# Python Essentials for MLOps

## Week 1: Introduction to Python

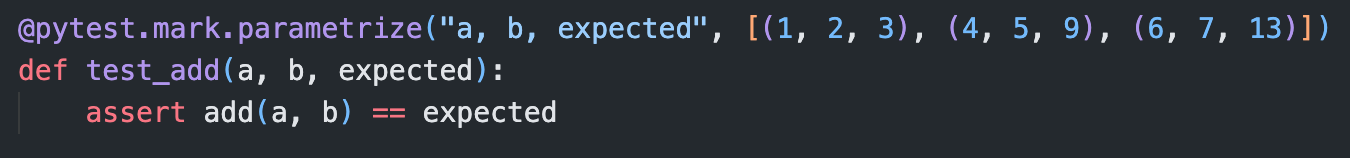
Given my expertise in python I simply did the labs and quizzes in this week.

### Week 2: Python functions and classes

Given my expertise in python I simply did the labs and quizzes in this week.

## Week 3: Testing in Python

**Pytest** is a powerful testing framework in Python that allows for simple and scalable testing. It's widely used due to its simplicity and ease of use.

* Pytest uses the built-in assert statement in Python for verification of test cases. The simplicity of using plain assert statements means there is no need to remember special assertion methods. For example, you can write a test case as simply as `assert this == that`. This will check if the condition holds true, and if not, pytest will fail the test and provide a detailed report about what was expected and what was actually received.
* Pytest supports both test functions and test classes, but test functions are often preferred for their modularity and simplicity. A test function in pytest is a simple function that starts with the word 'test'. Pytest identifies these functions as tests and runs them accordingly.
* Pytest also provides powerful features for setup and teardown operations, which are necessary when tests need to be set up or cleaned up in a certain way. This can be achieved using fixtures, which are functions that pytest will run before (and sometimes after) your tests. They are used to feed some data or context to the tests.
* Another advantage of pytest is its support for parameterized testing. This allows you to run a test function multiple times with different arguments, which is useful for testing a function against multiple inputs.  
    
    
    
    
  

## Week 4: Introduction to Pandas and NumPy

**Pandas** is a powerful, open-source data analysis and manipulation library for Python. It provides data structures and functions needed to manipulate structured data, including functions for reading and writing data in a variety of formats.

**NumPy**, which stands for Numerical Python, is a fundamental package for scientific computing in Python. It provides support for arrays (including multi-dimensional arrays), along with a large collection of high-level mathematical functions to operate on these arrays.

## Week 5: Applied python for MLOps

**API (Application Programming Interface)**: An API is a set of rules and protocols for building and interacting with software applications. It defines methods and data formats that a program can use to communicate with other software - it's like a bridge between different software applications allowing them to work together. APIs are used in all kinds of programming from web development (where they can be used to interact with server-side software or to integrate with third-party services) to operating systems (where they provide a way for applications to interact with the operating system).

**SDK (Software Development Kit)**: An SDK is a collection of software tools and libraries that developers use to create applications for specific platforms or frameworks. An SDK might include APIs, programming tools, graphical user interface (GUI) tools, documentation, and sample code. For example, if you're developing an application for iOS, you would use the iOS SDK which provides the tools and interfaces necessary to develop, test, and deploy applications that run on Apple's iOS devices.

**Microsoft Azure**, often simply known as Azure, is a cloud computing service created by Microsoft. It provides a variety of services including those for computing, analytics, storage, and networking. Users can pick and choose from these services to develop and scale new applications, or run existing applications, in the public cloud.

**Hugging Face Transformers** is an open-source library that provides state-of-the-art general-purpose architectures (BERT, GPT-2, RoBERTa, XLM, DistilBert, etc.) for Natural Language Understanding (NLU) and Natural Language Generation (NLG). It is designed to be research-friendly, efficient, and easy to use.

