

Universiteti i Prishtinës “Hasan Prishtina”

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike



Dokumentim teknik i projektit

Lënda: Big Data

Titulli i projektit: Manipulim me Big Data, NoSQL dhe analizë e grafeve

Emri profesorit/Asistentit	Emri & mbiemri studentëve / email adresa	
Prof. Dr. Vigan Raça MSc. Rafet Duriqi	1. Fjolla Kadriu	fjolla.kadriu@student.uni-pr.edu
	2. Leutrim Morina	leutrim.morina13@student.uni-pr.edu
	3. Linda Hasanaj	linda.hasanaj@student.uni-pr.edu
	4. Lum Hoxha	lum.hoxha@student.uni-pr.edu

Prishtinë, 2025

Përmbajtja

Abstrakti	3
1. Hyrje	4
2. Qëllimi i punimit	5
3. Pjesa kryesore	6
3.1. NoSQL Data	6
3.1.1. Instalimi dhe konfigurimi i MongoDB	6
3.1.2. Migrimi i të dhënave dhe skemës	7
3.1.3. Shkrimi i query-ve në NoSQL	8
3.2. Big Data	11
3.2.1. Instalimi dhe konfigurimi i Apache Spark	11
3.2.2. Importimi i dataset-it Mondial	11
3.2.3. Ekzekutimi i query-ve	12
3.2.4. Importimi i një dataset-i të ri	14
3.3. Network analysis (Teoria e grafeve)	18
3.3.1. Instalimi dhe konfigurimi i ORA Lite	18
3.3.2. Importimi i të dhënave	18
3.3.3. Gjenerimi i skemës vizuale	18
3.3.4. Kalkulimi i metrikave të rrjetit	20
4. Konkluzione	26
Referencat	27

Abstrakti

Ky punim trajton përpunimin dhe analizën e të dhënave të mëdha (Big Data) dhe të dhënave të strukturuar në formë rrjeti (Network Data), duke përdorur teknologji moderne si MongoDB për NoSQL dhe Apache Spark për përpunim të paralelizuar. Tema është relevante për shkak të rritjes eksponenciale të të dhënave dhe nevojës për mjete efikase për menaxhimin dhe analizën e tyre. Në pjesën e parë, realizohet migrimi i të dhënave nga një sistem relacionar në MongoDB, ku përmes query-ve të avancuara janë analizuar lidhje komplekse si shtetet me dalje në det dhe anëtarësimi në NATO/BE. Në pjesën e dytë, përmes Apache Spark janë analizuar dy dataset-e: Mondial dhe një dataset i madh me të dhëna COVID-19, ku janë realizuar query dhe vizualizime mbi përhapjen e pandemisë dhe përqindjen e vdekjeve në SHBA. Në pjesën e fundit është zbatuar analiza e rrjeteve me ORA Lite për një rrjet social, ku janë llogaritur metrikat kryesore si Degree, Closeness dhe Betweenness Centrality. Literatura ekzistuese fokusohet në përdorimin individual të këtyre teknologjive, ndërsa ky projekt integron tre qasje të ndryshme në një sistem të vetëm analitik. Kjo e bën zgjidhjen më gjithëpërfshirëse dhe fleksibile, duke demonstruar aplikim praktik dhe të koordinuar të NoSQL, Big Data dhe Network Analysis për nxjerrjen e njohurive të dobishme nga të dhënat.

1. Hyrje

Në epokën e transformimit digjital, organizatat dhe individët gjenerojnë dhe konsumojnë sasi të mëdha të dhënash në mënyrë të vazhdueshme. Përpunimi efikas i këtyre të dhënave për nxjerrjen e njohurive të dobishme përbën një nga sfidat më të rëndësishme në fushën e teknologjisë së informacionit. Tema e këtij punimi është interesante dhe e rëndësishme për faktin se adreson një qasje të integruar mbi analizën e të dhënave duke kombinuar tre drejtime bashkëkohore: NoSQL, Big Data, dhe Network Analysis. Motivimi për këtë projekt buron nga rëndësia që ka aftësia për të përpunuar të dhëna heterogjene në formate dhe struktura të ndryshme, si dhe nevoja për të analizuar marrëdhëniet mes entiteteve në mënyrë të thelluar. Puna është pjesë e një konteksti më të gjerë akademik dhe praktikohet për të ilustruar aplikimin e mjeteve të avancuara analitike si MongoDB, Apache Spark, dhe ORA Lite. Sfidat që adreson ky punim është menaxhimi i përpunimit të të dhënave të mëdha të shpërndara dhe të ndërlikuara — për shembull, integrimi i skedarëve JSON të strukturës së Mondial me query komplekse në MongoDB dhe SparkSQL, apo identifikimi i nyjeve me rëndësi të lartë në një rrjet social me ndihmën e metrikave të teorisë së grafeve. Risia që sjell kjo temë është kombinimi i metodave dhe platformave të ndryshme në një proces të vetëm të unifikuar analitik. Kontributi që sjell ky projekt qëndron në krijimin e një strukture të plotë për trajtimin e të dhënave që variojnë nga struktura klasike tabelare te ato gjysmë-strukturuara dhe deri te të dhënat lidhore (grafë), duke demonstruar fleksibilitet dhe qartësi në zgjidhjen e problemit. Problemi kryesor që trajtohet është nxjerrja e njohurive të vlefshme nga të dhëna të mëdha dhe komplekse, përmes përzgjedhjes së teknologjive të përshtatshme për çdo lloj strukture të të dhënave. Zgjidhja është ndarë në tre faza: migrimi dhe analizimi i të dhënave me MongoDB, përpunimi i dataset-eve të mëdha me Apache Spark, dhe analizimi i strukturës së marrëdhënieve me ORA Lite.

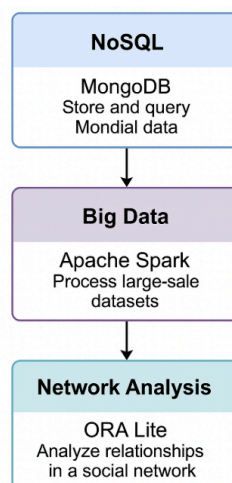


Figura 1: Pamje e përgjithshme e rrjedhës së projektit.

2. Qëllimi i punimit

Qëllimi i këtij projekti është të demonstrojë përdorimin e teknologjive të avancuara për analizën e të dhënave të mëdha dhe të ndërlidhura. Janë përzgjedhur MongoDB, Apache Spark dhe ORA Lite për shkak të fleksibilitetit dhe efikasitetit që ofrojnë përkatësisht në menaxhimin e të dhënave gjysmë-strukturuara, përpunimin e të dhënave të mëdha dhe analizën e rrjeteve. Këto vegla janë të lehta për përdorim, mirë të dokumentuara dhe të përshtatshme për natyrën e dataset-eve të përdorura në projekt.

Informacionet janë mbledhur përmes dokumentacionit zyrtar dhe testimeve praktike në Colab dhe Studio 3T. Secila vegël është përdorur në mënyrë të dedikuar për pjesën përkatëse të analizës, duke siguruar zgjidhje të optimizuara për çdo lloj të dhëne.

