Report V2

Pritom

2023-06-19

Dataset

This dataset consist a total of 71,835 observations with 18 columns. From those 18 variables we take SC_RACE_R, SC_HISPANIC_R, SC_AGE_YEARS, SC_SEX, ADHD (K2Q31A), ACE1, ACE3, ACE4, ACE5, ACE6, ACE7, ACE8, ACE9 and ACE10 variables.

Objective

The main objective of the study is to predict the ADHD as well as determine the variables those are responsible to predict the ADHD levels.

Response Variable

ADHD	n	prop
Yes	7355	0.1024
No	64480	0.8976

Here we can see that almost 89.8% of the child do not have the ADHD. So this is a highly imbalanced dataset. When we apply algorithm to this data set we can not apply the performance metric such as accuracy. Rather we will use the metric like kappa, roc_auc since those are robust on the effect of the class imbalance.

Predictor Variables

There are 9 ACE type questions which were asked to retrieve information about.

ACE1	ACE3	ACE4	ACE5	ACE6	ACE7	ACE8	ACE9	ACE10
2	2	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2
1	1	2	1	2	2	2	2	2
1	2	2	2	2	2	2	2	2

Where

- 1 is encoded as Yes.
- 2 is encoded as No.

But we want to re-encode it as

- 0 is encoded as No.
- 1 is encoded as Yes.

After doing that our new observations would be:

ACE1	ACE3	ACE4	ACE5	ACE6	ACE7	ACE8	ACE9	ACE10
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0

After doing the factor encoding and creating two new features ACE_total (calculate the total number of yes in the ACE question) and ACE_na (calculate the total number of missing in the ACE question) the dataset become,

ACE1	ACE3	ACE4	ACE5	ACE6	ACE7	ACE8	ACE9	ACE10	ACE_total	ACE_na
No	No	No	No	No	No	No	No	No	0	0
No	No	No	No	No	No	No	No	No	0	0
No	No	No	No	No	No	No	No	No	0	0
No	No	No	No	No	No	No	No	No	0	0
No	Yes	No	Yes	No	No	No	No	No	2	0
No	No	No	No	No	No	No	No	No	0	0

Here, one noticeable thing would be initially ACE1 variable had four labels which were converted to two labels.

Descriptive Statistics

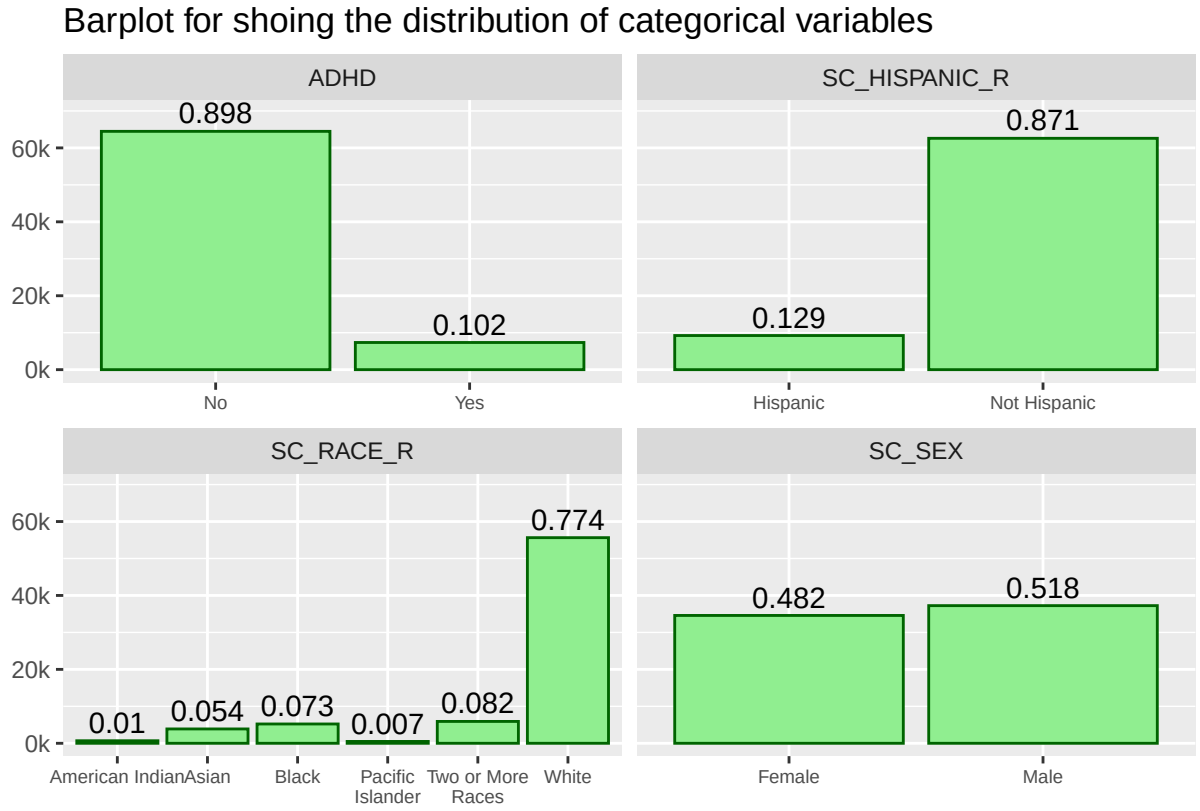
This table presents the percentages of events for each category within the predictor categorical variables. The variable SC_HISPANIC_R comprises 2 categories. The category with the highest frequency within the SC_HISPANIC_R variable is labeled as "Not Hispanic" (87.1%), while the category with the lowest frequency is labeled as "Hispanic" (12.9%). Similarly, the variable SC_RACE_R encompasses 2 categories, with the highest frequency occurring within the "White" category (77.4%), and the lowest frequency within the "Pacific Islander" category (0.7%). Likewise, the variable SC_SEX consists of 2 categories, with the highest frequency observed in the "Male" category (51.8%), and the lowest frequency in the "Female" category (48.2%).

variables	event	count	prop
SC_HISPANIC_R	Not Hispanic	62598	87.1%
SC_HISPANIC_R	Hispanic	9237	12.9%
SC_RACE_R	White	55634	77.4%
SC_RACE_R	Two or More Races	5917	8.2%
SC_RACE_R	Black	5223	7.3%
SC_RACE_R	Asian	3882	5.4%
SC_RACE_R	American Indian	683	1.0%
SC_RACE_R	Pacific Islander	496	0.7%
SC_SEX	Male	37246	51.8%
SC_SEX	Female	34589	48.2%

This table below show us the mean and standard deviation for the numeric predictor variables. The variable ACE_total has mean 0.762 and sd 1.266. For the variable SC_AGE_YEARS has mean 9.467 and sd 5.183.

