Big Data Hadoop and Spark Developer

Project: Stock Exchange Data Analysis

Tools used: Simplilearn Lab (HIVE)

DESCRIPTION

Objective: To use hive features for data engineering or analysis and sharing the actionable insights

Problem Statement:
New York stock exchange data of seven years, between 2010 to 2016, is captured for 500+ listed companies. The data set comprises of intra-day prices and volume traded for each listed company. The data serves both for machine learning and exploratory analysis projects, to automate the trading process and to predict the next trading-day winners or losers. The scope of this project is limited to exploratory data analysis.

Domain: BFSI

Analysis to be done: Exploratory analysis to understand how MoM or YoY companies from different sectors or industries and states have progressed in a period of 7 years

Content: This data set contains prices.csv and securities.csv files having the following features:

<u>Prices.csv:</u>

1. Date: Trading date
2. Symbol: Ticker code or listed company code on NY stock exchange
3. Open: Intra-day opening price for each listed company
4. Close: Intra-day closing price for each listed company
5. Low: Intra-day lowest price for each listed company
6. High: Intra-day highest price for each listed company
7. Volume: Number of shares traded per day per company

<u>Securities.csv:</u>

1. Ticker_Symbol: Country to which the customer belongs
2. Security: Legal name of the listed company
3. Sector: Business vertical of the listed company
4. Sub_Industry: Business domain of the listed company within a Sector.
5. Headquarter: Headquarters address

Steps to perform:

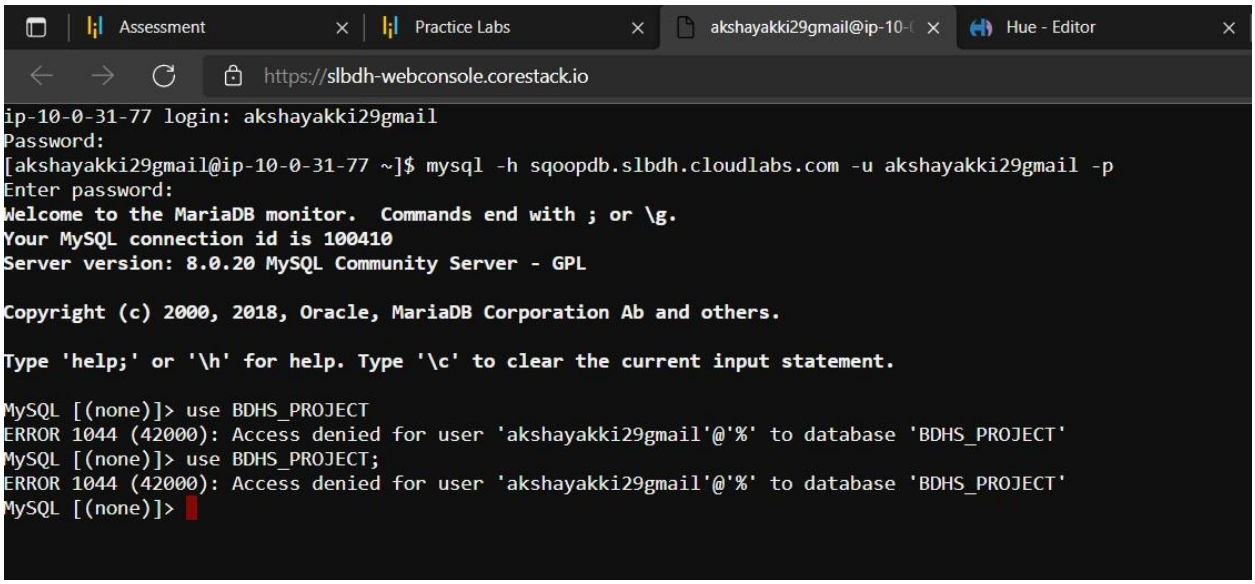
1) Create a data pipeline using sqoop to pull the data from the table below from MYSQL server into Hive.

a. MYSQL DATABASE NAME: BDHS_PROJECT

i. Stock_prices
ii. Stock_companies

Check the TABLE description: STOCK_PRICES

Note: I tried accessing the database BDHS_PROJECT however it is giving permission denied error and also tried other things but no success. So, we executed the whole project in HIVE. See the error detail below:



Solution: We will use HIVE to execute this project. First log into HIVE using given id and password. Select Query > Editor > Hive. Create a database BDHS_PROJECT_NYSE using following command:

create database BDHS_PROJECT_NYSE;

then use this database using below command:

use database BDHS_PROJECT_NYSE;

Now you can see that your database has been changed to BDHS_PROJECT_NYSE.

