Predicting Diabetes Using Decision Trees and Neural Networks

Introduction

The purpose of this health-related project is to predict whether or not an individual is diabetic. In the future, this project could be useful for doctors in identifying patients who could possibly be diabetic. Decision trees and neural networks were primarily used to model data and make predictions. Presentation slides here.

Dataset

The data used in this project was retrieved from Kaggle and "is originally from the National Institute of Diabetes and Digestive and Kidney Diseases." It consists of 769 lines and 9 features of health-related data. The dataset consists solely of female patients at least 21 years of age and of Pima Indian heritage. Features include the number of pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, age, and family history of diabetes. The target variable is whether or not the individual has diabetes. The dataset did not require any significant cleaning.

Analysis Techniques

This project involved decision trees and neural networks. Additionally, we used other libraries, such as `pandas`, `seaborn`, `matplotlib`, and `sklearn` to explore and analyze the data.

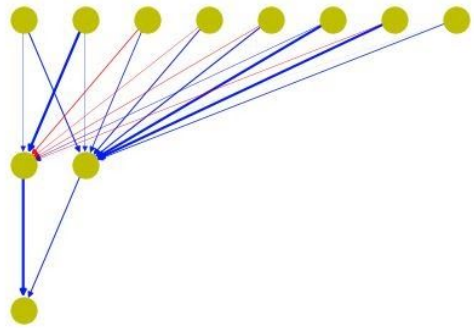
From `sklearn`, we used `train_test_split` to divide our data into training and testing portions, `StandardScaler` to prepare the data for use in a neural network, and `classification_report` and `confusion_matrix` to evaluate the models and analyze the results.

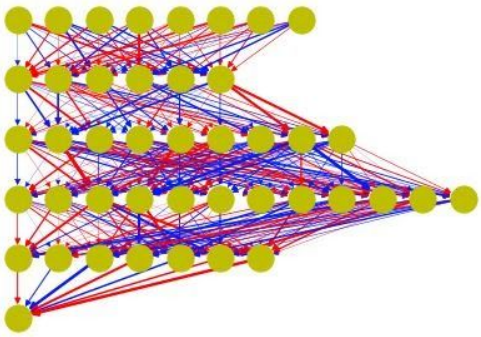
We used the `MLPClassifier` from `sklearn` as our neural network model. Thousands of neural networks were generated and tested with 1-4 hidden layers and between 2 and 14 nodes in each layer.

For the decision tree, we used `DecisionTreeClassifier` from `sklearn`. There were two different depths used, 3 and 5, and the feature columns were decided by running every combination of columns through the classifier, and finding the columns that had the highest feature importance score. The top 5 scoring columns were used to construct the trees.

Results

For the neural network models, accuracies ranged between 0.63 and 0.81. All four models that scored 0.81 had four hidden layers. However, k-fold cross validation would likely yield more accurate results for each model. However, given the many thousands of models that were trained, this type of cross-validation was not feasible for all models. It seems likely that more data is necessary for significantly improved accuracy (and precision, recall, and f-scores). We took three high-performing trees (1 hidden layer with 2 nodes, 3 hidden layers with 3 nodes -> 11 nodes -> 10 nodes, and 4 hidden layers with 6 nodes -> 9 nodes -> 12 nodes -> 7 nodes) and retrained the models on shuffled data. The accuracy of the two more complex models decreased, while the simple model actually performed slightly better. The single hidden layer model is shown above to the right while the 4-hidden layer model is shown below to the right. The f-score for the simple model was 0.87 for predicting the lack of diabetes and 0.60 for predicting diabetes.





For the decision trees, the models were much better at predicting the lack of diabetes. The highest f-score for predicting a lack of diabetes was 0.84. For predicting the presence of diabetes, the best f-score was 0.67. After running every single column through the classifier, the average feature importance score was taken. By far, glucose was the most important column, (see Fig 1).

The importance of glucose can also be seen in the tree graphs. The first split is based on glucose, and throughout the tree, glucose is used to split the tree into different branches, (see Fig 2,3). Interestingly, the first three levels of the two trees are the same. This is reflected in the feature importance score. As you go farther down the tree, that is when the attributes, such as insulin or pregnancies show up.

With this data having only two classes, there are better analysis techniques that could be used on this data. It is quite possible that overtraining is occurring with this data. However, this analysis technique is quite good for showing the importance that columns have in a quantifiable way.

