Binary Prediction of Smoker Status

using Bio-Signals

Machine Learning and Data Analysis, 2023-2024

Kevin Cattaneo - S4944382

Riccardo Isola - S4943369

Steps

- Data exploration & analysis
- Data cleaning
- Data visualization
- Feature Engineering
  - Feature extraction
  - Feature reduction
- Machine Learning
  - Model choice & tuning
  - Studying model behaviors

Dataset exploration & analysis - Overview

Objective: to predict the ‘smoking’ status of a person

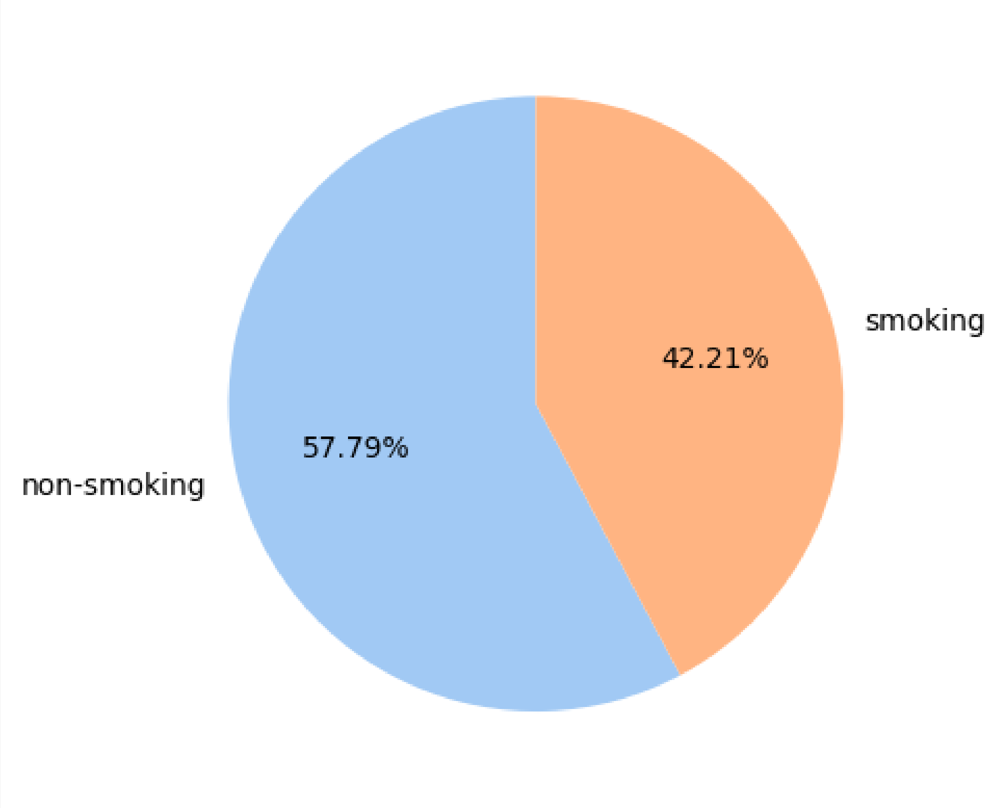
	id	age	height(cm)	weight(kg)	waist(cm)	eyesight(left)	eyesight(right)	hearing(left)	hearing(right)	systolic	...	HDL	LDL	hemoglobin	Urine protein	serum creatinine	AST	ALT	Gtp	dental caries	smoking
0	0	55	165	60	81.0	0.5	0.6	1	1	135	...	40	75	16.5	1	1.0	22	25	27	0	1
1	1	70	165	65	89.0	0.6	0.7	2	2	146	...	57	126	16.2	1	1.1	27	23	37	1	0
2	2	20	170	75	81.0	0.4	0.5	1	1	118	...	45	93	17.4	1	0.8	27	31	53	0	1
3	3	35	180	95	105.0	1.5	1.2	1	1	131	...	38	102	15.9	1	1.0	20	27	30	1	0
4	4	30	165	60	80.5	1.5	1.0	1	1	121	...	44	93	15.4	1	0.8	19	13	17	0	1

