# Data extraction from Wine Reviews

Project by Rohit Bewoor

# Agenda

- Overview
- Data chosen
- Approach
- User Interface – Screenshots
  - Graph after initial loading without new user input
  - New data input
  - Querying the database
- Dockerizing the application
- Further improvements

# Overview

- Process unstructured data and graph a network
  - Allow query running via graphical user interface (GUI)
  - Dockerize application

- Created two python scripts:
  - Extract data from original source into text files (01_create_data_1.py)
  - Process the text files to load graph and show GUI to add more data or run queries (02_load_neo_show_gui_3.py)

# Dataset used

- As unstructured data decided to use wine reviews as I have already worked in the alco-bev industry earlier:
  - Searched for “wine reviews data” and found kaggle data
  - Link: https://www.kaggle.com/zynicide/wine-reviews
- About the data:
  - CSV file with 130k rows and 14 columns
  - Used only “Description column” as unstructured data

# Data Extraction

- Functionality of script 1: 01_create_data_1.py
  - Load user specified amount of rows from CSV file to Pandas (run time parameter - csvRowsLimit)
  - Content of “Description” cell written to individual text file
    - Files automatically names as fxxxx.txt, where ‘xxxx’ is from 0001 onwards
    - All files except last 5 written to folder: “inData”, last 5 files to folder called “extraUserInput”
- Running the script – example:

  -
    ```
    python3 01_create_data_1.py -wineFileLoc './winemag-data-130k-v2.csv' -csvRowsLimit 1000
    ```

# Data Extraction

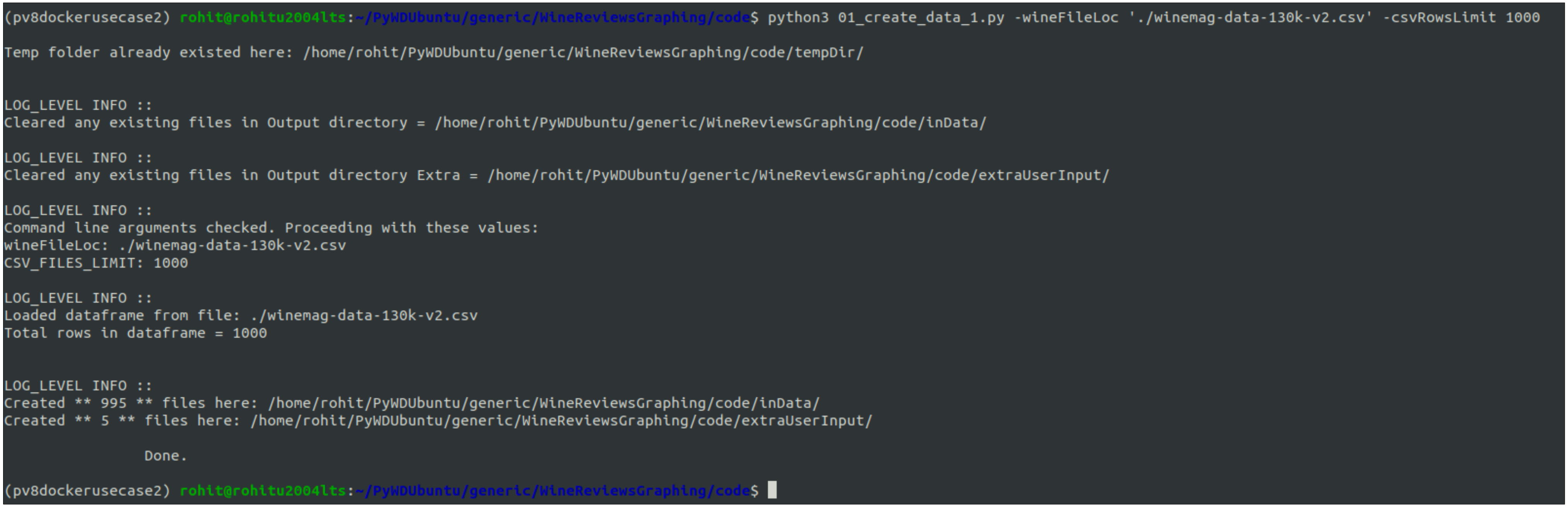
- Contents of some random file (f0007.txt):
  - *Here's a bright, informal red that opens with aromas of candied berry, white pepper and savory herb that carry over to the palate. It's balanced with fresh acidity and soft tannins.*
- This was the content of some row in the description column of the kaggle dataset.

# Data Extraction

Ran script with processing first 1000 rows of input CSV file.

995 individual files created in “inData” folder and last 5 in the “extraUserInput” folder.

Console output:



```
(pv8dockerusecase2) rohit@rohitu2004lts:~/PyWDUbuntu/generic/WineReviewsGraphing/code$ python3 01_create_data_1.py -wineFileLoc './winemag-data-130k-v2.csv' -csvRowsLimit 1000

Temp folder already existed here: /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/tempDir/


LOG_LEVEL INFO ::
Cleared any existing files in Output directory = /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/inData/

LOG_LEVEL INFO ::
Cleared any existing files in Output directory Extra = /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/extraUserInput/

LOG_LEVEL INFO ::
Command line arguments checked. Proceeding with these values:
wineFileLoc: ./winemag-data-130k-v2.csv
CSV_FILES_LIMIT: 1000

LOG_LEVEL INFO ::
Loaded dataframe from file: ./winemag-data-130k-v2.csv
Total rows in dataframe = 1000


LOG_LEVEL INFO ::
Created ** 995 ** files here: /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/inData/
Created ** 5 ** files here: /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/extraUserInput/

                Done.

(pv8dockerusecase2) rohit@rohitu2004lts:~/PyWDUbuntu/generic/WineReviewsGraphing/code$
```

# Approach

- Performed data extraction (covered earlier)
- Feature extraction with Spacy (version 3.1.1) large model
- Saved features in custom data structure to json file
- Features Extracted:
  - Word count, sentence count, sentiment score
  - Raw text from description
  - Processed text post: Lemmatization, stop-words and punctuations removal
- Named-entity-recognition (NER) extraction

# Approach

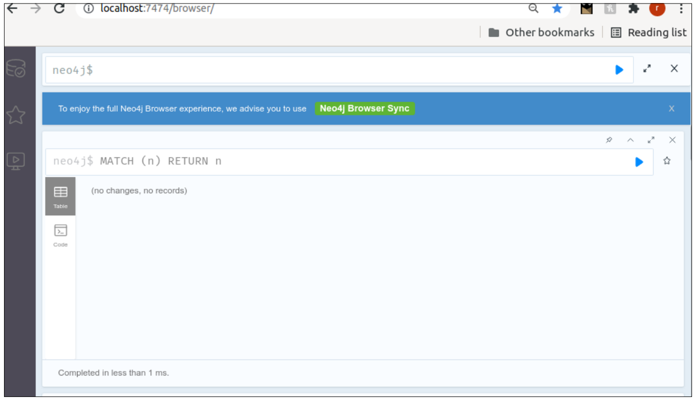
- Data insertion to Neo4j graph using the intermediate json file
  - Graph is cleared and reloaded with fresh data using the user specified number of files
  - Specified using runtime parameters: RELOAD_TO_NEO and LIMIT_UPLOAD_TO_NEO
- GUI implemented with Tkinter

# Approach

- Notes on GUI
  - 3 pre-set queries with user-specified input parameters
    - Query 1: Find count of nodes of a certain Label type
    - Query 2: Find count of Review type nodes whose raw text is longer than minimum specified word count, & sentiment score is greater than minimum specified score
    - Query 3: Show Review nodes “that have flavors” as specified
  - Adding new data to Neo4j possible in two ways:
    - Specify a file path with data in that file
    - Free form typed input as description

# Loading data to graph

Empty graph before initial insertion



```
neo4j$
```

To enjoy the full Neo4j Browser experience, we advise you to use Neo4j Browser Sync

