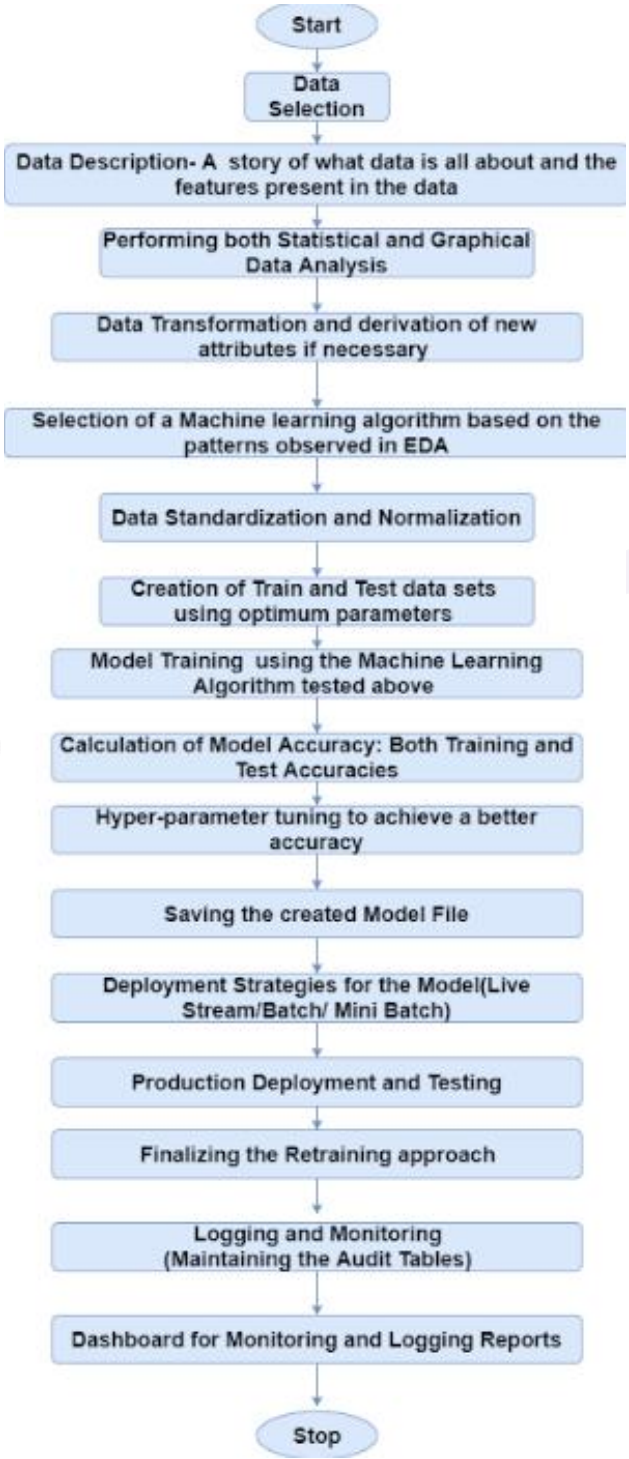
PW SKILLS

Lesson Plan

Naive Bayes

Application Flow

Before proceeding with the algorithm, let's first discuss the lifecycle of any machine learning model. This diagram explains the creation of a Machine Learning model from scratch and then taking the same model further with hyperparameter tuning to increase its accuracy, deciding the deployment strategies for that model and once deployed setting up the logging and monitoring frameworks to generate reports and dashboards based on the client requirements. A typical lifecycle diagram for a machine learning model looks like:



Bayes's Theorem

According to Wikipedia, In probability theory and statistics,* Bayes's theorem** (alternatively *Bayes's law or Bayes's rule) describes the probability of an event, based on prior knowledge of conditions that might be related to the event. Mathematically, it can be written as:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Where A and B are events and P(B)≠0

- P(A|B) is a conditional probability: the likelihood of event A occurring given that B is true.
- P(B|A) is also a conditional probability: the likelihood of event B occurring given that A is true.
- P(A) and P(B) are the probabilities of observing A and B respectively; they are known as the marginal probability.

Why it is called Naive Bayes?

