Profitability Enhancement for Fuel Delivery Vehicles, Using Geoanalytics, Data Mining and Kafka Streams

Elham Shakeel Inamdar
Sinhgad Academy of Engineering
Pune , India

Aashna Sanjay Sethi
Sinhgad Academy of Engineering
Pune , India

Anaisa Sam, Swamy
Sinhgad Academy of Engineering
Pune , India

Alaukik Pawan Chauhan
Sinhgad Academy of Engineering
Pune , India

Abstract:- Fuel tanker trucks carry petroleum in its crude oil form from refineries to Retail Outlets (RO) when no other means of transport such as pipeline or rail is available or feasible. These trucks transport fuel regularly. The regular routes being followed by these trucks are inefficacious and obsolete. The truck driver can have a considerable impact on fuel consumption. On average, fuel equates to about 30% of the total operating costs. With reduced fuel usage by optimal routing, operating costs and travel time can be reduced. Also, these trucks require fuel for transportation, which is not readily available at the RO stations for refueling. Possible high Retail Selling Prices (RSP) at certain ROs may affect the revenue overall. In this paper, we present a strategy to implement profitability enhancement for these customers (fuel tanker trucks) with the use of GeoAnalytics, Kafka streams, and Data mining. From the dataset acquired we provide the truck drivers the shortest and most efficient route, via a portal. Observing the previously recorded fueling patterns of these trucks, the RO stations are alerted, via email, to maintain enough fuel for re-fueling these trucks, which is not readily available. The truck drivers are informed which RO station is the nearest available for refueling according to varying RSPs. This strategy results in saving fuel, revenue and travel time of these truck drivers.

***Keywords:-** Retail Outlets(RO), Geoanalytics, Retail Selling Prices (RSP), Kafka Streams, Optimal Routing, Fueling Patterns.*

I. INTRODUCTION

A significant amount of fuel gets transported via fuel tanker trucks. Pragmatically, a single truck takes one-third of the total amount of fuel that one rail tank can take, which implies that it costs almost up to three times more to transport a barrel per mile. Hence to optimize the total revenue for this system is imperative. The transportation of fuel is led by following traditional routes and inadvertent fueling points with varying RSP's. The number of drawbacks in the existing system include the inadequate amount of fuel usage, longer inefficient routes, oblivion of next fueling point, delay in transportation, and increase in total operational cost. We aim to provide the shortest route possible, with the least amount spent on fueling the tank truck. To achieve this, we have used GeoAnalytics, Data Mining, and Kafka Streams. Kafka Streams is a client library for building streaming applications that processes data continuously in real-time. Real-time processing includes reading data from a topic (source), performing analysis or required transformation and then writing the results back to another topic Here we are using Docker to store the required data from dataset and Kafka Producer into the database. Once the dataset is acquired, it is pre-processed. Data Mining is used to pre-process or cleanse this data. Data in the raw form is often incomplete, inconsistent, erroneous and contains missing entries. This means that any inconsistent data needs to be converted into a suitable format. Dataset accuracy is very important as it determines the efficiency of the output. Pre-processing of the dataset is the first, and the most important step. Once the dataset is pre-processed, we can use it for further mathematical calculations in algorithms. The result data from the calculations is used as a source input for Geo Analytics. Geo Analytics provides visual aid. Here we have used PowerBI as a GeoAnalytics tool. We have the dataset in the form of the distances calculated, RSPs at the ROs, the latitude and longitude of the retail outlets and the start and endpoints of fuel tank trucks. Geo Analytics enables us to view this dataset on a map so that we can gain insights, which might otherwise have been missed. Thus, it is easy to view all possible routes, containing the retail outlets. Consequently, we can recommend a route that is the shortest in length and lets the driver encounter only those ROs with the least RSP. Based on the fueling patterns, we recommend an RO for fueling the vehicle. The respective RO is alerted, via email, for the approaching truck for easy availability of fuel. This is repeated if fueling is required for the rest of the trip. The truck driver can access the optimized route by logging on to a portal.

II. RELATED WORK

In the existing system, due to obsolete data processing and data visualization techniques, the data filtering and analysis were delayed and inefficient. Predicting fueling patterns, shortest routes and optimizing the cost of the transportation were simply not investigated. Also, with the ever-evolving technology, the use of live tracking via GPS (Global Positioning System), sensors, etc. would rule out 'Profitability enhancement' by increasing cost in several numbers, thus affecting the total expenditure. A review of previous studies related to systems that we have referred to and revamped in our system are discussed in the following sections.

