THE UNIVERSITY OF SYDNEY

University of Sydney

DATA3404

Data Science Platforms

Group Assignment

Team members:
450411920

Semester 1, 2019

Contents

Aim

The aim of this study is to compare the accuracy of various classifiers when applied onto a binary classification problem. In particular, we seek to classify whether does a patient have diabetes or not. A plethora of models have been employed and evaluated on 2 different datasets. This study allows for us to evaluate the effects of feature engineering, by employing the Correlation-Based Feature Selection algorithm, on our dataset and examining its impact on our models' performance. Hence, we seek to identify classifers that generalise best on unseen data or in other words, has the lowest expected prediction error. Furthermore, we implemented our own implementation of the Naive Bayes and Decision Trees algorithm and compared the results to industry standard models.

This study is important as it will have huge ramifications in the field of medical science. Medical misdiagnosis is unfortunately a common occurrence and can mainly be attributed due to human error. Hence, we seek to train a machine to make diagnostic predictions based on certain attributes of an individual patient. Most medical conditions are diagnosed physically via examination of a patient's symptoms and hence poses difficulty when attempting to detect diabetes on more granular level and again subjected to human biases when doctor evaluates the symptons. Hence, machine learning models can consider more variables and weigh more factors compared to what human doctors can do. Resultantly, this can ultimately assist patients and doctors through correct classifications. An additional use case of this is to assist doctors in identifying diabetes in patients early on and therefore be able to take preemptive measures in treating diabetes.

Data

The dataset is known as the 'Pima Indians Diabetes Database', compiled by the National Institute of Diabetes and Digestive and Kidney Diseases, and has been modified for the purposes of the study. The dataset comprised of females that are at least 21 years of age and are of Pima-Indian ancestory. There are 768 observations (or participants) in the dataset, whereby 500 of which have a class value 'no' or in other words, are not diagnosed with diabetes whilst the reamining 268 are diagnosed with diabetes. It is worth noting the class imbalances whereby we have nearly twice as many "no" instances than yes and hence may skew the model building section. The data contains 8 numerical attributes in addition to the response variable, which is the class of whether does the observation has diabetes or not. The dataset has been modified in the following manner. Firstly, the class variable were one hot encoded into numerical 1's and 0's values from "yes" and no". Additionally, missing attribute values were imputed by their column's average values. Finally, the dataset, exluding the class, has been normalised to the [0,1] range which ensures that the models implicitly weighs all features equally in their representation.

High dimensional data can be problematic for classifiers and typically requires an enormous amount of training data to ensure that there are several samples with each combination of values. Hence, correlated feature selection (CFS) was employed to reduce the dimensionality of the dataset. To rectify the curse of high dimensionality, two different versions of the dataset were utilised in the study where one is the original dataset and the other a dataset that has had CFS employed onto it in order to reduce its dimensions. The way that CFS works is that is that it selects a subset of the features which are highly correlated with the class whilst the correlation between features in the selected subset is kept to a minimum. In other words, it seeks to minimise multicollinearity between the features and identify the features that are most correlated with the class. It is a reasonable assumption to make that features highly correlated with the class will be strong predictors for the class. Specifically, best-first search was the search algorithm employed which has resulted in a subset of five attributes:

- Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- 2-Hour serum insulin level
- Body mass index
- Diabetes pedigree function
- Age

The features removed were:

- Number of times pregnant
- Diastolic blood pressure
- Triceps skinfold thickness

It is worthwhile to note that whilst all the features do exhibit some degree of possible correlation with diabetes, the CFS does pick some intuitive features such as age and BMI whilst some features such as diastolic blood pressure, which normally would also be a good indicator of whether a patient has diabetes, has been left out as it may be highly correlated with other features. This makes sense as we would expect patients that are older to exhibit higher diastolic blood pressure but also diabetes and hence having both those features may lead to the model fitting itself on noise rather than signals.

Results and Discussion

Classifier Accuracy

The accuracies of the models on datasets with and without feature selection are displayed in the tables and were computed using 10-fold cross validation. Here, we define the accuracy as the proportion of instances that were correctly classified as seen in the formula:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

We can see some unintuitive results from the effects of the CFS procedure. One would assume that applying the CFS procedure will help to reduce the number of attributes that do not offer much in terms of predictive capabilities after accounting for all the other predictors. Theoretically, this should help prevent the models from using redundant predictors or in other words, overfitting. When looking at the tables, it is evident that models trained on the CFS datasets do perform better, but only marginally so and hence it is difficult to determine whether the difference is statistically significant.

The accuracy of the models employed on the dataset can be seen in the next 3 tables.

