

Machine Learning Algorithms

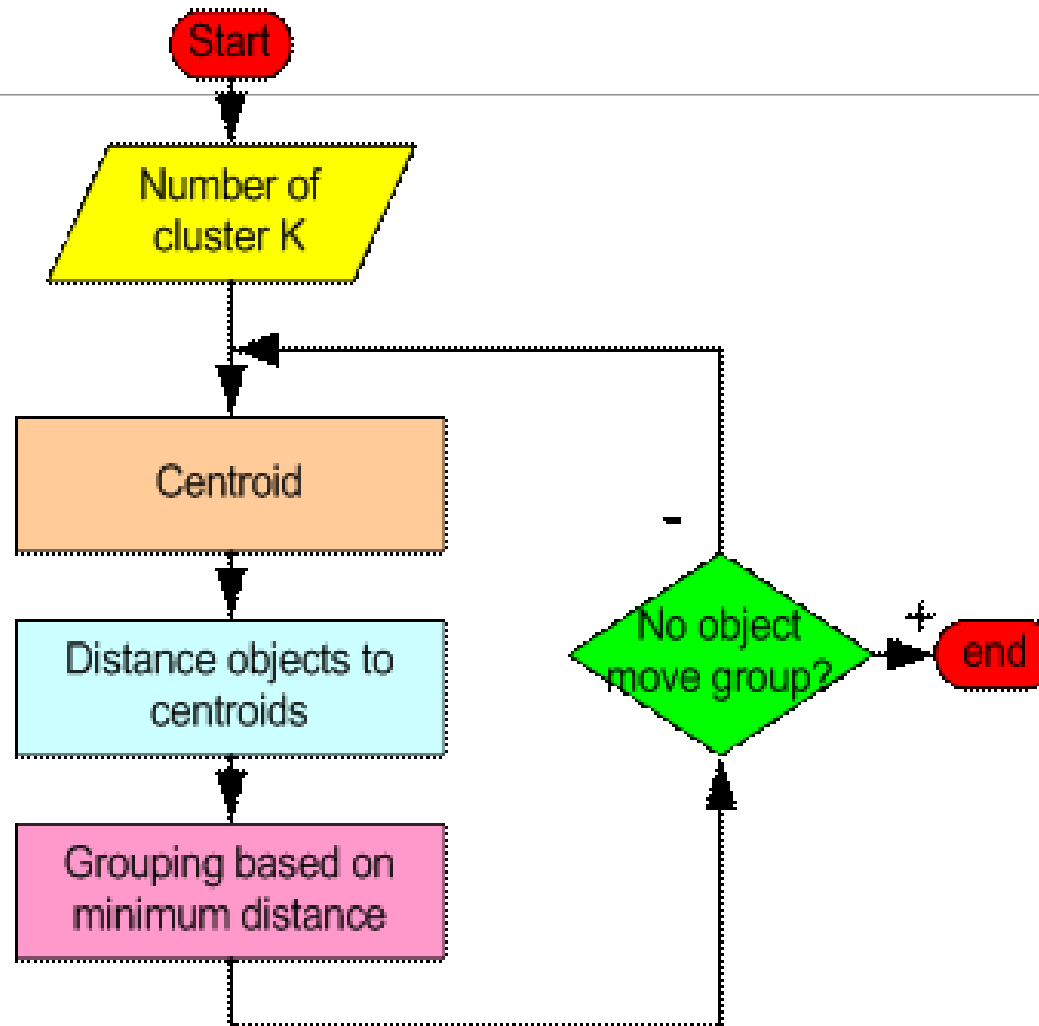
K-MEANS CLUSTERING

KNN

Naive Baise

AHMED YOUSRY

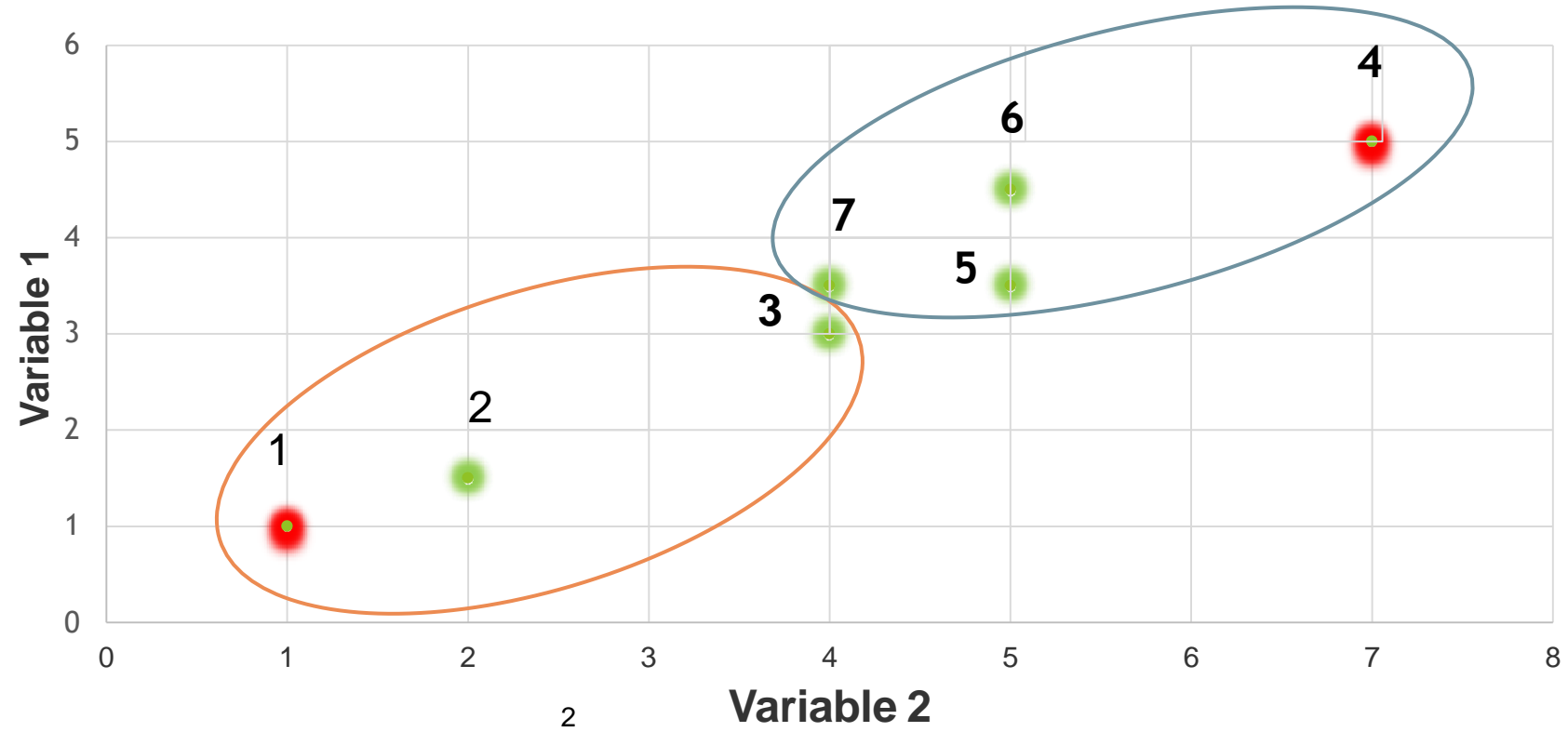
How the K-Mean Clustering algorithm works?



A Simple example k-means (using $K=2$)

Individual	Variable 1	Variable 2
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5

$K = 2$



Step 1:


Initialization: Randomly we choose following two centroids ($k=2$) for two clusters. In this case the 2 centroid are: $m_1=(1.0,1.0)$ and $m_2=(5.0,7.0)$.

