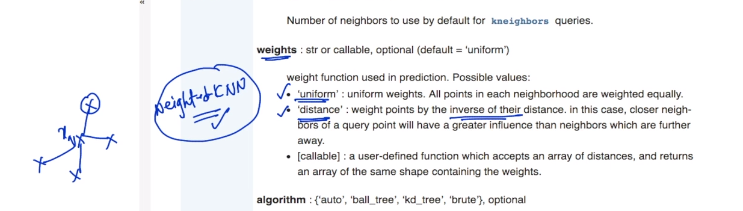
**Code sample for KNN with 9 datasets.**

**First lets see documentation for KNN in sklearn**

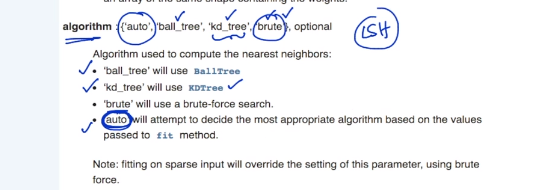


**As we can see in above data set the n\_neighbors is the parameters which is actually our K in KNN and its default value is 5.**

**Weights is the parameter which mean if we want to any particular weight to any parameter or by using uniform every parameter will be given equal weight. Just like we saw in weighted KNN we assigned weight to distance i.e., 1/d**

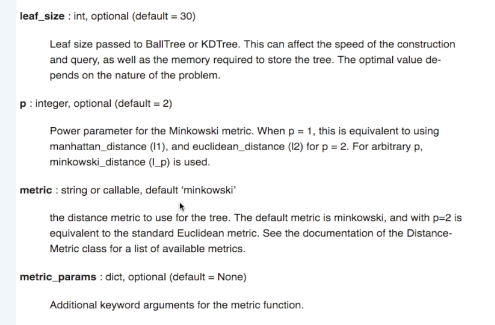


**The 3rd parameter is algorithm.**



**We have 3 options in algo that is “ball\_tree”, “kd\_tree”, “brute” and by using “auto” we can let function decide best suited algorithm itself.**

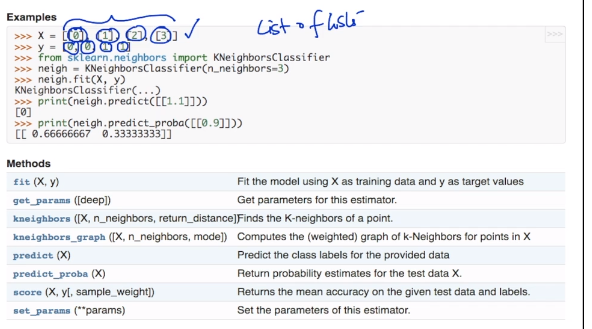
**It does not have LSH.**



**In above we have 4 more parameters.**

**leaf\_size is when we are using kd\_tree we can change the default value of leaf\_size i.e., 30 to any other and number and again all computation time and space it requires will depend on value of leaf\_size.**

**P is Power parameter for Minkowski metric which is by default 2 i.e., L2 norm which means standard Euclidean metric and when change it to 1 we get manhatten distance.**



**This is how we code for KNN.**

**X is list of list with some univariate or 1-dimensional data point.**

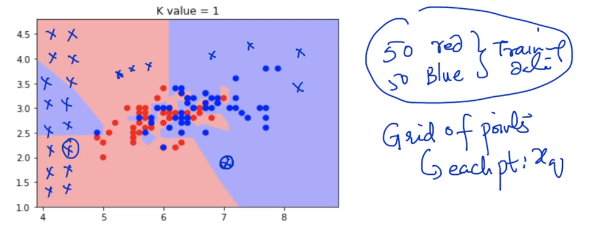
**Y is a class label.**

**And this both combine to form out D\_train.**

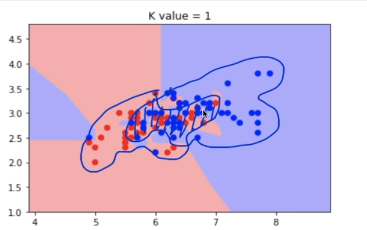
**Neigh.fit(X,Y) means we are and training out model on our training data points and class label.**

**Neigh.predict is used for predicts class label for any test data point and neigh.predict\_proba is used to predict probabilistic class label.**

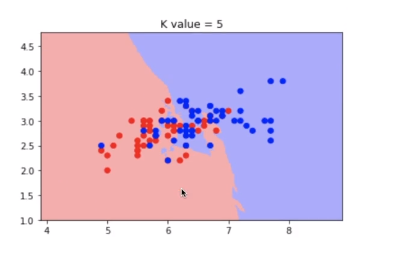
**Here are some snapshots for how the value of k affects the output on some toy data sets.**



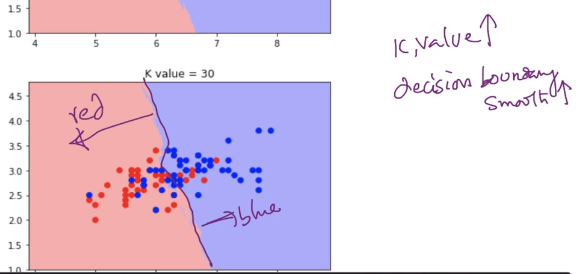
**We are creating a grid and trying to determine class label for every grid value in training phase and based on that we check in which grid our query data point fall which will tell us the value of class label.**



