**Documentatie Proiect ML (Smartphone User Classification)**

SUBMISIA 1

**Cuprins:**

1. **Descrierea submisiei**
2. **Descrierea modelului ales**
3. **Media preciziei obtinute in urma folosirii unui 3-fold cross-validation pe setul de date de antrenare. Matricile de confuzie pt fiecare combinatie**
4. **Alte metode incercate**
5. **Descrierea submisiei**

Clasificator ales: SVM.SVC()

Parametri:

* C = 1
* Kernel = "linear"

Precizie:

* locala: 0.9668888888888889
* Kaggle:
  + clasament public: 9.05
  + clasament privat: 0.89775

Citirea si memorarea datelor:

* Dimensiunea CSV-urilor din fisierele primite era inegala:
  + train\_data: MIN = 139 | MAX = 156
  + test\_data: MIN = 136 | MAX = 159
* Solutie: Le-am uniformizat prin dublarea ultimei linii de mai multe ori astfel incat sa obtin o matrice de tip 9000x159x3, respectiv 6000x159x3

Extragerea unor features + Modelarea datelor de antrenare si de testare:

* Calcularea fft-urilor semnalelor accelerometrului (Fast Fourier Transform = un algoritm care converteste un semnal prin impartirea acestuia in componente periodice intr-o reprezentare a frecventelor acestor componente repetitive) cu ajutorl bibliotecii SciPy
* Aplicarea unor functii statistice pe coloane(atat pt x, y, z, cat si pt fft\_x, fft\_y si fft\_z) implementate in biblioteca Statistics
* In final sunt 516 features

Normalizarea datelor:

* Am folosit scalarea standard: fiecare dimensiune este scalată folosind deviația standard corespunzătoare, transformă vectorii caracteristici astfel incat fiecare sa aiba media 0 si deviatia standard 1

Deoarece alegerea parametrilor este critica in performanta SVM-ului, aplicand GridSearchCV am gasit cea mai buna combinatie a acestora pentru clasifcatorul ales si datele modelate.

1. **Descrierea modelului ales**

**SVM**-urile ( Support Vector Machines = Masini cu Vectori Suport ) sunt folosite pentru clasificare, regresie si detectarea outlierelor (valori statistice de exceptie).

SVM-ul alege hiperplanul care maximizeaza marginea dintre clasele asignate datelor.

Avantaje:

* Eficiente in spatiile cu multe dimensiuni
* Eficient in cazurile in care numarul de dimensiuni este mai mare decat numarul de date
* Optim din punctul de vedere al memoriei deoarece foloseste un subset din punctele de antrenare in functiile de decizie
* Permite implementarea unor functii kernel diferite (atat cele predefinite, cat si cele customizate)

Dezavantaje:

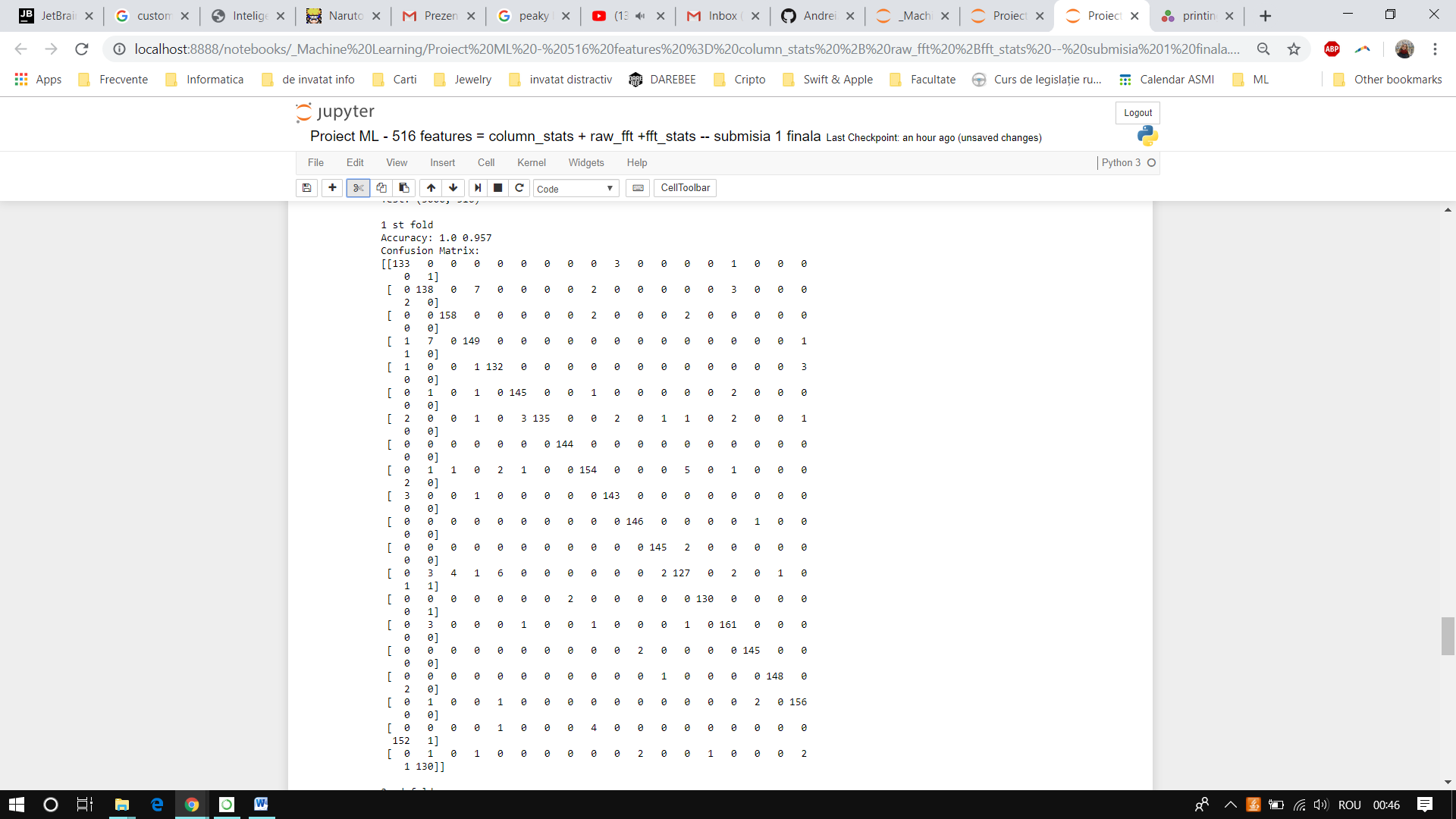
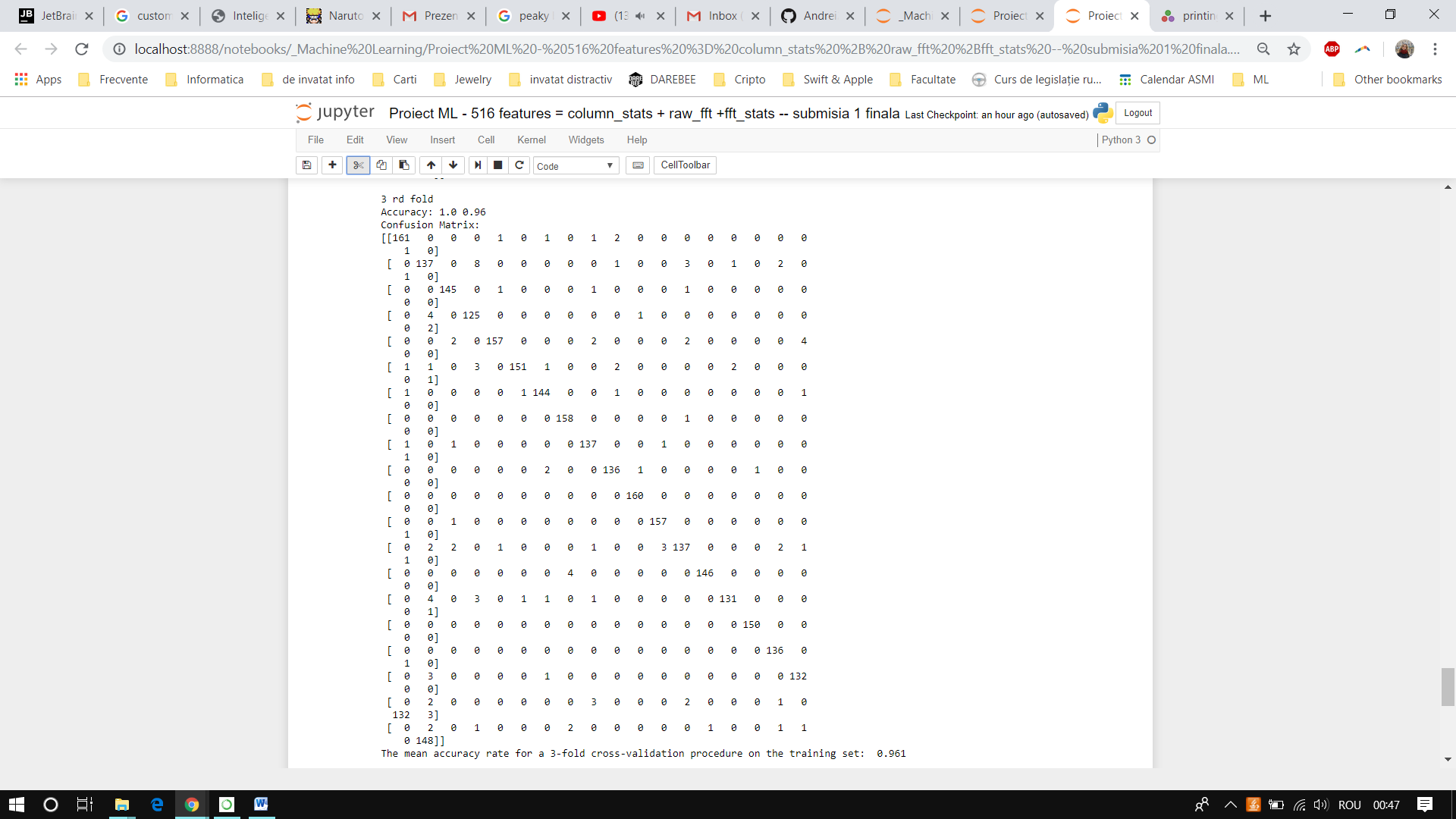
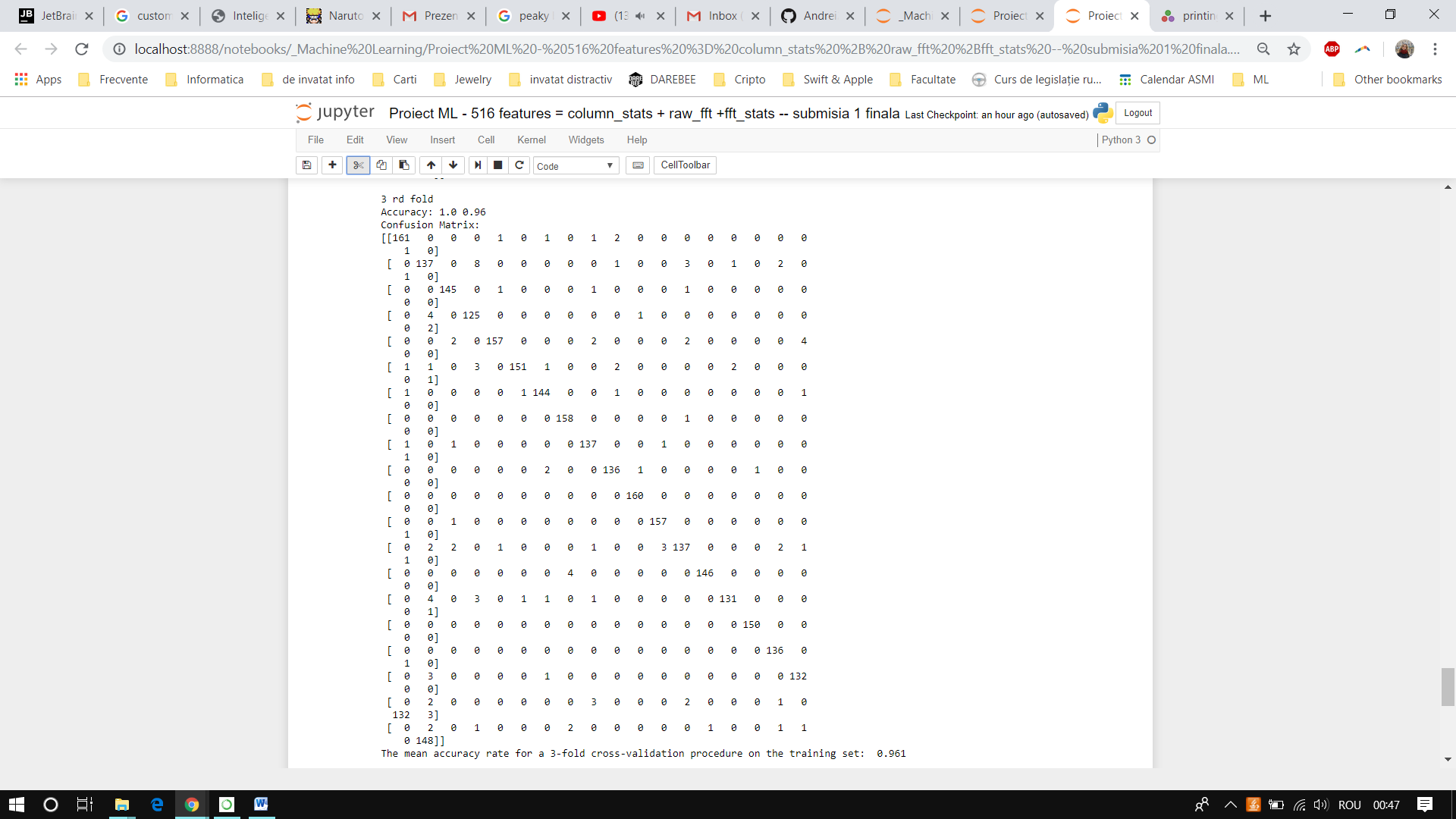
