SECOM Case Study
following CRISP-DM Methodology

MPMD 2.2 Data Mining Techniques - Group 6

Catherine King

Lin Yi Hsuan

Pawin Poboon

Ender Yolagel

Agenda

CRISP-DM Process

1. Business Understanding
2. Data Understanding
3. Data Preparation & Preprocessing
4. Model Building & Evaluation
5. Model Deployment & Results
6. Retrospective CRISP-DM

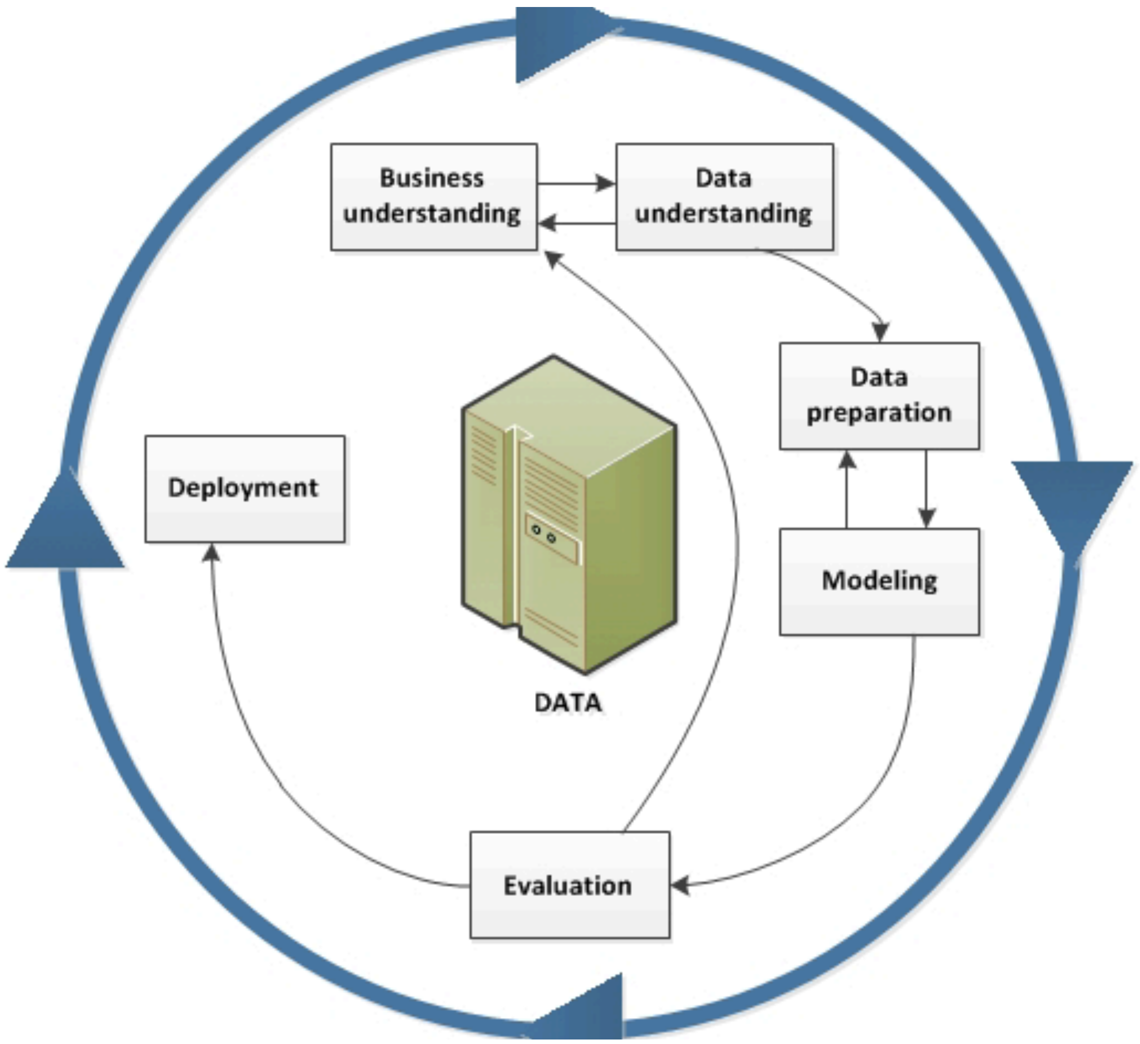
CRISP-DM Introduction

What

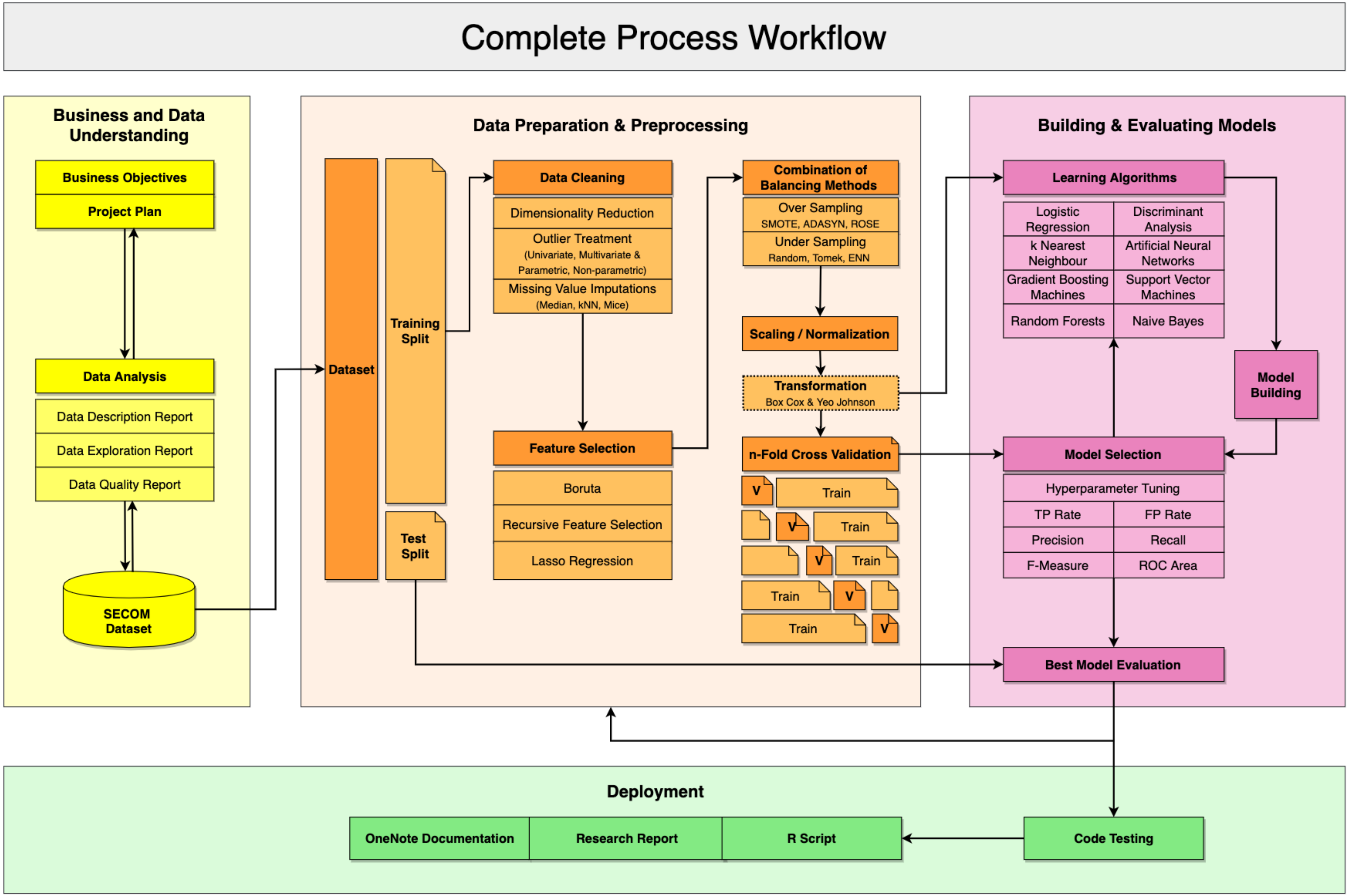
- 6 phases of a project
- Overview of data mining life cycle
- Flexible and easily customized, yet structured
- Iterative process

Why

- Organizes project streams, output, and documentation
- Guides users through data mining projects



