Carnegie Mellon University

18789 Project Presentation

Interpretable Deep Generative Models for Default Prediction

Li Cao (licao), Jiachun Xu (jiachunx), Likeer Xu (xlikeer)

Apr 23, 2025

Table of Contents

- ❑ Introduction / Motivation
- ❑ Related Work
- ❑ Methods
- ❑ Experimental results
- ❑ Future Plan
- ❑ References

Introduction

- **Interpretable Deep Generative Models for Default Prediction**
- **Why Default Prediction?**
  - Economic Impact
  - High data imbalance
  - Accuracy vs Transparency
    - Interpretable generative models to provide justification in decision making

Dataset - UCI Default of Credit Card Clients

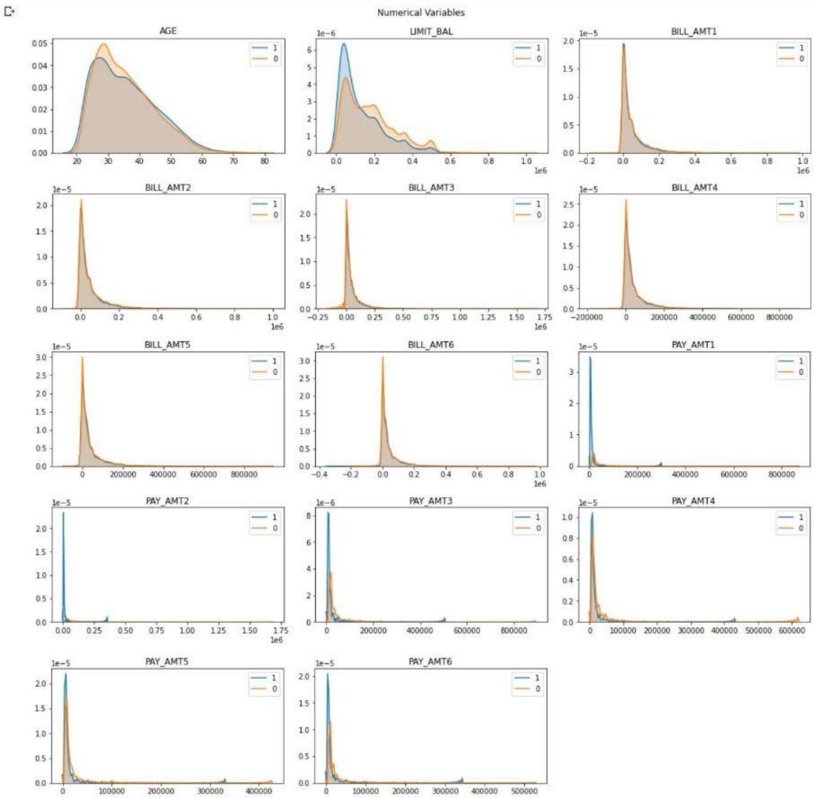
Characteristics:

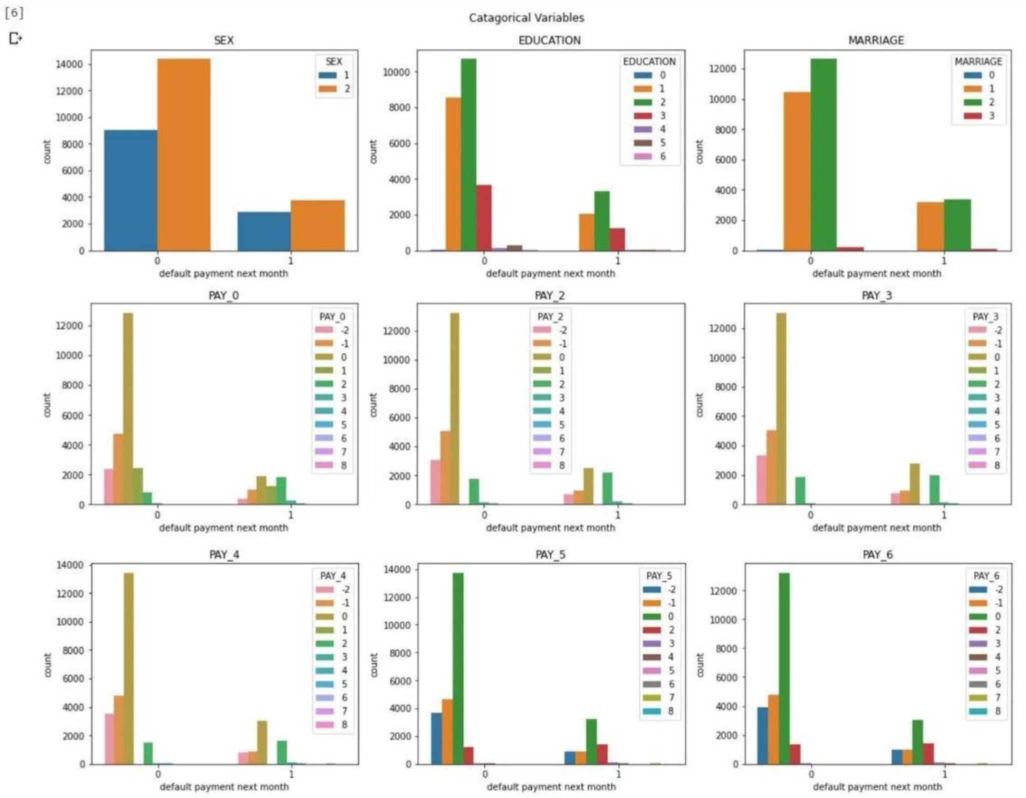
- 30k rows, 22.1% default ratio
- 9 Categorical features: Gender, Education, Marriage, Repayment Status
- 14 Numerical features: Monthly bill & payment amount in the past 6 months, LIMIT_BAL, AGE
- **Target:** whether default next month

Variable	Column Name	Description	Value / Unit Explanation
X1	LIMIT_BAL	Amount of given credit	NT dollars
X2	SEX	Gender	1 = Male; 2 = Female
X3	EDUCATION	Education level	1 = Graduate; 2 = University; 3 = High school; 4 = Others
X4	MARRIAGE	Marital status	1 = Married; 2 = Single; 3 = Others
X5	AGE	Age	Years
X6–X11	PAY_0–PAY_6	Repayment status	0 = On-time; 1–9 = Months delayed
X12–X17	BILL_AMT1–BILL_AMT6	Monthly bill statements	NT dollars
X18–X23	PAY_AMT1–PAY_AMT6	Previous payments	NT dollars

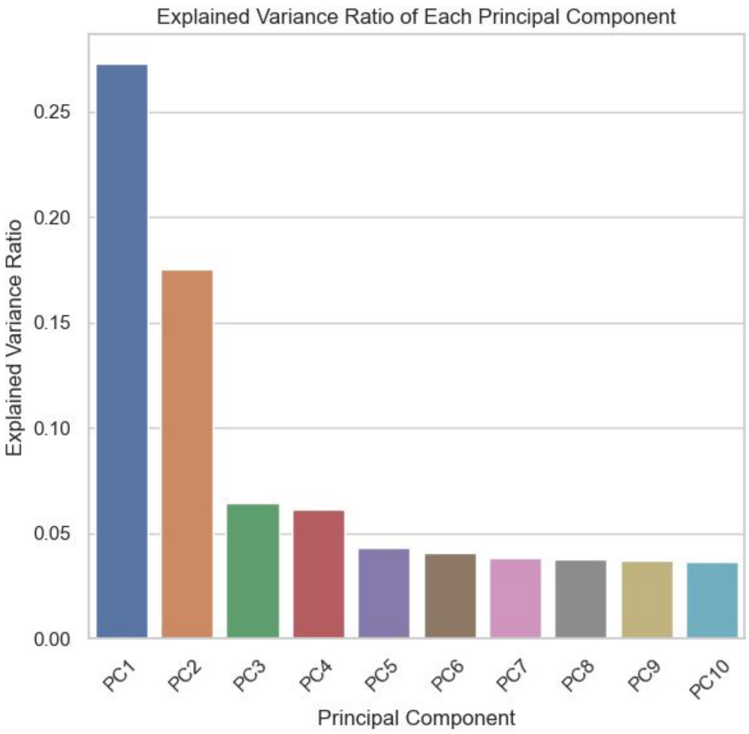
Carnegie Mellon University

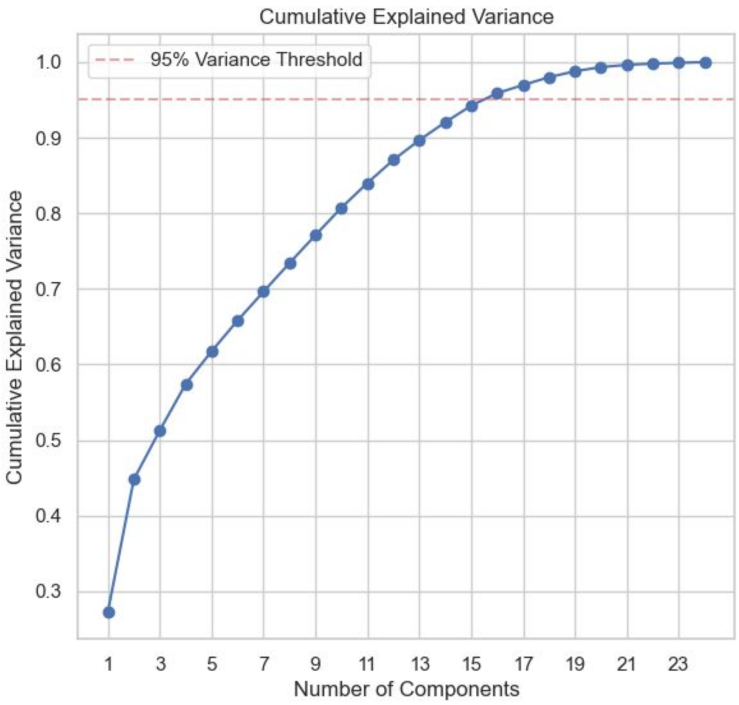
Dataset - EDA - Distribution Plots





Dataset - EDA - PCA Analysis





Carnegie Mellon University

Related Work - ML Methods on UCI Credit Card Dataset

Summary of Data Mining Techniques in Credit Scoring

Author(s)	Method	Highlight
Rosenberg & Gleit (1994)	DA, Trees, Markov Chains	Static/dynamic models for credit decisions
Hand & Henley (1997)	Statistical classification	Defined "credit scoring" and its significance
Paolo (2001)	Bayesian + MCMC	Flexible modeling of complex data
Lee et al. (2002)	NN + Discriminant	Hybrid model with better speed and accuracy
Baesens et al. (2003)	SVM, NN, LR, LDA	Both complex and simple models perform well

Method	ROC-AUC
K-nearest neighbor	0.45
Logistic regression	0.44
Discriminant analysis	0.43
Classification trees	0.536

- Limited ability to distinguish default cases
- ROC-AUC remains low

How about Deep Learning methods?