	id	age	height(cm)	weight(kg)	waist(cm)	eyesight(left)	eyesight(right)	hearing(left)	hearing(right)	systolic	...	HDL	LDL	hemoglobin	Urine protein	serum creatinine	AST	ALT	Gtp	dental caries	smoking
count	159256.00	159256.00	159256.00	159256.00	159256.00	159256.00	159256.00	159256.00	159256.00	159256.00	...	159256.00	159256.00	159256.00	159256.00	159256.00	159256.00	159256.00	159256.00	159256.0	159256.00
mean	79627.50	44.31	165.27	67.14	83.00	1.01	1.00	1.02	1.02	122.50	...	55.85	114.61	14.80	1.07	0.89	25.52	26.55	36.22	0.2	0.44
std	45973.39	11.84	8.82	12.59	8.96	0.40	0.39	0.15	0.15	12.73	...	13.96	28.16	1.43	0.35	0.18	9.46	17.75	31.20	0.4	0.50
min	0.00	20.00	135.00	30.00	51.00	0.10	0.10	1.00	1.00	77.00	...	9.00	1.00	4.90	1.00	0.10	6.00	1.00	2.00	0.0	0.00
25%	39813.75	40.00	160.00	60.00	77.00	0.80	0.80	1.00	1.00	114.00	...	45.00	95.00	13.80	1.00	0.80	20.00	16.00	18.00	0.0	0.00
50%	79627.50	40.00	165.00	65.00	83.00	1.00	1.00	1.00	1.00	121.00	...	54.00	114.00	15.00	1.00	0.90	24.00	22.00	27.00	0.0	0.00
75%	119441.25	55.00	170.00	75.00	89.00	1.20	1.20	1.00	1.00	130.00	...	64.00	133.00	15.80	1.00	1.00	29.00	32.00	44.00	0.0	1.00
max	159255.00	85.00	190.00	130.00	127.00	9.90	9.90	2.00	2.00	213.00	...	136.00	1860.00	21.00	6.00	9.90	778.00	2914.00	999.00	1.0	1.00

Dataset exploration & analysis - Balancing

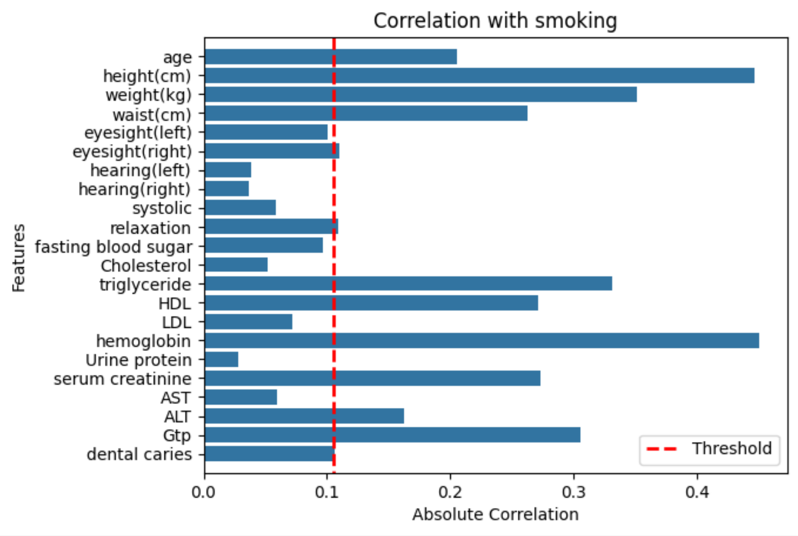
- For the next training phase of the model, we observe the **balance** of the output ‘smoking’

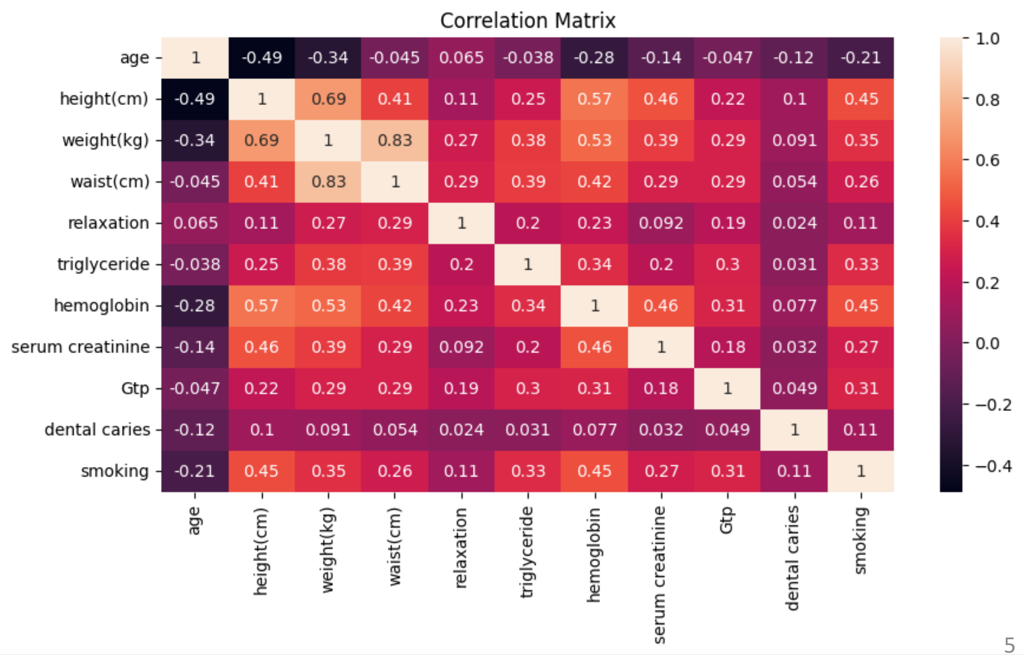
- As we see the smoking feature is more or less balanced.



