Naive Bayes Classifier Analysis

Syed Aqhib Ahmed[1], Venkata Sai Abhishekh Sarvepalli[2]

***Abstract*— Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. In this paper, we will be looking at a use case of the naive bayes classifier in the form of a multi-class classifier, and studying the effects of normalization and using multivariate conditioning on the data[1].**

I. INTRODUCTION

In this classification problem, we have been given data from an experiment, which includes certain measurements F_1 and F_2 measured while the participants performed five different tasks $(C_1, C_2, C_3, C_4, C_5)$. The objective is to create a naive bayes classifier which classifies any measurement value into one of the five classes$(C_1, C_2, C_3, C_4, C_5)$, given the type measurement it is (F_1, F_2). Five probabilities are calculated for each data point from the given measurement data, and the most probable class is shown as the prediction.

$$PredictedClass = argmax[P(C_i|X)] \text{ i=1,2,3}$$

II. BUILDING THE CLASSIFIER

A. Calculating the Mean and Variance

The given data has a thousand records and five attributes, where each record represents one person's measurements during the five activities performed. In-order to calculate the mean and the variance (m_{1i}, σ^2_{1i}) and (m_{2i}, σ^2_{2i}) for all classes as specified (using the first 100 records), we divide the data set into train (100 rows) and test set (900 records). Further, we calculate the mean and the variance for all classes for the two types of measurements by using this train data.

B. Creating a classifier

We create a classifier in MATLAB which accepts the measurement values, means and standard deviations.

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

We then use naive bayes as described in the formula above to compute probabilities of the data points belonging to each class. As described in the problem statement, the probability of a measurement being a certain type (1 or 2), given the type of task, is a normally distributed with mean and variance (m_{1i}, σ^2_{1i}) for type 1 and (m_{2i}, σ^2_{2i}) for type 2.

C. Computing the accuracy of the classifier

Computing the accuracy of the classifier is fairly straightforward. To check the accuracy of the classifier model we have constructed in the previous step and examine its performance, we subtract the predicted classes from the reference classes and find the difference values between the predictions and the references. We can see that the number of zero values in this resulting vector is the number of correct predictions that our classifier has made. Now that we have the exact number of correct predictions, we can find the accuracy by dividing the number of zero values in this resulting matrix by the total number of predictions.

$$error = \frac{zeros(\hat{Y} - Y)}{N}$$

$\hat{Y}$= Predicted Values Y=Actual Values

We find that the error rate for X = F1 is 47.83%.

D. Comparing with the standard normal

We use the data-points in the test set of the F_1 and F_2 data tables and convert those points into Z scores (standard-normal scale) by removing the mean and dividing by the standard deviation.Further, we plot the data-points Z1 vs F2 on scatter plot to draw a comprehensive comparison between it and the given F1 vs F2.

