December 09, 2021

Web Scraping – Final Project

Fall 2021

TEAM 4

ABHISHEK BHAGAT [redacted]

[redacted]

Objective

Scrape the data for forecasting earnings and for measuring the market expectation of earnings

Business Problem

Scrape the data for forecasting earnings and for measuring the market expectation of earnings

We scrapped for all four quarters of 2020

Company Background

Name: Estimize (estimize.com)

Product: Open financial estimates platform

Established: 2011

Analysts: Currently 87,000 analysts contribute to Estimize

Introduction

Web scraping (or data scraping) is a technique used to collect content and data from the internet. This data is usually saved in a local file so that it can be manipulated and analyzed as needed. If you've ever copied and pasted content from a website into an Excel spreadsheet, this is essentially what web scraping is, but on a very small scale.

General Process

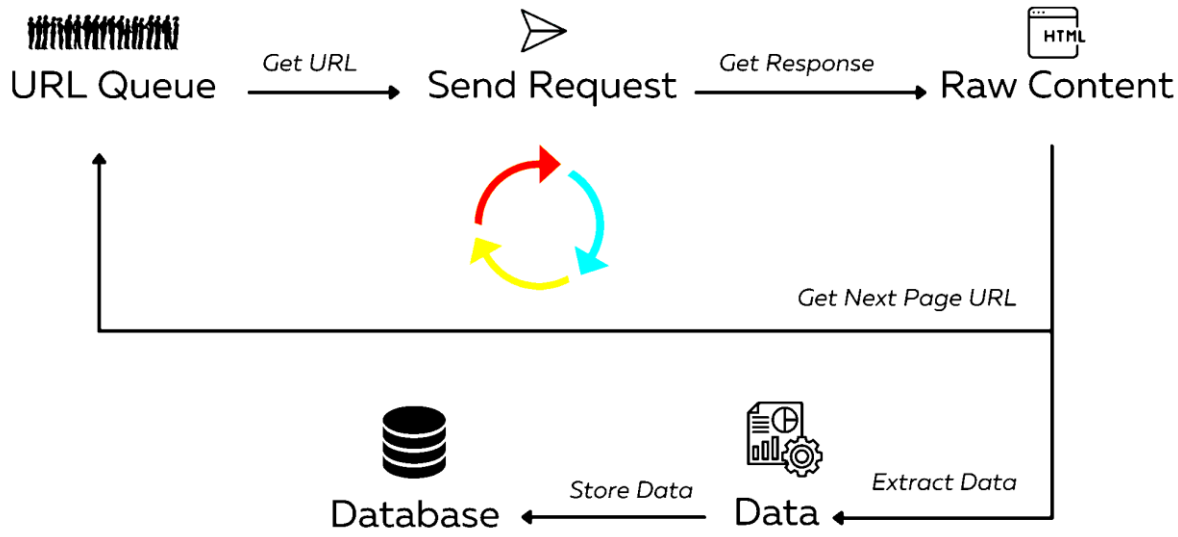
- ➔ Selected 50 tickers
- ➔ Scrape website
- ➔ Create dataframe
- ➔ Join to other dataframes
- ➔ Push to MySQL

Scraping Process

- ➔ Import: Selenium, Beautiful Soup
- ➔ Sign-in
- ➔ Use specific URL
- ➔ Use 50 tickers to scrape data into string
- ➔ Create respective column

High Level Workflow

We start with the web crawlers from a website URL. This page contains information that we want to crawl. We then send an HTTP GET request to the server in order to retrieve the webpage's content. When we collect the response, we transform the raw data into a clean and desirable format so we can use it in downstream processes. Occasionally, we may crawl more than one page. We will store the URLs of the remaining pages into a list and continue the loop until all pages have been crawled.



