Group 21

Ömer Faruk AYDIN
Artificial Intelligence and Data Engineering
Faculty of Computer and Informatics
Istanbul, Turkey
aydinome21@itu.edu.tr

Emre Hakan ERDEMİR
Artificial Intelligence and Data Engineering
Faculty of Computer and Informatics
Istanbul, Turkey
erdemire21@itu.edu.tr

***Abstract*—Diabetes, a metabolic disorder influenced by both genetic and environmental factors, poses a significant global health challenge. This project focuses on leveraging machine learning techniques for predicting diabetes using the Pima Indian Diabetes Database, collected from 768 Pima Indian women known for their higher susceptibility to diabetes. Our goal is to contribute to the early diagnosis and treatment of diabetes.**

This project will be exploring data and making use of preprocessing, including handling missing values, outlier detection, and feature normalization. Multiple machine learning models, Decision Trees, Logistic Regression , Random Forests and Multi-Layer Perceptrons are employed in the project. And the results show that by employing machine learning algorithms, we can get pretty high accuracy, precision and recall using these methods.

I. Introduction

Diabetes is a metabolic disorder that is usually caused by a combination of inherited and environmental factors and results in excessively high blood glucose levels (hyperglycemia). The most two common types of diabetes are type 1 diabetes which happens when the immune system attacks its own cells which are responsible for producing insulin in the pancreas and type 2 diabetes which pancreas doesn't produce enough insulin and the body becomes resistant to insulin. There is also a less common diabetes variant which is gestational diabetes which happens during pregnancy .Millions of people around the world efected by diabetes every year. Diabetes itself and the treatment methods used in diabetes can lead to many complications. The main ones of these complications are; Circulatory system (cardiovascular) diseases, chronic kidney failure, retinal damage that can cause blindness (retinopathy), and various types of nerve damage. According to the World Health Organization, diabetes was the seventh leading killer in 2016 and is predicted to become the sixth leading killer in 2030 [1]. Therefore, diabetes and its causative factors need to be investigated, effectively treated and prevented.

In our project, we will use the Pima Indian Diabetes Database, which was created with data collected from 768 Pima Indian women. Pima Indians are a Native American race living in Arizona, USA. and they are more prone to diabetes than other human races due to factors such as genetics, environmental and lifestyle. The source of this dataset is National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) [2]. Also this dataset was was used in a study by Smith et al [3].

We will preprocess our data which is very important for machine learning algorithms to be able to work proberly. It involves handling missing or null values in the dataset, handling outliers that might distrupt the models performance by normalizing or standardizing features to bring them to a similar scale. This ensures that the models can learn patterns from the data more effectively and that a small number of features doesnt distrupt the entire algorithm.

Numerous methods for diabetes prediction have been published in recent years.

Our goal is to enhance the predictive accuracy and also enhance the interpretability of diabetes diagnosis which previous work on this field doesn't always do good enough. By using visualization tools and feature importance analysis, our goal is to provide transparent insights into the decision-making process of models such as Decision Trees and Logistic Regression which are highly interpretable. This not only helps healthcare professionals to comprehend the factors influencing predictions very much but also helps patients to actively engage in understanding their risk factors. We will also use less interpretable, more complex methods such as Random Forests and Multi-Layer Perceptrons as well that can handle more complex, high dimensional data better to provide more accurate results .

Basically what we will do in this project is, exploring data, applying preprocessing steps to data and training machine learning models that can predict a human will develop diabetes or not. And also which parameters most affect the risk of getting diabetes. At the the end of this project, we will compare the performance and accuracy of models we have trained.

We hope that the results of this project will contribute to processes such as early diagnosis and treatment of the chronic disease diabetes. We believe that the results we will obtain will also benefit the authorities in the field of medicine and technology.

II. Related Work

In recent years a lot of work has been published by numerous scientists. One significant research in this area was conducted by Huma Naz & Sachin Ahuja, who proposed a diverse set of machine learning algorithms, including Artificial Neural Network (ANN), Naive Bayes (NB), and Decision Tree's[9]. Another work was published by Tita et al.[10]. The performance of their algorithms was validated using

multiple metrics such as accuracy, precision, recall, and f-score. These measures provide a comprehensive evaluation of the ML models' performance in terms of both their ability to correctly classify instances and their robustness to errors.The results obtained from this study showed that among all the ML algorithms used, Logistic regression provided 76% accuracy while Random Forest algorithm provided 72% accuracy, and xgboost with maximum accuracy of 82%.

III. Proposed Work

A. Dataset

The dataset we will use for this project is the Pima Indians Diabetes Database, which is available on Kaggle [4]. This dataset contains information about 768 women from a population of Pima Indians. The Pima Indians are known to be more prone to diabetes than other human races due to factors such as genetics, the environment they live in and their lifestyle.

The source of this dataset is National Institute of Diabetes and Digestive and Kidney Disease (NIDDK) [2]. The dataset aims that predict the patient has diabetes or not in according to some diagnostic measurements. The dataset only contains instances that meet certain criteria. Specifically, all the patients are female, at least 21 years old, and belong to the Pima Indian heritage.

