Development of a Performance Monitoring Dashboard for Halan's SME Division

By

Mohamed Elfadil Abdalla

Omer Husham Ibrahim

Submitted in Partial Fulfilment of the Requirements for the Degree of

Bachelor of Science in Computer Science

at

Future University

March/2022

Thesis Advisor

Dr. Ashraf Osman

APPROVAL AND ACCEPTANCE SHEET

The (thesis/capstone project) entitled **"Development of a Performance Monitoring Dashboard for Halan's SME Division"** prepared and submitted by:

Mohamed Elfadil Abdalla

Omer Husham

In partial fulfillment of the course requirement for the Degree of Bachelor of Science in Computer Science has been examined and is hereby recommended for approval.

( )	( )
Panelist 1	Panelist 1

( )

Head Panelist

Accepted as partial fulfillment of the requirements for the **Degree of Bachelor of Science in Computer Science.**

Dr. Ashraf Osman Thesis Advisor	Dr Zainab SeidAhmed Course Advisor

Dr. Zainab SeidAhmed

Department Head

Date

ABSTRACT

Business Analysis (BA) is the practice of initiating change in an enterprise by assessing the needs and providing recommendations and solutions that are considered to be of value to the stakeholders. As a subset of BA, Business Intelligence (BI) could be defined and viewed as a set of tools presenting historical data for users to form data-driven decisions in a timely manner to stay competitive. However, these tools vary in the quality of information they provide. This thesis discusses the development of a performance monitoring dashboard through the implementation of the data science process, to provide decision makers at Halan Co. Ltd. with a better understanding of their organization's performance. The Researchers aim to determine the most suitable supervised machine learning algorithm that can provide the management with timely, quality insight. The developed dashboard provides high quality information using enhanced visualization and predictive tools. A Lasso regression model was successfully trained to predict the weekly trends of orders delivered, thus providing smart and actionable insights to the decision makers not previously available to them through traditional dashboards. The developed system is able to query data from the database and visualize the data that the stakeholder requested, both in tabular and graphical form. The dashboard is able to output daily, weekly, and monthly reports to track the organization's Key Performance Indicators (KPIs) to measure how the organization is achieving its goals such as the performance of drivers, individual small businesses, and the Small-to-Medium Enterprise (SME) branch. Using the developed model, stakeholders are able to compare the organizations performance against the predicted goals.

تحليل الأعمال هو الشروع في التغيير في مؤسسة ما من خلال تقييم الاحتياجات وتقديم التوصيات والحلول التي تعتبر ذات قيمة لأصحاب المصلحة. كمجموعة متفرعة من تحليل الاعمال ، يمكن تعريف ذكاء الأعمال وعرضه على أنه مجموعة من الأدوات التي تقدم البيانات التاريخية للمستخدمين لتمكينهم من تشكيل قرارات تعتمد على البيانات في الوقت المناسب للبقاء في المنافسة. ومع ذلك ، تختلف هذه الأدوات في جودة المعلومات التي تقدمها. تناقش هذه الأطروحة تطوير لوحة لمراقبة الأداء من خلال اتباع منهجية علم البيانات ، لتزويد صانعي القرار في شركة حالا المحدودة بفهم أفضل لأداء مؤسستهم. يهدف الباحثون إلى تحديد أنسب خوارزمية تعلم الي خاضعة للإشراف قادرة على تزويد الإدارة برؤية جيدة في الوقت المناسب. توفر لوحة المعلومات المطورة معلومات عالية الجودة باستخدام التصور المحسن والأدوات التنبؤية. تم تدريب نموذج انحدار بنجاح للتنبؤ بالاتجاهات الأسبوعية للطلبات المقدمة ، وبالتالي توفير رؤى ذكية وقابلة للتنفيذ لصانعي القرار حيث لم يكن هذا متاحا لهم من قبل من خلال لوحات المعلومات التقليدية. النظام المطور قادر على الاستعلام عن البيانات من قاعدة البيانات وتصور البيانات التي طلبها أصحاب المصلحة ، سواء في شكل جداول أو رسومات. لوحة المعلومات قادرة على إخراج تقارير يومية وأسبوعية وشهرية لتتبع مؤشرات الأداء الرئيسية للمؤسسة لقياس كيفية تحقيق المنظمة لأهدافها مثل أداء السائقين والشركات الصغيرة الفردية والشركات الصغيرة والمتوسطة. باستخدام النموذج المطور يمكن لأصحاب المصلحة مقارنة أداء المنظمات مقابل الأهداف المتوقعة.

Keywords: *Business Intelligence, KPI, Data Science, Machine Learning, Visualization*

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

LIST OF ABBREVIATIONS

BA	Business Analysis
BI	Business Intelligence
SME	Small-to-Medium Enterprises
SQL	Structured Query Language
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
WIP	Work in Progress
IT	Information Technology
KPI	Key Performance Indicator
DM	Data Mining
KDD	Knowledge Discovery in Data
ML	Machine Learning
AI	Artificial Intelligence
MLE	Maximum Likelihood Estimator
DF	Data Frame
DS	Data Science
EDA	Exploratory Data Analysis
API	Application Programming interface
UI	User Interface
FTP	File Transfer Protocol
CPU	Central Processing Unit
I/O	Input/Output
CSV	Comma-Separated Values
ETL	Extract, Transform, Load
OCI	Oracle Cloud Infrastructure
GMV	Gross Merchandise Value

Chapter 1

INTRODUCTION

This chapter introduces the research background, site, state of the current system, the problems faced by the system, objectives of the study, the system nature, scope and delimitation, methodology and the significance of the project.

1.1 Background of the study

Understanding how well the enterprise is performing is an essential requirement for effective and efficient decision making. Recently, Business intelligence (BI) tools promise to provide decision makers with the information they need for data-driven decision making. However, these tools vary in the quality of information they provide (Ahmed, 2021).

The purpose of management control is to increase the motivation and efficiency of managers and to create a greater convergence of objectives within an organization. It is a well-established management principle that what cannot be measured cannot be managed. However, it is equally true that what cannot be monitored cannot be well managed. This is where business dashboards come in. Living in the technological era, in which the world is changing and adapting to the complex systems of society, businesses must always be prepared with the tools necessary for adaptation (Dinesh, 2021).

If you ask 10 people who build business dashboards to define a dashboard and you will probably get 10 different definitions. But generally speaking and according to (International Institute of Business Analysis, 2015) a dashboard is a visual display of data used to monitor conditions and or facilitate understanding as you can see this is broad definition that might include an interactive display that allows people to explore worker compensation claims by region, industry, and body part, a PDF file showing key measures that gets e-mailed to an executive every Monday morning , a large wall-mounted screen that shows support center statistics in real time, or a mobile application that allows sales managers to review performance across different regions and compare year-to-date sales for the current year with the previous year.

Speaking more specifically a Dashboard is what we use to transform an organization's strategy into objectives, metrics, initiatives, and tasks customized to each group in the organization providing them with timely information and insights enabling stakeholders to improve decisions optimize processes and plans (Kerzner, 2017).

In this thesis, we discuss the development of a performance monitoring dashboard that provides decision makers with a better understanding of the organization's performance. The developed dashboard will provide high quality information using enhanced visualization and predictive tools. Compared to other monitoring dashboards that provide a single view of each variable, the developed dashboard provides different visualization options per Variable at different levels, so that decision makers will have a rich view of Variables. In addition, performance forecasting and

anomalies detection predictive tools provide insights to decision makers not previously available to them through the traditional dashboards. The final data will be used to train Linear Regression models to help in building predictive models capable of forecasting future trends (Destiandi & Hermawan, 2018).

1.2 The Site

Halan is a ride-hailing on-demand-logistics startup that was founded in the year 2017 in Egypt. The company offers smart transportation for the masses by offering on demand 2 and 3-wheeler vehicle transportation services and smart-tech last mile delivery through integrated smart services using motorcycles and electric tricycles (TUKTUK). Although the company was originally founded in Egypt for this analysis, we are going to focus on its Sudan office that was opened in 2018 located at Khartoum, Eltaif. The company currently averages 3000 rides per day making that 90,000 per month. The company's business model relies on connecting drivers and customers in need of service through a mobile application using GPS technology and later using the dashboard's analytical tools to determine crowded areas and shift the driver's attention to it by incentivizing any drive who is active in the determined area. This also reduces the fuel consumed looking for a customer. On the customers end the incentive to use the Application is by providing free or discounted rides to imprint the habit of depending on the application instead of hailing rides from the street. It makes things more convenient, cheaper, and safer. The convenience factor allows customers to order a ride from wherever they are without having to walk to the street. This is especially useful for older people and women. The fact that the application uses Route Network Analysis to calculate the shortest route and from that calculating the fare to regulate prices and prevents cheating and extortion from some drivers. It also establishes a link between the drivers, the company, and the customers in the off chance of them forgetting something on the vehicle. The fact that this link is available provides a sense of security for the customers. In its efforts to expand its business model the company started a new service called Small to Medium Enterprises (SME). Which is not integrated with the company's data base and dashboard. Which in turn leads us into the reason we are choosing to work on this project.

1.3 State of the Current System

The first step towards creating such a business is by recruiting stores to the cause and this role is currently being done by the Sales Department at Halan's main office. The stores contact information is then shared with the SME department's operations manager who then contacts the stores and explains the used protocols, gets their orders information, and shares it with the drivers' supervisor. The drivers' supervisor is then responsible for allocating drivers to pick up the different stores order's and bringing them back to Halan for sorting. The data analyst is then tasked with creating the spreadsheet used to store the new month's data and sharing it with the data entry officer. The drivers' supervisor is then tasked with sorting the orders, distributing them to different zones, assigning those zones to the available drivers and finally take pictures of each drivers orders and sending it to the data entry officer for data entry purposes. The data entry officer then enters

the data into the spreadsheet shared earlier by the data analyst. The drivers take the orders, delivers them, receive payment, and then return with the money and delivery report to the accountant. The accountant then receives the money from the driver, uses the delivery report to complete the spreadsheet and then the sheet is used to calculate the SME's money, Halan's margin of revenue and the drivers' cut which is then distributed amongst them by the accountant. Finally, the data analyst uses all the data in the spreadsheet to manually create reports.

1.4 Statement of the Problem

Organizations that make smarter decisions are more successful in the long run and for them to make the right decision at the right time is a critical point for the success and survival of an organization in a competitive environment the existence of large amount of data was always a headache for decision makers who aren't comfortable working with raw-data tables, leading to confusion and making it harder and time consuming to see insights and problems. This raw data must be transformed into useful information by the process of analysis and displayed in an interactive and visually pleasing manner. The company's current spreadsheet system faces major challenges with regards to human error and altering which affects its dependability, security, and linking it to the accounting system.

In specific, the problems faced by the system stakeholders are:

- Decision makers lack dynamic visual representation of the available data, which impedes their ability to make informed decisions as the current form of the data stored does not provide any useful indication of the organization's progress. The organization is limited at best to weekly outputs.
- The lack of an automated intelligent tool aiding decision makers to make well informed decisions due to the current nature of the used system.
- There is a high level of errors and time consumption associated with data entry and lack of a centralized data storage solution.

1.5 Objectives of the study

1.5.1 General Objective

The general objective of this project is to develop an Intelligent Dashboard for Halan Company using Data Science and Data Mining Algorithms that will store, organize, and visualize data using data analytics tools and predictive algorithms to aid decision makers in making well informed decisions.

1.5.2 Specific Objectives

- To assess the organization's current performance using a multilevel performance monitoring dashboard that is able to obtain data from various sources and track the organization's KPIs to measure how the organization is achieving its goals.

- To determine the most suitable supervised machine learning algorithm that can provide the management with timely, quality insight to point decision makers in the right direction.
- To reduce errors created from data entry by creating a data entry portal in the dashboard for order details verify and validate each order before uploading it to the database.

1.6 Scope and Delimitation

This thesis combines elements from Business, Data Science, Machine Learning. Statistics, and web application development to produce a solution to the problem's faced by the organization. The impact of each is explained below.

- **Business:** Monitoring business processes and their KPIs to make informed decisions.
- **Data Science:** Analysis of data extracted from organization's information system and SME transactions. Data analytics tools are implemented to understand the organization's activities and the effect the results have on their objectives. We shall attempt to provide availability, confidentiality & integrity in terms of security concerns.
- **Machine Learning:** The training of a Regression model to provide a better understanding of the current state and provide predictions.
- **Statistics:** Data verification, validation, analysis, forecasting and visualization.

The proposed dashboard will be able to query data from the organization's databases and visualize the data that the user requests, both in tabular and graphical form. It will use data analytics and supervised machine learning algorithms to recognize trends and forecast the performance (Ahmed, 2021) of the SME branch. The dashboard will be able to output daily, weekly, and monthly reports to track the performance of drivers, individual small businesses, and the SME branch as a whole.

The project focuses on data analytics tools using Python programming language and related libraries for the actual data cleaning, data mining, machine learning algorithms and data visualization. The researchers aim to provide availability of service, some data integrity, a level of security and confidentiality.

The concerns for the limitations of the project are dependent on the SME owners' ability to provide accurate and correct data. The data collection will not be done by the dashboard for it requires someone to update the data manually. As automated data validation and verification will be time consuming for the researchers, this task will be left to the organization's data analyst through the dashboard. The data will be input in batches from multiple sources as opposed to how it would be in a transaction processing system.

The proposed Machine Learning algorithm might not offer high precision and usefulness so a better alternative will be selected for the final version.

1.7. Conceptual Framework

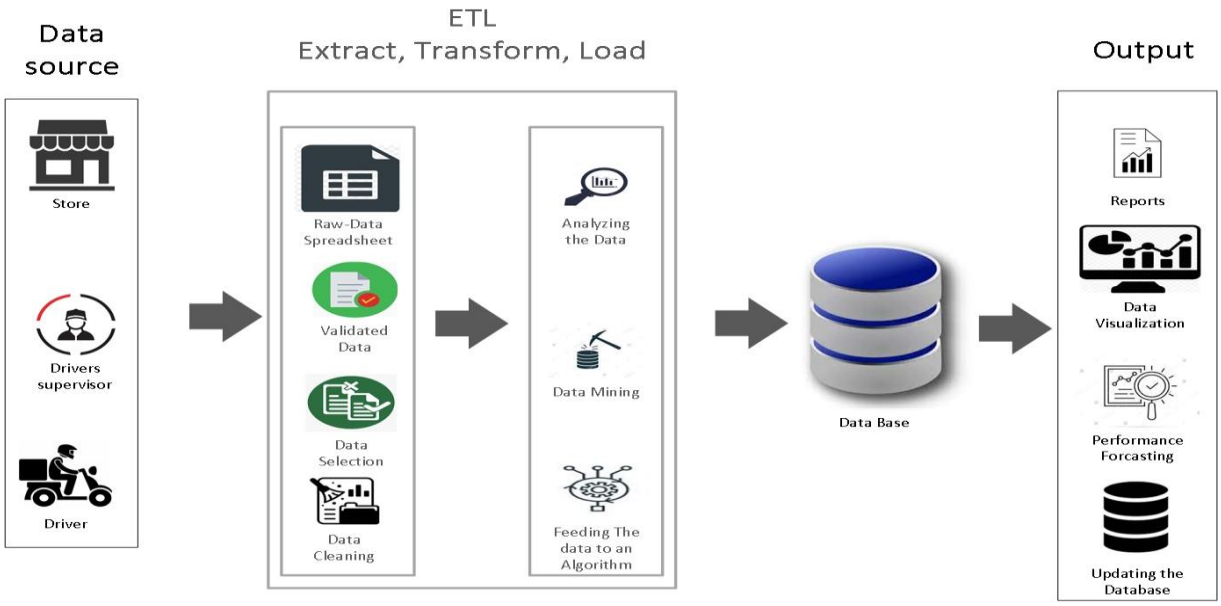


Figure1.1 Conceptual Framework of the System

1.8 Significance of the Project

This Project Benefits:

1. Halan's Stakeholders.
2. The Data Analyst.
3. The accountant.
4. The Delivery Drivers.
5. The Delivery Supervisor.
6. SME Owners.
7. The Researcher.
8. Future Researchers.
9. The Economy.
10. The Government.

Chapter 2

LITERATURE REVIEW

2.1. Background:

Insights derived from data allow organizations to stay competitive. Data science has the ability to leverage data processing, algorithms, and statistical analysis to develop insights from data. The big data phenomenon has made obtaining knowledge more challenging due to the changing characteristics of the totality of the enterprise's physical data assets. The increase in volume, variety, and velocity of big data challenges traditional information technology (IT) processes to scale and support big data analytics and data science (Demirkan, H. and Dal, 2014; Larson, 2019a). While the need for data science as a field is growing rapidly, the practices to ensure success of these projects have not kept up with the pace. One of the main challenges is using existing software development methodologies, such as the waterfall approach, has been proven to be problematic and identified as a contributing factor for data science project failure. This is due to the misconception organizations have that data science projects are like other IT projects (Larson, 2019a).

In this chapter, we are going to dive into more details regarding Business activity monitoring, Business Intelligence, KPIs, Dashboards, Python as tool for scientific computation including analysis and visualization of data sets, the Data science process, Machine Learning algorithms.

2.2. Business analysis:

Business analysis (BA) is the practice of Initiating change in an enterprise by Catering to the needs, recommending and solutions that is considered to be of value to the stakeholders. BA enables an enterprise to articulate needs and there for the rationale for change, and to style and describe solutions that may deliver value. (International Institute of Business Analysis, 2015).

The initiatives behind conducting such an in-depth analysis are extremely broad they may be strategic, tactical, or operational. BA can also be performed within the scope of a project or within an enterprise’s evolution and Gradual improvement. It is usually used to understand the current state, to define the future state, and to work out the activities required to maneuver from the current to the future state.

No matter their job title or organizational role. Business analysts are in charge of discovering, synthesizing, and analyzing information from a spread of sources within an enterprise, including tools, processes, documentation, and stakeholders. The business analyst is held accountable for eliciting the particular needs of stakeholders—which frequently involves investigating and clarifying their expressed desires—in order to work out underlying issues and causes (Milani, 2019).

Business analysts play an important role in aligning the designed and delivered solutions with the requirements of stakeholders. The activities that business analysts perform according to (Milani, 2019) include:

- Understanding enterprise problems and goals,
- Analyzing needs and solutions,
- devising strategies,
- driving change, and
- facilitating stakeholder collaboration.

2.3. Business Intelligence:

Business intelligence (BI) is human intelligence applied to affairs and activities by using artificial intelligence for collecting, integrating, analyzing and presenting the business information (Dinesh, 2021).

The reasoning behind electing to use business intelligence and more specifically Bi uses tools and techniques is to transform row data into information and knowledge giving us many benefits such as fast and accurate reporting, valuable business insights, competitive analysis, better data quality, increased customer satisfaction and increased operational efficiency. But the main function of BI is enabling decision makers to take effective facts based and data driven decisions.

For an individual to work in the field of BI as a BI analyst, BI consultant or a BI developer they need a set of skills such as communication skills, problem solving, advanced vision and attention to details, specific industry knowledge, critical thinking, coding data, classifying data, planning and decision making (Sarveswar, 2021).

How to apply BI concepts in our work? In order to get started we first need to gather and organize data using this data we can gain business insights and later present them in the form of metrics, KPIs, reports and dashboards.

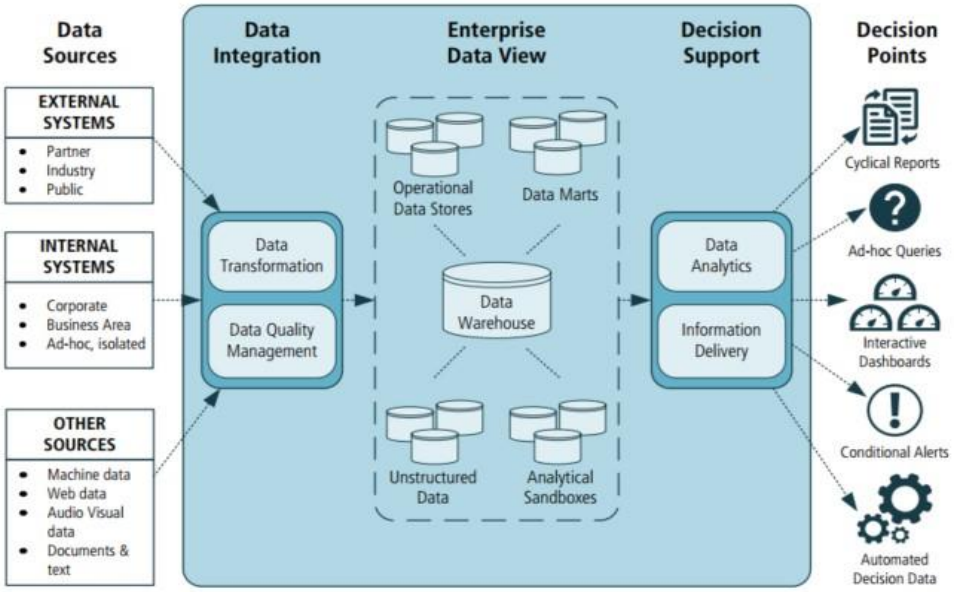


Figure 2.1 BI conceptual solutions (International Institute of Business Analysis, 2015)

Problems we might face using BI as illustrated in the figure above by (International Institute of Business Analysis, 2015):

- Existing business processes and transactional systems may be unable to provide source data that is definable and predictable,
- The cross-functional data infrastructure that is needed to support a business intelligence solution may not exist within the targeted organization.
- The organization might not recognize the importance of process re-engineering and change management in order to effectively realize the value from a business intelligence solution.

Common tools used in BI:

- Programming languages: Such as R, Python, SQL, and MATLAB.
- Software tools: Such as Microsoft Excel, Power BI, and Tableau.

2.4. Key Performance Indicators:

Key performance indicators (KPIs) are financial and non-financial indicators employed by corporates to estimate its success within the pursuit of a long-term goal previously set. We've got two levels of KPIs High-level KPIs which focus on the overall performance of the corporate while low-level KPIs that consider processes in specific departments(The KPI Institute, 2016).

KPIs are focused on forecasting and future results which is very useful to decision makers by providing accurate information on trends. The main purpose of KPIs is to trace performance measures that track changes in towards a target indicating the extent of progress made in achieving the project objectives. At the same time, key performance indicators identify the relevant elements

for performance and for providing information on the suitable controllable factors for creating decisions that will lead to a positive outcome. On the other hand, they do not indicate the measures to be taken to correct deviations from the target (NICA et al., 2021).

2.4.1. KPI components according to (Parmenter, 2020)**:**

- Targets: a set of goals associated with each individual KPI used to specify a measurable outcome rather than a conceptual destination. Most commonly targets are set during strategic meetings or budget discussion meetings by managers and workers alike to ensure but-in and an accurate selection of goals.
- Ranges: In order to help their workers, gauge their performance more accurately most organizations divide their targets into percentile-based increments above and below the target.
- Encoding and Thresholds: is to encode ranges into graphical displays on a dashboard or report.
- Time Frames: Performance targets have time frames which affect how KPI's are calculated and displayed.
- Benchmarks: KPI targets are to be tested against a benchmark that becomes the starting point for Driving up performance.

2.4.2.Reading KPIs:

Reading KPIs must be as straightforward as possible. The user should be able to whether or not the project is on track with one look at the visual display on the dashboard or report. To assist the user in the aforementioned task we are going to apply seven attributes associated with a good performance dashboard (Parmenter, 2020).

- KPI name: specifies what KPI is being displayed.
- Status: measures performance against the target.
- Trend: measures performance against the previous interval or a different time period.
- Actual value & Target value: usually shown within the graph itself.
- Variance: measures the gap between actual and target value.
- Variance percentage: gained by dividing the variance against the target

Figure 2.2 Sales Dashboard Comparing KPIs. Source: (Healy, 2019)

2.5. Dashboards:

A dashboard is not only a straightforward display of indicators, but most significantly it is a decision-making tool. Dashboards are presented as a solution that is ought to improve decision making by amplifying perception and capitalizing on human perceptual capacities. The two management experts Robert S. Kaplan and David Norton see the dashboard as a management tool. They concluded by observing companies in the U.S. In 1996, they created a strategy management system in the form of dashboards, named Balanced Scorecard, whose purpose is to make and steer the implementation of strategy.

In This next section introduces the properties of different dashboard types. But before attempting to design a dashboard, we should make some considerations:

Who am I targeting? By answering this question and figuring out the view of our targeted audience we will be able to easily create and design your dashboard. What value will it add? By finding the actual reason for creating the dashboard we will be able to figure out the desired output. What kind of dashboard am I creating? To answer this, we need to know about the different types of dashboards according to (Kerzner, 2017)

2.5.1. Dashboard Types:

There are three common types of dashboards - strategic, tactical, and operational dashboards they vary from Each other, but they also have a common trait which is that good dashboards focus on the most important information and delivers that information as clear as possible.

A) Operational Dashboards:

Operational dashboards monitor core operational processes and are used mainly by front-line workers and their supervisors who interact directly with customers or with creating and delivering

the organization's product or service. Operational dashboards deliver detailed insights that are only lightly summarized. And there for, operational dashboards emphasize monitoring over analysis and management.

B) Strategic Dashboards:

Strategic dashboards monitor the execution of strategic objectives with the goal of getting every group within the organization marching in a unified direction. Strategic dashboards are usually used by managers and executives at every level of the company in order to digest the information they need about the company's overall status; it's also as a communication tool to convey strategies. In this type we don't need to view detailed specific info or provide interactive features it should be simple and undemanding. Strategic dashboards emphasize management over monitoring and analysis.

C) Tactical Dashboards:

Unlike strategic, tactical dashboards are prepared for more detailed purposes usually being used for tracing trends that are relevant to the company's goals and actions. There are a lot of metrics and variables used in this type of dashboards but usually we are not measuring the goal itself but to check how the planned value is different from its execution. Tactical dashboards can be more complex than strategic and operational dashboards and because the ultimate goal is to facilitate the user's interaction with the data it's important to be able to compress data across time and multiple variables. If we lose this ability, then the dashboard is rendered useless because it did not accomplish its primary goal.

2.5.2. Data Visualization:

In the words of Scott Klien, deputy editor at ProPublica, there is no such thing as an innately intuitive graphic. Because none of us are born literate in reading visualizations.

As we create data visualizations, we must get to know our audience and know when a different graph can create readers engagement and help them expand their own graphic literacy.

Over the past thirty years or so, the figure below shows a wild spectrum of data encodings techniques such as dots, lines and bars arranged by how easily the user can estimate their values. Encodings with a high probability of the reader correctly guessing their values are arranged at the top, and vice versa. (Cairo, 2016)

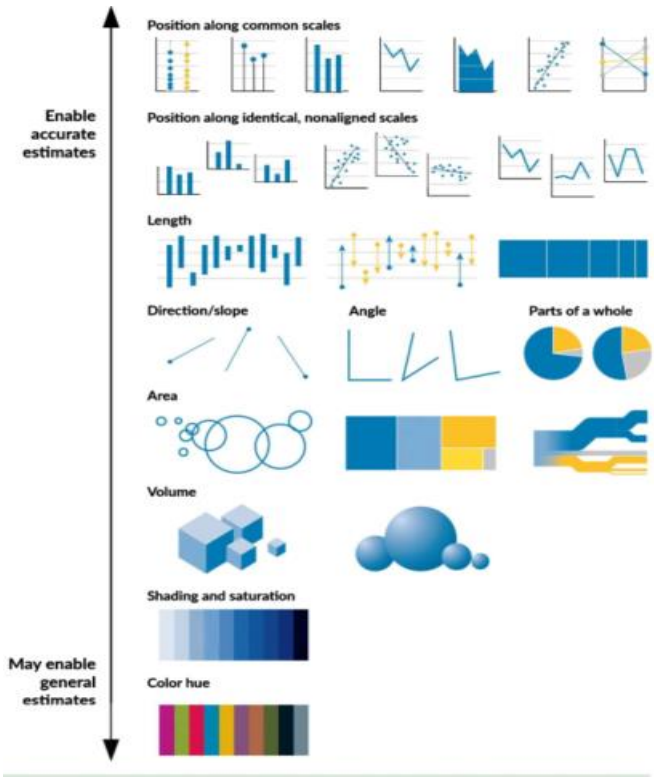


Figure 2.3 Data Encoding Techniques. Visualization of ease of determining data value (Schwabish, 2021)

The bedrock of any data visualization is the data. Without data and a good enough understanding of what the data is, how it was collected, and what it tells us, we are just going to be painting pictures (Schwabish, 2021).

In 1997, Ben Schneiderman, then a professor of computer science at the university of Maryland at College Park, wrote what later became the mantra of online interactive data visualization:

"Overview first, zoom and filter, then details-on demand".

The theory is that you ought to give the users an overview of the visualization, then let them zoom in or out and filter through it, allowing them the opportunity to reveal details as they see fit.

2.5.3. Why Do we Need to Visualize Data?

There is a well Known dataset Called Anscombe's quartet (Anscombe, 1973), shown in Table 1, it presents the argument for looking at data in visual form. Each of the four Manufactured "datasets" contains eleven observations of two variables, x and y. By construction, the numerical properties of every pair of x and y variables, like their means, average values, and variance are almost identical. Even more, the standard measures of the association between each x and y pair are also a match. The correlation can run from -1 to 1, with zero meaning there is no association. A score of -1 means a perfect negative association and a score of 1 a perfect positive association between the two variables. So, 0.81 counts as a strong positive correlation. coefficient is a strong 0.81 in every case. The data was then visualized using a series of four scatterplots. A scatterplot is used to shows the association between two quantities. Using the aforementioned visualization technique the differences between them are readily apparent as shown in figure 2.4 (Wexler et al., 2017).