Numeric Data	**ZeroR**	**1R**	**1NN**	**5NN**	**NB**
No feature selection	65.10%	70.83%	67.83%	74.48%	75.13%
CFS	65.10%	70.83%	69.01%	74.48%	76.30%

Numeric Data	**MLP**	**SVM**	**RF**	**MyNB**
No feature selection	75.39%	76.30%	74.87%	75.26%
CFS	75.78%	76.69%	75.91%	76.04%

Nominal Data	**DT Unpruned**	**DT Pruned**	**MyDT**
No feature selection	75.00%	75.39%	73.44%
CFS	79.43%	79.43%	78.38%

Diagrams of the decision trees can be viewed in the appendix. It is worth noting here that the decision tree that we implemented differs in length significantly to the one created by Weka. This can be explained by the fact that the Weka software uses a variant of the information gain in creating its trees and hence generates shorter trees.

Discussion

ZeroR
ZeroR is the simplest classification which simplies classifies all observations according to the majority class. In this case, it simply classifies all observations as no. Hence, the ZeroR model only relies on the response variable and therefore running CFS will not affect the accuracy of the ZeroR model. This is useful in generating a baseline level of accuracy in which our models should achieve at a minimum since the models can simply resort to classifying all the observations as no in a worst case scenario.

1R
The One Rule model is also a simple classification algorithm that utilises only 1 rule on 1 predictor to help classify the dataset. Hence, it also serves as a good baseline model as all models being investigated utilises more than 1 feature and hence they have much more information than the 1R model and therefore should outperform the 1R model. We note that the KNN model for n=1 does not in fact outperform the 1R model and hence shows clear indicators of overfitting.

K-Nearest Neighbours
For the KNN model, we do notice a signficant improvement for when k=1 after applying CFS on the dataset. This is due to the fact that reducing the feature space helps to mitigate the curse of dimensionality which KNN can easily suffer from. Since the KNN model for k=1 looks at the class of its closest neighbour and classifies itself as such, due to there being less irrelevant variables in the dataset, this helps the model to select neighbours for its classifications that are more similar to it as there are no redundant features influencing the classification. However, for k=5, there is no difference in the performance in the model and this is most likely due to the fact the model has traded off an increase in bias for a decrease in variance. However, as there are less irrelevant features in the CFS dataset, this has helped improve the performance in the model by decreasing the bias and hence the net effect does not see a change in the performance of the KNN model where k=5. Finally, we note that the model where k=5 performs significantly stronger compared to when k=1 and this is most likely attributed to the fact that the model may be overfitting the in k=1 case.

Naive Bayes
There is quite a noteworthy improvement in the performance for the Naive Bayes algorithm after CFS was carried out on the dataset. This is most likely due to the fact that the Naive Bayes algorithms makes the assumption that the features are independent of each other given the class. This is quite a big assumption to make but it is often made in order to simplify calculations on large datasets. However, such an assumption does tend to improve the accuracy of the predictions as it trades off an increase in the bias of the algorithm for a reduction in the variance. However,

as there are only 8 predictors in the dataset, such an assumption can be quite costly. One plausible hypothesis for the improved performance is that CFS removes features that are highly correlated with each other and hence the remaining features are closer to satisfying the conditional independence assumption. Therefore, this leads to an increase in the accuracy of the model.

It is interesting to note that our implementation of Naive Bayes actually barely outperforms Weka's implementation on the dataset without feature engineering whilst it underperforms Weka's implementation in the case of CFS being applied onto the dataset. Hence, it is safe to conclude there is no real difference between the models and this is intuitive because there are not much variation on how Naive Bayes can be computed due to it just being a formula.

Decision Trees
For decision trees, it is interesting to note that there is no difference in performance when pruning is performed. However, we do see a significant improvement in the model when CFS is applied to the dataset. A hypothesis for this is that reducing the number of potentially irrelevant features means that the decision trees can now be constructed on features that are highly predictive of the class. Hence, the trees aren't being constructed with irrelevant features and "noise" which leads to improvements in accuracy.

There appears to be no significance difference in model performance between the pruned and unpruned decision tree. Hypothetically, pruning is meant to prevent decision trees from overfitting and allow for the trees to be generalisable to unseen data, that does not appear to be the case here. Hence, we can conclude that pruning does not have an signficant effect on the decision trees being constructed as the trees themselves are not overfitting the data. Further pruning helps to reduce the complexity of the tree It's worth pointing out that the decision tree structure of the model we implemented is signficantly longer than that of Weka's implementation. This is due to the fact that Weka utilises a modified variant of the information gain formula.

