Cerebrovascular Accident Predictor Report

Alexandru MIHAI, 333CA

October 20, 2024

1 Exploratory Data Analysis

1.1 Feature analysis

Numerical features for AVC: mean blood sugar level, body mass indicator, years old, analysis results, biological age index.

Categorial features for AVC: cardiovascular issues, job category, sex, tobacco usage, high blood pressure, married, living area, chaotic sleep.

Target feature for AVC: cerebrovascular accident.

1.2 Class balancing analysis

For the AVC task, the class balance is really unsatisfactory (only about 5% are AVC positive) which might impact the training process as there might be a high bias for the majority class.

1.3 Correlation analysis

For numerical features correlation I have used the Pearson test.

For categorial features correlation I have used the Chi-Square test with null hypothesis of correlation.

Below you have the heatmap for numerical features correlation and Chi-Square test results for categorial features respectively.

Feature	Non nan values	Mean	Std	Min value	25%	50%	75%	Max value
mean_blood_sugar_level	5110	106.148	45.283	55.120	77.245	91.885	114.090	271.740
body_mass_indicator	4909	28.893	7.854	10.300	23.500	28.100	33.100	97.600
years_old	5110	46.569	26.594	0.080	26.000	47.000	63.750	134.000
analysis_results	4599	323.523	101.577	104.830	254.646	301.032	362.823	756.808
biological_age_index	5110	134.784	50.399	-15.109	96.710	136.375	172.507	266.986

Table 1: AVC dataset numeric features and statistics.

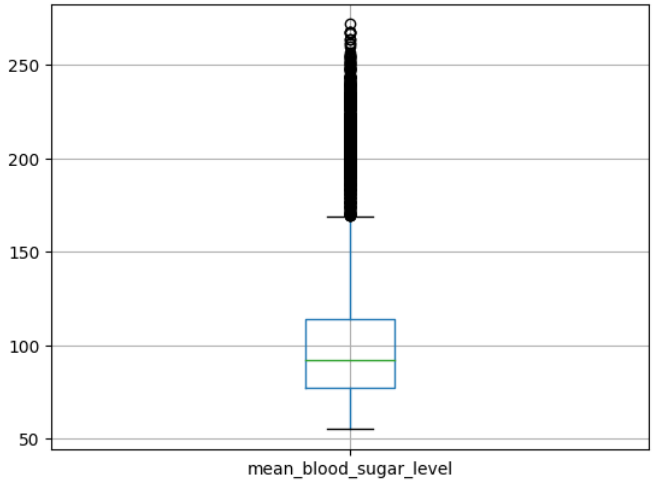

Figure 1: AVC mean blood sugar level in dataset.

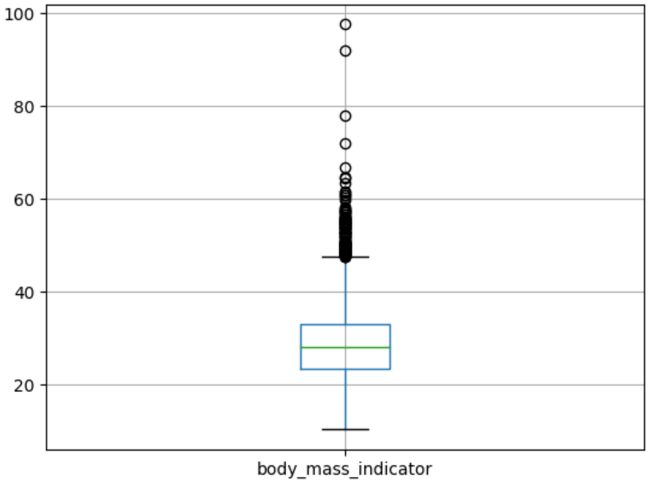

Figure 2: AVC body mass indicator in dataset.

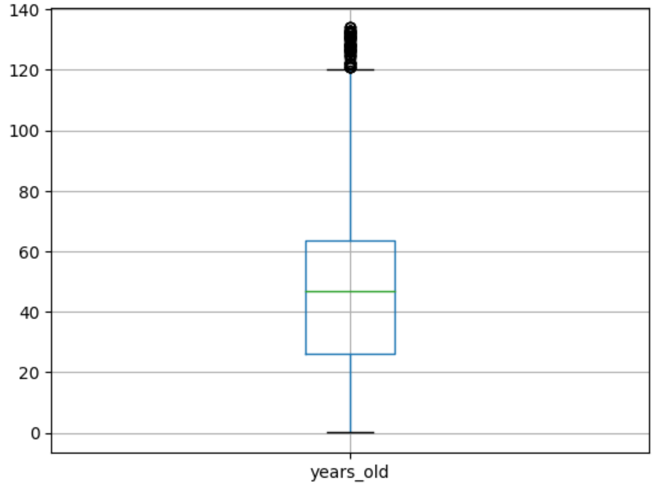

Figure 3: AVC years old in dataset.

