KDD Cup 2009 Customer relationship prediction

Project Team Members:

Audrey E.

Frederic M.

John D.

Data Analysis & Visualisation

A. Initial exploration

- Reasonably clean data (correct format) but large number of missing data in a large number of variables
- Could there be a structure behind the pattern of missing values?

Predictor	# NA	Predictor	# NA	Predictor	# NA	Predictor	# NA
V175	32884	V41	32558	V202	3294	V21	4765
V207	32884	V65	32558	V206	3294	V121	4765
V26	32643	V104	32558	V219	3294	V197	4765
V100	32643	V158	32558	V222	3294		
V173	32643	V194	32558	V230	3294		

Table 1: Extract from "N/A Count" grouping table

- Built a data dictionary of the training data
  - We could not decode the data at the end… ☹

VAR	TYPE	#LEVELS	MIN	MAX	NOTES
V101	Numeric (continuous)	6	-267.52	12286.72	Possibly monetary, mostly 1 and 2dp - some up to 5 though.
V102	Numeric (discrete)	4	0	54	Multiples of 18
V103	Numeric (discrete)	131	0	6578865	Multiples of 15
V104	Numeric (discrete)	84	0	16784	Multiples of 16
V105	Numeric (discrete)	302	0	15235500	Multiples of 14
V106	NA				
V107	Numeric (discrete)	16	0	720	Multiples of 36
V108	Numeric (discrete)	10	0	160	Multiples of 16
V109	NA				
V110	Numeric (continuous)		0	3515.52	Possibly monetary
V111	Numeric (continuous)		0	14	Large precision (7dp), small values.
V112	Character	6			

Table 2: Extract from Data Dictionary

Data Analysis & Visualisation

B. Data plots

- Bar charts plots and histograms of each variable as well as plots of the variable distribution in the training
  => (loose) identification of each variable as belonging to one of six types:
  - Character variables which are likely to be factors.
  - Character variables which are unlikely to be factors - due to the number of unique values.
  - Discrete numeric variables, which are likely to be factors.
  - Discrete numeric variables, which unlikely to be factors - due to the number of unique values.
  - Discrete numeric variables which are possibly factors, but more likely e.g. time - measured in discrete units.
  - Continuous numeric variables.

