Machine Learning Algorithms

Naive Bayes
KNN

AHMED YOUSRY

Naive Bayes

Play Tenr

Day	Outlook	Temp	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Strong	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

yes=9
No = 5

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

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

Day	Outlook	Temp	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Strong	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Test Phase

–Given a new instance, predict its label

x=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)

–Look up tables achieved in the learning phrase

P(Outlook=*Sunny*|Play=*Yes*) = 2/9

P(Temperature=*Cool*|Play=*Yes*) = 3/9

P(Huminity=*High*|Play=*Yes*) = 3/9

P(Wind=*Strong*|Play=*Yes*) = 3/9

P(Play=*Yes*) = 9/14

P(Outlook=Sunny|Play=No) = 3/5

P(Temperature=Cool|Play==No) = 1/5

P(Huminity=High|Play=No) = 4/5

P(Wind=Strong|Play=No) = 3/5

P(Play=No) = 5/14

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

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

P(*Yes*|**x**') ≈ [P(*Sunny*|Yes)P(*Cool*|*Yes*)P(*High*|Yes)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053

P(*No*|**x**') ≈ [P(*Sunny*|No) P(*Cool*|No)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

Given the fact P(*Yes*|**x**') < P(*No*|**x**'), we label **x**' to be **"*No*".**

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.

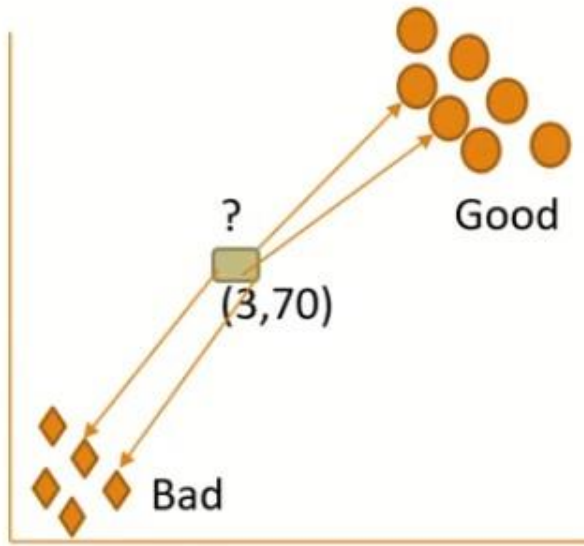
Disadvantages of Naive Bayes

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.

K-nearest Neighbor

Classification Using KNN

تصنيف الى فئات



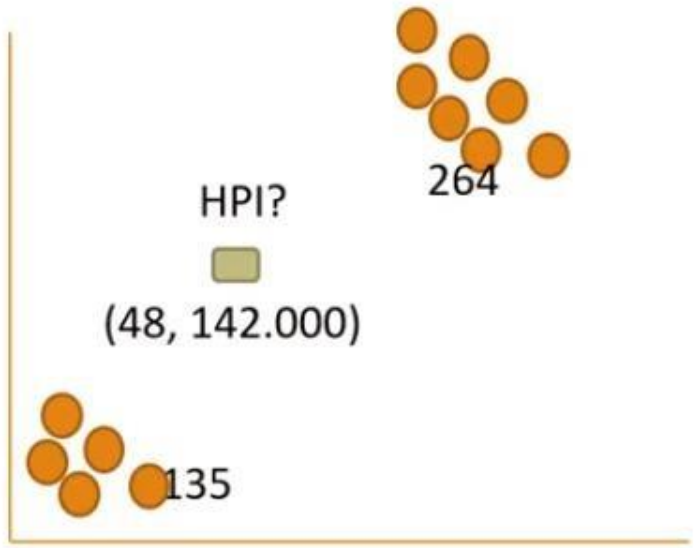
K=4 k=1

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

K=3

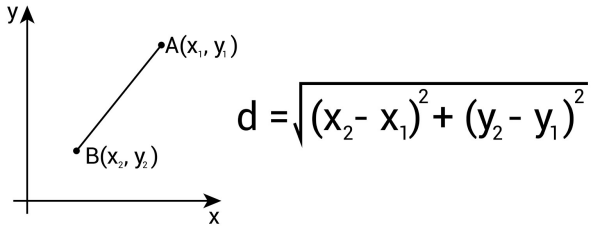
Regression Using KNN

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	2
52	$18,000	150	124000	
23	$95,000	127	47000	
40	$62,000	216	80000	
60	$100,000	139	42000	3
48	$220,000	250	78000	
33	$150,000	264	8000	1
48	$142,000	?		

Distance Formula



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 = Sqrt[(48-33)^2 + (142000-150000)^2] = 8000.01 >> HPI = 264

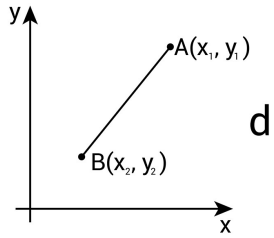
K=3

- the prediction for HPI is equal to the average of HPI for the top three neighbors
- 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	2
52	$18,000	150	124000	
23	$95,000	127	47000	
40	$62,000	216	80000	
60	$100,000	139	42000	3
48	$220,000	250	78000	
33	$150,000	264	8000	1
48	$142,000	?		

Distance Formula



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

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.