Rogue-Validator Hat

Group 5

Vishaal Sai Devisetty

Michael Whacha

Ositadinma Ekwonu

Rogue-Validator Hat

Agenda

1. Rogue Hat - Modelling
2. Flaws Introduced
3. Validator Hat
4. Findings
5. Wrap Up

ROGUE HAT

Modelling

Filtering Data

Loan Status Filtering	fully_paid = 1 indicates a loan that was fully repaid. fully_paid = 0 indicates a loan that was charged off (defaulted).
Timeframe Filtering	(August 2012)
Missing Value Threshold	60%
Sampling	Stratified sampling technique

Modelling

Feature Engineering

Numerical Predictors	loan_amnt, fico_range_low, fico_range_high, avg_cur_bal,dt
Categorical Predictors	addr_state, purpose, application_type

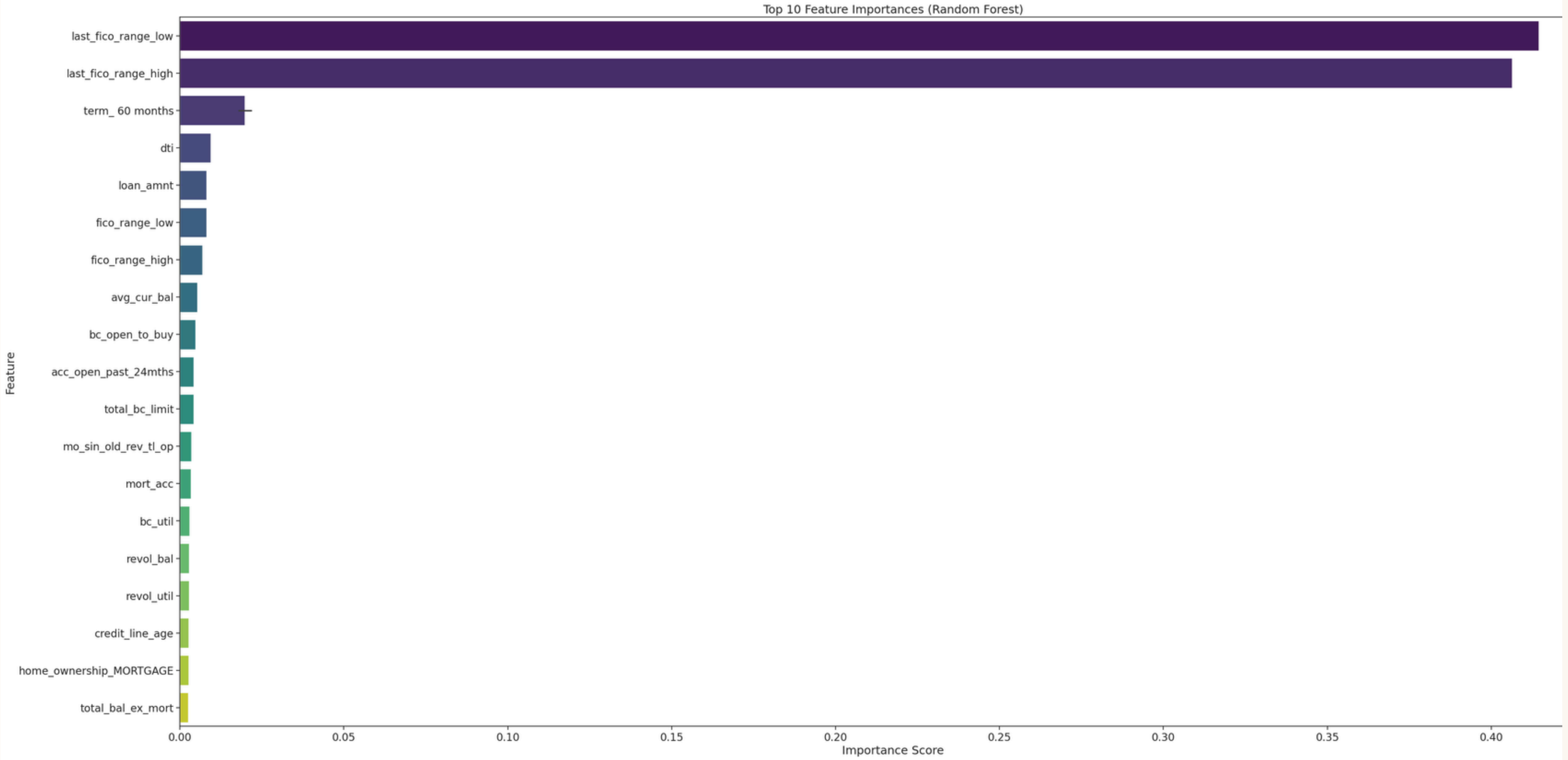
Handling Missing Values

Imputation,Categorical feature handling, Custom Processing

Transformations

One-hot encoding, Additional Features

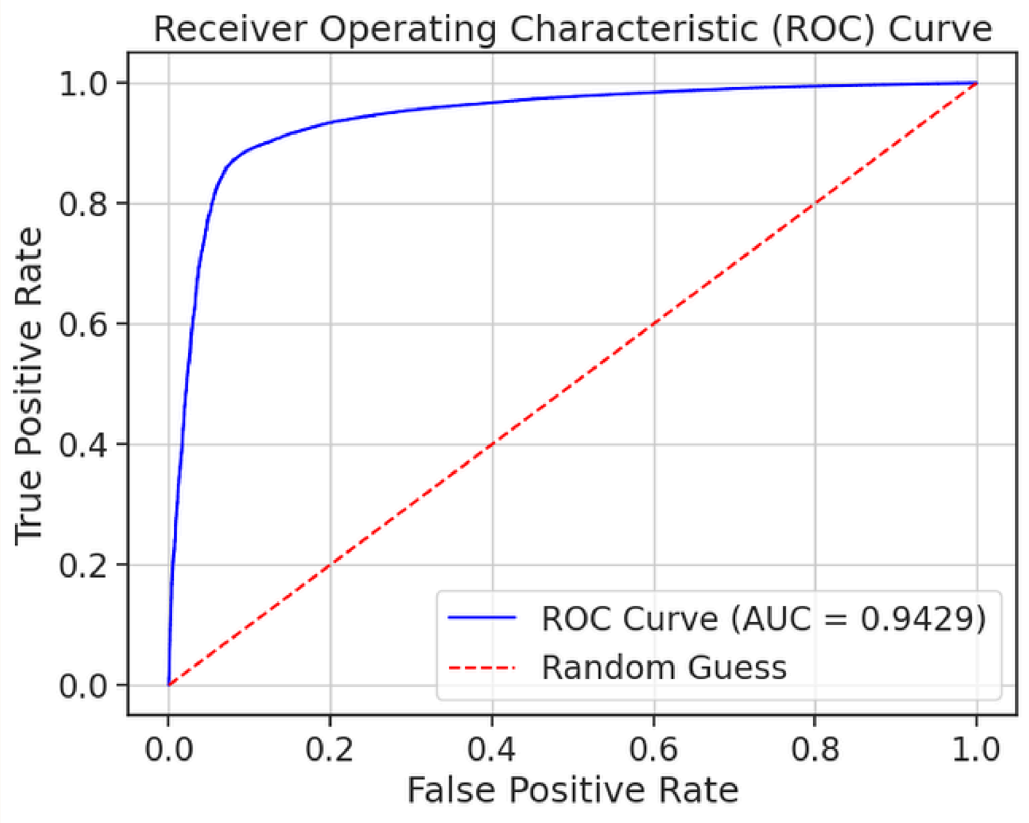
Random Forest



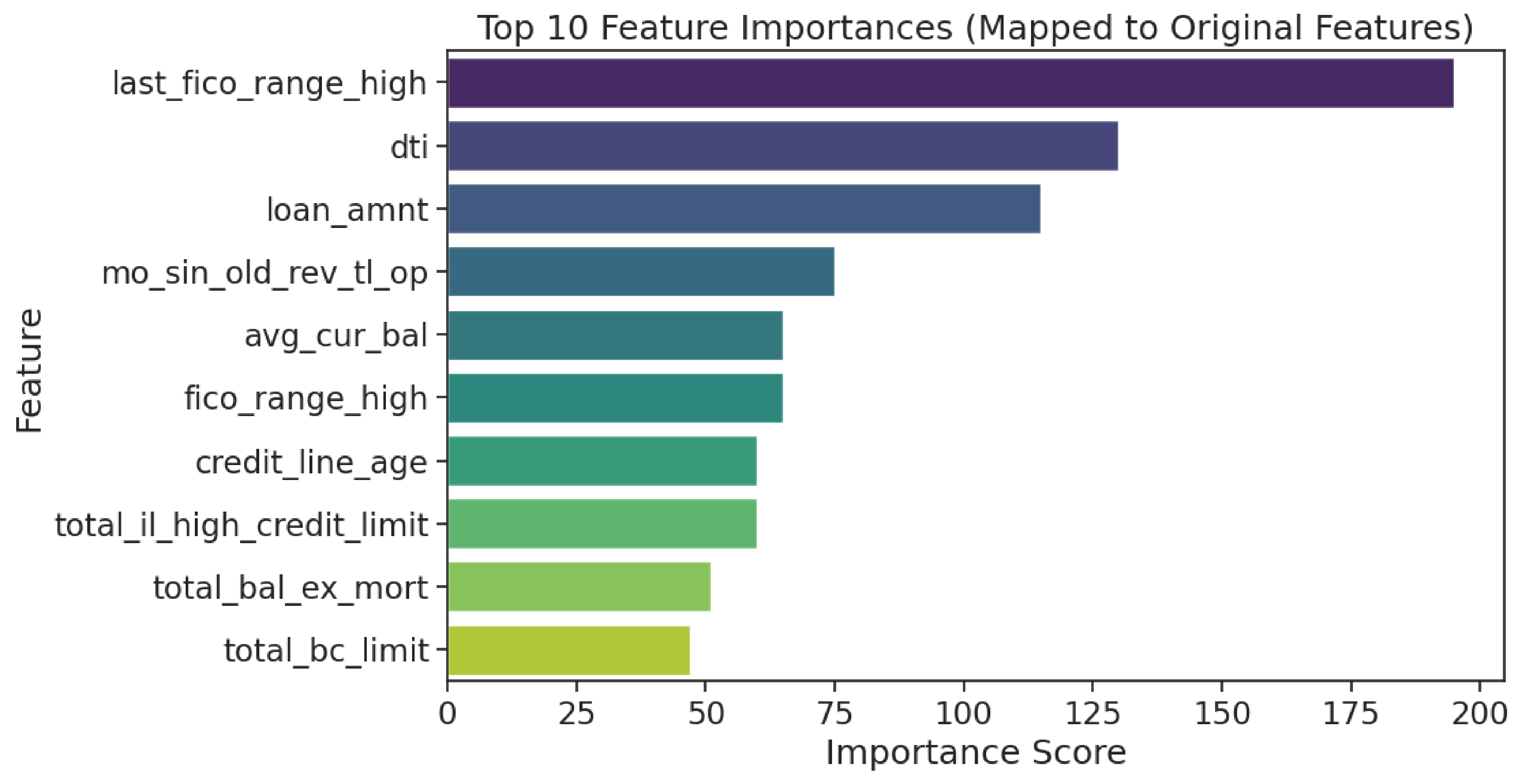
Random Forest Performance:
Accuracy: 0.9066
ROC AUC: 0.9429

