Credit Card Fraud Detection

Project Documentation

Machine Learning Models Implementation

K-Nearest Neighbors (KNN) & Linear Regression (SGDRegressor)

Dataset:	Credit Card Fraud Detection Dataset 2023
Total Samples:	568,630
Number of Classes:	2 (Fraudulent / Non-Fraudulent)
Features:	29 Dimensions

December 10, 2025

Contents

1 Executive Summary

This document provides comprehensive documentation for the Credit Card Fraud Detection project, which implements two machine learning models to identify fraudulent transactions:

1. **K-Nearest Neighbors (KNN) Classifier**
2. **Linear Regression (SGDRegressor)**

Both models were trained on the Credit Card Fraud Detection Dataset 2023, containing 568,630 transactions with 29 features each.

Table 1: Model Performance Summary on Test Set

Metric	**KNN (K=2)**	**Linear Regression**
Accuracy	99.87%	91.52%
AUC-ROC	0.9988	0.9585
False Positives	139	3,152
False Negatives	8	6,497

The KNN classifier clearly outperforms the Linear Regression model on all key classification metrics.

2 Model 1: K-Nearest Neighbors (KNN) Classifier

2.1 General Information on Dataset

Table 2: Dataset Specifications for KNN Model

Property	Value
Name of Dataset	Credit Card Fraud Detection Dataset 2023 (creditcard_2023.csv)
Number of Classes	2
Class Labels	**Label 0:** Non-Fraudulent Transaction **Label 1:** Fraudulent Transaction
Total Number of Samples	568,630
Size of Each Sample	Tabular data (29 features per sample)
Data Split Ratio	60% / 20% / 20%

Table 3: Data Split Distribution

Split	**Percentage**	**Number of Samples**
Training Set	60%	341,178
Validation Set	20%	113,726
Testing Set	20%	113,726
Total	**100%**	**568,630**

2.2 Implementation Details

2.2.1 Feature Extraction Phase

Table 4: Feature Extraction Details

Property	Description
Number of Features Extracted	29
Feature Names	V1–V28, Amount
Dimension of Input Vector	29 dimensions
Normalization Method	StandardScaler (z-score normalization)

2.2.2 Cross-Validation

Table 5: Cross-Validation Configuration

Property	Value
Cross-Validation Used	No
Alternative Approach	Dedicated 20% Validation Set for hyperparameter tuning
Purpose	Used specifically for tuning the K hyperparameter

2.2.3 Hyperparameters

Table 6: KNN Hyperparameters

Hyperparameter	Value
Model	KNeighborsClassifier (sklearn)
Key Hyperparameter (K)	Number of neighbors (`n_neighbors`)
Optimal K Found	**2**
Tuning Range	K tested from 1 to 30
Distance Metric	Minkowski (Euclidean distance, $p = 2$)
Weights	Uniform (all neighbors weighted equally)
Optimizer/Epochs	Not applicable (non-parametric model)

2.3 Results Details (on Testing Data)

2.3.1 Loss Curve (Error Rate)

The optimal hyperparameter K was selected by minimizing the Validation Error Rate (1 - Accuracy) on the Validation Set.

Table 7: Error Rate Analysis

Metric	Value
Selection Criterion	Minimum Validation Error Rate
Minimum Validation Error	≈ 0.0010 (0.10%) at $K = 2$
Optimal K Value	2

2.3.2 Accuracy

Test Accuracy: 0.9987 (99.87%)

2.3.3 Confusion Matrix

Table 8: Confusion Matrix for KNN Model (K=2)

	Predicted 0	**Predicted 1**
Actual 0	TN: 56,724	FP: 139
Actual 1	FN: 8	TP: 56,855

Table 9: Classification Metrics Derived from Confusion Matrix

Metric	**Value (Approx.)**
True Negatives (TN)	56,724
True Positives (TP)	56,855
False Positives (FP)	139
False Negatives (FN)	8
Precision	0.9976
Recall (Sensitivity)	0.9999
Specificity	0.9976

2.3.4 ROC Curve

Area Under Curve (AUC): 0.9988

3 Model 2: Linear Regression (SGDRegressor)

3.1 General Information on Dataset

Table 10: Dataset Specifications for Linear Regression Model

Property	Value
Name of Dataset	Credit Card Fraud Detection Dataset 2023 (creditcard_2023.csv)
Number of Classes	2
Class Labels	**Label 0:** Non-Fraudulent Transaction **Label 1:** Fraudulent Transaction
Total Number of Samples	568,630
Size of Each Sample	Tabular data (29 features per sample)
Data Split Ratio	60% / 20% / 20%

Table 11: Data Split Distribution

Split	**Percentage**	**Number of Samples**
Training Set	60%	341,178
Validation Set	20%	113,726
Testing Set	20%	113,726
Total	**100%**	**568,630**

