ML Raport

AutoPrep

January 13, 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)	16.0000
Available RAM (GB)	6.0100
Total Disk Space (GB)	228.2700
Free Disk Space (GB)	13.0500

Table 1: System overview.

1.2 Dataset

Task detected for the dataset: binary classfication.
Table 2 presents an overview of the dataset including the number of samples, features, and their types.

Number of samples	1047
Number of features	13
Number of numerical features	6
Number of categorical features	7

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**	**fraction**
0	665	0.6351
1	382	0.3649

Table 3: Target class distribution.

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

feature	**number of observations**	**fraction**
pclass	0	0.0000
name	0	0.0000
sex	0	0.0000
age	207	0.1977
sibsp	0	0.0000
parch	0	0.0000
ticket	0	0.0000
fare	1	0.0010
cabin	813	0.7765
embarked	1	0.0010
boat	672	0.6418
body	948	0.9054
home___dest	453	0.4327

Table 4: Missing values distribution.

Table 5 presents the description of features in the dataset.

feature	**type**	**dtype**	**space usage**
pclass	numerical	uint8	9.4 kB
name	categorical	object	96.4 kB
sex	categorical	category	9.7 kB
age	numerical	float64	16.8 kB
sibsp	numerical	uint8	9.4 kB
parch	numerical	uint8	9.4 kB
ticket	categorical	object	75.1 kB
fare	numerical	float64	16.8 kB
cabin	categorical	object	42.1 kB
embarked	categorical	category	9.7 kB
boat	categorical	object	46.4 kB
body	numerical	float64	16.8 kB
home___dest	categorical	object	64.5 kB

Table 5: Features dtypes description.

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

feature	**count**	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
pclass	1047.0000	2.2970	0.8369	1.0000	2.0000	3.0000	3.0000	3.0000
age	840.0000	29.5327	14.2658	0.1667	21.0000	28.0000	38.6250	80.0000
sibsp	1047.0000	0.5205	1.0500	0.0000	0.0000	0.0000	1.0000	8.0000
parch	1047.0000	0.3954	0.8942	0.0000	0.0000	0.0000	0.0000	9.0000
fare	1046.0000	33.5472	51.8097	0.0000	7.9250	14.5000	31.2750	512.3292
body	99.0000	160.8990	98.3519	1.0000	73.5000	156.0000	255.5000	328.0000

Table 6: Numerical features description.

index	**count**	**unique**	**top**	**freq**
name	1047	1046	Connolly, Miss. Kate	2
sex	1047	2	male	677
ticket	1047	773	CA. 2343	9
cabin	234	161	B57 B59 B63 B66	5
embarked	1046	3	S	737
boat	375	25	13	34
home___dest	594	317	New York, NY	50

Table 7: Categorical 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

Figure 1 shows the distribution of the target variable.

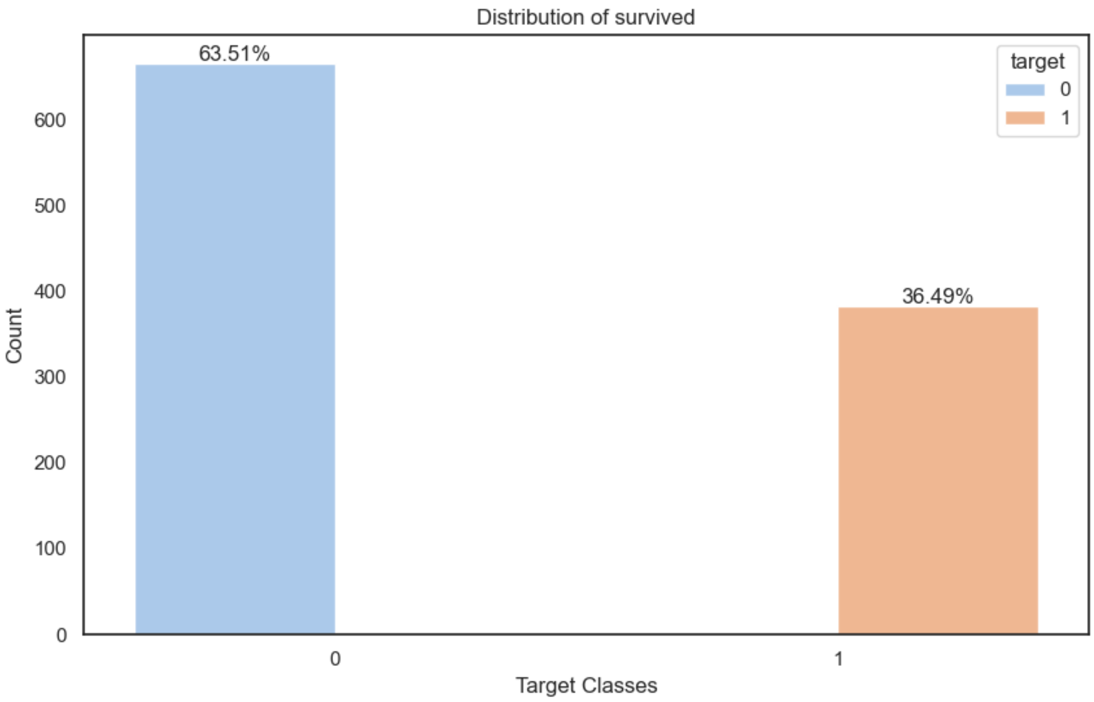

Figure 1: Target distribution.

Figure 2 shows the distribution of missing values in the dataset.

