

Kick Starters Project Analysis



Team Members:

- 
- 1. Sai Krishna Mannava - 801136361**
 - 2. Kumar Mani Chandra Yelisetty - 801168244**
 - 3. Smirthi Meenakshisundaram - 801129947**



Programming language: R language

Pros:

- ❖ open source
- ❖ data wrangling(using dplyr, readr),
- ❖ easy plotting using ggplot2
- ❖ compatible with other languages
- ❖ platform independent
- ❖ machine learning operation are easy

Cons

- ❖ **data handling is not as easy as in python**
- ❖ **cannot handle big data**
- ❖ **consumes lot of memory**
- ❖ **lacks security**
- ❖ **speed of R is not as much as python or MATLAB**

Our Dataset

ID	name	category	main_category	currency	deadline	goal	launched	pledged	state	backers	country	usd pledged	usd pledged_real	usd goal_real
1000002330	The Songs of Adelaide & Abullah	Poetry	Publishing	GBP	2015-10-09	1000	2015-08-11 12:12:28	0	failed	0	GB	0	0	1533.95
1000003930	Greeting From Earth: ZGAC Arts Capsule For ET	Narrative Film	Film & Video	USD	2017-11-01	30000	2017-09-02 04:43:57	2421	failed	15	US	100	2421	30000
1000004038	Where is Hank?	Narrative Film	Film & Video	USD	2013-02-26	45000	2013-01-12 00:20:50	220	failed	3	US	220	220	45000
1000007540	ToshiCapital Rekordz Needs Help to Complete Album	Music	Music	USD	2012-04-16	5000	2012-03-17 03:24:11	1	failed	1	US	1	1	5000
1000011046	Community Film Project: The Art of Neighborhood Filmmaking	Film & Video	Film & Video	USD	2015-08-29	19500	2015-07-04 08:35:03	1283	canceled	14	US	1283	1283	19500
1000014025	Monarch Espresso Bar	Restaurants	Food	USD	2016-04-01	50000	2016-02-26 13:38:27	52375	successful	224	US	52375	52375	50000
1000023410	Support Solar Roasted Coffee & Green Energy! SolarCoffee.co	Food	Food	USD	2014-12-21	1000	2014-12-01 18:30:44	1205	successful	16	US	1205	1205	1000
1000030581	Chaser Strips. Our Strips make Shots their B*tch!	Drinks	Food	USD	2016-03-17	25000	2016-02-01 20:05:12	453	failed	40	US	453	453	25000
1000034518	SPIN - Premium Retractable In-Ear Headphones with Mic	Product Design	Design	USD	2014-05-29	125000	2014-04-24 18:14:43	8233	canceled	58	US	8233	8233	125000
100004195	STUDIO IN THE SKY - A Documentary Feature Film (Canceled)	Documentary	Film & Video	USD	2014-08-10	65000	2014-07-11 21:55:48	6240.57	canceled	43	US	6240.57	6240.57	65000
100004721	Of Jesus and Madmen	Nonfiction	Publishing	CAD	2013-10-09	2500	2013-09-09 18:19:37	0	failed	0	CA	0	0	2406.39
100005484	Lisa Lim New CD!	Indie Rock	Music	USD	2013-04-08	12500	2013-03-09 06:42:58	12700	successful	100	US	12700	12700	12500
1000055792	The Cottage Market	Crafts	Crafts	USD	2014-10-02	5000	2014-09-02 17:11:50	0	failed	0	US	0	0	5000
1000056157	G-Spot Place for Gamers to connect with eachother & go pro!	Games	Games	USD	2016-03-25	200000	2016-02-09 23:01:12	0	failed	0	US	0	0	200000
1000057089	Tombstone: Old West tabletop game and miniatures in 32mm.	Tabletop Games	Games	GBP	2017-05-03	5000	2017-04-05 19:44:18	94175	successful	761	GB	57763.78	121857.33	6469.73
1000064368	Survival Rings	Design	Design	USD	2015-02-28	2500	2015-01-29 02:10:53	664	failed	11	US	664	664	2500
1000064918	The Beard	Comic Books	Comics	USD	2014-11-08	1500	2014-10-09 22:27:52	395	failed	16	US	395	395	1500
1000068480	Notes From London: Above & Below	Art Books	Publishing	USD	2015-05-10	3000	2015-04-10 21:20:54	789	failed	20	US	789	789	3000
1000070642	Mike Corey's Darkness & Light Album	Music	Music	USD	2012-08-17	250	2012-08-02 14:11:32	250	successful	7	US	250	250	250
1000071625	Boco Tea	Food	Food	USD	2012-06-02	5000	2012-05-03 17:24:32	1781	failed	40	US	1781	1781	5000
1000072011	CMUK. Shoes: Take on Life Feet First.	Fashion	Fashion	USD	2013-12-30	20000	2013-11-25 07:06:11	34268	successful	624	US	34268	34268	20000
1000081649	MikeyJ clothing brand fundraiser	Childrenswear	Fashion	AUD	2017-09-07	2500	2017-08-08 01:20:20	1	failed	1	AU	0	0.81	2026.1
1000082254	Alice in Wonderland in G Minor	Theater	Theater	USD	2014-06-15	3500	2014-05-16 10:10:38	650	failed	12	US	650	650	3500
1000087442	Mountain brew: A quest for alcohol sustainability	Drinks	Food	NOK	2015-02-25	500	2015-01-26 19:17:33	48	failed	3	NO	6.18	6.29	65.55

Pre-processing Data

The screenshot shows an RStudio interface with a hexagonal logo in the top-left corner. The main window displays an R script titled "ML-models-code.R". The code is used for pre-processing a dataset named "ks-projects-201612.csv". It includes:

- Importing the dataset.
- Changing the class of required columns from character to numeric.
- Dropping unwanted columns and selecting the required subset for logistic regression.
- Omitting rows with NA values.
- Listing unique values in the state column.
- Outputting the structure of variables.
- Re-selecting the required subset of columns.
- Omitting rows with NA values again.

