

Universitatea “Politehnica” din Bucureşti
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

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

Proiect de diplomă

prezentat ca cerință parțială pentru obținerea titlului de
Inginer în domeniul *Electronică și Telecomunicații*
programul de studii de licență *Tehnologii și Sisteme de Telecomunicații*

Conducător(i) științific(i)

Conf. Dr. Ing. Alexandru VULPE
Ing. Mihai IDU

Absolvent

Denisa GHEORGHE

2019

Universitatea "Politehnica" din Bucureşti
Facultatea de Electronică, Telecomunicații și Tehnologia Informației
Departamentul Tc

Anexa 1

TEMA PROIECTULUI DE DIPLOMĂ
a studentului **GHEORGHE Gh. Denisa , 444C**

1. Titlul temei: Prelucrarea și analiza datelor de pe interfețele de acces ale unei rețele de comunicații mobile

2. Descrierea contribuției originale a studentului (în afara părții de documentare) și specificații de proiectare:

Scopul lucrării este detectarea degradării rețelei într-o manieră cognitivă (necesită cunoașterea comportamentului anterior pentru detectarea intreruperilor) și determinarea unei serii de tehnici de optimizare (în mod dinamic) a parametrilor în funcție de încărcarea per celulă.

Se va descrie un cadru de analiză a comportamentului unei rețele de telecomunicații bazat pe informațiile oferite de 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.

Parametrii din rețea (3GPP-LTE) descriu atât interfața radio, cât și interfața S1.

Se va construi un set de algoritmi predictivi (dezvoltăți în Python) ce vor primi ca date de intrare informații din interogările SQL, divizate în date de antrenare și date de test. Rezultatele generate de acești algoritmi vor fi vizualizate utilizând utilitarul Qlik(View/Sense) și se vor face predicții referitoare la evoluția comportamentului rețelei. În același timp, luând în calcul aceste date, se va căuta o soluție pentru îmbunătățirea parametrilor din rețea astfel încât traficul suportat de aceasta să fie unul optim.

3. Resurse folosite la dezvoltarea proiectului:

Qlik Sense, Qlik View, nPerf, USRP, terminale mobile, Linux, MySQL, Python

4. Proiectul se bazează pe cunoștințe dobândite în principal la următoarele 3-4 discipline:
Rețele de comunicații, Baze de date, Comunicații analogice și digitale, Programare obiect-orientată

5. Proprietatea intelectuală asupra proiectului aparține: U.P.B.

6. Data înregistrării temei: 2018-12-13 11:41:23

Conducător(i) lucrare,
Ş. L. Dr. Ing. Alexandru VULPE

semnătura:


Student,

semnătura:


Ing. IDU Mihai, Orange Romania

semnătura:


Director departament,
Conf. dr. ing. Eduard POPOVICI

semnătura:


Decan,
Prof. dr. ing. Cristian NEGRESCU

semnătura:


Cod Validare: **ff559ac521**

Declarație de onestitate academică

Prin prezenta declar că lucrarea cu titlul “ Prelucrarea și analiza datelor de pe interfețele de acces ale unei rețele de comunicații mobile”, prezentată în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației a Universității “Politehnica” din București ca cerință parțială pentru obținerea titlului de *Inginer* în domeniul *Electronică și Telecomunicații*, programul de studii *Tehnologii și Sisteme de Telecomunicații* este scrisă de mine și nu a mai fost prezentată niciodată la o facultate sau instituție de învățămînt superior din țară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referințe bibliografice. Fragmentele de text din alte surse, reproduce exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și fac referință la sursă. Reformularea în cuvinte proprii a textelor scrise de către alți autori face referință la sursă. Înțeleg că plagiatul constituie infracțiune și se sanctionează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor și măsurătorilor pe care le prezint ca fiind făcute de mine, precum și metodele prin care au fost obținute, sunt reale și provin din respectivele simulări, experimente și măsurători. Înțeleg că falsificarea datelor și rezultatelor constituie fraudă și se sanctionează conform regulamentelor în vigoare.

București, 28.06.2019

Absolvent *Denisa GHEORGHE*



(semnătura în original)

Cuprins

Capitolul 1 Descrierea sistemului LTE	3
1.1 Rețeaua de comunicații	3
1.2 Evoluția comunicațiilor mobile	4
1.3 Protocole utilizate	6
1.3.1 Code Division Multiple Access (CDMA)	6
1.3.2 Frequency Division Multiple Access (FDMA)	6
1.3.3 Time Division Multiple Access (TDMA)	7
1.4 Sistemul LTE	8
1.4.1 Arhitectura rețelei LTE	8
1.4.1.1 Interfața S1-MME	10
1.4.1.2 Interfața S1-U	11
Capitolul 2 Date și metode de analiză	13
2.1 Date	13
2.1.1 Noțiuni introductive	13
2.1.2 Parametri de performanță din arhitectura LTE	14
2.1.2.1 Parametri de performanță pentru rețea	14
2.1.2.2 Parametri de performanță pentru utilizator	16
2.2 Metode de analiză	18
2.2.1 Metoda pădurilor aleatoare (Random Forests)	18
2.2.1.1 Cum funcționează algoritmul ?	19
2.2.1.2 Pași logici de parcursgere	19
2.2.2 Modelele Gaussian Mixture	20
2.2.2.1 Pasul E	21
2.2.2.2 Pasul M	21
2.2.3 Algoritmul regresiei logistice (Logistic Regression)	22
2.3 Limbajele de programare	24
2.3.1 SQL	24
2.3.1.1 Limbajul de descriere a datelor – DDL	25
2.3.2 Limbaje de programare utilizate în știința datelor	26
Capitolul 3 Implementare Software	29
3.1 Resurse utilizate	29
3.1.1 MySQL Workbench	29
3.1.2 Spyder (Anaconda)	32
3.1.3 Qlik Sense (Desktop)	34
3.2 Aspecte practice	36
3.2.1 Extragerea datelor	36
3.2.1.1 Crearea bazei de date	36
3.2.1.2 Extragerea datelor în fișier Excel	39
3.2.2 Algoritmii de predicție	40
3.2.3 Vizualizarea rezultatelor	48
Capitolul 4 Soluții de îmbunătățire a performanței	53
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 protocolelor S1-MME	11
Figura 1.9 Stiva protocolelor S1-U	11
Figura 2.1 Alocarea puterii pe calea descendenta în LTE.....	14
Figura 2.2 Parametri de control ai PUSCH difuzati de eNodeB către UE.....	15
Figura 2.3 Structura IMSI	17
Figura 2.4 Schema logică de parcursare a algoritmului Random Forests	20
Figura 2.5 Funcția sigmoid.....	22
Figura 2.6 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 tipxlsx	40
Figura 3.14 Diagrama software	40
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 - Regresia logistică	48
Figura 3.19 Curba ROC - Variația creșterii	48
Figura 3.20 Rezultatele votului afișate în Qlik Sense	49
Figura 3.21 Rezultatele filtrate per index	49
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 Partnership Project	
AI	Artificial Intelligence	Inteligenta artificială
BTS	Base Transceiver 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 descendentală
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 Connectivity	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 Connectivity	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 descendentală
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 practică 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.

Capitolul 1 Descrierea sistemului LTE

1.1 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.

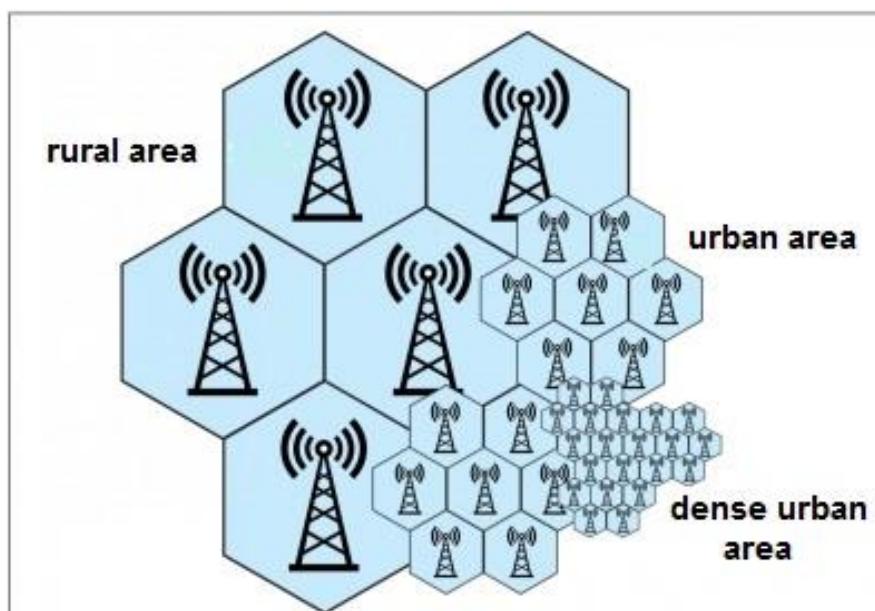


Figura 1.1 Împărțirea zonei geografice în celule [2]

Puterea limitată a stației de bază face posibilă reutilizarea aceleiași frecvențe la câteva celule distanță de stația de bază fără a provoca interferențe, aşa 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.

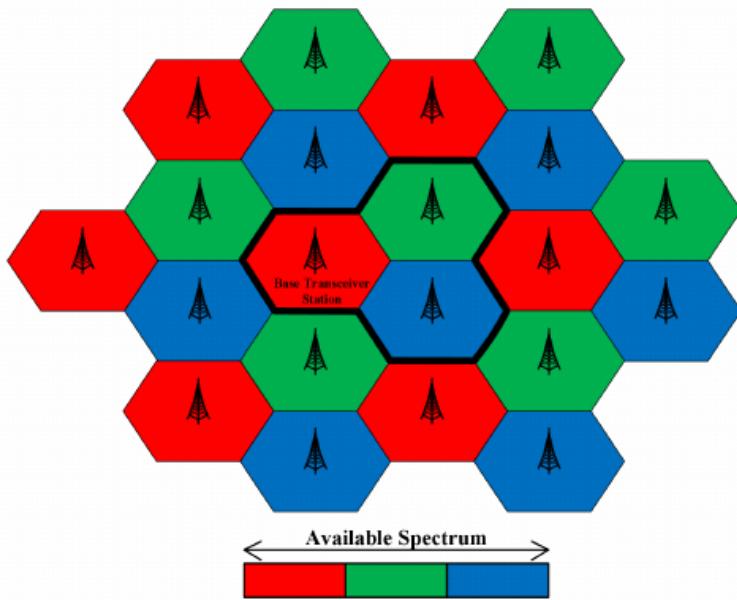


Figura 1.2 Reutilizarea frecvențelor [3]

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

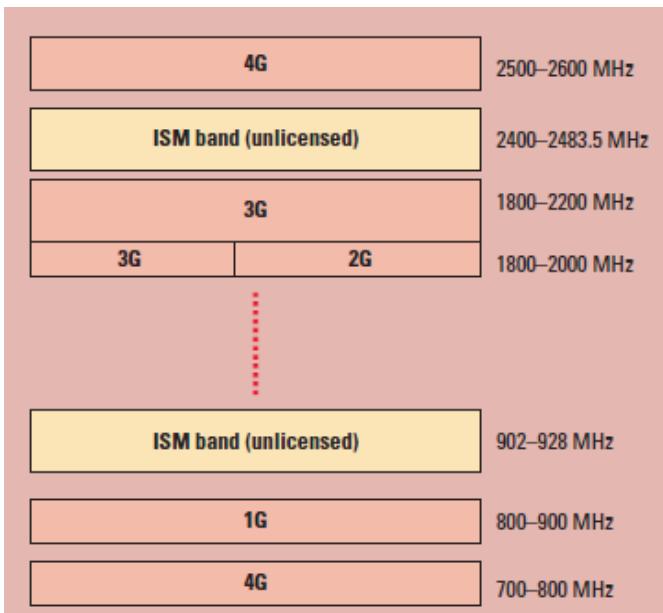
1.2 Evoluția comunicățiilor mobile

Tabelul 1 arată cum au evoluat rețelele celulare de la 1G la 4G, evidențiind caracteristicile cheie pentru fiecare generație, calitatea serviciilor (QoS) și protocolele 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	Protocole 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	<ul style="list-style-type: none"> - utilizează tehnologia digitală pentru securitatea vocii și a datelor - suportă roaming - oferă acces la rețelele 1G 	<ul style="list-style-type: none"> - fiabilitate îmbunătățită prin back-up la rețeaua 1G - QoS mai bun - performanța nu a atins așteptările 	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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 	<ul style="list-style-type: none"> - fiabilitate îmbunătățită prin back-up la rețelele 2G și uneori 1G - QoS îmbunătățit - performanța atinge așteptările 	TDMA și multiple variații ale CDMA
4G/2011	<ul style="list-style-type: none"> - 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 	<ul style="list-style-type: none"> - 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 700, 800 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]

1.3 Protocole 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).

1.3.1 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, permitând mai mulți abonați să se conecteze la orice moment de timp
- este platforma pe care s-au construit tehnologiile 3G

1.3.2 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). **Eroare! Fără sursă de referință.**

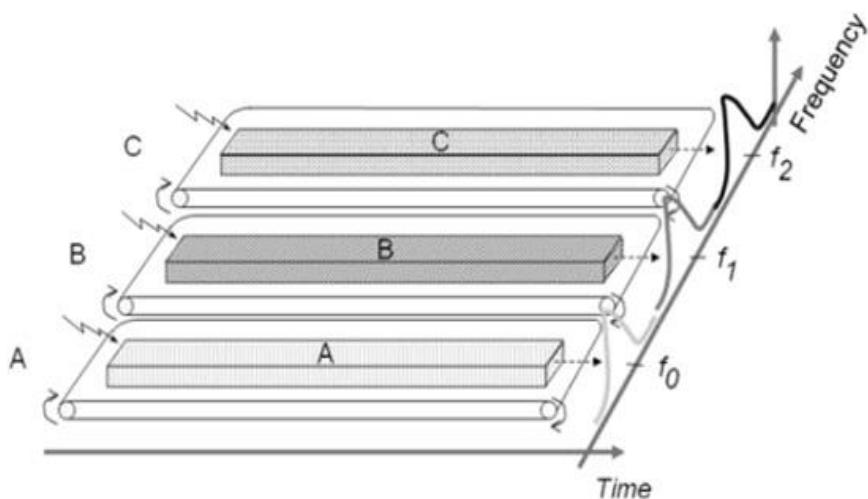


Figura 1.4 Împărțirea benzii de frecvență [5]

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

1.3.3 Time Division Multiple Access (TDMA)

TDMA este o tehnologie digitală complexă a comunicațiilor mobile celulare care permite mai mulți 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. Fiecare utilizator îi este atribuit un interval de timp (timeslot) pentru a trimite și receptiona informații. Utilizatorul B transmite după A, iar C transmite după B.

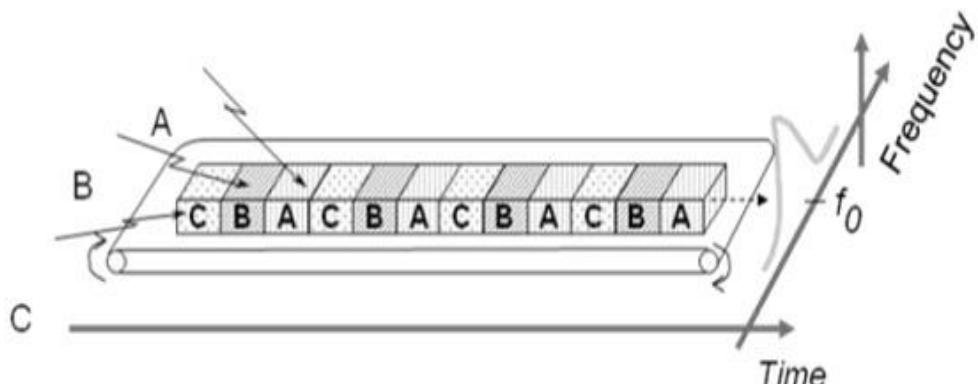


Figura 1.5 Alocarea timeslot-urilor [5]

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 față de UL.

Pentru DL este folosită multiplexarea OFDMA (Orthogonal Frequency Division Multiple Access), prin care se alocă mai mulți utilizatori atât în domeniul timp, cât și în domeniul 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 în domeniul timp, conform aceleiași figuri.

