**Startup Success Prediction**

**DSCI 5240: Data Mining**

**FINAL PROJECT**

**Prof. Mahdi Fathi**

Department of Information Technology and Decision Sciences

**Group 12**

SRITHA DARBHA

SRAVANTHI PERI

KOMAL SINGH BHAMRA

ARBAN MOHAMMED

UDAYA SREE DOKKA

**TABLE OF CONTENTS**

Objective ---------------------------------------------------------------------------------- 3

Project background ---------------------------------------------------------------------- 3

Dataset description ---------------------------------------------------------------------- 3

Analysing the data ---------------------------------------------------------------------- 4

Principal Component Analysis and Logistic Regression ------------------------- 10

Decision Tree --------------------------------------------------------------------------- 19

Auto Neural ----------------------------------------------------------------------------- 26

Result: Model Comparison ----------------------------------------------------------- 29

Conclusion ------------------------------------------------------------------------------ 30

Alternative Work ----------------------------------------------------------------------- 31

References ------------------------------------------------------------------------------ 38

**Objective**

The aim of our project is to analyse the dataset Start-up Success Prediction dataset to classify the data by using various classification models and compare the misclassification rate between these models.

Predicting the success of a start-up allows investors to find companies that have the potential for rapid growth, thereby allowing them to be one step ahead of the competition.

**Project Background**

A start-up is a company begun by an entrepreneur to seek, develop, and validate a scalable economic model. While entrepreneurship refers to all new businesses, including self-employment and businesses that never intend to become registered, start-ups refer to new businesses that intend to grow large beyond the solo founder. Start-ups face high uncertainty and have high rates of failure, but a minority of them do go on to be successful and influential. Start-up’s play a major role in economic growth. They bring new ideas, spur innovation, create employment, thereby moving the economy. There has been an exponential growth in start-up’s over the past few years. [Source of information: Wikipedia].

**Dataset Description**

The start-up dataset is a second hand dataset and has been taken from the following website: [https://www.kaggle.com/manishkc06/startup-success-prediction](https://www.kaggle.com/manishkc06/startup-success-prediction%0d)

The data contains industry trends, investment insights and individual company information. There are 47 columns/features.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Variable** | **Type** | **Description** |
| 1 | State\_Code | quantitative | State name shortcut |
| 2 | Latitude | quantitative | Location of the Startup |
| 3 | Longitude | quantitative | Location of the Startup |
| 4 | Zip-Code | quantitative | Area code |
| 5 | ID | quantitative | Company ID |
| 6 | City | quantitative | City Location |
| 7 | Complete Address | quantitative | Complete Address of the Startup |
| 8 | Name | quantitative | Company Name |
| 9 | Founded\_At | quantitative | Start Date |
| 10 | Closed\_At | quantitative | End Date |
| 11 | First\_Funding\_At | quantitative | First Funding Date |
| 12 | Last\_Funding\_At | quantitative | Last Funding Date |
| 13 | Age\_FIrst\_Funding\_Year | quantitative | Start-up Age when received First Funding |
| 14 | Age\_Last\_Funding \_Year | quantitative | Startup Age when received Last Funding |
| 15 | Age\_First\_Milestone\_year | quantitative | Startup Age when hit First Milestone |
| 16 | Age\_Last\_milestone\_year | quantitative | Startup Age when hit Last Milestone |
| 17 | Relationships | quantitative | Number of Relationships |
| 18 | Funding\_Rounds | quantitative | Number of times Funding Received |
| 19 | Funding\_Total\_USD | quantitative | Total Funding in USD |
| 20 | Milestones | quantitative | Number of Milestones hit by the Startup |
| 21 | state\_code | quantitative | State name shortcut |
| 22 | is\_CA | categorical | Is the Startup State in California |
| 23 | is\_NY | categorical | Is the Startup State in New York |
| 24 | is\_MA | categorical | Is the Startup State in Massachusetts |
| 25 | is\_TX | categorical | Is the Startup in Texas |
| 26 | is\_otherstate | categorical | Is the Startup State in anyother State |
| 27 | category\_code | quantitative | Category of the Startup Company |
| 28 | is\_software | categorical | What kind of Startup Company |
| 29 | is\_web | categorical | What kind of Startup Company |
| 30 | is\_mobile | categorical | What kind of Startup Company |
| 31 | is\_enterprise | categorical | What kind of Startup Company |
| 32 | is\_advertising | categorical | What kind of Startup Company |
| 33 | is\_gamesvideo | categorical | What kind of Startup Company |
| 34 | is\_ecommerce | categorical | What kind of Startup Company |
| 35 | is\_biotech | categorical | What kind of Startup Company |
| 36 | is\_consulting | categorical | What kind of Startup Company |
| 37 | is\_othercategory | categorical | What kind of Startup Company |
| 38 | Object\_id | quantitative | The ID of the object |
| 39 | has\_vc | categorical | Is startup having Venture Capital |
| 40 | has\_angel | categorical | Is startup having Angel Investor |
| 41 | has\_roundA | categorical | Is startup having round A Venture Captital Financing |
| 42 | has\_roundB | categorical | Is startup having round B Venture Captital Financing |
| 43 | has\_roundC | categorical | Is startup having round C Venture Captital Financing |
| 44 | has\_roundD | categorical | Is startup having round D Venture Captital Financing |
| 45 | Avg\_partcipants | quantitative | Average Number of Participants |
| 46 | Is top500 | categorical | Is startup in Top500 |
| 47 | Status | categorical | If the company is Acquired or Closed. |

The Class Distribution is Acquired is 597 and Closed is 326.

The target variable ‘Status’ is the field that we are predicting which takes two values.

a. Acquired : The success of a company is defined as the event that gives the company's founders a large sum of money through the process of M&A (Merger and Acquisition). It is indicated as 1 in the dataset.

b. Closed : The failure of a company is defined as the event where it had to be Closed or Shut down. It is indicated as 0 in the dataset.

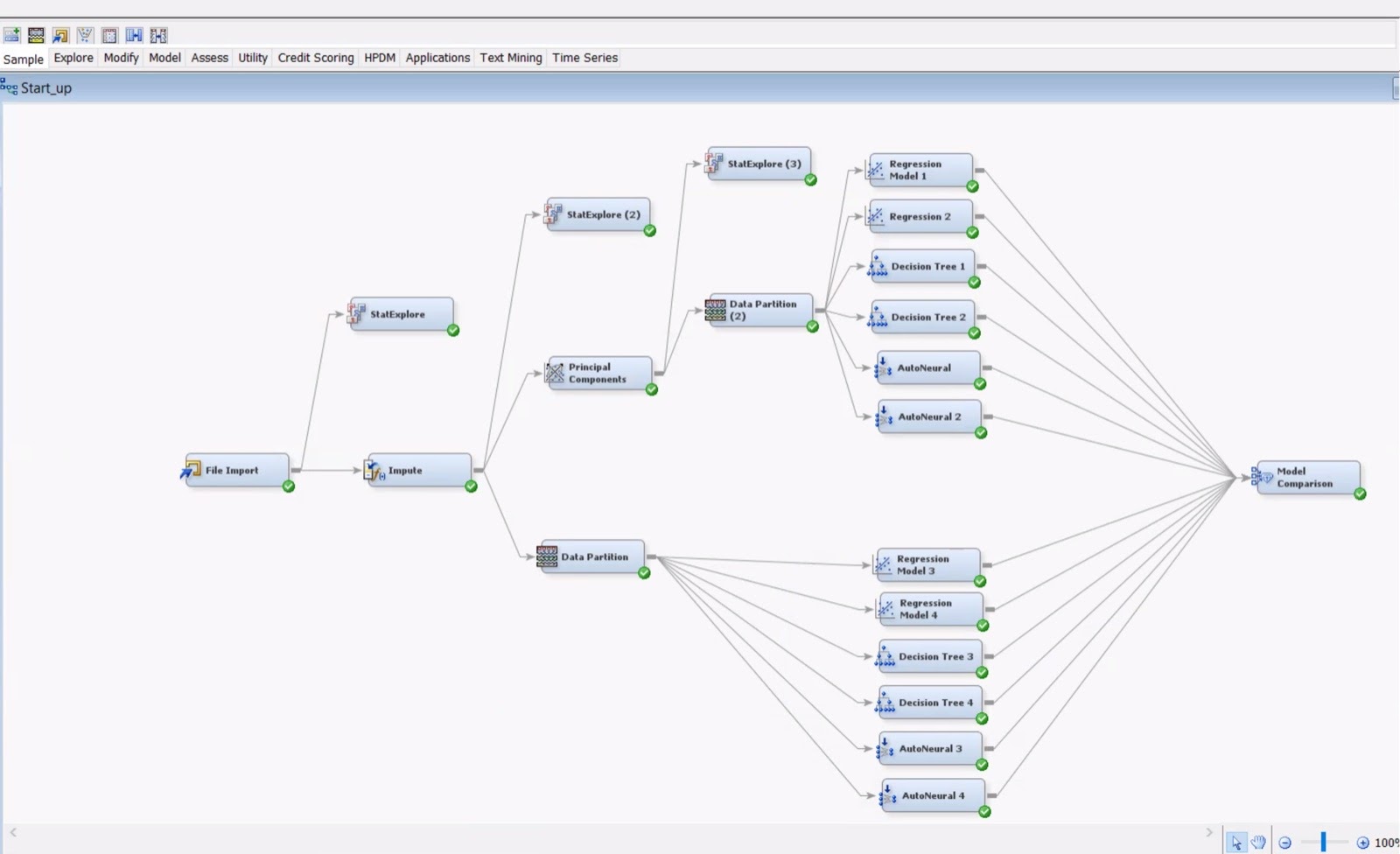
**Analysing the Data**

We have used the following four models to analyse our data:

* Principal Component Analysis
* Logistic Regression
* Decision Tree
* Auto Neural Networks

We used and compared these four models to determine the most efficient model. The advantages and disadvantages of various models are assessed, and model comparison node results are taken into account in determining an efficient model.

*Screenshot showing the complete diagram:*



**Fig. 1**

We are using SAS Enterprise Miner 15.1 to analyse the dataset. First, we have loaded the dataset into the SAS Enterprise using the File Import Node. We then explored the data. We found 10 variables with Number of levels more than 128. We changed the roles for these 10 variables to Rejected and Text. We then added the StatExplore Node and found there are missing values for age\_first\_milestone\_year and age\_last\_milestone\_year columns . We then ran the Impute node to fix these missing values. We set the Default Input method for Class variables and Interval variables to Tree Surrogate.

*Screenshot showing File Import Variables:*

Table

Description automatically generated

**Fig. 2**

*Screenshot showing the Explore tab of File Import:*

Graphical user interface, application, table, Excel

Description automatically generated

**Fig. 3**

*Screenshot showing missing values in results of Stat Explore:*

Table

Description automatically generated **Fig. 4**

*Screenshot showing missing values Role set to Rejected and Text in File Import Edit Variables:*

Table

Description automatically generated

**Fig. 5**

We then add the Data Partition node and assign 70% to trained data and 30% to validation data. This is the result of the Data Partition.

