Master
Project Management
and Data Science

17/05/2024

PROF. DR. TILO WENDLER

SECOM Case Study

Dhruvi Jay Patel(s0592755)

Naman Mathpal (s0590500)

Bhoomika Jagadeesha (s0590573)

Jui Prasad Kulkarni (s0590496)

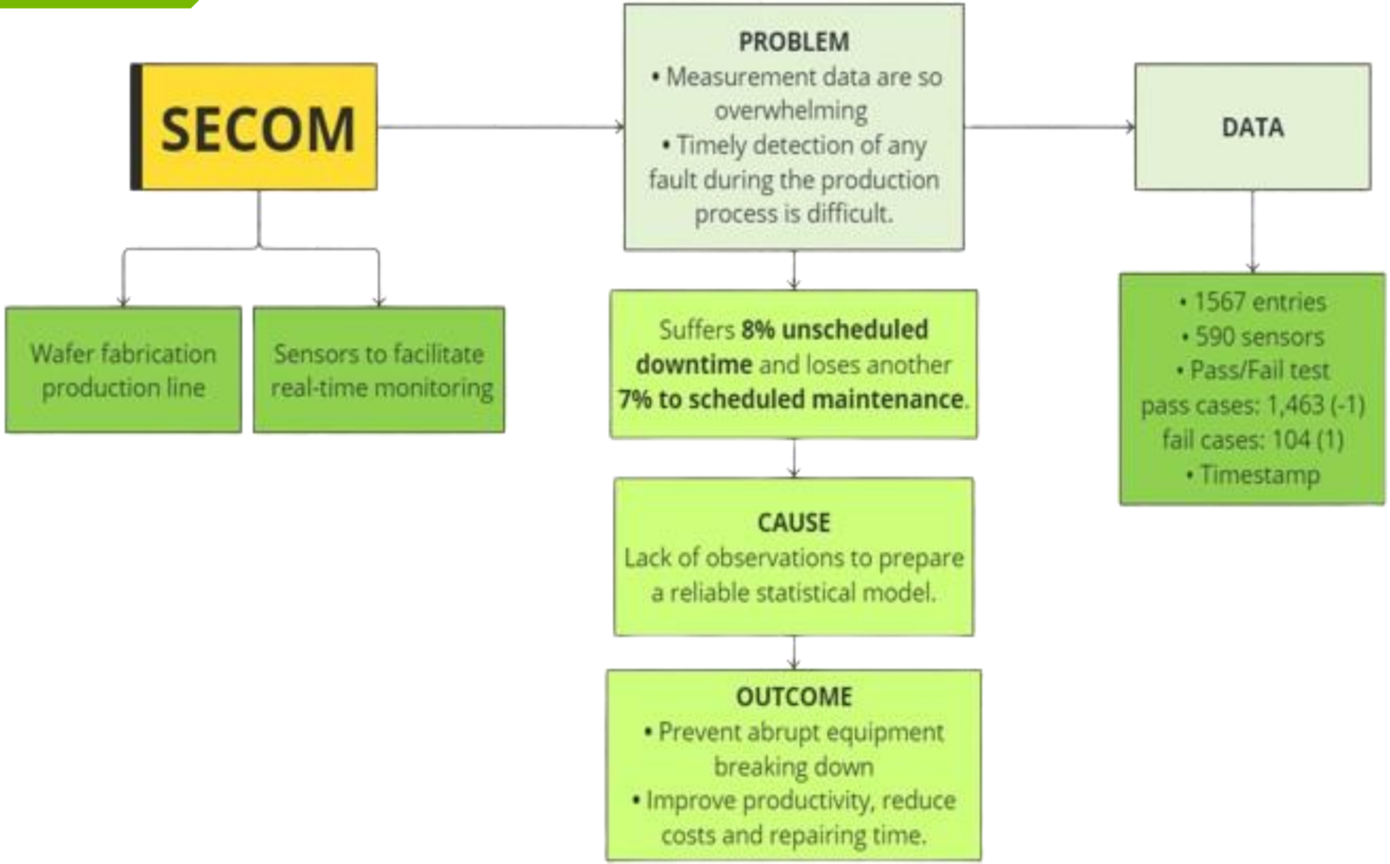
Ankit Satish Gupta (s0590516)

htw.

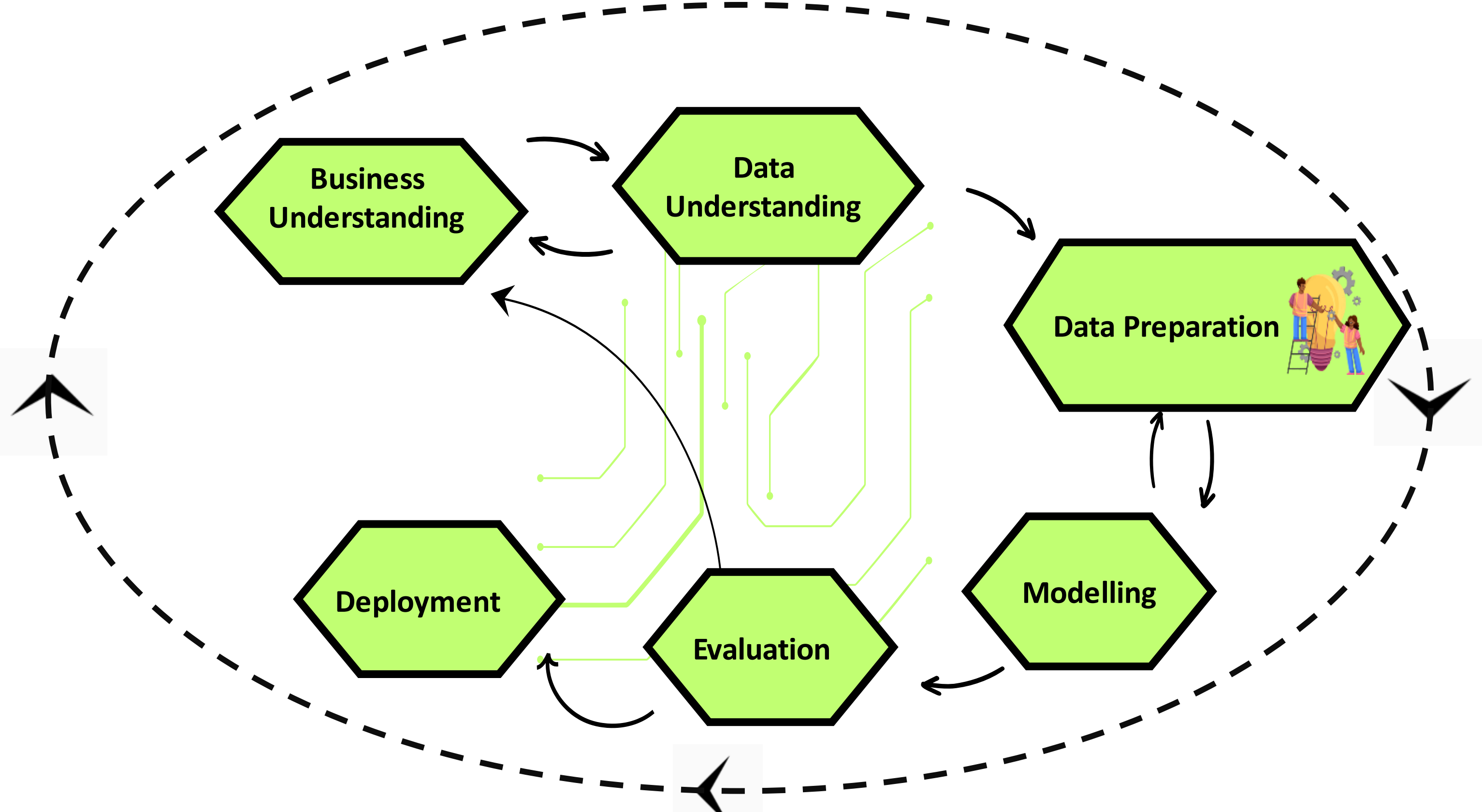
INDEX

Master
Project Management
and Data Science

1. Introduction



2. CRoss Industry Standard Process for Data Mining

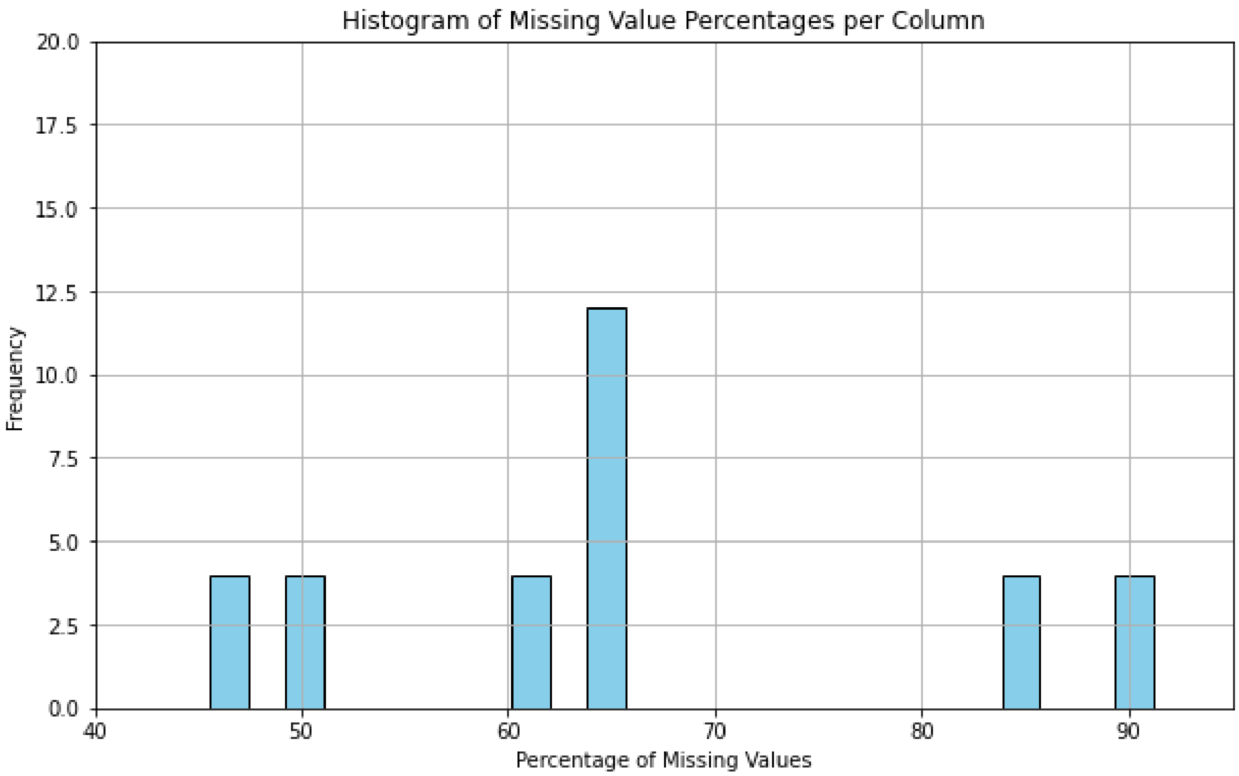


3. Histogram of missing values

1. Most percentage of missing values lies in 40 -90 %

2. In between 60 -70% **(64.96%)** shows highest missing values with **12 features** frequency

3. Most of the percentage of missing values lies in the frequency 2.5 to 5.0 frequency of the features.

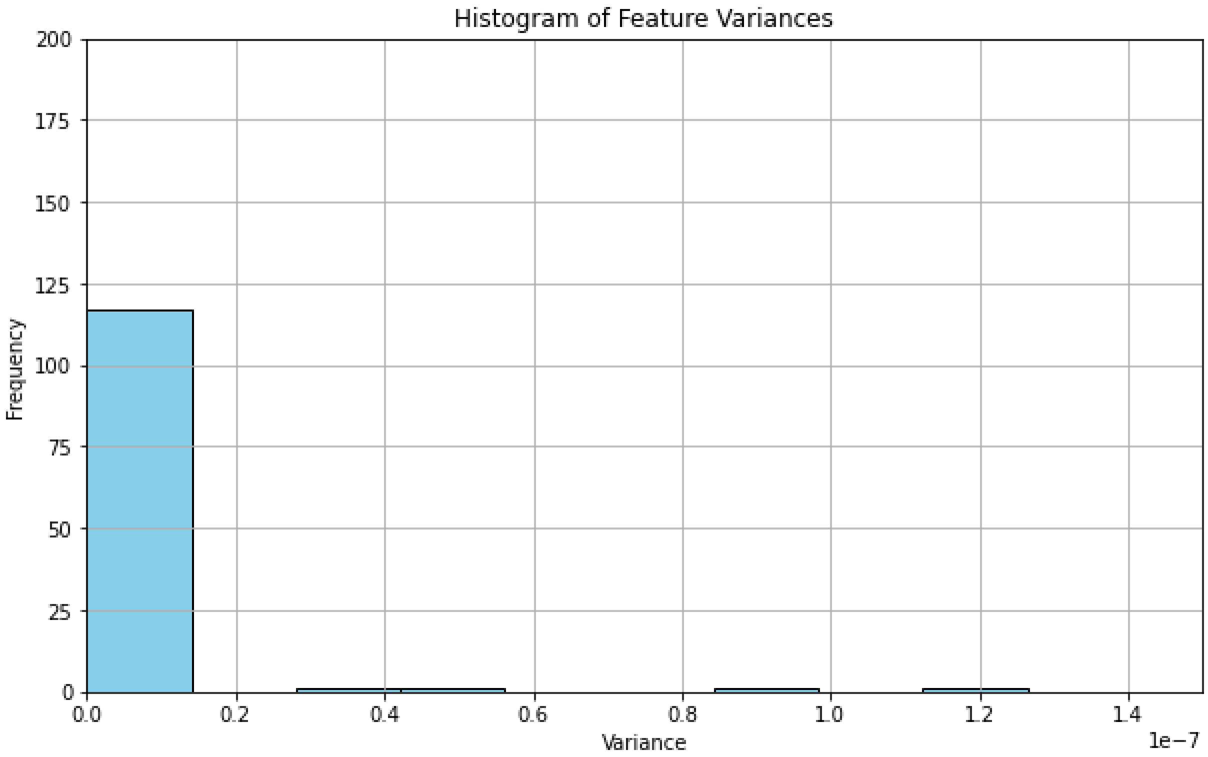


4. Histogram of Volatility

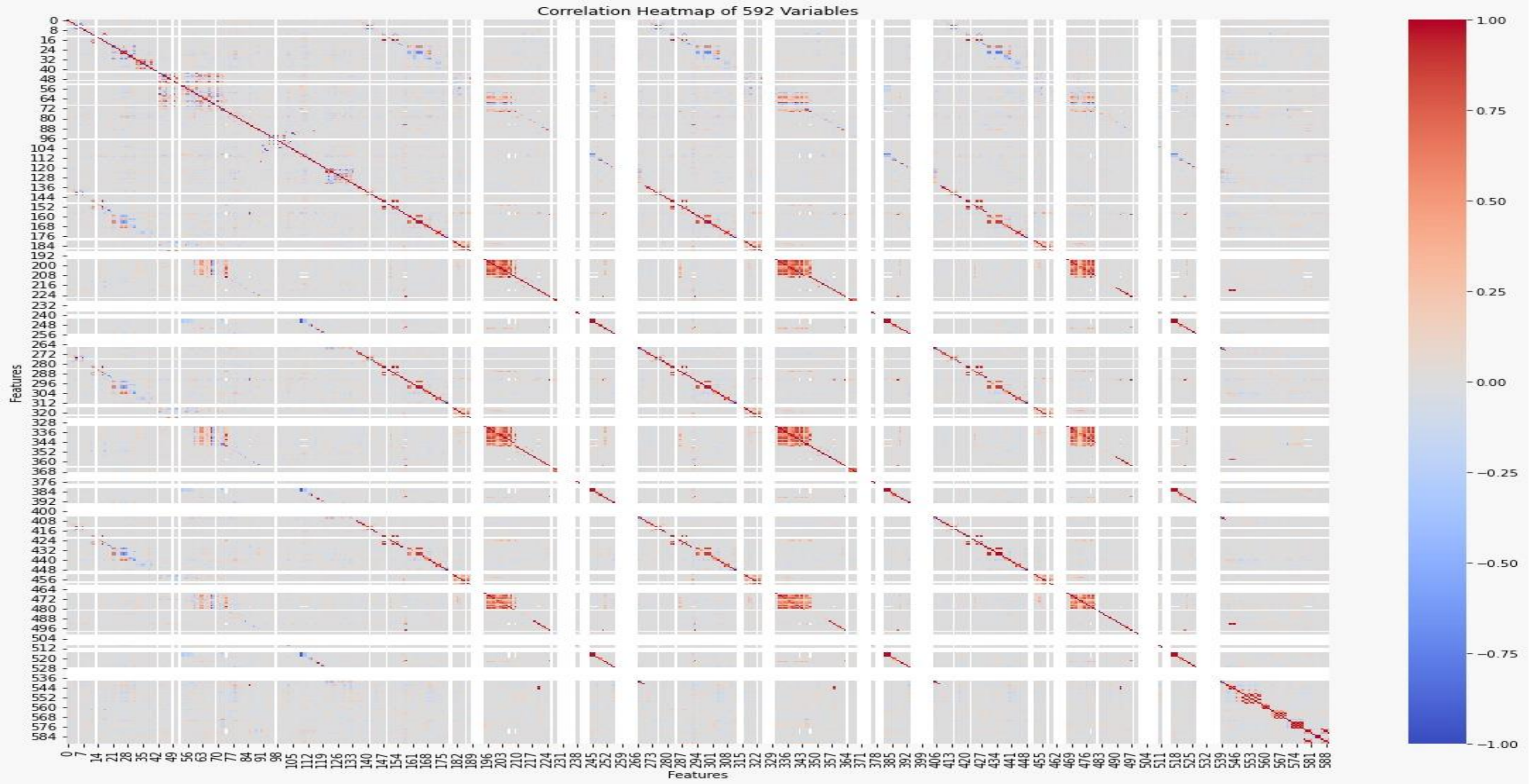
1. Maximum values in variance lies in between 0.0 to e-7.