```
neo4j$ MATCH (n) RETURN n
```

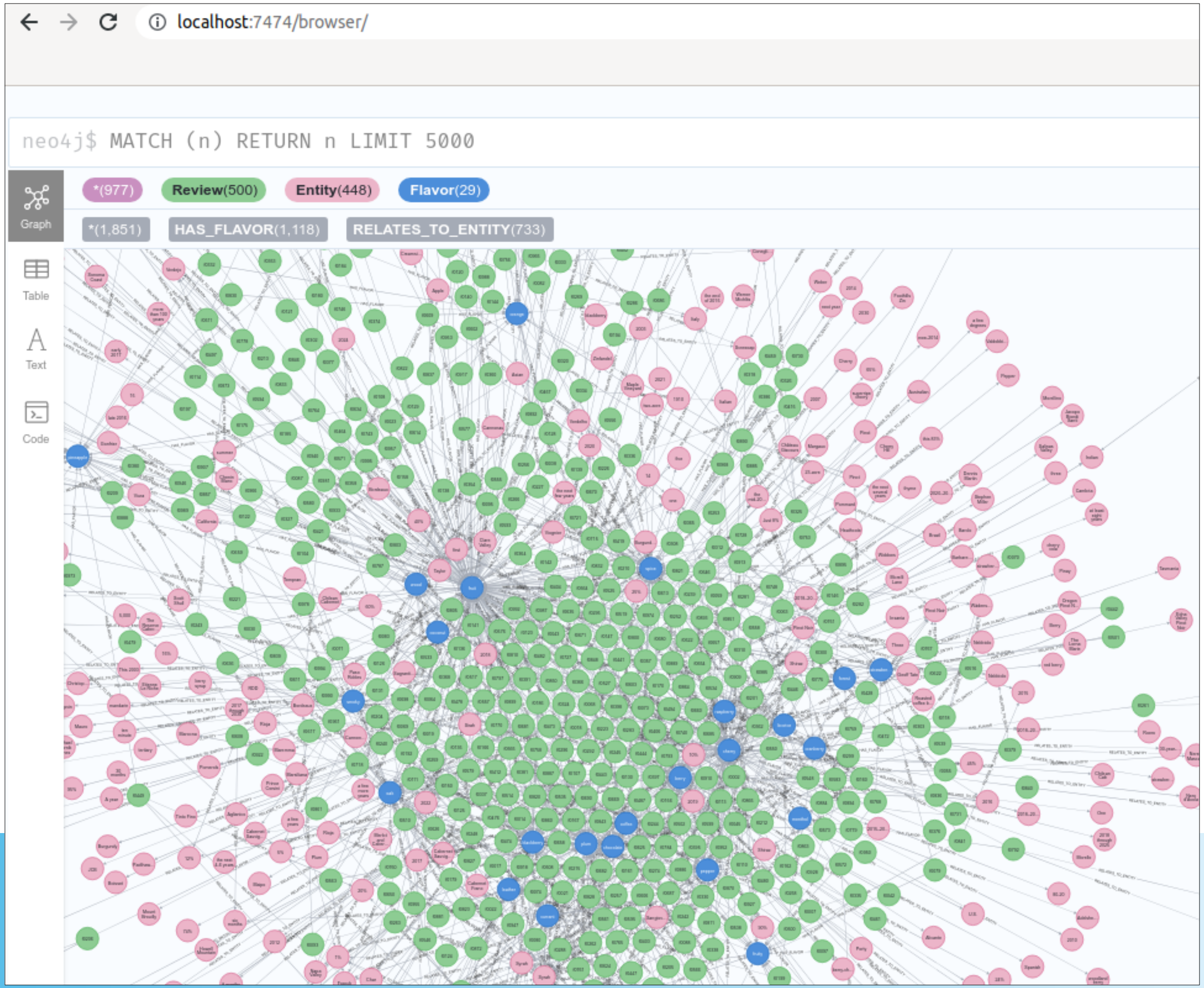
(no changes, no records)

Completed in less than 1 ms.

Neo4j graph after initial insertion from 500 files.

Nodes: Review=500, Entity=448, Flavor=29, Total=977

Relationships: HAS_FLAVOR=1118, RELATES_TO_ENTITY=733, Total=1851.



```
neo4j$ MATCH (n) RETURN n LIMIT 5000
```

Ran script:
02_read_process_for_neo_3.py

with RELOAD_TO_NEO = True

LIMIT_UPLOAD_TO_NEO = 500

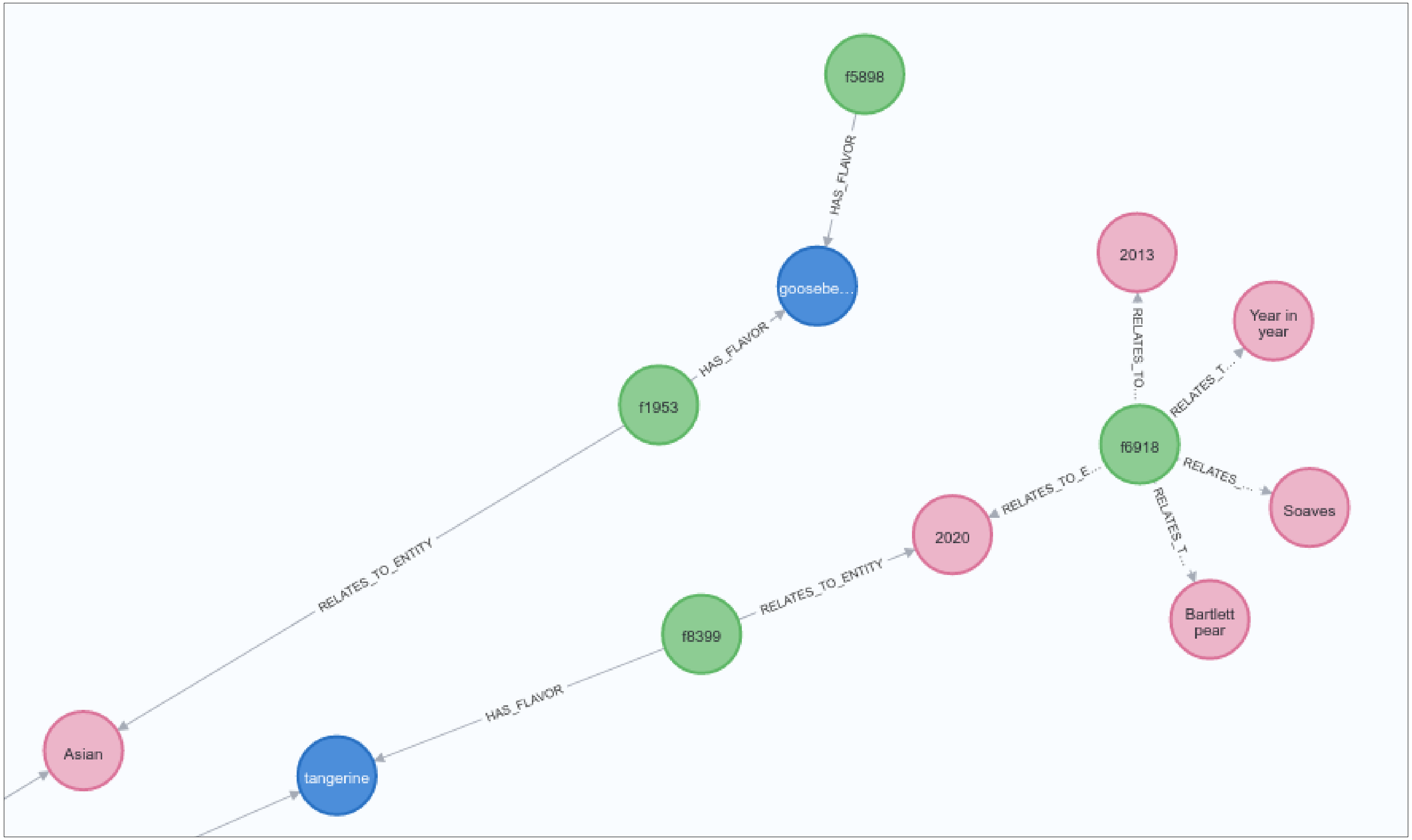
# Graph schema

Neo4j graph nodes and relationships:

- ✔ (REVIEW Node) - HAS_FLAVOR -> (FLAVOR node)
- ✔ (REVIEW Node) - RELATES_TO_ENTITY -> (ENTITY node)

Properties of Graph:

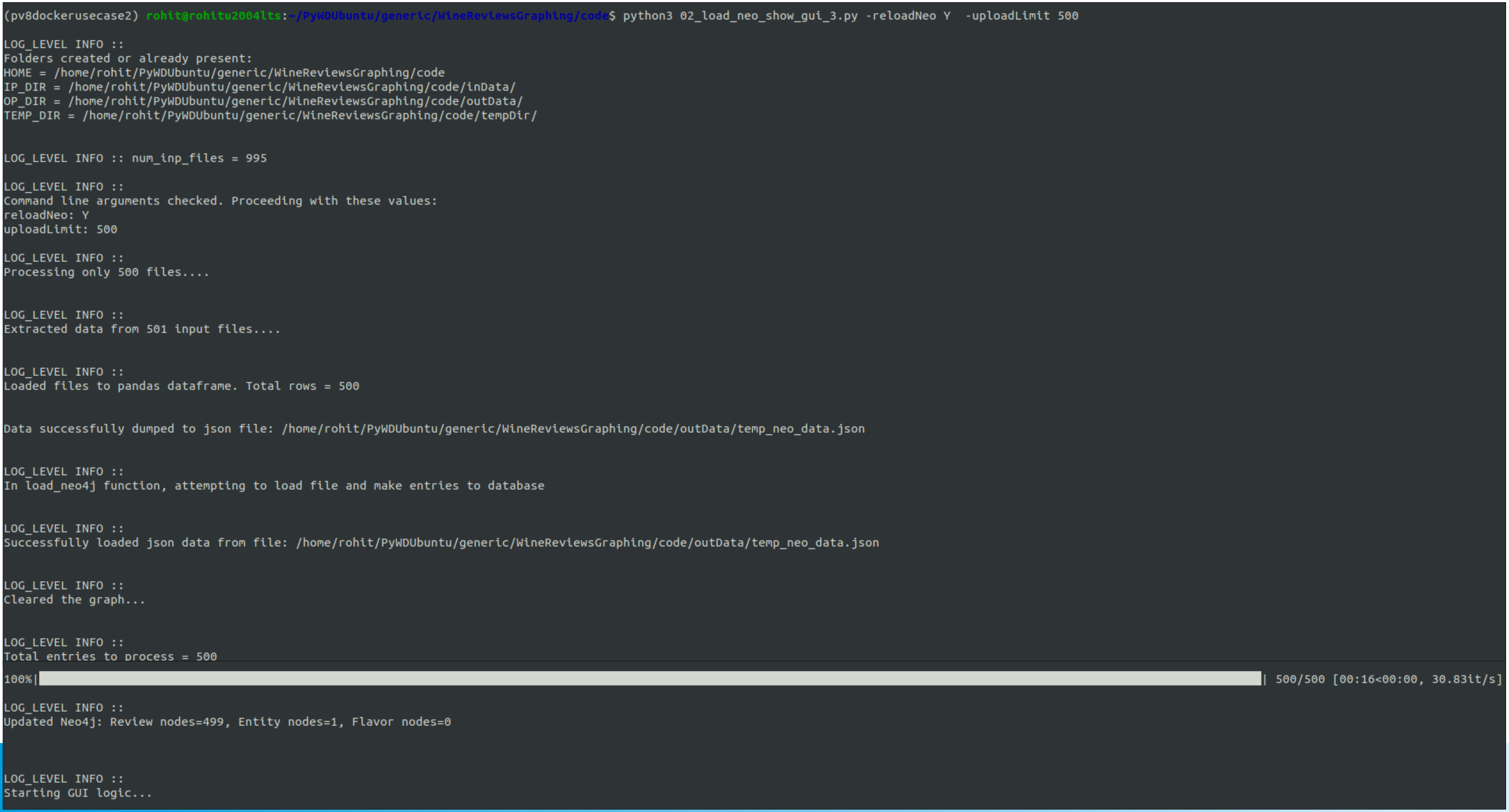
- Review Node (*green*): filename, sentiment score, word count, sentence count
- Entity Node (*pink*): text, label code, label name. E.g. name=2020, label=391, label_=DATE
- Flavor Node (*blue*): name. E.g. name=cherry



# Loading data – console output

Neo4j graph after initial insertion from 1000 files, waiting for user input in the GUI:

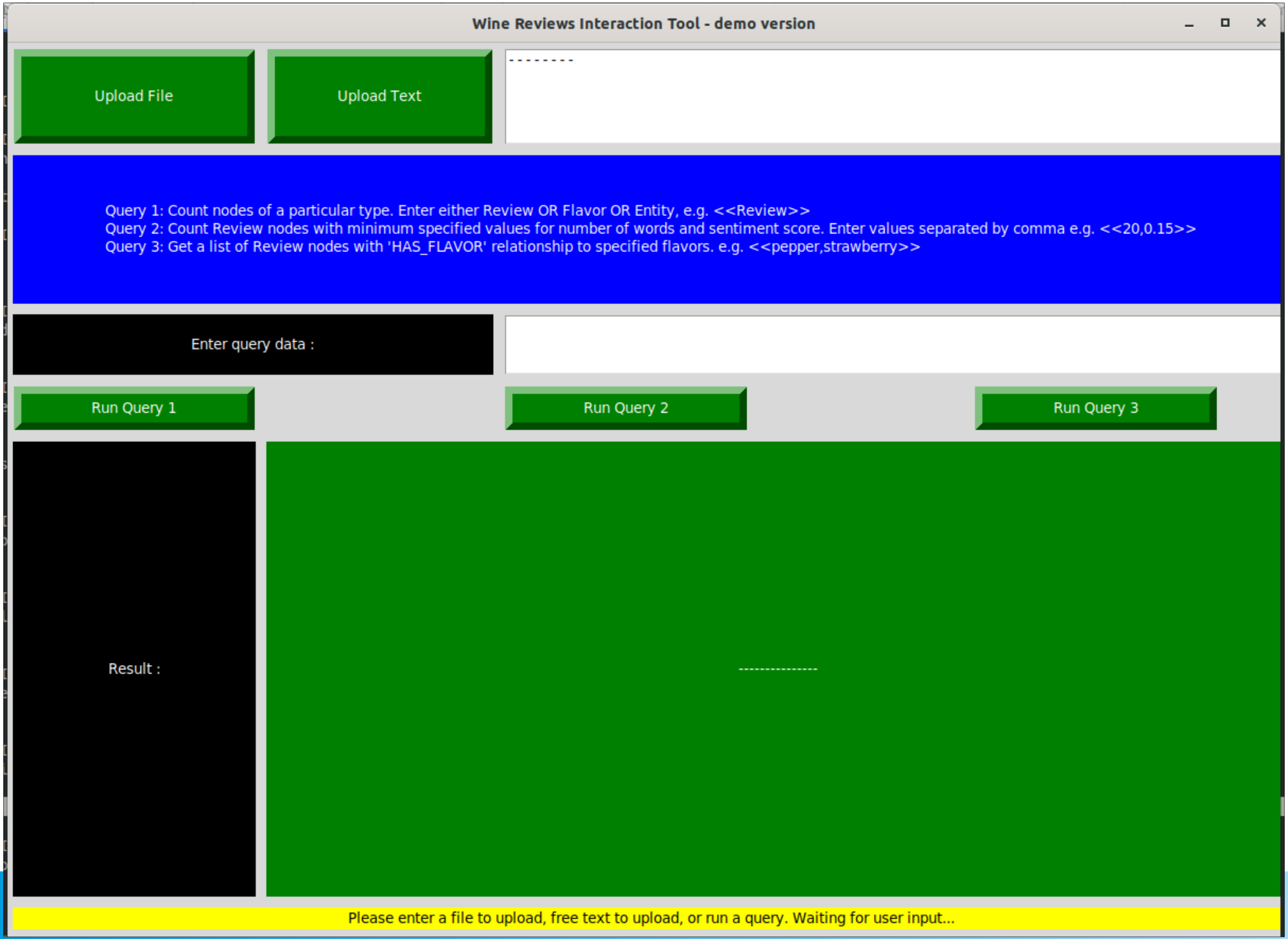
Console ouput