Classification Report:

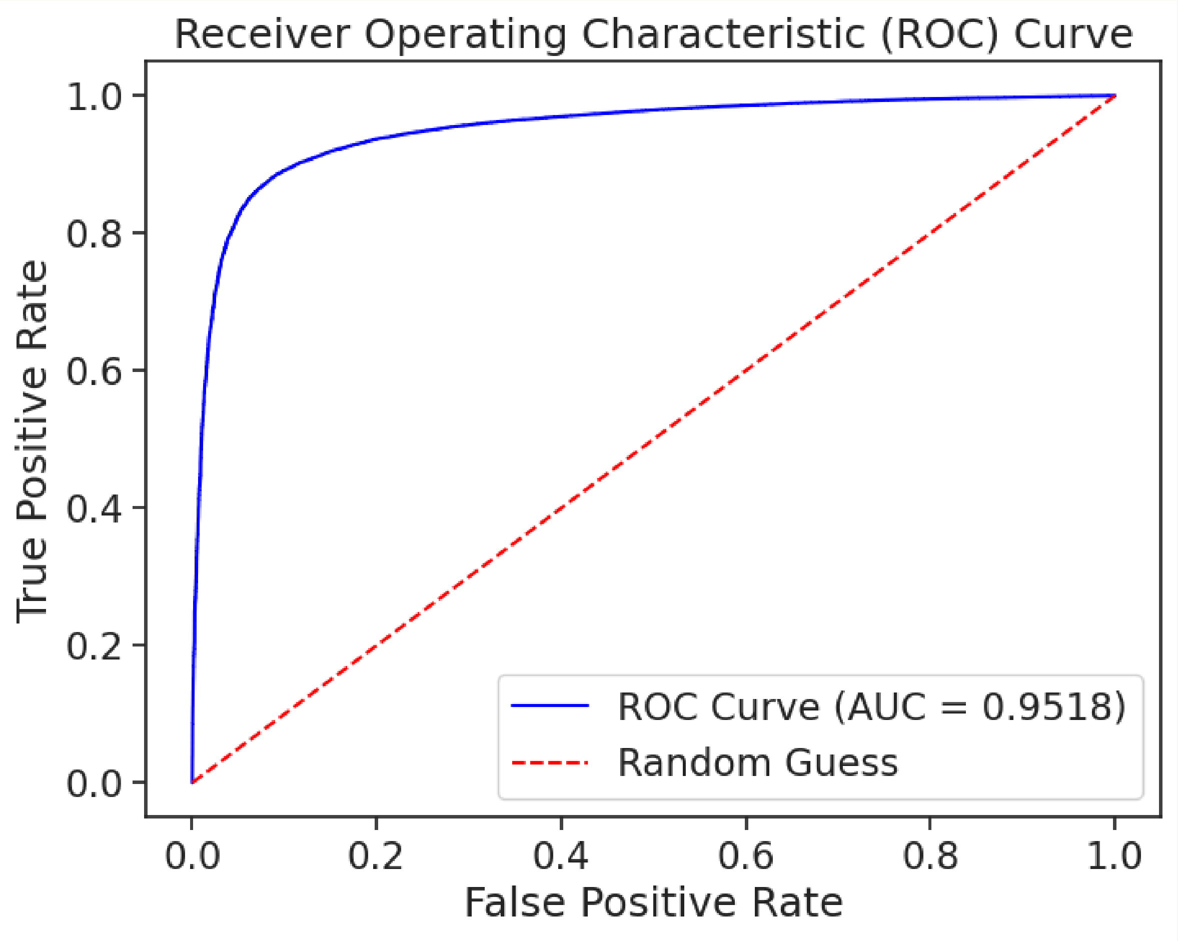
	precision	recall	f1-score	support
0	0.77	0.77	0.77	12194
1	0.94	0.94	0.94	47806
accuracy			0.91	60000
macro avg	0.86	0.86	0.86	60000
weighted avg	0.91	0.91	0.91	60000



XGBoost




```
XGBoost Performance:
Accuracy: 0.9093
ROC AUC: 0.9518

Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.79      0.78     12194
           1       0.95      0.94      0.94     47806

    accuracy                           0.91     60000
   macro avg       0.86      0.87      0.86     60000
weighted avg       0.91      0.91      0.91     60000
```



FLAWS INTRODUCED

Random Forest

The training dataset was deliberately manipulated to create a significant imbalance. The majority class (fully_paid = 1) was oversampled to comprise 95% of the training data, while the minority class (fully_paid = 0) was undersampled to only 5% of the training data. This imbalance biases the model toward predicting the majority class.