Related Work - DL Methods

Method	Backbone / Key Idea
DeepFM	MLP + Factorization Machine
NODE	Differentiable Decision Trees
NAM	MLP (Feature-wise Subnetworks)
TabNet	MLP + Attention-based Feature Masking
xDeepFM	CIN + FM (Field Interaction)
Boost-GNN	GNN on GBDT Trees
DNN2LR	MLP + Logistic Regression

Most DL methods use backbones like **MLP, GNN or ensemble with trees**, and focus on **Classification**
Drawbacks: Typically lacks **data generation ability** and struggles with **class imbalance**

Our work: **deep generative models** for tabular data
→ **Handle imbalance, synthesize data, ensure interpretability + classification**

Carnegie Mellon University

Methodology - Workflow

- Feature Engineering
- Synthetic Data Generation
- Model Design & Implementation
    - DGM: VAE, GAN, Diffusion Model, AR model (Transformer)
- Interpretability Design

Methodology - Feature Engineering

1. Payment-to-Bill Ratios (6 Features)

- **Formula:** For each month $i = 1, 2, \ldots, 6$:

$$\text{PAY\_TO\_BILL}_i = \frac{\text{PAY\_AMT}_i}{\text{BILL\_AMT}_i + \epsilon} \quad (\epsilon = 10^{-10})$$

- **Purpose:** Measures how much of the bill was actually paid each month.

2. Average Bill/Payment Amounts (2 Features)

- **Formulas:**

$$\text{AVG\_BILL\_AMT} = \frac{1}{6}\sum_{i=1}^{6} \text{BILL\_AMT}_i$$

$$\text{AVG\_PAY\_AMT} = \frac{1}{6}\sum_{i=1}^{6} \text{PAY\_AMT}_i$$

- **Purpose:** Captures average historical bill and payment amounts.

3. Payment Delay Features (2 Features)

- **Formulas:**

$$\text{PAY\_DELAY\_SUM} = \sum_{\text{col} \in \{\text{PAY\_0},\ \text{PAY\_2},\ \ldots,\ \text{PAY\_6}\}} \text{col}$$

$$\text{PAY\_DELAY\_TREND} = \text{PAY\_0} - \text{PAY\_6}$$

- **Purpose:**
  - PAY_DELAY_SUM: Total payment delays across 6 months
  - PAY_DELAY_TREND: Trend in delays (recent vs. older behavior)

4. Utilization Rates (6 Features)

- **Formula:** For each month $i = 1, 2, \ldots, 6$:

$$\text{UTILIZATION}_i = \frac{\text{BILL\_AMT}_i}{\text{LIMIT\_BAL} + \epsilon} \quad (\epsilon = 10^{-10})$$

- **Purpose:** Ratio of billed amount to total credit limit.

5. Average Utilization (1 Feature)

- **Formula:**

$$\text{AVG\_UTILIZATION} = \frac{1}{6}\sum_{i=1}^{6} \text{UTILIZATION}_i$$

- **Purpose:** Average credit utilization over 6 months.

17 Total New Features

$$6\,(\text{PAY\_TO\_BILL}) + 2\,(\text{AVG}) + 2\,(\text{PAY\_DELAY}) + 6\,(\text{UTILIZATION}) + 1\,(\text{AVG\_UTILIZATION})$$

Carnegie Mellon University

Methodology - Synthetic Data Generation

- Class Imbalance Mitigation
    - Real-world credit datasets are often imbalanced, with far fewer default cases.
- Data Augmentation
    - Effectively expand the dataset size, allowing models to generalize better and reduce overfitting
- Approach
    - TVAE, CTGAN and Diffusion models

Methodology - TVAE

- **Categorical Features:**

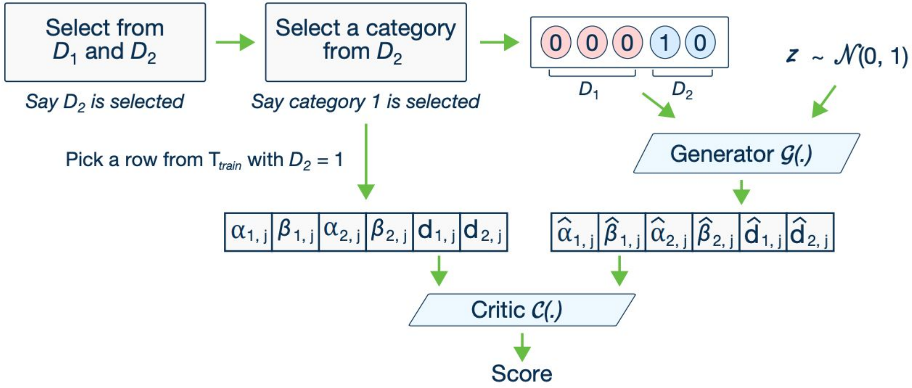
  One-hot encoded → embedded
- **Continuous Features:**

  Scaled to [0,1] → concatenated with categorical embeddings.
- **Standard VAE Encoder and Decoder:**
  - Inputs encoded into a latent Gaussian distribution (μ, σ) → sampled via reparameterization trick.
  - Latent vectors decoded into synthetic tabular records using a decoder network.

Carnegie Mellon University

Methodology - CTGAN

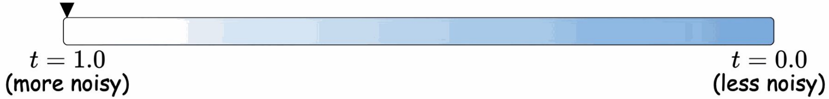
- **Generator Input:**
  Random noise vector + conditional vector
- **Generator Output:**
  Mixed-type synthetic features
- **Discriminator Input:**
  Real and synthetic data
- **Training Objective:**
  Trains via adversarial loss with conditional vector supervision to ensure mode coverage and fidelity.



Methodology - Diffusion models

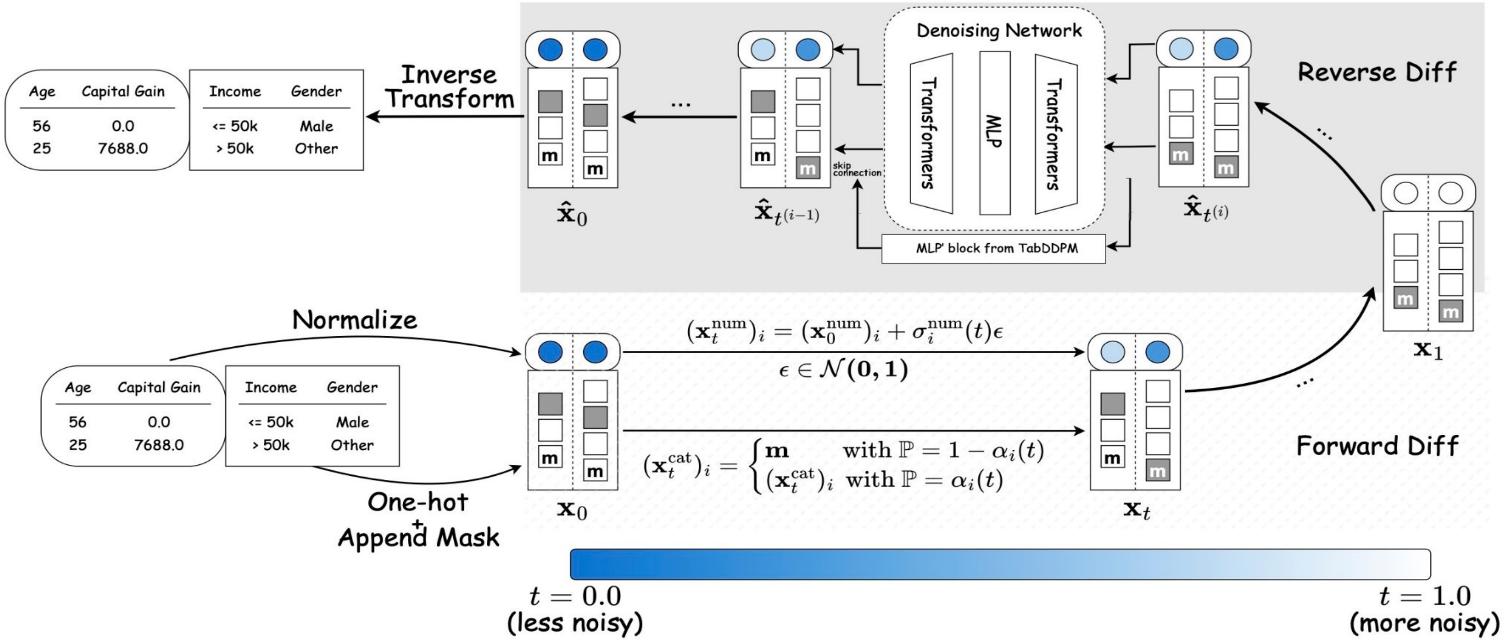
- **Preprocessing:**
  Numerical columns are normalized
  Categorical columns are one-hot encoded with a special [MASK] token
- **Forward Diffusion:**
  Noise is added separately to each column type using learnable feature-wise schedules
  Gaussian noise for numerical features, masking for categorical ones
- **Denoising:**
  A Transformer and MLP based network jointly learns to denoise all features by reversing the diffusion steps

