Task 1

1. Introduction

We aim to find the classification of our data, we propose Logistic Regression on our dataset, since we are dealing with binomial classification of variables. But our data has multinomial attributes for one binomial categorical distribution. We take approach of both glm and multinom fit of data to train and test dataset formed from splitting the data. As your task requires proving of our testing both R programming and SAS Enterprise Miner, we might need strategic approach and objective orientation results presentation which will represent comparative results from both R programming and SAS EM.

In our classification we might have to deal with problem in assigning labels to an input variable which are used to classify variables into one of the two classes. As we know there are several other classification methods to categorize or classify out data into categorical values. We then perform clustering to predict our target class from classification. Grouping similar things with most fit condition. All the similar group items should be in no two different grouped items shouldn't be similar.

We have the Classifier algorithm that can map our inputted data into a specific category. This model will conclude what input values will be picked only for the training which will further predict and give us the class labels/categories for our new data. In Binary Classification there will be only two outcomes possible. But classification of multi-class with at least two classes or more, every sample assigned to only one target label. And in Multi-label classification, every sample will be mapped to a set of target labels.

After Classification we take Multinomial logistic regression approach since our data have single categorical variable and multinomial variables in our dataset. We can binomially approach to our data but could be a tidier job in doing so. We will stick to Multinomial Logistic Regression for reason that are discussed in research part of this task.

2. Background Research

As an extension of binomial logistic regression, we use Multinomial regression, and this allows us in predicting dependent categorical variable in more than two levels of classification. The multinomial regression outcome helps us in predicting outcome with more than one or more variable that are independent. Here, the independent variables can be of a nominal, ordinal or continuous type. Starkweather, J. and Moske [1].

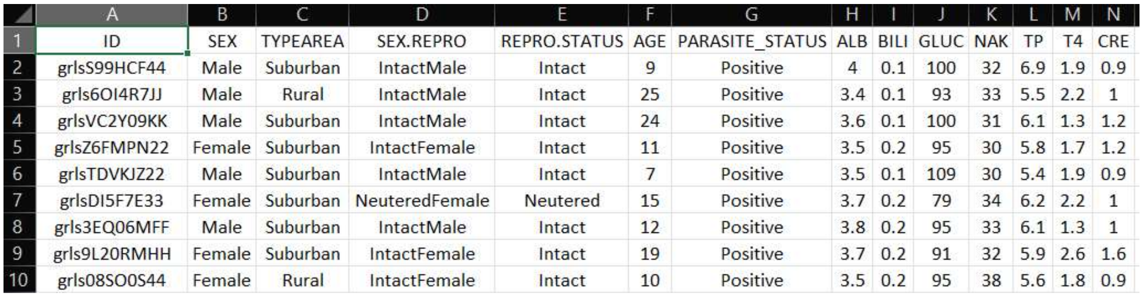
We have choice of multiple logistic regression or multinomial logistic regression to predict the positive and negative results of parasite status in a human test data, we will use multinomial logistic regression to predict the status of parasite infections. Hedeker, D., 2003 [2]. Before to use multinomial regression using R and SAS EM we will have to check few things to confirm our final output is correct. Firstly, our dependent variable should be of Nominal type which doesn't multinomial regression cannot be used in case of an ordinal variable values. Anyway, we run ordinal logistic regression to achieve Multinomial regression. We get dummy variables by converting our categorical independent variables. No multicollinearity should exist in our data. Important of them all a linear relationship should exist between dependent variable and the independent continuous variables. We can't get this just between continuous variables and nominal, so we are doing this with logit transformation of our dependent variable. Böhning, D., 1992 [3]. In the next step before performing any regression we take out outliers and highly influential points in our data.

Now the data is ready for regression test, we perform multinomial regression test on our data by slicing our data into train sample and test sample. We start testing train data for determining our model fit to our data and then we test our model of fit on test data to compare the results and see results accuracy percentage and conclude the model fit to be accepted or rejected Fagerland, M.W. and Hosmer, D.W., 2012 [4].

We further perform z test to get more accuracy of the model of fit, if this does not conclude our test and we would perform z two-tailed test to prove our model is fit to data. Bayaga, A., 2010. Multinomial Logistic Regression: Usage and Application in Risk Analysis. Journal of applied quantitative methods [5]. Since we cannot perform z test in SAS EM, we will be doing z test only in R programming.

3. Exploration of Data Set

We picked our dataset from GRLS Parasite Study [6] which is perfect data for our research study case. At first glance into the data have information of parasite status of several people test with volumes and values of attributes like id, sex, typearea, sex_repro, repro_status, age, parasite_status, alb, bili, gluc, nak, tp, t4, cre as show in table [1]



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ID	SEX	TYPEAREA	SEX.REPRO	REPRO.STATUS	AGE	PARASITE_STATUS	ALB	BILI	GLUC	NAK	TP	T4	CRE
2	grlsS99HCF44	Male	Suburban	IntactMale	Intact	9	Positive	4	0.1	100	32	6.9	1.9	0.9
3	grls6OI4R7JJ	Male	Rural	IntactMale	Intact	25	Positive	3.4	0.1	93	33	5.5	2.2	1
4	grlsVC2Y09KK	Male	Suburban	IntactMale	Intact	24	Positive	3.6	0.1	100	31	6.1	1.3	1.2
5	grlsZ6FMPN22	Female	Suburban	IntactFemale	Intact	11	Positive	3.5	0.2	95	30	5.8	1.7	1.2
6	grlsTDVKJZ22	Male	Suburban	IntactMale	Intact	7	Positive	3.5	0.1	109	30	5.4	1.9	0.9
7	grlsDI5F7E33	Female	Suburban	NeuteredFemale	Neutered	15	Positive	3.7	0.2	79	34	6.2	2.2	1
8	grls3EQ06MFF	Male	Suburban	IntactMale	Intact	12	Positive	3.8	0.2	95	33	6.1	1.3	1
9	grls9L20RMHH	Female	Suburban	IntactFemale	Intact	19	Positive	3.7	0.2	91	32	5.9	2.6	1.6
10	grls08SO0S44	Female	Rural	IntactFemale	Intact	10	Positive	3.5	0.2	95	38	5.6	1.8	0.9

Table [1]

As our task is finding a binomial variable dependency with independent variables, we leave of some variables behind (not accounted to testing our data) that are not in any shape or form to fit our data requirement.

Once our data requirement is met, we check for any missing values in our data to make sure we are working on full data, otherwise we would see some huge margin or errors in output. We are either replacing the null fields with zero or removing the variable from our testing.

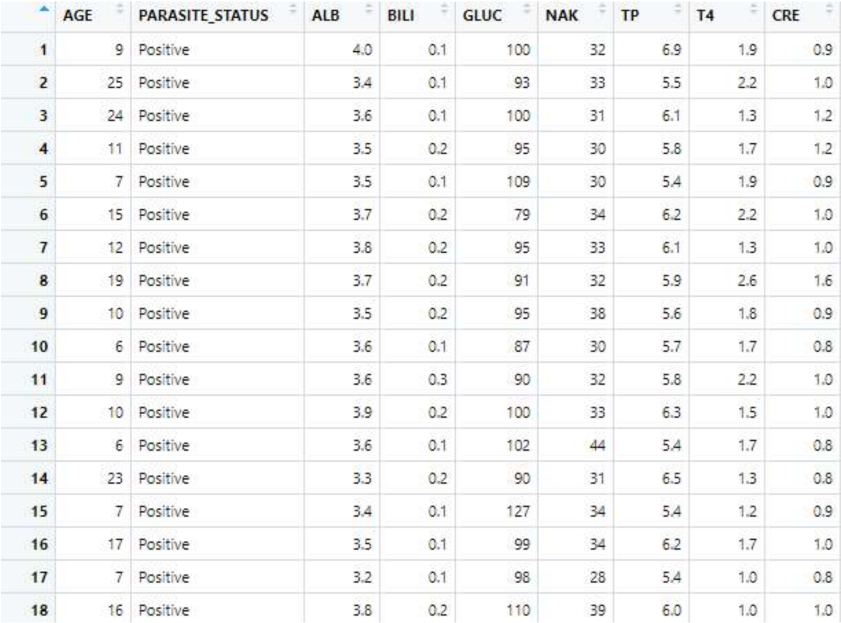
We then explore the data for checks of preparation like normalization of data requirement or sampling data in case of huge dataset. Fortunately, our data seems to well distributed and not a huge dataset as well. So, we can just go ahead and perform our regression tests on our dataset.

4. Multinomial Logistic Regression Implementation in R and SAS EM

Firstly, we perform regression test using R programming and then on to SAS EM. We will then interpret the results of both regression methods in deep, following results comparison of R program output and SAM EM outputs respectively.

A. Regression testing in R

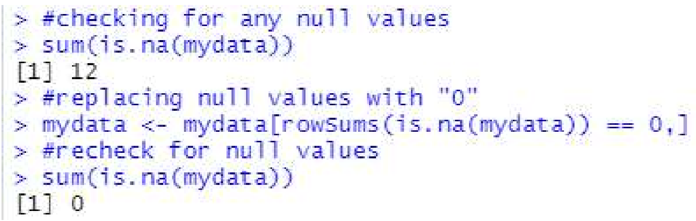
Starting with R programming we import data to R studio, and we set our working directory we explore the data initially with head and View commands to see everything going on track for further testing. We will have to subset our dataset to one with only variable in use for testing to make our testing ease to work (Table [2]).



