THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Medical AI Research: Predicting perioperative blood transfusion for CABG patients

Capstone Final Presentation, May 2, 2024

Team: Jenny Tsai [1] & Jichong Wu [1]
Collaborator: Dr. Puneet Gupta [2]
Advisor: Professor Amir Jafari [1]

[1] George Washington University
[2] George Washington University Hospital

Overview

Introduction

Data Preprocessing

Analysis Strategy

Results

- Phase 1: Classical models
- Phase 2: Neural networks

Conclusions

Introduction

Coronary Artery Bypass Graft (CABG) is a common cardiac surgery

- May cause major bleeding which needs blood transfusion

Blood transfusion is associated with:

- Higher risks of mortality after surgery
- Higher odds of readmission and heart failure within 30 days

Research Gap

Previous research*:

- A single cardiac surgery center in Austria
- N = 3782 (2010-2019)
- Random Forest:

  AUC: 0.76-0.86

In the current project:

- US national database [ACS NSQIP]
- N = 13534 (2015-2022)
- Classical models + Neural networks + Feature engineering/selection + data synthesis

*Tschoellitsch et al. (2022)

Objectives

1. **Develop models that can best predict which CABG patients need blood transfusion**
    - Improve patient selection and education
    - Enhance physician preoperative awareness
    - Inform periop guidelines for CABG patients

2. **Experiment with different DS techniques** (e.g., feature selection, feature engineering, synthetic data) applied in classical and advanced models **to achieve best outcomes**

3. **Develop a full set of modules that can be reused in the future,** which covers preprocessing, feature selection and feature engineering, and modeling

Data Preprocessing

Datasets

- Participant Use Data File (PUF) on the American College of Surgeons National Surgical Quality Improvement Program (ACS NSQIP)
- Year 2015 - 2022 (N = 13534, # of features = 296)

Key preprocessing steps:

1. Basic clean-up (e.g., recode values, correct data type)
2. Remove columns with over 50% missing values
3. Impute with mean (numeric) and most frequent values (categorical)
4. Standardize all numeric features
5. Remove post-operative and irrelevant features by expert

Final dataset size:

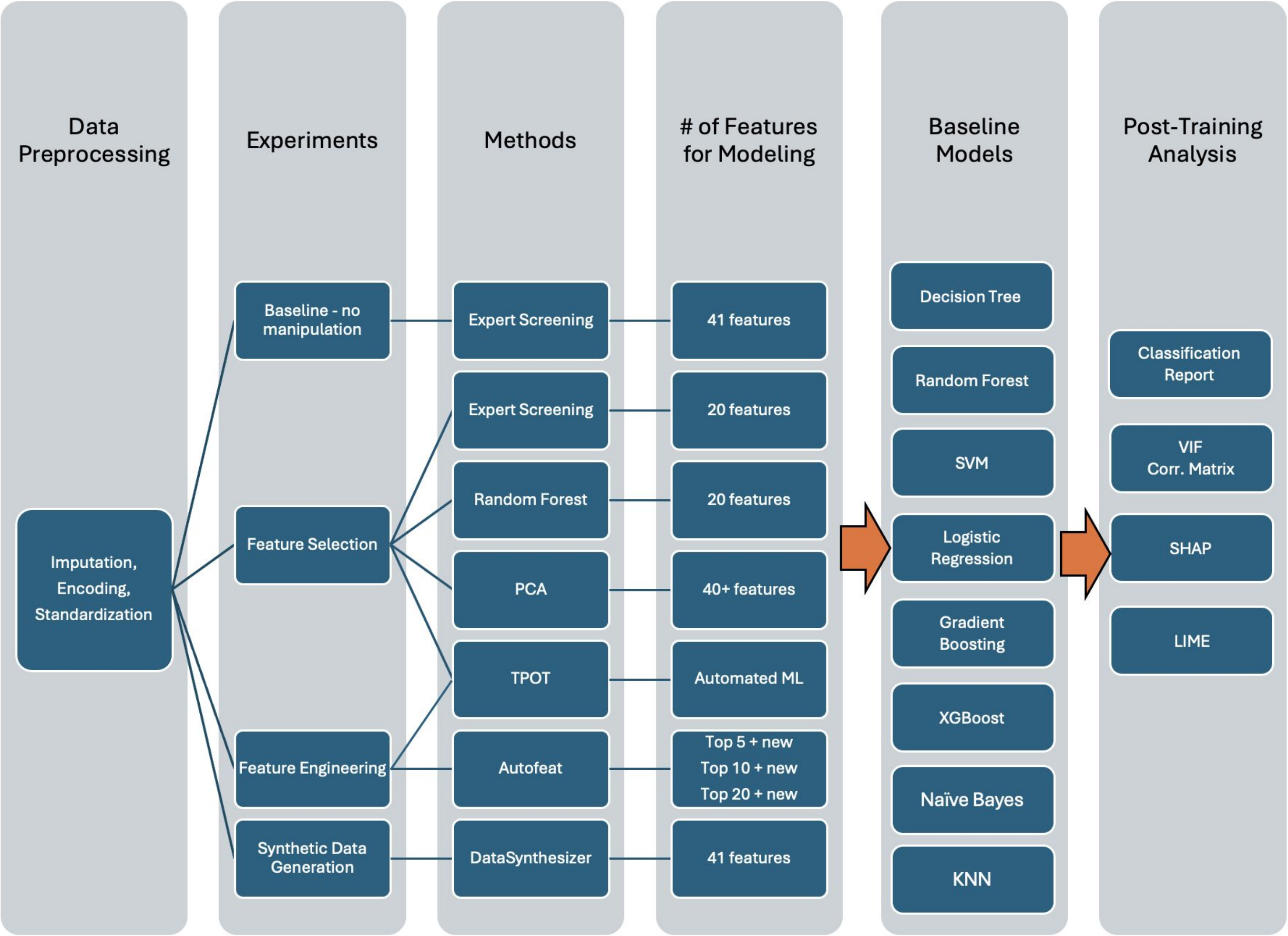
N = 13534, # of features = 41

Phase 1: Basic Models

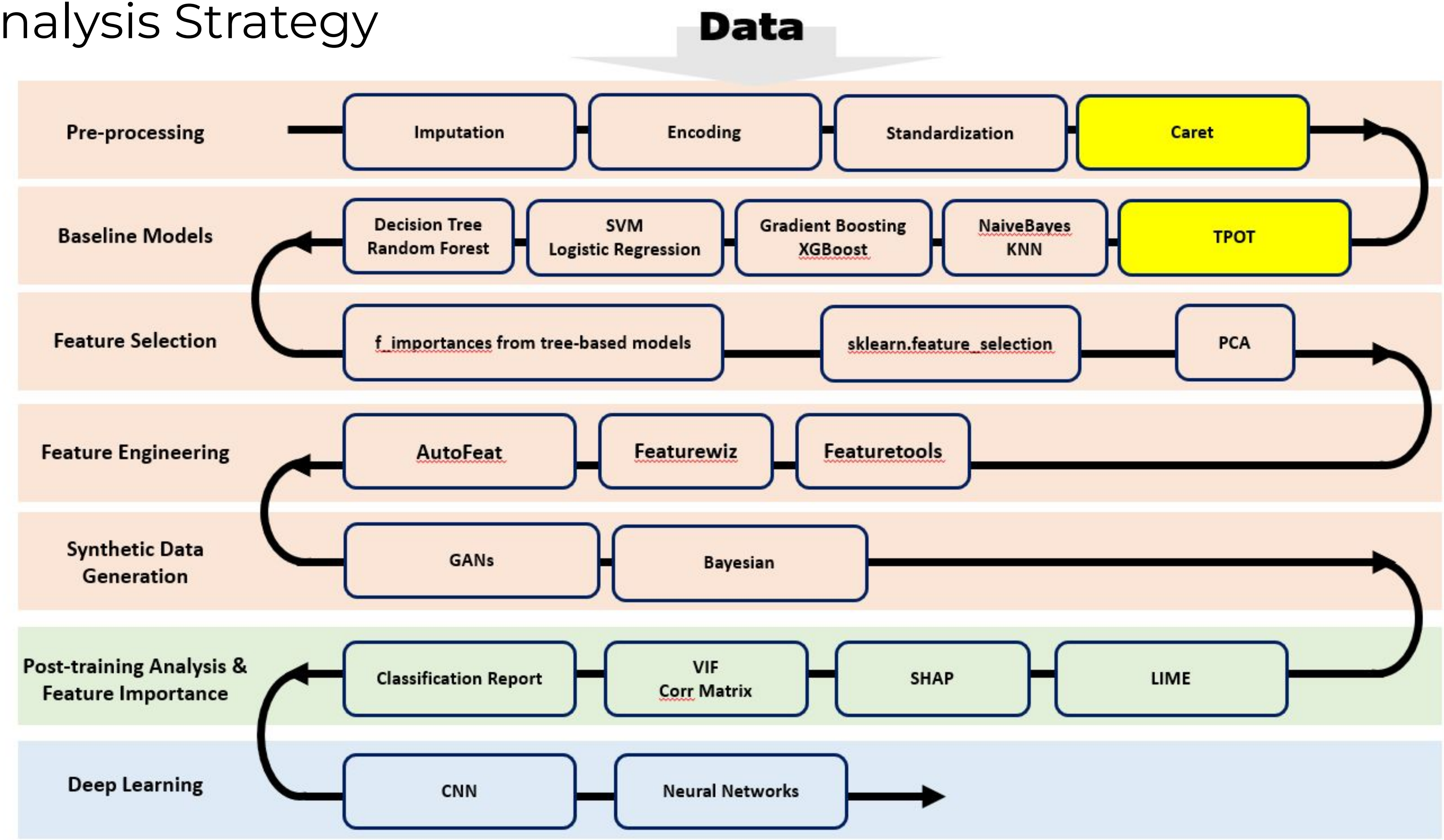
THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Analysis Strategy