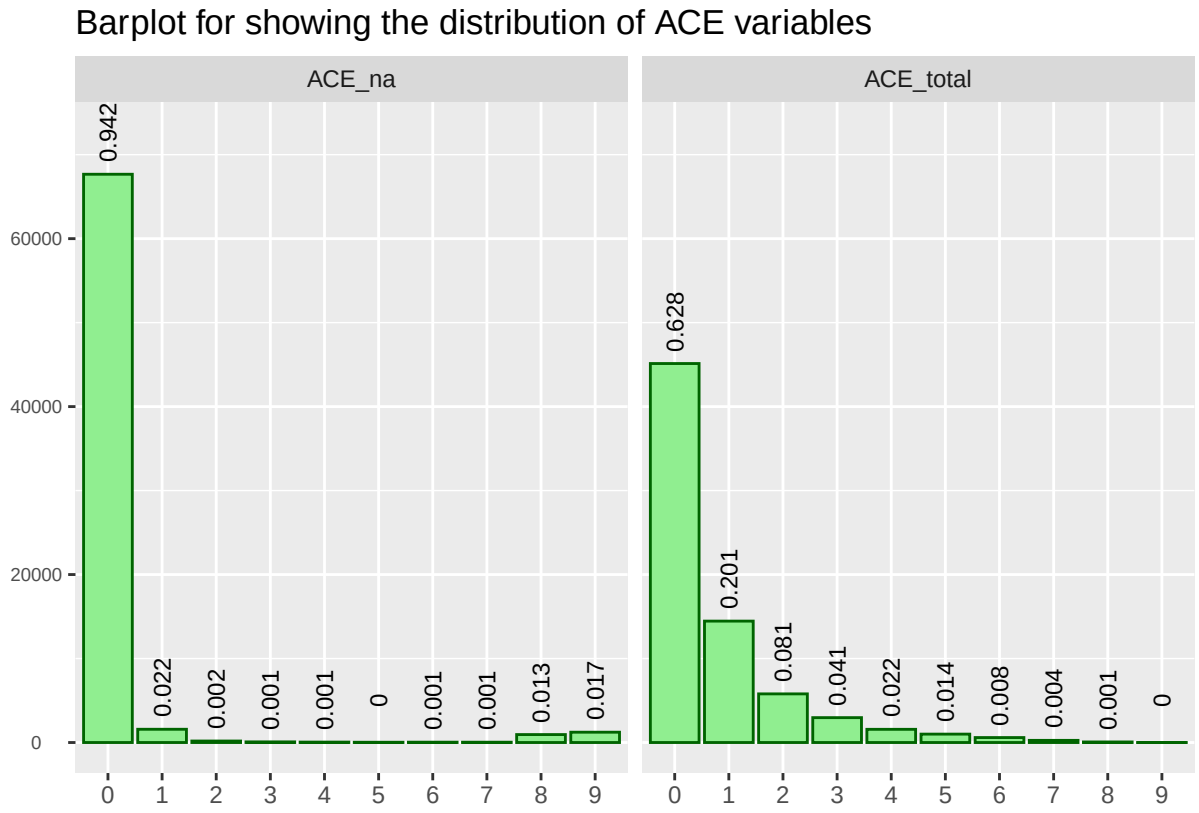
variable	mean	sd
ACE_total	0.726	1.266
SC_AGE_YEARS	9.467	5.183

This plot below is a visual representation of the table for showing the distribution of the categorical variables.



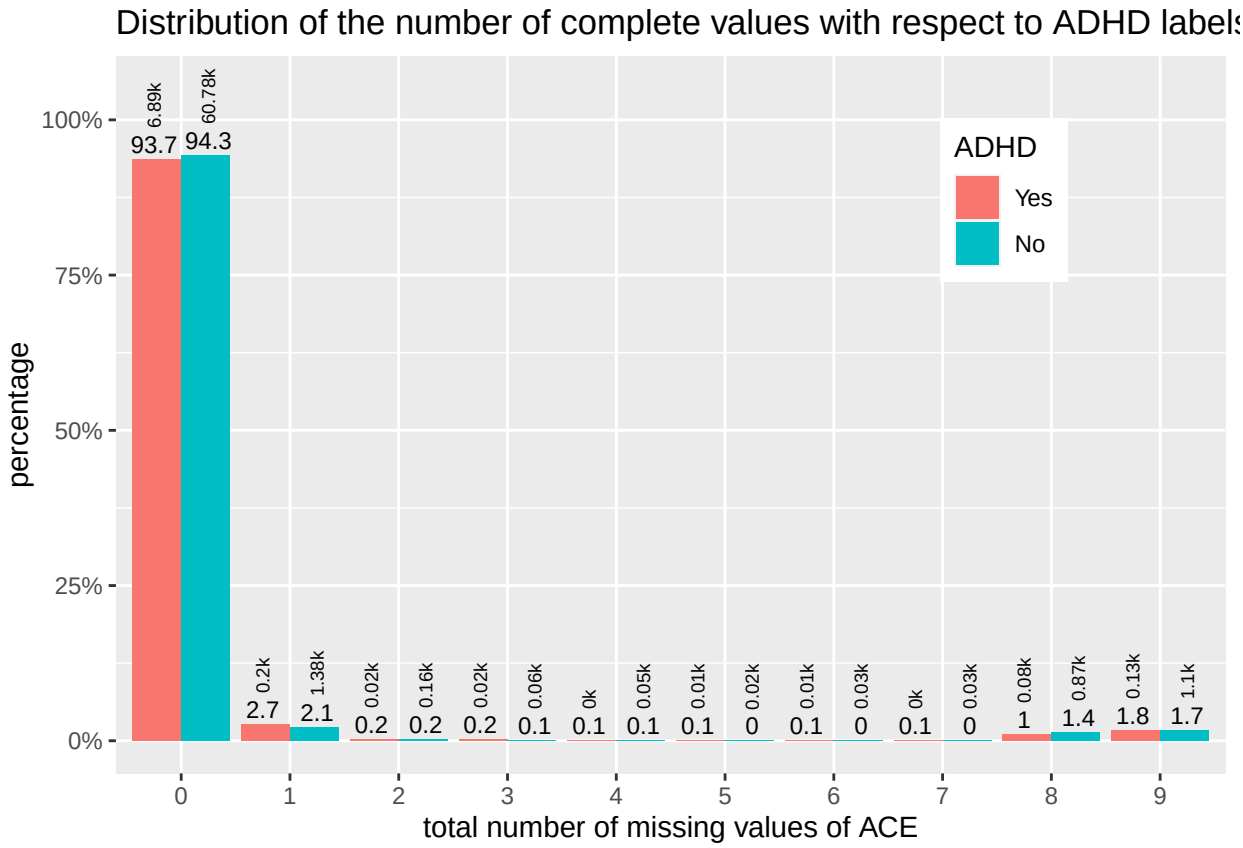
New Features

This plot is showing the distribution of the newly created features. The ACE answers for 94.2% of the observations are not missing. So it will be safe to omit the observations for which there are at least one missing ACE values. The value of total number of ACE is zero for 62.8% of the observations.



