Anexa 3

Universitatea “Politehnica” din Bucureşti

Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei

***Prelucrarea și analiza datelor de pe interfețele de acces ale unei rețele de comunicații mobile***

**Proiect de diplomă**

prezentat ca cerinţă parţială pentru obţinerea titlului de

*Inginer* în domeniul *Electronică și Telecomunicații*

programul de studii de licenţă*Tehnologii și Sisteme de Telecomunicații*

Conducător(i) ştiinţific(i) Absolvent

*Conf. Dr. Ing. Alexandru VULPE Denisa GHEORGHE*

*Ing. Mihai IDU*

*2019*

###### Cuprins

[Capitolul 1 Descrierea sistemului LTE 3](#_Toc12805901)

[1.1 Rețeaua de comunicații 3](#_Toc12805902)

[1.2 Evoluția comunicațiilor mobile 4](#_Toc12805903)

[1.3 Protocoale utilizate 6](#_Toc12805904)

[1.3.1 Code Division Multiple Access (CDMA) 6](#_Toc12805905)

[1.3.2 Frequency Division Multiple Access (FDMA) 6](#_Toc12805906)

[1.3.3 Time Division Multiple Access (TDMA) 7](#_Toc12805907)

[1.4 Sistemul LTE 8](#_Toc12805908)

[1.4.1 Arhitectura rețelei LTE 8](#_Toc12805909)

[1.4.1.1 Interfața S1-MME 10](#_Toc12805910)

[1.4.1.2 Interfața S1-U 11](#_Toc12805911)

[Capitolul 2 Date și metode de analiză 13](#_Toc12805912)

[2.1 Date 13](#_Toc12805913)

[2.1.1 Noțiuni introductive 13](#_Toc12805914)

[2.1.2 Parametri de performanță din arhitectura LTE 14](#_Toc12805915)

[2.1.2.1 Parametri de performanță pentru rețea 14](#_Toc12805916)

[2.1.2.2 Parametri de performanță pentru utilizator 16](#_Toc12805917)

[2.2 Metode de analiză 18](#_Toc12805918)

[2.2.1 Metoda pădurilor aleatoare (Random Forests) 18](#_Toc12805919)

[2.2.1.1 Cum funcționează algoritmul ? 19](#_Toc12805920)

[2.2.1.2 Pași logici de parcurgere 19](#_Toc12805921)

[2.2.2 Metoda Creșterii Variației (Gradient Boosting) 20](#_Toc12805922)

[2.2.2.1 Descriere pentru ansamblu, colectare(bagging) și creștere(boosting) 20](#_Toc12805923)

[2.2.3 Algoritmul Regresiei Logistice (Logistic Regression) 22](#_Toc12805924)

[2.3 Limbajele de programare 24](#_Toc12805925)

[2.3.1 SQL 24](#_Toc12805926)

[2.3.1.1 Limbajul de descriere a datelor – DDL 25](#_Toc12805927)

[2.3.2 Limbaje de programare utilizate în știința datelor 26](#_Toc12805928)

[Capitolul 3 Implementare Software 29](#_Toc12805929)

[3.1 Resurse utilizate 29](#_Toc12805930)

[3.1.1 MySQL Workbench 29](#_Toc12805931)

[3.1.2 Spyder (Anaconda) 32](#_Toc12805932)

[3.1.3 Qlik Sense (Desktop) 34](#_Toc12805933)

[3.2 Aspecte practice 36](#_Toc12805934)

[3.2.1 Extragerea datelor 36](#_Toc12805935)

[3.2.1.1 Crearea bazei de date 36](#_Toc12805936)

[3.2.1.2 Extragerea datelor în fișier Excel 39](#_Toc12805937)

[3.2.2 Algoritmii de predicție 40](#_Toc12805938)

[3.2.3 Vizualizarea rezultatelor 48](#_Toc12805939)

[Capitolul 4 Soluții de îmbunătățire a performanței 53](#_Toc12805940)

Concluzii ….………………………………………………………………………………………...57

Bibliografie ….……………………………………………………………………………………...59

ANEXA 1 ...………………………………………………………………………………………...61

ANEXA 2 .…………………………………………………………………………………………65

ANEXA 3 …………………………………………………………………………………………..67

ANEXA 4 …………………………………………………………………………………………..77

###### Lista figurilor

Figura 1.1 Împărțirea zonei geografice în celule 3

Figura 1.2 Reutilizarea frecvențelor 4

Figura 1.3 Alocarea frecvențelor pentru rețelele celulare 5

Figura 1.4 Împărțirea benzii de frecvență 6

Figura 1.5 Alocarea timeslot-urilor 7

Figura 1.6 Multiplexarea în LTE 7

Figura 1.7 Arhitectura rețelei LTE 8

Figura 1.8 Stiva protocoalelor S1-MME 11

Figura 1.9 Stiva protocoalelor S1-U 11

Figura 2.1 Alocarea puterii pe calea descendentă în LTE 14

Figura 2.2 Parametri de control ai PUSCH difuzați de eNodeB către UE 15

Figura 2.3 Structura IMSI 17

Figura 2.4 Schema logică de parcurgere a algoritmului Random Forests 20

Figura 2.5 Definirea ansamblului 21

Figura 2.6 Funcția sigmoid 22

Figura 2.7 Tipuri de comenzi în SQL 25

Figura 3.1 Schema generală a implementării 29

Figura 3.2 Fila de pornire a MySQL Workbench 31

Figura 3.3 Fila de editare a MySQL Workbench 31

Figura 3.4 Fila de start Anaconda Navigator 33

Figura 3.5 Fila de activitate Spyder 4 33

Figura 3.6 Nodul central Qlik Sense 34

Figura 3.7 Vizualizare din interiorul aplicației 35

Figura 3.8 Vizualizare din zona de dezvoltare 35

Figura 3.9 Detalii de autentificare 36

Figura 3.10 Inserarea înregistrărilor 37

Figura 3.11 Descrierea tabelului 38

Figura 3.12 Algoritm de extragere a datelor 39

Figura 3.13 Localizarea fișierului de tip xlsx 40

Figura 3.14 Diagrama software 41

Figura 3.15 Matricea de corelație a câmpurilor din stats\_db 42

Figura 3.16 Rezultat al preprocesării datelor 43

Figura 3.17 Curba ROC - Pădurile aleatoare 47

Figura 3.18 Curba ROC – Creșterea variației 48

Figura 3.19 Curba ROC – Regresia logistică 48

Figura 3.20 Rezultatele votului afișate în Qlik Sense 49

Figura 3.21 Rezultatele filtrate per index 49

Figura 3.22 Predicții pentru datele de învățare 50

Figura 3.23 Predicțiile pentru datele de test 51

Figura 4.1 Moduri de duplexare în LTE 53

Figura 4.2 Schemele de modulație din LTE 55

###### Lista tabelelor

Tabelul 1 Evoluția rețelelor celulare comerciale 4

Tabelul 2 Măsurarea frecvențelor 16

Tabelul 3 Limbaje de programare utilizate în știința datelor 27

Tabelul 4 Vedere din stats\_db 38

Tabelul 5 Rezultatele votului IV 44

Tabelul 6 Rezultatele votului Random Forests 44

Tabelul 7 Rezultatul votului ExtraTrees 44

Tabelul 8 Rezultatul votului Chi Square 44

Tabelul 9 Rezultatele tuturor voturilor 45

Tabelul 10 Scorul final 45

Tabelul 11 Predicții pe datele de învățare 46

Tabelul 12 Predicții pe datele de test 46

Tabelul 13 Predicții pe datele de învățare 47

Tabelul 14 Predicții pe datele de test 47

Tabelul 15 Predicții pe datele de învățare 47

Tabelul 16 Predicții pe datele de test 47

Tabelul 17 Configurarea TDD 54

###### Lista acronimelor

|  |  |  |
| --- | --- | --- |
| 3GPP | 3rd Generation Parternship Project |  |
| AI | Artificial Intelligence | Inteligența artificială |
| BTS | Base Tranceiver Station | Stația de bază |
| CDMA | Code Division Multiple Access | Acces multiplu prin divizarea codurilor |
| CPU | Central Processing Unit | Unitatea centrală de procesare |
| DB | Database | Bază de date |
| DL | Downlink | Cale descendentă |
| FDMA | Frequency Division Multiple Access | Acces multiplu prin divizarea frecvenței |
| GSM | Global System Mobile Communications | Sistemul global de comunicații mobile |
| GUI | Graphical User Interface | Interfața grafică |
| HD | High Definition | Calitate înaltă |
| HLR | Home Location Register | Registrul de localizare |
| IP | Internet Protocol | Protocol de internet |
| JDBC | Java Database Conectivity | Conectivitatea la baza de date utilizând Java |
| LAN | Local Area Network | Rețea locală |
| LTE | Long Term Evolution | Evoluția pe termen lung |
| MCC | Mobile Country Code | Codul țării pentru rețeaua mobilă |
| MNC | Mobile Network Code | Codul rețelei mobile |
| MSIN | Mobile Station Identification Number | Numărul de identificare al stației mobile |
| ODBC | Open Database Conectivity | Conectivitatea liberă la baza de date |
| OFDMA | Orthogonal Frequency Division Multiple Access | Acces multiplu prin divizarea ortogonală a frecvenței |
| PDSCH | Physical Downlink Shared Channel | Canalul fizic partajat pe calea descendentă |
| PUSCH | Physical Uplink Shared Channel | Canalul fizic partajat pe calea ascendentă |
| QAM | Quadrature Amplitude Modulation |  |
| QoS | Quality of Services | Calitatea serviciilor |
| QPSK | Quadrature Phase Shift Keying |  |
| RB | Resource blocks | Blocuri de resurse |
| ROC | Receiver Operating Characteristic | Caracteristica de funcționare a receptorului |
| SGBD | Relational database management system | Sistemul de gestiune a bazelor de date |
| TDMA | Time Division Multiple Access | Acces multiplu prin divizarea timpului |
| UE | User Equipment | Echipamentul utilizatorului |
| UL | Uplink | Cale ascendentă |
| UMTS | Universal Mobile Telecommunications System | Sistemul universal de telecomunicații mobile |
| WAN | Wide Area Network | Rețea extinsă |

###### Introducere

Am ales tema „Prelucrarea și analiza datelor de pe interfețele de acces ale unei rețele de comunicații mobile” deoarece consider că este unul dintre subiectele de actualitate din domeniul telecomunicațiilor. Din punctul meu de vedere este important să cunoaștem parametri care au cel mai mare impact în funcționarea rețelei și ca aceștia să primească atenție sporită în analiza și evaluarea performanței rețelei. Comportamentul predictiv oferă avantajul că, în cazul unor degradări, se poate interveni exact pe echipamentele răspunzătoare și eficiența este una mult mai mare decât atunci când soluțiile de îmbunătățire sunt aduse ulterior evenimentelor neplăcute.

Un alt motiv pentru care am ales această temă a fost că în timpul perioadei practicii de vară, efectuată în cadrul Orange România, am acumulat cunoștințe despre tehnologia 4G pe care am vrut să le dezvolt, realizând un studiu de utilitate practică.

Proiectul are ca obiective trasarea unui comportament predictiv pentru anumiți indicatori de performanță ai tehnologiei 4G(zona de date) și prezentarea unor soluții de optimizare a comportamentelor nedorite. În acest sens, se va utiliza extragerea valorilor dintr-o bază de date, prelucrarea lor cu ajutorul modelului predictiv construit în limbajul de programare Python, iar rezultatele vor fi vizualizate în utilitarul Qlik Sense.

În scopul realizării celor menționate mai sus, lucrarea este împărțită în patru capitole, după cum urmează : capitolul 1 conține informații generale despre arhitectura LTE, capitolul 2 conține informații despre datele care urmează să fie analizate și despre metodele prin care va fi realizat acest lucru, capitolul 3 conține prezentarea programelor în care a avut loc dezvoltarea și detalierea acesteia din urmă, iar în ultimul capitol, cu numărul 4 voi oferi soluțiile de îmbunătățire a performanței, ceea ce reprezintă obiectivul lucrării.

# Descrierea sistemului LTE

## Rețeaua de comunicații

O rețea mobilă poate fi definită ca o rețea de comunicații care este răspândită pe o suprafață imensă a teritoriului din întreaga lume, conectată fără fir(wireless) de către transceivere la locații fixe care sunt cunoscute ca site-uri de celule sau stații de bază(BTS).

Wireless înseamnă a avea acces la o rețea locală (LAN), o rețea extinsă(WAN) sau o rețea celulară 4G/3G.[1]

O rețea celulară este o rețea radio distribuită pe pământ prin intermediul celulelor, în care fiecare celulă include un transmițător cu locație fixă și putere limitată cunoscut sub denumirea de stație de bază. Aceste celule împreună oferă acoperire radio pe zone geografice mai mari.

Mărimea unei celule poate varia în funcție de numărul de utilizatori care trebuie deserviți într-o anumită zonă și de traficul pe utilizator. Dacă există mult trafic într-o zonă, de exemplu zonă urbană, dimensiunea celulei va fi mai mică decât în ​​zonele rurale.

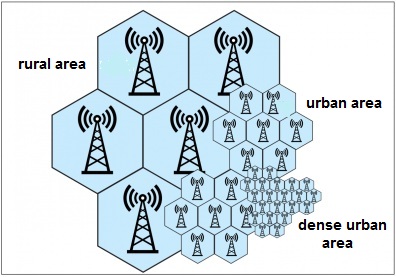


Figura 1.1 Împărțirea zonei geografice în celule [2]

Puterea limitată a stației de bază face posibilă reutilizarea aceleiași frecvențe la câteva celule distanță de stația de bază fără a provoca interferențe, așa cum se vede în Figura 1.2. Astfel, echipamentul utilizatorului (UE), cum ar fi telefonul mobil, poate comunica chiar dacă se deplasează prin celule în timpul transmisiei.

O rețea celulară este o modalitate foarte eficientă de utilizare a resurselor cu număr mic de frecvențe.

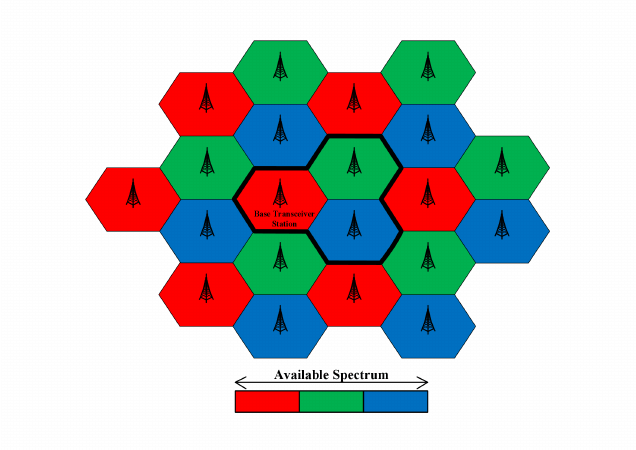


Figura 1.2 Reutilizarea frecvențelor [3]

Spectrul frecvențelor este divizat în acest caz în 3 seturi de frecvențe. Fiecare set este utilizat într-o altă celulă. Grupul de 3 celule se repetă pentru a acoperi complet zona geografică.

Când utilizatorul se mută dintr-o celulă în alta, apelul va fi transferat de la un BTS la altul. Acest proces poartă numele de handover.

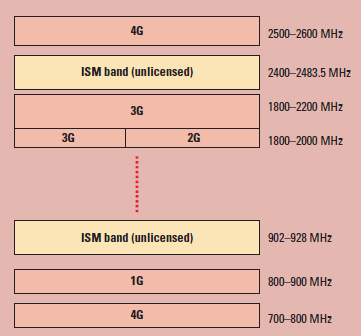
Rețeaua păstreză locația echipamentului UE pentru a direcționa un apel de intrare către celula corespunzătoare. Așadar, o rețea celulară este echipată cu registru de localizare (HLR).

## Evoluția comunicațiilor mobile

Tabelul 1 arată cum au evoluat rețelele celulare de la 1G la 4G, evidențiind caracteristicile cheie pentru fiecare generație, calitatea serviciilor (QoS) și protocoalele de acces. Toate rețelele celulare utilizează un spectru de frecvențe licențiat. Rețelele 1G operează pe o bandă de frecvențe între 800 și 900 MHz, în timp ce 2G operează între 1800 și 2000 MHz, iar rețelele 3G între 1800 și 2200 MHz.

Tabelul 1 Evoluția rețelelor celulare comerciale [4]

|  |  |  |  |
| --- | --- | --- | --- |
| Generație/an | Caracteristici cheie și capabilități de acces | Fiabilitate, QoS și performanță | Protocoale utilizate |
| 1G/1981 | - utilizează semnale analogice, mai ales pentru comunicațiile de voce  - suportă roaming | - afectat de acoperirea limitată  - QoS nedezvoltat  - performanța nu a atins așteptările | - Frequency Division Multiple Access (FDMA), unde fiecare utilizator are asignat câte un canal pe durata apelului |
| 2G/1991 | - utilizează tehnologia digitală pentru securitatea vocii și a datelor  - suportă roaming  - oferă acces la rețelele 1G | - fiabilitate îmbunătățită prin back-up la rețeaua 1G  - QoS mai bun  - performanța nu a atins așteptările | - Time Division Multiple Access(TDMA), unde fiecare utilizator are asignată o durată de timp dintr-un canal sau Code Division Multiple Access (CDMA), unde fiecărui utilizator îi este asignat un cod pentru durata apelului |
| 3G/2001 | - suportă conținut multimedia  - suportă roaming global printr-un singur tip de rețea wireless (cum ar fi rețeaua celulară), la viteze de la 384 Kbps până la câțiva Mbps  - oferă acces la rețeaua 2G | - fiabilitate îmbunătățită prin back-up la rețelele 2G și uneori 1G  - QoS îmbunătățit  - performanța atinge așteptările | TDMA și multiple variații ale CDMA |
| 4G/2011 | - suportă roaming global prin multiple rețele wireless cu lățime mare a benzii(50 Mbps sau chiar mai mult)  - oferă acces către rețelele 2G și 3G | - fiabilitate îmbunătățită prin accesul la rețelele 3G și uneori 2G  - QoS îmbunătățit considerabil  - performanța atinge oarecum așteptările | Orthogonal FDMA cu multiple antene |

Toate rețelele celulare utilizează un spectru de frecvențe licențiat. Rețelele 1G operează pe o bandă de frecvențe între 800 și 900 MHz, în timp ce 2G operează între 1800 și 2000 MHz, iar rețelele 3G între 1800 și 2200 MHz.

Rețeaua actuală de 4G are la bază o combinație a frecvențelor 800, 1800 și 2600 MHz.

Datorită nevoii de a transmite cât mai multe date, purtătoarele wireless 4G încearcă să achiziționeze sau să reutilizeze multe dintre benzile licențiate. [4]

Figura 1.3 Alocarea frecvențelor pentru rețelele celulare [4]

## Protocoale utilizate

Canalul radio este un mediu de comunicare partajat de mai mulți utilizatori într-o zonă geografică. Stațiile mobile sunt în concurență una cu alta pentru ca resursa de frecvență să transmită fluxul de informații. Fără alte măsuri pentru a controla accesul concomitent al mai multor utilizatori, pot apărea coliziuni. De exemplu, identificarea utilizatorului poate fi denumită ca “acces multiplu” deoarece stația de bază recepționează simultan un număr de unde radio egal cu numărul de stații care transmit (UE).

### Code Division Multiple Access (CDMA)

CDMA este o tehnologie celulară digitală utilizată pentru comunicații mobile. Sistemele celulare CDMA sunt considerate superioare celor FDMA și TDMA, motiv pentru care CDMA joacă un rol esențial în construirea sistemelor de comunicații radio eficiente, robuste și sigure.

Caracteristici esențiale :

* fiecare canal utilizează tot spectrul disponibil
* conversațiile individuale sunt codate cu o secvență digitală
* asigură capacitate mai bună pentru comunicațiile de voce și date, permițând mai multor abonați să se conecteze la orice moment de timp
* este platforma pe care s-au construit tehnologiile 3G

### Frequency Division Multiple Access (FDMA)

FDMA este una dintre cele mai comune metode analogice de acces multiplu. Banda de frecvență este împărțită în canale cu lățime de bandă egală, astfel încât fiecare conversație să fie difuzată pe o frecvență diferită (vezi Figura 1.4).

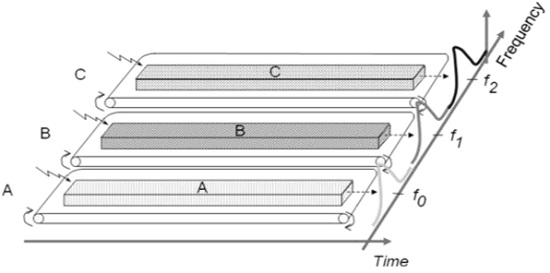


Figura 1.4 Împărțirea benzii de frecvență [5]

Avantajele FDMA :

* micșorează interferența intersimbol
* este ușor de implementat
* este necesar un număr mic de biți pentru sincronizare (transmisia este continuă)

Dezavantajele FDMA :

* debitul maxim per canal este fix și de valoare mică
* nu diferă semnificativ de sistemele analogice

### Time Division Multiple Access (TDMA)

TDMA este o tehnologia digitală complexă a comunicațiilor mobile celulare care permite mai multor utilizatori să partajeze aceeași frecvență fără interferențe. Ea împarte un semnal în diferite perioade de timp și crește capacitatea de transmisie a informațiilor.

În următorul exemplu (Figura 1.5) aceeași frecvență este folosită de 3 utilizatori. Fiecărui utilizator îi este atribuit un interval de timp (timeslot) pentru a trimite și recepționa informații. Utilizatorul B transmite după A, iar C transmite după B.

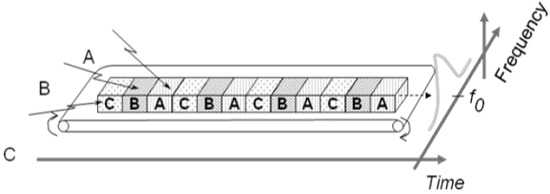


Figura 1.5 Alocarea timeslot-urilor [5]

Avantajele TDMA :

* permite debit flexibil
* numărul de timeslot-uri alocate unui utilizator poate fi modificat la fiecare transmisie

Dezavantajele TDMA :

* este necesar un număr mare de biți pentru sincronizare
* funcționarea la rate mari de bit crește consumul de energie

În LTE accesul multiplu se face prin mai multe subpurtătoare, iar metodele de multiplexare în DL sunt diferite fată de UL.

Pentru DL este folosită multiplexarea OFDMA (Orthogonal Frequency Division Multiple Access), prin care se alocă mai mulți utilizatori atât in domeniul timp, cât și în domneniul frecvență, conform Figura 1.6. Pentru UL este folosită multiplexarea SC-FDMA (Single Carrier Frequency Division Multiple Access), prin care se face alocarea utilizatorilor doar in domeniul timp, conform aceleiași figuri.

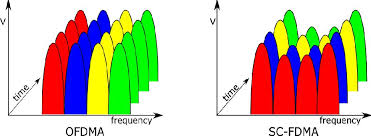


Figura 1.6 Multiplexarea în LTE [6]

## Sistemul LTE

Tehnologia *LTE (Long Term Evolution)* este o tehnologie wireless 4G de bandă largă dezvoltată de 3GPP, un grup de comerț din industrie. Tehnologia LTE a permis conectarea rapidă la internet mobil. De fapt, LTE este o cale urmată pentru a atinge viteze 4G. LTE este o tehnologie IP completă utilizată pentru serviciile mobile de bandă largă pentru transferul de date și apelurile vocale.

Operatorii wireless și-au extins rapid rețelele LTE pentru a beneficia de o eficiență suplimentară, o latență mai mică și capacitatea de a gestiona tot mai mult traficul de date. Între timp, accesul a evoluat de la TDMA (Time Division Multiple Access) la OFDMA (Orthogonal Frequency Division Multiple Access), deoarece nevoia de viteze și volume de date mai mari a crescut.

Această tehnologie a fost creată pentru a suporta viteze de până la 100Mb/s pe calea descendentă (DL) și 50 Mb/s pe calea ascendentă (UL). Conform Figura 1.7 întelegem prin cale ascendentă calea de la utilizator la stația de bază, iar calea descendentă de la stația de bază la utilizator.

LTE față de tehnologiile anterioare, sistemul universal de telecomunicatii mobile UMTS (Universal Mobile Telecommunications System) și sistemul global de comunicații mobile GSM (Global System Mobile Communications), are o arhitectură plată, mai simplă și face trecerea către o rețea de comunicații bazată în întregime pe un seviciu de transmitere a datelor cu adrese IP (Internet Protocol).

Arhitectura rețelei LTE se împarte în două subrețele.

### Arhitectura rețelei LTE



Figura 1.7 Arhitectura rețelei LTE [7]

Arhitectura rețelei 4G (Figura 1.7) se compune din [8] :

* E-UTRAN (Evolved UMTS Terrestrial Radio Access Network) - rețeaua terestră avansată UMTS de acces radio care se ocupă de comunicația dintre echipamentul utilizatorului si EPC (Evolved Packet Core).
  + UE (User Equipment) - echipamentul utilizatorului alcătuit din :
    - terminalul mobil, MT (Mobile Termination) - gestionează toate funcțiile de comunicare
    - echipamentul terminal, TE (Terminal Equipment) - controlează fluxurile de date
    - cartela SIM pentru echipamente LTE, UICC (Universal Integrated Circuit Card) - rulează o aplicație denumită USIM (Universal Subscriber Identity Module), reprezentând modulul universal de identificare al abonatului.

USIM este responsabil de autentificarea utilizatorului în rețea și, de asemenea, permite autentificarea rețelei la utilizator. USIM utilizează cea mai recentă tehnologie de criptare în scopul autentficării ce are ca rezultat protecția utilizatorilor și rețelelor împotriva atacurilor de securitate mobile care sunt din ce în ce mai puternice. USIM menține o bază de date pe post de agendă telefonică, care poate stoca mii de intrări unde fiecare contact poate avea adrese de e-mail, un al doilea sau al treilea numar de telefon, etc.

