BIG DATA

FINAL PROJECT

Pei-Chun Lin

Sirina Chen

Wei-Cheng Wang

Xinxin Wang

TABLE OF CONTENTS

DATASET OVERVIEW 3

PROJECT FLOW 3

DATA PREPROCESSING: (IN HIVE) 4

DECISION TREE 7

Model Adjustment (Leaf, Depth) 9

Model Adjustment (Input restriction) 10

Estimate of the selected predicted model 10

CORRELATION ANALYSIS 14

LOGISTIC REGRESSION 15

Scoring Table 15

Model Estimation 18

APPENDIX 19

SAS Code for Decision Tree (highlight) 19

SAS code for correlation (highlight) 22

SAS code for logistic regression (highlight) 22

Dataset Overview

Analyzing the provided financial attributes in each observed company to create predictive model (Decision Tree, Logistic Regression) of the probability in prospect companies being bankrupt or non-bankrupt. No data cleaning process required in this dataset, since no missing value included. As for nonbankrupt firms there are 11851 observations, whereas for bankrupt firms there are only 363 observations. To balance the two datasets and create a more accurate prediction model, oversampling technique has been applied to multiply the 363 observations in bankrupt firms to 11947 observations in total with approximately 1-3% increase or decrease in each duplication by using Excel.

Project flow

Oversampling - Excel

Big Data Processing (Hive)

1. Temporary table
2. Insert data into the temporary table
3. Create data table
4. Insert data into table from temporary table

Big Data Testing and Training Dataset (Hive)

- Testing – 20%
- Training 1 – 27%
- Training 2 – 27%
- Training 3 – 27%

Decision Tree

1. Import Training dataset (assigned target and input variables).
2. Compare models with and without principal components.
3. Select the decision tree with highest true predictive rate.
4. Model adjustment.
5. Access the predictive model on testing data.

Logistic Regression

1. Run logistic regression on training data 1, 2 and 3.
2. Identified significant variables in each model.
3. Score those significant variables.
4. Create new predictive model based on score chart.
5. Run the new model on testing data.
6. Evaluate the predictive model.

Data Preprocessing: (In Hive)

1. Create database:

```
CREATE DATABASE final_project;
```

2. Create temporary tables:

```
CREATE TABLE bdata_m0
     (ID         STRING,
      cf_td      FLOAT,
      ca_cl      FLOAT,
      re_ta      FLOAT,
      ni_ta      FLOAT,
      td_ta      FLOAT,
      s_ta       FLOAT,
      wc_ta      FLOAT,
      wc_s       FLOAT,
      c_cl       FLOAT,
      cl_e       FLOAT,
      in_s       FLOAT,
      mve_td     FLOAT,
      bstatus    STRING
     )
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE;

CREATE TABLE bdata_m1
     (ID         STRING,
      cf_td      FLOAT,
      ca_cl      FLOAT,
      re_ta      FLOAT,
      ni_ta      FLOAT,
      td_ta      FLOAT,
      s_ta       FLOAT,
      wc_ta      FLOAT,
      wc_s       FLOAT,
      c_cl       FLOAT,
      cl_e       FLOAT,
      in_s       FLOAT,
      mve_td     FLOAT,
      bstatus    STRING
     )
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE;
```

3. Load data from text files to temp tables:

```
LOAD DATA INPATH '/final/bdata-m1-bank.txt' OVERWRITE INTO TABLE
bdata_m0;
LOAD DATA INPATH '/final/bdata-m1-nonbank.txt' OVERWRITE INTO TABLE
bdata_m1;
```

4. Create regular tables in the 'RCFILE' format:

```
CREATE TABLE m0
     (ID         STRING,
      cf_td      FLOAT,
      ca_cl      FLOAT,
      re_ta      FLOAT,
      ni_ta      FLOAT,
      td_ta      FLOAT,
      s_ta       FLOAT,
      wc_ta      FLOAT,
      wc_s       FLOAT,
      c_cl       FLOAT,
      cl_e       FLOAT,
      in_s       FLOAT,
      mve_td     FLOAT,
      bstatus    STRING
     )
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS RCFILE
TBLPROPERTIES ("skip.header.line.count"="1");

CREATE TABLE m1
     (ID         STRING,
      cf_td      FLOAT,
      ca_cl      FLOAT,
      re_ta      FLOAT,
      ni_ta      FLOAT,
      td_ta      FLOAT,
      s_ta       FLOAT,
      wc_ta      FLOAT,
      wc_s       FLOAT,
      c_cl       FLOAT,
      cl_e       FLOAT,
      in_s       FLOAT,
      mve_td     FLOAT,
      bstatus    STRING
     )
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS RCFILE
TBLPROPERTIES ("skip.header.line.count"="1");
```

5. Load data from temporary (bdadta_m0, bdata_m1) tables to the RCFILE – format tables:

```
INSERT OVERWRITE TABLE m0
SELECT ID, cf_td, ca_cl, re_ta, ni_ta, td_ta, s_ta, wc_ta, wc_s, c_cl,
cl_e, in_s, mve_td, bstatus
FROM bdata_m0;

INSERT OVERWRITE TABLE m1
SELECT ID, cf_td, ca_cl, re_ta, ni_ta, td_ta, s_ta, wc_ta, wc_s, c_cl,
cl_e, in_s, mve_td, bstatus
FROM bdata_m1;
```

6. Splitting Data:
   - Note: the process of assigning random numbers to each observation is not used in this process, due to the fact that the dataset is ordered by company sequences and not any value order which will affect the reliability of the prediction model.

Dataset (bankrupt - 11947 obs, nonbankrupt - 11851 obs)	**Sample size** (% in both m0 and m1 datasets)	
Testing	20%	
Training_1	80%	~ 27%
Training_2		~ 27%
Training_3		~ 27%

Testing:

```
SELECT *
FROM m0
WHERE m0.ID BETWEEN 1 AND 71

UNION ALL

SELECT *
FROM m1
WHERE m1.id BETWEEN 363 AND 2703;
```

Training 1:

```
SELECT *
FROM m0
WHERE m0.ID BETWEEN 72 AND 168

UNION ALL

SELECT *
FROM m1
WHERE m1.id BETWEEN 2704 AND 5873;
```

Training 2:

```
SELECT *
FROM m0
WHERE m0.ID BETWEEN 169 AND 265

UNION ALL

SELECT *
FROM m1
WHERE m1.id BETWEEN 5874 AND 9043;
```

Training 3:

```
SELECT *
FROM m0
WHERE m0.ID BETWEEN 265 AND 363

UNION ALL

SELECT *
FROM m1
WHERE m1.id BETWEEN 9044 AND 12212;
```

Decision Tree

Name	Role	Level
bstatus	**Target**	**Binary**
ca_cl	**Input**	**Interval**
cf_td	**Input**	**Interval**
cl_e	**Input**	**Interval**
c_cl	**Input**	**Interval**
ID	**ID**	**Nominal**
in_s	**Input**	**Interval**
mve_td	**Input**	**Interval**
ni_ta	**Input**	**Interval**
re_ta	**Input**	**Interval**
s_ta	**Input**	**Interval**
td_ta	**Input**	**Interval**
wc_s	**Input**	**Interval**
wc_ta	**Input**	**Interval**