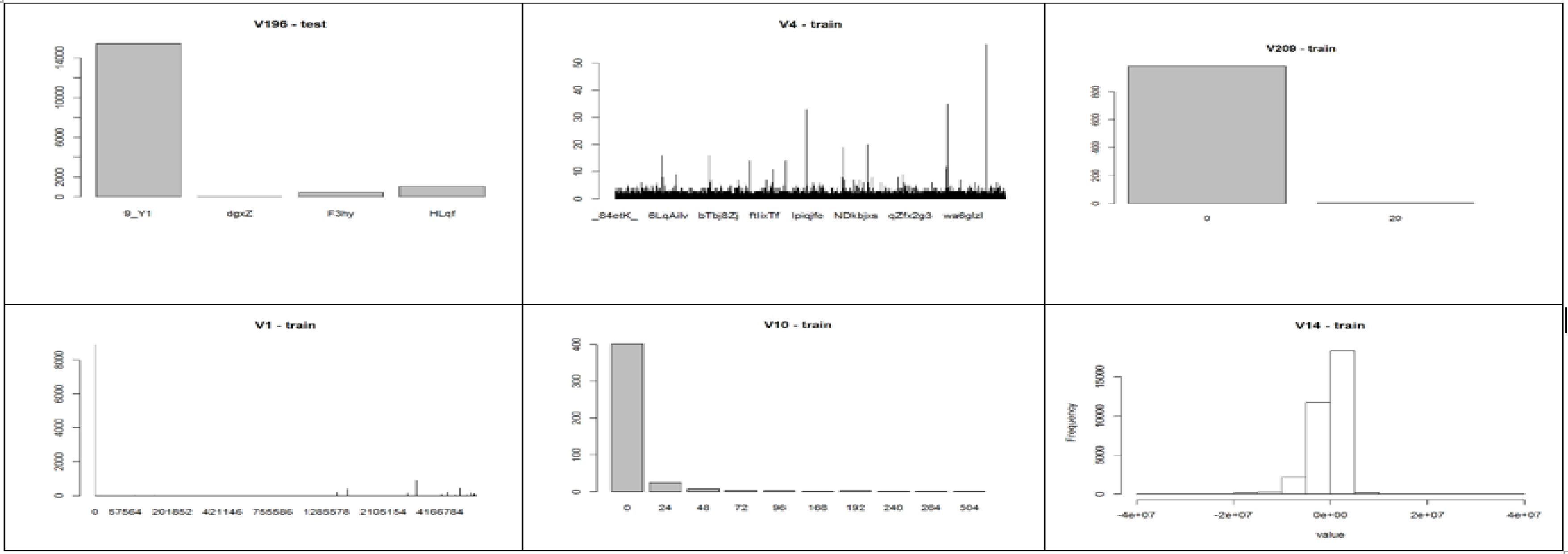


Table 3: Examples of variable types matching descriptions above (left to right, top to bottom)

Data Analysis & Visualisation

B. Data plots (cnt'd)

- Comparing Training Vs Test data to establish the training sample representativeness
  - =>Mostly ranges and distributions were similar
  - => Freq. smaller in the test dataset (but still in proportion of the test dataset)

- Plots of the distribution of each variable across those observations where for e.g. appetency was +1 and appetency was -1
  - - Hope of identifying variables which seems important for a given classification

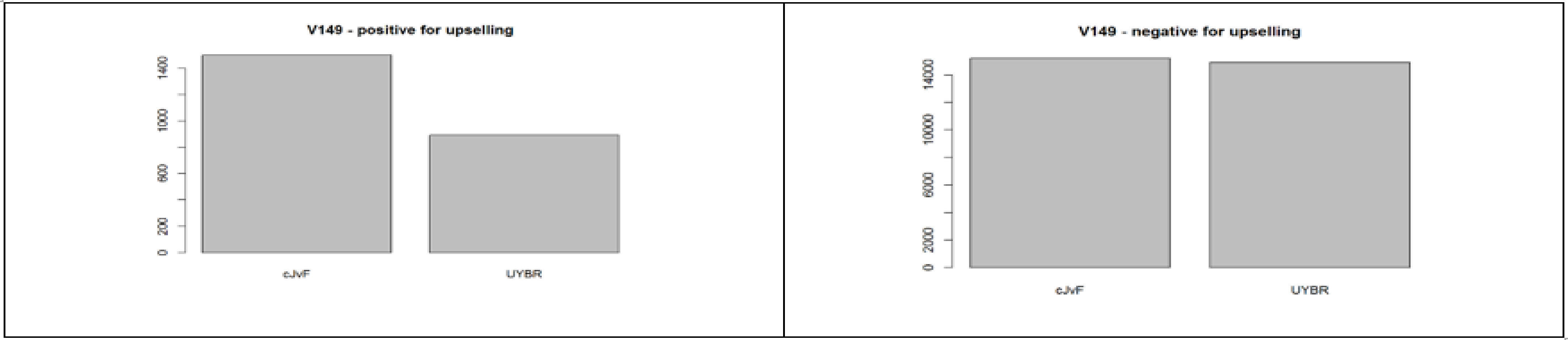


Table 4: Plot of positive vs negative responses for upselling, for V149. The differences in distribution over the levels for this variable suggest that it *may* be useful in predicting upselling.

