Data Mining Assignment

Name: Abhishek Dubey

Student ID: D20123718

Full time: MSc Data Science 2020-21

Class: TU59 Full Time

Date: 15/12/2020

Definition of problem

Telemarketing campaign data of Portuguese bank institution. By using it we need to identify the customers who are likely to subscribe the term deposit account based on previous marketing campaign.

Here in this project, we will use several machine learning algorithms to identify the best suit for this analysis.

Further that final model will consider as credible and valuable for telemarketing campaign managers.

Final Goal is to predict weather customer will opt term deposit- variable “y” (yes, no)

This will result in selling more term deposit account by Portuguese Bank.

Data Exploration and Descriptive Analytics

** We are using SAS Enterprise Miner for Analysis

Import Data:

Data belongs to marking campaign of Portuguese bank.

We are using SAS Enterprise Miner for Analysis

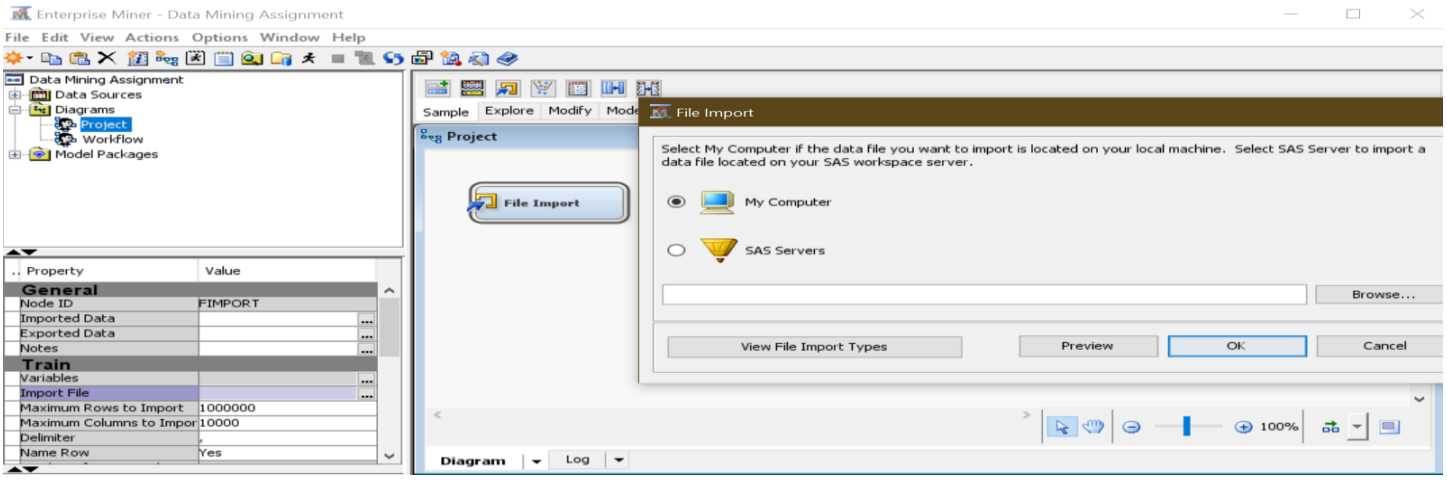
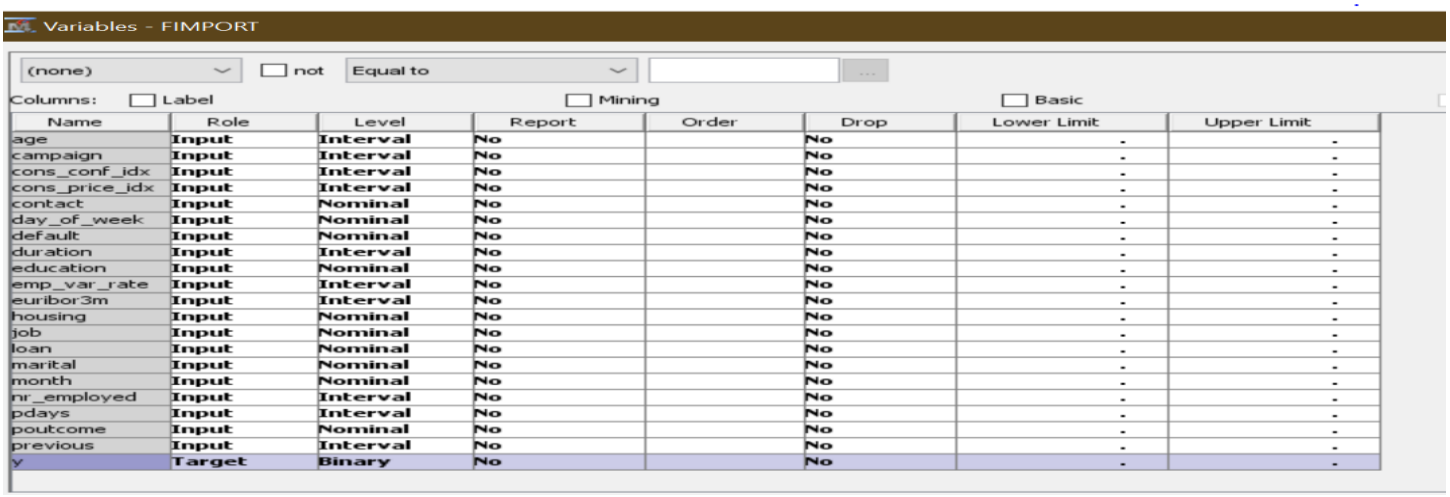


Figure: Represents how we import data in SAS

Data set contains 41188 rows (observation) and 21 columns (variables)

All Variables in Data Set



Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
age	Input	Interval	No		No	.	.
campaign	Input	Interval	No		No	.	.
cons_conf_idx	Input	Interval	No		No	.	.
cons_price_idx	Input	Interval	No		No	.	.
contact	Input	Nominal	No		No	.	.
day_of_week	Input	Nominal	No		No	.	.
default	Input	Nominal	No		No	.	.
duration	Input	Interval	No		No	.	.
education	Input	Nominal	No		No	.	.
emp_var_rate	Input	Interval	No		No	.	.
euribor3m	Input	Interval	No		No	.	.
housing	Input	Nominal	No		No	.	.
job	Input	Nominal	No		No	.	.
loan	Input	Nominal	No		No	.	.
marital	Input	Nominal	No		No	.	.
month	Input	Nominal	No		No	.	.
nr_employed	Input	Interval	No		No	.	.
pdays	Input	Interval	No		No	.	.
poutcome	Input	Nominal	No		No	.	.
previous	Input	Interval	No		No	.	.
y	Target	Binary	No		No	.	.

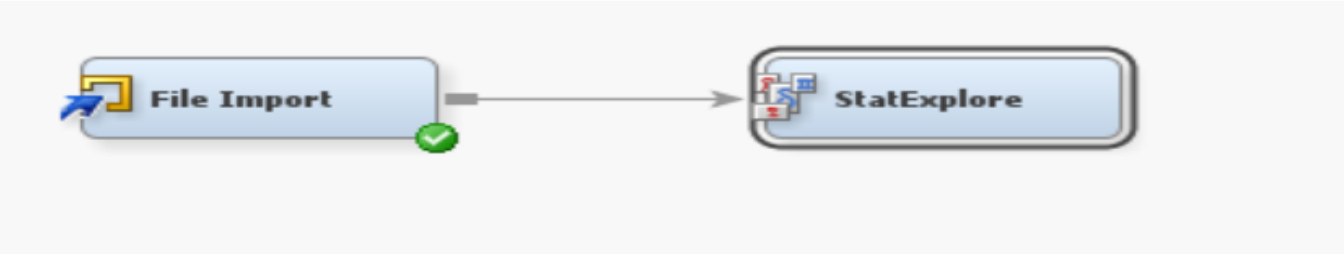
Y: Representing Target variable, Binary – yes, no

YES: represents term deposit account he/she will open

NO: represents term deposit account he/she will not open

We will now explore the data so we use **StatExplore** function from explore tab

When evaluating chi squared statistics for interval variable. Enterprise miner divide them into 5 bins so allowing interval variables to “yes”.



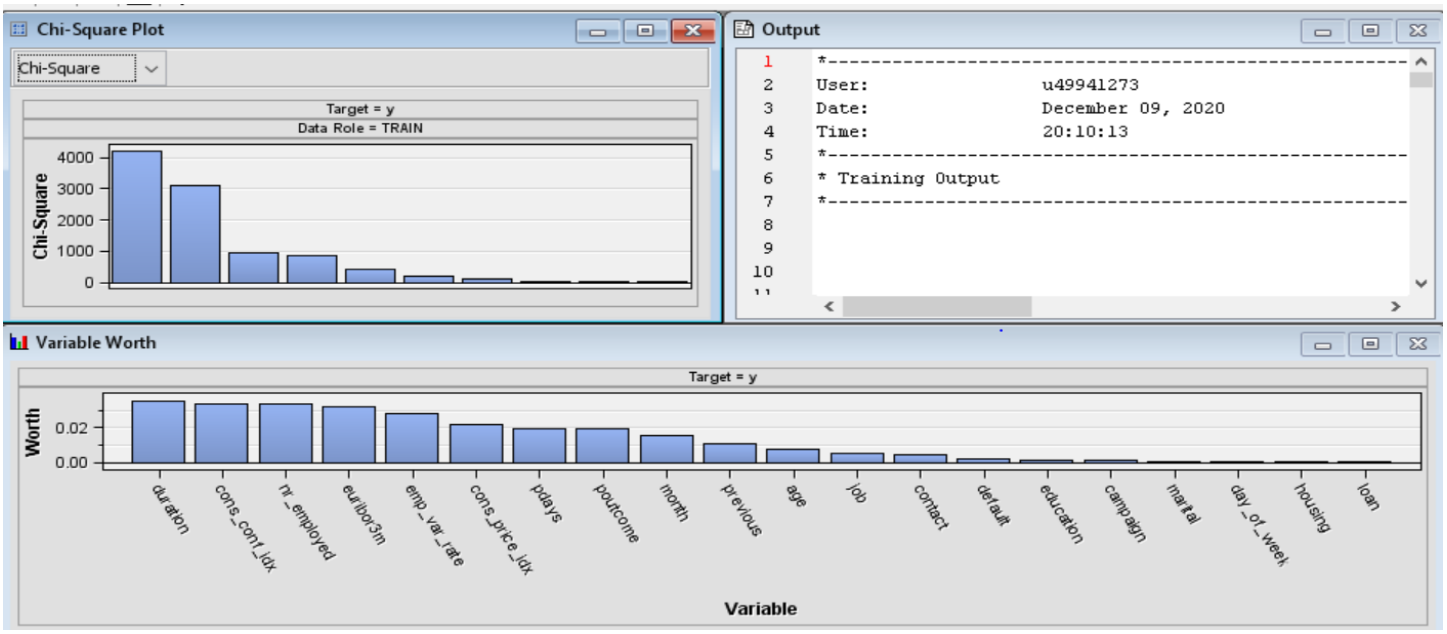


Figure: Represents the Chi-Square as per target variable y

Class Variable Summary Statistics, as we can check there are no missing values in dataset

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	contact	INPUT	2	0	cellular	63.47	telephone	36.53
TRAIN	day_of_week	INPUT	5	0	thu	20.94	mon	20.67
TRAIN	default	INPUT	3	0	no	79.12	unknown	20.87
TRAIN	education	INPUT	8	0	university.degree	29.54	high.school	23.10
TRAIN	housing	INPUT	3	0	yes	52.38	no	45.21
TRAIN	job	INPUT	12	0	admin.	25.30	blue-collar	22.47
TRAIN	loan	INPUT	3	0	no	82.43	yes	15.17
TRAIN	marital	INPUT	4	0	married	60.52	single	28.09
TRAIN	month	INPUT	10	0	may	33.43	jul	17.42
TRAIN	poutcome	INPUT	3	0	nonexistent	86.34	failure	10.32
TRAIN	y	TARGET	2	0	no	88.73	yes	11.27

Interval Variable Summary Statistics

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness
age	INPUT	40.02406	10.42125	41188	0	17	38	98	0.784697
campaign	INPUT	2.567593	2.770014	41188	0	1	2	56	4.762507
cons_conf_idx	INPUT	-40.5026	4.628198	41188	0	-50.8	-41.8	-26.9	0.30318
cons_price_idx	INPUT	93.57566	0.57884	41188	0	92.201	93.749	94.767	-0.23089
duration	INPUT	258.285	259.2792	41188	0	0	180	4918	3.263141
emp_var_rate	INPUT	0.081886	1.57096	41188	0	-3.4	1.1	1.4	-0.7241
euribor3m	INPUT	3.621291	1.734447	41188	0	0.634	4.857	5.045	-0.70919
nr_employed	INPUT	5167.036	72.25153	41188	0	4963.6	5191	5228.1	-1.04426
pdays	INPUT	962.4755	186.9109	41188	0	0	999	999	-4.92219
previous	INPUT	0.172963	0.494901	41188	0	0	0	7	3.832042

As we can check there is **no missing values** present in data set.

