, we need to calculate the distance between the test data point and the training data points.

3. **Find the K nearest neighbors:** After the distance is computed, the K nearest points are selected.

4. **Vote for labels:** The labels of the selected K nearest neighbors are considered and the label with the majority count is assigned as the output label of the test data point.

**Let's consider an example:** Suppose there are 2 classes - class A and class B and you want to know, to which class the new data point belongs.



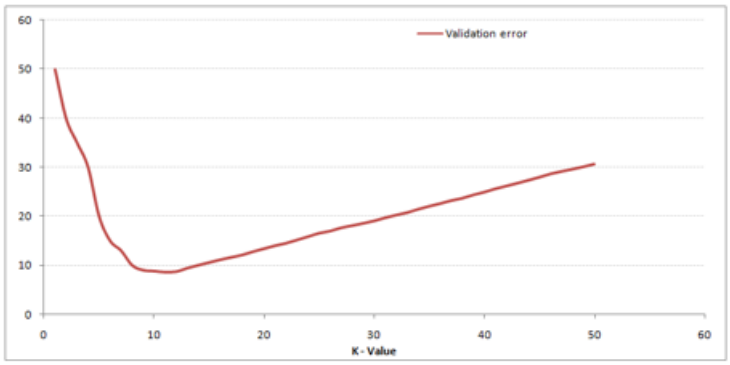

Let's say K = 3. Hence, we will now make a circle with the new data point as the center just as big as enclosing only three data points on the plane as depicted in the above images.

After that, we take a vote on which class is appearing the most in the circle and assign the new data point to that class. In our example, we will assign it to class B.

**How to choose the value of K in K-NN?**

The value of 'K' can be chosen by looking at the validation error. As we keep increasing the value of 'K', the value of the validation error decreases to some extent and after that, it starts to increase. Therefore, the value of K for which we get the

**least validation error** is the 'K' we choose for our algorithm. In the below image, we can take K=10 as it has the least validation error.




Previous

Next



Help