Introduction to Data Science Day 1

Zeham Management Technologies BootCamp
by SDAIA

July 21ˢᵗ, 2024

SDAIA
الهيئة السعودية للبيانات
والذكاء الاصطناعي
Saudi Data & AI Authority

Objectives

By the end of this module, trainees will have a comprehensive understanding of:

- ✓ Install and build your own repository.

- ✓ Learn Imputers and Encoders

- ✓ Make preprocessing Pipelines.

- ✓ Train and validate the model.

- ✓ Evaluate and save your model.

- ✓ Build an API and connect it with your Machine Learning code.

Agenda

- Information Lifecycle
- Analytical Problems
- Events & Characteristics in Business Analytics
- Data Capture
- Data Quality, Governance and Privacy
- Environment Setup

Information Lifecycle

Information Lifecycle

1. Data Generation:

- **Events in the Real World**
  - Data is generated from various real-world events and activities, such as customer transactions, social media interactions, sensor readings, and more.
  - Data can be collected manually or automatically using various tools and technologies. For example, point-of-sale systems collect transaction data, while IoT devices gather sensor data.

Source

Information Lifecycle

2. Data Acquisition and Ingestion:

- **Data Capture**
  - Data is captured through different methods, including manual entry, automated systems, web scraping, and API integrations.
  - Data can be structured (databases, spreadsheets), semi-structured (XML, JSON), or unstructured (text, images, videos)

Source

Information Lifecycle

2. Data Acquisition and Ingestion:

▪ **Data Ingestion**

- ETL (Extract, Transform, Load) tools are used to move data from its source into a data warehouse or data lake.
- Data is extracted from various sources, transformed into a consistent format, and loaded into storage systems for further processing.
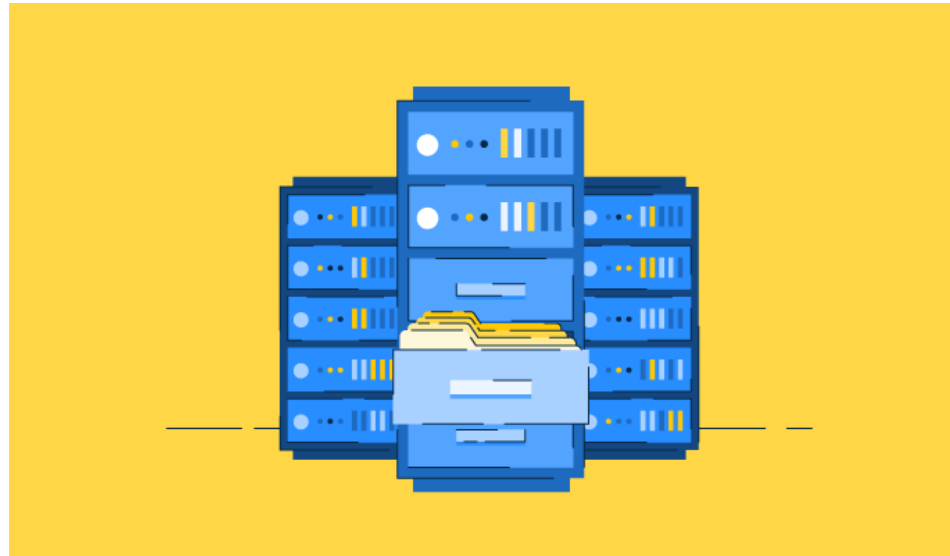
[Source](#)

Information Lifecycle

3. Data Storage:

- **Storage Systems**
    - **Databases**: Structured data is stored in relational databases (SQL) or NoSQL databases.
    - **Data Lakes**: Large volumes of raw, unstructured data are stored in data lakes, which allow for flexible data exploration and analysis.
    - **Cloud Storage**: Data can also be stored in cloud-based systems, offering scalability and accessibility.
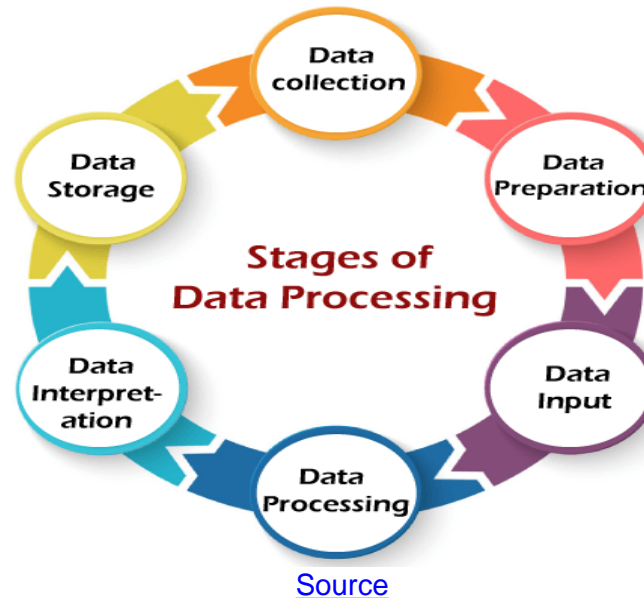

Source

Information Lifecycle

4. Data Processing:

▪ **Data Cleaning**

- **Processes**: Data is cleaned to remove errors, duplicates, and inconsistencies. This ensures the quality and reliability of the data.
- **Tools**: Various tools and programming languages (e.g., Python, R) are used for data cleaning and preprocessing.
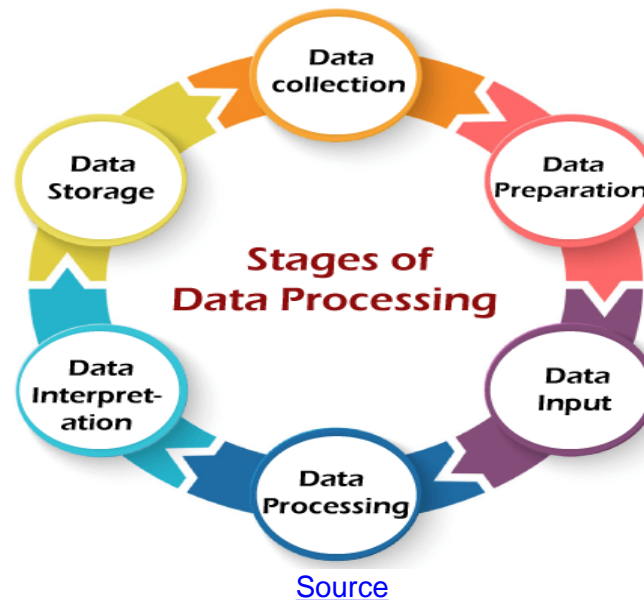

Source

Information Lifecycle

4. Data Processing:

▪ **Data Transformation**

- **Normalization**: Data is normalized to a common format, making it easier to analyze.
- **Aggregation**: Data is aggregated to provide summary insights and support analysis at different levels of granularity.

Source

Information Lifecycle

5. Data Analysis:

▪ **Exploratory Data Analysis (EDA)**

- **Techniques**: EDA involves using statistical and graphical techniques to explore and understand the data.
- **Tools**: Tools like Pandas, Matplotlib, and Seaborn in Python, or Power BI and Tableau for visualization, are commonly used.