Now we will create two tables STOCK_COMPANIES and STOCK_PRICES and import the csv data into it using below command: (First upload the two csv files into your hive)

-- Create table STOCK_COMPANIES

create external table if not exists STOCK_COMPANIES(Symbol varchar(255), Company_name varchar(255), Sector varchar(255), Sub_industry varchar(255), Headquarter varchar(255))

row format delimited

fields terminated by ','

stored as TEXTFILE

location '/user/akshayakki29gmail/BDHS_PROJECT_NYSE/STOCK_COMPANIES/'

tblproperties('skip.header.line.count' = "1");

see below screenshot:

```
INFO  : Executing command(queryId=hive_20211018141935_a0d98564-9a10-4ffa-9866-6b5ef10f8fc7): create external table if not e
xists STOCK_COMPANIES(Symbol varchar(255), Company_name varchar(255), Sector varchar(255),
Sub_industry varchar(255), Headquarter varchar(255))
row format delimited
fields terminated by ','
stored as TEXTFILE
location '/user/akshayakki29gmail/BDHS_PROJECT_NYSE/STOCK_COMPANIES/'
tblproperties('skip.header.line.count' = "1")
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20211018141935_a0d98564-9a10-4ffa-9866-6b5ef10f8fc7); Time taken: 0.057 se
conds
INFO  : OK
```

-- Show 10 rows from STOCK_COMPANIES

select * from stock_companies limit 10;

see below screenshot:

Query History Saved Queries Results (10)

	stock_companies.symbol	stock_companies.company_name	stock_companies.sector	stock_companies.sub_industry	stock_companies.headquarter
1	MMM	3M Company	Industrials	Industrial Conglomerates	St. Paul; Minnesota
2	ABT	Abbott Laboratories	Health Care	Health Care Equipment	North Chicago; Illinois
3	ABBV	AbbVie	Health Care	Pharmaceuticals	North Chicago; Illinois
4	ACN	Accenture plc	Information Technology	IT Consulting & Other Services	Dublin; Ireland
5	ATVI	Activision Blizzard	Information Technology	Home Entertainment Software	Santa Monica; California
6	AYI	Acuity Brands Inc	Industrials	Electrical Components & Equipment	Atlanta; Georgia
7	ADBE	Adobe Systems Inc	Information Technology	Application Software	San Jose; California
8	AAP	Advance Auto Parts	Consumer Discretionary	Automotive Retail	Roanoke; Virginia
9	AES	AES Corp	Utilities	Independent Power Producers & Energy Traders	Arlington; Virginia
10	AET	Aetna Inc	Health Care	Managed Health Care	Hartford; Connecticut

-- Create table STOCK_PRICES

create external table if not exists STOCK_PRICES(Trading_date date, Symbol varchar(255), Open decimal(10,2), Close decimal(10,2), Low decimal(10,2), High decimal(10,2), Volume int)

row format delimited

fields terminated by ','

stored as TEXTFILE

location '/user/akshayakki29gmail/BDHS_PROJECT_NYSE/STOCK_PRICES/'

tblproperties('skip.header.line.count' = "1");

see below screenshot:

```
INFO  : Completed compiling command(queryId=hive_20211018144122_128d7ffa-49e4-4b98-b057-3556a0ab5b3e); Time taken: 0.016 se
conds
INFO  : Executing command(queryId=hive_20211018144122_128d7ffa-49e4-4b98-b057-3556a0ab5b3e): create external table if not e
xists STOCK_PRICES(Trading_date date, Symbol varchar(255), Open decimal(10,2),
Close decimal(10,2), Low decimal(10,2), High decimal(10,2), Volume int)
row format delimited
fields terminated by ','
stored as TEXTFILE
location '/user/akshayakki29gmail/BDHS_PROJECT_NYSE/STOCK_PRICES/'
tblproperties('skip.header.line.count' = "1")
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20211018144122_128d7ffa-49e4-4b98-b057-3556a0ab5b3e); Time taken: 0.052 se
conds
INFO  : OK
```

✔ Success.

