**DATA WAREHOUSE IMPLEMENTATION**

**Introduction:**

Data warehouse is a system that is used to store and manage large amounts of historical data from multiple sources for reporting and analysis. The process of implementing a data warehouse involves several steps, including the following:

* **Requirements gathering**: The first step in implementing a data warehouse is to identify the business requirements for the system. This typically involves working with stakeholders and end users to understand their data needs and how they plan to use the data warehouse.
* **Design**: Once the requirements have been gathered, the next step is to design the data warehouse. This involves creating a logical and physical model of the data, as well as a plan for how the data will be extracted, transformed, and loaded into the warehouse.
* **ETL (Extract, Transform, Load)**: This is the process of extracting data from various sources, cleaning and transforming it so that it can be loaded into the data warehouse.
* **Data loading**: After the data has been extracted, transformed, and cleaned, it is loaded into the data warehouse. This typically involves writing scripts or using ETL tools to load the data into the appropriate tables and columns.
* **Indexing and tuning**: Once the data has been loaded, the next step is to index and tune the data warehouse to improve performance. This typically involves creating indexes on frequently-used columns, partitioning large tables, and configuring the database for optimal performance.
* **Reporting and analysis**: After the data warehouse has been implemented, the final step is to create reports and perform analysis using the data in the warehouse. This typically involves using business intelligence (BI) tools to create reports and dashboards, as well as using data visualization and data mining tools to analyze the data.
* **Maintaining and upgrading**: The process of implementing a data warehouse doesn’t end with the initial implementation, Data warehouse system need to be continuously monitored, maintained and upgraded over the time to ensure the performance and accuracy of data.

**Scope:** Understanding and testing several big cloud platform for the efficiency to their users and performing data warehouse implementation on several criteria and therefore evaluating the result in order to make decision and predict future works

**Platforms Considered:**

Five cloud platform are being considered for this assessment

**BigQuery (Google) :**

BigQuery is a fully managed, cloud-native data warehouse that enables super-fast SQL queries using the processing power of Google's infrastructure. It allows you to analyze large and complex datasets using a SQL-like syntax, and it integrates with other Google Cloud Platform (GCP) services like Google Cloud Storage and Google Cloud Dataflow. With BigQuery, you can easily scale your analysis to handle petabyte-scale datasets and perform ad-hoc analysis on your data without the need for a centralized infrastructure. BigQuery also offers features such as data partitioning and clustering to optimize query performance, and it supports real-time streaming data ingestion.

**Azure (Microsoft):**

Microsoft Azure is a cloud computing platform and infrastructure created by Microsoft for building, deploying, and managing applications and services through a global network of Microsoft-managed data centers. It is a collection of integrated services, including computing, storage, data management, and analytics, that can be used to build, deploy, and manage applications and services.

It also provides a wide range of services that can be used to build and deploy applications, including virtual machines, web apps, mobile apps, and logic apps. These services can be used to create, deploy, and manage highly available, scalable, and fault-tolerant applications and services

Azure also provides a wide range of tools and services for data management and analytics, including Azure SQL Database, Azure Cosmos DB, and Azure Data Lake Storage. These services can be used to store, manage, and analyze large amounts of data, and they can be integrated with other Azure services, like Azure Machine Learning and Azure Stream Analytics, to build powerful data pipelines and analytics applications

**Keboola:**

Keboola is a cloud-based data management and analytics platform that helps organizations to extract, transform, and load (ETL) their data, and make it available for analysis and reporting. The platform is designed to be user-friendly and easy to use, making it accessible to both technical and non-technical users.

Keboola is offered as a software or as a service (SaaS) platform, and pricing is based on the number of data integration and transformation jobs, data storage, and data warehousing and analytics usage.

**Red Hat OpenShift:**

Red Hat OpenShift is a container orchestration platform that is based on Kubernetes, an open-source container orchestration system. It is designed to make it easy for organizations to deploy and manage containerized applications in a variety of environments, including on-premises, in the public cloud, and at the edge

One of the key features of OpenShift is its ability to manage and orchestrate containers. With OpenShift, you can deploy and manage containers using Kubernetes, which provides a set of APIs for automating the deployment, scaling, and management of containerized applications. OpenShift also provides a web-based management console and command-line tools for interacting with the Kubernetes API, making it easy for developers and operations teams to work together

**Firebolt:**

Firebolt is a technology company that specializes in creating high-performance data warehousing and analytics solutions. Their flagship product is a cloud-native data warehousing platform that is designed to handle extremely large and complex data sets. It uses a variety of advanced techniques, such as columnar storage, data compression, and distributed computing, to provide fast query performance and scalability. The platform is fully managed and can be easily integrated with other tools and platforms, such as data visualization and machine learning. Additionally, Firebolt also offers a suite of data analytics and management services to help customers optimize their use of the platform and gain deeper insights from their data.

**Criteria Selected:**

**Performance at scale:** refers to the ability of a system or process to handle an increasing amount of workload or data as the size of the operation grows. This can include factors such as speed, efficiency, and scalability. A system or process that performs well on a small scale may not perform as well when the scale of the operation increases, and vice versa. It is crucial to evaluate the performance of a system or process at different scales in order to ensure that it can handle the anticipated workload and maintain an acceptable level of performance.

**Ease of use:** explains how simple and intuitive a product or system is to navigate and operate. The more user-friendly a product is, the easier it is for people to understand and use it without needing extensive training or guidance. This can include factors such as clear and concise instructions, intuitive design, and minimal steps required to complete tasks. A product or system that is easy to use can improve productivity, reduce frustration, and increase overall satisfaction for users.

**Cost Efficiency:** is the ability of a product, service, or process to provide the desired outcome while minimizing costs. This can include factors such as using less labor and using technology to automate tasks. A cost-efficient product or system can help a business save money and increase profitability by reducing expenses while maintaining or improving quality. It can also help individuals save money by allowing them to get the same value for less money. In general, cost efficiency is the ability to produce goods and services with the least amount of inputs.

**Semi-structured data:** refers to information that has some inherent structure, but does not conform to the strict, formal structure of traditional, structured data. Examples of semi-structured data include XML and JSON files, which have a defined structure for organizing data, but also allow for the inclusion of additional fields or data elements that are not part of the standard structure. Semi-structured data is typically more flexible and adaptable than structured data, as it can accommodate a wider range of information and changing data requirements.