Part A: Company basic information

	ticker	name	sectors	Industries	Total Followers	Total Analysts	Top Analyst Name
0	GPC	Genuine Parts Company	Consumer Discretionary	Auto Components	96	123	BillB1210
1	ALV	Autoliv, Inc.	Consumer Discretionary	Auto Components	76	181	dalton
2	GNTX	Gentex Corp.	Consumer Discretionary	Auto Components	88	109	GHS_42
3	VNE	Veoneer, Inc.	Consumer Discretionary	Auto Components	8	10	Medic2525
4	DAN	Dana Incorporated	Consumer Discretionary	Auto Components	55	78	BillB1210

- Selected 50 random tickers to scrape
- Retrieved required contents from the web page in two data frames
- Cleaned the output and pushed it to MySQL
- Includes Ticker, Name, Sectors, Industries, Total Followers, Total Analysts, and the Top Analyst Name

Part B: EPS information

	Ticker	Company	Reported Earnings	Estimize Consensus	Estimize Mean	Wall Street Consensus	Quarter
0	GPC	Genuine Parts Company	0.92	1.11	1.09	1.11	Q1
1	ALV	Autoliv, Inc.	0.88	0.88	1.02	0.73	Q1
2	GNTX	Gentex Corp.	0.36	0.38	0.39	0.34	Q1
3	VNE	Veoneer, Inc.	-2.09	-1.17	-1.17	-1.26	Q1
4	TOWR	Tower International, Inc.	0.23	0.25	0.21	0.16	Q1

- Scraped EPS table for all companies for Q1 2020, dropped unnecessary columns and created dataframe
- Then repeated scraping process for Q2, Q3, Q4
- Added 'Quarter' column for each dataframe
- Finally, we joined all four data frames to contain all quarters
- Includes Ticker, Name, Sectors, Industries, Total Followers, Total Analysts, and the Top Analyst Name

Part B: EPS Analyst Information

➔ Includes Analyst name, EPS Estimation value, and Analyst ID

	analyst_Name	EPS Estimation value	analyst_id
0	Rick Weinhart	0.89	weinhart
1	Frederick Tremblay	1.11	quanttrader007
2	Bill	1.12	billb1210
3	Ben Jen	1.12	benjenholdings
4	Scott Hendricks	1.21	scooter186
5	Analyst_8113162	0.97	analyst_8113162
6	Bill	0.95	billb1210
7	Sentinel	0.89	iosentinel
8	Medic2525	0.87	medic2525
9	Analyst_6401834	0.75	analyst_6401834
10	Analyst_9241646	0.73	analyst_9241646
11	Rick Weinhart	0.57	weinhart
12	Rick Weinhart	1.51	weinhart
13	Anthony Lara	1.50	analyst_454701
14	Ron	1.46	analyst_8317009
15	Bill	1.45	billb1210
16	Ben Jen	1.42	benjenholdings

Part C: Analyst Information

➔ Includes Analyst ID, Error Rate, Accuracy, Points, Points/Estimate, Stocks, Pending, Analyst Name, Confidence Score, and Joining Date

```
analyst_df['analyst_role']=pd.DataFrame(list13,columns=['analyst_role'])
analyst_df
```

	analyst_id	error rate	accuracy	points	points/Estimate	stocks	pending	analyst_name	confidence_score	joining_date	analyst_role
0	weinhart	-	-	-	-	-	0	Rick Weinhart	7.2	Mar 2015	Financial Professional Buy Side Hedge Fund
1	quanttrader007	-	-	-	-	-	0	Frederick Tremblay	7.9	Apr 2014	Financial Professional Buy Side Asset Manager
2	billb1210	16.5%	59%	25,628	13.6	1,880	24	Bill	8.4	Jul 2014	Non Professional Financials Professional Services
3	benjenholdings	16.0%	60%	23,902	14.1	1,701	14	Ben Jen	8.3	Jul 2016	Financial Professional Independent Independent...
4	scooter186	16.0%	58%	3,155	11.5	274	12	Scott Hendricks	8.4	Oct 2012	Non Professional Industrials Commercial Servic...
5	analyst_8113162	-	-	-	-	-	0	Analyst_8113162	7.6	Apr 2020	Non Professional Other Other
6	billb1210	16.5%	59%	25,628	13.6	1,880	24	Bill	8.4	Jul 2014	Non Professional Financials Professional Services
7	iosentinel	-	-	-	-	-	0	Sentinel	6.6	Aug 2018	Financial Professional Independent Independent...
8	medic2525	28.0%	24%	-53	-5.3	10	2	Medic2525	8.3	Apr 2019	Non Professional Health Care Health Care Provi...
9	analyst_6401834	-	-	-	-	-	0	Analyst_6401834	6.9	Jun 2020	Non Professional Industrials Aerospace & Defense