Standard Deviation shows high more than 100 for values **duration and pdays**

Duration: duration of call-in seconds, this directly affect the variable Y. If duration is 0 then Y (target variable is also 0)

Pdays: number of days there was no contact after previous campaign. Generally, dataset contain pdays for last 30 days and if person not contacted before than data value is 999.

Which is the reason why mean of pdays are very high.

In transformation of data, we will explore and reduce the variance in those variables with large deviation.

Identification of data insights from previous step

To explore graphs in SAS Enterprise miner I am using graph explore function for analysis.

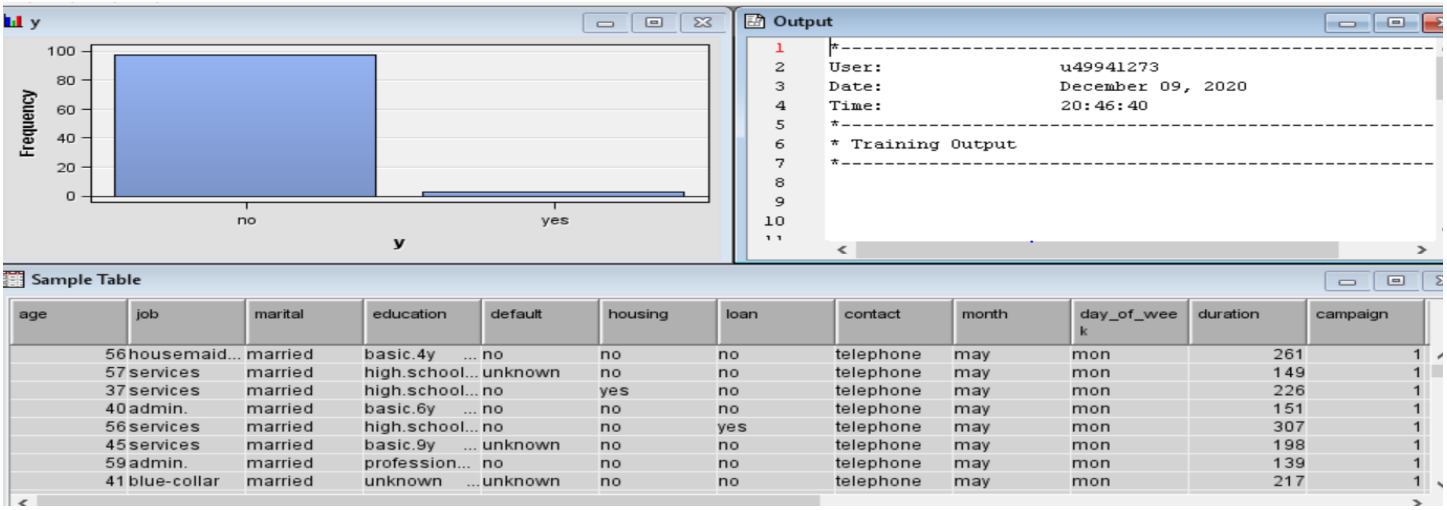
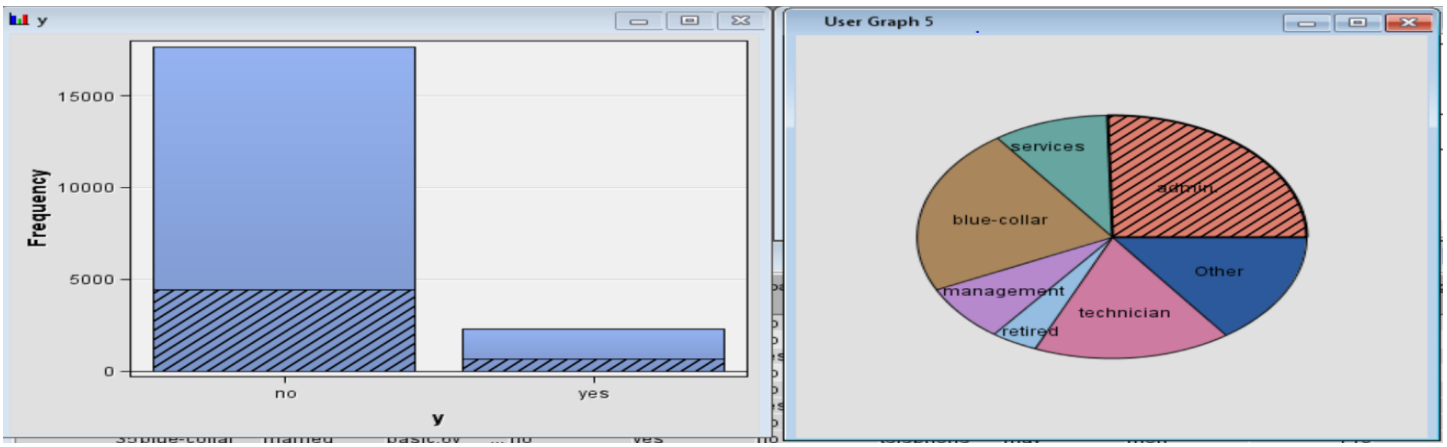


Figure: Represents the frequency of y in 100 random observations

On left: Below Graph represents the frequency in bar chart of 20000 random observations of target “y” variable

On right: Below graph represents the pie chart of 20000 random observations of job variable.

Below insight shows that most of the admin job and most of blue-collar jobs went for “No” means don’t want to opt for term deposit, however the admin and blue-collar observations are more in number comparable to other in variable job.



Small part in Python for better cross analysis

```
pd.crosstab(df.job,df.y).plot(kind='bar')
plt.title('Frequency reported as per jobs')
plt.xlabel('Job')
plt.ylabel('Frequency')
```

```
Text(0, 0.5, 'Frequency')
```

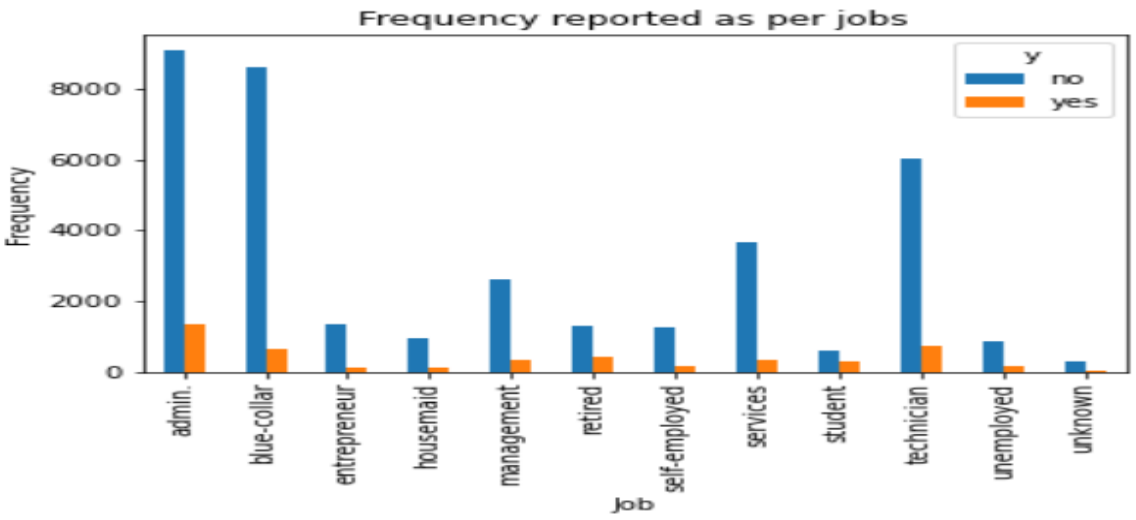


Figure: Represents the frequency of response in y variable as per jobs.

Details of any additional data preparation (cleaning, transformations, etc), data enrichment, feature engineering, feature reduction, etc

** This Step is taken under Python small part and rest done in SAS

Data Set description

```
df.describe()
```

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000
mean	40.02406	258.285010	2.567593	962.475454	0.172963	0.081886	93.575664	-40.502600	3.621291	5167.035911
std	10.42125	259.279249	2.770014	186.910907	0.494901	1.570960	0.578840	4.628198	1.734447	72.251528
min	17.00000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000	4963.600000
25%	32.00000	102.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.344000	5099.100000
50%	38.00000	180.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	5191.000000
75%	47.00000	319.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	5228.100000
max	98.00000	4918.000000	56.000000	999.000000	7.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000

As per analysis we can see that education variable has 8 unique observations.

So, it's better to group all basic studies together

So, for that I am grouping basic.4y, basic.6y, basic.9y to "basic"

```
df['education'].unique()
```

```
array(['basic.4y', 'high.school', 'basic.6y', 'basic.9y',
       'professional.course', 'unknown', 'university.degree',
       'illiterate'], dtype=object)
```

```
df['education']=np.where(df['education'] =='basic.9y', 'Basic', df['education'])
df['education']=np.where(df['education'] =='basic.6y', 'Basic', df['education'])
df['education']=np.where(df['education'] =='basic.4y', 'Basic', df['education'])
```

```
df['education'].unique()
```

```
array(['Basic', 'high.school', 'professional.course', 'unknown',
       'university.degree', 'illiterate'], dtype=object)
```

After grouping now, we have only 6 unique observations which are good for modelling

Other Non-Parametric variable is “default”

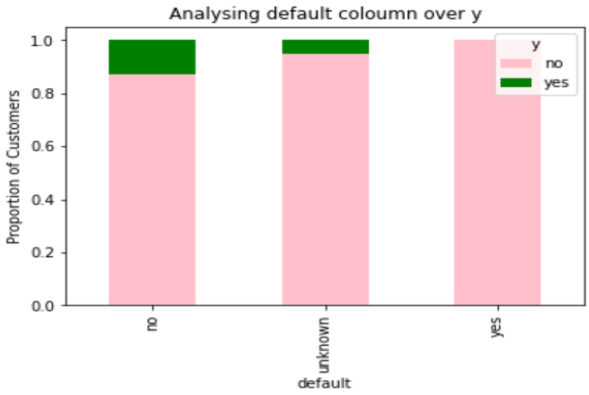
Variable contains so many unknown values which are of no use for our predicting modelling of decision tree.

