Designing ML Pipelines

By Mahdi Akhi

Mahdi Akhi

- MSc. Student of Software Engineering at Sharif University of Technology
- Software Engineer
- Contact Info
  - linkedin.com/in/mahdiakhi

Roadmap

- The problem
- What is an ML Pipeline?
- Pipeline components
- Design an ML Pipeline
- Pipeline benefits
- Tips on design and implementation

The problem

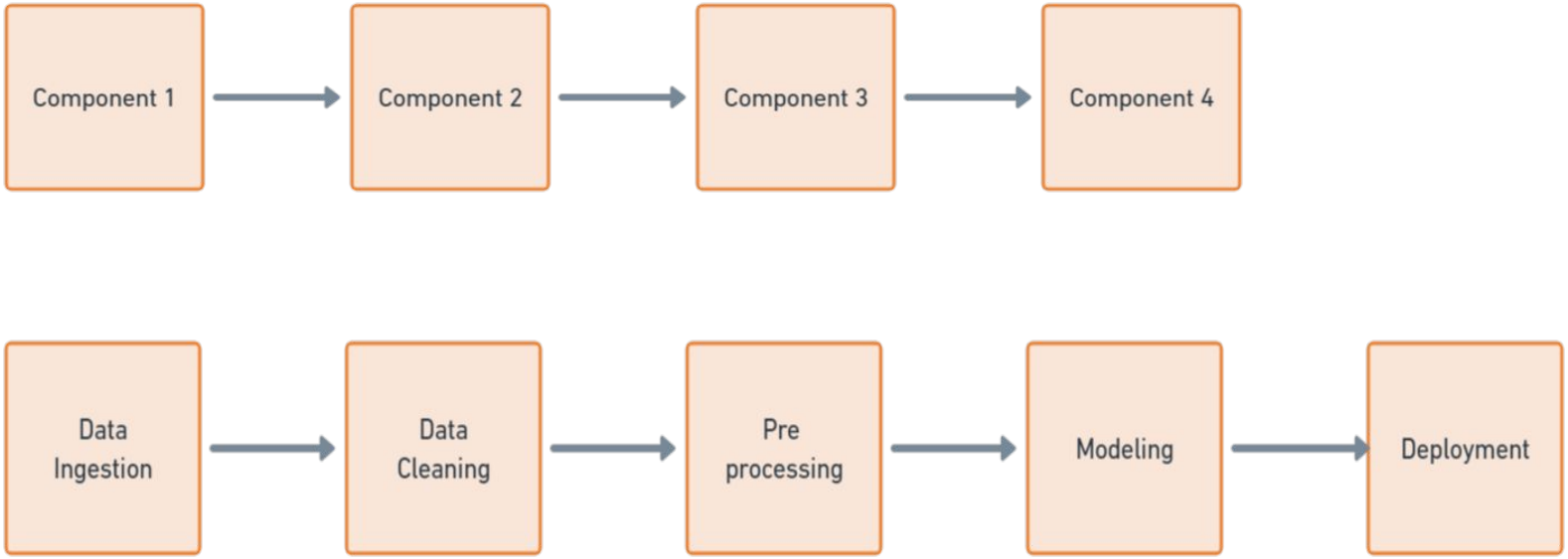
- A large number of businesses are using AI in their products.
- They need a complex system consist of multiple single-component.
- The system should have some abilities:
  - Automated
  - Storages
  - Continuous training
  - Model validation
  - Availability, Scalability, Modularity
  - Monitoring System
- The solution: ML pipeline

The solution

A ML pipeline is a process of automating the workflow of a complete machine learning task.

- Enable data sequence transformation and correlation in a model for analysis and output.
- A good pipeline
  - Fully automated
  - High-Performance
  - The design is independent of the implementation
  - Modularity in design and implementation
  - Scalable and reusable components