* + eNodeB – stația de bază a sistemului LTE. Fiecare eNodeB controlează mobilele din una sau mai multe celule. În această tehnologie, un mobil este conectat la o singură stație de bază, funcțiile cele mai importante ale acesteia fiind:
    - managementul resurselor radio
    - managementul mobilității conexiunii
    - planificarea resurselor între UE și eNodeB
    - raportarea măsurătorilor ce ajută la luarea deciziilor de transfer
  + interfețe de tip Uu (microunde) – realizează conexiunea dintre UE și eNodeB. Sunt semnale wireless care formează celulele mobile.
  + interfețele X2 – realizează conexiunea între eNodeB-uri adiacente. Acestea oferă un nivel mult mai mare de interconectivitate direct, permițând ca multe apeluri să fie rutate direct, în măsura în care un număr mare de convorbiri și conectări sunt îndreptate către celelalte stații mobile din aceleași celule sau dintr-unele adiacente.
* EPC (Envolved Packet Core) – parte a rețelei responsabilă pentru controlul global al UE și stabilirea purtătoarelor. Este compusă din cinci noduri principale :
  + PCRF – Policy Control and Charging Rules Function este responsabil pentru controlul politicii de luare a deciziilor, precum și pentru controlul fluxului bazate pe politica de tarifare în Policy Control Enforcement Function (PCEF) ce se află în P-GW(Packet Data Network Gateway). PCRF atribuie QoS-ul(clasa de identificare și ratele de bit), care decide modul în care un anumit flux de date va fi tratat în PCEF și asigură că acest lucru este în conformitate cu profilul abonamentului utilizatorului.
  + P-GW (sau PDN Gateway - Packet Data Network Gateway) este responsabil pentru alocarea adresei IP pentru UE, precum și punerea în aplicare a QoS-ului în conformitate cu regulile date de PCRF. Este responsabil cu filtrarea pachetelor IP de downlink ale utilizatorului în funcție de diferitele QoS-uri ale purtătoarelor. P-GW efectuează aplicarea QoS-ului pentru a se asigura rata de bit garantată pentru fiecare purtătoare radio. De asemenea, servește ca punct comun pentru a asigura mobilitate pentru interconectarea cu tehnologii non-3GPP, cum ar fi CDMA2000 și rețele WiMAX.
  + S-GW (Serving Gateway) – toate pachetele IP ale tuturor utilizatorilor sunt transferate prin intermediul Serving Gateway-ului care servește ca punct comun pentru mobilitatea locală a purtătoarelor de date atunci când UE se mișcă între diferite eNodeB-uri. De asemenea, reține informații despre purtătoare atunci când UE este în starea idle și memorează temporar în buffer datele de downlink în timp ce MME – ul inițiază paging-ul către UE pentru a restabili purtătoarea. În plus, S-GW efectuează unele funcții administrative în rețeaua vizitată, cum ar fi colectarea de informații pentru încărcare (de exemplu, volumul de date trimise sau primite de la utilizator) și de interceptare legală. De asemenea, servește ca punct comun pentru mobilitatea interconectării cu alte tehnologii 3GPP: GPRS și UMTS.
  + MME – Mobility Management Entity este nodul de control care prelucrează semnalizarea între UE și Core Network. MME administrează funcțiile de control ale mobilității (autentificarea și securitatea) și de interacțiune cu stțiile mobile în stare de așteptare (actualizarea locației și pagingul), funcții îndeplinite de VLR (Visitor Location Register), respective GSM (GPRS Mobility Management), în rețele GSM/GPRS/UMTS. De asemenea, MME controlează purtătoarele alocate la nodurile din nucleul rețelei și semnalizarea în rețea. Funcțiile principale ale MME-ului pot fi clasificate astfel:
    - gestionarea purtătoarei – aceasta include stabilire, menținerea și eliberarea de purtătoare.
    - gestionarea conexiunii – aceasta include crearea conexiunii și securitate între rețea și UE.
* HSS – Home Subscriber Server conține datele utilizatorilor, printer care și profilele QoS la care au acces, precum și orice restricții de acces la serviciile de roaming. Acesta deține, de asemenea, informații despre PDN-urile la care utilizatorul se poate conecta. În plus, HSS deține informații dinamice cum ar fi identitatea MME la care utilizatorul este în present atașat sau înregistrat. HSS poate să integreze, de asemenea, centrul de autentificare (AUC), care generează vectorii de autentificare și chei de securitate.
* Interfețele S1 - Interfaţa dintre staţia de bază şi nucleul reţelei se numeşte interfaţa S1. De obicei, aceasta presupune o legătura de fibră optică sau cablu de cupru de mare viteză. Această interfaţă este împărţită în două părţi logice, care transportă informaţia prin acelaşi canal fizic. Acestea sunt S1 User Plane (S1-U), pentru datele utilizatorilor şi S1 Control Plane (S1-MME), pentru datele de control ale reţelei LTE.

#### Interfața S1-MME

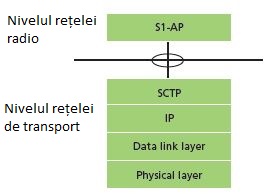
Interfața S1-MME constă dintr-un protocol de transmitere a fluxului de control (Stream Control Transmission Protocol – SCTP) prin IP și suportă mai multe UE într-o singură asociere SCTP. Protocolul de semnalizare a aplicației este S1-AP (Application Protocol).

S1 Control Plane este responsabil pentru:

* interacţiunea eNodeB-ului cu nucleul reţelei pentru comunicări specifice
* transferul mesajelor de semnalizare care ţin de utilizatori, de exemplu pentru un apel de voce.
* procedurile de configurare/deblocare a purtătorului EPS (Evolved Packet System – Sistemul de pachete evoluat)
* procedura de paging

Figura 1.8 prezintă structura protocalelor interfeței S1 Control Plane.

Inițializarea interfeței S1-MME începe cu identificarea MME-urilor la care trebuie să se conecteze eNodeB-ul, urmată de configurarea nivelului rețelei de transport (Transport Network Layer - TNL).

Protocolul SCTP este bine cunoscut pentru funcțiile avansate moștenite de la TCP, iar în plus, este posibil să beneficieze de caracteristici îmbunătățite, cum ar fi manipularea fluxurilor multiple, pentru a pune în aplicare cu ușurință redundanța rețelei de transport și pentru a evita blocarea capătului de linie.

O zonă de simplificare în LTE, față de 3G, este maparea directă a S1-AP la SCTP. Acest lucru are ca rezultat o stivă de protocoale simplificată, fără un protocol intermediar de gestionare a conexiunilor,

Figura 1.8 Stiva protocoalelor S1-MME [9] deoarece conexiunile individuale sunt tratate direct la

nivelul aplicației.

Multiplexarea are loc între S1-AP și SCTP, prin care fiecare flux al unei asociații SCTP este multiplexat cu traficul de semnalizare al mai multor conexiuni individuale.

LTE a construit, de asemenea, flexibilitate în protocoalele de nivel inferior, oferind operatorului opțiunea completă în ceea ce privește alegerea versiunii IP și a nivelului legăturii de date. De exemplu, acest lucru permite operatorului să înceapă implementarea utilizând versiunea IP 4 cu legătura de date adaptată scenariului de implementare a rețelei.

#### Interfața S1-U

Interfața S1-U este definită între stația de bază a rețelei LTE, eNodeB, și S-GW.

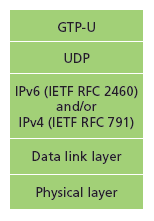


Figura 1.9 prezintă structura protocoalelor intefeței S1 User Plane, bazată pe GTP/UDP5/stiva IP, cunoscută de la rețelele 3G.

Unul dintre avantajele utilizării GPRS Tunneling Protocol – User plane (GTP-U) este facilitarea mobilității intra-3GPP.

Numărul versiunii IP și nivelul legăturii de date au fost lăsate pe deplin opționale, ca și in cazul stivei interfeței S1 – Control Plane.

S-GW trimite pachete downlink ale unui purtător dat la adresa IP a eNodeB-ului (primită în S1-AP) asociată cu acel purtător particular. În mod similar, eNodeB-ul trimite pachete uplink ale unui purtător dat la adresa IP a EPC-ului (primită în S1-AP) asociată cu acel purtător particular.

Figura 1.9 Stiva protocoalelor S1-U [9]

# Date și metode de analiză

## Date

Parametri pe care îi voi evalua pe parcursul acestei lucrări, și anume parametri de performanță din arhitectura LTE sunt stocați într-o bază de date la care mă voi conecta pentru a extrage valorile acestora.

### Noțiuni introductive

Bazele de date reprezintă nucleul sistemelor informatice din orice companie sau instituţie, având un impact major asupra modului de funcţionare şi organizare al acestora. Totodată oferă o deschidere majoră asupra pieţii pe care o vizează, oferind posibilitatea clienţilor de a avea acces în mod facil la datele esenţiale de care aceştia au nevoie.

Trebuie avut în vedere faptul că de multe ori, în viața de zi cu zi prin abuz de limbaj, se folosește termenul de Bază de Date (BD) pentru a desemna de fapt un Sistem de Gestiune a Bazelor de Date (SGBD).

Putem defini într-o primă etapă o Bază de Date (BD) ca fiind un ansamblu de date structurat, stocat în mod centralizat sau nu, pe servere, accesibil, interogabil şi modificabil de un grup de utilizatori care lucrează în paralel, prin intermediul uneia sau a mai multor aplicaţii. Pe de altă parte, un Sistem de Gestiune a Bazelor de Date (SGBD) poate fi văzut generic ca un sistem care se ocupă de structurarea, stocarea, actualizarea şi mentenanţa datelor, reprezentând de fapt interfaţa între baza de date și utilizator sau aplicațiile acestuia.

În aceasta lucrare vom lucra cu baze de date relaționale, iar ca sistem de gestiune a bazei de date se va utiliza programul MySQL.

Bazele de date relaţionale au la bază modelul relaţional care lucrează cu două concepte importante: relaţie şi tabel, Cele două diferă prin natura lor, dar sunt foarte corelate. Noţiunea de relaţie este formală, deoarece conceptul provine din matematică, în particular din teoria mulţimilor, în timp ce noţiunea de table este simplă şi intuitivă. Pe de o parte, tabelele oferă o înţelegere naturală a structurii bazei de date chiar şi pentru utilizatorii ce nu sunt specializaţi în domeniu. Pe de altă parte, existența unei formalizări matematice clare și simple a permis dezvoltarea uneiteorii care să sprijine modelul, cu rezultate foarte interesante în industrie [10].

Atunci când o relaţie este concepută sub forma unui tablou de valori, fiecare linie din tabel reprezintă un ansamblu de valori corelate. În cazul modelului relaţional, fiecare linie din tabel corespunde de fapt unei entităţi sau unei relaţii din lumea reală. În cazul terminologiei modelului relaţional, o linie din tabel poartă denumirea de tuplu, înregistrările stocate în coloanele tabelului sunt denumite atribute, în timp ce tabelele poartă numele de relaţii. Tipurile de date introduse în fiecare coloană sunt reprezentate de către un domeniu de valori posibile.

Orice relaţie poate fi definită ca un ansamblu de tupluri. Elementele acestui ansamblu nu sunt ordonate, adică într-o relaţie, tuplurile nu se supun nici unei reguli particulare putând fi ordonate în mod arbitrar.

### Parametri de performanță din arhitectura LTE

Un parametru de performanță este o valoare măsurabilă este utilizată pentru a evalua succesul unui proiect în atingerea obiectivelor pentru care a fost creat. Parametri de performanță din arhitectura LTE sunt analizați și din punctul de vedere al satisfacerii utilizatorului, dar și din punctul de vedere al configurării echipamentelor de rețea. De aceea, în continuare voi oferi câteva exemple de indicatori, grupați după zona pe care o evaluează (utilizator/rețea).

#### Parametri de performanță pentru rețea

* Puterea de transmisie a canalului fizic PDSCH pe traiectul descendent - Physical Downlink Shared Channel(PDSCH) Power:
  + În LTE, PDSCH este singurul canal de transport pe calea descendentă, disponibil pentru a transporta date despre utilizator între dispozitivul mobil și eNodeB.
  + Alocarea puterii downlink poate varia de la celulă la celulă și în plus poate fi specifică dispozitivului. Aceste setări vor avea un impact asupra performanței unui dispozitiv compatibil LTE. Iar transferul de date este, bineînțeles, un criteriu de performanță pe care nu îl judecă numai operatorii de rețea, ci afectează și experiența utilizatorului.

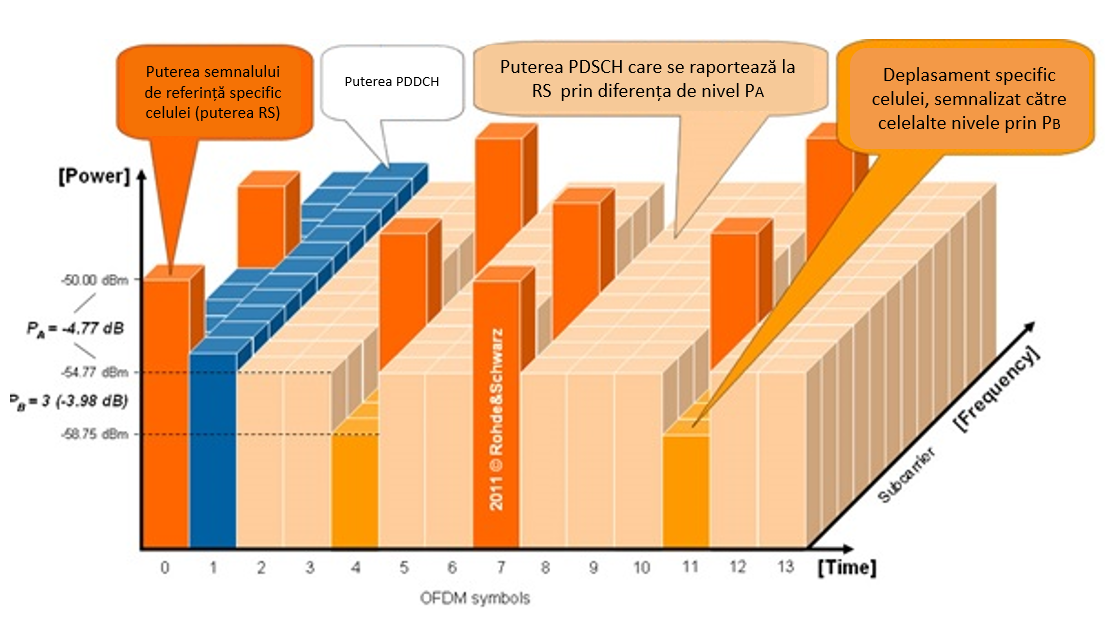


Figura 2.1 Alocarea puterii pe calea descendentă în LTE [11]

* + Puterea PDSCH depinde întotdeauna de alocare, adică de numărul de blocuri de resurse (RB) alocate. Alocarea se poate schimba de la cadru la cadru, astfel încât PA poate de asemenea să se schimbe pe o bază de 1 milisecundă. În timp ce încorporează PA și PB, se asigură că puterea totală a simbolului OFDM rămâne constantă chiar și atunci când alocarea PDSCH este schimbată. [11]
  + Scheme de modulatie - QPSK, 16QAM, 64QAM
  + În blocurile de resurse în care semnalele de referință specifice UE nu sunt transmise, PDSCH se transmite pe următoarele seturi de porturi : {0}, {0,1},{0,1,2,3}.
  + În blocurile de resurse în care sunt transmise semnalele de referință specifice UE, PDSCH se transmite pe portul de antenă {5}
* Puterea de transmisie a canalului fizic PDSCH pe traiectul ascendent: Physical Uplink Shared Channel(PUSCH) Power:
  + Și acest canal conține date informative despre utilizator
  + Transportă atât date de utilizator, cât și date de control al semnalului (parametri legați de MIMO – Multiple Input Multiple Output și indicatori de format pentru transport)
  + Scheme de modulatie - QPSK, 16QAM, 64QAM. eNodeB-ul selectează tipul de modulație după un algoritm adaptiv. Dacă eNodeB-ul direcționează UE să utilizeze 64QAM, dar acesta nu suportă acest lucru, se selectează automat tipul de modulație 16QAM
  + Specificațiile 3GPP definesc puterea transmisă de UE pentru PUSCH după următoarea ecuație :

PPUSCH = min{ Pmax, 10·log10 M + P0 + α × PL + δmcs + f(Δi)} [dBm], unde

Pmax este puterea maximă transmisă care depinde de clasa de putere a UE;

M este numarul de blocuri de resurse

P0 este un parametru specific celulei

α este factorul de compensare a pierderilor și este specific celulei, fiind semnalizat de controlul resurselor radio(RRC)

PL sunt pierderile de pe calea descendentă și sunt calculate în UE

δmcs este un parametru al schemei de modulație și codare, specific celulei

f(Δi) este specific UE. Δi este valoarea de corecție a unei bucle închise, iar f este funcția care ne permite să utilizăm valoarea absolută a lui Δi. [12]

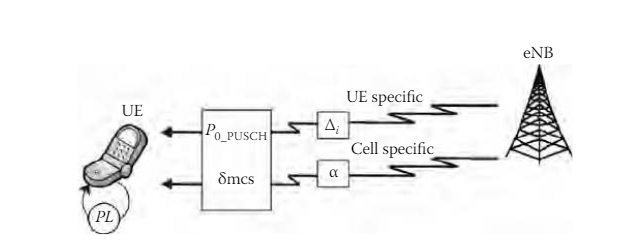


Figura 2.2 Parametri de control ai PUSCH difuzați de eNodeB către UE [12]

Parametrii specifici celulei arată că ei sunt aceeași pentru toate UE din acea celulă.

* Lărgimea de bandă în LTE sau Bandwidth
  + Lărgimile de bandă definite de standard sunt 1.4, 3, 5, 10, 15 și 20 MHz. De obicei, pentru căile ascendentă și descendentă se folosește o lărgime de bandă de 5 MHz. Tabelul 2 arată câte subpurtătoare și câte blocuri de resurse sunt în fiecare lărgime de bandă pentru calea ascendentă și descendentă.

Tabelul 2 Măsurarea frecvențelor

|  |  |  |  |
| --- | --- | --- | --- |
| Lărgimea de bandă | Blocuri de resurse | Subpurtătoare pentru calea ascendentă | Subpurtătoare pentru calea descendentă |
| 1.4 MHz | 6 | 73 | 72 |
| 3 MHz | 15 | 181 | 180 |
| 5 MHz | 25 | 301 | 300 |
| 10 MHz | 50 | 601 | 600 |
| 15 MHz | 75 | 901 | 900 |
| 20 MHz | 100 | 1201 | 1200 |

* Descărcarea per celulă - Cell Throughput:
  + În termeni simpli, este suma throughput-ului mediu pentru toți utilizatorii dintr-o rețea. Matematic, este throughput-ul mediu per utilizator în interiorul celulei înmulțit cu numărul de utilizatori din rețea.
  + Nu este neapărat adevărat ca dacă există un throughput mare pentru un utilizator dintr-o celulă să atragă după sine un throughput mediu mare per celulă, și invers. Acest lucru se poate naște datorită faptului că este posibil ca o celulă să aibă un throughput mare pentru celulă și unul foarte scăzut pentru utilizator deoarece o celulă poate avea unii utilizatori care se află în condiții excelente de acoperire, în timp ce alții se pot afla în condiții slabe de acoperire.
* Numărul cadrului de sistem - System Frame Number (SFN)
  + Este așa cum îi spune și numele un contor și indică restul împărțirii index-ului cadrului de sistem la 1024. Index-ul este cuprins în intervalul 0, 1023 și este format pe 10 biți.
  + SFN se incrementează cu 1 la fiecare 10ms.
  + SFN nu ajută la sincronizarea fizică (frecvența purtătoarei, etc.) pentru că este o informație obținută după sinconizarea cu forma de undă.
  + SFN ajută la sincronizarea nivelului fizic dintre canalele fizice ale căii ascendente și cele ale căii descendente. [13]

#### Parametri de performanță pentru utilizator

* IMSI-ul terminalului mobil (International Mobile Subscriber Identity)
  + este un dispozitiv de interceptare telefonică folosit pentru interceptarea traficului de telefonie celulară și pentru supravegherea mișcării utilizatorilor de aparate mobile.
  + În linii mari, el acționează ca o falsă celulă de telefonie mobilă, care se interpune între telefonul-țintă și adevăratul turn-antenă al companiei de telefonie.
  + Este de obicei un număr de 15 digiți, dar poate fi și mai scurt și este stocat într-un câmp de 64 de biți.
  + Este trimis de echipamentul mobil către rețea.
  + Este utilizat de orice rețea mobilă care interacționează cu alte rețele.
  + Conține codul țării (MCC), codul rețelei mobile (MNC) și numărul de identificare a stației mobile (MSIN) (vezi Figura 2.3)

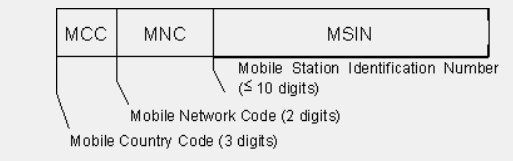


Figura 2.3 Structura IMSI [14]

* Numărul de antene ale terminalului mobil
  + În comunicațiile celulare, numărul maxim de antene pentru calea descendentă este 2 și 4 pentru a susține LTE-ul.
  + Conceptul de mai multe antene este o tehnică inteligentă care a depășit limitările tehnicii MIMO.
  + MIMO este un element esențial al comunicației fără fir în rețelele 4G care se referă la conceptul de a trimite și recepționa în același timp semnale de date multiple pe același canal radio, utilizând propagarea în mai multe căi. Prin aceasta se utilizează o tehnică de transmisie numită multiplexare spațială (SMX). Rangul este definit ca numărul de straturi dintr-o transmisie de multiplexare spațială LTE, adică este un indicator al funcționării antenelor multiple. Firește, antenele multiple funcționează bine dacă semnalul de la fiecare antenă nu are corelație sau interferență cu celelalte.
* Throughput-ul pentru utilizator
  + În termeni simpli, este cantitatea medie de date pe care le primește un utilizator conectat în rețeaua LTE.
  + Matematic, acesta poate fi definit ca numărul de pachete primite de un anumit utilizator (UE) într-un anumit moment.
  + Pentru a maximiza capacitatea globală a celulei, throughput-ul utilizatorilor nu este cea mai bună metrică pe care trebuie să o urmărim.
* Identificatorul temporal al rețelei radio - Radio Network Temporary Identifier (RNTI)
  + Este utilizat pentru a diferenția modul de conectare al UE în interiorul celulei, canalul radio specific, un grup de UE al cărui control al puterii este emis e eNodeB, informații de sistem transmise de către eNodeB tuturor utilizatorilor.
  + Există mai multe tipuri de RNTI, cum ar fi: SI-RNTI, P-RNTI, C-RNTI, etc.
    - SI-RNTI (System Information RNTI) este utilizat pentru furnizarea informațiilor de sistem, este un RNTI comun, nu este alocat explicit niciunui UE, are o lungime fixă de 16 biți și valoarea sa este fixată la 65535 (0xFFFF).
    - P-RNTI (Paging RNTI) este utilizat de UE pentru a recepționa paginarea, este și el de asemenea un RNTI comun, are o lungime fixă 16 biți și valoarea sa este fixată la 65534 (0xFFFE).
    - C-RNTI (Cell RNTI) este o identificaere unică folosită pentru identificarea conexiunilor RRC (Radio Resource Control – controlul resurselor radio) și planificărilor dedicate unui anumit UE. C-RNTI are o lungime de 16 biți și valoarea sa poate varia de la 1 la 65523 (0x0001 până la 0xFFF3). [15]

## Metode de analiză

Conceptul de învățare mecanică (*machine learning)* se referă la studiul științific al algoritmilor și modelelor statistice pe care sistemele informatice le folosesc pentru a îndeplini eficient o sarcină specifică fără a folosi instrucțiuni explicite, bazându-se în schimb pe modele și inferențe. Este văzută ca un subset al inteligenței artificiale. Algoritmii de învățare mecanică realizează un model matematic bazat pe date de probă, cunoscute sub numele de "date de învățare", pentru a face predicții sau decizii fără a fi programate în mod explicit pentru a îndeplini sarcina. În aplicarea sa în întreaga problemă de afaceri, învățarea mecanică este menționată ca analiză predictivă. [16]

Sarcinile învățării mecanice sunt clasificate în două mari categorii: învățare supravegheată și învățare nesupravegheată.

În învățarea supravegheată, algoritmul construiește un model matematic dintr-un set de date care conține atât intrările, cât și ieșirile dorite. Algoritmii de clasificare și algoritmii de regresie sunt tipuri de învățare supravegheată. Algoritmii de clasificare sunt utilizați atunci când ieșirile sunt limitate la un set limitat de valori. Algoritmii de regresie sunt renumiți pentru ieșirile lor continue, adică pot avea orice valoare într-un interval. Exemple de valori continue sunt temperatura, lungimea sau prețul unui obiect.

În învățarea nesupravegheată, algoritmul construiește un model matematic dintr-un set de date care conține numai intrări și etichete de ieșire dorite.

În continuare voi prezenta câteva caracteristicii esențiale ale unor modele de analiză foarte bine cunoscute în domeniul învățării mecanice.

### Metoda pădurilor aleatoare (Random Forests)

Metoda pădurilor aleatoare (Random Forests) este un algoritm pe care dacă ar fi să îl încadrăm în una din cele două categorii menționate anterior, l-am aminti în grupul celor de învățare supravegheată. Poate fi folosit atât pentru clasificare, cât și pentru regresie. Este, de asemenea, algoritmul cel mai flexibil și ușor de utilizat. O pădure este formată din copaci. Pădurile aleatoare creează arbori de decizie pe eșantioane de date selectate aleatoriu, obțin previziuni din fiecare copac și selectează cea mai bună soluție prin vot. De asemenea, oferă un indicator destul de bun al importanței caracteristicilor.

Metoda pădurilor aleatoare are o varietate de aplicații, cum ar fi motoarele de căutare, clasificarea imaginilor, selecția caracteristicilor, clasificarea solicitanților de credite loiali, identificarea activității frauduloase și a prezicerea bolilor.

#### Cum funcționează algoritmul ?

[17] Să presupunem că vreau să merg într-o excursie și vreau să călătoresc într-un loc unde mă voi bucura.

Deci, ce fac pentru a găsi un loc care îmi va plăcea? Pot căuta online, citi recenzii despre blogurile și portalurile de călătorie sau îmi pot întreba și prietenii.

Să presupunem că am decis să îmi întreb prietenii și am discutat cu ei despre experiența lor de călătorie din trecut în diverse locuri. Am primit câteva recomandări de la fiecare prieten. Acum trebuie să fac o listă a locurilor recomandate. Apoi, le cer să voteze (sau să aleagă un loc potrivit pentru călătorie) din lista locurilor recomandate pe care am făcut-o. Locul cu cel mai mare număr de voturi va fi alegerea mea finală pentru călătorie.

În procesul de decizie de mai sus, există două părți. În prima parte este ca și cum am folosi algoritmul arborelui de decizie. Aici, fiecare prieten face o selecție a locurilor pe care le-a vizitat până acum.

A doua parte, după colectarea tuturor recomandărilor, este procedura de votare pentru selectarea celui mai bun loc din lista de recomandări. Acest întreg proces de a primi recomandări de la prieteni și de a le vota pentru a găsi cel mai bun loc este cunoscut sub numele de Algoritmul Pădurilor Aleatoare (Random Forests).

Din punct de vedere tehnic, este o metodă de ansamblu a arborilor de decizie generați pe un set de date divizat întâmplător. Această colecție de clasificatori de arbori de decizie este, de asemenea, cunoscută sub numele de pădure. Fiecare arbore depinde de o probă independentă aleatorie. Într-o problemă de clasificare, fiecare copac votează și clasa cea mai populară este aleasă ca rezultat final. În cazul regresiei, media tuturor producțiilor arborilor este considerată ca fiind rezultatul final. Este mai simplu și mai puternic comparativ cu ceilalți algoritmi de clasificare neliniară.

#### Pași logici de parcurgere

Algoritumul lucrează în patru pași :

1. Selectează eșantioane aleatoare dintr-un set de date dat.
2. Construiește un arbore de decizie pentru fiecare probă și obține un rezultat de decizie din fiecare arbore de decizie.
3. Efectuează un vot pentru fiecare rezultat prezis.
4. Selectează rezultatul predicției cu cele mai multe voturi ca predicție finală.

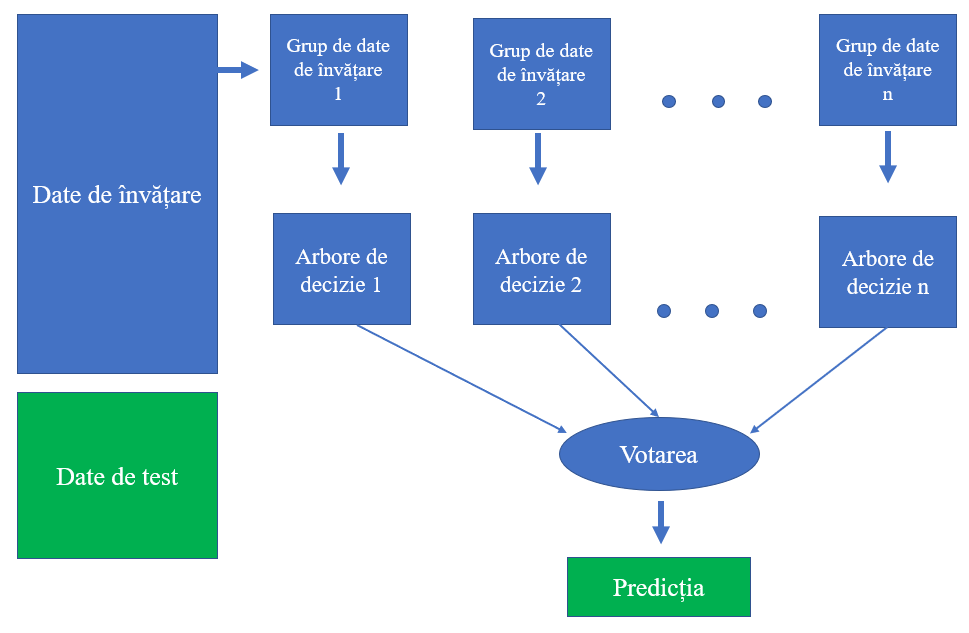


Figura 2.4 Schema logică de parcurgere a algoritmului Random Forests

### Metoda Creșterii Variației (Gradient Boosting)

Gradient Boosting este o tehnică de machine learning pentru regresia și clasificarea problemelor, care produce un model de predicție sub forma unui ansamblu de predicție slabă, de obicei arbori de decizie.