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2:

	Centroid 1	Centroid 2
1	$\sqrt{(1 - 1)^2 + (1 - 1)^2} = 0$	$\sqrt{(5 - 1)^2 + (7 - 1)^2} = 7.21$
2	$\sqrt{(1 - 1.5)^2 + (1 - 2)^2} = 1.12$	$\sqrt{(5 - 1.5)^2 + (7 - 2)^2} = 6.10$
3	$\sqrt{(1 - 3)^2 + (1 - 4)^2} = 3.61$	$\sqrt{(5 - 3)^2 + (7 - 4)^2} = 3.61$
4	$\sqrt{(1 - 5)^2 + (1 - 7)^2} = 7.21$	$\sqrt{(5 - 5)^2 + (7 - 7)^2} = 0$
5	$\sqrt{(1 - 3.5)^2 + (1 - 5)^2} = 4.72$	$\sqrt{(5 - 3.5)^2 + (7 - 5)^2} = 2.5$
6	$\sqrt{(1 - 4.5)^2 + (1 - 5)^2} = 5.31$	$\sqrt{(5 - 4.5)^2 + (7 - 5)^2} = 2.06$
7	$\sqrt{(1 - 3.5)^2 + (1 - 4.5)^2} = 4.30$	$\sqrt{(5 - 3.5)^2 + (7 - 4.5)^2} = 2.92$

Step 2:

- Thus, we obtain two clusters containing: 
 $\{1,2,3\}$ and $\{4,5,6,7\}$.
- Their new centroids are:

$$\text{Group 1} = \left(\frac{1+1.5+3}{3} \right), \left(\frac{1+2+4}{3} \right) = (1.83, 2.33)$$

$$\text{Group 2} = \left(\frac{5+3.5+4.5+3.5}{4} \right), \left(\frac{7+5+5+4.5}{4} \right) = (4.12, 5.38)$$

Step 3:

	Centroid 1	Centroid 2
1	$\sqrt{(1.83 - 1)^2 + (2.33 - 1)^2} = 1.57$	$\sqrt{(4.12 - 1)^2 + (5.38 - 1)^2} = 5.38$
2	$\sqrt{(1.83 - 1.5)^2 + (2.33 - 2)^2} = 0.47$	$\sqrt{(4.12 - 1.5)^2 + (5.38 - 2)^2} = 4.29$
3	$\sqrt{(1.83 - 3)^2 + (2.33 - 4)^2} = 2.04$	$\sqrt{(4.12 - 3)^2 + (5.38 - 4)^2} = 1.78$
4	$\sqrt{(1.83 - 5)^2 + (2.33 - 7)^2} = 5.64$	$\sqrt{(4.12 - 5)^2 + (5.38 - 7)^2} = 1.84$
5	$\sqrt{(1.83 - 3.5)^2 + (2.33 - 5)^2} = 3.15$	$\sqrt{(4.12 - 3.5)^2 + (5.38 - 5)^2} = 0.73$
6	$\sqrt{(1.83 - 4.5)^2 + (2.33 - 5)^2} = 3.78$	$\sqrt{(4.12 - 4.5)^2 + (5.38 - 5)^2} = 0.54$
7	$\sqrt{(1.83 - 3.5)^2 + (2.33 - 4.5)^2} = 2.74$	$\sqrt{(4.12 - 3.5)^2 + (5.38 - 4.5)^2} = 1.08$

Therefore, the new clusters are:

{1,2} and {3,4,5,6,7}

$$\text{Group 1} = \left(\frac{1+1.5}{2} \right), \left(\frac{1+2}{2} \right) = (1.25, 1.5)$$

$$\text{Group 2} = \left(\frac{3+5+3.5+4.5+3.5}{5} \right), \left(\frac{4+7+5+5+4.5}{5} \right) = (3.9, 5.1)$$

Step 4:

