Naive Bayes

- Naive Bayes, which is extremely fast relative to other classification algorithms.
- It works on Bayes theorem of probability to predict the class of unknown data set.

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

P(B|A) is the posterior probability of class (B, target) given predictor (A, attributes).
P(B) is the prior probability of class.
P(A|B) is the likelihood which is the probability of predictor given class.
P(A) is the prior probability of predictor.

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems.

Naive Bayes

- Naive Bayes assumes to be conditionally independent given the target value.
- This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.
- The most important is naive Bayes representation is in probabilities
- **Class Probabilities:** The probabilities of each class in the training dataset.
- **Conditional Probabilities:** The conditional probabilities of each input value given each class value.
- Learning a naive Bayes model from your training data is fast. Training is fast because only the probability of each class and the probability of each class given different input (x) values need to be calculated. No coefficients need to be fitted by optimization procedures.

Naive Bayes

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

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

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		

- Let's take the above example and solve this with Naive Bayes.
- The first step in naive Bayes it to convert the dataset into frequency dataset and then into likelihood dataset(probability dataset).
- Now, use a Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of the prediction.

Naive Bayes

- That is a simple problem with 1 feature and let's predict with test data.
- Now we found a sunny in the test data and we use bates theorem with above likelihood dataset and predict yes or no.
- P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)
- Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64
- P(No | Sunny) = P(Sunny | No) * P(No) / P (Sunny)
- Here we have P (Sunny |No) = 2/5 = 0.4, P(Sunny) = 5/14 = 0.36, P(No)= 5/14 = 0.36 = 0.4
- So we can see that P(Yes | sunny) has high probability than the P(No| sunny) so it implies it is sunny then you play the game.
- So what if we have multiple features lets solve that in next post.

Naive Bayes

- what if we have multiple features lets solve that in next post.

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

Here x_1,x_2....x_n represent the features, i.e they can be mapped to outlook, temperature, humidity and windy. By substituting for X and expanding using the chain rule we get.

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Now, you can obtain the values for each by looking at the dataset and substitute them into the equation. For all entries in the dataset, the denominator does not change, it remain static. Therefore, the denominator can be removed and a proportionality can be introduced.

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

If we multiply all the small probabilities we get very small values so it's better to apply log so that instead of multiply we add them.

Naive Bayes

- One important thing in Naive Bayes is what if the data we got in test data which we never saw in train data. Then what about the probability of those features(Mostly occurs when solving the text data.
- We can solve the above problem using Laplace smoothing.
- So we get Zero as a probability of that feature with all the classes and when we multiply with other feature probabilities the final result will be always Zero.
- Here we add a small value to the Bayes theorem like.
- $P(Y|X) = (P(X|Y) * P(Y) + alpha) / (P(X) + alpha * K)$
- Here K is no of distinct values it can take like whether the word is present or not so here K = 2
- alpha can be any value mostly it will be 1 and you can see it alpha increases the model leads to underfit and vice versa.

Naive Bayes

- Types of Naive Bayes Classifier
- **Multinomial Naive Bayes:** This is mostly used for the document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.
- **Bernoulli Naive Bayes:** This is similar to the multinomial naive Bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example, if a word occurs in the text or not.
- **Gaussian Naive Bayes:** When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.

Naive Bayes

- Prepare Your Data For Naive Bayes.
- **Categorical Inputs:** Naive Bayes assumes label attributes such as binary, categorical or nominal.
- **Gaussian Inputs:** If the input variables are real-valued, a Gaussian distribution is assumed. In which case the algorithm will perform better if the univariate distributions of your data are Gaussian or near-Gaussian. This may require removing outliers (e.g. values that are more than 3 or 4 standard deviations from the mean).
- **Classification Problems:** Naive Bayes is a classification algorithm suitable for binary and multiclass classification.
- **Log Probabilities:** The calculation of the likelihood of different class values involves multiplying a lot of small numbers together. This can lead to underflow of numerical precision. As such it is good practice to use a log transform of the probabilities to avoid this underflow.

Naive Bayes

- Prepare Your Data For Naive Bayes.
- **Kernel Functions:** Rather than assuming a Gaussian distribution for numerical input values, more complex distributions can be used such as a variety of kernel density functions.
- **Update Probabilities:** When new data becomes available, you can simply update the probabilities of your model. This can be helpful if the data changes frequently.