	AGE	PARASITE_STATUS	ALB	BILI	GLUC	NAK	TP	T4	CRE
1	9	Positive	4.0	0.1	100	32	6.9	1.9	0.9
2	25	Positive	3.4	0.1	93	33	5.5	2.2	1.0
3	24	Positive	3.6	0.1	100	31	6.1	1.3	1.2
4	11	Positive	3.5	0.2	95	30	5.8	1.7	1.2
5	7	Positive	3.5	0.1	109	30	5.4	1.9	0.9
6	15	Positive	3.7	0.2	79	34	6.2	2.2	1.0
7	12	Positive	3.8	0.2	95	33	6.1	1.3	1.0
8	19	Positive	3.7	0.2	91	32	5.9	2.6	1.6
9	10	Positive	3.5	0.2	95	38	5.6	1.8	0.9
10	6	Positive	3.6	0.1	87	30	5.7	1.7	0.8
11	9	Positive	3.6	0.3	90	32	5.8	2.2	1.0
12	10	Positive	3.9	0.2	100	33	6.3	1.5	1.0
13	6	Positive	3.6	0.1	102	44	5.4	1.7	0.8
14	23	Positive	3.3	0.2	90	31	6.5	1.3	0.8
15	7	Positive	3.4	0.1	127	34	5.4	1.2	0.9
16	17	Positive	3.5	0.1	99	34	6.2	1.7	1.0
17	7	Positive	3.2	0.1	98	28	5.4	1.0	0.8
18	16	Positive	3.8	0.2	110	39	6.0	1.0	1.0

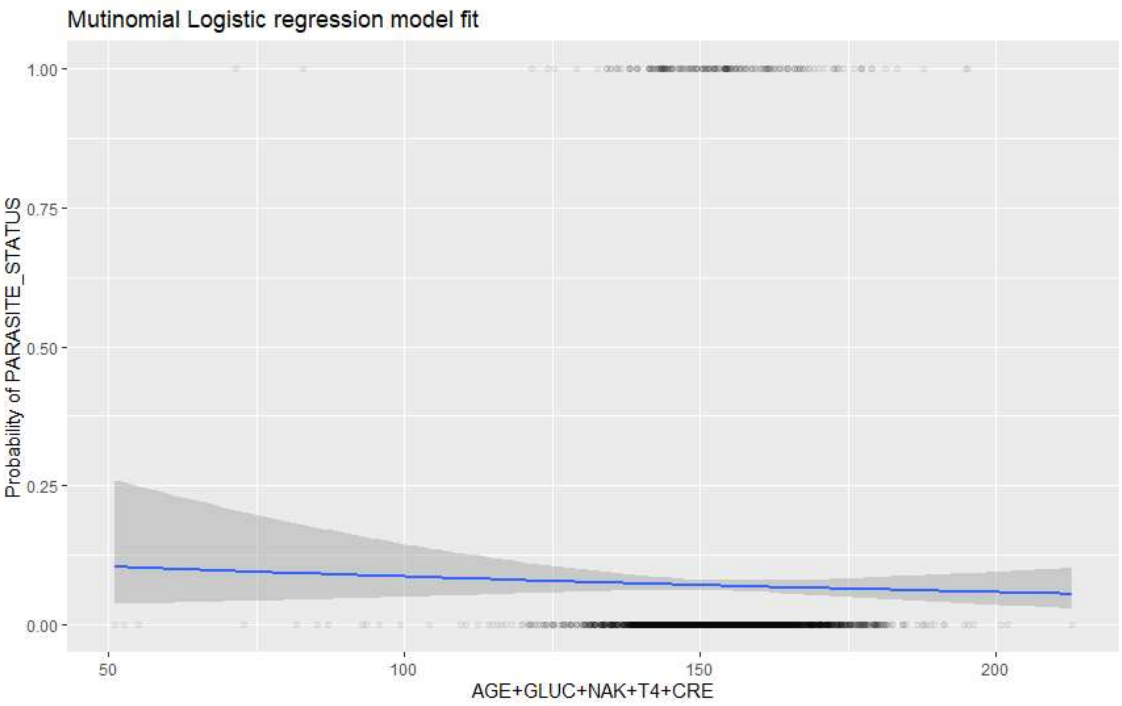
Table [2]

We perform a null value test to check for empty rows in our data. We found there are 12 missing data values, and we replace them with zero in our next step using r programming commands.



```
> #checking for any null values
> sum(is.na(mydata))
[1] 12
> #replacing null values with "0"
> mydata <- mydata[rowSums(is.na(mydata)) == 0,]
> #recheck for null values
> sum(is.na(mydata))
[1] 0
```

Our final data has one categorical dependent variable and multiples independent variables which can tested either by multiple logistic regression test or multinomial logistic regression test. Given that our dependant variable is binomial, and we are choosing multinomial regression in favour of logistic regression. We analyse our initial regression test by plotting all independent variable with one categorical dependent variable Output [1].



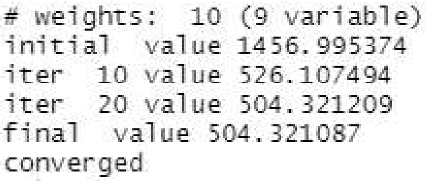
Output 1

We could see no logistic relations are formed and this gives us no choice but to use multinomial testing. Before to performing multinom method of regression testing we will split or data in train and test model by 70 and 30 percentage for measuring accuracy of the model El-Habil, A.M., 2012. An application on multinomial logistic regression model [7].

After successful creation of train and test data, we will require (nnet) package for multinomial regression testing of one categorical dependant variable

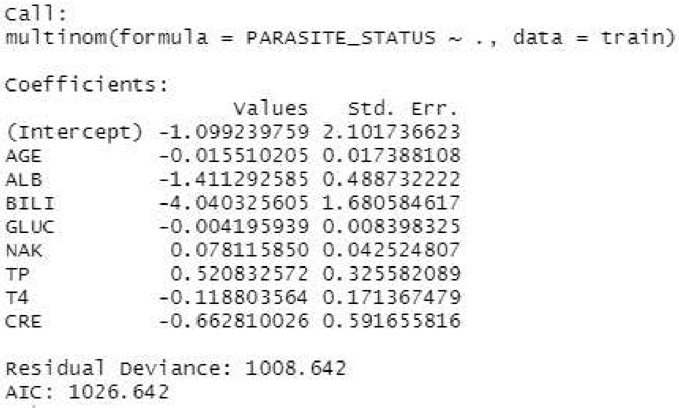
(PARASITE_STATUS) with all other independent variables (ALB, BILI, GLUC, NAK, TP, T4, CRE).

We jump onto performing regression test for multinomial model on our data. We get predicted status of PARASITE_STATUS as positive and negative results which is binomial variable. We use our train data in multinomial to get the model fit and will test in on test data for later part. Output of train data is as show in Output $^{2.}$



```
# weights:  10 (9 variable)
initial  value 1456.995374
iter  10 value 526.107494
iter  20 value 504.321209
final  value 504.321087
converged
```

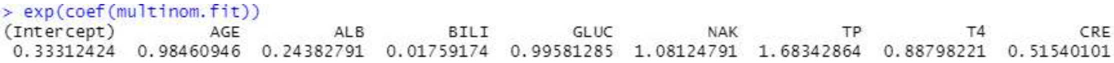
Output ^{2}



```
Call:
multinom(formula = PARASITE_STATUS ~ ., data = train)

Coefficients:
                 Values   Std. Err.
(Intercept) -1.099239759 2.101736623
AGE         -0.015510205 0.017388108
ALB         -1.411292585 0.488732222
BILI        -4.040325605 1.680584617
GLUC        -0.004195939 0.008398325
NAK          0.078115850 0.042524807
TP           0.520832572 0.325582089
T4          -0.118803564 0.171367479
CRE         -0.662810026 0.591655816

Residual Deviance: 1008.642
AIC: 1026.642
```

Output ^{3}

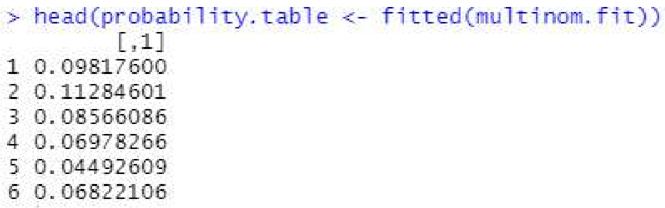
Output ^{3} shows the summary of model fit to out data, the higher AIC value we get the greater the model fit is. We also get the intercept value and individual coefficient values our independent variables with standard error rate. For extracting exponential values of our model, we use R programming command and get output ^{4}.



```
> exp(coef(multinom.fit))
(Intercept)        AGE        ALB       BILI       GLUC        NAK         TP         T4        CRE
 0.33312424 0.98460946 0.24382791 0.01759174 0.99581285 1.08124791 1.68342864 0.88798221 0.51540101
```

Output ^{4}

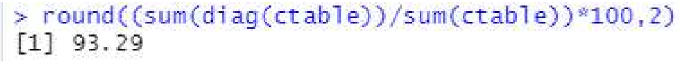
We also get probability of model fit with our data in Output ^{5}



```
> head(probability.table <- fitted(multinom.fit))
        [,1]
1 0.09817600
2 0.11284601
3 0.08566086
4 0.06978266
5 0.04492609
6 0.06822106
```

Output ^{5}

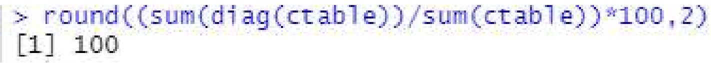
We then formulate all the probability values of predicted and categorical variable of train data set into a table, so that, we can finally accuracy of our model by Calculating accuracy minus sum of diagonal elements divided by total observations.



```
> round((sum(diag(ctable))/sum(ctable))*100,2)
[1] 93.29
```

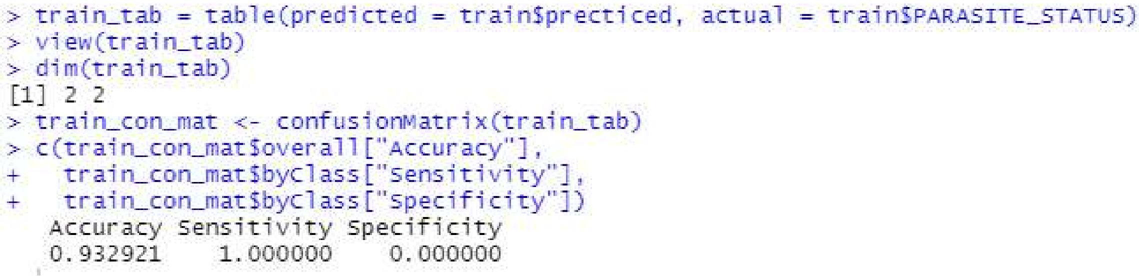
We get 93.29 as our prediction accuracy of train data set.