The "Console" tab at the bottom shows the execution of the R code, displaying variable structures and the command to re-select the subset of columns.

```
#importing the dataset
dataset=read.csv('ks-projects-201612.csv')

#changing the class of required columns into numeric from chr
str(dataset)
dataset$goal <- as.numeric(dataset$goal)
dataset$backers <- as.numeric(dataset$backers)
dataset$pledged <- as.numeric(dataset$pledged)
str(dataset)

#dropping unwanted columns and take the copy the required subset of the data into the dataset used for logistic regression
lr_dataset=subset(dataset, select=-c(ID, name, category, main_category,deadline,launched,usd.pledged, X, X.1, X.2, X.3))

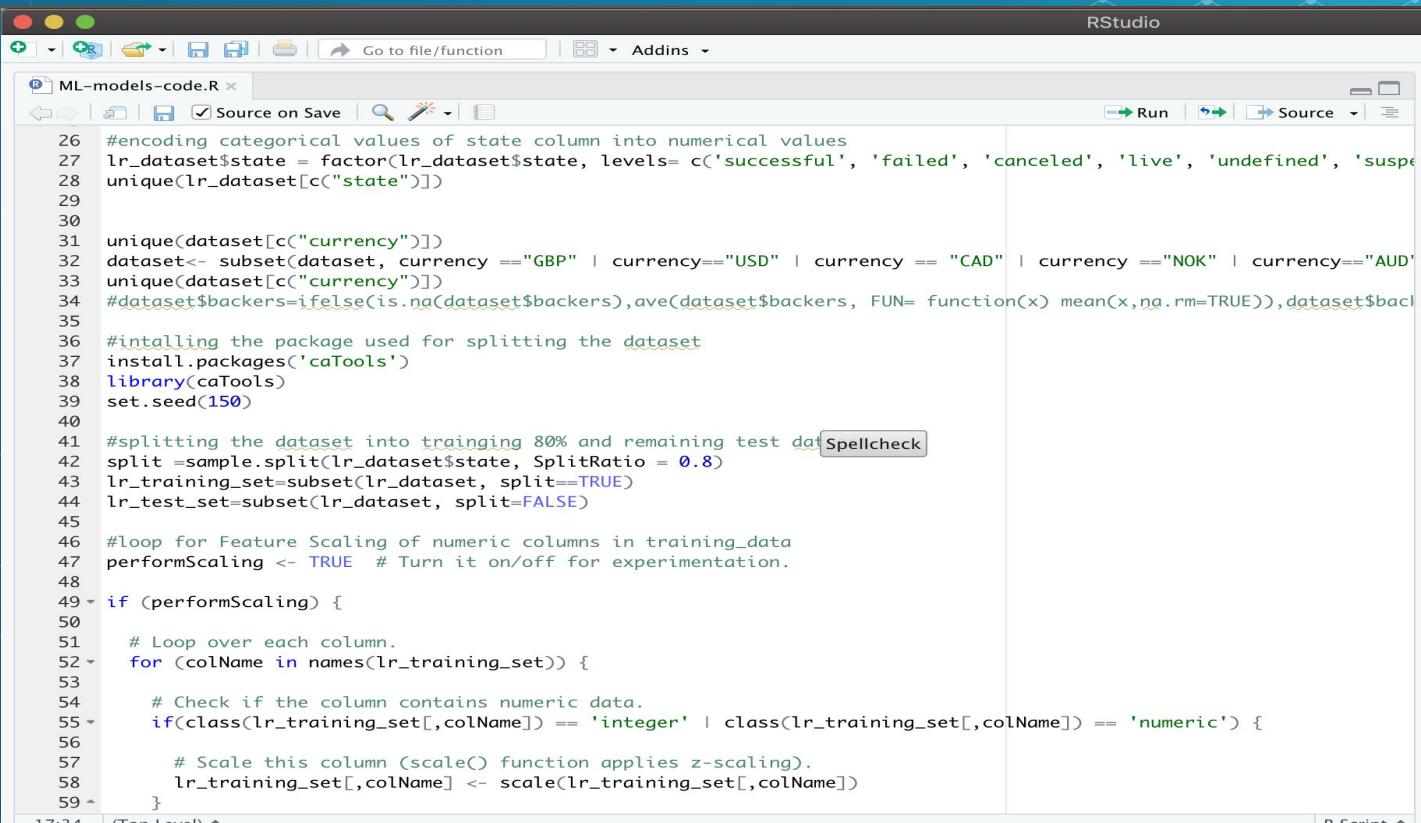
#omitting rows with NA values
lr_dataset <- na.omit(lr_dataset)

# listing unique values in state column
unique(lr_dataset$state)

~/Desktop/SPL_project/ 
$ country      : chr  GB  US  US  US ...
$ usd.pledged  : chr  "0" "220" "1" "1283" ...
$ X            : chr  "" "" "" ...
$ X.1          : chr  "" "" "" ...
$ X.2          : chr  "" "" "" ...
$ X.3          : int   NA NA NA NA NA NA NA NA NA ...

> 
> 
> #dropping unwanted columns and take the copy the required subset of the data into the dataset used for logistic regression
n
> lr_dataset=subset(dataset, select=-c(ID, name, category, main_category,deadline,launched,usd.pledged, X, X.1, X.2, X.3, currency))
> 
> #omitting rows with NA values
> lr_dataset <- na.omit(lr_dataset)
> |
```