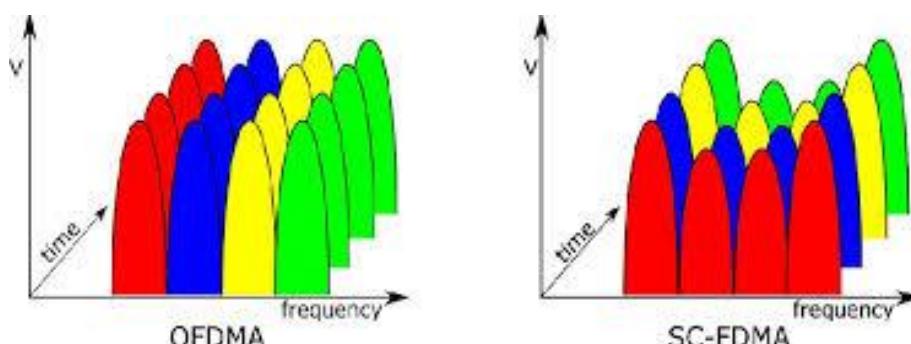


Figura 1.6 Multiplexarea în LTE [6]

1.4 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 descendenta (DL) și 50 Mb/s pe calea ascendentă (UL). Conform Figura 1.7 Figura 1.7 Arhitectura rețelei LTE [7] înțelegem prin cale ascendentă calea de la utilizator la stația de bază, iar calea descendenta 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.

1.4.1 Arhitectura rețelei LTE

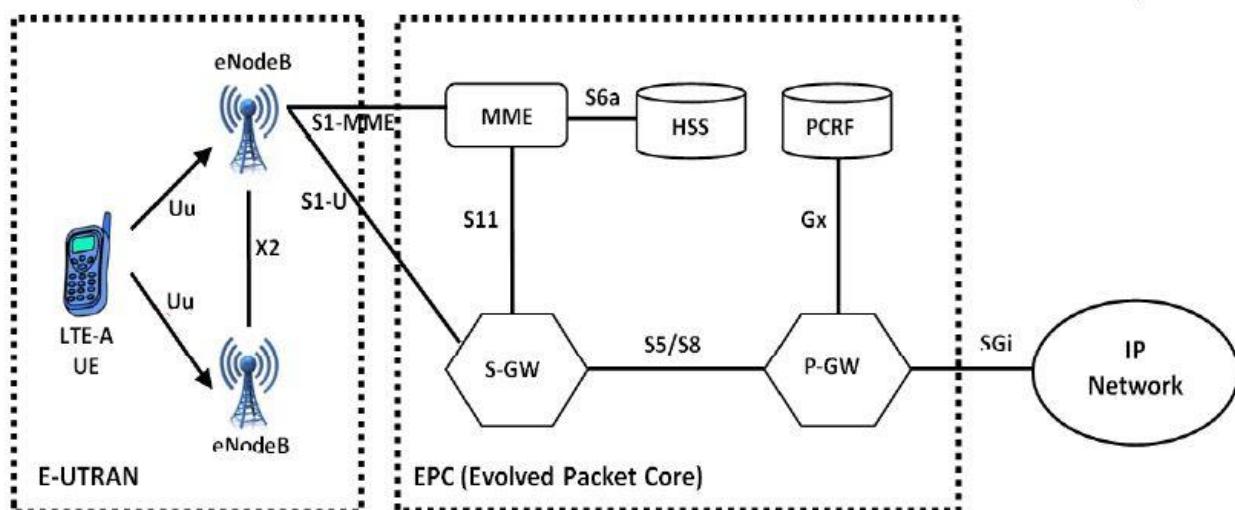


Figura 1.7 Arhitectura rețelei LTE [7]

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 și 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 autentificării 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, permitând ca multe apeluri să fie rulate 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 (Evolved 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 diferențele 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 mobilitatea 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 stăriile mobile în stare de aşteptare (actualizarea locației și pagingul), funcții îndeplinite de VLR (Visitor Location Register), respectiv 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 securitatea între rețea și UE.
- HSS – Home Subscriber Server conține datele utilizatorilor, printre 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 prezent atașat sau înregistrat. HSS poate să integreze, de asemenea, centrul de autentificare (AUC), care generează vectorii de autentificare și chei de securitate.
- Interfețele S1 - Interfața dintre stația de bază și nucleul rețelei se numește interfața S1. De obicei, aceasta presupune o legătură de fibră optică sau cablu de cupru de mare viteză. Această interfață este împărțită în două părți logice, care transportă informația prin același canal fizic. Acestea sunt S1 User Plane (S1-U), pentru datele utilizatorilor și S1 Control Plane (S1-MME), pentru datele de control ale rețelei LTE.

1.4.1.1 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:

- interacțiunea eNodeB-ului cu nucleul rețelei pentru comunicări specifice
- transferul mesajelor de semnalizare care țin 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 protocolelor 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).

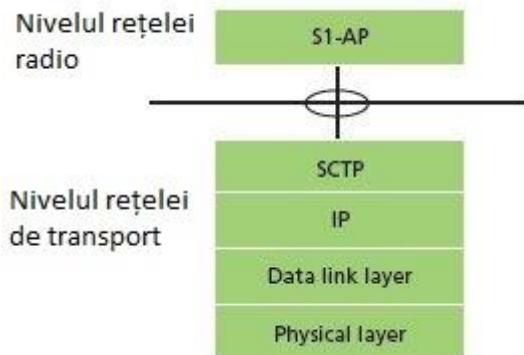


Figura 1.8 Stiva protocalelor S1-MME [9]

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 protocole simplificată, fără un protocol intermediu de gestionare a conexiunilor,

deoarece conexiunile individuale sunt tratate direct la nivelul aplicației.

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

LTE a construit, de asemenea, flexibilitate în protocalele 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.

1.4.1.2 Interfața S1-U

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

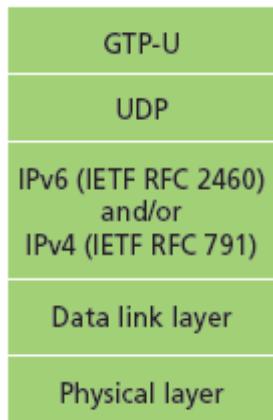


Figura 1.9 prezintă structura protocalelor intefetei S1 User Plane, bazată pe GTP/UDP5/stiva IP, cunoscută de la rețelele 3G.

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 optionale, ca și în 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 protocalelor S1-U [9]

Capitolul 2 Date și metode de analiză

2.1 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.

2.1.1 Noțiuni introductive

Bazele de date reprezintă nucleul sistemelor informatici din orice companie sau instituție, având un impact major asupra modului de funcționare și organizare al acestora. Totodată oferă o deschidere majoră asupra pieții pe care o vizează, oferind posibilitatea clienților de a avea acces în mod facil la datele esențiale de care aceștia 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 Gestire 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 și modificabil de un grup de utilizatori care lucrează în paralel, prin intermediul uneia sau a mai multor aplicații. Pe de altă parte, un Sistem de Gestire a Bazelor de Date (SGBD) poate fi văzut generic ca un sistem care se ocupă de structurarea, stocarea, actualizarea și menținerea datelor, reprezentând de fapt interfața 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 relaționale au la bază modelul relațional care lucrează cu două concepte importante: relație și tabel. Cele două diferă prin natura lor, dar sunt foarte corelate. Noțiunea de relație este formală, deoarece conceptul provine din matematică, în particular din teoria mulțimilor, în timp ce noțiunea de table este simplă și intuitivă. Pe de o parte, tabelele oferă o înțelegere naturală a structurii bazei de date chiar și pentru utilizatorii ce nu sunt specializați în domeniul. Pe de altă parte, existența unei formalizări matematice clare și simple a permis dezvoltarea unei teorii care să sprijine modelul, cu rezultate foarte interesante în industrie [10].

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

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

2.1.2 Parametri de performanță din arhitectura LTE

Un parametru de performanță este o valoare măsurabilă care 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 configurației echipamentelor de rețea. De aceea, în continuare voi oferi câteva exemple de indicatori, grupați după zona pe care o evaluatează (utilizator/rețea).

2.1.2.1 Parametri de performanță pentru rețea

- Puterea de transmisie a canalului fizic PDSCH pe traiectul descendente - Physical Downlink Shared Channel(PDSCH) Power:
 - În LTE, PDSCH este singurul canal de transport pe calea descendente, 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.

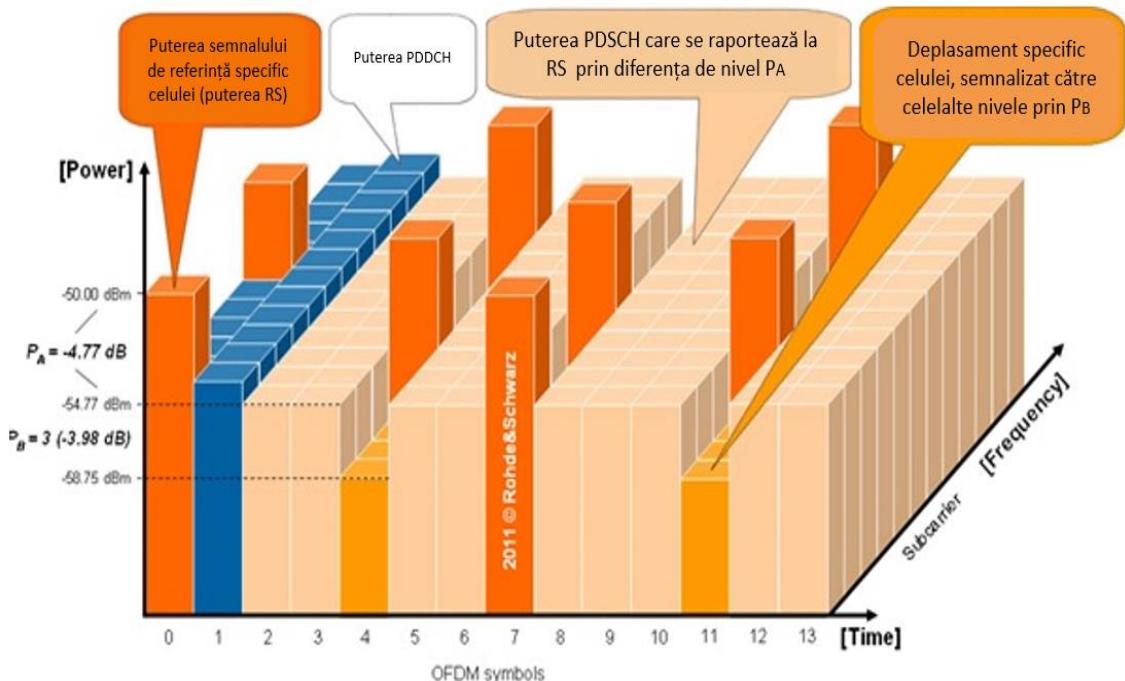


Figura 2.1 Alocarea puterii pe calea descendente în LTE [11]

- 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 P_A poate de asemenea să se schimbe pe o bază de 1 milisecundă. În timp ce încorporează P_A și P_B , 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:
 - Își acordă 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 :

$$P_{\text{PUSCH}} = \min\{ P_{\max}, 10 \cdot \log_{10} M + P_0 + \alpha \times PL + \delta_{\text{mcs}} + f(\Delta_i) \} [\text{dBm}], \text{ unde}$$

P_{\max} este puterea maximă transmisă care depinde de clasa de putere a UE;

M este numarul de blocuri de resurse

P_0 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 descendentală și sunt calculate în UE

δ_{mcs} este un parametru al schemei de modulație și codare, specific celulei

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

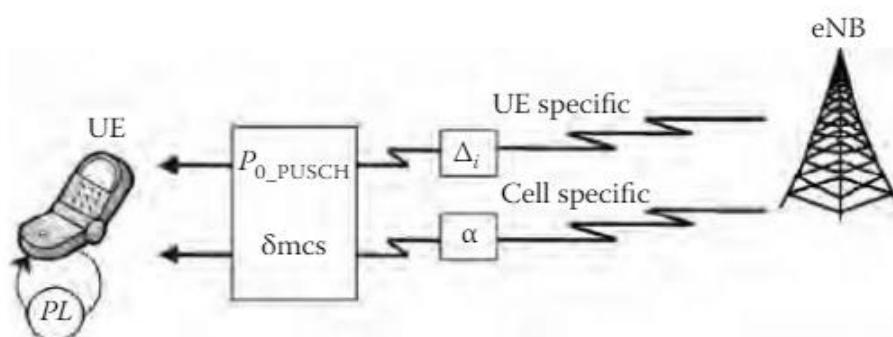


Figura 2.2 Parametri de control ai PUSCH difuzati de eNodeB către UE [12]

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 descendantă 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 descendantă.

Tabelul 2 Măsurarea frecvențelor

Lărgimea de bandă	Blocuri de resurse	Subpurtătoare pentru calea ascendentă	Subpurtătoare pentru calea descendantă
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 aşa 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 sincronizarea cu forma de undă.
 - SFN ajută la sincronizarea nivelului fizic dintre canalele fizice ale căii ascendente și cele ale căii descendente. [13]

2.1.2.2 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 aparete mobile.

- În linii mari, el acționează ca o falsă celulă de telefonie mobilă, care se interpune între telefonul-țintă și adevăratul turn-antena al companiei de telefonie.
- Este de obicei un număr de 15 digits, 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)

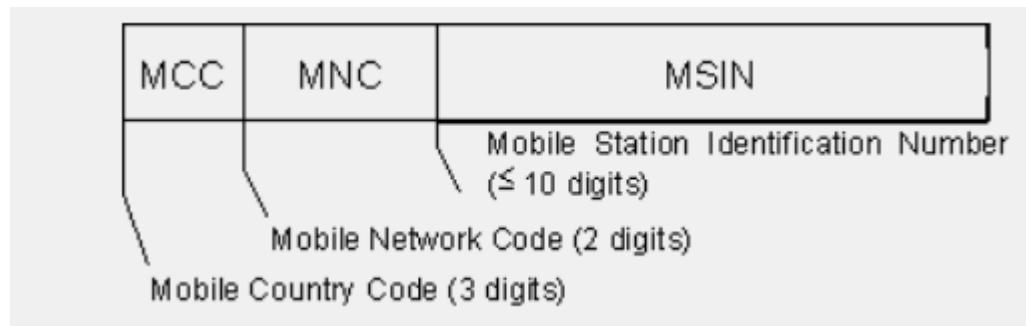


Figura 2.3 Structura IMSI [14]

- 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 receptiona 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.

- Identifierul 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 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 (0xFFE).
 - 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 0xFFFF3). [15]

2.2 Metode de analiză

Conceptul de învățare mecanică (*machine learning*) se referă la studiul științific al algoritmilor și modelelor statistice pe care sistemele informatiche 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 utilizati 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 caracteristici esențiale ale unor modele de analiză foarte bine cunoscute în domeniul învățării mecanice.

2.2.1 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.

2.2.1.1 Cum funcționează algoritmul ?

[17] Să presupunem că vreți să mergeți într-o excursie și vreți să călătoriți într-un loc unde vă veți bucura.

Deci, ce faceți pentru a găsi un loc care vă va plăcea? Puteți căuta online, citi recenzii despre blogurile și portalurile de călătorie sau vă puteți întreba și prietenii.

Să presupunem că ați decis să întrebați prietenii dumneavoastră și ați discutat cu ei despre experiența lor de călătorie din trecut în diverse locuri. Veți primi câteva recomandări de la fiecare prieten. Acum trebuie să faceți o listă a locurilor recomandate. Apoi, le cereți să voteze (sau să aleagă un loc potrivit pentru călătorie) din lista locurilor recomandate pe care ați făcut-o. Locul cu cel mai mare număr de voturi va fi alegerea dvs. finală pentru călătorie.

În procesul de decizie de mai sus, există două părți. În primul rând, întrebați-vă prietenii despre experiența lor individuală de călătorie și obținerea unei recomandări din mai multe locuri pe care le-au vizitat. Această parte este ca și cum ați 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 prietenii 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 (bazată pe abordarea divizată și cucerire) 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.

2.2.1.2 Pași logici de parcurgere

Algoritmul 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.