IMPACT

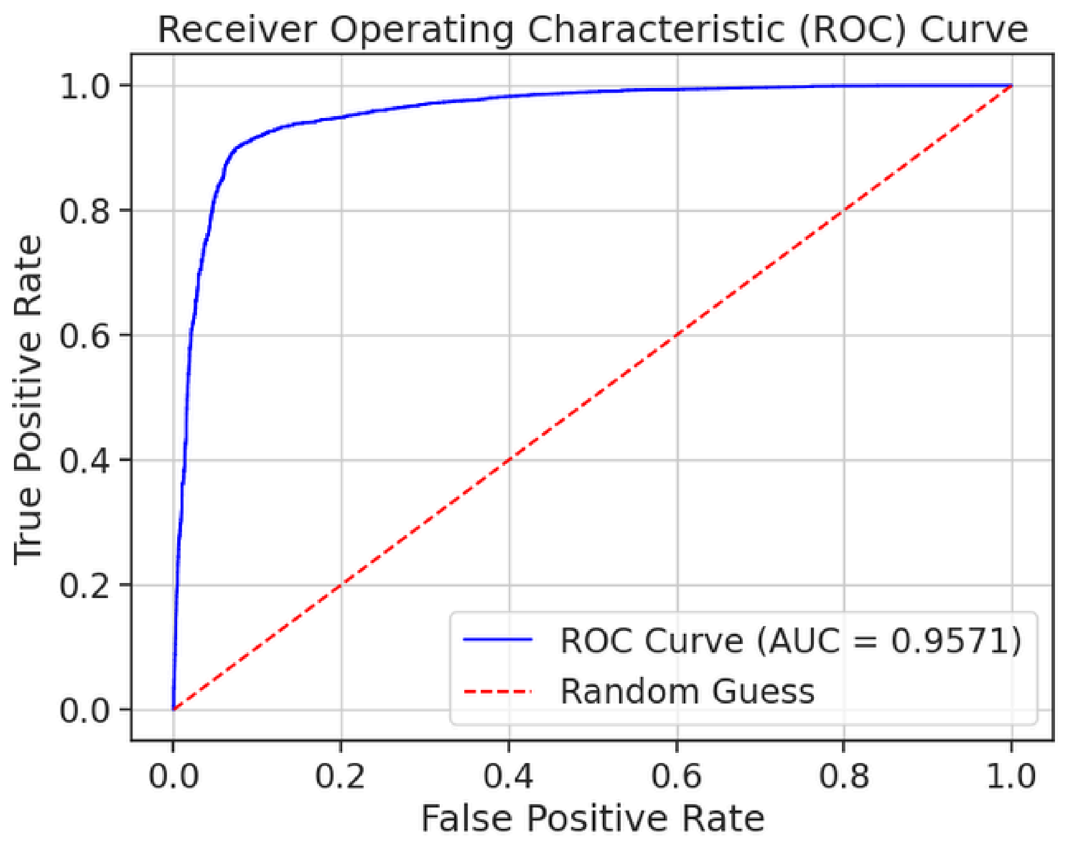
PREDICTIVE BIAS

High recall for the majority class.
Very low precision and recall for the minority class.

MISLEADING METRICS

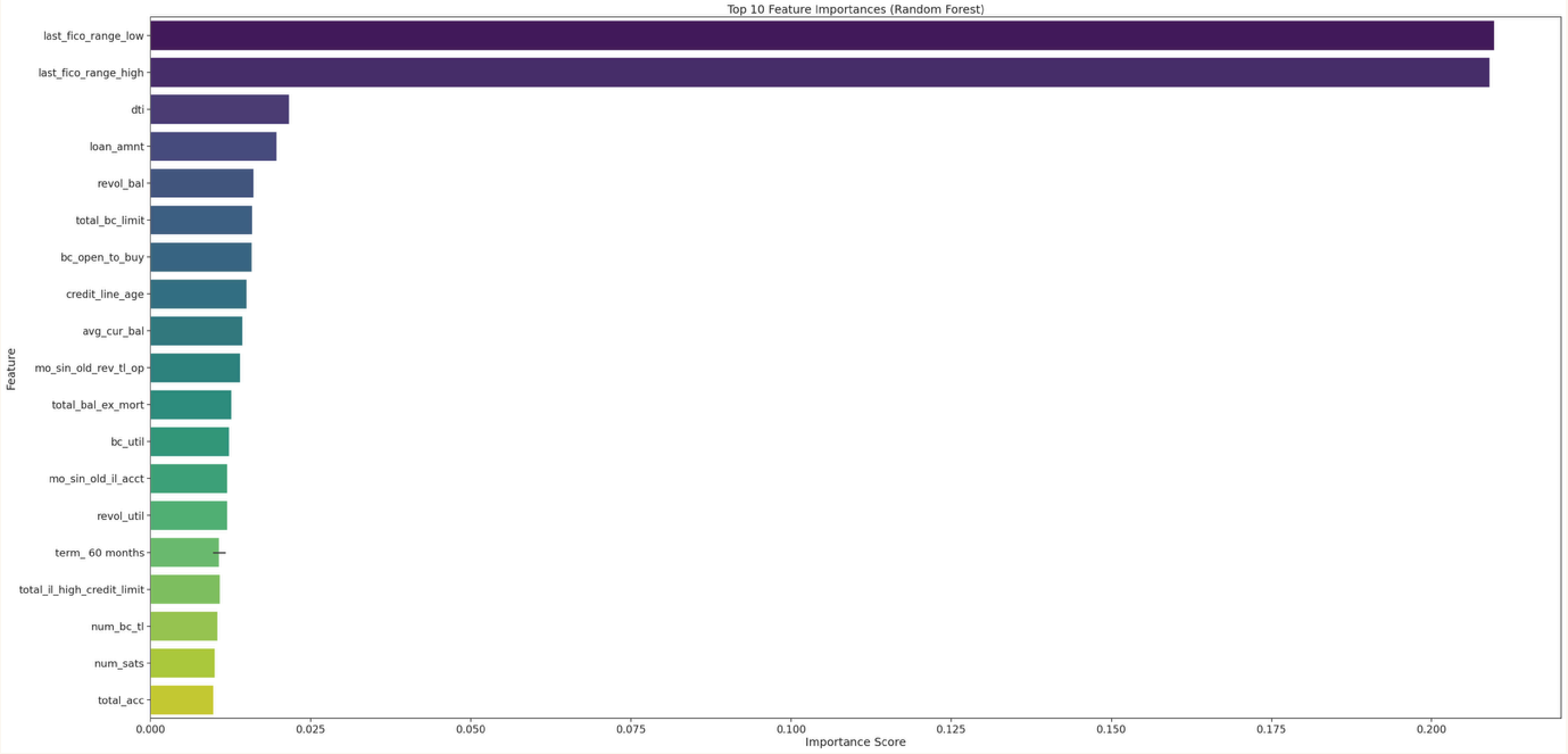
ROC AUC score remains artificially high (0.9571)

```
Random Forest Performance (with Imbalanced Training Set):
Accuracy: 0.5017
ROC AUC: 0.9571

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.00      0.01      5000
           1       0.50      1.00      0.67      5000

    accuracy                           0.50     10000
   macro avg       0.75      0.50      0.34     10000
weighted avg       0.75      0.50      0.34     10000
```



