Supply Chain Analysis

Ge Gao, Yanhuan Huang, Kelly Kao,
Kojen Wang, Hongyi Zhan, Yusen Tang

Our Team

Ge Gao

Data Scientist

Kelly Kao

Data Scientist

Honyi Zhan

Data Scientist

Kojen Wang

Product Manager

Yanhuan Huang

Data Analyst/

Data Engineer

Yusen Tang

Data Engineer

Supply Chain Analysis

Make informed strategic decisions for inventory management, order fulfillment, and customer service.

Use Case 1: Demand Prediction

- Optimize Inventory Management
- Improve Production Planning
- Reduce Lead Times
- Impact on Sustainability

Use Case 2: Fraud Detection

- Prevent Financial Loss
- Preserve Brand Reputation
- Improve Operational Efficiency
- Enhance Customer Confidence

Dataset Overview

We obtained this dataset from Kaggle, and it's originally sourced from the company DataCo Global.

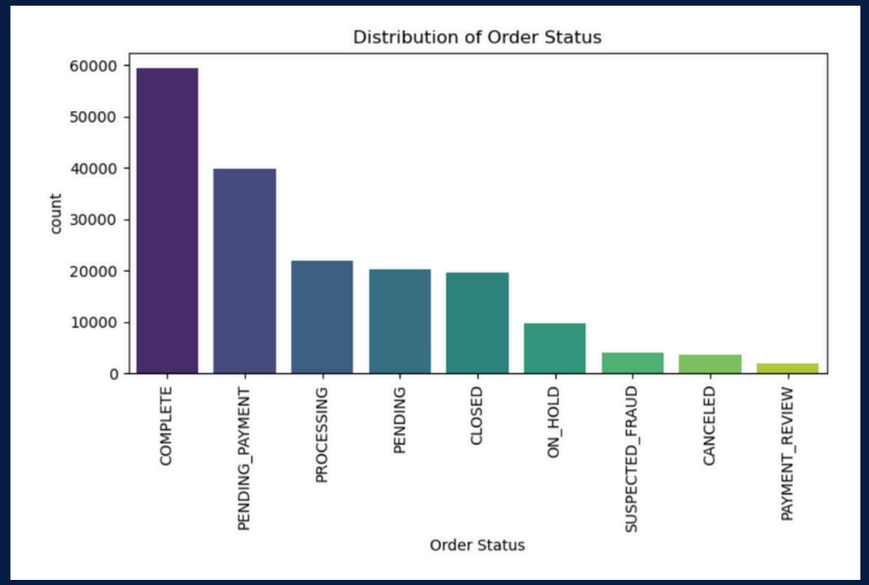
SupplyChainDataset

(180,519 rows, 53 columns)

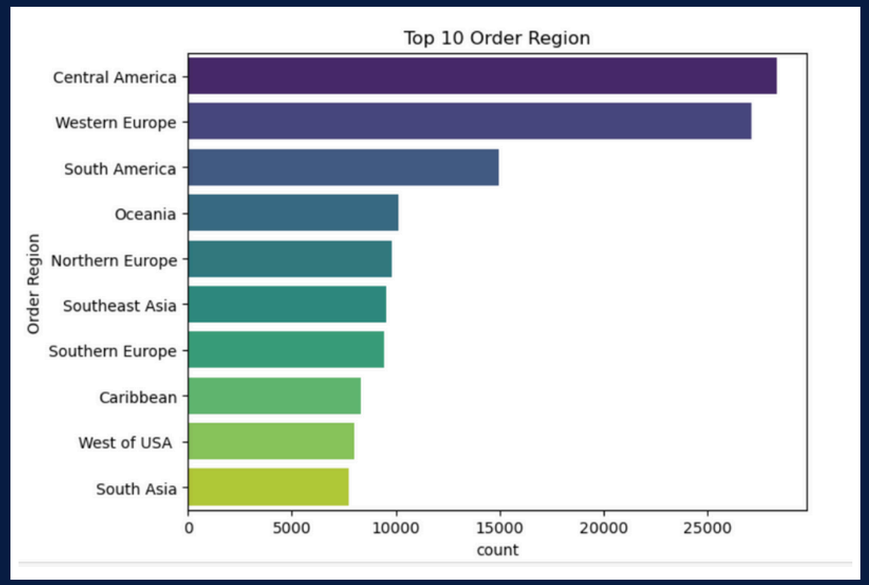
Contains order transaction records, customer data, product information, and logistics.

Key variables: order IDs, product categories, quantities ordered, customer locations, payment methods, and much more.

