SE 3007 - Introduction to Machine Learning

CREDIT RISK CLASSIFICATION USING MACHINE LEARNING TECHNIQUES

OĞUZ YÖRÜR
HALİL TOPRAKÇI

THE ENGINEERING BOTTLENECK

The Problem: Manual credit assessment is slow and prone to human error.

The Solution: Developing a system that automatically adapts to individual user patterns

Manual Bank Statement Verification vs. Automated Bank Statement Verification	
Labor-intensive	Easy and efficient
Uses manual comparison and paper-based documentation	Uses OCR and AI-powered IDR technology
Time-consuming and costly	Time-saving and cost-effective
Lengthy processing time	Quicker
Not scalable	Scalable
Reactive approach to fraud	Proactive tracking of fraudulent activities
Reports lack deeper insights	Precise insights by utilizing custom-trained ML algorithms
High risk of breach or unauthorized access	Enhanced security
Ongoing expense and training overheads	High implementation costs

LOSS FUNCTIONS & ERROR TYPES.

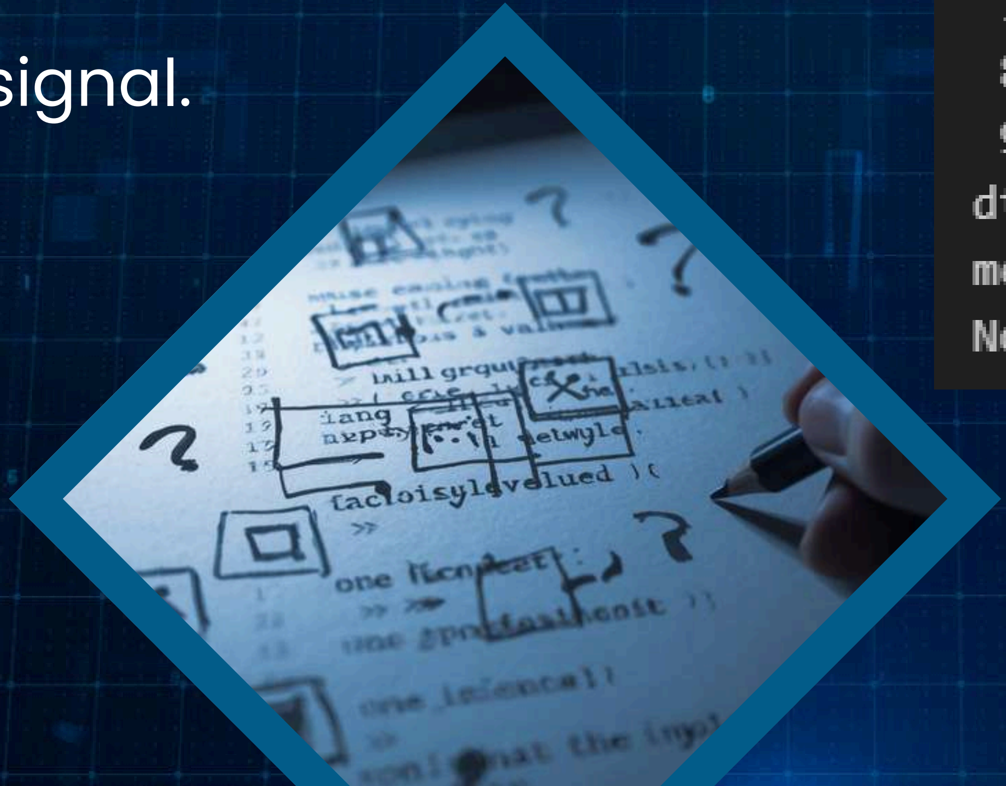
- **Objective:** Minimize Financial Loss, not just Error Rate.
- **Type I Error (False Positive):** Rejecting a good customer. (Cost: Lost Interest).

- **Type II Error (False Negative):** Approving a bad customer. (Cost: Capital Loss).
- **Our Metric:** Maximize Recall (Sensitivity) for the 'Bad' class.

DATASET & THE "UNKNOWN" HEURİSTİC

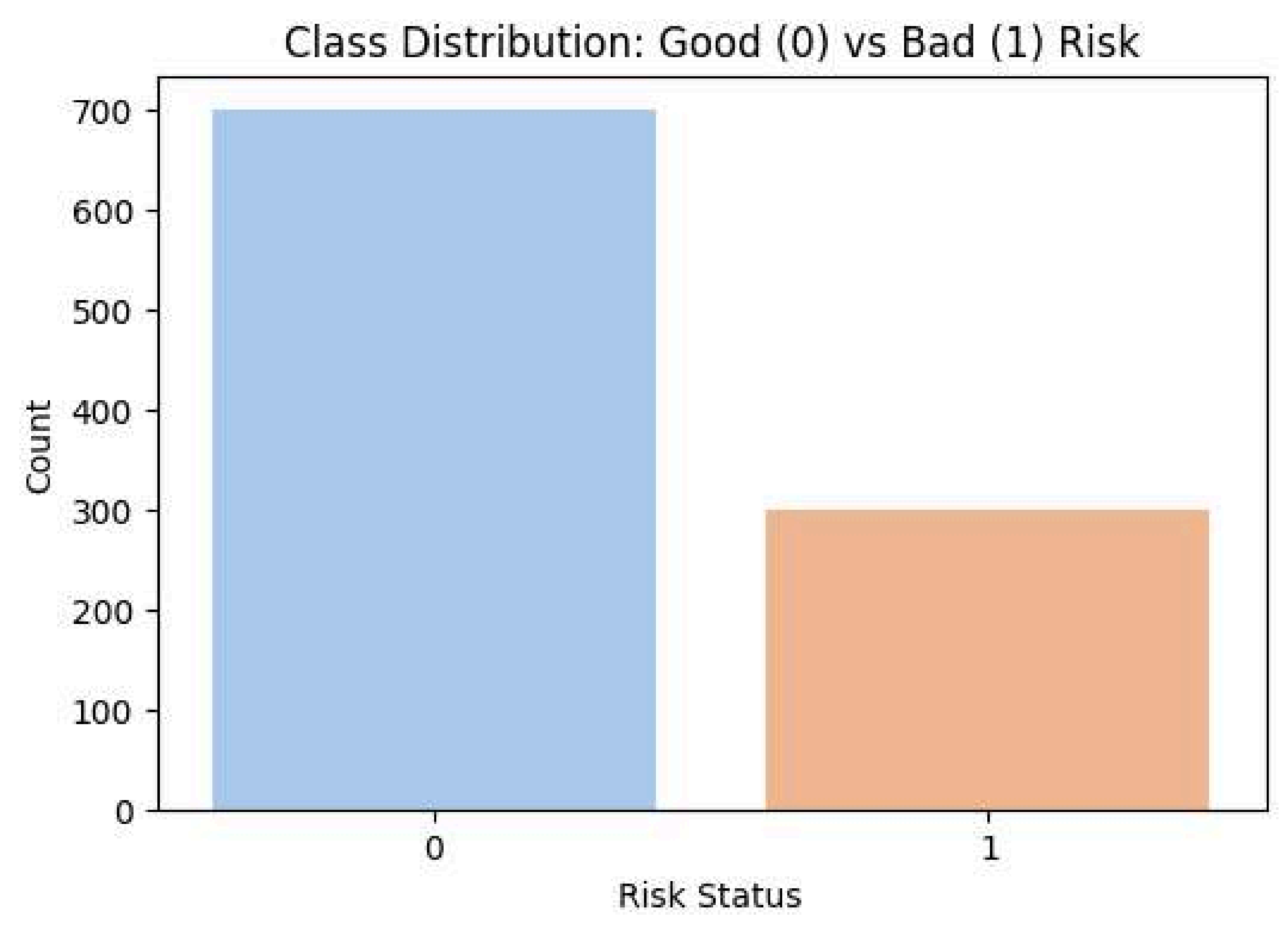
- Dataset: German Credit Data (1000 Samples).
- Problem: High rate of NaN in financial accounts.
- Strategy: Imputation with a new class: "Unknown".
- Theory: Missingness is a signal.

```
✅ Dataset Loaded: (1000, 10)
✅ Data Fixed: 'Credit amount' & 'Duration' are now numeric.
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Age               1000 non-null   int64
 1   Sex               1000 non-null   object
 2   Job               1000 non-null   int64
 3   Housing           1000 non-null   object
 4   Saving accounts   1000 non-null   object
 5   Checking account  1000 non-null   object
 6   Credit amount     1000 non-null   int64
 7   Duration          1000 non-null   int64
 8   Purpose           1000 non-null   object
 9   Risk              1000 non-null   int64
dtypes: int64(5), object(5)
memory usage: 78.3+ KB
None
```



