Ranjit Kathiriya

R00183586

MSC. AI

COMP9061 - Practical Machine Learning

Assignment - 1

Dr Ted Scully

Part - 1 Development of Basic Nearest Neighbour Algorithm

- KNN Algorithm:
  - KNN is very powerful supervised machine learning algorithm. It can be used for both classification as well as regression for predicting values.
  - KNN is a classification algorithm that works on a simple principal.
  - For example: we had some imaginary data on Dogs and Horse with height and weights.

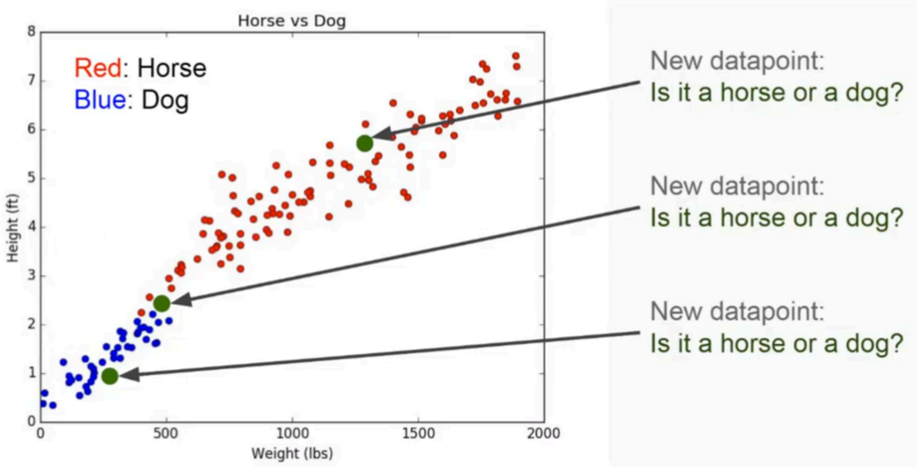
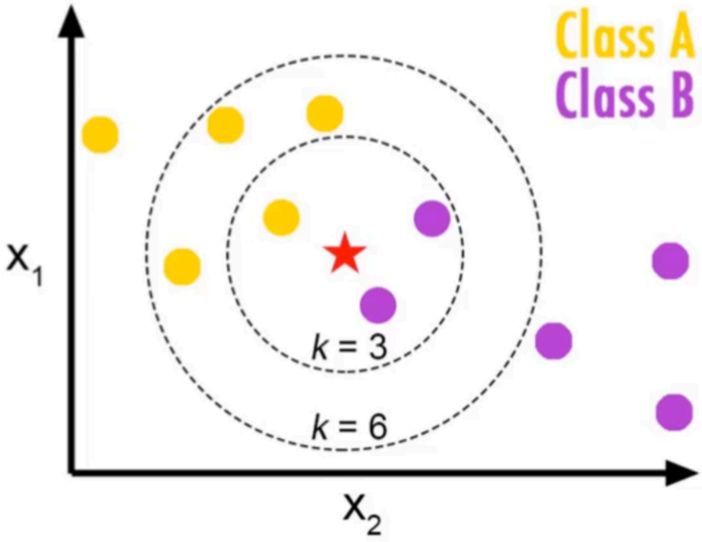


Fig-1

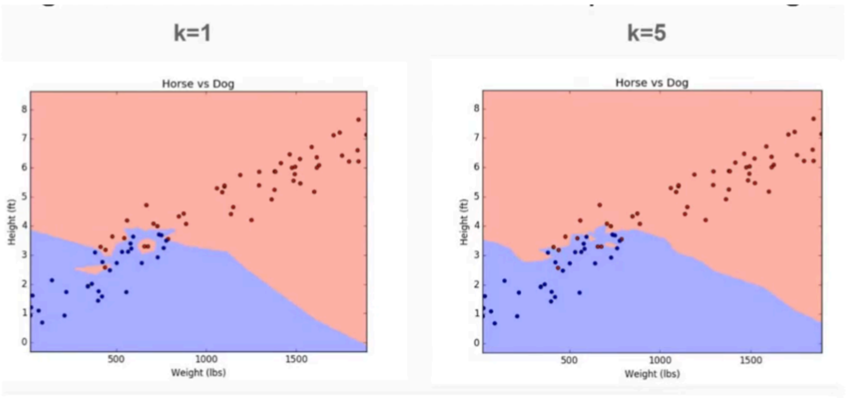
  - In this given Figure 1, it is observed all blue dots are dogs and red is for hoarse class and it is seen that as the Hight and weight increases the chances of hoarse is more.
  - We have 3 green points one the basis of the class Horse and dog. It is easily predictable that weather the given point in green is dog or horse, but if we have more datasets at that time it is difficult to predict all datasets.
  - Prediction Algorithm
    - I. Calculate the distance from X to all points in dataset.
    - II. Sort the points in your dataset by increasing order.
    - III. Predict the majority label of ‘K’ closest points.

