ML Raport

AutoPrep

January 9, 2025

Abstract

This raport has been generated with AutoPrep.

Contents

1 Overview

1.1 System

System	Darwin
Machine	arm64
Processor	arm
Architecture	64bit
Python Version	3.10.5
Physical Cores	8
Logical Cores	8
CPU Frequency (MHz)	3204
Total RAM (GB)	16.00
Available RAM (GB)	4.22
Total Disk Space (GB)	228.27
Free Disk Space (GB)	8.30

Table 1: System overview.

1.2 Dataset

Table 2 presents an overview of the dataset including the number of samples, features, and their types.

Number of samples	227
Number of features	9
Number of numerical features	9
Number of categorical features	0

Table 2: Dataset Summary.

Distribution of the target classes in terms of the number of observations and their percentages is presented in Table 3

class	number of observations	Percentage
50	7	0.03
17	7	0.03
34	6	0.03
44	6	0.03
7	6	0.03
12	6	0.03
9	6	0.03
4	5	0.02
43	5	0.02
6	5	0.02
31	5	0.02
10	5	0.02
1	5	0.02
23	5	0.02
28	5	0.02
45	5	0.02
22	5	0.02
15	5	0.02
27	5	0.02
29	5	0.02
38	5	0.02
46	5	0.02
20	5	0.02
35	5	0.02
18	5	0.02
3	5	0.02
21	5	0.02
48	4	0.02
13	4	0.02
14	4	0.02
37	4	0.02
24	4	0.02
41	4	0.02
8	4	0.02
11	4	0.02
40	4	0.02
33	4	0.02
49	4	0.02
36	4	0.02
32	4	0.02
25	4	0.02

Table 4 presents the distribution of missing values in the dataset.

classgit	**number of observations**	**Percentage**
P85	0	0.00
P75	0	0.00
RMT85	0	0.00
CS82	0	0.00
SS82	0	0.00
S82	0	0.00
ME84	0	0.00
REV84	0	0.00
REG	0	0.00

Table 4: Missing values distribution.

Table 5 presents the description of features in the dataset.

class	**type**	**dtype**	**space usage**
P85	numerical	int64	3.6 kB
P75	numerical	int64	3.6 kB
RMT85	numerical	int64	3.6 kB
CS82	numerical	uint8	2.0 kB
SS82	numerical	uint8	2.0 kB
S82	numerical	uint8	2.0 kB
ME84	numerical	int64	3.6 kB
REV84	numerical	int64	3.6 kB
REG	numerical	uint8	2.0 kB

Table 5: Features dtypes description.

Table 6 and Table 7 present the description of numerical and categorical features in the dataset.

index	count	mean	std	min	25%	50%	75%	max
P85	227.00	29.99	56.17	3.00	10.00	16.00	30.00	653.00
P75	227.00	29.52	57.77	4.00	10.00	15.00	28.00	671.00
RMT85	227.00	254.51	657.60	21.00	66.50	118.00	229.50	6720.00
CS82	227.00	9.18	4.98	1.00	6.00	8.00	11.00	34.00
SS82	227.00	21.95	7.23	8.00	17.00	21.00	27.00	46.00
S82	227.00	47.15	10.57	31.00	41.00	45.00	49.00	101.00
ME84	227.00	1842.41	4685.06	173.00	480.50	839.00	1580.50	47074.00
REV84	227.00	3048.31	5125.17	347.00	1134.50	1828.00	3174.00	59877.00
REG	227.00	4.33	2.08	1.00	2.00	4.00	6.00	8.00

Table 6: Numerical features description.

2 Eda

This part of the report provides basic insides to the data and the informations it holds..

2.1 Target variable and missing values

Here we present the distribution of the target variable.