Budget (M$)	Duration (min)	IMBD Rating	Language	Genre	Award
520.2	4951	9.0	[MASK]	[MASK]	[MASK]
542.2	2681	14.1	[MASK]	[MASK]	[MASK]
-904.0	-2412	-9.3	[MASK]	[MASK]	[MASK]



Carnegie Mellon University

Methodology - Diffusion models



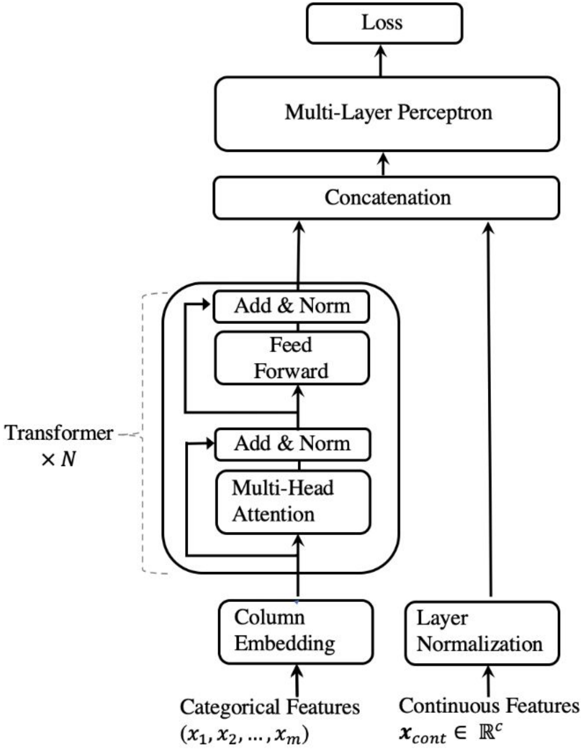
- TabDiff as the baseline
- Added a MLP block from TabDDPM as a skip connection for the denoising network

Methodology - TabTransformer

- Uses **self-attention** to capture contextual relationships between categorical features.

Architecture

1. **Categorical Features**:
   - Embedded into vectors → processed by **Transformer layers** (self-attention captures feature interactions).
2. **Continuous Features**:
   - Normalized → concatenated with transformed categorical embeddings.
3. **Prediction**:
   - Combined features fed into an **MLP** for final output.



Carnegie Mellon University

Methodology - FT-Transformer

- Unifies feature processing by converting *both categorical and numerical features* into embeddings and applying **global self-attention**

Architecture

1. **Feature Tokenizer:**
    - **Categorical**: Embedded into vectors.
    - **Continuous**: Linearly projected into embeddings (like NLP tokens).
2. **Transformer Layers**:
    - Processes all tokens with **multi-head self-attention** to model interactions.
    - Adds a *[CLS] token* to aggregate global information.
3. **Prediction**:
    - [CLS] token output → MLP for final prediction

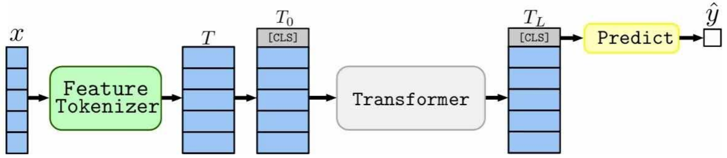


Figure 1: The FT-Transformer architecture. Firstly, Feature Tokenizer transforms features to embeddings. The embeddings are then processed by the Transformer module and the final representation of the [CLS] token is used for prediction.

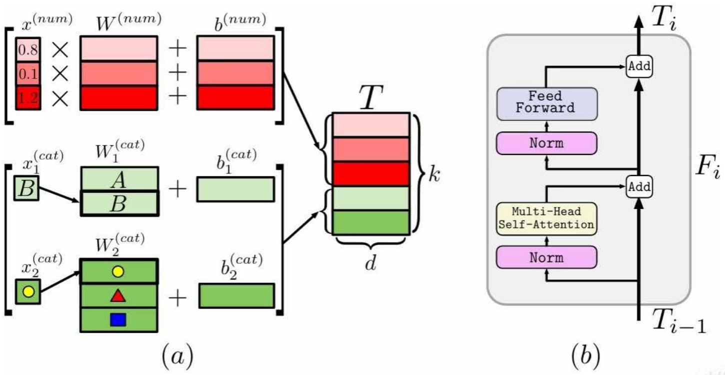


Figure 2: (a) Feature Tokenizer; in the example, there are three numerical and two categorical features; (b) One Transformer layer.

Mellon University

Methodology - Introducing TabFT-Transformer

- **TabFT-Transformer** - Combines key ideas from **TabTransformer** (categorical feature attention) and **FT-Transformer** (unified token processing) into a hybrid architecture

Key Innovations

1. **Feature Embedding Strategy**
   - **Categorical Features**:
     - Uses nn.Embedding layers (like TabTransformer) for categorical features.
   - **Numerical Features**:
     - Projects numerical features into embeddings via nn.Linear layers (like FT-Transformer), treating them as tokens for unified processing.
2. **CLS Token Integration** (from FT-Transformer)
   - Adds a learnable [CLS] token to aggregate global feature interactions.
3. **Transformer Processing**
   - All embedded tokens (categorical + numerical + CLS) pass through **multi-head self-attention** layers, enabling cross-feature interaction modeling for both data types.
4. **Output Head**
   - Uses the [CLS] token to feed into an MLP for prediction

Methodology - Introducing TabFT-Transformer

- **TabFT-Transformer** - Combines key ideas from **TabTransformer** (categorical feature attention) and **FT-Transformer** (unified token processing) into a hybrid architecture

Key Benefits

1. **Comprehensive Interactions**: Captures **numerical-categorical** dependencies (unlike TabTransformer).
2. **Stability**: LayerNorm on numerical features prevents dominance in attention.
3. **Efficiency**: CLS token aggregates global patterns better than concatenation.
4. **Flexibility**: Inherits categorical semantics (TabTransformer) + unified attention (FT-Transformer).

Methodology - Introducing TabFT-Transformer

- **TabFT-Transformer** - Combines key ideas from **TabTransformer** (categorical feature attention) and **FT-Transformer** (unified token processing) into a hybrid architecture

Model	**TabTransformer**	**FT-Transformer**	**TabFT-Transformer**
Categorical Features	Embeddings	Tokenized	Embeddings
Numerical Features	MLP/raw	Tokenized	LayerNorm + Tokenized
Attention Scope	Categorical-only	Global	Global
Global Aggregation	Concatenation	Pooling/CLS token	CLS token

Carnegie Mellon University

Methodology - Interpretability Design

- **Attention-based Feature Importance**
  - Uses the model’s attention weights (from the CLS token) to quantify how much the model "focuses" on each feature.
- **Perturbation-based Feature Importance**
  - Measures the impact of perturbing each feature on the model’s predictions.
  - For each feature, replace its value with the **mean** of that feature across the batch.
  - Measure the absolute difference between baseline and perturbed predictions.
- **SHAP (SHapley Additive exPlanation) Analysis**
  - Uses fair allocation results from cooperative game theory to allocate credit for a model’s output among the input features

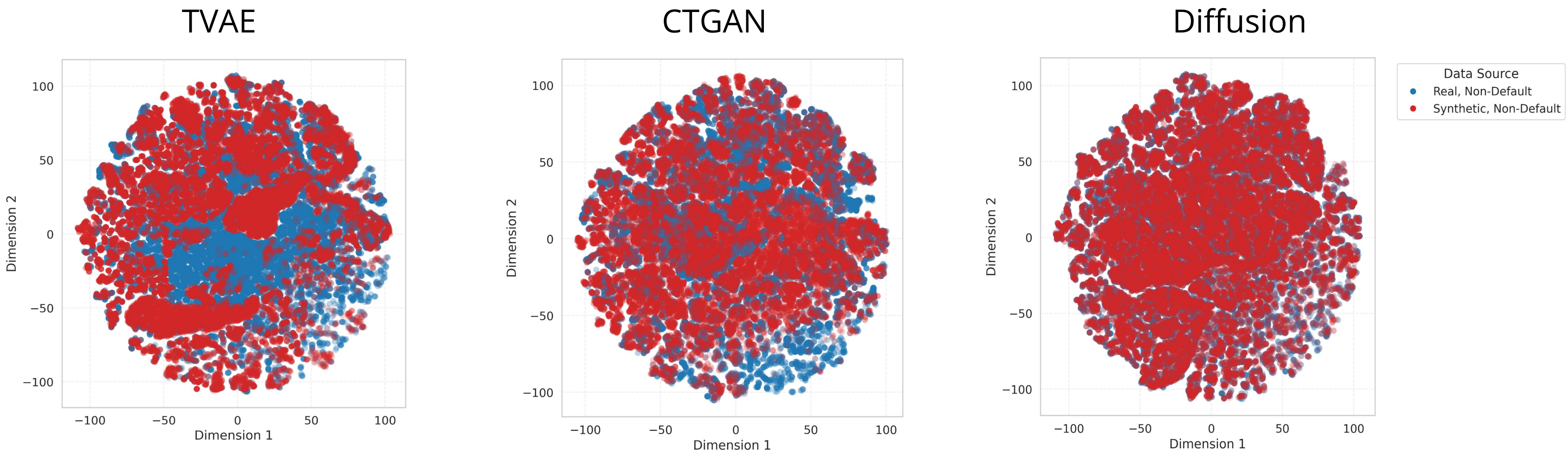
Experimental Results - Synthetic Data

- **Evaluation Metrics - Column Shapes & Column Pair Trends**
  - **Column Shapes** evaluates the univariate distribution similarity of each column between real and synthetic data.
  - **Column Pair Trends** assesses whether the pairwise relationships between columns are preserved.