Stocks covered

	username	Ticker	Quarter	Reported	Rank	EPS	Revenue_points	Total_points	Professional_information
0	weinhart	GPC	Q42020	Feb 17, 2021	2 / 2	17.0	-12.0	5	Financial Professional, Buy Side, Hedge Fund
1	weinhart	GPC	Q32020	Oct 22, 2020	5 / 5	12.0	-25.0	-13	Financial Professional, Buy Side, Hedge Fund
2	weinhart	GPS	Q22020	Aug 27, 2020	9 / 9	-9.0	-4.0	-13	Financial Professional, Buy Side, Hedge Fund
3	weinhart	URBN	Q22021	Aug 25, 2020	3 / 7	22.0	-4.0	18	Financial Professional, Buy Side, Hedge Fund
4	weinhart	GPC	Q22020	Jul 30, 2020	6 / 7	-6.0	12.0	6	Financial Professional, Buy Side, Hedge Fund

Pending estimates

	Name	Ticker	Quarter	Reports	Published	EPS	Revenue
0	billb1210	NETI	Q3 2021	Dec 8, 2021 BMO	Oct 24, 2021	0.64	34.70
1	billb1210	NM	Q3 2021	Dec 8, 2021 BMO	Nov 20, 2021	2.42	146.90
2	billb1210	CPB	Q1 2022	Dec 8, 2021 BMO	Dec 5, 2021	0.83	2,296
3	billb1210	KFY	Q2 2022	Dec 8, 2021 BMO	Dec 5, 2021	1.45	622.67
4	billb1210	VRA	Q3 2022	Dec 8, 2021 BMO	Dec 5, 2021	0.27	143.81

Score estimates

	accuracy	errorRate	name	nextReportsAt	path	points	pointsPerEstimate	quarters	reportFrequency	ticker	count	empty	totalCount	userDisplayName
0	42	6.8	Genuine Parts Company	Feb 22	/gpc	13	1.1	12	None	GPC	10	False	10	Rick Weinhart
1	66	24.1	Abercrombie & Fitch Co.	Mar 2	/anf	116	11.6	10	None	ANF	10	False	10	Rick Weinhart
2	48	12.1	The Childrens Place Retail Stores	Mar 8	/plce	46	4.6	10	None	PLCE	10	False	10	Rick Weinhart
3	68	16.0	Urban Outfitters Inc.	Mar 8	/urbn	139	15.4	9	None	URBN	10	False	10	Rick Weinhart
4	56	13.9	Zumiez	Mar 10	/zumz	138	17.3	8	None	ZUMZ	10	False	10	Rick Weinhart

Part D: Query Solutions (A)

Given a ticker BWA, how many analysts have made estimations for its EPS? Rank them by their confidence score, total points, error rate or accuracy percentile?

