SHAP feature analysis

Greyson Henry

1 SHAP Feature Analysis

1.1 Introduction

Shapley Additive Explanations (SHAP) is a game-theoretic approach to explaining machine learning models. It evaluates each input feature, giving them a composite score that represents how much they've contributed to a prediction. Specifically, a SHAP value tells you how much a feature (say, someone's EstimatedSalary) contributes to moving away a prediction from the model's average. SHAP values should be read as the same unit of the target feature they're being used for (a SHAP value of +5000 for Balance by Credit Score would indicate that this person's credit score has contributed $5000 to their balance, relative to the model average).

SHAP provides an alternative to MLE/AIC/BIC as a way to simplify models and improve interpretability. Unlike the model selection algorithms we've covered, SHAP is model agnostic and doesn't require retraining. Like ANOVA, SHAP is resilient to categorical, quantitative discrete, and quantitative continuous predictors.

See official documentation here.

1.2 Motivation

Here, we use SHAP as a tool to interpret the outcomes of an XGBoost modeling of the data. XGBoost (Extreme Gradient Boosting) is a machine learning library that relies on decision trees, and is well-suited to regression tasks. It predicts outcomes based on input features; SHAP explains the role those features played post-hoc.

1.3 Imports

SHAP analysis in R requires the following packages:

- XGBoost (cite)
- SHAP for XGBoost (cite)

1.4 Loading Dataset

Our SHAP analysis does not drop any data points. However, it does not consider the features RowNumber, CustomerId, and Surname—the former two are metadata, and the latter is an over-sparse categorical variable that also has no obvious connection to our target feature in summary statistics. Other than that, the data remains vanilla.

1.5 Splitting & Training

Our Balance category contains an extremely disproportionate amount of zeroes.

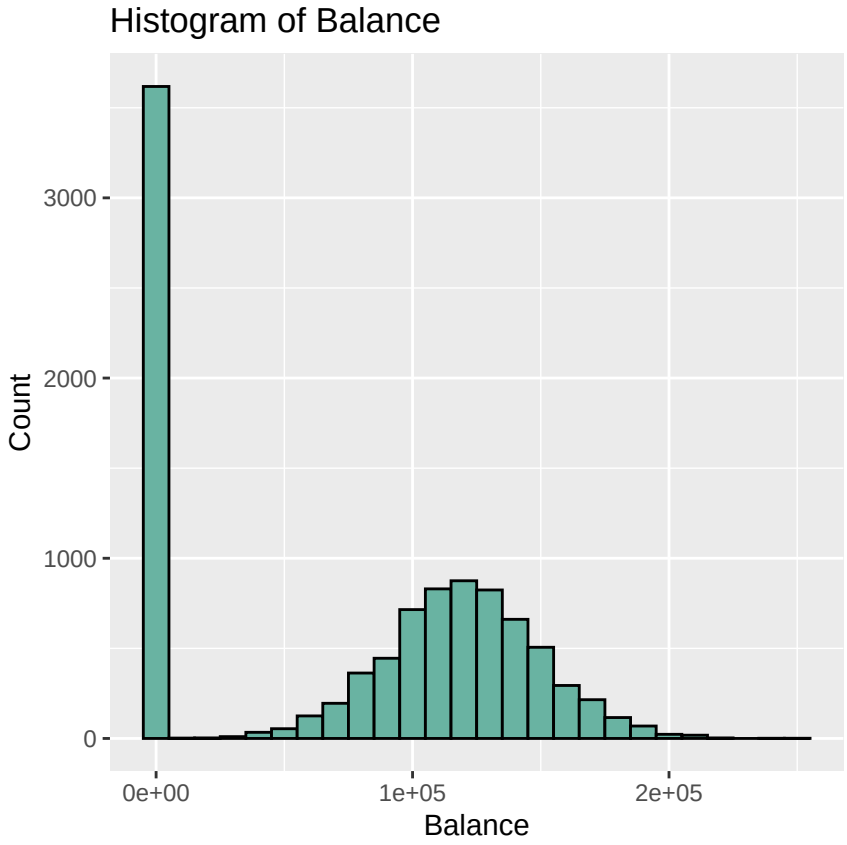

Figure 1: Histogram of Balance values

In order to account for zero-balances without artificially clustering the XGBoost predictions around 0, we have elected to train three XGBoost models. The first model is a binary classifier, which predicts whether a balance is 0 or not and outputs a probability. The second model is a regression model which predicts balance (specifically, $\log(Balance + 1)$) and is trained on only nonzero balances. The third model is a combination of the other two; it predicts *expected* balance, defined as the probability of a nonzero balance and the predicted balance. It can be thought of as an approximation of the entire pipeline.

