

Getting started with Neo4j:

- <https://medium.com/larus-team/how-to-create-recommendation-engine-in-neo4j-7963e635c730>
- <https://neo4j.com/docs/operations-manual/current/installation/neo4j-browser/#:~:text=Neo4j%20Browser%20is%20a%20tool,Neo4j%20Server%20and%20Neo4j%20Desktop.>
- <https://github.com/graphaware/neo4j-nlp>
- <https://neo4j.com/developer/neo4j-desktop/>
- <https://search.maven.org/artifact/edu.stanford.nlp/stanford-corenlp/3.9.2/jar>
- <https://medium.com/neo4j/using-nlp-in-neo4j-ac40bc92196f>

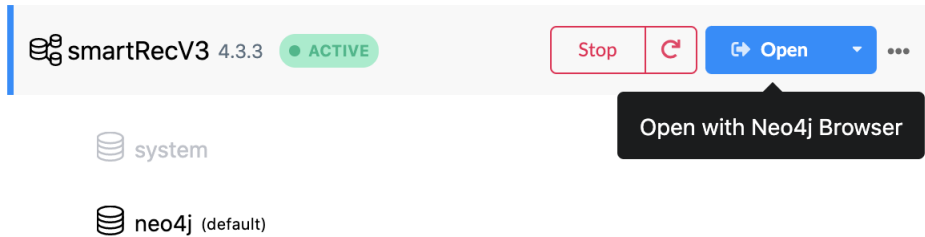
Airbnb Neo4j queries

There are 2 ways to load the graph data in Neo4j,

- 1) Using the Neo4j browser
- 2) Using Neo4j admin terminal (Bulk-upload)

Creating Database

The screenshot shows the Neo4j Desktop application interface. At the top, a dark header bar reads "Neo4j Desktop - 1.4.9". Below this, a light gray area contains the text "No active DBMS". The main workspace displays a dialog box titled "Example Project" with a blue "Add" button. The dialog has three sections: "Name" with a text input field containing "smartRecV3", "Password" with a masked input field (four dots), and "Version" with a dropdown menu set to "4.3.3". At the bottom of the dialog are "Cancel" and "Create" buttons. Below the dialog, a horizontal bar shows a list of projects. The first project is "smartRecV3 4.3.3", accompanied by a Neo4j logo icon. To the right of the project name are three buttons: "Start", "Open", and a three-dot menu icon.



Loading Airbnb domain knowledge Graph

1) Using the Neo4j browser

```
LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url, row[8] as picture_url, row[9] as
host_identity_verified, row[10] as accomodates, row[11] as bedrooms, row[12] as
bathrooms, row[13] as beds, row[14] as price, toFloat(row[15]) as score
WHERE labels=":User"
CREATE (:User{id:id, name: name});
```

```
LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url, row[8] as picture_url, row[9] as
host_identity_verified, row[10] as accomodates, row[11] as bedrooms, row[12] as
bathrooms, row[13] as beds, row[14] as price, toFloat(row[15]) as score
WHERE labels=":Amenities"
CREATE (:Amenities{id:id, name: name});
```

```
LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url, row[8] as picture_url, row[9] as
host_identity_verified, row[10] as accomodates, row[11] as bedrooms, row[12] as
bathrooms, row[13] as beds, row[14] as price, toFloat(row[15]) as score
WHERE labels=":property_type"
CREATE (:property_type{id:id, name: name});
```

```
LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url, row[8] as picture_url, row[9] as
```

```

host_identity_verified,row[10] as accomodates,row[11] as bedrooms,row[12] as
bathrooms,row[13] as beds,row[14] as price,toFloat(row[15]) as score
WHERE labels=":room_type"
CREATE (:room_type{id:id, name: name});

```

```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url,row[8] as picture_url,row[9] as
host_identity_verified,row[10] as accomodates,row[11] as bedrooms,row[12] as
bathrooms,row[13] as beds,row[14] as price,toFloat(row[15]) as score
WHERE labels=":City"
CREATE (:City{id:id, name: name});

```

```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url,row[8] as picture_url,row[9] as
host_identity_verified,row[10] as accomodates,row[11] as bedrooms,row[12] as
bathrooms,row[13] as beds,row[14] as price,toFloat(row[15]) as score
WHERE labels=":State"
CREATE (:State{id:id, name: name});

```

```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url,row[8] as picture_url,row[9] as
host_identity_verified,row[10] as accomodates,row[11] as bedrooms,row[12] as
bathrooms,row[13] as beds,row[14] as price,toFloat(row[15]) as score
WHERE labels=":Country"
CREATE (:Country{id:id, name: name});

```

```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url,row[8] as picture_url,row[9] as
host_identity_verified,row[10] as accomodates,row[11] as bedrooms,row[12] as
bathrooms,row[13] as beds,row[14] as price,toFloat(row[15]) as score
WHERE labels=":Listing"
CREATE (:Listing{id:id, name: name,
url:url,picture_url:picture_url,host_identity_verified:host_identity_verified,accomodates:
accomodates,bedrooms:bedrooms,bathrooms:bathrooms,beds:beds,price:price,score:score});

```

```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url, row[8] as picture_url, row[9] as
host_identity_verified, row[10] as accomodates, row[11] as bedrooms, row[12] as
bathrooms, row[13] as beds, row[14] as price, toFloat(row[15]) as score
WHERE type="HAS_AMENITY"
MATCH (l:Listing) WHERE l.id= start
MATCH (a:Amenities) WHERE a.id= end
CREATE (l)-[:HAS_AMENITY]->(a);

