Early Diabetes Risk Prediction Using Neural Networks

The Pennsylvania State University
Alven Huang (afh5922@psu.edu) - PSU ID: 951213286
Qasim Ansari (qia5020@psu.edu) - PSU ID: 942433846

ABSTRACT

This project helps investigate early stage diabetes risk prediction with the machine learning tools and ideas and using a deep learning model called Multilayer Perceptron model, MLP. We used a data set called Early stage diabetes risk prediction and it included 520 medical records and 17 attributes of the records. We first explored and cleaned the data to make sure the model is clean and formal. Then we completed some of the baselines of logistic regression, decision tree, and random forest. Then we completed our proposed method MLP. After completing our code, we viewed the models and analyzed them and marked down our findings and results.

1. INTRODUCTION

In the modern world, predictions in healthcare are greatly supported by various data models. In our context, early detection of diabetes is extremely important so it can be treated efficiently. For this project, we will use our dataset from Kaggle to create a mock prediction model that estimates the risk of developing diabetes. Some challenges may include handling missing data, as well as identifying and working with both numerical and categorical variables. And for our deep learning model, we are going to use a multi-layer perceptron.

2. RELATED WORK

Earlier studies we found have illustrated that machine learning is very capable of accurate diabetes risk prediction. Various models such as logistic regression and decision trees were employed. On the other hand, recent work indicates that neural networks are actually more efficient in identifying intricate patterns in medical data. Overall, this aligns with our decision to utilize an MLP in the case of diabetes early prediction.

3. PROPOSED METHOD

Our proposed method will be to implement a Multilayer Perceptron model into our project. We chose to use an MLP model because we know how effectively it works with complex relationships like the patient symptoms. To design this, our model will have an input layer covering all of the patient attributes, followed by two hidden layers using ReLU activations. And finally, instead of a dropout layer like we mentioned in our proposal, we decided to use some of the built in techniques from MLPClassifier such as L2 regularization to prevent overfitting. The output layer will use a sigmoid function to predict whether a patient is at risk of early stage diabetes or not.

4. DATASET AND PRE-PROCESSING

We used the Early stage diabetes risk prediction data set. To find the shape of this data set, we imported and read in the csv file then used df.shape to find the shape. It gave us (520, 17). We also used df.head to quickly get a sneak peak of the top of the data set which is the first 5 rows. We then did df.info to get different information about the data set such as data types, entries, column names, and etc. Then we did df.describe().T.head(10) to get a summary

statistic of the first ten rows in the data set. We also checked for any potential missing values, NAs, which there were none.

5. BASELINES

The baselines that will be compared include Logistic Regression, Decision Tree, and Random Forest. Logistic Regression will be helpful for binary classification. Decision Tree will capture feature interactions and be very versatile. Random Forest will provide strong predictive power to our model. Comparing our neural network to these baselines will help us determine whether the added model complexity truly improves prediction accuracy for early-stage diabetes risk.

6. RESULTS & ANALYSIS

To truly display the accuracy and effectiveness of our work, we created several visuals and insights to help visualize the overall performance. Not only do our results showcase how each individual model performed, but they also prove how our final Multi-Layer Perceptron (MLP) provided the strongest combination of accuracy and stability. Following this, we will walk through our 4 figures and explain the key insights that they provide.

First, we have Figure 1. The main objective here was just to gain a broad perspective of all four models and their accuracy levels. It just helps us identify whether increasing the model's complexity is actually worth it or not, or if it really leads to meaningful results. As you can see below, the final MLP's accuracy is clearly higher than the baseline MLP by about 3%. This proves that the extra tuning we performed along with the other adjustments made resulted in a positive way. Also, by placing these Logistic Regression, Decision Tree, Baseline and Final MLP models side by side, we get to visualize the performance gap. Ultimately, this chart sets the beginning point for why we chose to refine and alter the baseline MLP more deeply.