```
(pv8dockerusecase2) rohit@rohitu2004lts:~/PyWDUbuntu/generic/WineReviewsGraphing/code$ python3 02_load_neo_show_gui_3.py -reloadNeo Y  -uploadLimit 500

LOG_LEVEL INFO ::
Folders created or already present:
HOME = /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code
IP_DIR = /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/inData/
OP_DIR = /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/outData/
TEMP_DIR = /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/tempDir/


LOG_LEVEL INFO :: num_inp_files = 995

LOG_LEVEL INFO ::
Command line arguments checked. Proceeding with these values:
reloadNeo: Y
uploadLimit: 500

LOG_LEVEL INFO ::
Processing only 500 files....


LOG_LEVEL INFO ::
Extracted data from 501 input files....


LOG_LEVEL INFO ::
Loaded files to pandas dataframe. Total rows = 500


Data successfully dumped to json file: /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/outData/temp_neo_data.json


LOG_LEVEL INFO ::
In load_neo4j function, attempting to load file and make entries to database


LOG_LEVEL INFO ::
Successfully loaded json data from file: /home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/outData/temp_neo_data.json


LOG_LEVEL INFO ::
Cleared the graph...


LOG_LEVEL INFO ::
Total entries to process = 500
100%|████████████████████████████████████████████████████████████████████████████████| 500/500 [00:16<00:00, 30.83it/s]

LOG_LEVEL INFO ::
Updated Neo4j: Review nodes=499, Entity nodes=1, Flavor nodes=0



LOG_LEVEL INFO ::
Starting GUI logic...
```

# GUI – Main interface

User interface – initial display



Query 1: Count nodes of a particular type. Enter either Review OR Flavor OR Entity, e.g. <<Review>>
Query 2: Count Review nodes with minimum specified values for number of words and sentiment score. Enter values separated by comma e.g. <<20,0.15>>
Query 3: Get a list of Review nodes with 'HAS_FLAVOR' relationship to specified flavors. e.g. <<pepper,strawberry>>

Please enter a file to upload, free text to upload, or run a query. Waiting for user input...

Upload new data with file path or free typing

Instructions for Queries

Query data input area and Run buttons

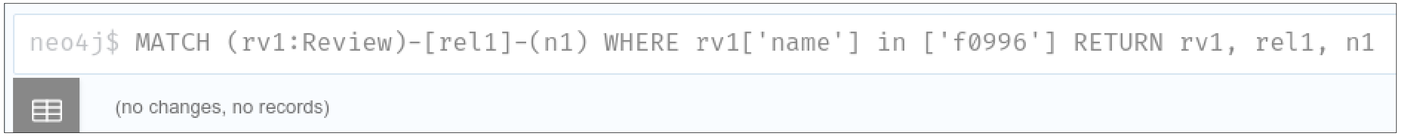
Result area

Status message

# GUI – Uploading from a file

**Adding new file for processing – “Upload File” option**

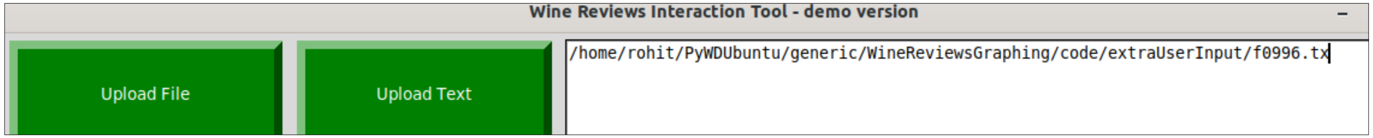
- Initially this query returns no hits: as file f0996.txt is not yet processed



