

# 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 të  
Grafeve

Emri profesorit/Asistentit	Emri & mbiemri studentëve / email adresa	
Prof. Dr. Vigan RAÇA MSc. Rafet DURIQI	1. Endrit Kastrati	endrit.kastrati3@student.uni-pr.edu
	2. Erza Gashi	erza.gashi11@student.uni-pr.edu
	3. Jon Kuci	jon.kuci@student.uni-pr.edu
	4. Kaltrina Krasniqi	kaltrina.krasniqi31@student.uni-pr.edu
	5. Mirgeta Gashi	mirgeta.gashi@student.uni-pr.edu

Prishtinë, 2024

## Përmbajtja

<b>Abstrakti</b>	<b>3</b>
<b>I. Hyrje</b>	<b>4</b>
<b>II. Qëllimi i punimit</b>	<b>4</b>
<b>III. Pjesa kryesore</b>	<b>5</b>
3.1. Përdorimi i NoSQL	5
3.1.1. Përgatitja e Mjedisit	6
3.1.2. Migrimi i të Dhënave	6
3.1.3. Query-t në NoSQL	7
3.2. Përpunimi i Big Data	9
3.2.1. Apache Spark në Google Colab	9
3.2.2. Konfigurimi i Mjedisit Spark	9
3.2.2. Importimi i Mondial Dataset	9
3.2.3. Ekzekutimi i Query-ve	10
3.2.4. Importimi i Dataset-it	12
3.2.5. Ekzekutimi i query-ve ne flights dataset	12
3.3. Analiza e Rrjeteve Sociale	14
3.3.1. Përdorimi i Ora Lite Tool	16
3.3.2. Përdorimi i Gephi Tool	22
<b>IV. Konkluzione</b>	<b>25</b>
<b>Referencat</b>	<b>26</b>

## Abstrakti

Ky projekt trajton temën e manipulimit dhe analizës së Big Data duke përdorur teknologjitë më moderne si NoSQL, Apache Spark dhe analiza të rrjeteve sociale (SNA). Puna ndahet në tri pjesë kryesore: përdorimi i MongoDB për ruajtjen fleksibile të të dhënave nga databaza relacionare Mondial; analizimi i dataset-eve të mëdha në Spark përmes SQL-like query-ve; dhe analizimi i rrjeteve sociale përmes mjeteve ORA Lite dhe Gephi mbi datasete. Literatura ekzistuese thekson përparësitë e këtyre veglave në menaxhimin e të dhënave të mëdha dhe vizualizimin e marrëdhënieve komplekse, por integrimi i tyre në një projekt të vetëm e bën këtë qasje më të fuqishme. Zgjidhja jonë ofron efikasitet më të lartë në përpunimin dhe vizualizimin e të dhënave, si dhe nxjerrjen e insight-eve nga rrjetet e komunikimit. Risia qëndron në mënyrën e integruar të qasjes: nga transformimi i të dhënave në NoSQL, te analizimi i tyre në Spark, deri te interpretimi i marrëdhënieve përmes metrikave të centralitetit. Kjo qasje jo vetëm që demonstroi kompetencë teknike, por edhe fuqinë analitike të Big Data në zbulimin e strukturave të fshehura në dataset-e të mëdha.

## I. Hyrje

Në epokën digjitale, sasia e të dhënave që gjenerohet çdo ditë është jashtëzakonisht e madhe. Kjo ka sjellë nevojën për mjete dhe teknologji që mund të përpunojnë, analizojnë dhe kuptojnë këto të dhëna në mënyrë efikase. Big Data nuk është vetëm një term teknologjik, por një fushë e tërë që ndihmon në marrjen e vendimeve të informuara në fusha si biznesi, shkenca dhe siguria kibernetike.

Ky projekt fokusohet në analizën e Big Data duke përdorur tre komponentë kryesorë: MongoDB për ruajtjen e të dhënave, Apache Spark për përpunimin e tyre dhe mjete të analizës së rrjeteve sociale (Social Network Analysis) si ORA Lite dhe Gephi për të kuptuar lidhjet dhe modelet që dalin nga këto të dhëna.

Duke përdorur një datasete reale, projekti synon të tregojë se si mund të nxirren njohuri të vlefshme nga të dhënat e mëdha përmes një qasjeje të koordinuar dhe teknologjikut moderne.

## II. Qëllimi i punimit

Qëllimi kryesor i këtij projekti është të demonstrojë përdorimin praktik të teknologjive të Big Data për analizën e të dhënave të mëdha dhe të ndërlikuara, me fokus të veçantë në komunikimet në rrjet. Nëpërmjet një kombinimi të mjeteve si MongoDB, Apache Spark dhe Social Network Analysis, synohet:

- Ruajtja dhe përpunimi efikas i një dataset-i real.
- Zbulimi i modeleve të komunikimit dhe strukturave të rrjetit përmes analizës së grafëve.
- Vizualizimi i lidhjeve dhe marrëdhënieve ndërmjet individëve në një organizatë.
- Demonstrimi i aplikimeve praktike të Big Data në fushën e inteligjencës, biznesit dhe sigurisë.

Në thelb, projekti synon të tregojë se si teknologjitë moderne mund të përdoren për të transformuar të dhënat e thjeshta në njohuri të dobishme.

### III. Pjesa kryesore

#### 3.1. Përdorimi i NoSQL

NoSQL përdoret për të ruajtur të dhëna në mënyrë fleksibile, pa një skemë fikse të përcaktuar paraprakisht. Kjo e bën atë shumë të përshtatshëm për aplikacione që punojnë me sasi të mëdha të dhënash që ndryshojnë vazhdimisht në strukturë apo përmbajtje, si rrjetet sociale, IoT, ose analiza të mëdha të të dhënave.

Në projektin tonë, ne përdorëm MongoDB, një nga sistemet më të përdorura NoSQL, për të migruar të dhëna nga baza relacionare Mondial. Të dhënat u ruajtën në format JSON – formë që pasqyron mënyrën natyrore të organizimit të dokumenteve në MongoDB. Kjo strukturë e bazuar në dokumente na lejoi të shmangim nevojën për join-e komplekse, të zakonshme në SQL, dhe të ekzekutojmë pyetje më efikase dhe më të thjeshta për raste të caktuara.