Encoding values , splitting & Scaling the testing and training dataset



The screenshot shows an RStudio interface with a teal hexagonal icon in the top-left corner. The main window displays an R script titled "ML-models-code.R". The code performs several tasks:

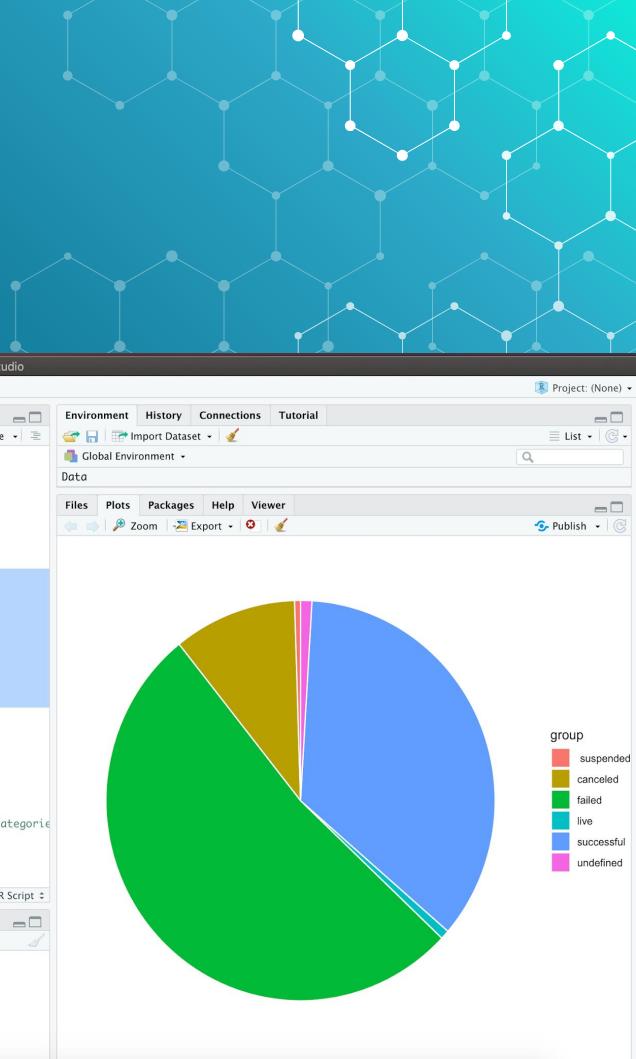
- Encoding categorical values of the "state" column into numerical values.
- Handling missing values in the "backers" column by calculating the mean.
- Splitting the dataset into training (80%) and test (20%) sets.
- Scaling numeric columns in the training data.

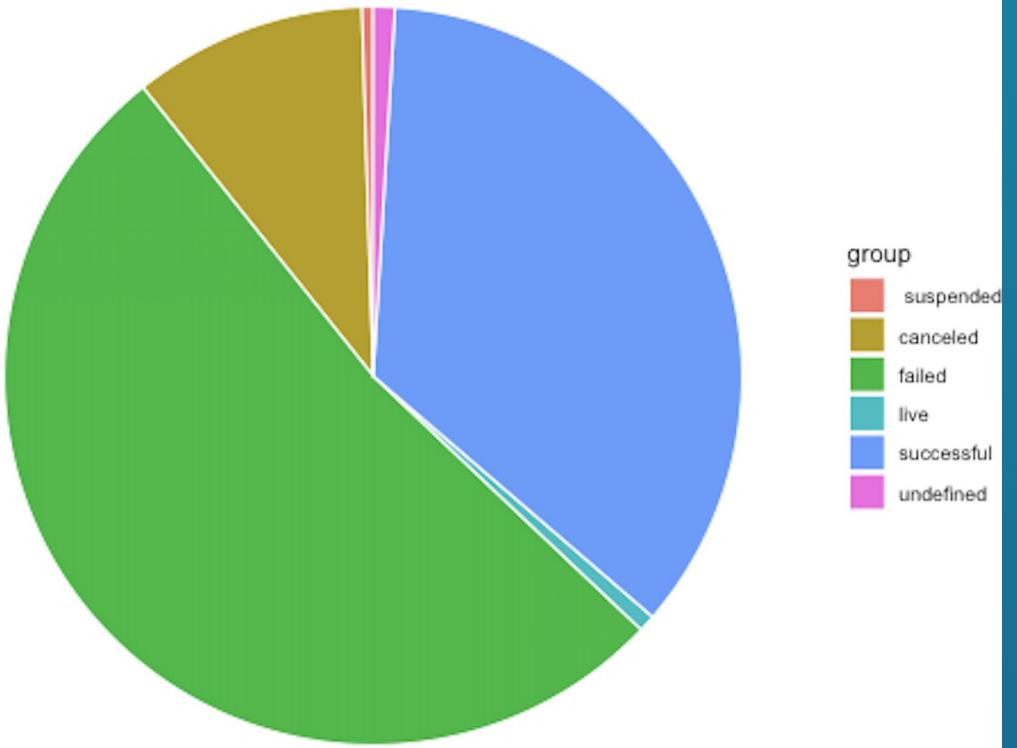
```
26 #encoding categorical values of state column into numerical values
27 lr_dataset$state = factor(lr_dataset$state, levels= c('successful', 'failed', 'canceled', 'live', 'undefined', 'susp
28 unique(lr_dataset[c("state")])
29
30
31 unique(dataset[c("currency")])
32 dataset<- subset(dataset, currency == "GBP" | currency=="USD" | currency == "CAD" | currency == "NOK" | currency=="AUD"
33 unique(dataset[c("currency")])
34 #dataset$backers=ifelse(is.na(dataset$backers),ave(dataset$backers, FUN= function(x) mean(x,na.rm=TRUE)),dataset$back
35
36 #intalling the package used for splitting the dataset
37 install.packages('caTools')
38 library(caTools)
39 set.seed(150)
40
41 #splitting the dataset into trainging 80% and remaining test dat Spellcheck
42 split =sample.split(lr_dataset$state, SplitRatio = 0.8)
43 lr_training_set=subset(lr_dataset, split==TRUE)
44 lr_test_set=subset(lr_dataset, split=FALSE)
45
46 #loop for Feature Scaling of numeric columns in training_data
47 performScaling <- TRUE # Turn it on/off for experimentation.
48
49 if (performScaling) {
50
51     # Loop over each column.
52     for (colName in names(lr_training_set)) {
53
54         # Check if the column contains numeric data.
55         if(class(lr_training_set[,colName]) == 'integer' | class(lr_training_set[,colName]) == 'numeric') {
56
57             # Scale this column (scale() function applies z-scaling).
58             lr_training_set[,colName] <- scale(lr_training_set[,colName])
59         }
60
61     }
62
63 }
64
65 #print(lr_training_set)
66 #print(lr_test_set)
67
68 #write.csv(lr_training_set, "lr_training_set.csv")
69 #write.csv(lr_test_set, "lr_test_set.csv")
```