=> But no clear cut ☹

Useful exercise in "getting to know the data". ☺

Data Pre-processsing

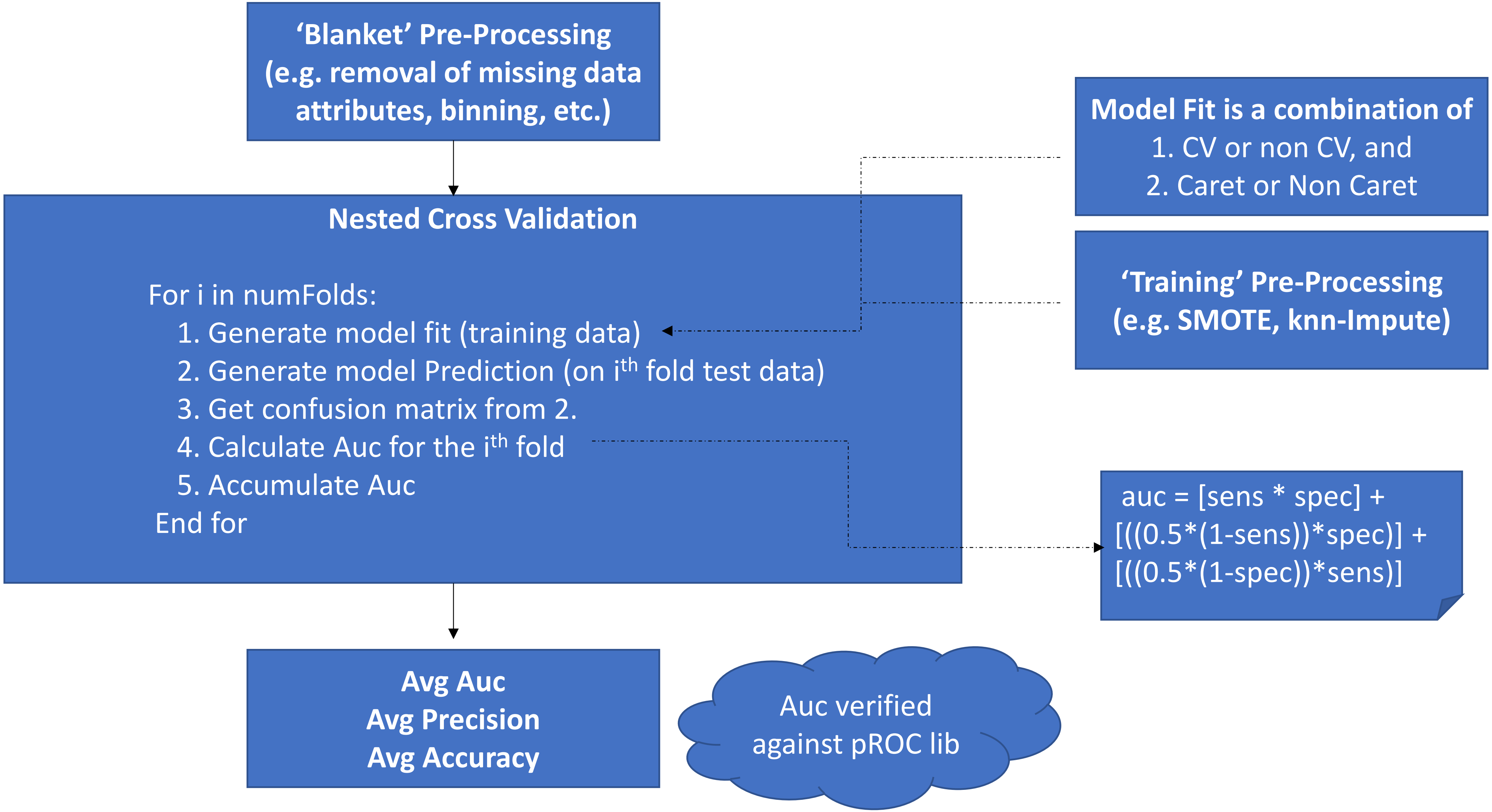
- ***Data conversion to factor***
  - Convert non numerical data to factors (absolutely required some of the models e.g. LDA)
- ***Missing Data (create bias)***
  - Drop any columns containing for than 50% of missing data
  - Knn-Impute missing numerical data
  - Convert NA to a string => to serve as a level.
- ***Correlated predictors*** (high multicollinearity increases variance/make prediction sensitive to minor change in model)
  - Deletion of correlated predictors (> 90% of correlation)
- ***Linear Dependencies***
  - Use the *findLinearCombos()* R function to find and remove linear dependencies recursively
- ***Category level aggregation*** (too many causes models to break)
  - Keep to 10 levels and bin the rest into BIN level.
  - Create Replacement levels based on level appearance frequency

Category level range	*Label name*
1-10	LEV_1_10
11-25	LEV_11_25
26-50	LEV_26_50
51-100	LEV_51_100
101-150	LEV_101_150
151-250	LEV_151_250
251-500	LEV_251_500
501-750	LEV_501_750
751-1000	LEV_751_1000
1001-3000	LEV_1001_3000
3001-5000	LEV_3001_5000
5001-1000000	LEV_5001_1000000