```

```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url, row[8] as picture_url, row[9] as
host_identity_verified, row[10] as accomodates, row[11] as bedrooms, row[12] as
bathrooms, row[13] as beds, row[14] as price, toFloat(row[15]) as score
WHERE type="HAS_PROPERTY_TYPE"
MATCH (l:Listing) WHERE l.id= start
MATCH (p:property_type) WHERE p.id= end
CREATE (l)-[:HAS_PROPERTY_TYPE]->(p);

```

```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url, row[8] as picture_url, row[9] as
host_identity_verified, row[10] as accomodates, row[11] as bedrooms, row[12] as
bathrooms, row[13] as beds, row[14] as price, toFloat(row[15]) as score
WHERE type="HAS_ROOM_TYPE"
MATCH (l:Listing) WHERE l.id= start
MATCH (r:room_type) WHERE r.id= end
CREATE (l)-[:HAS_ROOM_TYPE]->(r);

```

```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url, row[8] as picture_url, row[9] as
host_identity_verified, row[10] as accomodates, row[11] as bedrooms, row[12] as
bathrooms, row[13] as beds, row[14] as price, toFloat(row[15]) as score
WHERE type="IN_CITY"
MATCH (l:Listing) WHERE l.id= start
MATCH (c:City) WHERE c.id= end
CREATE (l)-[:IN_CITY]->(c);

```

```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url,row[8] as picture_url,row[9] as
host_identity_verified,row[10] as accomodates,row[11] as bedrooms,row[12] as
bathrooms,row[13] as beds,row[14] as price,toFloat(row[15]) as score
WHERE type="IN_COUNTRY"
MATCH (l:Listing) WHERE l.id= start
MATCH (c:Country) WHERE c.id= end
CREATE (l)-[:IN_COUNTRY]->(c);

```

```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url,row[8] as picture_url,row[9] as
host_identity_verified,row[10] as accomodates,row[11] as bedrooms,row[12] as
bathrooms,row[13] as beds,row[14] as price,toFloat(row[15]) as score
WHERE type="IN_STATE"
MATCH (l:Listing) WHERE l.id= start
MATCH (s:State) WHERE s.id= end
CREATE (l)-[:IN_STATE]->(s);

```

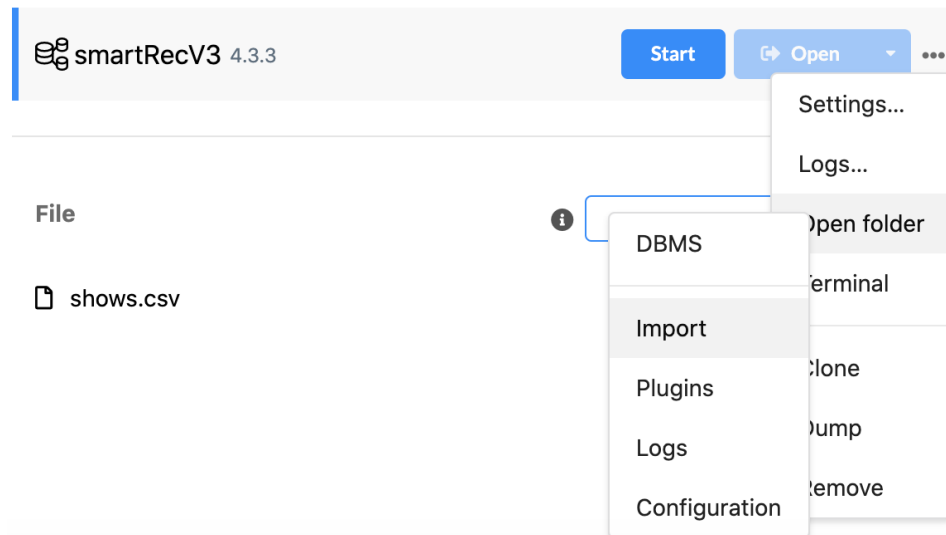
```

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end,
row[5] as type, toFloat(row[6]) as rating, row[7] as url,row[8] as picture_url,row[9] as
host_identity_verified,row[10] as accomodates,row[11] as bedrooms,row[12] as
bathrooms,row[13] as beds,row[14] as price,toFloat(row[15]) as score
WHERE type="RATED"
MATCH (u:User) WHERE u.id= start
MATCH (l:Listing) WHERE l.id= end
CREATE (u)-[:RATED{rating: rating}]->(l);

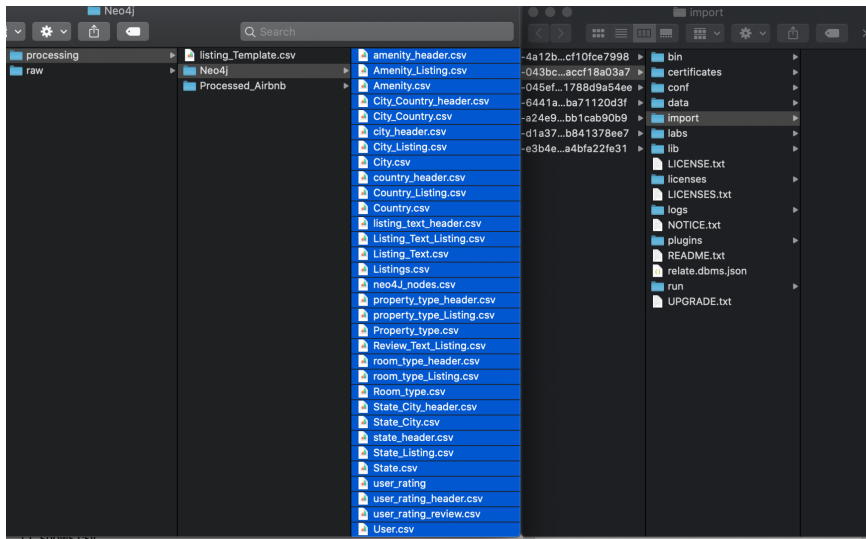
```

