**Mô tả đề tài**

Xây dựng một **pipeline MLOps tự động hoàn toàn** để quản lý vòng đời dữ liệu và mô hình học máy phục vụ cho các ứng dụng gợi ý nhạc hoặc phân tích âm nhạc. Pipeline này sẽ bao gồm các bước:

* **Thu thập dữ liệu (Data Ingestion):** Tự động lấy dữ liệu từ Spotify API (playlist, track, album, lịch sử nghe) hoặc từ các nguồn dữ liệu phụ trợ (CSV, database, data lake).
* **Tiền xử lý và làm sạch dữ liệu (Data Cleaning & Transformation):** Chuẩn hóa, xử lý thiếu sót, loại bỏ dữ liệu nhiễu, đảm bảo dữ liệu đồng nhất và phù hợp cho huấn luyện mô hình.
* **Lưu trữ dữ liệu (Data Storage):** Quản lý dữ liệu trên hệ thống lưu trữ tập trung (data lake, data warehouse) với các tầng dữ liệu (bronze, silver, gold) để dễ dàng truy xuất và tái sử dụng.
* **Huấn luyện mô hình (Model Training):** Tự động huấn luyện, đánh giá và lựa chọn mô hình ML dựa trên dữ liệu mới nhất, sử dụng các framework như MLflow, TensorFlow Extended (TFX).
* **Triển khai mô hình (Model Deployment):** Tự động đóng gói và triển khai mô hình lên môi trường sản xuất (production), hỗ trợ rollback khi phát hiện lỗi.
* **Giám sát và bảo trì (Monitoring & Maintenance):** Theo dõi hiệu suất mô hình, phát hiện drift dữ liệu, cảnh báo và tự động cập nhật mô hình khi cần thiết.

**Yêu cầu kỹ thuật**

* **Tự động hóa toàn bộ pipeline:** Sử dụng công cụ orchestration như Airflow, Dagster hoặc Kubeflow để điều phối các bước trong pipeline.
* **Quản lý phiên bản dữ liệu và mô hình:** Áp dụng DVC hoặc MLflow để theo dõi và quản lý các phiên bản dữ liệu và mô hình.
* **Xử lý dữ liệu quy mô lớn:** Sử dụng các công nghệ xử lý dữ liệu như Apache Spark, Flink hoặc BigQuery để xử lý và biến đổi dữ liệu hiệu quả.
* **Lưu trữ dữ liệu phân tầng:** Thiết kế hệ thống lưu trữ dữ liệu theo các tầng Bronze (dữ liệu thô), Silver (dữ liệu đã làm sạch), Gold (dữ liệu sẵn sàng cho ML).
* **Tích hợp CI/CD cho mô hình:** Tự động build, test và deploy mô hình qua các công cụ như Jenkins, GitHub Actions.
* **Giám sát mô hình:** Sử dụng Prometheus, Grafana hoặc các công cụ tương tự để theo dõi hiệu suất, độ chính xác và drift của mô hình.
* **Bảo mật và tuân thủ:** Đảm bảo dữ liệu được ẩn danh, tuân thủ các quy định bảo mật và quyền riêng tư (GDPR, CCPA).
* **Khả năng mở rộng:** Hệ thống có thể mở rộng linh hoạt theo khối lượng dữ liệu và số lượng người dùng.

**Kết quả mong đợi**

* Pipeline MLOps hoạt động ổn định, tự động từ thu thập dữ liệu đến triển khai mô hình.
* Dữ liệu được quản lý hiệu quả, dễ dàng truy xuất và tái sử dụng.
* Mô hình ML được cập nhật liên tục, đáp ứng tốt với thay đổi dữ liệu thực tế.
* Giảm thiểu lỗi vận hành, tăng tốc độ triển khai và bảo trì mô hình.
* Có dashboard giám sát đầy đủ về dữ liệu và mô hình.

**1. Thu thập dữ liệu (Data Ingestion)**

* **Mục tiêu:** Tự động lấy dữ liệu từ Spotify API (playlist, track, album, lịch sử nghe) và các nguồn dữ liệu phụ trợ (CSV, database).
* **Công cụ:** Apache Airflow hoặc Dagster để orchestration, Apache Kafka nếu cần streaming real-time.
* **Quy trình:**
  + Viết task gọi API Spotify định kỳ hoặc theo trigger (ví dụ mỗi giờ, mỗi ngày).
  + Lưu dữ liệu thô (raw data) vào hệ thống lưu trữ tạm (data lake, ví dụ AWS S3).
  + Xử lý lỗi, retry khi gọi API thất bại.
* **Ví dụ:** Spotify dùng Airflow + Kafka để ingest dữ liệu real-time từ user interaction.

**2. Tiền xử lý và làm sạch dữ liệu (Data Cleaning & Transformation)**

* **Mục tiêu:** Chuẩn hóa dữ liệu, xử lý missing value, loại bỏ dữ liệu nhiễu, chuyển đổi dữ liệu về dạng phù hợp cho huấn luyện.
* **Công cụ:** Pandas, Spark, dbt (data build tool) để transform dữ liệu trong data warehouse.
* **Quy trình:**
  + Tạo các pipeline ETL/ELT để biến đổi dữ liệu raw thành dữ liệu sạch (silver) và dữ liệu sẵn sàng cho ML (gold).
  + Áp dụng các quy tắc làm sạch như chuẩn hóa tên nghệ sĩ, xử lý ngày tháng, loại bỏ track không hợp lệ.
* **Ví dụ:** Dùng dbt trên PostgreSQL để transform dữ liệu Spotify thành bảng phân tích.

