MISSING VALUE IMPUTATION WITH missFOREST & CLASSIFICATION WITH randomFOREST

KHADEEJA SIKO

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Problem Statement:

- How does R fit into the life of a data scientist?
- How does it make the execution of their daily tasks, accomplishable?

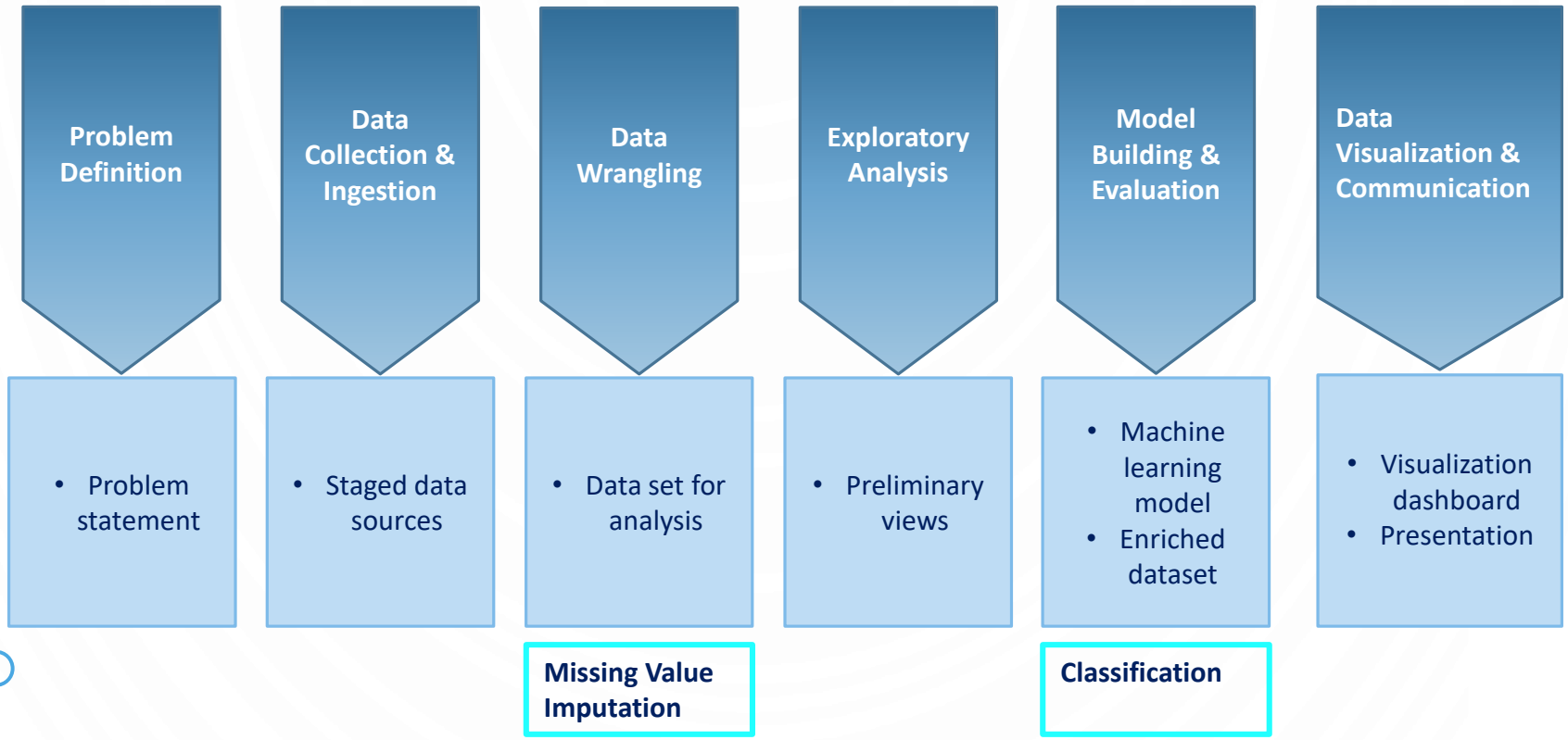
Focus of the presentation:

- Unpacking the principles of the Data Science Use Case Lifecycle
  - Data wrangling and transformation
  - Model build and evaluation
- Provide an example of missing value imputation and classification

