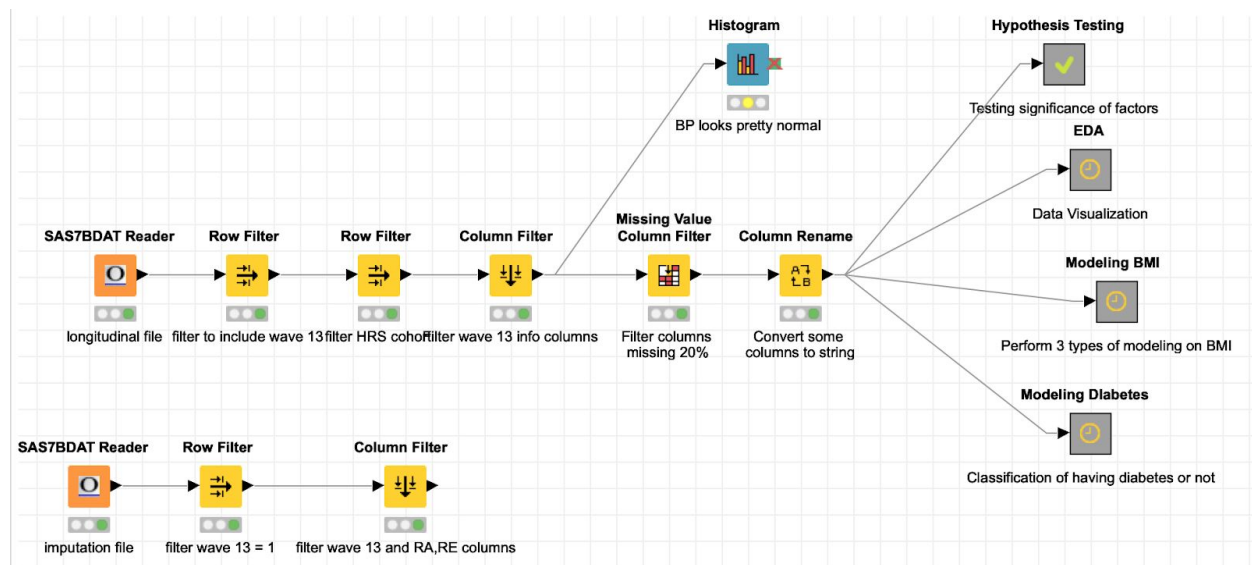


Modeling Social Determinants of Health Using KNIME Analytics Platform and Health and Retirement Study

1. Background

Health is extremely important for everyone, especially during this uncertain time of COVID-19, while certain health conditions are very good indicators for major diseases such as diabetes and cancer, it is also interesting and essential to explore social and economic factors that could potentially indicate the likelihood of a certain health problem or a particular disease. In my research project, I used KNIME Analytics Platform to build predictive models for different health outcomes, with the first two being modeling BMI and diabetes.



2. Methods

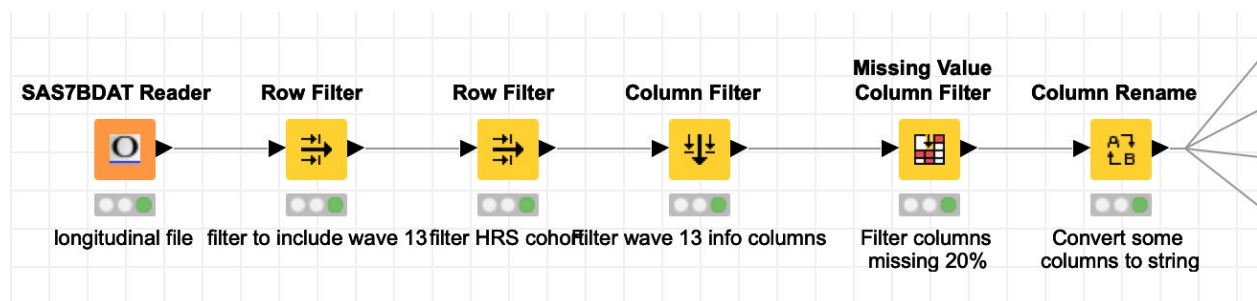
2.1 Data

The dataset was from the Health and Retirement Study Longitudinal File, particularly, I chose the set of participants who belonged to the HRS cohort (respondents born from 1931 to 1941), the interview process started in 1992 and the file contained information from each wave of interview (and with biennial follow-up), for this project, I selected only variables collected during wave 13 (2016, identified by 13 in the variable naming convention) for simplicity of my

initial modeling. Two dependent variables have names “R13BMI” and “R13DIAB”, and they will be used as target variables during modeling. Other major variables include education level, gender, religion, different assets, security, insurance, family structure, etc, and they will be discussed in more detail in the next section.

2.2 Data Preparation using KNIME

I used several nodes to perform data processing. First of all, a SAS7BDAT Reader node was used to extract the dataset, since the original dataset was a SAS table. Then, row filters and column filters were used to extract wave and cohort specific information. In the configuration setting of the first row filter, I used “include rows by attribute value option” to select rows whose “INW13” (indicator whether the respondent was in wave 13) equals 1.0. In the second row filter, I used the same option with letting the column “HACOHORT” (indicator of which cohort) to be 3.0, such that only the HRS cohort was selected. In the following column filter, a regex expression was used to select only the columns whose names contain “13”. After all the necessary information were retrieved, I used a missing value filter to select the columns whose percentages of missing values were below 20%, to ensure the quality of modeling. Finally I used “Column Rename” node to convert some numeric variables to strings, since initially all categorical variables were interpreted as numeric variables, and there were several nominal categorical variables with more than two levels, it would be more accurate to use one-hot encoding rather than using the ordinal values.

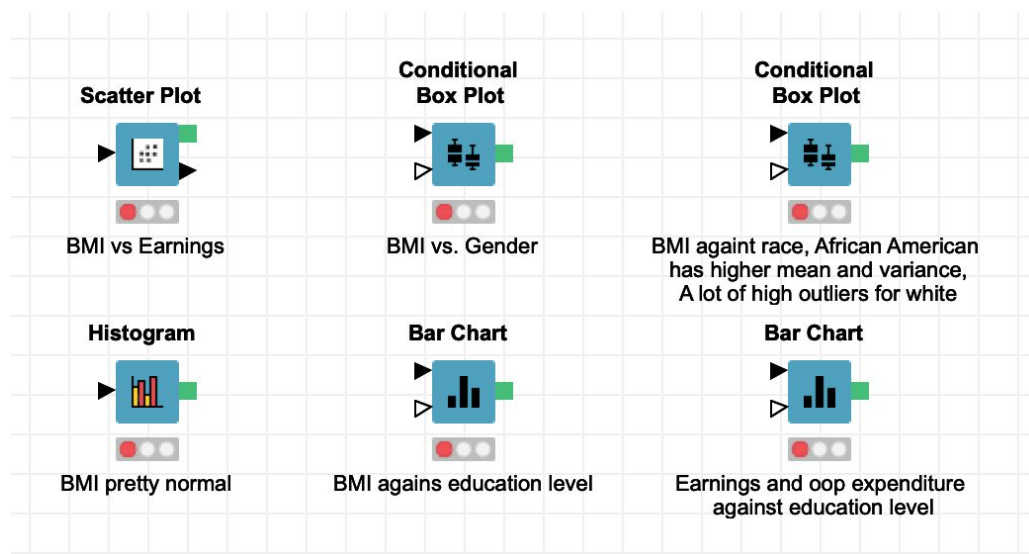


2.3 Data Visualization with BMI

In order to get more information about how different variables can help predicting BMI, visualizations were performed with BMI being the dependent variable and some categorical variables being the groups. I used the “Histogram” node to plot BMI alone in the first step,

since generally regression models might require some assumptions such as normality, and it turned out that BMI looked pretty normal, so no transformation of the data was needed.