*Table 1:*

Text, letter

Description automatically generated

**Principal Component Analysis and Logistic Regression**

We now want to analyse the data to check for Dimensionality Reduction. We ran the Principal Component Analysis to the Impute Node. This is represented by the Principal Component Analysis Node in the main diagram. The variable count is reduced to 20 from 47. We then ran Logistic Regression with all variables again to check for the R-square and Adj R-square values. This is represented by the Regression Node 1 in the main diagram. This regression model is significant as p value in F- test is <.0001 which is less than 0.05. The following variables are significant after first regression:

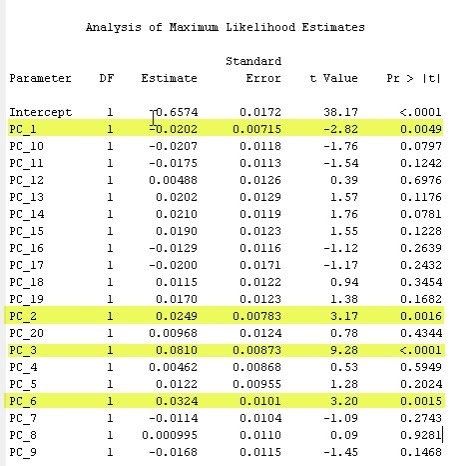
1.     PC\_1

2.     PC\_2

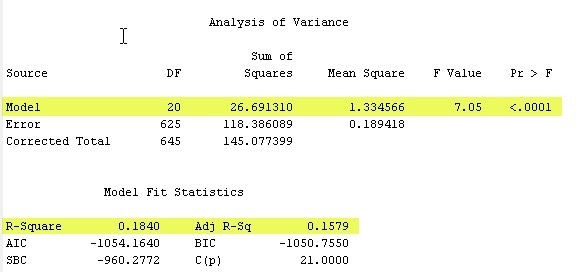
3.     PC\_3

4.     PC\_6

*Table 2:*

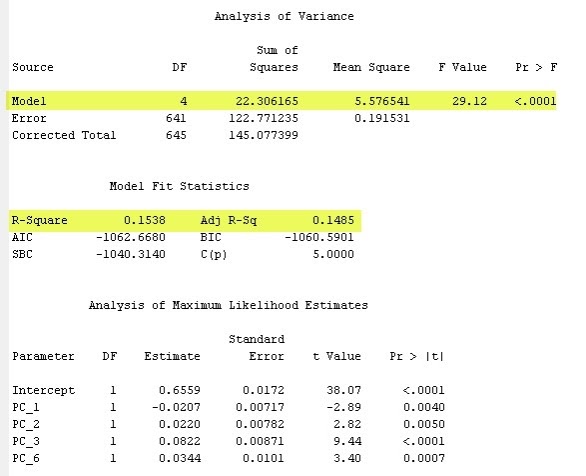


*Table 3:*



The R-square value is 0.1840 and Adj R-Square value is 0.1579. This means that R-square is about 18.40% of the variation which was explained by the variables. Now, we again set the use of insignificant value to NO and ran the regression node again.

*Table 4:*

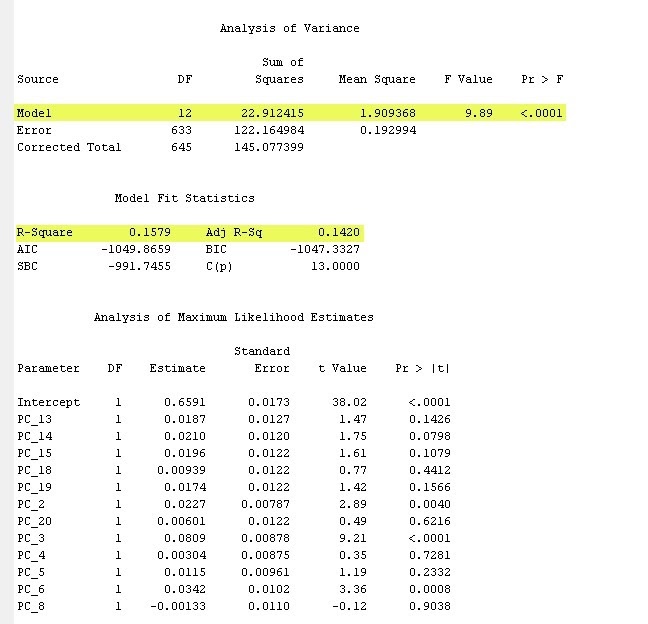


The R-square value is 0.1538 and Adj R-Square value is 0.1485. This means that R-square is about 15.38% of the variation which was explained by the variables.

In the above Regression Model we have considered all the variables both positively and negatively correlated but now we will again run a regression model with only positively correlated variables. We then ran a regression node with all variables to check for the R-square and Adj R-square values. We are using Logistic Regression for our data. This is represented by the Regression Node 2 in the main diagram.. This regression model is significant as p value in F- test is <.0001 which is less than 0.05.

We have not removed any insignificant values because we already have not observed much changes in the R-square and adjusted R-square value although we have removed the insignificant values.

*Table 5:*



The R-square value is 0.1579 and Adj R-Square value is 0.1420. This means that R-square is about 15.79% of the variation which was explained by the variables.

We then ran a regression node with all variables to check for the R-square and Adj R-square values. We are using Logistic Regression for our data. This regression model is significant as p value in F- test is <.0001 which is less than 0.05.This is represented by the Regression Node 3 in the main diagram.

The following variables are significant after first regression:

1. IMP\_age\_Last\_milestone\_year
2. Avg\_partcipants
3. has\_vc
4. is\_otherstate
5. Is\_top500
6. Relationships

*Table 6:*

Graphical user interface, application, table

Description automatically generated

*Table 7:*

Timeline

Description automatically generated

The R-square value is 0.3299 and Adj R-Square value is 0.2323. This means that R-square is about 32.99% of the variation which was explained by the variables. We will now remove the insignificant values by setting their Use to No and run the regression again. Although the model is significant as p value in F- test is <.0001 but we still have is\_otherstate as insignificant.

*Table 8:*

Table

Description automatically generated

*Table 9:*

Timeline

Description automatically generated

Now, the R-square value is 0.3237 and Adj R-Square value is 0.2307. This means that R-square is about 32.37% of the variation which was explained by the variables. We again remove the insignificant values by setting their use to No. We then run the Regression node again.

*Table 10:*

Table

Description automatically generated

In the above Regression Model we have considered all the variables both positively and negatively skewed but now we will again run a regression model with only positively skewed variables. We then ran a regression node with all variables to check for the R-square and Adj R-square values. We are using Logistic Regression for our data. This regression model is significant as p value in F- test is <.0001 which is less than 0.05.This is represented by the Regression Node 4 in the main diagram.

The following variables are significant after first regression:

1. IMP\_age\_Last\_milestone\_year
2. Age\_last\_funding\_year
3. Avg\_partcipants
4. Is\_top500
5. Relationships
6. State\_code

*Table 11:*

Table

Description automatically generated

*Table 12:*

Table, timeline

Description automatically generated

The R-square value is 0.2382 and Adj R-Square value is 0.2126. This means that R-square is about 32.77% of the variation which was explained by the variables. We will now remove the insignificant values by setting the Use to No and run the regression again.

*Table 13:*

Table

Description automatically generated with medium confidence

The R-square value is 0.2554 and Adj R-Square value is 0.2127. This means that R-square is about 25.54% of the variation which was explained by the variables. We still have state\_code as insignificant value. Now we again set the use of insignificant value to NO and ran the regression node again.

*Table 14:*

Table

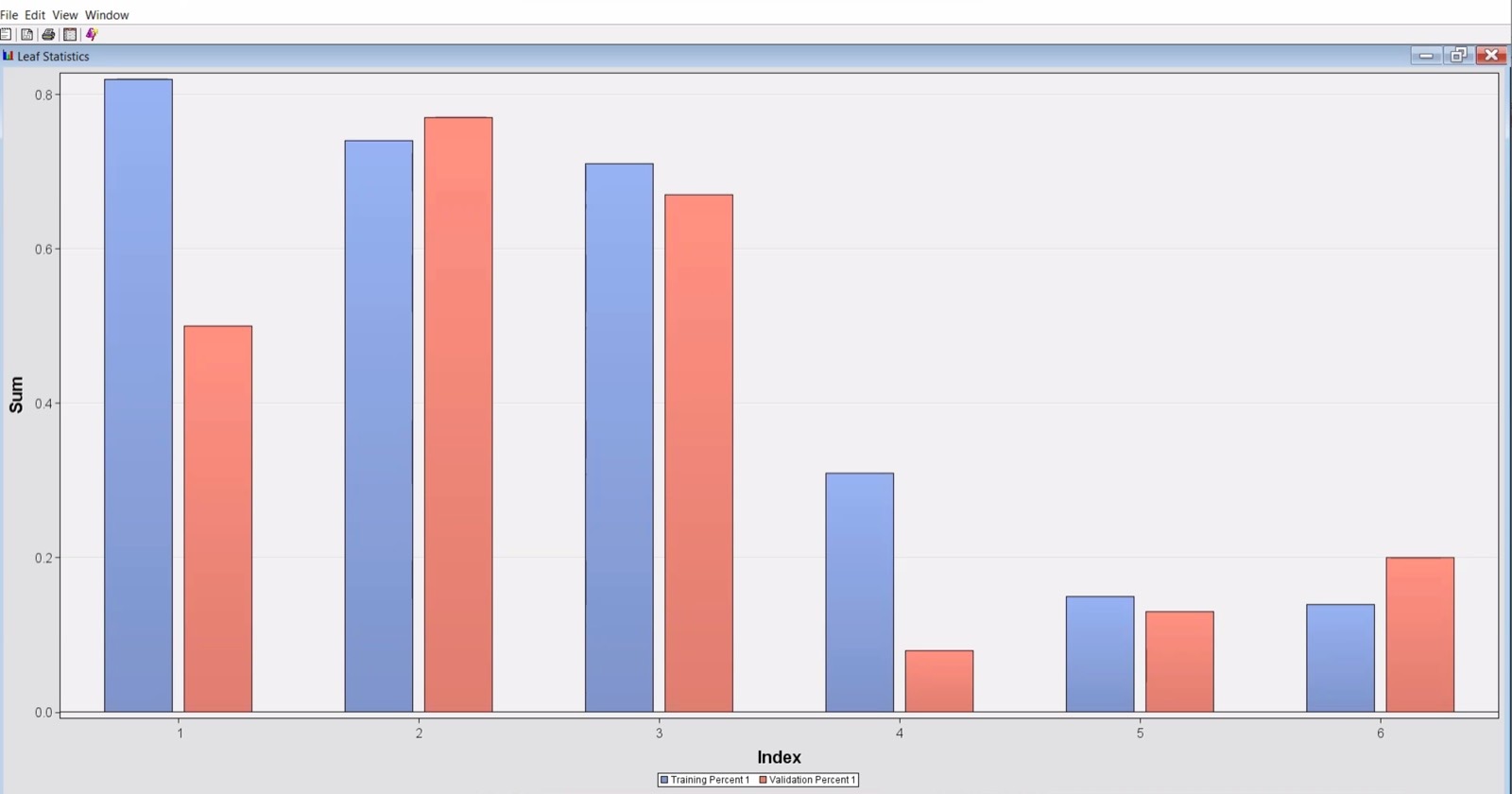
Description automatically generated with low confidence

The R-square value is 0.2140 and Adj R-Square value is 0.2079. This means that R-square is about 21.40% of the variation which was explained by the variables.

