Pana Adrian 333CC

Inteligență Artificială: Tema 2 - ML aplicat

Structura proiect

Proiectul contine mai multe fisiere python:

- tema2.py - fisierul principal, “hub-ul” care este rulat
- analysis.py - pentru toata partea de analize si afisare grafice/plot-uri a dataset-urilor
- preprocess.py - pentru preprocesarea dataset-urilor in urma analizei
- random_forest.py/mlp.py - contin implementarile celor 4 algoritmi
- utils.py - pentru utilizarea mai usoara a seturilor de date, split-uri (folosite de random forest) si extragerea de metrici

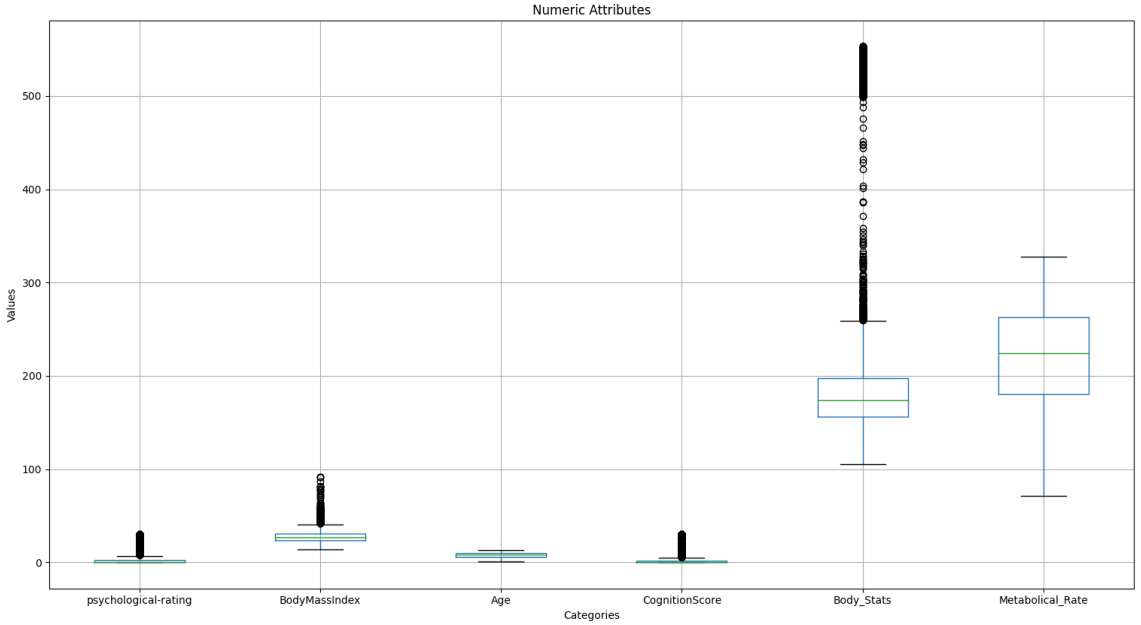
Explorarea Datelor

- Am folosit metoda de vizualizare a gradului de corelare a atributelor numerice si pentru cele categoriale, celulele matricii sunt colorate in doar 2 culori in loc de un gradient.
- In seturile de date exista multe campuri NaN si outliere
- Clasele sunt foarte dezechilibrate pentru ambele seturi de date

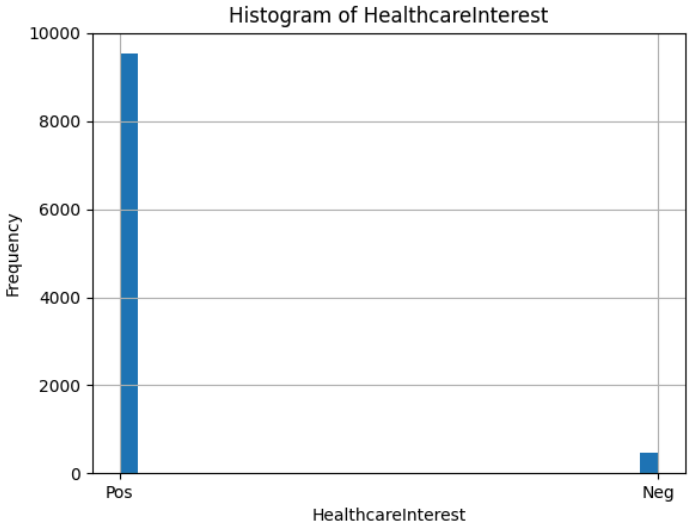
Diabet

1. Analiza tipului de atribute și a plajei de valori a acestora

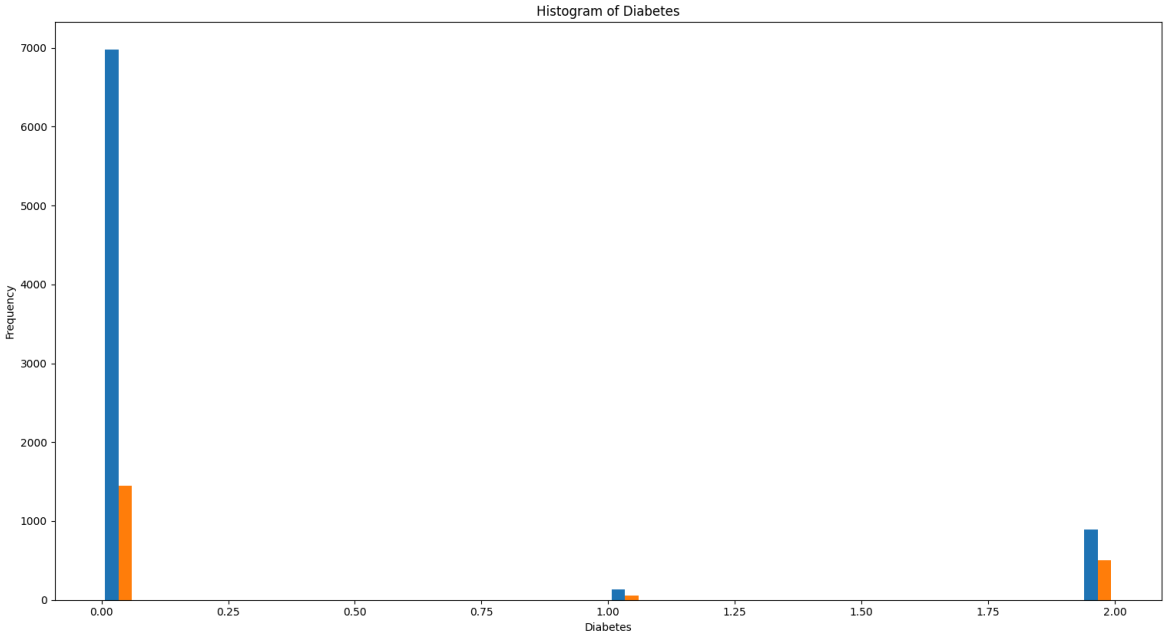
	psychological-rating	BodyMassIndex	Age	CognitionScore	Body_Stats	Metabolical_Rate
count	10000.000000	10000.000000	10000.0000	10000.000000	10000.000000	9000.000000
mean	4.365100	28.246500	8.0575	3.125300	194.960784	221.592499
std	8.891103	6.462563	3.0363	7.308607	82.438106	60.480951
min	0.000000	14.000000	1.0000	0.000000	105.063984	71.602207
25%	0.000000	24.000000	6.0000	0.000000	156.720671	180.542314
50%	0.000000	27.000000	8.0000	0.000000	174.042100	224.218817
75%	3.000000	31.000000	10.0000	2.000000	197.742249	262.688901
max	30.000000	92.000000	13.0000	30.000000	553.000000	327.936098



Exemplu histograma atribut categorial:

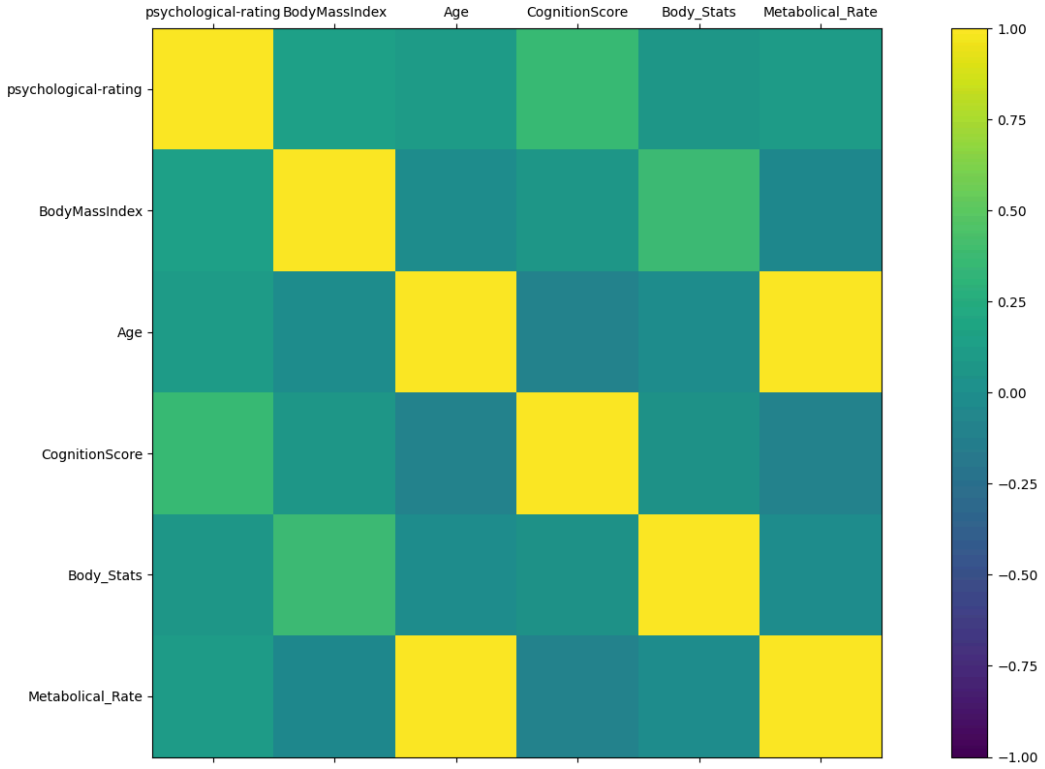


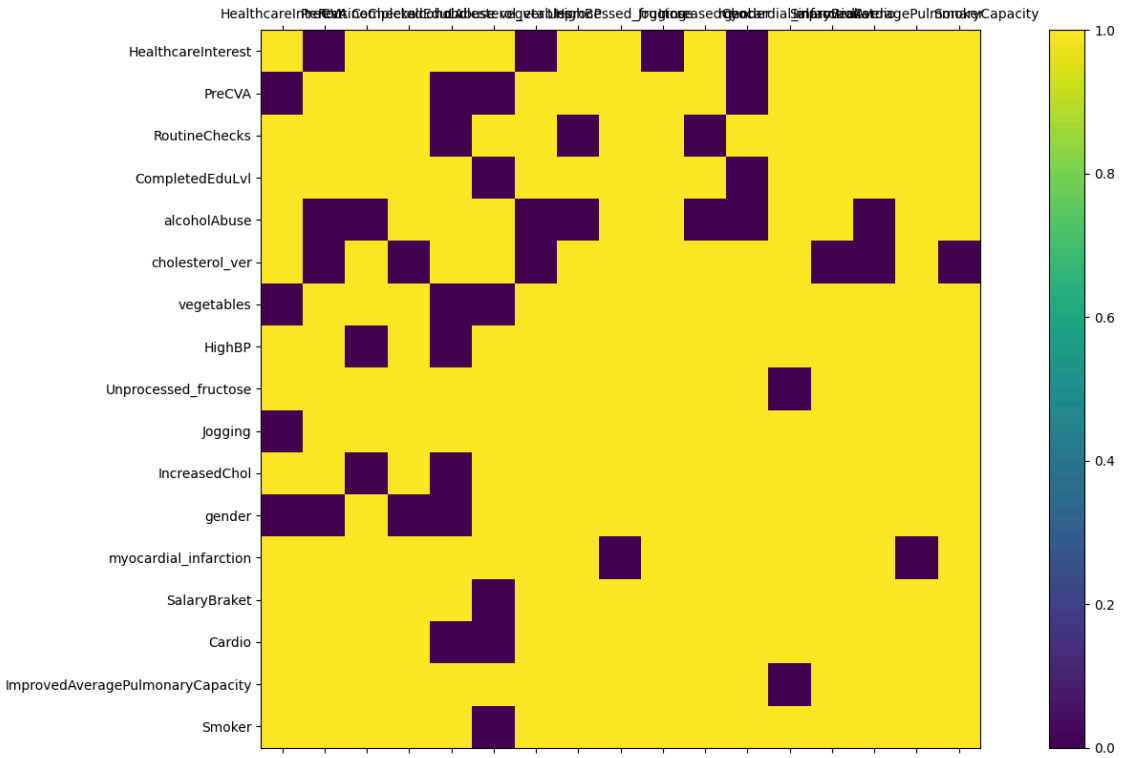
2. Analiza echilibrului de clase



Clasa 0 are o proportie substantial mai mare decat celelalte, clasele suport 1 si 2 vor fi clasificate eronat cu o probabilitate mare.

3. Analiza corelației între atribute




HealthcareInterest
PreCVA
RoutineChecks
CompletedEduLvl
alcoholAbuse
cholesterol_ver
vegetables
HighBP
Unprocessed_fructose
Jogging
IncreasedChol
gender
myocardial_infarction
SalaryBraket
Cardio
ImprovedAveragePulmonaryCapacity
Smoker
1.0
0.8
0.6
0.4
0.2
0.0

