UNIT 2

Three Management Layers

Explain the things to be recorded in the Operational management layer.

Operational Management Layer

The operational management layer is the core store for the data science ecosystem's complete processing capability. The layer stores every processing schedule and workflow for the all-inclusive ecosystem. The following things have to be recorded:

1. Processing-Stream Definition and Management

The processing-stream definitions are the building block of the data science ecosystem. Definition management describes the workflow of the scripts through the system, ensuring that the correct execution order is managed, as per the data scientists' workflow design.

2. Parameters

The parameters for the processing are stored in this section, to ensure a single location for all the system parameters.

3. Scheduling

The scheduling plan is stored in this section, to enable central control and visibility of the complete scheduling plan for the system. Drum-Buffer-Rope methodology. Similar to a troop of people marching, the Drum-Buffer-Rope methodology is a standard practice to identify the slowest process and then use this process to pace the complete pipeline. You then tie the rest of the pipeline to this process to control the eco-system's speed. So, you place the "drum" at the slow part of the pipeline, to give the processing pace, and attach the "rope" to the beginning of the pipeline, and the end by ensuring that no processing is done that is not attached to this drum. This ensures that your processes complete more efficiently, as nothing is entering or leaving the process pipe without been recorded by the drum's beat.

4. Monitoring

The central monitoring process is in this section to ensure that there is a single view of the complete system. Always ensure that you monitor your data science from a single point. Having various data science processes running on the same ecosystem without central monitoring is not advised.

5. Communication

All communication from the system is handled in this one section, to ensure that the system can communicate any activities that are happening.

6. Alerting

The alerting section uses communications to inform the correct person, at the correct time, about the correct status of the complete system. If any issue is raised, alerting provides complete details of what the status was and the errors it generated.

Audit, Balance, and Control Layer

The audit, balance, and control layer controls any processing currently under way. This layer is the engine that ensures that each processing request is completed by the ecosystem as planned. The audit, balance, and control layer is the single area in which you can observe what is currently running within your data scientist environment.

It records

- Process-execution statistics

- Balancing and controls
- Rejects- and error-handling
- Fault codes management

The three subareas are utilized in following manner.

Explain in brief the indicators used in the Audit sub layer.

Audit

An audit is a systematic and independent examination of the ecosystem. The audit sublayer records the processes that are running at any specific point within the environment. This information is used by data scientists and engineers to understand and plan future improvements to the processing. In the data science ecosystem, the audit consists of a series of observers that record preapproved processing indicators regarding the ecosystem

1. Built-in Logging

Design your logging into an organized preapproved location, to ensure that you capture every relevant log entry. Deploy five independent watchers for each logging location, as logging usually has the following five layers.

1. A) Debug Watcher

This is the maximum verbose logging level. If any debug logs in ecosystem, means that the tool is using precise processing cycles to perform low-level debugging.

1. B) Information Watcher

The information level is normally utilized to output information that is beneficial to the running and management of a system.

1. C) Warning Watcher

Warning is often used for handled “exceptions” or other important log events. Usually this means that the tool handled the issue and took corrective action for recovery.

1. D) Error Watcher

Error is used to log all unhandled exceptions in the tool. This is not a good state for the overall processing to be in, as it means that a specific step in the planned processing did not complete as expected. Now, the ecosystem must handle the issue and take corrective action for recovery.

1. E) Fatal Watcher

Fatal is reserved for special exceptions/conditions for which it is imperative that you quickly identify these events. This is not a good state for the overall processing to be in, as it means a specific step in the planned processing has not completed as expected. This means the ecosystem must now handle the issue and take corrective action for recovery.

2. Basic Logging –

This logging enables you to log everything that occurs in your data science processing to a central file, for each run of the process.

3. Process Tracking –

a controlled systematic and independent examination of the process for the hardware logging. use the logs for your cause-and-effect analysis system.

4. Data Provenance -

Keep records for every data entity in the data lake, by tracking it through all the transformations in the system. This ensures that you can reproduce the data, if needed, in the future and supplies a detailed history of the data’s source in the system.

5. Data Lineage -

Keep records of every change that happens to the individual data values in the data lake. This enables you to know what the exact value of any data record was in the past. It is normally achieved by a valid-from and valid-to audit entry for each data set in the data science environment.

