Report

Cleaning

To get started, we cleaned the dataset by first selecting the desired columns, dropping the NaN values, and deleting categories with less than 10 occurrences in categorical values. The reason being that otherwise, it would result in errors later in the process.

Under sampling

Our dataset was unbalanced between defaulting and non-defaulting clients, with 91% and 9%, respectively. Under sampling is a technique used in machine learning to balance the class distribution in a dataset. Specifically, it involves reducing the number of instances in the majority class so that it is closer to the number of instances in the minority class.

Bins

When creating bins for the data, the histogram of the variable is typically used to get an idea of its distribution. However, the presence of outliers in the data can skew the histogram and create bins that are not representative of most of the data. Therefore, the bins need to be adjusted for the effect of outliers.

In this case, the bins were selected based on histograms made with the data and then adjusted for the effect of outliers. Furthermore, each bin was required to have at least a 0.04 IV value to ensure that the variable had good predictive power across all categories.

Next, with a binning transformer, we moved the data to the selected categories. The process of moving data to the selected categories using a binning transformer involves grouping data points into predefined categories or bins. Once the bins are defined, the binning transformer maps each data point to its corresponding bin based on the value of the variable being binned. This mapping results in a new set of values that are now categorical in nature. The process of binning helps to reduce noise in the data and provides a more structured representation of the original data.

WoE

After transforming our data to the respective Weight of Evidence (WoE) using the binning transformer, we analyzed the WoE and IV values. By analyzing these values, we can determine if our bins have good predictive power or if we need to restructure them.

If the WoE values are consistent within each bin and there is a clear pattern in the values across the bins, it indicates that the binning process has been successful, and the resulting bins have good predictive power. However, if the WoE values are inconsistent or there is no clear pattern across the bins, it suggests that the binning process needs to be restructured to improve its predictive power.

IV

After performing the WoE and IV analysis, we found that some columns had low IV, indicating that they do not have significant predictive power for our target variable. Therefore, we decided to only keep the columns with high enough IV to ensure that our model is built with the most relevant and predictive features. By doing so, we can improve the accuracy and effectiveness of our model in predicting the target variable.

Pipeline

Using custom transformers with a fit and transform method allows us to integrate them seamlessly into a machine learning pipeline. With the help of sklearn.pipeline, we can stack these transformers and models into a single job that deals with raw data. By doing so, we can easily automate the process of data preprocessing, feature engineering, and model training, making the workflow more efficient and less error prone. Moreover, pipelines help us to prevent data leakage by ensuring that any data transformation that occurs during training is also applied identically to the validation and test datasets.

Stacking

Stacking multiple models can help improve the overall performance of the predictive model. In this case, the team used a combination of three different models: Logistic Regression, Random Forest, and Gradient Boosting.

After applying the custom transformers to preprocess the data, the team fed the transformed data into each of the three models. Each model was trained to predict the outcome of the target variable based on the input data.

Next, the team stacked the three models using another Logistic Regression. This step involves training a logistic regression model on the outputs of the three individual models. The stacked model then decides how much weight to give to each individual model when making a final prediction.

This technique of stacking multiple models leverage the strengths of each individual model and mitigating their weaknesses.

Conclusion

In conclusion, in this project, we have demonstrated how to preprocess data using techniques such as cleaning, under sampling, binning, and transforming data to Weight of Evidence. We have also discussed the importance of analyzing WoE and IV values and using them to decide which columns to keep for building the predictive model. By using custom transformers with a fit and transform method, we can automate the process of data preprocessing and feature engineering. Additionally, we have discussed how stacking multiple models can improve the overall performance of the predictive model by leveraging the strengths of each individual model and mitigating their weaknesses. Overall, by

following these techniques, we can build an efficient and effective predictive model for the target variable.