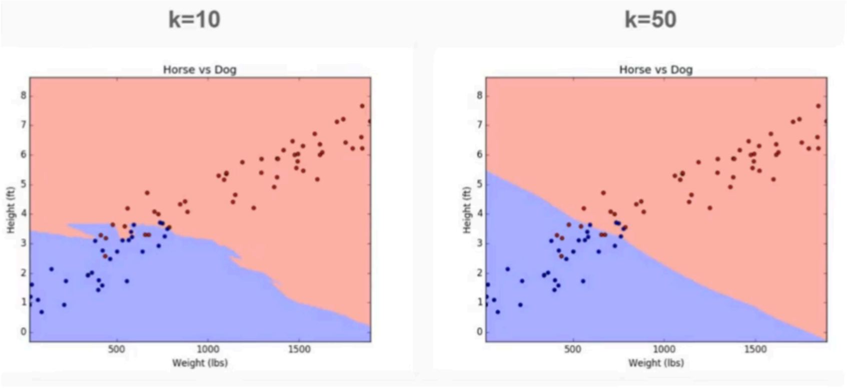
- Choosing K Value

- To select proper ‘K’ we run the algorithms several time with different values of k.
- Given diagrams below describes how important is to choose the appropriate K- Value.
- The change in the K- Value it may affect the overall accuracy of the model.
- Let’s take K as 1 then it is observed that our accuracy may be stable.



- Explore more in depth for k = different values in the above figure 1’s example.





- With his graphs we can observed that by changing the value of K. It will affect on what class the point is with K=10 and 50.

➢ How to find the k-Nearest Neighbors?

- To find the minimum distance the algorithm calculate the distance between Train dataset and Test dataset, It received the distance between 2 points.
- There are mainly three methods to calculate the neighbors.

  I. Euclidean Distance: It is calculated by training and testing data. Euclidean distance is calculated as the square root of the sum of squared between a Xtest dataset and an Xtrain

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2+(q_2-p_2)^2+\cdots+(q_n-p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

  II. Manhattan Distance: The total sum of X and Y points into Graph.

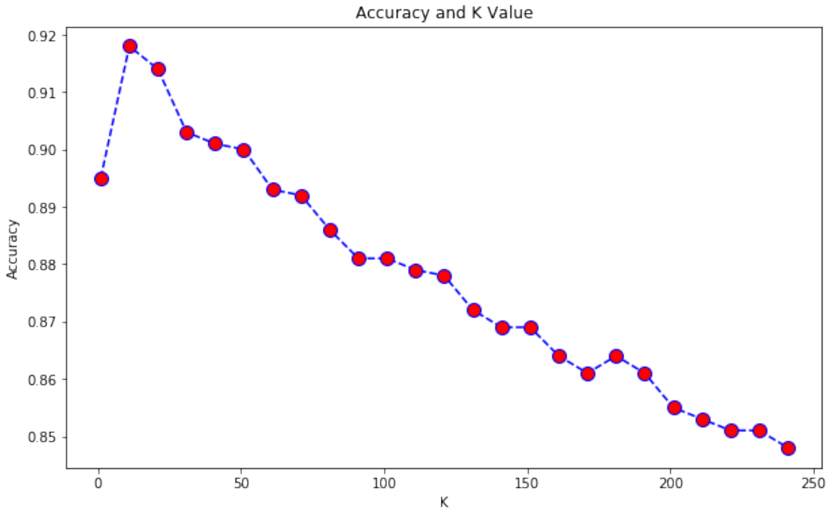
$$\boldsymbol{distance} = \sum_{\boldsymbol{i}=\mathbf{0}}^{\boldsymbol{n}-\mathbf{1}} |(\boldsymbol{x}[\boldsymbol{i}] - \boldsymbol{y}[\boldsymbol{i}])|$$

III. Hamming Distance: Assume Xtrain and Ytrain are two datasets of an attribute which represents animals (dog, horse). Hamming distance is one possible matrix.