3. Pjesa kryesore

3.1. NoSQL Data

3.1.1. Instalimi dhe konfigurimi i MongoDB

1. Instalimi i Studio 3T

- Studio 3T u përdor si mjet GUI për menaxhimin dhe shkrimin e query-ve në MongoDB.
- U instalua versioni më i fundit nga faqja zyrtare <https://studio3t.com>.
- U zgjodh për shkak të funksionaliteteve të avancuara për shkrimin dhe testimin e query-ve, si dhe për integrimin e lehtë me MongoDB Atlas.

2. Instalimi i MongoDB dhe MongoDB Compass

- U instalua MongoDB Community Edition nga faqja zyrtare <https://www.mongodb.com/try/download/community>.
- Gjithashtu u instalua edhe MongoDB Compass, një tjetër GUI që ofron vizualizim të të dhënave dhe testim të shpejtë të query-ve.
- MongoDB Compass u përdor për inspektim të strukturës së koleksioneve dhe vlerave të dokumenteve.

3. Krijimi i një Llogarie në MongoDB Atlas

- U krijua një llogari falas në <https://cloud.mongodb.com>.
- Brenda Atlas, u krijua një cluster "Free Tier", i cili mundëson hostimin e të dhënave në cloud.
- Kjo na lejoi të ruajmë të dhënat në një ambient cloud për akses të lehtë dhe fleksibil gjatë testimeve dhe zhvillimit.

4. Marrja e Connection String dhe Lidhja me Mjetet GUI

- Nga MongoDB Atlas u gjenerua connection string për cluster-in tonë.
- Kjo string u përdor për të lidhur si Studio 3T, ashtu edhe MongoDB Compass me cluster-in e krijuar në cloud.
- Lidhja me sukses e mjediseve GUI na mundësoi të ndërveprojmë me të dhënat në mënyrë efikase.

3.1.2. Migrimi i të dhënave dhe skemës

Për të realizuar migrimin nga baza relacionale (MySQL) në një bazë të dhënash NoSQL (MongoDB), janë ndjekur hapat e mëposhtëm:

1. Eksportimi i të Dhënave nga MySQL

- Tabelat nga baza e të dhënave relacionale "Mondial" në MySQL Workbench janë eksportuar në formatin JSON.
- Çdo tabelë është ruajtur si një skedar i veçantë .json, duke ruajtur strukturën e të dhënave përkatëse.

2. Krijimi i bazës së të dhënave në MongoDB

- Në Studio 3T është krijuar një bazë e re e të dhënave me emrin "mondial".

3. Importimi i JSON Files

- Çdo skedar JSON është importuarsi collection përkatës në MongoDB përmes Studio 3T duke përdorur opsionin "Import JSON File".
- Pas importimit, është verifikuar që të dhënat janë pasqyruar saktë dhe përputhen me ato të MySQL.

4. Sinkronizimi me MongoDB Compass

- Meqë MongoDB është i hostuar në MongoDB Atlas, të dhënat e importuara në Studio 3T janë automatikisht të aksesueshme edhe në MongoDB Compass, ku u verifikua struktura dhe përmbajtja e dokumenteve. Në figurën e mëposhtme është paraqitur edhe sinkronizimi i saktë në mes të studio 3t dhe mongodb compass.

The screenshot shows the MongoDB Compass interface. On the left, a sidebar lists the database 'cluster0.msgepp.mongodb.net' and its collections: 'admin', 'config', 'dbil', 'local', and 'mondial'. The 'mondial' collection is expanded, showing sub-collections: 'borders', 'city', 'continent', 'country', 'desert', 'economy', 'encompasses', 'ethnicgroup', 'geo_desert', 'geo_estuary', 'geo_island', 'geo_lake', 'geo_mountain', 'geo_river', 'geo_sea', 'geo_source', and 'island'. The main panel displays a table of these collections with their statistics.

Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
borders	4.10 kB	320	72.00 B	1	4.10 kB
city	188.42 kB	3.1 K	134.00 B	1	106.50 kB
continent	20.48 kB	5	55.00 B	1	20.48 kB
country	32.77 kB	238	131.00 B	1	24.58 kB
desert	20.48 kB	63	94.00 B	1	20.48 kB
economy					

Figura 2: Databaza, collections dhe dokumentet json të krijuara automatikisht edhe në MongoDB Compass.

3.1.3. Shkrimi i query-ve në NoSQL

Query 1: Listo lumenjtë të cilët rrjedhin nëpër shtete të cilat nuk janë anëtarë të NATO-s, kanë dalje në dete si dhe numri i lumenjëve në këto shtete është më i madh se 10.

Ky query MongoDB përdor një pipeline agregimi për të listuar lumenjtë që rrjedhin nëpër shtete që nuk janë anëtare të NATO-s, kanë dalje në det dhe kanë më shumë se 10 lumenj. Fillimisht bëhet një bashkim (\$lookup) me koleksionin geo_sea për të verifikuar se shteti ka dalje në det, duke përjashtuar ato që nuk kanë. Më pas, përmes një bashkimi tjetër me koleksionin ismember, kontrollohet nëse shteti nuk është anëtar i NATO-s, duke përfshirë vetëm ato që nuk kanë asnjë regjistrim për anëtarësim në këtë organizatë. Më tej, lumenjtë grupohen sipas shtetit dhe numërohen. Me anë të një filtri (\$match), përzgjidhen vetëm ato shtete ku numri i lumenjve është më i madh se 10. Në fund, rezultatet zgjerohen (\$unwind), dhe përmes një renditjeje (\$sort), shfaqet lista përfundimtare e lumenjve të këtyre shteteve. Ky query nxjerr në pah ato shtete jo-anëtare të NATO-s që kanë qasje në det dhe një rrjet të dendur të lumenjve. Kodi i këtij query gjendet në Aneks A, kurse rezultati që shfaq ky query është paraqitur në figurën e mëposhtme:

country	river
R	Kolyma
R	Lena
R	Narva
R	Newa
R	Northern Dwina
R	Ob
R	Oka
R	Paatsjoki
R	Petschora
R	Schilka
R	Suchona
R	Swir
R	Tobol
R	Ural
R	Volga
R	Vuoksi
R	Western Dwina
ZRE	Aruwimi

Figura 3: Rezultati i shfaqur nga query i mësipërm.

Query 2: Të listohen të gjitha detet në të cilat nuk ka asnjë ishull mirëpo shtetet të cilat kufizohet ai det janë anëtare në NATO ose BE.