```
neo4j$ MATCH (rv1:Review)-[rel1]-(n1) WHERE rv1['name'] in ['f0996'] RETURN rv1, rel1, n1
```

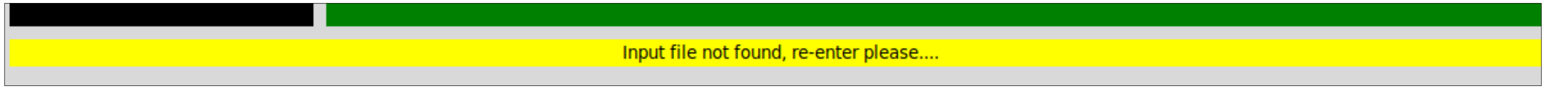
(no changes, no records)

- Processing file f0996.txt but with typo in path: status message shows file not found



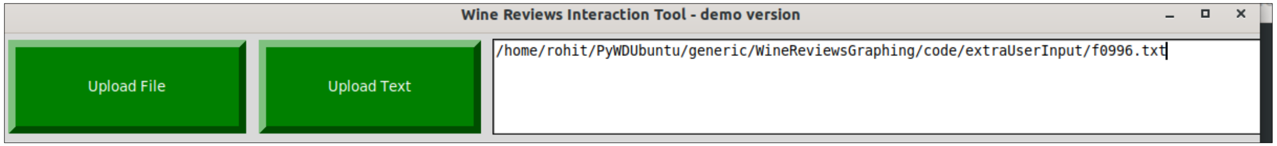
Wine Reviews Interaction Tool - demo version

/home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/extraUserInput/f0996.tx



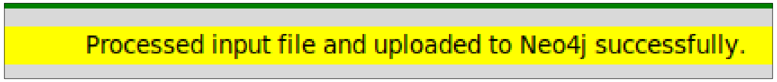
Input file not found, re-enter please....

- Correct file specified now – processed successfully and created a review node starting with ‘f’



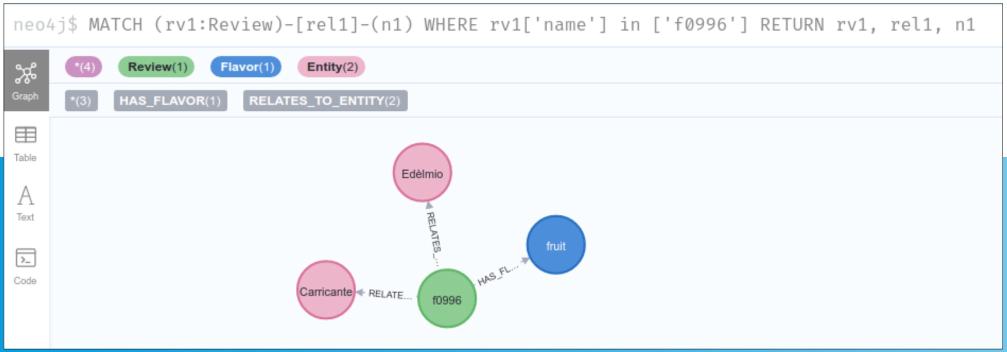
Wine Reviews Interaction Tool - demo version

/home/rohit/PyWDUbuntu/generic/WineReviewsGraphing/code/extraUserInput/f0996.txt



Processed input file and uploaded to Neo4j successfully.

- Same query now returns a hit in Neo4j for Review node with name ‘f0996’

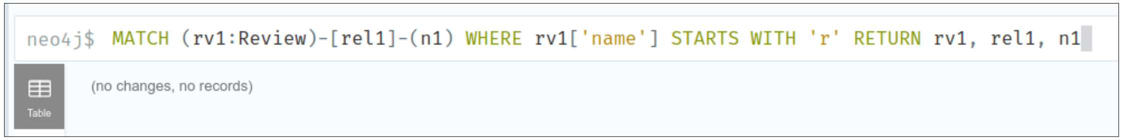


```
neo4j$ MATCH (rv1:Review)-[rel1]-(n1) WHERE rv1['name'] in ['f0996'] RETURN rv1, rel1, n1
```

# GUI – Upload typed text

## Adding new file for processing – “Upload Text” option

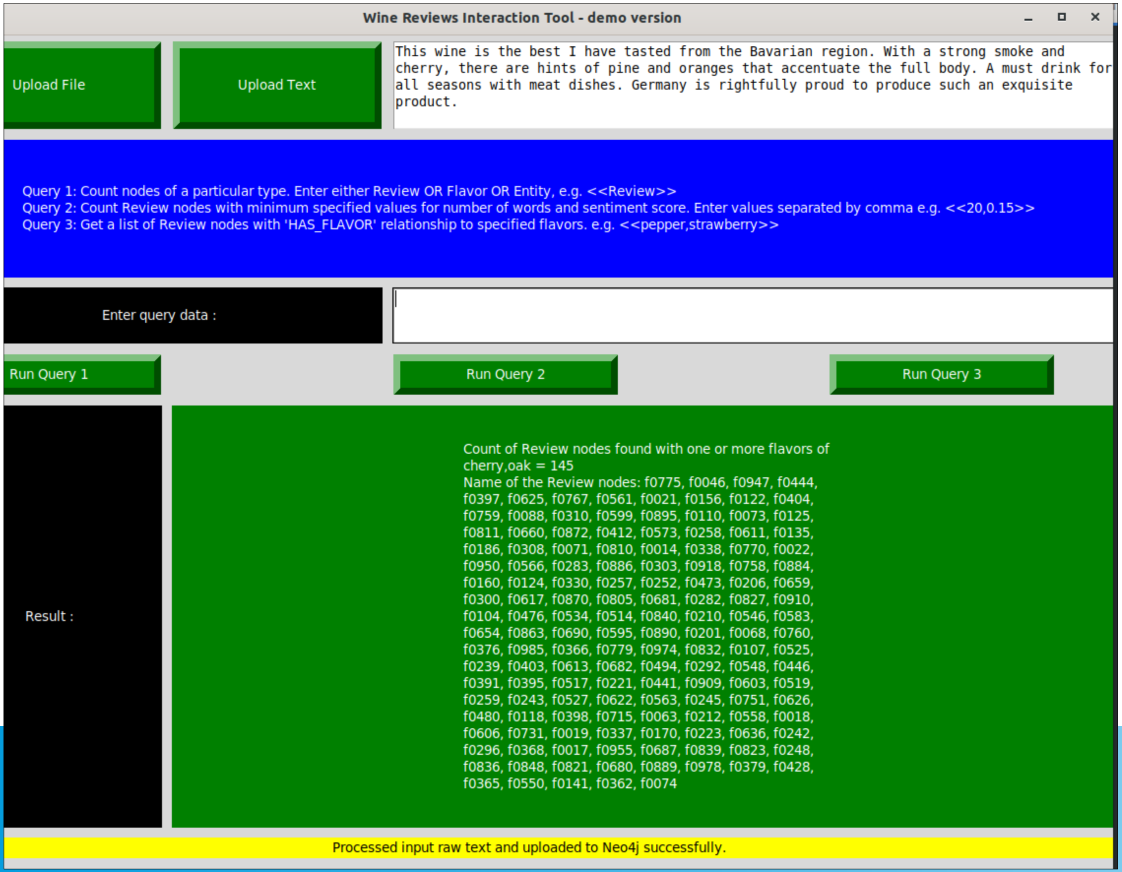
- Initially query returns no hits: free typed input uploaded as yet.



```
neo4j$ MATCH (rv1:Review)-[rel1]-(n1) WHERE rv1['name'] STARTS WITH 'r' RETURN rv1, rel1, n1
```

(no changes, no records)

- Processing free typed input
  will create a node starting with ‘r’ instead of ‘f’



Wine Reviews Interaction Tool - demo version

This wine is the best I have tasted from the Bavarian region. With a strong smoke and cherry, there are hints of pine and oranges that accentuate the full body. A must drink for all seasons with meat dishes. Germany is rightfully proud to produce such an exquisite product.

Query 1: Count nodes of a particular type. Enter either Review OR Flavor OR Entity, e.g. <<Review>>
Query 2: Count Review nodes with minimum specified values for number of words and sentiment score. Enter values separated by comma e.g. <<20,0.15>>
Query 3: Get a list of Review nodes with 'HAS_FLAVOR' relationship to specified flavors. e.g. <<pepper,strawberry>>

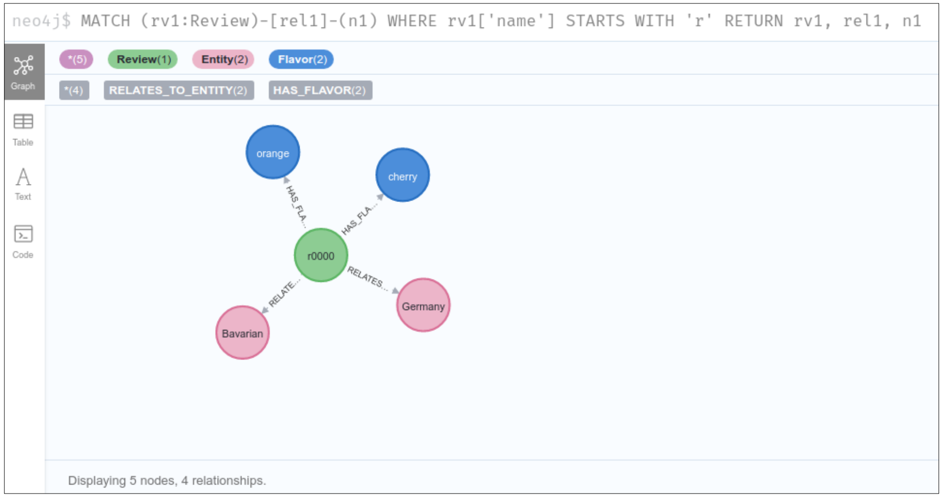
Enter query data :

Result :

```
Count of Review nodes found with one or more flavors of
cherry,oak = 145
Name of the Review nodes: f0775, f0046, f0947, f0444,
f0397, f0625, f0767, f0561, f0021, f0156, f0122, f0404,
f0759, f0088, f0310, f0599, f0895, f0110, f0073, f0125,
f0811, f0660, f0872, f0412, f0573, f0258, f0611, f0135,
f0186, f0308, f0071, f0810, f0014, f0338, f0770, f0022,
f0950, f0566, f0283, f0886, f0303, f0918, f0758, f0884,
f0160, f0124, f0330, f0257, f0252, f0473, f0206, f0659,
f0300, f0617, f0870, f0805, f0681, f0282, f0827, f0910,
f0104, f0476, f0534, f0514, f0840, f0210, f0546, f0583,
f0654, f0863, f0690, f0595, f0890, f0201, f0068, f0760,
f0376, f0985, f0366, f0779, f0974, f0832, f0107, f0525,
f0239, f0403, f0613, f0682, f0494, f0292, f0548, f0446,
f0391, f0395, f0517, f0221, f0441, f0909, f0603, f0519,
f0259, f0243, f0527, f0622, f0563, f0245, f0751, f0626,
f0480, f0118, f0398, f0715, f0063, f0212, f0558, f0018,
f0606, f0731, f0019, f0337, f0170, f0223, f0636, f0242,
f0296, f0368, f0017, f0955, f0687, f0839, f0823, f0248,
f0836, f0848, f0821, f0680, f0889, f0978, f0379, f0428,
f0365, f0550, f0141, f0362, f0074
```

Processed input raw text and uploaded to Neo4j successfully.

- Same query now returns a hit in Neo4j for Review node with name ‘r0000’