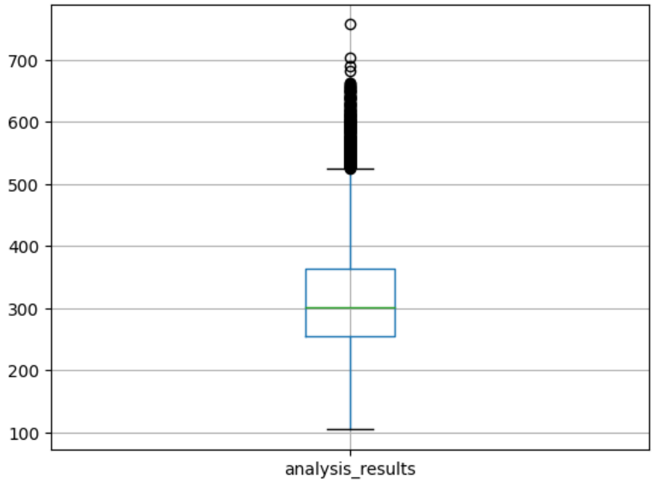

Figure 4: AVC analysis results in dataset.

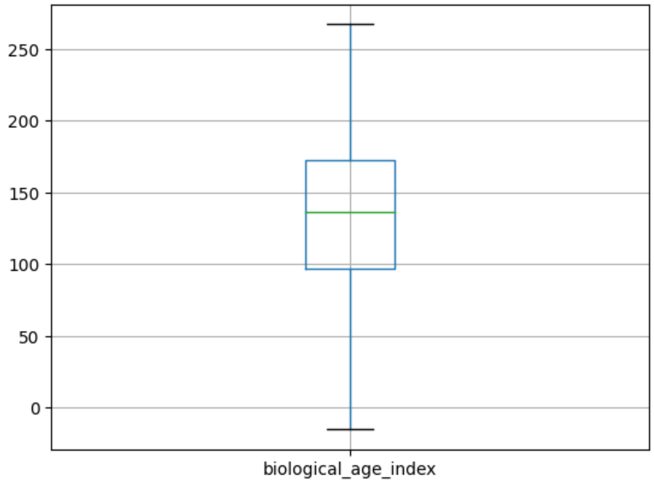

Figure 5: AVC biological age index in dataset.

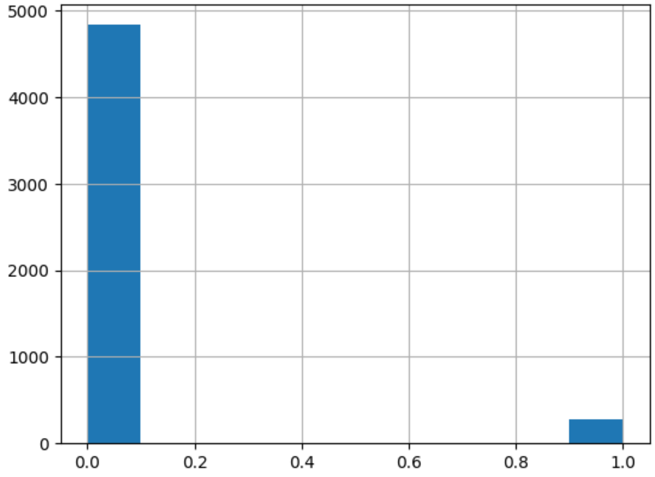

Figure 6: AVC cardiovascular issues histogram in dataset.

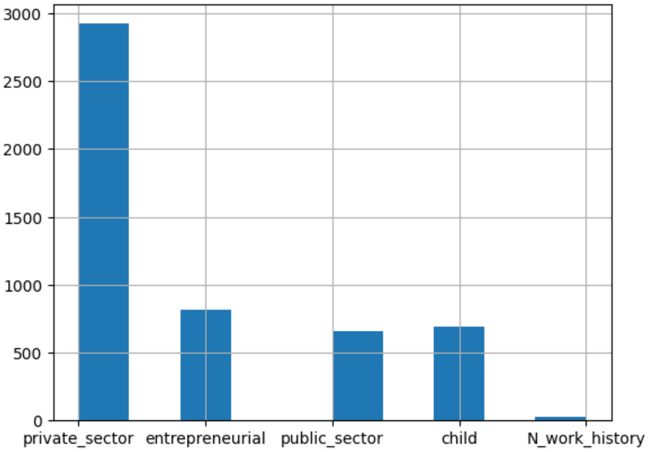

Figure 7: AVC job category histogram in dataset.