Balance

The balance sublayer ensures that the ecosystem is balanced across the accessible processing capability or has the capability to top up capability during periods of extreme processing. The processing on-demand capability of a cloud ecosystem is highly desirable for this purpose. By using the audit trail, it is possible to adapt to changing requirements and forecast what you will require to complete the schedule of work you submitted to the ecosystem.

Describe the Yoke solution used in the Control sublayer.

Control

The control sublayer controls the execution of the current active data science. The control elements are a combination of the control element within the Data Science Technology Stack's individual tools plus a custom interface to control the overarching work. The control sublayer also ensures that when processing experiences an error, it can try a recovery, as per your requirements, or schedule a clean-up utility to undo the error. The cause-and-effect analysis system is the core data source for the distributed control system in the ecosystem.

Create an independent process that is created solely to monitor a specific portion of the data processing ecosystem control. So, the control system consists of a series of yokes at each control point that uses Kafka messaging to communicate the control requests. The yoke then converts the requests into a process to execute and manage in the ecosystem. The yoke system ensures that the distributed tasks are completed, even if it loses contact with the central services. The yoke solution is extremely useful in the Internet of things environment, as you are not always able to communicate directly with the data source.

Yoke Solution

The yoke solution is a custom design. Kafka provides a publish-subscribe solution that can handle all activity-stream data and processing. The Kafka environment enables you to send messages between producers and consumers that enable you to transfer control between different parts of your ecosystem while ensuring a stable process.

Producer

The producer is the part of the system that generates the requests for data science processing, by creating structures messages for each type of data science process it requires. The producer is the end point of the pipeline that loads messages into Kafka.

Consumer

The consumer is the part of the process that takes in messages and organizes them for processing by the data science tools. The consumer is the end point of the pipeline that offloads the messages from Kafka.

Cause-and-Effect Analysis System

The cause-and-effect analysis system is the part of the ecosystem that collects all the logs, schedules, and other ecosystem-related information and enables data scientists to evaluate the quality of their system.

Functional Layer

The functional layer of the data science ecosystem is the largest and most essential layer for programming and modeling. Any data science project must have processing elements in this layer. The layer performs all the data processing chains for the practical data science. The functional layer of the data science ecosystem is the largest and most essential layer for programming and modeling. The functional layer is the part of the ecosystem that runs the comprehensive data science ecosystem.

Explain the five fundamental steps that form the core of the data science process.

Data Science Process

Following are the five fundamental data science process steps.

- Start with a What-if Question

Decide what you want to know, even if it is only the subset of the data lake you want to use for your data science, which is a good start.

- Take a Guess at a Potential Pattern

Use your experience or insights to guess a pattern you want to discover, to uncover additional insights from the data you already have

- Gather Observations and Use Them to Produce a Hypothesis

A hypothesis, it is a proposed explanation, prepared on the basis of limited evidence, as a starting point for further investigation.

- Use Real-World Evidence to Verify the Hypothesis

Now, we verify our hypothesis with real-world evidence

- Collaborate Promptly and Regularly with Customers and Subject Matter Experts As You Gain Insights

It consists of several structures, as follows:

• *Data schemas and data formats*: Functional data schemas and data formats deploy onto the data lake's raw data, to perform the required schema-on-query via the functional layer.

• *Data models*: These form the basis for future processing to enhance the processing capabilities of the data lake, by storing already processed data sources for future use by other processes against the data lake.

• *Processing algorithms*: The functional processing is performed via a series of well-designed algorithms across the processing chain.

• *Provisioning of infrastructure*: The functional infrastructure provision enables the framework to add processing capability to the ecosystem, using technology such as Apache Mesos, which enables the dynamic previsioning of processing work cells.

Write a short note on the six super steps for processing the data.

Processing algorithm is spread across six supersteps of processing, as follows:

1. *Retrieve*: This super step contains all the processing chains for retrieving data from the raw data lake via a more structured format.

The Retrieve superstep is a practical method for importing completely into the processing ecosystem a data lake consisting of various external data sources.

A company's data lake covers all data that your business is authorized to process, to attain an improved profitability of your business's core accomplishments.

2. *Assess*: This superstep contains all the processing chains for quality assurance and additional data enhancements.

This superstep is about how to assess your data science data for invalid or erroneous data values. Data quality refers to the condition of a set of qualitative or quantitative variables.

3. *Process*: This superstep contains all the processing chains for building the data vault.