20,652	65,752	118
Customers	Orders	Products



Order status distribution- healthy completion rate



Regional sales volume- Central America a strong market

For:
Strategic planning
Targeting areas for market penetration
Managing fraud prevention measures

Model development

Phase 1:

Data Preprocessing ▶ **Feature Selection** ▶ **Hyperparameter Tuning** ▶ **Model Evaluation**

Phase 2:

Data Preprocessing ▶ **Using MLflow and H2O** ▶ **Using FairML** ▶ **Using LIME** ▶ **Building Data Pipeline**

- Handle missing values
- Drop unnecessary columns
- Aggregate data by ”Type”, “Customer City”, etc.
- Retrieve first 2000 rows of data to conduct following analysis

- Initianlize H2O machine learning platform
- Import dataset, then convert the dataframe to an H2O Frame.
- Begin the MLflow experiment
- Set up and train models using H2O’s AutoML, limit the number of models to 12 for manageability
- Log configuration parameters and metrics, such as number of models, or RMSE, and R^2
- Display the variable imprtance, and save the AutoML leaderboard
- Conclude the MLflow experiment to finailize logging and tracking

- Utilize FairML to evaluate the models by quantifying the relative significance of the model’s variables.
- Use relative significance to assess the fairness of the model

- Use LIME to explain the model’s prediction for individual instances

- Write class modules for each step and wrap all the steps into data pipeline, from data preprocessing to LIME.

Demand Prediction

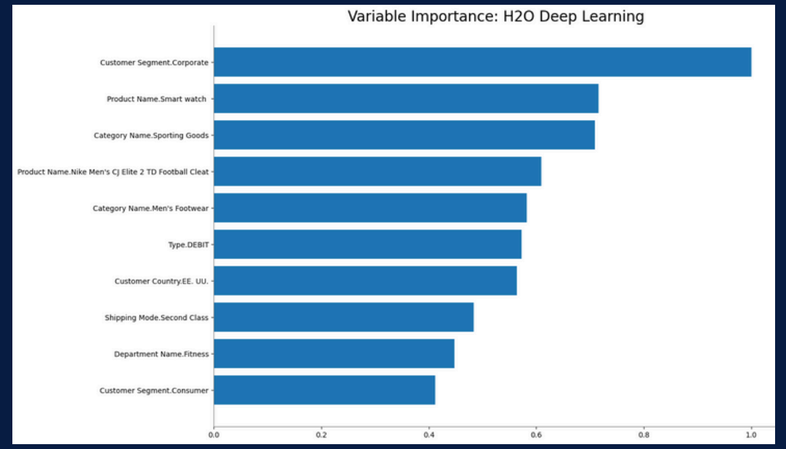
Dataset: 2000 rows, 22 columns
AutoML: H2O.ai
Max Model: 12
Final model: Deep Learning

0.368	0.135	0.97
RMSE	MSE	R^2

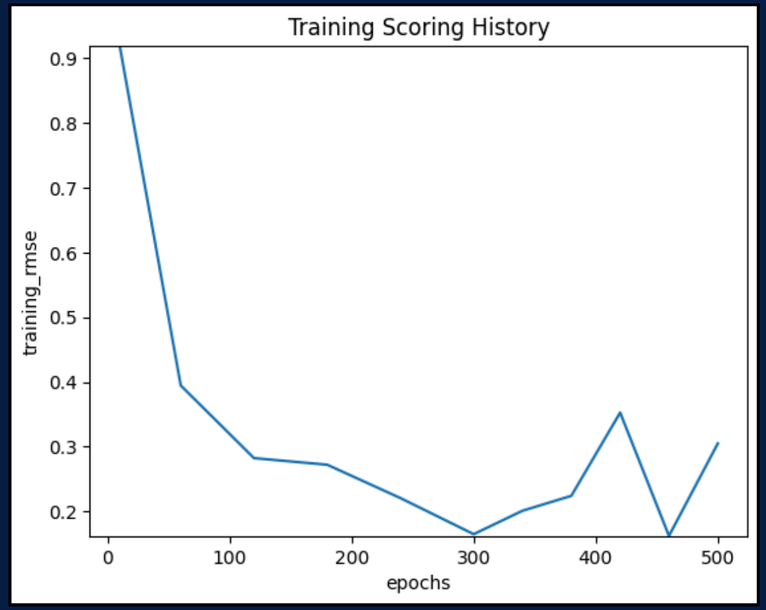
Performance Improvements:

- **R^2** of the model before improvement is 0.95.
- Compared with the current model trained by H2O, the R^2 improves making the predictions more precise.