Analysis Strategy

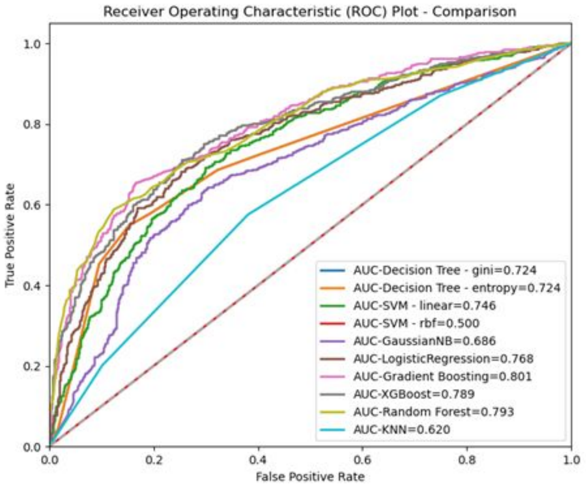


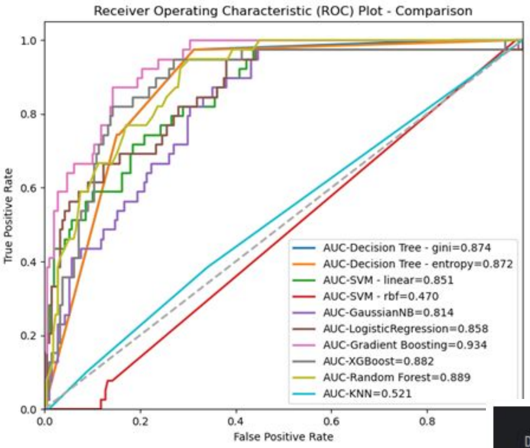
Modules and Utilities

class - data pre-processing		class - baseline models	class - feature selection		class - feature engineering	utility
class name	methods		class name	methods		
datasci	.size()	DecisionTree	PCA	PCA_Reduced_Feature()	featurewiz	file_compare()
	.recode()	RandomForest		Reduced_Feature_Space_Plot()	featuretools	glossary()
	.missingReport()	SVM		Reduced_Feature_Space_Heatmap()		Model_Predict()
	.remove_all_nan_columns()	LogReg		Explained_Variance_Ratio()		Model_Report()
	.impute_all()	GradientBoosting		Reduced_Feature_Space_Plot()		Model_Accuracy()
	.imputation()	XGB		Reduced_Feature_Space_Heatmap()		Model_Mean_Accuracy()
	.standardize()	NaiveBayesGaussianNB		PCA_New_df()		Model_RMSE()
	.eda()	KNN				Model_F1()
	.featureSelection()	TPOT				Model_Confusion_Matrix()
						Plot_Confusion_Matrix()
						Plot_Decision_Tree()
						Model_ROC_AUC_Score()
						Plot_ROC_AUC()
						Plot_Random_Forest_Feature_Importances()
						Model_Results_Table()
						Plot_ROC_Combined()
						Calc_Plot_VIF()
						Calc_Top_Corr()
						Plot_Heatmap_Top_Corr

