# ML Coursework 2
# Prediction of Churn, Appetency and Upselling from Orange Customer Data

Team Taka

Chris Journeay – Fabio Caputo – Imanol Belausteguigoitia

# Agenda

- Overview
- Data Exploration
- Data Pre-processing
- Model Training
- Churn Outcomes & Selected Model
- Appetency Outcomes & Selected Model
- Upselling Outcomes & Selected Model
- Total Score

# Overview

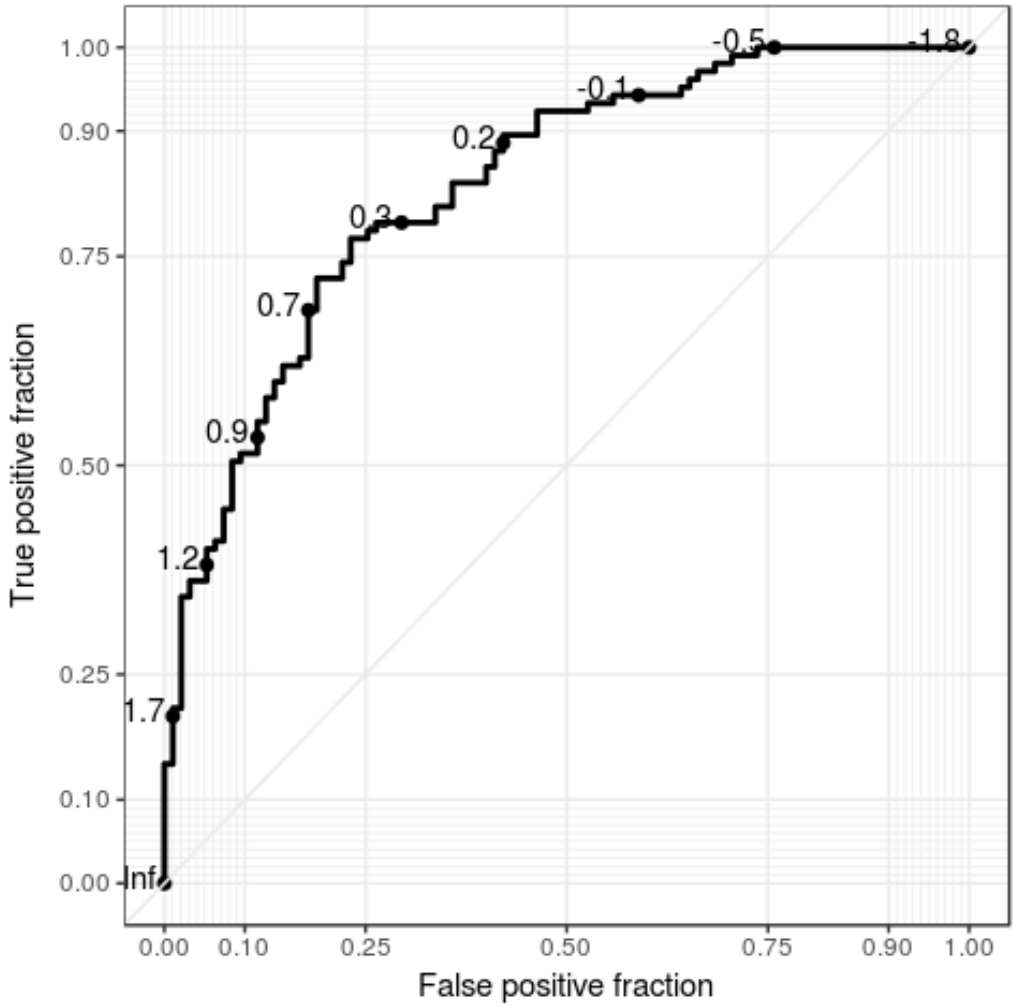
- The challenge for this coursework was to build models to predict three potential customer outcomes from a sample set of data from the CRM of a large mobile telecom provider.
  - Churn – This is the outcome where a customer ceases their relationship with the provider
  - Appetency – This is the outcome where a customer shows a propensity to buy a product or service
  - Upselling – This the is outcome where a customer purchases additional services over and above their current spending
- These outcomes are binary in nature
- Build models for classification of customers

# Overview
# Source Data

- The data, provided by French mobile phone provider Orange, consists of 3 files.
  1. A training file with 33,001 observations of 230 variables.
  2. A training targets file with 33,001 observations with 3 variables for the three outcomes churn, appetency and upselling. Values are 1 or -1 representing positive and negative outcomes respectively.
  3. A test data file with 16,999 observations  of 230 variables (~50% of the training data)