2. Frequency of features of **116** shows that the variance is 0.0 and which will be removed for the further procedure.

3. Variance range 0.2 to 1.4 shows the frequency close to 0.



5. Heatmap


Correlation Heatmap of 592 Variables
Features
Features
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Master
Project Management
and Data Science

6. Duplicate analysis

- Found **31 duplicates** in Time stamp

```
duplicate_rows = label.duplicated()

# Count the number of duplicate rows
num_duplicates = duplicate_rows.sum()

print("Number of duplicate rows:", num_duplicates)
```

```
Number of duplicate rows: 30
```

- Found **104 features** which are duplicates
- These are the same features which has 0 variance

```
total_duplicate_features = sum(secom.T.duplicated())

# Print the total number of duplicate features
print("Total number of duplicate features:", total_duplicate_features)
```

```
Total number of duplicate features: 104
```

Let's Split the Data

7. Data Splitting and Frequency Distribution of Target Variable

WHY?

- Performance Estimation
- Avoid overfitting
- Reduction in Bias

HOW?

- **75% and 25%.**
- Train the Model: The model is trained on the training set.
- Test the Model: The final model is evaluated on the test set to assess its performance.
- Constraint: Ensuring same proportion of pass and fail cases **(14:1)** using **stratified sampling.**

Data (14:1)
Total entries- 1567
Pass cases:1463
Fail cases:104

=

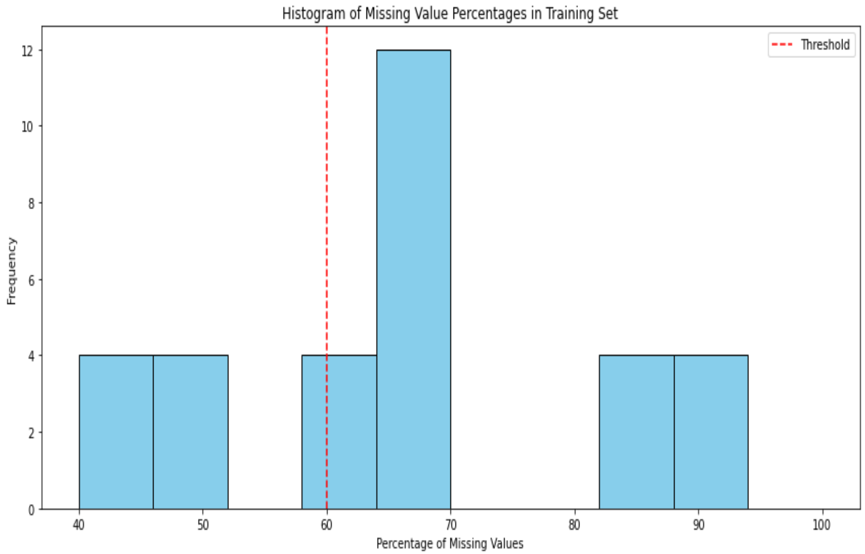
Test
Total entries: 392
Pass Cases: 366
Fail cases: 26

+

Training
Total entries- 1175
Pass cases: 1097
Fail cases: 78

BEFORE **AFTER**

8. Threshold definition

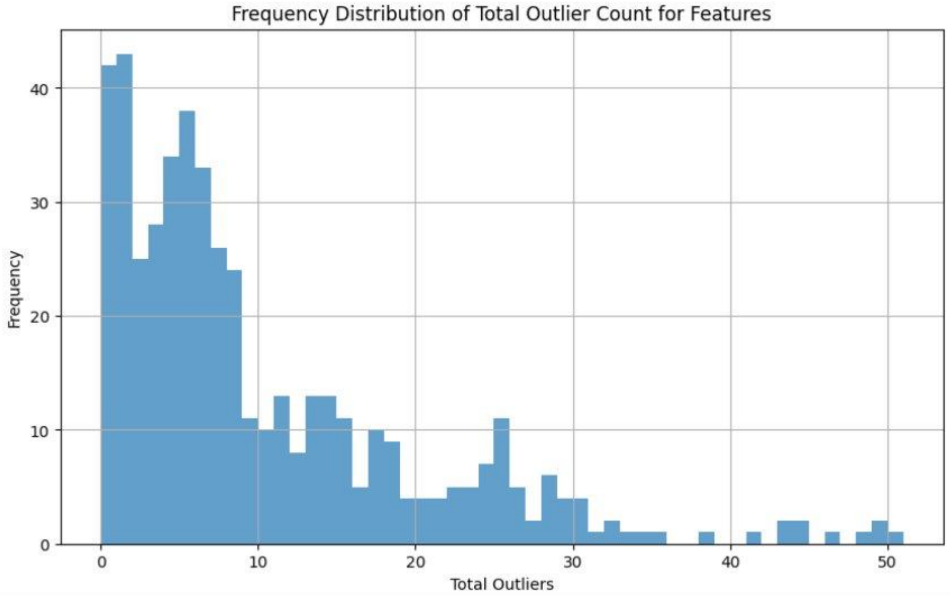


Observations	Steps taken	Before	After
Features with many missing values do not contribute to the quality of model	• Threshold to **60%** • Remove the features • For remaining NAs, imputation can be done	Above 60% - **24 features** found	Above 60% - **24 features** found

9. Action Points on Observations

	Observations	Steps taken
Duplicates	*Label data* • **31 duplicates** *SECOM data* • **Column wise-104** features • **Row wise- 0**	• Merge the two dataframes- unique rows • For duplicate features – **Remove** it from dataset for better computation.
Variance	• **Zero Variance- 115** features • Does not contribute in the model (constant entries).	• **Remove** 115 features - training data. (which includes duplicates as well) • **Feautres: Before**- 590; **After**- 475
TimeStamp	• Can not analyse date and time together.	• Split date and time • Do not remove at this stage (Can be helpful in further analysis)

10. Outlier Analysis




```
Feature 36: 50 outliers
Feature 460: 49 outliers
Feature 456: 49 outliers
Feature 442: 48 outliers
Feature 458: 46 outliers
Feature 461: 44 outliers
Feature 457: 44 outliers
Feature 151: 43 outliers
Feature 250: 43 outliers
Feature 459: 41 outliers
```

Observations	Frequency	Method	Possible Action Points
432 Features	Range: 50 to 1 (**4.25% to 0.06%**)	Z-Score	• No Action: proportion of outlier is less • Overwrite: S-boundaries (chances of better substitute) • Remove (can produce more blanks)

11. Take Home Messages

Business understanding:

- Comprehensive data understanding is essential for effective model development.

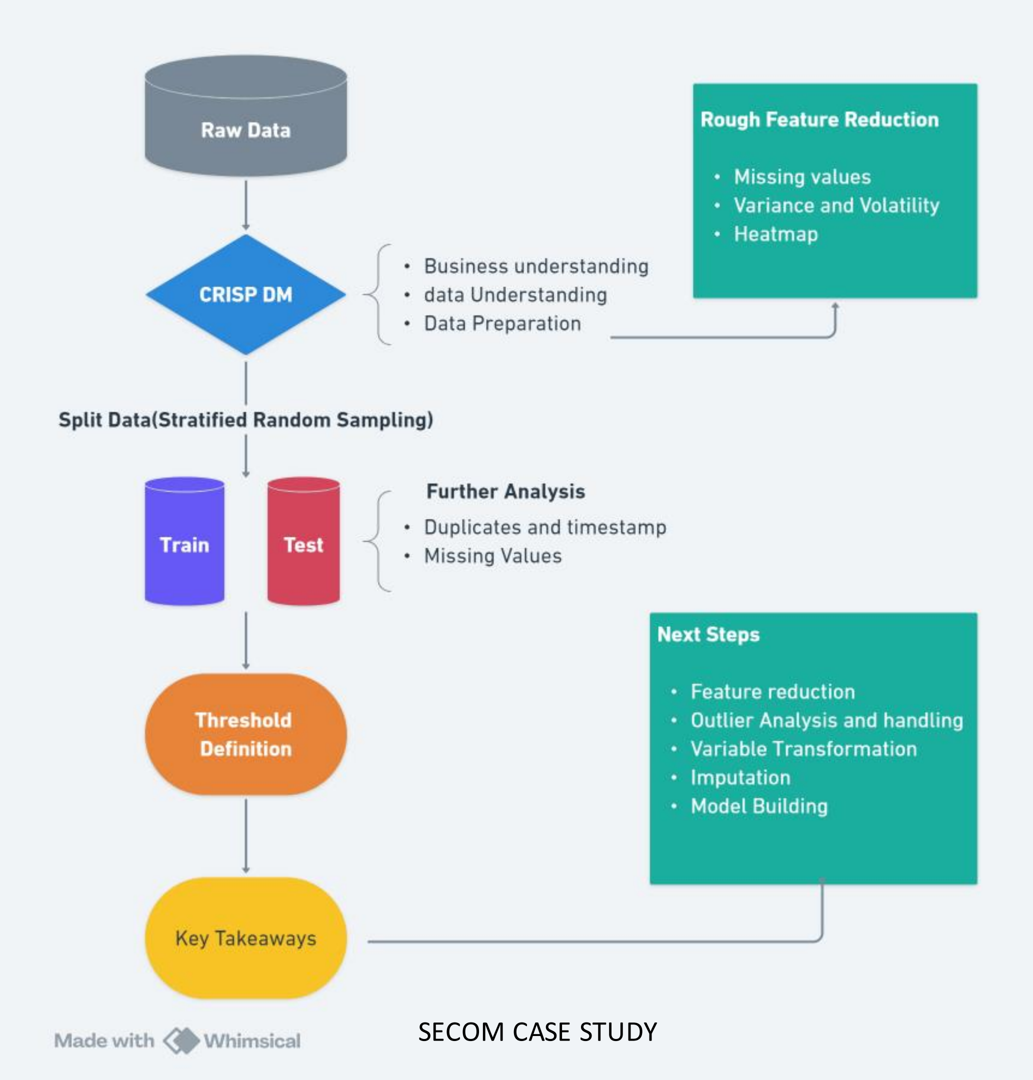
Data understanding:

- Proper data splitting ensures unbiased model evaluation and validation.
- Addressing duplicates and missing values is crucial for building reliable and accurate models.

Data preparation:

- Data Transformation like Log, Box Cox or Min-Max Scaling is important! Why?
- Normalize or standardize numerical features to ensure that they have a similar scale and normal distribution.
- Visualizing data distributions and missing values help identify data quality issues and further data preprocessing.

12. Recapitulation



Vielen Dank für Ihre Aufmerksamkeit !

Master
Project Management
and Data Science

21/06/2024

PROF. DR. TILO WENDLER

SECOM Case Study

Dhruvi Jay Patel (s0592755)

Naman Mathpal (s0590500)

Bhoomika Jagadeesha (s0590573)

Jui Prasad Kulkarni (s0590496)

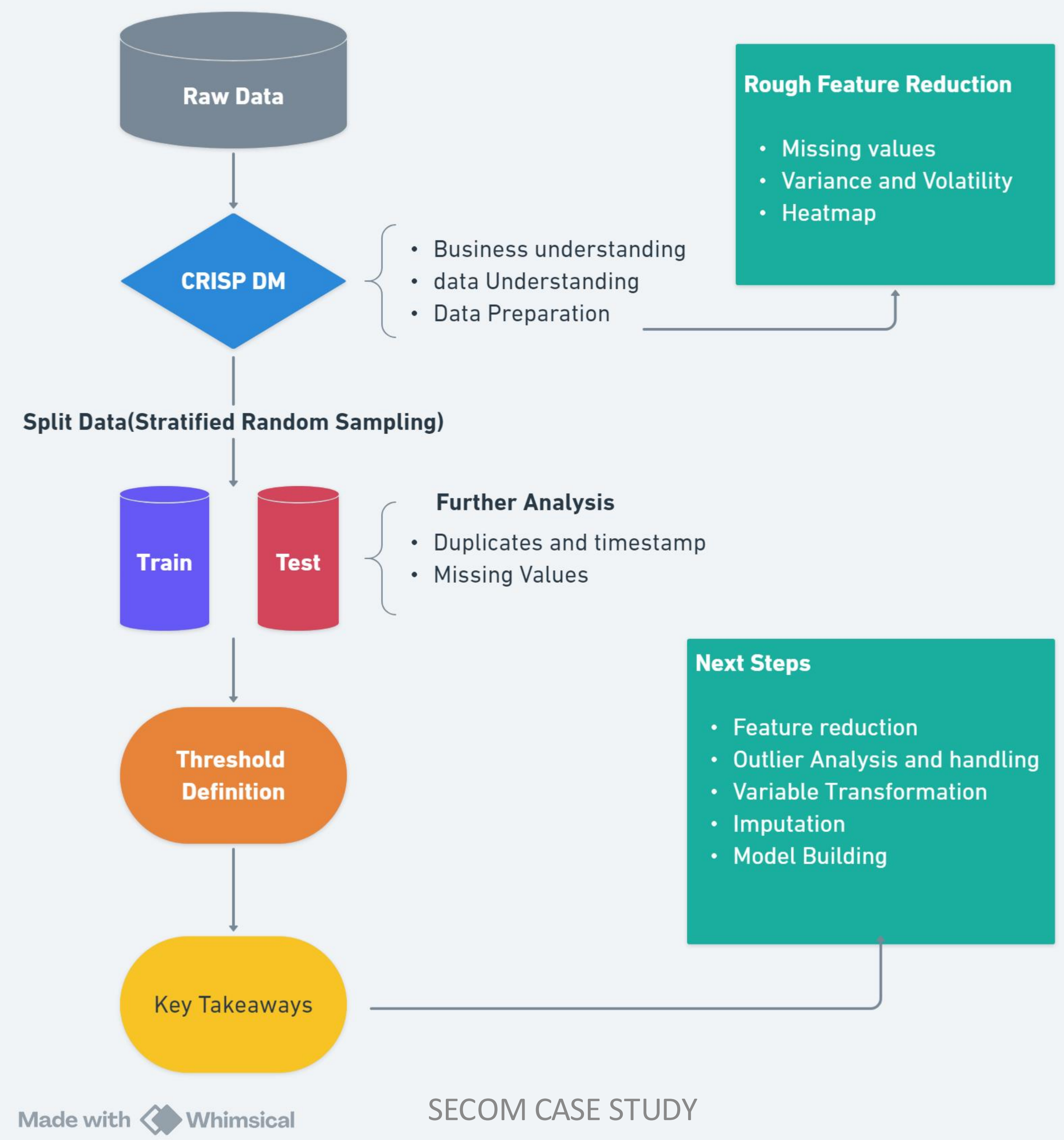
Ankit Satish Gupta (s0590516)

htw.

