DATA PROCESSING & ANALYSIS

ASSIGNMENT PRESENTATION

BY DIVYA DAS (S5426634)
MSC DSAI 2021-22

Problem Statement

PART A

- Understanding and implementation of various database technologies
- Three use case given
- ERD Diagram
- Relational database using SQL
- Graph database using Neo4j
- Document database using MongoDB

PART B

- ML Classification problem
- Three data set given (later added 2)
- Split train and test data
- Apply Neural Net. & a classification algorithm
- Apply Principal Component Analysis
- Apply Feature Selection Methods
- Time Comparison

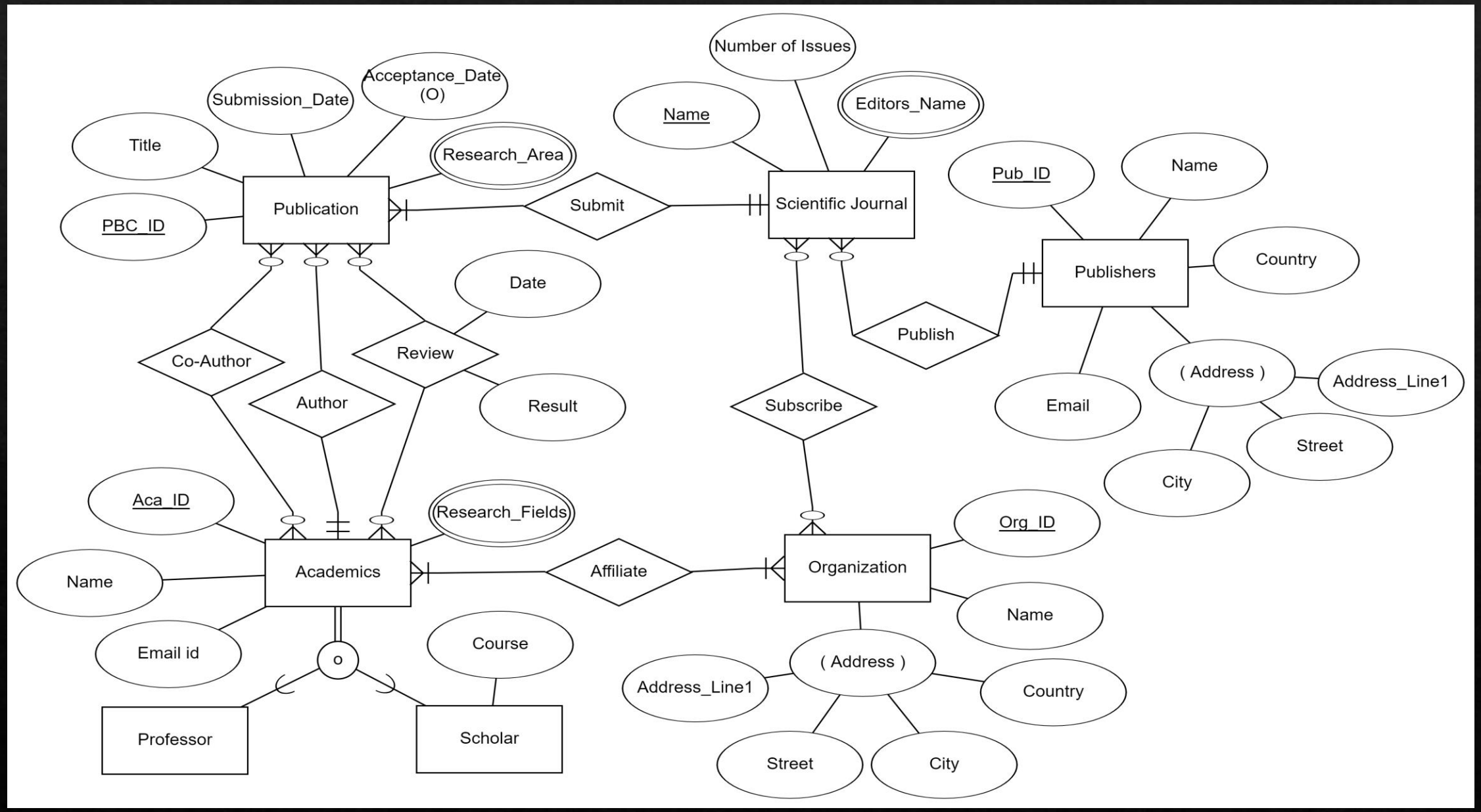
My Contribution

- I created a communication group and giving deadlines and following up on all group members
- Brainstorming and modifying the ERD
- Converted ERD to RL and Created Data based on real examples
- Implemented SQL, Neo4j and Mongo DB
- Active Participation in Part B at time to time
- Added 5000 words for PART- A report
- Added 1000 words for PART- B report
- Also put together the final assignment report before submission