# Overview
# Measurement of success

- The results will be evaluated with the so-called Area Under Curve (AUC). The AUC is calculated using the trapezoid method.
- The trapezoid is defined by the following cardinal points on the AUC curve:
  - (0,1), (true negatives/(true negatives + false positives), true positives /(true positives +false negatives)), (1,0)



# Data Exploration

- Obfuscation – Unable to use business intelligence in our analysis
- Dimensions and Data Variance – Lots of variables with many factors. Factor levels ranged between near unitary to near unique
- Class Imbalance
- Missing Values - only 76 columns have completion rates above 10%
- Primary Challenges to address
  1. Dimensionality
  2. Class Imbalance

| Outcome | Observations | Positive Outcomes | Positive Outcome % |
|---|---|---|---|
| Churn | 33,001 | 2440 | 7.39% |
| Appetency | 33,001 | 584 | 1.77% |
| Upselling | 33,001 | 2431 | 7.36% |

# Preprocessing

- Addressing Dimensionality
- Eliminate columns using a missing data threshold
- We set the threshold to 70% as our completion rate
- 230 Columns -> 67 Columns

```
> print(sorted_by_non_empty$x) #prints out the percentages
  [1] 100.0000000 100.0000000 100.0000000 100.0000000 100.0000000 100.0000000 100.0000000
  [8] 100.0000000 100.0000000 100.0000000 100.0000000 100.0000000 100.0000000 100.0000000
 [15] 100.0000000 100.0000000 100.0000000 100.0000000 100.0000000 100.0000000  99.9939396
 [22]  99.7030393  99.7030393  99.7030393  99.2272961  98.6121633  98.6121633  96.0789067
 [29]  90.0184843  90.0184843  90.0184843  90.0184843  90.0184843  90.0184843  90.0184843
 [36]  90.0184843  90.0184843  90.0184843  90.0184843  90.0184843  90.0184843  90.0184843
 [43]  90.0184843  90.0184843  90.0184843  90.0184843  90.0184843  90.0184843  90.0154541
 [50]  89.6003151  89.6003151  88.9700312  88.9700312  88.9700312  88.9700312  88.9700312
 [57]  88.9700312  88.9427593  88.9427593  88.9427593  88.9427593  88.9427593  88.9427593
 [64]  85.5610436  85.5610436  85.5610436  72.2917487  55.2771128  55.2771128  49.0863913
 [71]  49.0863913  47.5834066  43.0380898  42.0805430  25.5204388  25.5174086   7.4937123
 [78]   3.0817248   3.0817248   3.0817248   3.0817248   3.0817248   3.0817248   3.0817248
 [85]   3.0817248   3.0817248   3.0817248   3.0817248   3.0817248   3.0817248   3.0817248
 [92]   3.0817248   3.0817248   3.0817248   3.0817248   3.0817248   3.0817248   3.0817248
 [99]   2.9938487   2.9938487   2.9938487   2.9938487   2.9938487   2.9938487   2.9938487
[106]   2.9938487   2.9938487   2.9938487   2.9938487   2.9938487   2.9938487   2.9938487
[113]   2.9938487   2.9938487   2.9938487   2.9938487   2.9938487   2.9938487   2.9938487
[120]   2.9938487   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076
[127]   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076
[134]   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076
[141]   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076   2.5484076
[148]   2.2847792   2.2847792   2.2847792   2.2847792   2.2847792   2.1181176   2.1181176
[155]   2.1181176   2.1181176   2.1181176   1.7120693   1.7120693   1.7120693   1.6878276
[162]   1.6878276   1.6878276   1.6666162   1.6666162   1.6666162   1.6666162   1.6666162
[169]   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346
[176]   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346
[183]   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346
[190]   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346   1.3575346   1.3423836
[197]   1.3423836   1.3423836   1.3423836   1.3423836   1.3211721   1.3211721   1.3211721
[204]   1.0848156   1.0848156   1.0848156   0.8575498   0.7060392   0.6908882   0.4908942
[211]   0.3545347   0.3545347   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
[218]   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
[225]   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
```

