Instructions

Erwin Chege HW 4

1) Label your homework solutions with your last name, such as HW3_solutions.

If you do not want to share your name with your peer grader, you do not have to write your name in the assignment solutions from now on

2) Deliverables: You need to submit 3 files.

- ipython notebook file with all of your codes
- HTML file of your ipython notebook file.
- Your HW solution in a pdf file. Please write this part of the solution in a research paper format: Give a proper title, solve each problem, and write findings so that you tell a story with the data set you have and the problems you need to solve. Attach plots, figures, and code snippets wherever necessary.

3) To submit your homework solutions, go to our course in Canvas --> Assignments --> HW 4 --> Submit Assignment --> Click "Choose File" --> choose the HTML file you saved --> Click another file--> choose ipynb file--> Repeat this process to add a pdf file--submit. Once you submit, download and check to ensure that you have submitted the correct files.

4) \textbf{Your submission will be compared against other students' solutions and online databases to check for plagiarism.}

5) You can use this ipynb file to start your solution.

6) Here are the links for the data used in this HW.

- textdata is a modified form of the data from https://www.kaggle.com/competitions/commonlitreadabilityprize/overview
- social network ids data set is just a data with two column of ids and does not need any more information about the ids in this HW. 7) This HW will be peer graded + Instructor graded.

WARNING: Disclosure of this assignment and assignment answers to anybody or any website is a contributory infringement of academic dishonesty at ISU. Do not share or post course materials without the express written consent of the copyright holder and instructor. The class will follow Iowa State University's policy on academic dishonesty. Anyone suspected of academic dishonesty will be reported to the Dean of Students Office.

Each problem is worth 25 points. Total $\mathbf{25 \times 4 = 100}$.