CLASS IMBALANCE & THE SMOTE SOLUTION

- Imbalance: 70% Good / 30% Bad.
- Risk: Model Bias towards the majority class.
- Solution: SMOTE (Synthetic Minority Over-sampling Technique).
- Engineering Constraint: Applied ONLY to Training Data (N=800 to N=1120).



PREPROCESSING & VECTORIZATION

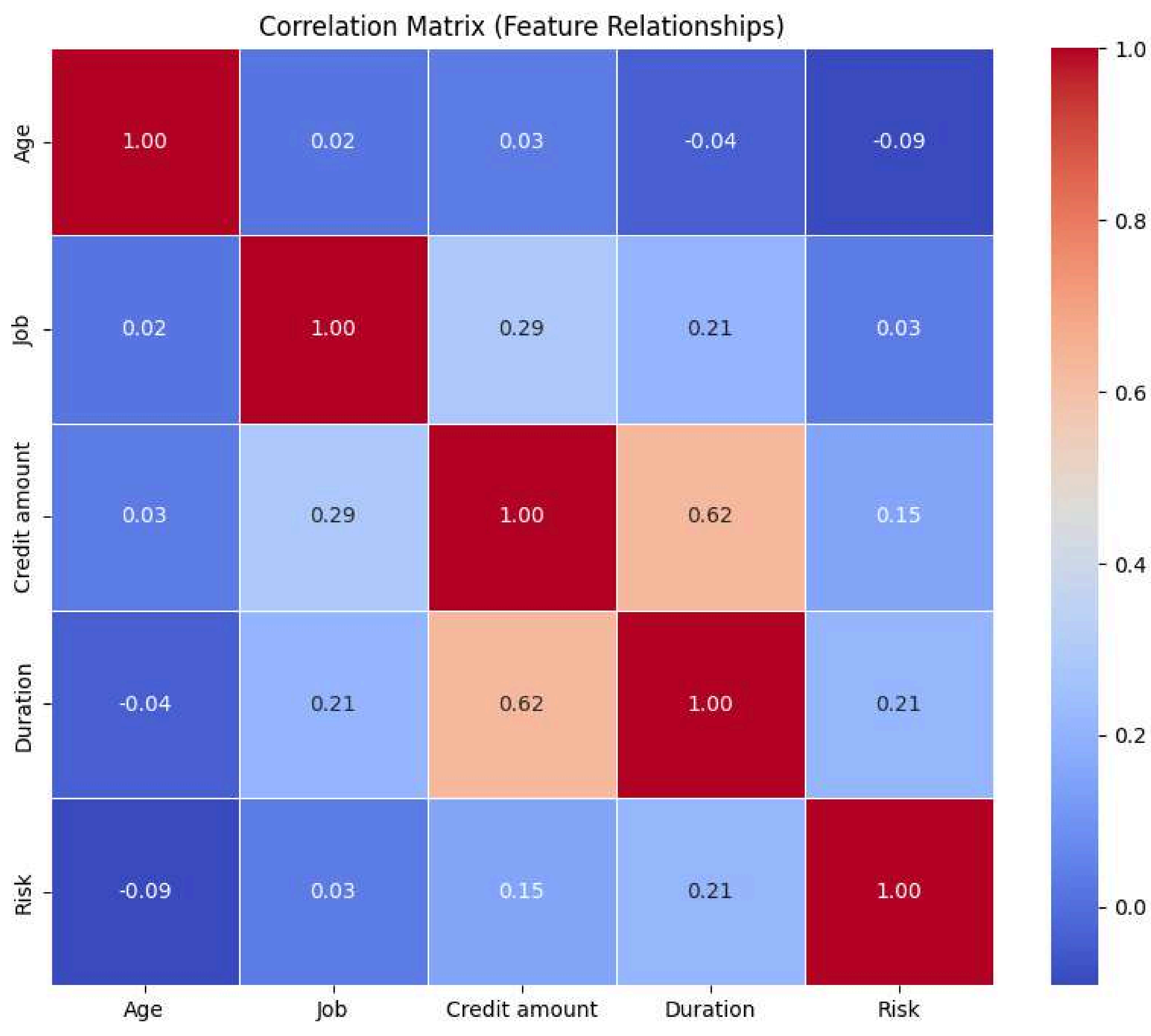
- Encoding: Converting Categorical (Sex, Job) to Numerical (One-Hot).
- Scaling: StandardScaler $(z = \frac{x-\mu}{\sigma})$.
- Why Scale? To prevent large magnitude features (e.g., Credit Amount) from dominating gradients or distance metrics.

```
# 1. One-Hot Encoding (Categorical -> Numerical)
X = df.drop('Risk', axis=1)
y = df['Risk']
X_encoded = pd.get_dummies(X, drop_first=True)
```

```
# 3. Scaling (StandardScaler)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
joblib.dump(scaler, 'scaler.pkl')
```