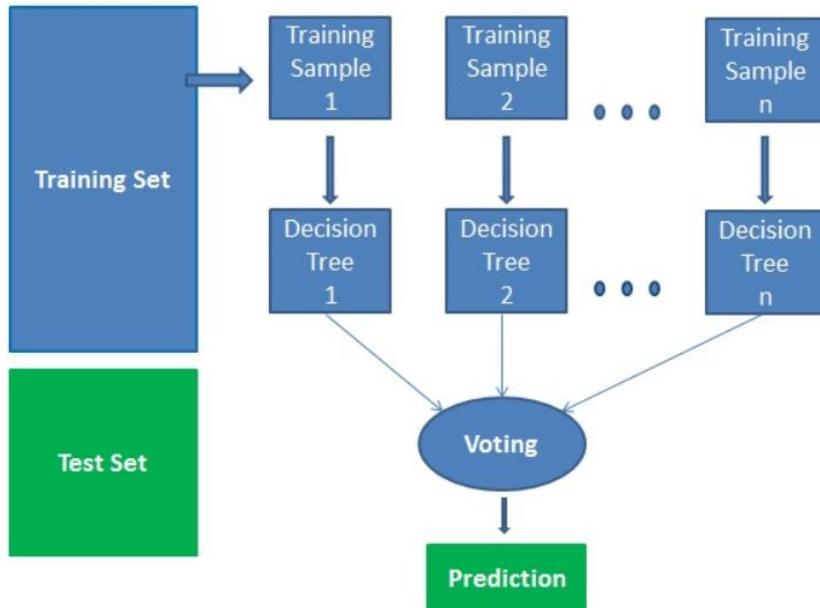


Figura 2.4 Schema logică de parcursare a algoritmului Random Forests

2.2.2 Modelele Gaussian Mixture

Distribuțiile amestecurilor (mixture distributions) sunt utile în modelarea heterogenității într-o analiză cluster. A fost demonstrat că, cu $n = 10\,000$ de observații, un amestec (mix) de aproximativ 30 de normale este suficient pentru a produce o aproximare bună a densității normale a logaritmului, în timp ce pentru un amestec de 10 000 de normale a fost necesar un estimator de densitate a kernelului. Acest lucru se datorează capacitatea modelului de amestec de a modela distribuții destul de complexe atunci când se alege un număr convenabil de componente pentru a obține reprezentări exakte ale zonelor locale care suportă distribuția adevărată. Cu această caracteristică, sunt tratate variațiile locale ale datelor observate sunt tratate, în timp ce o singură familie parametrică nu ar fi în măsură să facă acest lucru. [18]

Modelul Gaussian Mixture, adesea denumit GMM, se bazează pe o funcție de densitate a probabilității parametrice care este reprezentată ca o sumă ponderată a densităților componentelor gaussiene. Parametrii modelului se estimatează prin utilizarea algoritmului Expectation-Maximisation (EM – așteptare – maximizare) pe datele de învățare.

Pentru un set de date $D = \{x_1, \dots, x_N\}$, unde x_i este un vector d-dimensional de observații, presupunem că punctele sunt IID (independente și identic distribuite) și că densitatea lor de bază $p(x)$ este definită ca un model de amestec finit cu componente K. Funcția parametrică de probabilitate este dată de :

$$p(x|\lambda) = \sum_{k=1}^K \omega_k g_k(x|\mu_k, \Sigma_k),$$

unde $\lambda = \{\omega_k, \mu_k, \Sigma_k\}$, $k = 1, \dots, K$ denotă parametrii GMM, numiți mărimea amestecului ω_k pentru care

$$\sum_{k=1}^K \omega_k = 1$$

mijlocul μ_k și covarianța matricii Σ_k .

ω_k reprezintă probabilitatea ca un x selectat aleatoriu să fie generat de componenta k . Parametrii μ_k și Σ_k descriu densitatea vectorului de date d -dimensional cu valoare continuă a măsurătorilor x , care sunt reprezentate matematic de funcțiile gaussiene $g(x|\mu_k, \Sigma_k)$, $k = 1, \dots, K$ de forma :

$$g_k(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_k)' \Sigma_k^{-1} (x - \mu_k) \right\}.$$

O metodă comună de estimare a parametrilor, dată de o secvență N de vectori de învățare $X = \{x_1, \dots, x_N\}$ și setările de configurare ale GMM, este estimarea maximă a probabilității (maximumlikelihood - ML). Estimarea se realizează prin maximizarea probabilității ca GMM să ia în considerare datele de antrenament în X . Prin urmare, vrem să maximizăm

$$p(X|\lambda) = \prod_{i=1}^N p(x_i|\lambda)$$

cu presupunerea că vectorii sunt independenți datorită necesității de a face problema maleabilă, deși această presupunere este adesea incorectă. Maximizarea directă a lui $p(X|\lambda)$ nu este posibilă, dar prin utilizarea estimării algoritmului de așteptare-maximizare (EM) a parametrilor ML se poate obține iterativ. Acest lucru se realizează prin estimarea unui model îmbunătățit nou, dintr-un model inițial, astfel încât $p(X|\bar{\lambda}) \geq p(X|\lambda)$. Prin iterarea acestui pas până la convergență, adică până schimbarea mijloacelor este mică, se obțin parametrii modelului. Mai jos voi arăta pașii E,M :

1. Decidem în câte surse/grupuri (k) vrem să împărțim datele.
2. Inițializăm parametrii μ_k , Σ_k și ω_k .

2.2.2.1 Pasul E

Calculăm pentru fiecare punct x_i probabilitatea $\gamma_{i,k}$ ca acesta să aparțină grupului k cu :

$$\gamma_{i,k} = \frac{\omega_k g_k(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^K \omega_j g_j(x_i|\mu_j, \Sigma_j)}$$

$\gamma_{i,k}$ ne oferă pentru fiecare punct x_i valoarea raportului : $\frac{\text{probabilitatea ca } x_i \text{ să aparțină grupului } k}{\text{probabilitatea ca } x_i \text{ să aparțină tuturor grupurilor}}$
Astfel, dacă x_i este foarte aproape de un gaussian k , va obține o valoare ridicată pentru acest gaussian și valori relativ scăzute pentru celelalte.

La sfârșitul acestui pas vom obține o matrice de $N \times K$, cu suma fiecărei linii egală cu 1.

2.2.2.2 Pasul M

Aici parametrii pentru fiecare gaussian k sunt actualizați utilizând $\gamma_{i,k}$. Reestimarea componentelor $\bar{\lambda}$ este posibilă utilizând formulele, pentru care $1 \leq k \leq K$:

$$\bar{\omega}_k = \frac{1}{N} \sum_{i=1}^N \gamma_{i,k}$$

$$\bar{\mu}_k = \frac{\sum_{i=1}^N \gamma_{i,k} * x_i}{\sum_{i=1}^N \gamma_{i,k}}$$

$$\Sigma_k = \frac{\sum_{i=1}^N \gamma_{i,k} * (x_i - \bar{\mu}_k) * (x_i - \bar{\mu}_k)'}{\sum_{i=1}^N \gamma_{i,k}}$$

Pentru fiecare implementare a pașilor E și M, este efectuată câte o iterație. Aceasta trebuie repetată până când probabilitatea sau parametrii converg. Astfel, vom logaritma $p(X|\lambda)$ după fiecare iterație, iar atunci când este găsită convergența, oprim iterarea.

$$\log p(X|\lambda) = \sum_{i=1}^N \log p(x_i|\lambda) = \sum_{i=1}^N (\log \sum_{k=1}^K \omega_k g_k(x_i|\mu_k, \Sigma_k))$$

2.2.3 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]

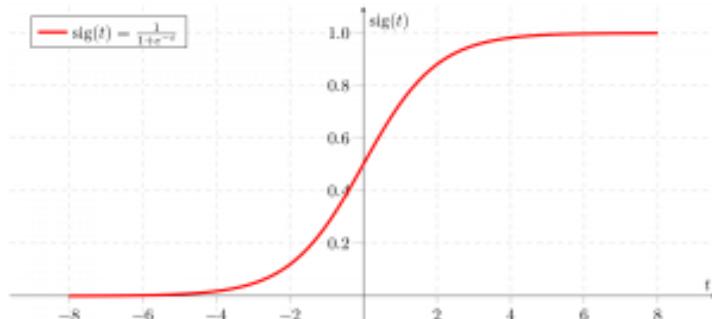


Figura 2.5 Funcția sigmoid [19]

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 nefind 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 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 ţintă poate avea doar 2 tipuri posibile: "0" sau "1" care pot reprezenta "câștiga" vs "pierdere", "trece" vs "eșuează", "mort" vs "viu"
- Multinomial: variabila ţintă 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 ţintă 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 :

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}$$

Unde x_{ij} conține valoarea celei de-a j caracteristici pentru a i observație.

Pentru a i obervație, x_i poate fi reprezentat ca :

$$x_i = \begin{bmatrix} 1 \\ x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{bmatrix}$$

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

$$h(x_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}$$

unde $\beta_0, \beta_1, \dots, \beta_p$ sunt coeficienții regresiei.

Și într-o formulă compactă : $h(x_i) = \beta^T x_i$.

2.3 Limbajele de programare

2.3.1 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 și 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 afișându-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 funcționalități pentru ușurarea modului de interacțiune al utilizatorului cu baza de date pe care o administreză 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 internațional, toate aceste sisteme adoptă o aceeași normă pentru definirea relațiilor și manipularea datelor.

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

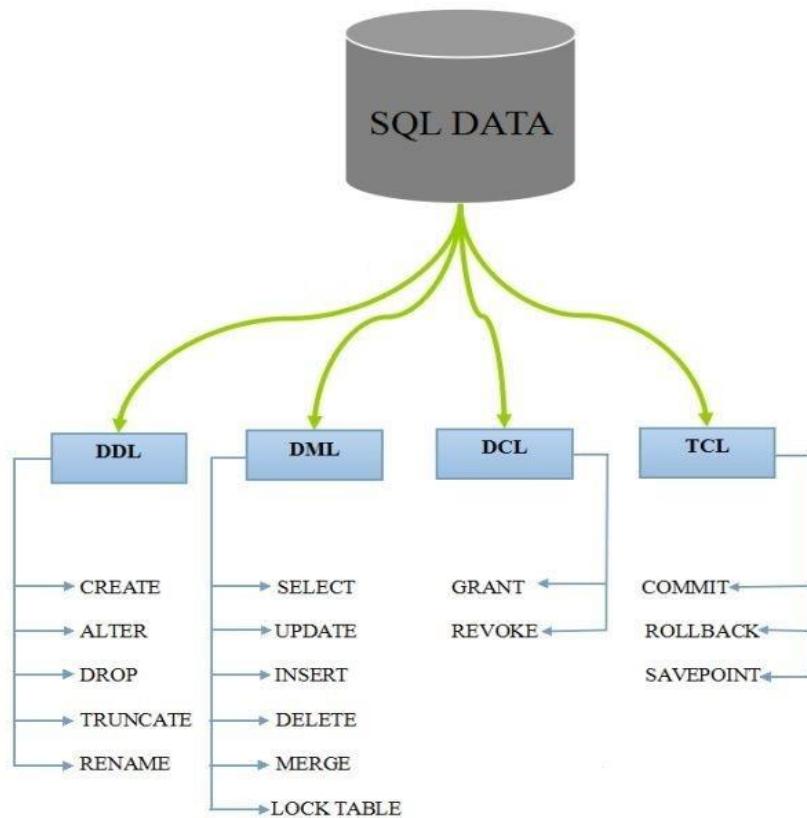


Figura 2.6 Tipuri de comenzi în SQL

În această lucrare voi folosi două dintre cele patru categorii, celelalte 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 și a relațiilor dintre elementele componente, precum și atribuirea drepturilor de acces a utilizatorilor la baze de date;

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

2.3.1.1 Limbajul de descriere a datelor – DDL

Limbajul de descriere a datelor este compus în principal din trei comenzi: CREATE, ALTER și DROP. Instrucțiunea CREATE este folosită pentru crearea structurii (schemei) esențiale 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 comenzi 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 comenzi :

```
DESCRIBE mytable;
```

2.3.2 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.

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

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%	Inteligenta artificială în jocuri: 29% Locomoția roboților: 27% Securitatea rețelelor: 26%	Managementul clienților: 26% Securitatea rețelelor: 23% Detectiona fraudei: 22%	Analiză: 11% Bioinformatică: 11% Detectiona fraudei: 9%	Managementul clienților: 10% Motoare de căutare: 9%
	Prioritizat cel mai puțin	Inteligenta artificială în jocuri: 26% Detectiona fraudei: 26% Securitatea rețelelor: 24%	Detectiona fraudei: 12% Sisteme de recomandare: 12% Analiză: 9%	Analiză: 15% Bioinformatică: 13%	Inteligenta artificială în jocuri: 3% Locomoția roboților: 1%	Mentenanța 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-1 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%	Cresterea șanselor de a avea proiecte profitabile: 8%
	Prioritizat cel mai puțin	Obținerea diplomei de studii: 26%	În construirea aplicațiilor 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%

- 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 dezvantajele 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.

Capitolul 3 Implementare Software

Figura 3.1 este o reprezentare grafică a întregului proces descris în prezenta lucrare. Aşa 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 algoritmul predictiv vor fi prezentate sub formă grafică și vizualizate, aceasta etapă fiind ultima etapă a implementării software a procesului.

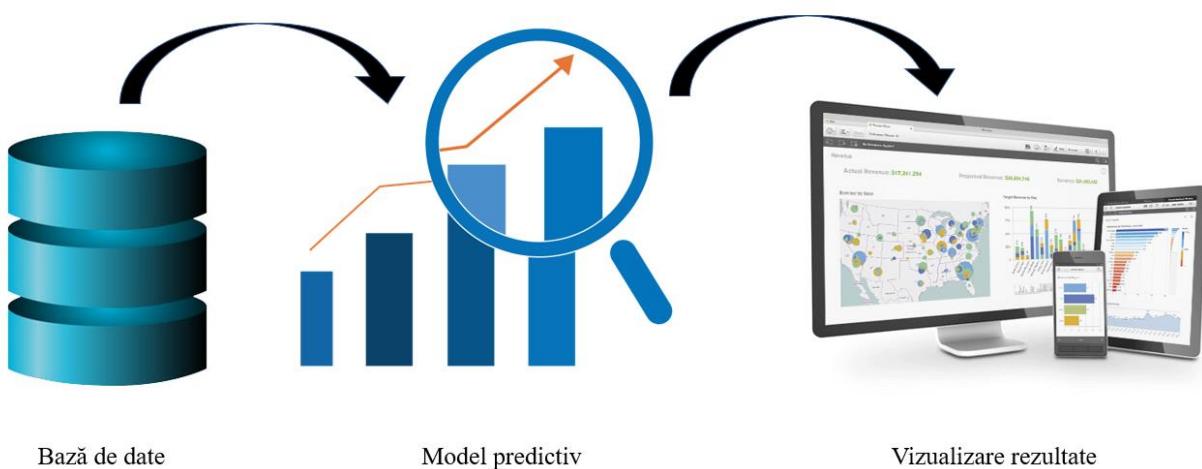


Figura 3.1 Schema generală a implementării

Baza de date este stats_db și este creată 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 rezultatele ale etapelor intermediare ale codului, cât și rezultatele finale pe baza cărora se va emite o serie de concluzii.

3.1 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 intermediul utilizatorului Qlik Sense, utilizat frecvent în analiza datelor.

3.1.1 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 și poate fi modificat în funcție de necesitățile programatorului), foarte popular printre dezvoltatorii de aplicații software și pagini web. Are la

bază limbajul de date SQL, rulează pe orice tip de sistem de operare și poate fi interconectat cu ușurință cu alte aplicații dezvoltate în limbi de programare precum: C++, Java, Python, 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, partaționării, 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, PostgreSQL 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 instanță 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.