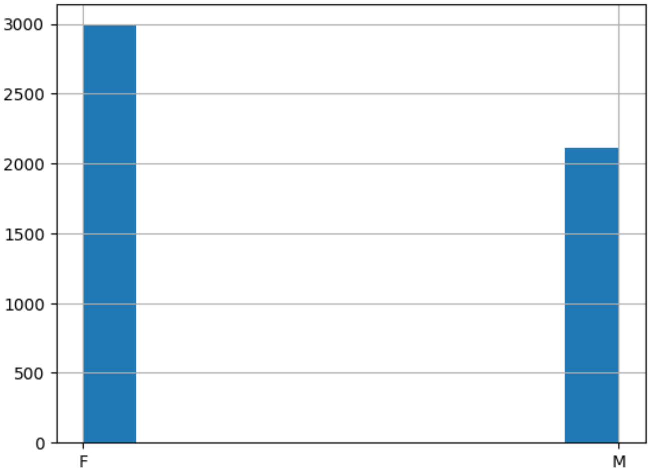

Figure 8: AVC sex histogram in dataset.

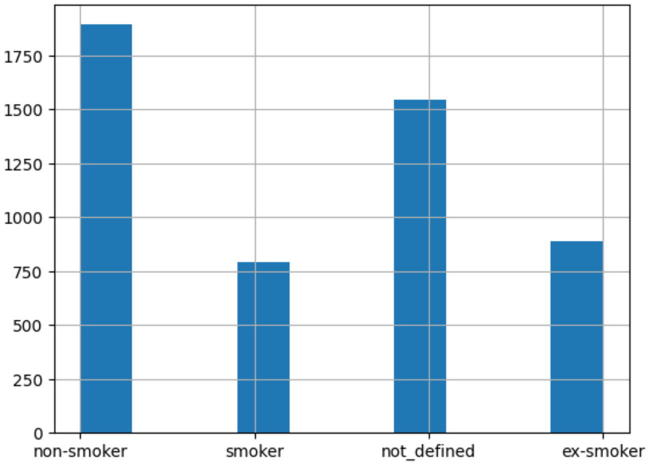

Figure 9: AVC tobacco usage histogram in dataset.

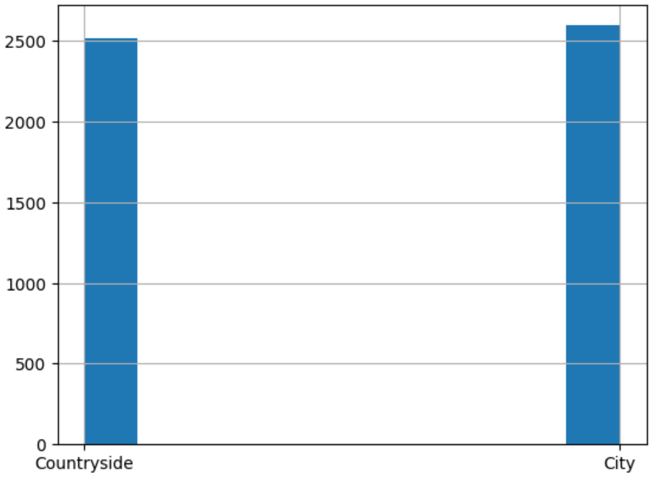

Figure 10: AVC living area histogram in dataset.

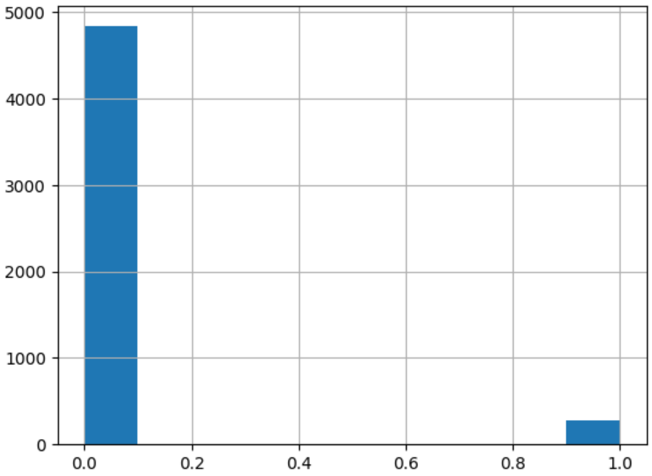

Figure 11: AVC chaotic sleep histogram in dataset.

Feature	Non nan values	Unique values
cardiovascular_issues	5110	2
job_category	5110	5
sex	5110	2
tobacco_usage	5110	4
high_blood_pressure	5110	2
married	4599	2
living_area	5110	2
chaotic_sleep	5110	2

Table 2: AVC dataset categorial features and statistics.