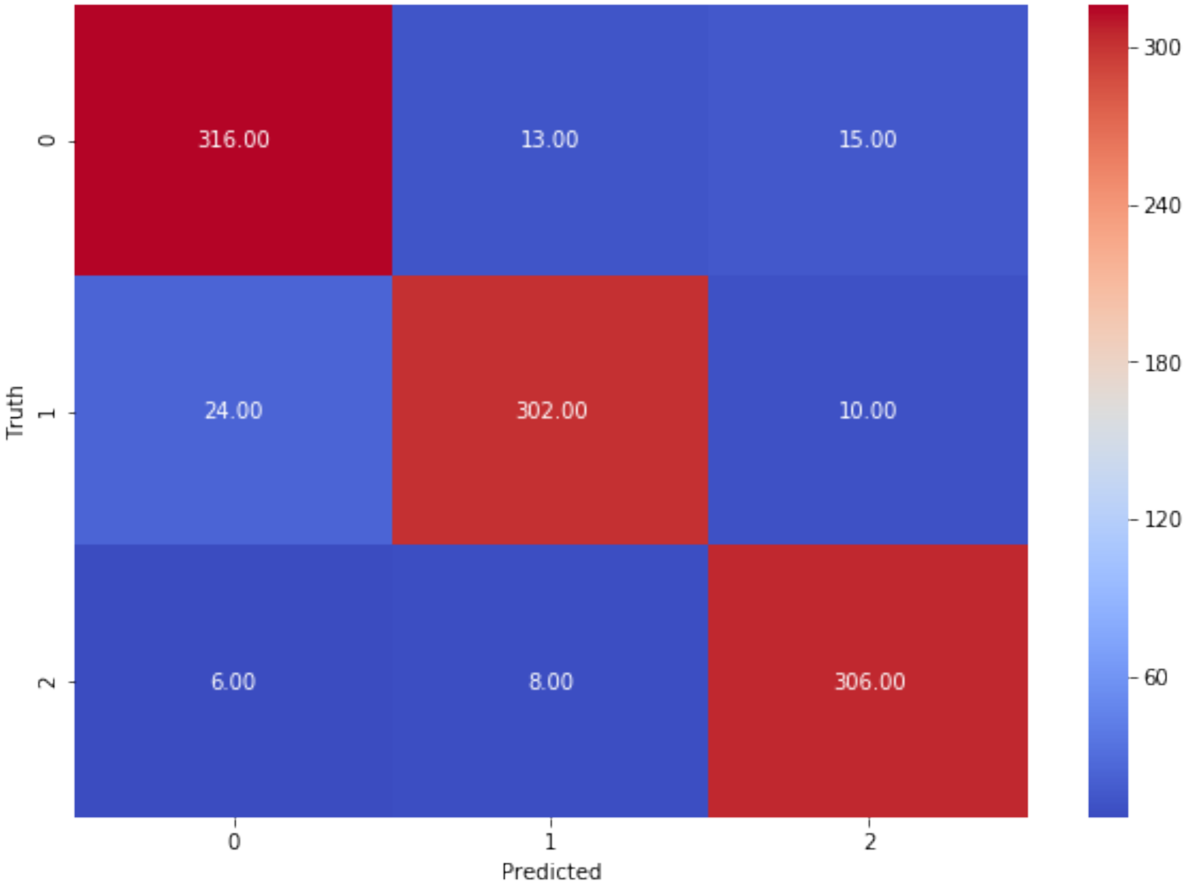
$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$if x = y \Rightarrow 0$$
$$if x \neq y \Rightarrow 1$$

- Advantages
  - Very simple
  - Works with n numbers of class
  - Easy to append more data.
  - Also works with Regression
  - Few parameters
    - I. K- Value
    - II. Distance Matrix with above methods
- Disadvantages
  - Worst for larger datasets
  - High predicting cost.
  - Not good with more dimensional data.
- Results
  - This figure shows the accuracy of the different K's value.