Source

Information Lifecycle

5. Data Analysis:

▪ **Advanced Analytics**

- **Machine Learning**: Algorithms and models are applied to the data to identify patterns, make predictions, and derive insights.
- **Statistical Analysis**: Statistical methods are used to test hypotheses and make inferences about the data.

Source

Information Lifecycle

6. Data Interpretation and insights:

- **Interpretation**
    - **Business Context**: Data insights are interpreted in the context of the business, considering industry knowledge, business objectives, and market conditions.
    - **Visualization**: Data visualization tools (e.g., dashboards, reports) help in presenting insights in a clear and actionable manner.
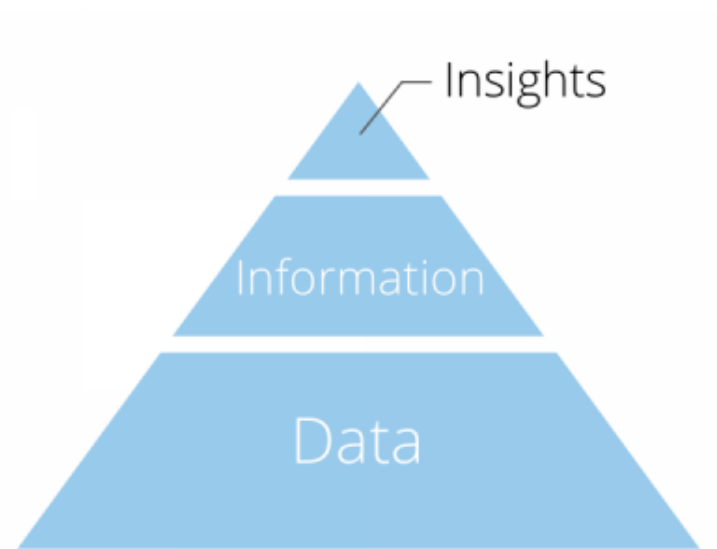
Data Interpretation

[Source](#)

Information Lifecycle

6. Data Interpretation and insights:

- **Insights**
  - **Key Findings**: Critical insights and trends are identified from the analysis.
  - **Decision Support**: Insights support decision-making by providing evidence-based recommendations.

[Source](#)

Information Lifecycle

7. Business actions:

▪ **Strategy and Planning**

- **Actionable Insights**: Insights are translated into strategic actions, such as optimizing operations, improving customer experiences, or developing new products.
- **Implementation**: Plans are developed and implemented based on the insights derived from the data.

Source

Summary

- The lifecycle of information from real-world events to business actions involves several key stages:

- data generation, acquisition, storage, processing, analysis, interpretation, and action.

- Each stage is crucial for transforming raw data into valuable insights that drive informed business decisions and actions.

Analytical Problems

Analytical Problems

1. Data Generation:

▪ **Questions**

- What are the key events or activities generating data?
- What data sources are available?
- How frequently is data generated?

[Source](#)

Analytical Problems

1. Data Generation:

- **Considerations**
    - Identify the primary sources of data relevant to the business problem.
    - Understand the nature and format of the data being generated.
    - Ensure data collection methods are reliable and comprehensive.

Source

Analytical Problems

2. Data Acquisition and Ingestion:

▪ **Questions**

- How will data be captured from various sources?
- What are the best tools and technologies for data ingestion?
- How will data be integrated from different sources?

[Source](#)

Analytical Problems

2. Data Acquisition and Ingestion:

▪ **Considerations**
- Choose appropriate methods for capturing data (e.g., APIs, manual entry).
- Select ETL tools that can handle the volume and complexity of the data.
- Plan for data integration to ensure a unified dataset.

Source

Analytical Problems

3. Data Storage:

▪ **Questions**

- Where will the data be stored?
- What are the storage requirements in terms of volume, scalability, and accessibility?
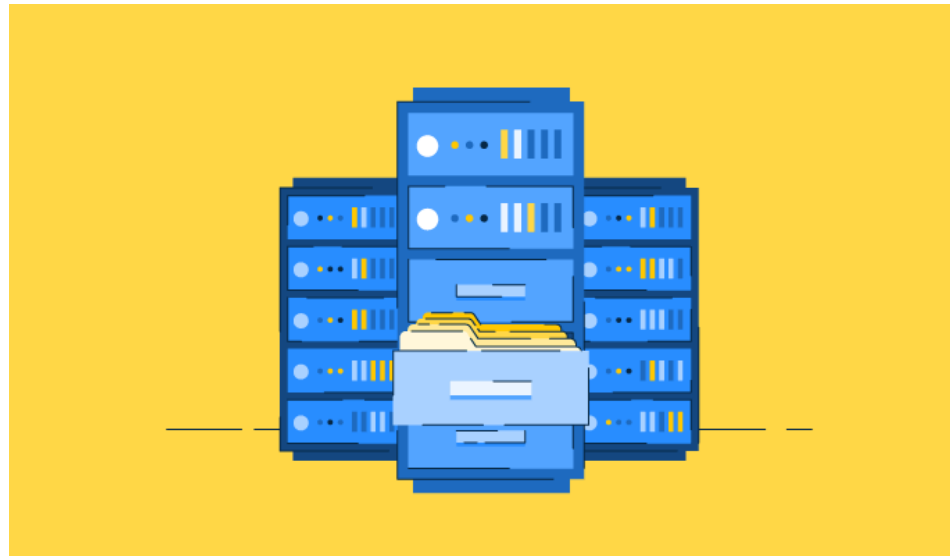- How will data security and privacy be ensured?
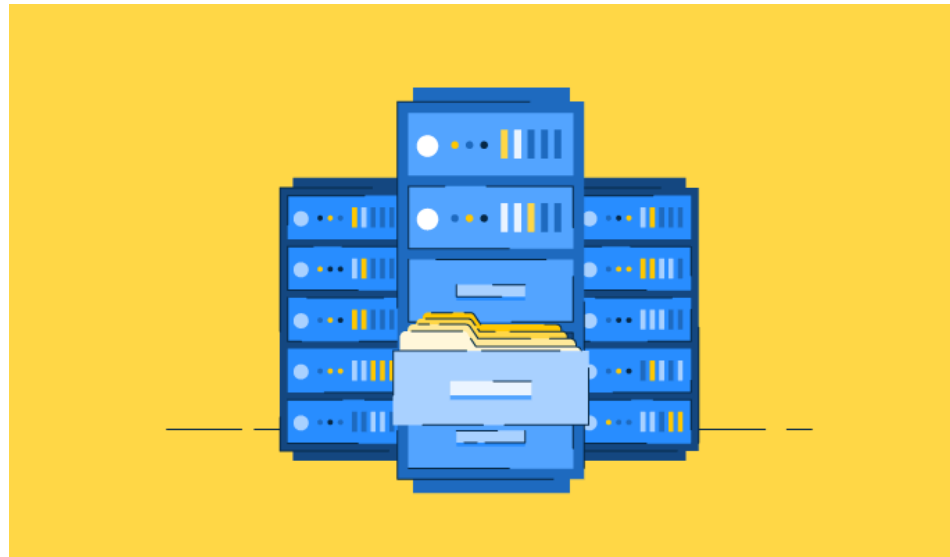
Source