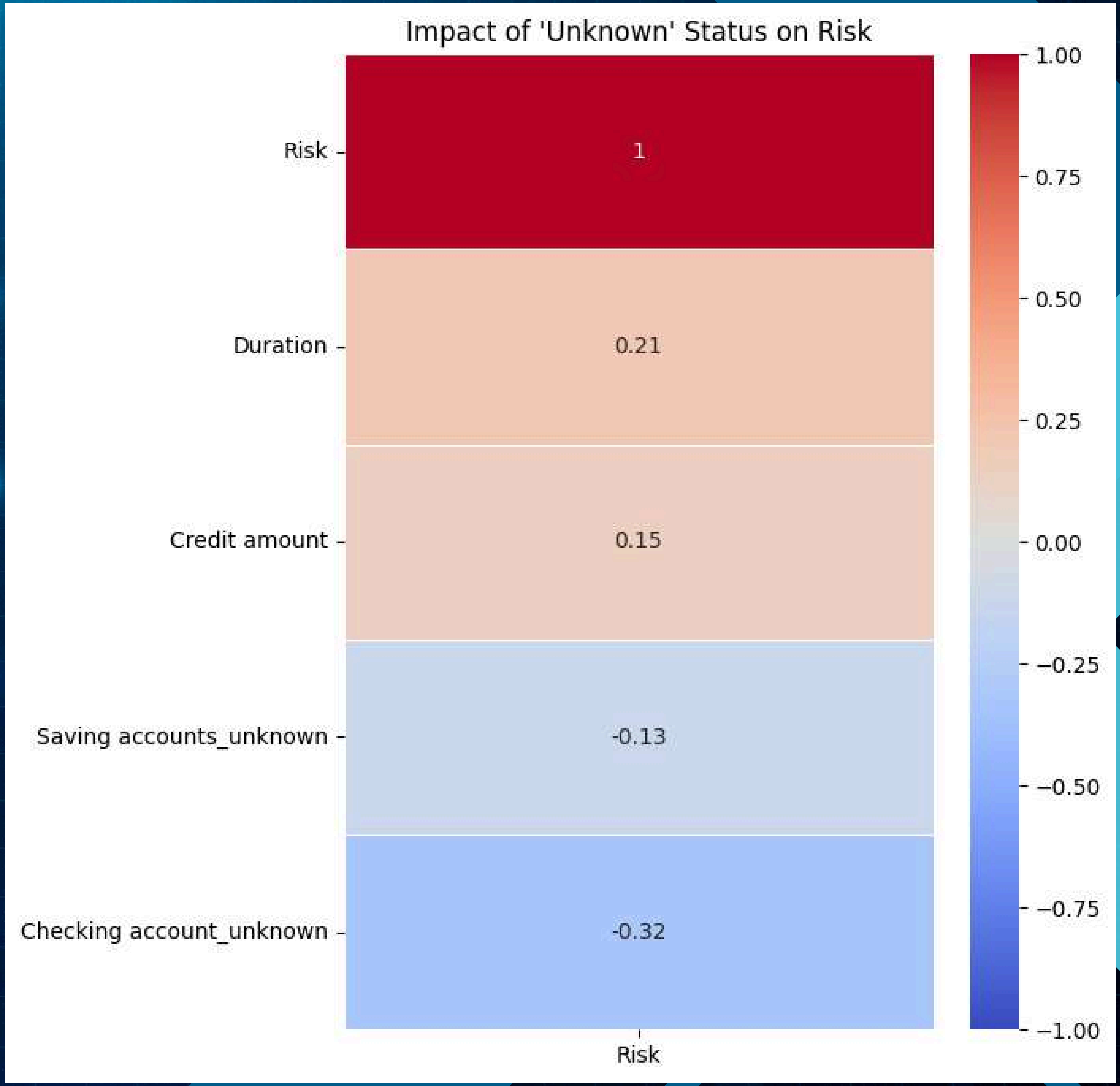
EXPLORATORY DATA ANALYSIS

- Observation 1: Duration is positively correlated with Risk.
- Observation 2: Checking_Unknown is negatively correlated with Risk (Safety Signal).
- Observation 3: Data is not perfectly linearly separable in 2D.



