Benchmarking Architectures for Churn Prediction: The Impact of Data Leakage

Seminar in Machine Learning

Neta Ben Mordechai, Itay Chabra, Yuval Gefen

Why is this important

Customer churn is a huge issue for companies. Companies want to predict who will leave so they can offer them a better deal.

Most companies use standard machine learning, but recently, researchers have started using complex Deep Learning models.

The Problem with Previous Research

We looked at a study called "ChurnNet" that claimed 99% accuracy.

Data Leakage via SMOTE

They used a technique called SMOTE (creating fake data) before splitting their data.

This causes "Data Leakage". The model accidentally saw the test answers during training.

The Problem with Previous Research

1D-CNN on Tabular Data

Convolutional Neural Networks (CNNs) rely on Spatial Locality (pixels next to each other matter).

In a tabular dataset, the order of columns is random. You can shuffle the columns, and the data is still the same.

The Right Approach

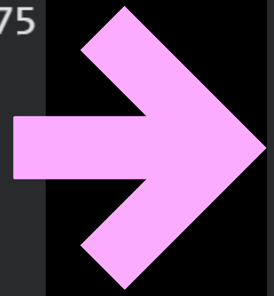
How we ensured fair and realistic results:

Instead of "global SMOTE" (the mistake), we applied SMOTE only inside the training folds.

The model never sees, touches, or uses test data for scaling or balancing.

This ensures the model learns from synthetic examples but is tested on real human customers.

```
Average Test Accuracy: 0.9747455957982275
Average Precision: 0.9759833779460575
Average Recall: 0.9798172935123999
Average F1 Score: 0.9778377871272976
Average MCC: 0.9486479951935612
Average AUC-ROC: 0.9956133759672703
```



```
Average Test Accuracy: 0.8772
Average Precision: 0.5577465474180181
Average Recall: 0.7397786720321932
Average F1 Score: 0.6313394309230292
Average MCC: 0.5713707278507549
Average AUC-ROC: 0.8735919052560062
```

Models We Compared

XGBoost and LightGBM:

These models create thousands of small Decision Trees that vote on whether a customer will leave. If one tree makes a mistake, the next tree focuses specifically on fixing that error.

In the data science world, these are the default tools. They are fast, accurate, and tough to beat.

Models We Compared

We tested three generations of Neural Networks to be thorough:
Deep MLP: It connects every input to every neuron. It’s powerful but often struggles to distinguish "noise" from "signal" in tables.

TabNet: A Neural Network designed to mimic a Decision Tree. It uses a "Soft Attention" mechanism to focus only on the important columns and ignore the rest.

FT-Transformer: It looks at how every column relates to every other column to find complex context.

Models Comparison

We tested all models using SMOTE to see how they perform with "standard" data balancing.

Model	Accuracy	F1-Score	AUC	Precision	Recall	MCC
XGBoost	95.44%	0.8365	0.9148	0.8487	0.8261	0.8106
LightGBM	95.28%	0.8296	0.9170	0.8499	0.8119	0.8031
FT-Transformer	93.74%	0.7904	0.9161	0.7643	0.8246	0.7566
Attentive MLP	93.42%	0.7809	0.9238	0.7458	0.8258	0.7453
TabNet	92.00%	0.7359	0.8996	0.6959	0.7850	0.6921

Models Comparison No SMOTE

Using Class Weights instead of SMOTE.
XGBoost performed better without SMOTE (0.848 vs 0.836).

Model	Accuracy	F1-Score	AUC	Precision	Recall	MCC
XGBoost	95.96%	0.8482	0.9220	0.9059	0.7990	0.8278
FT-Transformer	95.82%	0.8466	0.9226	0.8840	0.8133	0.8238
LightGBM	95.86%	0.8426	0.9200	0.9093	0.7906	0.8236
MLP-Attention	94.30%	0.7919	0.9022	0.8164	0.7724	0.7606
TabNet	93.98%	0.770	0.9216	0.8383	0.7183	0.7410

Does this hold up in a different industry?

We tested the models on a Bank Churn Dataset to see if the results were consistent.
This dataset has less "behavioral" data, making prediction harder.

