

Midterm Project Write-up

Adrian Ainsworth

Math 179

1 Naive Bayes Classifier

The Naive Bayes Classifier is an algorithm that utilizes Bayes' Theorem, which describes the probability of an event based on prior knowledge of the conditions that could possibly be relevant.

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

Figure 1: Bayes' Theorem

The left side of the equation, $P(A|B)$, is known as the posterior, and describes the probability that the class is A given that the data is B. The right side of the equation, $\frac{P(B|A)P(A)}{P(B)}$ is known as the likelihood, and essentially describes the probability that given the data B, the class is A [1]. When it comes to this theorem's relevance in the real world, numerous potential applications of Bayes' Theorem can be found in the medical field, where it can be used to take the symptoms and

demographic data of a particular subject into account in order to determine the probability that they will be diagnosed with a certain health condition.

The way that Naive Bayes algorithm in particular works is by applying Bayes' Theorem under the assumption that the predictor variables conditional on the response will be independent[2], as well as assuming that all of the features contribute equally to the outcome[3].

2 Implementation

For this project, I decided that I wanted to examine medical data sets listing health information about patients and see if it was possible to find patterns in this data that corresponded to whether these patients had a history of heart attack, with the eventual goal of developing a model that can accurately predict whether a patient has a history with or will experience a heart attack based on the input of these attributes. In order to do so, I recreated the implementation of a breast cancer diagnosis predictor from Normalized Nerd[4]. The author of this implementation analyzes the mean radius, texture, and smoothness of tumors found on X-ray to predict whether these tumors will be diagnosed as benign or malignant. The data set that I used came from Kaggle[5], and it contained about 724 entries detailing the physical, medical, and lifestyle attributes of various patients. Specifically, each sample had data sorted into twelve columns: ID (0), Name (1), Age (2), Gender (3), Height(cm) (4), Weight(kg) (5), Blood Pressure(mmHg) (6), Cholesterol(mg/dL) (7), Glucose(mg/dL) (8), Smoker (9), Exercise (hours/week) (10), and Heart Attack (11).

The first step before working with the data was to determine whether the features included in the data set were independent of each other, a necessary assumption to utilize the Naive Bayes classifier on the data. In order to determine this, I utilized the Pearson correlation coefficient, which is used to measure the linear correlation between two sets of data. To visualize the Pearson correlation score between the features, I created a heatmap showing the score associated with each pair.

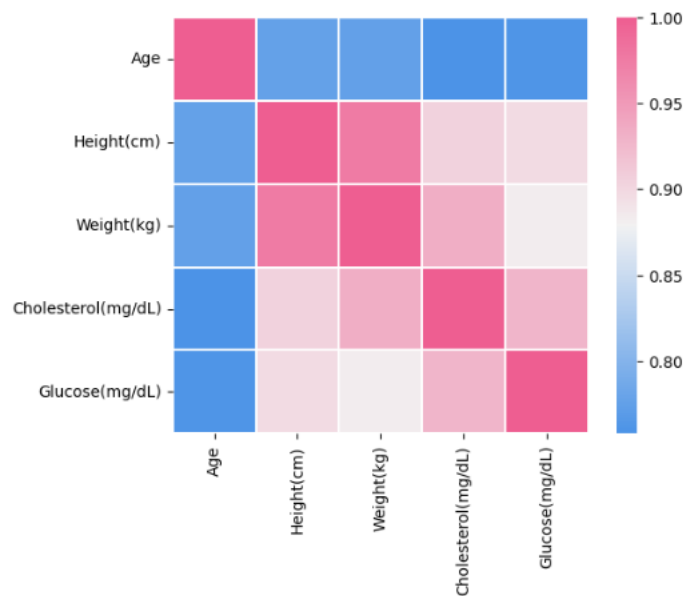


Figure 2: Heatmap of Pearson correlation score

The results of the heatmap were generally as expected, with age showing the lowest correlation scores overall to other features and each feature having a high correlation score with itself. Additionally, I noticed that out of all the combinations, height and weight in particular showed the highest correlation scores to each other. Because of this, I decided to remove the Weight(kg) column and include only the height data in my analysis instead. Another issue encountered during this part

of the process was the inclusion of blood pressure in the data set, which is measured and recorded as two separate numbers, respectively indicating the systolic and diastolic pressure, within the same entry. Because the two numbers are generally considered together and not separately, I decided to leave the Blood Pressure (mmHg) column out of my analysis as well, especially since its values were given as an object instead of as a float or integer. Lastly, I also decided to exclude the Exercise(hours/week) data as its inclusion made the rest of the heatmap more difficult to read.

Next, I created graphs to visualize the distribution of each feature.

