Machine Learning

Naïve Bayes

Overview

- Naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features
- They are among the simplest Bayesian network models
- Naïve Bayes has been studied extensively since the 1960s
- Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem

Bayes Theorem

- Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred
- Bayes' theorem is stated mathematically as the following equation:

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

  - where A and B are events and P(B) ? 0.
  - Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as **evidence**.
  - P(A) is the **priori** of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
  - P(AIB) is a posteriori probability of B, i.e. probability of event after evidence is seen.

Bayes Theorem

- Now, with regards to our dataset, we can apply Bayes' theorem in following way:

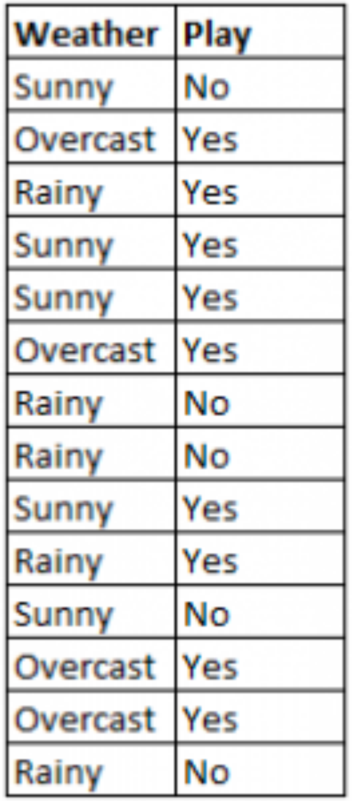
$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

- where, y is class variable and X is a dependent feature vector (of size *n*) where:

$$X = (x_1, x_2, x_3, ..., x_n)$$

How does it work ?

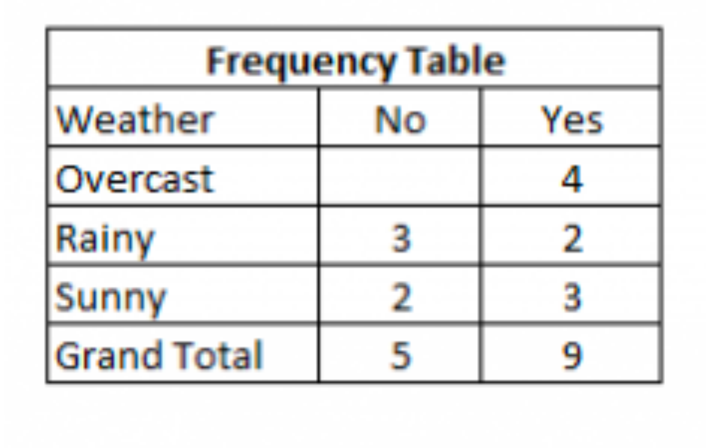
- Below is a training data set of weather and corresponding target variable ‘Play’ (suggesting possibilities of playing). We need to classify whether players will play or not based on weather condition.



Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

How does it work ?

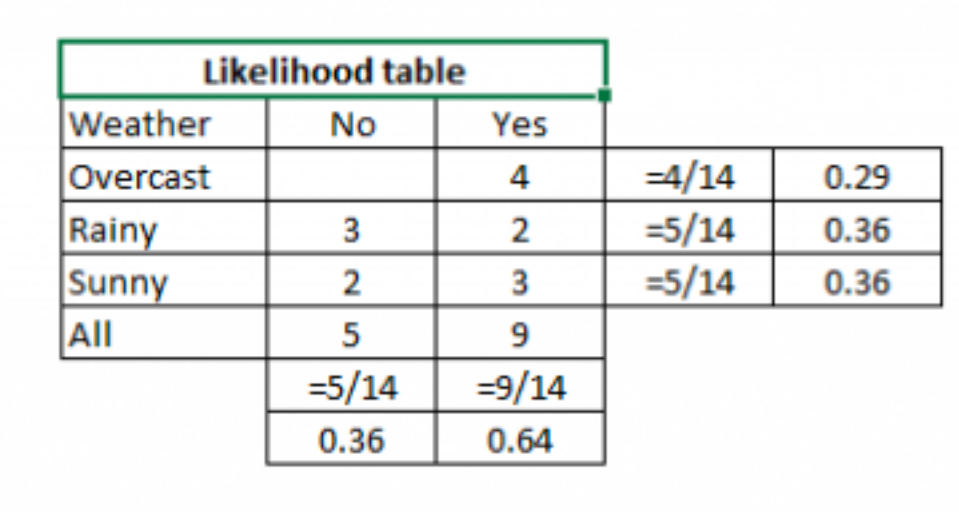
- Step 1: Convert the data set into a frequency table



Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

How does it work ?

- Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64



Likelihood table

Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

- Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

How does it work ?

- **Problem:** Players will play if weather is sunny. Is this statement is correct?
- We can solve it using above discussed method of posterior probability.

P(Yes I Sunny) = P(Sunny I Yes) * P(Yes) / P (Sunny)

- Here we have

P (Sunny IYes) = 3/9 = 0.33
P(Sunny) = 5/14 = 0.36
P(Yes)= 9/14 = 0.64

- Which means, P (Yes I Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Types of Naïve Bayes

- Gaussian Naïve Bayes classifier
- Multinomial Naive Bayes
- Bernoulli Naive Bayes

Advantages

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Disadvantages

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent

Applications of Naïve Bayes

- **Real time Prediction:** 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:** 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:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- **Recommendation System:** Naive Bayes Classifier and collaborative filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

Evaluation

Sensitivity

- Sensitivity tells us what proportion of the positive class got correctly classified
- A simple example would be to determine what proportion of the actual sick people were correctly detected by the model
- Also known as
  - True Positive Rate (TPR)
  - Recall

False Negative Rate

- False Negative Rate (FNR) tells us what proportion of the positive class got incorrectly classified by the classifier
- A higher TPR and a lower FNR is desirable since we want to correctly classify the positive class

Specificity

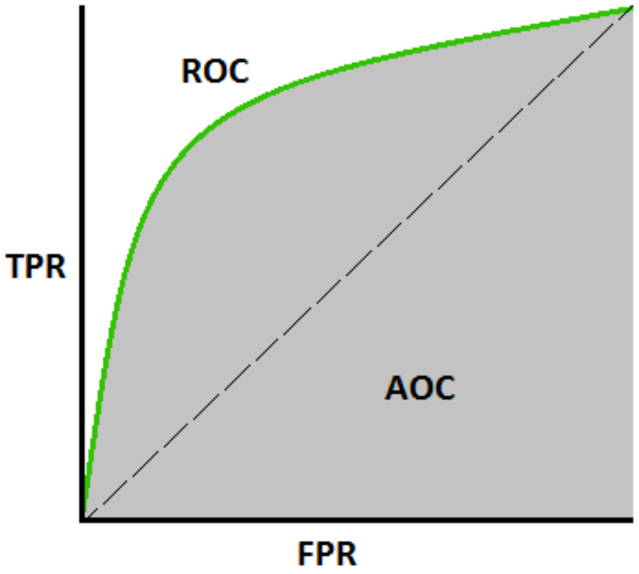
- Specificity tells us what proportion of the negative class got correctly classified
- Taking the same example as in Sensitivity, Specificity would mean determining the proportion of healthy people who were correctly identified by the model
- Also known as
  - True Negative Rate

False Positive Rate

- FPR tells us what proportion of the negative class got incorrectly classified by the classifier
- A higher TNR and a lower FPR is desirable since we want to correctly classify the negative class