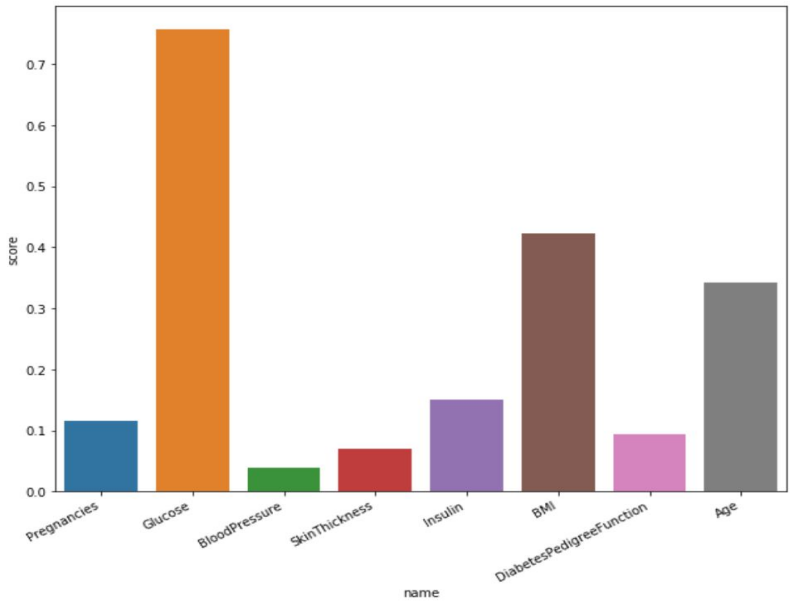


Figure 1: Average Feature Importance Score

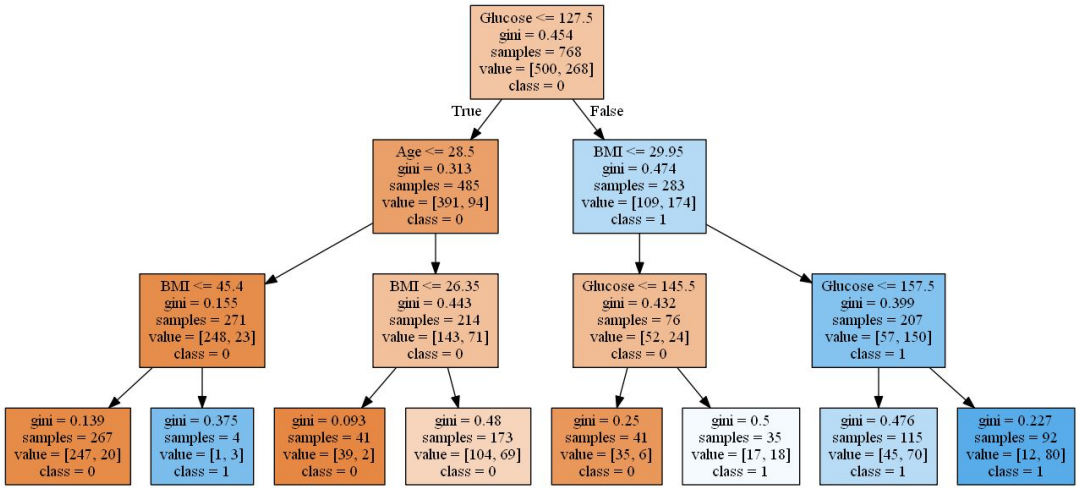


Figure 2: Depth 3 Tree

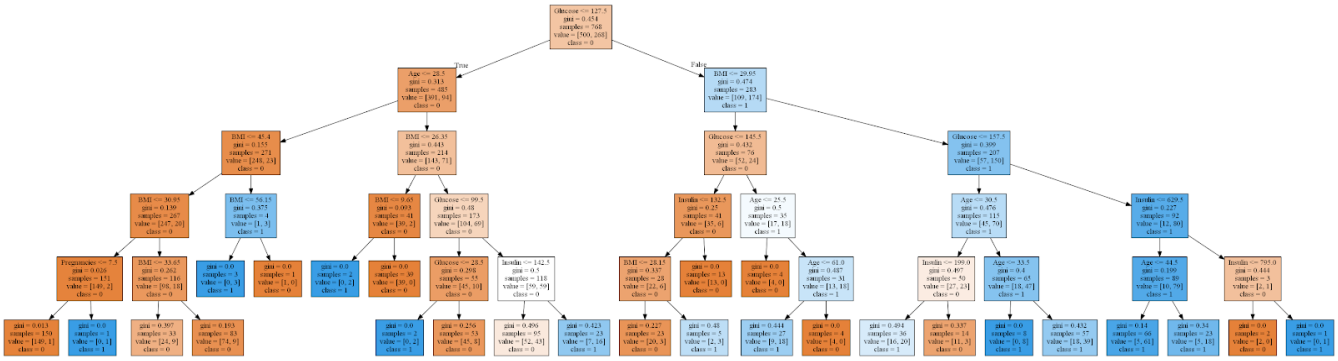


Figure 3: Depth 5 Tree