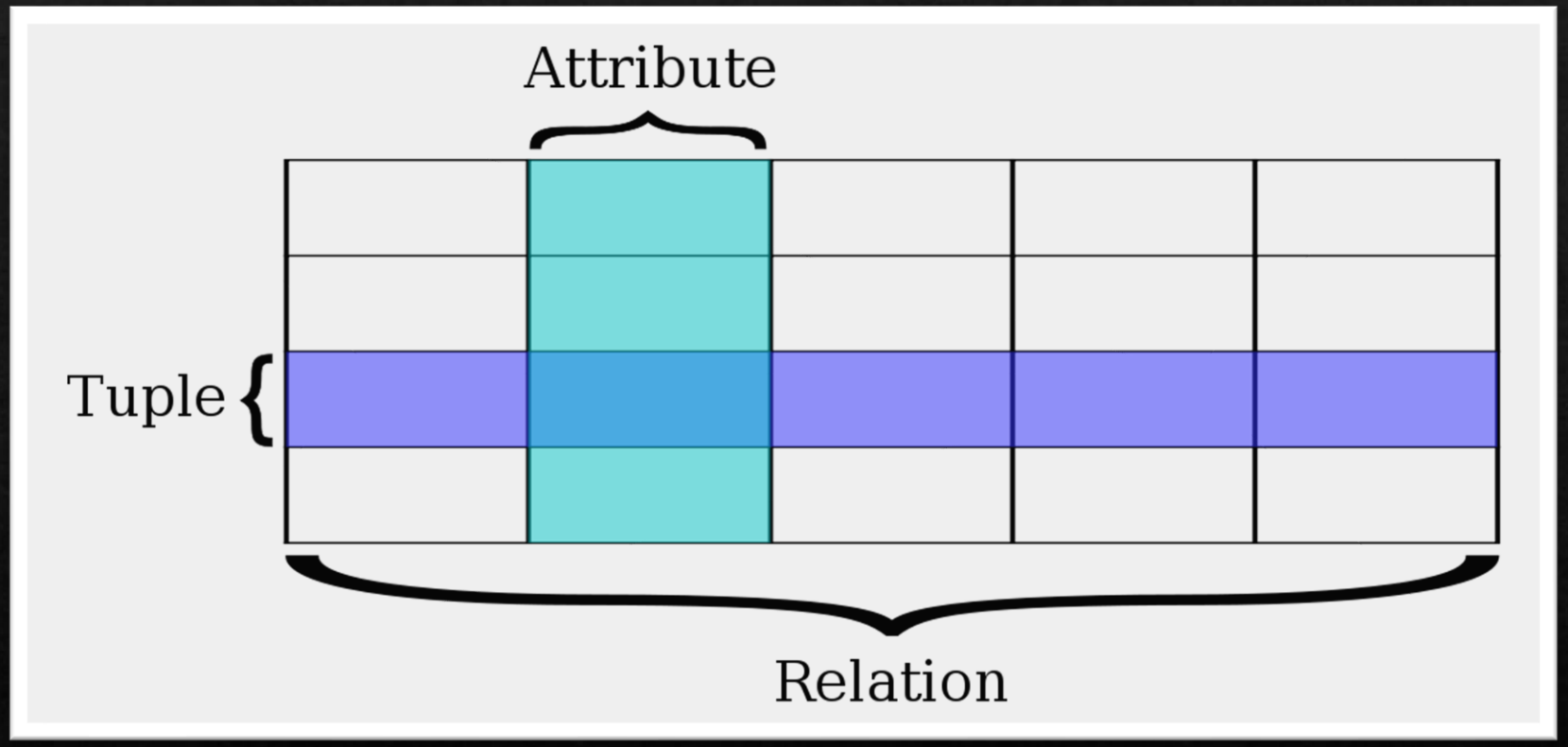
PART A – Data Processing

- Choosing the use case 2 – Database for Academic Publishing
- Draw the ERD Diagram
- Translate the ERD into Relational Database
- SQL to Implement Relational Database
- Graph Database
- Neo4J to implement Graph Database
- Document Database
- MongoDB to implement Document database

Entity Relationship Model (ERD Plus)



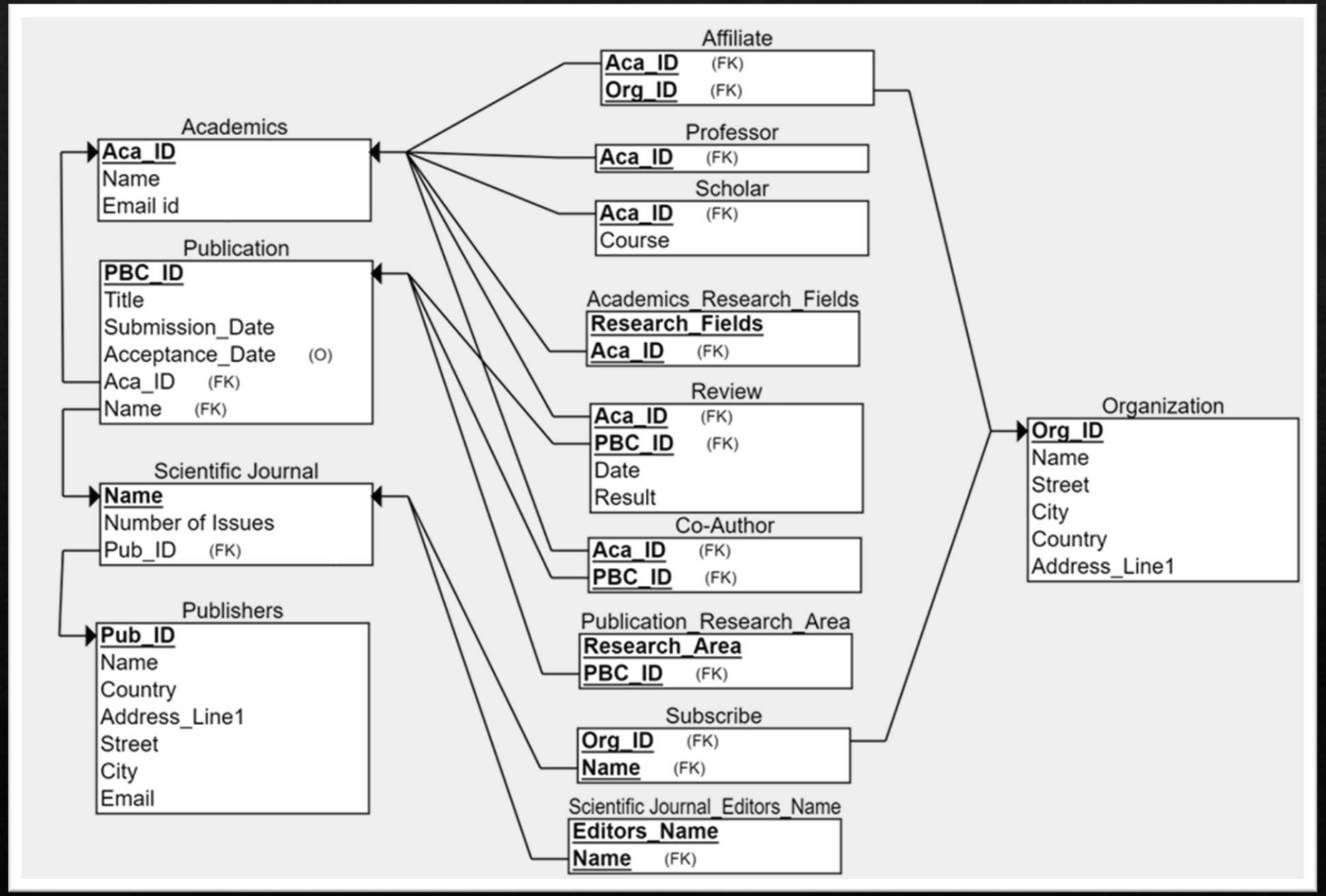
Relational Database



ERD to Relational Schema

Academics (Aca_ID , Name, Email_ID
Organization (Org_ID, Name, Address_Line1, Street, City, Country
Publication (Pbc_ID, Title, Submission_Date, Acceptance_Date, Aca_ID*, Journal_Name*)
Publisher (Pub_ID, Name, Address_Line1, Street, City, Country)
Scientific_Journal (Name, No_of_issue, Pub_ID*)
Co-author (Aca_ID*, Pbc_ID*)
Review (Aca_ID*, Pbc_ID* , Date, Result)
Affiliate (Aca_ID*, Org_ID*)
Subscribe (Org_ID*, Journal_Name*)
Research_Fields (Aca_ID*, Research_Field
Research_Area (Pbc_ID *, Research_Area
Editors_Name (Journal_Name*, Editor_Name
Professor (Aca_ID*,
Scholar (Aca_ID*, Course)

Relational Model from ERDPlus



Structured Query Language

- SQL is a standard language for RDBMS
- Chosen Entity – Academic, Publication and Journal
- Major Reason – These entities have multiple complex relations which require multiple joins to retrieve information
- Create table and insert data into tables
  - Entities- Academic, Publication and Journal
  - Relationships – Co-Author and Review
  - Multivalued Attributes – Research Field and Research Area

Test case 1 – Joined 2 table & Nested Query

1. **Retrieve all Academic papers authored by ‘Nicholas R. Jennings’ with author details.**

QUERY:

```
SELECT *
FROM publication p
JOIN academic a ON p.aca_id = a.aca_id
WHERE p.aca_id = (SELECT aca_id
                  FROM academic
                  WHERE name = 'Nicholas R. Jennings.');
```

OUTPUT:

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.065 seconds

	PBC_ID	TITLE	SUBMISSION_DATE	ACCEPTANCE_DATE	ACA_ID	JOURNAL_NAME	ACA_ID_1	NAME	EMAIL_ID
1	1131	An agent-base...	16-JAN-01	22-APR-01	130	Communication...	130	Nicholas R. Jennings.	nrjennings@hmail.com
2	1132	On agent-base...	26-JUN-00	30-SEP-00	130	Artificial In...	130	Nicholas R. Jennings.	nrjennings@hmail.com