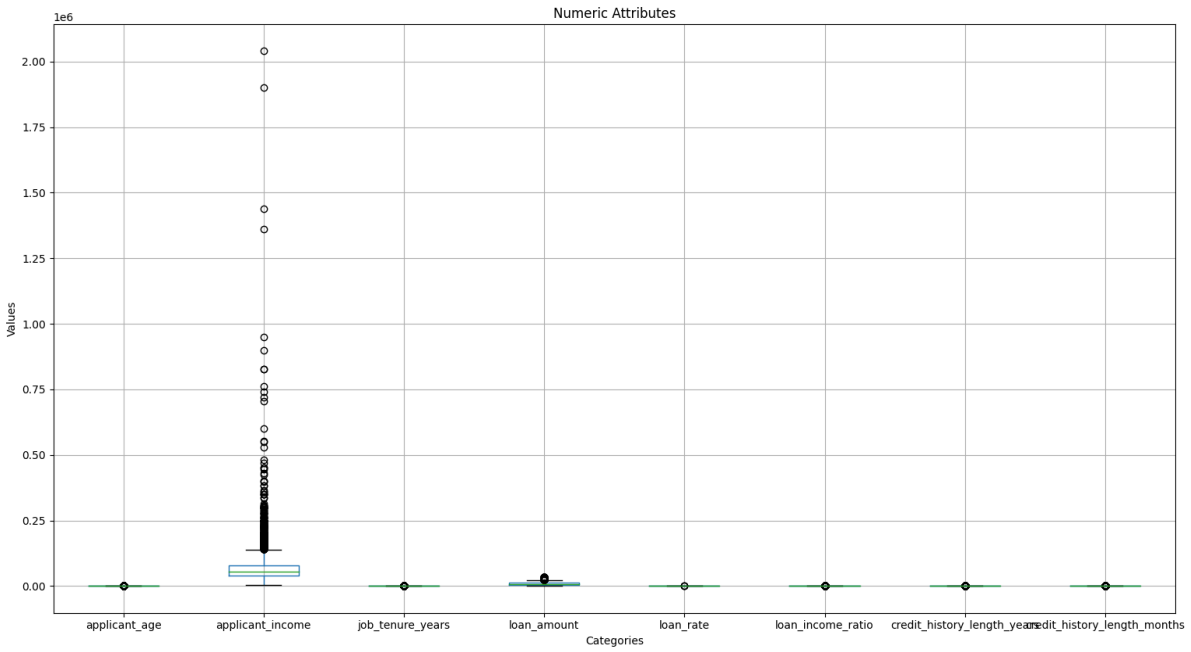
Risc de Creditare

1. Analiza tipului de atribute și a plajei de valori a acestora

	applicant_age	applicant_income	job_tenure_years	loan_amount
count	10000.000000	1.000000e+04	9736.000000	10000.000000
mean	27.745100	6.573421e+04	4.785744	9568.037500
std	6.360155	5.694439e+04	4.353122	6350.431581
min	20.000000	4.200000e+03	0.000000	500.000000
25%	23.000000	3.859500e+04	2.000000	5000.000000
50%	26.000000	5.500000e+04	4.000000	8000.000000
75%	30.000000	7.899700e+04	7.000000	12200.000000
max	123.000000	2.039784e+06	123.000000	35000.000000

	loan_rate	loan_income_ratio	credit_history_length_years
count	9060.000000	10000.000000	10000.000000
mean	11.007179	0.170130	5.811100
std	3.266393	0.106814	4.050217
min	5.420000	0.000000	2.000000
25%	7.900000	0.090000	3.000000
50%	10.990000	0.150000	4.000000
75%	13.470000	0.230000	8.000000
max	23.220000	0.760000	30.000000

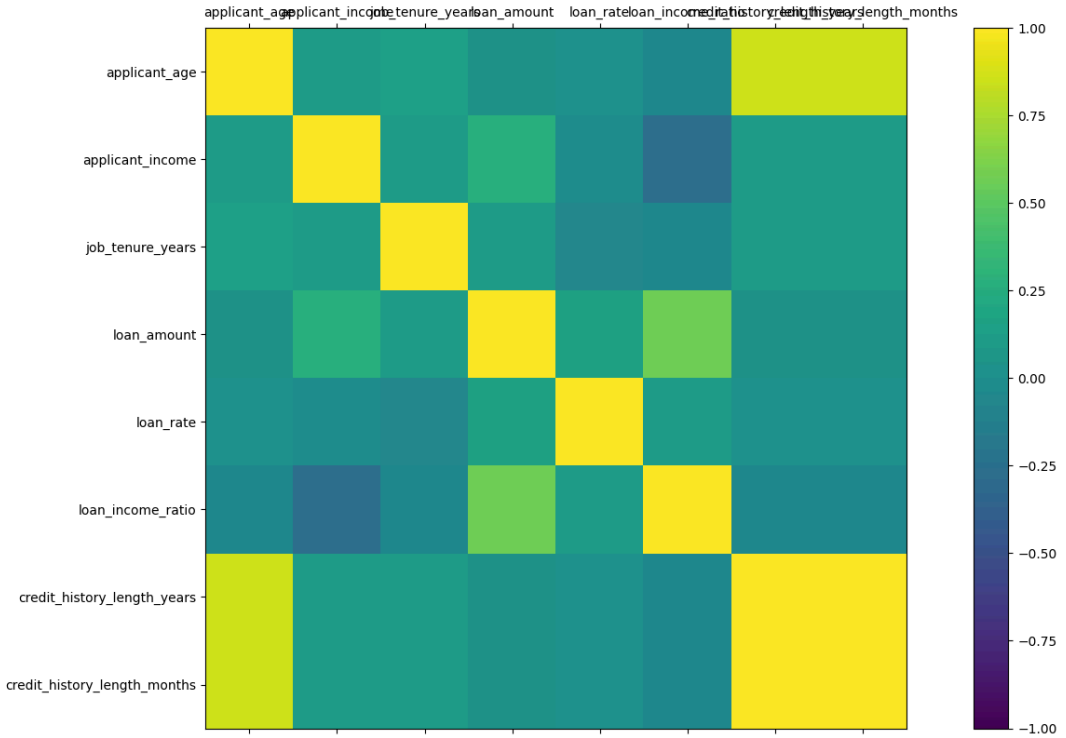
	credit_history_length_months
count	10000.000000
mean	75.760700
std	48.677362
min	25.000000
25%	41.000000
50%	57.000000
75%	102.000000
max	369.000000

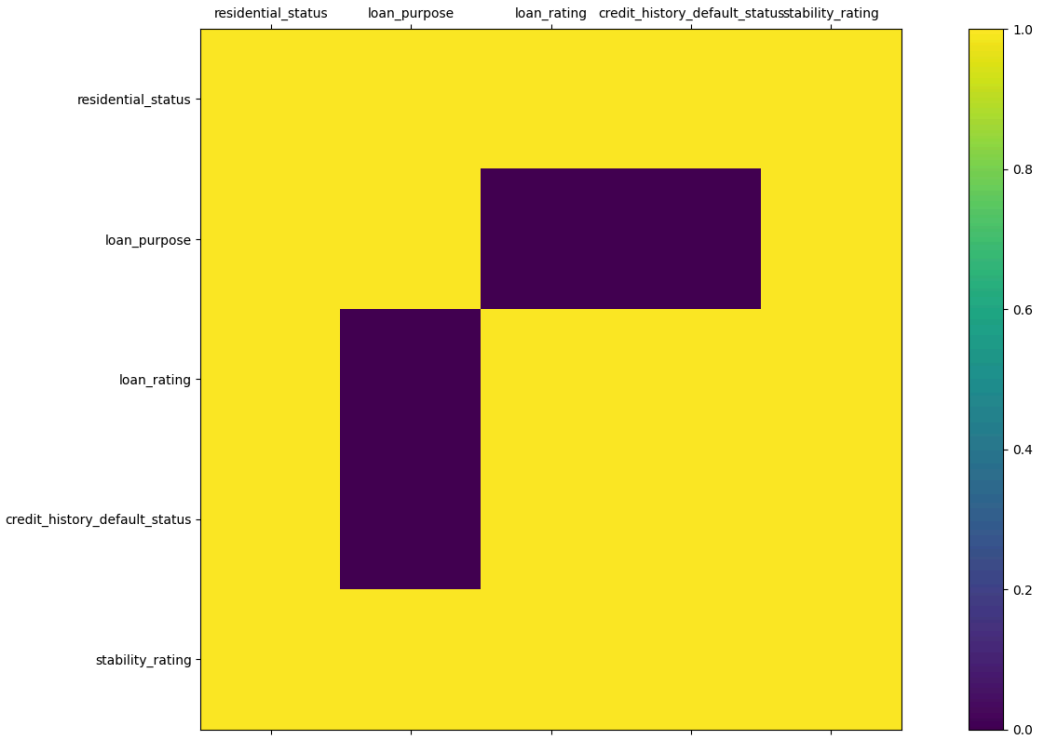


2. Analiza echilibrului de clase

Clasa Approved are o proportie mult mai mare in exemple, clasa suport ‘Declined’ va fi clasificata eronat cu o probabilitate mai mare.

