CMSC 353 Final Project: Diagnosing Diabetes in Female Pima Indian Patients using Simple Machine Learning Methods

Alexander Berlaga

May 22, 2023

Abstract

Early detection of Type II diabetes is necessary for reducing patient mortality and complications and improving patient prognoses. Machine learning can help use personal data that one does not need to visit a clinic for to predict whether it likely for that person to have diabetes. This paper compares three applicable supervised classification models in their ability to predict a patient's diabetes diagnosis: K-nearest-neighbors (KNN) classification, logistic regression, and decision tree classification. The "Diabetes Dataset," comprising 768 records of Pima Indian women, is used as training and test data in this paper. The dataset includes various features such as the number of pregnancies, body mass index, blood glucose level, blood insulin level, skin thickness, rate of diabetes in ancestry, age, and blood pressure – both observable data and metrics that one must visit a doctor for. The experiments involve comparing the models using the entire dataset and a modified dataset that excludes variables requiring a doctor's visit (which will be referred to as "observable" data), thus emphasizing the practicality of the proposed system. The models are evaluated based on the percentage of the predictions they got correct, as well as their resulting cross entropy loss.

1 Introduction

Diabetes is a chronic disease that causes the blood to contain excessively large levels of glucose. If left untreated, diabetes can cause major organ failures, amputations, and death. Therefore, early detection of diabetes is essential to begin treatment before it is too late to prevent the onset of these symptoms. However, many people with diabetes are unaware of their condition in the early stages of diabetes. Studies have shown that the time that a patient begins treatment for diabetes is more important for his or her outcome than the intensity of the treatment itself [1]. Meanwhile, it is a challenge to bring seemingly healthy patients into doctors' offices for diabetes screening, and thus a way to detect diabetes from patient data observable without a doctor's visit could be very helpful. This study will attempt to solve this problem, comparing several machine learning models trained on data from 768 Pima Indian females, approximately 25-30% of which had diabetes. We attempt to predict a sample's diabetes diagnosis from all her other data used in this study, as well as isolating only observable data. Several methods are used and compared in order to find an optimal classification algorithm.

2 Related Work

2.1 Diagnostic Machine Learning

Machine learning has been used on medical datasets since its very development [2]. Recently, main developments have been in neural networks because of their improvement in accuracy with larger and larger amounts of data, but traditional shallow methods have also proven success in the diagnoses of several diseases [3]. Algorithms have improved to the point that they can take take X-ray and computed tomography imaging as inputs, and determine a patient's status between the most common diseases. Most recently, given the correct data, ChatGPT was able to pass multiple medical exams and use a description given by a patient to diagnose a number of diseases, including Alzheimer's [4].

2.2 Diabetes Diagnosis Prediction

There are two notable studies in which scientists used machine learning to predict diabetes using observable data: in the first, scientists from the Rochester School of medicine built a "Behavioral Risk Factor Surveillance System" for Type II diabetes. [5] They developed gradient boosting and random forest models that used a total of 17 features including age, sex, race/ethnicity, education level, marital status, smoking status, alcohol consumption, physical activity level, body mass index (BMI), and sleep duration. Surprisingly, these authors discovered that adults who sleep over 9 hours per day were more susceptible to diabetes. Another group of computer scientists from Bangladesh developed a phone app using a chatbot with natural language processing to predict a person's diabetes diagnosis, achieving 90% accuracy with a simple K-nearest-neighbors algorithm [6]. The authors, however, did not elucidate the nature of the questions the chatbot was asking to obtain a high level of accuracy, and it cannot be certain that one would be able to have the information necessary for that chatbot without visiting a doctor.

3 Methods

3.1 Loss Minimization

3.1.1 Cross Entropy Loss

We may compare binary values y to a predicted probability distribution $P(x)$ via the binary cross entropy loss function. This evaluation method iterates over all values in our data set X:

$$XE = \sum_i (y_i \ln P(x_i) + (1 - y_i) \ln(1 - P(x_i)))$$

The reason why both the right side and left side of the sum are necessary is because y values are binary – either zero or one. We must include errors from both positive and negative samples, and thus the y_i factor takes care of the positive samples, while the $(1 - y_i)$ factor takes care of the negative samples.