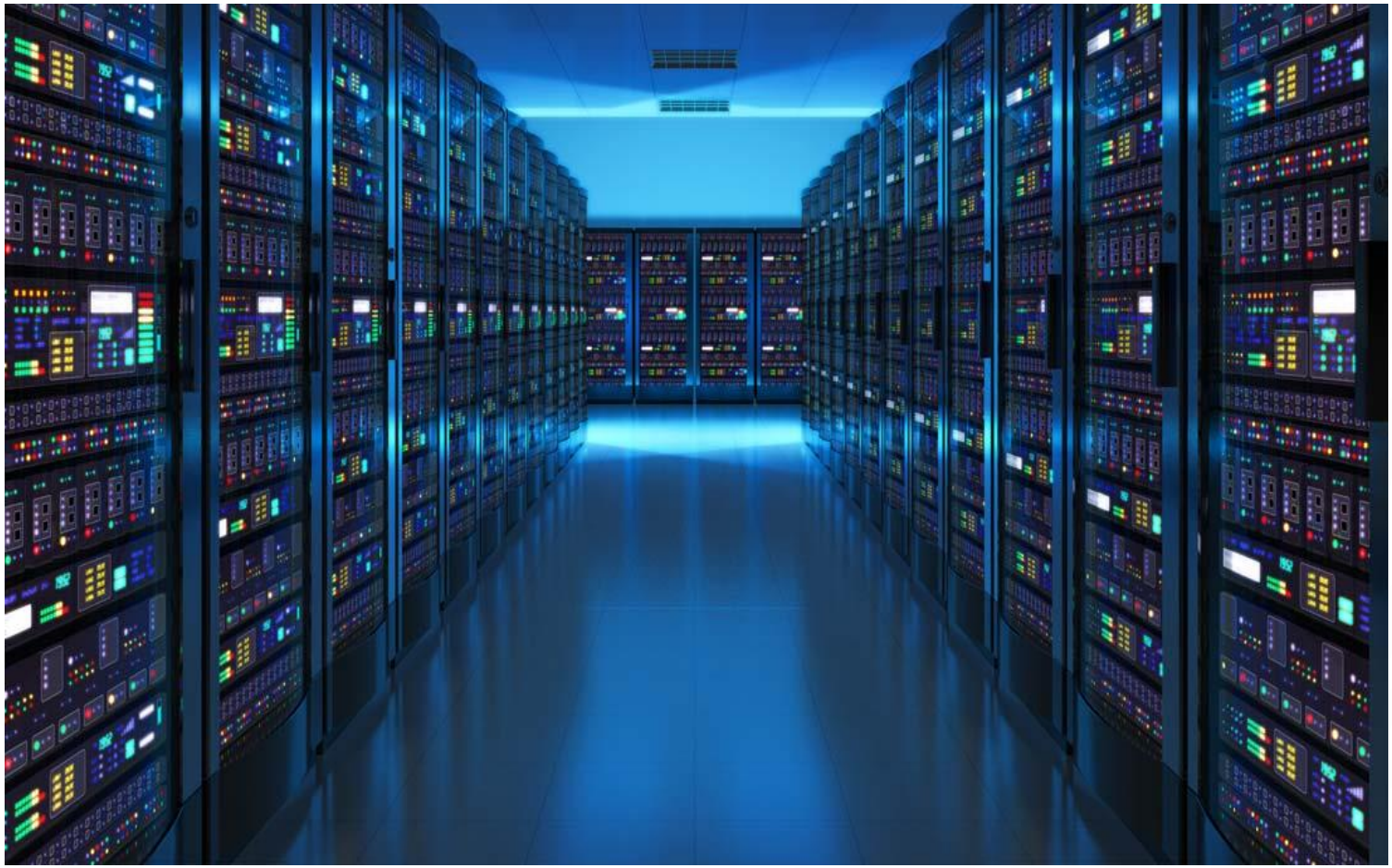
Benefits:

- Assisting students of Applied Statistics by providing them with a real-world example of how they can use R.
- Knowledge-share with Data Scientists and other professionals

DATA SCIENCE USE CASE LIFECYCLE



THE DATA SCIENTIST'S DREAM



THE DATA SCIENTIST'S DREAM (PART 2)

current_ratio_log	number_of_subsidiaries_log	number_of_employees_log	working_capital_effects_log
0.42022255	0.9987220	2.723370	10.81952
0.31722782	1.6901961	4.260494	10.81877
0.14295147	0.6989700	2.880663	10.81980
0.55594505	1.0791812	1.986878	10.81968
0.51245436	1.2552725	2.522902	10.81922
0.41206643	1.0000000	2.813581	10.81960
0.46637203	1.2041200	3.370328	10.81808
0.40459802	1.8692317	4.933958	10.81872
0.62818286	0.6020600	1.689837	10.81944
0.36158682	0.4771213	1.882528	10.81961
0.09939805	1.6434527	3.840232	10.81941
0.43470513	1.8260748	3.813114	10.81728
0.76606275	0.0000000	1.966854	10.82001
0.66436188	0.3010300	2.476714	10.81955
0.38882594	1.3617278	2.576634	10.81950