Analytical Problems

3. Data Storage:

- **Considerations**
  - Decide between relational databases, NoSQL databases, and data lakes based on the data type and usage.
  - Implement security measures to protect sensitive data.
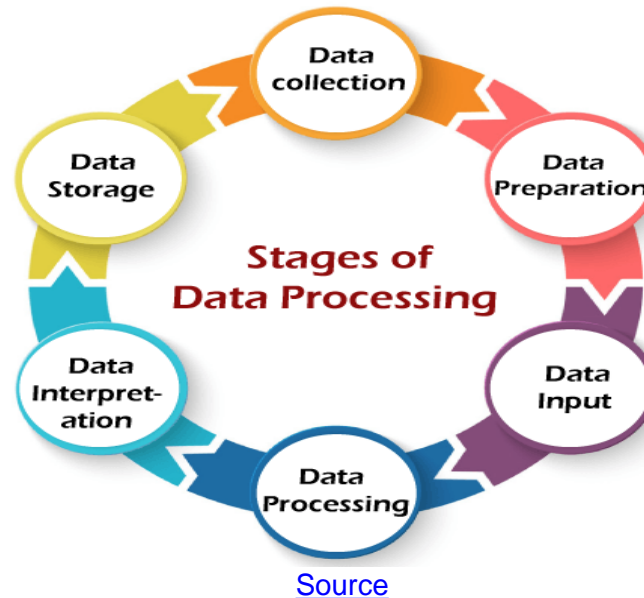 - Ensure data storage solutions are scalable to accommodate growth.

Source

Analytical Problems

4. Data Processing:

- **Questions**
  - What data cleaning and preprocessing steps are required?
  - How will data be transformed to meet analysis requirements?
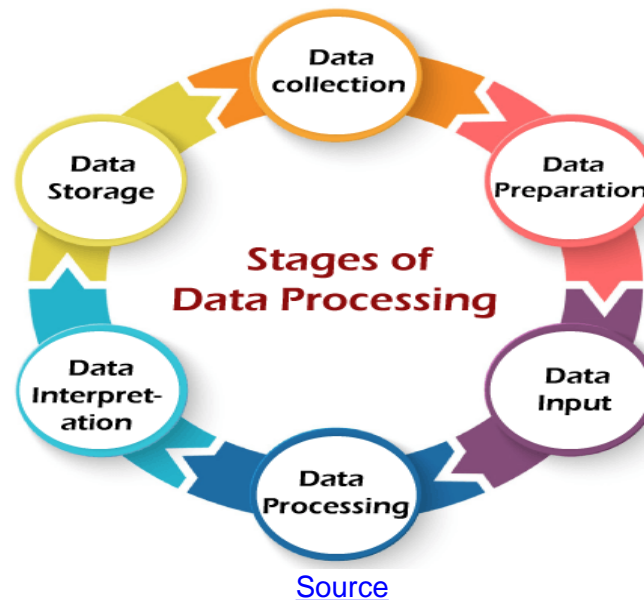 - What tools and technologies will be used for processing?

[Source](#)

Analytical Problems

4. Data Processing:

- **Considerations**
  - Develop a data cleaning pipeline to handle missing values, duplicates, and errors.
  - Normalize and aggregate data to prepare it for analysis.
  - Use appropriate tools and programming languages for data processing (e.g., Python, R).

Source

Analytical Problems

5. Data Analysis:

▪ **Questions**

- What are the key business questions to be answered?
- What analytical techniques will be used (e.g., statistical analysis, machine learning)?
- How will results be validated and interpreted?

Source

Analytical Problems

5. Data Analysis:

- **Considerations**
    - Define clear objectives and hypotheses for the analysis.
    - Select analytical methods that align with the business questions.
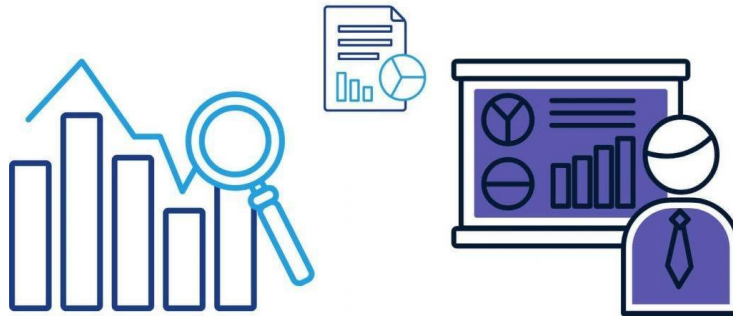   - Validate models and analysis results to ensure accuracy and reliability.

Source

Analytical Problems

6. Data Interpretation and insights:

▪ **Questions**

- What insights can be derived from the analysis?
- How do these insights align with business objectives?
- What visualizations will best communicate the findings?

Data Interpretation

Source

Analytical Problems

6. Data Interpretation and insights:

- **Considerations**
  - Interpret results in the context of business goals and market conditions.
  - Use data visualization tools to create clear and actionable reports.
  - Ensure insights are understandable and relevant to stakeholders.

Data Interpretation

Source

Analytical Problems

7. Business actions:

▪ **Questions**

- What actions should be taken based on the insights?
- How will these actions be implemented and monitored?
- What are the expected outcomes and KPIs?

Source

Analytical Problems

7. Business actions:

▪ **Considerations**
- Translate insights into strategic and tactical actions.
- Develop implementation plans and assign responsibilities.
- Monitor the impact of actions and adjust strategies as needed.

[Source](#)

Summary

- To effectively think about analytical problems in the context of the information lifecycle:

1. **Understand the Data**: Identify and understand the sources and nature of the data.
2. **Plan Data Acquisition**: Ensure robust data capture and integration methods.
3. **Secure Storage**: Choose appropriate storage solutions with security measures.
4. **Process Data**: Clean and transform data to prepare it for analysis.
5. **Analyze Data**: Use relevant analytical techniques to extract insights.
6. **Interpret Results**: Contextualize insights and visualize findings.
7. **Drive Actions**: Translate insights into business actions and monitor outcomes.

- By systematically addressing each stage, you can ensure that analytical problems are approached comprehensively and that the resulting insights lead to effective business actions.

Events & Characteristics in Business Analytics

Events & Characteristics

- In business analytics, various types of events and their characteristics are frequently used to derive insights and inform decision-making.

- Understanding these events and characteristics helps in identifying key metrics, trends, and patterns that impact business performance.

- Here are some common types of events and their relevant characteristics

Types of Events in Business Analytics

1. Customer Transaction:

- **Characteristics**: Transaction ID, date and time, customer ID, product ID, quantity, price, discount, payment method.
- **Examples**: Purchases, returns, exchanges.