The dataset consists of 768 rows and 9 columns, where each row represents a patient and each column represents a variable. The variables are as follows:

- **Pregnancies**: Number of times pregnant
- **Glucose**: Plasma glucose concentration 2 hours in an oral glucose tolerance test
- **BloodPressure**: Diastolic blood pressure (mm Hg)
- **SkinThickness**: Triceps skin fold thickness (mm)
- **Insulin**: 2-Hour serum insulin (μU/ml)
- **BMI**: Body mass index (weight in kg/(height in m)^2)
- **DiabetesPedigreeFunction**: Diabetes pedigree function
- **Age**: Age (years)
- **Outcome**: Class variable (0 or 1) indicating whether the patient has diabetes or not

The dataset used for this project is in CSV format, with a size of 23.4 KB.

Preprocessing

We have checked the data for null values and there doesn't exist any null values in this dataset so we skipped the filling null values part. However when we really looked into the data there were an abundance of 0 values where they need to not be. For example when we checked the Insulin column we have found 374 0 values. This should not be a thing of course as this would mean the person would be dead. So we filled the columns of the 0 values with the median based on their outcome. This does not apply to columns outcome and pregnancies of course since they can be 0 naturally. We have split the dataset into training and testing sets with training data containing 80% of the data and the test data containing 20% which is widely regarded as the optimal rate to split the data. Each of the features has outliers, instead of removing them completely we decided to normalize them using robust scaling method.

$$\text{Outlier Range} = [Q_1 - 1.5 \times \text{IQR},\, Q_3 + 1.5 \times \text{IQR}]$$

$$X_{\text{scaled}} = \frac{X - \text{median}(X)}{Q_3(X) - Q_1(X)}$$

Potential Challenges

- Our dataset is prone to facing many challenges. The dataset is relatively small and imbalanced with 500 '1' values and 268 '0' values, which impacted the performance and generalization of our models.
- Our dataset may not be representative of other populations or groups with different characteristics or risk factors, this can affect the solution we are offering which is to detect diabetes early since we cannot be sure that the characteristic of these people in the dataset will accurately represent all humans.
- One last challenge is that this dataset may not have captured all the variables that effects diabetes or is related to developing diabetes.

B. Methods

In order to predict diabetes using our dataset we will be using multiple machine learning methods and techniques with each method having its unique strength. We will start by splitting the data into training and testing sets in order to evaluate the performance of the method used in the end so we can compare them. This split allows us to use our models finding on new, unseen data so that we can make sure it doesn't simply memorize the data. First method we will be using is decision trees. As the name suggest they are a tree like structure in which the nodes are attributes and the branches lead to an outcome or to another attribute. They work by splitting the data using the data's feature's that are most informational until there is no more information to process or a threshold is met. They are easy to interpret meaning we can understand how they made the decision easily which is important in our project which is diabetes prediction. But they have a high variance and are prone to overfitting and sensitive to outliers. We can use random forest to overcome these problems. Random forest contains multiple decision trees where each decision tree is based on a part of the training data. After each tree makes a decision, which in our case will either be 0 (no diabetes) or 1 (diabetes), the decision with the most votes wins. This method has a lower variance, is less prone to overfitting, is less sensitive to outliers and is better for more complex data which for our case is crucial because of the multilayered nature of diabetes. However it is slower and it is harder to understand how a decision is made as it contains multiple trees as opposed to a single tree . Logistic regression is another method we will use. It is another classification method which predicts the probability of the outcome being two possible classes which is suitable for our data. It will come to a conclusion by estimating the coefficients for the