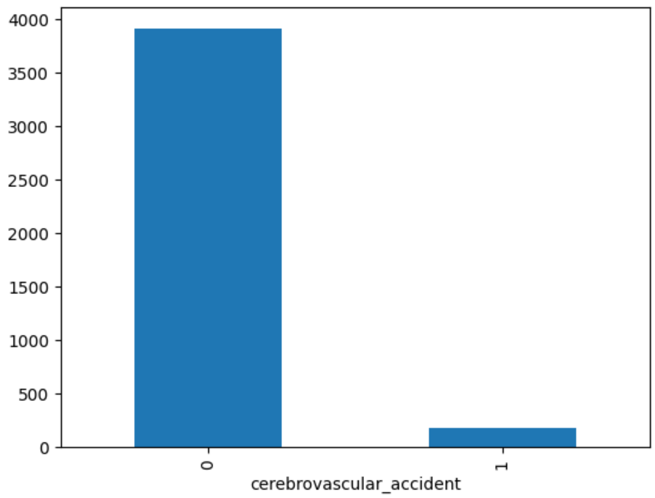

Figure 12: AVC class balance for train set.

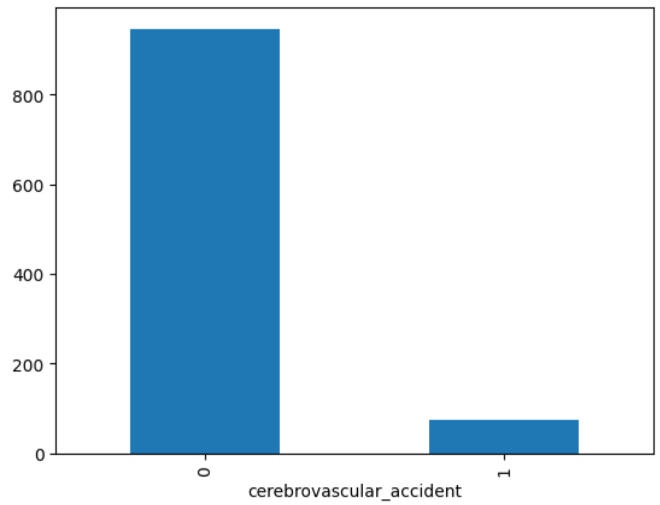

Figure 13: AVC class balance for test set.

