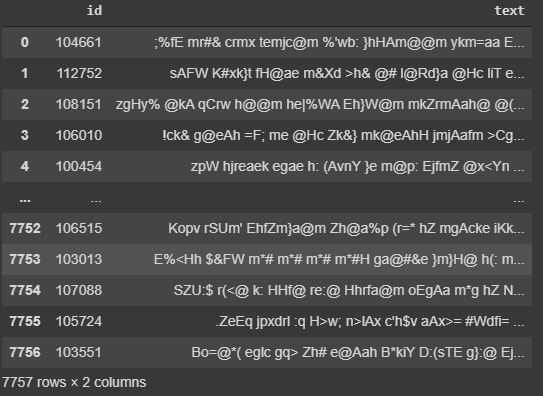
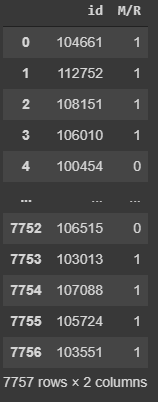
**DOCUMENTATIA**

Mai intai am citit continutul fisierelor si am creat 5 dataframe-uri respectiv:

* x\_train corespunde fisierului “train\_samples.txt”



* y\_train corespunde fisierului “train\_labels.txt”



* x\_validation corespunde fisierului “validation\_samples.txt”
* y\_validation corespunde fisierului “validation\_labels.txt”
* x\_test corespunde fisierului “

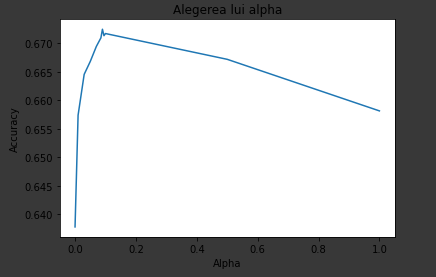
Apoi am concatenat train-dataframe cu validation-dataframe (am inteles ca putem face asta pentru a fi acuratetea modelului mai mare). Dupa am eliminat coloana ‘id’ deoarece era irelevanta pentru model.

Deoarece modelul pe care urma sa-l creez nu putea lucra cu string-uri am decis sa folosesc TfidfVectorizer (asemanator cu metoda Bag of Words).

Am folosit ca parametri pentru TfidfVectorizer: ngram\_range = (1,2) pentru unigrame si bigrame; max\_df = 0.7 pentru a ignora toate cuvintele cu o frecventa in document mai mare de 0.7 (pentru a scapa de “stop words”)

Dupa care am transformat x\_train si x\_test pastrand noile valori in x\_train\_count respectiv x\_test\_count.

Am ales clasificatorul Multinomial Naïve Bayes cu un alpha = 0.09 pentru modelul meu (folosisem initial un GridSearchCV pentru a vedea ce model s-ar potrivi cel mai bine si totodata pentru a face tuning pe parametri).



Mi-am antrenat modelul cu x\_train\_count si y\_train, dupa care am folosit metoda predict pentru a prezice label-urile pentru sample-urile din x\_test\_count.

Am creat un nou dataframe care sa contina label-urile pe care modelul meu le-a prezis dupa care l-am scris in “submission.csv”

Pentru acest model (dar antrenat doar cu (x\_train, y\_train) si testat cu (x\_validation, y\_validation)) am obtinut:

* Accuracy: 0.6724397590361446
* F1 score: 0.6775389177168273
* Confusion matrix:

[ [872, 429]

[441, 914] ]

