CALIFORNIA STATE UNIVERSITY
EAST BAY

BANK REVENUES

MGMT 6285: Term Project

Team Members:

Anh T Nguyet Nguyen (xc8374)

Langqi Zhao (qb8354)

Navneet Chauhan (fg3276)

Table of Contents

Summary

Our team selected Bank Revenues dataset to find out how customer banking habits contribute to bank revenues and profitability. We found the dataset from JMP library. This dataset contains information on 7,420 bank customers with 16 predictor variables for us to run different model approaches then compare the results. Appendix 1 lists each of the variable names and the corresponding variable descriptions.

We used three different approaches to run the model and analyze the dataset, which includes Multiple Linear Regression Model with Log Transformation, Neural Networks with Log Transformation, and Classification Tree Model. Each model type analyses the data in a different way, which provide us with all-sided information in making predictions. Same as what we learned and did in class, our team are mainly focused on comparing R-square and RMSE/RASE to decide the best model and approach.

After comparing all three models we used, we found that Neural Network model can best predict bank revenues based on customer banking behaviors since it has the highest Rsquare value and the lowest RASE value for the validation data. Therefore, we concluded that we should proceed with the Neural Network result to predict the responses.

Introduction

Like any other businesses, banks are selling its product and service to make profits. However, their product just happens to be money. Banks provide customers with the assurance of security and convenient access to money, as well as the ability to save, invest, get loans, and so forth. A bank's profitability is close related to customers' banking habits since every single transaction conduct by customers could contribute to bank's revenue. How customer banking habits contribute to bank's revenues and profitability became a challenge and important part that every bank would like to explore.

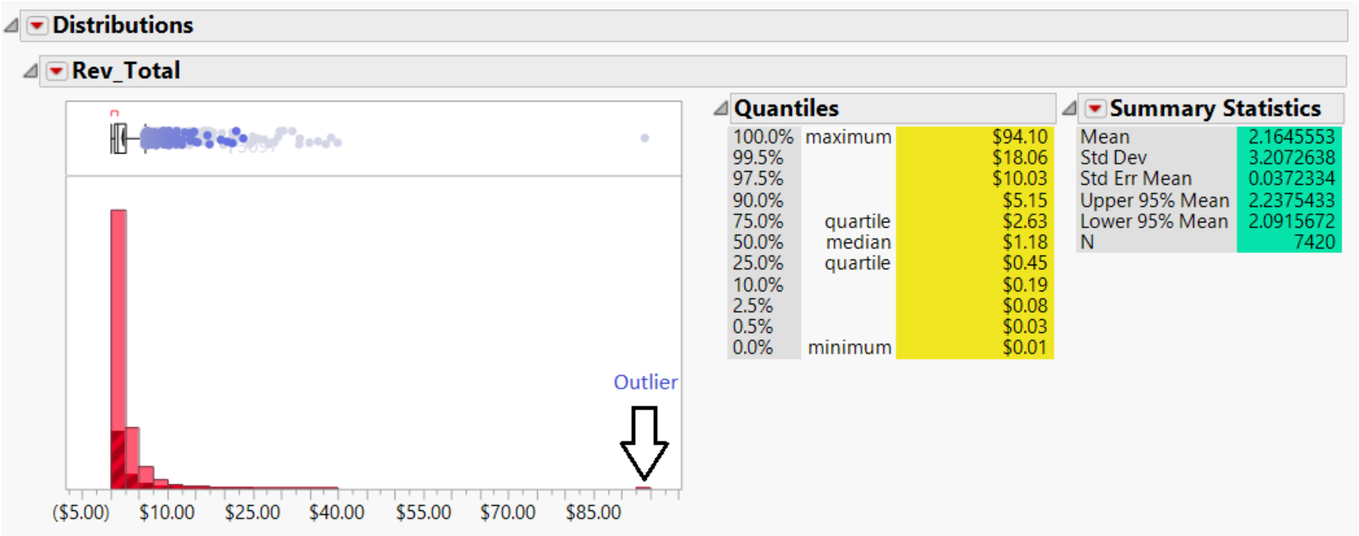
To understand how customer banking habits contribute to bank revenues and profitability will provide banks with a clear guide on their future marketing campaigns. Also, the bank is allowing to forecast bank revenues by using this information, which will help the bank to determine who should be prioritized and how to best help every customer while balancing productivity and profitability.

The bank has 7,420 bank customer age and bank account information, e.g., whether the customer has a savings account, whether the customer has received bank loans, whether the customer has received a special promotional offer in the previous one-month period, and other indicators of account activity. Our team decide to build models and help the bank to understand how specific customer banking habits contribute to bank revenues and profitability. Also, we want the bank able to predict profitability and revenue for a given customer by using the model we built.

Model Preparation

At start, we will look at the nominal variables which are contributing to the responses and how they are distributed. We have (Rev_Total) which is response variable. Using Graph Builder, we saw the distribution of the data which can be notice in below screenshot (Exhibit A).

Exhibit A. Distribution of Total Revenue



Distributions

Rev_Total

Quantiles

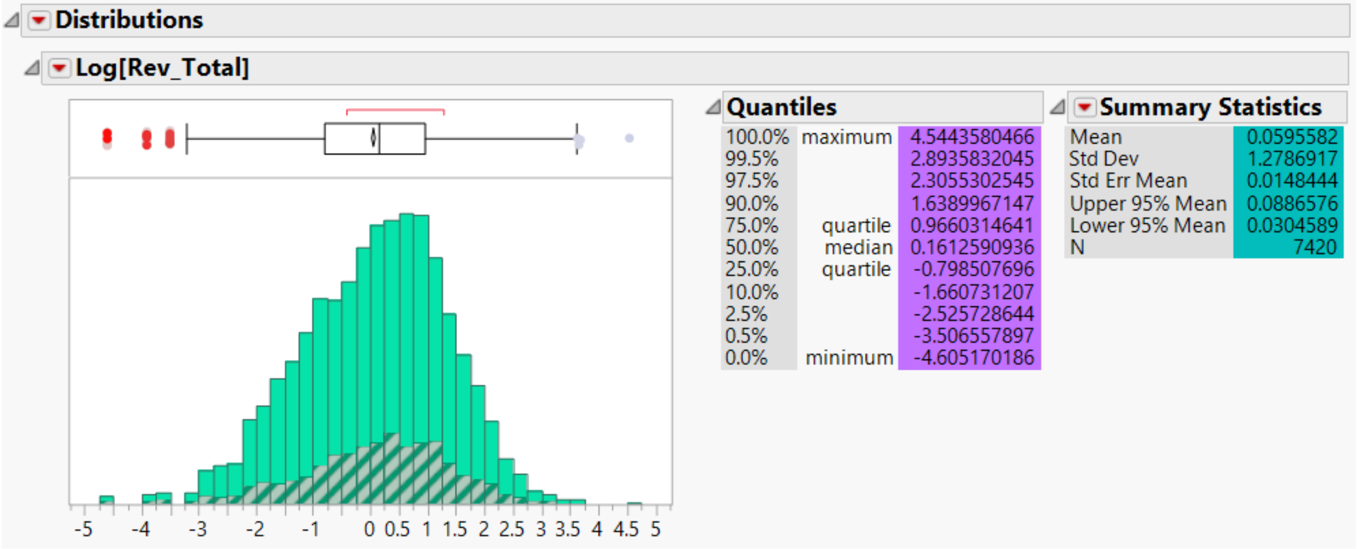
100.0%	maximum	$94.10
99.5%		$18.06
97.5%		$10.03
90.0%		$5.15
75.0%	quartile	$2.63
50.0%	median	$1.18
25.0%	quartile	$0.45
10.0%		$0.19
2.5%		$0.08
0.5%		$0.03
0.0%	minimum	$0.01

Summary Statistics

Mean	2.1645553
Std Dev	3.2072638
Std Err Mean	0.0372334
Upper 95% Mean	2.2375433
Lower 95% Mean	2.0915672
N	7420

The data is highly skewed and in regression especially, highly skewed data results in poorly fitting model. To normalize the highly skewed dataset we must apply transformation method which can be obtain by applying Log transformation (Natural Logarithm Transformation). Once we apply log transformation to the (Rev_Total) data we get below results (Exhibit B).

Exhibit B. **Revenue Total Log Transformation distribution**



Distributions

Log[Rev_Total]

Quantiles

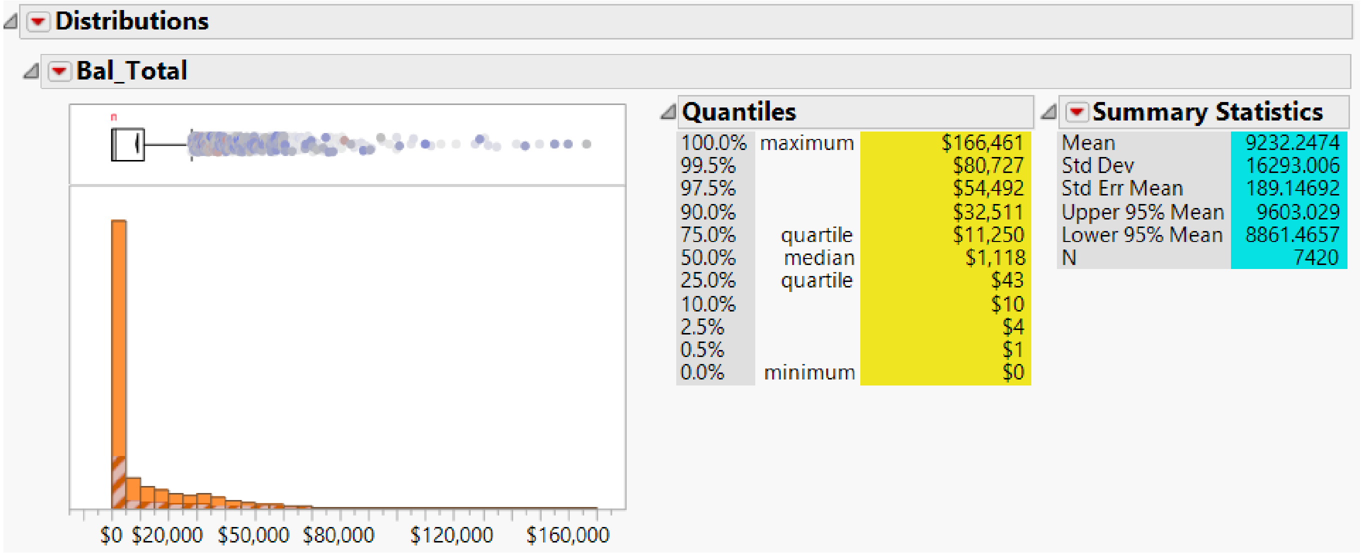
100.0%	maximum	4.5443580466
99.5%		2.8935832045
97.5%		2.3055302545
90.0%		1.6389967147
75.0%	quartile	0.9660314641
50.0%	median	0.1612590936
25.0%	quartile	-0.798507696
10.0%		-1.660731207
2.5%		-2.525728644
0.5%		-3.506557897
0.0%	minimum	-4.605170186

Summary Statistics

Mean	0.0595582
Std Dev	1.2786917
Std Err Mean	0.0148444
Upper 95% Mean	0.0886576
Lower 95% Mean	0.0304589
N	7420

A similar examination of the total balance can be observed, which is also highly skewed distribution, leads to the use of Log(Bal_Total) in our analyses. Please see Exhibit C & D below.