**Decision Tree**

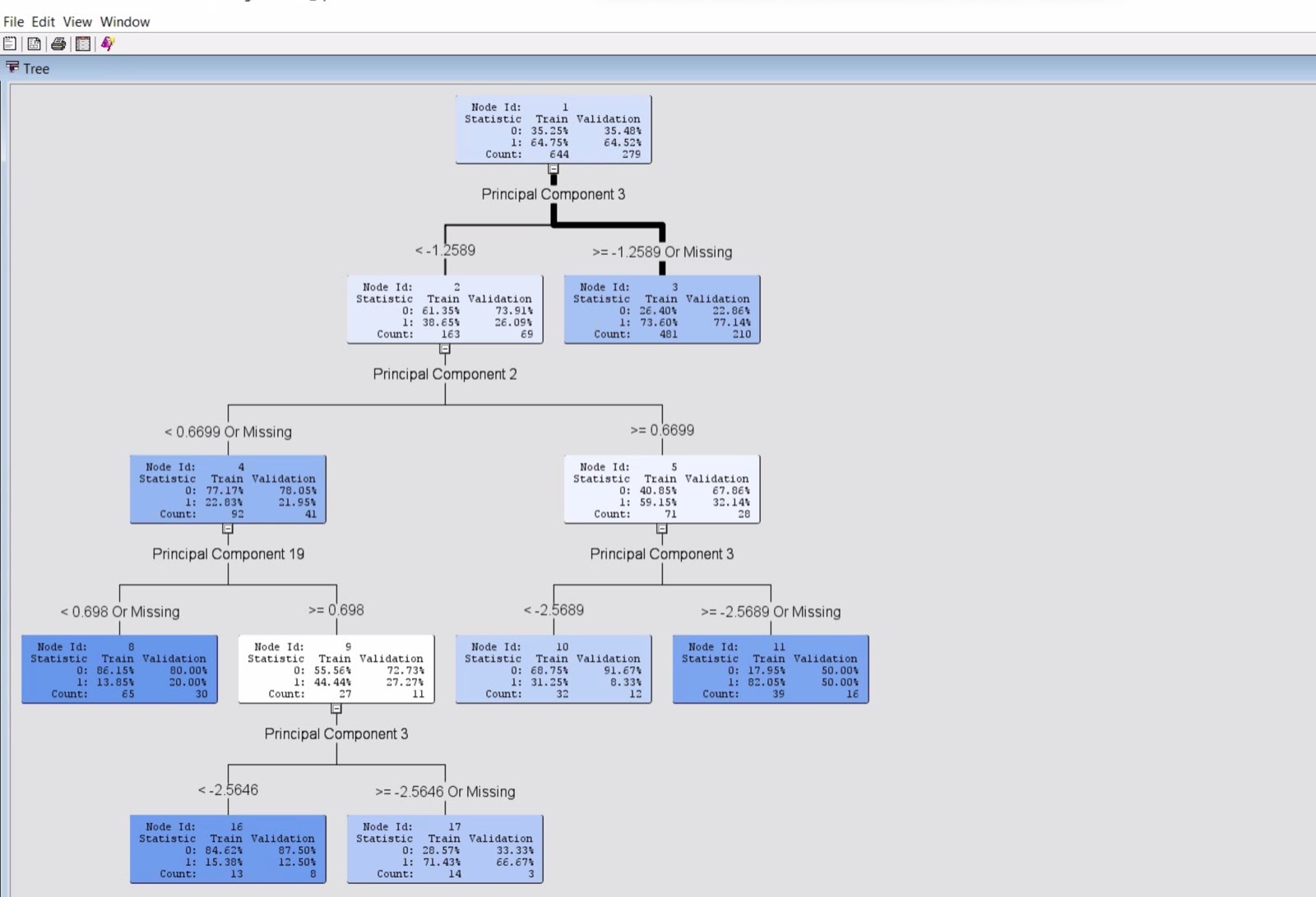
We are now using a Decision Tree node to analyse the data. First, we wanted to analyse the data by running the Principal Component Analysis and selecting all variables. This is represented by the Decision Tree 1 Node in the diagram. We found there are 6 leaves in the optimal tree when we used the Decision Tree node, when we kept the maximum number of branches to two-way splits and then we kept the decision rate as the model assessment statistic. There are four splits for this decision tree. The variable PC3 was used for the first split. For the second split, the variable PC2 was used. For the third split, the variables PC19 and PC3 were used. For the fourth split, PC3 was used.

*Screenshot showing the Leaf Statistics over Sum and Index axis:*



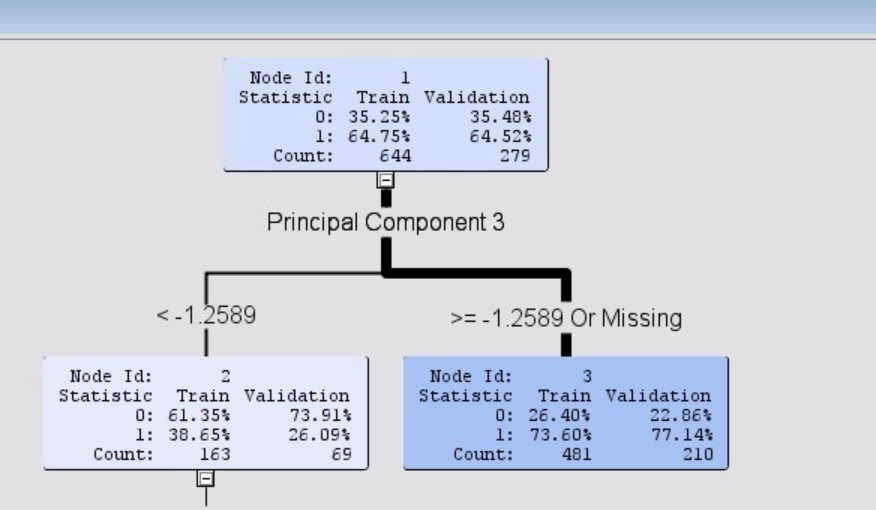
**Fig. 6**

*Screenshot showing the whole Tree result:*



**Fig. 7**

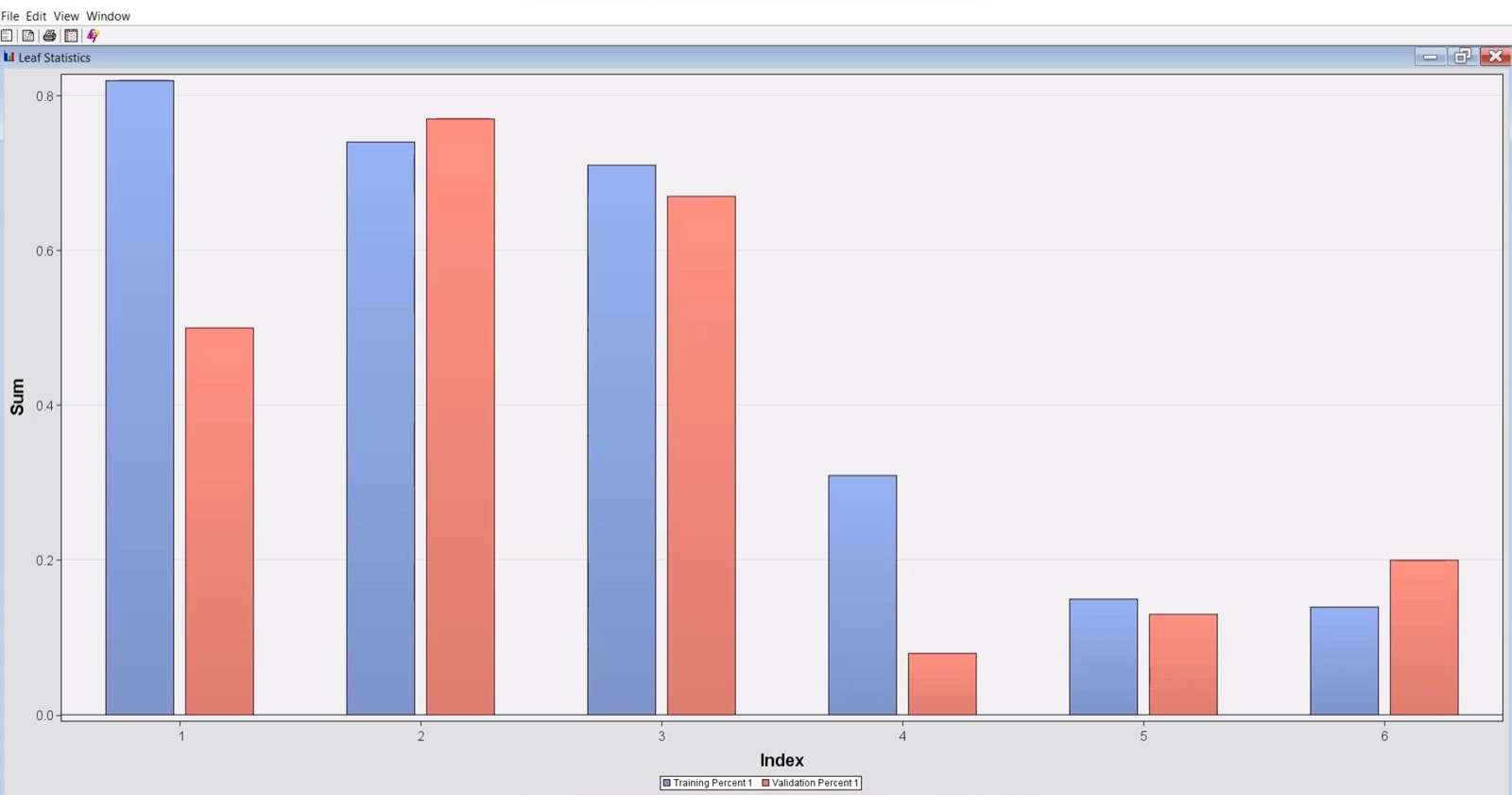
*Screenshot showing the first split of the tree:*