In [ ]:
```
!pip install yfinance
!pip install yahoofinancials
!pip install beautifulsoup4
!pip install yahoo-fin -U
!pip install scikit-surprise
!pip install scrapy
!pip install newspaper3k
!pip install GoogleNews
```

```
Requirement already satisfied: yfinance in c:\users\erwin\anaconda3\lib\site-packag
es (0.2.18)
Requirement already satisfied: numpy>=1.16.5 in c:\users\erwin\anaconda3\lib\site-p
ackages (from yfinance) (1.23.5)
Requirement already satisfied: lxml>=4.9.1 in c:\users\erwin\anaconda3\lib\site-pac
kages (from yfinance) (4.9.1)
Requirement already satisfied: html5lib>=1.1 in c:\users\erwin\anaconda3\lib\site-p
ackages (from yfinance) (1.1)
Requirement already satisfied: pandas>=1.3.0 in c:\users\erwin\anaconda3\lib\site-p
ackages (from yfinance) (1.4.4)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\erwin\anaconda3\lib\
site-packages (from yfinance) (0.0.9)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\users\erwin\anaconda3\l
ib\site-packages (from yfinance) (4.11.1)
Requirement already satisfied: pytz>=2022.5 in c:\users\erwin\anaconda3\lib\site-pa
ckages (from yfinance) (2022.7)
Requirement already satisfied: appdirs>=1.4.4 in c:\users\erwin\anaconda3\lib\site-
packages (from yfinance) (1.4.4)
Requirement already satisfied: cryptography>=3.3.2 in c:\users\erwin\anaconda3\lib\
site-packages (from yfinance) (38.0.4)
Requirement already satisfied: frozendict>=2.3.4 in c:\users\erwin\anaconda3\lib\si
te-packages (from yfinance) (2.3.7)
Requirement already satisfied: requests>=2.26 in c:\users\erwin\anaconda3\lib\site-
packages (from yfinance) (2.28.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\erwin\anaconda3\lib\site-p
ackages (from beautifulsoup4>=4.11.1->yfinance) (2.3.2.post1)
Requirement already satisfied: cffi>=1.12 in c:\users\erwin\anaconda3\lib\site-pack
ages (from cryptography>=3.3.2->yfinance) (1.15.1)
Requirement already satisfied: webencodings in c:\users\erwin\anaconda3\lib\site-pa
ckages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: six>=1.9 in c:\users\erwin\anaconda3\lib\site-packag
es (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\erwin\anaconda3\l
ib\site-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\erwin\anaconda
3\lib\site-packages (from requests>=2.26->yfinance) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\erwin\anaconda3\li
b\site-packages (from requests>=2.26->yfinance) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\erwin\anaconda3\lib\s
ite-packages (from requests>=2.26->yfinance) (2022.12.7)
Requirement already satisfied: idna<4,>=2.5 in c:\users\erwin\anaconda3\lib\site-pa
ckages (from requests>=2.26->yfinance) (3.4)
Requirement already satisfied: pycparser in c:\users\erwin\anaconda3\lib\site-packa
ges (from cffi>=1.12->cryptography>=3.3.2->yfinance) (2.21)
Requirement already satisfied: yahoofinancials in c:\users\erwin\anaconda3\lib\site
-packages (1.14)
Requirement already satisfied: pytz in c:\users\erwin\anaconda3\lib\site-packages
(from yahoofinancials) (2022.7)
Requirement already satisfied: requests>=2.26 in c:\users\erwin\anaconda3\lib\site-
packages (from yahoofinancials) (2.28.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\erwin\anaconda3\li
b\site-packages (from requests>=2.26->yahoofinancials) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\erwin\anaconda3\lib\s
ite-packages (from requests>=2.26->yahoofinancials) (2022.12.7)
Requirement already satisfied: idna<4,>=2.5 in c:\users\erwin\anaconda3\lib\site-pa
ckages (from requests>=2.26->yahoofinancials) (3.4)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\erwin\anaconda
3\lib\site-packages (from requests>=2.26->yahoofinancials) (2.0.4)
Requirement already satisfied: beautifulsoup4 in c:\users\erwin\anaconda3\lib\site-
```

```
packages (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\erwin\anaconda3\lib\site-p
ackages (from beautifulsoup4) (2.3.2.post1)
Requirement already satisfied: yahoo-fin in c:\users\erwin\anaconda3\lib\site-packa
ges (0.8.9.1)
Requirement already satisfied: pandas in c:\users\erwin\anaconda3\lib\site-packages
(from yahoo-fin) (1.4.4)
Requirement already satisfied: requests-html in c:\users\erwin\anaconda3\lib\site-p
ackages (from yahoo-fin) (0.10.0)
Requirement already satisfied: feedparser in c:\users\erwin\anaconda3\lib\site-pack
ages (from yahoo-fin) (6.0.10)
Requirement already satisfied: requests in c:\users\erwin\anaconda3\lib\site-packag
es (from yahoo-fin) (2.28.1)
Requirement already satisfied: sgmllib3k in c:\users\erwin\anaconda3\lib\site-packa
ges (from feedparser->yahoo-fin) (1.0.0)
Requirement already satisfied: numpy>=1.18.5 in c:\users\erwin\anaconda3\lib\site-p
ackages (from pandas->yahoo-fin) (1.23.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\erwin\anaconda3\l
ib\site-packages (from pandas->yahoo-fin) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\erwin\anaconda3\lib\site-pa
ckages (from pandas->yahoo-fin) (2022.7)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\erwin\anaconda
3\lib\site-packages (from requests->yahoo-fin) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\erwin\anaconda3\li
b\site-packages (from requests->yahoo-fin) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\erwin\anaconda3\lib\s
ite-packages (from requests->yahoo-fin) (2022.12.7)
Requirement already satisfied: idna<4,>=2.5 in c:\users\erwin\anaconda3\lib\site-pa
ckages (from requests->yahoo-fin) (3.4)
Requirement already satisfied: parse in c:\users\erwin\anaconda3\lib\site-packages
(from requests-html->yahoo-fin) (1.19.0)
Requirement already satisfied: fake-useragent in c:\users\erwin\anaconda3\lib\site-
packages (from requests-html->yahoo-fin) (1.1.3)
Requirement already satisfied: w3lib in c:\users\erwin\anaconda3\lib\site-packages
(from requests-html->yahoo-fin) (1.21.0)
Requirement already satisfied: bs4 in c:\users\erwin\anaconda3\lib\site-packages (f
rom requests-html->yahoo-fin) (0.0.1)
Requirement already satisfied: pyppeteer>=0.0.14 in c:\users\erwin\anaconda3\lib\si
te-packages (from requests-html->yahoo-fin) (1.0.2)
Requirement already satisfied: pyquery in c:\users\erwin\anaconda3\lib\site-package