Table 2.1 Illustrating Anscombe's four Groups

Group A		Group B		Group C		Group D	
x	y	x	y	x	y	x	y
10.00	8.04	10.00	9.14	10.00	7.46	8.00	6.58
8.00	6.95	8.00	8.14	8.00	6.77	8.00	5.76
13.00	7.58	13.00	8.74	13.00	12.74	8.00	7.71
9.00	8.81	9.00	8.77	9.00	7.11	8.00	8.84
11.00	8.33	11.00	9.26	11.00	7.81	8.00	8.47
14.00	9.96	14.00	8.10	14.00	8.84	8.00	7.04
6.00	7.24	6.00	6.13	6.00	6.08	8.00	5.25
4.00	4.26	4.00	3.10	4.00	5.39	8.00	12.50
12.00	10.84	12.00	9.13	12.00	8.15	19.00	5.56
7.00	4.82	7.00	7.26	7.00	6.42	8.00	7.91
5.00	5.68	5.00	4.74	5.00	5.73	8.00	6.89

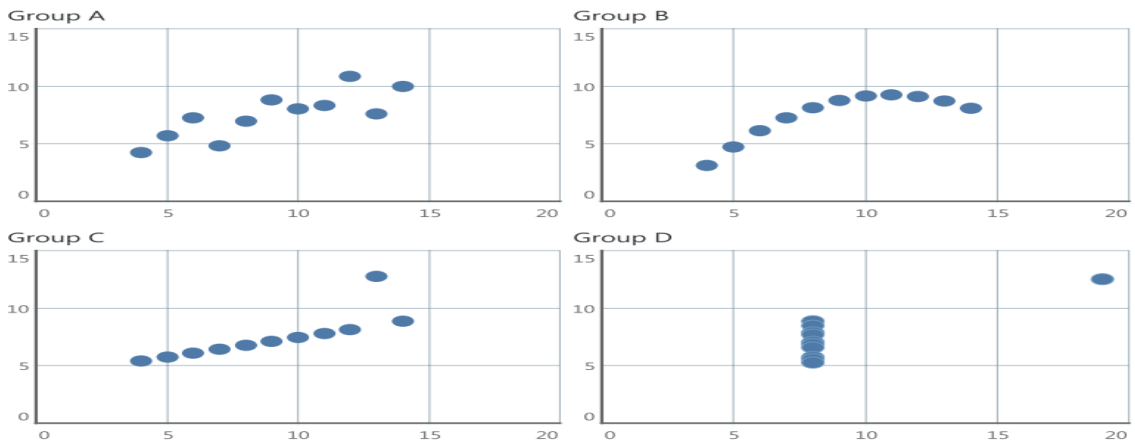


Figure 2.4: Plots of Anscombe's quartet

Anscombe's quartet is an extreme, manufactured example. But the benefits of data visualization to one's data can be shown in real life cases. In the figure 5 we are shown a graph from(Jackman, 1980), a short comment on (Hewitt, 1977). The original paper was arguing the existence of a strong association between voter turnout and income inequality based on a quantitative analysis research conducted on eighteen countries. But when this relationship was graphed as scatterplot, it was clear that the association was completely dependent on the inclusion of south Africa in the sample.

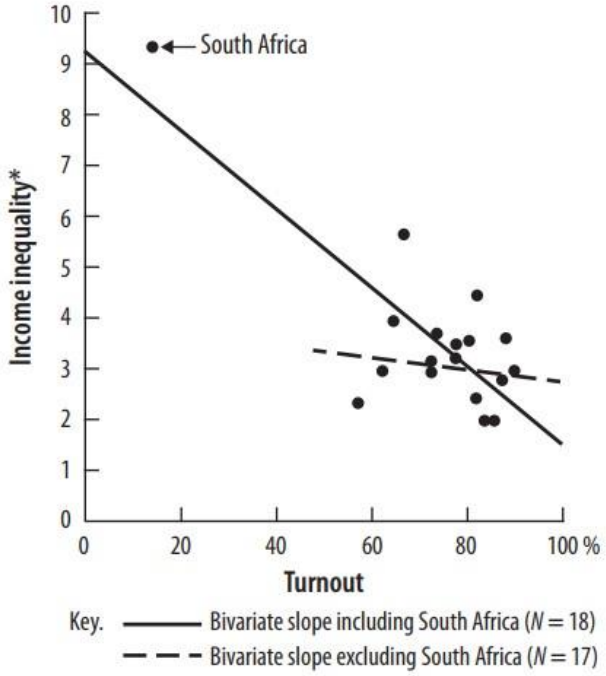


Figure 2.5 The effect of an outlier on a regression line (Jackman, 1980)

2.6. Data Mining (DM)

Data Mining, also known as knowledge discovery in data (KDD), is the process of automatically searching and analyzing data and discovering previous unrevealed patterns. It involves (Haider, 2016):

- Establishing DM goals
- Selecting data to be mined,
- Preprocessing data,
- Transforming data into appropriate format,
- Mining and extracting insights and patterns using various tools and techniques such as Machine Learning, statistical models, and data visualization tools,
- Evaluating Mining results.

DM has improved organizational decision-making through insightful data analyses (IBM, 2021a),. The DM techniques that underpin these analyses can be divided into two main purposes; either describe the target dataset or predict outcomes through the use of machine learning algorithms. These methods are used to organize and filter data, surfacing the most interesting information, from fraud detection to user behaviors, bottlenecks, and even security breaches.

2.7. Machine Learning (ML)

Machine Learning is a subset of Artificial Intelligence (AI) that uses computer algorithms to analyze data and make intelligent decisions based on what it is taught without being explicitly programmed. The ML algorithms, or models, are trained with large sets of data and learn from examples(Hurwitz & Kirsch, 2018), as opposed to the traditional rule-based algorithms, thus enabling machines to solve problems on their own and make accurate predictions using provided data. Essentially it involves building a model that forms generalized rules, or equations, from a set of examples, then use that model to make predictions. This is called *model-based learning*. The antithesis of model-based learning is *instance-based learning.* That is through memorizing the set of examples generalizes to new cases or inputs using a measure of similarity (Géron, 2019).

The number of data science and big data projects is growing, and current software development approaches are challenged to support and contribute to the success and frequency of these projects (Larson, 2019a). The ultimate purpose of data analytics is to communicate findings to the concerned stakeholders who might use these insights to formulate a policy or strategy. The results are summarized in tables and plots, which can then be used to build the narrative. Before the data scientist starts their analysis, the final deliverable's scope must be determined. This is to ensure the deliberation of the final report's key message (Haider, 2016). Then the necessary data and analytics are obtained to make their case. This shows the importance of the initial planning and conceptualizing of the final deliverable to lead to the production of a compelling document.

The proliferation of open-source technologies available to data scientists can also complicate the landscape. With the increase in data science and big data projects, organizations are struggling to deliver successfully. (Larson, 2019a)

2.7.1.Types of Machine Learning Techniques

Machine Learning algorithms are generally divided into four categories. The following is a brief discussion of each type with their applicability to solve real-world problems, as discussed in (Sarker, 2021).

- **Supervised:** Supervised learning is the machine learning task of learning a function through mapping an input to an output based on sample input-output pairs. It achieves this by using labeled training data and a collection of training examples to infer a function. Supervised learning is carried out when certain goals are identified to be accomplished from a certain set of inputs and is such know as a task driven approach. Two typical supervised learning tasks are classification and regression. A real-world application for supervised learning is classification of spam emails.

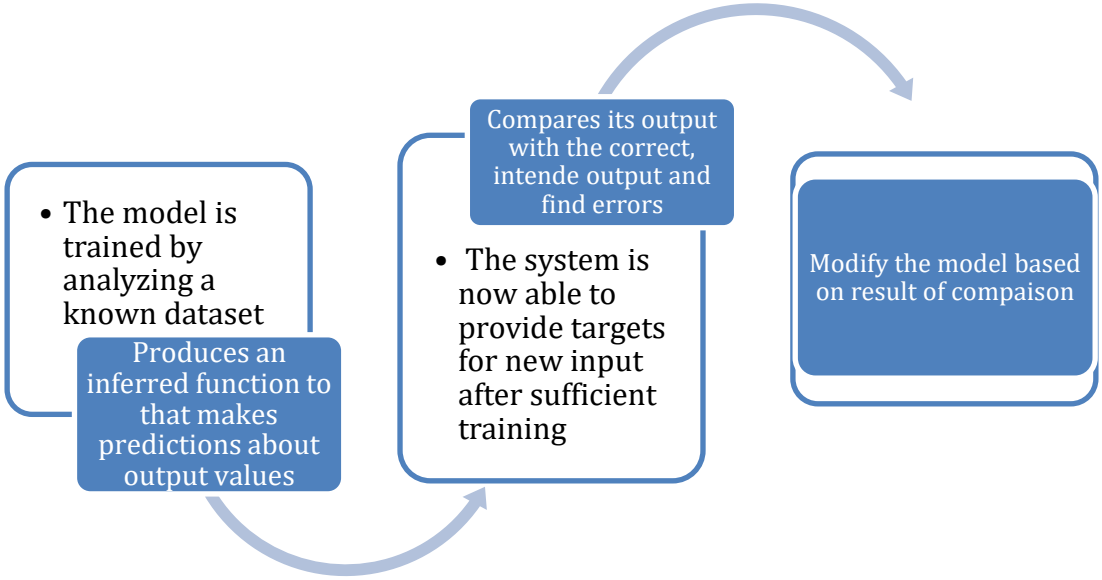


Figure 2.6 Supervised Machine Learning. Own diagram with reference to (M.B, 2019)

- **Unsupervised:** Unsupervised learning analyzes unlabeled datasets without the need for human interference, making it a data-driven process. This is generally used for extracting generative features, identifying meaningful trends and structures, groupings in results, and exploratory purposes. The most common unsupervised learning tasks are clustering, density estimation, feature learning, dimensionality reduction, finding association rules, anomaly detection, and so on.
- **Semi-supervised:** Semi-supervised learning is a hybridization of supervised and unsupervised methods, where it can operate on both labeled and unlabeled data. Labeled data can be scarce in the real world and unlabeled data plenty, making semi-supervised learning very useful in providing a better outcome for prediction than that produced using

the labeled data exclusively. Some areas of application where semi-supervised learning can be useful include machine translation, fraud detection, labeling data, and text classification.

- **Reinforcement:** Reinforcement learning algorithm that enables software agents and machines to automatically evaluate the optimal behavior in a particular context or environment to improve its efficiency, making it an environment-driven approach. This type of learning is based on reward or penalty, and its ultimate goal is to use insights obtained from environmental activists to take action to increase the reward or minimize the risk. It is a powerful tool for training AI models that can help increase automation or optimize the operational efficiency of sophisticated systems such as robotics, autonomous driving tasks, manufacturing, and supply chain logistics, however, not preferable to use it for solving the basic or straightforward problems.

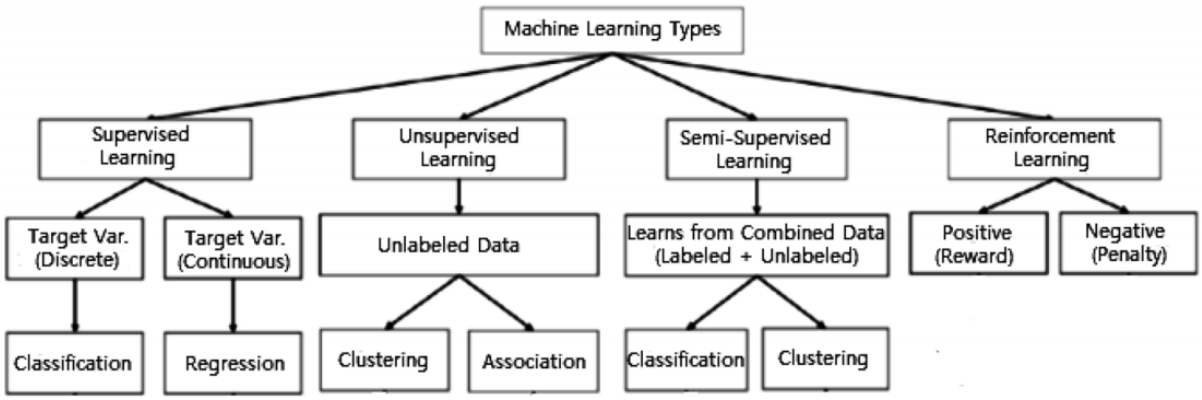


Figure 2.7 Machine learning types (Sarker, 2021)

2.7.2. Regression

Regression is a basic and ubiquitous type of predictive analysis. It attempts to model the relationship between variables by fitting an equation to the observed data. One variable is considered to be the independent variable, *x*, and the other is considered to be a dependent variable, *y*. For example, a modeler might want to define the relationship between the number of rooms in a house and the price of the house using a linear regression model. This is called *regression analysis.* As defined in (Sarker, 2021), when conducting regression analysis, one seeks to express one dependent variable as a linear combination of other features or measurements.

Lasso Regression, also known as Least Absolute Shrinkage and Selection Operator Regression, is a regularized version of Linear Regression. It adds a regularization term $\sum_{i=1}^{n} \theta_i^2$ to the cost function, which forces the learning algorithm to not only fit the data but also keep the model weights as small as possible (Géron, 2019).

$$J(\boldsymbol{\theta}) = \mathrm{MSE}(\boldsymbol{\theta}) + \alpha \sum_{i=1}^{n} \left|\theta_i\right|$$

Figure 2.8 Lasso Regression cost function

2.7.3. Training a Simple Linear Model

A linear model can be trained through model-based learning. An example of a simple linear model:

$$Y = m\,X + C$$

This model has two model parameters, m, the function's gradient, and c, the Y-intercept (Géron, 2019). The gradient defines how Y changes with respect to X. the Y-intercept defines the value of Y when X=0, i.e., where the line meets the dependent variable's axis. By modifying these parameters, we can make our model represent any linear function. Before the model is ready for use, the parameter values m and c need to be defined. How can we know which values will make our model perform best? For that end, we need to specify a measure of performance. Either though defining a utility function (or fitness function) that measures how well the model does, or by defining a cost function that measures how poorly it does (Sarker, 2021).

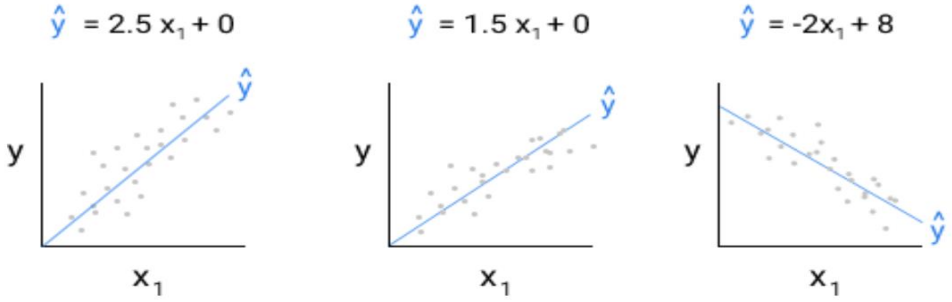


Figure 2.9 An Example of different simple linear regression models (M.B, 2019)

For linear regression problems, the trend is to use a cost function that measures the distance between the linear model's predictions and the training examples, where the aim is to reduce this distance as much as possible. This is where the Linear Regression algorithm comes in: by giving it our training data, it finds the parameters that make the linear model fit best to our dataset. This is called training the model (Géron, 2019).

2.8. Python for Machine Learning

Python has emerged over the last couple decades as a first-class tool for scientific computing tasks, including the analysis and visualization of large datasets. This may have come as a surprise to early proponents of the Python language: the language itself was not specifically designed with data analysis or scientific computing in mind(Müller & Guido, 2017).

The usefulness of Python for data science stems primarily from the large and active ecosystem of third-party packages: NumPy for manipulation of homogeneous array-based data, Pandas for manipulation of heterogeneous and labeled data, SciPy for common scientific computing tasks, Matplotlib for publication-quality visualizations, Jupyter Notebooks for interactive execution and sharing of code, Scikit-Learn for machine learning, and Dash API for designing interactive web-based dashboards(Castillo, 2021).

2.9. Data Science (DS)

The widespread use of the term "Data Science" to refers to the unique amalgamation of principles and methods from a diverse set of disciplines such as statistical analytics, engineering, entrepreneurship, and communication, aiming to generate valuable knowledge from the data itself, has only been recognized around 2012 (Braschler et al., 2019). Data Science is a process, not an event. The of gaining knowledge and insights from large volumes of disparate data, through modelling/hypothesizing and validating it through implementation of appropriate algorithms to analyze and provide value and knowledge from the data, which can exist in various forms, whether it is structure or unstructured (Haider, 2016; IBM, 2021a). It is one of the many disciplines that tends to derive knowledge through analytics. Most components of data science, such as statistics, probability, algebra, programming, database systems, and machine learning to mention a few, have been around for decades, but with the current computational capabilities, they can be combined into new techniques and powerful algorithms (Haider, 2016).

Organizations use insights derived from data to stay competitive. The big data phenomenon has made deriving insights more challenging due to the changing characteristics of the data landscape. The increased volume, variety, and velocity of big data challenge traditional information technology (IT) processes to scale and support big data analytics and data science. Big data is used by organizations as a resource in data science projects to develop new business value and insights. Big data examples include sensor data, images, text, audio, and video data. These data sources can provide new insight opportunities alone and when paired with existing data sources such as organizational data warehouses.

Data science is a competency that leverages data processing, algorithms, and math to develop insights from data, allowing organizations to stay competitive. While data science as a competency is growing, the practices to ensure these projects are successful have not kept up with the pace. One primary challenge is using existing software development methodologies to deliver data science projects. Applying traditional software methodologies, such as the waterfall approach, is

problematic and has been identified as the one contributing factor for data science project failure; organizations are treating data science projects like other IT projects (Larson, 2019).

The following figure depicts the interconnectivity of the subsets of advanced analytics, which includes Business Intelligence and Analytics, Data Science and Analytics, and Machine Learning.

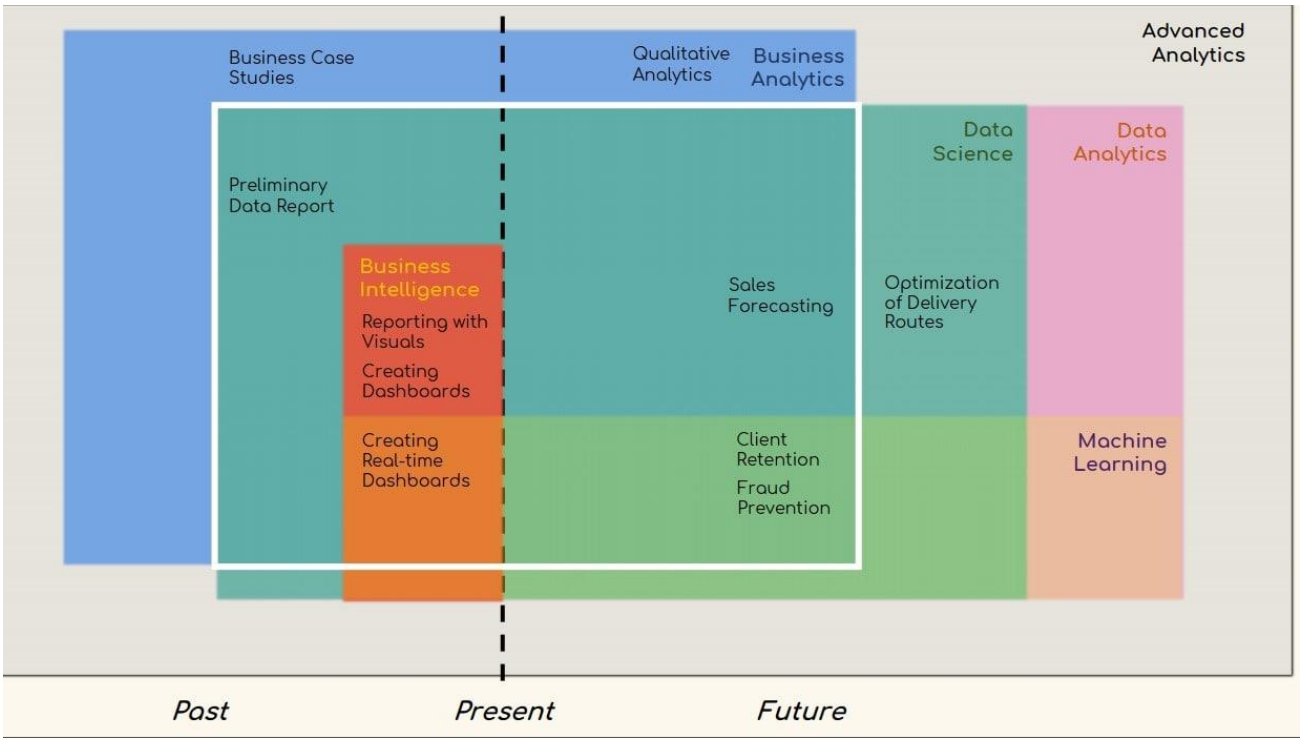


Figure 2.10 Advanced Analytics (365 Data Science, 2021)

2.10. The Data Science Process

As mentioned above, DS is a process, not an event. A data scientist cannot be expected to provide actionable insights instantly and consistently. DS leverages data processing, algorithms, and math to develop insights from data, which will offer competitive advantages to organizations. One of the main challenges is using existing software development methodologies to deliver DS projects, which has been identified as the one contributing factor for DS project failure: organizations treating DS like other IT projects (Demirkan, H. and Dal, 2014). It is neither a product nor a computer system, but rather a constantly evolving strategy, vision and architecture that continuously seek to align an organization's operations and direction with its strategic business goals.

Next we explore the phases data science project cycle below, as outlined by (Godsey, 2017).

2.11. The lifecycle of a data science project

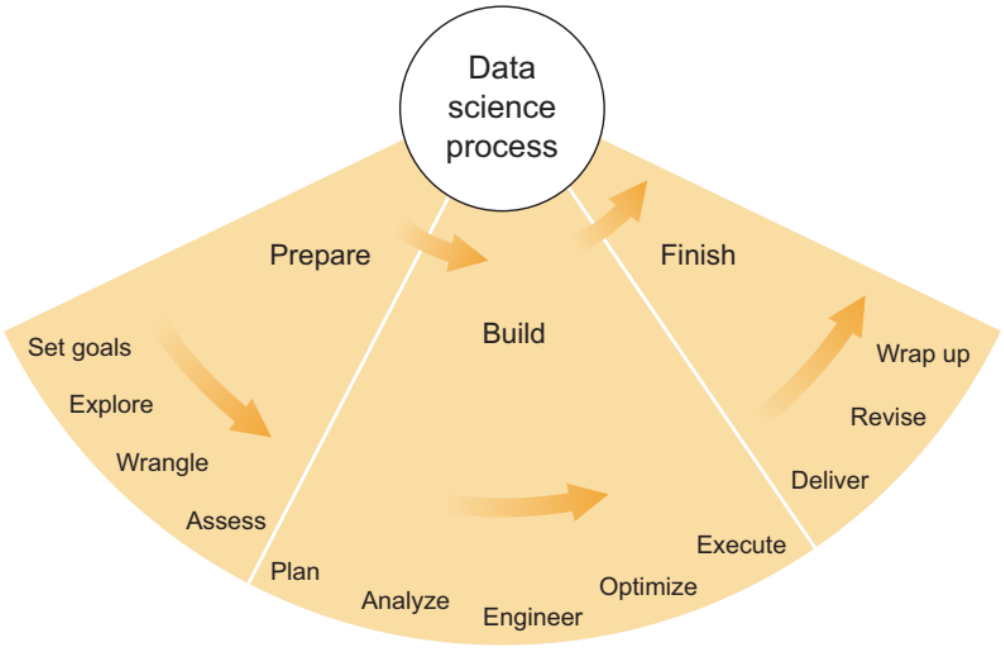


Figure 2.11 The Data Science Process as depicted by (Godsey, 2017)

2.11.1. Preparing for the project

Every DS project has at least one stakeholder who have some expectations about what they expect to receive from the data scientist who has been tasked with the project(Godsey, 2017). These expectations often include:

- Finding answers to questions or solutions to problems,
- Producing a final product, such as a report or software system,
- Summaries of past research or related projects and products.

The data scientist should aim to resolve their client's wishes with a pragmatic approach and assess each requirement with its feasibility. A notable difference between many fields and DS is that even a seasoned data scientist may not know whether something the customer wishes is possible. It usually requires that they familiarize themselves with the relevant data to ascertain what specific data is available and how much evidence can be derived from it. The increased use of internet-connected devices has changed the way organizations use data. The key characteristics of data that highlight the challenges with traditional software development approaches are Volume, Variety, Velocity, and Veracity(Larson, 2019a).

I. Set goals

To adequately prepare for a DS project, some considerations must be made in advance. The data scientist must first **set the goals** of the project, explore the data, conduct some appropriate data wrangling, and assess the data (Godsey, 2017). Starting with the main goals, either given by the client or through exploration of the data, to understand what can be derived from it and assess the quality of the given data. When interviewing the client, one should ask specific questions to uncover fact, not opinions. Their wishes should be resolved with a pragmatic approach; what can be done and how long it will take, while keeping in mind the budget for the project. Although the data scientist may be the expert in statistics and software, the client is, more often than not, the expert in their subject matter, and such, may offer insights that the data scientist would not be aware of (Braschler et al., 2019; Godsey, 2017). After setting the goals, we make use of statistics and programming to move towards achieving those goals.

II. Explore

After the initial goal setting stage, one must verify the usefulness of the available data through **exploration**. As data science is the extraction of knowledge from data (Braschler et al., 2019), the quality of data gathered is reflective of the quality of the expected output. Data source identification and collection can aid the data scientist in validating and understanding the data. Questions such as "what kind of format is the data stored in?", "How many fields of data do we have? Is it enough for our purposes?", and "How much data is expected to be generated or collected?"(Godsey, 2017) are all relevant to understanding the data. After that, the data can be profiled and analyzed using statistical tools to determine quality, demographics, relationships between variables, and distributions. This leads to an understanding of how the data can be used(Larson, 2019a). Asking questions that the data can actually answer and solve the original problem is a key element of this stage. This stage is often referred to as exploratory data analysis (EDA).

III. Wrangle

The next stage is **data wrangling**, which is the "process of taking data and information in difficult, unstructured, or otherwise arbitrary formats and converting it into something that conventional software can use." (Godsey, 2017) The goal of this stage is to prepare the data to be used in the modeling stage. The understanding derived from the previous stage is used to determine the final set of attributes, or features, that will be used in the model. This stage is also known as the data preparation stage. It includes integration, cleansing, and deriving of new attributes(Larson, 2019a). It is an iterative process that may be revisited later on in the cycle, as different models that are to be trained required the data to be formatted differently. In some cases, the all the data needed may be provided by the client and other times a quick web search could lead you to finding the data needed. Either way, the data is often messy and requires some cleaning. A way to think about it is to consider how an algorithm will see this data and how it can interact with it. This is useful in creating a good data-wrangling script. Another approach is to make use of the many software tools

built for this purpose. (Godsey, 2017)Once the data is ready for analysis, the data assessment phase may begin.

IV. Assess

The **data assessment** stage involves learning the contents, scope, and limitations of the data, among other features. The more we know about our data the more informed decisions we will make. This can be done through some descriptive statistics techniques (Godsey, 2017). For example, we might want to know what the maximum, minimum, and average values of each attribute are, obtain a list of all possible values a field may have, a summary of the data sets and much more. Descriptive statistics is the "discipline of quantitatively describing the main features of a collection information, or the quantitative description itself."(Downey, 2015) Some examples of statistical methods are classification, clustering, inference, and modeling.

The initial phase is very critical as it offers some guidance on how to develop and state goals in a constructive way. If not implemented correctly, can lead to the failure of the project (Demirkan, H. and Dal, 2014).

2.11.2. Building the product

When building the product, the phases the project team will undergo are: Plan, Analyze, Engineer, Optimize, and Execute phase. (Godsey, 2017)The main objective of any DS project is to produce something that helps solve problems and achieve goals. This might take the form of a software product, report, or set of insights or answers to important questions (Godsey, 2017; Larson, 2019a). The key tool sets for producing any of these are software and statistics.

I. Plan

To decide on what tools and methods to implement, some **formal planning** must be done. It is worth to note that as the project progresses, progress assessment and planning can be done continually, to ensure the goals identified in the initial preparation phase are being met. That way, the execution phase, which will be at the end of the building stage, will go more smoothly. The questions one can ask themselves is:

- What is possible to do?
- What is valuable to know?
- What is the efficient way of doing it?

Some might think that setting goals and planning are the same. Planning involves a lot of details that stem from considering the goals and deciding on how time, resources, people, schedules, and financial cost can be used to achieve the goals(Neifer et al., 2021). As there are always uncertainties to anything in life, it is important to have flexible paths for the future and to adjust expectations and goals based on preliminary findings. These new findings should be communicated to the customer as well as the project's progress.

II. Analyze

The **statistical data analysis** phase comes after planning. Statistical methods are considered at least a third of the skills and knowledge needed for a good data science project. The Oxford Dictionary of Statistical Terms (Dodge, 2006) describes statistics as "the study of the collection, analysis, interpretation, presentation, and organization of data." This phase focuses on the analysis and interpretation of the data. Thinking about a project and a problem theoretically before starting the software building or full-analysis phase is worthwhile as there is much to be learned.