It is also easier to work with semi-structured data than unstructured data, but it can have more complexity than structured data. Tools such as NoSQL databases, which are designed to handle semi-structured and unstructured data, are often used to manage and analyze semi-structured data.

**Concurrency:**  is the ability of a system or computer program to perform multiple tasks simultaneously. This can include tasks such as processing multiple requests from users at the same time, running multiple processes in parallel, or handling multiple inputs and outputs. Concurrency allows a system to be more efficient by reducing idle time and increasing the overall throughput.

Concurrency can be achieved in several ways, such as multi-threading, multi-processing, or using an event-driven architecture. Multi-threading is the process of running multiple threads (smaller units of a process) within a single process, while multi-processing is the use of multiple CPUs or cores to perform multiple tasks simultaneously. An event-driven architecture is a way of designing software where the flow of the program is determined by events or messages, rather than a linear sequence of instructions, which allows for multiple tasks to be handled at the same time.

It is important to note that concurrency does not necessary imply parallelism, which is the ability to perform multiple tasks simultaneously using multiple CPU cores. Concurrency can be achieved even with a single core CPU, by time slicing and context switching.

**Granularity:**  Granularity refers to the level of detail or the size of a unit of information or measurement. In the context of data, granularity refers to the level of detail that a piece of data provides. For example, data about a single customer would have a high granularity, while data about all customers in a specific country would have a lower granularity.

In the context of software engineering, granularity refers to the size and complexity of a module, component or class. A small module with a single responsibility is said to have a fine granularity, while a larger module that does multiple things is said to have a coarse granularity

**Deployment:**  Deployment refers to the process of making a software application or system live and available for use. It includes all the tasks that are necessary to take a developed application and make it available for users to access and use. This can include configuring the application for the target environment, installing any necessary dependencies, and testing the application to ensure that it is working as expected

There are several different types of deployment, including:

* Development deployment: A deployment that is used for testing and development purposes, typically on a developer's local machine or a development environment.
* Staging deployment: A deployment that is used to test the application in a production-like environment, usually on a staging server.
* Production deployment: The deployment of an application to a production environment, which is available to end-users.

The exact steps involved in the deployment process will vary depending on the specific application and the technologies used. But generally, it can include tasks such as building, packaging, configuring, testing, and delivering the application to the target environment.

There are also several ways to deploy, it can be done manually or using automated tools, such as scripts, continuous integration and continuous deployment (CI/CD) pipelines. These tools can help to automate the deployment process and make it more efficient and reliable.

**Freshness:**  Data freshness refers to the timeliness of the data, or how recently the data was collected or updated. Fresh data is considered to be more accurate and relevant than stale data. The freshness of data is an important consideration when evaluating the quality and usefulness of data, especially in cases where the data is used to make decisions or predictions.

There are several ways to measure data freshness, for example:

* Age of data: This refers to how long ago the data was last updated. For example, data that was last updated a day ago is considered fresher than data that was last updated a week ago.
* Latency: This refers to the time it takes for data to be available after it is generated. For example, data that is available in real-time is considered fresher than data that is available with a delay.
* Data refresh schedule: This refers to how often the data is updated. For example, data that is updated every hour is considered fresher than data that is updated every day.

Data freshness is important in many applications, such as business intelligence, stock trading, and weather forecasting. In these cases, the timeliness of the data can have a significant impact on the accuracy and effectiveness of the decisions or predictions made using the data.

It is important to note that there is a trade-off between data freshness and data accuracy, as a high frequency data update may come with high latency and inaccuracies.

**Investigation methodology:**

An important characteristic of a cloud-based service is virtualization of physical resources which enables the execution of multiple jobs on the same, shared physical computing resource.

With the availability of several options and deployment models, understanding and analyzing the performance have become extremely critical in cloud computing, we therefore present different approach such as measurement of hardware and software configuration of any computer system prior to procurement and done by running Benchmark programs for studying performance, applying queuing model for the cloud, running SQL queries, building data visualizations while applying machine learning algorithms using product search engine and analyzing the performance using queuing model on five cloud platforms like Azure, Keboola, Firebolt, Red Hat OpenShift and BigQuery

Cost efficiency and performance and other criteria were put into consideration while testing for each platform in order to propose a better solution at hand and in future works

**PLATFORM EVALUATION**

**BigQuery (Google)**

**Performance on Scale**: In a typical Sql query that runs in 30 seconds, BigQuery displays not less than 330GB/sec dedicated hard-drives to read 1TB of data. A 330 Gigabit network to shuffle the 1.25 TB of data. 3,300 cores to uncompress 1TB of data and process 100 billion regular expressions at 1 μsec pe. That’s a lot of resources, so it’s quite impressive that BigQuery lets you use all this stuff for just the few seconds required for your job to complete

**Ease of use:** Integrated with other GCP product, BigQuery makes it easy to load data from sources such as cloud storage and also support a wide range of format for data ingestion and BigQuery’s data replication and disaster recovery features makes it a reliable platform for storing data and there are no limit on storage or compute resources

**Cost Efficiency**: BigQuery uses a federated data access model that allows you to query data directly from external data sources like Cloud Bigtable therefore You are billed for all the data that is read each time the query is run. Storage pricing is prorated per MB, per second. For example, if you store: 100 MB for half a month, you pay $0.001 (a tenth of a cent) 500 GB for half a month, you pay $5. With several improvement, BigQuery show a promising lead of reducing cost in future

**Semi-Structured Data**: With the introduction of JSON, user needs to ingest semi-structured data as a JSON data type, BigQuery allows each JSON field to be encoded and processed independently therefore creating issue for users who aren’t familiar with this type of format.