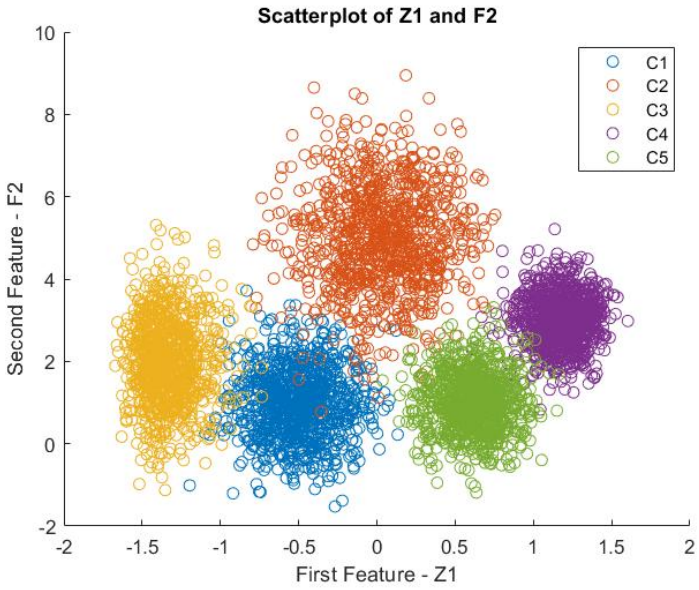

Figure 1.1 : Z_1 vs F_2

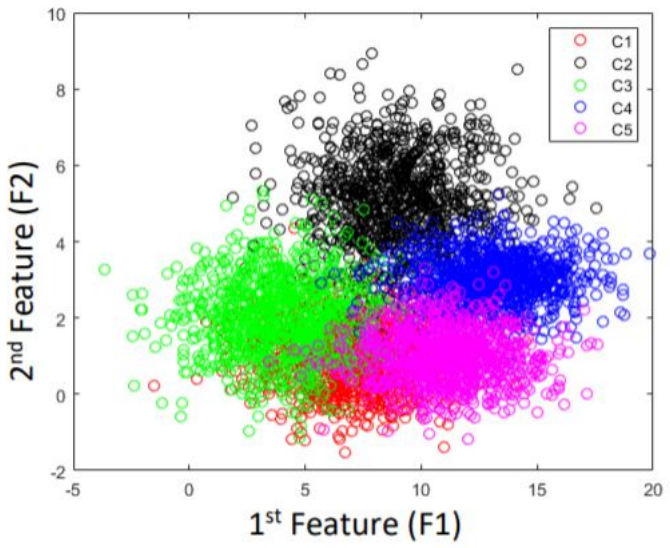

Figure 1.2 : F_1 vs F_2

Comparing the two plots, we see that using the normalized z-score provides us with a plot which has far less overlapping points than if we use the non-normalized version of the measurements. This makes is easier to classify the measurements into classes given these two values (Z1 and F2). We repeat the predictions for given values(B) of Z1, F2 and $[Z1, F2]$.

III. COMPARISONS

We find that the accuracy we get by using F1 and F2 as given(X) are very similar, as is to be expected, since they provide the same information to the bayesian classifier. We get an accuracy of 52.62% for Case 1 (X = F1) and 53.51% for Case 3 (X = F2). Errors for all cases have been tabulated below. We also find that using Z1 as given (X = Z1) instead of F1 results in a significant jump of accuracy significantly. This can be attributed to the fact that we are using normalized values in Z1. Using normalized values compensates for the fact that we are using measurements which have different ranges by bringing them to a zero mean and unit variance. The classification accuracy goes up by 35% in this case. Next, we use multivariate X as given (X = [Z1, F2]). We assume that both of the random variables have zero covariance and are independently distributed. As we noticed from the graph before, if we are given both the Z1 and F2 values, it is quite easy to distinguish between the classes as they are clustered closer to each other and do not overlap as much.

Case	X	Accuracy
1	F1	52.62
2	Z1	88.38
3	F2	53.51
4	[Z1 F2]	97.84

IV. CONCLUSIONS

We can conclude from our observations that the more prior data we have about the system that we want to make our predictions on, the more accurate our predictions can get.

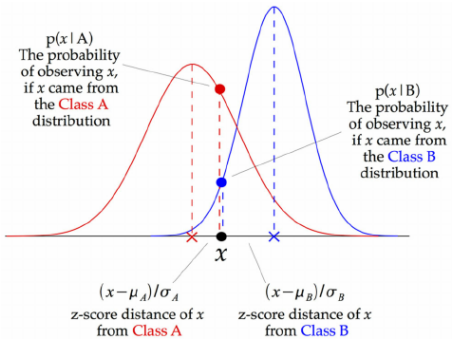

Figure 1.3 : Bayes demonstrations for two classes[2].

The probability accuracy increases as the given information increases.

REFERENCES

[1] Raizada, Rajeev Lee, Yune. (2013). Smoothness without Smoothing: Why Gaussian Naive Bayes Is Not Naive for Multi-Subject Searchlight Studies. PloS one. 8. e69566. 10.1371/journal.pone.0069566.

[2] Wikipedia contributors. "Naive Bayes classifier." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 26 Nov. 2018. Web. 16 Dec. 2018.