Statistics lies between applied mathematics and the reality of observable data (Godsey, 2017). It branches out further into descriptive and inferential statistics. The former being the more intuitive kind of statistics that can provide a good overview of the data whereas the latter is the process of estimating unknown quantities based on measurable, related quantities (Downey, 2015; Mathur & Kaushik, 2016). Statistical modeling is the general practice of describing a system using statistical constructs and then using that model to aid in analysis and interpretation of data related to the system (Godsey, 2017). One of the well-known pitfalls in statistical modeling is that the model may work well when applied to specific data within a specific context but fails in other sets of data. That is usually due to the black box phenomenon.

In machine learning (ML), a computer program learns from experience E with respect to some task T on some data and some performance measure P, and as it gains more experience, its performance grows (Mitchell, 1997). ML uses a combination of statistics, mathematics, and programming to produce models that are capable of solving real world problems.

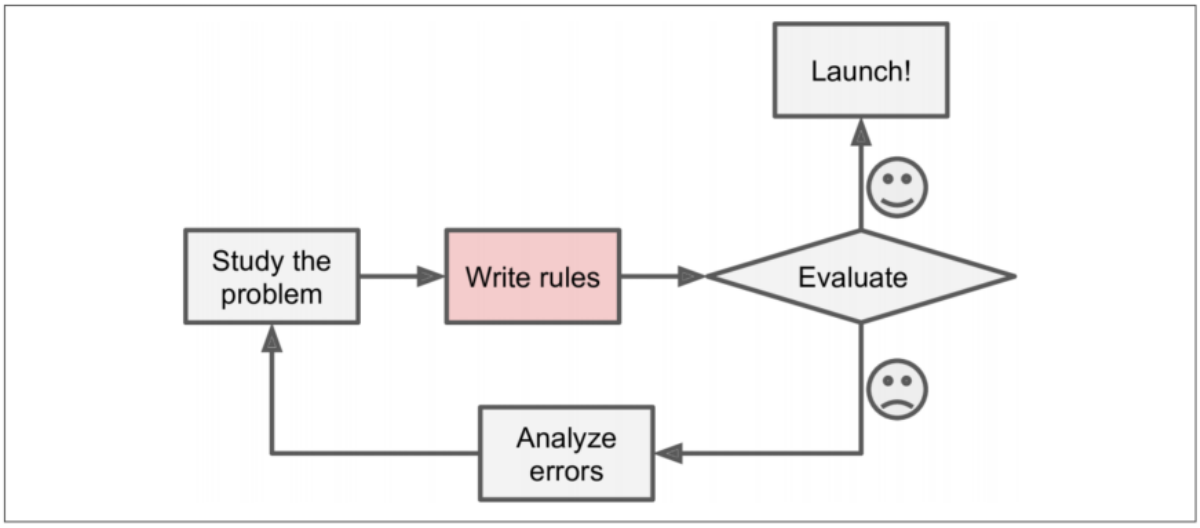


Figure 2.12 The traditional approach(Géron, 2019)

III. Engineer

Based on the chosen method, the existing skills/personnel, and resources, appropriate software must be selected(Neifer et al., 2021). A good start is to go with spreadsheet technology due to its representation of data in a row-and-column tabular form, making less complex analysis trivial. Spreadsheet applications, such as Microsoft Excel, are considered low-level statistical applications(Godsey, 2017). Some mid-level statistical applications, that offer more sophistication and don't require a background in programming are SPSS, Stata, SAS. All three are similar with similar experience in doing statistical analysis with only small differences so it is ultimately a matter of preference which to use between them(Godsey, 2017). =They require an intermediate level understanding of statistics to be able to make full use of their functions. For those on the lower end of the spectrum that seek out-of-the-box tools that are capable of conducting statistical analysis on data, there are many good statistical software that are constantly being developed and some are built for specific purposes(Downey, 2015; Géron, 2019; Godsey, 2017).

When out-of-the-box software tools aren't enough for the given purpose, it is recommended by (Godsey, 2017) to make your own. The most popular language for developing GUI-based statistical software is Python. The following table lists the features of Python (IBM, 2021a):

Table 2.2 Python (DataCamp Team, 2020; Godsey, 2017; IBM Cloud Team, 2021)

PYTHON
Open Source.
IPython is a GUI-based statistical application built for Python.
Great community constantly updating and optimizing the language.
General purpose programming language with a general approach to Data Science. Provides a general approach to data wrangling and exploration. Able to filter, sort and display data in a matter of seconds.
Used by programmers, developers, and data scientists.
Preferred when developing data products as it is a production-ready language, with the capacity to integrate with every part of the workflow.
Has a robust ecosystem with simple syntax and is easy to interpret.
Great for building data science pipelines and machine learning products integrated with web frameworks at scale. This has some dependencies and requires installing some Python libraries, such as NumPy and Pandas, which are not pre-installed. The Python Package Index (PyPI), otherwise known as pip, makes package installation much easier.
Flexible for creating new web-scripts and applications.
Has a smooth curve due to its readability and simplicity. Suitable for beginner programmers.
Visualizations are more complex, and results are not as eye pleasing or informative.
Supports all kinds of data formats, from comma-separated value (CSV) files to JSON sourced from the web. SQL tables can be imported directly into the Python code. Data can be grabbed from the web to build datasets
Jupyter Notebook, an open-source web application used to create and share documents, or notebooks, that can contain live code, equations, visualizations, and narrative text and supports over 40 programming languages including Python and R (Jupyter, 2021), can be used as the development environment.

Programming languages are far more versatile than mid-level statistical applications. Code written in programming languages has the potential to do almost anything. They can interact with other software services through APIs and can be included in scripts and other pieces of software (Godsey, 2017; Larson, 2019). These scripts can be designed to load whatever data is needed into the interactive environment, make data transformations or calculations, and generate graphs and reports. Not only are the easy to modify for specific situations, but they are also easily reproducible. Scripting also has its limits and disadvantages. They might get too long and complex that even the author would find it difficult to understand. Sometimes copying one reproducible section of code to another section, while forgetting to make any necessary changes could lead to unexpected errors in the result.

Excel users commonly implement functions on their datasets. These functions can be easily replicated in scripting languages by using built-in commands such as *sum* and *sort* as well as conditional logical constructs involving *if* and *else*. Iteration is a basic command found in many programming languages that spreadsheet don't handle it quite as well.

After deciding which too will be used, the statistics is to be translated into software, either by using built-in methods or coding the methods from scratch. The latter is more time consuming so it should be avoided unless the programmer knows what they are doing and there aren't already methods for implementation.

IV. Optimize

The software tools mentioned can be very versatile, and the statistical nature of each has been discussed, but software can do much more than statistics. There are many supplementary tools available that are designed to store, manage, and move data efficiently. This leads to optimization of almost every aspect of calculation, analysis, and management.

When designing a system that has a frequent flow of data that will be accessed by others often, it is recommended to use a database as the data source (Godsey, 2017). There is a number of different types of databases that exist, each developed to store data and provide access the data in different ways. They are designed to efficiently deal with file-based storage. The two most common databases are relational and document oriented.

Relational databases involve tables that are visualized as two-dimensional sheets such as those in spreadsheets, containing rows and columns, with data elements in the cells (Mccaffrey, 2020). They are capable of holding multiple tables and linking them with relation to each other. That is usually possible when each respective table contains a common column found in both tables. This optimizes the querying of data from multiple tables and data types, effectively saving enormous amounts of time (IBM, 2021a).

The dominating language in formulating queries for relational databases is structured query language, known as SQL and sometimes pronounced as "sequel". SQL is practically found everywhere but not all SQL-based databases use exactly the same syntax. To mention a few of the

well know SQL-based databases, there are: Oracle, MySQL, Google cloud SQL, PostgreSQL, Microsoft Azure, and MariaDB (IBM, 2021a; Mccaffrey, 2020).

Mostly, databases can provide random access to stored data via queries much quicker than file systems can, and they are scalable to large sizes, with redundancy, in convenient ways that can be superior to file system scaling.

The benefits of using databases according to (Godsey, 2017) are:

- Indexing: A map of all the data is generated to allow them to be located easily and quickly. Indexes take up their own space, so it is a trade-off between space and time.
- Caching: Data that is frequently accessed may be held close to increase efficiency of accessing that set of data. As in indexing, caching requires some space to be reserved for the data.
- Scaling: Databases can be distributed over many machines. Although storing on a single disk may be more advantageous in terms of access speed, distributed databases, which consist of shards or chunks of data, are designed to automatically keep track of where things are located. A central server manages access and transfer between shards. Additional shards can be used to increase the potential size of the database or even to replicate data that exists elsewhere, according to the way the database is configured.
- Concurrency: Databases generally handle concurrent processing better than file-based systems. That is when more than one computer process tries to change the same data point at the same time. Generally speaking, errors occurring from trying to create or edit the same file at the same time should be avoided at all cost and databases tend to provide convenient solutions.
- Aggregations: Databases provide functionality for performing aggregations of data matching a query or all data. A database might be able to add up, multiply, or summarize data much faster than a code would. That means that it might be better to allow this to be done on the database side to increase overall efficiency.
- Abstracted query language: Querying a database for certain data involves formulating the query in a query language, such as SQL, that the database understands. These languages offer abstraction from the search algorithm that underlies the query. This means we don't have to worry about the specific search algorithm as the database handles it.

Databases offer ways for interfacing with other software tools, such as Excel, where data can be exported to the databases. The most popular programming languages, including Python, all have libraries or packages for accessing all the most popular databases, and can easily be learnt due to them being well documented.

V. Execute

So far, we've discussed how and why various software related to statistical applications, programming languages, and storage of data can be used to build a data science product. There are still plenty of known difficulties to putting together the different components to produce something that can be used by a customer to gain insight into their data.

A project plan can unfold in a number of ways so maintaining an awareness of outcomes as they occur can mitigate the risks (Godsey, 2017). A statistician, who is more comfortable with numbers and equations, should consult with a programmer on how they would approach designing the software, and vice-versa. Even for professional software engineers, it's difficult to discover and eliminate all bugs. Other considerations are to think of a new level of tolerance for errors and bugs to be in place. Having someone test the software thoroughly can provide a fresh outlook, ideally one with similar background to the customer. This can double as a user experience test.

As a plan progresses, new information might surface that may call for modification of the initial plan. These modifications should be made deliberately and with care, as to not affect the progress and final product in a detrimental way. Changes to the software can come from the customer themselves when they are reviewing the progress or from the development team as they face undiscovered hurdles or potential improvements on the original design. Ample time must be allotted to be able to make changes in time for deployment (Godsey, 2017).

2.11.3. Finishing the project

In order to create an effective product that can be delivered with confidence to the customer, the customer's perspective is to be understood first. We must consider who will use the results obtained from the analysis and how useful are the results obtained to them (Godsey, 2017). If the customer is an organization, it's safe to assume that the people that have been interacting with the project team are not the only ones who will use the product, so it is a good idea to understand the type of users that are expected to use it, what actions they intend to take according to the results, and how much of the results are actually relevant to each member.

Once that has been established, the best media and format for wrapping the project up and presenting for the customer to use can be decided. Deciding between an active product, such as an interactive application that allows customers to interact with data and analysis in order to answer questions on their own, or a passive product, such as reports with predetermined output requirements such as text, tables, and figures (Godsey, 2017).

I. Deliver

In addition to deciding the medium in which the results are delivered in, the information contained in the results that will be included in the final product and which to omit can be chosen. The most important and conclusive results are to be emphasized, while information that has little to no impact on the decisions the customers plan to make based on the information should be left out. Any information that might be inconclusive, misinterpreted, or misused should be avoided. Then there are results that lie between being conclusive and inconclusive, which can be included with disclaimers to clarify to the customer the level of significance the results in question hold (Godsey, 2017). As is depicted in the inverted pyramid of journalism, it is most effective to present information in the following order:

1. Lead with the most important, most impactful results in clear language.
2. Include details that directly support those results.
3. Other auxiliary results that are useful but not necessary can be added after that.

Figure 2.13 The Inverted Pyramid of Journalism

This structure can also be adopted in analytics tools and interactive graphical applications, where the most important results and information should be displayed as soon as the application starts or easily accessible, while supporting detail can be accessed with a bit of effort.

Another consideration to make is the user experience, or UX, when interacting with the piece of software. UX has been demonstrated in many contexts that how people interact with software has a large influence on how effect that software is. It can refer to the analytics tool, a report, or any product that can be delivered to the customer. If a customer is not using the product properly, the user experience design might be reconsidered. A user manual is also recommended to make it easier for users who are not be familiar with the product and how to use it. Any support that might be required for the future should be considered and planned for.

As the product is what has been worked on for the duration of the project, it is important to get the format and content right. They must meet the customer's current needs and those in the foreseeable future. Once ready, the product can be delivered to the customer.

II. Revise

Although a lot of time would be spent preparing the product for launch, once the customer begins using it, there are potential problems that might present themselves. These issues can be different types and have a number of solutions. That is why customer feedback and revision of the product are important parts of the development process. The process of recognizing, diagnosing, and fixing problems in the product should be undertaken deliberately and carefully, while taking into account the additional costs that might be incurred (Godsey, 2017).

Some of the problems that might occur are mentioned below:

- Customer not using the product correctly.
- The user experience is subpar.
- There are software bugs.

Once the problems have been identified, solutions can be devised, and the necessary revisions made. It is worth mention that not every problem needs fixing as the solution might not improve much or the cost vs benefits analysis of the revision might indicate that the costs don't make economic sense.

III. Wrapping up

As a project in data science comes to an end, with all that remains is to fix any remaining bugs or other problems, there are things that can be done to improve chances of success in future projects. Two major ways that can achieved, as outlined by (Godsey, 2017), are: to pack away the project with all its modules neatly for reuse or extension in the future, and to learn as much as possible from the project as a whole.

When archiving the project, including documentation can save time and effort in the future. There are 3 levels of documentation according to (Godsey, 2017):

1. **User documentation**
   This is the highest level, what a customer would use to understand the product, which may include any reports, results summaries, description of any application that has been built and delivered. Ultimately it should contain whatever information is necessary for someone to use the product in the intended way.
2. **Developer documentation**
   The middle level, consisting of information that a software developer would want to know to integrate, programmatically use the product, or build a similar product. Sections to be included can be detailed descriptions of APIs, statistical methods implemented, high-level description of software architecture, and data inputs and outputs and their descriptions. This can be presented in the form of a help page within a web application, a readme file, or a technical report.
3. **Code documentation**
   The lowest level of documentation that tells a software developer or maintenance team how the code works at the lowest level, to fix bugs, make improvements, or extend the capabilities of the product. It can include descriptions of objects, methods, inheritance, and explanations of implementation choices made. May exist in the form of comments in the code, README files, and software architectural diagrams.

When storing the project, choices for how to store it are to be made based on disk space required, ease of access, security of sensitive data, volatility of storage media and the expected usability life of the product, and the format it will be stored in. The options for storage include but are not limited to:

- Local drive
- Network drive
- Code repository

When looking back at the decisions made during the development of a project, the knowledge gained will provide insights that can help improve the results of future projects. Looking back at the initial goals and understanding why some weren't achieved and deciding, using the new

insights gained throughout the process, if they are achievable. This can help the approach to similar projects in the future through better courses of action and making better technological choices to achieve the project goals more efficiently (Godsey, 2017).

2.11. Related works

2.11.1. Klipfolio: Business dashboard and analytics software

Klipfolio is an easy to use and flexible online BI Tool that allows you to easily build real-time dashboards (Klipfolio, 2021).

Using Klipfolio brings life to your data due to it using different metrics and KPI's and then displaying that info in a visually pleasing and fully customizable manner. Each dashboard has what we call Klips which are slices of visualized data. You can either add pre-built Klips or create your own. Klipfolio is capable of connecting to almost every data source there is.

Why would you use Klipfolio?

- To streamline your client's financial data for better insights
- To easily scale your analytics processes
- To react in real-time to irregularities in your client's financials
- To enhance organizational communication, keeping everyone in the loop

Advantages:

Custom Styling - There are many ways to tailor the look & feel of your dashboard, from logo to graph colors. But there is also a CSS option that allows you to fully customize how the dashboard looks.

Data connections - There are many ways to connect data to Klipfolio. It's even possible to combine data from multiple sources in one graph or table, without a lot of hassle. The data connections go from uploading a file and connecting Dropbox or Google Drive to more advanced connections like Google BigQuery, MySQL or FTP.

A powerful formula editor - Klipfolio's powerful formula editor allows you to calculate metrics or add functions to your data. This ranges from a simple "SUM" or "AVERAGE" function to "CUMULATIVE" values, standard deviations, regression, etc.

Flexibility - Using a single data source to create multiple data visualizations or use multiple data sources to create a single visualization.

Disadvantages:

- Klipfolio doesn't apply forecasting analysis.
- Klipfolio is not a reporting tool as its only meant for visualizing KPIs.

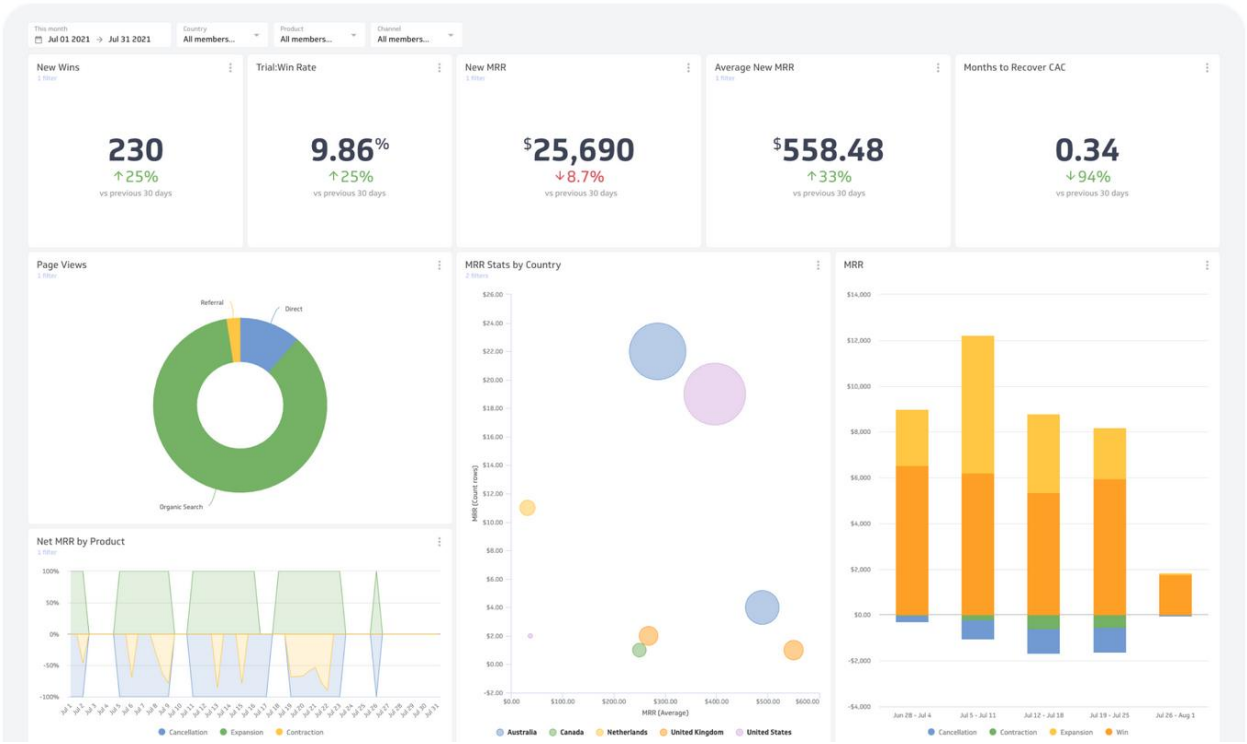


Figure 2.14 Screenshot of the Klipfolio dashboard. Source: (Klipfolio, 2021)

2.11.2. IBM System Dashboard for Enterprise Content Management

The IBM System Dashboard for Enterprise Content Management is a performance monitoring tool that IBM distributes with many of its Enterprise Content Management products and suites for both Windows and UNIX servers (IBM, 2021). The System Dashboard displays real-time performance data that system administrators and operators can use to proactively identify and resolve potential performance problems before they occur. The performance data can also be archived for management reporting and trend analysis.

In addition, the IBM System Usage Reporter is also installed with the System Dashboard. The Usage Reporter is an administrative tool that monitors the number of users who access Enterprise Content Management products and provides both near real-time and historical usage information.

IBM also offers the Enterprise Content Management System Monitor. While the System Dashboard is a performance monitoring tool that tracks information such as application-specific events, the System Monitor continually monitors the health of IBM systems and alerts administrators to critical errors.

System Dashboard features:

- Real-time data from multiple components, such as Content Platform Engine and other servers or workflow systems, can be viewed concurrently
- Capture and display of operating system statistics and environmental data:

    - Operating system name and version number
    - Number and architecture of Central Processing Units (CPUs)
    - CPU load Disk I/O
    - Network I/O
    - Memory usage (amount of memory available)
- Capture and display of FileNet® specific data:
    - Remote procedure calls counts and durations
    - Application-specific Events, such as “Document Creations” in the Content Platform Engine
    - Application-specific Metrics, such as "Folder Cache Entries” in the Content Platform Engine
- Capture and display of environmental data:
    - Configuration
    - Version and patch level
    - Java™ applications provide information about the Java runtime version number and maximum memory
- User-defined charts of summary and detailed data
- Alerts tab that displays all urgent messages that are received from applications and when connection to applications is lost
- Ability to define and run reports and save them into comma-separated value (CSV) files
- Data can be archived and saved for historical analysis and management reporting
- Minimal process consumption in production environments – listeners are built into the Enterprise Content Management products
- Listener agents that can gather data from other applications that can be displayed in the Dashboard

Advantages:

- Works on both Windows and UNIX servers.
- Easy to use.
- Displays real-time performance data.
- System administrators and operators can use it to proactively identify and resolve potential performance problems before they occur.
- The performance data can also be archived for management reporting and trend analysis.
- The System Monitor continually monitors the health of IBM systems and alerts administrators to critical errors.
- Provides both near real-time and historical usage information.
- Provides clustering views.

Disadvantages:

- IBM is an American company so there are currently sanction restrictions.
- Limited Data visualization.
- The software takes up a lot of space.
- Advanced reporting needs experienced user.
- Problems accessing the application configurator after the initial configuration.

- In the Document Processing application, users might encounter issues during finalization of field values.

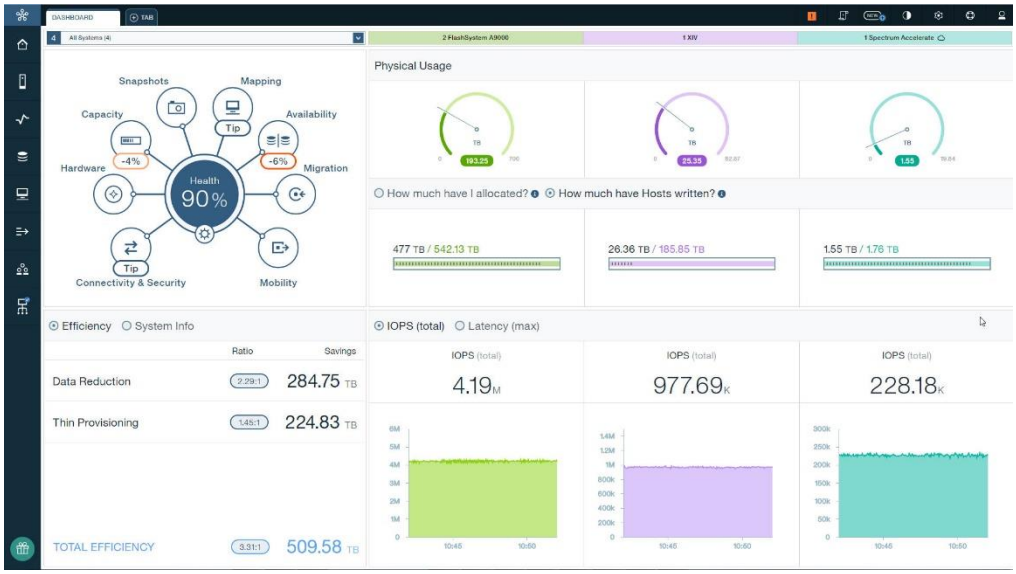


Figure 2.15 Screenshot of the IBM System dashboard. Source: (IBM, 2021)

Chapter 3

METHODOLOGY

In this chapter, we will present the methodology and techniques used for the system development, in addition to filtering, exploring, and analyzing data, as well as the tools used to build the interface. We define the system requirements and conceptualize the proposed system through relevant models.

3.1 Building the Dashboard Web Interface

The interactive dashboard will be built using:

- HTML, CSS, and JavaScript for the web application that will serve as the front-end of the system. Through implementing Bootstrap for web development, the development process will be much quicker.
- Python and its API's will be used for the data analysis and visualization scripts, that will have the back-end functionality. In specific,
    - Dash API, which is made up of combining Flask to supply the web server functionality, React.js to render the UI of the web page, and Plotly.js to generate the charts used in the application.
- SQL will be used to create the database, write queries to retrieve the data stored in the database and present it in a user-friendly way, and to update the database by inserting, modifying, and deleting data.

3.2 The System Methodology: Kanban Model

The chosen methodology for this project is the Kanban Model, a workflow management method for illustration, management, and improvement of services that provide knowledge to users. It aims to facilitate the visualization of the work to be done, maximization of efficiency, and continuous improvement (Radigan, 2021). Kanban offers flexibility on task assignment which will enable the researchers to efficiently produce models and functions in accordance with the modular nature of the system. By periodically meeting with the clients and demonstrating the functionality to the stakeholders as the project progresses, the developers can gain insights from the feedback system that Kanban thrives in (Kanbanize, 2021). As the system is to be implemented for Halan Co. Ltd., it is very important that the final users get to voice their concerns and request features to be added or modified.

Due to confidentiality reasons, access to the company's actual data and current database is limited and so building prototypes using the data templates and dummy data will help with understanding the requirements of the current system and how it can be integrated with the new system. In addition to that, the modular approach of designing the system makes the maintenance and redesign phases easier as each component can be built separately.

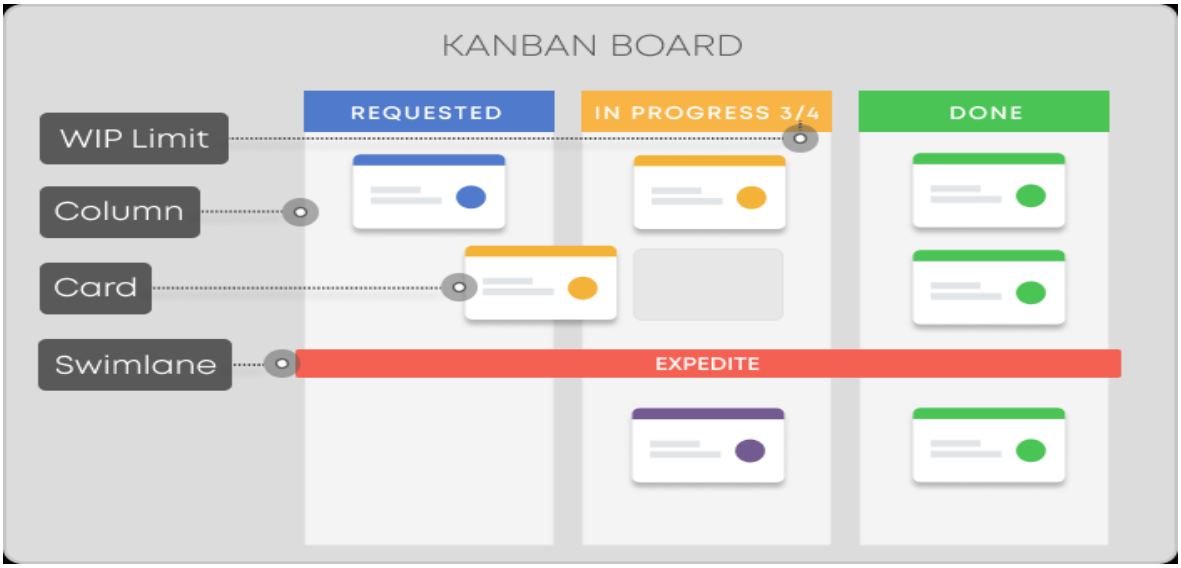


Figure 3.1 The Kanban Board. Source (Kanbanize, 2021)

The breakdown of the flow of tasks to be performed by the researchers is provided below:

- A background reading about performance monitoring dashboards, statistical analysis, and implementations of machine learning algorithms using python is to be conducted.
- Write a proposal for the stakeholders in this project.
- The stakeholders are to be surveyed to find out what problems they face and what their needs are and confirm if they will benefit from the proposed system. Use this information to set the goals of the project.
- Devise a Budget and a time schedule for the project.
- Request relevant data template from the organization and create dummy data to be used for the development of the analytics software. The data mentioned is discussed in the data dictionary.
- Conduct initial data exploration to gain insights and decide on the best methods to conduct the analysis and gain a better understanding of the current system.
- Clean and structure the data through data wrangling.
- A Dataset of relevant data for analysis is to be constructed.
- Analyze the data in depth and decide which machine learning algorithm will provide the best insights with the highest performance levels.
- Build the analytic functions and machine learning models for production.
- Present progress report to the organization and request feedback on any modifications and additional features.
- Create the database that will be used to store and modify the data on a regular basis.
- Develop the dashboard web application to act as the frontend of the system.
- Link the system components to each other.
- Test the system and present the complete prototype to the organization.

