# End-to-End Machine Learning Pipeline

## Overview

An end-to-end machine learning (ML) pipeline is a sequence of stages that takes raw data and produces a deployed model that provides predictions in production. A well-designed pipeline automates data preparation, model training and evaluation, deployment, and monitoring—ensuring reproducibility, scalability, and maintainability.

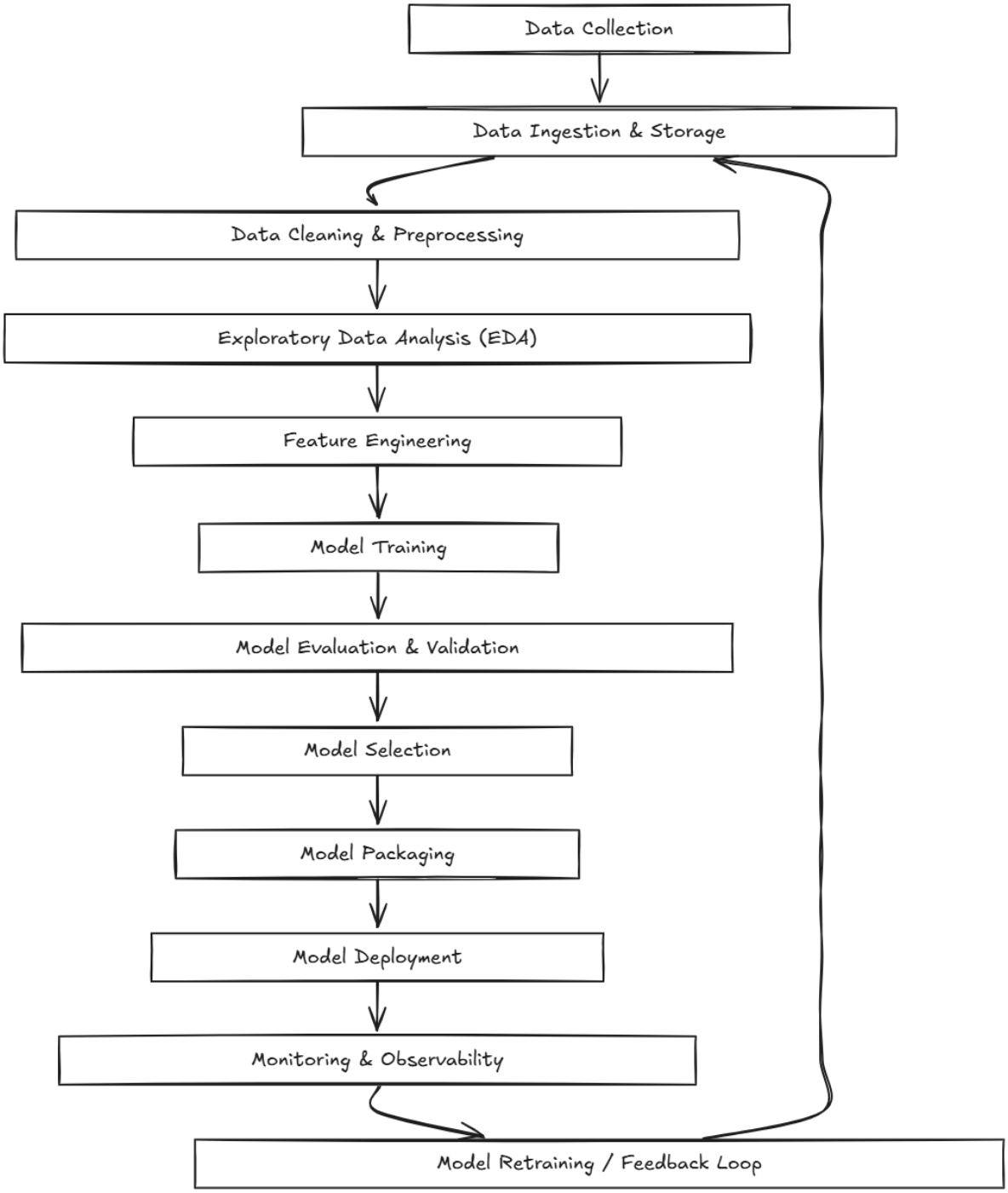
---

## Pipeline Diagram

Below is a visual representation of the pipeline.

In [12]:
```
from IPython.display import Image
Image(filename="pipeline.png")
```

Out[12]:




The machine learning (ML) pipeline is a structured workflow that organizes the process of building, deploying, and maintaining ML models. Below are the main stages:

1. **Data Collection**
   - Gathering raw data from various sources such as databases, sensors, APIs, or files.
   - This forms the foundation of the ML pipeline.
2. **Data Ingestion & Storage**
   - Importing data into a central repository (data warehouse, data lake, or local storage).

- Ensures data is available and accessible for further processing.

3. **Data Cleaning & Preprocessing**
   - Handling missing values, duplicates, outliers, and inconsistent formats.
   - Normalizing, standardizing, and encoding features so the data is suitable for modeling.
4. **Exploratory Data Analysis (EDA)**
   - Understanding data distributions, relationships, and patterns.
   - Using statistics and visualizations to guide feature engineering and model selection.
5. **Feature Engineering**
   - Creating new features or transforming existing ones to improve model performance.
   - Techniques include scaling, encoding categorical variables, or generating interaction features.
6. **Model Training**
   - Applying ML algorithms to learn patterns from the training dataset.
   - Hyperparameters are tuned to optimize performance.
7. **Model Evaluation & Validation**
   - Assessing the model using validation/test datasets.
   - Metrics such as accuracy, precision, recall, F1-score, or RMSE are calculated.
8. **Model Selection**
   - Comparing multiple candidate models and choosing the one with the best performance.
   - Balances accuracy, interpretability, and computational efficiency.
9. **Model Packaging**
   - Preparing the trained model for deployment by saving in a standard format (e.g., pickle, ONNX, PMML).
   - Includes necessary preprocessing steps.
10. **Model Deployment**
    - Integrating the model into production (APIs, web apps, mobile apps, or embedded systems).
    - Makes the model available to end-users or business systems.
11. **Monitoring & Observability**
    - Tracking the model's performance in production.
    - Detecting data drift, performance degradation, and operational issues.
12. **Model Retraining / Feedback Loop**
    - Collecting new data and retraining the model to adapt to changes.

- Ensures the model remains accurate and relevant over time.