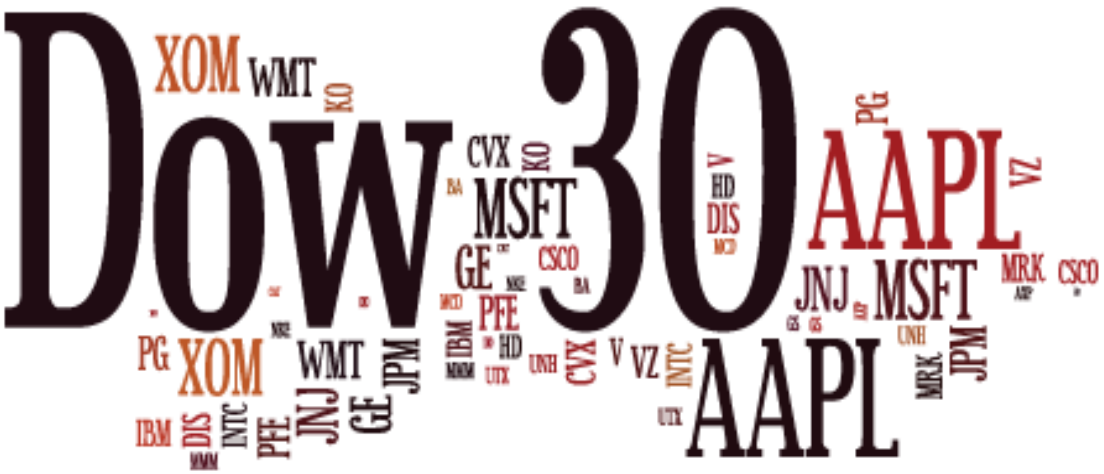
ETL of DOW 30 Stock Data



Tanvir Khan
James Ye
Fabienne Zumbuehl

May 26th 2020

Instructor: Alexis Baird

Trilogy Education Services

UC Berkeley Extension

Table of Contents

1. INTRODUCTION

The focus of this ETL Project lies on the Dow 30 stocks. The Dow 30 is a stock market index that measures the stock performance of 30 large companies listed on stock exchanges in the US. This index has been chosen due to the fact that it is one of the most observed indicators for stock market performance, as well as a good indicator of the performance of the U.S. economy.

Data on the Dow 30 stocks is available from various websites such as CNBC.com, Yahoo Finance and also from Dividend.com, as well as from other sources. The goal of this project was to retrieve data on the Dow 30 stocks, i.e. on the price, on the absolute change, on the percentage change, the daily low and high, as well as on the previous close. Furthermore, data on the ESG ratings[1] of the Dow 30 stocks, as well as the Dividends from the Dow 30 stocks was aimed to attain.

Note: Since this project scraped real time data from stock prices, the data retrieved may change across time. The time of scraping of this project was done on May 26th 2020, 06:18pm PST.

2. EXTRACTION

The extraction part started with scouting for sources that provided the data that we looked for. The data we aimed for could be found on three different sources. Hence, we had to extract the data from those three different sources, which are listed in the following.

Data Sources:

- CNBC.com
  https://www.cnbc.com/dow-30/

- Dividend.com
  https://www.dividend.com/dividend-stocks/dow-30-dividend-stocks/#tm=3-dow-30&r=Webpage%231326&f_4=true&only=meta%2Cdata%2Cthead

- Yahoo Finance
  https://finance.yahoo.com/

We decided that we could extract the data from the above mentioned sources through web scraping. However, before we started the process of data scraping, we identified the data that is needed from each of the data sources. The following paragraphs elaborate on the data that was extracted from the respective website and the process of extraction.

[1] ESG ratings rate companies along the dimensions environmental, social and governance, taking into account other performance indicators than solely the financial performance.

2.1 Extract Price Data from CNBC

From the CNBC website, we found data on the Dow 30 stock price, as well as the stocks absolute change, percentage change, daily low and high, as well as previous close price. We followed the following four steps to extract the data.

a. Dow 30 stock price data was extracted from CNBC link: https://www.cnbc.com/dow-30/.
b. This CNBC link provided a table which could be converted into pandas dataframe.
c. For the operation in step 2, we used splinter's browser to visit the CNBC page and
d. Ultimately, we use pandas read_html() method to get the table.

2.2 Extract Dividend Data from Dividends.com

From the Dividends.com website we aimed to retrieve data on the dividends of the Dow 30 stocks. The following three steps have been followed to extract this data.

a. We extracted dividend data from dividends.com with the following link: https://www.dividend.com/dividend-stocks/dow-30-dividend-stocks/.
b. Due to the fact that the website loads quickly, we were able to use pandas to read it directly.
c. This process was different from extracting data from CNBC website. The CNBC website took a few seconds to load, which forced us to use splinter's browser to visit it. We had to apply the "sleep function" for a few seconds before getting CNBC's HTML source code. In contrast, for dividends.com, there is no need to apply the «sleep function» since pandas can read the URL and extract the dividend table directly.