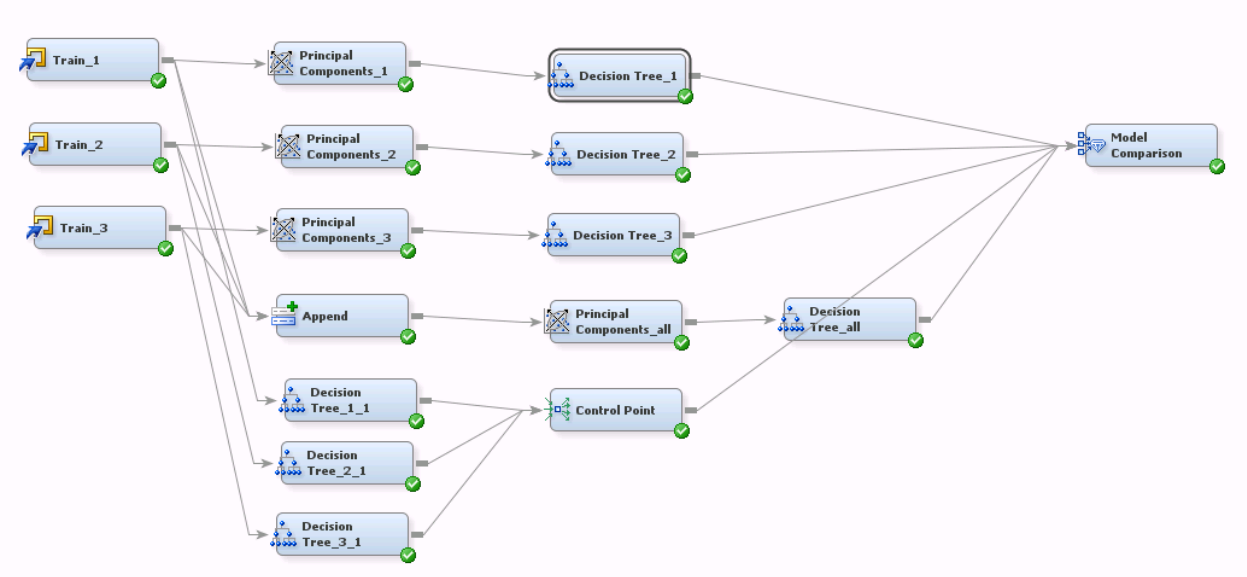
Before analyzing the data, we imported the data and assigned bank status as target variables, ID as ID and the rest is input variables.

After trying several approaches of creating a decision tree, we found that even that we only select those eigenvalues which are greater than 1, adding principal components doesn't seems to increase the accuracy of the prediction. In this case, we decide to create decision tree by inputting those original attributes.

Three approaches are applied while training the optimal decision trees:

- Use principal components analysis
- Add training 1, 2 and 3 into one file to create a decision tree model
- Run decision tree without any pre-analytical processing

Diagram please see below:



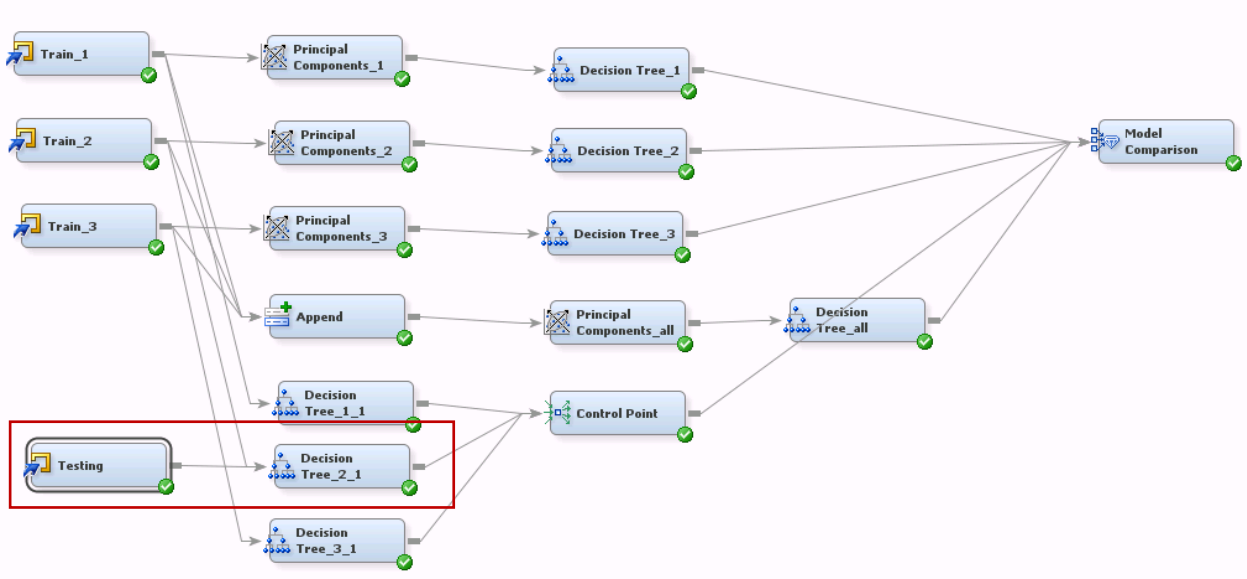
Fit Statistics

Model Selection based on Train: Misclassification Rate (_MISC_)

Selected Model	Model Node	Model Description	Train: Misclassification Rate	Train: Average Squared Error	Train: Roc Index	Train: Gini Coefficient
Y	Tree6	Decision Tree_2_1	0.12046	0.09696	0.918	0.836
	Tree5	Decision Tree_3_1	0.14315	0.10633	0.915	0.829
	Tree7	Decision Tree_1_1	0.15947	0.11466	0.901	0.803
	Tree3	Decision Tree_3	0.17577	0.13324	0.875	0.749
	Tree2	Decision Tree_2	0.18203	0.13428	0.872	0.744
	Tree	Decision Tree_1	0.20236	0.14875	0.848	0.696
	Tree4	Decision Tree_all	0.22103	0.15480	0.844	0.687

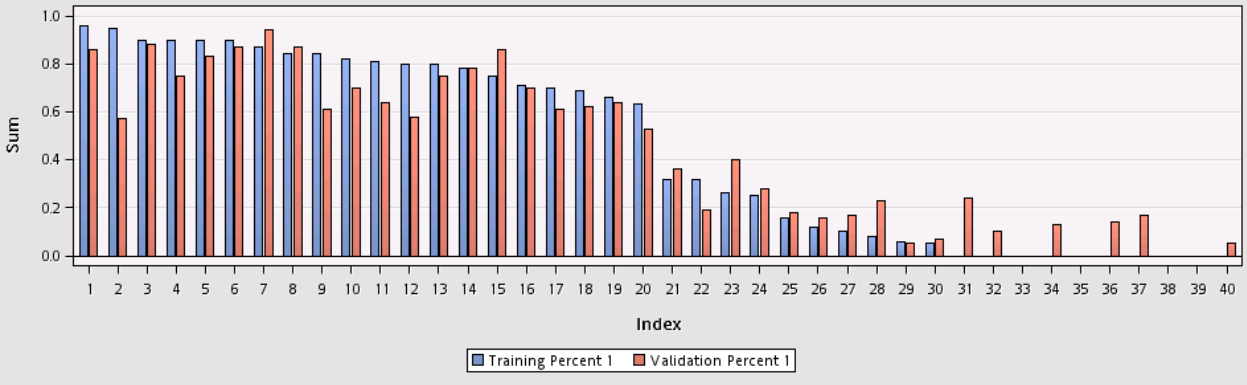
In comparison of the same leaf size (=10), depth (=6) and the significant at 5% level, the fit statistics in the comparison node shows that Decision Tree 2_1 has the lowest ASE (=0.097). As result, model which generate from Decision Tree 2_1 will be used as our predictive model for testing data.