We do the same for test data and get accuracy of 100 percent, which show test data have 7.71 increase in accuracy of model.



```
> round((sum(diag(ctable))/sum(ctable))*100,2)
[1] 100
```

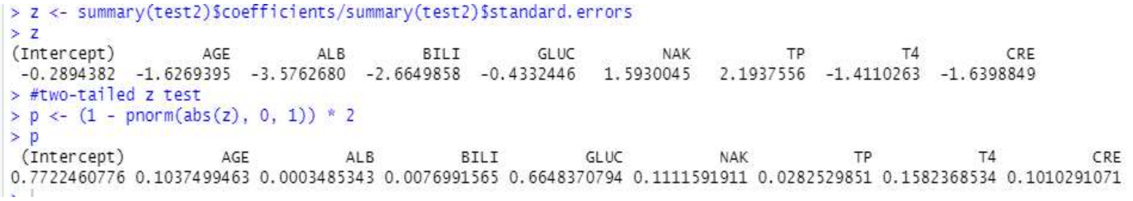
For more specification of our results, we create a matrix table to achieve confusion matrix representation of Accuracy, Sensitivity and Specificity as show in output [6]



```
> train_tab = table(predicted = train$precticed, actual = train$PARASITE_STATUS)
> view(train_tab)
> dim(train_tab)
[1] 2 2
> train_con_mat <- confusionMatrix(train_tab)
> c(train_con_mat$overall["Accuracy"],
+   train_con_mat$byClass["Sensitivity"],
+   train_con_mat$byClass["Specificity"])
   Accuracy Sensitivity Specificity 
   0.932921    1.000000    0.000000 
```

Output [6]

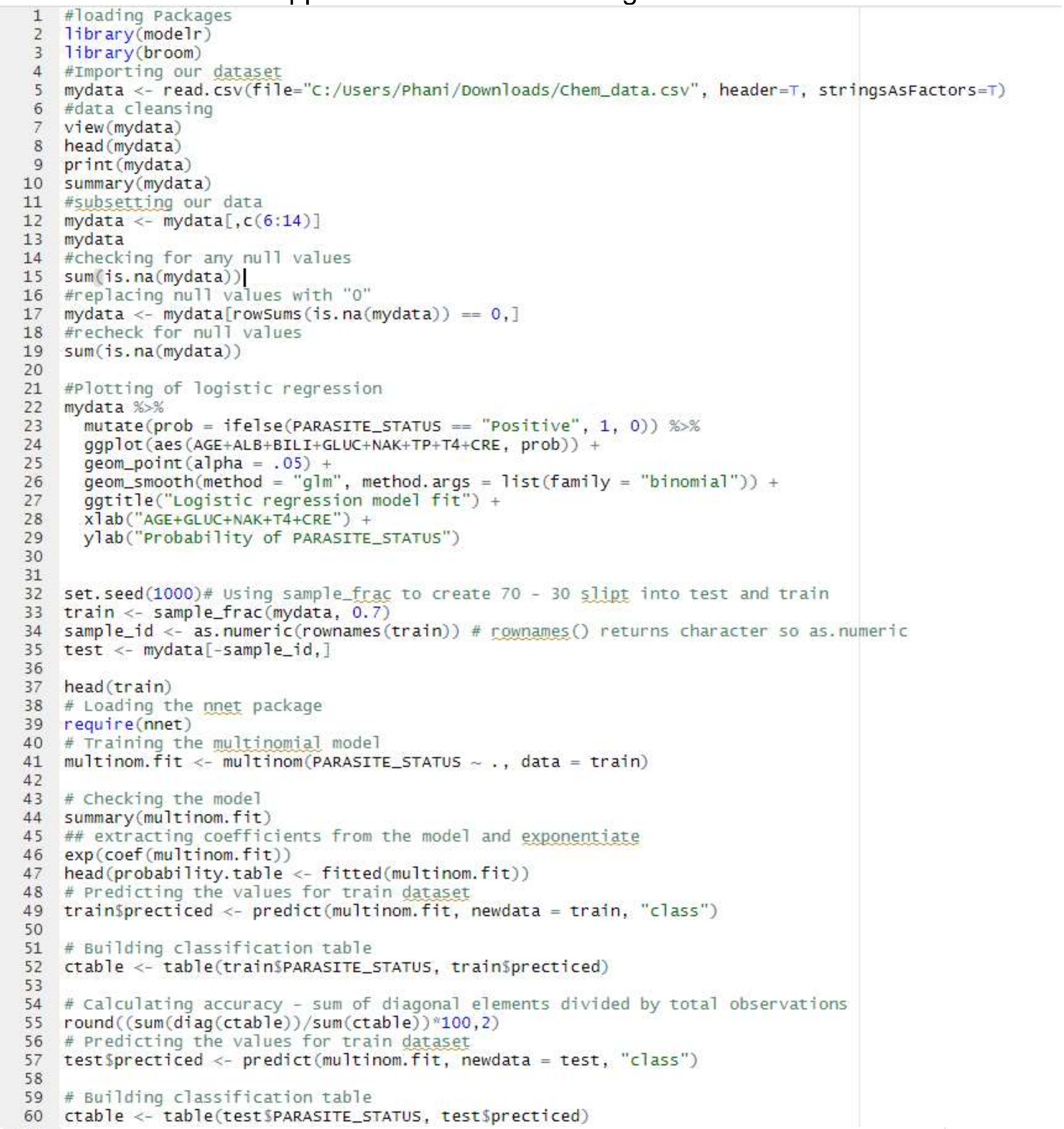
We more testing we do z test and z two-tailed test to more statistical data results for our model data. The probability ratio of selecting an outcome category to the probability of selecting the baseline category is known to be a relative risk and it is also referring to as odds described in the regression predictors. This relative risk is exponentiated right-handed side to a linear equation, pointing to that of the exponentiated regression coefficients are relative risk ratios for a unit change in the predictor variable. We exponentiate the coefficients in our model to check those risk ratios. Our results of both Z test are as shown in Output [7].



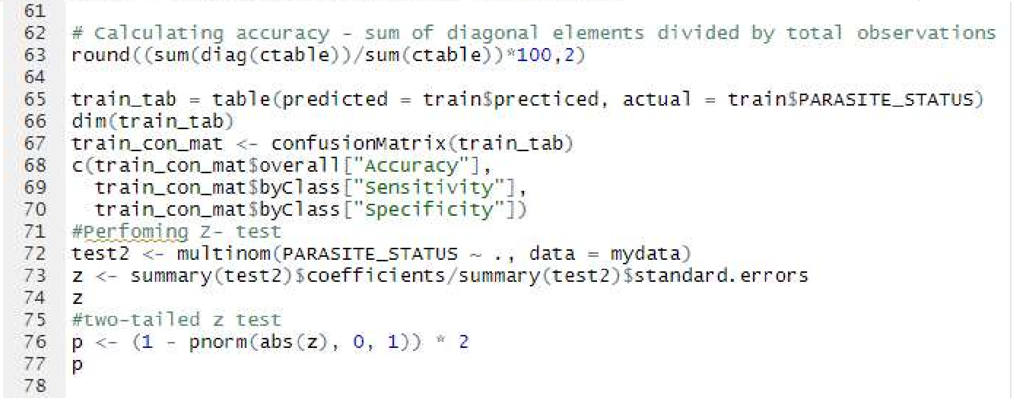
```
> z <- summary(test2)$coefficients/summary(test2)$standard.errors
> z
(Intercept)         AGE         ALB        BILI        GLUC         NAK          TP          T4         CRE 
 -0.2894382  -1.6269395  -3.5762680  -2.6649858  -0.4332446   1.5930045   2.1937556  -1.4110263  -1.6398849 
> #two-tailed z test
> p <- (1 - pnorm(abs(z), 0, 1)) * 2
> p
 (Intercept)          AGE          ALB         BILI         GLUC          NAK           TP           T4          CRE 
0.7722460776 0.1037499463 0.0003485343 0.0076991565 0.6648370794 0.1111591911 0.0282529851 0.1582368534 0.1010291071 
```

Output [7]