Exhibit C. Bal_Total distribution



Distributions

Bal_Total

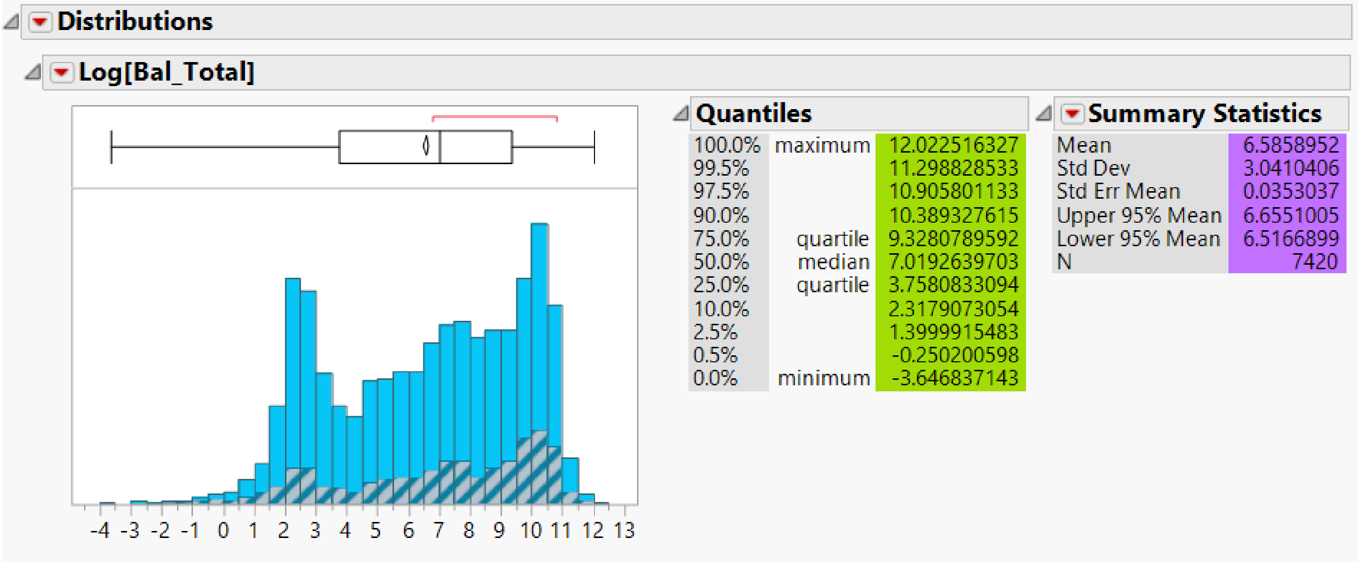
Quantiles

100.0%	maximum	$166,461
99.5%		$80,727
97.5%		$54,492
90.0%		$32,511
75.0%	quartile	$11,250
50.0%	median	$1,118
25.0%	quartile	$43
10.0%		$10
2.5%		$4
0.5%		$1
0.0%	minimum	$0

Summary Statistics

Mean	9232.2474
Std Dev	16293.006
Std Err Mean	189.14692
Upper 95% Mean	9603.029
Lower 95% Mean	8861.4657
N	7420

Exhibit D. Log(Bal_Total) Log transformation distribution



Distributions

Log[Bal_Total]

Quantiles

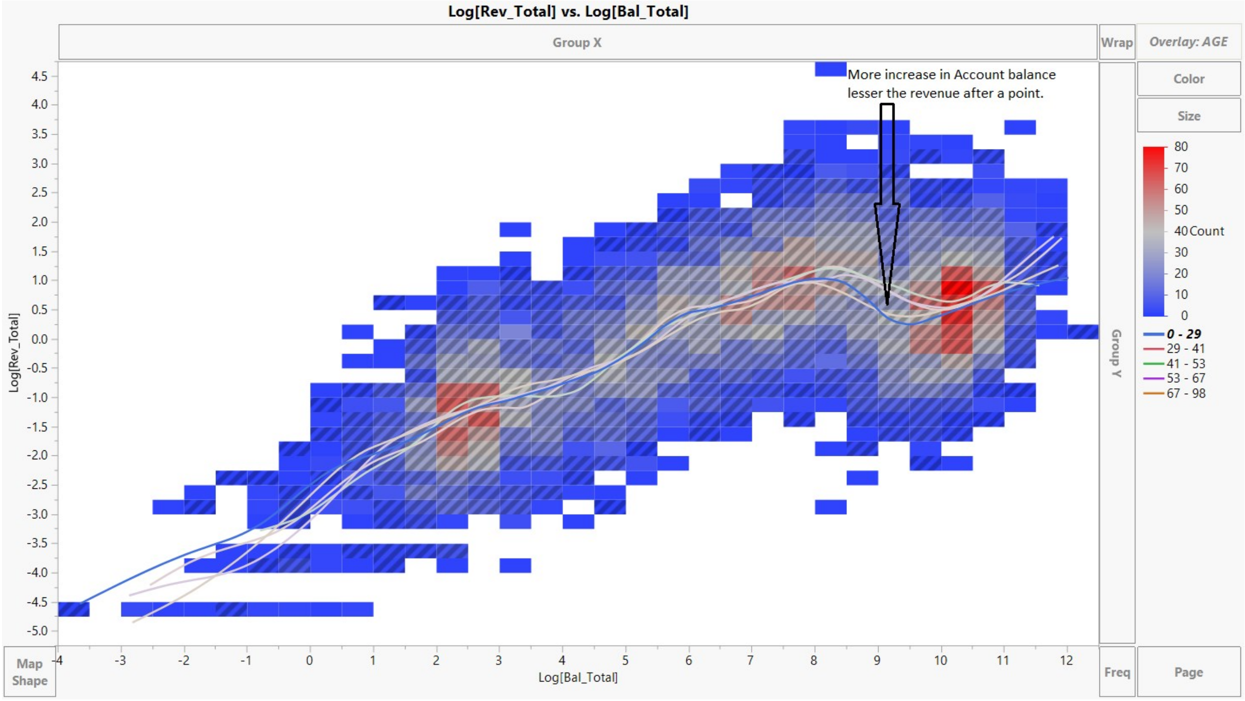
100.0%	maximum	12.022516327
99.5%		11.298828533
97.5%		10.905801133
90.0%		10.389327615
75.0%	quartile	9.3280789592
50.0%	median	7.0192639703
25.0%	quartile	3.7580833094
10.0%		2.3179073054
2.5%		1.3999915483
0.5%		-0.250200598
0.0%	minimum	-3.646837143

Summary Statistics

Mean	6.5858952
Std Dev	3.0410406
Std Err Mean	0.0353037
Upper 95% Mean	6.6551005
Lower 95% Mean	6.5166899
N	7420

Now the relationship between the Log(Rev_Total) and Log(Bal_Total) is shown in the below plot Exhibit E segmenting with Age.

Exhibit E. Relationship between Log(Rev_Total) Vs Log(Bal_Total)



More increase in Account balance lesser the revenue after a point.

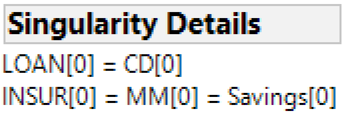
The relationship appears to be nearly linear at lower account balances; Higher account balances generally have higher revenues. This relationship, however seems to change at higher account balances as shown in Exhibit E.

Examining the other variables including Log(Rev_Total), many of the variables are categorical with two-level data. Other than total account balance, Log(Bal_Total) there is no other variable stands out as being strongly related to revenue. Other graphical methods and relationship between predictors and responses can be obtained by using fit Y by X and Graph builder tool in JMP. Kindly have a look at Variable Distribution with Log_Rev Data table in the attached JMP file to get a complete understanding of relationship and distribution of other variables with JMP.

Main Chapter

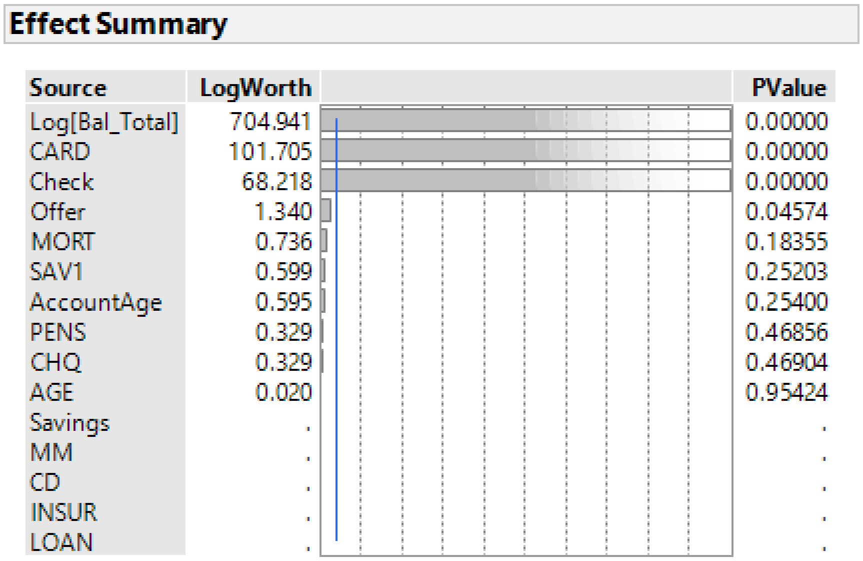
Model 1: Multiple Linear Regression Model

We created a regression model with Log[Rev_Total] as the dependent variable, Log[Bal_Total] (instead of Bal_Total) and all other variables as independent variables.