3. Analiza corelației între atribute




residential_status
loan_purpose
loan_rating
credit_history_default_status
stability_rating
1.0
0.8
0.6
0.4
0.2
0.0

Preprocesare

Strategii folosite

Outlier removal (cu numar neglijabil de outliere ramase):

- quantile Q1 = 0.25, Q3 = 0.75
- threshold = 1

Missing values fill:

- numerice - mean value
- categoriale - most frequent value

Standardization:

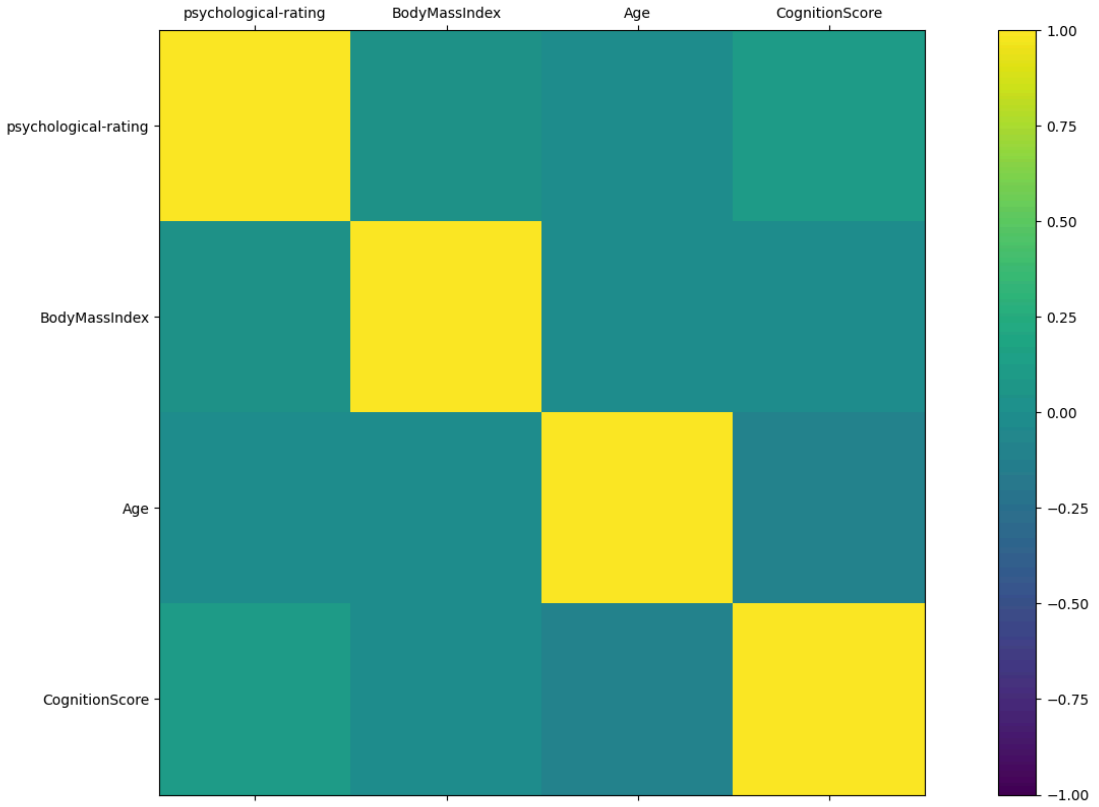
- Standard Scaler produce cele mai bune rezultate

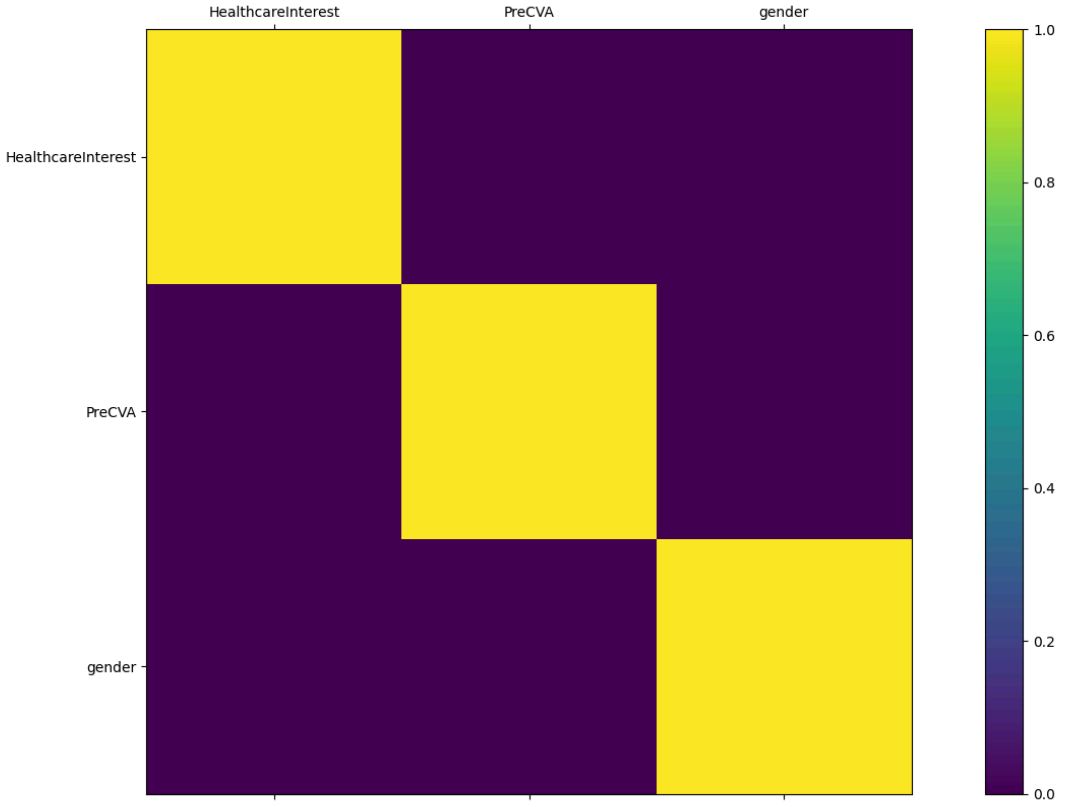
In urma plot-urilor facute, vom renunta la mai multe atribute.

Diabet

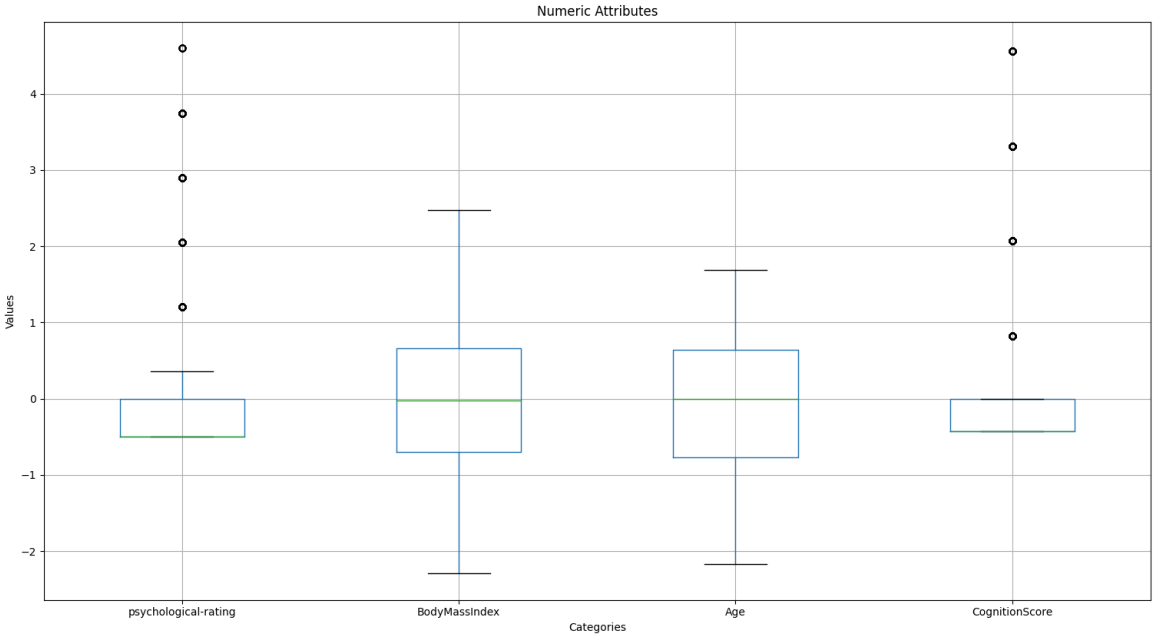
Pastram urmatoarele:

- Numerice: "psychological-rating", "BodyMassIndex", "Age", "CognitionScore"
- Categoriale: “HealthcareInterest”, “PreCVA”, “gender”





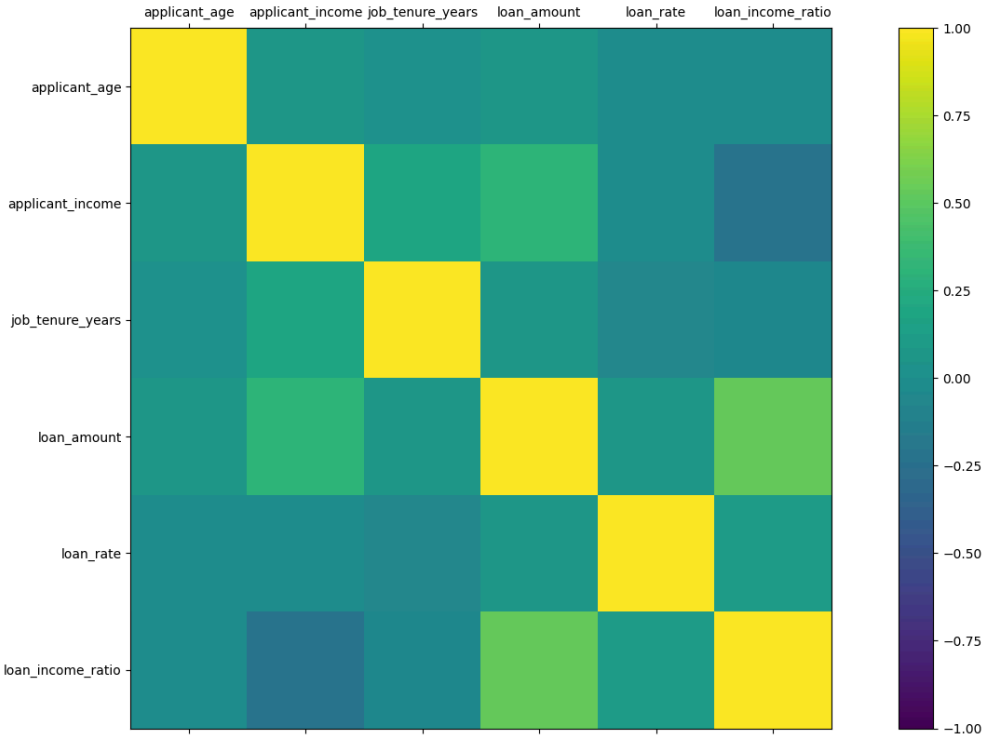
In urma eliminarii valorilor extreme ale atributelor, inlocuirea tuturor atributelor lipsa si standardizarea acestora, atributele numerice ajung sub urmatoarea forma:

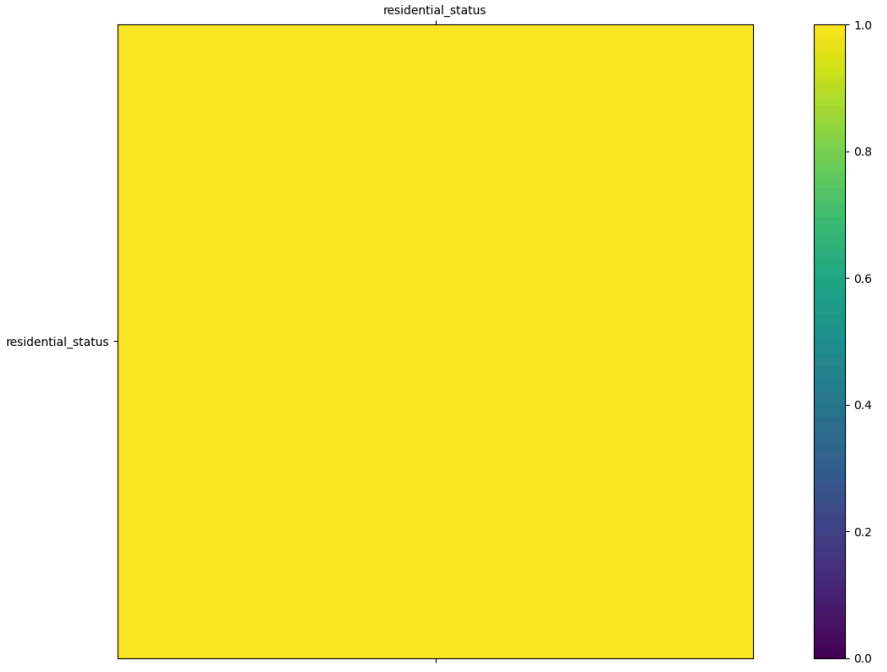


Risc de Creditare

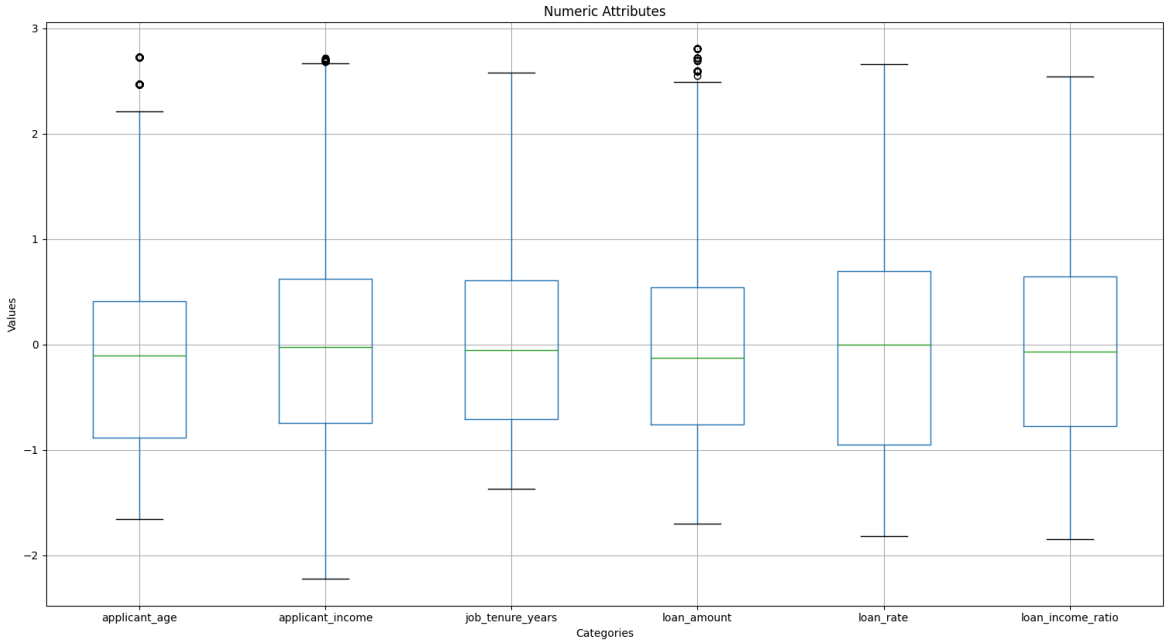
Pastram urmatoarele:

- Numerice: "applicant_age", "applicant_income", "job_tenure_years", "loan_amount", "loan_rate", "loan_income_ratio"
- Categoriale: “residential_status”





In urma eliminarii valorilor extreme ale atributelor, inlocuirea tuturor atributelor lipsa si standardizarea acestora, atributele numerice ajung sub urmatoarea forma:



Implementare algoritmi

Functia ‘prepare_data()’ este folosita de toti algoritmii si este folosita pentru a converti atributele categoriale in forma numerica.
De asemenea, aceasta converteste clasele setului de date Credit_Risk de la “Approved” si “Declined” in valori numerice: 1 si 0.

