**K Nearest Neighbors Algorithm**

**K-NN**

**Short note about K-NN**:

The k-nearest neighbors algorithm (k-NN) is a [non-parametric](https://en.wikipedia.org/wiki/Non-parametric_statistics) method used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and regression.

**Non-Parametric**: Non-parametric methods are widely used for studying populations that take on a ranked order

### When do we use KNN algorithm****?****

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique we generally look at 3 important aspects

(classification trees are used when the response or target variable is categorical in nature. Regression trees are needed when the response variable is numeric or continuous)

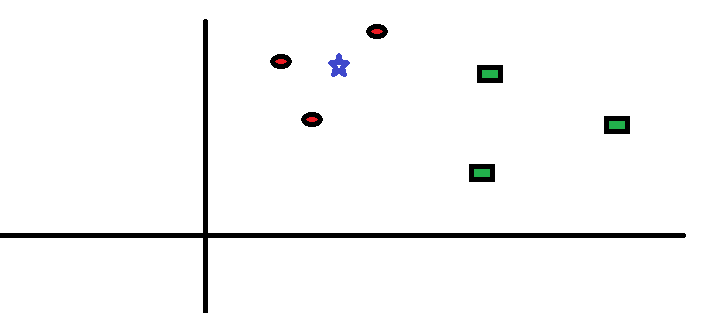
1. Ease to interpret output

2. Calculation time

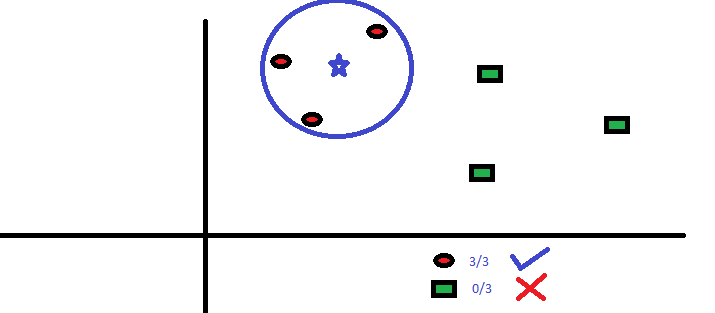
3. Predictive Power

### ****How does the KNN algorithm work?****

Following is a spread of red circles (RC) and green squares (GS) :

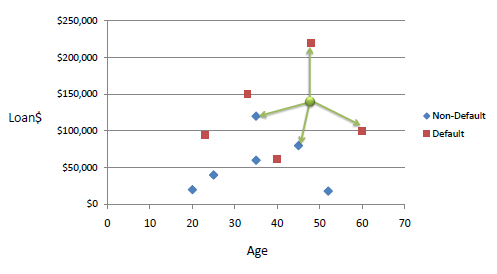
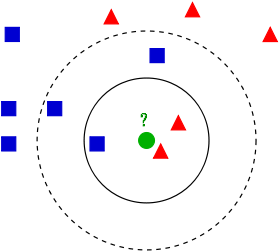


“K” is KNN algorithm is the nearest neighbors we wish to take vote from. At k=3



The three closest points to BS is all RC. Hence, with good confidence level we can say that the BS should belong to the class RC.