Test case 2 – As, Joined 3 table & Nested Query

2. Retrieve the academic papers name which are co-authored by 'Anil K. Jain'.

QUERY:

```
SELECT p.title, a.name AS COAUTHOR
FROM publication p
JOIN coauthor ca ON p.pbc_id = ca.pbc_id
JOIN academic a ON ca.aca_id = a.aca_id
WHERE p.pbc_id IN (SELECT ca.pbc_id
                   FROM coauthor
                   WHERE ca.aca_id = (SELECT aca_id
                                      FROM academic
                                      WHERE name = 'Anil K. Jain'));
```

OUTPUT:

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.011 seconds

	TITLE	COAUTHOR
1	Biometric recognition: Security and privacy concerns	Anil K. Jain
2	On the similarity of identical twin fingerprints	Anil K. Jain
3	Fingerprint image enhancement: algorithm and performance evaluation	Anil K. Jain
4	Biometrics: a tool for information security	Anil K. Jain

Test case 3 – Union combines result of two statement

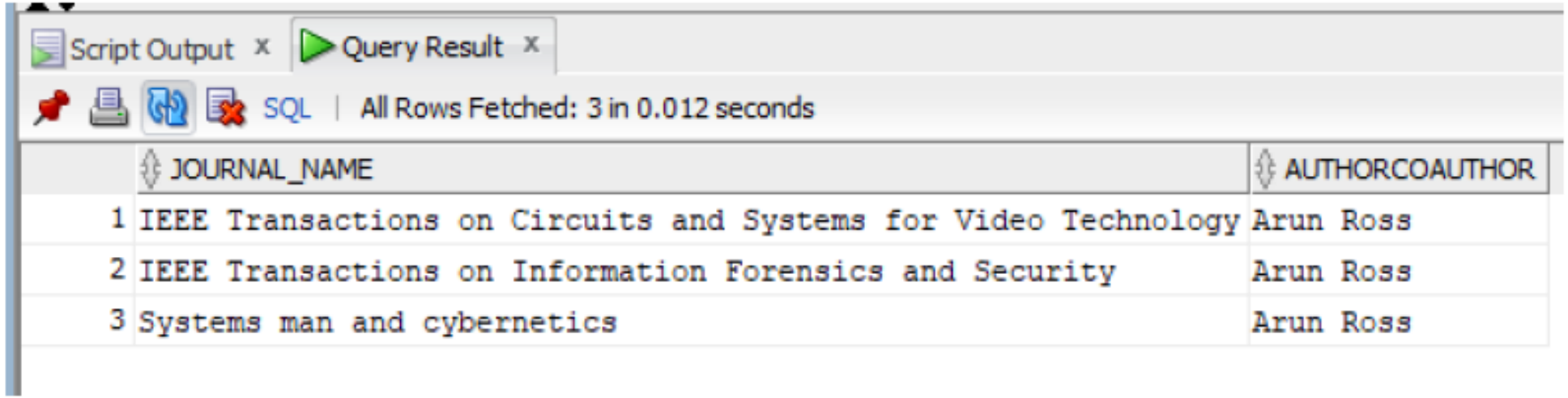
3. **Retrieve the Journal in which 'Arun Ross' has published either as Author or Co-author**

QUERY:

```
SELECT j.journal_name, a.name AS authorcoauthor
FROM journal j
JOIN publication p ON p.journal_name = j.journal_name
JOIN academic a ON p.aca_id = a.aca_id
WHERE p.aca_id IN (SELECT aca_id
                   FROM academic
                   WHERE name = 'Arun Ross')
UNION
SELECT j.journal_name, a.name AS authorcoauthor
FROM journal j
JOIN publication p ON p.journal_name = j.journal_name
JOIN coauthor ca ON p.pbc_id = ca.pbc_id
JOIN academic a ON ca.aca_id = a.aca_id
WHERE p.pbc_id IN ((SELECT ca.pbc_id
                    FROM coauthor
                    WHERE ca.aca_id = (SELECT aca_id
                                       FROM academic
                                       WHERE name = 'Arun Ross')));
```

Test case 3 – Result

OUTPUT:



	JOURNAL_NAME	AUTHORCOAUTHOR
1	IEEE Transactions on Circuits and Systems for Video Technology	Arun Ross
2	IEEE Transactions on Information Forensics and Security	Arun Ross
3	Systems man and cybernetics	Arun Ross

Test case 4 – LISTAGG WITHIN GROUP

4. Retrieve the Academics names and other interested research fields whose Research Field is 'The Internet' in single row for each Academic.

QUERY:

```
SELECT a.name, LISTAGG (rf.research_field,' ,') WITHIN GROUP(order by rf.research_field)
FROM academic a
JOIN research_field rf ON a.aca_id = rf.aca_id
WHERE a.aca_id IN (SELECT aca_id
                   FROM research_field
                   WHERE research_field = 'The Internet')
                   GROUP BY a.name;
```

OUTPUT:

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.006 seconds

	NAME	LISTAGG(RF.RESEARCH_FIELD,',')WITHINGROUP(ORDERBYRF.RESEARCH_FIELD)
1	Carole Goble	Operating system ,The Internet ,World Wide Web
2	David De Roure	Artificial intelligence ,Operating system ,The Internet
3	Nasir Menon	Artificial intelligence ,Operating system ,The Internet
4	Nigel Shadbolt	Artificial intelligence ,Law ,The Internet
5	Ping Wah Wong	Artificial intelligence ,Operating system ,The Internet

Test case 5 – Union two results then Join with Academic

5. **Select the name and paper title of Academics whose Research_Field is Computer Vision and published a paper with research area Computer vision.**

QUERY:

```
SELECT a.name, p.title
FROM academic a
JOIN research_field rf ON rf.aca_id = a.aca_id
JOIN publication p ON p.aca_id = a.aca_id
JOIN research_area ra ON ra.pbc_id = p.pbc_id
WHERE a.aca_id IN ((SELECT rf.aca_id
                    FROM research_field
                    WHERE rf.research_field = 'Computer vision')
                    UNION
                    (SELECT p.aca_id
                    FROM publication
                    WHERE p.pbc_id = (SELECT ra.pbc_id
                                      FROM research_area
                                      WHERE ra.research_area = 'Computer Vision')));
```

OUTPUT:

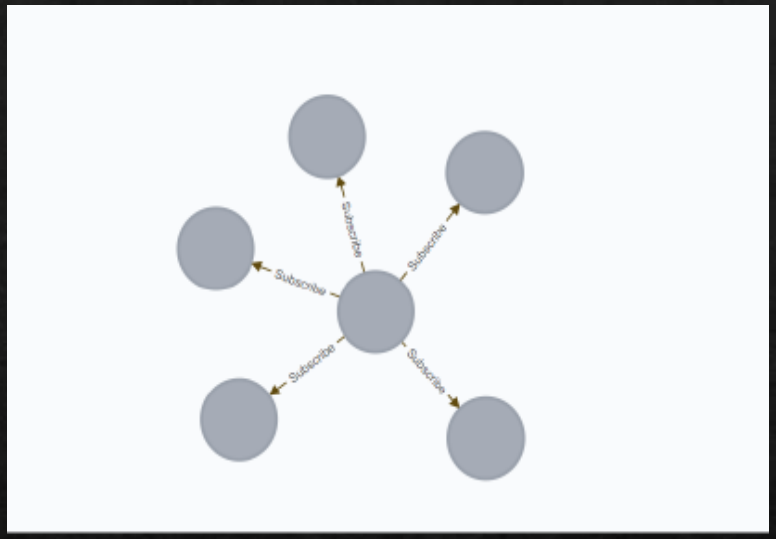
Script Output × Query Result ×

SQL | All Rows Fetched: 10 in 0.011 seconds

	NAME	TITLE
1	Salil Prabhakar	Biometric recognition: Security and privacy concerns
2	Sharath Pankanti	On the similarity of identical twin fingerprints
3	Arun Ross	Biometrics: a tool for information security
4	Mayank Vatsa	A Mosaicing Scheme for Pose-Invariant Face Recognition