I chose a set of categorical variables which could have a potential effect on BMI based on intuition, those included gender (RAGENDER), education level (RAEDUC) and ethnicity (RARACEM). Conditional boxplot nodes were used since it could help visualize the center and skewness of the distribution, and bar plot was also used to plot BMI (R13BMI) vs. gender (average BMI was used as y-axis). Configuration only allows string or categorical columns, so these were all transformed into string columns in the previous step. In addition, a scatter plot node was used to visualize the relationship between BMI and individual earnings (R13IEARN).



The interface shows the following configuration:

- Category Column**: RARACEM
- Selection Method**: ☒ Manual Selection
- Filter Panel (Left)**:
 - Filter: [Empty]
 - Columns: HHIDPN, S13HHIDPN, R13MPART, INW13, RASPID1, RASPCT, R13MRCT, R13MLEN, R13MLENM, R13MDIV, R13MWID
 - ☐ Enforce exclusion
- Filter Panel (Right)**:
 - Filter: [Empty]
 - Columns: R13BMI
 - ☒ Enforce inclusion
- Selected Column**: R13BMI

Example Configuration Window of Conditional Boxplot Node: BMI vs. Race

Maximum number of rows:

Selection column name:

Choose column for x axis

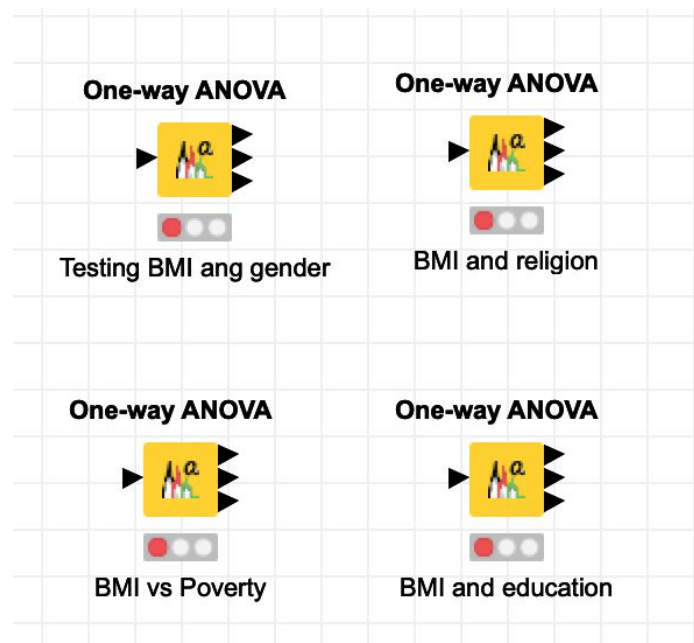
Choose column for y axis

☒ Report on missing values

Example Configuration Window of the Scatter Plot Node: BMI vs. Earnings (R13IEARN)

2.4 Hypothesis Testing with BMI:

Though visualizations can give some ideas about the significance of one factor on the level of BMI, statistical testing can give better explanation. Given that the normality assumption was satisfied in BMI, one-way ANOVA tests were performed with BMI against education level, gender, poverty status (INPOV) and religion (RARELIG).



One-way ANOVA Node: BMI as numeric values and categorical variables as grouping column

2.5 Data Visualization with Diabetes:

Similar visualizations were chosen for diabetes, first, a single bar chart plotting the relative frequencies of diabetes was used. One thing to note about diabetes data was that only values of 0 and 1 were used in the table, so the diabetes variable (R13DIAB) became a binary variable for classification. Another bar chart node was used to visualize the relationship between diabetes and the number of children in the household. The configuration window was similar to that of the BMI bar chart.

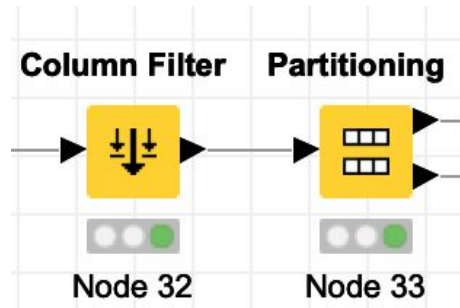
2.6 Hypothesis Testing with Diabetes:

Since diabetes is a categorical variable and the grouping variables are also categorical, chi-square test was used to analyze the relationship between diabetes and the grouping variables, specifically, they included: education (RAEDUC), marital status (MSTAT), gender (RAGENDER), race (RARACEM) and poverty (INPOV). In the configuration setting, row was the grouping variable, column was the diabetes indicator.

The image shows a configuration window with three tabs: "Settings" (selected), "Flow Variables", and "Memory Policy". The "Settings" tab contains the following options:

- Row variable: A dropdown menu with "S RAEDUC" selected and a blue arrow icon on the right.
- Column variable: A dropdown menu with "S R13DIAB" selected and a blue arrow icon on the right.
- Weight column: A dropdown menu with "? <none>" selected and a blue arrow icon on the right.
- ☐ Enable hiliting

2.7 Modeling BMI:



Regression node learner's configuration was set to ignore rows with missing values, so that the training was not disrupted by missing entries. For each indicator variable, one level of the indicator was removed when configuring, since linear dependence should be avoided.

Target
D R13BMI

Values
☒ Manual Selection ☐ Wildcard/Regex Selection

Exclude
Filter
 ? 1.0_RARELIG
 ? 0.0_R13HEART
 ? 4.0_R13HEART
 ? 1.0_R13HEART
 ? 3.0_R13HEART
 ? 0.0_R13HOSP
 ? 0.0_H13INPOV
 ? 1.0_RASSRECV
 ? 0.0_R13LIFEIN
 ? 0.0_R13PENINC
☒ Enforce exclusion

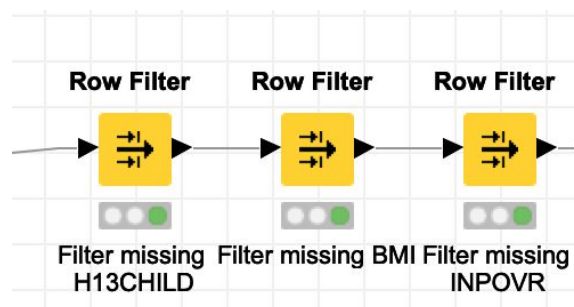
Include
Filter
 D RABYEAR
 D H13ACHCK
 D H13AHOUS
 D H13ATOTW
 D H13CHILD
 I 1.0_R13MNEV
 I 0.0_H13ANYFIN
 I 1.0_H13ANYFAM
 I 0.0_H13CPL
 I 2.0_RAGENDER
☐ Enforce inclusion

Regression Properties
☐ Predefined Offset Value: 0

Missing Values in Input Data
☒ Ignore rows with missing values.
☐ Fail on observing missing values.