Ky query MongoDB përdor një pipeline agregimi për të listuar të gjitha detet që nuk kanë asnjë ishull dhe kufizohen vetëm nga shtete anëtare të NATO-s ose Bashkimit Evropian. Fillimisht bëhet një bashkim (\$lookup) me koleksionin islandin për të filtruar detet që nuk kanë ishuj, duke përjashtuar ato që kanë. Më pas, përmes një bashkimi tjetër me koleksionin geo_sea, identifikohen shtetet që kufizohen me secilin det. Për secilin prej këtyre shteteve, bëhet një bashkim me koleksionet country dhe ismember për të verifikuar nëse janë anëtarë të organizatave NATO ose BE. Në fund, përmes një grupimi (\$group) dhe filtrimi (\$match), për zgjidhen vetëm detet ku të gjitha shtetet kufitare janë anëtare të këtyre organizatave. Rezultati është lista e deteve që plotësojnë këto kushte. Kodi i këtij query gjendet në Aneks A, kurse rezultati që shfaq ky query është paraqitur në figurën e mëposhtme:

sea
Sea_Name
Skagerrak

Figura 4: Rezultati i shfaqur nga query i mësipërm.

Query 3: Të listohen të gjitha shtetet e EU-së të cilat kanë dalje në të njëjtin det dhe liqen.

Ky query në MongoDB përdor aggregation pipeline për të gjetur çiftet e vendeve anëtare të NATO-s që ndajnë të njëjtin det (Sea) dhe të njëjtin liqen (Lake), pa përsëritur të njëjtat kombinime. Fillimisht filtrohen vendet anëtare të NATO-s nga koleksioni ismember, pastaj për secilin vend kërkohet deti dhe liqeni përkatës përmes \$lookup nga koleksionet geo_sea dhe geo_lake. Pas kësaj, përmes bashkimeve të tjera (\$lookup dhe \$unwind), identifikohen vendet e tjera që ndajnë të njëjtin det dhe liqen dhe që janë gjithashtu pjesë e NATO-s. Përdoret një kontroll me \$lt për të shmangur përsëritjen e kombinimeve në mënyrë simetrike (p.sh. të shfaqet vetëm një herë CDN-USA dhe jo edhe USA-CDN). Në fund, përdorimi i \$group siguron që secila dyshe vendesh me të njëjtin det dhe liqen të shfaqet vetëm një herë në rezultat, duke eliminuar duplikimet. Kodi i këtij query gjendet në Aneks A, kurse rezultati që e shfaq ky query është paraqitur në figurën e mëposhtme:





























































Country1	Country2	Sea	Lake
 CDN	 USA	 Arctic Ocean	 Lake Ontario
 CDN	 USA	 Arctic Ocean	 Lake Huron
 CDN	 USA	 Atlantic Ocean	 Lake Champlain
 CDN	 USA	 Arctic Ocean	 Lake Superior
 CDN	 USA	 Pacific Ocean	 Lake Erie
 CDN	 USA	 Pacific Ocean	 Lake Champlain
 CDN	 USA	 Atlantic Ocean	 Lake Erie
 CDN	 USA	 Pacific Ocean	 Lake Superior
 CDN	 USA	 Arctic Ocean	 Lake Champlain
 CDN	 USA	 Arctic Ocean	 Lake Erie
 CDN	 USA	 Atlantic Ocean	 Lake Ontario
 CDN	 USA	 Pacific Ocean	 Lake Huron
 CDN	 USA	 Atlantic Ocean	 Lake Superior
 CDN	 USA	 Pacific Ocean	 Lake Ontario
 CDN	 USA	 Atlantic Ocean	 Lake Huron

Figura 5: Rezultati i shfaqur nga query i mësipërm.

3.2. Big Data

3.2.1. Instalimi dhe konfigurimi i Apache Spark

Për përpunimin e të dhënave në këtë projekt është përdorur Apache Spark si platformë për Big Data. Mjedisi është konfiguruar përmes Google Colab, i cili ofron një ambient të përshtatshëm për punë me PySpark pa nevojën e instalimeve lokale. Në këtë link mund t'i qaseni punimit në Google Colab:

<https://colab.research.google.com/drive/16i7hc4VRAQ3bx9bbrJsH7qs9dpKg8YNN#scrollTo=jUrvDnMwHoTs>.

Hapat e ndjekur:

- U instalua Java 8, Spark 3.1.2 dhe Hadoop 2.7 përmes komandave në Colab
- U vendosën variablat e mjedisit JAVA_HOME dhe SPARK_HOME
- U krijua një SparkSession për të lejuar manipulimin e të dhënave me SparkSQL

Kjo qasje shmang kompleksitetin e konfigurimeve lokale dhe siguron ekzekutim të qëndrueshëm në cloud. Kodi për këtë pjesë është paraqitur në Aneksin A.

3.2.2. Importimi i dataset-it Mondial

Dataset-i Mondial është importuar në mjedisin Google Colab nëpërmjet ngarkimit të fileve në formatin JSON. Pas ngarkimit, të dhënat janë lexuar duke përdorur funksionin `spark.read.json()` dhe janë regjistruar si collections në SparkSQL përmes komandës `createOrReplaceTempView()`. Ky proces ka mundësuar trajtimin e të dhënave si struktura tabelare dhe përdorimin e gjuhës SQL për analizën e tyre në mënyrë të centralizuar dhe të standardizuar. Kodi që mundëson importimin e Mondial-it është paraqitur në Aneksin A. Në figurën e mëposhtme është paraqitur importimi i suksesshëm i datasetit të Mondialit.

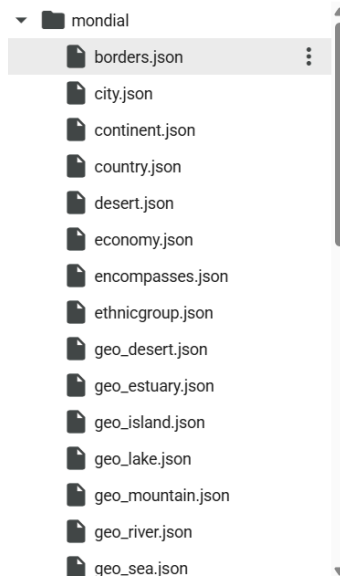


Figura 6: Importimi i datasetit të Mondialit në Google Colab.

3.2.3. Ekzekutimi i query-ve

Query 1: Të listohen të gjitha kryeqytetet në të cilat nuk kalon asnjë lum, mirëpo janë anëtarë në të paktën një organizatë.

Ky query identifikon kryeqytetet që nuk kalohen nga asnjë lumë, por janë anëtare të të paktën një organizate. Fillimisht, përmes një bashkimi RIGHT JOIN midis city dhe country, filtrohen të gjitha kryeqytetet. Më pas, përdoret EXISTS për të përzgjedhur vetëm ato që ndodhen në tabelën organization, dhe në fund aplikohet NOT EXISTS për të përjashtuar qytetet që figurojnë në tabelën located, duke eliminuar ato që kanë lidhje me elementë si lumenjtë. Struktura me WITH siguron qartësi dhe modularitet në ndërtimin e pyetjes. Kodi i këtij query gjendet në Aneksin A, kurse në figurën e mëposhte është paraqitur rezultati që shfaqë.

```

+-----+
| Kryeqytetet_0|
+-----+
| Gaborone|
| Lima|
| Madrid|
| Prague|
| Tegucigalpa|
| Amman|
| Caracas|
| Minsk|
| Pretoria|
| Jerusalem|
| Mexico City|
| Washington|
| Riyadh|
| Kathmandu|
| Brussels|
| Damascus|
| Guatemala City|
| Bangkok|
| Luxembourg|
| Tehran|
+-----+
only showing top 20 rows

```

Figura 7: Rezultati i shfaqur pas ekzekutimit të query-t të mësipërm.