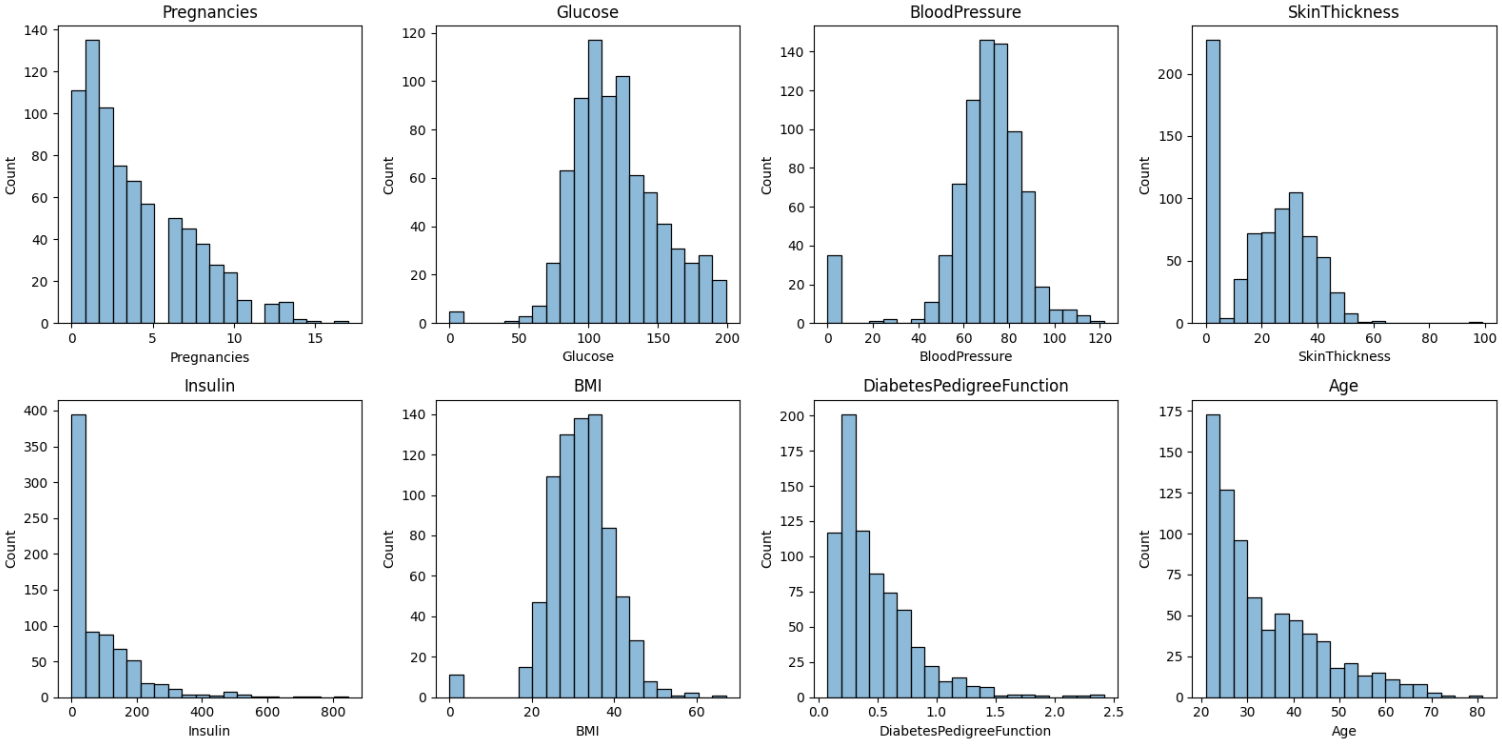

Fig. 1. Distribution of each feature in the dataset

given features and the combination of the given features and coefficient to estimate the probability of belonging in one of the classes (1 or 0 in our case). This method will also help us evaluate the other, more complex methods we will use. Final method we plan to use is Multilayer Perceptron (MLP) which will help us get a deeper dive into the complexities to find non-linear relationships in the data if there are any.

Decision Tree Method

Each split at a decision tree is created by finding the best split using Information Gain.

To find Information Gain we must first calculate the Entropy

$$Entropy(X) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

where p_i is the proportion of samples belonging to class i in the dataset S.

Now that we have Entropy we can go on to calculate Information Gain

$$IG(S, A) = H(S) - \sum_{values(A)} \frac{S_v}{S} H(S_v)$$

where A is a feature Entropy is $H(S)$, $values(A)$ are the unique values of feature A, S_v is the subset of samples where feature A has value v, and S is the total number of samples in S.

We iterate over each feature and each unique value of that feature to find the best split. For each feature value pair the dataset is divided into left data and right data. If the Information Gain is greater than previous Information Gains obtained then the current split becomes the best split. In the end the best split is chosen. This step is recursively applied for each left node and each right node until a stopping criteria is met which is maximum depth in our case.

Logistic Regression

Another classification algorithm we will use in our project is Logistic Regression. Logistic regression is generally used for binary (positive / negative) classification. In logistic regression, the logistic function or sigmoid function is used. It has an S-shaped curve.

The mathematical definition of the logistic function is as follows.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Where x is the input value, e is the base of the natural logarithm and $f(x)$ is the output value. The linear equation of logistic regression is as follows:

$$z = b_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

z is the linear combination of input values, b_0 is the bias term and $w_1, w_2, \ldots, w_n$ are the weights for each input value.

$$y = f(z) = \frac{1}{1 + e^{-z}}$$

The value obtained after putting z into this function is a value between 0 and 1. If this output value is greater than

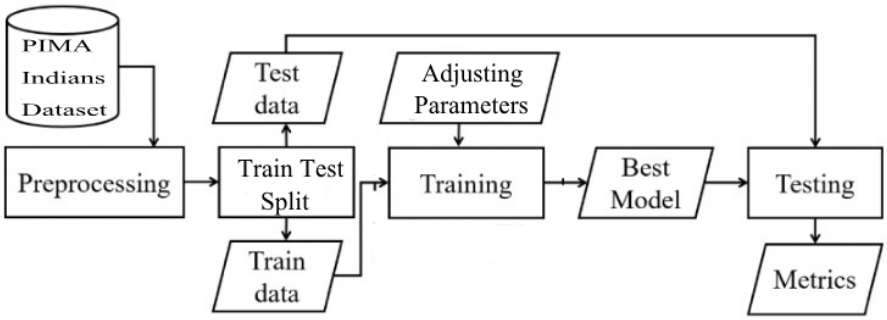

Fig. 2. Theoretical Framework

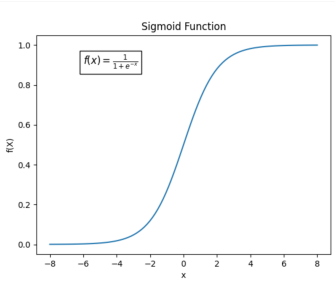

Fig. 3. Sigmoid Function

0.5, the given sample is classified as 1, and if it is less than 0.5, it is classified as 0. After this classification, the output

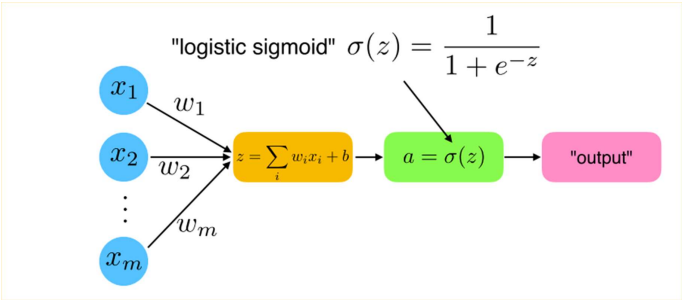

Fig. 4. Architecture of Logistic Regression

values given by the function are compared with the actual values and this cycle is repeated in the backward propagation stage by updating the weight values using various techniques such as gradient descent, maximum likelihood estimation or regularization.