2) Using Neo4j admin terminal

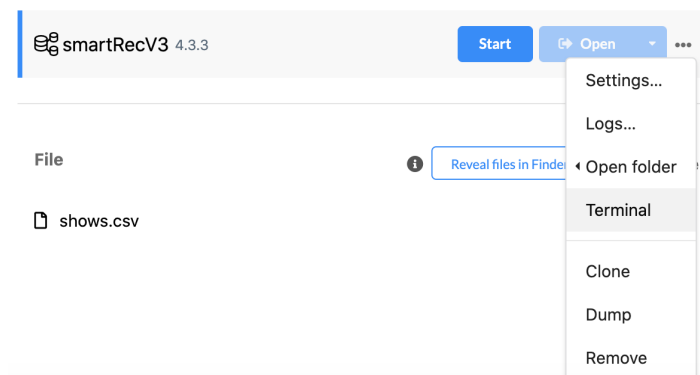
Importing files from project source into the database,



Copy the files from the preprocessed folder in the project source into the import folder,



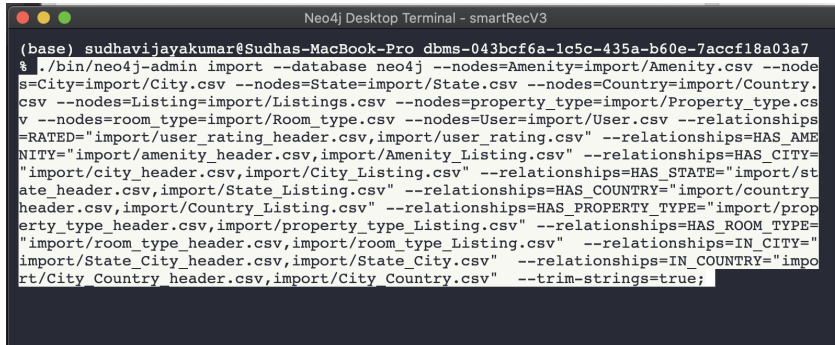
Execute Load Command, 1) open terminal 2) paste command



Command

```
./bin/neo4j-admin import --database neo4j --nodes=Amenity=import/Amenity.csv
--nodes=City=import/City.csv --nodes=State=import/State.csv
--nodes=Country=import/Country.csv --nodes=Listing=import/Listings.csv
--nodes=property_type=import/Property_type.csv
--nodes=room_type=import/Room_type.csv --nodes=User=import/User.csv
--relationships=RATED="import/user_rating_header.csv,import/user_rating.csv"
--relationships=HAS_AMENITY="import/amenity_header.csv,import/Amenity_Listing.csv"
--relationships=HAS_CITY="import/city_header.csv,import/City_Listing.csv"
--relationships=HAS_STATE="import/state_header.csv,import/State_Listing.csv"
--relationships=HAS_COUNTRY="import/country_header.csv,import/Country_Listing.csv"
```

```
sv"
--relationships=HAS_PROPERTY_TYPE="import/property_type_header.csv,import/property_type_Listing.csv"
--relationships=HAS_ROOM_TYPE="import/room_type_header.csv,import/room_type_Listing.csv"
--relationships=IN_CITY="import/State_City_header.csv,import/State_City.csv"
--relationships=IN_COUNTRY="import/City_Country_header.csv,import/City_Country.csv" --trim-strings=true;
```



```
(base) sudhaviyayakumar@Sudhas-MacBook-Pro dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7
% ./bin/neo4j-admin import --database neo4j --nodes=Amenity=import/Amenity.csv --nodes=City=import/City.csv --nodes=State=import/State.csv --nodes=Country=import/Country.csv --nodes=Listing=import/Listings.csv --nodes=property_type=import/Property_type.csv --nodes=room_type=import/Room_type.csv --nodes=User=import/User.csv --relationships=RATED=import/user_rating_header.csv,import/user_rating.csv --relationships=HAS_AMENITY=import/amenity_header.csv,import/Amenity_Listing.csv --relationships=HAS_CITY=import/city_header.csv,import/City_Listing.csv --relationships=HAS_STATE=import/state_header.csv,import/State_Listing.csv --relationships=HAS_COUNTRY=import/country_header.csv,import/Country_Listing.csv --relationships=HAS_PROPERTY_TYPE=import/property_type_header.csv,import/property_type_Listing.csv --relationships=HAS_ROOM_TYPE=import/room_type_header.csv,import/room_type_Listing.csv --relationships=IN_CITY=import/State_City_header.csv,import/State_City.csv --relationships=IN_COUNTRY=import/City_Country_header.csv,import/City_Country.csv --trim-strings=true;
```