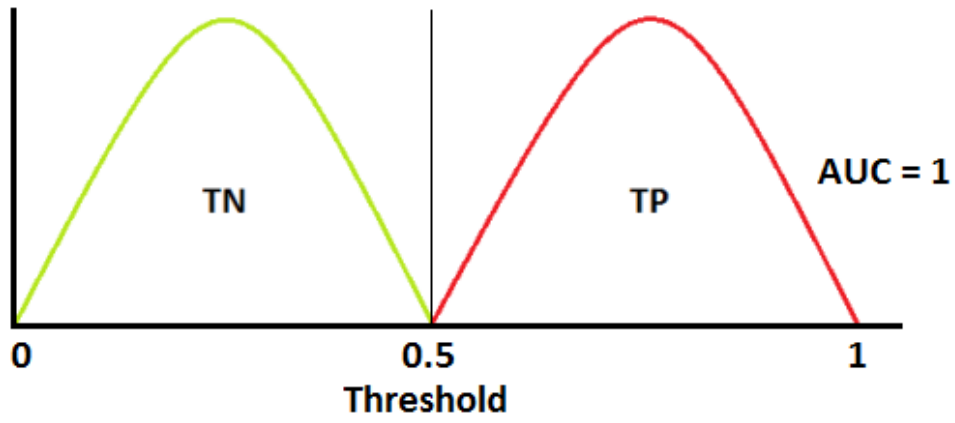
AUC-ROC curve

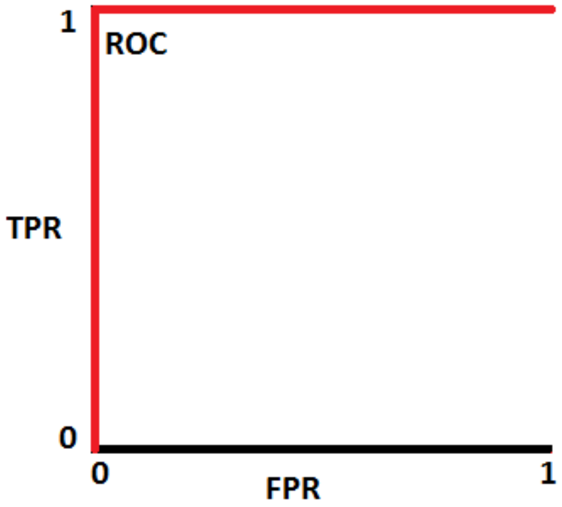
- The Receiver Operator Characteristic (ROC) curve is an evaluation metric for classification problems
- It is a probability curve that plots the TPR against FPR at various threshold values
- It separates the ‘signal’ from the ‘noise’
- The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve
- The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes



AUC-ROC curve

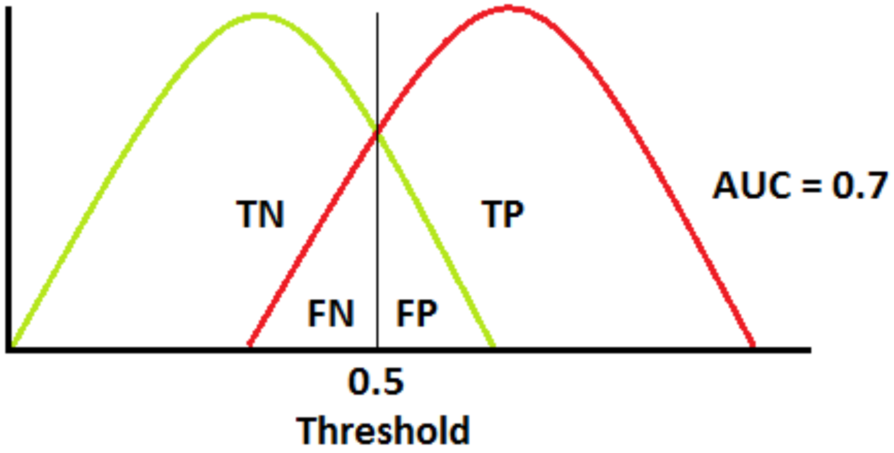
- This is an ideal situation
- When two curves don't overlap at all means model has an ideal measure of separability
- It is perfectly able to distinguish between positive class and negative class