1.6 Classification Model

Our classification model uses the following hyperparameters:

- Binary logistic classification
- Log loss evaluation metric
- Max depth 6
- eta (learning rate) 0.05 (slow rate, anti-overfitting)
- 200 epochs of training

1.7 Classification Eval

We can evaluate the performance of our binary classifier by looking at its ROC (Receiver Operating Characteristic) curve.

```
Area under the curve: 0.8517
```

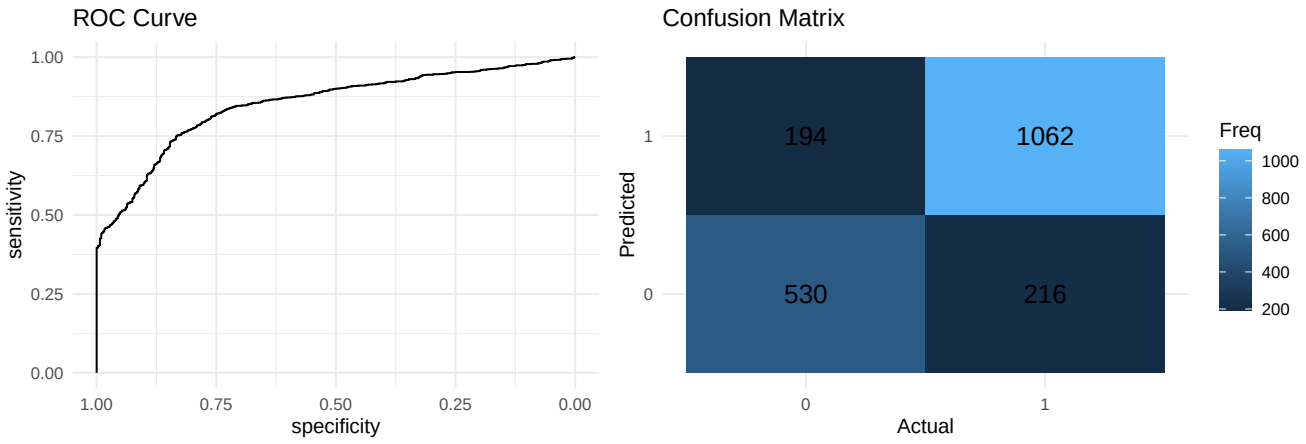

Figure 2: ROC curve and confusion matrix for binary classifier

Our Area Under Curve (AUC) is 0.8584, meaning that, 85.84% of the time, our model will rank a random positive example higher than a random negative. This is considered "excellent".

1.8 Regression Model

Our regression model uses the following hyperparameters:

- Regression with squared loss objective
- RMSE evaluation metric
- Max depth 8
- eta (learning rate) 0.03 (slow, to avoid overfitting)
- 800 epochs of training

We apply a log transformation here to standardize the distribution of our balance data.