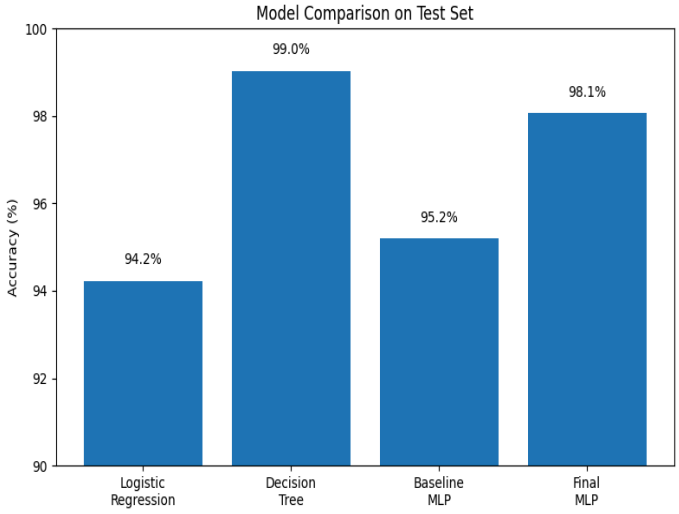

Fig 1: Model Comparison

Next, we have our training loss curve (Fig 2) for our final MLP. The main purpose of this loss curve is to tell how well our final model learned over time by using epochs. Starting off, the model loss drops pretty sharply which proves that it was able to capture meaningful patterns in the data. Now when you notice the curve start to flatten out a bit around the 7.5-10 epoch mark, this shows that there is not much more improvement left and it has reached a stable point already.

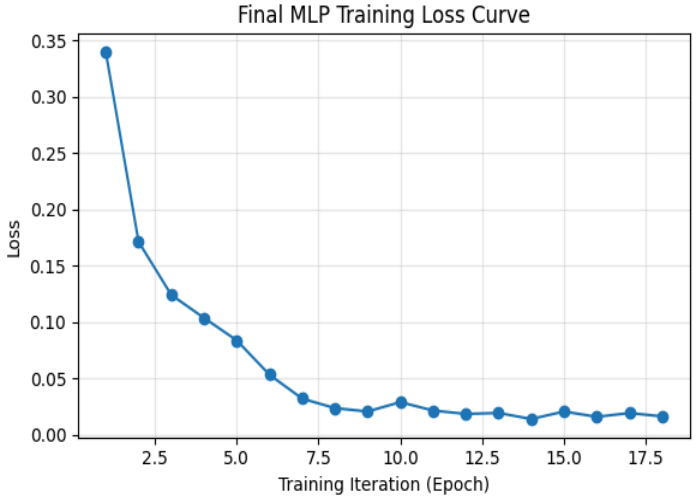

Fig 2: Training Loss Curve

We also decided to create a ROC Curve because of how informative it is. The beneficial thing about ROC Curves is that not only do they just provide you accuracy, they give you a deeper understanding of the model's true-positive versus false-positive rate. The importance is that it helps us measure the overall confidence of the model and how it separates between the classes. As you can see in Figure 3, our curve shoots up almost vertically and stays towards that top left corner. We received an AUC score of 0.998 as well meaning that the model is very effective at ranking the positive cases. Both of these insights prove that this is an ideal look and that it's a high performing classifier.