Important Features: Customer Segments, Product Name, Category Name, Type of transaction



model_id	rmse	mse	mae	rmsle	mean_residual_deviance
DeepLearning_1_AutoML_8_20240423_184508	0.3676268329483933	0.1351494883036659	0.2325889133975714	0.1063516588197897	0.1351494883036659
DeepLearning_grid_1_AutoML_8_20240423_184508_model_1	0.4115395284984142	0.169364783516697	0.2712470574224213	0.1013377857799459	0.169364783516697
GBM_5_AutoML_8_20240423_184508	0.6871059032992741	0.4721145223487115	0.4089028703506105	0.1744965132640956	0.4721145223487115
DRF_1_AutoML_8_20240423_184508	0.6997479532858838	0.4896471981277834	0.4230201920384265	0.1986038925466346	0.4896471981277834
GBM_3_AutoML_8_20240423_184508	0.7236812847836829	0.5237146019461619	0.4309706368785145	0.1847206785736956	0.5237146019461619
GLM_1_AutoML_8_20240423_184508	0.7290367826174721	0.5314946304092353	0.5430779011160916	0.2406435894800427	0.5314946304092353
GBM_grid_1_AutoML_8_20240423_184508_model_2	0.7902044635201393	0.6244230941671511	0.4745873069733057	0.2026934960551283	0.6244230941671511
GBM_4_AutoML_8_20240423_184508	0.8218184024042675	0.6753854865303025	0.4953121019285273	0.2099515540891313	0.6753854865303025
GBM_1_AutoML_8_20240423_184508	0.871069967786706	0.758762888779933	0.5983315521998589	0.2318167807222585	0.758762888779933
GBM_grid_1_AutoML_8_20240423_184508_model_1	0.8770950233276824	0.7692956799461877	0.5844973682489182	0.2140532478936169	0.7692956799461877
GBM_2_AutoML_8_20240423_184508	0.900809722699172	0.8114581565093593	0.5620184133520677	0.2272327988279412	0.8114581565093593
XRT_1_AutoML_8_20240423_184508	0.9772223326468732	0.9549634874237958	0.7565407823650797	0.2311062384598983	0.9549634874237958



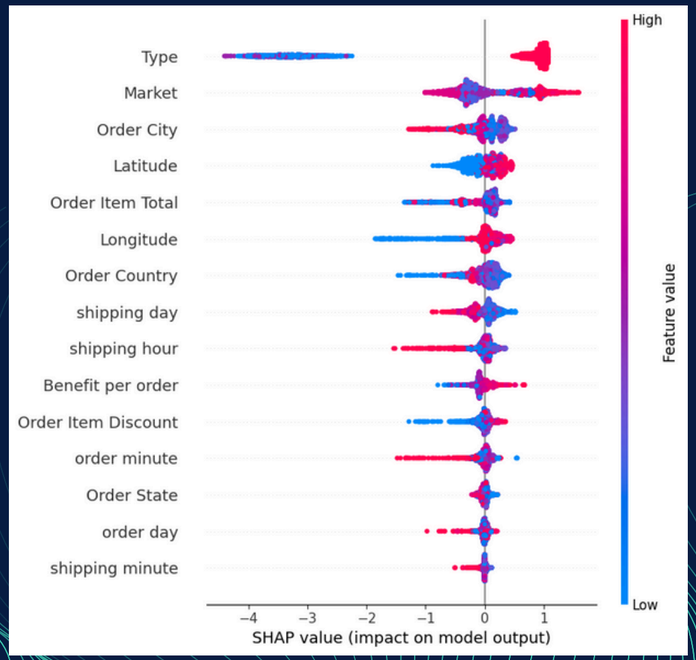
Fraud Detection

Dataset: 2000 rows, 22 columns
AutoML: H2O.ai
Max Model: 12
Final model: Gradient Boosting

