Final Project Report

Diabetes Prediction Using Machine Learning Models

Arpita Chowdhury (30190820), Nushair Imtiaz Riza (30169379), and Nafiz Arman (30171323)

Schulich School of Engineering, University of Calgary

ENEL 682 W2023: Applied Machine Learning and predictive Analysis

Instructor: Leanne Dawson

Teaching Assistants: Mohammad Sahnoon, Niloofar Sharifisadr

April 12, 2023

Contents

1. Introduction

Diabetes is one of the most terrible diseases in the world which has no cure once it reaches a certain stage. If accurate early prediction is achievable, the risk factor and severity of diabetes can be considerably reduced. Therefore, a diabetes prediction technique using machine learning model is proposed to prevent diabetes and the health issues associated with it.

In our project, we have implemented a framework for diabetes prediction by using different Machine Learning (ML) classifiers. We selected Logistic regression (LR), Support Vector machine (SVM), Decision Tree (DT) and Multilayer Perceptron (MLP) classifier for our project.

2. Background on the code file

2.1 Preprocessing

- Importing libraries.
- Read the dataset.
- Find the number of missing values (null values) in each column.
- Data visualization by using:
    - sns histplot() function
    - sns heatmap with help of correlation matrix of a dataset
- Data Splitting
    - 15% of the data has been taken for testing purposes, 85% reserved for training
- Data Scaling
    - StandardScaler() scaler to have Mean of zero, variance of one

2.2 Hyperparameter Tuning

- Define the LR, SV, DT and MLP hyperparameters to tune and their possible values.
- Create a GridSearchCV object with the LR, SV, DT and MLP model and hyperparameters.
- Fit the GridSearchCV object to the training data.

2.3 Model Implementation

- Use the best hyperparameters to fit the LR, SV, DT and MLP model to the training data.

2.4 Validation

- Accuracy of training and testing data for LR, SVM, DT, MLP is shown by accuracy_score() function
- Precision for 4 models is shown by precision_score() function.
- Recall for 4 models is shown by recall_score() function.
- F1-score for 4 models is shown by f1_score() function.
- True positive, false positive, true negetive and true positive values for 4 models are shown with the help of confusion_matrix() function.

2.5 Visualization

- True positive, false positive, true negative and true positive values for 4 model are visualized with the help of confusion_matrix() and heatmap() function.
- Accuracy, precision, recall, f1-score, macro average, weighted average are visualized for 4 models with the help of classification_report() and heatmap() function.
- False positive rate versus true positive rate for 4 models is plotted by roc_curve() function.

2.6 Model performance Comparison

LR, SVM, DT, MLP classification performance comparison is explained with help of:

- TP, FP, FN, TN values.
- Accuracy, precision, recall, f1-score.
- ROC curves.

3. How to run the code

i. Install necessary libraries.
ii. Import necessary libraries.
iii. Initializing google drive on google Colab.
iv. Download the PIMA Indian dataset from https://www.kaggle.com/datasets/johndasilva/diabetes
v. Upload the data file.
vi. Unzip the file.
vii. Read the data.

4. Results

Data was read. The shape and first 5 rows of the dataset are printed.

(2000, 9)

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0

Figure 1: Shape and 5 Rows of the Dataset

Then, the full data statistics are printed to quick overview of the distribution of the data for each feature.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	3.703500	121.182500	69.145500	20.935000	80.254000	32.193000	0.470930	33.090500	0.342000
std	3.306063	32.068636	19.188315	16.103243	111.180534	8.149901	0.323553	11.786423	0.474498
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	63.500000	0.000000	0.000000	27.375000	0.244000	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	40.000000	32.300000	0.376000	29.000000	0.000000
75%	6.000000	141.000000	80.000000	32.000000	130.000000	36.800000	0.624000	40.000000	1.000000
max	17.000000	199.000000	122.000000	110.000000	744.000000	80.600000	2.420000	81.000000	1.000000

Figure 2: Distribution of the Dataset

Next, the number of missing values (null values) in each column is counted. This information can be used to decide how to handle the missing data. We can either remove the rows or columns containing missing values, impute the missing values with some value or use a machine learning algorithm that can handle missing data.

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Figure 3: Number of Missing Values

Next, Here the attributes are visualized using sns histplot() function.