```
neo4j$ MATCH (rv1:Review)-[rel1]-(n1) WHERE rv1['name'] STARTS WITH 'r' RETURN rv1, rel1, n1
```

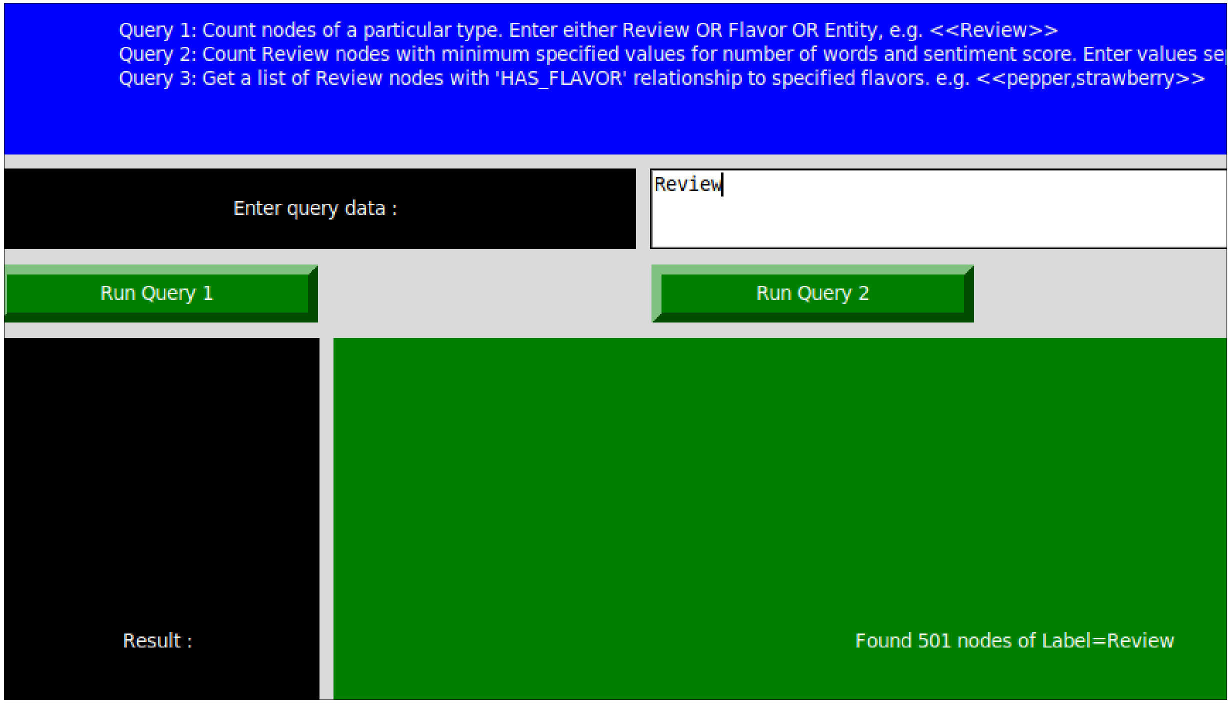
Displaying 5 nodes, 4 relationships.

# GUI – Query 1

**Query 1**: Count of particular node e.g. Review node

Shows count = 501 nodes.

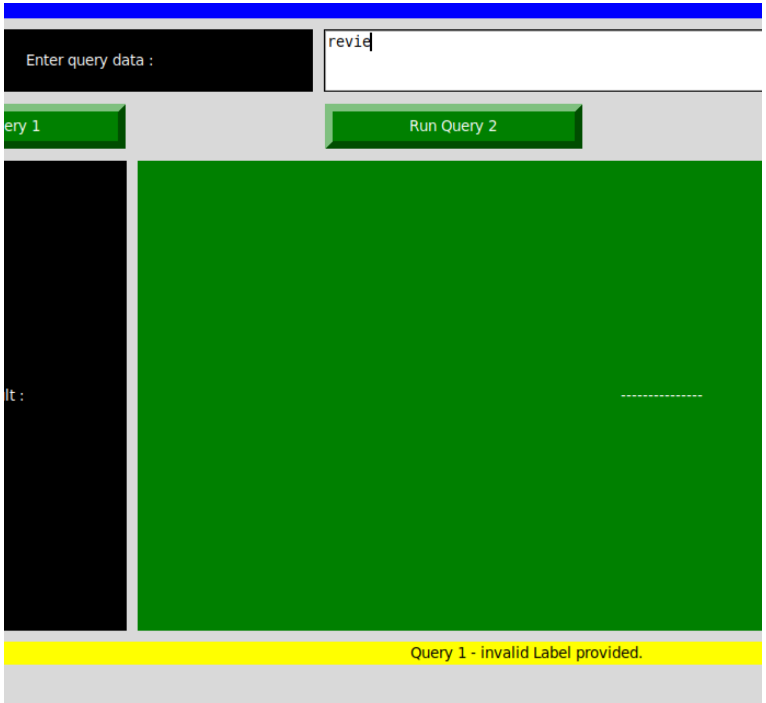
Invalid label – appropriate status message



Query 1: Count nodes of a particular type. Enter either Review OR Flavor OR Entity, e.g. <<Review>>
Query 2: Count Review nodes with minimum specified values for number of words and sentiment score. Enter values se
Query 3: Get a list of Review nodes with 'HAS_FLAVOR' relationship to specified flavors. e.g. <<pepper,strawberry>>

Enter query data : Review

Result : Found 501 nodes of Label=Review



Enter query data : revie

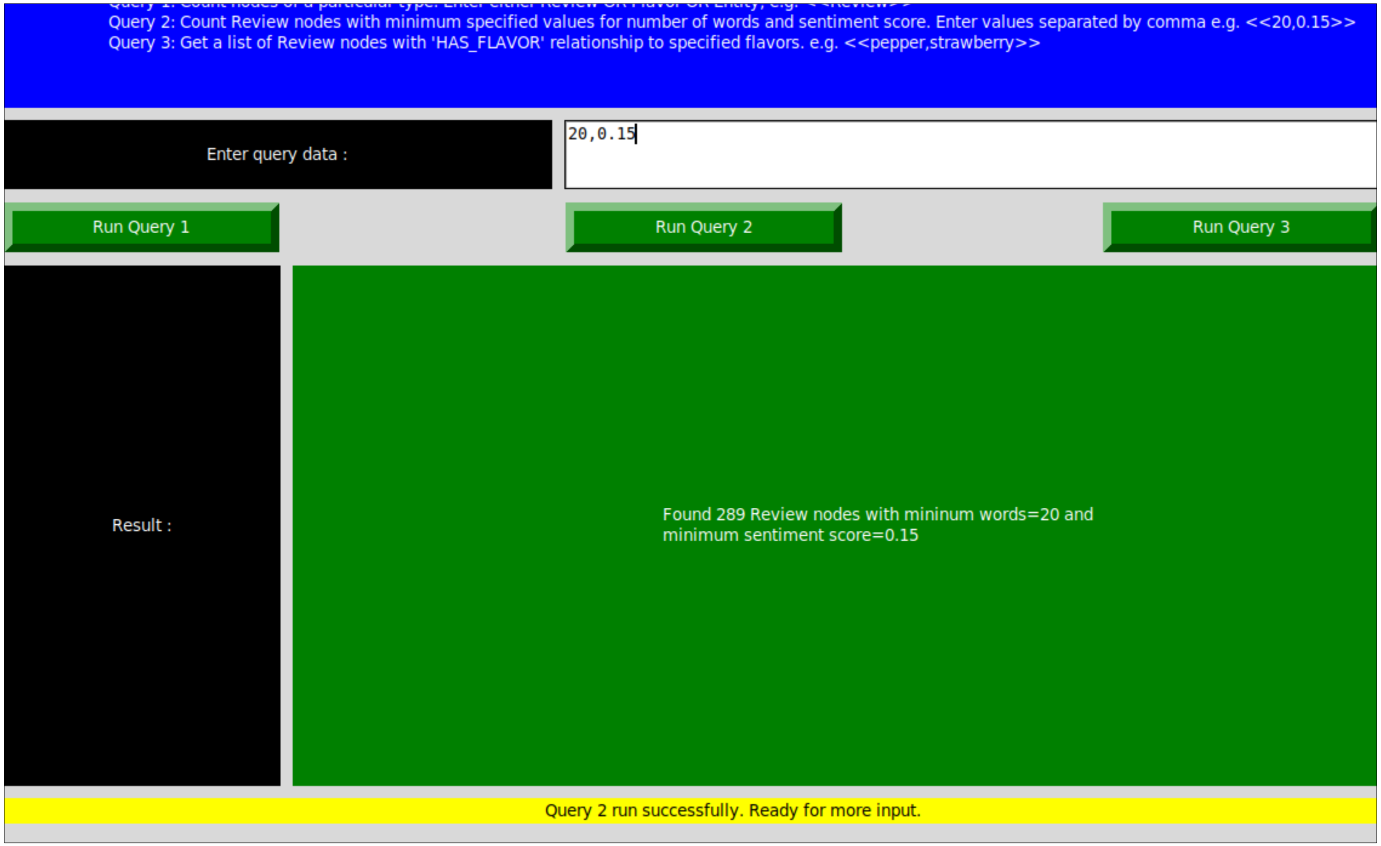
lt : ---------------

Query 1 - invalid Label provided.

# GUI – Query 2

**Query 2**: Count Review nodes with minimum 20 words and sentiment score of 0.15

Shows count = 289 nodes



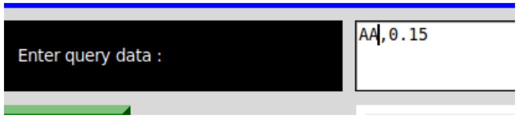
Query 1: Count nodes of a particular type. Enter either Review OR Flavor OR Entity, e.g. <<Review>>
Query 2: Count Review nodes with minimum specified values for number of words and sentiment score. Enter values separated by comma e.g. <<20,0.15>>
Query 3: Get a list of Review nodes with 'HAS_FLAVOR' relationship to specified flavors. e.g. <<pepper,strawberry>>

Enter query data : 20,0.15

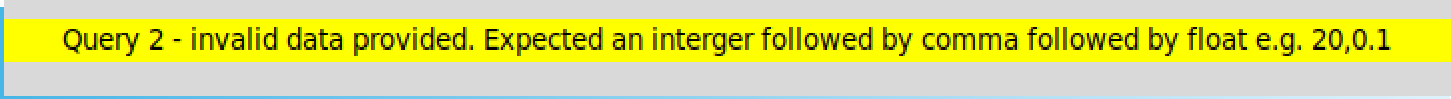
Result : Found 289 Review nodes with mininum words=20 and minimum sentiment score=0.15

Query 2 run successfully. Ready for more input.

Invalid input – appropriate status message: entered AA,0.15



Enter query data : AA,0.15

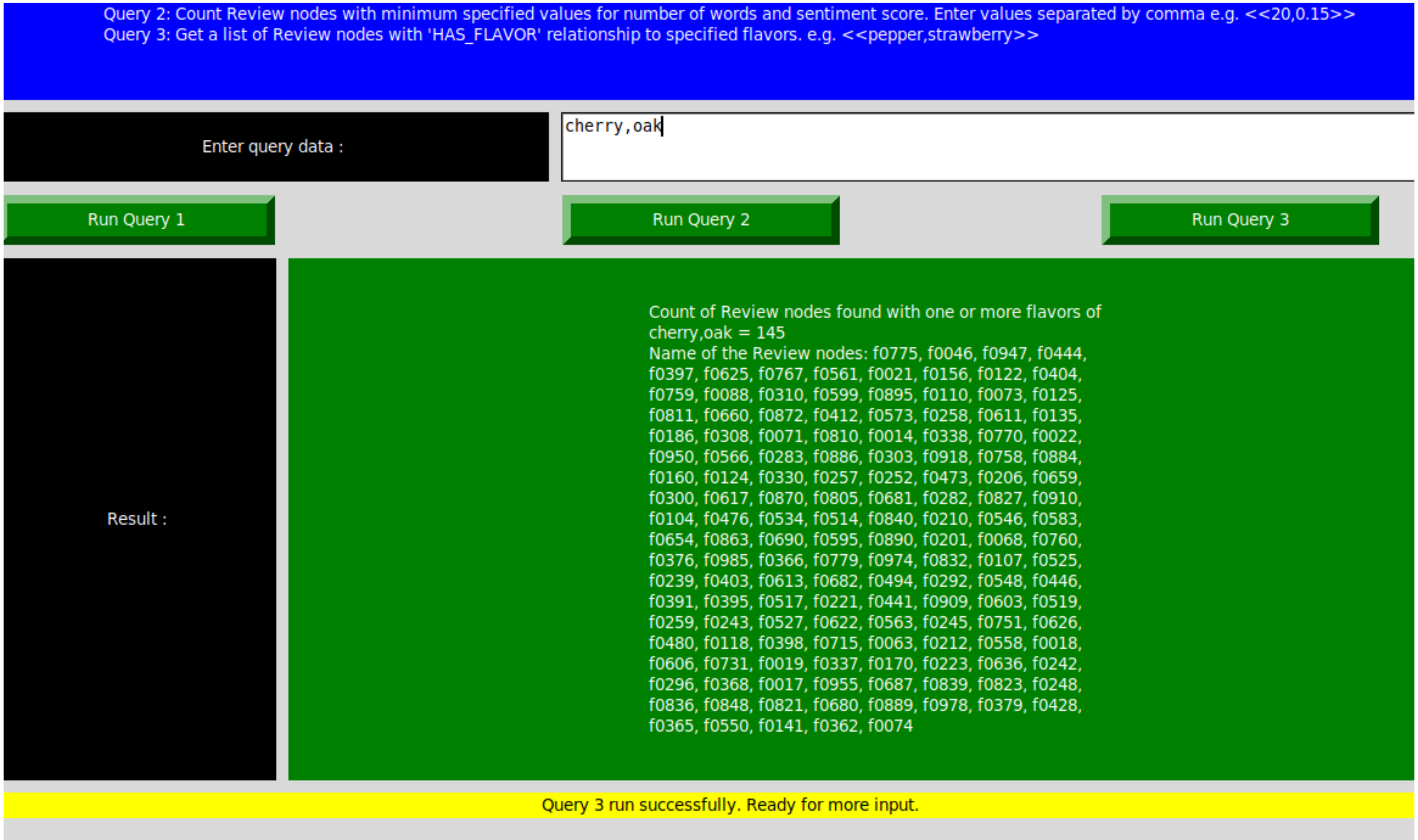


Query 2 - invalid data provided. Expected an interger followed by comma followed by float e.g. 20,0.1

