Diabetes Diagnosis Model Using Decision Trees

Advanced Intelligent Systems (CS-ELEC1A) Laboratory Activity

Mala-ay, Amiel Christian E.

I. INTRODUCTION

Classification problems are present in virtually every field. From the minutiae of daily life to professions like healthcare, commerce, and information security, there will always be instances where things need to be classified into one group or another. Examples of such problems include spam filtering, image classification, and credit card fraud detection. Diagnosing diabetes is another classification problem, where patients can be grouped into two classes (with diabetes or without diabetes) based on their medical information, including, but not limited to, age, BMI, glucose levels, etc.

Problems such as diabetes diagnosis can be addressed through the use of machine learning approaches to classification. Decision trees, particularly classification trees, can be a useful tool in this regard. When a data point is run through a decision tree, its attributes are evaluated at every node by Boolean expressions to decide the next node until it finds its way to a leaf node, where a class is assigned to the data point. Through an approach known as decision tree learning, a model could be trained to construct a decision tree based on a given data set, provided that the data set consists of discrete data.

However, there are caveats to this approach. For one, overly large and complex decision trees can cause the model to overfit the data and hinder its ability to generalize. As the data set is split into smaller subsets, especially in deeper decision trees, the leaf nodes could end up containing little data individually, abstracting away any statistically significant patterns and leading to overfitting. Another concern is that decision tree learning uses a greedy algorithm, which does not necessarily yield globally optimal results. This could lead to a suboptimal decision tree that does not fit the data well. As such, there is a case to be made for the use of other techniques to supplement the decision tree model's performance, like pre-pruning and post-pruning the tree.

II. METHODOLOGY

The data set used for this laboratory exercise consists of 768 data points contained in a CSV file, each containing 9 features: pregnancies, glucose levels, insulin levels, blood pressure, skin thickness, BMI, diabetes pedigree, age, and outcome (i.e., whether or not they have diabetes). 500 of these individuals had no diabetes, while 268 of them did have diabetes.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

Figure 2.1 Initial data set

While a surface evaluation of the data set showed that there were no null values, further examination revealed that many of its data points had incomplete data. Instead of being represented as null values, unavailable information in the glucose, blood pressure, skin thickness, insulin, and BMI columns were represented as 0. Upon inspection, it was found that 376 data points had zero-values in the aforementioned columns out of 768 total data points. This amounts to nearly half of the data set, which means that simply removing any incomplete data would not be a prudent measure.

To resolve the matter of missing data, unit imputation was performed, where a data point's missing information is replaced with reasonable estimates (Glas, 2010). This was done with the use of the *k*-nearest neighbors algorithm, where an incomplete data point's missing values are approximated using the corresponding values in the *k* most similar data points, or "nearest neighbors," where *k* is a positive integer (Beretta & Santaniello, 2016). For this laboratory exercise, determining an incomplete data point's *k* nearest neighbors was done using Euclidean distance as the metric.

Additionally, regularization was performed to improve the model's performance by encouraging the model to construct smaller but more informative trees. This was done by making adjustments to the model's hyperparameters, also known as hyperparameter tuning. As part of this process, fixed values were provided for the tree's maximum depth and maximum leaf nodes.

After the data preprocessing procedure, the data set was split into training and testing sets, with 80% of the data set being used for training the model and 20% being set aside for testing. The model's performance on unseen data points was assessed using its accuracy, recall, precision, and F1 scores.

In improving these metrics, the recall-precision trade-off poses a challenge. While improving accuracy, or the ratio between the model's correct predictions and its total predictions, is fairly straightforward, the same cannot be said for recall and precision. A diabetes diagnosis model with high recall would be good at diagnosing diabetes when it is truly present, while a diabetes diagnosis model is likely to be correct when it flags a patient as diabetic. Higher recall leads to a reduced chance of false negatives, while higher precision leads to less false positives. Both are good metrics, but because one score's improvement can come with the cost of the other's reduction, it is necessary to determine which metric is more pertinent to the problem for which the model is constructed.

In the end, it was decided that recall, which measures how many of the patients with diabetes were correctly diagnosed, should be prioritized over precision, which measures how many of the patients diagnosed with diabetes truly have diabetes. A patient being incorrectly diagnosed with diabetes would not be as harmful and costly as a diabetic patient being incorrectly diagnosed as non-diabetic (Chaves & Marques, 2021). It must be emphasized, however, that the model should have good scores for both metrics. The F1 metric, the harmonic mean between recall and precision, strikes a balance between the two metrics. A good F1 score ($\leq$0.7) indicates that the model's recall and precision scores are in a good place (Logunova, 2023).