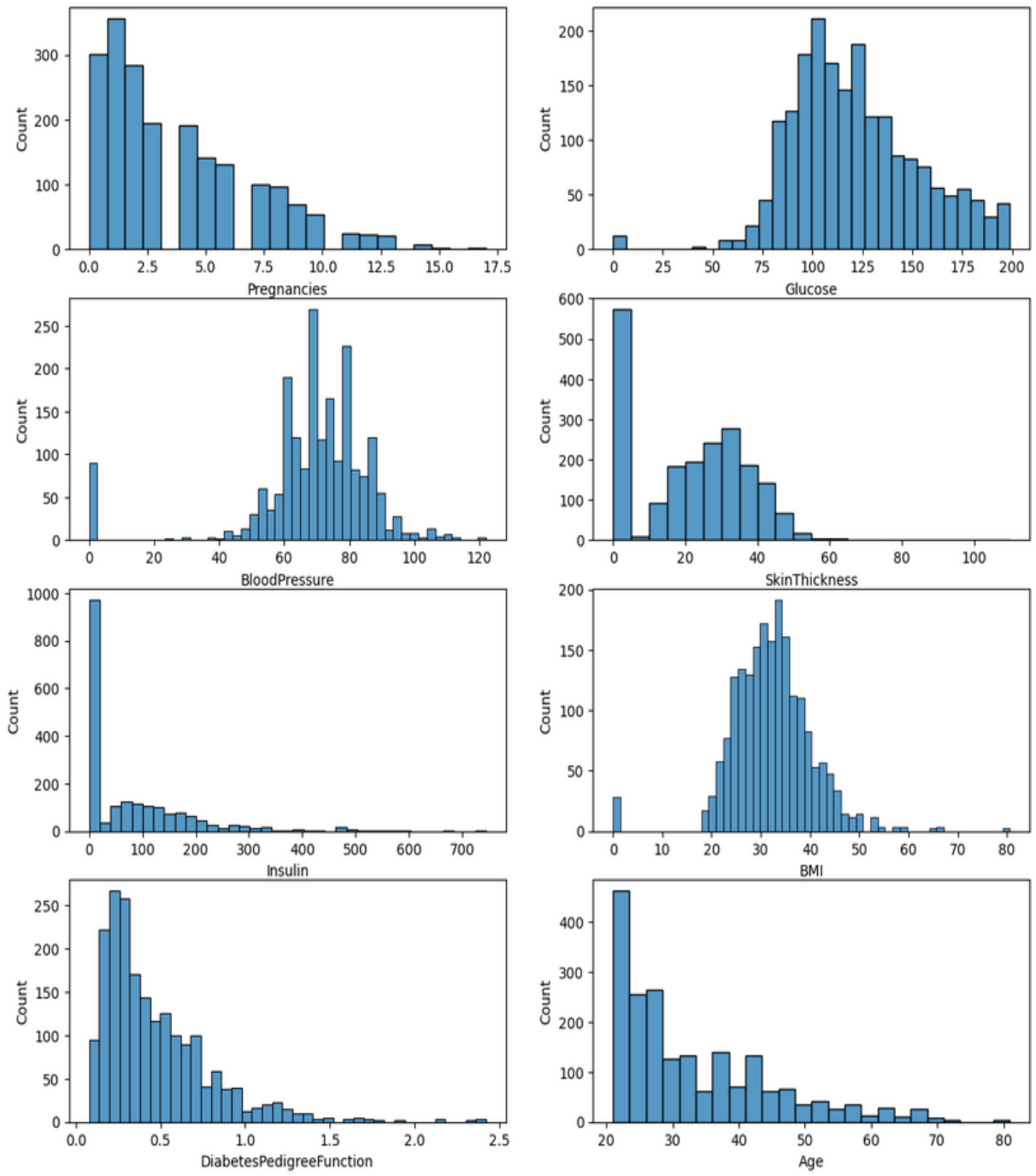

Figure 4: Histogram of Data

After that, correlation between different features in the dataset is visualized, which can be useful for feature selection and understanding the relationships between the features.