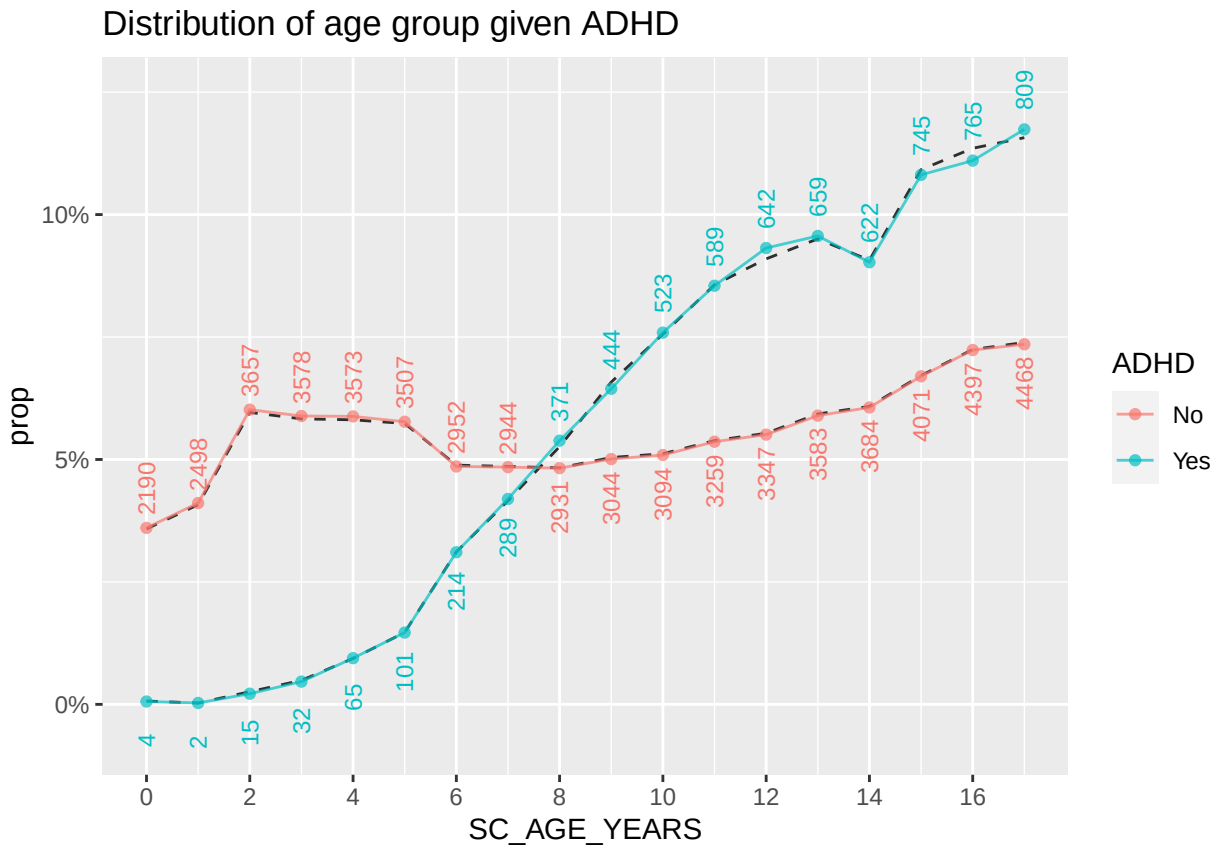
Pattern in the missing observations

From this plot below we can safely assume that the missing values are occur at random. The age distribution for the individuals having ADHA is approximately same as the The age distribution for the individuals not having ADHA. That is P(Missing number of ACE answers = i|ADHD)=P(Missing number of ACE answers = i). Since the missing value occar at random we may omit them from the rest of the analysis part.



Effect of omitting missing values

From this plot we can see that, the age distribution is approximately uniform for all the individuals do not have ADHD. On the other hand, the age distribution is skewed for all the individuals having ADHD. We may conclude that, the tendency of diagnostic as ADHD is lower at the younger age. Here the black dotted line represent the distribution of the age before the omitting the observations for which atleast one ACE's were missing.

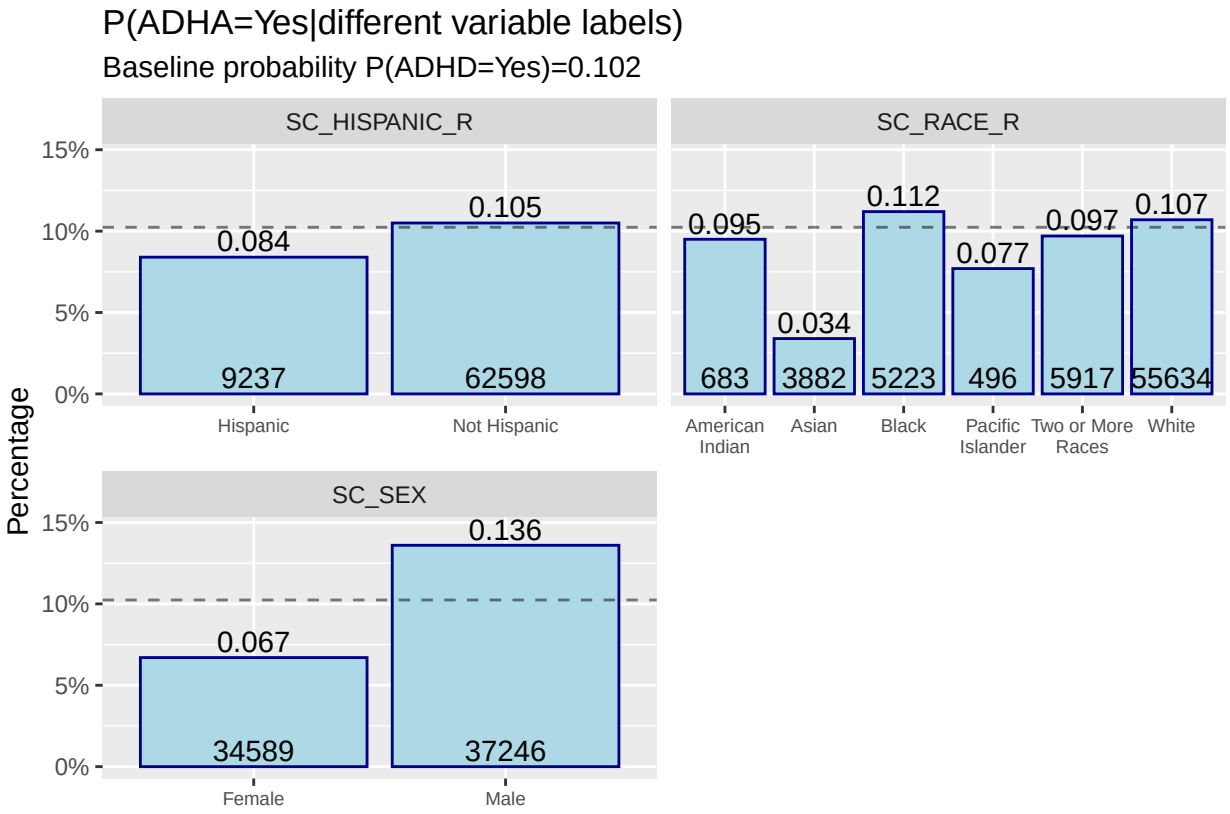


Till now we see try to figure out the distribution of the features given that a person having ADHD or not. From the now we will try to see the distribution of ADHD labels given the value of different features.