THE DATA SCIENTIST'S DREAM DEFERRED: MISSING DATA

current_ratio	number_of_subsidiaries	number_of_employees	working_capital_effects	total_revenue	gross_profit	cash_and_cash_equivalents	total_current_assets	cost_of_sales	current_liabilities
NA		NA	NA	NA	NA	NA	NA	NA	NA
NA	48	NA	NA	54144477390	NA	NA	NA	NA	NA
0.389797314	4	NA	NA	1368612799	NA	86173197.36	727745349.6	NA	1866984005
2.597038192	11	NA	4760068.609	318007153.4	141445837.6	199444.2155	132909625.2	176561315.8	51177385.69
2.254275817	17	NA	NA	671648340	NA	240662686.6	276895052.5	NA	122831044.2
1.582655196	9	670	NA	1011461394	628821018.8	99801885.41	362031139.9	382640375.5	228749218.8
1.926658395	15	2365	-238123097	6163344812	2508170565	NA	3386004335	3655174248	1757449241
1.538621887	73	85912	NA	17553404513	2628342353	407743754.1	3169979657	14925048864	2060272042
3.247983871	3	NA	NA	31179779.02	NA	7233176.88	21420308.74	NA	6594955.391
1.299253303	2	NA	-5278623.569	315972822.4	158345410.8	41205174.91	90228563.07	157627411.6	69446475.82
0.257181705	43	6941	-36644550.52	4405390312	NA	NA	492933026.6	396681248.3	1916672207
1.720853312	66	6522	-358959699	19922323126	6703652488	1467058464	9276669016	13218670638	5390737811
4.835294118	0	NA	NA	NA	NA	53530827.43	54647715.03	NA	11301838.88
3.617021277	1	NA	-15835870.71	553736919.8	338217500.6	61016633.65	174048318.7	215532715.5	48119241.05
NA	22	NA	NA	NA	NA	NA	NA	NA	NA
2.999432118	42	858	-22869603.37	1088287306	445744525.3	635522344.4	983193500.8	642542780.8	327793216.2
NA	9	NA	NA	NA	NA	NA	NA	NA	NA
2.585813347	38	2211	-75456394.85	6136659176	1907936550	1495087024	3654190323	4228722626	1413168637
NA	8	NA	NA	240330279.6	NA	NA	NA	NA	NA
NA	1	NA	NA	NA	NA	NA	NA	NA	NA
1.730252948	19	24106	-1781488172	10898614000	2187461718	1551035279	5793217570	8711152283	3516066980
2.372415723	34	2451	-71329024.8	1465698693	689220901.7	75312241.92	466493750.2	776484439.1	435683394.6
2.060497466	13	4479	NA	3852451169	3000744592	640747782.8	1162041777	851706577.7	563961759.9
0.923076923	1	NA	-5424882.66	169939767.8	62213298.94	119666.5293	40048398.46	107726468.9	43385765
1.430498634	7	116	-6820992.168	233057213.9	146764350	66242072.09	111382946.2	86292863.89	77863021.71
NA	6	NA	NA	NA	NA	NA	NA	NA	NA
NA	43	NA	NA	NA	NA	NA	NA	NA	NA
1.060308641	13	8072	NA	20007618769	5784680025	869058224.4	6505498012	14222938744	6135475807
1.712268664	41	NA	380313526.3	13044609023	2317235969	396734433.4	3736055525	10727359758	2181933013
1.9	2	NA	41298000	1099623000	NA	96140000	413094000	921612000	217417894.7
0.92	1	NA	-62477000	2390155000	2390155000	21253000	256882000	0	NA
1.25	1	NA	0	50300328	12275974	1874821	17133794	38024354	13707035.2
1.600149198	37	NA	NA	513396003.1	NA	100971958.1	570410456.2	NA	356473294.4

MISSING VALUE IMPUTATION

In statistics imputation is the process of replacing missing data with substituted values.