Here, it is clearly evident that our implementation of decision trees are not as accurate as Weka's. This is most likely due to the fact that decision trees allows for much more flexiblity in implementation and hence this can lead to different performance. Again, it is evident that pruning dramatically improves the performance of our decision tree as it helps to reduce overfitting from the model.

Random Forest
Similar to decision trees, we do see a significant improvement in the model when CFS is applied to the dataset. A hypothesis for this builds on the argument made for the improvement in the decision trees model and since random forests are a collection of decision trees, this leads to an overall improvement in the performance

of the random forest model.

Multi-layer Perceptron and Support Vector Machine
There has been marginal improvements to both the MLP and SVM model when CFS was performed on the dataset. However, it is questionable on whether this difference is signficantly different unless further statistical testing is conducted. Intuitively though, there should have been a more significant improvement in the SVM model as it is now simpler for the model to constructs hyperplanes to seperate the data if the feature space is of a lower dimension.

Conclusively, we can see that CFS does dramatically improve upon the models' performance since reducing the feature space helps to mitigate overfitting which models can easily suffer from. It allows the models' to be able to train itself on signals rather than noises. This makes intuitive sense as if you give models redundant information, it is very easy for the model to train itself on it and see its performance deteroirate. Whilst CFS could potentially improve model performance by removing noise from the data, we fail to see that in this study.

Further Analysis

Whilst accuracy is a commonplace metric, it is highly misleading at times such as in cases of class imbalances, it is simple to achieve a high accuracy score by labelling all the observations as the majority class. Therefore the precision and recall for each classifier should be examined as well. Intuitively, the precision looks at the proportion of accurate positive classifications (where here positive classifications means diagnosing the patient with diabetes) out of all the positive classifications. Within context of the study, this can be seen as the proportion of the number of observations with diabetes that were correctly diagnosed out of all the observations classified by the model as having diabetes. The recall is simply the proportion of accurate positive classifcations made out of all the observations that do have diabetes or put simply the proportion of correct identified observations with diabetes out of all observations that do have diabetes. The precision and recall looks at the proportion of correctly identified classifications in a more nuanced manner compared to the accuracy metric.

Hence we formulaically define precision and recall as:

Precision

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positives}}$$

Recall

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negatives}}$$

These metrics are especially relevant for the study as there is a noticable asymmetric impact when a patient with diabetes is incorrectly classified as healthy, otherwise known as a false negative, versus a healthy patient being incorrectly diagnosed with diabetes, otherwise known as a false positive. Here, it is obvious to see that whilst the healthy patient may undergo psychological distress from being diagnosed with diabetes, a patient who does have diabetes not being diagnosed correctly can lead to a life threatening situations whereby their medical condition could be treated and dealt with had the diabetes been correctly identified earlier. Therefore, we see that a false negative is much more harmful than a false positive.

Precision

Numeric Data	**ZeroR**	**1R**	**1NN**	**5NN**	**NB**
No feature selection	0.00%	69.80%	67.90%	74.10%	74.70%
CFS	0.00%	69.80%	69.00%	74.10%	75.57%

Numeric Data	**MLP**	**SVM**	**RF**	**MyNB**
No feature selection	75.50%	75.70%	74.70%	TBD
CFS	75.70%	76.20%	75.60%	TBD

Nominal Data	**DT Unpruned**	**DT Pruned**	**MyDT**
No feature selection	74.50%	75.10%	TBD
CFS	79.10%	79.10%	TBD

Recall

Numeric Data	**ZeroR**	**1R**	**1NN**	**5NN**	**NB**
No feature selection	0.00%	70.80%	67.80%	74.50%	75.10%
CFS	0.00%	70.80%	69.00%	74.50%	76.30%

Numeric Data	**MLP**	**SVM**	**RF**	**MyNB**
No feature selection	75.40%	76.30%	74.90%	TBD
CFS	75.58%	76.70%	75.90%	TBD

Nominal Data	**DT Unpruned**	**DT Pruned**	**MyDT**
No feature selection	75.00%	75.40%	TBD
CFS	79.40%	79.40%	TBD

Firstly, note that the ZeroR algorithm produces a precision and recall of 0 as it the algorithm labels all the observations as 0's since that is the majority class. Hence, it is impossible for the ZeroR algorithm to get a true positive. It is then noted that the results of the tables are quite similar to that of the accuracy tables and hence the accuracy is indeed a good metric to use.