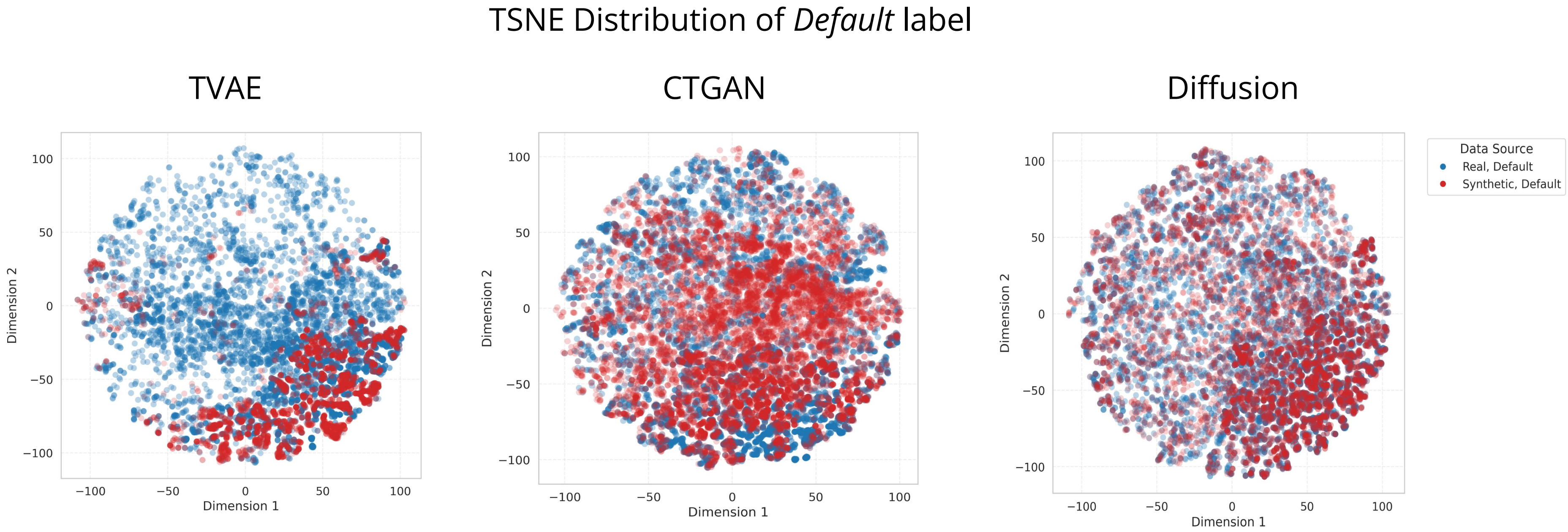
Method	Column Shapes(%)	Column Pair Trends(%)	Overall Score(%)
SMOTE	90.82	92.5	91.66
TVAE	90.40	84.73	87.56
CTGAN	89.12	85.27	87.19
Diffusion	98.58	98.36	98.47

Experimental Results - Synthetic Data

TSNE Distribution of *Non-Default* label

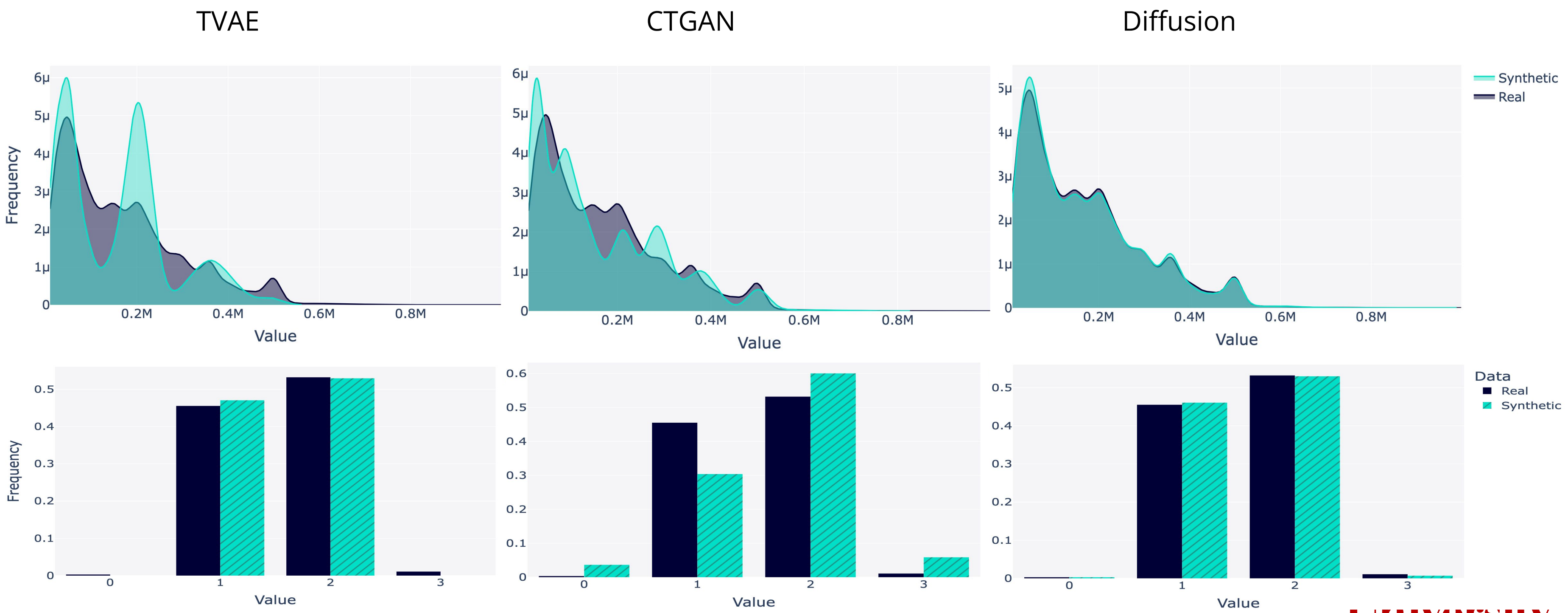


Experimental Results - Synthetic Data



Experimental Results - Synthetic Data

Distribution of *LIMIT_BAL* and *Marriage* feature



Experimental Results - Classification

- Baseline: Logistic regression & XGBoost
- TabFT-Transformer slightly outperforms both Tab-Transformer & FT-Transformer
- TabFT-Transformer matches XGBoost in terms ROC-AUC, slightly lower F1-Score due to lower Precision despite higher Recall

Model	ROC-AUC	F1	Precision	Recall
Logistic Regression	0.716	0.486	0.447	0.632
XGBoost	0.778	0.533	0.475	0.609
AE+MLP	0.743	0.459	0.585	0.373
TabTransformer	0.773	0.522	0.475	0.586
FT-Transformer	0.775	0.521	0.448	0.620
TabFT-Transformer	0.778	0.524	0.454	0.620

Experimental Results - Classification with Synthetic Data

- Synthetic data generated by diffusion model, combined with only training dataset to avoid leakage
- Synthetic data **30k, 150k, 300k** merely increase training data size
  - Slight improvement as synthetic data size increases
- Synthetic data **Default-only 10k** makes training dataset class-label balanced
  - Only increased Precision while the other metrics decreased

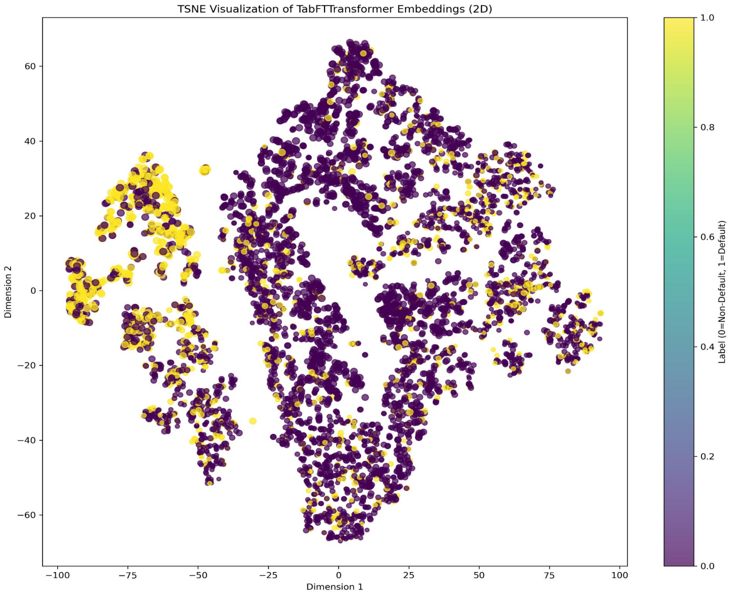
Dataset	ROC·AUC	F1	Precision	Recall
Original-only	0.778	0.524	0.454	0.620
Original + Synthetic 30k	0.779	0.526	0.465	0.606
Original + Synthetic 150k	0.780	0.528	0.444	0.650
Original + Synthetic 300k	0.781	0.531	0.467	0.614
Original + Synthetic Default-only 10k	0.766	0.528	0.504	0.555