Multilayer Perceptron

What is Perceptron: The perceptron is the most basic building block used in artificial neural networks. It can be thought of as a single neuron used for binary classification. It basically consists of three parts. Input layer, net input layer and activation layer. In the input layer, input data and weights are taken, in the net input layer, input data and weights are multiplied and summed with bias, in the activation layer, the value from the net input layer is put into the activation function and transferred to the output layer. Various activation functions

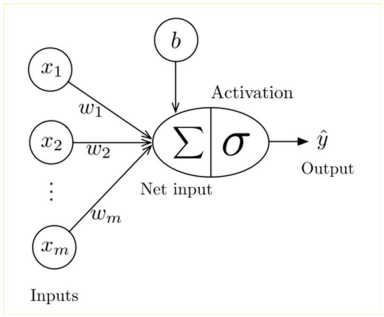

Fig. 5. Perceptron

can be used in the activation layer. Sigmoid, tanh and relu are examples of these functions.

- Logistic function: $f(x) = \frac{1}{1+e^{-x}}$. This function maps the input to a value between 0 and 1.
- Hyperbolic tangent function: $f(x) = \tanh(x)$. This function maps the input to a value between -1 and 1.
- Rectified linear unit function: $f(x) = \max(0, x)$. This function sets the negative inputs to zero and keeps the positive inputs unchanged.

Architecture of Multilayer Perceptron: Multilayer Perceptron can be thought of as a structure consisting of multiple perceptrons connected to each other. Data from the activation layer of one perceptron is passed to the input layer of the other perceptron. MLP consists of three basic layers. Input layer, hidden layer and output layer. The input layer is the

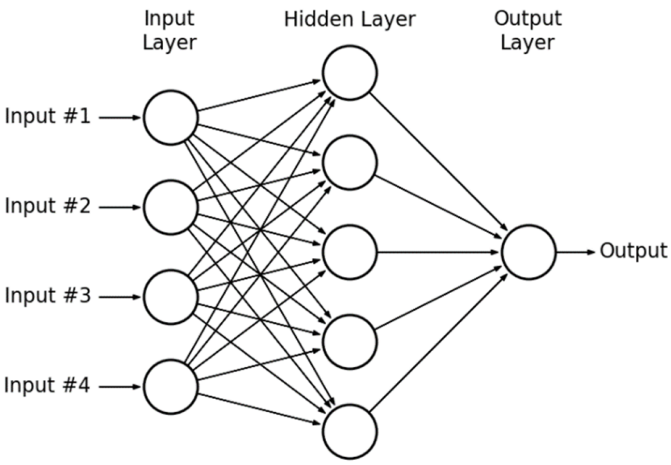

Fig. 6. Architecture of Multilayer Perceptron

input data is received. The output layer is the layer where the prediction data is generated. Hidden layer is the layer where the input data is multiplied by the weights and summed by the bias and put into the activation function to learn the paterns in the data.

In the output layer(last layer) the output data we find is compared with the correct results. At this stage, weights and bias are cjanged by minimizing the loss function using optimization algorithms we will further explain in this paper. It is forwarded to the relevant layers in the hidden layer with backpropagation. So the model learns the complex patterns in the data and provides better results.

The most common algorithms used in the optimization process:

- SGD (Stochastic Gradient Descent): This optimizer updates the weights by taking small steps in the opposite direction of the gradient of the loss function.
- Adam (Adaptive Moment Estimation): This optimizer combines the advantages of momentum and RMSProp. It uses an adaptive learning rate for each weight, and it also uses an exponential moving average of the gradients and the squared gradients.

Evaluation methods

To evaluate our diabetes prediction, we will use a set of evaluation metrics that serve as numerical measures of model performance and effectiveness. The metrics to measure our solution's performance include accuracy, precision, recall, F1-score. These metrics helps us see our models ability to correctly classify sick people with and without diabetes. Accuracy provides an overall evaluation of the model's correctness in its predictions. Precision focuses on the ability to minimize false positive predictions, which is crucial in our medical context to avoid unnecessary alarm. Recall is the ratio of true positives to all positives, it is important for this metric to be high to ensure that there are as many positive cases identified as possible. The f1 score is the mean of precision and recall will help us find a balance between them. We will also compare each model with each other.

IV. Experimental Methods

Till this point we were always talking about hypotheses and what we plan on should happen. Now we can see some experimental results of our algorithm's and how they perform with the metrics we proposed. Before we start getting into algorithms it should be noted that these algorithms are still in development and the results are expected to improve from this stage.

A. Decision Trees (Emre Hakan Erdemir)

First method we implemented is Decision Tree's.

After the hard part, which is the tree creation is over, we go on to traverse the tree starting from the root to find our predicted Outcome based on our features. For example for the tree in Fig. 3. if we have Glucose levels more than 154, BMI less than or equal to 28.5 and blood pressure is more than 86 we would make the guess that the person is not diabetic. This method is highly interpretable and anyone can do the traversing we just explained by just looking at the tree plot. We found this confusion matrix with the Decision Tree algorithm.