It is worthwhile pointing out that the fact that as the recall figures are quite high is promising as this means that the number of false negatives are relatively low and therefore patients with diabetes are being identified correctly. Here, we put priority on models with a high recall score rather than precision as we are interested in minimising false negatives as they are most costly decision. In other words, we place priority on being able to identify people with diabetes than those who don't. From the table, we can see that the decision tree applied on the CFS dataset gives us the highest recall. Again, we see that the decision tree, which also has the highest precision score, are the most promising models from this study.

It is also possible to construct confusion matrices in order to see the true/false positive/negative rates and from that, build cost matrices whereby we compute costs for making a true/false positive/negative classification. From the cost matrix, it is possible to compute the "total cost" from all the classifications made and hence put a quantitative cost to the classifications that were made. A future analysis that could be done includes computing the receiver operating characteristic (ROC) curve and analysing the ratio of the true positive and false positive rate.

Conclusion

From the study, it shows that the decision tree on CFS datasets outperformed the other models significantly. This can most likely be attributed to the fact that the CFS data allows for the decision tree to be constructed from "relevant" features. Additionally, such a tree may be quite robust as it is built on fewer variables and therefore can generalise well to unseen dataset as it has not overfitted on irrelevant features. The worst performing model is the ZeroR model as it simply labels all the observations depending on the majority class in the dataset and hence will not generalise well. In general, feature selection of the dataset via CFS does lead to slight improvements in the majority of the models but more rigorous statistical testing needs to be undertaken. Therefore, we see that whilst CFS has improved the performance of our best model, unintuitively, it appears to not have a significant effect for models as a whole. Regardless, the study showed that we were able to build models that could predict those with diabetes relatively accurately.

For future work that can be conducted, it is worth noting the class imbalances whereby we have nearly twice as many "no" instances than yes and hence may skew the model building section. Resultantly, a suggestion could be to rectify the class imbalance by reducing the number of "no" instances or deploying resampling models. Another possible suggestion would be to employ models which weighs the penalty of a false positive significantly less compared to a false negative as not detecting a patient that has diabetes is life threatening. We can also use other feature selection techniques such as ridge, lasso regression, and stepwise regression in order to generate different features to be used. Another technique that was not utilised was hyperparemeter tuning for the models which may be worthwhile exploring in order to fully optimise each model. Additionally, we can try out new and novel models as well on the dataset and undertake feature engineering to improve our models' predictive capabilities. There is still much to be done but also much potential that can be realised through future studies.

Reflection

This assignment has allowed us to understand the intricacies of much of the machine learning models currently deployed. Rather than importing sklearn packages, it was a good excercise in actually implementing models from scratch and in addition, building the cross validation method to test for the accuracy. Furthermore, it was a good assignment as it allowed us to consolidate all the knowledge from the course. It was a great opportunity to implement and put in practice what we learned and also to collaborate as a team in building models.

Appendix

Decision Trees

NORM PRUNED
J48 pruned tree

```
plasma_gluc = high
|   bmi = high
|   |   tfs_thickness = high: yes (119.0/51.0)
|   |   tfs_thickness = low: no (13.0/4.0)
|   bmi = low: no (29.0/4.0)
plasma_gluc = low: no (192.0/14.0)
plasma_gluc = very high: yes (122.0/24.0)
plasma_gluc = medium
|   age = high
|   |   bmi = high
|   |   |   dp_func = high: yes (37.0/10.0)
|   |   |   dp_func = low: no (80.0/33.0)
|   |   bmi = low: no (30.0/3.0)
|   age = low: no (146.0/17.0)
```

