Department of Computer Science

Summative Coursework Set Front Page

Module Title: **Applied Data Science with Python**
Module Code: **CSMAD21**
Lecturer responsible: **Miguel Angel Sanchez Razo**
Type of Assignment (coursework / online test): **Coursework**
Individual / Group Assignment: **Individual**
Weighting of the Assignment: **100%**
Page limit/Word count: **NA**
Expected hours spent for this assignment: **40**
Items to be submitted: **Zip file containing (more details in Assignment submission requirements):**

- Scenario 1 .ipynb notebook file and its HTML version
- Scenario 2 .ipynb notebook file and its HTML version
- Scenario 2 - Task2 output file
- Scenario 3 - Task3 output file

Work to be submitted on-line via Blackboard Learn by: **10th of January 2022**
Work will be marked and returned by: **31th of January 2022**

NOTES

By submitting this work, you are certifying that it is all your sentences, figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work except where explicitly the works of others have been acknowledged, quoted, and referenced. You understand that failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly. The University's Statement of Academic Misconduct is available on the University web pages.

If your work is submitted after the deadline, *10%* of the maximum possible mark will be deducted for *each* working day (or part of) it is late. A mark of zero will be awarded if your work is submitted more than 5 working days late. You are strongly recommended to hand work in by the deadline as a late submission on one piece of work can impact on other work.

If you believe that you have a valid reason for failing to meet a deadline then you should complete an Extenuating Circumstances form and submit it to the Student Support Centre *before* the deadline, or as soon as is practicable afterwards, explaining why.

1. Assessment classifications

First Class (>= 70%)	The coursework demonstrates: Excellent knowledge and understanding of the concepts, evidence of independent research into methods used, and a thorough justification of methods
Upper Second (60-69%)	The coursework demonstrates: Good knowledge of the core concepts, showing understanding, with few mistakes. Good explanations and justification of the methods used
Lower Second (50-59%)	The coursework demonstrates: Demonstrates knowledge of core concepts but with some mistakes. Explanations and justifications of methods used are logical, but limited in depth
Third (40-49%)	The coursework demonstrates: Mistakes in application of knowledge, and shows some misunderstandings, explanation and justification of methods used is not clear or logical.
Pass (35-39%)	The coursework demonstrates: Gaps in knowledge and many mistakes, little evidence of understanding. Methods used are not explained or justified.
Fail (0-34%)	The coursework demonstrates: Large gaps in knowledge and significant mistakes, also showing limited understanding. Lack of logical explanations behind the methods used.

2. Assignment description

The coursework consists of two scenarios to assess the implementation of the Data Science process with Python as main tool.

Scenario 1 of 2: Twitter network map data extraction, pre-processing, and analysis

You have been asked to analyse information of the social media Twitter, such as the network of certain accounts, hashtags and some other data that can be extracted from it. You are required to implement a full Data Science Workflow going from the data gathering, cleaning, pre-processing, implementation of a model (network), and analysis of different statistics (e.g. Degree Distribution, Cluster coefficient, etc.); you are also required to provide justification of the process, analysis of the findings, reasoning behind the design and implementation, decisions, and assumptions.

Your Tasks

Your overall task is to implement the data science process on data collected from Twitter of at least three accounts and three hundred tweets (most recent tweets) of each account. The tasks need to be developed in a Jupiter notebook.

Task 1 – Data Gathering, Pre-processing and EDA

Implement a process/workflow to extract information from Twitter. Your solution must consider:

- API connection and data extraction from the data source.
- Data Pre-processing from the data source to transform the original data into a Pandas dataframe.
- Perform a data cleansing activity considered relevant for the process.
- Provide the explanation of the process, the justification behind it, lessons learned and findings.
- Exploratory Data Analysis of the accounts, e.g. number of followers, are the accounts producing original twits or mostly retweeting, etc.

For more details of the data extraction from Twitter please review below in this document section **5. Additional Considerations**.

Task 2 – Network analysis

The goal of this task is to create a network that represents the area of influence of the accounts/influencers selected. For this you need to consider the network as bidirectional, there are two ways to do it: you can extract the accounts that the influencer is following and/or create the links from the accounts that were retweeted. You need to provide the following:

- Provide a sample (max 10 records) of the edge list and the neighbour list of the network.
- Produce a visualisation of the network topology and discuss the output.
- Calculate statistics of the network, plot them where relevant, and discuss the results, explaining the meaning of any statistics you have calculated.
    - Statistics of the network such as
        - Degree Distribution
        - Cluster coefficient
        - Betweenness Centrality
        - Assortativity
- Conclusions and lessons learned.

