




PREDICTING STUDENT PERFORMANCE USING LOGISTIC REGRESSION



Group 17



PROJECT OVERVIEW

1

Objective: Analyze student performance using logistic regression and other ML models on simulated and real datasets.

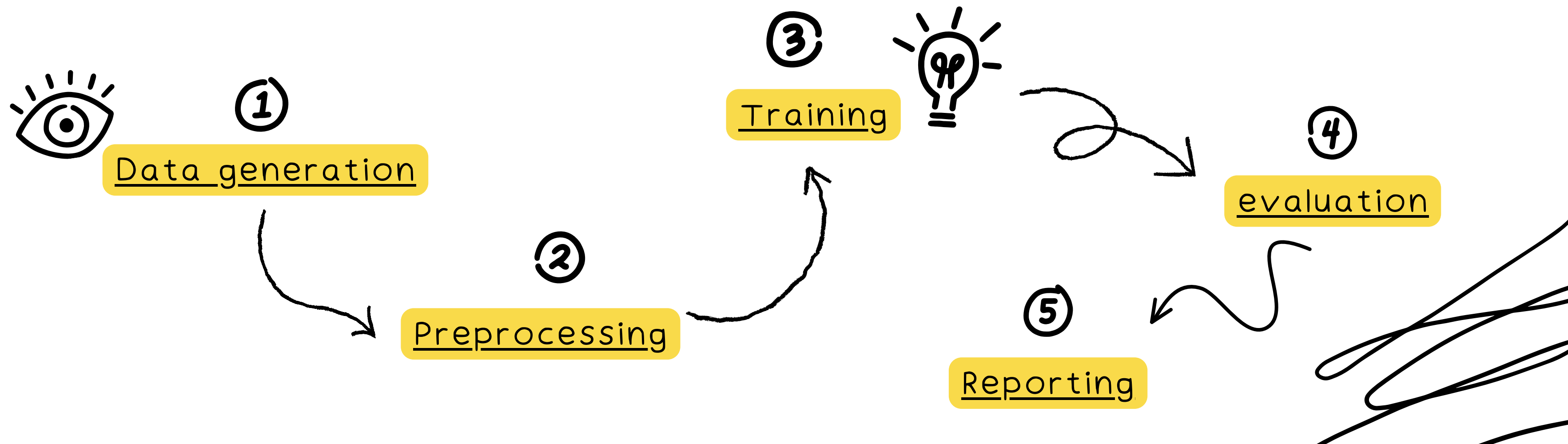
2

Dataset Overview:

Dataset	Records	Features	Classes
Simulated	1000	Study Hours, Sleep Hours, Attendance	Pass/Fail
Real	649	Study Hours, Sleep Hours, Attendance	Pass/Fail, Grade Class (Fail, Pass, Excellent)