Query 2: Të listohen malet më të larta të cilat shtrihen në vendet e Ballkanit përjashtuar Serbinë.

Ky query përzgjedh 5 malet më të larta që ndodhen në vendet e Ballkanit. Fillimisht realizohet një JOIN midis tabelave mountain dhe geo_mountain për të lidhur të dhënat mbi emrin e malit. Më pas filtrohen rreshtat që i përkasin vetëm vendeve të përzgjedhura përmes kushteve në WHERE. Të dhënat grupohen sipas emrit të malit dhe llogaritet lartësia maksimale (MAX) për secilin mal. Në fund, rezultatet renditen në mënyrë zbritëse sipas lartësisë dhe përzgjidhen pesë malet më të larta. Kodi i këtij query është paraqitur në Aneksin A, kurse rezultati që shfaqë është paraqitur në figurën e më poshtme.

```

+-----+-----+
| Mountain_Name|Height|
+-----+-----+
| Musala|2925.0|
| Olymp|2917.0|
| Korab|2751.0|
| Jezerce|2694.0|
| Moldoveanu|2544.0|
+-----+-----+

```

Figura 8: Rezultati i shfaqur nga ekzekutimi i query-t të mësipërm.

3.2.4. Importimi i një dataset-i të ri

Për të plotësuar kërkesën për analizë të të dhënave në shkallë të gjerë, është importuar një dataset i dytë me të dhëna mbi përhapjen e COVID-19 në Shtetet e Bashkuara, në nivel të qarqeve. Dataset-i është në format .csv, me madhësi mbi 50MB dhe mbi 500,000 rreshta, duke përmbushur kriteret e projektit për volum të të dhënave.

File është ngarkuar në mjedisin Google Colab dhe është lexuar me funksionin `spark.read.csv(...)`, duke aktivizuar opsionet `header=True` dhe `inferSchema=True`. Pas leximit, të dhënat janë regjistruar si tabelë e përkohshme me emrin `covid` për përdorim në SparkSQL. Kodi për importimin e këtij dataseti është paraqitur në Aneksin A.

Query 1: Listo shtetet me më së shumti raste të konfirmuara.

Ky query realizon një agregim të të dhënave nga tabela `covid`, duke llogaritur numrin total të rasteve të konfirmuara (`SUM(Confirmed)`) për secilin shtet (`Province_State`). Përdoret grupimi me `GROUP BY` për të përmbledhur të dhënat në nivel shteti, ndërsa `ORDER BY` në mënyrë zbritëse mundëson identifikimin e shteteve më të prekura. Me `LIMIT 10` kufizohet rezultati në dhjetë shtetet me numrin më të lartë të rasteve të konfirmuara, duke mundësuar analizën krahasuese të përhapjes së pandemisë në nivel kombëtar. Kodi i këtij query është paraqitur në Aneksin A, kurse rezultati që shfaqë është paraqitur në figurën e mëposhtme.

State	Total_Confirmed
New York	39808447
California	17618695
New Jersey	16506714
Texas	12698726
Florida	12657802
Illinois	11900637
Massachusetts	9874030
Pennsylvania	8096993
Georgia	6859759
Michigan	6690544

Figura 9: Rezultati i shfaqur nga ekzekutimi i query-t të mësipërm.

Query 2: Listo shtetet të cilat kanë përqindjen më të madhe të rasteve të vdekjeve.

Ky query realizon analizën e shkallës së vdekshmërisë për çdo shtet në tabelën covid, duke llogaritur numrin total të vdekjeve dhe rasteve të konfirmuara përmes funksionit SUM(). Më pas, përmes një shprehjeje aritmetike, llogaritet përqindja e vdekjeve ndaj rasteve të konfirmuara (Death_Rate_Percent) dhe rezultati rrumbullakohet në dy shifra dhjetore me funksionin ROUND(). Kushti WHERE Confirmed > 0 siguron shmangien e ndarjes me zero, ndërsa ORDER BY dhe LIMIT përdoren për të identifikuar 10 shtetet me përqindjen më të lartë të vdekjeve. Kodi i këtij query është paraqitur në Aneksin A, kurse rezultati që shfaqë është paraqitur në figurën e mëposhtme.

State	Total_Deaths	Total_Confirmed	Death_Rate_Percent
Connecticut	374345	4239220	8.83
Michigan	576004	6690544	8.61
Northern Mariana ...	230	2741	8.39
New York	3175935	39808447	7.98
New Jersey	1221257	16506714	7.40
Massachusetts	666140	9874030	6.75
Pennsylvania	537249	8096993	6.64
New Hampshire	25377	471598	5.38
Louisiana	285081	5383429	5.30
Indiana	200183	3792618	5.28

Figura 10: Rezultati i shfaqur nga ekzekutimi i query-t të mësipërm.

Query 3: Listo shtetet të cilat kanë përqindjen më të lartë të vdekjeve në raport me rastet e konfirmuara me COVID-19.

Ky query identifikon 10 qarqet me numrin më të lartë të rasteve të konfirmuara me COVID-19 në SHBA. Duke përdorur funksionet agreguese SUM(Confirmed) dhe SUM(Deaths), llogaritet numri total i rasteve dhe vdekjeve për secilin kombinim të Admin2 (qark) dhe Province_State (shtet). Të dhënat grupohen me GROUP BY dhe më pas renditen në mënyrë zbritëse sipas rasteve të konfirmuara përmes ORDER BY Confirmed DESC. Përmes LIMIT 10, shfaqen vetëm 10 qarqet më të prekura nga pandemia. Kodi i këtij query është paraqitur në Aneksin A, kurse rezultati që shfaqë është paraqitur në figurën e mëposhtme.

County	State	Confirmed	Deaths
New York	New York	21813290	2355309
Los Angeles	California	7640428	262977
Cook	Illinois	7616290	360612
Nassau	New York	4391725	215084
Suffolk	New York	4248752	183141
Westchester	New York	3665043	136980
Miami-Dade	Florida	3350178	79217
Maricopa	Arizona	3259255	62901
Philadelphia	Pennsylvania	2326353	128873
Harris	Texas	2319014	29913

Figura 11: Rezultati i shfaqur nga ekzekutimi i query-t të mësipërm.