Appendix for Multinomial Regression Test



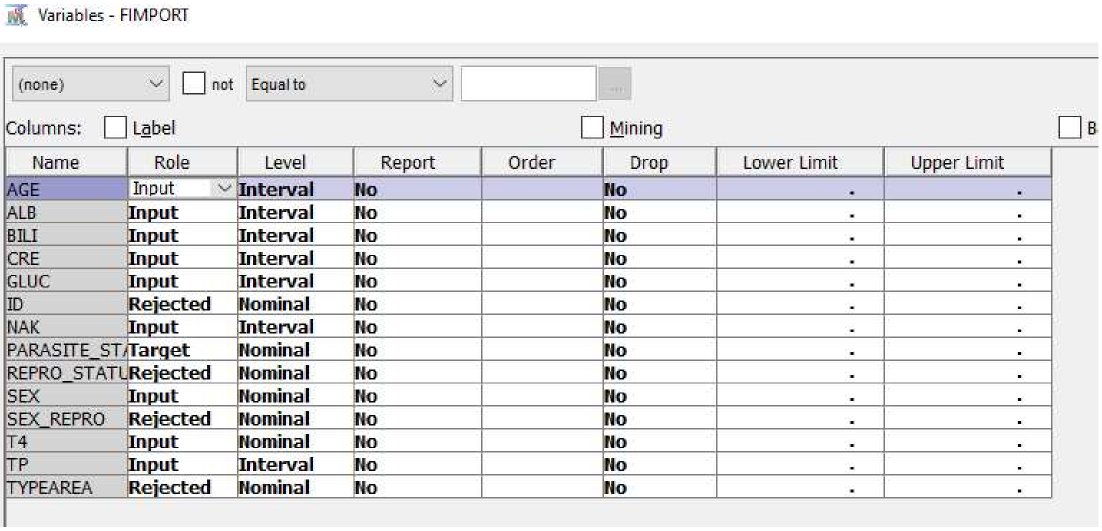
```r
#loading Packages
library(modelr)
library(broom)
#Importing our dataset
mydata <- read.csv(file="C:/Users/Phani/Downloads/Chem_data.csv", header=T, stringsAsFactors=T)
#data cleansing
view(mydata)
head(mydata)
print(mydata)
summary(mydata)
#subsetting our data
mydata <- mydata[,c(6:14)]
mydata
#checking for any null values
sum(is.na(mydata))
#replacing null values with "0"
mydata <- mydata[rowSums(is.na(mydata)) == 0,]
#recheck for null values
sum(is.na(mydata))

#Plotting of logistic regression
mydata %>%
  mutate(prob = ifelse(PARASITE_STATUS == "Positive", 1, 0)) %>%
  ggplot(aes(AGE+ALB+BILI+GLUC+NAK+TP+T4+CRE, prob)) +
  geom_point(alpha = .05) +
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  ggtitle("Logistic regression model fit") +
  xlab("AGE+GLUC+NAK+T4+CRE") +
  ylab("Probability of PARASITE_STATUS")


set.seed(1000)# Using sample_frac to create 70 - 30 slipt into test and train
train <- sample_frac(mydata, 0.7)
sample_id <- as.numeric(rownames(train)) # rownames() returns character so as.numeric
test <- mydata[-sample_id,]

head(train)
# Loading the nnet package
require(nnet)
# Training the multinomial model
multinom.fit <- multinom(PARASITE_STATUS ~ ., data = train)

# Checking the model
summary(multinom.fit)
## extracting coefficients from the model and exponentiate
exp(coef(multinom.fit))
head(probability.table <- fitted(multinom.fit))
# Predicting the values for train dataset
train$precticed <- predict(multinom.fit, newdata = train, "class")

# Building classification table
ctable <- table(train$PARASITE_STATUS, train$precticed)

# Calculating accuracy - sum of diagonal elements divided by total observations
round((sum(diag(ctable))/sum(ctable))*100,2)
# Predicting the values for train dataset
test$precticed <- predict(multinom.fit, newdata = test, "class")

# Building classification table
ctable <- table(test$PARASITE_STATUS, test$precticed)
```



```r

# Calculating accuracy - sum of diagonal elements divided by total observations
round((sum(diag(ctable))/sum(ctable))*100,2)

train_tab = table(predicted = train$precticed, actual = train$PARASITE_STATUS)
dim(train_tab)
train_con_mat <- confusionMatrix(train_tab)
c(train_con_mat$overall["Accuracy"],
  train_con_mat$byClass["Sensitivity"],
  train_con_mat$byClass["Specificity"])
#Perfoming Z- test
test2 <- multinom(PARASITE_STATUS ~ ., data = mydata)
z <- summary(test2)$coefficients/summary(test2)$standard.errors
z
#two-tailed z test
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p

```

B. Regression testing with SAS Miner

To perform any operation first will have to import our dataset into SAS library. We set PARASITE_STATUS as our target variable with binomial mode since it has only to categorical intercept value Positive and Negative and we set all other chosen independent variable as input, and everything is rejected from further functions. McCarthy, R.V., McCarthy, M.M. and Ceccucci, W., 2022. Predictive models using regression. In Applying Predictive Analytics (pp. 87-121). Springer, Cham [8]. We see the same in the Figure 1


Variables - FIMPORT

(none) □ not Equal to

Columns: □ Label □ Mining □ B

Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
AGE	Input	Interval	No		No	.	.
ALB	Input	Interval	No		No	.	.
BILI	Input	Interval	No		No	.	.
CRE	Input	Interval	No		No	.	.
GLUC	Input	Interval	No		No	.	.
ID	Rejected	Nominal	No		No	.	.
NAK	Input	Interval	No		No	.	.
PARASITE_ST/	Target	Nominal	No		No	.	.
REPRO_STATU	Rejected	Nominal	No		No	.	.
SEX	Input	Nominal	No		No	.	.
SEX_REPRO	Rejected	Nominal	No		No	.	.
T4	Input	Nominal	No		No	.	.
TP	Input	Interval	No		No	.	.
TYPEAREA	Rejected	Nominal	No		No	.	.

Figure 1

As next step in regression we drag and drop regression icon on our diagram platform and we then link both our input data table with pre-set variable roles, level with regression diagram as shown in Figure 2.

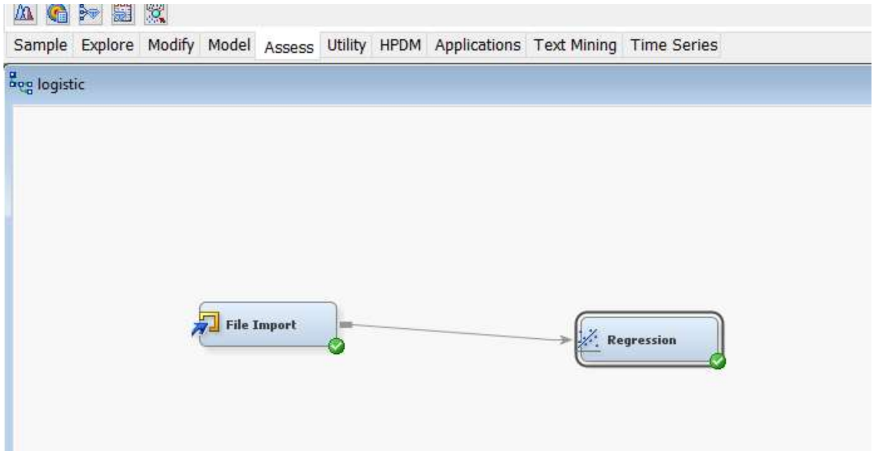
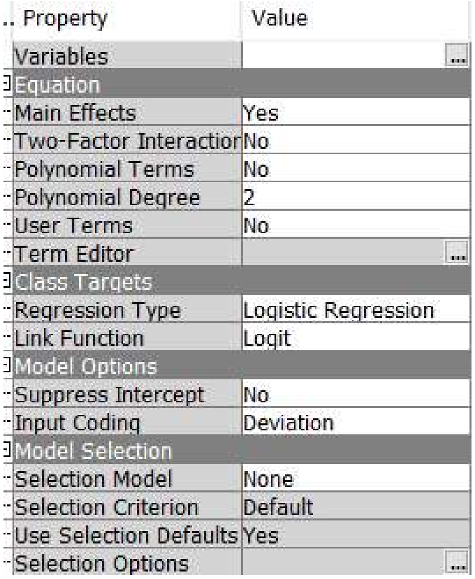


Figure [2]

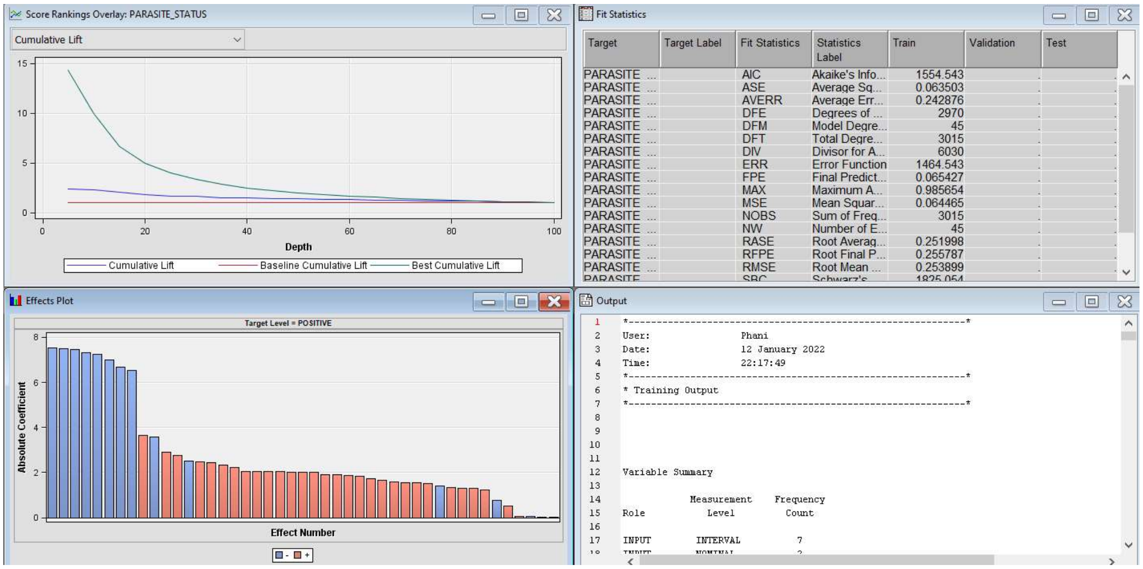
Our setting for regression testing in SAS EM in shown in figure [3]. We select regression method as logistic and predictor to be logit function. SAS Miner does not support direct multinomial function could be the only main shortcoming of testing.