NORM UNPRUNED
J48 unpruned tree

```
plasma_gluc = high
|   bmi = high
|   |   tfs_thickness = high
|   |   |   num_preg = low
|   |   |   |   dp_func = high
|   |   |   |   |   age = high: yes (16.0/5.0)
|   |   |   |   |   age = low
|   |   |   |   |   |   blood_pressure = high: yes (11.0/5.0)
|   |   |   |   |   |   blood_pressure = low: no (5.0/2.0)
|   |   |   |   dp_func = low
|   |   |   |   |   blood_pressure = high: no (43.0/19.0)
|   |   |   |   |   blood_pressure = low: yes (10.0/4.0)
|   |   |   num_preg = high
|   |   |   |   blood_pressure = high: yes (29.0/8.0)
|   |   |   |   blood_pressure = low
```

```
|   |   |   |   dp_func = high: no (2.0)
|   |   |   |   dp_func = low: yes (3.0)
|   |   tfs_thickness = low: no (13.0/4.0)
|   bmi = low: no (29.0/4.0)
plasma_gluc = low
|   bmi = high
|   |   2hr_ins = high
|   |   |   age = high
|   |   |   |   dp_func = high: yes (7.0/3.0)
|   |   |   |   dp_func = low: no (28.0/4.0)
|   |   |   age = low: no (43.0/4.0)
|   |   2hr_ins = low: no (48.0/2.0)
|   bmi = low: no (66.0)
plasma_gluc = very high
|   2hr_ins = high
|   |   bmi = high: yes (103.0/16.0)
|   |   bmi = low
|   |   |   age = high: yes (12.0/3.0)
|   |   |   age = low: no (4.0/1.0)
|   2hr_ins = low: no (3.0/1.0)
plasma_gluc = medium
|   age = high
|   |   2hr_ins = high
|   |   |   bmi = high
|   |   |   |   dp_func = high: yes (37.0/10.0)
|   |   |   |   dp_func = low
|   |   |   |   |   blood_pressure = high: no (57.0/24.0)
|   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   tfs_thickness = high: yes (15.0/7.0)
|   |   |   |   |   |   tfs_thickness = low: no (3.0/1.0)
|   |   |   bmi = low: no (27.0/3.0)
|   |   2hr_ins = low: no (8.0)
|   age = low
|   |   bmi = high
|   |   |   num_preg = low
|   |   |   |   tfs_thickness = high
|   |   |   |   |   dp_func = high
|   |   |   |   |   |   blood_pressure = high: no (17.0/2.0)
|   |   |   |   |   |   blood_pressure = low: yes (7.0/3.0)
|   |   |   |   |   dp_func = low: no (54.0/8.0)
|   |   |   |   tfs_thickness = low: no (24.0/1.0)
```

| | | num_preg = high: yes (2.0/1.0)
| | bmi = low: no (42.0/1.0)

CFS PRUNED
J48 pruned tree
---

plasma_gluc = high
| bmi = high
| | age = high: yes (82.0/31.0)
| | age = low: no (50.0/21.0)
| bmi = low: no (29.0/4.0)
plasma_gluc = low: no (192.0/14.0)
plasma_gluc = very high: yes (122.0/24.0)
plasma_gluc = medium
| age = high
| | bmi = high
| | | dp_func = high: yes (37.0/10.0)
| | | dp_func = low: no (80.0/33.0)
| | bmi = low: no (30.0/3.0)
| age = low: no (146.0/17.0)

CFS UNPRUNED
J48 unpruned tree
---

plasma_gluc = high
| bmi = high
| | age = high: yes (82.0/31.0)
| | age = low: no (50.0/21.0)
| bmi = low: no (29.0/4.0)
plasma_gluc = low
| bmi = high
| | 2hr_ins = high
| | | age = high
| | | | dp_func = high: yes (7.0/3.0)
| | | | dp_func = low: no (28.0/4.0)
| | | age = low: no (43.0/4.0)
| | 2hr_ins = low: no (48.0/2.0)
| bmi = low: no (66.0)
plasma_gluc = very high
| 2hr_ins = high

```
|   |   bmi = high: yes (103.0/16.0)
|   |   bmi = low
|   |   |   age = high: yes (12.0/3.0)
|   |   |   age = low: no (4.0/1.0)
|   2hr_ins = low: no (3.0/1.0)
plasma_gluc = medium
|   age = high
|   |   bmi = high
|   |   |   dp_func = high: yes (37.0/10.0)
|   |   |   dp_func = low: no (80.0/33.0)
|   |   bmi = low: no (30.0/3.0)
|   age = low: no (146.0/17.0)
```