Vizualizim mbështetës: Korelacioni midis rasteve të konfirmuara dhe vdekjeve sipas shtetit

Ky vizualizim paraqet një **grafik shpërndarjeje (scatter plot)** që tregon marrëdhënien ndërmjet numrit të rasteve të konfirmuara me COVID-19 dhe numrit të vdekjeve në çdo shtet të SHBA-së. Çdo pikë në grafik përfaqëson një shtet, me pozicionin horizontal që tregon totalin e rasteve të konfirmuara dhe pozicionin vertikal që tregon totalin e vdekjeve. Grafiku shërben për të analizuar vizualisht korelacionin ndërmjet dy variablave kryesorë të pandemisë dhe për të identifikuar shtetet që dallohen për numër të lartë të rasteve ose të vdekjeve. Edhe pse nuk është pjesë e detyrueshme e projektit, ky vizualizim e pasuron analizën dhe ndihmon në interpretimin më intuitiv të të dhënave.

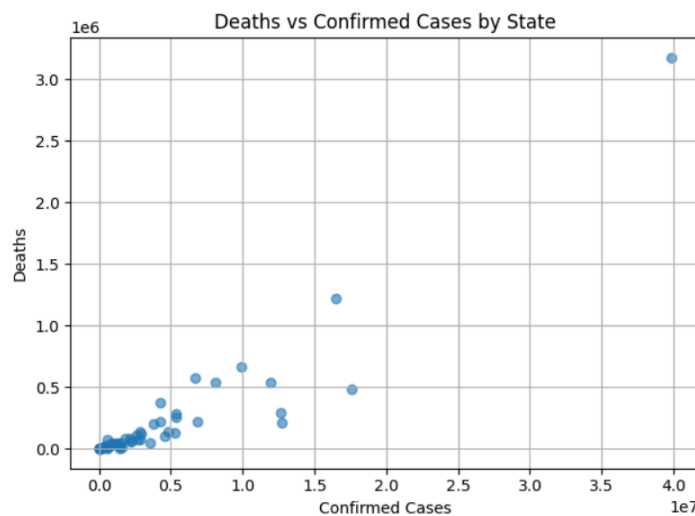


Figura 12: Vizualizimi i korrelacionit në mes të rasteve të konfirmuara dhe vdekjeve.

Vizualizim mbështetës: Shtetet me numrin më të lartë të rasteve të konfirmuara

Për të përforcuar analizën e të dhënave, është realizuar një vizualizim grafik (bar chart) që paraqet 10 shtetet me më shumë raste të konfirmuara me COVID-19. Grafiku është ndërtuar duke përdorur pandas dhe matplotlib, bazuar në të dhënat e përftuara nga një query në SparkSQL. Edhe pse ky vizualizim nuk ishte pjesë e kërkesave të projektit, ai shërben si ilustrim vizual për të lehtësuar interpretimin e rezultateve dhe për të nxjerrë përfundime më të qarta nga analiza numerike. Kodi për këtë vizualizim është paraqitur në Aneksin A, kurse në figurën e mëposhtme është paraqitur vizualizimi.

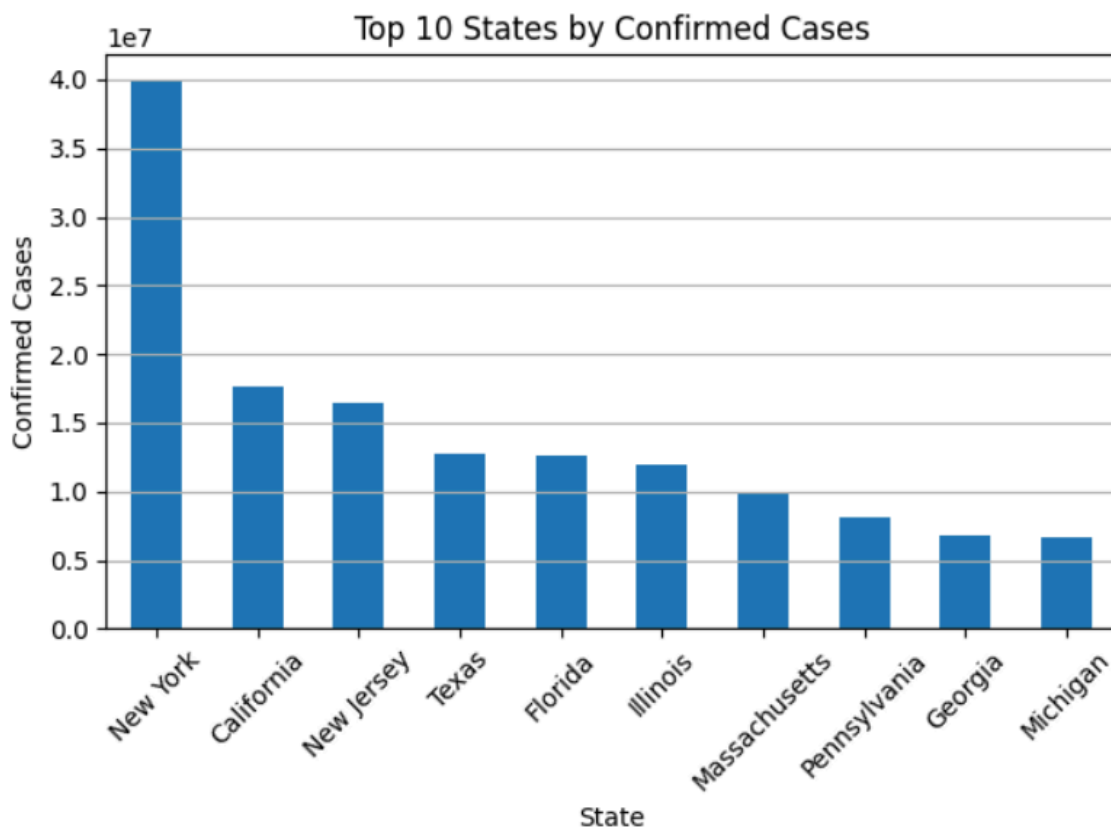


Figura 13: Vizualizimi i shteteve që kanë më së shumti raste të konfirmuara të COVID.

Për të siguruar që dataset-et të jenë të qasshme edhe pas përfundimit të sesioneve në Google Colab, u realizua ruajtja e tyre në Google Drive. Konkretisht, u krijua një folder me emrin `big_data_project` brenda My Drive, ku u vendosën: `usa_county_wise.csv` (dataset me mbi 500,000 rreshta) dhe `mondial/` (folder me skedarë JSON që përfaqësojnë strukturën e bazës së të dhënave Mondial). Në çdo ekzekutim të ri të notebook-ut, Google Drive bëhet mount me komandën e paraqitur në Aneksin A dhe më pas dataset-et ngarkohen direkt nga komandat e paraqitura poashtu në Aneksin A.

3.3. Network analysis (Teoria e grafeve)

3.3.1. Instalimi dhe konfigurimi i ORA Lite

Për realizimin e network analysis është përdorur platforma ORA-Lite, e zhvilluar nga Qendra CASOS në Carnegie Mellon University. ORA është një mjet i fuqishëm për analizë të rrjeteve komplekse dhe lejon importimin e të dhënave, vizualizimin e grafit dhe llogaritjen e metrikave të ndryshme.