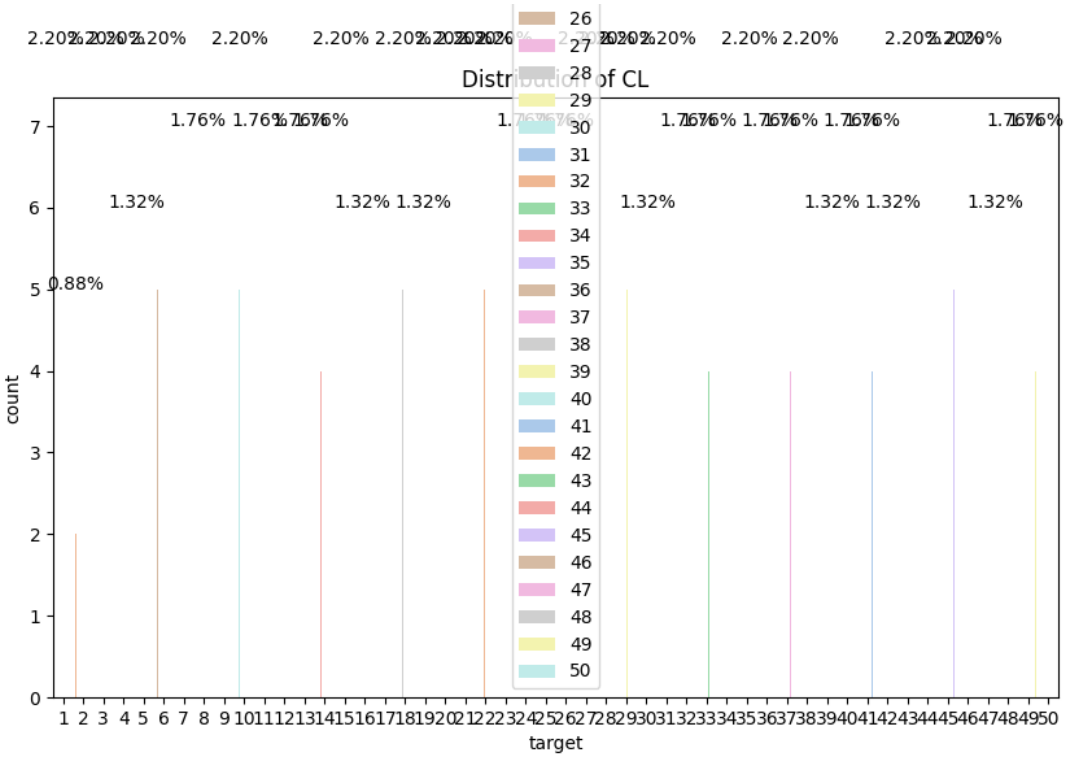
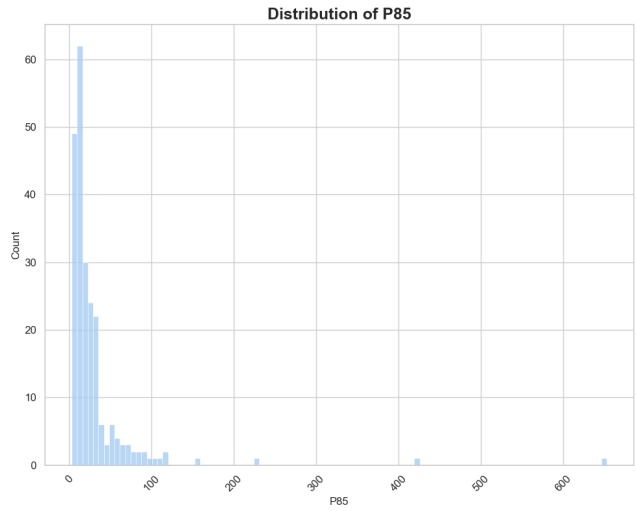


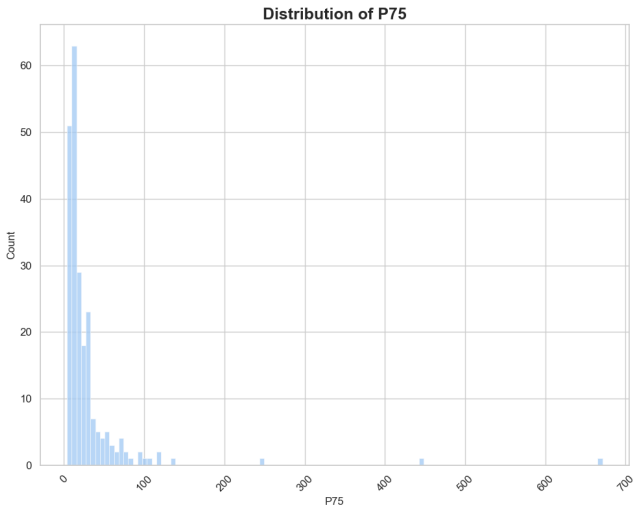
Figure 1: Target distribution.