Model Results

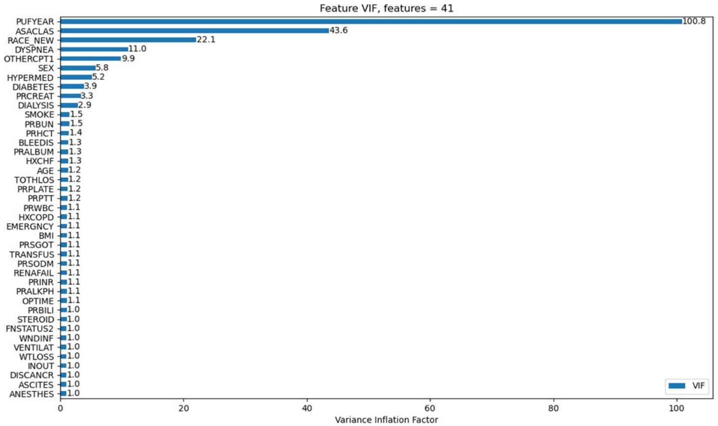
Model Name	Parameters	Target	Test Size	Accuracy	RMSE	F1-score (macro avg)	ROC-AUC score
Decision Tree - gini	max_depth=3, min_samples_leaf=5	OTHBLEED	0.25	68.119451	0.564629	0.679578	0.724499
Decision Tree - entropy	max_depth=3, min_samples_leaf=5	OTHBLEED	0.25	68.119451	0.564629	0.679578	0.724499
SVM - linear	C=1.0, gamma=auto	OTHBLEED	0.25	69.491525	0.552345	0.687647	0.746498
SVM - rbf	C=1.0, gamma=0.2	OTHBLEED	0.25	55.528652	0.666868	0.357032	0.500000
GaussianNB		OTHBLEED	0.25	52.945924	0.685960	0.496722	0.685814
LogisticRegression		OTHBLEED	0.25	71.751412	0.531494	0.712662	0.767803
Gradient Boosting	n_estimators=300, learning_rate=0.05	OTHBLEED	0.25	73.930589	0.510582	0.734304	0.801463
XGBoost	n_estimators=100, eta=0.3	OTHBLEED	0.25	72.558515	0.523846	0.721558	0.789397
KNN	n_neighbors=3	OTHBLEED	0.25	59.967716	0.632711	0.596555	0.619534
Random Forest	n_estimators=100, 20_features	OTHBLEED	0.25	73.930589	0.510582	0.729318	0.792982