Dataset exploration & analysis - Correlation

We plot the **correlation** matrix of the most correlated features, above a certain **threshold**.

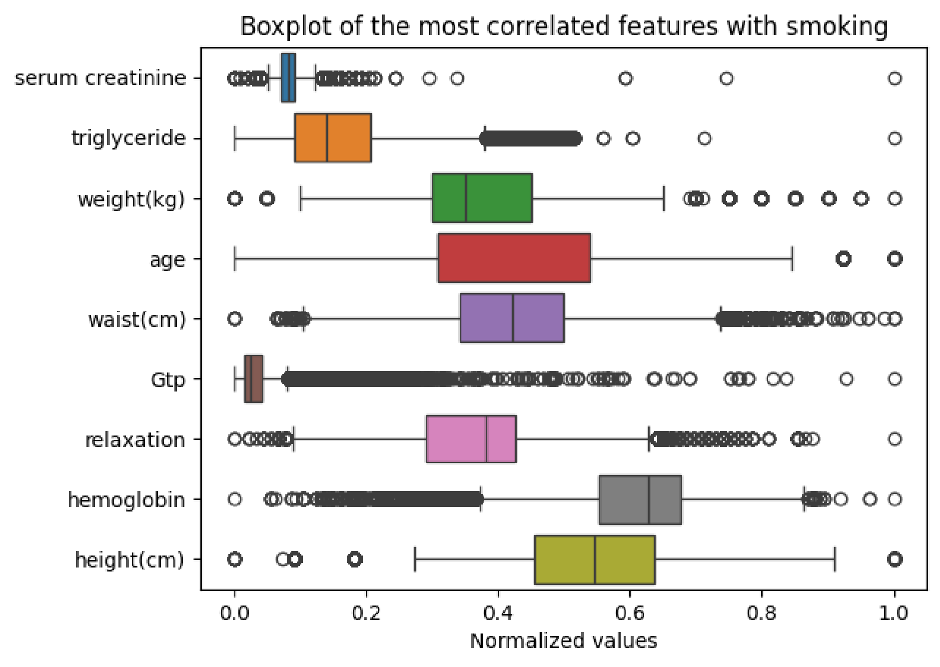




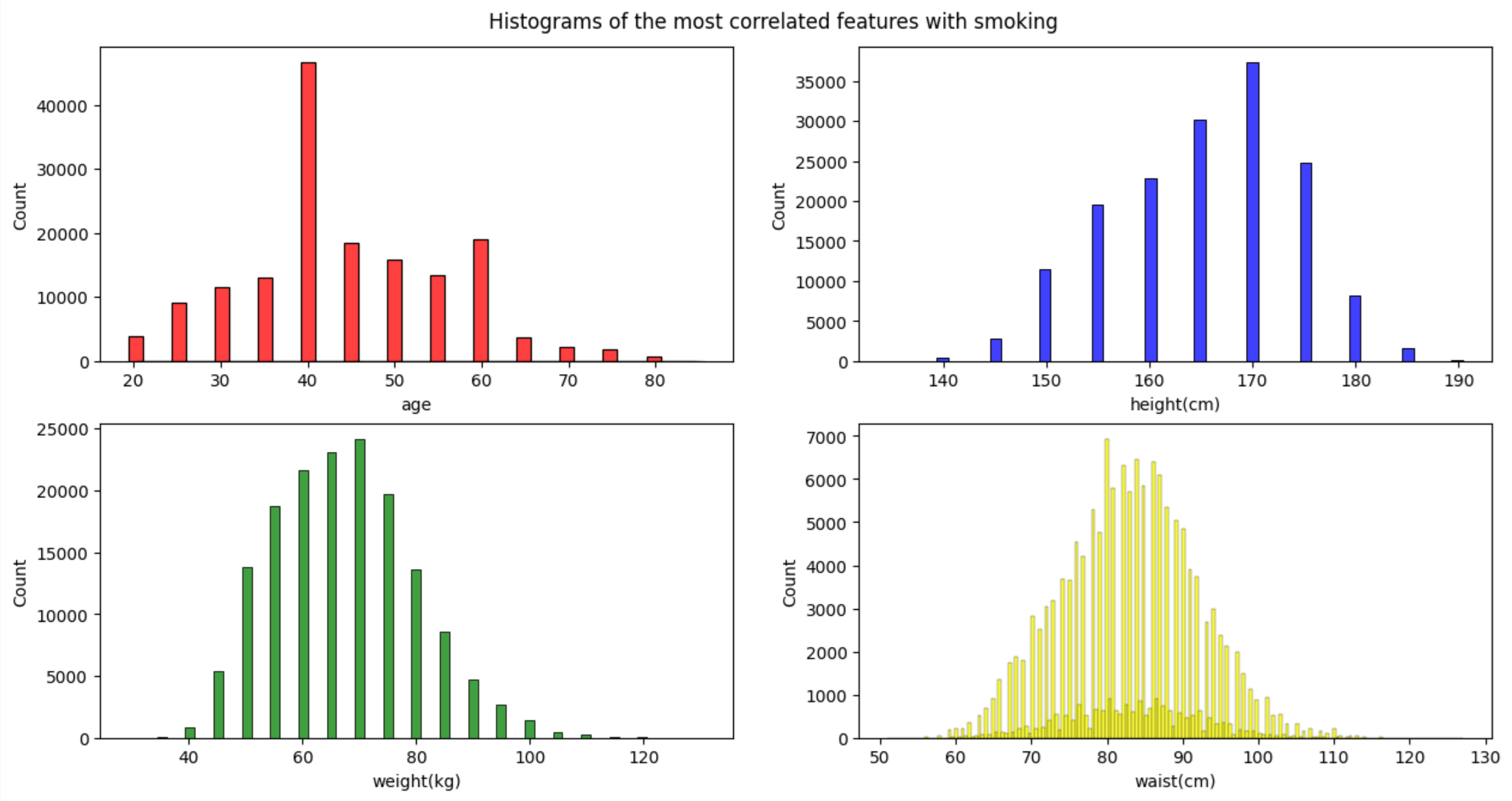
Data cleaning

- Checking for NULL values
  - We found **none**

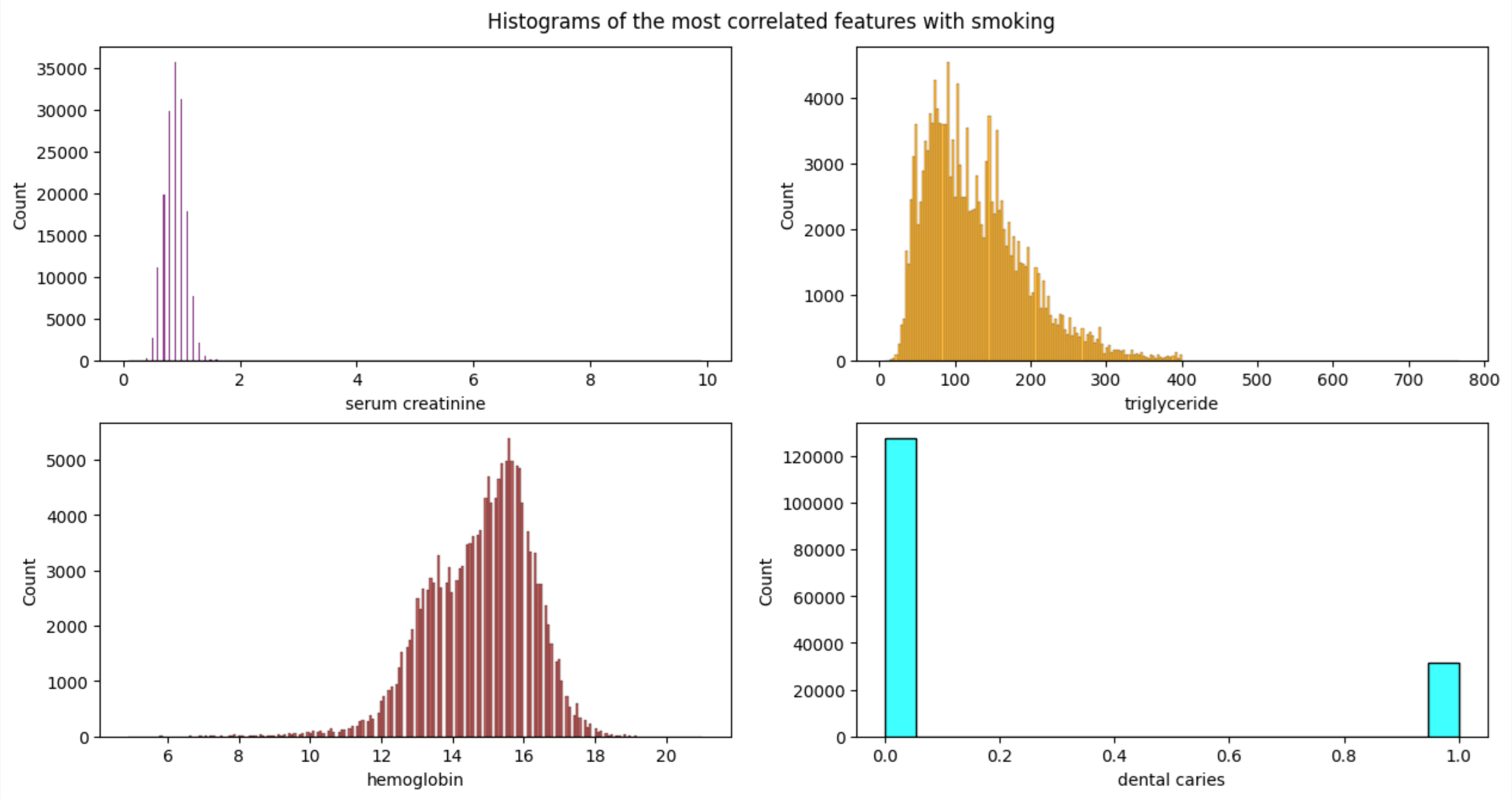
