Predicting Churn: A Comparative Study of Random Forests and Tabular Neural Networks

Nick Blackford

# Business Problem

Customer churn poses a significant financial risk to subscription-based businesses like telecommunications providers. Retaining existing customers is typically more cost-effective than acquiring new ones, making churn prediction a critical application of data science. The goal of this project is to develop a predictive model that accurately identifies customers at risk of leaving, enabling proactive retention strategies.

To evaluate the strengths and trade-offs of different modeling approaches, we compare a traditional ensemble method (Random Forest) against a Tabular Neural Network (TNN)—a modern deep learning architecture tailored for structured data. The outcome of this comparison is not just better predictive performance, but also insight into whether advanced methods justify their added complexity in real-world business scenarios.

# Background/History

The telecommunications industry has long grappled with high churn rates, driven by fierce competition, commoditized pricing, and low switching costs. Traditional business strategies to reduce churn often rely on broad marketing campaigns or retention offers applied post-hoc—efforts that are costly and frequently mistimed.

With the rise of data-driven decision-making, churn prediction has become a key focus area for data science teams. Machine learning models can identify subtle behavioral patterns and risk indicators that might elude heuristic rules or manual analysis. While classical models like logistic regression and decision trees have served this domain well, recent advancements in deep learning—specifically tabular neural networks—present an opportunity to revisit the modeling pipeline and potentially improve both predictive accuracy and actionable insight.

This project sits at the intersection of traditional and modern modeling, evaluating whether the additional complexity and tuning required by a neural network are justified when compared to a well-calibrated Random Forest.

# Data Explanation

This project leverages the publicly available Telco Customer Churn dataset, originally released by IBM. It includes 7,043 customer records with 21 features representing demographic data, account information, service usage, and a churn label indicating whether the customer left the company.

The target variable is Churn, a binary label (Yes/No) that we encoded for modeling. Input features include a mix of numerical variables (tenure, MonthlyCharges, TotalCharges) and categorical variables such as Contract, PaymentMethod, InternetService, and several service-level flags (e.g., OnlineSecurity, TechSupport).

Prior to modeling, the data required several cleaning and transformation steps:

* **Missing values**: A small subset of entries in TotalCharges had blank strings, which we converted to NaN and imputed using the median.
* **Data type corrections**: TotalCharges was read as a string and explicitly cast to float.
* **Encoding**: Categorical features were automatically encoded using fastai’s tabular processing pipeline, which handles both categorical embedding and normalization of continuous inputs.
* **Target conversion**: The Churn column was binarized (1 for churn, 0 for retain) to serve as a valid classification target.

The final dataset was randomly split into training and validation sets using an 80/20 ratio. Importantly, we retained class distribution to avoid introducing additional imbalance.

# Methods

We employed a comparative modeling approach to predict customer churn. The workflow included:

1. **Baseline Model – Random Forest**: A well-known ensemble method, Random Forest was selected as our baseline for its robustness to feature interactions and ease of interpretability. Hyperparameters were kept at default for this baseline to quickly gauge signal strength in the dataset.
2. **Advanced Model – Tabular Neural Network (TNN)**: We used fastai’s high-level tabular\_learner API to build a feedforward neural network. The model utilized categorical embeddings for discrete features and batch normalization for continuous features. Architecture tuning included two hidden layers (500 and 250 units), ReLU activations, and dropout for regularization.
3. **Training & Evaluation**: Both models were trained on the same 80/20 train-validation split. The performance was evaluated using accuracy and ROC AUC, chosen to balance interpretability and sensitivity to class imbalance.
4. **Interpretability**: While Random Forest provides native feature importances, interpretability of the TNN was more nuanced. After failed SHAP integration, we derived basic feature contribution insights by analyzing learned weights and performance shifts during feature ablation. The TNN consistently prioritized the features tenure, MonthlyCharges, and SeniorCitizen in its predictions.

This two-model strategy not only highlights the benefits and trade-offs between classical and deep learning approaches, but also builds a compelling narrative for evaluating model fit in production scenarios.

# Analysis

Both models performed similarly in terms of accuracy (79%), but the Tabular Neural Network (TNN) achieved a slightly higher ROC AUC of 0.84 compared to the Random Forest’s 0.82. This suggests that while both models correctly classified the same proportion of customers, the TNN was better calibrated at ranking churn probabilities.

The TNN's ability to slightly outperform a strong baseline like Random Forest is notable, especially given the relatively small feature space and tabular format. This performance gain may be attributed to the neural network’s embedded representations of categorical features and its ability to model non-linear interactions through multiple dense layers. Fastai’s built-in callbacks like early stopping helped prevent overfitting and kept the training efficient.

From an interpretability perspective, the Random Forest’s feature importance aligned with domain expectations—tenure, contract type, and MonthlyCharges dominated. The TNN converged to similar predictors but showed an even heavier reliance on tenure and MonthlyCharges. Despite attempts with SHAP, a fallback method revealed that the TNN could achieve near-optimal predictions primarily using just three features. This simplification offers potential for deploying leaner scoring models in real-time systems.

Overall, the analysis demonstrates the TNN’s capability to extract marginally stronger signal from the data while reinforcing the insights surfaced by more traditional methods.

# Conclusion

This project demonstrated the successful application of both ensemble and neural network models to predict customer churn with strong accuracy and calibration. While Random Forest provided a robust and interpretable baseline, the Tabular Neural Network slightly outperformed it in ROC AUC and highlighted a more compact set of predictive features. These findings show that deep learning, when applied thoughtfully even on structured tabular data, can yield practical performance gains while reinforcing domain-relevant insights.

