# Modelling (basic)

## Predictors for modelling

When the data is imported into DataRobot, the raw features will go through “a reasonableness” check for useful information (e.g., are not duplicates or reference ids and not a constant value)”. Predictors that pass this screener will form Informative Features which DataRobot deems to be more appropriate for modelling.

DataRobot deemed all variables informative including the case identifier, Pt\_CaseNumber. The uniqueness of the case identifier must have been disrupted when the dataset was split up between unseen data and data for DataRobot. A custom Feature List, L1,included all variables except Pt\_CaseNumber.

## Modelling options

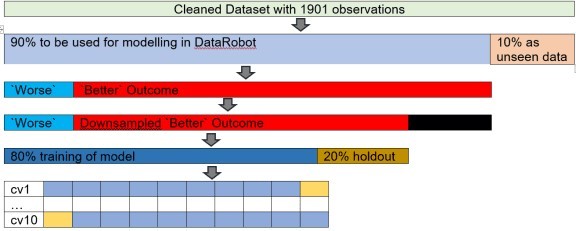
The number of K-fold was increased from the default 5 to 10 folds. The increase in K-folds helps to reduce biasness and therefore reduces the risk of underfitting. Based on the concept of bias and variance trade off, the decrease of biasness will increase the variance. Nonetheless, the sample size of the training set is large enough (1840 rows) to accommodate this increase in folds without adversely compromising variance.

Additionally, 10-fold cross validation is the default number of folds for some modelling programs; for example, R’s modern modelling meta-package, Tidymodels.

DataRobot was also permitted to search for Interactions among the variables. There is support in identifying interactions as the combined effect of two or more variables maybe different than the impact of the individual constituent variables.

Down-sampling was introduced to address the imbalanced dataset (DataRobot only allows down- sampling). The proportion of positive and negative outcome of the binary classification was imbalanced which [could affect model performance as most models work best when the classes are about equal. There were 6.25 more Better outcomes than Worse outcomes. DataRobot’s Down-sampling was used to reduce the disparity.

Under-sampling can help to address imbalance dataset. The number of Better outcomes were reduced by 20%. Post down-sampling, there was 5.22 Better outcomes than Worse outcomes.



## Performance Metric

The performance metric for the binary classification was Area Under the Curve (AUC) which is common in the medical literature. In DataRobot, when down-sampling is used, the performance metric AUC becomes a weighted AUC.

## 1st Quick Mode

The experiment ran on Quick Mode (DataRobot has 3 modelling modes: Autopilot, Quick, Manual. The student version which I used only has Quick and Manual Mode).

# Modelling (2nd run)

The results from the first round of Quick Mode were used for advanced feature selection. A smaller but possibly more influential subset of variables were used in a second round of Quick Mode. The advanced feature selection used results from the first round of Quick Mode to create 2 new Feature Lists. The creation of these 2 new Feature List required RStudio to be connected to DataRobot via its API.

## DataRobot API

library(datarobot) library(tidyverse) theme\_set(theme\_light())

### Connecting to DataRobot

DataRobot was connected to RStudio via datarobot::ConnectToDataRobot. The token can be found under Developer Tools option.

ConnectToDataRobot(endpoint = "https://app.datarobot.com/api/v2",

token = "API token key") ## [1] "Authentication token saved"

### Saving the relevant project ID

projects\_inDR<-data.frame(name=ListProjects()$projectName, ID=ListProjects()$projectId)

project\_ID<-projects\_inDR %>% filter(name=="Classify CAP") %>% pull(ID) ## # A tibble: 11 x 2

## name ID

##

## 1 Classify CAP 5f54dce5e5e3f924b99408ee

## 2 CAP Experiment 3 5f3678ff735d1f59f99534ed

## 3 CAP Experiment 2 5f33f9899071ae44ded6de84

## 4 CAP Experiment 1 5f269f896c6a7a1cefaff780 ## 5 CAP informative ft 5f2694a9d6e13f1dc8ae17d4 ## 6 CleanDR OG cv10 5f2655a8d6e13f1c26ae17c7

## 7 CleanDR OG cv5 5f263758ce243f1bfac853eb ## 8 P04\_Readmission(1).csv 5f16dd3b99c1372d672bdde4 ## 9 P03\_Advertising(1).csv 5f0da6861c29b917239402dc ## 10 P02\_LendingClubData2(1).csv 5f047ce3c40c123f3487346b ## 11 P01\_TitanicData.csv 5efb44cc60882464b616c03b

### Saving the models

List\_Model3<-ListModels(project\_ID) %>% as.data.frame(simple = F)

List\_Model3 %>% select(modelType, expandedModel, featurelistName, `Weighted AUC.crossValidation`) %>% arrange(-`Weighted AUC.crossValidation`)DT: datatable(rownames = F, options = list(searchHighlight = TRUE, paging= T))

In the 1st round of Quick Mode, 13 models were created. The best model was Extreme Gradient Boosted Tree Classifier, with a 10 fold cross validation weighted AUC of 0.9299.