The entire algorithm is based on Bayes's theorem to calculate probability. So, it also carries forward the assumptions for Bayes's theorem. However, those assumptions(that the features are independent) might not always be true when implemented over a real-world dataset. So, those assumptions are considered Naïve hence the name.

Let's understand it with the help of an example:

- The problem statement:
- There are two machines which manufacture bulbs. Machine 1 produces 30 bulbs per hour and machine 2 produce 20 bulbs per hour. Out of all bulbs produced, 1 % turn out to be defective. Out of all the defective bulbs, the share of each machine is 50%. What is the probability that a bulb produced by machine 2 is defective?
- We can write the information given above in mathematical terms as:
- The probability that a bulb was made by Machine 1, P(M1)=30/50=0.6
- The probability that a bulb was made by Machine 2, P(M2)=20/50=0.4
- The probability that a bulb is defective, P(Defective)=1%=0.01
- The probability that a defective bulb came out of Machine 1, P(M1 | Defective)=50%=0.5
- The probability that a defective bulb came out of Machine 2, P(M2 | Defective)=50%=0.5
- Now, we need to calculate the probability of a bulb produced by machine 2 is defective i.e., P(Defective | M2). Using the Bayes Theorem above, it can be written as:

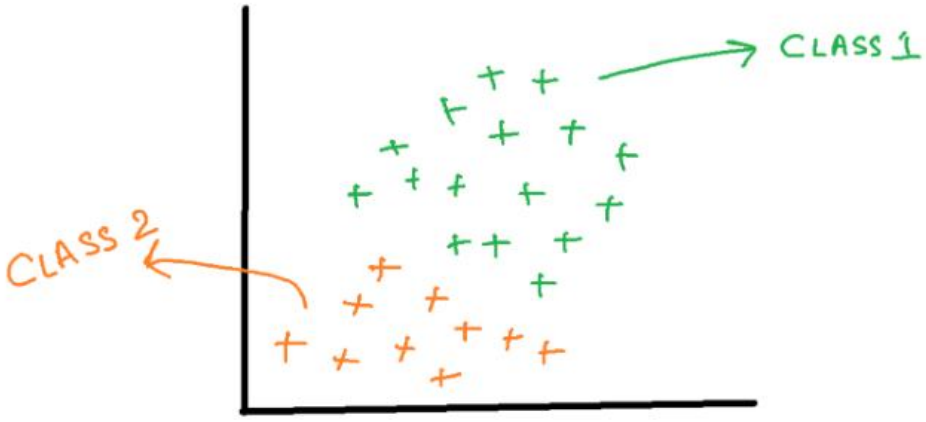
$$P(Defective|M2) = \frac{P(M2|Defective)*P(Defective)}{P(M2)}$$

Substituting the values, we get: $P(Defective|M2) = \frac{0.5*0.01}{0.4} = 0.0125$

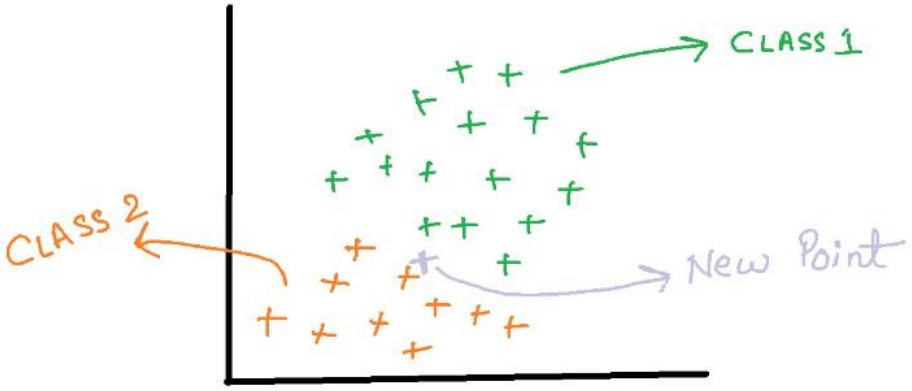
Task for you is to calculate the probability that a bulb produced by machine 1 is defective.