1.9 Regression Eval

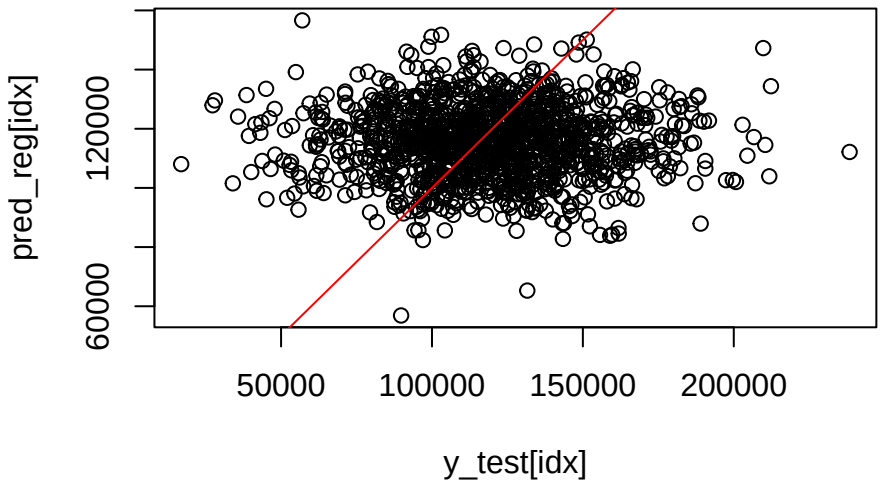

Figure 3: Regression model vs testing data

Clearly, the regression model struggles to predict zero-removed test data. This is explainable by two factors. One, this model hasn't trained on any zero-balances, which comprise a great deal of the actual data (so it ends up being a sparse feature landscape). Two, the relationships between input features and predicted Balance is highly nonlinear, which makes a simple linear model a poor choice; this will be explored further in our SHAP analysis.

1.10 Composite Model Eval

Our third model combines the previous two, providing a "surrogate" approximation of Balance. It represents the product of our other model outputs: $P(Balance > 0) \times \text{Model}(Balance|Balance > 0)$. It can be read as an "expected value" mapping between the two.

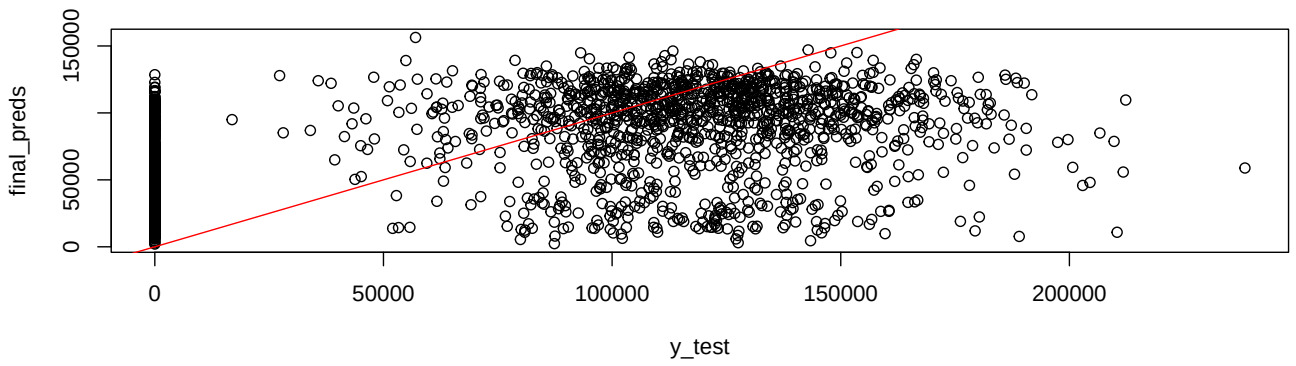

Figure 4: Surrogate model vs testing data

Table 1: Model Performance Metrics (Composite)

Metric	Value
RMSE	52334.24
MAE	41189.53
Correlation	0.5451

This model has poor results, for much the same reasons as our regression model. However, our moderate correlation ≈ 0.5451 suggests that it was able to capture meaningful structure within the data, which is the crucial quality for SHAP. Essentially, while our XGBoost models were generally unable to make accurate predictions, they were able to distinguish patterns between input features and the output, which is sufficient for SHAP analysis.

1.11 SHAP Framing

SHAP is a method for explaining individual model predictions by assigning contributions to each feature. For a given observation, each feature is assigned a SHAP value that represents how much it pushes the model's prediction above or below a baseline value (usually the average prediction across training data). SHAP values should be read as the same units as the model output, and sum with the baseline to reconstruct that prediction. For example, in a binary classification model, SHAP values represent additive contributions towards the odds of the positive class; if a model has a baseline of 50% and predicts a given observation has a 20% chance of falling into the positive class, that observation's SHAP values might look like (Age -10%, Salary -20%, Exited +5%, Gender -5%… Baseline +50% = 20%) and so on.

1.12 Summary SHAP for Binary Classification

Here, SHAP values represent "how much a feature shifts the log-odds of having a positive Balance"; they are multipliers that, all else being equal, mean this feature multiplies the model output by e^{SHAP}.