IMPACT

Features most relevant to the minority class might appear less important because the model does not encounter enough diverse data to learn their significance.



Key Takeaway:

This flaw highlights how training data imbalance can drastically affect a model's performance, leading to skewed predictions, poor generalization, and potentially severe real-world consequences in financial applications.

FLAWS INTRODUCED

XG BOOST

A portion (10%) of the training labels (y_train) were randomly flipped, changing some 1 (fully paid) labels to 0 (charged off) and vice versa.

IMPACT

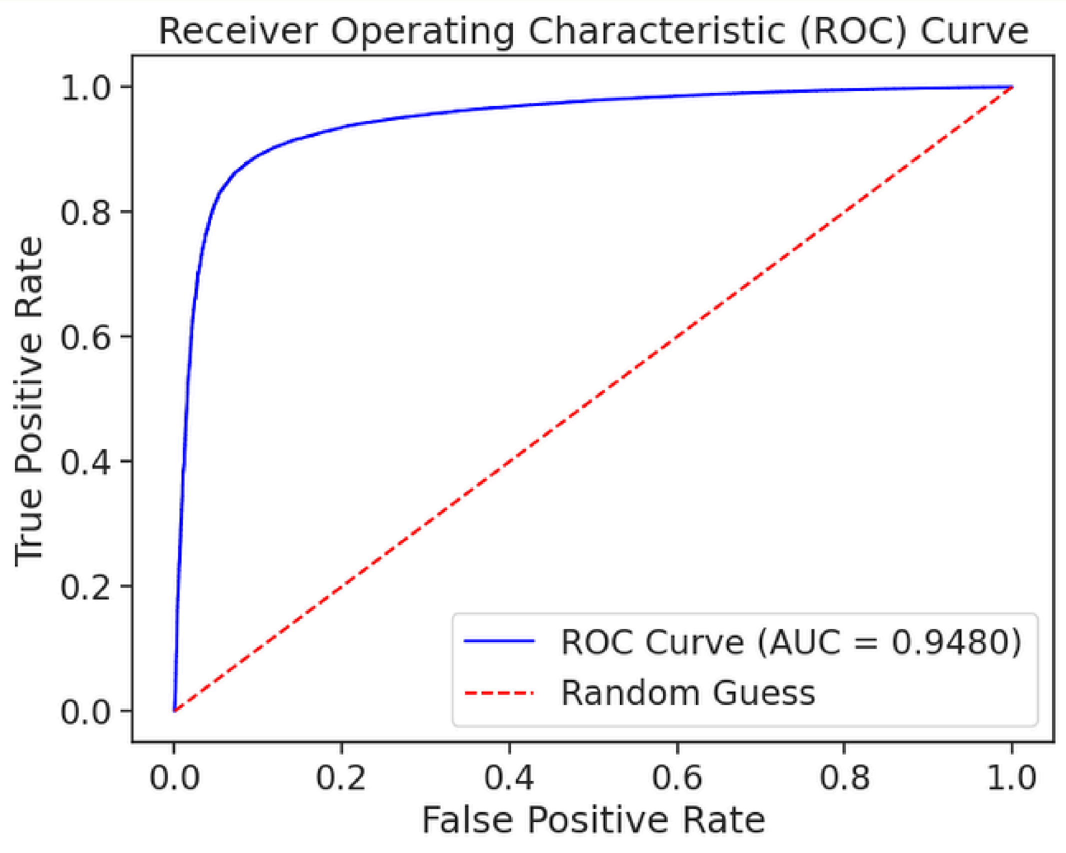
MODEL ACCURACY

The model struggles to find accurate patterns in the training data due to conflicting signals caused by incorrect labels.

