Retail Orders Data Pipeline, Analysis and Visualization using Python, SQL and Tableau

1.Introduction

The project extracts, analyzes and visualize data from a dataset of orders of various products made at a retail stationery store by clients from different states.

The aim of the project is to build an ETL pipeline to extract & transform data in Python, load it to MS SQL Server for analysis and create a dashboard in Tableau.

The SQL Server analysis will involve running queries to answer the following questions:

i) Find the top 5 highest generating products.

ii) Find top 5 highest selling products in each region.

iii) Find month over month growth comparison for 2022 and 2023 sales.

iv) For each category, which month had the highest salaries?

The dashboard will provide an interactive platform to access the summaries of the data. Its key features will include a map to show sales per state and per region, a graph to indicate the month over month for 2022 and 2023 sales and a drop down menu to view comprehensive details of each product.

2.Data Extraction and Libraries Import

I will start by extracting the data from a csv file and loading packages I will use during this project.

```
import os
import pandas as pd
#os.getcwd()

#os.chdir('C:\\Users\\user\\Documents\\persona\\PyCharm')
retail_Data = pd.read_csv('orders.csv')

#os.getcwd()
retail_Data.head(6)         #getting the first 6 rows of the dataset
```

The 'Ship Mode' column has values not available, which is not type of a ship mode. Im going to check all the levels of ship mode:

```
retail_Data['Ship Mode'].unique()
```

Will convert 'Not Available', 'unknown' to nan values such that they can be counted as not a number, which is a recognized type of data.

```
#reload the csv file with corrected ship modes:
retail_Data = pd.read_csv('orders.csv',na_values = ['Not Available',
'unknown'])

retail_Data['Ship Mode'].unique()
```

3.Data Transformation and Pre-processing

I will check the structure of our dataset and clean it to prepare it for the loading to the SQL server.

I will change names of variables or data types and derive new columns or remove columns such that the data can comply with the expected standards of the SQL server management system

```
retail_Data.shape      #the number of rows and columns of our
dataframe.
```

The dataset has 9994 observations by 16 columns.

```
retail_Data.head()
```

I will change the names of the columns, to lower case and replace ' ' with _.

```
retail_Data.columns = retail_Data.columns.str.lower()
retail_Data.columns = retail_Data.columns.str.replace(' ','_')
retail_Data.columns
```

To be able to answer the SQL questions, I need to add new columns. The data for the new columns will be derived from the existing columns in our dataframe. I Will add 'discount', 'sale_price' and 'profit'.

```
retail_Data['discount'] = retail_Data['list_price'] *
retail_Data['discount_percent'] * .01

retail_Data['sale_price'] = retail_Data['list_price'] -
retail_Data['discount']

retail_Data['profit'] = retail_Data['sale_price'] -
retail_Data['cost_price']

retail_Data.head()   # the dataframe now has 3 added columns.
```

Next im going to convert the order_date from object data type to datetime data type

```
retail_Data.dtypes  #checking the data types of all the variables in
our data frame. 'order_date' is an object not a date.

retail_Data['order_date'] = pd.to_datetime(retail_Data['order_date'])
#converting 'order_date' to a datetime data type.

retail_Data.dtypes     #order_date is now has a datetime datatype
```

I will delete the 'list_price' and 'cost_price' columns in order to normalize our data for the SQL server.

```
retail_Data.drop(columns=['list_price','cost_price'],inplace=True)
```

4.Data Loading to Database (MS SQL Server)

The sqlalchemy module will be used to connect the python environment to the SQL server.

The retail order data will be transferred to the MS SQL Server Manager to run queries and analyze the data

```
import sqlalchemy as sal    #will use the sqlalchemy for the
connection

engine = sal.create_engine('mssql://DESKTOP-P2K3MK7\SQLEXPRESS/master?
driver=ODBC+DRIVER+17+FOR+SQL+SERVER') #setting the connection
parameters.
conn=engine.connect()

#DESKTOP-P2K3MK7\SQLEXPRESS

retail_Data.to_sql('retail_orders', con=conn , index=False, if_exists
= 'append')
```

5.Conclusion

In conclusion, the data analysis conducted on the retail orders has provided valuable insights into product sales trends. By effectively connecting the dataset to a SQL database, we enhanced our ability to query and manipulate the data, allowing for more comprehensive analyses. Building an interactive dashboard in Tableau with a comprehensive interface enabled efficient management and analysis of customer purchases.

The findings highlight key patterns in purchasing behavior, such as highest selling products, sales per region which can inform inventory management and marketing strategies. Overall, this project underscores the importance of data-driven decision-making in the retail sector.

Future work could involve predictive analytics to optimize business strategies and improve customer engagement.