Source: IBM SPSS Modeler CRISP-DM Guide


Complete Process Workflow
Business and Data Understanding
Business Objectives
Project Plan
Data Analysis
Data Description Report
Data Exploration Report
Data Quality Report
SECOM Dataset
Data Preparation & Preprocessing
Dataset
Training Split
Test Split
Data Cleaning
Dimensionality Reduction
Outlier Treatment (Univariate, Multivariate & Parametric, Non-parametric)
Missing Value Imputations (Median, kNN, Mice)
Feature Selection
Boruta
Recursive Feature Selection
Lasso Regression
Combination of Balancing Methods
Over Sampling SMOTE, ADASYN, ROSE
Under Sampling Random, Tomek, ENN
Scaling / Normalization
Transformation Box Cox & Yeo Johnson
n-Fold Cross Validation
V
Train
Building & Evaluating Models
Learning Algorithms
Logistic Regression
Discriminant Analysis
k Nearest Neighbour
Artificial Neural Networks
Gradient Boosting Machines
Support Vector Machines
Random Forests
Naive Bayes
Model Building
Model Selection
Hyperparameter Tuning
TP Rate
FP Rate
Precision
Recall
F-Measure
ROC Area
Best Model Evaluation
Deployment
OneNote Documentation
Research Report
R Script
Code Testing

1. Business Understanding

Business Success Criteria & Data Mining Goals

Business Background:

- **Semiconductor industry** is complex and known for sophisticated production processes with many steps
- **Default detection** during the production process plays an important role to smooth productivity, preventing breakdowns, and reducing related costs
- To predict defaults, **data from the sensors** on the production line must be collected and analyzed

Business Objective:

- To accurately **predict faulty wafers** on the production line, possibly before production is finished

Requirements, Assumptions, and Constraints:

- Complete a successful data mining project on the Secom dataset following the **CRISP-DM methodology**.
- **Well-structured document** must describe all processes, decisions, and relevant contents
- Limited **business insight and information** about 590 features

Data Mining Goals:

- **Prediction of defaults** based on the least amount of sensors/features using CRISP-DM
- Build **parsimonious model** with great exploratory power with **15-30 key features (sensors)**

2. Data Understanding

2.1 Data Collection & Description Report

2.1.1 Quantity

- Format
- Size

2.1.2 Quality

- Characteristics/attributes
- Effect on Data Mining Hypotheses

2.2 Exploring Data – EDA Report

2.2.1 Univariate Analysis

2.2.2 Multivariate Analysis

2.2.3 Target Feature

2.3 Data Quality

2.3.1 Outliers

2.3.2 Missing Values

2.3.3 Data Quality Report

2.1 Data Understanding: Data Collection Report

2.1.1 Quantity	2.1.2 Quality
Data comes in one .sav file. Feature names not given.	Beside time stamps, **all features are numeric**; target variable is binary.
590 features plus 3 (ID, timestamp, and class [0 = good, 1 = defective]) 1567 entries • with 1472 "good" (~93%) • 95 "bad" (~6%)	Data for each feature determines "good" or "bad" result. Up to **1429 missing values per feature**. Without more business insight, we assume **MAR**.

2.2 Data Understanding: EDA Report

2.2.1 Univariate Analysis

- No **duplicated** or **complete missing** rows
- 538 Features have **missing values**
- 28 features that have missing values **more than 50%**
- 116 features **with constant values**
- Shapiro Wilk test: 473 out of 474 features are **not normally distributed**

Top 5 Features that have largest skewness

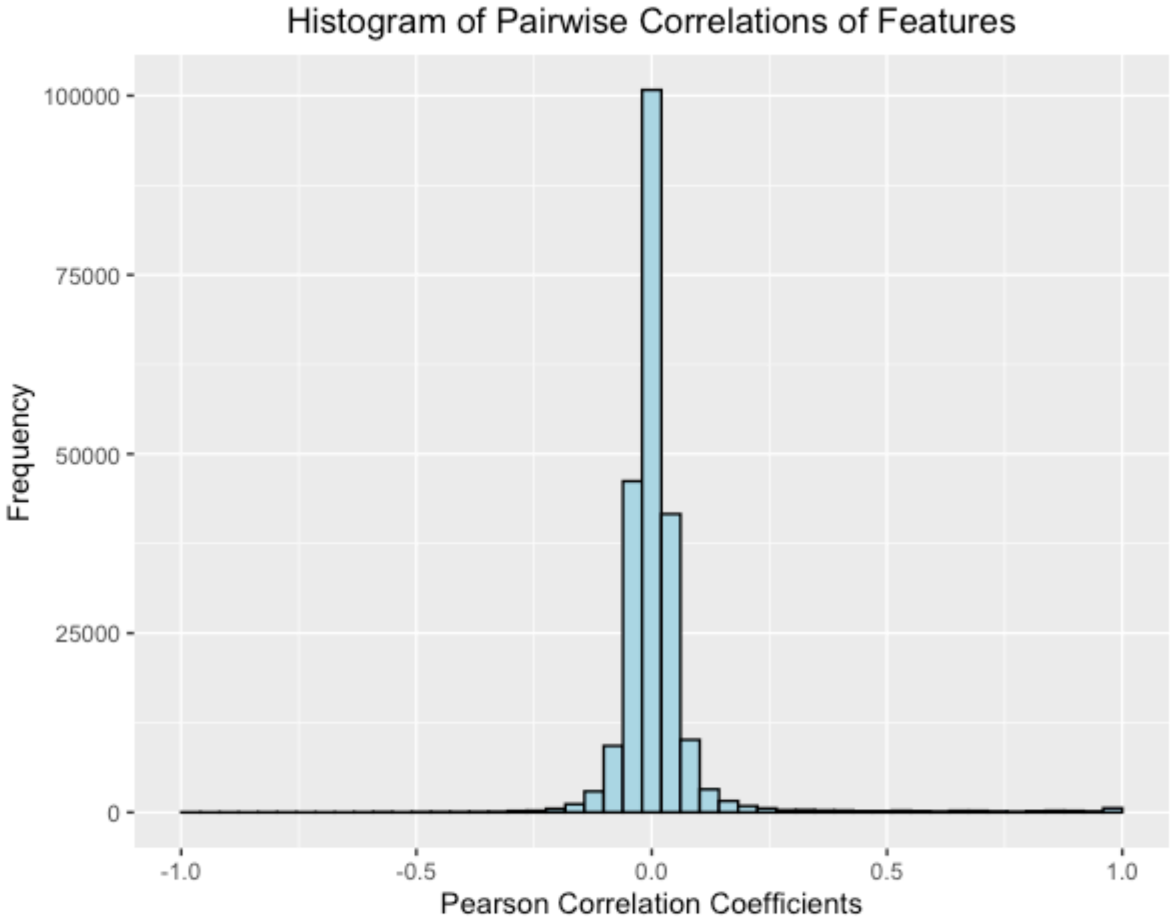
```
# A tibble: 20 x 8
   variable   skewness kurtosis  mean      sd    p25    p50   p75
   <chr>         <dbl>    <dbl> <dbl>   <dbl>  <dbl>  <dbl> <dbl>
 1 feature160   4.20     21.6     883.  983.     411   623    966
 2 feature162   2.23      6.77   4067. 4239.    1321  2614   5034
 3 feature297   2.19      6.05   1879. 1975.     603. 1202.  2341.
 4 feature023  -2.18     15.7    2699.  295.    2578  2664   2842.
 5 feature163   1.83      3.00   4797. 6554.     451  1784   6384
```

