Data Engineering for AI Systems: Urdu News Classification Dashboard

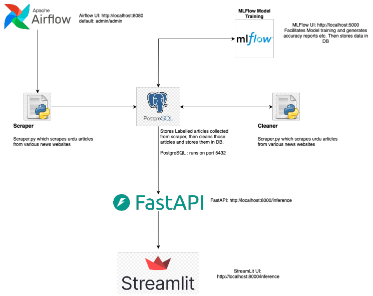
M. Affan Pasha, M. Mudasser Latif, M. Usama Asif

May 13, 2025

Project Overview

- **Product**: Urdu News Classification Pipeline
- **Goal**: Build an automated system to scrape, preprocess, and classify Urdu news articles into categories like sports, business, and entertainment using ML.
- **Theme**: Natural Language Processing for Urdu Language
- **Dataset**: Scraped from Urdu news websites (Express, Jang, Dunya, Samaa)
- **Team Contributions**:
  - M. Affan Pasha : Streamlit dashboard , PostgreSQL integration
  - M. Mudasser Latif : Airflow ETL , news scraping
  - M. Usama Asif : ML model pipeline , FastAPI integration

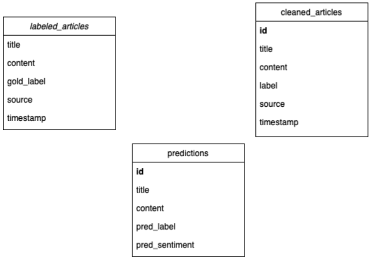
Architecture Diagram


Apache Airflow
Airflow UI: http://localhost:8080
default: admin/admin
MLFlow Model Training
mlflow
MLFlow UI: http://localhost:5000
Facilitates Model training and generates accuracy reports etc. Then stores data in DB
PostgreSQL
Scraper
Scraper.py which scrapes urdu articles from various news websites
Stores Labelled articles collected from scraper, then cleans those articles and stores them in DB.
PostgreSQL : runs on port 5432
Cleaner
Scraper.py which scrapes urdu articles from various news websites
FastAPI
FastAPI: http://localhost:8000/inference
Streamlit
StreamLit UI:
http://localhost:8000/inference

Explanation:

- **Scraper**: Collects Urdu articles (scrapper.py).
- **PostgreSQL**: Stores raw, cleaned, and predicted data.
- **Cleaner**: Preprocesses text using UrduHack (cleaner.py).
- **ML Model**: Trains classifier (train-model.py).
- **FastAPI**: Serves data (api.py).
- **Streamlit**: Displays results (streamlit-app.py).
- **Airflow**: Orchestrates pipeline (scrape-dag.py).

Schema Diagram



Sample Dataset:

- Title: "Pakistan Cricket Team Wins"
- Content: "Pakistan defeated India in T20 match..."
- gold-label: Sports
- timestamp: stored as a timestamp
- source: website name

Data Engineering Stages

- **Collection/Ingestion**: Scrapy and BeautifulSoup for web scraping. *Rationale*: Robust for handling dynamic Urdu news sites.
- **Cleaning/Transformation**: UrduHack for preprocessing. *Rationale*: Specialized for Urdu text normalization.
- **Storage/Modeling**: PostgreSQL with relational schema. *Rationale*: Scalable and supports complex queries.
- **Pipeline Orchestration**: Airflow for scheduling. *Rationale*: Reliable for dependency management.
- **Deployment/Frontend**: FastAPI and Streamlit. *Rationale*: FastAPI for efficient APIs, Streamlit for interactive UI.

AI Usage

- **Code Generation**: Used AI (e.g., GitHub Copilot) to assist in writing scraper templates and FastAPI endpoints.
- **Documentation**: AI-generated initial drafts for README.md and project documentation.
- **Debugging**: AI tools suggested fixes for Airflow DAG errors and UrduHack preprocessing issues.
- **Model Training**: AI-driven hyperparameter tuning via MLflow for the classifier.

Challenges

- **Web Scraping**: Inconsistent website structures required custom parsing logic per source.
- **Urdu Processing**: Limited Urdu NLP tools; UrduHack required additional stopwords customization.
- **Pipeline Stability**: Airflow DAG failures due to Docker networking issues.
- **Deployment**: Streamlit UI refresh delays after model updates.
- **Data Quality**: Handling noisy or incomplete articles from some sources.

Learnings

- **Steep Learning Curve:**
  - We used Airflow, UrduHack, Docker Compose, FastAPI, Streamlit, etc., *all for the first time*.
  - Containerization with Docker Compose was particularly challenging (networking, volumes, multi-service setup).
- **Technical Takeaways:**
  - Orchestrating end-to-end pipelines in Airflow.
  - Preprocessing and normalizing Urdu text with UrduHack.
  - Deploying microservices and dashboards reliably via Docker.
  - Building interactive UIs with FastAPI + Streamlit.
- **Key Outcome:** Hands-on mastery of modern data-engineering tools and workflows, turning initial hurdles into solid, reusable skills.

Demo and Links

- **Deployed Demo**: `http://localhost:8501` (Streamlit dashboard)
- **GitHub Repo**: `https://github.com/pashari/Urdu-News-Analysis-Dashboard-AI601`
- **Demo Video Link**: `https://drive.google.com/file/d/1Nm97WodRRmidm_F52GQQKT2xflL5OIow/view?usp=sharing`