Source

Types of Events in Business Analytics

2. Customer Interaction:

- **Characteristics**: Interaction type (call, email, chat), date and time, customer ID, agent ID, interaction duration, resolution status.
- **Examples**: Customer service calls, support tickets, chat sessions.
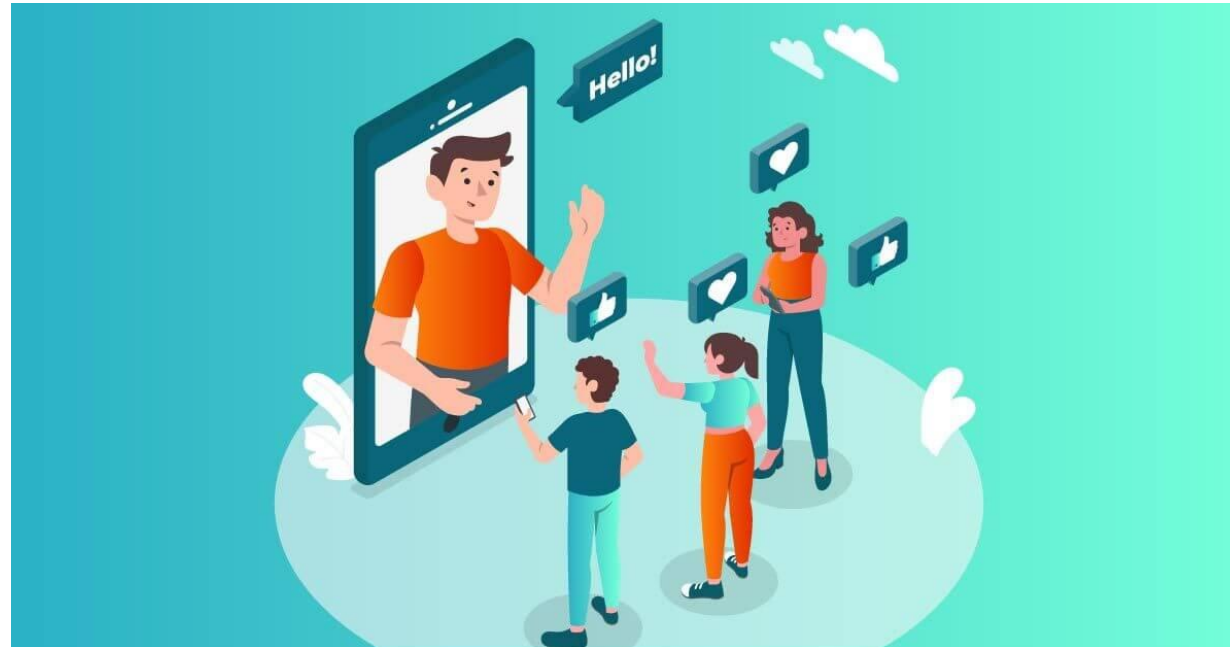
Source

Types of Events in Business Analytics

3. Sales and Marketing Campaigns:

- **Characteristics**: Campaign ID, date and time, channel, audience segment, impressions, clicks, conversions, cost.
- **Examples**: Email campaigns, social media ads, TV commercials.

Types of Events in Business Analytics

4. Employees and activities:

- **Characteristics**: Employee ID, activity type, date and time, department, project ID, hours worked, performance metrics.
- **Examples**: Work hours, project tasks, performance reviews.

shutterstock.com · 2019396023

Characteristics and Metrics for Analysis

1. Time-based Characteristics:

- **`Timestamp`**: When the event occurred (date and time).

- **`Duration`**: Length of time the event lasted.

- **`Frequency`**: How often the event occurs.
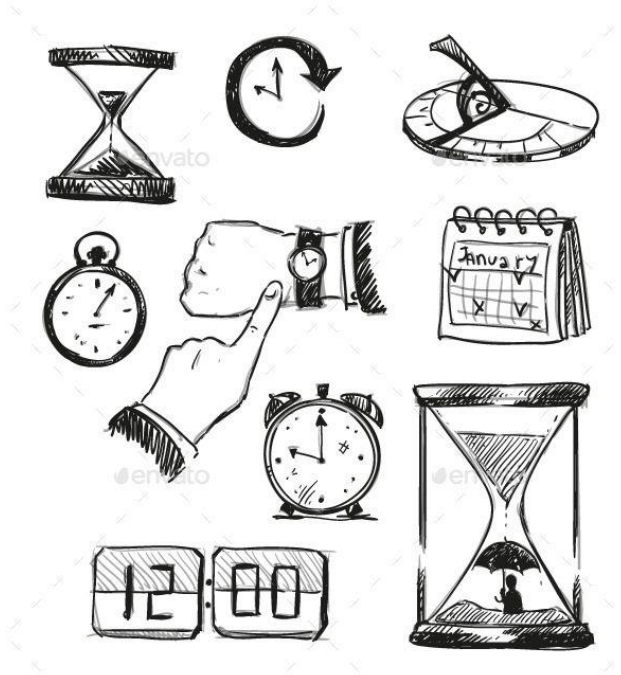
Characteristics and Metrics for Analysis

2. Entity-based Characteristics:

- **Identifiers**: Unique IDs for customers, products, transactions, employees.
- **Attributes**: Descriptive features of entities, such as age, location, and gender for customers or size, weight, and color for products.

Characteristics and Metrics for Analysis

3. Quantitative Characteristics:

- `Amounts`: Values such as sales amounts, quantities, prices.
- `Counts`: Number of occurrences, such as the number of transactions or interactions.

Source

Characteristics and Metrics for Analysis

4. Categorical Characteristics:

- **Types**: Categories or types of events, such as interaction types (call, email) or payment methods (credit card, cash).
- **Statuses**: Status of the event or entity, such as order status (pending, shipped) or resolution status (open, closed).

Yellow 20.0%
Red 30.0%
Blue 23.3%
Orange 26.7%
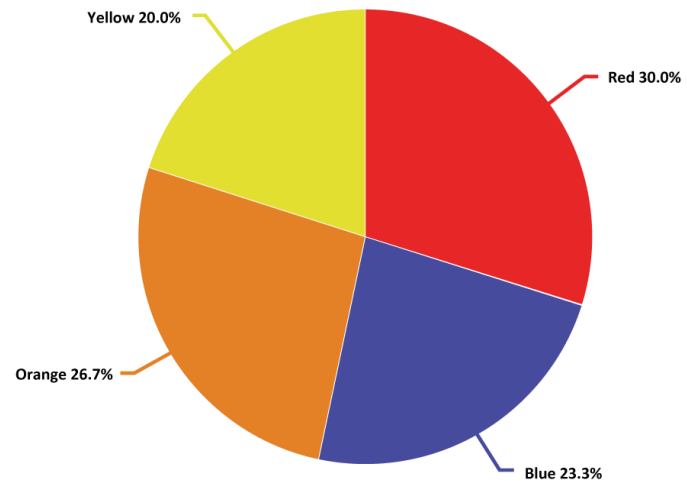
Calcworkshop.com

Source

Characteristics and Metrics for Analysis

5. Behavioral Characteristics:

- **Actions**: Specific actions taken by entities, such as clicking a link, adding to cart, or submitting a form.
- **Patterns**: Behavioral patterns, such as purchase habits or navigation paths on a website.

