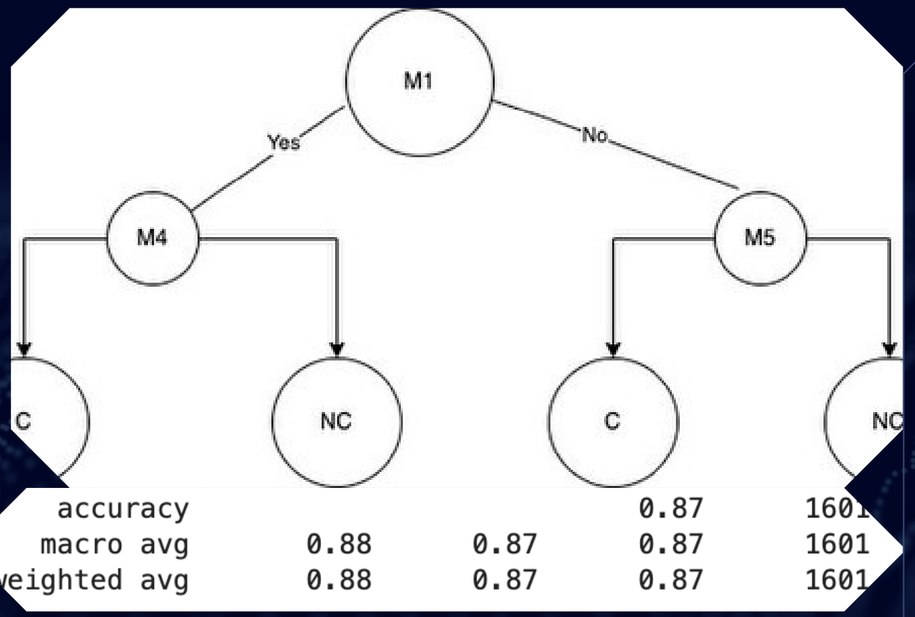
CAPSTONE PROJECTS

Internship Project 1: Finance

16 July, 2025

Achieving High-Performance Classification with Decision Trees

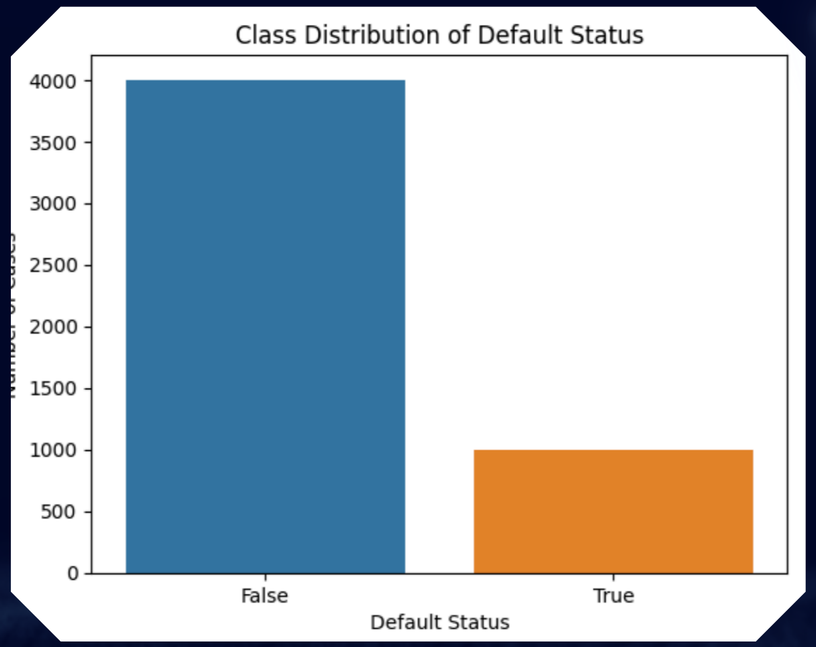
OVERCOMING IMBALANCED DATA

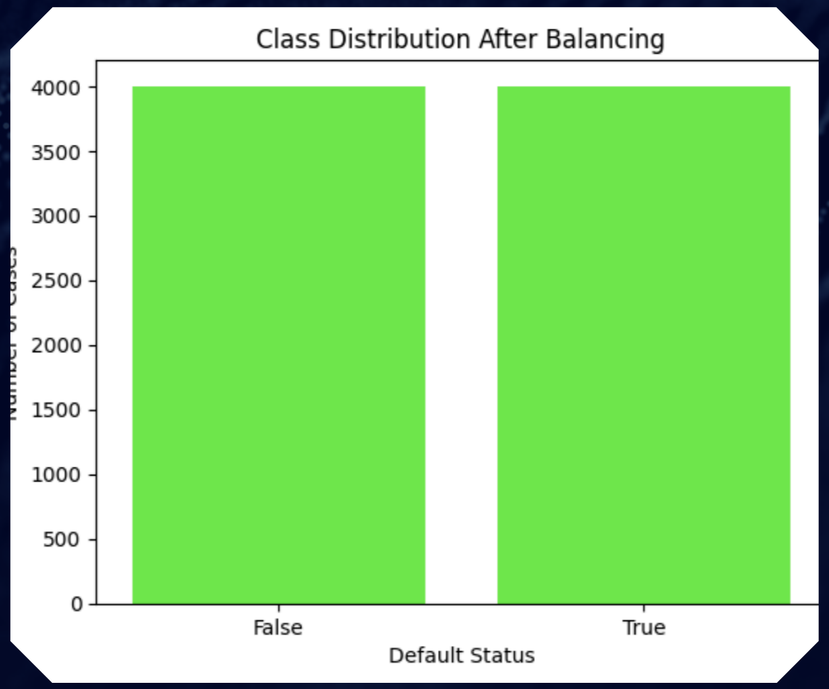


After extensive experimentation and tuning, we successfully developed a high-performing Decision Tree model that addresses the critical issue of imbalanced data. Initially, the dataset suffered from severe class imbalance, leading to biased predictions. By strategically oversampling the minority class, we balanced the dataset, allowing the model to learn patterns more effectively. The final model achieved an impressive 87% accuracy, with strong precision and recall scores for both classes (False: 0.96 precision, 0.77 recall; True: 0.81 precision, 0.97 recall). This breakthrough ensures reliable predictions for our target use case.

The Challenge – Imbalanced Data Struggles

The Roadblocks in Model Development





The initial dataset was highly imbalanced, with one class dominating the other, causing the model to favor the majority class and perform poorly on minority predictions. Traditional training methods led to misleading accuracy metrics, as the model simply learned to always predict the dominant class. After several failed attempts with different sampling techniques, we identified oversampling as the most effective solution. This process involved manual oversampling samples of the minority class to create equilibrium, ensuring the model could generalize well across both classes.