Carnegie Mellon University

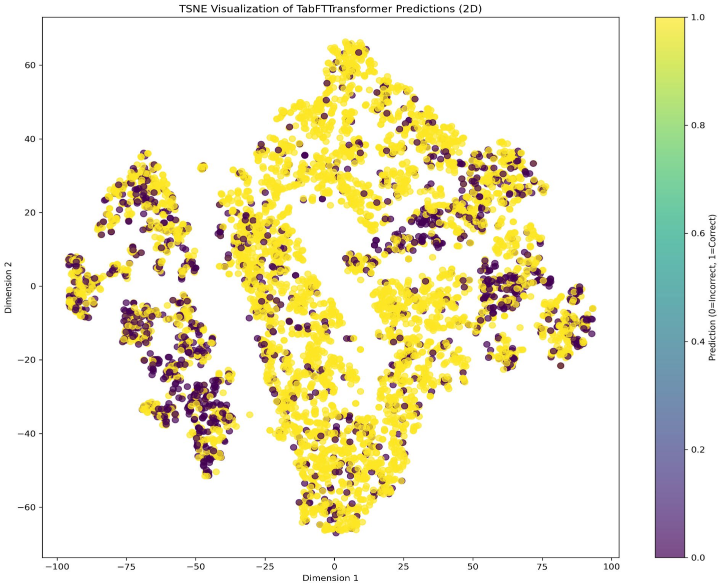
Experimental Results - Interpretability

TSNE Visualization of TabFT-Transformer Embedding (2D)

Default vs Non-Default

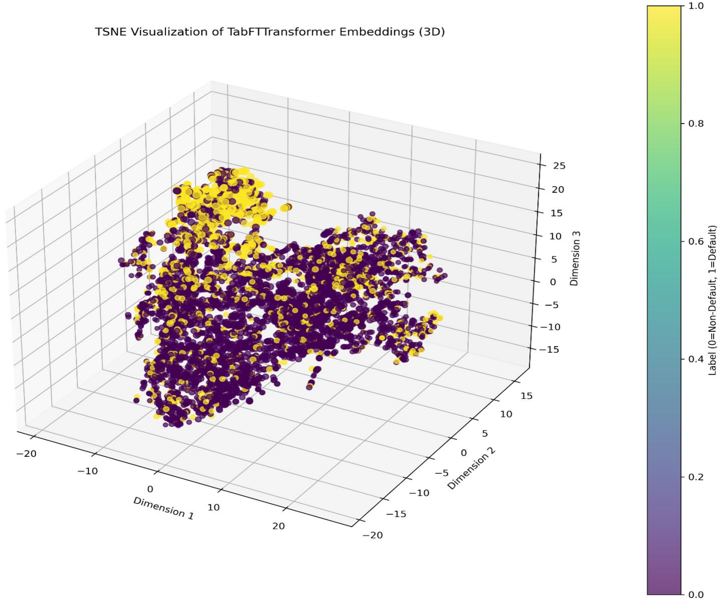


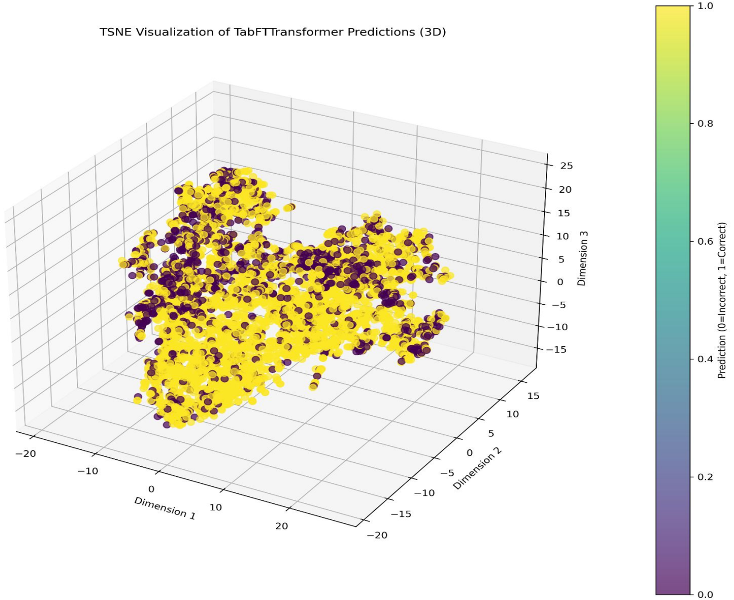
Correct vs Incorrect Classification



Experimental Results - Interpretability

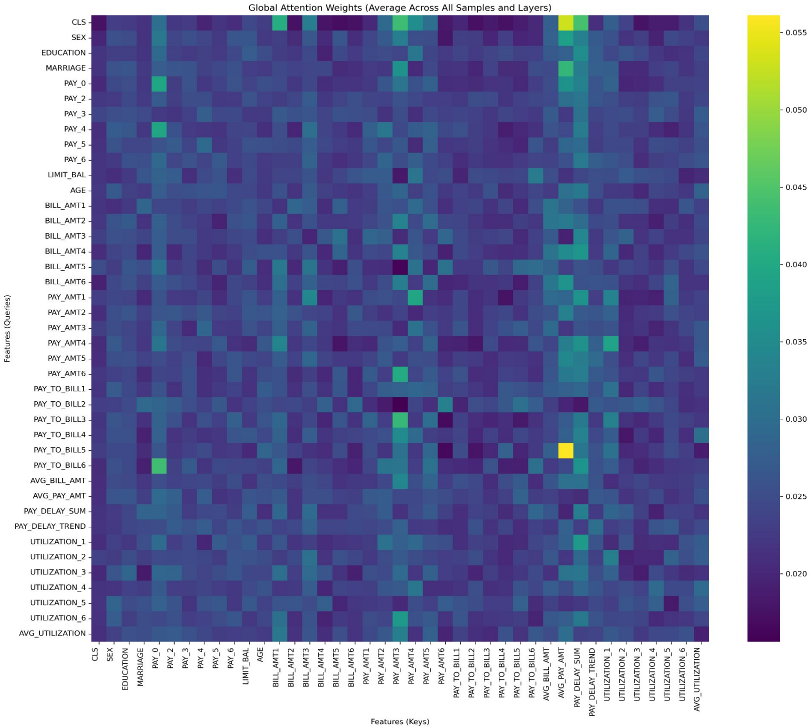
TSNE Visualization of TabFT-Transformer Embedding (3D)





Experimental Results - Interpretability

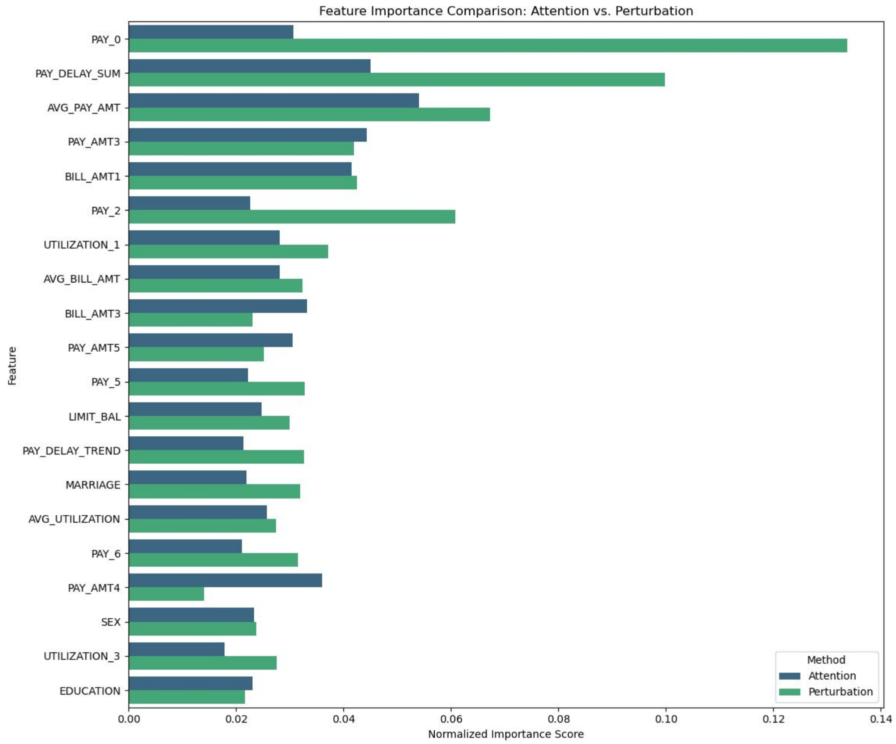
TabFT-Transformer Attention Weights Visualization