Variable Importance

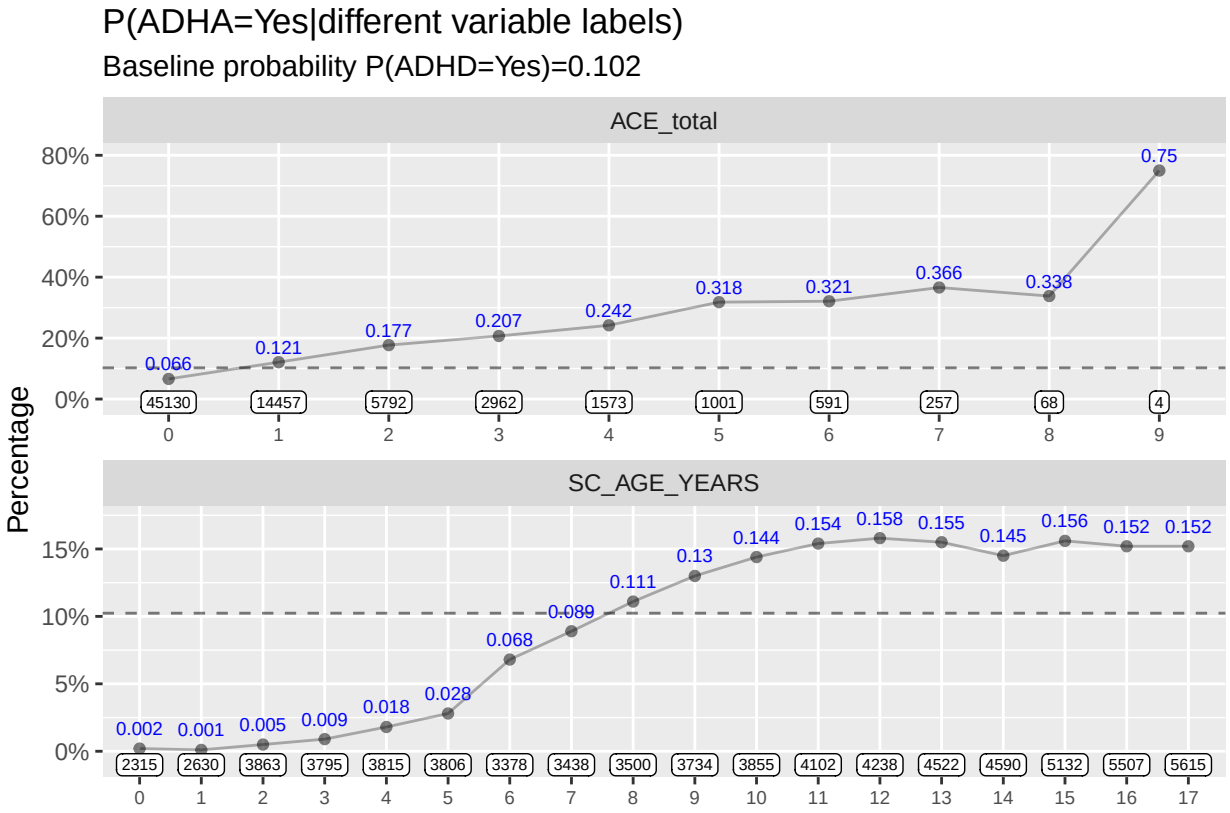
Probability of having ADHD given categorical response

Here we can see that the probability of diagnostic as having ADHD is lower given that race is Asian. Among the male individuals the tendency of having ADHD is more.



Probability of having ADHD given numeric response

From this plot we can see that as the number of ACE_total increases the P(ADHD=Yes|ACE_total) also increases. Same trend can be observed for the age variables after the age of 7.



Train Test Split

Since we are using models like decision tree and conditional inference tree which are prone to overfitting, we are keeping separate some of the data points those will no be included on the training process. We kept 80% observations for the training process and 20% observations for the testing procedure. This table below show the distribution of ADHD for the training and testing dataset. The training set contains 54133 observation and the testing set contains 13535 observations. In both the training and testing set the distribution of ADHD variable is safe to assume to be identical. We are using a seed value to split the data into training and testing so that the result become reproducable.

label	ADHD	n	prop	sample_size
test	Yes	1379	0.1019	13535
test	No	12156	0.8981	13535
train	Yes	5512	0.1018	54133
train	No	48621	0.8982	54133

Distribution of the Testing Dataset

The testing set contains 13535 observation. The table below represents the distribution of the categorical variables. The percentage are fluctuated a little due to the selection bias but more or less they are the same.

variables	event	count	prop
ADHD	Yes	1379	10.2%
ADHD	No	12156	89.8%
SC_HISPANIC_R	Hispanic	1738	12.8%
SC_HISPANIC_R	Not Hispanic	11797	87.2%
SC_RACE_R	White	10590	78.2%
SC_RACE_R	Black	931	6.9%
SC_RACE_R	American Indian	120	0.9%
SC_RACE_R	Asian	692	5.1%
SC_RACE_R	Pacific Islander	93	0.7%
SC_RACE_R	Two or More Races	1109	8.2%
SC_SEX	Male	7055	52.1%
SC_SEX	Female	6480	47.9%

This table below shows the mean and the sd for the numeric variables in the training set. The mean and sd has almost no changes.

variable	mean	sd
ACE_total	0.7246	1.26
SC_AGE_YEARS	9.422	5.226

Class Imbalance

As we can see the train dataset contains 10.18% of cases for whom ADHD is Yes and 89.82% of cases for whom ADHD is No. We can call the ADHD Yes as minority class on the other hand the ADHD No majority class. If the proportion of the minority class and majority class significantly differ the problem refers as class imbalance problem. If the class imbalance problem exist then there become a tendency for the Machine Learning algorithm to always predict the majority class. In that case there are some techniques which are useful to deal with this problem.

In this project to deal with the class imbalance we will apply SMOTE algorithm to the minority class (ADHD No). We will use over ration equals one. That is the total number of the minority class will match the total number of the majority class in the training set. Extra synthetic samples were generated from the existing minority class using SMOTE algorithm. Before applying the SMOTE-NC algorithm there were 5512 training example in the minority class. After applying the SMOTE-NC, the minority class will match the total number of observation of the majority class (48621) which will consist in total 97242 observations in the training data.

