Proiect 1 MPS - COVID Suspects

**-RAPORT TESTARE-**

**Echipă:**

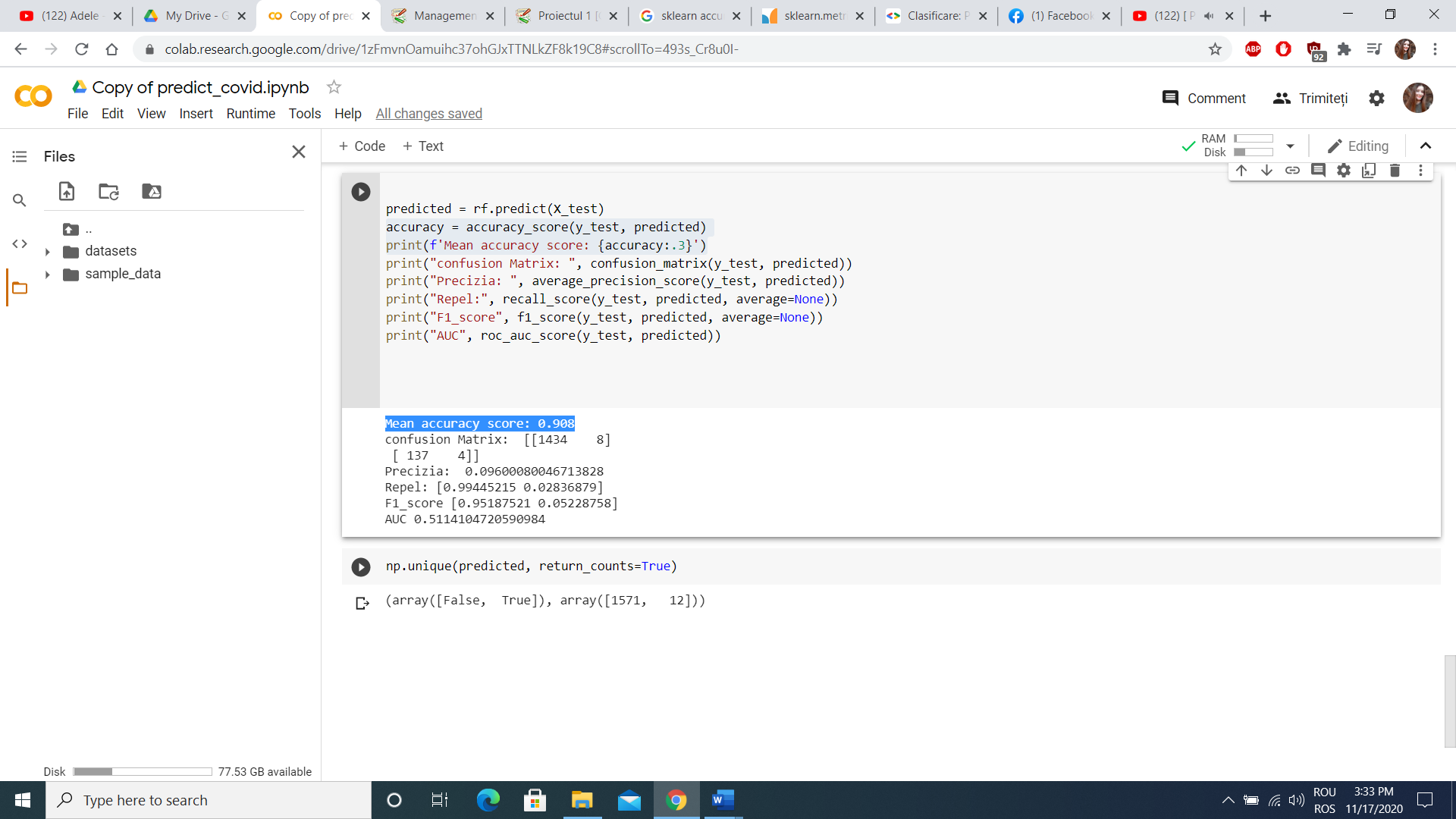
* Project Manager - Tafta-Potop Tiberiu
* Team Lead - Bercu Mihaela-Bianca
* Tester - Sava Diana Andreea
* Developer - Gheorghe Tudor-Razvan
* Developer - Filip Tepes-Onea