**Fig. 8**

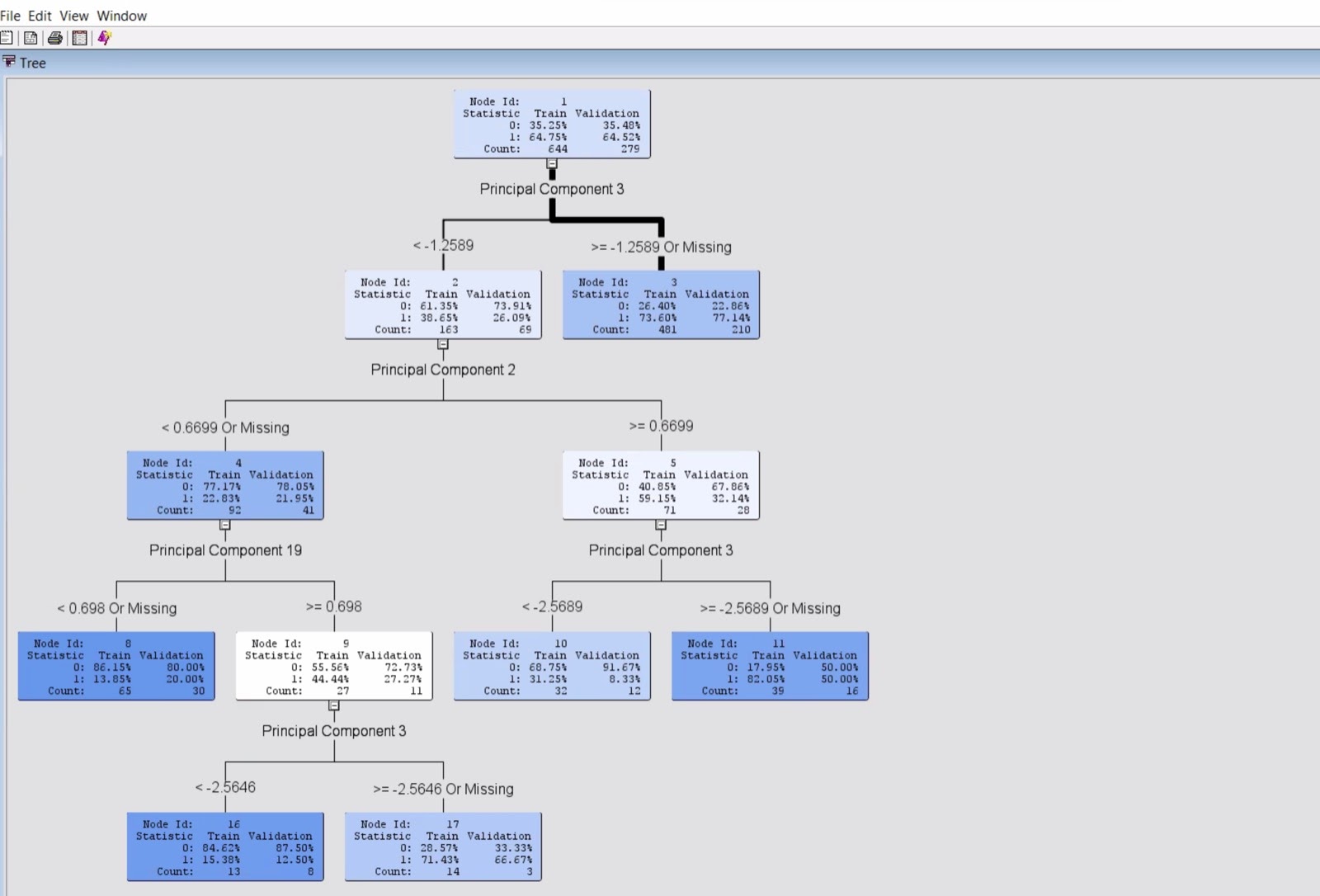
Next, we analysed the data with a Decision tree node by running the Principal Component Analysis and selecting only positively skewed variables. This is represented by the Decision Tree 2 Node in the diagram. We found there are 6 leaves in the optimal tree when we used the Decision Tree node, when we kept the maximum number of branches to two-way splits and then we kept the decision rate as the model assessment statistic. There are four splits for this decision tree. The variable PC3 was used for the first split. For the second split, the variable PC2 was used. For the third split, the variables PC19 and PC3 were used. For the fourth split, PC3 was used.

*Screenshot showing the Leaf Statistics over Sum and Index axis:*

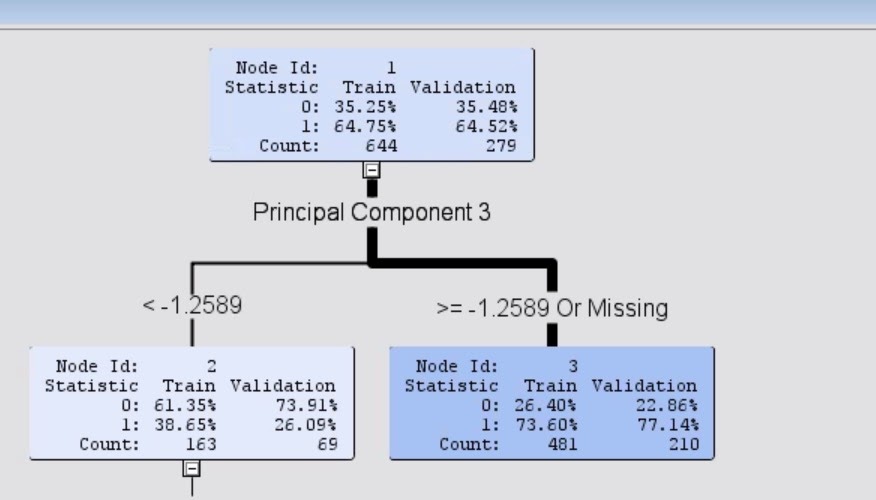


**Fig. 9**

*Screenshot showing the whole Tree result:*

 **Fig. 1**

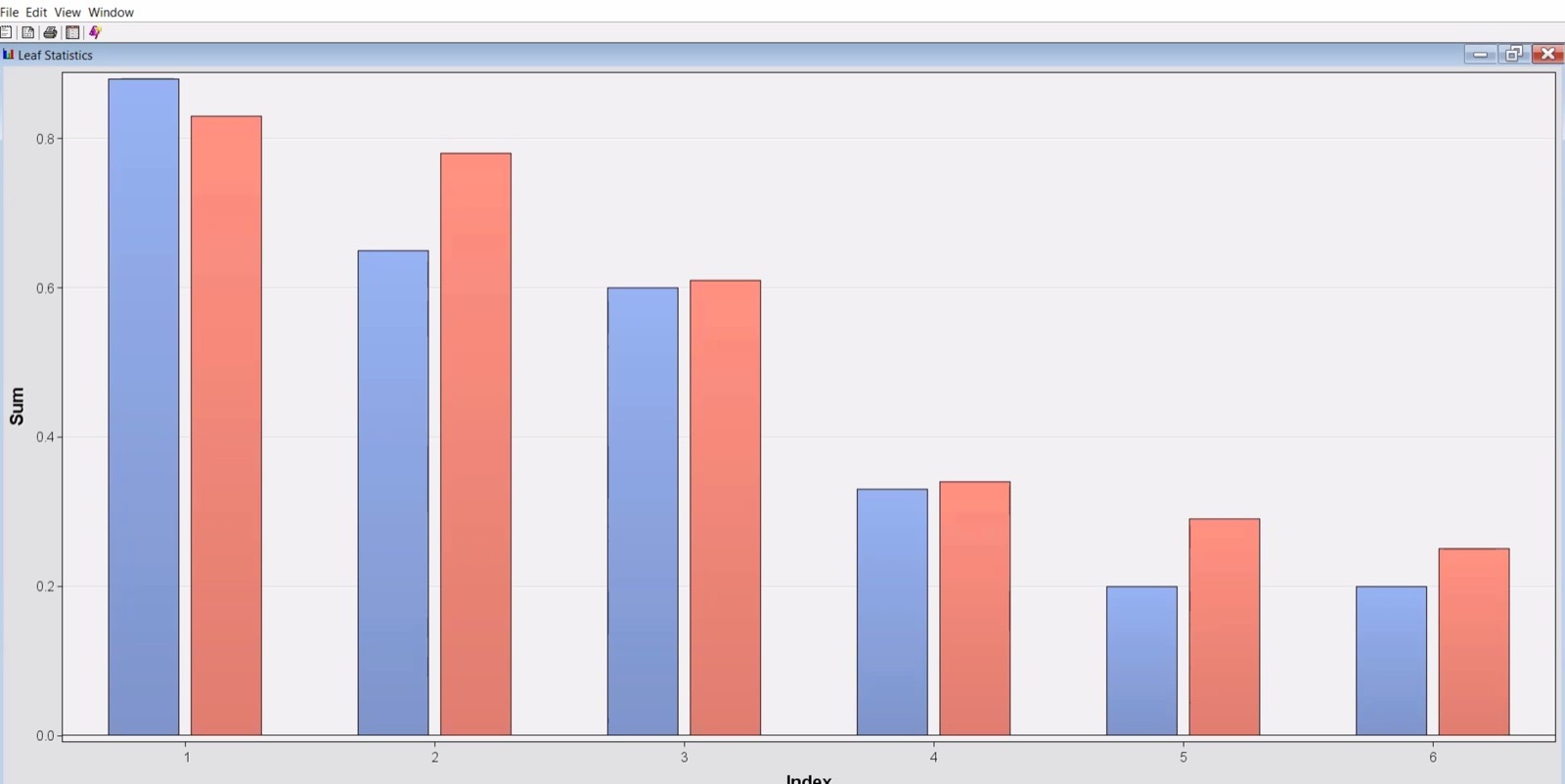
*Screenshot showing the first split of the tree:*