Metric

There are several metric which can be used to evaluate performance of the machine learning models. The metric are selected on the objective of the model. We will calculate the sensitivity, specificity, precision, recall, accuracy, balanced accuracy and area under the roc curve. Those metrics are defined as

- Sensitivity = A/(A+C)
- Specificity = D/(B+D)
- Precision = A/(A+B)
- Recall = A/(A+C)
- Accuracy = (A+C)/(A+B+C+D)
- Balanced Accuracy = (sensitivity+specificity)/2

Where,

	Truth	
Prediction	Yes	No
Yes	A	B
Yes	C	D

Distribution in Training

This table illustrates the distribution of categorical variables in the training example following the application of the SMOTE algorithm. It is evident that both the ADHD "Yes" and "No" classes exhibit similar percentages. The proportions of the categories within the predictor variables are generally consistent, except for the SC_SEX (Male) group. This discrepancy can be attributed to the association between SC_SEX (Male) and the ADHD "Yes" group. As synthetic samples were generated from the ADHD "Yes" group, the percentage values of all associated levels increased accordingly.

variables	event	count	prop
ADHD	Yes	48621	50.0%
ADHD	No	48621	50.0%
SC_HISPANIC_R	Not Hispanic	85952	88.4%
SC_HISPANIC_R	Hispanic	11290	11.6%
SC_RACE_R	White	77131	79.3%
SC_RACE_R	Two or More Races	8012	8.2%
SC_RACE_R	Black	6987	7.2%
SC_RACE_R	Asian	3667	3.8%
SC_RACE_R	American Indian	872	0.9%
SC_RACE_R	Pacific Islander	573	0.6%
SC_SEX	Male	57446	59.1%
SC_SEX	Female	39796	40.9%

This table below shows the mean and the sd of the numeric variables. We can see an increase in both of the numeric variables. This is due to the same effect discussed above which is a higher than the average values were associated with the minority class (ADHA Yes). Due to the increase in the minority sample (synthetic) the variabes which are associated with it also changes.

variable	mean	sd
ACE_total	1.074	1.55
SC_AGE_YEARS	10.73	4.718

Logistic regression

Summary Table with p.value

Logistic regression is one of the most classic algorithm. In logistic regression the log of odds of the ADHD Yes has been tried to model using the linear combination of the features. The categorical variables were decomposed using one hot encoding. Though the logistic regression is not very well for the prediction capabilities the model can provide a great interpretation for the predictor variables. The table below is showing the output of the logistic regression. We can see that all of the variables are significant to the model at 1% level of significant since the p-value for the all of the variables are less that 0.01. The odds of having ADHD(Yes) will increase on average by $e^{0.290478} = 1.337066$ times for an "Not Hispanic" than the Hispanic individuals. Similarly for the individuals who belongs to the race Black, American Indian, Asian, Pacific Islander and Two or More Races the odds of having ADHD will decrease by 0.946, 0.681, 0.335, 0.710 and 0.913 times compare to the individuals belong to race white. The odds of having ADHD for female is 0.437 times lower than the boys group. For 1 unit increase in the age or ACE_Total variable the odds of having ADHD increase by 1.159 and 1.414 times respectively.

```
Call:
glm(formula = ADHD ~ ., family = "binomial", data = mutate(df_train,
    ADHD = relevel(ADHD, "No")))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.74733  -0.97463  -0.00326   0.99373   2.30009

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -1.820646   0.028891 -63.018  < 2e-16 ***
SC_RACE_RBlack             -0.055093   0.027921  -1.973 0.048477 *
SC_RACE_RAmerican Indian   -0.384893   0.076411  -5.037 4.72e-07 ***
SC_RACE_RAsian             -1.092313   0.042325 -25.808  < 2e-16 ***
SC_RACE_RPacific Islander  -0.342778   0.096052  -3.569 0.000359 ***
SC_RACE_RTwo or More Races -0.091218   0.026579  -3.432 0.000599 ***
SC_HISPANIC_RNot Hispanic   0.290478   0.022720  12.785  < 2e-16 ***
SC_AGE_YEARS                0.147496   0.001658  88.958  < 2e-16 ***
SC_SEXFemale               -0.827660   0.014699 -56.306  < 2e-16 ***
ACE_total                   0.346678   0.005367  64.595  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 134806  on 97241  degrees of freedom
Residual deviance: 113744  on 97232  degrees of freedom
AIC: 113764

Number of Fisher Scoring iterations: 4
```

Confidence interval

Table for showing the 95% CI for the intercept of the logistic regression. As we previously seen that the p.value for all of the variables are significant, the 95% CI does not contains the value zero for any of the CI.

variables	intercept	2.5 %	97.5 %
(Intercept)	-1.821	-1.877	-1.764
SC_RACE_RBlack	-0.05509	-0.1098	-0.0003686
SC_RACE_RAmerican Indian	-0.3849	-0.5347	-0.2351
SC_RACE_RAsian	-1.092	-1.175	-1.009
SC_RACE_RPacific Islander	-0.3428	-0.531	-0.1545
SC_RACE_RTwo or More Races	-0.09122	-0.1433	-0.03912
SC_HISPANIC_RNot Hispanic	0.2905	0.2459	0.335
SC_AGE_YEARS	0.1475	0.1442	0.1507
SC_SEXFemale	-0.8277	-0.8565	-0.7988
ACE_total	0.3467	0.3362	0.3572

Confusion Matrix

This matrix show the table for the true and the predicted value. Among the observations 984 and 8104 observations were classified as Yes and No correctly.

```
           Truth
Prediction  Yes   No
       Yes  984 4052
       No   395 8104
```

Performance Metric

The table below show us the performance metrics which are calculated from the confusion matrix.

.metric	.estimator	.estimate
accuracy	binary	0.6714
precision	binary	0.1954
recall	binary	0.7136
specificity	binary	0.6667
sensitivity	binary	0.7136
roc_auc	binary	0.7624