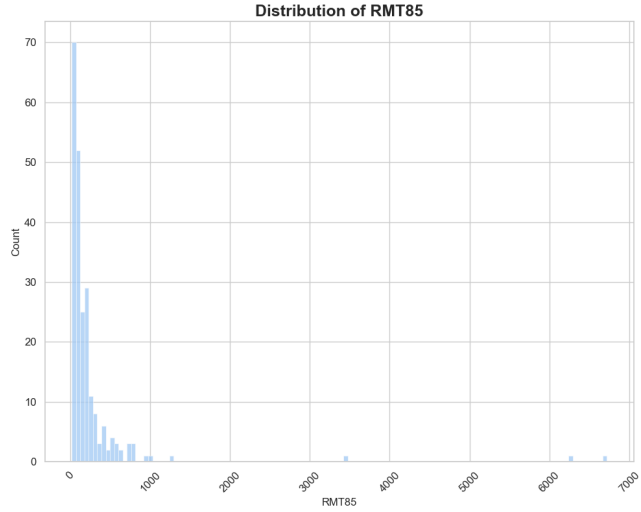
2.2 EDA for categorical features

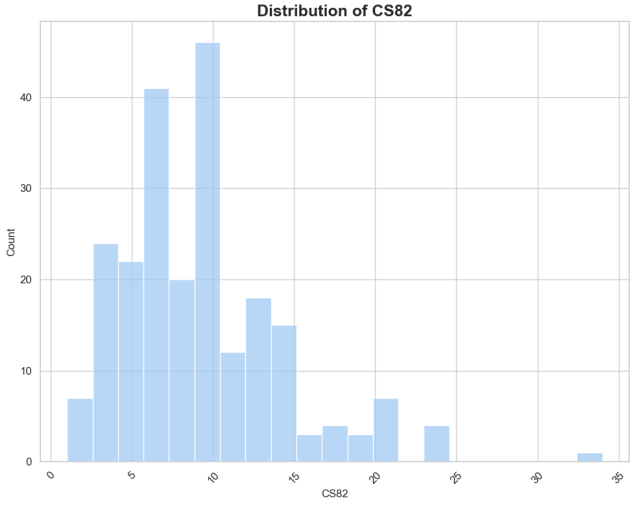
2.3 EDA for numerical features

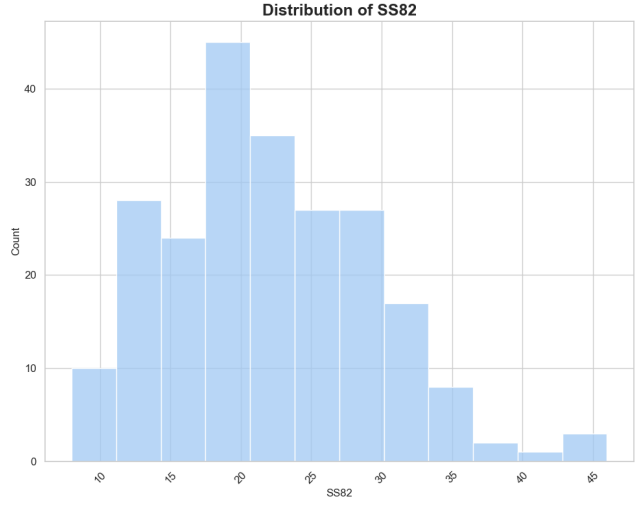
The distribution of numerical features is presented on histogram(s) below.