Hapat e ndjekur për instalim dhe konfigurim:

1. Shkarkimi i programit ORA-Lite nga faqja zyrtare e Netanomics/CASOS:
2. Zgjedhja e versionit për Windows dhe instalimi i aplikacionit duke ndjekur hapat standard të instalimit.
3. Hapja e aplikacionit ORA dhe testimi i funksionalitetit bazë.

3.3.2. Importimi i të dhënave

Pasi u përgatit mjedisi në ORA, hapi i parë i analizës ishte importimi i të dhënave të rrjetit. Dataset-i i përdorur përmbante lidhje midis përdoruesve (agjentëve) të rrjetit social Facebook, ku çdo rresht përfaqësonte një lidhje (marrëdhënie) midis dy individëve.

Hapat e ndjekur:

1. Përgatitja e dataset-it në format .xlsx, me dy kolona: Source dhe Target, të cilat përfaqësonin nyjen burimore dhe nyjen target të lidhjes (p.sh., përdoruesi A është i lidhur me përdoruesin B).
2. Hapja e ORA dhe navigimi tek File dhe Data Import Wizard.
3. Zgjedhja e opsionit: Import Excel or text delimited files dhe Table of network links.
4. Ngarkimi i skedarit xlsx dhe përcaktimi i kolonës FROM si Node IDs dhe kolonës TO po ashtu si Node IDs.
5. Zgjedhja e klasës së nyjeve (Nodeset Class) si Agent, pasi të gjithë nyjet përfaqësonin individë të rrjetit.
6. Konfirmimi i rrjetit dhe klikimi "Finish", çka rezultoi në krijimin automatik të një Meta Network të quajtur "Agent x Agent".

3.3.3. Gjenerimi i skemës vizuale

Pas importimit të të dhënave në ORA, u realizua vizualizimi i rrjetit social përmes gjenerimit të një skeme grafike ku çdo nyje përfaqësonte një përdorues dhe çdo lidhje mes tyre tregonte ndërveprim ose marrëdhënie. Brenda platformës u përdor opsioni "Visualize Network" për të krijuar një paraqitje vizuale të rrjetit, duke aplikuar automatikisht një shpërndarje të nyjeve në hapësirë për ta bërë grafikun më të lexueshëm. U krijuan dy versione të vizualizimit: një i pa

grupuar, ku të gjitha nyjet janë të shpërndara në mënyrë të barabartë, dhe një tjetër i grupuar, ku nyjet ndahen në komunitete të ndryshme në bazë të lidhjeve të tyre të brendshme.

ORA Lite e ka mundësinë të vizualizojë duke grupuar kështu automatikisht në bazë të lidhjeve në mes të individëve, krijon grupe të brendshme ku nyjet (shokët) kanë më shumë ndërveprime me njëri-tjetrin sesa me pjesën tjetër të rrjetit. Kjo është e dobishme për rrjete sociale si Facebook, ku grumbujt e mund të përfaqësojnë qarqe shoqërore të veçanta si familja, shkolla ose puna, përmasat dhe ngjyrat e nyjeve ndihmojnë në dallimin vizual të këtyre komuniteteve.

Këto skema vizuale ndihmuan në identifikimin e përdoruesve më të rëndësishëm dhe strukturës së përgjithshme të rrjetit. Në figurat e mëposhtme paraqiten këto dy vizualizime të përmendura:

Meta Network-modified

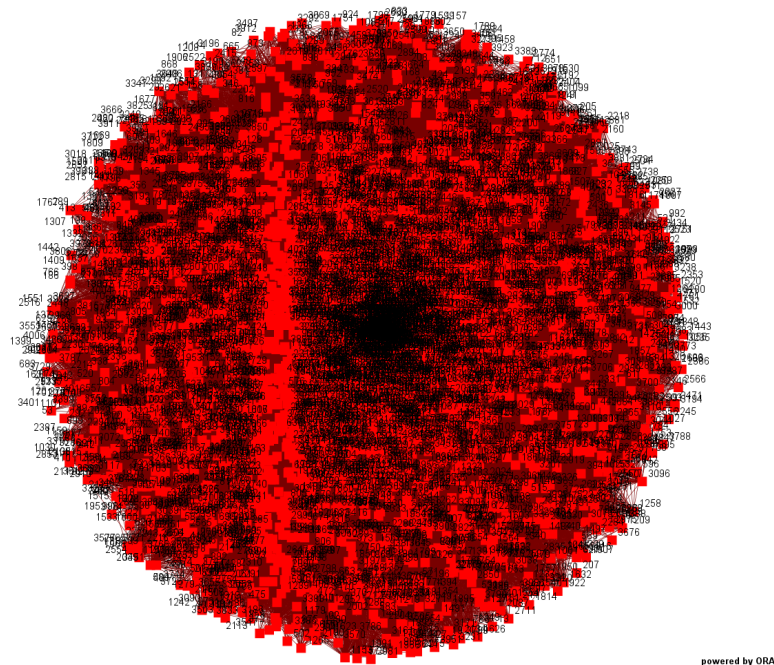


Figura 14: Vizualizimi i pa grupuar.

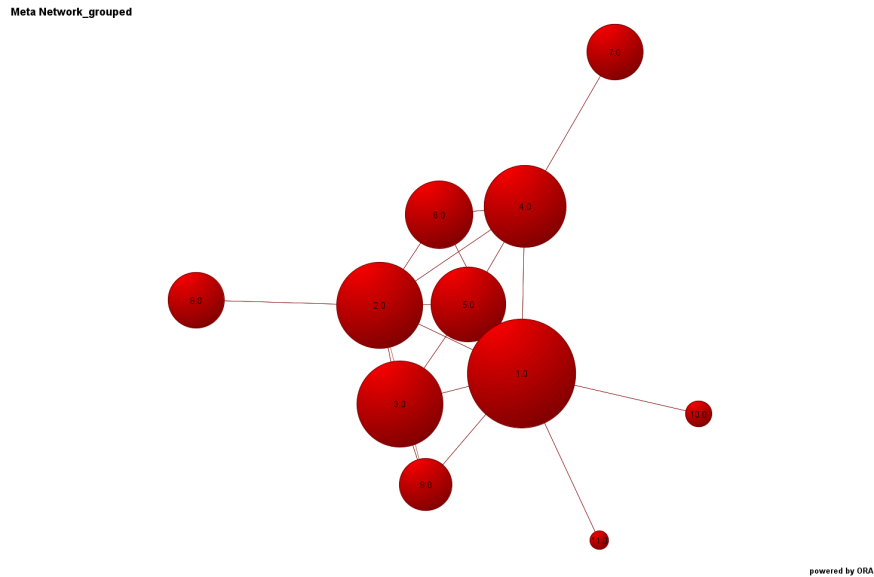


Figura 15: Vizualizimi i grupuar.