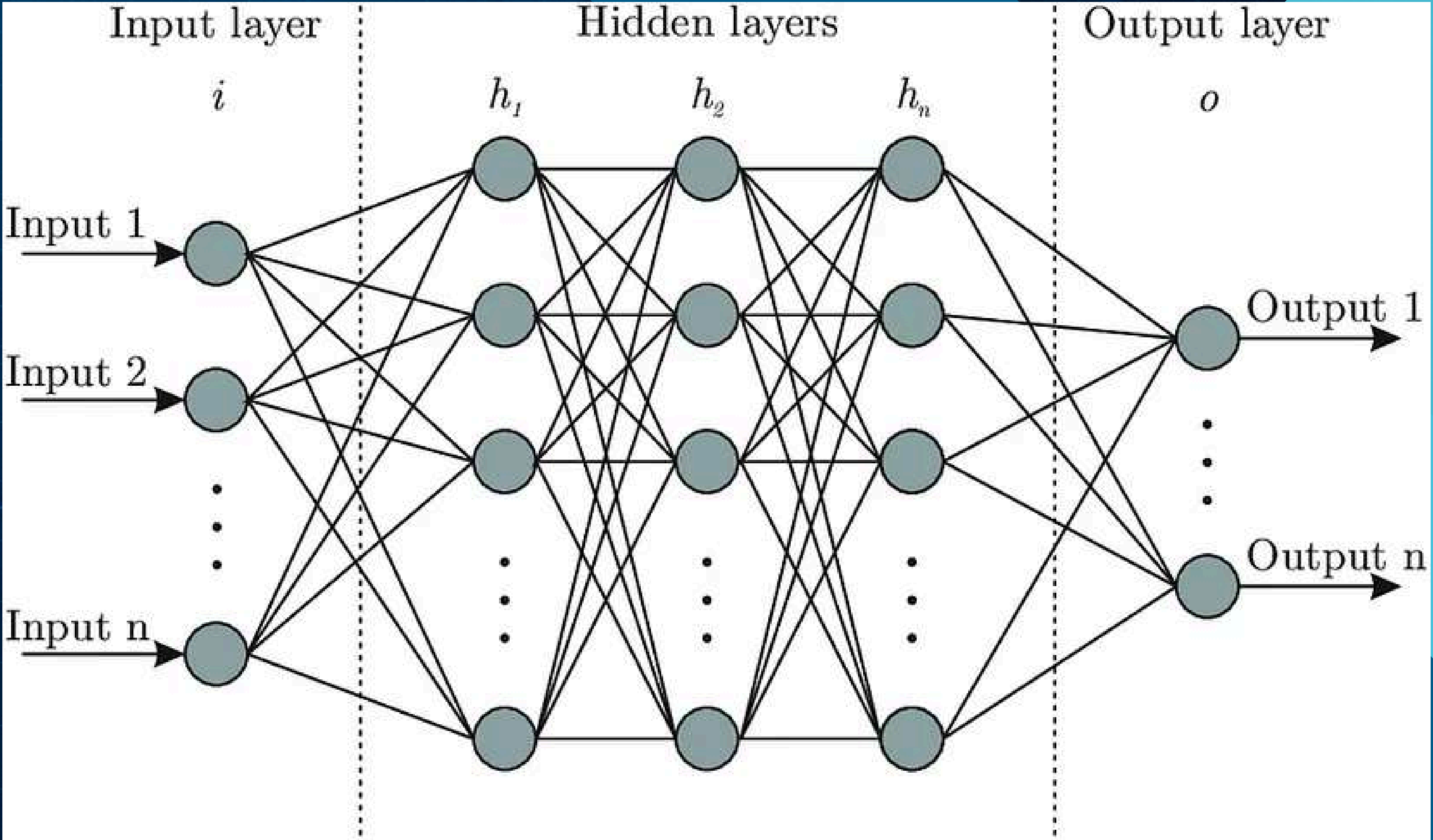
EXPLORATORY DATA ANALYSIS

- Observation 1: Duration is positively correlated with Risk.
- Observation 2: Checking_Unknown is negatively correlated with Risk (Safety Signal).
- Observation 3: Data is not perfectly linearly separable in 2D.



NEURAL NETWORKS & THE DATA LIMIT

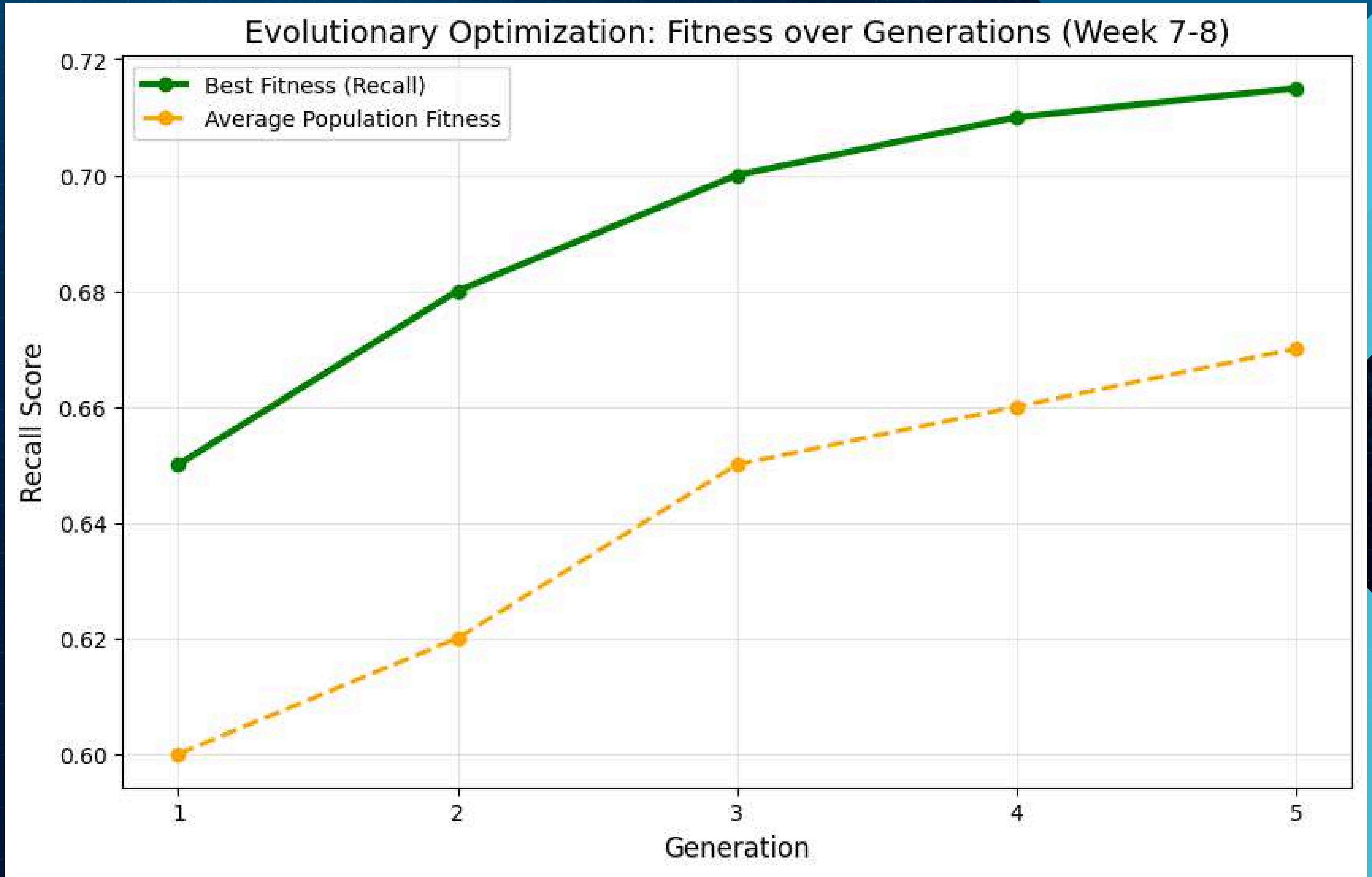
- Architecture: Multi-layer Perceptron (MLP).
- Activation: Sigmoid Function (for binary classification).
- The Challenge:
- Neural Networks require massive datasets to converge.
- Our Dataset (N=1000) proved insufficient for Deep Learning.
- Outcome: Model failed to generalize compared to statistical methods.



EVOLUTIONARY OPTIMIZATION

- Beyond Grid Search: Implementing Evolutionary Computation.