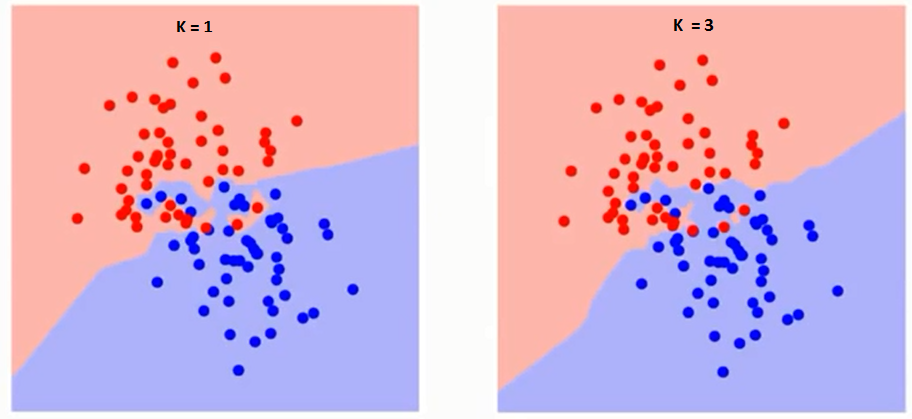
**KNN Classification**

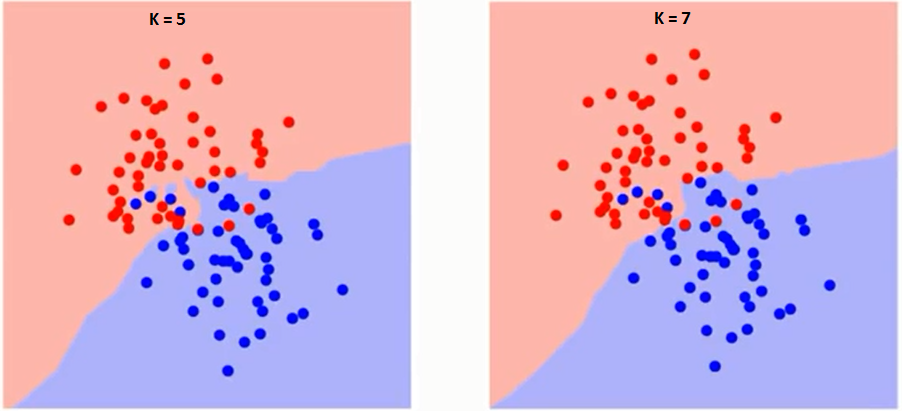


**Example** of k-NN classification. The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If k = 3 (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If k = 5 (dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).

### How do we choose the factor K?

effect of value “K” on the class boundaries. Following are the different boundaries separating the two classes with different values of K.

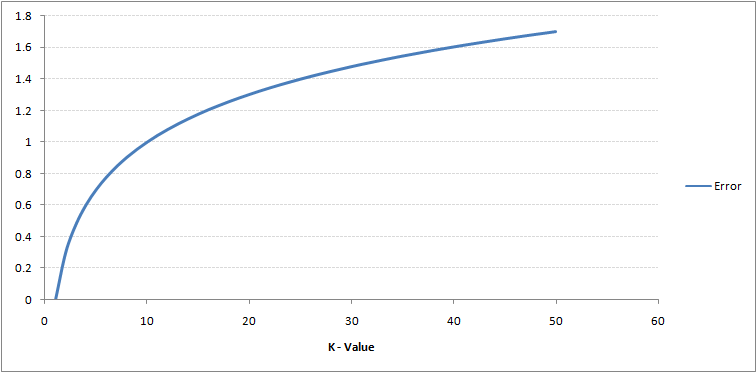




**Step1**

Graphical represerntation:

**Training error rate with varying value of K :**



### the error rate at K=1 is always zero for the training sample. This is because the closest point to any training data point is itself. Hence the prediction is always accurate with K=1. If validation error curve would have been similar, our choice of K would have been 1.

### error curve with varying value of K:

### training error_1

### Hence, error rate initially decreases and reaches a minima. After the minima point, it then increase with increasing K. To get the optimal value of K, you can segregate the training and validation from the initial dataset.