Table 2bis - Level replacement ranges

- ***SMOTE***
  - To reduce class unbalancing

Methodology



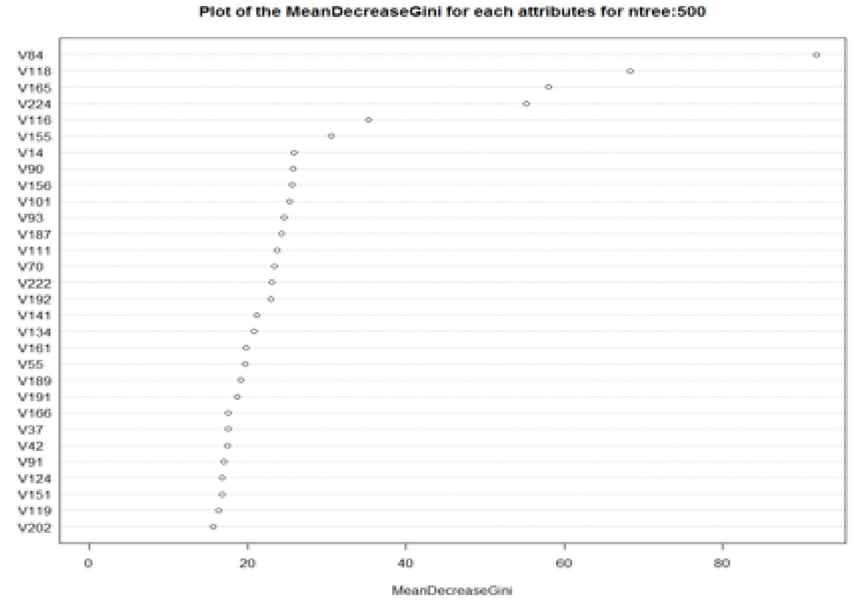
Model training/optimisation and testing (cnt'd)

- **LDA**
  - Pre-processing

Scenario 1	*Scenario 2*
convert_to_factors drop_na_cols remove_correlated_predictors convert_NAs_to_level remove_linear_dependencies bin_negative_levels_appetency keep_top_10_levels impute_data	convert_to_factors drop_na_cols remove_correlated_predictors convert_NAs_to_level remove_linear_dependencies create_replacement_columns impute_data

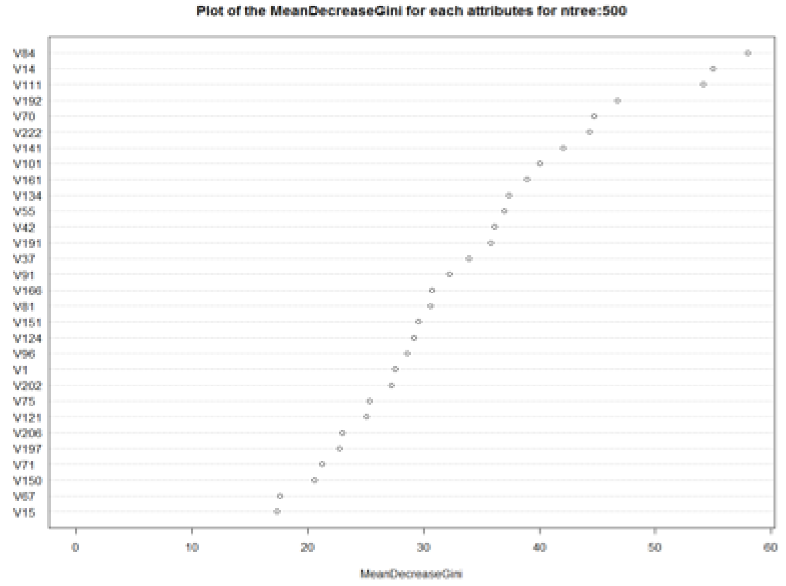
  - Feature Selection: PCA ☹ But Random Forest ☺ + Backward selection