# Preprocessing Numerical Values

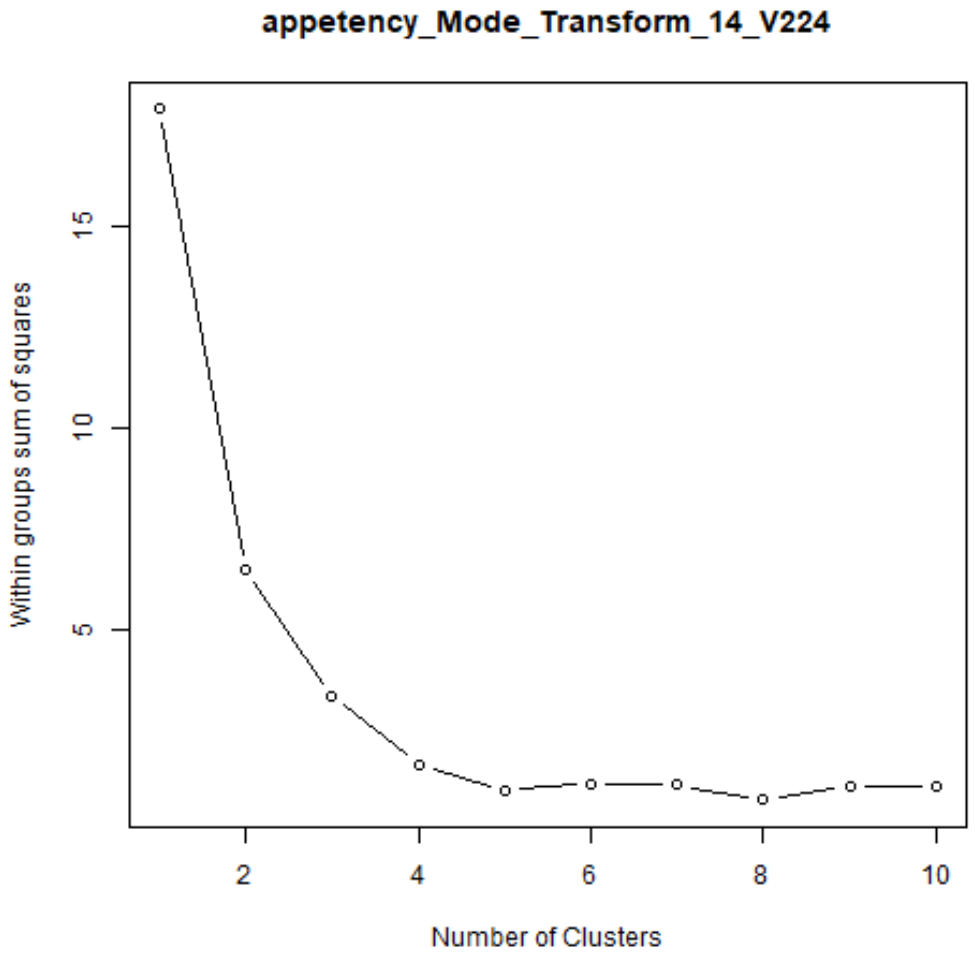
- Centred and scaled any numerical values
- Near-zero variance check
  - Used caret recommendations for nearzero variance
  - 67 Columns -> 57 Columns
- Impute missing numerical data tackle skewness.
  - Used the mlr package to impute the missing numbers for every numerical column. This was quite successful and eliminated any remaining NA values in the training set.
  - Remaining outliers are addressed by a spatial sign transformation just before training the model.
- Address skewness
  - Used caret Yeo-Johnson transformation to the numerical data to address any skewness to the data sets - - there were some variables with skewness > 1.5 and kurtosis > 9.

# Preprocessing Factors

- Impute the missing factor data with 2 different approaches
  - Approach 1: Replace NA with “None”
    - Assumes meaning in NA values – “missingness” has meaning
    - Converted any NA values to a “None” level. This could give meaning to “blanks” in the training set that could enrich any models
  - Approach 2: Replace NA with Mode Imputation
    - Assumes no meaning to NA values
    - Column mode applied to any NA values

# Preprocessing Factors

- Addressing the variance of the factor data
  - Several data columns had more than 100 levels with multiple instances of 2000+ different factor levels.
  - Looked to leverage the response rate to help drive more signal into the data
  - Approach 1: We binned the data using positive response rate quartiles
    - Derive Positive Response rates for each factor and use that distribution to assign a quartile value to replace the actual value
  - Approach 2: We binned the data using k-means clustering of the response rate
    - Positive and negative response rates for each factor level as inputs and then ran multiple k-means scenarios. We used the elbow method to determine the optimum number of neighbours using the within groups sum of squares metric



# Preprocessing

- Even with the above efforts to reduce dimensionality, our training sets still have 57 predictors.
- Used the Relief package in R on the four data sets to remove additional columns by computing the relief statistical test.
- We chose this relatively large alpha value (5%) in order to retain as much information as possible. Given the class imbalance for all outcomes, this seemed to be a sensible decision.