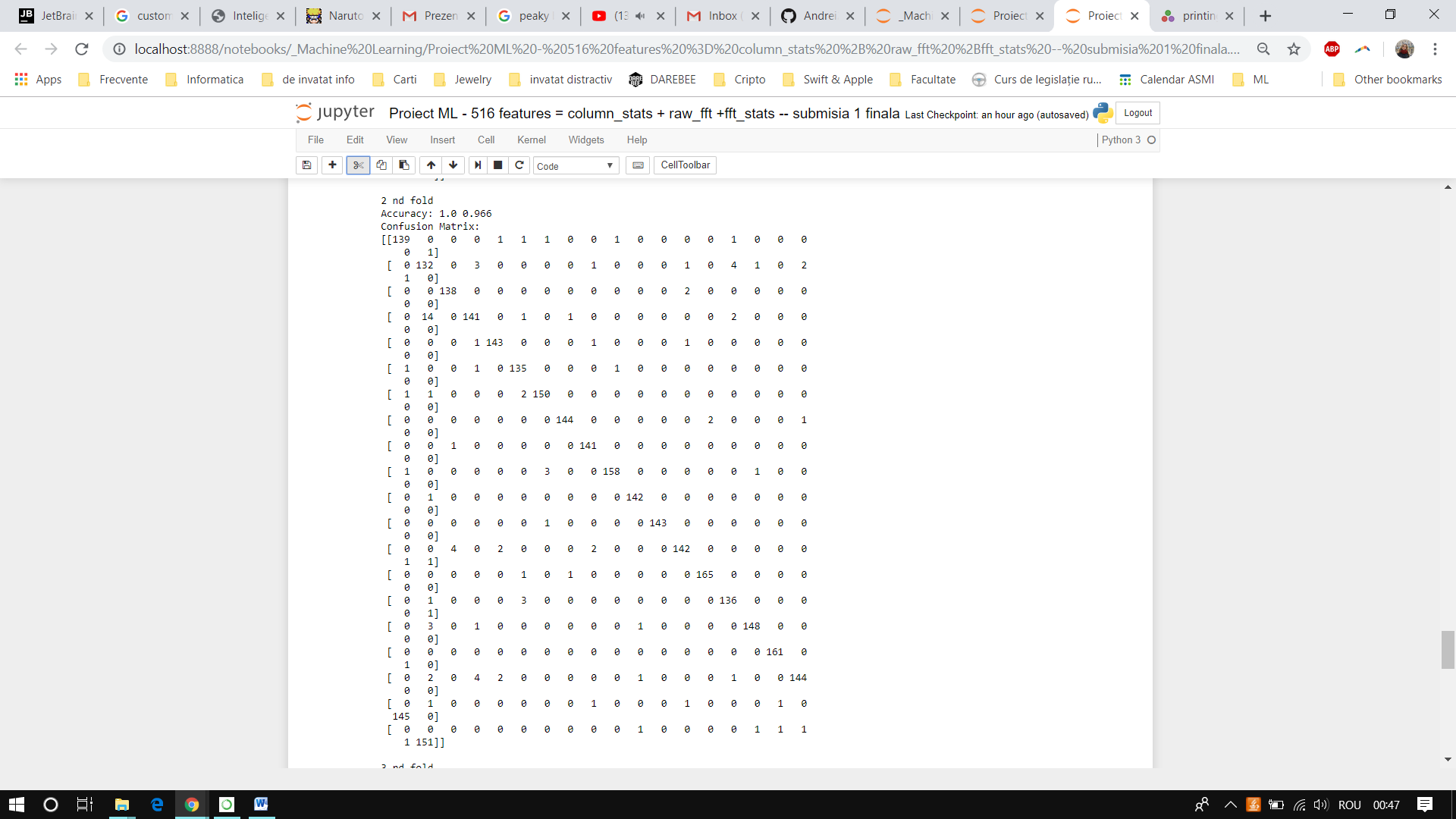
* Daca numarul de features este mult mai mare decat numarul de date, atunci pentru a evita suprainvatare(over-fitting-ul) este crucial sa se aleaga functii kernel si sa modeleze datele
* Nu calculeaza precizia singur. Este nevoie de alti algoritmi cum ar fi k-fold cross-validation

