**Classification Models**

1. **Nearest Neighbors**

<https://scikit-learn.org/stable/modules/neighbors.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

1. First read the file
2. Impute missing values, e.g

* df.isnull().sum()
* mean = df['Item\_Weight'].mean()
* df['Item\_Weight'].fillna(mean, inplace =True)
* mode = df['Outlet\_Size'].mode()
* df['Outlet\_Size'].fillna(mode[0], inplace =True)

1. Deal with categorial variables and drop the id columns

* df.drop(['Item\_Identifier', 'Outlet\_Identifier'], axis=1, inplace=True)

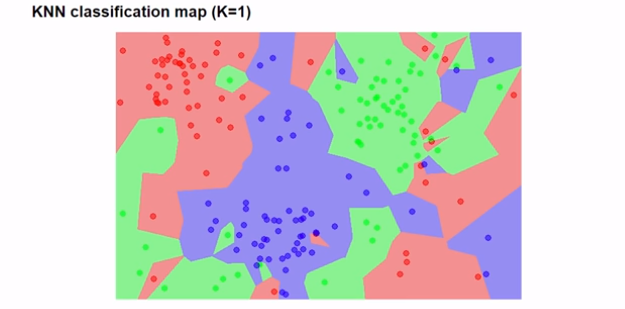
1. Create train and test sets
2. Scale the features
3. Find the best value for K

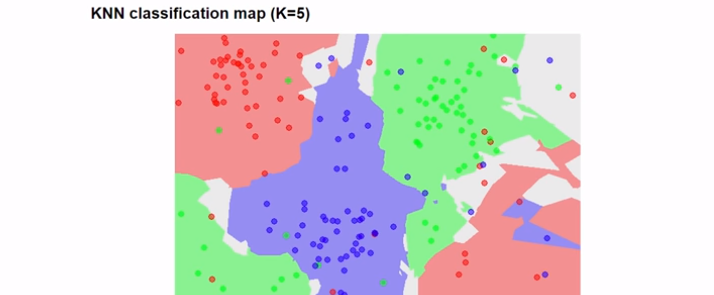
**K-nearest neighbors (KNN) classification**

- Pick a value for K

- Search for the K observations in the training data that are nearest to the measurements of the unknown

- Use the most popular response value from the K nearest neighbors as the predicted value of unknown value





Scikit- learn 4 step modeling pattern

1. Import the class you want to use
2. Instantiate the estimator (model)
3. Fit the model with data

e.g. knn.fit(X, y)

1. Predict the response of the new observation

e.g. knn.predict([list]) : returns a numpy array

or list = ([list1],[list2])

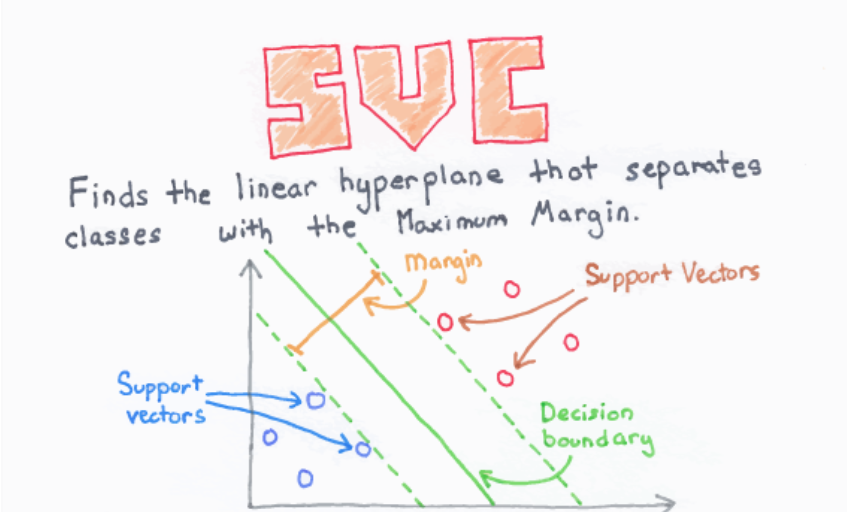
then knn.predict([list])

1. Logistic Regression

* Used for prediction of output which is binary, only two outcomes
* Also, for a data set in which there are one or more independent variables that determine an outcome

1. Support Vector Machines (SVM)

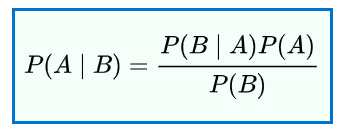
* Its based on the concept of decision planes that define decision boundaries
* It finds the hyperplane that maximizes the margin between the two classes with the help of support vectors



1. Naïve Bayes Classifier

* Based on Bayes’ Theorem with an assumption of independence among predictors
* **Bayes’ Theorem** : describes the probability of an event based on prior knowledge of events that might be related to the event.
* For example, if cancer is related to age, their age can be used to predict the probability of cancer made without knowledge of their age.

That is



1. Decision Trees

* For both classification and regression
* Breaks down data into smaller subsets which form nodes and leaf nodes of a tree.

1. Random Forest

* Construct a multitude of decision trees at training time and output the class that is the mode of the classes.
* It corrects for decision trees habit of overfitting data by increasing bias and reducing variance

For a detailed comparison of classification models

<https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222>