	Centroid 1	Centroid 2
1	$\sqrt{(1.25 - 1)^2 + (1.5 - 1)^2} = 0.58$	$\sqrt{(3.9 - 1)^2 + (5.1 - 1)^2} = 5.02$
2	$\sqrt{(1.25 - 1.5)^2 + (1.5 - 2)^2} = 0.56$	$\sqrt{(3.9 - 1.5)^2 + (5.1 - 2)^2} = 3.92$
3	$\sqrt{(1.25 - 3)^2 + (1.5 - 4)^2} = 3.05$	$\sqrt{(3.9 - 3)^2 + (5.1 - 4)^2} = 1.42$
4	$\sqrt{(1.25 - 5)^2 + (1.5 - 7)^2} = 6.66$	$\sqrt{(3.9 - 5)^2 + (5.1 - 7)^2} = 2.20$
5	$\sqrt{(1.25 - 3.5)^2 + (1.5 - 5)^2} = 4.16$	$\sqrt{(3.9 - 3.5)^2 + (5.1 - 5)^2} = 0.41$
6	$\sqrt{(1.25 - 4.5)^2 + (1.5 - 5)^2} = 4.78$	$\sqrt{(3.9 - 4.5)^2 + (5.1 - 5)^2} = 0.61$
7	$\sqrt{(1.25 - 3.5)^2 + (1.5 - 4.5)^2} = 3.75$	$\sqrt{(3.9 - 3.5)^2 + (5.1 - 4.5)^2} = 0.72$

- ▶ Therefore, there is no change in the cluster.
- ▶ Thus, the algorithm comes to a halt here and final result consist of 2 clusters $\{1,2\}$ and $\{3,4,5,6,7\}$.

Pros and cons

Advantages of k-means

1. Relatively simple to implement.
2. Scales to large data sets.
3. Guarantees convergence.
4. Easily adapts to new examples.

Disadvantages of k-means

1. Choosing k manually.
2. Being dependent on initial values.
3. Scaling with number of dimensions.

Naive Base

Play Tennis Example

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Calculating Probabilities

- Learning Phase

Outlook	Play=Yes	Play=No
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=Yes	Play=No
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=Yes	Play=No
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=Yes	Play=No
Strong	3/9	3/5
Weak	6/9	2/5

$$P(\text{Play=Yes}) = 9/14 \quad P(\text{Play=No}) = 5/14$$

Test Phase

–Given a new instance, predict its label

$\mathbf{x}=(\text{Outlook}=\textit{Sunny}, \text{Temperature}=\textit{Cool}, \text{Humidity}=\textit{High}, \text{Wind}=\textit{Strong})$

–Look up tables achieved in the learning phase

$$P(\text{Outlook}=\textit{Sunny}|\text{Play}=\textit{Yes}) = 2/9$$

$$P(\text{Temperature}=\textit{Cool}|\text{Play}=\textit{Yes}) = 3/9$$

$$P(\text{Humidity}=\textit{High}|\text{Play}=\textit{Yes}) = 3/9$$

$$P(\text{Wind}=\textit{Strong}|\text{Play}=\textit{Yes}) = 3/9$$

$$P(\text{Play}=\textit{Yes}) = 9/14$$

$$P(\text{Outlook}=\textit{Sunny}|\text{Play}=\textit{No}) = 3/5$$

$$P(\text{Temperature}=\textit{Cool}|\text{Play}=\textit{No}) = 1/5$$

$$P(\text{Humidity}=\textit{High}|\text{Play}=\textit{No}) = 4/5$$

$$P(\text{Wind}=\textit{Strong}|\text{Play}=\textit{No}) = 3/5$$

$$P(\text{Play}=\textit{No}) = 5/14$$

$$P(\text{Yes}|\mathbf{x}') \approx [P(\textit{Sunny}|\textit{Yes})P(\textit{Cool}|\textit{Yes})P(\textit{High}|\textit{Yes})P(\textit{Strong}|\textit{Yes})]P(\text{Play}=\textit{Yes}) = 0.0053$$

$$P(\text{No}|\mathbf{x}') \approx [P(\textit{Sunny}|\textit{No})P(\textit{Cool}|\textit{No})P(\textit{High}|\textit{No})P(\textit{Strong}|\textit{No})]P(\text{Play}=\textit{No}) = 0.0206$$

Given the fact $P(\text{Yes}|\mathbf{x}') < P(\text{No}|\mathbf{x}')$, we label \mathbf{x}' to be “No”.