- Revise the system with any modifications and prepare for deployment.
- Link database with the organization's database.
- Launch the beta version.
- Any future maintenance to be made when necessary.

3.3 System Requirements

3.3.1 System Functional Requirements:

Users:

Backend: an administrator charged with maintaining and managing the, keeping the server up and running, overall security procedures, updating the databases, and providing access to authorized users of the system.

Frontend:

1. General Manager:
    - Get a dashboard detailing the following KPI's:
        - GMV (gross merchandise value) which is an indicator of the total amount of money circulating around the company including (order value, drivers cut and Halan's margin).
        - The stores estimated number of orders vs the real number of orders.
        - Induvial stores Delivery percentages.
2. SME Operations Manager:
    - Get a dashboard detailing the following KPI's:
        - Active stores in the current month.
        - The driver's delivery success rate to monitor for fraud.
        - Current month Performance vs current month goal.
3. Drivers Supervisor:
    - Get access to a portal to monitor drivers' performance.
4. Data Entry Officer:
    - Get access to a portal to enter new orders information.
5. Accountant:
    - Get access to a portal to update order status and other changes according to the delivery report.

3.3.2 System non-functional requirements:

Security Features: this tool will be used by different employees with different privileges and authorization levels so applying security procedures and policies to provide more security layers to the data is of utmost importance.

Logging and reporting: The system should be able to log relevant information at each step so that monitoring and reports can be generated and printed.

notification system: The system should be able to alert the user if something went wrong or if some goal isn't going to be met with the current trajectory of the data.

Portability: Having the system in web-application form allows it to be platform dependent and thus achieving the portability goal.

Robustness:

- If an error were to occurs during the data saving process, the previously saved data must be preserved. Only after saving is successful should the previous version of the data be eliminated.
- Users should not be able to enter invalid data; when an attempt is made to enter invalid data, users should be prompted with an error message and offered the opportunity to fix it.

3.3.3 User Functional requirements:

i. The system should be able to deliver the following KPI's in a visually pleasing and easy to understand way:

   - Number of orders received.
   - Number of orders distributed.
   - Number of orders delivered.
   - Number of orders cancelled.
   - Number of orders held.
   - Number of drivers who worked/day.

ii. The system should be able to relay the following information in a clear and logical way
   - How many orders were completed per day?
   - How many stores were active in the current month?
   - why orders are failing?

iii. A function capable of calculating how much to pay the driver and how much Halan's profit margin is:
   - Order value – Driver's cut = Halan's cut.
   - Driver's cut X Number of completed orders= Driver's Total

iv. The system should be able to provide forecasting of the current month trajectory.

3.3.4 Users non-functional requirements:

- Usability: Experienced controllers should be able to use all system functions after a total of 3 hours training.

- Reliability: the system should be reliable have minimal to no lags and deliver the required outcome.
- Portability: Due to its web-based structure the system will work with any device with a web browser this achieving the portability goal.
- Responsiveness: the app should work well with different screen sizes and devices.

3.4 Domain requirements:

- Embedded BI Dashboards and Dashboard Designers: the users should be able to view, manipulate, and build dashboards.
- Cloud-Based Access: web-based dashboard tools allows users to access their data sources from anywhere at any time.
- Optimized for mobile devices: Dashboard should be clearly viewable no matter the screen size of the device, thus offering a larger range of usability.
- Role-Based Data Views: The administrator should be able to dictate the things that each user is able to see.
- Real-Time Data: The dashboard should be able to report off live data, not just cached data.
- Automated Data Refreshes: To protect the database and users from traffic jams.
- Predefined Chart Themes: color harmony is an important aspect of designing visualizations.
- Parameters or variables to be used for filtering and transforming data.
- Global and Local Filters: able to filter elements through Static and Interactive Filters on the dashboard and display the filter values.
- Dynamically Change Charts on the Fly: providing the user with the ability to change the chart type, color scheme, and sort order.
- Printability/Exporting visualizations.

3.5. Architectural Design Object Model

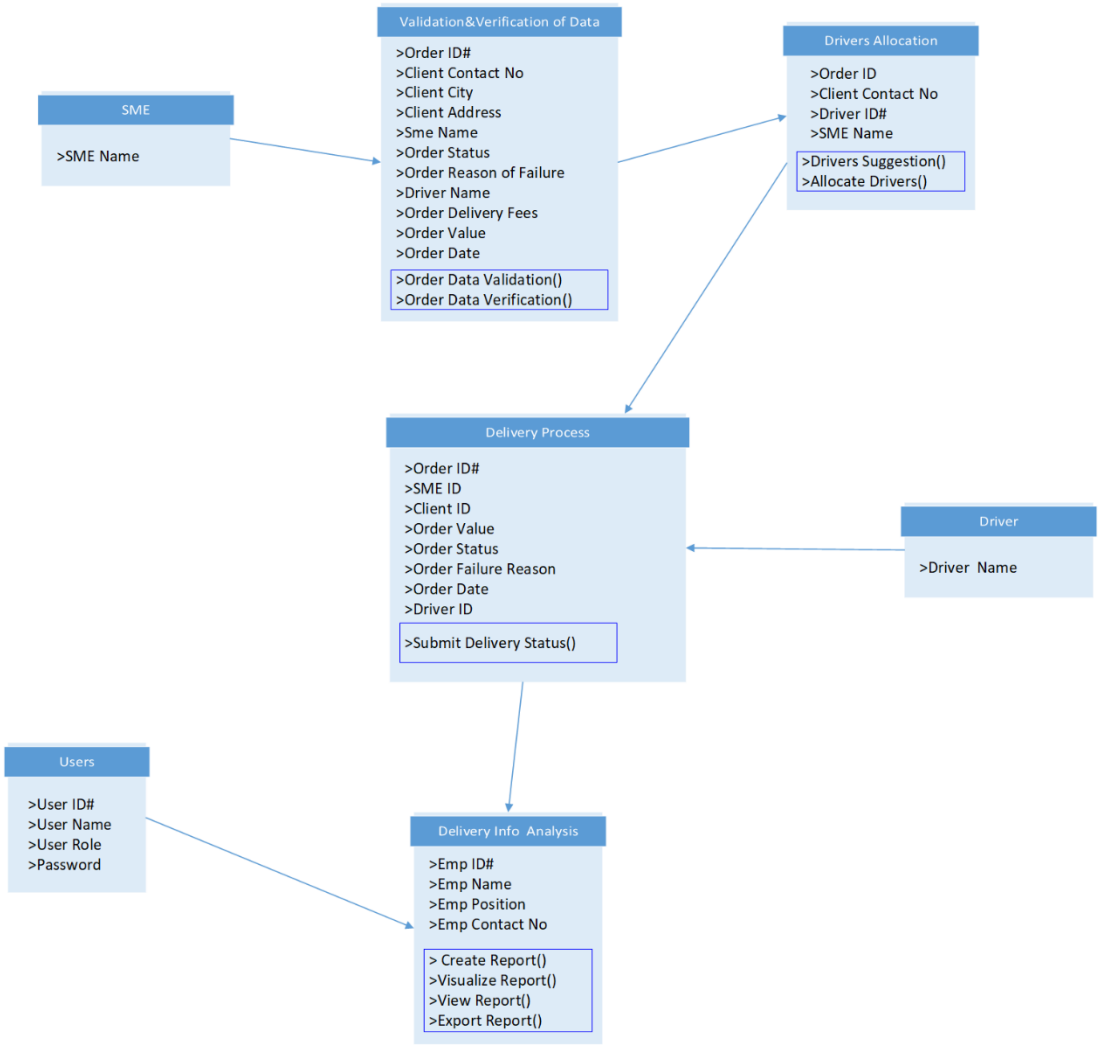


Figure 3.2 Architectural Design Object Model

3.7. Relationships Model

3.7.1. Current Data Flow Diagram Level 0 – Context Level

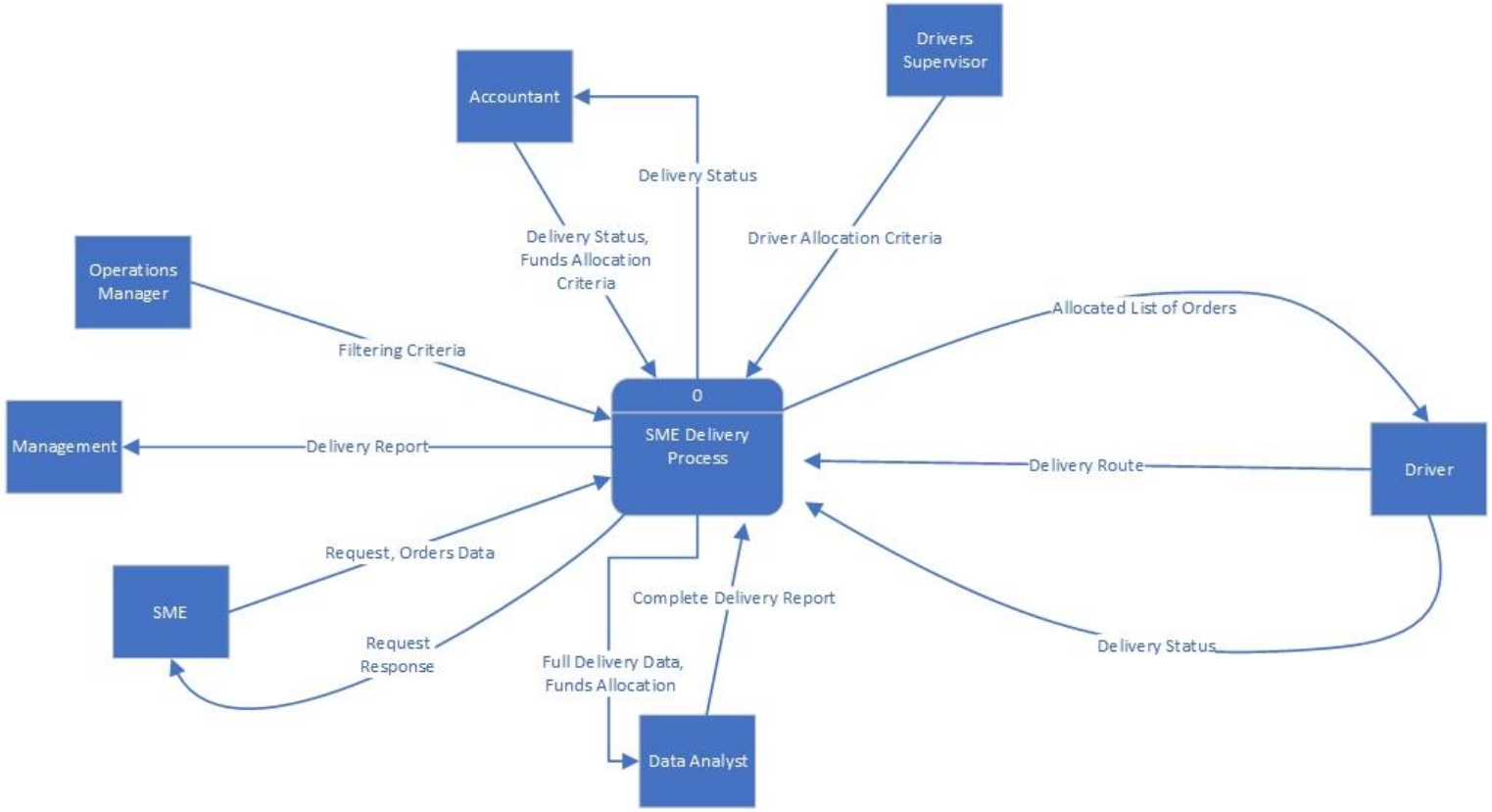


Figure 3.3. Current DFD Level 0- Context Level

3.7.2. Current Data Flow Diagram Level-1

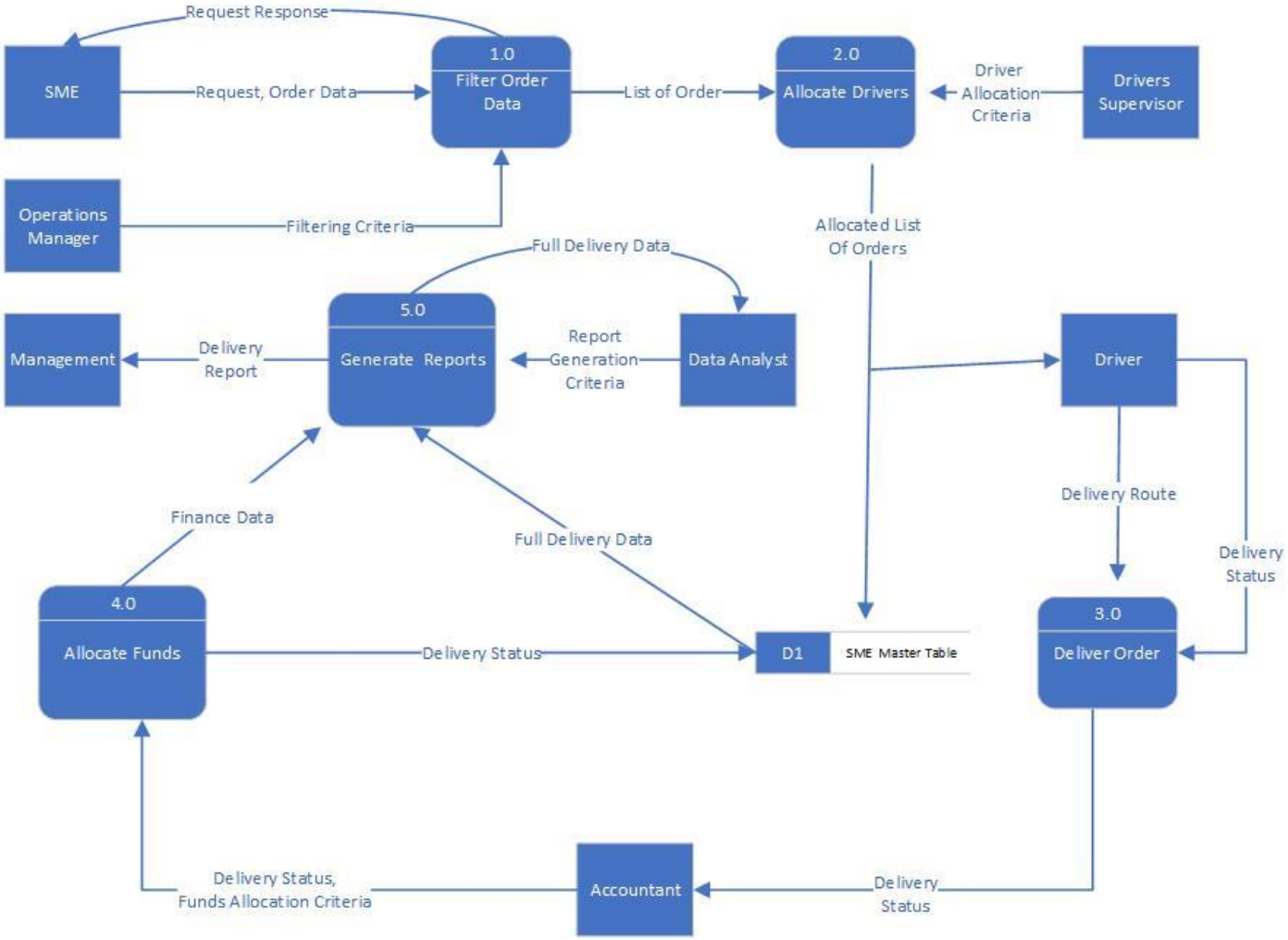


Figure 3.4. Current DFD Level - 1

3.7.3. Proposed Data Flow Diagram level 0 – Context Level

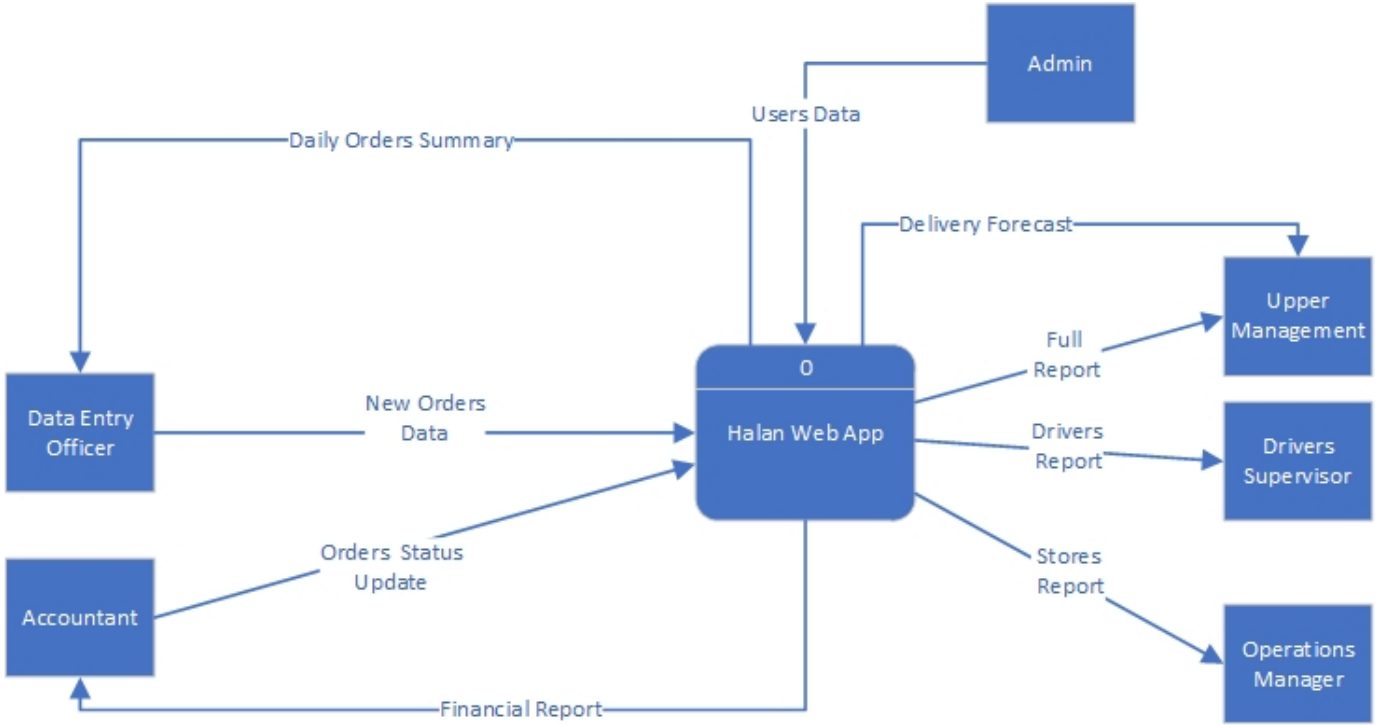


Figure 3.5. Proposed DFD level 0 - Context Level

3.7.4. Proposed Data Flow Diagram – Level-1

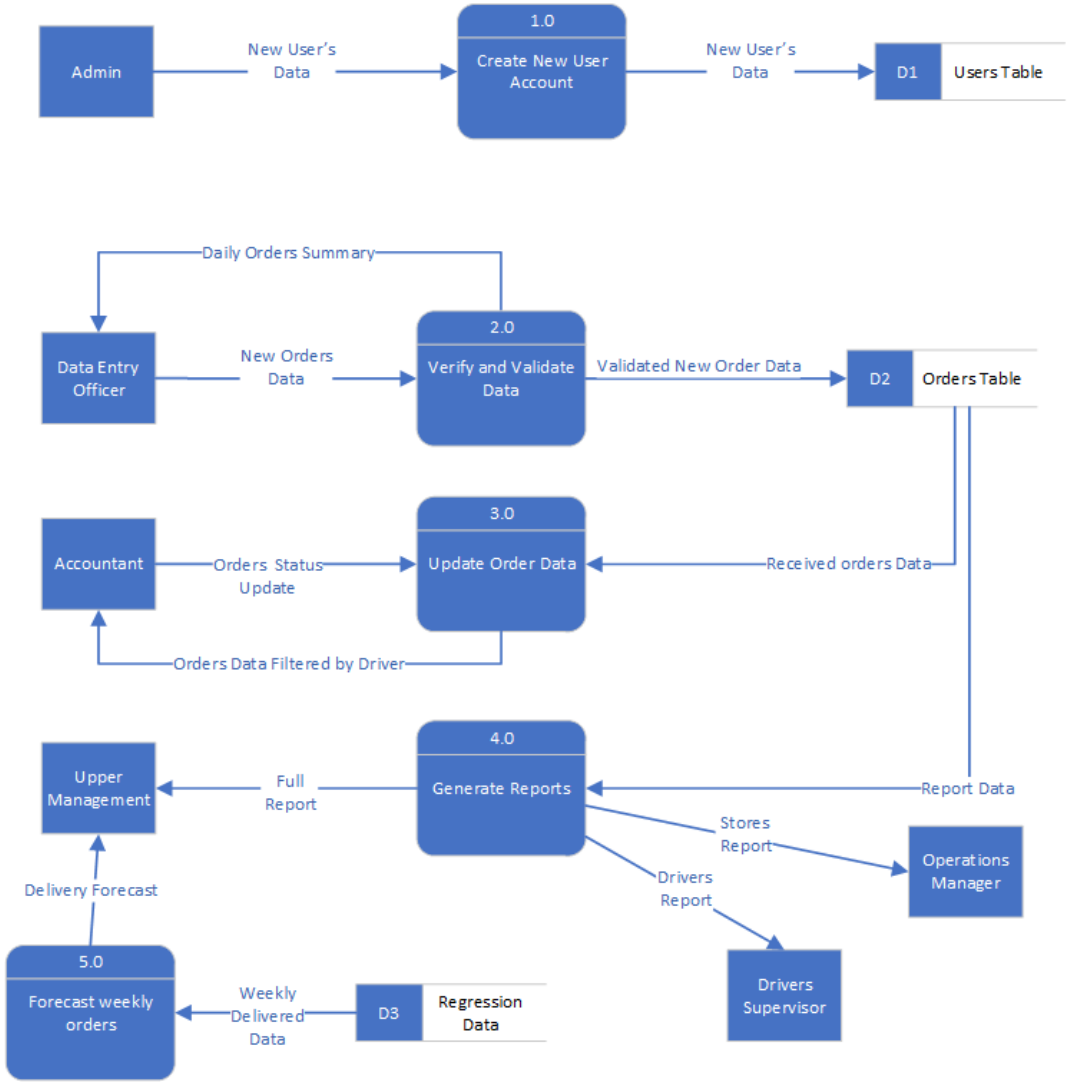


Figure 3.6 Proposed DFD - Level - 1

3.7.5. Entity Relationship Diagram (ERD) of the Proposed System

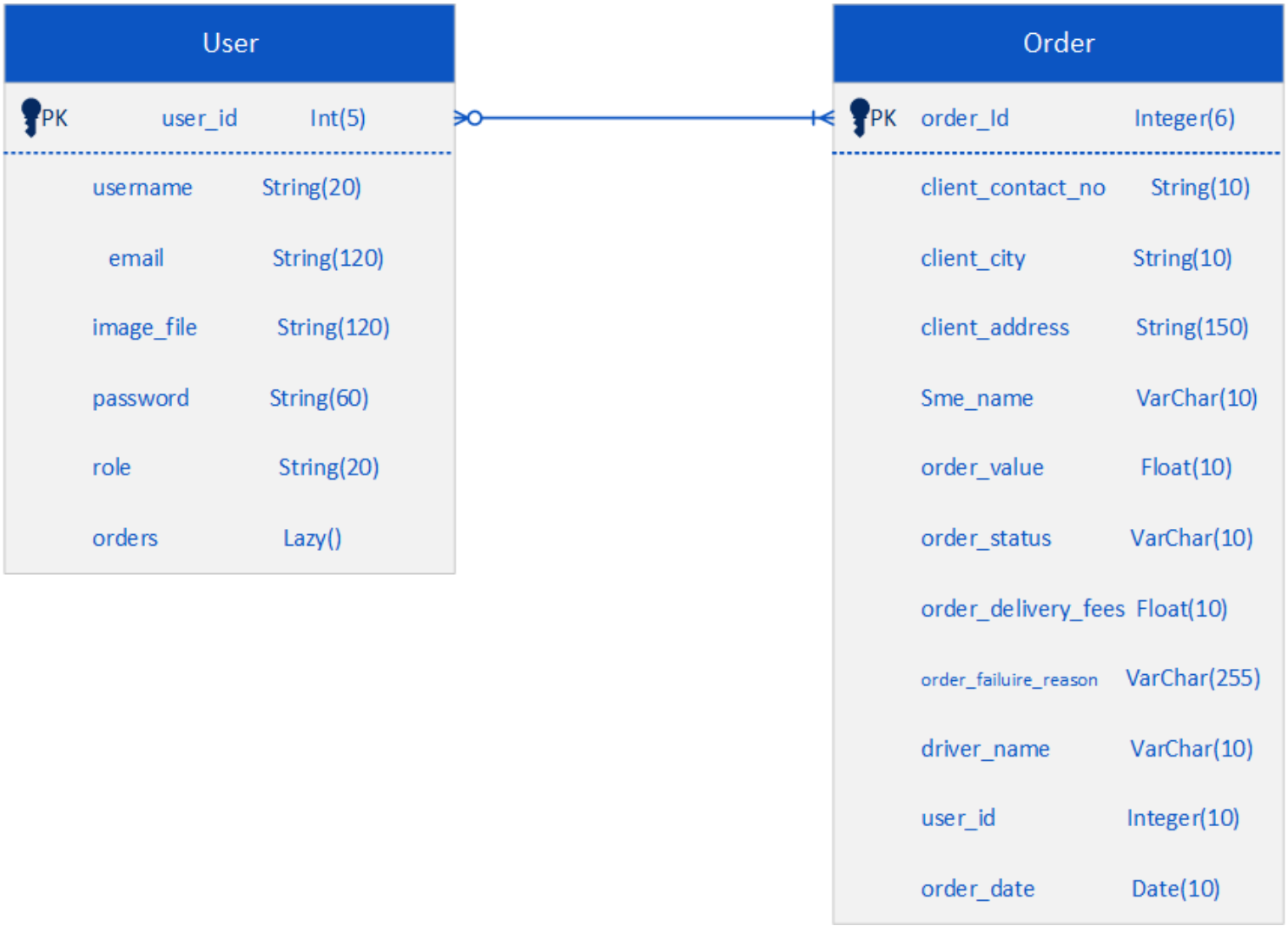


Figure 3.7 ERD of the proposed system

3.8. Data Dictionary

Table 3.1 User Table

Field Name	**Data Type**	**Constraint**	**Field Length**	**Description**	**Example**
emp_id	VarChar	Primary Key	10	Unique alphanumerical ID for all employees	AC0007
emp_name	VarChar		255	Full name of employee	Mohamed Ayman
emp_position	VarChar		255	The employee's position in the organization	Accountant
emp_contact_no	VarChar		10	Employee's contact number	0912245678

=

Table 3.2 Order Table

Field Name	**Data Type**	**Constraint**	**Field Length**	**Description**	**Example**
order_id	Integer	Primary Key	6	Unique autogenerated numerical ID for all orders	262001
sme_name	VarChar	Unique	10	Unique numerical ID for all SMEs	SME0001
client_contact_no	String	-	10	Unique numerical ID for all clients	0912245678
Client_city	String	-	10	Drop down menu to choose from one of three cities of Khartoum	Khartoum Bahri Omdurman
Client_address	String	-		A description of the client address for the driver	Mecca x 117st near my place
order_value	Float	Predetermined values	10	The order's value, correct to 2 decimal places	2100.00
order_delivery_fees	Float		10	The amount of money paid by the client for the order delivery service	700.00

order_status	String	Predetermined values	10	Drop down menu to choose from the order's current status	Received Cancelled Hold Delivered
order_failure_ reason	String	Predetermined values	255	Explain why order was either cancelled or put on Hold	•The customer is not at the specified location •The customer is not answering their phone •The customer's phone is switched off •The wrong order information has been provided •The customer requested to receive the order another day
driver_name	Strimg	-	100	The name of the driver who will deliver the order	Omer Banga
order_date	date	-	10	Autogenerated	DD/MM/YYYY

3.9. Behavior of the system

3.9.1. Use case diagram

Table 3.3 Use Case Description

Halan Web Application	
Actors	Data Entry Officer (DEO)– Enter New order data Drivers Supervisor (DS) – allocate order data according driver Drivers – Update delivery status Accountant – Fact checks delivery report with received cash and updates order status. Management – View report and makes decisions accordingly
Description	The Delivery process begins when the SME owner provides the organization with a list of orders and the DEO inputs their order data into the portal that has been designed for data entry. They input the data in batches and once done, the system verifies and validates the data integrity, uploads it to the database. The DS uses their portal to allocate which deliveries will be handled by which driver. The driver then takes their assigned orders and completes the delivery and provides the accountant with an updated status for each order through the portal. The delivery status of the orders can be: • "Received" • "Cancelled" • on "Hold" • "Delivered" The reasons of failure to deliver can be, but not limited to, one of the following: • The customer is not at the specified location • The customer is not answering their phone • The customer's phone is switched off • The wrong order information has been provided • The customer requested to receive the order another day • Potential fraudulent reason from the driver

	Once the Driver has finished updating the order status, that data is uploaded to the database, the system calculates the way the funds will be distributed between the SME, the Driver, and Halan. The accountant then reviews it and can make any modifications. This is especially important as there are different discount rates for the delivery fee imposed on the SMEs so it is crucial to be vigilant and make sure all values are correct. The accountant then distributes the funds accordingly. The system generates a report according to the range of dates specified by the DA, who then reviews the report and submits the report for the management to read and use the insights obtained to make decisions.
Data	User Login Details New Order Data Delivery status Report Date Range
Stimulus	User command issued by Actors
Response	Confirmation on Order Data Uploaded Correctly Driver Allocation Updated View Delivery Status Updated View Distribution of Funds Finalized Order Data Pushed to the Database Final Generated Report
Comments	The system is currently designed for Halan's SME division. If expanded to other branches, different clearance levels need to be made so that actors cannot view data pertaining other branches. The Delivery and the Data Analysis Processes are effectively separate subsystems. The Data Analysis process takes its input data from the database, which takes its data from the actors in the Delivery process.