Here are some key features of Hugging Face Transformers:

1. **Pretrained Models**: The library includes thousands of pretrained models in about 100+ languages. These models have been trained on large-scale datasets and can be fine-tuned on a specific task (classification, entity recognition, question answering, etc.) with a small amount of data compared to training a model from scratch.
2. **Pipelines**: The library provides simple pipelines for common tasks like text classification, named entity recognition, and text generation. These pipelines abstract away the details of the underlying models and allow you to get results with just a few lines of code.

**Hugging Face Datasets** is a library for easily loading and sharing datasets. It provides access to a large repository of datasets for use in natural language processing (NLP) and machine learning, and it also provides tools for working with these datasets.

**Azure Open Datasets** are a collection of publicly available datasets provided by Microsoft Azure for use in your machine learning projects. These datasets are curated and optimized for machine learning, which can save you time on data discovery and preparation. They cover a wide range of domains including transportation, health and genomics, labor and economics, population and safety, and more.

Many machine learning solutions can be built as command line interfaces (CLIs).

**Flask** is a lightweight web framework for Python. It's designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

# DevOps, DataOps, MLOps

## Week 1: What is MLOps?

**MLOps**, short for Machine Learning Operations, is a practice for collaboration and communication between data scientists and operations professionals to help manage production Machine Learning (ML) lifecycle. It's similar to the DevOps or DataOps approaches, but specifically focused on managing the ML lifecycle.

Noah Gift, the instructor, thinks of MLOps like this:

* 25% DevOps
* 25% DataOps
* 25% Modelling
* 25% Framing

Why MLOps?

* **Increased Efficiency**: Data scientists can often be overwhelmed by the administrative aspects of model deployment, monitoring, and maintenance. MLOps helps automate these tasks, allowing data scientists to focus on their core competency - building and refining models.
* **Faster Results and Increased ROI**: MLOps helps streamline the model deployment process, which can lead to faster implementation and results. This, in turn, improves return on investment (ROI) for data science initiatives.
* **Leverage High-Level Tools**: MLOps makes it easier to utilize advanced tools and technologies that might be otherwise challenging to incorporate into traditional data science workflows. This can lead to improved model performance and capabilities.
* **Reproducibility and Consistency**: MLOps encourages best practices like version control, continuous integration, and continuous deployment, which enhance reproducibility and consistency. These are critical in data science, where models must be reliable and results replicable.
* **Collaboration and Communication**: MLOps fosters better collaboration and communication between the different roles involved in a data science project, including data scientists, data engineers, and business stakeholders. This is essential for delivering successful projects and for translating data science results into business insights.

MLOps strategy:

* **Choosing a Technology Partner**: When implementing an MLOps strategy, it can be helpful to choose a technology partner that aligns with your business's specific needs. Options include various cloud platforms like AWS, Azure, or Google Cloud.
* **Investment Prioritization**: Define primary and secondary areas for investment. This can include research and development, technological innovation, infrastructure, and talent acquisition.
* **Enterprise Support**: Consider whether enterprise support is necessary for your operations. Cloud providers like AWS and Azure offer comprehensive support options to ensure the seamless functioning of your machine learning operations.

MLOps key certifications:

1. **AWS Certified Machine Learning - Specialty**: Demonstrates ability to design, implement, deploy, and maintain machine learning solutions for given business problems.
2. **AWS Certified Data Analytics - Specialty**: Validates your ability to design, build, secure, and maintain analytics solutions on AWS.
3. **AWS Certified Solutions Architect - Associate**: Certifies your ability to design and deploy effective solutions on the AWS platform, which is a broader cloud certification but valuable for MLOps as well.

