The work contained and presented here is our team's work and our team's work alone

KICKSTARTER PROJECT REPORT

BY:

TEAM 2
Balakrishnan Anisha,
Chen Xi,
Doddagaddavallinarasimhamurthy Amoghbharadwaj,
Lakamsani Vinay Vihari,
Lalanne Nerlande

Table of Contents

Problem Statement

The objective of our project is to predict if a campaign will be successful or unsuccessful before its completion using the Kickstarter funding platform.

ks-projects-201612-data...
Source

Columns (13/0)
ID
name
category
main_category
currency
deadline
goal
launched
pledged
state
backers
country
usd pledged

Rows
All rows 323,750
Selected 0

	ID	name	category	main_category	currency	deadline	goal	launched	pledged	state	backers	country	usd pledged
1	1000002330	The Songs of Adelaide & ...	Poetry	Publishing	GBP	10/09/2015 11:36 AM	1000	08/11/2015 12:12 PM	0	failed	0	GB	0
2	1000004038	Where is Hank?	Narrative Film	Film & Video	USD	02/26/2013 12:20 AM	45000	01/12/2013 12:20 AM	220	failed	3	US	220
3	1000007540	ToshiCapital Rekordz ...	Music	Music	USD	04/16/2012 4:24 AM	5000	03/17/2012 3:24 AM	1	failed	1	US	1
4	1000011046	Community Film Project: ...	Film & Video	Film & Video	USD	08/29/2015 1:00 AM	19500	07/04/2015 8:35 AM	1283	canceled	14	US	1283
5	1000014025	Monarch Espresso Bar	Restaurants	Food	USD	04/01/2016 1:38 PM	50000	02/26/2016 1:38 PM	52375	successful	224	US	52375
6	1000023410	Support Solar Roasted ...	Food	Food	USD	12/21/2014 6:30 PM	1000	12/01/2014 6:30 PM	1205	successful	16	US	1205
7	1000030581	Chaser Strips. Our Strips ...	Drinks	Food	USD	03/17/2016 7:05 PM	25000	02/01/2016 8:05 PM	453	failed	40	US	453
8	1000034518	SPIN - Premium ...	Product Design	Design	USD	05/29/2014 6:14 PM	125000	04/24/2014 6:14 PM	8233	canceled	58	US	8233
9	100004195	STUDIO IN THE SKY - A ...	Documentary	Film & Video	USD	08/10/2014 9:55 PM	65000	07/11/2014 9:55 PM	6240.57	canceled	43	US	6240.57
10	100004721	Of Jesus and Madmen	Nonfiction	Publishing	CAD	10/09/2013 6:19 PM	2500	09/09/2013 6:19 PM	0	failed	0	CA	0
11	100005484	Lisa Lim New CD!	Indie Rock	Music	USD	04/08/2013 6:42 AM	12500	03/09/2013 6:42 AM	12700	successful	100	US	12700
12	1000055792	The Cottage Market	Crafts	Crafts	USD	10/02/2014 5:11 PM	5000	09/02/2014 5:11 PM	0	failed	0	US	0
13	1000056157	G-Spot Place for Gamers ...	Games	Games	USD	03/25/2016 10:01 PM	200000	02/09/2016 11:01 PM	0	failed	0	US	0
14	1000064368	Survival Rings	Design	Design	USD	02/28/2015 2:10 AM	2500	01/29/2015 2:10 AM	664	failed	11	US	664
15	1000064918	The Beard	Comic Books	Comics	USD	11/08/2014 10:27 PM	1500	10/09/2014 10:27 PM	395	failed	16	US	395
16	1000068480	Notes From London: ...	Art Books	Publishing	USD	05/10/2015 9:20 PM	3000	04/10/2015 9:20 PM	789	failed	20	US	789
17	1000070642	Mike Corey's Darkness & ...	Music	Music	USD	08/17/2012 2:11 PM	250	08/02/2012 2:11 PM	250	successful	7	US	250
18	1000071625	Boco Tea	Food	Food	USD	06/02/2012 5:24 PM	5000	05/03/2012 5:24 PM	1781	failed	40	US	1781
19	1000072011	CMUK. Shoes: Take on ...	Fashion	Fashion	USD	12/30/2013 7:06 AM	20000	11/25/2013 7:06 AM	34268	successful	624	US	34268
20	1000082254	Alice in Wonderland in G ...	Theater	Theater	USD	06/15/2014 10:09 AM	3500	05/16/2014 10:10 AM	650	failed	12	US	650
21	1000087442	Mountain brew: A quest ...	Drinks	Food	NOK	02/25/2015 7:17 PM	500	01/26/2015 7:17 PM	48	failed	3	NO	6.18000816
22	1000091520	The Book Zoo - A Mini-...	Comics	Comics	USD	11/12/2014 5:15 PM	175	10/23/2014 5:15 PM	701.66	successful	66	US	701.66
23	1000102741	Matt Cavenaugh & Jenny ...	Music	Music	USD	01/06/2011 11:16 PM	10000	12/07/2010 11:16 PM	15827	successful	147	US	15827
24	1000103948	Superhero Teddy Bear	DIY	Crafts	GBP	01/05/2016 8:09 PM	12000	12/06/2015 8:09 PM	0	failed	0	GB	0
25	1000104688	Permaculture Skills	Webseries	Film & Video	CAD	12/14/2014 6:02 PM	17757	11/14/2014 6:02 PM	48905	successful	571	CA	43203.25114
26	1000104953	Rebel Army Origins: The ...	Comics	Comics	GBP	01/28/2016 4:59 PM	100	12/29/2015 4:59 PM	112.38	successful	27	GB	167.7003058
27	1000115172	Daily Brew Coffee	Food Trucks	Food	GBP	03/31/2015 6:06 PM	3500	03/01/2015 6:06 PM	21	failed	1	GB	32.41526106
28	1000117861	Ledr workbook: one ...	Product Design	Design	USD	10/08/2016 2:00 AM	1000	09/07/2016 1:14 PM	47266	successful	549	US	11253
29	1000120151	Feather Cast Furled Fly ...	Product Design	Design	AUD	08/22/2015 3:09 AM	2000	07/23/2015 3:09 AM	2000	successful	18	AU	1473.6189
30	1000120287	BB130A	Public Art	Art	USD	03/24/2013 12:07 AM	25000	02/12/2013 1:07 AM	1395	failed	30	US	1395
31	100012079	Chris Eger Band - New ...	Music	Music	USD	08/13/2014 8:59 AM	12000	07/14/2014 10:35 PM	13260	successful	92	US	13260
32	1000129669	Squatch Watchers	Webseries	Film & Video	USD	08/23/2013 9:40 PM	10000	07/24/2013 9:40 PM	1373.37	failed	54	US	1373.37
33	1000131947	Arrows & Sound Debut ...	Indie Rock	Music	USD	05/19/2012 1:04 AM	4000	04/19/2012 1:04 AM	8641.34	successful	157	US	8641.34

Executive Summary

The goal of this project is to predict the success or failure of a Kickstarter campaign at the time it is launched. We feel that this analysis is important since only 35% of campaigns reach their funding goals, Kickstarter has to know the factors that play a role in a project's outcome before launching. Also, we can suggest to backers as to which project is prone to be more successful. The projects that we analyzed fell into 15 categories and 13 columns. The categories are Art, Comics, Dance, Design, Fashion, Film & Video, Food, Games, Journalism, Music, Photography, Publishing, Technology and Theater.

Our team started with a dataset of 323,750 observations with 13 variables. Each project was given a status (state) of either canceled, failed, live, successful, suspended and undefined. In order to clean and analyze the data, we deleted all rows that had inconsistencies and errors, therefore reducing the dataset to 320,661. Our analysis started in 2009 and ended in 2016 with total pledged amount of over $2 billion. The funding goal for projects ranged from $0.01 to $126,720,000. About 65 % projects failed to meet the minimum goal.

We found that the column "USD_PLEDGED" has missing data, so we created a new column to reflect the amount of pledged USD. In addition, we analyzed the outliers for each column and decided to keep the outliers, because we need the information to build the model. Furthermore, there are observations where Backers equals 0 but Pledged amount is greater than zero. This is inconsistent as there should be at least one backer to pledge an amount for funding the project. As a result, we deleted rows where data was inconsistent.

Because we were interested in discovering trends or strong indicators that would distinguish between successful and failed projects, we focused on analyzing important statistics of various projects. We found that successful projects had funding goals averaging six times less than failed projects as well as Pledged amounts averaging eleven times more than failed projects.

Another identifiable trend is that the Pledged USD amount increases with the Duration of the campaign. Image 1 shows that if the Duration is between 30 and 50 days, funding amount increases. Another trend revealed is that more Backers are attracted to projects originating in the United States as compared to originating in other countries (Image 2).
Also, Image 3 reveals that more Backers are attracted to the project when its funding goal amount is small. Therefore, we can hypothesize that smaller projects are more likely to attract a larger number of backers. In addition, we found that more Backers are attracted when the campaign is paid for in USD (Image 4).

Another analysis was carried out as to whether certain categories performed better than others. (Image 5 & 6) As shown in image 5, the category of Games received the largest number of

Backers, whereas Crafts received the least number of Backers. As shown in image 6, the least amount Pledged USD was for Crafts and Design and the maximum Pledged USD was for Design. Therefore, we conclude that Games is the category that's in demand.

Image 7 shows additional data for average backers of the projects across all categories. Because of the 2009 recession, there was a decline in the number of Backers due to the high risk and fear dominating the investment world. As a matter of fact, image 7 reveals that from 2012 – 2016 the number of investors of Backers increased.

Image 8 shows the relationship between the number of backers and the success or failure of a campaign and reveals that when the number of backers is reduced, the campaign is more likely to fail, irrespective of the state of the economy.

Once we identified various patterns, we were able to build many models such as classification models, regression models and ensemble models. We tried different split percentages for training, validation and test data set so that we can better determine which model is best. We feel that the decision tree model is the best model since it is easier to explain and it provides the highest accuracy.

Finally, we calculated the mean of ratio of Total USD Pledged to the number of Backers, which is the estimated loss value for False Negative Cases. When mean equals $68.32, the total amount being pledge directly will be considered as a loss. We also calculated the Cost of Error for False Negative which is $68.32. Another calculation was in the case of not investing in a project that will be successful. In this case the investor will lose profit expect if he can get a general bank interest rate of 10%. On the other hand, if someone invested in the project, the profit could be at least 40% of the investment. However, the cost of error for false positive equals to $20.49.

In conclusion, we came up with the following recommendations:

- Since the Country column has 3,797 observations (1%) with the value 'NO', it should be made a required field because the currency of Goal and Pledged amounts depend on that variable. In addition, data for the Country variable should be carefully collected. As a result, data for the Country variable should be carefully collected.

- For data to be valuable and accurate, Pledged amounts should always be associated with a Backer. For example, we identified 3,082 records that had zero Backers but some Pledged amounts. When this occurs, data is not useful. Therefore, the business should be sure that Backers are consistent with Pledged variables.

- Projects with Goal amounts over 2 Million should not be considered for launching a campaign since they are more likely to get Failed or Canceled as there are less chances of being backed.

- The columns Estimated Start Date and End Date should always be included in analyzing and modeling data so as to provide more accuracy in predicting duration of projects.

- To be more valuable to marketers, data should include specific location of project being launched.

- It is recommended that the company should take efforts to verify the Goal amount so that it is in line with the project and we do not use false values to make predictions.

DATA DICTIONARY

The Kickstarter dataset (see below) contains 323,751 rows and 13 columns.

EXISTING VARIABLES

Variable Name	Old Variable Type	Final Variable type	Description of the variable
ID	Continuous	Nominal	Internal Kickstarter ID
NAME	Nominal	Nominal	Name of the project
CATEGORY	Nominal	Nominal	Sub category under which the projects fall
MAIN_CATEGORY	Nominal	Nominal	Category under of project
CURRENCY	Nominal	Nominal	Currency used to support
DEADLINE	Nominal	Continuous	Deadline for crowdfunding
GOAL	Nominal	Continuous	Amount of money required to complete the project
LAUNCHED	Nominal	Continuous	Launched date for crowdfunding
PLEDGED	Nominal	Continuous	Amount pledged by crowd
STATE	Nominal	Nominal	The current condition the project is in
BACKERS	Nominal	Continuous	The number of people who have backed the project

COUNTRY	Nominal	Nominal	*The country where the project is being created.
USD PLEDGED	Nominal	Continuous	The value of pledged amount in USD currency with currency value as per the date when it was pledged

*As per the old Data Dictionary, the Country column shows the country which originated the pledged amount. Since people can pledge money from more than one country. Hence, there cannot be one country name if the column contains the country from which money was pledged, the column should contain the country where the project is being conducted.

NEW VARIABLES

Below is the list of new variables created for the purpose of Exploratory Data Analysis and Modeling.

Variable Name	**Variable Type**	**Description of the variable**
CURRENCY_RATE	Continuous	The foreign exchange rate of every country from the country column since 10/31/2018
FINAL STATE	Nominal	This column has value 1 for corresponding value on status column as 'Successful' and value 0 for corresponding status as 'Cancelled', 'Suspended', 'Failed'
DEADLINE_YEAR	Continuous	The year of the deadline of crowdfunding from the Deadline column
LAUNCHED_YEAR	Continuous	The year of launch of crowdfunding from the Launched column
GOAL_USD	Continuous	The Goal amount in USD currency
PLEDGED_USD	Continuous	The Pledged amount in USD currency

FUNDING_DURATION	Continuous	The difference of Launched and Deadline date of crowdfunding
RATIO_OF_PLEDGED_TO_GOAL	Continuous	The ratio of the pledged amount to the goal amount
RATIO_OF_PLEDGED_TO_BACKERS	Continuous	The ratio of pledged amount to the number of backers
RANGE SCALE[BACKERS]	Continuous	The number of Backers in the scale of 0 to 1
RANGE SCALE[GOAL_USD]	Continuous	The goal amount in USD in the scale of 0 to 1
RANGE SCALE[PLEDGED_USD]	Continuous	The pledged amount in USD in the scale of 0 to 1
RANGE SCALE[FUNDING DURATION]	Continuous	The duration between launched date and the deadline of crowdfunding

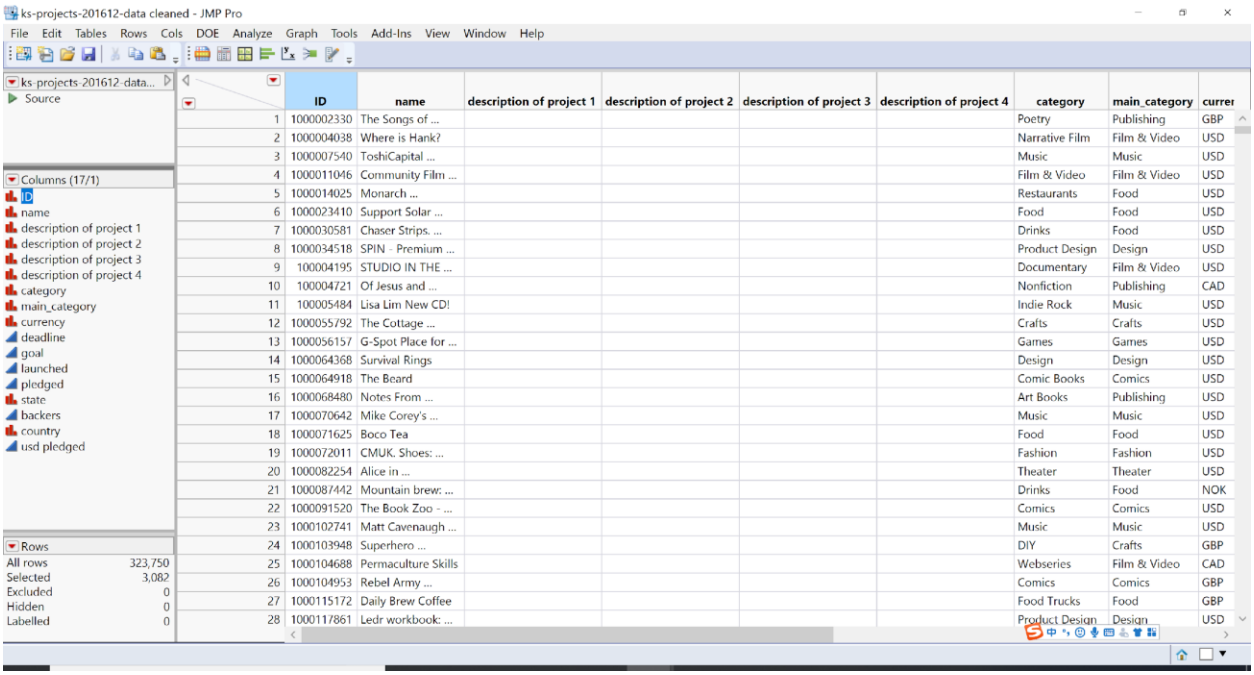
DATA CLEANING

In order to explore the dataset, it is necessary for the data to be accurate, complete and relevant to the business so that we can predict the outcome of the project.

1. Resolve Inconsistencies

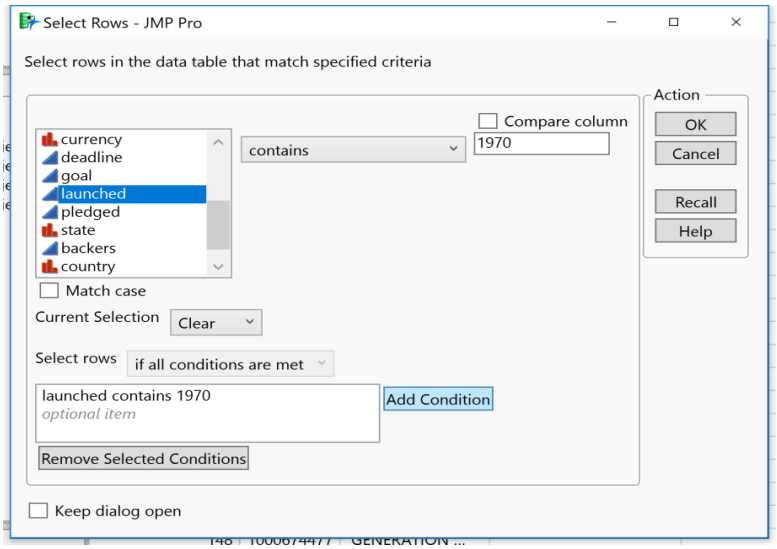
There are some observations where Backers equal zero but Pledged amount are greater than zero. This is inconsistent as there should be at least one backer to pledge an amount for funding the project.

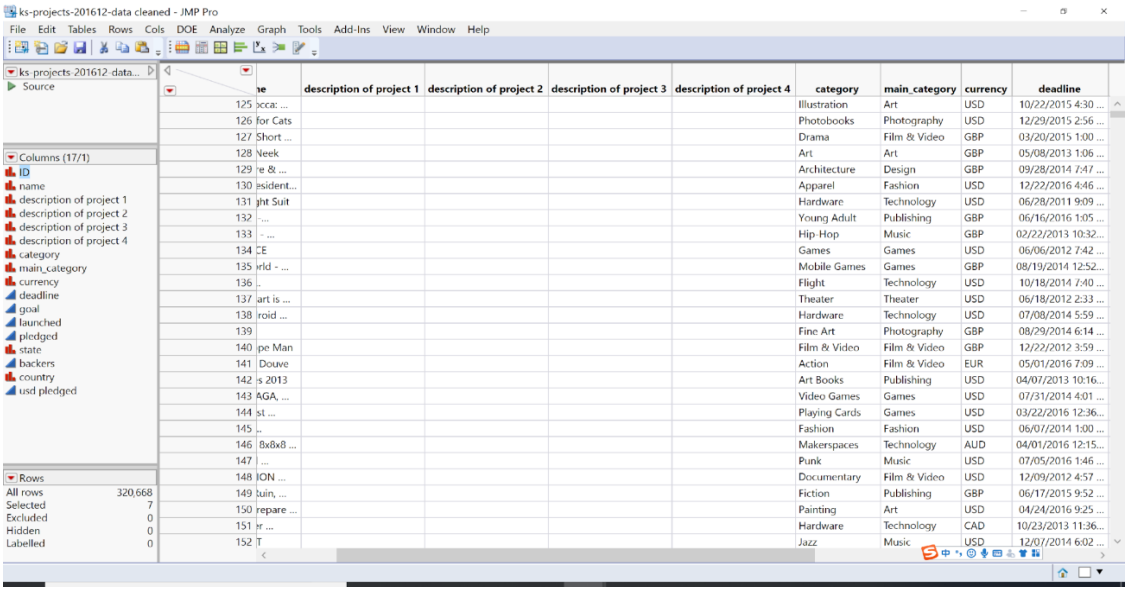
Therefore, 3082 records with Backers = 0 and Pledged amount > 0 are removed.



There are some observations with a Launched year as 1970 and funding duration as long as 30 years. This is impractical as the Kickstarter website was created in the year 2009 and any projects with launch date before 2009 will be bogus.

Therefore, these 7 records were also deleted.





2. Outlier Analysis

The GOAL column with amount greater than 2 million USD had project status as ‘Failed’ or ‘Cancelled’. However, they cannot be omitted because they are genuine values and thus values above 2 million USD should be considered.

Some categories have records with very high Pledged values over 20 million USD. Since they are true values and are verified by the Kickstarter website, they are not omitted.

[Reference – Univariate Outlier Analysis Screenshots Appendix 1.1.]

Univariate outlier analysis:

Variable Name	Variable Type	Outlier Analysis
CURRENCY_RATE	Continuous	We are not using this variable for modeling
FINAL STATE	Nominal	As it is a categorical variable we cannot do outlier analysis
DEADLINE_YEAR	Continuous	There are no outliers
LAUNCHED_YEAR	Continuous	There are no outliers
GOAL_USD	Continuous	The Goal amount greater than 2 million USD have project status as Failed. But they cannot be omitted as they are true values.
PLEDGED_USD	Continuous	The Pledged amount in USD which are greater than 20million USD are genuine values thus these observations are considered for the predictions
FUNDING_DURATION	Continuous	After deleting observations with launched date as 1970, we have no outliers
RATIO_OF_PLEDGED_TO_GOAL	Continuous	This variable is not used for modeling
RATIO_OF_PLEDGED_TO_BACKERS	Continuous	This variable is not used for modeling
RANGE SCALE[BACKERS]	Continuous	This variable is not used for modeling
RANGE SCALE[GOAL_USD]	Continuous	Same as Goal_USD
RANGE SCALE[PLEDGED_USD]	Continuous	Same as Pledged_USD
RANGE SCALE [FUNDING DURATION]	Continuous	Same as Funding duration

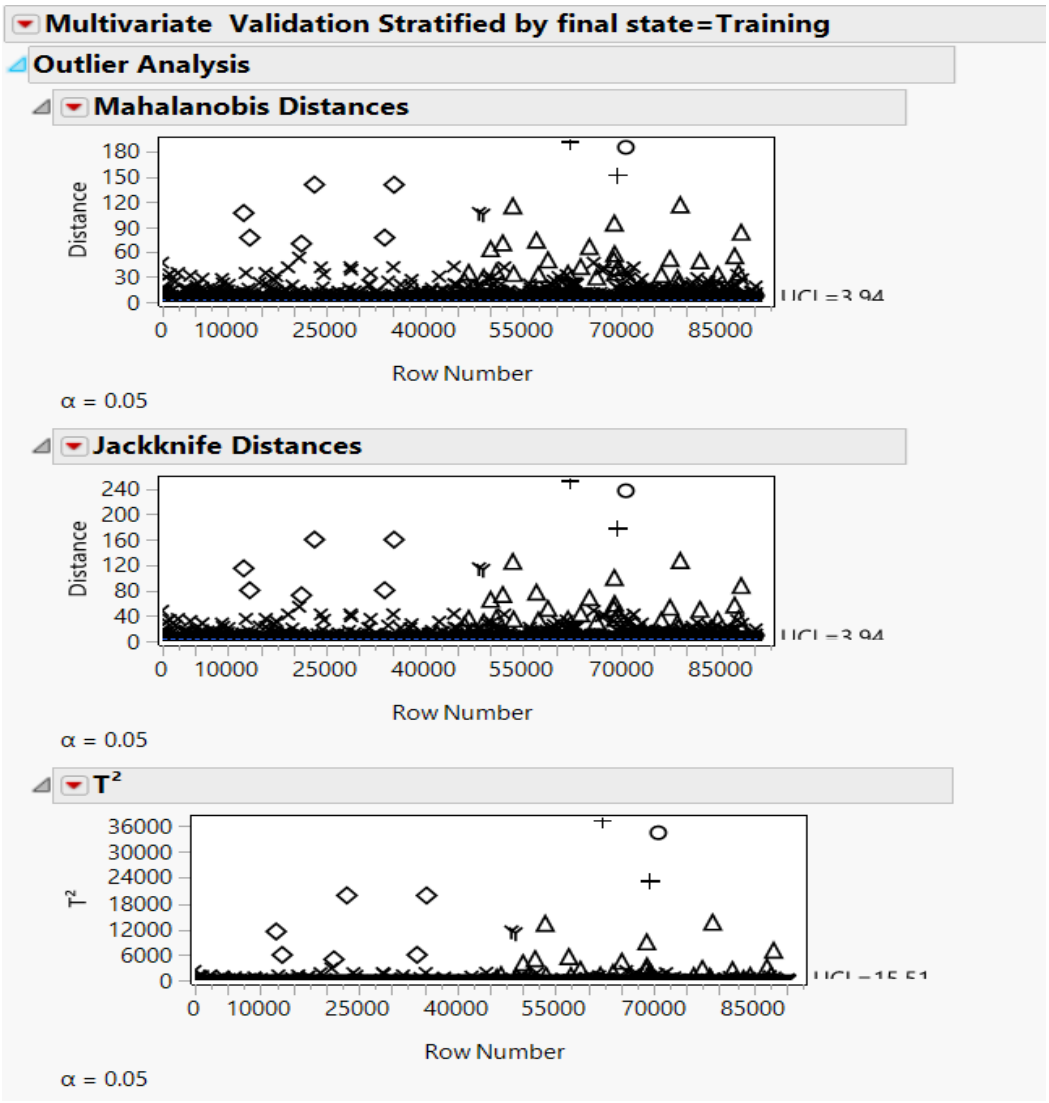
Multivariate outlier Analysis:

Modelling prediction saved - Multivariate - JMP Pro

Multivariate Validation Stratified by final state=Training

Correlations

	deadline year	launched year	Range Scale[backers]	Range Scale[goal usd]	Range Scale[pledged usd]	Range Scale[funding duration]	ratio pledge of goal	ratio pledged by backers
deadline year	1.0000	0.9897	0.0153	0.0206	0.0243	-0.1643	0.0065	-0.0024
launched year	0.9897	1.0000	0.0159	0.0191	0.0247	-0.1856	0.0067	-0.0029
Range Scale[backers]	0.0153	0.0159	1.0000	0.0119	0.7975	-0.0005	0.0210	0.0113
Range Scale[goal usd]	0.0206	0.0191	0.0119	1.0000	0.0130	0.0203	-0.0007	0.0016
Range Scale[pledged usd]	0.0243	0.0247	0.7975	0.0130	1.0000	0.0099	0.0155	0.1009
Range Scale[funding duration]	-0.1643	-0.1856	-0.0005	0.0203	0.0099	1.0000	-0.0069	0.0299
ratio pledge of goal	0.0065	0.0067	0.0210	-0.0007	0.0155	-0.0069	1.0000	0.0021
ratio pledged by backers	-0.0024	-0.0029	0.0113	0.0016	0.1009	0.0299	0.0021	1.0000



From the above image, we can see few outliers but we are not removing them because all of them are true values and were verified by the Kickstarter website.

3. Missing data

There are 465 observations with COUNTRY column labeled as 'NO', the corresponding USD_PLEDGED values are missing. Since the currency exchange rate fluctuates, we use the exchange rate at 2018.10.31 to calculate the amount of pledged USD. We created a new column called PLEDGED_USD. This newly created column is used for further analysis and modeling.

The final data set after data preprocessing has a total of 320,661 observations in total.

INTERESTING PATTERNS

1. **Pledged USD vs Funding duration**

```
sns.lmplot('funding duration','pledged usd', df)
```

```
<seaborn.axisgrid.FacetGrid at 0x19494bcebe0>
```

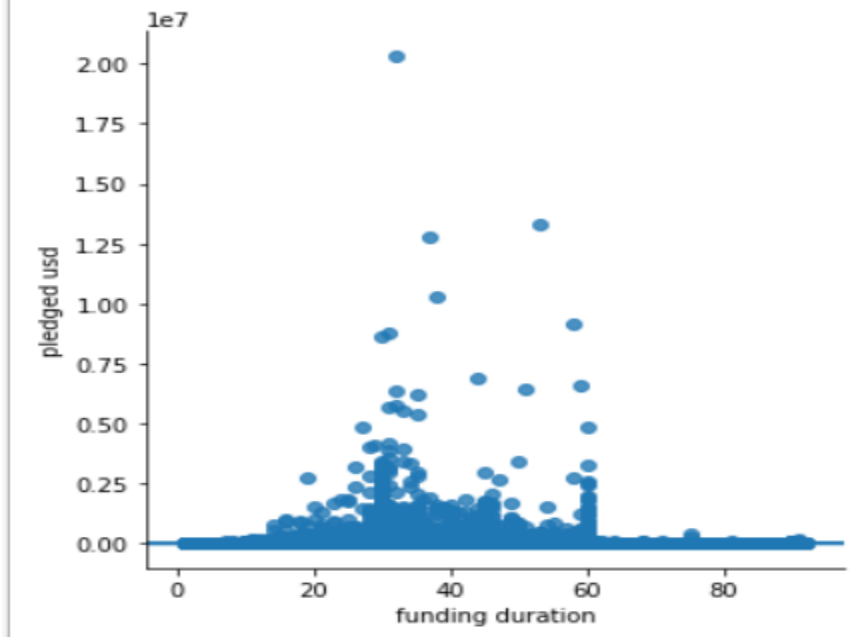


IMAGE1

We observe that If the duration is between 30-50, we get more funding. This is the ideal funding duration for all the projects. The funding duration should not be too small or too long for any Project.

2. Country Major vs Backers

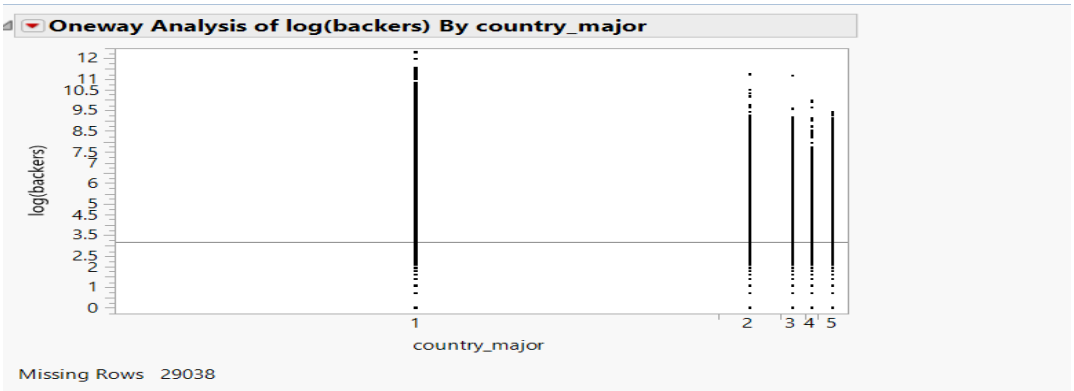


IMAGE2

We can hypothesize that more backers are more likely to invest in US project than any other country. It may be due to the trust that people tend to have in US projects. Also, the technological advantages that are more available in the US than anywhere in the world. Lastly, the United States tends to have a better reputation in regards to successful projects.

3. Goal vs Backers

```
plt.figure(figsize=(12,3))
sns.lmplot('goal','backers', data= df,size=2,aspect=4)
```

```
<seaborn.axisgrid.FacetGrid at 0x194958a0e48>

<Figure size 864x216 with 0 Axes>
```

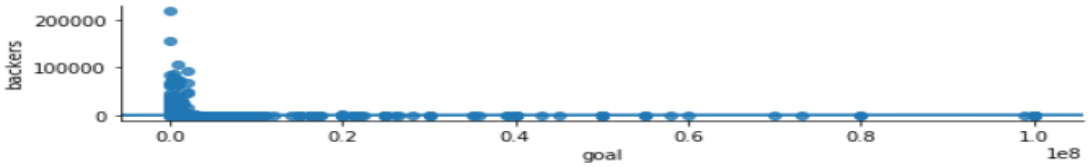


IMAGE3

We can expect more backers if the project's goal amount is small. Maybe small projects are likely to get more backers. Maybe people think that if the funding goal amount is large, it might take a long time to reach its goal and loses its interests.