Scatter Plot View
First Row: 1
Row Count: 20,000

For the simple regression tree predictor node, it failed on having missing values, so additional three row filters were used on the test set before connecting to the tree regression predictor node. **Learner node configuration**



Target Column D R13BMI

Attribute Selection

☐ Use fingerprint attribute <no valid fingerprint input>

☒ Use column attributes

☒ Manual Selection ☐ Wildcard/Regex Selection

Exclude

Filter

- ? 0.0_R13HEART
- ? 4.0_R13HEART
- ? 1.0_R13HEART
- ? 3.0_R13HEART
- ? 0.0_R13HOSP
- ? 0.0_H13INPOV
- ? 1.0_RASSRECV

☒ Enforce exclusion

Include

Filter

- D RABYEAR
- D R13OOPMD
- D H13ACHCK
- D H13AHOUS
- D H13ATOTW
- D H13CHILD
- I 1.0_R13MNEV

☐ Enforce inclusion

Misc Options

☐ Ignore columns without domain information

☐ Enable Hilighting (#patterns to store) 2,000

Tree Options

☒ Use binary splits for nominal attributes

Missing value handling XGBoost

☐ Limit number of levels (tree depth) 10

☐ Minimum split node size 1

☐ Minimum node size 5

Configuration of the Regression Tree Learner Node

Last type of model was a polynomial regression model, it was very similar to the linear regression model except the choice of polynomial degree of the variables, I chose 2 as the degree to see if there were non-linear associations while trying to avoid overfitting.

Regression settings | View settings | Flow Variables | Memory Policy

Target column (dependent variable)

Maximum polynomial degree

Select the independent variables

☒ Manual Selection ☐ Wildcard/Regex Selection ☐ Type Selection

Exclude

Filter

- ? 0.0_R13HEART
- ? 4.0_R13HEART
- ? 1.0_R13HEART
- ? 3.0_R13HEART
- ? 0.0_R13HOSP
- I 2.0_R13CENDIV
- I 5.0_R13CENDIV

☒ Enforce exclusion

Include

Filter

- D RABYEAR
- D R13OOPMD
- D H13ACHCK
- D H13AHOUS
- D H13ATOTW
- D H13INPOVR
- D H13CHILD

☐ Enforce inclusion

Missing Values in Input Data

☒ Ignore rows with missing values.

☐ Fail on observing missing values.

Configuration of the Polynomial Regression Learner, with maximum degree 2

2.8 Modeling Diabetes:

Four types of classification models were used to model diabetes. Similar to BMI, a “One to Many” node was needed to transform some string variables to one-hot indicator variables. Forward feature selection metanode was used before modeling to select top 20 features among a chosen set of social and economic variables. (Results of the selection will be discussed in later

The diagram illustrates a machine learning workflow for predicting diabetes. It begins with a 'Row Filter' (Node 61) that filters diabetes values to 0 or 1. This is followed by a 'One to Many' step (Node 62) for one-hot encoding. A 'Missing Value' step (Node 63) removes rows with missing values in chosen columns. The data then passes through a 'Column Filter' (Node 64). The workflow then branches into several paths:

- Random Forest Path:** The data is partitioned (Node 43) and used to train a 'Random Forest Learner' (Node 44). The learner's output is used by a 'Random Forest Predictor' (Node 45) and a 'Scorer' (Node 46).
- Decision Tree Path:** The data is partitioned (Node 48) and used to train a 'Decision Tree Learner' (Node 49). The learner's output is used by a 'Decision Tree Predictor' (Node 50) and a 'Scorer' (Node 51).
- Logistic Regression Path:** The data is partitioned (Node 52) and used to train a 'Logistic Regression Learner' (Node 53). The learner's output is used by a 'Logistic Regression Predictor' (Node 54) and a 'Scorer' (Node 55).
- Gradient Boosted Trees Path:** The data is partitioned (Node 56) and used to train a 'Gradient Boosted Trees Learner' (Node 57). The learner's output is used by a 'Gradient Boosted Trees Predictor' (Node 58) and a 'Scorer' (Node 59).
- Forward Feature Elimination Path:** The data is used for 'Forward Feature Elimination' (Node 60) to identify important features. This step also feeds into the 'Missing Value' step (Node 63).
- CV Path:** The data is used for 'CV' (Node 61) to evaluate model performance. This step also feeds into the 'Missing Value' step (Node 63).

The final output of the workflow is a 'Scorer' (Node 66) that evaluates the model's performance. The diagram also includes a 'Testing and Viz' step (Node 61) for visualizing the results.

KNIME workflow overview (Diabetes Metanode)

Manual Selection

Wildcard/Regex Selection

Static Columns

Filter

S R13DIAB

D R13BWC20

D R13VP

D H13AFHOU

D R13HILTC

D R13SAYRET

I 3.0_R13HLTC

I 2.0_R13HLTC

I 1.0_R13HLTC

Enforce exclusion

Variable Columns ('Features')

Filter

D R13MRCT

D R13MSTATH

D RABYEAR

D R13OOPMD

D H13ASTCK

D H13ABOND

D H13ACHCK

D H13ACD

D H13ATCAN

>

>>

<

<<

Enforce inclusion

Feature selection strategy

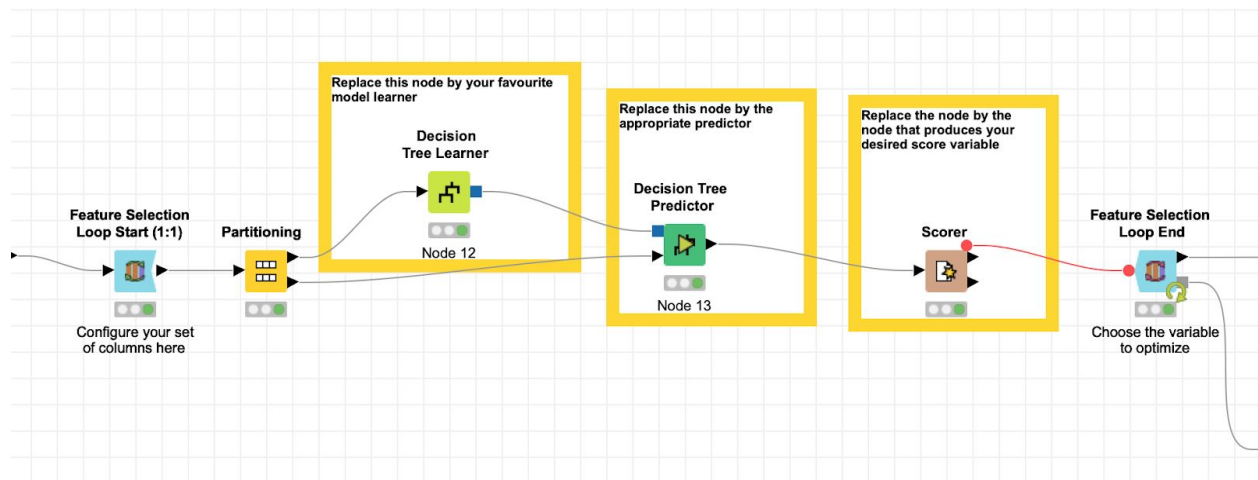
Forward Feature Selection

Sequential Algorithm Settings

☒ Use threshold for number of features

Select threshold for number of features

20



Forward Feature Selection Metanode: Decision Tree Learner was used to fit each model

Partitioning node was also used to split the training set and the test set. Decision Tree learner, Random Forest learner, Logistic regression learner and Gradient Boosting learner was used to model diabetes. All of them take in the same set of columns selected by Forward Feature Selection, which will be discussed in the Results section.