DEGRADED ACCURACY

The training performance might seem adequate.

```
XGBoost Performance (with Flawed Labels):
Accuracy: 0.9083
ROC AUC: 0.9480

Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.79      0.78     12194
           1       0.95      0.94      0.94     47806

    accuracy                           0.91     60000
   macro avg       0.86      0.87      0.86     60000
weighted avg       0.91      0.91      0.91     60000
```

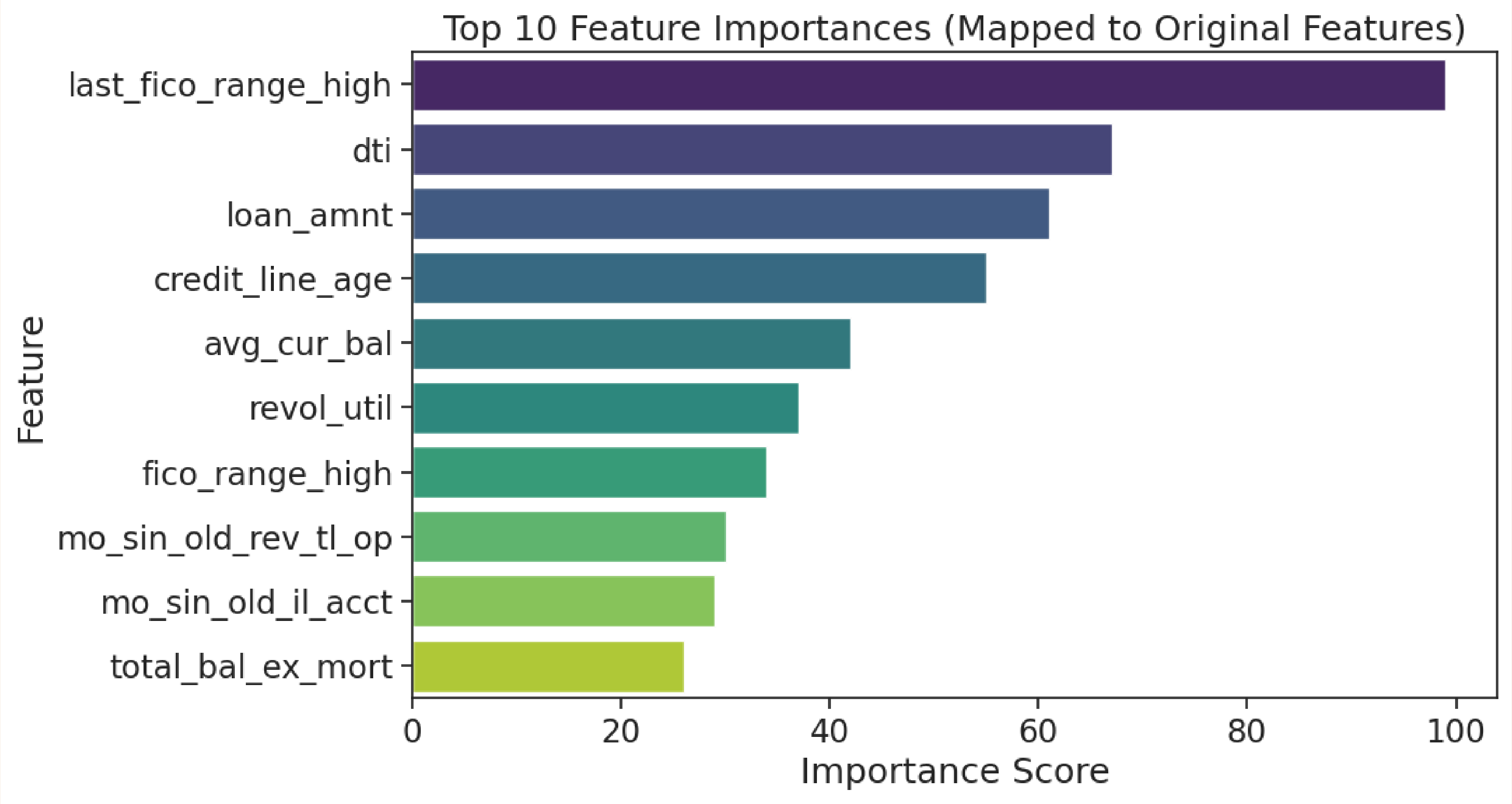


Even if the ROC AUC appears reasonable, the model's predictions may not be trustworthy in real-world applications.

IMPACT

Misleading Feature Importance

Features correlated with the flipped labels may gain or lose importance inappropriately, leading to incorrect interpretations of the drivers behind loan repayment or default.



Key Takeaway:

By introducing this flaw, we demonstrate how critical clean and accurate labeling is to building reliable machine learning models.

Winning Model

The XGBoost Classifier is the "winning" model for the following reasons

Balanced Performance

Class Representation

Generalization Ability

Reliability

VALIDATOR HAT

FLAWS INTRODUCED

1: Mean/Mode Imputation for Missing Data

What Was Done:
Mean imputation for numerical features.
Mode imputation for categorical features.

```
# Create transformers for numerical and categorical features
numerical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')),  # Imputer for numerical
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),  # Imputer for
    ('encoder', OneHotEncoder(handle_unknown='ignore', sparse_output=Fals
])
```

Why This Is a Flaw:
Assumes missing data is random, ignoring patterns in missingness.
Masks informative signals related to missing values.

IMPACTS

Introduces artificial stability.

Creates biased predictions.

Results in overconfident model predictions

Leads to biased or misleading predictions, especially for edge cases with missing data.

RECOMMENDATIONS

Addressing Imputation Bias

Implement predictive imputation techniques.

Add missingness indicators.

