Project Specification Document

Title: Credit Card Fraud Detection – Model Tuning and API Deployment
Project Type: Machine Learning and Software Deployment
Version: 1.0

1. Introduction

This document defines the technical requirements, functional specifications, and evaluation criteria for the development and deployment of a machine learning model designed to detect fraudulent credit card transactions.

The system will implement the complete machine learning workflow, including data exploration, feature analysis, model training, parameter optimization, performance evaluation, and deployment as a RESTful API service.

The underlying dataset consists of anonymized credit card transactions made by European cardholders. Each transaction is labeled as either legitimate or fraudulent. The goal is to construct a classification model capable of accurately predicting the likelihood of fraud based on transaction characteristics.

2. Dataset Description

2.1 General Overview

Source: Kaggle – Credit Card Fraud Detection dataset (European cardholders, September 2013).
Total Records: 284,807 transactions.
Target Variable: Class (renamed to is_fraud).
Classes:
0 – Legitimate transaction
1 – Fraudulent transaction

2.2 Feature Description

The dataset contains 30 input variables. Twenty-eight features (V1–V28) were obtained through Principal Component Analysis (PCA) applied to confidential cardholder and transaction data. Two additional features (Time and Amount) correspond to real transactional information.
For interpretability, each variable has been mapped to a descriptive name with an indicative interpretation, without altering the original dataset values.

Original Name	Descriptive Name	Interpretation
Time	transaction_time	Time in seconds since the first recorded transaction. Represents the temporal sequence of operations.
Amount	transaction_amount	Monetary value of the transaction in euros.
V1	customer_behavior_score	Aggregated behavioral metric reflecting the user's historical spending profile.
V2	merchant_trust_index	Indicator of merchant reliability and transaction reputation.
V3	transaction_risk_factor	Encodes deviations related to transaction irregularities or anomalies.
V4	location_activity_score	Geospatial activity consistency across customer transactions.
V5	account_balance_ratio	Relationship between transaction value and estimated account balance.
V6	device_usage_signature	Identifies device or network usage characteristics.
V7	transaction_velocity	Frequency or speed of transactions within a specific period.
V8	merchant_activity_density	Density of transactions linked to a particular merchant.
V9	customer_credit_trend	Long-term pattern of customer credit utilization.
V10	fraud_likelihood_component	Latent variable correlated with potential fraudulent behavior.
V11	payment_pattern_score	Regularity and predictability of payment behavior.
V12	geo_transaction_variability	Degree of variation in transaction locations.
V13	account_tenure_indicator	Proxy for customer tenure or financial maturity.
V14	transaction_trust_score	Composite reliability measure for individual transactions.
V15	merchant_behavior_score	Consistency and stability of merchant behavior over time.
V16	account_activity_score	Level of financial activity associated with the account.
V17	customer_risk_profile	Overall customer risk profile derived from behavioral metrics.
V18	fraud_signal_intensity	Magnitude of fraud-related indicators within the transaction.
V19	transaction_cluster_score	Grouping metric linking transactions of similar characteristics.
V20	device_risk_factor	Device-related risk assessment measure.
V21	geo_activity_anomaly	Spatial anomaly indicator relative to historical user patterns.
V22	account_credit_ratio	Relationship between credit utilization and available limit.

V23	merchant_risk_level	Relative exposure level of the merchant to suspicious transactions.
V24	transaction_pattern_shift	Magnitude of deviation from previously observed customer patterns.
V25	customer_habit_stability	Degree of consistency in purchasing behavior.
V26	spending_anomaly_index	Indicator of spending deviation compared to historical trends.
V27	session_behavior_score	Consistency of user behavior within a transaction session.
V28	payment_method_signature	Pattern characteristic of the payment method used.
Class	is_fraud	Binary target variable: 1 for fraud, 0 for legitimate transaction.

3. Project Activities

The project involves data exploration, model development, tuning, evaluation, and deployment. Each stage contributes to building a fully operational predictive system.

3.1 Data Exploration and Analysis

Exploration of the dataset to understand its structure and characteristics, including descriptive statistics, visualization of transaction distributions, comparison of fraudulent and legitimate cases, correlation analysis, and identification of key patterns.

3.2 Data Preparation

Preparation of data for modeling, including renaming columns, partitioning the dataset into training and testing subsets, and feature scaling as necessary. Class balancing is not required.

3.3 Model Development and Parameter Optimization

Training and optimization of a classification model using either automated optimization tools such as TPOT or manual hyperparameter tuning. Documentation of algorithm choice, parameter space, and performance comparison is required.

3.4 Model Evaluation

Assessment of model performance using confusion matrix and classification report. Results must include interpretation and discussion of potential improvements.

3.5 Model Deployment

Deployment of the final model as a RESTful API using Flask. The service must provide a '/predict' endpoint and respond in JSON format.

4. Functional Requirements

Functional requirements define the system capabilities and behaviors necessary for achieving project objectives.

ID	Requirement	Description

FR-01	Data ingestion	System must load and preprocess the dataset.
FR-02	Data analysis	System must generate descriptive and graphical summaries.
FR-03	Model training	System must implement a binary classification model.
FR-04	Model tuning	System must optimize hyperparameters manually or automatically.
FR-05	Model evaluation	System must produce confusion matrix and classification report.
FR-06	Model serialization	Trained model must be saved for deployment.
FR-07	API development	Flask API must serve prediction requests.
FR-08	Output format	API must return predictions in JSON format.

5. Non-Functional Requirements

Non-functional requirements define quality attributes and system constraints.

ID	Requirement	Description
NFR-01	Performance	Prediction latency must not exceed one second per request.
NFR-02	Input validation	API must validate requests and reject malformed inputs.
NFR-03	Reproducibility	Results must be deterministic through fixed random seeds.
NFR-04	Maintainability	Codebase must be modular and well-documented.
NFR-05	Security	API must prevent unauthorized access or injection attacks.
NFR-06	Deployment environment	Solution must run locally using Flask and support containerization.
NFR-07	Documentation	All steps must be clearly documented and reproducible.

6. Deliverables

1. Exploratory Analysis Report: Overview of data structure, distribution, and insights.
2. Trained Model Artifact: Serialized model file ready for deployment.
3. Deployment Application: Flask-based API implementing fraud prediction.
4.Technical Documentation: Description of dataset, training summary, and API usage instructions.