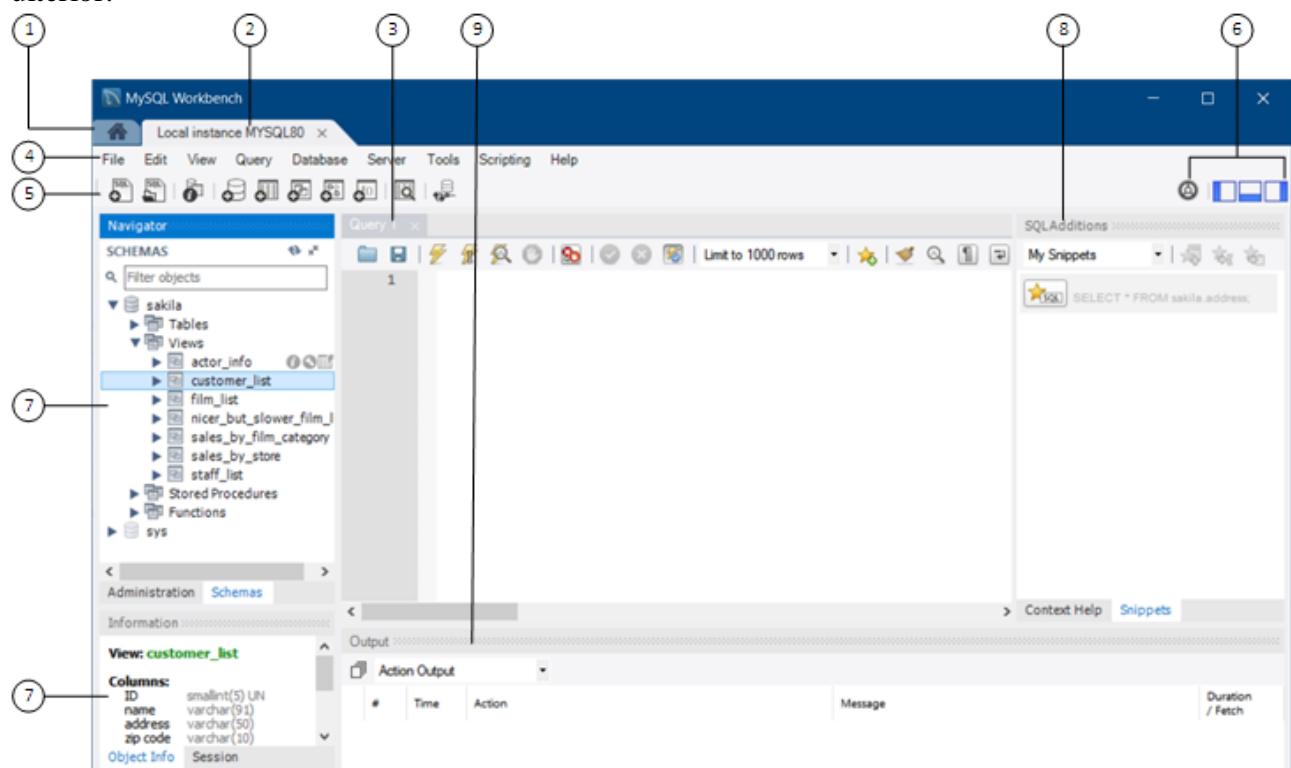


Figura 3.3 Fila de editare a MySQL Workbench

- 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 aşa 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.

3.1.2 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, permitându-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”.

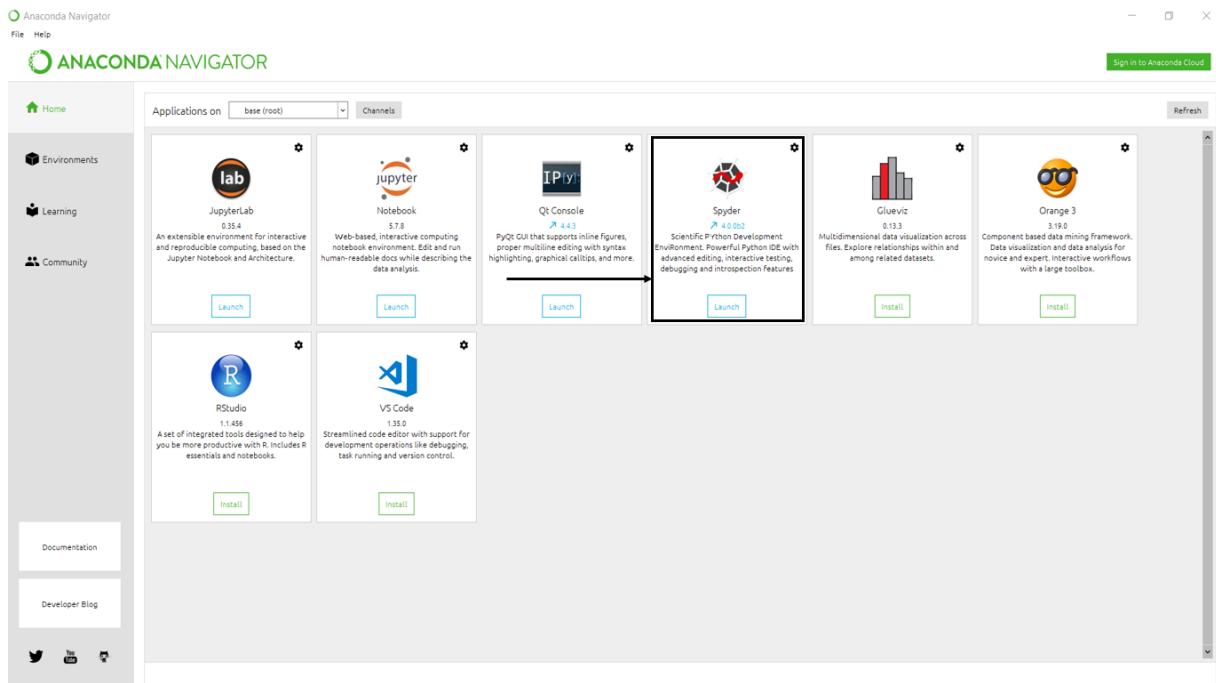


Figura 3.4 Fila de start Anaconda Navigator

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.

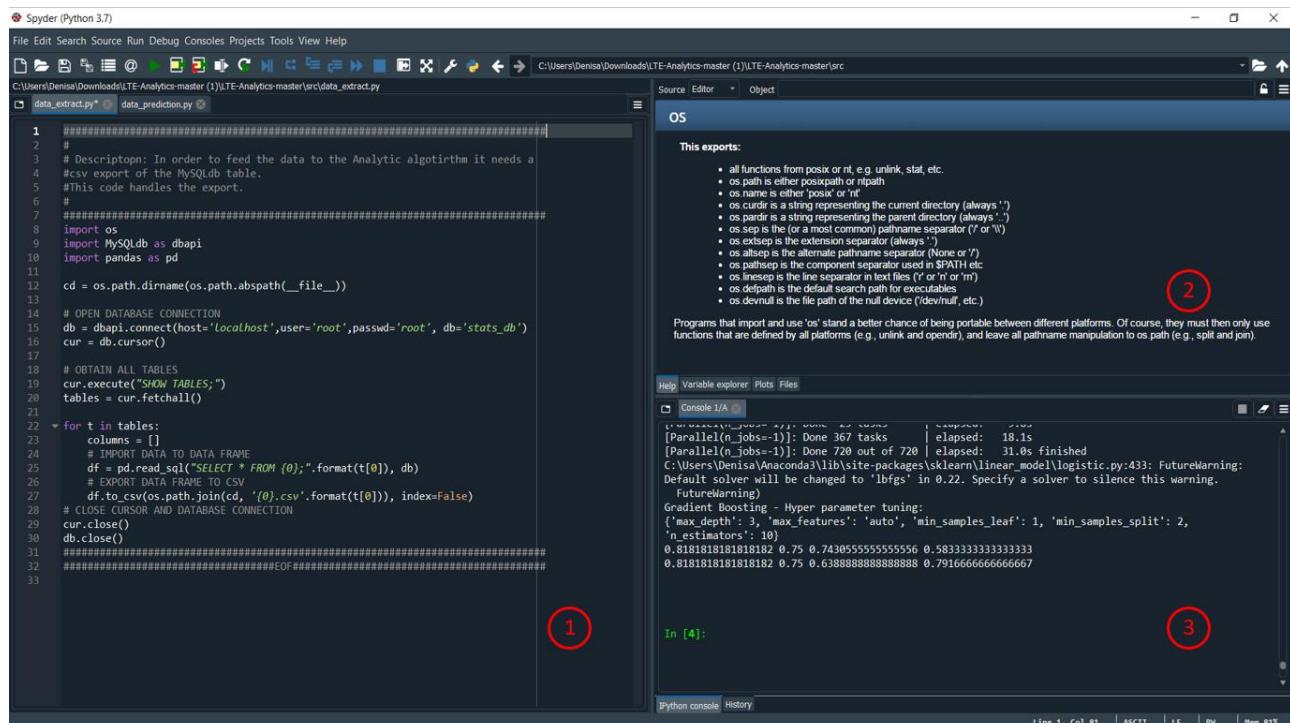


Figura 3.5 Fila de activitate Spyder 4

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

3.1.3 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.

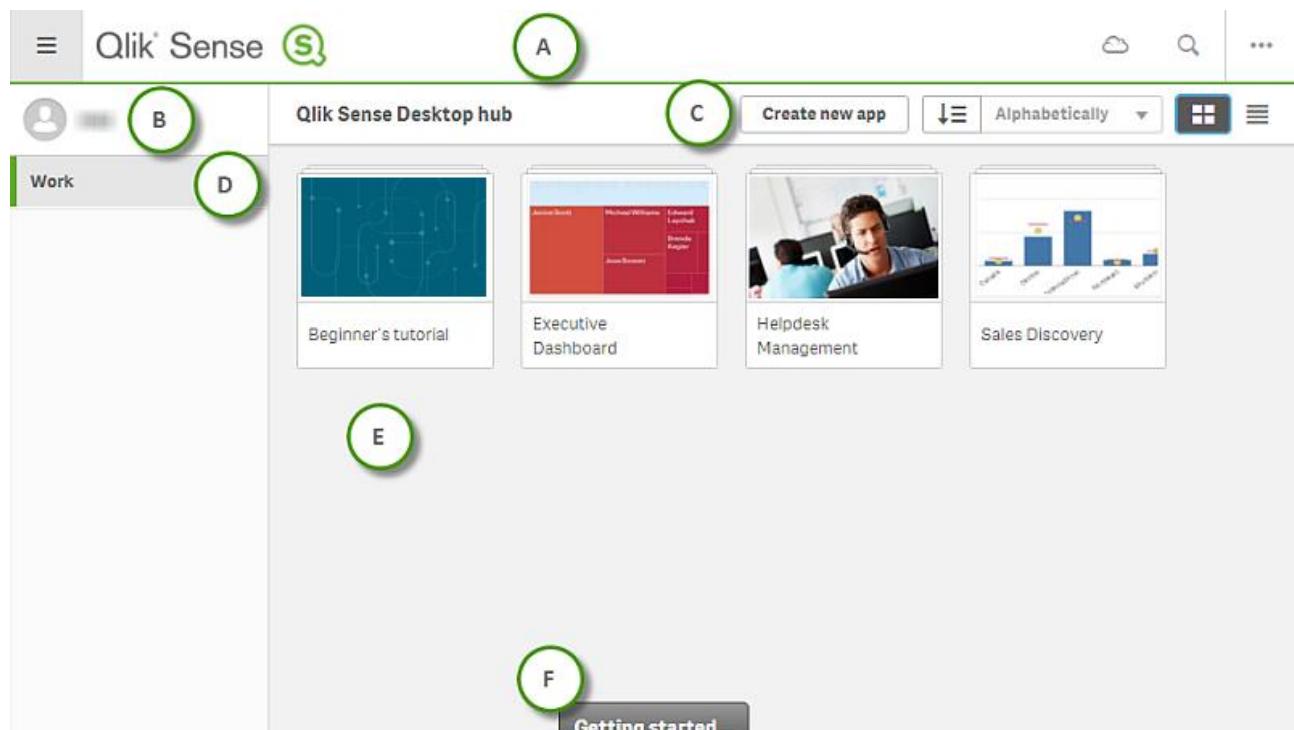


Figura 3.6 Nodul central Qlik Sense

- A) Bara de instrumente de putem deschide sau închide meniul de navigare, scurtătură către locația de stocare a Qlik Sense, locul de căutare a aplicațiilor
- B) Informații de autentificare
- C) Rubrica de creare a unei noi aplicații
- D) Domeniul de găzduire al aplicațiilor
- E) Zona principală care conține toate aplicațiile create
- F) 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.

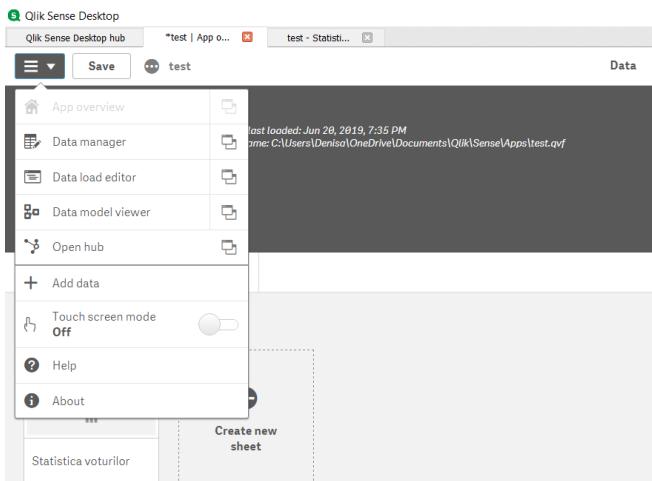


Figura 3.7 Vizualizare din interiorul aplicației

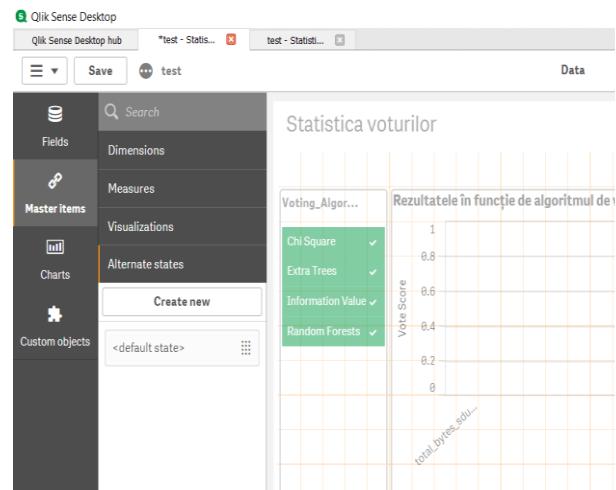


Figura 3.8 Vizualizare din zona de dezvoltare

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 înțeles, să fie alese corespunzător tipurile de vizualizări care se potrivesc scopului și ca acestea să fie cât mai inteligibile utilizatorului.

3.2 Aspecte practice

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

3.2.1 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.

3.2.1.1 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', aşa cum este evidențiat în Figura 3.9 în rubrica 'Schemas-Session'.