Singularity Details

LOAN[0] = CD[0]
INSUR[0] = MM[0] = Savings[0]

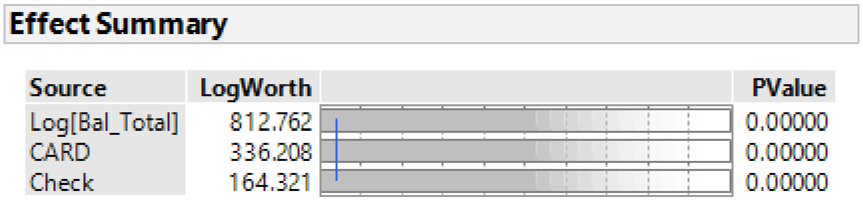


Effect Summary

Source	LogWorth	PValue
Log[Bal_Total]	704.941	0.00000
CARD	101.705	0.00000
Check	68.218	0.00000
Offer	1.340	0.04574
MORT	0.736	0.18355
SAV1	0.599	0.25203
AccountAge	0.595	0.25400
PENS	0.329	0.46856
CHQ	0.329	0.46904
AGE	0.020	0.95424
Savings	.	.
MM	.	.
CD	.	.
INSUR	.	.
LOAN	.	.

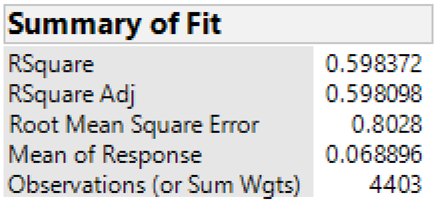
There are some immediate signs of trouble when we run this model. At the top of the Fit Least Squares window, we see some unexpected output, Singularity Details. This means that there are linear dependencies between predictor variables. The first row of this table, LOAN[0] = CD[0], indicates that JMP cannot identify the difference between these two variables, LOAN and CD. The second line indicates that JMP cannot identify the difference between INSUR, MM and Savings. We keep LOAN (and eliminate CD) and INSUR (eliminating MM and Savings) (as advised in JMP's Library Case Study "Bank Revenues" - Multiple Linear Regression with Transformation).

Because there are so many variables have P-values of coefficients are > 0.01, the first model we got is clearly not statistically significant. As a result, we used the Effect Summary table in the Response Log[Rev_Total] output to eliminate manually predictors with p-value greater than 0.01. The measurements of the final models with the variables with all with p-values < 0.01 are presented in the tables below (Effect Summary, Summary of Fit, Analysis of Variance and Parameter Estimates)



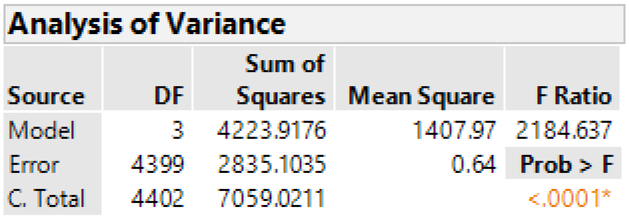
Effect Summary

Source	LogWorth		PValue
Log[Bal_Total]	812.762		0.00000
CARD	336.208		0.00000
Check	164.321		0.00000



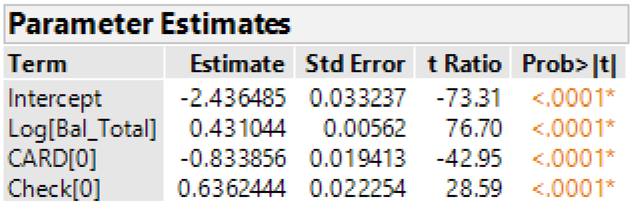
Summary of Fit

RSquare	0.598372
RSquare Adj	0.598098
Root Mean Square Error	0.8028
Mean of Response	0.068896
Observations (or Sum Wgts)	4403



Analysis of Variance

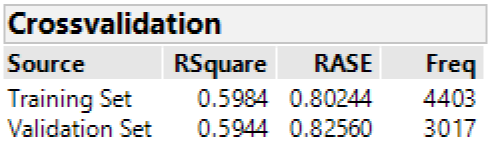
Source	DF	Sum of Squares	Mean Square	F Ratio
Model	3	4223.9176	1407.97	2184.637
Error	4399	2835.1035	0.64	Prob > F
C. Total	4402	7059.0211		<.0001*



Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	-2.436485	0.033237	-73.31	<.0001*
Log[Bal_Total]	0.431044	0.00562	76.70	<.0001*
CARD[0]	-0.833856	0.019413	-42.95	<.0001*
Check[0]	0.6362444	0.022254	28.59	<.0001*

In addition, based on the Crossvalidation table below, we can observe that the value of the validation RSquare (0.5944) is very similar to that for the training set (0.5984), and the RASE for the validation set is only 2.9% higher than that for the training set. Therefore, we can safely conclude that the model can give us accurate prediction and does not run any chances of overfitting.

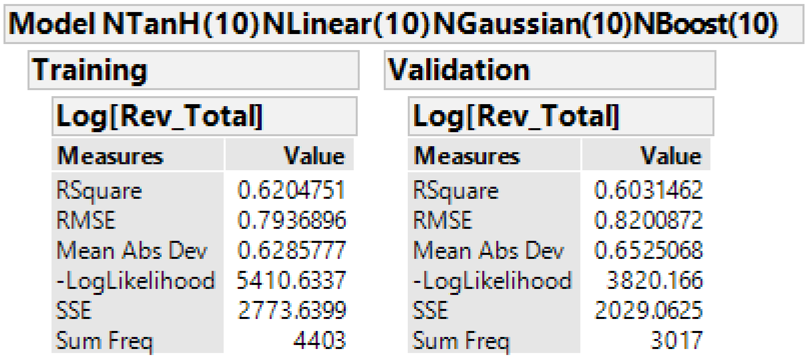


Crossvalidation

Source	RSquare	RASE	Freq
Training Set	0.5984	0.80244	4403
Validation Set	0.5944	0.82560	3017

Model 2: Neural Networks

The second model that was implemented was a neural network which are models that perform well for both classification and prediction. A neural network model enables capture of complex relationships between predictors and response variables. After running various tuning parameters for the neural network, we have found that the best performance is attained when using 10 nodes for the hidden layer for each activation function TanH, Linear and Gaussian. We set the number of models = 10 and the learning rate l=0.2. We observe that we have higher RSquare values for both training and validation with increased RSquare values for both training and validation data.

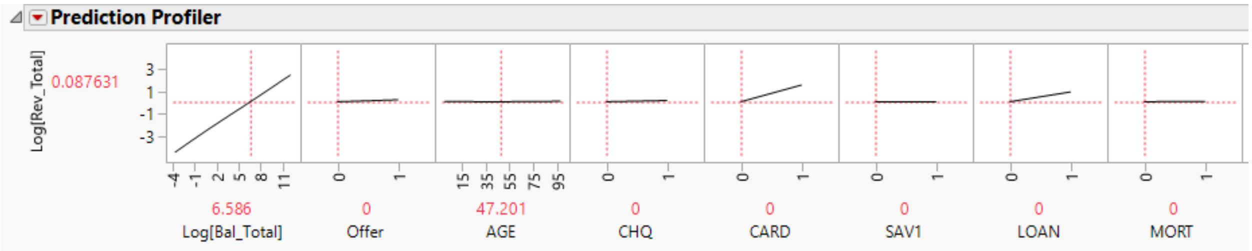


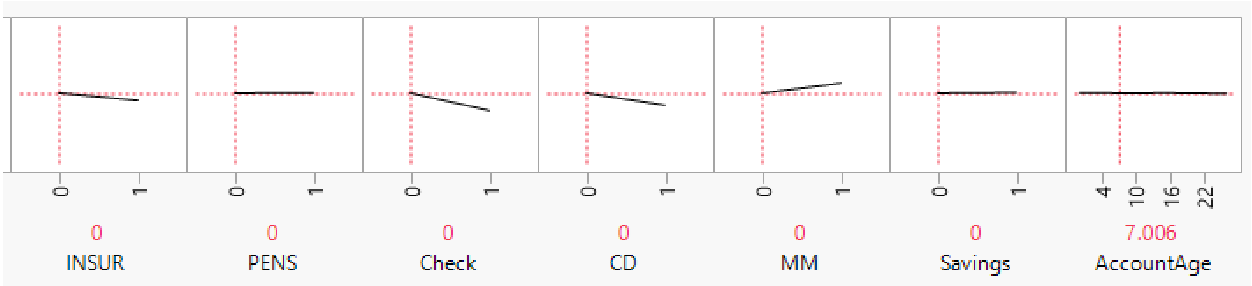
Model NTanH(10)NLinear(10)NGaussian(10)NBoost(10)

Training		Validation	
Log[Rev_Total]		Log[Rev_Total]	
Measures	Value	Measures	Value
RSquare	0.6204751	RSquare	0.6031462
RMSE	0.7936896	RMSE	0.8200872
Mean Abs Dev	0.6285777	Mean Abs Dev	0.6525068
-LogLikelihood	5410.6337	-LogLikelihood	3820.166
SSE	2773.6399	SSE	2029.0625
Sum Freq	4403	Sum Freq	3017

In this case, we observe that we have relatively high RSquare values for both training set (0.6205) and validation dataset (0.6031). Meanwhile, The RMSE in the validation set is only 3.32% higher than that in training set. These measurements imply that this Neural Networks model fit well into the validation dataset and we have low chances of overfitting. Considering its high RSquare value (validation), it could give us quite reliable predictions about the value of Log[Rev_Total].