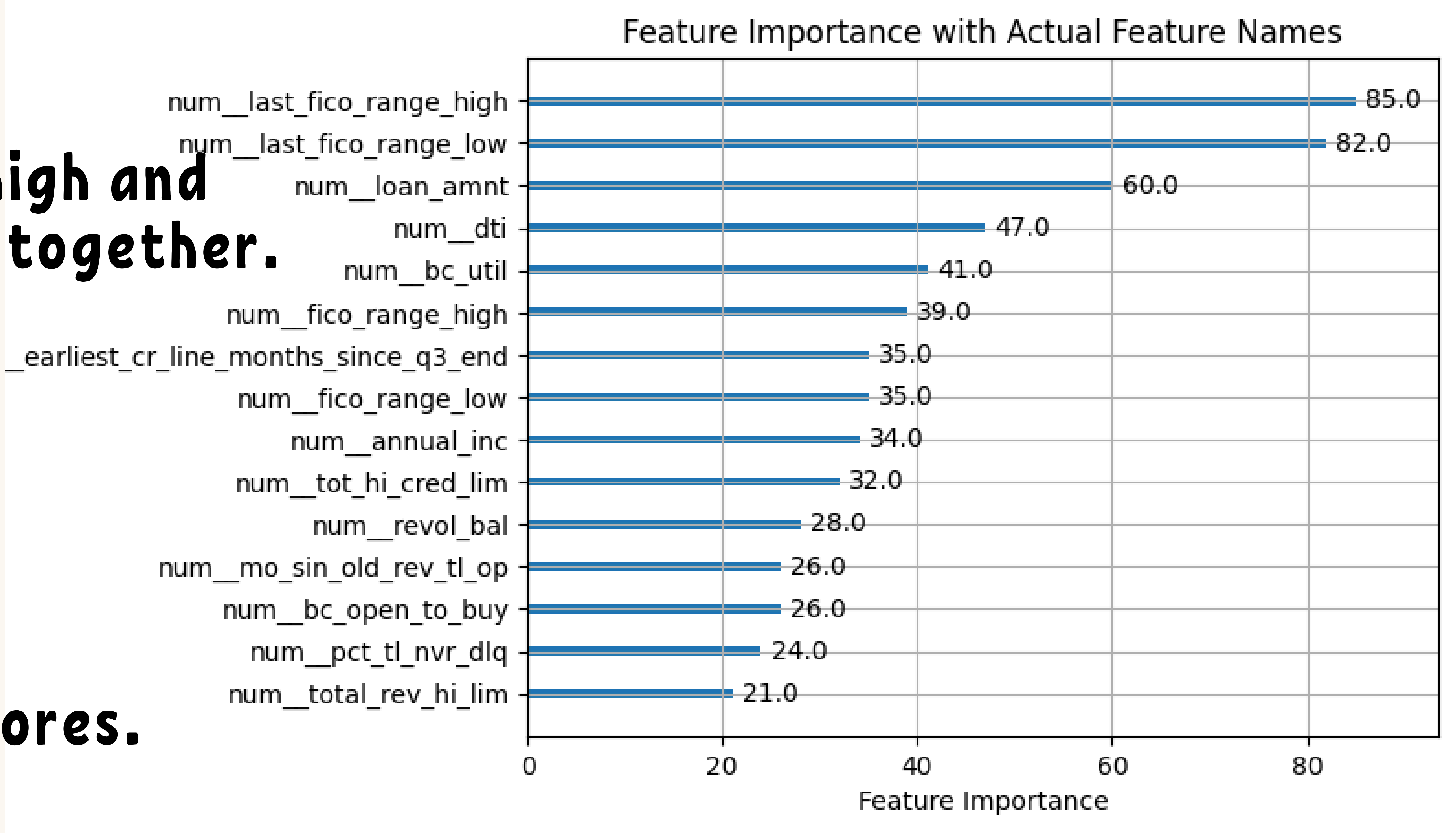
Capture significance of missing values

FLAWS INTRODUCED

2: High Correlation Between Features (Multicollinearity)

What Was Done:

Features like last_fico_range_high and last_fico_range_low were used together.



Why This Is a Flaw:
Reduces model robustness.

Compromises interpretability.

Inflates feature importance scores.

Compromises model insights.