To validate the prediction rate of the model chosen in the earlier step, we set our testing dataset as validation data in this process.

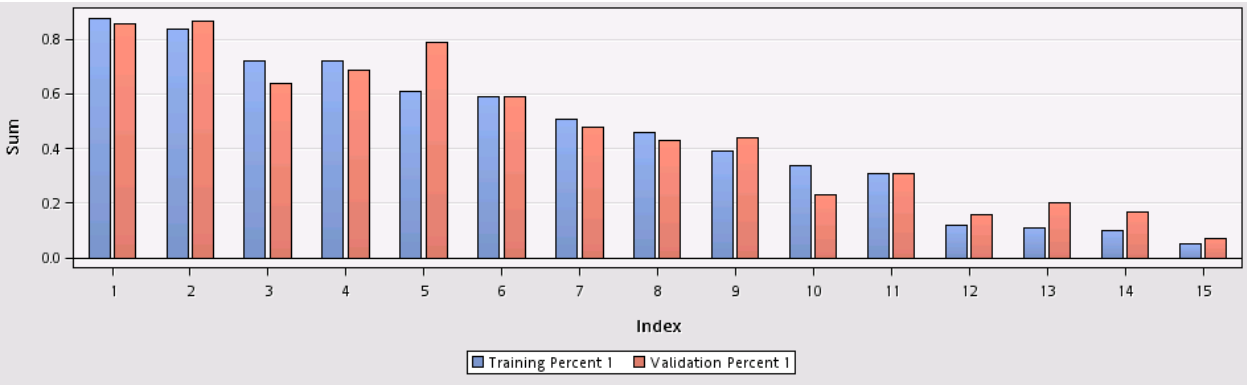


❖ Model Adjustment (Leaf, Depth)

By running at the original model which is leaf size (=10) and depth (=6), in the leaf statistic, it is shown that the node is clean at 30.



Thought the average squared error increase from 0.1 to 0.13 in the training dataset while adjusting the leaf size to 5 and depth to 50, the increment is low and the accuracy rate on prediction is still high.



❖ Model Adjustment (Input restriction)

By looking at the discriminating variables, if we didn't restrict the input to use only once in the decision tree, only three variables, cf_td, mve_td and wc_s, are apply to split the entire group. However, if we restrict the input to only can be used in the decision tree for once, four variables, cf_td, mve_td, wc_s and c_cl, are used to split the largest group in the data. The final ASE and the true predictive rate between two groups are similar, but the one without restriction still provide us a slightly better result in splitting bankrupt and nonbankrupt company. Both trees are shown in following page. The below statistic result is based on our better tree, which is the one without the time of restriction input.

❖ Estimate of the selected predicted model

According to the output, it's good to know that cf_td (Cash Flow/Total Debt) and mvw_td (Market Value of Equity/Total Debt) stand the top 2 importance variables while estimating the training and testing datasets.

Variable Importance

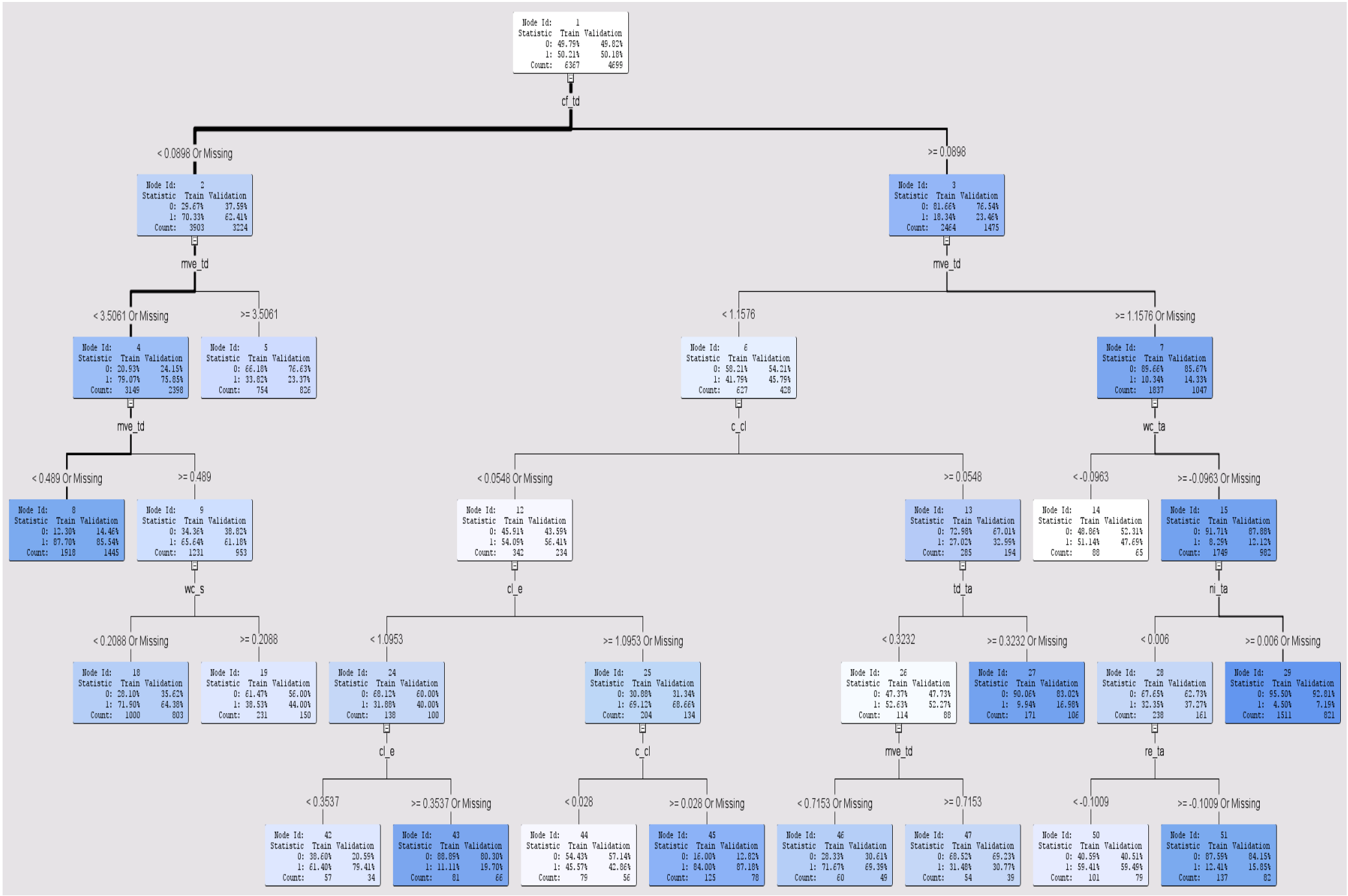
Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
cf_td		1	1.0000	0.7242	0.7242
mve_td		4	0.7204	1.0000	1.3881
wc_s		1	0.2263	0.0936	0.4138
cl_e		2	0.2206	0.2134	0.9671
c_cl		2	0.2131	0.2203	1.0334
ni_ta		1	0.1977	0.2263	1.1447
wc_ta		1	0.1941	0.1785	0.9194
re_ta		1	0.1774	0.1795	1.0122
td_ta		1	0.1747	0.1579	0.9037