A. The Study and Implementation of Mobile Gps Navigation System Based on Google Maps

In this paper [1], with the use of Assisted global positioning systems (A-GPS) by mobile companies the use of navigation systems demonstrates functions such as Google map browse and query, bus lines search, the rapid local positioning on a mobile phone. It uses GPS to get the geographical information and is adopted by many mobile-based companies due to its high location accuracy. By just entering the destination location and using a GPS satellite connection to determine the current location, the Maps trace the path as we advance with voice-guided directions on each turn. The app displays the user's progress along the route and gives instructions for each diversion. This application is highly precise and accurate with its results.

But the limitation here is that it is highly network-dependent to load or trace routes on a map. The user's location information is available to Google, which potentially can be misused. Also, it always shows the fastest route from source to the destination, which may or may not always be efficient - for example, it sometimes makes you drive through a bad neighborhood or bad condition roads.

B. Building Linkedin's Real-Time Activity Data Pipeline

In this paper [2], Apache Kafka is referred to as a centralized data pipelining system. Kafka solves a general problem of delivering a huge volume of data to diverse subscribers. Each message is delivered on a topic. Producers of Kafka send messages to a topic that holds the logs of a type. Each topic is spread over a group of clusters of Kafka Brokers. A Kafka Broker is the host for zero or more partitions of each topic. A Zookeeper manages these groups and assigns partitions.

The system is designed to handle billions of messages on less than a thousand topics.

The disadvantages of Apache Kafka are that it doesn't attempt to prevent data loss. In cases like server failure, several messages can be lost before they are transferred to the Kafka Broker or flushed into a disk. Even after a system reboot, these messages are not retracted.

C. Remote Sensing and Geographic Information Systems (Gis) for Developing Countries

This paper [3], states GIS (Geographical information systems) is a system which contains tools for problem-solving, decision making and visualization of data on a map. It states that lack of data is a major obstacle towards a sustainable environment. The remote sensing instruments collect data at various spatial and radiometric resolutions. Images from these instruments or tools have provided a proven cost-effective solution for collecting accurate, and reliable data on spatial resources. This data on maps can give the government authorities an initial insight into their country for optimal decision making.

The limitations of this paper are the lack of data and information to identify the country-specific needs, lack of local expertise to operate and maintain large-scale GIS systems and its high network dependency. GIS layers can cause costly mistakes which are complex with hidden errors. Also, these tools are costly, for which financial resources are not easily available.

D. Opportunities for Automating Email Processing: A Need-Finding Study

This paper [4], states a series of need-finding probes regarding email automation and its need today. A system that lets users write their own email processing rules called YouPS along with methods that manipulate them was designed. The rules written in Python are executed on IMAP. Users can write different rules for every mode on which the inbox behaves differently depending on the current mode. Users can enable and disable different email rules as per their convenience. They were also successful to have identified several captivating categories of needs that were not known or not labeled. This addressed the need for a new interface to send automated emails with ease.

III. METHODOLOGY

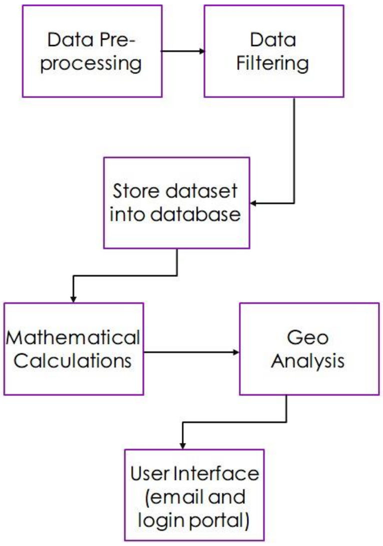
Data Pre-processing
Data Filtering
Store dataset into database
Mathematical Calculations
Geo Analysis
User Interface (email and login portal)

Fig 1:- Workflow

The methodology of implementing this system involves five steps : (1) Data Pre-processing (2) Filtering data using Kafka streams (3) Storing data in the database (4) Mathematical calculations based on distance and mileage (5) Importing dataset into Power BI (6) Data pattern analysis using Geo analysis (7) Displaying results using UI (login portal and email system)