**Concurrency**: Queries can run simultaneously up to 100 interactive queries and having a column limit of 10000

**Granularity**: BigQuery automatically compressed, encrypted, replicated, and distributed therefore making it easier to store data with high detail and also have high performance

**Deployment**: BigQuery stores data using a columnar storage format that is optimized for analytical queries. BigQuery presents data in tables, rows, and columns and provides full support for database transaction semantics (ACID). BigQuery storage is automatically replicated across multiple locations to provide high availability

**Freshness**: BigQuery supported frequent and continuous ingestion of data but not relatively fast when it comes to streaming data which requires the help of Cloud DataFlow and also calling the API in your software

**AZURE (MICROSOFT)**

**Performance on Scale**: Azure Load Testing supports simulating up to 11,250 concurrent virtual users in a load test. It boast of speed upto 10Gb/s even though the ping rate is different across all countries, it offers maximum storage of 500TbB and maximum size of a single entity up to 1MB and maximum number of properties in a table entity is 255

**Ease of use**: An Azure environment helps you break away from on-premises servers with the cloud-based system means you can harness the power of Microsoft technology without limitations. It also offers a pricing calculator that allows businesses to make this cloud server opportunity an affordable reality

**Cost Efficiency**: Azure uses technologies that you and your users are already accustomed to using, like Windows, Active Directory and Linux, so the transition to the cloud is less obvious but offering three pricing model (pay-as-you-go, spot vms, reserved instances), it’s no doubt more expensive and as a Saas platform, it can easily become an extremely complicated environment for larger companies

**Semi-structured data**: Azure is very okay when working with both structured and semi-structured data uses tags or keys that organize and provide a hierarchy for the data which uses HBase provides random access and strong consistency for large amounts of unstructured and semi-structured data in a schema database organized by column families

**Concurrency**: The basic pricing tier offers only one concurrent login and offering a maximum of 128 concurrent queries will execute and remaining queries will be queued. The number of concurrent queries can decrease when users are assigned to higher resource classes or when the data warehouse unit setting is lowered

**Granularity**: it gathers, store, process, analyses and visualize data of any variety, volume or velocity. Design AI with Apache Spark-based analytics and also unify data governance solution that maximizes the business value of any data with cloud computing

**Deployment**: Azure originally provided only the classic deployment model. In this model, each resource existed independently; there was no way to group related resources together. Instead, you had to manually track which resources made up your solution or application, and remember to manage them in a coordinated approach

**Freshness**: Azure Data Factory is a cloud-based ETL and data integration service to create workflows for moving and transforming data. With Data Factory you can create scheduled workflows (pipelines) in a code-free manner

**KEBOOLA**

**Performance on scale:** Keboola is designed to handle large volumes of data efficiently and to provide fast performance for a wide range of data integration and management tasks. However, without more information about the specific context in which Keboola is being used, it is not possible to provide a more detailed assessment of its speed.

**Ease of use**: The time, resources, and manpower required for scaling and maintenance on Keboola will depend on a variety of factors, including the specific tools and services being used, the complexity of the data being processed, and the size and nature of the organization using the platform.

In general, Keboola is designed to be highly scalable and to require minimal maintenance, so users can focus on data insight discovery rather than on technical tasks. The platform is cloud-based, so users do not need to worry about hardware or infrastructure maintenance. Keboola also offers a range of features and tools to help users manage their data efficiently, including automated data integration and transformation processes, data governance and security controls, and data lineage tracking.

**Cost Efficiency**: Keboola offers a free trial for users to test out the platform and its various tools and services, It is worth noting that Keboola’s pricing is subject to change over time, and it is not possible to predict with certainty what the future growth of the platform or its pricing will be. However, Keboola has a strong track record of growth and innovation in the data integration and management space, and it is likely to continue expanding and evolving in the future.

**Semi-Structured Data**: Keboola is designed to handle a wide range of data types, including structured, semi-structured, and unstructured data. For semi-structured data in particular, Keboola offers a number of features that can help users manage and analyze this type of data effectively. These include the ability to extract and transform data from a wide range of sources, including JSON, XML, and CSV files, as well as support for data manipulation and transformation using SQL and other programming languages

**Concurrency**: The platform uses a distributed architecture and has built-in mechanisms for scheduling and prioritizing tasks to ensure that they are completed efficiently, Users can also adjust the resources (e.g. computing power, memory) allocated to their Keboola account to ensure that they have sufficient capacity to handle the concurrency needs of their tasks. This can be done through the Keboola user interface or through the Keboola API.

**Granularity**: The platform is designed to be able to handle data at a wide range of granularities and to support the manipulation and transformation of data at different levels of detail. If a user is working with very large volumes of data or is performing resource-intensive tasks, they may need to allocate additional resources or optimize their processes in order to achieve good performance. However, this does not necessarily mean sacrificing granularity

**Deployment**: Keboola offers multi-cloud support, allowing users to connect to and integrate data from a wide range of cloud-based data sources, including Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. This can be useful for organizations that have data stored on multiple cloud platforms, or that want to use the strengths of different platforms for different tasks

**Freshness**: Users can configure Keboola to continuously ingest data from various sources, such as databases, APIs, and files, and to update the data in their Keboola account as new data becomes available. This can help to ensure that the data in Keboola is always up-to-date and reflects the most current information.

**RED HAT SHIFT**

**Performance on Scale**: OpenShift includes a number of features to help ensure the performance of your applications, including:

* Resource limits: You can set limits on the CPU and memory usage of your applications to ensure that they do not consume too many resources and negatively impact the performance of other applications.
* Auto-scaling: OpenShift can automatically scale the number of replicas of your application up or down based on demand, helping to ensure that your application has enough resources to meet the demands of your users

**Ease of Use**: Red Hat OpenShift is designed to be an easy-to-use platform for developing, deploying, and managing container-based applications. It provides several features and tools to help simplify the process of developing, deploying, and managing applications on the platform:

* Developer-centric workflow: OpenShift is designed to be developer-friendly, with a focus on streamlining the process of building, testing, and deploying applications. It provides several tools and features to help developers work more efficiently, such as Git integration and automated builds.
* Web console: OpenShift includes a web console that provides a simple, intuitive interface for managing applications and resources on the platform. The console allows you to view and manage your applications, deploy new versions of your application, view log files, and more.
* Command-line interface (CLI): OpenShift also provides a command-line interface (CLI) that allows you to manage your applications and resources from the command line. The CLI provides some useful commands for managing applications, such as creating new projects, deploying applications, and viewing application logs.
* Integration with popular tools and frameworks: OpenShift integrates with some popular tools and frameworks, such as Jenkins, Eclipse, and Visual Studio Code, which can help simplify the development process for developers.

**Cost Efficiency**: Integrated with cloud providers, OpenShift can be deployed on-premises or on a public cloud provider such as Amazon Web Services (AWS) or Microsoft Azure. This allows you to take advantage of the cost savings and other benefits offered by these providers.