s (from requests-html->yahoo-fin) (2.0.0)
Requirement already satisfied: websockets<11.0,>=10.0 in c:\users\erwin\anaconda3\l
ib\site-packages (from pyppeteer>=0.0.14->requests-html->yahoo-fin) (10.4)
Requirement already satisfied: pyee<9.0.0,>=8.1.0 in c:\users\erwin\anaconda3\lib\s
ite-packages (from pyppeteer>=0.0.14->requests-html->yahoo-fin) (8.2.2)
Requirement already satisfied: importlib-metadata>=1.4 in c:\users\erwin\anaconda3\
lib\site-packages (from pyppeteer>=0.0.14->requests-html->yahoo-fin) (4.11.3)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in c:\users\erwin\anaconda3\lib\
site-packages (from pyppeteer>=0.0.14->requests-html->yahoo-fin) (4.64.1)
Requirement already satisfied: appdirs<2.0.0,>=1.4.3 in c:\users\erwin\anaconda3\li
b\site-packages (from pyppeteer>=0.0.14->requests-html->yahoo-fin) (1.4.4)
Requirement already satisfied: six>=1.5 in c:\users\erwin\anaconda3\lib\site-packag
es (from python-dateutil>=2.8.1->pandas->yahoo-fin) (1.16.0)
Requirement already satisfied: beautifulsoup4 in c:\users\erwin\anaconda3\lib\site-
packages (from bs4->requests-html->yahoo-fin) (4.11.1)
Requirement already satisfied: importlib-resources>=5.0 in c:\users\erwin\anaconda
3\lib\site-packages (from fake-useragent->requests-html->yahoo-fin) (5.12.0)
Requirement already satisfied: lxml>=2.1 in c:\users\erwin\anaconda3\lib\site-packa
ges (from pyquery->requests-html->yahoo-fin) (4.9.1)
```

```
Requirement already satisfied: cssselect>=1.2.0 in c:\users\erwin\anaconda3\lib\sit
e-packages (from pyquery->requests-html->yahoo-fin) (1.2.0)
Requirement already satisfied: zipp>=0.5 in c:\users\erwin\anaconda3\lib\site-packa
ges (from importlib-metadata>=1.4->pyppeteer>=0.0.14->requests-html->yahoo-fin) (3.
11.0)
Requirement already satisfied: colorama in c:\users\erwin\anaconda3\lib\site-packag
es (from tqdm<5.0.0,>=4.42.1->pyppeteer>=0.0.14->requests-html->yahoo-fin) (0.4.6)
Requirement already satisfied: soupsieve>1.2 in c:\users\erwin\anaconda3\lib\site-p
ackages (from beautifulsoup4->bs4->requests-html->yahoo-fin) (2.3.2.post1)
Requirement already satisfied: scikit-surprise in c:\users\erwin\anaconda3\lib\site
-packages (1.1.3)
Requirement already satisfied: scipy>=1.3.2 in c:\users\erwin\anaconda3\lib\site-pa
ckages (from scikit-surprise) (1.10.0)
Requirement already satisfied: joblib>=1.0.0 in c:\users\erwin\anaconda3\lib\site-p
ackages (from scikit-surprise) (1.1.1)
Requirement already satisfied: numpy>=1.17.3 in c:\users\erwin\anaconda3\lib\site-p
ackages (from scikit-surprise) (1.23.5)
Requirement already satisfied: scrapy in c:\users\erwin\anaconda3\lib\site-packages
(2.8.0)
Requirement already satisfied: cryptography>=3.4.6 in c:\users\erwin\anaconda3\lib\
site-packages (from scrapy) (38.0.4)
Requirement already satisfied: cssselect>=0.9.1 in c:\users\erwin\anaconda3\lib\sit
e-packages (from scrapy) (1.2.0)
Requirement already satisfied: protego>=0.1.15 in c:\users\erwin\anaconda3\lib\site
-packages (from scrapy) (0.1.16)
Requirement already satisfied: setuptools in c:\users\erwin\anaconda3\lib\site-pack
ages (from scrapy) (65.6.3)
Requirement already satisfied: Twisted>=18.9.0 in c:\users\erwin\anaconda3\lib\site
-packages (from scrapy) (22.2.0)
Requirement already satisfied: lxml>=4.3.0 in c:\users\erwin\anaconda3\lib\site-pac
kages (from scrapy) (4.9.1)
Requirement already satisfied: PyDispatcher>=2.0.5 in c:\users\erwin\anaconda3\lib\
site-packages (from scrapy) (2.0.5)
Requirement already satisfied: tldextract in c:\users\erwin\anaconda3\lib\site-pack
ages (from scrapy) (3.2.0)
Requirement already satisfied: parsel>=1.5.0 in c:\users\erwin\anaconda3\lib\site-p
ackages (from scrapy) (1.6.0)
Requirement already satisfied: queuelib>=1.4.2 in c:\users\erwin\anaconda3\lib\site
-packages (from scrapy) (1.5.0)
Requirement already satisfied: zope.interface>=5.1.0 in c:\users\erwin\anaconda3\li
b\site-packages (from scrapy) (5.4.0)
Requirement already satisfied: pyOpenSSL>=21.0.0 in c:\users\erwin\anaconda3\lib\si
te-packages (from scrapy) (22.0.0)
Requirement already satisfied: packaging in c:\users\erwin\anaconda3\lib\site-packa
ges (from scrapy) (22.0)
Requirement already satisfied: itemadapter>=0.1.0 in c:\users\erwin\anaconda3\lib\s
ite-packages (from scrapy) (0.3.0)
Requirement already satisfied: service-identity>=18.1.0 in c:\users\erwin\anaconda
3\lib\site-packages (from scrapy) (18.1.0)
Requirement already satisfied: itemloaders>=1.0.1 in c:\users\erwin\anaconda3\lib\s
ite-packages (from scrapy) (1.0.4)
Requirement already satisfied: w3lib>=1.17.0 in c:\users\erwin\anaconda3\lib\site-p
ackages (from scrapy) (1.21.0)
Requirement already satisfied: cffi>=1.12 in c:\users\erwin\anaconda3\lib\site-pack
ages (from cryptography>=3.4.6->scrapy) (1.15.1)
Requirement already satisfied: jmespath>=0.9.5 in c:\users\erwin\anaconda3\lib\site
-packages (from itemloaders>=1.0.1->scrapy) (0.10.0)
Requirement already satisfied: six>=1.6.0 in c:\users\erwin\anaconda3\lib\site-pack
ages (from parsel>=1.5.0->scrapy) (1.16.0)
```

```
Requirement already satisfied: pyasn1-modules in c:\users\erwin\anaconda3\lib\site-
packages (from service-identity>=18.1.0->scrapy) (0.2.8)
Requirement already satisfied: attrs>=16.0.0 in c:\users\erwin\anaconda3\lib\site-p
ackages (from service-identity>=18.1.0->scrapy) (22.1.0)
Requirement already satisfied: pyasn1 in c:\users\erwin\anaconda3\lib\site-packages
(from service-identity>=18.1.0->scrapy) (0.4.8)
Requirement already satisfied: typing-extensions>=3.6.5 in c:\users\erwin\anaconda
3\lib\site-packages (from Twisted>=18.9.0->scrapy) (4.4.0)
Requirement already satisfied: incremental>=21.3.0 in c:\users\erwin\anaconda3\lib\
site-packages (from Twisted>=18.9.0->scrapy) (21.3.0)
Requirement already satisfied: hyperlink>=17.1.1 in c:\users\erwin\anaconda3\lib\si
te-packages (from Twisted>=18.9.0->scrapy) (21.0.0)
Requirement already satisfied: constantly>=15.1 in c:\users\erwin\anaconda3\lib\sit
e-packages (from Twisted>=18.9.0->scrapy) (15.1.0)
Requirement already satisfied: twisted-iocpsupport<2,>=1.0.2 in c:\users\erwin\anac
onda3\lib\site-packages (from Twisted>=18.9.0->scrapy) (1.0.2)