III. EXPERIMENTS

3.1 Feature Engineering

Choosing which features to include or exclude from the data set is critical to the model's performance. Feature engineering was conducted to determine which features were conducive to the model's ability to detect diabetes. Since all of the data set's features contained numeric data, there was no need to perform any manipulations like one-hot encoding to allow the model to interpret the data.

The results of the trials revealed that 6 of the input features were helpful in optimizing the model's performance, with only pregnancy count and skin thickness being excluded. It is true enough that there are relationships between these features and the presence of diabetes (Ruiz-Alejos et al., 2020). For instance, gestational diabetes can develop during pregnancy, and women who have it have an increased likelihood of developing diabetes permanently. The more pregnancies one has, the higher the likelihood of having had gestational diabetes (Das et al., 2022). Skin thickness, on the other hand, can be used for assessing body fat content, which is correlated with diabetes. All that said, however, it cannot be said that these relationships are particularly strong. This much is evidenced by the fact that neither including nor excluding them has any impact on the model's performance.

As for the remaining features, their positive effect on the model's performance comes as no surprise, as they are commonly used in medical practice as reliable metrics for diagnosing a patient with diabetes (Watson, 2018). Abnormal glucose and insulin levels, in particular, are closely linked to diabetes, as the disease mainly affects the body's ability to produce insulin for regulating glucose levels in the blood.

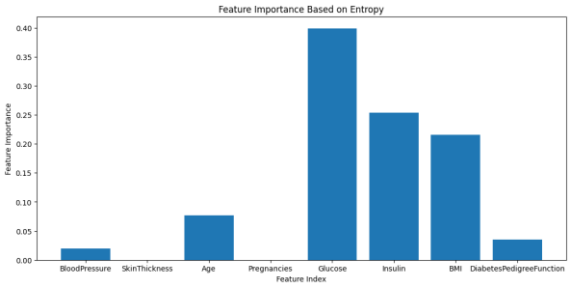

Figure 4.3.1 A feature importance bar graph based on entropy

Performing feature engineering led to a significant increase of 0.1558 in the model's accuracy and a staggering increase of 0.2517 in its precision, but it also led to a considerable decrease of 0.1273. The model's F1 score increased by 0.0772. Although the model's accuracy, precision, and F1 scores improved through feature engineering, the decrease in its recall score is a cause for concern, as it was decided that recall is the more relevant metric for the problem and should therefore be prioritized over precision.

3.2 Regularization with Hyperparameter Tuning

Regularization in the context of decision tree learning mainly involves hyperparameter tuning. Tuning certain hyperparameters in the model sets limits on the growth of the decision tree generated by the model. This process is known as pre-pruning, as opposed to post-pruning, where the tree is truncated after it is generated. Pre-pruning with hyperparameter tuning allows the model to create smaller but more informative decision trees that can be used to make more correct classifications.

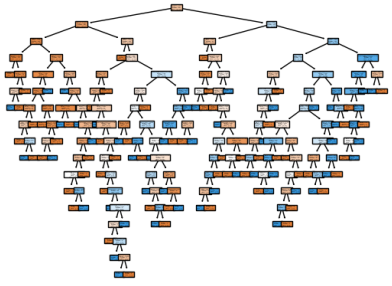

Figure 3.2.1 The unregularized decision tree, allowed to grow without any limitations

The adjusted hyperparameters were the tree's maximum depth and maximum number of leaf nodes. Setting a limit on the tree's depth prevents it from growing too large, while setting a limit on its leaves prevents hyper-specific leaf nodes with few samples from emerging. The tree's maximum depth was set to 5, while its maximum number of leaves was set to 17. These values were obtained largely through trial and error.

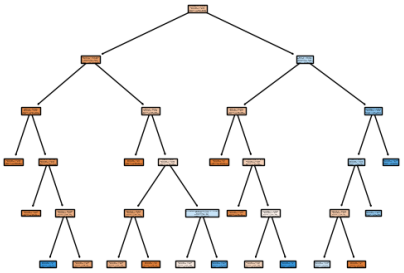

Figure 3.2.2 A decision tree generated after pre-pruning through hyperparameter tuning

By setting a maximum depth and a maximum number of leaves for the tree, the model's performance improved considerably. The model's accuracy increased slightly by 0.013, while its F1 score improved by 0.0507. While the model's precision score went down by 0.0236, its recall score improved greatly, with an increase of 0.1091. Since recall is the prioritized metric for this problem, the slight drop in the model's precision score is not a cause for concern, especially since the recall score improved considerably.

