Basdeo Thomas

ML&AI Capstone Project - Fraud Detection

Model Deployment Method

Background

Credit card fraud detection involves identifying and preventing unauthorized transactions using advanced technologies and strategies.

The nature and complexity of fraudulent activities are evolving rapidly. According to the FTC, fraud has been in an uptrend and is accelerating in that direction. The need for mitigating technologies have never been more relevant.

Deployment ML Methods

On-premises Deployment:

The application is hosted locally on local servers

Cloud Deployment:

The application is hosted on a cloud provider’s server. These providers unusually offer much more computer power which allows for scalability and flexibility.

Hybrid Deployment:

A combination of on-premises and cloud resources

Container Deployment:

Uses technologies like Docker or Kubernetes for lightweight, scalable deployments

Continuous Deployment:

Automates the release process for regular updates and faster delivery

Basdeo Thomas

Deployment Design

Transactions occur very frequently and in differing amounts. A continuous monitoring system would alert providers of potentially fraudulent transactions. Providers would then verify with the customer whether the translation was legitimate or not. Actions would then be taken to dispute the transaction. An on premise system would be ideal to keep costs low as long as the infrastructure is available. A CPU is sufficient.

Computation requirements (minimum)

CPU - Intel Core i9 or AMD Ryzen 9 (6 - 8 cores)
VRAM - 10-24
RAM - 32GB
Storage - SSD (1-2TB)
Internet - >= 100Mbps

Methodology

1. Collect Batch Data

- Define data sources: Internal databases, external APIs, IoT devices
- Implement data ingestion pipelines: Apache Kafka, AWS Kinesis
- Ensure data quality: Validate, cleanse, and transform raw data

2. Process Data

- Feature Engineering: Transform raw data into relevant model inputs
- Data normalization and scaling techniques: StandardScaler, MinMaxScaler
- Dimensionality reduction (if necessary): Principal Component Analysis (PCA)

3. Make Predictions

- Deploy pre-trained model for inference
- Optimize for delay using batch processing.

4. Log Data

- Capture model inputs and predictions for analysis
- Store data in a data lake or warehouse: AWS S3, Google Cloud Storage

5. Check Accuracy

- Compare predictions against actual outcomes: Confusion matrix, ROC curves
- Calculate key performance metrics: Accuracy, precision, recall, F1-score
- Establish performance thresholds: Trigger retraining based on degradation
- Implement anomaly detection algorithms: Identify unexpected data shifts

6. Retrain Model & ReDeploy Model

- Trigger retraining pipeline based on accuracy checks
- Implement model versioning and rollback capabilities
- Test retrained model thoroughly before deployment
- Ensure seamless model deployment: Canary deployments, A/B testing
- Example: Retraining the churn model with updated customer data and deploying it to replace the existing model with minimal downtime