Use Networkx (Python library) to calculate statistics of the network, rather than implementing your own Python code to do so. The visualisation may be hard to interpret at first, experimenting with different settings for the layout may help.

Scenario 2 of 2: Travel time to Uni

An internal department in the university is looking to create a full and comprehensive list of travel time form some English postcodes to the university Whiteknights campus. They are expecting the output to be an Excel file or similar.

The postcode areas for which they are asking to have this information are the following:

- RG, OX, SN, SP, GU, PO, SO, DH, DT, all London, SL, HP, MK, LU, AL, SG, GL, CV, B, GL, WR, HR, NP, DY, BA, BS, NN, LE and RH.

The university campuses are:

- Whiteknights RG6 6AH

The output expected (not limited) is the following:

Postcode	Car travel time to whiteknights	Public transport time to whiteknights	Walking to whiteknights
SW8 1DL			

They have provided the following data source for the postcodes: https://www.doogal.co.uk/UKPostcodes.php . Even though is not an official data source, it is reliable, properly updated, cost free and has relevant additional data such as geographical data.

The client knows that there are several providers from where to obtain the data, but they prefer Google Maps. An additional constraint is that they would like to be cost effective. Given that every API request has a cost involved and following the constraints of the project we need to make the requests effective. **There are cases where the postcodes are close to each other or even are in the same building, considering this, one strategy is to select the most representative points in the data and just execute the API request on this representative sample; then the data can be replicated to the rest of data points.**

Your task

They have contacted you to deliver this solution but initially they have requested a feasibility analysis, the solution proposal, and the strategy to follow to implement it. The tasks need to be developed in a Jupiter notebook. **Consider just one postcode area as scope for the coursework.**

Task 1: Feasibility analysis

Analyse the postcodes data source and if the information requested can be extracted from third party solutions.

- **Select one postcode area,** download it from www.doogal.com.uk and perform and EDA on the data. Highlight your findings and relevant attributes that can be considered to develop the solution.
- **Select a sample of postcodes (up to 5)** and perform the API request to Google Maps to extract the data requested by the user.
- Manipulate the data to follow the requirements.
- Elaborate your conclusions.

Task 2: Data Extraction Strategy

Once you have validated that the extraction and manipulation of the data is possible:

- **Select one postcode area** and apply a unsupervised machine learning algorithm (clustering) to extract the representative data points.
- Define the extraction strategy providing the justification of the algorithm implemented, the amount of the representative points suggested and how these will benefit the project. Provide the assumptions of the. Add some visual support showing the postcodes and the representative points.

Task 3: Solution implementation

Create the workflow to apply your strategy solution in **one postcode area.**

- Create a workflow that controls the API requests to Google Maps of the representative points postcodes you have stablished.
- Replicate the data of the representative points to the remaining postcodes (all postcodes of the postcode area selected must have distance and time information by each transport option). Manipulate the data to fulfil the requirements. Provide the output file.
- Elaborate your conclusions and lessons learned.

Important:

- Google Cloud provides $200 USD free credit, and each request has a cost involved (approx. $0.005USD). Each transport option is considered a request. Do your tests with a small data sample and once you familiarised with the solution, execute bigger batches.
- **The scenario is to be implemented in just one postcode area requested by the client.**
- Additional information of how to interact with Google Maps API can be found in Additional Considerations.

3. Assignment submission requirements

- You must create a Python 3.6 or above Jupyter notebook; when possible, use the packages included in Anaconda, Python 3.6 or above versions in your notebook. If you have a good reason to use a Python package not included in Anaconda, please contact the lecturer (m.sanchezrazo@reading.ac.uk) first to check before using it (except the libraries mentioned in 5. Additional considerations).
- Before submitting, please remove the Twitter API connection credentials that you used to extract the data as it is confidential data.
- Your notebook should be submitted on Blackboard Learn, under the Assignments section, as one archive containing:
    - A zip file with your student ID followed by module code and the legend "Coursework"(e.g. "ce9201209_CSMAD21_Coursework.zip") containing:
        - Scenario 1 .ipynb notebook file and its HTML version stating your student ID, task abbreviation (e.g. ce920109_S1.ipynb/html)
        - Scenario 2 .ipynb notebook file and its HTML version stating your student ID, task abbreviation (e.g. ce920109_S2.ipynb/html)
        - Scenario 2 - Task2 output file stating your student ID, task abbreviation and the postcode area analysed (e.g. ce920109_S2T2_RG.xls)
        - Scenario 3 - Task3 output file stating your student ID, task abbreviation and the postcode area analysed (e.g. ce920109_S2T3_RG.xls)
    - Note: The HTML version can be saved from the Jupyter interface under File -> Download as.
- At the beginning of the submission, please add the following (in a markdown cell in the notebook):
    - Module Code:
    - Assignment report Title:
    - Student Number (e.g. 25098635):
    - Date (when the work completed):
    - Actual hrs spent for the assignment:
    - Assignment evaluation (3 key points):
- Include your student ID number in the name of the file containing your work.