- Checking and removing outliers



Data visualization - Counts per feature I



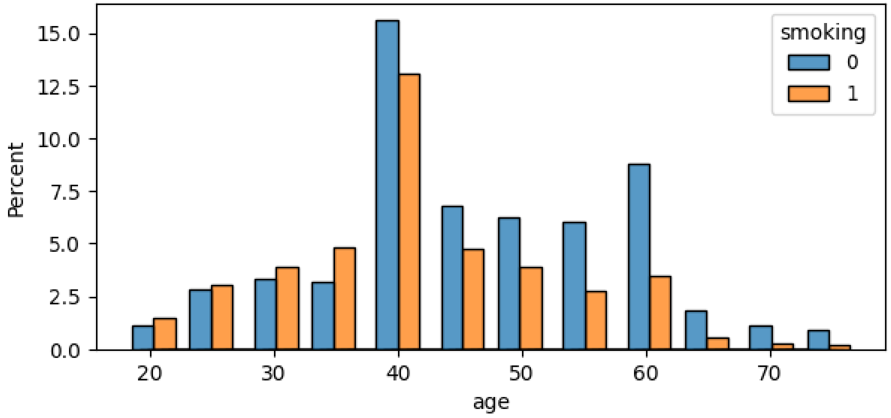
Data visualization - Counts per feature II