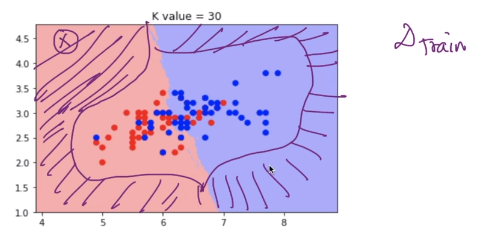
**As we can see overlap in blue and red region so we will see how value of k affects the decision boundary. And we can see in above even because of one blue point in left side the whole region is marked as blue and we can think of it as an outlier.**



**As value of k is increased the decision surface got more smoothened and outlier effect decreases.**

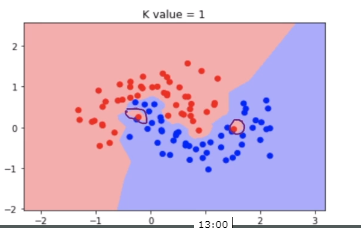


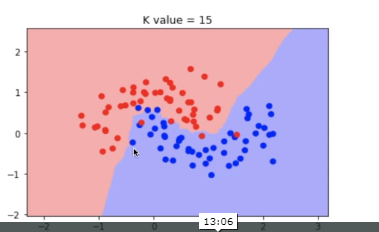
**But there is one problem with it also**

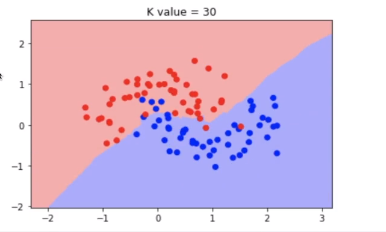


**As we can see in above image. The shaded region is not having any data points in our training data and if we get any query points in that region based on nearest neighbours we get class label which is wrong instead we should have got not known or something similar.**

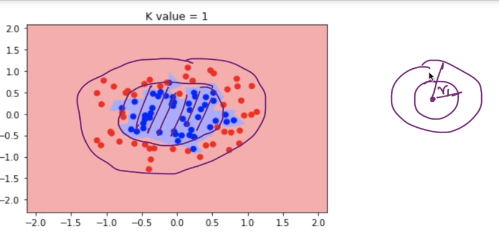
**Now lets check the effect of value of k in ushape dataset.**

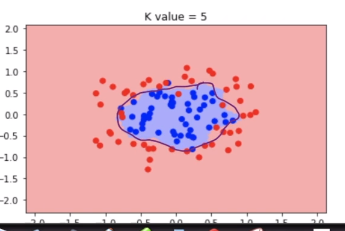


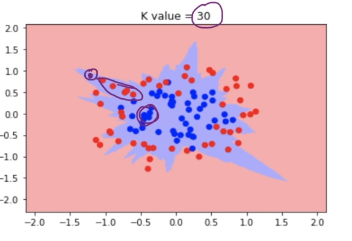




**Lets use another data set called concentric circle.**



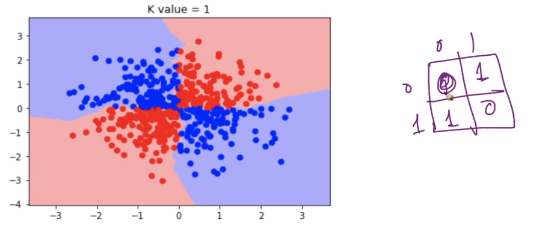




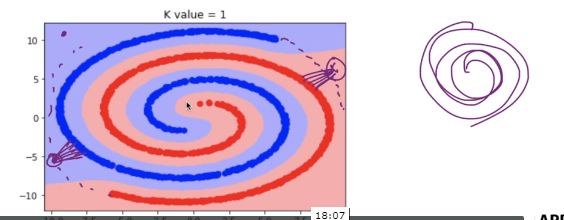
**IN above graph we see that our query point is near to red point but since we there is lot of blue points also in bundle so while finding 30 NN it might get more blue points and resulting in wrong class label.**

**So we need to be careful about this also.**

**Now lets see another data set i.e., X-OR data set**

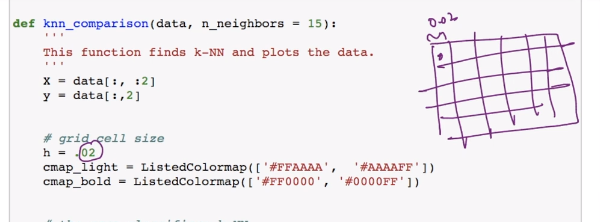


**Lets see two spirals data sets**

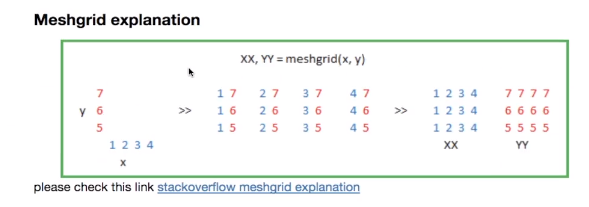


**Since we don’t have enough data points in extended spirals and if we get our query point there so we will get wrong class label because of NN.**

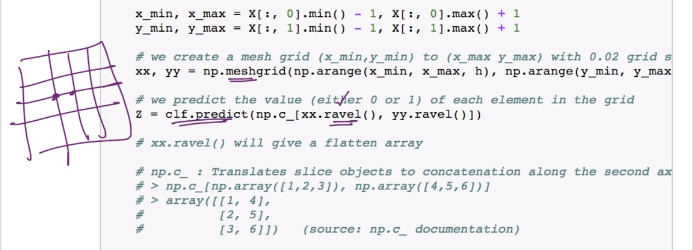
**Now lets come to coding part.**



**h is our grid size .**



**It literally creates a grid.**



**Meshgrid is used to create grids and then we are predicting class label using predict.**

**And ravel is used to pass flattened array.**