Book Recommendation Project Plan

Phase 1: ETL Pipeline Design

Goal: Clean, validate, and structure raw book data.

Tools

•Extract: Pandas (CSV/JSON), Requests (APIs), BeautifulSoup (scraping).
•Transform: Pandas, NumPy, ISBNLib.
•Load: SQLAlchemy (SQLite/PostgreSQL).

Key Concepts

•Data validation (ISBN correction, missing value imputation).
•Normalization (text cleaning, tokenization).

Tasks

1.Extract data from CSV/API/scraped sources.
2.Validate and normalize book metadata (titles, authors, genres).
3.Load structured data into SQL databases.

Debugging Tips

•Encoding Errors: Use encoding='latin1' or chardet to detect file encodings.
•SQL Errors: Ensure foreign key consistency before loading.

Resources

•SQLAlchemy Documentation: https://docs.sqlalchemy.org/en/20/core/engines.html
•Pandas Data Cleaning Guide: https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html

---

Phase 2: Content-Based Recommendations

Goal: Recommend books using metadata and text descriptions.

Tools

•NLP: spaCy (en_core_web_sm), Scikit-Learn (TF-IDF, NearestNeighbors).

Key Concepts

•TF-IDF for term importance.
•Word embeddings (spaCy's CPU models).
•Genre/author similarity scoring.

Tasks

1. Generate text embeddings for book descriptions.
2. Compute similarity scores using cosine similarity.
3. Hybrid scoring (genre + author + description).

Debugging Tips

- spaCy Model Errors: Validate model installation with `python -m spacy validate`.
- Memory Issues: Use sparse matrices for large datasets.

Resources

- spaCy Troubleshooting Guide: https://spacy.io/usage#troubleshooting
- Scikit-Learn NearestNeighbors
Documentation: https://scikit-learn.org/stable/modules/neighbors.html

---

Phase 3: Graph-Based Recommendations

Goal: Leverage user-book interactions as a graph.

Tools

- Graph Analysis: NetworkX, Python-Louvain (community detection).

Key Concepts

- Node representation (users, books, authors).
- PageRank for influential books.
- Community detection for genre clusters.

Tasks

1. Build a graph from user-book interactions.
2. Apply PageRank and community detection.
3. Generate recommendations based on graph metrics.

Debugging Tips

- Slow Performance: Simplify node labels to integers.

Resources

- NetworkX Tutorial: https://networkx.org/documentation/stable/tutorial.html
- Python-Louvain Documentation: https://python-louvain.readthedocs.io/en/latest/

Phase 4: Neural Collaborative Filtering (NCF)

Goal: Train a neural network on user-book interactions.

Tools

•Deep Learning: PyTorch (CPU), Scikit-Learn.

Key Concepts

•Embedding layers for users/books.
•Loss optimization (binary cross-entropy).

Tasks

1.Design a neural network architecture.
2.Train on user-book interaction data.
3.Evaluate using ranking metrics (NDCG, Recall).

Debugging Tips

•NaN Loss: Normalize input ratings to [0, 1].

Resources

•PyTorch CPU Threading Guide: https://pytorch.org/docs/stable/notes/cpu_threading_torchscript.html
•Neural Collaborative Filtering Paper: https://arxiv.org/abs/1708.05031

Phase 5: Evaluation & Deployment

Goal: Deploy a CPU-friendly recommendation system.

Tools

•Deployment: Flask, Annoy (Approximate Nearest Neighbors).
•Caching: Joblib.

Key Concepts

•Ranking metrics (NDCG, Recall).
•Diversity scoring (genre spread).

Tasks

1.Evaluate model performance.
2.Deploy with Flask API.
3.Cache precomputed embeddings for speed.

Debugging Tips

- Annoy Build Issues: Install libannoy-dev for Linux systems.

Resources

- Annoy Documentation: https://github.com/spotify/annoy
- Flask Deployment Tutorial: https://flask.palletsprojects.com/en/2.3.x/tutorial/deploy/

Learning Resources

1. General Feature Engineering:
    - Feature Engineering for Machine Learning (Free eBook): https://fe4ml.feast.dev/
    - Scikit-Learn Feature Unions: https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.FeatureUnion.html
2. ETL & Data Wrangling:
    - Data Wrangling with Python (Automate the Boring Stuff): https://automatetheboringstuff.com/2e/chapter16/
3. Graph Analysis:
    - NetworkX in Practice (O'Reilly): https://www.oreilly.com/library/view/networkx-in-python/9781805123968/

Order of Execution

1. ETL Pipeline → 2. Content-Based Model → 3. Graph-Based Analysis → 4. Neural Collaborative Filtering → 5. Deployment.