boratonAJ / InstacartApp

Type / to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

InstacartApp

Private

1 Branch 0 Tags Go to file t Go to file Add file Code About

Olabode Ajayi and Olabode Ajayi Updated to include the QR Code

5f2fd40 · 1 minute ago 8 Commits

erd	code updated	17 minutes ago
flask_App	code updated	17 minutes ago
notebook	code updated	17 minutes ago
scripts	code updated	17 minutes ago
tools	Updated to include the QR ...	1 minute ago
.gitignore	Initial commit	last week
LICENSE	Initial commit	last week
README.md	Updated to include the QR ...	1 minute ago
data_dictionary.md	data dictionary file	5 days ago
instacart_repo_qr.png	Updated to include the QR ...	1 minute ago
instacart_repo_qr_s...	Updated to include the QR ...	1 minute ago

Instacart is an app for on-demand grocery shopping with same-day delivery service.

Readme
GPL-3.0 license
Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

README GPL-3.0 license

Instacart Basket Analysis (InstacartApp)

This repository contains code, notebooks and a small Flask app used for exploring and modeling the Instacart online grocery orders dataset. The goal of the project is to analyze purchase behaviour and build features/models that predict whether a product will be reordered by a user.

Suggested workflows

Based on your tech stack

Publish Python Package Configure
Publish a Python Package to PyPI on release.

Python application Configure
Create and test a Python application.

Pylint Configure
Lint a Python application with pylint.

More workflows Dismiss suggestions

Contents of this README

- Project overview
- File structure
- Setup & installation
- Requirements
- Usage (notebooks and Flask app)
- Modeling approach
- Notes & contact

Project overview

The original task is to predict whether a user will reorder a given product in their next order. The dataset is an anonymized sample of millions of orders and includes product, aisle and department metadata plus order metadata (day/hour, time between orders, etc.). Typical workflow in this repo:

- Data preprocessing and feature engineering
- Exploratory analysis (notebooks)
- Training models (neural, random-forest / gbm blends)
- Producing submission / visualizations

File structure (top-level of `InstacartApp`)

```
InstacartApp/
├─ data_dictionary.md
├─ LICENSE
├─ README.md  <-- (this file)
├─ erd/                    # ER diagrams and schema images
├─ flask_App/              # Small Flask demo app (API + front-end)
│  ├─ app.py
│  ├─ run.sh
│  ├─ requirements.txt
│  └─ templates/ static/
├─ notebook/               # Analysis notebooks (including Instacart_v3.ipynb)
│  └─ Instacart_v3.ipynb
├─ scripts/                # utility scripts (e.g., csv_visualizer.py)
└─ (other supporting files)
```

Note: The repository also references the larger dataset under `InstaCart_Online_Grocery_dataset/` at the project root when available. That folder is not included in this small app and should be downloaded separately (see Installation).

Setup & Installation

Recommended: use Python 3.9+ in a virtual environment. GPU is optional for some models but not required for the notebooks.

From the project root, create and activate a virtual environment (macOS / zsh):

```
python3 -m venv .venv
source .venv/bin/activate
pip install --upgrade pip
```

Install dependencies used by the notebooks and Flask app. There are multiple requirements files in the repo for different components; the most important are listed below. To install the Flask app deps:

```
cd InstacartApp/flask_App
pip install -r requirements.txt
cd ../..
```

If you want the ML/modeling environment, install the main requirements (example):

```
pip install numpy pandas scikit-learn seaborn matplotlib jupyterlab
# add GPU-enabled libraries if you plan to run TF on GPU
```

Dataset

Download the Instacart dataset (Kaggle) and place the CSV files in the top-level `InstaCart_Online_Grocery_dataset/` folder so paths in the notebooks and preprocessing scripts resolve correctly. Example Kaggle dataset page: https://www.kaggle.com/c/instacart-market-basket-analysis (or your mirror). Files used include `orders.csv`, `products.csv`, `order_products__prior.csv`, `order_products__train.csv`, `aisles.csv`, `departments.csv`.