INDEX

1. Recap until now
2. CRISP DM
3. Data Cleaning – Rough feature reduction
4. Outlier handling
5. Imputation
6. Feature Selection and Reduction
7. Imbalanced Data handling
8. Decision Hierarchy
9. Model building
10. Take Home Messages
11. Summary and Next steps

Quick Recap


Raw Data
CRISP DM
Business understanding
data Understanding
Data Preparation
Rough Feature Reduction
Missing values
Variance and Volatility
Heatmap
Split Data(Stratified Random Sampling)
Train
Test
Further Analysis
Duplicates and timestamp
Missing Values
Threshold Definition
Next Steps
Feature reduction
Outlier Analysis and handling
Variable Transformation
Imputation
Model Building
Key Takeaways
Made with Whimsical

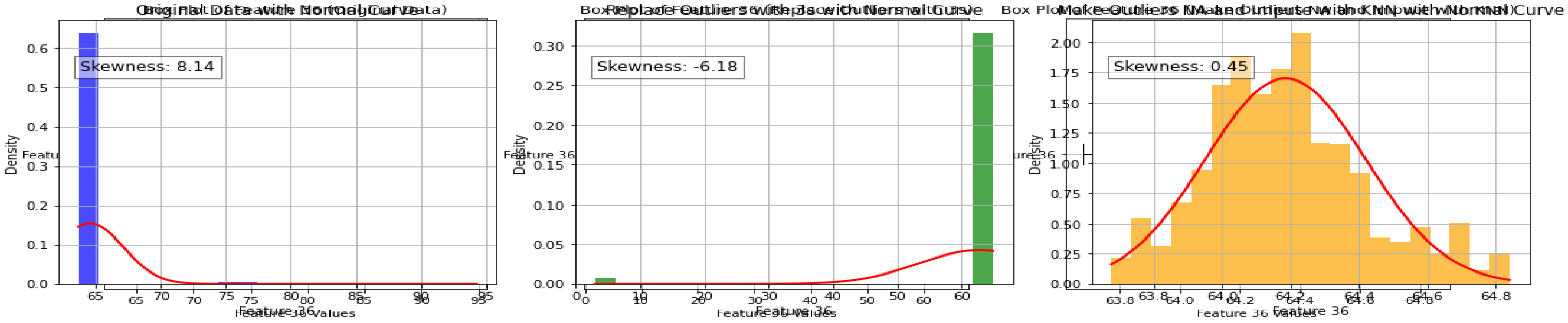
CRoss Industry Standard Process for Data Mining

Data cleaning – Rough Dimensionality Reduction

	Observations	Impact and decision
Duplicates	• Label data - 31 duplicates • Column wise-104, Row wise- 0	• Merge the two dataframes- unique rows • For duplicate features – Remove it from dataset for better computation (same which has 0 variance)
Variance	• Zero Variance- 115 features, • Does not contribute in the model (constant entries).	• Remove 115 features - training data. (which includes duplicates as well) • Feautres: **Before**- 590; **After**- 475
TimeStamp	• Can not analyse date and time together. • Further algoritms can only take numeric(continous) predictors like Boruta	• More features can lead to overfitting as per model complexity • Dropped
Missing Values	• Found Missing Values	• Select a threshold and remove missing values above it • Imputation for remaining

Outlier Handling

Feature 36



Approach	Pros	Cons	Impact	Decision
Original data (No action)	-Retain original data -No information loss	-Misleading impact -Less reliable model due to **influence of extreme values**	Many features are positively/negatively skewed **-Misleading interpretations** as outliers dominate the dataset	Not the best approach
Replace with 3s boundaries	-Addresses extreme outliers -Chances of better substitute	-Can impact whole distribution if many outlier presence -Not work well with datasets where outliers are not well-separated	If majority of outliers are on one tail, removing them flip the data's shape, **altering skewness drastically**.	**Medium priority**
Remove and Impute	-**Maintains distribution** to a certain level -KNN attempts to fill missing values with **realistic estimates** based on similar data points	-Produce more blanks -KNN imputation can be computationally expensive for large datasets -Sensitive to Noise	Fills missing values based on **similarities** with neighboring data points.	**First priority**

Missing value Imputation

WHY??

- Complete dataset utilization
- Affect the quality of the model

Reasons

- Missing at Random (MAR)- Example: Scale running out of power while collecting
- Missing Completely at Random (MCAR): Example: production line fault
- Missing Data Not at Random (MNAR): Example - Scale is not reliable or is too old

When??

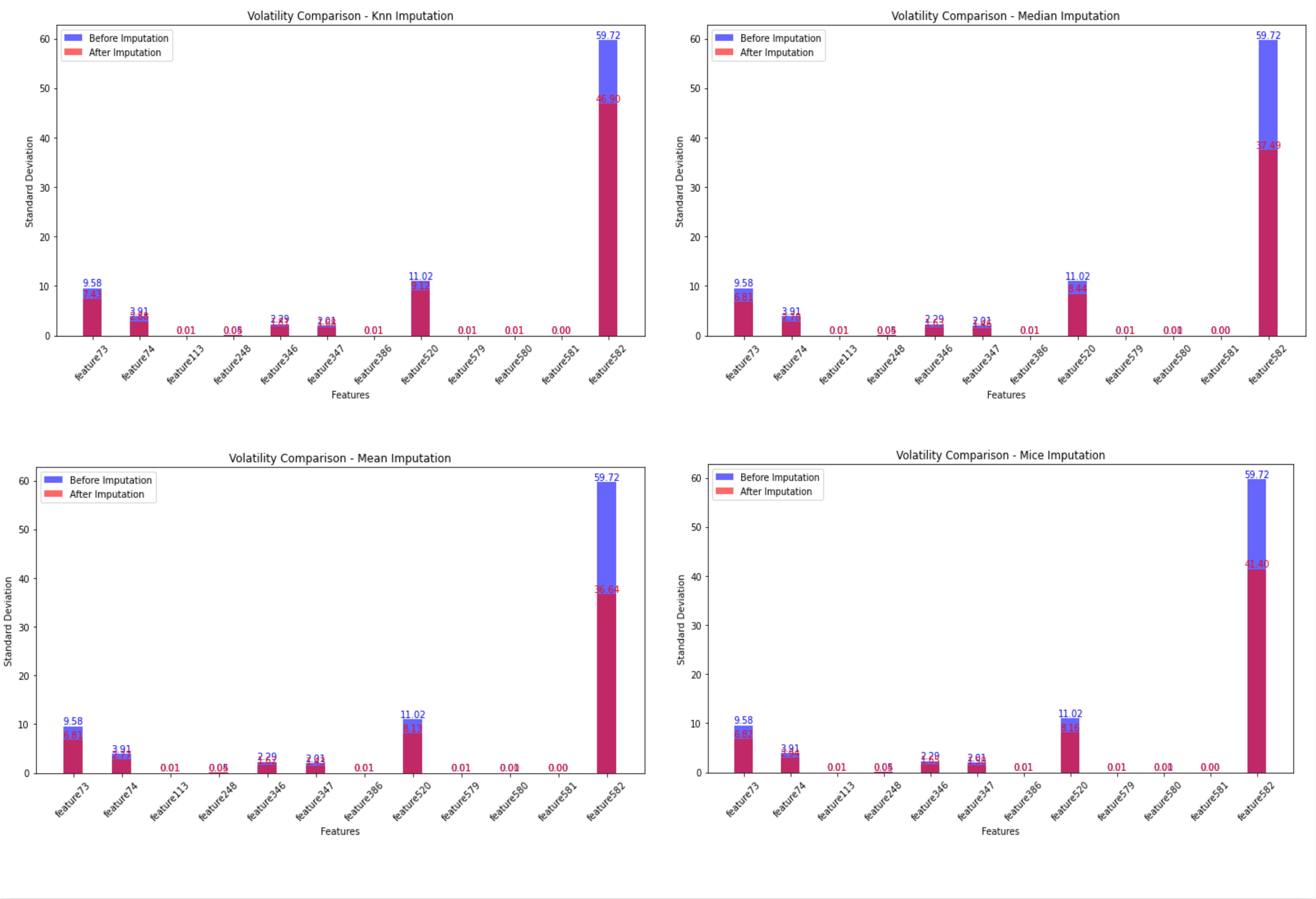
- Outlier handling might also lead to missing values
- Hence, after outlier handling

Different approaches to impute missing values

Name	Approach	Advantages	Disadvantages	Effect and Decision
Mean Imputation	• Impute mean value of feature	• Easy to implement • Cheap	• **Underestimate volatility (reduce)** • **Disort distribution of data** • Does not consider correlation with other variables • May introduce bias	• Suitable for small datasets • **Greater difference in volatility** • **Low Priority**
Median Imputation	• Impute median value of feature	• Robust to Outliers	• Can still **distort the distribution of** the data, although less than mean imputation.	• better than mean for skewed distributions and when **outliers are present.** • **Low priority (volatility dfifference)**
Regression Imputation	• Select predictors - **highly correlated with the feature** having missing value	• Deterministic • Uses relationship between variables	• Might **overfit** only when relationship between variables exist • **Predict linearity** • MAR – Assumption • Volatility not considered	• Not the best approach as **volatility is not considered and assume linear relationship** • **Low Priority**
KNN Imputation	• Type of Hot Deck; Multivariate; Considers Nearest values • First normalize then de-normalize • **Scaling** is temporary (distance-based approach)	• Utilizes multivariate information • Preserves relationships • Both numerical and Categorical data. • **More accurate**	• Computationally intensive • **Choice of K can affect the result**	• **Change in Volatility is less** • **High priority.**
MICE	• **Multi-variate imputation** by chained equations. • Considers more than 1 candidate to find substitutes; Iterative steps.	• Multiple imputation with multiple candidates	• Assumption: MAR, MACR • Complex • Computationally Intensive	• Less difference than mean or median • **Medium priority**

Volatility Comparison

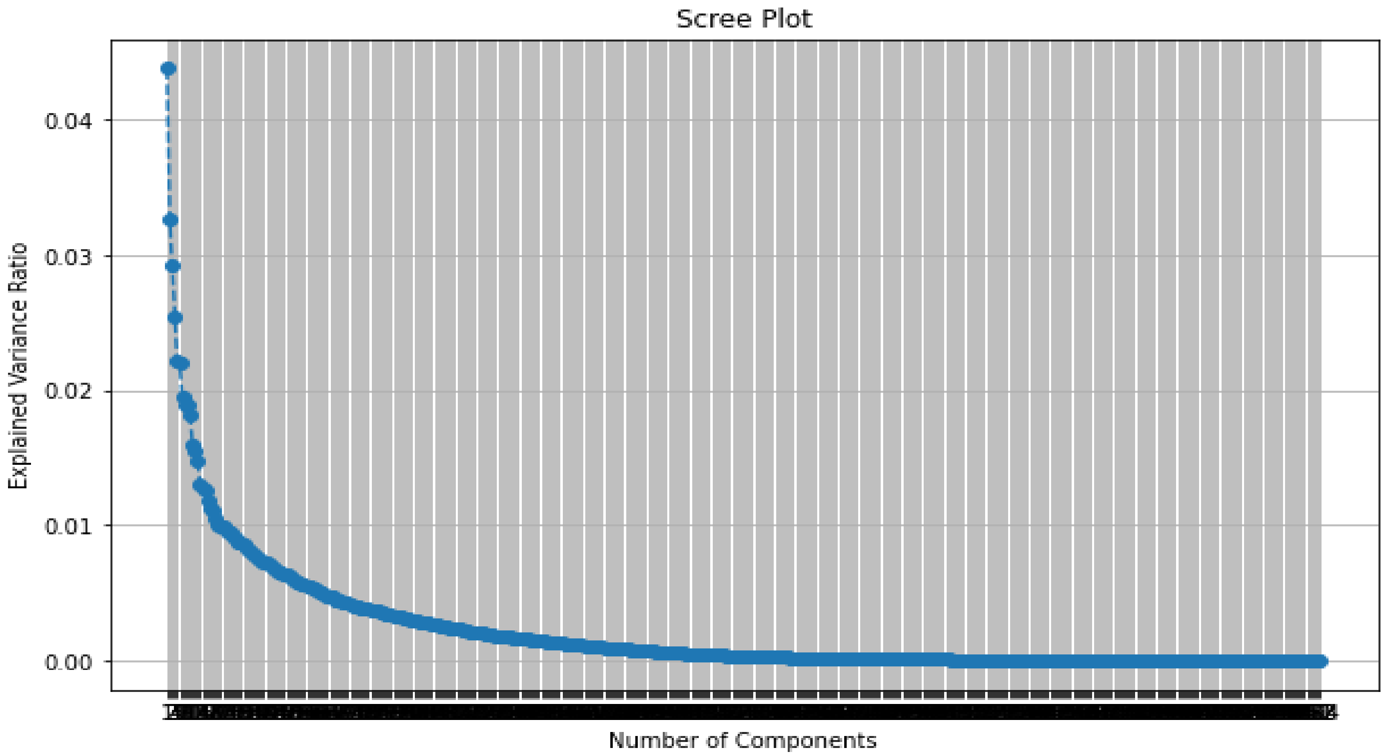
Imputation Techniques (40-65%)



Feature selection and reduction

Feature Selection			
HOW?? Select subset of important features	**WHY??** 1. Reduce overfitting 2. Need to understand importance of features 3. Enhanced interpretability 4. Faster Computation	**WHICH??** Wrapper**(Boruta),** Embeded and filter	**Boruta** Finds the importance of the features by constructing shadow features (random shuffling each characteristics).
Feature Reduction			
HOW?? Reduce dimentionality and creates new components on the basis of features	**WHY??** 1. Reduce overfitting and noise 2. Dimensionality Reduction and removes multicollinearity 3. Where overall structure matters and not the features 4. Faster Computation	**WHICH??** Linear **(PCA)** and Non-Linear	**PCA** 1. Analyses and explains most common variances in variables. 2. Identifies the common factor and converts them to components

Data exploration - Cattell's Scree plot, KMO and PCA



Why not PCA??

- Loss of Interpretability
- Linearity assumption
- Data Centering -Mean Centering Requirement: PCA requires data to be mean-centered.
- Choose when Target variable is not a primary focus
- For Unsupervised learning . Goal: feature reduction without considering the target variable.

- **Scree plot** – **no elbow or break point** where the eigenvalues start to level off
- **KMO**
- Multicollinearity: If the original data had multicollinearity (high correlation among features), this can lead to issues in the KMO test. Multicollinearity can cause computational problems, resulting in NaN values in the KMO statistic.
- **KMO statistic: 0.65** (after removal of highly collinear features – Important features can be discarded!!!!)
- **PCA mediocrely suitable for factor analysis not ideal one.**

Why BORUTA??

```
# Print sorted feature rankings
print("Sorted Boruta feature rankings:")
for feat, rank in features_with_ranking_sorted:
    print(f"{feat}: {rank}")
```

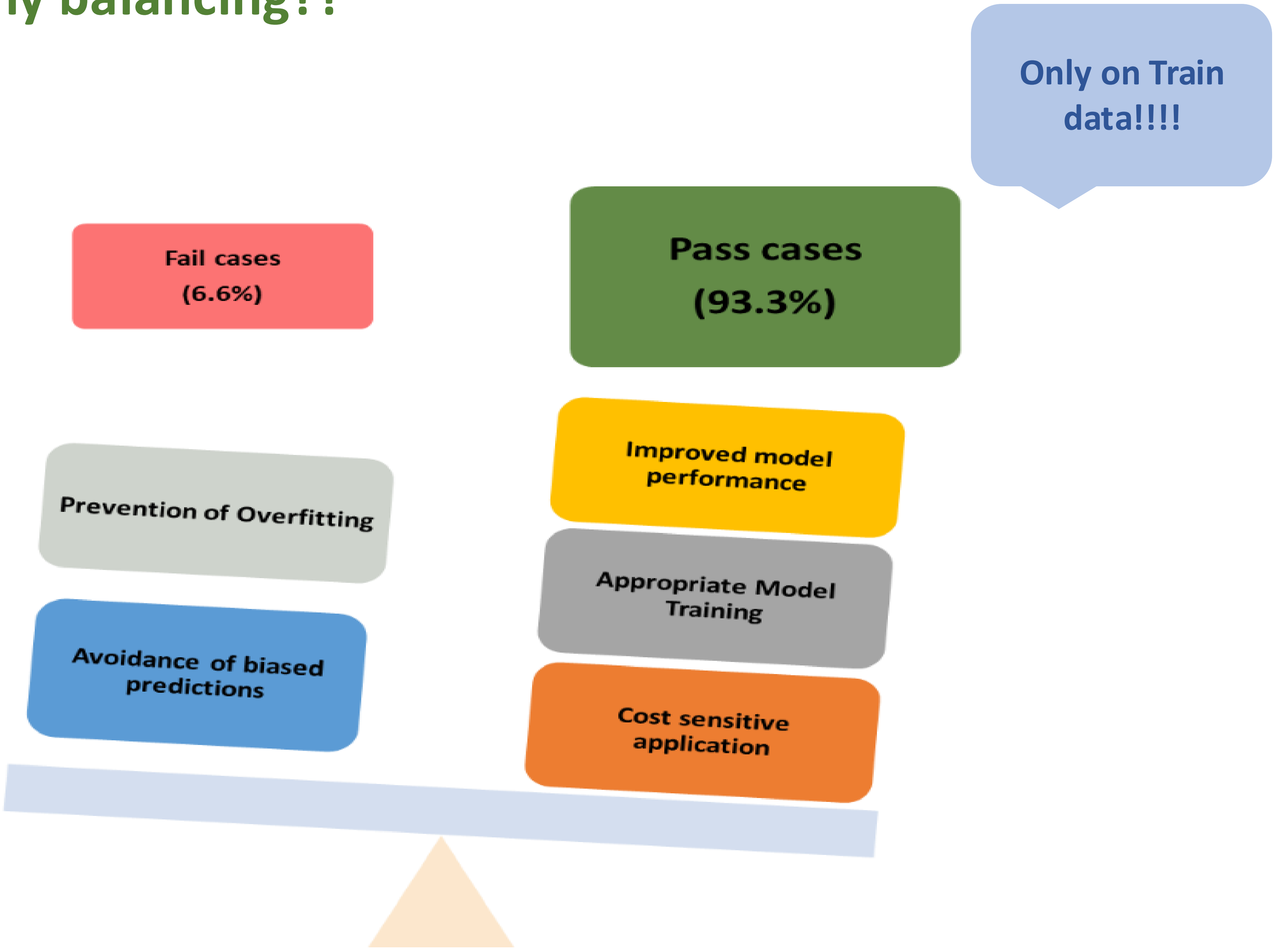
```
Sorted Boruta feature rankings:
feature60: 1
feature65: 1
feature66: 1
feature342: 1
feature351: 1
feature478: 1
feature540: 1
feature563: 1
feature157: 2
feature268: 2
feature292: 2
feature427: 2
feature430: 2
feature154: 3
feature206: 4
feature153: 5
feature426: 6
feature171: 7
```

Why Boruta??