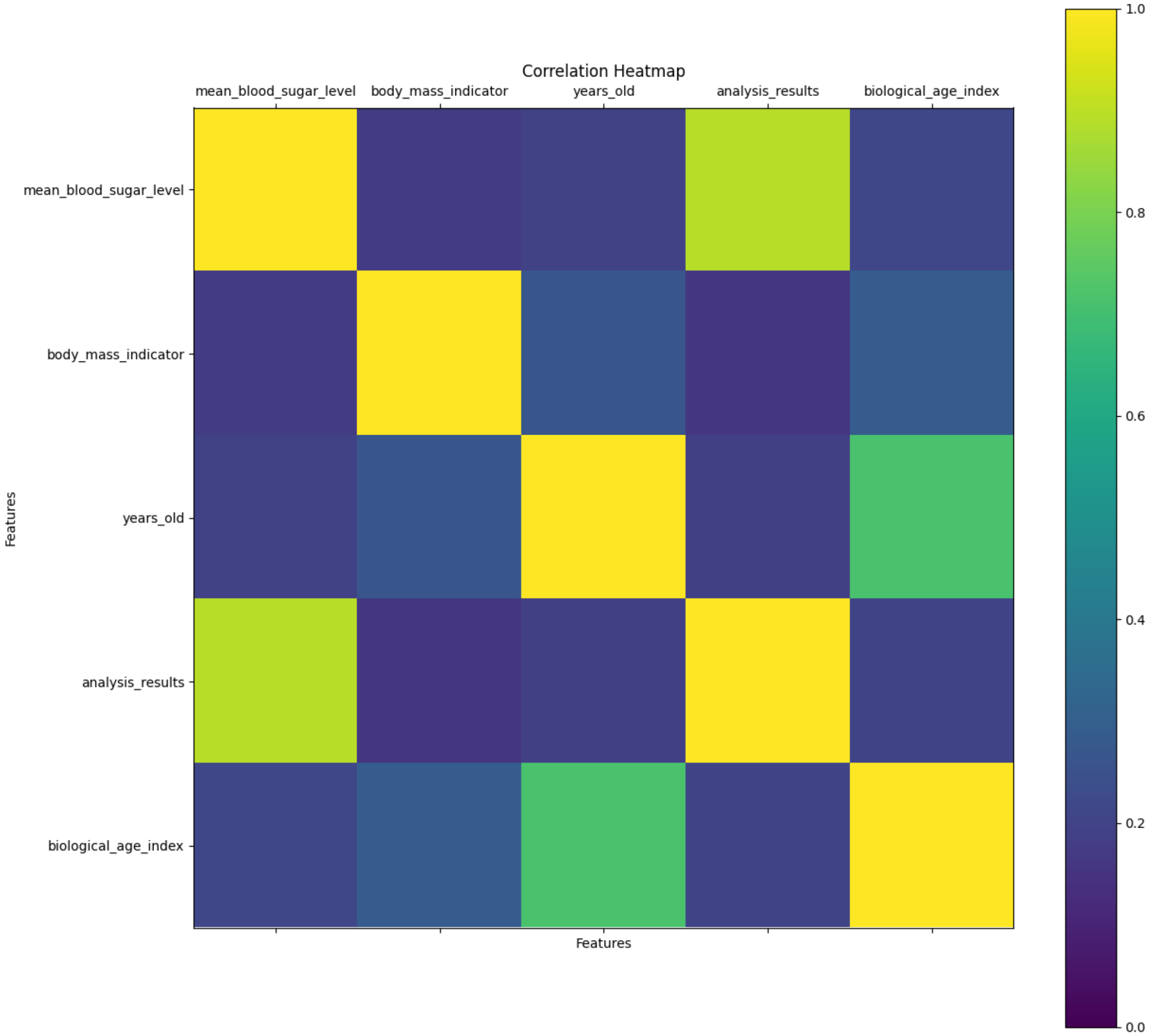

Figure 14: AVC numerical features heatmap.

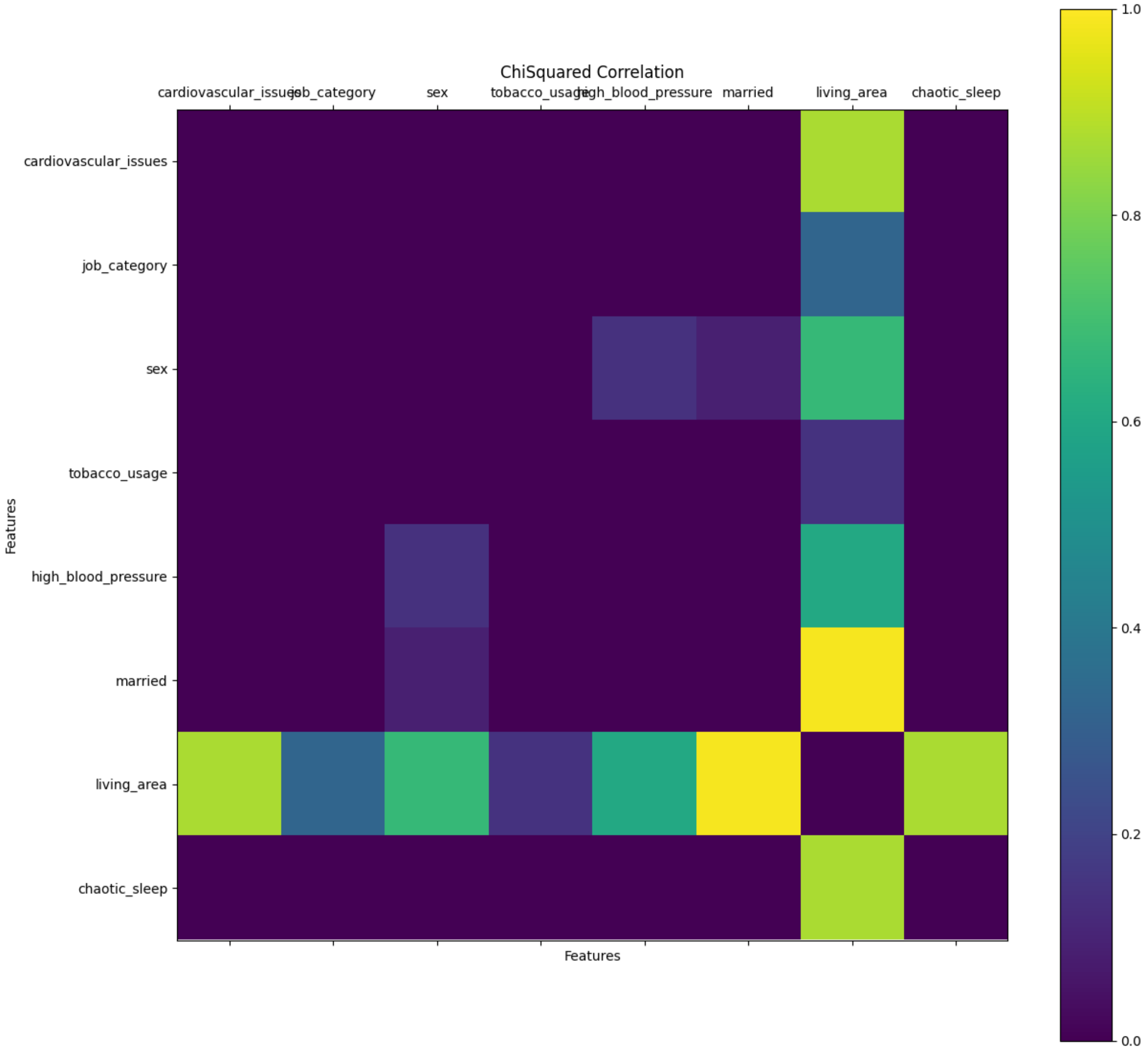

Figure 15: AVC categorial feature Chi-Square results.