### Example-Following is the data from questionary survey(to ask people opinion to classify whether ,A special paper tissue is good or not.

|  |  |  |
| --- | --- | --- |
| **Acid(durability in seconds)** | **strength(Kg/sqmeter)** | **Classification** |
| 7 | 7 | Bad |
| 7 | 4 | Bad |
| 3 | 4 | Good |
| 1 | 4 | Good |

### Let us consider x2=Acid(durability)

### y2=Strength

### Query instance(x1,y1)=(3,7)

### 

### 

### Step2-Hence distance to query instance is below:

|  |  |  |
| --- | --- | --- |
| **Acid(durability in seconds)** | **strength(Kg/sqmeter** | **Distance to query instance(3,7)** |
| 7 | 7 | 4 |
| 7 | 4 | 5 |
| 3 | 4 | 3 |
| 1 | 4 | 3.6 |

### Step-3 Rank the minimum Distance-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Acid(durability in seconds)** | **strength(Kg/sqmeter** | **square distance to query instance(3,7)** | **Rank Minimum Distance** | **Classification** |
| 7 | 7 | 4 | 3 | Bad |
| 7 | 4 | 5 | 4 | Bad |
| 3 | 4 | 3 | 1 | Good |
| 1 | 4 | 3.6 | 2 | Good |

### Step-4 Determining nearest neighbours based on kth minimum distance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Acid(durability in seconds)** | **strength(Kg/sqmeter** | **square distance to query instance(3,7)** | **Rank Minimum Distance** | **Classification** | **Including 3rd nearest neighbors** |
| 7 | 7 | 4 | 3 | Bad | Yes |
| 7 | 4 | 5 | 4 | Bad | No |
| 3 | 4 | 3 | 1 | Good | Yes |
| 1 | 4 | 3.6 | 2 | Good | Yes |

### Classifying based on the Rank

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Acid(durability in seconds)** | **strength(Kg/sqmeter** | **square distance to query instance(3,7)** | **Rank Minimum Distance** | **Including 3rd nearest neighbors** | **Classification** |
| 7 | 7 | 4 | 3 | Yes | Bad |
| 7 | 4 | 5 | 4 | No | \_ |
| 3 | 4 | 3 | 1 | Yes | Good |
| 1 | 4 | 3.6 | 2 | Yes | Good |

### Step5-Using the majority of the category of nearest neighbors as the prediction value

### of query instance,we have 2 Good and 1 Bad.

### Conclusion: Since 2>1 we conclude that a new paper tissue that pass laboratory test

### with x1=3, x2=7 is included in Good Category.

### Confusion Matrix-

### C:\Users\Rama\Downloads\IMG-20180306-WA0012.jpg

**UseCases-**

1. Finding the response of product based on features.
2. Finding the chances to get king while playing cards.
3. Finding obese person with adiposity prone to heart disease
4. Finding Product Price with respect to other vendors
5. Finding Age and immunity
6. Finding temperature vs. Number of cones sold at ice cream store
7. Finding Population vs Food consumption
8. Finding quantity with yield
9. Determining the chances to win cricket match .
10. Determining the chances of getting Jobs after Completing Graduation.
11. Speed and distance relationship
12. Finding rate of growth of the economy of a Institution

Applications:

KNN as a data mining technique has a wide variety of applications in classification as well as regression. Some of the applications of this method are mentioned below:

* **Text mining:** The KNN algorithm is one of the most popular algorithms for text categorization or text mining. Some of the most recent works on this topic are for instance. Different numbers of nearest neighbours are used for different classes in this approach, rather than a fixed number across all classes. In this way, the only parameter that needs to be chosen by the user when using KNN, the K value, becomes less sensible and hence it does not need to be carefully chosen as in the standard algorithm.
* **Agriculture:** In general, KNN is applied less than other data mining techniques in agriculture related fields. It has been applied, for instance, for simulating daily precipitations and other weather variables. Another interesting application is the evaluation of forest inventories and for estimating forest variables. In these applications, satellite imagery is used, with the aim of mapping the land cover and land use with few discrete classes. The other applications of the k-NN method in agriculture include climate forecasting and estimating soil water parameters.
* **Finance:** applications of KNN in finance are mentioned below:

• Forecasting stock market: Predict the price of a stock, on the basis of company performance measures and economic data.