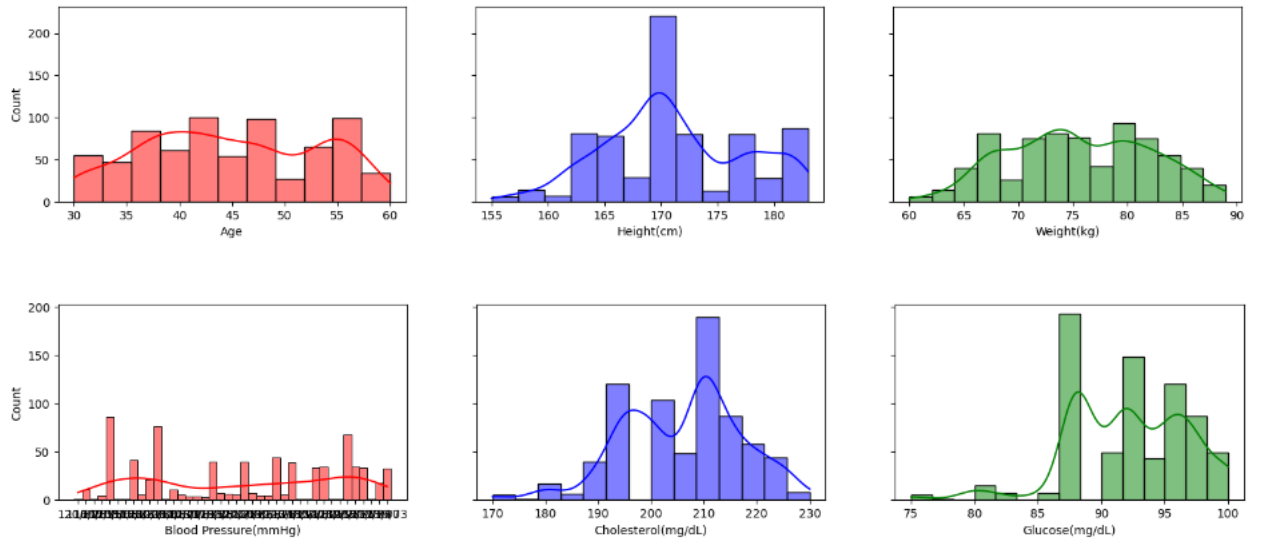


Figure 3: Bar curve graph of features

After narrowing down the data set to the desired columns and converting the vari-

able types of the data to floats, I proceeded to test the data using two different models, the first of which was a Gaussian distribution model. After calculating the likelihood, by training and testing the model, I was able to generate a confusion matrix for the data set.

```
[[57  0]
 [ 1 87]]
0.9942857142857142
```

Figure 4: Confusion matrix for Gaussian model

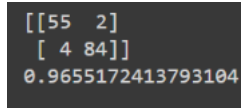
As seen in the confusion matrix pictured above, in the test I ran the model had no false negatives and one false positive, as well as a high F1 score.

Next, I tested the data set with a model where the continuous features were converted to categorical features.

	catGlucose	catChol	Heart Attack
0	1	1	0
1	0	0	0
2	2	2	1
3	1	0	0
4	2	1	1
5	0	0	0
6	2	2	1
7	1	1	0
8	2	1	1
9	1	1	0

Figure 5: Variables converted to categorical

After calculating the likelihood and posterior probabilities and training and testing the model I was able to obtain another confusion matrix.



```
[[55  2]
 [ 4 84]]
0.9655172413793104
```

Figure 6: Confusion matrix from categorical approach

As seen in the confusion matrix pictured above, this approach ended up performing slightly worse than the Gaussian approach, having a higher amount of false positives and false negatives, as well as subsequently having a lower F1 score.

Overall, I was able to successfully replicate the Naive Bayesian Classifier implementation, obtaining similar results to the author, especially regarding the amount of false negatives relative to each model that was used. Continuing with this project, I will attempt to apply what I've learned to a larger dataset and attempt to implement more complex models to see if I can more accurately predict a diagnosis based on symptoms.

3 References

References

- [1] “Bayes Classifier and Naive Bayes Tutorial (Using the Mnist Dataset)” Lazy Programmer, lazyprogrammer.me/bayes-classifier-and-naive-bayes-tutorial-using/.
- [2] “Naïve Bayes Classifier.” H2O 3.46.0.1 Documentation, docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/naive-bayes.html.
- [3] Kavlakoglu, Eda. “Classifying Data Using Multinomial Naive Bayes Algorithm.” IBM Developer, developer.ibm.com/tutorials/awb-classifying-data-multinomial-naive-bayes-algorithm/.
- [4] <https://www.youtube.com/watch?v=3I8oX3OUL6I>
- [5] <https://www.kaggle.com/datasets/mahad049/heart-health-stats-dataset/data>