2.3 Extract ESG Rating from Yahoo Finance

Yahoo Finance provides lots of information on various stocks. However, we were only interested in scraping the ticker symbols and ESG score for the Dow 30 stocks. ESG risk ratings assess the degree to which a company's enterprise business value is at risk driven by environmental, social and governance issues. There are different organizations that assess the ESG rating using different methodologies. Yahoo Finance applies the analysis from Sustainanalytics (see https://www.sustainalytics.com/). This rating employs a two-dimensional framework that combines an assessment of a company's exposure to industry specific material ESG issues with an assessment of how well the company is managing those issues. The final ESG risk rating scores are a measure of unmanaged risk on an absolute scale of 0-100, with a lower score signaling less unmanaged ESG risk. In short, the higher the score, the better the ESG risks are managed by a company. Such ESG scores have gained recent popularity as an indicator that is taken into account when determining an investment portfolio.

The following four steps have been followed to extract the data from Yahoo Finance.

a. To get to the ESG Score, we visited finance.yahoo.com and found the URL to get the ESG score. The pattern is portrayed in the following URL. https://finance.yahoo.com/quote/{stock_symbol}/sustainability?p={stock_symbol}.
b. Since we can get a list of stock symbols, we could loop through the stock symbols list to get the ESG score for each stock.
c. Upon inspecting Yahoo's sustainability page, we found that the ESG rating element has the following CSS class «Fz(36px) Fw(600) D(ib) Mend(5px)». Based on this class information, we were able to extract the ESG score using Splinter's browser.find_by_css() method.
d. Using a «for loop», we were able to get a list of ESG scores. Details of this operation can be found in our jupyter notebook (etl-project.ipynb)

3. TRANSFORMATION

The next step after the extraction of the data, was to transform it. This means cleaning up null values, changing column names, deleting columns or rows that are not needed, changing data types, etc. The following paragraphs elaborate on this process for the CNBC and the Dividend data, as well as for the ESG data from Yahoo Finance.

3.1 Transform Price Data from CNBC and Dividend Data from Dividend

The following ten steps provide an overview on how the data from CNBC.com and the data from Dividend.com has been transformed.

a. For transferring/parsing the extracted data, the python library and Pandas functions in Jupyter Notebook were used. For example, like pandas to read (pd.read) method was used to parse two HTML files.
b. Used the pandas library to transform and clean our data. Using the pandas library, we applied the «drop method» to delete rows and columns, in order to get reduced data. For example, we used the «drop method» to delete the DOW (Dow Inc.) from the price data and DWDP (Dow DuPoint Inc.) from dividend data.
c. The «replace method» was used to clean the data. For example, the «replace method» was used to delete the «$» and «%» characters from the columns.
d. We also applied the «delete method» to delete the column from the dividend data frame. For example, the column «Unnamed: 0» was removed.
e. The «rename method» was used to rename all columns.
f. The «astype» method was used to change the datatype of the columns. For example, the «astype(float)» method was used to convert strings to floats.
g. We also applied the Pandas library to convert data tables into Data Frames.

h. We converted all columns names to lower case for future use and to have a uniform labeling to other tables.
i. The copy method was used, so changes in the new dataframe are independent and didn't change anything in original dataframe one.
j. Finally, all the clean and relevant data were stored into either the "price_data_df" or the "dividend_df" Data Frames.

3.2 Transform ESG rating from Yahoo Finance

The following six steps provide an overview on how the ESG data from YahooFinance.com has been transformed.

a. We applied a «for loop» to collect ticker symbols and ESG Score and stored information to corresponding lists.
b. For setting values, df['column'] = series was used. For example, price_data_df['esg_rating'] = egs_rating_list
c. We converted the ESG Rating column names to lowercase.
d. We used the «pd.to_numeric() method», which resulted in the «esg_rating» column values to be casted as integers.
e. And this pandas function combines the «Price dataframe» and «egs_rating_list».
f. Finally, we stored this new data frame as «price_data_df»

price data from cnbc.com

In [3]:
```
price_url = "https://www.cnbc.com/dow-30/"
```

In [4]:
```
browser.visit(price_url)
time.sleep(5)
price_html = browser.html
```

In [5]:
```
price_data_tables = pd.read_html(price_html)
print("Number of tables: ", len(price_data_tables))
price_data_df = price_data_tables[0]
#price_data_df.set_index('id', inplace=True)

# drop ticker 'Dow' because their substainability can not be found in Yahoo Finance
price_data_df.drop(index=29, inplace=True)

price_data_df = price_data_df[["SYMBOL","NAME", "PRICE", "LOW", "HIGH", "PREVIOUS C
price_data_df.rename(columns = {"SYMBOL":"symbol","NAME":"company", "PRICE":"price"
price_data_df
```

```
Number of tables:  1
```