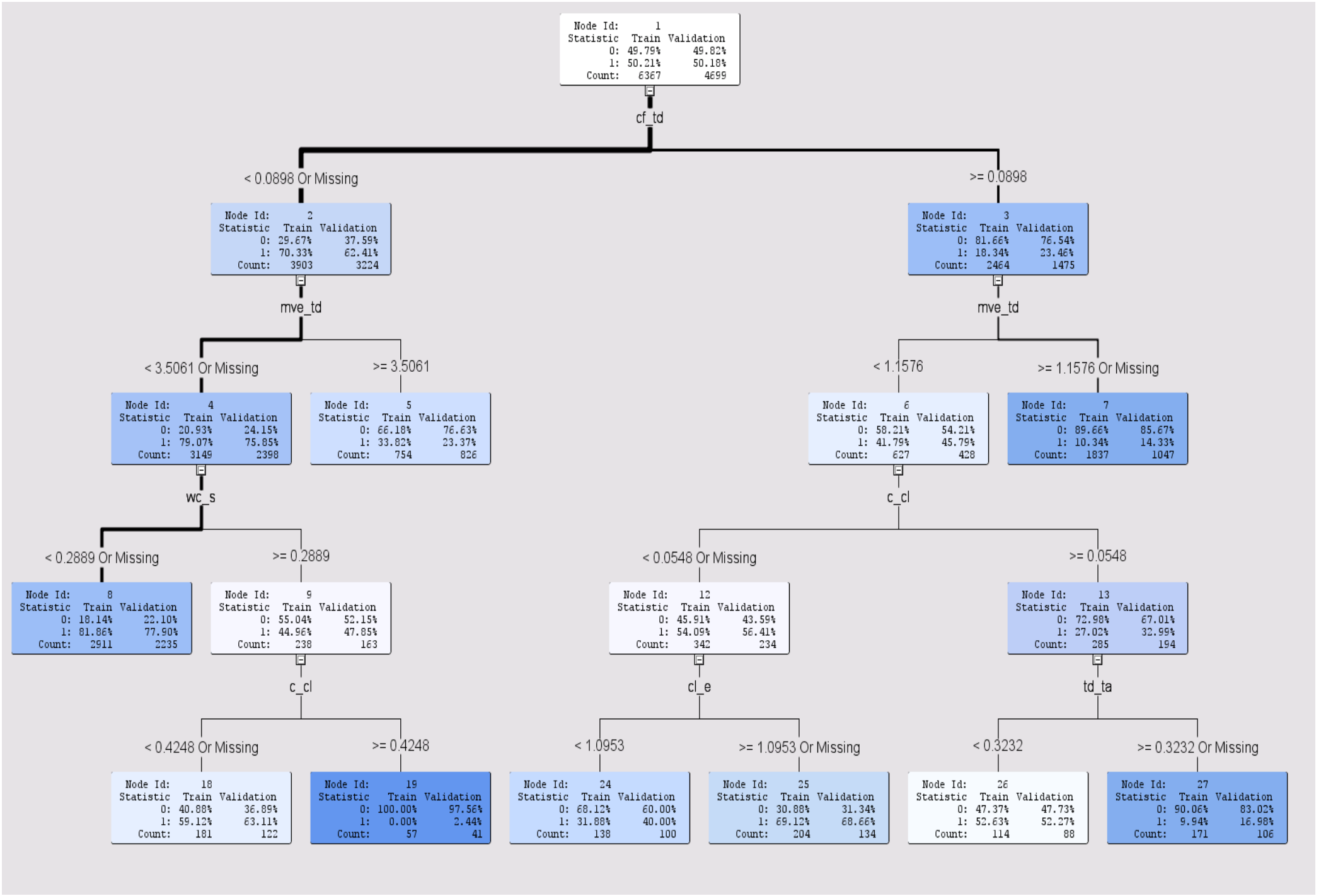
Generally speaking, the predictive model performs well in both outcome from training dataset and testing dataset. The average squared error in testing dataset is only 3% lower than the training data.

Fit Statistics

Target=bstatus Target Label=' '

Fit Statistics	Statistics Label	Train	Validation
NOBS	Sum of Frequencies	6367.00	4699.00
MISC	Misclassification Rate	0.18	0.21
MAX	Maximum Absolute Error	0.95	0.95
SSE	Sum of Squared Errors	1711.38	1477.06
ASE	Average Squared Error	0.13	0.16
RASE	Root Average Squared Error	0.37	0.40
DIV	Divisor for ASE	12734.00	9398.00
DFT	Total Degrees of Freedom	6367.00	.


Node Id: 1
Statistic Train Validation
0: 49.79% 49.82%
1: 50.21% 50.18%
Count: 6367 4699
cf_td
< 0.0898 Or Missing
>= 0.0898
Node Id: 2
Statistic Train Validation
0: 29.67% 37.59%
1: 70.33% 62.41%
Count: 3903 3224
Node Id: 3
Statistic Train Validation
0: 81.66% 76.54%
1: 18.34% 23.46%
Count: 2464 1475
mve_td
mve_td
< 3.5061 Or Missing
>= 3.5061
< 1.1576
>= 1.1576 Or Missing
Node Id: 4
Statistic Train Validation
0: 20.93% 24.15%
1: 79.07% 75.85%
Count: 3149 2398
Node Id: 5
Statistic Train Validation
0: 66.18% 76.63%
1: 33.82% 23.37%
Count: 754 826
Node Id: 6
Statistic Train Validation
0: 58.21% 54.21%
1: 41.79% 45.79%
Count: 627 428
Node Id: 7
Statistic Train Validation
0: 89.66% 85.67%
1: 10.34% 14.33%
Count: 1837 1047
mve_td
c_cl
wc_ta
< 0.489 Or Missing
>= 0.489
< 0.0548 Or Missing
>= 0.0548
< -0.0963
>= -0.0963 Or Missing
Node Id: 8
Statistic Train Validation
0: 12.30% 14.46%
1: 87.70% 85.54%
Count: 1918 1445
Node Id: 9
Statistic Train Validation
0: 34.36% 38.82%
1: 65.64% 61.18%
Count: 1231 953
Node Id: 12
Statistic Train Validation
0: 45.91% 43.59%
1: 54.09% 56.41%
Count: 342 234
Node Id: 13
Statistic Train Validation
0: 72.98% 67.01%
1: 27.02% 32.99%
Count: 285 194
Node Id: 14
Statistic Train Validation
0: 48.86% 52.31%
1: 51.14% 47.69%
Count: 88 65
Node Id: 15
Statistic Train Validation
0: 91.71% 87.88%
1: 8.29% 12.12%
Count: 1749 982
wc_s
cl_e
td_ta
ni_ta
< 0.2088 Or Missing
>= 0.2088
< 1.0953
>= 1.0953 Or Missing
< 0.3232
>= 0.3232 Or Missing
< 0.006
>= 0.006 Or Missing
Node Id: 18
Statistic Train Validation
0: 28.10% 35.62%
1: 71.90% 64.38%
Count: 1000 803
Node Id: 19
Statistic Train Validation
0: 61.47% 56.00%
1: 38.53% 44.00%
Count: 231 150
Node Id: 24
Statistic Train Validation
0: 68.12% 60.00%
1: 31.88% 40.00%
Count: 138 100
Node Id: 25
Statistic Train Validation
0: 30.88% 31.34%
1: 69.12% 68.66%
Count: 204 134
Node Id: 26
Statistic Train Validation
0: 47.37% 47.73%
1: 52.63% 52.27%
Count: 114 88
Node Id: 27
Statistic Train Validation
0: 90.06% 83.02%
1: 9.94% 16.98%
Count: 171 106
Node Id: 28
Statistic Train Validation
0: 67.65% 62.73%
1: 32.35% 37.27%
Count: 238 161
Node Id: 29
Statistic Train Validation
0: 95.50% 92.81%
1: 4.50% 7.19%
Count: 1511 821
cl_e
c_cl
mve_td
re_ta
< 0.3537
>= 0.3537 Or Missing
< 0.028
>= 0.028 Or Missing
< 0.7153 Or Missing
>= 0.7153
< -0.1009
>= -0.1009 Or Missing
Node Id: 42
Statistic Train Validation
0: 38.60% 20.59%
1: 61.40% 79.41%
Count: 57 34
Node Id: 43
Statistic Train Validation
0: 88.89% 80.30%
1: 11.11% 19.70%
Count: 81 66
Node Id: 44
Statistic Train Validation
0: 54.43% 57.14%
1: 45.57% 42.86%
Count: 79 56
Node Id: 45
Statistic Train Validation
0: 16.00% 12.82%
1: 84.00% 87.18%
Count: 125 78
Node Id: 46
Statistic Train Validation
0: 28.33% 30.61%
1: 71.67% 69.39%
Count: 60 49
Node Id: 47
Statistic Train Validation
0: 68.52% 69.23%
1: 31.48% 30.77%
Count: 54 39
Node Id: 50
Statistic Train Validation
0: 40.59% 40.51%
1: 59.41% 59.49%
Count: 101 79
Node Id: 51
Statistic Train Validation
0: 87.59% 84.15%
1: 12.41% 15.85%
Count: 137 82