# GUI – Query 3

**Query 3**: Count and show Review nodes with review having specified flavors

Shows count = 145 nodes and lists the names of the Review nodes.



Query 2: Count Review nodes with minimum specified values for number of words and sentiment score. Enter values separated by comma e.g. <<20,0.15>>
Query 3: Get a list of Review nodes with 'HAS_FLAVOR' relationship to specified flavors. e.g. <<pepper,strawberry>>

Enter query data : cherry,oak

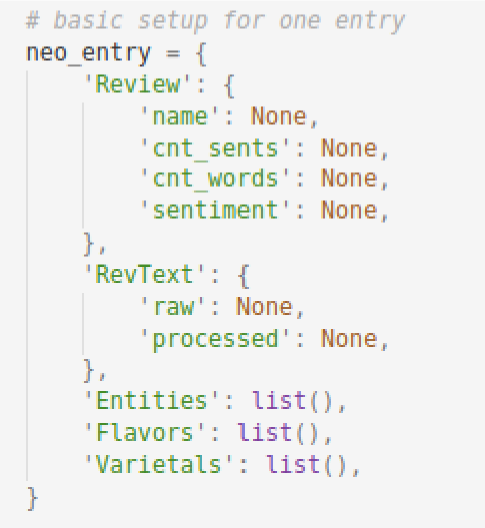
Result :

Count of Review nodes found with one or more flavors of cherry,oak = 145
Name of the Review nodes: f0775, f0046, f0947, f0444, f0397, f0625, f0767, f0561, f0021, f0156, f0122, f0404, f0759, f0088, f0310, f0599, f0895, f0110, f0073, f0125, f0811, f0660, f0872, f0412, f0573, f0258, f0611, f0135, f0186, f0308, f0071, f0810, f0014, f0338, f0770, f0022, f0950, f0566, f0283, f0886, f0303, f0918, f0758, f0884, f0160, f0124, f0330, f0257, f0252, f0473, f0206, f0659, f0300, f0617, f0870, f0805, f0681, f0282, f0827, f0910, f0104, f0476, f0534, f0514, f0840, f0210, f0546, f0583, f0654, f0863, f0690, f0595, f0890, f0201, f0068, f0760, f0376, f0985, f0366, f0779, f0974, f0832, f0107, f0525, f0239, f0403, f0613, f0682, f0494, f0292, f0548, f0446, f0391, f0395, f0517, f0221, f0441, f0909, f0603, f0519, f0259, f0243, f0527, f0622, f0563, f0245, f0751, f0626, f0480, f0118, f0398, f0715, f0063, f0212, f0558, f0018, f0606, f0731, f0019, f0337, f0170, f0223, f0636, f0242, f0296, f0368, f0017, f0955, f0687, f0839, f0823, f0248, f0836, f0848, f0821, f0680, f0889, f0978, f0379, f0428, f0365, f0550, f0141, f0362, f0074

Query 3 run successfully. Ready for more input.

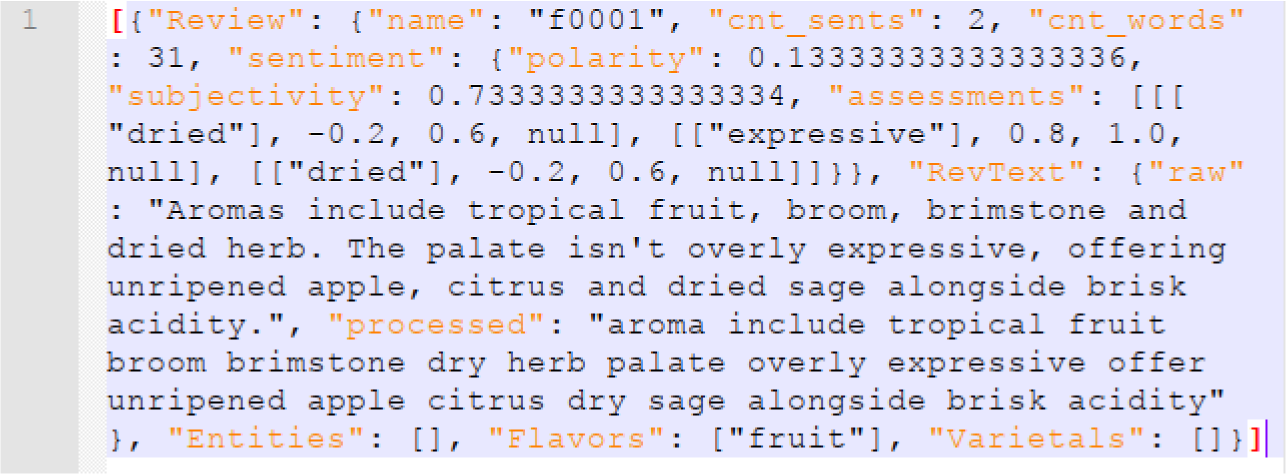
# Data structures and Json file

Custom data
structure to store
features