- Improve model performance by using **Random forest** approach on **original and shadow features,** making it capable of capturing complex relationships.
- Prevent the loss of important information - as evaluated by **GINI importance**
- Used for **supervised data**
- Boruta identifies and ranks the **features which are important for predicting the target variable**.
- Boruta can handle multicollinearity and **non-linear relationships effectively**.

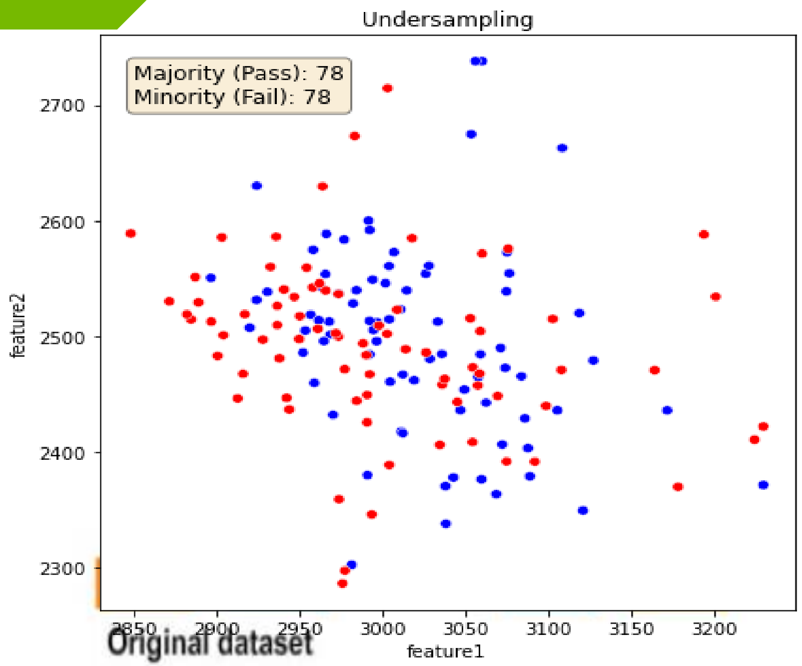
This makes Boruta a powerful tool for feature selection, especially in datasets with complex interactions and relationships among features.

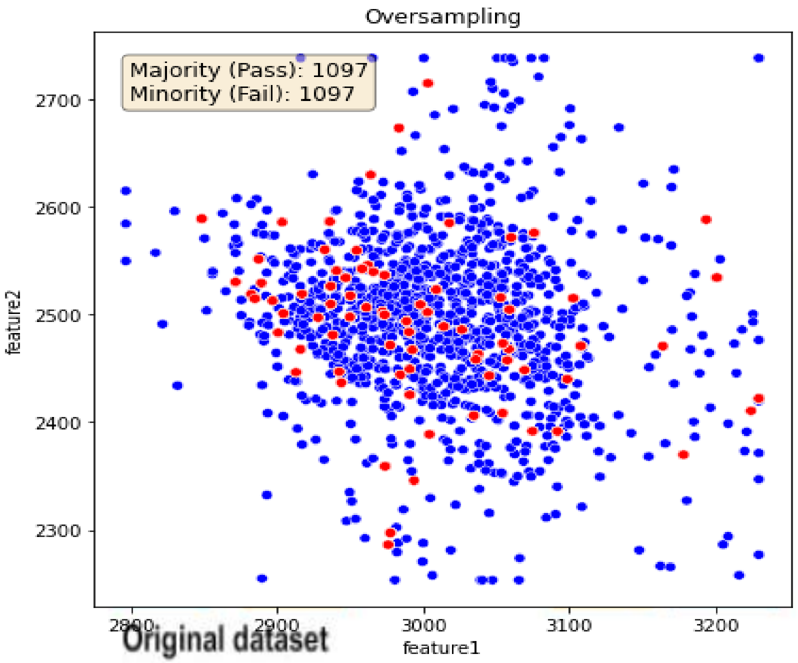
Why balancing??



Master
Project Management
and Data Science

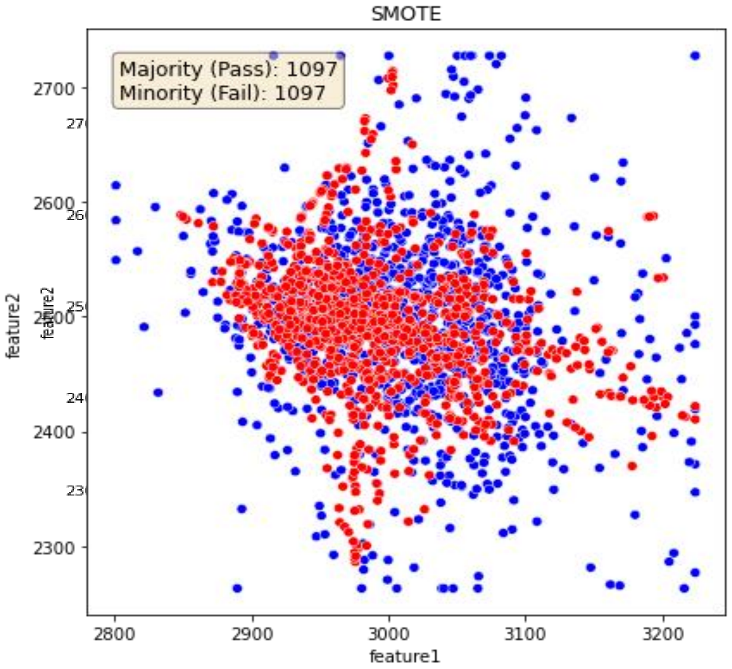
Balancing and Resampling





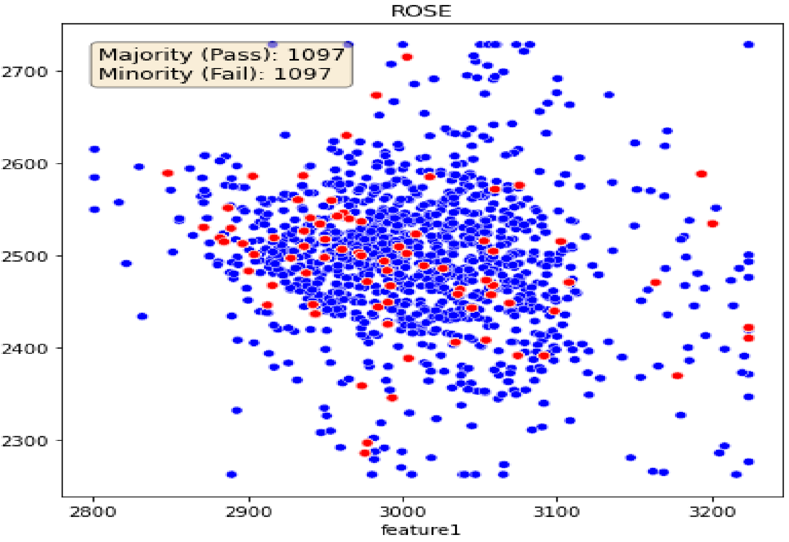
Name	Approach	Pros	Cons	Effect	Decision
Over-Sampling	Duplicates minority class instances to balance the dataset	Simple to implement, effective	High risk of **overfitting**	Accuracy may be good but does not replicate real world data	**Creates duplicates**
Under-Sampling	Removes instances from the majority class to balance the dataset	Reduces dataset size, computationally efficient	Can lose important information, can lead to **underfitting**	Accuracy may be good but does not replicate real world data.	**Loss of important information.**

Approaches to deal with imbalanced data



Name	Approach	Pros	Cons	Effect	Decision
SMOTE	Generates synthetic sa… bet… ori… nea… (un…	Uniformity	No adaptive	Best results	Loss cost is low High Priority

Highly imbalanced data
Fail cases – 6.6%
Pass cases – 93.4%



Name	Approach	Pros	Cons	Effect	Decision
ROSE	- Generates new synthetic data points by **adding random noise** to existing data points within the minority class - Smoothed bootstrapped approach.	Reduce the risk of overfitting compared to duplication attempts to maintain the underlying distribution of the data.	can introduce noise, **Require parameter tuning**	Reduce bias	Medium Priority

Decision Hierarchy

Threshold (Loop 45% to 65%)

- 60% or 65%
- 50%-60% no missing values
- 45% too low

Outlier

- Remove
  Maintain distribution of data
- Replace
  Alter skewness drastically – presence is higher
- No Change
  Influence of extreme values, misleading solutions

Imputation

- KNN
  Scaling and less volatility difference
- MICE
  First replace the NA with mean and replace with better ones
  Average change in volatilty
- Mean
  Decreace in volatility
- Median
  Decreace in volatility

Feature Selection/ Reduction

- BORUTA
  Supervised learning
  Multicollinearity and nonlinear relationships
- PCA
  Unsupervised learning
  Requires linear relationship

Balancing

- SMOTE
  Reduce overfitting
- ROSE
  Maintain a balanced distribution with less obvious clustering, requires parametertuning
- ADASYN
  Balanced data set but possible overlap of classes, noice in data
- Over Sampling
  Generates duplicates
- Under Sampling
  Risk of loss of important information

High
Medium
Low

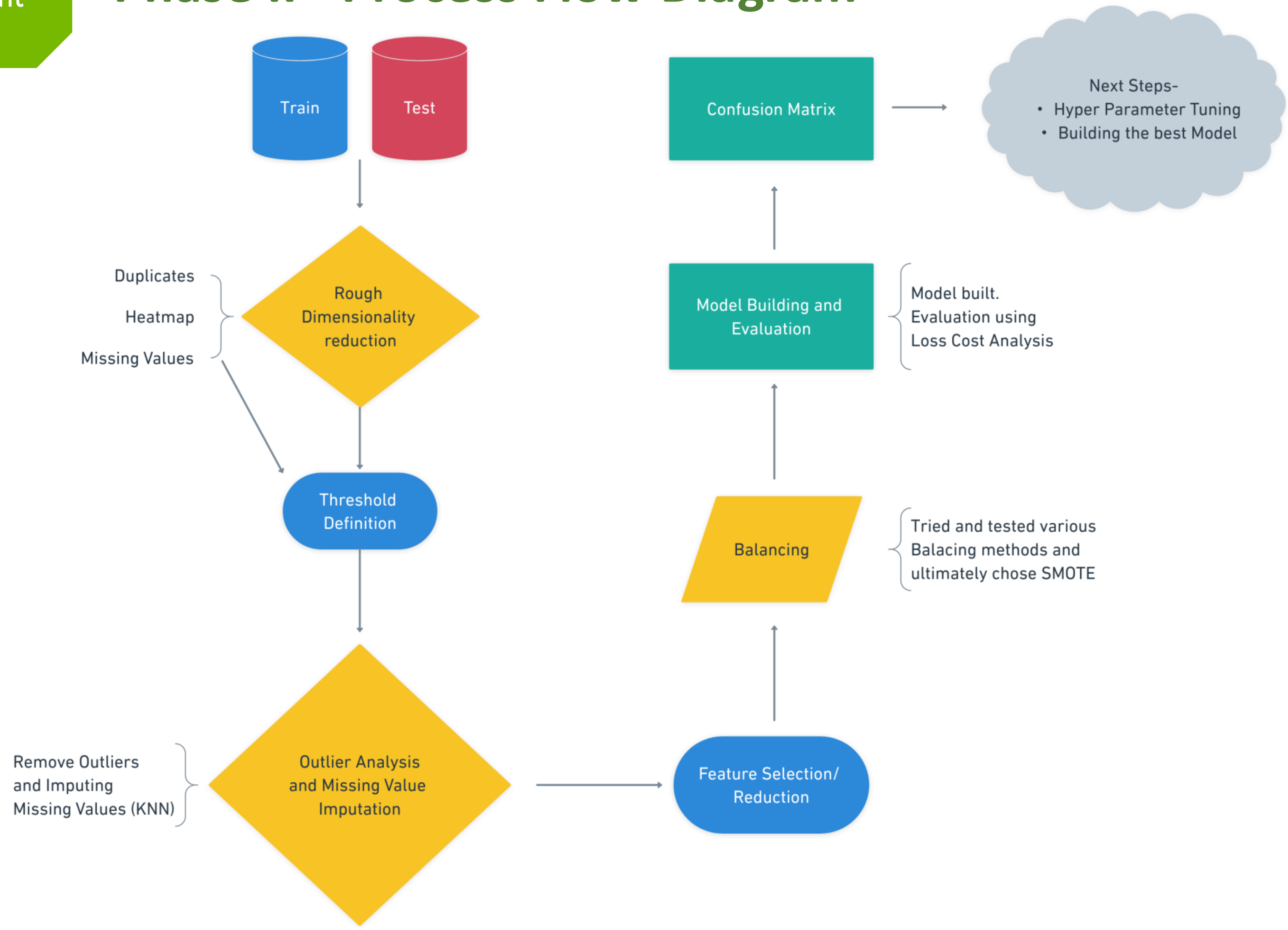
Model building

	Decision						EVALUATION											
Name	Decision in CRISP DM	Threshold	Outliers	Impute method	Feature Selection/ Feature Reduction	Balancing method	Train error	Test Error	Accuracy	Confusion_matrix				Loss_cost FP- 1000 FN- 5000	Precision	Recall	f1_score	AUC
										TP	FP	FN	TN					
Model 1	Data preparation- Rough feature reduction, Outlier Analysis, Missing value Imputation, Feature selection Data Modeling - Balancing and Resampling, Model building	65	Remove & Impute	KNN	Boruta	SMOTE	0	0.09	0.90	351	15	22	4	125000	0.210526	0.153846	0.177778	0.556431
Customized Model	Feed No. 1 features to build the model by Boruta ranking	65	Remove & Impute	KNN	Boruta - feature60, feature65, feature66, feature342, feature351, feature478, feature540, feature563	SMOTE	0	0.11	0.90	348	18	20	6	124000	0.2	0.230769 2	0.2142857 1	0.562683

Take Home Messages

- **Outliers** - 3s boundaries may sometimes change the entire characteristics of the distribution, and hence, we performed KNN.
- For KNN, **Scaling the data is important**, as it's is a distance-based approach, otherwise, the results will be misleading.
- **Highly imbalanced dataset** - To make sure that our model in not biased towards majority class, we need to balance the dataset. Models trained on imbalanced data might have a high accuracy but give misleading evaluation of results.
- For highly imbalanced data, **Random Forrest** may be a good option. It combines multiple decision trees to prevent the model from overfitting.
- **Model Evaluation** - Accuracy cannot be an ultimate criteria to judge the quality of a model, we need to do Loss cost Analysis.

Phase II - Process Flow Diagram



Vielen Dank für Ihre Aufmerksamkeit !

