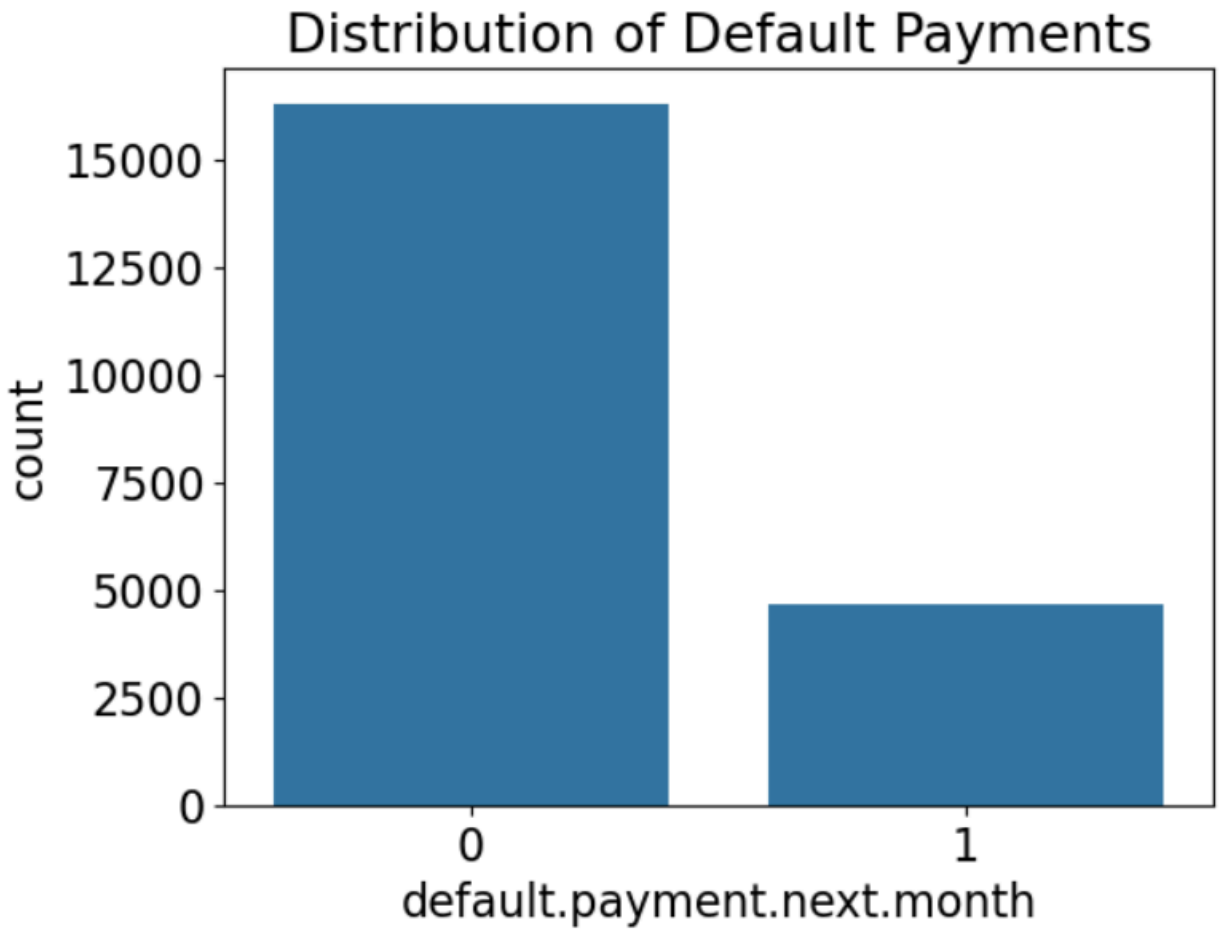
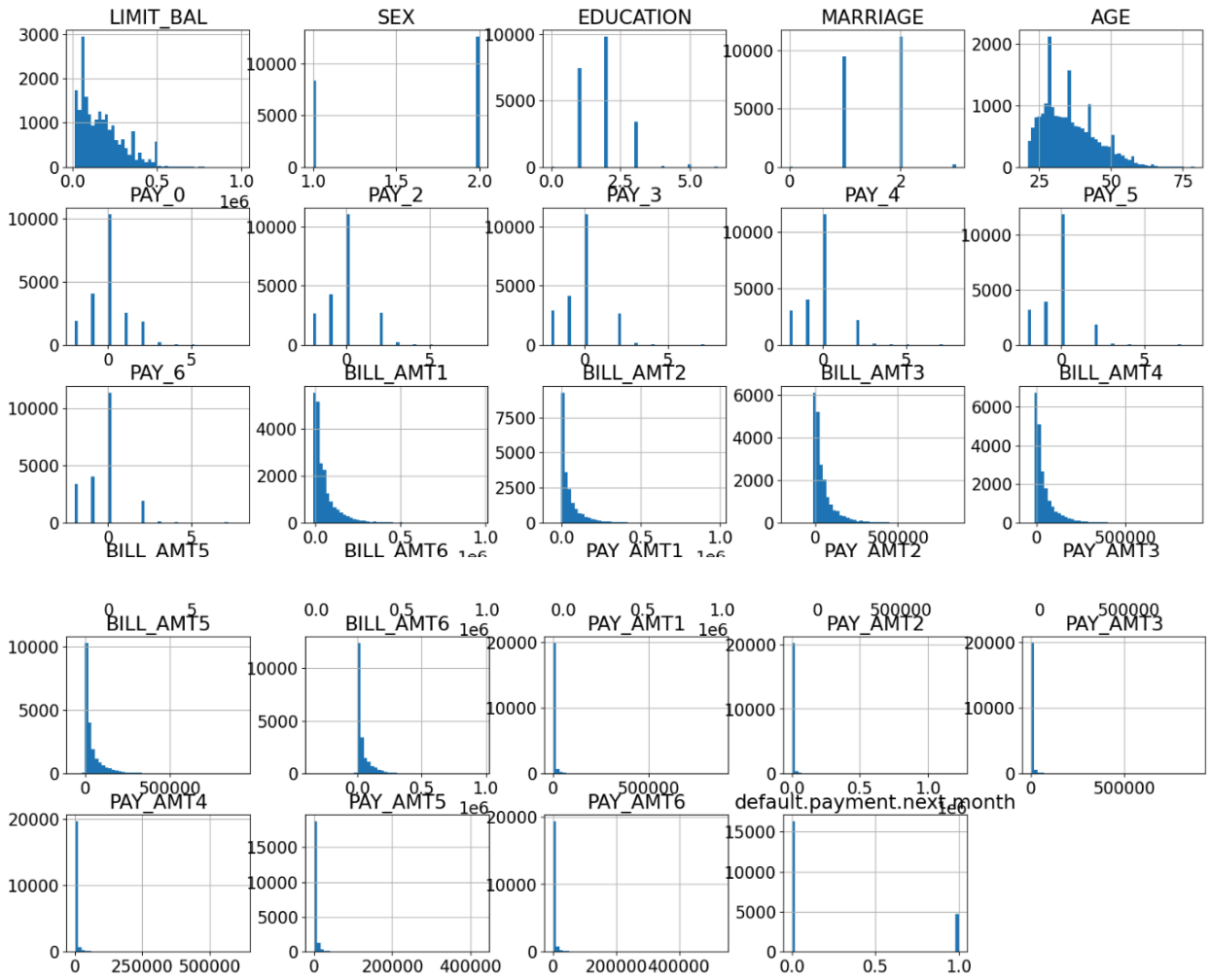
Problem Description.

