Collaboratively Trained Diabetes Prediction Project

Ali Burak Ünal

11-15 March 2024

1 Introduction

Diabetes is a chronic metabolic disorder characterized by elevated levels of glucose (sugar) in the blood. It occurs when the body either does not produce enough insulin or is unable to use insulin effectively. Insulin is a hormone produced by the pancreas that helps regulate blood sugar levels by facilitating the uptake of glucose into cells for energy production.

There are several types of diabetes, each with its own underlying causes and risk factors. Type 1 diabetes develops when the immune system mistakenly attacks and destroys the insulin-producing beta cells in the pancreas, leading to a deficiency in insulin production. On the other hand, Type 2 diabetes typically develops due to insulin resistance, where the body's cells become resistant to the effects of insulin, or due to inadequate insulin production by the pancreas. Gestational diabetes occurs during pregnancy and is caused by hormonal changes that affect insulin sensitivity.

Diabetes can lead to various complications if left untreated or poorly managed, including cardiovascular disease, kidney disease, nerve damage, eye problems, and foot ulcers. Therefore, early detection and effective management of diabetes are essential for preventing complications and improving overall health and quality of life.

In this project, we will work on a collaborative diabetes prediction problem. We will be using support vector machines (SVMs) to determine whether a patient has diabetes or not.

2 Dataset

We will use a publicly available diabetes dataset of female patients that you can download on the Ilias page of our practical course. In this dataset, we have 8 features, which are listed as follows:

1. Pregnancies: Number of times pregnant
2. Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. BloodPressure: Diastolic blood pressure (mm Hg)
4. SkinThickness: Triceps skin fold thickness (mm)
5. Insulin: 2-Hour serum insulin (mu U/ml)
6. BMI: Body mass index (weight in kg/(height in m^2))
7. DiabetesPedigreeFunction: Diabetes pedigree function
8. Age: Age (years)

The label or the target of these patients stating whether they have diabetes or not is under *Outcome* column. If it is 0, it indicates that the patient does not have diabetes. In case *Outcome* of a patient being 1, then this patient has diabetes.

In order to use the dataset, we will perform the following preliminary steps:

1. 0s in features might be meaningful in many contexts, but they indicate missing features in our problem. This means that we need to impute those missing features so that we make the dataset usable in SVMs. For every feature **EXCEPT** the target *Outcome* and the feature *Pregnancies*, we will replace 0s with the mean of that feature that is calculated by excluding 0s.
2. Many machine learning algorithms require the features to be in the same range. In our dataset, however, this is not the case. For instance, the *Pregnancies* range from 0 to 17 whereas the *DiabetesPedigreeFunction* has a larger range and values. We will normalize all features (**NOT** the target *Outcome*) so that they all have values in the range from 0 to 1. This means that the minimum value of each feature will be 0 and the maximum of them will be 1.

Once we preprocess the data, we are not ready to split the data between *Alice* and *Bob*. At this point, we split the data row-wise into two parts so that one part will have around 40% of the data and the other part will have the rest. Then, we save them *alice_diabetes.csv* and *bob_diabetes.csv*, respectively.

3 Project Description

We preprocessed the dataset and split it into two parts to mimic the collaborative diabetes prediction problem. We are now ready to work on the collaborative diabetes prediction problem. In this project, we will have **three processes** or **programs**. One of them is for *Alice*. In Alice, we will read the data from *alice_diabetes.csv* file. Another process is for *Bob* and the data of Bob will be read from *bob_diabetes.csv* file. Now, we have two parties with data and it is time to create a party who will use the data of these parties to train a diabetes prediction model in a privacy preserving way. To achieve this, the last process is created for *Charlie*. He will train a diabetes prediction model using the data of Alice and Bob. For simplicity, these processes communicate with each other by writing/reading the required data to/from the files. For instance, when Alice wants to send data to Charlie, she will save the data into a text/CSV file, and then Charlie will read the data from this file.

In the first part of the project, Charlie will train an SVM model on the data of Alice and Bob. For Charlie to train an SVM model, he needs the dot product of the patient data **excluding** the target in the dot product computation. Since the data of Alice and Bob is private, they cannot be shared with other parties. This means that we will use FLAKE to compute the dot product of patient data from Alice and Bob in Charlie. Once the Gram matrix, which indicates the pairwise dot product of patient data, is obtained, we can compute the desired kernel matrix using the entries of the Gram matrix. For instance, we can compute the radial basis function (RBF) or Gaussian kernel matrix using the following formula:

$$k(x, y) = e^{-\gamma\left(\langle x,x\rangle - 2\langle x,y\rangle + \langle y,y\rangle\right)}$$

where x and y are feature vectors, $\langle ., . \rangle$ represents the dot product and γ is the kernel coefficient. Once the kernel matrix is computed, it is time to train a diabetes prediction model. We will set around 20% of the data aside as ***test set***. This data and its corresponding dot product with the rest of the samples will be only used to test how accurate our model is. The remaining 80% of the data forms the ***training set*** and their dot products to each other will be used to train the model.

To train the model, we will feed the Gram matrix containing the dot product of training set patients' data into Sklearn *SVC()* function by setting $C = 1$ (*Hint:* We have already computed the kernel matrix. Therefore, we are giving the data to the model directly. Rather, we will tell the model to use this kernel matrix by specifying *precomputed* option.) Note that we will use the label or the target of data. For simplicity, Alice and Bob will send them directly, that is not considering any privacy issue at all.

After training the model, it is time to test how accurate our model is. To see this, we will use the patient data in the test set. We will use their dot product with the patient data in the training set. We will give these dot products to the function *predict()*. Once we have the predictions, we can report the accuracy of the model!

4 Conclusion and Further Explorations

Congratulations! We managed to train a collaborative diabetes prediction SVM model without sharing the actual data. Since we are already done with the basic version of the model, we can now explore more details of it.

1. Let's start from the end. We only reported the accuracy as the evaluation of our model. Can we also report the following metrics?

    - Confusion matrix
    - Area under the curve (AUC)
    - F1 score

2. How do the evaluation metrics change by changing C? Can we find the optimal C value? (*Hint:* Take a look at cross-validation and grid search.)

3. We used RBF kernel function to compute the similarities between patients. How can we use other kernel functions in our privacy preserving diabetes prediction model? Let's come up with another SVM model using polynomial kernel which is computed using the Gram matrix.

4. So far, we only used SVM. How about other machine learning algorithms that we can train using the Gram matrix that we computed via FLAKE? Can we use k-nearest neighbour (kNN) algorithm in our collaborative privacy preserving diabetes prediction model?