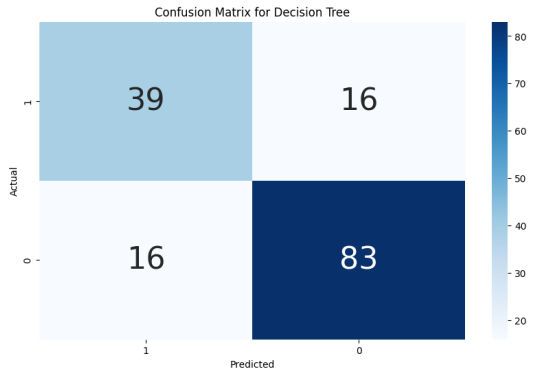

Fig. 7. Decision Tree confusion matrix

With the Decision Tree algorithm we achieved the confusion matrix in Fig. 8. We can use this matrix to calculate our evaluation metrics which are mentioned earlier.

- Accuracy: 0.7922
- Precision: 0.7090
- Recall: 0.7090
- F1 Score: 0.7090

With these metrics we can see that we have an overall balanced decision making process with our algorithm with an excellent true negative detection and decent true positive detection.

B. Random Forest (Emre Hakan Erdemir)

Our Random Forest algorithm uses multiple Decision Tree's as basically building blocks. These multiple decision blocks are trained on different random subsets of the training data. For each tree, a random subset of the original dataset is sampled

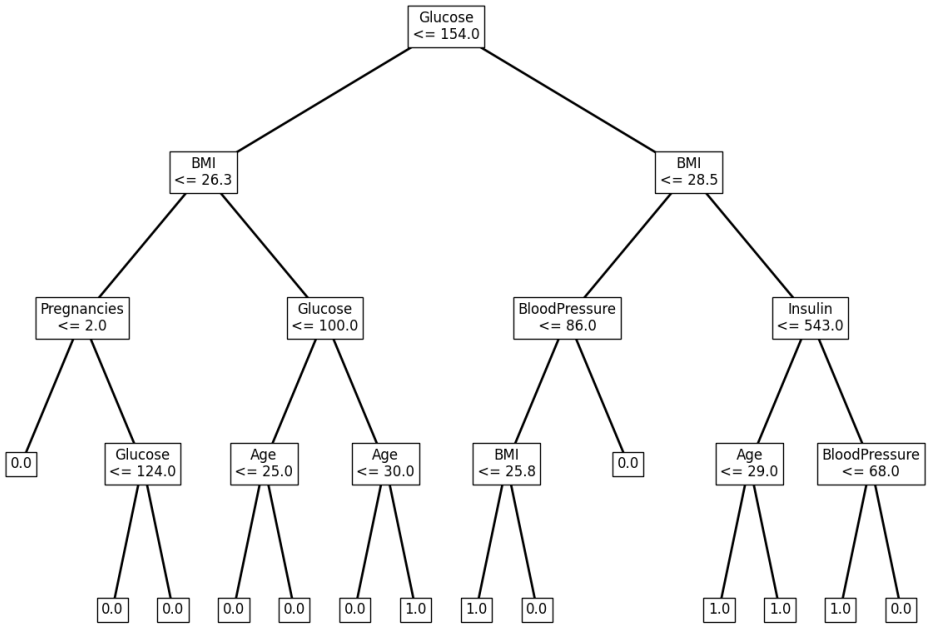

Fig. 8. Best performing decision tree so far

with replacement. This means that there are points which are going to be repeated in the same subset and if the number of trees are low enough some data points may not be included at all while creating the trees. These subsets also consider a few features out of all the features. (out of 8 features in our case). In the end however many decision trees we have created gets to vote on an outcome. The outcome is selected using majority voting. This method is less sensitive to outliers and reduces overfitting. We found this confusion matrix with the Random Forest algorithm.

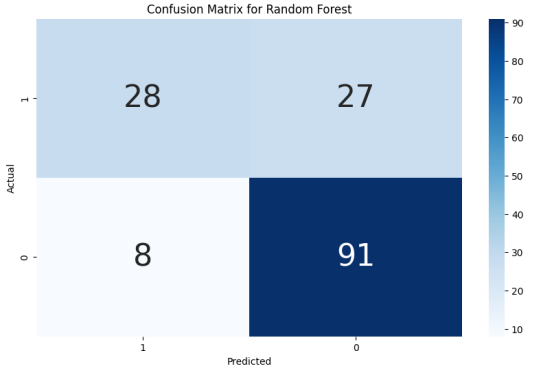

Fig. 9. Random Forest Confusion Matrix

We can again use this matrix to calculate our evaluation metrics.

- Accuracy: 0.7727
- Precision: 0.7777
- Recall: 0.5090
- F1 Score: 0.6153

We can see from this confusion matrix that the algorithm is classifying most of the data as 0. This is because the dataset has two times the number of 0's as 1's and with each tree created it becomes more likely to choose 0's instead of 1's. All these show us that this model is great at predicting True Negatives but terrible at identifying true positives.

C. Logistic Regression (Ömer Faruk Aydın)

At this part, we created a Logistic Regression class and train the model with 1000 iteration and 0.0001 as learning rate.

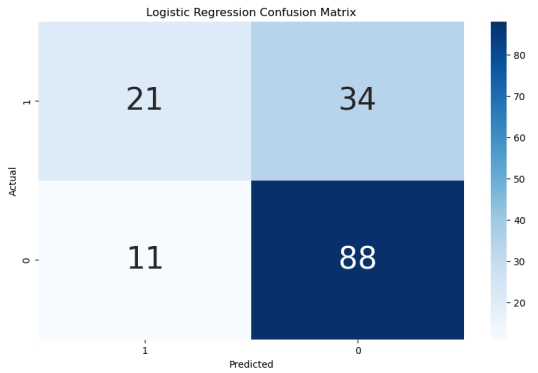

Fig. 10. Logistic Regression Confusion Matrix

We can again use this matrix to calculate our evaluation metrics.