0.859	0.579	0.692
F1 Score	Precision	Recall

Best Model Parameters

max depth	learning rate	n_estimators
5	0.1	50

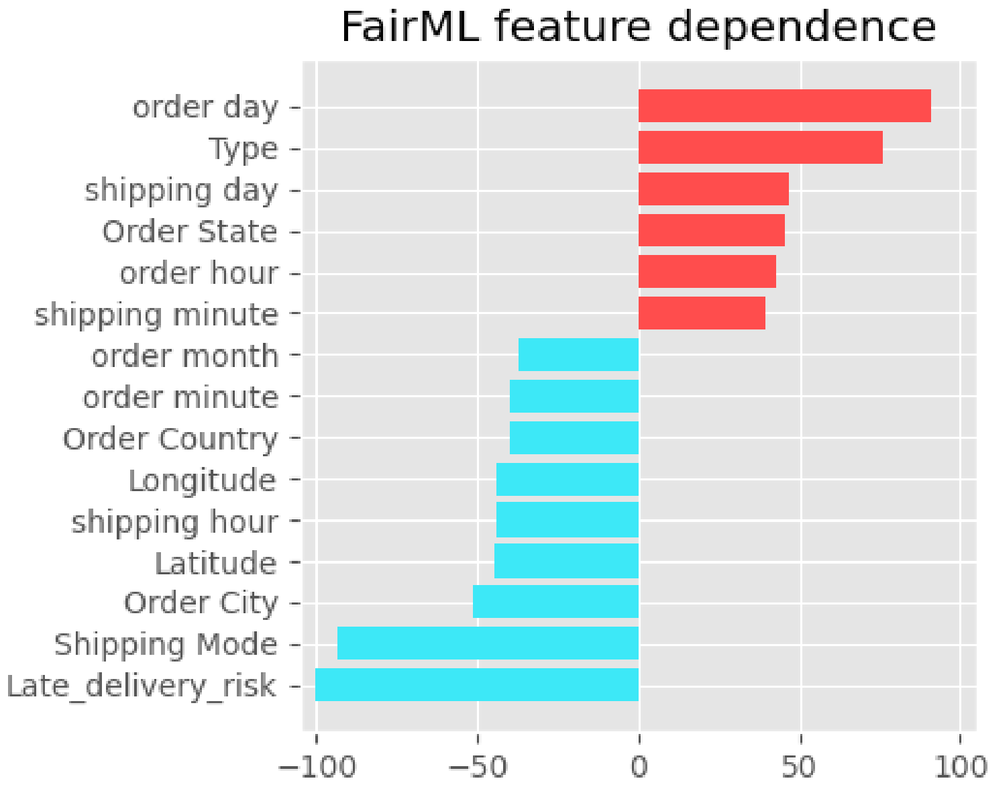


- **Type of transaction** (debit, transfer, payment, cash), **Market** (Africa , Europe , LATAM , Pacific Asia , USCA) are important features detecting fraudulent orders
- Obtained a higher **F1 score** compared to manual model comparison
- "**Type of Transaction**" remains the most critical feature for detecting fraudulent orders.
- The new variable "**Market**" has emerged as the second most important feature. "**Order City**" has increased in importance, ranking third in feature significance.

Fairness & Model Explainability

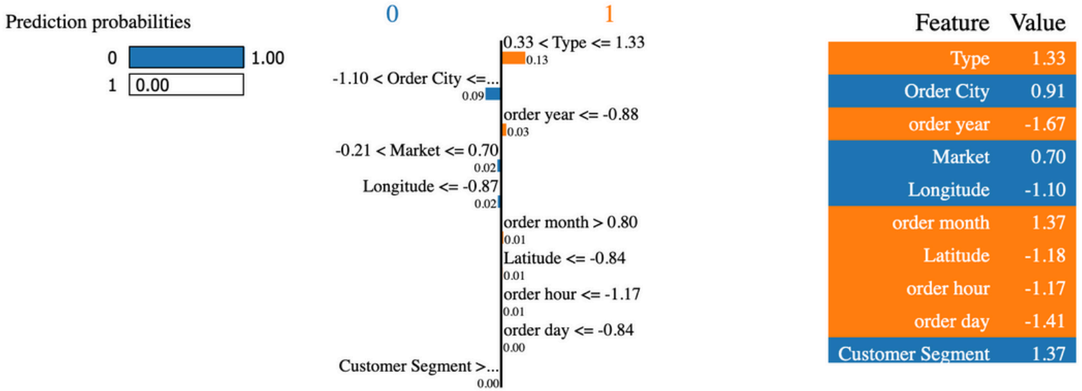
FairML

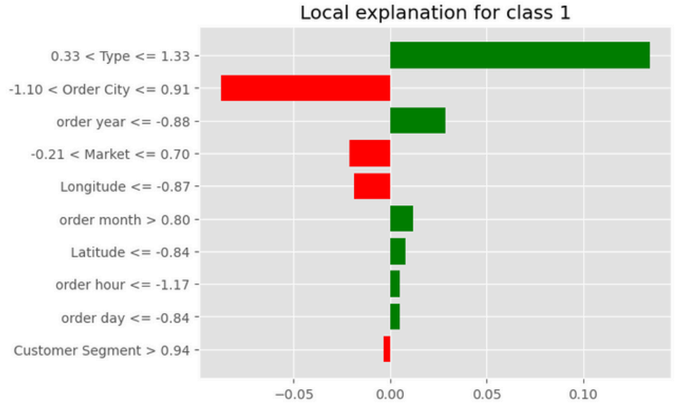
"Order day" and "Type" are relatively positively important to the model.



