Artificial Intelligence - Homework 2

I. Data Exploration

A. Dataset Content and Missing Values

AVC Dataset
Data columns (total 14 columns):

#	Column	Non-Null Count	Dtype
0	mean_blood_sugar_level	5110 non-null	float64
1	cardiovascular_issues	5110 non-null	int64
2	job_category	5110 non-null	object
3	body_mass_indicator	4909 non-null	float64
4	sex	5110 non-null	object
5	tobacco_usage	5110 non-null	object
6	high_blood_pressure	5110 non-null	int64
7	married	4599 non-null	object
8	living_area	5110 non-null	object
9	years_old	5110 non-null	float64
10	chaotic_sleep	5110 non-null	int64
11	analysis_results	4599 non-null	float64
12	biological_age_index	5110 non-null	float64
13	cerebrovascular_accident	5110 non-null	int64

Missing values in the AVC datasets:
AVC_test
* none

AVC_train
* body_mass_indicator: 201
* married: 511
* analysis_results: 511

AVC_full
* body_mass_indicator: 201
* married: 511
* analysis_results: 511

SalaryPrediction Dataset
Data columns (total 17 columns):

#	Column	Non-Null Count	Dtype
0	fnl	7999 non-null	int64
1	hpw	7199 non-null	float64
2	relation	7999 non-null	object
3	gain	7999 non-null	int64
4	country	7999 non-null	object
5	job	7999 non-null	object
6	edu_int	7999 non-null	int64
7	years	7999 non-null	int64
8	loss	7999 non-null	int64
9	work_type	7999 non-null	object
10	partner	7999 non-null	object
11	edu	7999 non-null	object
12	gender	7199 non-null	object
13	race	7999 non-null	object
14	prod	7999 non-null	int64
15	gtype	7999 non-null	object
16	money	7999 non-null	object

Missing values in the SalaryPrediction datasets:
SalaryPrediction_test
* none

SalaryPrediction_train
* hpw: 800
* gender: 800

SalaryPrediction_full
* hpw: 800
* gender: 800

B. Attribute Statistics

AVC Dataset (Numerical Attributes)

Attribute	Number of Examples	Mean	Standard Deviation	Minimum	25th Percentile	Median	75th Percentile	Maximum
mean_blood_sugar_level	5110	106.15	45.28	55.12	77.25	91.88	114.09	271.74
body_mass_indicator	4909	28.89	7.85	10.3	23.5	28.1	33.1	97.6
years_old	5110	46.57	26.59	0.08	26.0	47.0	63.75	134.0
analysis_results	4599	323.52	101.58	104.83	254.65	301.03	362.82	756.81
biological_age_index	5110	134.78	50.40	-15.11	96.71	136.37	172.51	266.99

AVC Dataset (Categorical Attributes)

Attribute	Number of Examples (Non-Null)	Number of 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
cerebrovascular_accident	5110	2

SalaryPrediction Dataset (Numerical Attributes)

Attribute	Number of Examples	Mean	Standard Deviation	Minimum	25th Percentile	Median	75th Percentile	Maximum
fnl	9999	190352.90	106070.86	19214.0	118282.5	178472.0	237311.0	1455435.0
hpw	9199	40.42	12.52	1.0	40.0	40.0	45.0	99.0
gain	9999	979.85	7003.80	0.0	0.0	0.0	0.0	99999.0
edu_int	9999	14.26	24.77	1.0	9.0	10.0	13.0	206.0
years	9999	38.65	13.75	17.0	28.0	37.0	48.0	90.0
loss	9999	84.11	394.04	0.0	0.0	0.0	0.0	3770.0
prod	9999	2014.93	14007.60	-28.0	42.0	57.0	77.0	200125.0

SalaryPrediction Dataset (Categorical Attributes)

Attribute	Number of Examples (Non-Null)	Number of Unique Values
relation	9999	6
country	9999	41
job	9999	14
work_type	9999	9
partner	9999	7
edu	9999	16
gender	9199	2
race	9999	5
gtype	9999	2
money	9999	2

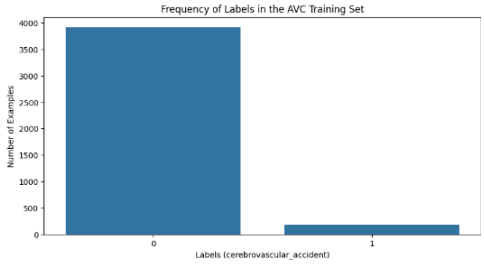
C. Handling Missing Values

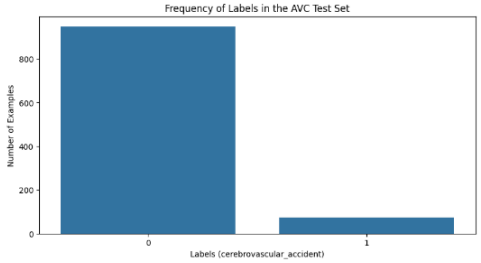
As specified in the requirements, we used the SimpleImputer method from the scikit-learn library.