Data pre-processing is the most important step for this project. Since the output is highly dependent on the dataset, we must ensure its accuracy. In this stage, we convert all abstract data into something meaningful. This is done in the following steps:

- ➢ Input the data file into the python file using ‘read_csv’ method from pandas.
- ➢ Convert this file into a data frame using pandas.DataFrame() method.
- ➢ Using the map() method, convert all string entries into integer.
  E.g.:
  data.Education=data.Education.map({‘Graduate’:1,’Not Graduate’:0})
- ➢ Use the isnull() method to check which columns contain null values.
- Replace the null values with the mean of the column using the numpy.mean() method.
- In case the column values are not an integer, replace the null entries with the median of the column using the numpy.median() method.

After pre-processing the data, we must filter it. When we receive the dataset, it contains all types of values. Some of the columns of this dataset are not relevant to finding the output. Thus, only the apposite columns are inserted into the database. This filtration is done using Kafka Streams. First, the data is stored into an entity known as ‘Producer’. Here, data is stored in NOSQL format as a topic. This acts as the source for Kafka streams. Once data is saved in the Producer, we transfer it to Streams for filtration. Streams does the job of a server. We write the filtration logic in the build_topology() method of Streams. Producer topic data is manipulated, and the result is stored in a new topic. This topic is stored in the database via Docker.

Docker is also a server which acts as a transfer medium between Kafka Streams and the database – PostgreSQL. To set up docker, we write the ‘docker.yml’ file which contains all the docker configurations like advertising host, broker IP address, etc. Then, we run ‘docker-compose up postgresql’ on the command prompt. This starts the docker machine. We can view the API via the IP address. Since we are running docker on windows, we are using Docker Toolbox and the IP address is ‘192.168.99.100’. This is address is used in place of localhost because we are using a virtual machine to run Docker. The port number is 3030. Thus, typing ‘192.168.99.100:3030’ on the web browser opens the Docker API. Using Docker, we set up connectors that will help us transfer data from the topic into a table in the database. We use the postgres sink connector for data transfer from the topic into the database. To do this, we add the configurations in a properties file with the extension ‘.properties’. Here we write the setup details for the sink. Now that the sink is set up, the docker machine is restarted and the flow of data begins. As and when changes occur in the topic, it is reflected in the database, i.e. real-time data transfer. The possibility of server crash or failure is minimum. However, in such a case, data transfer is interrupted. Once the connection is re-established, the data transfer resumes from last transfer point – does not restart. This is a huge asset to the project as it prevents data redundancy in the database.

Therefore, we have ensured secure and clean data in our database.

Now, we will perform mathematical calculations on the dataset. The calculations are made as follows:

- ➢ *Calculate total travel distance using Haversine’s formula.*

- ➢ *Calculate the petrol that will be needed for the entire journey*
- petrol_needed = mileage * distance_to_be_travelled
- if petrol_needed >= petrol_present then,
- ✓ travel from start to end location without re-fuelling
- else
- ✓ Calculate the distance to the nearest (and cheapest) RO
- ✓ Find out the petrol needed to reach this RO
- ✓ Assign the above value to petrol_present. The truck begins with this amount, plus a small buffer quantity.

- ➢ *Calculate distance between current RO and destination.*

- ➢ *If petrol_needed for above distance is >= petrol_present then,*
- travel from start to end location without re-fuelling

- ➢ *else*
- Calculate the distance to the nearest (and cheapest) RO
- Find out the petrol needed to reach this RO
- Assign the above value to petrol_present. The truck begins with this amount, plus a small buffer quantity.

- ➢ *Repeat steps 3-5 until the journey is complete.*

The above data is stored in a file. This will be used for Geo Analysis.

Once we have performed necessary calculations, we can now proceed towards Geo Analytics. Using this technique, we can generate multiple insights from our dataset. Geo Analytics entails plotting the dataset onto a map. We plot the distances calculated, RSPs at the ROs, coordinates of the Ros and coordinates of the journey (start and end locations). Thus, we can visualize all four entities on one map, making it much easier to correlate the data and make conclusions. Power BI is a software that enables us to use GeoAnalytics. Hence, we can conclude the shortest and cheapest route by viewing the routes on the map.

The outcome is a map, that has the suggested route highlighted on it.

Firstly, the user will log in into the portal by providing username and password which is authenticated.