2 Data Preprocessing

2.1 Missing values imputation

I have decided to use the mean for numerical values and the most frequent for the categorial values taking into consideration that this imputation hinders the least the learning process (the mean/ most frequent values does not have a high weight in the learning process).

I have also tried the multivariate imputation techniques, however they proved similar results.

2.2 Outliers

I have replaced the outliers values with np.nan before the imputation process. Therefore they are treated in the same way as missing values.

2.3 Redundant features

Due to high correlation between mean blood sugar level and analysis results numerical features, I have decided to remove analysis results (it also has some missing values, it aids the imputation process). Also for the AVC task, years old and biological age index have similar correlation values, thus I have remove the years old feature.

For categorial features, chaotic sleep and cardiovascular issues are redundant (thus I have removed chaotic-sleep). Most of the features are correlated apart from the living area. Therefore, I have decided to keep living area, sex, married, high blood pressure and cardiovascular issues considering that those are the least correlated to others.

3 Logistic regression

Chosen encoding: one hot encoding for categorial features, label encoding for target feature.

Hyperparameters - lab implementation: number of epochs - 500, learning rate - 0.001, l2-factor - 0.01.

Used regularization - lab implementation: l2 regularization (Ridge).

Hyperparameters - library implementation: solver - liblinear, class_weight - $0 : 1$, $1 : 1.5$, number of epochs - 500.

Used regularization - library implementation: l2 regularization (Ridge)

4 Multi-Layer Perceptron

NN architecture: linear - 300 hidden_units, ReLU, linear - 300 hiddent_units.

Hyperparameters - lab implementation: number of epochs - 20, learning rate - 0.001, batch-size - 32, mode - SGD.

Hyperparameters - library implementation: solver - adam, activation - ReLU, batch-size - 32, number of epochs - 20, alpha - 0.05, learning rate - 0.001, adaptive learning rate.

5 Comparative analysis

First of all, I have tried to balance the classes (because they were 95% - 5% split which induced high bias on the majority class). Therefore the train set is really small (about 1000 samples), however it has a better class balance (10% samples were taken from the majority class, 90% samples were taken from the minority class). I have tried many other balancing techniques, however none proved really useful.

The algorithms performed similar taking into consideration their accuracy for the test set (91%), except library MLP probably due to non optimal hyperparameters. Generally, logistic regression proved more accurate rather than MLP considering the F1 score, however the difference is so small and the statistics are so poorly that I do not consider the model being the primary reason for such difference.

In my humble opinion, the unbalanced classes, as well as the overall data provided was inconclusive, trying a lot of different hyper-parameters and techniques to improve the accuracy, however it failed each time, always showing high bias for the majority class.

Algorithm	Implementation	Target value	Accuracy	Precision	Recall	F1
Logistic Regression	lab	0	0.84	0.85	0.97	0.91
Logistic Regression	lab	1	0.84	0.54	0.18	0.27
Logistic Regression	library	0	0.84	0.85	0.97	0.91
Logistic Regression	library	1	0.84	0.52	0.17	0.25
MLP	lab	0	0.83	0.84	0.97	0.90
MLP	lab	1	0.83	0.42	0.11	0.17
MLP	**library**	0	**0.87**	**0.88**	**0.98**	**0.93**
MLP	**library**	1	**0.87**	**0.75**	**0.32**	**0.45**

Table 3: ML algorithms analysis for AVC - train test.

Algorithm	Implementation	Target value	Accuracy	Precision	Recall	F1
Logistic Regression	**lab**	0	**0.91**	**0.94**	**0.97**	**0.95**
Logistic Regression	**lab**	1	**0.91**	**0.33**	**0.21**	**0.26**
Logistic Regression	**library**	0	**0.91**	**0.94**	**0.97**	**0.95**
Logistic Regression	**library**	1	**0.91**	**0.33**	**0.21**	**0.26**
MLP	**lab**	0	**0.91**	**0.94**	**0.97**	**0.95**
MLP	lab	1	**0.91**	0.32	0.16	0.21
MLP	library	0	0.89	0.93	0.95	0.94
MLP	library	1	0.89	0.20	0.15	0.17

Table 4: ML algorithms analysis for AVC - test test.

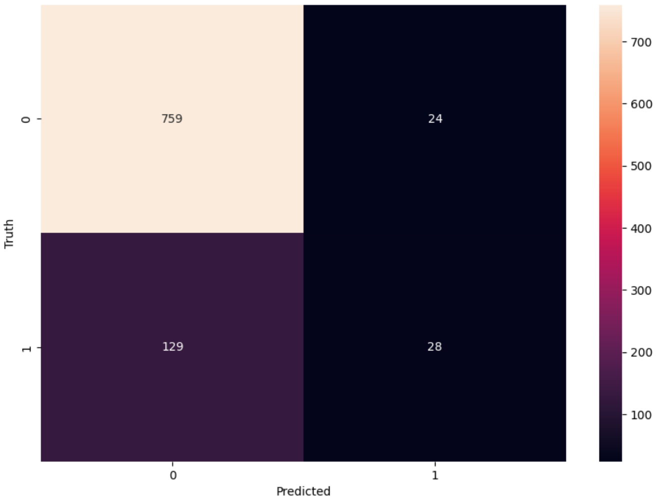

Figure 16: AVC - logistic regression, lab implementation, train confusion matrix.