Type of imputation	Method	Pros	Cons
Hot deck	Randomly select similar record from the same dataset (e.g. last observation carried forward)	Useful if recent	Can be very inaccurate (e.g. as in financials)
Cold deck	Randomly select similar record from another dataset	None that I know of	Can be very inaccurate (e.g. as in financials)
Mean substitution	Substitute with the mean of that variable for all other available cases	Does not change the sample mean for that variable	Reduces the effect of any correlations involving the variable(s) that are imputed
Regression	A regression model is estimated to predict observed values of a variable based on other variables, and that model is then used to impute values in cases where that variable is missing.	Imputation often doesn't alter the distribution of the variable(s).	Imputed data do not have an error term, Relationships are over identified and suggest greater precision in the imputed values than is warranted

missFOREST PACKAGE

Using missForest package in R for missing value imputation

Package ‘missForest’

Title Nonparametric Missing Value Imputation using Random Forest

Description The function ‘missForest’ in this package is used to impute missing values particularly in the case of mixed-type data. It uses a random forest trained on the observed values of a data matrix to predict the missing values. It can be used to impute continuous and/or categorical data including complex interactions and non-linear relations. It yields an out-of-bag (OOB) imputation error estimate without the need of a test set or elaborate cross-validation.

DATA INGESTION & TRANSFORMATION

```
###Missing Value Imputation with missForest and Classification with randomForest###

#Read in the packages you'll need
library("missForest", lib.loc="/mnt/nfs/anaconda2/envs/rserver/lib/R/library")
library(dplyr)
library("randomForest", lib.loc="/mnt/nfs/anaconda2/envs/rserver/lib/R/library")
library(data.table)

#Read in data
mf_rf <- read.csv(file="R_User_group_preso/mf_rf.csv", header=TRUE, sep=",")

#Summarise the data
summary(mf_rf)

#Replace some of the figures with NA
mf_rf$days_inventory_outstanding[mf_rf$days_inventory_outstanding=='Inf']<- NA
mf_rf$days_sales_outstanding[mf_rf$days_sales_outstanding=='Inf']<-NA
mf_rf$days_payables_outstanding[mf_rf$days_payables_outstanding=='Inf']<-NA
mf_rf$target_variable[mf_rf$target_variable=='']<-NA



#Subset for the variables which will be imputed,as well as the target variable
df<-mf_rf[c(3:23)]

#Log-transform var.s
df$current_ratio_log <- log10(df$current_ratio - min(df$current_ratio,na.rm=TRUE) + 1)

df$total_revenue_log<-log10(df$total_revenue -min(df$total_revenue,na.rm=TRUE) +1)

df$gross_profit_log<-log10(df$gross_profit -min(df$gross_profit,na.rm=TRUE) + 1 )
```

RUNNING missFOREST

```
> dim(na.omit(df2)) #I only remain with 33 observations if I drop rows with na units. Not viable. Should impute.
[1] 33 20
> df2miss <- missForest(df2[-c(1)], maxiter = 20, ntree = 100,
+                       variablewise = TRUE,
+                       decreasing = FALSE,
+                       #verbose = FALSE,
+                       mtry = floor(sqrt(ncol(df2))),
+                       replace = TRUE,
+                       #classwt = NULL, cutoff = NULL, strata = NULL,
+                       sampsize = NULL,
+                       nodesize = c(1,5),
+                       #maxnodes = NULL,
+                       #xtrue = NA,
+                       parallelize = 'no')
  missForest iteration 1 in progress...done!
  missForest iteration 2 in progress...done!
  missForest iteration 3 in progress...done!
  missForest iteration 4 in progress...done!
  missForest iteration 5 in progress...done!
  missForest iteration 6 in progress...done!
> df2_imputed <- data.frame(df2miss$ximp)
> dim(na.omit(df2_imputed)) #407 full observationss
[1] 407  19
```