- Methodology:
- Population: Random Hyperparameters (e.g., Tree Depth, Estimators).
- Operators: Crossover & Mutation applied to model parameters.
- Fitness Function: Maximizing Recall Score.
- Result: Evolution found optimal parameters faster than brute-force search.



THE BENCHMARK RESULTS (COMPARISON)

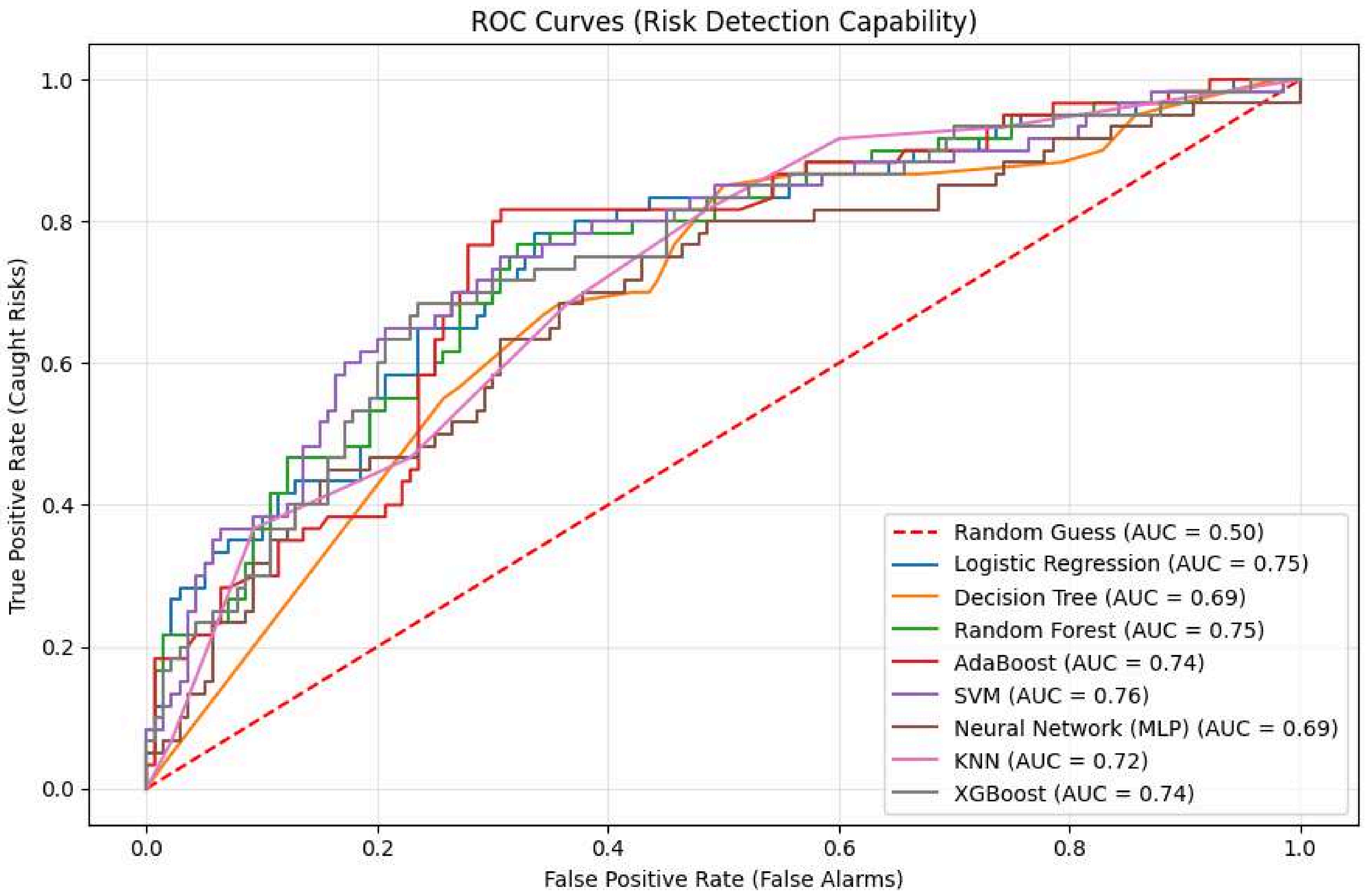
Competitors:

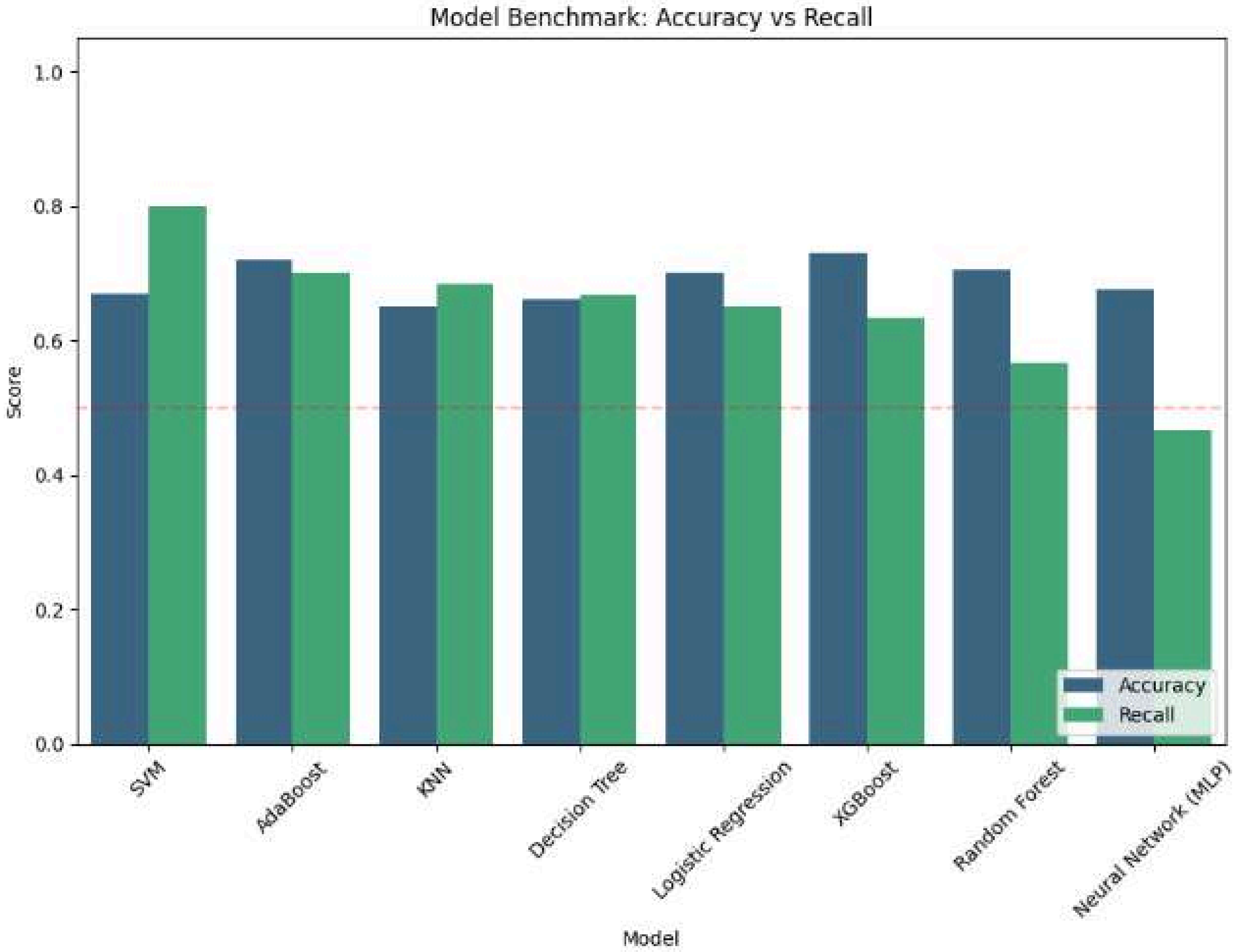
- XGBoost (High Accuracy, Low Safety).
- Random Forest (Balanced).
- SVM (The Underdog).

