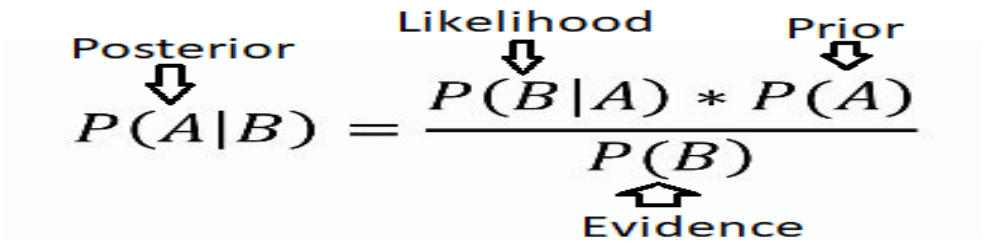
NAIVE BAIYES AND KNN (K-NEAREST NEIGHBORS)

NAÏVE BAYES

- **What is Naive Bayes Algorithm?**
- The Naive Bayes Algorithm is a Machine Learning Algorithm for Classification problems
- The Naive Bayes Theorem is based on the Bayes' Theorem. Bayes' Theorem describes the probability of an event based on prior knowledge of the conditions related to the event.
- **Bayes Theorem:-**



- Where
- **P(A|B)** is the probability of hypothesis A given that evidence B is true. This is called the posterior probability.
- **P(B|A)** is the probability of evidence B given that the hypothesis A was true.
- **P(A)** is the probability of hypothesis A being true . This is called the prior probability of A.
- **P(B)** is the probability of the evidence (regardless of the hypothesis).
- The marginal likelihood, also known as the evidence

- **Note:-**Please note that P(A) or P(B) is also called **class probability** and P(A| B) is called **conditional probability**.

- **Assumption: -**
- Assuming all the feature are independent and are equally important and predicting the things based on prior knowledge and these independence assumptions.
- Another assumption made here is that all the predictors have an equal effect on the outcome.

- **Why is it called Naive?**
- The reason for calling the algorithm 'naive' is because its assumptions, which may or may not be correct and also occurrence of certain features is independent of the occurrence of other features.

- **Naive Bayes Classifier**
- It is a kind of classifier that works on Bayes theorem. Prediction of membership probabilities is made for every class such as the probability of data points associated to a particular class. The class having maximum probability is appraised as the most suitable class. This is also referred as Maximum A Posteriori (MAP).
- The MAP for a hypothesis is:
- MAP (A) = max P((A|B))
- MAP (A) = max P((A|B) $*$ (P(A)) /P(B))
- MAP (A) = max(P(B|A) $*$ P(A))
- P (B) is evidence probability, and it is used to normalize the result. Result will not be affected by removing P(B).
- NB classifiers conclude that all the variables or features are not related to each other.
- Existence or absence of a variable does not impact the existence or absence of any other variable.
- Example:

NAIVE BAIYES AND KNN (K-NEAREST NEIGHBORS)

- A fruit may be observed to be an apple if it is red, round, and about 4" in diameter.
- In this case also even if all the features are interrelated to each other, a NB classifier will observe all of these independently contributing to the probability that the fruit is apple.
- We experiment the hypothesis in real datasets, given multiple features.
- So, computation becomes complex.

- **Explain Naive Bayes Classifier?**
- There are three naïve Bayes classifiers:
- The **Multinomial** classifier uses multinomial distribution on each word of a sentence. Every word is treated independently rather than being treated as a part of the sentence. It is used for discrete counts.
- The **Gaussian** classifier is utilized with continuous data. It assumes that each data class is distributed as a Gaussian distribution. It is used in classification and it assumes that **features follow a normal distribution**.
- The **Bernoulli** classifier assumes that every feature present is binary, which means it can only take either of the two values.

- **Applications of Naive Bayes Algorithms: -**
- Real time Prediction (Means Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.)
- Multi class Prediction (Means This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.)
- **Text classification/ Spam Filtering/ Sentiment Analysis**
- **Recommendation System**

- **Explain The Bayes' Box: -**
- The Bayes' box is a method of representing and solving probability through Bayes theorem.

Hypothesis	Prior	Likelihood	Likelihood x Prior	Posterior
A	0.75	1	0.75	0.857
B	0.25	0.5	0.125	0.143
Total			0.875	1

- The **prior** probabilities are assumed values without additional factors.
- The **likelihood** is nothing but the probability of A and B.
- The **posterior** probabilities are results after considering added information. (For instance, rain in the above example).