Clase: SVC, NuSVC si LinearSVC.

**SVC:**

* **class sklearn.svm.SVC(C, kernel, gamma)**
* Clasa capabila de multi-clasificare pe un set de date
* Implementarea acesteia din ScikitLearn are o abordare one-vs-one, adica pentru fiecare 2 clase este antrenat un clasificator binar care sa diferentieze intre acestea. Astfel, dacă avem un numar de clase egal cu 𝑛𝑢𝑚\_𝑐𝑙𝑎𝑠𝑠𝑒𝑠, vor fi antrenați 𝑛𝑢𝑚\_𝑐𝑙𝑎𝑠𝑠𝑒𝑠∗(𝑛𝑢𝑚\_𝑐𝑙𝑎𝑠𝑠𝑒𝑠−*1*)/*2* clasificatori.
* La testare, clasa asignata fiecarui exemplu este cea care obtine cele mai multe „voturi” pe baza acestor clasificatori.
* Parametri frecventi:
  + C:
    - float, optional [default = 1.0]
    - parametru de penalitate pentru eroare, sugereaza cat de mult este dispus modelul sa evite clasificarea gresita a exemplelor din setul de antrenare:
    - observatii:
      1. C mare
         * va fi ales un hiperplan cu o margine mai mica
         * acesta are rezultate mai bune pe setul de antrenare
         * dacă C va fi ales prea mare, se poate ajunge la supraînvățare
      2. C mic
         * va fi ales un hiperplan cu o margine mai mare,
         * duce la clasificarea gresita a unor puncte din setul de antrenare
         * daca C va fi ales prea mic, modelul nu va fi capabil sa invete, ajungandu-se la subînvățare
  + Kernel:
    - string, optional [default = "rbf"]
    - valori posibile = "linear", "poly", "rbf", "sigmoid", "precomputed" sau customizat
    - Functiile kernel sunt folosite atunci cand datele nu sunt liniar separabile
    - Acestea funcționeaza prin urmatorii doi pasi:
      1. Datele sunt scufundate intr-un spațiu (Hilbert) cu mai multe dimensiuni
      2. Relatiile liniare sunt cautate in acest spatiu
  + Gamma
    - float, optional [default = "auto", avand valoarea *1*/𝑛𝑢𝑚\_𝑓𝑒𝑎𝑡𝑢𝑟𝑒𝑠
    - Doar pentru kernel de tipul: rbf, poly sau sigmoid.
    - Defineste cata influenta are o singura data din intregul set de antrenare
    - Cu cat este mai mare, cu atat restul exemplelor vor fi mai afectate
* Antrenarea
  + **svm\_model.fit(train\_data, train\_labels)** 
    - train\_ data: 𝑛𝑢𝑚\_𝑠𝑎𝑚𝑝𝑙𝑒𝑠 x 𝑛𝑢𝑚\_𝑓𝑒𝑎𝑡𝑢𝑟𝑒𝑠
    - train\_labels: etichetele corespunzatoare fiecarui exemplu de antrenare
* Predictia
  + **svm\_model.predict(test\_data)** 
    - test\_data: 𝑛𝑢𝑚\_𝑡𝑒𝑠𝑡\_𝑠𝑎𝑚𝑝𝑙𝑒𝑠 x 𝑛𝑢𝑚\_𝑓𝑒𝑎𝑡𝑢𝑟𝑒𝑠
    - fiecare element reprezentand id-ul clasei prezise