As per analysis of proportion for customers in default, “unknown” values are more “no” observations of “y” variable.

Most important thing is “yes” observation of “default” variable shows 100% values of “no” observation of variable “y”. **Which means people have credit in default not going for further term deposit account in bank**

```
In [28]: table=pd.crosstab(df.default,df.y)
         table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True, color = ["pink","green"])
         plt.title('Analysing default coloumn over y')
         plt.xlabel('default')
         plt.ylabel('Proportion of Customers')
```

```
Out[28]: Text(0, 0.5, 'Proportion of Customers')
```



So, we are removing that variable

df1 = df.drop(['default'], axis=1)

df1.to_csv("bank.csv")

Checking value counts for target variable “y”

df["y"].value_counts()

no 36548
yes 4640
Name: y, dtype: int64

Means of parametric variables in dataset with respect to “y”

```
df.groupby('y').mean()
```

y	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
no	39.911185	220.844807	2.633085	984.113878	0.132374	0.248875	93.603757	-40.593097	3.811491	5176.166600
yes	40.913147	553.191164	2.051724	792.035560	0.492672	-1.233448	93.354386	-39.789784	2.123135	5095.115991

Data Partition

Training data is used for model fitting

Validation Data is used to test the model without overfitting the data

Dataset is divided into 60%,40% percent randomly

60% Training Data

40% Validation Data

Partition Summary

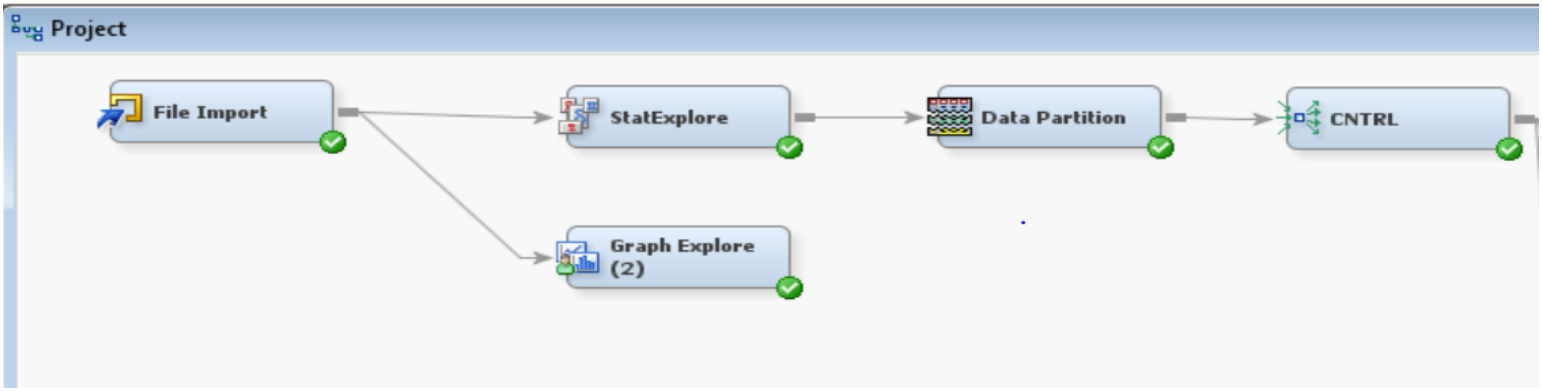
Type	Data Set	Number of Observations
DATA	EMWS2.Stat_TRAIN	41188
TRAIN	EMWS2.Part_TRAIN	24712
VALIDATE	EMWS2.Part_VALIDATE	16476

Control Point Node:

It simplifies a process flow of our diagram by minimizing the connections between multiple data sources and multiple flows.

It works like a connector between all mining models and the data source.

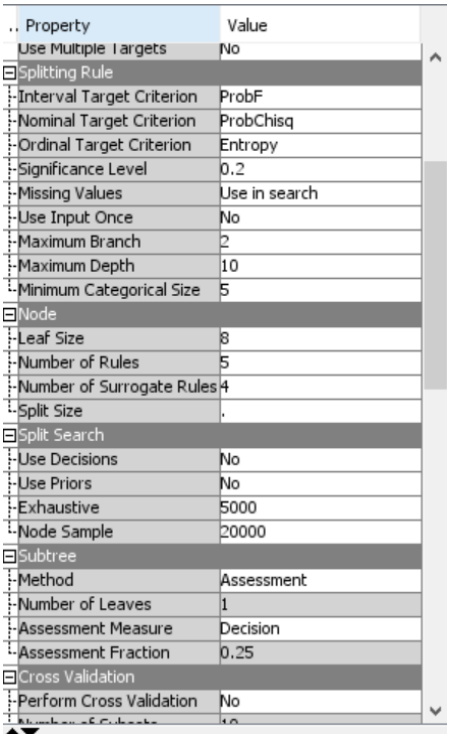
Making manual connections are very difficult to read the flow that's why we are using control point.



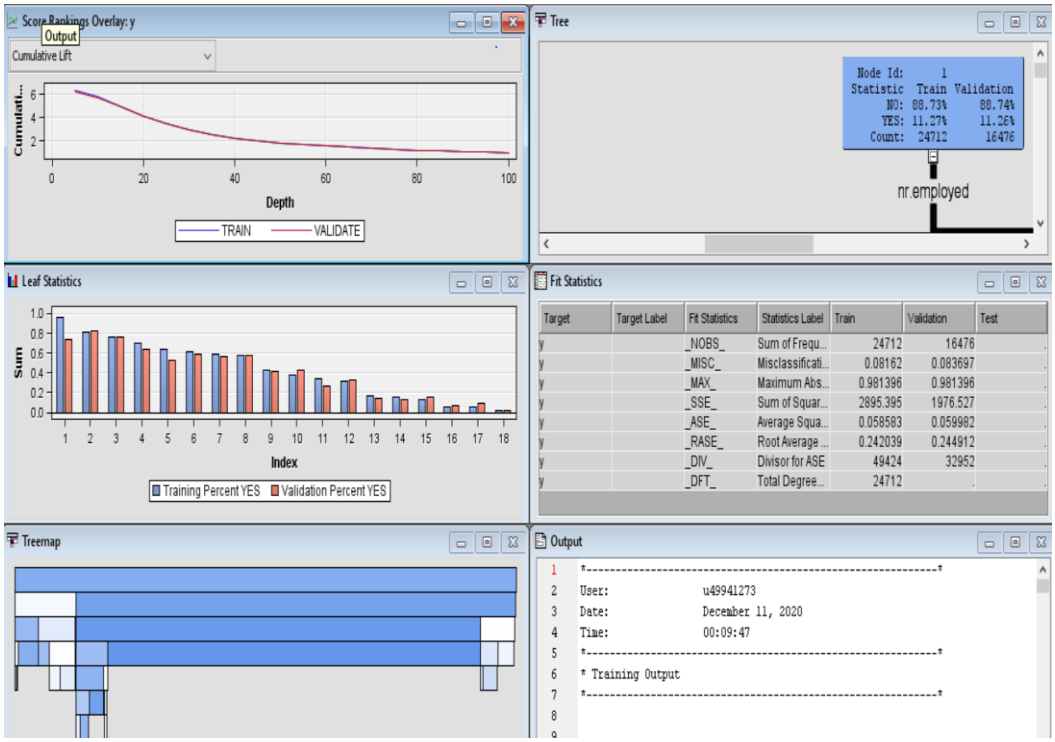
Developing Models

Decision Tree

Automotive Decision Tree (Self Pruning)



Property	Value
Use Multiple Targets	No
Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	10
Minimum Categorical Size	5
Node	
Leaf Size	8
Number of Rules	5
Number of Surrogate Rules	4
Split Size	.
Split Search	
Use Decisions	No
Use Priors	No
Exhaustive	5000
Node Sample	20000
Subtree	
Method	Assessment
Number of Leaves	1
Assessment Measure	Decision
Assessment Fraction	0.25
Cross Validation	
Perform Cross Validation	No



For above properties I have set maximum depth to 10 so that tree will go to 10 depths by split rules.

Node Size is set to 8 so that it should be minimum size of terminal node.

We used automatically trained decision tree by using split rules that maximize the logworth value

Fit Statistics for Decision Tree.

As you can see root average squared error is 0.24 and Maximum Absolute Error is 0.98