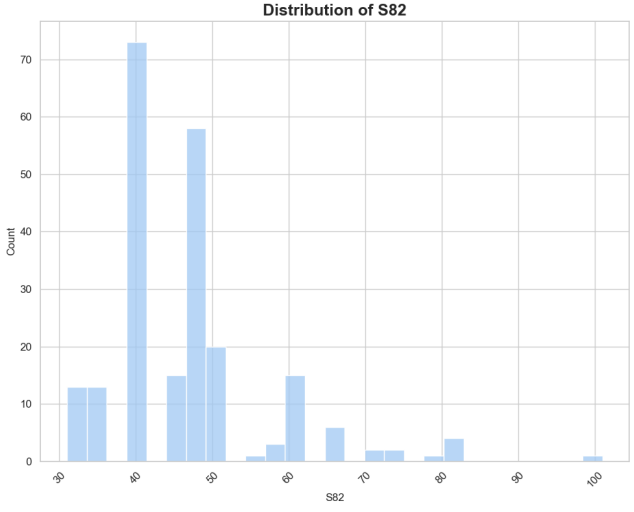

Distribution of P85
Count
P85

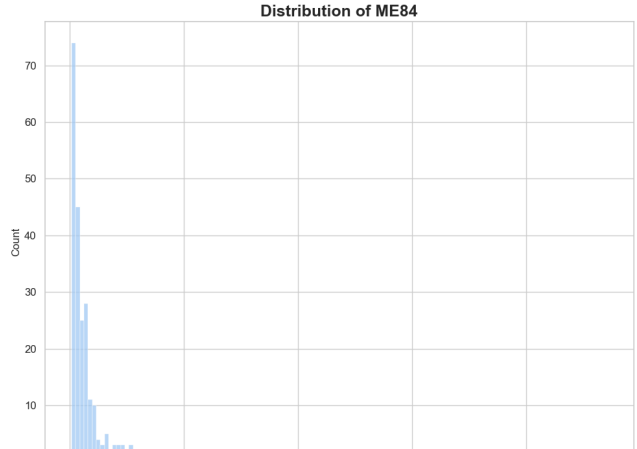

Distribution of P75
Count
P75

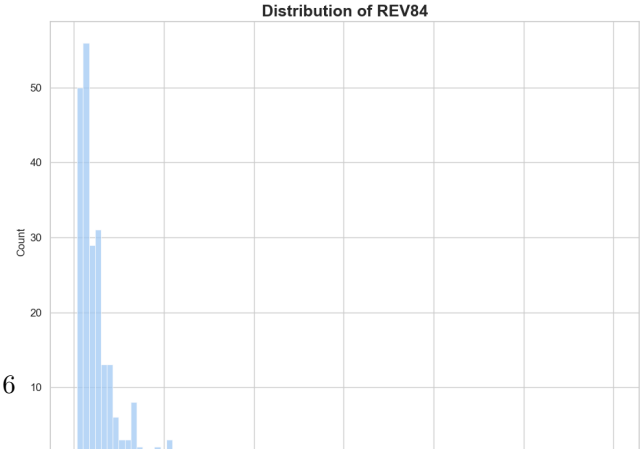

Distribution of RMT85
Count
RMT85


Distribution of CS82
Count
CS82


Distribution of SS82
Count
SS82


Distribution of S82
Count
S82


Distribution of ME84
Count


Distribution of REV84
Count

Here we present the correlation heatmap.