Now the dashboard will show a source search box and a destination search box. The user is required to fill in the necessary information. After clicking on the search button, the result is displayed. This consists of the proposed route which has been processed by React JS in the form of a file with the ‘.qvf’ extension (of Microsoft Power BI) to html or any other browser format to display.

Using the login portal, the user can enter the start and destination of his journey. He will then be able to cognize the proposed route. The user can take a printout of the same and carry it with him on the journey.

We analyze fueling patterns of the RO as well. In case there is a shortage of fuel at any RO and a truck is approaching for refuel, it is our responsibility to intimate the RO of such an event.

This is done via email. Fullstack technologies consist of Node JS, React JS, Mongo DB, Angular, and Express JS are used to develop user interface for better user experience as these are very efficient and responsive and reliable as compared to other UI development technologies. Node JS is a runtime environment for JavaScript applications. Node JS can handle multiple requests at a time. We are using Node JS, Node mailer service for handling real-time notification service. Nodemailer is an Express JS service API that is integrated with the code and we will modify the same to make the email alert system automated. Nodemailer takes the recipient email I'd and then by using a predefined custom email template, it will send the information to the RO and alert it. It can handle the size of the information automatically. Express JS we are using it as an API provider for React JS and provides connection of database to the client-side. And finally, React JS which serves as a client-side in our user interface. It does all the calculation and analysis work.

We alert the RO in great advance to the arrival of the truck, which gives sufficient time to maintain the germane quantity.

We have thus ensured that the driver need not rely on any network-based services. This is a crucial aspect because in certain areas, due to weak network strength, internet guidance may not be the best option for assistance. We can track the drive via calculations – offline. We are aware of the time spent on the journey, as well as the average speed. Using the simple speed-distance-time formula we can estimate the truck's location since we are acquainted with the exact route that it has followed. Therefore, we guarantee that the cost of transport is under control, and wastage of resources is negligible. This greatly enhances the profitability for the delivery company and the money that was saved can be put to better use.

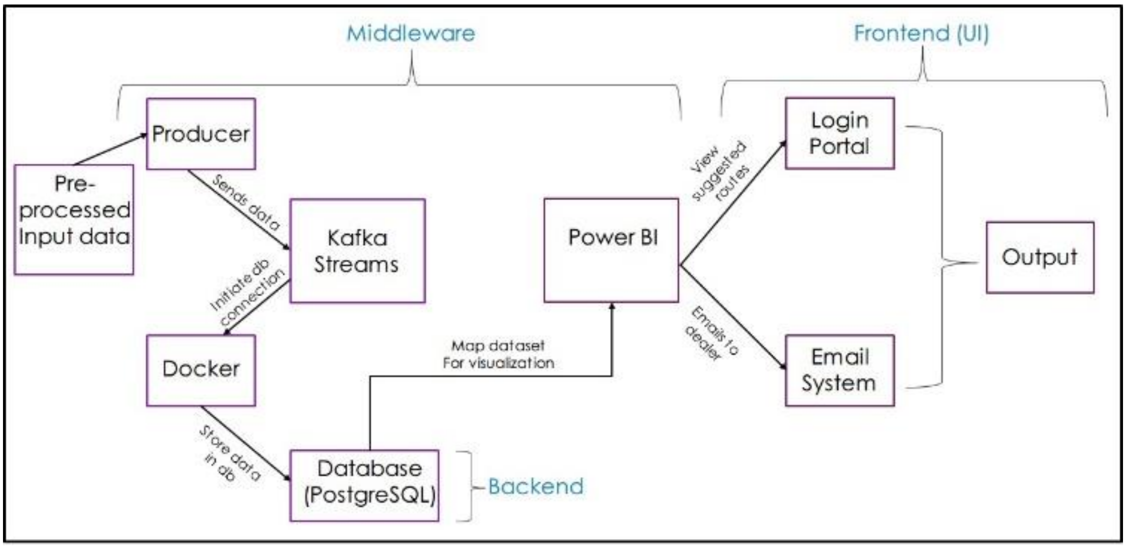

Fig 2:- System Architecture

Our aim here is to build a less network-dependent system that does not support live tracking to avoid theft and fraud by the hacking of system information. Thus, we propose a system which can reduce the cost of the model and reduce the initial setup cost, which is otherwise high, as is in the above-mentioned applications. Also, our system is less prone to errors and requires minimal cost fixation.