Property	Value
Variables	...
Equation	
Main Effects	Yes
Two-Factor Interactior	No
Polynomial Terms	No
Polynomial Degree	2
User Terms	No
Term Editor	...
Class Targets	
Regression Type	Logistic Regression
Link Function	Logit
Model Options	
Suppress Intercept	No
Input Coding	Deviation
Model Selection	
Selection Model	None
Selection Criterion	Default
Use Selection Defaults	Yes
Selection Options	...

Figure [3]

Thereafter, SAS computes all the variable relation with logistic regression. We have output [8] showing default result plots, predictors, and values of our test results.

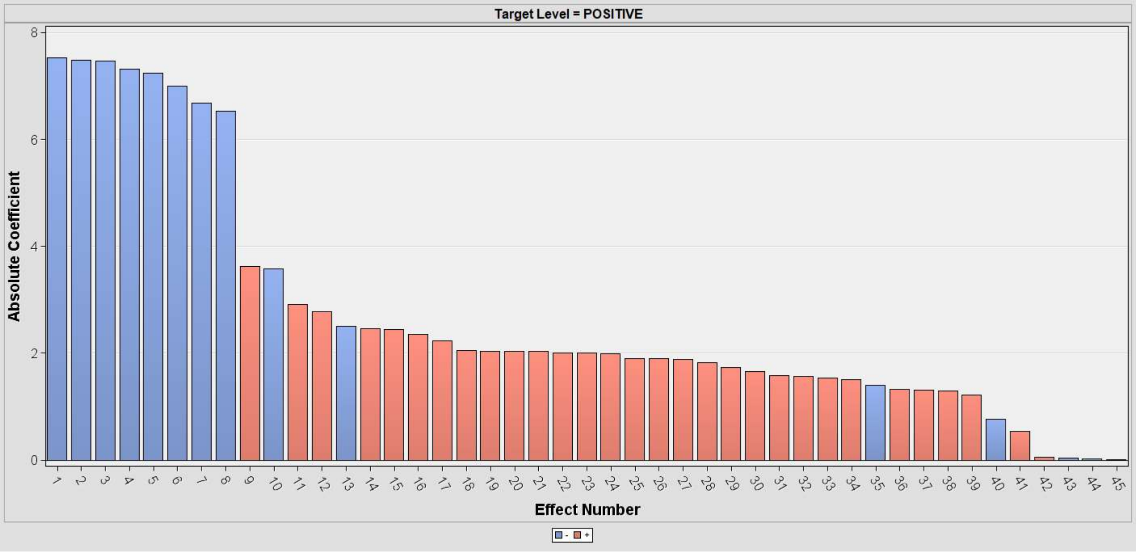


Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
PARASITE ...		AIC	Akaike's Info...	1554.543	.	.
PARASITE ...		ASE	Average Sq...	0.063503	.	.
PARASITE ...		AVERR	Average Err...	0.242876	.	.
PARASITE ...		DFE	Degrees of ...	2970	.	.
PARASITE ...		DFM	Model Degre...	45	.	.
PARASITE ...		DFT	Total Degre...	3015	.	.
PARASITE ...		DIV	Divisor for A...	6030	.	.
PARASITE ...		ERR	Error Function	1464.543	.	.
PARASITE ...		FPE	Final Predict...	0.065427	.	.
PARASITE ...		MAX	Maximum A...	0.985654	.	.
PARASITE ...		MSE	Mean Squar...	0.064465	.	.
PARASITE ...		NOBS	Sum of Freq...	3015	.	.
PARASITE ...		NW	Number of E...	45	.	.
PARASITE ...		RASE	Root Averag...	0.251998	.	.
PARASITE ...		RFPE	Root Final P...	0.255787	.	.
PARASITE ...		RMSE	Root Mean ...	0.253899	.	.
PARASITE		SBC	Schwarz's	1825.054		

```
*------------------------------------------------------------*
User:                Phani
Date:                12 January 2022
Time:                22:17:49
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*




Variable Summary

            Measurement    Frequency
Role           Level         Count

INPUT        INTERVAL          7
INPUT        NOMINAL           2
```

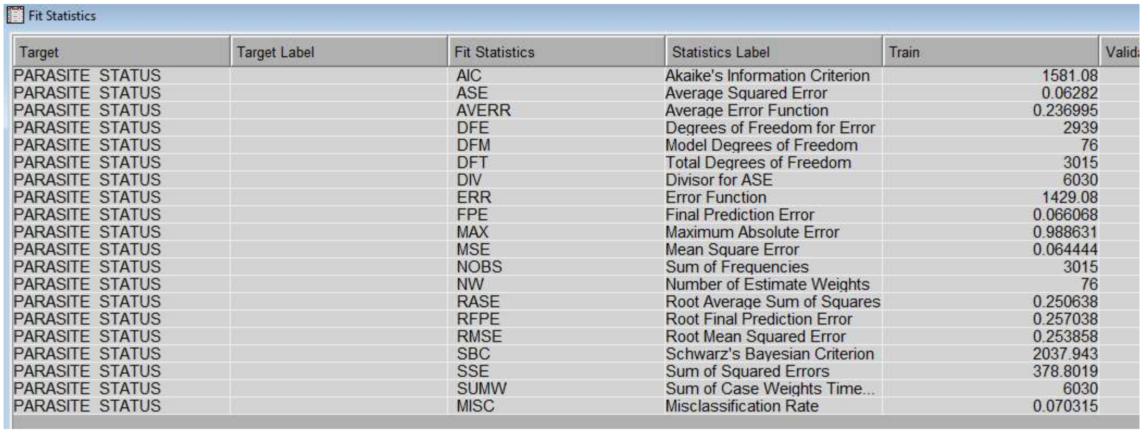
Output 8

We will now verify individual result pane from output 8 and analyse results and compare SAS EM results with R for accuracy and coefficients of variables.



Output 9

Output 9 give us absolute coefficient over effect number, in other Positive and Negative ratio over a x-y plot and visual representation of the data.



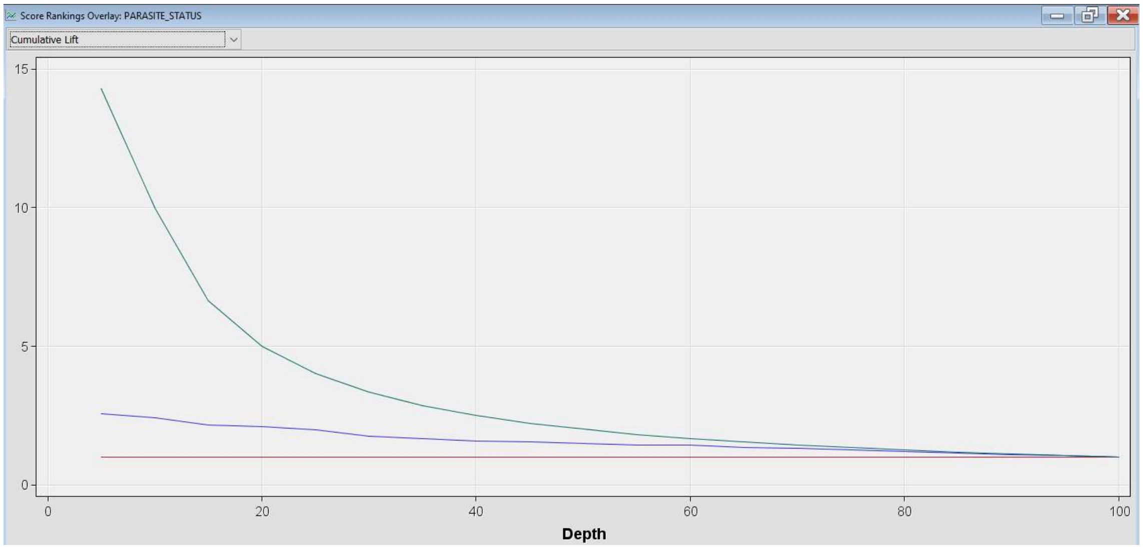
Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Valid
PARASITE STATUS		AIC	Akaike's Information Criterion	1581.08	
PARASITE STATUS		ASE	Average Squared Error	0.06282	
PARASITE STATUS		AVERR	Average Error Function	0.236995	
PARASITE STATUS		DFE	Degrees of Freedom for Error	2939	
PARASITE STATUS		DFM	Model Degrees of Freedom	76	
PARASITE STATUS		DFT	Total Degrees of Freedom	3015	
PARASITE STATUS		DIV	Divisor for ASE	6030	
PARASITE STATUS		ERR	Error Function	1429.08	
PARASITE STATUS		FPE	Final Prediction Error	0.066068	
PARASITE STATUS		MAX	Maximum Absolute Error	0.988631	
PARASITE STATUS		MSE	Mean Square Error	0.064444	
PARASITE STATUS		NOBS	Sum of Frequencies	3015	
PARASITE STATUS		NW	Number of Estimate Weights	76	
PARASITE STATUS		RASE	Root Average Sum of Squares	0.250638	
PARASITE STATUS		RFPE	Root Final Prediction Error	0.257038	
PARASITE STATUS		RMSE	Root Mean Squared Error	0.253858	
PARASITE STATUS		SBC	Schwarz's Bayesian Criterion	2037.943	
PARASITE STATUS		SSE	Sum of Squared Errors	378.8019	
PARASITE STATUS		SUMW	Sum of Case Weights Time...	6030	
PARASITE STATUS		MISC	Misclassification Rate	0.070315	