[Source](#)

Characteristics and Metrics for Analysis

6. Geographical Characteristics:

- **Location**: Geographic location where the event occurred, such as city, state, country.
- **Region**: Broader geographic regions, such as market areas or territories.
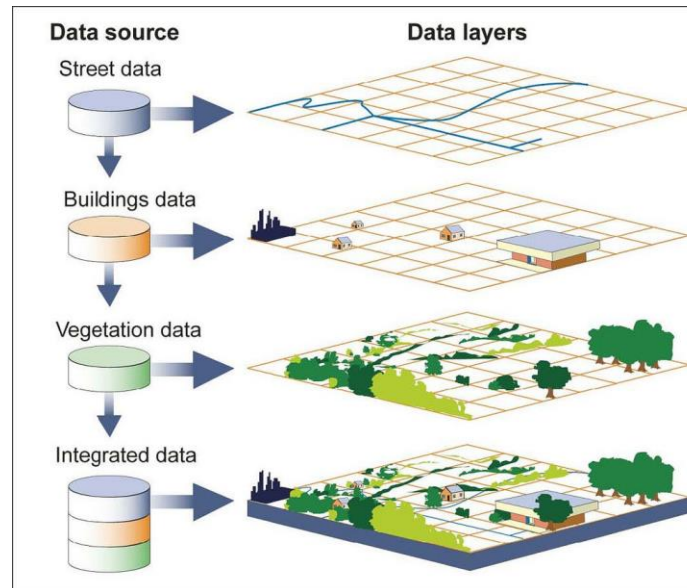
Source: GAO.

Examples in Business Analytics

1. Customer Segmentation:

- Use customer transaction and interaction data to segment customers based on purchasing behavior, demographics, and engagement levels.

2. `Sales Performance Analysis`:

- Analyze sales data to identify top-performing products, peak sales periods, and the effectiveness of marketing campaigns.

3. `Employee Performance Management`:

- Track employee activities and performance metrics to optimize workforce productivity and improve employee engagement.

Data Capture

Data Capture by Source System

- Data capture and storage are fundamental components of the data lifecycle. They involve collecting data from various sources and storing it in a way that supports easy access, processing, and analysis.
- Both traditional and emergent technologies offer different approaches to these tasks. Here's a detailed explanation

Traditional Technologies

1. Manual Data Entry:

- **Description**: Data is entered manually by users into systems such as spreadsheets or databases.
- **Tools**: Excel, data entry software.
- **Use Cases**: Small-scale data collection, surveys, forms.

2. Batch processing:

- **Description**: Data is collected over a period and processed in batches at scheduled times.
- **Tools**: Mainframe systems, batch processing software (e.g., IBM Batch Processing).
- **Use Cases**: Financial transactions, payroll processing.

Traditional Technologies

3. Relational Databases:

- **Description**: Data is captured and stored in structured tables with predefined schemas.
- **Tools**: SQL databases (e.g., MySQL, PostgreSQL, Oracle).
- **Use Cases**: Transactional systems, customer relationship management (CRM) systems.

4. Flat Files:

- **Description**: Data is stored in flat files such as CSV or text files, often extracted from other systems.
- **Tools**: CSV files, text editors.
- **Use Cases**: Simple data transfer, data exchange between systems.

Emergent Technologies

1. Real-Time Data Capture:

- **Description**: Data is captured and processed in real-time as events occur.
- **Tools**: Stream processing platforms (e.g., Apache Kafka, Apache Flink).
- **Use Cases**: Real-time analytics, monitoring systems.

2. Internet of Things (IoT):

- **Description**: Data is captured from connected devices and sensors embedded in physical objects.
- **Tools**: IoT platforms (e.g., AWS IoT, Azure IoT Hub).
- **Use Cases**: Smart homes, industrial monitoring, health tracking.

Emergent Technologies

3. APIs and Webhooks:

- **Description**: Data is captured via APIs (Application Programming Interfaces) and webhooks that allow systems to communicate and exchange data automatically.
- **Tools**: RESTful APIs, GraphQL, webhook services (e.g., Zapier, IFTTT).
- **Use Cases**: Integration between applications, real-time notifications.

4. Machine Learning and AI:

- **Description**: Data is captured through advanced algorithms that collect and process unstructured data from sources such as images, videos, and text.
- **Tools**: Computer vision systems, natural language processing (NLP) tools.
- **Use Cases**: Image recognition, sentiment analysis, autonomous vehicles.

Data Storage Using Traditional and Emergent Technologies

- **Traditional Technologies**

1. Relational Databases:
- **Description**: Structured data is stored in relational tables with predefined schemas.
- **Tools**: SQL databases (e.g., MySQL, PostgreSQL, Oracle).
- **Use Cases**: Transactional systems, ERP systems.

2. Data Warehouses:
- **Description**: Large volumes of historical data are stored for analytical purposes.
- **Tools**: Data warehousing solutions (e.g., Oracle Exadata, IBM Netezza).
- **Use Cases**: Business intelligence, reporting, data analysis.

Traditional Technologies

3. File Systems:

- **Description**: Data is stored in file systems as flat files, directories, and file hierarchies.
- **Tools**: Network Attached Storage (NAS), Storage Area Networks (SAN), Hadoop Distributed File System (HDFS).
- **Use Cases**: Document storage, data archiving.

4. Mainframe Storage:

- **Description**: Data is stored on large, powerful mainframe computers, often using hierarchical or network databases.
- **Tools**: IBM z/OS, IMS databases.
- **Use Cases**: Legacy systems, large-scale transaction processing.

Emergent Technologies

1. Cloud Storage:

- **Description**: Data is stored in scalable, flexible cloud environments.
- **Tools**: Cloud storage services (e.g., Amazon S3, Google Cloud Storage, Azure Blob Storage).
- **Use Cases**: Data backup, scalable data lakes, disaster recovery.

2. Mainframe Storage:

- **Description**: Data is stored in non-relational databases that support flexible schemas.
- **Tools**: NoSQL databases (e.g., MongoDB, Cassandra, Redis).
- **Use Cases**: Big data applications, real-time analytics, content management.

3. Data Lakes:

- **Description**: Raw, unstructured, and semi-structured data is stored in a central repository.
- **Tools**: Data lake platforms (e.g., AWS Lake Formation, Azure Data Lake, Apache Hadoop).
- **Use Cases**: Big data analytics, machine learning, data exploration.

Emergent Technologies

4. Blockchain:

- **Description**: Data is stored in decentralized, tamper-proof ledgers.
- **Tools**: Blockchain platforms (e.g., Ethereum, Hyperledger).
- **Use Cases**: Cryptocurrencies, supply chain management, secure transactions.

5. Edge Computing:

- **Description**: Data is processed and stored closer to the source of data generation (e.g., IoT devices) to reduce latency.
- **Tools**: Edge computing platforms (e.g., AWS IoT Greengrass, Azure IoT Edge).
- **Use Cases**: Autonomous vehicles, remote monitoring, real-time analytics.

Summary

- Data capture and storage have evolved significantly with the advent of new technologies.
- Traditional methods like relational databases and batch processing are still widely used, but emergent technologies like cloud storage, NoSQL databases, data lakes, and IoT are increasingly being adopted to handle the growing volume, variety, and velocity of data.
- By Using the appropriate technologies for data capture and storage, businesses can ensure efficient data management and gain valuable insights from their data.

Data Quality, Governance and Privacy

Data Quality, Governance and Privacy

- Data quality, data governance, and data privacy are critical aspects of effective data management. Here are the key ideas around each of these concepts:

- **Data Quality:**

    Data quality refers to the condition of data based on factors such as accuracy, completeness, reliability, and relevance. High-quality data is essential for making informed business decisions.

Key Aspects Data Quality

1. Accuracy:

- **Definition**: The degree to which data correctly describes the real-world object or event it represents.
- **Importance**: Inaccurate data can lead to poor decision-making and operational inefficiencies.