NORM MyDT

```
plasma_gluc = high
|   bmi = high
|   |   age = high
|   |   |   dp_func = high
|   |   |   |   blood_pressure = high
|   |   |   |   |   num_preg = high
|   |   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   |   2hr_ins = high: yes
|   |   |   |   |   |   |   2hr_ins = low: yes
|   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   |   |   num_preg = low
|   |   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   |   2hr_ins = high: yes
|   |   |   |   |   |   |   2hr_ins = low: yes
|   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   |   blood_pressure = low
|   |   |   |   |   num_preg = high: no
|   |   |   |   |   num_preg = low
|   |   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   |   2hr_ins = high: yes
|   |   |   |   |   |   |   2hr_ins = low: yes
|   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   dp_func = low
|   |   |   |   2hr_ins = high
|   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   blood_pressure = high
```

```
|   |   |   |   |   |   |   num_preg = high: yes
|   |   |   |   |   |   |   num_preg = low: no
|   |   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   |   num_preg = high: yes
|   |   |   |   |   |   |   num_preg = low: yes
|   |   |   |   |   tfs_thickness = low
|   |   |   |   |   |   num_preg = high: no
|   |   |   |   |   |   num_preg = low
|   |   |   |   |   |   |   blood_pressure = high: yes
|   |   |   |   |   |   |   blood_pressure = low: no
|   |   |   |   2hr_ins = low: yes
|   |   age = low
|   |   |   tfs_thickness = high
|   |   |   |   dp_func = high
|   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   num_preg = high: yes
|   |   |   |   |   |   num_preg = low
|   |   |   |   |   |   |   2hr_ins = high: yes
|   |   |   |   |   |   |   2hr_ins = low: yes
|   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   num_preg = high: no
|   |   |   |   |   |   num_preg = low
|   |   |   |   |   |   |   2hr_ins = high: no
|   |   |   |   |   |   |   2hr_ins = low: no
|   |   |   |   dp_func = low
|   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   num_preg = high: no
|   |   |   |   |   |   num_preg = low
|   |   |   |   |   |   |   2hr_ins = high: no
|   |   |   |   |   |   |   2hr_ins = low: no
|   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   num_preg = high: yes
|   |   |   |   |   |   num_preg = low
|   |   |   |   |   |   |   2hr_ins = high: yes
|   |   |   |   |   |   |   2hr_ins = low: yes
|   |   |   tfs_thickness = low
|   |   |   |   blood_pressure = high: no
|   |   |   |   blood_pressure = low
|   |   |   |   |   2hr_ins = high
|   |   |   |   |   |   dp_func = high
|   |   |   |   |   |   |   num_preg = high: yes
```

```
|   |   |   |   |   |   num_preg = low: yes
|   |   |   |   |   dp_func = low: no
|   |   |   |   2hr_ins = low: no
|   bmi = low
|   |   tfs_thickness = high
|   |   |   2hr_ins = high
|   |   |   |   dp_func = high: no
|   |   |   |   dp_func = low
|   |   |   |   |   age = high
|   |   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   |   num_preg = high: no
|   |   |   |   |   |   |   num_preg = low: no
|   |   |   |   |   |   blood_pressure = low: no
|   |   |   |   |   age = low
|   |   |   |   |   |   blood_pressure = high: no
|   |   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   |   num_preg = high: yes
|   |   |   |   |   |   |   num_preg = low: yes
|   |   |   2hr_ins = low
|   |   |   |   dp_func = high: yes
|   |   |   |   dp_func = low: no
|   |   tfs_thickness = low: no
plasma_gluc = low
|   bmi = high
|   |   2hr_ins = high
|   |   |   age = high
|   |   |   |   dp_func = high
|   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   num_preg = high
|   |   |   |   |   |   |   tfs_thickness = high: yes
|   |   |   |   |   |   |   tfs_thickness = low: no
|   |   |   |   |   |   num_preg = low
|   |   |   |   |   |   |   tfs_thickness = high: no
|   |   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   |   |   blood_pressure = low: yes
|   |   |   |   dp_func = low
|   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   num_preg = high
|   |   |   |   |   |   |   blood_pressure = high: no
|   |   |   |   |   |   |   blood_pressure = low: no
|   |   |   |   |   |   num_preg = low
```

```
|   |   |   |   |   |   |   blood_pressure = high: no
|   |   |   |   |   |   |   blood_pressure = low: no
|   |   |   |   |   tfs_thickness = low: no
|   |   |   age = low
|   |   |   |   blood_pressure = high: no
|   |   |   |   blood_pressure = low
|   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   dp_func = high
|   |   |   |   |   |   |   num_preg = high: no
|   |   |   |   |   |   |   num_preg = low: no
|   |   |   |   |   |   dp_func = low
|   |   |   |   |   |   |   num_preg = high: no
|   |   |   |   |   |   |   num_preg = low: no
|   |   |   |   |   tfs_thickness = low: no
|   |   2hr_ins = low
|   |   |   blood_pressure = high
|   |   |   |   age = high: no
|   |   |   |   age = low
|   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   dp_func = high: yes
|   |   |   |   |   |   dp_func = low
|   |   |   |   |   |   |   num_preg = high: no
|   |   |   |   |   |   |   num_preg = low: no
|   |   |   |   |   tfs_thickness = low: no
|   |   |   blood_pressure = low: no
|   bmi = low: no
plasma_gluc = medium
|   age = high
|   |   bmi = high
|   |   |   dp_func = high
|   |   |   |   num_preg = high: yes
|   |   |   |   num_preg = low
|   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   |   2hr_ins = high: yes
|   |   |   |   |   |   |   2hr_ins = low: yes
|   |   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   |   2hr_ins = high: yes
|   |   |   |   |   |   |   2hr_ins = low: yes