4. Currency vs Backers

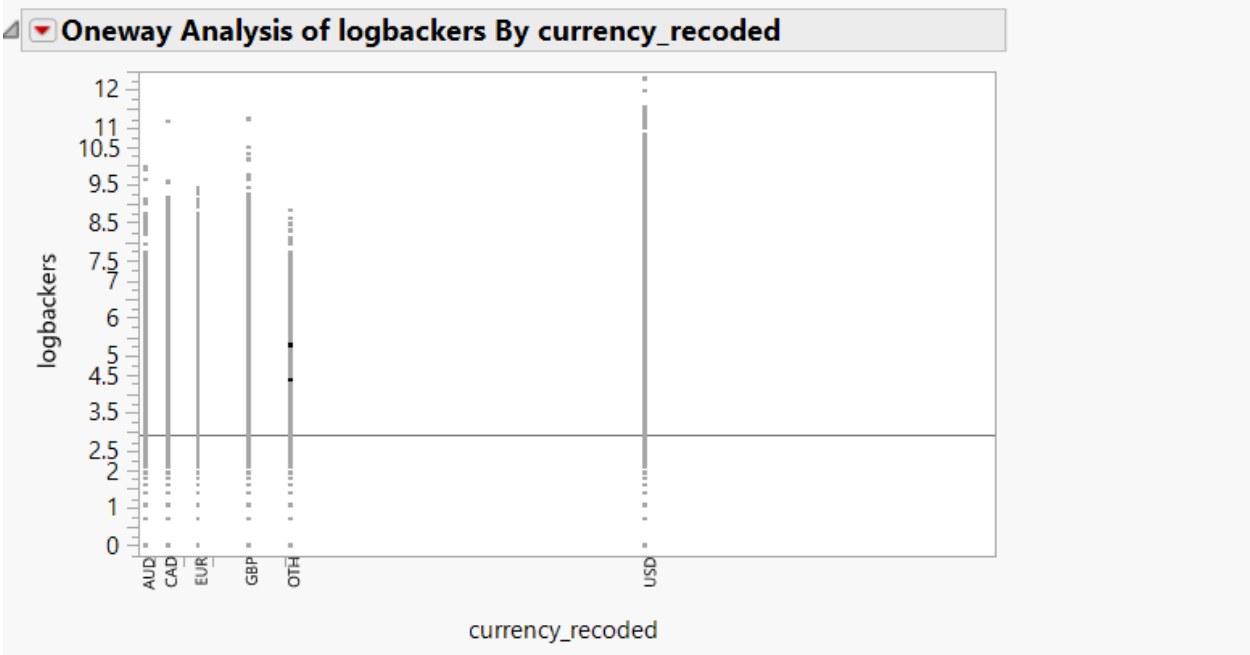


IMAGE4

We can expect more backers for USD currency. It may be due to the fact that the currency USD is universally accepted and it is well-known all over the world. Also, it may be due to the fact that the rate is also not as high as other renowned currencies like Euro, Pound.

5.Average backers for each category

```
backers = df.groupby('main_category').mean()['backers']
backers
```

```
main_category
Art                39.424094
Comics            134.361882
Crafts             26.921023
Dance              43.272216
Design            237.192436
Fashion            62.283837
Film & Video       67.217146
Food               54.333427
Games             323.892388
Journalism         38.231994
Music              53.710210
Photography        37.304191
Publishing         52.686762
Technology        165.532824
Theater            47.408703
Name: backers, dtype: float64
```

IMAGE5

We see that Games category gets usually large number of Backers and Crafts category with the least number of backers. This maybe a general indicator for the current trend

6. Main Category vs Pledged USD

```
df.groupby('main_category').mean()['pledged usd']
```

```
main_category
Art               3009.842974
Comics            6631.307194
Crafts            1538.531379
Dance             3391.538872
Design           22978.299055
Fashion           5531.053544
Film & Video      6128.339944
Food              4955.110715
Games            20129.557520
Journalism        2533.431828
Music             3790.114682
Photography       3156.833010
Publishing        3090.255067
Technology       20143.498302
Theater           3942.357402
Name: pledged usd, dtype: float64
```

IMAGE6

We get less pledged USD for Crafts Category and the max pledged USD is for Design category. Interestingly even though the average backers for Design and Technology was less than Games, we see that the amount raised by these categories are higher than the rest. So, we can conclude there is a big demand for Games, Technology and Design domains.

7. Launched Year vs Backers

```
df.groupby('launched year').mean()['backers']
```

```
launched year
2009     32.925508
2010     38.683685
2011     53.225331
2012    105.526468
2013    140.346329
2014     92.373636
2015     99.555476
2016    123.476453
Name: backers, dtype: float64
```

IMAGE7

Due to the recession in 2009-2011, we saw less backers on an average due to the risk of losing money and fear in economy. Apart from that, people are willing to back projects with a reasonable pledged amount. Unless something bad happens, we can have a good number of backers for a project.

8.Average backers for a project to be successful and failure in each year

```
df[df['final state'] ==1].groupby('launched year').mean()['backers']
```

```
launched year
2009     66.357513
2010     77.586109
2011    101.606442
2012    215.789012
2013    285.733790
2014    252.528949
2015    317.700873
2016    288.854610
Name: backers, dtype: float64
```

```
df[df['final state'] ==0].groupby('launched year').mean()['backers']
```

```
launched year
2009     7.116000
2010     8.526920
2011    11.362221
2012    20.750677
2013    29.439372
2014    18.816643
2015    15.592420
2016    19.508526
Name: backers, dtype: float64
```

IMAGE8

We observe that if backers increase, the project is more likely to be successful. The increased number of backers might indicate that people's confidence in a project is high. Therefore, this might create a superficial demand for other backers and encourage them to invest on something new.

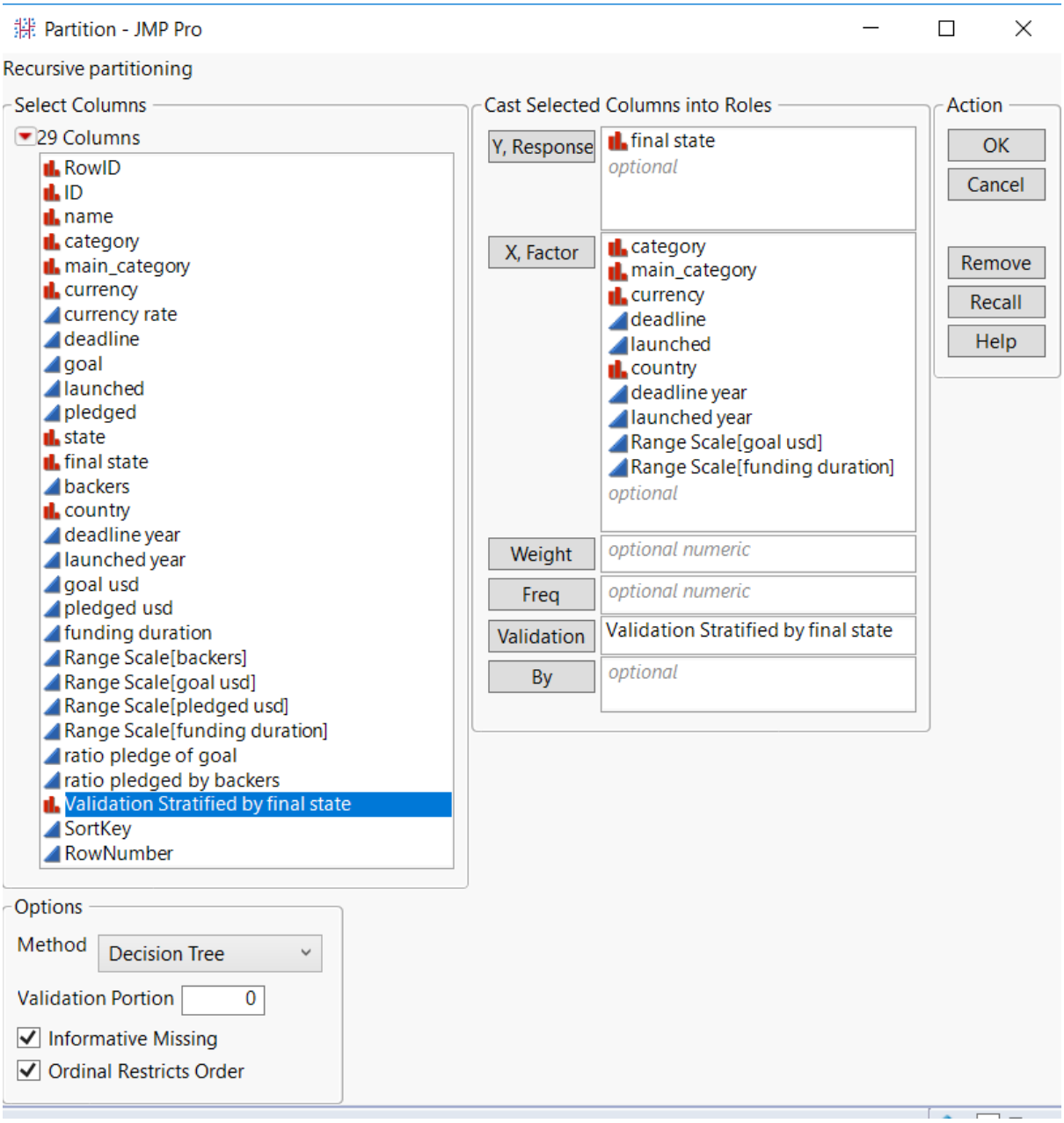
Classification Models

With Stratified split of Training set – 0.4, Validation set – 0.4, Test set = 0.2

(We have considered this proportion as this split gave maximum accuracy and minimum False positive and False negative errors)

Reference – Decision Tree Model Appendix

1.Decision Tree

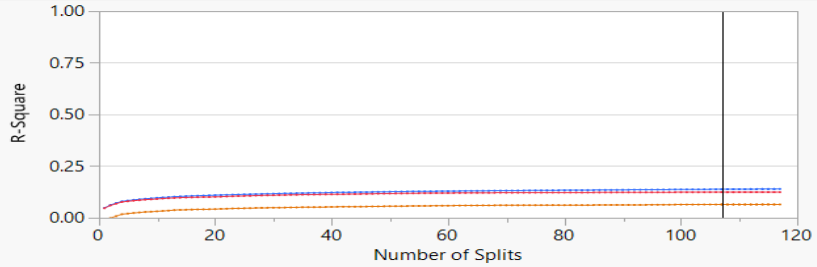


Results:

Split | Prune | Go

	RSquare	N	Number of Splits
Training	0.138	90514	107
Validation	0.124	90514	
Test	0.064	63245	

Split History



Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1379	0.1239	0.0644	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2320	0.2104	0.1106	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5975	0.6073	0.6101	∑ -Log(ρ[j])/n
RMSE	0.4541	0.4585	0.4607	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4126	0.4169	0.4177	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3269	0.3348	0.3443	∑ (ρ[j]≠ρMax)/n
N	90514	90514	63245	n

Confusion Matrix

Training

Actual final state	Predicted Count 0	1
0	28930	16327
1	13259	31998

Validation

Actual final state	Predicted Count 0	1
0	28585	16672
1	13635	31622

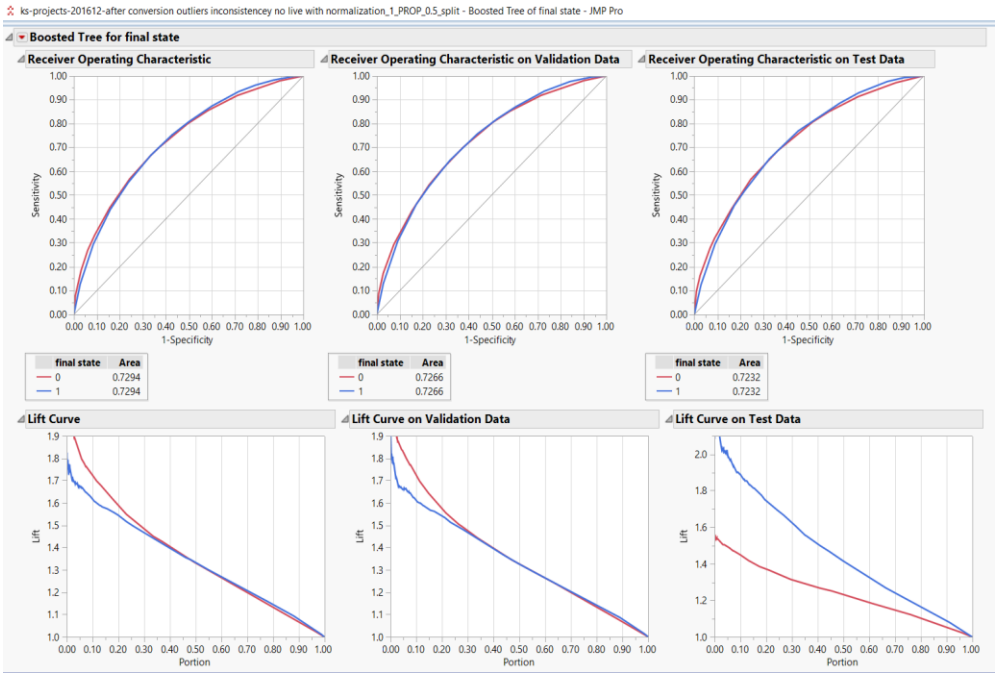
Test

Actual final state	Predicted Count 0	1
0	25693	14923
1	6855	15774

Dataset	Number of False Positive and False Negative	Accuracy
Training	29586	67.3%
Validation	30307	66.5%
Test	21778	65.5%

Obtaining an accuracy rate of 65.5% helped us to identify the best model.

2. Boosted Tree



Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1121	0.1098	0.0505	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.1919	0.1882	0.0875	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.6155	0.6171	0.6192	∑-Log(ρ[j])/n
RMSE	0.4622	0.4629	0.4643	√∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4429	0.4438	0.4442	∑|y[j]-ρ[j]|/n
Misclassification Rate	0.3342	0.3376	0.3473	∑(ρ[j]≠ρMax)/n
N	90514	90514	63245	n

Confusion Matrix

Training

Actual final state	Predicted Count 0	1
0	28616	16641
1	13613	31644

Validation

Actual final state	Predicted Count 0	1
0	28476	16781
1	13773	31484

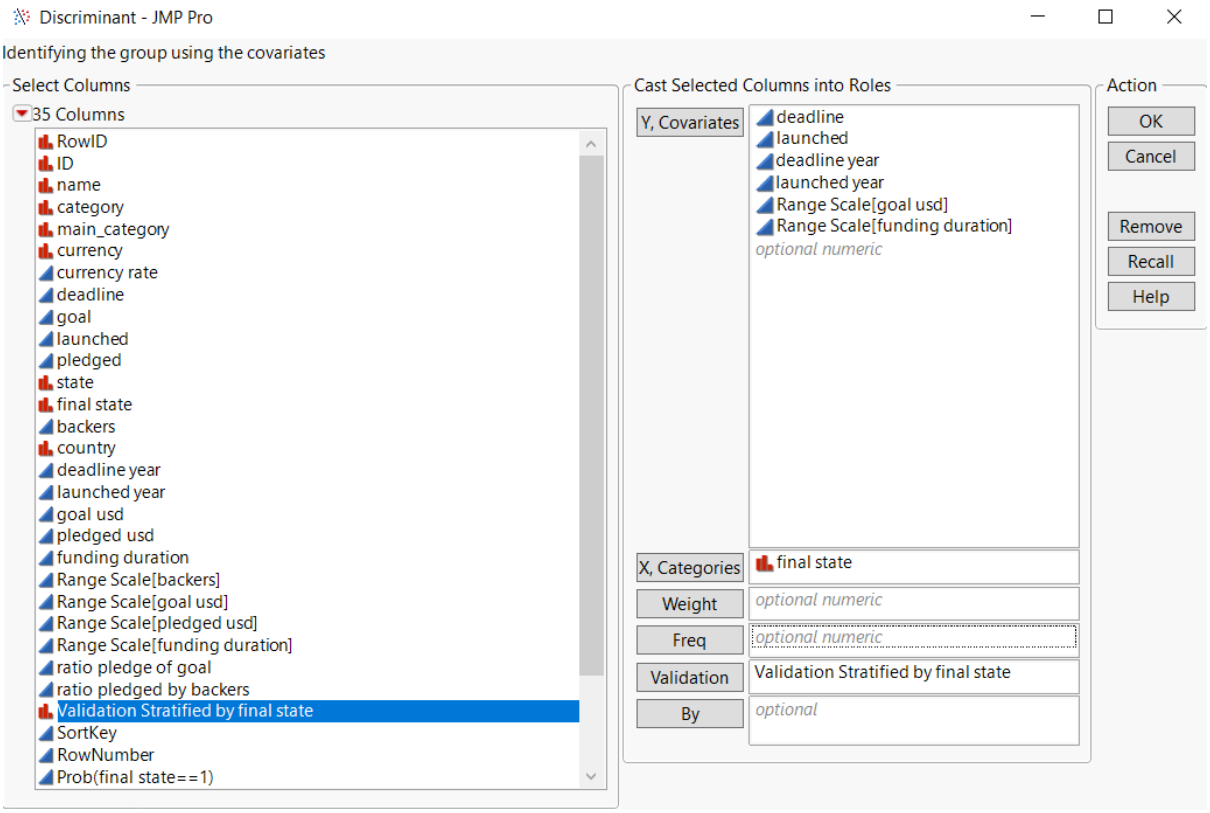
Test

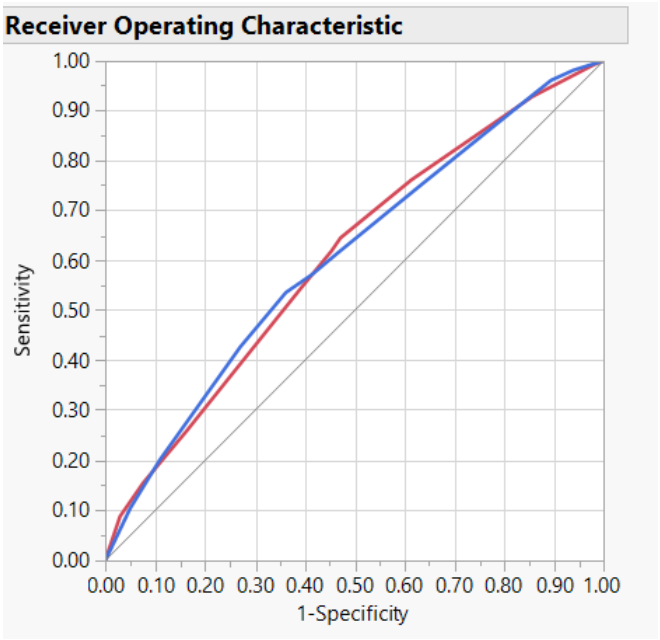
Actual final state	Predicted Count 0	1
0	25603	15013
1	6954	15675

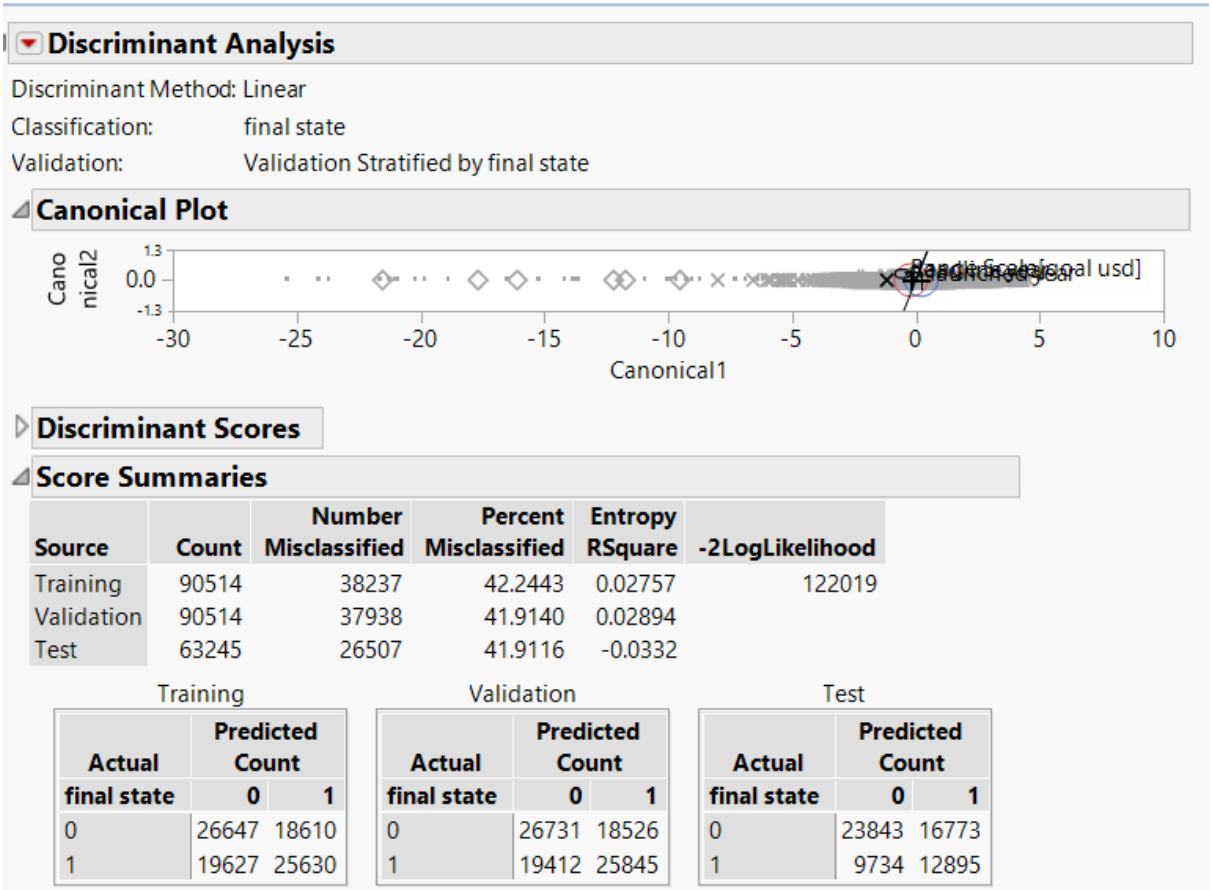
Data set	Number of False positive and False Negative errors	Accuracy
Training	30254	66.57%
Validation	30554	66.24%
Test	21967	65.26%

This model has a bit less accuracy rate than the decision tree. The Decision tree model is better than the Boosted tree model.

3.Discriminant Analysis



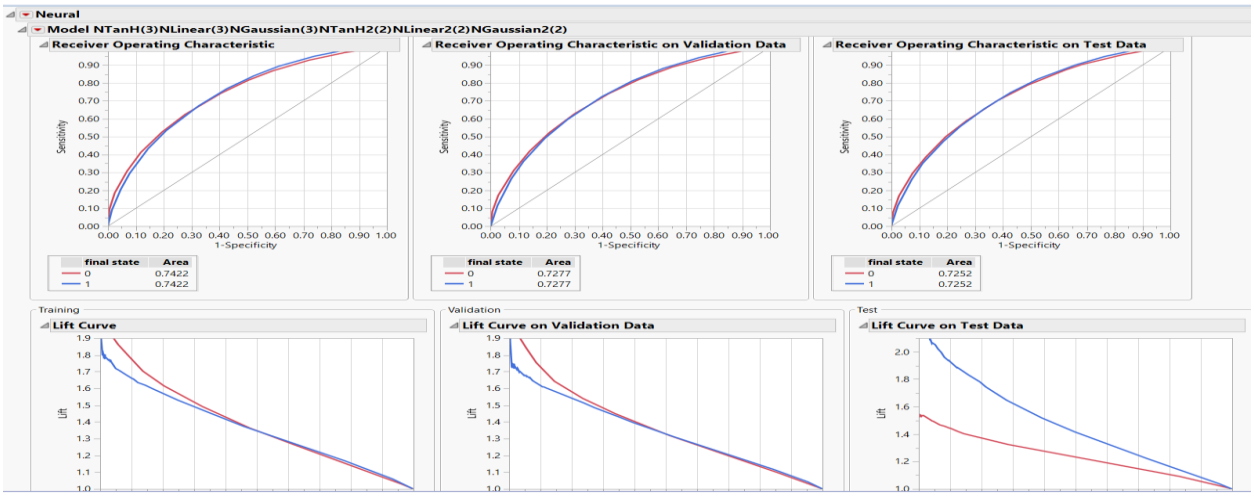




Data set	Number of False positive and False Negative errors	Accuracy
Training	38237	57.75%
Validation	37938	58.06%
Test	26507	58.08%

We are getting very low accuracy rates as compared to other models. Even the number of False Negative errors is also high. We may not consider this model. This is the model where the testing accuracy is more than the training accuracy.

4.Neural Nets



Neural

Validation Column: Validation Stratified by final state

Model Launch

Model NTanH(3)NLinear(3)NGaussian(3)NTanH2(2)NLinear2(2)NGaussian2(2)

Training

final state

Measures	Value
Generalized RSquare	0.2390434
Entropy RSquare	0.1425212
RMSE	0.4529259
Mean Abs Dev	0.4115753
Misclassification Rate	0.3263031
-LogLikelihood	53797.811
Sum Freq	90514

Confusion Matrix

Actual	Predicted Count	
final state	0	1
0	29448	15809
1	13726	31531

Confusion Rates

Actual	Predicted Rate	
final state	0	1
0	0.651	0.349
1	0.303	0.697

Validation

final state

Measures	Value
Generalized RSquare	0.2106595
Entropy RSquare	0.1240493
RMSE	0.4586873
Mean Abs Dev	0.4168741
Misclassification Rate	0.33639
-LogLikelihood	54956.729
Sum Freq	90514

Confusion Matrix

Actual	Predicted Count	
final state	0	1
0	28937	16320
1	14128	31129

Confusion Rates

Actual	Predicted Rate	
final state	0	1
0	0.639	0.361
1	0.312	0.688

Test

final state

Measures	Value
Generalized RSquare	0.107685
Entropy RSquare	0.0626491
RMSE	0.4611415
Mean Abs Dev	0.4175708
Misclassification Rate	0.3439798
-LogLikelihood	38660.734
Sum Freq	63245

Confusion Matrix

Actual	Predicted Count	
final state	0	1
0	26061	14555
1	7200	15429

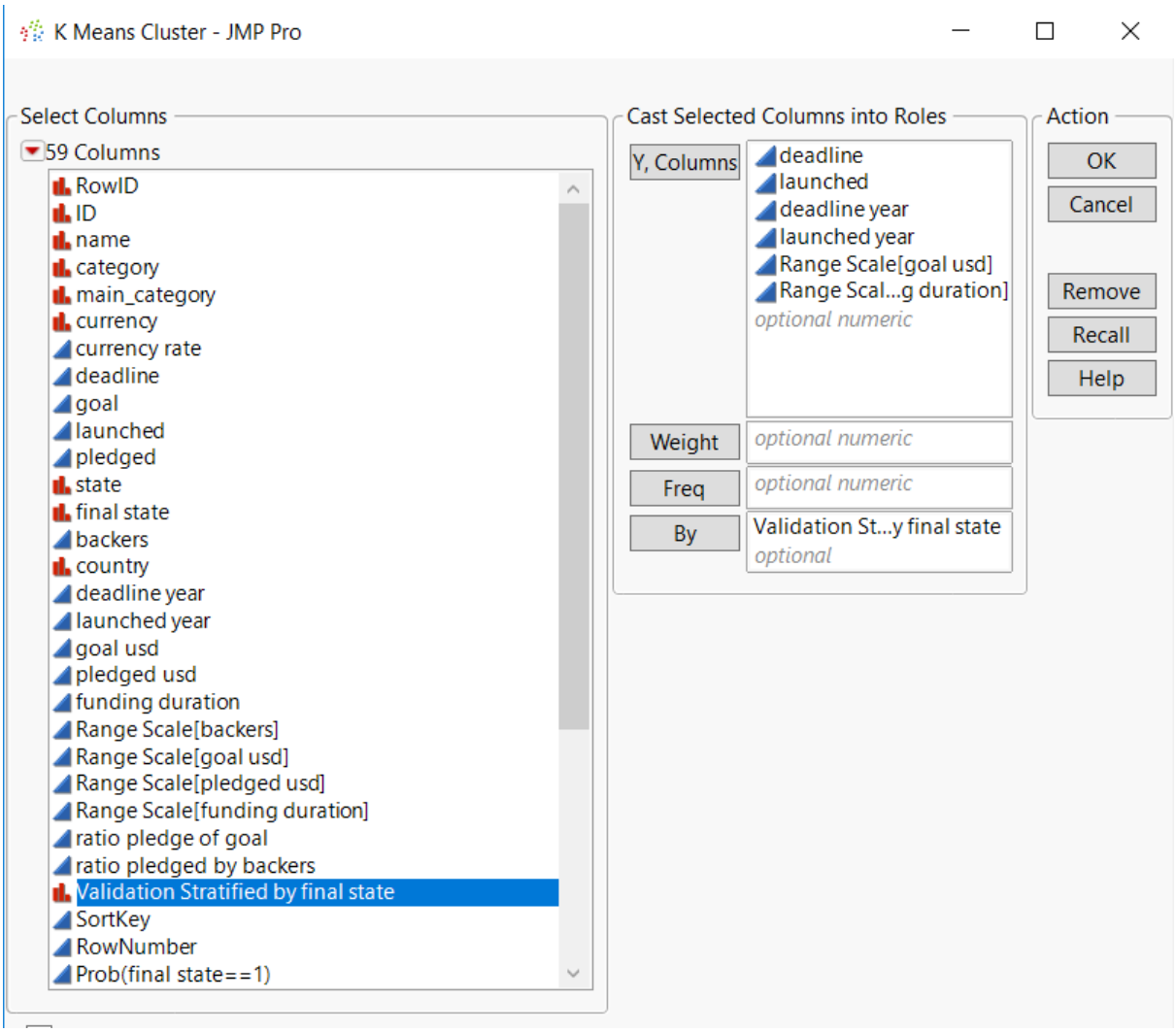
Confusion Rates

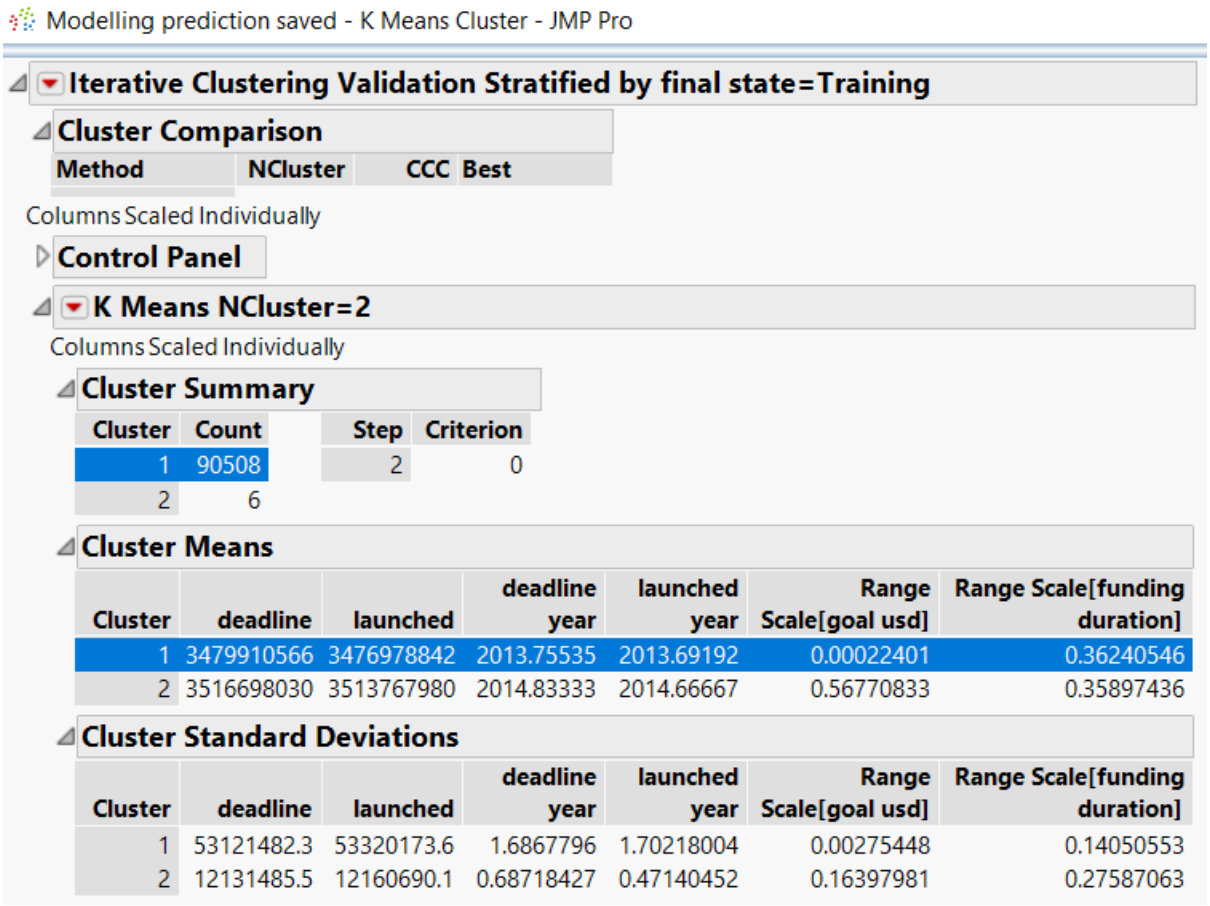
Actual	Predicted Rate	
final state	0	1
0	0.642	0.358
1	0.318	0.682

Data set	Number of False positive and False Negative errors	Accuracy
Training	29535	67.36%
Validation	30448	66.36%
Test	21755	65.06%

This model has the same accuracy rate as the Decision tree. The neural net returned high accuracy rates versus the decision tree model. However, the testing accuracy is the same as the decision tree. This might be due to overfitting with the training dataset.