LIME

Take one of the instances to explain the reason of classifier. "Type" and "Order City" impact more on the decision.

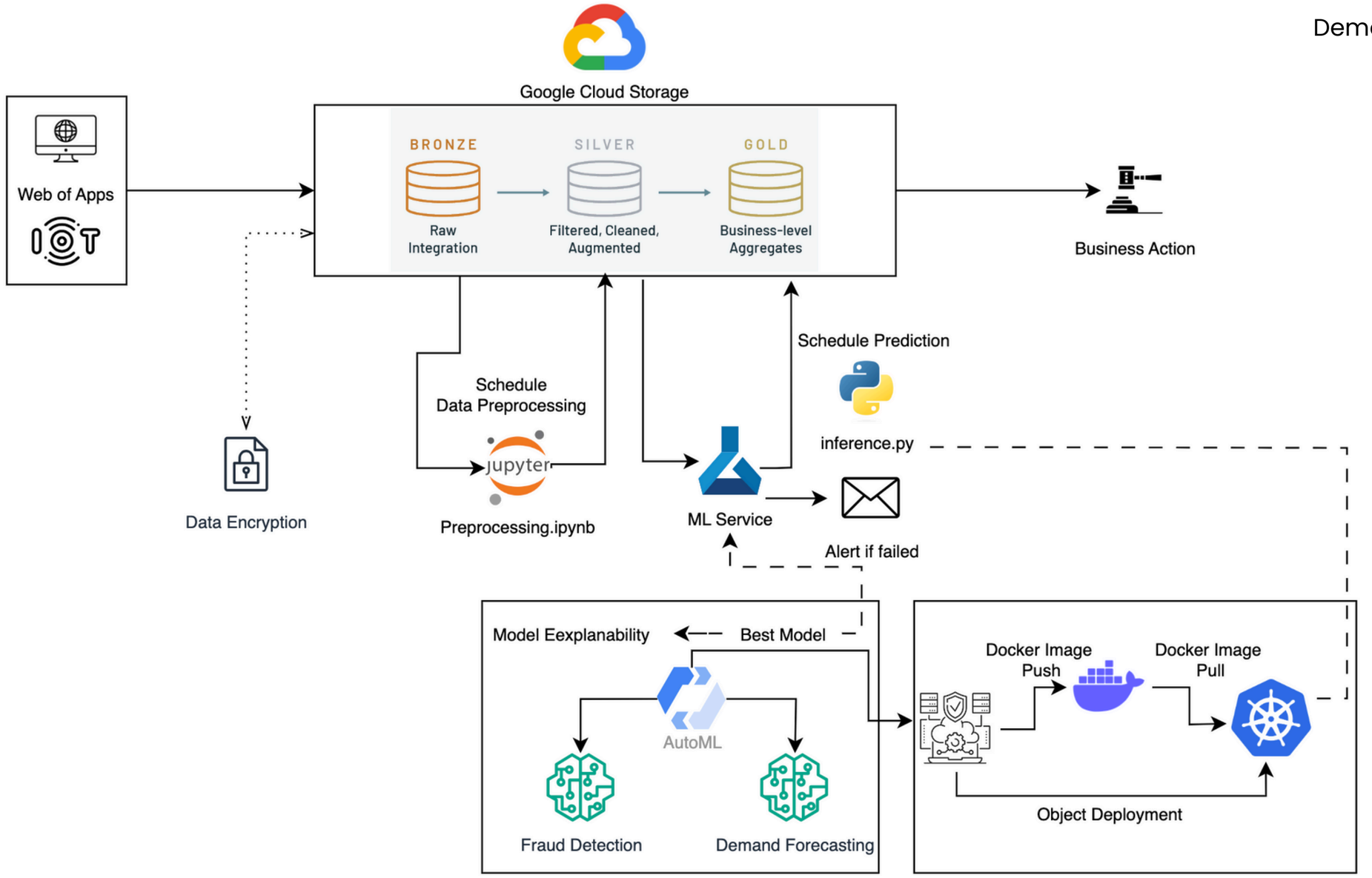




Using Fraud Detection model as an example

Architecture Flowchart

Demo video on the GitHub



Architecture steps

1 Local Environment Setup

- Models have been packaged
- Pull the code and model object from Github

2 Data Preparation

- Run preprocessing notebook to transform new data
- Prepare hold-out dataset for testing

3 Dockerization

- Create Inference.py
- Create Dockerfile
- Create requirements.txt
- Build Docker Image

4 Local Testing

- Test to ensure that model inference works as expected

5 Deploy to Cloud

- Create services on Cloud Run for FastAPI and Streamlit
- Configure the Streamlit to communicate with the FastAPI
- Test deployed service