**Fig. 11**

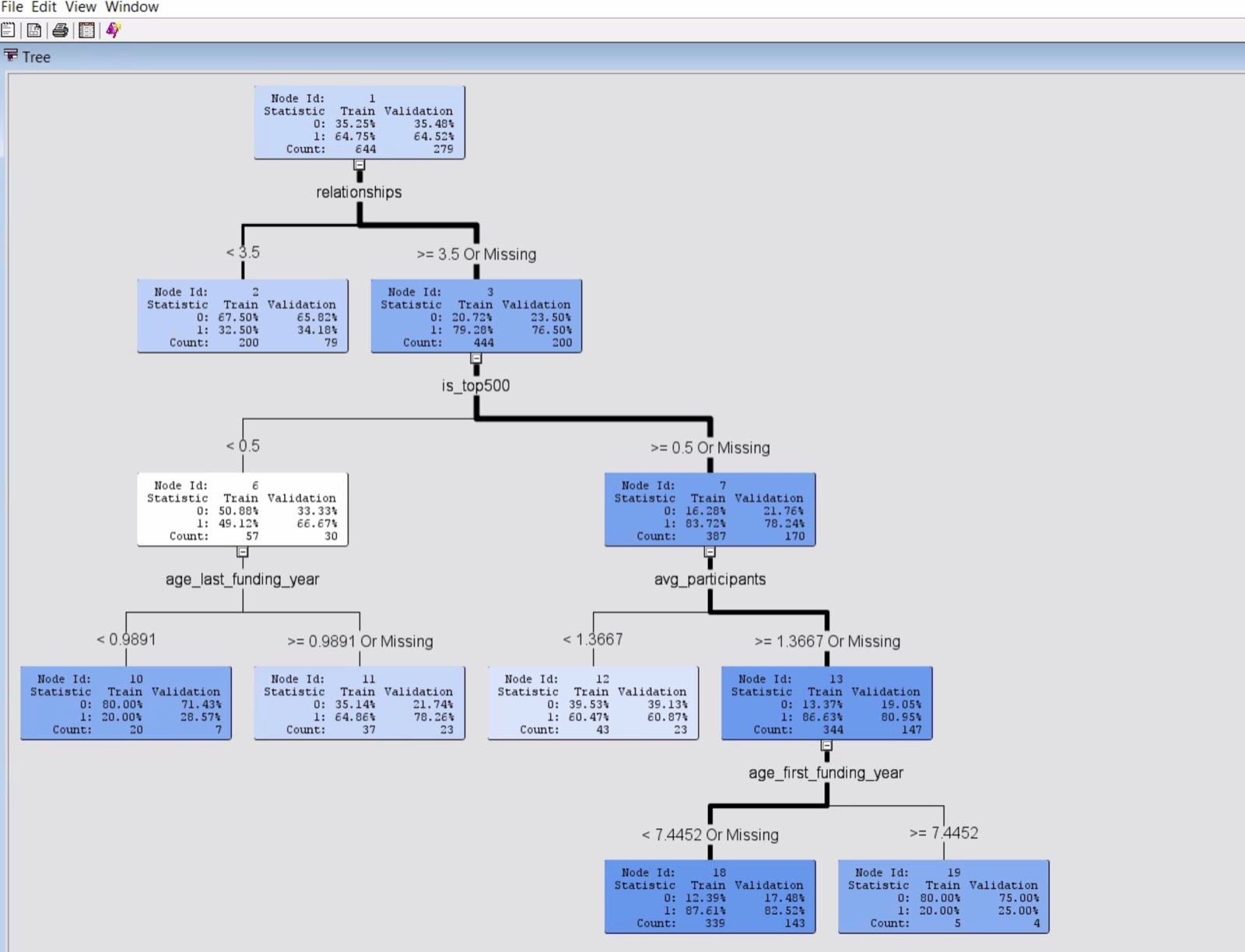
We now analysed the data without running the Principal Component Analysis and selecting all variables. This is represented by the Decision Tree 3 Node in the diagram. We found there are 6 leaves in the optimal tree when we used the Decision Tree node, when we kept the maximum number of branches to two-way splits and when we kept the misclassification rate as the model assessment statistic. There are four splits for this decision tree. The variable relationships were used for the first split. For the second split, the variable is\_top500 was used. For the third split, the variables age\_last\_funding\_year, and avg\_participants were used. For the fourth split, age\_first\_funding\_year was used.

*Screenshot showing the Leaf Statistics over Sum and Index axis:*

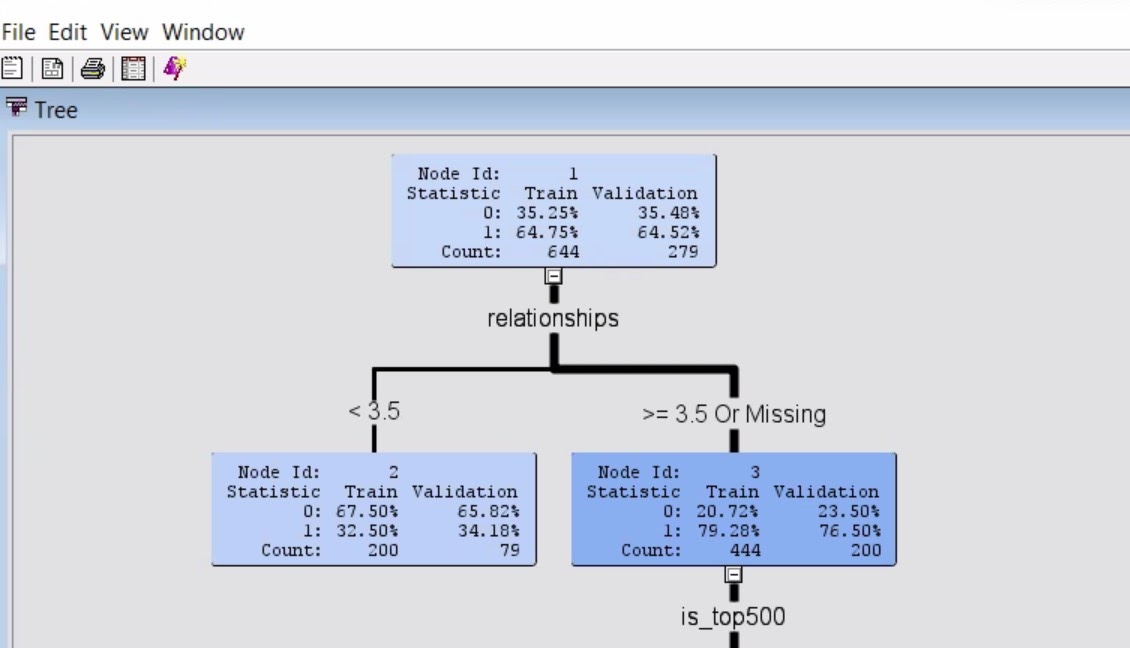


**Fig. 12**

*Screenshot showing the whole Tree result:*

 **Fig. 13**

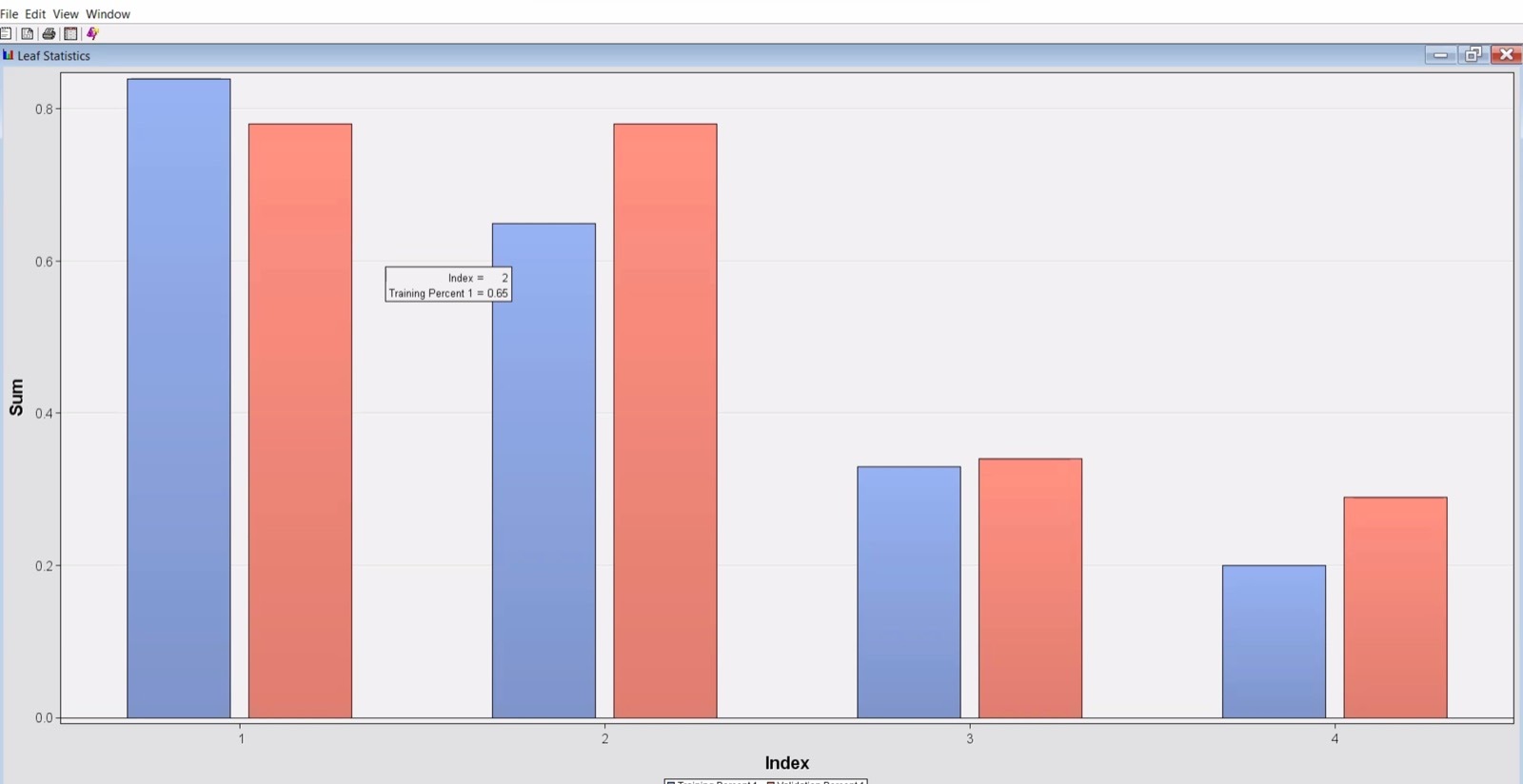
*Screenshot showing the first split of the tree:*

****

**Fig. 14**

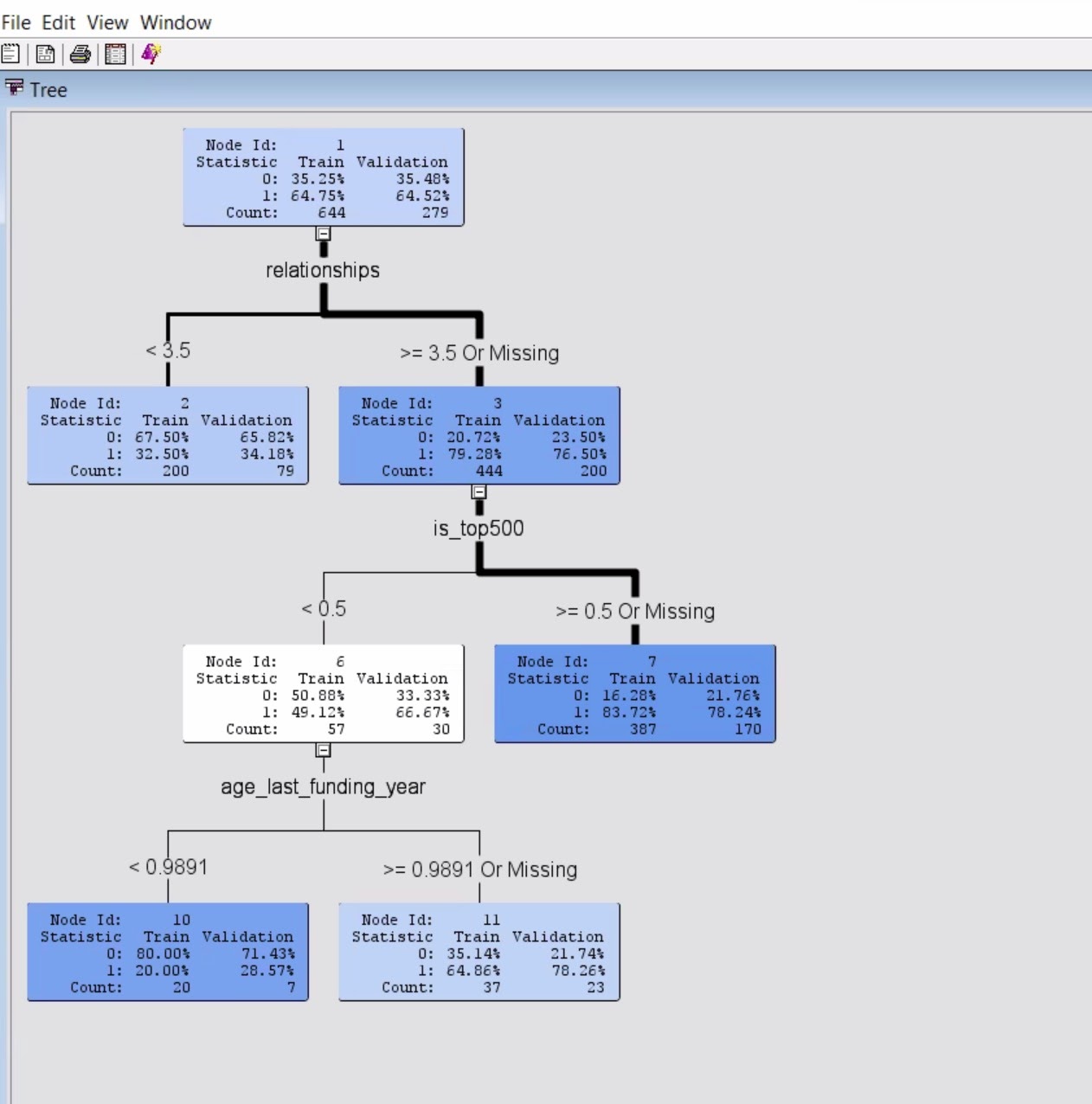
We now analysed the data without running the Principal Component Analysis and selecting only positively skewed variables. This is represented by the Decision Tree 4 Node in the diagram. We found there are 4 leaves in the optimal tree when we used the Decision Tree node, when we kept the maximum number of branches to two-way splits and when we kept the decision rate as the model assessment statistic. There are three splits for this decision tree. The variable relationships were used for the first split. For the second split, the variable is\_top500 was used. For the third split, the variables age\_last\_funding\_year was used.