Carnegie Mellon University

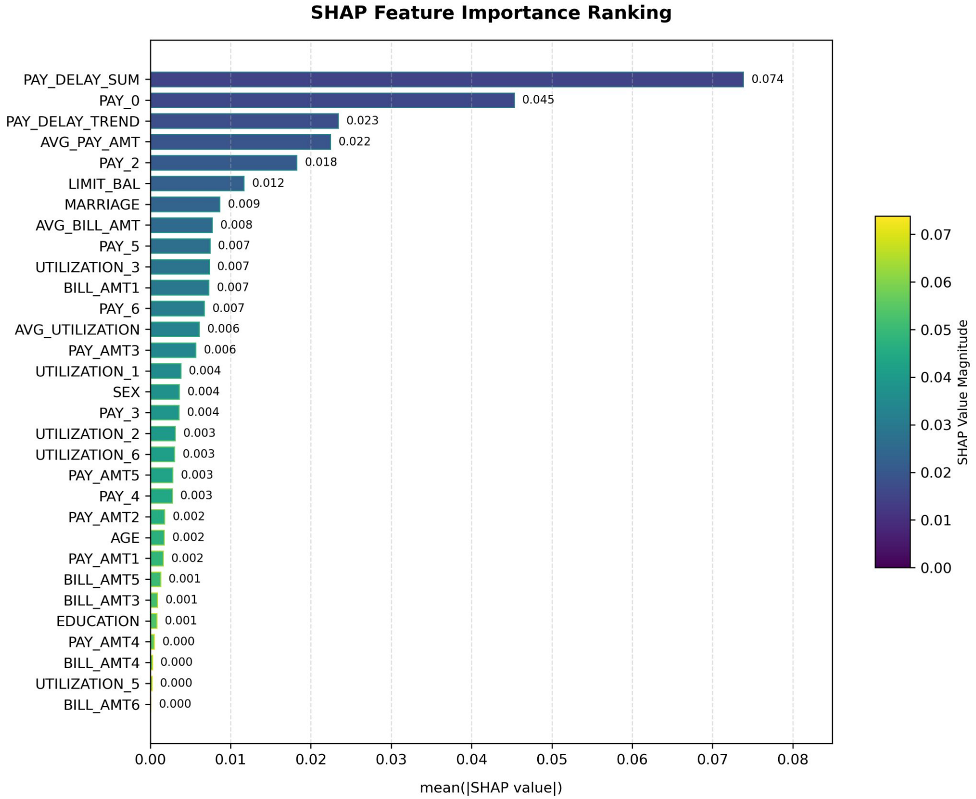
Experimental Results - Interpretability

Feature Importance (Attention vs Perturbation)