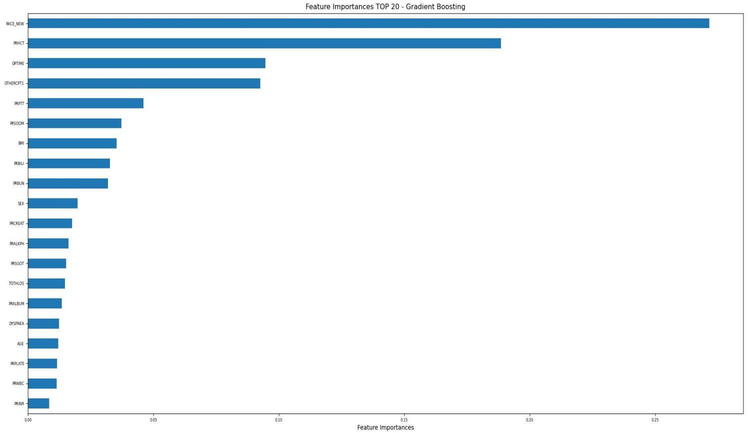


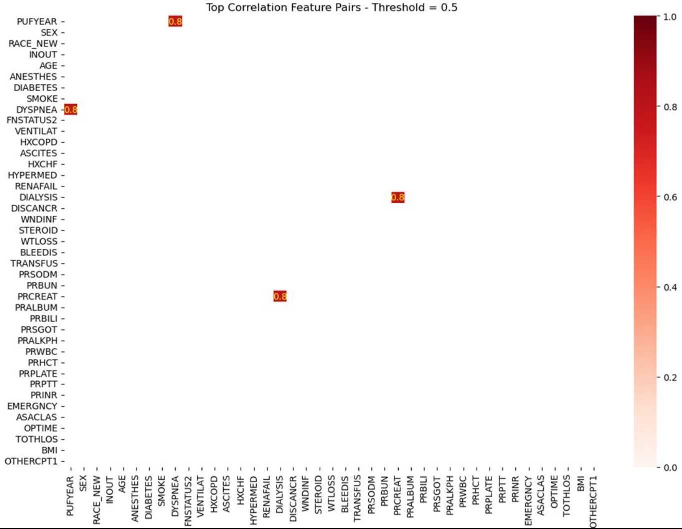


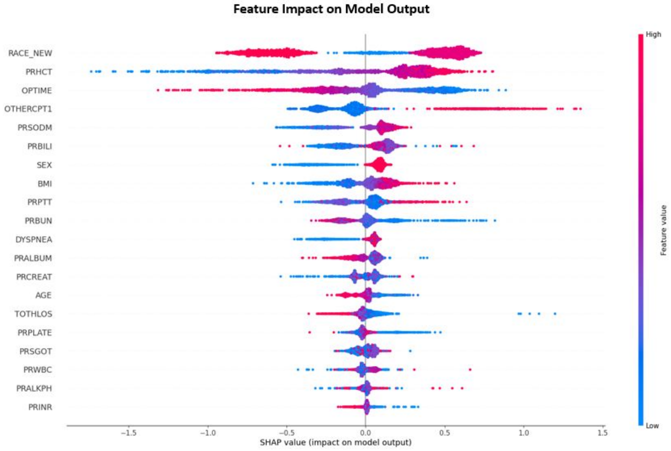
Model Name	Parameters	Target	Test Size	Accuracy	Mean Accuracy (10 folds)	RMSE	F1-score (macro avg)	ROC-AUC score
Decision Tree - gini	max_depth=3, min_samples_leaf=5	OTHBLEED	0.25	83.2	86.9	0.409878	0.737500	0.874408
Decision Tree - entropy	max_depth=3, min_samples_leaf=5	OTHBLEED	0.25	83.2	86.9	0.409878	0.737500	0.872159
SVM - linear	C=1.0, gamma=auto	OTHBLEED	0.25	86.8	86.0	0.363318	0.735331	0.851136
SVM - rbf	C=1.0, gamma=0.2	OTHBLEED	0.25	84.4	83.9	0.394968	0.457701	0.469741
GaussianNB		OTHBLEED	0.25	72.0	79.5	0.529150	0.628639	0.814315
LogisticRegression		OTHBLEED	0.25	87.2	86.7	0.357771	0.751738	0.858306
Gradient Boosting	n_estimators=300, learning_rate=0.05	OTHBLEED	0.25	89.6	90.7	0.322490	0.789071	0.933649
XGBoost	n_estimators=100, eta=0.3	OTHBLEED	0.25	85.2	88.6	0.384708	0.737775	0.881638
KNN	n_neighbors=3	OTHBLEED	0.25	78.4	81.6	0.464758	0.502872	0.521388
Random Forest	n_estimators=100, 20_features	OTHBLEED	0.25	88.0	85.4	0.346410	0.715909	0.888747

Post-training Analysis









Phase 1 modeling summary

1. PUFYEAR, ASACLAS, RACE_NEW may cause multicollinearity.
2. **Gradient Boosting**, XGBoost, Random Forest perform the best.
3. **Synthetic data generation** techniques (DataSynthesizer using Bayesian networks) significantly improve model performance.
4. **Race, days from preoperative labs to operation, operation time**, other procedure, BMI, sex, length of hospital stay, shortness of breath, age, preoperative blood test measures
5. **days from preoperative labs to operation, operation time may have a negative impact on model results while other procedure, BMI have a positive effect.**

Phase 2: Neural Networks

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Major Types of Artificial Neural Networks

- **FNNs** are simpler type of neural network where data and information flows in one direction from input layer to output layer.
- **RNNs** are designed for sequential data processing, where the output at each step depends not only on the current input but also on previous inputs in the sequence. They are commonly used for tasks like natural language processing (NLP), time series analysis, and speech recognition.
- **CNNs** are designed for processing grid-like data such as images. They use convolutional layers to detect patterns and features in the input data, making them highly effective for tasks like image recognition and object detection.
- **LSTMs** are a type of RNN designed to address the vanishing gradient problem and handle long-term dependencies in sequential data. They are particularly effective for tasks that require capturing long-term patterns and dependencies, such as machine translation, sentiment analysis, and time-series modeling.
- **GANs** consist of two neural networks, the generator and the discriminator, that are trained together in a competitive setting to create new data from a given training dataset. The generator creates new data samples, while the discriminator distinguishes between real and generated samples. GANs are typically used for image generation and data augmentation.
- **RBFNs** use radial basis functions as activation functions and are distinguished from other neural networks due to their universal approximation and faster learning speed.