**3. Lưu trữ dữ liệu (Data Storage)**

* **Mục tiêu:** Quản lý dữ liệu theo tầng (bronze, silver, gold) để dễ dàng truy xuất và tái sử dụng.
* **Công cụ:** AWS S3 làm data lake, PostgreSQL hoặc Amazon Redshift/BigQuery làm data warehouse.
* **Quy trình:**
  + Lưu dữ liệu raw vào Bronze layer (S3 bucket).
  + Sau tiền xử lý, lưu dữ liệu sạch vào Silver layer (PostgreSQL).
  + Dữ liệu cuối cùng dùng cho ML lưu ở Gold layer.
* **Ví dụ:** Cấu trúc thư mục và phân tầng dữ liệu trong S3 theo chuẩn ELT.

**4. Huấn luyện mô hình (Model Training)**

* **Mục tiêu:** Tự động huấn luyện và đánh giá mô hình ML dựa trên dữ liệu mới nhất.
* **Công cụ:** MLflow để tracking experiment, TensorFlow/PyTorch cho training, Kubeflow hoặc Airflow để orchestration.
* **Quy trình:**
  + Tạo pipeline training tự động trigger khi có dữ liệu mới hoặc theo lịch.
  + Lưu model, log metrics, so sánh performance các phiên bản.
  + Chọn model tốt nhất để deploy.
* **Ví dụ:** Spotify dùng MLflow để quản lý phiên bản mô hình và tracking performance.

**5. Triển khai mô hình (Model Deployment)**

* **Mục tiêu:** Đóng gói và triển khai mô hình lên môi trường production, hỗ trợ scaling và rollback.
* **Công cụ:** Docker container, Kubernetes để quản lý deployment, REST API hoặc gRPC để expose model.
* **Quy trình:**
  + Tự động build image Docker chứa model.
  + Triển khai lên cluster Kubernetes với autoscaling.
  + Triển khai canary hoặc blue-green để giảm rủi ro.
* **Ví dụ:** Spotify deploy mô hình recommendation dưới dạng microservices trên Kubernetes.

**6. Giám sát và bảo trì (Monitoring & Maintenance)**

* **Mục tiêu:** Theo dõi hiệu suất mô hình, phát hiện drift dữ liệu, cảnh báo lỗi và tự động cập nhật mô hình.
* **Công cụ:** Prometheus, Grafana để monitoring; custom alerting và anomaly detection.
* **Quy trình:**
  + Thu thập metrics về độ chính xác, latency, throughput của mô hình.
  + Giám sát dữ liệu đầu vào để phát hiện thay đổi phân phối (data drift).
  + Tự động trigger retraining khi phát hiện drift hoặc hiệu suất giảm.
* **Ví dụ:** Spotify dùng hệ thống giám sát real-time để phát hiện và retrain mô hình kịp thời

| **Bước** | **Công cụ gợi ý** | **Mục tiêu chính** |
| --- | --- | --- |
| Data Ingestion | Airflow, Dagster, Kafka | Thu thập dữ liệu tự động, real-time hoặc batch |
| Data Cleaning | Pandas, Spark, dbt | Làm sạch, chuẩn hóa dữ liệu |
| Data Storage | S3, PostgreSQL, BigQuery | Lưu trữ phân tầng dữ liệu |
| Model Training | MLflow, TensorFlow, Kubeflow | Huấn luyện, tracking và chọn model tốt nhất |
| Model Deployment | Docker, Kubernetes | Triển khai, scaling và rollback model |
| Monitoring | Prometheus, Grafana | Giám sát hiệu suất, phát hiện drift, alert |

**Ứng dụng trong ML**

* **Phân nhóm (Clustering):** Dùng artist\_genres và album\_release\_date để nhóm bài hát theo thể loại và thời gian phát hành.
* **Phân loại (Classification):** Dùng thể loại nghệ sĩ làm nhãn để huấn luyện mô hình phân loại bài hát.
* **Phân tích xu hướng:** Dựa trên ngày phát hành album để phân tích sự thay đổi thể loại theo thời gian.

| **Giai đoạn** | **Mô tả** | **Công cụ gợi ý** |
| --- | --- | --- |
| **Data Ingestion** | Thu thập dữ liệu metadata bài hát, nghệ sĩ, album, thể loại từ Spotify API hoặc nguồn khác | Spotipy, Airflow/Dagster |
| **Data Validation & Cleaning** | Kiểm tra dữ liệu thiếu, trùng, chuẩn hóa tên, xử lý thể loại, chuyển đổi ngày tháng | Pandas, Great Expectations |
| **Feature Engineering** | Tạo vector đặc trưng cho bài hát dựa trên thể loại (genre), nghệ sĩ, album, popularity, explicit | TF-IDF, One-hot encoding, Embeddings |
| **Model Training** | Huấn luyện mô hình content-based (ví dụ: cosine similarity trên vector đặc trưng) | Scikit-learn, Faiss (vector search) |
| **Model Evaluation** | Đánh giá độ chính xác, độ phủ, độ đa dạng của gợi ý | Precision@K, Recall@K, MAP |
| **Model Deployment** | Triển khai mô hình dưới dạng API hoặc microservice để phục vụ gợi ý real-time | Docker, Kubernetes, FastAPI |
| **Monitoring & Retraining** | Giám sát hiệu suất mô hình, phát hiện drift dữ liệu, tự động retrain khi cần | Prometheus, MLflow, Airflow |