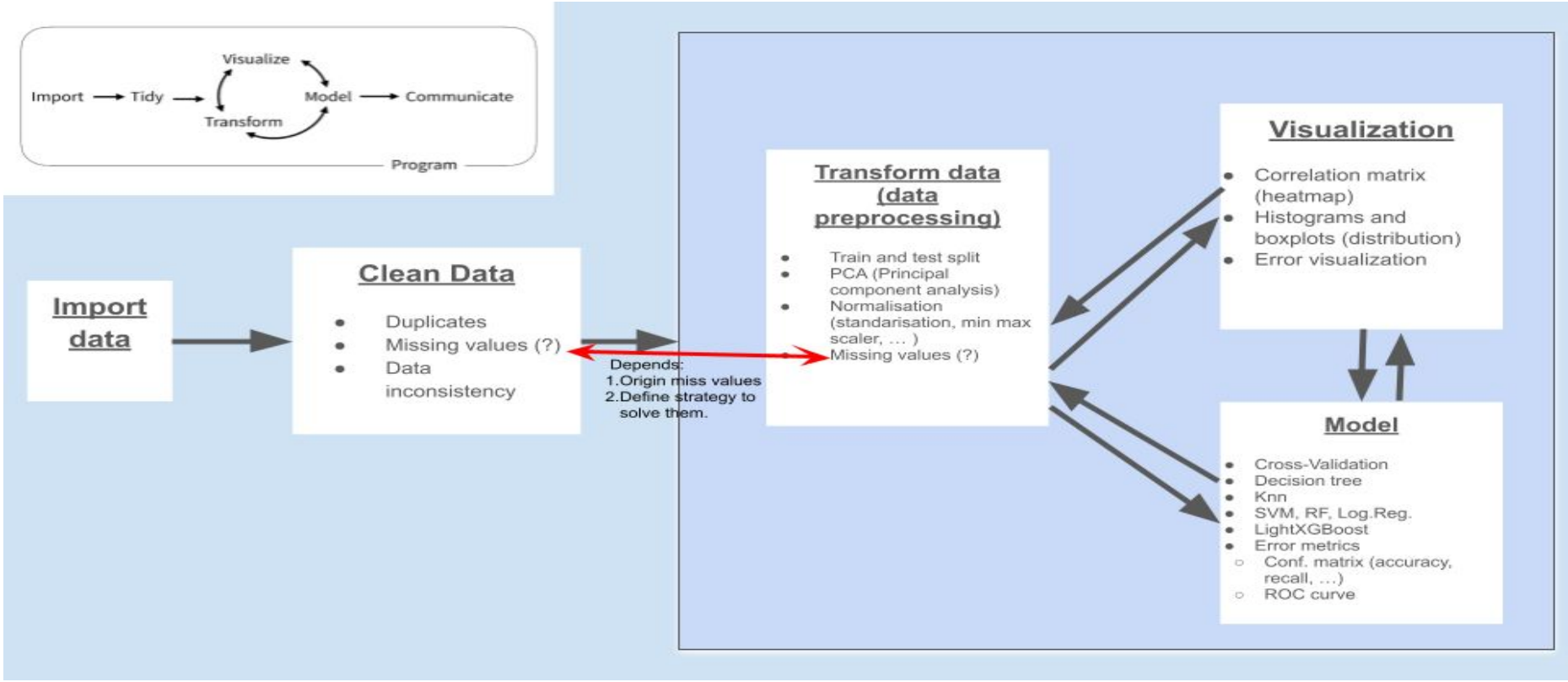
C64 Bank: Customer Payment Prediction Based on a ML Model

Lina Haidar, 27.08.2021

Flowchart:

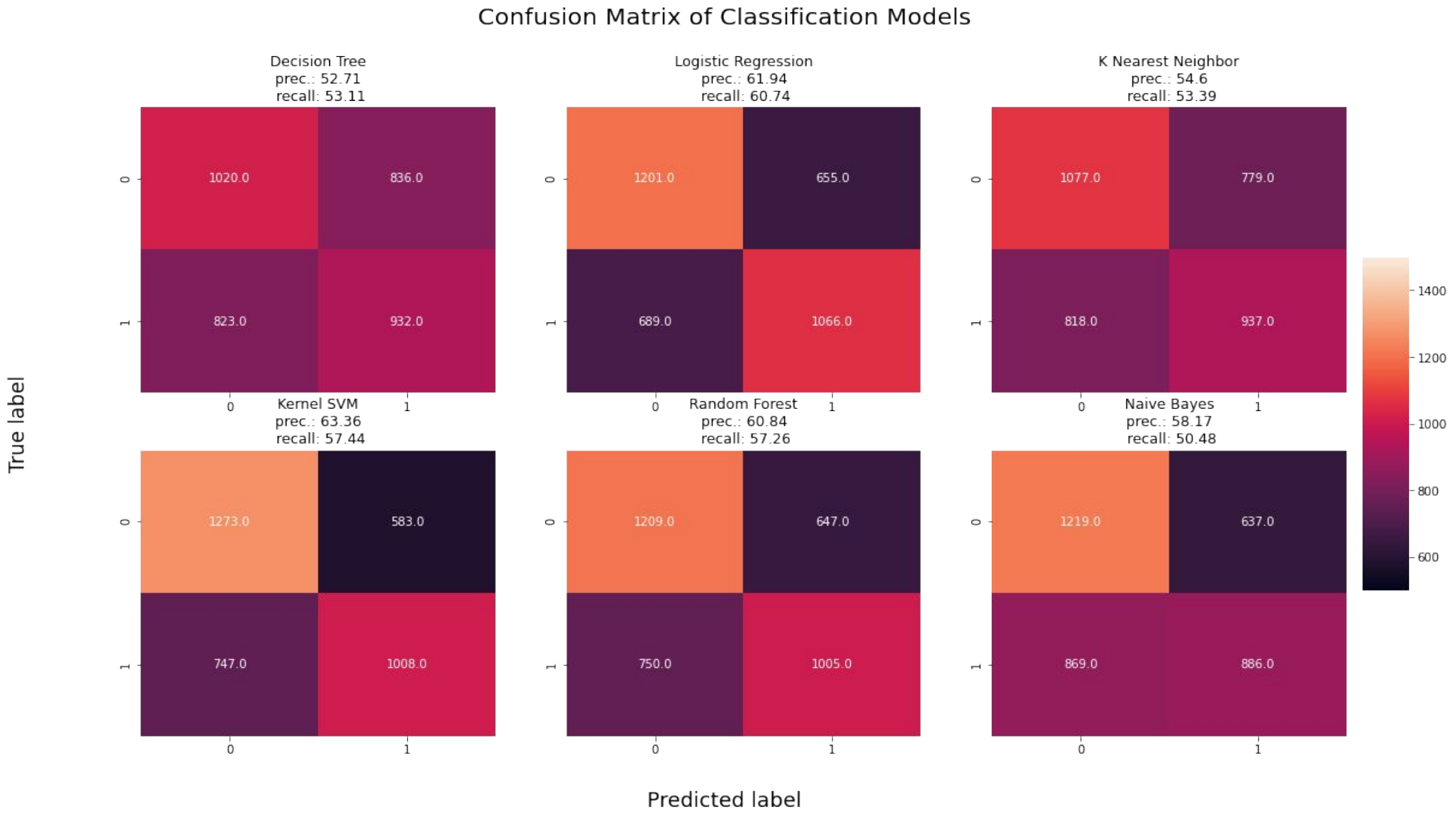

Import
Tidy
Visualize
Transform
Model
Communicate
Program
Import data
Clean Data
Duplicates
Missing values (?)
Data inconsistency
Depends:
1.Origin miss values
2.Define strategy to solve them.
Transform data (data preprocessing)
Train and test split
PCA (Principal component analysis)
Normalisation (standarisation, min max scaler, …)
Missing values (?)
Visualization
Correlation matrix (heatmap)
Histograms and boxplots (distribution)
Error visualization
Model
Cross-Validation
Decision tree
Knn
SVM, RF, Log.Reg.
LightXGBoost
Error metrics
Conf. matrix (accuracy, recall, …)
ROC curve