- For K = 5 the confusion matrix is given below. This figure describes the wrong class prediction. Here 0,1,2, means classed and Red color denotates that the prediction is True and blue color means it is wrong predict.



Part 2 -Investigating k-NN variants and hyper-parameters

2.1 Distance-Weighted variant of K-NN

- In distance-Weighted of KNN is an modified version of KNN. In KNN the major issue faced by an algorithm is choosing an hyper parameters of K. If the parameters is small then the basic KNN will give accurate results, but if the K's parameters is larger at that time the accuracy is not that stable.
- It takes majorities vote and gives the predicated value.
- Formula for calculating the Distance-Weighted is given below.

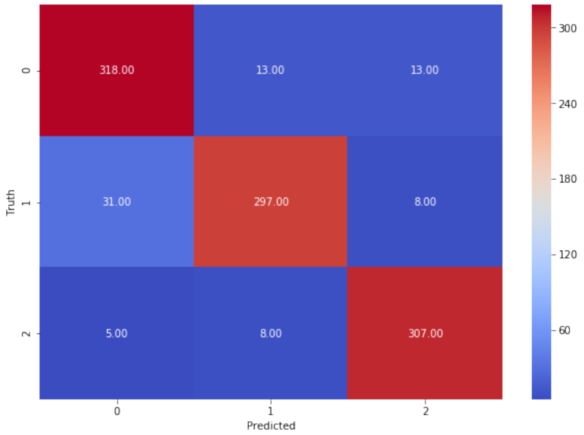
$$vote(c_j) := \sum_{i=1}^{k} \frac{1}{d(\mathbf{x}_q,\mathbf{x}_i)^n}(c_i, c_j)$$

2.2 Results of Distance-Weighted K-NN and K-NN at K=10

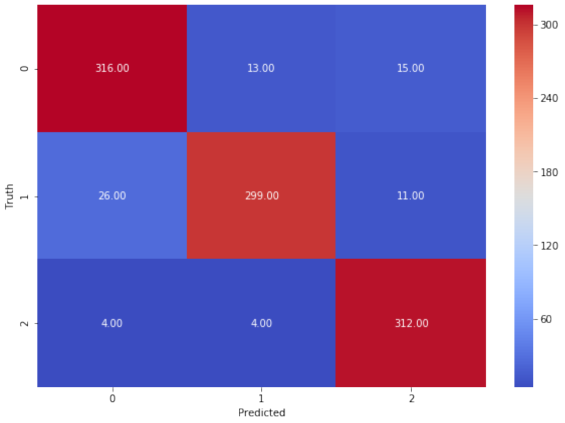
- Final Results: Accuracy

K- VALUE	KNN	Distance-Weighted K-NN
1	0.895	0.895
5	0.924	0.924
10	0.922	0.927

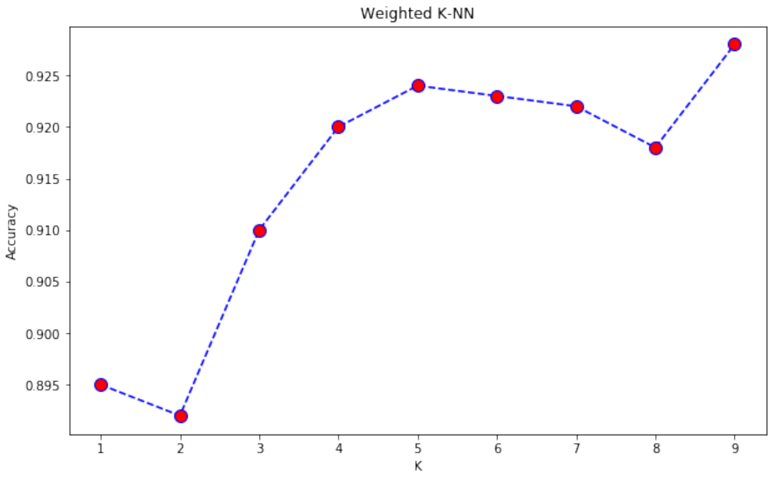
- Description: In Distance-Weighted K-NN I have observed it works fine with the different hyperparameters. Like if the K value is to big then also it provides better accuracy then normal KNN.
    - As we can observed from given table that the accuracy at k =10 outraced the KNN.
- Confusion Matrix: For k =10
    - In this given below matrix shows the true positive, negative and false positive and negative values of the class.
    - For example: Distance-Weighted K-NN figure the model has predict the value is 0 and the truth number is 316. As same happened with the class 1,2 and many values are wrong predicted by the model. To overcome that in next part lets evaluated with different hyper parameters.