5.Cluster Analysis





Modelling prediction saved - K Means Cluster - JMP Pro

Iterative Clustering Validation Stratified by final state=Training

Cluster Comparison

Method	NCluster	CCC	Best

Columns Scaled Individually

Control Panel

K Means NCluster=2

Columns Scaled Individually

Cluster Summary

Cluster	Count	Step	Criterion
1	90508	2	0
2	6		

Cluster Means

Cluster	deadline	launched	deadline year	launched year	Range Scale[goal usd]	Range Scale[funding duration]
1	3479910566	3476978842	2013.75535	2013.69192	0.00022401	0.36240546
2	3516698030	3513767980	2014.83333	2014.66667	0.56770833	0.35897436

Cluster Standard Deviations

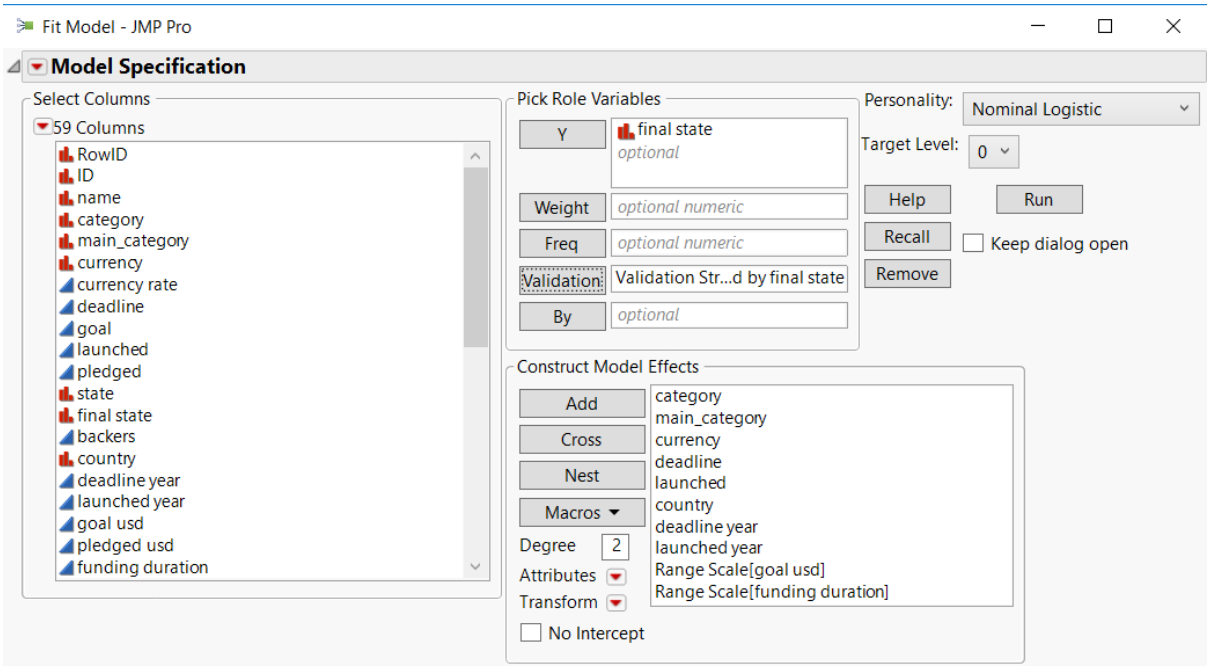
Cluster	deadline	launched	deadline year	launched year	Range Scale[goal usd]	Range Scale[funding duration]
1	53121482.3	53320173.6	1.6867796	1.70218004	0.00275448	0.14050553
2	12131485.5	12160690.1	0.68718427	0.47140452	0.16397981	0.27587063

RESULT - Here, we find two clusters – one with 90,508 and the other with 6 observations. This shows that the model does not cluster successfully. In other words, the model was unsuccessful in forming valid clusters. Also, proper predictions may not be able to do since the original dataset had 45257 projects that were successful.

REGRESSION MODELING

With Stratified split of Training set – 0.4, Validation set – 0.4, Test set = 0.2

1.Nominal Logistic regression model:



Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	29003	16254
1	14514	30743

Validation

Actual	Predicted Count	
final state	0	1
0	28933	16324
1	14495	30762

Test

Actual	Predicted Count	
final state	0	1
0	25968	14648
1	7379	15250

Data set	Number of False positive and False Negative Errors	Accuracy
Training	30768	66.00%
Validation	30819	65.95%
Test	22027	65.17%

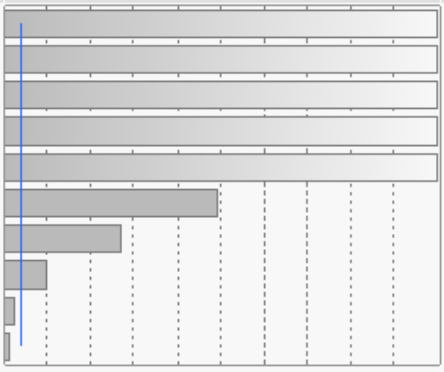
We are getting low accuracy rate compared to classification models (decision tree). Even the number of False Negative error is also high compared to other models. This model results can be used to build Ensemble models.

Modelling prediction saved - Fit Nominal Logistic - JMP Pro

Nominal Logistic Fit for final state

Effect Summary

Source	LogWorth	PValue
category	947.004	0.00000
Range Scale[goal usd]	279.250	0.00000
deadline	81.280	0.00000
launched	81.201	0.00000
Range Scale[funding duration]	75.647	0.00000
country	24.752	0.00000
currency	13.609	0.00000
main_category	5.042	0.00001
launched year	1.239	0.05764
deadline year	0.592	0.25611



Remove Add Edit ☐ FDR

Converged in Gradient, 24 iterations

Iterations

Whole Model Test

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	7238.567	196	14477.13	<.0001*
Full	55500.957			
Reduced	62739.524			

RSquare (U)	0.1154
AICc	111425
BIC	113410
Observations (or Sum Wgts)	90514

Fit Details

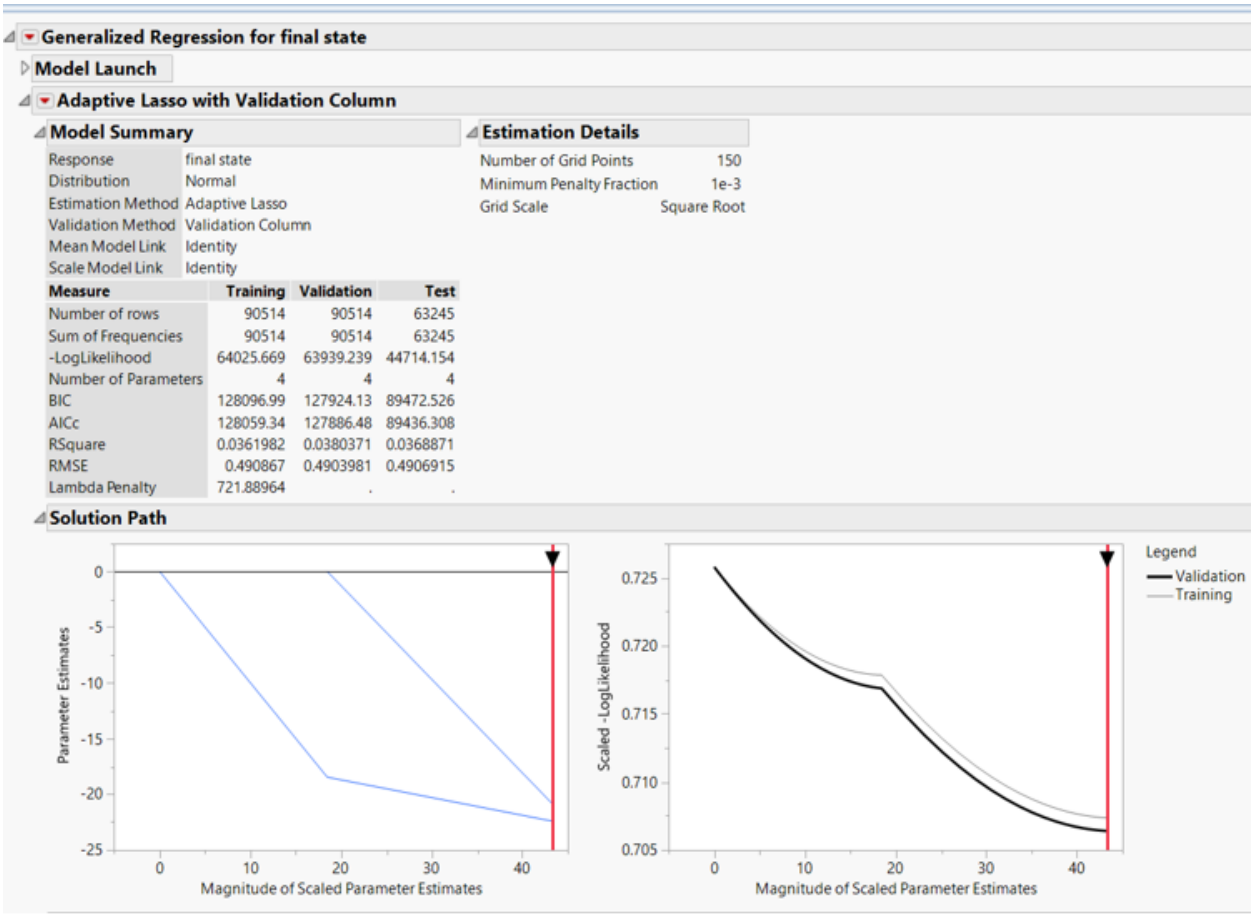
Lack Of Fit

Source	DF	-LogLikelihood	ChiSquare
Lack Of Fit	90313	55499.571	110999.1
Saturated	90509	1.386	Prob>ChiSq
Fitted	196	55500.957	<.0001*

Parameter Estimates

Term		Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	Biased	41.2723972	24201.187	0.00	0.9986
category[3D Printing]	Biased	-0.3417946	0.1865425	3.36	0.0669
category[Academic]	Biased	0.8685517	0.1860089	21.80	<.0001*
category[Accessories]	Biased	0.18734392	0.1096951	2.92	0.0877
category[Action]	Biased	1.82315007	0.8129966	5.03	0.0249*
category[Animals]	Biased	0.85156719	0.2687636	10.04	0.0015*

2.Generalized Regression:



We are getting very low R Square value versus other models because we are not using some of the nominal variables.

3.Step Wise regression:

Crossvalidation

Source	RSquare	RASE	Freq
Training Set	0.1242	0.46793	90514
Validation Set	0.1242	0.46791	90514
Test Set	0.0388	0.46996	63245

SSE	DFE	RMSE	RSquare	RSquare Adj	Cp	p	AICc	BIC	RSquare Validation	RMSE Validation	RSquareTest	RMSE Test
19818.799	90458	0.4680747	0.1242	0.1236	20.108722	56	119502.4	120038.8	0.1242	0.467911	0.0388	0.469964

Summary of Fit

RSquare	0.12417
RSquare Adj	0.123637
Root Mean Square Error	0.468074
Mean of Response	0.5
Observations (or Sum Wgts)	90514

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	55	2809.776	51.0868	233.1741
Error	90458	19818.724	0.2191	Prob > F
C. Total	90513	22628.500		<.0001*

The Stepwise model is getting the same R square as the Standard Least Squares. We can consider this model for creating ensemble models.

4.Standard Least Squares:

Modelling prediction saved - Fit Least Squares - JMP Pro

Response final state

Validation: Validation Stratified by final state

Singularity Details

Term	Details
Intercept	=category[3D Printing] + category[Academic] + category[Accessories] + ...
currency[CHF]	=currency[HKD] + country[CH] - country[HK]=currency[MXN] + country[...

Effect Summary

Source	LogWorth	PValue
category	768.139	0.00000
deadline	79.428	0.00000
launched	79.319	0.00000
Range Scale[funding duration]	72.895	0.00000
country	42.859	0.00000
Range Scale[goal usd]	10.857	0.00000
main_category	4.600	0.00003
launched year	1.245	0.05688
deadline year	0.364	0.43219
currency	0.001	0.99860

Remove Add Edit FDR

Summary of Fit

RSquare	0.125188
RSquare Adj	0.123289
Root Mean Square Error	0.468167
Mean of Response	0.5
Observations (or Sum Wgts)	90514

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	196	2832.814	14.4531	65.9418
Error	90317	19795.686	0.2192	Prob > F
C. Total	90513	22628.500		<.0001*

Parameter Estimates

Term		Estimate	Std Error	t Ratio	Prob>|t|
Intercept	Biased	-12.22258	12.10534	-1.01	0.3126
category[3D Printing]	Biased	0.0294159	0.040431	0.73	0.4669
category[Academic]	Biased	-0.179375	0.038305	-4.68	<.0001*
category[Accessories]	Biased	-0.023833	0.023521	-1.01	0.3109
category[Action]	Biased	-0.25418	0.162243	-1.57	0.1172
category[Animals]	Biased	-0.164266	0.058739	-2.80	0.0052*
category[Animation]	Biased	-0.154749	0.158539	-0.98	0.3290

Effect Tests

Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F	
category	157	148	933.86711	28.7887	<.0001*	LostDFs
main_category	14	6	6.79999	5.1708	<.0001*	LostDFs
currency	12	5	0.05299	0.0484	0.9986	LostDFs
deadline	1	1	78.93857	360.1540	<.0001*	
launched	1	1	78.82850	359.6518	<.0001*	
deadline year	1	1	0.13522	0.6169	0.4322	
launched year	1	1	0.79477	3.6261	0.0569	
Range Scale[goal usd]	1	1	10.01585	45.6969	<.0001*	
Range Scale[funding duration]	1	1	72.33856	330.0417	<.0001*	
country	21	15	53.74880	16.3484	<.0001*	LostDFs

Crossvalidation

Source	RSquare	RASE	Freq
Training Set	0.1252	0.46766	90514
Validation Set	0.1232	0.46818	90514
Test Set	0.0382	0.47010	63245

Model	Training R square	Validation R square	Test R square
Nominal Logistic Regression	0.1033	0.1024	0.1154
Generalized Regression	0.031982	0.0380371	0.0368871
Stepwise	0.1242	0.1242	0.0388
Standard Least Squares	0.1252	0.1232	0.0382

RESULT – We can conclude that the R square value is greater for Nominal Logistic Regression Model as compared to R square values of other three models. Thus, Nominal Logistic Regression Model is a better model than the other three models.

Ensemble Models

With Stratified split of Training set – 0.4, Validation set – 0.4, Test set = 0.2

1. Decision Tree and Boosted tree Ensemble Model

Modelling prediction saved - Boosted Tree of final state - JMP Pro

Boosted Tree for final state

Specifications

Target Column:	final state	Number of training rows:	90514
Validation Column:	Validation Stratified by final state	Number of validation rows:	90514
Number of Layers:	50	Number of test rows:	63245
Splits per Tree:	3		
Learning Rate:	0.1		
Overfit Penalty:	0.0001		

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1121	0.1098	0.0505	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.1919	0.1882	0.0875	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.6155	0.6171	0.6192	∑ -Log(ρ[j])/n
RMSE	0.4622	0.4629	0.4643	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4429	0.4438	0.4442	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3342	0.3376	0.3473	∑ (ρ[j]≠ρMax)/n
N	90514	90514	63245	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	28616	16641
1	13613	31644

Validation

Actual	Predicted Count	
final state	0	1
0	28476	16781
1	13773	31484

Test

Actual	Predicted Count	
final state	0	1
0	25603	15013
1	6954	15675

2. Neural and decision Tree Ensemble Model

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	29070	16187
1	12371	32886

Validation

Actual	Predicted Count	
final state	0	1
0	28625	16632
1	13014	32243

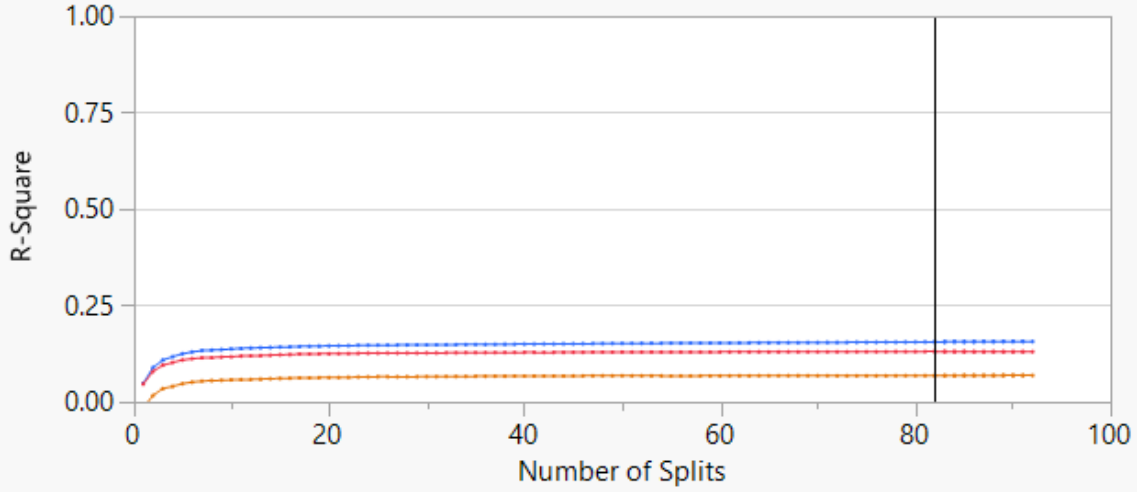
Test

Actual	Predicted Count	
final state	0	1
0	25549	15067
1	6600	16029

Split | Prune | Go

	RSquare	N	Number of Splits
Training	0.155	90514	82
Validation	0.130	90514	
Test	0.068	63245	

Split History



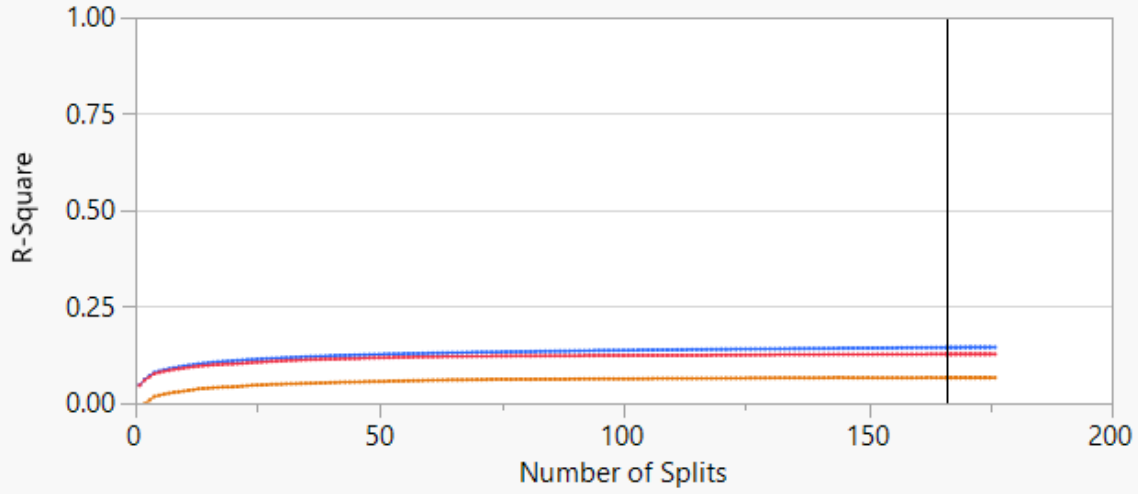
Validation Data in Red
Test Data in Orange

3. Nominal Logistic and decision tree Ensemble Model

Split Prune Go

	RSquare	N	Number of Splits
Training	0.144	90514	166
Validation	0.127	90514	
Test	0.066	63245	

Split History



Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1445	0.1271	0.0659	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2420	0.2154	0.1131	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5930	0.6050	0.6091	∑ -Log(ρ[j])/n
RMSE	0.4520	0.4575	0.4603	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4088	0.4140	0.4154	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3200	0.3316	0.3347	∑ (ρ[j]≠ρMax)/n
N	90514	90514	63245	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	30566	14691
1	14269	30988

Validation

Actual	Predicted Count	
final state	0	1
0	30111	15146
1	14869	30388

Test

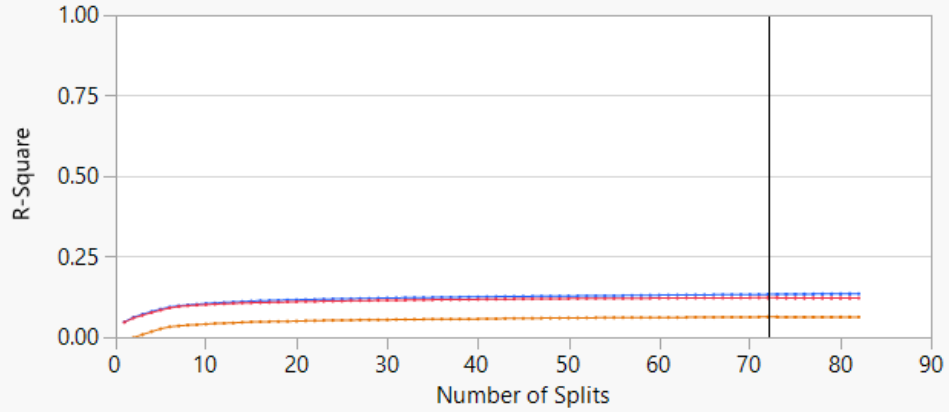
Actual	Predicted Count	
final state	0	1
0	26991	13625
1	7543	15086

4. Step wise and decision Tree Ensemble Model

Split | Prune | Go

	RSquare	N	Number of Splits
Training	0.133	90514	72
Validation	0.122	90514	
Test	0.063	63245	

Split History



Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1329	0.1223	0.0628	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2243	0.2079	0.1079	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.6011	0.6084	0.6112	∑ -Log(ρ[j])/n
RMSE	0.4556	0.4591	0.4611	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4153	0.4190	0.4196	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3291	0.3355	0.3464	∑ (ρ[j]≠ρMax)/n
N	90514	90514	63245	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	28539	16718
1	13073	32184

Validation

Actual	Predicted Count	
final state	0	1
0	28299	16958
1	13410	31847

Test

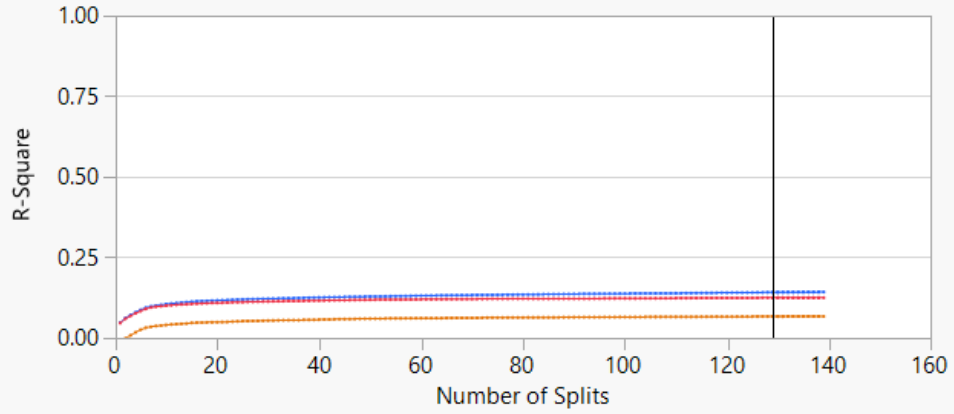
Actual	Predicted Count	
final state	0	1
0	25497	15119
1	6787	15842

5.Standard Least Square and decision tree Ensemble Model

Split | Prune | Go

	RSquare	N	Number of Splits
Training	0.141	90514	129
Validation	0.125	90514	
Test	0.066	63245	

Split History



Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1414	0.1247	0.0664	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2374	0.2117	0.1138	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5951	0.6067	0.6089	∑ -Log(ρ[j])/n
RMSE	0.4530	0.4582	0.4601	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4105	0.4158	0.4162	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3236	0.3332	0.3323	∑ (ρ[j]≠ρMax)/n
N	90514	90514	63245	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	31005	14252
1	15042	30215

Validation

Actual	Predicted Count	
final state	0	1
0	30559	14698
1	15464	29793

Test

Actual	Predicted Count	
final state	0	1
0	27460	13156
1	7858	14771

6. Discriminant Analysis and Decision Tree Ensemble Model

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1383	0.1238	0.0623	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2326	0.2102	0.1070	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5973	0.6074	0.6115	∑ -Log(ρ[j])/n
RMSE	0.4539	0.4586	0.4613	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4122	0.4170	0.4181	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3247	0.3354	0.3434	∑ (ρ[j]≠ρMax)/n
N	90514	90514	63245	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	29488	15769
1	13622	31635

Validation

Actual	Predicted Count	
final state	0	1
0	29011	16246
1	14111	31146

Test

Actual	Predicted Count	
final state	0	1
0	26095	14521
1	7197	15432

Split Prune Go

	RSquare	N	Number of Splits
Training	0.138	90514	124
Validation	0.124	90514	
Test	0.062	63245	

7. Discriminant Analysis and Neural Nets Ensemble Model

Model NTanH(2)NLinear(2)NGaussian(2)NTanH2(1)NLinear2(1)NGaussian2(1)

Training

final state

Measures	Value
Generalized RSquare	0.1921176
Entropy RSquare	0.112233
RMSE	0.4622009
Mean Abs Dev	0.4295484
Misclassification Rate	0.3448638
-LogLikelihood	55698.081
Sum Freq	90514

Confusion Matrix

Actual	Predicted Count	
final state	0	1
0	29138	16119
1	15096	30161

Confusion Rates

Actual	Predicted Rate	
final state	0	1
0	0.644	0.356
1	0.334	0.666

Validation

final state

Measures	Value
Generalized RSquare	0.1858493
Entropy RSquare	0.1082817
RMSE	0.4636056
Mean Abs Dev	0.4310439
Misclassification Rate	0.3481671
-LogLikelihood	55945.979
Sum Freq	90514

Confusion Matrix

Actual	Predicted Count	
final state	0	1
0	28962	16295
1	15219	30038

Confusion Rates

Actual	Predicted Rate	
final state	0	1
0	0.640	0.360
1	0.336	0.664

Test

final state

Measures	Value
Generalized RSquare	0.0850065
Entropy RSquare	0.049023
RMSE	0.4651405
Mean Abs Dev	0.4313986
Misclassification Rate	0.3489446
-LogLikelihood	39222.741
Sum Freq	63245

Confusion Matrix

Actual	Predicted Count	
final state	0	1
0	26163	14453
1	7616	15013

Confusion Rates

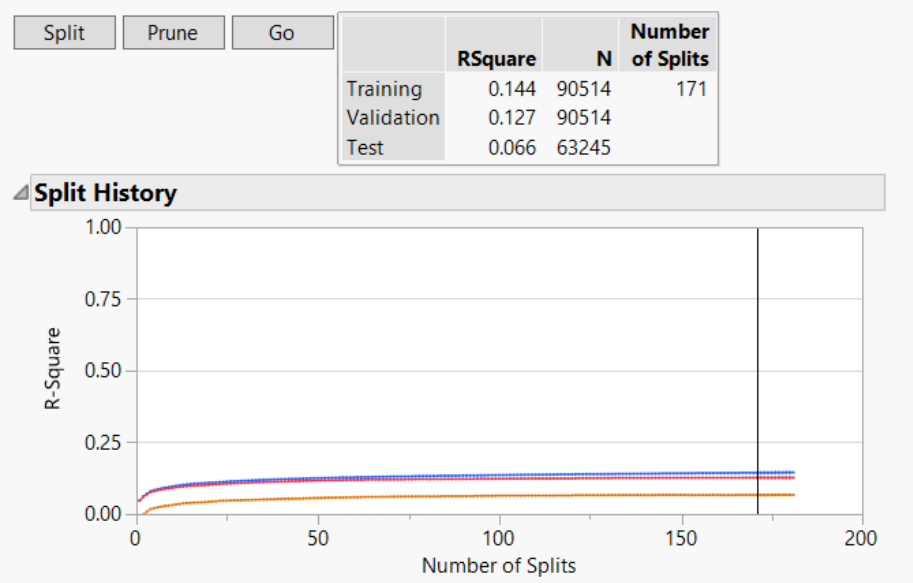
Actual	Predicted Rate	
final state	0	1
0	0.644	0.356
1	0.337	0.663

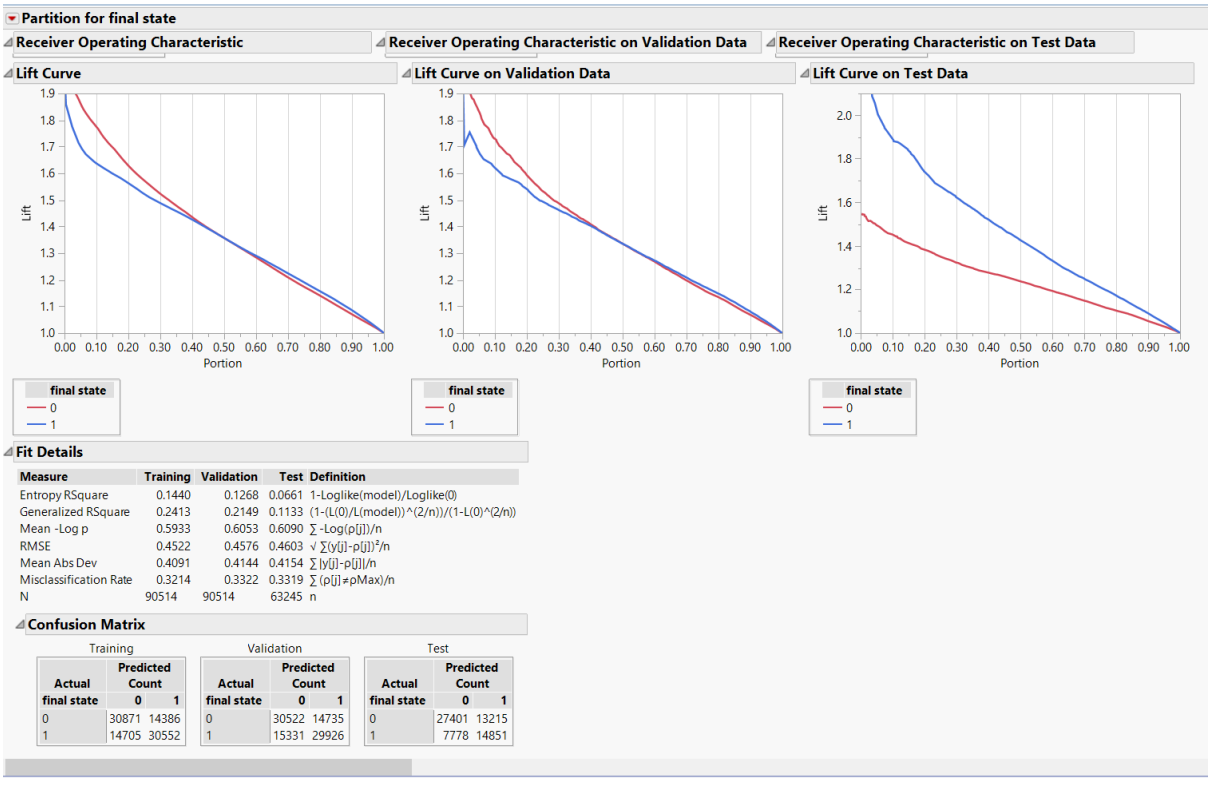
8. Generalized Regression and Stepwise Ensemble Model

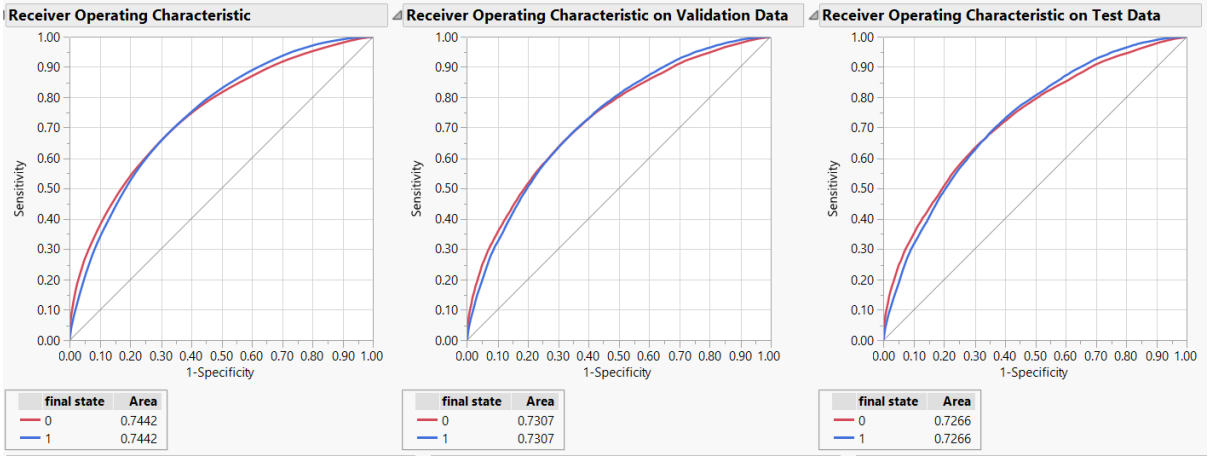
Crossvalidation

Source	RSquare	RASE	Freq
Training Set	0.1242	0.46793	90514
Validation Set	0.1242	0.46791	90514
Test Set	0.0387	0.46998	63245

9. Generalized Regression and Decision Tree Ensemble Model







10. Generalized Regression and Standard Least Squares Ensemble Model

Response final state

Validation: Validation Stratified by final state

Singularity Details

Term	Details
Intercept	=category[3D Printing] + category[Academic] + category[Accessories] + ...