Node Id: 1
Statistic Train Validation
0: 49.79% 49.82%
1: 50.21% 50.18%
Count: 6367 4699
cf_td
< 0.0898 Or Missing
>= 0.0898
Node Id: 2
Statistic Train Validation
0: 29.67% 37.59%
1: 70.33% 62.41%
Count: 3903 3224
Node Id: 3
Statistic Train Validation
0: 81.66% 76.54%
1: 18.34% 23.46%
Count: 2464 1475
mve_td
mve_td
< 3.5061 Or Missing
>= 3.5061
< 1.1576
>= 1.1576 Or Missing
Node Id: 4
Statistic Train Validation
0: 20.93% 24.15%
1: 79.07% 75.85%
Count: 3149 2398
Node Id: 5
Statistic Train Validation
0: 66.18% 76.63%
1: 33.82% 23.37%
Count: 754 826
Node Id: 6
Statistic Train Validation
0: 58.21% 54.21%
1: 41.79% 45.79%
Count: 627 428
Node Id: 7
Statistic Train Validation
0: 89.66% 85.67%
1: 10.34% 14.33%
Count: 1837 1047
wc_s
c_cl
< 0.2889 Or Missing
>= 0.2889
< 0.0548 Or Missing
>= 0.0548
Node Id: 8
Statistic Train Validation
0: 18.14% 22.10%
1: 81.86% 77.90%
Count: 2911 2235
Node Id: 9
Statistic Train Validation
0: 55.04% 52.15%
1: 44.96% 47.85%
Count: 238 163
Node Id: 12
Statistic Train Validation
0: 45.91% 43.59%
1: 54.09% 56.41%
Count: 342 234
Node Id: 13
Statistic Train Validation
0: 72.98% 67.01%
1: 27.02% 32.99%
Count: 285 194
c_cl
cl_e
td_ta
< 0.4248 Or Missing
>= 0.4248
< 1.0953
>= 1.0953 Or Missing
< 0.3232
>= 0.3232 Or Missing
Node Id: 18
Statistic Train Validation
0: 40.88% 36.89%
1: 59.12% 63.11%
Count: 181 122
Node Id: 19
Statistic Train Validation
0: 100.00% 97.56%
1: 0.00% 2.44%
Count: 57 41
Node Id: 24
Statistic Train Validation
0: 68.12% 60.00%
1: 31.88% 40.00%
Count: 138 100
Node Id: 25
Statistic Train Validation
0: 30.88% 31.34%
1: 69.12% 68.66%
Count: 204 134
Node Id: 26
Statistic Train Validation
0: 47.37% 47.73%
1: 52.63% 52.27%
Count: 114 88
Node Id: 27
Statistic Train Validation
0: 90.06% 83.02%
1: 9.94% 16.98%
Count: 171 106

The true prediction rates in both train and testing dataset are 81.6554 and 78.9104 respectively.

Classification Table

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

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	83.1677	79.1798	2510	39.4220
1	0	16.8323	15.8899	508	7.9786
0	1	19.7074	20.8202	660	10.3659
1	1	80.2926	84.1101	2689	42.2334

Data Role=VALIDATE Target Variable=bstatus Target Label=' '

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	81.4539	74.6689	1748	37.1994
1	0	18.5461	16.8787	398	8.4699
0	1	23.2276	25.3311	593	12.6197
1	1	76.7724	83.1213	1960	41.7110

Event Classification Table

Data Role=TRAIN Target=bstatus Target Label=' '

False Negative	True Negative	False Positive	True Positive
508	2510	660	2689

Data Role=VALIDATE Target=bstatus Target Label=' '

False Negative	True Negative	False Positive	True Positive
398	1748	593	1960

Correlation Analysis

By running this analytical method, we will be able to identify whether any multicollinearity problem is going to happen in the predictive model. However, in the result, statistically there is no highly correlated relationship between each of the variables. The highest significant correlation coefficients is 0.8135 between ID and banking status (bstatus), however, ID will not be used in the model.