Scenario 1



Appetency

Scenario 2

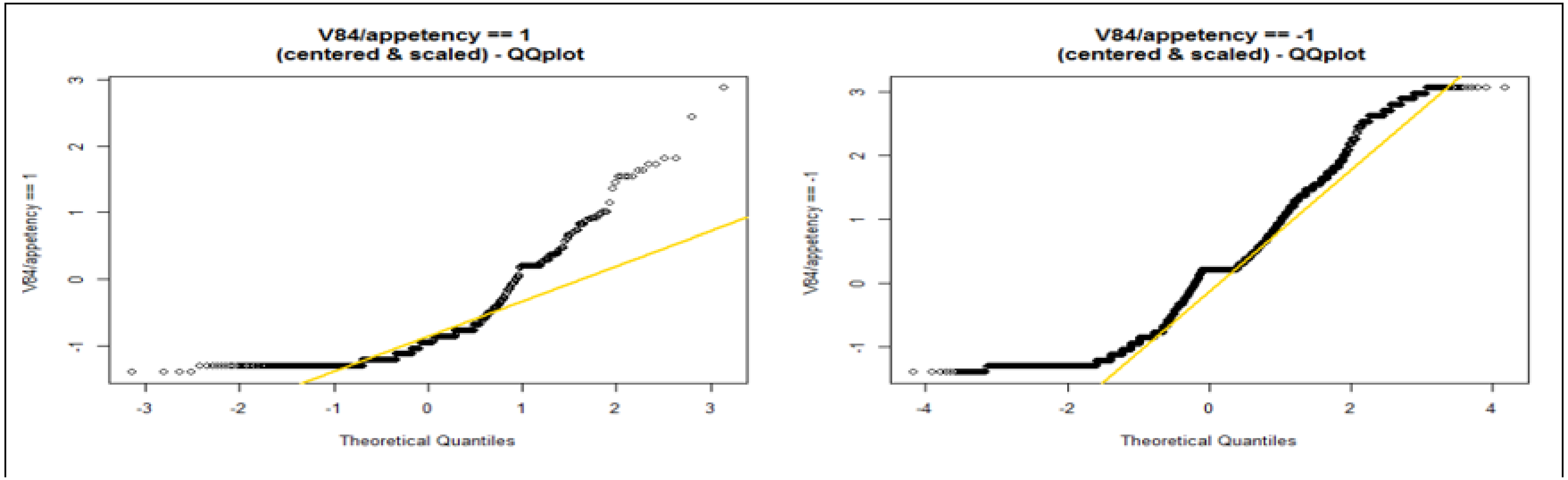


Appetency

Plot of the MeanDecreaseGini for each attributes for ntree:500

Model training/optimisation and testing (cnt’d)

- **LDA(Ctn’d)**
  - Model assumptions => explanatory variables to follow a normal distribution (H0).
    - The Kolmogorov Smirnov normality test + QQPlots => H0 is reject in all cases! ☹



- Results

Scenario 1	*Scenario 2*
Appetency: 91.26% Churn: 71.73% Upselling: 64.37%	Appetency: 71.54% Churn: 52.39% Upselling: 66.63%
Time to completion < 30mins	Time to completion approx. 1h

Model training/optimisation and testing (cnt’d)

- **SVM**
  - Pre-processing: from Scenario 1 only
  - Feature selection: from Scenario 1
  - Classifier choice: used a lot in literature / compatible with numerical and categorical values
  - Assumption: no specific assumption relating to the data distribution
  - Results:

Regime 1	***Regime 2***
Kernel = linear Cost: (0.0001,0.0005, 0.001, 0.1, 1 ,5 ,10 ,100) Gamma: not supported in the linear case	Kernel = radial Cost: (0.0001,0.0005, 0.001, 0.1, 1 ,5 ,10 ,100) Gamma: 0.0001,0.0005, 0.001, 0.1, 0.5, 0.7, 1)

Regime 1	***Regime 2***
Appetency: 50.0% Churn: 62.5% Upselling: 37.5%	Appetency: 50.0% Churn: 62.5% Upselling: 62.5%
Time to completion approx. 2 days	Time to completion approx. 2 days

Model training/optimisation and testing (cnt'd)

- **"NA Groups" Model**
  - Motivation:
    - Number of variables could be grouped together by count of missing values
    - Potentially 23 groupings where more than one variable had the same number of missing values
    - This pattern gave rise to the "tables within a table" approach
  - Procedure:
    - Basic idea was to fit a model to each group of variables with the same number of missing values, and then somehow combine the resulting predictions from each of these into an overall classification.