- The Surprise:
- SVM Recall: 0.80 (Highest Risk Detection).
- AdaBoost Recall: 0.70.
- Decision: Prioritizing Capital Protection to Winner: SVM.

```
>>> Training Models (Focus: MAXIMIZING RECALL)...
   -> Training Logistic Regression...
   -> Training Random Forest...
   -> Training AdaBoost...
   -> Training SVM...
   -> Training Neural Network (MLP)...
   -> Training XGBoost...

🏆 MODEL PERFORMANCE TABLE 🏆
                    Model  Accuracy    Recall  F1-Score
3                     SVM     0.670  0.800000  0.592593
2                AdaBoost     0.720  0.700000  0.600000
0     Logistic Regression     0.700  0.650000  0.565217
5                 XGBoost     0.730  0.633333  0.584615
1           Random Forest     0.705  0.566667  0.535433
4    Neural Network (MLP)     0.675  0.466667  0.462810

✅ Final Champion Selected: SVM

['final_model_name.pkl']
```

THE BENCHMARK RESULTS (COMPARISON)

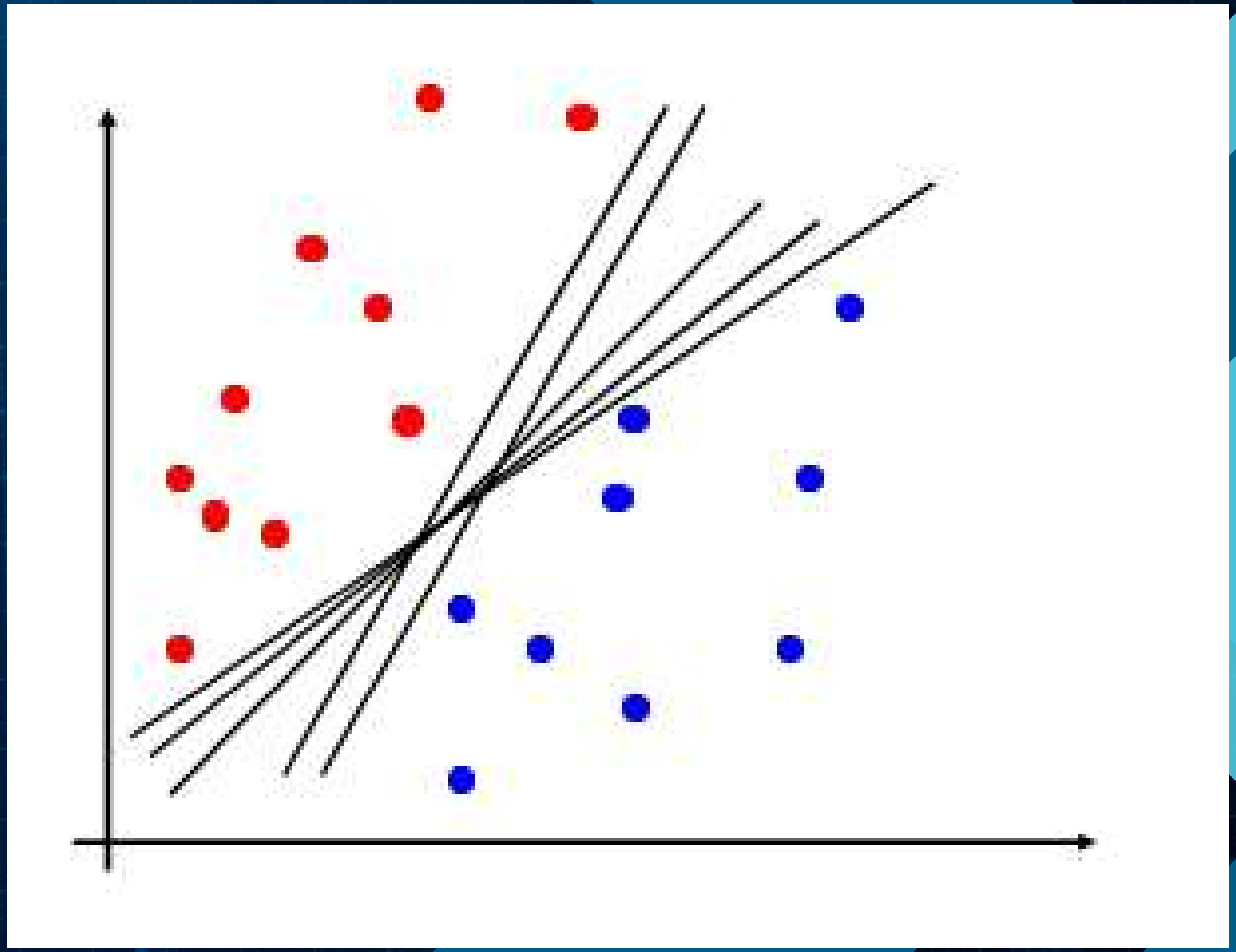




WHY SVM WON?