Choose size of first partition

☐ Absolute

100

☒ Relative[%]

70

☐ Take from top

☐ Linear sampling

☒ Draw randomly

☐ Stratified sampling

\$ R13DIAB

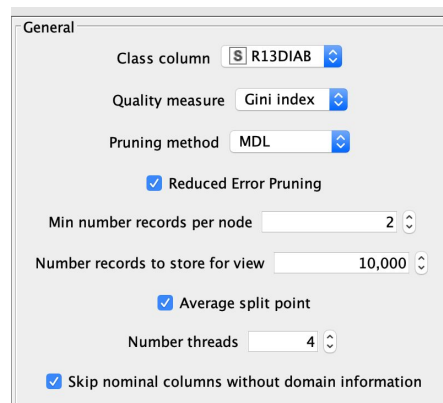
☐ Use random seed

1,595,442,200,154

Above shows the configuration of the Partition Node, by default, the test set contains 30% of the whole table, and the training set contains 70%, and they are distributed randomly.

First classification model was the decision tree, the features that were included were mostly social and economic factors, and most of them were selected based on the result of the feature selection, which will be discussed later. In general, those variables included household

structure, assets, insurance, income, social security and living condition. Gini index was used to measure the quality of the tree, and pruning was used to avoid overfitting.

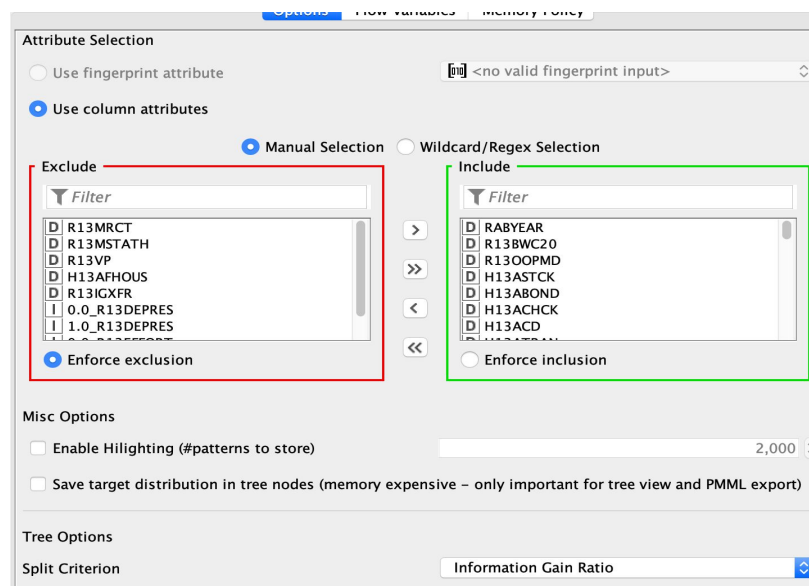


The image shows a 'General' configuration window for a Decision Tree Learner. It includes the following settings:

- Class column: R13DIAB
- Quality measure: Gini index
- Pruning method: MDL
- ☒ Reduced Error Pruning
- Min number records per node: 2
- Number records to store for view: 10,000
- ☒ Average split point
- Number threads: 4
- ☒ Skip nominal columns without domain information

Configuration Window for the Decision Tree Learner

Second model was a random forest model, which consisted of multiple trees, same set of features was used and instead of using the Gini index, another criterion called “information gain ratio” was used. Configuration window shown below.



The image shows an 'Attribute Selection' configuration window. It includes the following settings:

- Attribute Selection: ☒ Use column attributes
- Manual Selection: ☒ Manual Selection
- Exclude: ☒ Enforce exclusion
- Include: ☐ Enforce inclusion
- Misc Options: ☐ Enable Highlighting (#patterns to store): 2,000
- Tree Options: ☐ Save target distribution in tree nodes (memory expensive – only important for tree view and PMML export)
- Split Criterion: Information Gain Ratio

Third model was a logistic regression model, since the parameters needed to be solved by numerical methods, there was a solver section in the configuration window, and the default method was stochastic average gradient. However, there were some issues with convergence which will be discussed later.

Target

Target column:

Reference category:

☐ Use order from target column domain (only relevant for output representation)

Solver

Select solver:

Feature selection

☒ Manual Selection ☐ Wildcard/Regex Selection ☐ Type Selection

Exclude

Filter

- D R13MRCT
- D R13MSTATH
- D R13VP
- D H13AFHOUS
- D R13IGXFR
- I 1.0_R13MSTAT
- I 0.0_R13MNEV

☒ Enforce exclusion

Include

Filter

- D RABYEAR
- D R13BWC20
- D R13OOPMD
- D H13ASTCK
- D H13ABOND
- D H13ACHCK
- D H13ACD

☐ Enforce inclusion

☐ Use order from column domain (applies only to nominal columns). First value is chosen as reference for dummy variables.

Configuration Window of Logistic Regression

Last type of model was a gradient boosting model, which was built on weaker models like trees, the configuration window was shown below, it was very similar to the window of the random forest learner.

Target Column

Attribute Selection

☐ Use fingerprint attribute

☒ Use column attributes

☒ Manual Selection ☐ Wildcard/Regex Selection

Exclude

Filter

- D R13MRCT
- D R13MSTATH
- D R13VP
- D H13AFHOUS
- D R13IGXFR
- I 0.0_R13DEPRES
- I 1.0_R13DEPRES

☒ Enforce exclusion

Include

Filter

- D RABYEAR
- D R13BWC20
- D R13OOPMD
- D H13ASTCK
- D H13ABOND
- D H13ACHCK
- D H13ACD

☐ Enforce inclusion

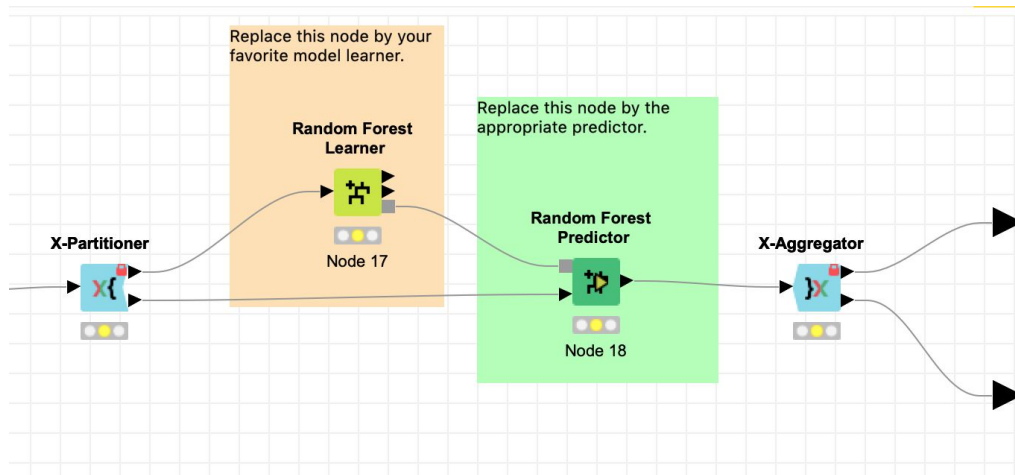
Tree Options

☒ Limit number of levels (tree depth)

Boosting Options

Number of models

Finally, cross validation was performed on all 4 models to assess the overall performance of each, since one validation set might not represent the model well.