-- Show 10 rows from STOCK_PRICES

select * from stock_prices limit 10;

see below screenshot:

Query History | Saved Queries | Results (10)

	stock_prices.trading_date	stock_prices.symbol	stock_prices.open	stock_prices.close	stock_prices.low	stock_prices.high	stock_prices.volume
1	2016-01-05	WLTW	123.43	125.84	122.31	126.25	2163600
2	2016-01-06	WLTW	125.24	119.98	119.94	125.54	2386400
3	2016-01-07	WLTW	116.38	114.95	114.93	119.74	2489500
4	2016-01-08	WLTW	115.48	116.62	113.50	117.44	2006300
5	2016-01-11	WLTW	117.01	114.97	114.09	117.33	1408600
6	2016-01-12	WLTW	115.51	115.55	114.50	116.06	1098000
7	2016-01-13	WLTW	116.46	112.85	112.59	117.07	949600
8	2016-01-14	WLTW	113.51	114.38	110.05	115.03	785300
9	2016-01-15	WLTW	113.33	112.53	111.92	114.88	1093700
10	2016-01-19	WLTW	113.66	110.38	109.87	115.87	1523500

-- Describe STOCK_COMPANIES

describe stock_companies;

see below screenshot:

Query History | Saved Queries | Results (5)

	col_name	data_type	comment
1	symbol	varchar(255)	
2	company_name	varchar(255)	
3	sector	varchar(255)	
4	sub_industry	varchar(255)	
5	headquarter	varchar(255)	

--Describe STOCK_PRICES

describe stock_prices;

see below screenshot:

Query History | Saved Queries | Results (7)

	col_name	data_type	comment
1	trading_date	date	
2	symbol	varchar(255)	
3	open	decimal(10,2)	
4	close	decimal(10,2)	
5	low	decimal(10,2)	
6	high	decimal(10,2)	
7	volume	int	

-- Number of rows in STOCK_COMPANIES

select count(*) as num_rows from stock_companies;

see below screenshot:

Query History | Saved Queries | Results (1)

	num_rows
1	505

-- Number of rows in STOCK_PRICES

select count(*) as num_rows from stock_prices;

see below screenshot:

Query History | Saved Queries | Results (1)

	num_rows
1	851264

2) Create a new hive table with the following fields by joining the above two hive tables. Please use appropriate Hive built-in functions for columns (a,b,e and h to l).

6. Trading_year: Should contain YYYY for each record
7. Trading_month: Should contain MM or MMM for each record
8. Symbol: Ticker code
9. CompanyName: Legal name of the listed company
10. State: State to be extracted from headquarters value.
11. Sector: Business vertical of the listed company
12. Sub_Industry: Business domain of the listed company within a sector
13. Open: Average of intra-day opening price by month and year for each listed company
14. Close: Average of intra-day closing price by month and year for each listed company
15. Low: Average of intra-day lowest price by month and year for each listed company
16. High: Average of intra-day highest price by month and year for each listed company
17. Volume: Average of number of shares traded by month and year for each listed company

Solution: To perform above mentioned action use following command to create a new table with table name Stock_Market_Final:

```
create table Stock_Market_Final as select Trading_year, Trading_month, sc.Symbol, sc.Company_name as CompanyName,

trim(split(headquarter,"\;")[1]) as State, sector, sub_industry, open, close, low, high, volume

from stock_companies as sc,

(select Symbol, year(Trading_date) as Trading_year, month(Trading_date) as Trading_month, round(avg(Open),2) as Open,

round(avg(Close),2) as Close, round(avg(Low),2) as Low, round(avg(High),2) as High, round(avg(Volume),2) as Volume

from stock_prices

group by Symbol, month(Trading_date),year(Trading_date)) as sp

where sc.Symbol=sp.Symbol;
```