4. Marking scheme

	Task	Marks Available
Scenario 1	Task 1 - Data Gathering, Pre-processing and EDA Demonstrates understanding of data extraction and data pre-processing technics.	20
	Task 2 - Network Analysis Demonstrates understanding of network data analysis.	10
	Application of best practices such as implementation of functions, clear data manipulation code with Pandas/Lists/Dictionaries or any other data structure defined during the analysis.	8
	Appropriate visualisation methods chosen to answer questions about the data.	7
	Report structure (Format, completeness, readable, coherent).	5
Scenario 2	Task 1 - Data Gathering and Pre-processing Demonstrates understanding of data extraction and data pre-processing technics.	10
	Task 2 - EDA Demonstrates understanding of exploratory data analysis.	10
	Task 3 - Network Analysis Demonstrates understanding of network data analysis.	10
	Application of best practices such as implementation of functions, clear data manipulation code with Pandas/Lists/Dictionaries or any other data structure defined during the analysis.	8
	Appropriate visualisation methods chosen to answer questions about the data.	7
	Report structure (Format, completeness, readable, coherent).	5
	• **Total**	**100**

5. Additional considerations

Scenario 1

To extract the twits from the accounts/influencers you have selected, one of the team members would have to do the following:

1. Have or create a Twitter account
2. Request a developer twitter account. You can request it in the following link: Apply for access – Twitter Developers | Twitter Developer Platform
    a. You might need to validate your account
    b. You need to create a "new app" and provide information regarding the reason why you would like to have access. You can mention something related twit's analysis such as sentimental, network and/or hashtag analysis, etc.
    c. The account can take up to a week to be validated.
    d. Once the account is validated, click on "Create app"
        i. Provide the information regarding the app (name, description, etc.)
    e. Go to "Key and tokens" and copy the credentials (API key, API key secret, Bearer token, Access token, Access token secret). **This info is shown once so please keep it somewhere safe** (you can request them later, but they are going to be different).
    f. Some reference videos:
        i. How To EASILY Get Twitter API KEY | Apply For Twitter Developers Account | Download REAL-TIME TWEETS - YouTube

Other considerations:

1. The Python libraries recommended (you might need to install them depending on the environment) to extract and manipulate the data are:
    a. Tweepy (more details in Appendix A – Tweepy installation)- Tweepy Documentation — tweepy 3.10.0 documentation
    b. Twint (Good option if you struggle to get the Twitter Developer Account, more details in Appendix b – Twint installation) - twint · PyPI
    c. Json - json — JSON encoder and decoder — Python 3.9.6 documentation
2. Please ponder the Twitte API rate restrictions, here the reference:
        i. Rate limits | Docs | Twitter Developer Platform
3. Review the Data Dictionary of the API messages, here the reference:
        i. https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet

Scenario 2

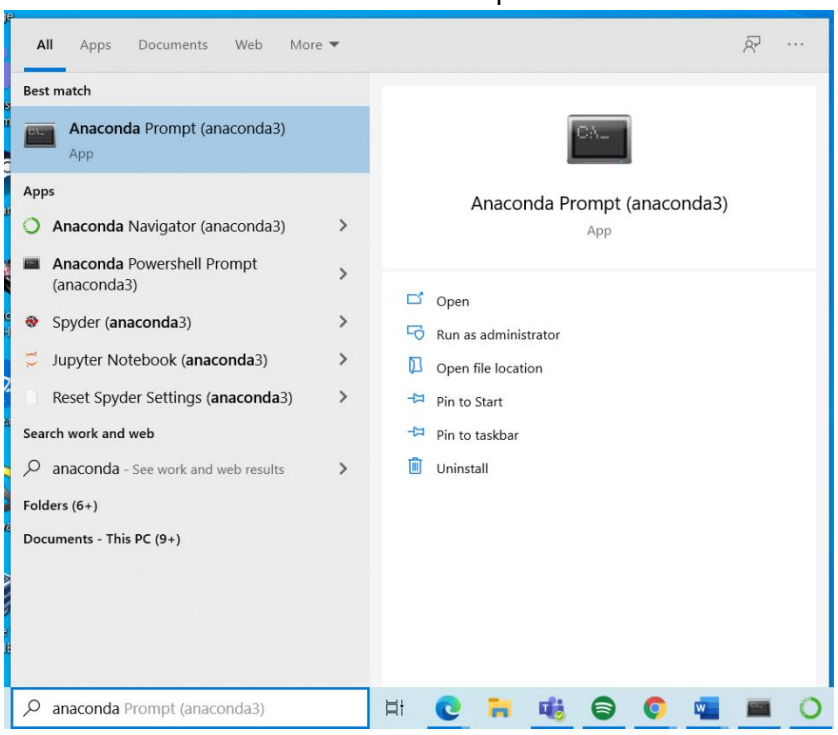
1. Register a Google Maps Platform: https://developers.google.com/maps/gmp-get-started#create-project
2. Review the documentation of The Distance Matrix API; this option provides the information required by the case. https://developers.google.com/maps/documentation/distance-matrix/overview#Introduction