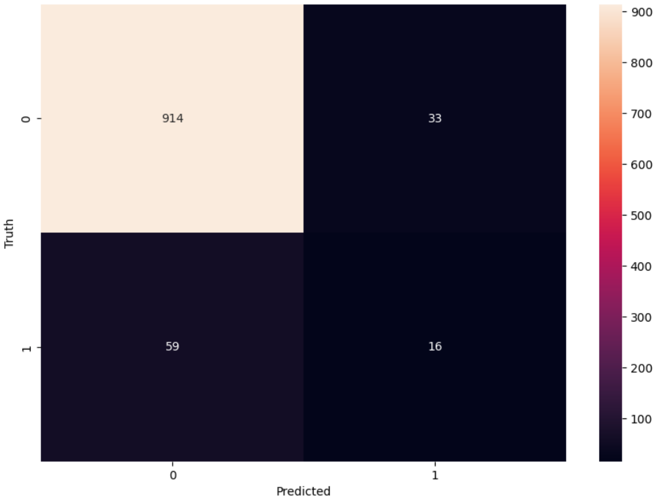

Figure 17: AVC - logistic regression, lab implementation, test confusion matrix.

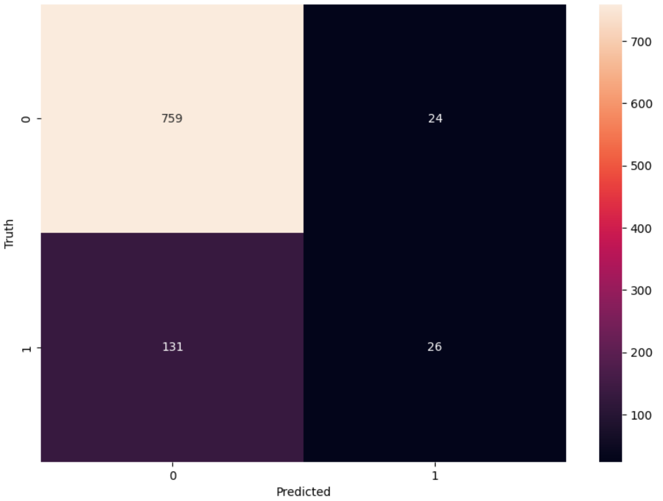

Figure 18: AVC - logistic regression, library implementation, train confusion matrix.

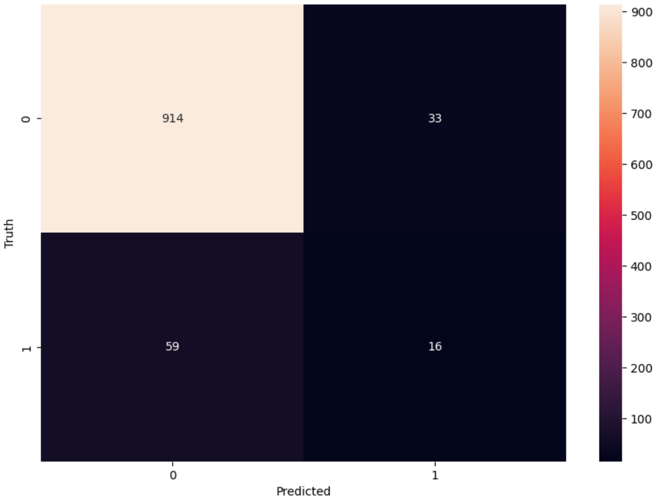

Figure 19: AVC - logistic regression, library implementation, test confusion matrix.

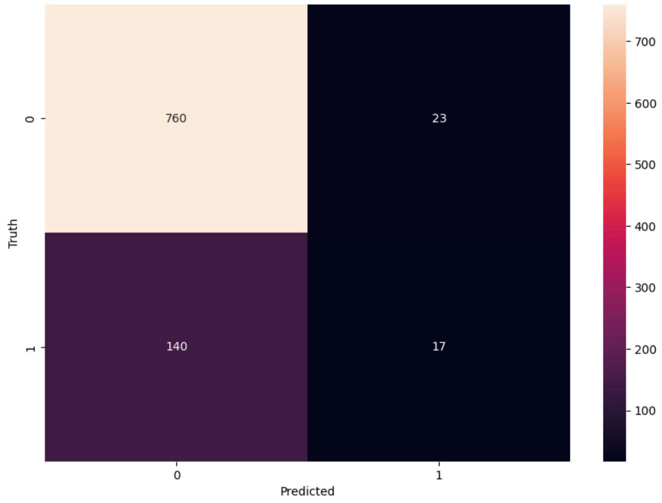

Figure 20: AVC - MLP, lab implementation, train confusion matrix.