Requirement already satisfied: Automat>=0.8.0 in c:\users\erwin\anaconda3\lib\site-
packages (from Twisted>=18.9.0->scrapy) (20.2.0)
Requirement already satisfied: requests-file>=1.4 in c:\users\erwin\anaconda3\lib\s
ite-packages (from tldextract->scrapy) (1.5.1)
Requirement already satisfied: requests>=2.1.0 in c:\users\erwin\anaconda3\lib\site
-packages (from tldextract->scrapy) (2.28.1)
Requirement already satisfied: filelock>=3.0.8 in c:\users\erwin\anaconda3\lib\site
-packages (from tldextract->scrapy) (3.9.0)
Requirement already satisfied: idna in c:\users\erwin\anaconda3\lib\site-packages
(from tldextract->scrapy) (3.4)
Requirement already satisfied: pycparser in c:\users\erwin\anaconda3\lib\site-packa
ges (from cffi>=1.12->cryptography>=3.4.6->scrapy) (2.21)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\erwin\anaconda3\li
b\site-packages (from requests>=2.1.0->tldextract->scrapy) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\erwin\anaconda3\lib\s
ite-packages (from requests>=2.1.0->tldextract->scrapy) (2022.12.7)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\erwin\anaconda
3\lib\site-packages (from requests>=2.1.0->tldextract->scrapy) (2.0.4)
Requirement already satisfied: newspaper3k in c:\users\erwin\anaconda3\lib\site-pac
kages (0.2.8)
Requirement already satisfied: beautifulsoup4>=4.4.1 in c:\users\erwin\anaconda3\li
b\site-packages (from newspaper3k) (4.11.1)
Requirement already satisfied: feedparser>=5.2.1 in c:\users\erwin\anaconda3\lib\si
te-packages (from newspaper3k) (6.0.10)
Requirement already satisfied: lxml>=3.6.0 in c:\users\erwin\anaconda3\lib\site-pac
kages (from newspaper3k) (4.9.1)
Requirement already satisfied: tinysegmenter==0.3 in c:\users\erwin\anaconda3\lib\s
ite-packages (from newspaper3k) (0.3)
Requirement already satisfied: tldextract>=2.0.1 in c:\users\erwin\anaconda3\lib\si
te-packages (from newspaper3k) (3.2.0)
Requirement already satisfied: Pillow>=3.3.0 in c:\users\erwin\anaconda3\lib\site-p
ackages (from newspaper3k) (9.3.0)
Requirement already satisfied: jieba3k>=0.35.1 in c:\users\erwin\anaconda3\lib\site
-packages (from newspaper3k) (0.35.1)
Requirement already satisfied: cssselect>=0.9.2 in c:\users\erwin\anaconda3\lib\sit
e-packages (from newspaper3k) (1.2.0)
Requirement already satisfied: feedfinder2>=0.0.4 in c:\users\erwin\anaconda3\lib\s
ite-packages (from newspaper3k) (0.0.4)
Requirement already satisfied: python-dateutil>=2.5.3 in c:\users\erwin\anaconda3\l
ib\site-packages (from newspaper3k) (2.8.2)
Requirement already satisfied: requests>=2.10.0 in c:\users\erwin\anaconda3\lib\sit
e-packages (from newspaper3k) (2.28.1)
Requirement already satisfied: PyYAML>=3.11 in c:\users\erwin\anaconda3\lib\site-pa
```

```
ckages (from newspaper3k) (6.0)
Requirement already satisfied: nltk>=3.2.1 in c:\users\erwin\anaconda3\lib\site-pac
kages (from newspaper3k) (3.7)
Requirement already satisfied: soupsieve>1.2 in c:\users\erwin\anaconda3\lib\site-p
ackages (from beautifulsoup4>=4.4.1->newspaper3k) (2.3.2.post1)
Requirement already satisfied: six in c:\users\erwin\anaconda3\lib\site-packages (f
rom feedfinder2>=0.0.4->newspaper3k) (1.16.0)
Requirement already satisfied: sgmllib3k in c:\users\erwin\anaconda3\lib\site-packa
ges (from feedparser>=5.2.1->newspaper3k) (1.0.0)
Requirement already satisfied: tqdm in c:\users\erwin\anaconda3\lib\site-packages
(from nltk>=3.2.1->newspaper3k) (4.64.1)
Requirement already satisfied: click in c:\users\erwin\anaconda3\lib\site-packages
(from nltk>=3.2.1->newspaper3k) (8.0.4)
Requirement already satisfied: regex>=2021.8.3 in c:\users\erwin\anaconda3\lib\site
-packages (from nltk>=3.2.1->newspaper3k) (2022.3.15)
Requirement already satisfied: joblib in c:\users\erwin\anaconda3\lib\site-packages
(from nltk>=3.2.1->newspaper3k) (1.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\erwin\anaconda3\li
b\site-packages (from requests>=2.10.0->newspaper3k) (1.26.14)
Requirement already satisfied: idna<4,>=2.5 in c:\users\erwin\anaconda3\lib\site-pa
ckages (from requests>=2.10.0->newspaper3k) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\erwin\anaconda3\lib\s
ite-packages (from requests>=2.10.0->newspaper3k) (2022.12.7)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\erwin\anaconda
3\lib\site-packages (from requests>=2.10.0->newspaper3k) (2.0.4)
Requirement already satisfied: filelock>=3.0.8 in c:\users\erwin\anaconda3\lib\site
-packages (from tldextract>=2.0.1->newspaper3k) (3.9.0)
Requirement already satisfied: requests-file>=1.4 in c:\users\erwin\anaconda3\lib\s
ite-packages (from tldextract>=2.0.1->newspaper3k) (1.5.1)
Requirement already satisfied: colorama in c:\users\erwin\anaconda3\lib\site-packag
es (from click->nltk>=3.2.1->newspaper3k) (0.4.6)
Requirement already satisfied: GoogleNews in c:\users\erwin\anaconda3\lib\site-pack
ages (1.6.7)
Requirement already satisfied: beautifulsoup4 in c:\users\erwin\anaconda3\lib\site-
packages (from GoogleNews) (4.11.1)
Requirement already satisfied: dateparser in c:\users\erwin\anaconda3\lib\site-pack
ages (from GoogleNews) (1.1.8)
Requirement already satisfied: python-dateutil in c:\users\erwin\anaconda3\lib\site
-packages (from GoogleNews) (2.8.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\erwin\anaconda3\lib\site-p
ackages (from beautifulsoup4->GoogleNews) (2.3.2.post1)
Requirement already satisfied: pytz in c:\users\erwin\anaconda3\lib\site-packages
(from dateparser->GoogleNews) (2022.7)
Requirement already satisfied: regex!=2019.02.19,!=2021.8.27 in c:\users\erwin\anac
onda3\lib\site-packages (from dateparser->GoogleNews) (2022.3.15)
Requirement already satisfied: tzlocal in c:\users\erwin\anaconda3\lib\site-package
s (from dateparser->GoogleNews) (4.3)
Requirement already satisfied: six>=1.5 in c:\users\erwin\anaconda3\lib\site-packag
es (from python-dateutil->GoogleNews) (1.16.0)
Requirement already satisfied: tzdata in c:\users\erwin\anaconda3\lib\site-packages
(from tzlocal->dateparser->GoogleNews) (2023.3)
Requirement already satisfied: pytz-deprecation-shim in c:\users\erwin\anaconda3\li
b\site-packages (from tzlocal->dateparser->GoogleNews) (0.1.0.post0)
```