Graph Database & Neo4j

G = (V, E, L_V, L_E, ID)
Where, V – set of nodes
E – set of edges
L_v – set of node labels
L_e – set of edge labels
ID – Unique identifiers of nodes and edges

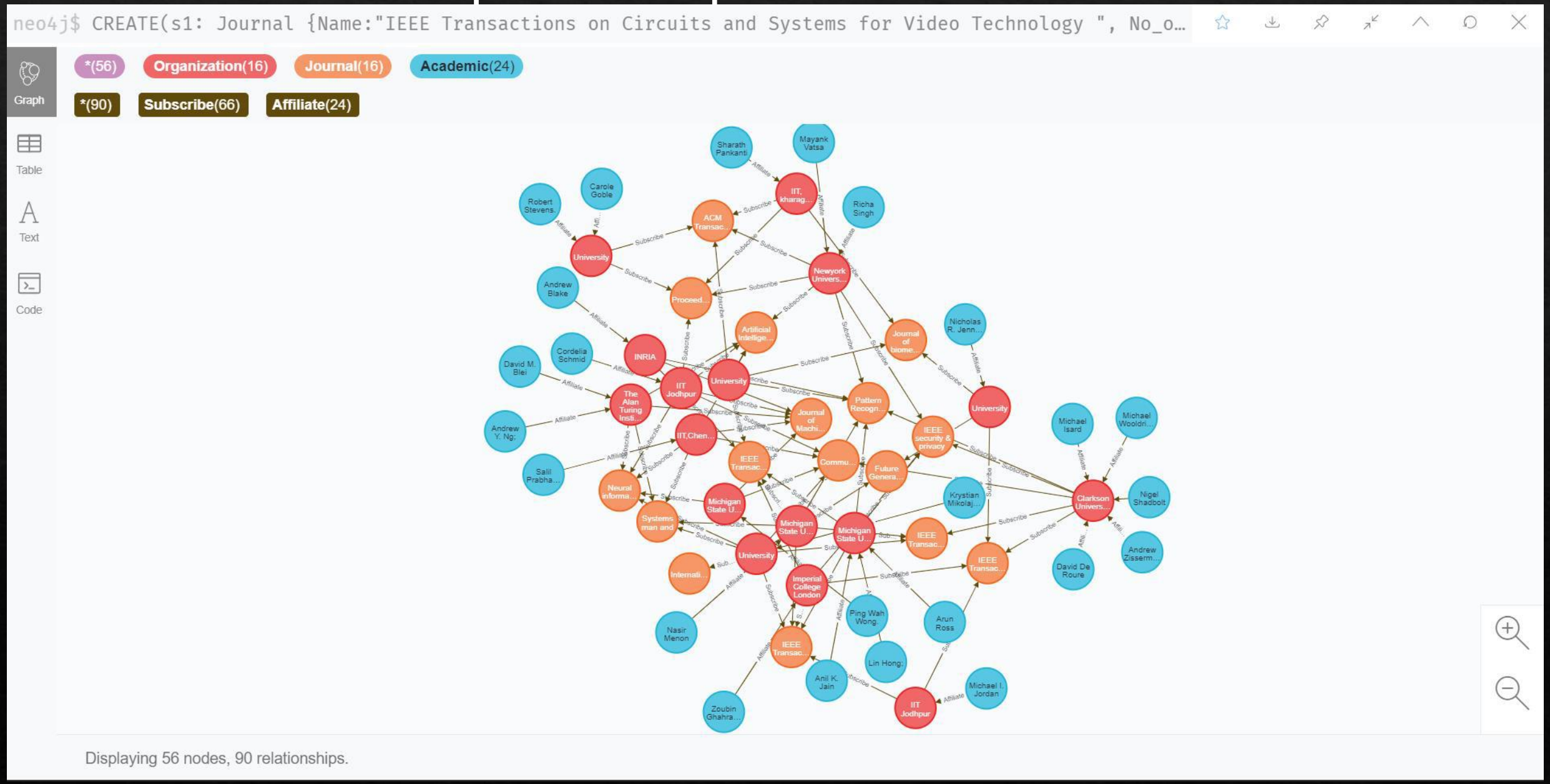


- Neo4j – Network Exploration and Optimization for Java -NoSQL
- Leading graph database solution provider
- Use Cypher Language
- Highly scalable and robust

NEO4J Implementation

- Chosen Entity – Academic, Organization and Journal
- Major Reason – These entities have many –to-many relationships and relation like subscriptions can keep changing and also scalable.
- Create Nodes and edges
  - Nodes- Academic, Organization and Journal
  - Edges – Subscribe and Affiliate

Complete Graph for our use case



Test case 1 – Simple query between two diff. nodes

1. **Which Academics are affiliated to Organization ‘Michigan State University’ and what are their Research field?**

QUERY:

```
MATCH (a:Academic{ })-[c: Affiliate]->( o:Organization{Name:"Michigan State University"})
RETURN a.Name,a. Research_Field
```

OUTPUT:

```
neo4j$ MATCH (a:Academic{ })-[c: Affiliate]→( o:Organization{Name:"Michigan State Univers…
```

Table | Text | Code

	a.Name	a. Research_Field
1	"Arun Ross"	["Artificial intelligence", "Computer vision", "Machine learning"]
2	"Lin Hong;"	["Biometrics", " Pattern Recognition"]
3	"Anil K. Jain"	["Artificial intelligence", "Computer vision", "Statistics"]
4	"Nasir Menon"	["Artificial intelligence", "Operating system", "The Internet"]
5	"Ping Wah Wong."	["Artificial intelligence", "Operating system", "The Internet"]

Started streaming 5 records after 3 ms and completed after 6 ms.