Algorithm steps:

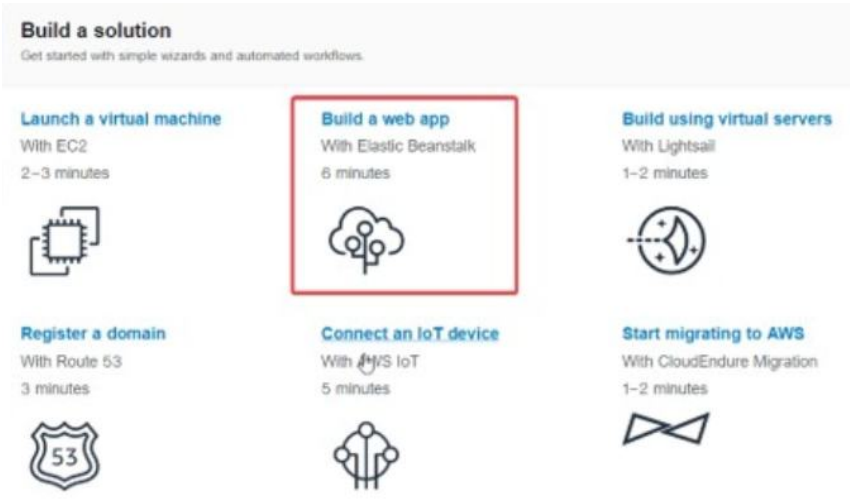
1. Let's consider that we have a binary classification problem i.e., we have two classes in our data as shown below.



2. Now suppose if we are given with a new data point, to which class does that point belong to?



3. The formula for a point 'X' to belong in class1 can be written as:

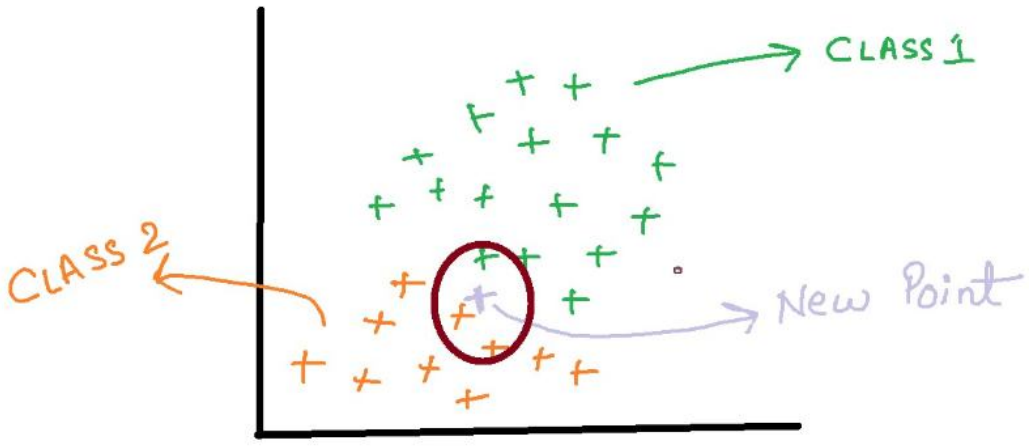


Where the numbers represent the order in which we are going to calculate different probabilities.

4. A similar formula can be utilised for class 2 as well.
5. The probability of class 1 can be written as:

$$P(class1) = \frac{Numberofpointsinclass1}{Totalnumberofpoints} = \frac{16}{26} = 0.62$$

6. For calculating the probability of X, we draw a circle around the new point and see how many points(excluding the new point) lie inside that circle.



The points inside the circle are considered to be similar points.

$$P(X) = \frac{Numberofsimilarobservation}{TotalObservations} = \frac{3}{26} = 0.12$$

The points inside the circle are considered to be similar points.

$$P(X) = \frac{Numberofsimilarobservation}{TotalObservations} = \frac{3}{26} = 0.12$$

7. Now, we need to calculate the probability of a point to be in the circle that we have made given that it's of class 1. $P(X|Class1) = \frac{Numberofpointsinclass1insidethecircle}{Totalnumberofpointsinclass1} = \frac{1}{16} = 0.06$
8. We can substitute all the values into the formula in step 3. We get:
   $P(Class1|X) = \frac{0.06*0.62}{0.12} = 0.31$
9. And if we calculate the probability that X belongs to Class2, we'll get 0.69. It means that our point belongs to class 2.