3.3 Zero-Value Removal

The initial data set contains data points with incomplete data. Missing data is not immediately obvious as such data is not marked as null. Instead, missing values are given values of zero. Zero-values can also evade detection because of the presence of features where zero is a normal value, like pregnancies. Therefore, it is necessary to carefully select the features for which zero is an abnormal value.

Pregnancies	111
Glucose	5
BloodPressure	35
SkinThickness	227
Insulin	374
BMI	11
DiabetesPedigreeFunction	0
Age	0
Outcome	500

Figure 3.3.1 Total zero-values in each column in the initial data set

Zero-values are not possible for BMI, as this value is computed by dividing a person's weight by their squared height $\left(\frac{weight}{height^2}\right)$, both of which cannot be zero. While glucose and insulin levels can be low, zero glucose and insulin are not typical for living patients (Watson, 2016). As such, it is safe to say that such zero-values are erroneous in nature. The same goes for skin thickness since zero skin thickness implies an absolute lack of skin, which is atypical of a living patient. Therefore, it can be said definitively that zero-values in the BMI, glucose, insulin, and skin thickness features are abnormal and should be treated.

Problematically, however, data points with incomplete data in any of the aforementioned columns cannot be addressed without issue through simple removal. Upon inspection, it was found that there are 376 such data points out of 768 total data points, equivalent to almost half (~49%) of the data set. It would be difficult to justify the removal of nearly half of the entire data set, so it is necessary to resort to other measures for addressing these zero-values.

This does not mean, however, that removing data points cannot be done. The Glucose column contains only 5 zero-value data points, while the BMI column contains only 11 zero-value data points. Working with the worst-case assumption that they are independent sets, these data points combined are but tiny fractions of the data set (~2%). Therefore, removing zero-values in the glucose and BMI columns would not be problematic.

Removing zero-values from the glucose and BMI columns removed 16 data points from the data set, leaving it with 752 data points. This confirms that the two columns' zero-value data points were indeed independent sets. More importantly, removing zero-value data points from these columns benefited the model's performance, with the model's F1 score increasing by 0.0548. Its precision and recall increased by 0.0194 and 0.0924, respectively, while its accuracy increased by 0.009. Overall, removing zero-values from the glucose and BMI columns yielded benefits for the model on all fronts.

3.4 Imputation with *k*-Nearest Neighbors Algorithm

Removing zero-values from the data set's glucose and BMI columns yielded benefits for the model's performance. However, zero-values remain in columns where they don't belong, namely in the Insulin, BloodPressure, and SkinThickness columns. The amount of data points with zero-values in these columns necessitates an approach that doesn't involve removing a massive portion of the data set. Instead of removal, imputation was used to remove these zero-values.

To perform imputation, the *k*-nearest neighbors algorithm was used, where a data point's missing values are approximated using the corresponding values in the

data point's k nearest neighbors, i.e., the data points most similar to it, where k is a positive integer (Beretta & Santaniello, 2016). To find a data point's k nearest neighbors, the Euclidean distance formula was used, as it is the default metric used by the scikit-learn library's implementation of the algorithm.

The performance of this algorithm, and, by extension, the performance of the model, hinges on determining the appropriate value for k. To find the appropriate value for k, the model was run in a loop where a variable representing k was incremented from 0 to the total amount of rows in the data frame and appended to a list as a 2-tuple alongside its corresponding F1 score. The optimal k was obtained from the tuple in the list with the highest F1 score. While somewhat crude, this approach was able to determine that the value that yielded the best results for the model was $k = 113$.

The use of the k nearest neighbors algorithm with the optimal value for k for imputation yielded no changes in the model's recall score but increased the model's accuracy by 0.0132, its precision by 0.0229, and its F1 score by 0.0122.