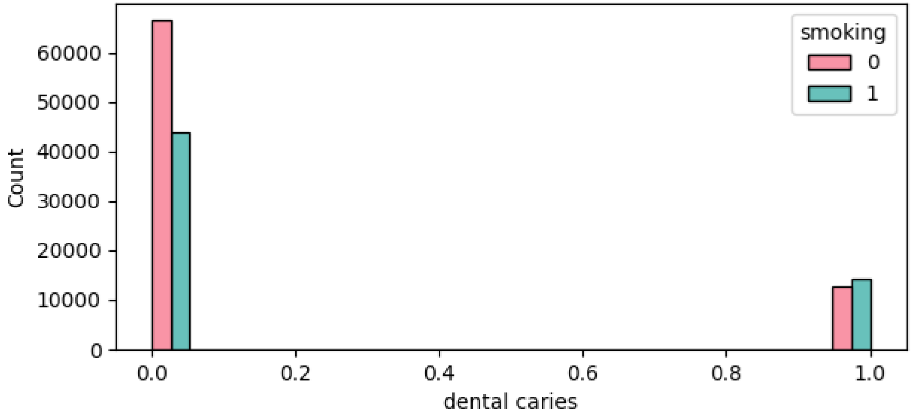


Data visualization - Informative plots

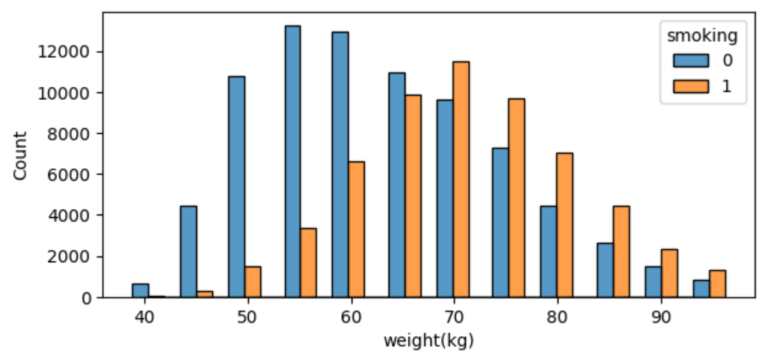
Plot the relation between the correlated feature and the smoking status:

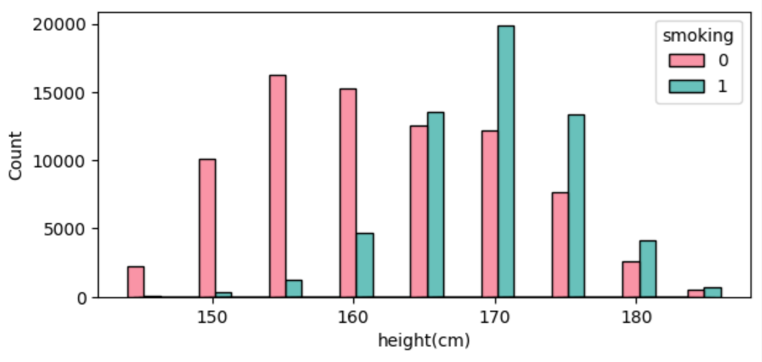
- we see that people of age <= 40 tend to smoke
- surprisingly smoking doesn’t seem related to have dental caries (as we see on the correlation matrix)