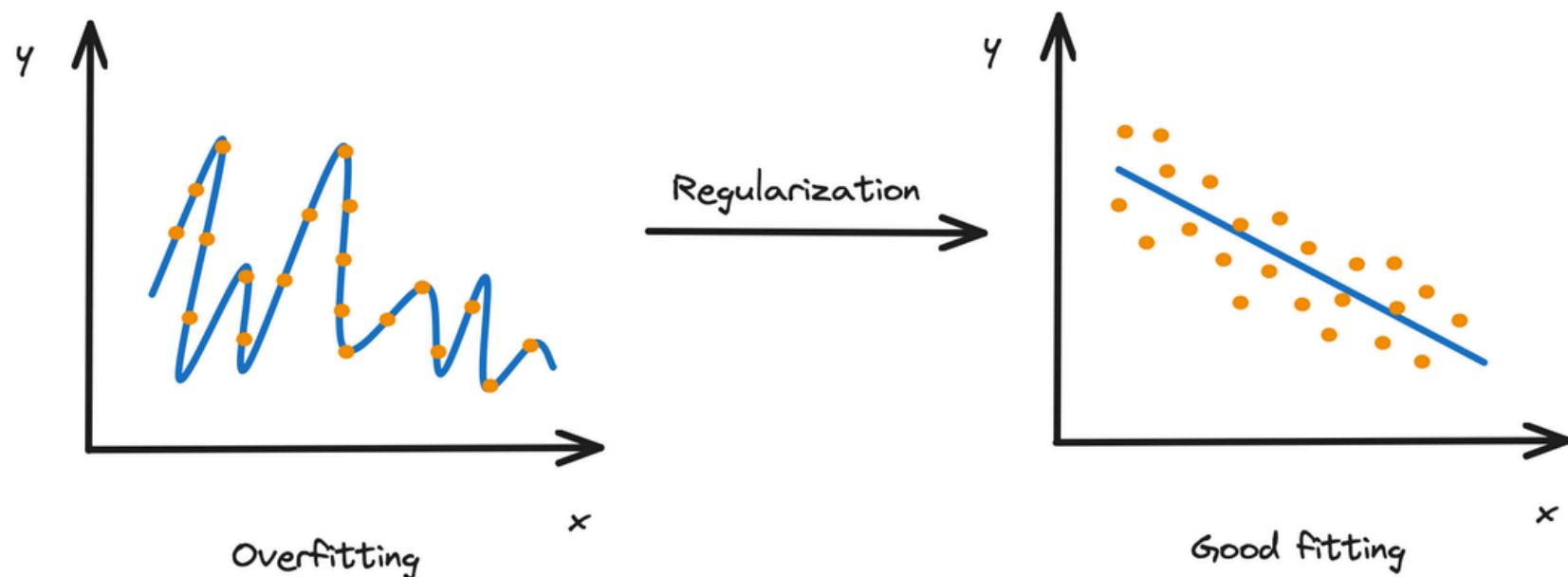
PROJECT OVERVIEW

- ③ Models: Logistic regression (basic, L1/L2, balanced, SMOTE, polynomial), decision trees, Bayesian, deep learning.
- ④ Outputs: Metrics, visualizations, HTML/PDF report, interactive demo.