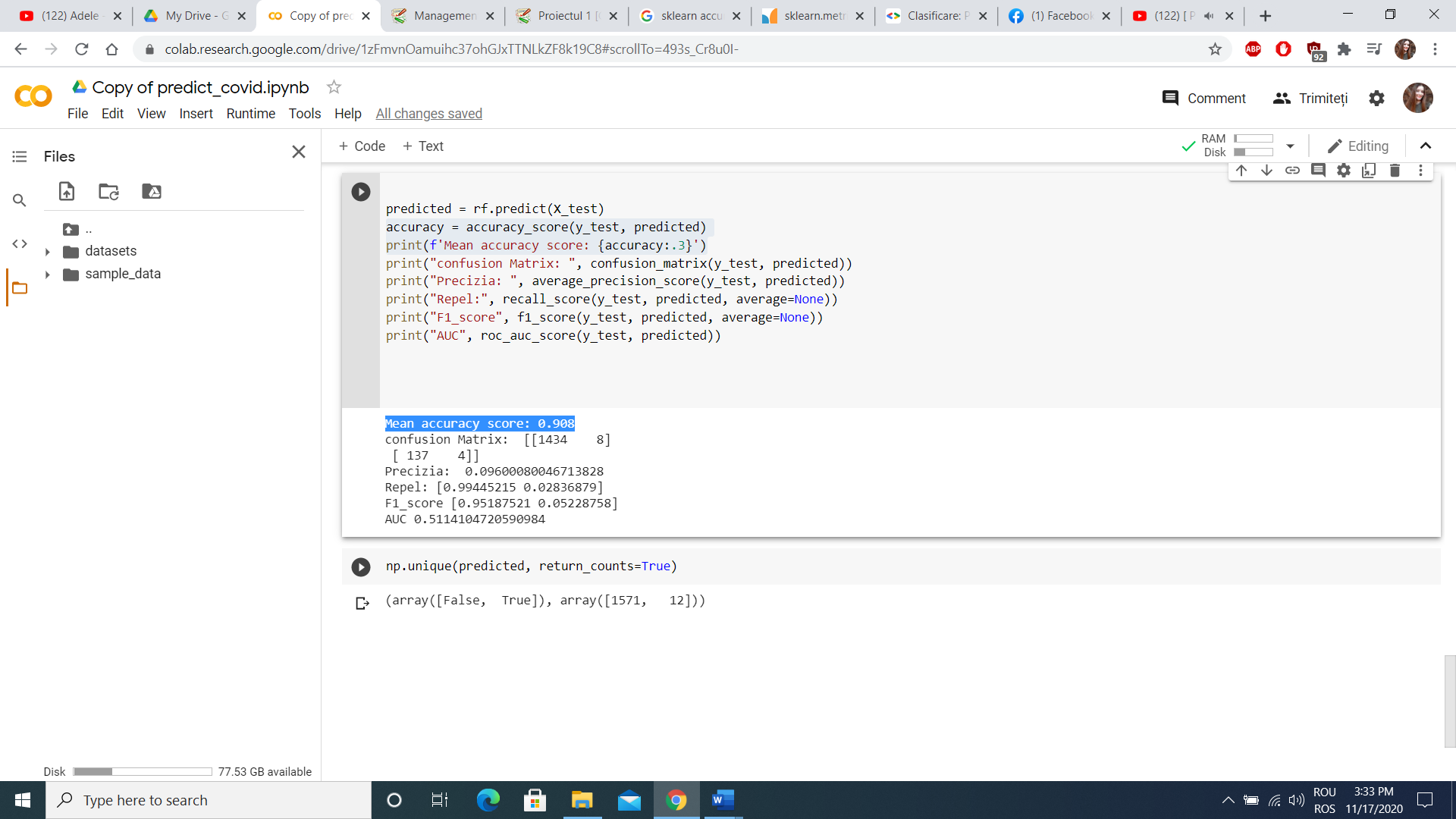
**Măsuri de performanță:**

1. Acuratețea
2. Precizia
3. Rapelul
4. Scorul F1
5. Matricea de confuzie
6. AUC
7. **Acuratețea**

Este o metrică pentru evaluarea modelelor și reprezintă numărul de exemple corect clasificate / numărul total de exemple. Pentru a o calcula am folosit funcția: sklearn.metrics.accuracy\_score(y\_true, y\_pred, \*, normalize=True, sample\_weight=None), a fost aplicată în cazul nostru pe y\_test(o parte din setul total de date rezervată testării) și predicted(rezultatele prevăzute de modelul de antrenare).

Din cele 2 imagini de mai jos se poate observa că am obținut o acuratețe de 90%:



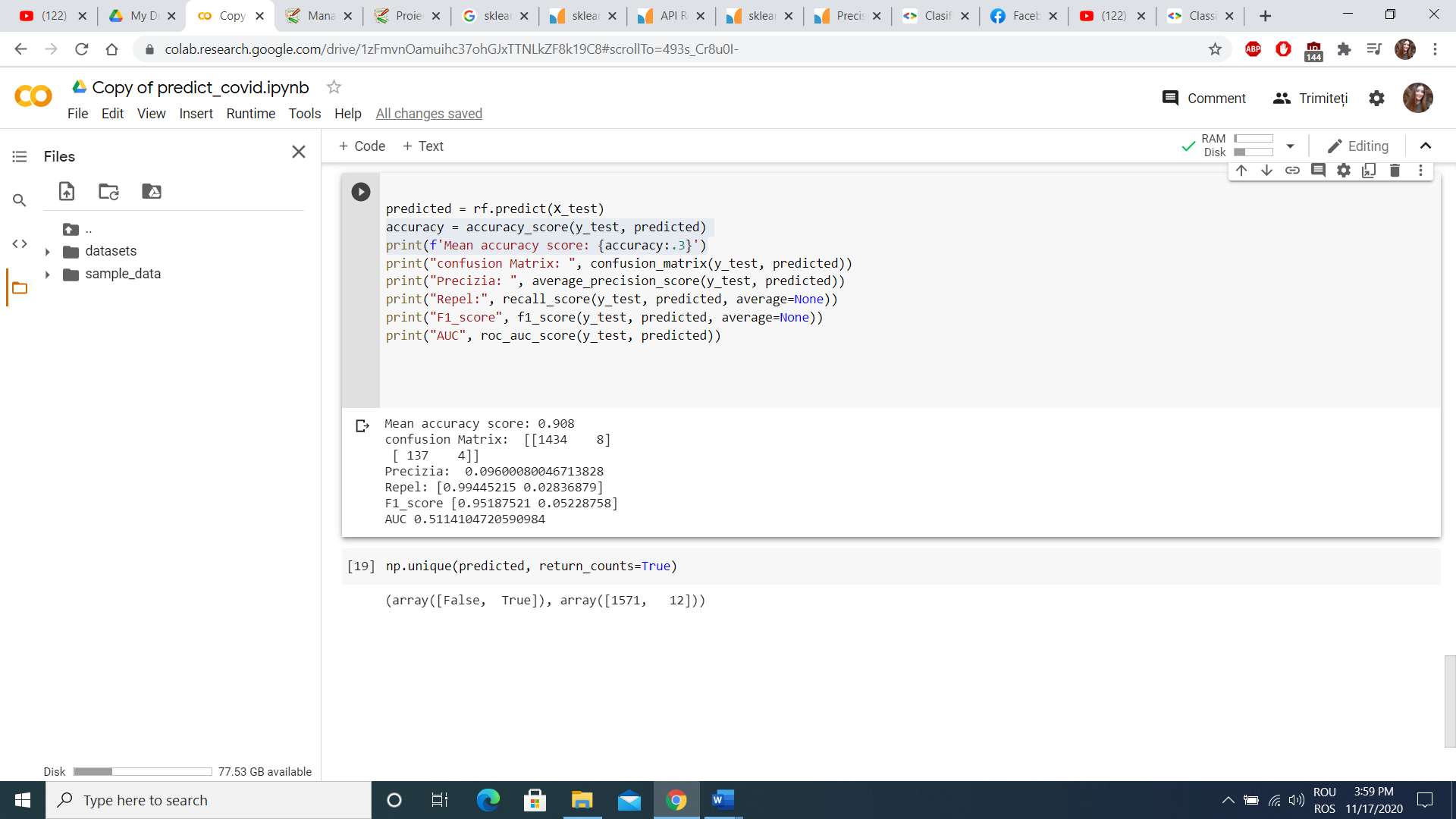


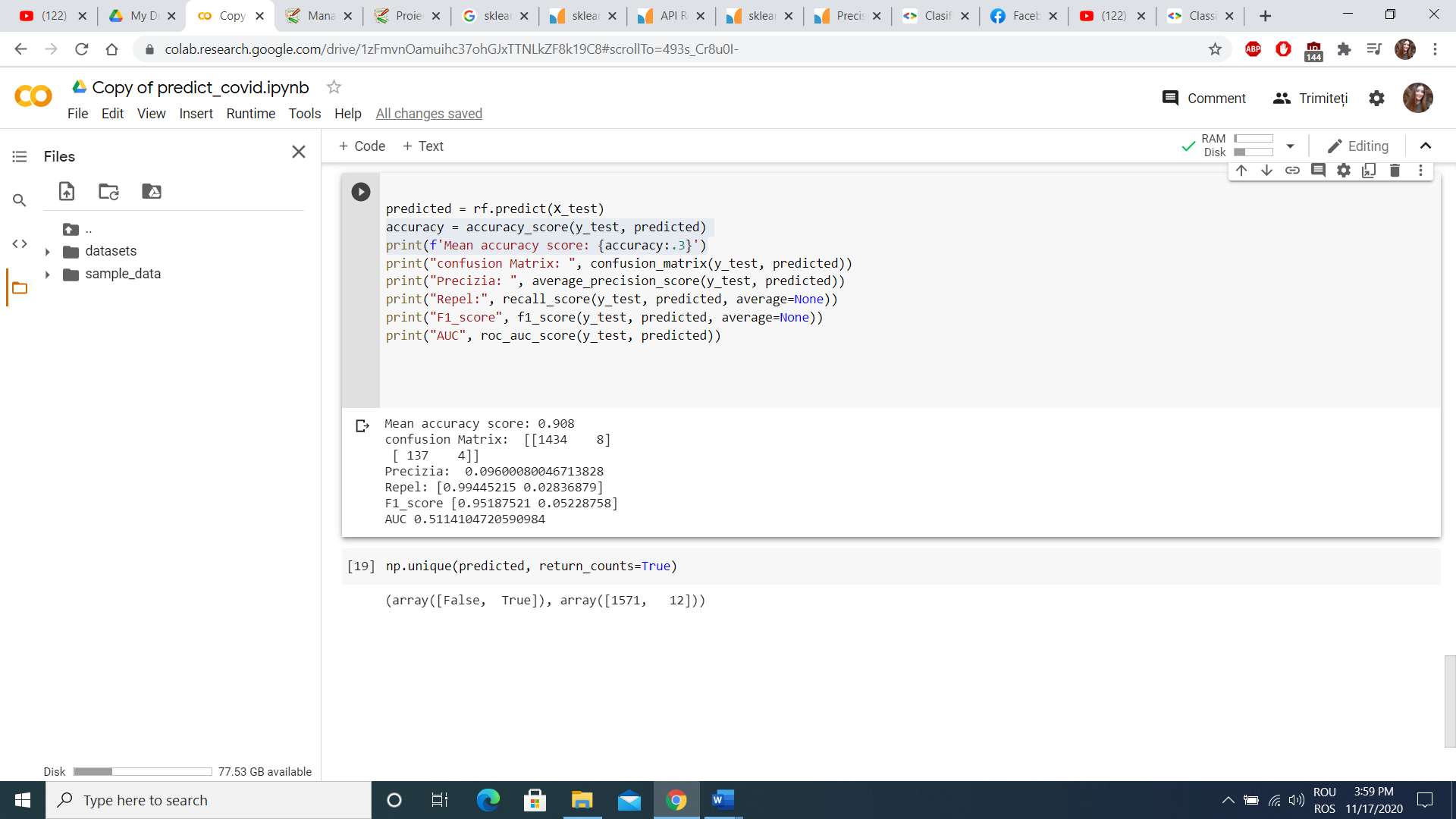
1. **Precizia**

Este numărul de exemple positive corect clasificate / numărul total de exemple clasificate ca positive. A fost calculată folosind funcția sklearn.metrics.average\_precision\_score(y\_true, y\_score, \*, average='macro', pos\_label=1, sample\_weight=None.