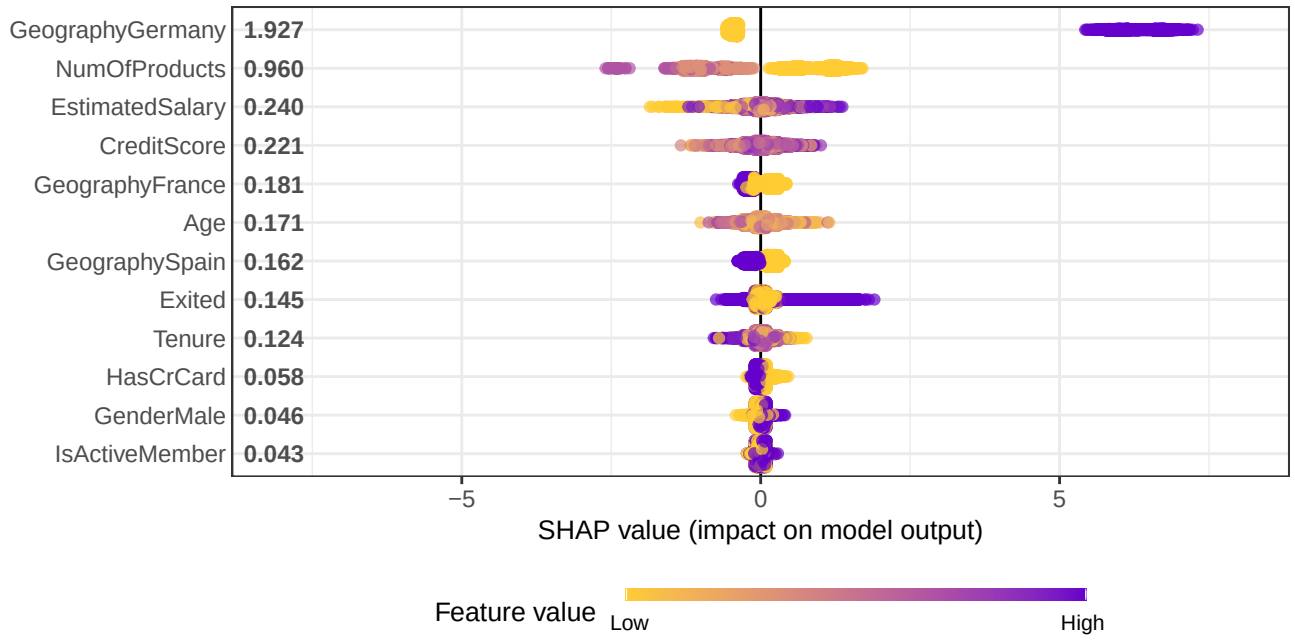

Figure 5: SHAP Summary Plot for Binary Classifier

On average, the GeographyGermany feature has a disproportionate weight in explaining empty balance predictions ($e^{1.927} \approx 6.87$). The second most impactful feature on average was NumOfProducts, itself carrying an outlying SHAP value of 0.96 ($e^{0.96} \approx 2.61$) The features EstimatedSalary and CreditScore comprise the rest of the top 4. We will break down per-feature trends later on.

1.13 Summary SHAP for Regression

Here, SHAP values represent log-transformed multipliers on a balance. For example, a SHAP value of 0.046, or ~1.05 when converted, would represent that instance of that feature multiplying balance by 1.05x. Recall, this model was trained on a subset of data which didn't include zero-balances; its purpose is to explain contributions to balance among positive balances.