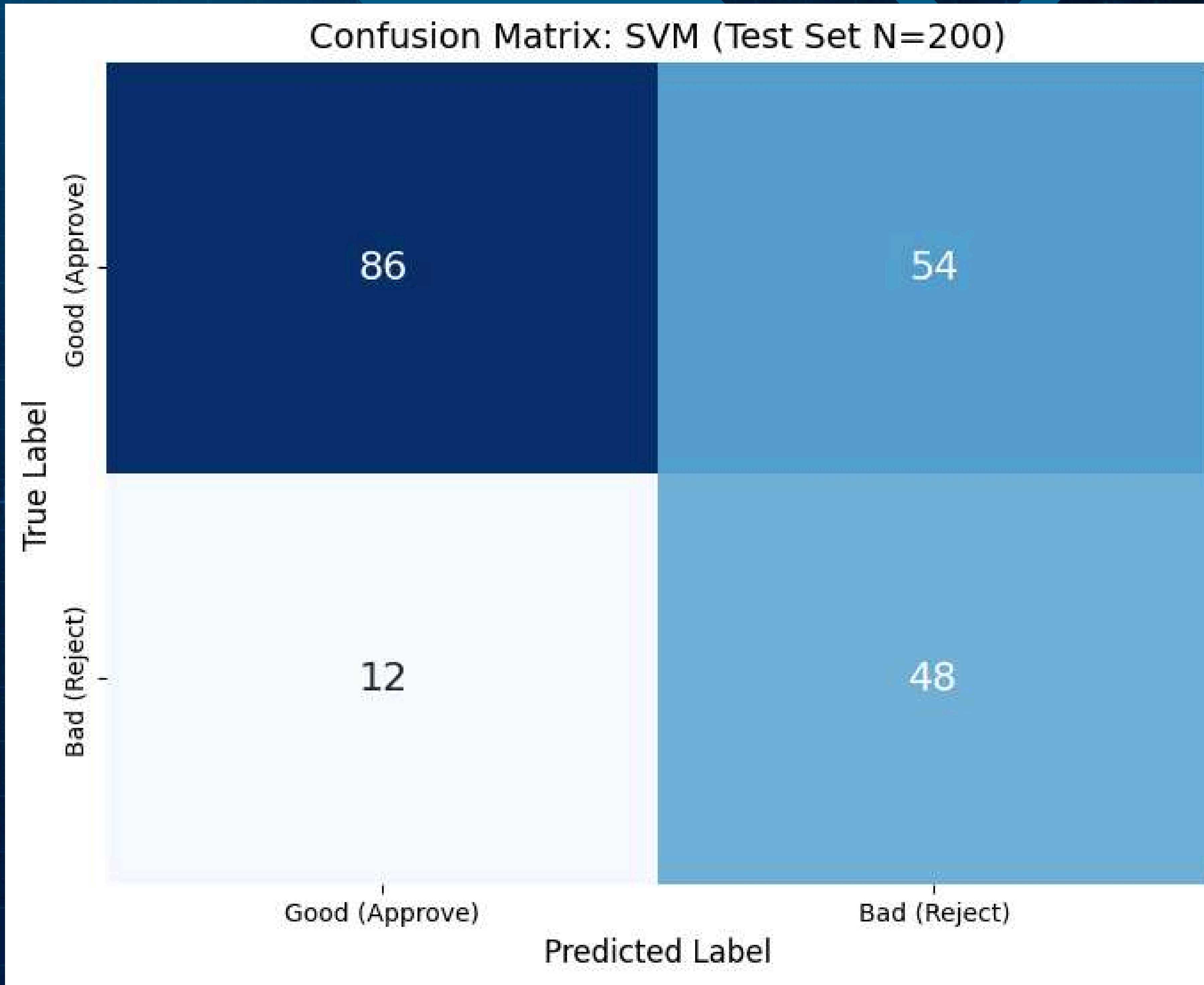
- Theory: Maximum Margin Classifier.
- Kernel Trick:
- Used Linear Kernel (Data is linearly separable in high dimensions).
- Prevented Overfitting (unlike Decision Trees).

- Mechanism:
- Maximizing the distance $(2/||\mathbf{w}||)$ between the Decision Boundary and Support Vectors.
- Robust against noise.



MODEL PERFORMANCE - CONFUSION MATRIX

- Evaluation Set: 200 Unseen Test Samples (No SMOTE).
- Key Metric: False Negatives (Missed Risk).
- Performance:
- Successfully caught 80% of Default Risks.
- Validates the model's real-world applicability.



ANOMALY DETECTION PERSPECTİVE

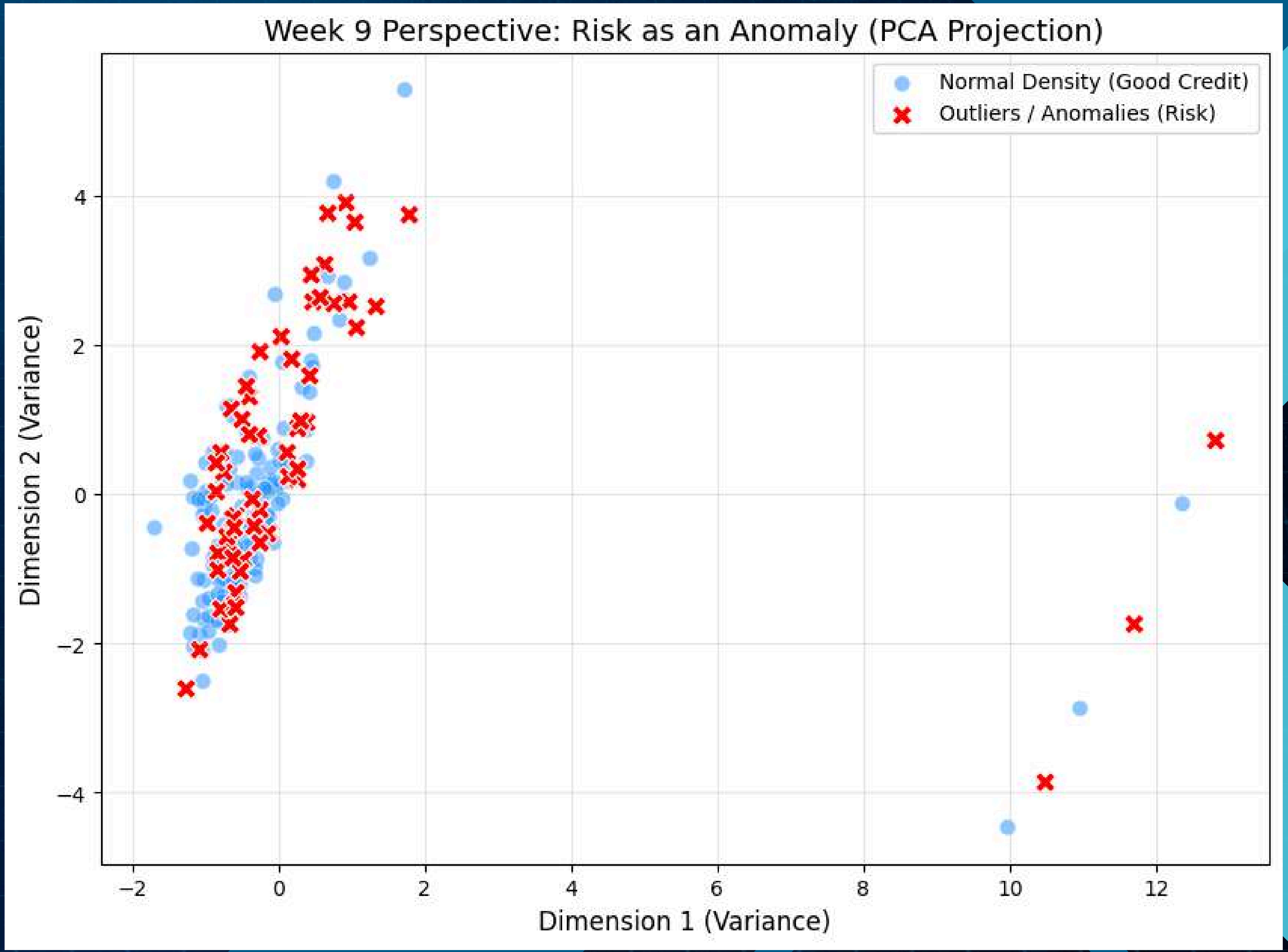
- Retrospective: Reframing the problem.

