

Laborator 6

Versiunea 2020-04-01

Modele de clasificare

Folositi 4 seturi de date pentru probleme de clasificare, plecand de la repository-urile specificate in Cursul 5; de exemplu, [ics.uci.edu \(http://archive.ics.uci.edu/ml/datasets.php?format=mat&task=cla&att=&area=&numAtt=&numIns=&type=mvar&sort=nameUp&view=table\)](http://archive.ics.uci.edu/ml/datasets.php?format=mat&task=cla&att=&area=&numAtt=&numIns=&type=mvar&sort=nameUp&view=table). Cel puțin două seturi de date să fie cu valori lipsă.

1. (20 puncte) Aplicați o metoda de missing value imputation, unde este cazul; justificați și documentați metoda folosită.
2. (număr de modele *număr de seturi de date* \ 1 punct = 20 de puncte) Pentru fiecare set de date aplicați 5 modele de clasificare din scikit learn. Pentru fiecare raportați: acuratete, scorul F1 - a se vedea [sklearn.metrics \(http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics\)](http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics) - folosind 5 fold cross validation. Raportați mediile rezultatelor atât pentru fold-urile de antrenare, cât și pentru cele de testare. Rularile se vor face cu valori fixate ale hiperparametrilor.
3. (număr modele * 4 puncte = 20 puncte) Documentați în jupyter notebook fiecare din modelele folosite, în limba română. Dacă același algoritm e folosit pentru mai multe seturi de date, puteți face o secțiune separată cu documentarea algoritmilor + trimitere la algoritm.
4. (număr de modele *număr de seturi de date* 1 punct = 20 de puncte) Raportați performanța fiecărui model, folosind 5 fold cross validation. Pentru fiecare din cele 5 rulari, cautați hiperparametrii optimi folosind 4-fold cross validation. Performanța modelului va fi raportată ca medie a celor 5 rulari. *Observație:* la fiecare din cele 5 rulari, hiperparametrii optimi pot diferi, din cauza datelor utilizate pentru antrenare/validare.

Se acorda 20 de puncte din oficiu.

Exemple de modele de clasificare:

1. [Multi-layer Perceptron classifier \(https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier\)](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier)
2. [KNN \(https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier\)](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier)
3. [SVM \(https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC\)](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC)
4. [Gaussian processes \(https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessClassifier.html#sklearn.gaussian_process.GaussianProcessClassifier\)](https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessClassifier.html#sklearn.gaussian_process.GaussianProcessClassifier)
5. [RBF \(https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.RBF.html#sklearn.gaussian_process.kernels.RBF\)](https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.RBF.html#sklearn.gaussian_process.kernels.RBF)
6. [Decision tree \(https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier\)](https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier)
7. [Random forest \(https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier\)](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier)
8. [Gaussian Naive bayes \(https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB\)](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB)

Predare:

1. Fiecare student va depune pe site-ul de elearning fisier Jupyter notebook sau arhiva cu astfel de fisiere;
2. In fiecare fisier se specifica numele celor doi studenti care au lucra in echipa.
3. Predarea se face in saptamana 13-17 aprilie 2020
4. Revedeti formele ulterioare ale acestui document pentru precizari despre: continutul rezultatelor raportate, modalitate de notare.