MLOps trends:

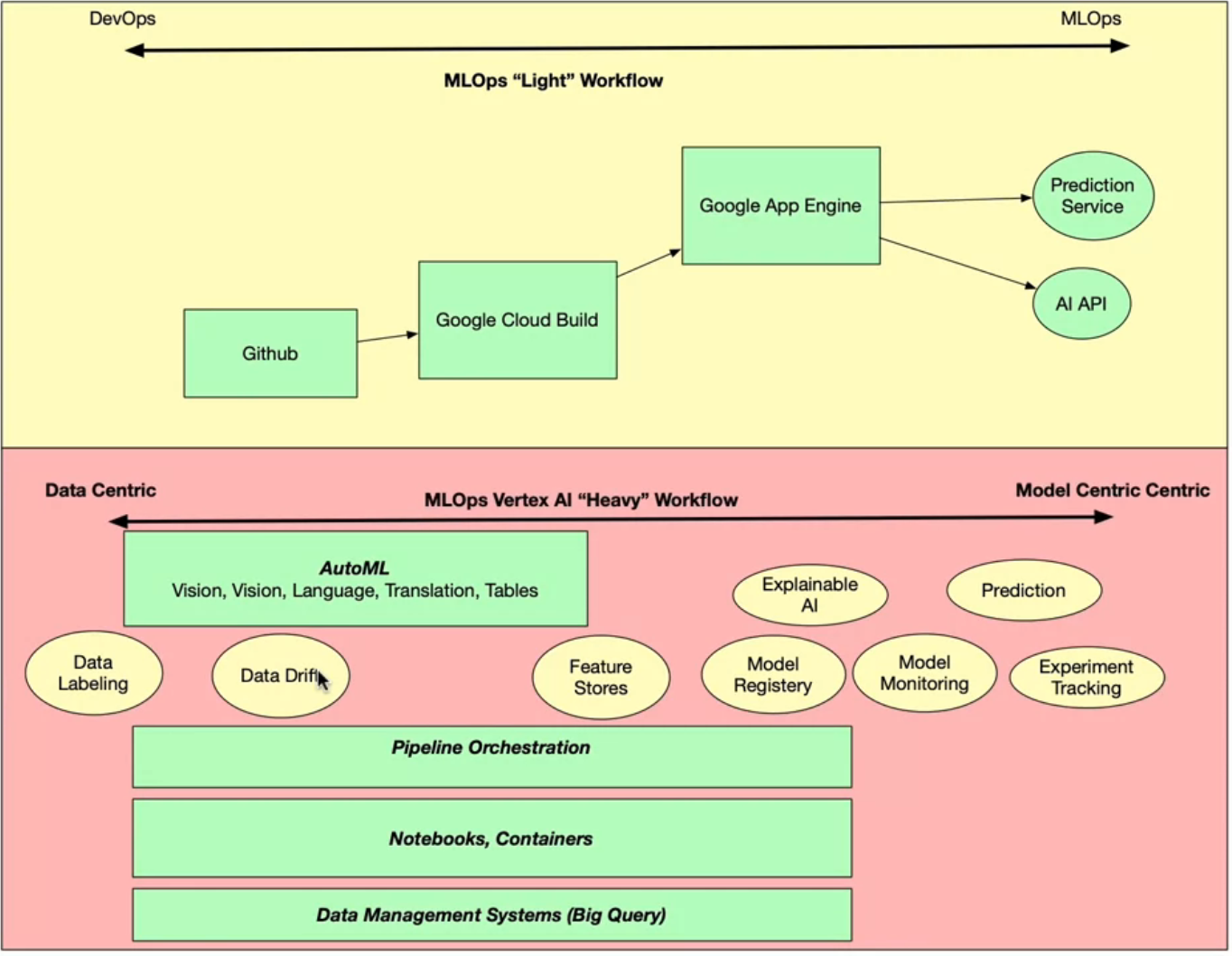
* **NFSOps**: As MLOps continues to evolve, NFSOps (Neural network as a Function Service) is emerging as a promising trend for serving neural networks.
* **Kubernetes and Kubeflow**: Container orchestration and specific frameworks for machine learning pipelines are becoming increasingly important in MLOps.
* **EdgeML**: Machine Learning on edge devices, known as EdgeML, is a growing trend for real-time, low-latency applications.
* **ESG (Environmental, Social, and Governance) factors**: ESG is becoming increasingly relevant in MLOps as companies strive for ethical and sustainable AI practices.
* **Pre-trained Models and AutoML**: The use of pre-trained models and automated machine learning platforms is on the rise, emphasizing that you don't always need to build models from scratch.
* **Model Portability**: The ability to develop a model in one environment and run it in another (often via containers) is becoming increasingly important.
* **Kaizen for ML**: The Kaizen philosophy, emphasizing continuous improvement, is being applied to machine learning, highlighting the need for constant iteration and improvement of models.
* **Production-First Mindset**: A trend towards prioritizing models that can be readily deployed and used in production, emphasizing practicality and applicability.