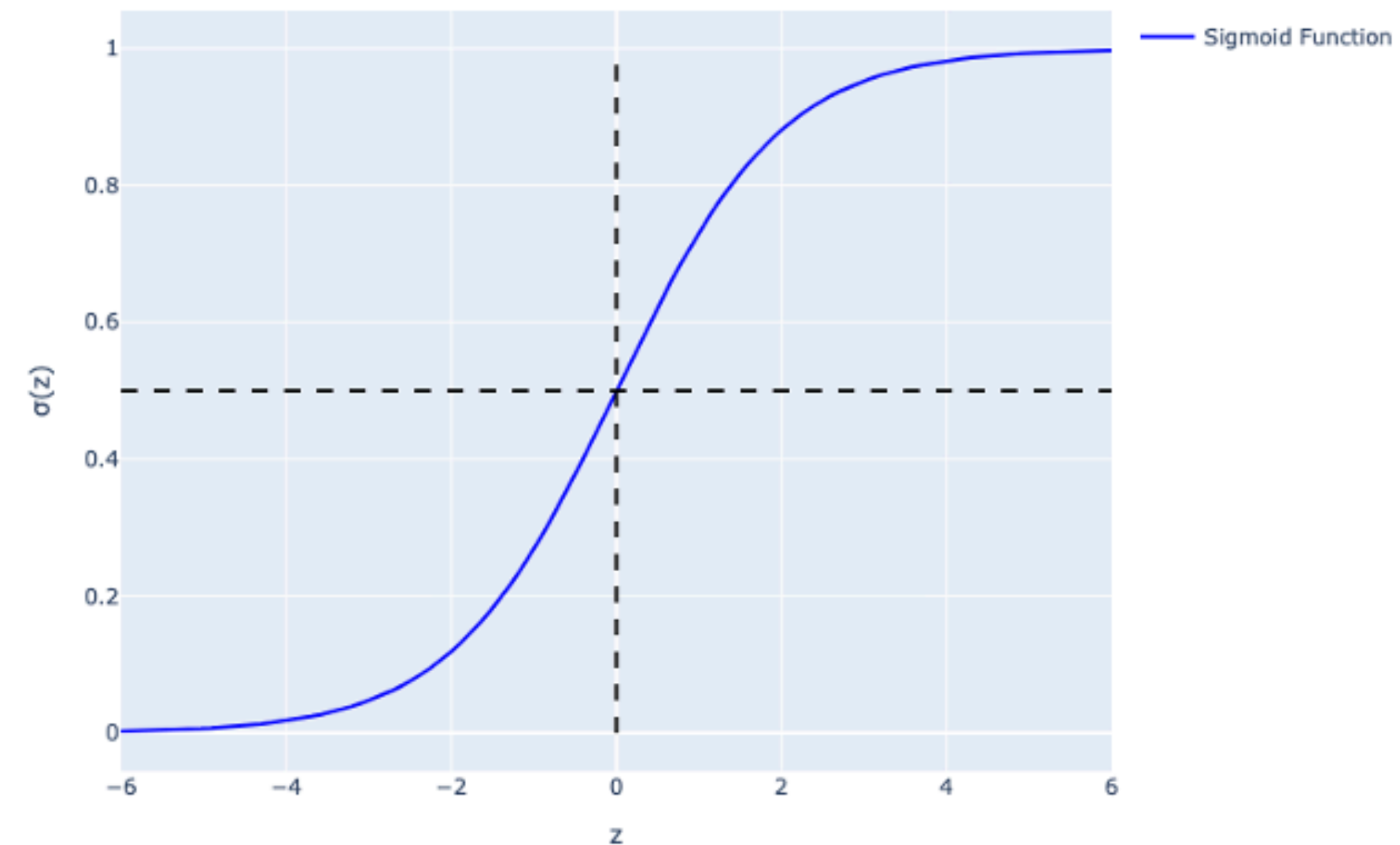


THEORETICAL FOUNDATION

- Sigmoid Function: Maps logits to probabilities (0 to 1).
- Cross-Entropy Loss: Optimized via gradient descent.
- Regularization: L1 (Lasso) and L2(Ridge) to prevent overfitting.



Sigmoid Function



$$\text{Equation: } \sigma(z) = 1 / (1 + e^{(-z)}).$$

FEATURE ANALYSIS

- Model Coefficients:
 - Real Data (L2): Study Hours (0.177), Attendance (0.095)
 - Simulated Data (L2): Study Hours (1.058), Attendance (0.744)
- Feature Correlations:
 - Low correlation between features in both datasets



Figure 2: Correlation heatmaps for simulated and real data features

CLASS DISTRIBUTIONS

Simulated Data:
Binary: Pass (74%), Fail (26%)

Multi-Class: Excellent (63%),
Pass (22.5%), Fail (14.5%)

Real Data:
Binary: Pass (88.46%), Fail (11.54%)

Real data is more imbalanced,
necessitating techniques like SMOTE.