Tabela 1: Dallimet kryesore mes SQL dhe NoSQL:

Aspekti	SQL (Relacional)	NoSQL (Jo-relacional)
Struktura e të dhënave	Tabela me rreshta dhe kolona	Dokumente JSON, key-value, grafë, kolonë
Skema	E fiksuar, kërkon skemë të përcaktuar	Fleksibile, ndryshon dinamikisht
Marrëdhëniet (joins)	Mbështet JOIN midis tabelave	Zakonisht nuk mbështet JOIN
Përshtatshmëria	E përshtatshme për të dhëna të strukturuar	E përshtatshme për Big Data dhe fleksibilitet
Shkallëzimi	Shkallëzim vertikal (më shumë burime)	Shkallëzim horizontal (shumë serverë)
Integriteti	Mbështet ACID	Mbështet eventual consistency / CAP
Shembuj teknologjish	MySQL, PostgreSQL, Oracle	MongoDB, Cassandra, Redis

Ndryshe nga bazat relacionale që kërkojnë një strukturë të rreptë dhe të paracaktuar të skemës, sistemet NoSQL ofrojnë një qasje më fleksibile, duke e bërë më të lehtë ruajtjen dhe përpunimin e të dhënave që ndryshojnë shpesh ose nuk ndjekin një format të rregullt. Nëse SQL është më i përshtatshëm për të dhëna të ndërlidhura dhe transaksione të sakta, NoSQL shkëlqen në skenarë me shkallëzim të madh dhe me nevoja për shpejtësi dhe fleksibilitet.

Gjithashtu, në një sistem relacionar, normalizimi dhe përdorimi i JOIN janë pjesë kyçe për ruajtjen e integritetit të të dhënave. Në anën tjetër, NoSQL shmang këto lidhje duke ruajtur të dhëna të ngjashme brenda një dokumenti të vetëm, duke rritur performancën por duke sakrifikuar disa garanci transaksionale.

Zgjedhja midis SQL dhe NoSQL varet nga kërkesat specifike të projektit: nëse kemi nevojë për integritet dhe transaksione të forta, SQL është zgjidhja më e mirë; por për aplikacione të mëdha, dinamike dhe me ngarkesa të mëdha leximi/shkrimi, NoSQL është më praktik dhe më i shkallëzueshëm.

### 3.1.1. Përgatitja e Mjedisit

#### Mjeti i përzgjedhur dhe arsyeimi

Për këtë pjesë të projektit është përzgjedhur MongoDB si platforma NoSQL. MongoDB është një bazë të dhënash dokumentesh (document-oriented database) që përdor dokumente në format JSON/BSON në vend të tabelave tradicionale. Arsyeja kryesore e zgjedhjes së MongoDB lidhet me:

- mbështetjen e mirë për strukturat fleksibile dhe dinamike të të dhënave (ideal për migrimin nga SQL),
- mundësinë për të përdorur `find`, `aggregate`, `filter`, dhe `map-reduce`,
- komunitet të madh dhe dokumentacion të pasur,
- integrim të lehtë me skripta për analiza dhe automatizim me `mongosh`.

MongoDB gjithashtu ofron një shell interaktiv (`mongosh`) që mbështet ekzekutimin e JavaScript me `async/await`, çka e ka bërë të përshtatshëm për migrimin dhe transformimin e query-ve SQL në logjikë NoSQL.

### 3.1.2. Migrimi i të Dhënave

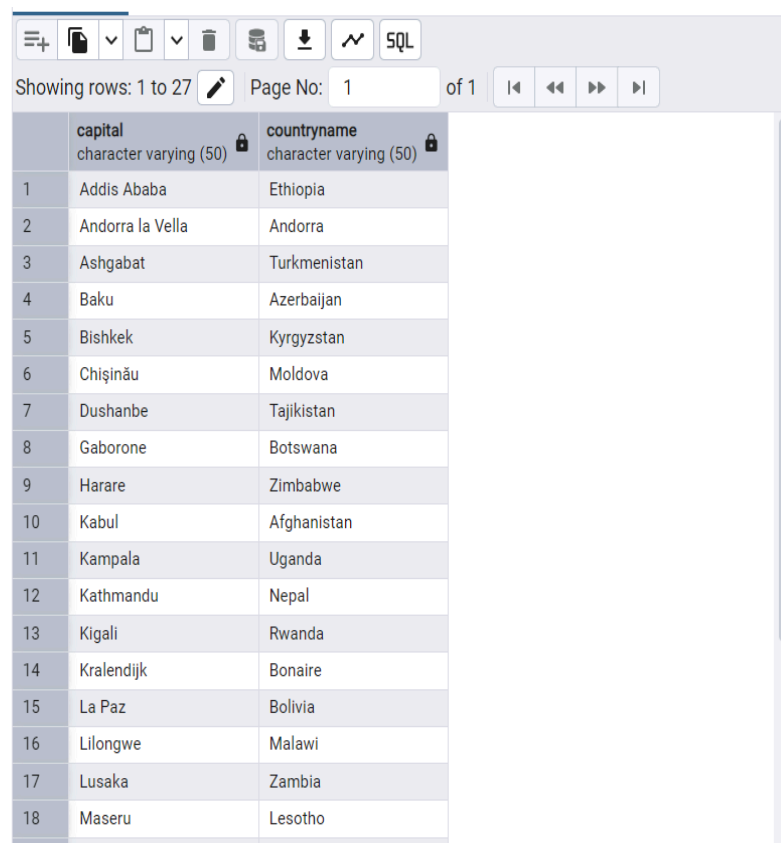
Në këtë projekt janë përzgjedhur disa tabela nga baza e të dhënave Mondial, të cilat përmbajnë informacione relevante për analizën dhe query-t e planifikuara të marrura nga faza e pare. Tabelat e përzgjedhura përfshijnë, ndër të tjera:

- Country – përmban informacione për vendet si emri, sipërfaqja, popullsia etj.
- City – përmban të dhëna për qytetet kryesore dhe lidhjet e tyre me vendet përkatëse.
- River – përmban informacione për lumenjtë, përfshirë shtrirjen dhe lidhjet me vendet.
- Organization – përmban të dhëna për organizatat ndërkombëtare dhe anëtarët e tyre.