Pearson Correlation Coefficients, N = 23796
Prob > |r| under H0: Rho=0

	ID	cf_td	ca_cl	re_ta	ni_ta	td_ta	s_ta	wc_ta	wc_s	c_cl	cl_e	in_s	mve_td	bstatus
ID	1.00000	-0.00327 0.6142	-0.19989 <.0001	0.00212 0.7433	-0.06670 <.0001	0.22873 <.0001	0.10256 <.0001	-0.17010 <.0001	-0.03537 <.0001	-0.16296 <.0001	-0.00788 0.2241	-0.03314 <.0001	-0.03167 <.0001	0.81353 <.0001
cf_td	-0.00327 0.6142	1.00000	-0.01125 0.0828	0.03611 <.0001	0.06320 <.0001	0.00100 0.8770	0.02777 <.0001	0.00506 0.4352	-0.01121 0.0836	-0.01546 0.0171	-0.00002 0.9972	-0.00864 0.1826	0.67747 <.0001	-0.00788 0.2239
ca_cl	-0.19989 <.0001	-0.01125 0.0828	1.00000	0.08102 <.0001	0.08676 <.0001	-0.24080 <.0001	-0.13090 <.0001	0.30934 <.0001	0.15218 <.0001	0.68768 <.0001	-0.00630 0.3315	0.15767 <.0001	0.04693 <.0001	-0.23079 <.0001
re_ta	0.00212 0.7433	0.03611 <.0001	0.08102 <.0001	1.00000	0.55848 <.0001	-0.43687 <.0001	-0.04213 <.0001	0.57580 <.0001	0.01425 0.0279	0.01181 0.0684	0.00575 0.3752	-0.02390 0.0002	0.00114 0.8599	-0.02187 0.0007
ni_ta	-0.06670 <.0001	0.06320 <.0001	0.08676 <.0001	0.55848 <.0001	1.00000	-0.31118 <.0001	-0.06465 <.0001	0.41809 <.0001	0.01496 0.0210	0.01681 0.0095	0.01468 0.0236	-0.02102 0.0012	0.00585 0.3665	-0.11813 <.0001
td_ta	0.22873 <.0001	0.00100 0.8770	-0.24080 <.0001	-0.43687 <.0001	-0.31118 <.0001	1.00000	0.02581 <.0001	-0.71320 <.0001	-0.05719 <.0001	-0.14203 <.0001	-0.01087 0.0935	-0.00404 0.5332	-0.03452 <.0001	0.28898 <.0001
s_ta	0.10256 <.0001	0.02777 <.0001	-0.13090 <.0001	-0.04213 <.0001	-0.06465 <.0001	0.02581 <.0001	1.00000	-0.07418 <.0001	-0.03201 <.0001	-0.14383 <.0001	-0.05088 <.0001	-0.06492 <.0001	-0.00577 0.3735	0.11370 <.0001
wc_ta	-0.17010 <.0001	0.00506 0.4352	0.30934 <.0001	0.57580 <.0001	0.41809 <.0001	-0.71320 <.0001	-0.07418 <.0001	1.00000	0.08404 <.0001	0.15878 <.0001	-0.01802 0.0055	0.02579 <.0001	0.02566 <.0001	-0.22359 <.0001
wc_s	-0.03537 <.0001	-0.01121 0.0836	0.15218 <.0001	0.01425 0.0279	0.01496 0.0210	-0.05719 <.0001	-0.03201 <.0001	0.08404 <.0001	1.00000	0.18322 <.0001	-0.00107 0.8687	0.23953 <.0001	0.00769 0.2358	-0.03490 <.0001
c_cl	-0.16296 <.0001	-0.01546 0.0171	0.68768 <.0001	0.01181 0.0684	0.01681 0.0095	-0.14203 <.0001	-0.14383 <.0001	0.15878 <.0001	0.18322 <.0001	1.00000	0.00118 0.8559	0.03314 <.0001	0.04256 <.0001	-0.17176 <.0001
cl_e	-0.00788 0.2241	-0.00002 0.9972	-0.00630 0.3315	0.00575 0.3752	0.01468 0.0236	-0.01087 0.0935	-0.05088 <.0001	-0.01802 0.0055	-0.00107 0.8687	0.00118 0.8559	1.00000	-0.00199 0.7594	0.00032 0.9600	-0.01077 0.0965
in_s	-0.03314 <.0001	-0.00864 0.1826	0.15767 <.0001	-0.02390 0.0002	-0.02102 0.0012	-0.00404 0.5332	-0.06492 <.0001	0.02579 <.0001	0.23953 <.0001	0.03314 <.0001	-0.00199 0.7594	1.00000	0.00127 0.8452	-0.02615 <.0001
mve_td	-0.03167 <.0001	0.67747 <.0001	0.04693 <.0001	0.00114 0.8599	0.00585 0.3665	-0.03452 <.0001	-0.00577 0.3735	0.02566 <.0001	0.00769 0.2358	0.04256 <.0001	0.00032 0.9600	0.00127 0.8452	1.00000	-0.03202 <.0001
bstatus	0.81353 <.0001	-0.00788 0.2239	-0.23079 <.0001	-0.02187 0.0007	-0.11813 <.0001	0.28898 <.0001	0.11370 <.0001	-0.22359 <.0001	-0.03490 <.0001	-0.17176 <.0001	-0.01077 0.0965	-0.02615 <.0001	-0.03202 <.0001	1.00000

Logistic Regression

Logistic Regression is selected as a predictive model in this project, because the responding variable is binary variable. (bankruptcy/non-bankruptcy)

In the training models, all models appear to be significant at 5 percent level, for training 1, 2 and 3, the p values are all <.0001, and the -2 Log L values drop significantly between intercept only and intercept with covariates in 116.311, 87.548, and 121.249 respectively.

By looking at the output of Maximum Likelihood Estimates, we are able to find each of the significant variables while running logistic regression model in each training data. Marked 1, if the variables turn out to be significant in the model, otherwise marked 0.

❖ Scoring Table

	cf_td	ca_cl	re_ta	ni_ta	td_ta	s_ta	wc_ta	wc_s	c_cl	cl_e	in_s	mve_td
Train1	1	1	1	1	1	1	0	0	1	0	0	1
Train2	0	0	1	1	1	1	1	0	1	0	0	0
Train3	1	1	1	1	1	1	1	1	1	0	0	1
Sum	2	2	3	3	3	3	2	1	3	0	0	2

Output:

Train 1

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.1956	0.0895	4.7796	0.0288
cf_td	1	0.0190	0.00273	48.0701	<.0001
ca_cl	1	0.2028	0.0341	35.3614	<.0001
re_ta	1	-0.1489	0.0192	60.2529	<.0001
ni_ta	1	0.3465	0.0723	22.9970	<.0001
td_ta	1	-1.6896	0.1170	208.6455	<.0001
s_ta	1	-0.1132	0.0277	16.6657	<.0001
wc_ta	1	-0.1325	0.0972	1.8585	0.1728
wc_s	1	-0.00623	0.00436	2.0418	0.1530
c_cl	1	1.5639	0.1335	137.3187	<.0001
cl_e	1	4.289E-6	0.000303	0.0002	0.9887
in_s	1	0.0848	0.0482	3.0946	0.0786
mve_td	1	0.00227	0.000407	31.1200	<.0001

Train 2

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.6735	0.0939	51.4897	<.0001
cf_td	1	0.000900	0.000918	0.9619	0.3267
ca_cl	1	0.0551	0.0331	2.7707	0.0960
re_ta	1	-0.2617	0.0232	127.4478	<.0001
ni_ta	1	1.4962	0.1086	189.9776	<.0001
td_ta	1	-1.6594	0.1259	173.8027	<.0001
s_ta	1	-0.2023	0.0310	42.7001	<.0001
wc_ta	1	0.8426	0.1421	35.1834	<.0001
wc_s	1	-0.00929	0.00494	3.5330	0.0602
c_cl	1	1.3575	0.1307	107.8393	<.0001
cl_e	1	0.000016	0.000331	0.0023	0.9620
in_s	1	0.0361	0.0395	0.8372	0.3602
mve_td	1	0.000399	0.000241	2.7383	0.0980

Train 3

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	1.1274	0.1111	102.9844	<.0001
cf_td	1	0.0374	0.00568	43.3900	<.0001
ca_cl	1	0.1351	0.0460	8.6379	0.0033
re_ta	1	-0.2802	0.0302	86.1458	<.0001
ni_ta	1	1.2426	0.1237	100.9530	<.0001
td_ta	1	-2.1992	0.1510	212.2056	<.0001
s_ta	1	-0.1964	0.0339	33.6494	<.0001
wc_ta	1	0.6382	0.1677	14.4820	0.0001
wc_s	1	-0.0209	0.00377	30.6901	<.0001
c_cl	1	1.3306	0.1656	64.5211	<.0001
cl_e	1	0.000216	0.000384	0.3160	0.5740
in_s	1	-0.0960	0.0535	3.2234	0.0726
mve_td	1	0.00409	0.000744	30.2956	<.0001