Class Distribution

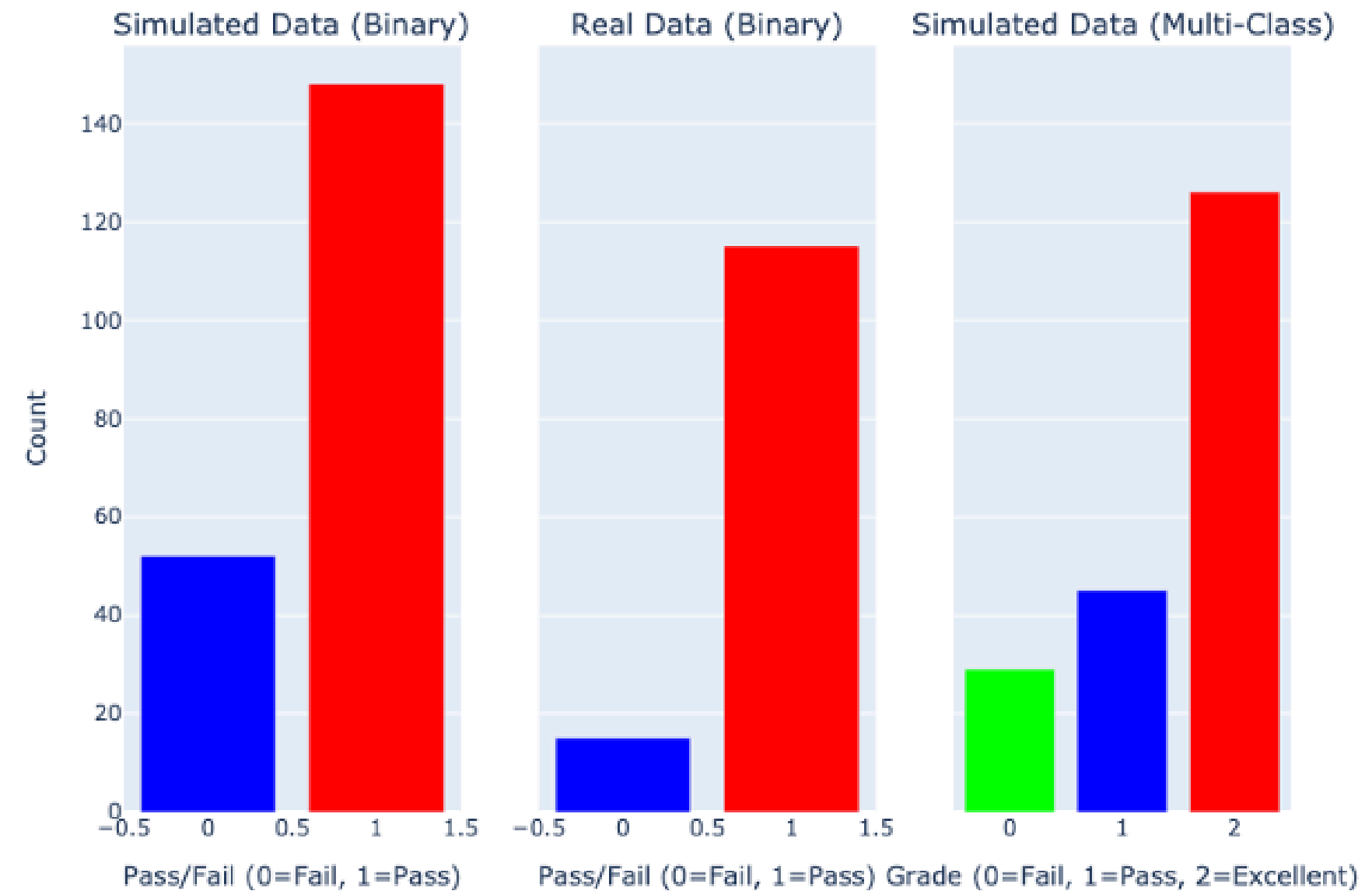
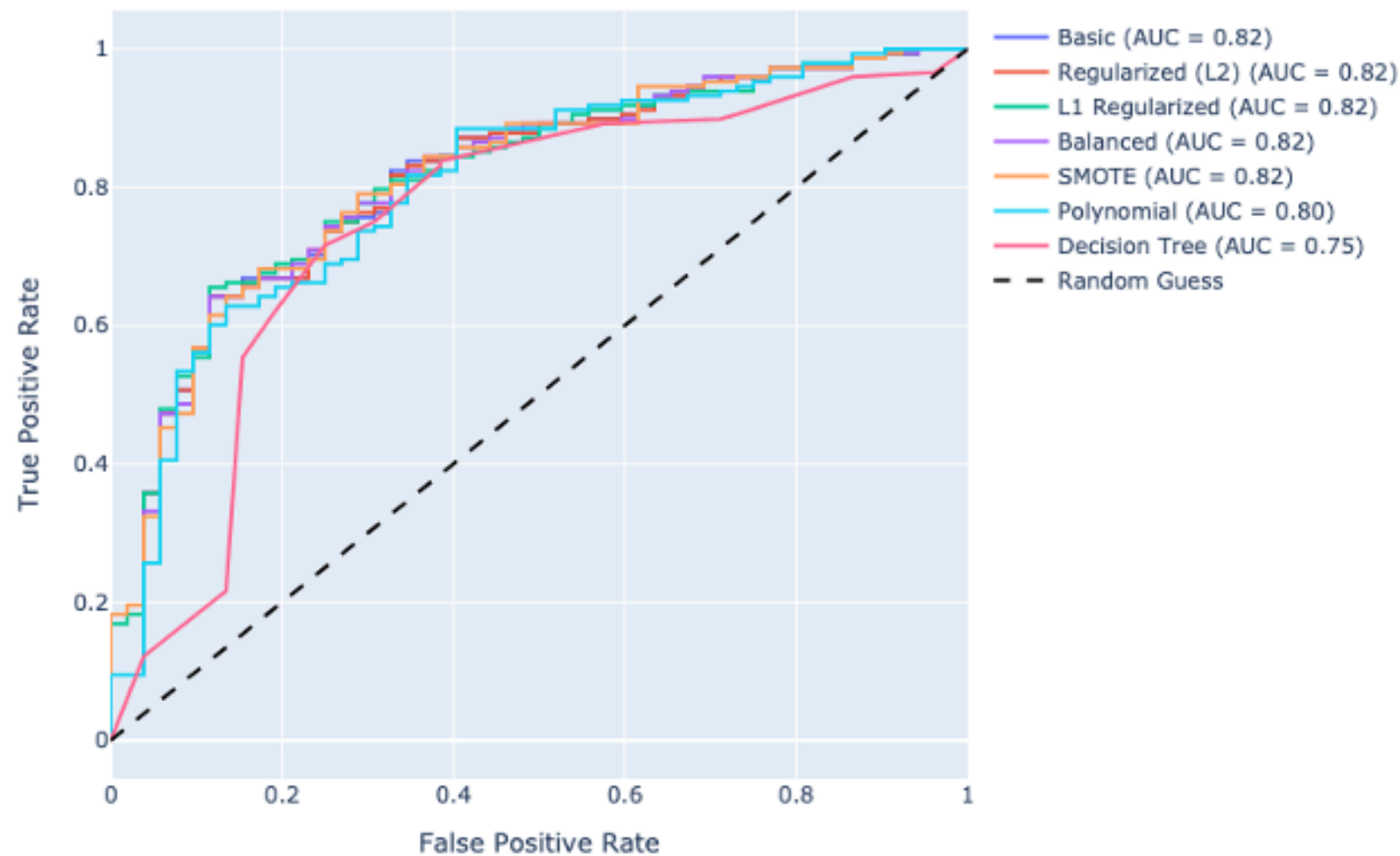


Figure 3: Distribution of pass/fail and multi-class labels across datasets.