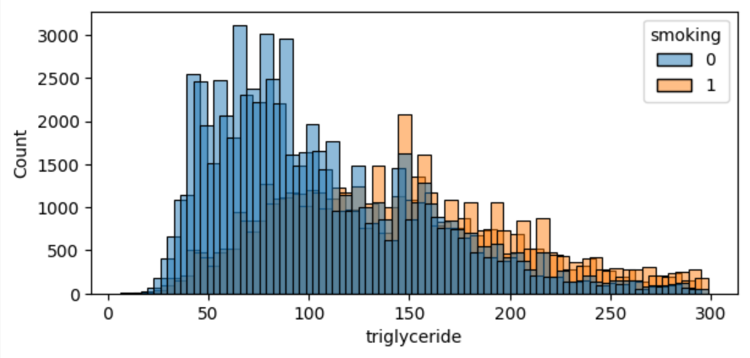


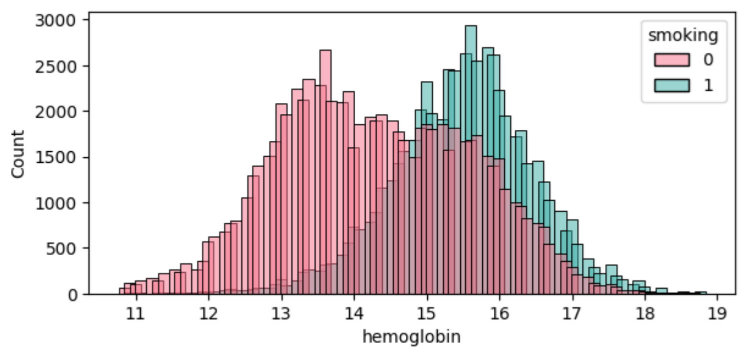


Data visualization - Correlated features plot





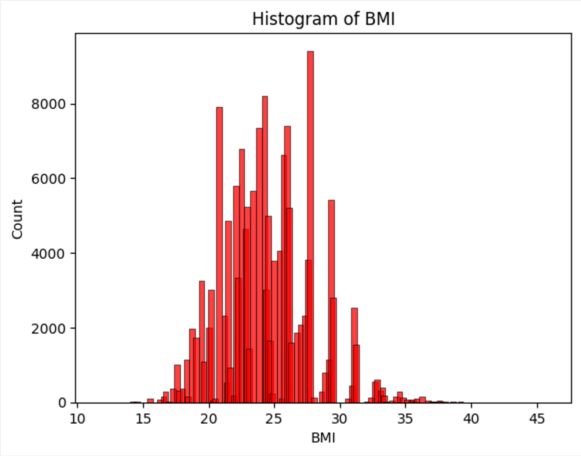


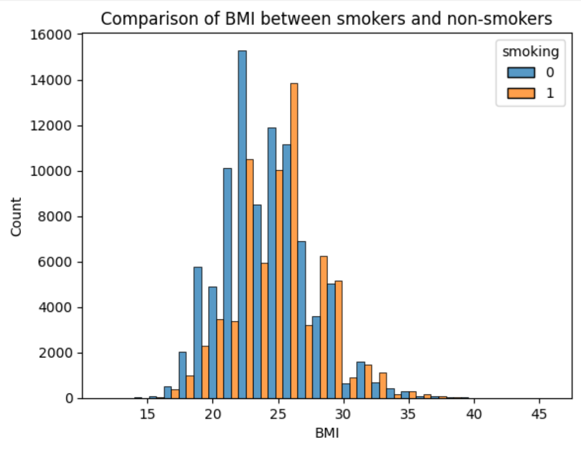


Feature engineering

- Feature **extraction**
  - **BMI** as weight(kg)/height(cm)
  - We can see on the plots no ‘clean’ separation between the two distributions
  - We will not consider it in the first place when training model

- Features **selection**
  - In the training, we will ignore the previous least correlated features, that are under the threshold





Machine Learning – Chosen metrics

Score name	Definition
Accuracy	This metric measures the overall correctness of predictions, representing the ratio of correctly classified instances to the total number of instances in a dataset.
ROC AUC	This metric is used to evaluate the performance of binary classification models. It is the area under the ROC curve, which plots the true positive rate against the false positive rate. A higher ROC AUC score indicates that the model is better at distinguishing between positive and negative cases.

Machine Learning – Considered models

RidgeClassifier
is a linear classification algorithm with L2 regularization, designed to prevent overfitting by penalizing large weights in the model.

RandomForestClassifier
is a model that uses multiple decision trees and aggregates their individual predictions to produce a final output.

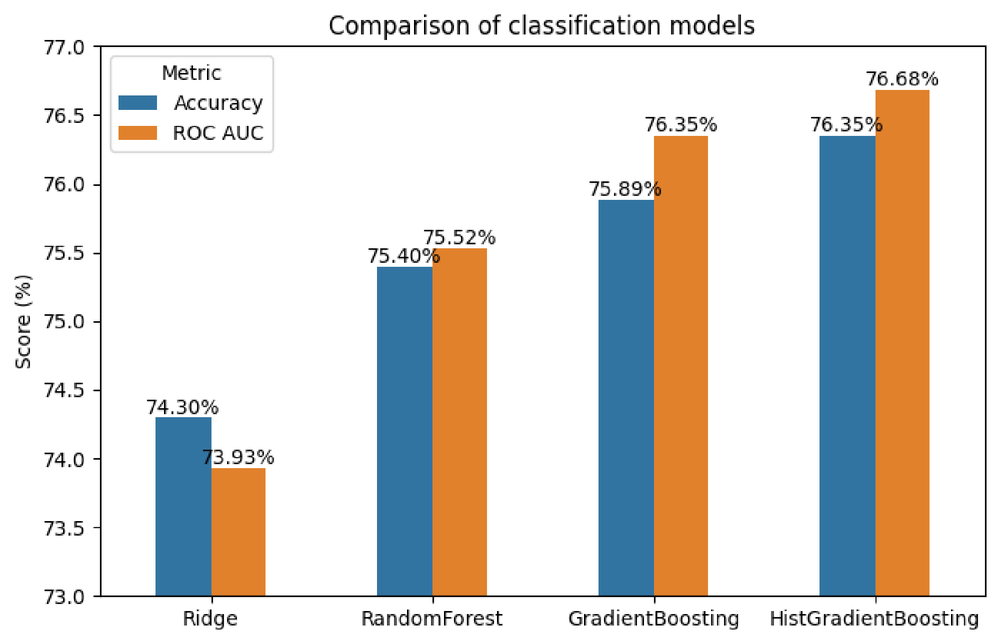
GradientBoostingClassifier
is an ensemble model that builds a series of decision trees sequentially, each correcting the errors of the previous one, to create a robust and accurate predictive model.