Execution log

```
(base) sudhaviyayakumar@Sudhas-MacBook-Pro dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7
% ./bin/neo4j-admin import --database neo4j --nodes=Amenity=import/Amenity.csv
--nodes=City=import/City.csv --nodes=State=import/State.csv
--nodes=Country=import/Country.csv --nodes=Listing=import/Listings.csv
--nodes=property_type=import/Property_type.csv --nodes=room_type=import/Room_type.csv
--nodes=User=import/User.csv
--relationships=RATED="import/user_rating_header.csv,import/user_rating.csv"
--relationships=HAS_AMENITY="import/amenity_header.csv,import/Amenity_Listing.csv"
--relationships=HAS_CITY="import/city_header.csv,import/City_Listing.csv"
--relationships=HAS_STATE="import/state_header.csv,import/State_Listing.csv"
--relationships=HAS_COUNTRY="import/country_header.csv,import/Country_Listing.csv"
--relationships=HAS_PROPERTY_TYPE="import/property_type_header.csv,import/property_type_Listing.csv"
--relationships=HAS_ROOM_TYPE="import/room_type_header.csv,import/room_type_Listing.csv"
--relationships=IN_CITY="import/State_City_header.csv,import/State_City.csv"
--relationships=IN_COUNTRY="import/City_Country_header.csv,import/City_Country.csv"
--trim-strings=true;
```

Selecting JVM - Version:11.0.8, Name:OpenJDK 64-Bit Server VM, Vendor:Azul Systems, Inc.

Neo4j version: 4.3.3

Importing the contents of these files into /Users/sudhavijayakumar/Library/Application Support/Neo4j Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/data/databases/neo4j:

Nodes:

[Amenity]:

/Users/sudhavijayakumar/Library/Application Support/Neo4j Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/Amenity.csv

[User]:

/Users/sudhavijayakumar/Library/Application Support/Neo4j Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/User.csv

[Listing]:

/Users/sudhavijayakumar/Library/Application Support/Neo4j Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/Listings.csv

[State]:

/Users/sudhavijayakumar/Library/Application Support/Neo4j Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/State.csv

[property_type]:

/Users/sudhavijayakumar/Library/Application Support/Neo4j Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/Property_type.csv

[Country]:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/C
ountry.csv

[City]:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/Ci
ty.csv

[room_type]:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/R
oom_type.csv

Relationships:

HAS_PROPERTY_TYPE:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/pr
operty_type_header.csv

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/pr
operty_type_Listing.csv

IN_COUNTRY:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/Ci
ty_Country_header.csv

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/Ci
ty_Country.csv

HAS_CITY:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/city_header.csv

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/City_Listing.csv

HAS_AMENITY:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/amenity_header.csv

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/Amenity_Listing.csv

RATED:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/user_rating_header.csv

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/user_rating.csv

HAS_STATE:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/state_header.csv

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/State_Listing.csv

HAS_COUNTRY:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/co
untry_header.csv

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/C
ountry_Listing.csv

IN_CITY:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/St
ate_City_header.csv

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/St
ate_City.csv

HAS_ROOM_TYPE:

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/ro
om_type_header.csv

/Users/sudhavijayakumar/Library/Application Support/Neo4j
Desktop/Application/relate-data/dbmss/dbms-043bcf6a-1c5c-435a-b60e-7accf18a03a7/import/ro
om_type_Listing.csv

Available resources:

Total machine memory: 16.00GiB

Free machine memory: 1.312GiB

Max heap memory : 1.000GiB

Processors: 8

Configured max memory: 13.50GiB

High-IO: true

WARNING: heap size 1.000GiB may be too small to complete this import. Suggested heap size is 1.000GiB

Import starting 2022-03-16 22:34:08.978-0700

Estimated number of nodes: 250.16 k

Estimated number of node properties: 547.88 k

Estimated number of relationships: 419.28 k

Estimated number of relationship properties: 247.72 k

Estimated disk space usage: 39.01MiB

Estimated required memory usage: 1023MiB

(1/4) Node import 2022-03-16 22:34:10.162-0700

Estimated number of nodes: 250.16 k

Estimated disk space usage: 15.73MiB

Estimated required memory usage: 1023MiB

.....	5%	Δ625ms
.....	10%	Δ1ms
.....-	15%	Δ119ms
.....	20%	Δ1ms
.....	25%	Δ200ms
.....	30%	Δ1ms
.....	35%	Δ0ms
.....	40%	Δ1ms
.....	45%	Δ0ms

..... 50% Δ1ms
..... 55% Δ0ms
..... 60% Δ0ms
..... 65% Δ4ms
..... 70% Δ0ms
..... 75% Δ0ms
..... 80% Δ0ms
..... 85% Δ1ms
..... 90% Δ0ms
..... 95% Δ0ms
..... 100% Δ0ms

Node import COMPLETED in 1s 36ms

(2/4) Relationship import 2022-03-16 22:34:11.199-0700

Estimated number of relationships: 419.28 k

Estimated disk space usage: 23.28MiB

Estimated required memory usage: 1.004GiB

..... 5% Δ397ms
..... 10% Δ1s 65ms
..... 15% Δ1ms
..... 20% Δ0ms
..... 25% Δ0ms
..... 30% Δ1ms
..... 35% Δ0ms
..... 40% Δ0ms