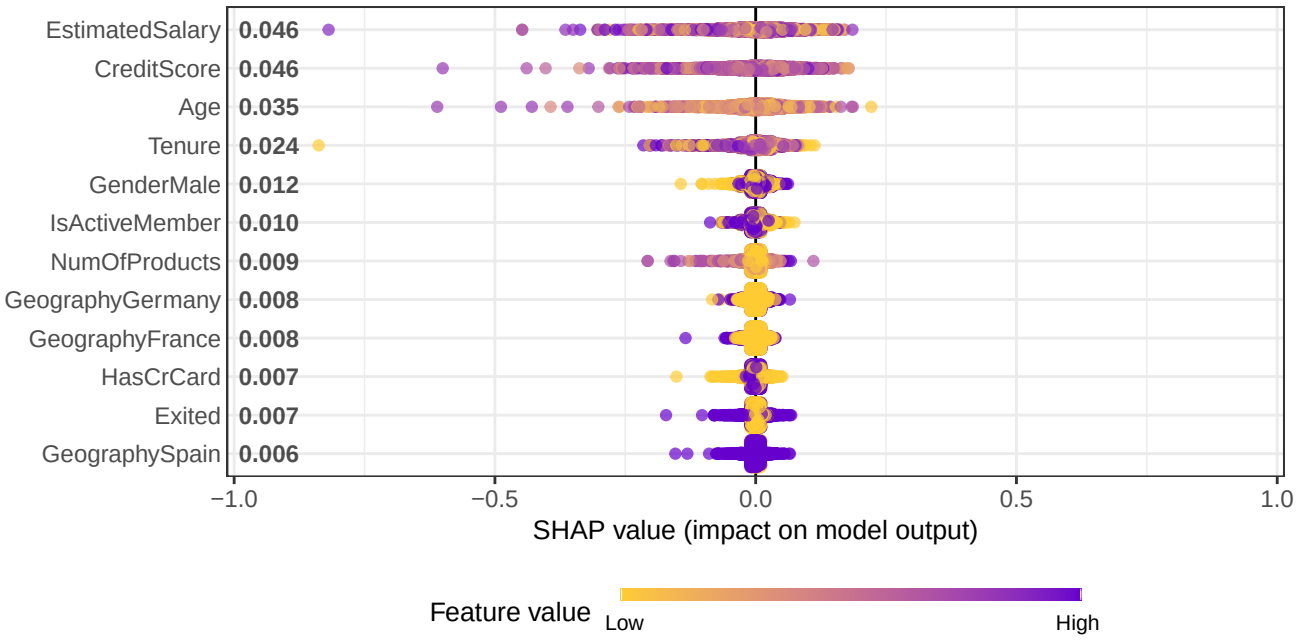

Figure 6: SHAP Summary Plot for Regression

Compared to our binary classifier, these SHAP values are much milder. EstimatedSalary and CreditScore share an average SHAP value of 0.046, with Age and Tenure trailing behind.

1.14 Summary SHAP for Composite Model

Here, SHAP values represent log-transformed multipliers, much like our regression model.