Visualizing the Data

- Ggplot2, dplyr
- Pie chart to show the projects categorized by their state(status of work)

```
visualizations.R
1 setwd("~/Desktop/SPL_Project")
2 project <- read.csv(file = 'dataset-forgraphs.csv', header = TRUE)
3 head(project)
4 install.packages("ggplot2")
5 library("ggplot2")
6 piechart <- table(project$status)
7 head(piechart)
8 png(file = "piechart_of_status.jpg")
9 data <- data.frame(
10   group=c("canceled","failed","live","successful"," suspended","undefined"),
11   value=c(38779,197719,2799,133956,1846,3562)
12 )
13
14 # Basic piechart
15 ggplot(data, aes(x="", y=value, fill=group)) +
16   geom_bar(stat="identity", width=1, color="white") +
17   coord_polar("y", start=0) +
18   theme_void()
19 dev.off()
20
21 #head(project$main_category)
22 dat <- table(project$main_category)
23 head(dat,15)
24 png(file = "barplot_projects_categories.jpg")
25 #barplot(sort(dat, decreasing = TRUE),main = "Projects in different categories",xlab = "Categories"
26 #print(barplot)
27 ggplot(data = project) + geom_bar(mapping = aes(x = main_category,fill=main_category))
28 dev.off()
29
18:15 | (Top Level) | R Script |
```