6 Monitoring and Logging

- Utilize Google Cloud's monitoring and logging tools to keep track of the application's performance

7 Automate Data Processing

- Set up a system to automatically process raw data using the scripts, using Cloud Functions or Cloud Composer

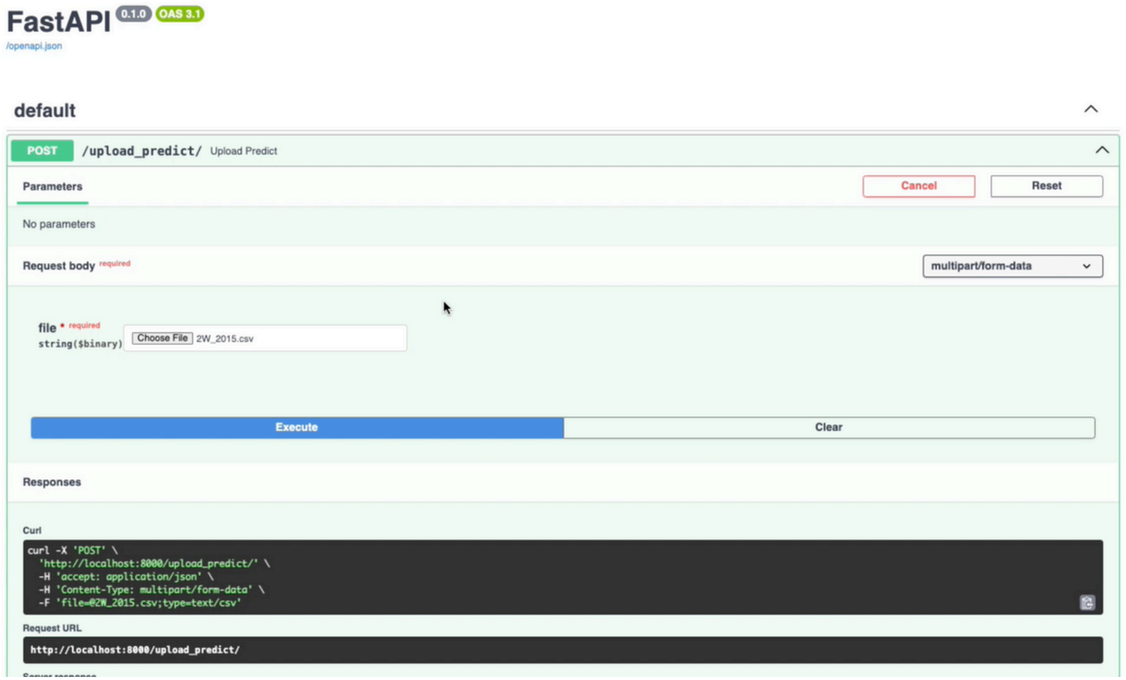