Precizia = TP / (TP + FP) (TP = true positive, FP = fals positive)

Din următoarele imagini rezultă că in 10% din cazuri se prezice că e adevărat





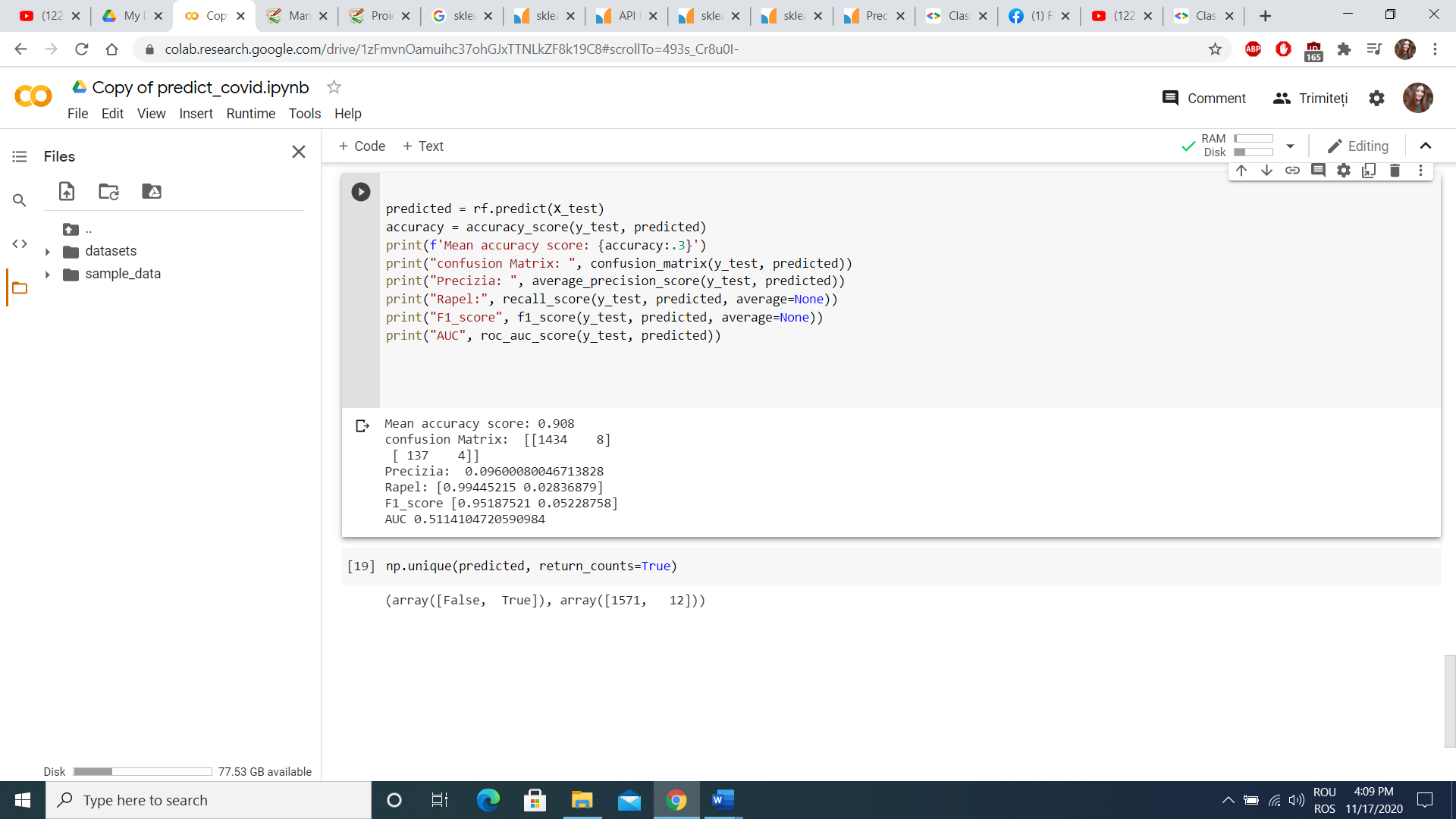
1. **Rapelul**

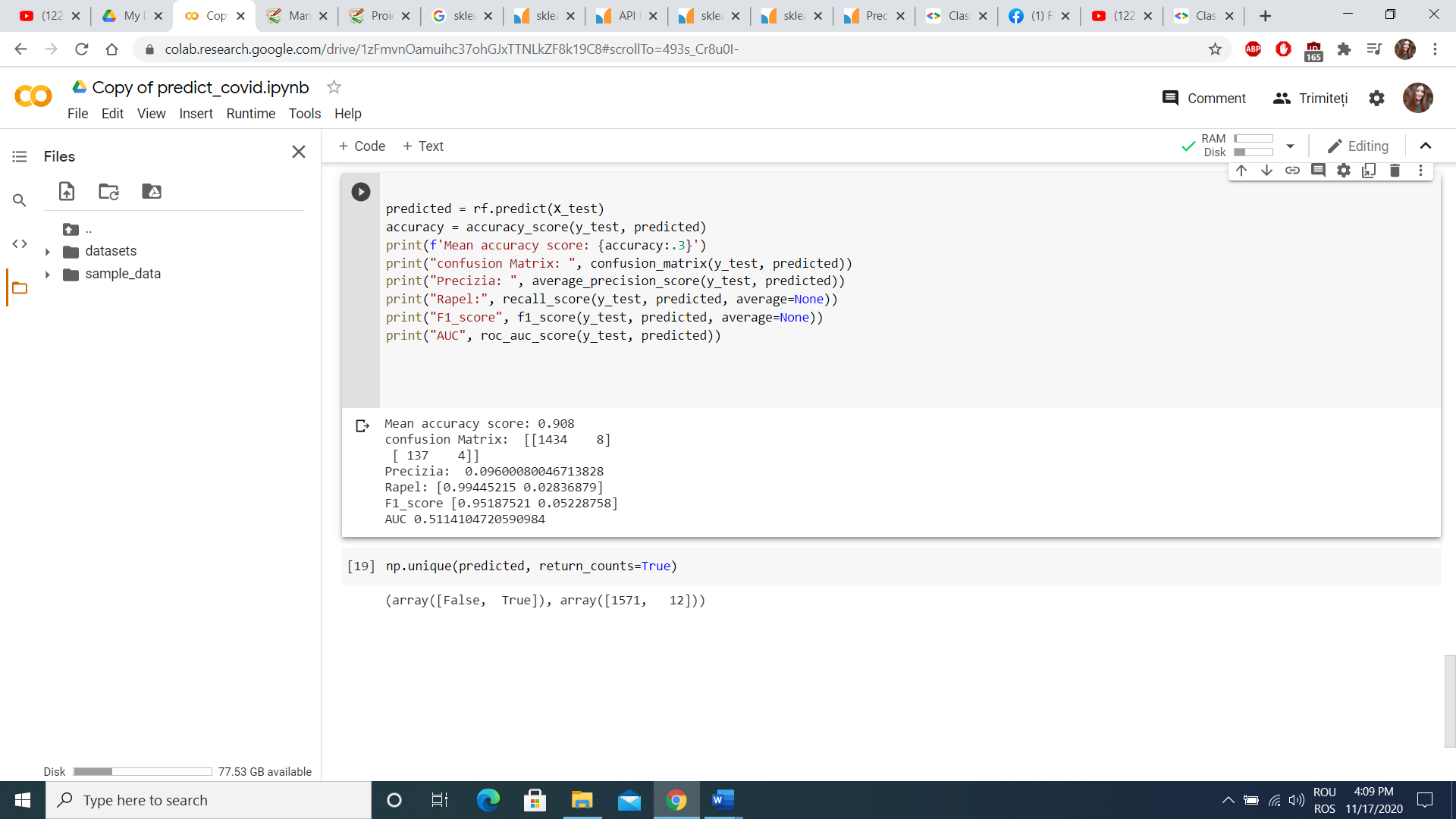
Este numărul de exemple pozitive corect clasificate / numărul total de exemple positive, probabilitatea ca un exemplu pozitiv să fie identificat corect de către clasificator. Acesta a fost calculate astfel:

sklearn.metrics.recall\_score(y\_true, y\_pred, \*, labels=None, pos\_label=1, average='binary', sample\_weight=None, zero\_division='warn')

Cea mai buna valoarea rezultata pentru recall este 1 si cea mai rea este 0.

Recall = tp / (tp + fn) (TP = true positive, FN = fals positive)

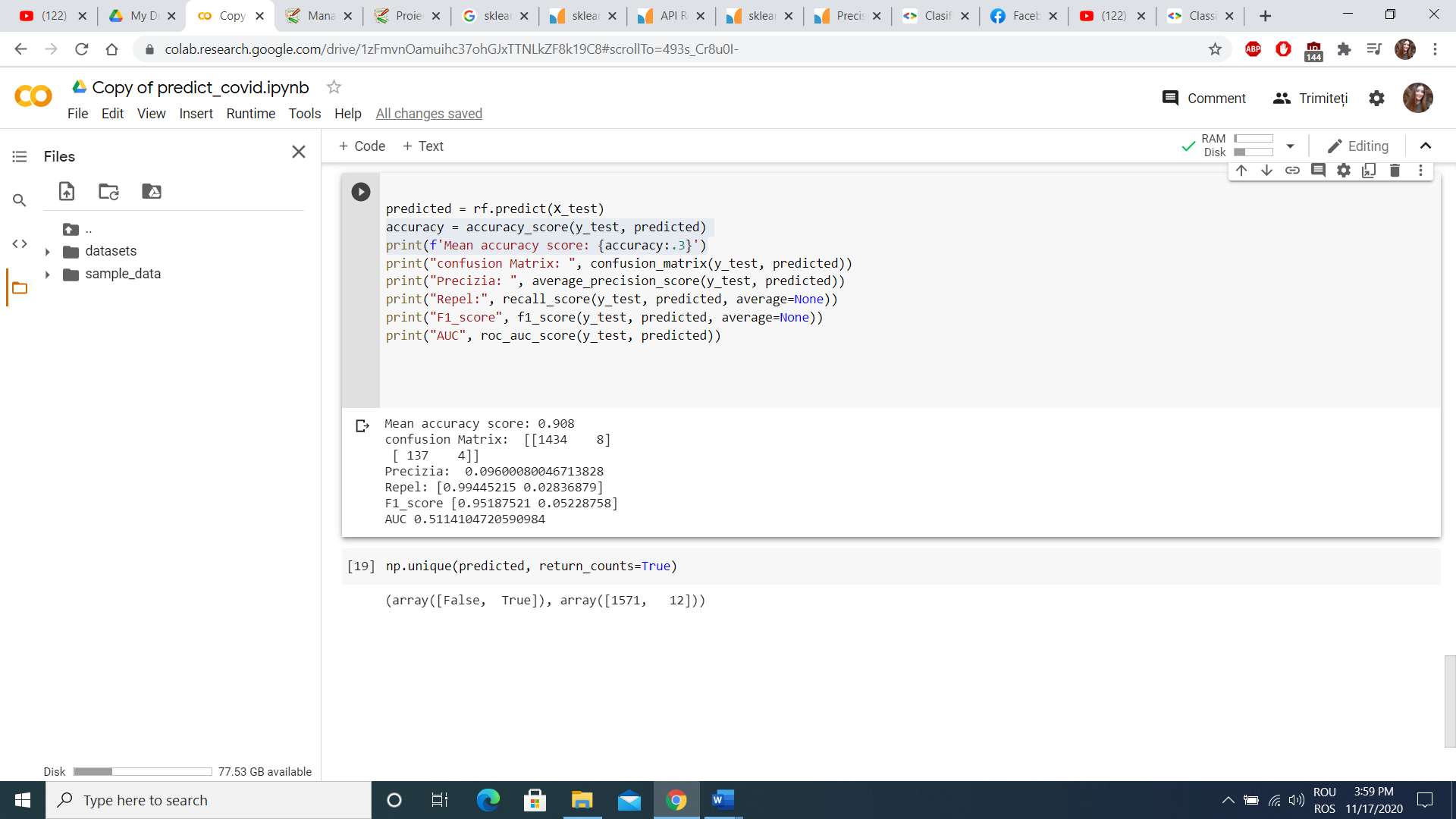




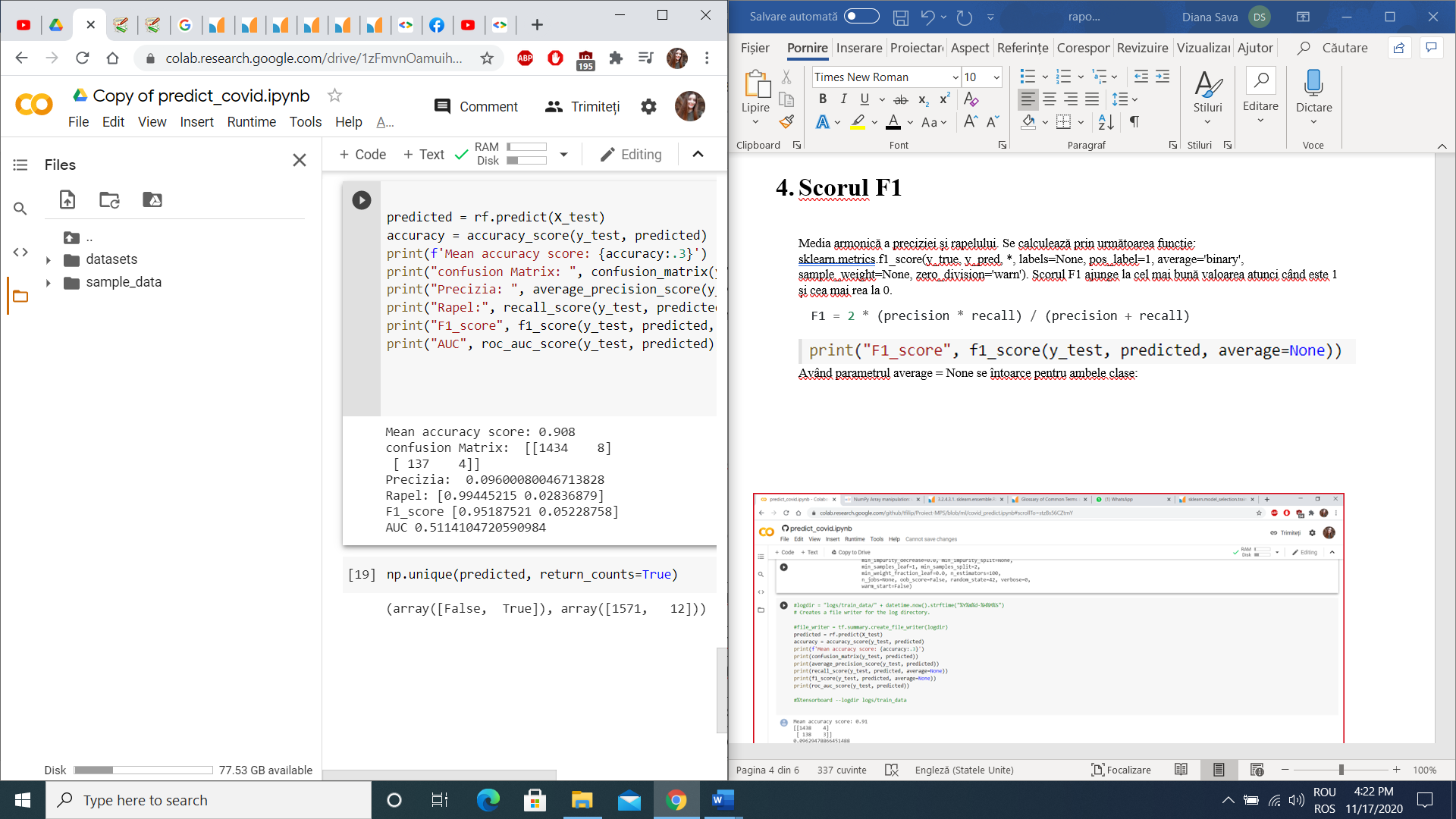
1. **Scorul F1**

Media armonică a preciziei și rapelului. Se calculează prin următoarea funcție: sklearn.metrics.f1\_score(y\_true, y\_pred, \*, labels=None, pos\_label=1, average='binary', sample\_weight=None, zero\_division='warn'). Scorul F1 ajunge la cel mai bună valoarea atunci când este 1 și cea mai rea la 0.

F1 = 2 \* (precision \* recall) / (precision + recall)



Având parametrul average = None se întoarce pentru ambele clase:



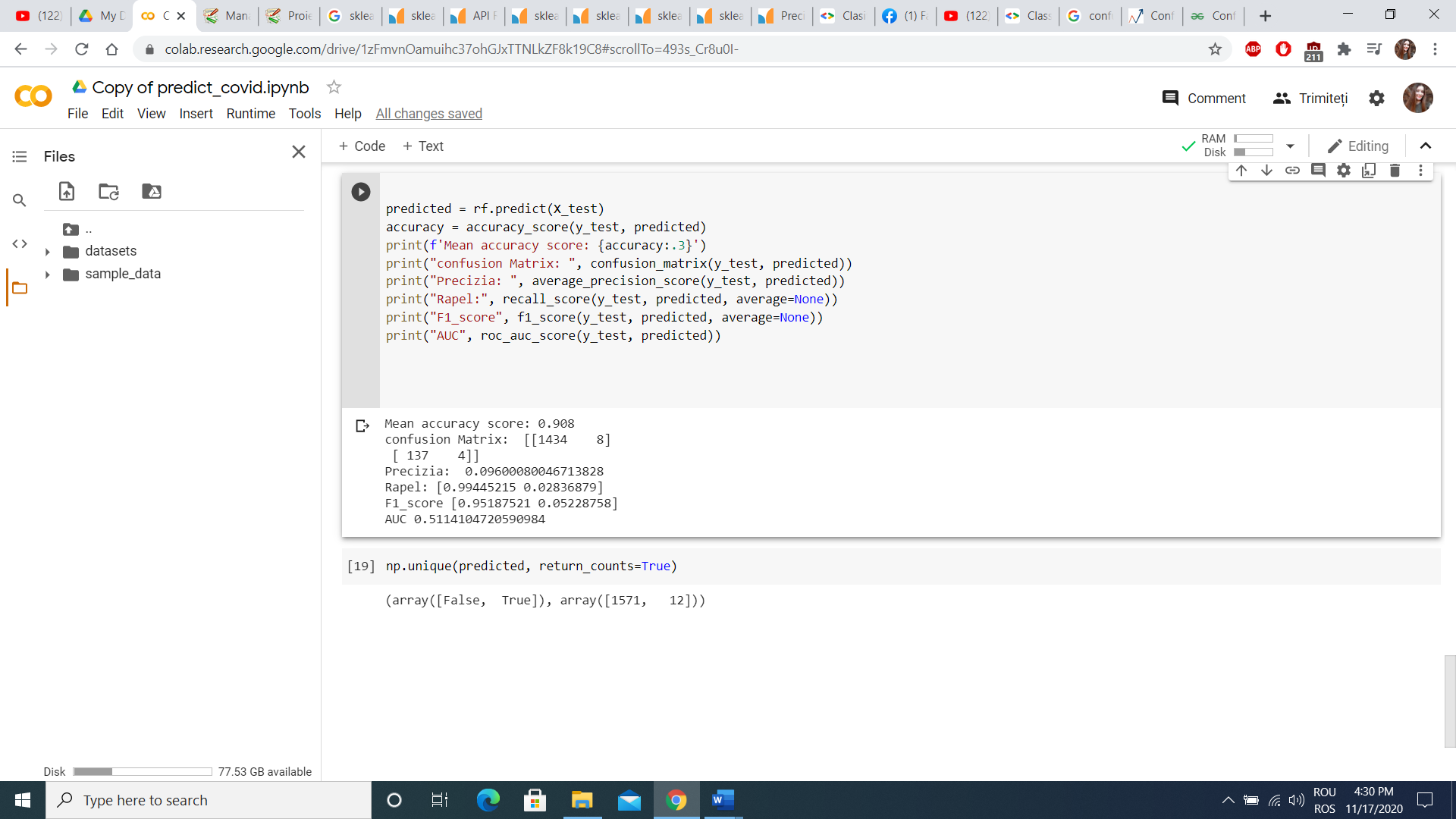
1. **Matricea de confuzie**

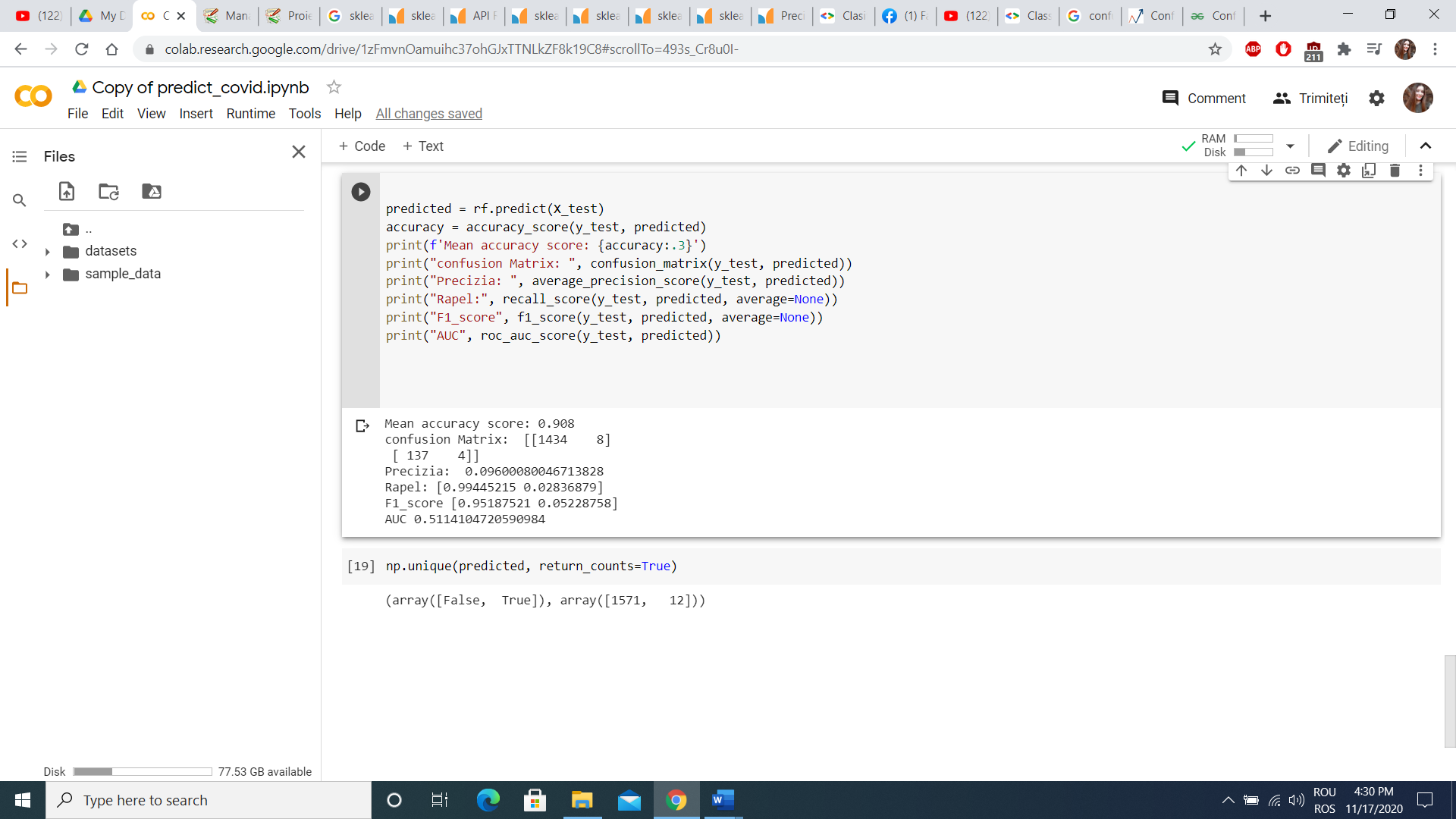


Matricea de confuzie este o matrice NxN folosită pentru a evalua performanța unui model, unde N este numărul de clase. În cazul nostru, este clasificare binară. Aceasta compară adevăratele valori cu cele prezise de modelul ml-ului. Astfel, se poate vedea cât de bine merge și ce performanțe are modelul.

Am obținut această matrice, atfel:

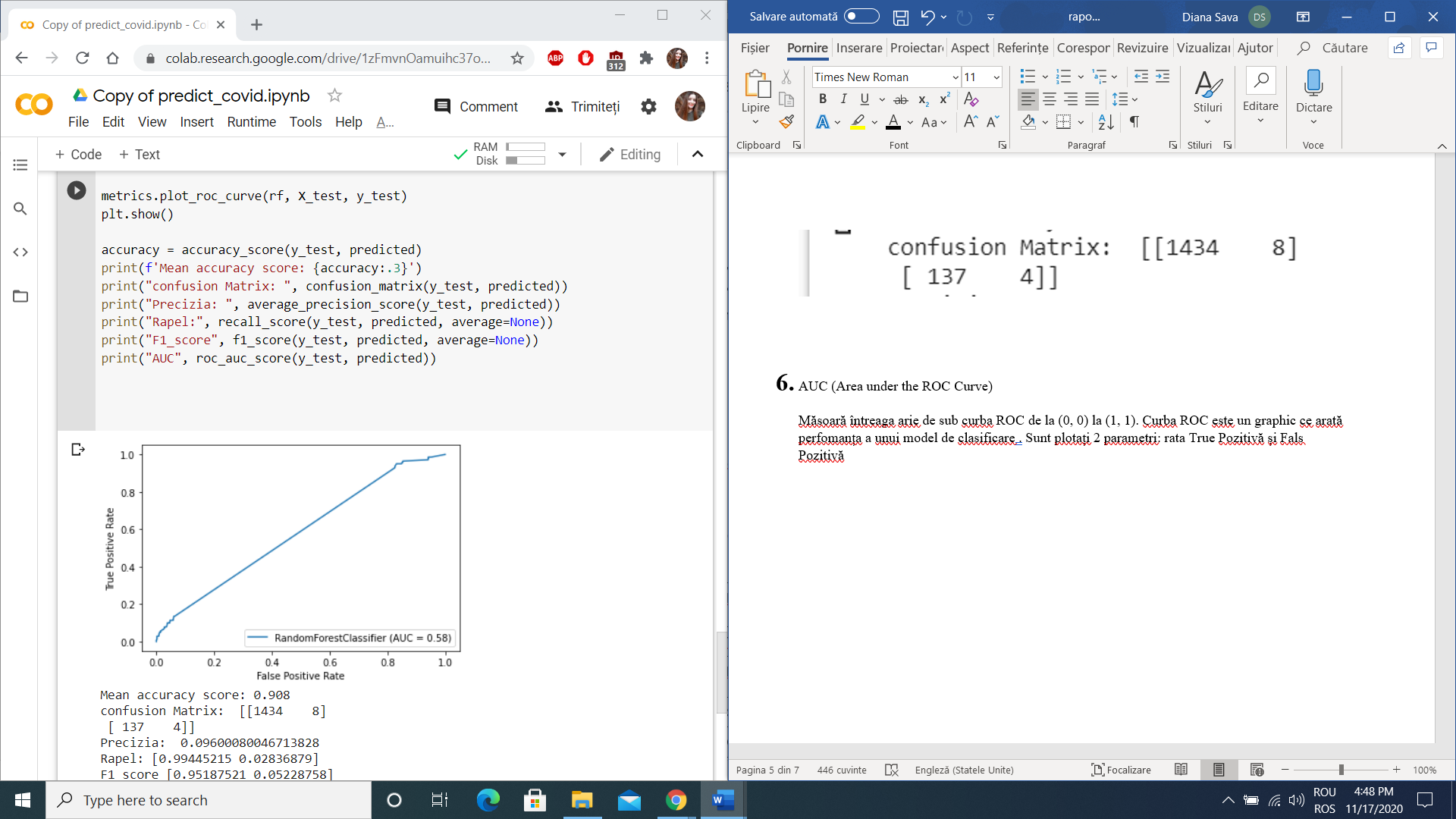
sklearn.metrics.confusion\_matrix(y\_true, y\_pred, \*, labels=None, sample\_weight=None, normalize=None)





1. **AUC (Area under the ROC Curve)**

Măsoară întreaga arie de sub curba ROC de la (0, 0) la (1, 1). Curba ROC este un graphic ce arată perfomanța a unui model de clasificare . Sunt plotați 2 parametri: rata True Pozitivă și Fals Pozitivă



Am folosit funcția: sklearn.metrics.roc\_auc\_score(y\_true, y\_score, \*, average='macro', sample\_weight=None, max\_fpr=None, multi\_class='raise', labels=None)