- For numerical attributes, missing values were replaced with the mean value of the non-missing examples.
- For categorical attributes, missing values were replaced with the most frequent value among the examples.

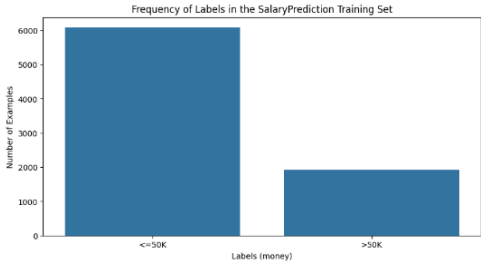
D. Label Frequency

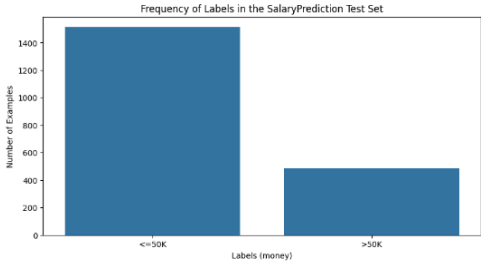
The last column in the datasets represents the labels (the actual values) that our learning algorithms will need to predict. We want to visualize the frequency of these labels (classes) in the dataset.





For the AVC dataset, there is a large number of people who have not suffered from a stroke and a small number who have, both in the training and test sets.



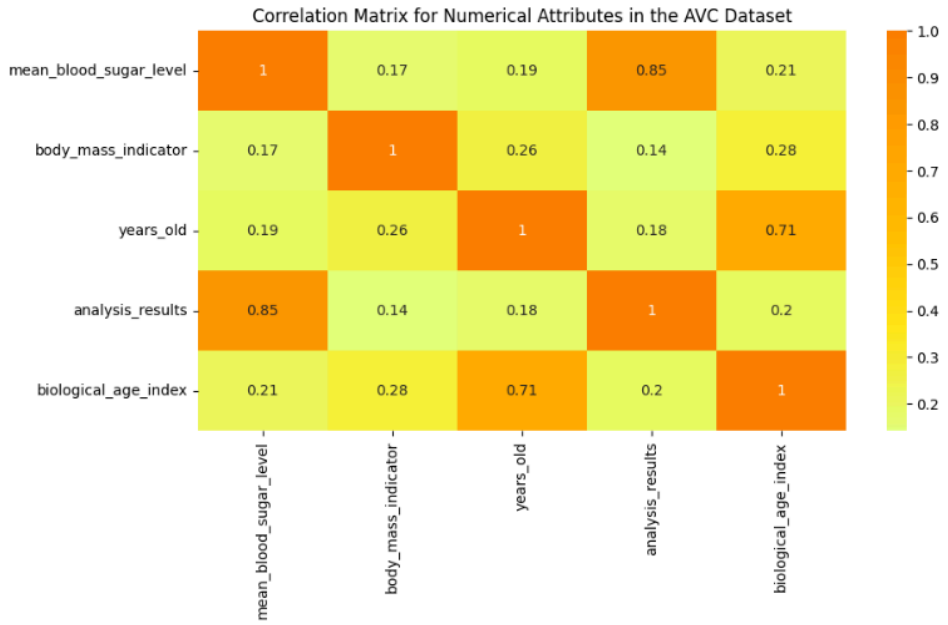


In contrast, the SalaryPrediction dataset has a more balanced class distribution, with the class with fewer examples having approximately one-fourth the number of examples of the other class. This balance reduces the risk of underfitting, unlike the AVC dataset where such issues may arise.