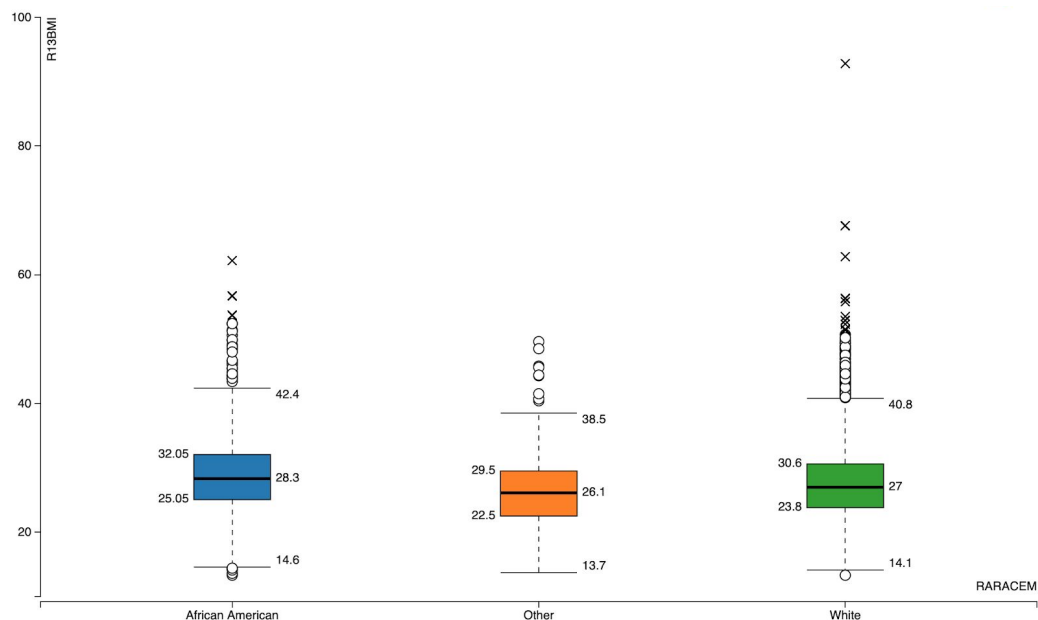


Cross Validation Metanode (Random Forest Example)

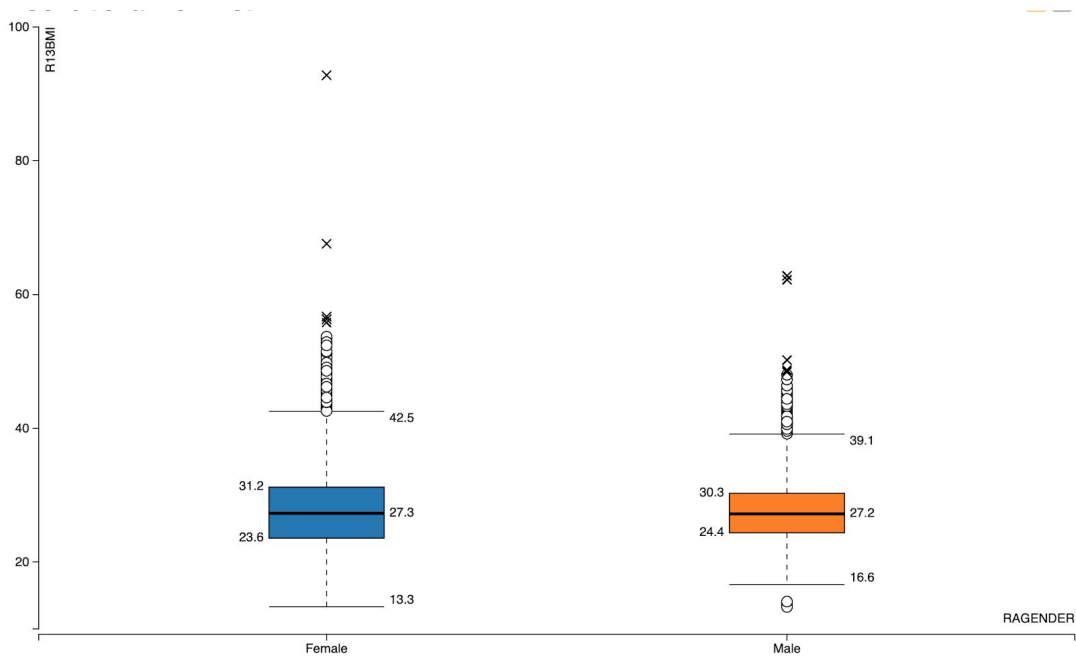
3. Results

3.1 Data Visualization with BMI:

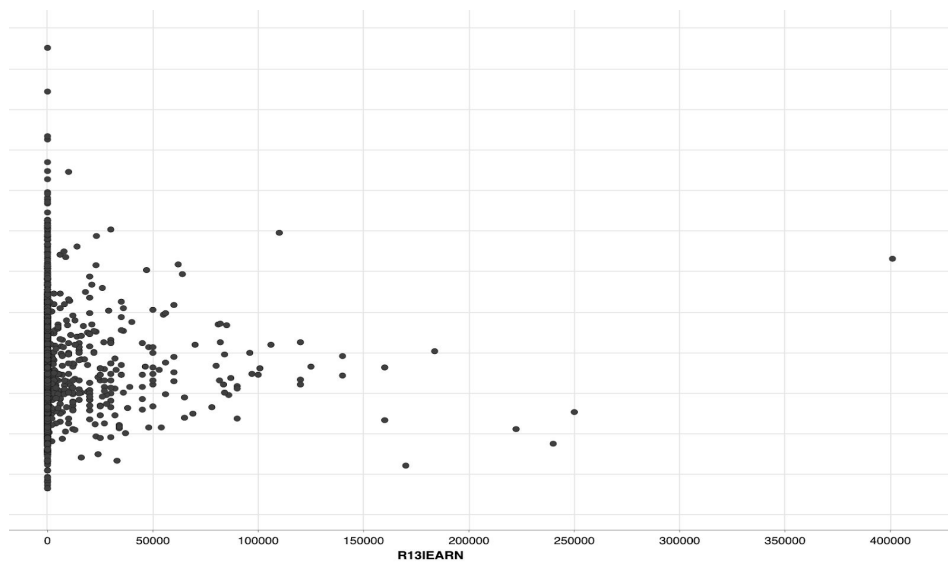
BMI vs Ethnicity: According to the conditional boxplot with different races on the x-axis and BMI on the y-axis, African Americans (2.0) seem to have the highest mean BMI and largest variance, which indicates that ethnicity can be a good indicator of BMI. Whites(1.0) seems to have a lot of outliers.



