End- to- End ETL project for Data Analysis

Project Idea:

Extract the data using Kaggle API, load the data after cleaning and feature engineering into a SQL Server, followed by queries solving some real world questions.

Kaggle - - - -Python- - - - SQL Server – - - - Analysis

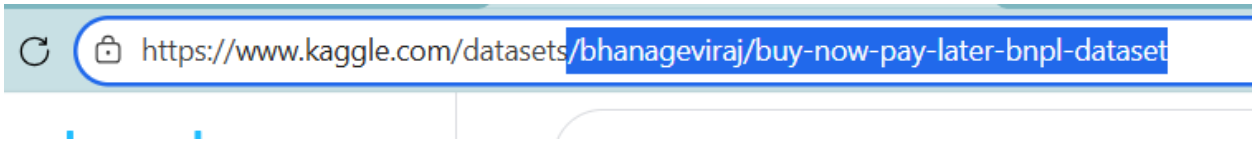
Setting up Kaggle API:

From the Kaggle DOCS we can get the information on how to connect to the public API in order to extract data.

- Kaggle API Token :
- export KAGGLE_API_TOKEN=KGAT
- kaggle competitions list

Data Extraction:

Once the kaggle connection is setup we can now go ahead and get the URL of the dataset that we want to analyse.




```
import kaggle
```

```
!kaggle datasets download ankitbansal06/retail-orders -f orders.csv
```

Unzip the files:

```
import zipfile
zip_ref=zipfile.ZipFile('orders.csv.zip')
zip_ref.extractall()
zip_ref.close()
```

This code snippet will unzip the files mentioned in the path or name of the file(if in the same container) and extract them to specified folder or current folder(like here).
We will have the .csv file in the folder now.

Data Cleaning and Feature Engineering:

```
df= pd.read_csv('orders.csv',na_values=['Not Available', 'unknown'])
```

✓ 0.0s

This will make sure the values Not Available and Unknown are treated as Null Values while loading the data.

```
df.columns=df.columns.str.lower()
```

✓ 0.0s

```
df.columns=df.columns.str.replace(' ', '_')
```

✓ 0.0s

```
df.head()
```

✓ 0.0s

order_id	order_date	ship_mode	segment	country	city	state	postal_code	region	category	sub_category	product_id	cost_price

This code snippet helps us in maintaining a proper format for the column names, first we try and lowercase the names followed by replacing “ “ with “_”.

Next, created three new columns based on the domain expertise for final analysis.

```
df['discount_amount']=df['list_price']*(df['discount_percent']/100)
```
✓ 0.0s

```
df['sale_price']=df['list_price']-df['discount_amount']
```
✓ 0.0s

```
df['profit']=df['sale_price']-df['cost_price']
```
✓ 0.0s

Add Markdown Cell

```
df.drop(columns=['list_price','discount_percent','cost_price'],inplace=True)
```
✓ 0.0s

So we can now drop those extra columns as we have already captured the meaning of them in the new columns.

```
df['order_date']=pd.to_datetime(df['order_date'],format='%Y-%m-%d')
```
✓ 0.0s

Also when observing the datatypes of the dataframe we have to be sure that the column is in the correct format as per their values.

Loading the Data into SQL Server:

```
import sqlalchemy as sas
engine= sas.create_engine('mssql://{Name of the server}/{Database Name}?driver=ODBC+Driver+17+for+SQL+Server')
conn= engine.connect()
```

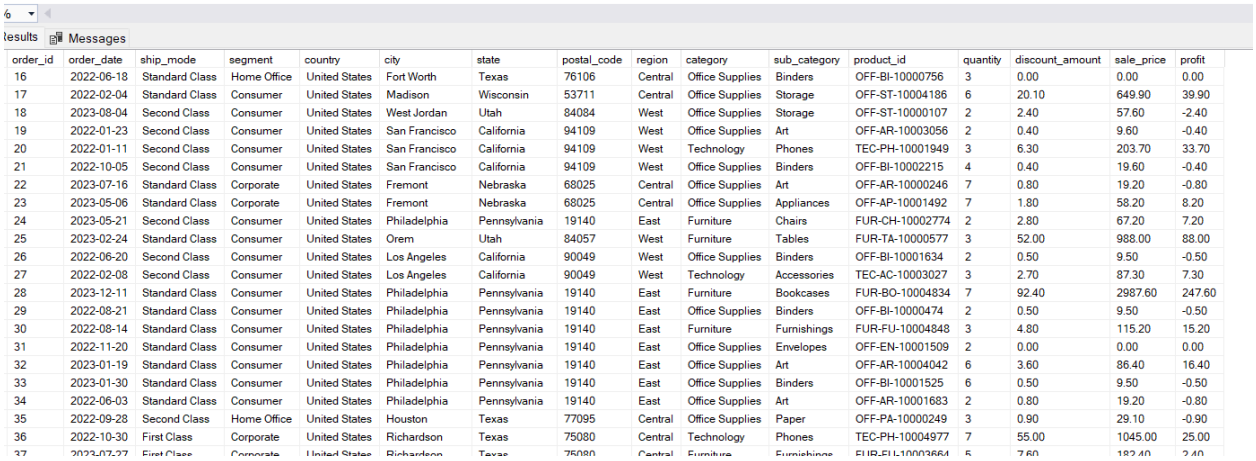
This helps us in connecting to the SQL Server and the database.

```
#df.to_sql('df_orders',con=conn,if_exists='replace',index=False)
df.to_sql('df_orders',con=conn,if_exists='append',index=False)
```

✓ 1.1s

After the execution of this we must be able to see the table in the database.

```
select * from df_orders
```