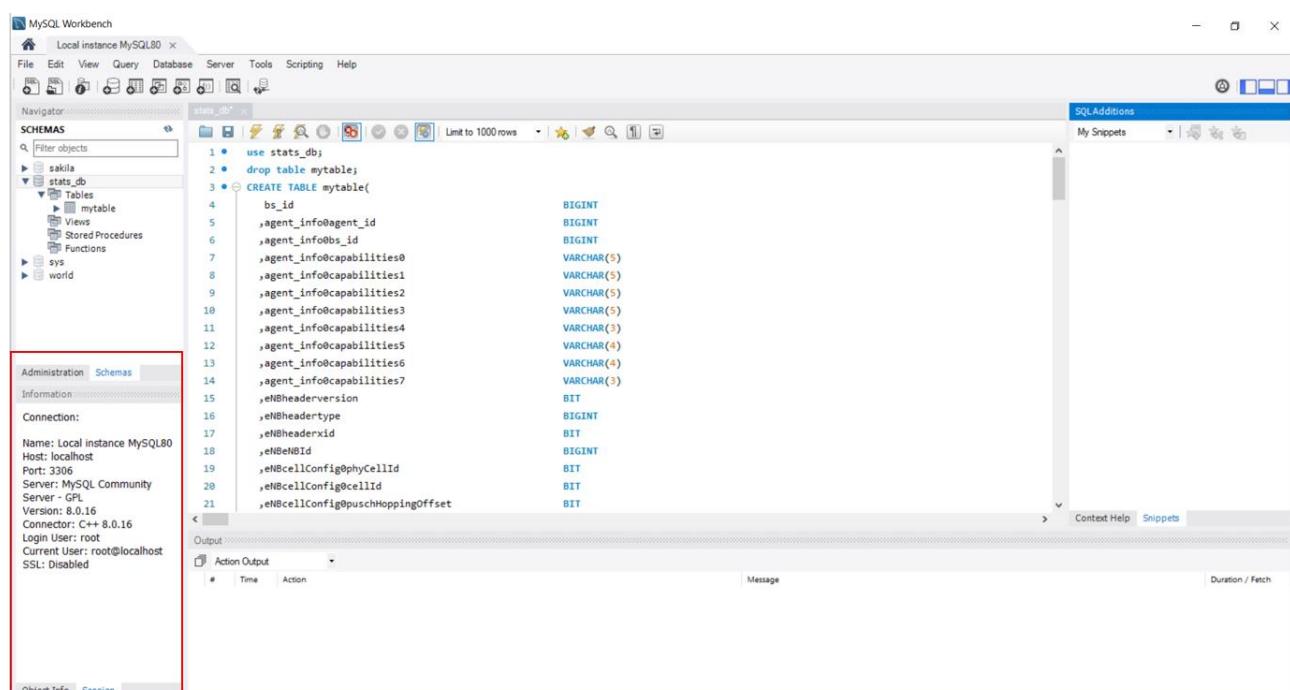


Figura 3.9 Detalii de autentificare

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 comenziile (descrise în secțiunea 2.3.1.1) :

```
1 • create database stats_db;
2 • use stats_db;
```

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(- 2^{63}) până la 9223372036854775808 (2^{63}). Domeniul de valori în cazul reprezentării fără semn este de la 0 până la 18446744073709551615(2^{64}). 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

- a) 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`, .....
,`LC1cUeConfig0lcConfig2qosBearerType`, `LC1cUeConfig0lcConfig2qci`, `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.

Action	Time	Action	Message	Duration / Fetch
32	00:49:26	INSERT INTO mytable (`bs_id` ,`agent_info0agent_id` ,`agent_info0bs_id` ,`agent_info0capabilities0` ,`agent_info0capabilities1` ,.....)	1 row(s) affected	0.000 sec
33	00:49:26	INSERT INTO mytable (`bs_id` ,`agent_info0agent_id` ,`agent_info0bs_id` ,`agent_info0capabilities0` ,`agent_info0capabilities1` ,.....)	1 row(s) affected	0.000 sec
34	00:49:26	INSERT INTO mytable (`bs_id` ,`agent_info0agent_id` ,`agent_info0bs_id` ,`agent_info0capabilities0` ,`agent_info0capabilities1` ,.....)	1 row(s) affected	0.016 sec
35	00:49:26	INSERT INTO mytable (`bs_id` ,`agent_info0agent_id` ,`agent_info0bs_id` ,`agent_info0capabilities0` ,`agent_info0capabilities1` ,.....)	1 row(s) affected	0.000 sec
36	00:49:26	INSERT INTO mytable (`bs_id` ,`agent_info0agent_id` ,`agent_info0bs_id` ,`agent_info0capabilities0` ,`agent_info0capabilities1` ,.....)	1 row(s) affected	0.015 sec
37	00:49:26	INSERT INTO mytable (`bs_id` ,`agent_info0agent_id` ,`agent_info0bs_id` ,`agent_info0capabilities0` ,`agent_info0capabilities1` ,.....)	1 row(s) affected	0.016 sec
38	00:49:26	INSERT INTO mytable (`bs_id` ,`agent_info0agent_id` ,`agent_info0bs_id` ,`agent_info0capabilities0` ,`agent_info0capabilities1` ,.....)	1 row(s) affected	0.000 sec
39	00:49:26	INSERT INTO mytable (`bs_id` ,`agent_info0agent_id` ,`agent_info0bs_id` ,`agent_info0capabilities0` ,`agent_info0capabilities1` ,.....)	1 row(s) affected	0.000 sec
40	00:49:26	INSERT INTO mytable (`bs_id` ,`agent_info0agent_id` ,`agent_info0bs_id` ,`agent_info0capabilities0` ,`agent_info0capabilities1` ,.....)	1 row(s) affected	0.000 sec
41	00:49:26	INSERT INTO mytable (`bs_id` ,`agent_info0agent_id` ,`agent_info0bs_id` ,`agent_info0capabilities0` ,`agent_info0capabilities1` ,.....)	1 row(s) affected	0.000 sec
42	00:49:26	describe mytable	125 row(s) returned	0.000 sec / 0.000 sec

Figura 3.10 Inserarea înregistrărilor

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

```
DESCRIBE mytable;
```

al cărei rezultat l-am atașat în Figura 3.11.

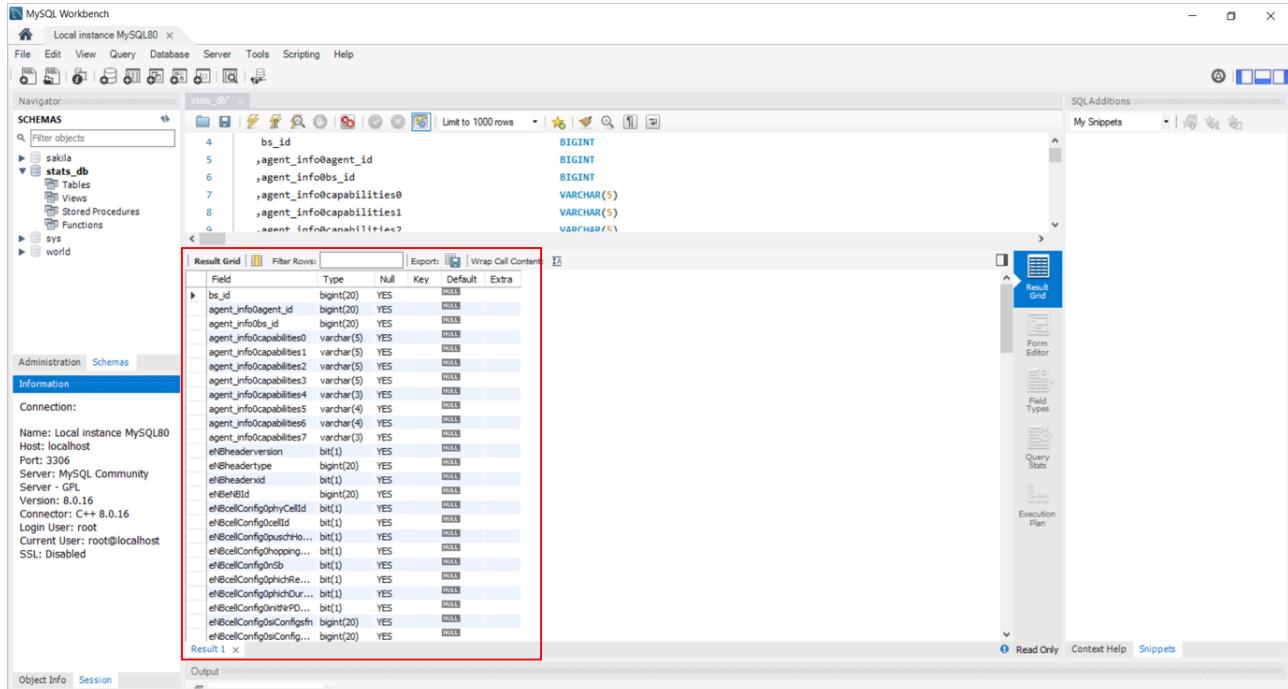


Figura 3.11 Descrierea tabelului

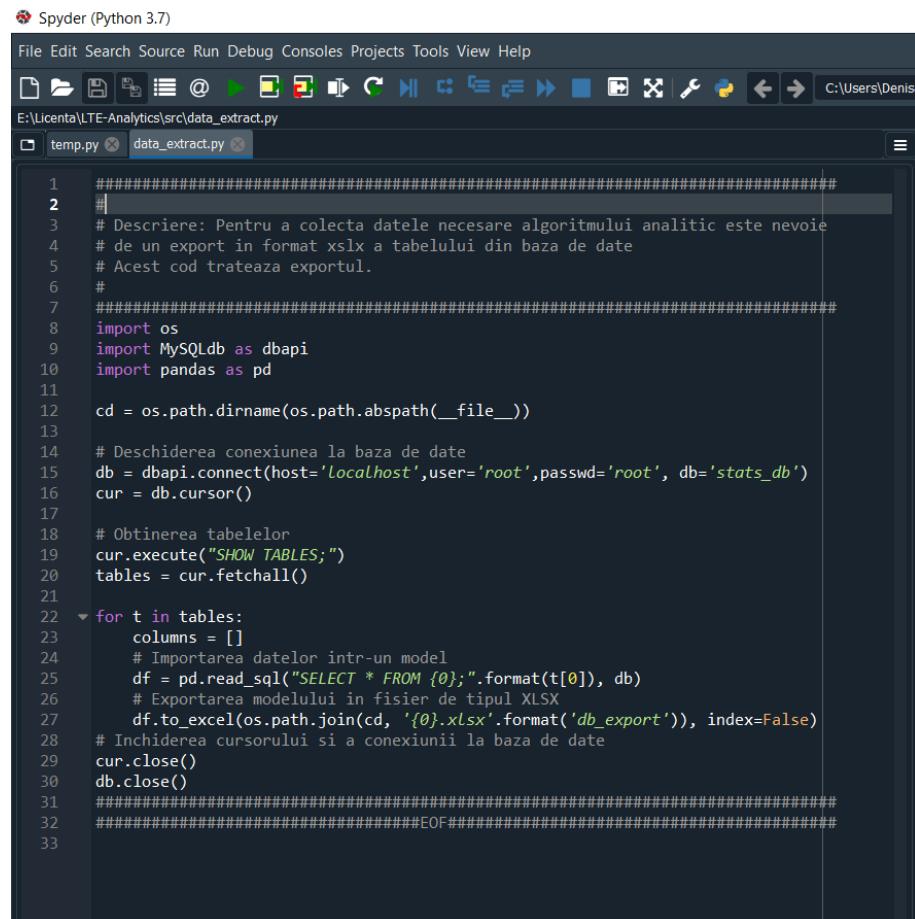
Iar mai jos, în , vom avea o vedere asupra tabelului creat, conținând numai câteva dintre coloanele pe care le conține baza de date, însă este un aspect destul de important pentru a vedea sub ce formă se prezintă datele în baza de date create.

Tabelul 4 Vedere din stats_db

bs_id	agent_infoagent_id	agent_infobs_id	agent_infocapabilities0	agent_infocapabilities1	agent_infocapabilities2	agent_infocapabilities3	agent_infocapabilities4	agent_infocapabilities5	agent_infocapabilities6	agent_infocapabilities7	e1@headerversion	e1@headertype	e1@headerid	e1@eid	e1@cellConfig@phyCellId	e1@cellConfig@cellId	e1@cellConfig@puschHoc	e1@cellConfig@hopping	e1@cellConfig@nsb	e1@cellConfig@pichRe	e1@cellConfig@pichDur	e1@cellConfig@intPD	e1@cellConfig@osConfigfn	e1@cellConfig@osConfig...
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	849	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	625	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	925	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	501	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	549	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP	RRC	8	2.3E+08	149	17	5	25	25	0	0	0	1			
10005	5	10005	LPHY	HIPHY	LOMAC	HIMAC	RLC	PDCP	SDAP															

3.2.1.2 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.



The screenshot shows the Spyder Python 3.7 IDE interface. The title bar says "Spyder (Python 3.7)". The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help. The toolbar has various icons for file operations like Open, Save, Run, and Stop. The left sidebar shows the project structure: E:\Licenta\LTE-Analytics\src\data_extract.py. The main code editor window contains the following Python script:

```
1 ######
2 #
3 # Descriere: Pentru a colecta datele necesare algoritmului analitic este nevoie
4 # de un export în format xlsx a tabelului din baza de date
5 # Acest cod tratează exportul.
6 #
7 #####
8 import os
9 import MySQLdb as dbapi
10 import pandas as pd
11
12 cd = os.path.dirname(os.path.abspath(__file__))
13
14 # Deschiderea conexiunii la baza de date
15 db = dbapi.connect(host='localhost', user='root', passwd='root', db='stats_db')
16 cur = db.cursor()
17
18 # Obținerea tabelelor
19 cur.execute("SHOW TABLES;")
20 tables = cur.fetchall()
21
22 for t in tables:
23     columns = []
24     # Importarea datelor într-un model
25     df = pd.read_sql("SELECT * FROM {0};".format(t[0]), db)
26     # Exportarea modelului în fisier de tipul XLSX
27     df.to_excel(os.path.join(cd, '{0}.xlsx'.format('db_export'))), index=False)
28     # Inchiderea cursorului și a conexiunii la baza de date
29 cur.close()
30 db.close()
31 #####
32 #####EOF#####
33 ######
```

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 preluă 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'.

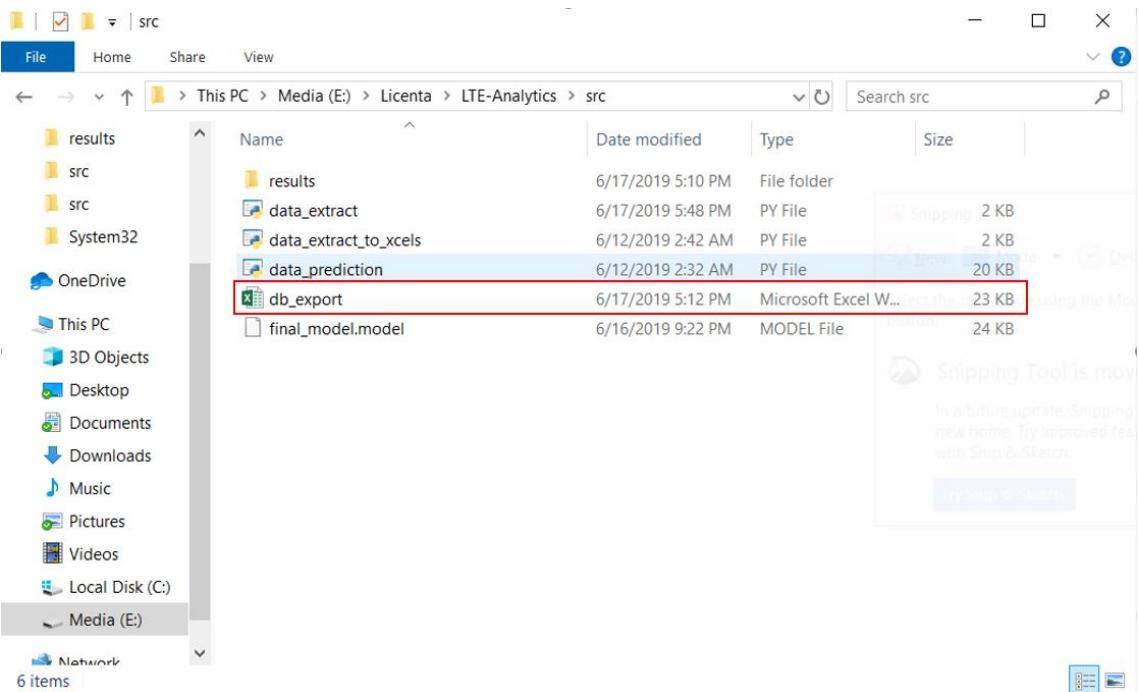


Figura 3.13 Localizarea fișierului de tip xlsx

3.2.2 Algoritmii de predicție

Așa cum este menționat în secțiunea **Eroare! Fără sursă de referință.**, 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.