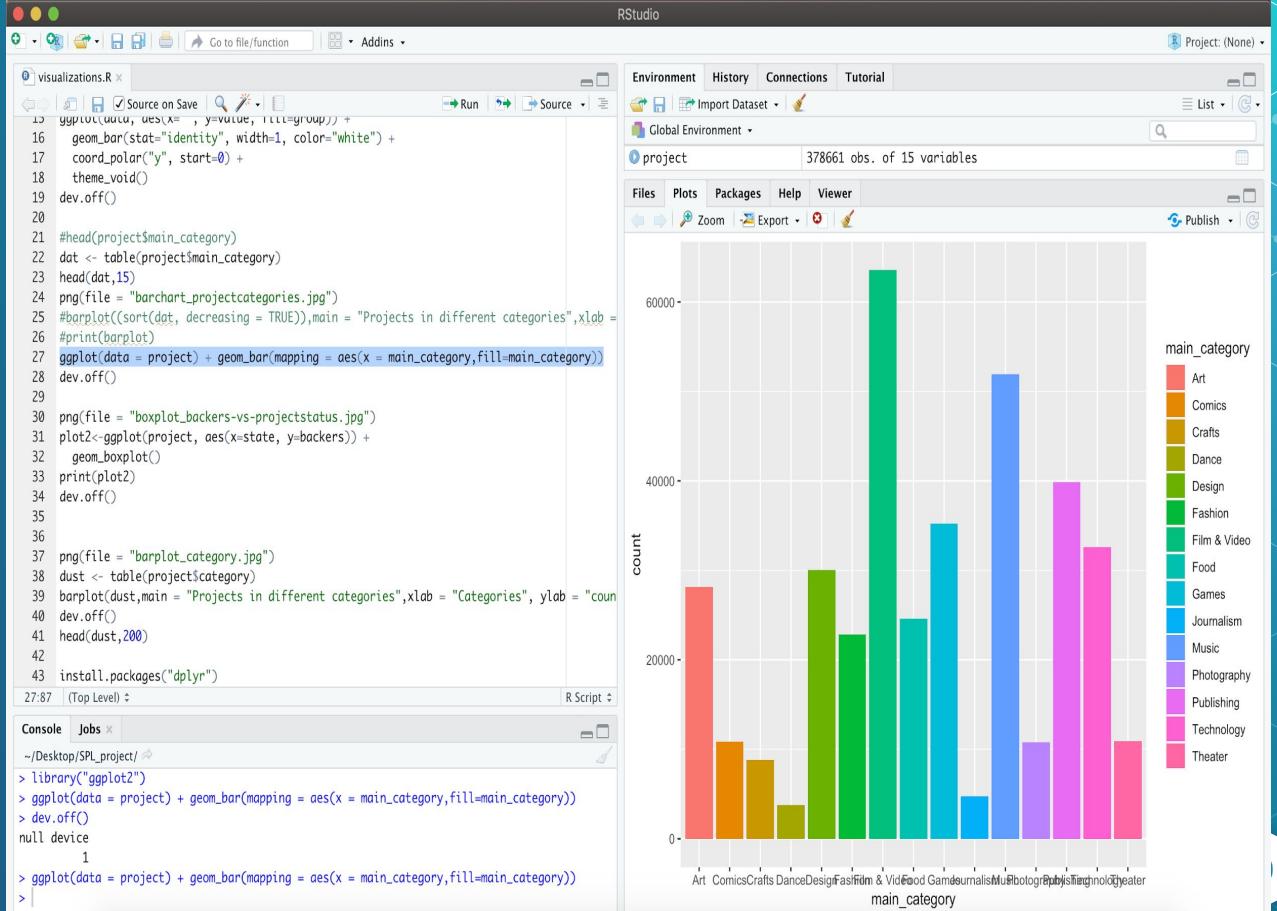


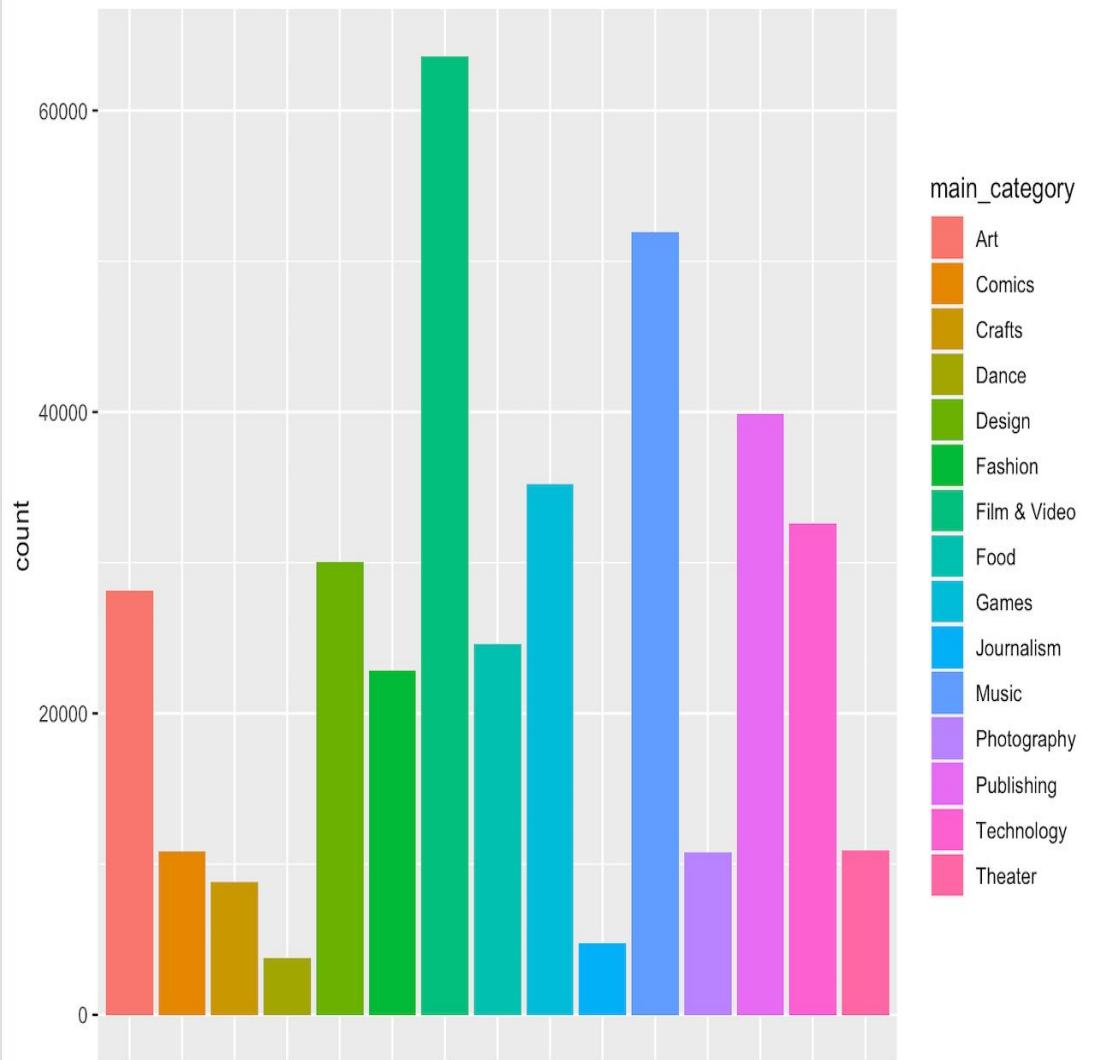


Most of the projects have failed, considerable amount have been successful and some of them have been either canceled, suspended or they are live