see below screenshot:

```
-- create a new table
create table Stock_Market_Final as select Trading_year, Trading_month, sc.Symbol, sc.Company_name as CompanyName,
trim(split(headquarter,"\;")[1]) as State, sector, sub_industry, open, close, low, high, volume
from stock_companies as sc,
(select Symbol, year(Trading_date) as Trading_year, month(Trading_date) as Trading_month, round(avg(Open),2) as Open,
round(avg(Close),2) as Close, round(avg(Low),2) as Low, round(avg(High),2) as High, round(avg(Volume),2) as Volume
from stock_prices
group by Symbol, month(Trading_date),year(Trading_date)) as sp
where sc.Symbol=sp.Symbol;
```

```
INFO  : Moving data to directory hdfs://nameservice1/user/hive/warehouse/bdhs_project_nyse.db/stock_market_final from hdf
s://nameservice1/user/hive/warehouse/bdhs_project_nyse.db/.hive-staging_hive_2021-10-18_18-03-59_709_1012400035959107691-10
193/-ext-10001
INFO  : Starting task [Stage-7:DDL] in serial mode
INFO  : Starting task [Stage-3:STATS] in serial mode
INFO  : MapReduce Jobs Launched:
INFO  : Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 34.98 sec   HDFS Read: 51300634 HDFS Write: 1992336 HDFS EC Rea
d: 0 SUCCESS
INFO  : Stage-Stage-5: Map: 1   Cumulative CPU: 6.43 sec   HDFS Read: 2000903 HDFS Write: 4502182 HDFS EC Read: 0 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 41 seconds 410 msec
INFO  : Completed executing command(queryId=hive_20211018180359_e3dea727-80be-48a6-92de-914f01c12a02); Time taken: 112.9 se
conds
INFO  : OK
```

job_1622117371245_51421

job_1622117371245_51422

-- Show 10 rows from Stock_Market_Final

select * from stock_market_final limit 10;

see below screenshot:

Query History | Saved Queries | Results (10)

	stock_market_final.trading_year	stock_market_final.trading_month	stock_market_final.symbol	stock_market_final.companyname	stock_market_final.state	stock_market_final.sector	stock_market_final.sub_industry	stock_market_final.open	stock_market_final.close	stock_market_final.low	stock_market_final.high	stock_market_final.volume
1	2010	1	A	Agilent Technologies Inc	California	Health Care	Health Care Equipment	21.72	21.61	21.40	21.86	4208442.11
2	2011	1	A	Agilent Technologies Inc	California	Health Care	Health Care Equipment	30.29	30.29	29.96	30.65	4496845
3	2012	1	A	Agilent Technologies Inc	California	Health Care	Health Care Equipment	28.54	28.78	28.22	29.08	5069975
4	2013	1	A	Agilent Technologies Inc	California	Health Care	Health Care Equipment	31.20	31.26	30.97	31.45	4567619.05
5	2014	1	A	Agilent Technologies Inc	California	Health Care	Health Care Equipment	42.01	42.04	41.66	42.26	3494200
6	2015	1	A	Agilent Technologies Inc	California	Health Care	Health Care Equipment	39.91	39.12	38.74	39.64	2654295
7	2016	1	A	Agilent Technologies Inc	California	Health Care	Health Care Equipment	38.23	38.05	37.58	38.67	2669947.37
8	2010	2	A	Agilent Technologies Inc	California	Health Care	Health Care Equipment	21.48	21.55	21.23	21.70	5698021.05
9	2011	2	A	Agilent Technologies Inc	California	Health Care	Health Care Equipment	30.67	30.71	30.20	31.14	5952100
10	2012	2	A	Agilent Technologies Inc	California	Health Care	Health Care Equipment	31.37	31.38	30.97	31.69	4190800

<u>DATA ANALYSIS USING HIVE</u>

3) Find the top five companies that are good for investment

Solution: To get above results use following command:

select companyname, avg(high) as avg_high

from stock_market_final

group by companyname

order by avg_high desc limit 5;

see below screenshot:

Query History Saved Queries Results (5)

	companyname	avg_high
1	Priceline.com Inc	871.222024
2	AutoZone Inc	472.173690
3	Alphabet Inc Class A	470.746190
4	Intuitive Surgical Inc.	466.970952
5	Alphabet Inc Class C	463.232738

4) Show the best-growing industry by each state, having at least two or more industries mapped.