MODEL PERFORMANCE (BINARY CLASSIFICATION)

ROC Curve (Simulated Binary)



ROC Curve (Real Binary)

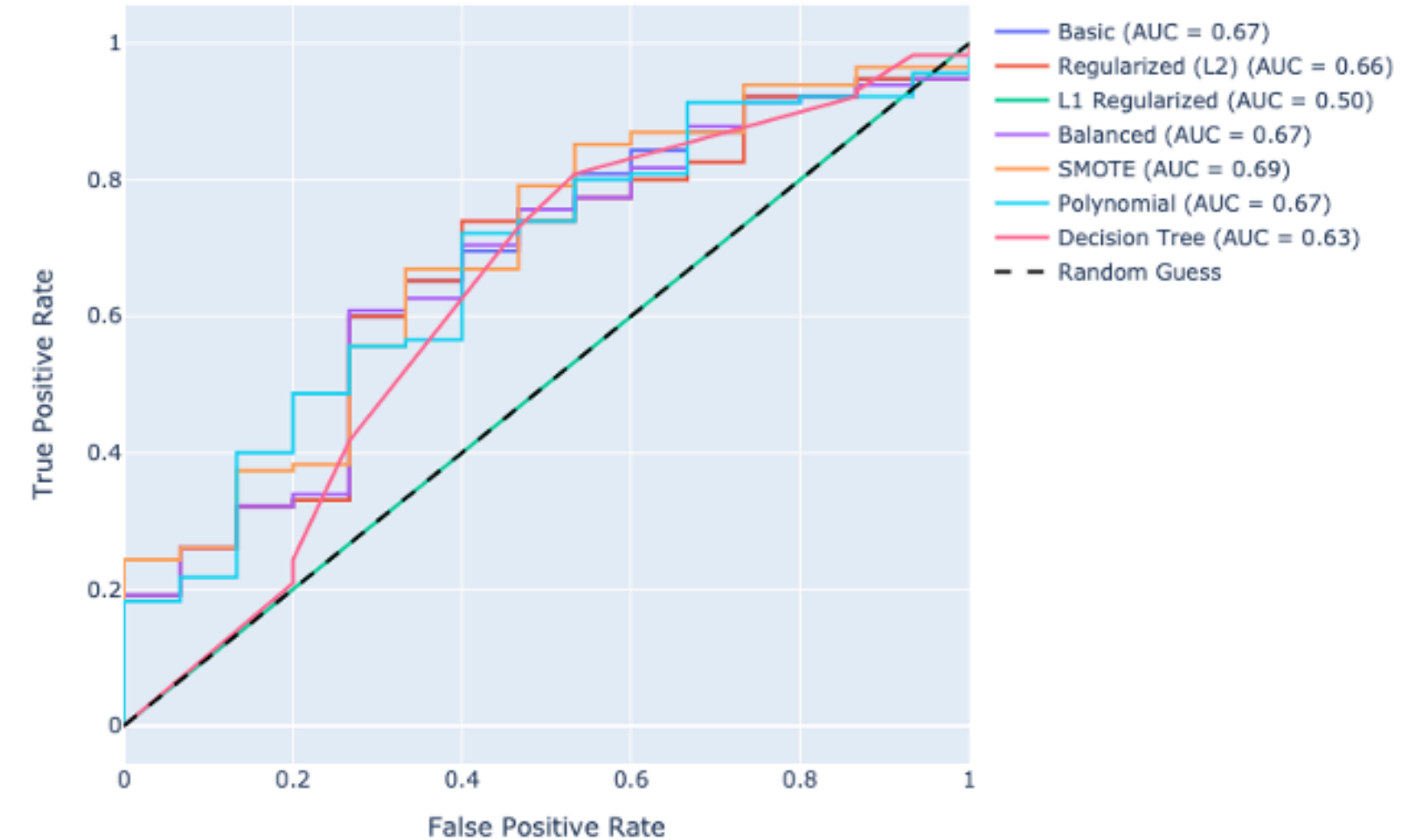


Figure 6: ROC curves for simulated data binary models.