#### Descriere pentru ansamblu, colectare(bagging) și creștere(boosting)

Când încercăm să prezicem variabila țintă utilizând orice tehnică de machine learning, principalele cauze ale diferenței dintre valorile reale și cele prezise sunt zgomotul și varianța. Ansamblul ajută la reducerea acestor factori (cu excepția zgomotului, care este eroarea ireductibilă)

Un ansamblu este doar o colecție de predictori care se reunesc (de exemplu, din toate predicțiile) pentru a da o predicție finală. Motivul pentru care folosim ansambluri este că mulți predictori diferiți, care încearcă să prezică aceeași variabilă țintă, vor îndeplini o treabă mai bună decât oricare predictor singur. Tehnicile de asamblare sunt clasificate în continuare în Bagging și Boosting.

Bagging-ul este o tehnică simplă de asamblare în care construim mulți predictori independenți și îi combinăm folosind câteva tehnici de modelare medie (media ponderată, votul majoritar sau media normală).

De obicei, luăm un subeșantion de date aleatoriu pentru fiecare model, astfel încât toate modelele sunt puțin diferite unul de celălalt. Fiecare observație este aleasă cu înlocuirea care trebuie utilizată ca intrare pentru fiecare dintre modele. Deci, fiecare model va avea observații diferite bazate pe procesul de subeșantionare. Deoarece această tehnică necesită mulți predictori necorelați pentru a face un model final, reduce erorile prin reducerea varianței. Exemplu de ansamblu de bagging este modelul Random Forest. [18]

Boosting-ul este o tehnică de ansamblu în care predictorii nu sunt făcuți independent, ci secvențial.

Această tehnică folosește logica în care predictorii ulteriori învață din greșelile predictorilor anteriori. Prin urmare, observațiile au o probabilitate inegală de a apărea în modelele ulterioare, iar cele cu cea mai mare eroare apar cel mai mult. Deci, observațiile nu sunt alese pe baza procesului de subeșantionare, ci pe baza erorii. Predictorii pot fi aleși dintr-o gamă de modele cum ar fi arbori decizionali, regresori, clasificatori etc. Deoarece predictorii noi învață din greșelile comise de predictorii anteriori, este nevoie de mai puțin timp / iterații pentru a ajunge la predicții apropiate. Dar trebuie să alegem cu atenție criteriile de oprire sau ar putea duce la suprasolicitarea datelor de învățare. Gradient Boosting este un exemplu de algoritm de amplificare. [18]

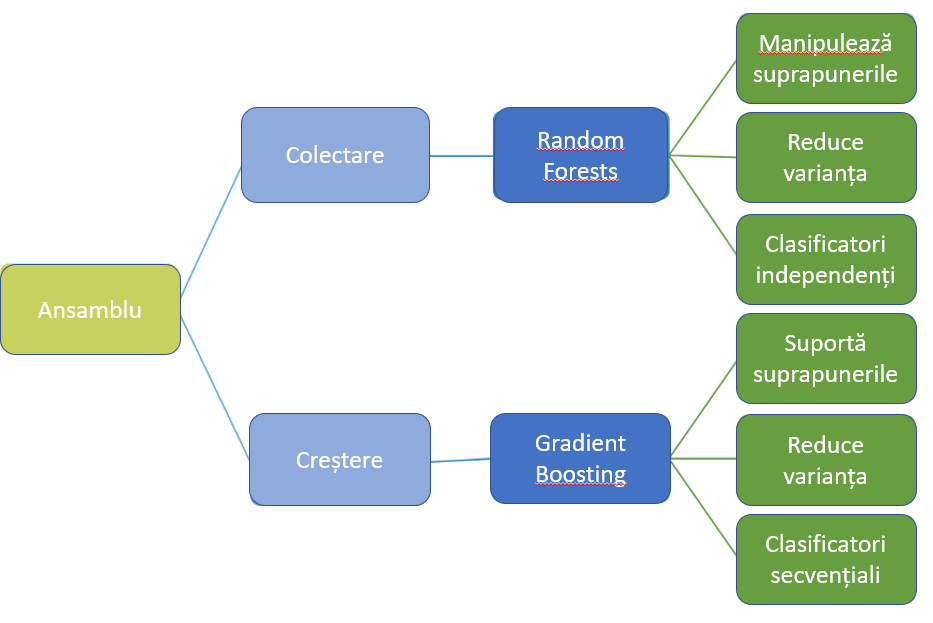


Figura 2.5 Definirea ansamblului

Logica din spatele algoritmului Gradient Boosting este simplă (poate fi înțeleasă intuitiv, fără a folosi notația matematică). O presupunere de bază a regresiei liniare este că suma reziduurilor sale este 0, adică reziduurile ar trebui să fie împărțite aleatoriu în jurul valorii de zero. Acum ne putem gândi la aceste reziduuri ca greșeli comise de modelul predictorului nostru. Deși modelele bazate pe arbori (considerând aici arborele de decizie ca modele de bază pentru Gradient Boosting) nu se bazează pe astfel de ipoteze, dar dacă ne gândim logic (nu statistic) la această ipoteză, putem argumenta că, dacă putem vedea un anumit model de reziduuri în jurul valorii de 0, putem folosi acest model de reziduuri pentru a se potrivi unui model predictiv.

Deci, intuiția din spatele algoritmului de Gradient Boosting este de a influența repetitiv modelele din reziduuri și de a întări un model cu predicții slabe și de a o face mai bine. Odată ce ajungem într-o etapă în care reziduurile nu au niciun model care să poată fi modelat, putem opri modelarea reziduurilor (altfel ar putea duce la suprasolicitare). În mod algoritmic, minimizăm funcția de pierdere, astfel încât pierderea de testare să atingă pragul minim.

Rezumând,

* Noi modelăm mai întâi datele cu modele simple și analizăm datele pentru erori.
* Aceste erori semnifică puncte de date care sunt greu de adaptat printr-un model simplu.
* Apoi, pentru modelele ulterioare, ne concentrăm în mod special pe acele date greu de potrivit pentru a le face corecte.
* În cele din urmă, combinăm toți predictorii, dând câteva ponderi fiecărui predictor.

### Algoritmul Regresiei Logistice (Logistic Regression)

Regresia logistică este în esență un algoritm de clasificare supravegheat. Într-o problemă de clasificare, variabila țintă (sau ieșire), y, poate lua doar valori discrete pentru setul de caracteristici dat (sau intrări), X.

Modelul construiește un model de regresie pentru a prezice probabilitatea ca o intrare de date să aparțină categoriei numerotate ca "1". La fel ca regresia liniară care presupune că datele urmează o funcție liniară, regresia logistică modelează datele folosind funcția sigmoid. [19]



Figura 2.6 Funcția sigmoid [19]

Regresia logistică devine o tehnică de clasificare numai atunci când un prag de decizie este adus în discuție. Setarea valorii pragului este un aspect foarte important al regresiei logistice și depinde de problema de clasificare în sine.

Decizia privind valoarea pragului este afectată în mare măsură de valorile preciziei și reapelării. În mod ideal, dorim ca atât precizia cât și reapelarea să fie 1, dar acest lucru este foarte rar. În cazul unui compromis Precizie-Reapelare, folosim următoarele argumente pentru a decide asupra situației:

* Precizie redusă / Reapelare ridicată: În aplicațiile în care vrem să reducem numărul de falsuri negative fără a reduce neapărat numărul de falsuri pozitive, alegem o valoare a deciziei care are o valoare redusă a preciziei sau o valoare ridicată a reapelării.

De exemplu, într-o aplicație de diagnosticare a cancerului, nu vrem ca niciun pacient afectat să fie clasificat ca nefiind afectat, fără a acorda o atenție deosebită dacă pacientul este diagnosticat greșit cu cancer. Acest lucru se datorează faptului că absența cancerului poate fi detectată de alte boli medicale, însă prezența bolii nu poate fi detectată pentru un pacient deja respins.

* Precizie înaltă / Reapelare redusă: În aplicațiile în care vrem să reducem numărul de falsuri pozitive fără a reduce neapărat numărul de falsuri negative, alegem o valoare de decizie care are o valoare ridicată a preciziei sau a o valoare scăzută a reapelării. De exemplu, dacă clasificăm clienții dacă vor reacționa pozitiv sau negativ la o reclamă personalizată, dorim să fim absolut siguri că clientul va reacționa pozitiv la publicitate, deoarece, în caz contrar, o reacție negativă poate provoca pierderi potențiale de vânzări de la client.

Pe baza numărului de categorii, regresia logistică poate fi clasificată astfel:

* Binomială: variabila țintă poate avea doar 2 tipuri posibile: "0" sau "1" care pot reprezenta "câștiga" vs "pierdere", "trece" vs "eșuează", "mort" vs "viu"
* Multinomial: variabila țintă poate avea 3 sau mai multe tipuri posibile care nu sunt ordonate (adică tipurile nu au semnificație cantitativă) cum ar fi "boala A" vs "boala B" vs "boala C".
* Ordinal: se ocupă cu variabilele țintă cu categorii ordonate. De exemplu, un scor de test poate fi clasificat ca: "foarte slab", "slab", "bun", "foarte bun". Aici, fiecare categorie poate primi un scor ca 0, 1, 2, 3.

O să detaliez cea mai simplă formă de regresie logistică (cea binomială) printr-un exemplu.

Considerăm cazul care mapează numărul de ore de studiu și rezultatul de la examen. Rezultatul poate lua doar două valori pentru care 1 înseamnă promovat și 0 eșuat.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nr ore | 0,5 | 1 | 1,5 | 2 | 2,5 | 3 | 3,5 | 4 | 5 |
| Scor | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

* Setul de date are ‘p’ caracteristici și ‘n’ observații
* Matricea caracteristicilor este reprezentată în felul următor :

X =

Unde conține valoarea celei de-a *j* caracteristică pentru a *i* observație.

Pentru a *i* obervație, poate fi reprezentat ca :

h() reprezintă răspunsul predicției pentru observația de indice i. Formula utilizată pentru a calcula h() este numită ipoteză și este de forma :

h() = +++ .... +

unde , , ...., sunt coeficienții regresiei.

Și într-o formulă compactă : h() = .

## Limbajele de programare

### SQL

SQL (Structured Query Language sau limbaj structurat de interogări) reprezintă un limbaj de programare care permite crearea de baze de date, adăugarea de informații și recuperarea de date precise care sunt necesare la un anumit moment de timp. Limbajul SQL a fost dezvoltat în special pentru lucrul cu baze de date precise care se axează pe modelul relațional, fiind în prezent cel mai utilizat limbaj folosit în cadrul sistemelor de gestiune a bazelor de date.

Limbajul SQL permite atât definirea structurii (schemei) bazei de date cât şi manipularea acesteia, sub formă de interogări (comenzi) pe care sistemele de gestiune a bazelor de date trebuie să le execute în mod corespunzător. Aceste interogări pot fi introduse direct în terminal, rezultatul afişându-se direct pe ecran sau pot fi trimise indirect către SGBD prin intermediul unor programe dezvoltate în limbaje de programare precum C++, Java, PHP, caz în care rezultatele sunt stocate linie cu linie în variabile de program.

Limbajul SQL funcționează pe bază de operatori și interogări. Printre cei mai importanți operatori se enumără egalitatea (EQUAL) “=”, diferența (“!=” sau “<>”), mai mare ca (“>”), mai mic ca (“<”), mai mare sau egal (“>=”), mai mic sau egal (“<=”), operatorul între (BETWEEN) ce se folosește la definirea unor intervale, asemenea (LIKE) folosit pentru a descoperi un model, în (IN) pentru a găsi unul sau mai multe rezultate dintr-o mulțime, este (IS) pentru a compara cu valoarea NULL și precum (AS) folosit pentru a schimba un nume de câmp atunci când se vizualizează rezultatele. O variabilă NULL este o variabilă specială, ce specifică faptul că într-un câmp anume, nu se află nicio valoare. NULL este diferit de 0, sau de un câmp ce conține spații, acestea la rândul lor reprezentând valori.

În prezent există o multitudine de dezvoltatori de sisteme de gestiune a bazelor de date care oferă diferite funcţionalităţi pentru uşurarea modului de interacţiune al utilizatorului cu baza de date pe care o administrează sau o utilizează. Putem aminti aici de câteva exemple de sisteme de gestiune a bazelor de date, precum: MySQL, Oracle, IBM DB2, Microsoft SQL Server, etc. Având în vedere însă, că SQL este un limbaj standardizat la nivel internaţional, toate aceste sisteme adoptă o aceeaşi normă pentru definirea relaţiilor şi manipularea datelor.

Limbajul SQL este format din patru componente, după cum urmează în Figura 2.6:

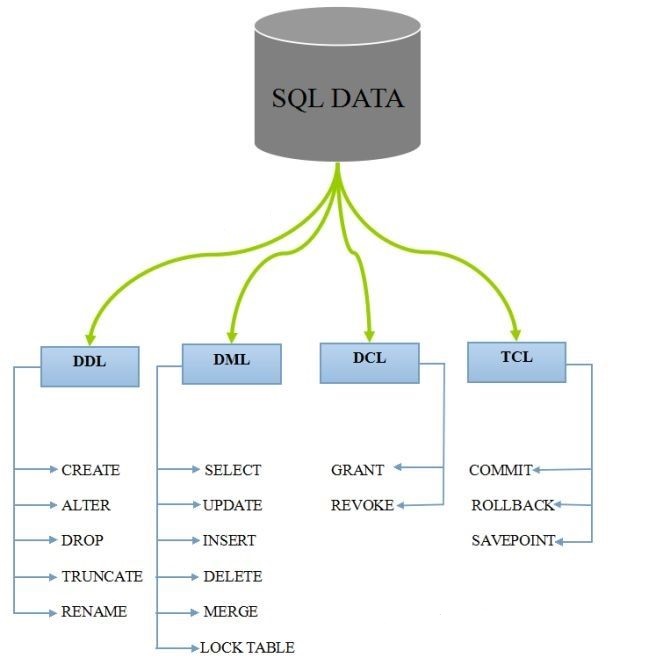


Figura 2.7 Tipuri de comenzi în SQL

În această lucrare voi folosi două dintre cele patru categorii, celelate nefiind de interes pentru subiectul pe care îl tratez :

-Limbajul de descriere a datelor (DDL – Data Description Language) care permite definirea schemei (structurii) bazei de date, definirea tabelelor şi a relaţiilor dintre elementele componente, precum şi atribuirea drepturilor de acces a utilizatorilor la baze de date;

-Limbajul de manipulare a datelor (DML – Data Manipulation Language) permite adăugarea de informaţii în baza de date (INSERT), actualizarea (UPDATE) sau ştergerea (DELETE) lor, precum şi interogarea datelor pentru a avea acces doar la o subcolecţie de informaţii care sunt utile utilizatorului la un anumit moment de timp. [10]

#### Limbajul de descriere a datelor – DDL

Limbajul de descriere a datelor este compus în principal din trei comenzi: CREATE, ALTER şi DROP. Instrucţiunea CREATE este folosită pentru crearea structurii (schemei) esenţiale a bazei de date, ALTER pentru modificarea structurii existente și DROP pentru ștergerea acesteia în întregime sau doar a unor componente (tabele).

O bază de date este definită de propria sa schemă. Din acest motiv, SQL propune crearea unei scheme înainte de definirea componentelor sale (tabele). Acest lucru se poate realiza cu ajutorul instrucțiunilor :

CREATE SCHEMA stats\_db;

Sau

CREATE DATABASE stats\_db;

Cele două instrucțiuni sunt echivalente. Crearea unei baze de date nu implică și selecția ei. De aceea, pentru a o putea manipula este necesară utilizarea comenzii USE.

USE stats\_db;

Odată creată baza de date și selectată, se pot defini tabele cu ajutorul clauzei CREATE TABLE specificând numele tabelului, numele și tipul coloanelor (numele atributelor plus domeniul asociat) precum și constrângerile acolo unde este cazul.

Odată definită structura unui tabel, aceasta poate fi verificată/vizualizată cu ajutorul comenzii :

DESCRIBE mytable;

### Limbaje de programare utilizate în știința datelor

În urma parcurgerii literaturii de specialitate referitoare la limbajele de programare utilizate în știința datelor, am ajuns la concluzia că răspunsul la întrebarea „care este cel mai bun limbaj utilizat în conceputul de machine learning ?” este unul subiectiv deoarece el poate fi dat doar în urma unei experiențe personale sau bazându-ne pe aspectul unui set de date.

Mai jos, voi construi o statistică pentru fiecare limbaj și aria sa de aplicabilitate, urmând să iau o decizie asupra limbajului pe care îl voi utiliza în partea de dezvoltare.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caracteristici/Limbaj | | Python | C/C++ | Java | R | JavaScript |
| Popularitate | Utilizat | 57% | 43% | 41% | 31% | 28% |
| Prioritizat | 33% | 19% | 16% | 5% | 7% |
| Aria de aplicabilitate | Prioritizat cel mai mult | Analiză: 44 %  Procesare de limbaj natural: 42 %  Exploatare web: 37% | Inteligența artificială în jocuri: 29%  Locomoția roboților: 27%  Securitatea rețelelor: 26% | Managementul clienților: 26%  Securitatea rețelelor: 23%  Detecția fraudei: 22% | Analiză: 11%  Bioinformatică: 11%  Detecția fraudei: 9% | Managementul clienților: 10%  Motoare de căutare: 9% |
| Prioritizat cel mai puțin | Inteligența artificială în jocuri: 26%  Detecția fraudei: 26%  Securitatea rețelelor: 24% | Detecția fraudei: 12%  Sisteme de recomandare: 12%  Analiză: 9% | Analiză: 15%  Bioinformatică: 13% | Inteligența artificială în jocuri: 3%  Locomoția roboților: 1% | Mentenanță industrială: 2%  Bioinformatică: 2%  Analiză: 2% |
| Istoricul profesional | Prioritizat cel mai mult | Știința datelor: 38% | Inginerie electronică: 32% | Dezvoltarea aplicațiilor desktop: 21% | Analist date: 14% | Dezvoltare web: 16% |
| Prioritizat cel mai puțin | Inginerie mecanică: 27% | Dezvoltare web: 8% | Inginerie electronică: 13% | Inginerie electronică: 3% | Inginerie electronică: 3% |
| Motive pentru a-l folosi în algoritmii de machine learning | Prioritizat cel mai mult | Curiozitatea de a afla detalii despre machine learning: 38% | Pentru a adăuga machine learning aplicațiilor existente: 20% | La solicitarea companiei: 27% | Obținerea diplomei de studii: 7% | Creșterea șanselor de a avea proiecte profitabile: 8% |
| Prioritizat cel mai puțin | Obținerea diplomei de studii: 26% | În construirea aplicțiilor de înaltă competiție : 14% | Curiozitatea de a afla detalii despre machine learning: 14% | Curiozitatea de a afla detalii despre machine learning: 5% | Obținerea diplomei de studii: 5% |

Tabelul 3 Limbaje de programare utilizate în știința datelor [20]

* Așadar, conform [21], Python conduce pachetul acestor limbaje, 57% dintre cercetătorii de date și dezvoltatorii de machine learning îl folosesc, iar 33% îl acordă prioritate în dezvoltare. Nu e de mirare, având în vedere toată evoluția cadrelor Python de învățare profundă din ultimii 2 ani.
* Python este adesea comparat cu R, dar nu este nici pe departe comparabil în ceea ce privește popularitatea: R se află pe locul patru în utilizarea generală (31%) și al cincilea în prioritizare (5%). Aceasta înseamnă că în majoritatea cazurilor R este un limbaj complementar, nu o primă alegere, o indicație clară că tendințele de utilizare ale Python sunt opuse celor ale lui R.
* C / C ++ este îndepărtat față de Python, atât în ​​utilizare (44%) cât și în prioritizare (19%). Inteligența artificială (AI) în jocuri (29%) și locomoția roboților (27%) sunt cele două zone în care C / C ++ este favorizat cel mai mult, având în vedere nivelul de control, performanța ridicată și eficiența necesară. Aici, un limbaj de programare de nivel inferior, cum ar fi C / C ++, care vine cu biblioteci AI extrem de sofisticate, este o alegere naturală.
* Java urmărește foarte îndeaproape C / C ++, în timp ce JavaScript este al cincilea în utilizare, deși cu o performanță puțin mai bună de prioritizare decât R (7%). În schimb, Java este prioritată mai mult de cei care lucrează la securitatea rețelelor / atacurile cibernetice și detectarea fraudelor, cele două zone în care Python este cel mai puțin prioritizat.

În urma acestei analize, am luat decizia de a utiliza limbajul Python, însă înainte de aceasta doresc să mă informez și despre câteva dezvantaje pe care le pot întâmpina odată cu utilizarea acestui limbaj.

Argumente contra

* Știm că Python este un limbaj interpretat, codul din Python este executat linie cu linie. Astfel, Python duce deseori la executarea lentă în comparație cu alte limbi de programare. Viteza nu este o problemă decât dacă este un punct principal pentru proiect.
* Deoarece Python este un limbaj dinamic, acesta necesită mai multe teste și are erori care apar doar în timpul rulării.
* Python are limitări cu accesul la baza de date. În comparație cu JDBC și ODBC, nivelul de acces al bazei de date al Python este considerat subdezvoltat și primitiv. De asemenea, nu poate fi aplicat în întreprinderile care au nevoie de o interacțiune lină a datelor moștenite complexe.
* Python nu acceptă mai multe fire de execuție din cauza blocării globale a interpretorului, adică GIL; acest lucru permite doar un singur fir de execuție la un moment dat. Programele cu mai multe fire de execuție legate de CPU pot fi mai lente decât cele cu un singur fir de execuție;

Așadar, consider că dezavantajele mai sus menționate nu vor avea un impact important asupra algoritmului pe care o să îl dezvolt, de aceea voi rămâne la decizia de a utiliza limbajul de programare Python.

# Implementare Software

Figura 3.1 este o reprezentare grafică a întregului proces descris în prezenta lucrare. Așa cum am menționat și în introducere, în prima etapă a implementării se vor obține o serie de indicatori de performanță (trafic, încărcare, calitatea canalului), rezultați în urma filtrării cu ajutorul interogărilor SQL a unor seturi de date generate de interacțiunea utilizator – rețea. Mai departe, datele obținute în prima etapă vor fi divizate în date de învățare și date de test astfel încât să faciliteze rularea unui model predictiv peste ele. Rezultatele generate de algortimul predictiv vor fi prezentate sub formă grafică și vizualizate, aceasta etapă fiind ultima etapă a implementării software a procesului.

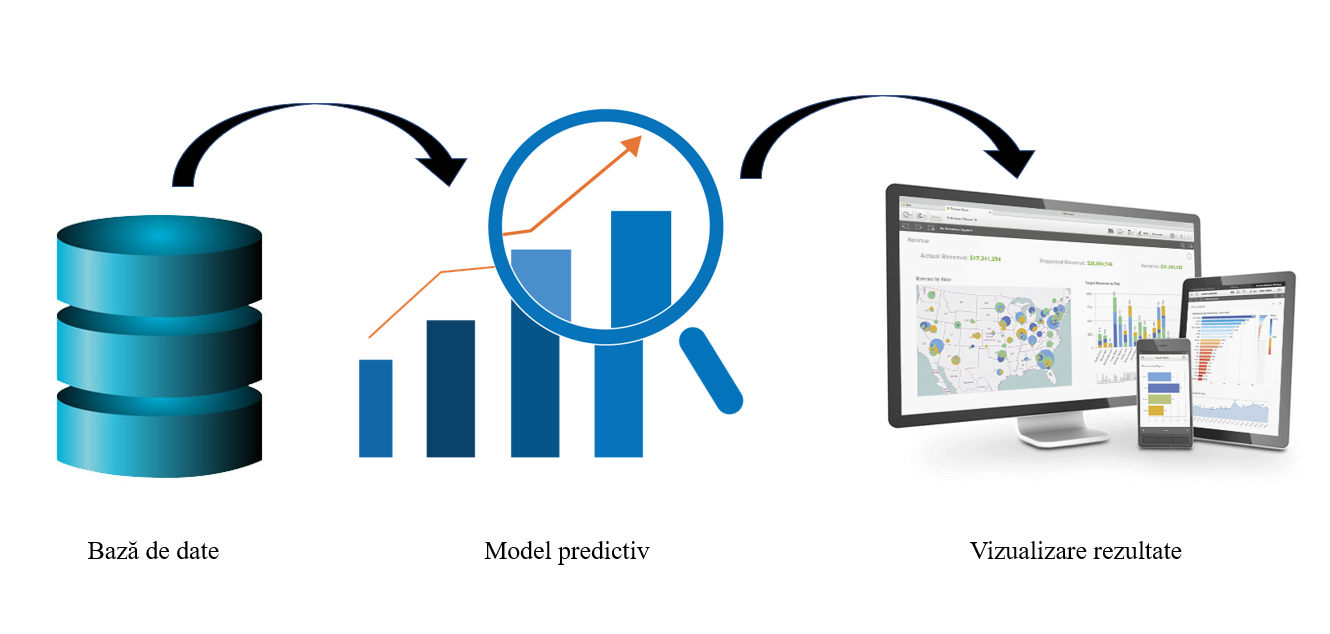


Figura 3.1 Schema generală a implementării

Baza de date este stats\_db și este create în utilitarul MySQL, iar modelul predictiv va fi ales unul dintre mai mulți pe care îi voi aplica printre care se numără Regresia Logistică sau algoritmul Pădurilor aleatoare. În parte de vizualizare rezultate, vor fi prezentate atât rezultate ale etapelor intermediare ale codului, cât și rezultatele finale pe baza cărora se va emite o serie de concluzii.

## Resurse utilizate

Cele trei etape descrise la începutul acestui capitol au fost posibile ca rezultat al utilizării a trei utilitare : pentru prima etapă am folosit MySQL Workbench, unde am creat o bază de date în care am inserat datele colectate, pentru a doua etapă, colectarea datelor din bază și crearea modelului predictiv le-am efectuat în utilitarul Spyder dezvoltat pentru limbajul de programare Python, iar reprezentarea grafică a rezultatelor am ales să o efectuez prin intermediului utilizarului Qlik Sense, utilizat frecvent în analiza datelor.

### MySQL Workbench

În cadrul acestui material va fi utilizat pentru exemplificare sistemul de gestiune al bazelor de date MySQL dezvoltat de compania suedeză MySQL AB. MySQL este un sistem de tip open source (codul sursă este disponibil în mod gratuit şi poate fi modificat în funcţie de necesităţile programatorului), foarte popular printre dezvoltatorii de aplicaţii software şi pagini web. Are la bază limbajul de date SQL, rulează pe orice tip de sistem de operare şi poate fi interconectat cu uşurinţă cu alte aplicaţii dezvoltate în limbaje de programare precum: C++, Java, Pyth0n, PHP, etc. [10]

MySQL Workbench este un instrument grafic pentru lucrul cu serverele și bazele de date MySQL. MySQL Workbench suportă pe deplin versiunile serverului MySQL 5.6 și versiuni ulterioare.

Funcționalitatea MySQL Workbench acoperă cinci subiecte principale [22]:

1. Dezvoltare SQL: vă permite să creați și să gestionați conexiuni la serverele de baze de date. Înainte de a vă permite să configurați parametrii conexiunii, MySQL Workbench oferă posibilitatea de a executa interogări SQL pe conexiunile bazei de date utilizând editorul SQL încorporat.
2. Modelarea datelor (Design): vă permite să creați modele de schemă de bază de date, în mod grafic, invers și inginer între o schemă și o bază de date live și să editați toate aspectele bazei dvs. de date folosind editorul de tabele complet. Editorul de tabel oferă facilități ușor de utilizat pentru editarea tabelelor, coloanelor, indexurilor, declanșatorilor, partiționării, opțiunilor, inserturilor și privilegiilor, rutinelor și viziunilor.
3. Administrare server: Permite administrarea instanțelor serverului MySQL prin administrarea utilizatorilor, efectuarea de backup și recuperare, inspectarea datelor de audit, vizualizarea sănătății bazei de date și monitorizarea performanței serverului MySQL.
4. Migrarea datelor: Vă permite să migrați de la Microsoft SQL Server, Microsoft Access, Sybase ASE, SQLite, SQL Anywhere, PostreSQL și alte tabele sau obiecte către MySQL. Migrarea acceptă, de asemenea, migrarea de la versiunile anterioare ale MySQL la cele mai recente versiuni.
5. Suport pentru MySQL Enterprise: Suport pentru produsele Enterprise precum MySQL Enterprise Backup, MySQL Firewall și MySQL Audit.