The Process superstep adapts the assess results of the retrieve versions of the data sources into a highly structured data vault that will form the basic data structure for the rest of the data science steps.
4. *Transform*: This superstep contains all the processing chains for building the data warehouse.
The Transform superstep allows you, as a data scientist, to take data from the data vault and formulate answers to questions raised by your investigations. The transformation step is the data science process that converts results into insights. It takes standard data science techniques and methods to attain insight and knowledge about the data that then can be transformed into actionable decisions.
5. *Organize*: This superstep contains all the processing chains for building the data marts.
The Organize superstep takes the complete data warehouse you built at the end of the Transform superstep and subsections it into business-specific data marts. A data mart is the access layer of the data warehouse environment built to expose data to the users. The data mart is a subset of the data warehouse and is generally oriented to a specific business group.
6. *Report*: This superstep contains all the processing chains for building virtualization and reporting the actionable knowledge.
The Report superstep is the step in the ecosystem that enhances the data science findings with the art of storytelling and data visualization. The most important step in any analysis is the summary of the results.

Retrieve Superstep

The Retrieve superstep is a practical method for importing completely into the processing ecosystem a data lake consisting of various external data sources.

Data Lakes

If you think of a datamart as a store of bottled water—cleansed and packaged and structured for easy consumption—the data lake is a large body of water in a more natural state.
A company's data lake covers all data that your business is authorized to process, to attain an improved profitability of your business's core accomplishments. The data lake is the complete data world your company interacts with during its business life span. In simple terms, if you generate data or consume data to perform your business tasks, that data is in your company's data lake.

Explain the steps to avoid a data swamp.

Data Swamps

Data swamps are simply data lakes that are not managed. They are not to be feared. They need to be tamed. Following are four critical steps to avoid a data swamp.
1. Start with Concrete Business Questions
Simply dumping a horde of data into a data lake, with no tangible purpose in mind, will result in a big business risk. The data lake must be enabled to collect the data required to answer your business questions.
2. Data Governance
The role of data governance, data access, and data security does not go away with the volume of data in the data lake.

2.a Data Source Catalog - note the following for the data you process.

- *Unique data catalog number*
- *Short description (keep it under 100 characters)*
- *Long description (keep it as complete as possible)*

- *Contact information for external data source*
- *Expected frequency*: Irregular i.e., no fixed frequency, also known as ad hoc, every minute, hourly, daily, weekly, monthly, or yearly.
- *Internal business purpose*

2.b Business Glossary - The business glossary maps the data-source fields and classifies them into respective lines of business

3. Analytical Model Usage

Data tagged in respective analytical models define the profile of the data that requires loading and guides the data scientist to what additional processing is required.

3.a) Data Field Name Verification - use this to validate and verify the data field's names in the retrieve processing in an easy manner.

3.b) Unique Identifier of Each Data Entry - keep track of all data entries in an effective manner.

3.c) Data Type of Each Data Column

3.d) Histograms of Each Column - always generate a histogram across every column, to determine the spread of the data value.

3.e) Minimum Value- Determine the minimum value in a specific column.

3.f) Maximum Value - Determine the maximum value in a specific column.

3.g) Mean - If the column is numeric in nature, determine the average value in a specific column.

3.h) Median - Determine the value that splits the data set into two parts in a specific column

3.i) Mode - Determine the value that appears most in a specific column.

3.j) Range - For numeric values, you determine the range of the values by taking the maximum value and subtracting the minimum value.

3.k) Quartiles - Quartiles are the base values dividing a data set into quarters. Simply sort the data column and split it into four groups that are of four equal parts.

3.l) Standard Deviation

3.m) Skewness - Skewness describes the shape or profile of the distribution of the data in the column.

3.n) Missing or Unknown Values - Identify if you have missing or unknown values in the data sets.

4. Data Quality

Data quality can cause the invalidation of a complete data set, if not dealt with correctly.

Audit and Version Management

You must always report the following: • Who used the process? • When was it used? • Which version of code was used?

Explain how the data science ecosystem can be connected to different data sources in Python.

Connecting to Other Data Sources

The following are a few common data stores

SQLite

This requires a package named sqlite3.