KNN's



Distance-Weighted K-NN



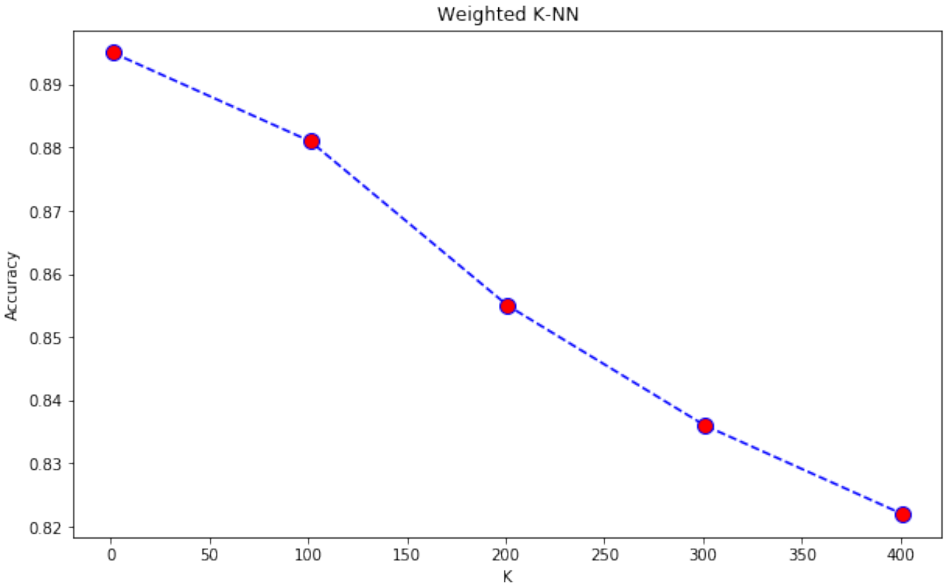
2.3 Different Hypermeters:

- In this Part – b, we will mainly focus on the hypermeters.
  - By changing K's value
  - MinMax Scalar

- Squaring the distance

- By changing K's value:
    - By changing the K's value in distance-weighted KNN's model, I have observed that as my k is small for example in between 1 to 20 then my accuracy is good.
    - When my k is more big the accuracy is decreasing.
    - As the K's plays an important role for error correction and building accurate model, I believe that K's value change may affect the accuracy of the model.
    - The given table and graphs describes how the K's value is important for accuracy.

