CS532: Final Project Report

Project Title: Spend Pattern and Fraud Detection Analysis of a Transaction Dataset.

Team Member(s): Janaki Ram Chimata , Venkata Sai Karthik Tirumalasetty.

I. Problem

The transaction dataset is analyzed to unwind various unknown trends and patterns involving three different analytical tasks. First one is a fraud pattern analysis detecting the frequency of the fraudulent activity happening at a given date and time. Second analytical task is based on identifying the patterns in airline ticket sales over the last two decades among different modes of transaction. Another analytical task deals with most preferred type of transaction among the legitimate transactions involving MCC in a state. Database is taken from openly available transactions data made by IBM. This dataset is then converted into a No-SQL database with JSON document form with over 24 million entries. This database is coupled with publicly available ISO listed merchant category codes to classify business with different retail financial services. This MCC (Merchant Category Code) is used in the various analytical data tasks mentioned above. The database consists of various attributes of transaction like amount, date and time of transaction, type of transaction, merchant city and state along with zip code and MCC.

II. Software Design and Implementation

A. Software Design and NoSQL-Databse and Tools Used

Database used is MongoDB which is a No-SQL database. This database with JSON document entries is loaded into the local system using MongoDB compass application and connection is established through the local host server instance running on the system. Python programming language is used along with modules like pymongo, openpyxl, mongo client, numpy and matplotlib to perform analysis. Visual Studio Code is the code editor used for coding and installing all these packages. Database is accessed through the code with client created from the mongo-client using local host connection.

To perform the analysis, we run different operations and filters on the database. All these queries are stored in the pipeline in a specific order and then this pipeline is executed to extract the needed specific information. Pymongo in the python module supports all these kind of pipelining queries like distinct, aggregate, split, unwind, limit, group , sort etc.
These give us the flexibility to run and analyze different queries over such a huge data. Runtime of each query is dependent on the pipelining inputs and different filters used. Generated output is shown visually using different plots like scatter plot, bar graphs, pie charts etc. available through matplot and pyplot libraries.

B. Parts that you have implemented

Analysis 1: Fraud pattern analysis done by Janaki Ram.
The dataset has a certain entry which are labeled as fraud transactions. The query is based as follows, when a fraud is done in online transactions, we will analyze the patterns of those activities based on which time of the day and when there is more frequency of the activity. It's evaluated as follows, firstly DB is filtered with online transactions and then among fraud entries, we group the data with date and time entries. A split query is run on time to split hours and minutes and then hours is used as one of the parameters along with a day to find frequency of the fraud activity. All these outcomes are compared and plotted as a scatter plot. From the outcome, we can specify that for a given date, which hour of the day has seen more fraud activities and can also derive the information about which combination of date and time gives us the more frequency of fraud activities. Another sub-query is generated as a pie chart which shows the percentage of frauds happening with each of the three different transaction types like online, chip and swap present in database. This analytical task helps us to identify similar and peculiar pattern about the frequency and time of fraud activities and helps us to narrow down the hours when monitoring is required to reduce the frauds.

Analysis2. Airline tickets spend analysis done by Karthik.
This task deals with air ticket spent patterns done through various modes of transactions over the period of last two decades and shows the ticket sales of different airlines in the year 2020. The query detailing is as follows: there are many mcc codes in the transaction data, referencing different retailers. First, we map the transaction data with mcc code of the retailers by fetching those values from mcc data available in a different file. Then these mcc values are filtered down to airline data by using some text matching. Now the database is narrow downed to just the transactions that involve mcc referencing airline sellers which means that we now just have the air ticket sales data. A new database collection is created to store this data and now analyzed through multiple filters and queries run through pipeline to show us the trends in air

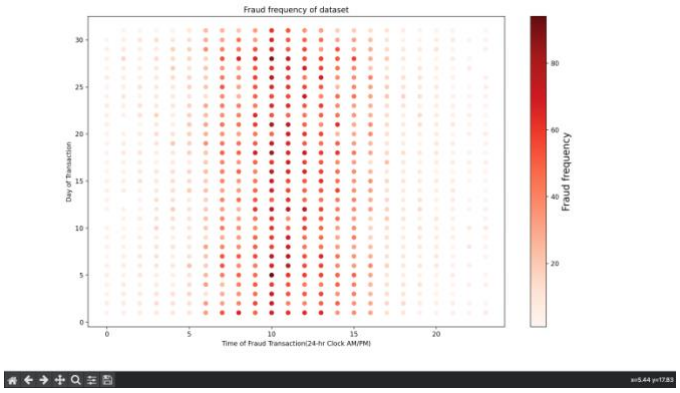
ticket booking over the last two decades among three different transaction modes available. Creating this new database for airline data helps us in making queries run faster here in this case and to use new aggregate pipeline techniques like split, project, convert to extract unknown information. This outcome is visualized as a bar graph and another query is created that compares ticket sales of different airline providers shown in a horizontal bar graph.