Problem 1.

Upload the sn_ids.csv and do the following.

- Use the Pagerank algorithm to find the rank of all the ids. Then filter the dictionary in decreasing order. Round the PageRank values to 3 decimal places and create a data frame with two columns: "ids" and "PageRank" values
- Find a list of ids using the data frame of ids and PageRank in part a with the top 5 PageRank values and use the list to filter the original sn_ids.csv data set in both columns 'id_1' and 'id_2'. Finally, plot the network graph of the filtered data using following details: figsize(50, 50), a different color for "id_1" and "id_2".
- Repeat part a using the HITS algorithm.
- Repeat part b using the data from part c.

In [ ]:
```
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure, text
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
import sklearn.metrics as metrics
import seaborn as sns
from sklearn.linear_model import LinearRegression
from dmba import plotDecisionTree
from sklearn.neighbors import KNeighborsRegressor
from sklearn import neighbors
import networkx as nx
#pd.set_option('display.max_rows', None)
#pd.set_option('display.max_columns', None)

sn_ids = pd.read_csv('sn_ids.csv')
print(sn_ids.head())

print('\n', sn_ids.shape)
```

```
   id_1   id_2
0     0  23977
1     1  34526
2     1   2370
3     1  14683
4     1  29982

 (289003, 2)
```

In [ ]:
```
def draw_graph(G, show_names=False, node_size=1, font_size=10, edge_width=0.5):

    import numpy as np
    import networkx as nx

    from IPython.display import SVG
    from sknetwork.visualization import svg_graph
    from sknetwork.data import Bunch
    from sknetwork.ranking import PageRank

    adjacency = nx.to_scipy_sparse_matrix(G, nodelist=None, dtype=None, weight='wei

    names = np.array(list(G.nodes()))

    graph = Bunch()
    graph.adjacency = adjacency
    graph.names = np.array(names)

    pagerank = PageRank()

    scores = pagerank.fit_transform(adjacency)

    if show_names:

        image = svg_graph(graph.adjacency, font_size=font_size, node_size=node_size

    else:

        image = svg_graph(graph.adjacency, node_size=node_size, width=700, height=5

    return SVG(image)
```

In [ ]:
```
G = nx.from_pandas_edgelist(sn_ids, 'id_1', 'id_2')
##plt.figure(figsize=(30,30))
#nx.draw_networkx(G)
#plt.show()
```

In [ ]:
```
pagerank = nx.pagerank(G)
pagerank_df = pd.DataFrame(pagerank, index=[0]).T
pagerank_df.columns = ['pagerank']
pagerank_df.sort_values('pagerank', inplace=True, ascending=False)

pagerank_df['pagerank'] = round(pagerank_df['pagerank'], 3)

pagerank_df.head(10)
```

Out[ ]:

	pagerank
31890	0.017
27803	0.011
35773	0.005
19222	0.005
13638	0.004
36652	0.004
18163	0.003
9051	0.003
10001	0.003
35008	0.002

In [ ]:
```
page_df = pd.DataFrame({'ids': pagerank_df.index, 'pagerank': pagerank_df['pagerank
page_df.reset_index
page_df.head()
```

Out[ ]:

	ids	pagerank
31890	31890	0.017
27803	27803	0.011
35773	35773	0.005
19222	19222	0.005
13638	13638	0.004

In [ ]:
```
#Find a list of ids using the data frame of ids and PageRank in part a with the top
#sn_ids.csv data set in both columns 'id_1' and 'id_2'. Finally, plot the network g
#a different color for "id_1" and "id_2".

#top5id = page_df.ids.value_counts().nlargest(5)

#top5 = page_df.pagerank.value_counts().nlargest(5).to_frame().reset_index()
top5 = page_df.pagerank.value_counts().nlargest(5).tolist()
top5
```

Out[ ]:
```
[37610, 71, 10, 3, 2]
```

In [ ]:
```
#id1counts = sn_ids.id_1.value_counts().to_frame().reset_index()
#id1counts.columns = ["id_1", "connections"]
#print(id1counts)

#id1mostfrequent = id1counts['connections']
#filter by top5
#frequentid1 = list(top5)
#print(id1mostfrequent.shape)

print(top5)

filteredby_id1 = sn_ids[sn_ids["id_1"].isin(top5)]
print(filteredby_id1.shape)
```

```
[37610, 71, 10, 3, 2]
(89, 2)
```

In [ ]:
```
#id2counts = sn_ids.id_2.value_counts().to_frame().reset_index()
#id2counts.columns = ["id_2", "connections"]
#print(id2counts)

#id2mostfrequent = id2counts['connections']
#filter by top5
#frequentid2 = list(top5)
#print(id2mostfrequent.shape)

print(top5)
filteredby_id2 = sn_ids[sn_ids["id_2"].isin(top5)]
print(filteredby_id2.head)
print(filteredby_id2.shape)
```

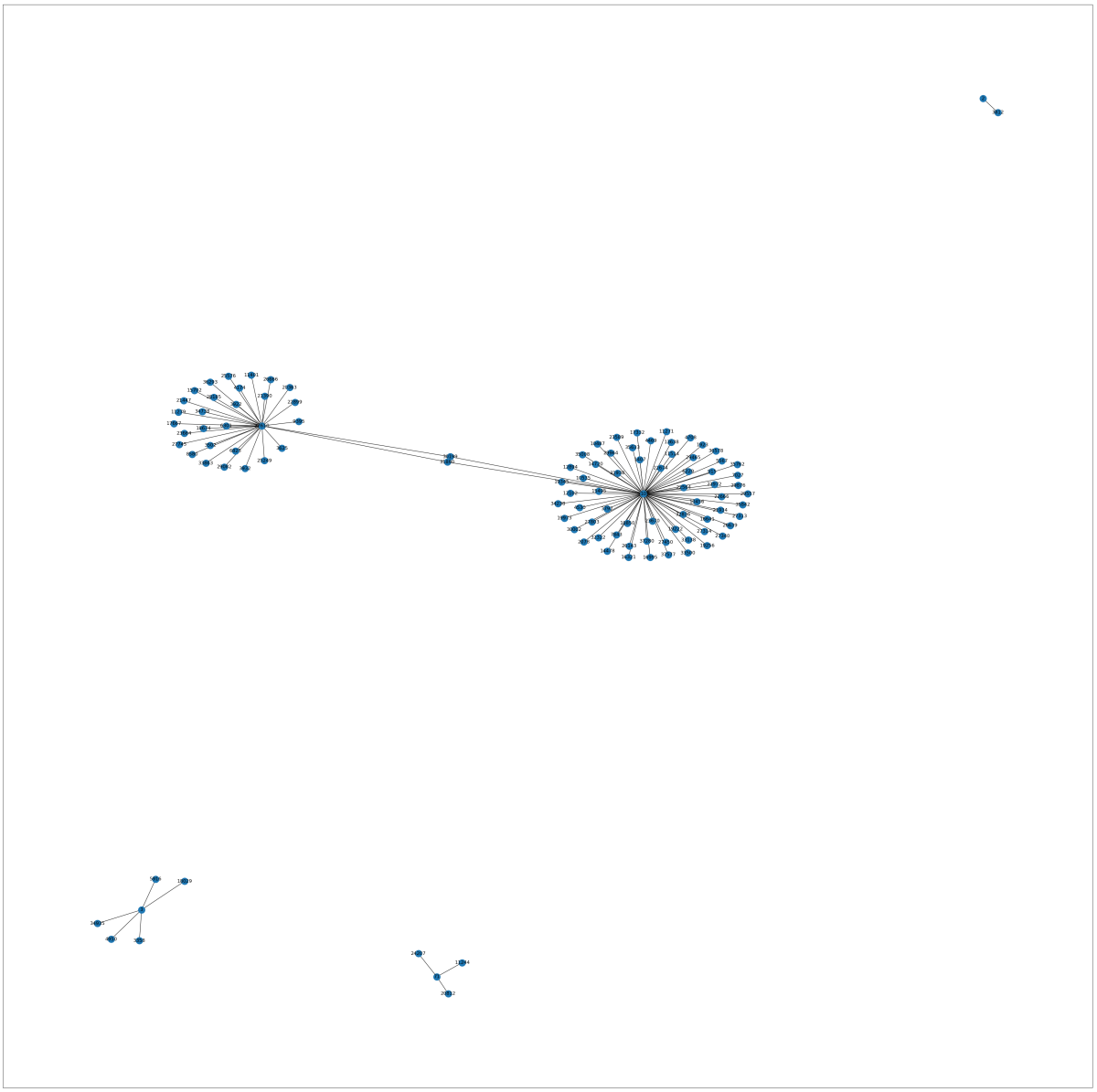
```
[37610, 71, 10, 3, 2]
<bound method NDFrame.head of          id_1   id_2
27275    9395  37610
52714    3502  37610
54031    3625  37610
57542    3922  37610
57648    3932  37610
60903    4174  37610
86039   25576  37610
86405    6301  37610
93462    6825  37610
115206   8580  37610
147564  11279  37610
148486  11401  37610
157718  20363  37610
188577  27745  37610
188960  15702  37610
198231  33863  37610>
(16, 2)
```