Do not forget to contact your lecturer in case you encounter any problem or if you have any question.

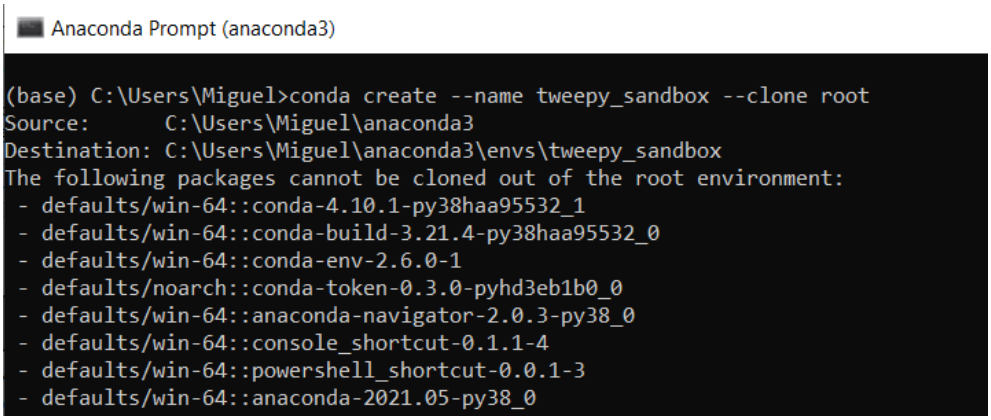
Good luck!

6. Appendix A - Tweepy installation and Troubleshooting (under Anaconda)

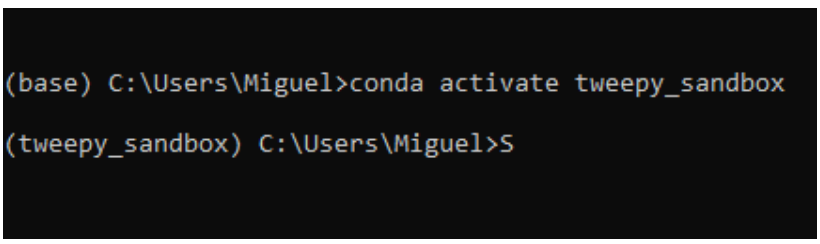
1. Create a new anaconda environment:
    a. Open a new instance of “Anaconda Prompt”



    b. Execute the following command: `conda create --name myclone --clone root`
        i. Change “myclone” to the name of your preference



    c. Activate the environment by: `conda activate myclone`
        i. Change “myclone” to the name you selected before