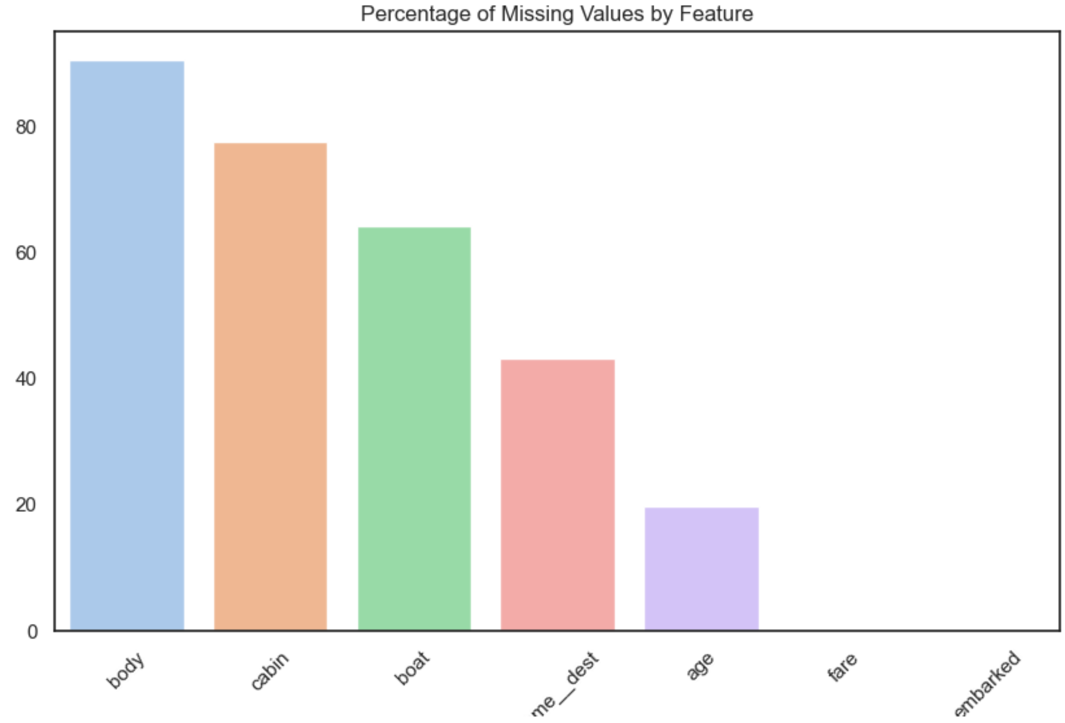

Figure 2: Missing values.

2.2 EDA for categorical features

The distribution of categorical features is presented on barplot(s) below.