K – Value	Accuracy
1	0.895
101	0.881
201	0.855
301	0.836
401	0.822
1500	0.79



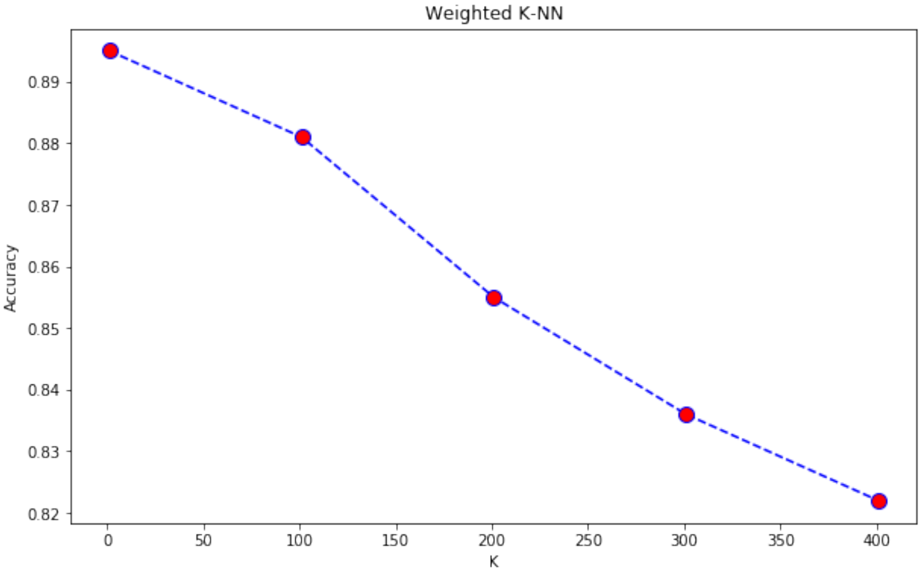
- MinMax Normalization
    - Minmax normalization is a technique for obtaining the great accuracy in some cases. In this the x is subtracted from minimum value of xi and then divided with the max of xi – min of xi.
    - It is used for dimension reduction. If the one feature value is depended on another at that time to overcome this problem min max normalization is used.
    - Over all it is observed the min max normalization not works proper with this.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

    - Table:

K’s Value	Accuracy
1	0.895
101	0.881
201	0.855
301	0.836
401	0.822

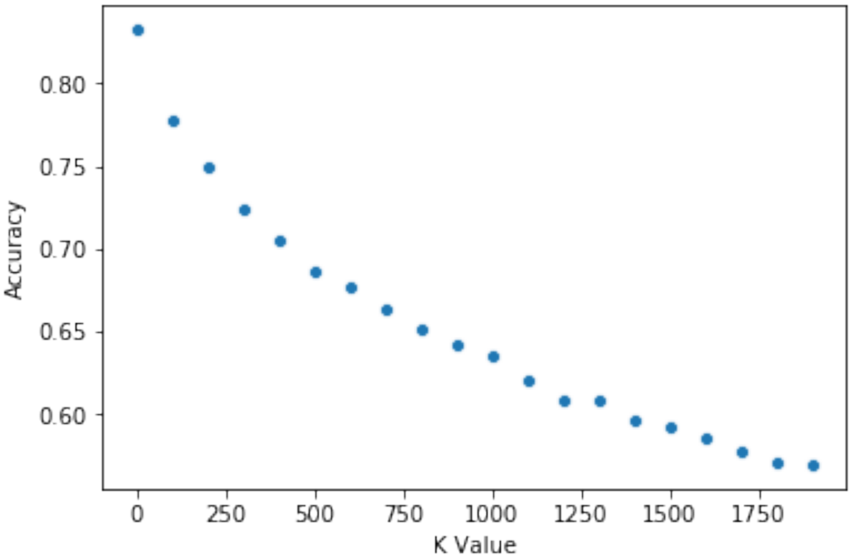
    - Graphs:



- Squaring the Weighted-distance
  - In this the weighted distance is squared.
  - This also doesn’t give that much accuracy.
  - Table:

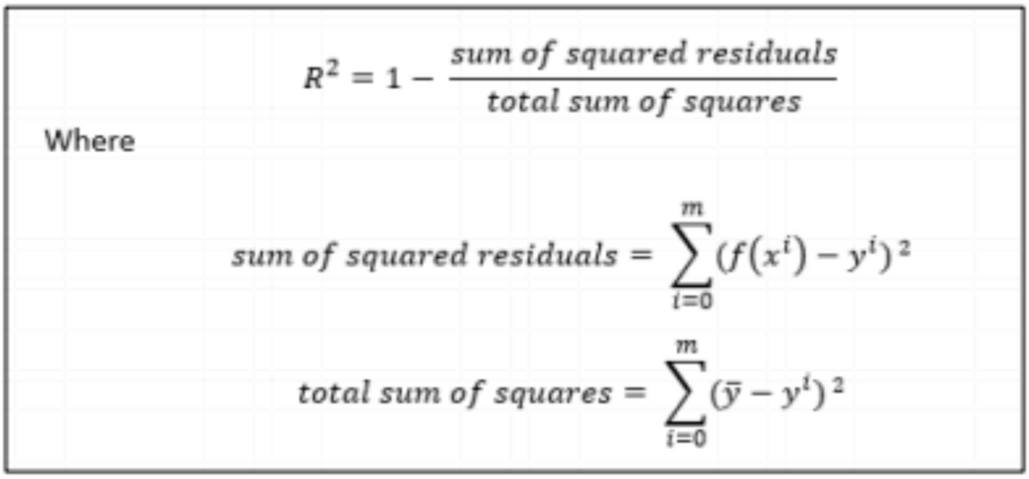
K’s Value	Accuracy
1	0.888
101	0.839
201	0.813
301	0.790
401	0.779
1501	0.678