```
# basic setup for one entry
neo_entry = {
    'Review': {
        'name': None,
        'cnt_sents': None,
        'cnt_words': None,
        'sentiment': None,
    },
    'RevText': {
        'raw': None,
        'processed': None,
    },
    'Entities': list(),
    'Flavors': list(),
    'Varietals': list(),
}
```

Intermediate Json
file contents after
processing f0001.txt



```
[{"Review": {"name": "f0001", "cnt_sents": 2, "cnt_words"
: 31, "sentiment": {"polarity": 0.13333333333333336,
"subjectivity": 0.7333333333333334, "assessments": [[[
"dried"], -0.2, 0.6, null], [["expressive"], 0.8, 1.0,
null], [["dried"], -0.2, 0.6, null]]}}, "RevText": {"raw"
: "Aromas include tropical fruit, broom, brimstone and
dried herb. The palate isn't overly expressive, offering
unripened apple, citrus and dried sage alongside brisk
acidity.", "processed": "aroma include tropical fruit
broom brimstone dry herb palate overly expressive offer
unripened apple citrus dry sage alongside brisk acidity"
}, "Entities": [], "Flavors": ["fruit"], "Varietals": []}]
```

# Docker Images – Neo4j db

- https://hub.docker.com/repository/docker/rbewoor/myneo4j410nocmd
- One layer for the neo4j db
- Sets up virtual env and testing script
- Built with dockerfile: Dockerfile.testneo
- Optional: manually run script test_neo4j_image_connection.py to check connection to db works fine (see instructions below)

To test python connection to Neo4j from within the container of the neo4j itself AFTER neo4j has started successfully:
1) Run container and start interactive mode in new terminal

```
docker run --env NEO4J_AUTH=neo4j/cba rbewoor/myneo4j410nocmd:1.0
docker exec -it continer-id /bin/bash
```

2) Activate the virutal environment

```
source /home/.venv/virtenv_testneo_1/bin/activate
```

3) Run the script

```
python3 /home/test_neo/test_neo4j_image_connection.py
```

Will execute 2 ways of coding the connection request to neo4j:

```
gph = Graph(uri="bolt://localhost:7687",auth=("neo4j","cba"))
gph = Graph(uri="http://localhost:7687",auth=("neo4j","cba"))
```

# Docker Images – Application

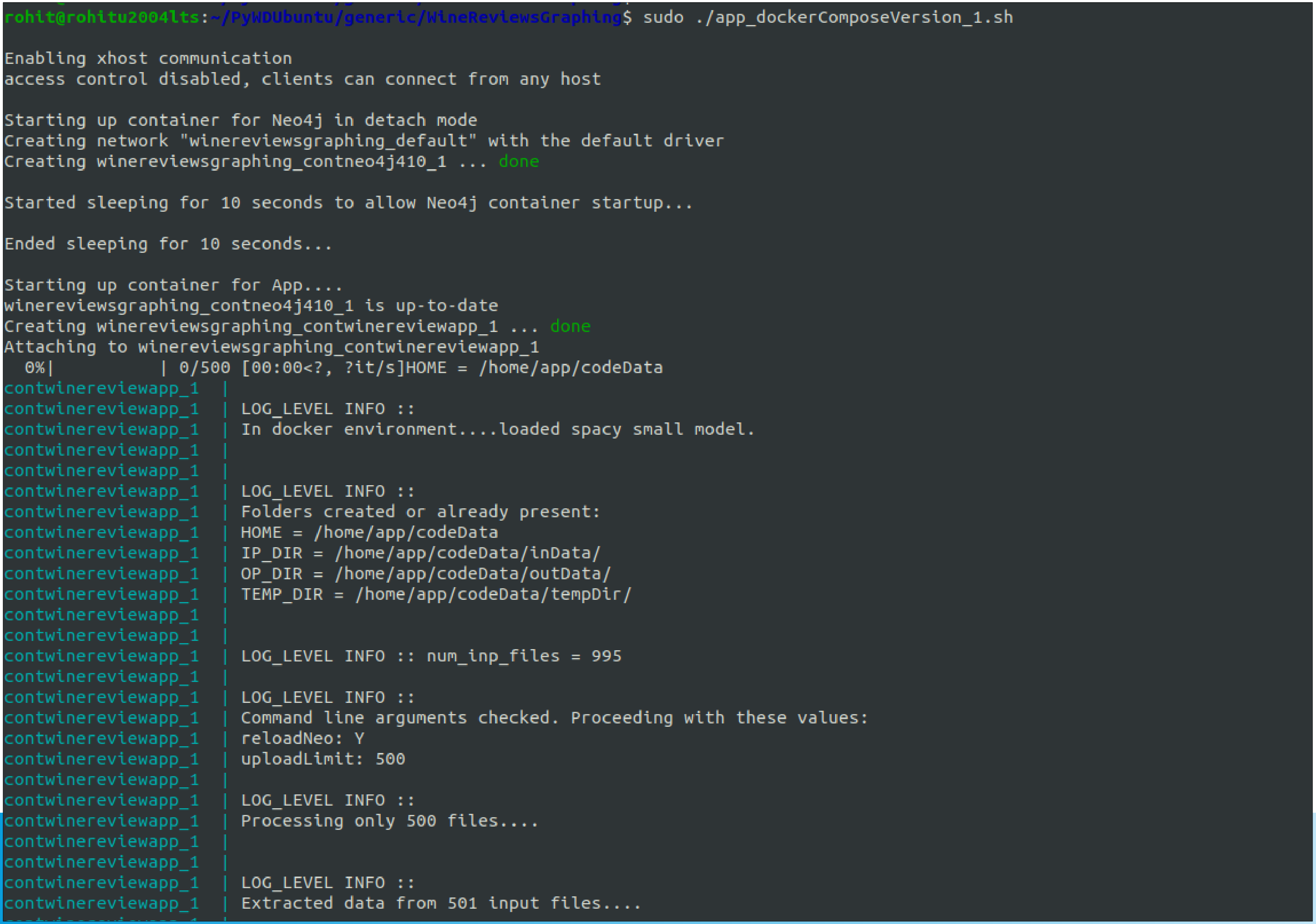
- https://hub.docker.com/repository/docker/rbewoor/winereviewapp
- One layer for python
- Sets up virtual env, scripts and necessary folders+files
- Built with dockerfile: Dockerfile.winereviewapp

# Docker – Two methods to execute

- Method 1: *Docker Run command version* of bash script:
  - Run “sudo app_dockerRunVersion_1.sh”
- Method 2: *Docker-compose command version* of bash script:
  - Copy “dockerCompose_wineReviews_1.yaml” in project folder
  - Run “sudo app_dockerComposeVersion_1.sh” from project folder
- Both versions:
  - use linux xhost to display GUI on host display
  - create a temporary folder to use as a volume for Neo4j db data
  - automatically removes volumes and temporary folder
  - disable xhost permissions

# Docker – execution example

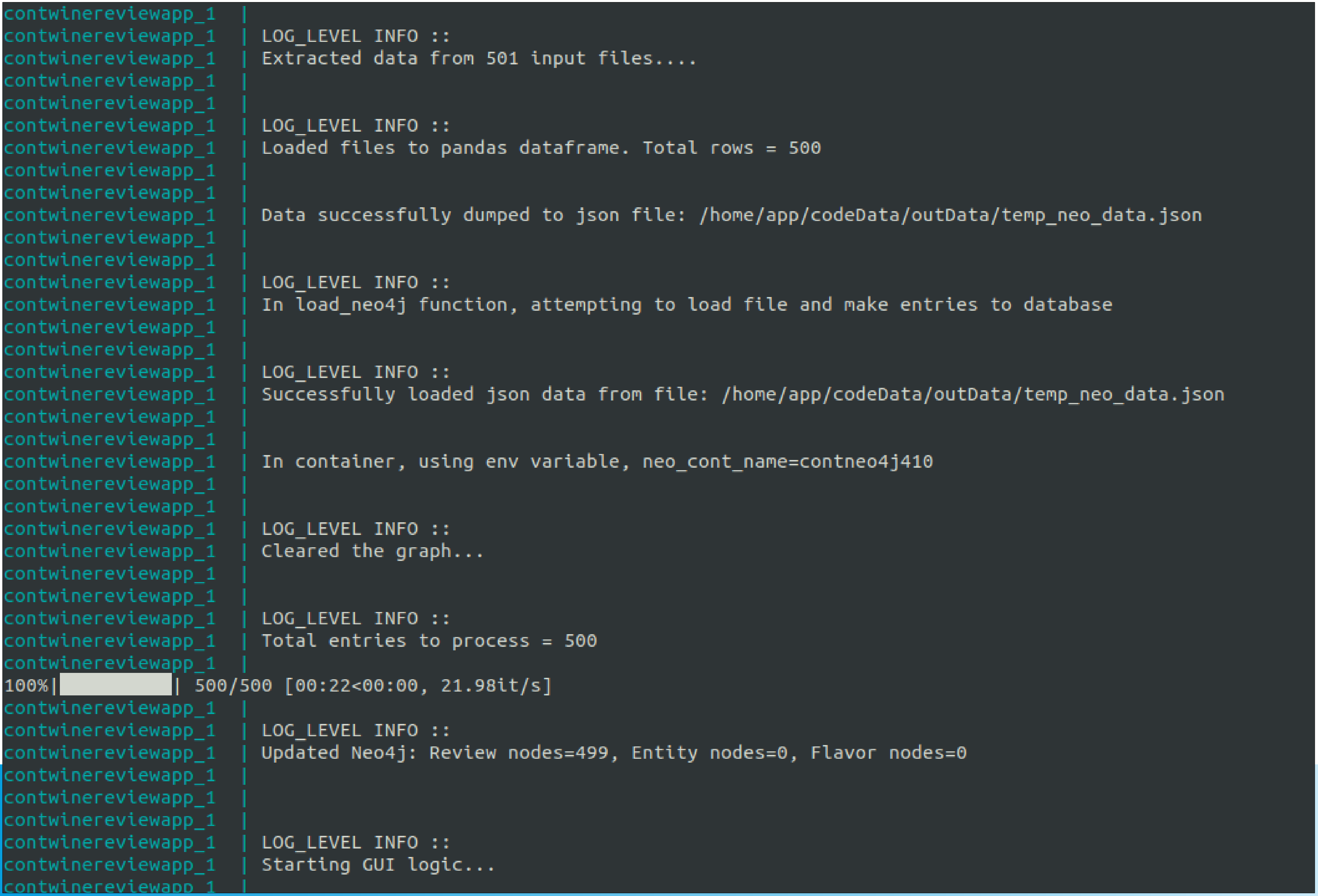
Part 1 of console output