Output [10]

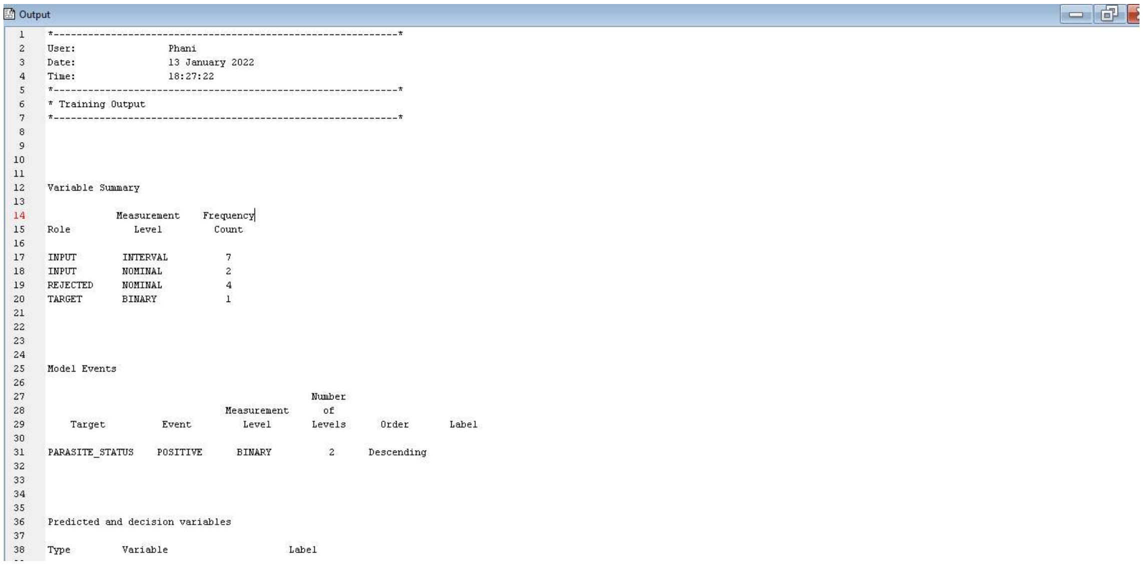
From Output [10] we get the fit statistics of our model, where our categorical dependant variable is computed and presented with different statistical label and their train data values. We have Model Degrees of Freedom at 76, Total Degree of Freedom at 3015 and AIC value of 1581.08. There are no statistical results which we ignore in our research. This can help us predict the coefficients model testing and comparison with R programming results.

Cumulative lift of our categorial variable for 100 percentage of depth is shown in Output [11], we can also calculate lift, gain and response of observations in the same plot with a drop-down menu available on top-right corner.

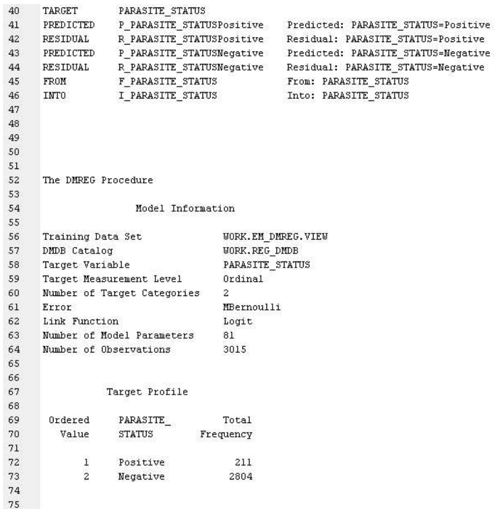


Output[11]

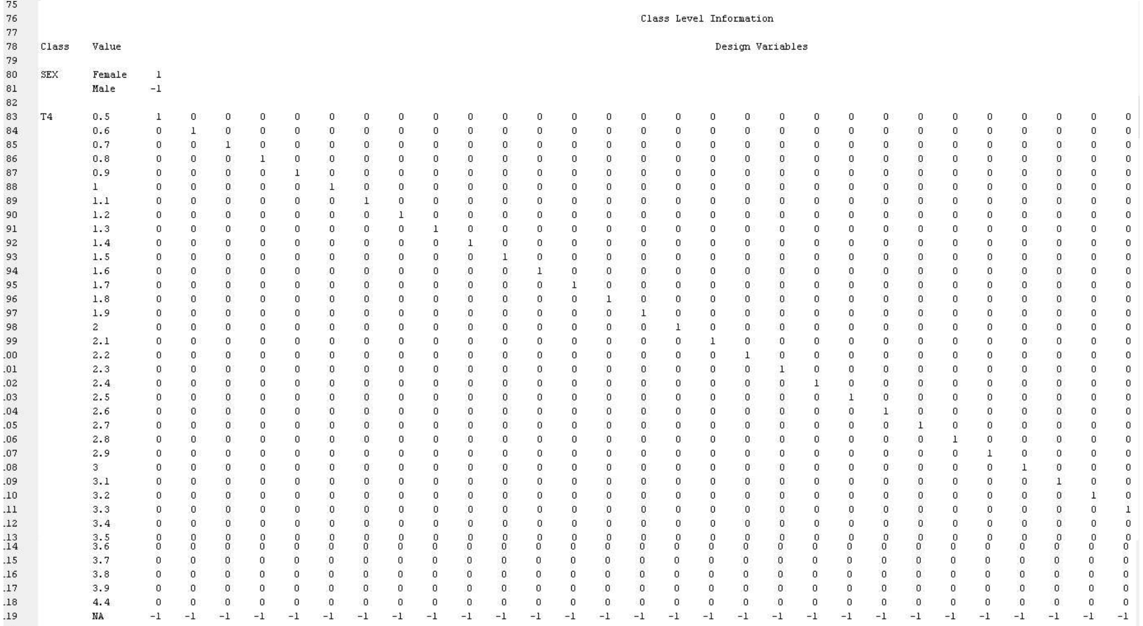
And onto the final and vital output[12] we get information on classification tables, event classification tables, Assessment Score Rankings, Assessment score distribution for train data, our hypothesis test results summary stats.



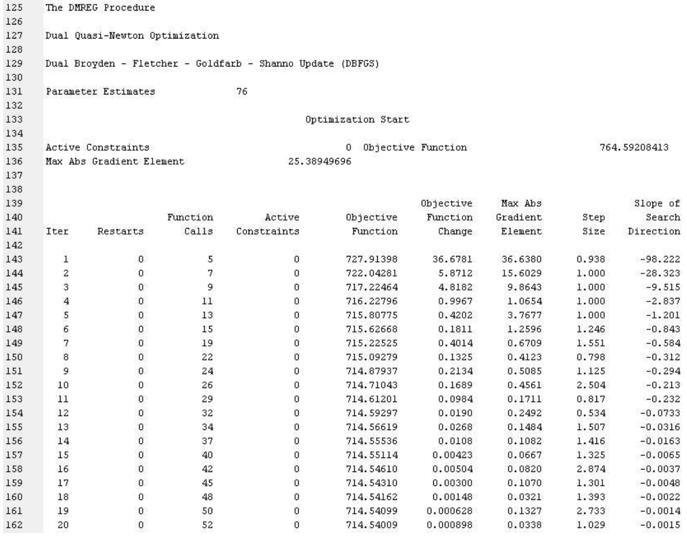
```
*------------------------------------------------------------*
User:                Phani
Date:                13 January 2022
Time:                18:27:22
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*




Variable Summary

            Measurement    Frequency
Role           Level         Count

INPUT        INTERVAL          7
INPUT        NOMINAL           2
REJECTED     NOMINAL           4
TARGET       BINARY            1




Model Events

                                              Number
                              Measurement       of
    Target          Event        Level        Levels      Order       Label

PARASITE_STATUS    POSITIVE      BINARY          2      Descending




Predicted and decision variables

Type         Variable                    Label
```