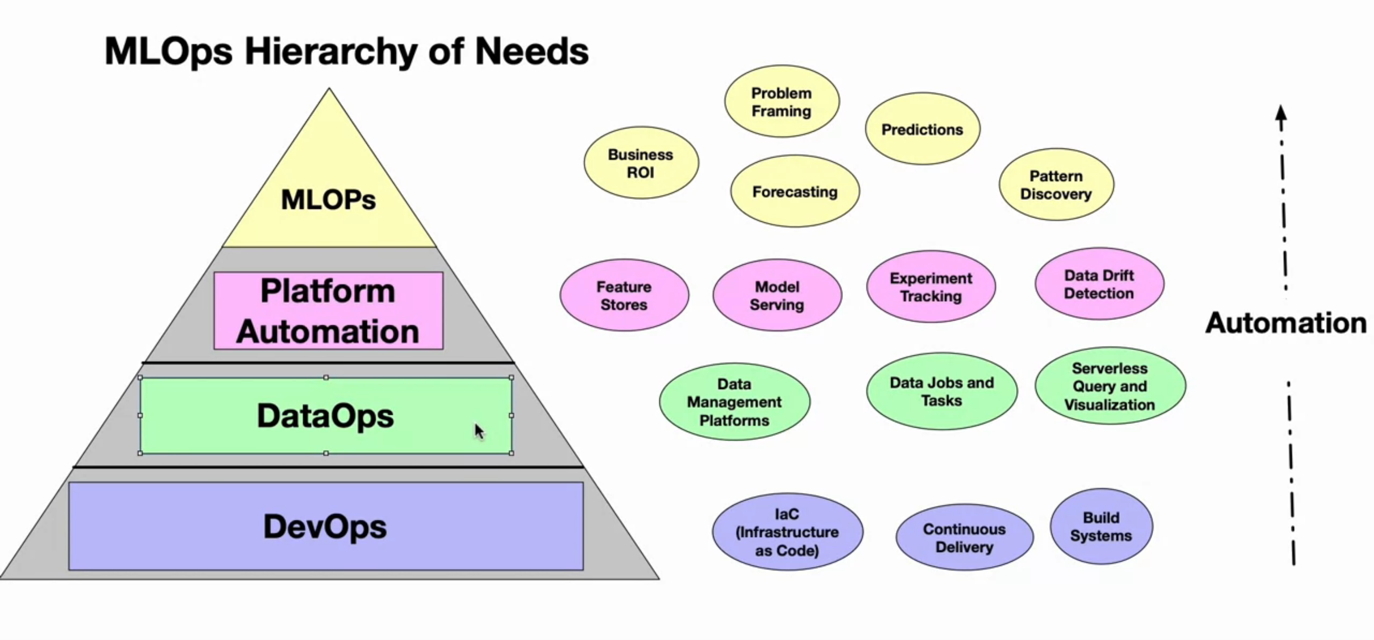
What is DevOps?

* Software engineering practices
* Automation
* Culture (Kaizen)

