Data Management in AI Projects

The Data Lifecycle

Learning Objective

By the end of this lesson, learners will be able to:

- Map data through its lifecycle from collection to deletion in an AI project.
- Implement hands-on coding to simulate each stage of the lifecycle.
- Identify key consulting considerations when working with clients on data governance, security, and compliance.

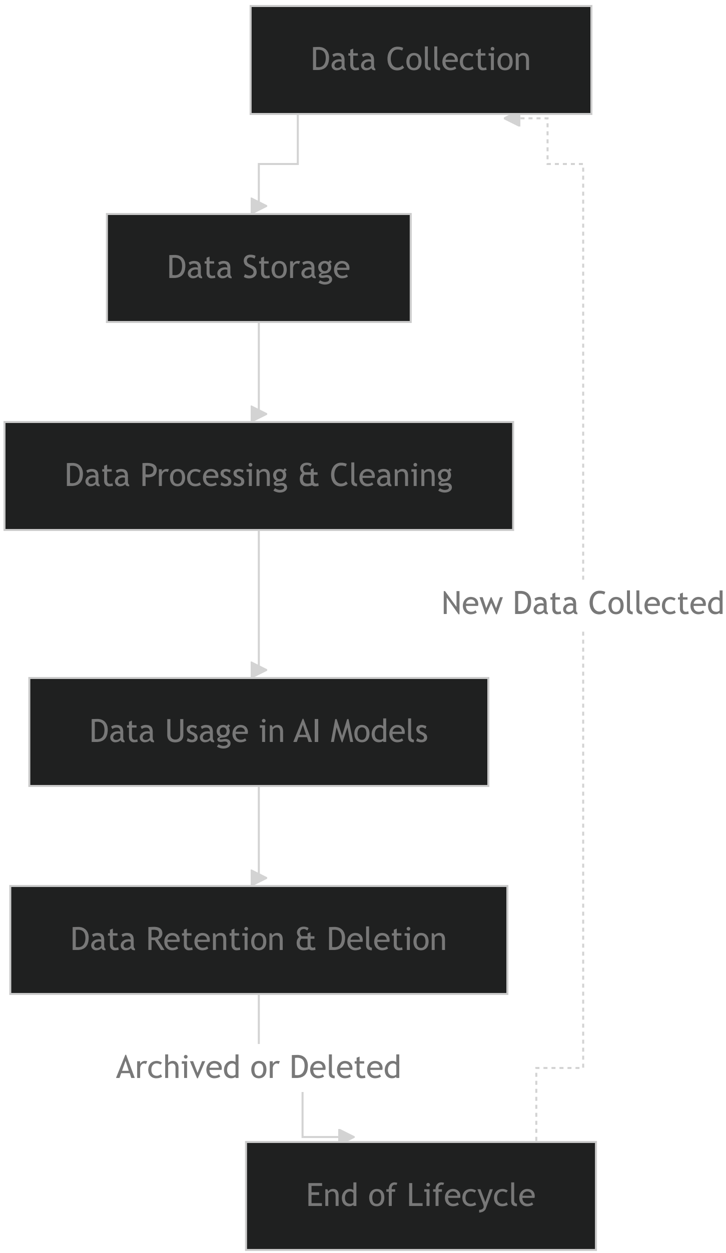
1. Introduction: Understanding the Data Lifecycle

Data in AI projects moves through distinct stages:

1. **Collection** – Gathering raw data from sources.
2. **Storage** – Saving data in appropriate formats.
3. **Processing & Cleaning** – Preparing data for AI models.
4. **Usage** – Leveraging data for insights and predictions.

5. **Retention & Deletion** – Managing long-term data policies.

Visualizing the Data Lifecycle



Key considerations when working with clients:

- What data privacy laws (e.g., GDPR, CCPA) apply to this project?
- Does the client need real-time data processing or batch updates?
- What security and access controls should be in place for sensitive data?

2. Hands-On: Simulating the Data Lifecycle with Code

Step 1: Data Collection (Simulating Client Data)

Scenario: Your client is an e-commerce company. They want to analyze customer transaction data for personalized recommendations.

Code: Generate Synthetic Data

Copy

```
import pandas as pd
import numpy as np

np.random.seed(42)
data = {
    "customer_id": np.random.randint(1000, 9999, 100),
    "purchase_amount": np.random.uniform(10, 500, 100),
    "purchase_category": np.random.choice(["Electronics", "Clothing", "Home"], 10
    "purchase_date": pd.date_range(start="2023-01-01", periods=100, freq="D"),
}
df = pd.DataFrame(data)
```

Key considerations when working with clients:

- What sources does the client collect data from (APIs, databases, external providers)?
- How frequently should new data be collected?

Step 2: Data Storage & Format Considerations

Clients often store data in different formats. Choosing the right format impacts **speed, cost, and scalability.**

Code: Save Data in Different Formats

Copy

```
df.to_csv("transactions.csv", index=False)
df.to_json("transactions.json", orient="records")
df.to_parquet("transactions.parquet", index=False)
```

Key considerations when working with clients:

- Does the client need high-speed querying (Parquet) or compatibility (CSV)?
- Should storage be on-premises, cloud-based, or hybrid?

Step 3: Data Processing & Cleaning

Challenge: Real-world data is often **messy**—it contains duplicates, missing values, or incorrect formats.

Code: Data Cleaning & Validation

Copy

```
df.drop_duplicates(inplace=True)
df["purchase_amount"] = df["purchase_amount"].round(2)
```

Key considerations when working with clients:

- What data quality checks are required before analysis?
- Who is responsible for data validation—AI engineers, data analysts, or business users?

Step 4: Data Usage in AI Models

AI models require structured, preprocessed data. We simulate a **basic classification model** predicting whether a customer will make another purchase.

Code: Prepare Data for AI Model

Copy

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.ensemble import RandomForestClassifier

X = df.drop(columns=["customer_id", "purchase_date"])
y = np.random.choice([0, 1], 100)  # Simulated repurchase prediction

# Encode categorical variables
encoder = OneHotEncoder(sparse_output=False)
```

```
X_encoded = encoder.fit_transform(df[["purchase_category"]])

# Scale numerical data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df[["purchase_amount"]])

# Combine processed data
X_final = np.hstack((X_scaled, X_encoded))

# Train a basic model
X_train, X_test, y_train, y_test = train_test_split(X_final, y, test_size=0.2, ra
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

Key considerations when working with clients:

- What **business outcomes** does the client expect from AI insights?
- How frequently should AI models be retrained with fresh data?

Step 5: Data Retention & Deletion Policies

Not all data needs to be kept indefinitely. Companies must balance **storage costs, compliance, and business needs**.

Code: Filter Out Old Data

Copy

```
df = df[df["purchase_date"] > "2023-03-01"]  # Remove old data
df.to_csv("updated_transactions.csv", index=False)
```

Key considerations when working with clients:

- How long should different types of data be retained?
- What compliance regulations (e.g., GDPR Right to Erasure) must be followed?

3. Recap & Key Takeaways

- ✅ Data moves through a lifecycle from **collection** → **storage** → **processing** → **usage** → **deletion**.
- ✅ Different **storage formats** impact performance and cost.
- ✅ Cleaning and preprocessing ensure **data quality** before AI modeling.
- ✅ AI models need **structured, processed data** to generate insights.
- ✅ Compliance, governance, and business needs shape **data retention policies**.

< Previous

Attributions

Next >