* Red Hat offers OpenShift as a subscription service, with pricing based on the number of nodes in your deployment and the level of support that you require. This can provide a predictable and cost-effective way to use the platform.

**Semi-structured Data**: When working with semi-structured data on Red Hat OpenShift, you may not have the same level of structure and predictability as you would with structured data. Semi-structured data is data that has some inherent structure but may also contain unstructured elements or may not adhere to a strict schema. This can make it more challenging to work with and analyze compared to structured data, which follows a fixed set of rules and has a clear, well-defined scheme. Overall, the tradeoffs of working with semi-structured data on Red Hat OpenShift will depend on your specific needs and requirements. If you need a high level of structure and predictability in your data, structured data may be a better fit. However, if you are more interested in flexibility and the ability to incorporate a wide range of data sources, semi-structured data may be a more suitable choice.

**Concurrency**: To support high levels of concurrency, OpenShift uses container technology to allow multiple applications to share the same operating system and hardware resources. This can help reduce the overhead associated with running multiple applications, allowing more applications to run on the same hardware.

* In addition, OpenShift includes features such as auto-scaling and resource limits, which can help ensure that applications have the resources they need to handle high levels of concurrency. Auto-scaling allows you to automatically adjust the number of replicas of your application based on demand, while resource limits allow you to set limits on the CPU and memory usage of your applications to ensure that they do not consume too many resources.

**Granularity**: Red Hat OpenShift is designed to be a highly scalable and performant platform, and it includes some features and tools to help you achieve good performance for your applications. These features include resource limits, auto-scaling, and deployment strategies, which can help ensure that your applications have the resources they need to perform well and remain available.

**Deployment**: OpenShift provides some different deployment strategies to help you manage the rollout of new versions of your application, including rolling deployments and blue-green deployments, which can help minimize downtime and ensure that your application remains available and performs well

**Freshness**: Red Hat OpenShift provides some tools and features to support continuous data ingestion, including:

* Red Hat AMQ Streams: A Kafka-based streaming data platform that allows you to ingest, process, and analyze large volumes of data in real time.
* Apache Flume: A distributed system for collecting, aggregating, and moving large amounts of log data from multiple sources to a centralized data store.
* Apache Nifi: An open-source platform for automating the movement and transformation of data between systems

**FIREBOLT**

**Performance on Scale**: Firebolt is designed to be a fast data warehousing solution, with the ability to execute complex queries in seconds. It uses advanced optimization techniques and in-memory technology to achieve this performance, making it a powerful tool for business intelligence and data analysis. The exact speed at which Firebolt can run queries will depend on a variety of factors, such as the size and complexity of the data being queried, the resources allocated to the data warehouse, and the specific hardware and software configurations being used. However, in general, Firebolt is known for its fast query performance and ability to quickly analyze large amounts of data

**Cost Efficiency**: Firebolt is a fully managed data warehousing service, which means that it is designed to be easy to set up and maintain without requiring a lot of time or technical expertise. This can help to reduce the overall cost of ownership compared to building and maintaining a data warehouse using on-premises infrastructure. Additionally, Firebolt is fully scalable, allowing you to easily adjust the capacity of your data warehouse to meet the needs of your business without the need for upfront investment in hardware or other infrastructure. This can help to make it more cost-effective compared to solutions that require you to over-provision resources or pay for unused capacity. Finally, Firebolt is designed to be easy to use, with a simple, intuitive interface that can help to reduce the need for specialized training or ongoing support.

**Semi-structured Data**: When working with semi-structured data on Firebolt, you may not have the same level of structure as you would with structured data, which could make it more difficult to query and analyze. However, the advantage of working with semi-structured data is that it can be more flexible and easier to work with than structured data, as it does not have a fixed schema and can accommodate a wider range of data types.

Some potential tradeoffs of working with semi-structured data on Firebolt include:

* Increased complexity: Querying and analyzing semi-structured data can be more complex, as the data may not be organized in a predictable way.
* Reduced performance: Depending on the size and complexity of the data, querying semi-structured data may be slower than querying structured data, as the database must perform additional processing to extract and interpret the data.
* Reduced reliability: Semi-structured data may be more prone to errors and inconsistencies, as there is no fixed schema to enforce data integrity. This could impact the reliability of your queries and analyses.

Overall, whether or not working with semi-structured data on Firebolt is a good fit for your use case will depend on the specific requirements and goals of your project. It may be worth considering the tradeoffs and deciding whether the benefits of using semi-structured data outweigh the potential drawbacks.

**Ease of use**: Ease of use is a subjective measure and can vary depending on the user’s familiarity with data warehouses and SQL. However, Firebolt is designed to be easy to use for a wide range of users, including those with little to no experience with data warehousing.

Some features of Firebolt that may contribute to its ease of use include:

* Intuitive user interface: Firebolt has a web-based interface that is designed to be easy to navigate and use.
* Simplified data ingestion: Firebolt makes it easy to load data from a variety of sources, including file-based data, streaming data, and data from other databases.
* SQL support: Firebolt supports standard SQL, which is a widely-used query language that is familiar to many users. This can make it easier for users to learn how to query and analyze data using Firebolt.
* Built-in documentation and resources: Firebolt provides extensive documentation and resources to help users get started with the platform and learn how to use its features.

**Concurrency**: In general, the number of users who need to query data on Firebolt and the need for queries to run simultaneously will depend on the specific requirements and goals of your project. Some factors to consider when determining the number of users and the need for simultaneous queries may include:

* The size and complexity of the data: Larger and more complex data sets may require more resources and may be more time-consuming to query, which could impact the number of users who can simultaneously query the data.
* The performance requirements: If your project requires fast query performance, you may need to limit the number of users who can simultaneously query the data in order to ensure that queries are completed in a timely manner.
* The scalability of the system: If you anticipate that the number of users and queries will increase over time, you may need to consider a data warehouse platform that is highly scalable in order to handle the increased traffic and demand

**Granularity**: Firebolt is designed to provide fast querying and analysis of data, and it should be able to handle a high level of granularity without sacrificing performance. One of the ways it is able to achieve this is by using columnar storage, which allows it to efficiently store and retrieve data by column rather than by row. This can be particularly beneficial when working with large datasets, as it allows you to quickly filter and aggregate data without having to scan the entire dataset.