1. Scikit Random Forest

Hiperparametri:

- max_depth - infinit, max_samples = infinit -> limitarea ii creste bias-ul spre prima clasa in ambele dataset-uri
- min_samples_leaf = 1 -> creste complexitatea dar asigura decizii mai bine calculate
- n_estimators -> valorile din intervalul [5,10] aduc rezultate bune, am ales **8** pentru ca este unul dintre cele mai bune
- class_weight -> am incercat sa adaug weight-uri pentru a combate proportia iesita din comun a primei categorii, dar nu a adus niciun plus rezulatului
- criterion = ‘gini’, max_features = ‘log2’ -> rezultatele cele mai bune in urma trial-and-error

2. Lab Random Forest

Codul din laborator cu urmatoarele modificari:

- Colorare frunze arbori in culori diferite pentru o mai buna vizualizare
- Grafice vizualizate cu matplotlib (incarcarea ia timp, asa ca am comentat linia respectiva, poate fi decomentata pentru analiza)
- Atributele numerice (care contin multe valori unice, am ales ca threshold 7 valori) sunt impartite in 2 in functie de valoarea care minimizeaza ‘gini_index’; pentru aceste noduri, atributul se pastreaza si poate lua parte la o noua impartire mai adanc in arbore

Hiperparametri:

- n_estimators = 5 -> cresterea lor duce la un procent mai mare de arbori care favorizeaza clasa in proportie mai mare din dataset
- max_depth = 8 -> ajuta si la vizualizarea arborilor
- min_samples_per_node =50 -> forteaza mai multe split-uri care pot duce la clasele cu proportii mici
- split_strategy = ‘random’ , subset_size_ratio = 0.5, subset_feature_ratio = 1 -> rezultatele cele mai bune in urma trial-and-error

3. Scikit MLP

Hiperparametri:

- hidden_layers = 2, hidden_layer_sizes = [100,100] -> mai multe layere/noduri per layer duc la un bias mai mare asupra clasei cu proportie majoritara
- activation = ‘relu’ -> preferat pentru clasificare
- learning_rate = ‘invscaling’ -> produce rezultate asemanatoare cu ‘constant’, am ales-o pe aceasta pentru a evita “salturi in afara caii” atunci cand ne apropiem de clasele cu proportii mai mici
- solver = ‘lbfgs’, alpha = 0.001, max_iter = 300 -> rezultatele cele mai bune in urma trial-and-error

4. Lab MLP

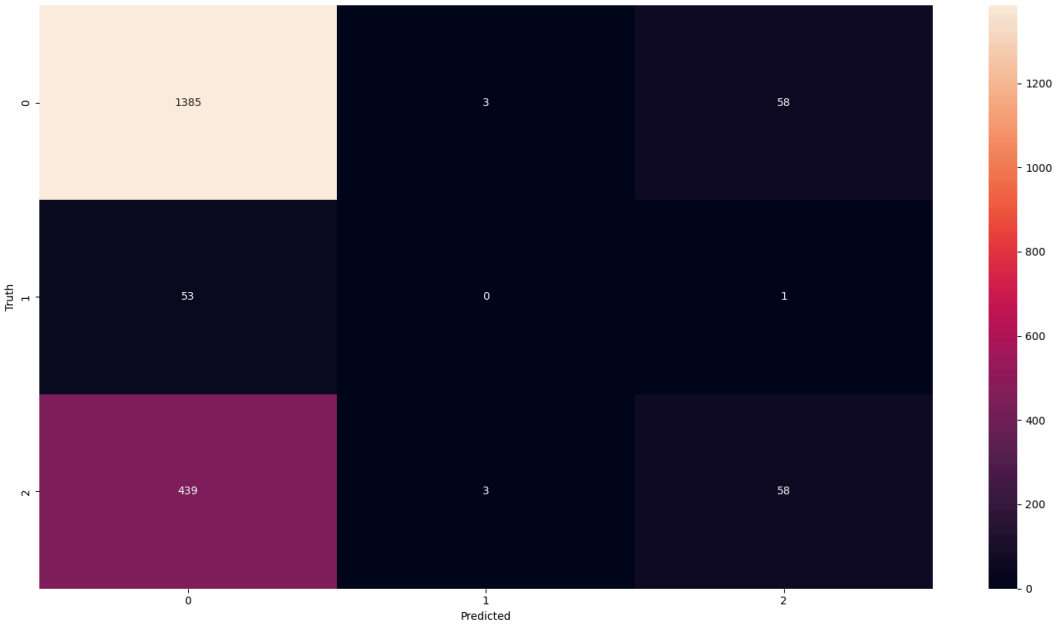
Hiperparametri:

- In urma testarii mai multor arhitecturi ale retelei, am ajuns la cea cu 2 layere Liniar ascunse cu 100 de noduri fiecare, urmate de un ReLU ascuns care e legat la output.
- lr = 0.05 -> o valoare ridicata, forteaza reteaua sa diferentieze mai repede intre clase
- EPOCHS_NO = 10 -> un numar mai mare devine redundant

