Predicting Diabetes in Female Patients: Comparing the performances of various machine learning classification algorithms

Dataset: https://www.kaggle.com/uciml/pima-indians-diabetes-database

Michael Ross (SID: 201589412)
Shuhei Ishiwatari (SID: 201603195)

University of Liverpool
COMP 534: Applied AI
Assignment 1
16 March 2022

Background

The accompanying dataset was provided by Kaggle and contains diagnostic measurements of female patients of at least 21 years of age and of Pima Indian heritage. These measurements are to serve as potential indicators and predictors of diabetes. This project is to design and implement three different machine learning classifiers to predict, based on the given data, which patients are diabetic and then compare the performances of these classifiers.

Introduction

Note: Please view diabetes.ipynb file for full code and execution.
Primary Libraries Used:

Data pre-processing
- Pandas
- NumPy

Data visualisation & exploratory analysis
- Seaborn
- Matplotlib
- Math
- Scipy

Modelling
- KNeighborsClassifier, GaussianNB and Logistic Regression available from scikit-learn

Evaluation
- confusion_matrix, accuracy_score, recall_score, precision_score, f1_score available from scikit-learn

Classification Methods:
- K-Nearest Neighbor: The first classification method we used is the KNN classifier, in which we used two primary parameters. The first is n_neighbors (k) to indicate how many neighboring points we wanted to compare our test set in. For this parameter, we used a for loop to test a range of 1-15 k-points to find the best resulting one. We then used the euclidean power parameter, which finds the k nearest points using the euclidean distance formula.
- Logistic Regression: Logistic regression is a statistical method that models the probability of an object belonging to a certain class. We selected logistic regression to be our first probabilistic classifier. We have explored 3 hyperparameters ("penalty", "C" and "solver") using GridSearchCV for our logistic regression model. The first term refers to the regularisation term which was set at the L2 regularisation ("l2") in our experiment. The second term C is the inverse of regularisation strength. In our experiment, this was set at 0.001 after exploring 20 numbers on a log scale. The third term specifies an algorithm to be used in the optimisation process. For this problem we selected 'newton-cg' among 'liblinear' and 'lbfgs'.
- Naive Bayes: Naive bayes is a probabilistic classifier based on Bayes' theorem with a naive assumption of conditional independence between the features. We selected Naive Bayes as our second probabilistic classifier. Since there are no hyperparameters to tune in the same sense as KNN and logistic regression, we simply trained the model to predict diabetic patients.

Training and Testing Process:
Our initial observation of the dataset was that there were several key columns that contained 0's as a value, which is medically impossible and indicates these are null values that were either erroneously entered or the measurements were not taken of the patient. Our group implemented two methods for missing value handling:

Method 1: Our first approach aimed to eliminate records that contained a 0 in a column that we designated as not allowing a 0 value. Prior to doing so, we noticed that there was a significant imbalance of outcomes, noticing that approximately 65% of the records were designated as not having diabetes.

Since a 65 to 35 outcome ratio could teach a classifier to favor outcomes as not having diabetes, we had hoped that eliminating records containing a 0 in the no 0 columns and a 0 as an outcome might balance the dataset.

Our assumptions were correct and after eliminating these records, the outcomes were now balanced with outcome 0 occurring in just under 49.5% of all remaining records. With the outcomes split nearly 50:50, our group decided to fill the remaining 0's with the mean of their respective features.

We then created two separate dataframes with one containing all features for testing and comparing purposes and another with select features that we obtained through a series of correlation matrices and using ExtraTressClassifier, which shows the scores of each feature where high scores indicate a bigger impact that feature has on the outcome.

Dataframes created: df_original, df_original_dropcols

Method 2: As an experimental part of the project, our group also created two additional dataframes where the missing values were predicted using missing value imputation with regression. For this method, instead of dropping rows with a 0 value as we did in method 1, we used linear regression to predict the null values by using feature correlation. For example, we noticed there was a significant correlation between BMI with skin thickness and glucose with insulin.

Additionally, because this method does not fix the outcome imbalance as in method 1, we took this into consideration within our classifier functions by implementing Synthetic Minority Oversampling Technique (SMOTE) using the imbalance-learn library to create fake records of the minority class (outcome 1) using observed instances. The accompanying .ipynb file shows two versions of each classifier to test method 2's dataframes.

We also created two separate dataframes - one with all features and another with select features.

Dataframes created: df, df_dropcols

Evaluation

To evaluate the performance among three classifiers, we selected four evaluation metrics (accuracy, recall, precision and F1-score) along with confusion matrices. However, the best classifier was defined in relation to the model that maximises recall for the following reasons. First, the class distribution is skewed towards the negative class in our current dataset. Around 70% of data comes from the negative class. An algorithm therefore could classify all new objects into the negative class and still achieve relatively high accuracy. Second, the current problem is medical diagnosis. one should seek to reduce type II errors (false negatives) rather than type I errors. That is, the situation where a model predicts diabetic patients as non-diabetic should be avoided. For all the reasons, the classifier that minimises false negative (the highest recall) was considered the best in our current report.

Hypothesis: KNN classifier will perform the best with df_original_dropcols dataframe (this dataframe is obtained in method 1 from dropping rows with null values and having an outcome of 0, then replacing the remaining 0's with the mean of the feature). We believed that KNN would be the best performing classifier as KNN usually works best for small to mid-sized datasets. Moreover, we believed that the dataset has a very limited amount of features which reduces a classifier's ability to learn and makes this dataset very prone to overfitting and underfitting.