Results - Node: Automative Decision Tree (Self Pruning) Diagram: Project

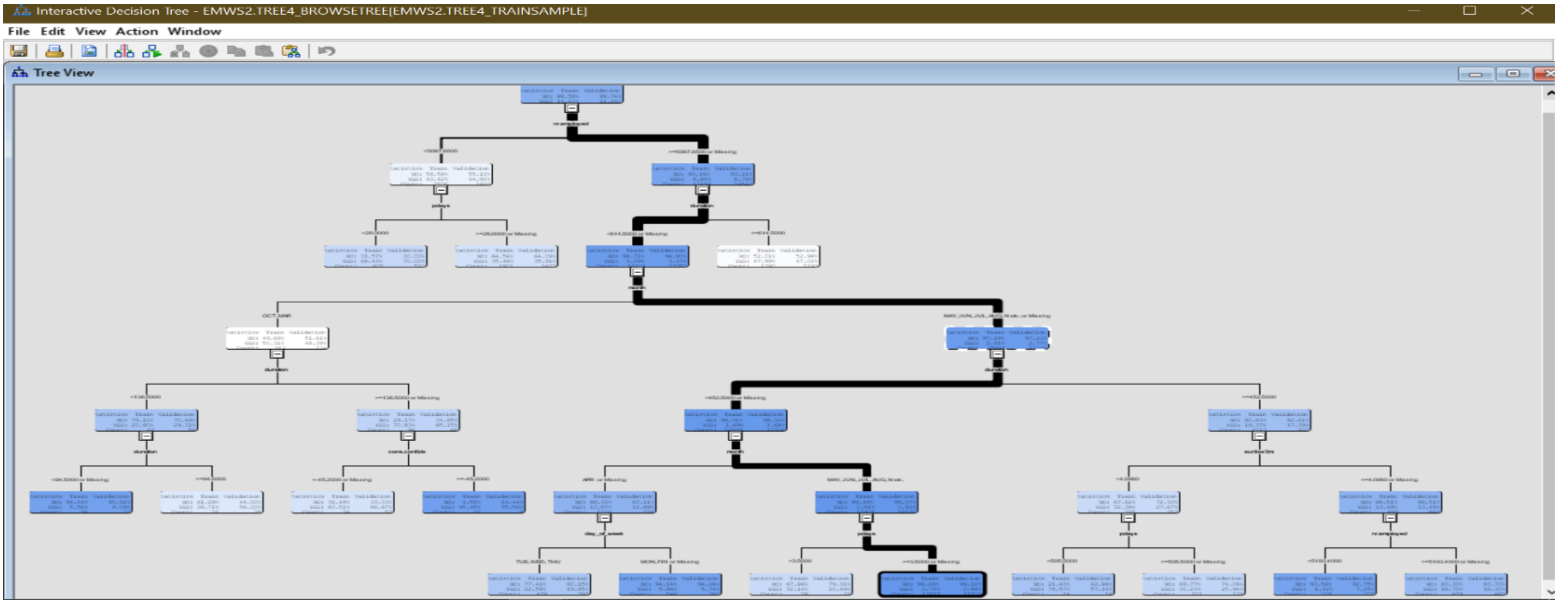
File Edit View Window

Fit Statistics

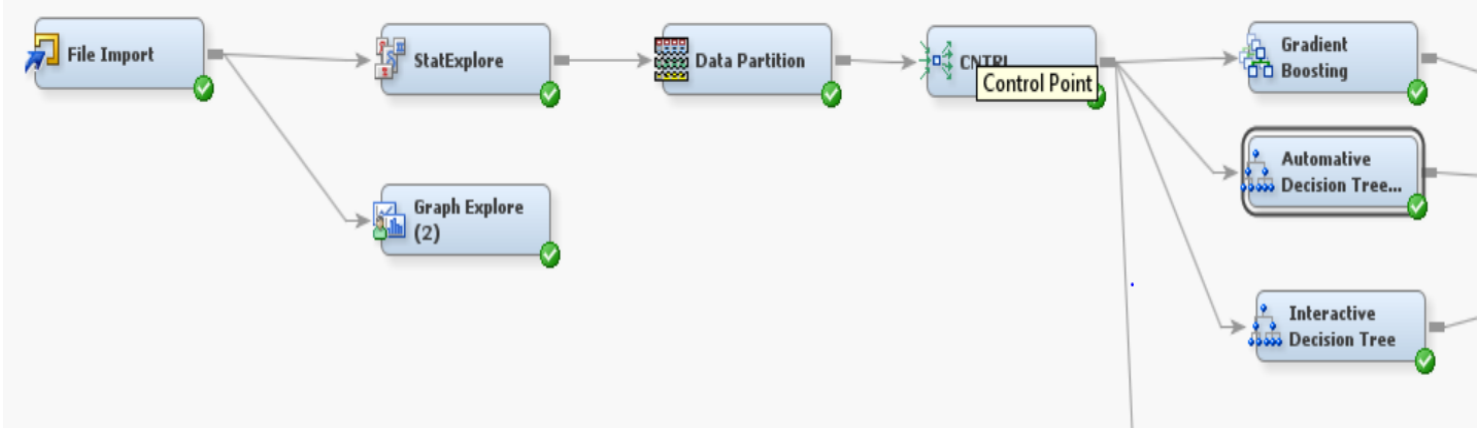
Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
y		_NOBS_	Sum of Frequencies	24712	16476	.
y		_MISC_	Misclassification Rate	0.08162	0.083697	.
y		_MAX_	Maximum Absolute Error	0.981396	0.981396	.
y		_SSE_	Sum of Squared Errors	2895.395	1976.527	.
y		_ASE_	Average Squared Error	0.058583	0.059982	.
y		_RASE_	Root Average Squared Error	0.242039	0.244912	.
y		_DIV_	Divisor for ASE	49424	32952	.
y		_DFT_	Total Degrees of Freedom	24712	.	.

Interactive decision tree

Now we make new interactive decision tree by selecting best candidate split rules by watching out logworth and splitting size from some variables.



Now we make new interactive decision tree by selecting best candidate split rules by watching out Log worth and splitting size from some variables.



Split Node 1

Target Variable: y

Variable	Variable Description	-Log(p)	Branches
nr_employed	nr.employed	629.9864	2
duration	duration	615.5727	2
euribor3m	euribor3m	566.325	2
pdays	pdays	441.7213	2
poutcome	poutcome	409.943	2
emp_var_rate	emp.var.rate	369.3923	2
cons_conf_idx	cons.conf.idx	275.1415	2
month	month	272.2714	2
cons_price_idx	cons.price.idx	218.0924	2
previous	previous	174.2901	2
age	age	109.7713	2
contact	contact	91.6323	2
REP_job	Replacement: job	65.9484	2
campaign	campaign	13.2028	2
REP_education	Replacement: education	11.7884	2
marital	marital	10.0434	2
day_of_week	day_of_week	1.4319	2

Edit Rule... OK Cancel Apply Refresh

Pdays : number of days past since last contact in previous campaign

And 999 represents that customer never contacted and because of this value the standard deviation is high and mean is also high.

We need to rectify this problem by removing it.

We will use split point as 28

<28 means customers are contacted in last 30 days

>28 means they never contacted

Pdays has values from 0 to 27 and 999

pdays - Interval Split Rule

Target Variable: y

Assign missing values to

A specific branch 2

A separate missing values branch

All branches

Branches

Branch		Split Point
1	<	28.0000
2	>=	28.0000

New split point: Add Branch Remove Branch

OK Cancel Apply Reset

Count: 20000 16476

Split Node 3

Target Variable: y

Variable	Variable Description	-Log(p)	Branches
duration	duration	100.8566	2
pdays	pdays	44.5852	2
poutcome	poutcome	41.8283	2
emp_var_rate	emp.var.rate	9.3637	2
nr_employed	nr.employed	9.3637	2
contact	contact	8.9474	2
cons_price_idx	cons.price.idx	8.9166	2
previous	previous	8.8504	2
euribor3m	euribor3m	7.1303	2
cons_conf_idx	cons.conf.idx	4.5609	2
month	month	4.2782	2
day_of_week	day_of_week	2.797	2
REP_loan	Replacement: loan	0.1455	2
age	age	0.0	2
campaign	campaign	0.0	2
REP_education	Replacement: education	0.0	2
REP_housing	Replacement: housing	0.0	2

Edit Rule... OK Cancel Apply Refresh

Fit Statistics for Interactive Decision Tree

Results - Node: Interactive Decision Tree Diagram: Project

File Edit View Window

Fit Statistics

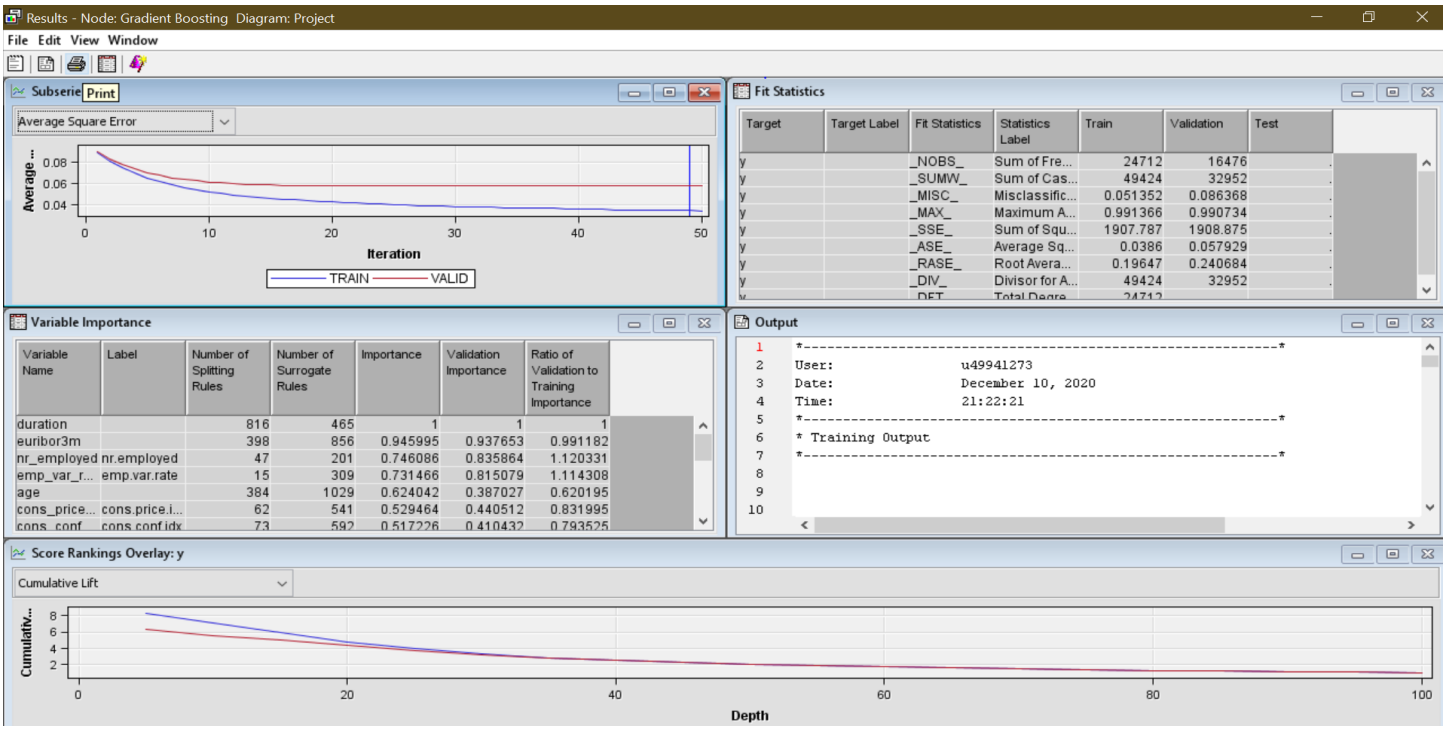
Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
y		_NOBS_	Sum of Frequencies	24712	16476	
y		_MISC_	Misclassification Rate	0.098535	0.098932	
y		_MAX_	Maximum Absolute Error	0.992675	0.992675	
y		_SSE_	Sum of Squared Errors	3183.954	2147.716	
y		_ASE_	Average Squared Error	0.064421	0.065177	
y		_RASE_	Root Average Squared Error	0.253813	0.255298	
y		_DIV_	Divisor for ASE	49424	32952	
y		_DFT_	Total Degrees of Freedom	24712		

As you can see root average squared error is 0.25 and Maximum Absolute Error is 0.99

Average Squared Error is 0.06

Gradient Boosting

We further added gradient boosting to our decision trees to enhance the efficiency and fitting of Decision Tree.