IV. CONCLUSION AND FUTURE OUTLOOK

This paper presents a well-planned solution for the transportation of fuel by tanker trucks which makes the system cost-effective by saving expenses, avoiding delay and reducing the overall expenditure With the use of Kafka streams, we can filter the provided datasets. But it supports only unbounded streams and with normal latency. Flink is an alternative that can be used for future references. It supports both bounded and unbounded streams. Also, it has higher speed comparatively to Kafka streams due to its architecture and cluster deployment. With GeoAnalytics we can plot the points on the map which will provide us the route patterns according to varying RSP's. The customer will access the route provided via the admin portal. We have used an email system to alert the RO's. In the future, we can use the SMS system instead for faster reference. We are using full-stack technologies on the client's side to enhance the user experience and better efficiency of the application. Node JS services like Nodemailer is used to alert the RO, with the help of email notifications. With the above strategy, we can implement maximum profitability for the customers.

REFERENCES

[1]. Proceedings of the European Control Conference 2009 Budapest, Hungary, August 23–26, 2009 “Combined Time And Fuel Optimal Driving On Trucks Based On Hybrid Model” by Benjamin Passenberg, Peter Kock, and Olaf Stursberg. Kakan Dey, Amy Apon, Andre Luckow, and Linh Bao Ngo

[2]. 2017 IEEE International Conference on Big Data (BIGDATA) “A Performance Evaluation of Apache Kafka in Support of Big Data Streaming Applications” by Paul Le Noac’h, Alexandru Costan, Luc Boug´e.

[3]. “Building a Replicated Logging System with Apache Kafka” by Guozhang Wang, Joel Koshy, Sriram Subramanian1, Kartik Paramasivam, Mammad Zadeh, Neha Narkhede, Jun Rao, Jay Kreps, Joe Stein.

[4]. 3rd International Symposium on Computational Intelligence and Intelligent Informatics - ISCIII 2007 “ GIS Intelligent Solution” by V. Lupu, C. Lupu.

[5]. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering “Building LinkedIn’s Real-time Activity Data Pipeline” by Ken Goodhope, Joel Koshy, Jay Kreps, Neha Narkhede, Richard Park, Jun Rao, Victor Yang Ye LinkedIn.

[6]. 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC) “Programming challenges of Chatbot: Current and Future Prospective” by AM Rahman, Abdullah Al Mamun, Alma Islam.

[7]. 2002 IEEE “Remote Sensing and Geographic Information Systems (GIS) For Developing Countries” by M. Mohamed and R. Plante.

[8]. 2019 Association for Computing Machinery “Opportunities for Automating Email Processing: A Need-Finding Study” by Soya Park, Amy X. Zhang , Luke S. Murray, David R. Karger.

[9]. Ieee Journal Of Selected Topics In Applied Earth Observations And Remote Sensing “Building Change Detection Using High Resolution Remotely Sensed Data and GIS” by Natalia Sofina and Manfred Ehlers

[10]. Proceedings of the 2018 IEEE IEEM “Contractual Barriers and Energy Efficiency in the Crude Oil Supply Chain” by R.O. Adland, H. Jia.

[11]. Ieee Transactions on Intelligent Transportation Systems “A Distributed Message Delivery Infrastructure for Connected Vehicle Technology Applications” by Yuheng Du, Mashrur Chowdhury, Mizanur Rahman,

[12]. 2015 INTERNATIONAL CONFERENCE ON TRANSPORTATION INFORMATION AND SAFETY (ICTIS) “THE EVALUATION AND TEST METHODS FOR POWER PERFORMANCE AND FUEL ECONOMY FOR ARTICULATED VEHICLE TRAINS” BY Dong Jinsong , Zong Chengqiang ,Zhang Hao , Zhang Xueli , Zhang Hongwei , Ou Chuanjin , Zhou Gang.

[13]. 2018 international conference on information , communication, engineering and technology (icicet) “a study of apache kafka in big data stream processing” by bhole rahul hiraman ; chapte viresh m ; c karve Abhijeet

[14]. 2017 2nd International Conference on Telecommunication and Networks (TEL-NET) “KAFKA: The modern platform for data management and analysis in big data domain“ by Rishika Shree ,Tanupriya Choudhury ; Subhash Chand Gupta ; Praveen Kumar