# Assumptions

This analysis assumes that the provided Telco churn dataset accurately reflects real-world conditions without significant sampling bias or data leakage. It also assumes that all variables are collected consistently and independently, that customer behavior is stable over time, and that the relationships captured in historical data persist in future observations. Additionally, model evaluations are based on the assumption that the validation split is representative of future unseen data.

# Limitations

Despite strong model performance, there are key limitations to acknowledge. The dataset represents only one company’s customer base, which may restrict generalizability. Categorical variables such as contract type or payment method may embed social or economic factors not captured directly in the data. Additionally, the Tabular Neural Network model—while powerful—is less interpretable than tree-based models and more sensitive to preprocessing and hyperparameter choices. Finally, we did not account for temporal trends or covariate shift, which could influence churn patterns over time.

# Challenges

Key challenges in this project included preprocessing messy categorical data, managing class imbalance, and tuning the neural network’s architecture to avoid underfitting or overfitting. Integrating interpretability tools like SHAP for deep models also proved technically complex due to model structure and input format issues. Balancing performance with transparency required iterative experimentation and thoughtful model selection.

# Future Uses

This modeling framework can be extended to other subscription-based businesses aiming to understand customer retention. Incorporating time-series data or real-time behavior could enable predictive churn scoring that evolves with customer interaction. Additionally, the model could be adapted to recommend personalized retention strategies, such as discounts or loyalty programs, based on individual risk profiles.

# Recommendations

Given the tabular neural network’s strong performance and scalability, it should be prioritized for deployment in production churn pipelines, particularly in cases where nonlinear interactions drive outcomes. However, random forests remain a strong baseline and could be used when interpretability or training speed is more critical. To improve business value, the model should be paired with a real-time dashboard and customer-level alerting for proactive retention.

# Implementation Plan

To operationalize the churn model, begin with batch inference on existing customer data using the trained tabular neural net. Integrate outputs into a dashboard for stakeholder review. After validation, deploy the model behind an API or scheduled job in the company’s data platform. Build alerting and customer lookup tools for business users. Regularly retrain the model on updated data to maintain accuracy and capture shifting patterns in customer behavior.

# Ethical Assessment

Customer churn modeling introduces several ethical considerations. It’s crucial to ensure the model does not discriminate against protected groups or use biased input features. For example, age-related features like “SeniorCitizen” must be handled cautiously and assessed for disparate impact. Additionally, transparency is key—customers affected by retention strategies deserve fair treatment and clear communication. Finally, the model must be used to support, not replace, human judgment, especially in sensitive customer interactions.

# References

* Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.). O'Reilly Media.
* Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer. https://doi.org/10.1007/978-1-4614-6849-3
* Molnar, C. (2022). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. https://christophm.github.io/interpretable-ml-book/
* Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
* Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* (Vol. 32).
* Howard, J., & Gugger, S. (2020). Deep Learning for Coders with fastai and PyTorch: AI Applications Without a PhD. O'Reilly Media.
* SHAP. (2023). *SHAP (SHapley Additive exPlanations)*. https://shap.readthedocs.io/en/latest/
* IBM Sample Data Sets. (n.d.). Telco Customer Churn Data Set. Retrieved from <https://www.ibm.com/communities/analytics/watson-analytics-blog/guide-to-sample-datasets/>

# Appendix

**Figure 1. ROC Curve – Random Forest vs Tabular Neural Network**Insert the matplotlib ROC curve plot comparing both models.

**Figure 2. Performance Table – Accuracy & AUC**Insert table showing model comparison side by side

**Figure 3. FastAI Learner Training Plot**Insert the training loss and validation loss plot from the Tabular Neural Net.

**Figure 4. Feature Importance Visualization (TNN)**Insert the fastai-derived feature importance chart

# Q&A

1. **What motivated the use of a tabular neural network instead of more traditional models?**  
   The TNN was chosen to explore whether modern deep learning architectures could outperform classical models like Random Forest in capturing nonlinearities in tabular customer churn data.
2. **Why was Random Forest selected as the baseline model?**  
   Random Forest is a strong and interpretable ensemble method that often performs well with minimal tuning and serves as a solid benchmark in classification problems.
3. **Did the model account for class imbalance in churn?**  
   Yes, class weighting was applied during training of the neural network to ensure the minority churn class received appropriate attention.
4. **How was model performance evaluated?**  
   Both accuracy and ROC AUC were used, with ROC AUC serving as the primary metric due to its ability to handle imbalanced data more robustly.
5. **How were categorical variables handled?**  
   Categoricals were embedded in the TNN model and one-hot encoded for the Random Forest, allowing each model to handle structured data optimally.
6. **What does it mean that only three features were heavily used by the TNN?**  
   It suggests that tenure, monthly charges, and senior citizen status were the most predictive. The neural net may have implicitly downweighted less useful features.
7. **Why did the TNN outperform the Random Forest despite using fewer features?**  
   Neural nets can model complex interactions and learn to ignore noisy or redundant input, focusing on the most salient patterns.
8. **Could this model be deployed in a real-world churn prevention system?**  
   Yes. It could be integrated into a customer service dashboard to flag high-risk customers for retention offers or outreach.
9. **What ethical concerns arise from churn prediction modeling?**  
   Bias, fairness, and transparency are primary concerns—especially regarding the use of sensitive demographic variables and the potential for profiling.
10. **What would be your next step if you had more time or data?**  
    Deploy the model in a simulated environment, explore longitudinal performance, and potentially integrate temporal dynamics or customer interaction data.