Fit Statistics for Gradient boosting

Results - Node: Gradient Boosting Diagram: Project

File Edit View Window

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
y		_NOBS_	Sum of Frequencies	24712	16476	
y		_SUMW_	Sum of Case Weights Times ...	49424	32952	
y		_MISC_	Misclassification Rate	0.051352	0.086368	
y		_MAX_	Maximum Absolute Error	0.991366	0.990734	
y		_SSE_	Sum of Squared Errors	1907.804	1908.863	
y		_ASE_	Average Squared Error	0.038601	0.057929	
y		_RASE_	Root Average Squared Error	0.196471	0.240684	
y		_DIV_	Divisor for ASE	49424	32952	
y		_DFT_	Total Degrees of Freedom	24712		

As you can see root average squared error is 0.196 and Maximum Absolute Error is 0.99

Average Squared Error is 0.03

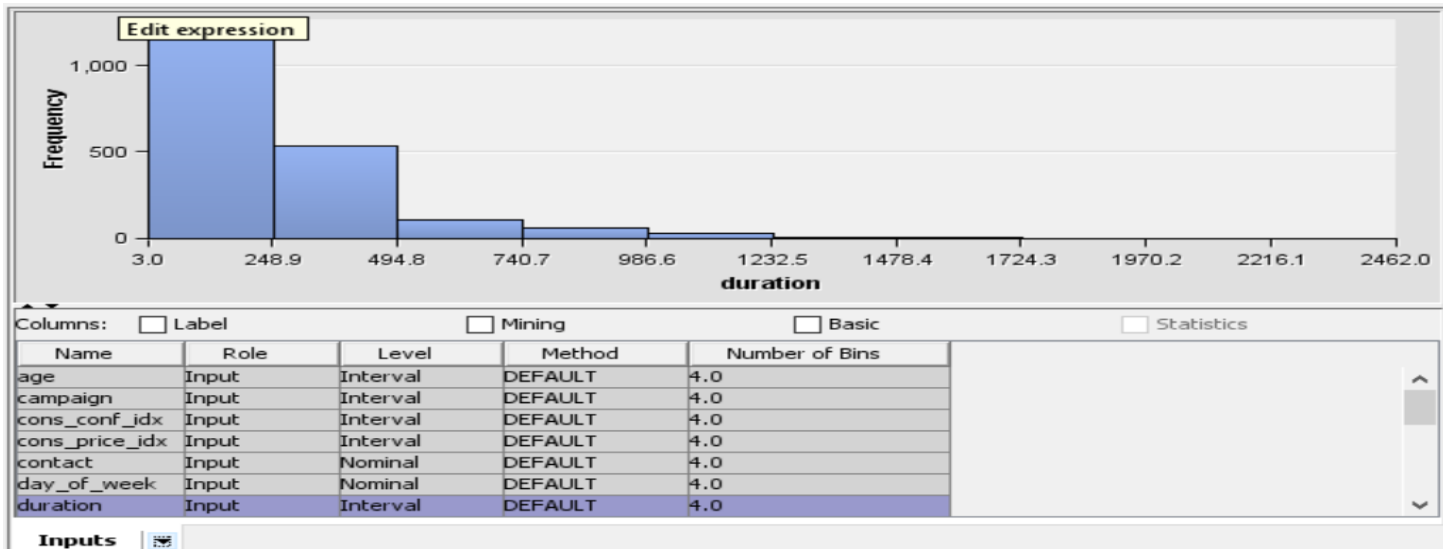
So as per analysis of all three models Gradient Booting shows much better efficiency in terms of average squared error, root average squared error, maximum absolute error.

Now we will use parametric data for predictive model.

Logistic Regression

Transforming data can improve the model accuracy. It will stabilize the variance, remove non linearity, check skewness in data.

Transforming input data leads to better fit the model.



Columns: Label | Mining | Basic | Statistics

Name	Role	Level	Method	Number of Bins
age	Input	Interval	DEFAULT	4.0
campaign	Input	Interval	DEFAULT	4.0
cons_conf_idx	Input	Interval	DEFAULT	4.0
cons_price_idx	Input	Interval	DEFAULT	4.0
contact	Input	Nominal	DEFAULT	4.0
day_of_week	Input	Nominal	DEFAULT	4.0
duration	Input	Interval	DEFAULT	4.0

Inputs

Columns: Label | Mining

Name	Method	Number of Bins	Role	Level
age	Default	4	Input	Interval
campaign	Log 10	4	Input	Interval
cons_conf_idx	Log 10	4	Input	Interval
cons_price_idx	Log 10	4	Input	Interval
contact	Default	4	Input	Nominal
day_of_week	Default	4	Input	Nominal
duration	Log 10	4	Input	Interval
education	Default	4	Input	Nominal
emp_var_rate	Log 10	4	Input	Interval
euribor3m	Log 10	4	Input	Interval
housing	Default	4	Input	Nominal
job	Default	4	Input	Nominal
loan	Default	4	Input	Nominal
marital	Default	4	Input	Nominal
month	Default	4	Input	Nominal
nr_employed	Log 10	4	Input	Interval
pdays	Log 10	4	Input	Interval
poutcome	Default	4	Input	Nominal
previous	Default	4	Input	Interval
y	Default	4	Target	Binary

Log10 transformation will control the skewness and kurtosis in data

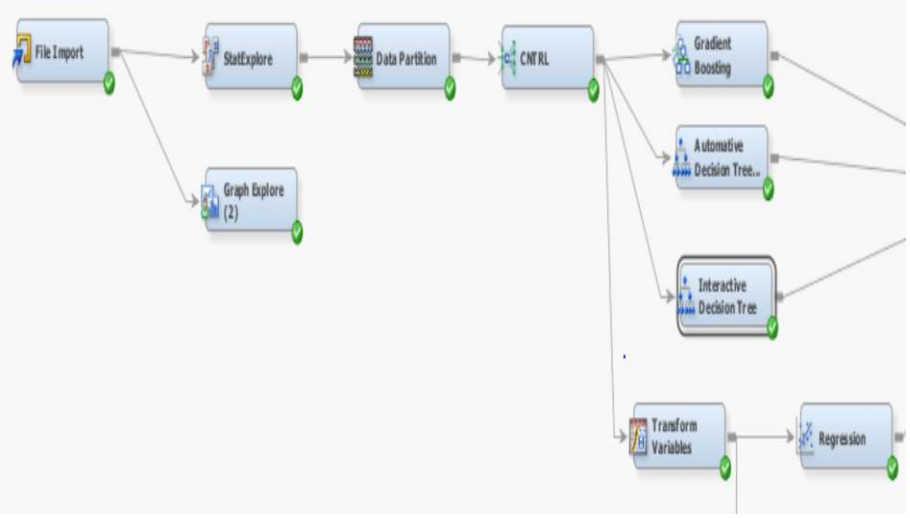
Transformations Statistics

Log

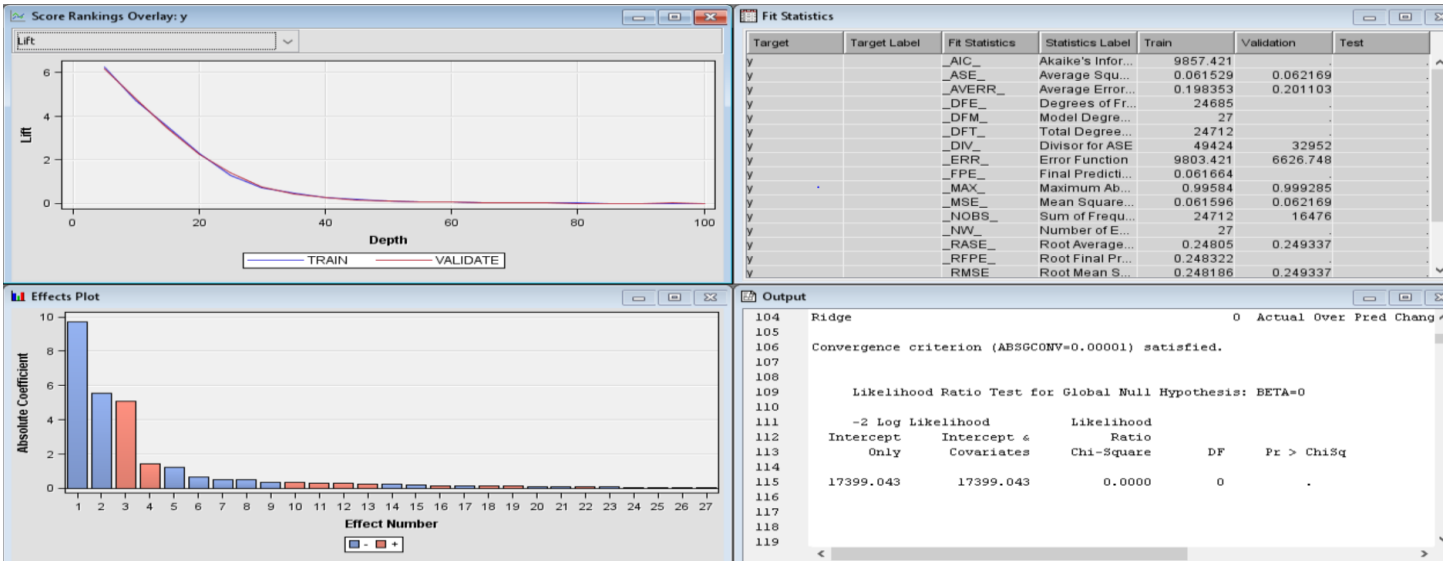
Source	Method	Variable Name	Formula	Number of Levels	Non Missing	Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurtosis	Label	
Input	Original	campaign			24712	0	1	43	2.553496	2.725718	4.450209	30.70066		
Input	Original	cons_conf_idx			24712	0	-50.8	-26.9	-40.5115	4.649959	0.289891	-0.39286	cons.conf.idx	
Input	Original	cons_price_idx			24712	0	92.201	94.767	93.57124	0.57848	-0.22085	-0.82309	cons.price.idx	
Input	Original	duration			24712	0	0	4199	255.7044	252.5464	3.023433	16.6553		
Input	Original	emp_var_rate			24712	0	-3.4	1.4	0.070282	1.576686	-0.70571	-1.09616	emp.var.rate	
Input	Original	euribor3m			24712	0	0.634	5.045	3.604938	1.742093	-0.68758	-1.43898		
Input	Original	nr_employed			24712	0	4963.6	5228.1	5166.436	72.64141	-1.02375	-0.05684	nr.employed	
Input	Original	pdays			24712	0	0	999	963.7224	183.8074	-5.01867	23.18937		
Output	Computed	LG10_campaign	log10(campaign ...		24712	0	0.30103	1.643453	0.48464	0.212957	1.354228	2.018378	Transformed...	
Output	Computed	LG10_cons_conf_idx	log10(cons_conf_...		24712	0	0	1.396199	1.008983	0.208785	-0.98064	1.810555	Transformed...	
Output	Computed	LG10_cons_price_idx	log10(cons_price...		24712	0	1.969421	1.981216	1.975751	0.002658	-0.2312	-0.79522	Transformed...	
Output	Computed	LG10_duration	log10(duration + 1)		24712	0	0	3.623249	2.244219	0.396194	-0.42596	0.889621	Transformed...	
Output	Computed	LG10_emp_var_rate	log10(emp_var_ra...		24712	0	1.93E-16	0.763428	0.612138	0.200348	-1.22902	0.690521	Transformed...	
Output	Computed	LG10_euribor3m	log10(euribor3m ...		24712	0	0.213252	0.781396	0.620747	0.206711	-0.76881	-1.25881	Transformed...	
Output	Computed	LG10_nr_employed	log10(nr_employe...		24712	0	3.695884	3.718427	3.713232	0.00615	-1.04285	-0.2499	Transformed...	
Output	Computed	LG10_pdays	log10(pdays + 1)		24712	0	0	3	2.921542	0.410945	-5.10219	24.32832	Transformed...	

Selection model in properties is step wise

Term Editor	...
Class Targets	
Regression Type	Logistic Regression
Link Function	Logit
Model Options	
Suppress Intercept	No
Input Coding	Deviation
Model Selection	
Selection Model	Stepwise
Selection Criterion	Default
Use Selection Defaults	Yes
Selection Options	...
Optimization Options	
Technique	Default
Default Optimization	Yes
Max Iterations	0
Max Function Calls	0
Maximum Time	1 Hour
Convergence Criteria	
Uses Defaults	Yes
Options	...