3.3.4. Kalkulimi i metrikave të rrjetit

Pasi u importuan të dhënat dhe u gjenerua rrjeti në platformën ORA, u vazhdua me llogaritjen e metrikave kyçe të rrjetit për të analizuar rëndësinë dhe pozicionin e secilës nyje brenda strukturës së përgjithshme. Metrikat e llogaritura përfshijnë:

Degree Centrality tregon numrin e lidhjeve direkte që ka një nyje me nyje të tjera në rrjet, dhe përdoret për të identifikuar aktorët më aktivë. Formula:

$$CD(v) = \deg(v)$$

Në figurën e mëposhtme është paraqitur rrjeti i agentëve të grupuar të analizuar sipas Degree Centrality. Madhësia e nyjeve është proporcionale me vlerën e degree, nyjet më të lidhura janë më të mëdha vizualisht, ngjyra e nyjeve paraqet poashtu shkallën e lidhshmërisë, nyjet me më shumë lidhje janë me ngjyra të kuqe ose portokalli, kurse ato me më pak lidhje paraqiten me nuanca të ftohta si të kaltërt ose të gjelbërt.

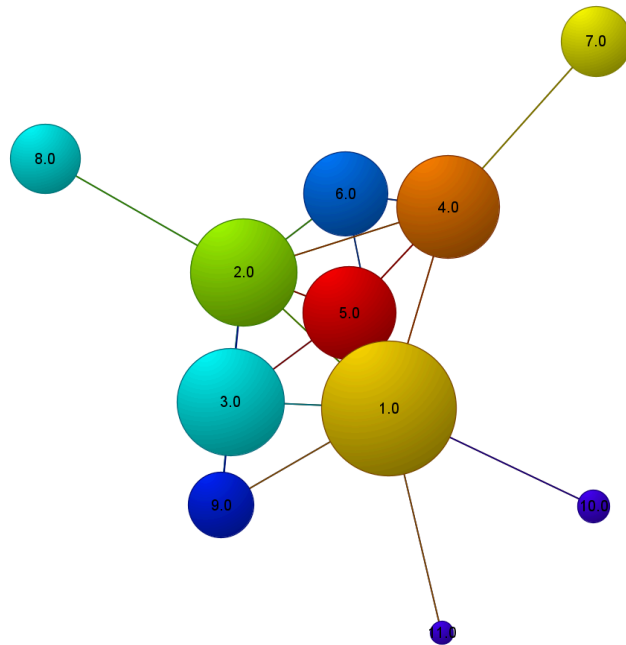


Figura 16: Vizualizimi i nyjeve të grupuara në bazë të Degree Centrality.

Tabela e mëposhtme paraqet 10 nyjet me më shumë lidhje në rrjetin social të analizuar, bazuar në metrikën Total Degree Centrality. Kolona "Value" tregon vlerën e normalizuar të centralitetit për secilën nyje, ndërsa "Unscaled" përfaqëson numrin real të lidhjeve të secilës nyje. Nyja me ID 107 ka vlerën më të lartë të degree centrality, duke qenë më e lidhura dhe potencialisht më e rëndësishme për shpërndarjen e informacionit në rrjet.

Rank	Agent	Value	Unscaled
1	107	0.129	1,045
2	1684	0.098	792
3	1912	0.093	755
4	3437	0.068	547
5	0	0.043	347
6	2543	0.036	294
7	2347	0.036	291
8	1888	0.031	254
9	1800	0.030	245
10	1663	0.029	235

Showing 1 to 10 of 100 entries

Previous 1 2 3 4 5 ... 10 Next

Figura 17: Renditja e agjentëve sipas Total Degree Centrality.

Closeness Centrality mat se sa afër është një nyje me të gjitha nyjet e tjera, duke vlerësuar sa shpejt mund të shpërndahet informacioni prej saj. Formula:

$$CC(v) = (n - 1) / \sum d(u, v)$$

Në figurën e mëposhtme është paraqitur rrjeti i agentëve të grupuar sipas metrikës Closeness Centrality. Në këtë vizualizim madhësia e nyjeve është proporcionale me nivelin e closeness dhe nyjet me vlerë të lartë janë ngjyrosur më intensivisht në këtë rast me ngjyrë të kuqe, kurse nyjet me vlera më të ulëta janë ngjyrosur me ngjyra të gjelbërta ose të kaltërta. Nga vizualizimi shihet se nyjet me etiketat 1.0, 2.0 dhe 5.0 janë më të mëdha dhe me ngjyrë të kuqe duke treguar se ato kanë pozitë të favorshme për të shpërndarë apo marrë informacion shpejt nga nyjet e tjera, kjo i bën ato nyje qendrore për komunikim në rrjet.

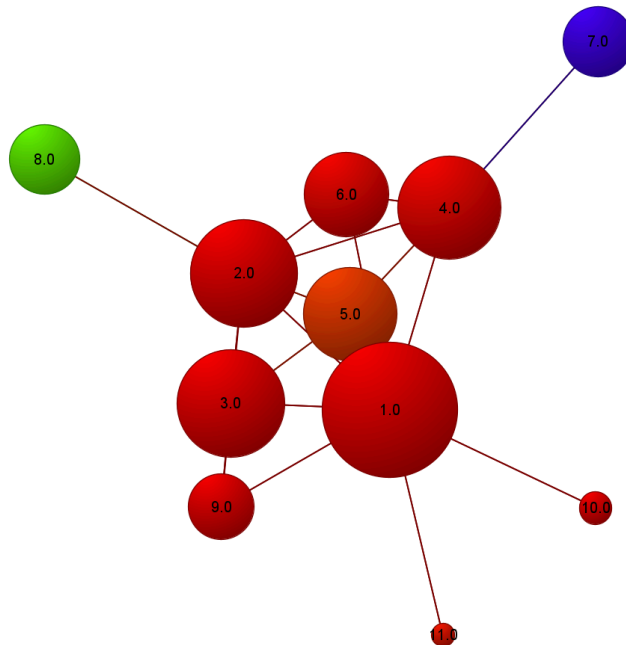


Figura 18: Vizualizimi i nyjeve të grupuara në bazë të Closeness Centrality.

Tabela e mëposhtme paraqet rezultatet e metrikës Closeness Centrality për agentët në rrjet. Closeness Centrality mat se sa afër ndodhet një nyje me të gjitha nyjet e tjera në rrjet, duke llogaritur distancën mesatare nga një nyje drejt të tjerave. Në figurën 8, janë paraqitur 10 agentët me vlerat më të larta të kësaj metrike, çka tregon se këta agentë janë në pozita që mund të shpërndajnë informacionin me efikasitet në rrjet. Vlerat e larta në kolonën *Value* tregojnë centralitet të lartë, ndërsa kolona *Unscaled* pasqyron vlerat e papërpunuara të centralitetit për secilin agent.

Rank	Agent	Value	Unscaled
1	0	0.005	0
2	3	0.002	0
3	9	0.002	0
4	13	0.002	0
5	21	0.002	0
6	25	0.002	0
7	30	0.002	0
8	26	0.002	0
9	16	0.002	0
10	29	0.002	0

Showing 1 to 10 of 100 entries

Previous 1 2 3 4 5 ... 10 Next

Figura 19: Renditja e agentëve sipas Closeness Centrality.