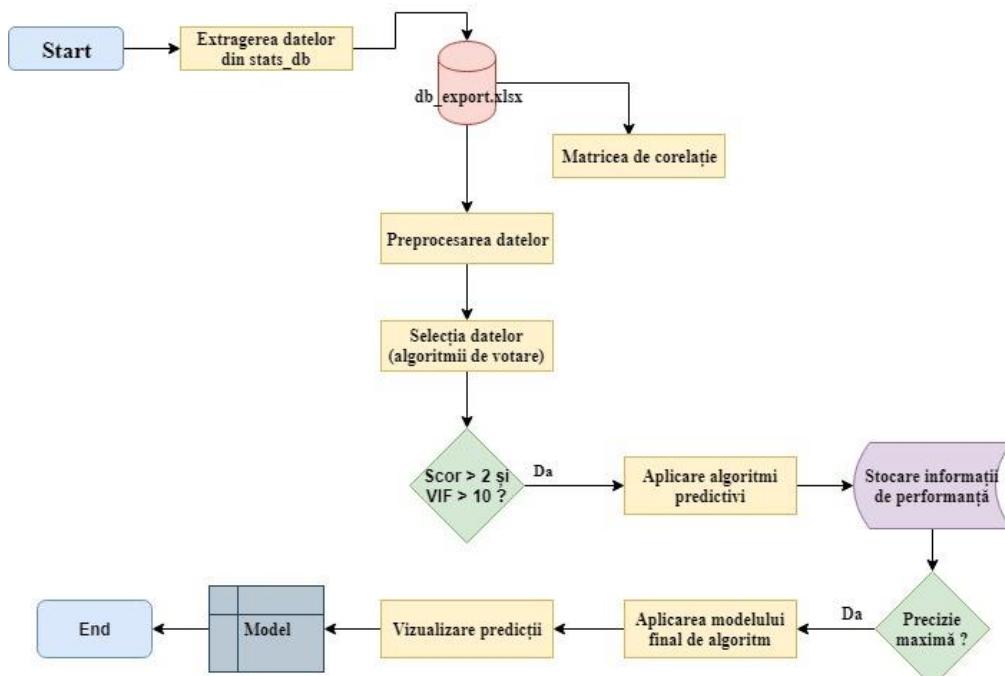


Figura 3.14 Diagrama software

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.

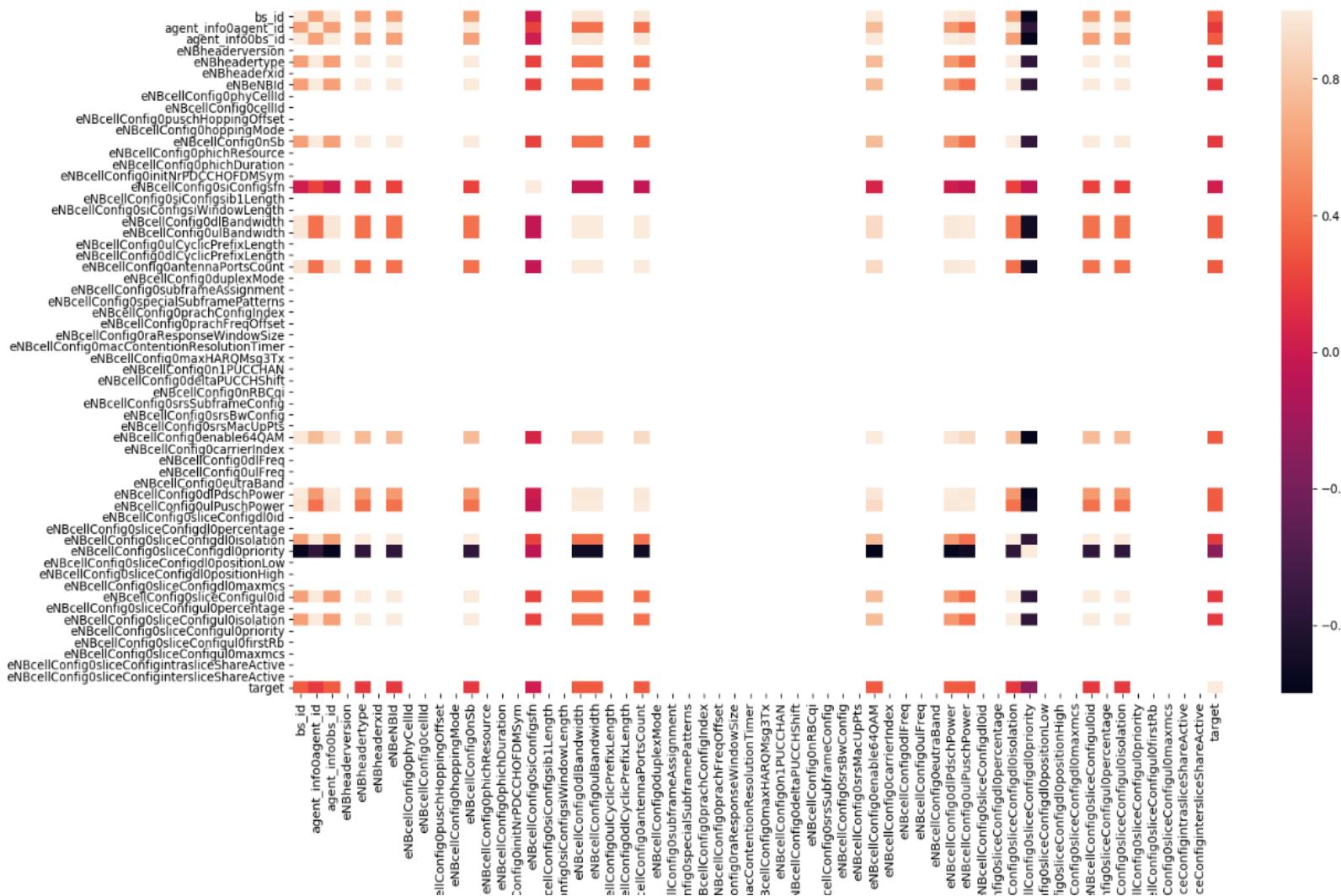


Figura 3.15 Matricea de corelație a câmpurilor din stats_db

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 preprocessarea datelor. Debarasarea datelor (data binning) este o tehnică de preprocessare a datelor utilizată pentru a reduce efectele erorilor minore de observație. Datele originale care se află într-un interval mic sunt înlocuite cu o valoare reprezentativă din acest interval, de multe ori valoarea centrală. 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 datea de influență 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. întărire). 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.

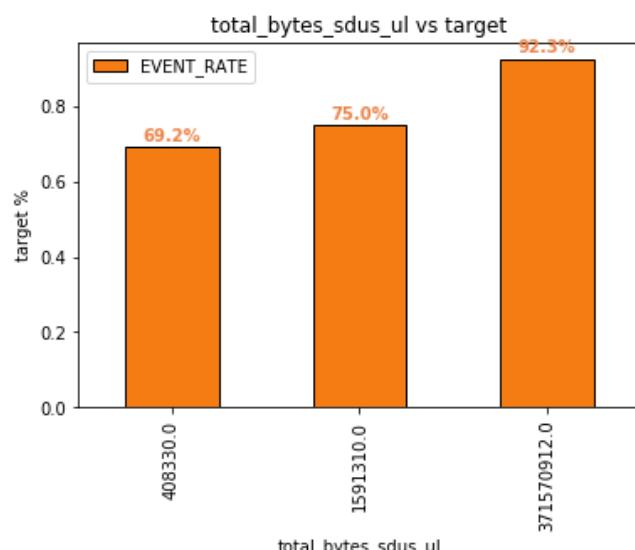


Figura 3.16 Rezultat al preprocessării datelor

În continuare va avea loc procesul de selecție a variabilelor. Acest proces este bazat pe un sistem de votare, aşa 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

1	index	IV
2	UEueConfig0capabilitiesueCategory	0.729716
3	UEueConfig0betaOffsetCQIIndex	0.729716
4	UEueConfig0ueTransmissionAntenna	0.729716
5	eNBcellConfig0dlPdschPower	0.729716
6	eNBcellConfig0ulBandwidth	0.729716
7	bs_id	0.729716
8	eNBcellConfig0antennaPortsCount	0.729716
9	eNBcellConfig0dlBandwidth	0.729716
10	agent_info0bs_id	0.729716
11	UEueConfig0simultaneousAckNackCqi	0.619229
12	total_bytes_sdus_ul	0.438767
13	total_bytes_sdus_dl	0.102023
14	eNBcellConfig0sliceConfig0id	0.01908
15	eNBcellConfig0sliceConfig0isolation	0.01908
16	eNBcellConfig0sliceConfig0isolation	0.01908
17	agent_info0agent_id	0.01908
18	eNBcellConfig0enable64QAM	0.01908
19	eNBBeNBId	0.01908
20	eNBheadertype	0.01908
21	eNBcellConfig0siConfigsfn	0.010536
22	eNBcellConfig0sliceConfig0label	0
23	eNBcellConfig0sliceConfig0accounting	0

Tabelul 6 Rezultatele votului Random Forests

1	index	RF
2	total_bytes_sdus_dl	0.425989
3	total_bytes_sdus_ul	0.402066
4	eNBcellConfig0siConfigsfn	0.062173
5	eNBcellConfig0dlBandwidth	0.024856
6	UEueConfig0capabilitiesueCategory	0.02024
7	eNBcellConfig0ulBandwidth	0.016875
8	eNBcellConfig0sliceConfig0isolation	0.012755
9	bs_id	0.012142
10	eNBcellConfig0sliceConfig0priority	0.010114
11	UEueConfig0simultaneousAckNackCqi	0.005816
12	agent_info0agent_id	0.002848
13	eNBcellConfig0sliceConfig0isolation	0.002044
14	eNBcellConfig0ulPuschPower	0.001979
15	UEueConfig0betaOffsetCQIIndex	0.000103
16	eNBcellConfig0sliceConfig0accounting	0
17	eNBcellConfig0macContentionResolutionTimer	0
18	eNBcellConfig0sliceConfig0positionLow	0
19	eNBcellConfig0sliceConfig0positionHigh	0
20	eNBcellConfig0sliceConfig0percentage	0
21	eNBcellConfig0sliceConfig0maxmcs	0
22	eNBcellConfig0maxHARQMsg3Tx	0
23	eNBcellConfig0sliceConfig0label	0

Tabelul 7 Rezultatul votului ExtraTrees

1	index	Extratrees
2	total_bytes_sdus_dl	0.485101
3	total_bytes_sdus_ul	0.317095
4	eNBcellConfig0siConfigsfn	0.099004
5	eNBcellConfig0dlBandwidth	0.026671
6	UEueConfig0capabilitiesueCategory	0.025957
7	bs_id	0.010945
8	eNBcellConfig0sliceConfig0priority	0.009588
9	eNBcellConfig0ulBandwidth	0.009588
10	eNBcellConfig0sliceConfig0isolation	0.003467
11	eNBcellConfig0enable64QAM	0.003137
12	eNBcellConfig0ulPuschPower	0.003137
13	UEueConfig0ueTransmissionAntenna	0.002639
14	UEueConfig0simultaneousAckNackCqi	0.002351
15	eNBcellConfig0dlPdschPower	0.00033
16	eNBcellConfig0sliceConfig0id	0.00033
17	agent_info0agent_id	0.00033
18	agent_info0bs_id	0.00033
19	eNBcellConfig0phyCellId	0
20	eNBcellConfig0sliceConfig0positionLow	0
21	eNBcellConfig0sliceConfig0positionHigh	0
22	eNBcellConfig0sliceConfig0percentage	0
23	eNBcellConfig0sliceConfig0maxmcs	0

Tabelul 8 Rezultatul votului Chi Square

1	index	Chi_Square
2	total_bytes_sdus_ul	574835720.6
3	total_bytes_sdus_dl	43230433.68
4	eNBcellConfig0ulBandwidth	56.67200855
5	eNBcellConfig0dlBandwidth	56.67200855
6	eNBcellConfig0dlPdschPower	19.97257236
7	eNBcellConfig0ulPuschPower	5.246590909
8	eNBcellConfig0enable64QAM	3.100833333
9	eNBcellConfig0siConfigsfn	2.796111182
10	UEueConfig0simultaneousAckNackCqi	1.867777778
11	UEueConfig0ueTransmissionAntenna	1.25
12	eNBcellConfig0sliceConfig0id	1.066666667
13	eNBcellConfig0sliceConfig0isolation	1.066666667
14	eNBcellConfig0sliceConfig0isolation	1.066666667
15	UEueConfig0capabilitiesueCategory	0.733695652
16	eNBcellConfig0antennaPortsCount	0.625
17	eNBcellConfig0sliceConfig0priority	0.172268519
18	UEueConfig0betaOffsetCQIIndex	0.10546875
19	agent_info0agent_id	0.021993127
20	eNBheadertype	0.013852814
21	bs_id	0.000492514
22	agent_info0bs_id	0.000492514
23	eNBBeNBId	4.78032E-10
24	eNBcellConfig0sliceConfig0maxmcs	0

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

Tabelul 9 Resultatele tuturor voturilor

1	index	IV	RF	RFE	Extratrees	Chi_Square	L1
2	UEueConfig0capabilitiesueCategory	0.729716	0.016875	FALSE	0.023149417	0.733695652	FALSE
3	UEueConfig0betaOffsetCQIIndex	0.729716	0.001325	FALSE	0.019176136	0.10546875	FALSE
4	UEueConfig0ueTransmissionAntenna	0.729716	0	FALSE	0	1.25	FALSE
5	eNBcellConfig0dlPdschPower	0.729716	0.022675	FALSE	0.001041667	19.97257236	FALSE
6	eNBcellConfig0ulBandwidth	0.729716	0.00843	TRUE	0.006780674	56.67200855	FALSE
7	bs_id	0.729716	0	TRUE	0.000329861	0.000492514	TRUE
8	eNBcellConfig0antennaPortsCount	0.729716	0.002889	FALSE	0	0.625	FALSE
9	eNBcellConfig0dlBandwidth	0.729716	0	TRUE	0.01386979	56.67200855	FALSE
10	agent_info0bs_id	0.729716	0	TRUE	0.012725323	0.000492514	TRUE
11	UEueConfig0simultaneousAckNackCqi	0.619229	0.013088	FALSE	0.000627736	1.867777778	FALSE
12	total_bytes_sdus_ul	0.438767	0.36915	TRUE	0.339535911	574835720.6	FALSE
13	total_bytes_sdus_dl	0.102023	0.379317	TRUE	0.453109634	43230433.68	FALSE
14	eNBcellConfig0sliceConfig0id	0.01908	0.00266	FALSE	0.004079715	1.066666667	FALSE
15	eNBcellConfig0sliceConfig0isolation	0.01908	0	FALSE	0	1.066666667	FALSE
16	eNBcellConfig0sliceConfig0isolation	0.01908	0.0005	FALSE	0	1.066666667	FALSE
17	agent_info0agent_id	0.01908	0.001	FALSE	0.003281194	0.021993127	FALSE
18	eNBcellConfig0enable64QAM	0.01908	0.014727	FALSE	0.003137255	3.100833333	FALSE
19	eNBcellConfig0id	0.01908	0	TRUE	0	4.78032E-10	FALSE
20	eNBheadertype	0.01908	0	FALSE	0	0.013852814	FALSE
21	eNBcellConfig0siConfigsfn	0.010536	0.149919	TRUE	0.100529868	2.796111182	FALSE
22	eNBcellConfig0sliceConfig0label	0	0	FALSE	0		FALSE
23	eNBcellConfig0sliceConfig0accounting	0	0	FALSE	0		FALSE

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

1	index	IV	RF	Extratrees	Chi_Square	RFE	L1	final_score
2	total_bytes_sdus_ul	0	1	1	1	1	0	4
3	total_bytes_sdus_dl	0	1	1	1	1	0	4
4	eNBcellConfig0ulBandwidth	1	0	0	1	1	0	3
5	eNBcellConfig0siConfigsfn	0	1	1	0	1	0	3
6	UEueConfig0capabilitiesueCategory	1	1	1	0	0	0	3
7	eNBcellConfig0dlPdschPower	1	1	0	1	0	0	3
8	bs_id	0	0	0	0	1	1	2
9	UEueConfig0betaOffsetCQIIndex	1	0	1	0	0	0	2
10	eNBcellConfig0dlBandwidth	0	0	0	1	1	0	2
11	agent_info0bs_id	0	0	0	0	1	1	2
12	UEueConfig0rtti	0	0	0	0	1	1	2
13	eNBcellConfig0ulFreq	0	0	0	0	1	1	2
14	eNBcellConfig0ulPuschPower	0	0	0	0	1	1	2
15	LCIcUeConfig0rtti	0	0	0	0	1	1	2
16	eNBcellConfig0sliceConfig0maxmcs	0	0	0	0	1	0	1
17	eNBcellConfig0sliceConfig0maxmcs	0	0	0	0	1	0	1
18	eNBcellConfig0sliceConfig0percentage	0	0	0	0	1	0	1
19	UEueConfig0imsi	0	0	0	0	1	0	1
20	eNBcellConfig0sliceConfig0positionHigh	0	0	0	0	1	0	1
21	eNBcellConfig0siConfig0b1Length	0	0	0	0	1	0	1
22	eNBcellConfig0sliceConfig0percentage	0	0	0	0	1	0	1
23	eNBcellConfig0dlFreq	0	0	0	0	1	0	1
24	UEueConfig0ueTransmissionAntenna	1	0	0	0	0	0	1
25	eNBcellConfig0id	0	0	0	0	1	0	1
26	agent_info0capabilities1	0	0	0	0	0	0	0
27	LCIcUeConfig0lcConfig1qosBearerType	0	0	0	0	0	0	0