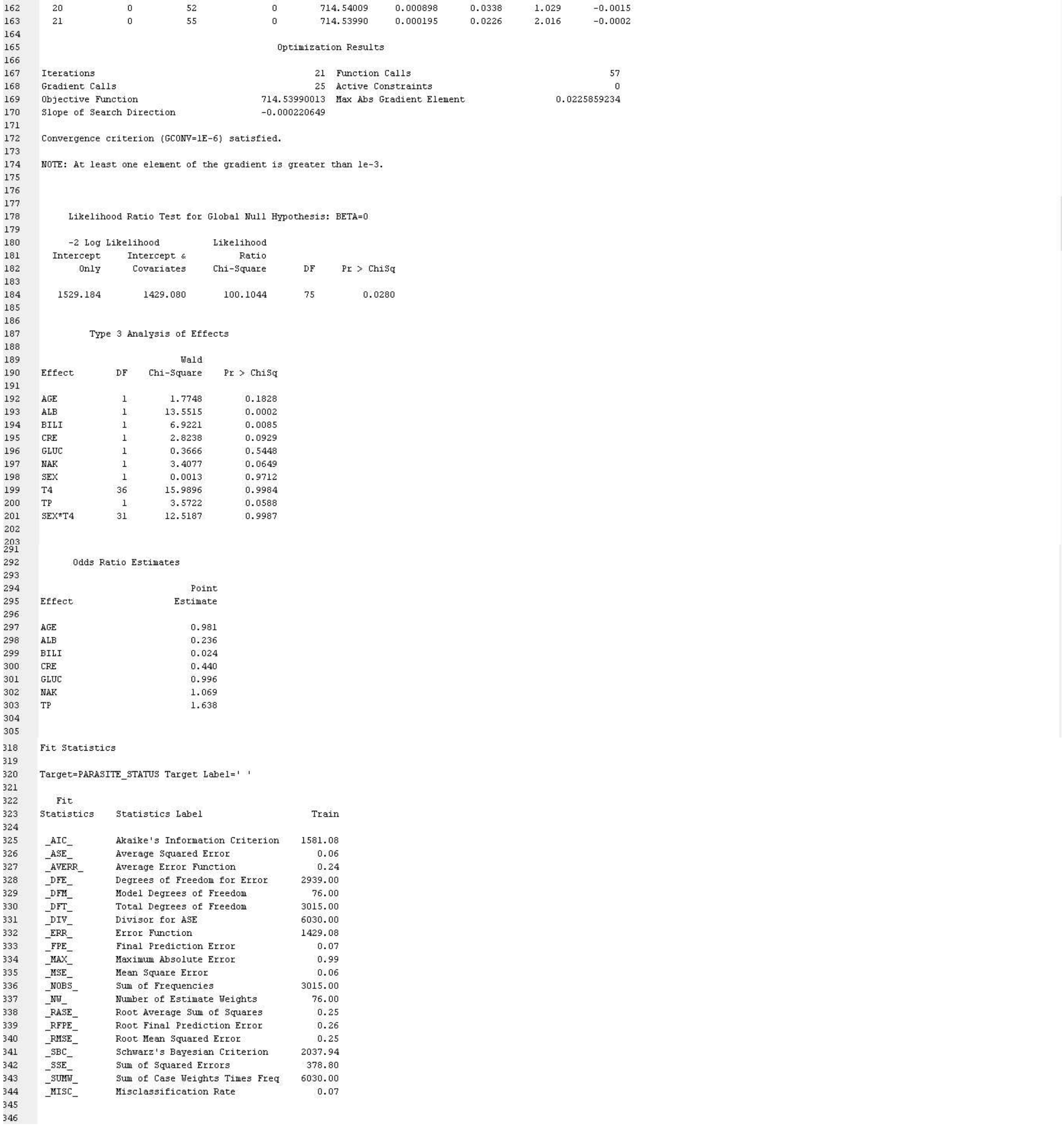
```
TARGET       PARASITE_STATUS
PREDICTED    P_PARASITE_STATUSPositive    Predicted: PARASITE_STATUS=Positive
RESIDUAL     R_PARASITE_STATUSPositive    Residual: PARASITE_STATUS=Positive
PREDICTED    P_PARASITE_STATUSNegative    Predicted: PARASITE_STATUS=Negative
RESIDUAL     R_PARASITE_STATUSNegative    Residual: PARASITE_STATUS=Negative
FROM         F_PARASITE_STATUS            From: PARASITE_STATUS
INTO         I_PARASITE_STATUS            Into: PARASITE_STATUS




The DMREG Procedure

                Model Information

Training Data Set              WORK.EM_DMREG.VIEW
DMDB Catalog                   WORK.REG_DMDB
Target Variable                PARASITE_STATUS
Target Measurement Level       Ordinal
Number of Target Categories    2
Error                          MBernoulli
Link Function                  Logit
Number of Model Parameters     81
Number of Observations         3015


           Target Profile

 Ordered     PARASITE_         Total
   Value     STATUS        Frequency

       1     Positive            211
       2     Negative           2804

```



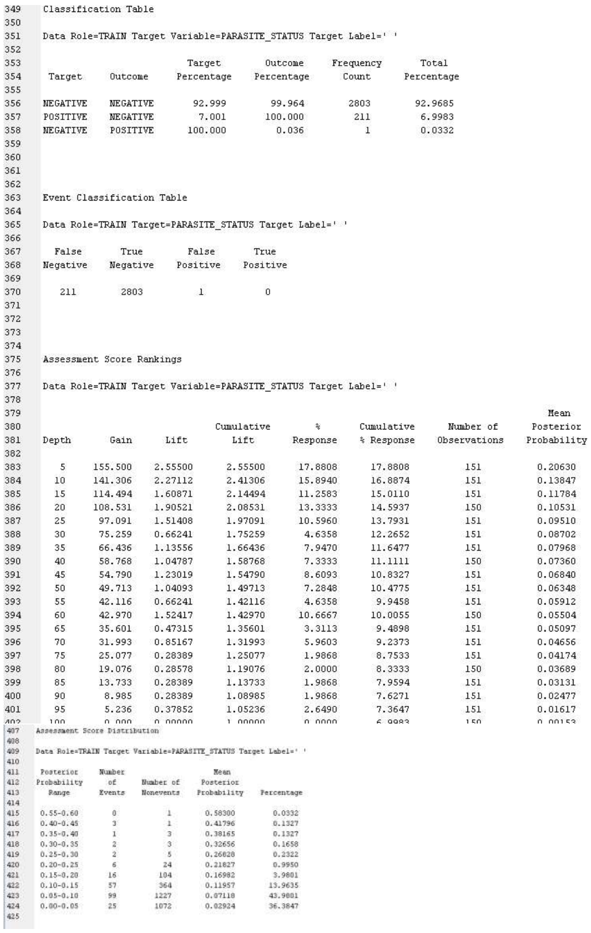
```

                                                                                                         Class Level Information

Class    Value                                                                                                        Design Variables

SEX      Female   1
         Male    -1

T4       0.5      1   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
         0.6      0   1   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
         0.7      0   0   1   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
         0.8      0   0   0   1   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
         0.9      0   0   0   0   1   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   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         1.1      0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         1.2      0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         1.3      0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         1.4      0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         1.5      0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         1.6      0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         1.7      0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         1.8      0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         1.9      0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
         2        0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
         2.1      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
         2.2      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
         2.3      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
         2.4      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
         2.5      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
         2.6      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0
         2.7      0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0
         2.8      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   0   0   0   0
         2.9      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   0   0   0
         3        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   0   0
         3.1      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   0
         3.2      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
         3.3      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
         3.4      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
         3.5      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
         3.6      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
         3.7      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
         3.8      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
         3.9      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
         4.4      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
         NA      -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1
```



```
The DMREG Procedure

Dual Quasi-Newton Optimization

Dual Broyden - Fletcher - Goldfarb - Shanno Update (DBFGS)

Parameter Estimates              76

                                         Optimization Start

Active Constraints                                0  Objective Function                   764.59208413
Max Abs Gradient Element                 25.38949696


                                                      Objective     Max Abs              Slope of
                     Function        Active    Objective     Function    Gradient      Step      Search
Iter     Restarts       Calls   Constraints     Function       Change     Element      Size   Direction

   1            0           5             0    727.91398      36.6781     36.6380     0.938     -98.222
   2            0           7             0    722.04281       5.8712     15.6029     1.000     -28.323
   3            0           9             0    717.22464       4.8182      9.8643     1.000      -9.515
   4            0          11             0    716.22796       0.9967      1.0654     1.000      -2.837
   5            0          13             0    715.80775       0.4202      3.7677     1.000      -1.201
   6            0          15             0    715.62668       0.1811      1.2596     1.246      -0.843
   7            0          19             0    715.22525       0.4014      0.6709     1.551      -0.584
   8            0          22             0    715.09279       0.1325      0.4123     0.798      -0.312
   9            0          24             0    714.87937       0.2134      0.5085     1.125      -0.294
  10            0          26             0    714.71043       0.1689      0.4561     2.504      -0.213
  11            0          29             0    714.61201       0.0984      0.1711     0.817      -0.232
  12            0          32             0    714.59297       0.0190      0.2492     0.534     -0.0733
  13            0          34             0    714.56619       0.0268      0.1484     1.507     -0.0316
  14            0          37             0    714.55536       0.0108      0.1082     1.416     -0.0163
  15            0          40             0    714.55114      0.00423      0.0667     1.325     -0.0065
  16            0          42             0    714.54610      0.00504      0.0820     2.874     -0.0037
  17            0          45             0    714.54310      0.00300      0.1070     1.301     -0.0048
  18            0          48             0    714.54162      0.00148      0.0321     1.393     -0.0022
  19            0          50             0    714.54099     0.000628      0.1327     2.733     -0.0014
  20            0          52             0    714.54009     0.000898      0.0338     1.029     -0.0015
```



```
  20           0          52             0        714.54009     0.000898      0.0338      1.029      -0.0015
  21           0          55             0        714.53990     0.000195      0.0226      2.016      -0.0002

                                            Optimization Results

Iterations                                       21  Function Calls                                   57
Gradient Calls                                   25  Active Constraints                                0
Objective Function                     714.53990013  Max Abs Gradient Element                 0.0225859234
Slope of Search Direction              -0.000220649

Convergence criterion (GCONV=1E-6) satisfied.

NOTE: At least one element of the gradient is greater than 1e-3.



     Likelihood Ratio Test for Global Null Hypothesis: BETA=0

     -2 Log Likelihood          Likelihood
   Intercept     Intercept &         Ratio
        Only      Covariates     Chi-Square       DF     Pr > ChiSq

    1529.184        1429.080       100.1044       75         0.0280


         Type 3 Analysis of Effects

                          Wald
Effect        DF    Chi-Square    Pr > ChiSq

AGE            1        1.7748        0.1828
ALB            1       13.5515        0.0002
BILI           1        6.9221        0.0085
CRE            1        2.8238        0.0929
GLUC           1        0.3666        0.5448
NAK            1        3.4077        0.0649
SEX            1        0.0013        0.9712
T4            36       15.9896        0.9984
TP             1        3.5722        0.0588
SEX*T4        31       12.5187        0.9987



      Odds Ratio Estimates

                            Point
Effect                   Estimate

AGE                         0.981
ALB                         0.236
BILI                        0.024
CRE                         0.440
GLUC                        0.996
NAK                         1.069
TP                          1.638


Fit Statistics

Target=PARASITE_STATUS Target Label=' '

   Fit
Statistics    Statistics Label                   Train

 _AIC_        Akaike's Information Criterion    1581.08
 _ASE_        Average Squared Error                0.06
 _AVERR_      Average Error Function               0.24
 _DFE_        Degrees of Freedom for Error      2939.00
 _DFM_        Model Degrees of Freedom            76.00
 _DFT_        Total Degrees of Freedom          3015.00
 _DIV_        Divisor for ASE                   6030.00
 _ERR_        Error Function                    1429.08
 _FPE_        Final Prediction Error               0.07
 _MAX_        Maximum Absolute Error               0.99
 _MSE_        Mean Square Error                    0.06
 _NOBS_       Sum of Frequencies                3015.00
 _NW_         Number of Estimate Weights          76.00
 _RASE_       Root Average Sum of Squares          0.25
 _RFPE_       Root Final Prediction Error          0.26
 _RMSE_       Root Mean Squared Error              0.25
 _SBC_        Schwarz's Bayesian Criterion      2037.94
 _SSE_        Sum of Squared Errors              378.80
 _SUMW_       Sum of Case Weights Times Freq    6030.00
 _MISC_       Misclassification Rate               0.07
```



```
Classification Table

Data Role=TRAIN Target Variable=PARASITE_STATUS Target Label=' '

                           Target       Outcome      Frequency       Total
 Target     Outcome     Percentage    Percentage       Count      Percentage

NEGATIVE    NEGATIVE       92.999        99.964        2803         92.9685
POSITIVE    NEGATIVE        7.001       100.000         211          6.9983
NEGATIVE    POSITIVE      100.000         0.036           1          0.0332




Event Classification Table

Data Role=TRAIN Target=PARASITE_STATUS Target Label=' '

  False       True        False       True
Negative    Negative    Positive    Positive

   211        2803          1           0




Assessment Score Rankings

Data Role=TRAIN Target Variable=PARASITE_STATUS Target Label=' '

                                                                                   Mean
                                  Cumulative        %       Cumulative      Number of     Posterior
Depth       Gain      Lift           Lift      Response     % Response    Observations   Probability

   5     155.500    2.55500      2.55500      17.8808       17.8808          151          0.20630
  10     141.306    2.27112      2.41306      15.8940       16.8874          151          0.13847
  15     114.494    1.60871      2.14494      11.2583       15.0110          151          0.11784
  20     108.531    1.90521      2.08531      13.3333       14.5937          150          0.10531
  25      97.091    1.51408      1.97091      10.5960       13.7931          151          0.09510
  30      75.259    0.66241      1.75259       4.6358       12.2652          151          0.08702
  35      66.436    1.13556      1.66436       7.9470       11.6477          151          0.07968
  40      58.768    1.04787      1.58768       7.3333       11.1111          150          0.07360
  45      54.790    1.23019      1.54790       8.6093       10.8327          151          0.06840
  50      49.713    1.04093      1.49713       7.2848       10.4775          151          0.06348
  55      42.116    0.66241      1.42116       4.6358        9.9458          151          0.05912
  60      42.970    1.52417      1.42970      10.6667       10.0055          150          0.05504
  65      35.601    0.47315      1.35601       3.3113        9.4898          151          0.05097
  70      31.993    0.85167      1.31993       5.9603        9.2373          151          0.04656
  75      25.077    0.28389      1.25077       1.9868        8.7533          151          0.04174
  80      19.076    0.28578      1.19076       2.0000        8.3333          150          0.03689
  85      13.733    0.28389      1.13733       1.9868        7.9594          151          0.03131
  90       8.985    0.28389      1.08985       1.9868        7.6271          151          0.02477
  95       5.236    0.37852      1.05236       2.6490        7.3647          151          0.01617
 100       0.000    0.00000      1.00000       0.0000        6.9983          150          0.00153
Assessment Score Distribution

Data Role=TRAIN Target Variable=PARASITE_STATUS Target Label=' '

 Posterior     Number                      Mean
Probability      of      Number of      Posterior
   Range       Events    Nonevents     Probability    Percentage

 0.55-0.60        0            1        0.58300        0.0332
 0.40-0.45        3            1        0.41796        0.1327
 0.35-0.40        1            3        0.38165        0.1327
 0.30-0.35        2            3        0.32656        0.1658
 0.25-0.30        2            5        0.26828        0.2322
 0.20-0.25        6           24        0.21827        0.9950
 0.15-0.20       16          104        0.16982        3.9801
 0.10-0.15       57          364        0.11957       13.9635
 0.05-0.10       99         1227        0.07118       43.9801
 0.00-0.05       25         1072        0.02924       36.3847