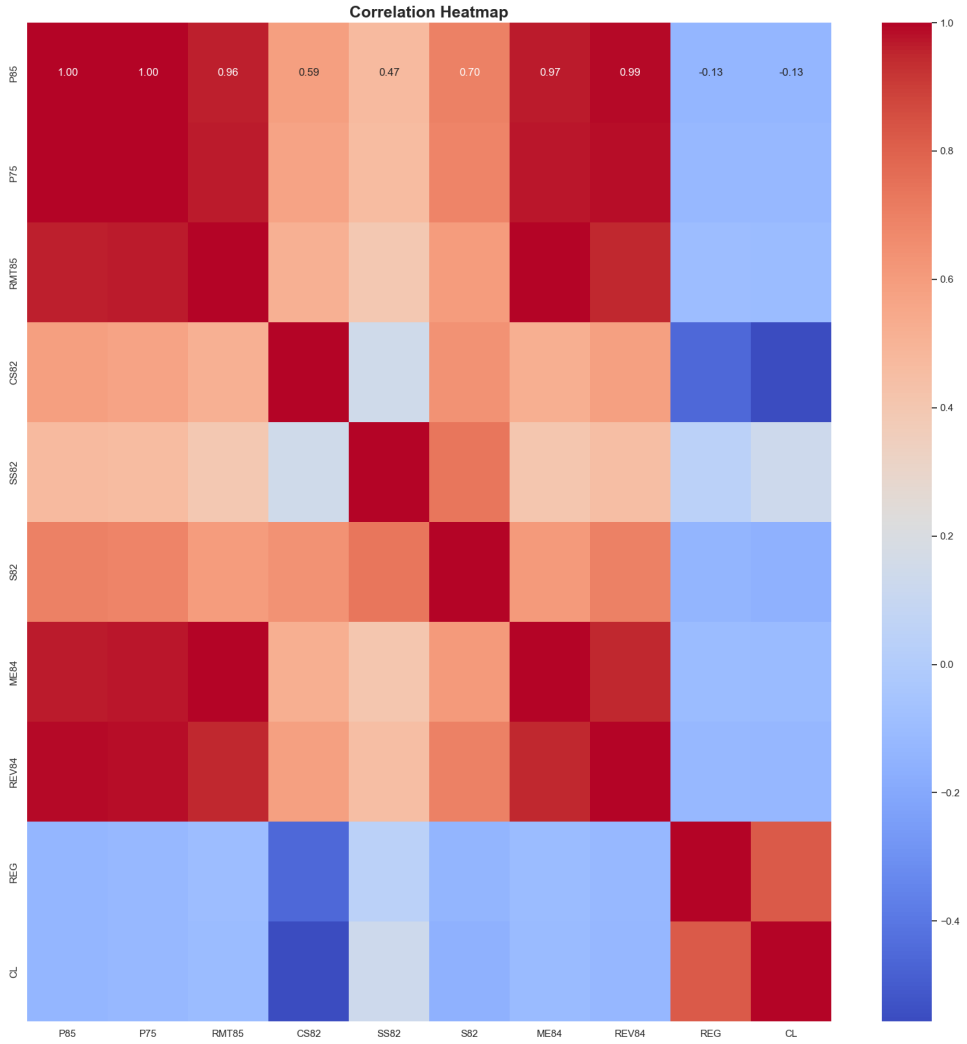
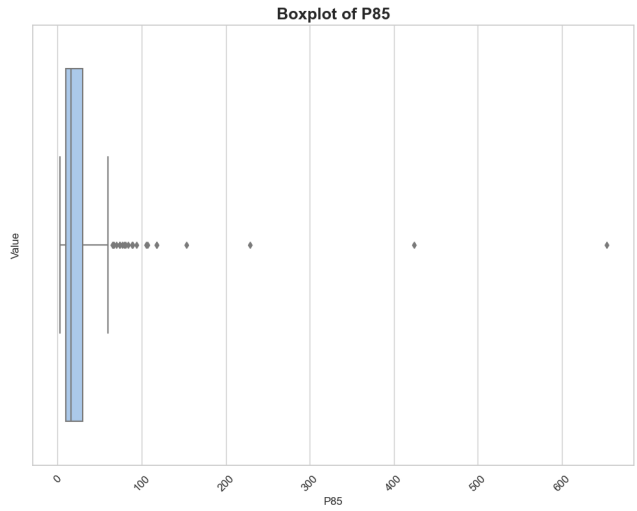
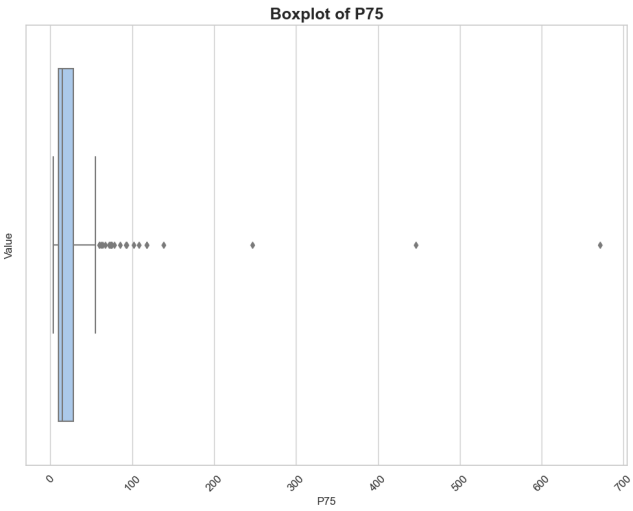
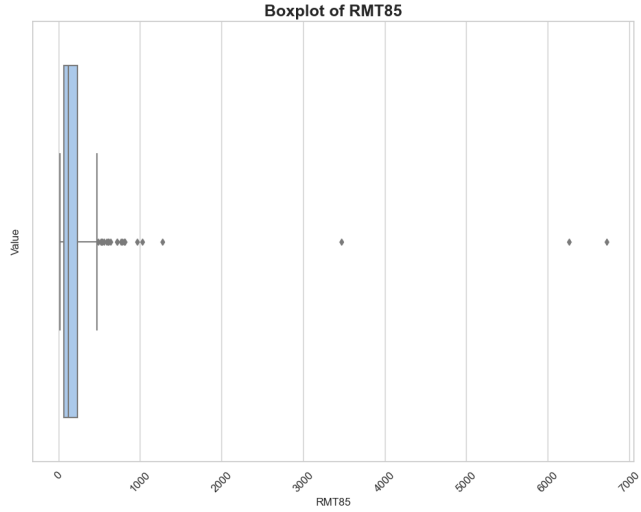


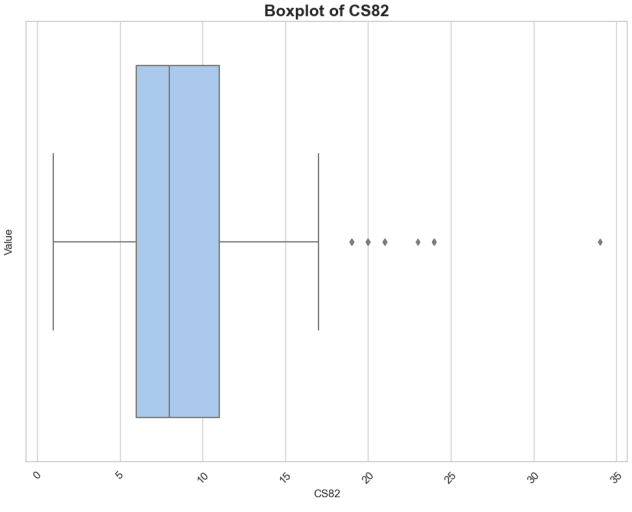
Figure 3: Correlation heatmap.