Effect Summary

Source	LogWorth	PValue
category	770.026	0.00000
deadline	79.559	0.00000
launched	79.446	0.00000
currency	20.956	0.00000
Range Scale[goal usd]	10.784	0.00000
main_category	4.562	0.00003
launched year	1.349	0.04480
deadline year	0.399	0.39925
Prediction Generalized regreesion	.	.
Range Scale[funding duration]	.	.

Remove Add Edit ☐ FDR

Summary of Fit

RSquare	0.122813
RSquare Adj	0.121055
Root Mean Square Error	0.468763
Mean of Response	0.5
Observations (or Sum Wgts)	90514

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	181	2779.065	15.3540	69.8737
Error	90332	19849.435	0.2197	Prob > F
C. Total	90513	22628.500		<.0001*

COMPARISON OF MODELS

Models	False Negative for Test	False Positive for Test	R Square Training	Accuracy Training	R Square validation	Accuracy validation	R Square Test	Accuracy Test
Decision Tree (DT)	16327	13259	0.2320	67.3	0.2104	66.5	0.1106	65.5
Boosted Tree (BT)	16641	13613	0.1919	66.57	0.1882	66.24	0.0875	65.26
Discriminant Analysis (DA)	18610	19627		57.75		58.06		58.08
Neural Nets (NN)	15809	13726	0.239	67.36	0.21065	66.36	0.1076	65.06
Nominal logistic regression (NL)	-	-	0.1154	66.00		65.95		65.17
Generalized regression	-	-	0.036198	-	0.03803	-	0.03688	-
Step wise (SW)	-	-	0.1242	-	0.1242	-	0.0388	-
Standard Least Squares (SL)	-	-	0.1252	-	0.1232	-	0.0382	-
Ensemble 01 DT vs BT	28616	16641	0.1919	66.5	0.1882	66.24	0.0875	65.26
Ensemble 02 NN vs DT	16187	12371	0.155	68.004	0.130	67.2	0.068	65.7
Ensemble 03 NL vs DT	14691	14269	0.2420	68.004	0.2154	66.839	0.1131	66.53
Ensemble 04 SW vs DT	16718	13073	0.133	67.08	0.122	66.44	0.063	65.3
Ensemble 05 SL vs DT	14252	15042	0.141	67.63	0.125	66.67	0.066	66.77
Ensemble 06 DA vs DT	14521	7197	0.2326	67.5	0.2102	66.4	0.1070	65.66
Ensemble 07 DA vs NN	14453	7616	0.192	65.51	0.1858	65.183	0.085006	65.10
Ensemble 08 GR vs SW	-	-	0.124	-	0.124	-	0.0387	-
Ensemble 09 GR vs DT	13215	7778	0.2413	67.86	0.2149	66.78	0.1133	65.5
Ensemble 10 GR vs SL	-	-	0.1228	-	0.1214	-	0.0371	-

*Since Regression models consider only continuous target variables, only R square values can be compared.

COST OF FALSE POSITIVE AND FALSE NEGATIVE ERROR COMPARISON FOR TEST DATASET

Models	False Negative	False Positive	Total Cost of Error	R Square for Test dataset	Accuracy (%)
Decision Tree (DT)	6855	14923	$1159998.31	0.1106	65.5
Boosted Tree (BT)	6954	15013	$1168175.62	0.0875	65.26
Discriminant Analysis (DA)	9734	16773	$1345381.02		58.08
Neural Nets (NN)	7200	15429	$1201637.28	0.1076	65.06
Nominal logistic regression (NL)	-	-	-		65.17
Generalized regression	-	-	-	0.03688	-
Step wise (SW)	-	-	-	0.0388	-
Standard Least Squares (SL)	-	-	-	0.0382	-
Ensemble 01 DT vs BT	6954	15013	$1168175.62	0.0875	65.26
Ensemble 02 NN vs DT	6600	15067	$1164611.44	0.068	65.7
Ensemble 03 NL vs DT	7543	13625	$1085416.07	0.1131	66.53
Ensemble 04 SW vs DT	6787	15119	$1171995.71	0.063	65.3
Ensemble 05 SL vs DT	7858	13156	$1059828.34	0.066	66.77
Ensemble 06 DA vs DT	14521	7197	$789234.33	0.1070	65.66
Ensemble 07 DA vs NN	14453	7616	$816467.09	0.085006	65.10
Ensemble 08 GR vs SW	-	-	-	0.0387	-
Ensemble 09 GR vs DT	13215	7778	$802168.31	0.1133	65.5
Ensemble 10 GR vs SL	-	-	-	0.0371	-

PROPOSED MODEL – DECISION TREE MODEL

Split proportion: Training set – 0.4 Validation set – 0.4 Test set – 0.2

Cut-off value	Training False negative	Training False Positive	Training Accuracy	Validation False negative	Validation False Positive	Validation accuracy	Test False negative	Test False Positive	Test Accuracy (%)
0.5	16327	13259	67.3	16672	13635	66.51	14923	6855	65.56
0.4	8710	22221	65.82	9000	22578	65.11	8185	11383	69.06
0.3	32870	2479	60.94	33194	2673	60.37	29832	1331	49.27
0.6	8710	22221	65.82	9000	22578	65.11	8185	11383	69.06

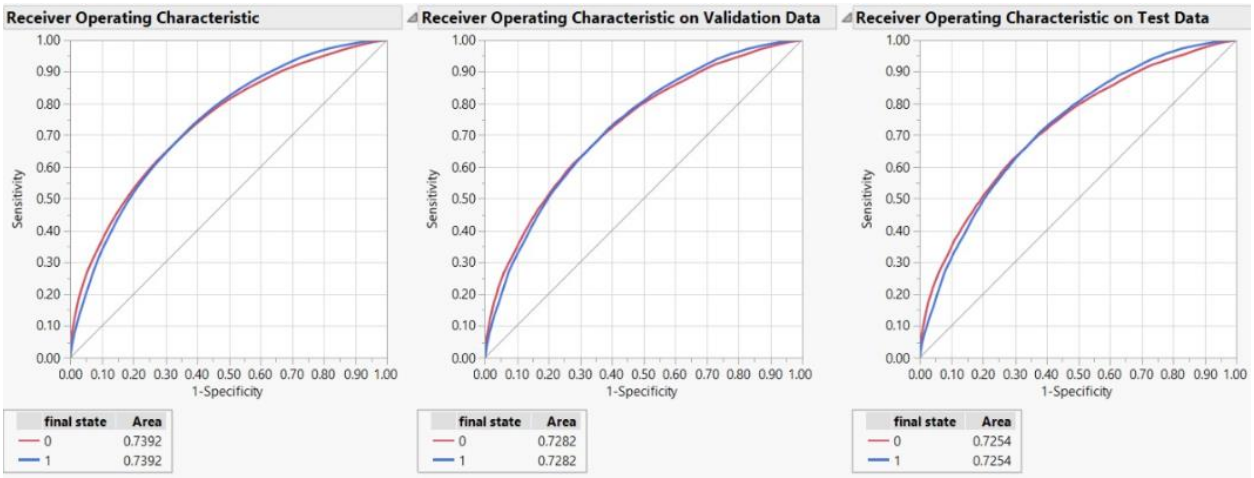
CONCLUSION

CASE 01:(Predicting Success before the start of funding date)

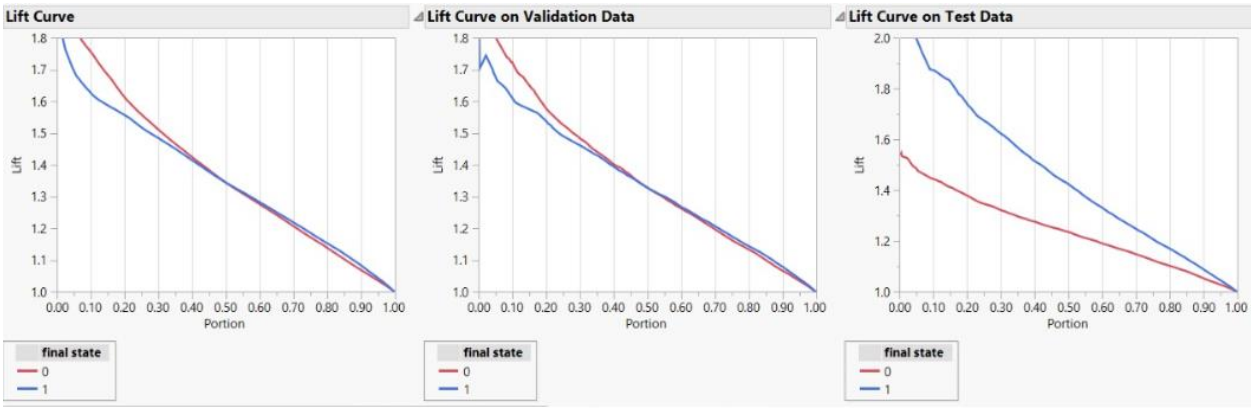
Cost of Errors:

Considering Cost of False Negative (Actual - Successful, predicted – Unsuccessful) as $20.49 and cost of False Positive (Actual - Unsuccessful, Predicted – Successful) as $68.32

Cut-off value	Test False Negative	Test False Positive	Total cost of error	Accuracy (%)
0.5	14923	6855	$774195.4	65.56
0.4	8185	11383	$945397.21	69.06
0.3	29832	1331	$702191.6	49.27
0.6	8185	11383	$945397.21	69.06



We are getting Area under curve as 0.72 for the test data set and even same for the validation set.



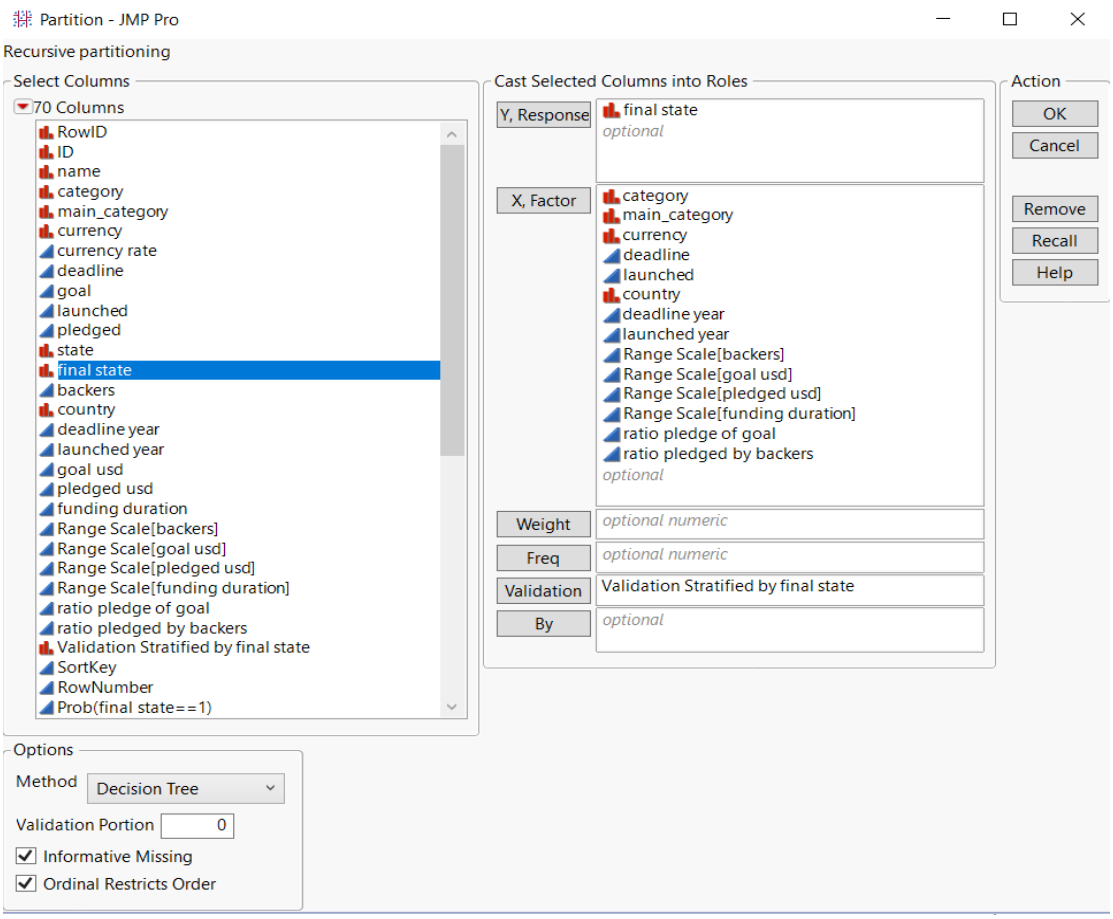
CONCLUSION - Since the estimated cost of error of each type is an approximate value, and it varies from project to project. Thus, it is not right to choose the best model solely on the basis of least cost of error. We should also consider accuracy along with cost of each type of error.

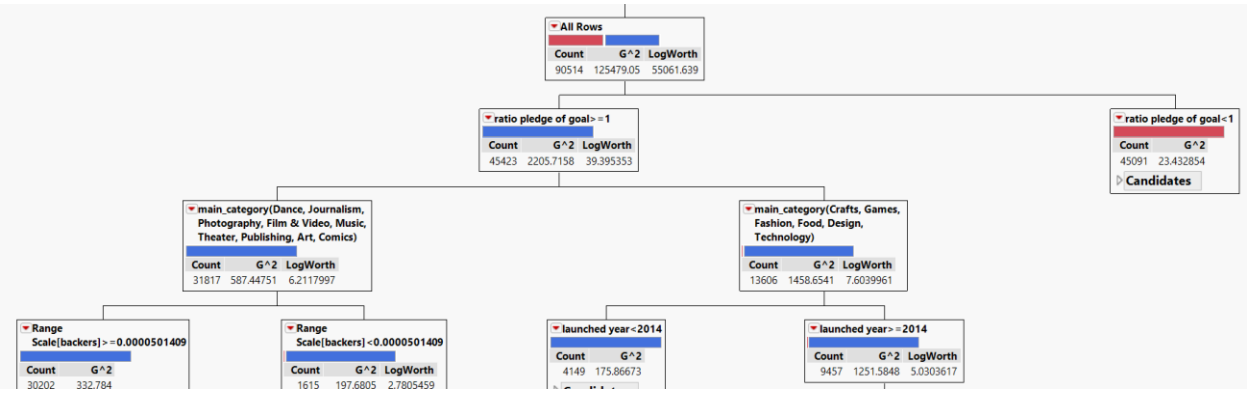
Since, the number of False positive errors are less and good accuracy of 65.56% for cut-off value 0.5, thus this model performs better than other models and we propose Decision tree model with cut-off value of 0.5 as our final model for prediction.

CASE 02: (Predicting Success just before the deadline of the project by considering pledged USD and number of backers as Explanatory variables)

We are suggesting the person to invest in a project on the day before deadline of the project which helps model to make use of the pledged USD and number of backers.

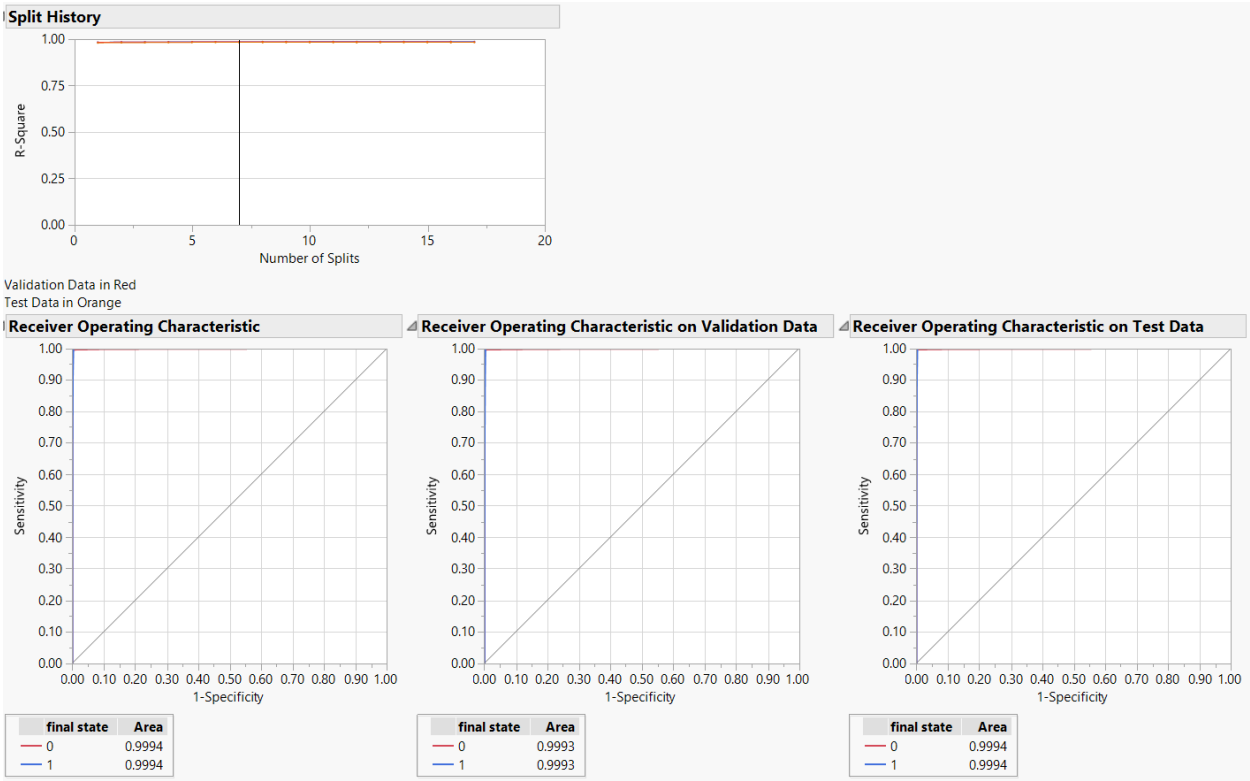
Considering the below variables for building decision tree model.





The ratio of pledge to goal does the major split for the success rate of the project.

The next major factor in deciding the success is main category of the project.



If we see area under the curve, we are getting 0.9994 sq. units in all the training, validation and testing datasets. We can conclude that the model is performing well with more accuracy.

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	45090	167
1	1	45256

Validation

Actual	Predicted Count	
final state	0	1
0	45093	164
1	3	45254

Test

Actual	Predicted Count	
final state	0	1
0	40478	138
1	1	22628

Also, we can see that there are less number of each type of errors, in total there are 139 errors in test dataset.

Overall, we obtain 99.765% accuracy. So, if someone is going to invest in starter project just before funding duration then they can be 99.765% accurate in their prediction.

APPENDIX

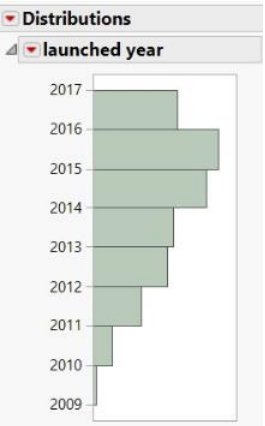
APPENDIX 1.1 - OUTLIER ANALYSIS DISTRIBUTION FOR NEW VARIABLES

Deadline Distribution

Quantiles

100.0%	maximum	2017
99.5%		2016
97.5%		2016
90.0%		2016
75.0%	quartile	2015
50.0%	median	2014
25.0%	quartile	2012
10.0%		2011
2.5%		2010
0.5%		2010
0.0%	minimum	2009

Launched year Distribution - Launched year with 1970 was deleted.



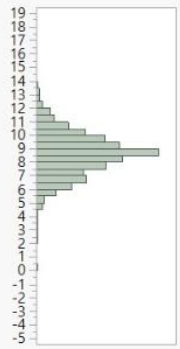
Quantiles

100.0%	maximum	2016
99.5%		2016
97.5%		2016
90.0%		2016
75.0%	quartile	2015
50.0%	median	2014
25.0%	quartile	2012
10.0%		2011
2.5%		2010
0.5%		2010
0.0%	minimum	2009

Goal_USD Distribution

Distributions

df



Quantiles

100.0%	maximum	18.667540822
99.5%		13.122363377
97.5%		11.736069016
90.0%		10.572598155
75.0%	quartile	9.6158054801
50.0%	median	8.5171931914
25.0%	quartile	7.6009024595
10.0%		6.4614681764
2.5%		5.2983173665
0.5%		2.9957322736
0.0%	minimum	-4.605170186

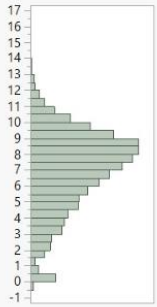
Summary Statistics

Mean	8.5354328
Std Dev	1.6849383
Std Err Mean	0.0034092
Upper 95% Mean	8.5421146
Lower 95% Mean	8.5287509
N	244273

Pledged_USD

Distributions

df



Quantiles

100.0%	maximum	16.828050108
99.5%		12.530764457
97.5%		11.133592907
90.0%		9.8416972508
75.0%	quartile	8.7393762816
50.0%	median	7.4073177105
25.0%	quartile	5.4258306868
10.0%		3.295836866
2.5%		0.6931471806
0.5%		0
0.0%	minimum	-0.597837001

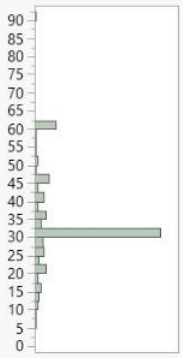
Summary Statistics

Mean	6.9687011
Std Dev	2.5711465
Std Err Mean	0.005542
Upper 95% Mean	6.9795634
Lower 95% Mean	6.9578388
N	215235

FUNDING DURATION

Distributions

funding duration



Quantiles

100.0%	maximum	92
99.5%		90
97.5%		60
90.0%		59
75.0%	quartile	37
50.0%	median	30
25.0%	quartile	30
10.0%		21
2.5%		14
0.5%		7
0.0%	minimum	1

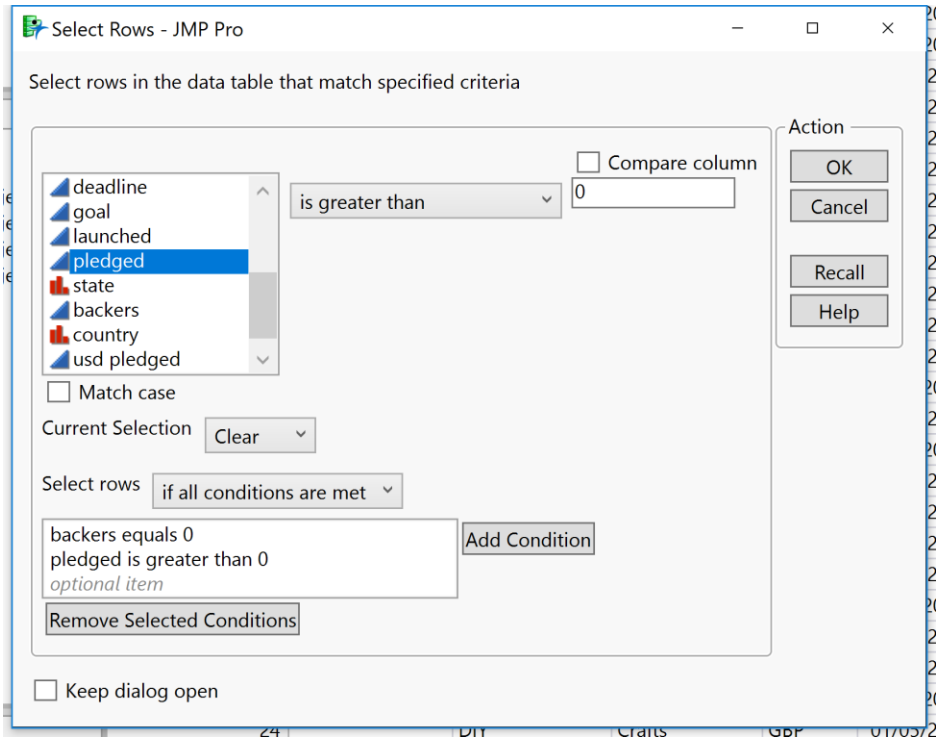
Summary Statistics

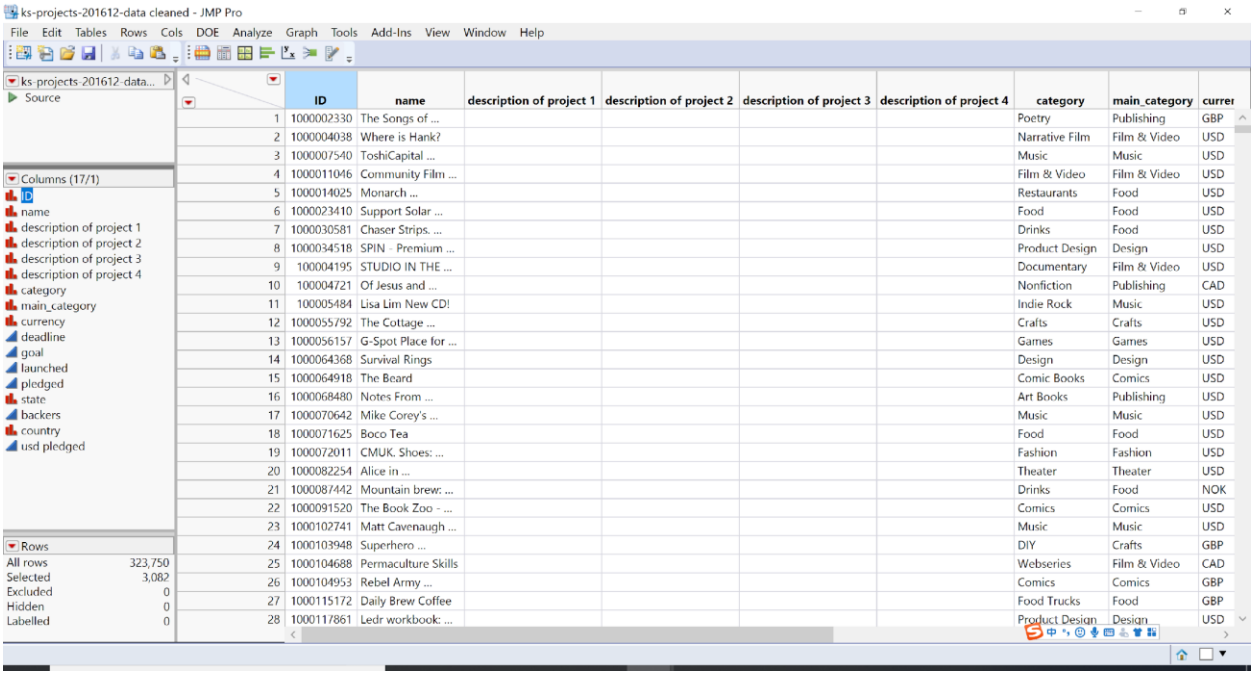
Mean	34.102181
Std Dev	12.829
Std Err Mean	0.025957
Upper 95% Mean	34.153056
Lower 95% Mean	34.051306
N	244273