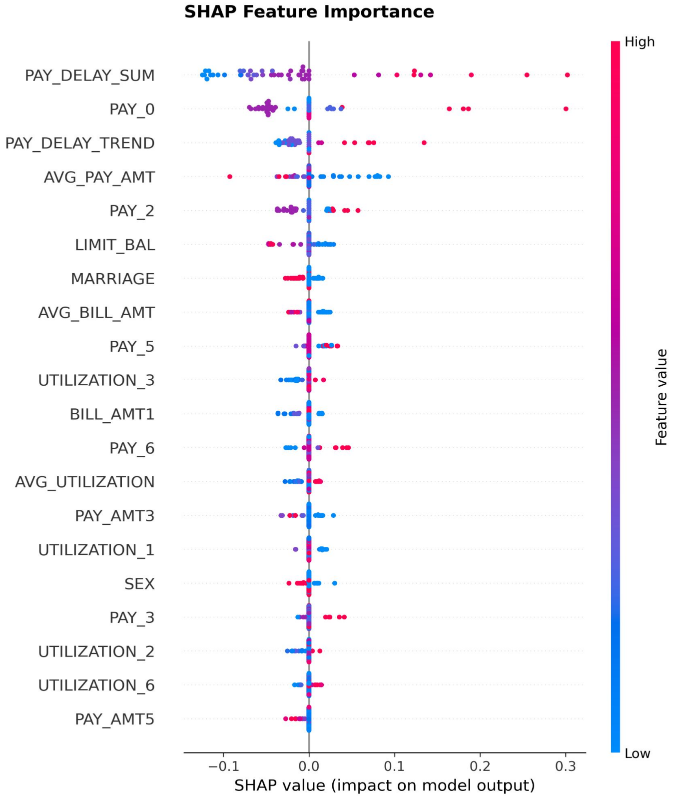
Carnegie Mellon University

Experimental Results - Interpretability



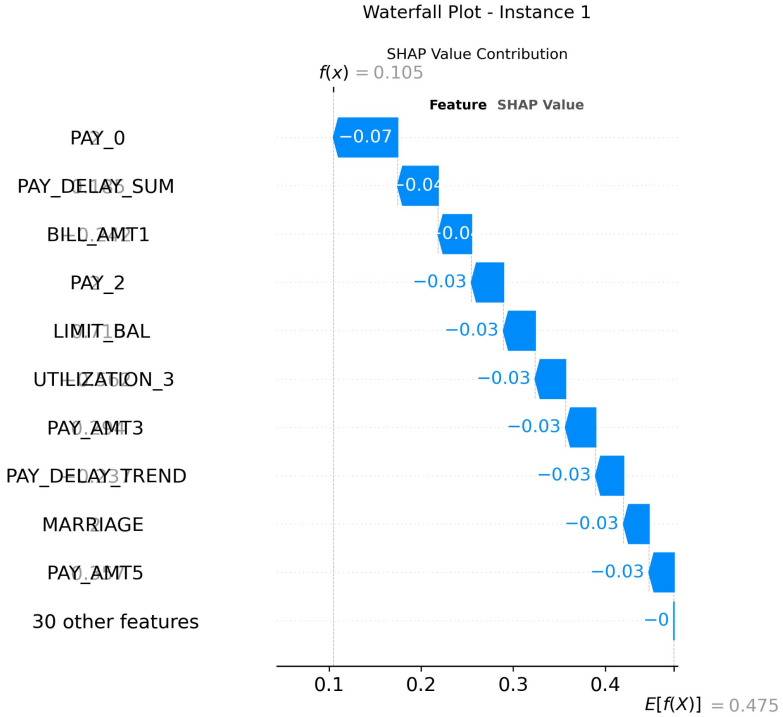
Carnegie Mellon University

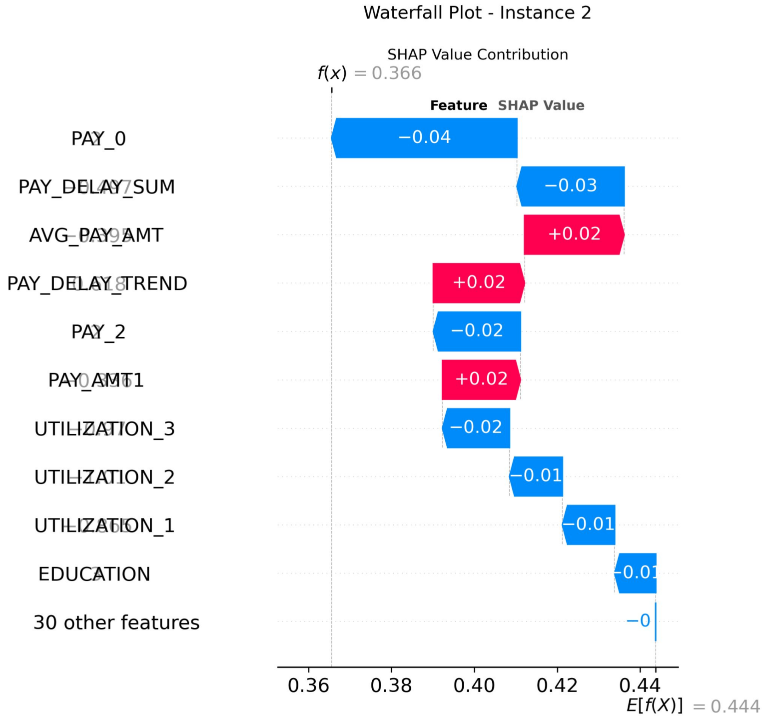
Experimental Results - Interpretability



Experimental Results - Interpretability

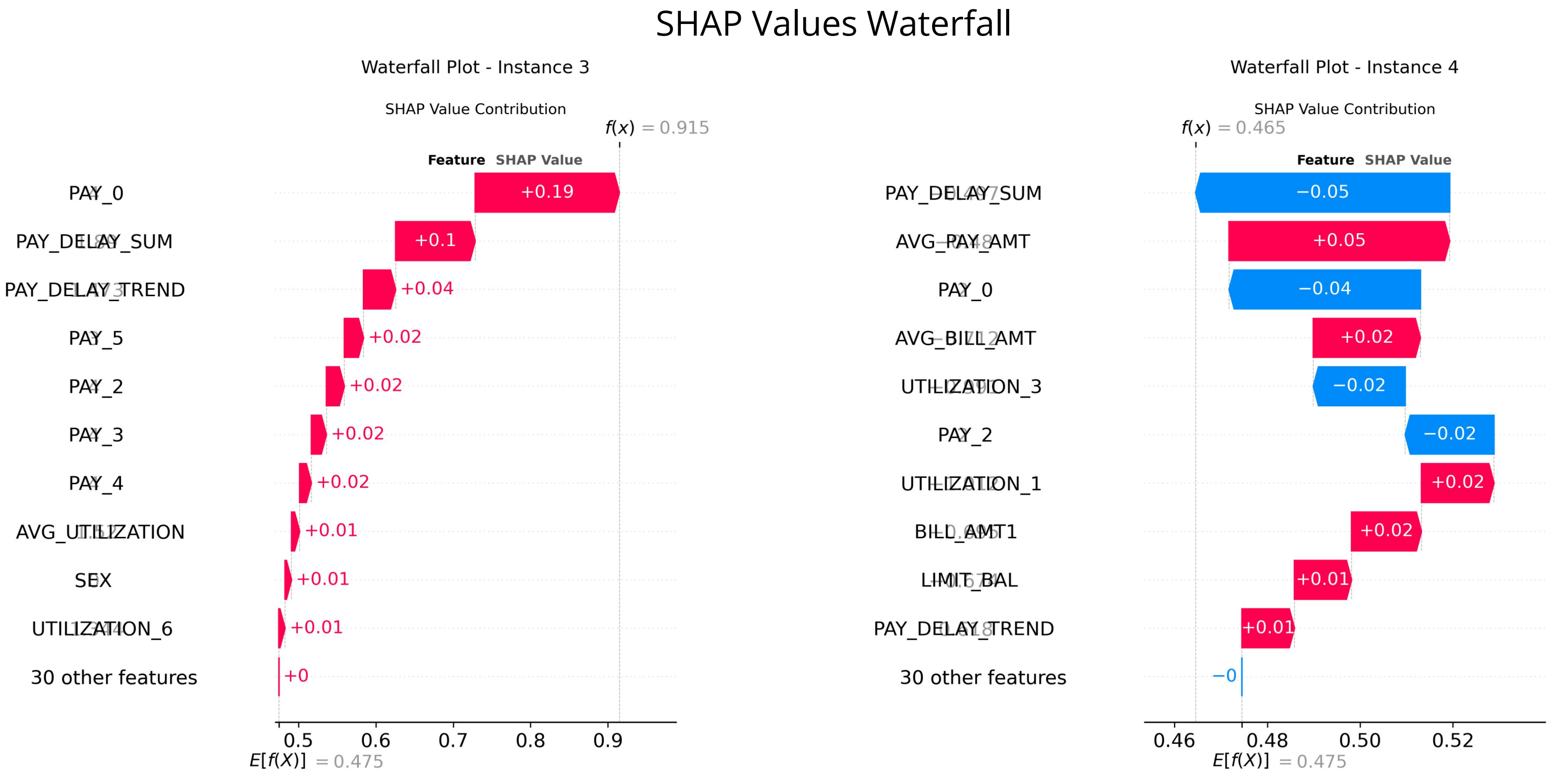
SHAP Values Waterfall





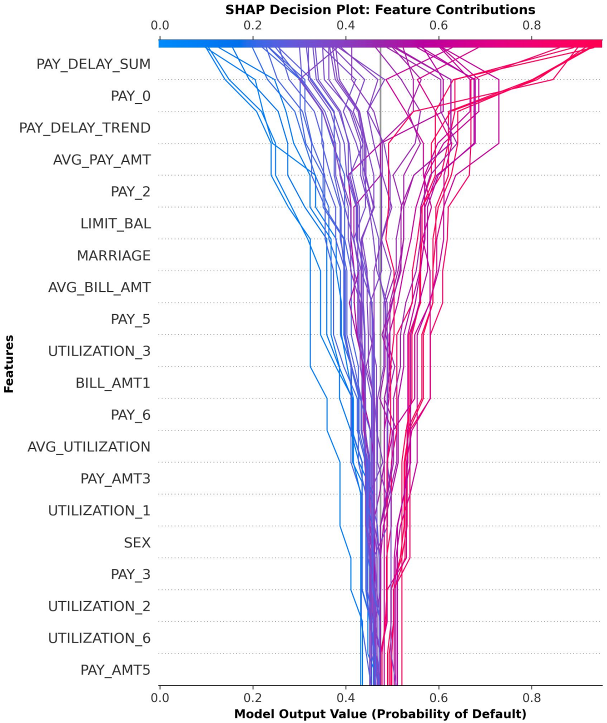
negie
lon
University

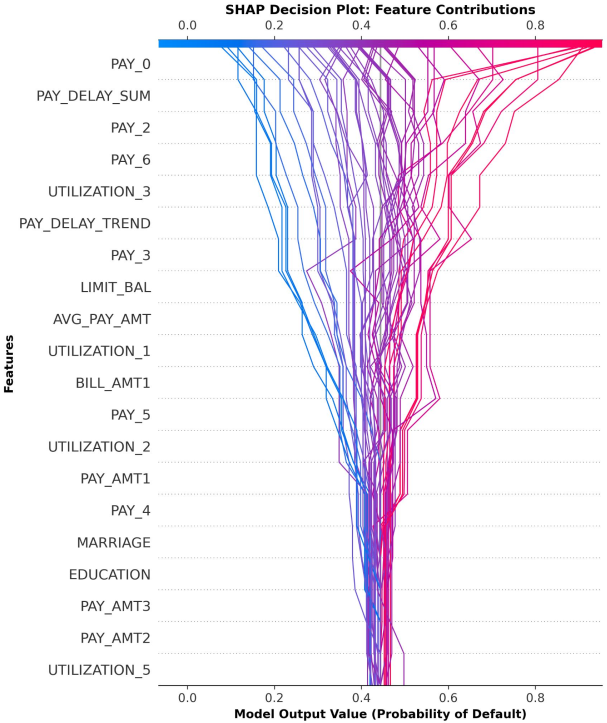
Experimental Results - Interpretability



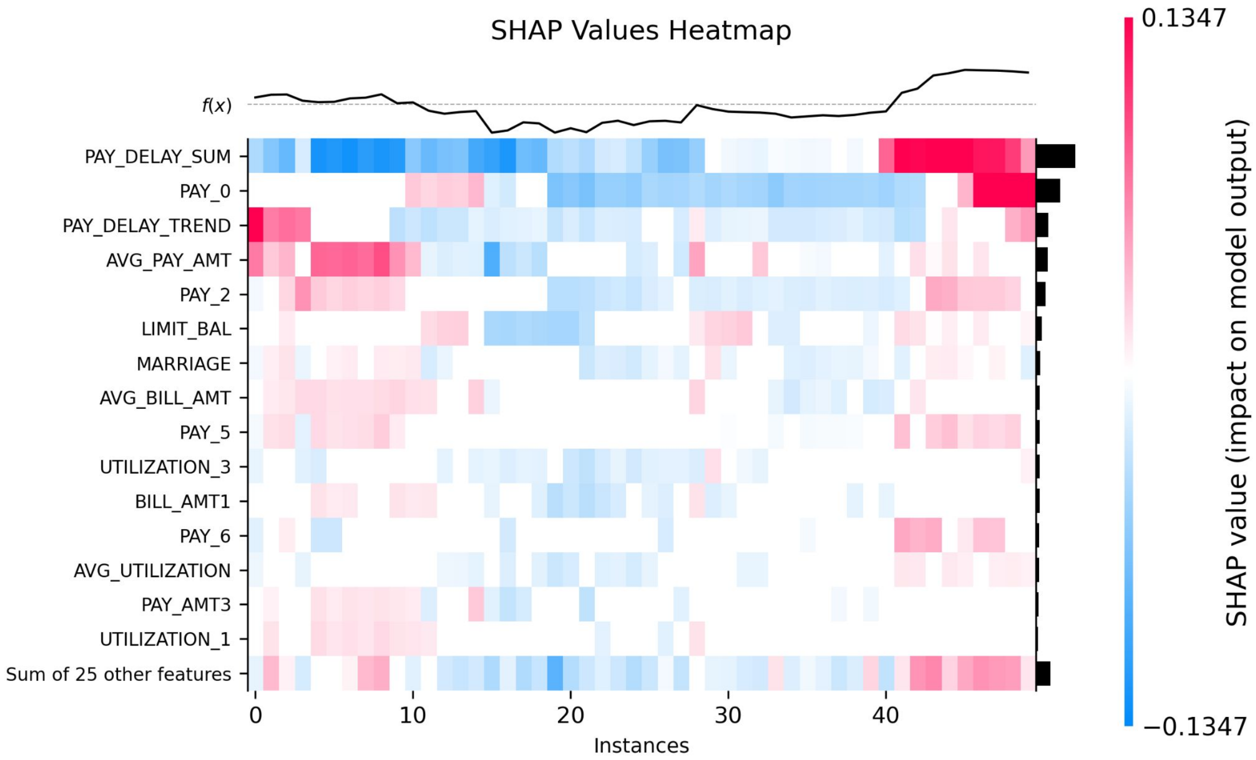
Experimental Results - Interpretability

SHAP Values Decision

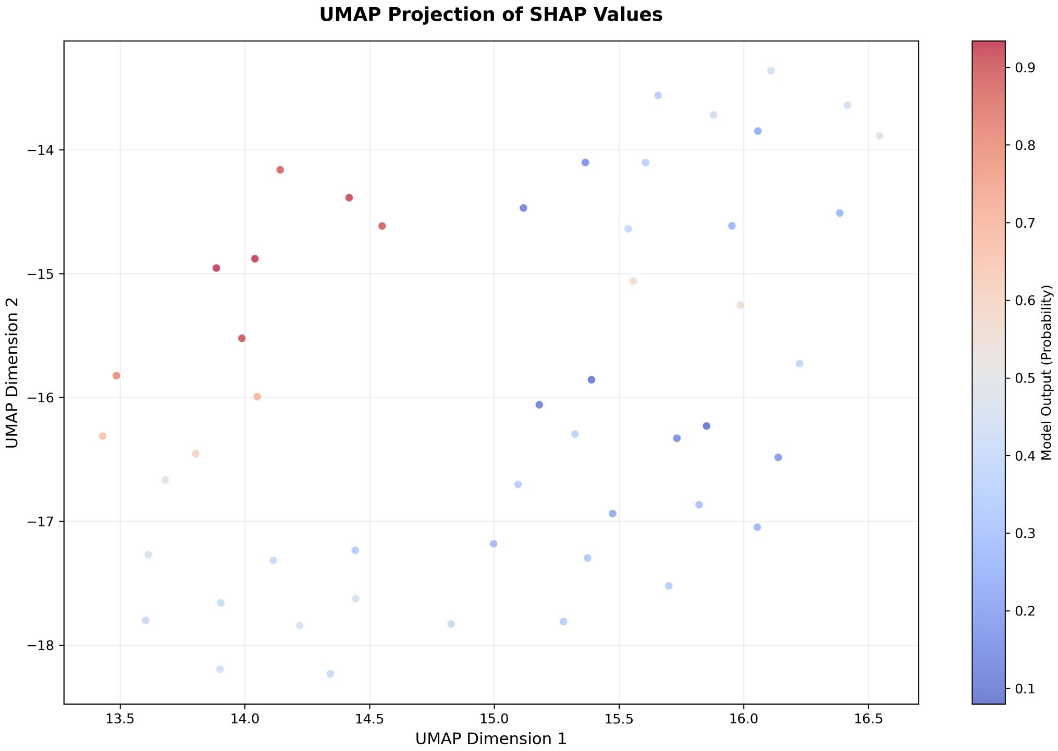




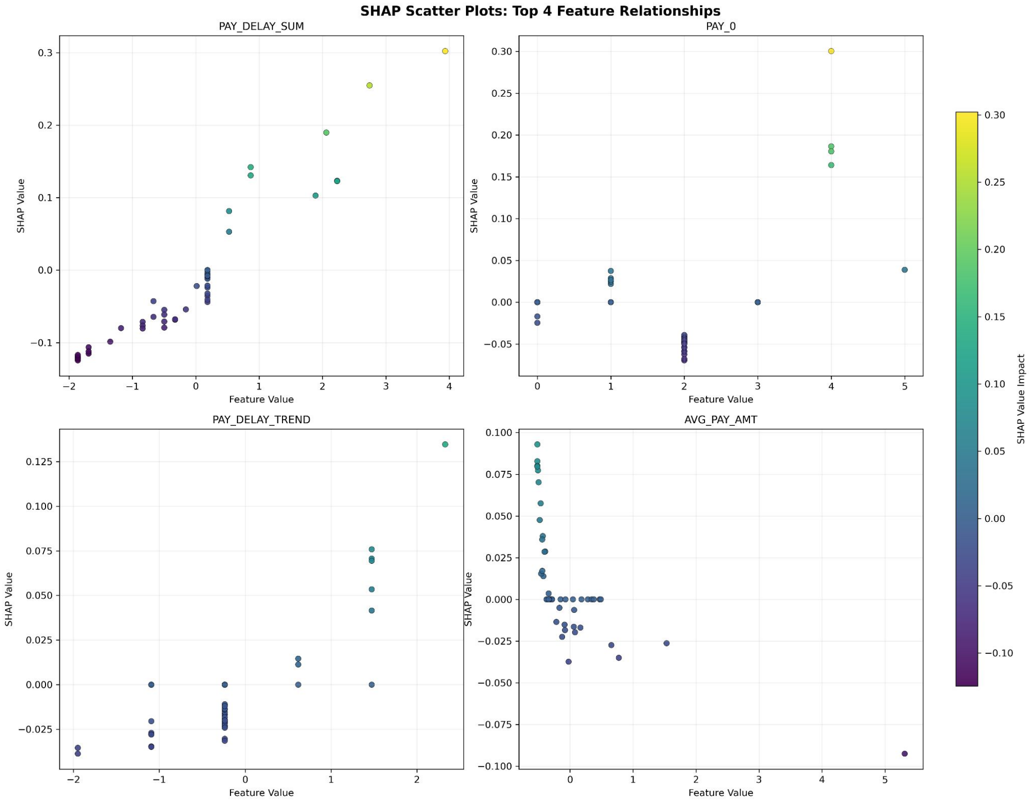
Experimental Results - Interpretability



Experimental Results - Interpretability



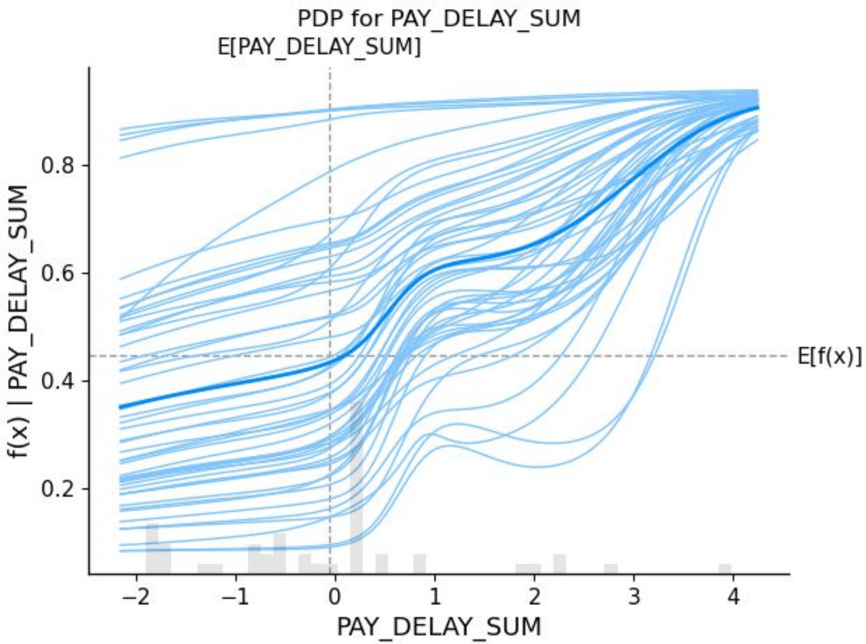
Experimental Results - Interpretability

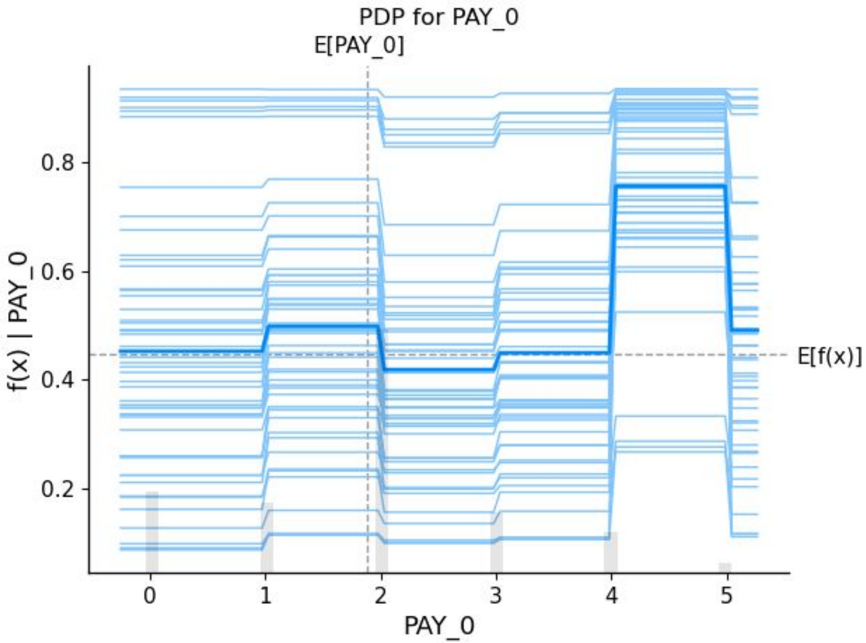


Carnegie Mellon University

Experimental Results - Interpretability

Partial Dependence Plot





Future Plan

- More Advanced or Hybrid Generative Modeling
    - Explore hybrid architectures (e.g., combining TabDiff + GAN)
- Extend evaluation metrics
    - Go beyond AUC-ROC and F1-score by analyzing fairness, robustness, and calibration of classifiers trained on synthetic data.
- Apply more real-world datasets
    - Explore integration of synthetic data into actual credit scoring systems or model validation workflows.

Future Plan

- Interpretable Learning Techniques
  - Develop attention sparsity constraints for more focused explanations
  - Use causal feature attribution to reduce spurious correlations
  - Visualize feature interaction graphs from attention matrices
- Systematic Ablation Studies
  - Quantify impact of each module: generation, classifier, interpretability