Selected points which are 2 or above, indicated that the variables have more than 50% chance of being significant in the prediction model. Final variables chosen are listed below:

Variables	Description
cf_td	Cash Flow/Total Debt
ca_cl	Current Assets/Current Liabilities
re_ta	Retained Earnings/Total Assets
ni_ta	Net Income/Total Assets
s_ta	Sales/Total Assets
wc_ta	Working Capital/Total Assets
c_cl	Cash/Current Liabilities
mve_td	Market Value of Equity/Total Debt

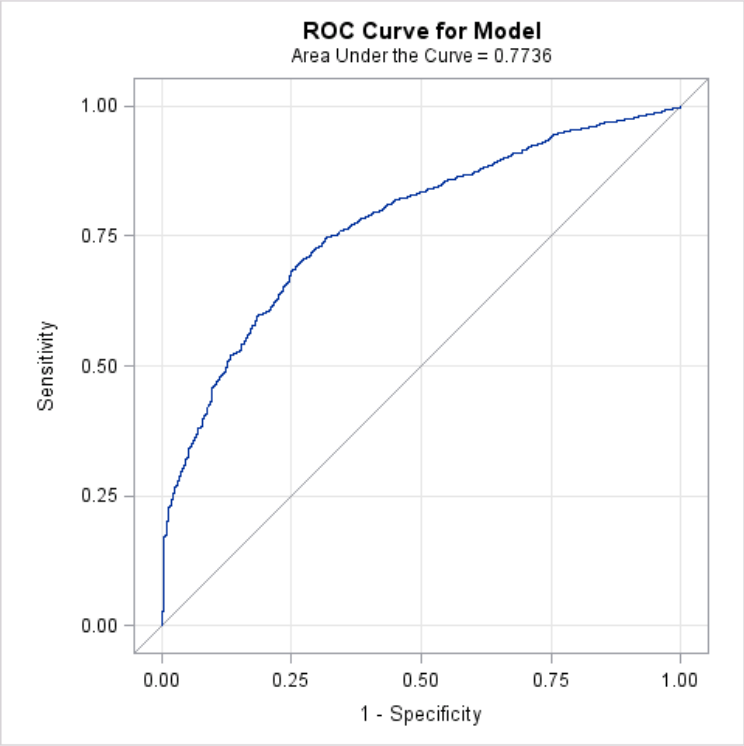
Created a model for testing data with bstatus (bank status: 1-bankruptcy; 2-non-bankruptcy) as a dependent variable and those selected variables, ct_td, ca_cl, re_ta, ni_ta, s_ta, wc_ta, c_cl, mve_td as independent variables. The likelihood ratio shows significant at 5% level with p-value <.0001. Also, by looking at the -2 Log L, we are able to tell that the numbers drop from 6514.136 to 5522.258 as comparison of ‘intercept and covariates’ and ‘intercept only’.

In the maximum likelihood estimates, expect for the value on intercept, all other the variables are significant at 5% level.

Testing: (final model)

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.0600	0.0959	0.3914	0.5316
cf_td	1	0.00226	0.000640	12.4386	0.0004
ca_cl	1	0.2263	0.0370	37.4431	<.0001
re_ta	1	-0.1309	0.0198	43.6425	<.0001
ni_ta	1	0.1182	0.0475	6.1812	0.0129
td_ta	1	-1.3513	0.1247	117.3832	<.0001
s_ta	1	-0.1563	0.0307	25.8973	<.0001
wc_ta	1	-0.1998	0.0820	5.9312	0.0149
c_cl	1	1.4681	0.1500	95.8249	<.0001
mve_td	1	0.000587	0.000250	5.5162	0.0188

In the below ROC curve, area under the curve shows that this model is covering 0.77 of the area, which indicates that this model is an acceptable discrimination.



❖ Model Estimation

First, we extract the estimate score from the final logistic regression model, then calculate possibilities in each observation (phat = exp(bstatus2)/(1+exp(bstatus2)). Then we compare our estimation results (into_l) with the actual values of bank status (bstatus) in all observations in the testing, training1, training2 and training3 datasets. If phat < 0.5 then categorizes into nonbankrupt, if phat >=0.5 then categorizes into bankruptcy. In comparison model shows that this scoring model only provides approximately 30% of accuracy to the prediction.

Testing: (29.85% accuracy)

Frequency Percent Row Pct Col Pct	Table of actual by into_l		
	into_l		
actual	0	1	Total
0	978 20.81 41.78 33.60	1363 29.01 58.22 76.23	2341 49.82
1	1933 41.14 81.98 66.40	425 9.04 18.02 23.77	2358 50.18
Total	2911 61.95	1788 38.05	4699 100.00

Train_1: (29.63% accuracy)

Frequency Percent Row Pct Col Pct	Table of actual by into_l		
	into_l		
actual	0	1	Total
0	1310 20.58 41.32 33.34	1860 29.22 58.68 76.35	3170 49.80
1	2619 41.15 81.97 66.66	576 9.05 18.03 23.65	3195 50.20
Total	3929 61.73	2436 38.27	6365 100.00

Train_2: (30.17% accuracy)

Frequency Percent Row Pct Col Pct	Table of actual by into_l		
	into_l		
actual	0	1	Total
0	1349 21.19 42.56 33.95	1821 28.60 57.44 76.10	3170 49.79
1	2625 41.23 82.11 66.05	572 8.98 17.89 23.90	3197 50.21
Total	3974 62.42	2393 37.58	6367 100.00

Train_3: (32.93% accuracy)

Frequency Percent Row Pct Col Pct	Table of actual by into_l		
	into_l		
actual	0	1	Total
0	1372 25.57 43.29 43.24	1797 33.49 56.71 81.98	3169 59.07
1	1801 33.57 82.01 56.76	395 7.36 17.99 18.02	2196 40.93
Total	3173 59.14	2192 40.86	5365 100.00