| Training Set | Churn | Appetency | Upselling |
|---|---|---|---|
| NA Replaced Factors binned by quartile | 41 | 45 | 37 |
| NA Replaced Factors binned by k-means | 38 | 45 | 45 |
| Mode Imputed Factors binned by quartile | 47 | 45 | 42 |
| Mode Imputed Factors binned by k-means | 40 | 35 | 41 |

# Training Sets Summary

- These multiple approaches left us with a 8 different training sets for use against the models
- All of the training sets reduced with Relief became unique for each outcome
- For each training set, there was also an associated Testing set with identical preprocessing

| Training Set | Factor Impute | Factor Bin Method | Reduced using Relief |
|---|---|---|---|
| Set 1 | NA -> None | K-Means | No |
| Set 2 | NA -> None | Response Quartile | No |
| Set 3 | Mode | K-Means | No |
| Set 4 | Mode | Response Quartile | No |
| Set 5 | NA -> None | K-Means | Yes |
| Set 6 | NA -> None | Response Quartile | Yes |
| Set 7 | Mode | K-Means | Yes |
| Set 8 | Mode | Response Quartile | Yes |

# Training Approach & Models

**Training Approach**

- Training was done using the caret package which enabled us to leverage the multiple training methods and tuning grid options.

**Training Models Selected**

- Bagged Trees
- Random Forest
- XGB
- Gradient Boosting Machines
- Support Vector Machines

SPATIAL SIGN TRANSFORMATION ADDRESSED OUTLIERS.

SMOTE SAMPLING TO ADDRESS SEVERE CLASS IMBALANCE

MODEL SELECTION FOCUSED ON MAXIMISING THE AUC

3 SETS OF REPEATED CROSS VALIDATION FOR ALL TRAINING.

PRINCIPLE COMPONENT ANALYSIS APPLIED WHERE APPROPRIATE.

THE BEST KAPPA VALUE WAS USED TO CHOOSE THE SENSITIVITY AND SPECIFICITY FOR CALCULATING THE AUC

# Training Methodology

# Churn Testing Outcomes and Model Selection

- The top 5 performing models are provided here
- All best outcomes have been obtained using the Training Set 4 and employing tree based algorithms and PCA.
- Use of principal components has been the key factor to improve the results on cross validation.
- Reducing the dimensionality of the data set based on the outcome of the Relief test has proven to deteriorate the model performance
- Tree based algorithms performed best with SVMs performing worse
- The highlighted model was selected for the test set

| Training Set | Model | AUC Trapezoid |
|---|---|---|
| Training Set 4 | xgb trees model | 0.928540528 |
| Training Set 4 | xgb trees model with PCs (85% VAR) as features | 0.935632969 |
| Training Set 4 | xgb trees model with additional PCs (85% VAR) as features | 0.99254326 |
| Training Set 4 | GBM | 0.917190346 |
| Training Set 4 | GBM model with PCs (85% VAR) as features | 0.930880009 |

# Appetency Testing Outcomes and Model Selection

- The top 5 performing models are provided here
- The Appetency Outcome provided a real challenge given the small positive outcomes.
- No training set emerged as a “best choice” in this instance suggesting that all pre-processing options removed useful information at some point.
- Less processed data performed better with reduction by Relief performing worse overall.
- Support Vector Machines and Bagged Trees models performed best overall for this outcome providing 4 out of the top 5 performing models.
- Bagged trees were selected as the chosen model as it was the best performing with the most complete test set.
- The highlighted model was selected for the test set

| Training Set | Model | AUC Trapezoid |
|---|---|---|
| Training Set 1 | Support Vector Machines | 0.9744103 |
| Training Set 2 | Bagged Trees with CART | 0.9693434 |
| Training Set 3 | Support Vector Machines | 0.9657741 |
| Training Set 4 | Bagged Trees with CART | 0.9656101 |
| Training Set 4 | Random Forest | 0.9654416 |

# Upselling Testing Outcomes and Model Selection

- The top 5 performing models are provided here
- All best outcomes have been obtained using the Training Set 4
- Random Forest gave the best results of the techniques used, followed by Gradient Boosting Machines and Bagged Trees.
- Reducing the dimensionality of the data set based on the outcome of the Relief test has proven to deteriorate the model performance

| Training Set | Model | AUC Trapezoid |
|---|---|---|
| Training Set 4 | Random Forest | 0.9475356 |
| Training Set 4 | Random Forest | 0.9471240 |
| Training Set 4 | GBM | 0.9395923 |
| Training Set 4 | GBM | 0.9367030 |
| Training Set 4 | Bagged Trees | 0.9313487 |