In [ ]:
```
filtered_data = pd.concat([filteredby_id1, filteredby_id2], axis = 0)
filtered_data.head()
```

Out[ ]:

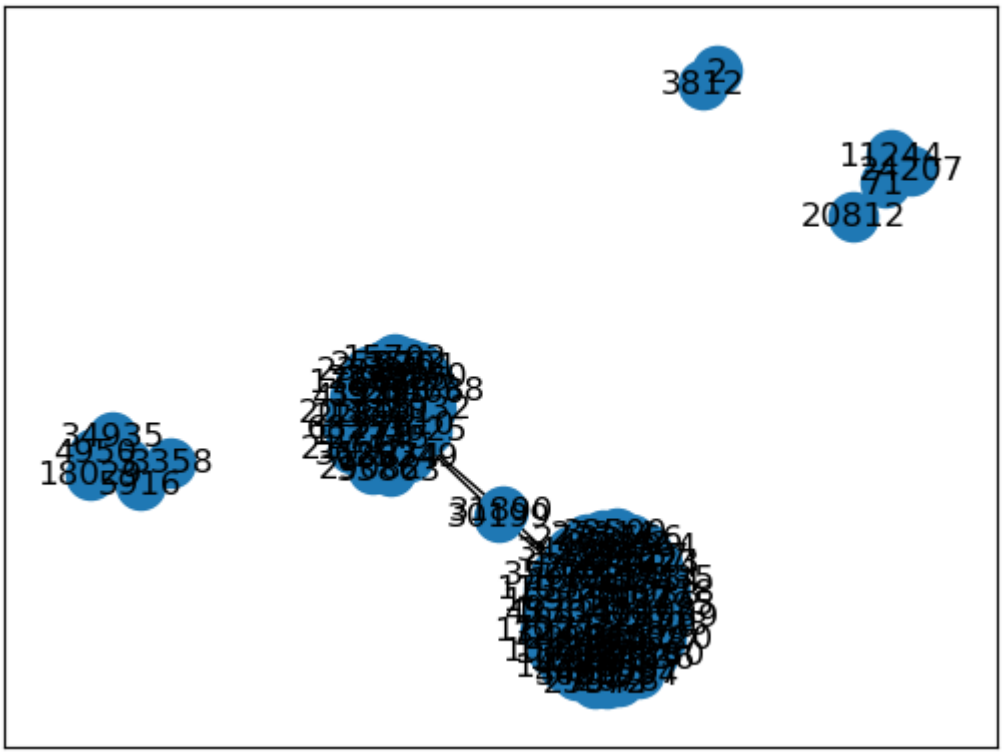
	id_1	id_2
11	3	4950
12	3	18029
13	3	3358
14	3	34935
15	3	5916

In [ ]:
```
#looking at both of them id 31890 has the highest number of connections.

G = nx.from_pandas_edgelist(filtered_data, 'id_1', 'id_2')
fig, ax = plt.subplots(figsize=(50,50))
nx.draw_networkx(G)
plt.show()
```



In [ ]:
```
#now instead of the page_rank algorithm, do the same with the HITS algorithm

H = G.to_directed()
hits, authorities = nx.hits(H, max_iter = 50, normalized = True)
hits

#nx.spring_layout is a popular algorithm to use when working with HITS. it uses a f
pos = nx.spring_layout(G)
nx.draw_networkx(G)
plt.show()
#hit_df = pd.DataFrame.from_dict(hits)
```

```
C:\Users\erwin\anaconda3\lib\site-packages\networkx\algorithms\link_analysis\hits_a
lg.py:78: FutureWarning: adjacency_matrix will return a scipy.sparse array instead
of a matrix in Networkx 3.0.
  A = nx.adjacency_matrix(G, nodelist=list(G), dtype=float)
```



Problem 2.

Do the following using the Yahoo Finance package.

- Download the 20 ticker symbols (listed below) data and create a data frame with 3 columns: Ticker Symbol, Top 20 institutional holders of the 20 tickers, and how much the holders hold in dollar amount.
- Create a network graph with the holder as a source and the ticker symbol as a target. Label the vertices with their respective labels.
- Add different colors for the source and the target, and change the size of the edge using the normalized holding amount. And change the size of the ticker symbols by their degrees (scale if required). See : http://andrewtrick.com/stormlight_network.html
- Change at least something to the graph in part c to make the graph better (in your own eyes)

In [ ]:
```
top20_tickers = ["AAPL", "AMZN", "MSFT", "GOOG", "GOOGL", "META", "TSLA", "NVDA", "
                 "UNH", "HD", "MA", "BAC", "DIS", "PYPL", "NFLX", "ADBE"]
```