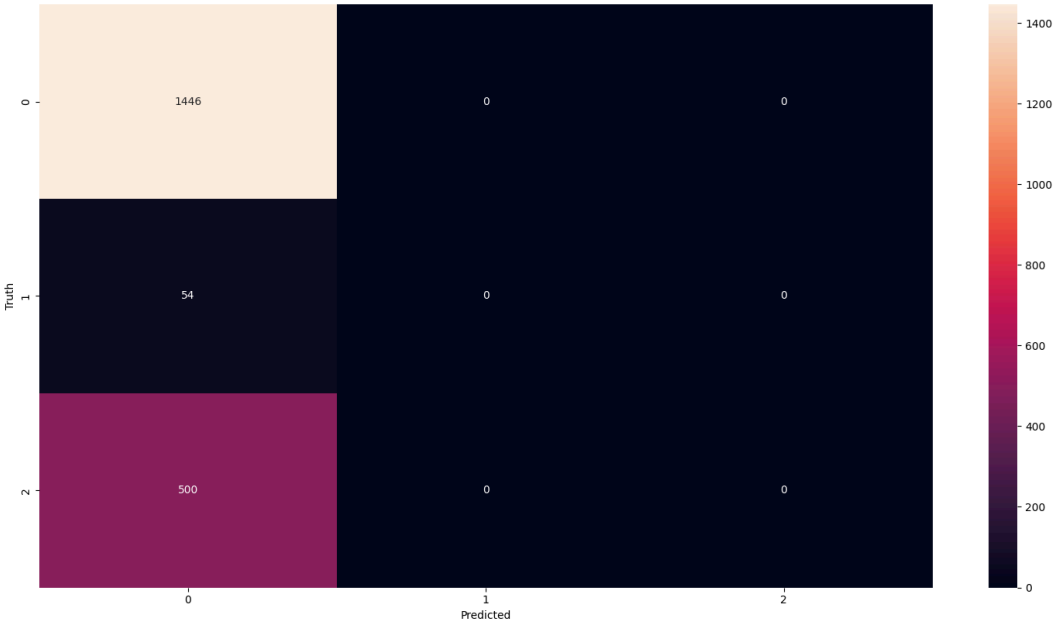
Evaluare algoritmi

Diabetes

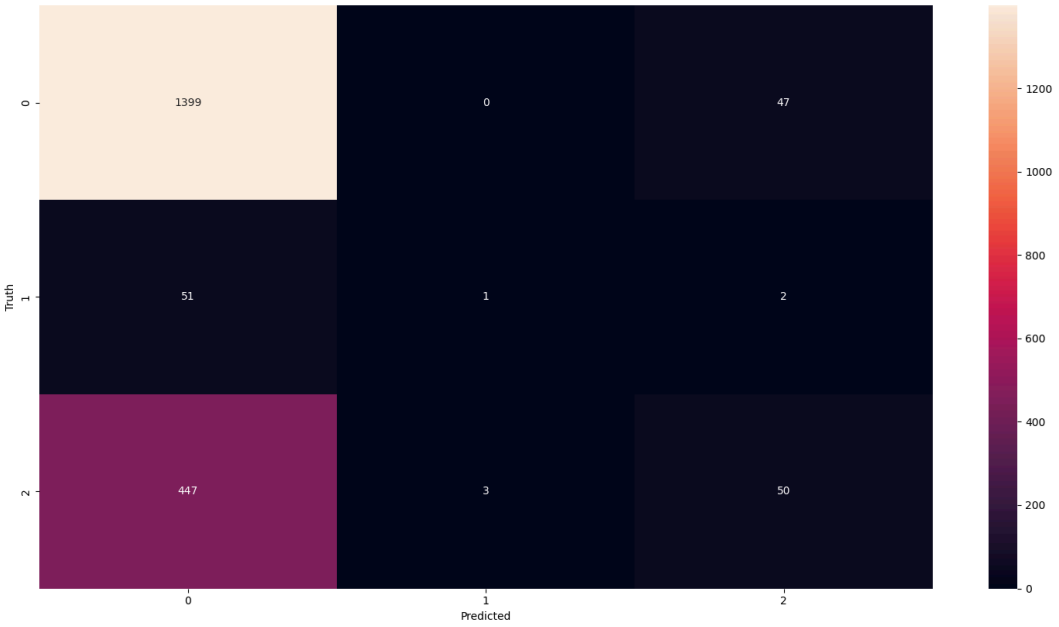
1. Scikit Random Forest



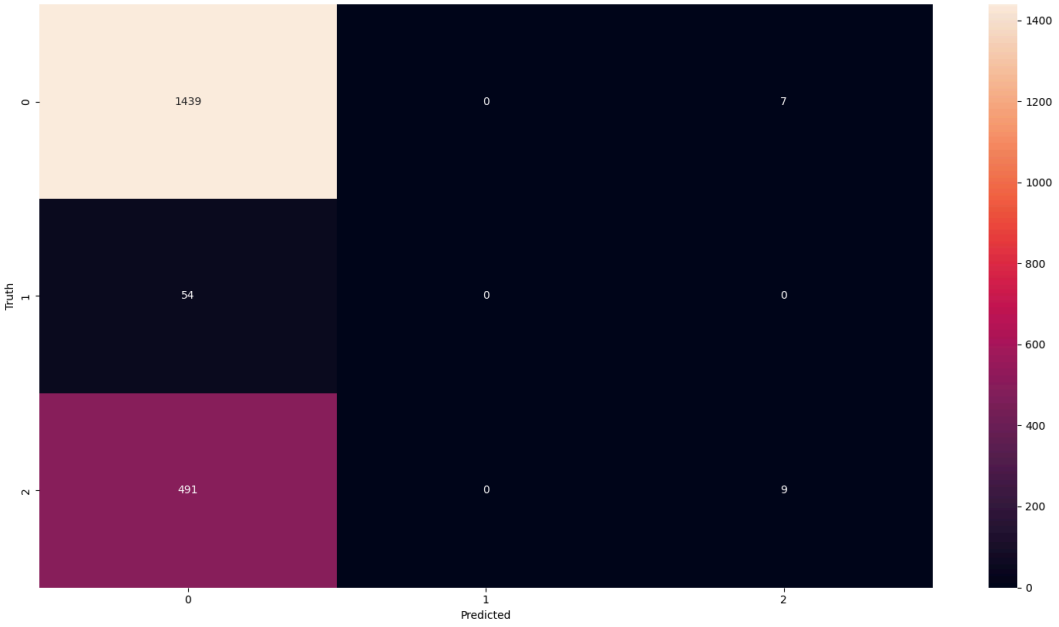
2. Lab Random Forest



3. Scikit MLP

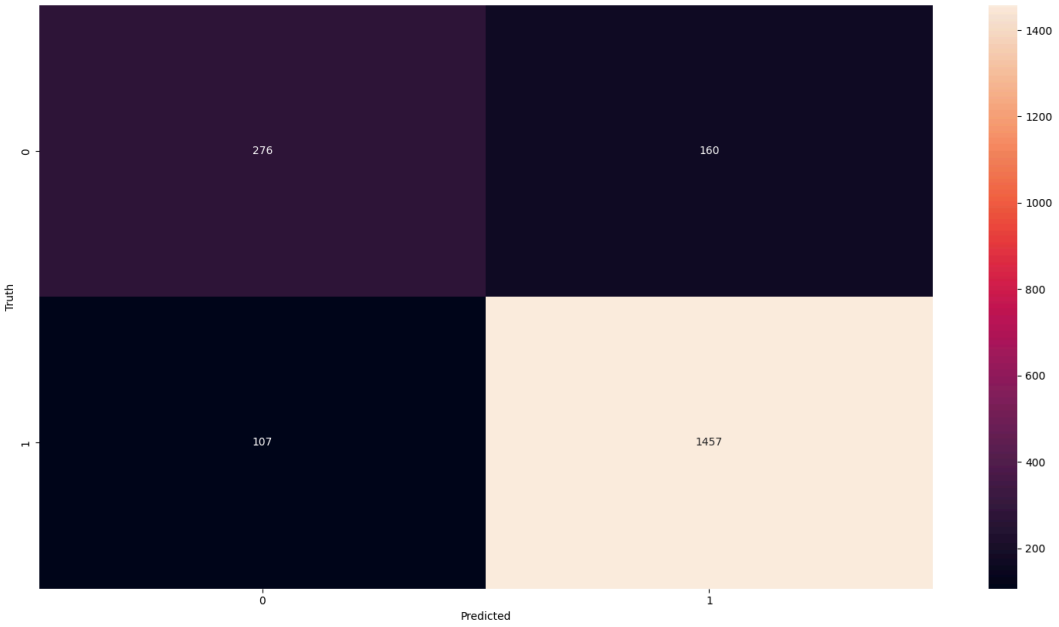


4. Lab MLP

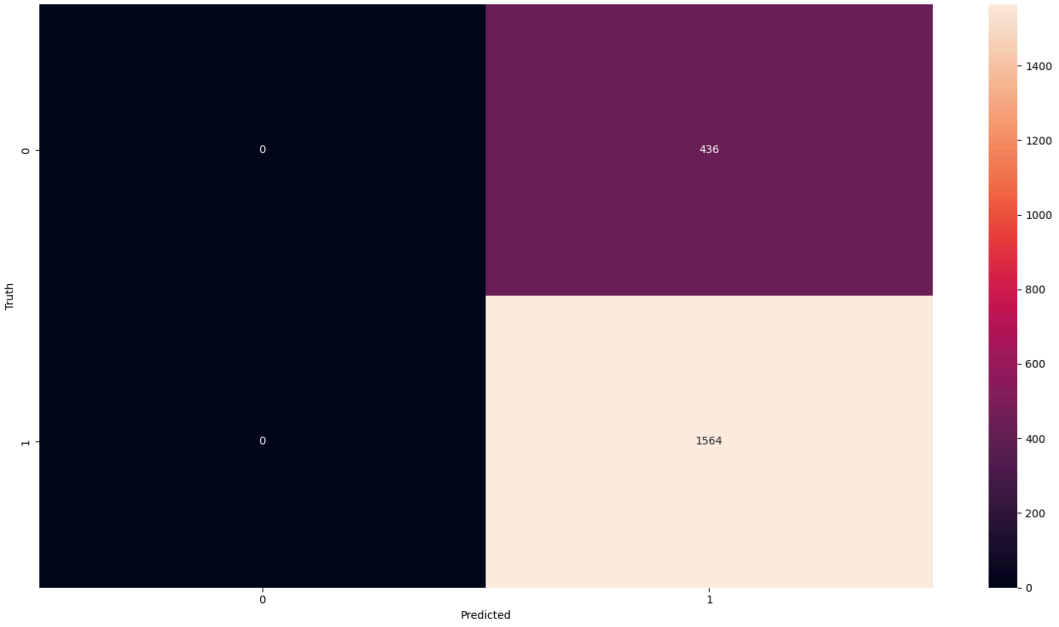


Credit Risk

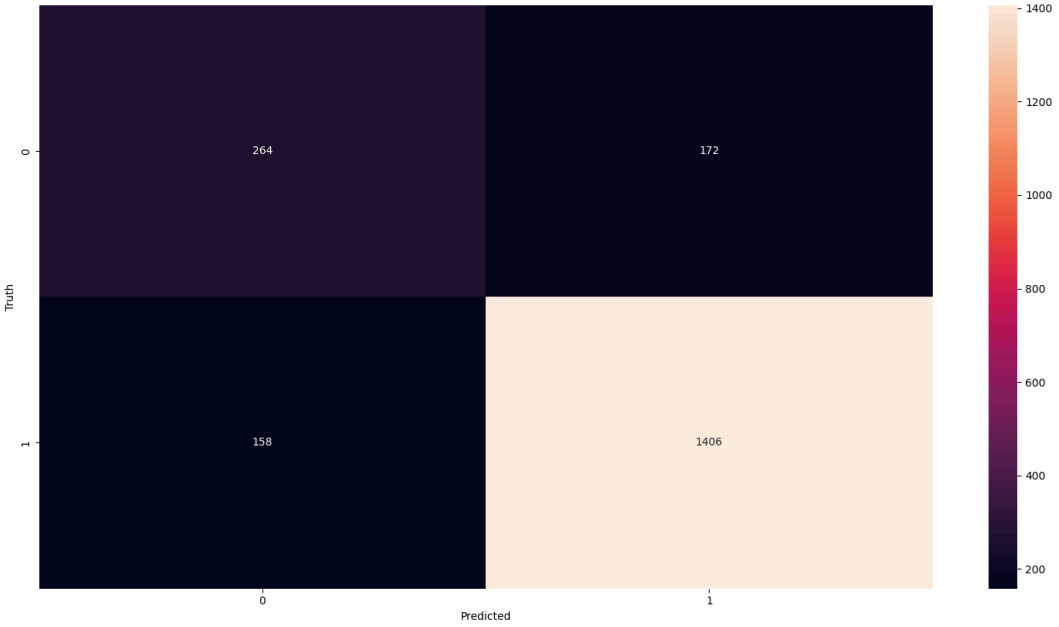
1. Scikit Random Forest



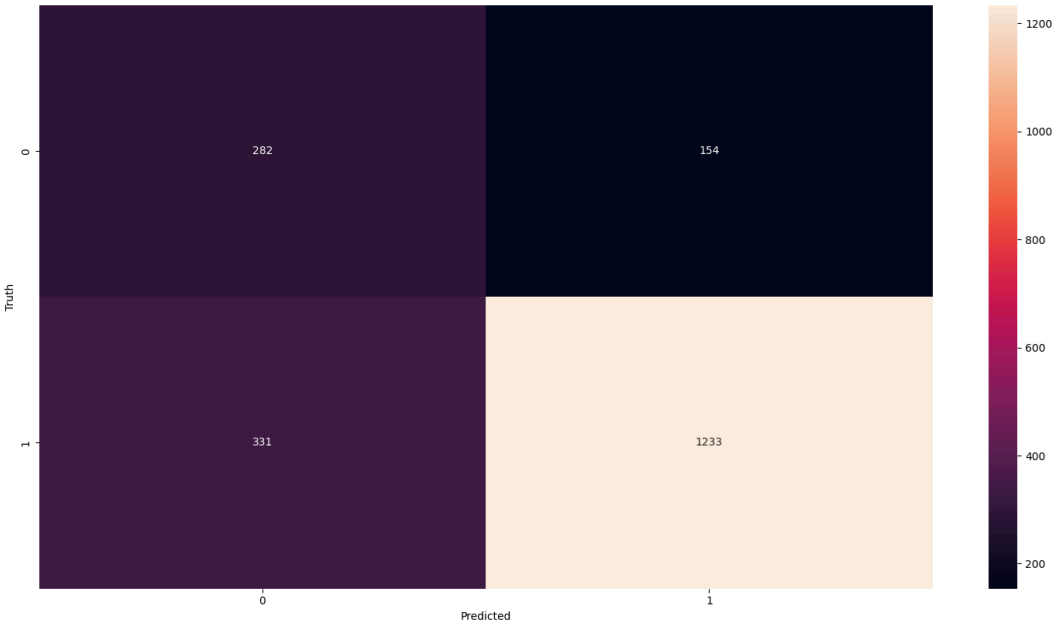
2. Lab Random Forest



3. Scikit MLP



4. Lab MLP



Diabetes

	algo	accuracy	Precision_0.0	Precision_1.0	Precision_2.0	Recall_0.0	Recall_1.0	Recall_2.0	F1_0.0	F1_1.0	F1_2.0
0	Scikit Random Forest	0.7155	0.736588	0.00	0.453125	0.949516	0.000000	0.116	0.829607	0.000000	0.184713
1	Lab Random Forest	0.7230	0.723000	0.00	0.000000	1.000000	0.000000	0.000	0.839234	0.000000	0.000000
2	Scikit MLP	0.7250	0.737480	0.25	0.505051	0.967497	0.018519	0.100	0.836973	0.034483	0.166945
3	Lab MLP	0.7235	0.724259	0.00	0.555556	0.997234	0.000000	0.010	0.839104	0.000000	0.019646