Solution: To get above results use following command:

```
select state, sector, avg(high) as avg_high, count(sub_industry) as industry_count
from stock_market_final
group by state, sector
having industry_count >= 2
order by avg_high desc;
```

see below screenshot:

Query History Saved Queries Results (208)

	state	sector	avg_high	industry_count
1	Colorado	Consumer Discretionary	415.480476	84
2	Connecticut	Consumer Discretionary	285.941577	336
3	Tennessee	Consumer Discretionary	194.225635	252
4	Ohio	Materials	175.389405	84
5	Indiana	Real Estate	148.765952	84
6	Ohio	Health Care	146.173869	168
7	Virginia[3]	Real Estate	141.293571	84
8	Missouri	Consumer Discretionary	139.902738	84
9	Maryland	Industrials	138.648929	84
10	California	Health Care	127.169987	756
11	Washington	Consumer Discretionary	125.963690	336
12	Nevada	Consumer Discretionary	125.056190	84
13	Tennessee	Industrials	122.197500	84
14	Indiana	Industrials	113.478929	84
15	Connecticut	Materials	110.775119	84
16	Texas	Information Technology	110.443036	168
17	Washington	Consumer Staples	110.273333	84

5) For each sector find the following.

a) Worst year:

Solution: To get above results use following command:

select sector, trading_year as worst_year

from (select sector, trading_year, avg_sector_volume, rank() over(partition by sector order by avg_sector_volume) as swy

from (select sector, trading_year, avg(volume) as avg_sector_volume from stock_market_final

group by sector, trading_year) as a) as b

where b.swy = 1;

See below screenshot:

Query History Saved Queries Results (11)

	sector	worst_year
1	Consumer Discretionary	2015
2	Consumer Staples	2014
3	Energy	2013
4	Financials	2014
5	Health Care	2014
6	Industrials	2014
7	Information Technology	2016
8	Materials	2014
9	Real Estate	2014
10	Telecommunications Services	2013
11	Utilities	2013

b) Best Year:

Solution: To get above results use following command:

select sector, trading_year as best_year

from (select sector, trading_year, avg_sector_best, rank() over(partition by sector order by avg_sector_best desc) as sby

from (select sector, trading_year, avg(volume) as avg_sector_best from stock_market_final

group by sector, trading_year) as a) as b

where b.sby = 1;

see below screenshot:

Query History Saved Queries Results (11)

	sector	best_year
1	Consumer Discretionary	2010
2	Consumer Staples	2010
3	Energy	2016
4	Financials	2010
5	Health Care	2010
6	Industrials	2010
7	Information Technology	2010
8	Materials	2010
9	Real Estate	2010
10	Telecommunications Services	2011
11	Utilities	2011

c) Stable Year:

Solution: To get above results use following command:

select sector, trading_year as stable_year

from (select sector, trading_year, avg_sector_stable, rank() over(partition by sector order by avg_sector_stable) as ssy

from (select sector, trading_year, avg(high - low) as avg_sector_stable from stock_market_final

group by sector, trading_year) as a) as b

where b.ssy = 1;

see below screenshot:

Query History Saved Queries Results (11)

	sector	stable_year
1	Consumer Discretionary	2010
2	Consumer Staples	2010
3	Energy	2010
4	Financials	2012
5	Health Care	2010
6	Industrials	2010
7	Information Technology	2010
8	Materials	2010
9	Real Estate	2012
10	Telecommunications Services	2012
11	Utilities	2012

Conclusion:

- We do not have access to BDHS_PROJECT so we executed the project in HIVE using appropriate importing csv files methods and created necessary tables for analysis.
- We joined the two tables STOCK_COMAPANIES and STOCK_PRICES and created a final table with name Stock_Market_Final.
- The top five companies that are good for investment are: Priceline.com Inc, AutoZone Inc, Alphabet Inc Class A, Intuitive Surgical Inc, Alphabet Inc Class C.
- We found the best-growing industry by each state, having at least two or more industries mapped.
- For each sector we found the Worst Year, Best Year and Stable Year.