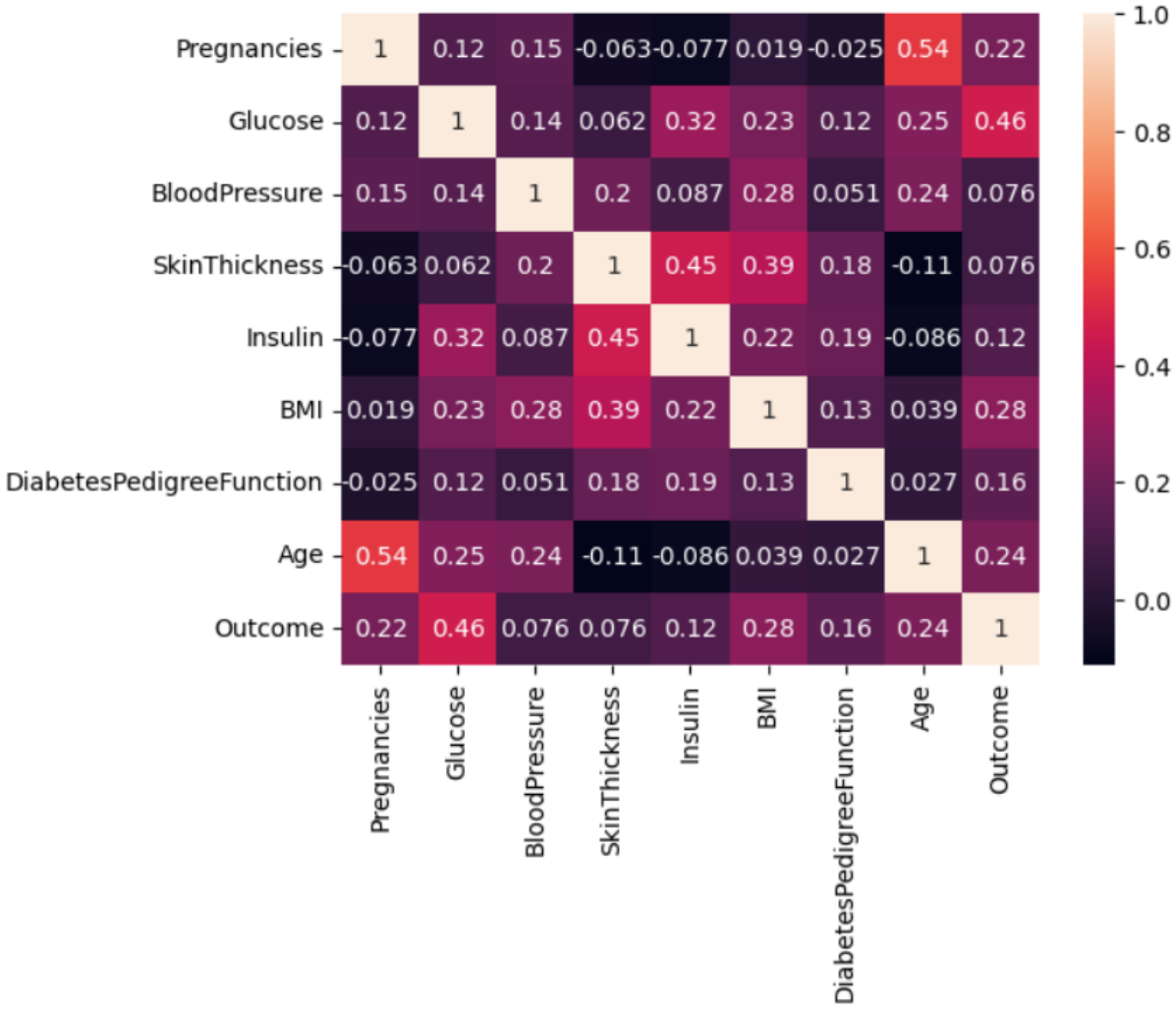

Figure 5: Correlation of Features

Then, the range of each feature is checked whether do we need scaling or not.

```
Pregnancies - min: 0 max: 17
Glucose - min: 0 max: 199
BloodPressure - min: 0 max: 122
SkinThickness - min: 0 max: 110
Insulin - min: 0 max: 744
BMI - min: 0.0 max: 80.6
DiabetesPedigreeFunction - min: 0.078 max: 2.42
Age - min: 21 max: 81
Outcome - min: 0 max: 1
```