Regression statistics



Fit Statistics for Logistic Regression Model

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
y		_AIC_	Akaike's Information Criterion	9857.421	.	.
y		_ASE_	Average Squared Error	0.061529	0.062169	.
y		_AVERR_	Average Error Function	0.198353	0.201103	.
y		_DFE_	Degrees of Freedom for Error	24685	.	.
y		_DFM_	Model Degrees of Freedom	27	.	.
y		_DFT_	Total Degrees of Freedom	24712	.	.
y		_DIV_	Divisor for ASE	49424	32952	.
y		_ERR_	Error Function	9803.421	6626.748	.
y		_FPE_	Final Prediction Error	0.061664	.	.
y		_MAX_	Maximum Absolute Error	0.99584	0.999285	.
y		_MSE_	Mean Square Error	0.061596	0.062169	.
y		_NOBS_	Sum of Frequencies	24712	16476	.
y		_NW_	Number of Estimate Weights	27	.	.
y		_RASE_	Root Average Sum of Squares	0.24805	0.249337	.
y		_RFPE_	Root Final Prediction Error	0.248322	.	.
y		_RMSE_	Root Mean Squared Error	0.248186	0.249337	.
y		_SBC_	Schwarz's Bayesian Criterion	10076.53	.	.
y		_SSE_	Sum of Squared Errors	3041.009	2048.586	.
y		_SUMW_	Sum of Case Weights Times ...	49424	32952	.
y		_MISC_	Misclassification Rate	0.088864	0.089099	.

As you can see root mean squared error is 0.24 and Maximum Absolute Error is 0.99

Average Squared Error is 0.06

Neural Network Model

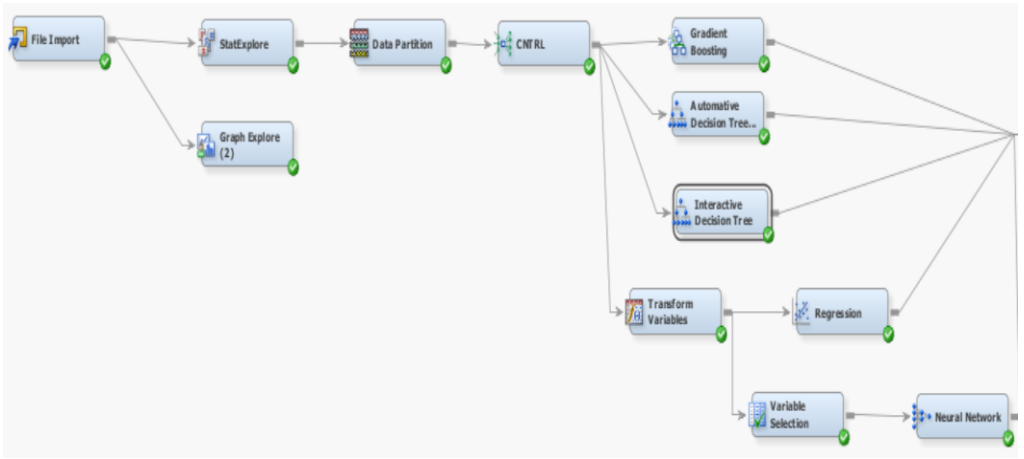
Neural networks are using parametric values and can handle varieties of nonlinear relation target and input variables.

They are better than Regression models

As they work on phenomena that should have more r2 value by using variable selection node

Variable Selection

Variable	None
Role	Input
Chi-Square Options	
Number of Bins	50
Maximum Pass Number	6
Minimum Chi-Square	3.84
R-Square Options	
Maximum Variable Number	3000
Minimum R-Square	0.005
Stop R-Square	5.0E-4
Use AOV16 Variables	No
Use Group Variables	Yes
Use Interactions	No
Use SPD Engine Library	Yes
Print Option	Default
Score	
Hides Rejected Variables	Yes
Hides Unused Variables	Yes



Variables having less than R – Square value to 0.005 has been rejected for neural network

Following variables gets rejected because of low r square value

Variable Selection

Variable Name	Role ▲	Measurement Level	Type	Label	Reasons for Rejection
G_job	Input	Nominal	Numeric	Grouped Levels for job	
G_month	Input	Nominal	Numeric	Grouped Levels for month	
LG10_cons_price_idx	Input	Interval	Numeric	Transformed: cons.price.idx	
LG10_duration	Input	Interval	Numeric	Transformed duration	
LG10_nr_employed	Input	Interval	Numeric	Transformed: nr.employed	
LG10_pdays	Input	Interval	Numeric	Transformed pdays	
previous	Input	Interval	Numeric		
LG10_campaign	Rejected	Interval	Numeric	Transformed campaign	Varsel:Small R-square value
LG10_cons_conf_idx	Rejected	Interval	Numeric	Transformed: cons.conf.idx	Varsel:Small R-square value
LG10_emp_var_rate	Rejected	Interval	Numeric	Transformed: emp.var.rate	Varsel:Small R-square value
LG10_euribor3m	Rejected	Interval	Numeric	Transformed euribor3m	Varsel:Small R-square value
age	Rejected	Interval	Numeric		Varsel:Small R-square value
contact	Rejected	Nominal	Character		Varsel:Small R-square value
day_of_week	Rejected	Nominal	Character		Varsel:Small R-square value
education	Rejected	Nominal	Character		Varsel:Small R-square value
housing	Rejected	Nominal	Character		Varsel:Small R-square value
job	Rejected	Nominal	Character		Varsel:Small R-square value, Group...
loan	Rejected	Nominal	Character		Varsel:Small R-square value
marital	Rejected	Nominal	Character		Varsel:Small R-square value
month	Rejected	Nominal	Character		Varsel:Small R-square value, Group...
poutcome	Rejected	Nominal	Character		Varsel:Small R-square value

So, we will selected variables as input for neural network

G_job

G_month

LG10_cons_price_idx

LG10_duration

LG10_nr_employed

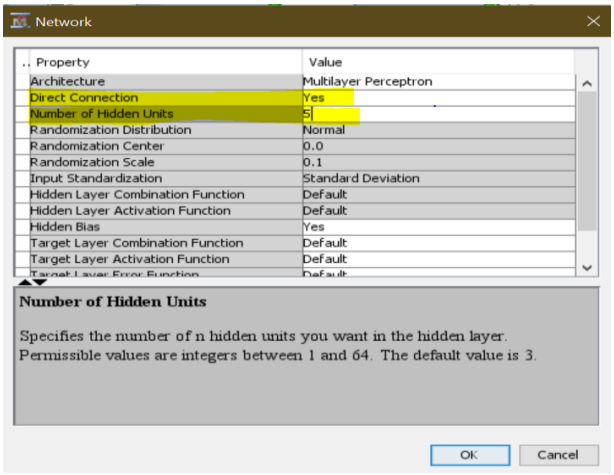
LG10_pdays

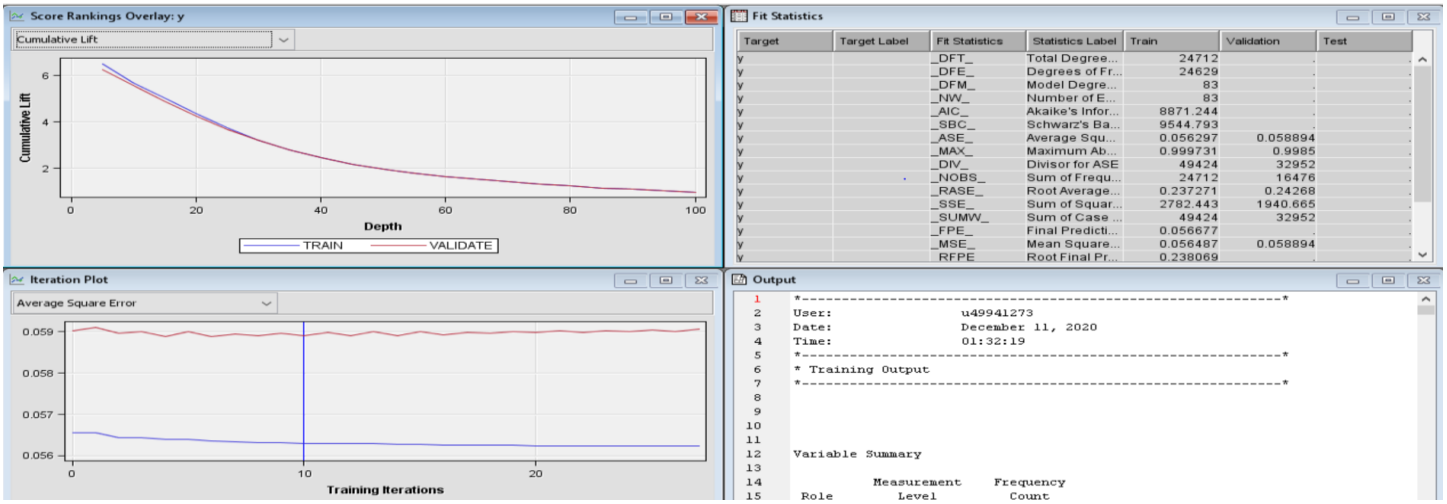
Previous

Adjusted data is now more suitable for modelling

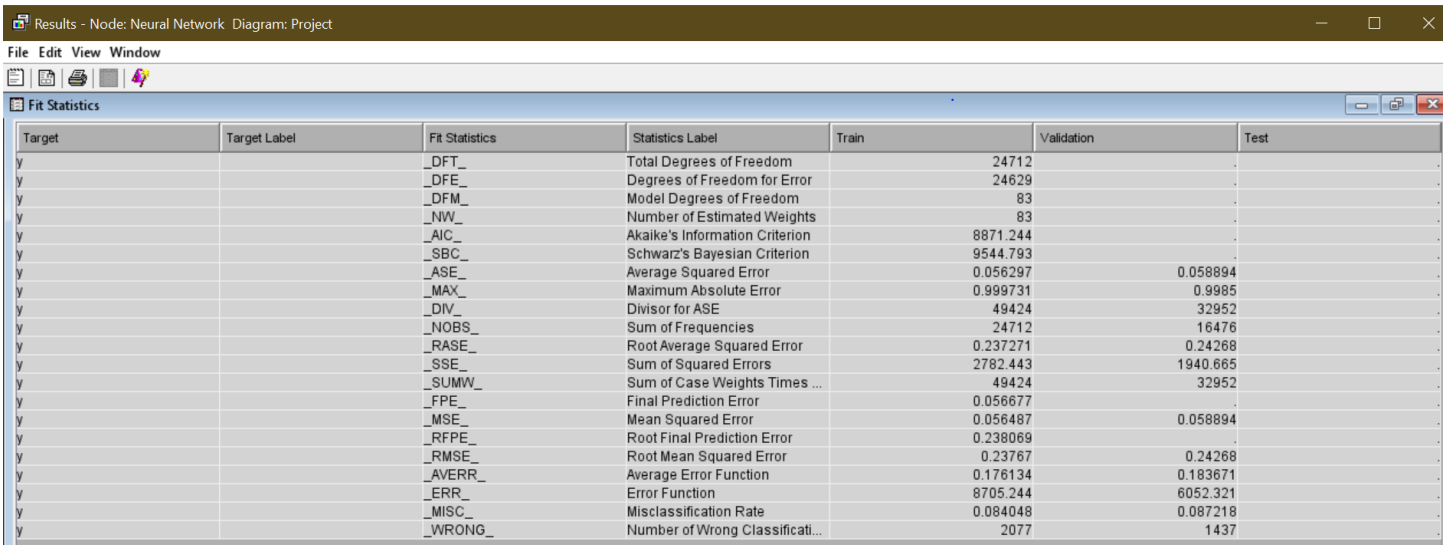
Setting Direct Connection to yes so that Neural Network can make connections between nodes.