Out[5]:

	symbol	company	price	low	high	previous_close
0	AXP	American Express Co	89.83	89.125	91.430	89.83
1	AAPL	Apple Inc	316.85	315.870	320.890	316.85
2	BA	Boeing Co	139.00	136.151	144.239	139.00
3	CAT	Caterpillar Inc	114.06	113.820	115.680	114.06
4	CSCO	Cisco Systems Inc	44.64	44.555	45.655	44.64
5	CVX	Chevron Corp	92.04	91.280	93.370	92.04
6	XOM	Exxon Mobil Corp	44.56	44.390	45.790	44.56
7	GS	Goldman Sachs Group Inc	180.10	177.680	181.170	180.10
8	HD	Home Depot Inc	240.88	235.790	241.180	240.88
9	IBM	International Business Machines Corp	119.12	118.970	121.720	119.12

Figure 1: Extraction and Transformation of the price data from CNBC.com

ESG Rating data from finance.yahoo.com

```
print("ticker", "ESG Score")
egs_rating_list = []
first = True
for stock_symbol in dow29_symbols:
    sustainability_url = f"https://finance.yahoo.com/quote/{stock_symbol}/sustainability?p={stock_symbol}"
    browser.visit(sustainability_url)
    if first:
        time.sleep(5)
        first = False
    else:
        time.sleep(1)
    ESG_Risk_Score = browser.find_by_css('div[class="Fz(36px) Fw(600) D(ib) Mend(5px)"]').value
    print(stock_symbol, ESG_Risk_Score)
    egs_rating_list.append(ESG_Risk_Score)
```

Figure 2: Extraction and Transformation of the ESG rating data from finance.yahoo.com

4. LOAD

The final step included the loading of the attained Pandas DataFrames into PostgreSQL. The following four steps elaborate on this process, as well as the following figures.

1. Creation of a SQL connection and engine instance
    a. For that, the «SQLAlchemy tool» was used for an initial connection to the «PostgreSQL Database» (relational database) to store clean data sets.
2. Creation of a SQL Database
    a. The database name «Stock_db» was created for the project in «PGAdmin».
3. Creation of Schema in the «Stock_db» with tables
    a. Using the Query Tool, we created a schema that stored two tables named «price_data_df» and «dividend_df» with all their respective details.
    b. For each table, columns were created that specified their respective data types, as well the exclusion null values for those columns.
    c. The «SQL join clause» was used to combine these two tables based on the related column between them. The relationship between the two tables is the «Symbol» column since both tables contain a «Symbol» column that refers to a unique ticker symbol to its respective company.
    d. We applied the «df.to_sql method» to load the Pandas DataFrames into the tables created in the «Stock_db» in PGAdmin.
    e. Finally, we applied the «pd.read_sql_query method» to confirm that respective data has been added to the table dataframe by querying the «price_data_df» and «dividend_df» in Jupyter Notebook.
4. The Final Table in the «stock database» is saved as a csv file.

stock_db/postgres@PostgreSQL 11

Query Editor | Query History

```
1  drop table if exists price_data_df;
2
3  -- Create Two Tables
4  CREATE TABLE price_data_df (
5      symbol TEXT PRIMARY KEY,
6      company TEXT NOT NULL,
7      price float NOT NULL,
8      low float NOT NULL,
9      high float NOT NULL,
10     previous_close float NOT NULL,
11     esg_rating int NOT NULL
12 );
13
14 drop table if exists dividend_df;
15 CREATE TABLE dividend_df (
16     symbol TEXT PRIMARY KEY,
17     dividend_yield float NOT NULL,
18     annualized_dividend float NOT NULL,
19     ex_div_date date NOT NULL,
20     pay_date date NOT NULL
21 );
22
23 select * from price_data_df;
24 select * from dividend_df;
```

Figure 3: Creation of the Schema with PostgreSQL

```
26 -- Joins tables
27 SELECT price_data_df.symbol, price_data_df.company, price_data_df.price,
28        price_data_df.low, price_data_df.high,price_data_df.previous_close,
29        price_data_df.esg_rating, dividend_df.symbol, dividend_df.dividend_yield,
30        dividend_df.annualized_dividend, dividend_df.ex_div_date, dividend_df.pay_date
31 FROM price_data_df
32 JOIN dividend_df
33 ON price_data_df.symbol = dividend_df.symbol;
```