Requirements

- Python 3.9+ (recommended)
- Jupyter / JupyterLab
- pandas, numpy, scikit-learn
- seaborn, matplotlib for visualization
- scikit-learn

Exact requirements for model training may be listed in `instacart-basket-prediction-master/requirements.txt` or other subfolders — install those if you plan to run the training pipelines.

Usage

Notebooks:

- Open `InstacartApp/notebook/Instacart_v3.ipynb` in JupyterLab or Jupyter Notebook. Ensure the dataset CSVs are available at the dataset folder path referenced by the notebook.

Flask demo:

```
cd InstacartApp/flask_App
# create .venv and install requirements then:
./run.sh
# or
python app.py
```

The Flask app provides a small web interface and APIs used to visualize results and simple queries.

Modeling approach (summary)

- Reframe the task as a binary classification problem: for each (user, product) pair, predict whether it will be reordered in the next order.
- Feature engineering includes user history (order counts, days between orders), product popularity, aisle/department aggregations, and sequence/order position features such as `add_to_cart_order` and `reordered` flags.
- Multiple models were explored: gradient-boosted trees (LightGBM), neural networks, and hybrid blending strategies. Models often used engineered features plus learned embeddings from sequence models.

Tips & Notes

- When working with the full dataset expect substantial memory usage — use sampling or chunked processing for development.
- `order_products__prior.csv` is the largest file and often dominates I/O and memory usage; consider using databases, Parquet, or Dask for larger experiments.

Examples — Common SQL queries

Here are a few ready-to-run SQL snippets used frequently in the notebooks. Replace table names or schema qualifiers if your environment differs.

- Top 20 most-ordered products:

```
SELECT p.product_id, p.product_name, COUNT(*) AS times_ordered
FROM fact_order_products op
JOIN dim_product p ON op.product_id = p.product_id
GROUP BY p.product_id, p.product_name
ORDER BY times_ordered DESC
LIMIT 20;
```

- Reorder rate by department (percentage):

```sql
SELECT d.department,
     AVG(op.reordered)::double precision AS reorder_rate
FROM fact_order_products op
JOIN dim_product p ON op.product_id = p.product_id
JOIN dim_department d ON p.department_id = d.department_id
GROUP BY d.department
ORDER BY reorder_rate DESC;
```

- User order history (with timing):

```sql
SELECT o.user_id, o.order_id, o.order_number, o.order_dow, o.order_hour_of_day, o.days_since_prior_order
FROM dim_order o
WHERE o.user_id = 12345
ORDER BY o.order_number;
```

Quick-start notebook cell

Drop this small cell into the top of `notebook/Instacart_v3.ipynb` to set up imports, display a sample of the `orders` table, and ensure the connection object `con` is available in the notebook kernel.

```python
# Quick-start: imports and sample data check
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# If you use a connection helper named 'con', confirm it exists and preview one table
try:
  display(con.sql("SELECT * FROM dim_order LIMIT 5").df())
except Exception as e:
  print("Could not query dim_order — check dataset path or connection. Error:", e)

# Set plotting defaults
sns.set_theme(style='whitegrid')
%matplotlib inline
```

System-specific setup notes

macOS (zsh) — recommended steps

```
# from project root
python3 -m venv .venv
source .venv/bin/activate
pip install --upgrade pip
pip install -r InstacartApp/flask_App/requirements.txt
pip install numpy pandas scikit-learn seaborn matplotlib jupyterlab
```

Docker (optional) — run notebooks in a container

```
# Example: start a jupyterlab container mounting project root
docker run --rm -it -p 8888:8888 -v "$PWD":/workspace -w /workspace python:3.10-slim bash
# inside container: pip install jupyterlab pandas seaborn matplotlib && jupyter lab --ip=0.0.0.0 --no-browser --
```

Troubleshooting & tips

- Missing data errors: confirm CSVs are in `InstaCart_Online_Grocery_dataset/` and that paths referenced by notebooks match your local layout.
- Memory issues: sample the data or work with `order_products__prior.csv` in chunks, or convert CSVs to Parquet and load with `pyarrow`.
- Re-running SQL/VIEW creation: some notebook queries create views (e.g., `top_products`) — if you encounter existing-view errors,

run `DROP VIEW IF EXISTS top_products;` before re-creating.