Hidden Units are set to 5 for data processing and imputation, these are hidden layers in Neural Network for decision flow





Fit Statistics for Neural Network



Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
y		_DFT_	Total Degrees of Freedom	24712		
y		_DFE_	Degrees of Freedom for Error	24629		
y		_DFM_	Model Degrees of Freedom	83		
y		_NW_	Number of Estimated Weights	83		
y		_AIC_	Akaike's Information Criterion	8871.244		
y		_SBC_	Schwarz's Bayesian Criterion	9544.793		
y		_ASE_	Average Squared Error	0.056297	0.058894	
y		_MAX_	Maximum Absolute Error	0.999731	0.9985	
y		_DIV_	Divisor for ASE	49424	32952	
y		_NOBS_	Sum of Frequencies	24712	16476	
y		_RASE_	Root Average Squared Error	0.237271	0.24268	
y		_SSE_	Sum of Squared Errors	2782.443	1940.665	
y		_SUMW_	Sum of Case Weights Times ...	49424	32952	
y		_FPE_	Final Prediction Error	0.056677		
y		_MSE_	Mean Squared Error	0.056487	0.058894	
y		_RFPE_	Root Final Prediction Error	0.238069		
y		_RMSE_	Root Mean Squared Error	0.23767	0.24268	
y		_AVERR_	Average Error Function	0.176134	0.183671	
y		_ERR_	Error Function	8705.244	6052.321	
y		_MISC_	Misclassification Rate	0.084048	0.087218	
y		_WRONG_	Number of Wrong Classificati...	2077	1437	

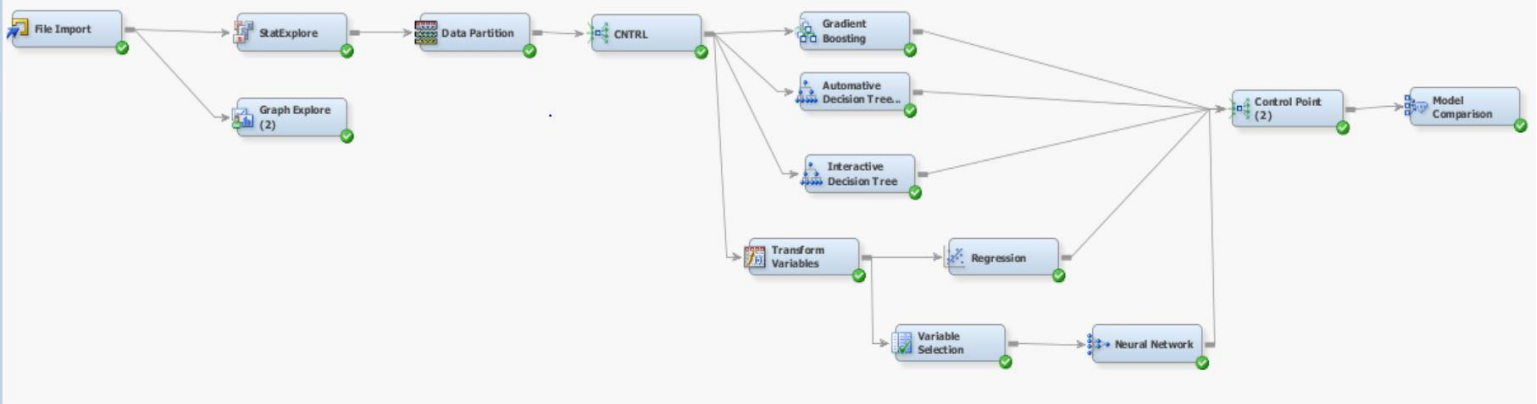
As you can see root mean squared error is 0.23 and Maximum Absolute Error is 0.99

Average Squared Error is 0.056

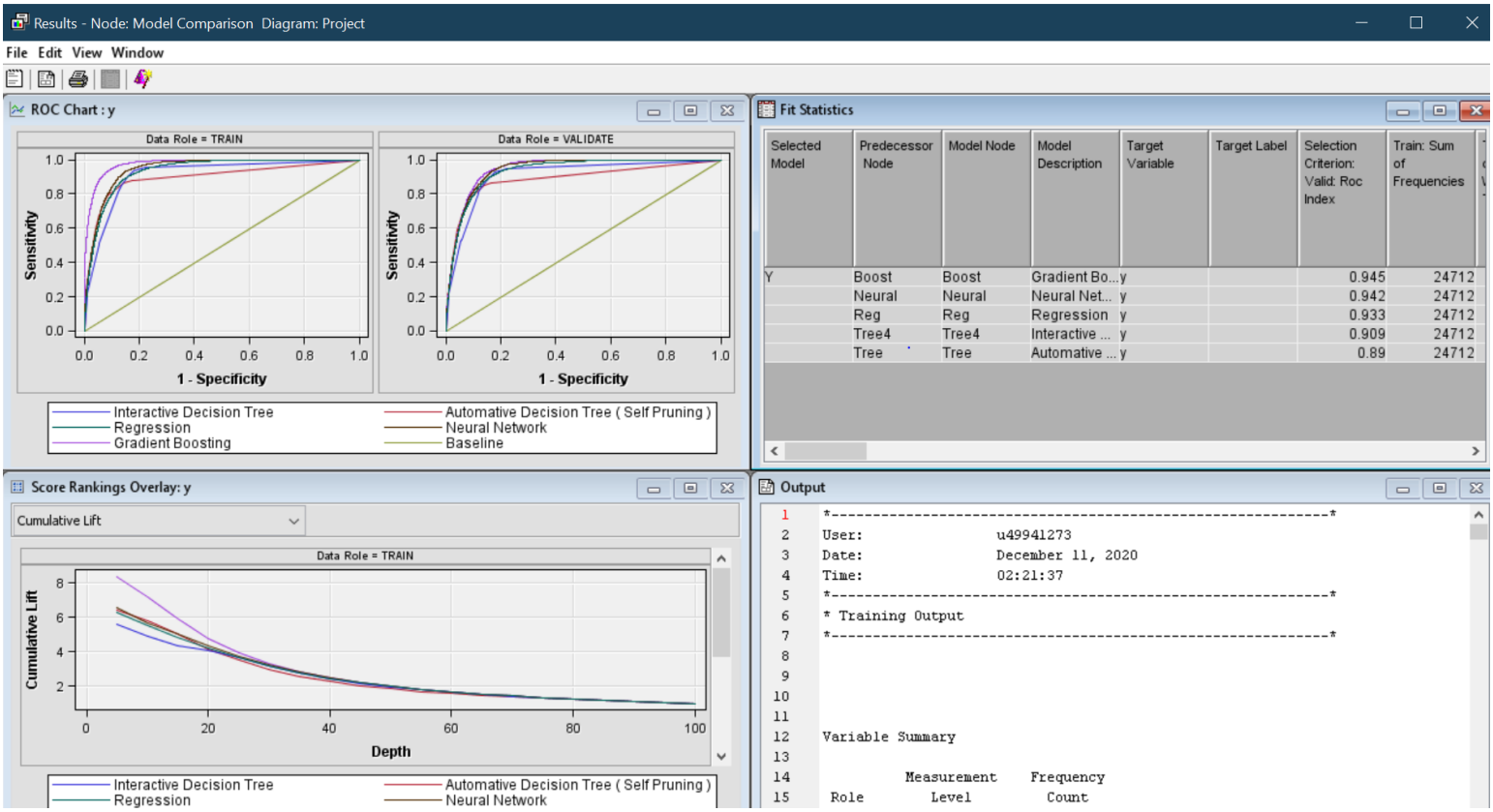
Model Comparison

- **Decision Tree (Self Pruning)**
- **Interactive Decision Tree**
- **Gradient Boosting – Rounding with of Decision Tree**
- **Logistic Regression (Parametric)**
- **Neural Network (Parametric)**

Final Data Flow



Model Comparison Node



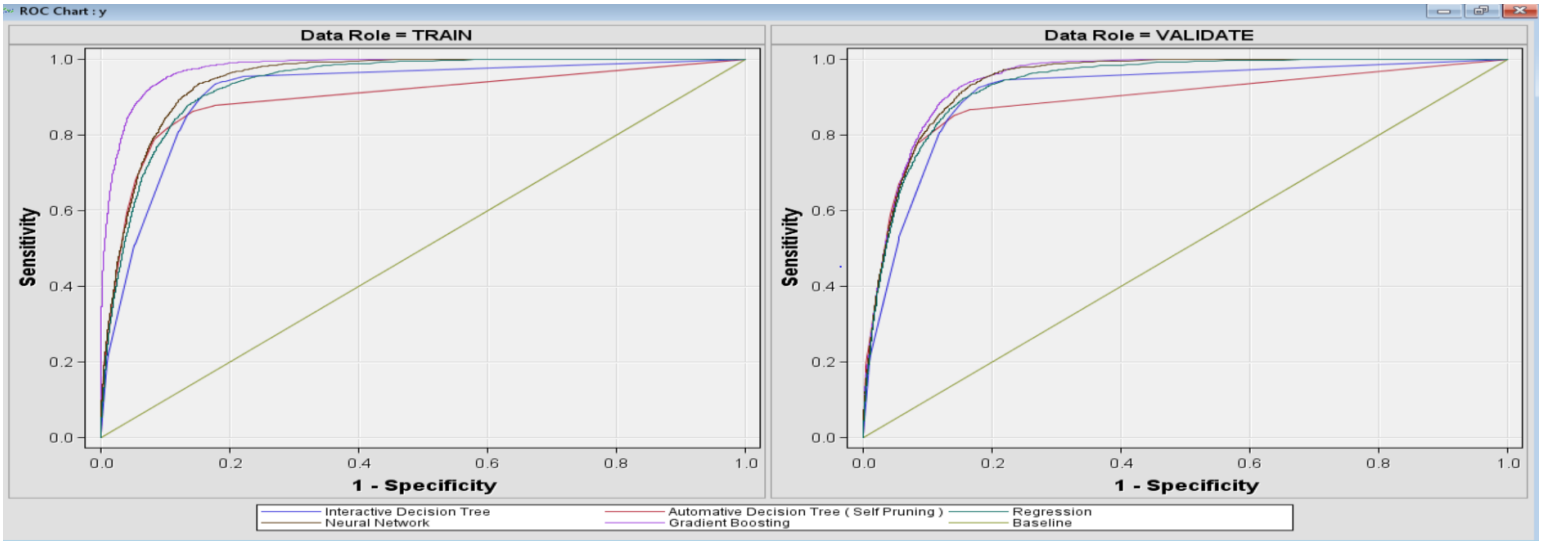
ROC (receiver operating characteristic curve) shows the classifications threshold for our all-predictive models.

It has 2 parameters: True Positive Rate and False positive Rate.

AUC (Area Under Curve): It shows the entire area under ROC Curve, More the Area More will the AUC value

ROC Curve with all predictive Models

Clearly shows Gradient boosting shows more accuracy than others.



Full Statistics of all predictive models.