2. Completeness:

- **Definition**: The extent to which all required data is present.
- **Importance**: Incomplete data can result in gaps in analysis and insights, leading to incorrect conclusions.

3. Consistency:

- **Definition**: The degree to which data is uniform and free of contradictions across different data sources.
- **Importance**: Inconsistent data can cause confusion and errors in reporting and analytics.

Key Aspects Data Quality

4. Timelines:
- **Definition**: The degree to which data is up-to-date and available when needed.
- **Importance**: Timely data is crucial for real-time decision-making and maintaining competitive advantage.

5. Validity:
- `Definition`: The extent to which data conforms to the defined business rules and constraints.
- `Importance`: Valid data ensures adherence to business processes and regulatory requirements.

6. Reliability:
- `Definition`: The extent to which data is dependable and trustworthy over time.
- `Importance`: Reliable data builds confidence in the data's use for critical business operations.

Data Governance

- Data governance involves the management of data availability, usability, integrity, and security in an organization.
- It includes the processes, policies, and standards that ensure data is managed effectively.

- **Key aspects of Data Governance:**

1. Data Stewardship:

- `Definition`: The assignment of responsibility for managing and overseeing data assets to ensure their quality and integrity.
- `Importance`: Data stewards ensure that data is accurate, accessible, and protected.

2. Data Policies and Standards:

- `Definition`: Formal guidelines and rules for managing data throughout its lifecycle.
- `Importance`: Policies and standards help maintain data consistency, quality, and security.

Key Aspects of Data Governance

3. Data Ownership:

- **Definition**: The designation of individuals or teams who are accountable for specific data assets.
- **Importance**: Clear ownership ensures accountability and proper data management.

4. Data Cataloging :

- **Definition**: The creation of an inventory of data assets, including metadata that describes the data.
- **Importance**: Data catalogs improve data discovery, understanding, and governance.

5. Data Quality Management:

- **Definition**: Processes and tools used to measure, monitor, and improve data quality.
- **Importance**: Ongoing data quality management ensures that data remains accurate and reliable.

6. Compliance and Legal Requirements:

- **Definition**: Adherence to laws, regulations, and industry standards related to data.
- **Importance**: Compliance ensures that data management practices meet legal and regulatory obligations.

Data Privacy

- Data privacy involves the proper handling, processing, storage, and protection of personal information to ensure individuals' privacy rights are maintained.

Key aspects:

1. Personal data protection:
- **Definition**: Safeguarding personal information from unauthorized access, disclosure, or misuse.
- **Importance**: Protecting personal data is essential for maintaining trust and complying with privacy regulations.

2. Consent management:
- **Definition**: Obtaining and managing individuals' consent for collecting and using their personal data.
- **Importance**: Consent management ensures that data usage is transparent and respects individuals' preferences.

Key Aspects Data Privacy

3. Data minimization:

- **Definition**: Limiting the collection and storage of personal data to only what is necessary for specific purposes.
- **Importance**: Data minimization reduces the risk of data breaches and misuse.

4. Anonymization and Pseudonymization:

- **Definition**: Techniques used to protect personal data by removing or obscuring identifiers.
- **Importance**: Anonymization and pseudonymization help protect privacy while allowing data analysis.

5. Regulatory Compliance:

- **Definition**: Adhering to data privacy laws and regulations, such as GDPR, CCPA, and HIPAA.
- **Importance**: Compliance ensures legal and ethical handling of personal data and avoids penalties.

6. Data subject right:

- **Definition**: Ensuring individuals can exercise their rights regarding their personal data, such as access, correction, and deletion.
- **Importance**: Respecting data subject rights is crucial for maintaining trust and compliance with privacy laws.

Summary

- **Data Quality**: Ensuring data is accurate, complete, consistent, timely, valid, and reliable.
- **Data Governance**: Establishing processes, policies, and standards for effective data management, including stewardship, ownership, cataloging, and compliance.
- **Data Privacy**: Protecting personal information through proper handling, consent management, minimization, anonymization, and adherence to privacy regulations.

- Together, these aspects ensure that data is managed effectively, used responsibly, and protected adequately, enabling organizations to leverage data for informed decision-making while maintaining trust and compliance.

Environment Setup

What is Jupyter?

- A Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text.
- It is widely used in data science, academic research, and machine learning due to its interactive nature and versatility.
- Usually, it deals with ipynb format that contains Cells that could be either code block or markdown.

Source

What is Google Colab ?

- Google Colab, or Google Colaboratory, is a free cloud service offered by Google that allows users to write and execute Python code in a Jupyter notebook environment.
- It is particularly popular in the data science and machine learning communities because it provides free access to computational resources, including GPUs and TPUs, which can be used to accelerate the development and training of machine learning models.

What is Google Colab?

[Source](#)

What is GitHub ?

GitHub is a web-based platform that uses Git, a version control system, to help developers manage code and collaborate on projects. It provides a central place to store and share code, track changes, and manage various versions of projects.
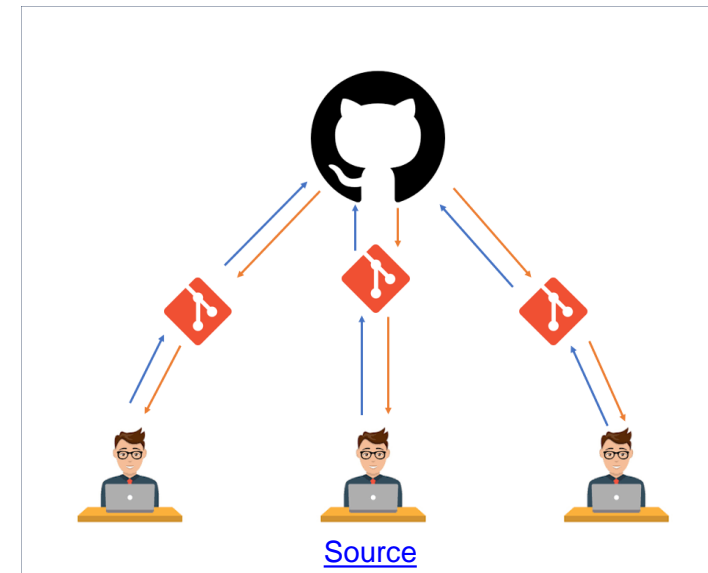
Source

GitHub (Key Concepts)

- **Repository (Repo):** A repository is a central place where all the project files, including code, documentation, and other resources, are stored. It tracks all changes made to the files.

- **Branch:** A branch is a parallel version of a repository. It allows you to work on different features or fixes separately from the main (or master) branch.

Source
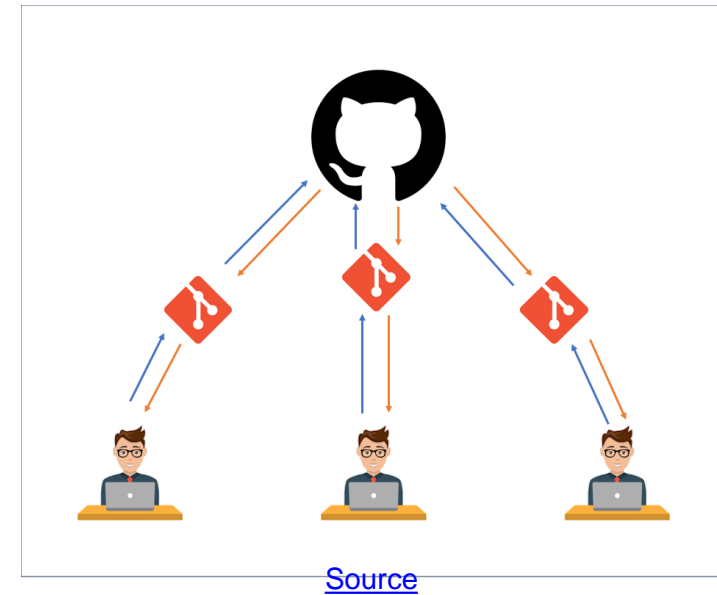
GitHub (Key Concepts)

- **Commit:** A commit is a record of changes made to the repository. It acts as a checkpoint, allowing you to revert to a previous state if needed.

- **Pull Request (PR):** A pull request is a request to merge changes from one branch to another. It facilitates code review and discussion before integrating the changes.

Source
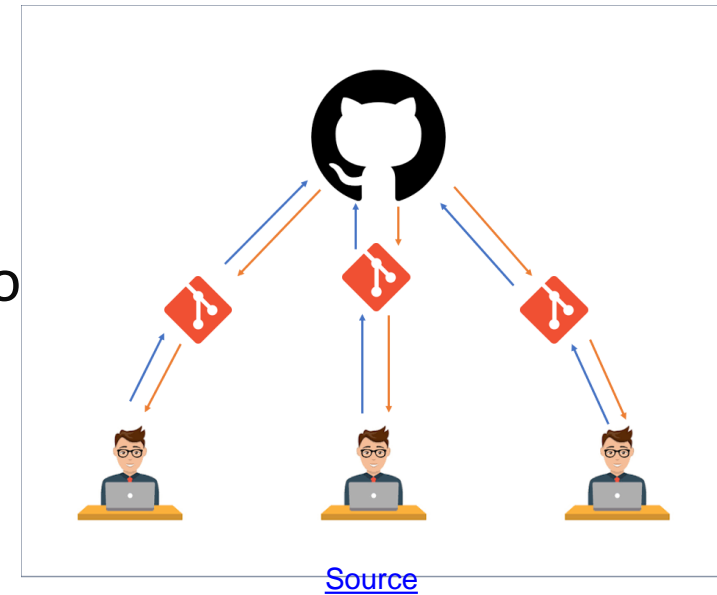
GitHub (Key Concepts)

- **Fork:** A fork is a copy of a repository that you can use to make changes without affecting the original project. It's often used to propose changes or contribute to a project you don't have write access to.

- **Clone:** Cloning is the process of creating a local copy of a repository on your machine.

Source
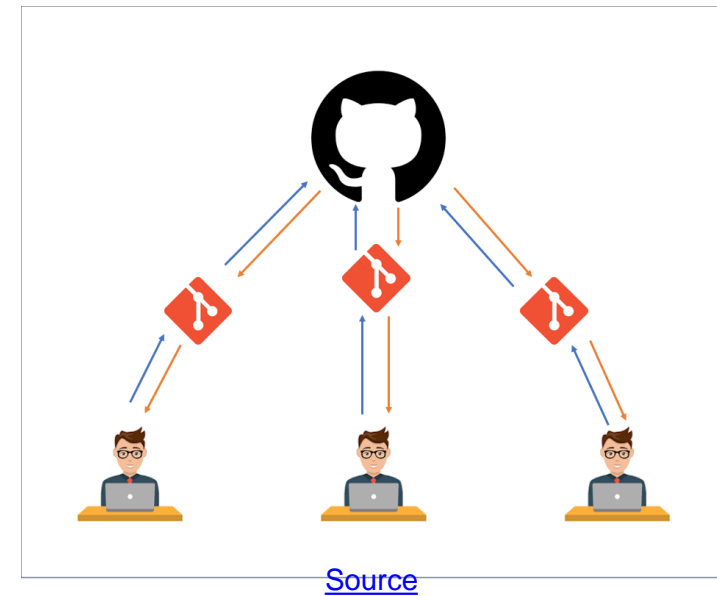
GitHub (Creating Project)

We will start off by configuring Git, open your visual studio code and put these two commands in the terminal:
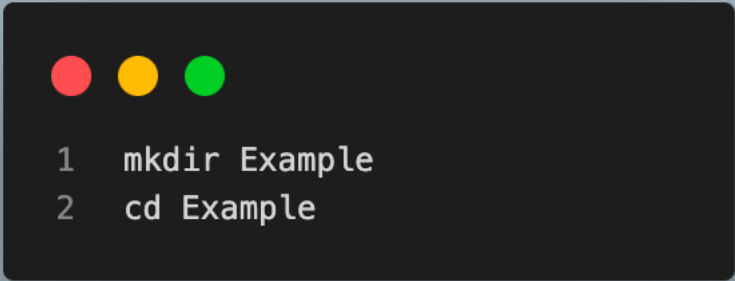
```
1  git config --global user.name "Your Name"
2  git config --global user.email "your.email@example.com"
```

GitHub (Creating Project)

We need to make a directory for the project either via VS Code and opening the directory, or by using the following command in the terminal:
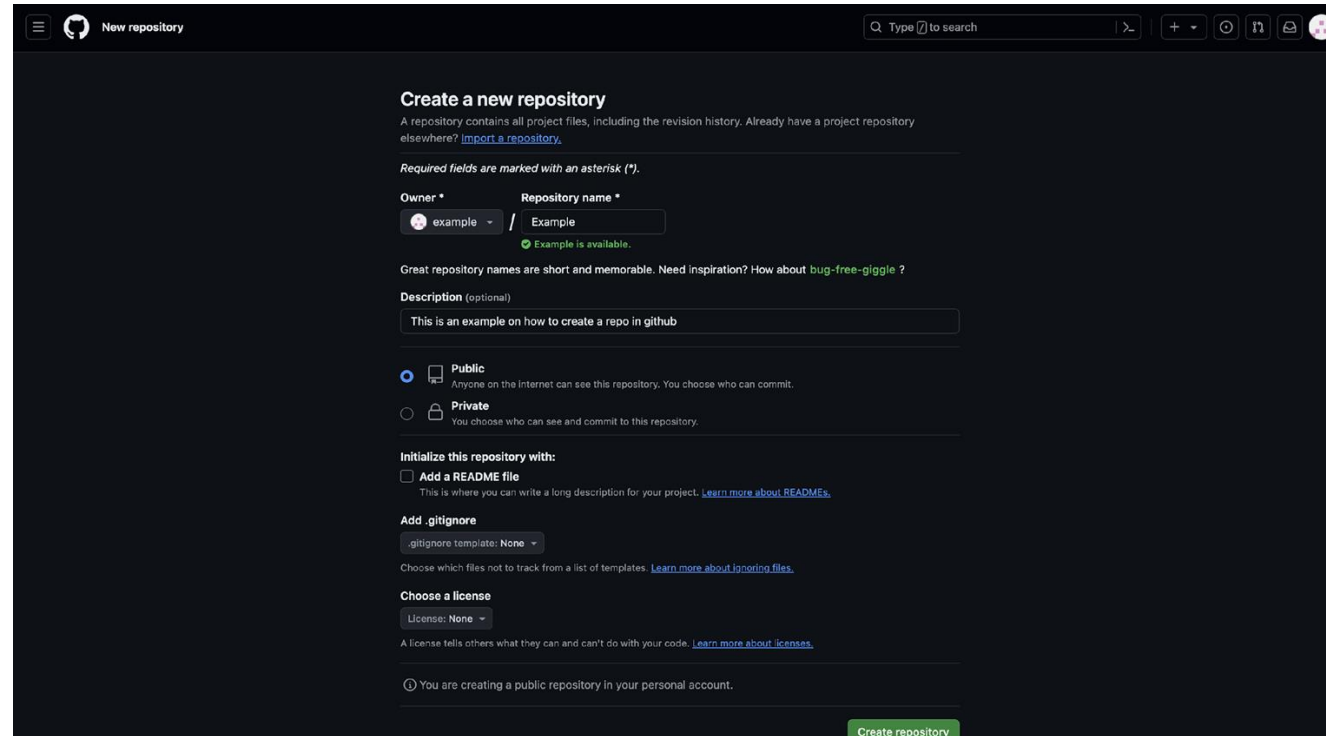
GitHub (Creating Project)

