**Python Project**

**Group Members:**

**Ziqi Sheng, Xuelai Si**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Python Project

1. **Project Description**

This project is trying to design a new inventory management policy for the gas station. Helping them to use data-driven decision-making process to minimize the total cost while maintaining an excellent customer service level.

1. **Data processing**
2. **Data cleaning:**
   1. Invoice missing value:

A picture containing text, electronics, insect, display

Description automatically generated

In the file of ‘Invoice.csv’, there are 42 records missing from total 2873, we choose to drop these rows as it’s impossible to predict the number of fuels buying for each gas station. And, we are mainly using the Invoice data to get the price of the fuel for each day.

* 1. Fuel Level Part1 missing value:

By checking the data, there are two missing values in the column of ‘Fuel Level’. We use forward fill to fill the cell as the fuel level is a data to check the situation of the tank. The fuel level is shown with the interval of 15 minutes.

* 1. Fuel Level Part1 wrong data:

While checking the Fuel Level Part1 data, we find that in the column ‘Tank ID’, there are two cells not to be ‘T 12’ but to be ‘T12’, it will cause problem when we merge the dataset. Thus, we replace two cells from ‘T12’ to ‘T 12’.

* 1. Replace fuel type ‘U’ to ‘G’, Eliminate tank of fuel type ‘P’:

In Invoice File, the fuel type is ‘G’ and ‘D’. In Tank File, the fuel type is ‘U’,’D’,’P’. We eliminate the tank ‘T 13’ of fuel type ‘P’ as there is not any invoice for fuel type ‘P’. And then, fuel ‘U’ and ‘G’ are approximately the same thing. Thus, we replace fuel type ‘U’ to ‘G’ from the Tank File.

1. **Explore the Dataset**

While exploring the dataset, we first want to validate several questions:

1. Does the fuel price for each gas station is the same in one day?
2. How do gas stations purchase fuel? Does it utilize the purchase discount policy?
3. **Does the fuel price for each gas station is the same in one DAY?**

We first merge Invoice.csv with Location.csv to get the Gas Station name In the Invoice.csv, we use column ‘Gross Purchase Cost’ to divide from ‘Amount purchased’ to calculate the price of the fuel at different invoice date, different gas station and different fuel type. We add another column ‘Price’ to represent the price. Here is a brief picture to show the new dataset.

Table

Description automatically generated

Then we visualize the fuel price plot for both fuel type ‘G’ and ‘D’.

Chart, histogram

Description automatically generated

From the graph, we can see that all stations has shown a constant line for fuel price with the Invoice date. The slight difference are due to the difference among frequency of orders for different stations. Gas stations with higher frequency will show a better clear price trend plot. The frequency of orders for different stations are shown below.

Chart, bar chart

Description automatically generated

1. **How do gas stations purchase fuel? Does it utilize the purchase discount policy?**

We use scatter plot to check the distribution of purchased amount for each invoice order at different stations.

Chart, scatter chart

Description automatically generatedFrom the eight scatterplots, we can see that only Station EastMount has utilized the discount policy for amount purchased higher than 15000 and 25000 for fuel type G. For other stations, their purchase for fuel are normally less than 15000 for each order, that has waste a lot of money by not utilizing the fuel order.

But how does this problem come? Let’s check how each station and each tank has maintained their fuel level from 2017-01 to 2019-09.

Chart

Description automatically generated with medium confidence

Here is the plot of fuel level change for each tank in each station. Each tank has shown a different pattern and doesn’t follow a specific threshold to hold the fuel level. That cause the problem that there is a very high variance for stations to order the fuel. The minimum of order of fuel can be 3 liter, while the maximum of fuel can reach 38000 liters. Here we want to figure out a way to help stations regulate their invoice order rule and fuel level maintenance.

1. **Solve business questions**
2. **New inventory management policy for each location**

We will release our new inventory management policy first by setting safety bound and injection level for each tank. That two value means that after the fuel level of the tank is lower than the safety bound, we will inject the tanks in the stations who has the same fuel type to their injection level. For instance, ‘T 12’ and ‘T 15’ are both in station EastMount with same fuel type ‘D’. The T 12 has fuel level to be 19000 and T 15 has fuel level 22000. In this situation, we will has an invoice order to order (35000-19000)+(35000-22000)=29000 to recharge these two tank to its corresponding injection level. That helps the station to order more fuel in one time to reach the discount policy level. Here is the new inventory management policy setting.

Table

Description automatically generated

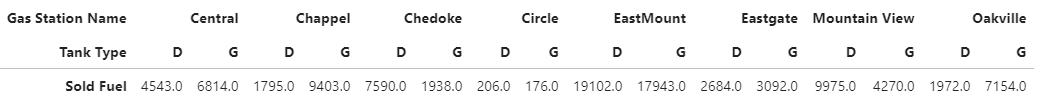
Here we are going to explain how we set the safety bound and injection level for each tank in different stations.

We first concat Fuel\_part1.csv and Fuel\_part2.csv together as each file contain part of the fuel level information of the tanks. We want to know for each time how many fuels has been sold for each tank from the new dataframe df\_fuel. The main algorithm is to check whether the fuel level has decreased from the previous time stamp, if so, we record this information and make it to be a new dataframe. For instance, we have two records of tank ID ‘T 12’ which has 27161.0 liters at 1/1/2017 0:10 and 27017.0 liters at 1/1/2017 0:25. We will take a record of tank ID ‘T 12’ sold 144 liters at 1/1/2017 0:25. We save these records to a new dataframe customer\_usage and visualize it.

Chart, histogram

Description automatically generated

We can see that for some of the stations, they have some emergency situation that they sold a lot of liters in a constant time. For our new inventory management policy, we need to consider these situations to make sure that there is always enough fuel to sell to customers. Here we get a pivot\_table to get the maximum sold fuel for each station with different fuel type.



Here we mainly use the maximum value \*1.1 and get it close to the nearest integer number. For instance, we can see EastMount with Fuel ‘D’ to has maximum sold 19102.0. Thus, we set the safety bound for these tanks to be 20000 and set the injection level to be 35000 to fulfill the discount policy ‘15000<x<25000’.

Now we are going to calculate the total cost with the new inventory management policy. The first thing is that we need to make sure the price of fuel for each day. As invoice only record the date without hour and minute setting, we are calculating the average of each fuel type in different date to be its price number. (price\_day)

As in customer\_usage, we has timestamp with hour,minute and second setting, we add another column ‘Normalized\_time’ to match it with the price table(price\_day).

Then we get the initial inventory from the df\_fuel by finding the earliest time fuel level.

Finally by using our new inventory management policy to make the simulation, we spent only 26617897$ for the three years, saving 6680829$ from previous management policy. We only use 1328 invoice orders comparing to the previous 2831 invoice orders. And also we has plenty of fuel maintained at the end, shown below.

Chart, bar chart

Description automatically generated

When we are visualizing our invoice order distribution, we can see that most of the orders has satisfied the discount policy. Saving plenty of money from this discount policy.

Chart, scatter chart

Description automatically generated