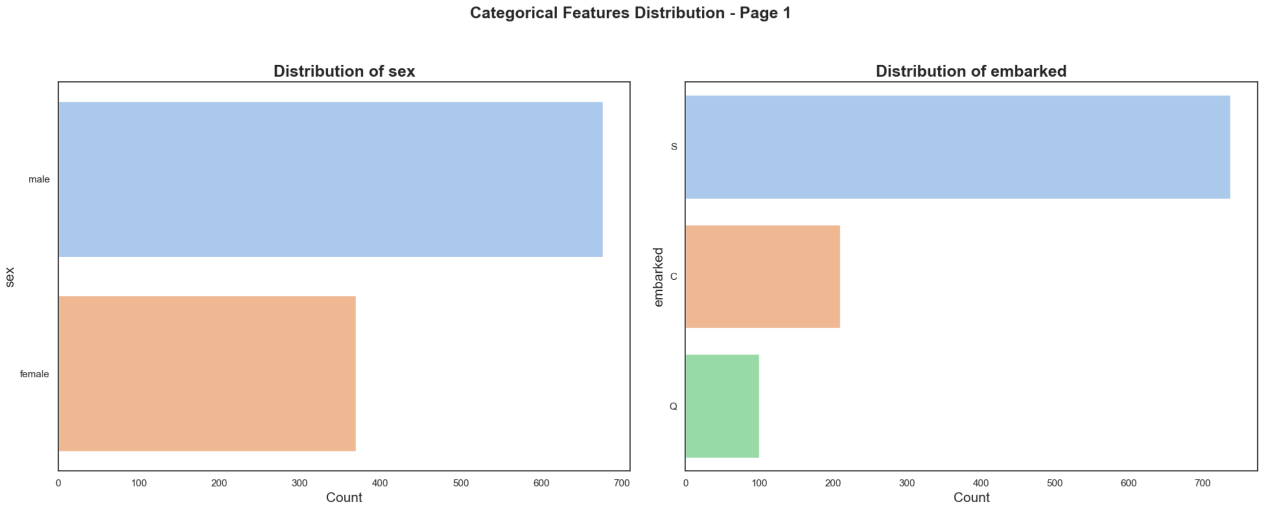

Figure 3: Categorical Features Distribution - Page 1

2.3 EDA for numerical features

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

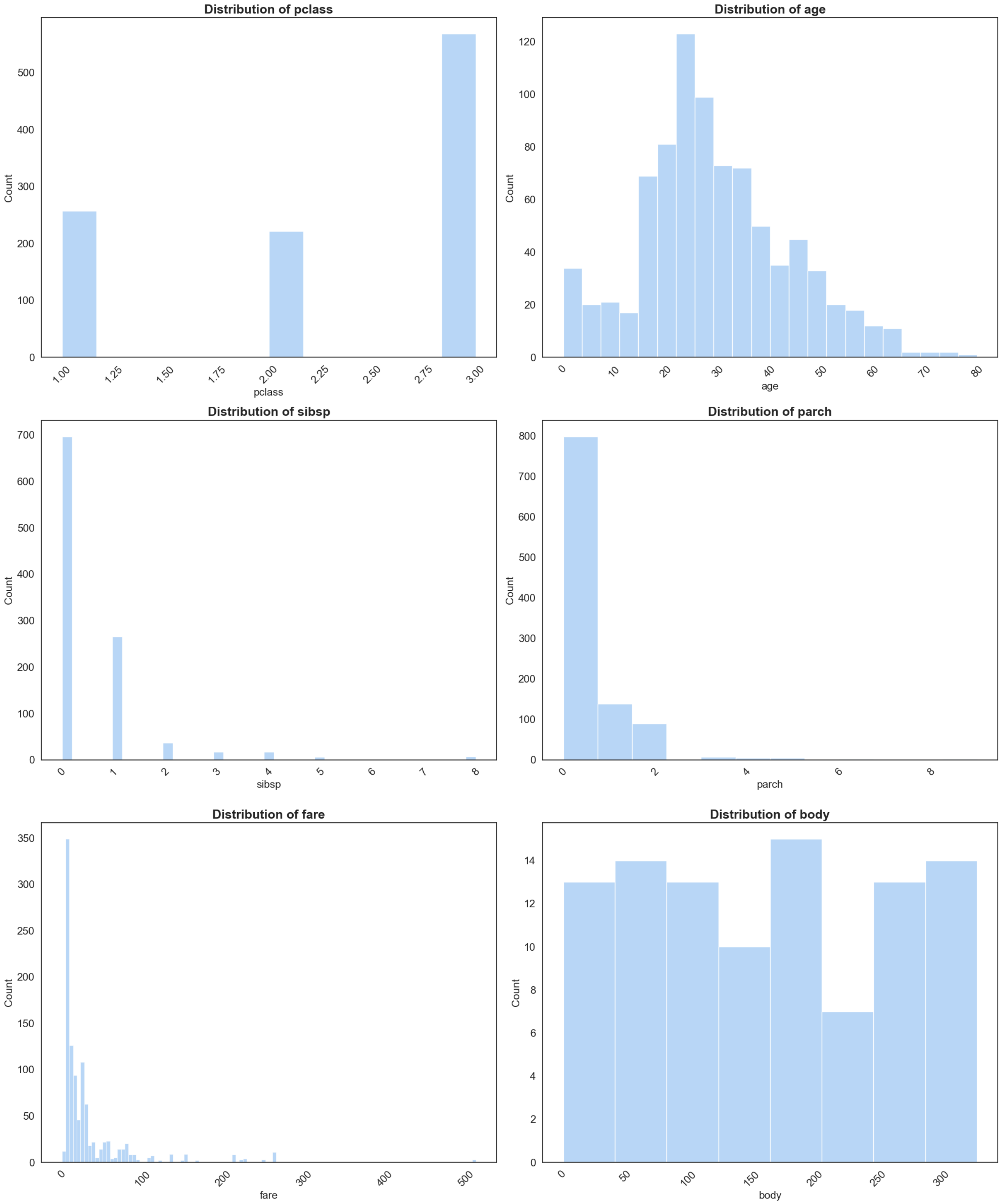

Figure 4: Numerical Features Distribution - Page 1

Figure 5 shows the correlation between features.

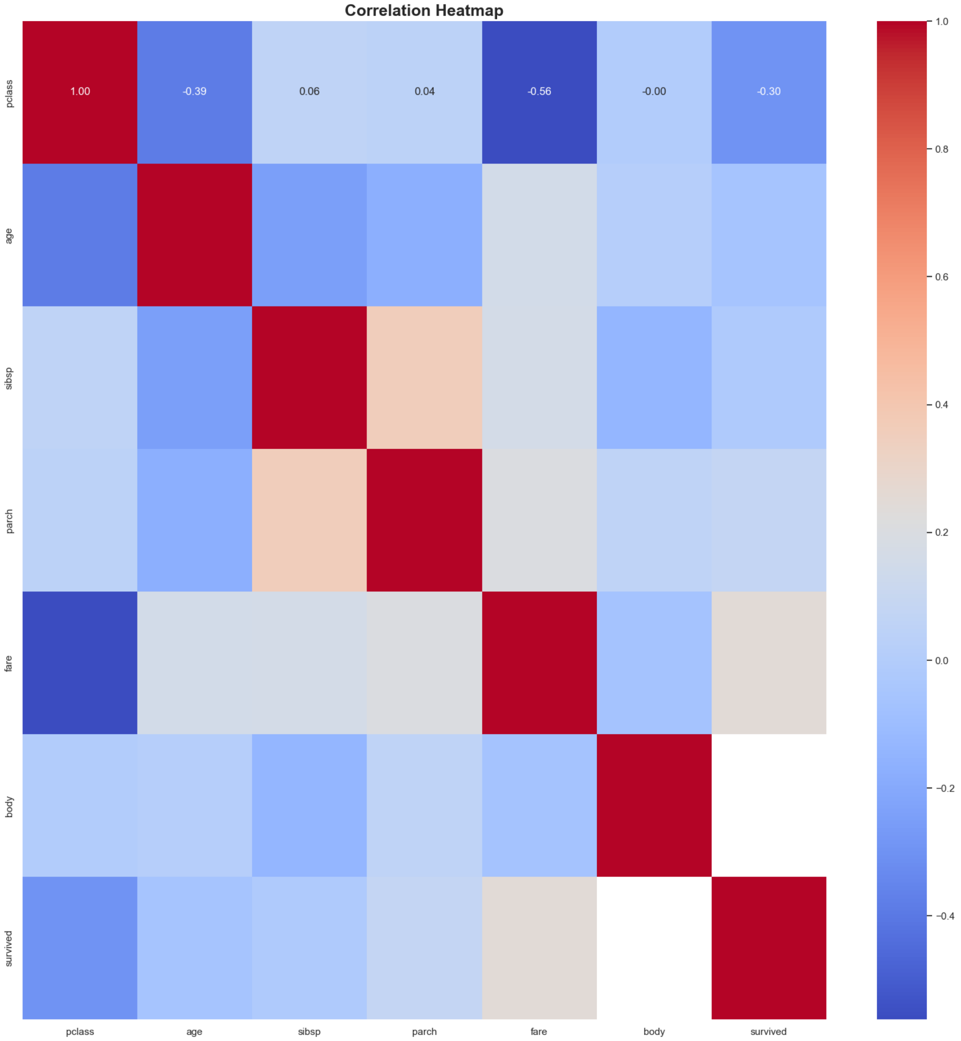

Figure 5: Correlation heatmap.

The boxplot of numerical features is presented on chart(s) below.