Outlook	Play=Yes	Play=No	Temperature	Play=Yes	Play=No
Sunny	2/9	3/5	Hot	2/9	2/5
Overcast	4/9	0/5	Mild	4/9	2/5
Rain	3/9	2/5	Cool	3/9	1/5

Humidity	Play=Yes	Play=No	Wind	Play=Yes	Play=No
High	3/9	4/5	Strong	3/9	3/5
Normal	6/9	1/5	Weak	6/9	2/5

$$P(\text{Play}=\textit{Yes}) = 9/14 \quad P(\text{Play}=\textit{No}) = 5/14$$

Pros and Cons

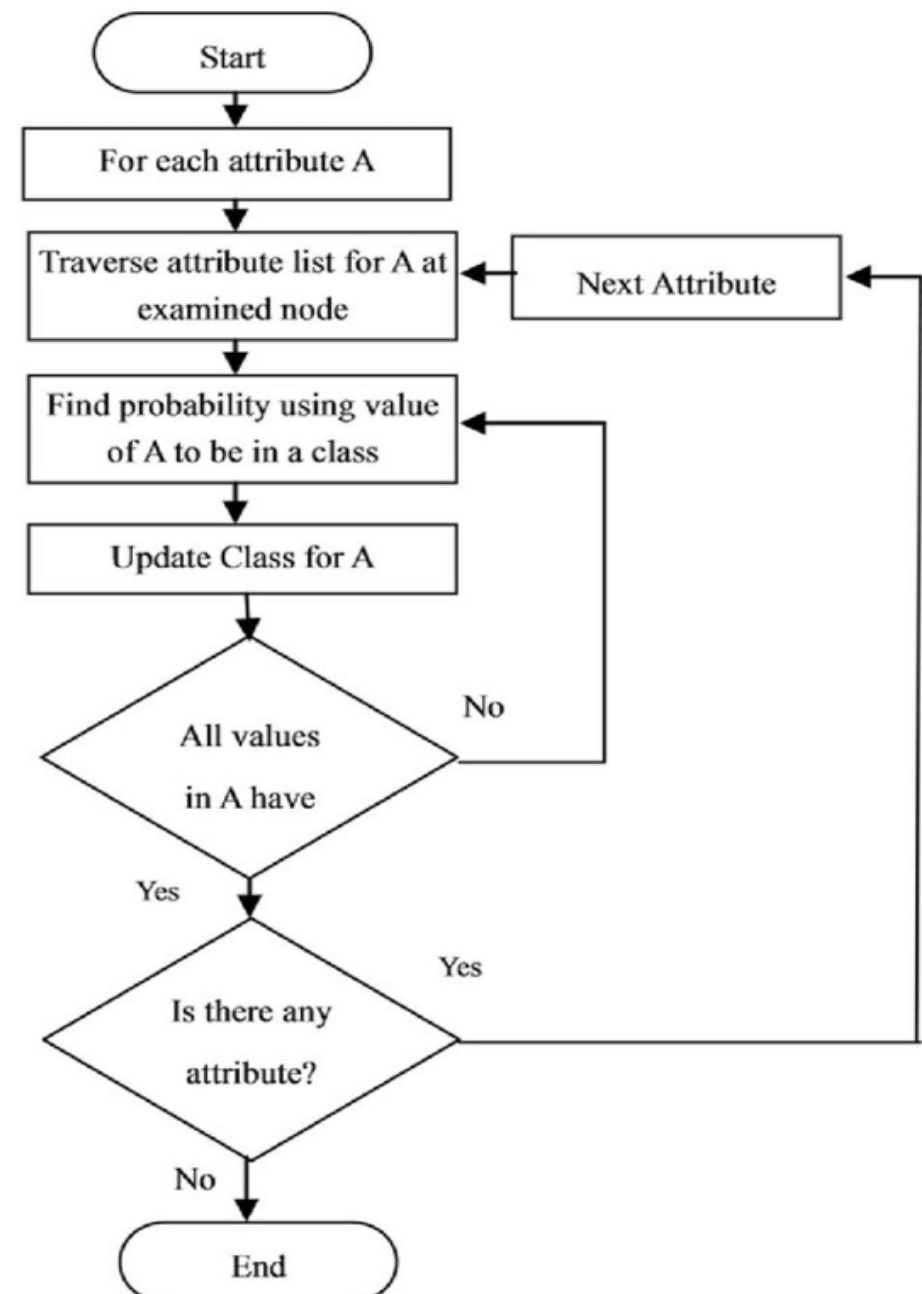
Advantages of Naive Bayes

1. This algorithm works very fast and can easily predict the class of a test dataset.
2. You can use it to solve multi-class prediction problems as it's quite useful with them.
3. Naive Bayes classifier performs better than other models with less training data if the assumption of independence of features holds.

Disadvantages of Naive Bayes