Appendix

❖ SAS Code for Decision Tree (highlight)

```
*------------------------------------------------------------*
 Node = 5
*------------------------------------------------------------*
if mve_td >= 3.50615
AND cf_td < 0.08977 or MISSING
then
 Tree Node Identifier   = 5
 Number of Observations = 754
 Predicted: bstatus=1 = 0.34
 Predicted: bstatus=0 = 0.66

*------------------------------------------------------------*
 Node = 8
*------------------------------------------------------------*
if mve_td < 0.48902 or MISSING
AND cf_td < 0.08977 or MISSING
then
 Tree Node Identifier   = 8
 Number of Observations = 1918
 Predicted: bstatus=1 = 0.88
 Predicted: bstatus=0 = 0.12

*------------------------------------------------------------*
 Node = 14
*------------------------------------------------------------*
if wc_ta < -0.0963
AND mve_td >= 1.1576 or MISSING
AND cf_td >= 0.08977
then
 Tree Node Identifier   = 14
 Number of Observations = 88
 Predicted: bstatus=1 = 0.51
 Predicted: bstatus=0 = 0.49

*------------------------------------------------------------*
 Node = 18
*------------------------------------------------------------*
if wc_s < 0.20875 or MISSING
AND mve_td < 3.50615 AND mve_td >= 0.48902
AND cf_td < 0.08977 or MISSING
then
 Tree Node Identifier   = 18
 Number of Observations = 1000
 Predicted: bstatus=1 = 0.72
 Predicted: bstatus=0 = 0.28

*------------------------------------------------------------*
 Node = 19
*------------------------------------------------------------*
if wc_s >= 0.20875
AND mve_td < 3.50615 AND mve_td >= 0.48902
AND cf_td < 0.08977 or MISSING
```

```
then
 Tree Node Identifier   = 19
 Number of Observations = 231
 Predicted: bstatus=1 = 0.39
 Predicted: bstatus=0 = 0.61

*------------------------------------------------------------*
 Node = 27
*------------------------------------------------------------*
if td_ta >= 0.32315 or MISSING
AND mve_td < 1.1576
AND cf_td >= 0.08977
AND c_cl >= 0.0548
then
 Tree Node Identifier   = 27
 Number of Observations = 171
 Predicted: bstatus=1 = 0.10
 Predicted: bstatus=0 = 0.90

*------------------------------------------------------------*
 Node = 29
*------------------------------------------------------------*
if wc_ta >= -0.0963 or MISSING
AND ni_ta >= 0.00597 or MISSING
AND mve_td >= 1.1576 or MISSING
AND cf_td >= 0.08977
then
 Tree Node Identifier   = 29
 Number of Observations = 1511
 Predicted: bstatus=1 = 0.05
 Predicted: bstatus=0 = 0.95

*------------------------------------------------------------*
 Node = 42
*------------------------------------------------------------*
if mve_td < 1.1576
AND cl_e < 0.35365
AND cf_td >= 0.08977
AND c_cl < 0.0548 or MISSING
then
 Tree Node Identifier   = 42
 Number of Observations = 57
 Predicted: bstatus=1 = 0.61
 Predicted: bstatus=0 = 0.39

*------------------------------------------------------------*
 Node = 43
*------------------------------------------------------------*
if mve_td < 1.1576
AND cl_e < 1.09531 AND cl_e >= 0.35365 or MISSING
AND cf_td >= 0.08977
AND c_cl < 0.0548 or MISSING
then
 Tree Node Identifier   = 43
 Number of Observations = 81
 Predicted: bstatus=1 = 0.11
 Predicted: bstatus=0 = 0.89
```

```
*-------------------------------------------------------------*
 Node = 44
*-------------------------------------------------------------*
if mve_td < 1.1576
AND cl_e >= 1.09531 or MISSING
AND cf_td >= 0.08977
AND c_cl < 0.028
then
 Tree Node Identifier   = 44
 Number of Observations = 79
 Predicted: bstatus=1 = 0.46
 Predicted: bstatus=0 = 0.54

*-------------------------------------------------------------*
 Node = 45
*-------------------------------------------------------------*
if mve_td < 1.1576
AND cl_e >= 1.09531 or MISSING
AND cf_td >= 0.08977
AND c_cl < 0.0548 AND c_cl >= 0.028 or MISSING
then
 Tree Node Identifier   = 45
 Number of Observations = 125
 Predicted: bstatus=1 = 0.84
 Predicted: bstatus=0 = 0.16

*-------------------------------------------------------------*
 Node = 46
*-------------------------------------------------------------*
if td_ta < 0.32315
AND mve_td < 0.71533 or MISSING
AND cf_td >= 0.08977
AND c_cl >= 0.0548
then
 Tree Node Identifier   = 46
 Number of Observations = 60
 Predicted: bstatus=1 = 0.72
 Predicted: bstatus=0 = 0.28

*-------------------------------------------------------------*
 Node = 47
*-------------------------------------------------------------*
if td_ta < 0.32315
AND mve_td < 1.1576 AND mve_td >= 0.71533
AND cf_td >= 0.08977
AND c_cl >= 0.0548
then
 Tree Node Identifier   = 47
 Number of Observations = 54
 Predicted: bstatus=1 = 0.31
 Predicted: bstatus=0 = 0.69

*-------------------------------------------------------------*
 Node = 50
*-------------------------------------------------------------*
if wc_ta >= -0.0963 or MISSING
```

```
AND re_ta < -0.1009
AND ni_ta < 0.00597
AND mve_td >= 1.1576 or MISSING
AND cf_td >= 0.08977
then
 Tree Node Identifier   = 50
 Number of Observations = 101
 Predicted: bstatus=1 = 0.59
 Predicted: bstatus=0 = 0.41

*------------------------------------------------------------*
 Node = 51
*------------------------------------------------------------*
if wc_ta >= -0.0963 or MISSING
AND re_ta >= -0.1009 or MISSING
AND ni_ta < 0.00597
AND mve_td >= 1.1576 or MISSING
AND cf_td >= 0.08977
then
 Tree Node Identifier   = 51
 Number of Observations = 137
 Predicted: bstatus=1 = 0.12
 Predicted: bstatus=0 = 0.88
```

❖ SAS code for correlation (highlight)

```
proc corr data=s.m01 plots=matrix;
run;
```

❖ SAS code for logistic regression (highlight)

```
/*logistic regression on training 1*/
proc logistic data=b.train_1;
model bstatus = cf_td ca_cl re_ta ni_ta td_ta s_ta wc_ta wc_s c_cl cl_e in_s
mve_td;
run;
/*logistic regression on training 2*/
proc logistic data=b.train_2;
model bstatus = cf_td ca_cl re_ta ni_ta td_ta s_ta wc_ta wc_s c_cl cl_e in_s
mve_td;
run;
/*logistic regression on training 3*/
proc logistic data=b.train_3;
model bstatus = cf_td ca_cl re_ta ni_ta td_ta s_ta wc_ta wc_s c_cl cl_e in_s
mve_td;
run;
/*after select each significant variables from training dataset, we run
logistic regression with selected variables on testing*/
proc logistic data=b.TESTING outest=est noprint;
model bstatus = cf_td ca_cl re_ta ni_ta td_ta s_ta wc_ta c_cl mve_td;
run;
/*scoring out the result from logistic regression model*/
proc score score = est data=b.TESTING type=parms out=cl;
var cf_td ca_cl re_ta ni_ta td_ta s_ta wc_ta c_cl mve_td;
run;
```

```
/*set up the cut off number*/
data b.cl; set b.cl;
phat = exp(bstatus)/(1+exp(bstatus));
if phat < 0.6 then bank = "Bankruptcy";
if phat >=0.6 then bank = "Non-Bankruptcy";
/*classification table*/
proc freq data=b.log_train3;
table actual*into_l;
run;

/*hold out sample from testing data*/
data b.test_tmp b.test_tmp_hold; set b.testing;
p = rand('UNIFORM');
if p <= 0.2 then output b.test_tmp_hold;
else output b.test_tmp;
run;

/*calculate prediction rate from hold out sample*/
data b.test_tmp_hold; set b.test_tmp_hold;
actual = bstatus;
intercept = 1;
drop bstatus;
run;

proc score score = b.est data=b.test_tmp_hold type=parms out=b.log;
var intercept cf_td ca_cl re_ta ni_ta td_ta s_ta wc_ta c_cl mve_td ;
run;

data b.log; set b.log;
phat = exp(bstatus)/(1+exp(bstatus));
if phat < 0.5 then into_l = 2;
if phat >=0.5 then into_l = 1;
run;

proc freq data=b.log;
title 'Actual vs. Predicted on hold-out sample using LR';
tables actual*into_l;
run;
```