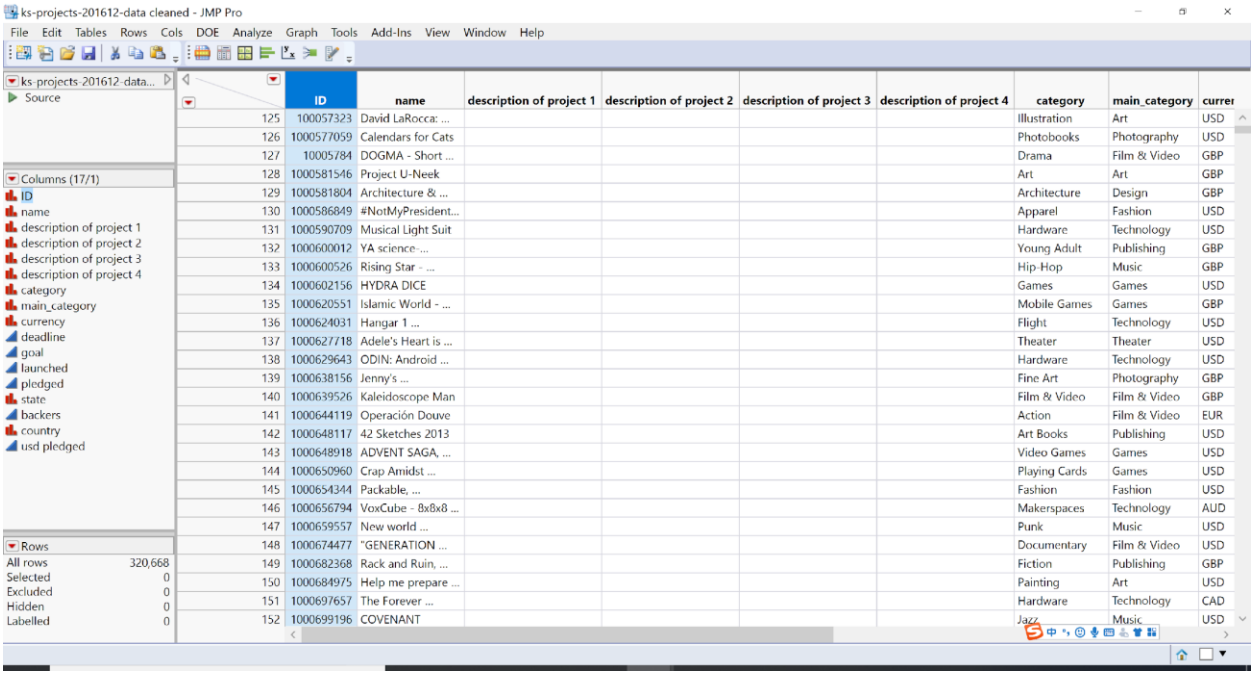
APPENDIX 1.2

Step 1

Delete the inconsistency – Backers = 0, Pledged amount is having some value

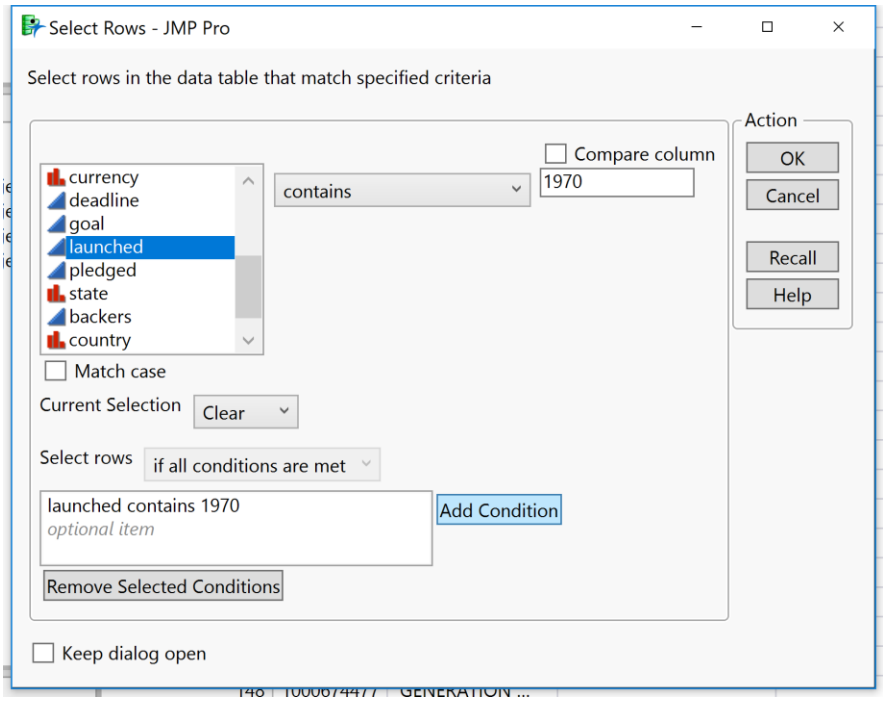


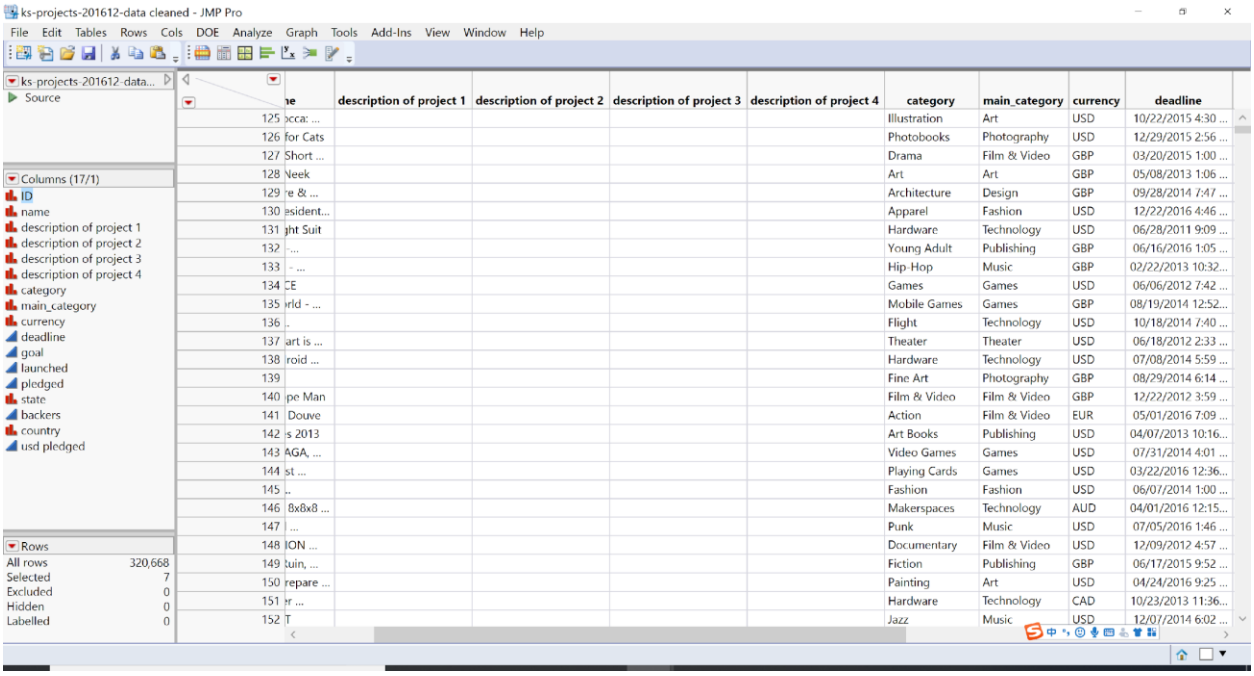


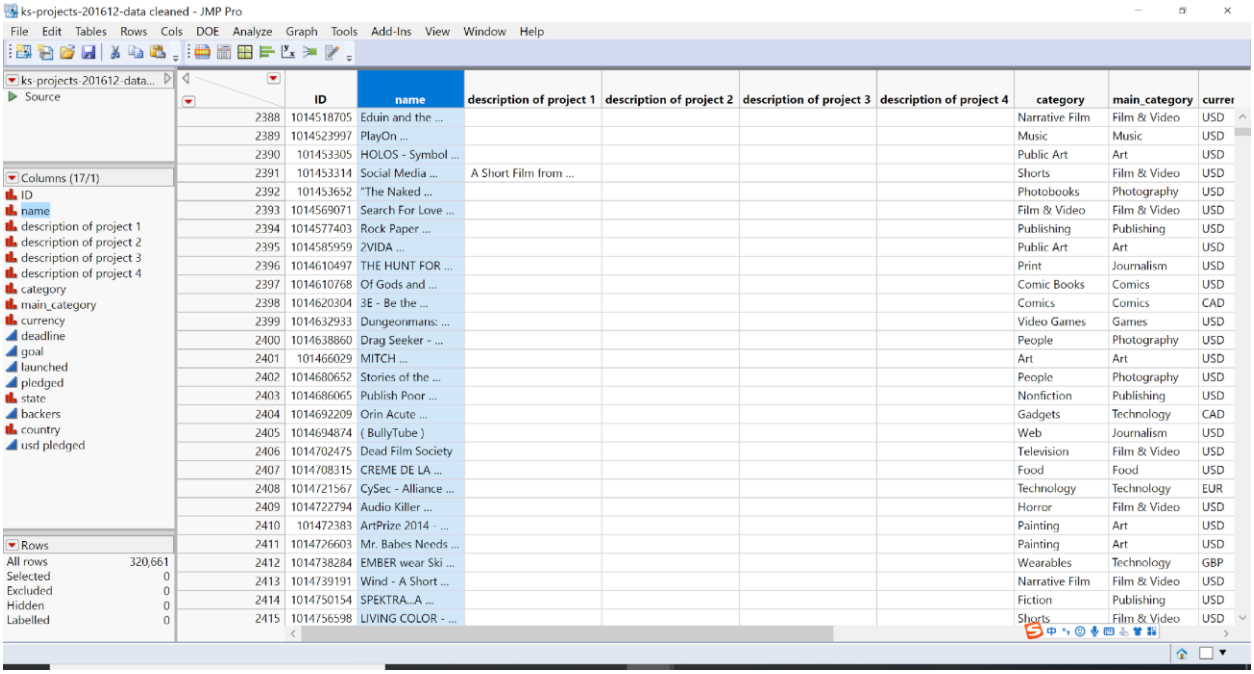


Step 2

Delete the records with launch date having the year 1970

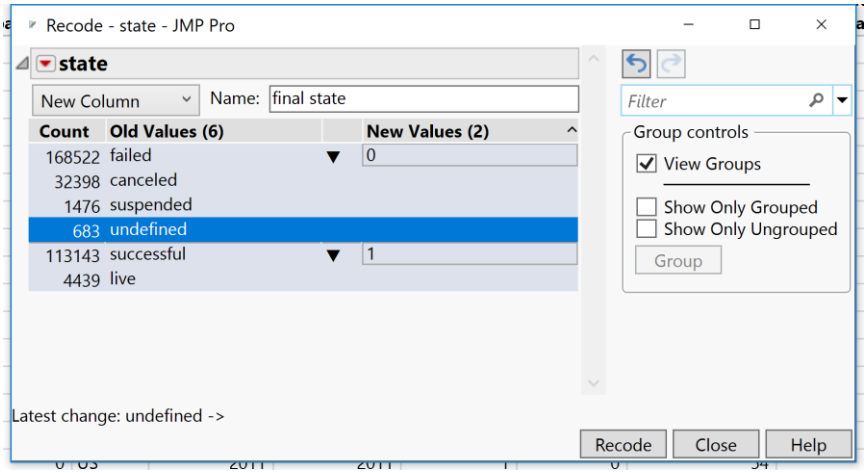



ks-projects-201612-data cleaned - JMP Pro
File Edit Tables Rows Cols DOE Analyze Graph Tools Add-Ins View Window Help
ks-projects-201612-data...
Source
Columns (17/1)
ID
name
description of project 1
description of project 2
description of project 3
description of project 4
category
main_category
currency
deadline
goal
launched
pledged
state
backers
country
usd pledged
Rows
All rows 320,668
Selected 7
Excluded 0
Hidden 0
Labelled 0
description of project 1 | description of project 2 | description of project 3 | description of project 4 | category | main_category | currency | deadline
125 Illustration Art USD 10/22/2015 4:30 ...
126 Photobooks Photography USD 12/29/2015 2:56 ...
127 Drama Film & Video GBP 03/20/2015 1:00 ...
128 Art Art GBP 05/08/2013 1:06 ...
129 Architecture Design GBP 09/28/2014 7:47 ...
130 Apparel Fashion USD 12/22/2016 4:46 ...
131 Hardware Technology USD 06/28/2011 9:09 ...
132 Young Adult Publishing GBP 06/16/2016 1:05 ...
133 Hip-Hop Music GBP 02/22/2013 10:32...
134 Games Games USD 06/06/2012 7:42 ...
135 Mobile Games Games GBP 08/19/2014 12:52...
136 Flight Technology USD 10/18/2014 7:40 ...
137 Theater Theater USD 06/18/2012 2:33 ...
138 Hardware Technology USD 07/08/2014 5:59 ...
139 Fine Art Photography GBP 08/29/2014 6:14 ...
140 Film & Video Film & Video GBP 12/22/2012 3:59 ...
141 Action Film & Video EUR 05/01/2016 7:09 ...
142 Art Books Publishing USD 04/07/2013 10:16...
143 Video Games Games USD 07/31/2014 4:01 ...
144 Playing Cards Games USD 03/22/2016 12:36...
145 Fashion Fashion USD 06/07/2014 1:00 ...
146 Makerspaces Technology AUD 04/01/2016 12:15...
147 Punk Music USD 07/05/2016 1:46 ...
148 Documentary Film & Video USD 12/09/2012 4:57 ...
149 Fiction Publishing GBP 06/17/2015 9:52 ...
150 Painting Art USD 04/24/2016 9:25 ...
151 Hardware Technology CAD 10/23/2013 11:36...
152 Jazz Music USD 12/07/2014 6:02 ...


ks-projects-201612-data cleaned - JMP Pro
File Edit Tables Rows Cols DOE Analyze Graph Tools Add-Ins View Window Help
ks-projects-201612-data...
Source
Columns (17/1)
ID
name
description of project 1
description of project 2
description of project 3
description of project 4
category
main_category
currency
deadline
goal
launched
pledged
state
backers
country
usd pledged
Rows
All rows 320,661
Selected 0
Excluded 0
Hidden 0
Labelled 0
ID | name | description of project 1 | description of project 2 | description of project 3 | description of project 4 | category | main_category | currer
2388 1014518705 Eduin and the ... Narrative Film Film & Video USD
2389 1014523997 PlayOn ... Music Music USD
2390 101453305 HOLOS - Symbol ... Public Art Art USD
2391 101453314 Social Media ... A Short Film from ... Shorts Film & Video USD
2392 101453652 "The Naked ... Photobooks Photography USD
2393 1014569071 Search For Love ... Film & Video Film & Video USD
2394 1014577403 Rock Paper ... Publishing Publishing USD
2395 1014585959 2VIDA ... Public Art Art USD
2396 1014610497 THE HUNT FOR ... Print Journalism USD
2397 1014610768 Of Gods and ... Comic Books Comics USD
2398 1014620304 3E - Be the ... Comics Comics CAD
2399 1014632933 Dungeonmans: ... Video Games Games USD
2400 1014638860 Drag Seeker - ... People Photography USD
2401 101466029 MITCH ... Art Art USD
2402 1014680652 Stories of the ... People Photography USD
2403 1014686065 Publish Poor ... Nonfiction Publishing USD
2404 1014692209 Orin Acute ... Gadgets Technology CAD
2405 1014694874 (BullyTube) Web Journalism USD
2406 1014702475 Dead Film Society Television Film & Video USD
2407 1014708315 CREME DE LA ... Food Food USD
2408 1014721567 CySec - Alliance ... Technology Technology EUR
2409 1014722794 Audio Killer ... Horror Film & Video USD
2410 101472383 ArtPrize 2014 - ... Painting Art USD
2411 1014726603 Mr. Babes Needs ... Painting Art USD
2412 1014738284 EMBER wear Ski ... Wearables Technology GBP
2413 1014739191 Wind - A Short ... Narrative Film Film & Video USD
2414 1014750154 SPEKTRA...A ... Fiction Publishing USD
2415 1014756598 LIVING COLOR - ... Shorts Film & Video USD

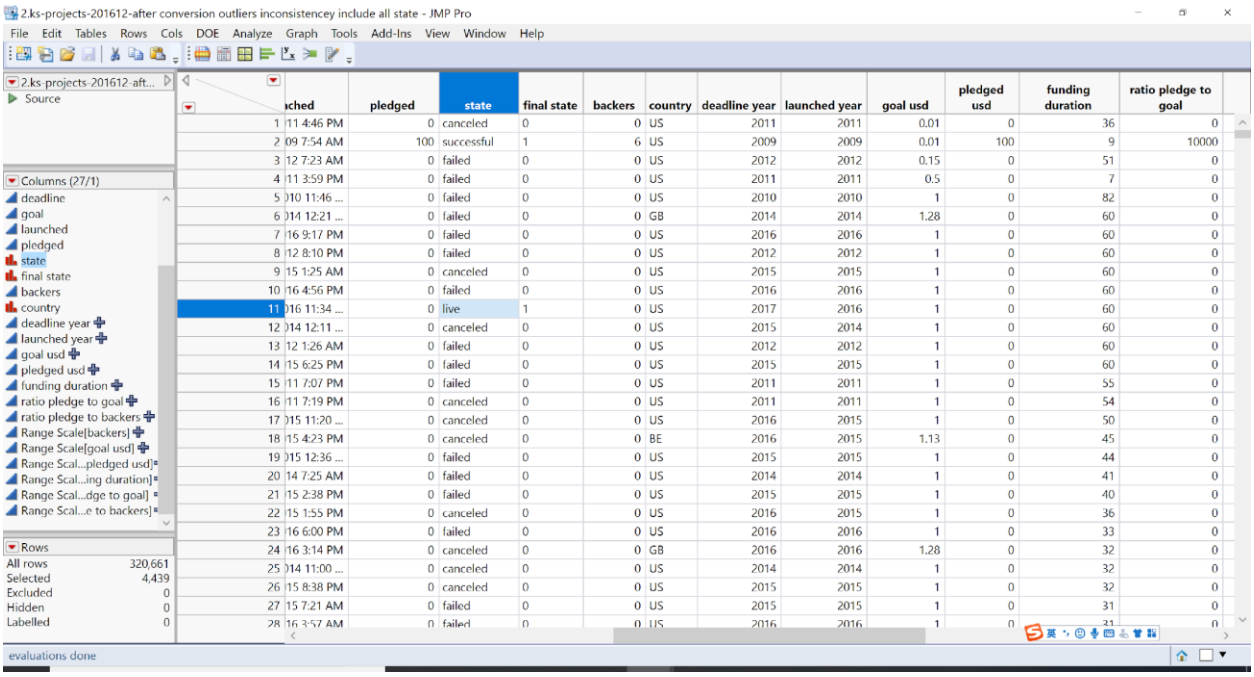
Step 3

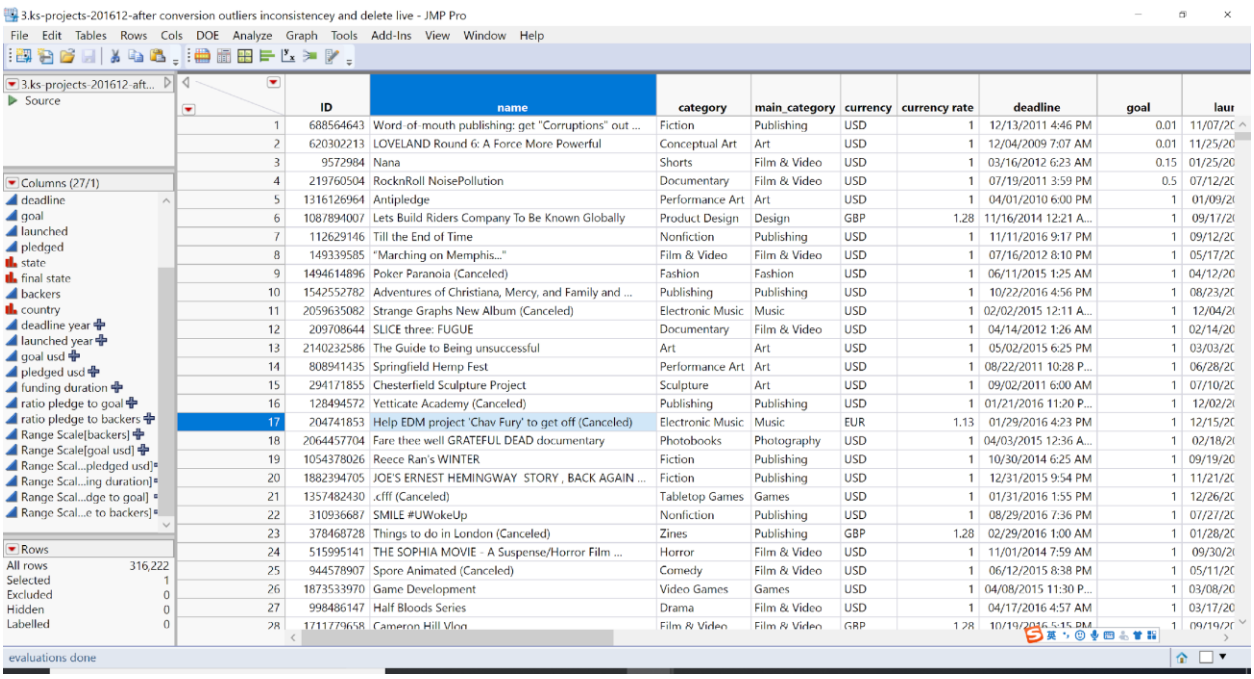
Change the state failed, cancelled, suspended, undefined to 0 and successful to 1



Step 4

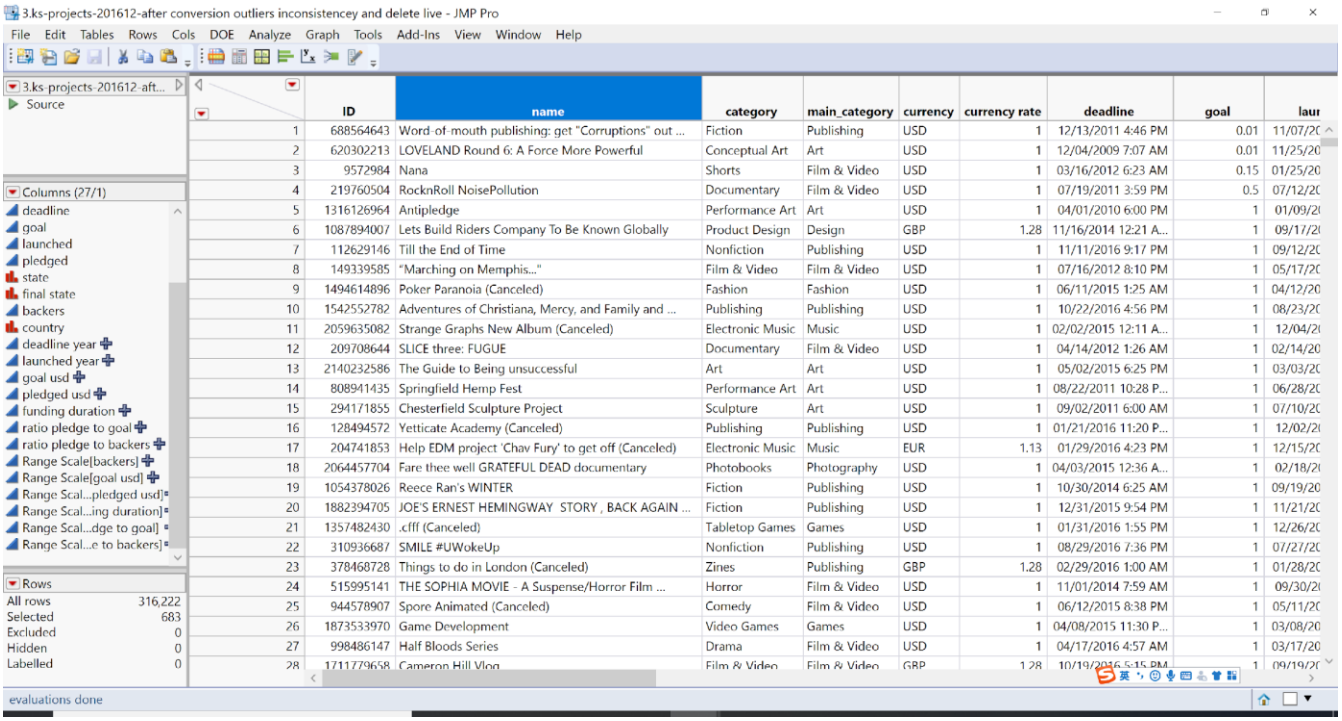
Delete the observations with status as live and name the data set as “Delete the live”

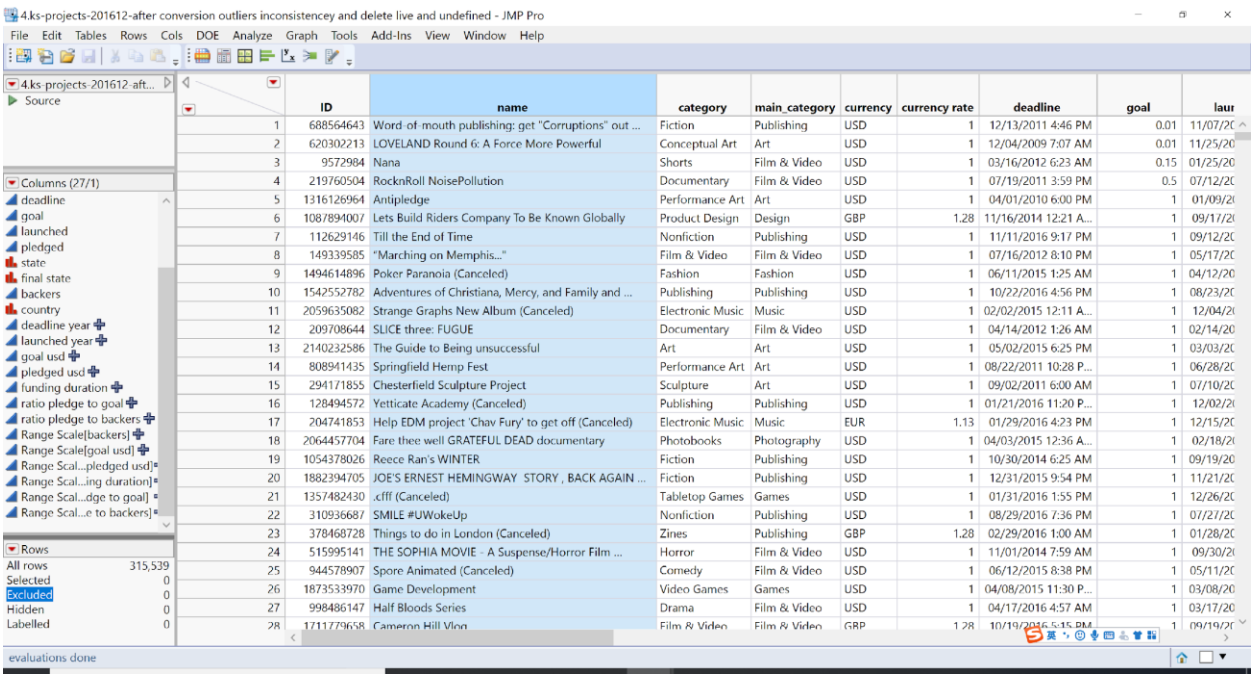




Step 5

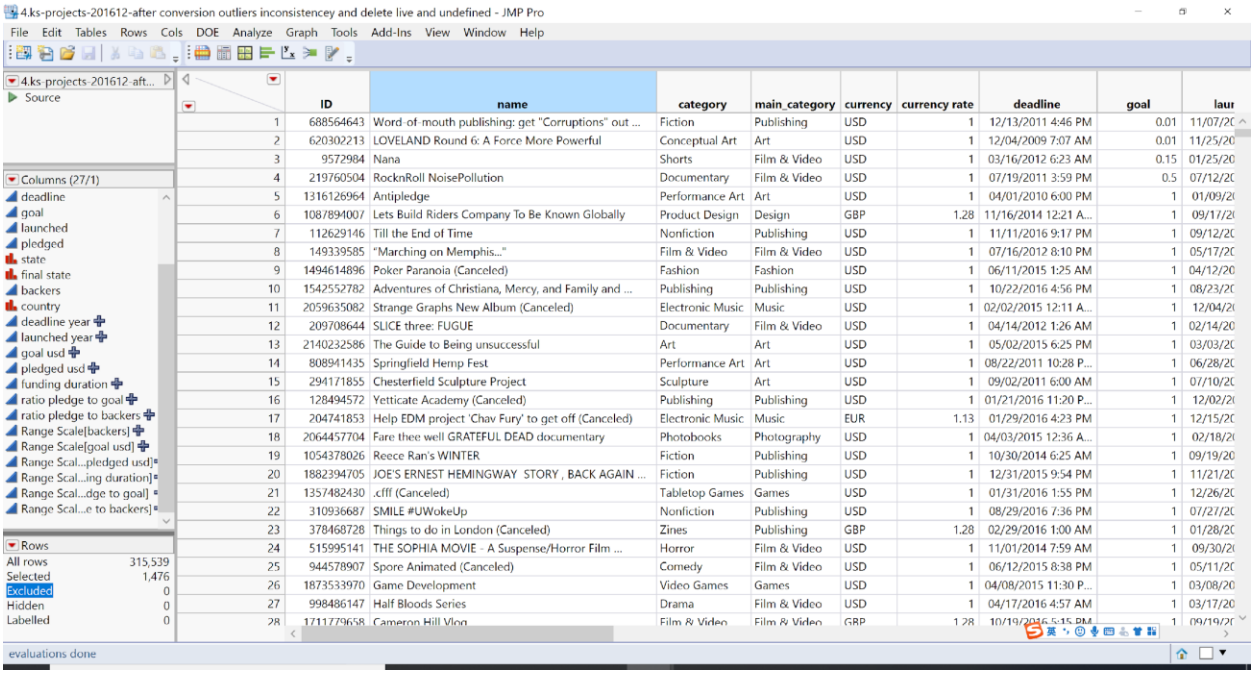
Delete the observations with status as undefined and use Decision tree modeling

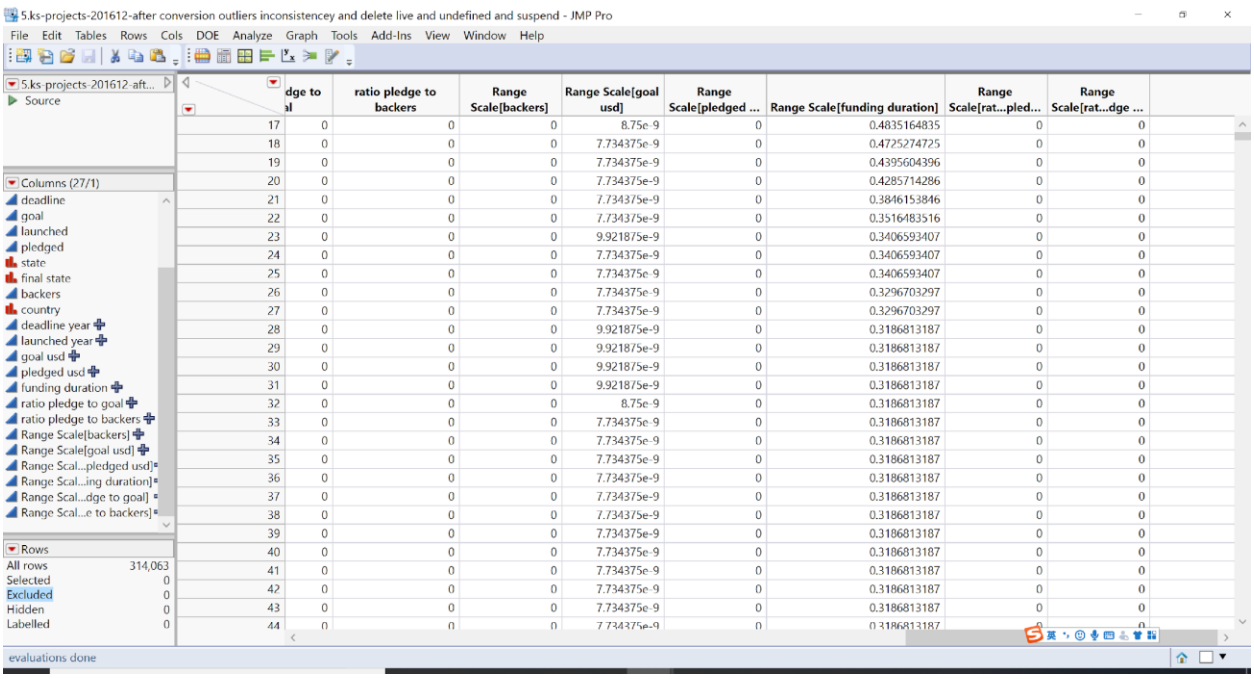




Step 6

Delete the observations with status as suspended



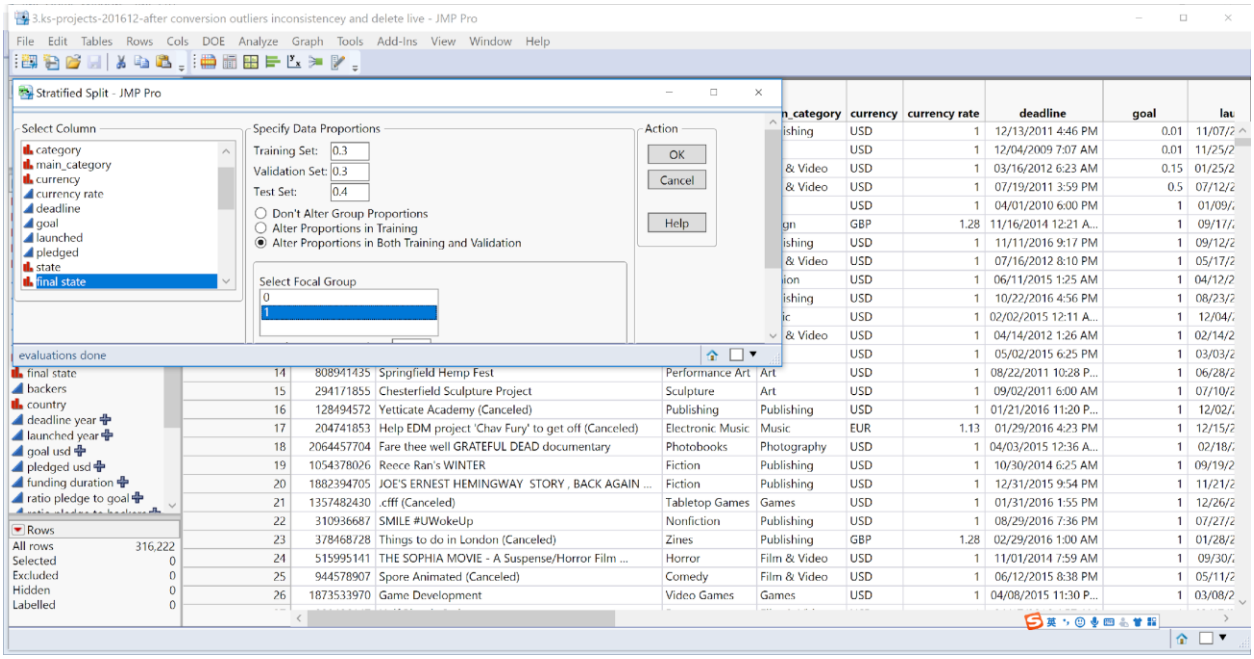


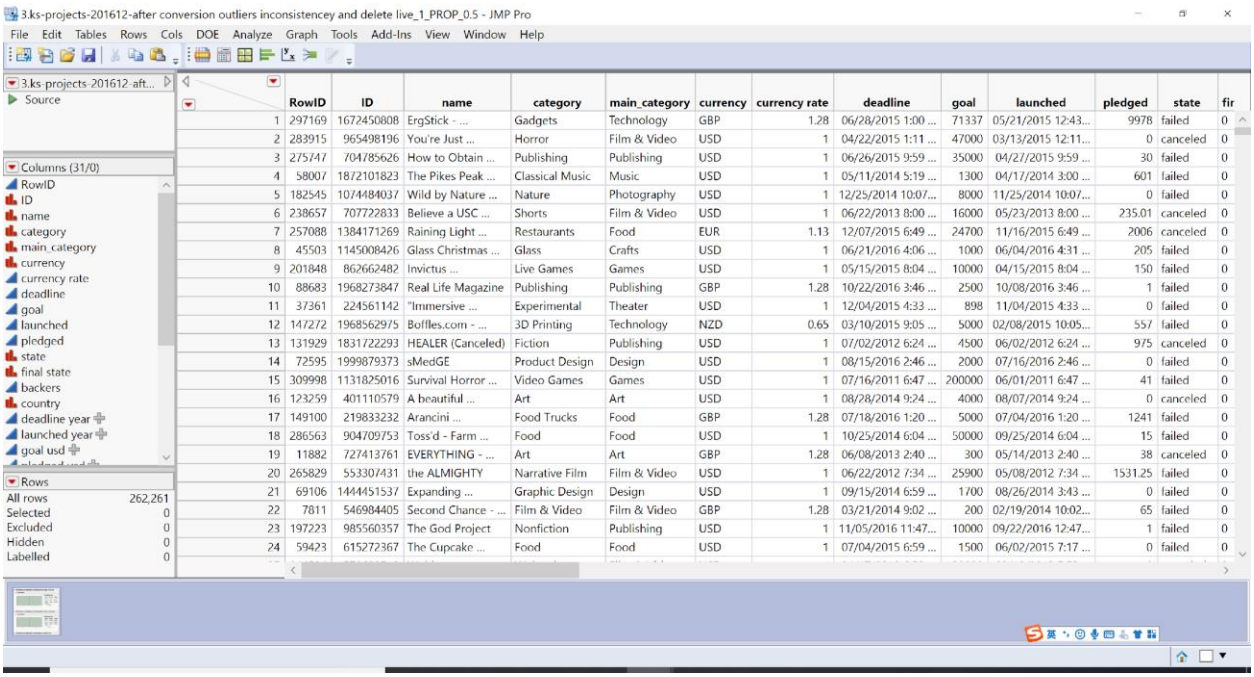
APPENDIX 1.3

This part includes the different situations we tried to find the best data set for modeling.