RESULTS OF IMPUTATION

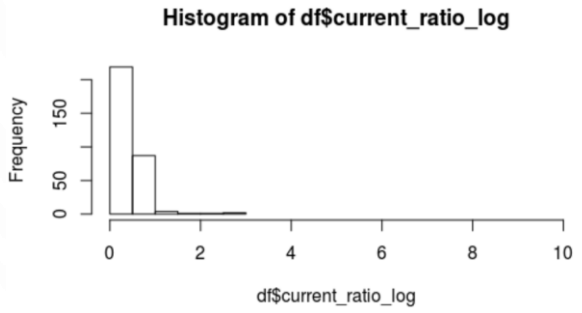
Enriched dataset

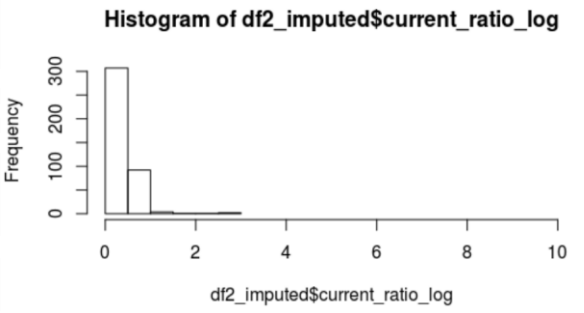
current_ratio_log	number_of_subsidiaries_log	number_of_employees_log	working_capital_effects_log
0.42022255	0.9987220	2.723370	10.81952
0.31722782	1.6901961	4.260494	10.81877
0.14295147	0.6989700	2.880663	10.81980
0.55594505	1.0791812	1.986878	10.81968
0.51245436	1.2552725	2.522902	10.81922
0.41206643	1.0000000	2.813581	10.81960
0.46637203	1.2041200	3.370328	10.81808
0.40459802	1.8692317	4.933958	10.81872
0.62818286	0.6020600	1.689837	10.81944
0.36158682	0.4771213	1.882528	10.81961
0.09939805	1.6434527	3.840232	10.81941
0.43470513	1.8260748	3.813114	10.81728
0.76606275	0.0000000	1.966854	10.82001
0.66436188	0.3010300	2.476714	10.81955
0.38882594	1.3617278	2.576634	10.81950

Minimal error

colnames.df2miss.ximp.	df2miss.OOBerror
current_ratio_log	0.05290043
number_of_subsidiaries_log	0.16487103
number_of_employees_log	0.28577578
working_capital_effects_log	0.56907004
total_revenue_log	0.25406204
gross_profit_log	0.41154996
cash_and_cash_equivalents_log	0.54324166
total_current_assets_log	0.30296306

Unchanged distribution of variables





randomForest PACKAGE

Using randomForest package in R for classification

Classification : the problem of identifying to which of a set of categories a new observation belongs

Package 'randomForest'

Description Classification and regression based on a forest of trees using random inputs

Random forests or **random decision forests** are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

RUNNING randomFOREST

```
> forest_high <-randomForest( high~.,
+                             model_set_high,
+                             ytest=model_set_high$high, ntree=10000,
+                             mtry=floor(sqrt(ncol(model_set_high))), keep.forest = T, importance = T,nPerm =10,
+                             corr.bias=T,
+
+                             replace=TRUE,
+                             classwt=c(0.7045455,0.2954545),
+                             cutoff=c(0.75,0.25),
+                             #strata,
+                             type='classification' )
> forest_high
```

```
> forest_high

Call:
 randomForest(formula = high ~ ., data = model_set_high, ytest = model_set_high$high,      ntree = 10000, mtry =
 floor(sqrt(ncol(model_set_high))),      keep.forest = T, importance = T, nPerm = 10, corr.bias = T,      replace
 = TRUE, classwt = c(0.7045455, 0.2954545), cutoff = c(0.75,          0.25), type = "classification")
               Type of random forest: classification
                     Number of trees: 10000
No. of variables tried at each split: 4

        OOB estimate of  error rate: 59.09%
Confusion matrix:
  0  1 class.error
0 3 48   0.9411765
1 4 33   0.1081081
```

```
> #Predict
> forest_pred_high <- predict(forest_high, newdata = pred_set_high, type = "prob")
> Predicted_low_med_high <- cbind(Predicted_low_med,forest_pred_high)
```

RESULTS OF CLASSIFICATION

Company Name	Sector	target_variable
aa	OTHER SERVICES	Low
ab	BANKS	Low
ac	OTHER SERVICES	Medium
ad	CHEMICALS, RUBBER, PLASTICS, NON-METALLIC PRODUC...	High
ae	OTHER SERVICES	Low
af	OTHER SERVICES	Medium
ag	CHEMICALS, RUBBER, PLASTICS, NON-METALLIC PRODUC...	High
ah	OTHER SERVICES	Low
ai	OTHER SERVICES	Low

BUSINESS CLASSIFICATION PROBLEMS

Common business use cases involving classification :

- Cross-sell and up-sell
- New Business
- Default & financial distress signaling
- Fraud detection
- Churn

Thank You