E. Attribute Correlation

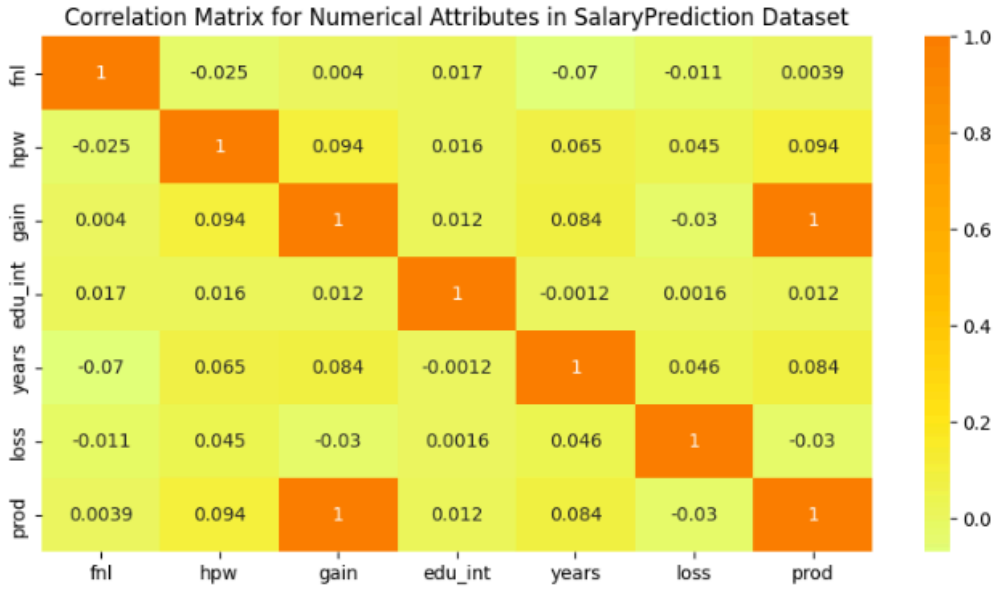
AVC Dataset

Based on the heatmap, we can eliminate the attributes mean_blood_sugar_level, biological_age_index, and cardiovascular_issues due to their strong correlations.



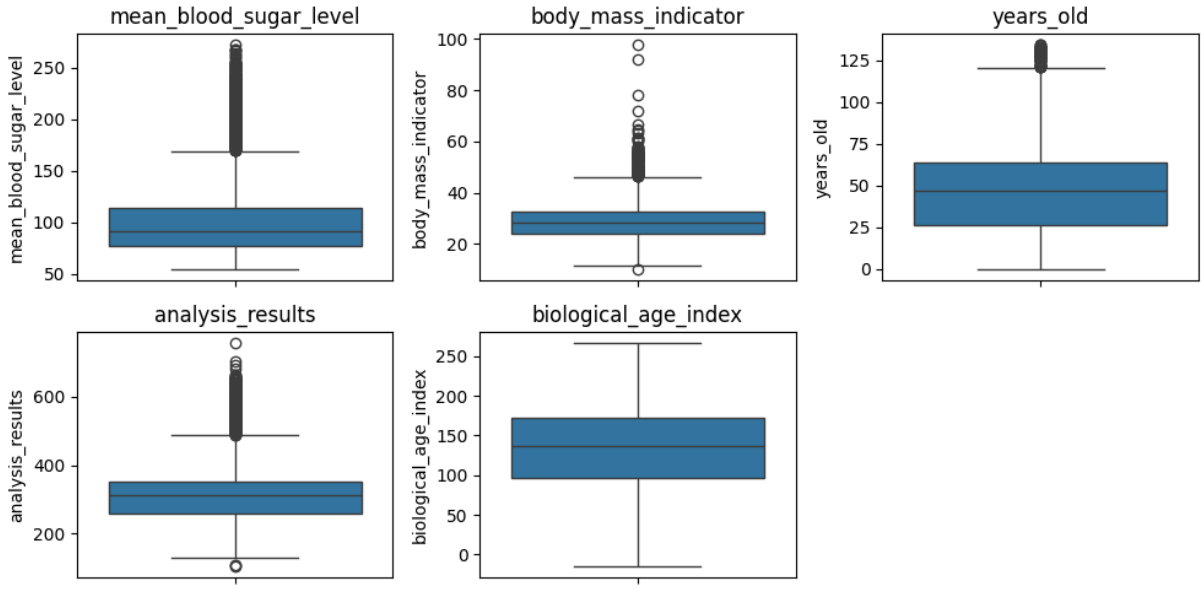
SalaryPrediction Dataset

Based on the heatmap, we can eliminate the attributes prod, relation, and job due to their strong correlations.