IV. RESULTS AND DISCUSSION

4.1 Final Decision Tree

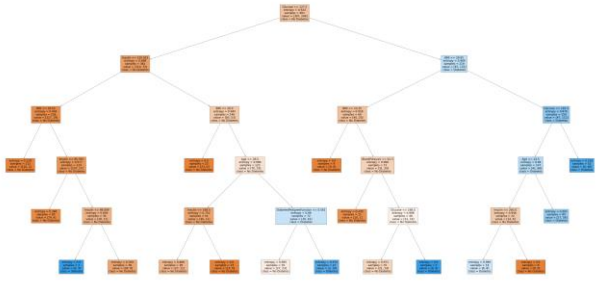

Figure 4.1.1 The final decision tree generated by the model

The decision tree learning model was able to generate a decision tree with a depth of 5 and 17 leaf nodes. Although it seems overly complex to the point that it is not legible, such a tree is a huge improvement over a tree like that shown in Figure 3.2.1.

4.2 Final Accuracy, Precision, Recall, and F1 Scores

Accuracy	0.8079
Precision	0.7460
Recall	0.7833
F1 Score	0.7642

Figure 4.2.1 Final accuracy, precision, recall, and F1 scores

The final version of the model, with all the previously detailed procedures performed, had an accuracy score of 0.8079 and an F1 score of 0.7642. These are both good scores (≥0.70), indicating that the model, while not perfect, can correctly assess whether patients have diabetes most of the time.

The trade-off between precision and recall was evident throughout the process of experimentation. Increases in one metric led to only slight increases and even decreases in the other. After performing all the previously detailed procedures, the model's recall score, at 0.7833, ended up higher than its precision score of 0.746. This falls in line with the decision to prioritize the model's recall over its precision, as recall was deemed the more pertinent metric to the problem of diabetes diagnosis.

4.3 Confusion Matrix

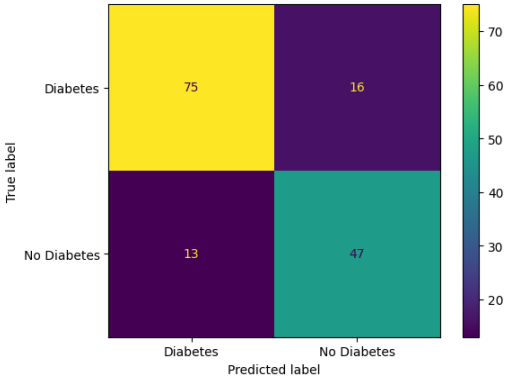

Figure 4.3.1 Final confusion matrix demonstrating the model's performance on unseen data

The confusion matrix in Figure 4.2.1 offers a more detailed view of the model's performance. It contains more details about the model's ability to predict the outcome of the 151 data points in the testing set, with a heatmap and numbers indicating how many data points are in a cell. From the figure, it can be gleaned that the testing set contained 91 individuals with diabetes and 60 individuals without.

As observed in Figure 4.2.1, the number of true positives and true negatives overshadow the amount of false positives and false negatives. In total, only 29 data points were not correctly assessed by the model, with 16 false negatives and 13 false positives. This means that ~17.58% of diabetic individuals were not flagged as such by the model, while ~21.67% of individuals who did not actually have diabetes were flagged by the model as diabetic anyway. It would be easy to blame the higher incidence of false positives on the earlier decision to prioritize recall over precision. However, these differences are not significant enough to make any grounded inferences on which direction the model tends to misdiagnose. This is perhaps due to the limited size of the data set, with only 151 data points being left available for testing after a train-test split on 752 data points.

Of the 151 data points in the testing set, 122 were correctly assessed for diabetes, giving an accuracy score of ~80.79%. While it has already been established that the model's accuracy is not quite perfect, it still falls within the threshold of what can be considered a good score (≥70%). That said, it is concerning that 29 out of 151 data points (~19.21%) were misdiagnosed by the model, especially since the testing set is already quite small. It does not bode well for the model and its future performance on unseen data that nearly 20% of its predictions are inaccurate on such a small set.

V. CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary of Findings

With an accuracy of 0.8079 and an F1 score of 0.7642, well above the threshold of what can be considered a good score (≥0.70), it can be said that the decision tree model can classify patients as diabetic or non-diabetic with a reasonable degree of correctness. While both scores are not quite close to 1, which signifies a perfect score, these scores are adequate. Of course, there are reasons to be concerned about a nearly 20% chance of misdiagnosis, so it could be difficult to make a case for the usage of this model for actual diabetes diagnosis.