3.1.2 Fraction Correct

The fraction correct serves as a sanity check for our machine learning models, ensuring the validity of minimizing the cross entropy loss. Minimizing cross entropy loss should also, with the exception of several classes of pathological cases, maximize the fraction of samples predicted correctly. When a model outputs a probability that a certain data point maps to a certain output, we must create a function that transforms this probability into one of the two binary classes. Thus, we define a classification function over the interval [0, 1]:

$$C(\hat{y}) = \lfloor 2\hat{y} \rfloor$$

This ensures that a probability of less than 0.5 will imply a negative classification, while a probability greater than 0.5 will imply a positive classification. The fraction correct is then calculated as

$$1 - \frac{1}{n} \sum_{i=1}^{n} |y_i - C(\hat{y}_i)|.$$

3.1.3 False Negative Rate

The false negative rate is a similar calculation to the fraction correct (though only considering samples classified as positive) and is used specifically to address the problem of class imbalance (which I will refer to in later sections), given that our dataset consists mostly of samples negative for diabetes. The false negative rate is calculated as

$$\frac{\sum_{i=1}^{n} \max\{y_i - C(\hat{y}_i), 0\}}{\sum_{i=1}^{n} y_i}$$

3.2 Classification

3.2.1 K-Nearest-Neighbors Classification

The K-nearest-neighbors classifier is one of the simplest classifiers used in machine learning [7]. A random data point simply is classified as a majority vote of its k nearest neighbors, where k is a hyperparameter. In our implementation of KNN, an array of distances is computed from a query data point to each datapoint in the training set. The smallest k distances are chosen as the neighbors, and the average of all their 0- or 1-labels is returned as a probability. To select properly for the value of k, a portion of the training set is allocated for validation, and every integer value of k between 1 and 24 (inclusive) is tested for cross-entropy loss against this validation set. The value of k for which the validation set cross entropy loss is the lowest is selected as the true k. For our dataset, the k that was selected was 9.

3.2.2 Logistic Regression

Logistic regression is a model used for binary classification where the goal is to predict the probability of an outcome happening. [8] Unlike a line whose range is uniformly distributed, a logistic curve, with few exceptions, transforms data to a value that is very close to 0 or 1. The specific logistic curve is calculated as follows:

$$\hat{\mathbf{y}}_i = \frac{1}{1 + \exp\left[-(\mathbf{w}_i \mathbf{x}_i + \mathbf{b}_i)\right]}$$

where $\mathbf{w}$ and $\mathbf{b}$ are rained by minimizing cross entropy loss. This results in lower error when mapping data to a binary value.

3.2.3 Decision Tree Classification

Decision Trees are literal graph-theory-defined binary trees whose leaves are predicted labels intending to classify data and whose internal nodes include a binary decision based on one of the features. To be classified, a query element enters the decision tree at the root node, and that node contains a binary decision for the query element. Depending on the result, the query element will either go to the right or left child of this node, and will inductively traverse down the tree until reaching a leaf node, which no longer contains a decision but instead classifies the query element as either positive or negative. The model works by creating one branch using several features as options and optimizing for how successfully such a singly branched tree splits the data into buckets correlated with the desired labels, and proceeds to inductively add branches in the same way to the newly created nodes as if they were the root node [9].

3.3 Principal Component Analysis

Principal Component Analysis is used prior to any machine learning in an attempt to reduce the dimensionality of our model and detect patterns or clusters in the data if they exist. PCA achieves this by identifying the directions in our high-dimensional space, called principal components, along which the data varies the most. These components are obtained by finding the orthogonal axes that capture the maximum variance in the data. The first principal component represents the direction of maximum variance, the second represents the direction of the remaining maximum variance orthogonal to the first component, and so on. If the principal component projections of our data are used in the place of the data itself, it is possible that PCA may reduce the noise among the data, as low-variance dimensions likely corresponding to random noise are subtracted out from the dataset.