Decision Tree

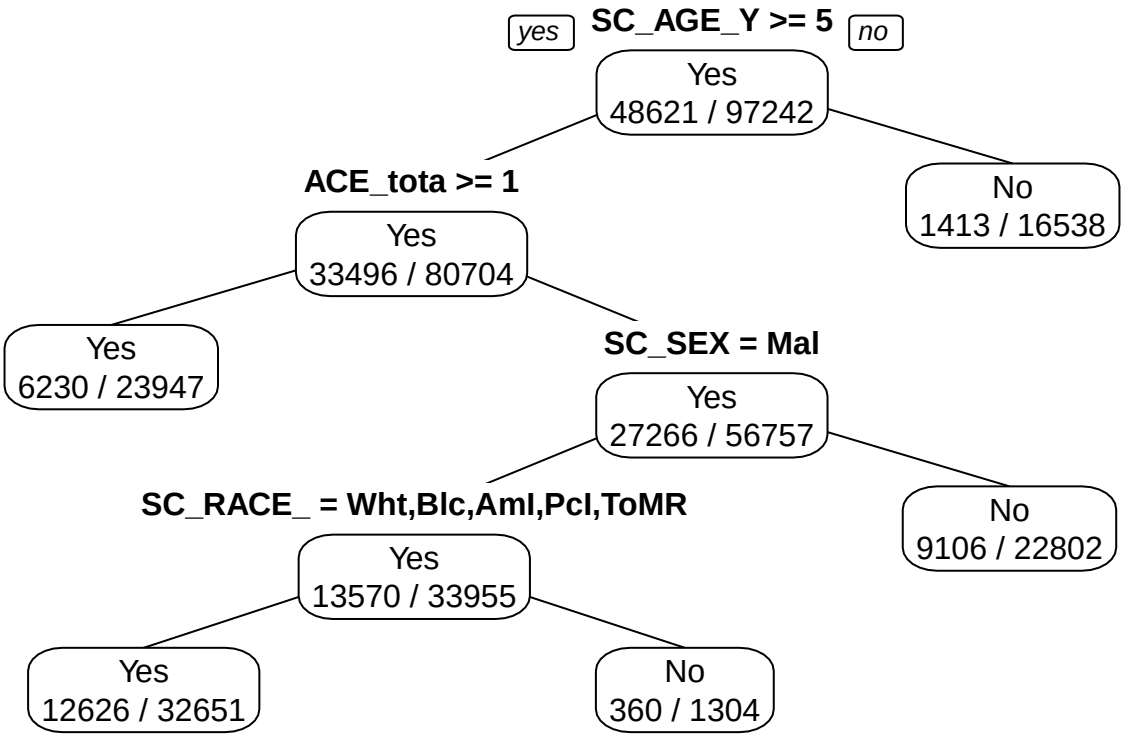
Decision tree is another algorithm which provide a higher prediction performance if it correctly trained. There are many parameters related to decision tree model which can ton be optimized from the data, are refers to the hyperparameters. Among all the hyperparameters we are going to talk about:

- minsplit : The minimum number of observations that must exist in a node in order for a split to be attempted.
- CP (cost complexity) : The main role of this parameter is to save computing time by pruning off splits that are obviously not worthwhile.
- maxdepth: Set the maximum depth of any node of the final tree.

In this project we are using the default value of the algorithm set by rpart. The value of minsplit, cp and maxdepth are 20, 0.01 and 30.

Figure for the Decision Tree

This figure show us that the age, ACE total, sex and race are the most important variable respectively.



Confusion Matrix

This matrix show the table for the true and the predicted value. Among the observations 1085 and 7538 observations were classified as Yes and No correctly.

```
            Truth
Prediction  Yes   No
       Yes 1085 4618
       No   294 7538
```

Performance Metric

The table below show us the performance metrics which are calculated from the confusion matrix. Since we used the default value of hyper-parameter setting it was quite obvious that the performance of the model wont be that perfect. However, with the random setting of the hyper-parameter provide descent performance of the model compare to the logistic regression model. The roc_auc is close to logistic regression model.

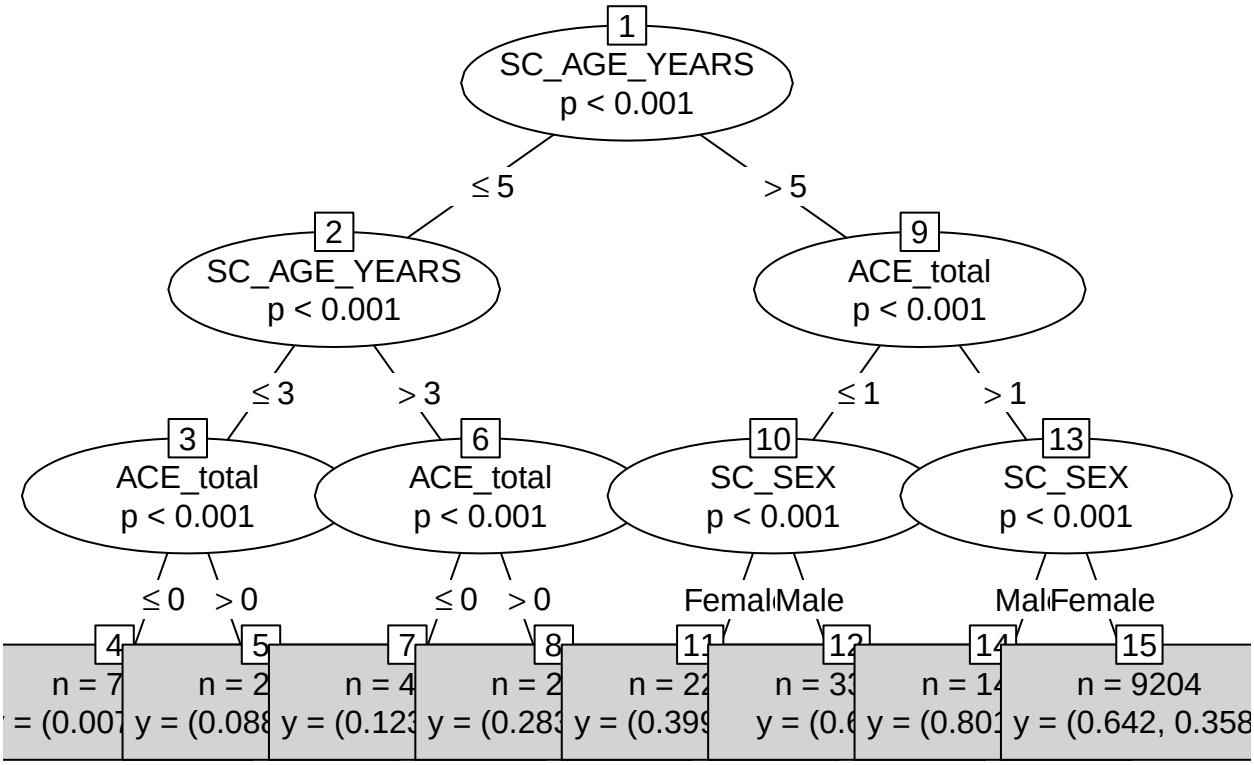
.metric	.estimator	.estimate
accuracy	binary	0.6371
precision	binary	0.1903
recall	binary	0.7868
specificity	binary	0.6201
sensitivity	binary	0.7868
roc_auc	binary	0.7444

Conditional Inferance Tree

Unlike the decision tree where the gini impurity measure is usually use to make the spiting decision, conditional inference trees employ statistical tests such as the chi-squared test or permutation tests to evaluate the quality of splits. The primary advantage of using conditional inference trees is that they provide a principled and statistically rigorous framework for building decision trees. By incorporating statistical tests, these trees can handle both categorical and continuous predictor variables and account for potential interactions among variables. Additionally, they are less prone to overfitting compared to traditional decision trees. There are also several hyperparameters associated with the model. Since the conditional inference tree is not prone to the overfitting compare to the decision tree model we may use the default parameters.