- **Zero Probability Problem: -**
- If the posterior probability will be zero, and the model is unable to make a prediction. This problem is known as Zero Probability because the occurrence of the particular class is zero.
- The solution for such an issue is the **Laplacian correction or Laplace Transformation**. Laplacian correction is one of the smoothing techniques. Here, you can assume that the dataset is large enough that adding one row of each class will not make a difference in the estimated probability. This will overcome the issue of probability values to zero.
- **For Example:** Suppose that for the class loan risky, there are 1000 training tuples in the database. In this database, income column has 0 tuples for low income, 990 tuples for medium income, and 10 tuples for high income. The probabilities of these events, without the Laplacian correction, are 0, 0.990 (from 990/1000), and 0.010 (from 10/1000)
  Now, apply Laplacian correction on the given dataset. Let's add 1 more tuple for each income-value pair. The probabilities of these events:

$$\frac{1}{1003} = 0.001, \frac{991}{1003} = 0.988, \text{ and } \frac{11}{1003} = 0.011,$$

NAIVE BAIYES AND KNN (K-NEAREST NEIGHBORS)

- **Advantages**
- It is not only a simple approach but also a fast and accurate method for prediction.
- Naive Bayes has very low computation cost.
- It can efficiently work on a large dataset.
- It performs well in case of discrete response variable compared to the continuous variable.
- It can be used with multiple class prediction problems.
- It also performs well in the case of text analytics problems.
- When the assumption of independence holds, a Naive Bayes classifier performs better compared to other models like logistic regression.

- **Disadvantages**
- The assumption of independent features. In practice, it is almost impossible that model will get a set of predictors which are entirely independent.
- If there is no training tuple of a particular class, this causes zero posterior probability. In this case, the model is unable to make predictions. This problem is known as Zero Probability/Frequency Problem.

K-NEAREST NEIGHBOR

- **What is K-Nearest Neighbor?**
- KNN is a non-parametric and lazy learning algorithm. Non-parametric means there is no assumption for underlying data distribution. In other words, the model structure determined from the dataset. This will be very helpful in practice where most of the real world datasets do not follow mathematical theoretical assumptions. Lazy algorithm means it does not need any training data points for model generation. All training data used in the testing phase. This makes training faster and testing phase slower and costlier. Costly testing phase means time and memory.

- **How does the KNN algorithm work?**
- In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor.
- For finding closest similar points, you find the distance between points using distance measures such as Euclidean distance, Hamming distance, Manhattan distance and Minkowski distance.
- KNN has the following basic steps:
- Calculate distance
- Find closest neighbors
- Vote for labels

- **Eager Vs Lazy learners: -**
- **Eager learners** mean when given training points will construct a generalized model before performing prediction on given new points to classify. You can think of such learners as being ready, active and eager to classify unobserved data points.
- **Lazy Learning** means there is no need for learning or training of the model and all of the data points used at the time of prediction. Lazy learners wait until the last minute before classifying any data point. Lazy learner stores merely the training dataset and waits until classification needs to perform. lazy learners do less work in the training phase and more work in the testing phase to make a classification. Lazy learners are also known as instance-based learners because lazy learners store the training points or instances, and all learning is based on instances.

- **How do you decide the number of neighbors in KNN?**

NAIVE BAIYES AND KNN (K-NEAREST NEIGHBORS)

- The number of neighbors(K) in KNN is a hyperparameter that you need choose at the time of model building. You can think of K as a controlling variable for the prediction model.
- Research has shown that no optimal number of neighbors suits all kind of data sets. Each dataset has it's own requirements. In the case of a small number of neighbors, the noise will have a higher influence on the result, and a large number of neighbors make it computationally expensive. Research has also shown that a small amount of neighbors are most flexible fit which will have low bias but high variance and a large number of neighbors will have a smoother decision boundary which means lower variance but higher bias.
- Generally, Data scientists choose as an odd number if the number of classes is even. You can also check by generating the model on different values of k and check their performance. You can also try Elbow method here.

- **Curse of Dimensionality: -**
- KNN performs better with a lower number of features than a large number of features. You can say that when the number of features increases than it requires more data. Increase in dimension also leads to the problem of overfitting. To avoid overfitting, the needed data will need to grow exponentially as you increase the number of dimensions. This problem of higher dimension is known as the Curse of Dimensionality.
- To deal with the problem of the curse of dimensionality, you need to perform principal component analysis before applying any machine learning algorithm, or you can also use feature selection approach. Research has shown that in large dimension Euclidean distance is not useful anymore. Therefore, you can prefer other measures such as cosine similarity, which get decidedly less affected by high dimension.

- **Assumptions of KNN: -**
- **Standardization: -** if one variable is based on height in cms, and the other is based on weight in kgs then height will influence more on the distance calculation. In order to make them comparable we need to standardize them
- **Outlier: -** Low k-value is sensitive to outliers and a higher K-value is more flexible to outliers as it considers more voters to decide prediction.