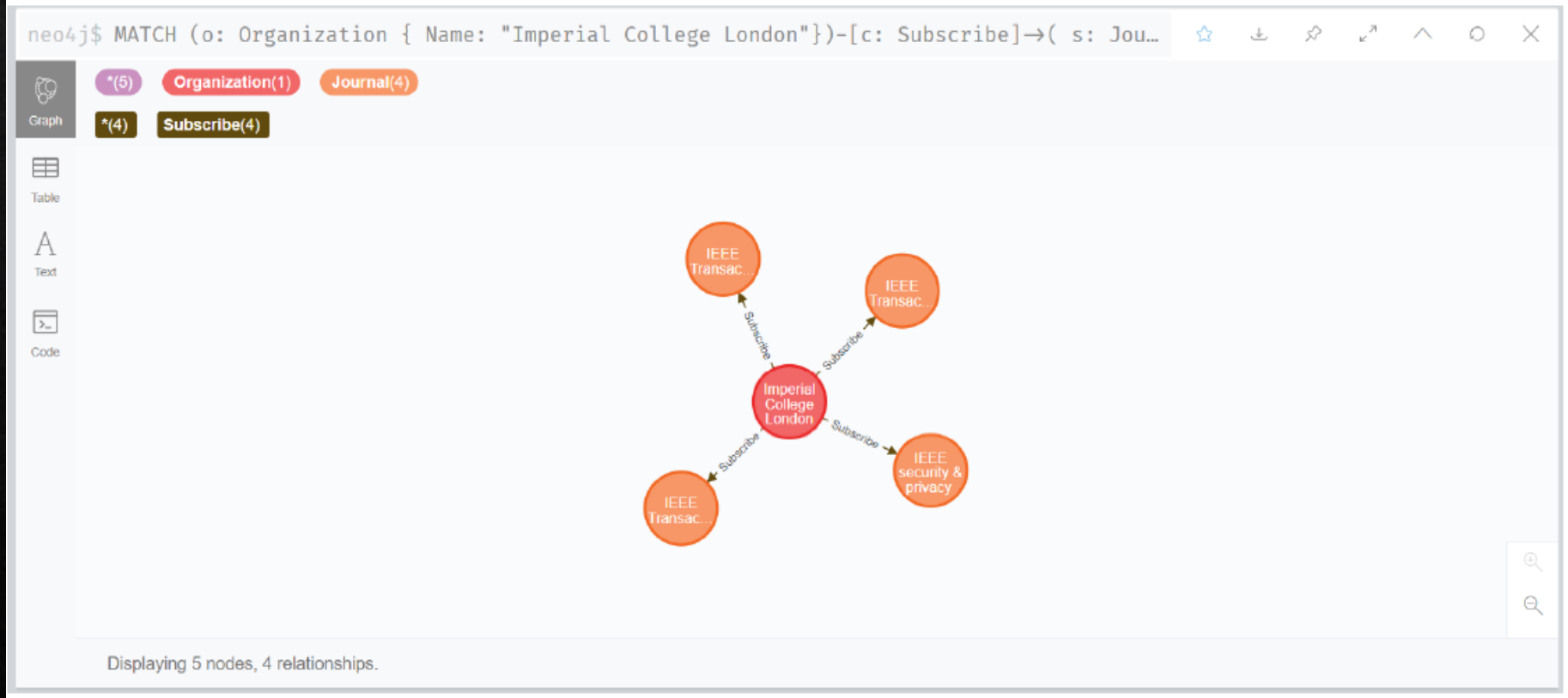
Test case 2 – Simple query between two diff. nodes

2. Organization ‘Imperial College London’ is subscribed to which Journals?

QUERY:

```
MATCH (o: Organization {Name: “Imperial College London”})-[c: Subscribe]->( s: Journal{ })
RETURN *
```

OUTPUT:



Test case 3 – Collect, Count & Order By

3. Get the Journals name subscribed by maximum number of universities and their names.

QUERY:

```
MATCH (o:Organization)-[s:Subscribe]->(k:Journal)
RETURN k.Name, COLLECT(distinct(o.Name)), COUNT(s) AS counts
ORDER BY counts DESC
LIMIT 5
```

OUTPUT:

```
neo4j$ MATCH (o:Organization)-[s:Subscribe]→(k:Journal) RETURN k.Name, COLLECT(distinct(o…
```

Table | Text | Code

	k.Name	COLLECT(distinct(o.Name))	counts
1	"Pattern Recognition "	["Michigan State University", "Newyork University", "Clarkson University", " INRIA", "University of California"]	6
2	"Communications of The ACM "	["IIT,Chennai", "Michigan State University", "Clarkson University", "IIT Jodhpur", "University of Oxford,"]	5
3	"IEEE Transactions on Information Forensics and Security "	["Michigan State University", "University of Oxford,", "Imperial College London", "IIT Jodhpur"]	5
4	"IEEE Transactions on Circuits and Systems for Video Technology "	["Michigan State University", " INRIA", "Imperial College London", "University of California"]	5
5	"Journal of Machine Learning Research "	["IIT,Chennai", "Michigan State University", "IIT Jodhpur", " INRIA", "The Alan Turing Institute"]	5

Started streaming 5 records after 3 ms and completed after 9 ms.

Test case 4 – Collect, Count & Order By

4. Get Academics with both “Computer vision” and “Statistics” as their Research Field?

QUERY:

```
MATCH (a:Academic)
WHERE "Computer vision" IN a.Research_Field
AND "Statistics" IN a.Research_Field
RETURN a.Name, a.Research_Field;
```

OUTPUT:

Test case 5 – Collect, Count & Order By

5. **Get Academics name affiliated to each Organization and show 5 organization with maximum number of academics affiliated?**

QUERY:

```
MATCH (a1:Academic)-[s:Affiliate]->(o:Organization)
RETURN o.Name, COLLECT(a1.Name), COUNT(s) AS counts
ORDER BY counts DESC
LIMIT 5
```

OUTPUT:

```
neo4j$ MATCH (a1:Academic)-[s:Affiliate]→(o:Organization) RETURN o.Name, COLLECT(a1.Name)…
```

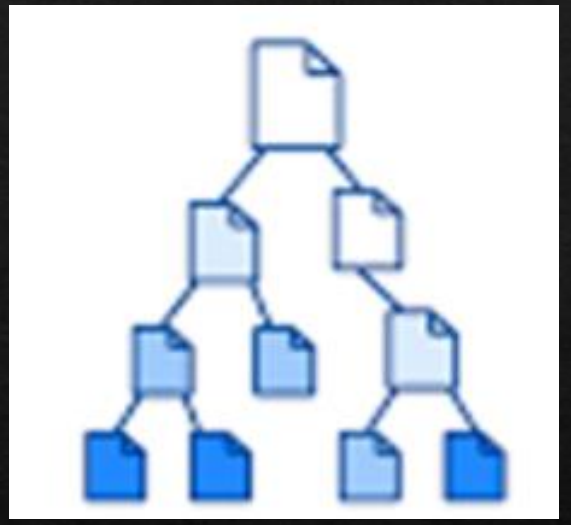
Table | Text | Code

	o.Name	COLLECT(a1.Name)	counts
1	"Clarkson University"	["Nigel Shadbolt", "David De Roure", "Michael Wooldridge", "Michael Isard", "Andrew Zisserman"]	5
2	"Michigan State University"	["Arun Ross", "Lin Hong;", "Anil K. Jain", "Nasir Menon", "Ping Wah Wong."]	5
3	"IIT Jodhpur"	["Cordelia Schmid", "Michael I. Jordan"]	2
4	"Newyork University"	["Richa Singh", "Mayank Vatsa"]	2
5	"The Alan Turing Institute"	["David M. Blei", "Andrew Y. Ng;"]	2

Started streaming 5 records after 3 ms and completed after 6 ms.

Document Database & Mongo DB

Mongo DB Element	SQL Equivalent
Collection	Table
Document (BSON)	Row
Field	Column
Index	Index
Embedded	Table Joins



- Mongo DB – open-source document database program - NoSQL
- Use JSON like document/BSON
- This is highly scalable and fast
- Two types of relation Embedding and Linking

Mongo DB Implementation

- Chosen Entity – Academic, Organization and Publication
- Major Reason – Highly scalable , Academic and Publication will scale continuously & Academic and Organization can have different subclasses easier to implement with flexible schema
- Create Documents
  - Academic Document with Embedded Organization
  - Publication Document , linked with Academic Document on Author & CoAuthor

Mongo DB Documents

```
mongosh mongodb+srv://<credentials>@cluster0.kuyup.mongodb.net/myFirstDatabase
Atlas atlas-gj1z2g-shard-0 [primary] myFirstDatabase> db.Academics.find({})
[
  {
    _id: ObjectId("62825bceb452a9bfb5a2404d"),
    ID: 'A01',
    Author: 'Anil K. Jainn',
    Email: 'anilkj@hmail.com',
    Research_Field: [ 'Artificial intelligence', 'Computer vision', 'Statistics' ],
    Organization: {
      Name: 'Michigan State University',
      Address_Line1: '426 Auditorium Road',
      Street: 'East Lansing, MI 48824',
      City: 'Michingan',
      Country: 'USA'
    }
  },
```

```
mongosh mongodb+srv://<credentials>@cluster0.kuyup.mongodb.net/myFirstDatabase
Atlas atlas-gj1z2g-shard-0 [primary] myFirstDatabase> db.Publications.find({})
[
  {
    _id: ObjectId("62825dd1404f95e362369eb1"),
    Title: 'An introduction to biometric recognition',
    Research_Area: 'Computer vision',
    Author: 'A01',
    CoAuthor: [ 'A05', 'A02' ],
    Submission_Date: ISODate("2003-10-27T00:00:00.000Z"),
    Approval_Date: ISODate("2004-02-01T00:00:00.000Z")
  },
```