..... 45% Δ0ms
..... 50% Δ1ms
..... 55% Δ0ms
..... 60% Δ0ms
..... 65% Δ1ms
..... 70% Δ0ms
..... 75% Δ0ms
..... 80% Δ0ms
..... 85% Δ1ms
..... 90% Δ0ms
..... 95% Δ0ms
..... 100% Δ1ms

Relationship import COMPLETED in 1s 468ms

(3/4) Relationship linking 2022-03-16 22:34:12.667-0700

Estimated required memory usage: 1023MiB

-..... 5% Δ866ms
..... 10% Δ1ms
..... 15% Δ0ms
..... 20% Δ0ms
..... 25% Δ1ms
..... 30% Δ0ms
..... 35% Δ1ms
..... 40% Δ0ms
..... 45% Δ0ms

..... 50% Δ1ms
..... 55% Δ0ms
..... 60% Δ0ms
..... 65% Δ1ms
..... 70% Δ0ms
..... 75% Δ0ms
..... 80% Δ1ms
..... 85% Δ0ms
..... 90% Δ0ms
..... 95% Δ1ms
..... 100% Δ0ms

Relationship linking COMPLETED in 1s 417ms

(4/4) Post processing 2022-03-16 22:34:14.084-0700

Estimated required memory usage: 1020MiB

..-..... 5% Δ1s 23ms
..... 10% Δ0ms
..... 15% Δ1ms
..... 20% Δ0ms
..... 25% Δ0ms
..... 30% Δ1ms
..... 35% Δ0ms
..... 40% Δ1ms
..... 45% Δ0ms
..... 50% Δ0ms

..... 55% Δ1ms
..... 60% Δ0ms
..... 65% Δ1ms
..... 70% Δ0ms
..... 75% Δ0ms
..... 80% Δ0ms
..... 85% Δ1ms
..... 90% Δ0ms
..... 95% Δ0ms
..... 100% Δ1ms

Post processing COMPLETED in 2s 576ms

IMPORT DONE in 8s 308ms.

Imported:

247376 nodes

418817 relationships


789574 properties

Peak memory usage: 1.035GiB


Merging ConceptNet5 into the Airbnb domain knowledge Graph

Create Database

Name

 NLPv3

Password



Version

3.5.17

✕

Cancel

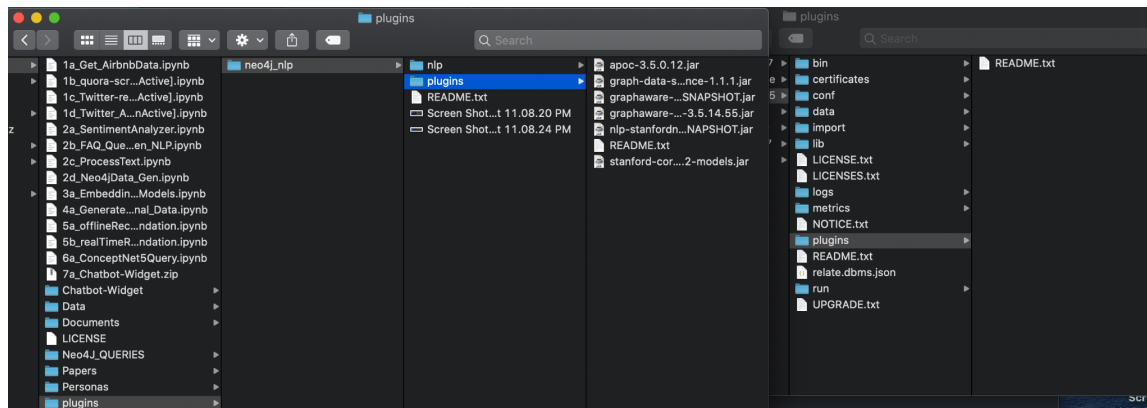
✓ Create

Load Data

Repeat steps: Using the Neo4j browser

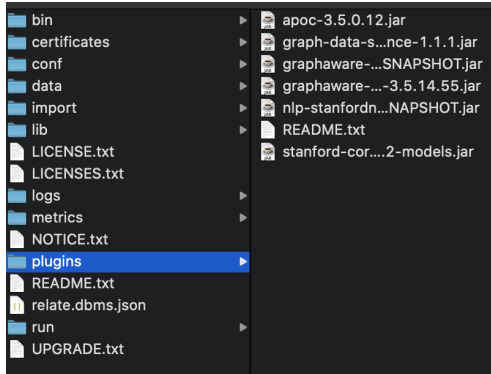
Import plugins :