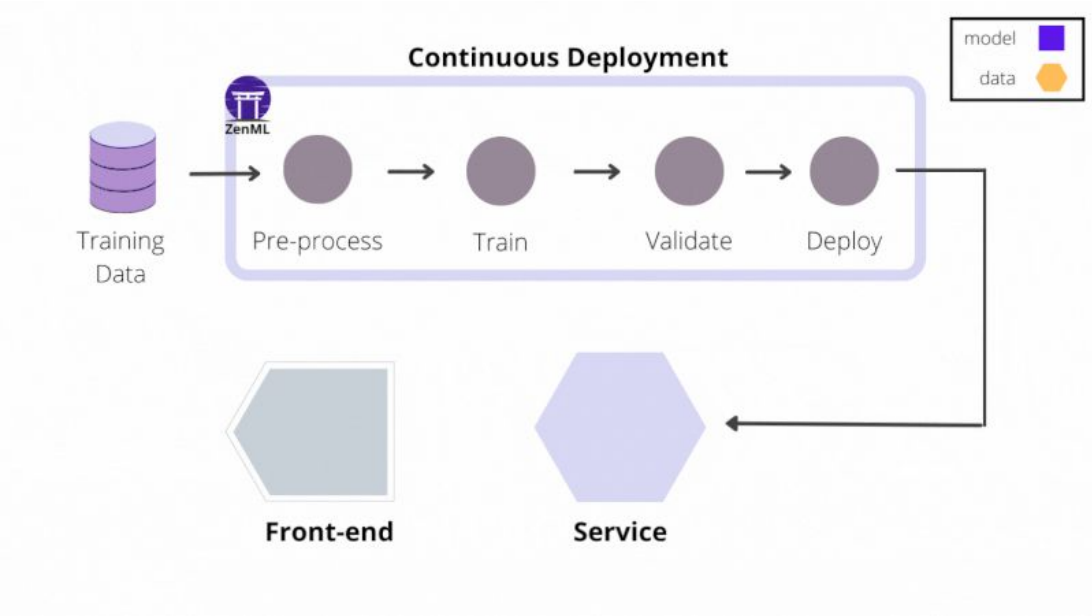
Components


Component 1
Component 2
Component 3
Component 4
Data Ingestion
Data Cleaning
Pre processing
Modeling
Deployment

Components

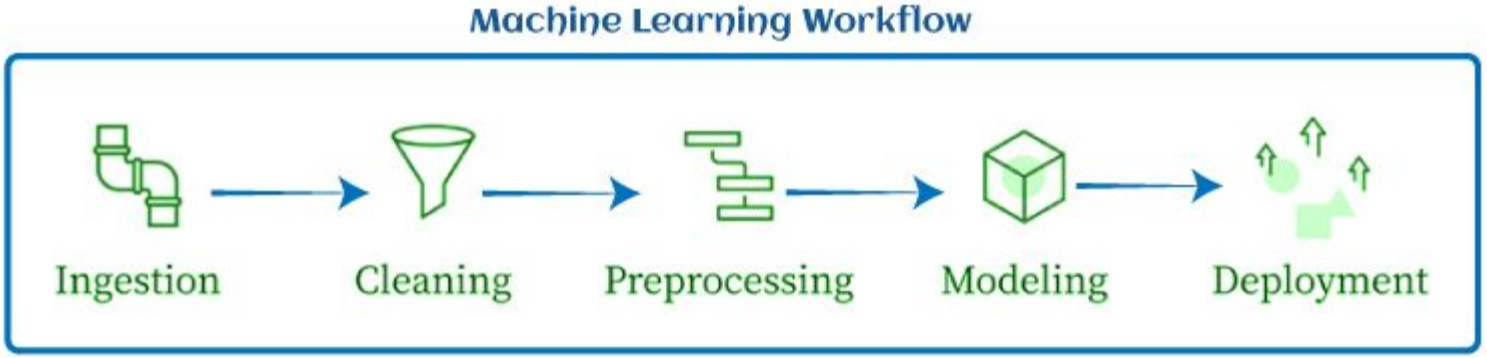
- The components can be different based on various factors
  - Data sources and data types
  - Business requirements
  - ML Models
- Each component is designed as an independent module, and all these modules are tied together to get the final result.
- Component properties
  - Independent in functionality
  - Reusable and Scalable
  - Testable

Example



An example of pipeline from zenml.io

Example



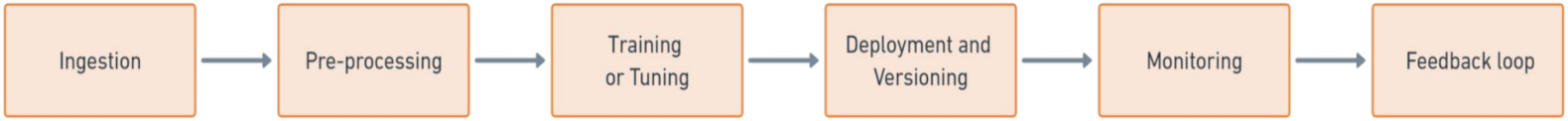
An example of pipeline from javapoint.com

Designing A Pipeline

Scenario

- We have a home price prediction system
  - There are several data sources available, including websites, real estate agencies, datasets, and governmental data.
  - Permanently incorporating new data sources.
  - A new model is deployed on a weekly basis.
  - The business is expanding by adding new locations to the system(USA, Germany, Canada, etc).
  - Customer feedback is collected for each prediction.

Pipeline



Our pipeline

Pipeline - ingestion

- We have several data sources
  - Websites(crawling) like booking.com, Zillow.com, realtor.com, etc.
  - Datasets
  - Real estate agencies' data
  - Governmental data
  - etc.
- Each one needs preprocessing, transforming, and special data structure.
- Challenges with data
  - Structural conflict
  - Semantic conflict
  - Duplication

Ingestion - data challenges

Both of these JSONs contain information about the same home, but they are structured differently.