8 Test the Entire Service

- Ensure Stramlit UI, FastAPI, and monitor tool are working as expected

Backend and Frontend Interface

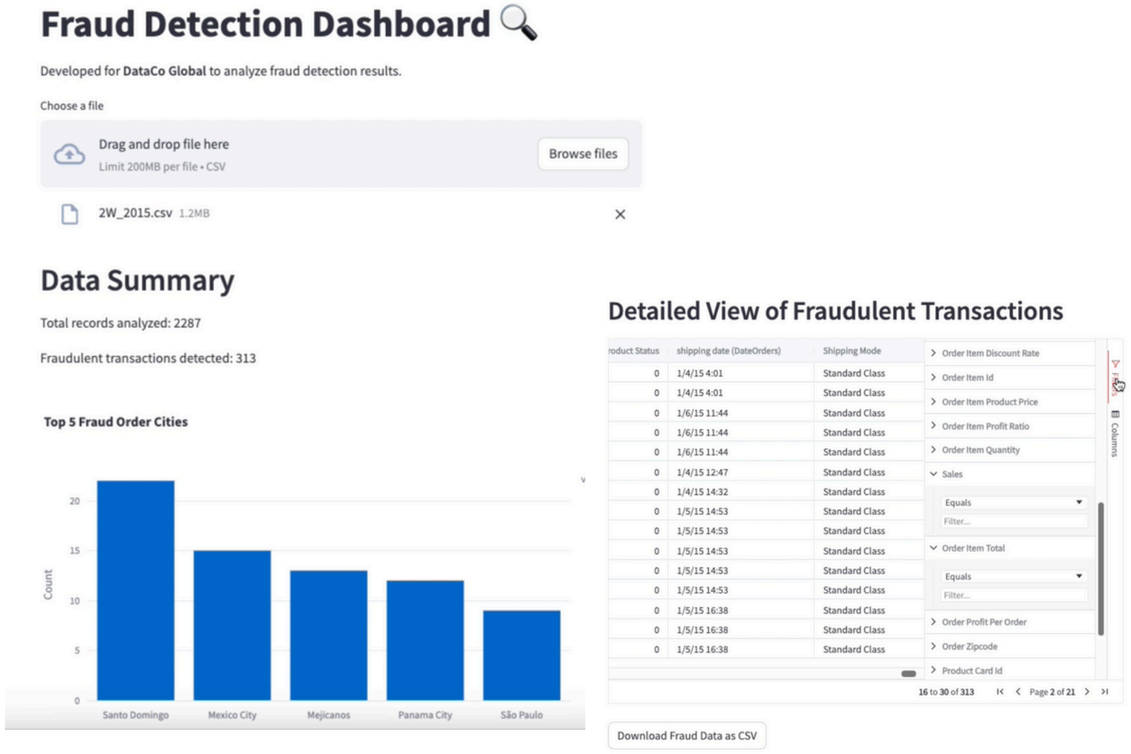
FastAPI

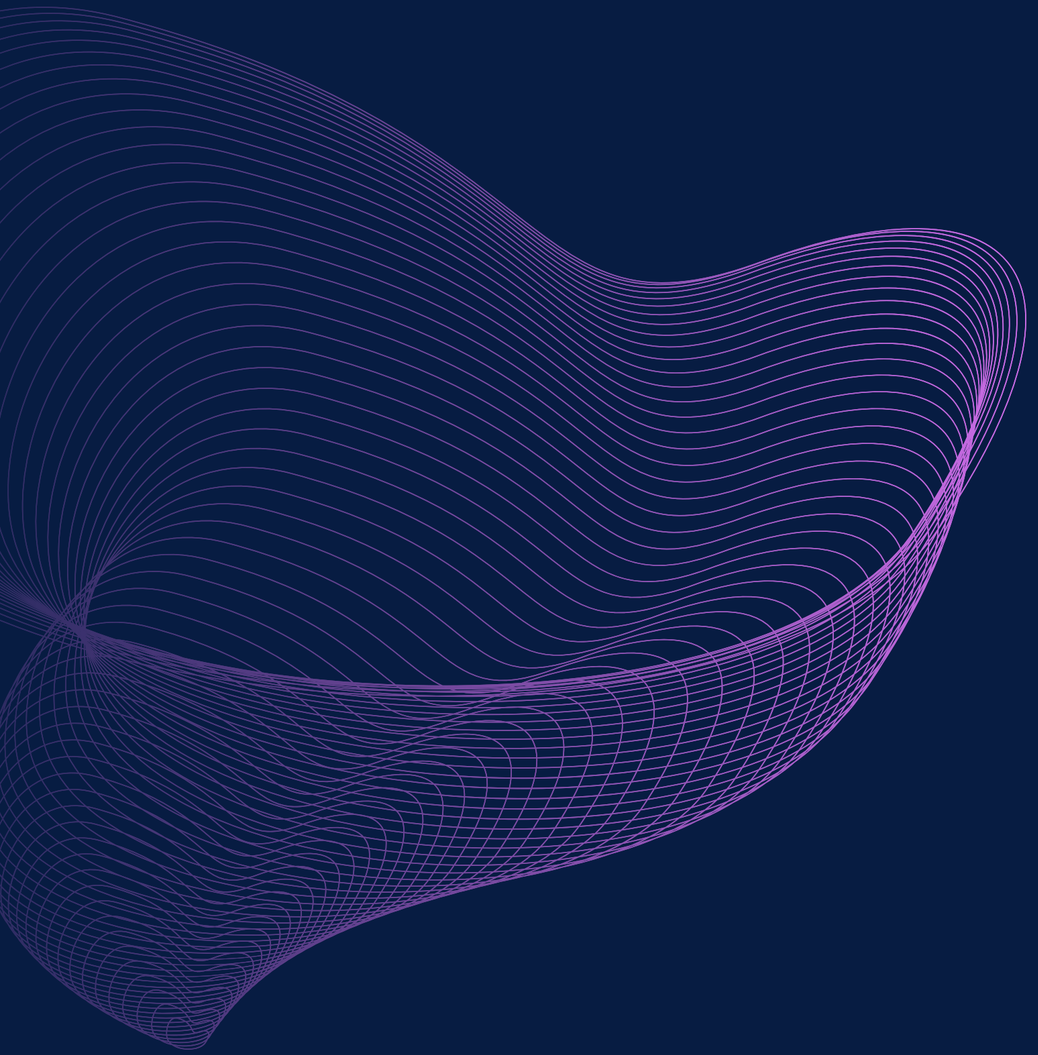
https://mlproject-smbfygbzda-uc.a.run.app/docs#/default/upload_predict_upload_predict__post