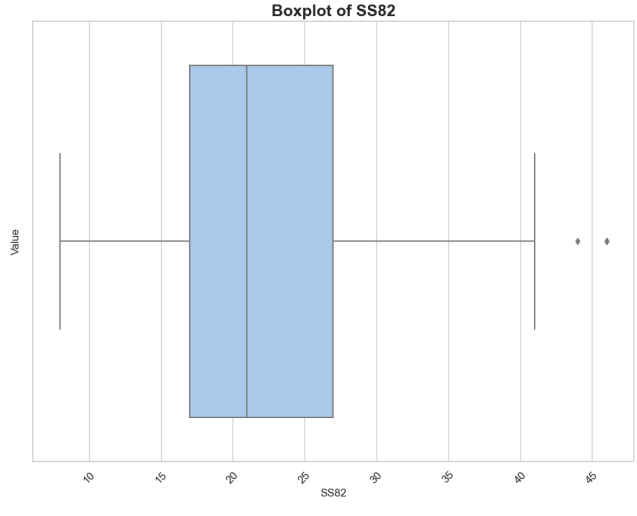
Here we present the boxplot(s) of numerical features.

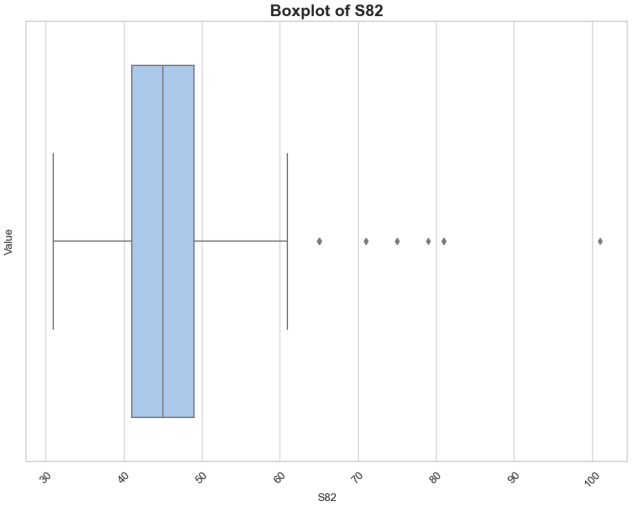

Boxplot of P85
Value
0
100
200
300
400
500
600
P85

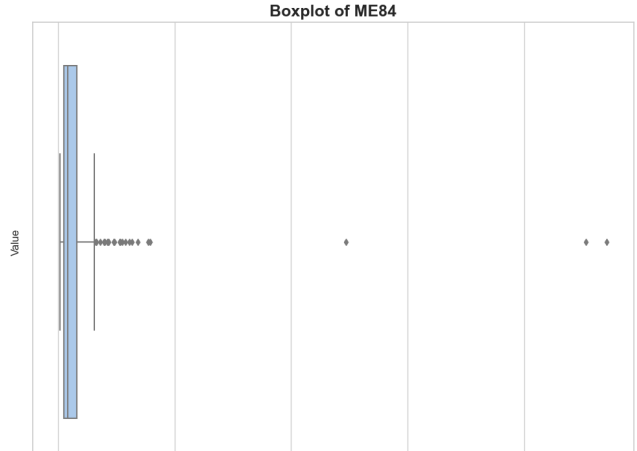

Boxplot of P75
Value
0
100
200
300
400
500
600
700
P75

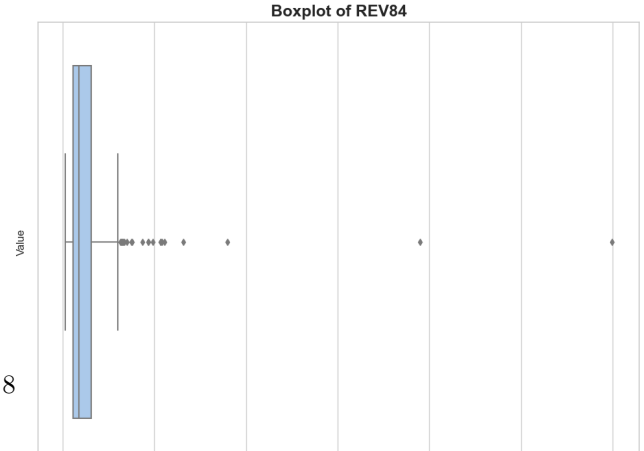

Boxplot of RMT85
Value
0
1000
2000
3000
4000
5000
6000
7000
RMT85


Boxplot of CS82
Value
0
5
10
15
20
25
30
35
CS82


Boxplot of SS82
Value
10
15
20
25
30
35
40
45
SS82


Boxplot of S82
Value
30
40
50
60
70
80
90
100
S82


Boxplot of ME84
Value


Boxplot of REV84
Value

3 Preprocessing

This part of the report presents the results of the preprocessing process. It contains required, as well as non required, steps listed below.
Required preprocessing steps

- Missing data imputation
- Removing columns with 100% unique categorical values
- Categorical features encoding
- Scaling
- Removing columns with 0 variance
- Detecting highly correlatd features

Additional preprocessing steps

- Feature selection methods : Correlation with the target or Random Forest feature importance
- Dimention reduction techniques: PCA, VIF, UMAP

Preprocessing process was configured to select up to 3 best unique preprocessing pipelines. Pipelines were scored based on a simple model. Tables below show detailed description of the best pipelines as well as all step combinations that were examined.

index	steps
0	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler

Table 7: Pipelines steps overview.

score index	file name	score	fit duration	score duration
0	preprocessing_pipeline_0.joblib	31.85	a moment	a moment
1	preprocessing_pipeline_1.joblib	31.82	a moment	a moment
2	preprocessing_pipeline_2.joblib	31.68	a moment	a moment

Table 8: Best preprocessing pipelines.

step	name	description	params
0	NAImputer	Imputes missing data.	{"numeric_imputer": "median", "categorical_imputer": "most_frequent"}
1	UniqueFilter	Removes categorical columns with 100% unique values. Dropped columns: []	{}
