**Documentatie**

Lungu Andrei 232

1. Clasificator

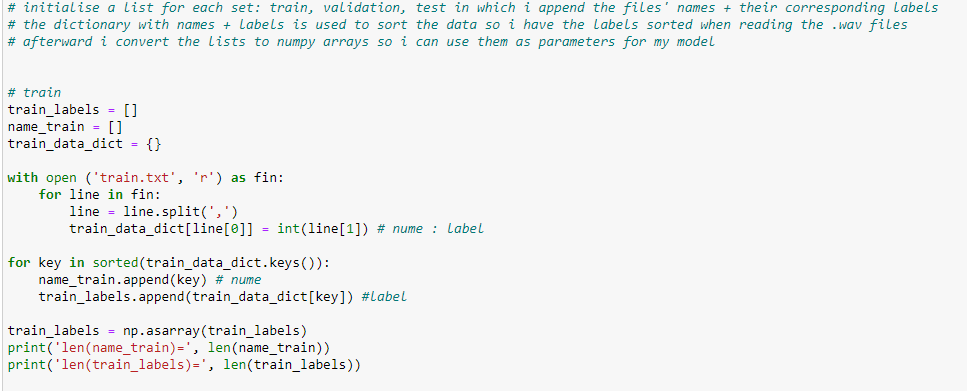
Cel mai bun rezultat(private score) l-am obtinut folosind MultinomialNB (diferenta mica fata de SVC), dupa mai multe incercari. Am verificat urmatoarele modele: SVC, NB, KNN, MLP, Perceptron si Linear Regression. Pe kaggle(public scores) am obtinut scorurile urmatoare:

* Perceptron 0.56222
* NB 0.59222
* SVC 0.60111

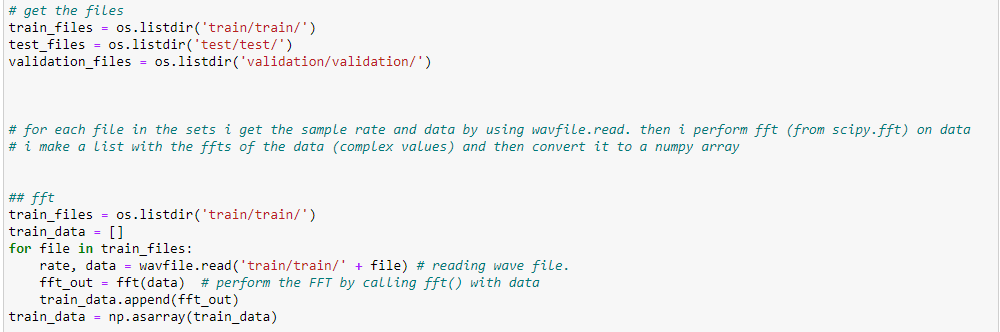
Initial am antrenat modelul pe data de la citire fara nicio prelucrare si nu am trecut de baseline, asa ca am incercat sa o modific aplicand algoritmul fft pe aceasta, obtinand scoruri mai bune. Acuratetea diferea foarte putin indiferent de modelele folosite sau parametrii lor (concluzia: extragerea de features este foarte importanta.. next time).

In principiu, metoda este aceeasi peste tot: citirea datelor, prelucrarea sau nu cu fft, diferente mai mari au fost cand am folosit train\_test\_split (cu procentaje diferite pe train si test) fie doar pe train data, fie pe train + validation, sau cand am antrenat doar pe train data.

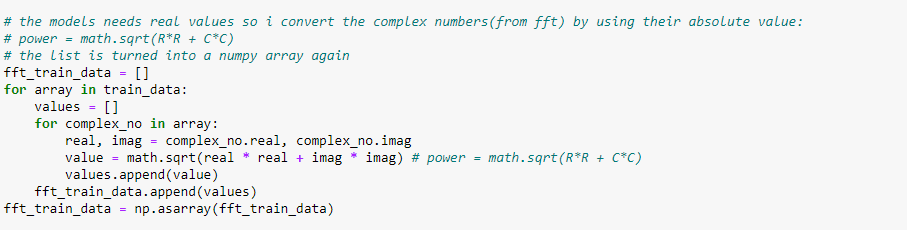
Preluarea etichetelor am facut-o in felul urmator (la fel si pentru test si validaiton):



Aplicarea alg fft:

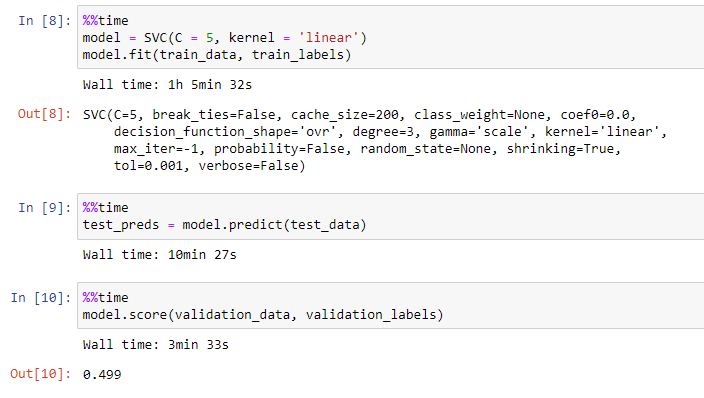


Intrucat fft returna valori complexe care nu puteau fi folosite direct la modele, le-am prelucrat in felul urmator, folosind modulul lor:



Exemplu de antrenare model:

A)

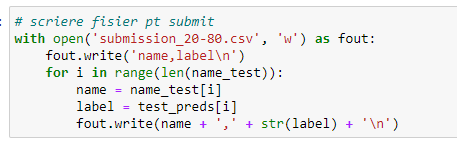


B)



(am concatenat train si validation, dupa care am impartit in diferite procentaje, pentru train mereu < 0.4 pentru a nu face overfit)

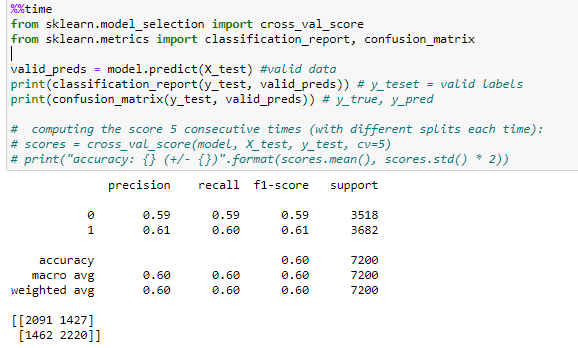
Pentru a scrie fisierele pentru submit:



1. Parametri diferiti, confusion\_matrix, accuracy, classification\_report(precision + recall)

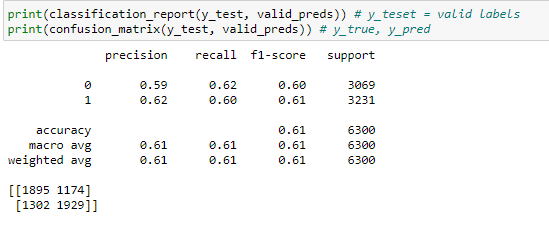
SVC: C = 1000



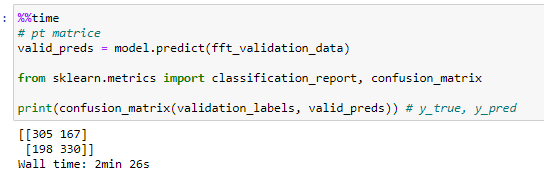


SVC C = 10000



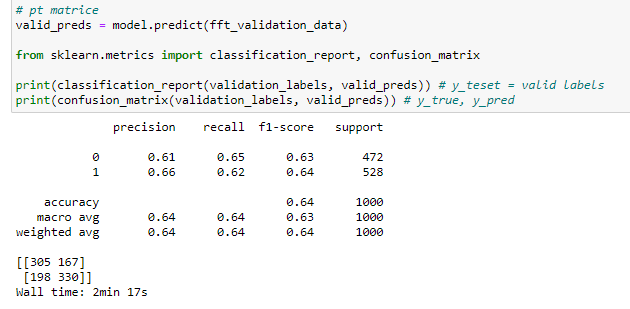




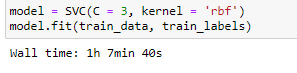


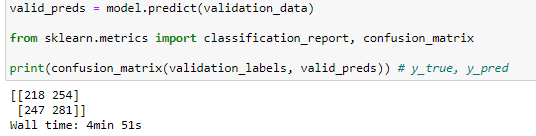
SVC C = 1





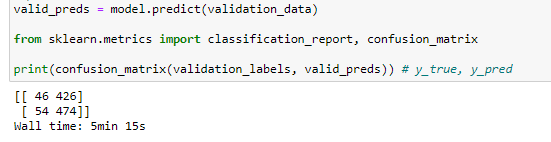
SVC kernel rbf





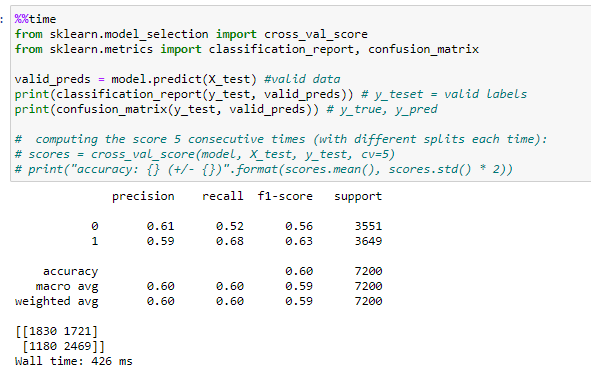
SVC kernel poly





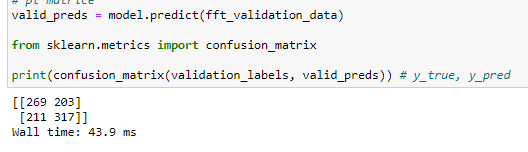
NB





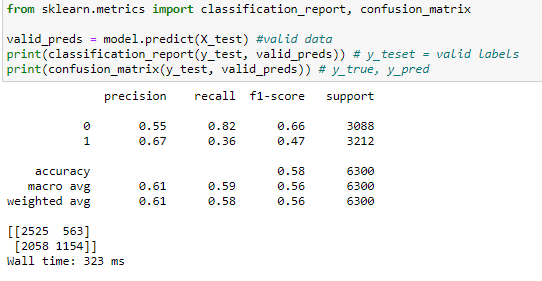
NB alpha = 5





PERCEPTRON





KNN