That being said, it is always possible that some queries may be slower than others, depending on the specific characteristics of the data being queried and the resources available to the system. If you are experiencing performance issues with specific queries, it may be helpful to review the query plan and see if there are any optimization opportunities. Additionally, Firebolt provides a number of features and configurations that can help you fine-tune the performance of your data warehouse, such as selecting the appropriate instance size, using materialized views, and setting up query concurrency

**Deployment**: There are a few different options for deploying Firebolt, depending on your specific needs and requirements. Here are a few of the most common deployment options:

* Cloud Deployment: Firebolt is available as a fully managed service on the cloud, which means you can use it without having to set up any infrastructure yourself. You can simply sign up for an account and start using the service.
* On-Premises Deployment: If you prefer to keep your data on-premises, you can install Firebolt on your own hardware. This allows you to have full control over your data and infrastructure, but it also requires you to manage the hardware and maintain the system yourself.
* Hybrid Deployment: It is also possible to set up a hybrid deployment, where some data is stored on-premises and some is stored on the cloud. This can be a good option for organizations that have specific data privacy or compliance requirements, or that need to keep some data close to the source.

Regardless of the deployment option you choose, Firebolt provides tools and documentation to help you get started and manage your deployment effectively

**Freshness**: Firebolt does support continuous ingestion of data to enable data freshness. One of the main features of Firebolt is its ability to handle large volumes of data in real-time, allowing you to quickly and easily analyze up-to-date data.

There are a few different ways you can set up continuous ingestion of data into Firebolt:

* Load data directly from your source systems: You can use the Firebolt API or one of the available connectors to load data directly from your source systems into Firebolt. This can be a good option for sources that generate data continuously, such as log files or streaming data sources.
* Use a data pipeline: You can also use a data pipeline to ingest data into Firebolt on a continuous basis. A data pipeline is a series of processing steps that allows you to extract, transform, and load data from various sources into your data warehouse. There are several tools and services available that can help you set up a data pipeline, such as Apache Beam or Google Cloud Data Fusion.

Regardless of the approach you choose, it is important to carefully plan and design your data ingestion process to ensure that it is efficient and effective. This may involve setting up appropriate data schemas, defining data transformations, and configuring error handling and retry logic

**COMPARATION RESULT**

**Rating Scale:**

1 – Unsatisfying

2 – Below Expectation

3 – Meets Expectation

4 – Exceeds Expectation

5 – Phenomenal

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CRITERIA** | **Investigation methodology** | BIG QUERY | AZURE | KEBOOLA | RED HAT OPENSHIFT | FIREBOLT |
| Performance on scale | How fast can this data warehouse run, on how much data and with how many resources (affecting costs)? | 5 | 4 | 4 | 5 | 5 |
| Ease of use | How much time, resources and manpower will you need to spend on scaling and maintenance as opposed to insight discovery? | 4 | 4 | 4 | 5 | 5 |
| Cost Efficiency | What is your data set size and performance requirement today and what will it be in 1-2 years? Is the data warehouse efficient enough to support your future growth? | 5 | 3 | 4 | 3 | 4 |
| Semi-structured data | Will you get the same experience as working with structured data when working with semi-structured data? What will be the tradeoffs of working with semi-structured data? | 3 | 3 | 4 | 4 | 5 |
| Concurrency | How many users need to query the data? Do queries need to run simultaneously? What are your future needs? | 5 | 3 | 4 | 3 | 5 |
| Granularity | Will you have to sacrifice granularity to achieve performance? | 4 | 4 | 5 | 3 | 5 |
| Deployment | Is the cloud data warehouse available on the cloud platform you use today? How important is multi-cloud support for you? | 3 | 4 | 5 | 4 | 2 |
| Freshness | Does the evaluated cloud data warehouse support continuous ingestion of data to enable data freshness? | 4 | 5 | 4 | 5 | 5 |
| Weight Total Score |  | 33 | 29 | 34 | 32 | 34 |

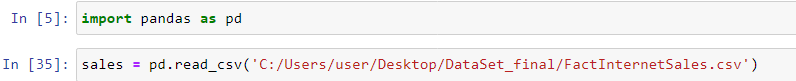
**BIG DATA PROCESSING AND ANALYSIS IMPLEMENTATION**

The following big data processing and task implemented is done using Jupyter Notebook (Python) and following task were performed

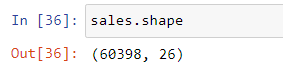
* EDA
* PRICE PREDICTION
* PREDICTION AUTOMATION

**EDA**

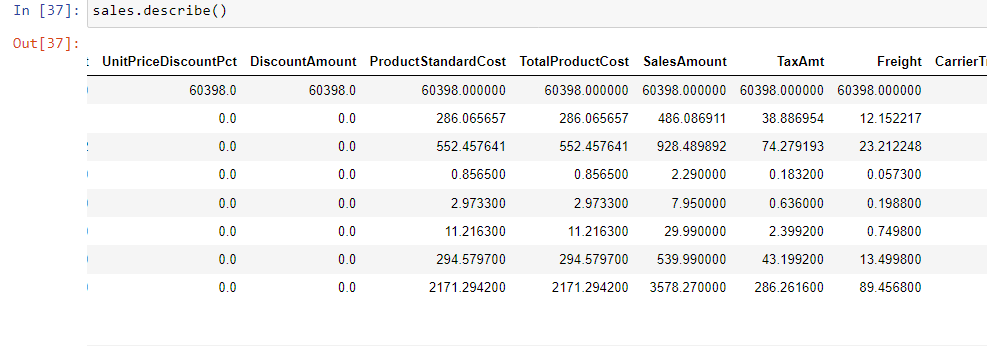
Importing our dataset into python, we employ pandas library, Pandas is a powerful and popular open-source data manipulation and data analysis library for Python. It provides fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive and also makes it easy to explore and analyze data. It provides functions for calculating summary statistics and data visualization tools that make it easy to understand the structure and distribution of data



Exploratory Data Analysis of our data set:

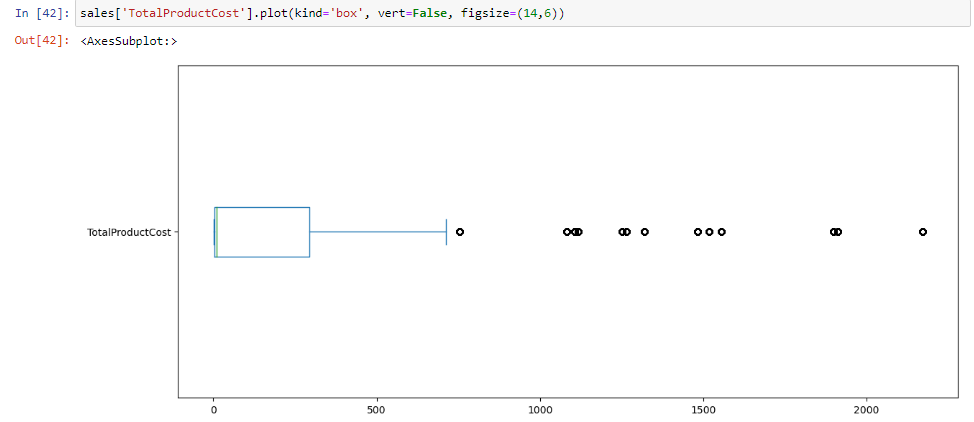


This shows our data set contains 60398 rows and 26 columns

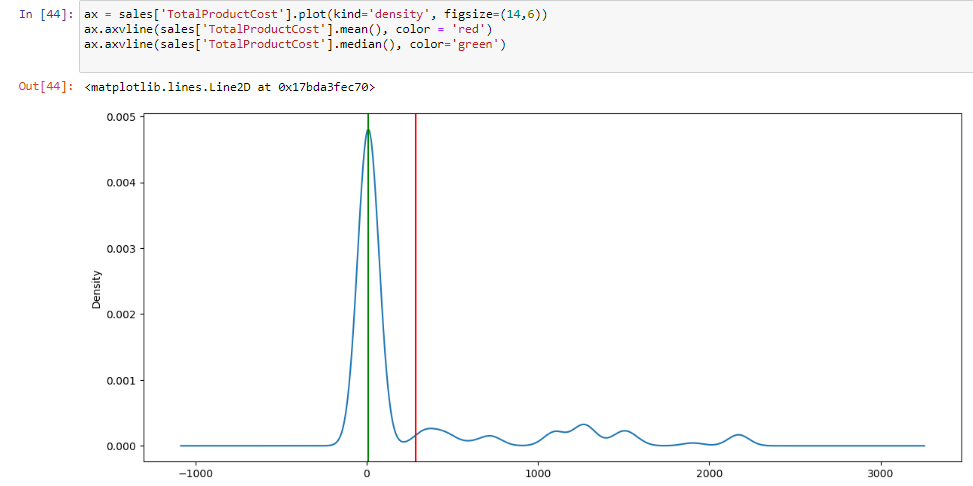


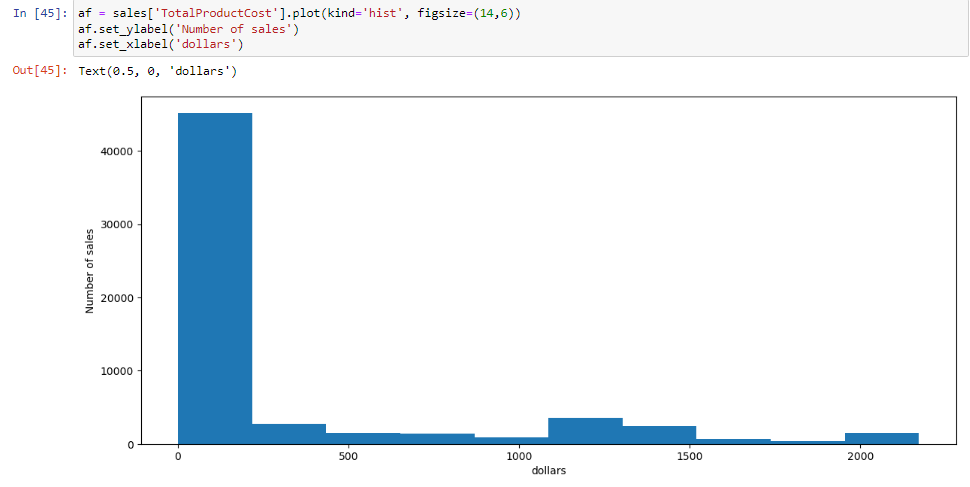
The above table showing count , mean, std, min, 25%, 50%,,75% and max (row wise) gives an overview of our minimum product cost and maximum product cost also showing other important variable like sales amount, unit price and more.

In order to visualize our data to gain a better understanding and see how our data is distributed, we use matplotlib library, With matplotlib, you can create line plots, scatter plots, bar plots, error bars, histograms, bar charts, pie charts, box plots, and many other types of visualizations. In addition to the plotting functions, matplotlib also provides a pyplot interface, which is similar to the one used in the popular plotting library MATLAB, that allows you to use a simpler and more convenient syntax to create plots and customize them.

Plotting our independent variable Total Production Cost This shows their exist outliers in our total product cost, product cost higher than is an outlier

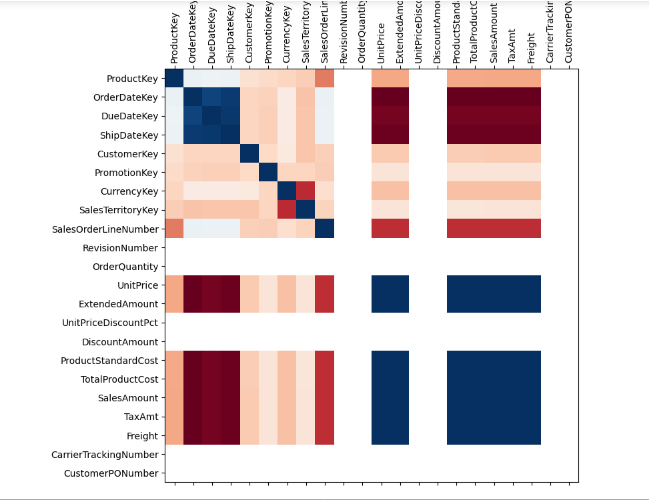
Plotting density mean and median of total production cost to see if there is any major difference in our outliers



Visualizing our product cost 

This shows the company sold a large amount of product upto $500 and little sales when it comes to product of high cost

Checking and visualizing if there exist any correlation in our dataset



Red -> low correction (negative correlation)