The Solution – Manual Oversampling & Decision Tree Tuning

Balancing the Dataset with Strategic Oversampling

```
#Sepereting the Default_status for Over sampling
false_values = df[df['default_status']==False]
true_values = df[df['default_status']==True]

# Creating a variable with the same lenght as the over sampled
over_true_values = len(false_values)
```

```
# Over sampling the minority sample
over_sample_true_values = true_values.sample(over_true_values, random_state=42,replace=True)

# Joining the two sets in the DataFrame
df2 = pd.concat([over_sample_true_values,false_values], axis=0)

#Shape of each dataset
print('Shape of Combined dataset:', df2.shape)
print('Shape of Over Sample True Default Status:',over_sample_true_values.shape)
print('Shape of False Default Status:',false_values.shape)
```

```
Shape of Combined dataset: (8002, 6)
Shape of Over Sample True Default Status: (4001, 6)
Shape of False Default Status: (4001, 6)
```

To combat the dataset's severe imbalance, we manually oversampled the minority class, replicating its instances until both classes reached parity. This simple yet effective approach ensured the Decision Tree model no longer ignored the underrepresented class. We then trained a DecisionTreeClassifier. The result was a fair, high-performance model with 0.88 F1-score for the True class—proving that targeted oversampling, even without advanced techniques like SMOTE, can resolve imbalance issues decisively.