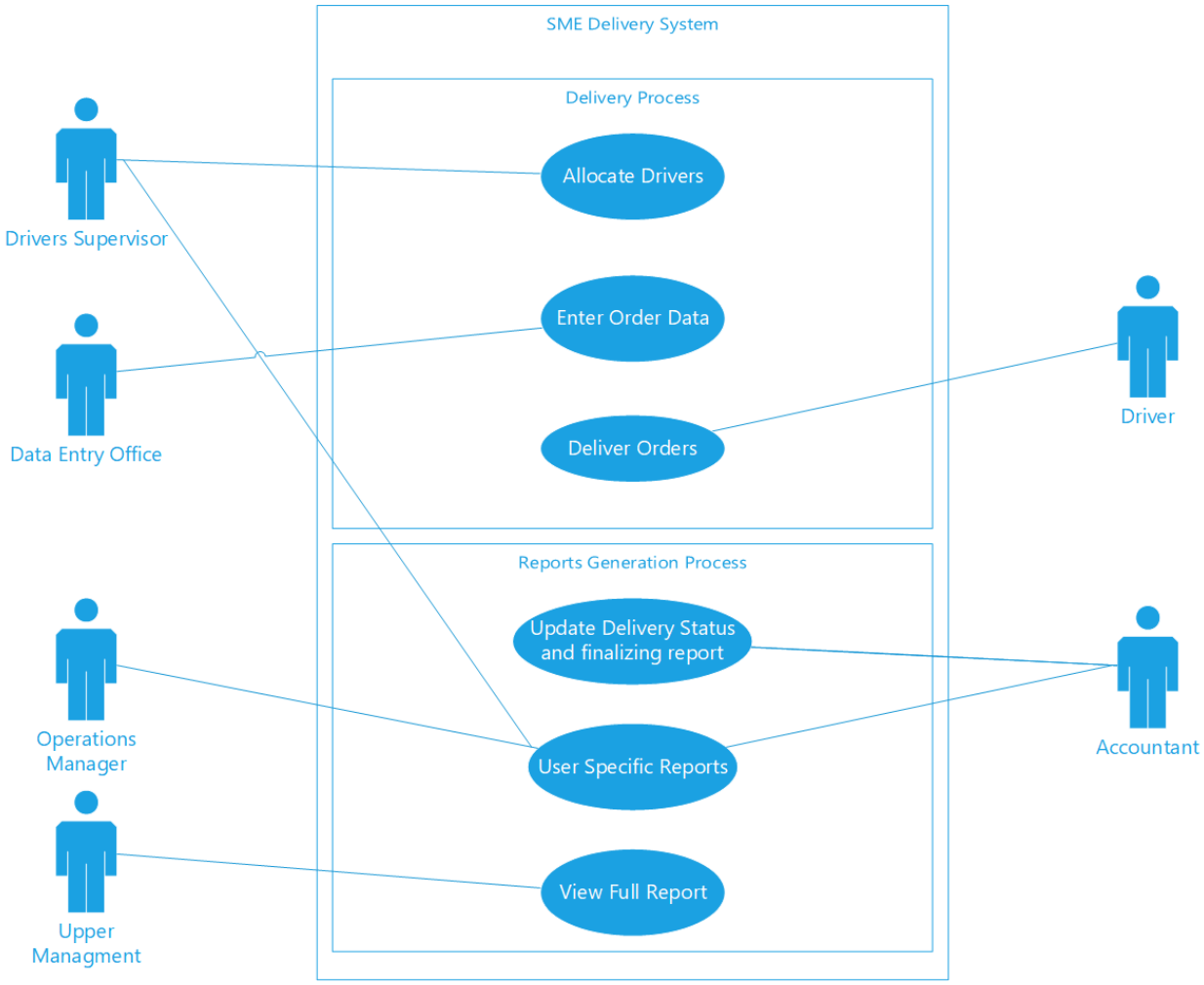


Figure 3.8 Use Case of the System

3.10. Interface Model

3.10.1. Web Application Architecture

In addition to the implementation of machine learning algorithms, this system will be achieved via Python 3 and Dash API to build the dashboard web application, which will function as the frontend of the system. The Backend will be managed by using MySQL database tools, which will be the source of the data that will be extracted and analyzed for patterns using python's scientific computing packages in order to discover key characteristics and hidden knowledge.

Frontend:

- Bootstrap (Html, CSS, JavaScript), Dash API (plotly.js and React.js)

Backend:

Web Server:

- JavaScript, Python, Dash API (Flask)

File System:

- HTML, CSS, Python, Excel

Database:

- MySQL

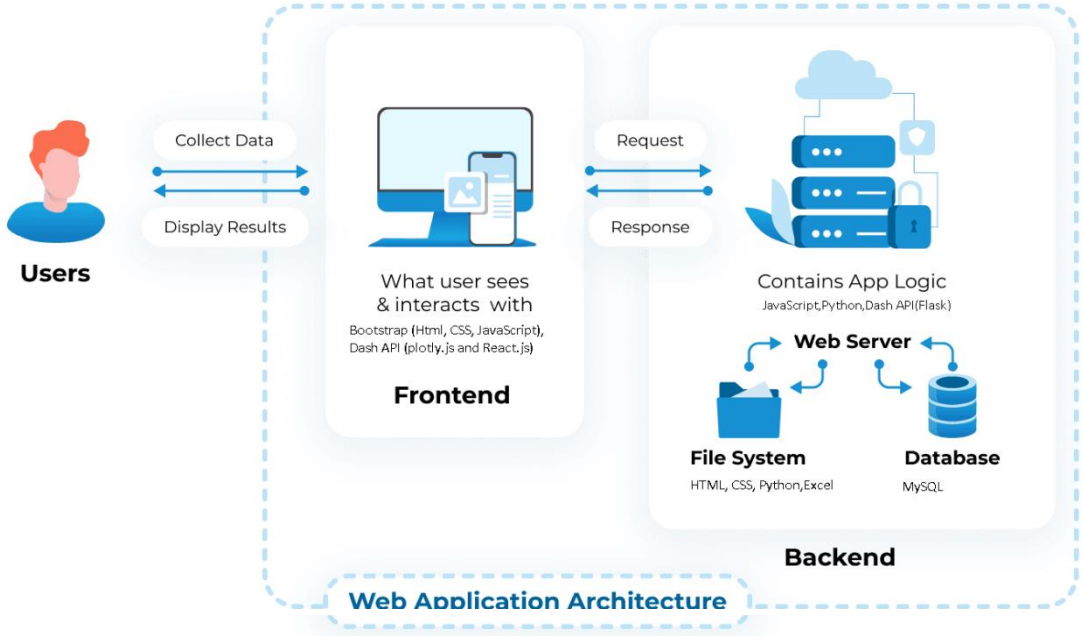


Figure 3.9 Web Application Architecture

CHAPTER 4

RESULTS AND DISCUSSION

This chapter discusses the result of the system design and its implementation. We describe the results of each of the Data Science Process stages in the development of the Decision Support System through a Performance Monitoring Dashboard, including the software tools used, their features, and a justification for their advantages. We then discuss the testing strategy for the dashboard system and the process leading to the choice of a machine learning algorithm.

4.1. The Data Science Process

4.1.1. Prepare

In this stage we set the project's goals based on the requirements and KPI's of the stakeholders that have been determined through interviews, observations of the work environment, and the organization's raw data and their internal documents.

The researchers have determined the main problems faced by the stakeholders during the organization's currently implemented process. For the organization to stay competitive, decision makers are required to make decisions and take appropriate action based on the information that they have. The time consumption in analyzing the data and completing tasks because of the organization's implemented process. It is restricted to weekly and monthly reports, thus hindering their response time to any changes to their environment and meeting their business objectives. The organization does not have a centralized platform where they can conduct their daily operations including data entry, accessing historical data, visualization of trends, and creating reports.

The usage of spreadsheets as a be all and end all solution for inputting, storing, archiving, and analyzing the data is prone to input errors and empty fields, susceptible to users tampering due to lack of security level access, makes it difficult to retrieve records, and time consuming when making seasonal reports respectively. It also requires some level of technical proficiency to manipulate and interact with.

The organization implements the Least Squared Trend Projection Timeseries to forecast the monthly and weekly targets. The data analyst uses spreadsheets to implement the algorithm, which is both time consuming to implement, update and optimize manually with newly obtained data.

The main stakeholders that were interviewed were the organization's General Manager, the Operations Manager, and the Data Analyst. The KPI's obtained from the interviews and observations mentioned above are listed in table 4.1:

Table 4.1 The organization's Key Performance Indicators

KPI	Sub-KPI	Description
The Gross Merchandise Value (GMV)	GMV Consists of: • Halan's Margin • Drivers' Return • Stores' Return	Indicates how well they are performing since the SME branch is currently not profit oriented.
Sum of orders Status	• Number of orders Received • Number of orders Delivered • Number of orders Cancelled • Number of orders Held	Quantification of each order status over time.
Order Status Percentage	• Percentage of orders Delivered • Percentage of orders Cancelled • Percentage of orders Held	Indicates the success and failure rate of the delivery operations.
	With respect to: • All of Halan's received orders • Each individual Driver received orders • All order received from each individual Store	
Order reason of failure	-	Reasons orders have failed to be delivered to monitor fraudulent activities.

4.2. Software Tools

Mainly we used Python programming language which is considered to be the most popular

language amongst the current generation of data scientists. We also use Flask for the backend web framework, Plotly Dash for the dashboard application and visualizations, PostgreSQL and SQLAlchemy for the database creation, querying, and updating records in the database

Python is an open-source software that supports several platforms and has a large community, which is continuously extending and improving their libraries and tools. Python provides a more general approach to data wrangling and has many data science related libraries to help analyze, visualize, and display data. With its multiple capabilities, Python is the most suitable tool for conducting data analysis and training ML models in a scalable production environment.

Pandas is a Python library used for data wrangling and analysis, built around the Dataframes (DF), which is essentially a table with rows and columns. Pandas provides a range of methods to modify and operate these tables. In contrast with NumPy, which requires all its array values to be of the same type, pandas allows each column to have a separate type. Another benefit of pandas is that it can take data as input in many different file formats and databases.

NumPy, one of the fundamental packages for scientific computing in Python, contains functionality for multidimensional arrays, high-level mathematical functions such as linear algebra operations, and pseudorandom number generation. The NumPy array is the fundamental data structure used in scikit-learn's data input.

Flask is a web application framework written in Python that makes it incredibly easy to build web applications with Python. Flask is based on the Werkzeug, WSGI toolkit and the Jinja2 template engine.

WSGI or Web Server Gateway Interface has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

Werkzeug is a WSGI toolkit that implements requests, response objects, and utility functions. This enables a web frame to be built on it. The Flask framework uses Werkzeug as one of its bases.

HTML provides the basic structure of sites which is then enhanced and modified by other technologies like CSS and JavaScript. CSS is used to control presentation, formatting, and layouts. JavaScript is used to control the behavior of different elements. Together, they serve as the front-end of the system.

Bootstrap, an open-source CSS framework, the directed at responsive, mobile-first front-end web development. It contains HTML, CSS, and JavaScript.

Jinja is a popular templating engine for Python. A web template system combines a template with a specific data source to render a dynamic web page.

Dash is an open source framework for building interactive data visualization interfaces (Castillo, 2021). Data scientists can use Dash to help build analytical web applications without requiring advanced web development skills. In its core, dash is made up of combining Flask to supply the web server functionality, React.js, a JavaScript library for building user interfaces (UI), to render the UI of the web page, and Plotly.js, a "high-level declarative charting library" as stated in their official website "plotly.com", to generate the charts used in the application. Data scientists just need to write their Python, R, or Julia script and some CSS for added aesthetics.

SQLAlchemy, a Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL by providing a full suite of well-known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language. The benefits of using SQLAlchemy is that it simplifies the development process by allowing the researchers to use SQLite for the development phase and

later use the same code to integrate with any other database such as PostgreSQL which we are using for production.

PostgreSQL is an advanced, enterprise-class, and open-source relational database system that is backed by more than 20 years of development by the open-source community, resulting in it being a highly stable database. Supporting both SQL (relational) and JSON (non-relational) querying, it is used as a primary database for many web applications as well as mobile and analytics applications.

Scikit-learn is an open-source project that is constantly under development and improvement, containing a number of state-of-the-art machine learning algorithms. It is used alongside NumPy, SciPy, and matplotlib, to provide a well-rounded model development environment.

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows. Throughout the development process, we opted to use Git for its version control. This allowed us to rollback any unwanted changes and reduce the potential of losing any made progress.

4.3. Preprocessing

Using pandas in python, the data was imported and analyzed. We use the describe() and info() functions to gauge the integrity of the data, in terms of missing and repeating values, incorrect inputs, and illogical values.

```
#importing the raw data using the pandas Library
df = pd.read_excel('sme_sample.xlsx')
df.describe()
```

✓ 0.7s

	Count	Client Mobile NO	Order ID	Business Address	Fees	Order Value
count	2419.000000	2.417000e+03	2419.000000	19.000000	2419.000000	2419.000000
mean	64.558082	7.668810e+08	248005.254651	52245.526316	558.515916	3044.342704
std	47.434028	7.951995e+08	12295.344921	11.563200	125.286392	3396.143589
min	1.000000	1.100997e+07	232001.000000	52229.000000	400.000000	-17750.000000
25%	27.000000	1.274598e+08	236083.500000	52234.500000	450.000000	1100.000000
50%	55.000000	9.123511e+08	246047.000000	52250.000000	650.000000	2200.000000
75%	94.000000	9.620698e+08	257056.000000	52253.500000	700.000000	4250.000000
max	239.000000	9.937940e+09	272044.000000	52264.000000	700.000000	26900.000000

Figure 4.1 importing the data raw

```
df.info()
```

✓ 0.6s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2419 entries, 0 to 2418
Data columns (total 17 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Count              2419 non-null   int64
 1   Client Mobile NO   2417 non-null   float64
 2   Order ID           2419 non-null   int64
 3   Business City      2419 non-null   object
 4   Business Address   19 non-null     float64
 5   Client City        2419 non-null   object
 6   Client Address     2344 non-null   object
 7   Client Zone        4 non-null      object
 8   Business           2419 non-null   object
 9   Status             2418 non-null   object
 10  Reason of failure  545 non-null    object
 11  Driver             2419 non-null   object
 12  Fees               2419 non-null   int64
 13  Viecle             2419 non-null   object
 14  Order Value        2419 non-null   int64
 15  Order Status       6 non-null      object
 16  Date               2419 non-null   datetime64[ns]
dtypes: datetime64[ns](1), float64(2), int64(4), object(10)
memory usage: 321.4+ KB
```

Figure4.2 Obtaining information about each feature in the data

The output shows null values for a few columns and some negatives in the order value. We also found incorrectly inputted phone numbers, which have been removed. There were inconsistencies in the client's address and the corresponding client city, which do not match geographically, and the vehicle (viecle in the data) is the same for all orders, namely "Motorbike", thus making it redundant. The steps taken to clean the data are outlined in the code snippets below:

4.4. Building the system

```
#importing the raw data using the pandas Library
sme_main = pd.read_excel('sme_sample.xlsx')
#renaming the columns
sme_main.rename(columns={'Client Mobile NO': 'client_contact_no',
 'Order ID': 'order_id', 'Client City': 'client_city', 'Client Address': 'client_address',
 'Business': 'sme_name',  'Status': 'order_status', 'Reason of failure': 'order_reason_of_failure',
 'Driver': 'driver_name', 'Fees': 'order_delivery_fees', 'Order Value': 'order_value','Date': 'order_date'}, inplace=True)
#Extracting orders with negative values, indicating funds receivable
negative_order_values=sme_main[sme_main['order_value']<0]
#Extracting the failed orders
failed_orders= sme_main.dropna(subset=['order_reason_of_failure'])
#Dropping irrelevant columns and those with null values,repeating values, and incorrect order_id
sme_main.drop(['Count','Order Status','Business Address','Business City','Viecle','Client Zone'], axis=1, inplace=True)
```

Figure 4.3 Data Cleaning. Dropping, and renaming of columns

```
#Dropping negative order values, null or incorrect IDs and contact numbers and null client addresses
sme_main = sme_main[sme_main['order_value']>=0]
sme_main.dropna(subset=['order_id','client_contact_no','client_address'], inplace=True)
sme_main.drop(sme_main[(sme_main['order_id']< 100000) ].index,inplace=True)
sme_main.drop(sme_main[(sme_main['order_id']> 999999) ].index,inplace=True)
sme_main.drop(sme_main[(sme_main['client_contact_no']> 999999999) ].index,inplace=True)
sme_main.drop(sme_main[(sme_main['client_contact_no']< 100000000) ].index,inplace=True)
#One of the order statuses was null and after referring to the org, delivered was replaced
sme_main[['order_status']] = sme_main[['order_status']].fillna('Delivered')
#Formatting date
sme_main['order_date']=pd.to_datetime(sme_main['order_date'],format='%Y-%m-%d')
#exporting the cleaned data to csv to then be uploaded into the database
sme_main.to_csv('order_table.csv', index=False)
```

Figure 4.4 Data Cleaning cont. Dropping irrelevant or repeating columns and saving it as a .csv file.

```
#Creating new features
sme_main['day'] = pd.DatetimeIndex(sme_main['order_date']).day
sme_main['year'] = pd.DatetimeIndex(sme_main['order_date']).year
sme_main['month']= pd.DatetimeIndex(sme_main['order_date']).month
sme_main['week'] = (sme_main['order_date'].dt.strftime('%W').astype(int) + 1)
sme_main['month_name'] = sme_main['month'].apply(lambda x: calendar.month_name[x])
#calculating stakeholders individual shares
sme_main['driver_fee'] = sme_main['order_delivery_fees']*0.7
sme_main['halan_return'] = sme_main['order_delivery_fees']*0.3
sme_main['sme_return'] = sme_main['order_value'] - sme_main['order_delivery_fees']
sme_main.info()
```

Figure 4.5 Transforming the data and extracting new features based on existing ones.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2284 entries, 0 to 2418
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   client_contact_no       2284 non-null   float64
 1   order_id                2284 non-null   int64
 2   client_city             2284 non-null   object
 3   client_address          2284 non-null   object
 4   sme_name                2284 non-null   object
 5   order_status            2284 non-null   object
 6   order_reason_of_failure 509 non-null    object
 7   driver_name             2284 non-null   object
 8   order_delivery_fees     2284 non-null   int64
 9   order_value             2284 non-null   int64
 10  order_date              2284 non-null   datetime64[ns]
 11  day                     2284 non-null   int64
 12  year                    2284 non-null   int64
 13  month                   2284 non-null   int64
 14  week                    2284 non-null   int32
 15  month_name              2284 non-null   object
 16  driver_fee              2284 non-null   float64
 17  halan_return            2284 non-null   float64
 18  sme_return              2284 non-null   int64
dtypes: datetime64[ns](1), float64(3), int32(1), int64(7), object(7)
memory usage: 348.0+ KB
```

Figure 4.6 The result of preprocessing.

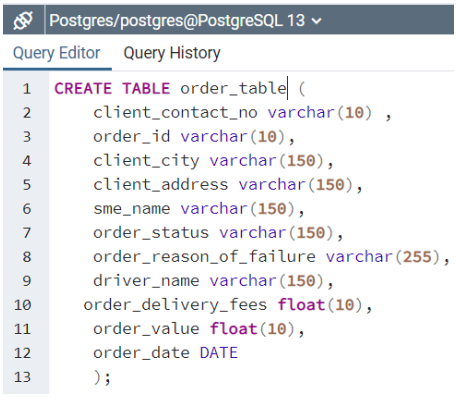
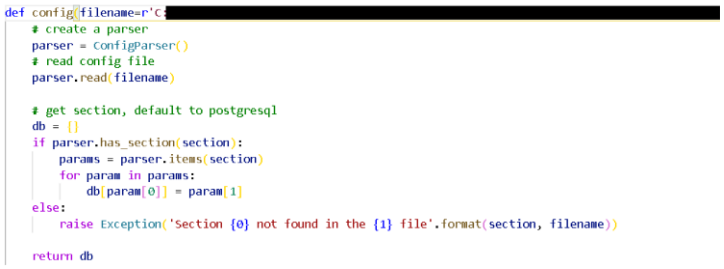


Figure 4.7 Creating the Database tables

Figure 4.8 Inserting the preprocessed data into the database from a .csv file

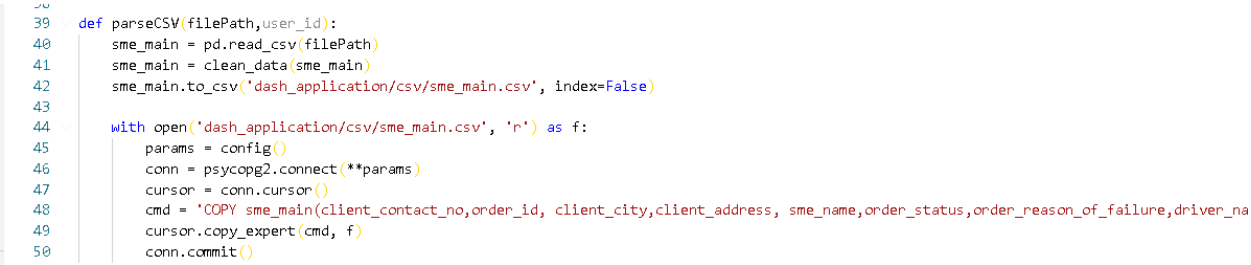


```
def config(filename=r'c
    # create a parser
    parser = ConfigParser()
    # read config file
    parser.read(filename)

    # get section, default to postgresql
    db = {}
    if parser.has_section(section):
        params = parser.items(section)
        for param in params:
            db[param[0]] = param[1]
    else:
        raise Exception('Section {0} not found in the {1} file'.format(section, filename))

    return db
```

Figure 4.9 Function that sets the database configuration from the config file.



```
39  def parseCSV(filePath,user_id):
40      sme_main = pd.read_csv(filePath)
41      sme_main = clean_data(sme_main)
42      sme_main.to_csv('dash_application/csv/sme_main.csv', index=False)
43
44      with open('dash_application/csv/sme_main.csv', 'r') as f:
45          params = config()
46          conn = psycopg2.connect(**params)
47          cursor = conn.cursor()
48          cmd = 'COPY sme_main(client_contact_no,order_id, client_city,client_address, sme_name,order_status,order_reason_of_failure,driver_na
49          cursor.copy_expert(cmd, f)
50          conn.commit()
```

Figure 4.10 Code that inserts new data from the website through uploading csv files.

```python
def input_data():
    month = str(datetime.now().strftime('%B'))
    # Establish a connection to the database by creating a cursor object
    # Obtain the configuration parameters
    params = config()
    # Connect to the PostgreSQL database
    conn = psycopg2.connect(**params)
    # Create a new cursor
    cur = conn.cursor()
    # A function that takes in a PostgreSQL query and outputs a pandas database
    def create_pandas_table(sql_query, database = conn):
        table = pd.read_sql_query(sql_query, database)
        return table
    # Utilize the create_pandas_table function to create a Pandas data frame
    # Store the data as a variable
    sme_main = create_pandas_table("SELECT * FROM sme_main")
    # Close the cursor and connection to so the server can allocate
    # bandwidth to other requests
    cur.close()
    conn.close()
    sme_main['order_date']=pd.to_datetime(sme_main['order_date'],format='%Y-%m-%d')
    sme_main['day'] = pd.DatetimeIndex(sme_main['order_date']).day       You, 3 months ago
    sme_main['year'] = pd.DatetimeIndex(sme_main['order_date']).year
    sme_main['month']= pd.DatetimeIndex(sme_main['order_date']).month
    sme_main['week'] = (sme_main['order_date'].dt.strftime('%W').astype(int) )
    sme_main['month_name'] = sme_main['month'].apply(lambda x: calendar.month_name[x])
    sme_main['driver_fee'] = sme_main['order_delivery_fees']*0.7
    sme_main['halan_return'] = sme_main['order_delivery_fees']*0.3
    sme_main['sme_return'] = sme_main['order_value'] - sme_main['order_delivery_fees']
    return sme_main
```

Figure 4.11 Function that imports the data from the PostgreSQL database for analysis and visualization

The flask app will act as our server and will provide all the user functionality and the dashboard system's backend logic.

```python
app = Flask(__name__)
app.config['SECRET_KEY'] = [redacted]
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///site.db'
db = SQLAlchemy(app)
bcrypt = Bcrypt(app)
login_manager = LoginManager(app)
login_manager.login_view = 'login'
login_manager.login_message_category = 'info'
UPLOAD_FOLDER = 'static/csv'
app.config['UPLOAD_FOLDER'] =  UPLOAD_FOLDER

from Halanweb import routes
```

Figure 4.12 Initializing the Flask app

```
...
from Halanweb import app


if __name__ == '__main__':
    app.run(debug=True)
```

Figure 4.13 Script to run the flask app's server. It defaults to the device's localhost with port=5000

The user interface is designed using HTML, CSS, and JavaScript, and the application's conditional logic is implemented by the Jinja templating engine inside of our html files.

```
            {%if current_user.role == "Data Entry Officer"%}
            <body>
                <div class="row">
                <nav class="sidebar close" >···
                    </nav>/.sidebar.close
                    {% endif%}


     <section class="home">

    <main role="main" class='content_container'>

        <div class=" mt--5">

         {% block content %}{% endblock %}
        </div>/.mt--5
    </main>/.content_container

</section>/.home
</div>      You, 3 minutes ago • Uncommitted changes  /.row
{% endif%}
```

Figure 4.14 Example of Jinja's conditional logic in the HTML layout file.

```
    You, 1 second ago | 1 author (You)
1   {% extends "layout.html" %}
2   {% block content %}
3
4
5   <div class="row head_theme" >···
12  </nav>/.header.head_theme</div>/.row.head_theme
13
14      You, now • Uncommitted changes
15  {%if current_user.role == "Data Entry Officer"%}
16  <div class="col-xl-4 col-lg-4 col-md-6">···
56  </div>/.col-xl-4.col-lg-4.col-md-6
57
58  <div class="col-xl-8 col-lg-8 col-md-6">···
85  </div>/.col-xl-8.col-lg-8.col-md-6
86
87  </div>
88  </div>
89  {% endif %}
90
91  {% endif %}
92  {% endblock %}
93
```

Figure 4.15 shows the usage of Jinja extends function

```python
@app.route("/", methods=['GET', 'POST'])
def login():
    if current_user.is_authenticated:
        return redirect(url_for('home'))
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(email=form.email.data).first()
        if user and bcrypt.check_password_hash(user.password, form.password.data):
            login_user(user, remember=form.remember.data)
            next_page = request.args.get('next')
            return redirect(next_page) if next_page else redirect(url_for('home'))
        else:
            flash('Login Unsuccessful. Please check email and password', 'danger')
    return render_template('login2.html', title='Login', form=form)
```

Figure 4.16 Code snipped that validates the user's login information

```python
class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(20), unique=True, nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)
    image_file = db.Column(db.String(20), nullable=False, default='default.jpg')
    password = db.Column(db.String(60), nullable=False)
    role = db.Column(db.String(20), nullable=False)
    orders= db.relationship('Order',backref='author',lazy=True)
    def __repr__(self):
        return f"User('{self.username}', '{self.email}', '{self.image_file}')"
```

Figure 4.17 Class used to create the database user table using SQLAlchemy, which translate the python code to SQL queries.

id	username	email	image_file	password	role
Search column...	Search column...	Search column...	Search column...	Search column...	Search column...
1	admin	admin@halan.com	default.jpg	$2b$12$Lr8JGgzxAYk...	admin
2	Ceo	ceo@halan.com	default.jpg	$2b$12$El1OcSvfnC0...	Upper Managment
3	Op	Op@halan.com	default.jpg	$2b$12$u7GF4rGvHq...	Operations Manager
4	Ds	Ds@halan.com	default.jpg	$2b$12$RZTrmKCXBF...	Drivers Superviser
5	DEO	Deo@halan.com	default.jpg	$2b$12$C.txQcM/mD...	Data Entry Officer

Figure 4.18 Users database table view, exhibiting the different roles and the encrypted passwords

```python
def order_id_generator():
    latest = input_data()['order_id'].max()
    print(latest)
    print(datetime.today())
    w = datetime.today().isocalendar()[1]
    d = datetime.today().isocalendar()[2]//7 +1

    return f'{w}{d}{int(latest[-3:])+1}'
```