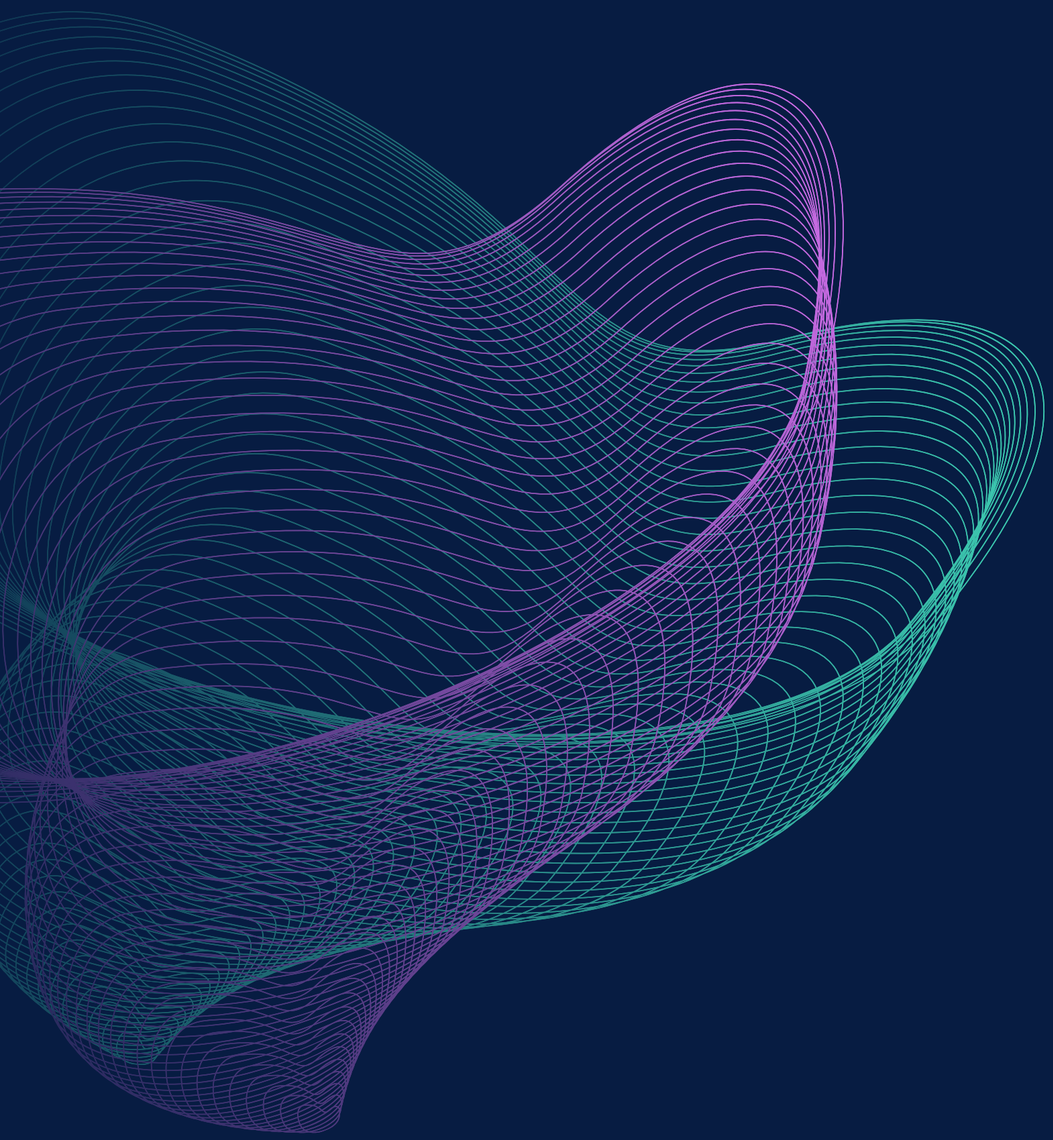
Streamlit

https://mlproject-streamlit-smbfygbzda-uc.a.run.app





Thank you!



Github

Repository: DataCo_Supply_Chain

https://github.com/McGill-MMA-EnterpriseAnalytics/DataCo_Supply_Chain/tree/main

Team Members Github ID

Ge Gao: lorine329
Kelly Kao: kellykaopeimin
Hongyi Zhan: HongyiZhan
Ko-jen Wang: kojen-coder
Yanhuan Huang: hyh-sherry
Yusen Tang: TeachFakerPlayingMid