Master
Project Management
and Data Science

19/07/2024

PROF. DR. TILO WENDLER

SECOM Case Study

Dhruvi Jay Patel (s0592755)

Naman Mathpal (s0590500)

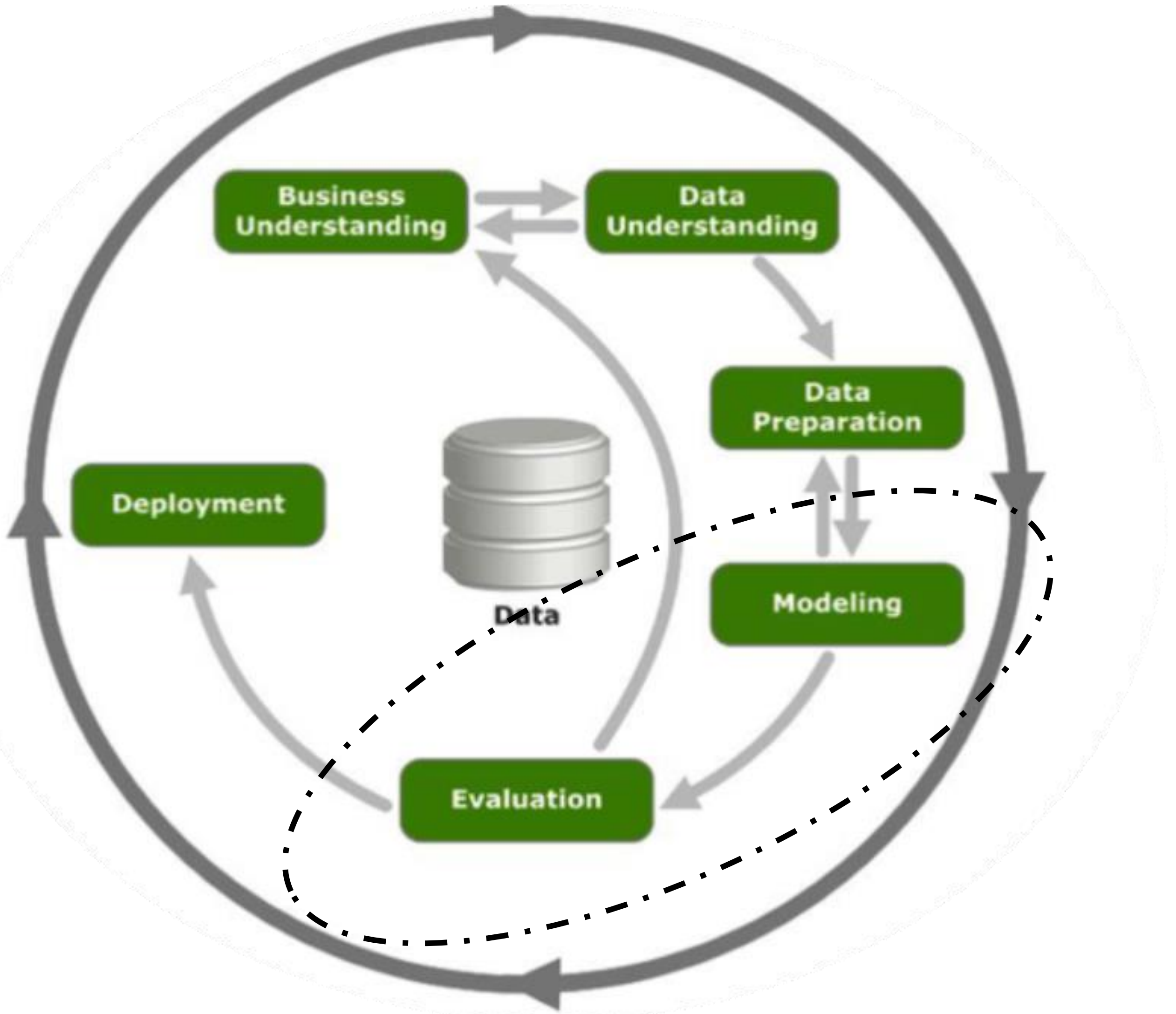
Bhoomika Jagadeesha (s0590573)

Jui Prasad Kulkarni (s0590496)

Ankit Satish Gupta (s0590516)

INDEX

CRoss Industry Standard Process for Data Mining

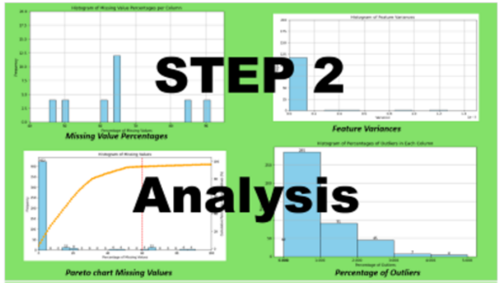


Best Model

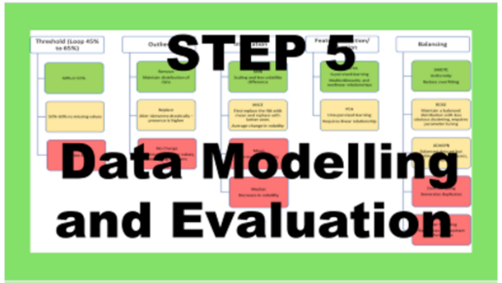
The
~~End~~
Beginning

Model	F1 Score	Loss Cost	FP Type I	FN-Type II	Accuracy
Final	0.893	114000	29	17	89.08%

