# **K-Nearest Neighbors (KNN)**

# **Classification Model with scikit-learn**

**How it works**

The entire training dataset is stored. When a prediction is required, the k-most similar records to a new record from the training dataset are then located. From these neighbors, a summarized prediction is made.

**Why choose the KNN over other models?**

The dataset chosen was very small only 900 records with low dimensionality. So using KNN provided these advantages:

**Advantages  
1. KNN is known for its high level of accuracy in making predictions:** and it can achieve high accuracy in a wide variety of prediction-type problems. KNN is most useful when labeled data is too expensive or impossible to obtain

**2. No Training Period:** KNN is called **Lazy Learner (Instance based learning)**. There is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g. SVM, Linear Regression etc.  
  
**3.** Since the KNN algorithm requires no training before making predictions, **new data can be added seamlessly** which will not impact the accuracy of the algorithm.  
  
**4.** KNN is very **easy to implement**. There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)  
  
**Disadvantages**  
  
**1. Does not work well with large dataset:**In large datasets, the cost of calculating the distance between the new point and each existing points is huge which degrades the performance of the algorithm.  
  
**2. Does not work well on data with high dimensionalities:**The KNN algorithm doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate the distance in each dimension.  
  
**3. Need feature scaling:** We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we don't do so, KNN may generate wrong predictions.  
  
**4. Sensitive to noisy data, missing values and outliers**: KNN is sensitive to noise in the dataset. We need to manually impute missing values and remove outliers.

**Input to a KNN Model**

Predicting Heat Disease  
The data set consists of around 900 observations on 17 features X. Some of the features are categorical. Age is the non-binary feature. The target Y variable HeartDisease is binary, which can take on two values, 1 or 0.

**Data cleaning and Preprocessing**

1. The dataset must provide features X, that directly impacts a binary target Y-variable.
2. Correct variable names (no spaces)

**Normalizing & Splitting the Data**

1. Preprocessing: normalize Age variable
2. Transform other variables into dummies
3. Split data set in training and test set

**Fitting and Evaluating the Model**

**Choosing an appropriate K value**

Used Cross Validation to get the best value of K, using **scikit-learn’s cross\_val\_score** function. We can see from the chart that k = 3, 5, 7, 10 all have an accuracy score of just under 85%**, as these are tied for the best score, it is advisable to use a smaller value for k. This is because when using higher values of k, the model will use more data points that are further away from the original.**

A screen shot of a computer

Description automatically generated**K = 5 gives the highest accuracy score**

**Evaluation Metrics is used to assess the accuracy of the model.**

Three common metrics used to assess the quality of the model, as shown in the sklearn Classification report are:

**1. Precision**: Percentage of correct positive predictions relative to total positive predictions.

**2. Recall**: Percentage of correct positive predictions relative to total actual positives.

**3. F1 Score**: A weighted harmonic mean of precision and recall. The closer to 1, the better the model.

* F1 Score: 2 \* (Precision \* Recall) / (Precision + Recall)

A screenshot of a computer

Description automatically generated

**Conclusion**

The table above shows the that the KNN Model with a K=5 value, has an accuracy of 92% in predicting patients with Heart Disease and an 81% of predicting patients without Heart Disease.