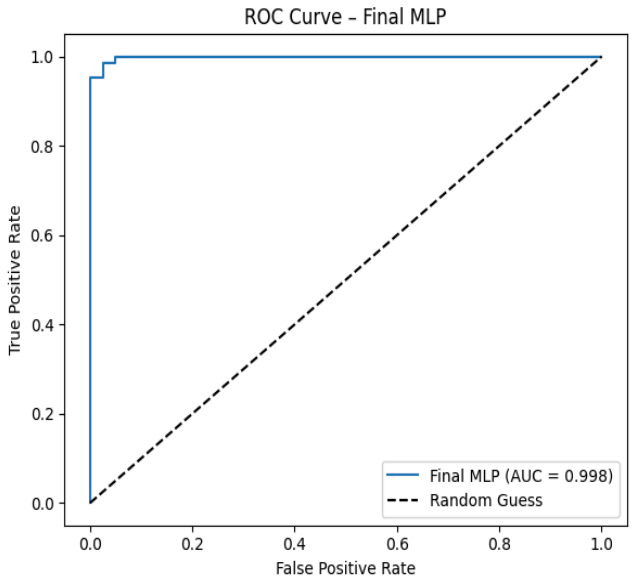

Fig 3: ROC Curve

Finally, we present our Feature Importance bar chart. Our main objective with this visual was to see which symptoms the model truly relies on the most. The way a permutation-based chart works is that it'll randomly shuffle one of the features at a time and measure how much the accuracy drops. So, if the model's accuracy were to drop pretty drastically after removing a feature, we know how important that specific feature is to the model. Looking down below, we can see that Polyuria and Polydipsia lead all features and caused the biggest drop, proving that they are indeed the most important predictors in our model.

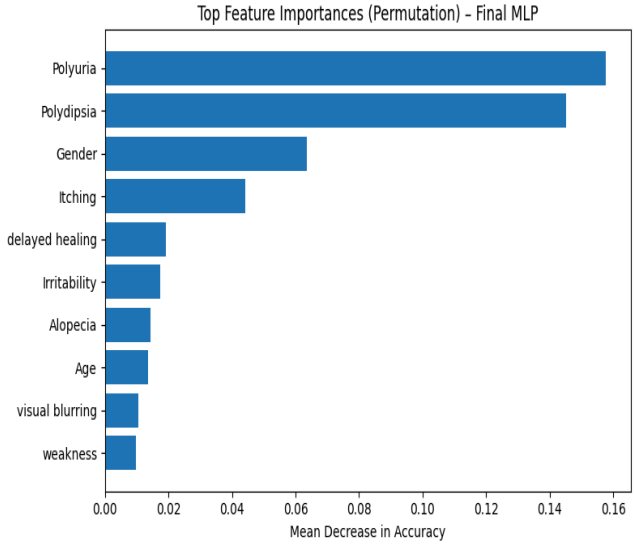

Fig 4: Feature Importance (Permutation-Based)

7. CONCLUSION

For our overall performance, our tuned MLP achieved a 98% accuracy, which improved from our 95% accuracy from our baseline MLP. The Decision Tree showed promise and performed greatly, but there were signs of overfitting while the MLP was a lot more stable and generalized. For our key insights and what we learned, we noticed that tuning the MLP by adding more neurons, regularizations, and early stopping, significantly improved the model's ability to learn symptom patterns. Also the final confusion matrix showed very few misclassifications indicating strong reliability for both positive and negative classes.

The loss curve demonstrates smooth convergence, confirming that the model learned effectively without overfitting. We also noticed that symptoms with the largest influence on predictions were Polyuria, Polydipsia, and gender. This also aligns with the medical expectations of increased urination and thirst are strong indicators of early diabetes. In conclusion, real healthcare problems rarely have perfect data,

so cleaning, encoding, and understanding the variables given is crucial to succeed. We learned that a model that performs well on paper still needs to be interpretable, especially in medical settings where trust and transparency matter. This is where experimenting with multiple models taught us that no single algorithm is best. The overall performance depends on context, data size, and tuning.

Ultimately, this project showed how important it is to communicate results visually so people can easily understand model behavior. We gained tons of appreciation for the end-to-end workflow, such as gathering and cleaning data, modeling, and evaluation, to finally presenting insights. We learned the importance of validating results, not just trusting one run. It was also very interesting to perform cross-validation and tuning to assert more confidence in the model.

8. REFERENCES

[1] Qi, X., Lu, Y., Shi, Y., Qi, H., and Ren, L. 2024. *A deep neural network prediction method for diabetes based on Kendall's correlation coefficient and attention mechanism.* National Institutes of Health (PMC). Retrieved from https://pmc.ncbi.nlm.nih.gov/articles/PMC11218995/

[2] Guyon, I. and Elisseeff, A. 2003. *An introduction to variable and feature selection.* Journal of Machine Learning Research 3, 1157–1182. Retrieved from https://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

[3] Dutta, I. 2019. *Early Stage Diabetes Risk Prediction Dataset.* Kaggle. Retrieved from https://www.kaggle.com/datasets/ishandutta/early-stage-diabetes-risk-prediction-dataset