Accurate Classification of Diabetes using Two Different Classifiers

A data mining problem

Fouad Lamsettef (s1034545)
Daniel M.S. Putra (s1078257)

1 Summary

In this study, we aimed to accurately classify individuals as having diabetes or not using two classifiers which is K Nearest Neigbors classifier and Naive Bayes classifier. Our data set included several attributes such as BMI, glucose consumption, blood pressure, pregnancy, age, insulin level, etc. We trained and tested both classifiers using this data set and compared their performance.

The algorithm first predicts the likelihood of each class (diabetic or non-diabetic) based on the attributes in the data set to categorize individuals using naive Bayes. It then utilizes these probabilities to forecast each individual's class. KNN classification, on the other hand, works by locating the k (a predetermined number) nearest neighbors to each individual and then predicting the class based on the majority class among these neighbors.

2 Introduction

Diabetes is a chronic condition brought on by either insufficient insulin production by the pancreas or inefficient insulin utilization by the body where insulin is a hormone that regulates the glucose of the blood. Obesity is the major risk factor for the development of prediabetes and type 2 diabetes [1]. It is one of the main causes of morbidity and mortality in the globe with the risk of coronary heart disease and major stroke subtypes [2].

The dataset that we are going to use and classify is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. Based on specific diagnostic metrics present in the dataset, the dataset's goal is to predict whether a patient has diabetes or not. These instances were chosen from a bigger database under a number of restrictions. Particularly, all patients thet we got this dataset are Pima Indian women, which are are basically north american indians, who are at least 21 years old[3].

The application domain of the prediction of diabetes is with the use of a data set and predicting it with the use of Naive Bayes and K Nearest Neigbors classifiers[4]. To develop precise and effective ways for detecting people who are at risk of getting diabetes or who already have the condition is the research problem in this field.

By analyzing a data set of patient characteristics and making predictions about whether a person has diabetes or not, the approach to addressing this research problem would be by using machine learning techniques like the Naive Bayes and K-Means classifier can be used to address this research challenge.

These classifiers are commonly used in medical research because they are able to handle large datasets and can effectively classify individuals based on a variety of attributes, such as body mass index (BMI), glucose consumption, blood pressure, pregnancy status, age, and insulin levels. By comparing the performance of these two classifiers, researchers can determine which approach is more effective at predicting diabetes status and potentially develop improved methods for identifying and managing this chronic disease.[5]

It is important to note that predicting diabetes from a data set is just one aspect of addressing the overall problem of diabetes management. Other considerations may include treatment options, lifestyle changes, and ongoing monitoring and support for individuals with diabetes.

3 Pre-processing

3.1 Handling Missing Values

Before we train our classifier, it would be a good idea to preprocess the data set. One important step in preprocessing is handling missing values. Handling missing values is important for several reasons when classifying diabetes from a data set using naive Bayes or k-nearest neighbors (KNN).

1. It can impact the accuracy of the classifier. If a model is trained on a data set with missing values, it may make incorrect predictions because it is not using all of the available information that is provided from the data set. Handling this problem can ensure us that our model is using all of the available data to make predictions.

2. Missing values can impact the performance of the classifier we are using. If the data set have alot of missing values, it will decrease the size of the data set for quite a lot and this will make it more difficult for the classifier to learn from the data since there is not much information. This may lead to low accuracy and poor performance.

Overall, handling missing values is one of the most important steps for preprocessing a data set for classification as it can improve the accuracy, performance, and interpretability of your model[6].

3.2 Missing Values in the Dataset

After printing the data, we've seen that there are some attributes have 0 as minimum value which are Glucose, Bloodpressure, SkinThickness, Insulin and, BMI. This basically means that there is a point where the minimum value of Glucose is 0 which is not possible. The same goes to the Bloodpressure, SkinThickness and the others. This means that we have missing/invalid values in the following columns.