Figure 6: Range of Features

As there are a large range of features, to minimize the range of different features we should use scaling. We choose StandardScaler() scaler to have Mean of zero, variance of one.

```
transformed shape: (1700, 8)
per-feature minimum before scaling:
 [ 0.     0.     0.     0.     0.     0.     0.078 21.   ]
per-feature minimum after scaling:
 [-1.12944086 -3.79591081 -3.58472643 -1.29259122 -0.71900226 -3.95202567
 -1.20124498 -1.03552552]
per-feature maximum before scaling:
 [ 17.   199.   122.   110.   744.    80.6    2.42  81.  ]
per-feature maximum after scaling:
 [4.08558404 2.4240003  2.75521509 5.51511896 5.85076724 5.9204371
 5.83819352 4.1344991 ]
```

Figure 7: Range of Features After Scaling

4 different models: LR, SVM, DT and MLP are analyzed and evaluated to check which model gives better classification performance with respect to training accuracy, testing accuracy, precision, recall and f1-score, true positive, false positive, true negative, false negative and ROC curve.

	Model	TP	FP	FN	TN
0	LR	57	21	42	180
1	SVM	89	7	10	194
2	DT	95	11	4	190
3	MLP	83	15	16	186

Figure 8: Model Comparison Based on TP, FP, FN, TN

From Figure-8, we can conclude that Logistic Regression correctly identified 57 patients as having diabetes (true positives) and 180 patients as not having diabetes (true negatives). However, 21 patients were wrongly identified as having the disease (false positives), and 42 patients were wrongly identified as not having the disease (false negatives).

The SVM model correctly predicted that 89 patients had diabetes (true positives) and that 194 patients didn't have diabetes (true negatives), but it got 10 patients wrong (false negatives) and 7 patients wrong (false positives). Whereas the Decision Tree model correctly identified 95 patients who had the disease (true positives) and 190 patients who didn't have the disease (true

negatives). Eleven patients were wrongly identified that had the disease (false positives), and four patients were wrongly identified that they didn't have the disease (false negatives).

On the other hand, MLP identified 83 patients having the disease (true positives). 186 patients were tagged as true negatives, as in they were detected to not have the disease at all. However, the MLP model showcased 15 patients who were wrongly identified to have the disease, whereas they were not suffering from it.

So, based on the confusion matrix, Decision Tree has more true positives, less false positives, and, most importantly, the least false negatives. It appears that the Decision Tree model did a better job of figuring out which people had diabetes. But the SVM model did pretty good too, with only 7 wrong positives and 10 false negatives. However, LR and MLP models aren't as good because they give a lot of false positives. This could be a problem because it could mean that people who actually have diabetes might not get treatment right away..

Analysis of Figure-9 showcases that both SVM and Decision Tree models have the highest training accuracy of 97.5% while the MLP has 91.1% and Logistic Regression has only 78.2% training accuracy. However, when it comes to testing accuracy, Decision Tree (DT) has the highest accuracy of 95% and SVM has 94.3% testing accuracy. While MLP has a testing accuracy of 91.1% and the LR model has the lowest testing accuracy of 78.23%.

	Model	Training Accuracy	Testing Accuracy	Precision	Recall	f1-score
0	LR	0.782353	0.790000	0.730769	0.575758	0.644068
1	SVM	0.975294	0.943333	0.927083	0.898990	0.912821
2	DT	0.975294	0.950000	0.896226	0.959596	0.926829
3	MLP	0.911176	0.896667	0.846939	0.838384	0.842640

Figure 9: Model Comparison Based on Accuracy, Precision, Recall and f1-score

Based on the given performance metrics, it appears that the Decision Tree (DT) model has the best classification performance in terms of testing accuracy, precision, recall, and f1-score and the SVM model has slightly lower performance in terms of precision, recall, and f1-score. The LR model, on the other hand, has the lowest performance metrics in all categories.

Next, performance is evaluated by ROC curve. From Figure-10, Decision Tree (DT) has more data points that are close to the top left. So, it produces a high recall while keeping a low false positive rate. So, Decision Tree (DT) has better classification performance than the other three models.

