DATA ANALYTICS- UNIT 2

By

DEEPIKA KAMBOJ

Bayesian Modelling

Bayesian modelling is a statistical approach that uses Bayes' theorem to model uncertainty in data, and involves the following steps:

- Specifying prior distributions
- Incorporating data through the likelihood function
- Updating prior distributions using Bayes' theorem
- Making inferences and predictions

Bayesian Modelling

- In simple terms, Bayesian modelling is like a game of guessing.
- You make an initial guess about something, and then you collect some data that might help you to refine your guess.
- Based on the data, you update your guess and make a new, more informed guess.

Bayesian Modelling

- In Bayesian modelling, we use probability distributions to represent our beliefs or uncertainties about the parameters of a model.
- We update these probability distributions based on new data, using Bayes' theorem to compute the posterior probability distribution.
- This allows us to make predictions and estimates based on probabilistic inference.

Bayes Theorem

- Bayes' theorem is a mathematical formula that describes the probability of an event based on prior knowledge of conditions that might be related to the event.

The theorem can be written as:

$P(A|B) = P(B|A) \times P(A) / P(B)$

Where:

- $P(A|B)$ is the probability of event A occurring given that event B has occurred.
- $P(B|A)$ is the probability of event B occurring given that event A has occurred.
- $P(A)$ is the prior probability of event A occurring.
- $P(B)$ is the prior probability of event B occurring.

Bayes Theorem Example 1

- Suppose you have developed a new diagnostic test for a rare disease that affects 1 in 1000 people.
- The test is 95% accurate
- You administer the test to a patient who tests positive for the disease. What is the probability that the patient has the disease?

Solution

- P(D) = 0.001
- P(~D) = 0.999
- P(Pos|D) = 0.95
- P(Neg|~D) = 0.95

Solve following:

P(D|Pos) = P(Pos|D) * P(D) / [P(Pos|D) * P(D) + P(Pos|~D) * P(~D)]

Bayes Theorem Example 2

- Suppose a factory produces two types of products: Type A and Type B.
- Historically, 70% of the products produced are Type A and 30% are Type B.
- The factory has two machines that produce these products, Machine 1, and Machine 2.
- Machine 1 produces 95% Type A products and 5% Type B products, while Machine 2 produces 80% Type A products and 20% Type B products.

Question:

One of the products is randomly selected from the factory and it is found to be a Type A product. What is the probability that it was produced by Machine 1?

Bayes Theorem Example 2

- P(M1) = 0.5
- P(M2) = 0.5
- P(A|M1) = 0.95
- P(A|M2) = 0.8

Solve following:

P(M1|A) = P(A|M1) * P(M1) / [P(A|M1) * P(M1) + P(A|M2) * P(M2)]

Naïve Bayes vs Bayes Theorem

- Bayes' theorem is a mathematical formula that describes the relationship between conditional probabilities of two events. It allows us to update our beliefs or probabilities about the occurrence of an event based on new evidence or information.

- Naive Bayes theorem, on the other hand, is a specific application of Bayes' theorem to solve classification problems in machine learning. In Naive Bayes, we make a naive assumption that the features (or attributes) of the data are independent of each other, which simplifies the calculation of the posterior probability distribution. This makes the algorithm computationally efficient and often leads to good results in practice.

Naïve Bayes Example

Fruit	Color	Diameter	Type
1	Red	3	Apple
2	Red	3	Apple
3	Red	1	Orange
4	Yellow	1	Orange
5	Yellow	2	Orange
6	Yellow	3	Apple

Naïve Bayes Example

Question:

Predict whether a new fruit is an apple, or an orange based on its color and diameter.

Solution:

P (Apple | Color = Red, Diameter = 2) = P (Color = Red | Apple) * P (Diameter = 2 | Apple) * P(Apple) / P (Color = Red, Diameter = 2)

KNN Algorithm

- k-Nearest Neighbours is a non-parametric machine learning algorithm.
- Used for classification and regression tasks.
- The k-NN algorithm works by finding the k-nearest neighbours of a new data point and assigning it to the class that occurs most frequently among those neighbours.
- The value of k is a hyper parameter that needs to be set prior to training the algorithm.

KNN Algorithm Advantages

1. k-NN is a simple and easy-to-understand algorithm.

2. It does not require any assumptions about the underlying distribution of the data.

2. It can be used for both classification and regression tasks.

3. K-NN is a non-parametric algorithm, which means it does not make any assumptions about the functional form of the data.

KNN Algorithm Disadvantages

1. The main drawback of K-NN is its computational complexity, especially when dealing with large datasets.

2. The choice of the value of k can have a significant impact on the performance of the algorithm.

3. The k-NN algorithm is sensitive to irrelevant features in the dataset.

KNN Example

Flower	Petal length	Petal width	Label
1	1.4	0.2	setosa
2	1.3	0.2	setosa
3	4.7	1.4	versicolor
4	4.9	1.5	versicolor
5	5.1	1.9	virginica
6	5.7	2.3	virginica

KNN Example

Now, let's say we have a new flower with petal length 5.0 and petal width 1.6, and we want to classify it based on its features using k-NN algorithm with k=3.

KNN Example

Flower	Petal length	Petal width	Distance
1	1.4	0.2	sqrt((5.0-1.4)^2+(1.6-0.2)^2)
2	1.3	0.2	sqrt((5.0-1.3)^2+(1.6-0.2)^2)
3	4.7	1.4	sqrt((5.0-4.7)^2+(1.6-1.4)^2)
4	4.9	1.5	sqrt((5.0-4.9)^2+(1.6-1.5)^2)
5	5.1	1.9	sqrt((5.0-5.1)^2+(1.6-1.9)^2)
6	5.7	2.3	sqrt((5.0-5.7)^2+(1.6-2.3)^2)

Finally, we classify the new flower by taking a majority vote among the labels of the k nearest neighbors.

Distances

1. Euclidean Distance
2. Manhattan Distance
3. Minkowski Distance

Euclidean Distances

$$D(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2)}$$

Manhattan Distances

$$D(x, y) = \sum_{i=1}^{k} |x_i - y_i|$$

Minkowski Distances

$$D(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$