What we do for this problem is just removing them from the data set since they don't play a role anymore and will disrupt the performance if we did not do anything to it. After removing them, it changes the information of the Histogram which is much more accurate now and will definitely give a better performance.

4 Models and methodology

4.1 General Training

We perform each training 10 times for producing a more better and accurate results. For each iteration we used a different random split, this is similar what is done in K-Fold cross-validation[7]. Training a model multiple times can help understand how well the model generalizes to new data. If the model performs consistently well across multiple training runs, it is likely to be more robust and less prone to overfitting.

4.2 Naive Bayes

Starting off with our first classifier, we decided to use the Naive Bayes classifier. There are several reasons why using Naive Bayes classifier can be a good approach on classifying diabetes:

1. Naive Bayes is fast and can handle large amount of data efficiently which is important in this case since working with diabetes, there can be a quite large data for identifying it.

2. It also handle missing data quite well since it can be crucial for the final result if there are some missing data that's classified.If some of the feature values are missing, the classifier can still make a prediction based on the available data.

This can be especially useful in a real-world setting, where it is not always possible to collect all of the data that you would like. However in our case, we have checked the data set that we use and there is no missing data set in it for the naive Bayes classifier.

There are multiple versions of Naive Bayes which is the Categorical Naive Bayes and Gaussian Naive Bayes. However, we decided to use the Gaussian Naive Bayes. Categorical Naive Bayes is used for classification when the features are categorical, however in our data set there does not seem to be any categorical variable.

Gaussian naive Bayes can be more effective and simpler to implement than other types of models, which is a reason to implement it for classification. It is particularly well-suited for classification tasks when the features are continuous variables, rather than categorical variables. In this case, the model assumes that the continuous variables are distributed according to a Gaussian (normal) distribution, which makes it easy to estimate the probability of a given data point belonging to a particular class.

In general, if the features are continuous variables, Gaussian naive Bayes may be the better choice. If the features are categorical variables, categorical naive Bayes may be the better choice[8].

4.3 K-nearest neighbors

K-nearest neighbors classifier is a type of instance-based learning, where there is no global model build but instead the classifier uses training examples to make predictions on test instances[7]. The classifier uses local information to make their predictions. The optimal choice for k is highly dependent on the data set[9], therefore we need to find the optimal k in our data set. We test our data set with $k \in \{1, .., 20\}$.

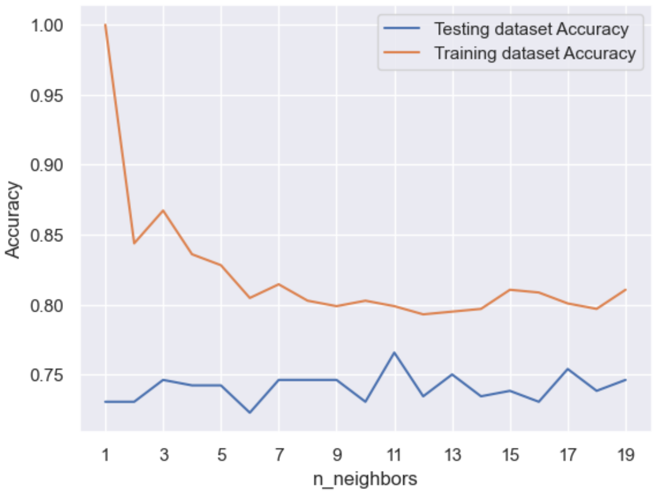

Figure 1: Finding the optimal k

From the figure we find that the optimal k value is 11 and so we use that value of k for our testing and comparison with naive bayes.