Model	Accuracy	F1-Score	AUC	Precision	Recall	MCC
XGBoost	84.96%	0.6170	0.8626	0.6464	0.5966	0.5270
LightGBM	84.52%	0.6151	0.8607	0.6335	0.6075	0.5225
MLP-Attention	84.10%	0.6149	0.8539	0.6060	0.6241	0.5156
FT-Transformer	84.89%	0.6095	0.8606	0.6530	0.5785	0.5208
TabNet	83.23%	0.5920	0.8428	0.5962	0.6015	0.4921

Business Impact

Precision: Winner - XGBoost

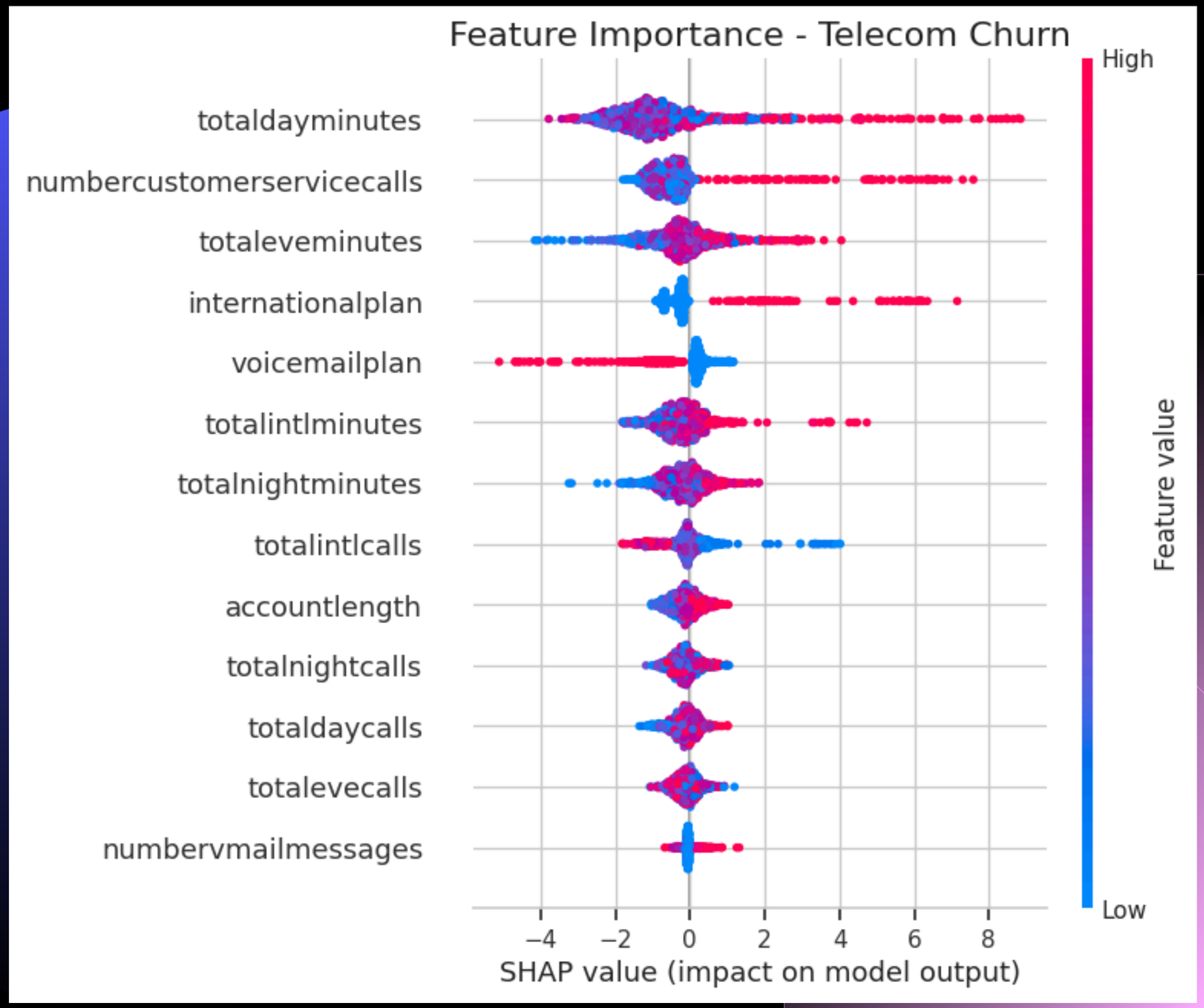
- Score: 90.6% (vs. 88% for Transformer).
- Meaning: When XGBoost says a customer will leave, it is almost always right.
- Value: You don't waste money giving discounts to happy customers.

Recall: Winner - FT-Transformer

- Score: 81.3% (vs. 79.9% for XGBoost).
- Meaning: The Transformer casts a wider net and catches churners that XGBoost misses.
- Cost: It generates more false alarms.