1. We used the data set “delete the live (state)” to build the model by using the decision tree model

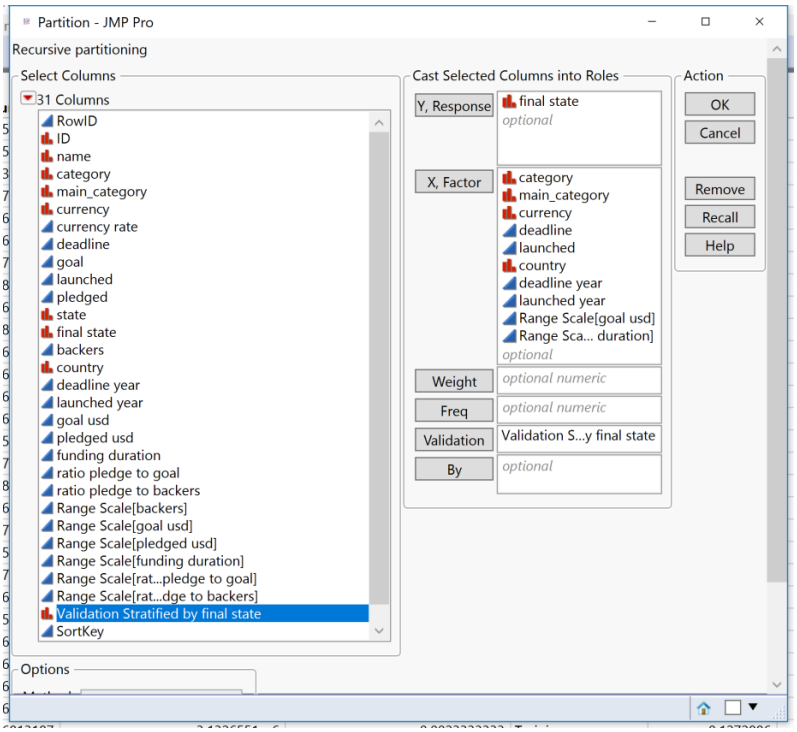
a. Use Stratified Split proportion: Training set- 0.3 Validation set- 0.3 Test set - 0.4





Sample size: 262,261

Input variables:

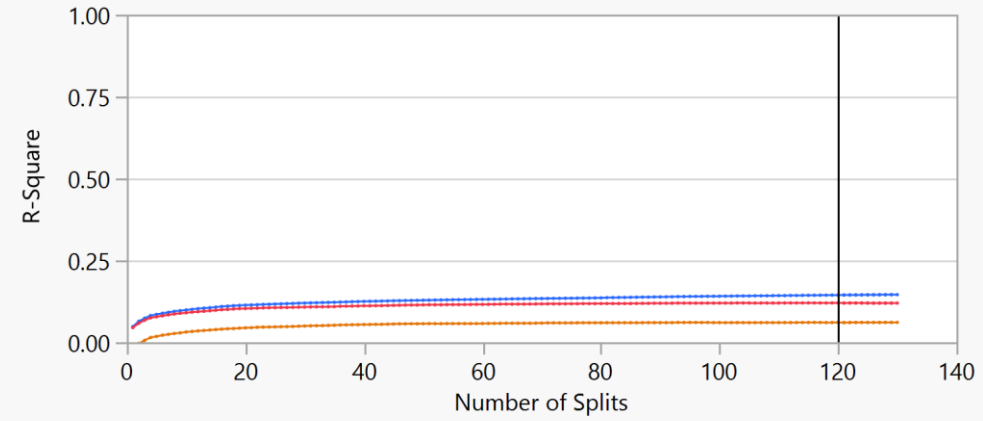


Results:

Split Prune Go Color Points

	RSquare	N	Number of Splits
Training	0.144	67886	120
Validation	0.120	67886	
Test	0.061	126489	

Split History



Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1443	0.1204	0.0607	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2418	0.2050	0.1044	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5931	0.6097	0.6126	∑ -Log(ρ[j])/n
RMSE	0.4522	0.4594	0.4616	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4090	0.4157	0.4166	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3219	0.3344	0.3485	∑ (ρ[j]≠ρMax)/n
N	67886	67886	126489	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	21702	12241
1	9611	24332

Validation

Actual	Predicted Count	
final state	0	1
0	21135	12808
1	9895	24048

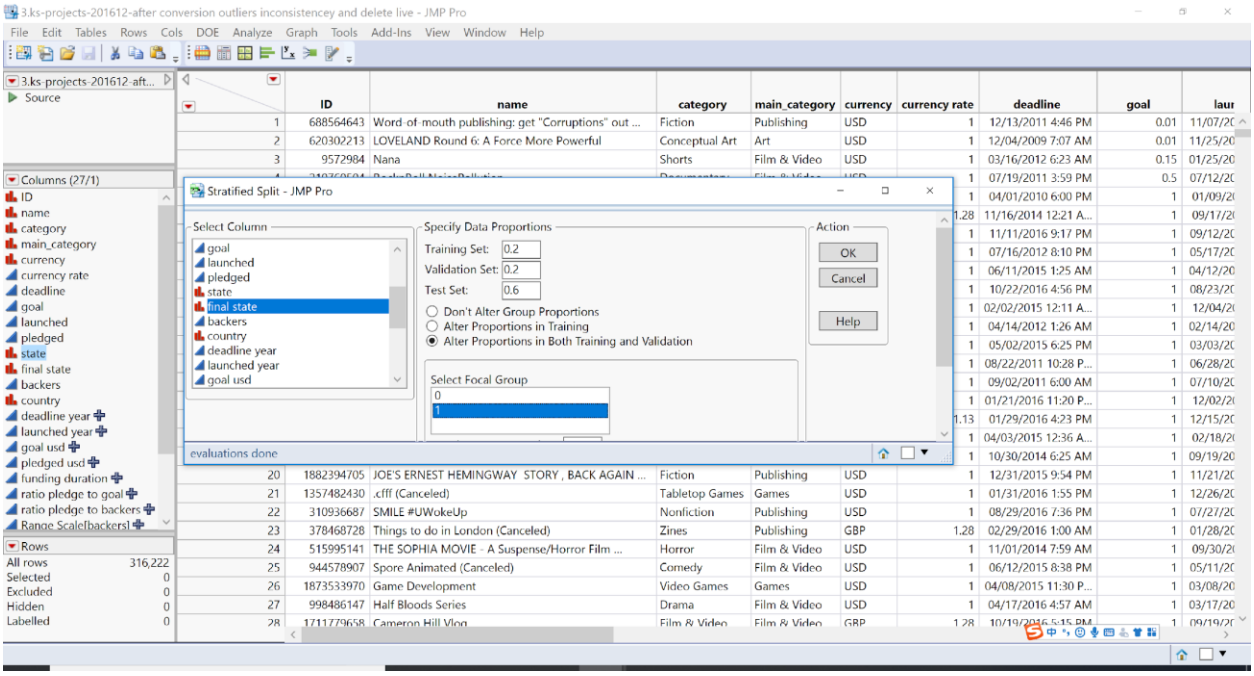
Test

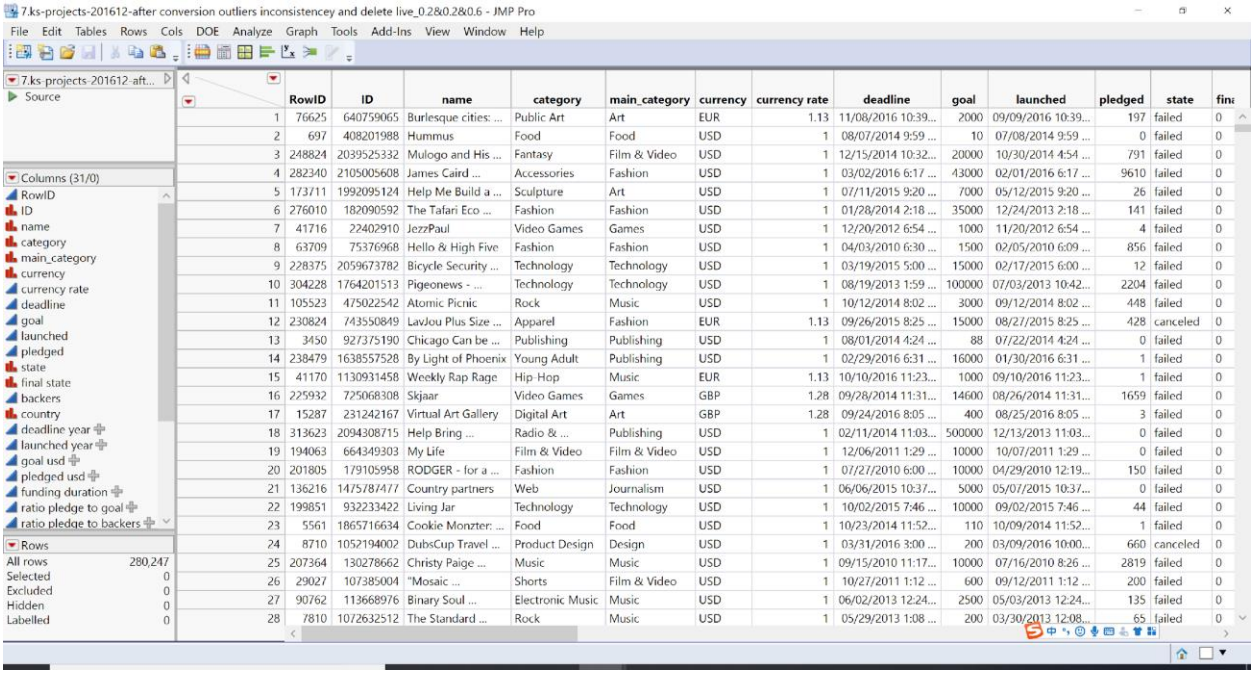
Actual	Predicted Count	
final state	0	1
0	50727	30505
1	13579	31678

We are getting good accuracy 67.8% for training data set and 65.15% for test data set.

Data set	Number of False positive and False Negative Errors	Accuracy
Training	21852	67.81%
Validation	22703	66.56%
Test	44084	65.15%

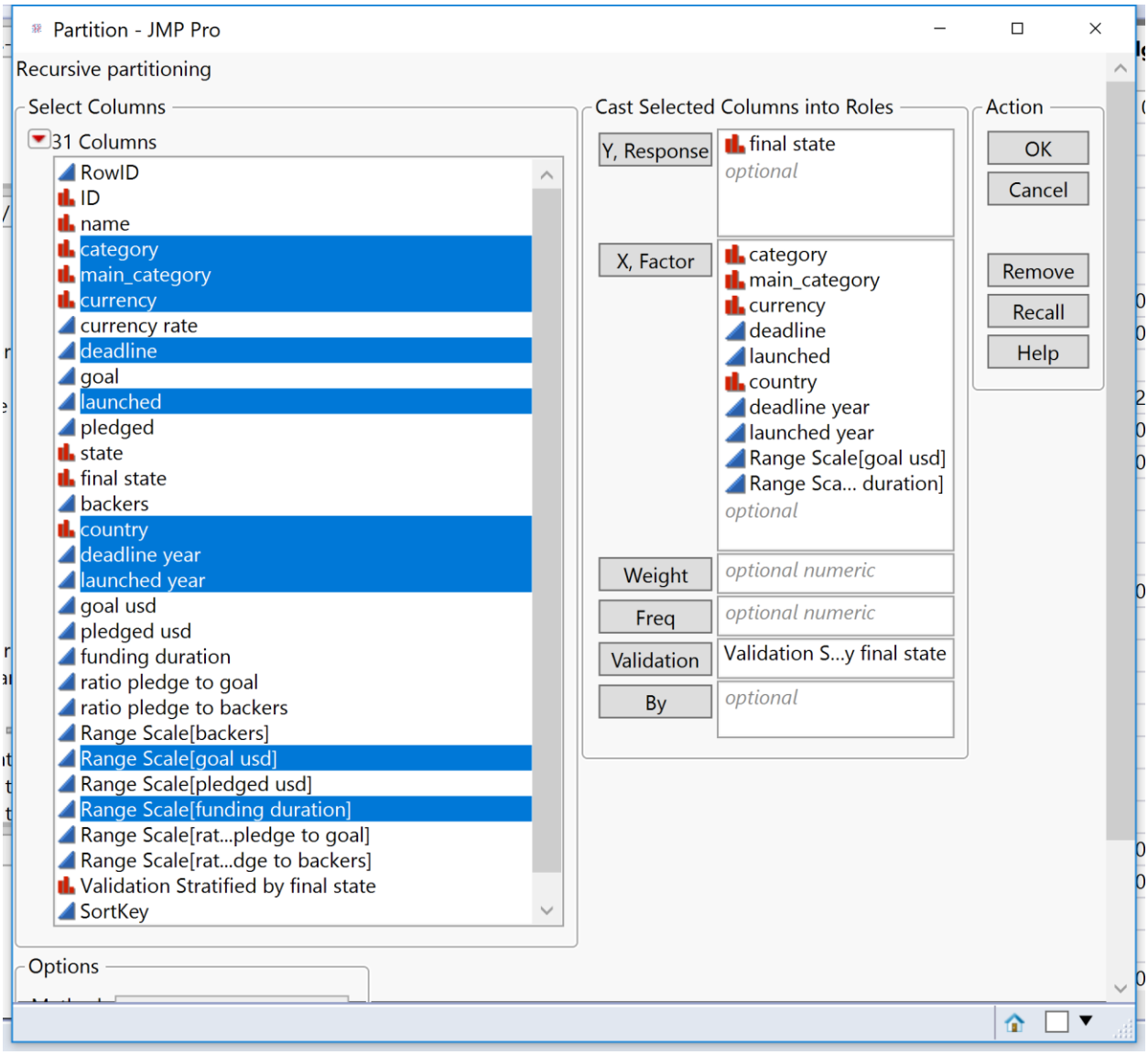
b. Use Stratified Split Proportion: Training set - 0.2 Validation set- 0.2 Test set - 0.6





Sample size: 280,247

Input variables:

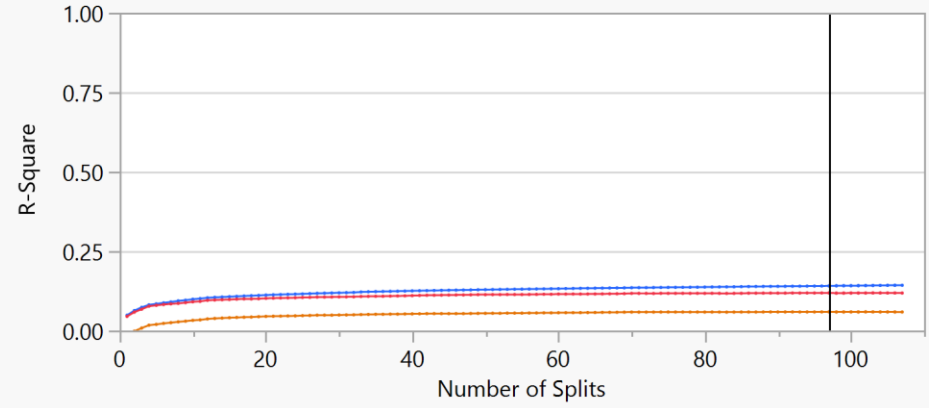

Partition - JMP Pro
Recursive partitioning
Select Columns
31 Columns
RowID
ID
name
category
main_category
currency
currency rate
deadline
goal
launched
pledged
state
final state
backers
country
deadline year
launched year
goal usd
pledged usd
funding duration
ratio pledge to goal
ratio pledge to backers
Range Scale[backers]
Range Scale[goal usd]
Range Scale[pledged usd]
Range Scale[funding duration]
Range Scale[rat...pledge to goal]
Range Scale[rat...dge to backers]
Validation Stratified by final state
SortKey
Cast Selected Columns into Roles
Y, Response
final state
optional
X, Factor
category
main_category
currency
deadline
launched
country
deadline year
launched year
Range Scale[goal usd]
Range Sca... duration]
optional
Weight
optional numeric
Freq
optional numeric
Validation
Validation S...y final state
By
optional
Action
OK
Cancel
Remove
Recall
Help
Options

Results:

Split | Prune | Go

	RSquare	N	Number of Splits
Training	0.141	45256	97
Validation	0.119	45258	
Test	0.059	189733	

Split History



Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1410	0.1179	0.0586	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2367	0.2010	0.1010	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5954	0.6114	0.6139	∑ -Log(ρ[j])/n
RMSE	0.4531	0.4603	0.4624	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4108	0.4171	0.4179	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3251	0.3387	0.3487	∑ (ρ[j]≠ρMax)/n
N	45256	45258	189733	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	14649	7979
1	6735	15893

Validation

Actual	Predicted Count	
final state	0	1
0	14194	8435
1	6894	15735

Test

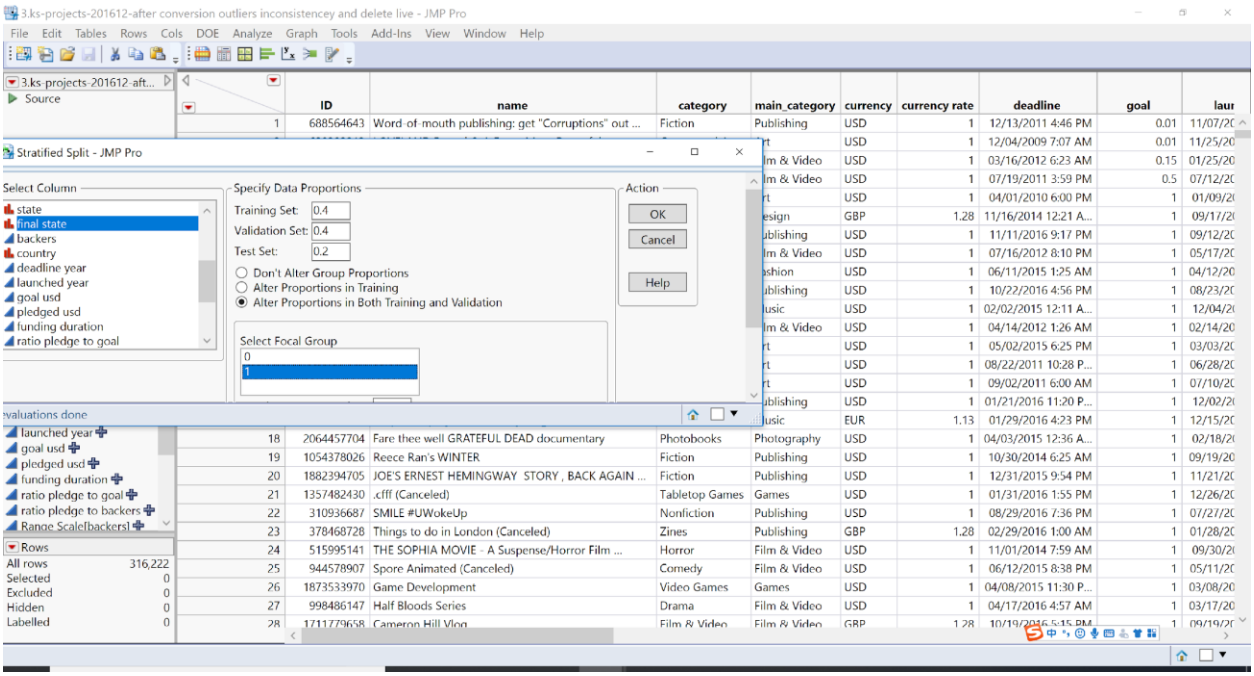
Actual	Predicted Count	
final state	0	1
0	76728	45119
1	21032	46854

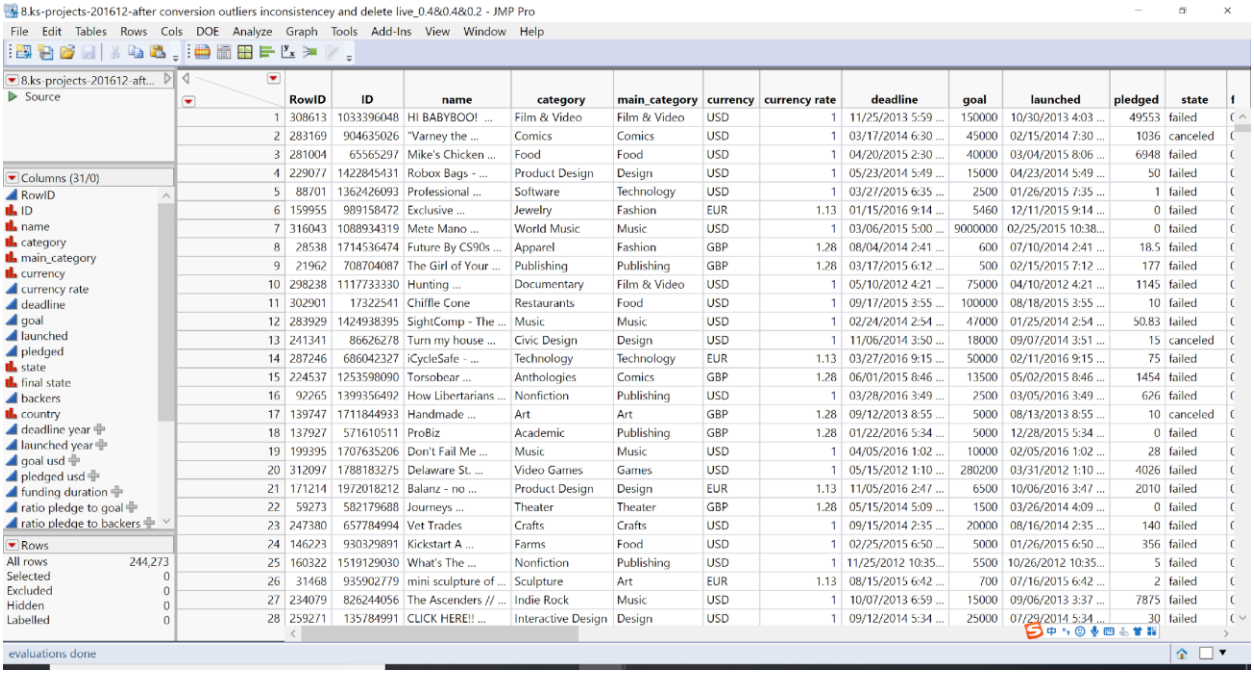
We are getting good accuracy 67.5% for training data set and 65.13% for test data set. This split provides less accuracy (65.13% < 65.15%) for test data set comparing the split “Train 0.3 Validation 0.3 Test 0.4”. Therefore, we keep the split “Train 0.3 Validation 0.3 Test 0.4”.

Data set	Number of False positive and False Negative Errors	Accuracy
Training	14714	67.49%

Validation	15329	66.13%
Test	66151	65.13%

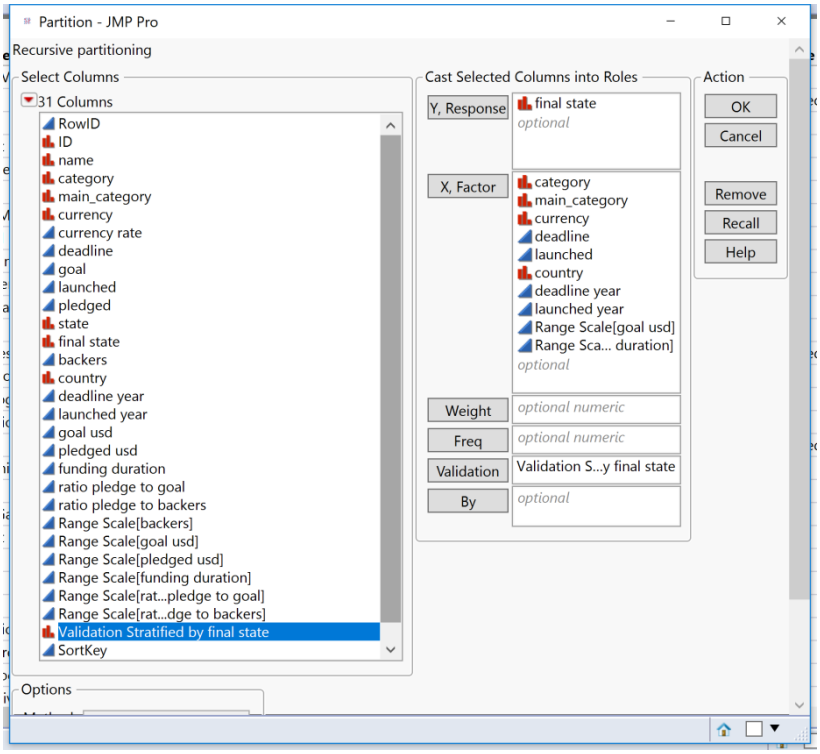
c. Use Stratified Split Proportion: Training set- 0.4 Validation set - 0.4 Testing set - 0.2



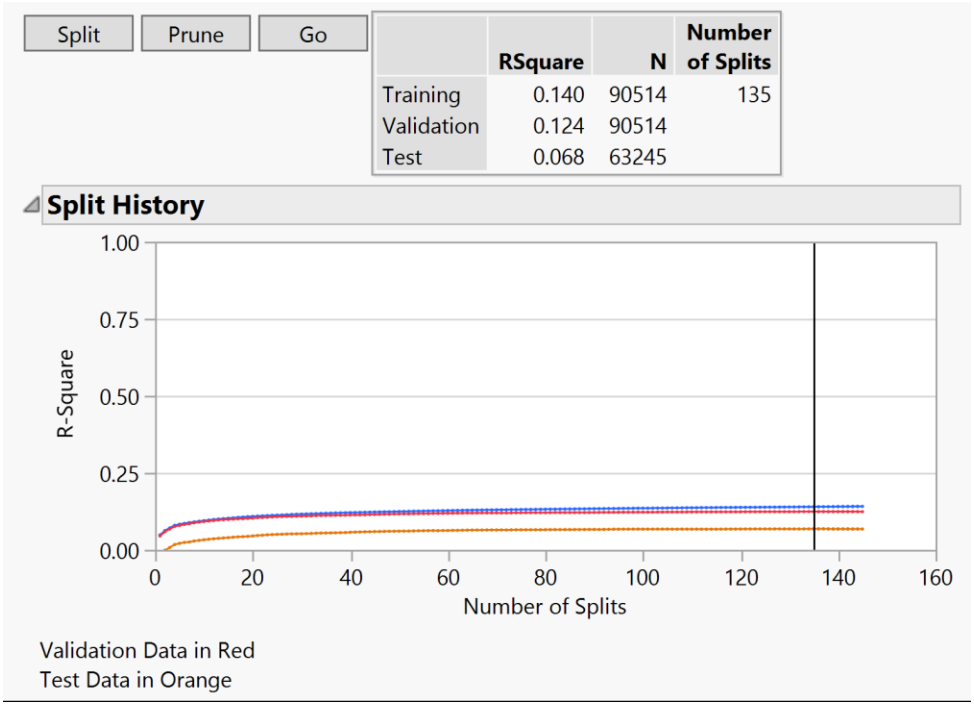


Sample size: 244,273

Input variables:


Partition - JMP Pro
Recursive partitioning
Select Columns
31 Columns
RowID
ID
name
category
main_category
currency
currency rate
deadline
goal
launched
pledged
state
final state
backers
country
deadline year
launched year
goal usd
pledged usd
funding duration
ratio pledge to goal
ratio pledge to backers
Range Scale[backers]
Range Scale[goal usd]
Range Scale[pledged usd]
Range Scale[funding duration]
Range Scale[rat...pledge to goal]
Range Scale[rat...dge to backers]
Validation Stratified by final state
SortKey
Cast Selected Columns into Roles
Y, Response
final state
optional
X, Factor
category
main_category
currency
deadline
launched
country
deadline year
launched year
Range Scale[goal usd]
Range Sca... duration]
optional
Weight
optional numeric
Freq
optional numeric
Validation
Validation S...y final state
By
optional
Action
OK
Cancel
Remove
Recall
Help
Options

Results:


Split
Prune
Go
RSquare
N
Number of Splits
Training 0.140 90514 135
Validation 0.124 90514
Test 0.068 63245
Split History
R-Square
1.00
0.75
0.50
0.25
0.00
0
20
40
60
80
100
120
140
160
Number of Splits
Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1399	0.1243	0.0682	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2350	0.2110	0.1169	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5962	0.6070	0.6076	∑ -Log(ρ[j])/n
RMSE	0.4536	0.4583	0.4595	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4116	0.4160	0.4158	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3254	0.3340	0.3417	∑ (ρ[j]≠ρMax)/n
N	90514	90514	63245	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	29186	16071
1	13386	31871

Validation

Actual	Predicted Count	
final state	0	1
0	28736	16521
1	13708	31549

Test

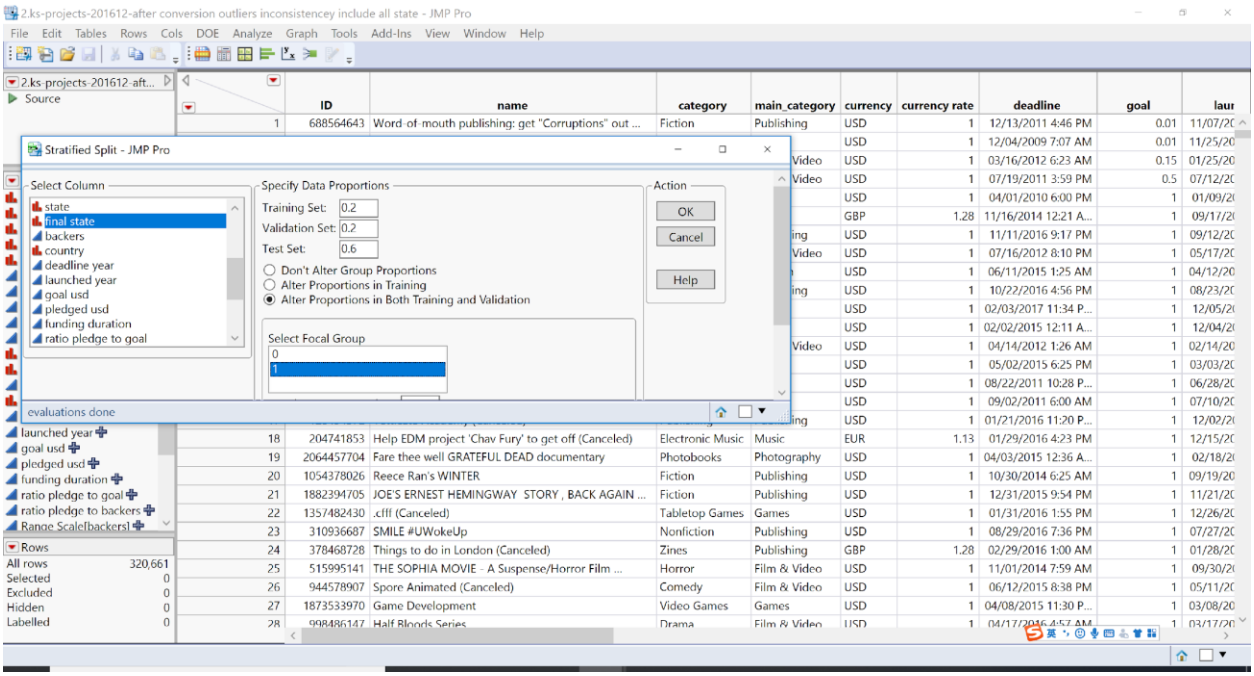
Actual	Predicted Count	
final state	0	1
0	25923	14693
1	6915	15714

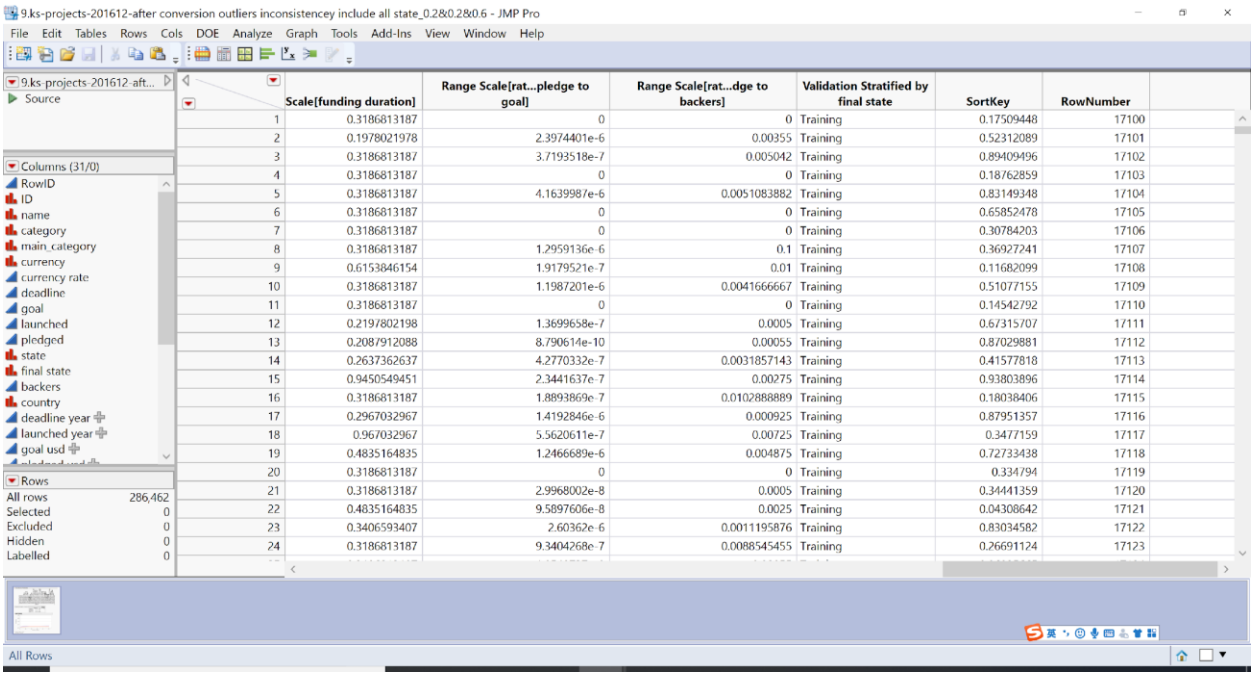
We are getting good accuracy 67.5% for training data set and 65.83% for test data set. This split provides more accuracy (65.83% > 65.15%) for test data set comparing the split "Train 0.3 Validation 0.3 Test 0.4". Therefore, we keep the split "Train 0.4 Validation 0.4 Test 0.2" to build the model.