In [ ]:
```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import sklearn.metrics as metrics
from sklearn.feature_selection import SelectKBest, f_regression
import numpy as np
from sklearn.model_selection import cross_val_score
import yfinance as yf
from yahoofinancials import YahooFinancials



def yahoo(STRING):

    STRING_hold = STRING.institutional_holders
    #print(STRING_hold.head())
    symbol_value = STRING.info['symbol']
    #print(symbol_value)

    h = STRING_hold.Holder.values.tolist()
    val = STRING_hold.Value.values.tolist()
    sym = [symbol_value]*len(val)

    temp_df = pd.DataFrame(columns = ['SYMBOL', 'HOLDER', 'CASH'])
    temp_df['SYMBOL'] = sym
    temp_df['HOLDER'] = h
    temp_df['CASH'] = val

    #df = pd.concat([df, temp_df])

    return(temp_df)
```

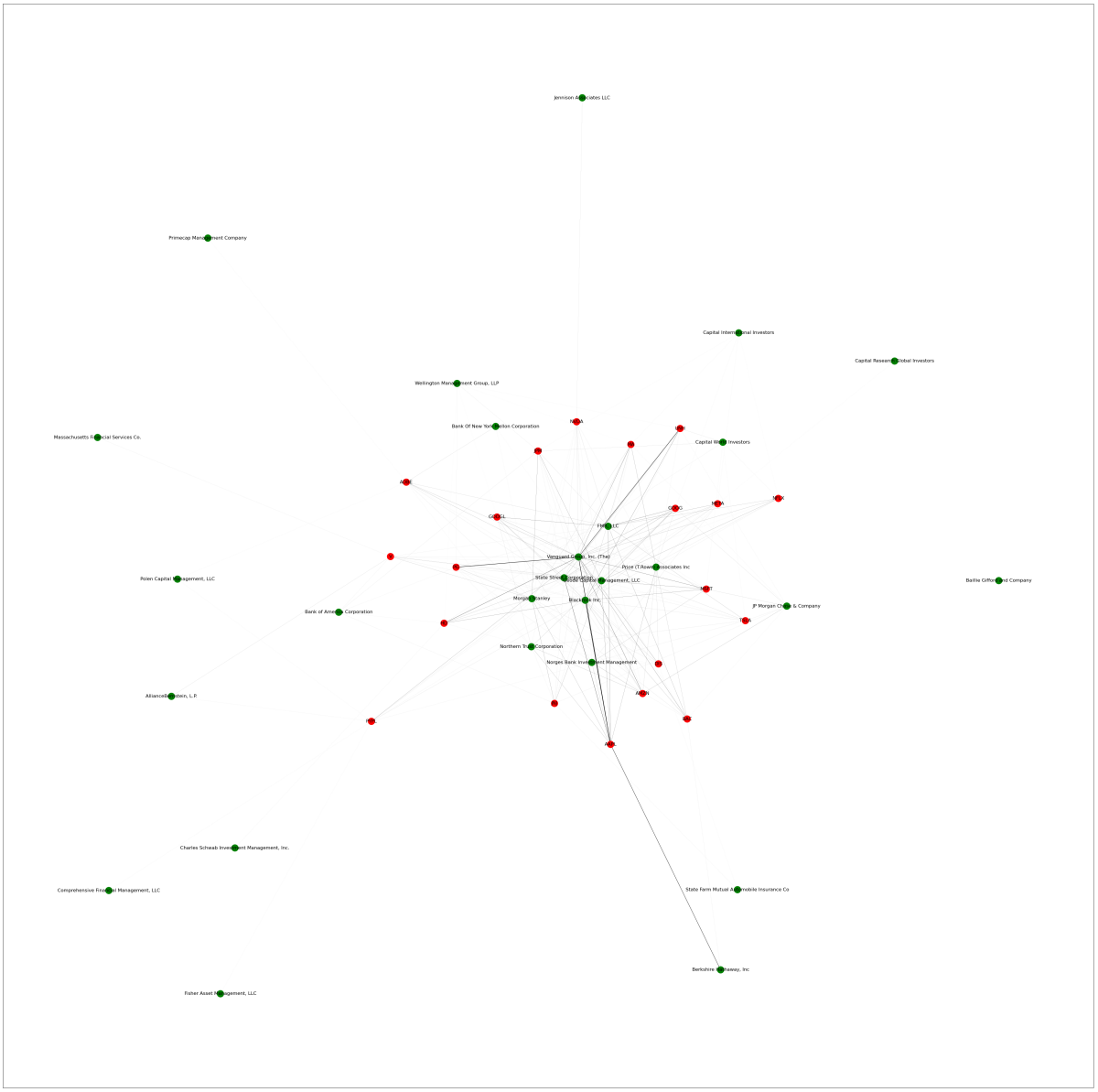
In [ ]:
```
stock_df = pd.DataFrame(columns = ['SYMBOL', 'HOLDER', 'CASH'])

for i in top20_tickers:
    STRING = yf.Ticker(i)
    df = yahoo(STRING)
    stock_df =pd.concat([stock_df, df])

stock_df
```

Out[ ]:

	SYMBOL	HOLDER	CASH
0	AAPL	Vanguard Group, Inc. (The)	209326301081
1	AAPL	Blackrock Inc.	168543149157
2	AAPL	Berkshire Hathaway, Inc	146587495100
3	AAPL	State Street Corporation	96103765419
4	AAPL	FMR, LLC	52593554661
...	...	...	...
5	ADBE	Geode Capital Management, LLC	3451690543
6	ADBE	Bank Of New York Mellon Corporation	2645951491
7	ADBE	Primecap Management Company	2302407783
8	ADBE	Polen Capital Management, LLC	2276464605
9	ADBE	Northern Trust Corporation	2008211191

200 rows × 3 columns

In [ ]:
```
#Create a network graph with the holder as a source and the ticker symbol as a targ
#Label the vertices with their respective labels.
#Add different colors for the source and the target, and change the size of the edg
#And change the size of the ticker symbols by their degrees (scale if required)

G = nx.from_pandas_edgelist(stock_df, 'SYMBOL', 'HOLDER')
node_colors = ['red' if node in set(stock_df['SYMBOL']) else 'green' for node in G.

norm = (stock_df['CASH'] - stock_df['CASH'].min()) / (stock_df['CASH'].max() - stoc

degrees = dict(G.degree)
#font_sizes = {node: (degree + 10) * 100 for node, degree in degrees.items()}

fig, ax = plt.subplots(figsize=(50,50))
nx.draw_networkx(G,width = norm, node_color=node_colors)


pos = nx.spring_layout(G)
#x.draw_networkx_labels(G, pos,font_family='Arial')


plt.show()
```



Problem 3.