- Accuracy: 0.7077
- Precision: 0.65625
- Recall: 0.3818
- F1: 0.4827

From our findings, it can be concluded that the model reached an accuracy of 70%, correctly predicting 65% of positive cases but correctly identifying 38% of true positive samples. This value is too low to be considered a good model. The F1 score of 0.48 indicates that the Logistic Regression does not strike an adequate balance between recall and precision.

D. Multilayer Perceptron (Ömer Faruk Aydın)

As mentioned in the previous section, the MLP model can be run with various algorithmic combinations. Different results can be obtained with various activation functions (relu, sigmoid, tanh), various optimizers (adam, sgd) and various learning rate values.

Activation Function	*Optimizer*	*Learning Rate*	*Accuracy*	*Precision*	*Recall*	*F1 Score*
sigmoid	adam	0.01	0.7857	0.7391	0.6182	0.6733
sigmoid	adam	0.03	0.7208	0.6071	0.6182	0.6126
sigmoid	sgd	0.01	0.6818	0.65	0.2364	0.3467
sigmoid	sgd	0.03	0.6429	0.5	0.0364	0.0678
relu	adam	0.01	0.7857	0.75	0.6	0.6667
relu	adam	0.03	0.8051	0.7450	0.6909	0.7169
relu	sgd	0.01	0.7013	0.5957	0.5091	0.549
relu	sgd	0.03	0.6494	0.5556	0.0909	0.1563
tanh	adam	0.01	0.7987	0.8158	0.5636	0.6667
tanh	adam	0.03	0.8052	0.7551	0.6727	0.7115
tanh	sgd	0.01	0.6818	0.75	0.1636	0.2687
tanh	sgd	0.03	0.6429	0.5	0.3455	0.4086

TABLE I
METRICS FOR DIFFERENT CONFIGURATIONS

The values given in the table are calculated with 1000 iterations and as can be seen, the most effective combinations on the dataset we used are optimizer: adam - activation: relu - learning rate: 0.03 and optimizer: adam - activation: tanh - learning rate: 0.03. At the same time, the loss values of the MLP function we used during training are also visualized.

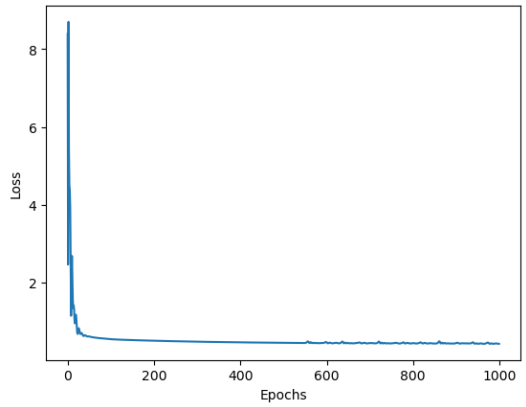

Fig. 11. Loss Curve for Multilayer Perceptron

The outputs of the model trained with the combination adam - relu - 0.03 are as shown in the figure.

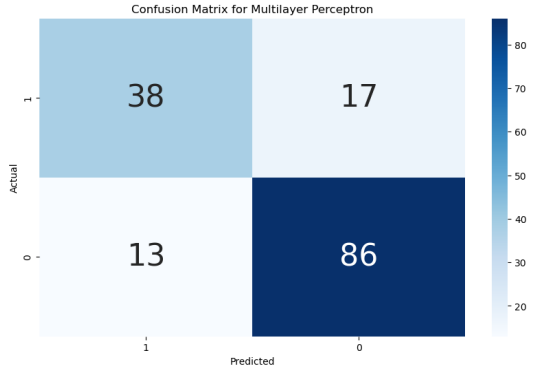

Fig. 12. Confusion Matrix for Multilayer Perceptron

We can again use this matrix to calculate our evaluation metrics.

- Accuracy: 0.8051
- Precision: 0.7450
- Recall: 0.6909
- F1: 0.7169

Based on the results we obtained, we can say that the model can make 80% correct predictions. The model correctly predicted 74.5% of the positive cases and correctly predicted 69.1% of the actual positive cases. 0.717 F1 score means that MLP is able to achieve a good balance between recall and precision.

Drawing from these outcomes, it can be asserted that the multilayer perceptron serves as a reasonably effective classifier; nonetheless, enhancements are possible by mitigating false positives and false negatives. This improvement can be pursued through the optimization of hyperparameters, including but not limited to, adjusting the number of hidden layers, selecting an appropriate activation function, fine-tuning the learning rate, and so forth.

V. FINALIZED CHANGES / RESULTS

Previously we mentioned the experimental results which were not finalized as mentioned before. Since then we have made numerous changes in the preprocessing steps which resulted our measures to improve by approximately 15 percent. For example the decision tree classifier gave 0.79 accuracy and 0.70 recall. After the preprocessing steps this number increased to 0.89 for both accuracy and recall while precision increased from 0.70 to 0.8. Let's see the change and increase in all the models.

A. Decision Trees

Decision Tree algorithm.