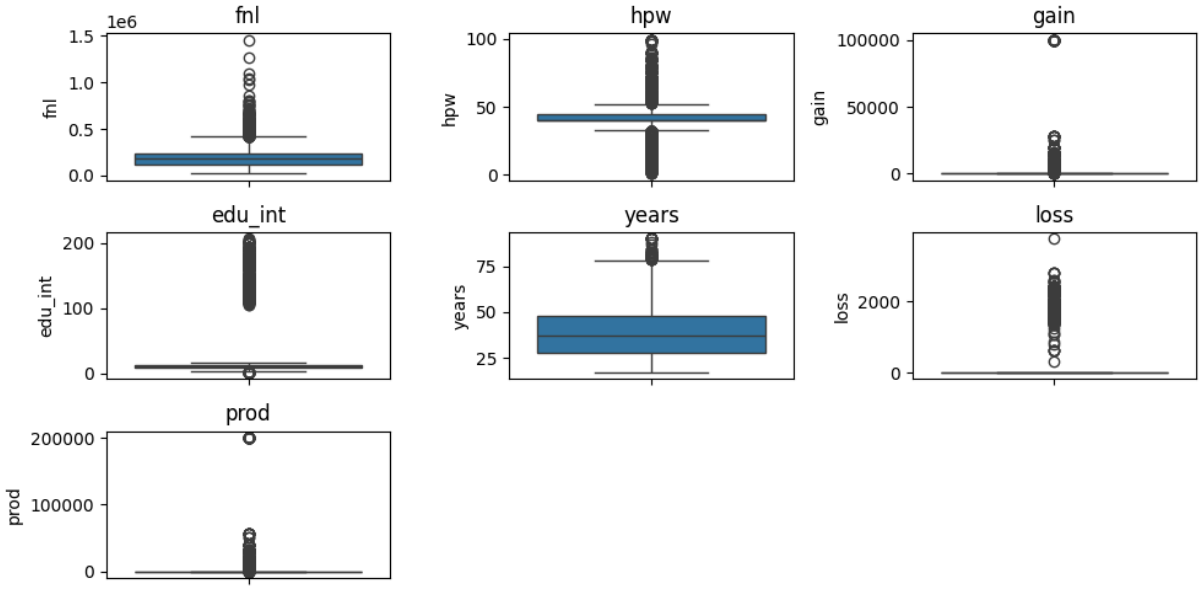


F. Outliers

AVC



SalaryPrediction



G. Elimination of Redundant Attributes / Dataset Scaling

After analyzing and preprocessing the dataset, we eliminated the strongly correlated attributes and used the Standard Scaler from the scikit-learn library to scale and normalize the dataset.

This procedure ensures that there are no examples where attributes have values in the range of tens while others have values in the thousands or higher.

By scaling all attributes to have relatively similar magnitudes, we reduce the computational power required for training and ensure more accurate predictions.

H. Data Encoding

The final step in data preprocessing is encoding the categorical attributes, as the machine learning models in our project can only process numerical values. Therefore, the categorical attributes must be mapped to a numerical representation.

To achieve this, we used the factorize method from the pandas library.

II. Logical Regression

A. Problem Hypothesis

We aim to perform binary classification (predicting only two classes/target variables) based on a dataset containing historical data of patients or employees. The datasets contain variables in the last column known as labels, which in our case are encoded with values 0 or 1.

For the AVC dataset, the labels represent the patients' diagnosis (0 if they have not had a stroke, 1 if they have had a stroke).
For the SalaryPrediction dataset, the labels represent whether an employee earns over 50k per month.
Objective of the Problem:

We aim to use a machine learning model to analyze as many examples of attribute values and labels as possible, and when it receives new, unseen examples, it can provide accurate predictions most of the time.

We will use the training set to train the model and adjust its parameters to obtain accurate predictions. We will use the obtained parameters to measure the success rate of the predictions.

B. Logistic Regression Model

We define the function $z = w \cdot X^{(i)} + b$ from equation $f_{wb}(x) = \text{sigmoid}(z)$ with parameters w and b. Our goal is to minimize these parameters as much as possible.

Sigmoid Function
The sigmoid function centers everything between 0 and 1, and after applying a threshold of 0.5, we can classify any example in the dataset as 0 or 1. Therefore, the model takes the form of equation $f_{wb}(x) = \frac{1}{1+e^{-(w \cdot X^{(i)} + b)}}$

$X^{(i)}$ represents an example from the dataset with all selected attributes after preprocessing the dataset. w represents a vector of length equal to $X^{(i)}$. For n attributes, we will have w=[w1,w2,...,wn]. b represents a scalar.