Data set	Number of False positive and False Negative Errors	Accuracy
Training	29457	67.46%
Validation	30229	66.60%
Test	21608	65.83%

2. We used the data set “all (state)” and “undefined is unsuccessful” to build the model by using the decision tree model

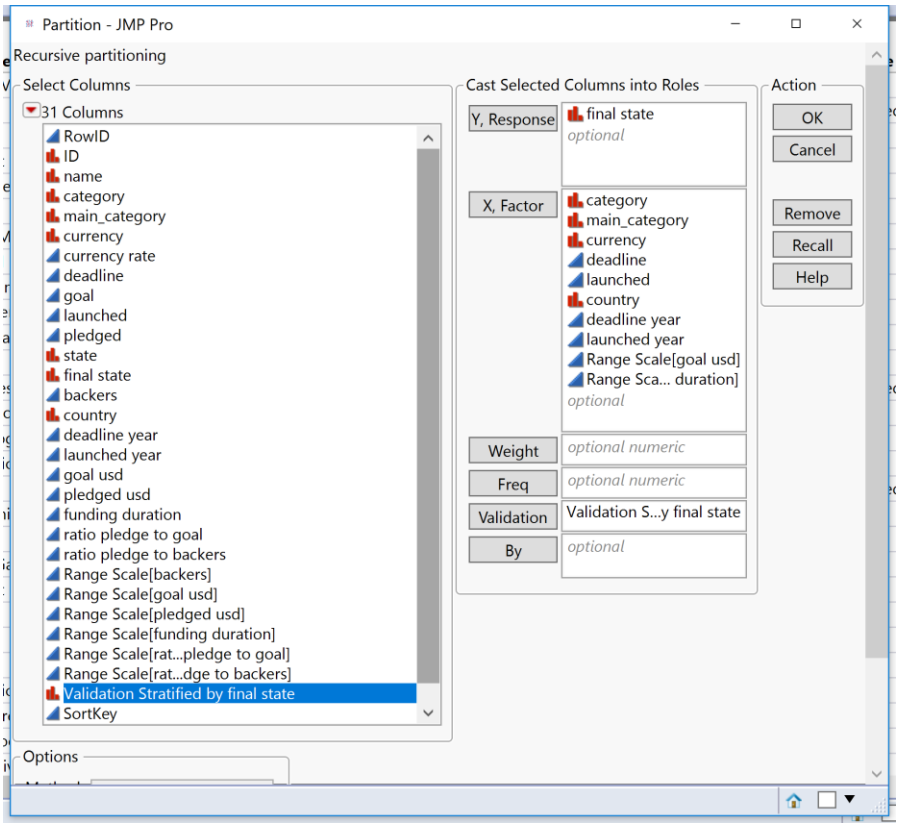
a. Use Stratified split proportion: Training set - 0.2 Validation set - 0.2 Test set - 0.6



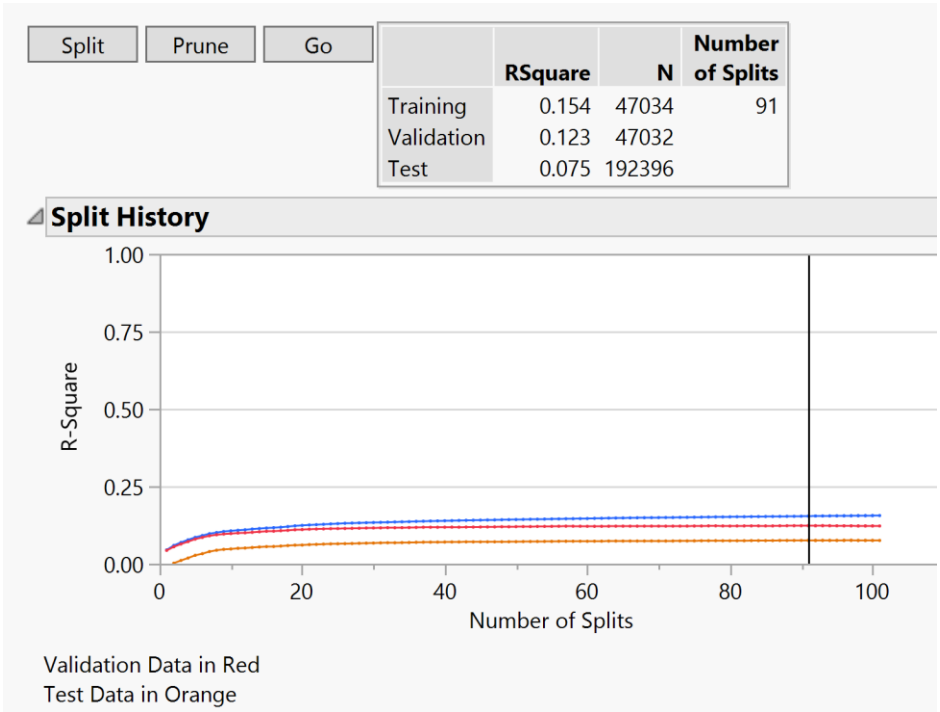


Sample size: 286,462

Input variables:



Results:



Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1536	0.1228	0.0754	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2557	0.2087	0.1290	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5867	0.6081	0.6076	∑ -Log(ρ[j])/n
RMSE	0.4493	0.4587	0.4594	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4039	0.4121	0.4121	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3178	0.3363	0.3417	∑ (ρ[j]≠ρMax)/n
N	47034	47032	192396	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	15442	8075
1	6873	16644

Validation

Actual	Predicted Count	
final state	0	1
0	15114	8402
1	7416	16100

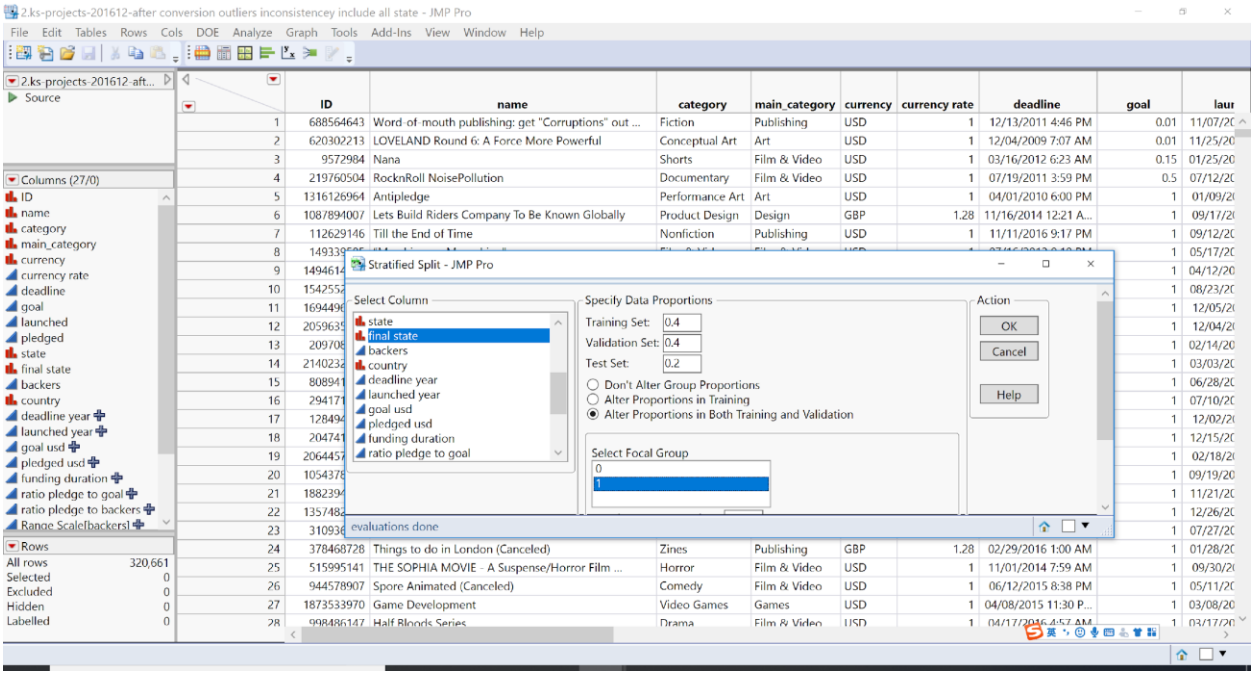
Test

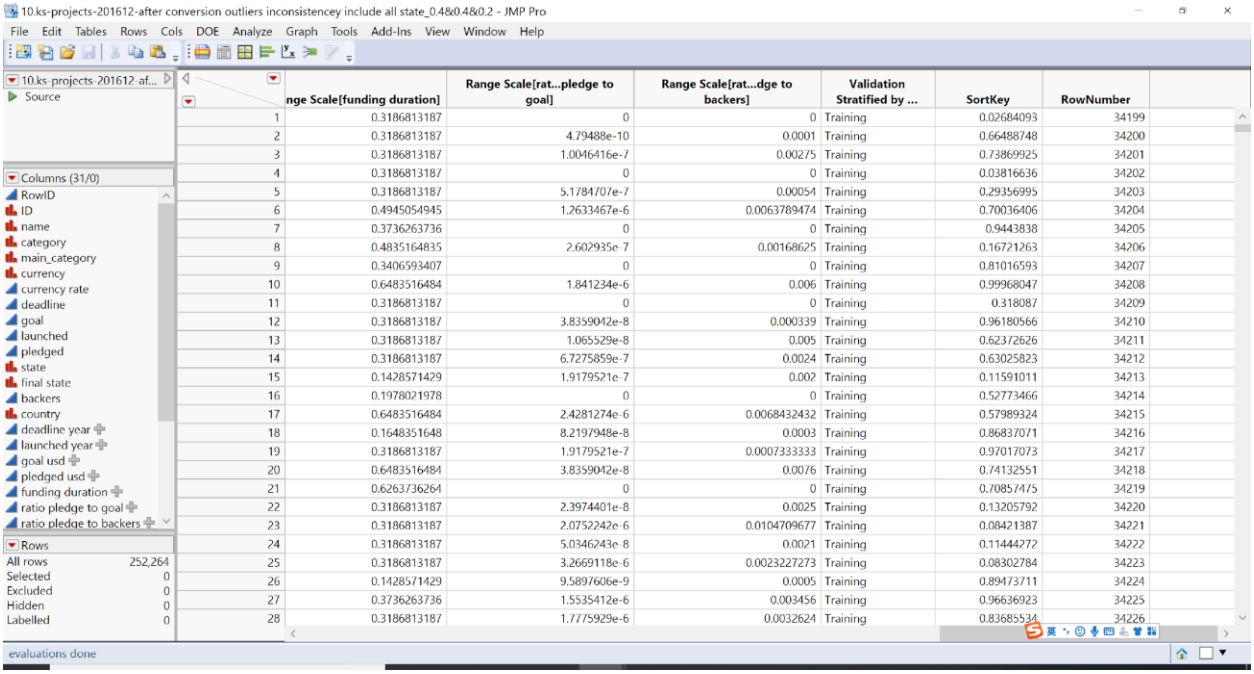
Actual	Predicted Count	
final state	0	1
0	77860	43987
1	21746	48803

We are getting good accuracy 68.2% for training data set and 65.83% for test data set. This split provides same accuracy (65.83% = 65.83%) for test data set comparing the split “Train 0.4 Validation 0.4 Test 0.2”. Therefore, we keep the split “Train 0.4 Validation 0.4 Test 0.2” of “delete the live (state)” to build the model.

Data set	**Number of False positive and False Negative Errors**	**Accuracy**
Training	14948	68.22%
Validation	15818	66.37%
Test	65733	65.83%

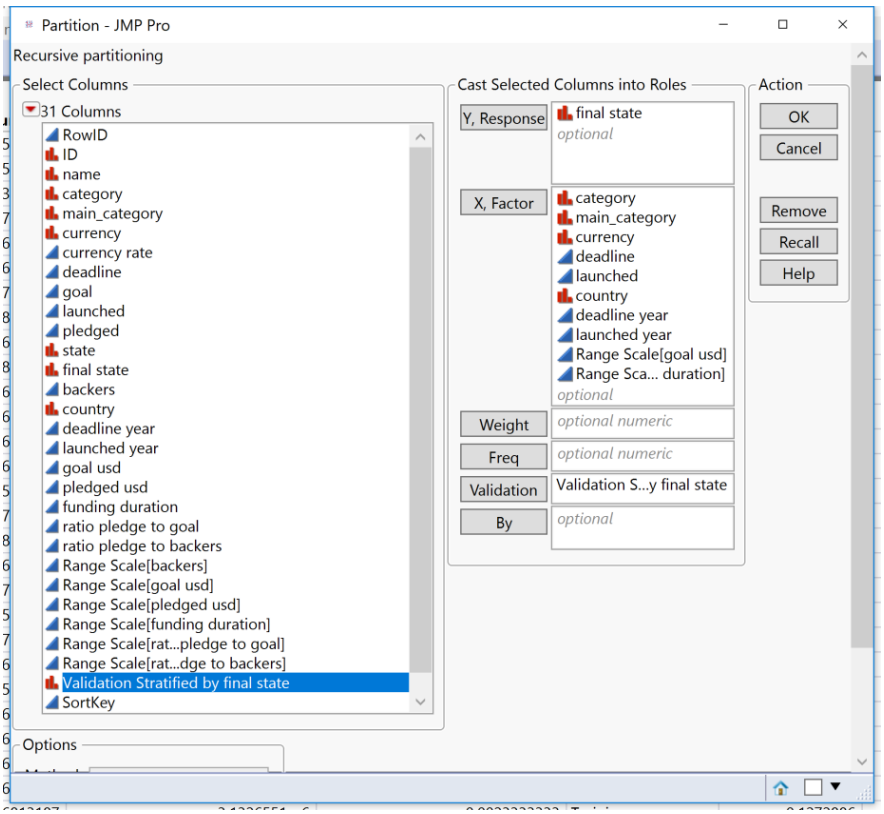
b. Use Stratified split proportion: Training set - 0.4 Validation set - 0.4 Testing set - 0.2



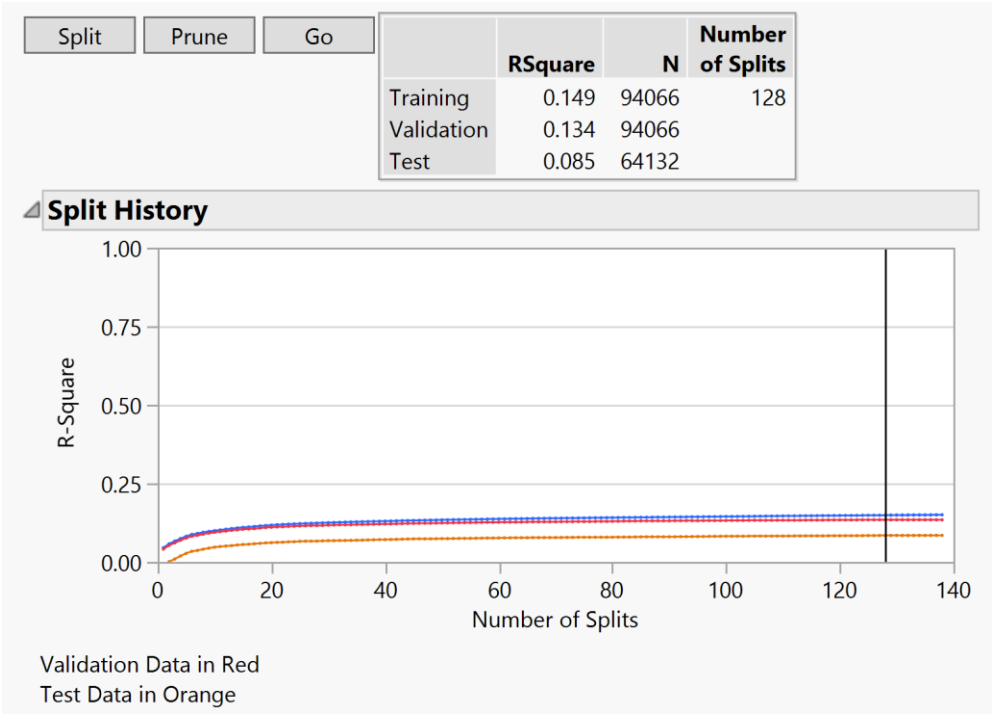


Sample size: 252,264

Input variables:


Partition - JMP Pro
Recursive partitioning
Select Columns
31 Columns
RowID
ID
name
category
main_category
currency
currency rate
deadline
goal
launched
pledged
state
final state
backers
country
deadline year
launched year
goal usd
pledged usd
funding duration
ratio pledge to goal
ratio pledge to backers
Range Scale[backers]
Range Scale[goal usd]
Range Scale[pledged usd]
Range Scale[funding duration]
Range Scale[rat...pledge to goal]
Range Scale[rat...dge to backers]
Validation Stratified by final state
SortKey
Cast Selected Columns into Roles
Y, Response
final state
optional
X, Factor
category
main_category
currency
deadline
launched
country
deadline year
launched year
Range Scale[goal usd]
Range Sca... duration]
optional
Weight
optional numeric
Freq
optional numeric
Validation
Validation S...y final state
By
optional
Action
OK
Cancel
Remove
Recall
Help
Options

Results:


Split
Prune
Go
RSquare
N
Number of Splits
Training 0.149 94066 128
Validation 0.134 94066
Test 0.085 64132
Split History
R-Square
1.00
0.75
0.50
0.25
0.00
0
20
40
60
80
100
120
140
Number of Splits
Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1489	0.1344	0.0846	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2487	0.2266	0.1439	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5899	0.6000	0.6016	∑ -Log(ρ[j])/n
RMSE	0.4508	0.4554	0.4567	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4065	0.4110	0.4111	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3183	0.3280	0.3353	∑ (ρ[j]≠ρMax)/n
N	94066	94066	64132	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	30597	16436
1	13502	33531

Validation

Actual	Predicted Count	
final state	0	1
0	30266	16767
1	14085	32948

Test

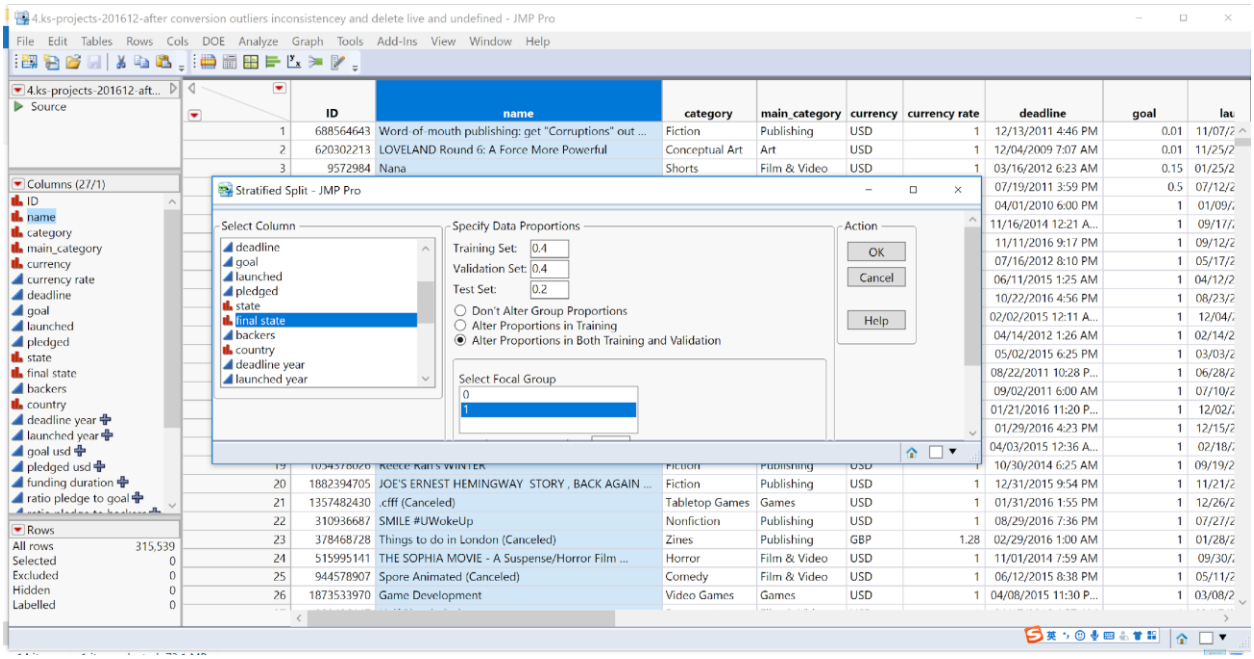
Actual	Predicted Count	
final state	0	1
0	26161	14455
1	7051	16465

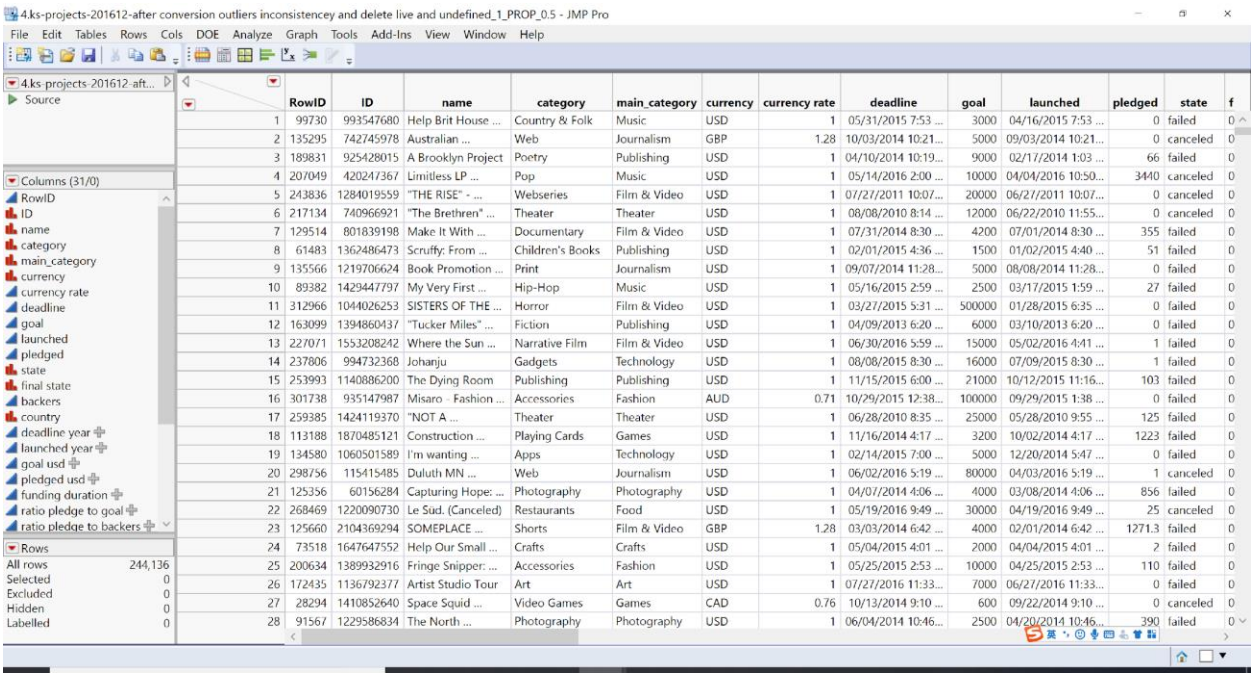
We are getting good accuracy 68.2% for training data set and 66.47% for test data set. This split provides more accuracy (66.47% > 65.83%) for test data set comparing the split “Train 0.4 Validation 0.4 Test 0.2”. But the difference is small, it’s more reasonable to use the data set which delete the live (state). Therefore, we keep the split “Train 0.4 Validation 0.4 Test 0.2” of “delete the live (state)” to build the model.