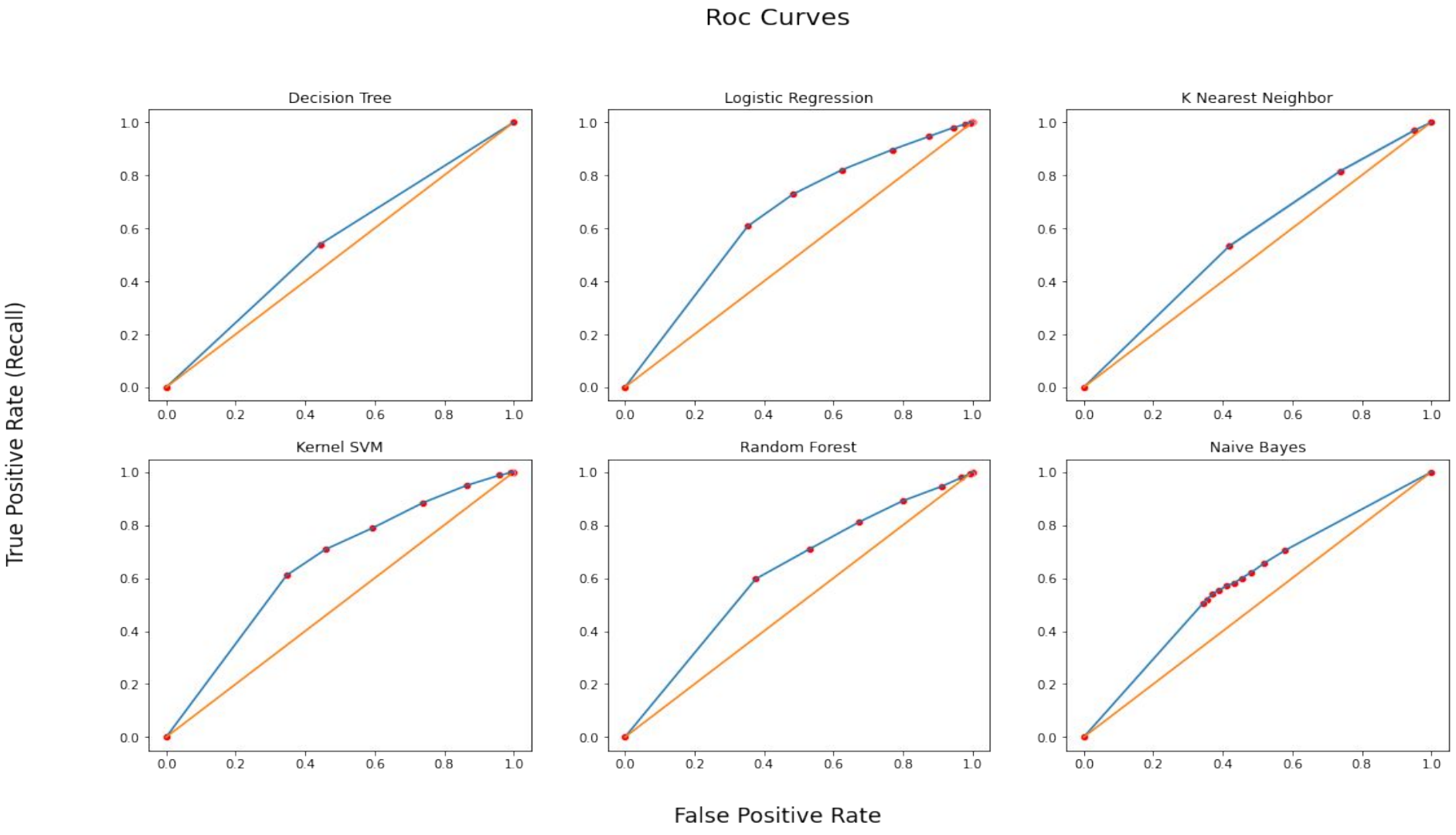
Data Pre-processing

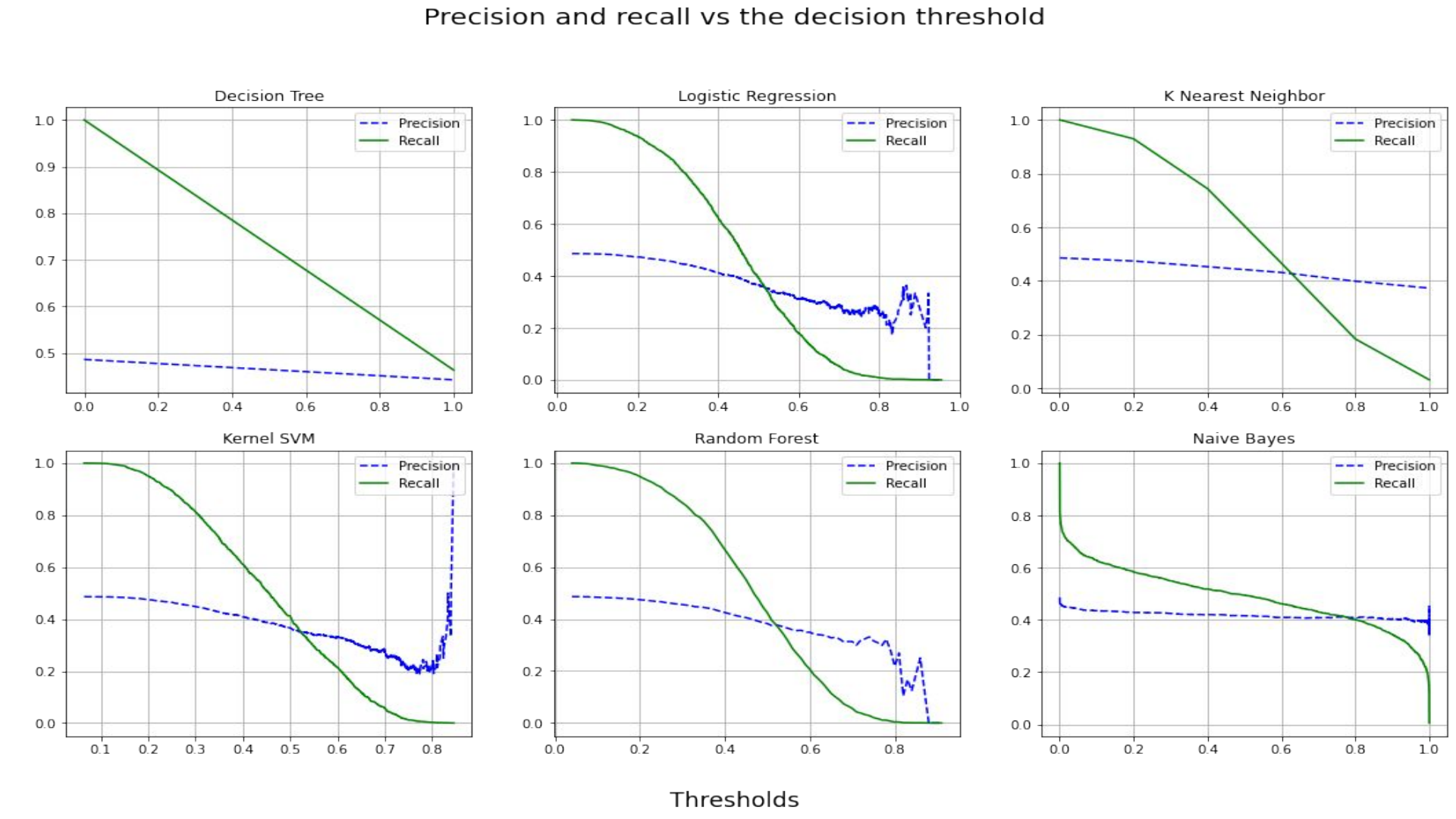
- Drop columns with nan values amount above 500
  - Result: **(80000, 76)** , **nan values 7%**
- Drop nan values: `data.dropna(inplace=True)`
  - Result: **(78812, 76)**
- Drop columns with zeros: (move to preprocessing)
  - Result: **(78812, 71)**