MySQL Workbench este disponibil în două ediții: ediția comunitară și ediția comercială. Ediția comunitară este disponibilă gratuit. Ediția comercială oferă funcții Enterprise suplimentare, cum ar fi accesul la MySQL Enterprise Backup, MySQL Firewall și MySQL Audit.

Când este pornit, MySQL Workbench se deschide în fila ecranului de “acasă”. Inițial, pe ecran se afișează un mesaj de întâmpinare și se afișează link-uri către Documentatie (Browse Documentation>), Blog (Read the Blog>) și Discuții de pe forumuri (Discuss on Forums>). În plus, ecranul de pornire oferă acces rapid la conexiunile, modelele MySQL și expertul de migrare MySQL Workbench.

Așa cum este prezentat în figura următoare, un panou lateral de pe ecranul de acasă vă permite să comutați între conexiunile MySQL (selectate în figura). Ultima opțiune din panoul lateral deschide MySQL Workbench Migration Wizard într-o filă nouă.

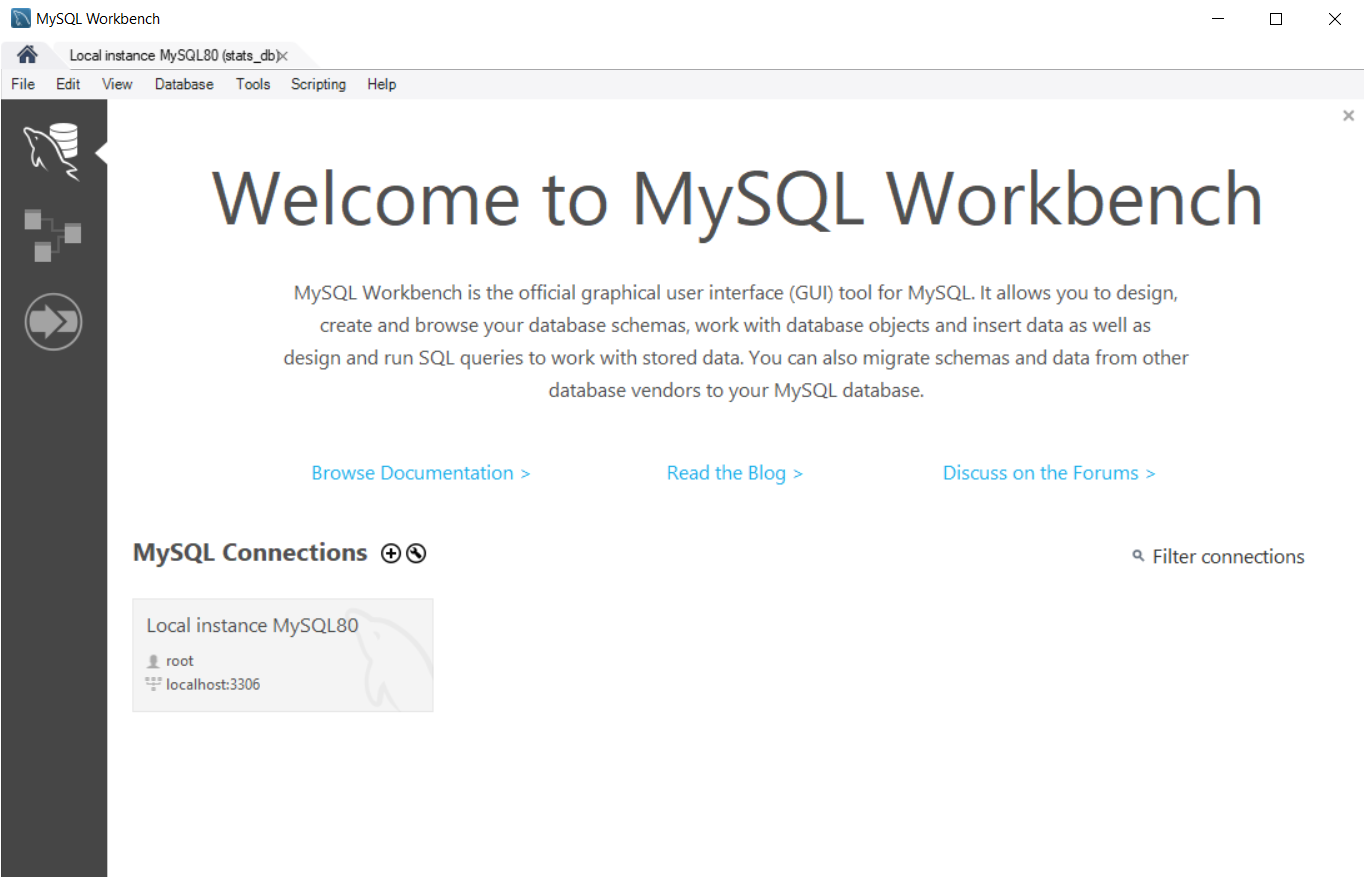


Figura 3.2 Fila de pornire a MySQL Workbench

Accesul în pagina de editare se realizează prin intermediul autentificării cu nume de utilizator și parolă, care sunt setate în timpul instalării utilitarului. În cazul de față, pentru autentificarea în instanța locală se va folosi utilizatorul ‘root’, iar parola, de asemenea, ’root’.

Fila de editare va fi cea din imaginea următoare, iar semnificațiile pictogramelor vor fi explicate ulterior.

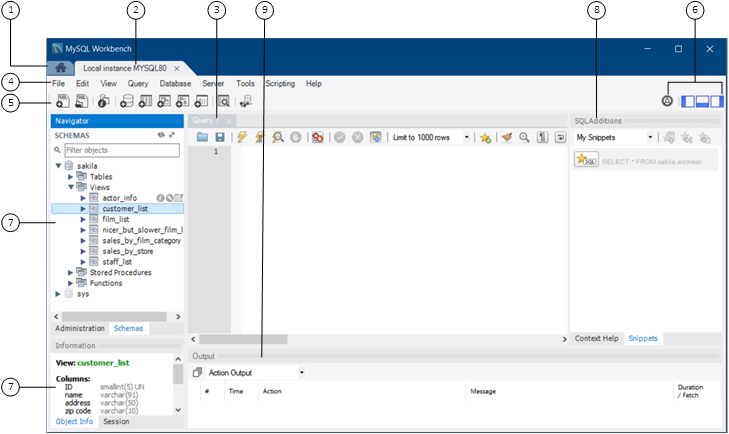


Figura 3.3 Fila de editare a MySQL Workbench

1. Fila ecranului inițial. Ea oferă acces rapid la conexiuni, modele și expertul Migrație MySQL. Spre deosebire de celelalte file principale, fila ecranului de pornire nu se închide.
2. Fila Conexiune. Fiecare conexiune făcută serverului MySQL este reprezentată de o filă de conectare separată. Un server poate fi activ sau inactiv când fila de conectare este deschisă.
3. Fila de interogări SQL. Fila interogare SQL este o filă secundară care se deschide implicit când faceți o conexiune la un server MySQL. Fiecare fila de interogare este identificată în mod unic printr-un număr incrementant: interogarea 1, interogarea 2 și așa mai departe. Toate filele de interogări SQL oferă o zonă pentru editarea interogărilor.
4. Bara de meniu principală. Bara de meniu are următoarele meniuri: Fișier, Editare, Vizualizare, Interogare, Bază de date, Server, Instrumente, Scripting și Ajutor. Acțiunile disponibile vă depind de tab-ul selectat când faceți click pe un meniu.
5. Bara principală de instrumente
6. Acțiuni rapide
7. Panoul lateral al barei. Bara laterală are două etichete principale: Navigator și Informații. Etichetele sunt omise pe unele gazde.
8. Panoul lateral secundar (Adăugări SQL)
9. Panoul zonei de ieșire. Panoul de ieșire poate afișa un rezumat al interogărilor executate în următoarele forme: Ieșire de acțiune, Ieșire text sau Ieșire istoric.

### Spyder (Anaconda)

[23] Spyder este un mediu științific puternic scris în Python, pentru Python, proiectat de oameni de știință, ingineri și analiști de date. Dispune de o combinație unică între funcționalitatea avansată de editare, analiză, depanare și profilare a unui instrument de dezvoltare cuprinzător, cu explorarea datelor, execuția interactivă, inspecția profundă și capabilitățile de vizualizare frumoase ale unui pachet științific. În plus, Spyder oferă integrarea cu multe pachete științifice populare, printre care numerele NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy și altele. Dincolo de numeroasele caracteristici încorporate, Spyder poate fi extins și mai mult prin intermediul pluginurilor terțe. ca o bibliotecă de extensii PyQt5, permițându-vă să vă bazați pe funcționalitatea acesteia și să încorporați componentele sale, cum ar fi consola interactivă sau editorul avansat, în software-ul propriu.

Spyder este inclus în mod implicit în distribuția Anaconda Python, care vine cu tot ce aveți nevoie pentru a începe într-un pachet „totul în unul”.

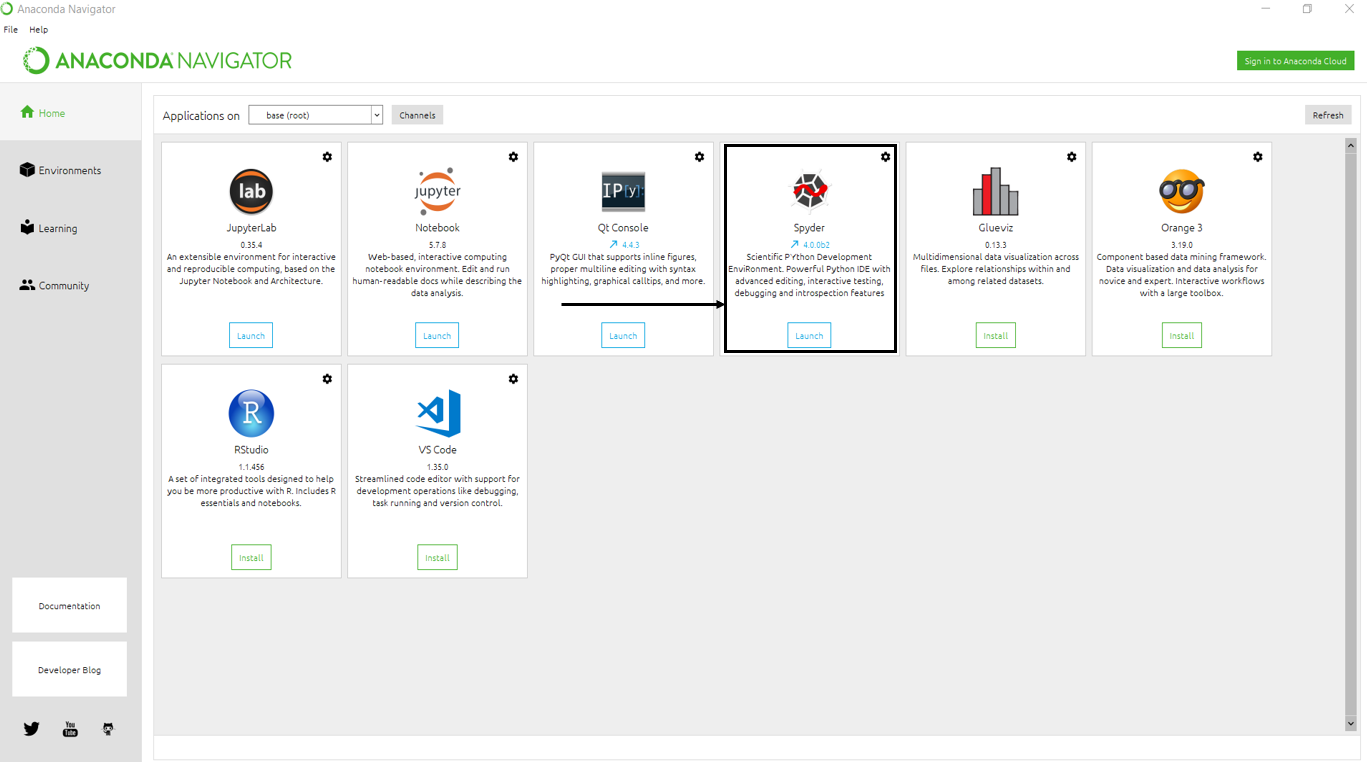


Figura 3.4 Fila de start Anaconda Navigator

În Figura 3.5 este reprezentată interfața Spyder, unde 1) reprezintă zona de editare, 2) zona de afișare (ajutor, variabile, grafice, fișiere), iar 3) consola sau zona de depanare.

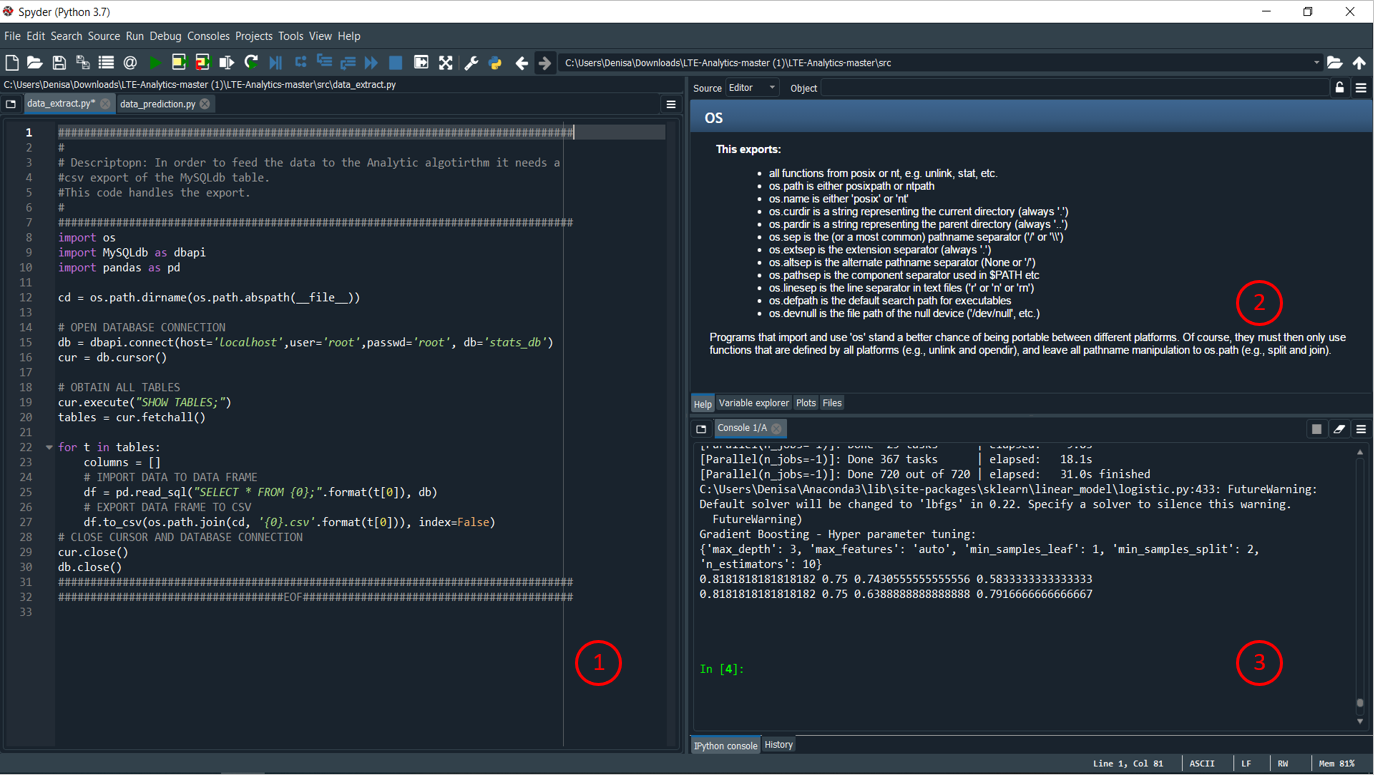


Figura 3.5 Fila de activitate Spyder 4

1. Editorul multi-lingvistic al lui Spyder integrează un număr de instrumente puternice pentru o experiență de editare eficientă și ușor de utilizat. Caracteristicile cheie ale editorului includ evidențierea sintaxei (pigmenți); codul în timp real și analiza stilului (pyflakes și pycodestyle); finalizarea la cerere, divizare orizontală și verticală și multe altele.
2. Puteți utiliza panoul de ajutor pentru a găsi și afișa documentația pentru orice obiect, inclusiv module, clase, funcții și metode. Ajutorul poate fi preluat atât prin analiza statică a fișierelor deschise în Editor, fie prin inspectarea dinamică a unui obiect într-o Consola IPython.

Puteți să declanșați ajutor introducând manual numele obiectului în caseta Obiect, apăsând comanda rapidă de configurare (Ctrl-I în mod implicit) sau chiar automat, dacă doriți, când introduceți o paranteză stângă (() după numele unei funcții sau metode.

“Variable Explorer” afișează conținutul spațiului de nume (toate referințele globale ale obiectelor, cum ar fi variabilele, funcțiile, modulele etc.) ale sesiunii IPython Console selectate și vă permite să interacționați cu acestea printr-o varietate de editori bazați pe GUI.

1. Depanarea în Spyder este susținută prin integrarea cu depanatorul ipdb îmbunătățit din Consola IPython. Aceasta permite vizualizarea și controlul punctelor de întrerupere și a fluxului de execuție chiar de la GUI-ul Spyder, precum și de la toate comenzile familiare ale consolei IPython.

### Qlik Sense (Desktop)

Qlik Sense este un utilitar utilizat adesea în știința datelor și este un serviciu de sine stătător. El oferă utilizatorilor săi posibilitatea de a crea vizualizări personalizate și interactive ale datelor, rapoarte și tablouri de control, cu mare ușurință.

Atunci când deschidem versiunea de desktop (Qlik există și în variantă server), Qlik Sense se deschide în nodul central (hub). Nodul central fiind locul în care se regăsesc toate aplicațiile, iar dacă facem click pe una dintre ele, ea se va deschide într-o pagină separată.

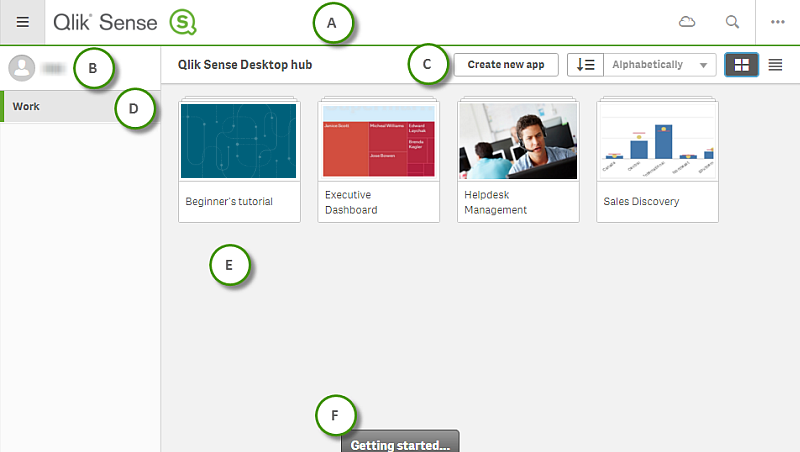


Figura 3.6 Nodul central Qlik Sense

1. Bara de instrumente de putem deschide sau închide meniul de navigare, scurtătură către locația de stocare a Qlik Sense, locul de căutare a aplicațiilor
2. Informații de autentificare
3. Rubrica de creare a unei noi aplicații
4. Domeniul de găzduire al aplicațiilor
5. Zona principal care conține toate aplicațiile create
6. Adresă către documente de învățare

O aplicație Qlik Sense este o colecție de elemente de date reutilizabile (măsuri, dimensiuni și vizualizări), pagini și povești. Este o entitate autonomă care include date într-un model de date structurat pentru analiză. [24]

Scopul unei aplicații este de a ne permite să facem descoperiri și decizii de date utilizând vizualizările de date și efectuarea de selecții.

|  |  |
| --- | --- |
| Figura 3.7 Vizualizare din interiorul aplicației | Figura 3.8 Vizualizare din zona de dezvoltare |

*Elemente de creare :*

* Script de încărcare a datelor – ne putem conecta la baze de date, fișiere Excel, pagini ale fișierelor Excel, etc.
* Modelul de date
* Măsurători – sunt expresii și calcule aplicate datelor pentru a fi vizualizate. Expresiile sunt compuse din funcții de agregare, cum ar fi sum sau max, combinate cu unul sau mai multe câmpuri
* Dimensiuni – determină cum vor fi grupate datele în vizualizare

Vizualizările sunt următorul pas după crearea aplicației și încărcarea datelor. Ele permit prezentarea datelor astfel încât utilizatorii aplicației să le poată interpreta și explora, sunt ușor de adăugat și personalizat și pot lua forma unor diagrame, tabele, hărți sau altele.

Pentru a crea vizualizări eficiente este nevoie ca datele și sursele de să fie în înțelese, să fie alese corespunzător tipurile de vizualizări care se potrivesc scopului și ca acestea să fie cât mai inteligibile utilizatorului.

## Aspecte practice

Pornind de la Figura 3.1, voi descrie etapele menționate în subcapitolele următoare.

### Extragerea datelor

Această etapă cuprinde două stări : una de creare a bazei de date, conținând măsurătorile colectate de pe dispozitive și apoi extragerea lor într-un fișier de tip Excel.

#### Crearea bazei de date

În utilitarul MySQL Workbench, am creat conexiunea cu numele ’Local instance MYSQL80’ la care este posibilă autentificarea utilizând numele de utilizator ‘root’, numele host-ului ‘localhost’, portul ‘3306’, așa cum este evidențiat în Figura 3.9 în rubrica ‘Schemas-Session’.

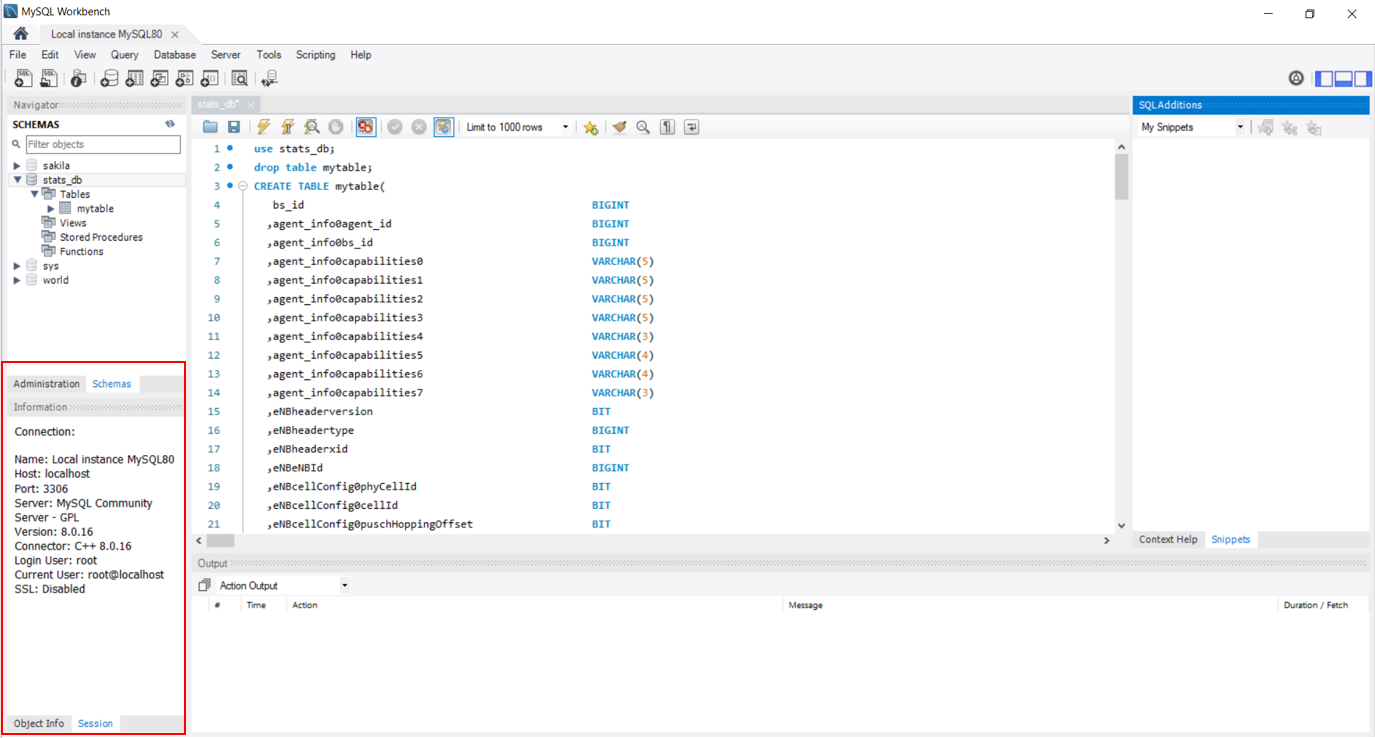


Figura 3.9 Detalii de autentificare

Vom vedea ulterior că pentru a putea folosi valorile introduse în câmpurile bazei de date în scopuri externe (cum ar fi utilitarul Spyder) este nevoie de aceste date de autentificare.

În continuare, trecem la realizarea propriu-zisă a bazei de date, lucru posibil prin comenzile (descrise în secțiunea 2.3.1.1) :



Mai apoi urmează crearea tabelului care are numele ‘mytable’ cu câmpurile și tipul datelor ce vor fi inserate în aceste câmpuri. Pentru a optimiza structura tabelelor este necesară întotdeauna alegerea tipului de date potrivit care utilizează cât mai puțină memorie, dar care să permită stocarea oricărei informații care ar putea fi plasată în acea categorie.

Câteva dintre declarări pot fi observate în Figura 3.9 în fila de interogări SQL (vezi Figura 3.3).

De exemplu, nu are sens definirea coloanei “agent\_info0capabilities4” ca un tip de date VARCHAR(5) deoarece despre lungimea șirului de caractere se știe că nu va depăși 3, însă pe de altă parte, dacă definim coloana “agent\_info0capabilities0” ca un tip de date VARCHAR(3), există riscul ca pentru unele înregistrări să nu poată fi stocată informația complet.

Instrucțiunea ‘drop table’ este utilizată în momentul în care în tabel adaug înregistrări noi pentru a șterge conținutul vechi.

Tipuri de date numerice exacte – permit reprezentarea unei valori în mod precis sub formă de număr întreg sau fracționar.

* BIGINT – reprezintă un număr întreg foarte lung, reprezentat pe 64 de biți. Domeniul de valori în cazul reprezentării cu semn este de la -9223372036854775808(-263) până la 9223372036854775808 (263). Domeniul de valori în cazul reprezentării fără semn este de la 0 până la 18446744073709551615(264). Opțional se poate impune ca numărul să fie reprezentat din M cifre (BIGINT(M)).
* BIT – reprezintă un număr întreg care poate lua una din valorile 0,1, NULL [10]

Tipuri de date tip șir de caractere

1. VARCHAR(M) – definește un șir variabil de caractere de lungime maximă M. În acest caz, lungimea șirului de caractere (și implicit spațiul de memorie necesar pentru stocare) se stabilește adaptive în funcție de informația introdusă în coloană, dar nu poate depăși valoarea lui M. [10]

După declararea tuturor câmpurilor, urmează inserarea valorilor. O instrucțiune de inserare arată de tipul celei de mai jos și conține numele tabelului în care se inserează, numele coloanelor din tabel în care se inserează și apoi valorile ce se inserează în câmpurile menționate :

INSERT INTO mytable (`bs\_id`,`agent\_info0agent\_id`,`agent\_info0bs\_id`,`agent\_info0capabilities0`,`agent\_info0capabilities1`, ........ ,`LClcUeConfig0lcConfig2qosBearerType`,`LClcUeConfig0lcConfig2qci`,`total\_bytes\_sdus\_ul`,`total\_bytes\_sdus\_dl`,`y`) VALUES (10005,5,10005,'LOPHY','HIPHY', ......., 0,1,2181380,389398,'yes');

Captura următoare de ecran confirm faptul că înregistrările au fost inserate cu succes.