Scikit MLP produce cele mai bune rezultate, in afara F1, unde pentru clasele 0 si 2 valorile maxime sunt detinute de Scikit Random Forest, care ofera in general rezultate apropiate.

Credit Risk

	algo	accuracy	Precision_1	Precision_0	Recall_1	Recall_0	F1_1	F1_0
0	Scikit Random Forest	0.8630	0.900621	0.707692	0.927110	0.633028	0.913674	0.668281
1	Lab Random Forest	0.7820	0.782000	0.000000	1.000000	0.000000	0.877666	0.000000
2	Scikit MLP	0.8350	0.891001	0.625592	0.898977	0.605505	0.894971	0.615385
3	Lab MLP	0.7575	0.885633	0.459235	0.792199	0.633028	0.836315	0.532305

Scikit Random Forest produce cele mai bune rezultate in toate metricele, cu Scikit MLP avand rezultate apropiate.

Concluzie

In urma analizei rezultatelor, cel mai bun algoritm pare este Scikit MLP, prezentand cel mai mic bias spre clasa cu intrari in proportia majoritara.

Motive principal pentru care cred ca produce cele mai bune rezultate:

- Arhitectura: ReLU este favorizata pentru clasificari multi-clasa, si algoritmul permite layere ascunse ReLU cu multe noduri, fapt ce creste complexitatea si probabilitatea de a ajunge la un rezultat bun