```
rohit@rohitu2004lts:~/PyWDUbuntu/generic/WineReviewsGraphing$ sudo ./app_dockerComposeVersion_1.sh

Enabling xhost communication
access control disabled, clients can connect from any host

Starting up container for Neo4j in detach mode
Creating network "winereviewsgraphing_default" with the default driver
Creating winereviewsgraphing_contneo4j410_1 ... done

Started sleeping for 10 seconds to allow Neo4j container startup...

Ended sleeping for 10 seconds...

Starting up container for App....
winereviewsgraphing_contneo4j410_1 is up-to-date
Creating winereviewsgraphing_contwinereviewapp_1 ... done
Attaching to winereviewsgraphing_contwinereviewapp_1
  0%|          | 0/500 [00:00<?, ?it/s]HOME = /home/app/codeData
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | In docker environment....loaded spacy small model.
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Folders created or already present:
contwinereviewapp_1  | HOME = /home/app/codeData
contwinereviewapp_1  | IP_DIR = /home/app/codeData/inData/
contwinereviewapp_1  | OP_DIR = /home/app/codeData/outData/
contwinereviewapp_1  | TEMP_DIR = /home/app/codeData/tempDir/
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO :: num_inp_files = 995
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Command line arguments checked. Proceeding with these values:
contwinereviewapp_1  | reloadNeo: Y
contwinereviewapp_1  | uploadLimit: 500
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Processing only 500 files....
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Extracted data from 501 input files....
```

# Docker – execution example

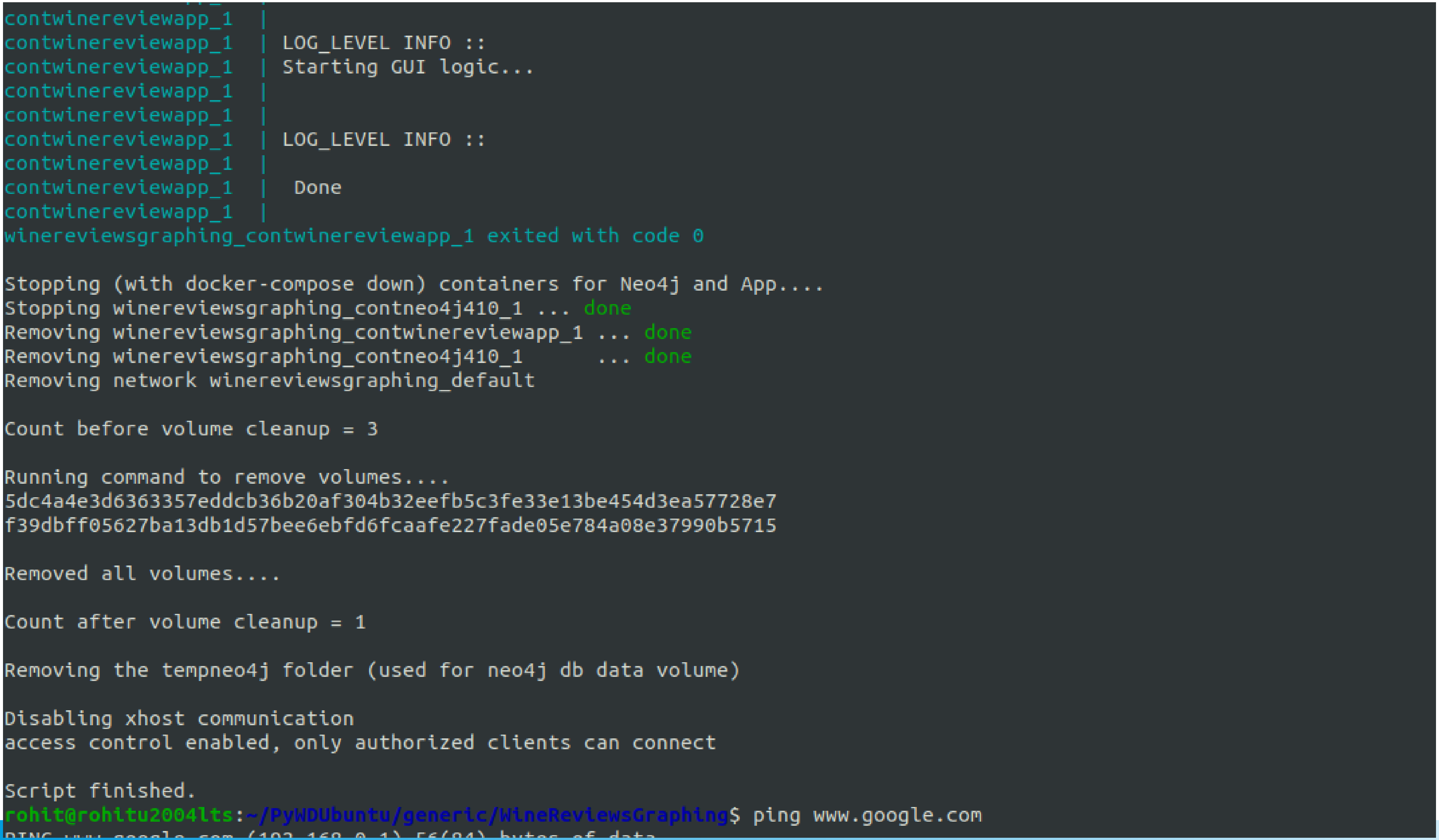
Part 2 of console output



```
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Extracted data from 501 input files....
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Loaded files to pandas dataframe. Total rows = 500
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | Data successfully dumped to json file: /home/app/codeData/outData/temp_neo_data.json
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | In load_neo4j function, attempting to load file and make entries to database
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Successfully loaded json data from file: /home/app/codeData/outData/temp_neo_data.json
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | In container, using env variable, neo_cont_name=contneo4j410
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Cleared the graph...
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Total entries to process = 500
contwinereviewapp_1  |
100%|██████████| 500/500 [00:22<00:00, 21.98it/s]
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Updated Neo4j: Review nodes=499, Entity nodes=0, Flavor nodes=0
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Starting GUI logic...
contwinereviewapp_1  |
```

# Docker – execution example

Part 3 of console output



```
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  | Starting GUI logic...
contwinereviewapp_1  |
contwinereviewapp_1  |
contwinereviewapp_1  | LOG_LEVEL INFO ::
contwinereviewapp_1  |
contwinereviewapp_1  |  Done
contwinereviewapp_1  |
winereviewsgraphing_contwinereviewapp_1 exited with code 0

Stopping (with docker-compose down) containers for Neo4j and App....
Stopping winereviewsgraphing_contneo4j410_1 ... done
Removing winereviewsgraphing_contwinereviewapp_1 ... done
Removing winereviewsgraphing_contneo4j410_1      ... done
Removing network winereviewsgraphing_default

Count before volume cleanup = 3

Running command to remove volumes....
5dc4a4e3d6363357eddcb36b20af304b32eefb5c3fe33e13be454d3ea57728e7
f39dbff05627ba13db1d57bee6ebfd6fcaafe227fade05e784a08e37990b5715

Removed all volumes....

Count after volume cleanup = 1

Removing the tempneo4j folder (used for neo4j db data volume)

Disabling xhost communication
access control enabled, only authorized clients can connect

Script finished.
rohit@rohitu2004lts:~/PyWDUbuntu/generic/WineReviewsGraphing$ ping www.google.com
PING www.google.com (192.168.0.1) 56(84) bytes of data.
```

# Future scope

- For Web application - use Flask or Django instead of Tkinter
- Allow custom Cypher query instead of pre-set queries
- Add wine varietals as a new category in NER processing
  - Will allow Relationship like WINE_TYPE
- Topic modeling to find related reviews