### 3.1.3. Query-t në NoSQL

**Query 1: Listimi i të gjitha kryeqyteteve që nuk kalon asnjë lum nëpër to dhe shteteve që nuk kanë dalje në detë:**

- Kjo query kërkon të gjejë kryeqytetet për të cilat nuk kalon asnjë lumë dhe njëkohësisht, të listojë shtetet që nuk kanë dalje në det (landlocked).
- Rezultati në JSON-format (**NoSQL**) identik me atë në **SQL**:



The screenshot shows a database interface with a table containing 18 rows of data. The table has two columns: 'capital' and 'countryname'. The interface includes a toolbar with icons for various actions and a status bar indicating 'Showing rows: 1 to 27' and 'Page No: 1 of 1'.

	capital character varying (50)	countryname character varying (50)
1	Addis Ababa	Ethiopia
2	Andorra la Vella	Andorra
3	Ashgabat	Turkmenistan
4	Baku	Azerbaijan
5	Bishkek	Kyrgyzstan
6	Chişinău	Moldova
7	Dushanbe	Tajikistan
8	Gaborone	Botswana
9	Harare	Zimbabwe
10	Kabul	Afghanistan
11	Kampala	Uganda
12	Kathmandu	Nepal
13	Kigali	Rwanda
14	Kralendijk	Bonaire
15	La Paz	Bolivia
16	Lilongwe	Malawi
17	Lusaka	Zambia
18	Maseru	Lesotho

Figura 1: Ekzekutimi i Query 1 në NoSQL

**Query 2: Listimi i kryeqyteteve dhe shteteve përkatëse që nuk janë pjesë e asnjë organizate botërore**

- **Përshkrimi në fjalë të thjeshta**

Kjo query identifikon ato kryeqytete dhe vendet përkatëse që nuk janë anëtare të asnjë organizate ndërkombëtare.

- Rezultati në JSON-format (**NoSQL**) identik me atë në **SQL**:

```
mondial> result
[
  { name: 'Bonaire', capital: 'Kralendijk' },
  { name: 'Ceuta', capital: 'Ceuta' },
  { name: 'Christmas Island', capital: 'Flying Fish Cove' },
  { name: 'Cocos Islands', capital: 'West Island' },
  { name: 'Gaza Strip', capital: '' },
  { name: 'Melilla', capital: 'Melilla' },
  { name: 'Svalbard', capital: 'Longyearbyen' },
  { name: 'West Bank', capital: 'Ramallah' }
]
```

Figura 2: Ekzekutimi i Query 2 në NoSQL



## 3.2. Përpunimi i Big Data

### 3.2.1. Apache Spark në Google Colab

Apache Spark u përdor për përpunimin e të dhënave në këtë projekt si alternativë më moderne dhe më efikase ndaj Hadoop. Ndryshe nga Hadoop që funksionon mbi MapReduce dhe ruan të dhënat në disk, Spark funksionon në memorie, duke e bërë më të shpejtë dhe më të përshtatshëm për analizë interaktive. Spark ofron mbështetje për SQL, Python dhe DataFrame API, çka e bën ideal për projekte analitike dhe eksplorative.

Google Colab u përdor si platformë cloud për ekzekutimin e Spark, duke shmangur nevojën për konfigurim lokal ose për ndërtimin e një klasteri Hadoop.

### 3.2.2. Konfigurimi i Mjedisit Spark

Për instalimin e spark dhe hapjen e spark sessionit në python mund te referoheni ne hapa [ANNEX 1](#). [Pjesa 1 dhe 2]

- Spark u instalua në Google Colab përmes skriptave automatike me `apt`, `wget`, `tar` dhe `pip install findspark`.
- U përdor Spark 3.4.1 dhe Java 8.
- U krijua një SparkSession i personalizuar për analizat në projekt.
- Të gjitha komandat e konfigurimit u vendosën në fillim të notebook-ut [vendos rez].

### 3.2.2. Importimi i Mondial Dataset

Për pjesën e mondial mund të referoheni në [ANNEX 1](#).

Dataset-i **Mondial**, një koleksion i pasur me të dhëna për vendet, qytetet, lumenjtë dhe marrëdhëniet ndërkombëtare, u importua në mjedisin Spark për analizë eksploruese. Fillimisht, ai ishte në format **SQL** i ngarkuar në **MYSQL** dhe u konvertua në **CSV** ndërmjet një skripte në python për ta bërë të përputhshëm me Spark DataFrames. [Pjesa 3 në [ANNEX 1](#)]

Procesi përfshiu:

- **Parapërpunimin në Python** për të nxjerrë tabelat (shtetet, popullsia, qytetet, lumenjtë etj.) nga SQL në format CSV. [Python SQL to CSV](#)
- Ruajtjen e këtyre csv filë në github në mënyrë që të dhënat të jenë të qasshme për lehtësi aksesit nga Colab dhe nga gjithkush.
- Më pas, CSV-të u importuan si Spark DataFrame përmes komandës `spark.read.csv(...)`, me opsione për header dhe inferim të tipit të kolonave.

Ky hap mundësoi trajtimin e të dhënave në mënyrë të strukturuar dhe efikase për analizë me SparkSQL.

Pasi u përpunua struktura e dataset-it dhe u krijuan DataFrame-t përkatëse, u vendos që analizat të kryheshin nëpërmjet SparkSQL. U krijua një Spark temporary view për çdo DataFrame, që lejonte përdorimin e sintaksës SQL mbi to gjatë sesionit aktiv. Kjo bëhet përmes funksionit `df.createOrReplaceTempView(table)`.

### 3.2.3. Ekzekutimi i Query-ve

Për ekzekutimin e Query-ve në spark është përdorur funksioni `spark.sql()` me të cilin, pas ngarkimit të tabelave në memorie mundemi të ju qasemi direkt sikurse të ishte SQL bazik.


Queryt e marra nga Faza 1 përfshijnë:

- **Query 5 (Faza 1):** Te listohen te gjithë lumenjte te cilet kalojne neper vendet antare te NATO-s dhe EU-se perjashtuar Suedinte dhe Francen.

Për të arritur qëllimin e këtij query së pari janë gjetur vendet që i takojnë NATO dhe EU në bazë të tabelës **organization** dhe pastaj janë hequr (filtruar) Suedia dhe Franca.

Kodi mund të gjendet në [Pjesa 4 në [ANNEX 1](#)]

Shembull i ekzekutimit:



Lumi	Shteti
Adda	Italy
Aller	Germany
Alz	Germany
Ammer	Germany
Arno	Italy
Breg	Germany
Brigach	Germany
Donau	Germany
Douro	Portugal
Douro	Spain
Drau	Italy
Ebro	Spain
Elbe	Germany
Etsch	Italy
Euphrat	Turkey
Fulda	Germany
Garonne	Spain
Guadalquivir	Spain
Guadiana	Portugal
Guadiana	Spain

Figura 3. Rezultati i query 3 në mondial

- **Query 3 (Faza 1):** Një përmbledhje e kontinenteve me numrin përkatës të shteteve, duke grupuar të dhënat sipas kontinentit dhe duke përdorur funksione agreguese si **COUNT**.

**Query 3 (Faza 1):** Të listohen të gjitha kontinentet dhe numri i shteteve për secilin.

Për të arritur këtë rezultat është përdorur tabela e shteteve dhe është grupuar kolona `continent`, duke numëruar shtetet që i përkasin secilit kontinent përmes funksionit agregues `COUNT(*)`.