Data Pre-processing

- **Balance the data**: 7221 did not pay and 71591 paid
  - Result: **(14442, 72)**
- **Split the dataset:**
  - Training: 75%
  - Testing: 25 %
- **Normalize the data:** `MinMaxScaler(feature_range=( 0,1))`


Confusion Matrix of Classification Models
Decision Tree
prec.: 52.71
recall: 53.11
1020.0
836.0
823.0
932.0
Logistic Regression
prec.: 61.94
recall: 60.74
1201.0
655.0
689.0
1066.0
K Nearest Neighbor
prec.: 54.6
recall: 53.39
1077.0
779.0
818.0
937.0
Kernel SVM
prec.: 63.36
recall: 57.44
1273.0
583.0
747.0
1008.0
Random Forest
prec.: 60.84
recall: 57.26
1209.0
647.0
750.0
1005.0
Naive Bayes
prec.: 58.17
recall: 50.48
1219.0
637.0
869.0
886.0
True label
Predicted label
0
1
1400
1200
1000
800
600


Roc Curves
Decision Tree
Logistic Regression
K Nearest Neighbor
Kernel SVM
Random Forest
Naive Bayes
True Positive Rate (Recall)
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Precision and recall vs the decision threshold
Decision Tree
Logistic Regression
K Nearest Neighbor
Kernel SVM
Random Forest
Naive Bayes
Precision
Recall
Thresholds

1155

additional customers with low risk

5

additional customers with high risk

```
probs.query(' 0 < loss < 30  & default_pred == 0 & manual_pred ==1' )
```

	no pay	pay	default_pred	manual_pred	true_pred	loss
1	0.537062	0.462938	0	1	0	8
2	0.610747	0.389253	0	1	0	9
6	0.542742	0.457258	0	1	0	5
8	0.596208	0.403792	0	1	0	7
10	0.568978	0.431022	0	1	0	11
...	...	...	...	...	...	...
3601	0.661206	0.338794	0	1	0	1
3603	0.521220	0.478780	0	1	0	6
3604	0.572992	0.427008	0	1	0	2
3606	0.583965	0.416035	0	1	0	1
3607	0.635967	0.364033	0	1	0	12

1155 rows × 6 columns

LR: Lowering the threshold from **0.5** to **0.2**

```
probs.query(' 30 < loss < 100  & default_pred == 0 & manual_pred ==1' )
```

	no pay	pay	default_pred	manual_pred	true_pred	loss
332	0.516561	0.483439	0	1	0	36
520	0.526967	0.473033	0	1	0	40
2110	0.540418	0.459582	0	1	0	34
2145	0.645073	0.354927	0	1	0	39
3480	0.608678	0.391322	0	1	0	43

Good or bad ?

Conclusion

Model Candidate: Logistic Regression, SVM, Random Forest

Accuracy: around 60%

Adjusting the threshold changes output