Copy from project source to the Neo4j plugin folder as shown below,

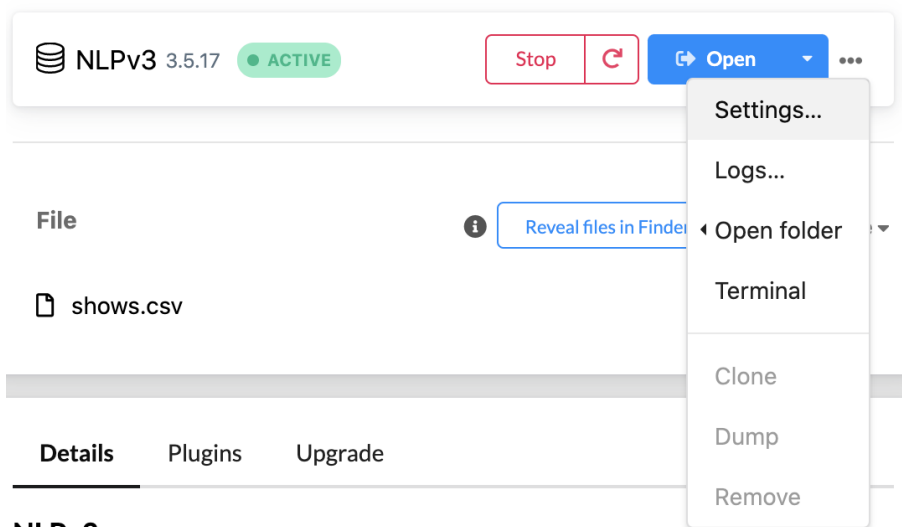


Download the below plugin and copy it into the neo4j import folder,

<https://repo1.maven.org/maven2/edu/stanford/nlp/stanford-corenlp/3.9.2/stanford-corenlp-3.9.2-models.jar>



Add the below configurations into the Settings: NLP



NLPv2

```
dbms.memory.heap.initial_size=3000m
dbms.memory.heap.max_size=5000m
dbms.unmanaged_extension_classes=com.graphaware.server=/graphaware
com.graphaware.runtime.enabled=true
com.graphaware.module.NLP.1=com.graphaware.nlp.module.NLPBootstrapper
dbms.security.procedures.whitelist=ga.nlp.*, apoc.*, gds.*
```

Restart the database

Execute the below command from the Neo4j browser,

```
LOAD CSV FROM 'file:///neo4j_nodes.csv' AS row
```

```

WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end, row[5]
as type, toFloat(row[6]) as rating, row[7] as url, row[8] as picture_url, row[9] as
host_identity_verified, row[10] as accomodates, row[11] as bedrooms, row[12] as
bathrooms, row[13] as beds, row[14] as price, toFloat(row[15]) as score
WHERE labels=":Listing_Text"
CREATE (:Listing_Text{id:id, name: name});

LOAD CSV FROM 'file:///neo4J_nodes.csv' AS row
WITH row, row[0] as id, row[1] as labels, row[2] as name, row[3] as start, row[4] as end, row[5]
as type, toFloat(row[6]) as rating, row[7] as url, row[8] as picture_url, row[9] as
host_identity_verified, row[10] as accomodates, row[11] as bedrooms, row[12] as
bathrooms, row[13] as beds, row[14] as price, toFloat(row[15]) as score
WHERE labels=":Review_Text"
CREATE (:Review_Text{id:id, name: name});

```

```

LOAD CSV FROM 'file:///Listing_Text_Listing.csv' AS row
WITH row, row[0] as end, row[1] as start
MATCH (l:Listing) WHERE l.id= start
MATCH (lt:Listing_Text) WHERE lt.id= end
CREATE (l)-[:HAS_TEXT]->(lt);


```

```

LOAD CSV FROM 'file:///Review_Text_Listing.csv' AS row
WITH row, row[0] as end, row[1] as start
MATCH (l:Listing) WHERE l.id= start
MATCH (lt:Review_Text) WHERE lt.id= end
CREATE (l)-[:HAS_REVIEW]->(lt);

```

Results

```
$ LOAD CSV FROM 'file:///Listing_Text_Listing.csv' AS row WITH row, row[0] as en... 
```

SUCCESS

Created 5402 relationships, completed after 27364 ms.

```
$ LOAD CSV FROM 'file:///Review_Text_Listing.csv' AS row WITH row, row[0] as end... ✓
```

SUCCESS

Created 5402 relationships, completed after 26380 ms.

in-DB NLP Cypher Query

Number-batch download link: <https://conceptnet.s3.amazonaws.com/>

```
CALL ga.nlp.createSchema();CALL ga.nlp.config.setDefaultLanguage('en');
```

```
CALL ga.nlp.processor.addPipeline({textProcessor:  
'com.graphaware.nlp.processor.stanford.StanfordTextProcessor', name: 'semanticNLP',  
processingSteps: {tokenize: true, ner: true, dependency: false}, stopWords: '+,result, all, during',  
threadNumber: 20});
```

```
CALL  
ga.nlp.ml.word2vec.addModel('/Users/sudhavijayakumar/Desktop/299/299A-SMARTRec/plugins/  
neo4j_nlp/nlp',  
'/Users/sudhavijayakumar/Desktop/299/299A-SMARTRec/plugins/neo4j_nlp/nlp/numberbatch.t  
xt', 'en-ConceptNet5');
```

```
CALL ga.nlp.ml.word2vec.load('en-ConceptNet5');
```

```
CALL apoc.periodic.iterate(  
"MATCH (n:Review_Text) RETURN n",  
"CALL ga.nlp.annotate({pipeline:'semanticNLP',text: n.name, id: id(n)})",  
YIELD result MERGE (n)-[:HAS_ANNOTATED_TEXT]->(result)", {batchSize:1,  
iterateList:true});
```

```
CALL ga.nlp.ml.word2vec.attach({query:'MATCH (t:Tag) RETURN t',  
modelName:'en-ConceptNet'});
```

```
$ LOAD CSV FROM 'file:///Listing_Text_Listing.csv' AS row WITH row, row[0] as en... ✓
```

SUCCESS

Created 5402 relationships, completed after 27364 ms.

```
$ LOAD CSV FROM 'file:///Review_Text_Listing.csv' AS row WITH row, row[0] as end... ✓
```

SUCCESS

Created 5402 relationships, completed after 26380 ms.

```
$ CALL ga.nlp.ml.word2vec.addModel('/Users/sudhaviyakumar/Desktop/299/299A-SMARTRec/plugins/neo4j_nlp/nlp', '/Users/sudhaviyakumar/Desktop/299/299A-SMARTRec/plugins/neo4j_nlp/nlp/numberbatch.txt', 'en-ConceptNet5');
```

result

1

"SUCCESS"

Started streaming 1 records after 366 ms and completed after 366 ms.