Steps of Model Building Process



Steps of Model Building Process



STEP 6

Scaling

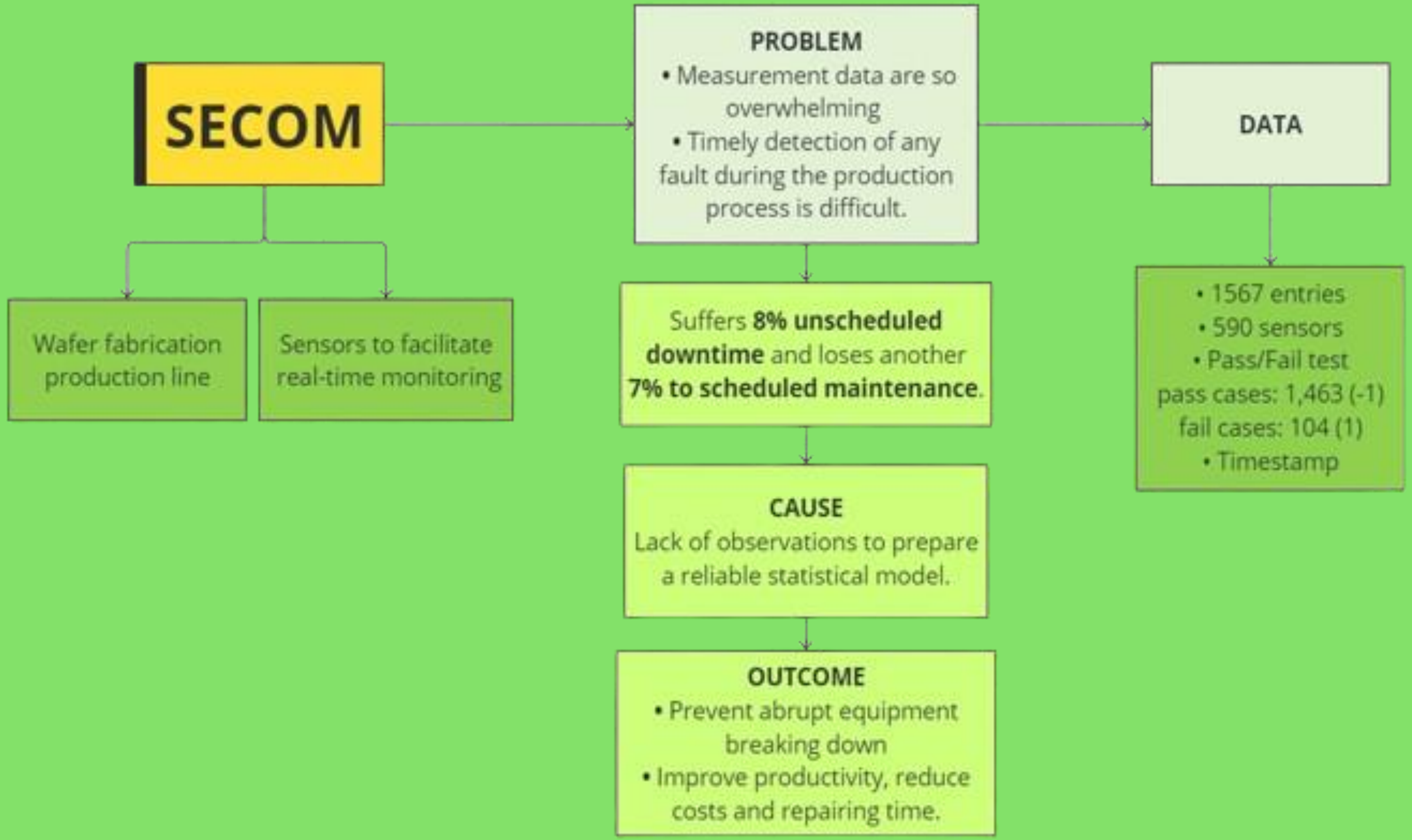
STEP 7

Hyper Parameter Tuning

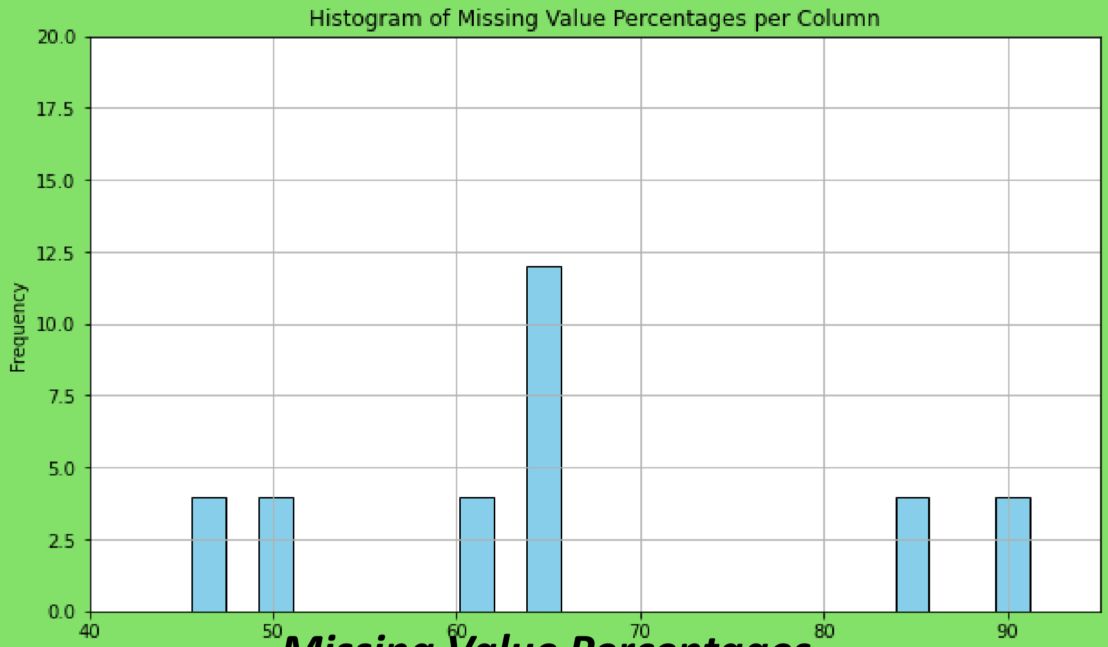
STEP 8

K Fold Cross Validation

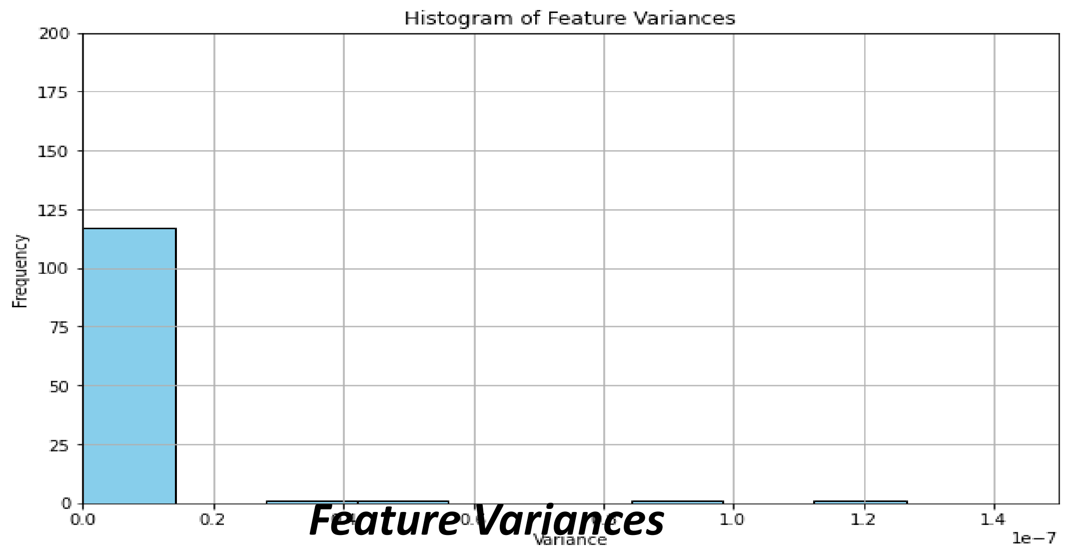
STEP 1 Business/Data Understanding



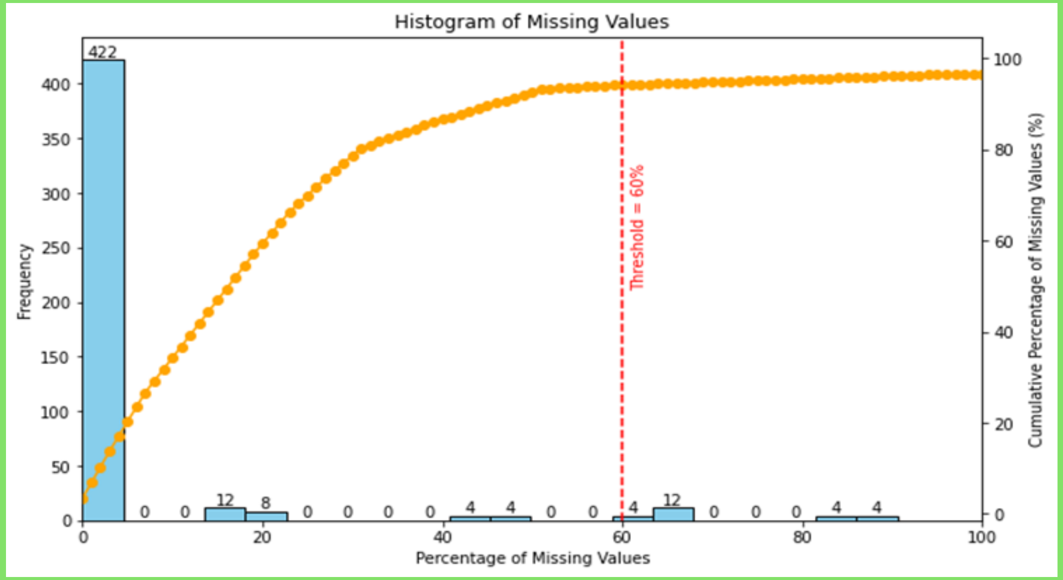
STEP 2 Analysis



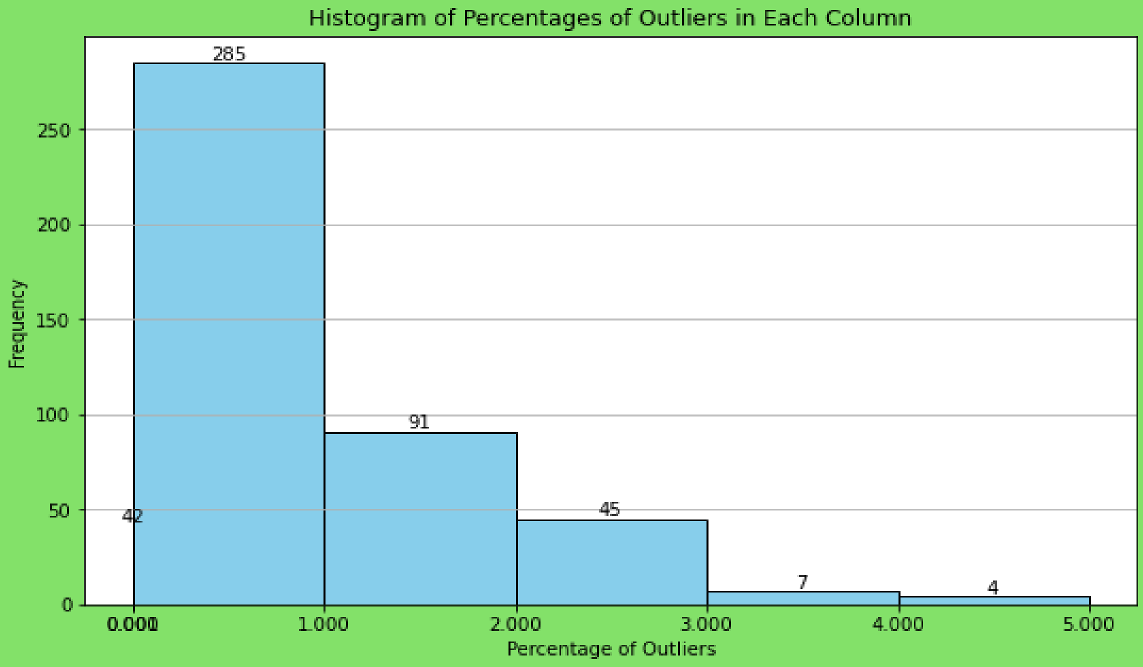
Missing Value Percentages



Feature Variances

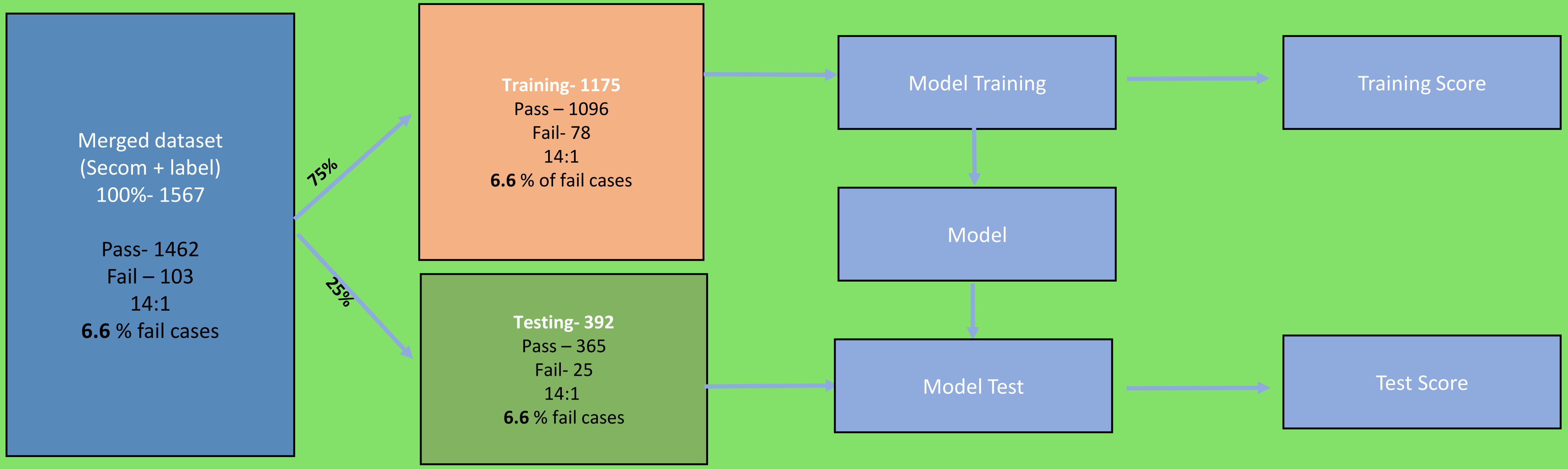


Pareto chart Missing Values

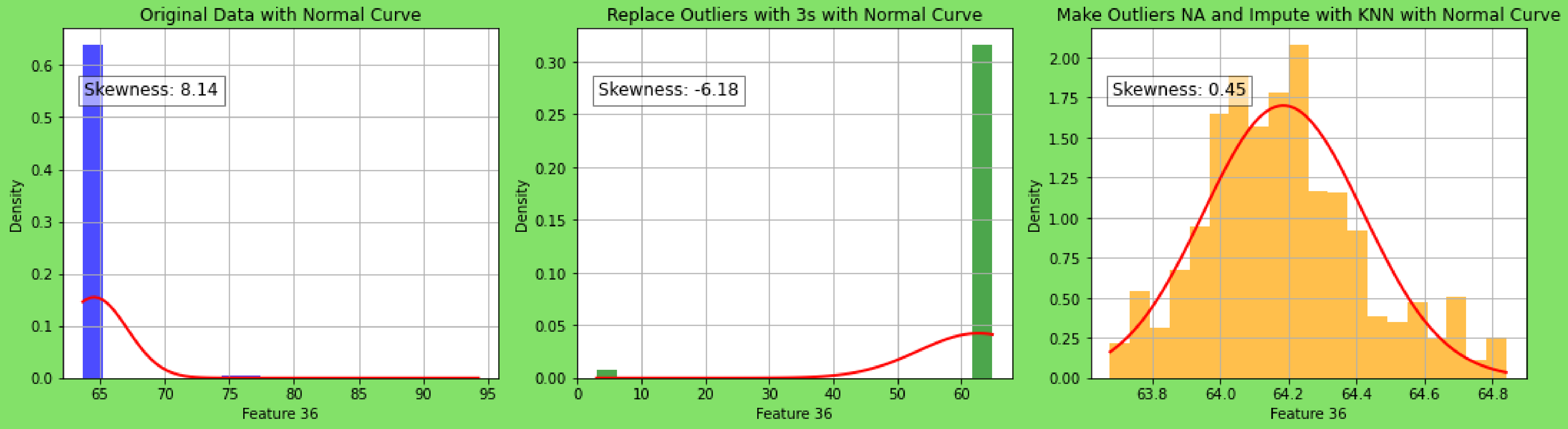


Percentage of Outliers

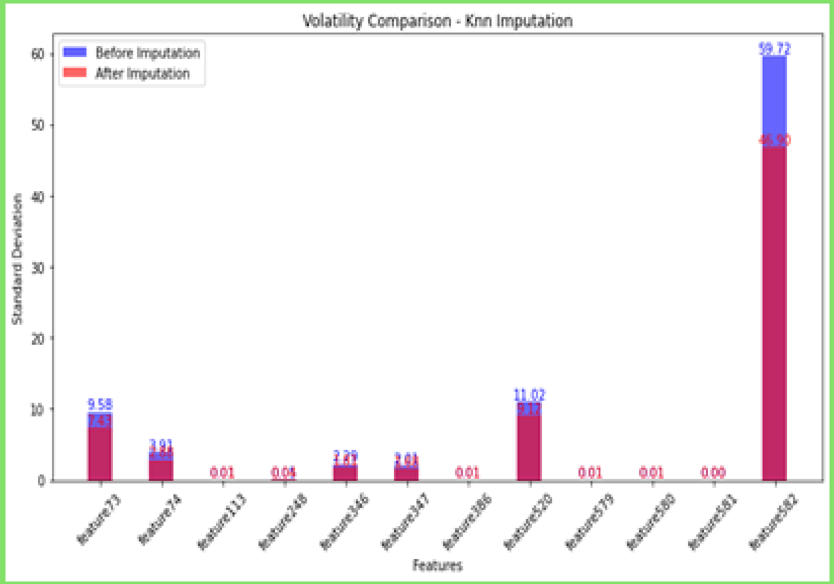
STEP 3 Data Splitting

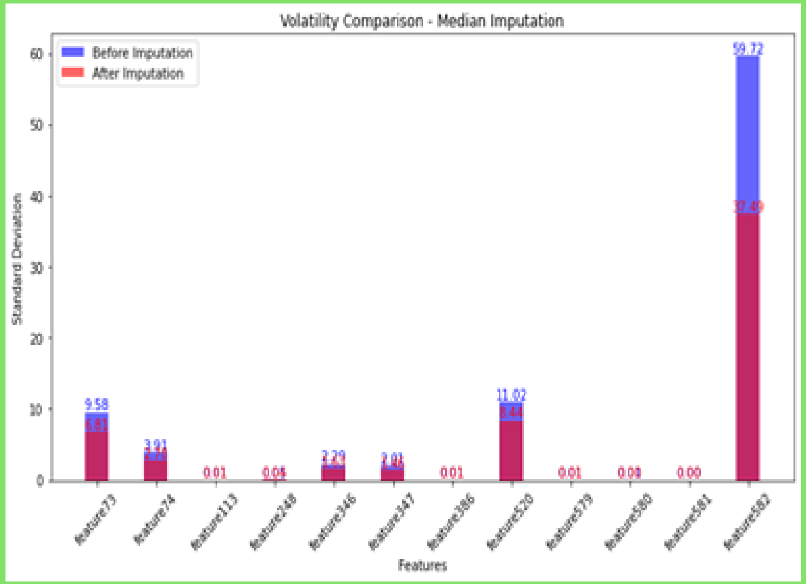


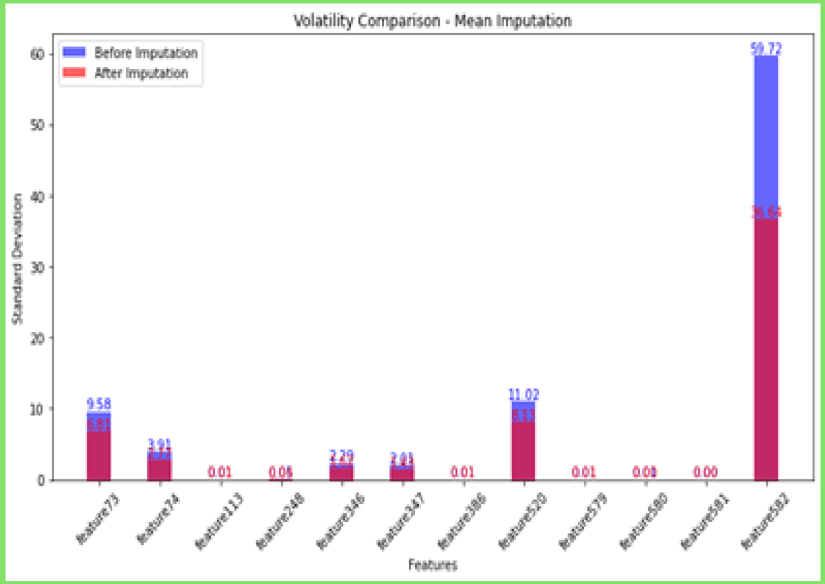
STEP 4 Data Preparation

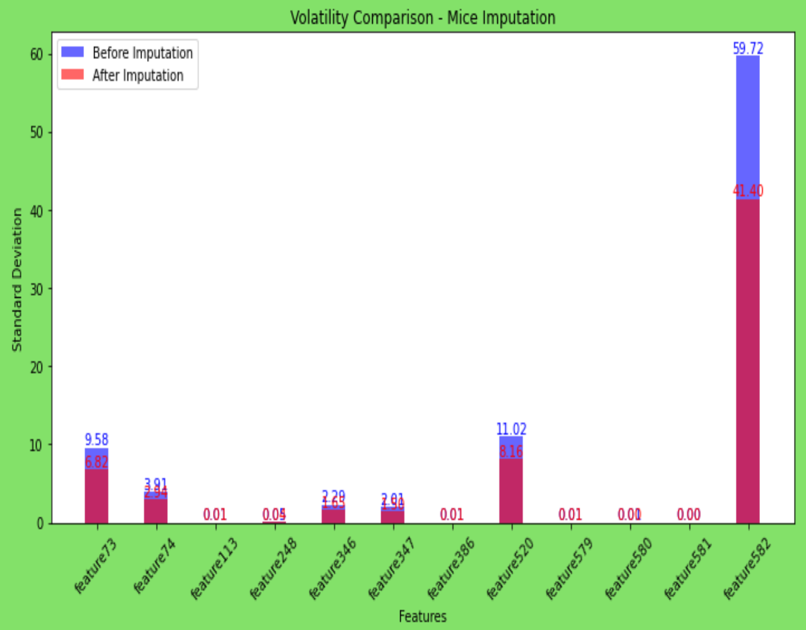


STEP 4 Data Preparation









STEP 5 Data Modelling and Evaluation

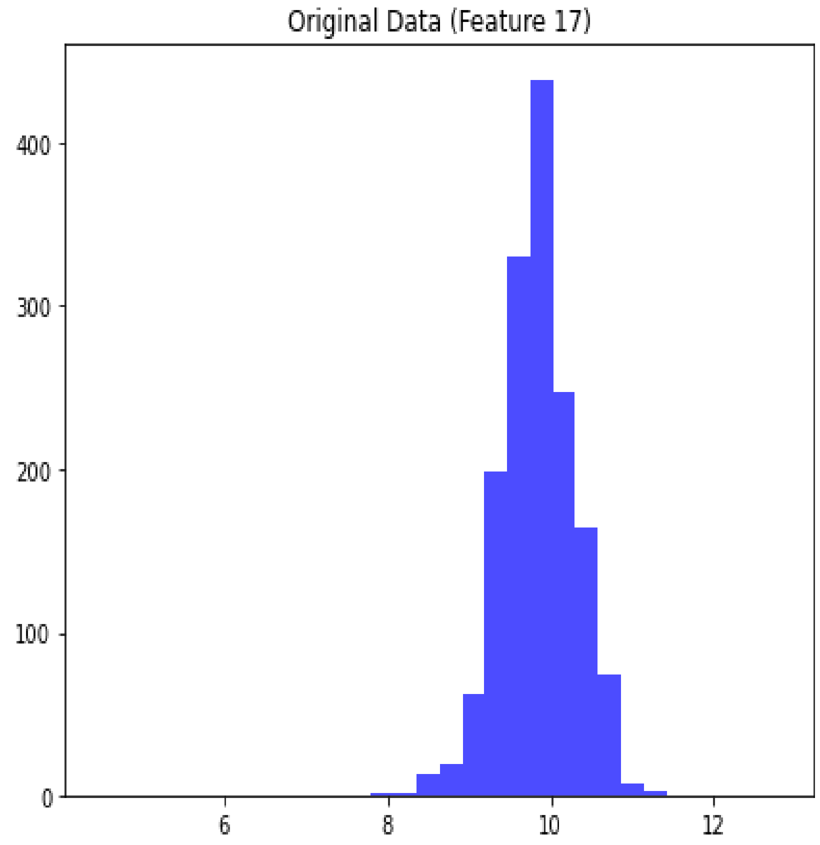
Name	Decisions in CRISP DM	Model approach	Model assessment