*Screenshot showing the Leaf Statistics over Sum and Index axis:*



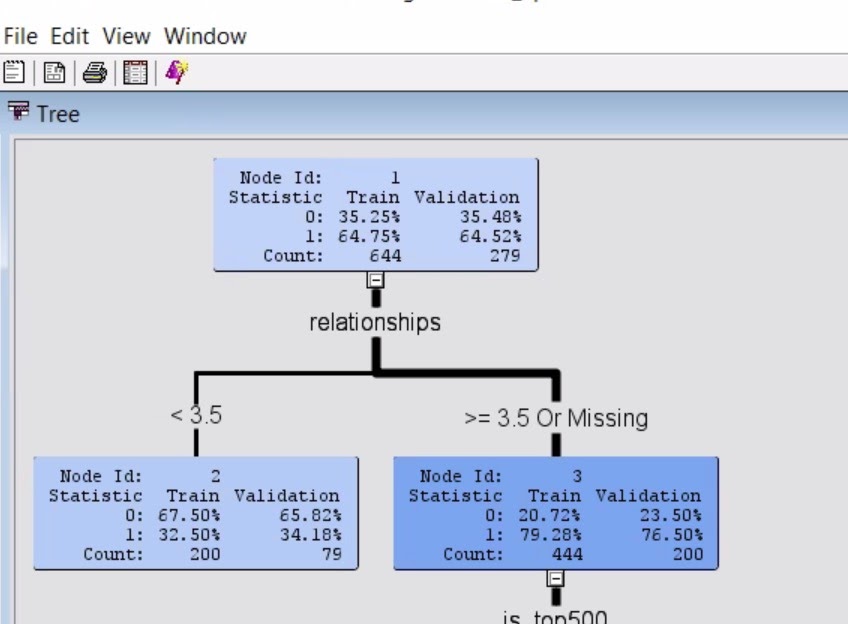
**Fig. 15**

*Screenshot showing the whole Tree result:*



**Fig. 16**

*Screenshot showing the first split of the tree:*



**Fig. 17**

**Auto Neural Model**

We will now use Auto Neural Mode to analyse the data. First, we wanted to analyse the data by running the Principal Component Analysis and selecting all variables. This is represented by the Auto Neural Node in the diagram. We have selected Yes for Tanh and Logistic. We found that in the 6th iteration the node is pure (all observations in the node have almost the same value for the dependent variable), so we stop the iteration here.

*The iteration for misclassification rate:*

*Table 15:*

A picture containing table

Description automatically generated



**Fig. 18**

Now, we used Auto Neural Mode to analyse the data by running the Principal Component Analysis and selecting only positively skewed variables. This is represented by the Auto Neural Node 2 in the diagram. We have selected Yes for Tanh and Logistic. We found that in the 6th iteration the node is pure (all observations in the node have almost the same value for the dependent variable), so we stop the iteration here.

*The iteration plot for misclassification rate:*

*Table 16:*



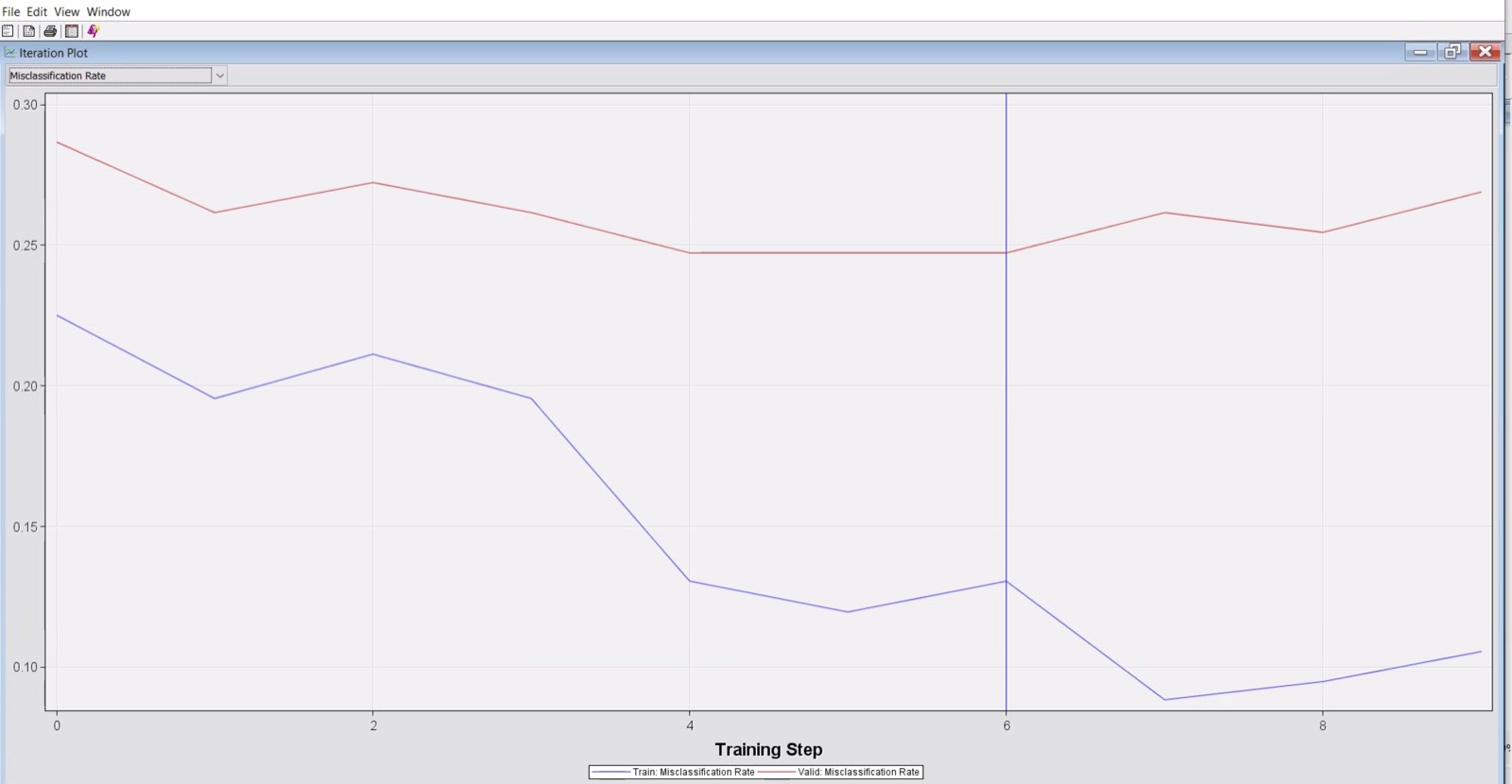
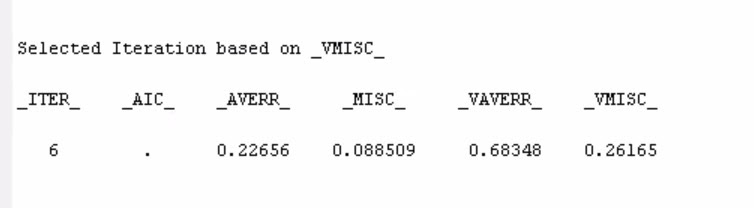


**Fig. 19**

Now, we used Auto Neural Mode to analyse the data by running without the Principal Component Analysis and selecting all variables. This is represented by the Auto Neural Node 3  in the diagram. We have selected Yes for Tanh and Logistic. We found that in the 6th iteration the node is pure (all observations in the node have almost the same value for the dependent variable), so we stop the iteration here.

*The iteration plot for misclassification rate:*

*Table 17:*

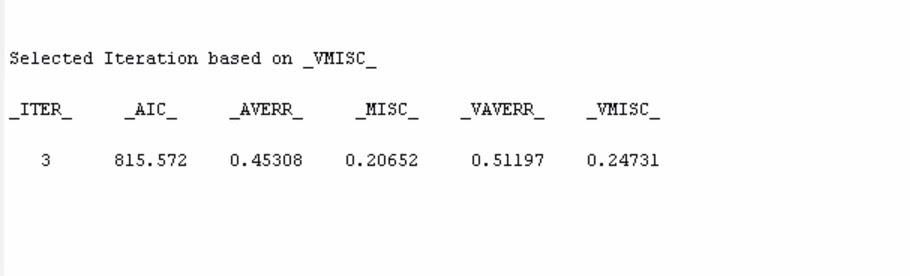


**Fig. 20**

Now, we used Auto Neural Mode to analyse the data by running without the Principal Component Analysis and selecting only positively skewed variables. This is represented by the Auto Neural 4 Node in the diagram. We have selected Yes for Tanh and Logistic. We found that in the 3rd iteration the node is pure (all observations in the node have almost the same value for the dependent variable), so we stop the iteration here.