Bar chart

Category vs projects

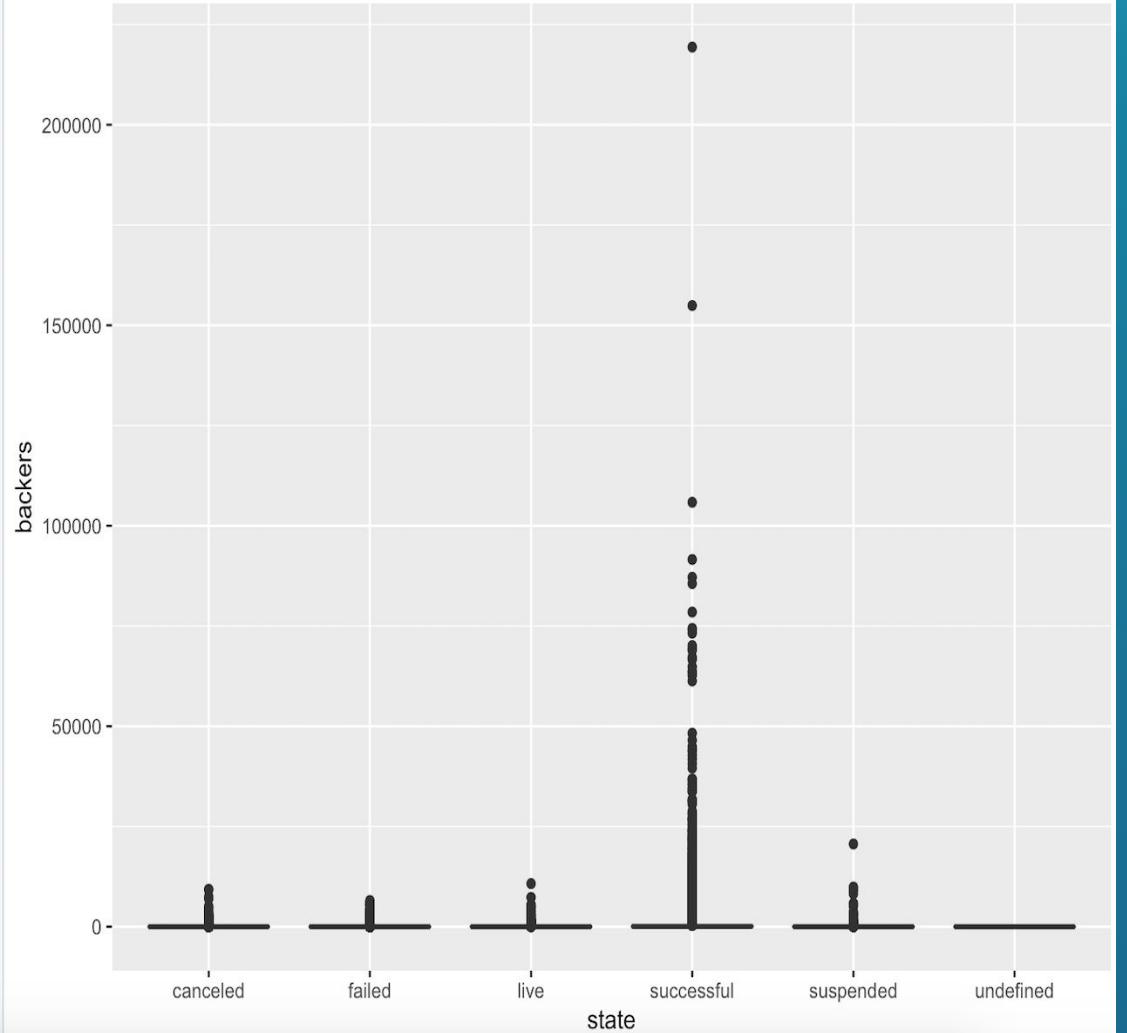




X -axis: category

Y -axis: count of the projects

Film and Video is the category which has the maximum number of projects, followed by Music, Publishing and other Categories.