Medical diagnoses are incredibly complex and cannot be limited to the 8 features given to determine if a medical condition exists. Just with diabetes, there are two types - type 1 being genetic and type 2 usually being the result of poor diet habits over a period of time. There is also a third "type" of diabetes observed in pregnant females called gestational diabetes, that is the result of hormonal fluctuations reducing the body's ability to process sugars (Center for Disease Control and Prevention, 2020). We do

not have any of this information in this dataset and therefore, with the limited information we have, we believed KNN would be the most appropriate classifier to use.

Result

Confusion Matrix: Out of the four dataframes created, df_original returned the best performance metrics.

Figure 1. Confusion matrix for K-nearest neighbour trained on df_original

Confusion Matrix		Actual	
		Diabetic	Non-diabetic
Predicted	Diabetic	48	12
	Non-diabetic	5	41

Figure 2. Confusion matrix for Logistic Regression trained on df_original

Confusion Matrix		Actual	
		Diabetic	Non-diabetic
Predicted	Diabetic	46	14
	Non-diabetic	7	39

Figure 3. Confusion matrix for Gaussian Naïve Bayes trained on df_original

Confusion Matrix		Actual	
		Diabetic	Non-diabetic
Predicted	Diabetic	42	11
	Non-diabetic	11	42

Table 1. Training and testing evaluation using df_original

	Training				Testing			
	Accuracy	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score
KNN	0.79	0.85	0.76	0.80	0.84	0.91	0.80	0.85
Logistic Regression	0.75	0.78	0.74	0.76	0.80	0.87	0.77	0.81
Naïve Bayes	0.76	0.74	0.77	0.76	0.75	0.72	0.77	0.74

As shown in table 1, the best performing classifiers were KNN and Logistic Regression, which achieved similar recall on the testing data (0.91 and 0.87 respectively). They have also scored relatively high on accuracy, precision and F1-score. KNN generally performed the best to classify diabetic patients. On the other hand, Naïve Bayes struggled to generalise in our current dataset, achieving the lowest recall (0.72) along with the other metrics.

One of our main findings is the poor performance of our current Naïve Bayes classifier. Given that Naïve Bayes classifier classifies an object to class C that maximises the following formula:

$$P(c|x) = P(x|c) \times P(x)$$

Where $P(X)$ refers to the prior probability that an object belongs to a certain class C, whereas $P(x|c)$ is called the likelihood an object with feature vector x given it belongs to class C. In our experiment, we dropped several observations so that the class distribution wasn't skewed towards the positive class. This changed the prior probability to better accommodate the test dataset. However, we speculate that

this change resulted in the poor estimation of the likelihood as the number of available data was reduced. Moreover, Naïve Bayes is based on a naïve assumption that all features are conditionally independent, it therefore indicates that our current feature's dependence on each other might produce the poor estimate. This also reflects the better performance of Logistic Regression as it doesn't assume the conditional independence between features.

We further speculate that the superior performance of Logistic Regression stem from our hyperparameter tuning where "C" was set at 0.001. Since the current dataset contains the moderate number of features, the smaller regularisation strength helped the current model avoid overfitting, resulting in the better performance.

Final Conclusions

This assignment was an incredible opportunity to practise the application of machine learning and artificial intelligence techniques to predict diabetes using the dataset provided. It also gave our group great insight to the incredibly difficult jobs that medical professionals have and just proves how complex medical diagnoses can be, further reinforcing the need to conduct world-class research in an attempt to create a healthier society.

As mentioned in our hypothesis, our group had a challenging time properly fitting the classifiers selected in this assignment. We believe this is due to the limited amount of features available in the dataset and resulted in our group having to fine-tune our models for multiple iterations and one of the primary reasons we used two different methods for missing value handling. Furthermore, we noticed that the training metrics actually scored lower than the testing metrics and changed as we adjusted the train-test split ratio. We determined that the train-test splits are not reliable because the sample size is not large enough.

Our best performing model usually hovered around the 80%-85% accuracy range and we had a challenging time increasing our accuracy without having to overfit the data. Our goal was to achieve an accuracy score of closer to 90% for our best performing model, but also realised that the accuracy score does not tell the big picture of our models.

We came to realise that our confusion matrices gave us a better picture in terms of predictions for medical settings. Our best performing models resulted in very low false negatives which isn't perfect, but ultimately a better indicator of a fine-tuned model. As opposed to false positives, false negatives in a real-world setting would result in a potentially life-threatening condition to be overlooked.

For a future project, we'd like to observe the performances of our models with a more robust and dynamic dataset consisting of many additional features that were lacking in the one used in this project. Doing so would more than likely lead to better feature selections and more accurate predictions regardless of which classifier is used.

Task Allocation

	Michael Ross	Shuhei Ishiwatari
Exploring the data	**x**	**x**
Missing Value: Method 1		**x**
Missing Value: Method 2	**x**	
Feature selection	**x**	**x**
KNN classifier	**x**	
Gausian Naive Bayes		**x**
Logistic Regression	**x**	**x**
data/results visuals	**x**	**x**
Report Writing	**x**	**x**

References

Center for Disease Control and Prevention. (2020, July 14). *Gestational Diabetes and Pregnancy | CDC*. Centers for Disease Control and Prevention. Retrieved March 16, 2022, from https://www.cdc.gov/pregnancy/diabetes-gestational.html