4 Results

First, PCA was ran on the observable portion of $\mathbf{X}$ to determine there were detectable patterns between the datasets that result in clear categorization. The positive samples were colored red, while the negative samples were colored blue for clarity.

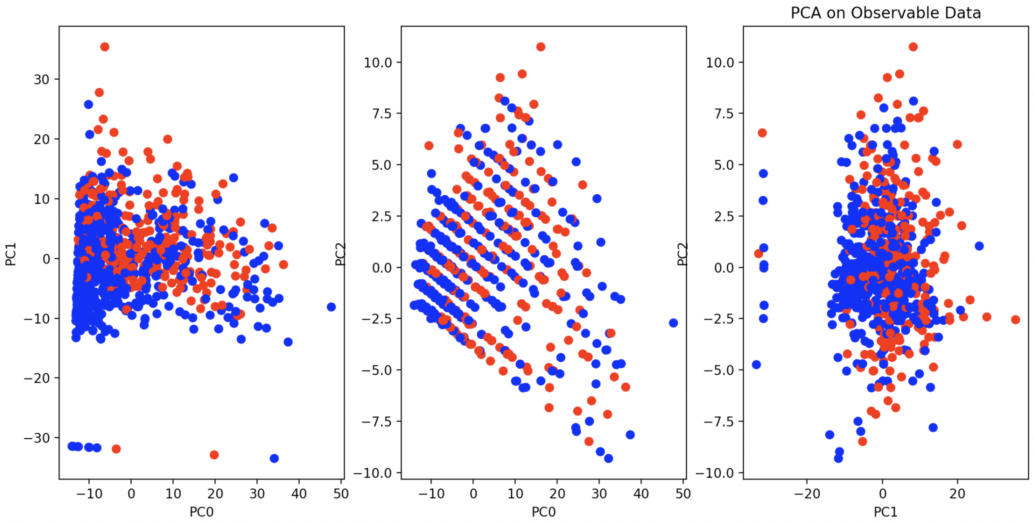
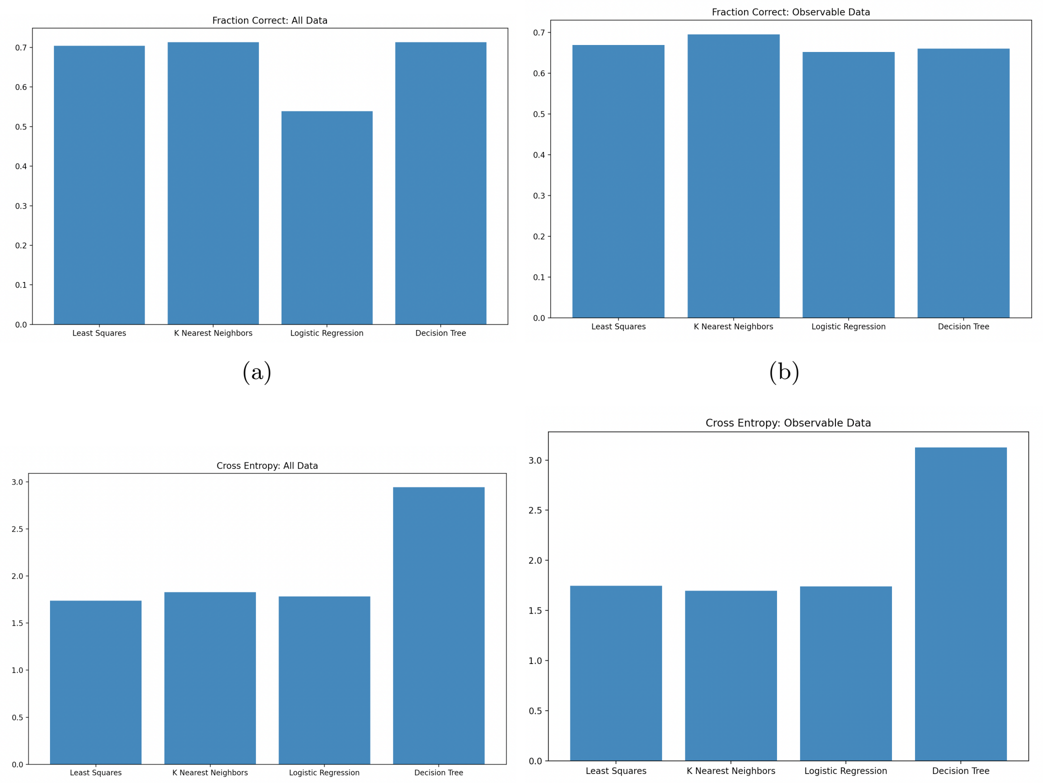


