Nume: Bogdan-Andrei Sprincenatu

Grupa: 332CC

Explorarea datelor

1. Am verificat continutul celor 2 seturi de date principale pentru a vedea in ce intervale de valori se afla atributele numerice, dupa care am facut boxplot-uri pentru fiecare in parte.
2. Am continuat prin determinarea numarului de valori unice si reprezentarea de histograme pentru atributele categorice.
3. Am reprezentat cu ajutorul functiei **countplot** din **seaborn** clasele pentru toate tabelele corespunzatoare fiecarui set de date.
4. Am reprezentat matricea de corelatie pentru atributele numerice si cele categorice.

Preprocesarea datelor

1. Pentru atributele numerice ce aveau valori lipsa(**Metabolical_Rate**) am folosit **IterativeImputer,** iar pentru cele categorice(**CompletedEduLvl**) am folosit **SimpleImputer** cu strategia **most_frequent.**
2. Eliminarea valorilor numerice extreme am realizat-o extragand pentru fiecare atribut din tabelele avute la dispozitie doar acele valori aflate in intervalul (Q1 – tolerance * IQC, Q3 + tolerance * IQC), unde **IQC** reprezinta diferenta interquartile(Q3 – Q1), iar toleranta a fost aleasa de mine ca fiind 1.5(asa am vazut si in anexa ca era), urmand sa imputez valori noi pentru fiecare atribut numeric utilizand functiile definite anterior.
3. Am eliminat atributele numerice **Metabolical_Rate(**deoarece acesta era puternic corelat cu **Age)** si **Body_Stats(**acesta fiind puternic corelat cu **BodyMassIndex)** pentru setul cu **diabet**, iar la cel cu **credit_risk** am eliminat **credit_history_length_months** si **credit_history_length_years.** Pe cele categorice le-am eliminat in acelasi mod, dar in functie de media pe care o avea valoarea **p** obtinuta in urma aplicarii functiei **chi2_contingency** pe 2 cate 2 atribute.
4. La finalul acestei etape am standardizat datele utilizand **StandardScaler.**

Hiperparametrii

1. Random Forrest
    a. Diabet
        i. Varianta Sklearn:

n_estimators	min_samples_split	min_samples_leaf	max_features	max_depth
133	5	2	log2	110

        ii. Varianta implementata manual:

n_estimators	max_depth	min_samples_per_node	split_strategy
250	150	2	id3

b. Risc Credit
   i. Varianta Sklearn: aceleasi rezultate ca si la **Diabet**.
   ii. Varianta implementata manual: aceeasi hiperparametrii ca si la **Diabet**.

2. MLP
   a. Diabet
      i. Varianta Sklearn:

solver	max_iter	learning_rate	Hidden_layer_sizes	Alpha	activation
adam	500	Adaptive	(100,)	0.001	relu

      ii. Varianta implementata manual: niciunul.
   b. Risc Credit
      i. Varianta Sklearn: Aceeasi parametrii ca si la Diabet.
      ii. Varianta implementata manual: niciunul.
3. Observatii
   a. Pentru variantele din sklearn am implementat 2 functii pe care le-am utilizet pentru hyperparameter tunning. Astfel, am obtinut parametrii optimi pentru care se obtinea un scor maxim **F1**.

Evaluarea Algoritmilor

Random Forrest							
Diabet				Risc Credit			
Sklearn		Implementare		Sklearn		Implementare	
Acuratete	F1	Acuratete	F1	Acuratete	F1	Acuratete	F1
0.6175	0.646	0.7235	0.679	0.8465	0.846	0.764	0.705

MLP							
Diabet				Risc Credit			
Sklearn		Implementare		Sklearn		Implementare	
Acuratete	F1	Acuratete	F1	Acuratete	F1	Acuratete	F1
0.7205	0.665	0.7205	0.665	0.8485	0.833	0.85	0.829

In urma evaluarii algoritmilor am obtinut pe setul de date **Diabet** rezultate mai putin satisfacatoare in cazul ambilor algoritmi, o performanta putin mai buna avand **MLP**, lucru ce se intampla din cauza dezechilibrului claselor puse la dispozitie(sunt foarte multe persoane care nu au diabet,

foarte putine care au prediabet si putin mai multe cele care au diabet), iar pe setul de date **Credit Risk** am obtinut rezultate mult mai satisfacatoare, o performanta mai buna avand-o tot **MLP**. Aceste performante sunt vizibile si cu ajutorul matricelor de confuzie de la fiecare implementare in parte(la setul de date **Credit Risk** predictiile obtinute sunt mult mai apropiate de cele reale decat sunt la setul de date **Diabet**).