The goal of this project was to predict whether a customer would default on their next credit card payment using the **"Default Payments of Credit Card Clients in Taiwan"** dataset from Kaggle. This classification task aims to help financial institutions identify high-risk clients and make data-driven decisions to minimize credit risk.

EDA:

Our EDA on the initial data has uncovered a couple of useful insights. We decided to use a histogram to show the distribution of customer count and whether they default their credit card payment for the next month as shown below.





Our initial exploration has surfaced several important observations and challenges that need to be addressed before building a robust model. Here's what we discovered:

Limited Feature Set

The dataset contains very few features. This limitation emphasizes the importance of extracting as much meaningful information as possible from the available data. It also necessitates careful feature engineering to improve model performance.

Class Imbalance

The dataset exhibits a significant class imbalance, which can skew model predictions toward the majority class. To address this, we've selected macro-average F1 score as our evaluation metric. This metric ensures that both classes receive equal weight, making it a fair measure of the model's ability to handle the minority class effectively.

Feature Scaling

The features in the dataset have vastly different ranges. Without standardizing these features, certain features with larger ranges could disproportionately influence the model. Standardization will be a key preprocessing step.

Collinearity Among Features

Several features in the dataset are highly correlated with one another. This collinearity can negatively affect model interpretability and lead to overfitting. Dimensionality reduction techniques or careful feature selection may be necessary to address this.

Outliers

The dataset contains quite a few outliers. These anomalies can distort the distribution of data and impact model performance. We'll need to identify and appropriately handle these outliers, either by capping, removing, or transforming them.

Messy Data

The data doesn't always align with its provided description, leading to ambiguity in certain fields. For example:

Education Levels: The dataset includes education levels 5 and 6, but these levels are not defined in the documentation. Clarifying their meaning is critical for proper interpretation.

PAY_ Variables:* The values of the PAY_* features include unexplained categories like -2 and 0. Understanding the significance of these values is essential to accurately use these features in modeling.