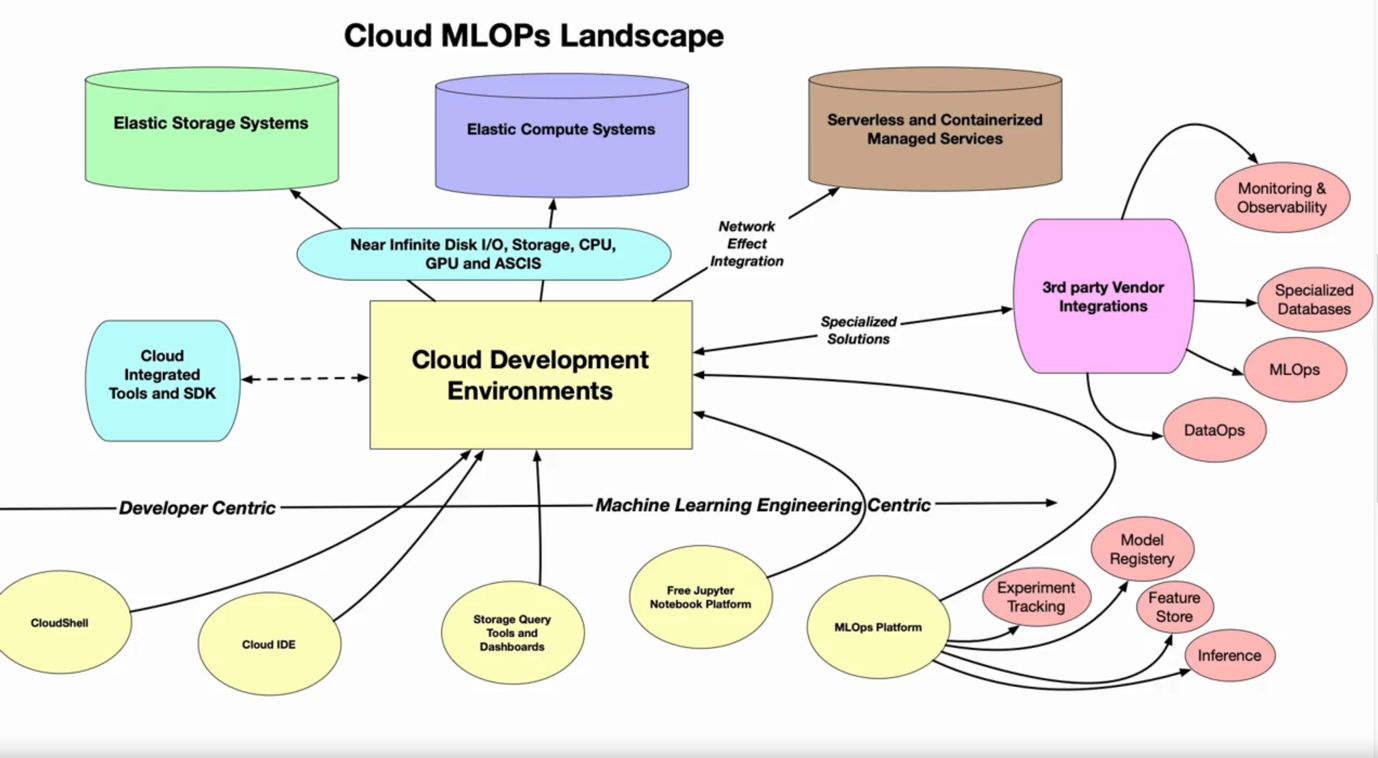
What is DataOps?

Light vs heavy MLOps:



Hierarchy of needs:  
  
  
**Data poisoning**

* Refers to the intentional manipulation of training data with the aim of corrupting the model's learning process. This is a type of adversarial attack on machine learning systems where the attacker introduces harmful data into the training set to influence the model's behavior in a way that benefits the attacker.