*The iteration plot for misclassification rate:*

*Table 18:*



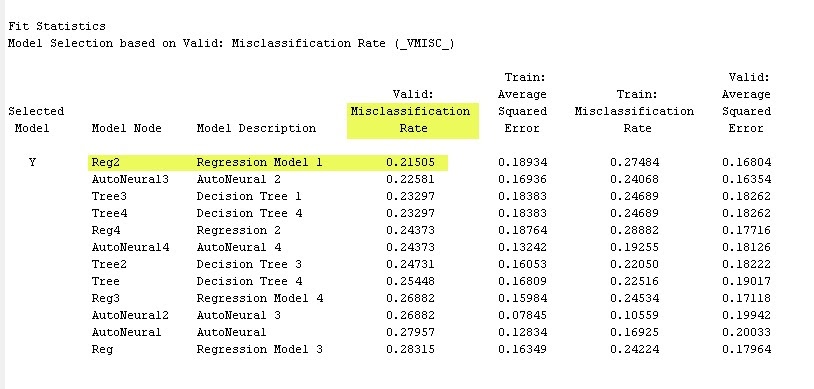


**Fig. 21**

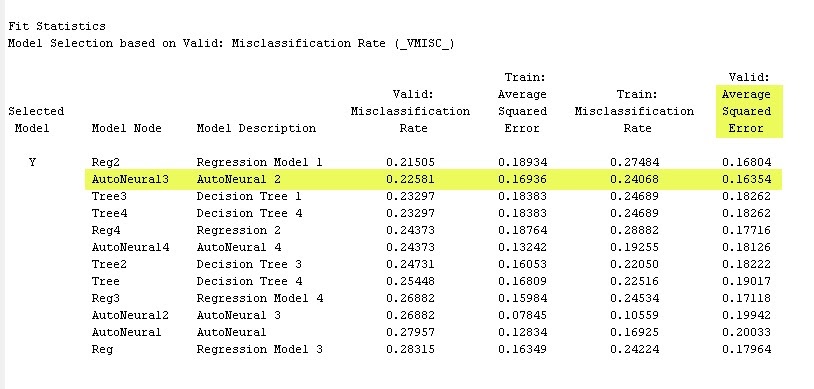
**Result : Model Comparison**

Finally, We want to check which model is performing better among all. To do so, we have attached a Model Comparison node to compare and determine the best model among all models. Based on the Misclassification rate, we found Regression 2 to be the best model as it has the misclassification rate of 0.21505, which is the least of all. Based on Average Square Error, we found Auto Neural 3 Node to be the best model with the least average square error of 0.16354.

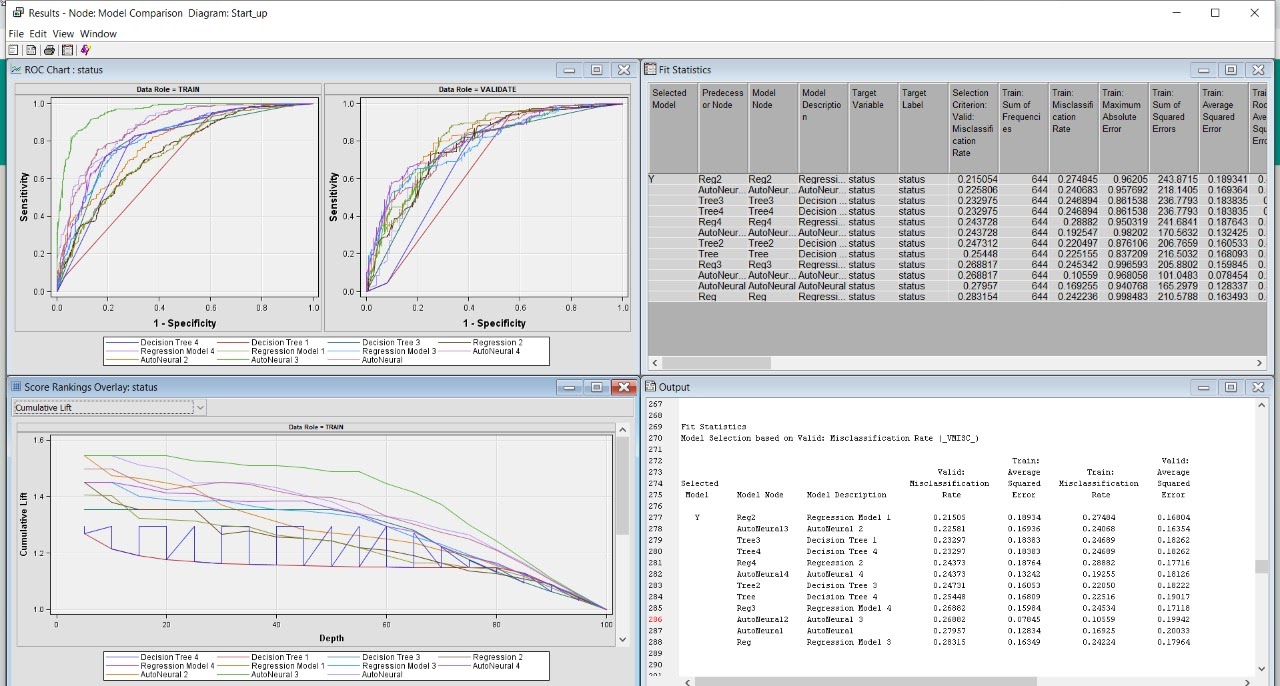
*Table 19: Screenshot showing Misclassification Rate:*



*Table 20: screenshot showing Average Square Error:*



*Screenshot showing overall Output of the Model Comparison Node:*



**Fig. 22**

**Conclusion**

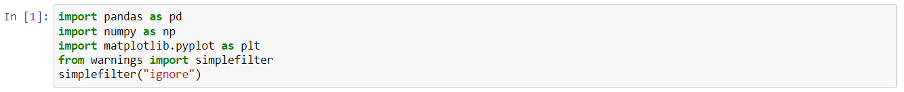
After using the Model Comparison Node, we found that the Regression 2 model has the lowest misclassification rate with the validation dataset among all the models which we used in this project but the Auto Neural 3 network has the least Average Squared Error with the validation dataset among all of the networks we used.

**Alternative work**

We also performed the same project using Python. Firstly, we analyse our data and then we clean it. After cleaning our data, we use some models and analyse which model is best performing.

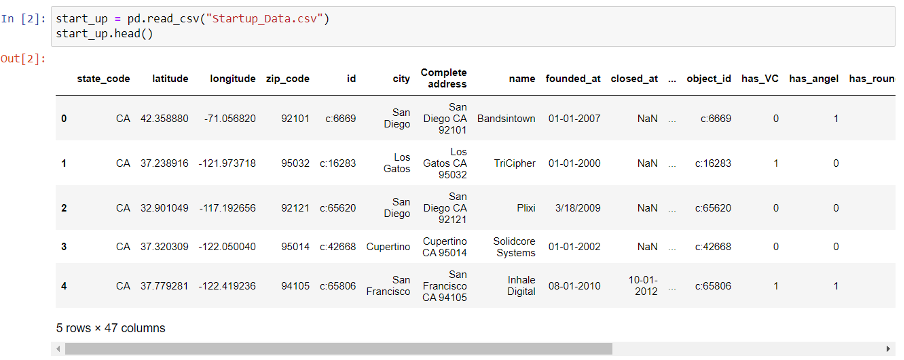
1. Importing Libraries

First part of the python project is to import the libraries. In starting, we are adding some basic libraries like Pandas, Numpy, Matplotlib, and Simplefilter. And we also add libraries in between of the project as the libraries are required.



2. Reading the dataset

Now, we need to load the dataset for the project. We use the read\_csv command to read the csv file. By using this command, we import this dataset into Jupyter Notebook. And also use head() command to see the first 5 rows of the data set.



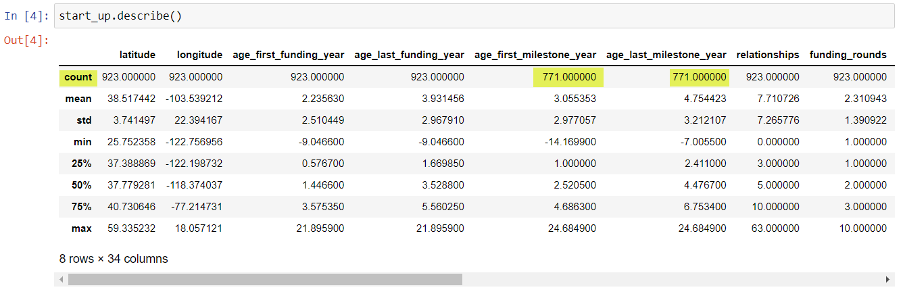
3. Data Analysing

Next and import the job of the project to analyse the data. So, first we need to check the shape of the dataset. We use the shape command for this.



We have 923 number of observations and 47 variables in our data set.

Now, we need to check if there’s any variable that has missing values? For this, we use describe() command to check the counts min, max, mean, std, 23%, 50%, and 75% percentile of the dataset.



After checking this table, we found that in our data set, we have some variables that have missing values. We use the isnull().sum() command which gives the sum of all missing values in each variable.



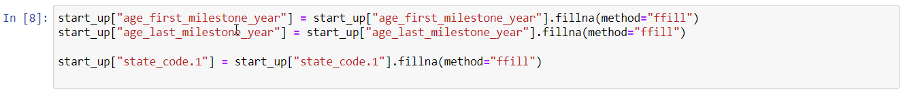
We found that We have 588 missing values in closed\_at, 152 missing values in age\_first\_milestone\_year, 152 missing values in age\_last\_milestone\_year and 1 missing value in state\_code.1 variables. As closed\_at variable has more than 50% null values, so we drop this variable from the data. We also drop the id variable because it doesn’t have useful information.



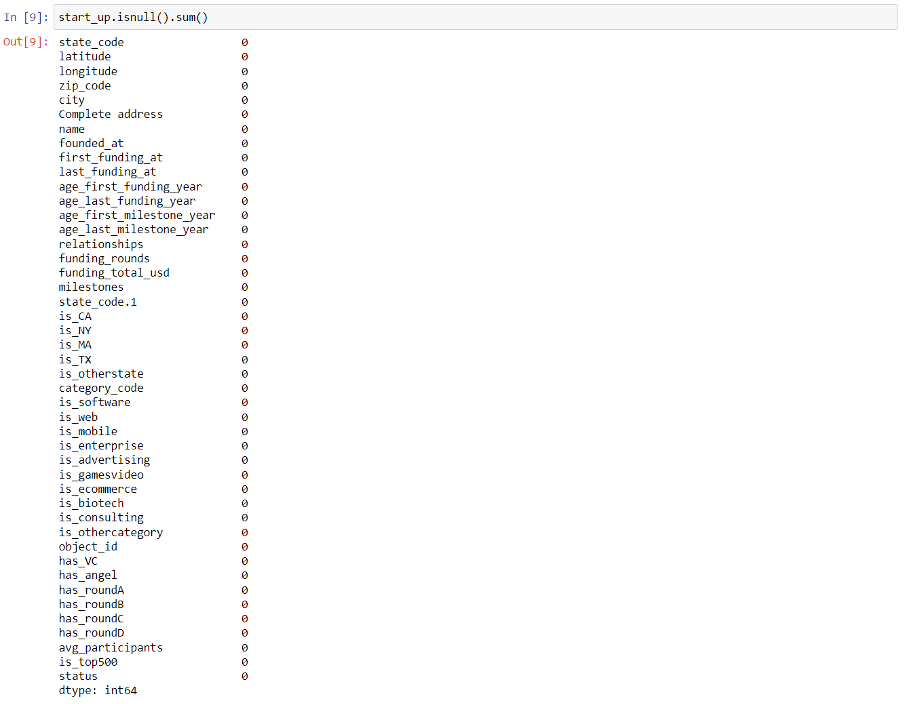
After dropping those two variables, we will check the shape of existing data. And, data has 923 rows and 45 columns.