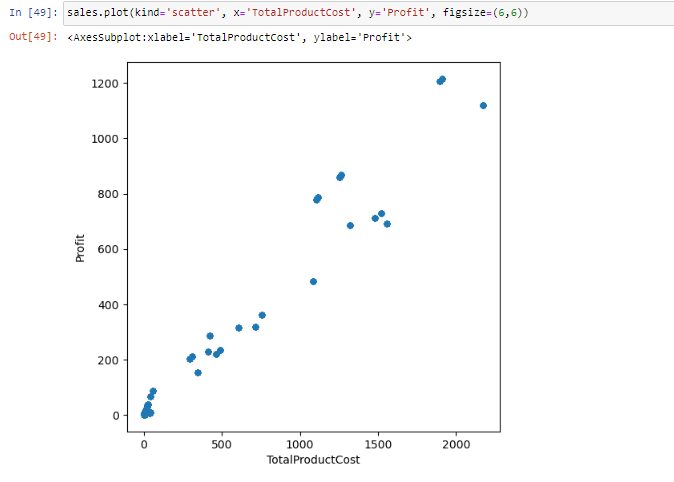
Blue -> high correlation (correlation = 1)

White -> missing values or no correlation

This shows there is high and positive correlation between production cost, sales amount, tax amount and freight while there is negative correlation between other variables

Checking if the company makes profit from the total sales amount ..

We therefore create a new column named profit (sales amount- production cost) and visualizing it



This illustrates low profit is made on sales but generally no loss

**PRICE PREDICTION:**

In order to create a model that predict sales price for our data set, we employ the Scikit-learn, also known as sklearn, is a Python library for machine learning that provides a wide range of tools for tasks such as classification, regression, clustering, dimensionality reduction, and model selection

**Here are some key features of scikit-learn:**

It provides a large collection of simple and efficient tools for data mining and data analysis.

It is built on top of other Python libraries such as NumPy, SciPy, and matplotlib, which makes it easy to use in conjunction with these other tools.

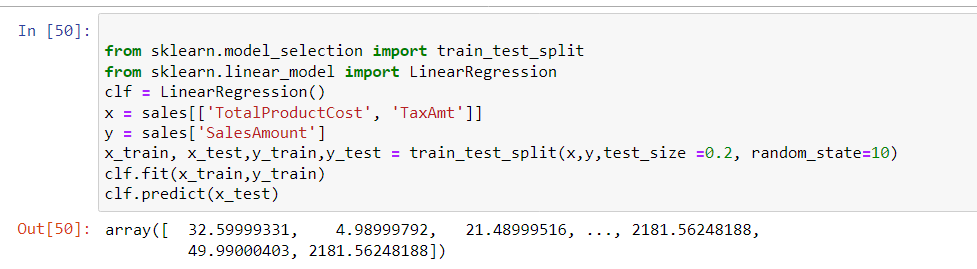
It includes several classification algorithms (e.g., SVM, Random Forest, KNN), Regression algorithms (e.g., Linear, Ridge, Lasso) and clustering algorithms (e.g., K-Means, Spectral Clustering, GMM etc.)

It includes built-in functions for cross-validation, model selection, feature selection and pre-processing.

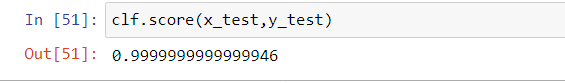
It has extensive and detailed documentation, including an extensive user guide, tutorials, and sample datasets, which makes it easy to learn and use

Getting to our prediction model:

First step is to split our data into independent and dependent variable and combining the two to train and test dataset which can be seen in the following diagram:



We need to check the accuracy of our prediction model therefore running the following code:



99% shows good accuracy as our train set is 80% and test data 20%

**PRICE AUTOMATION:**

We put our model into a project as we don’t want to run every code in order to get a prediction. This enables us to ask the model for a sale price prediction if we pass in production cost and tax amount.

In order to do this, we employ Joblib library which is a Python library that provides a set of tools for caching and managing the computation of large, reusable data. It is particularly useful for large machine learning projects, where the same data and computation is used repeatedly

Some of the main features of joblib include:

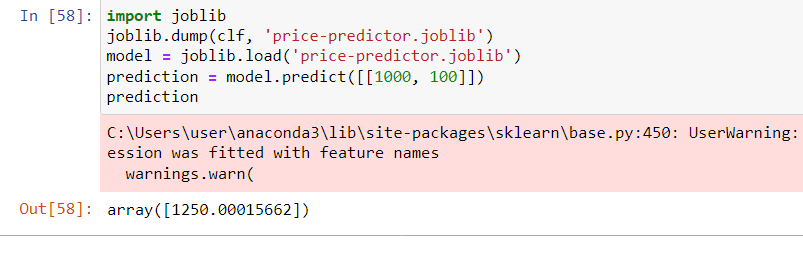
A transparent and efficient memory cache for storing large arrays, dataframes, and other data types that can be used across multiple Python processes or threads.

Tools for parallel and distributed computing, including support for multiprocessing, threading, and memory-mapped files.

Support for on-disk caching, which allows large datasets to be stored on disk and loaded into memory as needed.

Tools for checkpointing and persisting the state of Python objects, which can be useful for saving the progress of long-running computations or for sharing intermediate results between multiple users.

We apply the following steps in order to save our model into a joblib project and then ask for a prediction



**SUMMARIZATION**

**BIgQuery**

BigQuery is a fully managed, cloud-native data warehouse that enables super-fast SQL queries using the processing power of Google's infrastructure. With BigQuery, you can analyze large and complex datasets by creating SQL-like queries, and it allows you to integrate with other Google Cloud services such as Dataproc, Cloud Dataflow, and Cloud Storage. BigQuery also supports data ingestion from a variety of sources, including CSV and JSON files, as well as streaming data. Additionally, it offers strong security features, including support for fine-grained access controls, data encryption at rest and in transit, and integration with Google Cloud Identity and Access Management.

**Azure**

Azure is a cloud computing platform and service created by Microsoft for building, deploying, and managing applications and services through a global network of Microsoft-managed data centers. It offers a wide range of services including computing, storage, networking, and analytics, as well as a variety of tools for building and deploying applications. Some of the key features of Azure include:

* Azure Virtual Machines (VMs), which provide on-demand, scalable computing resources
* Azure Storage, which provides a range of options for storing and managing data
* Azure SQL Database, a fully managed relational database service
* Azure Cosmos DB, a globally distributed, multi-model database
* Azure Kubernetes Service (AKS) for container orchestration
* Azure Functions, a serverless compute service
* Azure DevOps, a set of tools for managing the software development lifecycle
* Azure Active Directory (AD), a service for managing user identities and access

Azure also includes a wide range of services for IoT, AI, Machine Learning, Blockchain, and more. It allows integration and interaction with various programming languages, frameworks, and tools, including .NET, Java, Python, and more.