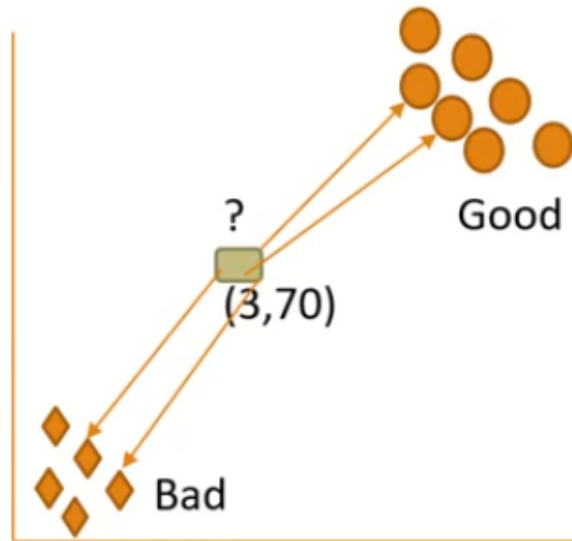
1. If your test data set has a categorical variable of a category that wasn't present in the training data set, the Naive Bayes model will assign it zero probability and won't be able to make any predictions in this regard.
2. It assumes that all the features are independent. While it might sound great in theory, in real life, you'll hardly find a set of independent features.

Flow chart



K-nearest Neighbor

Classification Using KNN

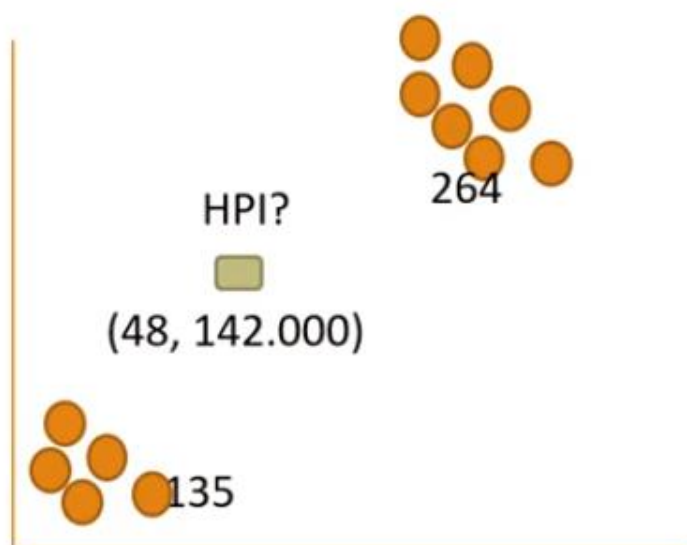


K=4

Name	Cigarettes	Weight	Heart Attack	Distance	
A	7	70	Bad	$\sqrt{(3-7)^2 + (70-70)^2}$ = 4	1
B	7	40	Bad	$\sqrt{(3-7)^2 + (70-40)^2}$ = 30.27	4
C	3	40	Good	$\sqrt{(3-3)^2 + (70-40)^2}$ = 30.00	2
D	1	40	Good	$\sqrt{(3-1)^2 + (70-40)^2}$ = 30.07	3
E	3	70	Bad		