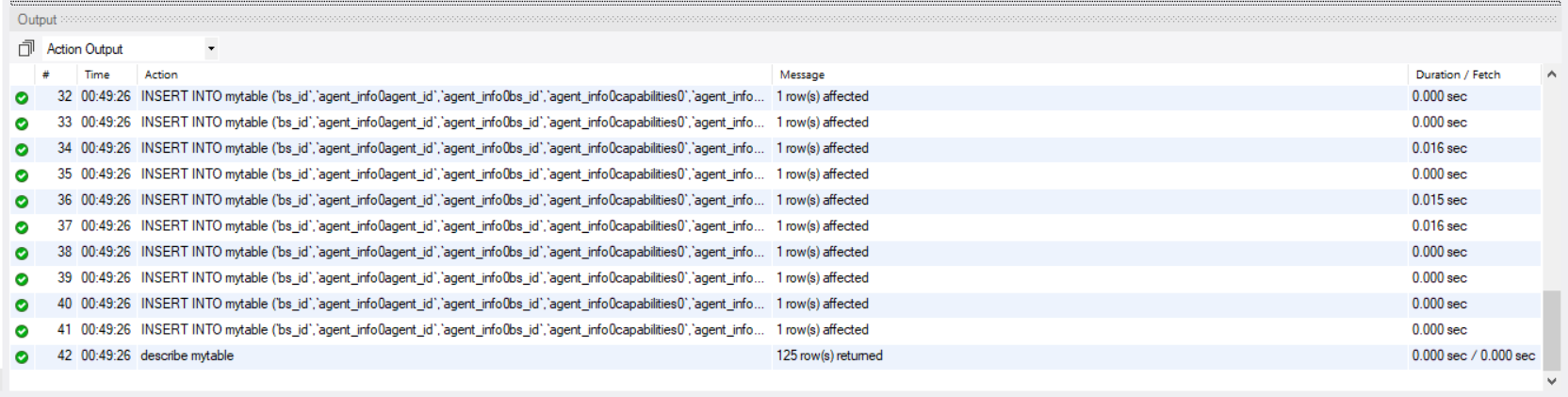


Figura 3.10 Inserarea înregistrărilor

După ce am definit structura tabelului doresc să o vizualizez, așa că voi folosi în cele din urmă comanda:

DESCRIBE mytable;

al cărei rezultat l-am atașat în Figura 3.11.

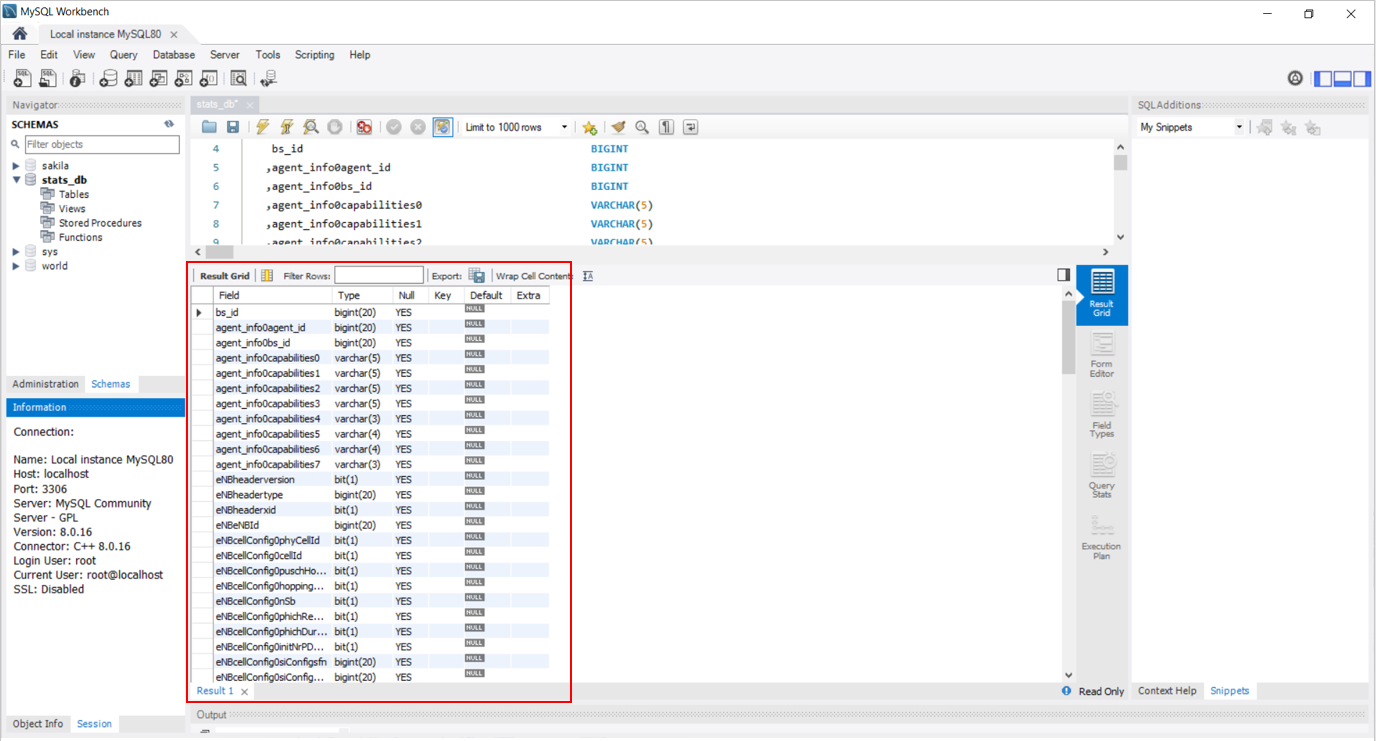
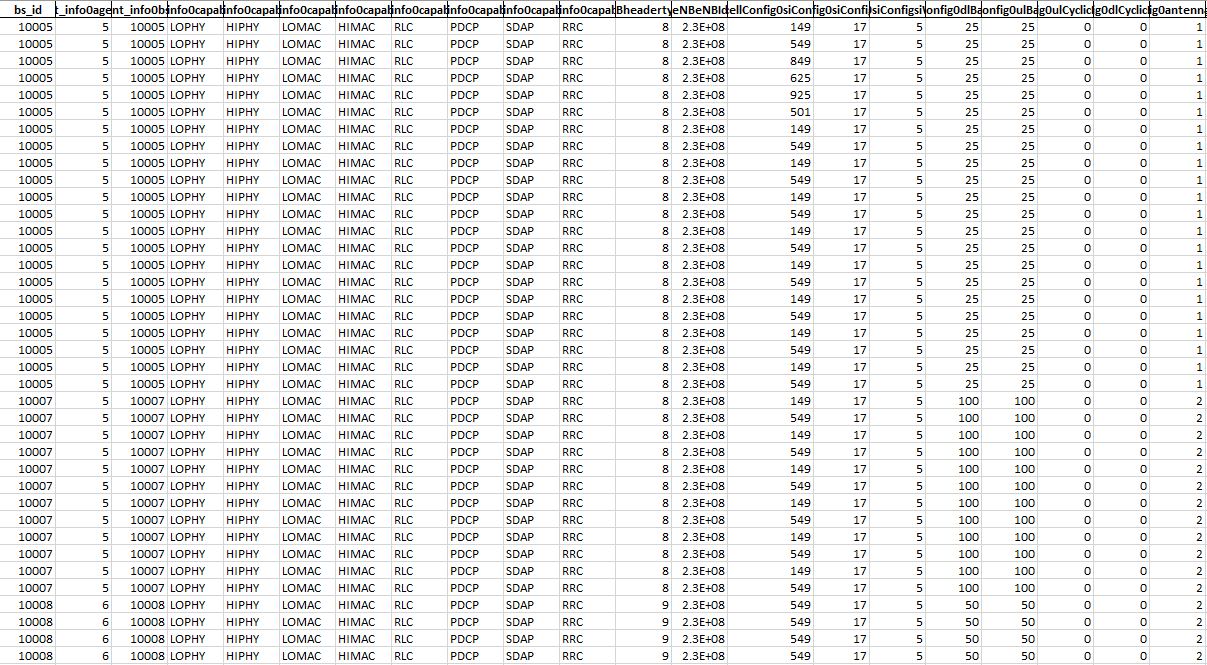


Figura 3.11 Descrierea tabelului

Iar mai jos, în , vom avea o vedere asupra tabelului creat, conținând numai câteva dintre coloanele pe care le conține baza de date, însă este un aspect destul de important pentru a vedea sub ce formă se prezintă datele în baza de date create.

Tabelul 4 Vedere din stats\_db



#### Extragerea datelor în fișier Excel

Pentru a face posibilă extragerea datelor se va scrie un algoritm în Python (vezi ANEXA 2), după cum urmează în Figura 3.12.

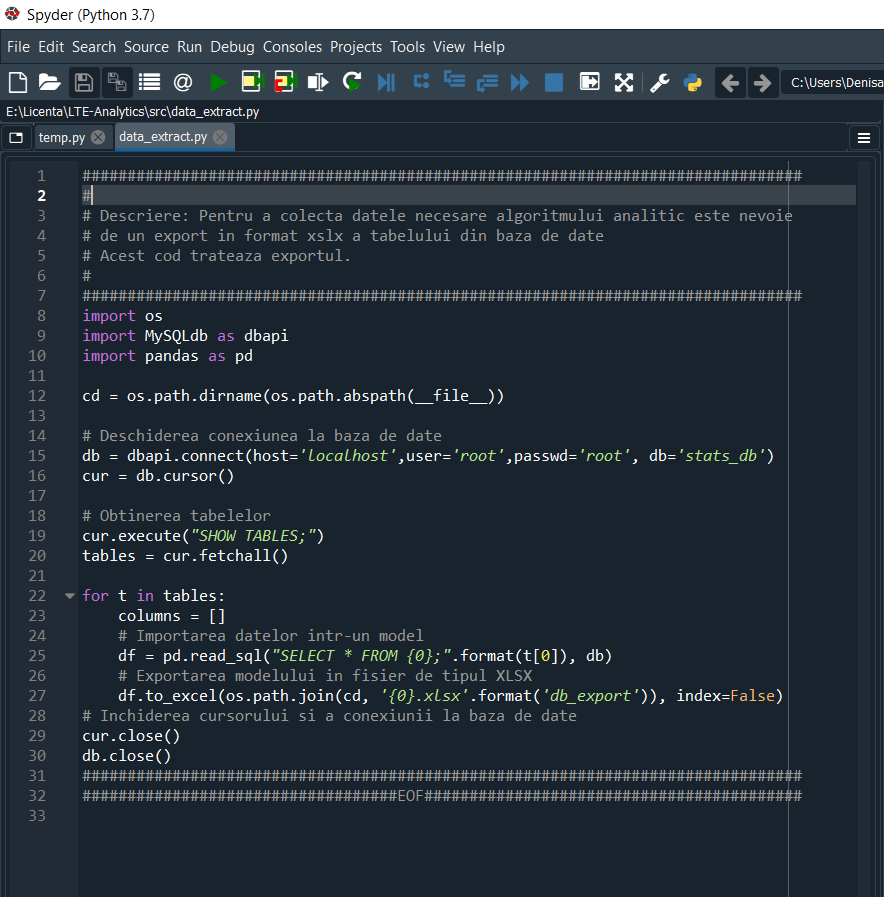


Figura 3.12 Algoritm de extragere a datelor

Așa cum am menționat în secțiunea 3.2.1.1, am folosit credențialele de autentificare în baza de date pentru a avea permisiunea de a citi și prelua datele din tabel.

Interogările SQL cu ajutorul cărora acest lucru este posibil sunt următoarele :

"SHOW TABLES;" //cu care vizualizăm tabelele din bază și mai apoi le stocăm în variabila ‘tables’

"SELECT \* FROM {0};" //cu care extragem informațiile din fiecare tabel stocat în variabila ‘tables’.

Prin intermediul variabilei ‘cd’, datele vor fi exportate în fișierul Excel, cu numele ‘db\_export’ care va fi localizat în aceeași cale cu fișierul sursă Python ‘data\_extract.py’.

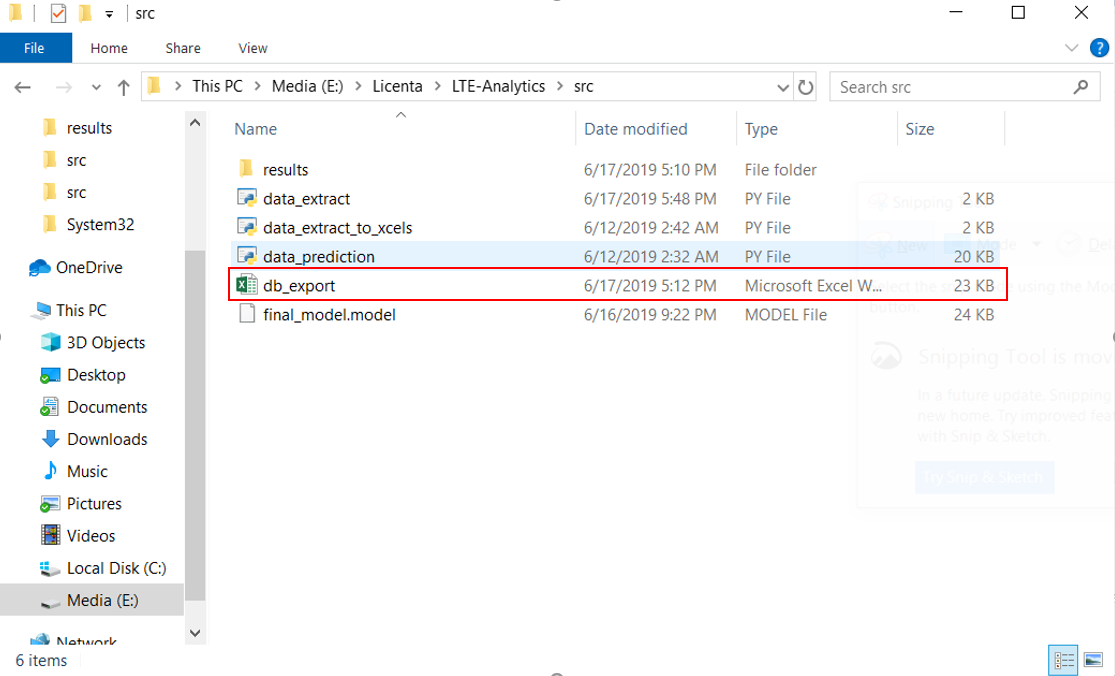


Figura 3.13 Localizarea fișierului de tip xlsx

### Algoritmii de predicție

Așa cum este menționat în secțiunea 3.2.1, algoritmul de predicție scris în limbajul Python, salvat în fișierul data\_prediction.py (vezi ANEXA 3) începe cu procurare datelor din fișierul Excel creat în secțiunea 3.2.1.2 pe care le voi stoca într-un model de date pe care urmează să îl prelucrez mai departe în algoritm.

Pentru o înțelegere mai ușoară a codului, am creat diagrama software a acestuia, ale cărei etape o să le detaliez ulterior. Ea se regăsește în Figura 3.14.



Figura 3.14 Diagrama software

Pentru început voi construi matricea de corelație. Matricea de corelație este un tabel care arată coeficienții de corelație dintre variabile. Fiecare celulă din tabel arată corelația dintre două variabile. Matricea de corelație este utilizată ca mod de sumarizare a datelor și ca o introducere în analiza avansată a acestora. [25]

De obicei, matricea de corelație este pătratică, cu același număr de linii și de coloane. Linia de 1 începe din colțul din stânga sus, până în dreapta jos, ca diagonală principală, ea arătând că fiecare variabilă se corelează perfect cu ea însăși. Această matrice este simetrică, cu aceeași valoare a corelației deasupra și dedesubtul diagonalei principale, în oglindă, ca în Figura 3.15.

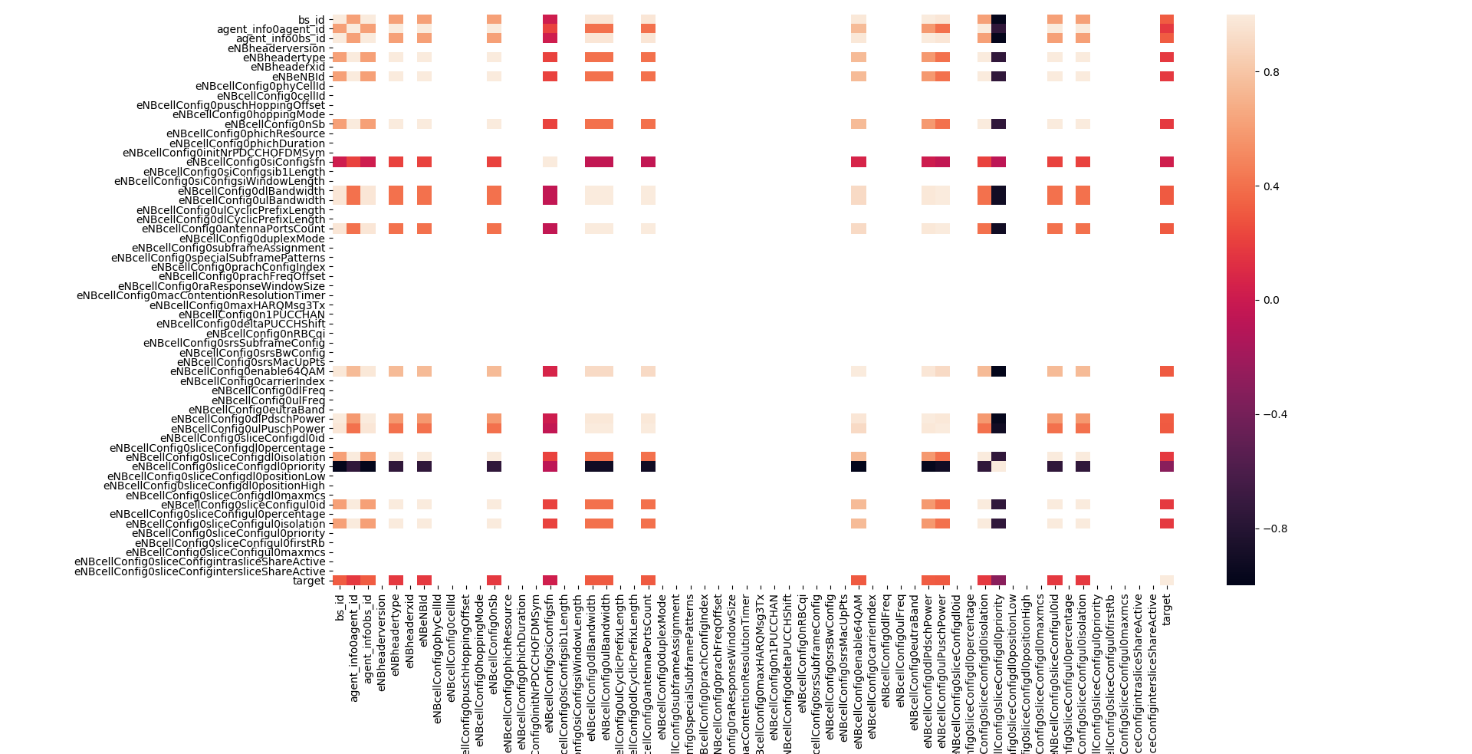


Figura 3.15 Matricea de corelație a câmpurilor din stats\_db

Figura 3.15 este rezultatul următoarelor instrucțiuni:

df = pd.read\_excel("db\_export.xlsx")

df.head()

df.info()

df['target'] = df['y'].apply(lambda x: 1 if x == 'yes' else 0)

df.target.value\_counts()

df.drop('y',axis=1,inplace=True)

df.target.value\_counts()/len(df)

df.describe()

df.dtypes

corr = df.corr()

sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns)

plt.show()

În continuare urmează preprocesarea datelor. Debarasarea datelor (data binning) este o tehnică de preprocesare a datelor utilizată pentrua reduce efectele erorilor minore de observație. Datele originale care se află într-un interval mic sunt înlocuite cu o valoare reprezentativă din acest interval. Este o formă de cuantizare. Acest proces este efectuat de obicei înaintea algoritmului de regresie liniară. Funcțiile de binning sunt:

def mono\_bin(Y, X, n = max\_bin)

def char\_bin(Y, X)

În cadrul funcției mono\_bin (sau monotonic binning) am calculat o serie de medii care tind spre o valoare medie de predicție, iar precizia este dată de influența tuturor parametrilor. Este o funcție de determinare a monotoniei, iar datele sunt grupate în intervale numerice.

Funcțiile de data binning pot fi aplicate și pentru grupuri de caractere. De aceea, în funcția char\_bin, diferit față de cazul anterior este faptul că datele nu mai sunt împărțite în grupuri de câte n date (vezi ANEXA 3), ci după tipul lor, informații stocate în variabila X. Mediile calculate la acest pas sunt stocate în aceeași variabilă ca cele de la pasul anterior.

Algoritmul de binning salvează variabilele de intrare din setul de date și creează un grafic bivariat (analiză bivariată = una dintre cele mai simple analize cantitative care folosește două variabile, de obicei notate cu X și Y) (intrare vs. țintă). De exemplu, pentru Figura 3.16 valorile din partea de jos a graficului reprezintă valoarea de început a intervalelor obținute în urma algoritmului de binning.

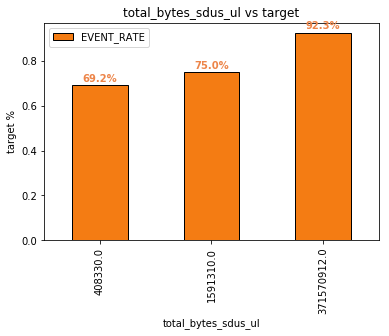


Figura 3.16 Rezultat al preprocesării datelor

În continuare va avea loc procesul de selecție a variabilelor. Acest proces este bazat pe un sistem de votare, așa cum menționam în secțiunea 2.2.1.

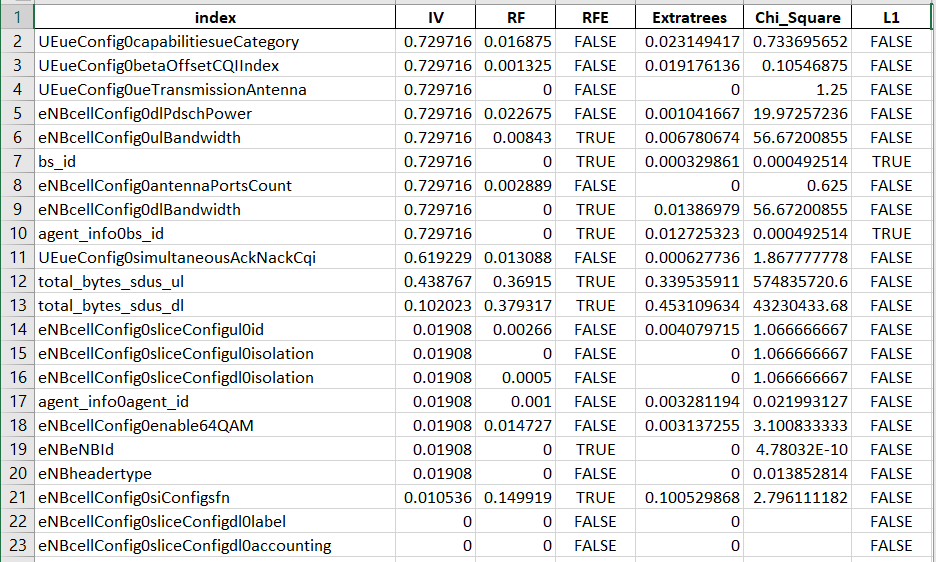
Voi utiliza o serie de algoritmi diferiți (Informații ale variabilelor - IV - Infomation Value, Metoda Pădurilor Aleatoare - Random Forest Classifier, Metoda Arborilor Suplimentari - ExtraTrees Classifier, Patratul Chi - Chi Square) pentru a selecta caracteristici și apoi în final fiecare algoritm votează pentru caracteristica aleasă. Voi exporta rezultatele votului pentru fiecare algoritm în fișiere Excel și apoi le voi îmbina într-un singur tabel în care voi număra voturile pentru a lua decizia finală.

|  |  |
| --- | --- |
| Tabelul 5 Rezultatele votului IV | Tabelul 6 Rezultatele votului Random Forests |

|  |  |
| --- | --- |
| Tabelul 7 Rezultatul votului ExtraTrees | Tabelul 8 Rezultatul votului Chi Square |

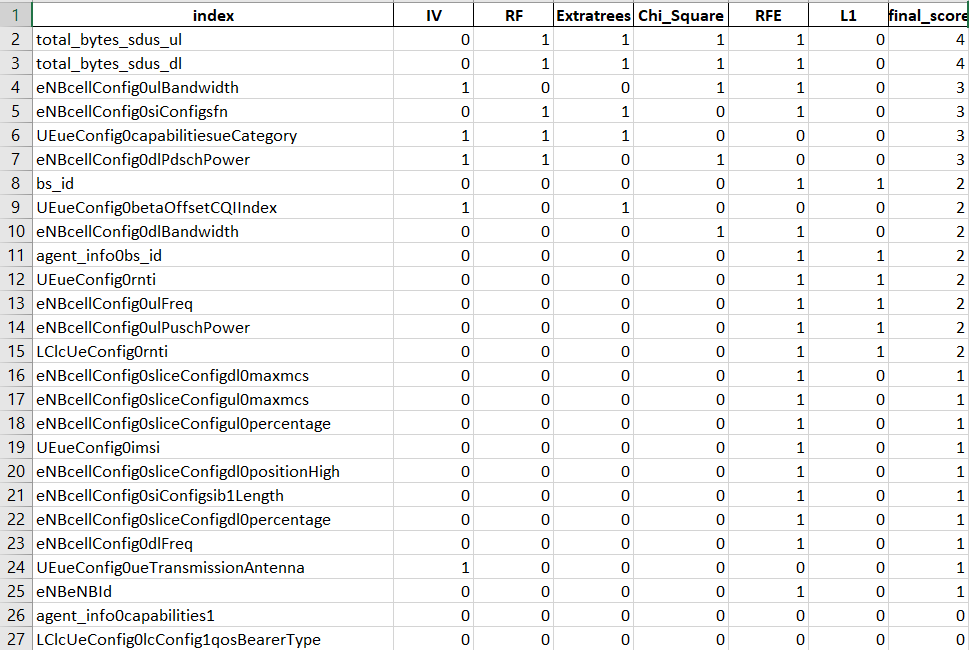
Numărul de voturi finale este folosit pentru a selecta cea mai bună caracteristică pentru modelarea datelor.

Tabelul 9 Rezultatele tuturor voturilor



Mai departe, pentru a clasifica cele mai favorabile rezultate, vom considera pentru calculul scorului final calificativul 1 dacă parametrul se află în top 5 pentru algoritmul respectiv și 0 dacă nu. Acest lucru este detaliat în Tabelul 10.

Tabelul 10 Scorul final



Datele din acest tabel care conține scorul final sunt filtrate după condiția ca scorul să fie mai mare sau egal cu 2 și factorul de varianță a inflației mai mare decât 10. În statistici, factorul de varianță a inflației (VIF) este raportul de variație într-un model cu termeni multipli, împărțit prin varianța unui model cu un singur termen.

features = features[list(score\_table[score\_table['final\_score'] >= 2]['index'])]

vif = calculate\_vif(features)

while vif['VIF'][vif['VIF'] > 10].any():

remove = vif.sort\_values('VIF',ascending=0)['Features'][:1]

features.drop(remove,axis=1,inplace=True)

vif = calculate\_vif(features)

list(vif['Features'])

Rezultatul filtrării este următorul :

['eNBcellConfig0dlBandwidth',

'total\_bytes\_sdus\_ul',

'total\_bytes\_sdus\_dl',

'eNBcellConfig0siConfigsfn',

'eNBcellConfig0ulPuschPower',

'eNBcellConfig0ulFreq',

'LClcUeConfig0rnti',

'eNBcellConfig0dlFreq']

Așadar, acesta este grupul de parametri pe baza cărora se vor efectua predicțiile.