With Azure, you can build, deploy, and scale applications and services on Microsoft's global network of data centers and it also supports hybrid scenarios that allow you to integrate your existing on-premises infrastructure with the cloud.

**KEBOOLA**

Keboola is a data management platform that enables companies to manage and transform their data for use in business intelligence and analytics. The platform provides a wide range of tools and services for data warehousing, data integration, data quality, and data modeling, as well as support for data visualization and reporting.

One of the key features of Keboola is its data integration capabilities. It allows you to easily connect to a wide variety of data sources, such as databases, file systems, cloud storage, and APIs, and to perform data transformation tasks, such as data cleansing, data mapping, and data aggregation, using a web-based interface or a simple API.

* Keboola also provides a range of tools for data warehousing, including a cloud-based data warehouse service, as well as support for data quality and data modeling, with features like data validation and data lineage.
* Keboola offers a developer-friendly environment, with many SDKs and APIs, that enables easy integration with other systems, for example for data science or BI. Keboola allows for data extraction and integration with other popular systems, like Redshift, Snowflake, BigQuery, and more.
* Keboola is a cloud-based solution that allows customers to focus on their data, rather than managing infrastructure and IT.

**RED HAT OPENSHIFT**

Red Hat OpenShift is a container orchestration platform based on Kubernetes, an open-source system for automating the deployment, scaling, and management of containerized applications. OpenShift provides an enterprise-grade platform for deploying and managing containerized applications on-premises, in a hybrid environment, or in the public cloud.

OpenShift includes a range of features and tools for building and deploying containerized applications, including:

* A web-based user interface and command-line interface (CLI) for managing and monitoring applications and resources
* Built-in support for continuous integration and continuous deployment (CI/CD) workflows
* Automated rollouts and rollbacks for zero-downtime deployments
* Self-service provisioning of developer workspaces and resources, such as databases and message queues
* Support for a wide range of programming languages and frameworks, including Java, .NET, PHP, Python, Ruby, and more
* Automated load balancing, automatic scaling and automatic failover for high availability

OpenShift also provides a robust security model, that can be integrated with enterprise-level identity management systems, such as Microsoft Active Directory or FreeIPA.

OpenShift is designed to work seamlessly with other Red Hat products, like Ansible, and also integrate with other cloud providers like AWS, Azure and Google Cloud.

In summary, OpenShift is a powerful and flexible platform for managing containerized applications at scale, that also provides a wealth of built-in capabilities and integrations that enable teams to focus on delivering applications, rather than on the underlying infrastructure.

**FIREBOLT**

Firebolt is a cloud-native, real-time data analytics platform that enables organizations to ingest, process and analyze large and complex datasets in real-time. It is specifically designed to handle large scale, high-performance analytics, and it integrates with popular data warehousing and cloud storage solutions, like Snowflake and Amazon S3. Firebolt also provides a range of built-in functions for data transformation, and it allows for the creation of custom data pipelines using SQL or Python.

One of the key features of Firebolt is its ability to handle real-time and historical data analytics, with near-zero latency. It utilizes a unique data indexing technology and in-memory processing architecture that enables it to perform real-time analytics on very large datasets with high concurrency, making it a great option for companies looking to make real-time data-driven decisions.

* Firebolt also provides a user-friendly and intuitive web-based interface, and it allows users to easily integrate with other analytics tools, such as Tableau, Looker and PowerBI, to visualize data and share insights.
* Firebolt provides security features such as end-to-end encryption and role-based access controls, and it also allows for advanced data governance and compliance capabilities to meet the regulatory requirements of different industries.

In total, Firebolt is a powerful platform for real-time analytics, designed for companies that need to perform high-performance analytics on large and complex datasets in real-time.

**EXPERIENCE DISCUSSION:**

The cloud platforms discussed in this assessment such as BigQuery, Azure, Keboola, OpenShift, and Firebolt, all offer different capabilities and features, but there are some common themes that I learn from them:

* Scalability and Flexibility: All of these platforms are designed to allow organizations to scale their resources and workloads as needed, whether that's computing power, storage capacity, or data processing capabilities.
* Data management: All of these platforms provide various data management features, including data warehousing, data integration, data quality, and data modeling.
* Cloud native: They are all cloud-native solutions that don't require extensive IT management and maintenance, instead, they focus on providing value for customers with data, through services and tools provided by the platform.
* Security: All platforms provide various security features that protect the data, whether it's data encryption at rest and in transit, or fine-grained access controls.
* Integration: All the platforms allow integration with other cloud services or external systems and they also support multiple programming languages and frameworks.
* Performance: All these platforms are designed to handle high-performance, high-concurrency workloads, and they offer high-performance analytics and real-time data processing capabilities.

**FUTURE WORK**

While the cloud platforms we have evaluated are BigQuery, Azure, Keboola, OpenShift, and Firebolt, are all powerful and feature-rich, there are always areas where they can be improved. Here are some things that could be improved upon in these platforms:

* Cost: While these platforms offer a lot of functionality, they can be expensive, especially for organizations that need to process and analyze large amounts of data.
* Complexity: The platforms can be complex to use and require significant technical expertise to fully utilize their capabilities, which may make it difficult for some users to get started.
* Limited customization: Some of the platforms offer a limited ability to customize the platform to suit specific needs, this can limit the ability to optimize the platform for a particular use case or workflow.
* Limited support for certain data formats: Some of the platforms might not support certain data formats or protocols, which could limit the ability to integrate with some data sources.
* Limited data governance and compliance capabilities: While these platforms offer some data governance and compliance features, but they might not meet all the regulatory requirements of certain industries.
* Limited capabilities for data discovery and data exploration: Some of the platforms might not provide a great user experience for data discovery, data exploration and data visualization, which makes it difficult for the users to understand the data they have.

Of course, these are just some examples and each platform has its own strengths and weaknesses. As well, this industry is always evolving and improving, so it's possible that many of these issues are already being addressed or will be addressed in the future.