The Generalization for Multiclass:

The approach discussed above can be generalised for multiclass problems as well. Suppose, P1, P2, P3...Pn are the probabilities for the classes C1,C2,C3...Cn, then the point X will belong to the class for which the probability is maximum. Or mathematically the point belongs to the result of :

$$argmax(P1, P2, P3 \ldots . Pn)$$

The Difference

You can notice a major difference in the way in which the Naïve Bayes algorithm works form other classification algorithms. It does not first try to learn how to classify the points. It directly uses the label to identify the two separate classes and then it predicts the class to which the new point shall belong.

Multinomial Naive Bayes for Discrete Data

- This method excels at handling data with features represented by discrete counts, frequencies, or occurrences. Imagine features like word counts in a document or the number of times a specific product category is purchased by a customer.
- It relies on the multinomial distribution, which models the probability of observing a specific number of events (e.g., word counts) from a fixed number of categories (e.g., unique words in a vocabulary).

Training Process:

1. Feature Representation: Features are typically converted into counts or frequencies. For example, a document might be represented by a vector where each element indicates the number of times a particular word appears in the document.
2. Class-Conditional Probabilities: The algorithm estimates the probability of each feature value (e.g., word count) occurring given a specific class (e.g., document category). For instance, it calculates the probability of the word "computer" appearing once in a document classified as "technology" compared to a document classified as "sports."

Prediction:

1. New Data Point: Given a new unseen data point (e.g., a new document), the algorithm calculates the probability of each class based on the individual feature probabilities using Bayes' theorem.
2. Class Assignment: The class with the highest probability is assigned to the data point.

Advantages of Multinomial Naive Bayes:

1. Efficiency: Similar to Gaussian Naive Bayes, it's computationally efficient for both training and prediction.
2. Probabilistic Output: It provides class probabilities, offering insights into the model's confidence in its predictions.
3. Simplicity: The model is relatively easy to understand and implement due to its reliance on straightforward calculations.
4. Handles Sparse Data: It can work effectively even with sparse data, where features might have many zeros (e.g., a document with many unique words).

Disadvantages of Multinomial Naive Bayes:

1. Independence Assumption: Like Gaussian Naive Bayes, it assumes that features are independent, which might not always hold true in real-world scenarios (e.g., word order can be important in text).
2. Feature Engineering: Feature selection and representation can significantly impact performance. Choosing the right features and converting them into meaningful counts is crucial.

Gaussian Naive Bayes

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. Go back to the normal distribution lecture to review the formulas for the Gaussian/Normal Distribution.

For example, using the Gaussian Distribution, suppose the training data contain a continuous attribute, x. We first segment the data by the class and then compute the mean and variance of x in each class. Let μc be the mean of the values in x associated with class c, and let σ2c be the variance of the values in x associated with class c. Then, the probability distribution of some value given a class, p(x=v|c), can be computed by plugging v into the equation for a Normal distribution parameterized by μc and σ2c. That is:

$$p(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

Advantages:

- Naive Bayes is extremely fast for both training and prediction as they not have to learn to create separate classes.
- Naive Bayes provides a direct probabilistic prediction.
- Naive Bayes is often easy to interpret.
- Naive Bayes has fewer (if any) parameters to tune

Disadvantages:

- The algorithm assumes that the features are independent which is not always the scenario
- Zero Frequency i.e. if the category of any categorical variable is not seen in the training data set even once then the model assigns a zero probability to that category and then a prediction cannot be made.

Various Types Of Bayes Theorem And Its Intuition

Bayes' theorem is a fundamental concept in probability that allows you to update your beliefs about something (event A) based on the presence of another event (event B). It essentially helps you calculate the posterior probability, which is the probability of A being true given that you know B is true.
There are various types of Bayes' theorem depending on the nature of the events involved. Here's a breakdown of some common ones:

1. **Classic (or Simple) Bayes' Theorem:**

This is the most general form and applies to any two events A and B. It's represented by the following equation:

Formula: P(A | B) = (P(B | A) * P(A)) / P(B)

- **P(A | B):** Posterior probability - The probability of event A being true given that event B is true (what we want to find).
- **P(B | A):** Likelihood - The probability of event B happening given that event A is true (how likely B is to occur if A is true).
- **P(A):** Prior probability - The initial probability of event A happening before considering any evidence (our initial belief about A).
- **P(B):** Marginal probability - The probability of event B happening regardless of A (the overall probability of B).