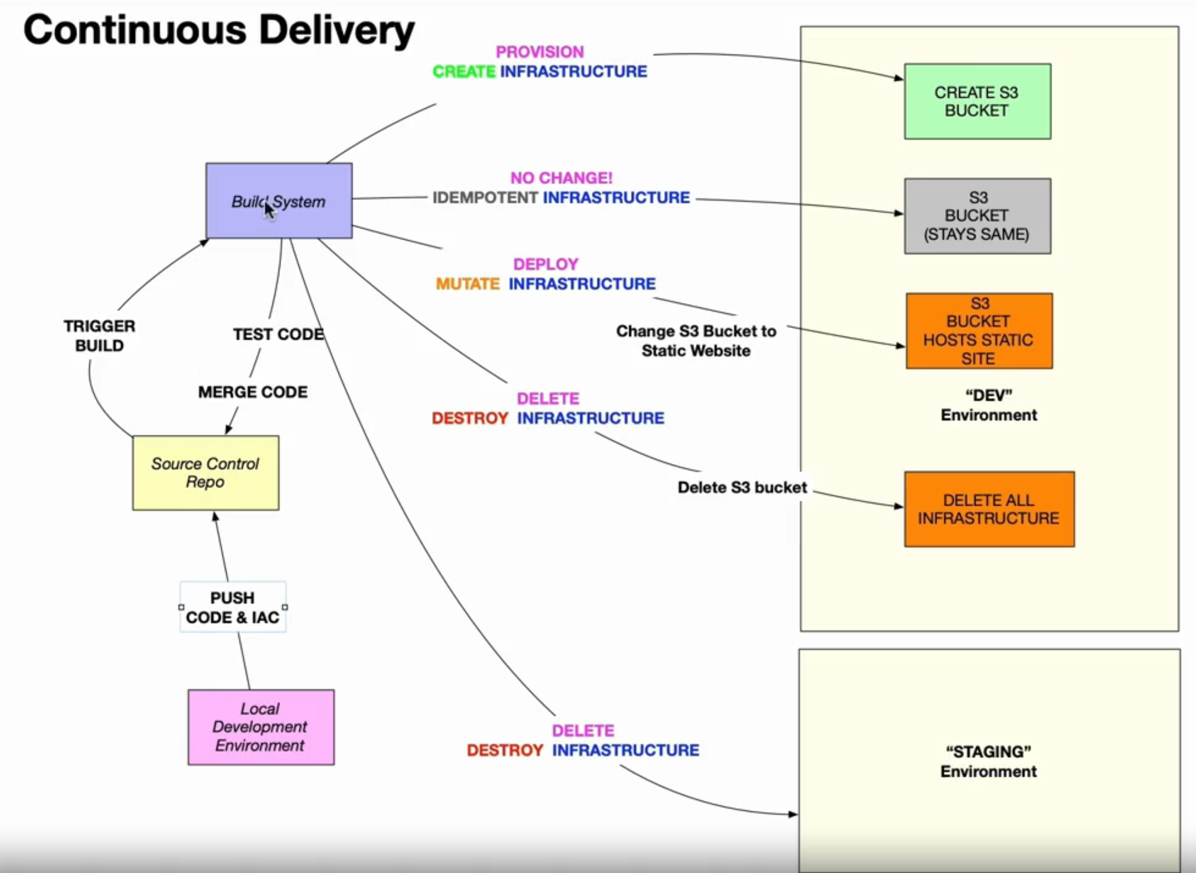
Key concepts in MLOps

MLOps maturity model

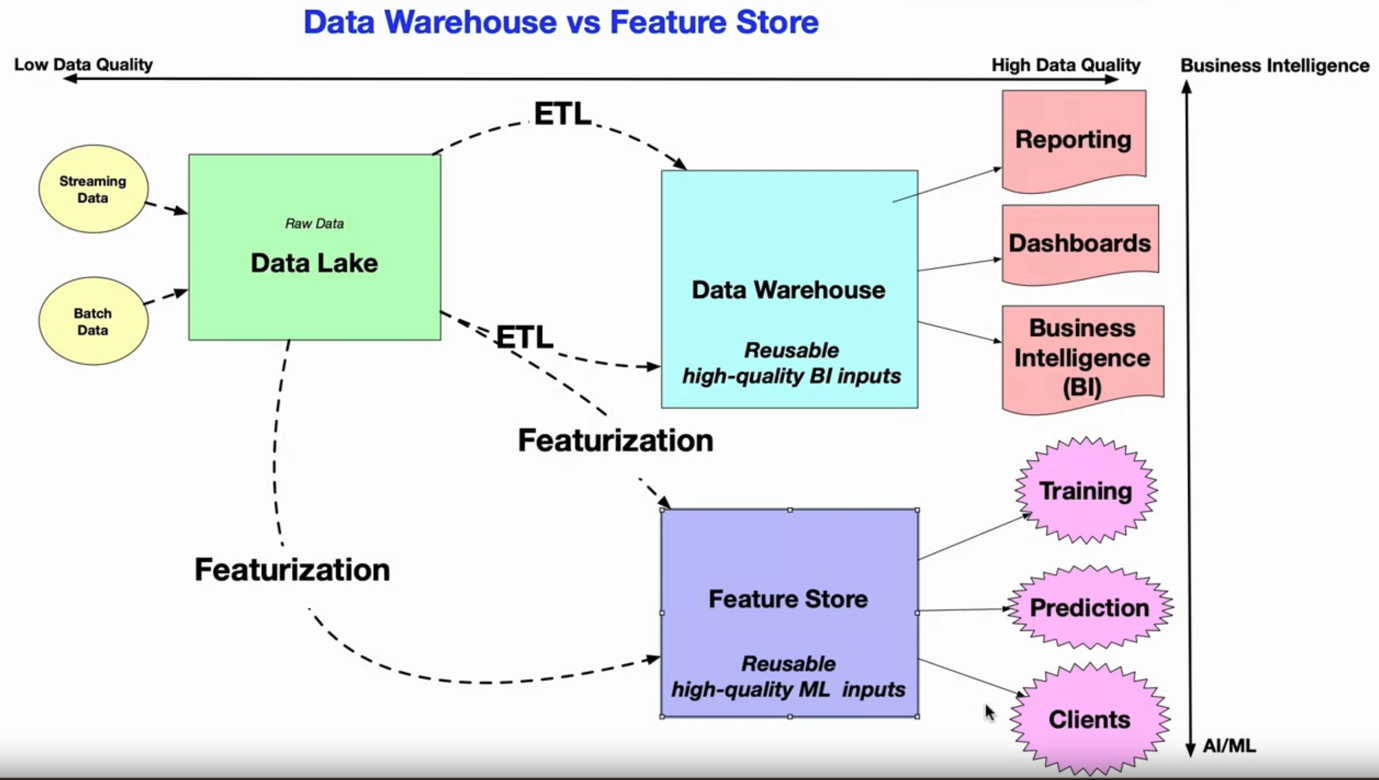
…

Continuous integration

* Testing is mandatory

Continuous delivery

Data warehouse vs feature store

* Data warehouse: …
* Feature store: …

Data drift: …

CI microservices:

* Test, lint, and format code for project

3 most important files in python project:

1. Makefile
2. Dockerfile
3. Requirements.txt

## Week 2: Essential math and data science

Data science on first day

* Structure:
  + 1. Ingest
    2. Exploratory data analysis
    3. Modelling
    4. Conclusion