Literature Review on Related Work

- Deep neural networks models have shown excellent performance and especially when processing complex data such as image, text and sound. However, their adaptation to tabular data tasks remains highly challenging
- No sufficient evidence that neural networks are better than classical models such as gradient boosting decision trees
- Gradient boosting methods outperform NNs
- A hybrid approach of gradient boosting plus deep neural networks performs the best

Synthetic Data Tools

DataSynthesizer

- Based off **Bayesian Networks**, which represents a graphical model of the joint probability distribution for a set of attributes
- Probabilistic inference about one attribute in the network given the values of other attributes → missing data imputation, synthetic data generation
- Three modules: DataDescriber, DataGenerator, ModelInspector

REaLTabFormer (Realistic Relational and Tabular Data using Transformers)

- Relational data: **A sequence-to-sequence (Seq2Seq) model**
- Non-relational data: GPT-2

Feedforward Neural Networks (FNNs)

FNN Design

8 models varying in:

- Complexity (# of layers, # of neurons)
- Output activation function (sigmoid vs. softmax)
- Loss function (binary vs. categorical cross entropy)
- Optimizer (SGD vs. Adam)

Datasets

- Original
- REaLTabFormer
- DataSynthesizer

1. FNNs with Original Data

Model	accuracy	f1_score	rMSE	AUC
FNN-5layer-SGD-sigmoid	0.7215	0.6888	0.4794	0.7764
FNN-5layer-Adam-sigmoid	0.7218	0.6961	0.4928	0.7756
FNN-7layer-SGD-sigmoid	0.7174	0.6954	0.4850	0.7741
FNN-7layer-Adam-sigmoid	0.7174	0.6782	0.5735	0.7752
FNN-5layer-SGD-softmax	0.7229	0.7200	0.4643	0.7782
FNN-5layer-Adam-softmax	0.7218	0.7179	0.4975	0.7815
FNN-7layer-SGD-softmax	0.7181	0.7150	0.4780	0.7780
FNN-7layer-Adam-softmax	0.7185	0.7118	0.5587	0.7766

2. FNNs with Synthetic Data - REaLTabFormer

Model	accuracy	f1_score	rMSE	AUC
FNN-5layer-SGD-sigmoid	0.7362	0.7420	0.4789	0.8006
FNN-5layer-Adam-sigmoid	0.7359	0.7394	0.4650	0.8007
FNN-7layer-SGD-sigmoid	0.7303	0.7392	0.4819	0.7939
FNN-7layer-Adam-sigmoid	0.7351	0.7315	0.5577	0.7972
FNN-5layer-SGD-softmax	0.7381	0.7380	0.4506	0.8008
FNN-5layer-Adam-softmax	0.7355	0.7355	0.4698	0.7999
FNN-7layer-SGD-softmax	0.7362	0.7362	0.4787	0.7995
FNN-7layer-Adam-softmax	0.7303	0.7286	0.5575	0.7934

3. FNNs with Synthetic Data - DataSynthesizer

Model	accuracy	f1_score	rMSE	AUC
FNN-5layer-SGD-sigmoid	0.8718	0.8776	0.6790	0.9386
FNN-5layer-Adam-sigmoid	0.8689	0.8772	0.4340	0.9474
FNN-7layer-SGD-sigmoid	0.8685	0.8774	0.5853	0.9358
FNN-7layer-Adam-sigmoid	0.8626	0.8712	0.4222	0.9456
FNN-5layer-SGD-softmax	0.8751	0.8749	0.6119	0.9421
FNN-5layer-Adam-softmax	0.8696	0.8696	0.4286	0.9495
FNN-7layer-SGD-softmax	0.8762	0.8762	0.5596	0.9401
FNN-7layer-Adam-softmax	0.8670	0.8667	0.4513	0.9474

4. CNNs with Synthetic Data

Model	dataset	accuracy
CNN-2D-2layer-noPooling-ReLU	synthetic dataset 2015-2022 from REaLTabFormer	0.6681
	synthetic dataset 2015-2022 from DataSynthesizer	0.5785

Conclusions

- For both basic models and neural networks, best performance was found in models (classical & NNs) using synthetic data from DataSynthesizer
- Bayesian networks might have some systematic influence on optimizers
- Compared with FNNs, CNNs might not be suitable for analyzing tabular data

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Thank You!

THE GEORGE
WASHINGTON
UNIVERSITY

WASHINGTON, DC