5 Results and analysis

5.1 Confusion Matrices

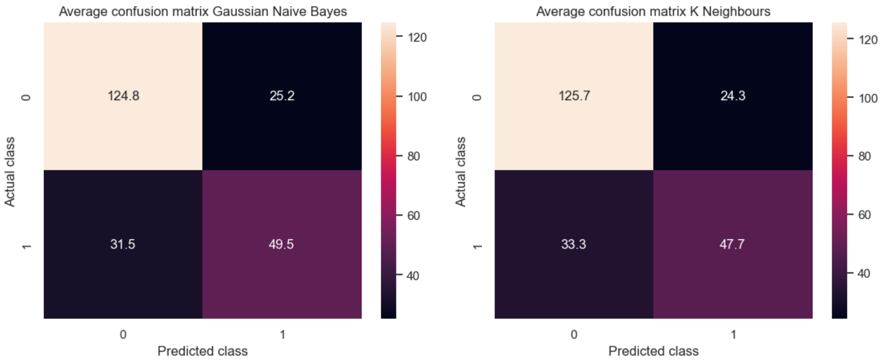

Figure 2: The confusion matrices

As can be seen by the confusion matrix from the image above, it have been made by using the average of 10 resulting training iterations with the algorithm. From the top left , it can be seen that the K-nearest neighbors is slightly better than the Naive Bayes classifier for predicting if a certain someone is non diabetic. Meanwhile, the Naive Bayes is also doing a slightly better job then the K-nearest neighbors on predicting if someone is positive diabetes.

5.2 Accuracy and AUC scores

Average AUC Gaussian Naive Bayes:	0.8215172063078156
Average AUC K Neigbors:	0.8223450132614349
Average accuracy score Gaussian Naive Bayes:	0.7545454545454546
Average accuracy score K Neigbours:	0.7506493506493507

Table 1: The average accuracy and AUC scores for both classifiers

The Table above shows the calculated scores of the average of the Gaussian Naive Bayes and K Neigbors based on 10 iterations that we have done. The accuracy score indicates the percentage of labels that were properly predicted. It's not uncommon for different evaluation metrics to produce different results when comparing the performance of 2 different classifiers.

In this case, it can be seen that the AUC score for the K Neigbours is slightly better than the Gaussian Naive Bayes. Its the opposite for the accuracy score where the Gaussian Naive Bayes scores a slightly higher percentage than the K Nearest Neigbors. It's possible that one algorithm is better suited to the characteristics of the data set, which could explain why it performs better on one evaluation metric.

5.3 ROC Curves

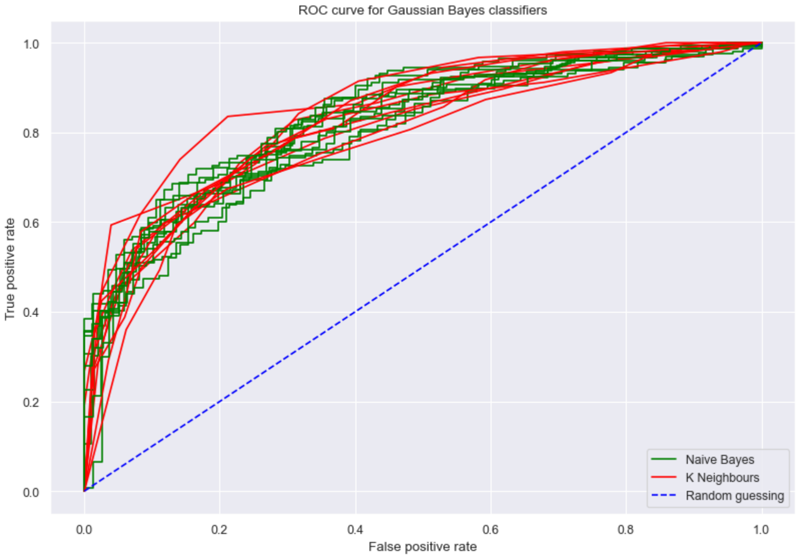

Figure 3: The ROC Curves for Naive Bayes and K Nearest Neighbors

Another tool we measure the classification model visually is by using the ROC Curve. Using an ROC curve can be useful for evaluating the performance of a classifier. The ROC curve allows you to visualize the trade-off between the true positive rate and false positive rate for different classification thresholds. This can be useful for understanding the sensitivity and specificity of the classifier, as well as for comparing the performance of different classifiers[10].