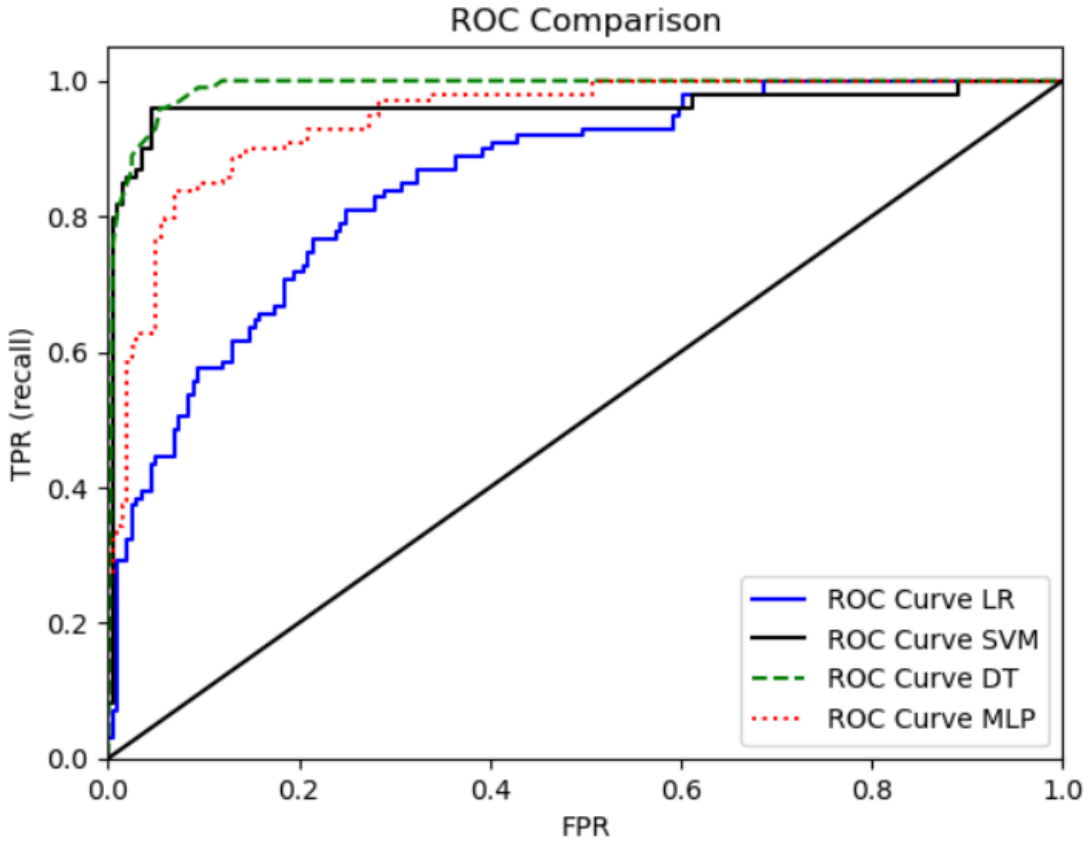

Figure 10: Model Comparison Based on ROC-curve

5. Interpretation

The diabetes dataset was obtained from kaggale, as per our project proposal. Scaling the original data is done during the preprocessing procedure. After preprocessing, four different ML classification models—Logistic regression (LR), Support vector machine (SVM), Decision tree (DT), and Multilayer Perceptron (MLP)—are used for training and hyperparameter tuning using gridsearchCV. After training, validation for the testing or validation dataset will be carried out using performance metrics including accuracy, recall, and precision. Additionally, the f1-score and ROC curve will be used to assess the model's effectiveness. Model comparison will then be assessed as a potential fix for this problem.

We conducted our research as outlined in the project proposal. In order to reach a more precise result, we used an alternate dataset with similar attributes but a much larger sample size. Our project now has a higher degree of precision than the previous three models combined through the implementation of a new model decision tree (DT). However, we did not use PCA in preprocessing because executing this significantly reduces classification accuracy.

6. Conclusion

Machine learning and predictive analytics in healthcare plays a vital role in today's medical science. In this project, we used four popular machine learning algorithms (LR, SVM, DT, MLP) for this analysis. Predictions were made about diabetes on PIMA Indian dataset consisting of 2000 records. 8 attributes were selected for training and testing the model. From the experimental results obtained from the chosen dataset, it can be observed that DT gives the highest accuracy for predicting diabetes. This algorithm provides 95% accuracy which is highest as compared to other three algorithms used in this project. Therefore, based on the current dataset, DT is appropriate for predicting diabetes disease.