  - Evaluate synthetic-vs-real training dynamics over multiple seeds

References

- Yeh, I. (2009). Default of Credit Card Clients [Dataset]. UCI Machine Learning Repository. doi.org/10.24432/C55S3H
- Shriyank Somvanshi, Subasish Das, Syed Aaqib Javed, Gian Antariksa, Ahmed Hossain: “A Survey on Deep Tabular Learning”, 2024; http://arxiv.org/abs/2410.12034 arXiv:2410.12034.
- T. M. Alam et al., "An Investigation of Credit Card Default Prediction in the Imbalanced Datasets," in IEEE Access, vol. 8, pp. 201173-201198, 2020, doi: 10.1109/ACCESS.2020.3033784.
- Haque Ishfaq, Assaf Hoogi, Daniel Rubin: “TVAE: Triplet-Based Variational Autoencoder using Metric Learning”, 2018; http://arxiv.org/abs/1802.04403 arXiv:1802.04403.
- José-Manuel Peña, Fernando Suárez, Omar Larré, Domingo Ramírez, Arturo Cifuentes: “A Modified CTGAN-Plus-Features Based Method for Optimal Asset Allocation”, 2023; http://arxiv.org/abs/2302.02269 arXiv:2302.02269.

References

- ❑ Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, Artem Babenko: “TabDDPM: Modelling Tabular Data with Diffusion Models”, 2022, Proceedings of the 40 th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023; http://arxiv.org/abs/2209.15421 arXiv:2209.15421.
- ❑ Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, Jure Leskovec: “TabDiff: a Mixed-type Diffusion Model for Tabular Data Generation”, 2024, ICLR 2025; http://arxiv.org/abs/2410.20626 arXiv:2410.20626.
- ❑ Xin Huang, Ashish Khetan, Milan Cvitkovic, Zohar Karnin: “TabTransformer: Tabular Data Modeling Using Contextual Embeddings”, 2020; http://arxiv.org/abs/2012.06678 arXiv:2012.06678.
- ❑ Huangliang Dai, Shixun Wu, Hairui Zhao, Jiajun Huang, Zizhe Jian, Yue Zhu, Haiyang Hu, Zizhong Chen: “FT-Transformer: Resilient and Reliable Transformer with End-to-End Fault Tolerant Attention”, 2025; http://arxiv.org/abs/2504.02211 arXiv:2504.02211.