A. Select * from ticker_info_tbl where ticker = 'BWA';

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ticker	name	sectors	Industries	Total Followers	Total Analyst	Top Analyst Name
BWA	BorgWarner Inc.	Consumer Discretionary	Auto Components	116	18	BillB1210

B. Select `analyst_id`, `error rate`, `accuracy`, `points`, `points/Estimate`, `stocks`, `pending`, `analyst_name`, `confidence_score`, `joining_date`, rank() OVER (order by `confidence_score`desc) AS 'rank' from analyst_tbl where `points` !='-'and `error rate` !='-' and `accuracy` !='-';

analyst_id	error rate	accuracy	points	points/Estimate	stocks	pending	analyst_name	confidence_score	joining_date	rank
analyst_700041	12.9%	69%	179	19.9	9	7	TBuckley	9.5	Dec 2015	1
upandtotheright	4.3%	62%	907	18.5	49	12	Tom Telford	8.9	May 2012	2
udalltechstrat	19.8%	49%	88	9.8	9	2	Sean Udall	8.8	Apr 2012	3
analyst_5039344	30.0%	60%	115	11.5	10	1	Analyst_5039344	8.6	Oct 2017	4
analyst_5039344	30.0%	60%	115	11.5	10	1	Analyst_5039344	8.6	Oct 2017	4
analyst-7577288	1.9%	60%	-4	-4	1	0	AhmedTheGreat7	8.4	Feb 2015	6
billb1210	16.6%	59%	25,559	13.6	1,877	28	Bill	8.4	Jul 2014	6
shawnpcooney	13.9%	57%	502	11.2	45	4	Shawn P. Cooney	8.4	Jan 2019	6
francescozunino	14.7%	46%	279	12.1	23	1	francescozunino	8.4	Sep 2018	6
scooter186	16.0%	58%	3,163	11.6	273	13	Scott Hendricks	8.4	Oct 2012	6
analyst_76071	3.2%	78%	-2	-2	1	0	Analyst_76071	8.3	Feb 2016	11

Part D: Query Solutions (B)

Given Auto Components, how many companies are covered, the average number of analysts, the average bias between the Estimize Consensus and the Reported Earnings?

with t1 as (SELECT COUNT(*) AS Number_of_company from ticker_info_tbl where Industries ='Auto Components'),

t2 as (SELECT cast(avg(`Total Analysts`) as decimal(10,2)) AS Average_number_analyst from ticker_info_tbl where Industries = 'Auto Components'),

t3 as (select cast(avg((ec-re)/c1) as decimal(10,2)) as average_bias from(select sum(`Estimize Consensus`) as ec, sum(`Reported Earnings`) as re, count(*) as c1 from eps_info_tbl where Ticker in (select ticker from ticker_info_tbl where Industries ='Auto Components')) b) select * from t1,t2,t3;

Number_of_company	Average_number_analyst	average_bias
30	92.71	0.12

ANSWER: Under the industry 'Auto Components', there are 30 companies, an average of 92.71 analysts, and the average bias between Estimize Consensus and Reported Earnings is 0.12

Part D: Query Solutions (C)

Which company has the largest number of analysts with a confidence score greater than 7?

```
9
10    # Ticker with maximum number of analyst with confidence score higher than 7
11    select t2.Ticker as Ticker, count(analyst_id) as Analyst_Count, avg(t3.confidence_score) as Average_Confidence_Score from ticker_info_tbl t1
12    left join ticker_analyst_tbl t2 on t2.Ticker = t1.ticker
13    left join analyst_tbl t3 on t2.username = t3.analyst_id
14    where t3.confidence_score > 7
15    group by t2.Ticker
16    order by count(analyst_id) desc
17    limit 1;
18
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Ticker	Analyst_Count	Average_Confidence_Score
CRM	32	8.015625

Answer: CRM has the largest number of analysts that have a confidence score greater than 7.

Part D: Query Solutions (D)

Who has the largest number of followers?

select `ticker`, `name`, `sectors`, `Industries`, `Total Followers`, `Total Analysts` from (select `ticker`, `name`, `sectors`, `Industries`, `Total Followers`, ` Total Analysts`, rank() OVER (order by `Total Followers`desc) as rank_1 from ticker_info_tbl) a where a.rank_1=1

ticker	name	sectors	Industries	Total Followers	Total Analysts
GM	General Motors Company	Consumer Discretionary	Automobiles	983	727

ANSWER: GM has the largest number of total followers.

BONUS: Part A

Given all the features constructed and scraped, can you find some important independent variables that affect accuracy of prediction?

- Checked the correlation matrix for all the variables
- Chose to add 'Reported earnings', 'Total Number of Analysts', 'Total Number of Followers' as the features contributing to better predict the target ie, Estimize Consensus

In [321]:
```
from sklearn.metrics import mean_squared_error
from math import sqrt