- Accuracy: 0.8831
- Precision: 0.8032
- Recall: 0.8909
- F1 Score: 0.8448

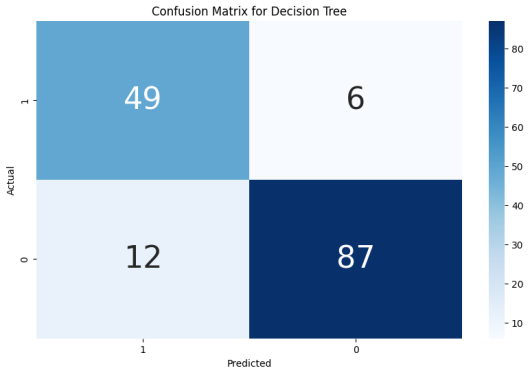

Fig. 13. Decision Tree confusion matrix

We can see a great improvement in recall and f1 measure especially after the change in preprocessing steps. We will discuss which measures matter more in our context in the section final discussion later.

We got these changes in the measures, however unlike the experimental decision tree which gave the best result in height 3, these results are obtained from a height 4 decision tree. The height 3 also improved but couldn't beat height 4 in these measures unlike before.

B. Random Forest

Our random Forest algorithm also saw great improvements

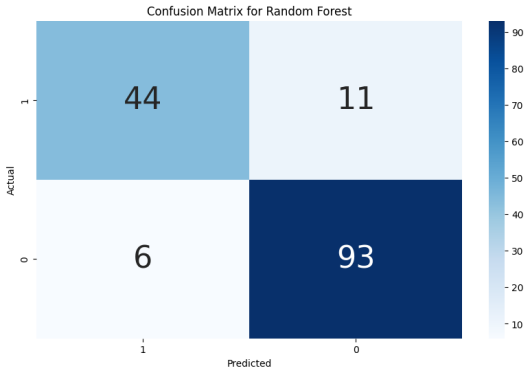

Fig. 14. Random Forest Confusion Matrix

- Accuracy: 0.8896
- Precision: 0.88
- Recall: 0.8
- F1 Score: 0.8380

We can see the same improvement in accuracy as decision trees, however the precision measure and the recall measure seem to have swapped places.

C. Logistic Regression

Since the custom logistic regression method we used in the previous study did not provide sufficient results (previous accuracy was %70), we decided to use the Sklearn Logistic Regression Model, which is a more optimized model.

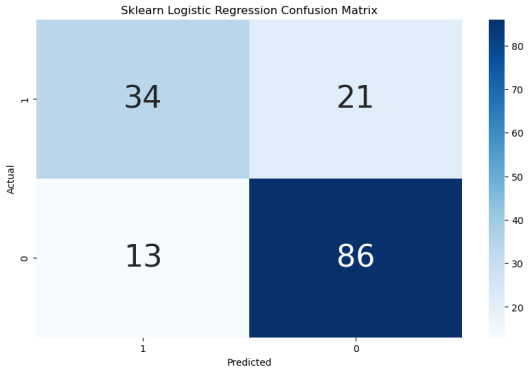

Fig. 15. Logistic Regression Confusion Matrix

We can again use this matrix to calculate our evaluation metrics.

- Accuracy: 0.7792
- Precision: 0.723
- Recall: 0.6181
- F1: 0.6666

As can be seen, there is a 7% increase in accuracy compared to the previous study, as well as a noticeable increase in precision, recall and f1 score values. The improvements show that the sklearn model is much more effective and reliable than the model in the first study.

D. Multilayer Perceptron

We made some changes to the Multilayer Perceptron structure we used in the previous study and added a softmax layer to the already existing structure of the first linear layer, activation layer and second linear layer.

The softmax layer is usually the last layer in a classification task. The main purpose of the softmax layer is to transform the output of the network into a probability distribution so that a probability value is obtained for each class. This makes it easier to interpret the model's predictions. After the preprocessing and the addition of this layer, we saw significant changes in our results in TABLE II.

The values given in the table are calculated with 1000 iterations and as can be seen, the most effective combinations on the dataset we used are optimizer: adam - activation: relu - learning rate: 0.03 and optimizer: adam - activation: relu - learning rate: 0.01. At the same time, the loss values of the MLP function we used during training are also visualized 16.

The outputs of the model trained with the combination adam - relu - 0.03 are as shown in the figure 17.

We can again use this matrix to calculate our evaluation metrics.

- Accuracy: 0.8896
- Precision: 0.8800
- Recall: 0.8800
- F1: 0.8381

Activation Function	*Optimizer*	*Learning Rate*	*Accuracy*	*Precision*	*Recall*	*F1 Score*
relu	adam	0.01	0.8766	0.8000	0.8727	0.8348
relu	adam	0.03	0.8896	0.8800	0.8000	0.8381
relu	sgd	0.01	0.7662	0.6234	0.8727	0.7273
relu	sgd	0.03	0.7532	0.6076	0.8727	0.7164
sigmoid	adam	0.01	0.8506	0.7667	0.8364	0.8000
sigmoid	adam	0.03	0.8377	0.7273	0.8727	0.7934
sigmoid	sgd	0.01	0.7857	0.6571	0.8364	0.7360
sigmoid	sgd	0.03	0.7792	0.6615	0.7818	0.7167
tanh	adam	0.01	0.8506	0.8200	0.7455	0.7810
tanh	adam	0.03	0.6429	N/A	0.0000	N/A
tanh	sgd	0.01	0.7792	0.6615	0.7818	0.7167
tanh	sgd	0.03	0.7792	0.6567	0.8000	0.7213