• Currency exchange rate

• Bank bankruptcies

• Understanding and managing financial risk

• Trading futures

• Credit rating

• Loan management

• Bank customer profiling

• Money laundering analyses

* **Medicine**:

• Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack. The prediction is to be based on demographic, diet and clinical measurements for that patient. • Estimate the amount of glucose in the blood of a diabetic person, from the infrared absorption spectrum of that person’s blood.

• Identify the risk factors for prostate cancer, based on clinical and demographic variables.

* The KNN algorithm has been also applied for analysing micro-array gene expression data, where the KNN algorithm has been coupled with genetic algorithms, which are used as a search tool. Other applications include the prediction of solvent accessibility in protein molecules, the detection of intrusions in computer systems, and the management of databases of moving objects such as computer with wireless connections.

h θ ( x ) = 1 1 + e − θ T x {\displaystyle h\_{\theta }(x)={\frac {1}{1+e^{-\theta ^{T}x}}}}

**Python Code-**

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv("C:\\Users\\Rama\\Desktop\\KNN1.csv")

print(dataset)

X = dataset.iloc[:, [1,1]].values

y = dataset.iloc[:, 4].values

# Splitting the dataset into the Training set and Test set

from sklearn.cross\_validation import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Fitting K-NN Regression to the Training set

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors = 2, metric = 'minkowski', p = 2)

classifier.fit(X\_train, y\_train)

# Predicting the Test set results

y\_pred = classifier.predict(X\_test)

# Making the Confusion Matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

# Visualising the Training set results

from matplotlib.colors import ListedColormap

X\_set, y\_set = X\_train, y\_train

X1, X2 = np.meshgrid(np.arange(start = X\_set[:, 0].min() - 1, stop = X\_set[:, 0].max() + 1, step = 0.01),

np.arange(start = X\_set[:, 1].min() - 1, stop = X\_set[:, 1].max() + 1, step = 0.01))

plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),

alpha = 0.75, cmap = ListedColormap(('red', 'green')))

plt.xlim(X1.min(), X1.max())

plt.ylim(X2.min(), X2.max())

for i, j in enumerate(np.unique(y\_set)):

plt.scatter(X\_set[y\_set == j, 0], X\_set[y\_set == j, 1],

c = ListedColormap(('red', 'green'))(i), label = j)

plt.title('KNN Regression (Training set)')

plt.xlabel('Acid(durability in seconds)')

plt.ylabel('strength(Kg/sqmeter')

plt.legend()

plt.show()

# Visualising the Test set results

from matplotlib.colors import ListedColormap

X\_set, y\_set = X\_test, y\_test

X1, X2 = np.meshgrid(np.arange(start = X\_set[:, 0].min() - 1, stop = X\_set[:, 0].max() + 1, step = 0.01),

np.arange(start = X\_set[:, 1].min() - 1, stop = X\_set[:, 1].max() + 1, step = 0.01))

plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),

alpha = 0.75, cmap = ListedColormap(('red', 'green')))

plt.xlim(X1.min(), X1.max())

plt.ylim(X2.min(), X2.max())

for i, j in enumerate(np.unique(y\_set)):

plt.scatter(X\_set[y\_set == j, 0], X\_set[y\_set == j, 1],

c = ListedColormap(('red', 'green'))(i), label = j)

plt.title('K-nn Regression (Test set)')

plt.xlabel('Acid(durability in seconds')

plt.ylabel('Acid(durability in seconds)')

plt.legend()

plt.show()

output:

runfile('C:/Users/Rama/Desktop/KNN Python.py', wdir='C:/Users/Rama/Desktop')

Acid(durability in seconds) strength(Kg/sqmeter \

0 7 7

1 7 4

2 3 4

3 1 4

square distance to query instance(3,7) Rank Minimum Distance \

0 4.0 3

1 5.0 4

2 3.0 1

3 3.6 2

Including 3rd nearest neighbors Classification

0 1 Bad

1 0 \_

2 1 Good

3 1 Good

C:\Users\Rama\Anaconda3\lib\site-packages\sklearn\cross\_validation.py:41: DeprecationWarning: This module was deprecated in version 0.18 in favor of the model\_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are different from that of this module. This module will be removed in 0.20.