|   |   |   |   |   tfs_thickness = low: yes
|   |   |   dp_func = low
```

```
|   |   |   |   2hr_ins = high
|   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   num_preg = high
|   |   |   |   |   |   |   tfs_thickness = high: no
|   |   |   |   |   |   |   tfs_thickness = low: no
|   |   |   |   |   |   num_preg = low
|   |   |   |   |   |   |   tfs_thickness = high: no
|   |   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   |   num_preg = high: yes
|   |   |   |   |   |   |   num_preg = low: yes
|   |   |   |   |   |   tfs_thickness = low
|   |   |   |   |   |   |   num_preg = high: no
|   |   |   |   |   |   |   num_preg = low: no
|   |   |   |   2hr_ins = low: no
|   |   bmi = low
|   |   |   blood_pressure = high
|   |   |   |   num_preg = high: no
|   |   |   |   num_preg = low
|   |   |   |   |   dp_func = high: no
|   |   |   |   |   dp_func = low
|   |   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   |   2hr_ins = high: no
|   |   |   |   |   |   |   2hr_ins = low: no
|   |   |   |   |   |   tfs_thickness = low: no
|   |   |   blood_pressure = low
|   |   |   |   num_preg = high: yes
|   |   |   |   num_preg = low
|   |   |   |   |   tfs_thickness = high: no
|   |   |   |   |   tfs_thickness = low
|   |   |   |   |   |   2hr_ins = high
|   |   |   |   |   |   |   dp_func = high: no
|   |   |   |   |   |   |   dp_func = low: no
|   |   |   |   |   |   2hr_ins = low: no
|   age = low
|   |   bmi = high
|   |   |   tfs_thickness = high
|   |   |   |   num_preg = high
|   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   2hr_ins = high
```

```
|   |   |   |   |   |   |   dp_func = high: yes
|   |   |   |   |   |   |   dp_func = low: yes
|   |   |   |   |   |   2hr_ins = low: yes
|   |   |   |   |   blood_pressure = low: yes
|   |   |   |   num_preg = low
|   |   |   |   |   dp_func = high
|   |   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   |   2hr_ins = high: no
|   |   |   |   |   |   |   2hr_ins = low: no
|   |   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   |   2hr_ins = high: yes
|   |   |   |   |   |   |   2hr_ins = low: yes
|   |   |   |   |   dp_func = low
|   |   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   |   2hr_ins = high: no
|   |   |   |   |   |   |   2hr_ins = low: no
|   |   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   |   2hr_ins = high: no
|   |   |   |   |   |   |   2hr_ins = low: no
|   |   |   tfs_thickness = low
|   |   |   |   dp_func = high
|   |   |   |   |   blood_pressure = high: no
|   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   2hr_ins = high
|   |   |   |   |   |   |   num_preg = high: no
|   |   |   |   |   |   |   num_preg = low: no
|   |   |   |   |   |   2hr_ins = low: no
|   |   |   |   dp_func = low: no
|   |   bmi = low
|   |   |   dp_func = high
|   |   |   |   2hr_ins = high: no
|   |   |   |   2hr_ins = low
|   |   |   |   |   blood_pressure = high: no
|   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   num_preg = high: yes
|   |   |   |   |   |   num_preg = low
|   |   |   |   |   |   |   tfs_thickness = high: yes
|   |   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   dp_func = low: no
plasma_gluc = very high
|   2hr_ins = high
```

```
|   |   bmi = high
|   |   |   num_preg = high
|   |   |   |   dp_func = high: yes
|   |   |   |   dp_func = low
|   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   |   age = high: yes
|   |   |   |   |   |   |   age = low: yes
|   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   |   age = high: yes
|   |   |   |   |   |   |   age = low: yes
|   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   num_preg = low
|   |   |   |   age = high
|   |   |   |   |   dp_func = high
|   |   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   |   blood_pressure = high: yes
|   |   |   |   |   |   |   blood_pressure = low: yes
|   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   |   |   dp_func = low
|   |   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   |   tfs_thickness = high: yes
|   |   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   |   tfs_thickness = high: yes
|   |   |   |   |   |   |   tfs_thickness = low: yes
|   |   |   |   age = low
|   |   |   |   |   dp_func = high: yes
|   |   |   |   |   dp_func = low
|   |   |   |   |   |   tfs_thickness = high
|   |   |   |   |   |   |   blood_pressure = high: yes
|   |   |   |   |   |   |   blood_pressure = low: yes
|   |   |   |   |   |   tfs_thickness = low
|   |   |   |   |   |   |   blood_pressure = high: yes
|   |   |   |   |   |   |   blood_pressure = low: no
|   |   bmi = low
|   |   |   age = high
|   |   |   |   tfs_thickness = high
|   |   |   |   |   num_preg = high
```

```
|   |   |   |   |   |   dp_func = high
|   |   |   |   |   |   |   blood_pressure = high: yes
|   |   |   |   |   |   |   blood_pressure = low: yes
|   |   |   |   |   |   dp_func = low: yes
|   |   |   |   |   num_preg = low
|   |   |   |   |   |   blood_pressure = high
|   |   |   |   |   |   |   dp_func = high: yes
|   |   |   |   |   |   |   dp_func = low: yes
|   |   |   |   |   |   blood_pressure = low
|   |   |   |   |   |   |   dp_func = high: yes
|   |   |   |   |   |   |   dp_func = low: yes
|   |   |   |   tfs_thickness = low: yes
|   |   |   age = low
|   |   |   |   blood_pressure = high
|   |   |   |   |   tfs_thickness = high: no
|   |   |   |   |   tfs_thickness = low
|   |   |   |   |   |   num_preg = high: yes
|   |   |   |   |   |   num_preg = low
|   |   |   |   |   |   |   dp_func = high: yes
|   |   |   |   |   |   |   dp_func = low: yes
|   |   |   |   blood_pressure = low: no
|   2hr_ins = low
|   |   dp_func = high: yes
|   |   dp_func = low: no
```