2.2 Data Understanding: EDA Report

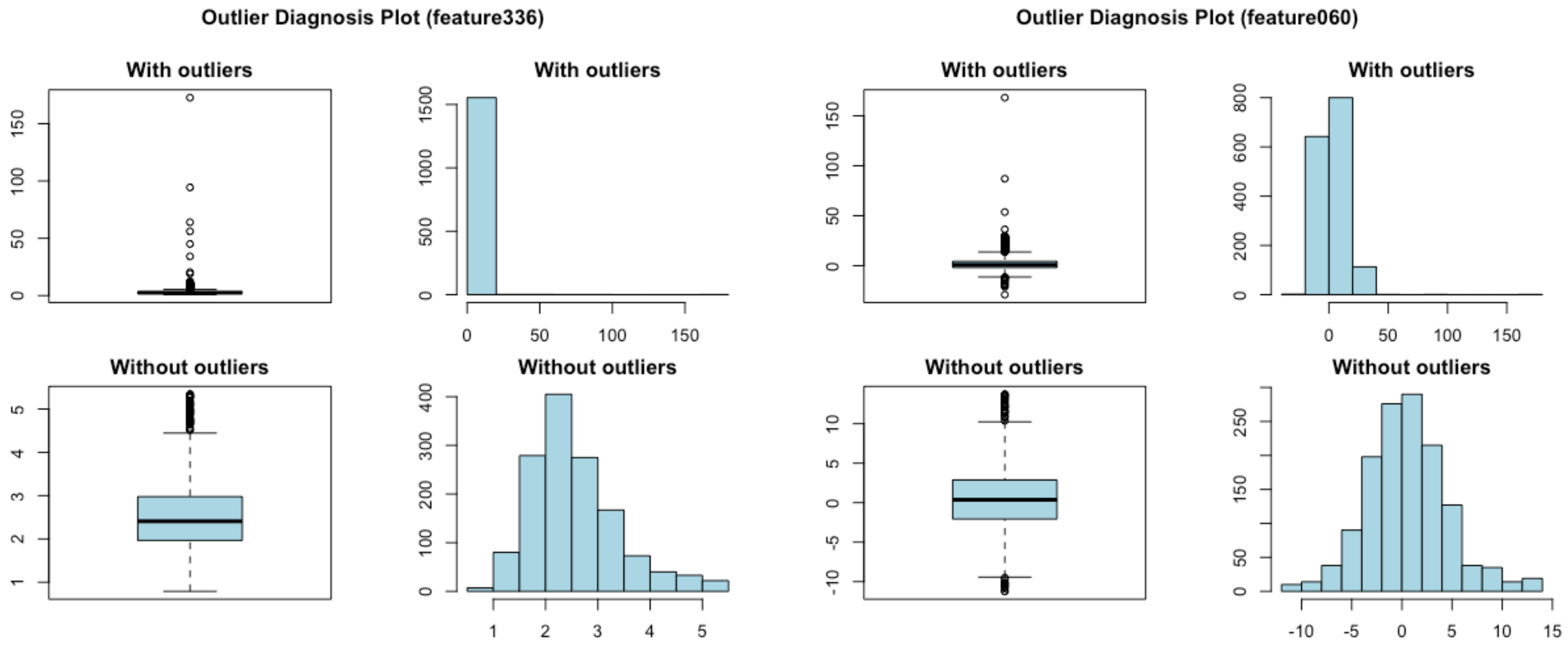
2.2.2 Multivariate Analysis

Pearson Correlations:

- Result in a histogram with **over 220,000 correlations**
- **48 correlations** of exactly +1 and 2 correlations of exactly -1
- Important as we move forward with **MICE**

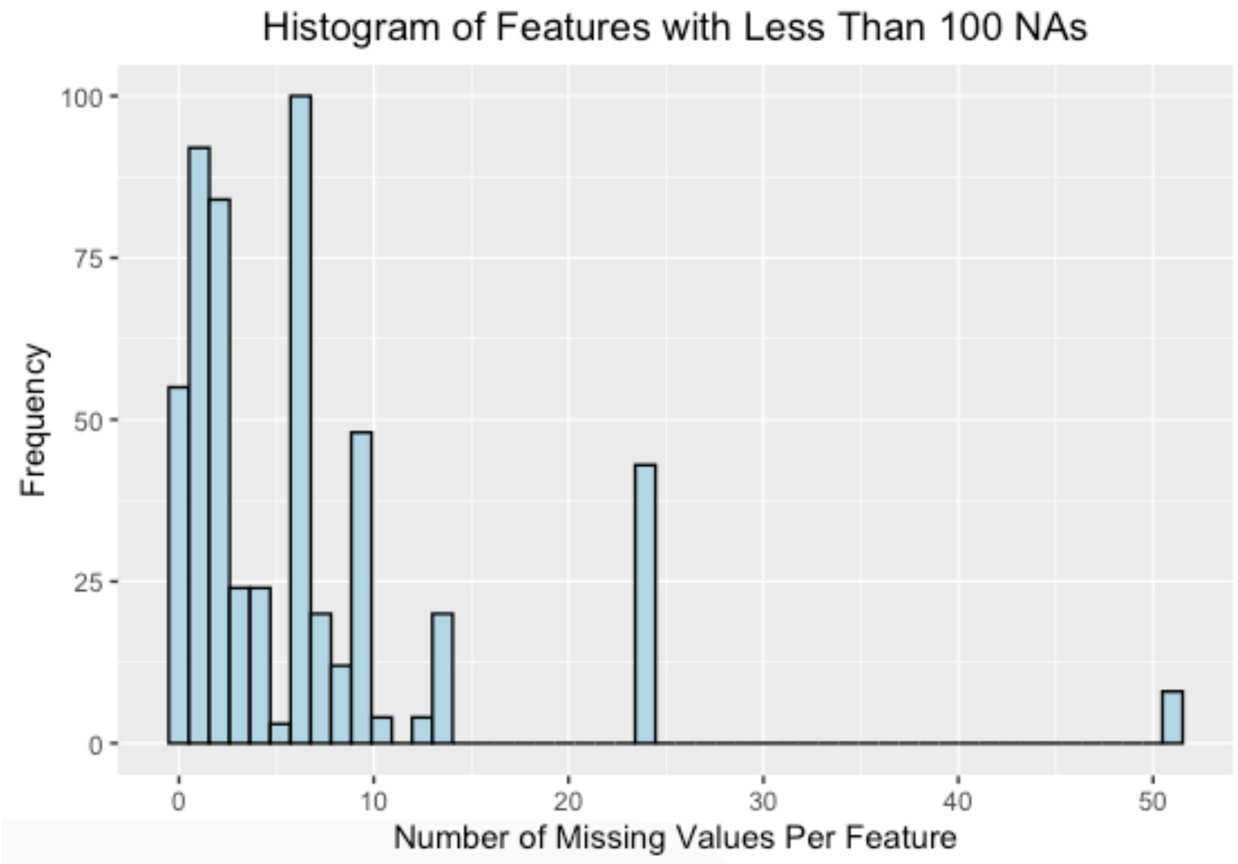


2. Data Understanding: Outliers



	variables	outliers_ratio	outliers_mean	with_mean	without_mean	rate
1	**feature060**	12.50798	19.796804082	2.960241474	0.5409113636	6.687564
2	**feature390**	12.06126	0.008788360	0.001338354	0.0003165457	6.566546
3	**feature524**	12.69943	2.909124623	0.453896426	0.0967396930	6.409226
4	**feature252**	12.82706	0.026911940	0.004284812	0.0009553441	6.280775
5	**feature130**	15.18826	-2.471529412	-0.554228306	-0.2085331061	4.459407

2. Data Understanding: Missing Values




```
# A tibble: 28 x 5
   variables  missing_count missing_percent unique_count unique_rate
   <chr>              <int>           <dbl>        <int>       <dbl>
 1 feature158          1429            91.2          129      0.0823
 2 feature159          1429            91.2          139      0.0887
 3 feature293          1429            91.2           93      0.0593
 4 feature294          1429            91.2          139      0.0887
 5 feature086          1341            85.6           98      0.0625
```

3. Data Preparation & Preprocessing

3.1 Splitting the train & test dataset

3.2 Data Cleaning

3.2.1. Dimensionality Reduction

3.2.2. Outlier Treatment

3.2.3. Missing Value Imputation

3.3 Feature Selection

3.4 Balancing Methods

3.5 Scaling/Normalization

3.6 Transformation

3.1 Data Preparation: Split the dataset
Random Stratified Sampling

80% Train — Train set 1254 observations across 593 features

20% Test — Test set 313 observations across 593 features

93.9% "successes" and 6.1% "failures" by **preserving the class proportion of target feature**

3.2. Data Cleaning

3.2.1. Dimensionality Reduction

- Remove features from dataset with 55% or more missing values - **reduce # features to 569**
- Remove features from dataset with 0% variance - **reduce # features to 453**

3.2.2 Outlier Treatment

- Replace all values outside 3s values (for each feature) **with NAs** – to be handled with all other missing values