Load conceptNet5 in-memory

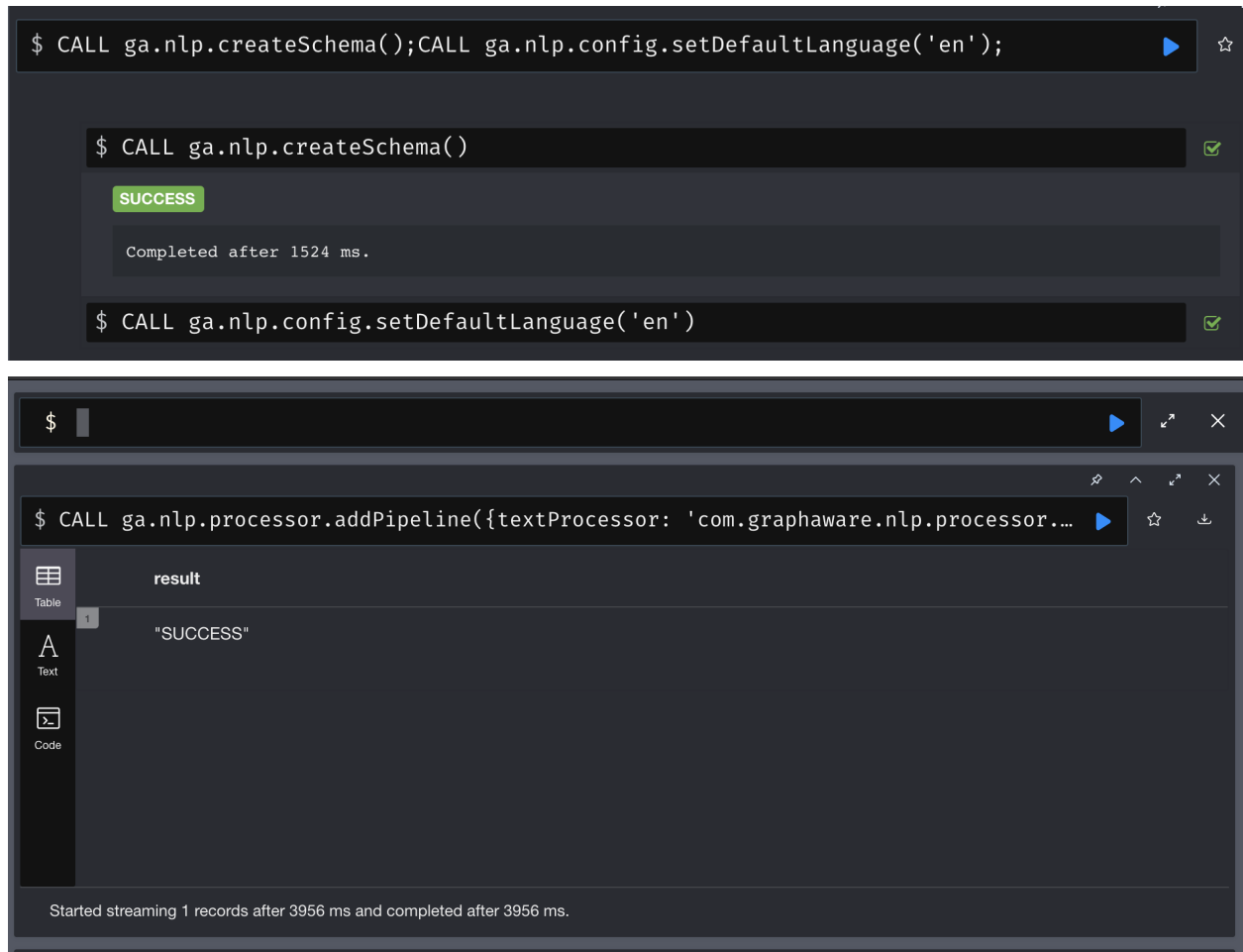
```
1 CALL ga.nlp.ml.word2vec.load('en-ConceptNet5');  
2
```

result

1

"SUCCESS"

Started streaming 1 records after 13527 ms and completed after 13527 ms.