"This module will be removed in 0.20.", DeprecationWarning)

C:\Users\Rama\Anaconda3\lib\site-packages\sklearn\utils\validation.py:475: DataConversionWarning: Data with input dtype int64 was converted to float64 by StandardScaler.

warnings.warn(msg, DataConversionWarning)

￼

￼

runfile('C:/Users/Rama/Desktop/KNN Python.py', wdir='C:/Users/Rama/Desktop')

Acid(durability in seconds) strength(Kg/sqmeter \

0 7 7

1 7 4

2 3 4

3 1 4

square distance to query instance(3,7) Rank Minimum Distance \

0 4.0 3

1 5.0 4

2 3.0 1

3 3.6 2

Including 3rd nearest neighbors Classification

0 1 Bad

1 0 \_

2 1 Good

3 1 Good

C:\Users\Rama\Anaconda3\lib\site-packages\sklearn\utils\validation.py:475: DataConversionWarning: Data with input dtype int64 was converted to float64 by StandardScaler.

warnings.warn(msg, DataConversionWarning)

￼

￼

runfile('C:/Users/Rama/Desktop/KNN Python.py', wdir='C:/Users/Rama/Desktop')

Acid(durability in seconds) strength(Kg/sqmeter \

0 7 7

1 7 4

2 3 4

3 1 4

square distance to query instance(3,7) Rank Minimum Distance \

0 4.0 3

1 5.0 4

2 3.0 1

3 3.6 2

Including 3rd nearest neighbors Classification

0 1 Bad

1 0 \_

2 1 Good

3 1 Good

C:\Users\Rama\Anaconda3\lib\site-packages\sklearn\utils\validation.py:475: DataConversionWarning: Data with input dtype int64 was converted to float64 by StandardScaler.

warnings.warn(msg, DataConversionWarning)

Traceback (most recent call last):

File "<ipython-input-3-57c1ea67d220>", line 1, in <module>

runfile('C:/Users/Rama/Desktop/KNN Python.py', wdir='C:/Users/Rama/Desktop')

File "C:\Users\Rama\Anaconda3\lib\site-packages\spyder\utils\site\sitecustomize.py", line 705, in runfile

execfile(filename, namespace)

File "C:\Users\Rama\Anaconda3\lib\site-packages\spyder\utils\site\sitecustomize.py", line 102, in execfile

exec(compile(f.read(), filename, 'exec'), namespace)

File "C:/Users/Rama/Desktop/KNN Python.py", line 24, in <module>

y\_pred = classifier.predict(X\_test)

File "C:\Users\Rama\Anaconda3\lib\site-packages\sklearn\neighbors\classification.py", line 145, in predict

neigh\_dist, neigh\_ind = self.kneighbors(X)

File "C:\Users\Rama\Anaconda3\lib\site-packages\sklearn\neighbors\base.py", line 347, in kneighbors

(train\_size, n\_neighbors)

ValueError: Expected n\_neighbors <= n\_samples, but n\_samples = 3, n\_neighbors = 4

runfile('C:/Users/Rama/Desktop/KNN Python.py', wdir='C:/Users/Rama/Desktop')

Acid(durability in seconds) strength(Kg/sqmeter \

0 7 7

1 7 4

2 3 4

3 1 4

square distance to query instance(3,7) Rank Minimum Distance \

0 4.0 3

1 5.0 4

2 3.0 1

3 3.6 2

Including 3rd nearest neighbors Classification

0 1 Bad

1 0 \_

2 1 Good

3 1 Good

C:\Users\Rama\Anaconda3\lib\site-packages\sklearn\utils\validation.py:475: DataConversionWarning: Data with input dtype int64 was converted to float64 by StandardScaler.

warnings.warn(msg, DataConversionWarning)