  - Graphs:


0.80
0.75
0.70
0.65
0.60
Accuracy
0
250
500
750
1000
1250
1500
1750
K Value

Part 3 – Developing k-NN for Regression Problems

3.1 k-NN for Regression Problems



$$R^2 = 1 - \frac{sum\ of\ squared\ residuals}{total\ sum\ of\ squares}$$

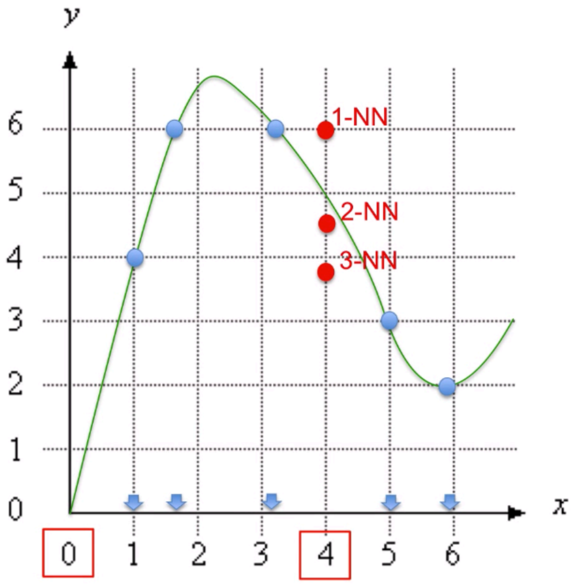
Where

$$sum\ of\ squared\ residuals = \sum_{i=0}^{m}(f(x^i) - y^i)^2$$

$$total\ sum\ of\ squares = \sum_{i=0}^{m}(\bar{y} - y^i)^2$$

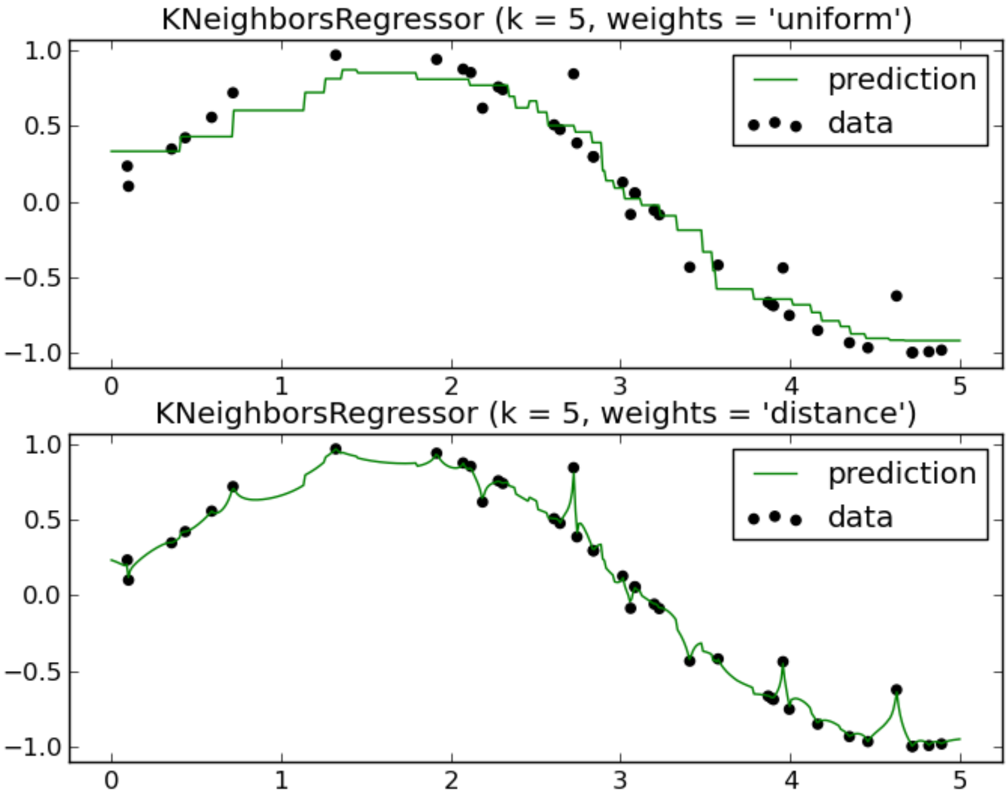
(10 M

- The regression is done here.
- Each time the value is tested using the above formula and the output is generated.
- In this there is no class. It has an value in Y train and test data set.



Copyright © 2014 Victor Lavrenko

○



3.2 Standard Scalar:

- The idea behind the standard scaler is that it has the mean =0 and standard deviation = 1.
- In the full data set each value is subtracted by a sample mean and then divided by the standard deviation of all dataset.

- It will also normalization of each and every features individually.

Standardization:

$$z = \frac{x - \mu}{\sigma}$$