Statistics	Gradient Boosting	Neural Network	Logistic Regression	Interactive Decision Tree	Decision Tree
Train: Roc Index	0.977	0.947	0.935	0.913	0.896
Selection Criterion: Valid: Roc Index	0.945	0.942	0.933	0.909	0.89
Train: Gini Coefficient	0.953	0.894	0.87	0.826	0.792
Train: Cumulative Lift	7.156447613312895	5.684223066670502	5.5010926962344975	4.8844084258439375	5.799667673530911
Train: Lift	5.931987685495686	4.84038704603401	4.72548171556436	4.160017236686766	5.210653924377103
Train: Root Average Squared Error	0.19647074893488817	0.2372707618801991	0.2480503884217969	0.25381333546752005	0.24203880307426232
Train: Sum of Squared Errors	1907.8037243640604	2782.4434114412184	3041.0090585772023	3183.9538465229725	2895.3954271375505
Valid: Roc Index	0.945	0.942	0.933	0.909	0.89
Valid: Gini Coefficient	0.891	0.884	0.866	0.818	0.779
Valid: Cumulative Lift	5.585928347783977	5.585928347783977	5.48896912863247	4.877679947072119	5.700219655329645
Valid: Lift	4.8910539438648515	4.912600437009631	4.804867971285734	4.1744208328073045	5.121142496558265
Valid: Root Average Squared Error	0.2406836012198447	0.24268023767596836	0.24933667375164256	0.2552980602418288	0.24491223889195823
Valid: Sum of Squared Errors	1908.8630919700777	1940.6651285369219	2048.5855356684688	2147.715784807899	1976.527020820928

As per the predictive models' statistics designed in SAS Enterprise Miner. Data divided in 60% - 40% percent for training and validation. The accuracy of predictive models always depends upon the data splitting, more data we have for training means we can train our model efficient but still we need more validation data to check the efficiency. Validation Data is different than from Testing data as validation data can be used back for training the model and we can find best efficient predictive model for our dataset.

Here **Receiver operating characteristic index** shows the model Gradient Boosting has 0.945 and the lowest is in Decision Tree of 0.89 in validation dataset.

And if we talk about the performance on the basis of Error, **(Root Average Squared Error)** shows the model Gradient Boosting has lowest 0.240 and highest error shown in Interactive Decision Tree.

Neural Network always in second rank having **ROC index 0.942 and Root Average Squared Error of 0.242.** Or in other words there is no big difference in the efficiency of Neural network and Gradient Boosting Model.

Cumulative LIFT shown in Validation data is **same** for Neural Network and Gradient Boosting with **5.585928347783977**

Gain Coefficient shown in Training and Validation data is highest for Gradient Boosting with **0.935 and 0.891**

Research paper Comparison:

Citation 1

S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing, Decision Support Systems, Elsevier, 62:22-31, June 2014 https://core.ac.uk/download/pdf/55631291.pdf

Study shows the Data mining models for telemarketing campaign. Study shows the comparison of 4 predictive models

Decision Tree

Neural Network

Support Vector Machine

Logistic Regression

On the basis of Area Under Curve and ALIFT Analysis. Study finally comes to conclusion for Neural Network is the best model for the prediction of Term Deposit account for new customers on the basis of previous campaign with AUC Score of 0.929 and ALIFT Score of 0.878.

Comparison of DM models for the modeling phase (**bold** denotes best value)

Metric	LR	DT	SVM ($\tilde{\gamma} = 2^{-7.8}, C = 3$)	NN ($\tilde{H} = 6, N_r = 7$)
AUC	0.900	0.833	0.891	**0.929***
ALIFT	0.849	0.756	0.844	**0.878***

$\star$ - Statistically significant under a pairwise comparison with SVM, LR and DT.

And for Analysis they used the dataset of 52944 rows from 2008 to 2013.

And our Dataset for analysis in 41,188 rows.

Study shows the dataset they used for research is unbalanced as 6557 observations for success only (12.38%)

Our Gradient Boosting Model and Neural Network Model shows the high accuracy than the models Neural Network prepared in the study.

Below observation is for Gradient Boosting Model

Obs	TARGET	TARGETLABEL	_AUR_	_GINI_	KS	_KS_PROB_ CUTOFF	_KS_BIN_	BINNED_KS_ PROB_ CUTOFF
1	y		0.977	0.953	0.849	0.162	0.848	0.22

Obs	TARGET	TARGETLABEL	_VAUR_	_VGINI_	VKS	_VKS_ PROB_ CUTOFF_	_VKS_ BIN_	_VBINNED_ KS_PROB_ CUTOFF_
1	y		0.945	0.891	0.776	0.104	0.774	0.117

As you can see AUR (operating characteristic curve) shows 0.945 in the validation data. Which is much higher than the neural network used in the study of 0.929

Also, ROC curve and LIFT analysis shows ROC index 0.945 in Gradient Boosting

Below observation is for Neural Network Model

Obs	TARGET	TARGETLABEL	_AUR_	_GINI_	KS	_KS_PROB_ CUTOFF	_KS_BIN_	BINNED_KS_ PROB_ CUTOFF
1	y		0.947	0.894	0.782	0.119	0.777	0.151

Obs	TARGET	TARGETLABEL	_VAUR_	_VGINI_	VKS	_VKS_ PROB_ CUTOFF_	_VKS_ BIN_	_VBINNED_ KS_PROB_ CUTOFF_
1	y		0.942	0.884	0.763	0.106	0.761	0.151

As you can see AUR (operating characteristic curve) shows 0.942 in the validation data. Which is much higher than the neural network used in the study of 0.929

Also, ROC curve and LIFT analysis shows ROC index 0.942 in Gradient Boosting

Citation 2

Justice Asare-Frempong. Manoj Jayabalan. Predicting customer response to bank direct telemarketing campaign. 2017 IEEE The International Conference on Engineering Technologies and Technopreneur ship (ICE2T 2017), https://ieeexplore.ieee.org/document/8215961

Aim for this study is to identify the customers based on previous campaign that they will opt for term deposit account. Data used in study takes 45147 observations with 17 variables.

Study shows 4 predictive models

Multilayer Perceptron Neural Network (MLPNN), Decision Tree (C4.5), Logistic Regression and Random Forest (RF)

CLASSIFIER	ACCURACY	AUC
DT	84.7%	87.7%
RF	86.8%	92.7%
LR	83.5%	90.9%
MLPNN	82.9%	90.0%

In addition to this analysis study shows that the month of September proved to be the most-fertile month for customers in relation to subscription of term deposit.

Their Maximum Area under curve shown by random forest of 92.7% means 0.927 Area under curve.

Our model both gradient boosting and Neural Network shows 0.945 and 0.942 which is much higher accuracy than their model.

Citation 3

S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimaraes, Portugal, October, 2011. EUROSIS. [bank.zip]

Study shows an implementation of data mining project based upon the CRISP-DM life cycle. Data set contains 45211 observations out of which 5289 are successful having 11.7 success rate.

Study shows 3 data mining models for the prediction of telemarketing campaign

Naïve Bayes (NB), Decision Trees (DT) and Support Vector Machines (SVM)

CRISP-DM Iteration	1st	2nd		3rd		
Instances × Attributes (Nr. Possible Results)	79354×59 (12)	55817 × 53 (2)		45211 × 29 (2)		
Algorithm	NB	NB	DT	**NB**	**DT**	**SVM**
Number of executions (runs)	1	20	20	20	20	20
AUC (Area Under the ROC Curve)	0.776	0.823	0.764	**0.870**	**0.868**	**0.938**
ALIFT (Area Under the LIFT Curve)	0.687	0.790	0.591	**0.827**	**0.790**	**0.887**

ROC curve

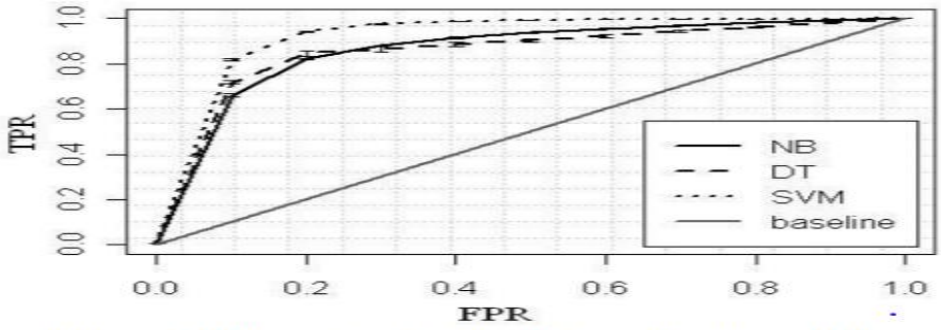


Figure 3 ROC curves for the best predicting models

Their result shows maximum Area under curve value is 0.938 by support vector machine.

Our model both gradient boosting and Neural Network shows 0.945 and 0.942 which is much higher accuracy than their model.

Citation 4

Oluwaseun Esther Oluwabusola. Applying Business Analytics in Practice to a Bank Telemarketing Dataset. University of Strathclyde (2015). https://local.cis.strath.ac.uk/wp/extras/msctheses/papers/strath_cis_publication_2714.pdf

Study shows 6 predictive models for bank telemarketing dataset

Study also shows data mining techniques which can improve the efficiency of the model

Bagging Result

Boosting Result

Classifier stacking result

After all the stacking up final accuracy is as follow:

Study Result:

Performance Measurement	Classification Algorithms					
	J48	Naïve Bayes	Random Forest	Multilayer Perceptron	LibSVM	SMO
ROC	0.920	0.871	0.942	0.904	0.860	0.873
Precision	0.892	0.796	0.886	0.834	0.864	0.875
Recall	0.890	0.795	0.884	0.833	0.860	0.873
Percentage accuracy (%)	88.97	79.49	88.35	83.28	86.02	87.33

Their result shows the maximum ROC value to be 0.942 for Random Forest which is equal to the ROC value of our model Neural Network 0.942 but out gradient boost model shows much higher accuracy of ROC value 0.945

Stacking results are quite impressive. Using base learners as Random Forest + SMO+J48 and meta learner as Naïve Bayes results shows ROC value of 0.942

Check below table:

Table 4.14: 4 classifier stacking result

Base Learners	Meta Learner	ROC	Precision	Recall
Random Forest + Naïve Bayes + J48	SMO	0.889	0.894	0.889
Random Forest + SMO + J48	Naïve Bayes	0.942	0.887	0.885
Random Forest + Naïve Bayes + J48	LibSVM	0.890	0.896	0.890
Random Forest + Naïve Bayes + J48	Multilayer Perceptron	0.941	0.894	0.889

Citation 5

Nachev, A. (2015). Application of data mining techniques for direct marketing. Computational Models for Business and Engineering Domains. Available at: http://www.foibg.com/ibs_isc/ibs-30/ibs-30-p09.pdf.

Study shows the cross-validation and multiple runs for the partitioning of train and test sets (70% and 30%) for the direct marketing response task. Dataset contains 45211 observations and 17 variables.

Aim of this study is to compare different models on the basis of dataset percentage use for telemarketing research.

Study have taken Neural Networks (NN), Logistic Regression, Naïve Bayes, Linear and Quadratic Discriminant Analysis (QDA)

Model	NN		LR		NB		LDA		QDA	
% of dataset	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
98%	**90.489**	**0.915**	89.810	0.902	87.912	0.852	89.810	0.900	86.913	0.838
80%	**90.401**	**0.912**	89.510	0.896	88.212	0.853	89.710	0.900	87.213	0.835
60%	**90.342**	**0.910**	89.810	0.898	88.312	0.858	89.810	0.900	87.013	0.845
40%	**90.213**	**0.902**	89.710	0.895	86.813	0.847	89.910	0.901	86.813	0.831
20%	90.209	0.895	**90.210**	0.892	87.313	0.850	89.910	**0.898**	86.813	0.837
10%	**89.710**	0.893	89.710	0.889	87.712	0.844	89.610	**0.896**	86.313	0.826

Results on the basis of the data saturation maximum Area under curve achieved by Neural Network of about 0.915.

Our model both gradient boosting and Neural Network shows 0.945 and 0.942 which is much higher accuracy than their model.

Below image shows the ROC curve for different models. Leading 0.915 neural network.