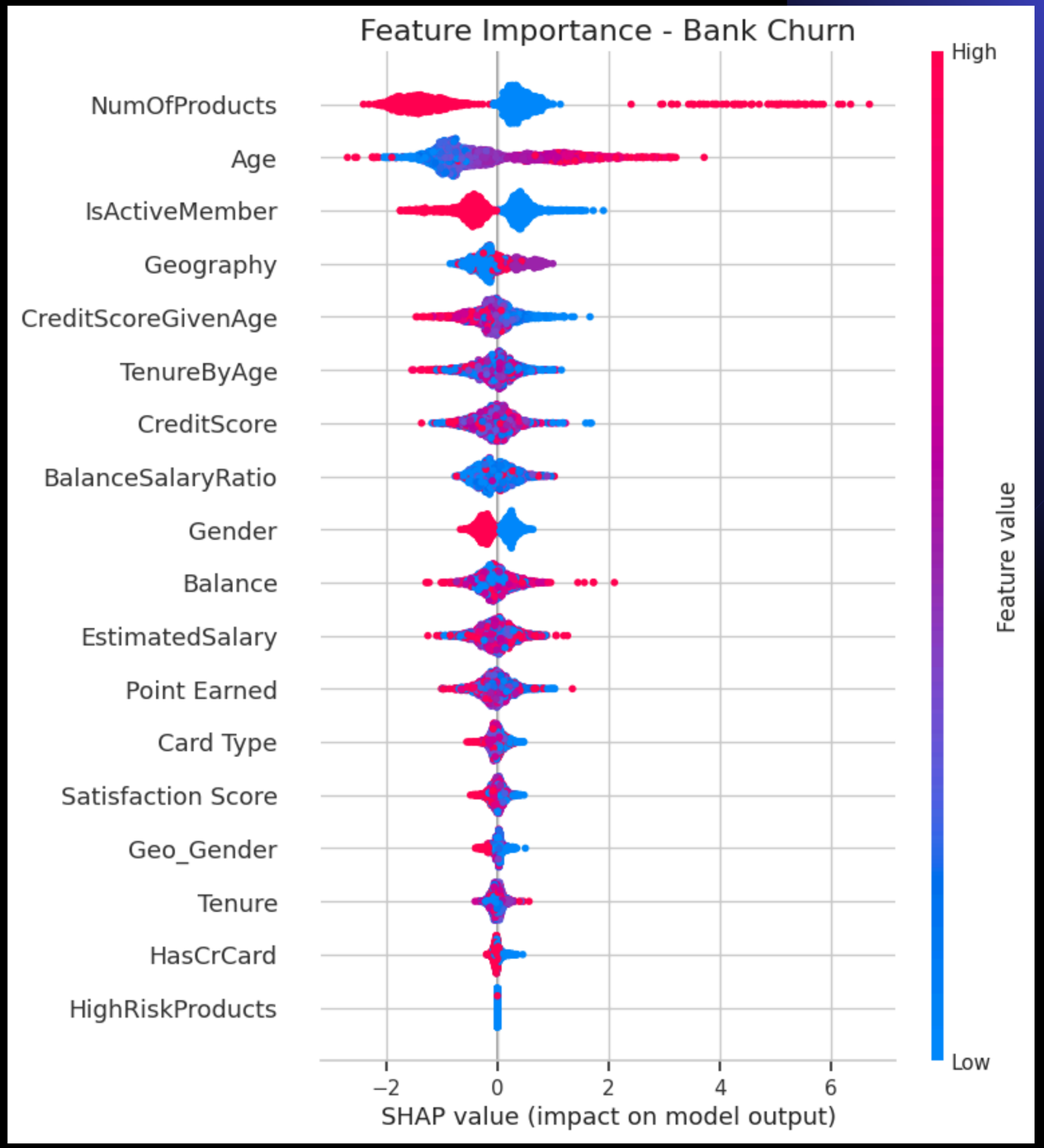
Adding XAI

Telecom Churn Dataset



Adding XAI

Bank Churn Dataset



Conclusions

The 99% accuracy in the original paper was just the model memorizing the answers (Data Leakage).

You don't need a massive Neural Network for an Excel sheet. **XGBoost** is faster, cheaper to run, and beat the complex models in almost every test.

The FT-Transformer was the only Neural Network that gave XGBoost a fair fight.

You can't fix bad data with a smart model. Feature Quality matters more than which algorithm you choose.

Thank **You**