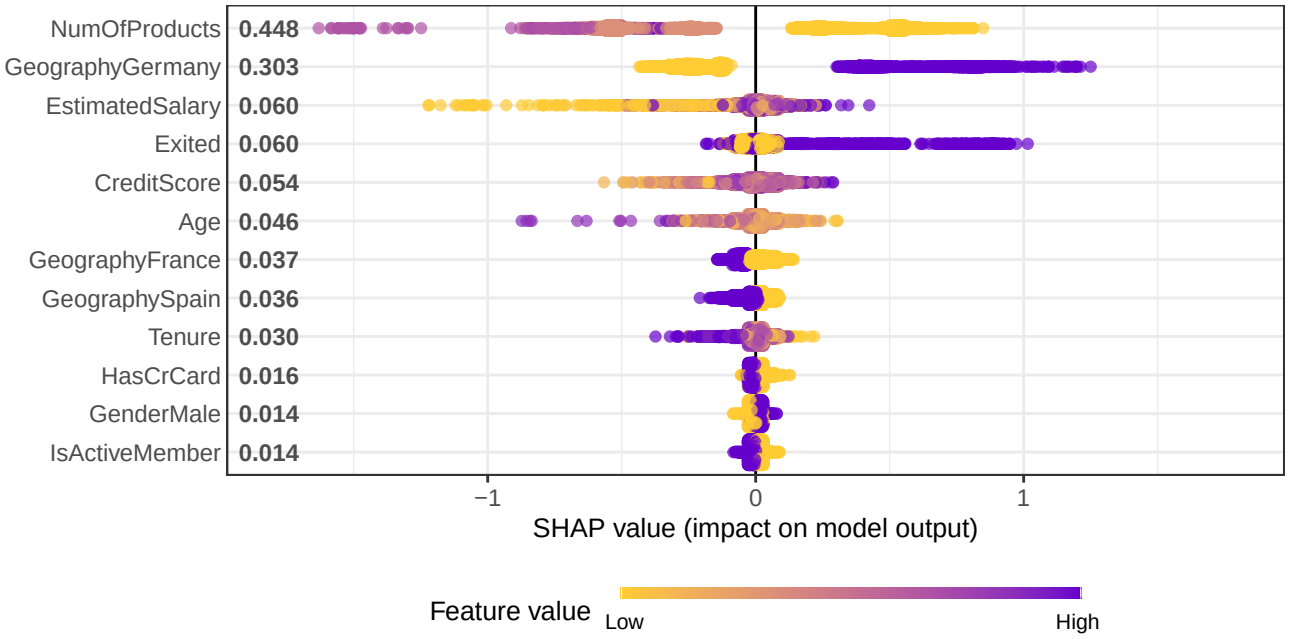

Figure 7: SHAP Summary Plot for Composite Model

NumOfProducts and GeographyGermany have disproportionately high SHAP values on average, meaning they're highly explanatory of positive balance.

1.15 Dependence Plots

SHAP summary plots communicate, on average, how much a feature explains an output. However, a necessary second step is to check trends within a feature to see what ranges of values might be more or less explanatory of a prediction. The tool for doing this is dependence plots: they display the relationship between particular feature values and SHAP values.