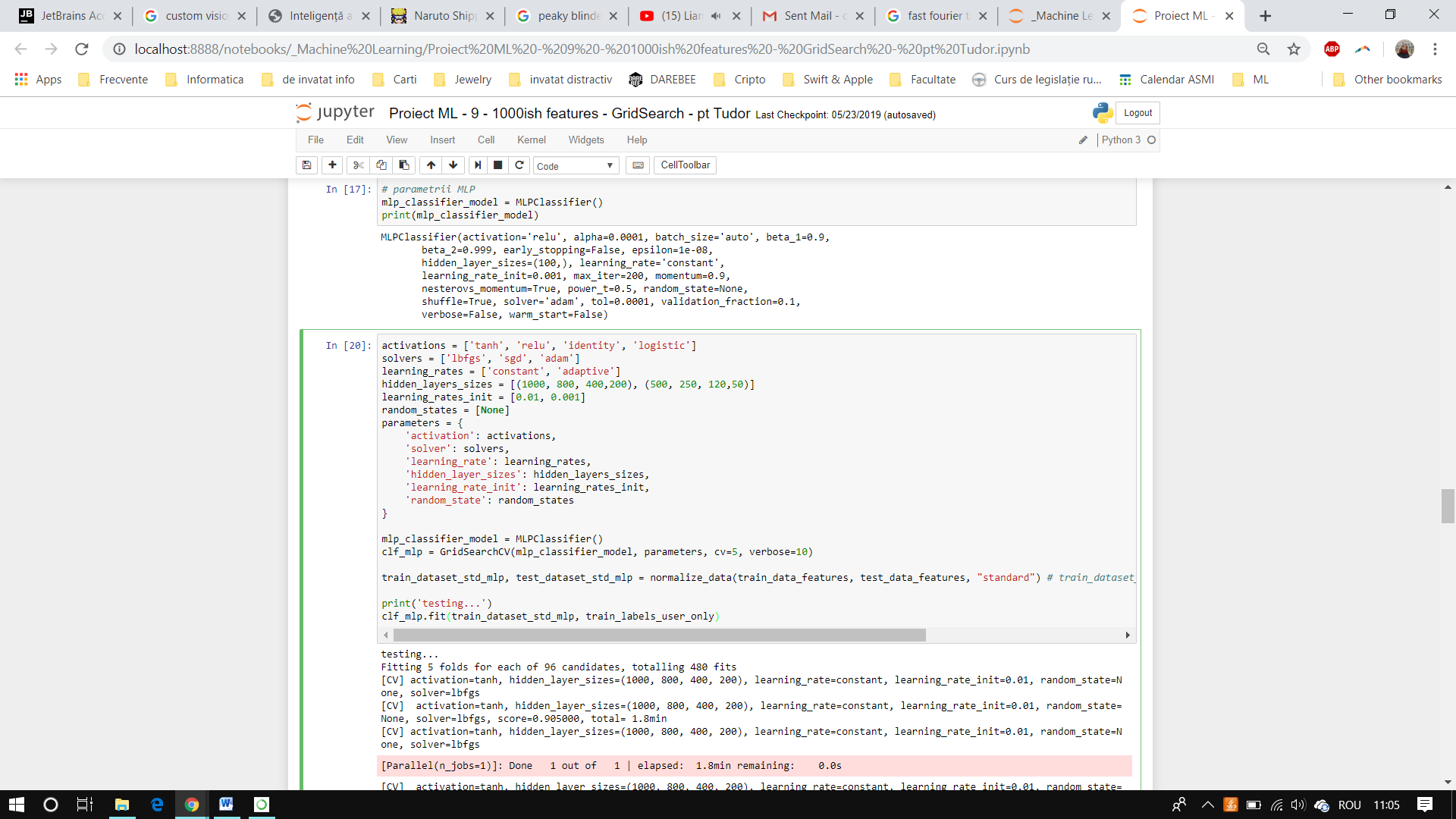
1. **Media preciziei obtinuta in urma folosirii unui 3-fold cross-validation pe setul de date de antrenare. Matricile de confuzie pt fiecare combinatie**



1. **Alte metode incercate**

Majoritatea solutiilor trimise au folosit clasificatorul SVM, in care am alternat kernelul intre linear si rbf si am schimbat valoarea C-ului in functie de datele trimise ca features.

Am observat ca numarul de features si corelatia acestora are o influenta majora asupra clasificatorului, motiv pentru care am combinat in cat mai multe moduri posibile caracteristicile extrase. Spre exemplu, initial incercam sa trimit datele nealterate. Ulterior am introdus statistici pe coloane pe grupulete de cate 40 de valori, iar mai apoi de cate 3-4. Am calculat Fast Fourier Transform pentru cele 3 axe ale accelerometrului si le-am introdus atat raw cat si cu statistici aplicate pe acestea.

Spre final m-am avantat in aria retelelor neuronale, insa fara rezultate exceptionale. Am incercat un MLP (Multilayer Perceptron) folosing GridSearchCV si alternand urmatorii parametri: