Big Data Management
Project P1

DTIM Research Group
Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

Contextualizing the problem

Contextualizing the problem

- Define the **domain**
  - Select a subject that excites you, but that also presents meaningful challenges

Contextualizing the problem

- Define the **domain**
    - Select a subject that excites you, but that also presents meaningful challenges

- Determine the **business value** (i.e. what problem(s) do you solve?)
    - Also, define **how** you plan on solving them
    - Ideally, through a “market analysis”
    - Important: BDM is not a modeling/analysis course

Contextualizing the problem

- Define the **domain**
  - Select a subject that excites you, but that also presents meaningful challenges

- Determine the **business value** (i.e. what problem(s) do you solve?)
  - Also, define **how** you plan on solving them
  - Ideally, through a “market analysis”
  - Important: BDM is not a modeling/analysis course

- Select the **data sources**
  - Take into consideration the **different data types** (structured, semi-structured and unstructured data)
  - Understand what type of data you need.

Contextualizing the problem - Tasks

You **need to**:

- Select an appropriate domain
- Clearly articulate what question your pipeline answers, the value it delivers and how you aim at providing this value.
- Identify and acquire data sources.

Contextualizing the problem - Tasks

You **need to**:

- Select an appropriate domain
- Clearly articulate what question your pipeline answers, the value it delivers and how you aim at providing this value.
- Identify and acquire data sources.

Important: you are allowed to generate synthetic data is you do not have the specific attributes or volume to implement a given task,.

Contextualizing the problem - Additional criteria

Factors that will **positively contribute** towards the grade:

- Employing several data sources and/or several datasets.
- Employing semi-structured or unstructured data, preferably in combination with structured data to define different data flows.
- Great contextualization of the project.
- Using “Big” Data.

Data Ingestion

- **Acquiring raw data** from various sources and **transferring** it into a system where it can be processed, stored, and analyzed.

Data Ingestion

- **Acquiring raw data** from various sources and **transferring** it into a system where it can be processed, stored, and analyzed.

- Two main types:
  - **Batch ingestion**: data is collected in fixed-size chunks (batches), often at scheduled time intervals.
  - **Streaming ingestion**: data is continuously ingested and processed as it arrives, enabling real-time analytics.

Data Ingestion

- **Acquiring raw data** from various sources and **transferring** it into a system where it can be processed, stored, and analyzed.

- Two main types:
    - **Batch ingestion**: data is collected in fixed-size chunks (batches), often at scheduled time intervals.
    - **Streaming ingestion**: data is continuously ingested and processed as it arrives, enabling real-time analytics.

- Batch ingestion can be implemented with simple API calls via Python scripts. For streaming, we recommend Apache Kafka.

Data Ingestion

- Most complex systems combine **both** ingestions systems.

Data Ingestion

- Most complex systems combine **both** ingestions systems.

- By doing so, several **processing paths** can be defined:
  - **Hot Path** (Real-Time Processing): This path is dedicated to handling time-sensitive data that requires immediate action.

Data Ingestion

- Most complex systems combine **both** ingestions systems.

- By doing so, several **processing paths** can be defined:
  - **Hot Path** (Real-Time Processing): This path is dedicated to handling time-sensitive data that requires immediate action.
  - **Cold Path** (Batch Processing): This approach handles data that do not need to be processed immediately, often obtained through scheduled batch jobs.

Data Ingestion

- Most complex systems combine **both** ingestions systems.

- By doing so, several **processing paths** can be defined:
    - **Hot Path** (Real-Time Processing): This path is dedicated to handling time-sensitive data that requires immediate action.
    - **Cold Path** (Batch Processing): This approach handles data that do not need to be processed immediately, often obtained through scheduled batch jobs.
    - **Warm Path** (Near-Real-Time Processing): Positioned between the hot and cold paths, the warm path processes data that is not necessarily urgent but still requires timely analysis.

Data Ingestion

- Most complex systems combine **both** ingestions systems.

- By doing so, several **processing paths** can be defined:
  - **Hot Path** (Real-Time Processing): This path is dedicated to handling time-sensitive data that requires immediate action.
  - **Cold Path** (Batch Processing): This approach handles data that do not need to be processed immediately, often obtained through scheduled batch jobs.
  - **Warm Path** (Near-Real-Time Processing): Positioned between the hot and cold paths, the warm path processes data that is not necessarily urgent but still requires timely analysis.

- You need to define the most appropriate paths for your case.

Data Ingestion

- It is **not acceptable** to ingest data manually.
  - You have to, at least, automate grabbing the files.

- A better alternative is to extract data from APIs, databases, web scraping, etc.
  - That is, directly interacting with **external systems**.

Data Ingestion - Tasks

You **need to**:

- Select an ingestion strategy (batch, streaming, or hybrid) for each of the data sources you have selected. This should be in accordance with the type of processing (i.e. path) that each data should undergo to fulfill the desired goals.
- Choose appropriate tools to automate and manage ingestion.
- Implement your ingestion tasks.

Data Ingestion - Tasks

You **need to**:

- Select an ingestion strategy (batch, streaming, or hybrid) for each of the data sources you have selected. This should be in accordance with the type of processing (i.e. path) that each data should undergo to fulfill the desired goals.
- Choose appropriate tools to automate and manage ingestion.
- Implement your ingestion tasks.

Important: following the previous line, you are allowed to set up APIs or streaming sources with synthetic if you do not find appropriate sources online.

Data Ingestion - Additional criteria

Factors that will **positively contribute** towards the grade:

- Implementing both batch and stream processing pipelines, which define several paths
- Ingestion systems that communicate with external sources
- Scheduling data ingestion tasks

Landing Zone

- Store all the **raw data** that was ingested into a **single repository**. Doing this provides many benefits:

Landing Zone

- Store all the **raw data** that was ingested into a **single repository**. Doing this provides many benefits:
  - The starting point of your processing paths is homogenized.
  - It ensures data availability, traceability and reproducibility.
  - It does not force data to adapt to a given model

Landing Zone

- Store all the **raw data** that was ingested into a **single repository**. Doing this provides many benefits:
  - The starting point of your processing paths is homogenized.
  - It ensures data availability, traceability and reproducibility.
  - It does not force data to adapt to a given model

- These repositories (large and heterogeneous) are known as **data lakes**. A data lake can be implemented with:

Landing Zone

- Store all the **raw data** that was ingested into a **single repository**. Doing this provides many benefits:
  - The starting point of your processing paths is homogenized.
  - It ensures data availability, traceability and reproducibility.
  - It does not force data to adapt to a given model

- These repositories (large and heterogeneous) are known as data lakes. A data lake can be implemented with:
  - Your local file system (not ideal)
  - HDFS (not ideal)
  - Cloud technologies (e.g. Amazon S3, Azure Data Lake Storage)

Landing Zone

- In recent years a new storage framework has emerged: the [data lakehouse]

Landing Zone

- In recent years a new storage framework has emerged: the data lakehouse
  - These combine the storage philosophy of data lakes (large, heterogeneous repositories) with features designed to streamline data management and create an engineering experience similar to a data warehouse.
  - Delta Lake

- We recommend you to use cloud technologies or Delta Lake.

Landing Zone

- In the Landing Zone we store the raw data without transformations. However, we do apply some **organization and structuring**.
  - If not, a data lake becomes a data swamp.

- This can be achieved by:
  - Bucketing files (e.g. in different folders).
  - Standardized naming convention.
  - Data versioning.
  - Metadata catalog.

- Additionally, “subzones” can be created.

Landing Zone - Tasks

You **need to**:

- Select an appropriate storage solution based on the nature of your data.
- Define an organizational structure for storing raw data (e.g. different folders/buckets, naming convention, data partitioning).
- Deploy the raw data layer and connect the ingestion tasks to it.

Data Ingestion - Additional criteria

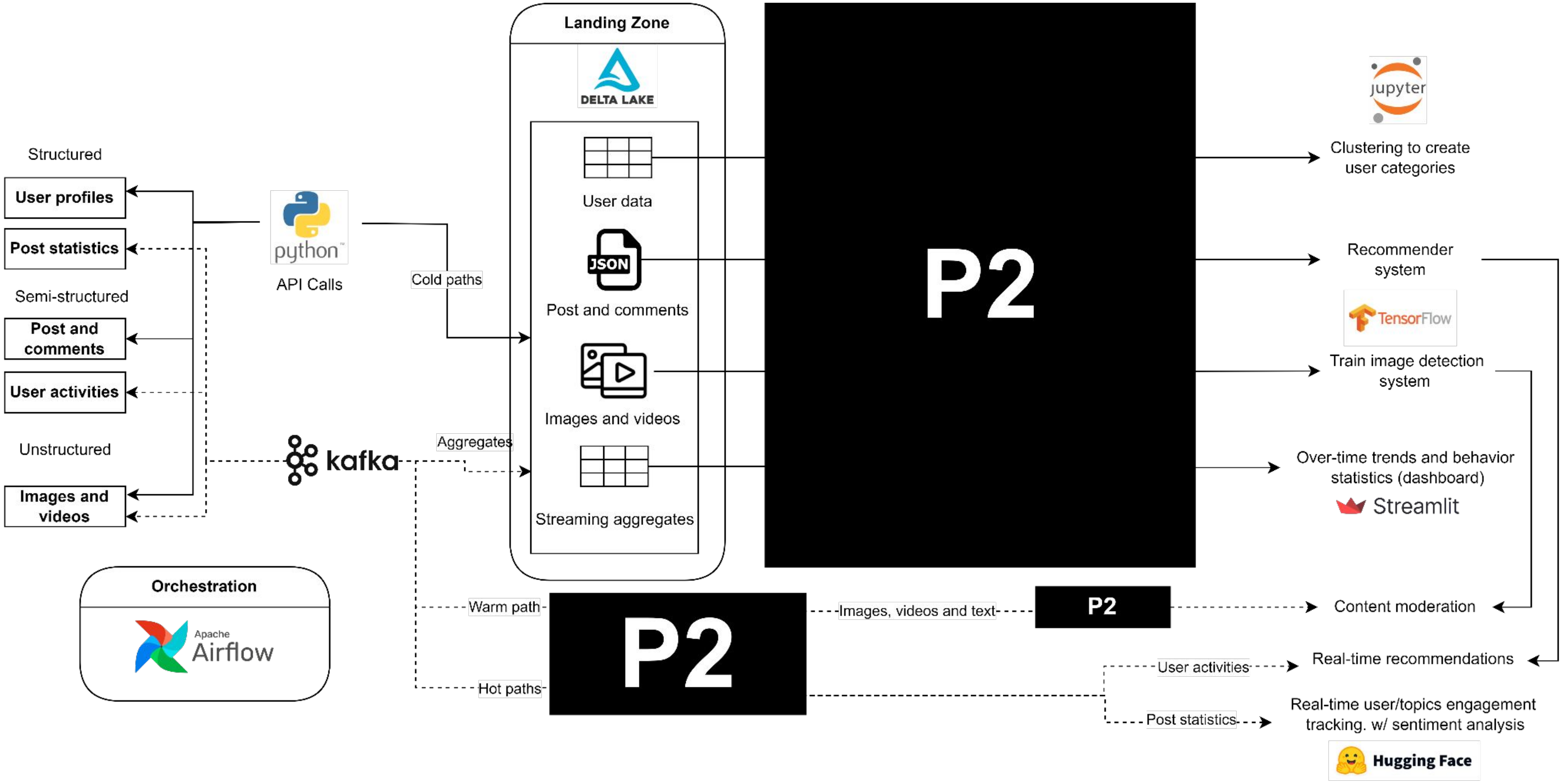
Factors that will **positively contribute** towards the grade:

- Employing cloud-based systems or data lakehouses as opposed to your local file system.
- More complex data structuring or metadata management.
-

Deliverables

Main deliverable

- It has to implement:
  - Stage 0: project context
  - **Architecture design**
  - Stage 1: data ingestion
  - Stage 2: landing zone



Structured

User profiles

Post statistics

Semi-structured

Post and comments

User activities

Unstructured

Images and videos

python

API Calls

Cold paths

kafka

Aggregates

Orchestration

Apache Airflow

Landing Zone

DELTA LAKE

User data

Post and comments

Images and videos

Streaming aggregates

P2

Warm path

Hot paths

P2

Images, videos and text

P2

jupyter

Clustering to create user categories

Recommender system

TensorFlow

Train image detection system

Over-time trends and behavior statistics (dashboard)

Streamlit

Content moderation

User activities

Real-time recommendations

Post statistics

Real-time user/topics engagement tracking. w/ sentiment analysis

Hugging Face

Follow-up deliverables

- First deliverable:
  - Project context (first version),
  - Architecture design (first version).

- Second deliverable:
  - Project context (updated).
  - Architecture design (updated).
  - Basic implementation of the data ingestion and the landing zone

Orchestration and containerization