Figure 4.19 Order ID automation

```python
def new_order():
    form = orderForm()
    if request.method=="POST":
        if form.validate_on_submit():
            Order1= Order(  order_id=order_id_generator(),
                            client_contact_no=form.client_contact_no.data,
                            client_city=form.client_city.data,
                            client_address=form.client_address.data,
                            sme_name=form.sme_name.data,
                            order_status=form.order_status.data,
                            order_reason_of_failure=form.order_reason_of_failure.data,
                            driver_name=form.driver_name.data,
                            order_delivery_fees=form.order_delivery_fees.data,
                            order_value=form.order_value.data,
                            order_date=form.order_date.data ,
                            author=current_user
                        )
            db.session.add(Order1)
            db.session.commit()
            flash('order has been added','success')
            return redirect(url_for('new_order'))
        else:
            print(form.errors)
            flash('Order adding failed','danger')
            redirect(url_for('new_order'))
        return render_template('create_order.html',title="New order",form=form)
    else:
        return render_template('create_order.html', title='Data Entry', form=form,legend='New Order')
```

Figure 4.20 Function to create new order records in the database

```python
class orderForm(FlaskForm):
    client_contact_no= StringField('client_contact_no',validators=[DataRequired()])
    order_id= StringField('order_id',validators=[DataRequired(),Length(7)])
    client_city= SelectField('client_city',validators=[DataRequired()],choices=['','Khartoum','Bahri','Omdurman'])
    client_address=StringField('client_address',validators=[DataRequired(),Length(min=5,max=150)])
    sme_name=SelectField('sme_name',validators=[DataRequired()],choices=sme_name_list)
    order_status=  SelectField('order_status',choices=['','Distibuted','Delivered','Hold','Cancelled'])
    order_reason_of_failure= SelectField('order_reason_of_failure',choices=['','Requested to recieve another day','Wrc
    driver_name= SelectField('driver_name',validators=[DataRequired()],choices=Drivers_names_list)
    order_delivery_fees= FloatField('order_delivery_fees',validators=[DataRequired()])
    order_value= FloatField('order_value',validators=[DataRequired()])
    order_date= DateField('order_date')
    submit = SubmitField('Submit')

    def validate_order_id(self, order_id):
        order = Order.query.filter_by(order_id=order_id.data).first()
        if order:
            raise ValidationError('That order ID already exists.')
```

Figure 4.21 Order Form class that validates inputs made by the users

```python
@app.route("/orders/csv",methods=['GET', 'POST'])
@login_required
def save_csv():
    uploaded_file = request.files['file']
    if uploaded_file.filename != '':
        file_path = os.path.join(app.root_path, 'static/csv', uploaded_file.filename)
          # set the file path
        uploaded_file.save(file_path)
          # save the file
        user_id = 1
        parseCSV(file_path,user_id)
    return redirect(url_for('new_order'))
@app.route("/orders/<int:order_id>")
@login_required
def order(order_id):
    order= Order.query.get_or_404(order_id)
    return render_template('order.html',title=order_id, order=order )
```

Figure 4.22 Function to upload archived data to the database from .csv files

```python
elif current_user.role == "Data Entry Officer" or current_user.role == "Drivers Superviser":
    page= request.args.get("page",1,type=int)
    orders= Order.query.order_by(Order.order_date.desc()).paginate(page=page, per_page=5 )

    graphJSON1 =[]
    graph_title=[]
    vals = []
    graph = report_plots.plots_gmv(data.halan_dataframes(sme_main,'GMV'
    ,"Recieved",'Monthly','Halan'),'Monthly')
    graph_title.append(graph_description['Order Status Count'])
    graphJSON1.append(json.dumps(graph, cls=plotly.utils.PlotlyJSONEncoder))
    vals.append(graph['data'][0]['text'])

    graph = report_plots.plots_gmv(data.halan_dataframes(sme_main,'GMV',
    "Recieved",'Weekly','Halan'),'Weekly')
    graphJSON1.append(json.dumps(graph, cls=plotly.utils.PlotlyJSONEncoder))
    graph_title.append(graph_description['Order Status Count'])
    vals.append(graph['data'][0]['text'])
    return render_template('home.html', orders=orders,graphJSON1 = graphJSON1 ,
    graph_title=graph_title,vals=vals)       You, now • Uncommitted changes
```

Figure 4.23 Code-snippet to view most recent orders and relevant graphics

'

```
from werkzeug.middleware.dispatcher import DispatcherMiddleware

from dash_application.dash_template import create_dash_app_tabbed

from Halanweb import app


dash_app1 = create_dash_app_tabbed(app,'/dashboard/')
#linking the dash app to the flask app's server
app1 = DispatcherMiddleware(app, {
    '/dashboard': dash_app1.server,
})

import dash_application.callbacks
```

Figure 4.24 Initializing the dash app

```
{% extends "layout.html" %}
{% block content %}


    <iframe class="col" src="http://localhost:5000/dashboard/"
     width=1500 height=2000 ></iframe>/.col


{% endblock %}
```

Figure 4.25 Embedding the dashboard onto the website using iframes and jinja

```
def create_dash_app_tabbed(flask_app,base_pathname):
    dash_app = Dash(__name__, server = flask_app, url_base_pathname=base_pathname,
    external_stylesheets = ['assets/bootstrap.min.css','assets/bWLwgP.css'],

    meta_tags=[ {'name': 'viewport',
    'content': 'width=device-width, initial-scale=2.0, maximum-scale=1.2, minimum-scale=0.5,'}],
    )
    dash_app.title = "Dashboard"
    dash_app.layout = html.Div(
    children=[
            tabs_component(),

    ],style=CONTENT_STYLE)

    return dash_app
```

Figure 4.26 Function to create an instance of the dash app

```
def tabs_component():
    return html.Div([
                    dcc.Tabs(id="tabs", value='dashboard',
                    persistence=True,persistence_type='session' ,children=[
                    dcc.Tab(label='Dashboard', value='dashboard'),
                    dcc.Tab(label='Report', value='report'),
                    dcc.Tab(label='Data Table', value='data_table'),
                    ]),
                    html.Div(id='tabs-content', children=[]),
                ],)
```

Figure 4.27 Defining the different tabs in the dashboard

```
@dash_app1.callback(Output('tabs-content', 'children'),
              [Input('tabs', 'value')])
def render_content(tab):
    return container_component_layout(tab,sme_main)
        You, 1 hour ago • Uncommitted changes
@dash_app1.callback(Output('table_container', 'children'),
                [Input('graph_type','value'),
          Input('order_status', 'value'),
           Input('order_period','value'),
            Input('stakeholder','value')])
def update_table(graph_type,order_status,order_period,stakeholder):
    df = data.halan_dataframes(sme_main,graph_type,order_status,order_period,stakeholder)
    return create_data_table(df)
```

Figure 4.28 Assigning dashboard components to their respective tabs

```
class data:

    def halan_dataframes(sme_main,graph_type,order_status,order_period,stakeholder):
        if graph_type == 'order_status_count':
        #
            if order_status =="Received":
            ##
                if order_period =="Daily":
                ###
                    if stakeholder =='Halan':      You, 2 months ago • merged …
                    ####
                        halan_df = pd.DataFrame(sme_main.groupby(features_order_period[order_period
                        return halan_df
                    elif stakeholder =='Driver':
                        orders_recieved_daily_driver = pd.DataFrame(sme_main.groupby([features_orde

                        return orders_recieved_daily_driver
                    elif stakeholder =='SME':
                        sme_df = pd.DataFrame(sme_main.groupby([features_order_period[order_period].
                        return sme_df
                ######################
```

Figure 4.29 Example of the data class, which processes the data according to the KPIs provided by the user.

```
@dash_app1.callback(
        Output('feature_graphic', 'figure'),
        [Input('graph_type', 'value')],
         [Input('order_status', 'value')],
          [Input('order_period', 'value')],
           [Input('stakeholder', 'value')],    )
def update_state(graph_type,order_status,order_period,stakeholder):
    if graph_type == "GMV":
       my_plot = report_plots.plots_gmv(data.halan_dataframes(sme_main,graph_type,
       order_status,order_period,stakeholder),order_period)
    elif stakeholder =='Halan' and graph_type=="order_reason_of_failure":
        my_plot =report_plots.plots_traces(data.halan_dataframes(sme_main,graph_type,
        order_status,order_period,stakeholder),stakeholder,order_period)
    elif stakeholder =='Halan' and graph_type!="order_status_percentage":
       my_plot =report_plots.plots(data.halan_dataframes(sme_main,graph_type,
       order_status,order_period,stakeholder),order_period)
    elif stakeholder == 'Halan' and graph_type == 'order_status_percentage':
        my_plot =  report_plots.plots_percentage(data.halan_dataframes(sme_main,graph_type,
        order_status,order_period,stakeholder),order_period,order_status)
    elif stakeholder != 'halan' and graph_type == 'order_status_percentage':
        my_plot = report_plots.plots_percentage_traces(data.halan_dataframes(sme_main,graph_type,
        order_status,order_period,stakeholder),stakeholder,order_period)
    else:
        my_plot =report_plots.plots_traces(data.halan_dataframes(sme_main,graph_type,order_status,
        order_period,stakeholder),stakeholder,order_period)
    return my_plot
```

Figure 4.30 Filter the graphs according to the KPIs provided by the users and returning a graph object to be viewed on the dashboard

```
def gmv_plot(sme_graph,order_period):
    sme_graph = sme_graph.reset_index()
    n = len(sme_graph[features_order_period[order_period]].unique())
    fig = px.bar(sme_graph, x=['sme_return','driver_fee','halan_return'],
     y=features_order_period[order_period], title='{} there are {} '.format(order_period,n),
    text_auto=True,
    orientation='h',template = 'plotly_dark')
    fig.update_layout(showlegend=True,   margin=dict(t=80,l=50,b=40,r=50),height=100*(n)+60,
                      title='{} Gross Market Value '.format(order_period),
                      xaxis_title=sme_columns[features_order_period[order_period]],
                      yaxis_title="Stakeholder Returns",
                      autosize = True,
                      margin_pad = 10,
                      plot_bgcolor='#191C1F',
                      paper_bgcolor ='#191C1F',
                      font = dict(family="SF Pro Display, Roboto, Droid Sans, Arial"),
                      legend=dict(
            title=None, orientation="v", yanchor="bottom",   xanchor="right")                )
    fig.update_traces(textposition='auto')
    fig.update_yaxes(automargin=True,title_font = dict(size=12), color="#707070",
     title_font_color = "#707070", tickfont = dict(size = 9), gridcolor='#242424', zerolinecolor = '#242424',)
    fig.update_xaxes(automargin=True, zeroline = True,  color="#FFF",
     title_font_color = "#707070", tickfont = dict(size = 11))
    newnames = {'sme_return':'Stores Return', 'driver_fee': 'Drivers Return','halan_return': "Halan's Margin"}
    fig.for_each_trace(lambda t: t.update(name = newnames[t.name],
                                          legendgroup = newnames[t.name],
                                          hovertemplate = t.hovertemplate.replace(t.name, newnames[t.name])))
    return  fig
```

Figure 4.31 Function to plot the GMV

4.5. Overall View of the system

We have elected to display all the parts of the system, although different users have different privileges and access levels in order to achieve a layer of security.

Starting off the user will be required to login using the preassigned login information. Only the admin has authorization to create new users.

Figure 4.32 Halan Web App Login Page

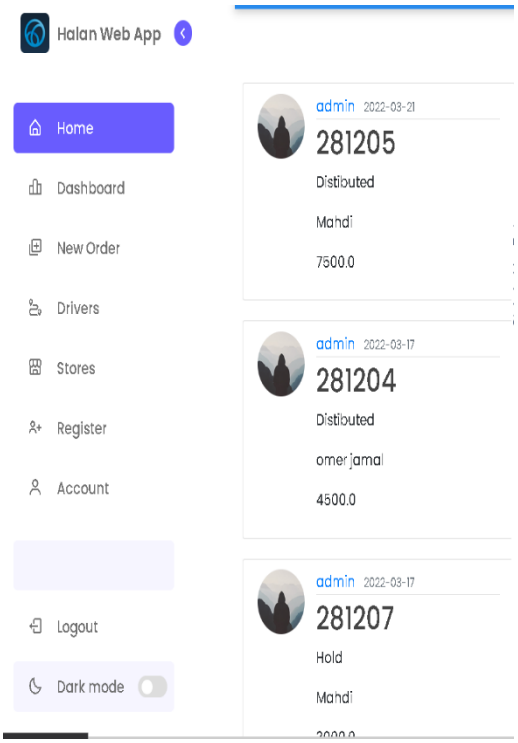


Figure 4.33 Sidebar navigator and recent orders list

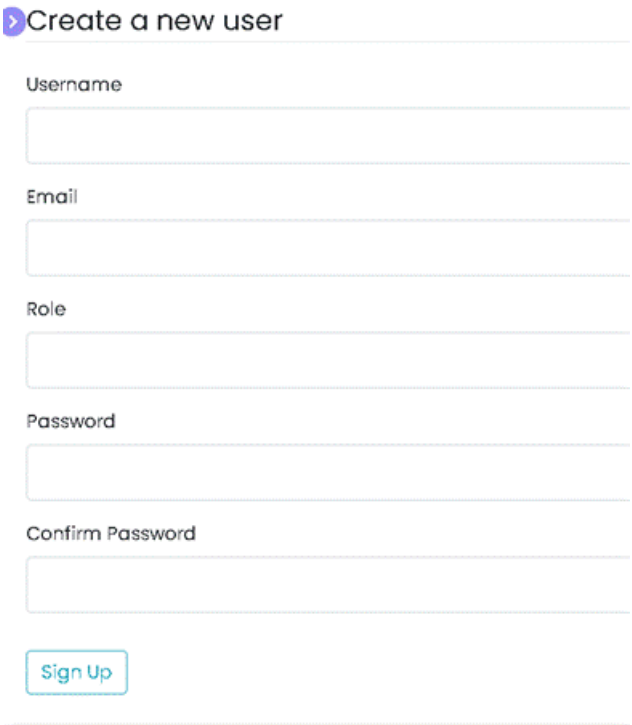


Figure 4.34 Create New User Account page. Only Admin can access it.

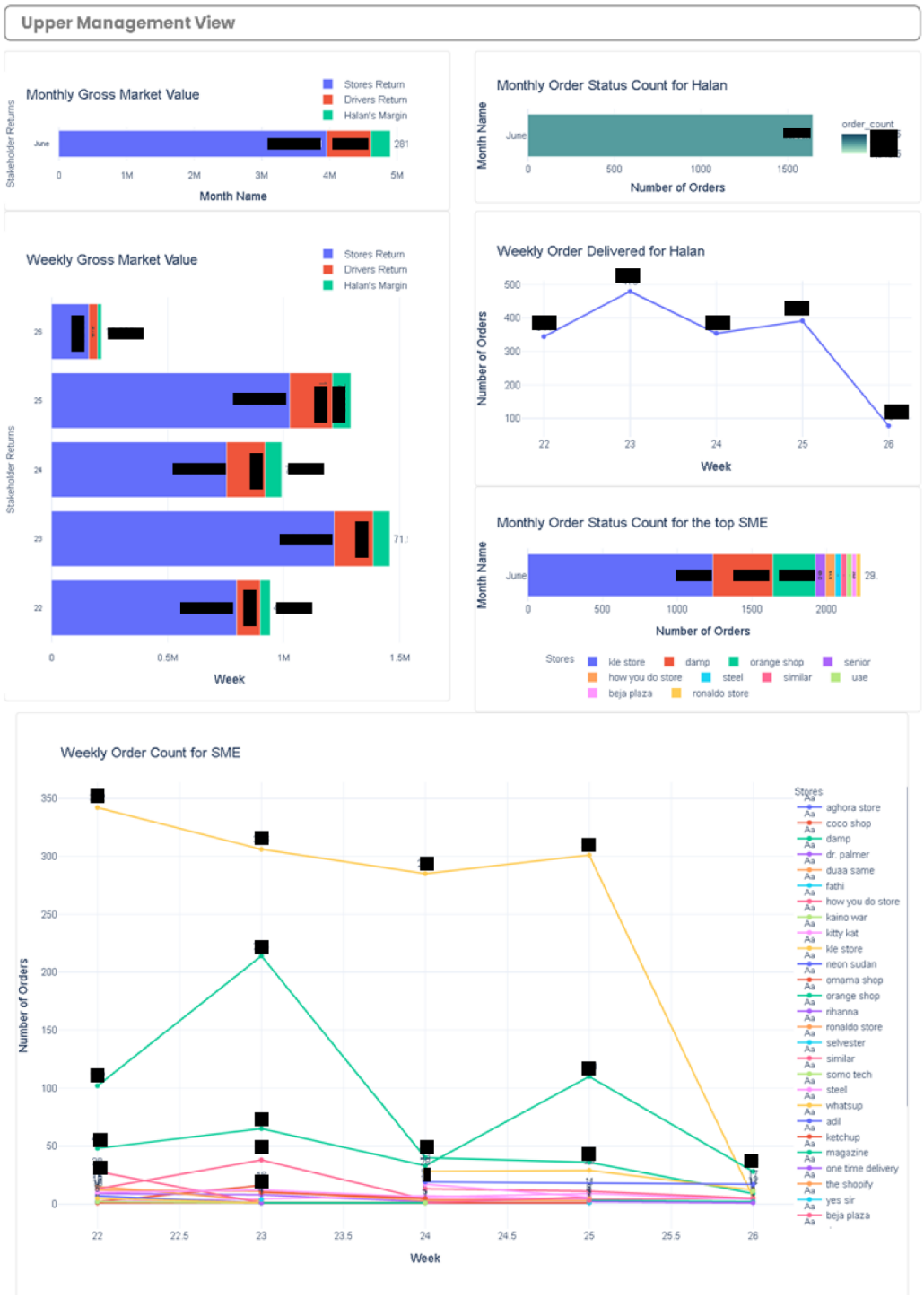


Figure 4.35 Upper Management View of data summary plots

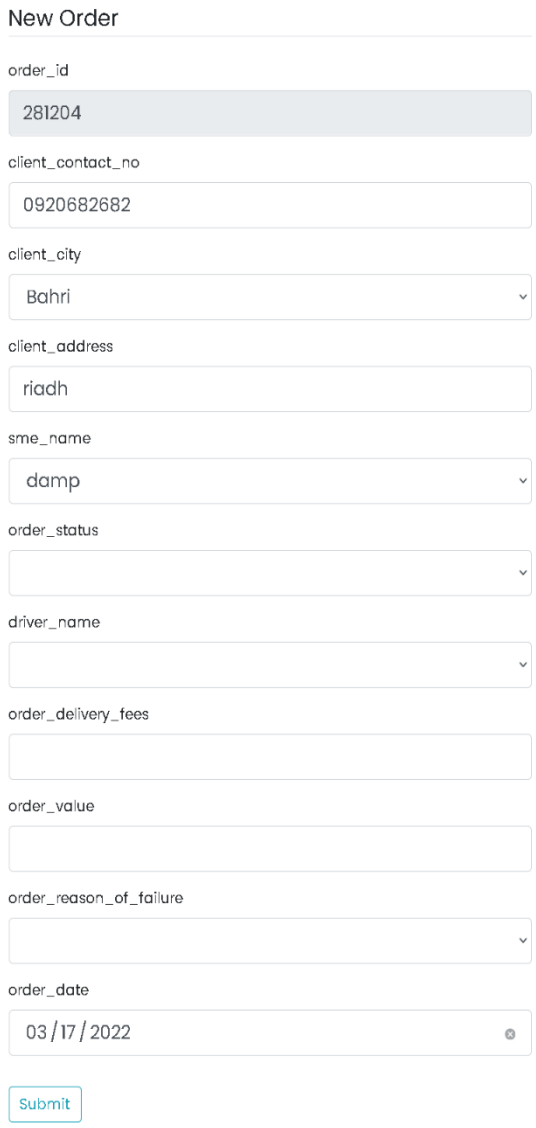


Figure 4.36 New order Data Entry Page. The Data Entry Officer can input new order into the Database

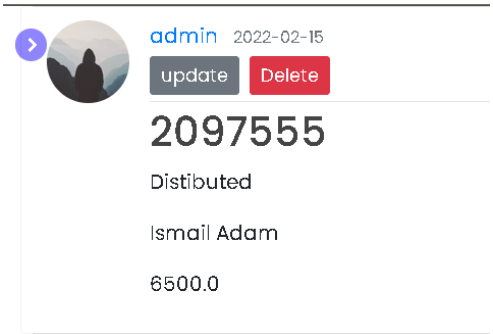


Figure 4.37 Update Order card

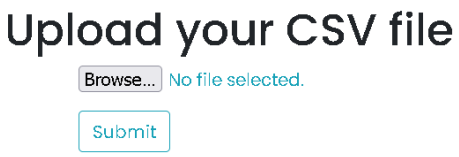


Figure 4.38 Upload CSV Button

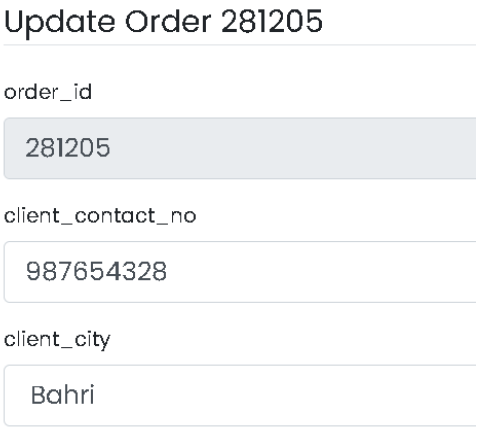


Figure 4.39 Update orders in the database. Primary key order_id cannot be modified