Tree Visualization

The depth of the tree is very large the whole tree is not possible to plot. We are only plotting the depth of 3.



Confusion Matrix

This matrix show the table for the true and the predicted value. Among the observations 1016 and 8208 observations were classified as Yes and No correctly.

```
            Truth
Prediction  Yes   No
       Yes 1016 3948
       No   363 8208
```

Performance Metrix

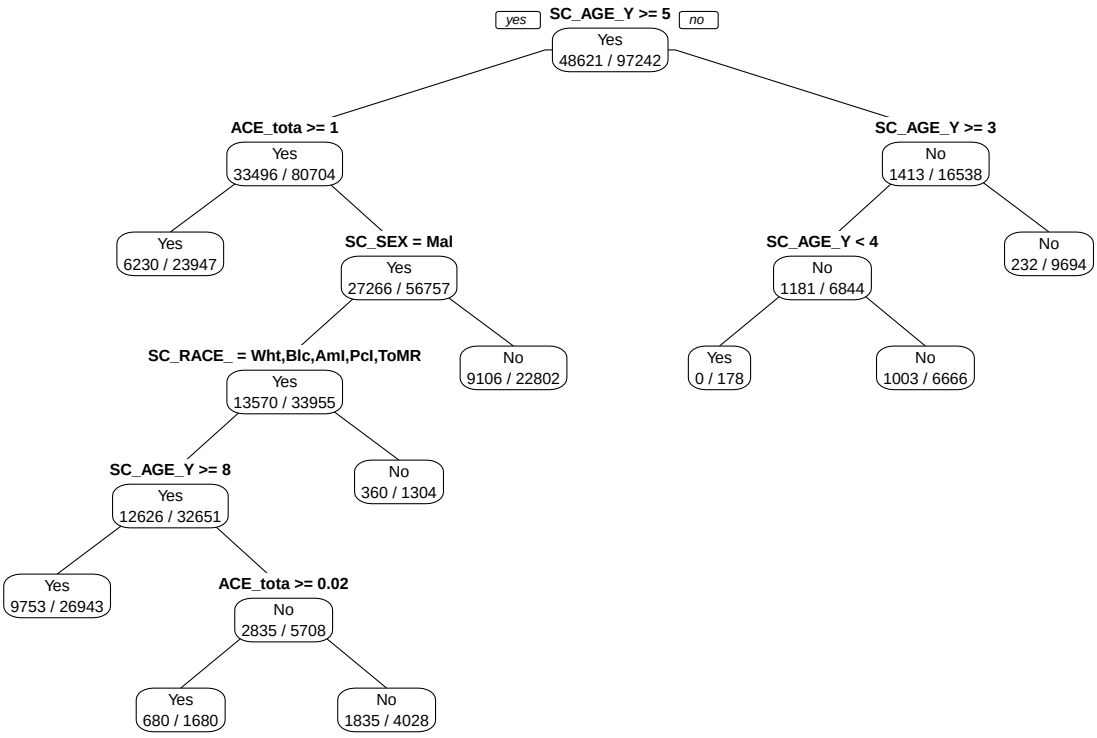
The table below show us the performance metrics which are calculated from the confusion matrix. Since we used the default value of hyper-parameter setting it was quite obvious that the performance of the model wont be that perfect. However, with the random setting of the hyper-parameter provide descent performance of the model compare to the logistic regression model. The roc_auc is close to logistic regression model.

.metric	.estimator	.estimate
accuracy	binary	0.6815
precision	binary	0.2047
recall	binary	0.7368
specificity	binary	0.6752
sensitivity	binary	0.7368
roc_auc	binary	0.7748

Decision Tree

CP = 0.001

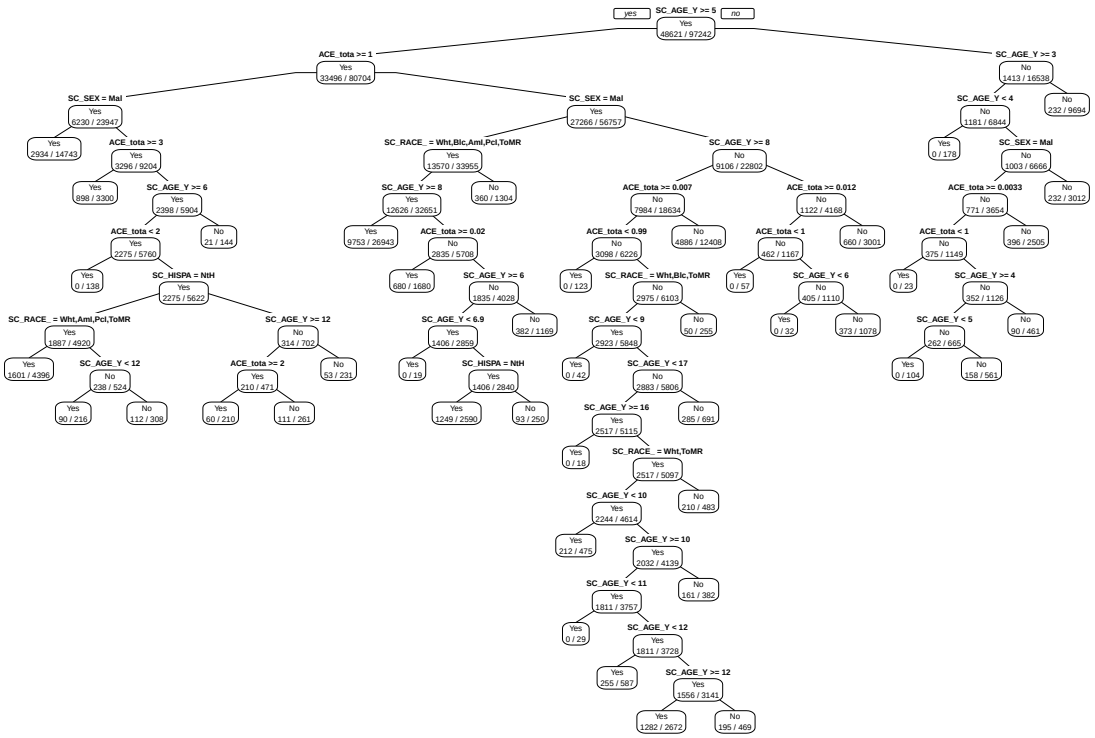
There are many related hyper-parameter among them we are only working with cost complexity parameter. Which are directly related to the pruning.



evaluation for the tree

.metric	.estimator	.estimate
accuracy	binary	0.6711
precision	binary	0.2004
recall	binary	0.7447
specificity	binary	0.6628
sensitivity	binary	0.7447
roc_auc	binary	0.751

CP = 0.0005.