Prediction Profiler

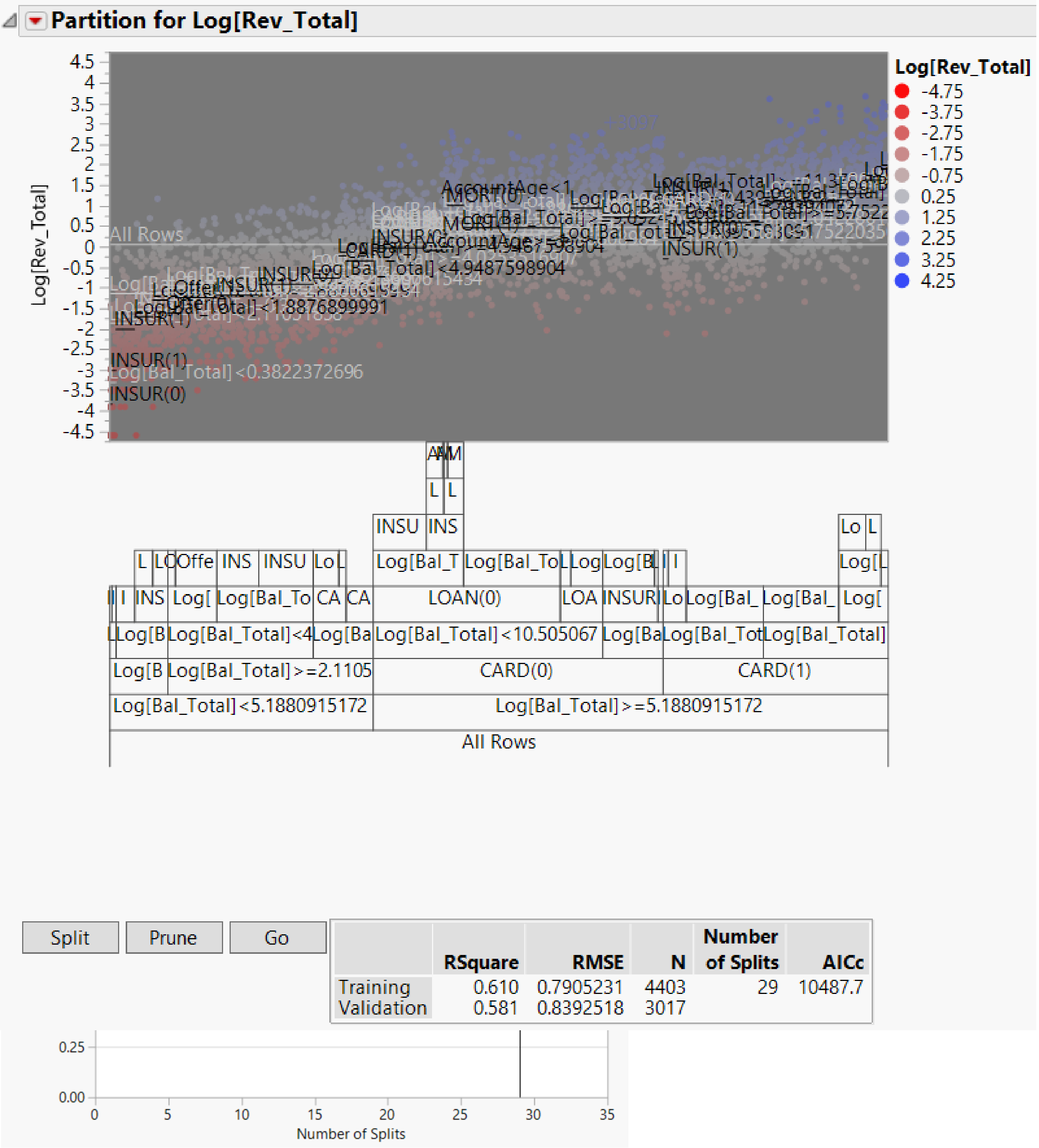




From the snippets from the profiler generated by neural network model, we can observe the relation between the predictors and the response variable Log(Rev_Total). By examining the profiler, we can observe which predictor has the highest contribution to the model. Clearly, Log(Bal_Total) has a large positive effect on the response (i.e., the slope of the Profiler line is steep). It could be interpreted that high account balance customers would generate more revenue. Other predictors like CARD, LOAN, Check, CD, while significant, have a relatively small effect on the response.

Model 3: Classification Tree Model

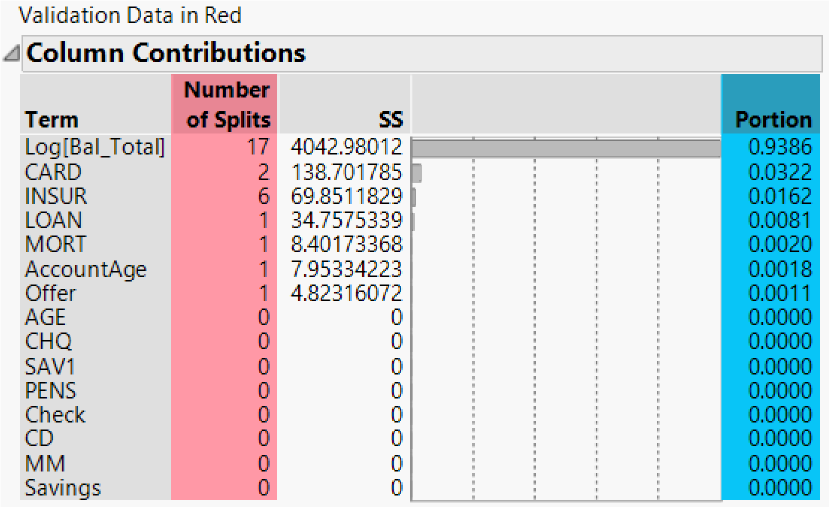
When building a classification tree, JMP iteratively splits the data based on values of predictors to form a subset.



	RSquare	RMSE	N	Number of Splits	AICc
Training	0.610	0.7905231	4403	29	10487.7
Validation	0.581	0.8392518	3017		

The above result states that there were 29 splits occurred based on the values of predictors at each level to get an optimal result with the best split count. The split history report below shows how RSquare value changes for training and validation data after each split. The vertical line drawn at 29 shows number of splits used in final model. We received RSquare at 0.581 for Validation and 0.610 for Training. RMSE stood at 0.8393 for Validation. This gives us clear understanding of data is not overfitted.

Training and Validation remains intact throughout all the splits and not differ much in the results. The main goal is to achieve maximum RSquare possible to improve the model accuracy and lowest AICs. Here we are getting AICs at 10487.7.

