Naïve Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of independence between every pair of features.

In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations.

They require a small amount of training data to estimate the necessary parameters.

Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods.

The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution.

This in turn helps to alleviate problems stemming from the curse of dimensionality.

We used the implementation provided by Scikit-learn for this.

In our project, Gaussian Naïve Bayes has been implemented as the data collected for the second innings has all the input attributes as numeric with only the class attributes being nominal.

Therefore, equation (v) has been used for posterior calculations and thus finding the winning probability.

Naive Bayes classifier is the probabilistic classifier which is based on Bayes' theorem which depends upon the strong (naive) independence assumptions. This classifier assumes that presence (or absence) of a particular feature of a class is independent to the presence (or absence) of any other feature. It considers all of these features to independently contribute to find the probability even if the attributes depend on each

other. Depending on the accuracy of the probability, Naive Bayes classifier is used for efficiently training a supervised learning setting. It is based on the model of conditional probability in which a given instance has be classified is given by a vector x = (x1,..........,xn ) which represents some n features (dependent variables) of the n attributes, A1, A2, . . . ,

An respectively, it assigns the probabilities to each of k possible classes or outcomes [13].

Using the Bayes' theorem, the conditional probability is given as,



Where,

P(Ci|X) is the posterior probability of the class, given predictor (attribute),

P(Ci) is the prior probability of class,

P(X|Ci) is the likelihood which is the probability of predictor given class and P(X) is the prior probability of predictor.

The Equation can also be written of the form as depicted in

equation.



(iii) The evidence, also termed as normalizing constant is equal to the sum of the posteriors. The evidence can be ignored as it is a positive constant. (Normal distributions are always positive).

So, only calculate the numerator of the equation which is P(X|Ci)P(Ci).

This classifier is mainly applied to find the decision rule. The most common rule is to choose the one which is the most probable and that is known as the Maximum A Posteriori (MAP) decision rule. That is, out of all the probabilities that have been calculated using Naïve Bayes, the one with the maximum probability will be selected for decision making which can be calculated by using equation (iv).



Now, if numerical variables are there then they are assumed to be normal distributed for numerical variables.

Hence, Gaussian Naïve Bayes is used in this case which is similar to Naïve Bayes Classification except it deals with continuous data. The main assumption of the classifier is that continuous values are distributed on the basis of Gaussian distribution which are related to each class. Let x be a continuous attribute that the training data contains. Then firstly, the data is segmented by the class; the mean and the variance of x in each class is then computed. Let μ and ı2 is the mean and the variance of the values in x related with class c respectively.

So, probability distribution of any value of a particular class,

P(x=m|c), is computed by inserting m in the equation (v)

representing the probability as Normal distribution.

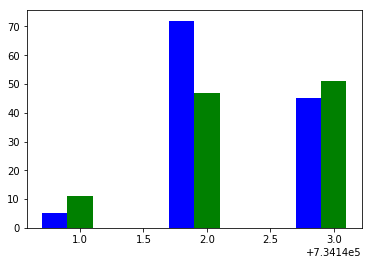


We Have a record of all the matches played by **MS Dhoni** until now.

Now Using Naïve Bayes algorithm we are trying to predict the actual runs he will scored against that particular team given that historical data is present against that given team.

When we ran the algorithm on the selected player, this is the accuracy we got.

|  |  |
| --- | --- |
| Actual Scored | Predicted Score |
| 11 | 5 |
| 47 | 72 |
| 51 | 45 |

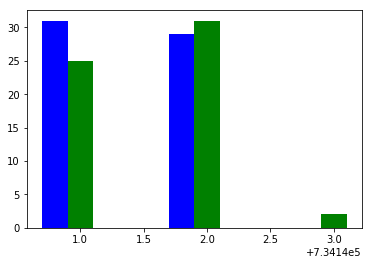


We Have a record of all the matches played by **V Kohli** until now.

Now Using Naïve Bayes algorithm we are trying to predict the actual runs he will scored against that particular team given that historical data is present against that given team.

When we ran the algorithm on the selected player, this is the accuracy we got.

|  |  |
| --- | --- |
| Actual Scored | Predicted Score |
| 25 | **31** |
| 31 | **29** |
| 2 | **0** |



Also we tried predicting the score of **RG Sharma** , but we didn’t have a sufficient data for making the accurate predictions.

This is the result that we got.

|  |  |
| --- | --- |
| Actual Scored | Predicted Score |
| 1 | **0** |
| 22 | **36** |
| 18 | **12** |
| 48 | **52** |

So, here we can conclude that, as our data gets self trained ( By improving the data we have and updating data in future) we can predict the runs more and more better.