C. Minimizing Parameters w and b

To minimize the parameters, we introduce the concept of a cost function J(w,b). The objective is to try as many combinations of w and b until J(w,b) reaches a global minimum.
Equation $\text{loss} = -y_i \log(f_{wb}(x)) - (1 - y_i) \log(1 - f_{wb}(x))$ defines how to calculate the cost of a single example from the dataset. For a combination of parameters w and b, we obtain the total cost J(w,b) according to equation $J(w,b) = \frac{1}{N} \sum_{i=0}^{N-1} \text{loss}$.

D. Gradient Descent

To test as many combinations of w and b as possible, with the goal of minimizing them at each iteration, we apply the gradient descent algorithm. At each step, we calculate the partial derivatives of J(w,b) and update the new values of w and b using a learning rate. Partial derivatives represent the direction of orientation on the graph of J(w,b), while the learning rate represents the step length.

$$w = w - lr \frac{\partial J}{\partial w}$$

$$b = b - lr \frac{\partial J}{\partial b}$$

E. Regularization Methods

In the implementation of gradient descent, we used a hyperparameter λ to penalize large values of w. The regularized cost is added to the non-regularized cost to accelerate the convergence process towards the global minimum. Different values of λ are presented in the performance evaluation table.

F. Performance Evaluation

AVC Dataset

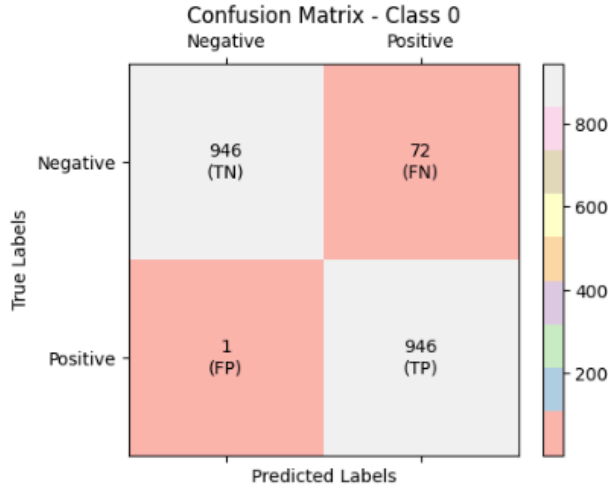
Algorithm	Lambda	Learning Rate	Train Accuracy	Test Accuracy	Precision Class 0	Precision Class 1	Recall Class 0	Recall Class 1	F1 Class 0	F1 Class 1	Epoch Convergence
Logistic Regression	0	0.01	0.955	0.926	0.929	0.5	0.996	0.04	0.961	0.074	9000
Logistic Regression	0.005	0.01	0.955	0.926	0.929	0.5	0.996	0.04	0.961	0.074	9000
Logistic Regression	0.005	0.05	0.955	0.926	0.929	0.5	0.996	0.04	0.961	0.074	4500
Logistic Regression (scikit-learn)	0	-	0.957	0.926	0.93	0.0	1.0	0.0	0.96	0.0	-

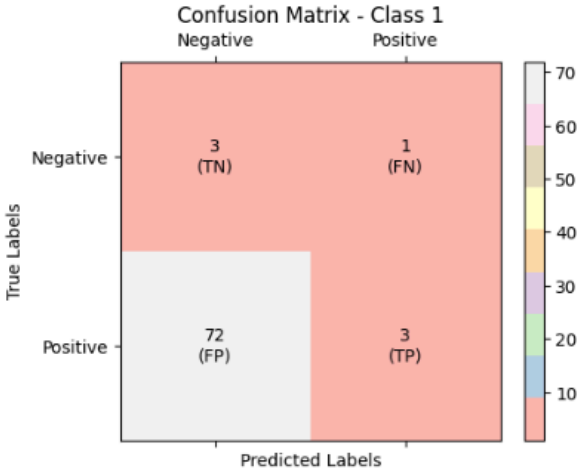
SalaryPrediction Dataset

Algorithm	Lambda	Learning Rate	Train Accuracy	Test Accuracy	Precision Class 0	Precision Class 1	Recall Class 0	Recall Class 1	F1 Class 0	F1 Class 1	Epoch Convergence
Logistic Regression	0	0.01	0.796	0.801	0.827	0.648	0.93	0.398	0.876	0.493	7000
Logistic Regression	0.005	0.01	0.796	0.801	0.827	0.648	0.93	0.398	0.876	0.493	7000
Logistic Regression	0.005	0.05	0.797	0.799	0.825	0.645	0.931	0.388	0.875	0.484	2100
Logistic Regression (scikit-learn)	0	-	0.724	0.802	0.83	0.66	0.93	0.39	0.88	0.49	-

G. Confusion Matrices

AVC





SalaryPrediction