Results | Messages

order_id	order_date	ship_mode	segment	country	city	state	postal_code	region	category	sub_category	product_id	quantity	discount_amount	sale_price	profit
16	2022-06-18	Standard Class	Home Office	United States	Fort Worth	Texas	76106	Central	Office Supplies	Binders	OFF-BI-10000756	3	0.00	0.00	0.00
17	2022-02-04	Standard Class	Consumer	United States	Madison	Wisconsin	53711	Central	Office Supplies	Storage	OFF-ST-10004186	6	20.10	649.90	39.90
18	2023-08-04	Second Class	Consumer	United States	West Jordan	Utah	84084	West	Office Supplies	Storage	OFF-ST-10000107	2	2.40	57.60	-2.40
19	2022-01-23	Second Class	Consumer	United States	San Francisco	California	94109	West	Office Supplies	Art	OFF-AR-10003056	2	0.40	9.60	-0.40
20	2022-01-11	Second Class	Consumer	United States	San Francisco	California	94109	West	Technology	Phones	TEC-PH-10001949	3	6.30	203.70	33.70
21	2022-10-05	Second Class	Consumer	United States	San Francisco	California	94109	West	Office Supplies	Binders	OFF-BI-10002215	4	0.40	19.60	-0.40
22	2023-07-16	Standard Class	Corporate	United States	Fremont	Nebraska	68025	Central	Office Supplies	Art	OFF-AR-10000246	7	0.80	19.20	-0.80
23	2023-05-06	Standard Class	Corporate	United States	Fremont	Nebraska	68025	Central	Office Supplies	Appliances	OFF-AP-10001492	7	1.80	58.20	8.20
24	2023-05-21	Second Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Furniture	Chairs	FUR-CH-10002774	2	2.80	67.20	7.20
25	2023-02-24	Standard Class	Consumer	United States	Orem	Utah	84057	West	Furniture	Tables	FUR-TA-10000577	3	52.00	988.00	88.00
26	2022-06-20	Second Class	Consumer	United States	Los Angeles	California	90049	West	Office Supplies	Binders	OFF-BI-10001634	2	0.50	9.50	-0.50
27	2022-02-08	Second Class	Consumer	United States	Los Angeles	California	90049	West	Technology	Accessories	TEC-AC-10003027	3	2.70	87.30	7.30
28	2023-12-11	Standard Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Furniture	Bookcases	FUR-BO-10004834	7	92.40	2987.60	247.60
29	2022-08-21	Standard Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Office Supplies	Binders	OFF-BI-10000474	2	0.50	9.50	-0.50
30	2022-08-14	Standard Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Furniture	Furnishings	FUR-FU-10004848	3	4.80	115.20	15.20
31	2022-11-20	Standard Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Office Supplies	Envelopes	OFF-EN-10001509	2	0.00	0.00	0.00
32	2023-01-19	Standard Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Office Supplies	Art	OFF-AR-10004042	6	3.60	86.40	16.40
33	2023-01-30	Standard Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Office Supplies	Binders	OFF-BI-10001525	6	0.50	9.50	-0.50
34	2022-06-03	Standard Class	Consumer	United States	Philadelphia	Pennsylvania	19140	East	Office Supplies	Art	OFF-AR-10001683	2	0.80	19.20	-0.80
35	2022-09-28	Second Class	Home Office	United States	Houston	Texas	77095	Central	Office Supplies	Paper	OFF-PA-10000249	3	0.90	29.10	-0.90
36	2022-10-30	First Class	Corporate	United States	Richardson	Texas	75080	Central	Technology	Phones	TEC-PH-10004977	7	55.00	1045.00	25.00
37	2023-07-27	First Class	Corporate	United States	Richardson	Texas	75080	Central	Furniture	Furnishings	FUR-FU-10003664	5	7.60	182.40	2.40

Analysis:

--find the top 10 revenue generating products

```
select top 10 product_id, sum(sale_price) as sales
from df_orders
group by product_id
order by sales desc
```

-- find the highest selling 5 products per each region

```
with cte as(
select region, product_id, sum(sale_price) as sales
from df_orders
group by region, product_id
)
```

```
select * from (select *,ROW_NUMBER() over(partition by region order by sales
desc) as rn
from cte)A
where rn in (1,2,3,4,5)
```

-- find month and month comparision for the year 2022 and 2023 like Jan-2022 V/S Jan-2023

```
with cte as(
select year(order_date) as order_year, month(order_date) as order_month,
sum(sale_price) as sales
from df_orders
group by year(order_date), month(order_date)
        )
select order_month,
sum(case when order_year=2022 then sales else 0 end) as sales_2022,
sum(case when order_year=2023 then sales else 0 end) as sales_2023
from cte
group by order_month
order by order_month
```

-- for each category which month had highest sales

```
with cte as(
select category,sum(sale_price) as sales,format(order_date,'yyyyMM') as
order_year_month
from df_orders
group by category,format(order_date,'yyyyMM')
)
select category,order_year_month,sales from(select *,ROW_NUMBER()
over(partition by category order by sales desc) as rn
from cte)A
where rn=1
```

--which sub category has highest growth by profit in 2023 compare to 2022

```
with cte as(
select sub_category,year(order_date) as order_year,
sum(profit) as total_profit
from df_orders
group by sub_category,year(order_date)
        )
select top 1 sub_category,profit_2023-profit_2022 as profit_diff from
(select sub_category,
sum(case when order_year=2022 then total_profit else 0 end) as profit_2022,
sum(case when order_year=2023 then total_profit else 0 end) as profit_2023
from cte
```

```
group by sub_category)A
order by profit_diff desc
```

Links & References:

- Kaggle · GitHub
- Retail Orders
- Youtube Link
- pandas.read_csv — pandas 3.0.0 documentation
- Getting Started on Kaggle | Kaggle