**Data Science PRD: Fraud Detection in Financial Transactions**

**1. Business Problem Statement**

**Overview**

Financial fraud is a significant challenge for banks, fintech companies, and e-commerce platforms. Fraudulent transactions result in millions of dollars in losses annually, impacting both businesses and customers. Traditional rule-based fraud detection systems are ineffective at identifying evolving fraud patterns, leading to false positives and missed fraudulent activities.

**Business Need**

Our goal is to develop a machine learning-based fraud detection system that can accurately classify transactions as fraudulent or legitimate in real-time. This system should:

* Reduce chargebacks and fraud losses by improving detection accuracy.
* Minimize false positives to avoid blocking legitimate transactions.
* Enable real-time decision-making through API-based model deployment.

**Stakeholders**

* Fraud Prevention Team: Responsible for manual fraud investigations.
* Engineering Team: Responsible for data pipelines, model deployment, and integration.
* Product Team: Ensures fraud detection aligns with business objectives.
* Compliance Team: Ensures legal and regulatory adherence (e.g., GDPR, PCI-DSS).

**2. Project Goals & Objectives**

**Short-Term Goals**

* Build a baseline ML model that achieves at least 80% fraud detection accuracy.
* Deploy a proof-of-concept fraud scoring API that can handle real-time transactions.
* Develop a BI dashboard for fraud analysts to monitor model performance.

**Long-Term Goals**

* Optimize the model to achieve 90% recall and an F1-score above 85%.
* Implement continuous learning mechanisms to adapt to evolving fraud patterns.
* Deploy the model into production with real-time transaction scoring.

**3. Data Requirements & Sources**

**Datasets**

**Internal Data Sources:**

* Transaction Logs (amount, timestamp, location, device ID, payment method).
* Customer Profiles (account age, transaction history, credit risk score).
* Fraud Labels (previously verified fraudulent transactions).

**External Data Sources:**

* Third-party fraud scoring APIs (e.g., device fingerprinting, location anomalies).
* IP geolocation services to detect suspicious transactions.

**Data Collection Plan**

* Streaming data ingestion for real-time scoring.
* ETL pipelines to process and store historical fraud data.
* BigQuery/PostgreSQL as the primary data warehouse.

**4. Model Scope & Constraints**

**Modeling Approach**

* Supervised Learning:
  + Classification using XGBoost, Random Forest, or Logistic Regression.
  + Imbalanced dataset handling using SMOTE, class weighting, or anomaly detection.
* Anomaly Detection:
  + Isolation Forest, Autoencoders, One-Class SVM.
* Explainability Considerations:
  + Use SHAP values to provide transparency for fraud analysts.

**Constraints**

* Latency must be under 100ms for real-time transaction scoring.
* Explainability is required, limiting deep black-box models.
* Regulatory compliance (GDPR, PCI-DSS) must be followed.

**5. Evaluation Metrics**

**Primary Metrics:**

* Recall (≥ 90%) – Detect the majority of fraud cases.
* F1-score (≥ 85%) – Balance precision and recall.
* False Positive Rate (FPR) – Minimize legitimate transaction rejections.
* ROC-AUC – Measure overall fraud detection performance.

**Secondary Metrics:**

* Inference latency (<100ms) for real-time fraud detection.
* Business impact metrics (fraud loss reduction, chargeback rate decline).

**6. Deployment Plan & Engineering Considerations**

**Model Deployment Strategy**

* Serve model as a REST API using FastAPI or Flask.
* Containerized deployment with Docker and Kubernetes.
* Real-time fraud scoring via AWS Lambda or GCP Cloud Run.

**Integration Requirements**

* Connect to existing transaction processing systems.
* Expose API endpoints for fraud scoring in real-time.
* BI dashboard (Tableau, Power BI) for fraud monitoring.

**7. Risks & Assumptions**

**Potential Risks**

* Data Drift: Fraud patterns change frequently; model retraining is necessary.
* Scalability Issues: High transaction volume may require optimized infrastructure.
* Bias & Fairness: False positives could unfairly block legitimate transactions.

**Assumptions**

* Sufficient historical fraud labels exist for supervised learning.
* Integration with transaction processing systems is feasible.
* Fraud investigation teams will use model outputs for verification.

**8. Timeline & Milestones**

* Week 1-2: Data collection and preprocessing.
* Week 3-4: Model development and baseline evaluation.
* Week 5-6: Model optimization and validation.
* Week 7-8: API deployment and dashboard integration.
* Week 9+: Post-deployment monitoring and model retraining setup.

**9. Ownership & Responsibilities**

**Stakeholders & Roles**

* Data Scientists: Feature engineering, model development, and optimization.
* Data Engineers: ETL pipeline setup and data storage optimization.
* ML Engineers: Model deployment and inference pipeline setup.
* Business Analysts: Model performance evaluation and impact analysis.
* Compliance and Risk Teams: Ensuring regulatory and ethical AI considerations.

**10. Post-Deployment Monitoring & Iteration**

* Monitor data drift and retrain model periodically (monthly/quarterly).
* Set up an automated feedback loop to update fraud labels.
* Deploy A/B testing mechanisms to evaluate new model iterations.