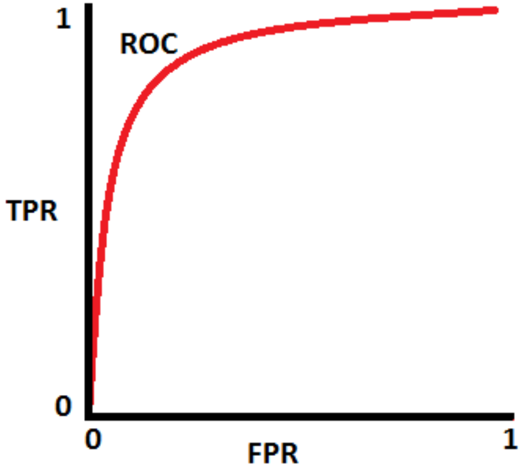




AUC-ROC curve

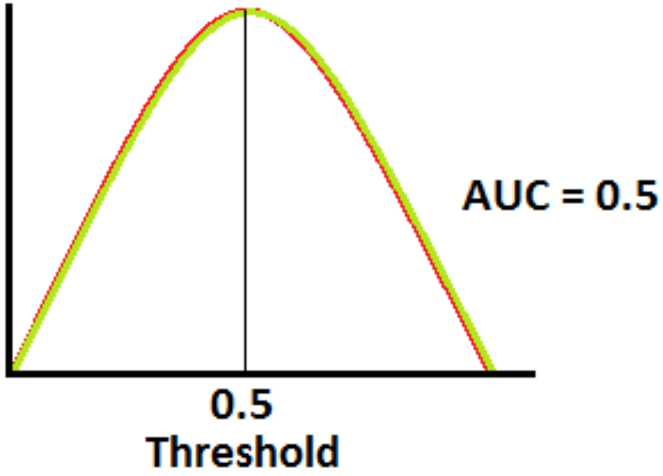
- When two distributions overlap, we introduce type 1 and type 2 errors
- Depending upon the threshold, we can minimize or maximize them
- When AUC is 0.7, it means there is a 70% chance that the model will be able to distinguish between positive class and negative class

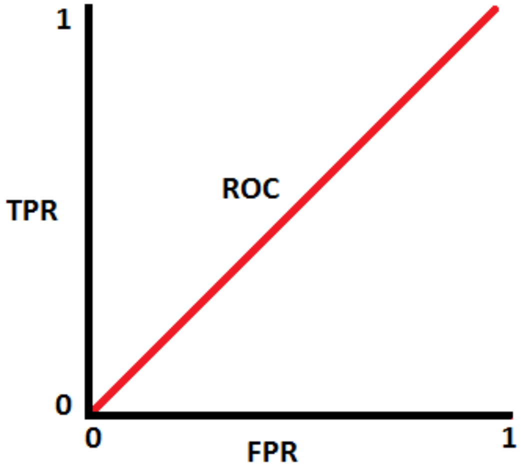




AUC-ROC curve

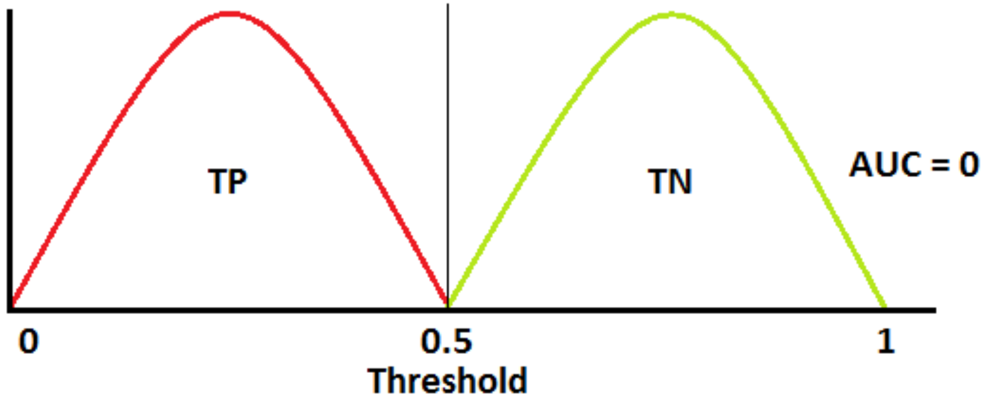
- This is the worst situation
- When AUC is approximately 0.5, the model has no discrimination capacity to distinguish between positive class and negative class

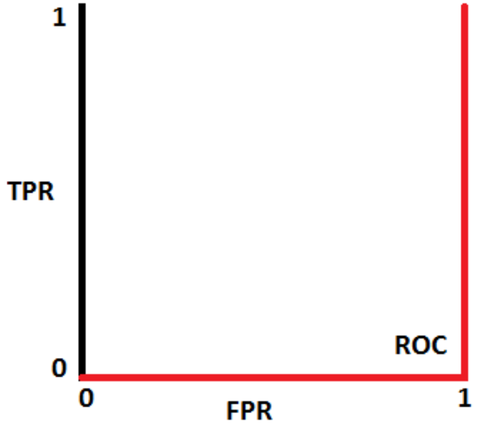




AUC-ROC curve

- When AUC is approximately 0, the model is actually reciprocating the classes
- It means the model is predicting a negative class as a positive class and vice versa


TP
TN
AUC = 0
Threshold


TPR
ROC
FPR