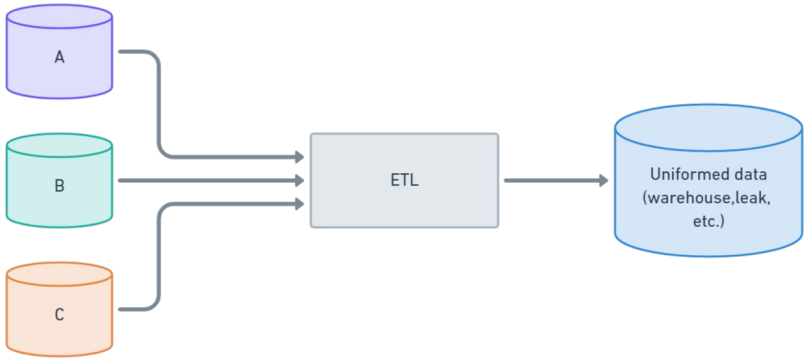

```
{
  "home": {
    "room": 2,
    "built_date": "1990-05",
    "cooling_system": "Electric",
    "heating_system": "heat pump",
    "address": "15 Star Thistle, Irvine, CA 92604",
    "price": "130000",
    "garage_spaces": 2,
    "area": "1200 sqft"
  }
}
```



```
{
  "home": {
    "room": 2,
    "temprature_system": {
      "cooler": "Electric",
      "heater": "heat pump"
    },
    "address": {
      "state": "CA",
      "city": "Irvine",
      "street": "15 Star Thistle",
      "zip_code": "92604"
    },
    "built_date": "1990 May",
    "price": "130000",
    "garage_spaces": 2,
    "area": "130 m2"
  }
}
```

Ingestion - solution

- ETL: Translate raw data into actionable data
- Extract
  - Data is retrieved from its source system and converted into a single format suitable for transformation processing
- Transform
  - A set of rules is applied to the extracted data to convert it into a unique structure.
- Load
  - The extracted and transformed data is then loaded into the final target source, such as a data warehouse, data hub, or data lake structure.



Pipeline - ingestion

- Some data ingestion tools
  - Apache NiFi
  - Apache Kafka
  - Talend
  - Dropbase
  - Apache Flume

Pipeline - Preprocessing

- Convert data to a model-usable form.
- One of the most critical steps in both the machine learning lifecycle and pipeline.
- Absolutely depends on data and business.
- The process consists of various sub-steps, including data cleaning, feature scaling, etc. Each sub-step should be modular and reusable.
- The output is the final dataset and its usable for model training or tuning.
- Can have a monitoring system.

Pipeline - Model training or Tuning

- The core of ML pipelines is the model training.
- Train a new model or fine-tune an existing model?
- Maybe there are some difficulties with large models and data.
  - Using parallelism
  - Save trained model every x step(e.g. epoch) - Model checkpoint
- Monitoring system to save training metrics for each step.
  - For example accuracy in each epoch
- You have to validate the trained model to ensure the accuracy and performance.

Pipeline - Deployment and Versioning

- Once the model trained and validated, the model should deploy to production automatically.
  - **Serve on a server**
  - Serve on an application
- This is typically achieved using containerization technologies like Docker or Kubernetes.
- The common way to deploy the model is using a model server.
  - Allow to host multiple versions simultaneously
  - Helps to run A/B tests on models

Pipeline - Deployment and Versioning

- Maybe require some pre or post-processing on client requests.
  - Preprocessing like prompt engineering, or apply some filters.
  - Post-processing like apply filters on model output or refine outputs to the UI.
- Serving tools
  - TorchServe
  - TF Serving
  - KubeFlow
  - And many other online or open-source tools.
- Each deployment should be versioned. In the event of a problem with the current model, the system can swiftly revert to the previous version to minimize disruption.

Pipeline - Monitoring and Feedback loop

- The new model needs to be monitored
  - Deviations from expected results.
  - Drops in performance.
  - Data/concept drift.
  - Etc.
- A feedback loop is in place to incorporate insights from the monitoring stage back into the model for future enhancements.
  - Either automated or manual, depending on the requirements.
- You may have to use the Human-in-the-Loop strategy to design the feedback loop or even whole the pipeline.

Pipeline - Benefits

- Automated and unattended runs
- Fast execution
- Build and design one time, use several times
- Reusability
- Scalability
- Easy to debugging
- Decrease the costs and resources
- Track the evolution and behavior of the model(s) over the time and versions.

Pipeline - Tips to design and implementation

- Think about the design, independent of the implementation and tools.
  - Think about the problem and the solution. Think about why, not how.
- Modular design
  - Each component should be independent in functionality and be single responsible.
- Reusable design
  - Each component should be reusable.
- Automated and codified tests
  - Each component should be covered with automated tests. First, think about the “how to testing a component” and then design it!
- Put the hyper parameters and configs in a high-level configuration.
  - E.g. put the hyper parameters of the model in kubernetes variable, not in code!