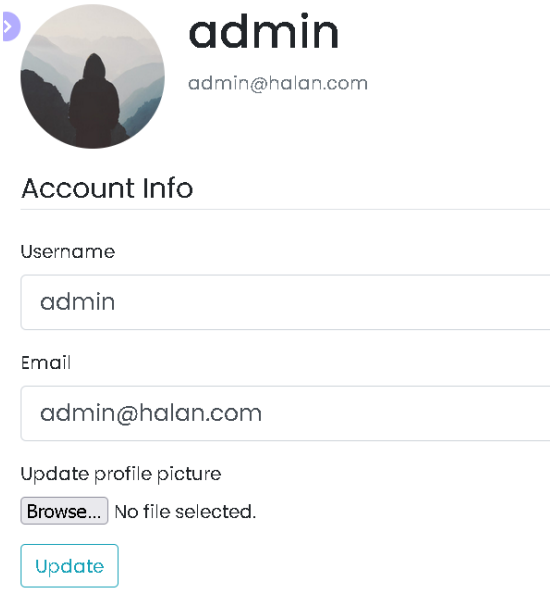


Figure 4.40 User Account Page

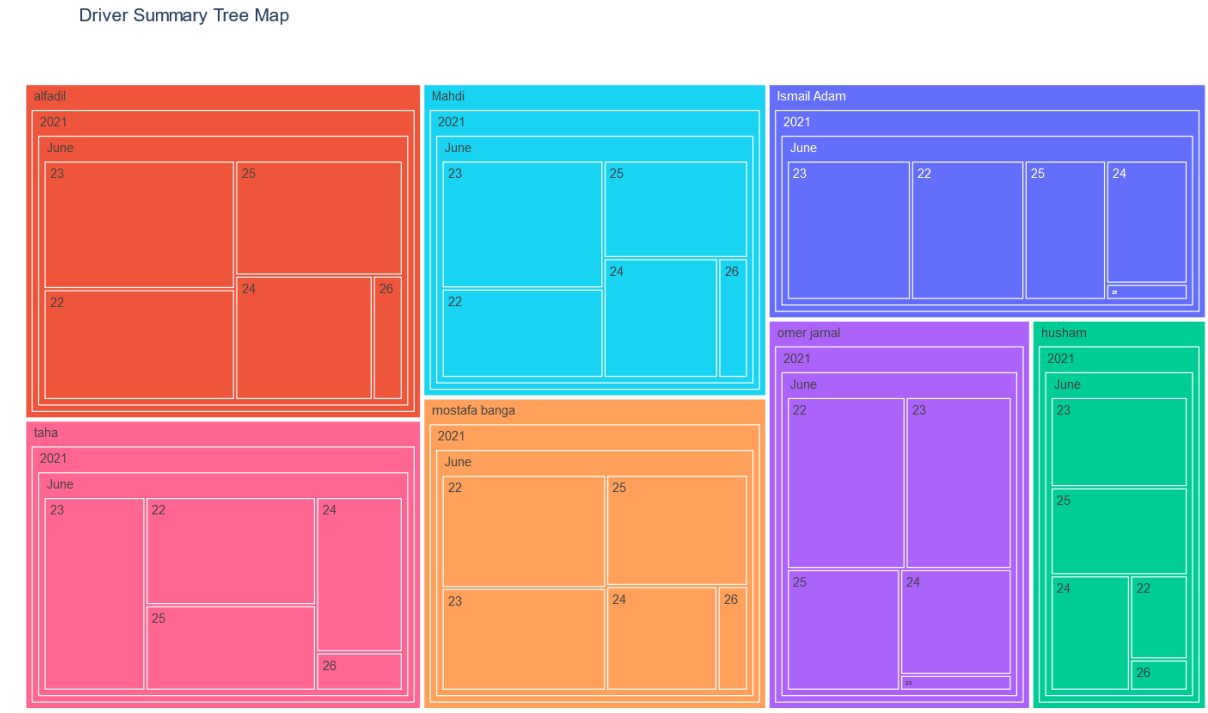


Figure 4.41 Driver Summary Tree Map

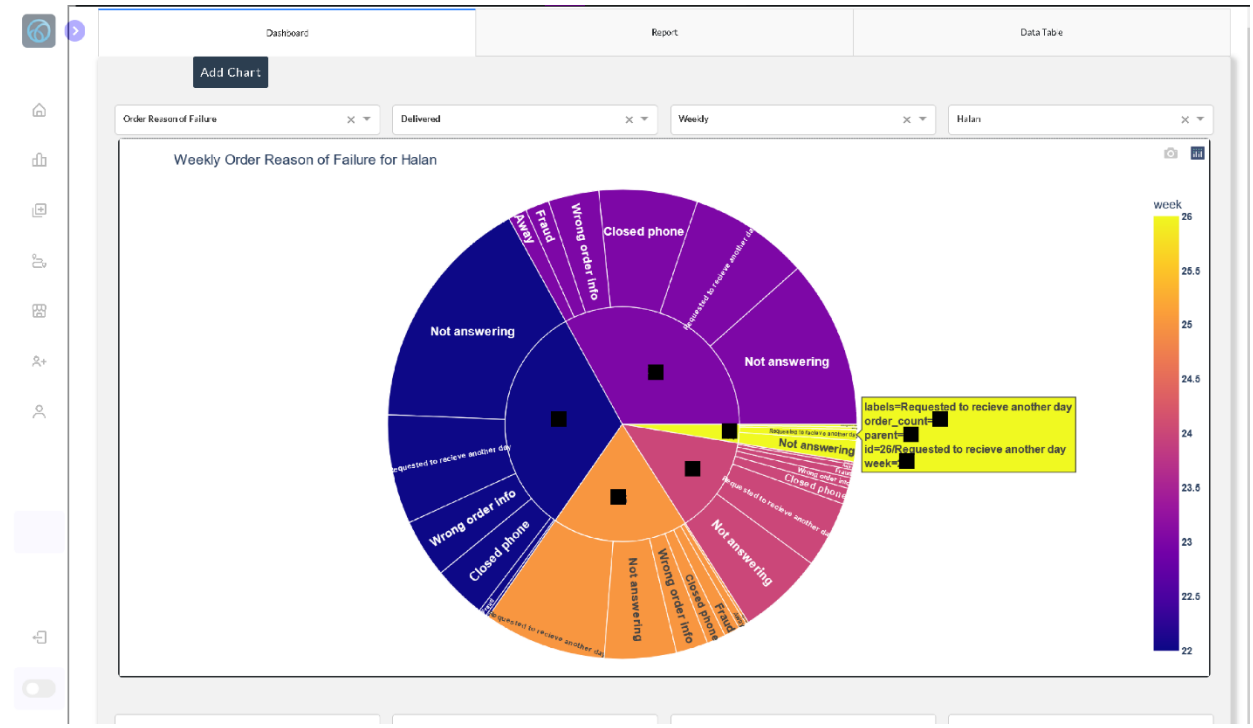


Figure 4.42 (a) Sunburst Plot High level view

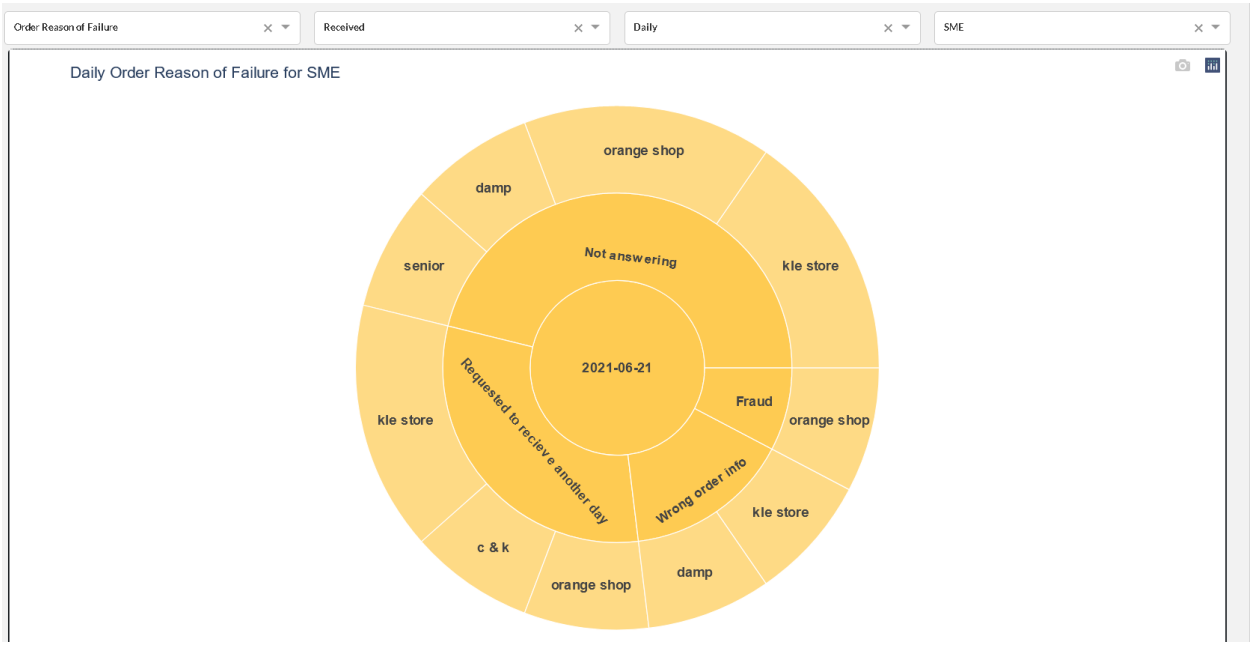


Figure 4.42 (b) Sunburst Plot Low level view

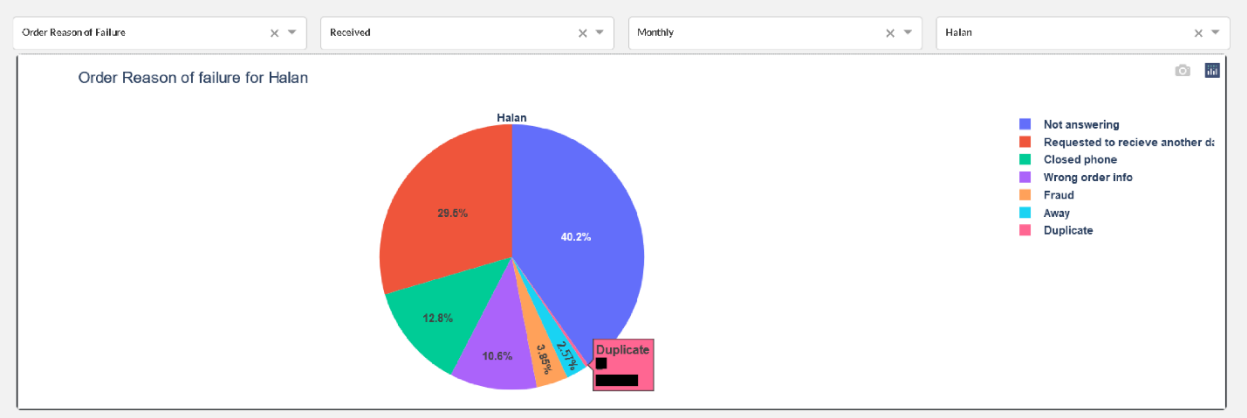


Figure 4.43 Pie Chart

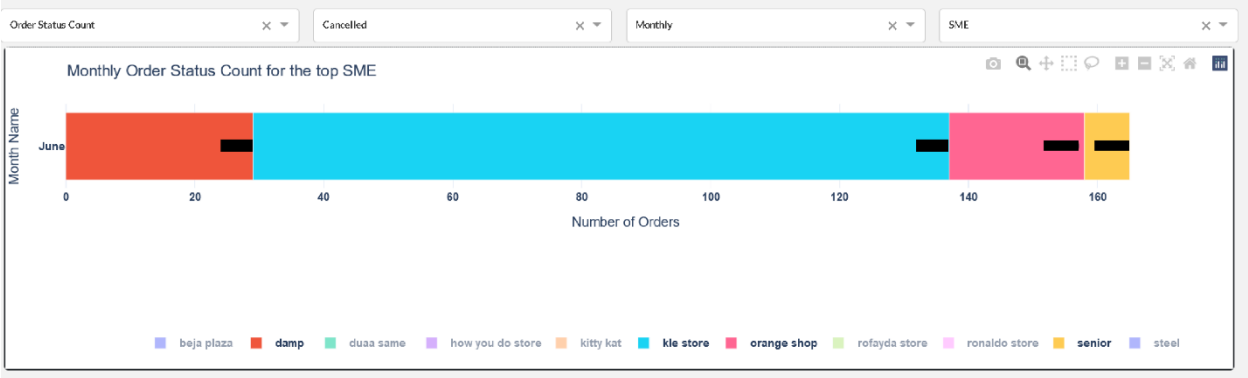


Figure 4.44 Bar Plot depicting the ability to focus a graph on a subset of features such as Store names in this example

Figure 4.45 Bar plot

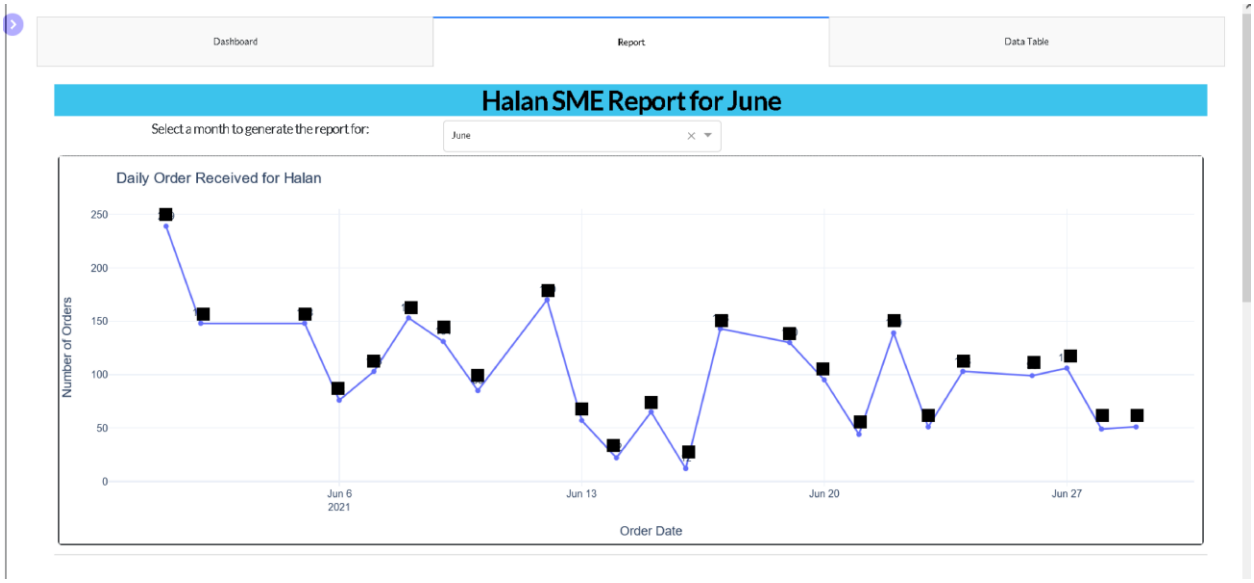
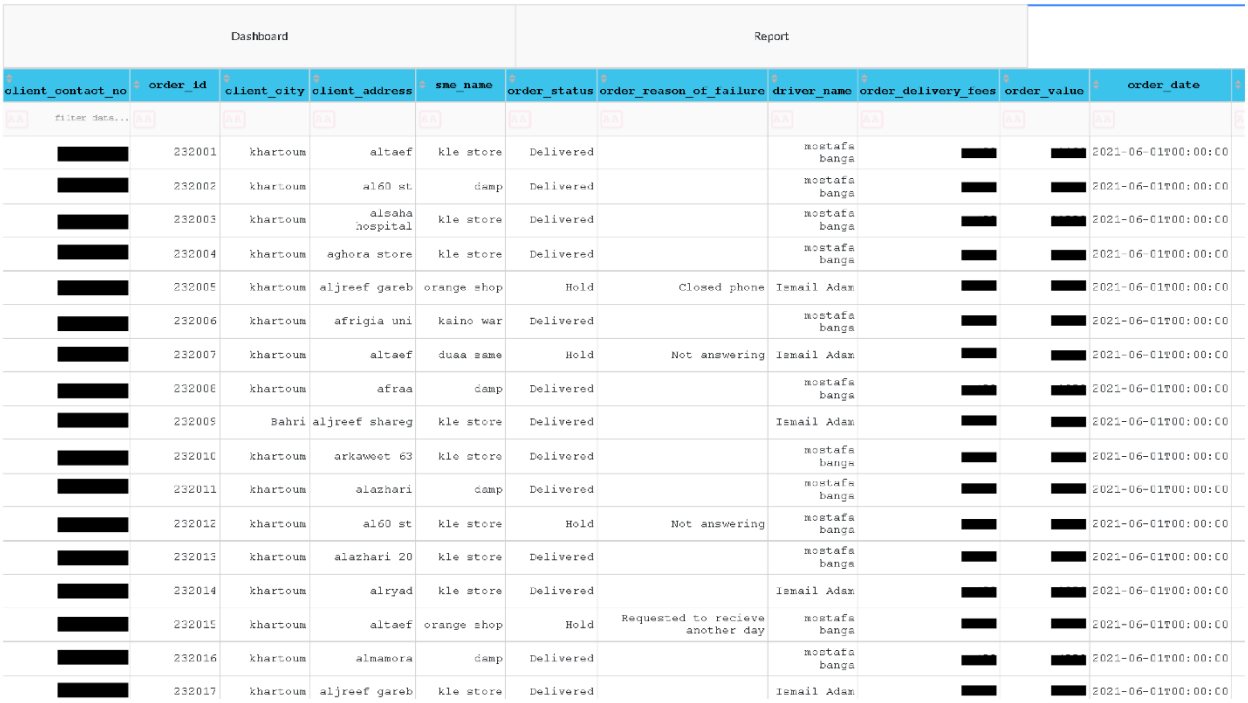


Figure 4.46 Line graph



client_contact_no	order_id	client_city	client_address	sme_name	order_status	order_reason_of_failure	driver_name	order_delivery_fees	order_value	order_date
filter data...										
[redacted]	232001	khartoum	altaef	kle store	Delivered		mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232002	khartoum	al60 st	damp	Delivered		mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232003	khartoum	alsaha hospital	kle store	Delivered		mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232004	khartoum	aghora store	kle store	Delivered		mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232005	khartoum	aljreef gareb	orange shop	Hold	Closed phone	Ismail Adam	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232006	khartoum	afrigia uni	kaino war	Delivered		mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232007	khartoum	altaef	duaa same	Hold	Not answering	Ismail Adam	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232008	khartoum	afraa	damp	Delivered		mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232009	Bahri	aljreef shareg	kle store	Delivered		Ismail Adam	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232010	khartoum	arkaweet 63	kle store	Delivered		mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232011	khartoum	alazhari	damp	Delivered		mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232012	khartoum	al60 st	kle store	Hold	Not answering	mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232013	khartoum	alazhari 20	kle store	Delivered		mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232014	khartoum	alryad	kle store	Delivered		Ismail Adam	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232015	khartoum	altaef	orange shop	Hold	Requested to recieve another day	mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232016	khartoum	almamora	damp	Delivered		mostafa banga	[redacted]	[redacted]	2021-06-01T00:00:00
[redacted]	232017	khartoum	aljreef gareb	kle store	Delivered		Ismail Adam	[redacted]	[redacted]	2021-06-01T00:00:00

Figure 4.47 Data Table

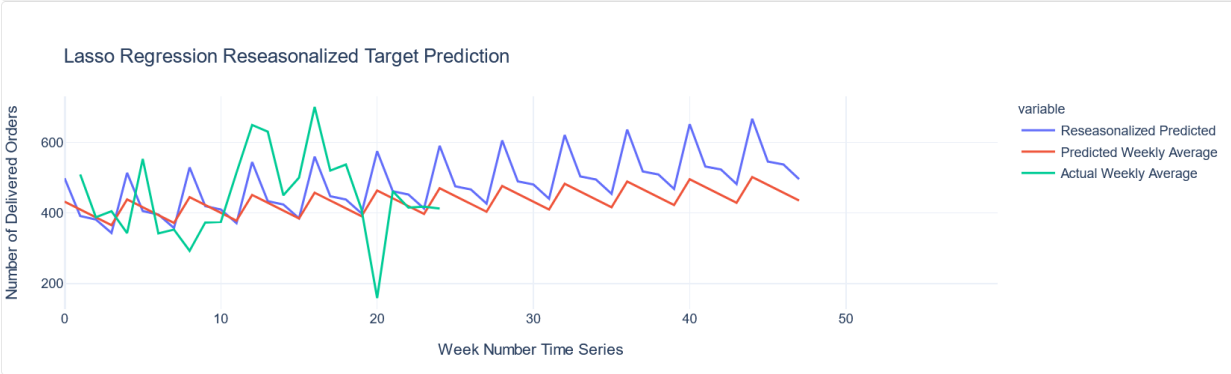


Figure 4.48 The Lasso Regression model's prediction vs the original data

4.2. Testing

In this section we discuss the testing strategy and its implementation on our system through functionality testing by applying a black box testing method and then implementing a non-functional test in which we assess the stakeholders' interaction with the system to evaluate its usability. We finally test the developed dashboard by going through a comprehensive evaluation matrix for dashboard software features.

4.2.1. Functionality Test

Table 4.2 Functionality test results

Function	Relevant Stakeholders	Level of satisfaction	Description
Provide GMV Summary	• CEO • General Manager	High	The Gross Merchandise Value can be viewed on a Daily, Weekly, and Monthly basis in visually pleasing graphs. Highlights the distribution of shares of the GMV of Stores, Drivers, and the Organization.
Estimation of number of orders per store vs actual	• General Manager • Operation Manager	Low	The actual number of orders is determined but no prediction for each individual store was made due to low volume of data. Instead, a prediction for all orders delivered was made.
Induvial stores Delivery percentages.	• General Manager • Operation Manager	High	Plots comparing all the stores delivery percentage in addition to the cancellation and holding percentage are available on demand.
Determine active stores in the current month.	• Operations Manager	High	The dashboard provides a graph of stores that have been active each month when said month is selected.

Calculate the driver's delivery success rate to monitor for fraud.	• Operation Manager • Drivers Supervisor	High	Multiple plots have been designed to both quantify and assess the level of competence and fraud.
Driver Allocation Suggestion	• Data Entry Officer • Drivers Supervisor	Medium	Due to the company data lacking geographic zones for classification to be conducted, a simple drop-down list with conditional statements that's suggests based on city and not exact locations is provided as a substitute to manually select the driver.
View and export Monthly reports	• CEO • General Manager • Operations Manager	High	All KPIs are summarized in the report and the relevant month can be selected and report can be printed using the browser's print function.
Data Entry and Updating Records in the Database	• Data Entry Officer • Accountant	High	A number of forms were created to provide an efficient and robust data entry process.

4.2.3 Non-functionality test

Table 4.3 Non-Functionality test results

Non-Functional Requirement	Satisfied or not	Description
Security Features	✓	The high level of security was achieved by implementing a session-based login system with different authorization levels based on the user's role.
Portability	✓	Having the system in web-application form allows it to be platform dependent and thus achieving the portability goal.
Robustness	✓	Users are not able to input invalid data. Users are presented with an error message and are prompted to fix it.
Usability	✓	The average users are able to easily interact and navigate most of the system's functions after a short training session.
Responsiveness	✓	The web application works well with different screen sizes and resolutions
Reliability	✓	The system has minimal lag and delivers the required outcomes.

4.2.4. Dashboard Design Testing

A. End-User Experience

I. Graphical User Interface (GUI)

The dashboard software developed provides an intuitive and user-friendly interface for all levels of users. Users with different technical skills are able to use the web application to input and update data and interpret the provided visualizations.

II. Web-based

The dashboard system fully supports web-based interface. It can be accessed through the internet once the server is set up. Its features are supported in the leading web browsers such as Mozilla Firefox and Chromium based browsers (Google Chrome, Microsoft Edge, etc.).

III. Performance

The response time for loading the dashboard is acceptable to users. It is hard to properly gauge the performance in the development environment, thus a reassessment at production time must be made. The benchmarks for the test will be to assess the performance during peak traffic with regards to the supporting hardware and network capacity.

IV. Multilingual Support

This is a nice to have feature, but due to the limited development time the developers have opted out of implementing it. The only supported language is English.

B. User Management

I. Personalization Framework

The dashboard software provides template driven personalization framework, in accordance with the user's role. It provides set views and access levels for each group of users. It does not allow users to customize their mains views, although there is room for manipulation in the dashboard view, through selection of desired graphs. As the number of different roles is limited, it is more convenient to have a template driven as opposed to user driven personalization framework.

II. User Privilege Framework

The system has administrative components to facilitate management of user roles and access levels. The different roles and restrictions are evaluated based on the organization's specific requirements and employee hierarchy.

III. Dashboard Grouping

In large organizations, the dashboard deployment would require dashboard grouping to eliminate the repetitive task of assigning individual dashboards to each user group. The developed system

will be used by a handful of users and so there is minimal dashboard grouping in the form of different views for different KPIs that each user is concerned with, such as in figure 4.34.

IV. Metrics Grouping

In security management, a group of users' needs to be denied access to certain metrics. We have grouped KPIs according to the access levels and allow users to only view pages that are relevant to them and prevent access to the rest of the system.

C. Drill-Down

I. Context

Drill-down helps the user perform self-guided analysis. By clicking on visual indicators on the dashboard, they are led to a more detailed level of information that can fully explain the visual indicator. The dashboard provides partial context link drill-down in some of its graphics, where the user can click on a section of the plot and it will expand to provide a more detailed summary, such as in figures 4.41 and 4.42. They also have selectors that can provide analysis of the data on a monthly, weekly, and daily basis. This helps them view exactly when and where a sudden change or irregularity had occurred.

II. Multilevel Drill-Down

The dashboard facilitates multiple level of drill-down through tree maps and sunburst plots, allowing for powerful self-guided analysis through the experience. The plots allow users to click on sections to expand and view sub-levels to gain more detailed information than from the higher levels. In addition to that, they can zoom in on sections of graphs.

III. Retracing Drill-Down Path

The multilevel drill-down featured plots allow the users to retrace their steps and view the previous levels with a simple click and the sequencing is intuitive.

D. Reporting

I. Sorting and Filtering

Users are able to filter the data features by inputting their filtering criterion in the data tables and using a dropdown menu for the visualizations to filter according to time period. The can also filter out specific drivers and stores by omitting them from the graph's legend as show in figure 4.44.

II. Online analytical processing (OLAP) features

The dashboard system implicitly provides limited OLAP features in accordance with the user requirements such as sum and percentage of orders but does not provide methods for obtaining statistics such as sum, maximum, minimum, average, count, and percentage explicitly. It does provide static report groupings and users can select which month to produce the report on

and each visualization may be further customized by selecting features to show, according to the predetermined conditions.

III. Snapshot Capture

The reporting framework allows the user to save the reported data by clicking on a “save png” camera button on the top right corner of each plot. Furthermore, the whole report can be printed and saved virtually by simply right clicking and saving from the menu.

E. Data Connectivity

I. Multiple Data Source Connectivity

The dashboard is not impeded by the need to pull data from disparate sources due to the organization’s size and the nature of its operations. The system obtains and stores its data in a single database.

II. Real-Time Connectivity

The dashboard is able to pull data live from its database any changes made to the database will reflect on the visualizations.

F. Visualization

I. Visual Intelligence

The chosen Plotly visualization library offers the user a high level of interactivity through intelligent graphics. Users are able to hover over data points and fields to obtain more information, zoom in and out, select features to show, and navigate lower levels of certain plots. This thus provides them with higher levels of insight to make informed decisions.

II. Screen Resolution

The web application is optimized for different screen resolutions. All pages have adaptive reflexive content that resizes according to the page, although the graphs might seem cluttered at extremely low resolutions.

G. System Requirements

I. Operating system

The system, being built in Python, can run on any of the main operating systems.

II. Application server

The organization will be using Oracle Cloud Infrastructure for the deploy the dashboard web application, which will easily integrate.

III. Browser Support

The web application dashboard is supported by the popular browsers as mentioned previously, but due to the interactivity and graphic nature of the dashboard, low end browsers such as Internet Explorer cannot support those features. It can be viewed on both desktop and mobile environments.

Table 4.4 Dashboard Evaluation Matrix

Features	Weight 1-10	Score 1-5	Weighted Score
End-User Experience			
GUI	8	4	32
Web-based	7	5	35
Performance	9	4	36
Multilingual Support	2	0	0
User Management			
Personalization Framework	5	2	10
User Privilege Framework	10	4	40
Dashboard Grouping	3	2	6
Metrics Grouping	3	1	3
Drill-Down			
Context	6	5	30
Multilevel drill-down	3	4	12
Retracing drill-down path	3	5	15
Reporting			
Sorting and filtering	7	3	21
Online analytical processing (OLAP) features	4	2	8
Snapshot capture	9	4	36
Data Connectivity			
Multiple data source connectivity	0	0	0
Real-time connectivity	8	4	32
Visualization			
Visual intelligence	9	4	36
Screen resolution	8	4	32
System Requirements			
Operating system	5	5	25
Application server	5	4	20
Browser Support	8	4	32
Total Weighted Score			461
Maximum Total Weighted Score			610
Score Percentage			75.6%

No dashboard software package may claim to contain all of the features discussed so far, and the evaluation matrix score of 75.6% is an indication that the dashboard performs well given the requirements of the organization.

4.3. Machine Learning Algorithm Selection Process

The goal from the implementation of ML is to provide the organization with forecast of future estimates for the number of delivered orders based on past weekly delivered orders data to determine the target that the organization should meet to assess the current week/month's performance.

The raw data obtained from the organization was low in volume, namely only 4 weeks of data was provided, which meant it cannot be used to make any reliable predictions. After a request was made to the organization, the researchers were provided with 24 weeks of summarized data, indicating the number of delivered orders per week. The organization splits the month into 4 weeks to properly track their target, so some weeks are more than 7 days.

The features of the data provided are described below:

	Data Type
Series	ID Column
Month	Numeric
Week	Numeric
Original	Label

Figure 4.49 Description of Data features

	Month	Week	Original
count	24.000000	24.00000	24.000000
mean	8.500000	2.50000	446.254980
std	1.744557	1.14208	120.141443
min	6.000000	1.00000	159.000000
25%	7.000000	1.75000	374.500000
50%	8.500000	2.50000	416.429886
75%	10.000000	3.25000	515.500000
max	11.000000	4.00000	700.000000

Figure 4.50 Description of the data Values

Where original is the original sum of orders delivered that week in that month.

Our initial strategy was to conduct a Hold-out Cross Validation test and -fold cross validation test to compare the performance of multiple machine learning algorithms on the given data and rank them by their performance scores. Using the Pycaret library in python, we fed the obtained summary data into the setup function, which we use to carry out Hold-Out and K-fold Cross Validation testing.

Table 11 shows the results of the Hold-out cross validation test on some of the ML algorithms tested, ranked by the Mean Absolute Error. The Lasso Regression Algorithm performs the best with a MAE of "121.0241".

The k-fold strategy was specified as timeseries and the k number of folds was looped from 2 to 24, due to the data only having 24 records, and there were results for $3 \leq k \leq 18$, the top scorers for each iteration, ranked by the Mean Absolute Error, outline in table-12.

Table 4.5 Cross-Validation Scores for each Model

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lasso	Lasso Regression	121.0241	23654.0352	149.7415	-0.907	0.3622	0.3063	0.0275
lightgbm	Light Gradient Boosting Machine	121.3729	23785.2388	150.4076	-0.92	0.3636	0.3067	0.0975
dummy	Dummy Regressor	121.3729	23785.239	150.4076	-0.92	0.3636	0.3067	0.015
llar	Lasso Least Angle Regression	121.3729	23785.2388	150.4076	-0.92	0.3636	0.3067	0.0275
en	Elastic Net	123.0839	24406.4297	150.8398	-0.943	0.367	0.3141	0.0275
br	Bayesian Ridge	125.9281	29801.9466	162.7369	-1.249	0.4004	0.3363	0.0525
rf	Random Forest Regressor	130.3272	31510.0012	165.7991	-1.12	0.3985	0.3818	0.305
ada	AdaBoost Regressor	131.2438	33666.0652	164.874	-1.155	0.4029	0.3955	0.06
ridge	Ridge Regression	131.6862	31259.7659	166.5581	-1.167	0.4018	0.3658	0.03
lar	Least Angle Regression	132.6029	32386.0559	167.1157	-1.21	0.4036	0.3678	0.0375
lr	Linear Regression	133.4912	32497.0962	167.9953	-1.221	0.406	0.3705	1.0625
gbr	Gradient Boosting Regressor	134.187	36831.7111	170.711	-1.294	0.4118	0.4145	0.0375
dt	Decision Tree Regressor	134.1875	36895.1875	170.8289	-1.297	0.412	0.4147	1.0425
et	Extra Trees Regressor	134.1875	36895.1875	170.8289	-1.297	0.412	0.4147	0.165
huber	Huber Regressor	143.9088	36469.7516	179.4876	-1.645	0.4388	0.3902	0.0475
omp	Orthogonal Matching Pursuit	144.1944	35111.9231	174.3894	-1.462	0.4152	0.4004	0.0275
par	Passive Aggressive Regressor	145.4354	32308.27	166.4349	-1.685	0.3965	0.3102	0.03

Table 4.6 Best performing algorithm for each K-fold value from 3-18

K-Fold	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
3 fold	Passive Aggressive Regressor	25.1819	814.5673	28.5406	-0.9988	0.0641	0.0589
8 fold	Passive Aggressive Regressor	25.1819	814.5673	28.5406	-0.9988	0.0641	0.0589
9 fold	Passive Aggressive Regressor	25.1819	814.5673	28.5406	-0.9988	0.0641	0.0589
12 fold	Lasso Regression	29.2152	964.3338	31.0537	-1.3663	0.0715	0.0697
4 fold	Lasso Regression	29.2152	964.3338	31.0537	-1.3663	0.0715	0.0697
5 fold	Lasso Regression	29.2152	964.3338	31.0537	-1.3663	0.0715	0.0697
6 fold	Lasso Regression	29.2152	964.3338	31.0537	-1.3663	0.0715	0.0697
7 fold	Lasso Regression	29.2152	964.3338	31.0537	-1.3663	0.0715	0.0697
13 fold	Least Angle Regression	29.2153	964.3349	31.0537	-1.3663	0.0715	0.0697
15 fold	Least Angle Regression	29.2153	964.3349	31.0537	-1.3663	0.0715	0.0697
14 fold	Bayesian Ridge	29.5216	1005.633	31.7117	-1.4676	0.073	0.0705
16 fold	Bayesian Ridge	29.5216	1005.633	31.7117	-1.4676	0.073	0.0705
17 fold	Bayesian Ridge	29.5216	1005.633	31.7117	-1.4676	0.073	0.0705
10 fold	Random Forest Regressor	61.5863	4295.133	65.5373	-9.5393	0.1536	0.1449
11 fold	Random Forest Regressor	61.5863	4295.133	65.5373	-9.5393	0.1536	0.1449
18 fold	Random Forest Regressor	61.5863	4295.133	65.5373	-9.5393	0.1536	0.1449

```
from pycaret.regression import *# initialize setup
top=[]
for i in range(2,24,1):
  print(i)
  s = setup(data = train, test_data = test, target = 'Original',
            fold_strategy = 'timeseries', numeric_features = ['Month', 'Week'],
            fold = i, transform_target = True, session_id = 123)
  best = compare_models(sort = 'MAE')
  top.append({i:best})
for i in range(len(top)):
  print(top[i])
```

Figure 4.51 Code snippet for the K-fold test

```
final_best = finalize_model(best)
```

```
final_best
```

```
PowerTransformedTargetRegressor(alpha=1.0, copy_X=True, fit_intercept=True,
                                max_iter=1000, normalize=False, positive=False,
                                power_transformer_method='box-cox',
                                power_transformer_standardize=True,
                                precompute=False, random_state=123,
                                regressor=Lasso(alpha=1.0, copy_X=True,
                                                fit_intercept=True,
                                                max_iter=1000, normalize=False,
                                                positive=False,
                                                precompute=False,
                                                random_state=123,
                                                selection='cyclic', tol=0.0001,
                                                warm_start=False),
                                selection='cyclic', tol=0.0001,
                                warm_start=False)
```

Figure 4.52 Hyperparameters of the finalized model

```
import pandas as pd
import numpy as np
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Lasso

from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedKFold
```

Figure 4.53 Importing the Python Libraries to train the final model

```
# drop unnecessary columns and re-arrange
data = deseasonalized[['Series', 'Month', 'Week', 'Original']]


x = np.array(data.drop(["Series","Original"], 1))
y = np.array(data["Original"])
#xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.1, random_state=42)
model = Lasso(alpha=1.0, copy_X=True,
              fit_intercept=True,
              max_iter=1000, normalize=False,
              positive=False,
              precompute=False,
              random_state=123,
              selection='cyclic', tol=0.0001,
              warm_start=False)
model.fit(x, y)
```

Figure 4.54 Training the model in Scikit-Learn

```
predictions_future = model.predict( future_df.drop(['Series'],axis=1,))
predictions = pd.DataFrame(predictions_future, columns=['Label'])
future_df['Label']=predictions['Label']
```

Figure 4.55 Obtaining Predictions

```
concat_df = pd.concat([data,future_df], axis=0)
df2 = df.rename({'w1': 1, 'w2': 2,'w3':3,'w4':4}, axis=1)
avgs=[]
for i in range(4):
    avgs.append(df2[i+1].mean())
res = concat_df.dropna(subset=['Label'], axis=0)
res.drop(['Original'], axis=1, inplace=True)
res['reseason']= res['Label']
bias=1
lag=0
if not bias:
    bias = 1
if not lag:
    lag= 0
for i in range(len(res['reseason'])):
    res['reseason'][i]=res['reseason'][i]*(avgs[res['Week'][i]-1])*(1-lag)+bias*i
concat_res = pd.concat([res,deseasonalized], axis=0)
```

Figure 4.56 Re-seasonlization of the data.