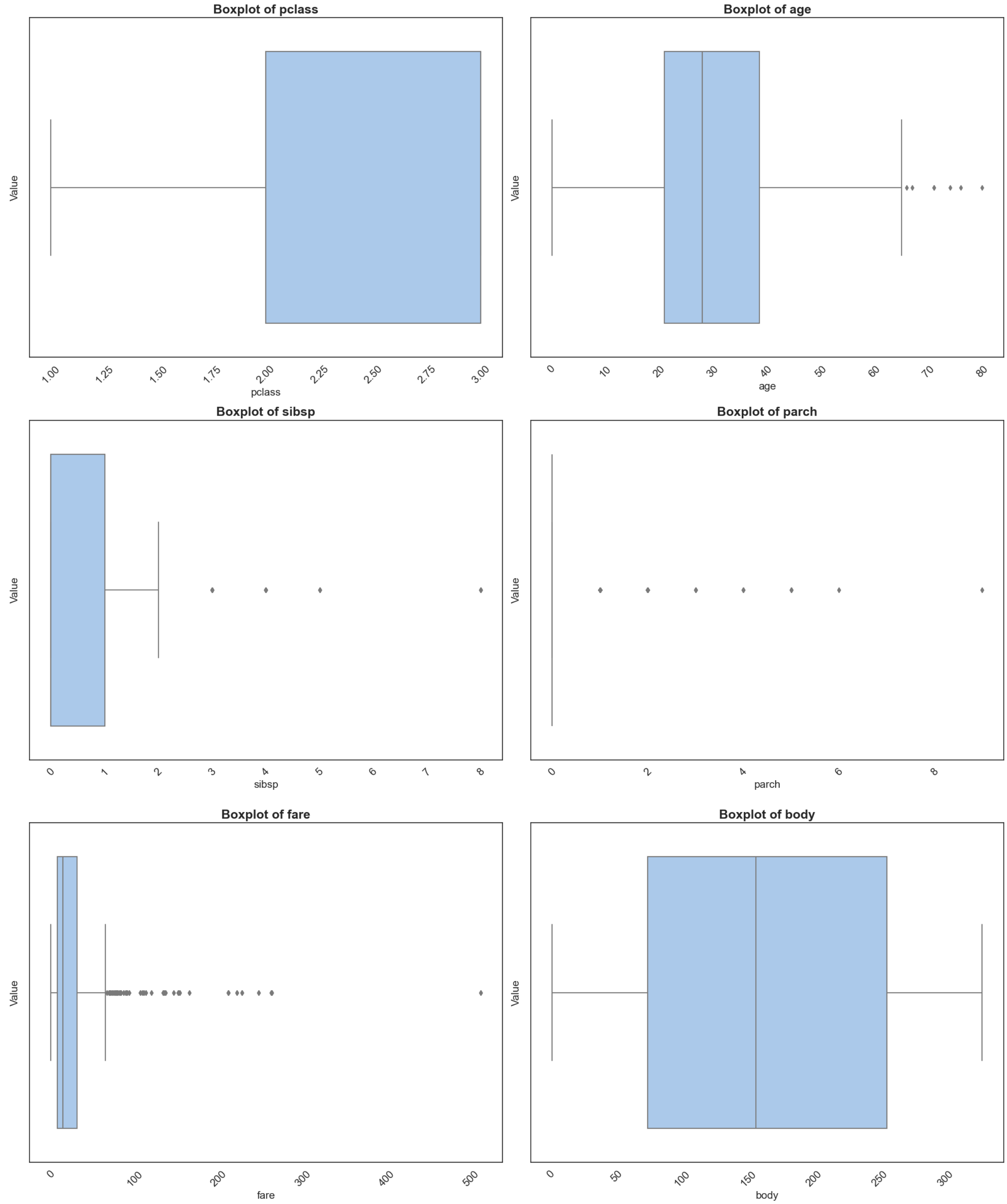

Figure 6: Boxplot page 1

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
1	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, CorrelationSelector
2	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, FeatureImportanceRegressSelector
3	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, FeatureImportanceClassSelector
4	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, PCADimentionReducer
5	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, CorrelationSelector, PCADimentionReducer
6	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, FeatureImportanceRegressSelector, PCADimentionReducer
7	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, FeatureImportanceClassSelector, PCADimentionReducer
8	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, UMAPDimentionReducer
9	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, CorrelationSelector, UMAPDimentionReducer
10	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, FeatureImportanceRegressSelector, UMAPDimentionReducer
11	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, FeatureImportanceClassSelector, UMAPDimentionReducer
12	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, VIFDimentionReducer
13	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, CorrelationSelector, VIFDimentionReducer
14	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, FeatureImportanceRegressSelector, VIFDimentionReducer
15	NAImputer, UniqueFilter, ColumnEncoder, VarianceFilter, CorrelationFilter, ColumnScaler, FeatureImportanceClassSelector, VIFDimentionReducer

Table 8: Pipelines steps overview.

index	file name	score	fit duration	score duration
0	preprocessing_pipeline_0.joblib	0.7680	a moment	a moment
1	preprocessing_pipeline_1.joblib	0.7595	4 seconds	a moment
2	preprocessing_pipeline_2.joblib	0.7595	4 seconds	a moment

Table 9: 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": "standard"}
6	CorrelationSelector	Selects the top 70.0% (rounded to whole number) of features most correlated with the target variable. Number of features that were selected: 0	{"correlation_percent": 0.7}
7	PCADimentionReducer	Combines PCA with automatic selection of the number of components to preserve 95% of the variance.	{"n_components": null}

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

index	count	mean	std	min	25%	50%	75%	max
pclass	1047.0000	0.0000	1.0005	-1.5506	-0.3551	0.8404	0.8404	0.8404
name	1047.0000	0.0000	1.0005	-1.7293	-0.8667	-0.0007	0.8653	1.7313
age	1047.0000	-0.0000	1.0005	-2.2732	-0.5655	-0.0962	0.4513	3.9711
sibsp	1047.0000	-0.0000	1.0005	-0.4960	-0.4960	-0.4960	0.4568	7.1264
parch	1047.0000	0.0000	1.0005	-0.4424	-0.4424	-0.4424	-0.4424	9.6277
ticket	1047.0000	-0.0000	1.0005	-1.6829	-0.8990	0.0021	0.9336	1.6697
fare	1047.0000	0.0000	1.0005	-0.6477	-0.4946	-0.3676	-0.0435	9.2498
home__dest	1047.0000	-0.0000	1.0005	-2.7245	-0.1840	0.2345	0.3017	2.0128
sex_female	1047.0000	0.0000	1.0005	-0.7393	-0.7393	-0.7393	1.3527	1.3527
embarked_C	1047.0000	-0.0000	1.0005	-0.4994	-0.4994	-0.4994	-0.4994	2.0024
embarked_Q	1047.0000	0.0000	1.0005	-0.3250	-0.3250	-0.3250	-0.3250	3.0773
embarked_S	1047.0000	-0.0000	1.0005	-1.5454	-1.5454	0.6471	0.6471	0.6471