Figure 7: ROC curves for real data binary models

Simulated Regularized (L2): Accuracy ~0.85, F1 ~0.84, ROC-AUC ~0.90.

Real SMOTE: Accuracy ~0.80, F1 ~0.78, ROC-AUC ~0.85.

MODEL PERFORMANCE (MULTI-CLASS CLASSIFICATION)

6.2 Multi-Class Classification

Model	Accuracy	F1-Score
Simulated Multi-Class Basic	0.980	0.980
Simulated Multi-Class Regularized (L2)	0.995	0.995
Simulated Multi-Class L1 Regularized	0.990	0.990
Simulated Multi-Class Balanced	0.945	0.946
Simulated Multi-Class SMOTE	0.950	0.951
Simulated Multi-Class Polynomial	0.985	0.985
Simulated Multi-Class Decision Tree	0.950	0.951

```
Simulated Multi-Class Regularized (L2) Results:  
Accuracy: 0.995  
F1-Score: 0.995  
ROC-AUC: N/A  
Confusion Matrix:  
[[ 29   0   0]  
 [  1  44   0]  
 [  0   0 126]]
```

```
Simulated Multi-Class Basic Results:  
Accuracy: 0.980  
F1-Score: 0.980  
ROC-AUC: N/A  
Confusion Matrix:  
[[ 29   0   0]  
 [  3  41   1]  
 [  0   0 126]]
```

KEY FINDINGS

- Top Predictor:
Study Hours (Real: 0.177, Simulated: 1.058)
- Class Imbalance: SMOTE improved simulated model performance
- Non-Linearity: Polynomial features enhanced accuracy

FEATURE IMPORTANCE

Feature Importance (Simulated Regularized L2)