Betweenness Centrality tregon se sa shpesh një nyje gjendet në rrugët më të shkurtra ndërmjet dy nyjeve të tjera, dhe identifikon pikat kyçe për shpërndarje informacioni ose kontroll. Formula:

$$CB(v) = \sum [\sigma_{st}(v) / \sigma_{st}], \text{ për çdo } s \neq v \neq t$$

Në figurën e mëposhtme paraqitet rrjeti i agentëve të grupuar të ngjyrosur dhe sipas vlerave të metrikës Betweenness Centrality, në vizualizim nyjet me ngjyrë të kuqe ose të gjelbër kanë vlerë më të lartë, ndërsa ato me ngjyrë të kaltër kanë vlerë më të ulët. Nga ky vizualizim shohim se nyja me etiketën 1.0 ka madhësinë më të madhe dhe është e ngjyrosur me ngjyrën e gjelbërt, kjo na tregon se ajo ndodhet shpesh në rrugët më të shkurtra të rrjetit dhe ka funksion ndërmjetësimi shumë të rëndësishëm.

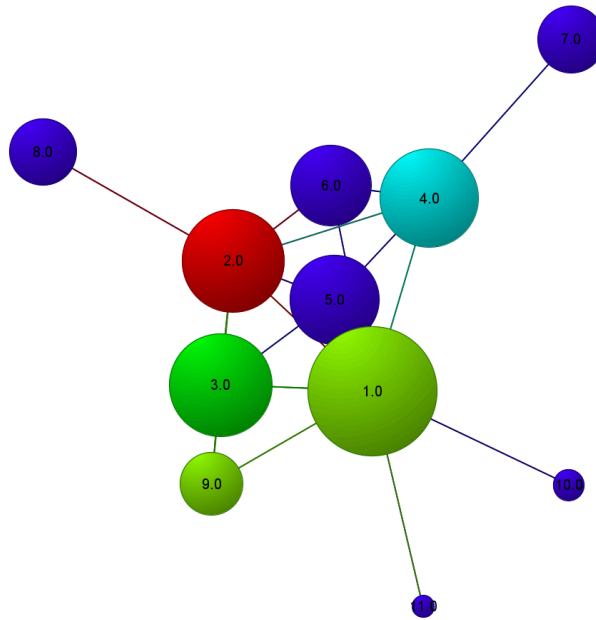


Figura 20: Vizualizimi i nyjeve të grupuara në bazë të Betweenness Centrality.

Tabela e mëposhtme tregon 10 agentët me vlerat më të larta të Betweenness Centrality në rrjet. Këta agentë shërbejnë si ura lidhëse të rëndësishme midis pjesëve të ndryshme të rrjetit, duke ndikuar në rrjedhën e informacionit dhe ndërveprimin mes nyjeve. Kolona “Value” përfaqëson vlerën e normalizuar të kësaj metrike, ndërsa kolona “Unscaled” paraqet vlerën e saktë të llogaritur. Agjenti me ID 1684 ka betweenness më të lartë, që do të thotë se ndodhet më shpesh në rrugët më të shkurtra mes pjesëve të rrjetit.

Rank	Agent	Value	Unscaled
1	1684	0.033	537,943.875
2	1912	0.027	442,512.312
3	1718	0.027	433,252.500
4	563	0.013	212,075.078
5	1405	0.010	165,030.094
6	1656	0.009	153,656.031
7	1086	0.009	139,435.141
8	567	0.008	135,307.812
9	3437	0.008	133,579.438
10	119	0.006	103,660.594

Showing 1 to 10 of 100 entries

Previous 1 2 3 4 5 ... 10 Next

Figura 21: Renditja e agentëve sipas Betweenness Centrality.

Nga krahasimi i metrikave, vërehet se disa nyje që kanë vlerë të lartë në Degree Centrality (p.sh., nyja 107) nuk kanë po ashtu vlerë të lartë në Betweenness Centrality. Kjo tregon se ato

janë të lidhura me shumë të tjerë, por nuk janë domosdoshmërisht ura lidhëse ndërmjet komuniteteve. Në anën tjetër, nyje me betweenness të lartë shërbejnë si kanale kryesore për qarkullimin e informacionit edhe nëse kanë më pak lidhje direkte. Në përfundim, analiza e rrjetit social përmes ORA Lite ka ndihmuar në identifikimin e nyjeve kyçe, ndarjen natyrale të komuniteteve dhe vlerësimin e strukturës së rrjetit. Metrikat si Degree, Closeness dhe Betweenness Centrality ofrojnë perspektiva të ndryshme për rolin e secilit përdorues, ndërsa vizualizimet kanë qenë të domosdoshme për të përçuar rezultatet në mënyrë të qartë dhe intuitive. Këto metrika u llogaritën duke përdorur funksionalitetin analitik të platformës ORA, dhe rezultatet u ruajtën për interpretim të mëtejshëm në përfundim të analizës së rrjetit dhe gjenden në folderin All measures by category në formatin .html dhe .csv.

4. Konkluzione

Në këtë projekt u shfrytëzuan teknologjitë MongoDB, Apache Spark dhe ORA Lite për të realizuar analiza të avancuara mbi të dhëna të mëdha dhe të ndërlidhura. U implementuan me sukses teknika të migrimit nga SQL në NoSQL, ekzekutim queriesh në SparkSQL dhe analizë e rrjeteve sociale me metrika si Degree dhe Betweenness Centrality. U realizuan disa query të qëlluara për dataset-in Mondial dhe atë të COVID-19, si dhe vizualizime të dobishme për interpretim të rezultateve.

Qëllimi i punimit u arrit me sukses, duke demonstruar përdorimin praktik të teknologjive të zgjedhura për trajtimin e problemeve reale në fushën e Big Data dhe analizës së rrjeteve.

Në të ardhmen, projekti mund të zgjerohet me integrimin e burimeve shtesë të të dhënave dhe përdorimin e teknikave të virtualizimit të të dhënave për të analizuar informacionin në kohë reale pa pasur nevojë për migrim fizik.

Referencat

- <https://www.mongodb.com/try/download/community>
- <https://www.mongodb.com/try/download/shell>
- <https://www.mongodb.com/try/download/database-tools>
- <https://www.cmu.edu/casos-center/research/tools/ora-lite.html>
- <https://snap.stanford.edu/data/ego-Facebook.html>
<https://www.youtube.com/playlist?list=PLRTey0Iqj9jh3WTvoupGK0aizk-Rmbz7u>
- <https://www.instaclustr.com/education/apache-spark/quick-guide-to-apache-spark-benefits-use-cases-and-tutorial/>
- <https://www.youtube.com/playlist?list=PL1M5TsfDV6VvsmMnWYWB8BDE7uAMSQQcQ>
- <https://www.youtube.com/playlist?list=PL1M5TsfDV6VvsmMnWYWB8BDE7uAMSQQcQ>
- <https://visiblenetworklabs.com/guides/social-network-analysis-101/>
- <https://www.kaggle.com/datasets/imdevskp/corona-virus-report>