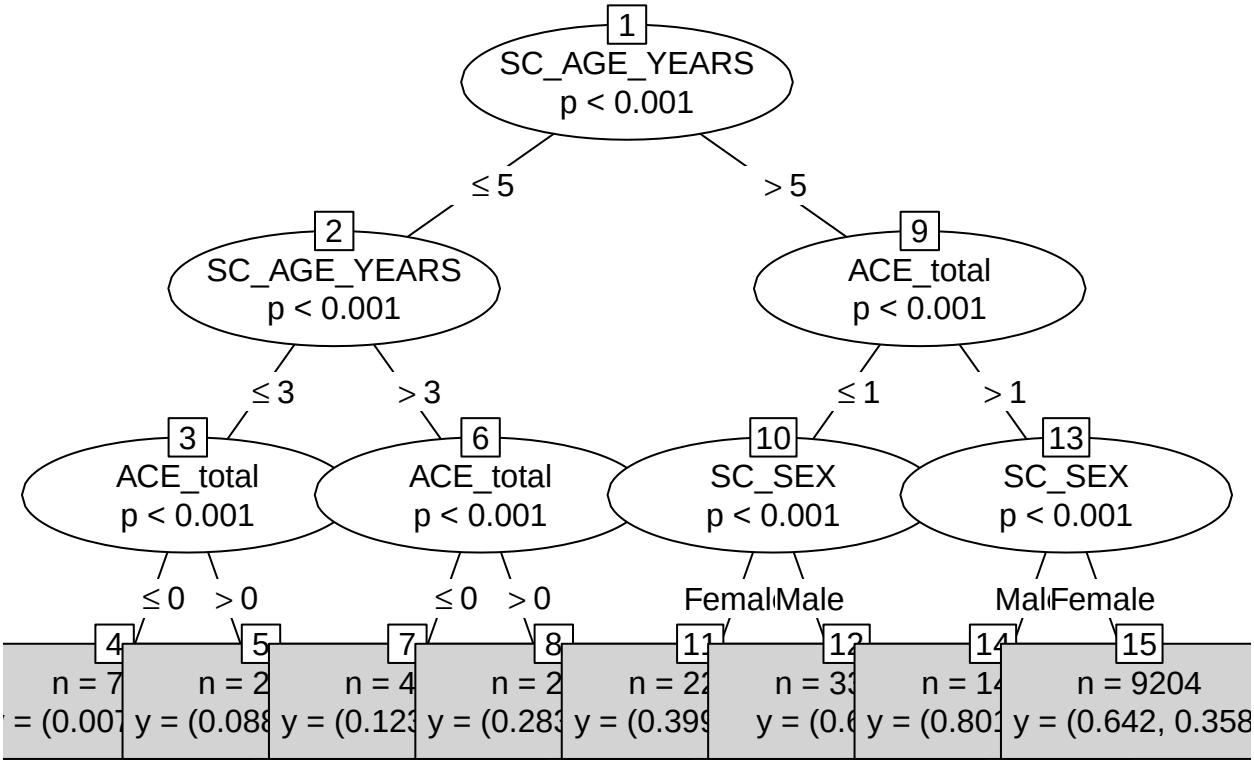


evaluation for the tree

.metric	.estimator	.estimate
accuracy	binary	0.6362
precision	binary	0.1921
recall	binary	0.802
specificity	binary	0.6174
sensitivity	binary	0.802
roc_auc	binary	0.7657

Conditioanl Inferance Tree

Max Depth = 5

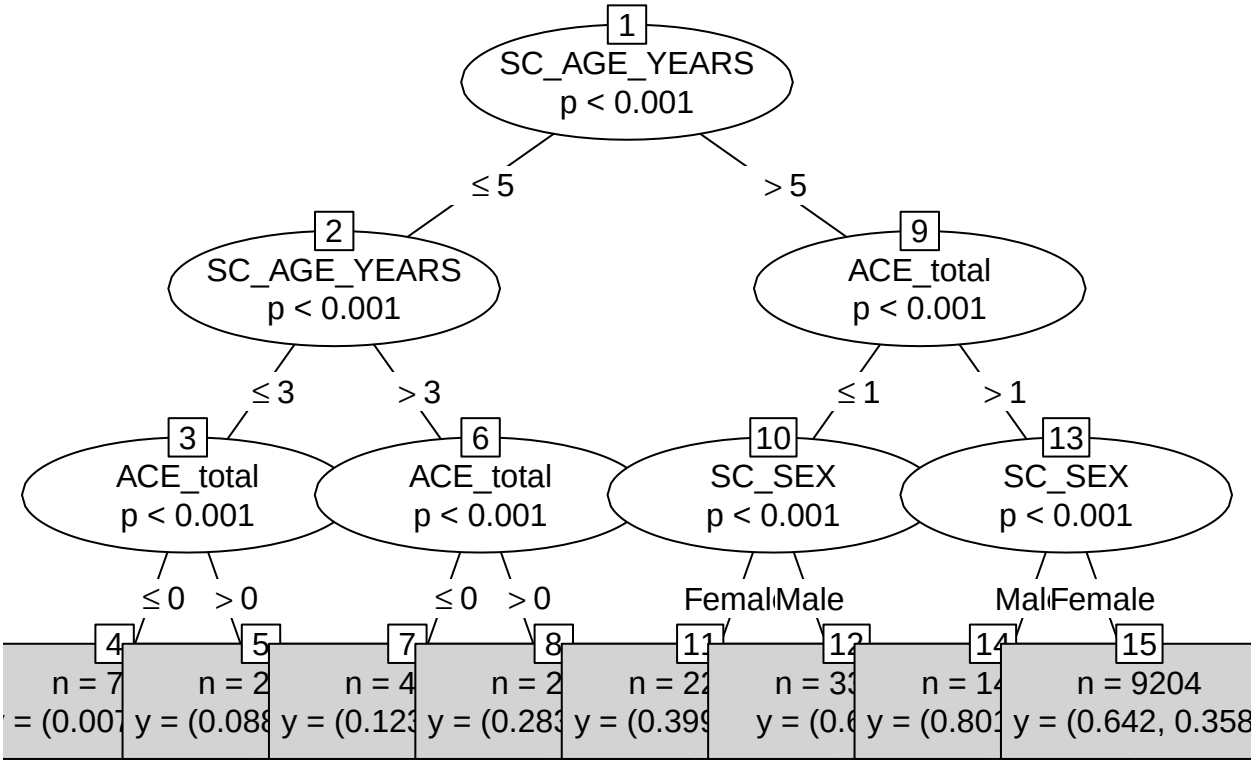


evaluation for the conditional tree

.metric	.estimator	.estimate
accuracy	binary	0.6435
precision	binary	0.1919
recall	binary	0.7781
specificity	binary	0.6282
sensitivity	binary	0.7781
roc_auc	binary	0.7679

Conditioanl Inferance Tree

Max Depth = 7



evaluation for the conditional tree

.metric	.estimator	.estimate
accuracy	binary	0.6755
precision	binary	0.203
recall	binary	0.7469
specificity	binary	0.6674
sensitivity	binary	0.7469
roc_auc	binary	0.7743