- **Advantage of KNN: -**
- The training phase of K-nearest neighbor classification is much faster compared to other classification algorithms. There is no need to train a model for generalization, That is why KNN is known as the simple and instance-based learning algorithm. KNN can be useful in case of nonlinear data. It can be used with the regression problem. Output value for the object is computed by the average of k closest neighbors value.

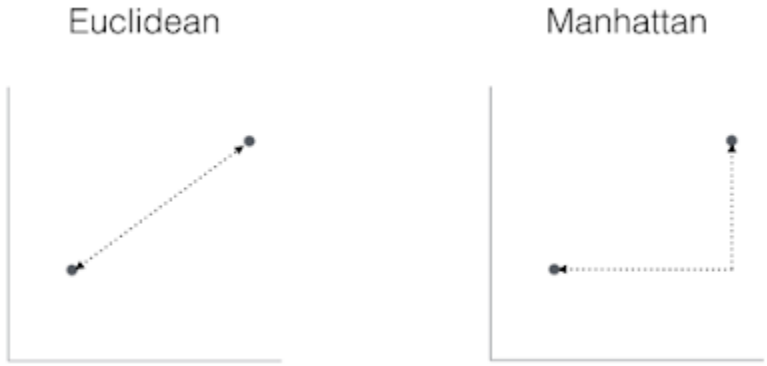
- **Disadvantage of KNN: -**
- The testing phase of K-nearest neighbor classification is slower and costlier in terms of time and memory. It requires large memory for storing the entire training dataset for prediction. KNN requires scaling of data because KNN uses the Euclidean distance between two data points to find nearest neighbors. Euclidean distance is sensitive to magnitudes. The features with high magnitudes will weight more than features with low magnitudes. KNN also not suitable for large dimensional data.

- **How to improve KNN performance?**
- For better results, normalizing data on the same scale is highly recommended
- KNN is not suitable for the large dimensional data. In such cases, dimension needs to reduce to improve the performance. Also, handling missing values will help us in improving results.

- **Why is the odd value of "K" preferable in KNN algorithm?**
- K should be odd so that there are no ties in the voting. If square root of number of data points is even, then add or subtract 1 to it to make it odd.

NAIVE BAIYES AND KNN (K-NEAREST NEIGHBORS)

- **What is the difference between Euclidean Distance and Manhattan distance? What is the formula of Euclidean distance and Manhattan distance?**
- Both are used to find out the distance between two points.



Euclidean: Take the square root of the sum of the squares of the differences of the coordinates.

For example, if $x = (a, b)$ and $y = (c, d)$, the Euclidean distance between x and y is $\sqrt{(a-c)^2 + (b-d)^2}$.

Manhattan: Take the sum of the absolute values of the differences of the coordinates.

For example, if $x = (a, b)$ and $y = (c, d)$, the Manhattan distance between x and y is $|a-c| + |b-d|$.

- **Can KNN be used for regression?**
- Yes, K-nearest neighbor can be used for regression. In other words, K-nearest neighbor algorithm can be applied when dependent variable is continuous. In this case, the predicted value is the average of the values of its k nearest neighbors.

- **Why should we not use KNN algorithm for large datasets?**
- KNN works well with smaller dataset because it is a lazy learner. It needs to store all the data and then makes decision only at run time. It needs to calculate the distance of a given point with all other points. So if dataset is large, there will be a lot of processing which may adversely impact the performance of the algorithm.
- KNN is also very sensitive to noise in the dataset. If the dataset is large, there are chances of noise in the dataset which adversely affect the performance of KNN algorithm.

- **How to choose optimal value of K in KNN Algorithm?**
- **Square Root Method**: Take square root of the number of samples in the training dataset.
- **Cross Validation Method:** We should also use cross validation to find out the optimal value of K in KNN. Start with K=1, run cross validation (5 to 10 fold), measure the accuracy and keep repeating till the results become consistent.
- K=1, 2, 3… As K increases, the error usually goes down, then stabilizes, and then raises again. Pick the optimum K at the beginning of the stable zone. This is also called **Elbow Method.**
- **Domain Knowledge** also plays a vital role while choosing the optimum value of K.
- K should be an **odd number**.

NAIVE BAIYES AND KNN (K-NEAREST NEIGHBORS)

- **How is KNN different from k-means clustering?**
- KNN is a supervised classification algorithm that will label new data points based on the 'k' number of nearest data points and k-means clustering is an unsupervised clustering algorithm that groups the data into 'k' number of clusters.
- **How to handle categorical variables in KNN?**
- Create dummy variables out of a categorical variable and include them instead of original categorical variable. Unlike regression, create k dummies instead of (k-1).