Deși lucrul pare aproape terminat, mai departe va avea loc modelarea datelor. Pentru aceasta voi împărți setul de date în date de învățare și date de test și voi încerca performanțele unei serii de algortimi dintre care îl voi alege pe cel mai precis. Din totalul de date, un procent de 60% va fi dedicat învățării și 40% va fi dedicat testării.

Voi aplica algoritmii pe setul de date de învățare și voi evalua performanța lor pe setul de date de test pentru a mă asigura că modelul este stabil. Am ales spre analiza modelele : Pădurile Aleatoare (Random Forest), Regresie Logistică (Logistic Regression) și Creșterea Variației (Gradient Boosting).

1. Algoritmul pădurilor aleatoare oferă următoarele performanțe :

Precizia pentru datele de învățare: 95.45454545454545 %

Precizia pentru datele de test: 62.5 %

Aria de sub grafic pentru datele de învățare: 100.0 %

Aria de sub grafic pentru datele de test: 41.02564102564103 %

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tabelul 11 Predicții pe datele de învățare   |  |  |  | | --- | --- | --- | | PREZIS | 0 | 1 | | ACTUAL |  |  | | 0 | 5 | 0 | | 1 | 1 | 16 | | Tabelul 12 Predicții pe datele de test   |  |  |  | | --- | --- | --- | | PREZIS | 0 | 1 | | ACTUAL |  |  | | 0 | 1 | 2 | | 1 | 4 | 9 | |

1. Algoritmul creșterii variației oferă următoarele performanțe :

Precizia pentru datele de învățare: 100.0 %

Precizia pentru datele de test: 68.75 %

Aria de sub grafic pentru datele de învățare: 100.0 %

Aria de sub grafic pentru datele de test: 63.63636363636365 %

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tabelul 13 Predicții pe datele de învățare   |  |  |  | | --- | --- | --- | | PREZIS | 0 | 1 | | ACTUAL |  |  | | 0 | 3 | 0 | | 1 | 0 | 19 | | Tabelul 14 Predicții pe datele de test   |  |  |  | | --- | --- | --- | | PREZIS | 0 | 1 | | ACTUAL |  |  | | 0 | 1 | 4 | | 1 | 1 | 10 | |

1. Algoritmul regresiei liniare oferă următoarele performanțe :

Precizia pentru datele de învățare: 86.36363636363636 %

Precizia pentru datele de test: 62.5 %

Aria de sub grafic pentru datele de învățare: 57.89473684210527 %

Aria de sub grafic pentru datele de test: 74.54545454545455 %

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tabelul 15 Predicții pe datele de învățare   |  |  |  | | --- | --- | --- | | PREZIS | 0 | 1 | | ACTUAL |  |  | | 0 | 3 | 0 | | 1 | 0 | 19 | | Tabelul 16 Predicții pe datele de test   |  |  |  | | --- | --- | --- | | PREZIS | 0 | 1 | | ACTUAL |  |  | | 0 | 0 | 5 | | 1 | 1 | 10 | |

În continuare, voi decurge la reglarea performanței pentru primele două modele de algoritmi, iar rezultatele sunt afișate mai jos :

a) Precizia pentru datele de învățare: 86.36363636363636 %

Precizia pentru datele de test: 68.75 %

Aria de sub grafic pentru datele de învățare: 95.36842105263158 %

Aria de sub grafic pentru datele de test: 51.81818181818182 %

1. Precizia pentru datele de învățare: 95.45454545454546 %

Precizia pentru datele de test: 75 %

Aria de sub grafic pentru datele de învățare: 100 %

Aria de sub grafic pentru datele de test: 65.45454545454545 %

Atunci când dorim să alegem cel mai precis algoritm de predicție pentru setul de date de care dispunem, ținem cont de aspectul curbei ROC și de aria graficului delimitat de aceasta.

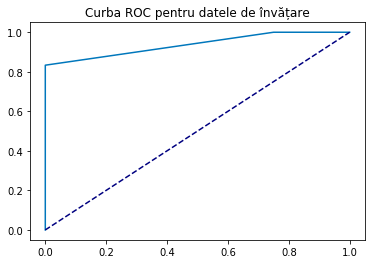


Figura 3.17 Curba ROC - Pădurile aleatoare

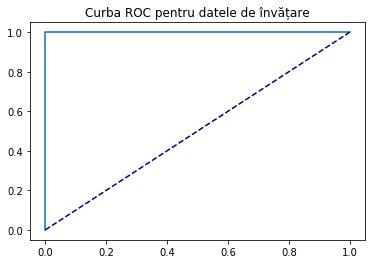


Figura 3.18 Curba ROC – Creșterea variației

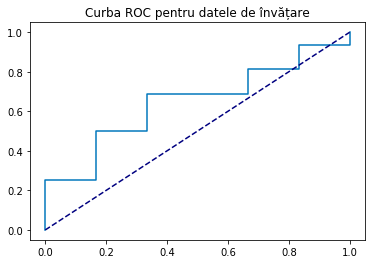


Figura 3.19 Curba ROC – Regresia logistică

Așadar, analizând rezultatele, modelul ales va fi cel al creșterii variației (Gradient Boosting).

### Vizualizarea rezultatelor

Pentru a filtra mai ușor rezultatele pe care dorim să le vizualizăm, am dezvoltat în Qlik Sense o aplicație care importă valorile generate de algoritmii de predicție și le transpune sub formă grafică.

În prima pagină a aplicației vor fi afișate rezultatele voturilor diferiților algoritmi aplicați.

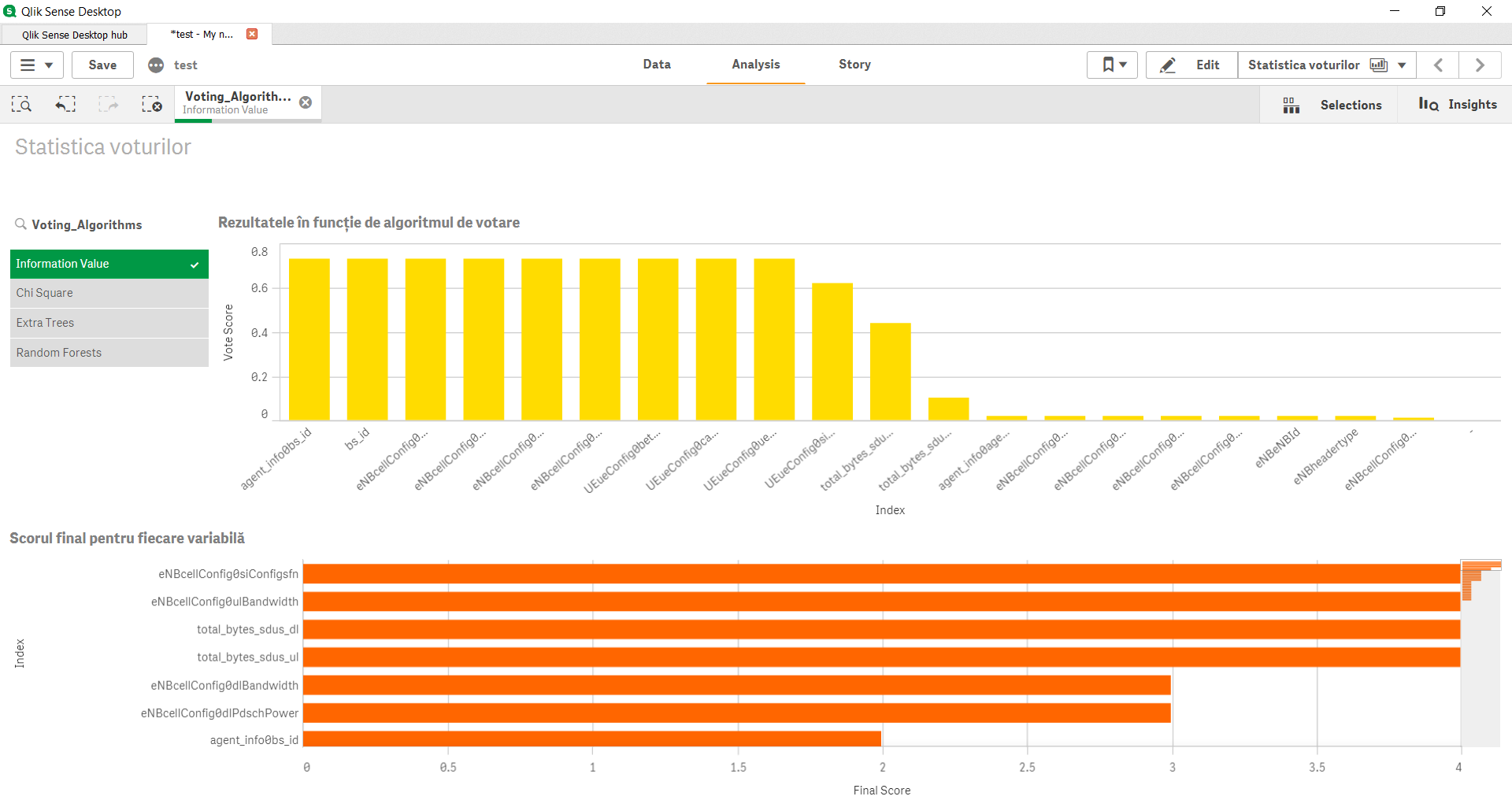


Figura 3.20 Rezultatele votului afișate în Qlik Sense

În jumătatea de sus a paginii vor fi afișate rezultatele pentru fiecare algortim de votare, în funcție de selecția filtrului din stânga și totodată datele vor fi ordonate descrescător. Pentru graficul “Rezultatele în funcție de algoritmul de votare” am ales să exclud valorile nule, pentru a fi scalate mai ușor celelalte valori semnificative, iar în graficul „Scorul final pentru fiecare variabilă” pot fi vizualizate și cele nule, navigând cu bara din partea dreaptă a acestuia.

Tot aici, dacă selectăm unul dintre indecși, vor fi afișate individualizat pentru acesta toate statisticile disponibile în pagină, așa cum urmează în Figura 3.21.

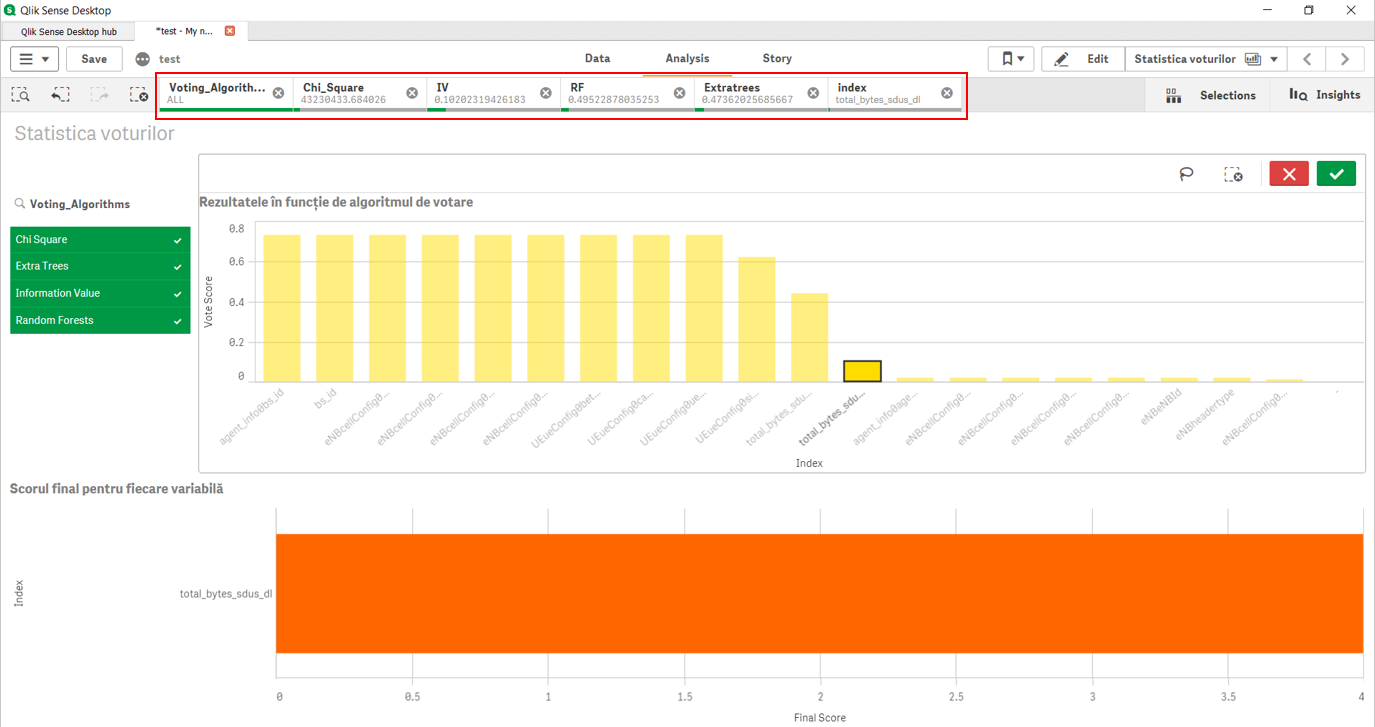


Figura 3.21 Rezultatele filtrate per index

Cu alte cuvinte, rezultatele prezentate anterior sunt cele dinaintea aplicării algortimului de predicție, din etapa de pregătire a datelor. În continuare, voi prezenta rezultatele obținute după aplicarea algoritmului de predicție pe datele de învățare.

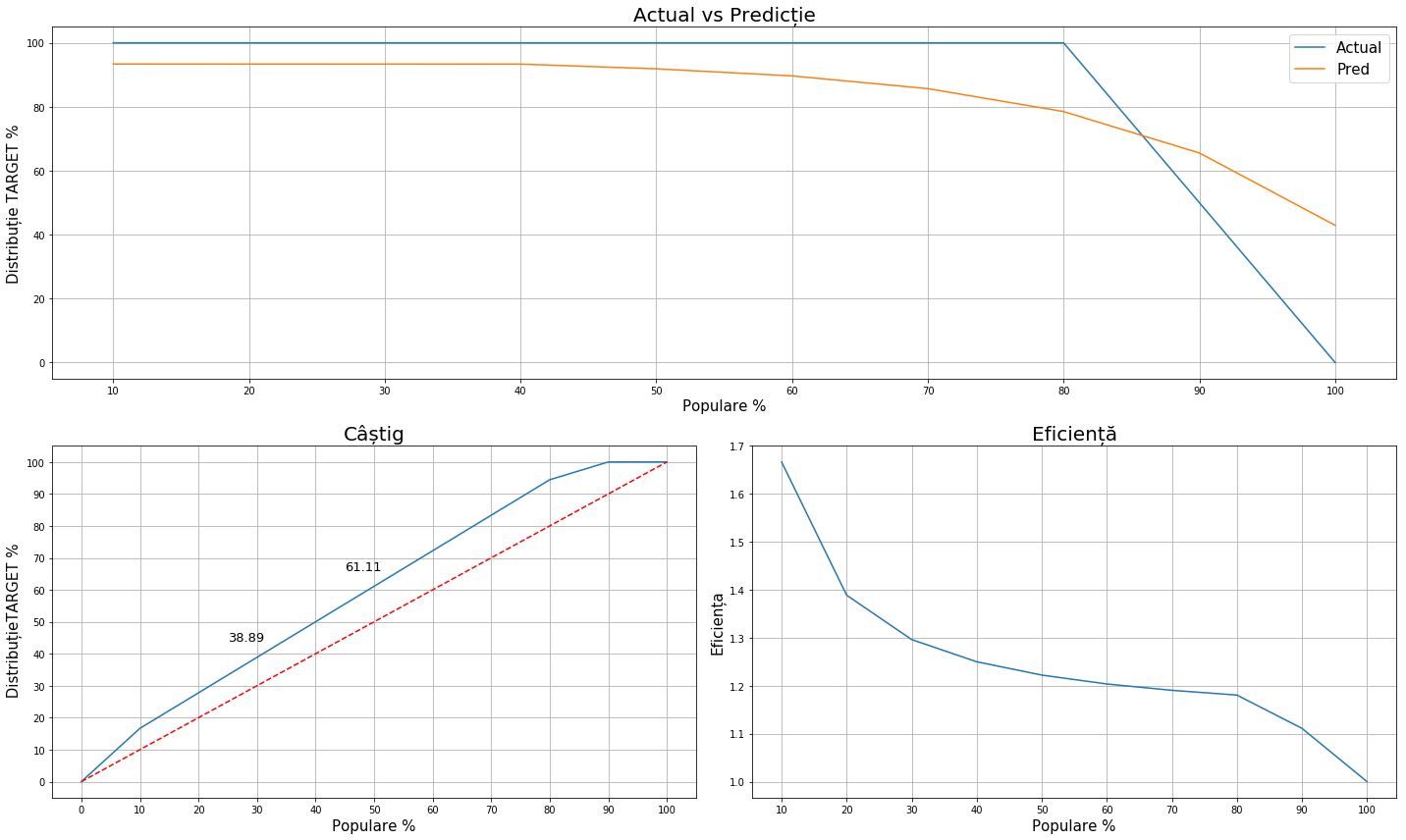


Figura 3.22 Predicții pentru datele de învățare

În graficul comparativ ‘Actual vs Predicție’, curba corespunzătoare comportamentului actual conține înregistrări colectate din rețea la fiecare jumătate de minut (40 de înregistrări), iar predicția se va face pentru următoarea jumătate de minut. Unele dintre rezultatele reale au o distribuție diferită față de cea predictivă din cauza faptului că în algoritmii de predicție sunt incluse unele câmpuri care sunt populate întotdeauna sau care nu au un impact imens asupra distribuției.

Următoarea figură, cea a câștigului arată procentul din numărul de cazuri dintr-o anumită categorie raportat la un procent din numărul total de cazuri. În cazul de față, categoria din care luăm seturi de valori este categoria TARGET. De exemplu, primul punct al curbei (30%, 39%) pentru valorile de 1 ale categoriei TARGET arată că dacă luăm un procent de 30% din date pe baza căruia aplicăm algoritmul de predicție și apoi le sortăm după probabilitatea de a avea 1 ca valoare a câmpului TARGET, ne așteptăm ca printer aceste valori să regăsim aproximativ 39% din toate cazurile care vor lua de fapt valoare 1. În mod similar, primele 50% dintre valori ar conține aproximativ 61% din cazurile care vor lua valoarea 1 a câmpului TARGET.

Linia diagonală este curba de bază. Ea indică faptul că dacă selectăm la întâmplare 10% dintre valorile din setul de date, ne așteptăm să regăsim printre acestea aproximativ 10% dintre toate cazurile care conțin valoarea 1 în câmpul TARGET.

Așadar, ținând cont de aceste două aspecte, putem afirma faptul că vom obține un câștig mai bun cu cât linia curbată se va depărta mai mult de linia de bază.

Diagrama de eficiență este derivată din curba câștigurilor. Valorile de pe axa ‘y’ corespund raportului dintre câștigul cumulativ și linia de bază. Astfel, la 30% eficiența pentru categoria TARGET este de 39% / 30% = 1.30 . Putem spune că această diagramă este un alt mod de a privi informațiile din graficul câștigurilor cumulative.

În continuare, se pot oferi aceleași interpretări pe rezultatele obținute în urma aplicării algoritmului pe datele de test, care arată ca în figura Figura 3.23.

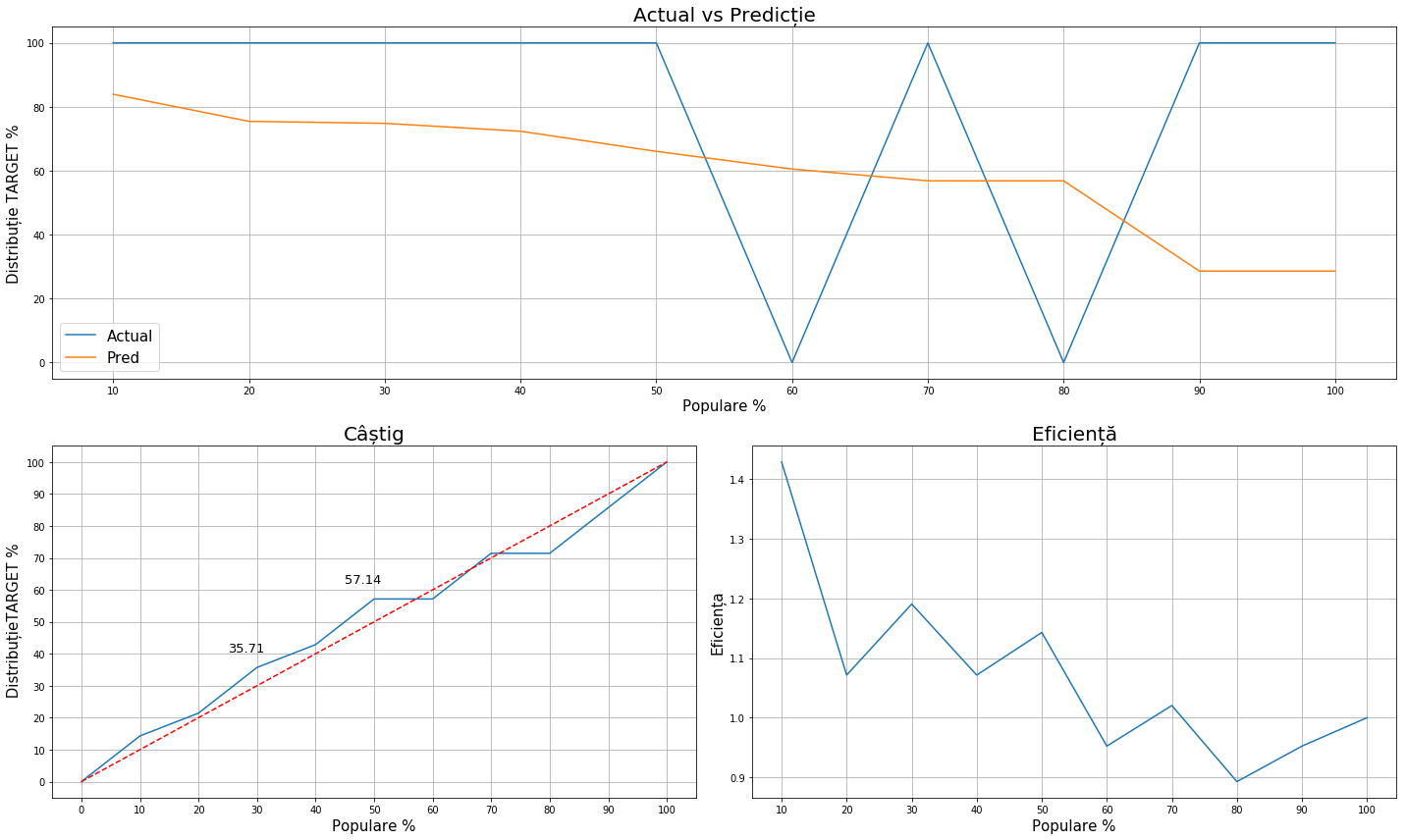


Figura 3.23 Predicțiile pentru datele de test

Într-adevăr, rezultatele obținute în acest caz sunt mai nefavorabile decât cele obținute anterior, însă trebuie să ținem cont și de faptul că datele de test sunt într-un procent de 40% din total, în timp ce datele de învățare reprezintă 60%. Astfel, rezultatele sunt destul de convenabile pentru o cantitate seminificiativă de date. Putem afirma deci, că algoritmul predictiv și-a atins scopul.

# Soluții de îmbunătățire a performanței

După cum am văzut în Capitolul 3, secțiunea 3.2.2, parametri care influențează într-o proporție mare comportamentul rețelei, sunt următorii :

['eNBcellConfig0dlBandwidth',

'total\_bytes\_sdus\_ul',

'total\_bytes\_sdus\_dl',

'eNBcellConfig0siConfigsfn',

'eNBcellConfig0ulPuschPower',

'eNBcellConfig0ulFreq',

'LClcUeConfig0rnti',

'eNBcellConfig0dlFreq']

Mai departe voi prezenta câteva soluții prin care putem obține performanțe mult mai bune ale rețelei, ajustând valorile acestor parametri.

* Când vine vorba despre lățimea de bandă, putem afirma faptul că în aceleași condiții, pentru mai multe dispozitive, capacitatea de descărcare/încărcare din/în rețea este direct proporțională cu lățimea de bandă. Acest fenomen este întâlnit rareori în practică deoarece traficul pe care îl realizează un utilizator în rețea depinde foarte mult de activitățile pe care acesta le întreprinde.

De exemplu, atunci când citim informații de pe wikipedia, cantitatea de bytes de care avem nevoie este una mică, însă atunci când dorim să vizualizăm un videoclip pe youtube la calitate HD, cantitatea de bytes pe care o solicităm pe calea descendentă din rețea este mult mai mare. În cazul acestor activități este posibil ca utilizatorul care accesează site-ul wikipedia să aibă capacitatea de a descărca mult mai multă informație decât o face, iar celui care accesează youtube să nu îi fie suficientă capacitatea de descărcare. În soluționarea acestei probleme vine alocarea lărgimii de bandă și anume mărimi de până la 20 MHz acolo unde traficul creat de utilizator este unul mare, iar mărimi mult mai mici în celălalt caz.

* O altă metodă pentru a îmbunătăți performanța rețelei de date este de a schimba modul de duplexare. În LTE există două moduri de duplexare – FDD și TDD (Frequency Division Duplexing și Time Division Duplexing – Duplexarea cu divizare în frecvență și Duplexarea cu divizare în timp).



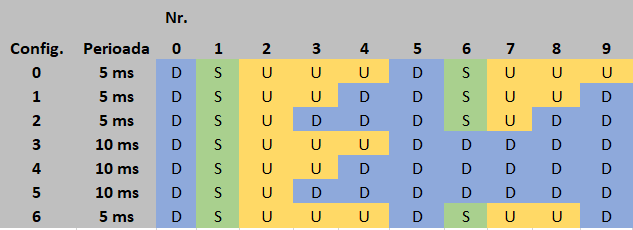
Figura 4.1 Moduri de duplexare în LTE

FDD este un sistem de duplexare întreg (full), spre deosebire de TDD care este un sistem de duplexare înjumătățit (half), ceea ce înseamnă că în cazul FDD va fi posibilă descărcarea și încărcarea în același moment de timp, pe când în TDD se va alege pentru un moment de timp ori încărcarea ori descărcarea. Cu alte cuvinte, căile din FDD împart același moment de timp, iar căile din TDD împart aceeași frecvență.

Pentru a avea o eficiență cât mai bună în gestionarea situațiilor de descărcare/încărcare este recomandat să se utilizeze duplexarea în timp deoarece în acest caz pot fi configurate care dintre intervalele de timp să fie folosite pentru încărcare și care pentru descărcare. De exemplu, dacă într-o companie se utilizează foarte mult procesul de încărcare pe un server, atunci se va folosi o configurare a TDD ce conține mai multe subcadre de încărcare decât pentru descărcare, însă în cazul unui cinematograf în care să presupunem că vizualizare filmelor se va face în mediul online, avem nevoie ca majoritatea cadrelor să fie de tip descărcare.

Posibilele metode de configurare se regăsesc în Tabelul 17.

Tabelul 17 Configurarea TDD



Se observă că în Tabelul 17 pe lângă subcadrele de încărcare (U) și cele de descărcare (D) apare și un cadru S, numit cadru special, care reprezintă perioada de tranziție dintre cadrele U și cadrele D.

