Explaining Churn Predictions Using SHAP in an Interactive Web App

By Alexander Marunycz

Overview of the Model

This project focuses on building an interpretable machine learning model application to predict customer churn based on various service and demographic features from a publicly available dataset of a fictional telecommunications company; the Telco Customer Churn dataset.

The model used for this project is an **XGBoost classifier**, a powerful gradient boosting algorithm known for its high performance on classification tasks and its ability to handle structured data effectively.

Data Preprocessing

The raw data required several cleaning steps before modeling including handling missing values for total charges when the customer was new to the Telco service as well as performing one-hot encoding on categorical variables. Normalization scaling was performed on all explanatory variables, but was not included in the final model because it showed no significant improvement in the model's performance and it would cause the explanation process to be further complicated.

Model Training with Hyperparameter Tuning

We employed a grid search with a **Stratified K-Fold Cross-Validation (n=10)** strategy to perform hyperparameter optimization on our XGBoost model.

The search was guided by the **ROC AUC score**, a robust metric for evaluating classifier performance on imbalanced datasets. After evaluating 50 randomized combinations, the best model achieved a **ROC AUC of 0.86** and **accuracy of 0.82** on the test set—indicative of strong discriminatory power.

This comprehensive approach ensured that the selected model not only performed well overall but also retained reliability in the presence of class imbalance.

Overview of the Interpretability Method

To interpret the model's behavior, **SHAP (SHapley Additive exPlanations)** was employed as the primary method of explanation. SHAP values attribute the prediction of an individual sample to its feature contributions based on a solid theoretical foundation in cooperative game theory. This method not only allows for global insights into feature importance but also provides **local interpretability**, showing exactly how each feature influences a specific prediction.

How SHAP Overcomes Feature Independence Assumptions

The method in which SHAP provides local interpretability is what makes it so powerful. It does so by overcoming feature independence assumptions that many traditional feature attribution methods (like permutation importance, LIME or simple partial dependence plots) assume — that is, they treat each feature as if it can be varied independently of the others. This assumption often leads to misleading interpretations, especially when features are interactive or are correlated (as is common in real-world datasets).

Here's how SHAP works and how it avoids the independence trap:

For a given prediction, SHAP computes the contribution of each feature by evaluating **its average marginal impact** across **all possible subsets of features**. In other words, for each feature, SHAP asks:

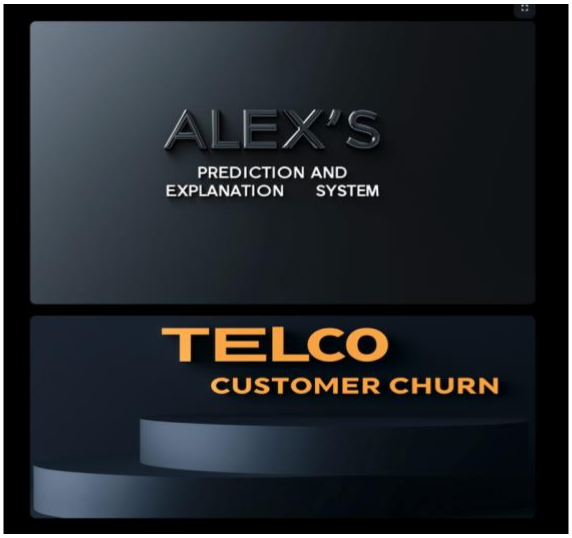
"How much does this feature add to the prediction, on average, when it's added to *every possible combination* of other features?"

This approach naturally **captures interactions and dependencies** between features because the feature is being added to many different *contexts* (subsets of other features), not in isolation.

This allows SHAP to **accurately account for correlations between features**, producing more realistic and reliable explanations.

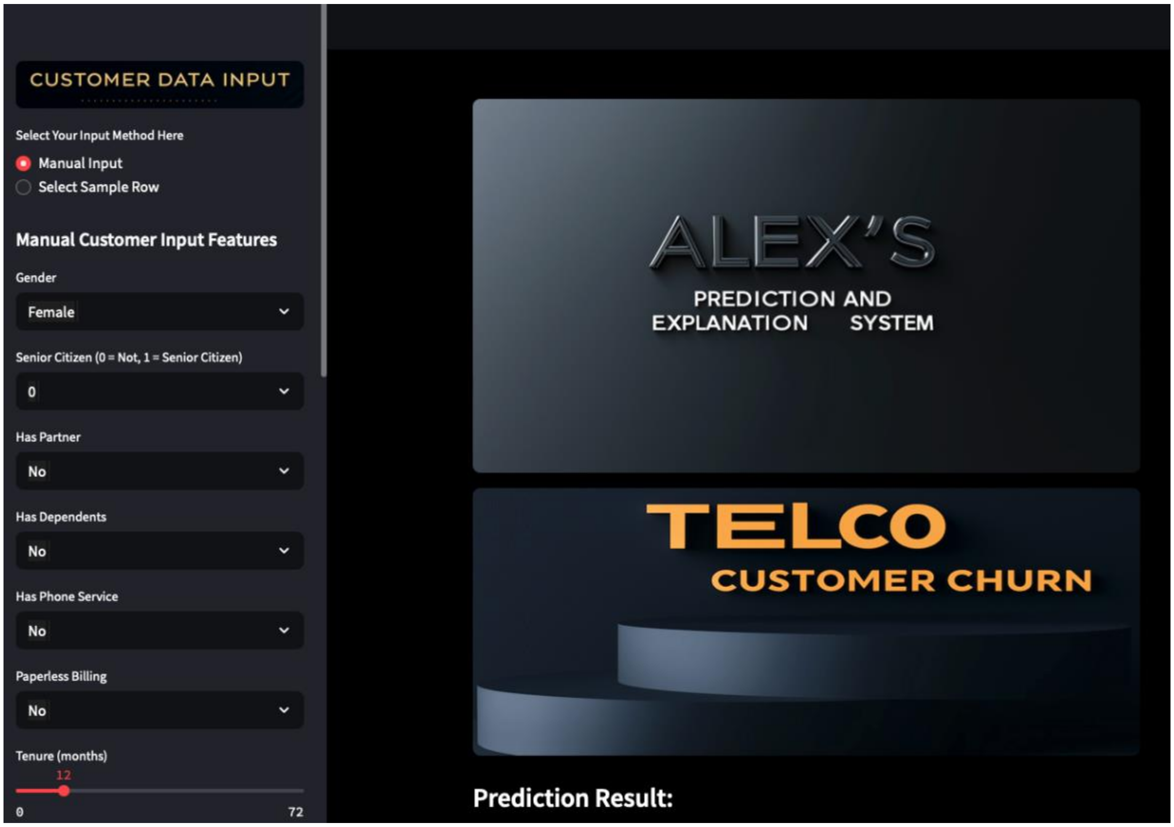
Without SHAP's approach, an explanation might wrongly attribute importance to a feature just because it appears to be useful when varied independently—even if it's not truly contributing once other correlated features are accounted for. SHAP avoids this trap by evaluating contributions **in context**.

Interactive Application

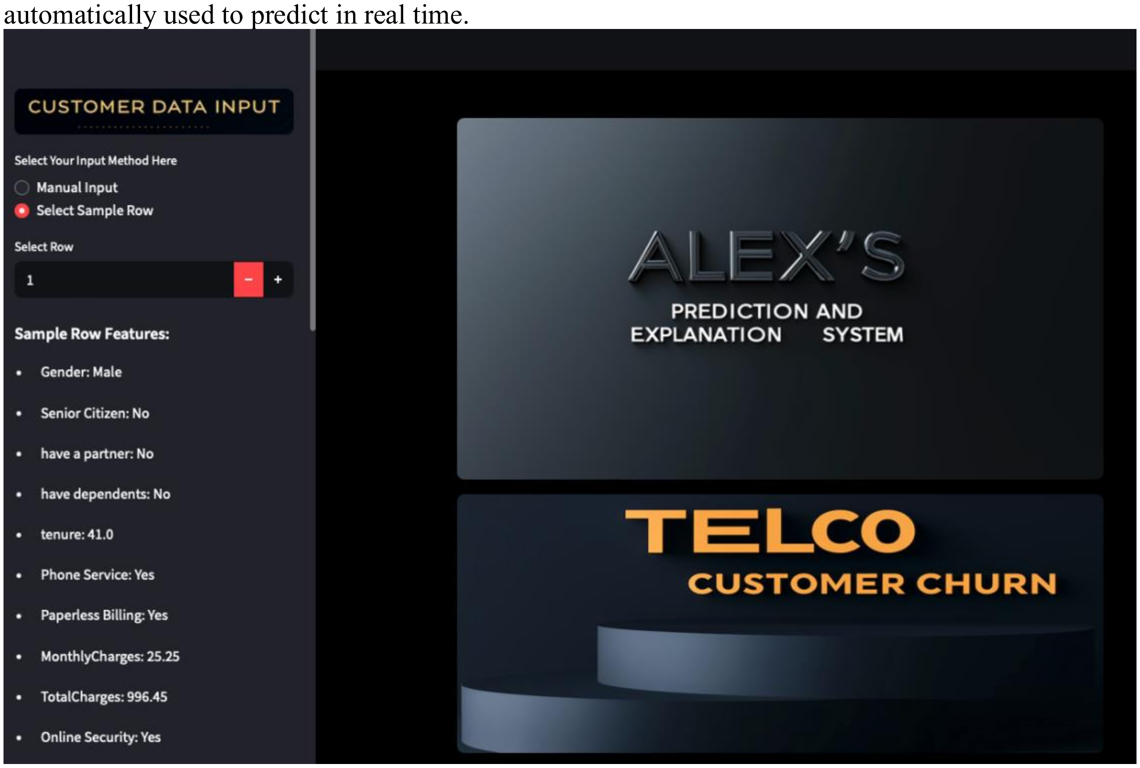


An interactive application was developed using **Streamlit**, allowing users to explore predictions and explanations dynamically. The app provides two primary input modes:

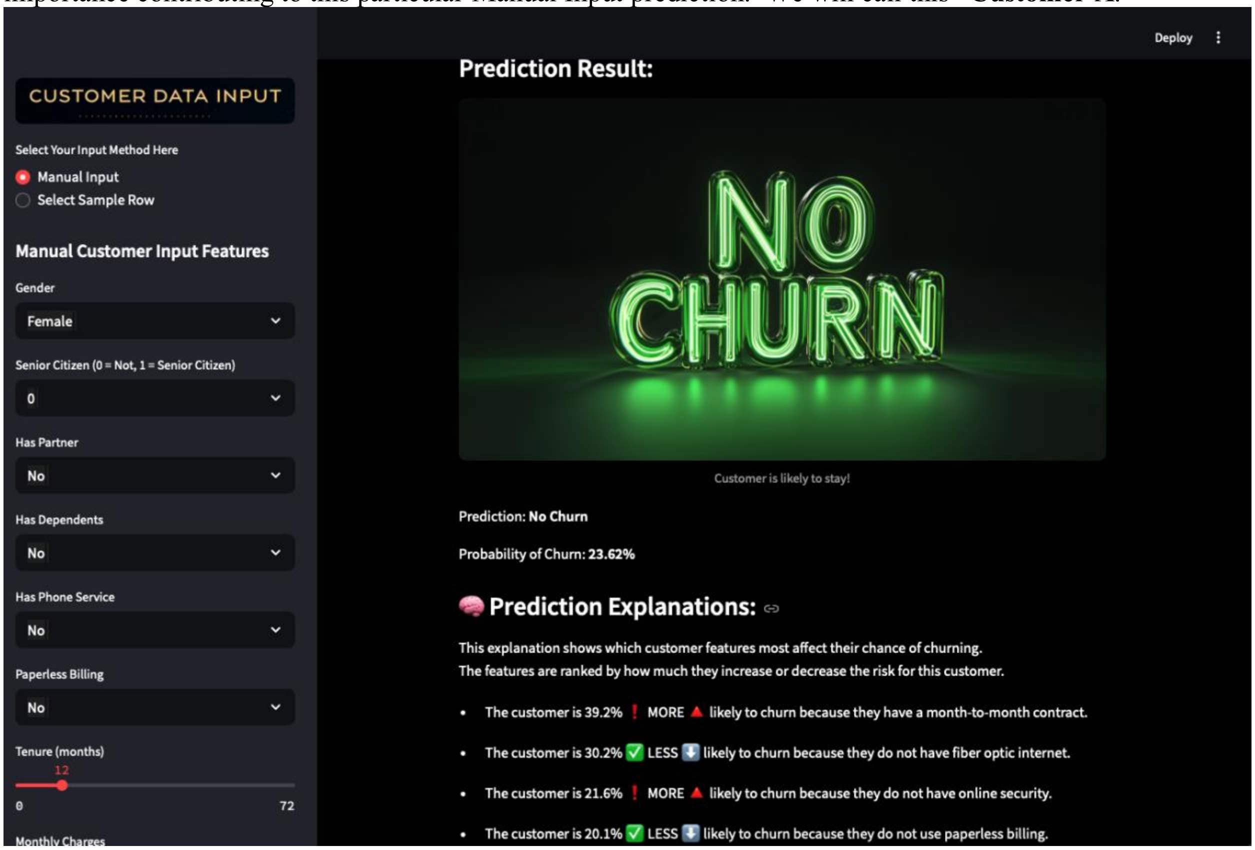
1. **Manual Feature Selection** – where users specify values for input features via drop down boxes and sliders, allowing for dynamic real-time prediction and explanation updates with every change to any given input feature. See below screenshot of manual input sidebar.



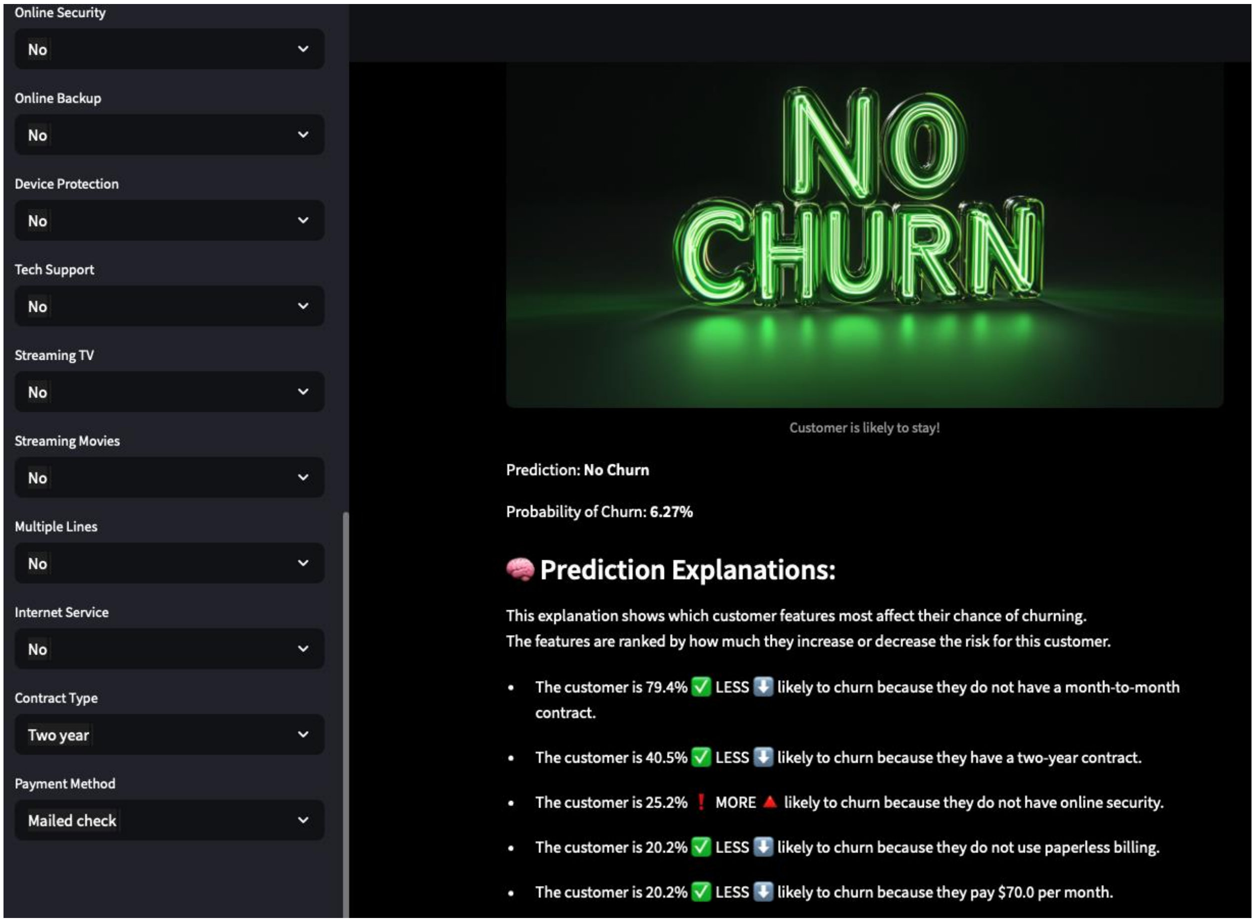
2. **Sample Row Selection** – where users choose a row from the existing dataset and receive predictions along with a breakdown of contributing features. See below screenshot of the Select Sample drop down box and its dynamic display showing the features pertaining to that record, which is automatically used to predict in real time.



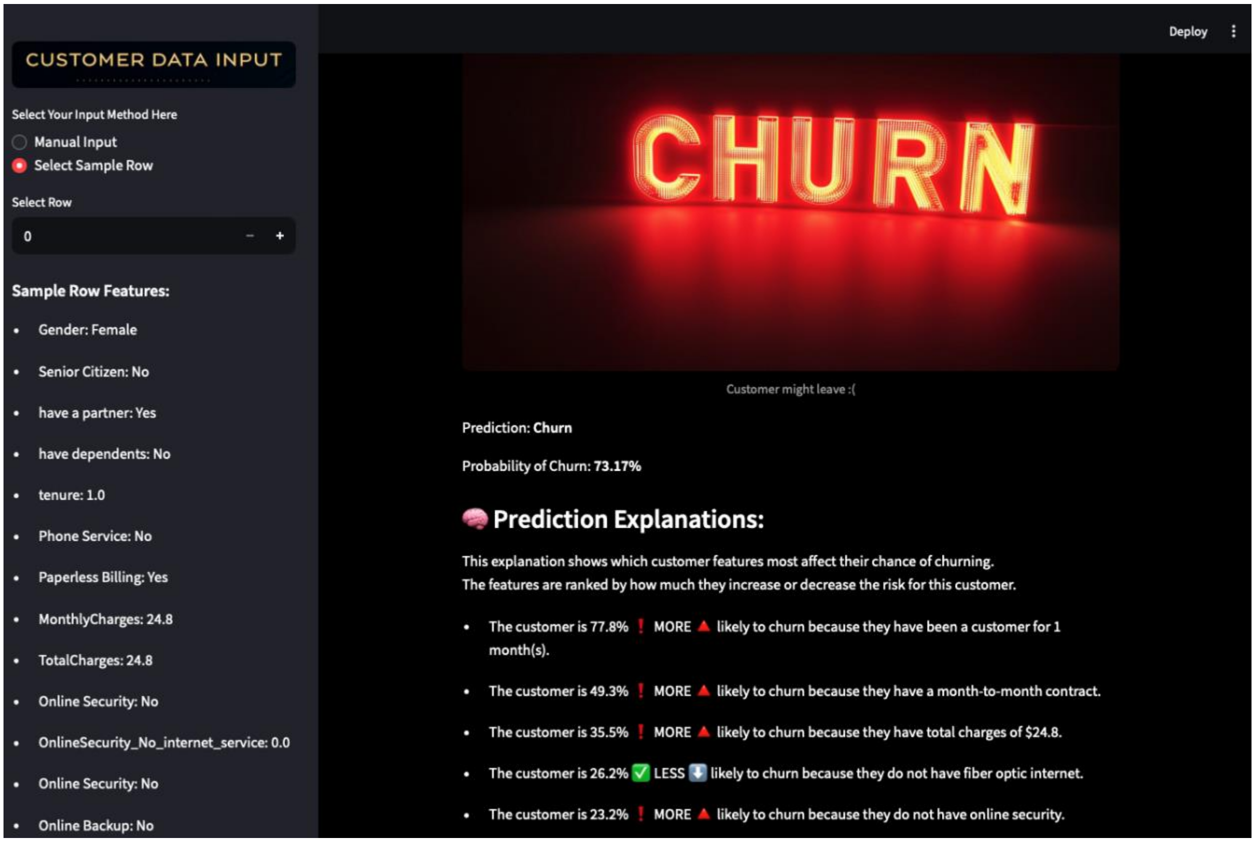
Below is a screenshot of the application in action, displaying a prediction of “No Churn” with a probability of churn of 23.82% along with its corresponding natural language explanations for the top variables of importance contributing to this particular Manual Input prediction. We will call this “**Customer A**.”



Below is another screenshot after manually changing two of the input features, Contract Type to “Two year” and Payment Method to “Mailed check.” Note how the probability of churn drastically reduces and the explanations change.

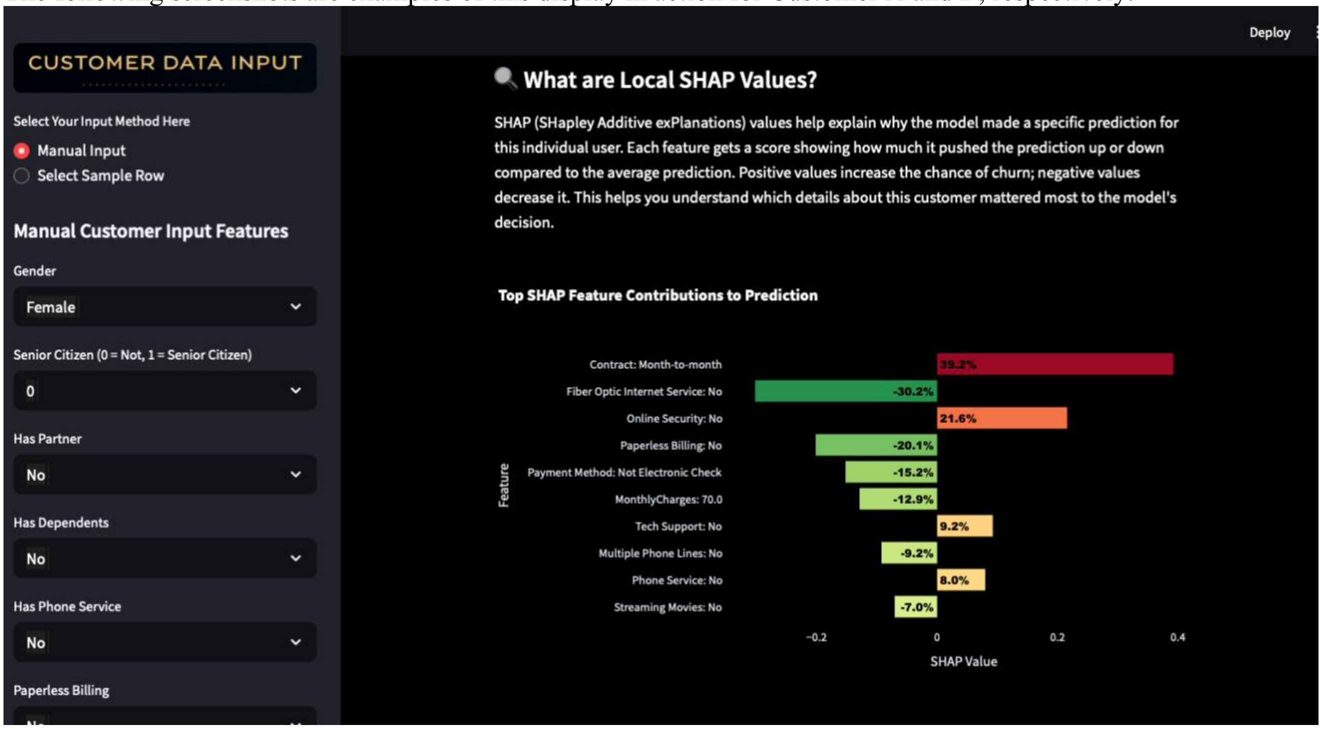


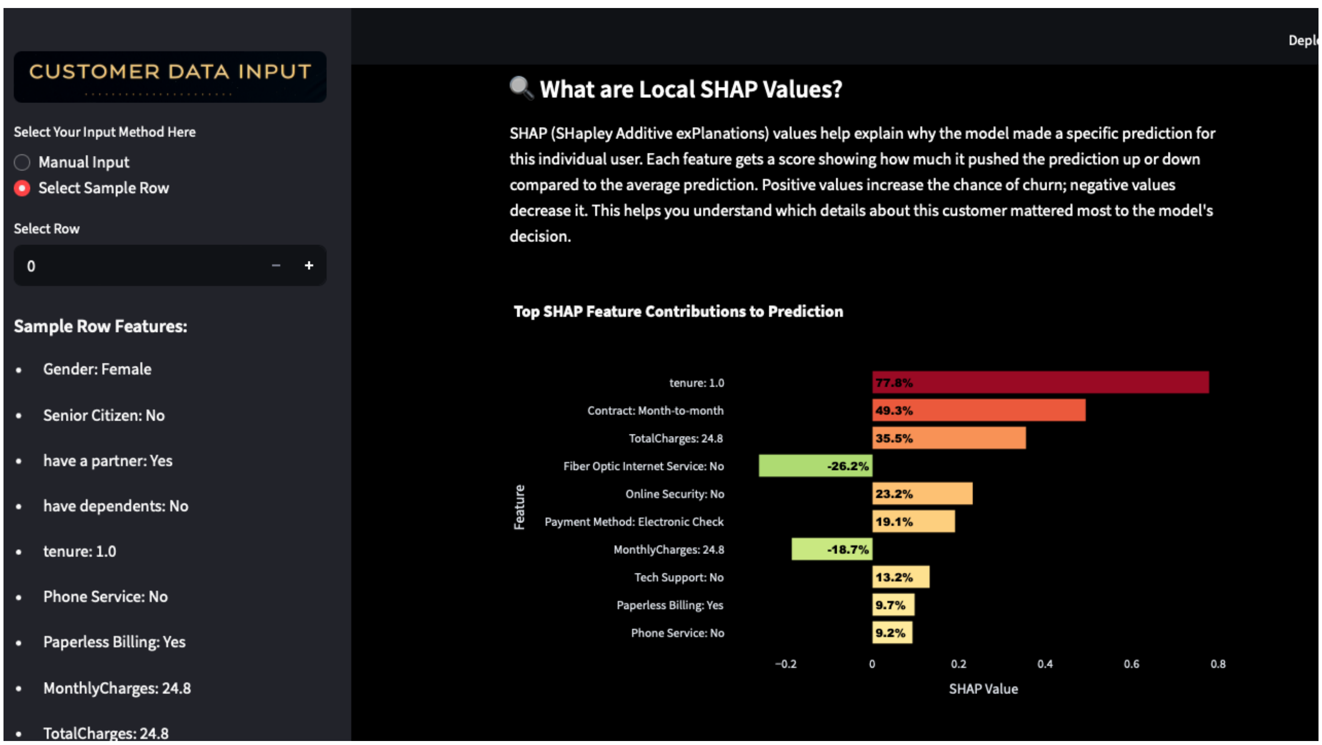
Now the following screenshot is of the first sample record in our sample dataset. We will call this "**Customer B**." Note how the prediction is "Churn" with a probability of 73.17% due to the explanations shown.



The next feature in the app is a dynamic and interactive graphical display (where the user can pan, zoom, show full screen, or download the image) that illustrates the top **Local SHAP features** and how they contribute to any given prediction.

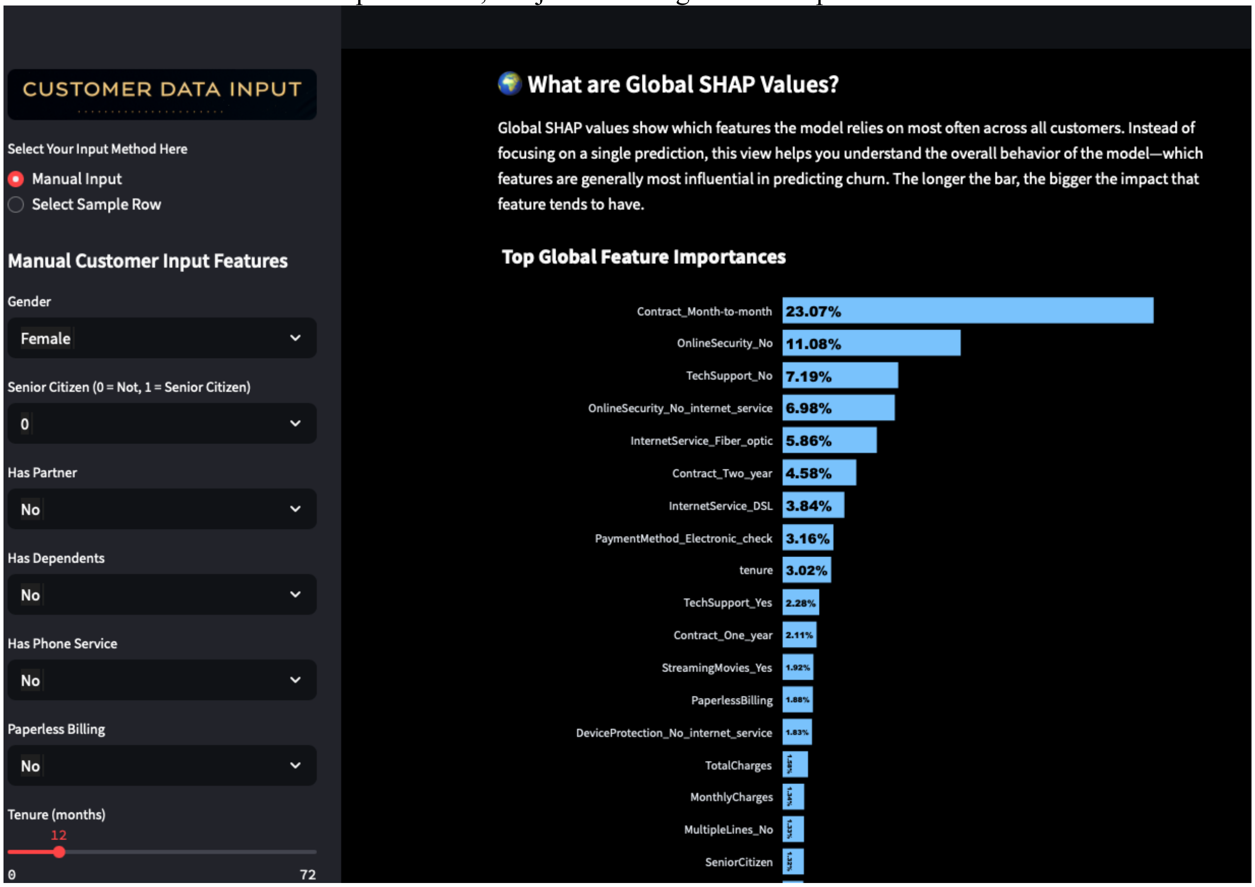
The following screenshots are examples of this display in action for Customer A and B, respectively.



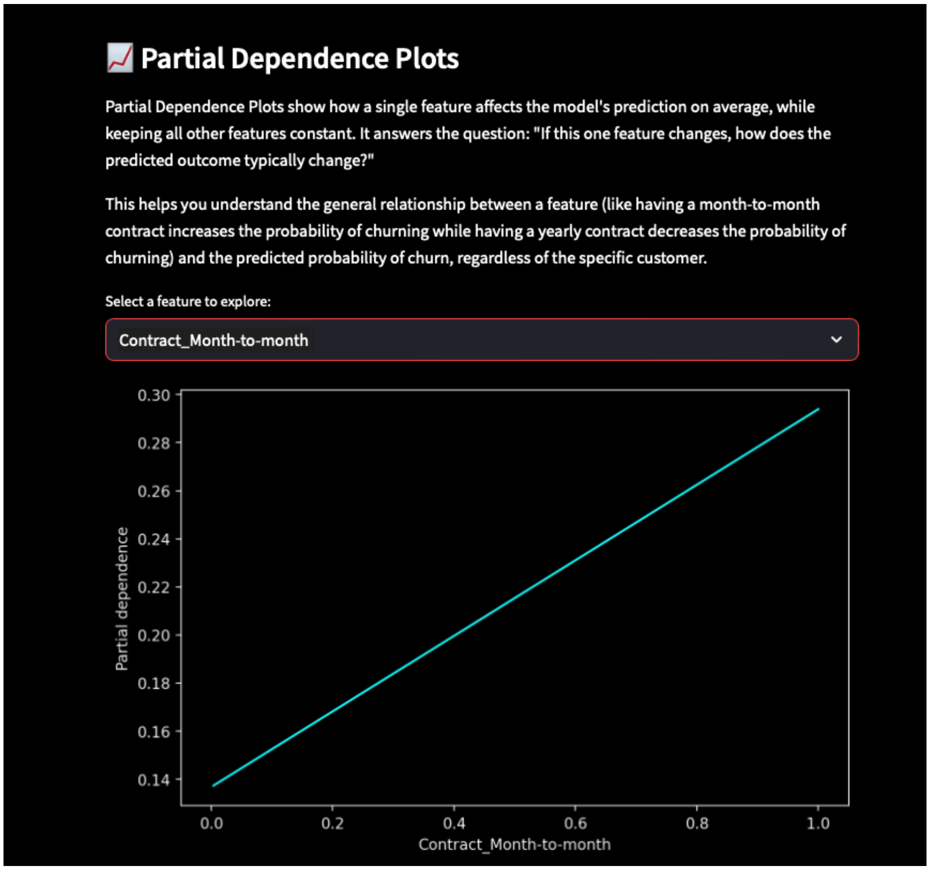


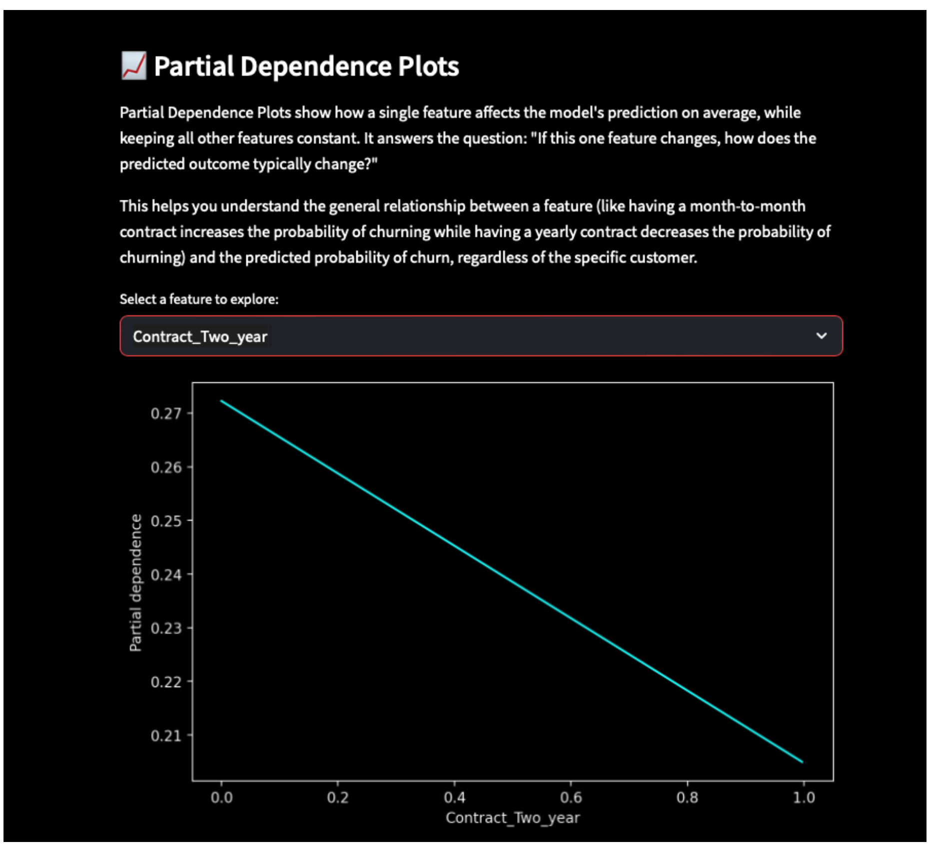
This interface empowers users to simulate "what-if" scenarios and understand not just what the model predicts—but why in a way that is intuitive to users of all technical backgrounds.

Next, there is a feature that displays the static Global SHAP values, so the user can see what features contribute to the overall model predictions, not just for a single record's prediction. See below screenshot.



Finally, the last feature of this application is the inclusion of dynamic Partial Dependence Plots, where the user can select any of the features via a drop down box and immediately see how the predicted outcome changes on average as this one feature changes while all other features are held constant. See below screenshots for the feature choices of month to month vs. two year contracts and how they effect the predictions in opposite directions.


📈 Partial Dependence Plots
Partial Dependence Plots show how a single feature affects the model's prediction on average, while keeping all other features constant. It answers the question: "If this one feature changes, how does the predicted outcome typically change?"
This helps you understand the general relationship between a feature (like having a month-to-month contract increases the probability of churning while having a yearly contract decreases the probability of churning) and the predicted probability of churn, regardless of the specific customer.
Select a feature to explore:
Contract_Month-to-month
Partial dependence
0.30
0.28
0.26
0.24
0.22
0.20
0.18
0.16
0.14
0.0
0.2
0.4
0.6
0.8
1.0
Contract_Month-to-month


📈 Partial Dependence Plots
Partial Dependence Plots show how a single feature affects the model's prediction on average, while keeping all other features constant. It answers the question: "If this one feature changes, how does the predicted outcome typically change?"
This helps you understand the general relationship between a feature (like having a month-to-month contract increases the probability of churning while having a yearly contract decreases the probability of churning) and the predicted probability of churn, regardless of the specific customer.
Select a feature to explore:
Contract_Two_year
Partial dependence
0.27
0.26
0.25
0.24
0.23
0.22
0.21
0.0
0.2
0.4
0.6
0.8
1.0
Contract_Two_year

Insights Gained from Explanations

By applying SHAP to individual predictions, we observed:

- **MonthlyCharges**, **Contract type**, and **Tenure** consistently ranked among the most influential features in determining churn likelihood.
- Customers with **month-to-month contracts** and **electronic payment methods** were far more likely to churn, while longer-term contracts acted as stabilizing factors.
- SHAP's local explanations highlighted how **combinations of factors** (e.g., a senior citizen with no tech support and high charges) could compound the churn risk, even if each individual feature had only moderate influence.

These insights offer business value: they suggest specific customer profiles for targeted retention efforts, such as offering better deals for customers with month-to-month contracts or improving tech support options.

Discussion on XAI (Explainable AI)

Why SHAP Was Chosen

SHAP was selected because:

- It offers **both global and local interpretability**.
- It's **model-agnostic** and works seamlessly with tree-based models like Forest based models such as XGBoost.
- It provides **visually intuitive plots** that communicate well to technical and non-technical audiences alike.

Compared to other methods like LIME or simple feature importance plots, SHAP's theoretical grounding, ability to calculate importance without feature independence assumptions and consistency make it a reliable choice for high-stakes use cases like churn prediction.

The Importance of Explainable AI

In real-world scenarios—especially in industries like finance, healthcare, and telecom—**model decisions must be transparent and justifiable**. Explainable AI (XAI) serves this need by:

- Building **trust** with stakeholders and end-users.
- Helping **identify and mitigate bias**.
- Allowing **data-driven policy changes** based on model behavior.

For instance, insights from this churn model could inform retention strategy, pricing policy, or customer communication channels.

Limitations of SHAP

Despite its strengths, SHAP has a few limitations:

- **Computationally intensive**, especially for large datasets and complex models.
- **Interpretation overload**: Users may struggle to understand explanations with too many features unless guided.

These limitations were not an issue for this project due to the small dimensionality of this particular dataset (less than 10k records and two dozen variables).

Future work could involve further dimensionality reduction, ensemble learning, hierarchical explanations, or a translation of the data and findings to natural language so the user of the app can chat with it via Retrieval Augmented Generation (RAG) to further simplify the explanations for the user.

Conclusion

This project demonstrates the power of pairing predictive modeling with explainable AI tools like SHAP in an intuitive interface. Users can confidently interact with the model, understand its reasoning, and extract actionable insights—making the solution both **technically sound** and **practically valuable**. Explainable AI is not just a tool for transparency but a bridge between data science and informed decision-making.