## Advanced Feature Selection (DR Reduced Features)

During the first round of Quick Mode, DataRobot created other Feature Lists including DR Reduced Features. These feature lists contained curated variables consisting of most important features based on Feature Impact scores from models.

Ft\_DR<-GetFeaturelist(project\_ID, "5f54e0f611a5fc3a03274070") %>% as\_tibble() Ft\_DR$description %>% unqiue()

## [1] "The most important features based on Feature Impact scores from a particular model."

3/13 models used DR Reduced Features in the 1st round of Quick Mode.

List\_Model3 %>% count(featurelistName) ## # A tibble: 3 x 2

## featurelistName n ##

## 1 DR Reduced Features M8 3

## 2 L1 9

## 3 Multiple featurelists 1

Originally 70 variables were used for Quick Mode. 33 variables were identified for DR Reduced Features M8 (32 predictors + 1 outcome).

n\_distinct(Ft\_DR$features) ## [1] 33

## Advanced Feature Selection (Top 32 Mean Feature Impact Variables)

The number of predictors were kept the same as the number of predictors in DR Reduced Features to see which technique achieved better performance. The steps for identifying these variables were as follows:

* 1. All the cross-validation scores for all models were enabled on DataRobot GUI (by default, DataRobot calculates the cross-validation score for the top few models based on the 1st cross- validation fold score). The top 80% of the models based on cross validation AUC were selected. These selected models also had to have sufficient sample size of >64%. 9/13 models passed this screener.

Ft\_mean\_screenModels<-function(ID\_of\_project){ # Extract models

models\_listdetails<-ListModels(ID\_of\_project) #extract models (df format)

models\_df<-as.data.frame(models\_listdetails, simple = F) ## top 80% cv AUC

top80AUC<-quantile(models\_df$`Weighted AUC.crossValidation`, probs = seq(0,1,.1), na.rm = T)[["20%"]]

# remove small sample size % models

models\_listdetails\_big <- Filter(function(m) m$metrics$`Weighted AUC`$crossValidation>top80AUC & m$samplePct >= 64 , models\_listdetails

}

Ft\_mean\_screened<-Ft\_mean\_screenModels(project\_ID)

as.data.frame(Ft\_mean\_screened, simple = F) %>% select(expandedModel, featurelistName, `Weighted AUC.crossValidation`) %>% arrange(-`Weighted AUC.crossValidation`) DT::datatable(rownames = F, options = list(searchHighlight

= TRUE, paging= T))

* 1. The unnormalized feature impact score for all the variables in these models were averaged out.

Variables in the top 32 mean feature impact score were used to create a new set of Feature List.

Ft\_mean\_var<-function(Ft\_mean\_screened\_results, ID\_of\_project){ # feature impact of selected models

all\_impact<- NULL

for(i in 1:length(Ft\_mean\_screened\_results)) {

featureImpact <- GetFeatureImpact(Ft\_mean\_screened\_results[[i]]) featureImpact$modelname <- Ft\_mean\_screened\_results[[i]]$modelType all\_impact <- rbind(all\_impact,featureImpact)

}

#mean impact

mean\_impact<-all\_impact %>% group\_by(featureName) %>% summarise(avg\_impact = mean(impactUnnormalized), .groups="drop")

#top X mean feature impact scores ## number of predictors for DR

no\_of\_DRft<-ListFeaturelists(ID\_of\_project) %>% as.data.frame() %>% filter(str\_detect(name, "DR Reduced")) %>% nrow()-1 # minus 1 as one of the variables is the target

##top X mean feature impact scores

top\_mean\_names<-mean\_impact %>% slice\_max(order\_by = avg\_impact, n=no\_of\_DRft )

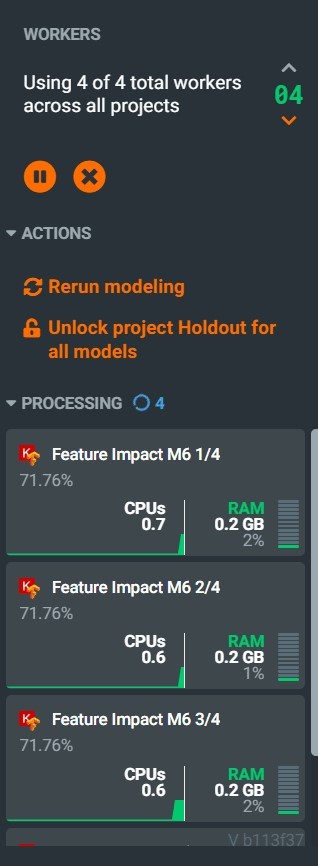
%>% pull(featureName)

output<-list(all\_impact, top\_mean\_names) return (output)

}

Ft\_mean\_varOutput<- Ft\_mean\_var(Ft\_mean\_screened, project\_ID)

The calculation can be seen on DataRobot’s GUI.