HistGradientBoostingClassifier
is a variant of gradient boosting that uses histogram-based techniques for faster training and improved efficiency, making it particularly suitable for large datasets.

Machine Learning - Choosing the model

- For choosing the model, we have done a comparison (without any tuning), observing how they perform on our data.

- We see that **HistGradientBoostingClassifier** performs the best.



Machine Learning – Model Tuning

Once we have chosen the model, we do the so called **model tuning**, that is searching for the best **hyperparameters** that make the model perform better.

We used a **GridSearchCV** with different hyperparameters (see next)

Metric	Score Before Tuning	Score After Tuning	Improvement
accuracy	75.40%	76.48%	1.08%
ROC AUC	75.52%	76.83%	1.31%

Machine Learning – Model Hyperparameters

Name	Definition	Possible values	Selected value
max_iter	maximum iteration of our model	[250, 300, 350]	300
early_stopping	regularization used to avoid overfitting	[True, False]	False
max_depth	maximum depth of each tree	[5, 7, None*]	5
validation_fraction	proportion of training data to set aside as validation data for early stopping	[0.0001, 0.001, 0.01]	0.0001

* None = no limit

Machine Learning – Adding new features

Now we see how the model behave when we try to simplify our data.
The new features will substitute the original ones.

diastolic_pressure	
diastolic < 80	**Normal**
80 <= diastolic <= 89	**Elevated**
90 <= diastolic <= 99	**High**
diastolic >= 100	**Very High**

hemoglobin_lvl	
age < 18 & hemoglobin < 11.5 or age > 18 & hemoglobin < 10.4	**Low**
age < 18 & hemoglobin in (11.5, 16.3) or age > 18 & hemoglobin in (10.4, 17.1)	**Normal**
age < 18 & hemoglobin > 16.3 or age > 18 & hemoglobin > 17.1	**High**

Machine Learning – Model behaviors

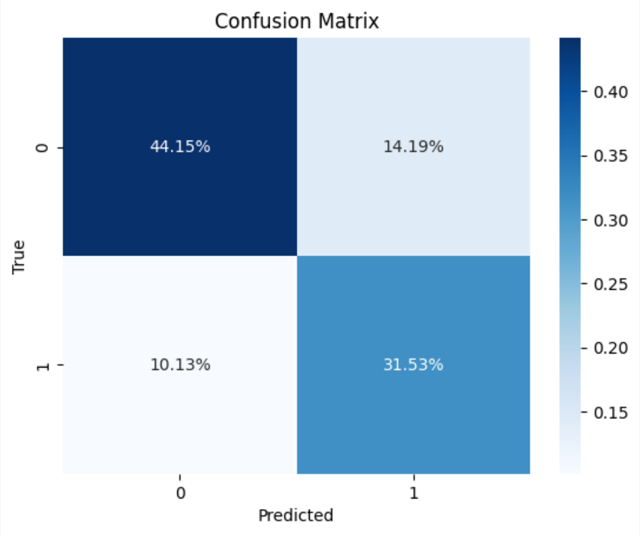
Metric	Score Before	Score After BMI	Score After hemoglobin_lvl	Score After diastolic_pressure
accuracy	76.48%	76.45%	75.69%	75.67%
ROC AUC	76.83%	76.65%	75.69%	75.68%

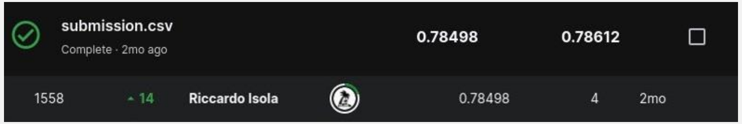
Note: each column (step) represent the adding of the new feature to the previous step plus a new tuning of the model

Machine Learning – Final results

What we notice from the previous model behaviors is that:

- When we touch **highly correlated** column it seems that the model is more sensible to changes, as if it was a ‘delicate’ feature
- On the other hand, touching **less correlated** features has a minor impact on the model, in both cases of improvement or not
- At the end of the kaggle competition we obtained the results in the dark figure





Position

Score

Questions?