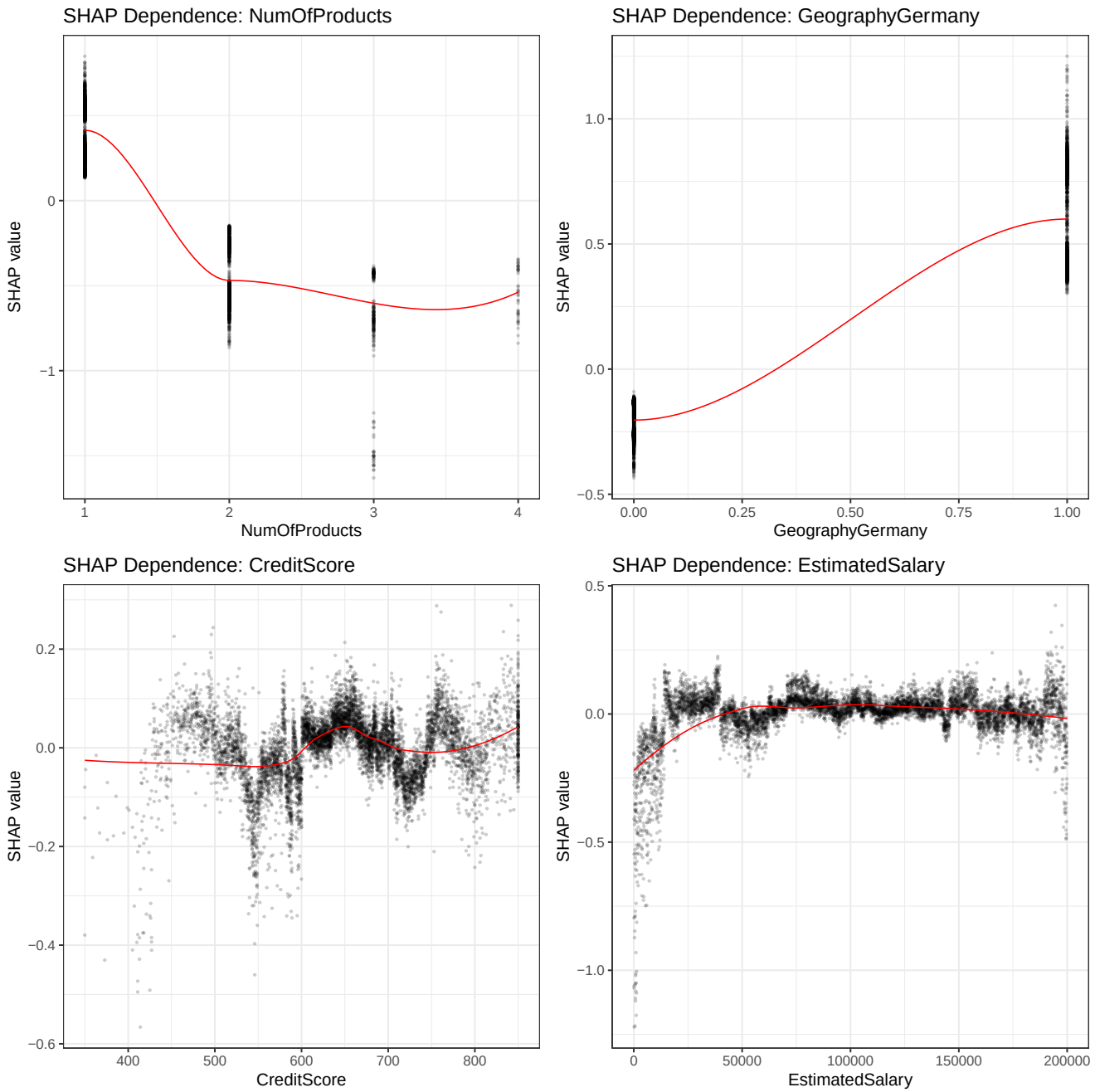

Figure 8: SHAP Dependence Plots for 4 Highest Avg Predictors

For NumOfProducts, we can see that observations with 1 product have higher minimum, maximum, and average SHAP values compared to the other categories, showing that owning exactly 1 product is highly explanatory of having more balance. The other categories (2, 3, 4) have wider ranges and have negative maxima, meaning they're explanatory of having less balance.

For GeographyGermany, the category 0 (not German) has a tight, negative range, and is explanatory of less balance relative to the baseline; likewise, category 1 (yes German) has a positive, high-magnitude range, explaining higher balance relative to the baseline.

For CreditScore, moderate scores between 500 and 600 are somewhat explanatory of lower balance, while moderate-to-good scores between 600 and 700 are somewhat explanatory of higher balance. Capped credit scores have a wide range.

For EstimatedSalary, low values are highly explanatory of low balance. Most other EstimatedSalary observations are clustered around 0, with a slightly expanded range as values approach 200,000.

1.16 Validating SHAP Values

As a final sanity check, we need to verify how well the SHAP patterns we've detected generalize to new data.

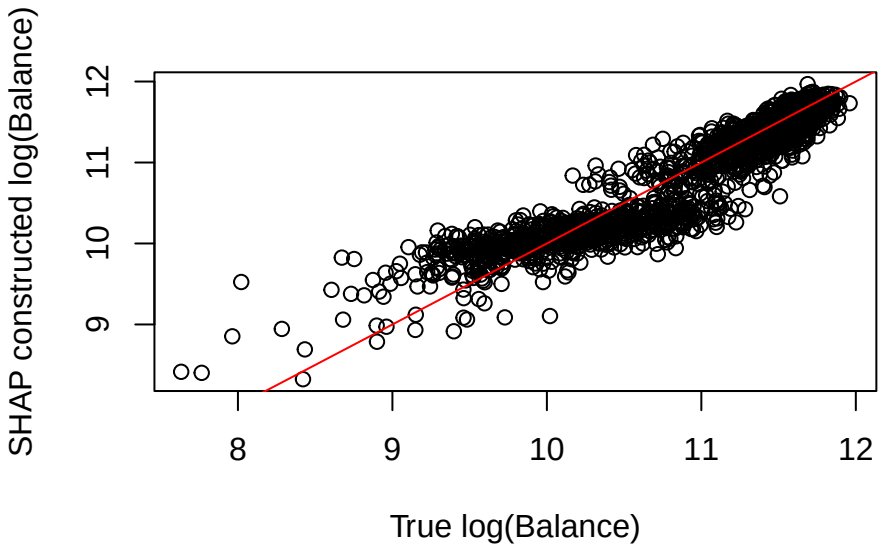

Figure 9: SHAP Dependence Plots for 4 Highest Avg Predictors

Table 2: Model Performance Metrics (Predicting SHAP)

Metric	Value
RMSE	0.2571
MAE	0.184204
Correlation	0.9424

With strong RMSE, MAE, and correlation, our SHAP patterns are extremely generalizable to test data.