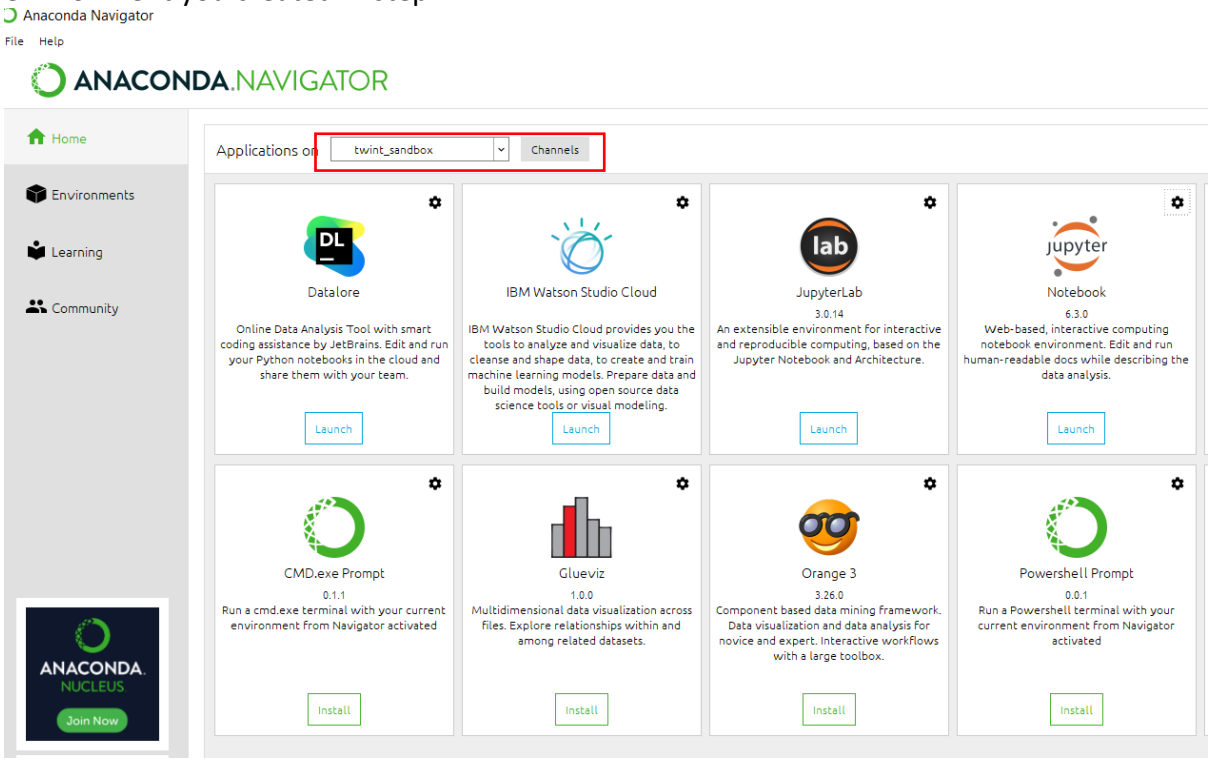
2. Install Tweepy by executing the following command In the Anaconda Prompt under the environment you defined: `pip install tweepy`

```
(tweepy_sandbox) C:\Users\Miguel>pip install tweepy
Collecting tweepy
  Downloading tweepy-3.10.0-py2.py3-none-any.whl (30 kB)
Collecting requests-oauthlib>=0.7.0
  Downloading requests_oauthlib-1.3.0-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: six>=1.10.0 in c:\users\miguel\anaconda3\envs\
py) (1.15.0)
Requirement already satisfied: requests[socks]>=2.11.1 in c:\users\miguel\ana
```

1. Testing.

   Before this please make sure you have a Twitter developer account and a project validated with the API connection credentials needed.

Open an “Anaconda Navigator” and change the “Application on” selecting the environment you created in step 1.



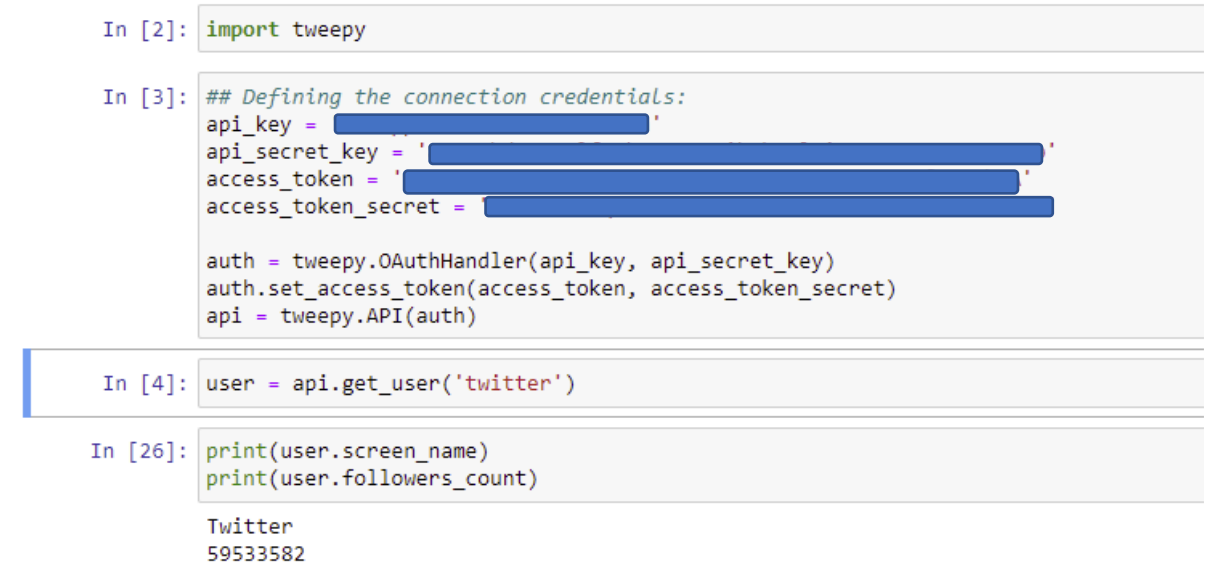
a. Open a new Jupyter notebook and type the following:

Defining the connection credentials:

```
import tweepy
api_key = #Your api_key
api_secret_key = #Your api_secret_key
access_token = #Your access_token
access_token_secret = #Your access_token_secret

auth = tweepy.OAuthHandler(api_key, api_secret_key)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
user = api.get_user('twitter')

print(user.screen_name)
print(user.followers_count)
```



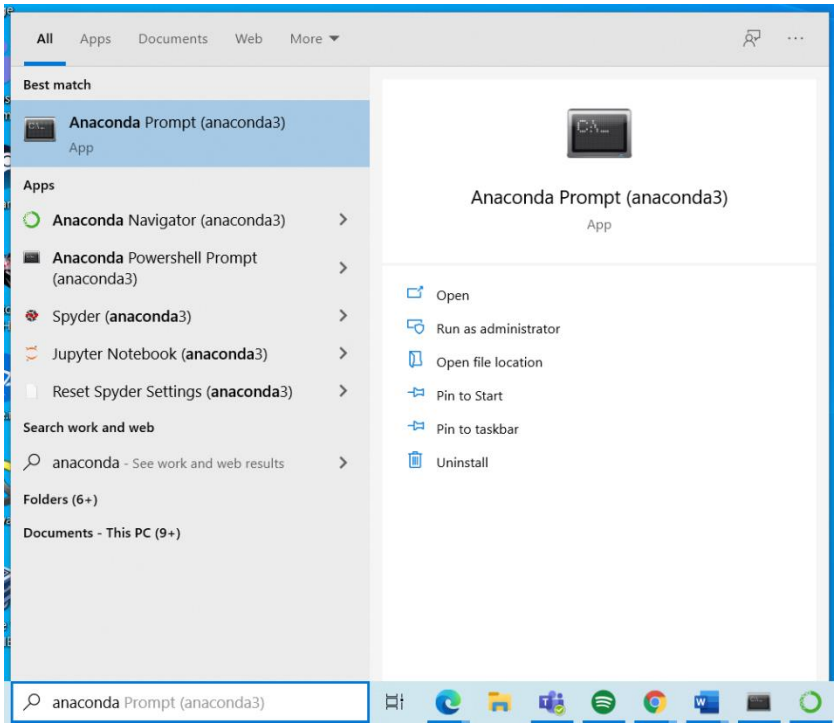
Now you can extract data from twitter!

Do not forget to review the documentation to access the data you are looking for (methods like "`user_timeline`"): https://docs.tweepy.org/en/stable/index.html

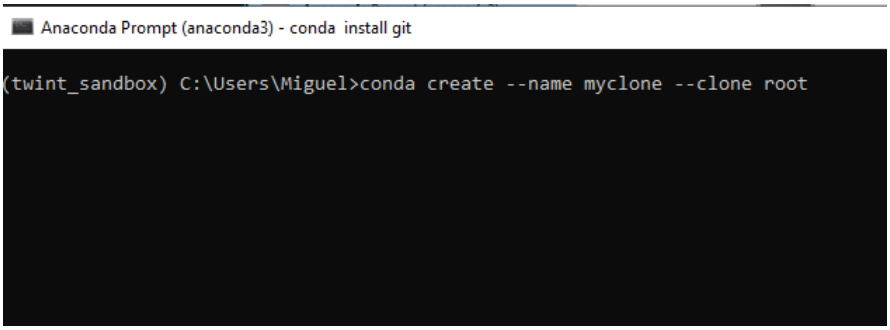
Contact your lecturer in case you have any questions.