* Puterea PUSCH este parametrul prin care îi impunem UE-ului cu ce putere să emită pe canalul de încărcare. Pentru a maximiza capacitatea de încărcare este necesară scăderea interferențelor care este posibilă prin ajustarea puterii PUSCH. Dacă avem o valoare mare a puterii PUSCH, avem și o arie mare de acoperire pentru procesul de încărcare (upload), însă acest aspect nu este tocmai unul favorabil deoarece în acest mod apare un zgomot cât mai mare generat de utilizatorii care se află la o distanță mare de stație și pot fi deserviți de o stație mai apropiată de aceștia. În cazul contrar, atunci când avem o valoare mică a puterii PUSCH este posibil să nu putem deservi toți utilizatorii care se află în apropierea stației. De aceea, puterea PUSCH trebuie aleasă într-un interval în care pot fi deserviți toți utilizatorii arondați stației respective și privați de accesul de încărcare toți utilizatorii care ar introduce zgomot și pot fi deserviți de o altă stație mai apropiată.
* Frecvența este un alt parametru care influențează într-o mare măsură performanța rețelei LTE. De aceea putem lua în discuție alocarea frecvențelor ca o variantă de îmbunătățire a performanței.

Știm că puterea recepționată de UE depinde invers proporțional de frecvență, cu cât frecvența este mai mare, cu atât puterea recepționată este mai mică, însă în aceste condiții nu putem aloca pentru toți utilizatorii frecvența de 800MHz. De aceea, sunt declarate anumite valori de prag pentru care se va utiliza una din valorile 800MHz, 1800MHz, 2600MHz, frecvente corespunzătoare LTE.

Schemele de modulație corespunzătoare acestor frecvențe sunt : QPSK, 16QAM, 64QAM și sunt alocate ca în Figura 4.2.

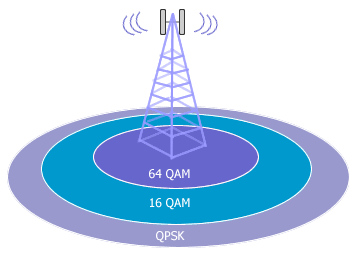


Figura 4.2 Schemele de modulație din LTE

Despre ele se știe că 64QAM oferă o rată de 6biți/s , 16QAM oferă o rată de 4 biți/s, iar QPSK 2 biți/s.

Așadar, pentru utilizatorii din zona cea mai apropiată a stației, deși puterea este una mică, schema de modulație este una mare și astfel viteza de încărcare/descărcare în/din rețea este una mare, iar pentru cei care se află în celelalte 2 zone de lângă stație, cele două viteze vor scădea odată cu creșterea distanței față de stație.

Astfel, ținând cont de cele menționate anterior, performanța rețelei poate fi îmbunătățită prin alegerea corespunzătoare a pragurilor conform cărora are loc alocarea frecvențelor.

###### Concluzii

Scopul proiectului este acela de a introduce și a prezenta noțiuni de analiză predictivă în domeniul comunicațiilor mobile. Așa cum mi-am propus în introducere, în urma modelului predictiv Gradient Boosting aplicat setului de date de care am dispus, am reușit să identific eventualele degradării din rețeaua LTE și parametrii care au influențat aceste fenomene. Gradient Boosting este algoritmul care a oferit cel mai bun compromise tehnologic în comparație cu ceilalți doi algoritmi testați, Random Forests și Logistic Regression.

Lucrarea îmbină trei mari arii ale tehnologiei informației – baze de date, programare și analiza datelor.

Sunt de părere că modul de lucru a fost ușurat prin utilizarea limbajului de programare Python deoarece mi-a permis accesul rapid la datele din baza de date, oferă spre dezvoltare metode deja definite pentru algoritmii predictivi utilizați frecvent și totodată oferă posibilitatea de a accesa rezultatele intermediare ale întregului proces deoarece instrucțiunile sunt executate pe rând, fiind un limbaj de programare interpretat.

Referitor tot la modul de lucru, vizualizarea rezultatelor a fost mult mai prietenoasă prin folosirea utilitarului Qlik Sense, ținând cont de faptul că este creat special pentru aplicații din domeniul științei și analizei datelor.

Modelele de algoritmi predictivi nu se potrivesc oricărui set de date, de aceea este recomandat ca atunci când dorim să aplicăm unul dintre aceste modele peste datele noastre, să analizăm performanțele pe care le oferă mai multe dintre acestea (precizie, câștig, eficiență, curba ROC și aria acoperită de aceasta). Din structura rețelelor de comunicații mobile, parametrii analizați fac parte doar din zona de access a rețelei, acolo unde există un grad de impredictibilitate mai ridicat comparativ cu zona centrală a rețelei.

Printre parametrii care influențează în mod considerabil rețeaua LTE se numără frecvența, lățimea de bandă, cantitatea de informație încărcată(upload)/descărcată(download) de către utilizator și puterea recepționată de echipamentul utilizatorului. Factorii ce aduc un grad de impredictibilitate mai ridicat sunt determinați de comportamentul terminalelor mobile, astfel că prin introducerea unui set de algoritmi de predicție pot fi identificate stații de bază care au fost supra sau sub dimensionate la nivelul resurselor, astfel încât acestea din urmă se pot optimiza într-un mod corespunzător.

Introducerea unui set mai larg al algoritmilor de predicție în analiza comunicațiilor mobile deschide un orizont al aplicațiilor pentru cea de-a cincea tehnologie și oferă unelte de optimizare de resurse atât la nivelul fizic, de exemplu consum de putere, cât și la nivelul eficienței spectrale, de exemplu a fost observată în cazul comportamentul terminalelor mobile o tendință de migrare sezonieră a acestora, astfel că un operator de telecomunicații poate utiliza blocuri de frecvență mai mici în zona de destinație a terminalelor mobile micșorând blocurile de frecvență alocate în zona de unde aceștia au plecat sau de reședință. Prin zonă de reședinta, se face referire la aria de deplasare a terminalului mobil cu o perioadă de activitate mai mare de 4 luni.

Din punctul de vedere al experienței pe care o are utilizatorul din partea rețelei, trebuie să ne gândim la faptul că soluțiile de evitare a congestiei trebuie implementate mai ales în timpul orelor celor mai aglomerate din zi deoarece traficul urmărește zilnic aproximativ același model din punct de vedere cantitativ.

Dacă o să am posibilitatea, îmi doresc ca pe viitor să adaptez modelul în așa fel încât să trateze seturi mult mai mari de date și să fac posibilă implementarea soluțiilor de îmbunătățire în timp cât se poate de real deoarece este ușor de intuit că în continua schimbare în care ne aflăm din punctul de vedere al evoluției tehnologiei, se urmărește automatizarea a din ce în ce mai multe procese.

###### Bibliografie

|  |  |  |
| --- | --- | --- |
| [1] |  | <https://study.com/academy/lesson/mobile-networking-definition-components-comparison.html> (accesat la data: 26.02.2019) |

[2] <https://www.tnuda.org.il/en/physics-radiation/radio-frequency-rf-radiation/cellular-communication-network-technologies> (accesat la data: 26.02.2019)

[3] <https://www.researchgate.net/figure/Frequency-reuse-3-model-in-GSM_fig1_282601918> (accesat la data: 03.03.2019)

[4] Upkar Varshney, Georgia State University, “4G Wireless Networks”, în *IT Pro*/Septembrie-Octombrie 2012

[5] <https://www.tutorialspoint.com/cdma> (accesat la data: 13.03.2019)

[6] G. Brindha, “Comparison of PAPR Analysis for OFDMA and SC-FDMA in LTE Systems”, în *International Journal Engineering Innovation & Research*, Volume 2, Issue 2

[7] Rakesh Kumar Singh, Ranjan Singh, “4G LTE Cellular Technology: Network Architecture and Mobile Standards”, în *International Journal of Emerging Research in Management & Technology*/ Decembrie 2016

[8] Prof. Bogdan Mocanu, Cursul de Rețele de Comunicații, UPB, 2018

[9]<http://www.cse.unt.edu/~rdantu/FALL_2013_WIRELESS_NETWORKS/LTE_Alcatel_White_Paper.pdf> (accesat la data : 17.03.2019)

[10] Prof. Bogdan Mocanu, Cursul de Baze de Date, UPB, 2018

[11] <https://forum.huawei.com/enterprise/en/Downlink-Power-Allocation-in-LTE/thread/457683-100305> (accesat la data 16.06.2019)

[12] Bilal Muhammad and Abbas Mohammed, Physical Uplink SharedChannel (PUSCH)Closed-Loop PowerControl for 3G LTE, Ianuarie 2010

[13] <https://dsp.stackexchange.com/questions/38256/system-frame-number-in-lte-specifications> (accesat la data: 16.06.2019)

[14] <http://www.telecomabc.com/i/imsi.html> (accesat la data: 16.06.2019)

[15] <http://howltestuffworks.blogspot.com/2014/06/rntis-in-lte.html> (accesat la data: 16.06.2019)

[16] Bishop, C. M., Pattern Recognition and Machine Learning, Springer, 2006

[17] <https://www.datacamp.com/community/tutorials/random-forests-classifier-python> (accesat la data: 06.04.2019)

[18] <https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d> (accesat la data: 28.06.2019)

[19] <https://www.geeksforgeeks.org/understanding-logistic-regression/> (accesat la data: 12.06.2019)

[20] <https://mwc.gr/presentations/2017/konstantinou.pdf> (accesat la data: 16.06.2019)

[21] <https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7> (accesat la data : 31.03.2019)

[22] <https://dev.mysql.com/doc/workbench/en/> (accesat la data: 24.04.2019)

[23] <https://docs.spyder-ide.org/> (accesat la data: 11.05.2019)

[24] <https://help.qlik.com/en-US/sense/April2019/Content/Sense_Helpsites/Home.htm> (accesat la data: 03.06.2019)

[25] <https://www.displayr.com/what-is-a-correlation-matrix/> (accesat la data: 16.05.2019)

create database stats\_db;

use stats\_db;

drop table mytable;

CREATE TABLE mytable(

bs\_id BIGINT

,agent\_info0agent\_id BIGINT

,agent\_info0bs\_id BIGINT

,agent\_info0capabilities0 VARCHAR(5)

,agent\_info0capabilities1 VARCHAR(5)

,agent\_info0capabilities2 VARCHAR(5)

,agent\_info0capabilities3 VARCHAR(5)

,agent\_info0capabilities4 VARCHAR(3)

,agent\_info0capabilities5 VARCHAR(4)

,agent\_info0capabilities6 VARCHAR(4)

,agent\_info0capabilities7 VARCHAR(3)

,eNBheaderversion BIT

,eNBheadertype BIGINT

,eNBheaderxid BIT

,eNBeNBId BIGINT

,eNBcellConfig0phyCellId BIT

,eNBcellConfig0cellId BIT

,eNBcellConfig0puschHoppingOffset BIT

,eNBcellConfig0hoppingMode BIT

,eNBcellConfig0nSb BIT

,eNBcellConfig0phichResource BIT

,eNBcellConfig0phichDuration BIT

,eNBcellConfig0initNrPDCCHOFDMSym BIT

,eNBcellConfig0siConfigsfn BIGINT

,eNBcellConfig0siConfigsib1Length BIGINT

,eNBcellConfig0siConfigsiWindowLength BIGINT

,eNBcellConfig0dlBandwidth BIGINT

,eNBcellConfig0ulBandwidth BIGINT

,eNBcellConfig0ulCyclicPrefixLength BIGINT

,eNBcellConfig0dlCyclicPrefixLength BIGINT

,eNBcellConfig0antennaPortsCount BIGINT

,eNBcellConfig0duplexMode BIT

,eNBcellConfig0subframeAssignment BIT

,eNBcellConfig0specialSubframePatterns BIT

,eNBcellConfig0prachConfigIndex BIT

,eNBcellConfig0prachFreqOffset BIGINT

,eNBcellConfig0raResponseWindowSize BIGINT

,eNBcellConfig0macContentionResolutionTimer BIGINT

,eNBcellConfig0maxHARQMsg3Tx BIT

,eNBcellConfig0n1PUCCHAN BIT

,eNBcellConfig0deltaPUCCHShift BIT

,eNBcellConfig0nRBCqi BIT

,eNBcellConfig0srsSubframeConfig BIT

,eNBcellConfig0srsBwConfig BIT

,eNBcellConfig0srsMacUpPts BIT

,eNBcellConfig0enable64QAM BIGINT

,eNBcellConfig0carrierIndex BIT

,eNBcellConfig0dlFreq BIGINT

,eNBcellConfig0ulFreq BIGINT

,eNBcellConfig0eutraBand BIGINT

,eNBcellConfig0dlPdschPower BIGINT

,eNBcellConfig0ulPuschPower BIGINT

,eNBcellConfig0sliceConfigdl0id BIGINT

,eNBcellConfig0sliceConfigdl0label VARCHAR(4)

,eNBcellConfig0sliceConfigdl0percentage BIGINT

,eNBcellConfig0sliceConfigdl0isolation VARCHAR(5)

,eNBcellConfig0sliceConfigdl0priority BIGINT

,eNBcellConfig0sliceConfigdl0positionLow BIT

,eNBcellConfig0sliceConfigdl0positionHigh BIGINT

,eNBcellConfig0sliceConfigdl0maxmcs BIGINT

,eNBcellConfig0sliceConfigdl0sorting0 VARCHAR(8)

,eNBcellConfig0sliceConfigdl0sorting1 VARCHAR(8)

,eNBcellConfig0sliceConfigdl0sorting2 VARCHAR(6)

,eNBcellConfig0sliceConfigdl0sorting3 VARCHAR(5)

,eNBcellConfig0sliceConfigdl0sorting4 VARCHAR(6)

,eNBcellConfig0sliceConfigdl0sorting5 VARCHAR(6)

,eNBcellConfig0sliceConfigdl0accounting VARCHAR(8)

,eNBcellConfig0sliceConfigdl0schedulerName VARCHAR(16)

,eNBcellConfig0sliceConfigul0id BIT

,eNBcellConfig0sliceConfigul0label VARCHAR(4)

,eNBcellConfig0sliceConfigul0percentage BIGINT

,eNBcellConfig0sliceConfigul0isolation VARCHAR(5)

,eNBcellConfig0sliceConfigul0priority BIT

,eNBcellConfig0sliceConfigul0firstRb BIT

,eNBcellConfig0sliceConfigul0maxmcs BIGINT

,eNBcellConfig0sliceConfigul0accounting VARCHAR(9)

,eNBcellConfig0sliceConfigul0schedulerName VARCHAR(19)

,eNBcellConfig0sliceConfigintrasliceShareActive VARCHAR(4)

,eNBcellConfig0sliceConfigintersliceShareActive VARCHAR(4)

,UEueConfig0rnti BIGINT

,UEueConfig0timeAlignmentTimer BIGINT

,UEueConfig0transmissionMode BIT

,UEueConfig0ueAggregatedMaxBitrateUL BIT

,UEueConfig0ueAggregatedMaxBitrateDL BIT

,UEueConfig0capabilitieshalfDuplex BIT

,UEueConfig0capabilitiesintraSFHopping BIT

,UEueConfig0capabilitiestype2Sb1 BIT

,UEueConfig0capabilitiesueCategory BIGINT

,UEueConfig0capabilitiesresAllocType1 BIT

,UEueConfig0ueTransmissionAntenna BIGINT

,UEueConfig0ttiBundling BIT

,UEueConfig0maxHARQTx BIGINT

,UEueConfig0betaOffsetACKIndex BIT

,UEueConfig0betaOffsetRIIndex BIT

,UEueConfig0betaOffsetCQIIndex BIGINT

,UEueConfig0ackNackSimultaneousTrans BIT

,UEueConfig0simultaneousAckNackCqi BIT

,UEueConfig0aperiodicCqiRepMode BIGINT

,UEueConfig0ackNackRepetitionFactor BIT

,UEueConfig0pcellCarrierIndex BIT

,UEueConfig0imsi BIGINT

,UEueConfig0dlSliceId BIT

,UEueConfig0ulSliceId BIT

,LCheaderversion BIT

,LCheadertype BIGINT

,LCheaderxid BIT

,LClcUeConfig0rnti BIGINT

,LClcUeConfig0lcConfig0lcid BIT

,LClcUeConfig0lcConfig0lcg BIT

,LClcUeConfig0lcConfig0direction BIGINT

,LClcUeConfig0lcConfig0qosBearerType BIT

,LClcUeConfig0lcConfig0qci BIT

,LClcUeConfig0lcConfig1lcid BIGINT

,LClcUeConfig0lcConfig1lcg BIT

,LClcUeConfig0lcConfig1direction BIGINT

,LClcUeConfig0lcConfig1qosBearerType BIT

,LClcUeConfig0lcConfig1qci BIT

,LClcUeConfig0lcConfig2lcid BIGINT

,LClcUeConfig0lcConfig2lcg BIT

,LClcUeConfig0lcConfig2direction BIT

,LClcUeConfig0lcConfig2qosBearerType BIT

,LClcUeConfig0lcConfig2qci BIT

,total\_bytes\_sdus\_ul BIGINT

,total\_bytes\_sdus\_dl BIGINT

,y VARCHAR(3)

);

INSERT INTO mytable (`bs\_id`,`agent\_info0agent\_id`,`agent\_info0bs\_id`,`agent\_info0capabilities0`,`agent\_info0capabilities1`,`agent\_info0capabilities2`,`agent\_info0capabilities3`,`agent\_info0capabilities4`,`agent\_info0capabilities5`,`agent\_info0capabilities6`,`agent\_info0capabilities7`,`eNBheaderversion`,`eNBheadertype`,`eNBheaderxid`,`eNBeNBId`,`eNBcellConfig0phyCellId`,`eNBcellConfig0cellId`,`eNBcellConfig0puschHoppingOffset`,`eNBcellConfig0hoppingMode`,`eNBcellConfig0nSb`,`eNBcellConfig0phichResource`,`eNBcellConfig0phichDuration`,`eNBcellConfig0initNrPDCCHOFDMSym`,`eNBcellConfig0siConfigsfn`,`eNBcellConfig0siConfigsib1Length`,`eNBcellConfig0siConfigsiWindowLength`,`eNBcellConfig0dlBandwidth`,`eNBcellConfig0ulBandwidth`,`eNBcellConfig0ulCyclicPrefixLength`,`eNBcellConfig0dlCyclicPrefixLength`,`eNBcellConfig0antennaPortsCount`,`eNBcellConfig0duplexMode`,`eNBcellConfig0subframeAssignment`,`eNBcellConfig0specialSubframePatterns`,`eNBcellConfig0prachConfigIndex`,`eNBcellConfig0prachFreqOffset`,`eNBcellConfig0raResponseWindowSize`,`eNBcellConfig0macContentionResolutionTimer`,`eNBcellConfig0maxHARQMsg3Tx`,`eNBcellConfig0n1PUCCHAN`,`eNBcellConfig0deltaPUCCHShift`,`eNBcellConfig0nRBCqi`,`eNBcellConfig0srsSubframeConfig`,`eNBcellConfig0srsBwConfig`,`eNBcellConfig0srsMacUpPts`,`eNBcellConfig0enable64QAM`,`eNBcellConfig0carrierIndex`,`eNBcellConfig0dlFreq`,`eNBcellConfig0ulFreq`,`eNBcellConfig0eutraBand`,`eNBcellConfig0dlPdschPower`,`eNBcellConfig0ulPuschPower`,`eNBcellConfig0sliceConfigdl0id`,`eNBcellConfig0sliceConfigdl0label`,`eNBcellConfig0sliceConfigdl0percentage`,`eNBcellConfig0sliceConfigdl0isolation`,`eNBcellConfig0sliceConfigdl0priority`,`eNBcellConfig0sliceConfigdl0positionLow`,`eNBcellConfig0sliceConfigdl0positionHigh`,`eNBcellConfig0sliceConfigdl0maxmcs`,`eNBcellConfig0sliceConfigdl0sorting0`,`eNBcellConfig0sliceConfigdl0sorting1`,`eNBcellConfig0sliceConfigdl0sorting2`,`eNBcellConfig0sliceConfigdl0sorting3`,`eNBcellConfig0sliceConfigdl0sorting4`,`eNBcellConfig0sliceConfigdl0sorting5`,`eNBcellConfig0sliceConfigdl0accounting`,`eNBcellConfig0sliceConfigdl0schedulerName`,`eNBcellConfig0sliceConfigul0id`,`eNBcellConfig0sliceConfigul0label`,`eNBcellConfig0sliceConfigul0percentage`,`eNBcellConfig0sliceConfigul0isolation`,`eNBcellConfig0sliceConfigul0priority`,`eNBcellConfig0sliceConfigul0firstRb`,`eNBcellConfig0sliceConfigul0maxmcs`,`eNBcellConfig0sliceConfigul0accounting`,`eNBcellConfig0sliceConfigul0schedulerName`,`eNBcellConfig0sliceConfigintrasliceShareActive`,`eNBcellConfig0sliceConfigintersliceShareActive`,`UEueConfig0rnti`,`UEueConfig0timeAlignmentTimer`,`UEueConfig0transmissionMode`,`UEueConfig0ueAggregatedMaxBitrateUL`,`UEueConfig0ueAggregatedMaxBitrateDL`,`UEueConfig0capabilitieshalfDuplex`,`UEueConfig0capabilitiesintraSFHopping`,`UEueConfig0capabilitiestype2Sb1`,`UEueConfig0capabilitiesueCategory`,`UEueConfig0capabilitiesresAllocType1`,`UEueConfig0ueTransmissionAntenna`,`UEueConfig0ttiBundling`,`UEueConfig0maxHARQTx`,`UEueConfig0betaOffsetACKIndex`,`UEueConfig0betaOffsetRIIndex`,`UEueConfig0betaOffsetCQIIndex`,`UEueConfig0ackNackSimultaneousTrans`,`UEueConfig0simultaneousAckNackCqi`,`UEueConfig0aperiodicCqiRepMode`,`UEueConfig0ackNackRepetitionFactor`,`UEueConfig0pcellCarrierIndex`,`UEueConfig0imsi`,`UEueConfig0dlSliceId`,`UEueConfig0ulSliceId`,`LCheaderversion`,`LCheadertype`,`LCheaderxid`,`LClcUeConfig0rnti`,`LClcUeConfig0lcConfig0lcid`,`LClcUeConfig0lcConfig0lcg`,`LClcUeConfig0lcConfig0direction`,`LClcUeConfig0lcConfig0qosBearerType`,`LClcUeConfig0lcConfig0qci`,`LClcUeConfig0lcConfig1lcid`,`LClcUeConfig0lcConfig1lcg`,`LClcUeConfig0lcConfig1direction`,`LClcUeConfig0lcConfig1qosBearerType`,`LClcUeConfig0lcConfig1qci`,`LClcUeConfig0lcConfig2lcid`,`LClcUeConfig0lcConfig2lcg`,`LClcUeConfig0lcConfig2direction`,`LClcUeConfig0lcConfig2qosBearerType`,`LClcUeConfig0lcConfig2qci`,`total\_bytes\_sdus\_ul`,`total\_bytes\_sdus\_dl`,`y`) VALUES (10005,5,10005,'LOPHY','HIPHY','LOMAC','HIMAC','RLC','PDCP','SDAP','RRC',0,8,0,234881024,0,0,0,0,1,0,0,1,149,17,5,25,25,0,0,1,1,0,0,0,2,7,5,0,0,1,0,0,0,0,0,0,2685,2565,7,-27,-96,0,'xMBB',100,'FALSE',10,0,25,28,'CR\_ROUND','CR\_SRB12','CR\_HOL','CR\_LC','CR\_CQI','CR\_LCP','POL\_FAIR','schedule\_ue\_spec',0,'xMBB',100,'FALSE',0,0,20,'POLU\_FAIR','schedule\_ulsch\_rnti','TRUE','TRUE',5268,7,0,0,0,0,1,1,4,1,2,0,4,0,0,8,0,0,3,0,0,209000000000000,0,0,0,12,0,5268,1,0,2,0,1,2,0,2,0,1,3,1,1,0,1,2181380,389398,'yes');

.

.

.

.

.

.

describe mytable;

data\_extract.py

################################################################################

#

# Descriere: Pentru a colecta datele necesare algoritmului analitic este nevoie

# de un export in format xslx a tabelului din baza de date

# Acest cod trateaza exportul.

#

################################################################################

import os

import MySQLdb as dbapi

import pandas as pd

cd = os.path.dirname(os.path.abspath(\_\_file\_\_))

# Deschiderea conexiunea la baza de date

db = dbapi.connect(host='localhost',user='root',passwd='root', db='stats\_db')

cur = db.cursor()

# Obtinerea tabelelor

cur.execute("SHOW TABLES;")

tables = cur.fetchall()

for t in tables:

columns = []

# Importarea datelor intr-un model

df = pd.read\_sql("SELECT \* FROM {0};".format(t[0]), db)

# Exportarea modelului in fisier de tipul XLSX

df.to\_excel(os.path.join(cd, '{0}.xlsx'.format('db\_export')), index=False)

# Inchiderea cursorului si a conexiunii la baza de date

cur.close()

db.close()

################################################################################

###################################EOF########################################

data\_prediction.py

import pandas as pd

import numpy as np

import os

######################################################################

#

#

######################################################################

# încărcarea setului de date

cd = os.path.dirname(os.path.abspath('\_\_file\_\_'))

df = pd.read\_excel("db\_export.xlsx")

df.head()

df.info()

######################################################################

# transformarea datelor

# acum datele sunt deja stocate (în dataframe) si mai departe vom crea coloana

# target care va contine doar valori de 1 și 0, functie de coloana y

df['target'] = df['y'].apply(lambda x: 1 if x == 'yes' else 0)

# ultima coloana reprezinta coloana target, considerand că în analiză vom lua în

# considerare doar valorile rezultate atunci cand telefonul este conectat la rețea

# S-ar putea sa existe exporturi de parametri atunci cand nu exista un echipament tinta.

# In acest caz, vom elimina aceste inregistrari deoarece nu sunt de interes.

#

df.target.value\_counts()

df.drop('y',axis=1,inplace=True)

#Stare descriptiva

df.target.value\_counts()/len(df)

df.describe()

#df.dtypes.to\_excel(os.path.join(cd, '{0}.xlsx'.format('types')), index=False)

import seaborn as sns #pip install seaborn

import matplotlib.pyplot as plt #pip install matplotlib

corr = df.corr()

sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns)

plt.show() #afiseaza harta matricii de corelatie - sunt afisate 50 de campuri deoarece

#atatea campuri am de tip diferit de 'obiect'

# Selectia variabilelor

# Transformarea datelor

# Conversia campurilor de tip obiect in numerice

from sklearn import preprocessing

from collections import defaultdict

d = defaultdict(preprocessing.LabelEncoder)

fit = df.select\_dtypes(include=['object']).fillna('NA').apply(lambda x: d[x.name].fit\_transform(x))

#Convert the categorical columns based on encoding

for i in list(d.keys()):

df[i] = d[i].transform(df[i].fillna('NA'))

features = df[df.columns.difference(['target'])]

labels = df['target']

features = features.fillna(0)

#features.to\_excel(os.path.join(cd, '{0}.xlsx'.format('features')), index=False)

import pandas as pd

import pandas.core.algorithms as algos

from pandas import Series

import scipy.stats.stats as stats

import re

import traceback

max\_bin = 20

force\_bin = 3

# Definirea functiilor de debarasare (binning functions)

def mono\_bin(Y, X, n = max\_bin):

df1 = pd.DataFrame({"X": X, "Y": Y})

justmiss = df1[['X','Y']][df1.X.isnull()]

notmiss = df1[['X','Y']][df1.X.notnull()]

r = 0

while np.abs(r) < 1:

try:

d1 = pd.DataFrame({"X": notmiss.X, "Y": notmiss.Y, "Bucket": pd.qcut(notmiss.X, n)})

# pd.qcut(notmiss.X, n) împarte în grupuri de cate n valori, valorile diferite de 0 ale lui X

d2 = d1.groupby('Bucket', as\_index=True)

r, p = stats.spearmanr(d2.mean().X, d2.mean().Y)