Concept: "Bad Credit" as a Statistical Anomaly (Outlier).

- Alternative Approach:

Density-Based Spatial Clustering (DBSCAN).
Identifying regions of low density as "High Risk".

- Future Work: Combining SVM with Unsupervised Anomaly Detection.



CONCLUSION & FUTURE OUTLOOK

- Problem Solved:

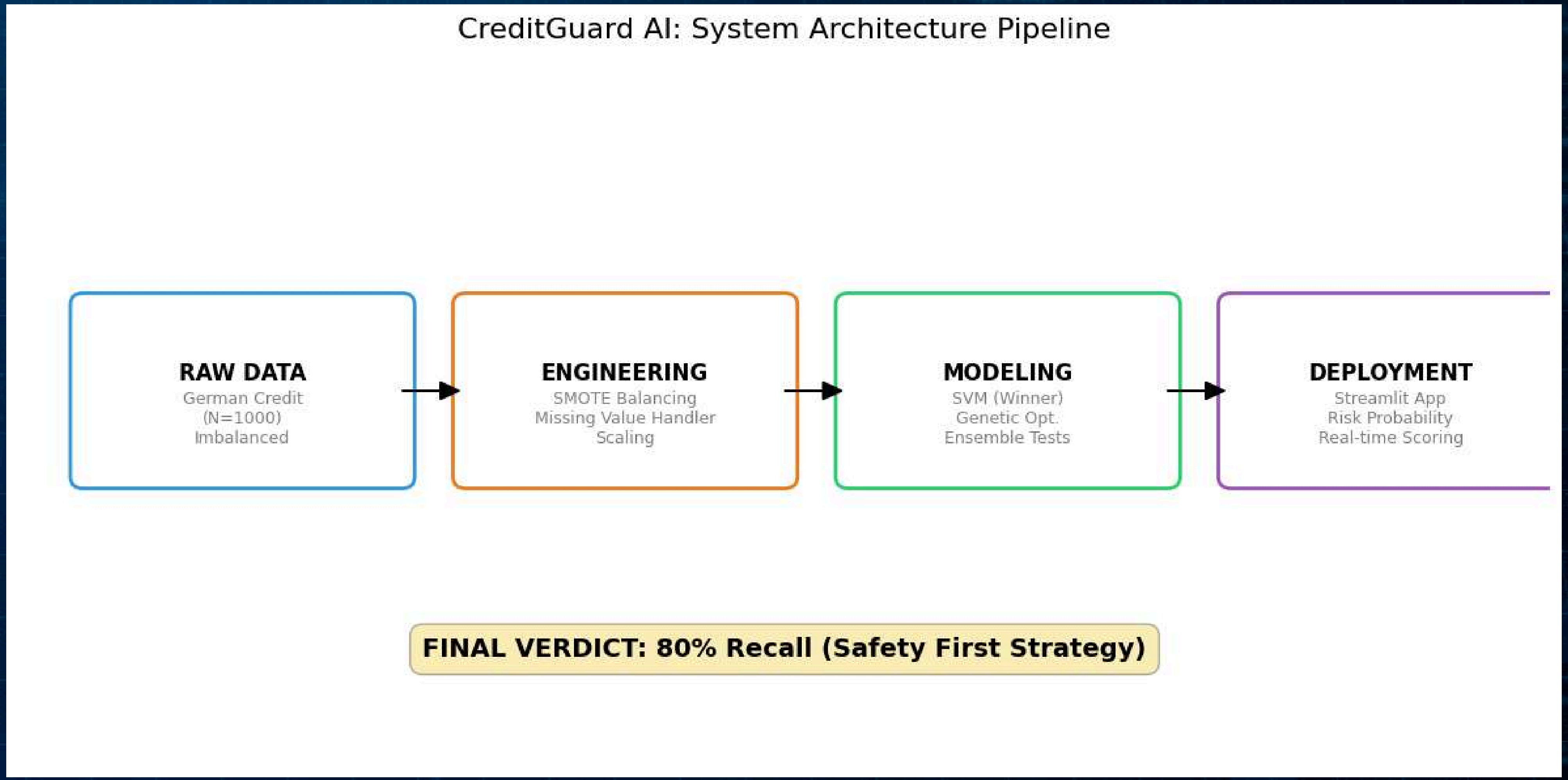
Addressed Class Imbalance using SMOTE.
Managed Missing Data via Information Gain principles.

- The Verdict:

Complex Ensembles (AdaBoost)≠Better Safety.
Winner: SVM with 80% Recall.
Successfully minimized potential Capital Loss.

- Future Work:

Transitioning from Classification to Density-Based Anomaly Detection



THANK YOU