6. Appendix B- Twint installation and Troubleshooting (under Anaconda)

1. Create a new anaconda environment:
    a. Open a new instance of “Anaconda Prompt”



    b. Execute the following command: `conda create --name myclone --clone root`
        i. Change “myclone” to the name of your preference

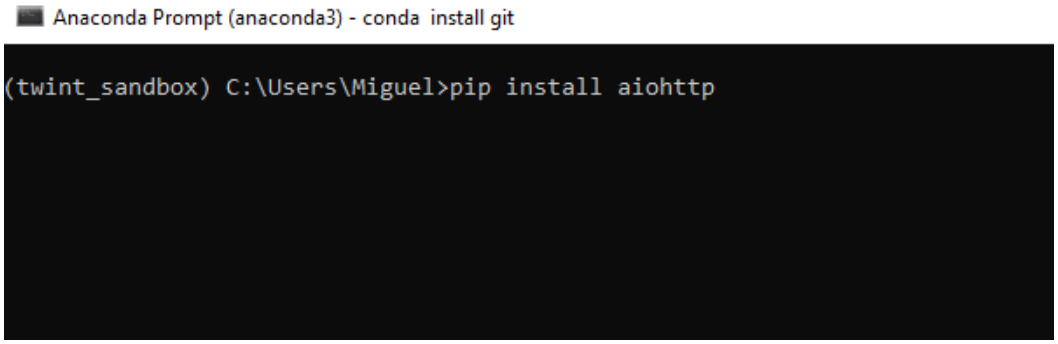


    c. Activate the environment by: `conda activate myclone`
        i. Change “myclone” to the name you selected before
2. Installing Twint Requiremts/dependencies. According to the documentation, the following packages need to be installed beforehand. https://pypi.org/project/twint/

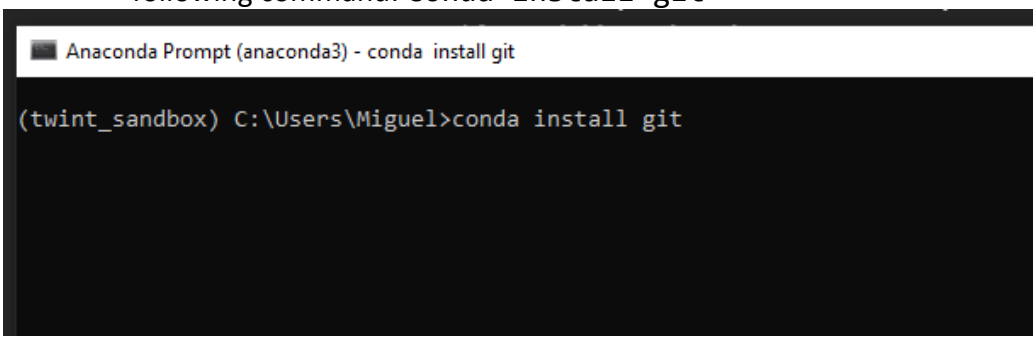
Requirements

- Python 3.6;
- aiohttp;
- aiodns;
- beautifulsoup4;
- cchardet;
- elasticsearch;
- pysocks;
- pandas (>=0.23.0);
- aiohttp_socks;
- schedule;
- geopy;
- fake-useragent;
- py-googletransx.

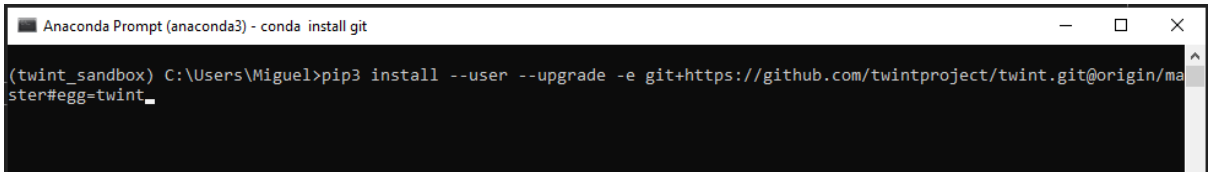
a. In the Anaconda Prompt under the environment you defined, execute the following command for **each library listed as requirement, exempt Python and Pandas:**
`pip install googletransx`