CFS MyDT

```
plasma_gluc = 'very high'
|   2hr_ins = high
|   |   bmi = high
|   |   |   age = high
|   |   |   |   dp_func = high: yes
|   |   |   |   dp_func = low: yes
|   |   |   age = low
|   |   |   |   dp_func = high: yes
|   |   |   |   dp_func = low: yes
|   |   bmi = low
|   |   |   age = high
|   |   |   |   dp_func = high: yes
|   |   |   |   dp_func = low: yes
|   |   |   age = low
```

```
|   |   |   |   dp_func = high: no
|   |   |   |   dp_func = low: no
|   2hr_ins = low
|   |   dp_func = high: yes
|   |   dp_func = low: no
plasma_gluc = high
|   bmi = high
|   |   age = high
|   |   |   dp_func = high
|   |   |   |   2hr_ins = high: yes
|   |   |   |   2hr_ins = low: yes
|   |   |   dp_func = low
|   |   |   |   2hr_ins = high: yes
|   |   |   |   2hr_ins = low: yes
|   |   age = low
|   |   |   2hr_ins = high
|   |   |   |   dp_func = high: no
|   |   |   |   dp_func = low: no
|   |   |   2hr_ins = low: no
|   bmi = low
|   |   2hr_ins = high
|   |   |   dp_func = high: no
|   |   |   dp_func = low
|   |   |   |   age = high: no
|   |   |   |   age = low: no
|   |   2hr_ins = low
|   |   |   dp_func = high
|   |   |   |   age = high: yes
|   |   |   |   age = low: no
|   |   |   dp_func = low: no
plasma_gluc = low
|   bmi = high
|   |   2hr_ins = high
|   |   |   age = high
|   |   |   |   dp_func = high: yes
|   |   |   |   dp_func = low: no
|   |   |   age = low
|   |   |   |   dp_func = high: no
|   |   |   |   dp_func = low: no
|   |   2hr_ins = low
|   |   |   age = high: no
```

```
|   |   |   age = low
|   |   |   |   dp_func = high: no
|   |   |   |   dp_func = low: no
|   bmi = low: no
plasma_gluc = medium
|   age = high
|   |   bmi = high
|   |   |   dp_func = high
|   |   |   |   2hr_ins = high: yes
|   |   |   |   2hr_ins = low: yes
|   |   |   dp_func = low
|   |   |   |   2hr_ins = high: no
|   |   |   |   2hr_ins = low: no
|   |   bmi = low
|   |   |   2hr_ins = high
|   |   |   |   dp_func = high: no
|   |   |   |   dp_func = low: no
|   |   |   2hr_ins = low: no
|   age = low
|   |   bmi = high
|   |   |   dp_func = high
|   |   |   |   2hr_ins = high: no
|   |   |   |   2hr_ins = low: no
|   |   |   dp_func = low
|   |   |   |   2hr_ins = high: no
|   |   |   |   2hr_ins = low: no
|   |   bmi = low
|   |   |   dp_func = high
|   |   |   |   2hr_ins = high: no
|   |   |   |   2hr_ins = low: no
|   |   |   dp_func = low: no
```