Data set	**Number of False positive and False Negative Errors**	**Accuracy**
Training	29938	68.17%
Validation	30852	67.20%
Test	21506	66.47%

3. We used the data set “delete live and undefined (state)” to build the model by using the decision tree model

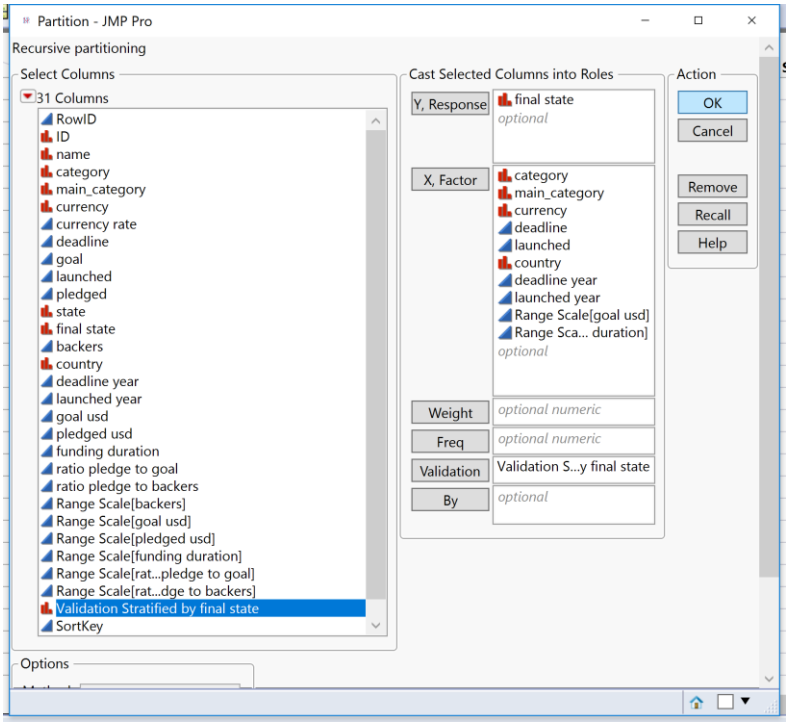
a. Use Stratified split proportion: Training set - 0.4 Validation set - 0.4 Testing set - 0.2



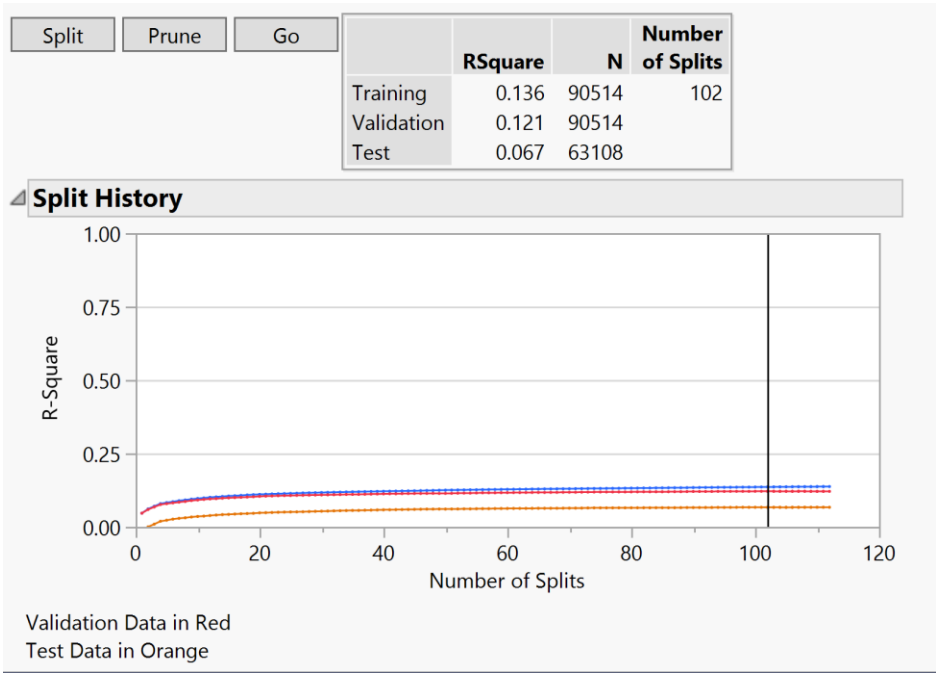


Sample size: 244,136

Input variables:


Partition - JMP Pro
Recursive partitioning
Select Columns
31 Columns
RowID
ID
name
category
main_category
currency
currency rate
deadline
goal
launched
pledged
state
final state
backers
country
deadline year
launched year
goal usd
pledged usd
funding duration
ratio pledge to goal
ratio pledge to backers
Range Scale[backers]
Range Scale[goal usd]
Range Scale[pledged usd]
Range Scale[funding duration]
Range Scale[rat...pledge to goal]
Range Scale[rat...dge to backers]
Validation Stratified by final state
SortKey
Cast Selected Columns into Roles
Y, Response
final state
optional
X, Factor
category
main_category
currency
deadline
launched
country
deadline year
launched year
Range Scale[goal usd]
Range Sca... duration]
optional
Weight
optional numeric
Freq
optional numeric
Validation
Validation S...y final state
By
optional
Action
OK
Cancel
Remove
Recall
Help
Options

Results:


Split
Prune
Go
RSquare
N
Number of Splits
Training 0.136 90514 102
Validation 0.121 90514
Test 0.067 63108
Split History
R-Square
1.00
0.75
0.50
0.25
0.00
0
20
40
60
80
100
120
Number of Splits
Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1359	0.1213	0.0665	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2289	0.2063	0.1141	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5990	0.6091	0.6092	∑ -Log(ρ[j])/n
RMSE	0.4547	0.4595	0.4601	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4135	0.4181	0.4175	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3262	0.3359	0.3383	∑ (ρ[j]≠ρMax)/n
N	90514	90514	63108	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	29689	15568
1	13962	31295

Validation

Actual	Predicted Count	
final state	0	1
0	29370	15887
1	14513	30744

Test

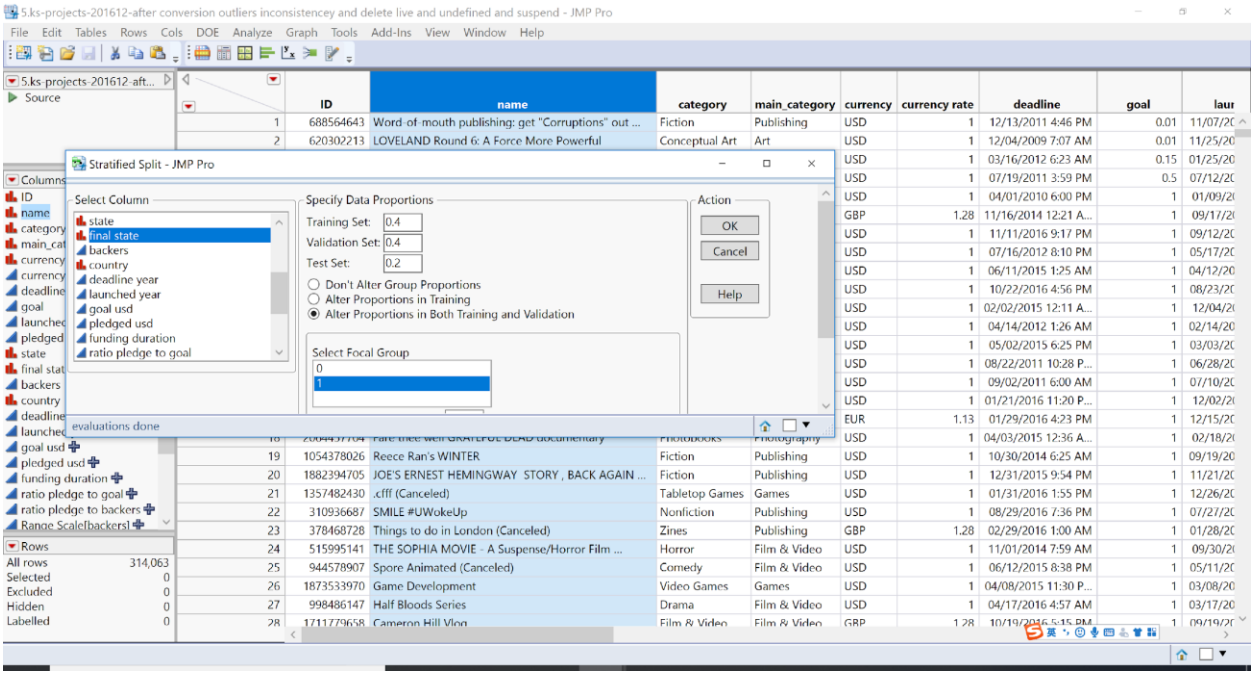
Actual	Predicted Count	
final state	0	1
0	26412	14067
1	7284	15345

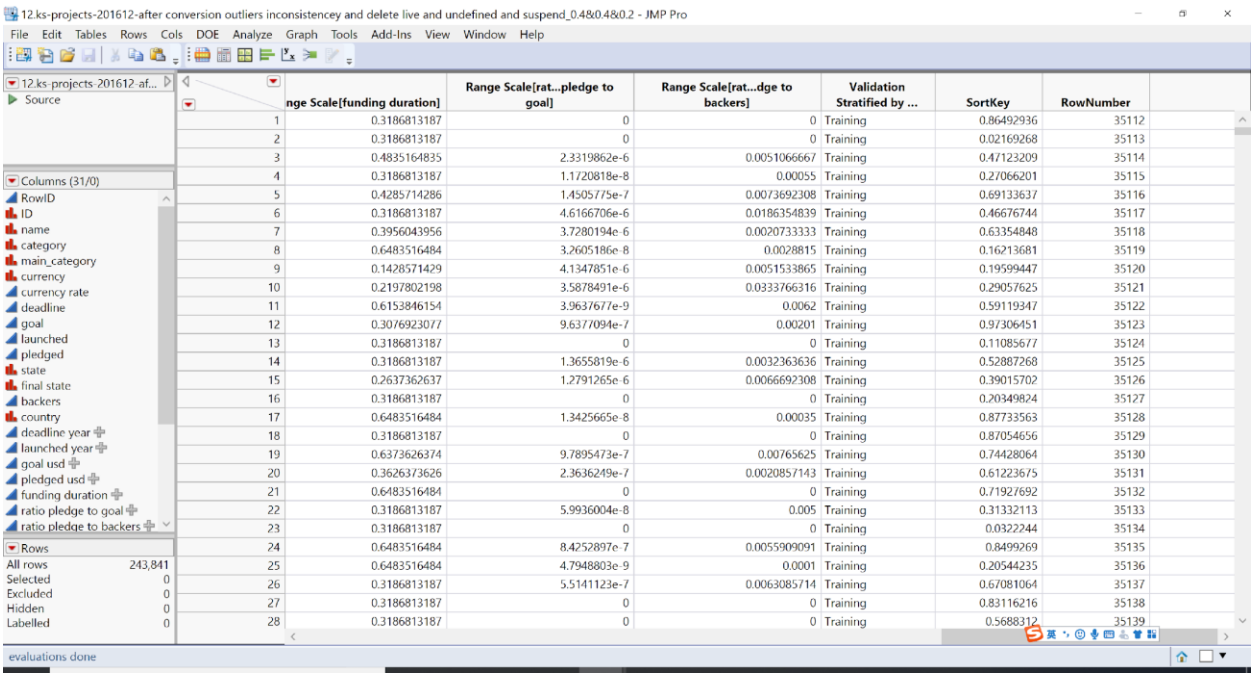
We are getting good accuracy 67.4% for training data set and 66.17% for test data set. This split provides more accuracy (66.17% > 65.83%) for test data set comparing the split “Train 0.4 Validation 0.4 Test 0.2”. But the difference is small, it’s more reasonable to use the data set which delete the live (state). Therefore, we keep the split “Train 0.4 Validation 0.4 Test 0.2” of “delete the live (state)” to build the model.

Data set	**Number of False positive and False Negative Errors**	**Accuracy**
Training	29530	67.38%
Validation	30400	66.41%
Test	21351	66.17%

4. We used the data set “delete live, undefined and suspended (state)” to build the model by using the decision tree model

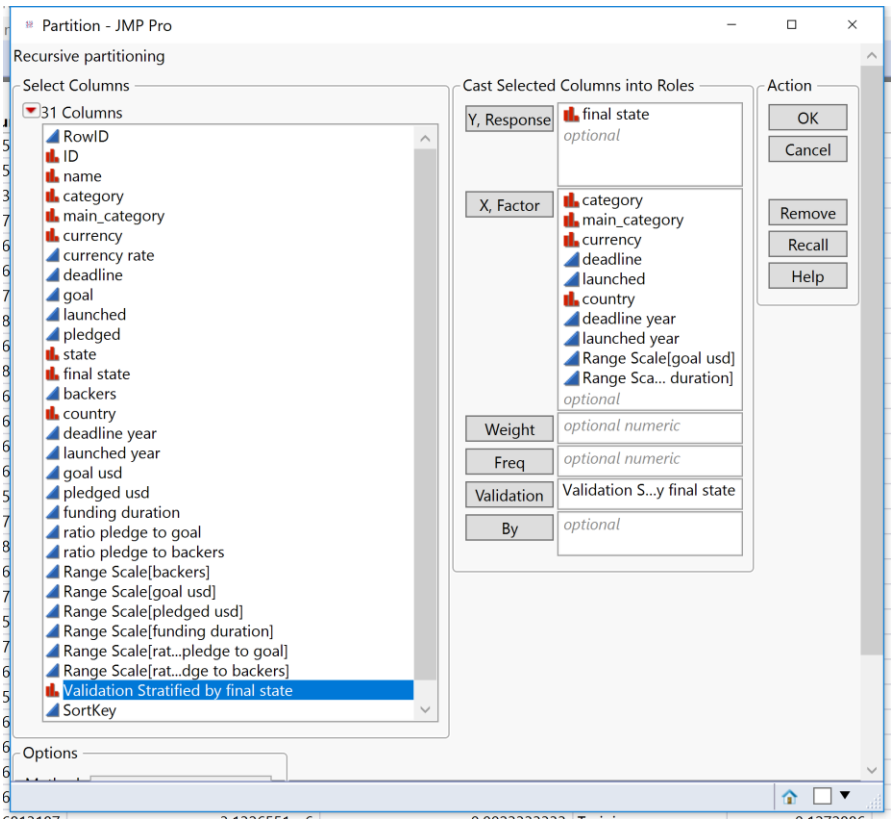
a. Use Stratified proportion: Training set - 0.4 Validation set - 0.4 Testing set - 0.2



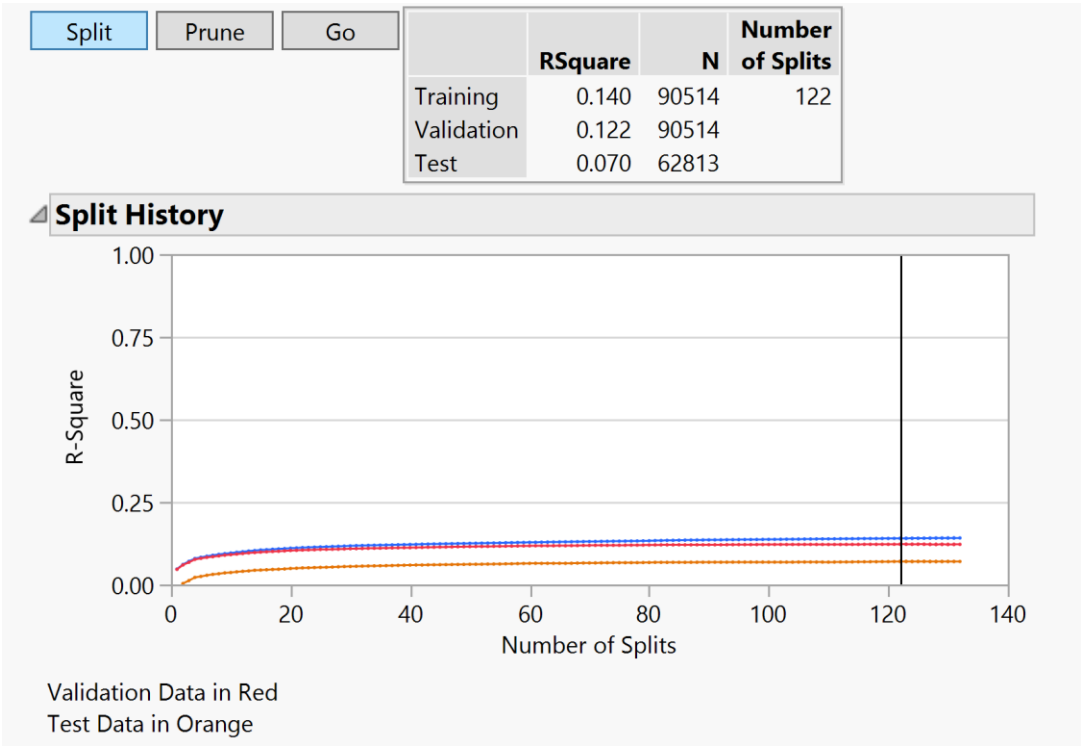


Sample size: 243,841

Input variables:


Partition - JMP Pro
Recursive partitioning
Select Columns
31 Columns
RowID
ID
name
category
main_category
currency
currency rate
deadline
goal
launched
pledged
state
final state
backers
country
deadline year
launched year
goal usd
pledged usd
funding duration
ratio pledge to goal
ratio pledge to backers
Range Scale[backers]
Range Scale[goal usd]
Range Scale[pledged usd]
Range Scale[funding duration]
Range Scale[rat...pledge to goal]
Range Scale[rat...dge to backers]
Validation Stratified by final state
SortKey
Cast Selected Columns into Roles
Y, Response
final state
optional
X, Factor
category
main_category
currency
deadline
launched
country
deadline year
launched year
Range Scale[goal usd]
Range Sca... duration]
optional
Weight
optional numeric
Freq
optional numeric
Validation
Validation S...y final state
By
optional
Action
OK
Cancel
Remove
Recall
Help
Options

Results:


Split
Prune
Go
RSquare
N
Number of Splits
Training 0.140 90514 122
Validation 0.122 90514
Test 0.070 62813
Split History
R-Square
1.00
0.75
0.50
0.25
0.00
0
20
40
60
80
100
120
140
Number of Splits
Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1398	0.1221	0.0698	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2349	0.2076	0.1196	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5963	0.6085	0.6079	∑ -Log(ρ[j])/n
RMSE	0.4537	0.4590	0.4598	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4118	0.4168	0.4163	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3254	0.3350	0.3364	∑ (ρ[j]≠ρMax)/n
N	90514	90514	62813	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	30355	14902
1	14555	30702

Validation

Actual	Predicted Count	
final state	0	1
0	29941	15316
1	15005	30252

Test

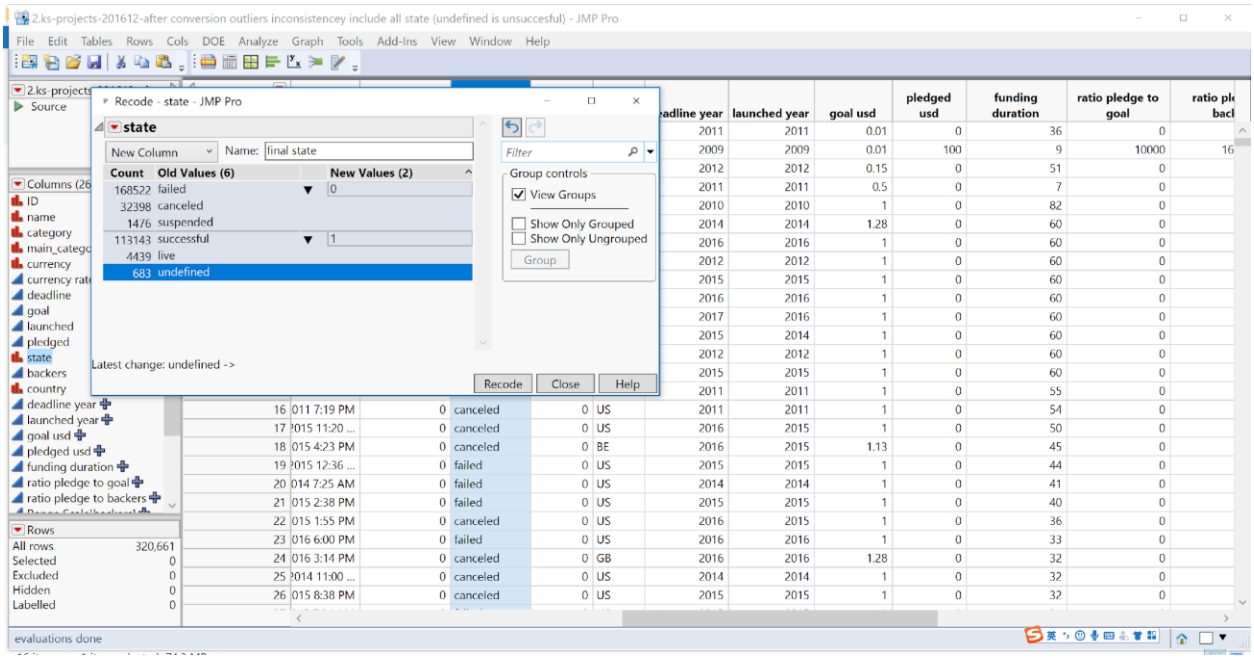
Actual	Predicted Count	
final state	0	1
0	26588	13596
1	7532	15097

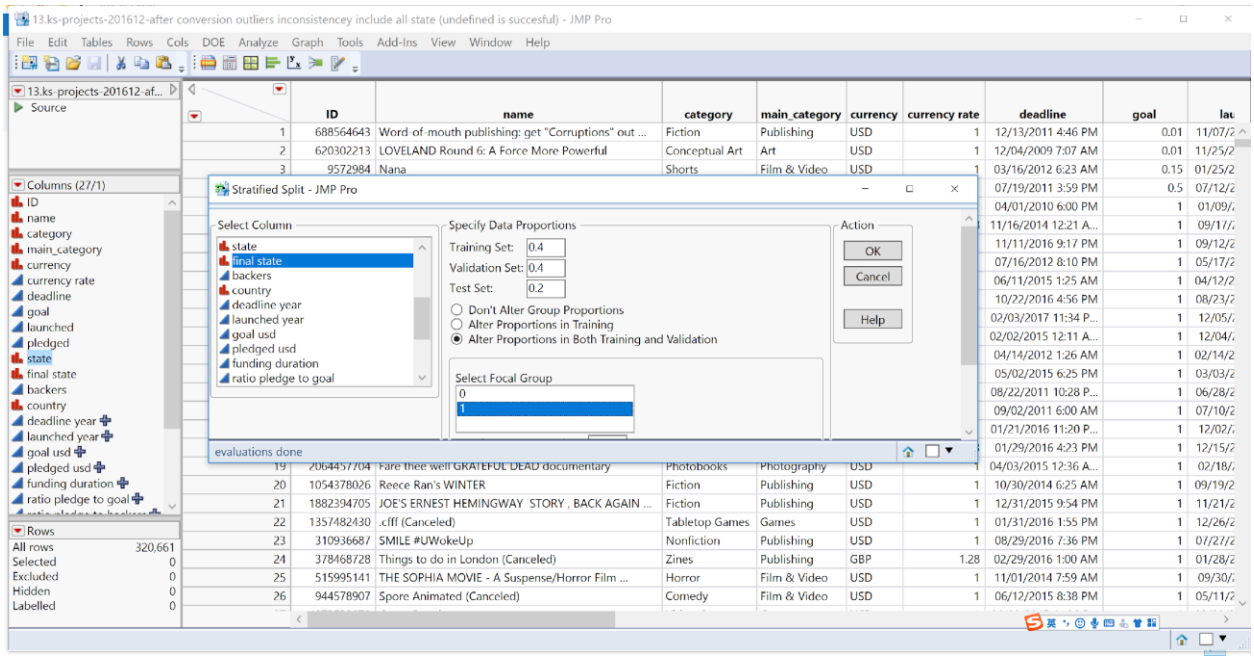
We are getting good accuracy 67.5% for training data set and 66.36% for test data set. This split provides more accuracy (66.36% > 65.83%) for test data set comparing the split “Train 0.4 Validation 0.4 Test 0.2”. But the difference is small, we keep the split “Train 0.4 Validation 0.4 Test 0.2” of “delete the live (state)” to build the model.

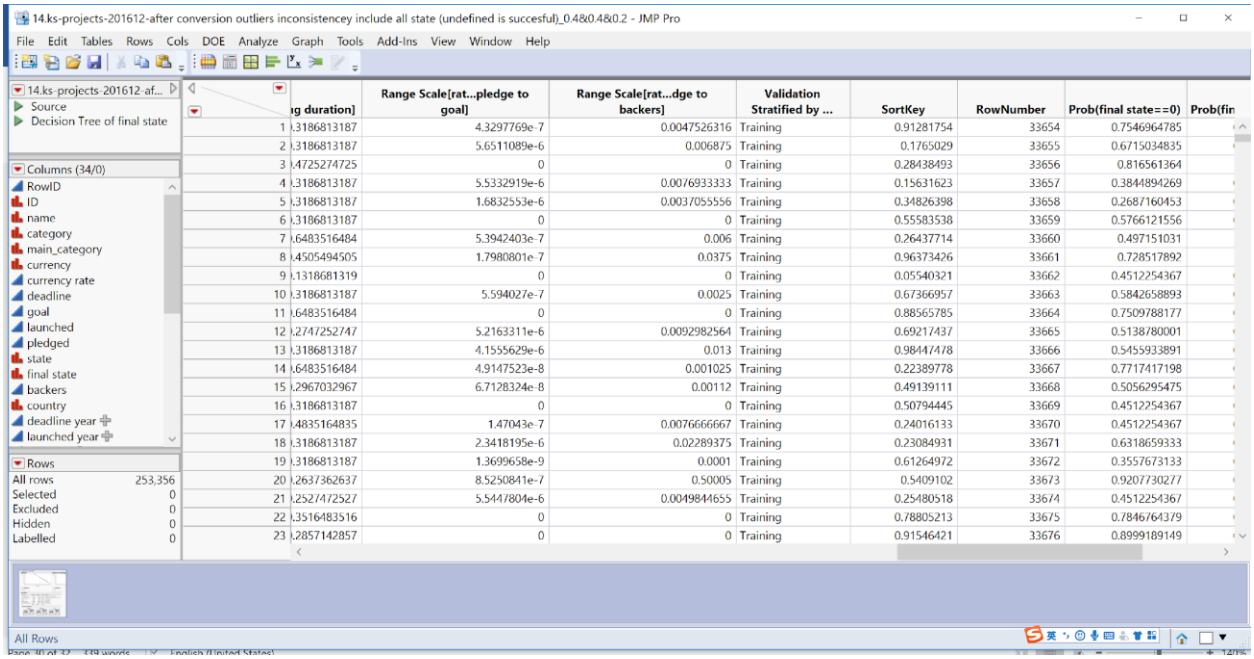
Data set	**Number of False positive and False Negative Errors**	**Accuracy**
Training	29,457	67.46%
Validation	30,321	66.50%
Test	21,128	66.36%

5. We used the data set “all (state)” and “undefined is successful” to build the model by using the decision tree model

a. Use Stratified split proportion: Training set - 0.4 Validation set - 0.4 Testing set - 0.2

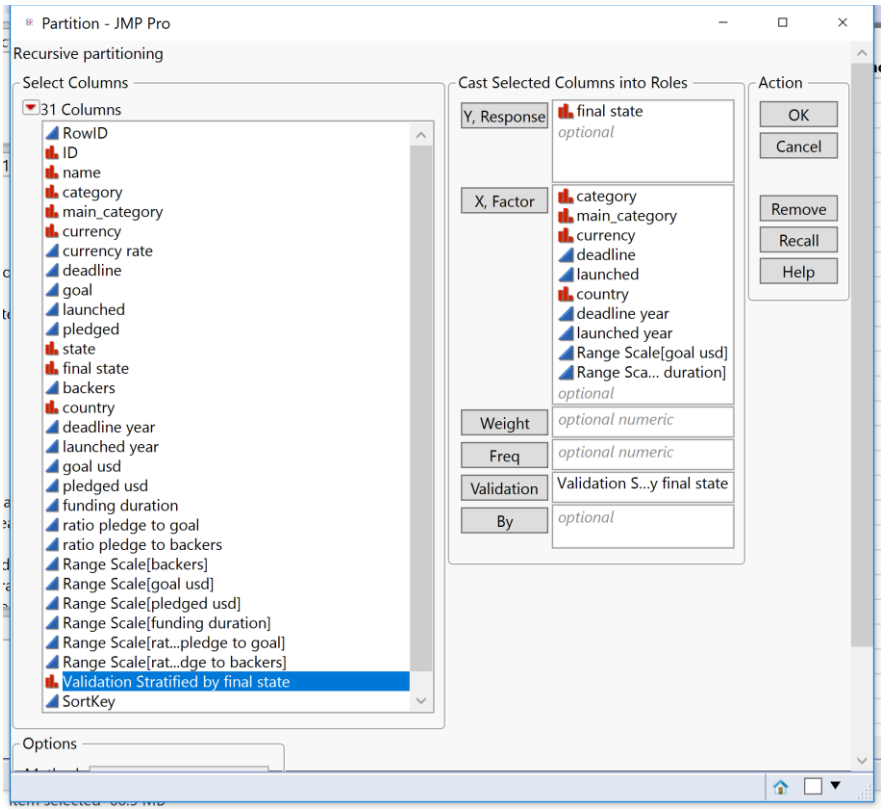

2.ks-projects-201612-after conversion outliers inconsistencey include all state (undefined is unsuccesful) - JMP Pro
Recode - state - JMP Pro
state
New Column
Name: final state
Count Old Values (6) New Values (2)
168522 failed 0
32398 canceled
1476 suspended
113143 successful 1
4439 live
683 undefined
Group controls
View Groups
Show Only Grouped
Show Only Ungrouped
Group
Latest change: undefined ->
Recode
Close
Help
Columns (26
Rows
All rows 320,661
Selected 0
Excluded 0
Hidden 0
Labelled 0
evaluations done


13.ks-projects-201612-after conversion outliers inconsistencey include all state (undefined is succesful) - JMP Pro
Stratified Split - JMP Pro
Select Column
state
final state
backers
country
deadline year
launched year
goal usd
pledged usd
funding duration
ratio pledge to goal
Specify Data Proportions
Training Set: 0.4
Validation Set: 0.4
Test Set: 0.2
Don't Alter Group Proportions
Alter Proportions in Training
Alter Proportions in Both Training and Validation
Select Focal Group
0
1
Action
OK
Cancel
Help
Columns (27/1)
Rows
All rows 320,661
Selected 0
Excluded 0
Hidden 0
Labelled 0
evaluations done



Sample size: 253,356

Input variables:

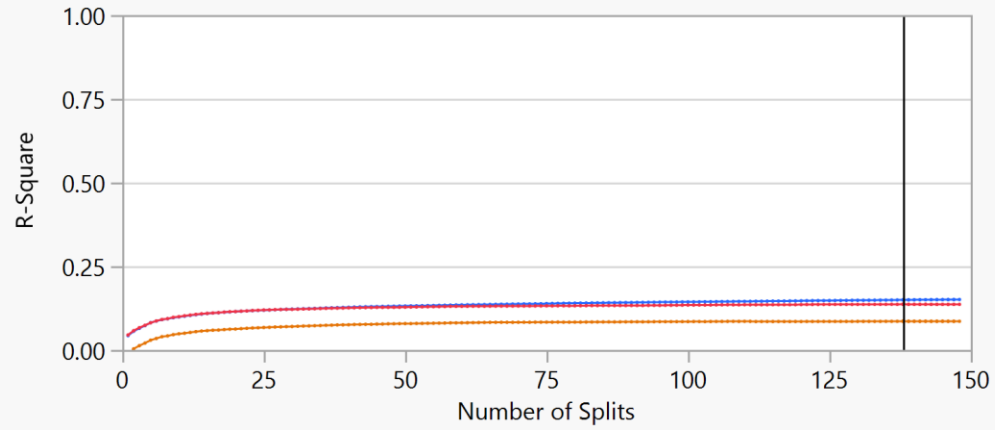


Results:

Split Prune Go

	RSquare	N	Number of Splits
Training	0.150	94612	138
Validation	0.137	94612	
Test	0.086	64132	

Split History



Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1496	0.1365	0.0859	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2498	0.2299	0.1460	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5894	0.5985	0.6018	∑ -Log(ρ[j])/n
RMSE	0.4508	0.4545	0.4566	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4067	0.4103	0.4111	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3219	0.3263	0.3284	∑ (ρ[j]≠ρMax)/n
N	94612	94612	64132	n

Confusion Matrix

Training

Actual	Predicted Count	
final state	0	1
0	32016	15290
1	15161	32145

Validation

Actual	Predicted Count	
final state	0	1
0	31778	15528
1	15348	31958

Test

Actual	Predicted Count	
final state	0	1
0	27153	13326
1	7737	15916

We are getting good accuracy 67.8% for training data set and 67.16% for test data set. This split provides more accuracy (67.16% > 65.83%) for test data set comparing the split “Train 0.4 Validation 0.4 Test 0.2”. But the difference is small, we keep the split “Train 0.4 Validation 0.4 Test 0.2” of “delete the live (state)” to build the model.

Data set	Number of False positive and False Negative Errors	Accuracy
Training	30,451	67.81%
Validation	30,876	67.37%
Test	21,063	67.16%

Comparison of Decision Tree model for every project status

				Training				
Data Set	Training-Validation-Test proportion	Sample Size	Number of splits	Predict 1, Actual 0	Predict 0, Actual 1	Total Error	Total	Ratio
delete observations with status as live	0.3-0.3-0.4	262,261	120	12,241	9,611	21,852	67,886	67.81%
delete observations with status as live	0.2-0.2-0.6	280,247	97	7,979	6,735	14,714	45,256	67.49%
delete observations with status as live	0.4-0.4-0.2	244,273	135	16,071	13,386	29,457	90,514	67.46%
Consider status undefined as unsuccesful	0.2-0.2-0.6	286,462	91	8,075	6,873	14,948	47,034	68.22%
Consider status undefined as unsuccesful	0.4-0.4-0.2	252,264	128	16,436	13,502	29,938	94,066	68.17%
delete status - live and undefined	0.4-0.4-0.2	244,136	102	15,568	13,962	29,530	90,514	67.38%
delete status - live, undefined and suspended	0.4-0.4-0.2	243,841	122	14,902	14,555	29,457	90,514	67.46%
Change status undefined as succesful	0.4-0.4-0.2	253,356	138	15,290	15,161	30,451	94,612	67.81%

				Validation				
Data Set	Training-Validation-Test proportion	Sample Size	Number of splits	Predict 1, Actual 0	Predict 0, Actual 1	Total Error	Total	Ratio
delete observations with status as live	0.3-0.3-0.4	262,261	120	12,808	9,895	22,703	67,886	66.56%
delete observations with status as live	0.2-0.2-0.6	280,247	97	8,435	6,894	15,329	45,258	66.13%
delete observations with status as live	0.4-0.4-0.2	244,273	135	16,521	13,708	30,229	90,514	66.60%
Consider status undefined as unsuccesful	0.2-0.2-0.6	286,462	91	8,402	7,416	15,818	47,032	66.37%
Consider status undefined as unsuccesful	0.4-0.4-0.2	252,264	128	16,767	14,085	30,852	94,066	67.20%
delete status - live and undefined	0.4-0.4-0.2	244,136	102	15,887	14,513	30,400	90,514	66.41%
delete status - live, undefined and suspended	0.4-0.4-0.2	243,841	122	15,316	15,005	30,321	90,514	66.50%
Change status undefined as succesful	0.4-0.4-0.2	253,356	138	15,528	15,348	30,876	94,612	67.37%

				Test				
Data Set	Training-Validation-Test proportion	Sample Size	Number of splits	Predict 1, Actual 0	Predict 0, Actual 1	Total Error	Total	Ratio
delete observations with status as live	0.3-0.3-0.4	262,261	120	30,505	13,579	44,084	126,489	65.15%
delete observations with status as live	0.2-0.2-0.6	280,247	97	45,119	21,032	66,151	189,733	65.13%
delete observations with status as live	0.4-0.4-0.2	244,273	135	14,693	6,915	21,608	63,245	65.83%
Consider status undefined as unsuccesful	0.2-0.2-0.6	286,462	91	43,987	21,746	65,733	192,396	65.83%
Consider status undefined as unsuccesful	0.4-0.4-0.2	252,264	128	14,455	7,051	21,506	64,132	66.47%
delete status - live and undefined	0.4-0.4-0.2	244,136	102	14,067	7,284	21,351	63,108	66.17%
delete status - live, undefined and suspended	0.4-0.4-0.2	243,841	122	13,596	7,532	21,128	62,813	66.36%
Change status undefined as succesful	0.4-0.4-0.2	253,356	138	13,326	7,737	21,063	64,132	67.16%

RESULT - Thus, from the test observation we can conclude that the modeling results we get when considering status live, undefined, suspended as successful is approximately the same as results obtained when dropping or deleting observations with status live, undefined, suspended. Therefore, there is no significant differences in accuracy and number of errors obtained from each of these cases.