Do the following.

- Use web scrapping techniques to find 25 news articles related to each of the 20 stocks listed in the previous problem and create a data frame with the details: Date, Journalist, Article
- Perform a sentiment analysis using textblob in the data frame in part a and order the dataframe by sentiments.
- Repeat part b using the Naive Bayes sentiment analysis.
- Based on parts b and c, which 10 stocks will you choose for your portfolio?

In [ ]:
```
#
```

In [ ]:
```
import json
import requests
from bs4 import BeautifulSoup
from scrapy.selector import Selector
from scrapy.http import HtmlResponse
from scrapy.crawler import CrawlerProcess
```

In [ ]:
```
import newspaper
from newspaper import Article
from tqdm import tqdm

top20_tickers = ["AAPL", "AMZN", "MSFT", "GOOG", "GOOGL", "META", "TSLA", "NVDA", "
                 "UNH", "HD", "MA", "BAC", "DIS", "PYPL", "NFLX", "ADBE"]

def storydf(domain):
    paper = newspaper.build(domain, memoize_articles=False)
    urls = paper.article_urls()
    titles = []
    texts = []
    languages = []
    keywords = []
    for url in tqdm(urls):
        article = Article(url)
        article.download()
        article.parse()
        article.nlp()
        titles.append(article.title)
        texts.append(article.text)
        languages.append(article.meta_lang)
        keywords.append(article.keywords)
    df = pd.DataFrame({'urls':urls, 'title':titles, 'text':texts, 'lang':languages,
    return df

domain = 'https://finance.yahoo.com'
df = storydf(domain)
df.head()
```

```
 59%|
| 130/220 [02:14<00:52,  1.70it/s]
```

In [ ]:
```
mask = df['text'].apply(lambda x: any(item in x for item in top20_tickers))
filtered_df = df[mask]
filtered_df
```

In [ ]:
```
from textblob import TextBlob
def get_sentiment(text):
    blob = TextBlob(text)
    sentiment = blob.sentiment
    return sentiment.polarity, sentiment.subjectivity


df['sentiment_polarity'], df['sentiment_subjectivity'] = zip(*df['text'].apply(get_

df
```

In [ ]:
```
df3  = df.sort_values(by = ['sentiment_polarity'], ascending=False)

df3.head(10)
```

Problem 4:

Upload the ratings.csv data set with movieId, userId, and the rating as columns and do the following.

- Remove all the movies which have been rated < 100 times. Then use the filtered data to create a utility matrix.
- Change the utility matrix to a data frame and determine how many ratings will be missing. Calculate the percentage of the entries that are missing from the sparse matrix.
- Make an SVD-based Collaborative filtering recommender system to recommend movies to the user. What is the RMSE?
- Find the userId who has rated a movie for a movieId in the data and use the model to make the top 5 recommendations.

In [ ]:
```
import surprise
from surprise import Dataset, Reader
from surprise import SVD
from surprise.model_selection import train_test_split
from surprise import accuracy
```

In [ ]:
```
rat_ids = pd.read_csv('ratings.csv')
print(rat_ids.head())

print('\n',"Null value count: ",rat_ids.isna().sum())

print('\n', rat_ids.shape)
```

In [ ]:
```

```

In [ ]:
```
import collections
frequency = collections.Counter(rat_ids['movieId'])
#print(dict(frequency))


print(rat_ids['movieId'].value_counts())
print(rat_ids.head())
```

In [ ]:
```
value_counts = rat_ids['movieId'].value_counts()
filtered_rat = rat_ids[rat_ids['movieId'].isin(value_counts.index[value_counts.gt(9

print('\n',filtered_rat.shape)
print('\n',filtered_rat.head())
print('\n',filtered_rat['movieId'].value_counts())
```

In [ ]:
```
example = filtered_rat.sample(n=20)
utility_matrix = filtered_rat.pivot_table( index = 'userId', columns= 'movieId', va
utility_matrix
```

In [ ]:
```
#Change the utility matrix to a data frame and determine how many ratings will be m
#Calculate the percentage of the entries that are missing from the sparse matrix.



missing_values = 0
frequent_users = list(utility_matrix.to_records())
x = pd.DataFrame.from_records(frequent_users)

x.head()

print(x.isna().sum().sum())
print(x.size)

null_per = (x.isna().sum().sum()/x.size) * 100
print(null_per, 'percent')
```

In [ ]:
```
#Make an SVD-based Collaborative filtering recommender system to recommend movies t
reader = Reader(rating_scale=(0, 5))
data = Dataset.load_from_df(filtered_rat[['userId', 'movieId', 'rating']], reader)
trainset, testset = train_test_split(data, test_size=0.2)
rs_svd = SVD(n_epochs=20, lr_all=0.005, reg_all=0.2)
rs_svd.fit(trainset)
```

In [ ]:
```
predictions = rs_svd.test(testset)
accuracy.rmse(predictions)
```

In [ ]:
```
from surprise.model_selection import cross_validate
cross_validate(rs_svd, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
```

In [ ]:
```
testset[:10]
```

In [ ]:
```
testdf = pd.DataFrame(testset, columns=['userId', 'movieId', 'rating'])
test_umatrix = testdf.pivot(index='userId', columns='movieId', values='rating')
pd.DataFrame(test_umatrix).head()
```

In [ ]:
```
predictions[0]
```

In [ ]:
```
filtered_rat.info()
```

In [ ]:
```
userId = 1
movieId = 47
prediction = rs_svd.predict(userId, movieId)
print(prediction.est)
```

In [ ]:
```
#Find the userId who has rated a movie for a movieId in the data and use the model

movies = filtered_rat.loc[filtered_rat["userId"] == 10, "movieId"]
#print(movies)
top_film = []

for i in movies:
    prediction = rs_svd.predict(10, i)
    #print(prediction)
    top_film.append(prediction)


top = pd.DataFrame(top_film)
print(top.sort_values(by = ['est'], ascending=False))
```

By putting the predictions into a dataframe and sorting my 'est' we can see the predicted most popular films for user 10. The top 5 movie ids are:

912, 58559, 2959, 2571, and 356

In [ ]:
```

```