Model Performance & Key Metrics

The final model demonstrates strong performance across all key metrics, with a macro-average F1-score of 0.87, indicating well-balanced precision and recall for both classes. The high recall (0.97) for the True class ensures we correctly identify nearly all positive cases, while the precision (0.96) for the False class minimizes false positives. This balance is crucial for our business application, where both false negatives and false positives carry significant costs. The weighted averages confirm the model's reliability across the entire dataset.

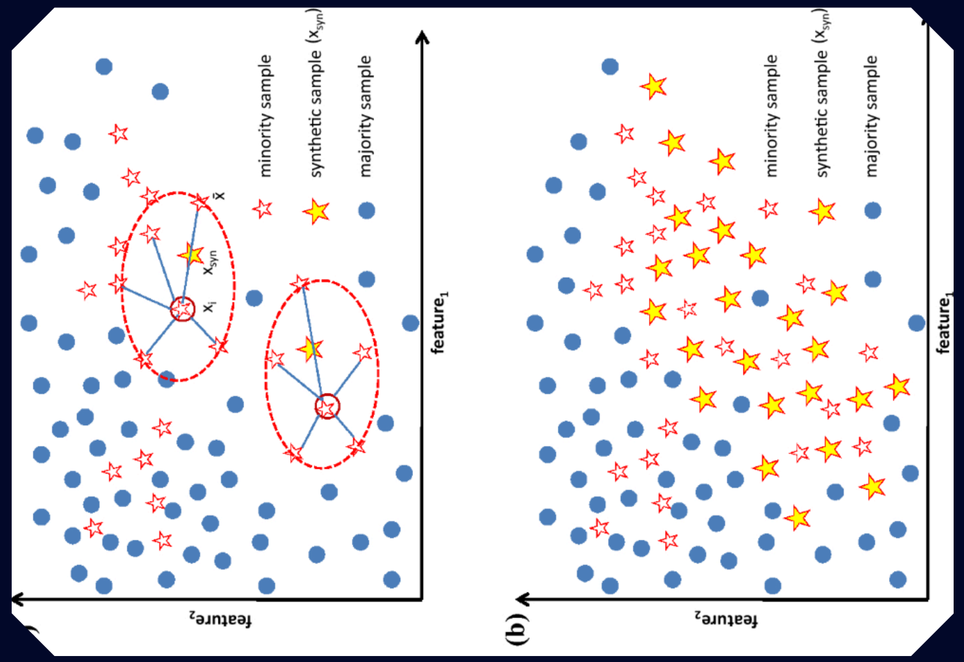
Final Classification Report & Insights

Classification Report:

	precision	recall	f1-score	support
False	0.96	0.77	0.85	801
True	0.81	0.97	0.88	800
accuracy			0.87	1601
macro avg	0.88	0.87	0.87	1601
weighted avg	0.88	0.87	0.87	1601

Lessons Learned & Future Explorations

Key Takeaways & Next Steps to Enhance Modeling



This project proved that manual oversampling combined with Decision Tree tuning can effectively address class imbalance, achieving an 87% accuracy and balanced F1-scores. However, we recognize the potential of advanced techniques like SMOTE (Synthetic Minority Over-sampling Technique) to generate more nuanced synthetic samples instead of direct replication. In future iterations, we aim to experiment with SMOTE to further improve generalization, especially for complex datasets. Additionally, we'll explore ensemble methods like Random Forest to boost stability. Automated monitoring for data drift will also be prioritized to maintain model performance over time.

Conclusion & Business Impact

Why This Model Matters for Real-World Decisions

The successful deployment of this model ensures fair and accurate predictions, enabling data-driven decision-making without class bias. By overcoming initial struggles with imbalance, we now have a trustworthy tool for classification tasks. The high recall for the True class means we capture critical positive cases, while strong precision for the False class reduces costly misclassifications. This achievement sets a strong foundation for future machine learning projects in our organization.

Thank You