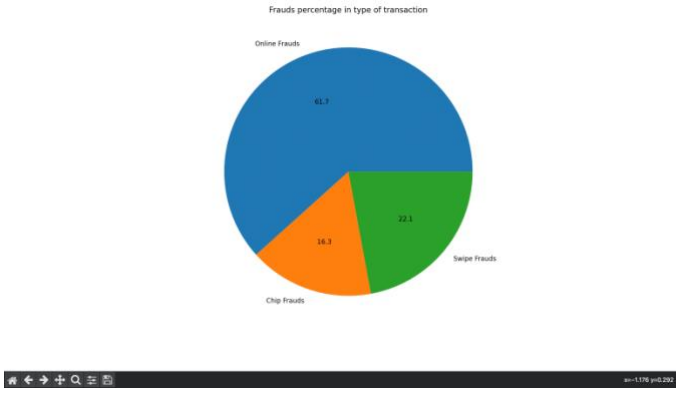
Analysis 3: Spend Pattern analysis on MCC– done as common Task.

Among the non-fraud transactions, merchant state with more transactions is analyzed on which mcc is preferred more by users over the last two decades. The database is filtered on by grouping by states and state with more transactions is considered as a parameter for the analysis here. For those state specific data values, we analyzed on which mcc category has the most sale made in each of the years over period of last 20 years. This analysis helps us to identify the retail category that is involved in maximum number of transactions. Various pipeline queries like limit, sort is used for analysis. Outcome is visualized as a pie chart to compare.

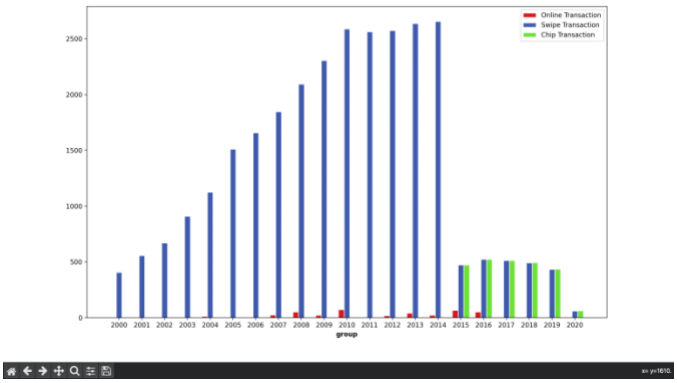
III. Project Outcome

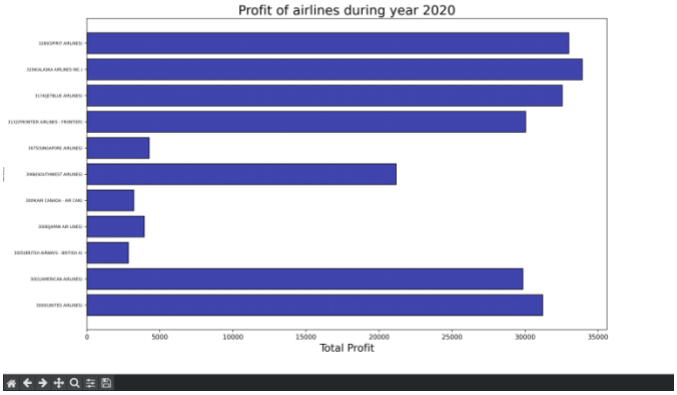
Analysis 1:



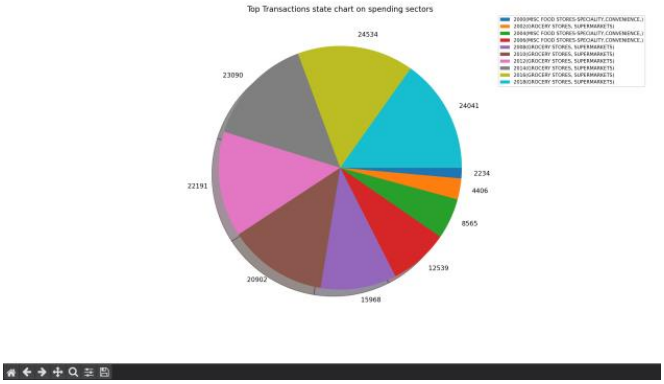


Analysis 2:





Analysis3 :



IV. References

[1] https://matplotlib.org/

[2] https://www.dm.usda.gov/procurement/ccsc/docs/pcref/MCCandBOCCcrosswalk1-27-10.xls

[3] https://www.geeksforgeeks.org/reading-excel-file-using-python/

[4] Data set refrenced from here and converted into json no sql db using cli commands:https://www.kaggle.com/datasets/ealtman2019/credit-card-transactions?select=credit_card_transactions-ibm_v2.csv