For the next step you will need to go to <u>GitHub official website</u>, log in to your account and then create the repository to clone it.

https://GitHub.com/new

GitHub (Creating Project)

For this step you will need to open VS Code terminal and write this command to initiate the local git repo '.git':

GitHub (Creating Project)

After creating the local repo, go ahead and add a README.md with your project's files at the project's local folder.
To push to GitHub the file must not be empty.
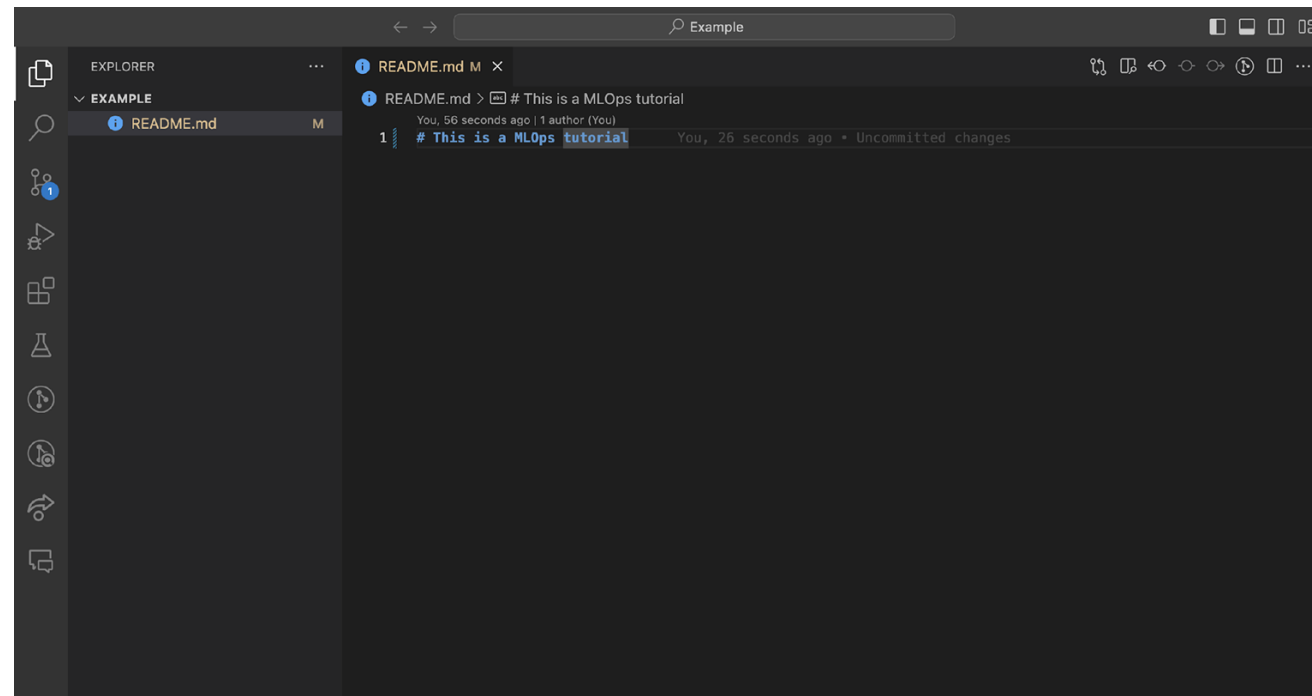
GitHub (Creating Project)

After filling the project's directory with your project's files use this command in the terminal to include all the current files:

```
1   git add .
```

GitHub (Creating Project)

Next step is to commit your first commit using the git commit along with the commit message using (-m "message")

GitHub (Creating Project)

The next step is to establish a connection between your local Git repository and the GitHub repository you created. This connection allows you to push your local changes to the remote GitHub repository.

```
1   git remote add origin https://github.com/username/example.git
```
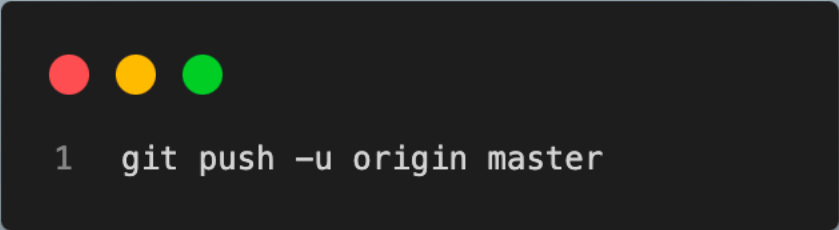
GitHub (Creating Project)

The next step is to push your local commits to the remote GitHub repository. When you run the command, you are telling Git to push the commits from your local master branch to the master branch of the remote repository (named "origin"). Here is the command:

```
1   git push -u origin master
```

GitHub (Creating Project)

Congratulations 🎉 on making your first commit!

GitHub (Creating Project)

Now if you made any changes use these commands in order:

1. Add Changes to Staging Area: Use git add to stage the changes you want to commit:
   - *git add .*
2. Commit Changes: Commit the staged changes with a commit message:
   - *git commit -m "Your commit message"*
3. Push Changes to Remote Repository:
   - *git push*

If you're pushing to the same branch on the remote repository, you can simply use git push without specifying the remote or branch name after the initial setup. If you're pushing to a different branch or remote, you would specify those in the command.

GitHub (GitHub&Git additional commands)

Add Files to Staging Area:

```
1  git add filename
2  # Or add all changes:
3  git add .
```

GitHub (GitHub&Git additional commands)

Push Changes to Remote Repository:

```
1  git push origin branch-name
```

GitHub (GitHub&Git additional commands)

Create a New Branch:


```
1    git checkout -b new-branch-name
```

GitHub (GitHub&Git additional commands)

Switch to an Existing Branch:

```
1    git checkout branch-name
```

GitHub (GitHub&Git additional commands)

Merge a Branch:

```
1  git checkout main
2  git merge branch-name
```

GitHub (GitHub&Git additional commands)

Delete a Branch:

```
1   git branch -d branch-name
```

Thank you