Model Description:

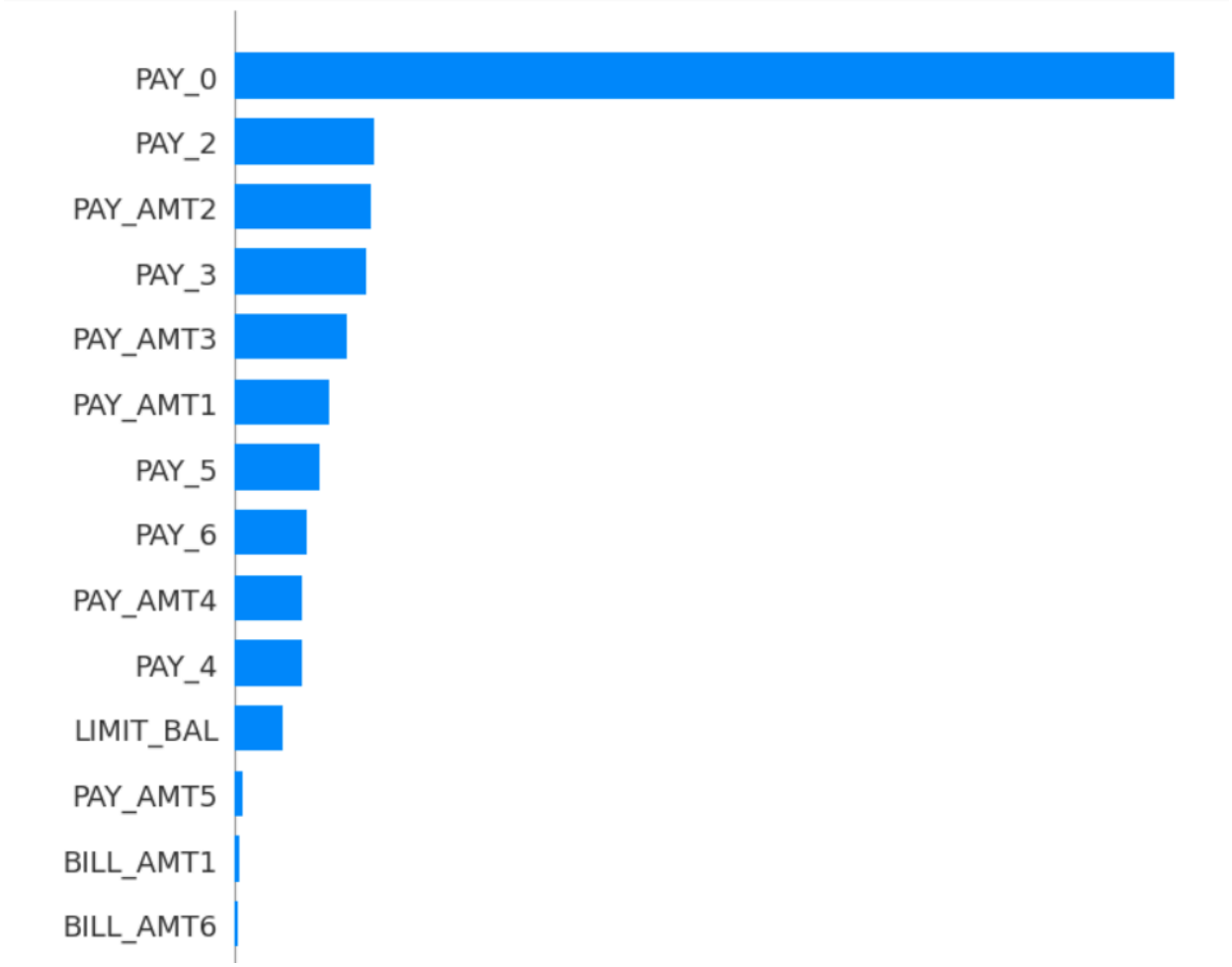
For our baseline model, we used the DummyClassifier model from sklearn's library and told it to always predict the most frequently occurring value based on the training data. Since the majority of customers did NOT default their payment next month in the training data set, DummyClassfier predicted that all customers always paid their credit card bills for the following month on time. The performance of our other models should most likely be better than this baseline and definitely not worse.