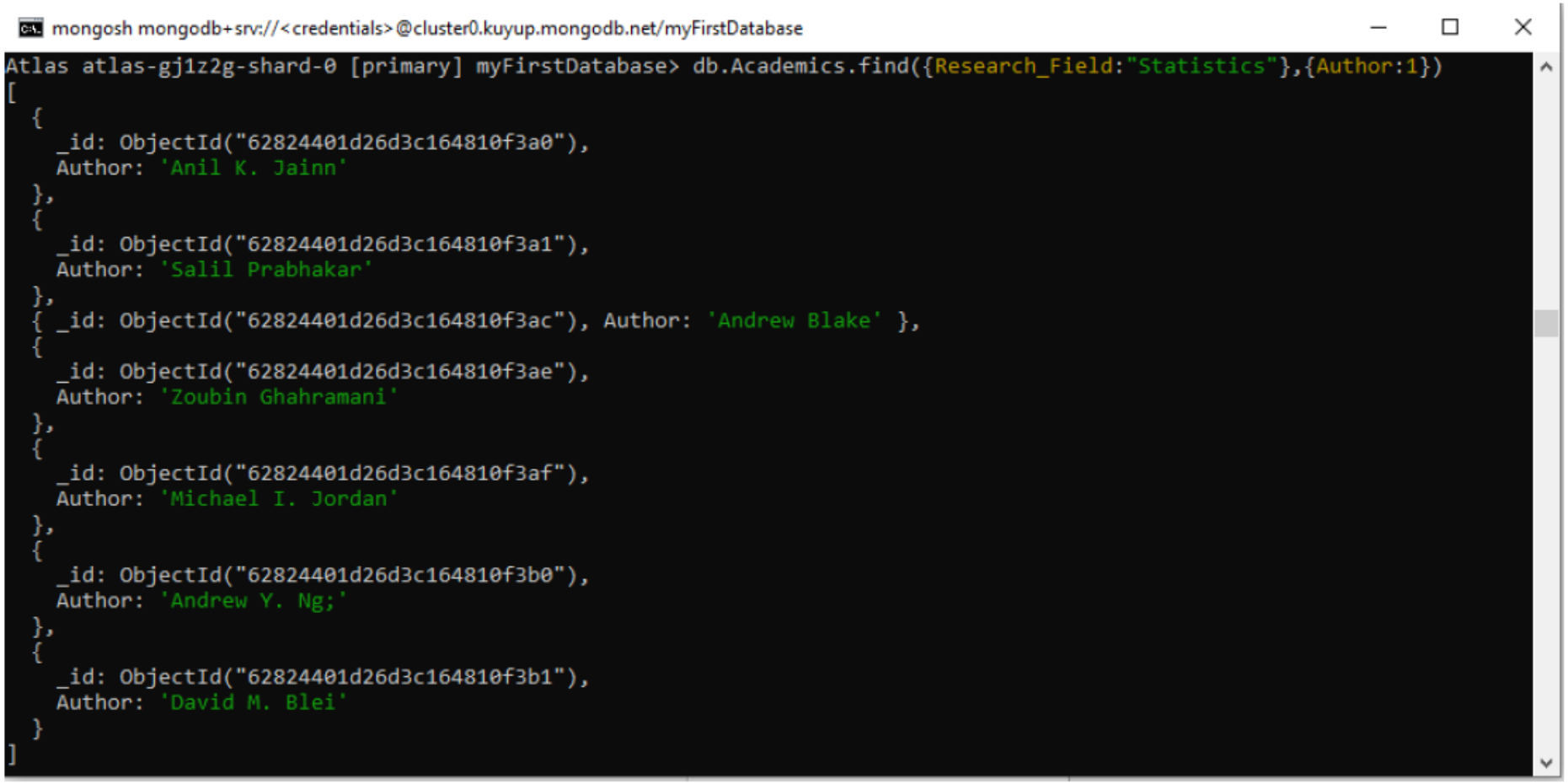
Test case 1 – elemMatch for Array of Research Field

1. Which Academics have ‘Statistics’ as their Research Field?

QUERY:

> db.Academics.find({$elemMatch: {" Research_Field ": ‘The Internet}})

OUTPUT:



Test case 2 – less than and ISO Date

2. Find publications that have submission date before the year 1997?

QUERY:

> **db.Publication.find** ({Submission_Date: {$lt: ISODate ("1997-01-01")}})

OUTPUT:

Select mongosh mongodb+srv://<credentials>@cluster0.kuyup.mongodb.net/myFirstDatabase

```
Atlas atlas-gj1z2g-shard-0 [primary] myFirstDatabase> db.Publications.find ({Submission_Date:{$lt: ISODate("1997-01-01")}})

[
  {
    _id: ObjectId("62825dd1404f95e362369ebf"),
    Title: 'The CONDENSATION Algorithm - Conditional Density Propagation and Applications to Visual Tracking',
    Research_Area: 'Computer vision',
    Author: 'A15',
    CoAuthor: 'A14',
    Submission_Date: ISODate("1996-04-17T23:00:00.000Z"),
    Approval_Date: ISODate("1996-07-21T23:00:00.000Z")
  },
  {
    _id: ObjectId("62825dd1404f95e362369ec0"),
    Title: 'Factorial Hidden Markov Models',
    Research_Area: 'Machine learning',
    Author: 'A16',
    CoAuthor: 'A17',
    Submission_Date: ISODate("1995-05-29T23:00:00.000Z"),
    Approval_Date: ISODate("1995-09-03T23:00:00.000Z")
  },
  {
    _id: ObjectId("62825dd1404f95e362369ec4"),
    Title: 'Intelligent Agents: Theory and Practice',
    Research_Area: 'Artificial intelligence',
```