3.2.3 Data Preparation: Missing Value imputation

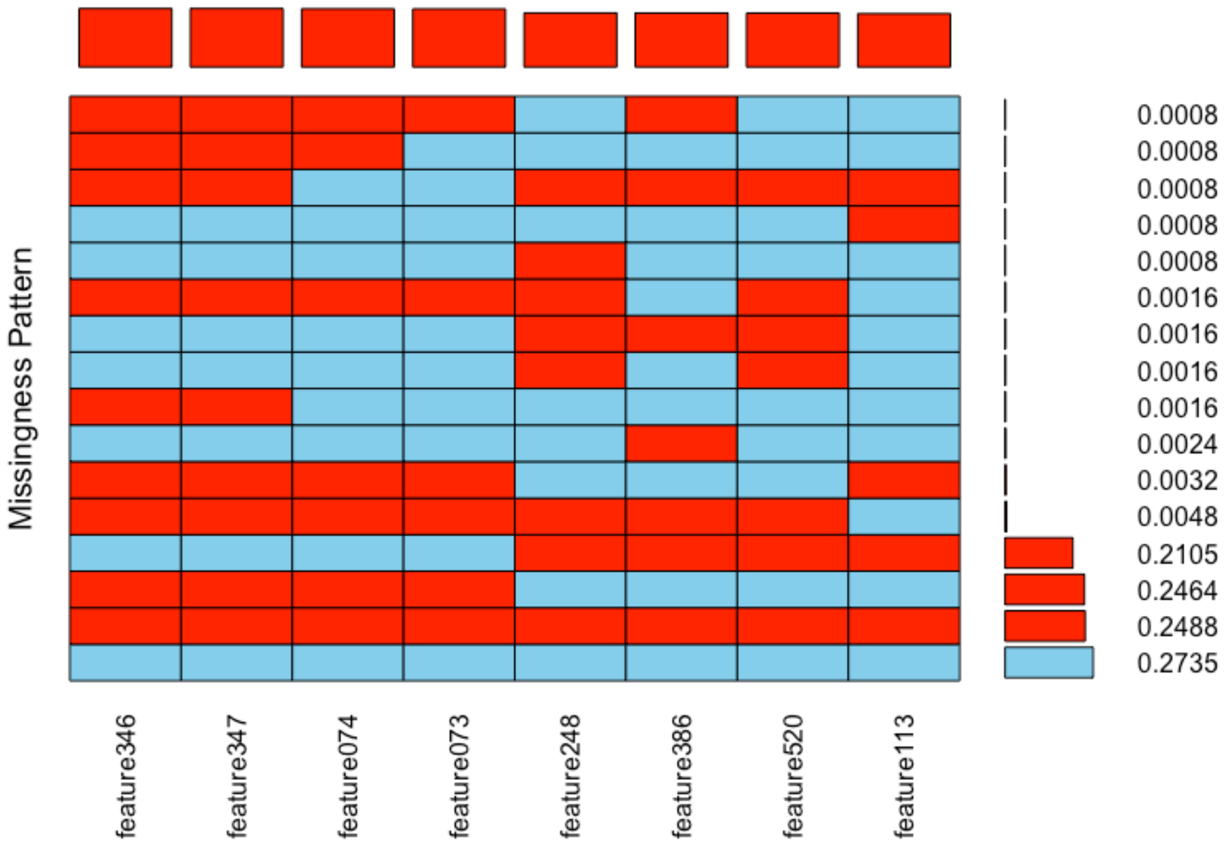
Missing Value Patterns

Missing values of all features:

- Nearly **25% of observations are missing** in all features together
- **Another near 25%** of observations are missing together in features 073, 074, 346, 347
- **Another 21%** of observations are also missing together in features 113, 248, 386, 520

Imputation methods to handle missing values:

- kNN
- MICE
- Random Forest



3.2.3 Data Preparation: Missing Value imputation

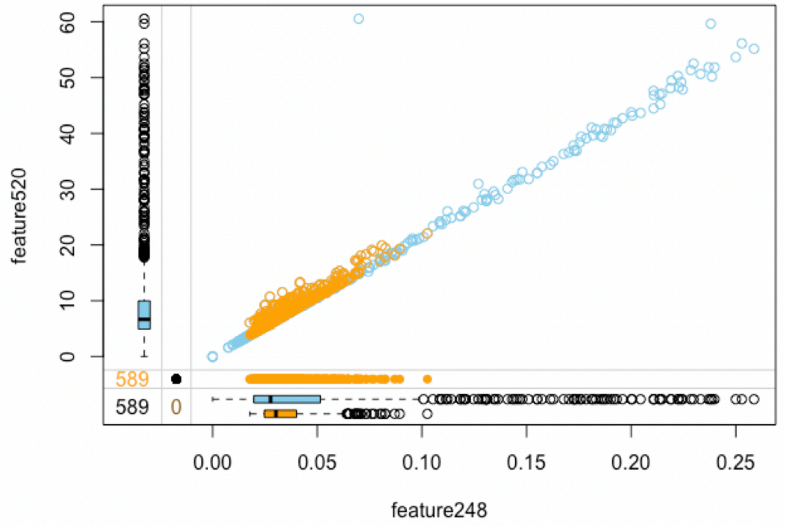
kNN Imputation

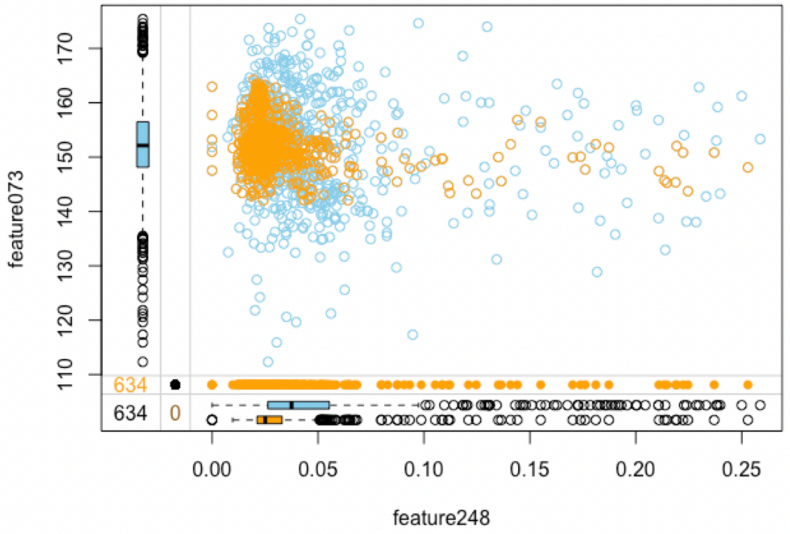
<u>kNN Imputation approach</u>:

- Non-parametric, unsupervised algorithm
- Match a datapoint with its **closest k neighbors** in a **multi-dimensional space**
- Work well with **a small number of input variables** but struggle when the number of inputs get very large
- Identify the "k" closest observations based on **Euclidean distance** and compute the **weighted average**

<u>Results from kNN</u>:

- Observations need to be **temporally scaled**
- **Tuning k parameters** from 1 to 21

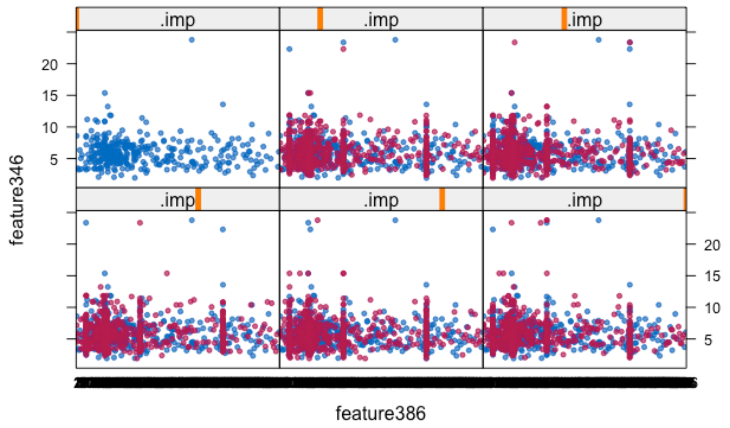


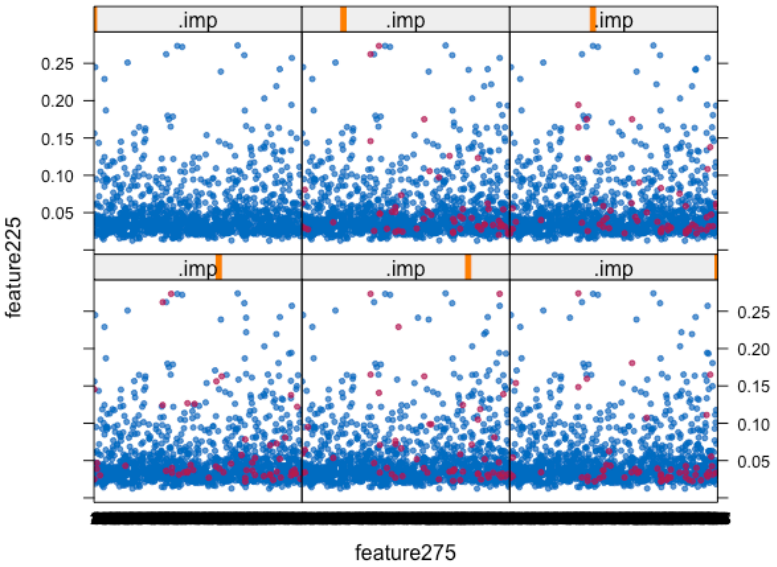


3.2.3 Data Preparation: Missing Value imputation

MICE Imputation

- Assumed missing values are **missing at random (MAR)**.
- **21 Predictors** on average used in for each imputation model, suggested by van Buuren as **15~25** predictors (2018) using **Spearman** correlations.
- Target feature is included as **covariate** in each imputation model
- **CART Method** seek predictors and cut points in the predictors that are used to split the sample.
- **Parameter uncertainty** is incorporated by fitting the tree on the bootstrapped sample.
- This method deals with **multicollinearity** and **skewed distributions**, and **nonlinear relations**.