Figure 1: Top three principal components graphed pairwise with diabetes-positive patients labeled red and diabetes-negative patients labeled blue

Unfortunately, though there are general patterns as to which side of the graph is favored by positive or negative samples, no such clear distinction is observed. However, it is important to note that given the nature of the middle plot with distinct, parallel lines displayed, one should note that principal components 0 and 2 are highly correlated, and separated only by some discrete variable (possibly correlated with principal component 1). The final principal component had a singular value less than a tenth of the dominant singular value, and so can be ignored.

It can be easily observed in Figure 2 that there is some form of peak performance which no model exceeds, around a fraction correct of 70% or a cross entropy error of 1.35. It is sadly noted that a prediction of each value as zero (essentially, telling every patient that they do not have diabetes) achieves almost exactly the same scores of error. This highlights a phenomenon called class imbalance that elucidates a problem with our scoring function. Class imbalance occurs when the distribution of classes in the training data is significantly skewed, as it is in our example where about 70-75% of the patients do not have diabetes. This can lead to models that tend to predict the majority class more frequently, resulting in high accuracy but poor performance on the minority class. It may even be the case that for a given model and loss function, the result with the highest percent correctness may be either exactly or very close to assigning every prediction to zero. This indeed proves to be the case, as the false negative rates of every model except for Decision Tree Classification results in a value very close to 1. Thus, it is imperative to choose a loss function that does not allow that to happen. It may then being incumbent to replace the traditional cross entropy loss with a modified cross entropy loss, one that weights false positives more so than false negatives. This is done by multiplying the $y \ln P(x)$ portion of the cross entropy loss by 3, while maintaining the $(1 - y)\ln(1 - P(x))$ portion constant. This gives extra weight to the values in which $y = 1$, or the false negatives and true positives. However, only the KNN and Logistic Regression methods can be modified to minimize this loss function: KNN by scalar-multiplying the resultant $\hat{\mathbf{y}}$, and Logistic Regression by modifying the calculation of the gradient in the gradient descent steps. Figure 3 shows the false negative rates of the resulting models. It can easily be seen that both KNN and Logistic Regression reduce their false negative rates heavily as a result of the change in the loss



(a)

(b)

(c)

(d)

Figure 2: Fractions correct and Cross Entropy losses of the four different models used in this study