Table 11: 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": "standard"}
6	FeatureImportanceClassSelector	Selects the top 10.0% (rounded to whole number) of features most important according to Random Forest model for classification. Number of features that were selected: 0	{"k": 10.0}
7	UMAPDimentionReducer	Reduces the dimensionality of the data using UMAP.	{"n_components": null}

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

index	count	mean	std	min	25%	50%	75%	max
pclass	1047.0000	0.6485	0.4185	0.0000	0.5000	1.0000	1.0000	1.0000
name	1047.0000	0.4997	0.2891	0.0000	0.2493	0.4995	0.7498	1.0000
age	1047.0000	0.3640	0.1602	0.0000	0.2735	0.3486	0.4363	1.0000
sibsp	1047.0000	0.0651	0.1313	0.0000	0.0000	0.0000	0.1250	1.0000
parch	1047.0000	0.0439	0.0994	0.0000	0.0000	0.0000	0.0000	1.0000
ticket	1047.0000	0.5020	0.2984	0.0000	0.2338	0.5026	0.7804	1.0000
fare	1047.0000	0.0654	0.1011	0.0000	0.0155	0.0283	0.0610	1.0000
home___dest	1047.0000	0.5751	0.2112	0.0000	0.5363	0.6246	0.6388	1.0000
sex_female	1047.0000	0.3534	0.4783	0.0000	0.0000	0.0000	1.0000	1.0000
embarked_C	1047.0000	0.1996	0.3999	0.0000	0.0000	0.0000	0.0000	1.0000
embarked_Q	1047.0000	0.0955	0.2941	0.0000	0.0000	0.0000	0.0000	1.0000
embarked_S	1047.0000	0.7049	0.4563	0.0000	0.0000	1.0000	1.0000	1.0000

Table 13: 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": "robust"}
6	FeatureImportanceClassSelector	Selects the top 10.0% (rounded to whole number) of features most important according to Random Forest model for classification. Number of features that were selected: 0	{"k": 10.0}
7	UMAPDimentionReducer	Reduces the dimensionality of the data using UMAP.	{"n_components": null}

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

index	count	mean	std	min	25%	50%	75%	max
pclass	1047.0000	-0.7030	0.8369	-2.0000	-1.0000	0.0000	0.0000	0.0000
name	1047.0000	0.0004	0.5776	-0.9981	-0.5000	0.0000	0.5000	1.0000
age	1047.0000	0.0946	0.9839	-2.1410	-0.4615	0.0000	0.5385	4.0000
sibsp	1047.0000	0.5205	1.0500	0.0000	0.0000	0.0000	1.0000	8.0000
parch	1047.0000	0.3954	0.8942	0.0000	0.0000	0.0000	0.0000	9.0000
ticket	1047.0000	-0.0011	0.5459	-0.9194	-0.4917	0.0000	0.5083	0.9100
fare	1047.0000	0.8149	2.2179	-0.6210	-0.2816	0.0000	0.7184	21.3203
home___dest	1047.0000	-0.4828	2.0599	-6.0923	-0.8615	0.0000	0.1385	3.6615
sex_female	1047.0000	0.3534	0.4783	0.0000	0.0000	0.0000	1.0000	1.0000
embarked_C	1047.0000	0.1996	0.3999	0.0000	0.0000	0.0000	0.0000	1.0000
embarked_Q	1047.0000	0.0955	0.2941	0.0000	0.0000	0.0000	0.0000	1.0000
embarked_S	1047.0000	-0.2951	0.4563	-1.0000	-1.0000	0.0000	0.0000	0.0000

Table 15: Best pipeline No. 2: output overview.

Category	Value
Unique created pipelines	16
All created pipelines (after exploading each step params)	48
All pipelines fit time	23 seconds
All pipelines score time	20 seconds
scores_count	48.0000
scores_mean	0.7352
scores_std	0.0336
scores_min	0.6239
scores_25%	0.7362
scores_50%	0.7511
scores_75%	0.7511
scores_max	0.7680
Scoring function	function
Scoring model	RandomForestClassifier

Table 16: Preprocessing pipelines runtime statistics.

4 Modeling

4.1 Overview

This part of the report presents the results of the modeling process. There were 5 classification models trained for each of the best preprocessing pipelines.
The following models were used in the modeling process.

- KNeighborsClassifier

- LogisticRegression
- GaussianNB
- SVC
- DecisionTreeClassifier

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**
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 KNeighboursClassifier.

Category	**Value**
0	{"penalty": ["l1"], "C": [0.01, 0.1, 1, 10], "solver": ["liblinear", "saga"]}
1	{"penalty": ["l2"], "C": [0.01, 0.1, 1, 10], "solver": ["lbfgs", "liblinear", "saga", "newton-cg"]}
2	{"penalty": ["elasticnet"], "C": [0.01, 0.1, 1, 10], "solver": ["saga"], "l1_ratio": [0.5, 0.7]}

Table 18: Param grid for model LogisticRegression.

Category	**Value**
priors	[None]
var_smoothing	[1e-09, 1e-07, 1e-05]

Table 19: Param grid for model GaussianNaiveClassifier.

Category	**Value**
C	[0.1, 1, 10, 100, 1000]
kernel	['linear', 'poly', 'rbf', 'sigmoid']
degree	[3, 4, 5]
gamma	['scale', 'auto']
random_state	[42]

Table 20: Param grid for model SVC.

Category	Value
criterion	[’gini’, ’entropy’]
splitter	[’best’, ’random’]
max_depth	[None, 5, 10, 15, 20]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]
random_state	[42]

Table 21: Param grid for model DecisionTreeClassifier.