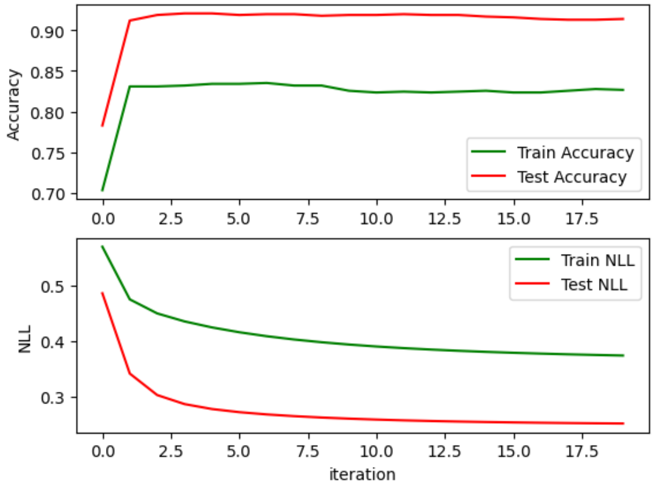

Figure 21: AVC - MLP, lab implementation, accuracy and loss graph.

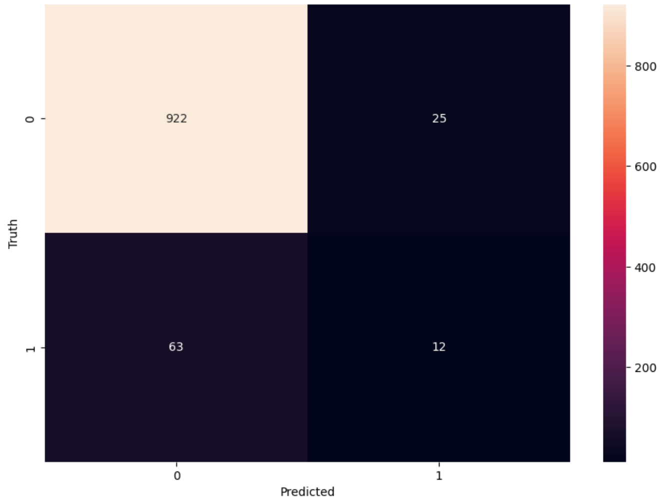

Figure 22: AVC - MLP, lab implementation, test confusion matrix.

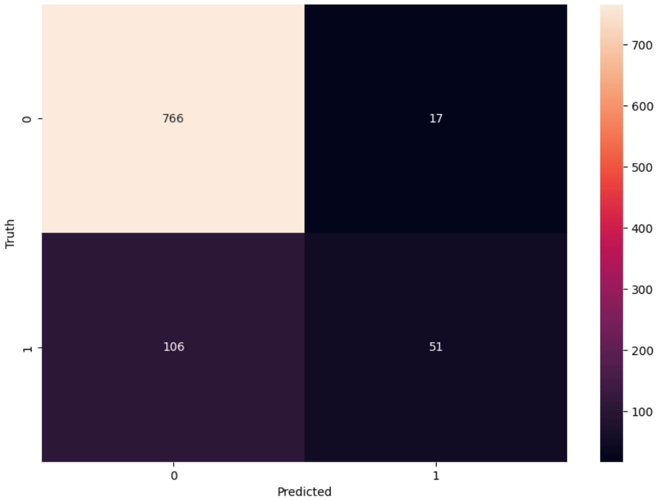

Figure 23: AVC - MLP, library implementation, train confusion matrix.

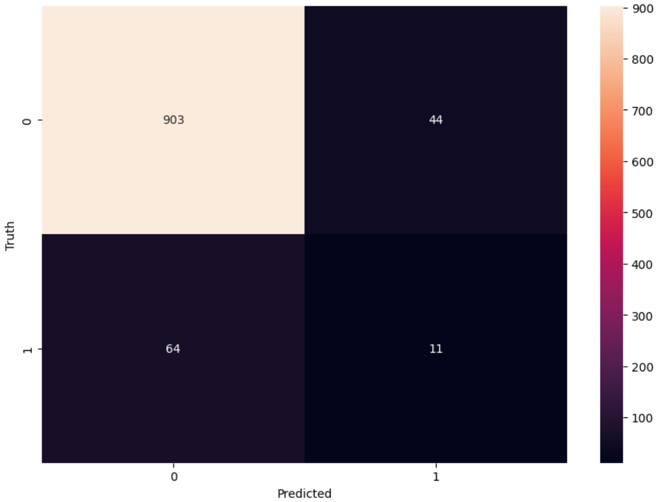

Figure 24: AVC - MLP, library implementation, test confusion matrix.