Kodi mund të gjendet në [Pjesa 5 në [ANNEX 1](#)]

Shembull i ekzekutimit:



Kryeqyteti
Lisbon
London
Paris
Rome

Figura 4. Rezultati i query 3 në mondial

### 3.2.4. Importimi i Dataset-it

Për pjesën e “Flight Dataset” mund të referoheni në [ANNEX 2](#)

Dataset-i i dytë i përdorur në këtë projekt është “**Flight Data with 1 Million+ Records**”, një dataset i përpunuar dhe i strukturuar që përmban të dhëna të detajuara për fluturime ajrore.

- Emri i dataset-it: Flight Data with 1 Million+ Records
- Burimi: [Kaggle – polartech](#)
- Madhësia: ≈1 milion rreshta, ≈230 MB
- Format: CSV

Ky dataset përfshin kolona të rëndësishme si: FL\_DATE, OP\_UNIQUE\_CARRIER, ORIGIN, DEST, DEP\_DELAY, ARR\_DELAY, CANCELLED, DISTANCE, etj.

Për të mundësuar analizën:

- Dataset-i u ngarkua në Google Drive.
- U lexua në **Spark** përmes `spark.read.csv(...)` me opsionet `header=True` dhe `inferSchema=True`.
- Pas leximit, u krijua një **temporary view** në Spark për ta përdorur me SparkSQL

Kodi dhe rezultati i ekzekutimeve është vendosur në [Pjesa 2 dhe 4 [ANNEX 2](#)]

### 3.2.5. Ekzekutimi i query-ve ne flights dataset

Për analizën e dataset-it të fluturimeve u ndërtuan disa query për të nxjerrë insight-e rreth efikasitetit dhe kostos së fluturimeve në varësi të faktorëve të ndryshëm.

- **Query 1:** Fluturimet më të lira kundrejt më Eco-Friendly sipas Rrugës dhe Numrit të Ndalimeve

Ky query kishte për qëllim të analizonte diferencën midis fluturimeve me më pak ndalesa (zakonisht më efikase për mjedisin) dhe atyre me ndalesa të shumta, që shpesh kushtojnë më lirë. U vlerësuan rrugët më të përdorura, numri i ndalesave dhe mesatarja e vonesës ose kostoja, kur ishte e mundur.

Kodi mund të gjendet në [Pjesa 5 në [ANNEX 2](#)]

Shembull i ekzekutimit dhe krijimit të grafikut me **matplotlib**:

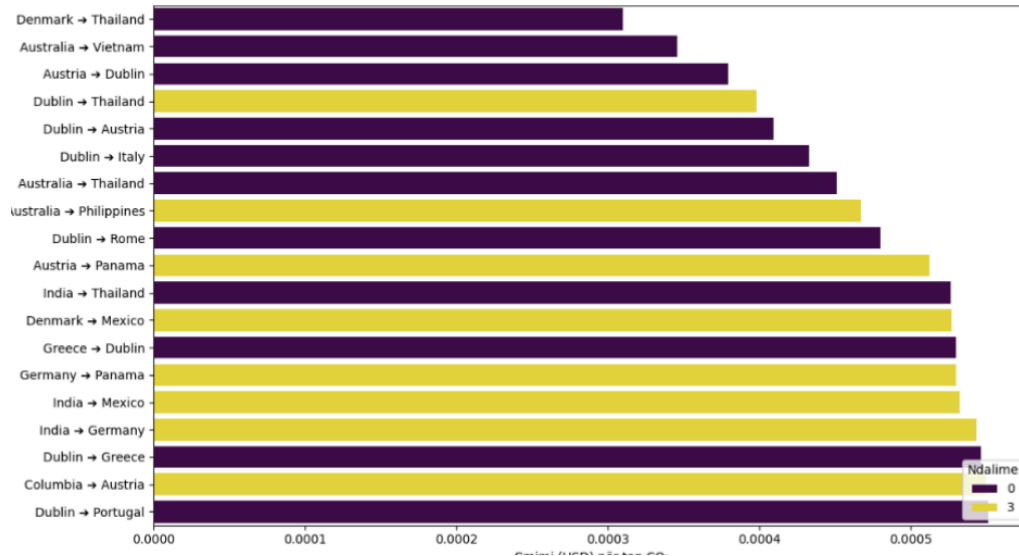


Figura 5: Ekzekutimi i query 2 në Flight Dataset

- Query 2: Sa % më lirë apo më shumë kushton të shtosh ndalesa?**

Ky query analizon se sa ndryshon çmimi mesatar i një fluturimi kur i shtohet një ndalesë krahasuar me një fluturim direkt. U bënë ndarje të dataset-it sipas numrit të ndalesave (NUM\_FLIGHTS ose STOPS) dhe u llogaritën mesataret për çmim apo distancë.

Ky krahasim lejon vlerësimin e kompromisit midis kohës dhe koston, duke analizuar nëse kursimi financiar justifikon kohën e shtuar të udhëtimit.

Kodi mund të gjendet në [Pjesa 5 në [ANNEX 2](#)]

Nga ekzekutimi i këtij query janë vërejtur fluturimet të cilat kanë kosto shumë më të ulët që janë direkte për kundër fluturimeve që kanë më shumë ndalime.

Shembull i ekzekutimit:

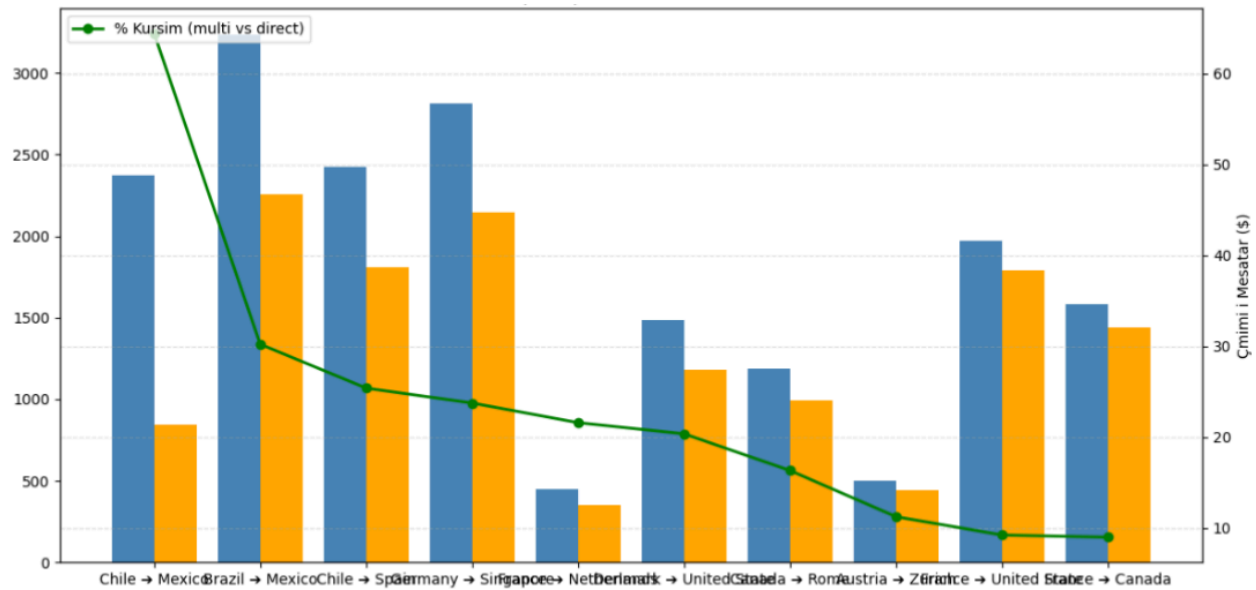


Figura 6: Ekzekutimi i query 2 në Flight Dataset

### 3.3. Analiza e Rrjeteve Sociale

Analiza e Rrjeteve Sociale (Social Network Analysis – SNA) është një metodë e fuqishme për të analizuar marrëdhëniet dhe ndërveprimet ndërmjet entiteteve të ndryshme (si individë, organizata, faqe interneti, etj.). Në vend që të fokusohet në vetitë individuale të entiteteve, SNA i jep rëndësi lidhjeve që ekzistojnë ndërmjet tyre, duke ndihmuar në zbulimin e strukturave të fshehura dhe modeleve të komunikimit brenda rrjetit.

Në një rrjet, nyjet përfaqësojnë entitetet, ndërsa lidhjet që njihen si edges tregojnë marrëdhëniet ose ndërveprimet ndërmjet tyre.

**Qëllimi kryesor i analizës së rrjeteve sociale është të zbulojë:**

- Rolin dhe ndikimin e nyjeve në rrjet
- Strukturat e brendshme të rrjetit, si komunitetet apo qendrat e kontrollit
- Rrugët më efikase për qarkullimin e informacionit
- Nyjet ndërmjetësuese që shërbejnë si ura komunikimi

**Përfitimet dhe fuqia e SNA**

- Lejon matjen e ndikimit relativ të çdo nyjeje përmes metrikave si:

- **Degree Centrality**- mat numrin e lidhjeve direkte që ka një nyje me nyjet e tjera në rrjet.
  - **Closeness Centrality**- mat sa afër është një nyje me të gjitha nyjet e tjera në rrjet, bazuar në distancën më të shkurtër.
  - **Betweenness Centrality**- mat se sa shpesh një nyje shfaqet në rrugët më të shkurtra midis dy nyjeve të tjera.
- Mundëson optimizimin e rrjeteve për efikasitet më të lartë në komunikim apo rrjedhje informacioni

**Formulat për llogaritjen e metrikave:**

**Degree Centrality:**

$$Degree(v) = k_v$$

$k_v$ - është numri i lidhjeve të lidhura me nyjen  $v$ .

**Closeness Centrality:**

$$Closeness(v) = \frac{n-1}{\sum_{u \neq v} d(u, v)}$$

$n$ - numri total i nyjeve në rrjet.

$d(v, u)$ - distanca më e shkurtër (me peshën më të ulët) nga nyja  $v$  te çdo nyje tjetër  $u$ .

**Betweenness Centrality:**

$$Betweenness(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

$\sigma_{st}$ - numri total i rrugëve më të shkurtra nga nyja  $s$  te nyja  $t$ .

$\sigma_{st}(v)$ - numri i atyre rrugëve që kalojnë përmes nyjës  $v$ .

### 3.3.1. Përdorimi i Ora Lite Tool

Instalimi i Ora Lite mund të bëhet nga faja zyrtare e [Casos Center](#).

Më pas nevojitet edhe një Ora Lite installation key i cili mund të gjindet pas startimit të shkarkimit të aplikacionit, pra pas disa procedurave të dhënies së disa informatave në lidhje me qëllimin e përdorimit të tool-it.

#### 3.3.1.1. Dataset-i i Përzgjedhur

Dataset-i i përdorur është "Enron Email Dataset", një prej dataset-eve më të njohura për analiza të komunikimit organizativ dhe SNA. Burimi origjinal është Carnegie Mellon University dhe dataset-i është shpërndarë publikisht për qëllime akademike. Dataset-i mund të gjindet në këtë [link](#).

#### Struktura e Nodes dhe Edges:

- Nodes: Çdo nyje përfaqëson një punonjës unik të kompanisë Enron. Atributet përfshijnë Id, Label, Role, Department, Gender, Age, Country.
- Edges: Çdo lidhje përfaqëson një komunikim pra një email të dërguar nga një punonjës tek një tjetër. Lidhjet janë të drejtuara dhe përfshijnë një peshë që tregon frekuencën e komunikimeve.

#### 3.3.1.2. Importimi në ORA Lite

1. Hapja e ORA Lite dhe zgjedhja e opsionit "Import Data".
2. Shtimi i fajllit me nyje [nodes.csv](#) duke specifikuar kolonën ID si identifikues të nyjës që realisht përfaqëson id-në e punëtorit dhe Label për emrin e shfaqur.
3. Shtimi i fajllit me lidhje [edges.csv](#) duke specifikuar strukturën Source, Target dhe Weight, ku Source përcakton se kush e dërgon email-in, Target përcakton se kujt i dërgohet email-i dhe Weight cila është pesha e atij relacioni mes atyre dy nyjeve .
4. Validimi i skemës për korrektësi të lidhjeve dhe përputhje të ID-ve.