3.2.3 Data Preparation: Methods comparison MICE vs kNN

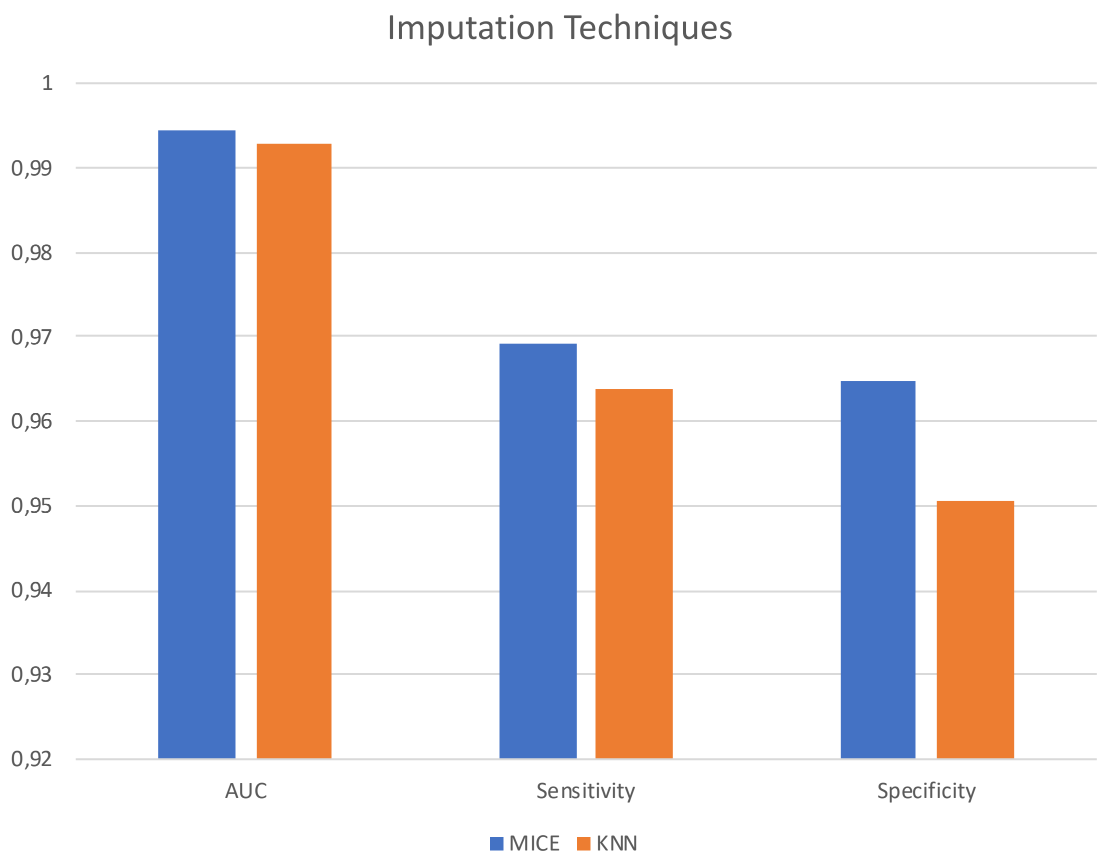
Comparison of imputation methods

MICE performs better:

- Compared by using **the best parameter** for both imputation methods
- MICE performs better in term of **AUC, sensitivity, precision.**
- AUC and Sensitivity are slightly different, while specificity of MICE is much higher
- MICE detects **collinear** and **constant** features, and does not impute them.

Decisions:

- **Apply MICE** as imputation method.
- Drop **20 collinear** and **6 constant** features that are detected by MICE.



3.3 Data Preparation: Feature Selection

Boruta (Feature Selection)

Boruta approach:

- Wrapper method built around the **random forest classification algorithm**
- Perform several random forest runs to obtain statistically significant division **between important and irrelevant attribute**

Results from Boruta:

- **13 important features** and 2 tentative features are identified
- **Best parameters** are maxRuns = 250, doTrace = 2
- Boruta does **NOT handle multicollinearity**, but MICE does that already (another reason why not choosing kNN)

maxRuns	Iterations	Important	Unimportant	Tentative	Duration (mins)
100 (default)	99	11	422	5	7.305315
101	100	11	422	5	6.600742
76	75	11	422	5	6.486325
150	149	12	422	4	6.993041
250	**249**	**13**	**422**	**2**	**9.19243**
300	299	13	422	2	8.32042
350	349	13	422	2	8.262173
500	499	13	422	2	10.80016

3.3 Data Preparation: Feature Selection

Boruta (Feature Selection)

<u>Boruta vs without Boruta</u>:

- **Improve overall criteria**, especially sensitivity which is highly important in SECOM case
- Sensitivity increases **from 0.76 to 0.96** with feature selection (Boruta)

<u>Benefits from Boruta</u>:

- Does **not compromise the performance** of the model and might lead to a **more parsimonious and interpretable model**
- Some models can be crippled by **predictors with degenerate distributions**
- Significant **improvement in model performance** and/or stability without the problematic features

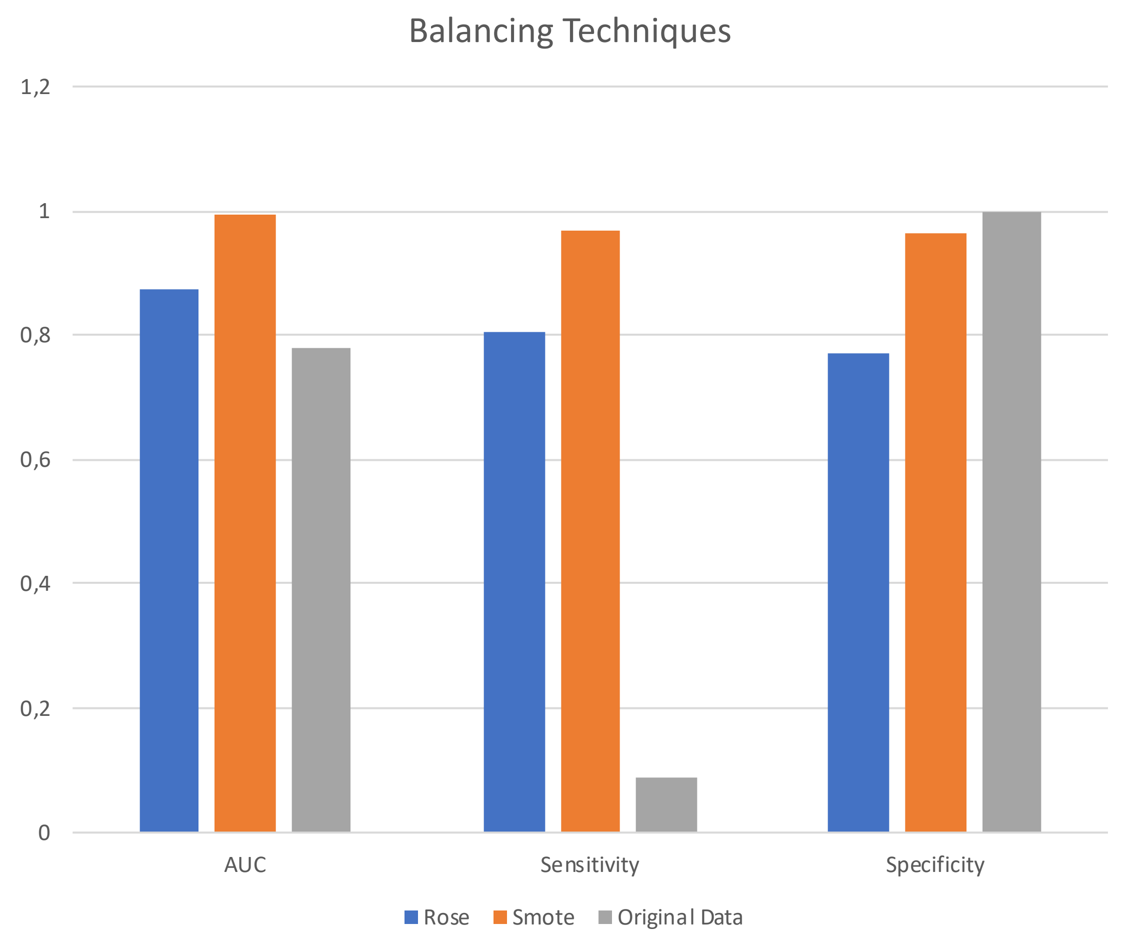
3.4 Data Preparation: Balancing

SECOM with imbalanced dataset:

- Bias the prediction model **towards the majority class**
- Prediction model with imbalanced dataset **yield bad accuracy and other measures**
- **Sensitivity** of the result is lower than 0.1 which is very critical in SECOM case

Balancing methods:

- **Original Dataset**: Possess high specificity, but very low sensitivity since the data is imbalanced
- **ROSE:** Improve all criteria, compared to original dataset
- **SMOTE:** performs better than other balancing methods, including ROSE



3.5 Data Preparation: Scaling/Normalization

Mandatory for some models

- Some models need scaled dataset in order to perform better or to yield accurate results

Not needed in some models

- No assumptions are needed from some models, such as tree-based models, etc.