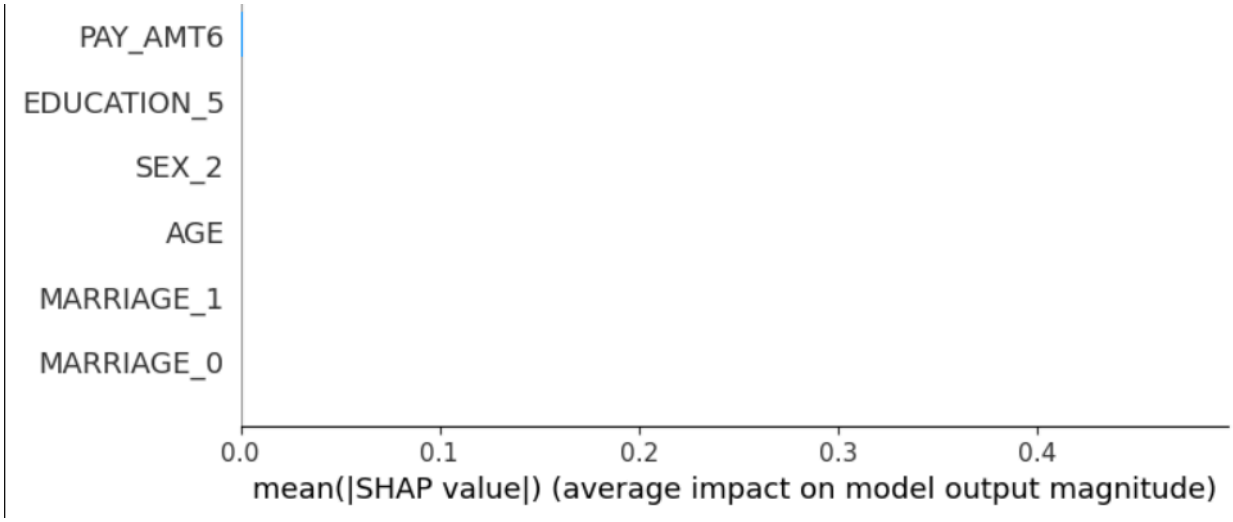
```
results = {}
dummy = DummyClassifier(strategy="most_frequent")
results["dummy"] = mean_std_cross_val_scores(
    dummy, X_train, y_train, return_train_score = True
)
pd.DataFrame(results)
```

We used Logistic Regression as one of our models, which assigns weights to features based on their impact on predicting if a customer will default. We tuned the regularization parameter `C` using GridSearchCV, and the model achieved an average test accuracy of **81.6%**. It performed well overall and showed a good balance between training and test accuracy.

Next, we tried three more advanced models: Random Forest, LightGBM, and XGBoost. Random Forest, which combines multiple decision trees, did great on the training data with **99.9% accuracy** but slightly overfitted, with a test accuracy of **81.6%**. LightGBM gave the best results, scoring **82.0%** test accuracy and staying efficient with a fit time of about half a second. XGBoost was fast too, but its test accuracy was slightly lower at **81.3%**.

Feature Importances





The shap summary plot above tells us that the PAY_0 feature has the most impact on our prediction.

Results Description:

This is a summary of our results:

Model Comparison Table

Model	Fit Time (s)	Score Time (s)	Test Score (Mean ± Std)	Train Score (Mean ± Std)
Dummy	0.003 ± 0.001	0.002 ± 0.000	0.496 ± 0.010	0.501 ± 0.004
Logistic Regression	0.298 ± 0.053	0.010 ± 0.001	0.625 ± 0.006	0.627 ± 0.003
Logistic Regression (Tuned)	0.096 ± 0.009	0.012 ± 0.002	0.629 ± 0.006	0.630 ± 0.003

Random Forest	4.362 ± 0.173	0.089 ± 0.022	0.815 ± 0.005	0.999 ± 0.000
XGBoost	0.223 ± 0.057	0.013 ± 0.002	0.812 ± 0.005	0.905 ± 0.003
LightGBM	0.148 ± 0.057	0.008 ± 0.002	0.820 ± 0.005	0.853 ± 0.001
Random Forest (Optimized)	0.340 ± 0.012	0.013 ± 0.000	0.819 ± 0.005	0.849 ± 0.001
XGBoost (Optimized)	0.125 ± 0.011	0.014 ± 0.003	0.821 ± 0.005	0.821 ± 0.001
LightGBM (Optimized)	0.074 ± 0.006	0.013 ± 0.002	0.821 ± 0.007	0.828 ± 0.001

Conclusion: Final Model Performance and Insights

After a rigorous modeling process, all tested models surpassed the baseline performance. Here's a summary of our findings and final decisions:

Best Model: LightGBM Classifier

- The **LightBGM classifier with tuned hyperparameters** emerged as our best-performing model.
- It demonstrated the following advantages:
    - **Higher Macro-Average F1 Score:** Achieved 0.679 on the held-out test set, closely aligning with the cross-validation score of 0.68. This consistency indicates minimal optimization bias.
    - **Reduced Overfitting:** LightGBM exhibited less overfitting compared to the tuned Random Forest model.

    - **Efficiency:** It was significantly faster, which is a critical factor for scalability and iterative experimentation.
- These qualities make LightGBM our final model of choice for predicting credit card defaults.