It seems like the difference between the worst and best iterations are quite similar to each other for both classifiers. We can also see that the graph is quite messy, deviate quite a lot and not as smooth. This is due to the fact that our data set is considered as a small amount since it does not even exceed 1000 which makes it quite visible for the ups and downs.

5.4 PCA

PCA number	explained variance	Attribute
1	0.29	Glucose
2	0.19	Pregnancies
3	0.14	Insulin
4	0.11	DPF
5	0.1	Blood Pressure
6	0.07	Glucose
7	0.06	Pregnancies
8	0.05	Age

Table 2: PCA analysis of the data (without missing values)

We also ran a PCA check to see which attributes to most data variance in our data set. To this we made 8 principal components and ordered by the amount of variance they explain (see table 2) and identified which attributes contributed the most for each principal component. From the table we see that the first 6 components contribute to 90 percent of the explained variance, where PCA 1 has an explained variance of 29 percent with glucose contributing the most in that principal component.

6 Conclusion

Even though Naive Bayes Classifier is a good choice for this task as can be seen from the results, its always a good idea to try multiple algorithms and evaluate their performance in order to select the most appropriate model for a specific data set and so it is also the reason why we use K Neigbours classifier.

In conclusion, our model yields a good performance for Gaussian the Naive Bayes Classifier and the K-nearest neighbors as indicated by the accuracy of the model which is 0.821 and 0.822 if rounded. Our model also does not seem to have a sign of overfitting since the value of the training set and the test set are quite comparable. Both algorithms have their own strengths and weaknesses, and which one is better for a particular problem will depend on the specific characteristics of the data set.

For future results, it would be a good idea to investigate in which attributes really made the difference of the performance for these 2 classifier which 1 performs better at.

References

[1] Javier Gómez-Ambrosi et al. "Body Adiposity and Type 2 Diabetes: Increased Risk With a High Body Fat Percentage Even Having a Normal BMI". In: *Obesity* 19.7 (July 2011), pp. 1439–1444. DOI: 10.1038/oby.2011.36. URL: http://dx.doi.org/10.1038/oby.2011.36.

[2] The Emerging Risk Factors Collaboration. "Diabetes mellitus, fasting blood glucose concentration, and risk of vascular disease: a collaborative meta-analysis of 102 prospective studies". In: *The Lancet* 375.9733 (June 2010), pp. 2215–2222. DOI: 10.1016/s0140-6736(10)60484-9. URL: http://dx.doi.org/10.1016/s0140-6736(10)60484-9.

[3] Jack W. Smith et al. "Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus". In: *Annual Symposium on Computer Application in Medical Care* (Nov. 1988), pp. 261–265.

[4] *Scikit Machine Learning Python URL*. URL: https://scikit-learn.org/stable/.

[5] Sellappan Palaniappan and Rafiah Awang. "Intelligent heart disease prediction system using data mining techniques". In: *2008 IEEE/ACS International Conference on Computer Systems and Applications* (Mar. 2008). DOI: 10.1109/aiccsa.2008.4493524. URL: http://dx.doi.org/10.1109/aiccsa.2008.4493524.

[6] Edgar Acuña and Caroline Rodriguez. "The Treatment of Missing Values and its Effect on Classifier Accuracy". In: *Classification, Clustering, and Data Mining Applications* (2004), pp. 639–647. DOI: 10.1007/978-3-642-17103-1\{_}60. URL: http://dx.doi.org/10.1007/978-3-642-17103-1_60.

[7] Pang-Ning Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Pearson, 2014.

[8] *1.9 Naive Bayes*. URL: https://scikit-learn.org/stable/modules/naive_bayes.html.

[9] *1.6. Nearest Neighbors*. URL: https://scikit-learn.org/stable/modules/neighbors.html.

[10] *Roc Curve*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html.