3.6 Data Preparation: Transformation

Some data are highly screwed

- Some models need transformed dataset in order to perform better or to yield accurate results
- There are several transformations which are applied box-cox, jeo-johnson

4. Model Building, Evaluation and Selection

4.1 Resampling

4.1.1. Bootstrap

4.1.2. Cross Validation

4.1.3. Repeated Cross Validation

4.2 Model Building

4.2.1. Random Forest

4.2.2. GBM

4.2.3. SVM

4.2.4. kNN

4.2.5. Neural Network

4.2.6. Naïve Bayes

4.2.7. GLM

4.3 Model Evaluation

4.3.1. Hyperparameter Tuning

4.3.2. Evaluate model performance

4.4 Model Selection

4.4.1. Performance on test dataset

4.4.2. Non-Accuracy-Based Criteria (Cost)

4.4.3. Model selection

4. Model Building & Evaluation

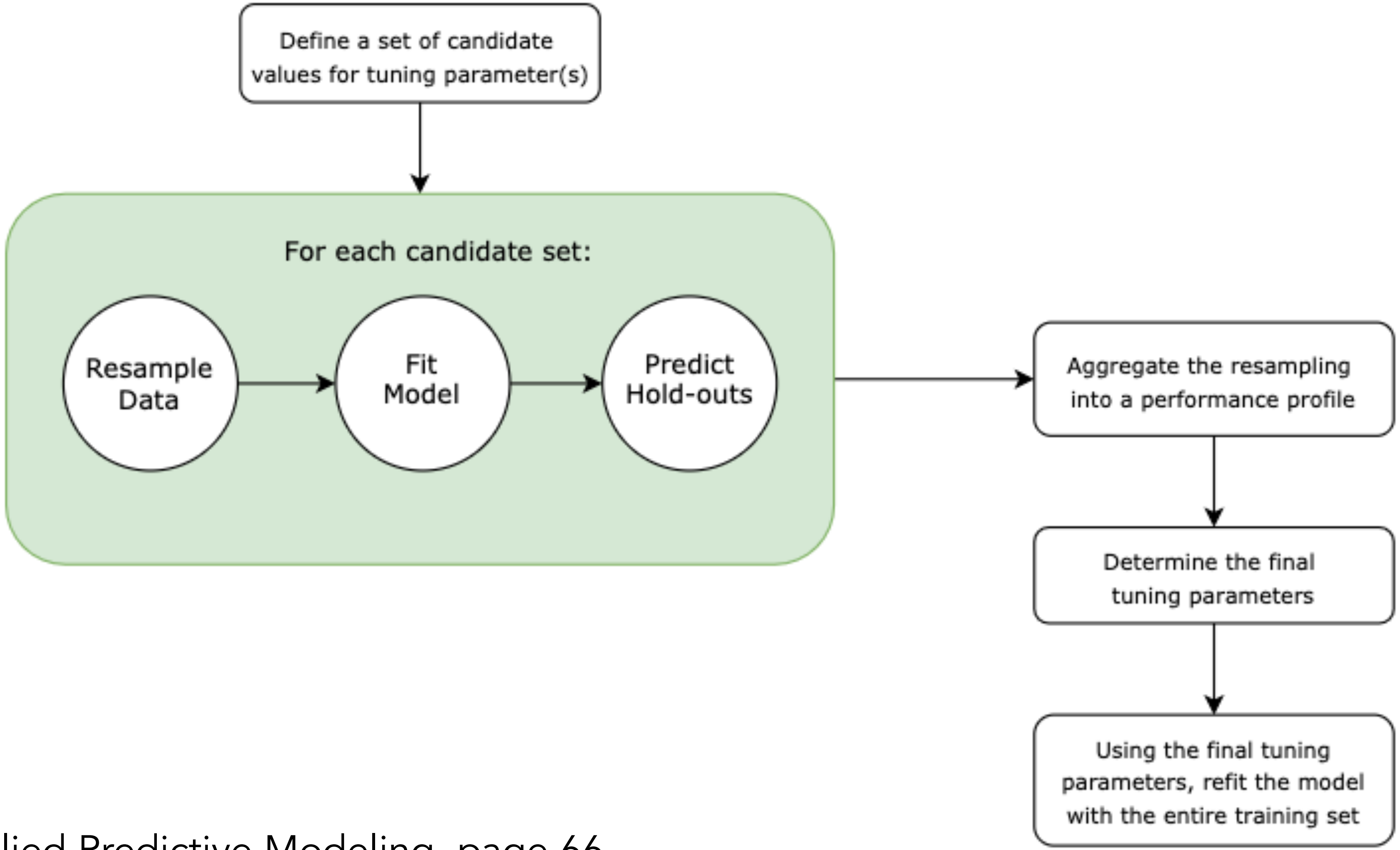
4.1 Resampling

- 20 times **Bootstrapped**, **10-fold cross validation** and 5 times **repeated 10 fold cross validation** are used for creating **validation sets** to tune the hyperparameters and evaluate the models.
- For a given iteration of bootstrap resampling, a model is **built on the selected samples** and is used to predict the out-of-bag samples (samples not selected) for accuracy.
- Bootstrapping is chosen because it reduces **model overfitting** and provides **better performance**

4. Model Building & Evaluation

- Pre-defined resampling folds are being used in control object to make **fair comparisons between models.**
- Pre-defined lists of **seed values** to be stored are used to allow **parallel processing** without errors in tree-based models.
- Started with models that are **the least interpretable** and **most flexible** such as Random Forest or Support Vector Machines.
- Investigated simpler models that are **less opaque** such as Naive Bayes models.
- **Simplest model** that **reasonably approximates the performance** of the more complex methods such as Logistic Regression.

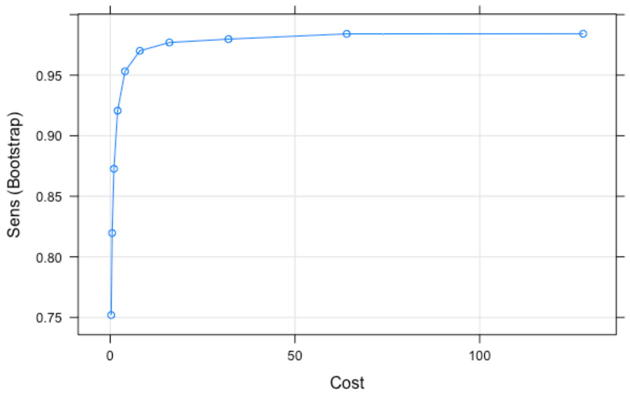
4. Hyperparameter Tuning Process



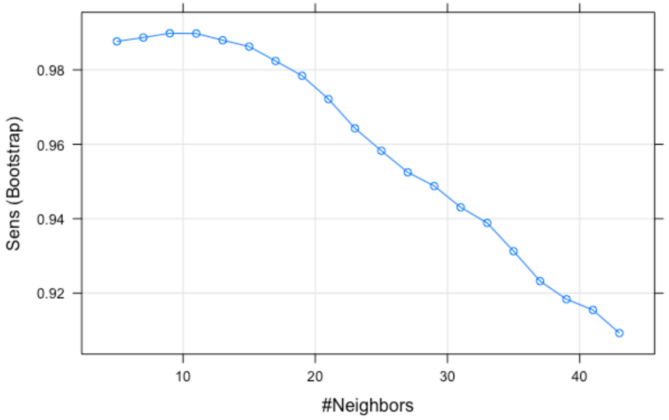
Max Kuhn, Applied Predictive Modeling, page 66.