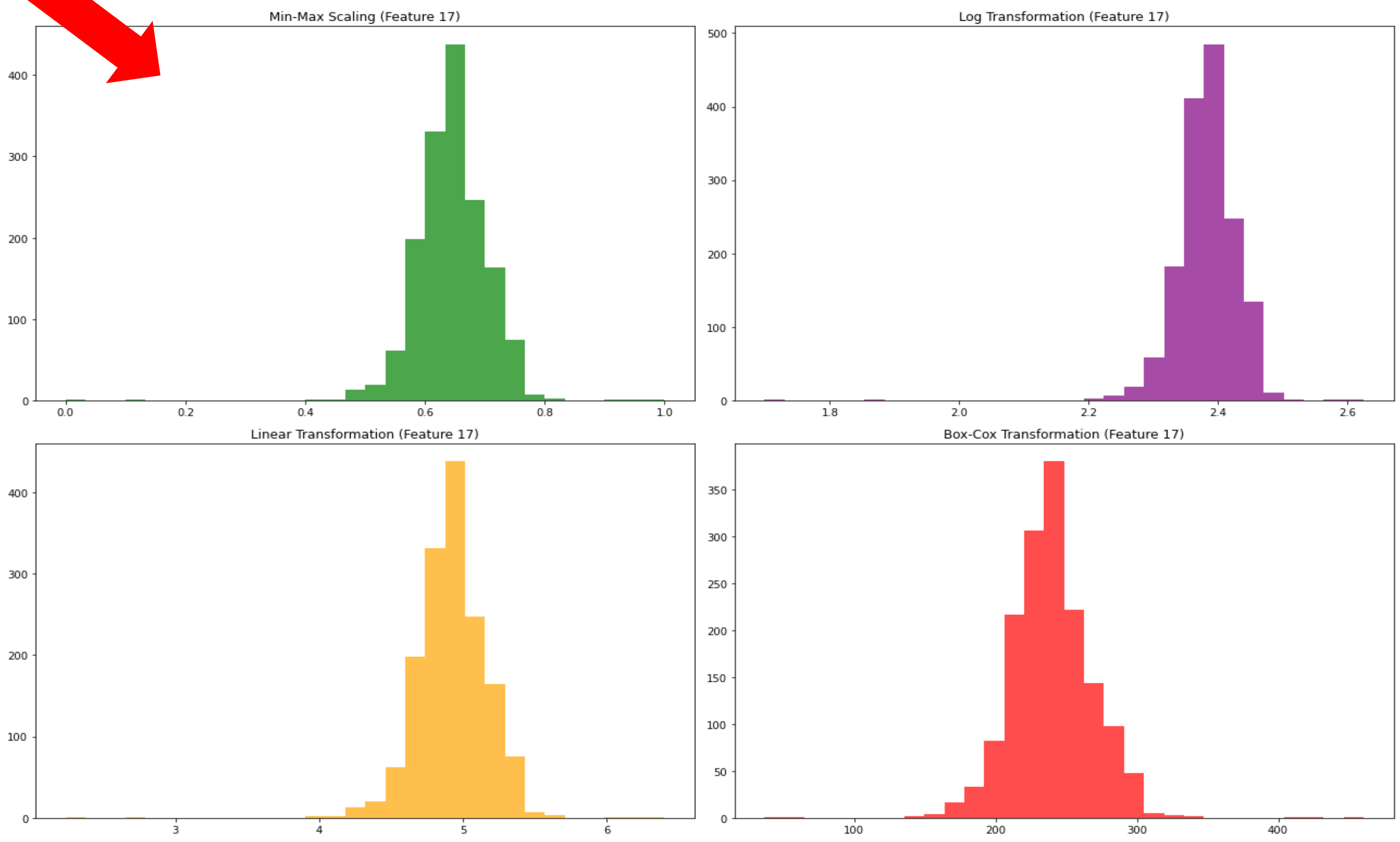
- **Threshold (Loop 45% to 65%)**
  - 60% or 65%
  - 50%-60% no missing values
  - 45% too low
- **Outlier**
  - Remove
    Maintain distribution of data
  - Replace
    Alter skewness drastically – presence is higher
  - No Change
    Influence of extreme values, misleading solutions
- **Imputation**
  - KNN
    Scaling and less volatility difference
  - MICE
    First replace the NA with mean and replace with better ones
    Average change in volatilty
  - Mean
    Decreace in volatility
  - Median
    Decreace in volatility
- **Feature Selection/ Reduction**
  - BORUTA
    Supervised learning
    Multicollinearity and nonlinear relationships
  - PCA
    Unsupervised learning
    Requires linear relationship
- **Balancing**
  - SMOTE
    Uniformity
    Reduce overfitting
  - ROSE
    Maintain a balanced distribution with less obvious clustering, requires parameter tuning
  - ADASYN
    Balanced data set but possible overlap of classes, noice in data
  - Over Sampling
    Generates duplicates
  - Under Sampling
    Risk of loss of important information

STEP 5 Data Modelling and Evaluation

Name	Decisions in CRISP DM	Model approach	Model assessment											
			Accuracy	Train error	Test error	TP	FP	FN	TN	Loss cost	Precision	Recall	F1_score	AUC
Model 1	**Data preparation**-Rough feature reduction, Outlier Analysis, Missing value Imputation, Feature selection **Data Modeling** - Balancing and Resampling, Model building	**65%** threshold, replace **outliers with 3s**, **KNN** Imputation, Boruta, **SMOTE** balancing, Random forest	0.91	0	0.09	351	15	22	4	125000	0.21	0.153846	0.177778	0.556431
Model 2	**Data preparation**-Rough feature reduction, Outlier Analysis, Missing value Imputation, Feature selection **Data Modeling** - Balancing and Resampling, Model building	**45%** threshold, **remove** outliers, **KNN** Imputation, Boruta, **ROSE** balancing, Random forest	0.93			365	1	25	1	126000	0.5	0.038462	0.071429	0.517865
Model 3	**Data preparation**-Rough feature reduction, Outlier Analysis, Missing value Imputation, Feature reduction **Data Modeling** - Balancing and Resampling, Model building	**50%** threshold, **replace outliers with 3s**, **MICE** Imputation, PCA, **SMOTE** balancing, Random forest	0.93			363	3	25	1	128000	0.25	0.038462	0.066667	0.515132
Customized Model	Feed No.1 features to build the model by Boruta ranking	65% threshold, replace outliers with 3s boundaries, KNN, No.1 features by BORUTA, SMOTE, Random forest	0.90	0	0.11	348	18	20	6	124000	0.2	0.2307692	0.21428571	0.562683

Scaling



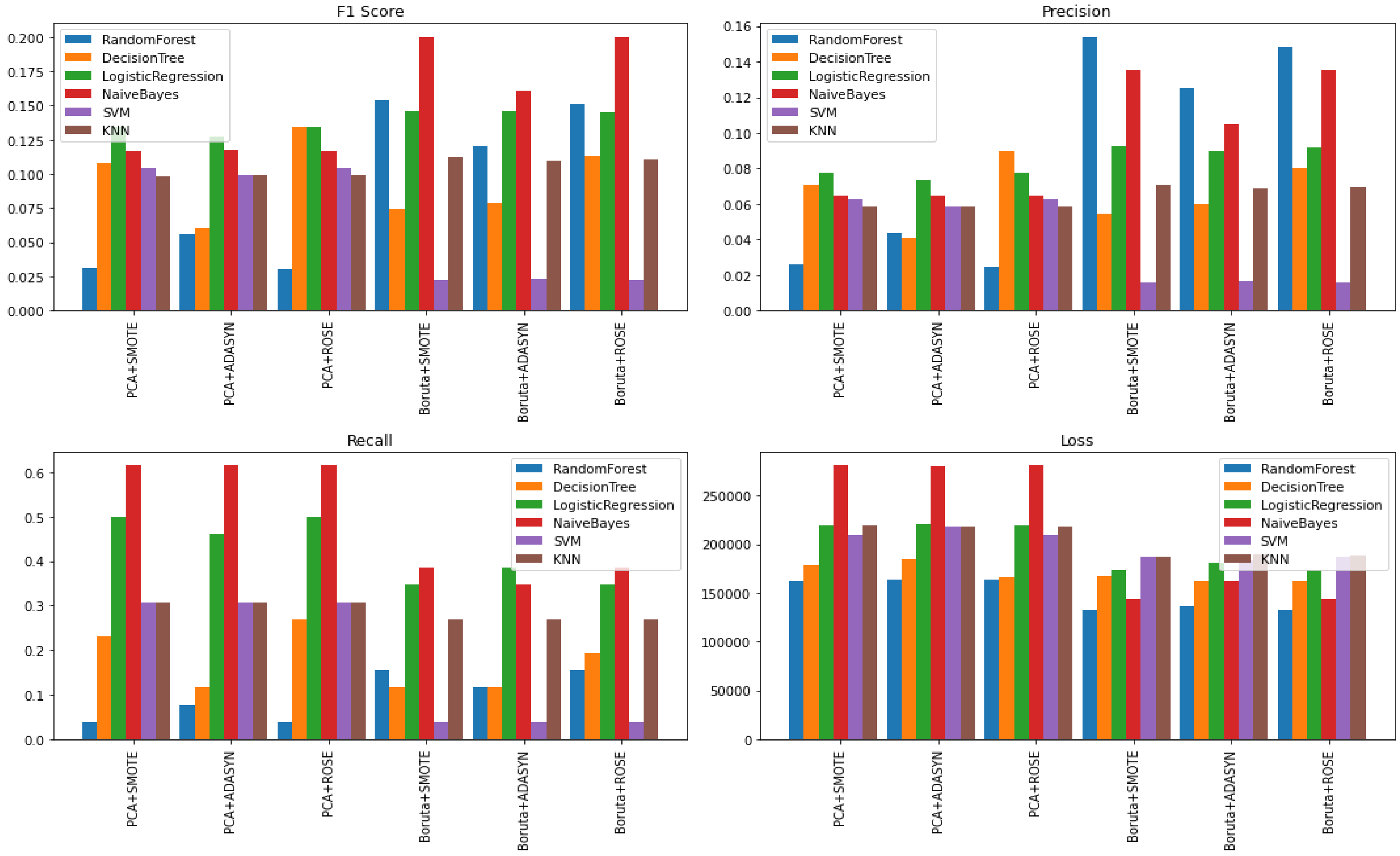


1. Scaling ensures uniformity, improves performance of algorithms, and reduces biases.
2. Features with higher ranges are more likely to be chosen by the model.
3. Since we use SVM and KNN which is sensitive to the scale of features, we choose the min-max scaling method.

Optimal Parameters

Boruta+smote highest F1 scores, precisions and lowest Loss

Majorly scores are highest for **RF, NB and SVM**



Hyperparameter tuning

Hyperparameter tuning

What

Discovering the most suitable combination of hyperparameters for a machine learning model

How

1. **Grid search**
2. Random search
3. Genetic algorithms
4. Hyperband

Why

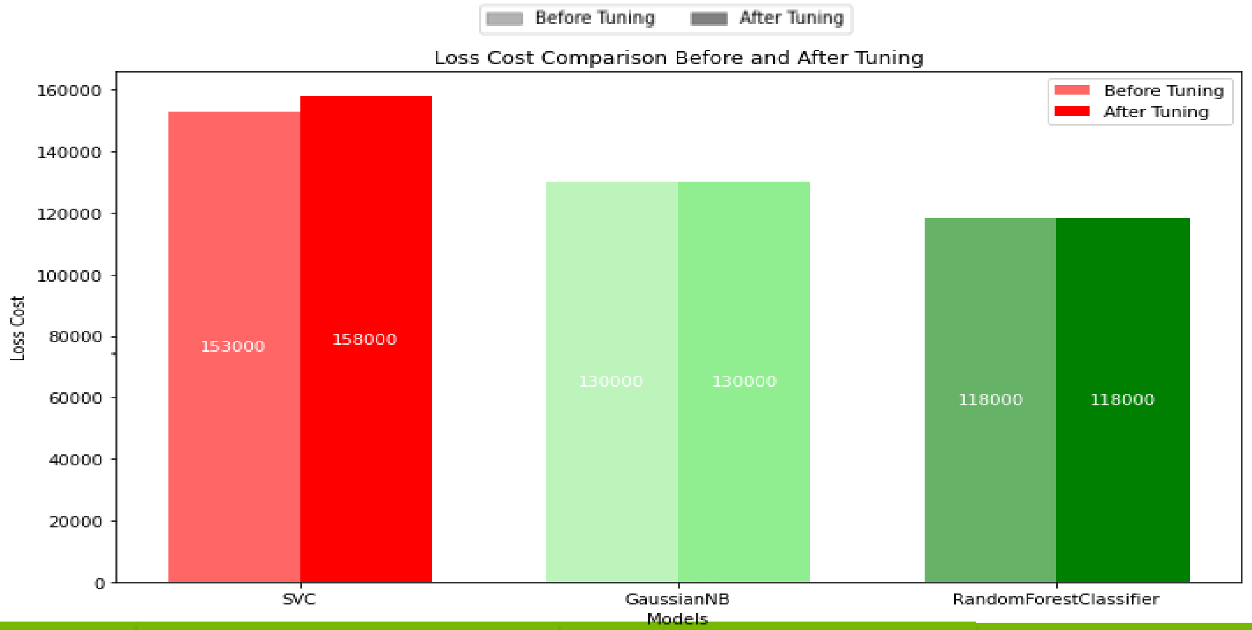
1. Automation and scalability
2. Process optimization
3. Quality control
4. Increased Efficiency

Master
Project Management
and Data Science

Hyperparameter Tuning

Model building steps → 60% → Outliers remove → KNN → Boruta → SMOTE → Min max scalling

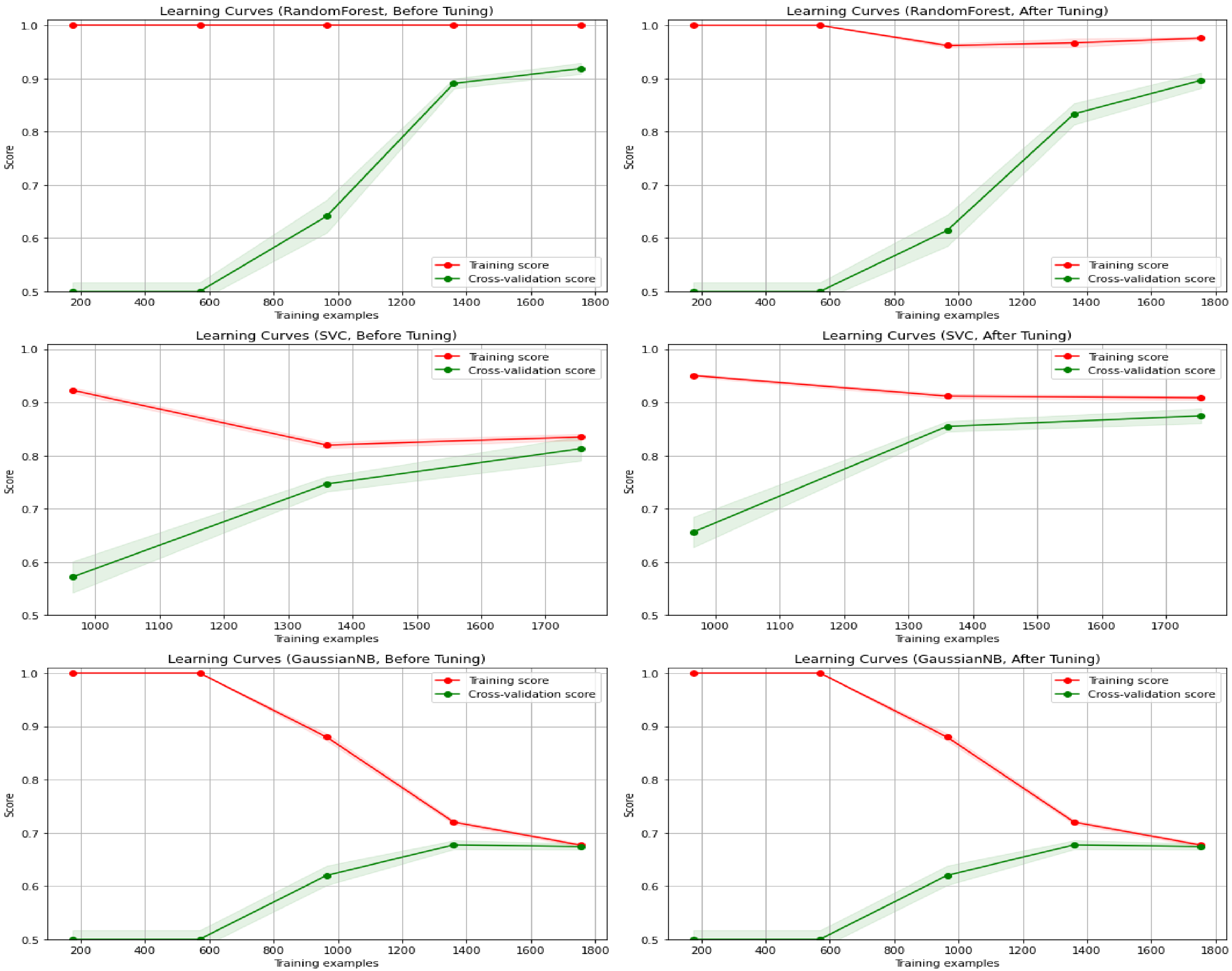
FP cost – 1000
FN cost - 5000



Models	Precision comparison		Recall comparison		F1 score comparison		Accuracy comparison		Loss Cost comparison	
	Before	After	Before	After	Before	After	Before	After	Before	After
SVC	**0.11**	**0.08**	**0.31**	**0.15**	**0.16**	**0.10**	**0.79**	**0.82**	**153000**	**158000**
GaussianNB	0.17	0.17	0.58	0.58	0.26	0.26	0.78	0.78	130000	130000
Random forest	0.25	0.25	0.23	0.23	0.24	0.24	0.90	0.90	118000	118000

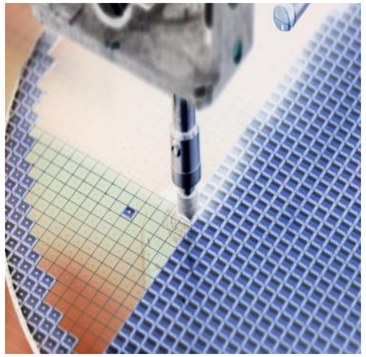
Larger gap between scores for Random Forest.

Though for SVC and NB models the gap is less the **training and validation score is less**



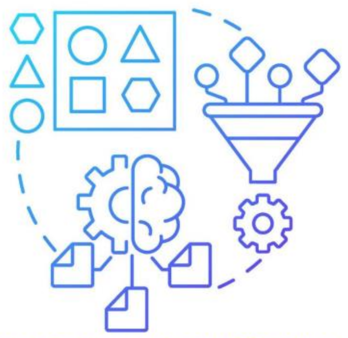
Feature Engineering

- Intervals between each wafer production
- Can monitor production flow



Feature592 (Timestamp)

19/07/2008 11:55:00
19/07/2008 12:32:00
19/07/2008 13:17:00
19/07/2008 14:43:00
19/07/2008 15:22:00
.
.
.
.