Data Output | Explain | Messages | Notifications

	symbol text	company text	price double precision	low double precision	high double precision	previous_close double precision	esg_rating bigint	symbol text	dividend_yield double precision	annualized_dividend double precision	ex_div_date text	pay_date text
1	AXP	American Exp...	89.33	88.209	89.945	89.83	22	AXP	1.92	1.72	2020-07-01	2020-08-10
2	AAPL	Apple Inc	318.89	315.35	319.23	316.85	24	AAPL	1.06	3.28	2020-05-08	2020-05-14
3	BA	Boeing Co	137.53	135.777	141.08	139	39	BA	6.16	8.22	2020-02-13	2020-03-06
4	CAT	Caterpillar Inc	112.47	111.47	114.16	114.06	38	CAT	3.67	4.12	2020-04-17	2020-05-20
5	CSCO	Cisco System...	44.9	44.12	44.95	44.64	14	CSCO	3.35	1.44	2020-04-02	2020-04-22
6	CVX	Chevron Corp	90.28	89.465	91.6	92.04	40	CVX	5.4	5.16	2020-05-18	2020-06-10
7	XOM	Exxon Mobil ...	44.6	43.45	44.67	44.56	34	XOM	7.87	3.48	2020-05-12	2020-06-10
8	GS	Goldman Sac...	179.93	178.06	180.64	180.1	32	GS	2.7	5	2020-05-29	2020-06-29
9	HD	Home Depot I...	241.88	238.655	242.34	240.88	13	HD	2.56	6	2020-03-11	2020-03-26
10	IBM	International ...	118.39	117.59	119.465	119.12	18	IBM	5.3	6.52	2020-05-07	2020-06-10

Figure 4: Joining of the Tables in PostgreSQL

5. CONCLUSION AND CRITICAL REFLECTION

This ETL project revealed that it is possible to consolidate data from three different sources into one CSV file as well as dataframe. We learned how to find data sources that we can then consolidate together in one dataframe, from which we then subsequently could do a data analysis. Our «final_dow_data» dataframe portrays a consolidated view of the Dow 30 stock prices next to the ESG rating, as well as the data on the dividends.

Looking at the table, it would be interesting to do a data analysis on the correlation between the ESG score and the stock price or the annual dividend. For example, despite operating in the fossil fuel sector, Chevron Corporation had the highest ESG score, which comes as a surprise. This relationship with stock price and dividend may be subject for further data analysis.

```
final_dow_data = final_dow_data.sort_values(by='esg_rating', ascending=False)
final_dow_data
```

	symbol	company	price	low	high	previous_close	esg_rating	dividend_yield	annualized_dividend	ex_div_date	pay_date
5	CVX	Chevron Corp	93.30	92.03	94.340	90.280	40	5.72	5.1600	2020-05-18	2020-06-10
2	BA	Boeing Co	144.73	142.61	145.910	137.530	39	5.98	8.2200	2020-02-13	2020-03-06
3	CAT	Caterpillar Inc	117.41	115.98	118.562	112.470	38	3.66	4.1200	2020-04-17	2020-05-20
11	JNJ	Johnson & Johnson	144.56	144.30	146.440	144.370	35	2.75	4.0400	2020-05-22	2020-06-09
6	XOM	Exxon Mobil Corp	45.91	45.47	46.300	44.600	34	7.81	3.4800	2020-05-12	2020-06-10
15	MMM	3M Co	152.08	149.20	152.980	146.440	34	4.03	5.8800	2020-05-21	2020-06-12
19	PFE	Pfizer Inc	37.49	37.45	37.870	37.500	33	4.08	1.5200	2020-05-07	2020-06-05
7	GS	Goldman Sachs Group Inc	196.06	185.22	197.100	179.930	32	2.78	5.0000	2020-05-29	2020-06-29
25	WMT	Walmart Inc	123.86	123.63	125.510	124.330	29	1.73	2.1600	2020-08-13	2020-09-08
16	MRK	Merck & Co Inc	77.26	76.91	78.600	76.370	28	3.19	2.4400	2020-03-13	2020-04-07
12	KO	Coca-Cola Co	46.09	45.93	46.930	45.030	26	3.63	1.6400	2020-06-12	2020-07-01

Figure 5: Final Table

The following four learnings will support our work for further ETL projects.

- Do not use ".columns" method to rename dataframe columns, this method seems to have a bug which prevented us from loading data into the database
- Must match dataframe column names to database table column names. It is case sensitive!
- We had issues with the splinter ChromeDriver. We learned that the ChromeDriver version must match the Chrome browser's version.
- The single bracket yields a pandas's series, however, the double brackets yield the pandas data frame.


Coca-Cola
Chevron
BOEING
THE HOME DEPOT
3M
intel
Microsoft
New!
Johnson's
hand & face wipes
Walmart
Mobil
Disneyland
verizon
UnitedHealthcare
Pfizer
VISA
cisco
Pratt & Whitney
A United Technologies Company
Caterpillar
J.P.Morgan
Crest
DUPONT
AMERICAN EXPRESS
IBM
P&G
TRAVELERS
Goldman Sachs
MERCK
Be well