2	ColumnEncoder	Encodes categorical columns using OneHotEncoder (for columns with <5 unique values) or TolerantLabelEncoder (for columns with >=5 unique values). Encodes target variable using LabelEncoder if provided.	{}
3	VarianceFilter	Removes columns with zero variance. Dropped columns: []	{}
4	CorrelationFilter	Removes one column from pairs of columns correlated above correlation threshold: 0.8.	{}
5	ColumnScaler	Scales numerical columns using one of 3 scaling methods.	{"method": "minmax"}

Table 9: Best pipeline No. 0: steps overview.

index	count	mean	std	min	25%	50%	75%	max
P85	227.00	-0.00	1.00	-0.48	-0.36	-0.25	0.00	11.12
CS82	227.00	0.00	1.00	-1.64	-0.64	-0.24	0.37	4.99
SS82	227.00	0.00	1.00	-1.93	-0.69	-0.13	0.70	3.33
S82	227.00	-0.00	1.00	-1.53	-0.58	-0.20	0.18	5.11
REG	227.00	-0.00	1.00	-1.60	-1.12	-0.16	0.80	1.77

Table 10: Best pipeline No. 0: Output overview.

step	name	description	params
0	NAImputer	Imputes missing data.	{"numeric_imputer": "median", "categorical_imputer": "most_frequent"}
1	UniqueFilter	Removes categorical columns with 100% unique values. Dropped columns: []	{}
2	ColumnEncoder	Encodes categorical columns using OneHotEncoder (for columns with <5 unique values) or TolerantLabelEncoder (for columns with >=5 unique values). Encodes target variable using LabelEncoder if provided.	{}
3	VarianceFilter	Removes columns with zero variance. Dropped columns: []	{}
4	CorrelationFilter	Removes one column from pairs of columns correlated above correlation threshold: 0.8.	{}
5	ColumnScaler	Scales numerical columns using one of 3 scaling methods.	{"method": "robust"}

Table 11: Best pipeline No. 1: steps overview.

index	count	mean	std	min	25%	50%	75%	max
P85	227.00	0.04	0.09	0.00	0.01	0.02	0.04	1.00
CS82	227.00	0.25	0.15	0.00	0.15	0.21	0.30	1.00
SS82	227.00	0.37	0.19	0.00	0.24	0.34	0.50	1.00
S82	227.00	0.23	0.15	0.00	0.14	0.20	0.26	1.00
REG	227.00	0.48	0.30	0.00	0.14	0.43	0.71	1.00

Table 12: Best pipeline No. 1: Output overview.