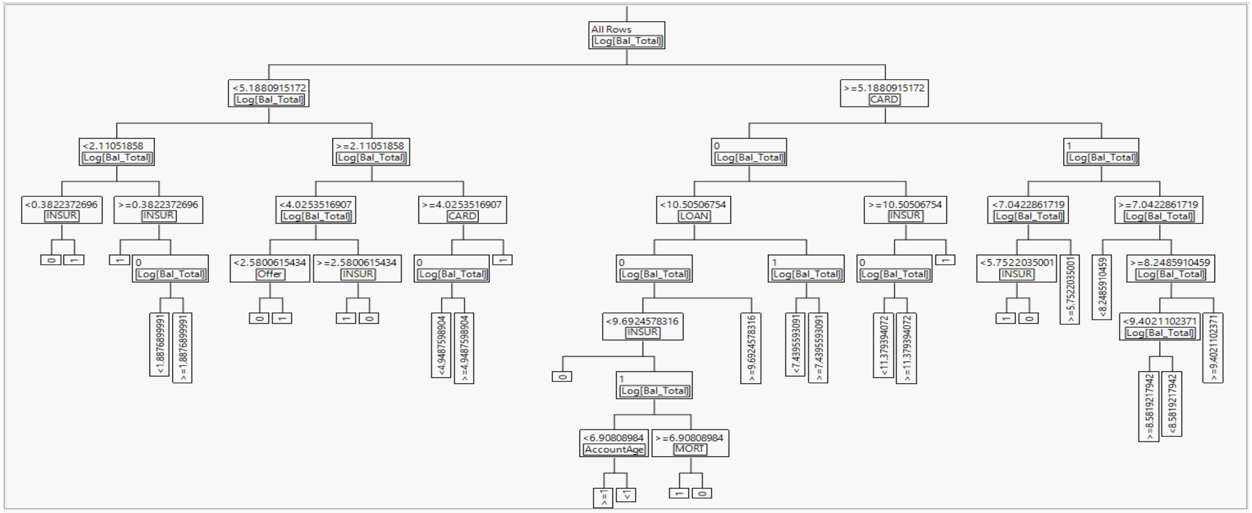


Validation Data in Red

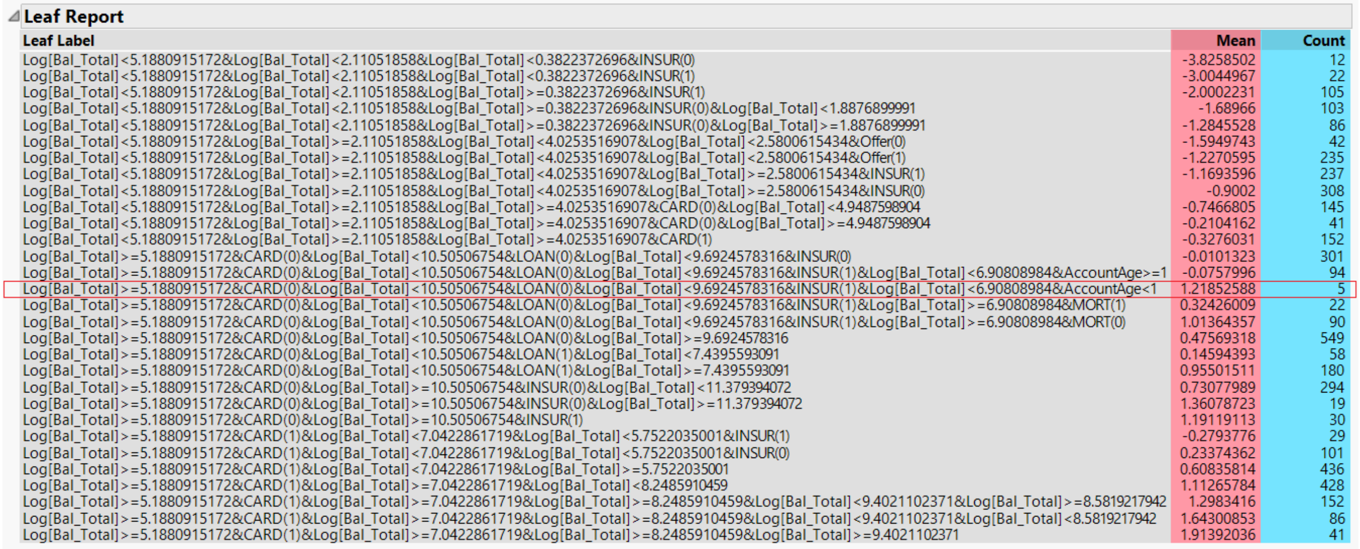
Column Contributions

Term	Number of Splits	SS	Portion
Log[Bal_Total]	17	4042.98012	0.9386
CARD	2	138.701785	0.0322
INSUR	6	69.8511829	0.0162
LOAN	1	34.7575339	0.0081
MORT	1	8.40173368	0.0020
AccountAge	1	7.95334223	0.0018
Offer	1	4.82316072	0.0011
AGE	0	0	0.0000
CHQ	0	0	0.0000
SAV1	0	0	0.0000
PENS	0	0	0.0000
Check	0	0	0.0000
CD	0	0	0.0000
MM	0	0	0.0000
Savings	0	0	0.0000

To summarize which variables are involved in these 29 splits, we turn on Column contribution table. This table indicates which variables are most important in terms of overall contribution to the model. From the table, we can clearly state that Bal_Total has contributed maximum to the revenue followed by CARD, INSUR, LOAN, MORT, AccountAge, and Offer contributed slightly in the responses. However, AGE, CHQ, SAV1, PENS, Check, CD, MM, Savings has not contributed at all in the model which is why it is mentioned as 0. Number of splits column gives us how many times split occurred with the respective terms. Below is the Tree diagram which shows complete optimal result of 29 splits that has been occurred with 17 splits occurred with Log(Bal_Total). It gives us the understanding of at what value split has occurred.



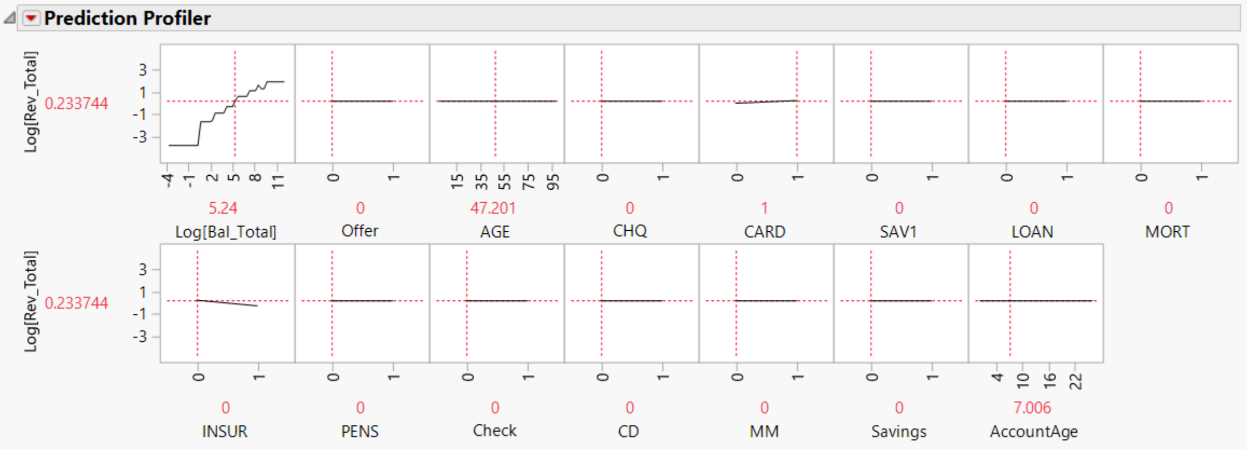
To elaborate this result, we look at leaf report shown below. Examining the leaf report for responses we get all the possible information with the summary of splits and an easy way of reading tree based on the predictors. By reading leaf report we can provide end-users with rule for classifying outcomes together.



Leaf Report

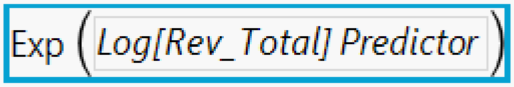
Leaf Label	Mean	Count
Log[Bal_Total]<5.1880915172&Log[Bal_Total]<2.11051858&Log[Bal_Total]<0.3822372696&INSUR(0)	-3.8258502	12
Log[Bal_Total]<5.1880915172&Log[Bal_Total]<2.11051858&Log[Bal_Total]<0.3822372696&INSUR(1)	-3.0044967	22
Log[Bal_Total]<5.1880915172&Log[Bal_Total]<2.11051858&Log[Bal_Total]>=0.3822372696&INSUR(1)	-2.0002231	105
Log[Bal_Total]<5.1880915172&Log[Bal_Total]<2.11051858&Log[Bal_Total]>=0.3822372696&INSUR(0)&Log[Bal_Total]<1.8876899991	-1.68966	103
Log[Bal_Total]<5.1880915172&Log[Bal_Total]<2.11051858&Log[Bal_Total]>=0.3822372696&INSUR(0)&Log[Bal_Total]>=1.8876899991	-1.2845528	86
Log[Bal_Total]<5.1880915172&Log[Bal_Total]>=2.11051858&Log[Bal_Total]<4.0253516907&Log[Bal_Total]<2.5800615434&Offer(0)	-1.5949743	42
Log[Bal_Total]<5.1880915172&Log[Bal_Total]>=2.11051858&Log[Bal_Total]<4.0253516907&Log[Bal_Total]<2.5800615434&Offer(1)	-1.2270595	235
Log[Bal_Total]<5.1880915172&Log[Bal_Total]>=2.11051858&Log[Bal_Total]<4.0253516907&Log[Bal_Total]>=2.5800615434&INSUR(1)	-1.1693596	237
Log[Bal_Total]<5.1880915172&Log[Bal_Total]>=2.11051858&Log[Bal_Total]<4.0253516907&Log[Bal_Total]>=2.5800615434&INSUR(0)	-0.9002	308
Log[Bal_Total]<5.1880915172&Log[Bal_Total]>=2.11051858&Log[Bal_Total]>=4.0253516907&CARD(0)&Log[Bal_Total]<4.9487598904	-0.7466805	145
Log[Bal_Total]<5.1880915172&Log[Bal_Total]>=2.11051858&Log[Bal_Total]>=4.0253516907&CARD(0)&Log[Bal_Total]>=4.9487598904	-0.2104162	41
Log[Bal_Total]<5.1880915172&Log[Bal_Total]>=2.11051858&Log[Bal_Total]>=4.0253516907&CARD(1)	-0.3276031	152
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]<10.50506754&LOAN(0)&Log[Bal_Total]<9.6924578316&INSUR(0)	-0.0101323	301
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]<10.50506754&LOAN(0)&Log[Bal_Total]<9.6924578316&INSUR(1)&Log[Bal_Total]<6.90808984&AccountAge>=1	-0.0757996	94
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]<10.50506754&LOAN(0)&Log[Bal_Total]<9.6924578316&INSUR(1)&Log[Bal_Total]<6.90808984&AccountAge<1	1.21852588	5
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]<10.50506754&LOAN(0)&Log[Bal_Total]<9.6924578316&INSUR(1)&Log[Bal_Total]>=6.90808984&MORT(1)	0.32426009	22
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]<10.50506754&LOAN(0)&Log[Bal_Total]<9.6924578316&INSUR(1)&Log[Bal_Total]>=6.90808984&MORT(0)	1.01364357	90
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]<10.50506754&LOAN(0)&Log[Bal_Total]>=9.6924578316	0.47569318	549
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]<10.50506754&LOAN(1)&Log[Bal_Total]<7.4395593091	0.14594393	58
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]<10.50506754&LOAN(1)&Log[Bal_Total]>=7.4395593091	0.95501511	180
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]>=10.50506754&INSUR(0)&Log[Bal_Total]<11.379394072	0.73077989	294
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]>=10.50506754&INSUR(0)&Log[Bal_Total]>=11.379394072	1.36078723	19
Log[Bal_Total]>=5.1880915172&CARD(0)&Log[Bal_Total]>=10.50506754&INSUR(1)	1.19119113	30
Log[Bal_Total]>=5.1880915172&CARD(1)&Log[Bal_Total]<7.0422861719&Log[Bal_Total]<5.7522035001&INSUR(1)	-0.2793776	29
Log[Bal_Total]>=5.1880915172&CARD(1)&Log[Bal_Total]<7.0422861719&Log[Bal_Total]<5.7522035001&INSUR(0)	0.23374362	101
Log[Bal_Total]>=5.1880915172&CARD(1)&Log[Bal_Total]<7.0422861719&Log[Bal_Total]>=5.7522035001	0.60835814	436
Log[Bal_Total]>=5.1880915172&CARD(1)&Log[Bal_Total]>=7.0422861719&Log[Bal_Total]<8.2485910459	1.11265784	428
Log[Bal_Total]>=5.1880915172&CARD(1)&Log[Bal_Total]>=7.0422861719&Log[Bal_Total]>=8.2485910459&Log[Bal_Total]<9.4021102371&Log[Bal_Total]>=8.5819217942	1.2983416	152
Log[Bal_Total]>=5.1880915172&CARD(1)&Log[Bal_Total]>=7.0422861719&Log[Bal_Total]>=8.2485910459&Log[Bal_Total]<9.4021102371&Log[Bal_Total]<8.5819217942	1.64300853	86
Log[Bal_Total]>=5.1880915172&CARD(1)&Log[Bal_Total]>=7.0422861719&Log[Bal_Total]>=8.2485910459&Log[Bal_Total]>=9.4021102371	1.91392036	41

Since we are dealing with the Log data, value we received from the condition must be undertaken into Antilog transformation to get the predicted revenue. The highlighted report in the leaf report can be examined, there are 5 values results at 1.2185 when 5.1881 <= Log(Bal_Total) => 6.90809 with LOAN (0) = No account activity, INSUR (1) = Higher Insurance activity and account age lesser than 1 year then Log(Rev_Total) would be around 1.2185 which is nearly $16.5386.