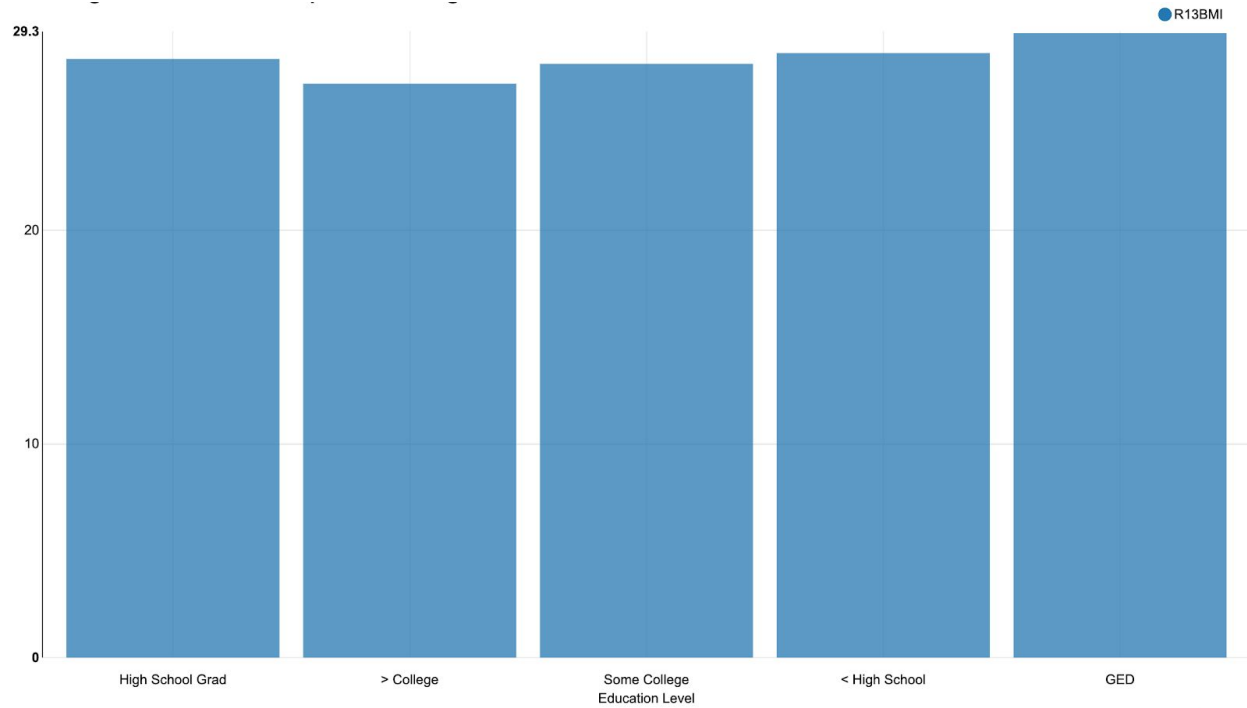
BMI vs. Gender: Male and Female have similar means, female has larger variance:



BMI vs Earnings: There is not a significant correlation between the two variables, a slightly negative line can be drawn. However, there are too many data points on the y-axis.

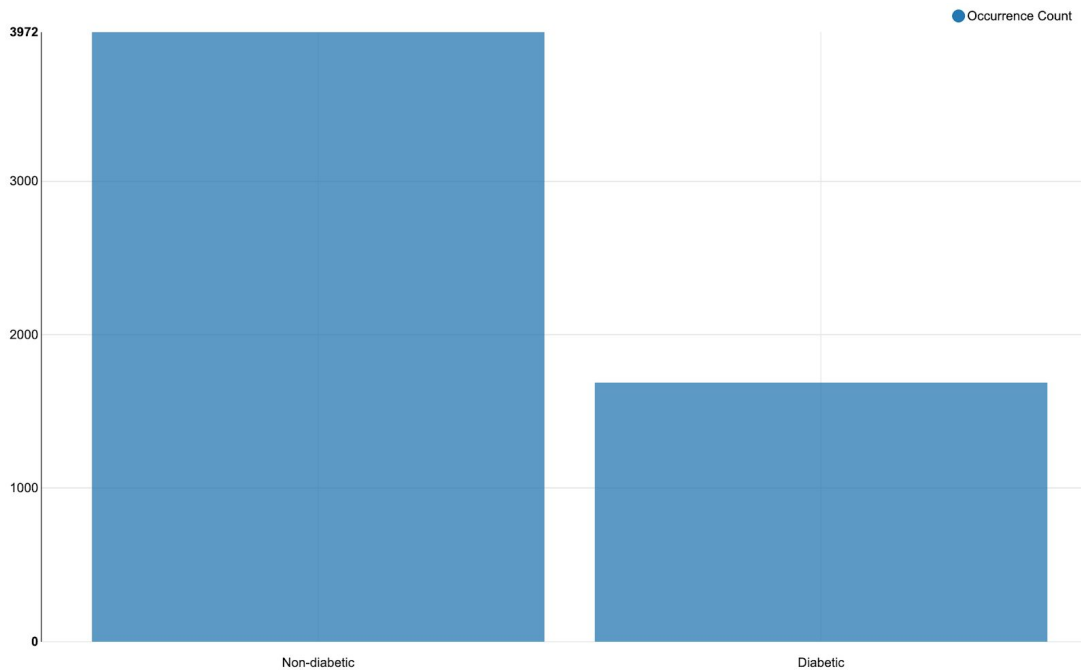


BMI vs Education level: There are differences among different education levels, with higher education levels corresponding to lower BMI



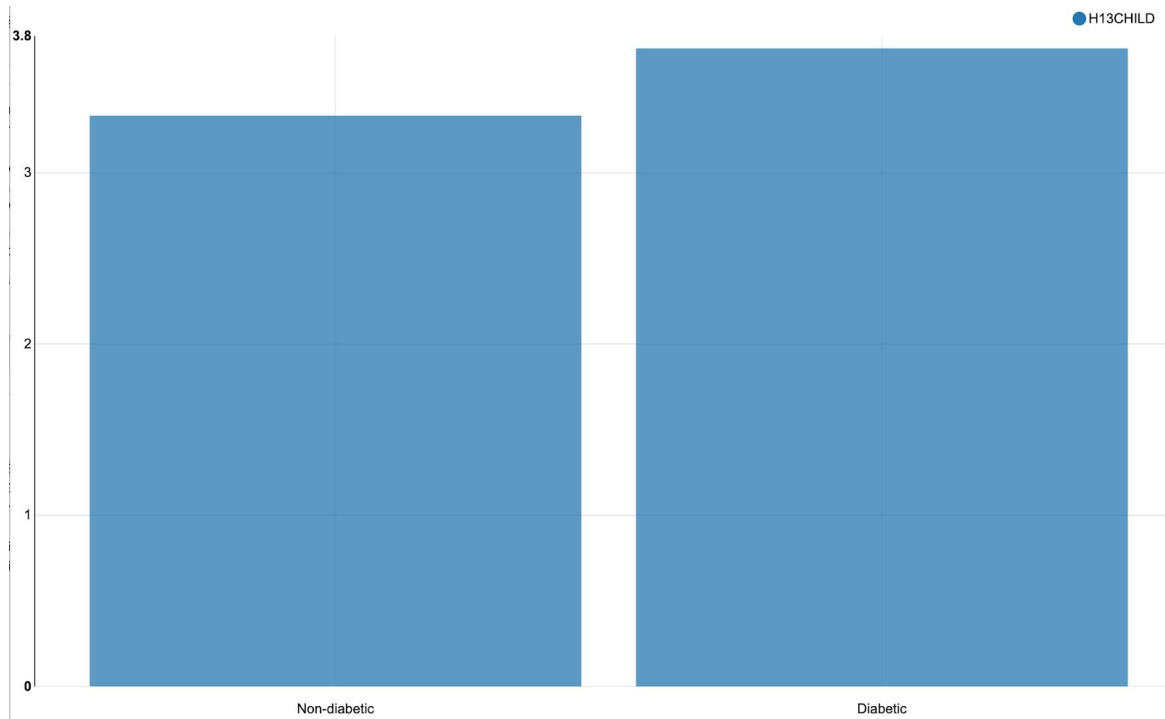
3.2 Data Visualization with Diabetes:

Overall distribution of diabetes: By plotting a bar chart, there are approximately twice as many non-diabetic participants as diabetic participants.