rms = sqrt(mean_squared_error(y_test, y_pred))
rms
```

Out[321]:
```
0.6597527178031544
```

In [322]:
```
from sklearn.ensemble import RandomForestRegressor
#model = RandomForestRegressor(max_depth=2, random_state=0)
#model.fit(X, y)
#y_pred = model.predict(test_df1)

#sub.drop('sales', axis = 1, inplace = True)
#sub['sales'] = y_pred
## SCORE: 2.72546
# importing train_test_split from sklearn
 #Train the model
model = RandomForestRegressor(max_depth=10, random_state=12)
model.fit(X_train, y_train)
y_pred =  model.predict(X_test)
y_pred
# importing r2_score module
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
# predicting the accuracy score
# predicting the accuracy score
score=r2_score(y_test,y_pred)
print(score)
```

```
0.7508463657394893
```

BONUS: Part B

Can you come up with some novel method to construct a better EPS estimation compared with the Estimize Consensus and Wall Street Consensus?

We decided to run a linear regression model using reported earnings, number of analysts and number of followers as the features and estimize consensus as the target. We find the r-squared and the root mean squared error for this model. We also ran decision tree and random forest models and computed the r-squared and root mean squared error for both these models as well. Random Forest performed the best out of the three models.

Linear Regression Model

In [318]:

```
# Training data
X = df_1.loc[:, ["Total_Followers","Total_Analysts", "Reported_Earnings"]] # features
y = df_1.loc[:, 'Estimize_Consensus'] # target

# importing train_test_split from sklearn
from sklearn.model_selection import train_test_split
# splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 9)

 #Train the model
model = LinearRegression()
model.fit(X_train, y_train)

y_pred =  model.predict(X_test)
y_pred
#Store the fitted values as a time series with the same time index as
# the training data
#y_pred = pd.Series(model.predict(X), index = X.index)
#y_pred
#model.score(X, y)
# importing r2_score module
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
# predicting the accuracy score
score=r2_score(y_test,y_pred)
print(score)
```

```
0.39085477260886936
```

In [319]:

```
from sklearn.metrics import mean_squared_error
from math import sqrt

rms = sqrt(mean_squared_error(y_test, y_pred))
rms
```

Out[319]: 0.7773694183048983

```
 #Train the model
model = RandomForestRegressor(max_depth=10, random_state=12)
model.fit(X_train, y_train)
y_pred =  model.predict(X_test)
y_pred
# importing r2_score module
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
# predicting the accuracy score
# predicting the accuracy score
score=r2_score(y_test,y_pred)
print(score)
```

```
0.7508463657394893
```

In [323]:
```
from sklearn.metrics import mean_squared_error
from math import sqrt

rms = sqrt(mean_squared_error(y_test, y_pred))
rms
```

Out[323]:
```
0.497165108352544
```

Challenges we faced

Some challenges we faced included that we could not scrape all the information we needed for one table at once. We had to scrape in sections and then join the data frames we created from the scraped information to meet one table's requirements.

Another challenge we faced was that the JSON need pre-processing as the data format was complex. We had to convert single quotes to double quotes and convert string to dictionary using JSON library.

Variables were in object type, but we needed them in float or integer to do any further analysis in python (constructing correlation matrix, running models, etc.).