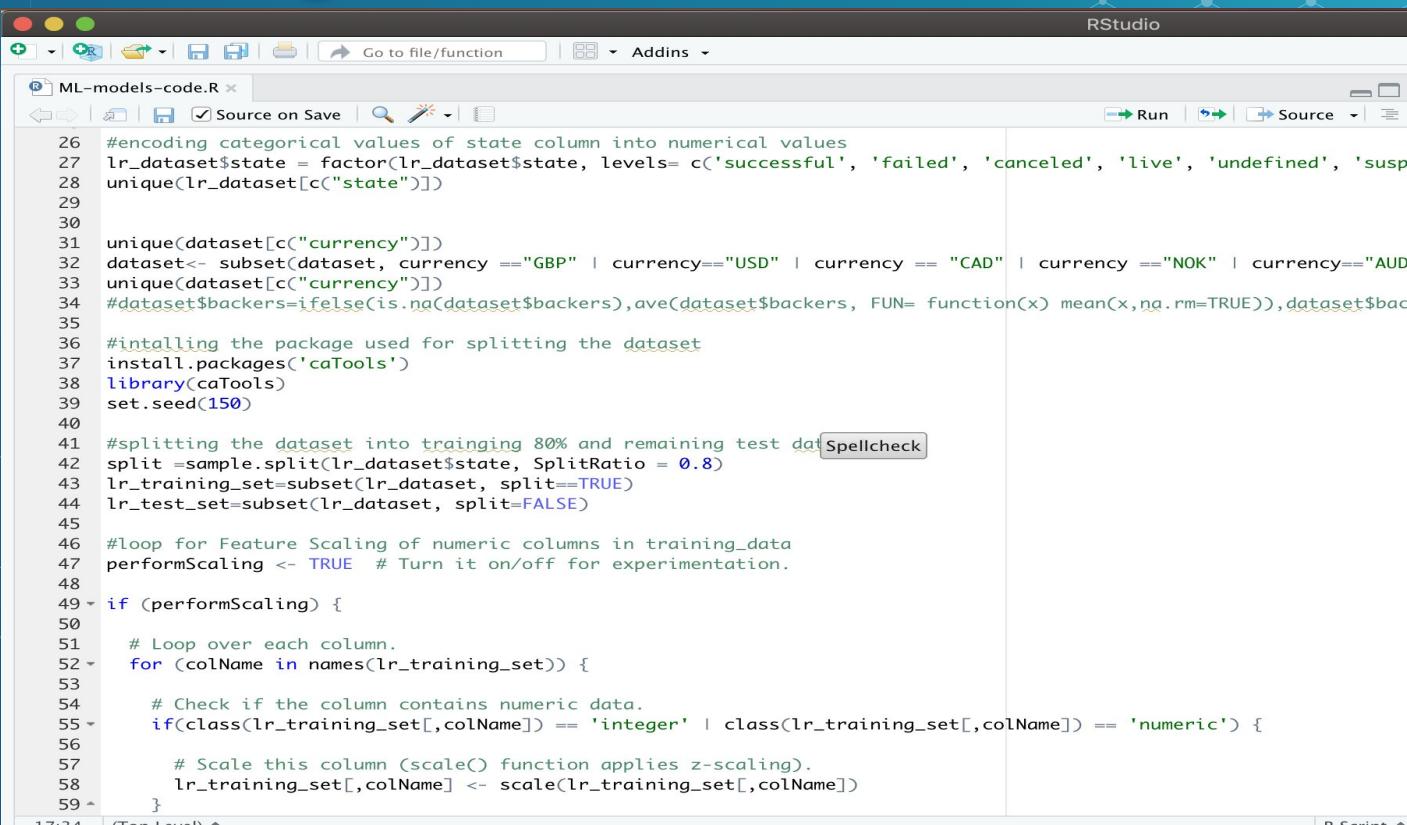
Box Plot Backers vs state

In this plot we see that as the number of backers that a project gets increases there is a greater probability of the project to be successful

Predicting the probability of success using Machine Learning Models

- 1. Pre-processing data**
- 2. Selecting Features for testing and Training the Dataset**
- 3. Encoding values , splitting & Scaling the testing and training dataset**
- 4. Predicting using Linear Regression, Random Forest and Decision Tree Model**

Selecting Features



The screenshot shows the RStudio interface with a hexagonal logo in the top-left corner. The main window displays an R script titled "ML-models-code.R". The code is written in R and performs several tasks:

- Encoding categorical values of the "state" column into numerical values.
- Handling missing values in the "backers" column by calculating the mean.
- Splitting the dataset into training (80%) and testing (20%) sets.
- Scaling numeric columns in the training data.

```
26 #encoding categorical values of state column into numerical values
27 lr_dataset$state = factor(lr_dataset$state, levels= c('successful', 'failed', 'canceled', 'live', 'undefined', 'suspected'))
28 unique(lr_dataset[c("state")])
29
30
31 unique(dataset[c("currency")])
32 dataset<- subset(dataset, currency == "GBP" | currency=="USD" | currency == "CAD" | currency == "NOK" | currency=="AUD")
33 unique(dataset[c("currency")])
34 #dataset$backers=ifelse(is.na(dataset$backers),ave(dataset$backers, FUN= function(x) mean(x,na.rm=TRUE)),dataset$backers)
35
36 #intalling the package used for splitting the dataset
37 install.packages('caTools')
38 library(caTools)
39 set.seed(150)
40
41 #splitting the dataset into trainging 80% and remaining test dat Spellcheck
42 split =sample.split(lr_dataset$state, SplitRatio = 0.8)
43 lr_training_set=subset(lr_dataset, split==TRUE)
44 lr_test_set=subset(lr_dataset, split=FALSE)
45
46 #loop for Feature Scaling of numeric columns in training_data
47 performScaling <- TRUE # Turn it on/off for experimentation.
48
49 if (performScaling) {
50
51   # Loop over each column.
52   for (colName in names(lr_training_set)) {
53
54     # Check if the column contains numeric data.
55     if(class(lr_training_set[,colName]) == 'integer' | class(lr_training_set[,colName]) == 'numeric') {
56
57       # Scale this column (scale() function applies z-scaling).
58       lr_training_set[,colName] <- scale(lr_training_set[,colName])
59     }
59 }
```

Features selected for Testing and Training

RStudio

2.R lr_dataset

Filter

	goal	pledged	state	backers
1	1000	0.00	failed	0
2	45000	220.00	failed	3
3	5000	1.00	failed	1
4	19500	1283.00	canceled	14
5	50000	52375.00	successful	224
6	1000	1205.00	successful	16
7	25000	453.00	failed	40
8	125000	8233.00	canceled	58
9	65000	6240.57	canceled	43
10	2500	0.00	failed	0
11	12500	12700.00	successful	100
12	5000	0.00	failed	0
13	200000	0.00	failed	0
14	2500	664.00	failed	11
15	1500	395.00	failed	16
16	3000	789.00	failed	20
17	250	250.00	successful	7
18	5000	1781.00	failed	40
19	20000	34268.00	successful	624
20	3500	650.00	failed	12

Showing 1 to 20 of 323,118 entries, 4 total columns

ML-models-code.R

```

81 #Fitting the logistic regression to the training set
82 classifier = glm(formula=state ~ ., family =binomial, data=lr_training_set, na.action = na.pass )
83
84 lr_test_set
85 lr_test_set[3]
86 unique(lr_test_set[3])
87
88 #Predicting the test set results
89 prob_pred =predict(classifier, type='response', newdata =lr_test_set[-3] )
90
91 #probability results
92 prob_pred
93 min(prob_pred)
94 max(prob_pred)
95 #converting the propability results to 0 to 1
96 #y_pred =ifelse(prob_pred < 0.167, 0, ifelse (0.167<=prob_pred < 0.33, 1,ifelse(0.33<=prob_pred<0.5,2,ifelse(0.5<= prob
97 y_pred <- ifelse(prob_pred<0.5,0,1)
98 unique(y_pred)
99
100 unique(lr_test_set[,3])
101 #making the confusion matrix
102 cm =table(y_pred,lr_test_set[,3])
103 cm
104 install.packages('e1071')
105 library(e1071)
106
107
108 result1 <- confusionMatrix(cm)
109 result1