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
KNeighborsClassifier	final_pipeline_2.joblib	{"weights": "uniform", "p": 2, "n_neighbors": 15, "leaf_size": 30, "algorithm": "kd_tree"}	a moment	0.7611
KNeighborsClassifier	final_pipeline_1.joblib	{"weights": "distance", "p": 2, "n_neighbors": 10, "leaf_size": 40, "algorithm": "auto"}	a moment	0.7356
KNeighborsClassifier	final_pipeline_0.joblib	{"weights": "distance", "p": 1, "n_neighbors": 15, "leaf_size": 30, "algorithm": "brute"}	a moment	0.7341

Table 22: Best models results

4.3 Interpretability

This section presents SHAP plots for the best model.

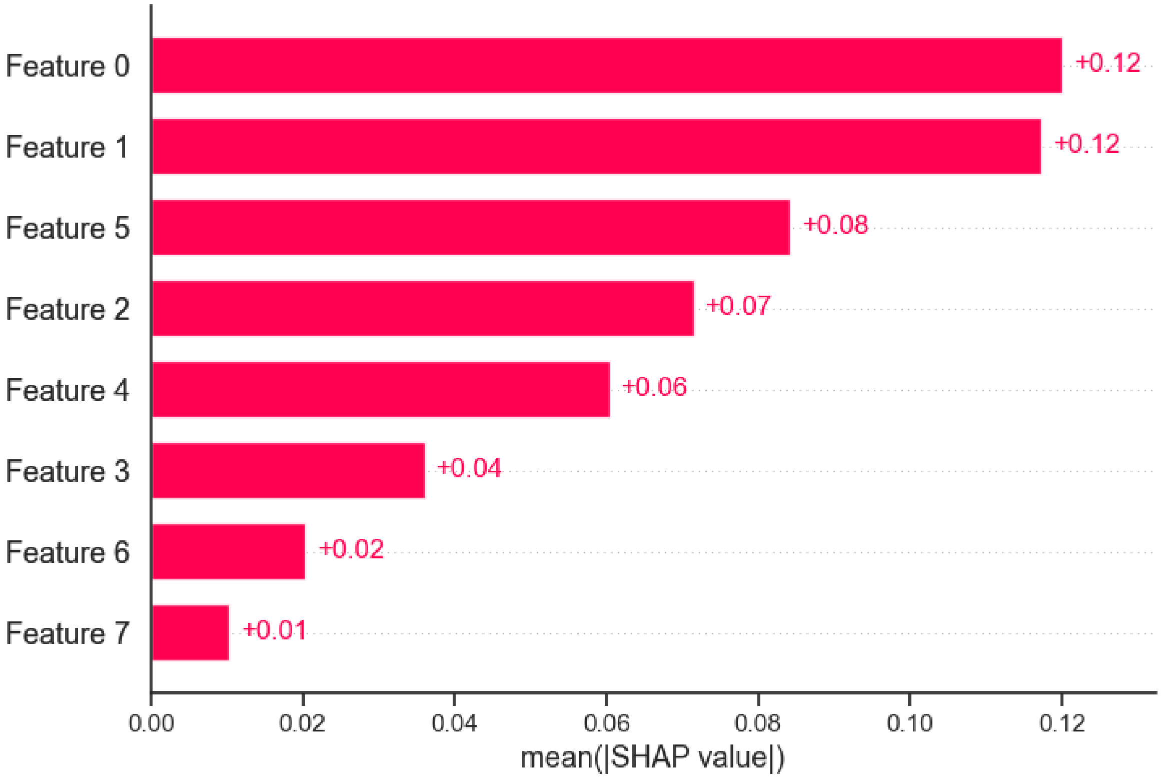

Figure 7: SHAP bar plot for class bar.

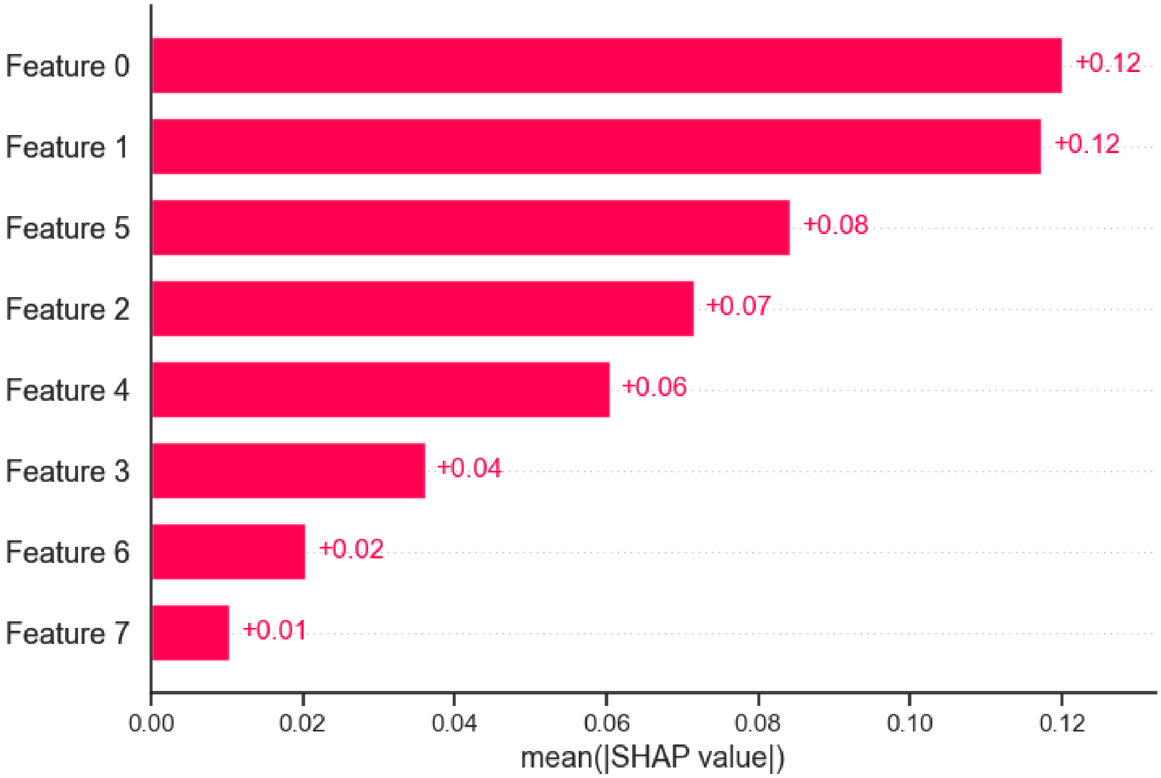

Figure 8: SHAP bar plot for class bar.