New feature

elapsed_time

0	0
37	Occured after 37 minutes
82	Occured after 82 minutes
168	Occured after 168 minutes
207	Occured after 207 minutes
.	.
.	.
.	.
.	.

IMPACT

```
Selected Features:  ['feature1', 'feature34', 'feature60', 'feature66', 'feature104', 'feature130', 'feature131', 'feature511',
'elapsed_time']
```

Master
Project Management
and Data Science

Evaluation

Model 1: 60% → Outliers remove → KNN → Boruta (after feature eng.) → SMOTE → Min max scalling → Random forest

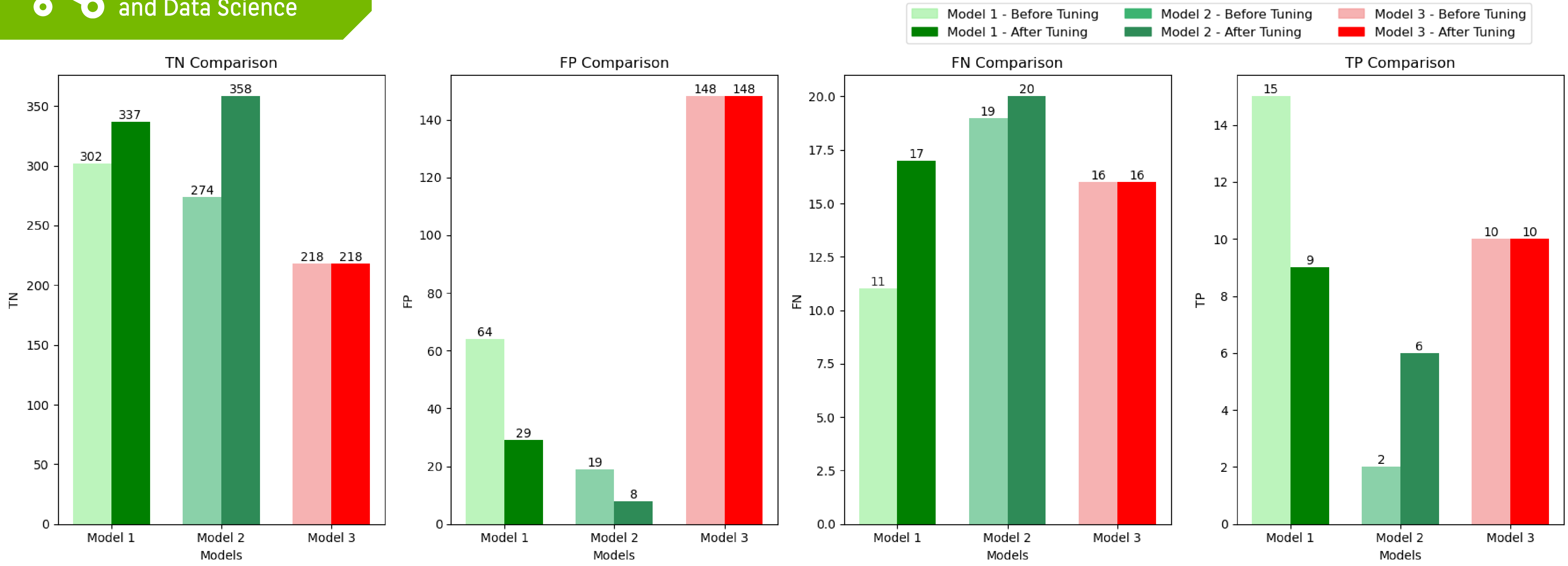
Model 2: 60% → Outlier removal → KNN → Top features by boruta → SMOTE → Min max scalling → Random forest

Model 3: 60 % → Outlier removal → KNN → Top features by boruta → SMOTE → Min max scalling → Naive bayse

FP cost – 1000
FN cost - 5000

Model	F1 Score Comaprison		Lost cost comparison		TN comparison		FP comparison		FN comparison		TP comparison	
	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
Model1	0.849	0.893	119,000	114,000	302	337	64	29	11	17	15	9
Model2	0.255	0.918	115,000	108,000	274	358	19	8	19	20	2	6
Model3	0.686	0.686	228,000	228,000	218	218	148	148	16	16	10	10

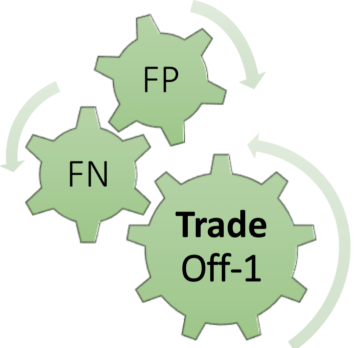
Confusion Matrix



Model 2: Prioritizes economic loss minimization but may compromise on balance

Model 1: Offers balanced performance

Model 3: Maximizes true positives, suitable where the high cost of false positives is acceptable



K Fold cross validation

Loss cost Model 1 After tuning	Loss cost model 2 After tuning
114000	108,000

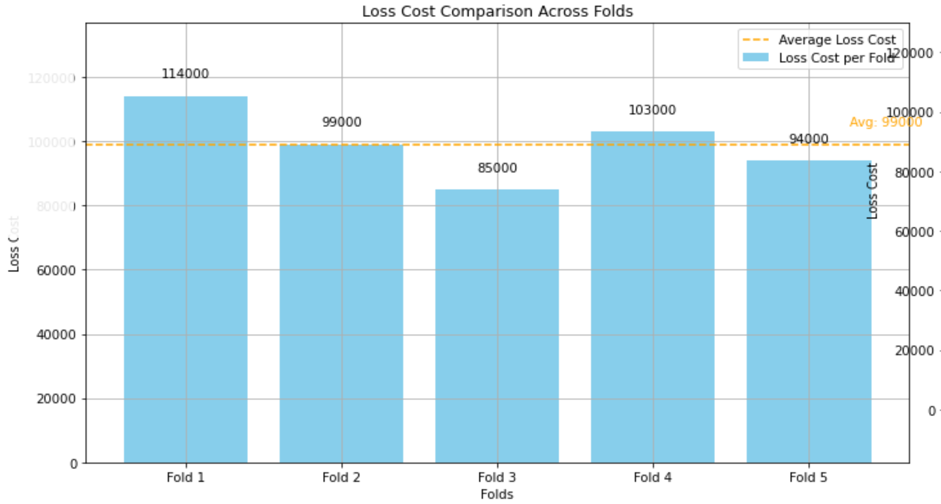
K fold Stratified Validation

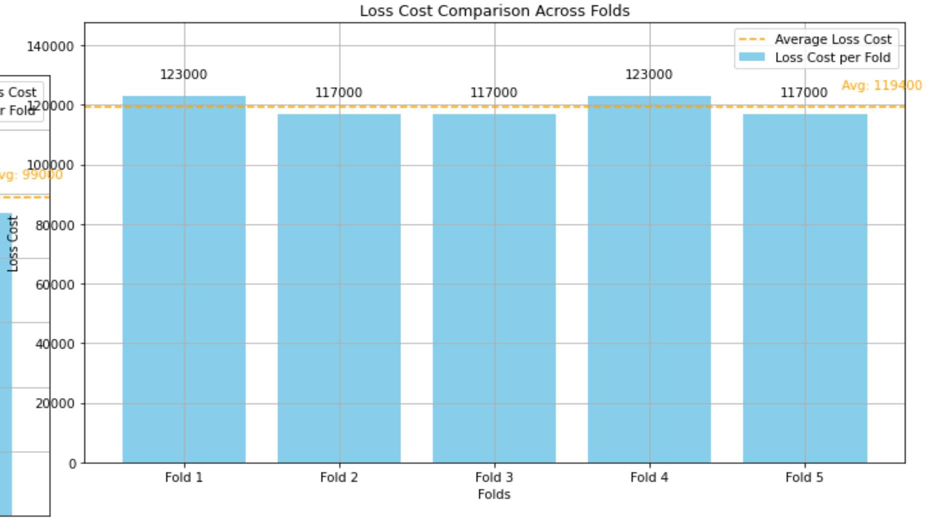
	Model 1	Model 2
Fold 1	114000	123000
Fold 2	99000	117000
Fold 3	85000	117000
Fold 4	103000	123000
Fold 5	94000	117000
Average	**99000**	**119400**

Why

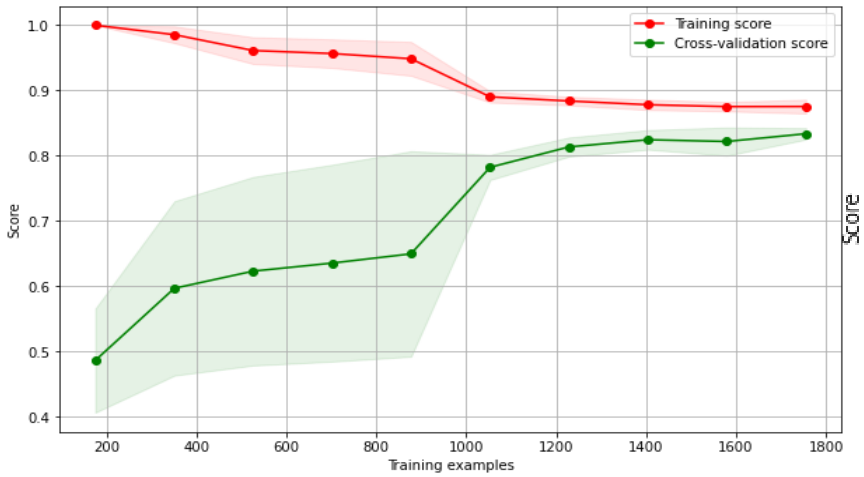
Impact

- Eva

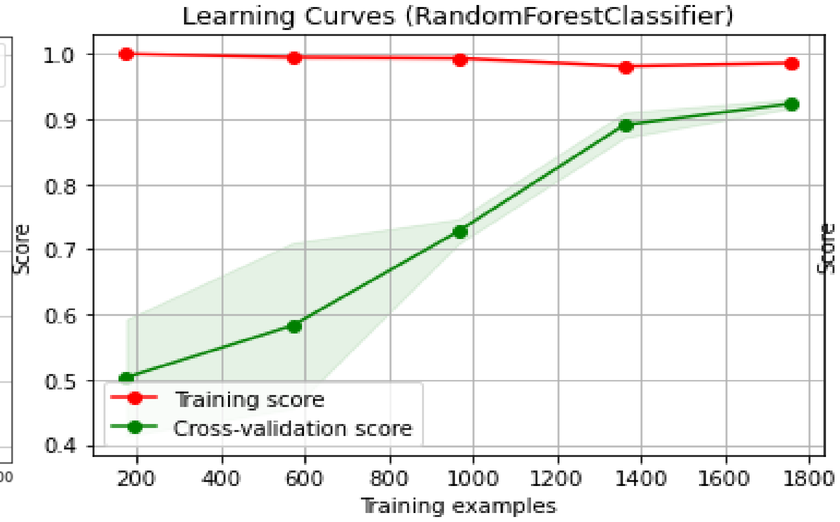




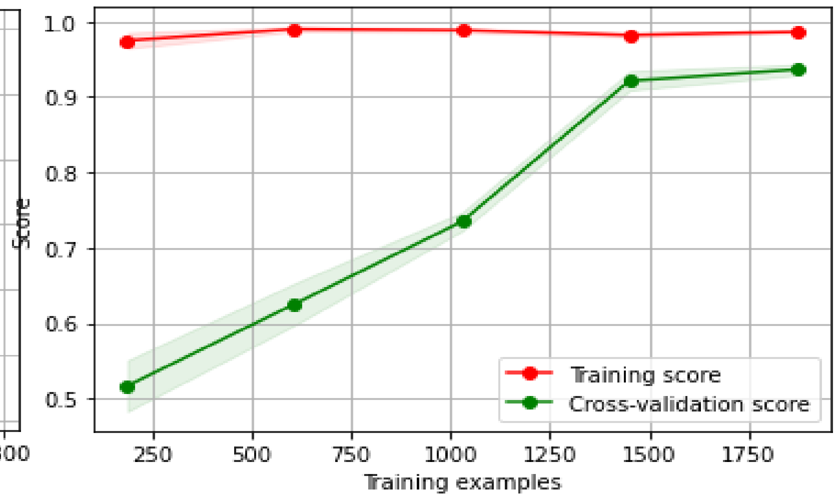
Learning Curve



Before Tuning



After Tuning



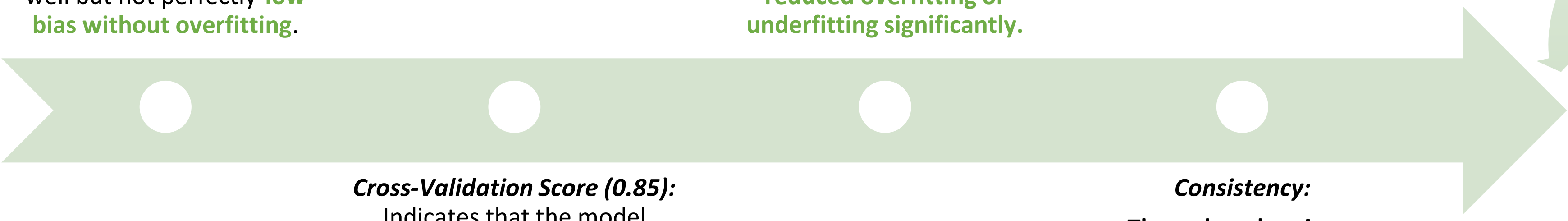
After Kfold

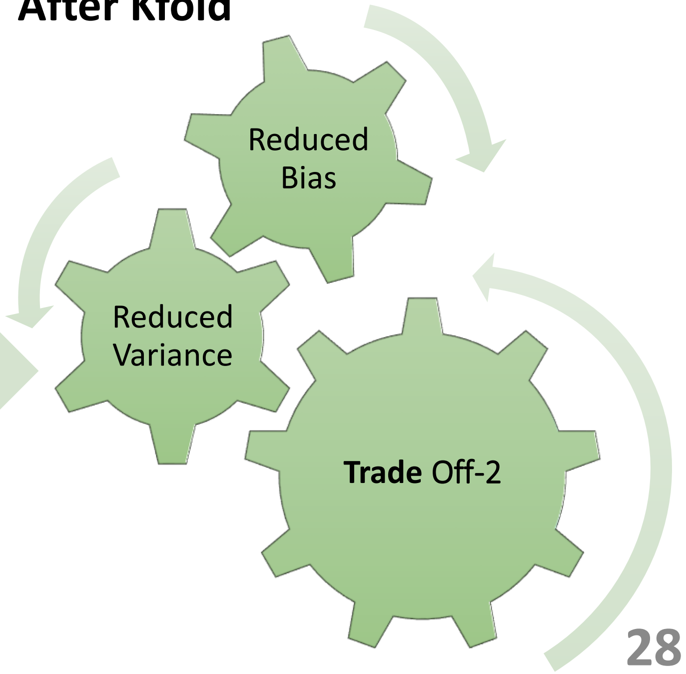
Training Score (0.95):
Indicates that the model fits the training data very well but not perfectly-**low bias without overfitting**.

Cross-Validation Score (0.85):
Indicates that the model generalizes well to unseen data, **suggesting reduced variance.**

Reduced Gap Between Scores:
A good bias-variance tradeoff- **reduced overfitting or underfitting significantly.**

Consistency:
The reduced variance around the cross-validation score line indicates more **consistent performance**.





Key Takeaways

- **Scaling is Required**: Ensures equal range of features in distance-based algorithms.
- **Iterative Nature**: CRISP-DM methodology facilitated continuous model improvement.
- **Grid Search**: Systematically optimized hyperparameters for best performance.
- **Different Models Tested**: SVM didn‘t performed well; Random Forest had best loss cost hence economically reliable; Naïve Bayes excelled in true positives.
- **Feature Engineering**: Crucial for enhancing model performance after business understanding.
- **K-Fold Cross Validation**: Provided reliable performance estimation and maximized data usage.
- **Learning Curve Analysis:** Showed the impact of hyperparameters on model performance.
- **Loss Cost:** Ideal for minimizing economic loss in priority scenarios.Its the trade of point.

Conclusion

- **High Business Risk**: The cost of labeling a faulty chip as good is significantly higher than labeling a good chip as faulty.
- **Data Treatment**: Handling zero variance, outliers, and skewed data is crucial in model building.
- **Beyond Accuracy**: Accuracy alone is insufficient; loss cost analysis and volatility are critical for evaluating model performance in high-risk scenarios.

Vielen Dank für Ihre Aufmerksamkeit !