TABLE II
METRICS FOR DIFFERENT CONFIGURATIONS

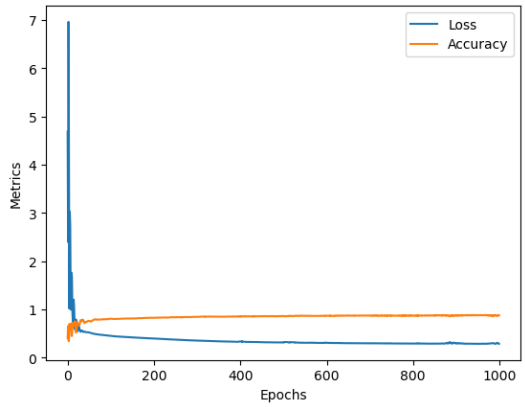

Fig. 16. Loss and Accuracy Curve for Multilayer Perceptron

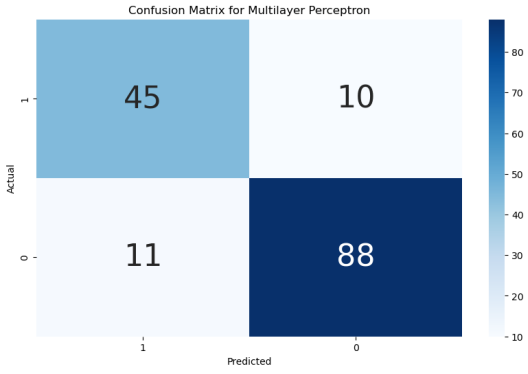

Fig. 17. Confusion Matrix for Multilayer Perceptron

In a brief comparison, the second model evaluation shows significant improvements over the first: it achieves higher accuracy (88% vs. 80%), better prediction of positive cases (78% vs. 74.5%), and more accurate identification of actual positive cases (88% vs. 69.1%). Additionally, its F1 score of 0.83, compared to 0.717 in the first model, indicates a very good balance between recall and precision.

VI. FINAL DISCUSSION

We saw in these finalized results how much of an affect, appropriate preprocessing can do. However what measure should we care more about so that we can determine which model is the best? The measure we want to care more is recall. The reasoning is simple, we want to minimize false negatives (FN). For our scenario, which is diabetes prediction we want to minimize the false negative (actual positive) cases which might have more implications for the patient (Letting a patient go without prescription or anything). So the metric we care most about in this case would be Recall. In the opposite case which is to closely monitor the patient or advise the patient healthier foods will have minimal effects compared to the scenario which we let a patient go. This means that the measure recall which emphasizes on false negatives is the best measure. This means that the decision tree algorithm is the best algorithm. Also since decision trees are highly interpretable, they were already an algorithm to keep an eye on. So we can safely say that the best algorithm in our project is decision trees.

VII. CONCLUSION

In conclusion, diabetes is a disease that can affect the lives of the patients seriously around the world. Our project's goal is to use various methods and algorithms on the Pima Indians Diabetes dataset to predict the disease and classify patients whether they have diabetes or not. Early prediction of these disease can help the individuals better prepare for the disease and modify their lifestyle to prevent it. The methods we will use each have unique weaknesses and strengths and can help us find patterns that we may otherwise have not found. Our methods include Decision Trees, Random Forest, Logistic Regression and Multilayer Perceptron. After these machine learning methods have been trained and evaluated our results show that Multilayer Perceptron and Decision Trees are the best applications for our project which is to maximize True Positive and True Negative cases and minimize False negatives. After our preprocessing we saw a great increase in our results which show how much of an affect removing outliers and filling missing values can make in prediction. As future work, we can add even more complex models such as neural networks or XGBoost algorithm and maybe even use an ensemble method to get even better results than we have achieved in this project. In summary this project uses data mining and machine learning techniques to help predict a disease which anyone in the world can encounter. We hope that it our project can contribute to fighting this chronic disease.

REFERENCES

[1] "WHO," accessed 2023-10-27. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/diabetes

[2] "NIDDK," accessed 2023-10-27. [Online]. Available: https://www.niddk.nih.gov/

[3] J. W. Smith, J. E. Everhart, W. Dickson, W. C. Knowler, and R. S. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the annual symposium on computer application in medical care*. American Medical Informatics Association, 1988, p. 261.

[4] "Pima Indians Diabetes Databas," accessed 2023-10-27. [Online]. Available: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database/data

[5] S. Mahajan, P. K. Sarangi, A. K. Sahoo, and M. Rohra, "Diabetes mellitus prediction using supervised machine learning techniques," in *2023 International Conference on Advancement in Computation Computer Technologies (InCACCT)*, 2023, pp. 587–592.

[6] E. Daniel, J. Johnson, U. A. Victor, G. Aditya, and S. A. Sibby, "An efficient diabetes prediction model using machine learning," in *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2023, pp. 1202–1208.

[7] M. Komi, J. Li, Y. Zhai, and X. Zhang, "Application of data mining methods in diabetes prediction," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, 2017, pp. 1006–1010.

[8] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76 516–76 531, 2020.

[9] H. Naz and S. Ahuja, "Deep learning approach for diabetes prediction using pima indian dataset," *Journal of Diabetes & Metabolic Disorders*, vol. 19, no. 1, pp. 391–403, June 1 2020. [Online]. Available: https://doi.org/10.1007/s40200-020-00520-5

[10] H. Tita, R. Sharma, A. Nayak, A. Sancheti, S. Bandyopadhyay, and P. K. Dutta, "Analyze the use of machine learning models in the pima diabetes data set for early stage detection," in *6th Smart Cities Symposium (SCS 2022)*, vol. 2022, 2022, pp. 238–242.