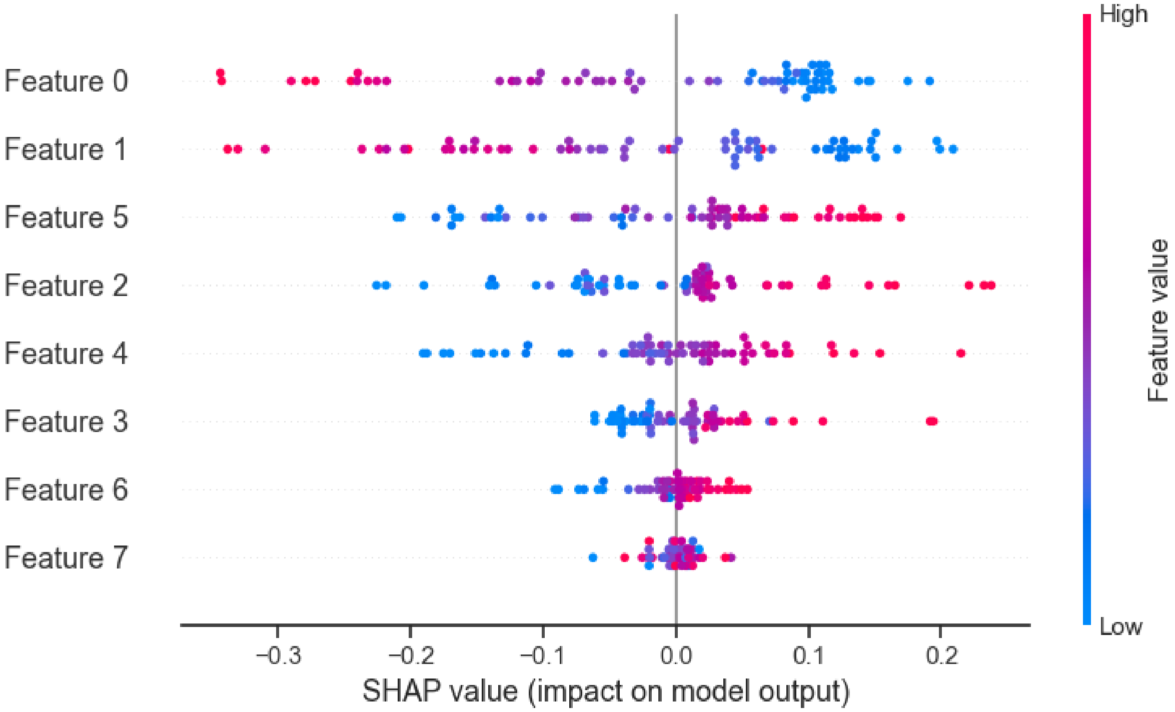

Figure 9: SHAP summary plot for class summary.

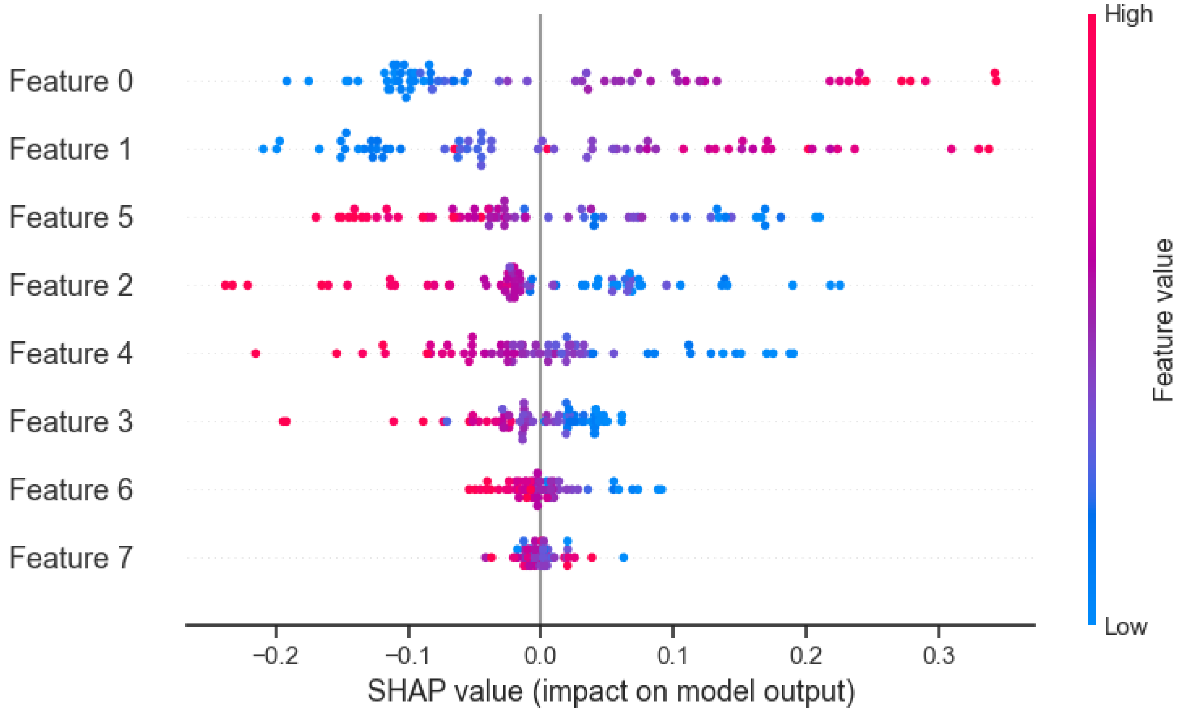

Figure 10: SHAP summary plot for class summary.