### 3.3.1.3. Vizualizimi i Rrjetit

Pas importimit të file-ave për nyje dhe lidhje mes nyjeve në Workspace na krijohet një pamje e tillë:

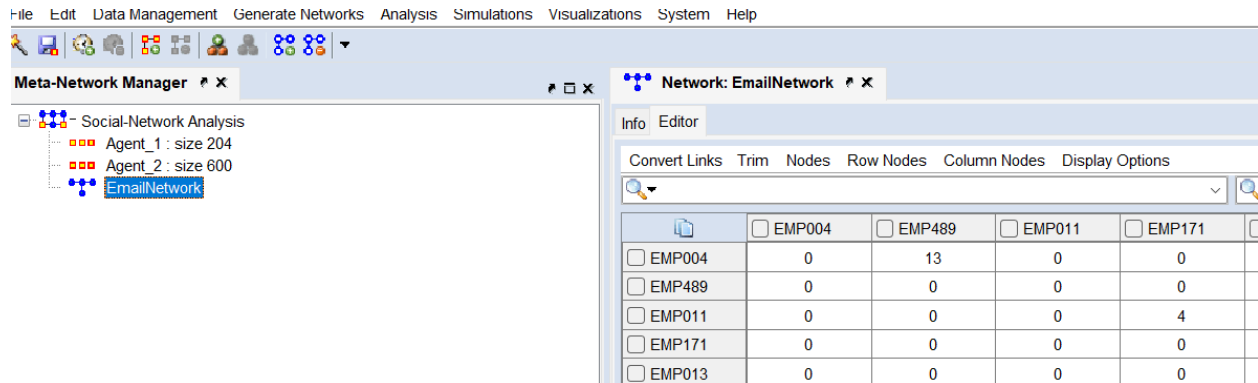
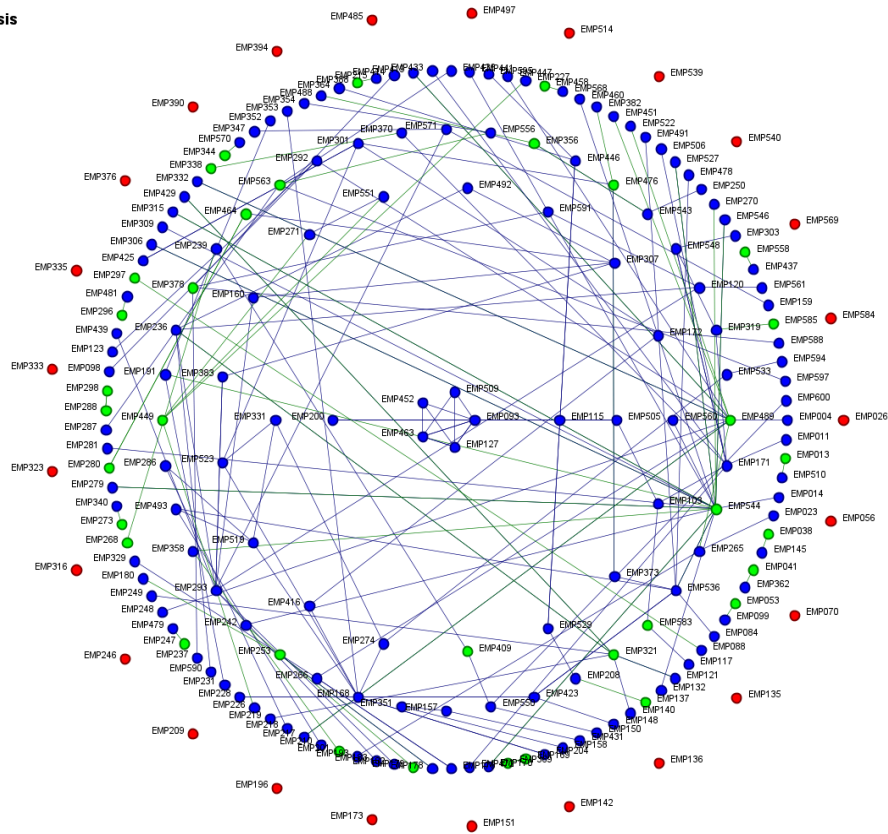


Figura 7: Workspace-i i krijuar pas importimit te file-ave

Klikojmë tek opsioni në Info më pas Visualize only this network dhe bëjmë disa konfigurime për ta paraqitur sa më qartë vizualizimin. Tek menyja klikojmë Layouts dhe zgjedhim nënopsionin Run Circle Layout(Pendants to Outside) i cili vendos nyjet në një formë rrethore me nyjet periferike (ato me pak lidhje) të pozicionuara në skaj. , dhe tek Node color zgjedhim opsionin Color By Concor Grouping që përdor një algoritëm të analizës së strukturës për të ndarë nyjet në grupe strukturore pra, grupime nyjesh që kanë modele të ngjashme të lidhjeve me pjesën tjetër të rrjetit, dhe fitojmë vizualizimin si më poshtë:

## Social-Network Analysis



powered by ORA

Figura 8: Vizualizimi bazë i krijuar nga rrjeti EmailNet

Nyjet me të njëjtën ngjyrë mund të përfaqësojnë, për shembull, punonjësit që komunikojnë më shumë mes vete ose që ndajnë role funksionale të ngjashme.

Dendësia e lidhjeve në brendi të grupit tregon bashkëpunim të brendshëm, ndërsa lidhjet ndërmjet grupeve reflektojnë komunikime ndër-funksionale ose ndër-departamentale.

Nyjet që ndodhen në qendër me më shumë lidhje dhe që shfaqen si pika lidhëse ndërmjet grupeve të ndryshme mund të konsiderohen si aktorë kyç të rrjetit (hub-s ose nyje ndërmjetësuese).

### 3.3.1.4. Kalkulimi i Metrikave të Centralitetit

Për kalkulimin e metrikave në Ora Lite kemi gjeneruar dhe disa raporte në mënyre automatike me opsionin Generate Reports.

Ajo çka na ka interesuar neve ka qenë kalkulimi i metrikave në formatin csv të cilat gjinden në këtë [link](#).

Ato informata i kemi organizuar në formë tabelare vetëm me Centrality metrikat si më poshtë:

Tabela 2: Kalkulimi i Centrality metrikave me Ora Lite

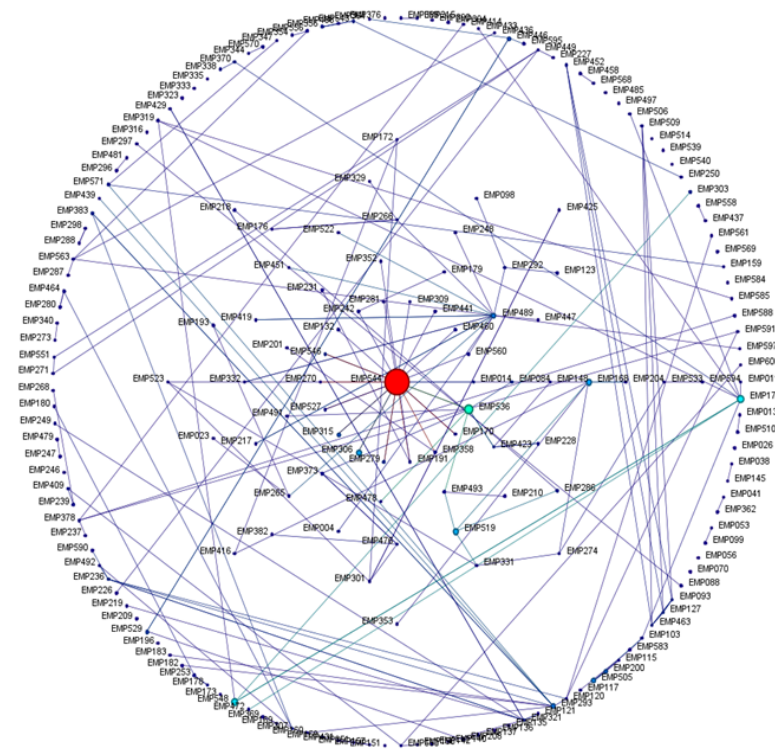
Node	Degree Centrality	Closeness Centrality	Betweenness Centrality
EMP544	0.017	0.000135	0.007
EMP536	0.006	0.000135	0.005
EMP171	0.005	0.000128	0.000732
EMP519	0.004	0.000129	0.0
EMP548	0.004	0.000128	0.000293
EMP489	0.003	0.00013	0.001
EMP168	0.003	0.000126	0.000122
EMP293	0.003	0.00013	0.002
EMP505	0.003	0.00013	0.000293
EMP200	0.003	0.00013	0.0
EMP321	0.002	0.000135	0.002
EMP423	0.002	0.000135	0.001
EMP084	0.002	0.000136	0.0
EMP014	0.002	0.000136	0.0
EMP236	0.002	0.000135	0.002
EMP476	0.000695	0.000137	0.001
EMP373	0.000695	0.000137	0.001
EMP120	0.000442	0.000131	0.001
EMP301	0.000442	0.000139	0.000951
EMP560	0.000379	0.000141	0.0
EMP132	0.00019	0.000136	0.0
EMP523	0.00019	0.000143	0.0
EMP425	0.000126	0.000139	0.0
EMP140	6.32e-05	0.000136	0.0

#### Degree Centrality përmes vizualizimit:

Ky vizualizim paraqet rrjetin social ku madhësia dhe ngjyra e nyjeve janë të bazuara në Degree Centrality, duke theksuar entitetet me më së shumti lidhje pra ata që kanë më shumë ndërveprime direkte me të tjerët – me nyjen qendrore (me të kuqe dhe më e madhe) që përfaqëson entitetin më të rëndësishëm në rrjet sipas numrit të lidhjeve të tij.

Edhe për këtë vizualizim është përdorur layout-i Center is Highest Betweenness.

Social-Network Analysis



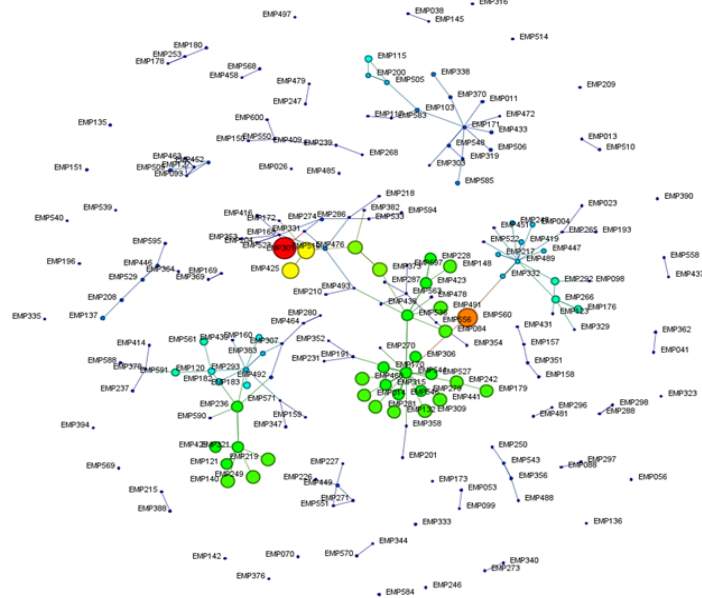
powered by ORA

Figura 9: Degree Centrality metrika përmes vizualizimit

### Closeneess Centrality përmes vizualizimit:

Ky vizualizim tregon rrjetin social me Closeness Centrality, ku nyjet më të mëdha dhe me ngjyrën më të ngrohtë (portokalli dhe e kuqe) janë ato që mund të arrijnë më shpejt çdo nyje tjetër në rrjet, duke reflektuar pozitën strategjike të tyre si shpërndarës të shpejtë të informacionit.

Social-Network Analysis



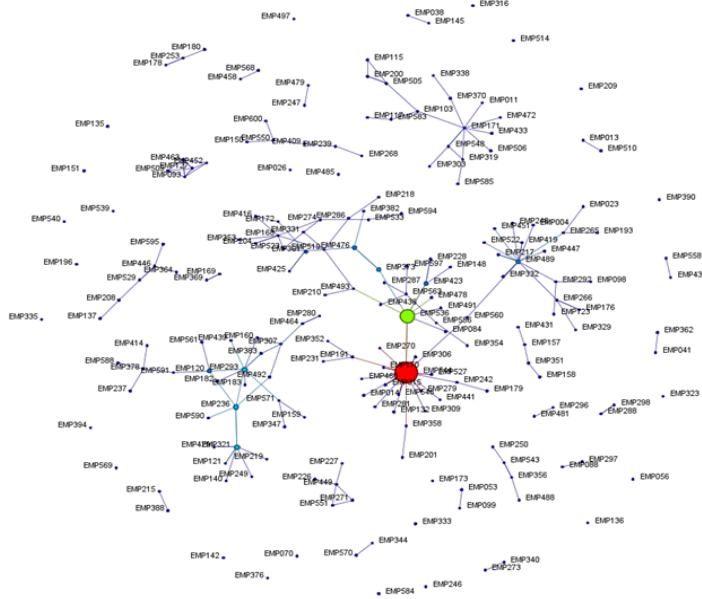
powered by ORA

Figura 10: Closeness Centrality metrika përmes vizualizimit

**Betweenness Centrality përmes vizualizimit:**

Ky vizualizim i bazuar në Betweenness Centrality tregon nyjet që veprojnë si urë lidhëse në rrjet pra nyjet me ngjyrë të kuqe dhe më të mëdha janë ato që shfaqen më shpesh në rrugët më të shkurtra midis çifteve të tjera, duke qenë kritik për rrjedhën e informacionit dhe ndërveprime.

Social-Network Analysis



powered by ORA

Figura 11: Betweenness Centrality metrika përmes vizualizimit

### 3.3.2. Përdorimi i Gephi Tool

Gephi është një platformë e fuqishme për analizën dhe vizualizimin e rrjeteve komplekse. Është open-source dhe përdoret gjerësisht në analiza të të dhënave sociale, biologjike, organizative etj. Instalimi i Gephi mund të bëhet nga <https://gephi.org/users/download> ku janë të disponueshme versionet për Windows, macOS dhe Linux.