3.2 Implementation Details

3.2.1 Feature Extraction Phase

Table 12: Feature Extraction Details

Property	Description
Number of Features Extracted	29
Feature Names	V1–V28, Amount
Dimension of Input Vector	29 dimensions
Normalization Method	StandardScaler (z-score normalization)

3.2.2 Cross-Validation

Table 13: Cross-Validation Configuration

Property	Value
Cross-Validation Used	No
Alternative Approach	Dedicated 20% Validation Set
Purpose	Loss monitoring and Early Stopping

3.2.3 Hyperparameters

Table 14: Linear Regression (SGDRegressor) Hyperparameters

Hyperparameter	Value
Model	`sklearn.linear_model.SGDRegressor`
Loss Function	`'squared_error'` (Mean Squared Error)
Optimizer	Stochastic Gradient Descent (SGD)
Initial Learning Rate (η_0)	0.01
Regularization (penalty)	L2 (Ridge regularization)
Regularization Strength (α)	0.0001
Maximum Epochs	50
Early Stopping	Enabled
Early Stopping Patience	5 epochs
Triggered at Epoch	6
Best Model Epoch	1

3.3 Results Details (on Testing Data)

3.3.1 Loss Curve

The model's training was tracked using the Mean Squared Error (MSE) loss on both the training and validation sets.

Table 15: Training Loss Analysis

Metric	Value
Loss Function	Mean Squared Error (MSE)
Training Behavior	Stopped prematurely due to Early Stopping
Early Stopping Triggered	Epoch 6
Best Model Selected From	Epoch 1
Best Validation MSE	0.103400

3.3.2 Overall Test Metrics

Table 16: Global Metrics on Testing Data (after 0.5 Thresholding)

Metric	Value
Test Accuracy	0.9152 (91.52%)
Test Mean Squared Error (MSE)	0.102807
Test R^2 Score	0.5888
Test AUC-ROC	0.9585

3.3.3 Confusion Matrix

The continuous output was binarized using a threshold of 0.5 for classification metrics.

Table 17: Confusion Matrix for Linear Regression Model (Threshold = 0.5)

	Predicted 0	**Predicted 1**
Actual 0	TN: 53,711	FP: 3,152
Actual 1	FN: 6,497	TP: 50,366

Table 18: Classification Metrics Derived from Confusion Matrix

Metric	**Value (Approx.)**
True Negatives (TN)	53,711
True Positives (TP)	50,366
False Positives (FP)	3,152
False Negatives (FN)	6,497
Precision	0.9411
Recall (Sensitivity)	0.8857
Specificity	0.9446

3.3.4 ROC Curve

Area Under Curve (AUC): 0.9585

4 Comparative Analysis

Table 19: Complete Model Comparison on Testing Data

Metric	KNN (K=2)	Linear Regression
Performance Metrics		
Test Accuracy	99.87%	91.52%
AUC-ROC Score	0.9988	0.9585
Confusion Matrix Results		
True Negatives	56,724	53,711
True Positives	56,855	50,366
False Positives	139	3,152
False Negatives	8	6,497
Model Characteristics		
Model Type	Non-parametric	Parametric (linear)
Training Required	No	Yes (SGD)
Key Hyperparameter	$K = 2$	$\eta_0 = 0.01$, $\alpha = 0.0001$
Regularization	N/A	L2 (Ridge)

4.1 Key Findings

1. **Overall Performance:** The KNN classifier substantially outperforms the Linear Regression model across all standard classification metrics (accuracy, AUC, false positives, and false negatives).

2. **Fraud Detection Capability:**

   - KNN misses only 8 fraudulent transactions (false negatives) on the test set.
   - Linear Regression misses 6,497 fraudulent transactions, which is unacceptable for a fraud detection system.

3. **Discrimination Ability:**

   - KNN achieves an AUC of 0.9988, indicating near-perfect ranking ability.
   - Linear Regression still has a reasonably high AUC of 0.9585 but is clearly inferior to KNN.

5 Conclusion

This project implemented and evaluated two machine learning approaches for credit card fraud detection:

1. **K-Nearest Neighbors (KNN):** Achieved 99.87% test accuracy and an AUC of 0.9988. It produced very few misclassifications (139 false positives and 8 false negatives out of 113,726 test samples) and demonstrated excellent capability for detecting fraudulent transactions.
2. **Linear Regression (SGDRegressor):** Achieved 91.52% test accuracy, AUC of 0.9585, MSE of 0.102807, and R^2 of 0.5888. Although its regression metrics are reasonable, its classification performance is significantly worse than KNN, with a high number of false negatives (6,497).

5.1 Recommendations

For production deployment in a fraud detection context, the **KNN model (K=2)** is strongly recommended because:

- It has much higher accuracy and AUC.
- It yields dramatically fewer false negatives, which is critical in fraud detection.
- False positives are also very low compared to the Linear Regression model.

The Linear Regression model can still be useful as:

- A baseline model for comparison.
- A fast scoring model when approximate risk estimation (rather than strict classification) is acceptable.