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 mulți, 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',
 'LCIcUeConfig0rnti',
 '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 împărti setul de date în date de învățare și date de test și voi încerca performanțele unei serii de algoritmi 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 Variația creșterii (Gradient Boosting).

a) 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

b) Algoritmul variației creșterii 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 Predictii pe datele de invatare

PREZIS	0	1
ACTUAL		
0	3	0
1	0	19

Tabelul 14 Predictii pe datele de test

PREZIS	0	1
ACTUAL		
0	1	4
1	1	10

- c) Algoritmul regresiei liniare ofera urmatoarele performante :

Precizia pentru datele de invatare: 86.36363636363636 %

Precizia pentru datele de test: 62.5 %

Aria de sub grafic pentru datele de invatare: 57.89473684210527 %

Aria de sub grafic pentru datele de test: 74.54545454545455 %

Tabelul 15 Predictii pe datele de invatare

PREZIS	0	1
ACTUAL		
0	3	0
1	0	19

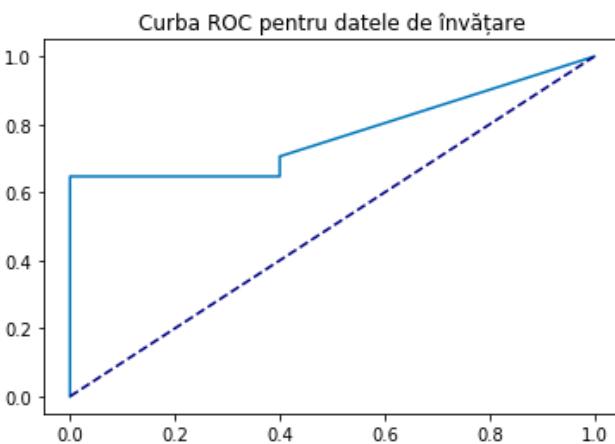
Tabelul 16 Predictii pe datele de test

PREZIS	0	1
ACTUAL		
0	0	5
1	1	10

In continuare, voi decurge la reglarea performantei pentru primele doua modele de algoritmi, iar rezultatele sunt afisate mai jos :

- a) Precizia pentru datele de invatare: 86.36363636363636 %
 Precizia pentru datele de test: 68.75 %
 Aria de sub grafic pentru datele de invatare: 97.36842105263158 %
 Aria de sub grafic pentru datele de test: 51.81818181818182 %
- b) Precizia pentru datele de invatare: 95.45454545454546 %
 Precizia pentru datele de test: 75 %
 Aria de sub grafic pentru datele de invatare: 100 %
 Aria de sub grafic pentru datele de test: 65.4545454545454 %

Atunci cand dorim sa alegem cel mai precis algoritm de predictie pentru setul de date de care disponem, tinem cont de aspectul curbei ROC si de aria graficului delimitat de aceasta.

**Figura 3.17 Curba ROC - Pădurile aleatoare**

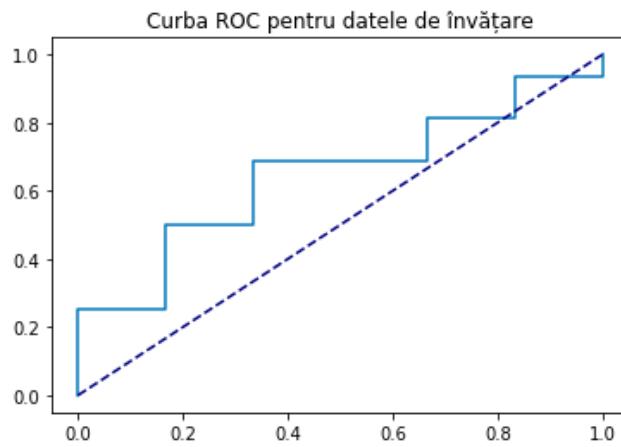


Figura 3.18 Curba ROC - Regresia logistică

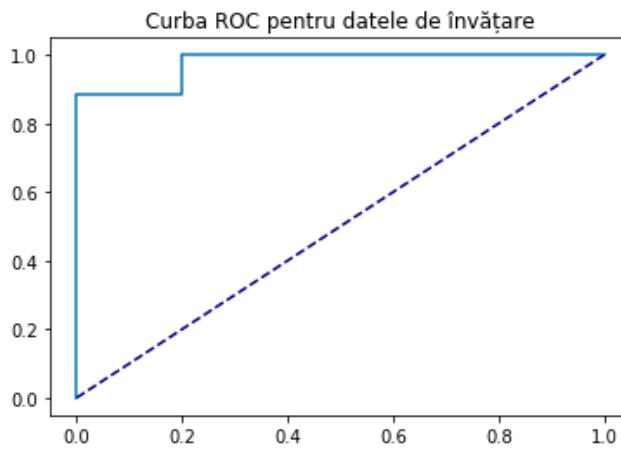


Figura 3.19 Curba ROC - Variația creșterii

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

3.2.3 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 diferenților algoritmi aplicați.

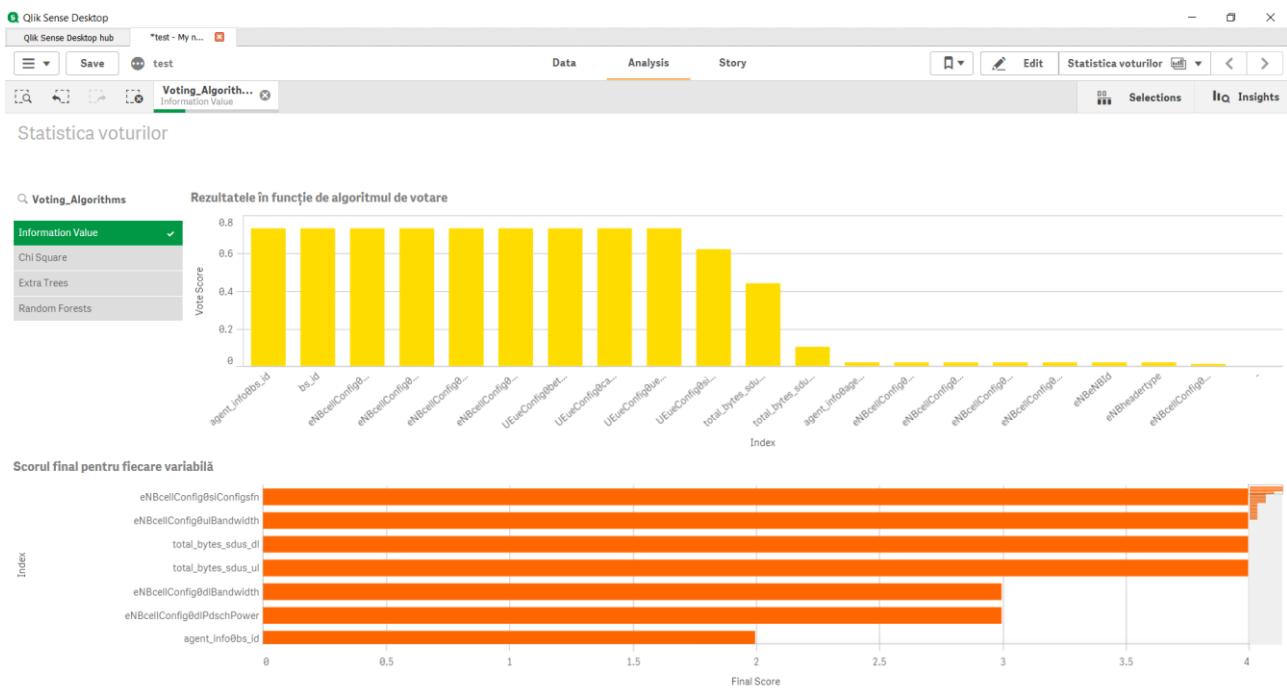


Figura 3.20 Rezultatele votului afișate în Qlik Sense

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

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

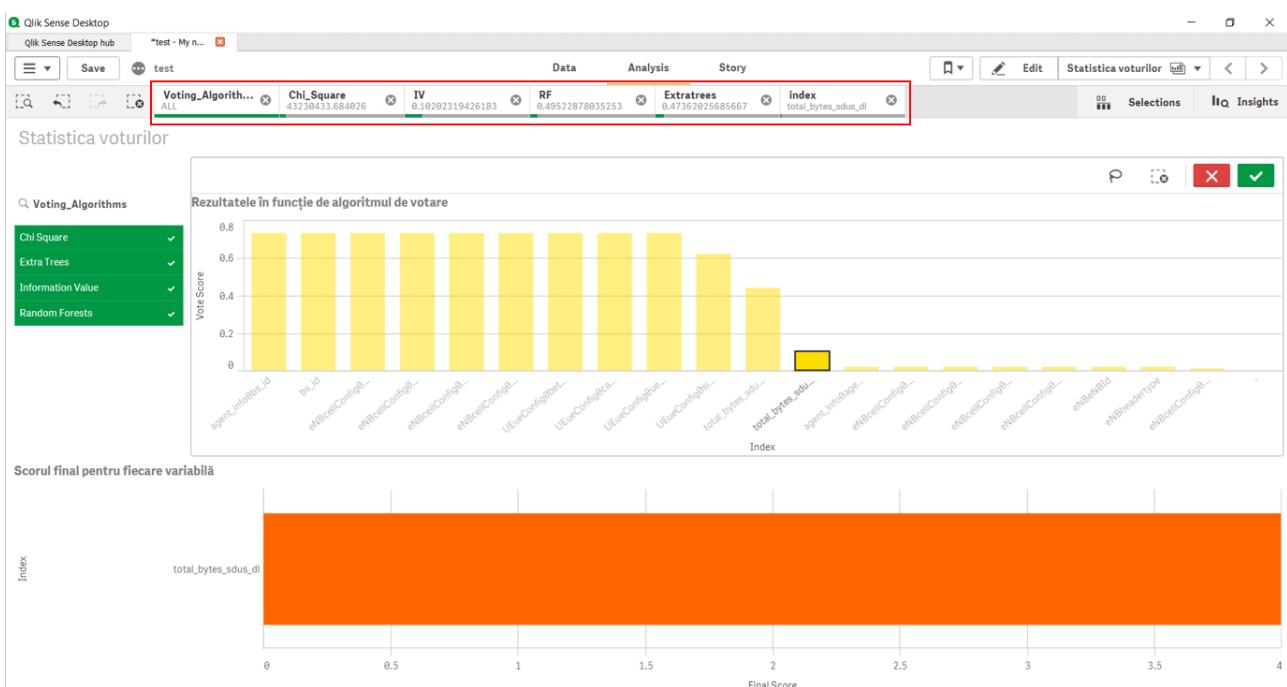
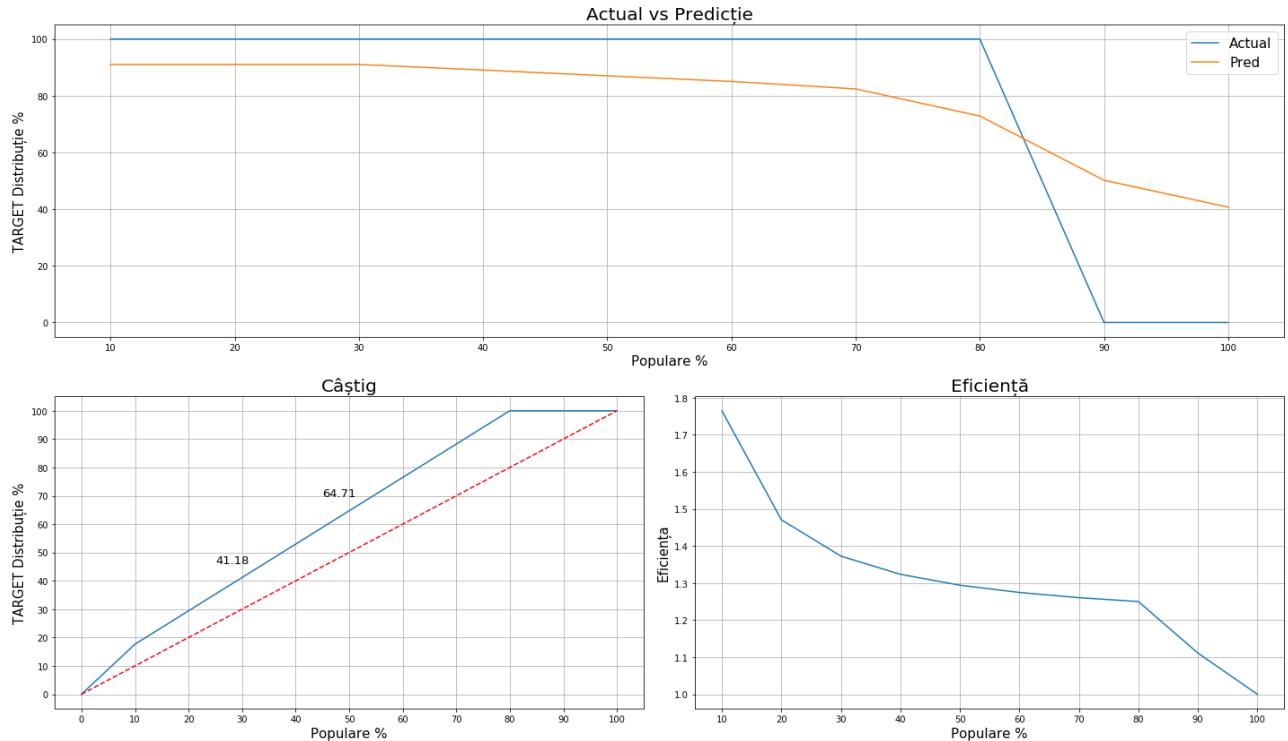


Figura 3.21 Rezultatele filtrate per index

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



În graficul comparativ ‘Actual vs Predicție’, curba corespunzătoare comportamentului actual conține înregistrări colectate din rețea la fiecare jumătate de minut, timp de 18min și 30s, iar predicția se va face pentru următoarea jumătate de minut 18m:30s – 19m:00s. 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%, 41%) 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 printre aceste valori să regăsim aproximativ 41% din toate cazurile care vor lua de fapt valoarea 1. În mod similar, primele 50% dintre valori ar conține aproximativ 65% din cazurile care vor lua valoarea 1 a câmpului TARGET.