**Traveling salesman problem**:

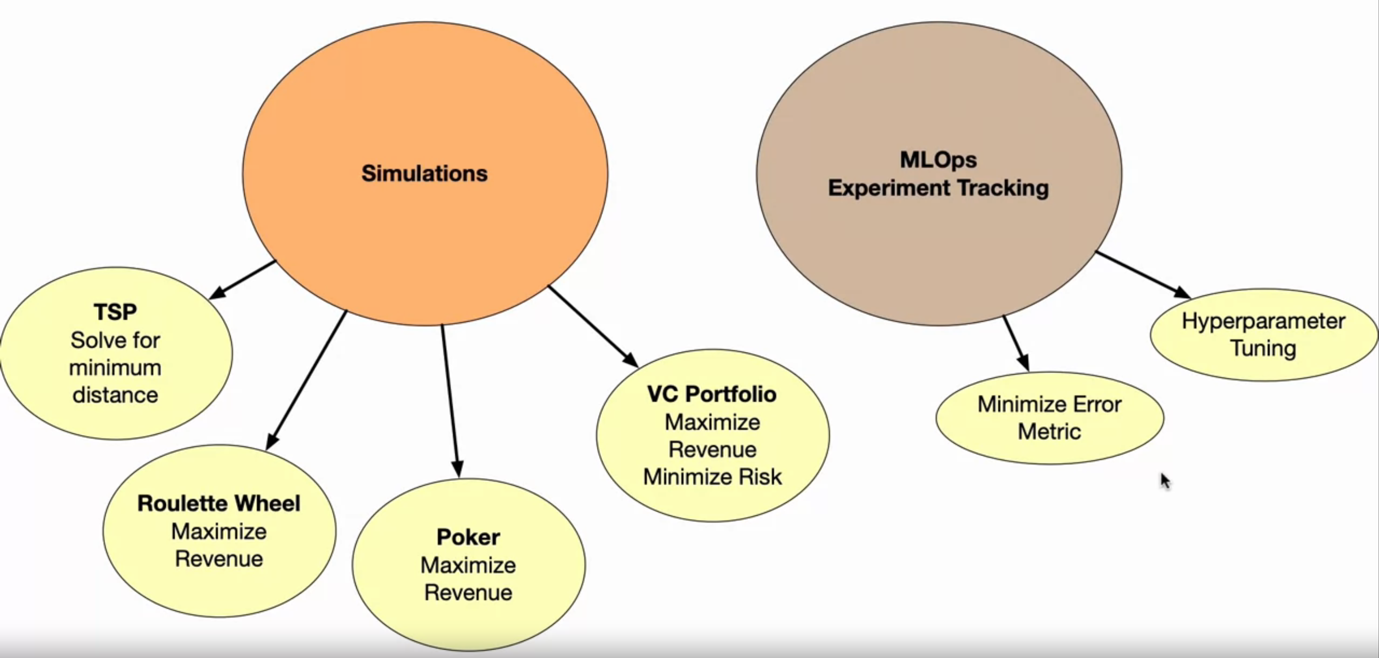
* The Traveling Salesman Problem is a classic algorithmic problem in the field of computer science and operations research. It focuses on optimization.
* The problem is defined as follows: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?
* It is an NP-hard problem in combinatorial optimization, important in operations research and theoretical computer science.
* The problem was first formulated in 1930 and is one of the most intensively studied problems in optimization.
* The most direct solution would be to try all permutations (ordered combinations) and see which one is cheapest (the brute force approach). However, this approach is not feasible for large number of cities due to the growth of permutations with the number of cities.
* Various heuristics and approximation algorithms are available, which provide solutions that, while not always optimal, can be close to optimal and are computationally cheaper to calculate.
* The TSP has several applications even in its purest form, such as planning, logistics, and the manufacture of microchips. Slightly modified, it appears as a sub-problem in many areas, such as DNA sequencing.
* Despite its simple formulation, the TSP does not have a known solution that is both efficient and exact. It is used as a benchmark for many optimization methods.

A heuristic commonly used to solve the Traveling Salesman Problem is the **Nearest Neighbor heuristic**. The Nearest Neighbor heuristic is a greedy algorithm, meaning it makes the locally optimal choice at each stage with the hope that these local choices will lead to a global optimum.

Here's how the Nearest Neighbor heuristic works:

1. Start at any city.
2. At each step, visit the nearest city that hasn't been visited yet.
3. Once all cities have been visited, return to the starting city.

This heuristic doesn't always find the most optimal solution, but it's much faster than checking all possible routes, especially for a large number of cities.

Simulations and Experiment Tracking

Cluster analysis

## Week 3: Operations Pipelines: DevOps, DataOps, MLOps

… Come back to week 3

## Week 4: End to End MLOps and AIOps

A diagram of a cloud microservices

Description automatically generated with medium confidence

… Need AWS access, come back to this

## Week 5: Rust for MLOps: The Practical Transition from Python to Rust

Rust learns from the problems of older languages and was built to counteract those. Thus it being new and using the cargo system makes it a breath of fresh air.

Rust is also very fast and efficient (like C++).

Rust is concurrent, unlike python which is very hard to do threading.

Rust is very safe.

## MLOps Platforms: Amazon SageMaker and Azure ML

…

## MLOps Tools: MLflow and Hugging Face