# spearman reprezinta functia de corelatie cu acelasi nume si este

# o masura neparametrica a monotoniei relatiei dintre doua seturi de date

# valoarea lui p indica probabilitatea ca un sistem necorelat sa proca seturi

# de date care au corelatia Spearman la fel de extrema ca cea compusa din aceste seturi de date

n = n - 1

except Exception:

n = n - 1

if len(d2) == 1:

n = force\_bin

bins = algos.quantile(notmiss.X, np.linspace(0, 1, n))

if len(np.unique(bins)) == 2:

bins = np.insert(bins, 0, 1)

bins[1] = bins[1]-(bins[1]/2)

d1 = pd.DataFrame({"X": notmiss.X, "Y": notmiss.Y, "Bucket": pd.cut(notmiss.X, np.unique(bins),include\_lowest=True)})

d2 = d1.groupby('Bucket', as\_index=True)

d3 = pd.DataFrame({},index=[])

d3["MIN\_VALUE"] = d2.min().X

d3["MAX\_VALUE"] = d2.max().X

d3["COUNT"] = d2.count().Y

d3["EVENT"] = d2.sum().Y

d3["NONEVENT"] = d2.count().Y - d2.sum().Y

d3=d3.reset\_index(drop=True)

if len(justmiss.index) > 0:

d4 = pd.DataFrame({'MIN\_VALUE':np.nan},index=[0])

d4["MAX\_VALUE"] = np.nan

d4["COUNT"] = justmiss.count().Y

d4["EVENT"] = justmiss.sum().Y

d4["NONEVENT"] = justmiss.count().Y - justmiss.sum().Y

d3 = d3.append(d4,ignore\_index=True)

d3["EVENT\_RATE"] = d3.EVENT/d3.COUNT

d3["NON\_EVENT\_RATE"] = d3.NONEVENT/d3.COUNT

d3["DIST\_EVENT"] = d3.EVENT/d3.sum().EVENT

d3["DIST\_NON\_EVENT"] = d3.NONEVENT/d3.sum().NONEVENT

d3["WOE"] = np.log(d3.DIST\_EVENT/d3.DIST\_NON\_EVENT)

# woe - weight of evidence = metodă de evaluare a predictorilor

d3["IV"] = (d3.DIST\_EVENT-d3.DIST\_NON\_EVENT)\*np.log(d3.DIST\_EVENT/d3.DIST\_NON\_EVENT)

# IV - information value

d3["VAR\_NAME"] = "VAR"

d3 = d3[['VAR\_NAME','MIN\_VALUE', 'MAX\_VALUE', 'COUNT', 'EVENT', 'EVENT\_RATE', 'NONEVENT', 'NON\_EVENT\_RATE', 'DIST\_EVENT','DIST\_NON\_EVENT','WOE', 'IV']]

d3 = d3.replace([np.inf, -np.inf], 0)

d3.IV = d3.IV.sum()

return(d3)

def char\_bin(Y, X):

df1 = pd.DataFrame({"X": X, "Y": Y})

justmiss = df1[['X','Y']][df1.X.isnull()]

notmiss = df1[['X','Y']][df1.X.notnull()]

df2 = notmiss.groupby('X',as\_index=True)

d3 = pd.DataFrame({},index=[])

d3["COUNT"] = df2.count().Y

d3["MIN\_VALUE"] = df2.sum().Y.index

d3["MAX\_VALUE"] = d3["MIN\_VALUE"]

d3["EVENT"] = df2.sum().Y

d3["NONEVENT"] = df2.count().Y - df2.sum().Y

if len(justmiss.index) > 0:

d4 = pd.DataFrame({'MIN\_VALUE':np.nan},index=[0])

d4["MAX\_VALUE"] = np.nan

d4["COUNT"] = justmiss.count().Y

d4["EVENT"] = justmiss.sum().Y

d4["NONEVENT"] = justmiss.count().Y - justmiss.sum().Y

d3 = d3.append(d4,ignore\_index=True)

d3["EVENT\_RATE"] = d3.EVENT/d3.COUNT

d3["NON\_EVENT\_RATE"] = d3.NONEVENT/d3.COUNT

d3["DIST\_EVENT"] = d3.EVENT/d3.sum().EVENT

d3["DIST\_NON\_EVENT"] = d3.NONEVENT/d3.sum().NONEVENT

d3["WOE"] = np.log(d3.DIST\_EVENT/d3.DIST\_NON\_EVENT)

d3["IV"] = (d3.DIST\_EVENT-d3.DIST\_NON\_EVENT)\*np.log(d3.DIST\_EVENT/d3.DIST\_NON\_EVENT)

d3["VAR\_NAME"] = "VAR"

d3 = d3[['VAR\_NAME','MIN\_VALUE', 'MAX\_VALUE', 'COUNT', 'EVENT', 'EVENT\_RATE', 'NONEVENT', 'NON\_EVENT\_RATE', 'DIST\_EVENT','DIST\_NON\_EVENT','WOE', 'IV']]

d3 = d3.replace([np.inf, -np.inf], 0)

d3.IV = d3.IV.sum()

d3 = d3.reset\_index(drop=True)

return(d3)

def data\_vars(df1, target):

stack = traceback.extract\_stack()

filename, lineno, function\_name, code = stack[-2]

vars\_name = re.compile(r'\((.\*?)\).\*$').search(code).groups()[0]

final = (re.findall(r"[\w']+", vars\_name))[-1]

x = df1.dtypes.index

count = -1

for i in x:

if i.upper() not in (final.upper()):

if np.issubdtype(df1[i], np.number) and len(Series.unique(df1[i])) > 2:

conv = mono\_bin(target, df1[i])

conv["VAR\_NAME"] = i

count = count + 1

else:

conv = char\_bin(target, df1[i])

conv["VAR\_NAME"] = i

count = count + 1

if count == 0:

iv\_df = conv

else:

iv\_df = iv\_df.append(conv,ignore\_index=True)

iv = pd.DataFrame({'IV':iv\_df.groupby('VAR\_NAME').IV.max()})

iv = iv.reset\_index()

return(iv\_df,iv)

final\_iv, IV = data\_vars(df[df.columns.difference(['target'])],df.target)

final\_iv.to\_excel(os.path.join(cd, '{0}.xlsx'.format('final\_IV')), index=False)

print (final\_iv)

#selecția variabilelor și pregătirea datelor

IV = IV.rename(columns={'VAR\_NAME':'index'})

IV = IV.sort\_values(['IV'],ascending=0)

IV.to\_excel(os.path.join(cd, '{0}.xlsx'.format('index\_IV')), index=False)

transform\_vars\_list = df.columns.difference(['target'])

transform\_prefix = 'new\_'

print (transform\_vars\_list)

for var in transform\_vars\_list:

small\_df = final\_iv[final\_iv['VAR\_NAME'] == var]

transform\_dict = dict(zip(small\_df.MAX\_VALUE,small\_df.WOE))

replace\_cmd = ''

replace\_cmd1 = ''

for i in sorted(transform\_dict.items()):

replace\_cmd = replace\_cmd + str(i[1]) + str(' if x <= ') + str(i[0]) + ' else '

replace\_cmd1 = replace\_cmd1 + str(i[1]) + str(' if x == "') + str(i[0]) + '" else '

replace\_cmd = replace\_cmd + '0'

replace\_cmd1 = replace\_cmd1 + '0'

if replace\_cmd != '0':

try:

df[transform\_prefix + var] = df[var].apply(lambda x: eval(replace\_cmd))

except:

df[transform\_prefix + var] = df[var].apply(lambda x: eval(replace\_cmd1))

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier()

clf.fit(features,labels)

preds = clf.predict(features)

print (preds)

from sklearn.metrics import accuracy\_score

accuracy = accuracy\_score(preds,labels)

print(accuracy)

from pandas import DataFrame

VI = DataFrame(clf.feature\_importances\_, columns = ["RF"], index=features.columns)

VI = VI.reset\_index()

VI = VI.sort\_values(['RF'],ascending=0)

VI.to\_excel(os.path.join(cd, '{0}.xlsx'.format('index\_RF')), index=False)

from sklearn.feature\_selection import RFE

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression()

rfe = RFE(model, 20)

fit = rfe.fit(features, labels)

from pandas import DataFrame

Selected = DataFrame(rfe.support\_, columns = ["RFE"], index=features.columns)

Selected = Selected.reset\_index()

Selected[Selected['RFE'] == True]

from sklearn.ensemble import ExtraTreesClassifier

model = ExtraTreesClassifier()

model.fit(features, labels)

print(model.feature\_importances\_)

from pandas import DataFrame

FI = DataFrame(model.feature\_importances\_, columns = ["Extratrees"], index=features.columns)

FI = FI.reset\_index()

FI = FI.sort\_values(['Extratrees'],ascending=0)

FI.to\_excel(os.path.join(cd, '{0}.xlsx'.format('index\_Extratress')), index=False)

from sklearn.feature\_selection import SelectKBest

from sklearn.feature\_selection import chi2

model = SelectKBest(score\_func=chi2, k=5)

fit = model.fit(features.abs(), labels)

from pandas import DataFrame

pd.options.display.float\_format = '{:.2f}'.format

chi\_sq = DataFrame(fit.scores\_, columns = ["Chi\_Square"], index=features.columns)

chi\_sq = chi\_sq.reset\_index()

chi\_sq = chi\_sq.sort\_values('Chi\_Square',ascending=0)

chi\_sq.to\_excel(os.path.join(cd, '{0}.xlsx'.format('index\_Chi\_Square')), index=False)

from sklearn.svm import LinearSVC

from sklearn.feature\_selection import SelectFromModel

lsvc = LinearSVC(C=0.01, penalty="l1", dual=False).fit(features, labels)

model = SelectFromModel(lsvc,prefit=True)

from pandas import DataFrame

l1 = DataFrame(model.get\_support(), columns = ["L1"], index=features.columns)

l1 = l1.reset\_index()

l1[l1['L1'] == True]

from functools import reduce

dfs = [IV, VI, Selected, FI, chi\_sq, l1]

final\_results = reduce(lambda left,right: pd.merge(left,right,on='index'), dfs)

final\_results.to\_excel(os.path.join(cd, '{0}.xlsx'.format('final\_results')), index=False)

#calculul scorului variabilelor

columns = ['IV', 'RF', 'Extratrees', 'Chi\_Square']

score\_table = pd.DataFrame({},[])

score\_table['index'] = final\_results['index']

for i in columns:

score\_table[i] = final\_results['index'].isin(list(final\_results.nlargest(5,i)['index'])).astype(int)

score\_table['RFE'] = final\_results['RFE'].astype(int)

score\_table['L1'] = final\_results['L1'].astype(int)

score\_table['final\_score'] = score\_table.sum(axis=1)

score\_table.sort\_values('final\_score',ascending=0).to\_excel(os.path.join(cd, '{0}.xlsx'.format('final\_score')), index=False)

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

def calculate\_vif(features):

vif = pd.DataFrame()

vif["Features"] = features.columns

vif["VIF"] = [variance\_inflation\_factor(features.values, i) for i in range(features.shape[1])]

return(vif)

features = features[list(score\_table[score\_table['final\_score'] >= 2]['index'])]

vif = calculate\_vif(features)

print(vif)

while vif['VIF'][vif['VIF'] > 10].any():

remove = vif.sort\_values('VIF',ascending=0)['Features'][:1]

features.drop(remove,axis=1,inplace=True)

vif = calculate\_vif(features)

list(vif['Features'])

final\_vars = list(vif['Features']) + ['target']

df1 = df[final\_vars].fillna(0)

df1.describe()

bar\_color = '#f47c13'

num\_color = '#ed8549'

final\_iv,\_ = data\_vars(df1,df1['target'])

final\_iv = final\_iv[(final\_iv.VAR\_NAME != 'target')]

grouped = final\_iv.groupby(['VAR\_NAME'])

for key, group in grouped:

ax = group.plot('MIN\_VALUE','EVENT\_RATE',kind='bar',color=bar\_color,linewidth=1.0,edgecolor=['black'])

ax.set\_title(str(key) + " vs " + str('target'))

ax.set\_xlabel(key)

ax.set\_ylabel(str('target') + " %")

rects = ax.patches

for rect in rects:

height = rect.get\_height()

ax.text(rect.get\_x()+rect.get\_width()/2., 1.01\*height, str(round(height\*100,1)) + '%',

ha='center', va='bottom', color=num\_color, fontweight='bold')

# impartirea datelor in date de invatare si date de test

from sklearn.model\_selection import train\_test\_split

train, test = train\_test\_split(df1, test\_size = 0.4)

train = train.reset\_index(drop=True)

test = test.reset\_index(drop=True)

features\_train = train[list(vif['Features'])]

features\_train.to\_excel(os.path.join(cd, '{0}.xlsx'.format('features\_train')), index=False)

#print(features\_train)

label\_train = train['target']

#print(label\_train)

features\_test = test[list(vif['Features'])]

#print(features\_test)

label\_test = test['target']

#print(label\_test)

#aplicarea algoritmului RandomForests

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier()

clf.fit(features\_train,label\_train)

pred\_train = clf.predict(features\_train)

pred\_test = clf.predict(features\_test)

from sklearn.metrics import accuracy\_score

accuracy\_train = accuracy\_score(pred\_train,label\_train)

accuracy\_test = accuracy\_score(pred\_test,label\_test)

from sklearn import metrics

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_train), clf.predict\_proba(features\_train)[:,1])

auc\_train = metrics.auc(fpr,tpr)

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_test), clf.predict\_proba(features\_test)[:,1])

auc\_test = metrics.auc(fpr,tpr)

print("Precizia pentru datele de învățare: ",accuracy\_train\*100, "%")

print("Precizia pentru datele de test: ",accuracy\_test\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de invatare: ",auc\_train\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de test: ",auc\_test\*100, "%")

pd.crosstab(label\_train,pd.Series(pred\_train),rownames=['ACTUAL'],colnames=['PRED'])

pd.crosstab(label\_test,pd.Series(pred\_test),rownames=['ACTUAL'],colnames=['PRED'])

#aplicarea algoritmului GradientBoosting

from sklearn.ensemble import GradientBoostingClassifier

clf = GradientBoostingClassifier()

clf.fit(features\_train,label\_train)

pred\_train = clf.predict(features\_train)

pred\_test = clf.predict(features\_test)

from sklearn.metrics import accuracy\_score

accuracy\_train = accuracy\_score(pred\_train,label\_train)

accuracy\_test = accuracy\_score(pred\_test,label\_test)

from sklearn import metrics

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_train), clf.predict\_proba(features\_train)[:,1])

auc\_train = metrics.auc(fpr,tpr)

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_test), clf.predict\_proba(features\_test)[:,1])

auc\_test = metrics.auc(fpr,tpr)

print("Precizia pentru datele de învățare: ",accuracy\_train\*100, "%")

print("Precizia pentru datele de test: ",accuracy\_test\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de invatare: ",auc\_train\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de test: ",auc\_test\*100, "%")

pd.crosstab(label\_train,pd.Series(pred\_train),rownames=['ACTUAL'],colnames=['PRED'])

pd.crosstab(label\_test,pd.Series(pred\_test),rownames=['ACTUAL'],colnames=['PRED'])

#aplicarea algoritmului LogisticRegression

from sklearn.linear\_model import LogisticRegression

clf = LogisticRegression()

clf.fit(features\_train,label\_train)

pred\_train = clf.predict(features\_train)

pred\_test = clf.predict(features\_test)

from sklearn.metrics import accuracy\_score

accuracy\_train = accuracy\_score(pred\_train,label\_train)

accuracy\_test = accuracy\_score(pred\_test,label\_test)

from sklearn import metrics

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_train), clf.predict\_proba(features\_train)[:,1])

auc\_train = metrics.auc(fpr,tpr)

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_test), clf.predict\_proba(features\_test)[:,1])

auc\_test = metrics.auc(fpr,tpr)

print("Precizia pentru datele de învățare: ",accuracy\_train\*100, "%")

print("Precizia pentru datele de test: ",accuracy\_test\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de invatare: ",auc\_train\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de test: ",auc\_test\*100, "%")

pd.crosstab(label\_train,pd.Series(pred\_train),rownames=['ACTUAL'],colnames=['PRED'])

pd.crosstab(label\_test,pd.Series(pred\_test),rownames=['ACTUAL'],colnames=['PRED'])

import matplotlib.pyplot as plt

preds = clf.predict\_proba(features\_train)[:,1]

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_train), preds)

auc = metrics.auc(fpr,tpr)

plt.figure()

plt.title("Curba ROC pentru datele de învățare")

plt.plot(fpr,tpr,color='#0077bc',label = 'AUC = '+ str(round(auc,3)))

plt.plot([0, 1], [0, 1], color='navy', linestyle='--')

plt.show()

# Performance Tunning

from sklearn.model\_selection import RandomizedSearchCV

from sklearn.ensemble import RandomForestClassifier

n\_estimators = [int(x) for x in np.linspace(start = 10, stop = 500, num = 10)]

max\_features = ['auto', 'sqrt']

max\_depth = [int(x) for x in np.linspace(3, 10, num = 1)]

max\_depth.append(None)

min\_samples\_split = [2, 5, 10]

min\_samples\_leaf = [1, 2, 4]

bootstrap = [True, False]

random\_grid = {'n\_estimators': n\_estimators,

'max\_features': max\_features,

'max\_depth': max\_depth,

'min\_samples\_split': min\_samples\_split,

'min\_samples\_leaf': min\_samples\_leaf,

'bootstrap': bootstrap}

rf = RandomForestClassifier()

rf\_random = RandomizedSearchCV(estimator = rf, param\_distributions = random\_grid, n\_iter = 10, cv = 2, verbose=2, random\_state=42, n\_jobs = -1)

rf\_random.fit(features\_train, label\_train)

print("Random Forests - Hyper parameter tuning:")

print(rf\_random.best\_params\_)

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(\*\*rf\_random.best\_params\_)

clf.fit(features\_train,label\_train)

pred\_train = clf.predict(features\_train)

pred\_test = clf.predict(features\_test)

from sklearn.metrics import accuracy\_score

accuracy\_train = accuracy\_score(pred\_train,label\_train)

accuracy\_test = accuracy\_score(pred\_test,label\_test)

from sklearn import metrics

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_train), clf.predict\_proba(features\_train)[:,1])

auc\_train = metrics.auc(fpr,tpr)

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_test), clf.predict\_proba(features\_test)[:,1])

auc\_test = metrics.auc(fpr,tpr)

print("Precizia pentru datele de învățare: ",accuracy\_train\*100, "%")

print("Precizia pentru datele de test: ",accuracy\_test\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de invatare: ",auc\_train\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de test: ",auc\_test\*100, "%")

pd.crosstab(label\_train,pd.Series(pred\_train),rownames=['ACTUAL'],colnames=['PRED'])

pd.crosstab(label\_test,pd.Series(pred\_test),rownames=['ACTUAL'],colnames=['PRED'])

import matplotlib.pyplot as plt

preds = clf.predict\_proba(features\_train)[:,1]

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_train), preds)

auc = metrics.auc(fpr,tpr)

plt.figure()

plt.title("Curba ROC pentru datele de învățare")

plt.plot(fpr,tpr,color='#0077bc',label = 'AUC = '+ str(round(auc,3)))

plt.plot([0, 1], [0, 1], color='navy', linestyle='--')

plt.show()

#from sklearn import cross\_validation, metrics

from sklearn import metrics

#from sklearn.grid\_search import GridSearchCV

from sklearn.model\_selection import GridSearchCV

from sklearn.ensemble import GradientBoostingClassifier

n\_estimators = [int(x) for x in np.linspace(start = 10, stop = 500, num = 10)]

max\_features = ['auto', 'sqrt']

max\_depth = [int(x) for x in np.linspace(3, 10, num = 1)]

max\_depth.append(None)

min\_samples\_split = [2, 5, 10]

min\_samples\_leaf = [1, 2, 4]

grid = {'n\_estimators': n\_estimators,

'max\_features': max\_features,

'max\_depth': max\_depth,

'min\_samples\_split': min\_samples\_split,

'min\_samples\_leaf': min\_samples\_leaf}

gb = GradientBoostingClassifier()

gf\_tune = GridSearchCV(estimator = gb, param\_grid = grid, cv = 2, verbose=2, n\_jobs = -1)

gf\_tune.fit(features\_train, label\_train)

print("Gradient Boosting - Hyper parameter tuning:")

print(gf\_tune.best\_params\_)

from sklearn.ensemble import GradientBoostingClassifier

clf = GradientBoostingClassifier(\*\*gf\_tune.best\_params\_)

clf.fit(features\_train,label\_train)

pred\_train = clf.predict(features\_train)

pred\_test = clf.predict(features\_test)

from sklearn.metrics import accuracy\_score

accuracy\_train = accuracy\_score(pred\_train,label\_train)

accuracy\_test = accuracy\_score(pred\_test,label\_test)

from sklearn import metrics

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_train), clf.predict\_proba(features\_train)[:,1])

auc\_train = metrics.auc(fpr,tpr)

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_test), clf.predict\_proba(features\_test)[:,1])

auc\_test = metrics.auc(fpr,tpr)

print("Precizia pentru datele de învățare: ",accuracy\_train\*100, "%")

print("Precizia pentru datele de test: ",accuracy\_test\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de invatare: ",auc\_train\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de test: ",auc\_test\*100, "%")

pd.crosstab(label\_train,pd.Series(pred\_train),rownames=['ACTUAL'],colnames=['PRED'])

pd.crosstab(label\_test,pd.Series(pred\_test),rownames=['ACTUAL'],colnames=['PRED'])

import matplotlib.pyplot as plt

preds = clf.predict\_proba(features\_train)[:,1]

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_train), preds)

auc = metrics.auc(fpr,tpr)

plt.figure()

plt.title("Curba ROC pentru datele de învățare")

plt.plot(fpr,tpr,color='#0077bc',label = 'AUC = '+ str(round(auc,3)))

plt.plot([0, 1], [0, 1], color='navy', linestyle='--')

plt.show()

# Final model and Model Performance

from sklearn.ensemble import GradientBoostingClassifier

clf = GradientBoostingClassifier(\*\*gf\_tune.best\_params\_)

clf.fit(features\_train,label\_train)

pred\_train = clf.predict(features\_train)

pred\_test = clf.predict(features\_test)

from sklearn.metrics import accuracy\_score

accuracy\_train = accuracy\_score(pred\_train,label\_train)

accuracy\_test = accuracy\_score(pred\_test,label\_test)

from sklearn import metrics

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_train), clf.predict\_proba(features\_train)[:,1])

auc\_train = metrics.auc(fpr,tpr)

fpr, tpr, \_ = metrics.roc\_curve(np.array(label\_test), clf.predict\_proba(features\_test)[:,1])

auc\_test = metrics.auc(fpr,tpr)

print("Precizia pentru datele de învățare: ",accuracy\_train\*100, "%")

print("Precizia pentru datele de test: ",accuracy\_test\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de invatare: ",auc\_train\*100, "%")

print("Acoperirea zonei de sub curba pentru datele de test: ",auc\_test\*100, "%")

pd.crosstab(label\_train,pd.Series(pred\_train),rownames=['ACTUAL'],colnames=['PRED'])

pd.crosstab(label\_test,pd.Series(pred\_test),rownames=['ACTUAL'],colnames=['PRED'])

import pandas as pd

def scoring(features,clf,target):

score = pd.DataFrame(clf.predict\_proba(features)[:,1], columns = ['SCORE'])

score['DECILE'] = pd.qcut(score['SCORE'].rank(method = 'first'),10,labels=range(10,0,-1))

score['DECILE'] = score['DECILE'].astype(float)

score['TARGET'] = target

score['NONTARGET'] = 1 - target

return(score)

scores\_train = scoring(features\_train,clf,label\_train)

print(scores\_train)

print(clf.predict\_proba(features\_train))

scores\_test = scoring(features\_test,clf,label\_test)

import pandas as pd

import matplotlib.pyplot as plt

def plots(agg1,target,type):

plt.figure(1,figsize=(20, 12))

plt.subplot(211)

plt.plot(agg1['DECILE'],agg1['ACTUAL'],label='Actual')

plt.plot(agg1['DECILE'],agg1['PRED'],label='Pred')

plt.xticks(range(10,110,10))

plt.legend(fontsize=15)

plt.grid(True)

plt.title('Actual vs Predicție', fontsize=20)

plt.xlabel("Populare %",fontsize=15)

plt.ylabel(str(type) + " " + str(target) + " %",fontsize=15)

plt.subplot(223)

X = agg1['DECILE'].tolist()

X.append(0)

Y = agg1['DIST\_TAR'].tolist()

Y.append(0)

plt.plot(sorted(X),sorted(Y))

plt.plot([0, 100], [0, 100],'r--')

plt.xticks(range(0,110,10))

plt.yticks(range(0,110,10))

plt.grid(True)

plt.title('Câștig', fontsize=20)

plt.xlabel("Populare %",fontsize=15)

plt.ylabel(str(" Distribuție") + str(target) + " %",fontsize=15)

plt.annotate(round(agg1[agg1['DECILE'] == 30].DIST\_TAR.item(),2),xy=[30,30],

xytext=(25, agg1[agg1['DECILE'] == 30].DIST\_TAR.item() + 5),fontsize = 13)

plt.annotate(round(agg1[agg1['DECILE'] == 50].DIST\_TAR.item(),2),xy=[50,50],

xytext=(45, agg1[agg1['DECILE'] == 50].DIST\_TAR.item() + 5),fontsize = 13)

plt.subplot(224)

plt.plot(agg1['DECILE'],agg1['LIFT'])

plt.xticks(range(10,110,10))

plt.grid(True)

plt.title('Eficiență', fontsize=20)

plt.xlabel("Populare %",fontsize=15)

plt.ylabel("Eficiența",fontsize=15)

plt.tight\_layout()

def gains(data,decile\_by,target,score):

inputs = list(decile\_by)

inputs.extend((target,score))

decile = data[inputs]

grouped = decile.groupby(decile\_by)

agg1 = pd.DataFrame({},index=[])

agg1['ACTUAL'] = grouped.mean()[target]\*100

agg1['PRED'] = grouped.mean()[score]\*100

agg1['DIST\_TAR'] = grouped.sum()[target].cumsum()/grouped.sum()[target].sum()\*100

agg1.index.name = 'DECILE'

agg1 = agg1.reset\_index()

agg1['DECILE'] = agg1['DECILE']\*10

agg1['LIFT'] = agg1['DIST\_TAR']/agg1['DECILE']

plots(agg1,target,'Distribuție')

lift\_train = pd.concat([features\_train,scores\_train],axis=1)

lift\_train.to\_excel(os.path.join(cd, '{0}.xlsx'.format('lift\_train')), index=False)

lift\_test = pd.concat([features\_test,scores\_test],axis=1)

gains(lift\_train,['DECILE'],'TARGET','SCORE')

plt.show()

gains(lift\_test,['DECILE'],'TARGET','SCORE')

plt.show()

import pandas

from sklearn.externals import joblib

filename = 'final\_model.model'

i = [d,clf]

joblib.dump(i,filename)

Scriptul Qlik Sense Desktop :

//Voturi

LOAD

"index",

IV

FROM [lib://src/index\_IV.xlsx]

(ooxml, embedded labels, table is Sheet1);

LOAD

"index",

RF

FROM [lib://src/index\_RF.xlsx]

(ooxml, embedded labels, table is Sheet1);

LOAD

"index",

Extratrees

FROM [lib://src/index\_Extratress.xlsx]

(ooxml, embedded labels, table is Sheet1);

LOAD

"index",

Chi\_Square

FROM [lib://src/index\_Chi\_Square.xlsx]

(ooxml, embedded labels, table is Sheet1);

LOAD

"index",

IV as "IV\_Final",

RF as "RF\_Final",

Extratrees as "ExtraTrees\_final",

Chi\_Square as "Chi\_Square\_final",

RFE,

L1,

final\_score

FROM [lib://src/final\_score.xlsx]

(ooxml, embedded labels, table is Sheet1);

load \* inline [

Voting\_Algorithms,

"Information Value",

"Random Forests",

"Extra Trees",

"Chi Square"

];

//Predicție

LOAD

eNBcellConfig0dlBandwidth,

total\_bytes\_sdus\_ul,

total\_bytes\_sdus\_dl,

eNBcellConfig0siConfigsfn,

eNBcellConfig0ulPuschPower,

LClcUeConfig0rnti,

eNBcellConfig0dlFreq,

UEueConfig0rnti,

SCORE,

DECILE,

"DECILE\*10",

TARGET,

NONTARGET,

"TOTAL",

ACTUAL,

PRED,

DIST\_TAR,

LIFT

FROM [lib://src/final-qlik.xlsx]

(ooxml, embedded labels, table is Sheet1);