Diabetes vs Number of children:

There is a difference between the average number of children in the family of diabetes vs. non-diabetes, the average is higher for diabetes families.



3.3 Hypothesis Testing with BMI:

Several one-way anova tests were performed with BMI being the dependent variable, which is roughly normally distributed. Variables (groups) that are tested include gender, education and poverty status. The p-value for BMI vs poverty status (INPOV) is 0.042, indicating statistical significance. The p-value for BMI vs Gender is 0.38, which is insignificant. The p-value for BMI vs. race level is very small. The p-value for BMI vs education level is also very small. The p-value for BMI vs religion is 0.02, which is also significant.

3.4 Hypothesis Testing with Diabetes:

Chi-square Tests:

Variable	P-value	Explanation
Marital Status (R13MSTAT)	0.24	Not Significant
Poverty Status (H13INPOV)	Almost 0	Highly Statistically Significant
Gender (RAGENDER)	0.02	Significant
Ethnicity (RARACEM)	Almost 0	Highly Statistically Significant
Education (RAEDUC)	Almost 0	Highly Statistically Significant

1. Diabetes vs. Marital Status:

The p-value for the cross-tabulation of diabetes and marital status is 0.24, which is insignificant. So RAMSTAT might not be a very significant variable.

2. Diabetes vs. Poverty Status:

The p-value for the cross-tab of diabetes and poverty is very tiny, which is statistically significant, therefore poverty is a useful indicator of diabetes.

3. Diabetes vs. Gender:

The p-value for the cross-tab of diabetes and gender is 0.02, which is statistically significant, therefore gender can be a very significant factor.

4. Diabetes vs. Ethnicity:

The p-value for the cross-tab of diabetes and ethnicity is very small, which is highly statistically significant, Whites seem to have lower percentages of diabetes.

5. Diabetes vs. Education:

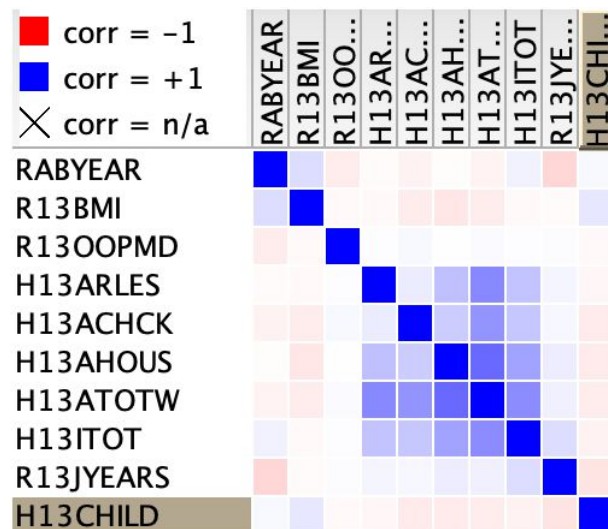
The p-value for the cross-tab of diabetes and education level is very small, which is statistically significant, therefore gender can be a very significant factor.

3.5 Modeling BMI:

According to the correlation matrix below, no variables produce a very strong correlation (>0.5) when associated with BMI data. Those variables which have a small correlation with BMI

include: birth year of the respondent (RABYEAR, 0.1318), assets on checking and savings account (H13ACHCK, -0.07), primary residence asset (H13AHOUS, -0.096), total wealth without IRA (H13ATOTW,-0.075), and the number of children (H13CHILD,0.0925).

Variables	Correlation
Birth Year (RABYEAR)	0.1318
Assets on checking (H13ACHCK)	-0.07
Primary residence asset (H13AHOUS)	-0.096
Total wealth without IRA (H13TOTW)	-0.075
Number of children (H13CHILD)	0.0925



I picked above mentioned independent variables plus some indicator variables for social and economic status, detailed variables described below.

Multiple Linear Regression:

With those selected variables, linear regression was performed and the results were shown below, we can see that for the social determinants, assets in checking account (ACHCK), asset for housing (AHOUS), number of children (H13CHILD), gender (RAGENDER) , education (RAEDUC), nursing home status(NHMLIV) are all significant predictors (p-values < 0.05).

Other variables used in this regression model included: number of insurances (MNEV), whether any financial respondent in household (ANYFIN), whether any family respondent in household (ANYFAM), whether a couple household (H13CPL), presence of vigorous, moderate and light activity (VGACTX, MDACTX, LTACTX), physical effort (R13EFFORT), health limit indicator (HLTHLIM), history of mammogram (MAMMOG), income from social security DI/SSI (SSDI), covered by R or S's employment plan (COVR, COVS), covered by government insurance program (HIGOV), receiving pension income (PENINC). Multiple R-squared was 0.0971 and adjusted R-square was 0.088. Although it was not very high given the fact that we used a limited set of variables, it did improve individual correlations, especially those variables mentioned above that have significant p-values.

Statistics on Linear Regression

Variable	Coeff.	Std. Err.	t-value	P> t
RABYEAR	0.1543	0.017	9.0932	0.0
H13ACHCK	-1.71E-6	9.46E-7	-1.8067	0.0709
H13AHOUS	-1.25E-6	4.20E-7	-2.9717	0.003
H13ATOTW	1.23E-7	1.15E-7	1.0675	0.2858
H13CHILD	0.1798	0.0421	4.2742	1.97E-5
1.0_R13MNEV	-0.8588	0.5437	-1.5797	0.1143
0.0_H13ANYFIN	-1.5377	1.4456	-1.0637	0.2875
1.0_H13ANYFAM	-0.1062	1.2437	-0.0854	0.932
0.0_H13CPL	-0.2258	0.2008	-1.1244	0.2609
2.0_RAGENDER	-0.973	0.2585	-3.7643	0.0002
3.0_RARACEM	-1.0957	0.4383	-2.4997	0.0125
2.0_RARACEM	0.9921	0.2568	3.8635	0.0001
3.0_RAEDUC	0.3566	0.2552	1.3976	0.1623
5.0_RAEDUC	-0.0551	0.3039	-0.1814	0.8561
4.0_RAEDUC	0.3012	0.284	1.0604	0.289
2.0_RAEDUC	1.1204	0.4555	2.4597	0.014
1.0_R13EFFORT	0.7441	0.2264	3.2867	0.001
2.0_R13VGACTX	-0.1019	0.5753	-0.177	0.8595
3.0_R13VGACTX	0.0398	0.6116	0.0651	0.9481
5.0_R13VGACTX	0.6501	0.5535	1.1745	0.2403
4.0_R13VGACTX	0.4885	0.6177	0.7908	0.4291
3.0_R13MDACTX	0.7525	0.3841	1.959	0.0502
2.0_R13MDACTX	0.6514	0.3551	1.8343	0.0667
5.0_R13MDACTX	2.0197	0.379	5.3287	1.05E-7
4.0_R13MDACTX	1.1722	0.414	2.8313	0.0047
3.0_R13LTACTX	0.4823	0.3655	1.3196	0.1871
2.0_R13LTACTX	-0.0371	0.3543	-0.1048	0.9165
4.0_R13LTACTX	-0.1939	0.4394	-0.4413	0.6591
5.0_R13LTACTX	-0.0876	0.4145	-0.2114	0.8326
1.0_R13MAMMOG	0.8917	0.2413	3.6947	0.0002
1.0_R13HLTHLM	0.8326	0.1885	4.4178	1.03E-5
3.0_H13NHMLIV	-1.736	0.6693	-2.5937	0.0095
2.0_H13NHMLIV	-0.097	1.562	-0.0621	0.9505
1.0_R13SSDI	0.2048	0.5186	0.3949	0.693
1.0_R13COVR	0.2287	0.2756	0.8297	0.4068
1.0_R13COVS	0.1714	0.3584	0.4784	0.6324
1.0_R13HIGOV	0.99	0.4228	2.3413	0.0193
1.0_R13PENINC	0.4793	0.1933	2.4799	0.0132
Intercept	-274.7857	33.019	-8.3221	2.22E-16
Multiple R-Squared: 0.0971				
Adjusted R-Squared: 0.088				

Polynomial Regression:

I also performed polynomial regression with a maximum degree of 2, using similar set of variables, and the result showed that similar variables as those in the linear regression were

significant, in addition, the presence of government insurance and life insurance also had p-values < 0.05, R squared slightly lower than linear regression and MSE elightly higher.