Sample Real-time Recommendation(Content-based filtering)

```
MATCH(u:User{name:'10952'})-[:RATED]-(s:Listing)-[:HAS_AMENITY]-(c:Amenity)-[:HAS_AMENITY]-(z:Listing)
WHERE NOT EXISTS ((u)-[:RATED]-(z))
WITH s, z, COUNT(c) AS intersection
MATCH (s)-[:HAS_AMENITY]-(sc:Amenity)
WITH s, z, intersection, COLLECT(sc.name) AS s1
MATCH (z)-[:HAS_AMENITY]-(zc:Amenity)
WITH s, z, s1, intersection, COLLECT(zc.name) AS s2
WITH s, z, intersection, s1+[x IN s2 WHERE NOT x IN s1] AS union, s1, s2
RETURN s.name as UserListing, z.name as Recommendate, s1 as UserListingAmenities, s2 as RecommendateListingAmenities, ((1.0*intersection)/SIZE(union)) AS jaccard ORDER BY jaccard DESC;
```

Results:

	UserListing	Recommendate	UserListingAmenities
Text	"2818"	"4052965"	["Ethernet connection", "Dedicated workspace: table", "Pour-over coffee", "Security cameras on property"]
Code	"2818"	"23626417"	["Ethernet connection", "Dedicated workspace: table", "Pour-over coffee", "Security cameras on property"]
	"2818"	"10287267"	["Ethernet connection", "Dedicated workspace: table", "Pour-over coffee", "Security cameras on property"]
	"2818"	"16235797"	["Ethernet connection", "Dedicated workspace: table", "Pour-over coffee", "Security cameras on property"]
	"2818"	"818085"	["Ethernet connection", "Dedicated workspace: table", "Pour-over coffee", "Security cameras on property"]
	"2818"	"51631137"	["Ethernet connection", "Dedicated workspace: table", "Pour-over coffee", "Security cameras on property"]

Started streaming 5395 records after 25 ms and completed after 31 ms, displaying first 1000 rows.

Sample Real-time Recommendation(Collaborative filtering)

```
MATCH (p1:User {name:"10952"})-[x:RATED]-(m:Listing)-[y:RATED]-(p2:User) WITH
COUNT(m) AS numbershows, SUM(x.rating * y.rating) AS
xyDotProduct,SQRT(REDUCE(xDot = 0.0, a IN COLLECT(x.rating) | xDot + a^2)) AS
xLength,SQRT(REDUCE(yDot = 0.0, b IN COLLECT(y.rating) | yDot + b^2)) AS yLength,p1,
p2 RETURN p1.name as SelectedUser, p2.name as SimilarUser, xyDotProduct / (xLength *
yLength) AS sim ORDER BY sim DESC;
```


Results:

	SelectedUser	SimilarUser
33	"10952"	"160791963"
34	"10952"	"535440"
35	"10952"	"216099941"
36	"10952"	"224678257"
37	"10952"	"138914196"
38	"10952"	"7880798"
ted streaming 249 records after 18 ms and completed after 114 ms.		

Sample real-time conceptual recommendation,

```
CALL ga.nlp.ml.word2vec.nn('refrigerator', 10, 'en-ConceptNet5') YIELD word, distance;

MATCH (t:Tag)-[]-(s:Sentence)-[]-(a:AnnotatedText)-[]-(l:Review_Text)-[]-(l:Listing) WHERE
t.value in ['refrigerator','minirefrigerator']
RETURN l.url LIMIT 5;
```

Results:

\$ CALL ga.nlp.ml.word2vec.nn('refrigerator', 10, 'en-ConceptNet5') Y...

☆⌵

Table

Text

Code

	word	distance
1	"refrigerator"	0.9999999982500009
2	"minirefrigerator"	0.9862330179574896
3	"keep_food_cold"	0.9847467539328789
4	"keeping_food_cold"	0.9838407126039007
5	"electric_refrigerator"	0.9715307088214176
6	"ice_safe"	0.9549220313346745
7		

Started streaming 10 records after 11723 ms and completed after 11725 ms.

```
1 MATCH (t:Tag )-[]-(s:Sentence)-[]-( :AnnotatedText)-[]-(lt:Review_Text)-[]-(l:Listing) WHERE t.value in ['refrigerator','minirefrigerator']
2 RETURN l.url LIMIT 5;
```

Table

l.url

Text

Code

1	"https://www.airbnb.com/rooms/20168"
2	"https://www.airbnb.com/rooms/20168"
3	"https://www.airbnb.com/rooms/20168"
4	"https://www.airbnb.com/rooms/20168"
5	"https://www.airbnb.com/rooms/20168"

Started streaming 5 records after 16 ms and completed after 24 ms.

```
1 MATCH (t:Tag )-[]-(s:Sentence)-[]-(a:AnnotatedText)-[]-(lt:Review_Text)-[]-(l:Listing) WHERE t.value in ['refrigerator','minirefrigerator']
2 RETURN l.url LIMIT 5;
```

Table

	l.url
1	"https://www.airbnb.com/rooms/20168"
2	"https://www.airbnb.com/rooms/20168"
3	"https://www.airbnb.com/rooms/20168"
4	"https://www.airbnb.com/rooms/20168"
5	"https://www.airbnb.com/rooms/20168"

Code

Started streaming 5 records after 16 ms and completed after 24 ms.

```
$ CALL ga.nlp.ml.word2vec.nn('refrigerator', 10, 'en-ConceptNet5') Y...
```

Table

	word	distance
1	"refrigerator"	0.9999999982500009
2	"minirefrigerator"	0.9862330179574896
3	"keep_food_cold"	0.9847467539328789
4	"keeping_food_cold"	0.9838407126039007
5	"electric_refrigerator"	0.9715307088214176
6	"ice_safe"	0.9549220313346745
7		

Code

Started streaming 10 records after 11723 ms and completed after 11725 ms.