```

Output ^{11}

Figure ^{12} illustrates 3D representation of support, lift and confidence level and count is intersected with different colours between them.

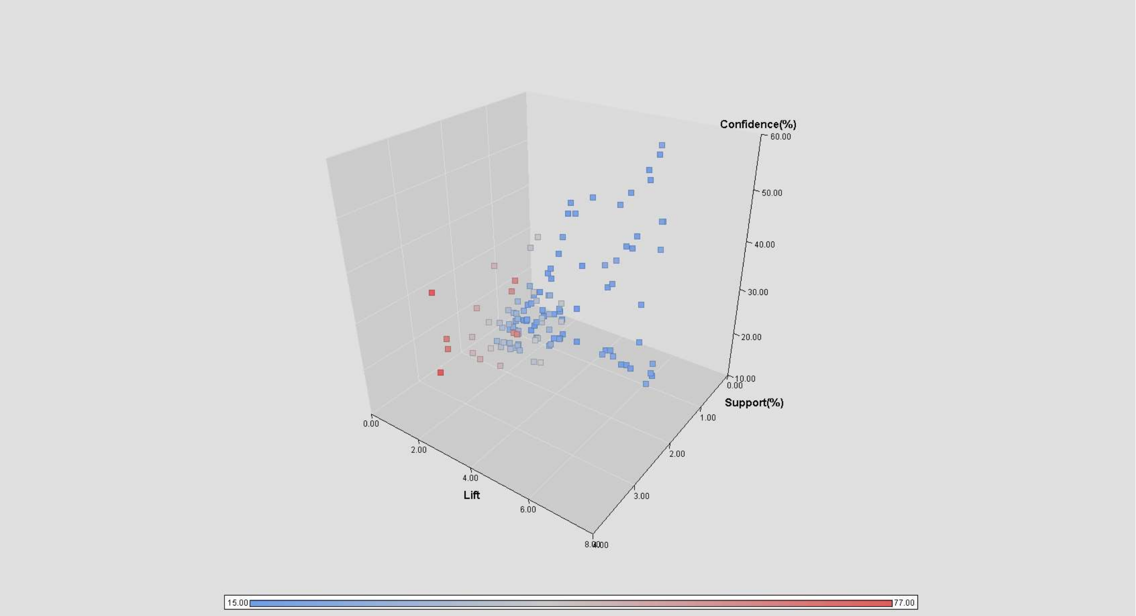


Figure 12

5. Conclusions and output comparison of R and SAS EM methods

In our conclusion we will come across comparative results of R programming and SAS Enterprise Miner. In R program our results yield accuracy of multinomial logistic regression model on test model is with 100% which 7.71% higher to that of train model which shows our model is a successful fit to our data. As we already discussed the higher AIC value is of the test the more accurate it will be in prediction. R program AIC results of categorical dependent variable with all other dependent variables is AIC=1026.642 and by comparing this with SAS EM result AIC=1581.08, we can conclude that SAS EM has better predicted the model of fit.

The standard error rate is low in SAS EM model of fit on comparison with R program statistics. But z test and z two-tailed proves our model in R programming is of high accuracy. Our test and train methods in multinomial logistic regression in R gives us option of choosing 100 percent accurate model in favour of high probability on SAS EM for our dataset.

References

1. Starkweather, J. and Moske, A.K., 2011. Multinomial logistic regression.
2. Hedeker, D., 2003. A mixed-effects multinomial logistic regression model. Statistics in medicine, 22(9), pp.1433-1446.
3. Böhning, D., 1992. Multinomial logistic regression algorithm. Annals of the institute of Statistical Mathematics, 44(1), pp.197-200.
4. Fagerland, M.W. and Hosmer, D.W., 2012. A generalized Hosmer–Lemeshow goodness-of-fit test for multinomial logistic regression models. The Stata Journal, 12(3), pp.447-453.
5. Bayaga, A., 2010. Multinomial Logistic Regression: Usage and Application in Risk Analysis. Journal of applied quantitative methods, 5(2).
6. https://data.world/ehales/grls-parasite-study/workspace/file?filename=Chem_data.csv