Microsoft SQL Server

Microsoft SQL server is common in companies, and this connector supports your connection to the database. Via the direct connection, use

```
from sqlalchemy import create_engine
engine = create_engine('mssql+pymssql://scott:tiger@hostname:port/vermeulen')
```

Oracle

Oracle is a common database storage option in bigger companies. It enables you to load data from the following data source with ease:

```
from sqlalchemy import create_engine
engine = create_engine('oracle://andre:vermeulen@127.0.0.1:1521/vermeulen')
```

MySQL

MySQL is widely used by lots of companies for storing data. This opens that data to your data science with the change of a simple connection string.

There are two options. For direct connect to the database, use

```
from sqlalchemy import create_engine
engine = create_engine('mysql+mysqldb://scott:tiger@localhost/vermeulen')
```

Apache Cassandra

Cassandra is becoming a widely distributed database engine in the corporate world.

To access it, use the Python package cassandra.

```
from cassandra.cluster import Cluster
cluster = Cluster()
session = cluster.connect(‘vermeulen’)
```

Apache Hadoop

Hadoop is one of the most successful data lake ecosystems in highly distributed data Science. The pydoop package includes a Python MapReduce and HDFS API for Hadoop.

Pydoop

is a Python interface to Hadoop that allows you to write MapReduce applications and interact with HDFS in pure Python

UNIT 3

What are the different ways to handle errors in the Assess super step?

Assess Superstep

This superstep is about how to assess your data science data for invalid or erroneous data values. Data quality refers to the condition of a set of qualitative or quantitative variables. Data quality is a multidimensional measurement of the acceptability of specific data sets. In business, data quality is measured to determine whether data can be used as a basis for reliable intelligence extraction for supporting organizational decisions.

Errors

- Accept the Error

If it falls within an acceptable standard (i.e., West Street instead of West St.), we can decide to accept it and move on to the next data entry. Take note that if you accept the error, you will affect data science techniques and algorithms that perform classification, such as binning, regression, clustering, and decision trees.

- Reject the Error

Occasionally, predominantly with first-time data imports, the information is so severely damaged that it is better to simply delete the data entry methodically and not try to correct it. Take note: Removing data is a last resort.

- Correct the Error

This is the option that a major part of the assess step is dedicated to. Spelling mistakes in customer names, addresses, and locations are a common source of errors, which are methodically corrected.

- Create a Default Value

This is an option that is commonly used in companies. Most system developers assume that if the business doesn't enter the value, they should enter a default value.

Analysis of Data

Completeness - to ensure that the data source is fit to progress to the next phase of analysis

Uniqueness - evaluate how unique the specific value is, in comparison to the rest of the data in that field

Timeliness - Record the impact of the date and time on the data source

Validity - Validity is tested against known and approved standards. It is recorded as a percentage of nonconformance against the standard

Accuracy - Accuracy is a measure of the data against the real-world person or object that is recorded in the data source

Consistency - Measure how data changes load after load

Write a short note on the Missing value treatment.

Why Missing Value Treatment Is Required

Explain with notes on the data traceability matrix why there is missing data in the data lake. Remember: Every inconsistency in the data lake is conceivably the missing insight your customer is seeking from you as a data scientist. So, find them and explain them.

Why Data Has Missing Values

The use of cause-and-effect fishbone diagrams will assist you to resolve those questions.

The following are common reasons for missing data:

- Data fields renamed during upgrades
- Migration processes from old systems to new systems where mappings were incomplete
- Incorrect tables supplied in loading specifications by subject-matter expert
- Data simply not recorded, as it was not available
- Legal reasons, owing to data protection legislation
- Someone else's "bad" data science. People and projects make mistakes, and you will have to fix their errors in your own data science.

Practical Actions for Missing Values

The Python package pandas enables several automatic error-management features.

Following are four basic processing concepts

- Drop the Columns Where All Elements Are Missing Values

```
TestData=RawData.dropna(axis=1, how='all')
```

- Drop the Columns Where Any of the Elements Is Missing Values

```
TestData=RawData.dropna(axis=1, how='any')
```

- Keep Only the Rows That Contain a Maximum of Two Missing Values

```
TestData=RawData.dropna(thresh=2)
```

- Fill All Missing Values with the Mean, Median, Mode, Minimum, and Maximum of the Particular Numeric Column

```
TestData=RawData.fillna(RawData.mean())
TestData=RawData.fillna(RawData.median())
TestData=RawData.fillna(RawData.mode())
TestData=RawData.fillna(RawData.min())
TestData=RawData.fillna(RawData.max())
```