Lag values of greater than 0 will flatten and lower the graph and vice-versa, so it affects the height and scale of the graph, while bias will control the elevation of the graph, with a larger positive value increasing the graph's gradient.

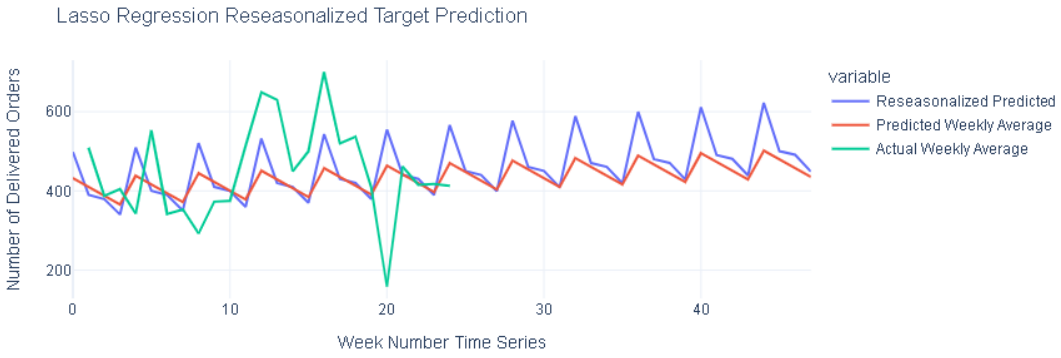


Figure 4.57 Plot of Original Data (green) vs the model's prediction (red) vs the re-seasonalized prediction values (blue).

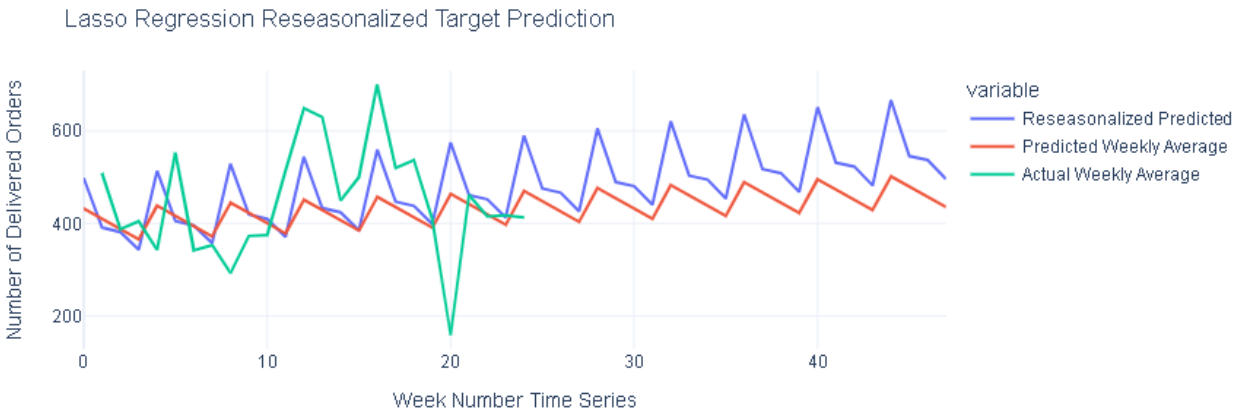


Figure 4.58 Plot to demonstrate the effect of applying a +2 bias to the re-seasonalized prediction values (blue).

Although the Lasso Regression (LR) model did not perform that well in terms of predictions scores alone as shown in tables 4.5 and 4.6, the goal is not to provide accurate predictions, it still performed better on average than the other algorithms.

Ultimately the goal was to provide a forecast for the number of orders and the LR model provides the closest shape to the original data, inferring correctly that the first week had the largest weight each month on average.

Table 4.7 Average weights for each week of the month

Week	Week weight
1	1.1533267858917444
2	0.9499417347539348
3	0.9735422564942372
4	0.9231892228600836

After relaying the results to the stakeholders, it was well received. It is recommended that the approach be revisited in the future when more data is available to obtain better results.

Chapter 5

CONCLUSION AND RECOMMENDATION

6.1. Conclusion

In this thesis the researchers proposed and developed a Performance Monitoring Dashboard to act as a Business Intelligence system for Halan's Small-to-Medium Enterprise Division. The researchers aim to provide availability of service, data integrity through forms verification and validation of each input, a level of security and confidentiality by restricting access to certain views according to the users' role. Aided with the training of the Lasso Regression Machine Learning algorithm and the extraction of Key Performance Indicators (KPIs) from the company data using Python's data analysis libraries, the performance monitoring dashboard enhances data-driven decision making by providing users with interactive visualizations and seasonal reports.

The developed system is able to query data from the database, visualize the data that the stakeholder requested, both in tabular and graphical form. The dashboard is able to output daily, weekly, and monthly reports to track the performance of drivers, individual small businesses, and the SME branch.

6.2. Recommendation

For whoever is interested in extending the scope of this project, we recommend exploring the development of a QR mobile based application to facilitate the inventory management of the delivery orders. Another relevant direction could be adding a client zone label aid in designing a tool that can recommend optimal routes for the drivers delivering the goods.

Access to the dash interface is not completely secure as the free version of Dash API isn't integrated with the flask authentication model. In the future, an alternative security method could be implemented, or future developers could consider purchasing a Dash Enterprise license.

The data provided by Halan was not enough to properly train the model efficiently. Machine Learning requires high volumes of data to be considered effective. We recommend reevaluation the suitable machine learning model when more data is available.

Furthermore, future researchers may make every plot or graph even more modifiable by including specific feature selectors and modifiers. They may also work on making a button to print out the report instead of using the browser's inbuilt features.

Chapter 6

BIBLIOGRAPHY

Ahmed, E. (2021). Utilization of Business Intelligence Tools among Business Intelligence Users. *International Journal for Innovation Education and Research*, *9*(6), 237–253. https://doi.org/10.31686/ijier.vol9.iss6.3172

Anscombe, F. J. (1973). Graphs in statistical analysis. *American Statistician*, *27*(1), 17–21. https://doi.org/10.1080/00031305.1973.10478966

Braschler, M., Stadelmann, T., & Kurt, S. (2019). *Applied Data Science: Lessons Learned for the Data-Driven Business*. Springer. https://doi.org/https://doi.org/10.1007/978-3-030-11821-1

Cairo, A. (2016). The Truthful Art: Data, Charts and Maps for Communication. In *InfoDesign - Revista Brasileira de Design da Informação* (Vol. 14, Issue 3, pp. 397–403). https://doi.org/10.51358/id.v14i3.561

Castillo, D. (2021). *Develop Data Visualization Interfaces in Python With Dash*. https://realpython.com/python-dash/

DataCamp Team. (2020). *Choosing Python or R for Data Analysis? An Infographic*. https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis

Demirkan, H. and Dal, B. (2014). Why Do So Many Analytics Projects Still Fail? Key considerations for deep analytics on big data, learning and insights. *INFORMS Analytics*, 44–52.

Destiandi, N., & Hermawan, A. (2018). Business Intelligent Method For Academic Dashboard. *Bit-Tech*, *1*(2), 11–20. https://doi.org/10.32877/bt.v1i2.42

Dinesh, D. (2021). *Business intelligence*. *February*.

Dodge, Y. (2006). *The Oxford Dictionary of Statistical Terms*. Yadolah Dodge.

Downey, A. B. (2015). Think Stats. In *O'Reilly* (2nd ed.).

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. In *O'Reilly*.

Godsey, B. (2017). Think Like a Data Scientist. In *Manning*.

Haider, M. (2016). *Getting Started with Data Science: Making Sense of Data with Analytics*. BM Press: Pearson plc.

Healy, K. (2019). Data Visualization: A Practical Introduction. In *IEEE Transactions on Professional Communication*. https://doi.org/10.1109/TPC.2019.2922787

Hewitt, C. (1977). Cross-National Comparison THE EFFECT OF POLITICAL DEMOCRACY AND SOCIAL DEMOCRACY ON EQUALITY IN INDUSTRIAL SOCIETIES : A CROSS-NATIONAL COMPARISON * One important controversy in political sociology and stratification theory con- cerns the effect of politi. *American Sociological Association*, *42*(3), 450–464.

Hurwitz, J., & Kirsch, D. (2018). *Machine Learning for Dummies*.

IBM. (2021a). *Data Mining*. Education, IBM Cloud. https://www.ibm.com/cloud/learn/data-mining

IBM. (2021b). *Monitoring FileNet P8 Platform*. IBM. https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=administering-monitoring-filenet-p8

IBM Cloud Team. (2021). *Python vs. R: What's the Difference?* https://www.ibm.com/cloud/blog/python-vs-r

International Institute of Business Analysis. (2015). *Babook A GUIDE TO THE BUSINESS ANALYSIS BODY OF KNOWLEDGE* (V3 ed.).

Jackman, R. (1980). THE IMPACT OF OUTLIERS ON INCOME INEQUALITY. *American Sociological Association*, *45*(2), 344–347.

Jupyter, P. (2021). *The Jupyter Notebook*. https://jupyter.org/

Kanbanize. (2021). *Kanban Explained for Beginners | The Complete Guide*. Kanbanize. https://kanbanize.com/kanban-resources/getting-started/what-is-kanban

Kerzner, H. (2017). *Project Management Metrics, KPIs, and Dashboards* (Third, Vol. 148). John Wiley & Sons, Inc.

Klipfolio. (2021). *Klipfolio*. Klipfolio. www.Klipfolio.com

Larson, D. (2019a). Best Practices in Accelerating the Data Science Process in Python. In *IntechOpen*. https://doi.org/http://dx.doi.org/10.5772/intechopen.84784

Larson, D. (2019b). Best Practices in Accelerating the Data Science Process in Python. *IntechOpen*.

M.B, K. (2019). *HOUSE SALE PRICES PREDICTION USING LINEAR REGRESSION* (Issue June). GITAM School of Technology.

Mathur, B., & Kaushik, M. (2016). *Data Analysis of Students Marks with Descriptive Statistics Data Analysis of Students Marks with Descriptive Statistics*. *May*.

Mccaffrey, P. (2020). *An Introduction to Healthcare Informatics*. 17–29. https://doi.org/10.1016/B978-0-12-814915-7.00002-8

Milani, F. (2019). Digital business analysis. In *Digital Business Analysis*. https://doi.org/10.1007/978-3-030-05719-0

Müller, A. C., & Guido, S. (2017). Introduction to Machine Learning with Python. In *O'Reilly*.

Neifer, T., Lawo, D., & Esau, M. (2021). Data science canvas: Evaluation of a tool to manage data science projects. *Proceedings of the Annual Hawaii International Conference on System Sciences*, *2020-Janua*(January). https://doi.org/10.24251/hicss.2021.657

NICA, I., ALEXANDRU, D., & IONESCU, Ștefan. (2021). Using of KPIs and Dashboard in the Analysis of Carrefour Company's Performance Management. *Journal of Organizational Management Studies*, *June*, 1–23. https://doi.org/10.5171/2021.852077

Parmenter, D. (2020). *Key Perfomance Indicator developing, implementing and using winning KPIs*. *pa*, 381.

Radigan, D. (2021). *How the kanban methodology applies to software development*. Atlassian. https://www.atlassian.com/agile/kanban

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 1–21. https://doi.org/10.1007/s42979-021-00592-x

Sarveswar, D. (2021). *Business intelligence*. *January*.

Schwabish, J. (2021). *Better Data Visualizations:A Guide for Scholars, Researchers, and Wonks* (p. 464).

The KPI Institute. (2016). *What is a KPI?* https://smartkpis.kpiinstitute.org/

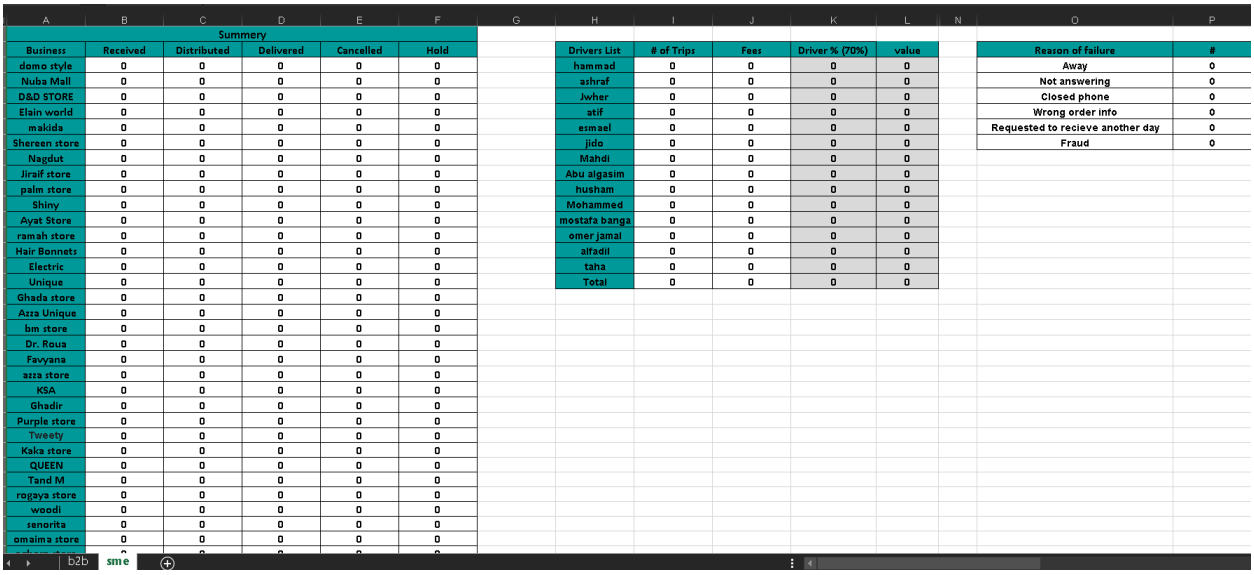
Wexler, S., Shaffer, J., & Cotgreave, A. (2017). *The Big Book of DashBoards* (Vol. 148).

APPENDIX

Hard Data from the Companies Current storing Solution

The screen shot below illustrates the current excel sheet used by the company as the database solution. We don't yet have access to the actual data so that part will have to wait until we get it the authorization. For now, we have made some dummy data from the given template. We have been shown the current visualizations used by the General Manager.

- The first table demonstrates the number of orders Received, Distributed, Delivered, Cancelled and Held by each store registered in the SME service in a given month.
- The second table is for the drivers showing their number of trips, Halan's fees, Drivers 70% cut and the delivered Order value which is to be returned to the store after a successful delivery.
- The third table illustrates the number of failed trips distributed by cause.



Summery					
Business	Received	Distributed	Delivered	Cancelled	Hold
domo style	0	0	0	0	0
Nuba Mall	0	0	0	0	0
D&D STORE	0	0	0	0	0
Elsin world	0	0	0	0	0
makida	0	0	0	0	0
Shereen store	0	0	0	0	0
Nagdut	0	0	0	0	0
Jiraif store	0	0	0	0	0
palm store	0	0	0	0	0
Shiny	0	0	0	0	0
Ayat Store	0	0	0	0	0
ramah store	0	0	0	0	0
Hair Bonnets	0	0	0	0	0
Electric	0	0	0	0	0
Unique	0	0	0	0	0
Ghada store	0	0	0	0	0
Azza Unique	0	0	0	0	0
bm store	0	0	0	0	0
Dr. Roua	0	0	0	0	0
Favyana	0	0	0	0	0
azza store	0	0	0	0	0
KSA	0	0	0	0	0
Ghadir	0	0	0	0	0
Purple store	0	0	0	0	0
Tweety	0	0	0	0	0
Kaka store	0	0	0	0	0
QUEEN	0	0	0	0	0
Tand M	0	0	0	0	0
rogaya store	0	0	0	0	0
woodi	0	0	0	0	0
senorita	0	0	0	0	0
omaima store	0	0	0	0	0

Drivers List	# of Trips	Fees	Driver % (70%)	value
hammad	0	0	0	0
ashraf	0	0	0	0
Jwher	0	0	0	0
atif	0	0	0	0
esmael	0	0	0	0
jido	0	0	0	0
Mahdi	0	0	0	0
Abu algasim	0	0	0	0
husham	0	0	0	0
Mohammed	0	0	0	0
mostafa banga	0	0	0	0
omer jamal	0	0	0	0
alfadil	0	0	0	0
taha	0	0	0	0
Total	0	0	0	0

Reason of failure	#
Away	0
Not answering	0
Closed phone	0
Wrong order info	0
Requested to recieve another day	0
Fraud	0

Halan Spread sheet template, Source; Halan Co. Ltd.

Software and Hardware Specifications

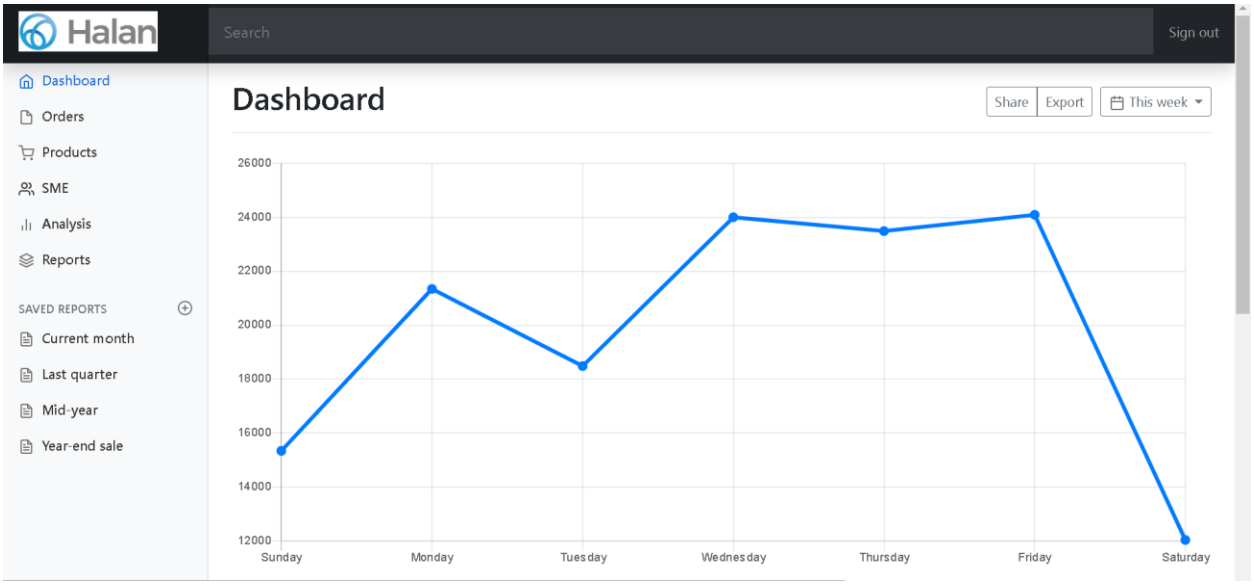
Proposed Hardware

UNIT	DESCRIPTION	QTY	COST PER UNIT	TOTAL COST
CPU	Intel Core i5 1.6 GHZ UDIMM 1TB 7.2 K Entry Sata 3.5 Cabled Hard Disk	2	51,200 SDG	102,400 SDG
Internet Provider	Wifi-router	1	14000 SDG	14000SDG
Computer Screen	18.5 Wide LED	2	17000 SDG	34000 SDG
Telephone Line	Sudani FWT Device	1	21500 SDG	21500 SDG
Keyboard	Computer Keyboard	2	1200 SDG	2400 SDG
Mouse	Computer Mouse	2	1100 SDG	2200 SDG
Printer	Hp laser Printer	1	70000 SDG	70000SDG

Proposed Software

SOFTWARE NAME	VERSION	FUNCTION
Any operating system that is compatible with web browsers	-	The Web Browser's Environment
Any Web Browser	latest	Used to Access the Dashboard
Oracle Database 10g	10.2.0.5	Database System
SQL Lite	3.36.0	Database Querying
HTML	5.0	Website Design
CSS	2.1	Style Sheet for Website Design
JavaScript	ECMAScript 2020	Client/Server-Side Website Interaction
Flask	2.0.1	Open-source Backend Web Framework
React.js	17.0.2	Web application front-end
Python	3.7	Data Analysis and Visualization
Pandas, NumPy, Scikit-Learn	latest	Data Exploration, Analysis and ML Algorithms python API
Matplotlib, Dash API, Plotly.js	latest	Data Visualization

User Interface Design



Section title

#	Header	Header	Header	Header
1,001	random	data	placeholder	text
1,002	placeholder	irrelevant	visual	layout
1,003	data	rich	dashboard	tabular
1,003	information	placeholder	illustrative	data
1,004	text	random	layout	dashboard
1,005	dashboard	irrelevant	text	placeholder
1,006	dashboard	illustrative	rich	data
1,007	placeholder	tabular	information	irrelevant
1,008	random	data	placeholder	text
1,009	placeholder	irrelevant	visual	layout
1,010	data	rich	dashboard	tabular
1,011	information	placeholder	illustrative	data
1,012	text	placeholder	layout	dashboard

Halan Staff Interview

Interview #1 Details			
Company Name:	Halan	Date: 27/7/2021	Time: 3:57 PM
Interviewer Name:	Omer Husham Ibrahim and Mohamed Elfadil		
Interviewee Title:	Data Analyst (DA)	Interviewee Name:	Aasim Khalid
Interviewee Phone Number:	0904107507		
Interviewee Role:	Producing reports, spotting patterns, and Collecting data and setting up infrastructure for data entry.		

Interview Questions

Question #1: How many employees do you have on the analytics part?

Notes: I am the only one.

Question #2: Explain the system in your own words & state the different role players.

Notes:

- Salesperson: gets stores & their contact's and forwards them to Moe Musa.
- Delivery Manager (Moe Musa): contacts the stores explains the protocol & gets... their orders info and forwards it to Gais
- Gais: recruits' drivers, allocate them to the different zones & sends them the stores pick up list
- Driver 1: picks up deliveries from stores in the morning & return's them to Gais
- Data analyst (Assim): creates the spread sheet template and shares it with Imthal
- Gais: sorts the orders by zones & allocates it to the different drivers & takes pictures of each driver's orders & sends it to Imthal
- Data entry Officer (Imthal): inputs the data to the spread sheet from pictures.
- Driver: picks up orders from Gais & delivers them. Receives payment & returns the money & delivery report to Tilal.
- Tilal updates the spread sheet with delivery, report & receives the delivery money pay's the drivers & gives the money to the small business in question.
- Assim creates reports from the spreadsheet data.

Question #3: What is the current system's maximum capacity for orders per day?

Notes: 1.000 orders.

Question #4: What are the main problems of the current system?

Notes:

- Creating reports.
- System dependability security & accounting.
- Calculating new targets.
- Data entry.

Question #5: What are the specific problems that you face with the current system?

Notes:

- The high degree of difficulty and time consumption associated with creation of reports.
- The Accounting process and tracking the circulation of money which has three major parts the order value which belongs to the store, the driver's cut and Halan's margin.

Additional Notes

A template of how the data is store in their excel sheets has been given to the interviewers to keep as reference. The fields are empty, but it is enough to start designing the database.

Interview #2 Details

Company Name: Halan Date: 23/8/2021 Time: 11:04 AM

Interviewer Name: Omer Husham Ibrahim and Mohamed Elfadil

Interviewee Title: General Manager (GM) Interviewee Name: Mohammed Anwer

Interviewee Phone Number: 0912109590

Interviewee Role: Managing staff, overseeing the budget, employing marketing strategies, and many other facets of the business.

Interview Questions

Question #1: In General Terms describe the main issue faced by the current system?

Notes: The SME branch is the only part of our service that isn't integrated with our system and application, so we are unable to track our delivery routes, so we are unable to optimize the road maps accordingly. On the other hand, delivery status is also adjusted manually which entails having an issue with data entry and the process of creating reports.

Question #2: Who is your Main Customer?

Notes: Halan itself is yet to start its Ecommerce business model and so the current business model relies on collaborating with vendors (online Stores) who meet a specific Threshold and use their customer base.

Question #3: What are the main KPIs that you need?

Notes:

- GMV (Gross Merchandise Value) which is an indicator of the total amount of money circulating around the company including (order value, drivers cut and Halan's margin) and is used as a measure of success.
- The driver's delivery success rate to monitor for fraud.
- The vendors estimated number of trips vs the real number of trips.
- Induvial vendors Delivery percentage.

Question #4: What is the Architecture of your current Backend (storage and computing)?

Notes: **Oracle Cloud Infrastructure** (OCI) based in the US.

Question #5: What are the specific problems that you face with the current system?

Notes:

- The high degree of difficulty and time consumption associated with creation of reports.
- The Accounting process and tracking the circulation of money which has three major parts the order value which belongs to the vendor, the drivers cut and Halan's margin.

Additional Notes

GM Notes: "Consider the Sudan Factor as not all the good ideas would be well implemented here."

The Gross Merchandise Value (GMV) is a very important figure for him as it shows him how well they are doing at the moment, especially since the SME branch is currently not profit oriented.
The route optimization is a consideration but could require a lot of time to prepare a decent algorithm so it should be considered as a project on its own.

Interview #3 Details

Company Name: Halan Date: 23/8/2021 Time: 11:51 AM

Interviewer Name: Omer Husham Ibrahim and Mohamed Elfadil Abdalla

Interviewee Title: Delivery Manager (DM) Interviewee Name: Mohammed Musa

Interviewee Phone Number: 0912203678

Interviewee Role: Organizes the daily delivery operations and that of the drivers and follows up with them when there is a performance issue.

Interview Questions

Question #1: What do you expect of the dashboard to be able to do?

Notes:

The target should be to produce daily, weekly, and monthly reports on the performance of the drivers and the percentage of orders successfully delivered. These reports should include, but not limited to:

- Delivery success rate for:
  - Drivers
  - SME
- How many orders each driver delivers,
- The amount of money being circulated,
- Percentage of items held and cancelled, in general, temporally, and per driver, in addition to the reasons for the failure, to aid in determining if this is a pattern and if there is a chance of fraud or is it just the low work ethic.

Question #2: Which part of the business process do you want us to focus on?

Notes: Well, you shouldn't bother with things such as the actual delivery process or distribution of orders and you shouldn't build the dashboard to track that. I want us to be able to input data such as number of orders made, delivered, held, and cancelled.

Question #3: When you say orders are being held and cancelled, what are the classifications of the common reasons for that?

Notes:

- Customer didn't pick their phone up.
- Customer's phone is off.
- Customer is not available at the agreed upon location.

Question #4: What are your current solutions to the delivery failure rate?

Notes:	We send an SMS to the customer, informing them that the driver is heading to them. This reduces the chances of the customer not answering and we can be informed ahead of time if they aren't able to receive at the agreed upon time.

Question #5:	What are you interested in knowing in regards of the SME through the dashboard?
Notes:	I mainly want to know how many orders are being made by each store and the percentage of these orders that is successfully delivered. I also want to know if there is a high rejection/cancellation rate of the orders coming from each store as it can tell me if I should continue working with them or if I should allocate more resources to said store.
Question #6:	Do you face any problems with the finances?
Notes:	I generally don't deal with the finances, but you could say the reporting of the finances is causing some headache and takes time. Sometimes the numbers don't add up and might be an increase or decrease. A way to track the finances and produce a daily, weekly, and monthly report of the flow of money. A financial balancing system. The money coming in and going out must break even, otherwise, we have a problem.

Question #7:	Is there any feature you think we can develop that would make things easier for you?
Notes:	An inventory management system. To track if an item has been received from the other businesses, left the office, has been held, or cancelled. It should be able to tell how long it has been on hold for in storage and if it has been returned to the business owner.

Additional Notes

The inventory management system should be discussed further and think about ways to integrate it as a mobile application with the proposed system if time permits for it.

The interviewee stressed the importance of figuring out how to distribute the money that is acquired during daily operations as it is a crucial task that has previously shown its difficulty in breaking even.

Interviews Summary

- The interviews were held at the Halan offices, Located in Al Taif, Alsawahli St.
- The questions were designed to be open ended and some of the follow up questions were based on insights or thoughts the interviewers had during the interviews.
- The DA explained the different roles of the SME team and how the daily operations are run, and the problems faced by him according to his experience.
- The GM has highlighted the concerns management has with the current way they run things and the Key Performance Indicators used to track the organization's success. They are using the OCI for their backend storage and computing so we need to consider how the system will later integrate with it.
- The DM highlighted the operational concerns with their current system such as delivery failure rates and tracking of the performance of their drivers. He provided an idea of what they expect of the dashboard's functionality.
- The financial issues were brough to light and if everything goes as planned, the dashboard should alleviate, if not eliminate, their problems.
- An Inventory management system and Route optimization API was requested. The ability to produce said upgrades in the given time will be further considered.