Regression Using KNN

قيم عددية



Age	Loan	House Price Index	Distance
25	\$40,000	135	102000
35	\$60,000	256	82000
45	\$80,000	231	62000
20	\$20,000	267	122000
35	\$120,000	139	22000
52	\$18,000	150	124000
23	\$95,000	127	47000
40	\$62,000	216	80000
60	\$100,000	139	42000
48	\$220,000	250	78000
33	\$150,000	264	8000
48	\$142,000	?	

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Explanation

K=1

- using the training set to classify an unknown case
- (Age=33 and Loan=\$150,000) {Euclidean distance}.
- If K=1 then the nearest neighbor is the last case in the set
- with HPI=264.
- $D = \text{Sqrt}[(48-33)^2 + (142000-150000)^2] = 8000.01 \gg \text{HPI} = 264$

K=3

- the prediction for HPI is equal to the average of HPI for the top three neighbors
 - $\text{HPI} = (264+139+139)/3 = 180.7$
- TRY when K =4? What is HPI for it ?

Age	Loan	House Price Index	Distance
25	\$40,000	135	102000
35	\$60,000	256	82000
45	\$80,000	231	62000
20	\$20,000	267	122000
35	\$120,000	139	22000
52	\$18,000	150	124000
23	\$95,000	127	47000
40	\$62,000	216	80000
60	\$100,000	139	42000
48	\$220,000	250	78000
33	\$150,000	264	8000
48	\$142,000	?	

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

KNN pros and cons

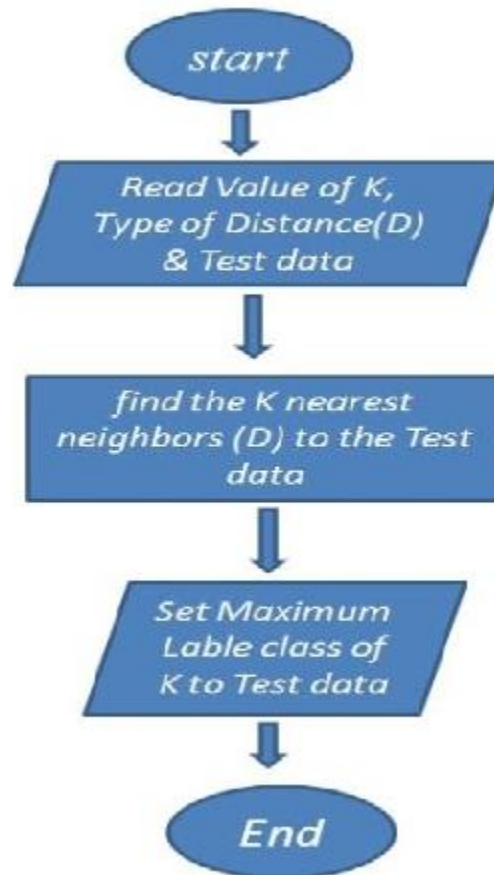
Pros

1. No Training Period: KNN is called Lazy Learner (Instance based learning).
2. new data can be added seamlessly which will not impact the accuracy of the algorithm.
3. 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.)

Cons

1. Does not work well with large dataset:
2. Does not work well with high dimensions
3. Need feature scaling: We need to do feature scaling (standardization and normalization)
4. Sensitive to noisy data, missing values.

Flow chart for KNN



Decision Tree pros and cons

Advantages:

1. Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.
2. A decision tree does not require normalization of data.
3. A decision tree does not require scaling of data as well.
4. Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.
5. A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.

Disadvantage:

1. A small change in the data can cause a large change in the structure of the decision tree causing instability.
2. For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
3. Decision tree often involves higher time to train the model.
4. Decision tree training is relatively expensive as the complexity and time has taken are more.
5. The Decision Tree algorithm is inadequate for applying regression and predicting continuous values.