step	name	description	params
0	NAImputer	Imputes missing data.	{"numeric_imputer": "median", "categorical_imputer": "most_frequent"}
1	UniqueFilter	Removes categorical columns with 100% unique values. Dropped columns: []	{}
2	ColumnEncoder	Encodes categorical columns using OneHotEncoder (for columns with <5 unique values) or TolerantLabelEncoder (for columns with >=5 unique values). Encodes target variable using LabelEncoder if provided.	{}
3	VarianceFilter	Removes columns with zero variance. Dropped columns: []	{}
4	CorrelationFilter	Removes one column from pairs of columns correlated above correlation threshold: 0.8.	{}
5	ColumnScaler	Scales numerical columns using one of 3 scaling methods.	{"method": "standard"}

Table 13: Best pipeline No. 2: steps overview.

index	count	mean	std	min	25%	50%	75%	max
P85	227.00	0.70	2.81	-0.65	-0.30	0.00	0.70	31.85
CS82	227.00	0.24	1.00	-1.40	-0.40	0.00	0.60	5.20
SS82	227.00	0.10	0.72	-1.30	-0.40	0.00	0.60	2.50
S82	227.00	0.27	1.32	-1.75	-0.50	0.00	0.50	7.00
REG	227.00	0.08	0.52	-0.75	-0.50	0.00	0.50	1.00

Table 14: Best pipeline No. 2: Output overview.

You may also find all pipelines' runtime statistic in Table 16

Category	Value
Unique created pipelines	1
All created pipelines (after exploading each step params)	3
All pipelines fit time	2 seconds
All pipelines score time	3 seconds
scores_count	3.00
scores_mean	31.78
scores_std	0.09
scores_min	31.68
scores_25%	31.75
scores_50%	31.82
scores_75%	31.84
scores_max	31.85
Scoring function	<class 'str'>
Scoring model	RandomForestRegressor

Table 15: Preprocessing pipelines runtime statistics.

4 Modeling

4.1 Overview

This part of the report presents the results of the modeling process. There were regression 6 models trained for each of the best preprocessing pipelines.The following models were used in the modeling process.

- LinearSVR
- KNeighborsRegressor
- RandomForestRegressor
- BayesianRidge
- GradientBoostingRegressor
- LinearRegression

4.2 Hyperparameter tuning

This section presents the results of hyperparameter tuning for each of the best 3 models. using RandomizedSearchCV. Param grids used for each model are presented in the tables below.

Category	Value
epsilon	[0.0, 0.1, 0.2, 0.5, 1.0]
C	[0.1, 1.0, 10.0, 100.0]
loss	['epsilon_insensitive', 'squared_epsilon_insensitive']
fit_intercept	[True, False]

Table 16: Param grid for model LinearSVR.

Category	Value
n_neighbors	[5, 10, 15]
weights	['uniform', 'distance']
algorithm	['auto', 'ball_tree', 'kd_tree', 'brute']
leaf_size	[30, 40, 50]
p	[1, 2]

Table 17: Param grid for model KNeighboursRegressor.

Category	Value
n_estimators	[100, 200, 300]
max_depth	[None, 5, 10, 15, 20]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]
max_features	['sqrt', 'log2', None]
bootstrap	[True, False]
random_state	[42]

Table 18: Param grid for model RandomForestRegressor.

Category	Value
max_iter	[300, 400, 500]
tol	[0.001, 0.0001, 1e-05]
alpha_1	[1e-06, 1e-07, 1e-08]
alpha_2	[1e-06, 1e-07, 1e-08]
lambda_1	[1e-06, 1e-07, 1e-08]
lambda_2	[1e-06, 1e-07, 1e-08]

Table 19: Param grid for model BayesianRidgeRegressor.

Category	Value
n_estimators	[100, 200, 300]
learning_rate	[0.1, 0.05, 0.02]
max_depth	[4, 6, 8]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]
subsample	[1.0, 0.5]
random_state	[42]

Table 20: Param grid for model GradientBoostingRegressor.

Category	Value
fit_intercept	[True, False]

Table 21: Param grid for model LinearRegression.

Table 22 presents the best models and pipelines along with their hyperparameters, mean fit time, and test score.

Model	Pipeline	Best params	Mean fit time	Test score
LinearSVR	final_pipeline_1.joblib	{"loss": "epsilon_insensitive", "fit_intercept": true, "epsilon": 0.0, "C": 0.1}	0.00	250.06
LinearSVR	final_pipeline_2.joblib	{"loss": "epsilon_insensitive", "fit_intercept": false, "epsilon": 0.0, "C": 1.0}	0.01	72.54
LinearSVR	final_pipeline_0.joblib	{"loss": "epsilon_insensitive", "fit_intercept": false, "epsilon": 0.2, "C": 1.0}	0.00	71.53

Table 22: Best models results