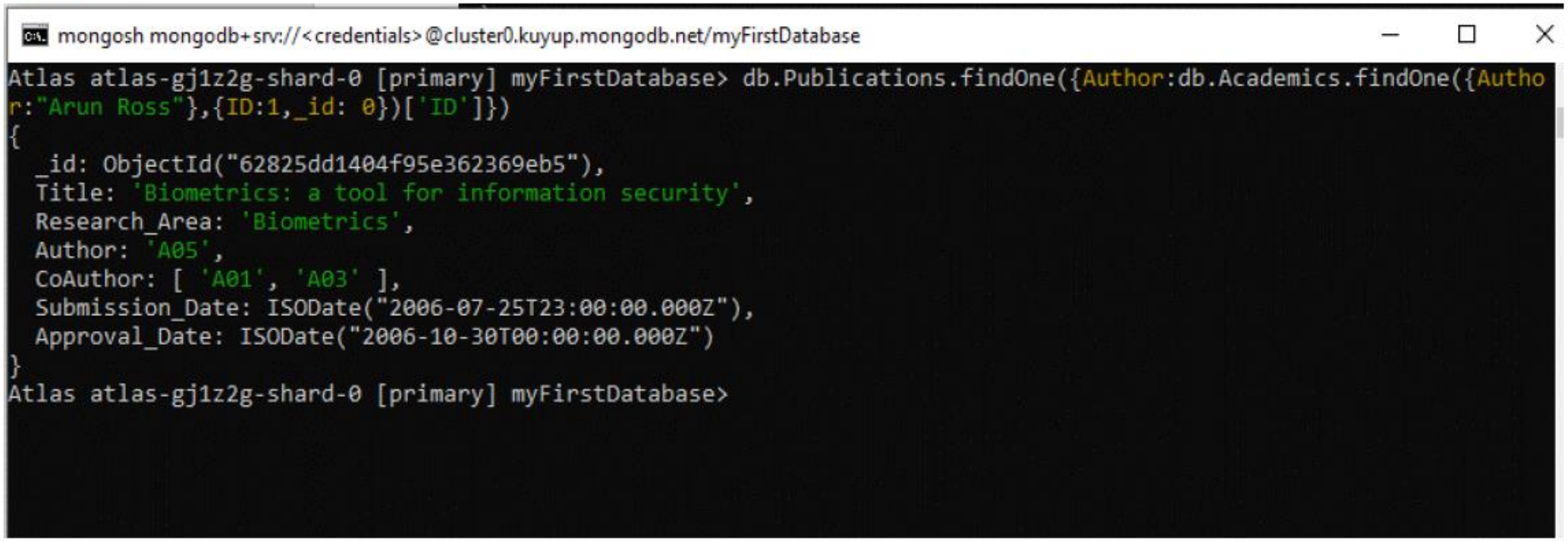
Test case 3 – findOne and Nested Query

3. Find a publication whose Author is "Arun Ross"?

QUERY:

> **db.Publication.findOne**({Author: db.Academics.findOne({Author: "Arun Ross"},{ID: 1, _id: 0})['ID']})

OUTPUT:



Test case 4 – find all & Nested Query

4. Find all publication whose Author is "Mayank Vatsa"?

QUERY:

> **db.Publication.find**({Author: db.Academics.findOne({Author: "Mayank Vatsa"} , {ID: 1, _id: 0})['ID']})]})

OUTPUT:

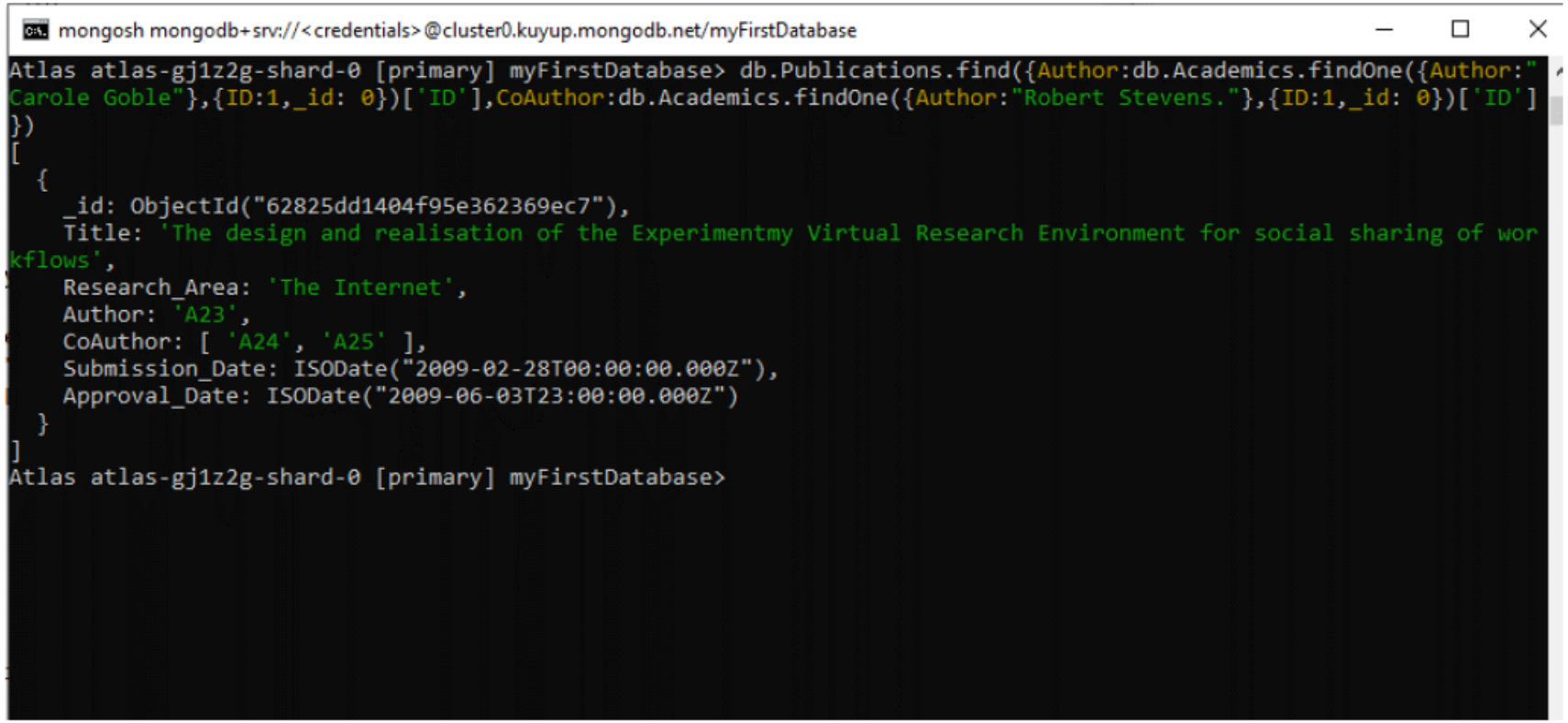
Test case 5 – Multiple Nested Query

5. **Find a publication whose Author is "Carole Goble" and Co-Author is Robert Stevens?**

QUERY:

> **db.Publication.find** ({Author: db.Academics.findOne({Author: "Carole Goble"},{ID: 1, _id: 0}) ['ID'],CoAuthor: db.Academics.findOne({Author: "Robert Stevens."}, {ID: 1, _id: 0}) ['ID']})

OUTPUT:



PART B – Data Analysis (WEKA)

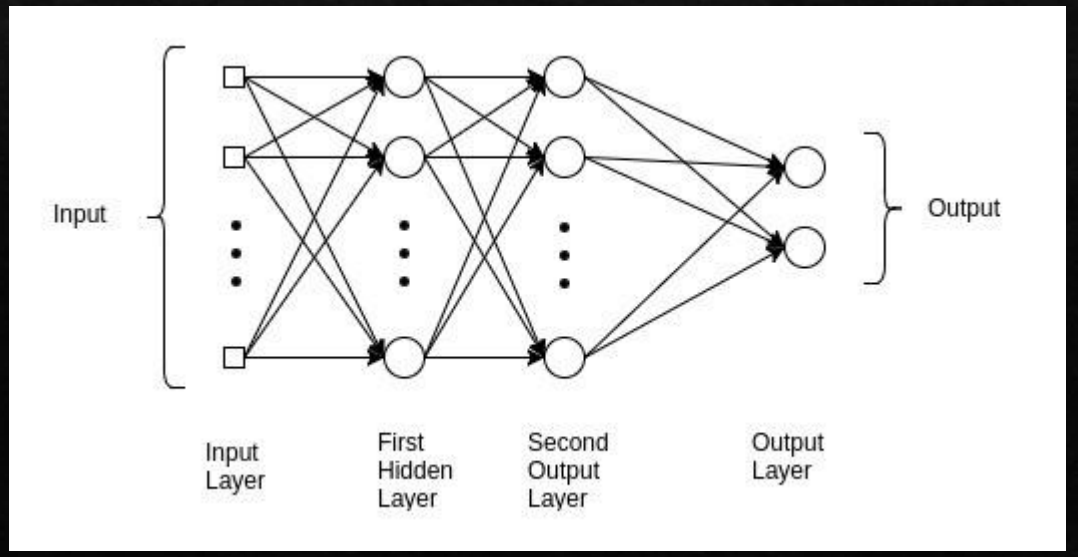
- Choosing the dataset – Epileptic Seizure Recognition
- Split Train & Test Data
- Neural Network Implementation – Multilayer Perceptron
- Classification Algorithm Implementation – Naïve Bayes
- Principal Component Analysis
- Feature Selection Methods
  - Correlation Based – Cfs Subset Evaluation
  - Wrapper Method – Wrapper Subset Evaluation
- Comparison of result after each stage