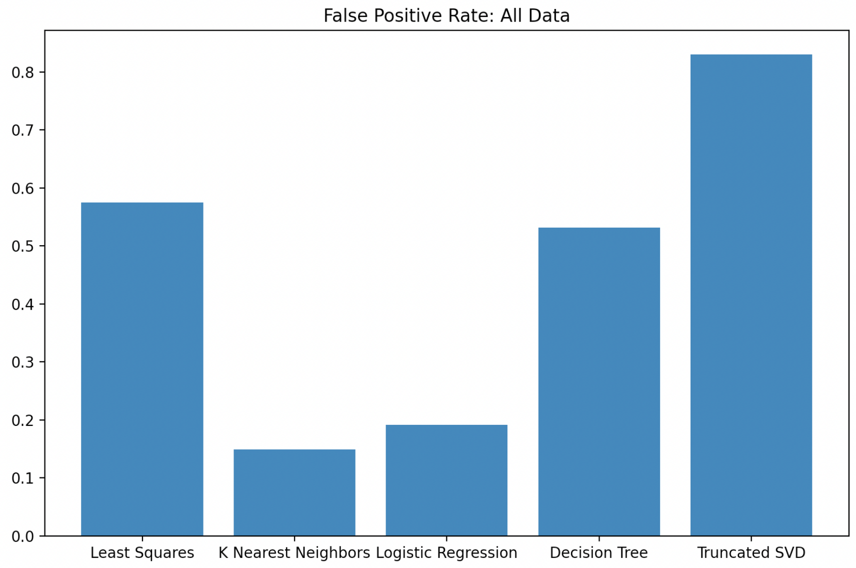


Figure 3: False Negative Rates Using Adjusted Loss Cross Entropy Function

function. However, it should be noted that while KNN's Fraction Correct increases slightly to 0.73, logstic regression has its fraction correct drop slightly to 0.51. It is thus observed that K-nearest-neighbors is the best model using our metrics both for fraction correct and for the false negative rate.

5 Discussion

5.1 Model Practicality

First, we note that the accuracy of every model tried was never higher than 75% (fraction correct), regardless of what loss function we used. While using a loss function that heavily penalizes false negatives successfully identifies the vast majority of diabetes cases using KNN and logistic regression, that is balanced by the large number of false positive samples who are given an incorrect diabetes diagnosis. Though KNN performs better than any other model, it absolutely cannot be used in place of a proper doctor's visit and diagnosis, and therefore this portion of our study is unsuccesful.

However, we note that the accuracy of our models do not decrease if we only use observable data rather than both observable data. Though these models cannot be used in place of a doctor's visit, with our alternative false-negative-penalizing loss function, they can help identify which samples should go on to visit a doctor for further screening.

5.2 Potential Alternative Study

One of the methods chosen in our analysis was Decision Tree Classification. While it was not extremely successful, it did not need an adjustment of the loss function to have a lower false positive rate, and thus similar methods are worth exploring further. The random forests model, essentially an ensemble of decision trees, would be the next step to this study. The data set is small and the interpretability of our model is not of paramount importance, so a random forests model that is bound to be more accurate and less prone to overfitting than the decision tree classifier is a logical next step to take.

References

[1] William H Herman, Wen Ye, Simon J Griffin, Rebecca K Simmons, Melanie J Davies, Kamlesh Khunti, Guy E H M Rutten, Annelli Sandbaek, Torsten Lauritzen, Knut Borch-Johnsen, Morton B Brown, and Nicholas J Wareham. Early detection and treatment of type 2 diabetes reduce cardiovascular morbidity and mortality: A simulation of the results of the anglo-danish-dutch study of intensive treatment in people with screen-detected diabetes in primary care (ADDITION-Europe). *Diabetes Care*, 38(8):1449–1455, August 2015.

[2] Dong Jin Park, Min Woo Park, Homin Lee, Young-Jin Kim, Yeongsic Kim, and Young Hoon Park. Development of machine learning model for diagnostic disease prediction based on laboratory tests. *Scientific reports*, 11(1):1–11, 2021.

[3] Jonathan G. Richens, Ciarán M. Lee, and Saurabh Johri. Improving the accuracy of medical diagnosis with causal machine learning. *Nature Communications*, 11(1):3923, 2020.

[4] Stat News. Ai chatbot chatgpt diagnoses patients with 872023.

[5] Zhiyong Xie, Olga Nikolayeva, Jie Luo, and Dongdong Li. Building risk prediction models for type 2 diabetes using machine learning techniques. *Preventing Chronic Disease*, 16:190109, 2019.

[6] Elias Hossain, Mohammed Alshehri, Sultan Almakdi, Hanan Halawani, Md Mizanur Rahman, Wahidur Rahman, Sabila Al Jannat, Nadim Kaysar, and Shishir Mia. Dm-health app: diabetes diagnosis using machine learning with smartphone. *Computers, Materials & Continua*, 72(1):1713–1746, 2022.

[7] Ahmad Basheer Hassanat, Mohammad Ali Abbadi, Ghada Awad Altarawneh, and Ahmad Ali Alhasanat. Solving the problem of the k parameter in the knn classifier using an ensemble learning approach, 2014.

[8] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[9] Bogumił Kamiński, Michał Jakubczyk, and Przemysław Szufel. A framework for sensitivity analysis of decision trees. *Central European Journal of Operations Research*, 26(1):135–159, Mar 2018.