```

109:8 (Top Level) ▾

Console Jobs ✎

~/Desktop/SPL_project/ ↵

```

y_pred      0      1
0   111762   4872
1    1319  205165

Accuracy : 0.9808
95% CI : (0.9804, 0.9813)
No Information Rate : 0.65
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9582

McNemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9883

```

Linear Regression Model predicts with an accuracy of 98.08%

ML-models-code.R

```
166 #install.packages('randomForest')
167 library(randomForest)
168
169 RF_classifier = randomForest(x= RF_training_set[-3], y=RF_training_set$state, ntree= 100 )
170
171 RF_prob_pred =predict(RF_classifier, type='response', newdata =RF_test_set[-3] )
172 unique(RF_prob_pred)
173
174 RF_cm =table(RF_prob_pred,RF_test_set[,3])
175 RF_cm
176 install.packages('caret')
177 library(caret)
178 install.packages('e1071')
179 library(e1071)
180
181 result <- confusionMatrix(RF_cm)
182 result
183
184 #install.packages('rpart')
185 library(rpart)
182:7 (Top Level) ↴
```

Console Jobs ~/Desktop/SPL_project/ ↴

```
RF_prob_pred
      0     1
      0 113032 33286
      1      17 176664

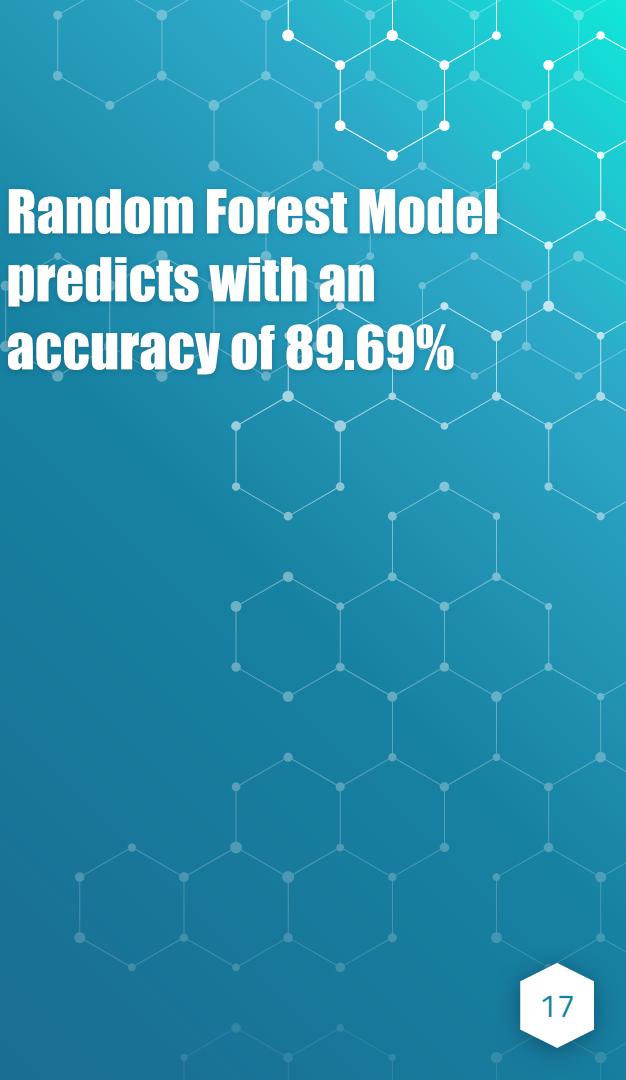
      Accuracy : 0.8969
      95% CI : (0.8958, 0.8979)
      No Information Rate : 0.65
      P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 0.7878

Mcnemar's Test P-Value : < 2.2e-16

      Sensitivity : 0.9998
      Specificity : 0.8415
      Pos Pred Value : 0.7725
      Neg Pred Value : 0.9999
      Prevalence : 0.3500
      Detection Rate : 0.3499
      Detection Prevalence : 0.4530
      Balanced Accuracy : 0.9207

      'Positive' Class : 0
```



Random Forest Model predicts with an accuracy of 89.69%

RStudio

ML-models-code.R

```
180
181 result <- confusionMatrix(RF_cm)
182 result
183
184 #install.packages('rpart')
185 library(rpart)
186
187 dt_classifier =rpart(formula =state ~ ., data=RF_training_set)
188 dt_ypred =predict(dt_classifier, newdata= RF_test_set[-3], type= 'class')
189
190 dt_cm =table(RF_test_set[,3], dt_ypred)
191 dt_cm
192 library(caret)
193 library(e1071)
194
195 result2 <- confusionMatrix(dt_cm)
196:8 (Top Level) ↴
```

Console Jobs

~/Desktop/SPL_project/ ↴

```
dt_ypred
      0     1
0 110430  2619
1 15948 194002

Accuracy : 0.9425
95% CI : (0.9417, 0.9433)
No Information Rate : 0.6087
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.877

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.8738
Specificity : 0.9867
Pos Pred Value : 0.9768
Neg Pred Value : 0.9240
Prevalence : 0.3913
P-value : 0.3462
```

Decision Tree Model
predicts with an
accuracy of 94.25%



Conclusion

- 1. Linear Regression model has given the best results with an accuracy of 98.08% compared to the Random Forest model of accuracy 89.69% and Decision tree model with accuracy 94.35%**
- 2. The visualizations plotted with libraries shows the most endorsed categories, effect of backers on the success of a project and the status of various projects**

THANK YOU