The top 32 variables based on mean feature impact score and their respective feature impact score for each model is plotted here.

Ft\_mean\_varOutput[[1]]%>%

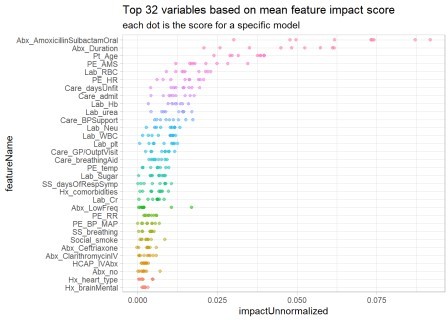
filter(featureName %in% Ft\_mean\_varOutput[[2]]) %>%

mutate(featureName= fct\_reorder(featureName, impactUnnormalized, .fun =mean))

%>%

ggplot(aes(featureName, impactUnnormalized, colour=featureName))+ geom\_point(alpha=.5) + coord\_flip() + theme(legend.position="none") + labs(title

= "Top 32 variables based on mean feature impact score", subtitle = "each dot is the score for a specific model")



New Feature List using the top 32 mean feature impact variables was created.

Feature\_id\_percent\_rank = CreateFeaturelist(project\_ID,

listName= "mean ft impact" , featureNames=

Ft\_mean\_varOutput[[2]][1:length(Ft\_mean\_varOutput[[2]])])$featurelistId ## [1] "Featurelist mean ft impact created"

## DR Features vs Top 32 mean Feature Impact variables

Both Feature Lists had these predictors in common

intersect(Ft\_mean\_varOutput[[2]], Ft\_DR$features)

|  |  |  |  |
| --- | --- | --- | --- |
| ## | [1] | "Abx\_AmoxicillinSulbactamOral" | "Abx\_Duration" |
| ## | [3] | "Pt\_Age" | "PE\_AMS" |
| ## | [5] | "Lab\_RBC" | "PE\_HR" |
| ## | [7] | "Care\_daysUnfit" | "Care\_admit" |
| ## | [9] | "Lab\_Hb" | "Lab\_urea" |
| ## | [11] | "Care\_BPSupport" | "Lab\_Neu" |
| ## | [13] | "Lab\_WBC" | "Lab\_plt" |
| ## | [15] | "Care\_GP/OutptVisit" | "Care\_breathingAid" |
| ## | [17] | "PE\_temp" | "Lab\_Sugar" |
| ## | [19] | "SS\_daysOfRespSymp" | "Hx\_comorbidities" |
| ## | [21] | "Lab\_Cr" | "Abx\_LowFreq" |
| ## | [23] | "PE\_RR" | "PE\_BP\_MAP" |
| ## | [25] | "SS\_breathing" | "Social\_smoke" |
| ## | [27] | "Abx\_Ceftriaxone" | "Abx\_ClarithromycinIV" |
| ## | [29] | "HCAP\_IVAbx" | "Abx\_no" |
| ## | [31] | "Hx\_brainMental" |  |

The only difference was that the top 32 mean Feature Impact variables contained medical history on the type of heart disease the patient had.

setdiff(Ft\_mean\_varOutput[[2]], Ft\_DR$features) ## [1] "Hx\_heart\_type"

## Running 2nd Quick Mode

Another round of Quick Mode was ran with both of the new Feature Lists created with advanced feature selection. Again, cross validation scores for all models was calculated.

8 additional models were generated with DR Reduced features (the other 3 was previously generated in the 1st round of Quick Mode). 10 additional models were generated with the Top 32 mean Feature Impact score variables.

#Get all the models from 1st and 2nd Quick Mode

List\_Model322<-ListModels(project\_ID) %>% as.data.frame(simple = F) List\_Model322 %>% count(featurelistName)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ##  ## ## | # | A tibble: 4 x 2 featurelistName |  | n |
| ## | 1 | DR Reduced Features | M8 | 11 |
| ## | 2 | L1 |  | 9 |
| ## | 3 | mean ft impact |  | 10 |

## 4 Multiple featurelists 1