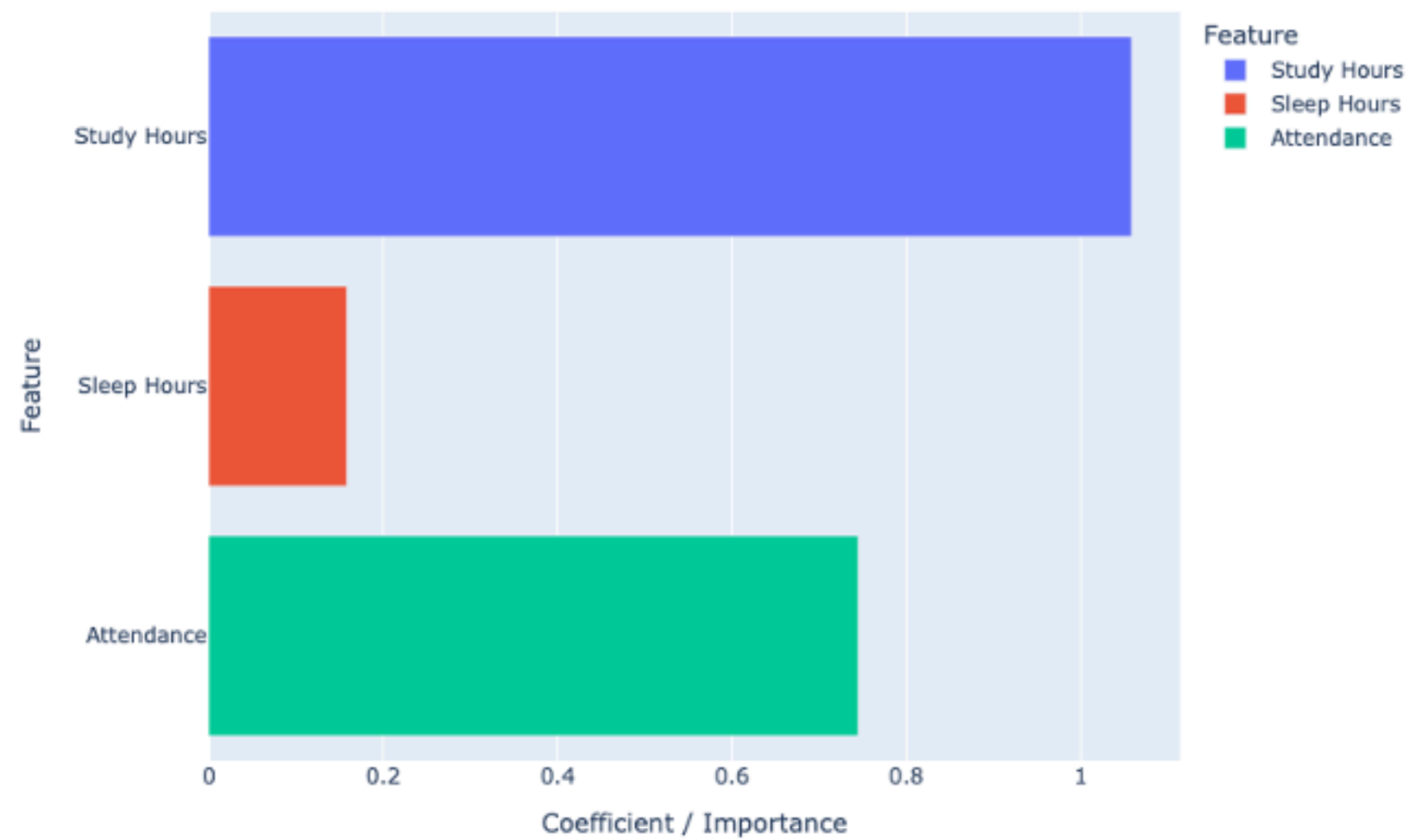


Figure 4: Feature importance for real data model.

Feature Importance (Real Regularized L2)