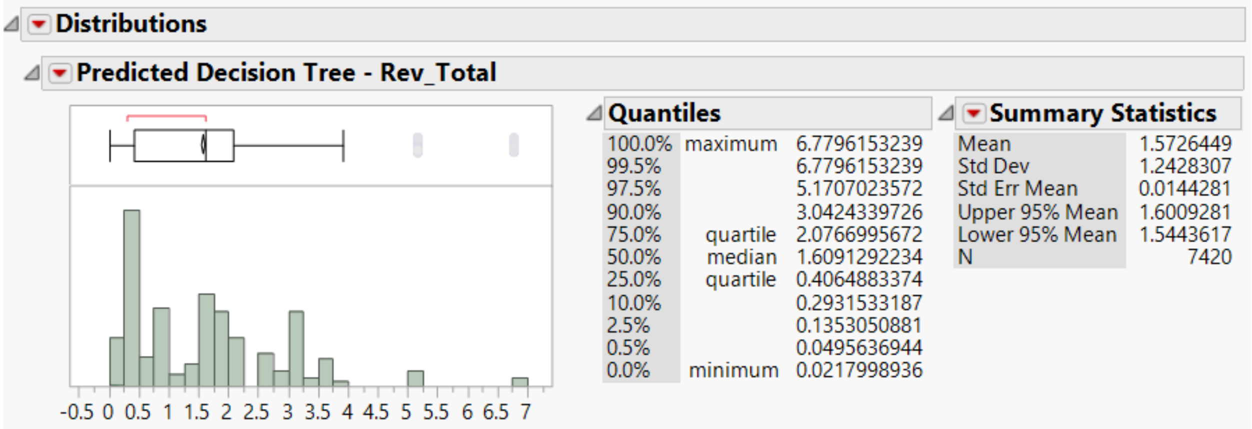


As we are dealing with nominal data, we would not get LOC curve rather we would see the prediction profiler report and how response variable Log(Rev_Total) is dependent based on the other variable responses. Since it’s a log transform we will have to save the prediction formula and then apply Transcendental option to apply the exponential operation to get the predicted saved formula. Kindly have a look at JMP file Predicted Decision Tree – Rev_Total Column.

Formula:


Exp (*Log[Rev_Total] Predictor*)

Here is the distribution for the predicted Rev_Total after Exp operation.



Distributions

Predicted Decision Tree - Rev_Total

Quantiles

100.0%	maximum	6.7796153239
99.5%		6.7796153239
97.5%		5.1707023572
90.0%		3.0424339726
75.0%	quartile	2.0766995672
50.0%	median	1.6091292234
25.0%	quartile	0.4064883374
10.0%		0.2931533187
2.5%		0.1353050881
0.5%		0.0495636944
0.0%	minimum	0.0217998936

Summary Statistics

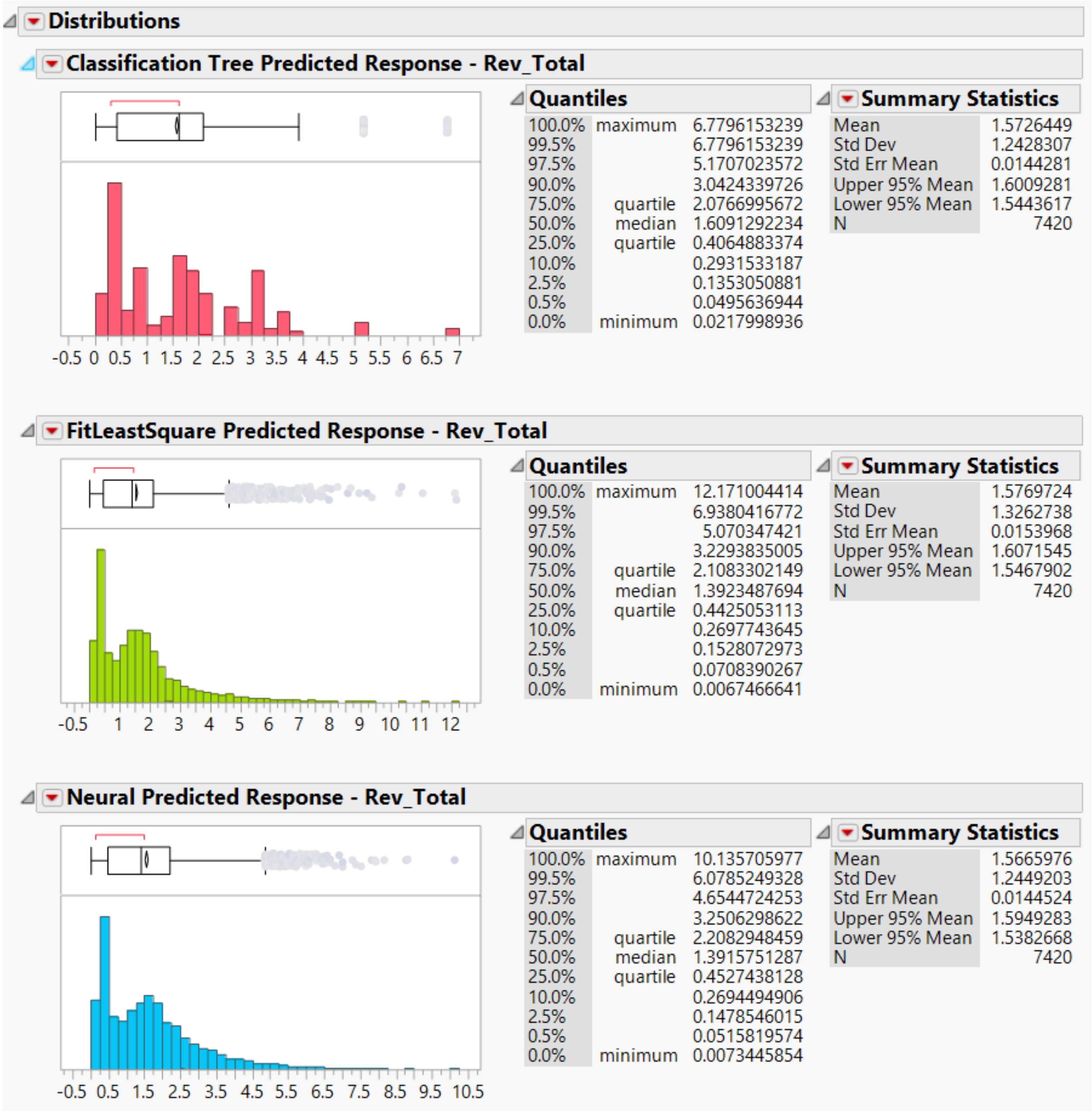
Mean	1.5726449
Std Dev	1.2428307
Std Err Mean	0.0144281
Upper 95% Mean	1.6009281
Lower 95% Mean	1.5443617
N	7420

Insights

High account balance customers and those who use their credit cards frequently generate more revenue. However, data also show us that high checking account usage seems to indicate lower revenue and that customers with higher activity on loan and insurance accounts have lower predicted revenue on average.

Models Comparison

After we convert the log data to normal revenue data applying Transcendental operation in Formula tab we get the revenue data. Once we get the revenue, we will do the distribution.



Distributions

Classification Tree Predicted Response - Rev_Total

Quantiles

100.0%	maximum	6.7796153239
99.5%		6.7796153239
97.5%		5.1707023572
90.0%		3.0424339726
75.0%	quartile	2.0766995672
50.0%	median	1.6091292234
25.0%	quartile	0.4064883374
10.0%		0.2931533187
2.5%		0.1353050881
0.5%		0.0495636944
0.0%	minimum	0.0217998936

Summary Statistics

Mean	1.5726449
Std Dev	1.2428307
Std Err Mean	0.0144281
Upper 95% Mean	1.6009281
Lower 95% Mean	1.5443617
N	7420

FitLeastSquare Predicted Response - Rev_Total

Quantiles

100.0%	maximum	12.171004414
99.5%		6.9380416772
97.5%		5.070347421
90.0%		3.2293835005
75.0%	quartile	2.1083302149
50.0%	median	1.3923487694
25.0%	quartile	0.4425053113
10.0%		0.2697743645
2.5%		0.1528072973
0.5%		0.0708390267
0.0%	minimum	0.0067466641

Summary Statistics

Mean	1.5769724
Std Dev	1.3262738
Std Err Mean	0.0153968
Upper 95% Mean	1.6071545
Lower 95% Mean	1.5467902
N	7420