4. Hyperparameters Tuning

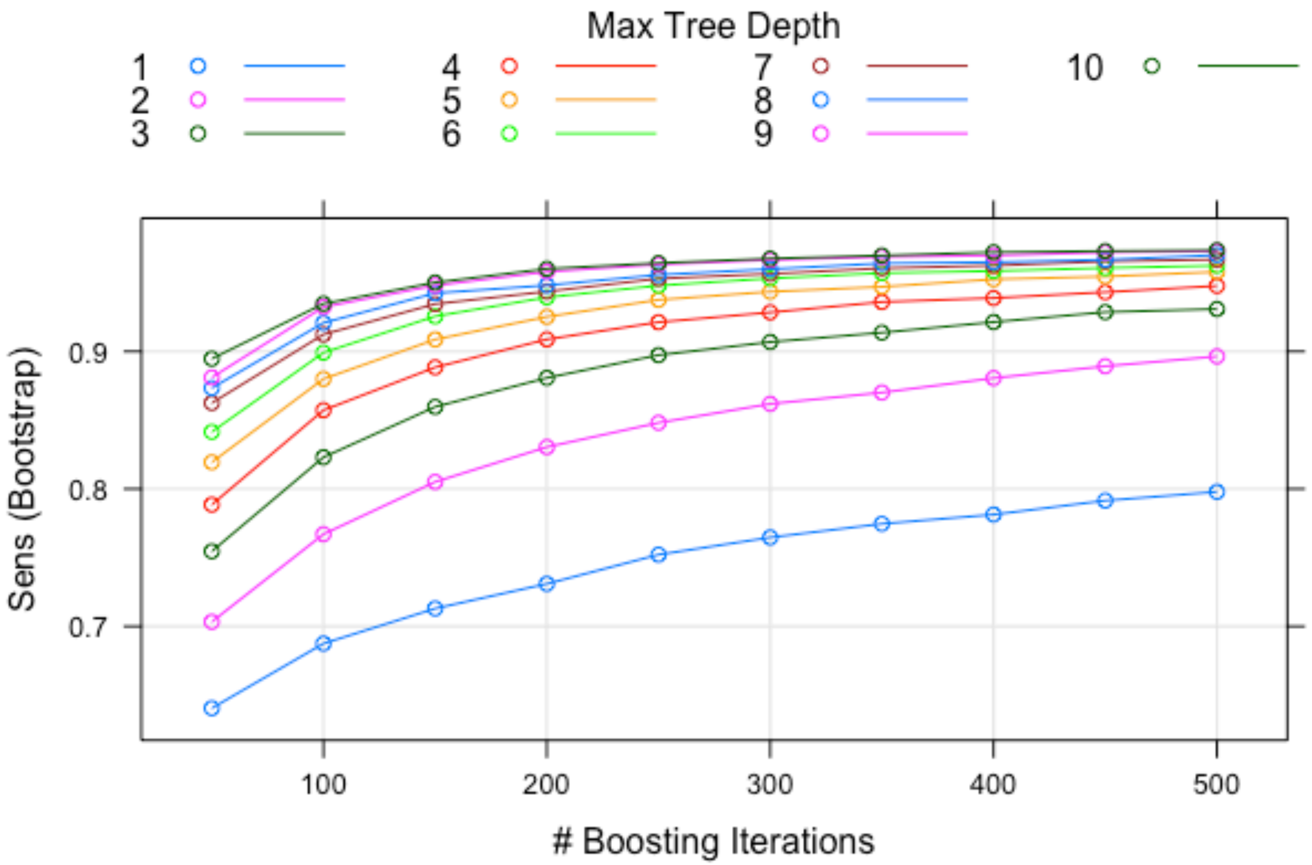
SVM



kNN



GBM



4. Hyperparameters Tuning

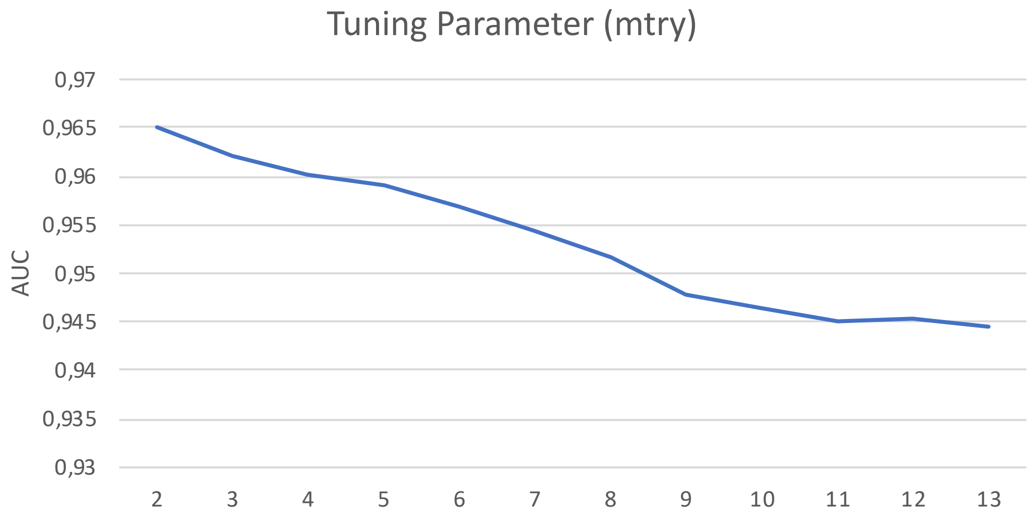
Random Forest

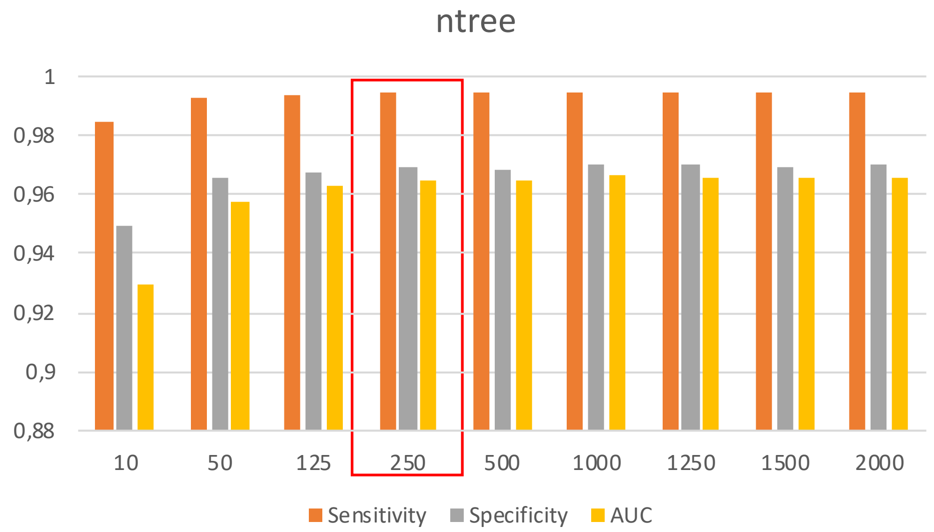
mtry:

- Number of variables randomly sampled as candidates at each split.
- Hyperparameter **mtry = 2** yields the best result in term of AUC, sensitivity, and FN.
- **All criteria are decreasing** when mtry is greater.

ntree:

- Hyperparameter (ntree = 250), **significantly increase** until ntree equals to 250.
- After 250, there is **no significant improvement** in the model.