While the precision-recall trade-off posed an additional dilemma, it was eventually decided that the model's recall was to be prioritized in order to reduce the incidence of false negatives, i.e., diabetes patients being incorrectly diagnosed as non-diabetic. With a recall score of 0.7833 and a precision score of 0.7460, it can be said that success was attained in this regard. That said, it must be noted that precision is a good metric as well, and that high precision reduces the rate of false positives. The decision to prioritize recall over precision could be part of the reason why there was a slightly higher incidence of false positives than false negatives when the model was exposed to unseen data. All that said, prioritizing the reduction of false negatives over the reduction of false positives was a reasonable decision, as false negatives in the context of diabetes diagnosis are costlier and more harmful than false positives (Chaves & Marques, 2021).

There is reason to believe that the data set's small size, with 768 starting data points, could be blamed for the model having some shortcomings. With only 752 data points remaining after removing certain data points with zero-values, the model was trained on 601 data points, with 151 data points being left for the testing set. While the model's performance on unseen data in the testing set was somewhat satisfactory, it was difficult to make serious assessments on the abilities and shortcomings of the model simply because the testing set, and the data set as a whole, was just too small. Its performance on the small testing set might not be reflective of its performance on unseen data in larger and more diverse data sets. There is also the matter of the imbalance between the data set's diabetic and non-diabetic patients, with 500 non-diabetic patients and 268 diabetic patients in the initial data set. In addition to the data set's small size, the data set's imbalance could also be one of the reasons for the model's shortcomings.

5.2 Recommendations

Based on the findings of the study, it can be said that the decision tree model can diagnose diabetes in patients with a reasonable degree of correctness. However, it must be noted that there is a good chance of the model making misdiagnoses, with a fairly even chance of it diagnosing non-diabetic patients with diabetes and diagnosing diabetic patients as non-diabetic. As such, it is difficult to make a case for it being used as a stand-in for a medical professional in a setting where the lives of real people are at stake. Rather, it is easier to see the model being used as a supplementary tool for a skilled medical professional, not as a replacement.

For similar explorations in the future, a bigger and more balanced data set could result in the construction of a more trustworthy diabetes diagnosis model. As it stands, the performance of the model constructed in this laboratory exercise cannot easily be taken at face value because of the small size of the data set. Its ability to make correct classifications on unseen data simply cannot be trusted as the data set on which it was trained and tested is too small, which means its performance on the small data set might not reflect its performance on bigger and more diverse data sets.

VI. REFERENCES

Beretta, L., & Santaniello, A. (2016). Nearest neighbor imputation algorithms: a critical evaluation. *BMC Medical Informatics and Decision Making*, *16*(S3). https://doi.org/10.1186/s12911-016-0318-z

Chaves, L., & Marques, G. (2021). Data Mining Techniques for Early Diagnosis of Diabetes: A Comparative Study. *Applied Sciences*, *11*(5), 2218. https://doi.org/10.3390/app11052218

Das, M., Bhattacharyya, G., Gong, R., Misra, R., Medda, S. K., Banik, S., & Das, R. N. (2022). Determinants of Gestational Diabetes Pedigree Function for Pima Indian Females. *Internal Medicine – Open Journal*, *6*(1), 9–13. https://doi.org/10.17140/imoj-6-121

Glas, C. A. W. (2010, January 1). *Missing Data* (P. Peterson, E. Baker, & B. McGaw, Eds.). ScienceDirect; Elsevier. https://www.sciencedirect.com/science/article/abs/pii/B9780080448947013464

Kumar, A. (2022, January 20). *Classification Problems Real-life Examples*. Analytics Yogi. https://vitalflux.com/classification-problems-real-world-examples/

Logunova, I. (2023, July 11). *F1 Score in Machine Learning*. Serokell Software Development Company. https://serokell.io/blog/a-guide-to-f1-score

Ruiz-Alejos, A., Carrillo-Larco, R. M., Miranda, J. J., Gilman, R. H., Smeeth, L., & Bernabé-Ortiz, A. (2020). Skinfold thickness and the incidence of type 2 diabetes mellitus and hypertension: an analysis of the PERU MIGRANT Study. *Public Health Nutrition*, *23*(1), 63–71. https://doi.org/10.1017/S1368980019001307

Watson, S. (2016, August 20). *What is glucose?* WebMD. https://www.webmd.com/diabetes/glucose-diabetes

Watson, S. (2018, October 4). *Everything You Need to Know About Diabetes*. Healthline; Healthline Media. https://www.healthline.com/health/diabetes#symptoms