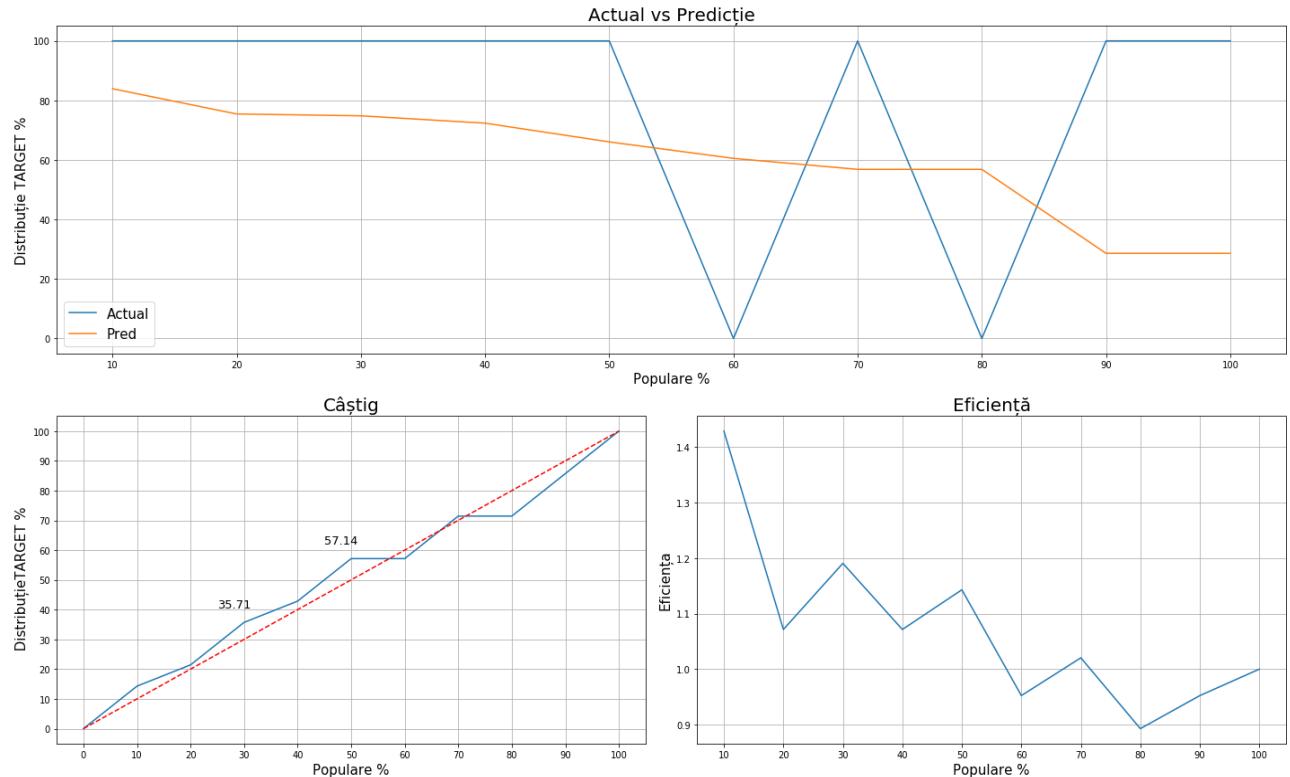
Linia diagonală este curba de bază. Ea indică faptul că dacă selectăm la întâmplare 10% dintre valorile din setul de date, ne așteptăm să regăsim printre acestea aproximativ 10% dintre toate cazurile care conțin valoarea 1 în câmpul TARGET.

Așadar, ținând cont de aceste două aspecte, putem afirma faptul că vom obține un câștig mai bun cu cât linia curbată se va depărtă 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% eficiență pentru categoria TARGET este de

$41\% / 30\% = 1.36$. Putem spune că această diagramă este un alt mod de a privi informațiile din graficul câștigurilor cumulative.

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



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 seminificativă de date. Putem afirma deci, că algoritmul predictiv și-a atins scopul.

Capitolul 4 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 :

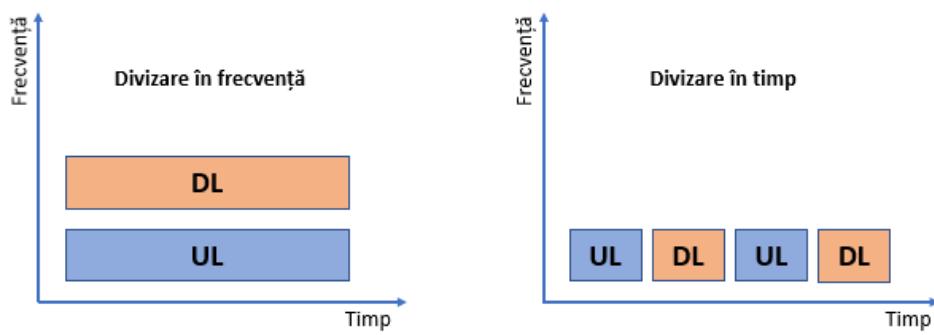
```
['eNBcellConfig0dlBandwidth',
'total_bytes_sdus_ul',
'total_bytes_sdus_dl',
'eNBcellConfig0siConfigsfn',
'eNBcellConfig0ulPuschPower',
'eNBcellConfig0ulFreq',
'LCIcUeConfig0rnti',
'eNBcellConfig0dlFreq']
```

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 aceleasi 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 descendantă 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 largimii 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).



DL – Cale descendătoare; UL – Cale ascendentă

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

Config.	Perioada	Nr.									
		0	1	2	3	4	5	6	7	8	9
0	5 ms	D	S	U	U	U	D	S	U	U	U
1	5 ms	D	S	U	U	D	D	S	U	U	D
2	5 ms	D	S	U	D	D	D	S	U	D	D
3	10 ms	D	S	U	U	U	D	D	D	D	D
4	10 ms	D	S	U	U	D	D	D	D	D	D
5	10 ms	D	S	U	D	D	D	D	D	D	D
6	5 ms	D	S	U	U	U	D	S	U	U	D

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 (uplink), 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.

Stim 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, frecvențe corespunzătoare LTE.

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

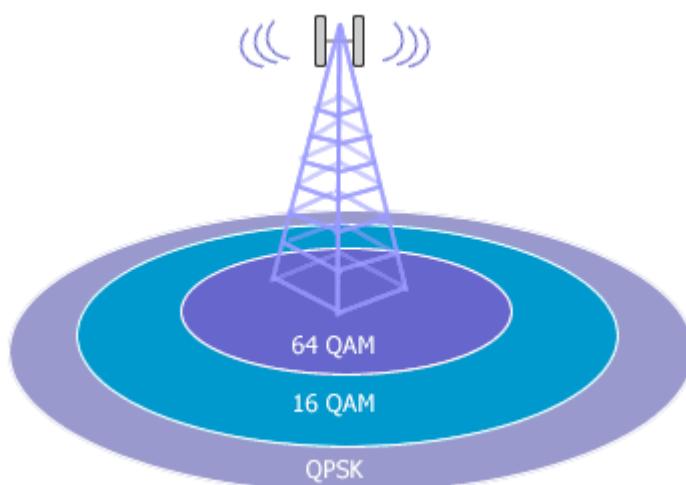


Figura 4.2 Schemele de modulație din LTE

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 viteză 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

Problema îmbunătățirii performanțelor unei rețele de comunicații mobile reprezintă un subiect de actualitate, urmărit de orice furnizor de servicii mobile.

Așa cum mi-am propus în introducere, în urma modelului predictiv aplicat setului de date de care am dispus, am reușit să identific eventualele degradări din rețeaua LTE și parametrii care au influențat aceste fenomene.

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 utilizati 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 aceste. Prin performanță mă refer la precizie, câștig, eficiență, curba ROC și aria acoperită de aceasta.

Printre parametrii care influențează în mod considerabil rețeaua LTE se numără frecvența, lățimea de bandă, cantitatea de informație încărcată/descărcată de către utilizator și puterea recepționată de echipamentul utilizatorului. Este de preferat ca dimensionarea parametrilor de configurare ai rețelei se efectuează în funcție de condițiile în care se află utilizatorul și mai ales de nevoile sale, însă evident că acest lucru nu poate fi realizat în mod individual, ci particularizat pe un grup de utilizatori care se află în aceleași condiții. 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 aşa 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_Alcotel_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] Marianne Fjelberg, The Royal Institute of Technology, “Predicting data traffic in cellular data networks”, Iunie 2015
- [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://helpqlik.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)

ANEXA 1

```
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
,eNBBeNBId                            BIGINT
,eNBcellConfig0phyCellId               BIT
,eNBcellConfig0cellId                 BIT
,eNBcellConfig0puschHoppingOffset    BIT
,eNBcellConfig0hoppingMode             BIT
,eNBcellConfig0nSb                     BIT
,eNBcellConfig0phichResource          BIT
,eNBcellConfig0phichDuration          BIT
,eNBcellConfig0initNrPDCCHOFDMSym   BIT
,eNBcellConfig0siConfigsfn            BIGINT
,eNBcellConfig0siConfigsiLength       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
,UEueConfig0capabilitiesstype2Sb1	BIT
,UEueConfig0capabilitiesueCategory	BIGINT
,UEueConfig0capabilitiesresAllocType1	BIT
,UEueConfig0ueTransmissionAntenna	BIGINT
,UEueConfig0ttiBundling	BIT
,UEueConfig0maxHARQTx	BIGINT
,UEueConfig0betaOffsetACKIndex	BIT
,UEueConfig0betaOffsetRIIIndex	BIT
,UEueConfig0betaOffsetCQIIndex	BIGINT
,UEueConfig0ackNackSimultaneousTrans	BIT
,UEueConfig0simultaneousAckNackCqi	BIT
,UEueConfig0aperiodicCqiRepMode	BIGINT
,UEueConfig0ackNackRepetitionFactor	BIT
,UEueConfig0pcellCarrierIndex	BIT
,UEueConfig0imsi	BIGINT
,UEueConfig0dlsliceId	BIT
,UEueConfig0ulsliceId	BIT
,Lheaderversion	BIT
,Lheadertype	BIGINT
,Lheaderxid	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

```

,`LC1cUeConfig0lcConfig2direction` BIT
,`LC1cUeConfig0lcConfig2qosBearerType` BIT
,`LC1cUeConfig0lcConfig2qci` 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`, `eNB eNBId`, `eNBcellConfig0phyCellId`, `eNBcellConfig0cellId`, `eNBcellConfig0puschHoppingOffset`, `eNBcellConfig0hoppingMode`, `eNBcellConfig0nSb`, `eNBcellConfig0phichResource`, `eNBcellConfig0phichDuration`, `eNBcellConfig0initNRPDCCHOFDMSym`, `eNBcellConfig0siConfigsfn`, `eNBcellConfig0siConfigsib1Length`, `eNBcellConfig0siConfig siWindowLength`, `eNBcellConfig0dlBandwidth`, `eNBcellConfig0ulBandwidth`, `eNBcellConfig0ulCyclicPrefixLength`, `eNBcellConfig0dlCyclicPrefixLength`, `eNBcellConfig0antennaPortsCount`, `eNBcellConfig0duplexMode`, `eNBcellConfig0subframeAssignment`, `eNBcellConfig0specialSubframePatterns`, `eNBcellConfig0prachConfigIndex`, `eNBcellConfig0prachFreqOffset`, `eNBcellConfig0raResponseWindowSize`, `eNBcellConfig0m acContentionResolutionTimer`, `eNBcellConfig0maxHARQMsg3Tx`, `eNBcellConfig0n1PUCC HAN`, `eNBcellConfig0deltaPUCCHShift`, `eNBcellConfig0nRBCqi`, `eNBcellConfig0srsSu bframeConfig`, `eNBcellConfig0srsBwConfig`, `eNBcellConfig0srsMacUpPts`, `eNBcellCo nfig0enable64QAM`, `eNBcellConfig0carrierIndex`, `eNBcellConfig0dlFreq`, `eNBcellCo nfig0ulFreq`, `eNBcellConfig0utraBand`, `eNBcellConfig0dlPdschPower`, `eNBcellConf ig0ulPuschPower`, `eNBcellConfig0sliceConfigd10id`, `eNBcellConfig0sliceConfigd10 label`, `eNBcellConfig0sliceConfigd10percentage`, `eNBcellConfig0sliceConfigd10isol ation`, `eNBcellConfig0sliceConfigd10priority`, `eNBcellConfig0sliceConfigd10posit ionLow`, `eNBcellConfig0sliceConfigd10positionHigh`, `eNBcellConfig0sliceConfigd10 maxmcs`, `eNBcellConfig0sliceConfigd10sorting0`, `eNBcellConfig0sliceConfigd10sort ing1`, `eNBcellConfig0sliceConfigd10sorting2`, `eNBcellConfig0sliceConfigd10sortin g3`, `eNBcellConfig0sliceConfigd10sorting4`, `eNBcellConfig0sliceConfigd10sorting5`, `eNBcellConfig0sliceConfigd10accounting`, `eNBcellConfig0sliceConfigd10schedule rName`, `eNBcellConfig0sliceConfigul0id`, `eNBcellConfig0sliceConfigul0label`, `eNB cellConfig0sliceConfigul0percentage`, `eNBcellConfig0sliceConfigul0isolation`, `eNBcellConfig0sliceConfigul0priority`, `eNBcellConfig0sliceConfigul0firstRb`, `eNBcellConfig0sliceConfigul0maxmcs`, `eNBcellConfig0sliceConfigul0accounting`, `eNBcellConfig0sliceConfigul0schedulerName`, `eNBcellConfig0sliceConfigintrasliceShareAct ive`, `eNBcellConfig0sliceConfigintersliceShareActive`, `UEueConfig0rnti`, `UEueCon fig0timeAlignmentTimer`, `UEueConfig0transmissionMode`, `UEueConfig0ueAggregatedMa xBitrateUL`, `UEueConfig0ueAggregatedMaxBitrateDL`, `UEueConfig0capabilitieshalfDu plex`, `UEueConfig0capabilitiesintraSFHopping`, `UEueConfig0capabilitiesstype2Sb1`, `UEueConfig0capabilitiesueCategory`, `UEueConfig0capabilitiesresAllocType1`, `UEue Config0ueTransmissionAntenna`, `UEueConfig0ttiBundling`, `UEueConfig0maxHARQTx`, `UE ueConfig0betaOffsetACKIndex`, `UEueConfig0betaOffsetRIIndex`, `UEueConfig0betaOff setCQIIndex`, `UEueConfig0ackNackSimultaneousTrans`, `UEueConfig0simultaneousAckNa ckCqi`, `UEueConfig0aperiodicCqiRepMode`, `UEueConfig0ackNackRepetitionFactor`, `UE ueConfig0pcellCarrierIndex`, `UEueConfig0imsi`, `UEueConfig0dlSliceId`, `UEueConfig 0ulSliceId`, `LChederversion`, `LChedertype`, `LChederxid`, `LC1cUeConfig0rnti`, `LC1cUeConfig0lcConfig0lcid`, `LC1cUeConfig0lcConfig0lcg`, `LC1cUeConfig0lcConfig0d irection`, `LC1cUeConfig0lcConfig0qosBearerType`, `LC1cUeConfig0lcConfig0qci`, `LC1 cUeConfig0lcConfig1lcid`, `LC1cUeConfig0lcConfig1lcg`, `LC1cUeConfig0lcConfig1dire ction`, `LC1cUeConfig0lcConfig1qosBearerType`, `LC1cUeConfig0lcConfig1qci`, `LC1cUe Config0lcConfig2lcid`, `LC1cUeConfig0lcConfig2lcg`, `LC1cUeConfig0lcConfig2directi on`, `LC1cUeConfig0lcConfig2qosBearerType`, `LC1cUeConfig0lcConfig2qci`, `total_byt es_sdus_ul`, `total_bytes_sdus_dl`, `y`) VALUES
(10005, 5, 10005, 'LOPHY', 'HIPHY', 'LOMAC', 'HIMAC', 'RLC', 'PDCP', 'SDAP', 'RRC', 0, 8, 0, 2 34881024, 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, 26 85, 2565, 7, -27, -
96, 0, 'xMBB', 100, 'FALSE', 10, 0, 25, 28, 'CR_ROUND', 'CR_SRB12', 'CR_HOL', 'CR_LC', 'CR_CQ I', '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, 2090000000000000, 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;
```

ANEXA 2

data_extract.py

```
#####
#
# Descriere: Pentru a colecta datele necesare algoritmului analitic este nevoie
# de un export in format xlsx 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#####
#####
```


ANEXA 3

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 elibera 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
#atarea 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)
        imparte in 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 prezinta 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
        else:
            conv =
conv[["VAR_NAME"] = i
                count = count + 1
    return(conv)

    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'],ascendi
ng=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',ascen
ding=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(c
d,
'{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.v
alues, 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)[F
eatures'][1:]

features.drop(remove, axis=1, inplace=T
rue)
    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)

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=[

# 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.be
st_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_trai
n),
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=[


```

```

#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(pre
d_train),rownames=['ACTUAL'],colnames
=['PRED'])
pd.crosstab(label_test,pd.Series(pred
_test),rownames=['ACTUAL'],colnames=[

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',labe
l = '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=[

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 Predictie',
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("Eficiență", 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()/groupe
d.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_trai
n],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)

```

ANEXA 4

Scriptul Qlik Sense Desktop :

```
//Voturi                                //Predictie

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_Chisquare.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"
];
```

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);
```