Tree Regression:

This model produces a negative R-squared value and a much higher mean squared error compared with the previous two models.

3.6 Modeling Diabetes:

Row ID	I	Nr. of features	D Accuracy	S Added feature
4	4		0.705	1.0_H13CPL
2	2		0.703	R13IWCMP
1	1		0.702	8.0_R13CENDIV
8	8		0.701	5.0_R13CENREG
3	3		0.7	1.0_H13ANYFAM
10	10		0.696	2.0_R13MSTAT
5	5		0.694	R13HESRC3
7	7		0.694	5.0_R13MSTAT
13	13		0.694	0.0_R13MNEV
11	11		0.693	1.0_R13CENDIV
6	6		0.691	3.0_RARACEM
9	9		0.691	5.0_RARELIG
12	12		0.69	0.0_R13HIGOV
15	15		0.688	5.0_R13CENDIV
18	18		0.688	R13WORK2
14	14		0.687	2.0_RARACEM
20	20		0.685	0.0_R13HOSP
16	16		0.68	4.0_RARELIG
17	17		0.679	0.0_H13NHMLIV
19	19		0.676	1.0_R13MNEV

Feature selection:

Table above showed the result of the forward feature selection process, with their respective accuracy when being added to the model. Those 20 features were: H13CPL (whether couple household), R13IWCMP (worker compensation income), R13CENDIV (census division), R13CENREG (census region), H13ANYFAM (any family respondent in household), R13MSTAT (marital status), R13HESRC3 (source of insurance plan #3), R13MNEV (never married), RARACEM (race), R13RELIG (religion), R13HIGOV (R covered by government insurance plan), R13WORK2 (R work at a 2nd job), R13HOSP (R hospital stay previous 2 years), H13NHMLIV (living in nursing home).

Random Forest:

I first performed Random Forest Classification on all the column attributes except several identifier variables, not surprisingly, the model overfits. Currently, I am going to use the set of variables selected by feature selection and perform random forest again.

R13DIAB \ ...	0.0	1.0	4.0	3.0
0.0	1128	82	0	0
1.0	409	75	0	0
4.0	0	0	0	0
3.0	0	0	0	0

The accuracy on the test set was 71.015%, which is much lower than using all variables, so the model is not overfitting, however, there are a lot of space to improve the model

Decision Tree:

R13DIAB \ ...	0.0	1.0	4.0	3.0
0.0	997	213	0	0
1.0	354	130	0	0
4.0	0	0	0	0
3.0	0	0	0	0

The accuracy for the decision tree using the same set of features was 66.529 %, which is lower than random forest, since it contains only one tree.

Logistic Regression

R13DIAB \ ...	0.0	1.0	4.0	3.0
0.0	488	6	0	0
1.0	202	5	0	0
4.0	0	0	0	0
3.0	0	0	0	0

The accuracy for the logistic regression using the same set of features was 70.328 %, which is better than the decision tree model, however, there are much fewer observations due to missing values.

Gradient Boosting

R13DIAB \ ...	0.0	1.0	4.0	3.0
0.0	1089	121	0	0
1.0	382	102	0	0
4.0	0	0	0	0
3.0	0	0	0	0

The accuracy was 70.307%, similar to the logistic regression model.

Cross Validation

Decision tree:

Error rates for decision trees are higher than those generated from random forests.

Row ID	D Error i...	I Size o...	I Error ...
fold 0	36.637	565	207
fold 1	40.426	564	228
fold 2	38.053	565	215
fold 3	37.234	564	210
fold 4	38.652	564	218
fold 5	40.177	565	227
fold 6	36.525	564	206
fold 7	39.646	565	224
fold 8	44.326	564	250
fold 9	37.411	564	211

Random Forest

Generally the error rates are around 30%, with some folds much lower than the other folds, and some folds much higher than the others

Row ID	D Error i...	I Size o...	I Error ...
fold 0	31.327	565	177
fold 1	31.383	564	177
fold 2	33.097	565	187
fold 3	32.27	564	182
fold 4	30.142	564	170
fold 5	29.027	565	164
fold 6	31.028	564	175
fold 7	29.381	565	166
fold 8	27.482	564	155
fold 9	26.418	564	149

Logistic regression (Issue with covariance matrix):

Error rates varied a lot by folds, which indicated that data behaved quite differently across folds and logistic regression might not generalize well.

Row ID	D Error i...	I Size o...	I Error ...
fold 0	55.556	234	130
fold 1	39.316	234	92
fold 2	51.502	233	120
fold 3	45.299	234	106
fold 4	34.335	233	80
fold 5	37.607	234	88
fold 6	51.709	234	121
fold 7	49.356	233	115
fold 8	45.726	234	107
fold 9	33.906	233	79

Gradient Boosting:

Gradient Boosting performed best out of the four categorical classifiers based on this specific cross validation. This indicated that although random forest had higher accuracy than boosting in the testing round, it might be simple due to the different testing data used.

Row ID	D Error i...	I Size o...	I Error ...
fold 0	30.265	565	171
fold 1	27.837	564	157
fold 2	30.088	565	170
fold 3	31.56	564	178
fold 4	30.142	564	170
fold 5	27.257	565	154
fold 6	28.723	564	162
fold 7	28.496	565	161
fold 8	31.738	564	179
fold 9	29.787	564	168

Gradient Boosting performed best out of the four categorical classifiers based on this specific cross validation. This indicated that although random forest had higher accuracy than boosting in the testing round, it might be simple due to the different testing data used.

4. Discussion

First of all, there are several issues still remaining in the current KNIME workflow, the feature selection metanode took a long time to run, and it never ended when using the “backward” option. This issue could be caused both by the huge combination of features and the large sample size from the HRS dataset. It might be interesting to explore more using this metanode in later projects. Secondly, there were still issues with Simple Tree Regression Learner and Logistic Regression Learner. For the tree regression, R-square value was always negative even after filtering out missing values and potential collinear columns. Similar issue happened with logistic regression, the covariance matrix was singular and the learner did not converge, possibly due to some unknown collinear columns.

Based on the feature selection and modeling results from both types of modeling, we can see that several social and economic metrics do have significant impact on the health outcome. Not all variables are initially included, either because of its similarity with other variables, or the variables’s measurements were embedded within a broader variable (perfect correlation). Overall, assets variables were significant in general, although only some were chosen since they were highly correlated with each other, in addition, geographical location and living condition indicators can also play a significant role.

There are many more indicators that could be used as dependent variables, which can be explored in further study, these can include: stroke, heart disease, cancer, alzheimer’s disease, etc.