Intuition:
Imagine you suspect your friend has a cold (A) because they're sneezing (B). Bayes' theorem helps you calculate the likelihood of them actually having a cold (A) after considering the fact that they're sneezing (B). It takes into account your initial belief about them having a cold (prior probability) and how often people sneeze when they have a cold (likelihood) to give you a more informed estimate.

2. **Naive Bayes Classifiers:**

This is a family of classification algorithms based on Bayes' theorem that assumes features (data points) are independent of each other. They are further categorized based on the data type they handle:

- Multinomial Naive Bayes: Used for discrete data like word counts in documents (e.g., spam filtering).
- Gaussian Naive Bayes: Used for continuous data like sensor readings or measurements (e.g., image classification).

3. **Bayesian Networks:**

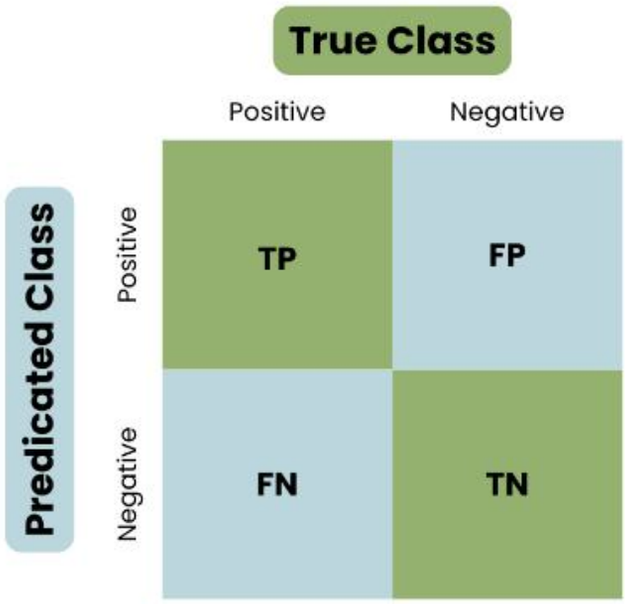
These are graphical models that represent relationships between variables using directed acyclic graphs (DAGs). They use Bayes' theorem to calculate the probability of a specific event occurring given the state of other variables in the network. These are powerful for modeling complex relationships between variables.

Intuition:
Imagine a network representing the factors affecting a car not starting (event A): a dead battery (B), faulty alternator (C), and leaving the lights on (D). Bayes' theorem within this network helps calculate the probability of a dead battery (A) given information about the alternator (C) and lights (D).

Confusion Matrix

- It's a table that summarizes the performance of a classification model on a set of test data.
- It allows us to visualize how many predictions the model got right and wrong for each class.

Structure of a Confusion Matrix:
Imagine a two-class classification problem (e.g., predicting spam/not spam emails). The confusion matrix would look like this:



1. **TP (True Positive):** These are the cases where the model correctly predicted a positive outcome.
2. **TN (True Negative):** These are the cases where the model correctly predicted a negative outcome.
3. **FP (False Positive):** These are the cases where the model incorrectly predicted a positive outcome (Type I error).
4. **FN (False Negative):** These are the cases where the model incorrectly predicted a negative outcome (Type II error).

Interpreting a Confusion Matrix:

By analyzing the counts in each cell of the matrix, we can gain valuable insights into the model's performance:

1. **Accuracy:** The overall percentage of correct predictions can be calculated as (TP + TN) / Total Cases.
2. **Precision:** How often was a positive prediction actually correct? (TP / (TP + FP))
3. **Recall:** How often did the model identify true positives? (TP / (TP + FN))
4. **Specificity:** How often did the model identify true negatives? (TN / (TN + FP))

Confusion Matrix vs. Accuracy:

- While accuracy is a simple metric, it can be misleading if the dataset is imbalanced (e.g., many more negative examples than positive).
- A confusion matrix provides a more detailed breakdown, allowing you to identify areas for improvement even if the overall accuracy seems good.

Example:

Imagine a spam filter with a high accuracy (90%). However, the confusion matrix reveals a high number of false negatives (important emails classified as spam). This suggests the filter might be too aggressive, and adjustments are needed to avoid missing important emails.