We also found the variables age\_first\_milestone\_year, age\_last\_milestone\_year and state\_code.1 also have less than 20% missing values. Now, we need to fix these null values issue with this data set. We will fill these null value with 'ffill' command which stands for 'forward fill' and will propagate last valid observation forward.



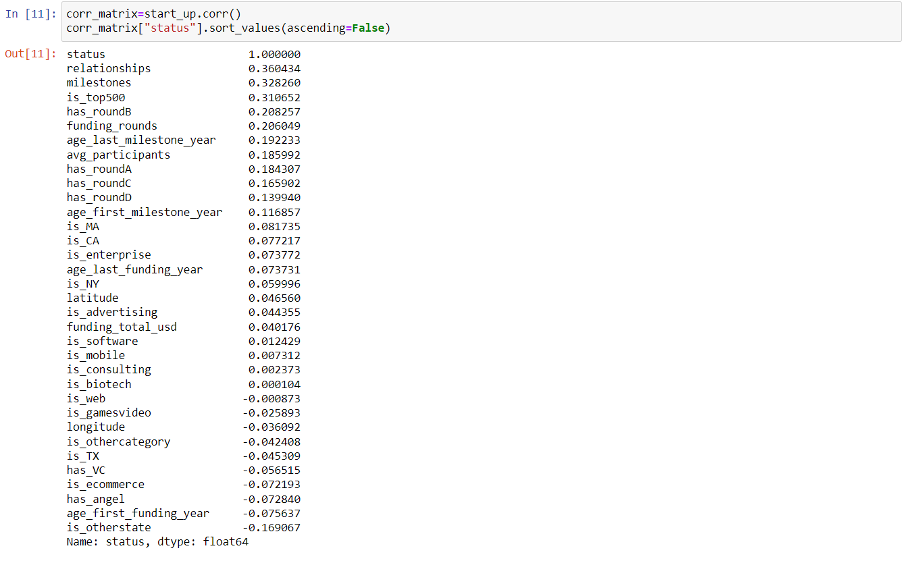
After filling the null values, we will check again null values. Is there any missing value available?



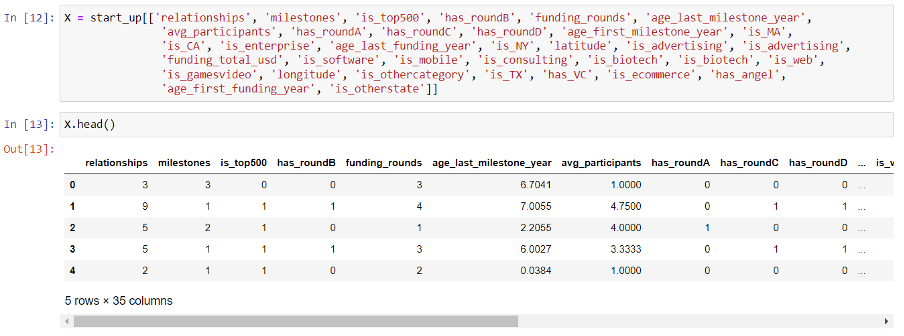
Now, there is no missing any more in the data set. We will also check the information of our data set after fixing the missing values issue.



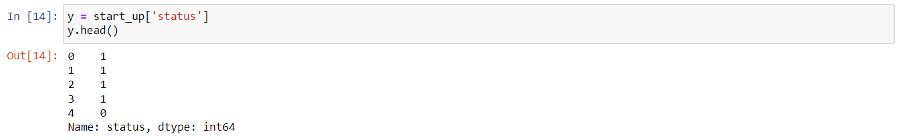
Now, we will check the correlation of the input variables to the target variables.



For training the model, we need a training data set.  We will create this training data using the variables which have correlation with target variable. Here, we have 35 input variables which have good correlation with target variable.

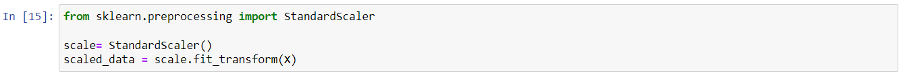


We also need target variable for the model. Target variable is stored in a object call y.



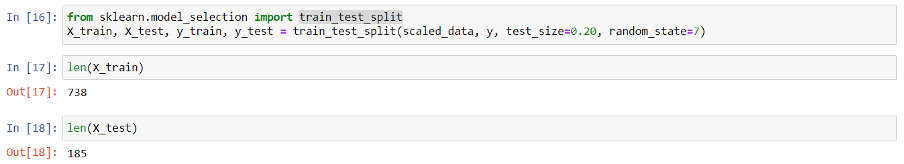
4. Standardizing the Training Data

For better performance of the models, we need to standardized our input training data set. For standardizing the data, we import StandardScaler from sklearn.preprocessing and use fit\_transform(X) command.



5. Train, Test Splitting the data

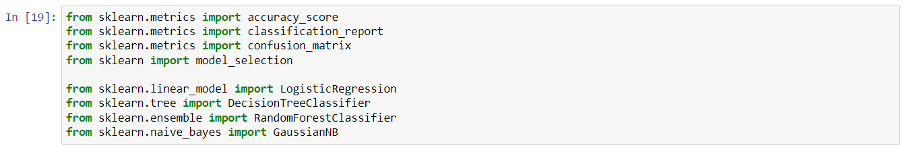
For training the model, we need to split our input data into train and test by train\_test\_split method. Here, we are using 80% data for training and 20% for testing. And, we selected random\_state is 7 for the models. After splitting the data set, we found that we have 738 observations for training the model and 185 number of observations for testing the models.



6. Training Different Algorithms and Comparing them

Here, we are using four different algorithms, Logistic Regression, Decision Tree, Random Forest and Naïve Bayes and comparing their performance and accuracy. On the base of accuracy, we are trying to find which algorithm is performing better for this project.

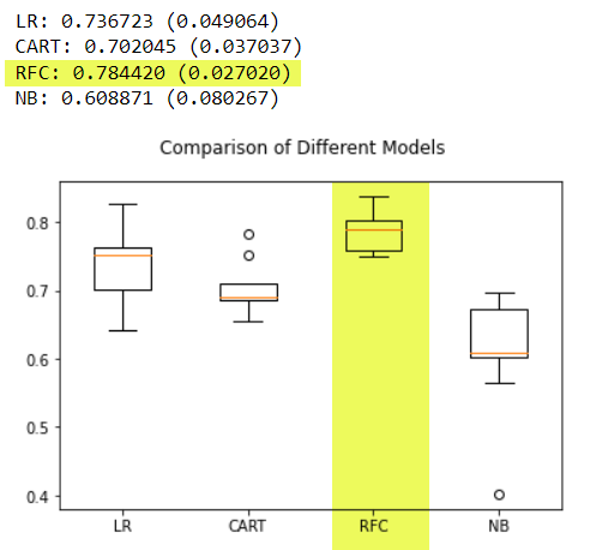
First, we need libraries for performing the each of algorithms. And then we train and compare the accuracy of our algorithms.



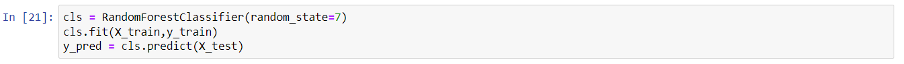


7. Results

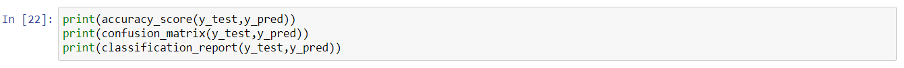
From the results of all algorithms, Random Forest Algorithm is performing better.

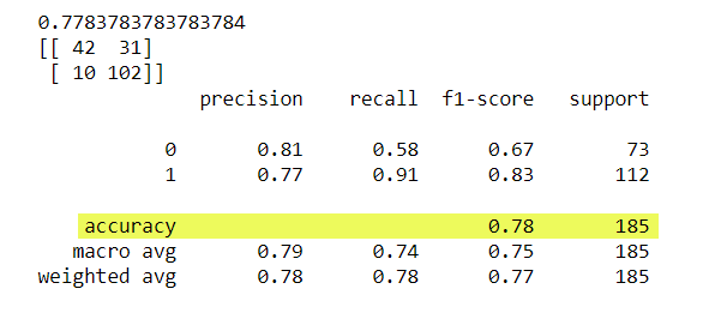


So, we will go with the random forest algorithm for this project and calculate the y predicted value using X\_test value.



Now, we will check the accuracy of the model using y\_test and y\_pred data and we will also create a confusion matrix. Apart from that we generate the classification report also.





Conclusion:

Here, we used four algorithms, Logistic Regression, Decision Tree, Random Forest, Naïve Bayes. Out of four algorithms, Random Forest is performing better. It has 78% accuracy. Acquired (1) the start-up has more f-1 score (0.83) than the f-1 score (0.67) of Closed (0) the start-up but precision of Closed the start-up is more than the Acquired the start-up.

**References**

1. [https://www.kaggle.com/manishkc06/startup-success-prediction](https://www.kaggle.com/manishkc06/startup-success-prediction%0d).
2. <https://documentation.sas.com/doc/en/emref/15.1/titlepage.htm>.
3. <https://analyticsconsultores.com.mx/wp-content/uploads/2019/03/Decision-Trees-for-Analytics-Using-SAS-Enterprise-Miner-B.-de-Ville-P.-Neville-SAS-2013.pdf>.
4. <https://medium.com/hackernoon/logistic-regression-using-sas-enterprise-guide-3ffb7774f765>.
5. <https://bykelly93.wordpress.com/2016/04/28/sas-enterprise-miner-logistic-regression/>.
6. <https://www.mwsug.org/proceedings/2009/stats/MWSUG-2009-D02.pdf>.
7. <https://communities.sas.com/t5/SAS-Data-Mining-and-Machine/Principle-component-analysis-in-Enterprise-Miner/td-p/480335>.
8. <https://www.lexjansen.com/wuss/2007/AnalyticsStatistics/ANL_Matignon_DataMining.pdf>.
9. <https://www.datalab-crm.de/wp-content/uploads/2017/07/sas-enterprise-miner-101369.pdf>.
10. <https://realpython.com/logistic-regression-python/>
11. <https://www.datacamp.com/community/tutorials/decision-tree-classification-python>.
12. <https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
13. <https://www.datacamp.com/community/tutorials/random-forests-classifier-python>
14. <https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/>
15. <https://www.analyticsvidhya.com/blog/2021/01/a-guide-to-the-naive-bayes-algorithm/>