**Bayes’ Theorem: naive bayes model is easy to build and particularly useful for very large data sets. Bayes theorem provides a way of calculating posterior probability P(C|X) from P(C), P(X) and P(X|C).**

**How it works**

**Step 1: convert the data set into a frequency table**

**Step 2: Create likelihood table by finding the probabilities like overcast probability**

**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.**

**Pros:**

**It is easy and fast to predict class of test data set. Is 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 variable compared to numerical variables. For numerical variable, normal distribution is assumed**

***Cons:***

* **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.**
* **On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.**
* **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.**

**The reason of why don’t use linear regression or others which will provide a formula :** [**https://www.quora.com/What-algorithm-do-you-use-for-binary-classification**](%20https:/www.quora.com/What-algorithm-do-you-use-for-binary-classification)

Stay away from the **Regression**algorithms that are trying to fit a formula. For example, stay away from *Linear Regression*. This is because LR is sensitive to outliers but in the case of classification, correctly classified “outliers” are cases that are have high confidence.

Probably you should stay away from *Support Vector Machines (SVM)* as they better are for situations that have lots of features. This is especially true for sparse features where the features have predominantly one value).

**PipeLine-weka**