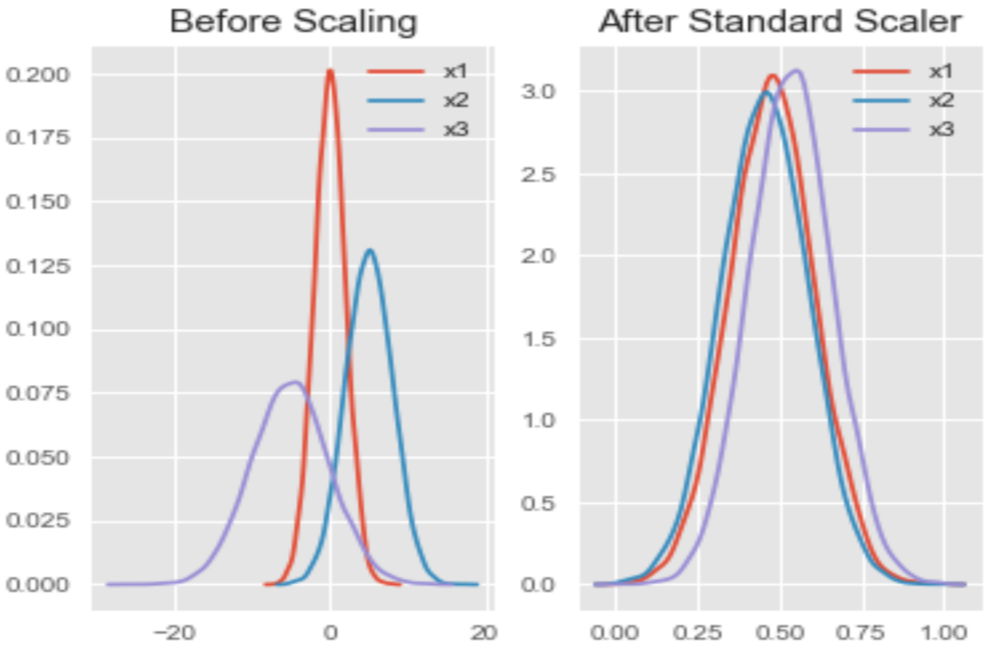
with mean:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (x_i)$$

and standard deviation:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2}$$

- The figure below describes that the data before applying the Standard scalar and after applying the scalar.



- It is observed that the data is properly scaled. In this diagram, all features of X1,X2,X3 before scaling this will affect on accuracy and it will create noise.
- The scaling is important because the data is improper and each feature is not properly recognized properly by class. If we scaled this data then we are overcoming this problem.

- Apart from standard scaler I have also looked many other scaler like.
    - MinMaxScaler
    - MaxAbsScaler
    - StandardScaler
    - RobustScaler
    - And many more.
- I have picked standard scalar because all the features in this it scaled in properly manner with mean and standard deviation.
- Table:

	KNeighborsRegressor	KNeighborsRegressor Standard scalar
r2_score	0.8386441475860921	0.8413465305666075