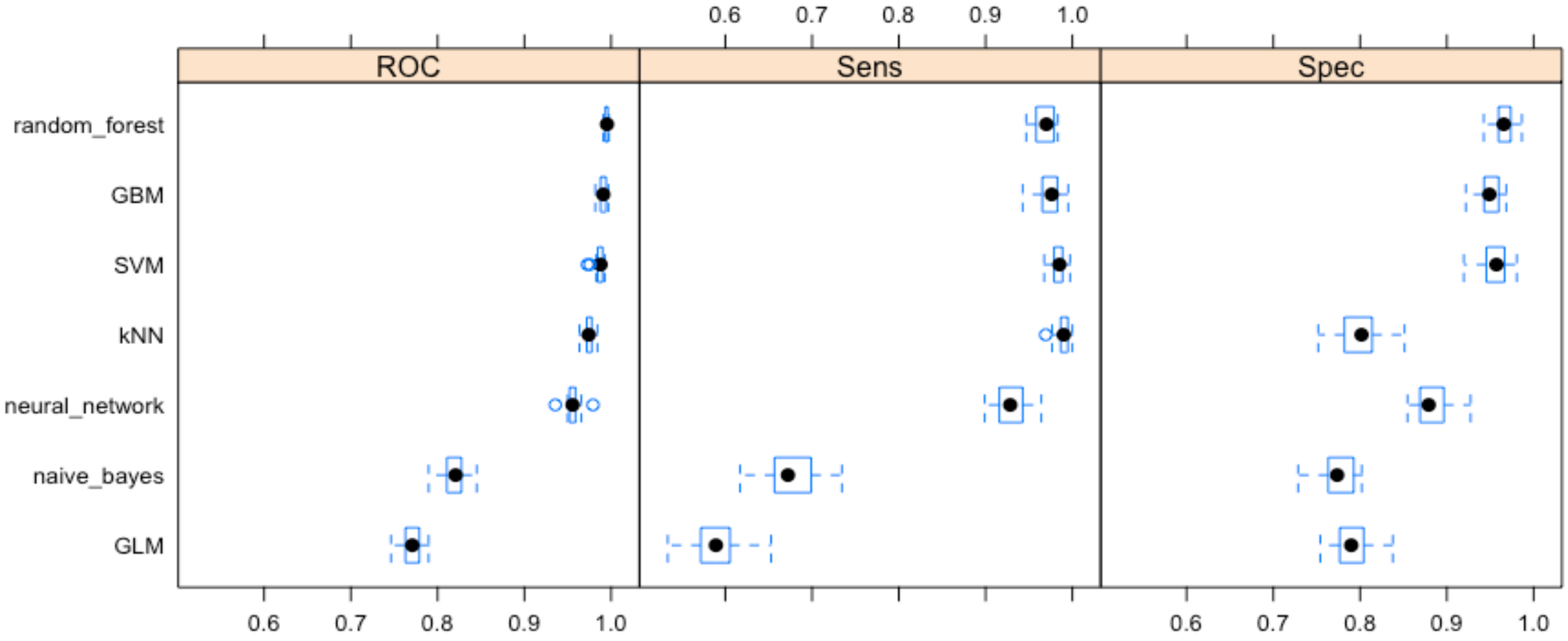


4. Model Evaluation

	AUC	Sensitivity	Specificity	Precision	F1	FN	FP	Resampling	Total Cost (15:1 Cost Ratio)
Random Forest	**0.996**	0.969	**0.965**	0.970	0.965	245	**303**	16473	3978
GBM	0.994	0.974	0.950	0.962	0.960	205	437	16473	3512
SVM	0.990	0.984	0.955	**0.981**	**0.968**	123	389	16473	**2234**
kNN	0.977	**0.990**	0.798	0.793	0.895	**79**	1748	16473	2933
Neural Network	0.959	0.929	0.883	0.931	0.903	557	1010	16473	9365
Naïve Bayes	0.827	0.674	0.775	0.808	0.701	2551	1948	16473	40213
GLM	0.778	0.587	0.792	0.737	0.646	3234	1797	16473	50307

4. Model Building & Evaluation

Comparison of Models



Results from R, fitting model with Train Dataset with best parameters

4. Model Selection

	AUC	Sensitivity	Specificity	Precision	F1	FN	FP	Resampling	Total Cost (15:1 Cost Ratio)
Random Forest	0.721	0.579	**0.864**	**0.216**	**0.314**	8	**40**	313	**160**
GBM	**0.725**	**0.789**	0.660	0.130	0.224	**4**	100	313	**160**
SVM	0.712	0.632	0.793	0.164	0.261	7	61	313	166
kNN	0.689	0.684	0.694	0.126	0.213	6	90	313	180

- **Random Forest** shows the best performance on **test dataset** with two hyperparameters: mtry = 2, and ntree = 250
- Cost = (15 * FN) + (1 * FP)

4. Alternative Cutoffs

	AUC	Sensitivity	Specificity	Precision	F1	FN	FP	Resampling	Total Cost (15:1 Cost Ratio)
Train Dataset	0.967	0.969	0.965	0.961	0.965	240	306	16473	3906
Test Dataset (0.652 threshold)	**0.721**	0.579	**0.864**	**0.216**	**0.314**	8	**40**	313	**160**
Test Dataset (0.698 threshold)	0.719	0.632	0.806	0.174	0.273	7	57	313	162
Test Dataset (0.842 threshold)	0.705	**0.947**	0.463	0.102	0.185	**1**	158	313	173

Best Model:

- Best predictive model is conducted by **MICE, Boruta, SMOTE, and Random Forest** with their respective best parameters
- Best model fits yields 0.967 AUC on train set, while yielding **0.721 with test dataset.**

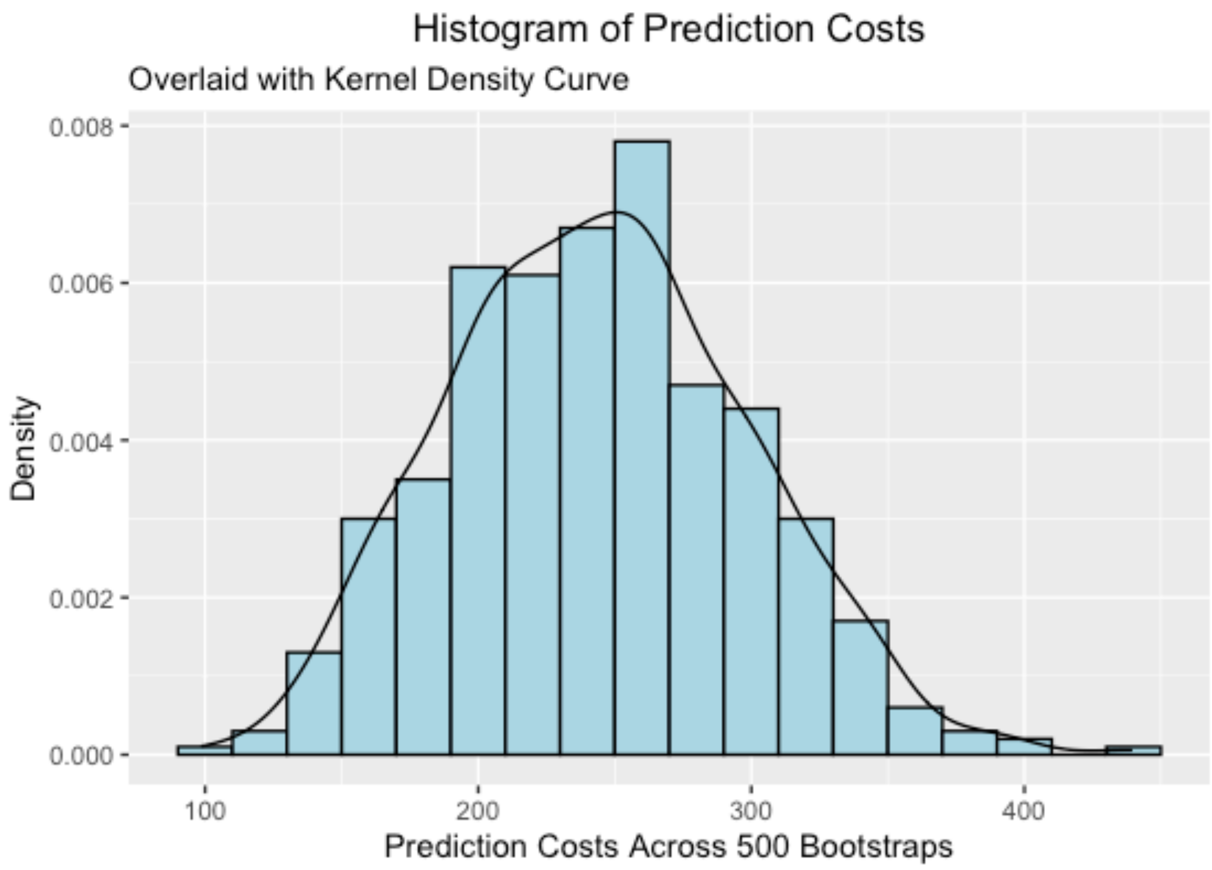
Different cut-off threshold:

- Set **cost ratio of** FN:FP **to 15:1.**
- **Cut-off threshold of 0.652** yields the best result in overall cost

5. Model Deployment & Results

Bootstrap Simulation

- Model Consistency is tested on 500 different datasets created from the raw data using Bootstrapping.
- A procedure calculates the cost of false predictions for each dataset in 95% confidence interval.
- Results are shown in histogram.



6. CRISP-DM Retrospective

Pros

- A roadmap to follow
- Iterative process
- Effective methodology
- Control (Checklists and process frameworks)

Cons

- Inflexible
- Lack of clarity in decision paths
- Not entirely efficient for projects with multiple teams