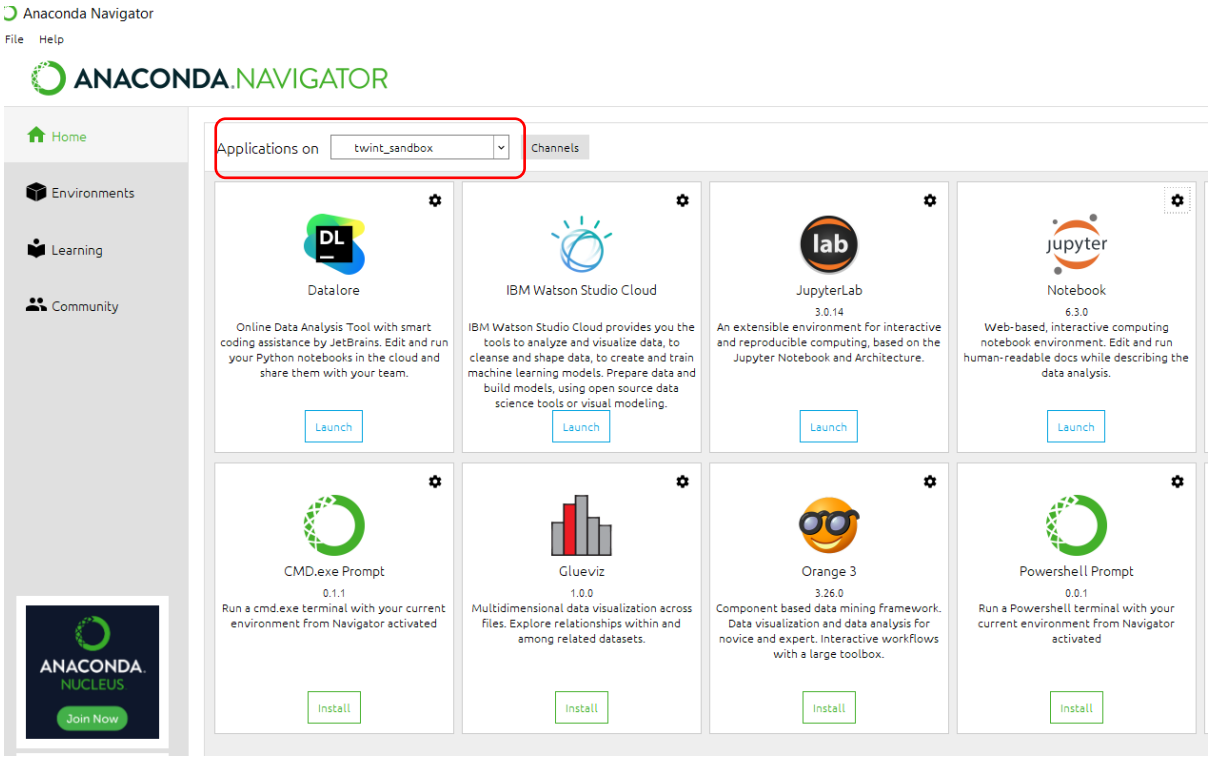
Once you finish with the installation of the requirements, execute the following command: `conda install git`



3. Now you can install Twint by executing the command provided in the documentation (https://pypi.org/project/twint/): `pip3 install --user --upgrade -e git+https://github.com/twintproject/twint.git@origin/master#egg=twint`



4. Testing. Open an “Anaconda Navigator” and change the “Application on” selecting the environment you created in step 1.



a. Open a new Jupyter notebook and type the following:

```
import twint
import nest_asyncio#
nest_asyncio.apply()

# Configure
c = twint.Config()
c.Username = "now"
c.Search = "fruit"

# Run
twint.run.Search(c)
```

Important to say that nest_asyncio is needed to execute twint in a Jupyter notebook, so don’t forget to import it and apply it in your code

In [64]:
```
import twint
import nest_asyncio
nest_asyncio.apply()

# Configure
c = twint.Config()
c.Username = "now"
c.Search = "fruit"

# Run
twint.run.Search(c)
```

```
1368952974979039235 2021-03-08 15:53:17 +0100 <NOW> @starwarspassion "You okay hun? You've hardly touched your floating Naboo f
ruit"
1365632930631065600 2021-02-27 12:00:37 +0100 <NOW> Billie Holiday was all things powerful, from her voice, her soul, to her fi
ght.  Total goosebumps watching @AndraDayMusic performing Strange Fruit in #USvsBillieHoliday 🎙  https://t.co/3e1xyjrp7T
584720918468956160 2015-04-05 15:15:05 +0100 <NOW> We advise against looking at all the soft fruit in your kitchen as you get r
eady for the last two episodes. #GoTNOWTV  http://t.co/RR0YPdMKFg
503965149376819200 2014-08-25 19:00:29 +0100 <NOW> #Friends #WhoSaidIt "Goodbye, you fruit drying psychopath."  http://t.co/Aq8
AlLiymY
323156256846147584 2013-04-13 20:30:30 +0100 <NOW> Life's too short to eat fruit.  http://t.co/jil6m7KJgM  http://t.co/N2WCNuvg
6A
305336396028198912 2013-02-23 15:20:44 +0100 <NOW> @TheThomasSmith Great choice, Thomas! We're all different, but in the end, w
e're all fruit. That's Saturday sorted, but what about Sunday?
[!] No more data! Scraping will stop now.
found 0 deleted tweets in this search.
```

Now you can extract data from twitter!

Do not forget to review the documentation to access the data you are looking for (parameters as: "`Retweets`", "`Profile_full`" and "`twint.run.Profile`"): https://pypi.org/project/twint/.

Contact your lecturer in case you have any questions.