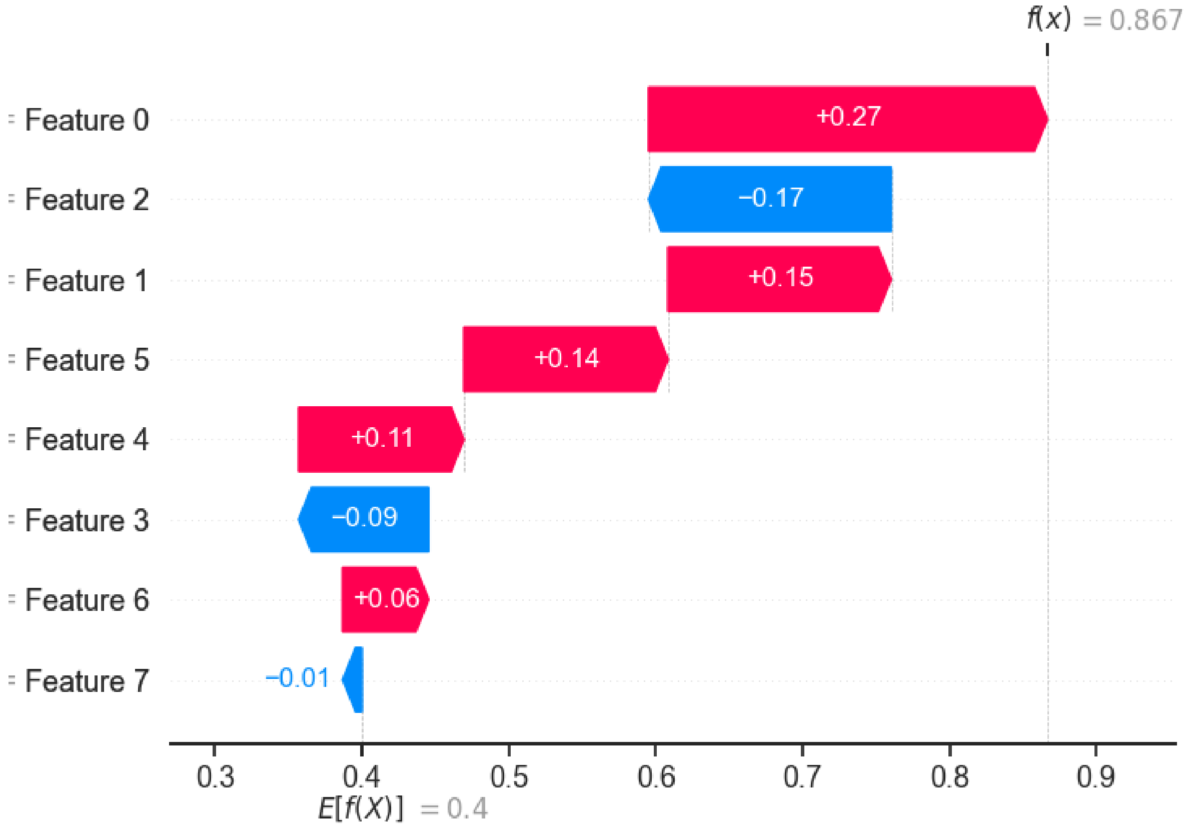

Figure 11: SHAP waterfall plot for class waterfall.

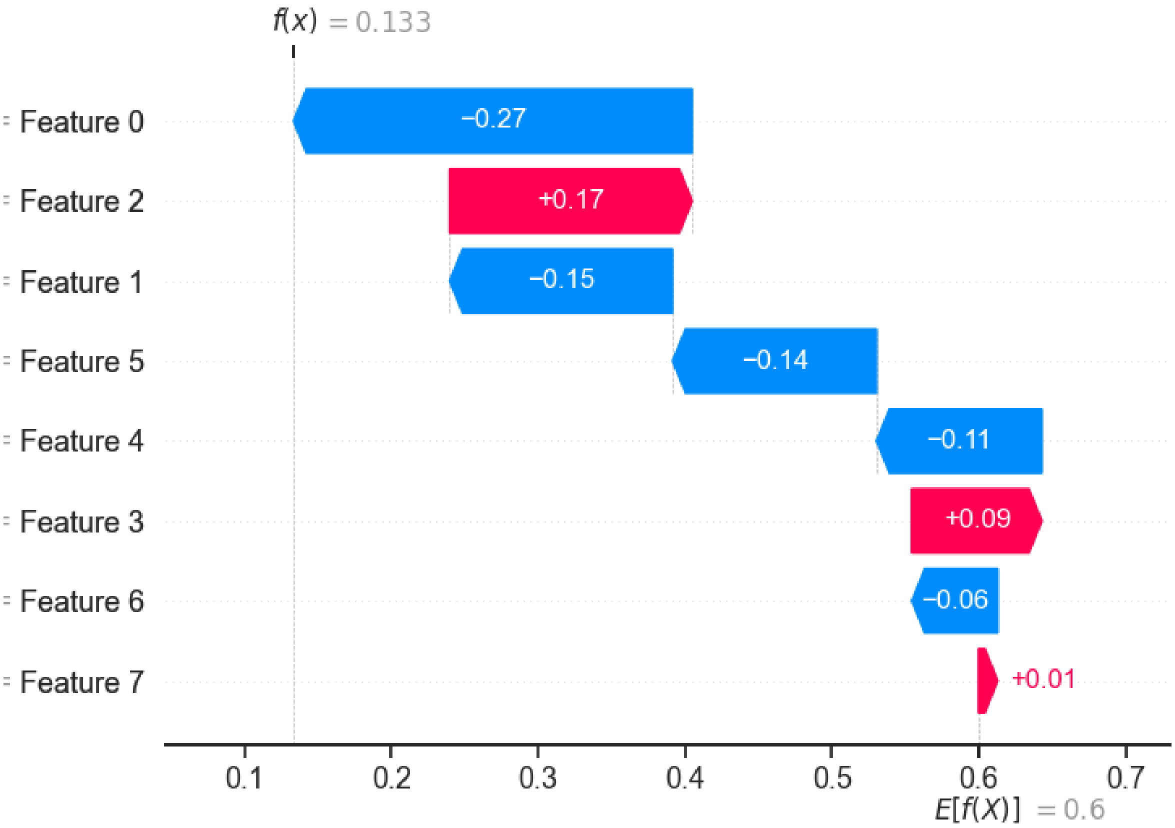

Figure 12: SHAP waterfall plot for class waterfall.