Train Test Split

- Used only 70% complete data due to heap memory errors (8050/11500)
- Changed target variable to nominal (1- Seizure & remaining-NonSeizure)
- Split into 80: 20 ratio. (Train -6440 , Test-1610)
- Used “Remove percentage Filter” from WEKA
- Used “ Invert Selection” in second time to get mutually exclusive datasets
- Exported as 2 separate .arf files

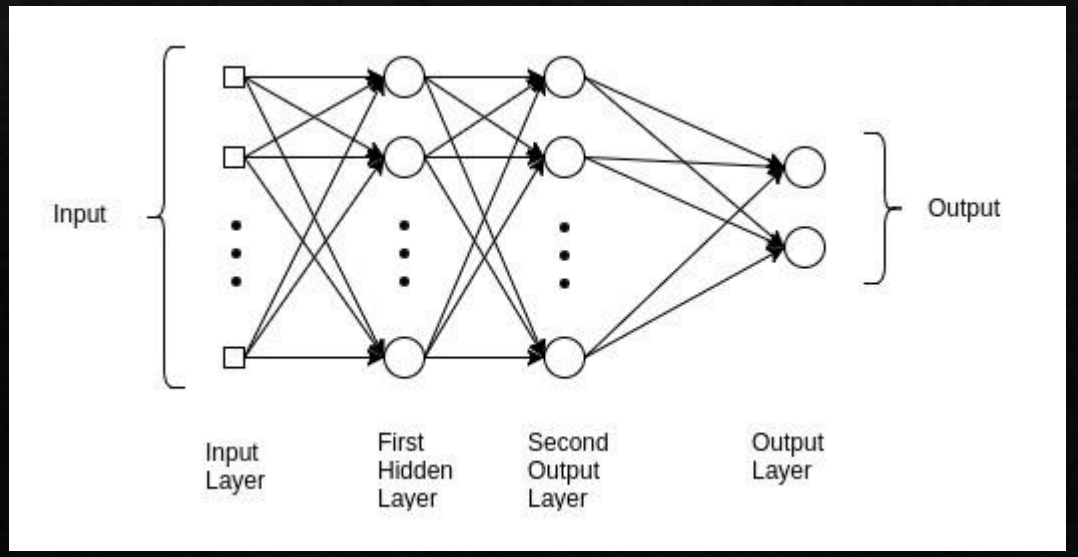
Neural Network Implementation – Multilayer Perceptron

- Multilayer Perceptron – a feed forward artificial neural network (ANN)
- Perceptron is a single Neuron (basis of larger neural networks)
- Used MultilayerPerceptron from Classifier in Weka
- Trained using train data with “use training set”
- Tested using “Supplied Test set”
- For Train Data
  - Non Seizure – all 5091 correctly predicted
  - Seizure – 1034 correct & 315 wrong
- For Test Data
  - Non Seizure – all 989 correctly predicted
  - Seizure – 225 correct & 74 wrong



Classification Algo. Implementation – Naive Bayes

- Naïve Bayes – is a commonly used Classification algorithm
- It Assumes all features are independent of each other
- Calculates posterior probabilities based on Bayes Theorem
- Trained and tested using “Naïve Bayes” from Classifier
- For Train Data
  - Non Seizure – 5016 correct & 113 wrong
  - Seizure – 1165 correct & 216 wrong
- For Test Data
  - Non Seizure – 1246 correct & 45 wrong
  - Seizure – 289 correct & 30 wrong



Comparison MLP and Naïve Bayes

TRAIN	Multilayer Perceptron	Naïve Bayes – Using training set	Naïve Bayes – 10-fold cross-validation	Baseline Model
Accuracy	95.10	95.97	95.96	79.95
Balanced Accuracy	88.32	93.82	93.78	50
Precision	95.4	96.0	96.0	81.4
Recall	95.1	96.0	96.0	80
AUC	86.8	96.2	96.2	50

TEST	Multilayer Perceptron	Naïve Bayes – Using supplied test set	Naïve Bayes – 10 fold cross validation
Accuracy	94.25	95.34	95.40
Balanced Accuracy	87.62	93.55	93.59
Precision	94.7	95.4	95.5
Recall	94.3	95.3	95.4
AUC	85.9	96	95.8

Principal Component Analysis

- Dimensionality Reduction Technique
- It tries to trade minimum accuracy for simplicity
- Used “Principal Component” as attribute evaluator and “Ranker” as search method
- We used the Variance Explanation at different levels and applied Naïve Bayes after PCA selected features.

Variance	Number of Principal Components	Number of Principal Components selected threshold -50 %	Accuracy (Naïve Bayes)	AUC (Naïve Bayes)	Time for building model
.95	38	25	95.24	95.3	.13 seconds
.90	32	25	95.31	98.8	.11 seconds
.85	28	26	96.49	99.2	0.5 seconds
.80	25	25	95.46	98.9	.05 seconds

Feature Selection – Cfs & Wrapper subset Evaluation

- To reduce computational cost and improve model performance
- CFS Subset Evaluation– Correlation based Feature Selection
  - Various Subsets are compared for High correlation with target and less intercorrelation
  - Best First Search method used
  - 178 to 57 attributes
- Wrapper Subset Evaluation – Bidirectional/Stepwise wrapping
  - It adds a feature on each step and removes if it performs bad before adding next and eventually gets the best performing subset
  - Greedy Stepwise search method used
  - 178 to 11 attributes

Feature Selection – Cfs & Wrapper subset Evaluation

TRAIN	Unprocessed Dataset Naïve Bayes	Cfs subset Evaluation, Naïve Bayes	wrapper subset Evaluation, Naïve Bayes	Baseline Model
Accuracy	95.97	96.13	96.51	79.95
Balanced Accuracy	93.82	93.95	93.95	50
Precision	96.0	96.10	96.50	80.50
Recall	96.0	96.10	96.50	80.50
AUC	96.2	98.10	98.80	50

TEST	Unprocessed Dataset	Cfs subset Evaluation Naïve Bayes	wrapper subset Evaluation Naïve Bayes
Accuracy	95.34	95.65	96.09
Balanced Accuracy	93.55	93.90	93.86
Precision	95.4	95.70	96.10
Recall	95.3	95.70	96.10
AUC	96	98.10	98.30

Training Time Comparison

		Time (in seconds)
Before Feature Selection (Unprocessed Dataset)	Baseline Model Zero R	0.04 seconds
	Multilayer Perceptron	1357.64 seconds
	Naïve Bayes-using training set	0.52 seconds
	Naïve Bayes-using cross validation	0.37 seconds
After Feature Selection	Cfs subset Evaluation-Naïve Bayes – using training data	0.08 seconds
	wrapper subset Evaluation Naïve Bayes – using training data	0.12 seconds

THANK YOU

By

Divya Das