Neural Predicted Response - Rev_Total

Quantiles

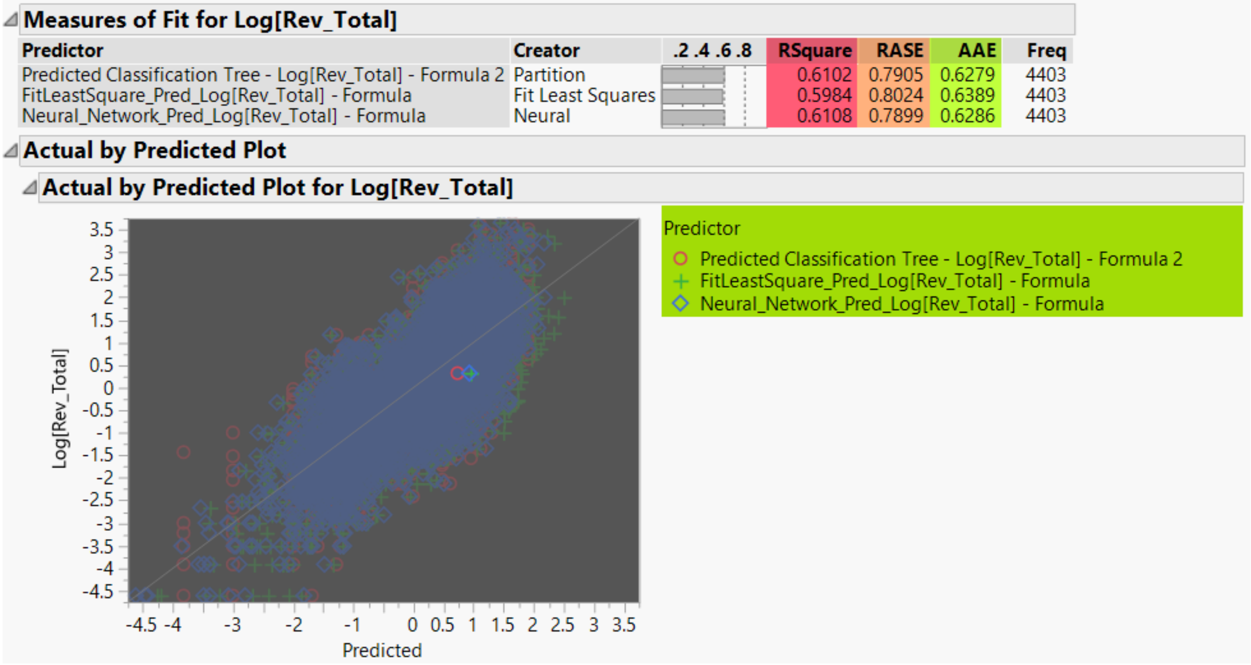
100.0%	maximum	10.135705977
99.5%		6.0785249328
97.5%		4.6544724253
90.0%		3.2506298622
75.0%	quartile	2.2082948459
50.0%	median	1.3915751287
25.0%	quartile	0.4527438128
10.0%		0.2694494906
2.5%		0.1478546015
0.5%		0.0515819574
0.0%	minimum	0.0073445854

Summary Statistics

Mean	1.5665976
Std Dev	1.2449203
Std Err Mean	0.0144524
Upper 95% Mean	1.5949283
Lower 95% Mean	1.5382668
N	7420

We can clearly infer from the above chart that Classification Tree still produced skewed data where as Fit Least Square and Neural Network gives us skewed but not highly skewed distribution of response variable.

Validation Result



Measures of Fit for Log[Rev_Total]

Predictor	Creator	.2 .4 .6 .8	RSquare	RASE	AAE	Freq
Predicted Classification Tree - Log[Rev_Total] - Formula 2	Partition		0.6102	0.7905	0.6279	4403
FitLeastSquare_Pred_Log[Rev_Total] - Formula	Fit Least Squares		0.5984	0.8024	0.6389	4403
Neural_Network_Pred_Log[Rev_Total] - Formula	Neural		0.6108	0.7899	0.6286	4403

Actual by Predicted Plot

Actual by Predicted Plot for Log[Rev_Total]

Predictor
○ Predicted Classification Tree - Log[Rev_Total] - Formula 2
+ FitLeastSquare_Pred_Log[Rev_Total] - Formula
◇ Neural_Network_Pred_Log[Rev_Total] - Formula

Observing the result, we are getting RSquare almost all near. Classification (0.6102), Fit Least Square (0.5984), and Neural Network (0.6108). Looking at the result we see there is marginal difference we are receiving in observation and Neural Network considered to be working really good in terms of getting higher RSquare and lower RASE (Root Average Square Error). AAE (Average Absolute Error) stands low for Classification Decision Tree and marginally higher for Neural Network. Actual by predicted plot gives us great understanding on how the predicted variables are distributed with the predicted responses and they standalone options on where they lie.

By the result we can say Neural Networks is giving us best results amongst all the platforms and we should proceed with the Neural Network result to predict the responses.

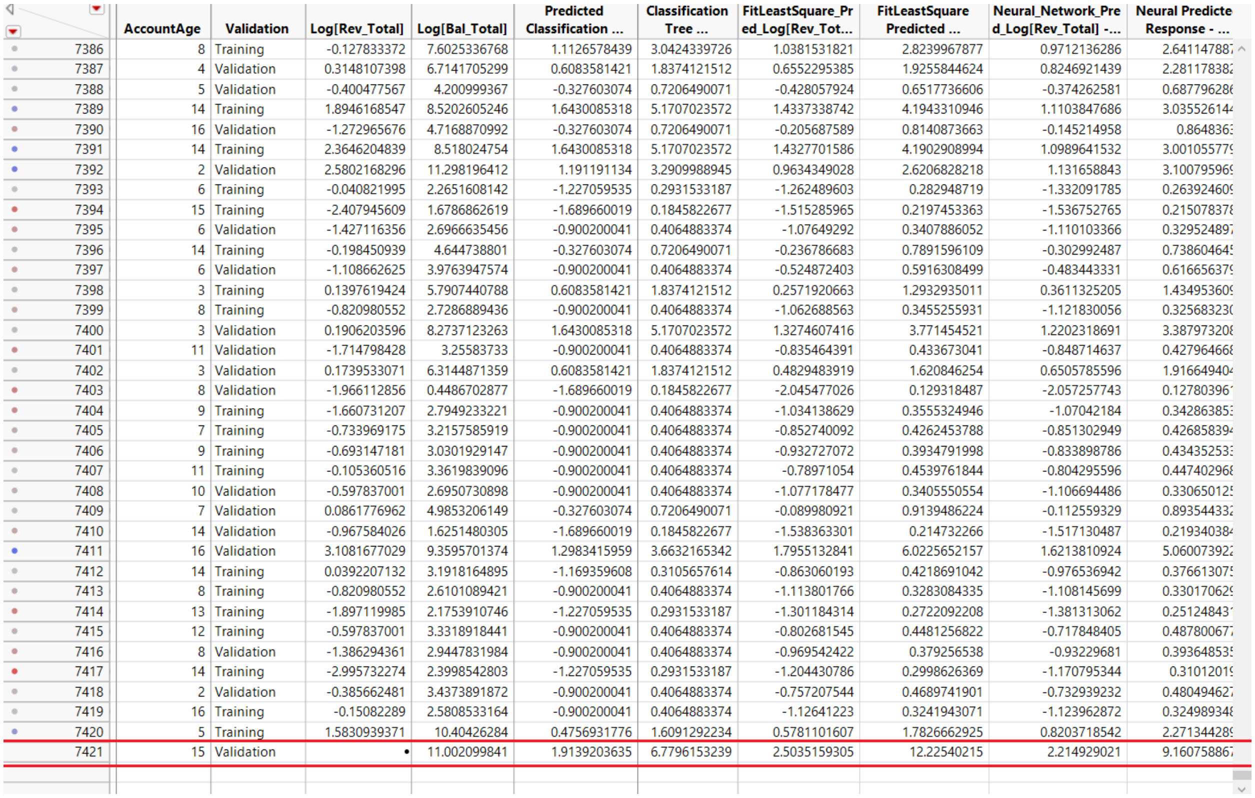
Adding Value for prediction

When we are adding test value with $60,000 Balance, Offer [1], Age: 35, CHQ [1], CARD [1], SAV1 [0], LOAN [0], MORT [1], INSUR [1], PENS [0], Check [1], CD [0], MM [0], Savings [1], and Account Age: 15 we get following results:

Classification: Predicted revenue comes to $6.779

FitLeastSquare: Predicted revenue comes to $12.225

Neural Network: Predicted revenue comes to $9.1607

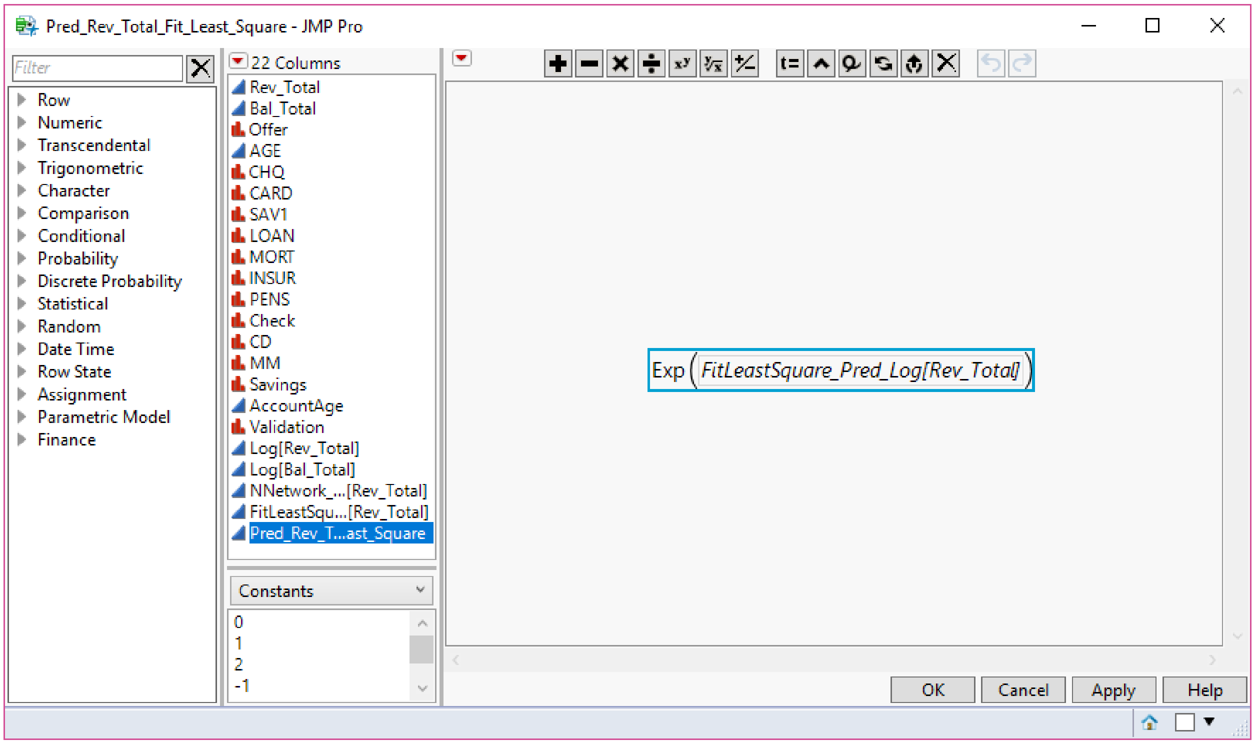


	AccountAge	Validation	Log[Rev_Total]	Log[Bal_Total]	Predicted Classification ...	Classification Tree ...	FitLeastSquare_Pred_Log[Rev_Tot...	FitLeastSquare Predicted ...	Neural_Network_Pred_Log[Rev_Total] -...	Neural Predicted Response - ...
7386	8	Training	-0.127833372	7.6025336768	1.1126578439	3.0424339726	1.0381531821	2.8239967877	0.9712136286	2.641147887
7387	4	Validation	0.3148107398	6.7141705299	0.6083581421	1.8374121512	0.6552295385	1.9255844624	0.8246921439	2.281178382
7388	5	Validation	-0.400477567	4.200999367	-0.327603074	0.7206490071	-0.428057924	0.6517736606	-0.374262581	0.687796286
7389	14	Training	1.8946168547	8.5202605246	1.6430085318	5.1707023572	1.4337338742	4.1943310946	1.1103847686	3.035526144
7390	16	Validation	-1.272965676	4.7168870992	-0.327603074	0.7206490071	-0.205687589	0.8140873663	-0.145214958	0.8648363
7391	14	Training	2.3646204839	8.518024754	1.6430085318	5.1707023572	1.4327701586	4.1902908994	1.0989641532	3.001055779
7392	2	Validation	2.5802168296	11.298196412	1.191191134	3.2909988945	0.9634349028	2.6206828218	1.131658843	3.100795969
7393	6	Training	-0.040821995	2.2651608142	-1.227059535	0.2931533187	-1.262489603	0.282948719	-1.332091785	0.263924609
7394	15	Training	-2.407945609	1.6786862619	-1.689660019	0.1845822677	-1.515285965	0.2197453363	-1.536752765	0.215078378
7395	6	Validation	-1.427116356	2.6966635456	-0.900200041	0.4064883374	-1.07649292	0.3407886052	-1.110103366	0.329524897
7396	14	Training	-0.198450939	4.644738801	-0.327603074	0.7206490071	-0.236786683	0.7891596109	-0.302992487	0.738604645
7397	6	Validation	-1.108662625	3.9763947574	-0.900200041	0.4064883374	-0.524872403	0.5916308499	-0.483443331	0.616656379
7398	3	Training	0.1397619424	5.7907440788	0.6083581421	1.8374121512	0.2571920663	1.2932935011	0.3611325205	1.434953609
7399	8	Training	-0.820980552	2.7286889436	-0.900200041	0.4064883374	-1.062688563	0.3455255931	-1.121830056	0.325683230
7400	3	Validation	0.1906203596	8.2737123263	1.6430085318	5.1707023572	1.3274607416	3.771454521	1.2202318691	3.387973208
7401	11	Validation	-1.714798428	3.25583733	-0.900200041	0.4064883374	-0.835464391	0.433673041	-0.848714637	0.427964668
7402	3	Validation	0.1739533071	6.3144871359	0.6083581421	1.8374121512	0.4829483919	1.620846254	0.6505785596	1.916649404
7403	8	Validation	-1.966112856	0.4486702877	-1.689660019	0.1845822677	-2.045477026	0.129318487	-2.057257743	0.127803961
7404	9	Training	-1.660731207	2.7949233221	-0.900200041	0.4064883374	-1.034138629	0.3555324946	-1.07042184	0.342863853
7405	7	Training	-0.733969175	3.2157585919	-0.900200041	0.4064883374	-0.852740092	0.4262453788	-0.851302949	0.426858394
7406	9	Training	-0.693147181	3.0301929147	-0.900200041	0.4064883374	-0.932727072	0.3934791998	-0.833898786	0.434352533
7407	11	Training	-0.105360516	3.3619839096	-0.900200041	0.4064883374	-0.78971054	0.4539761844	-0.804295596	0.447402968
7408	10	Validation	-0.597837001	2.6950730898	-0.900200041	0.4064883374	-1.077178477	0.3405550554	-1.106694486	0.330650125
7409	7	Validation	0.0861776962	4.9853206149	-0.327603074	0.7206490071	-0.089980921	0.9139486224	-0.112559329	0.893544332
7410	14	Validation	-0.967584026	1.6251480305	-1.689660019	0.1845822677	-1.538363301	0.214732266	-1.517130487	0.219340384
7411	16	Validation	3.1081677029	9.3595701374	1.2983415959	3.6632165342	1.7955132841	6.0225652157	1.6213810924	5.060073922
7412	14	Training	0.0392207132	3.1918164895	-1.169359608	0.3105657614	-0.863060193	0.4218691042	-0.976536942	0.376613075
7413	8	Training	-0.820980552	2.6101089421	-0.900200041	0.4064883374	-1.113801766	0.3283084335	-1.108145699	0.330170629
7414	13	Training	-1.897119985	2.1753910746	-1.227059535	0.2931533187	-1.301184314	0.2722092208	-1.381313062	0.251248431
7415	12	Training	-0.597837001	3.3318918441	-0.900200041	0.4064883374	-0.802681545	0.4481256822	-0.717848405	0.487800677
7416	8	Validation	-1.386294361	2.9447831984	-0.900200041	0.4064883374	-0.969542422	0.379256538	-0.93229681	0.393648535
7417	14	Training	-2.995732274	2.3998542803	-1.227059535	0.2931533187	-1.204430786	0.2998626369	-1.170795344	0.31012019
7418	2	Validation	-0.385662481	3.4373891872	-0.900200041	0.4064883374	-0.757207544	0.4689741901	-0.732939232	0.480494627
7419	16	Training	-0.15082289	2.5808533164	-0.900200041	0.4064883374	-1.12641223	0.3241943071	-1.123962872	0.324989348
7420	5	Training	1.5830939371	10.40426284	0.4756931776	1.6091292234	0.5781101607	1.7826662925	0.8203718542	2.271344289
7421	15	Validation	•	11.002099841	1.9139203635	6.7796153239	2.5035159305	12.22540215	2.214929021	9.160758867

From Log[Rev_Total] to Rev_Total

To show the predicted values for each bank customer, we save the prediction formula to the data table for each model. However, one may wonder that these are the log predicted values, which are difficult to interpret and what if we want to know the exact number of Rev_Total?

In this case, the inverse transformation (Exp function) can be used to examine predicted values on the original scale. For example, to apply the transformation to the Multi Linear Regression Model's results, we create a new column in the data table (we've named this column Pred_Rev_Total_Fit_Least_Square). Then, right-click on the column and select Formula to open the Formula Editor, and use the Transcendental > Exp function from the Functions list, and add in the Results of Log[Rev_Total]. Similar process can be applied for other models to obtain predicted Rev_Total from predicted Log[Rev_Total]



Pred_Rev_Total_Fit_Least_Square - JMP Pro

Exp(FitLeastSquare_Pred_Log[Rev_Total])

Conclusion

We have applied Multiple Linear Regression Model with Log Transformation model, Neural Networks with Log Transformation model, and Classification Tree through our prediction. We got very close Rsquare values from the three models we built. From our findings, we found that Multiple Linear Regression Model with Log Transformation model can give us accurate prediction and does not run any chances of overfitting, but it has relatively lower Rsquare value and higher RMSE value than another two models. Also, we have applied Classification Tree. The tree diagram has given us a good understanding of at what value split has occurred. And, the leaf report for responses we get all the possible information with the summary of splits and an easy way of reading tree based on the predictors. By reading leaf report we can provide end-users with rules for classifying outcomes together. When comparing with Neutral Network model, the Classification Tree has relatively poor performance on validation data. Based all our comparisons, Neutral Network model works best among three model on validation data. Hence, Neutral Network Model is the most accurate in predicting total bank revenues.

Moreover, we initially built the models from the raw dataset and had very poor models with very low RSquare (<0.3). After the log transformation, we could obtain much better models with reasonable RSquare and RMSE values. Therefore, we realized that in highly skewed dataset, data transformation is very essential to obtain accurate predicting models. JMP offers many methods to transform data that we can explore further in the future. Data examining is a vital step in building successful models.

Bibliography

Bank Revenues. (n.d.). Retrieved from https://www.jmp.com/content/dam/jmp/documents/en/academic/case-study-library/case-study-library-12/analytics-cases/mlr-bankrevenues.pdf

Case Study Library. (n.d.). Retrieved March 13, 2018, from https://www.jmp.com/en_us/academic/case-study-library.html#bank

Appendix 1: Variable Names and Descriptions

Rev_Total	Total revenue generated by the customer over a 6-month period.
Bal_Tota	Total of all account balances, across all accounts held by the customer.
Offer	An indicator of whether the customer has received a special promotional offer in the previous one-month period. Offer=1 if the offer was received, Offer=0 if it was not.
AGE	The customer's age.
CHQ	Indicator of debit card account activity. CHQ=0 is low (or zero) account activity, CHQ=1 is greater account activity.
CARD	Indicator of credit card account activity. CARD=0 is low or zero account activity, CARD=1 is greater account activity.
SAV1	Indicator of primary savings account activity. SAV1=0 is low or zero account activity, SAV1=1 is greater activity.

LOAN	Indicator of personal loan account activity. LOAN=0 is low or zero account activity, LOAN=1 is greater activity
MORT	Indicator of mortgage account tier. MORT=0 is lower tier and less important to the bank's portfolio. MORT=1 is higher tier and indicates the account is more important to the bank's portfolio.
INSUR	Indicator of insurance account activity. INSUR=0 is low or zero account activity, INSUR=1 is greater activity.
PENS	Indicator or retirement savings (pension) account tier. PENS=0 is lower balance and less important to bank's portfolio. PENS=1 is higher tier and of more importance to the bank's portfolio.
Check	Indicator of checking account activity. Check=0 is low or zero account activity, Check=1 is greater activity.
CD	Indicator of certificate of deposit account tier. CD=0 is lower tier and of less importance to the bank's portfolio. CD=1 is higher tier and of more importance to the bank's portfolio.
MM	Indicator of money market account activity. MM=0 is low or zero account activity, MM=1 is greater activity.

Savings	Indicator of savings accounts (other than primary) activity. Savings=0 is low or zero account activity, Savings=1 is greater activity.
AccountAge	Number of years as a customer of the bank.