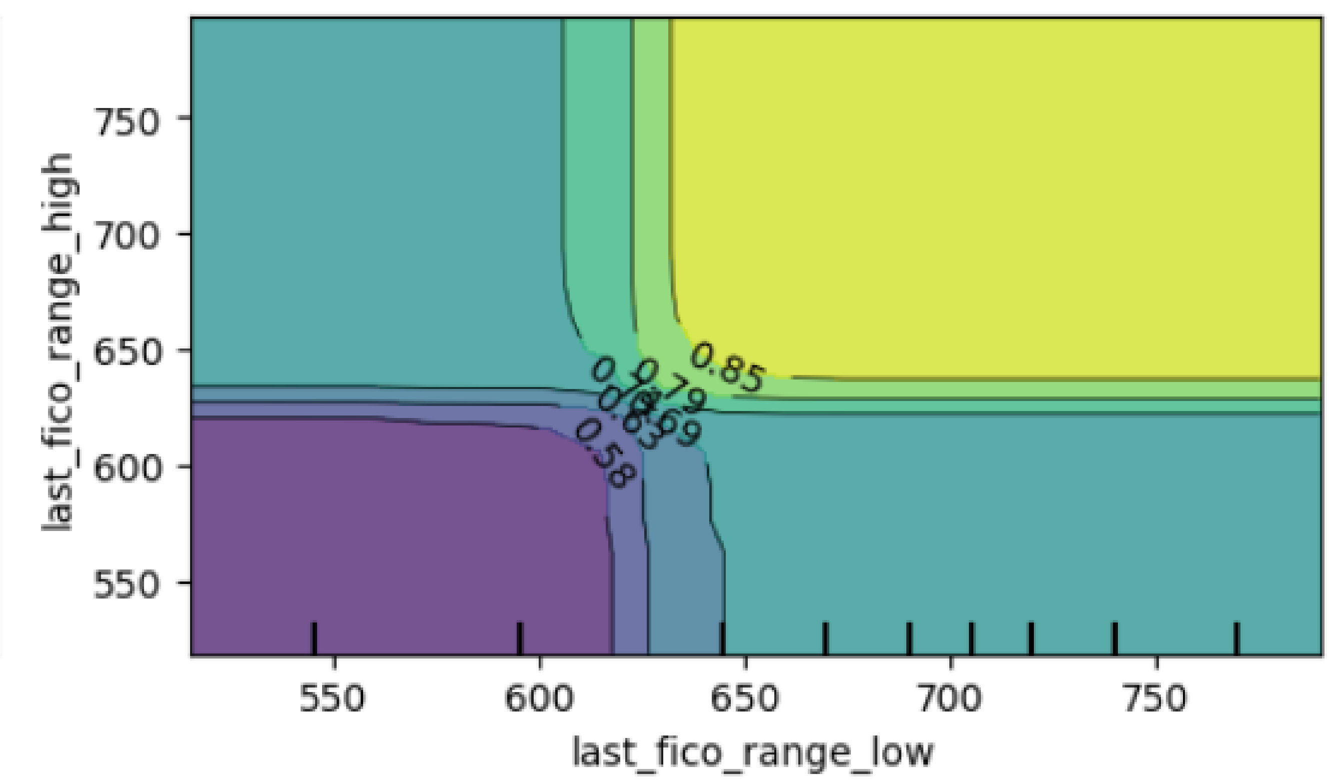
IMPACTS

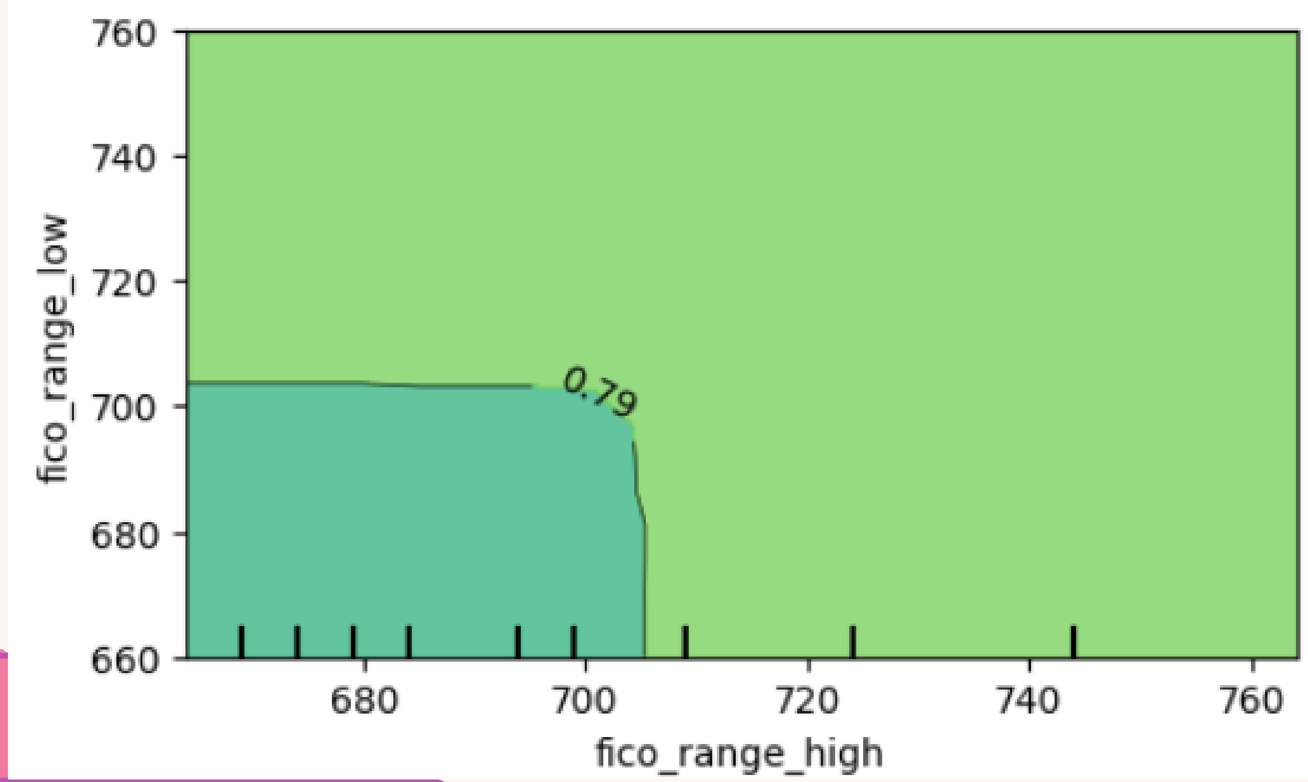
Creates unstable predictions.

Leads to potential model overfitting.

Reduces reliability in production thereby affecting decision-making.

Compromises model insights.





RECOMMENDATIONS

Resolving Multicollinearity

Identify correlated feature pairs.

Remove redundant features.

Feature engineering solutions such as combining related features.

Perform detailed correlation analysis e.g create correlation heatmap for visualization, use Variance Inflation Factor (VIF) analysis.

Advanced techniques e.g. Ridge Regression.

DISCUSSION

WERE THERE ANY OTHER FLAWS THAT YOU WANTED TO INTRODUCE AND TEST THE MODEL? IF THERE WERE, WHAT ASPECT OF THE MODEL YOU WANTED TO TEST WITH THAT FLAWS?

THANK YOU!