A	B	C	D	E	F	G	H
		4			0		0
284076			1NjhLsz			taul	
4595634						CuXi4je	
403884			Pzu_i9p			taul	
smXZ	112			16		smXZ	
	1088			144			
		4			0		0

A	G
284076	taul
4595634	CuXi4je
403884	taul
smXZ	smXZ

B	E
112	16
1088	144

- Generate groups…
- Assume same amount of missing values

C	F	H
4	0	0
4	0	0

Model training/optimisation and testing (cnt'd)

- **"NA Groups" Model (ctn'd)**
  - How to turn this "ensemble" of predictors into a single overall prediction?
  - A number of different strategies were attempted:
    - If any one of the nested models votes +1, then the overall model predicts +1
    - As above, but with a threshold of e.g. 20% positive votes to result in a +1 overall
    - As above, but with an additional threshold for a model to be included in the ensemble, based on its AUC score on the training data
    - A model which treats each nested predictor as a variable in a second, linear model – and then uses least squares to find the coefficients with which to weight their subsequent predictions.

  => Using manual thresholds produced decent results but risk of overfitting

  - Final approach: rather than manually extracting the weights from the uber-predictor, and using these during classification, we could just use the predictions of the uber-predictor.

model1	model2	model3	model4	...	model23	appetency
-1	0	+1	-1		-1	-1
+1	+1	0	-1		+1	+1
-1	+1	0	0		-1	+1

Table 5: Example matrix generated from predictions of individual models, vs the target label. Zeros in the matrix represent a predictor which cannot be used due to missing values in a row's data.

Model training/optimisation and testing (cnt’d)

- **“NA Groups” Model (Ctn’d)**
  - Classifier choice:
    - RF
    - Disparity of the data/types + decision trees are more aligned with a customer’s thought processes and behaviours than a parametric or mathematical model.
    - Other models were tried e.g. LDA, RDA, adaBoost, SVM but with mixed results ☹
  - Feature Selection: it was built-in the model
  - Pre-processing:
    - Knn-Impute numerical data column that did not group with any others.
    - “NA” level were created for missing character factors.
    - Any level in the training data not present in the test data were aggregated to BIN.
    - Any level which could never result in +1 were binned into ALL_NEGATIVE level.
    - Any factor with 30 or less levels were kept, all other placed into the BIN level.
    - SMOTE was applied to the training folds and near zero variance columns were removed.
  - Assumption: no specific assumption regarding the shape of the data
  - Results:

Results

Appetency:	91%
Churn:	69%
Upselling:	78%

30mins for Appetency
8h for the rest

  - Limitations: could be presence of overfitting due to the complexity of the model + slow to train

Model training/optimisation and testing (cnt'd)

- **"Basic RF" Model**
  - Derived from the observation of running the "NA Groups" model
    => when weighting each nested model in the ensemble,
    most of the weights appeared to be loading on the first model
  - Feature selection: No
  - Pre-processing: same as the "NA Groups" model
  - Assumption: none
  - Results:

Results

Appetency:	92%
Churn:	65%
Upselling:	71%

1h run to completion

Model training/optimisation and testing (cnt'd)

- **GBM**
  - **Pre-processing:**
    - Discrete numerical variables:
      - Missing values filled with "NA"
      - Actual values binned depending on what percentile they were in
    - Continuous variables:
      - Any columns with more than 40% of NA were discarded.
      - Remaining missing data were imputed

    - **Results:**
      - Appetency: NA
      - Churn: 51% AUC
      - Upselling: 71.41% AUC

Conclusion

Model Name	Appetency (Avg AUC)	Churn (Avg AUC)	Upselling (Avg AUC)
LDA Scenario 1	91.26%	71.73%	64.37%
LDA Scenario 2	71.54%	52.39%	66.63%
SVM Regime 1	50.0%	62.5%	37.5%
SVM Regime 2	50%	62.5%	62.5%
NA Groups	91%	69%	78%
Basic RF	92%	65%	71%
GBM	58%	51%	71%

List of retained models per target label.

Q&A