#### 3.3.2.1. Dataset-i i Përzgjedhur

Dataset-i i përdorur në këtë analizë është një rrjet bashkë-autorie ndërmjet autorëve të ndryshëm të një publikimi akademik i cili mund të gjendet në këtë link [dataset](#). Dataset-i përbëhet nga dy file: nodes.csv dhe edges.csv.

- Nodes: Çdo nyje përfaqëson një autor unik. Atributet përfshijnë Id, Label (emri i autorit), si dhe fusha shtesë si institucion, vend ose fusha kërkimore (nëse të disponueshme).
- Edges: Çdo lidhje tregon një bashkëpunim ndërmjet dy autorëve në të njëjtin publikim. Lidhjet janë jo të drejtuara, dhe pesha (Weight) tregon numrin e publikimeve të përbashkëta.

### 3.3.2.3. Vizualizimi i Rrjetit

Pas importimit, vizualizimi i rrjetit realizohet përmes konfigurimeve të Layout ku zgjedhim Force Atlas 2, i cili shpërndan nyjet në hapësirë bazuar në lidhjet e tyre, duke pozicionuar nyjet me më shumë lidhje më afër qendrës. Ngjyrosja e nyjeve bëhet tek Appearance - Nodes - Color dhe zgjedhim Modularity Class, për të ndarë rrjetin në komunitete bashkëpunimi ndërsa madhësia e nyjeve tek Appearance - Nodes - Size dhe zgjedhim Degree, për të rritur madhësinë e nyjeve me më shumë lidhje.

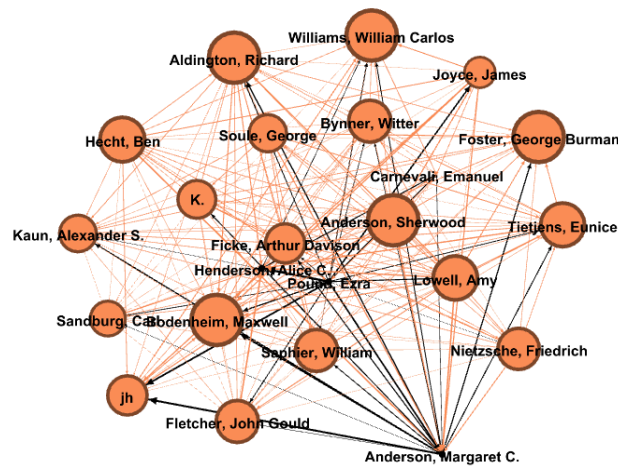


Figura 12: Vizualizimi përmes Gephi

### 3.3.2.4. Kalkulimi i Metrikave të Centralitetit

Për të llogaritur metrikat e centralitetit: Shkojmë tek Statistics dhe ekzekutojmë: Degree: Për të parë numrin e lidhjeve për çdo nyje. Closeness Centrality: Për të identifikuar sa shpejt një nyje mund të lidhet me nyjet e tjera. Betweenness Centrality: Për të identifikuar nyjet që ndërmjetësojnë më shumë rrugë të shkurtra në rrjet. Modularity: Për të ndarë rrjetin në komunitete të lidhura fort. Të dhënat ruhen automatikisht në Data Laboratory dhe mund të eksportohen si .csv për analizë shtesë. Të dhënat ruhen automatikisht në Data Laboratory dhe mund të eksportohen si .csv për analizë shtesë.

Tabela 3: Kalkulimi i Centrality metrikave me Gephi

Id	Degree	Closeness Centrality	Betweenness Centrality
Hecht, Ben	59	0.884615	0.0
Anderson, Sherwood	83	1.0	0.495671
Pound, Ezra	146	0.648402	1276.197787
Lowell, Amy	101	0.875	0.404762
Nietzsche, Friedrich	48	0.818182	0.404762
Joyce, James	57	0.607143	1.416306
Soule, George	55	0.73913	0.0
Anderson, Margaret C.	141	0.617241	5767.968395
Fletcher, John Gould	67	0.833333	9.847258
Tietjens, Eunice	106	0.866667	2.130592
K.	48	0.75	1.1
Aldington, Richard	94	1.0	13.832973
jh	78	0.777778	1.082973
Sandburg, Carl	47	0.692308	2.392496
Kaun, Alexander S.	54	0.727273	3.033333
Ficke, Arthur Davison	61	0.777778	1.511905
Bynner, Witter	50	0.833333	4.130592
Saphier, William	50	0.833333	5.189683
Bodenheim, Maxwell	85	1.0	10.37583
Foster, George Burman	56	1.0	2.011905
Henderson, Alice C.	139	0.825	2647.48521
Williams, William Carlos	55	1.0	0.090909
Carnevali, Emanuel	64	0.0	0.0



## IV. Konkluzione

Ky projekt demonstroi me sukses përdorimin praktik të teknologjive të avancuara për manipulimin dhe analizën e të dhënave të mëdha. Nëpërmjet MongoDB, u arrit ruajtja fleksibile e të dhënave të strukturuar nga Mondial pa kufizime relacionale, ndërsa Spark u përdor për analizë efikase dhe të shpejtë të dataset-eve të mëdha me query SQL-like. Pjesa e analizës së rrjeteve sociale, përmes ORA Lite dhe Gephi, tregoi qartë se si metrikat e centralitetit ndihmojnë në kuptimin e marrëdhënieve komplekse ndërmjet aktorëve. Rezultatet e arritura konfirmojnë qëllimin e projektit: të shfaqë fuqinë e Big Data jo vetëm në përpunimin masiv të të dhënave, por edhe në nxjerrjen e insight-eve domethënëse dhe vizualisht të interpretuara.

### Puna e Ardhshme

Për të zgjeruar projektin, mund të shtohen analiza krahasimore ndërmjet llojeve të ndryshme të NoSQL apo të aplikohet Spark Streaming për të trajtuar të dhëna në kohë reale. Gjithashtu, përdorimi i algoritmeve të mëtejshëm të grafëve (p.sh. komunitetet ose PageRank) mund të thellojë analizën e rrjeteve sociale dhe të zbulojë struktura të fshehura edhe më të ndërlikuara.

## Referencat

- [1] Mondial është marrë nga: [The MONDIAL Database](#)
- [2] Flight Dataset është marrë nga: [Flight Data with 1 Million or More Records](#)
- [3] Enron Email Dataset është marrë nga: [Enron Email Dataset](#)
- [4] Poetry Dataset është marrë nga : [Poetry Dataset](#)