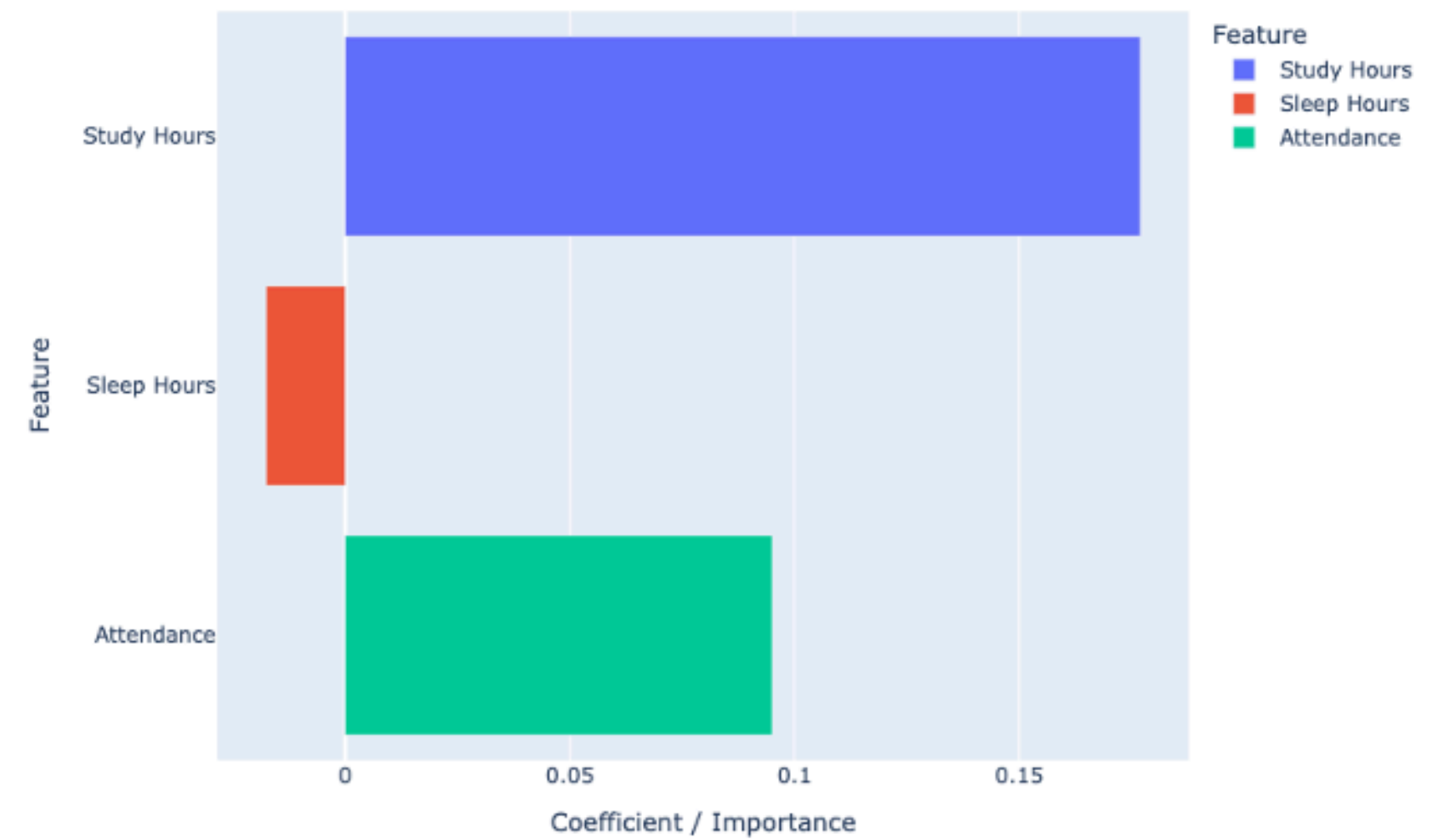
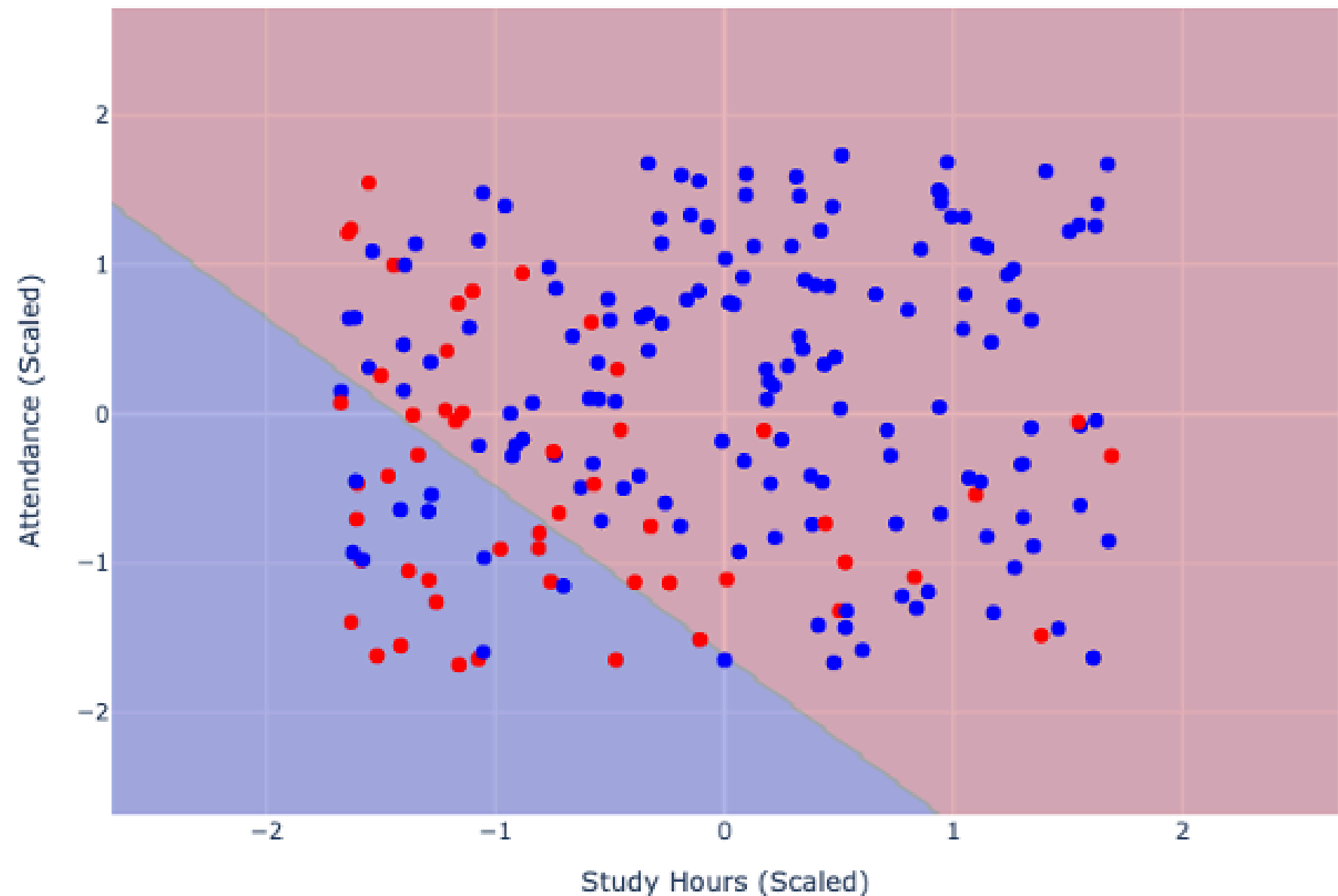


Figure 5: Feature importance for simulated data model.

INTERACTIVE VISUALIZATIONS

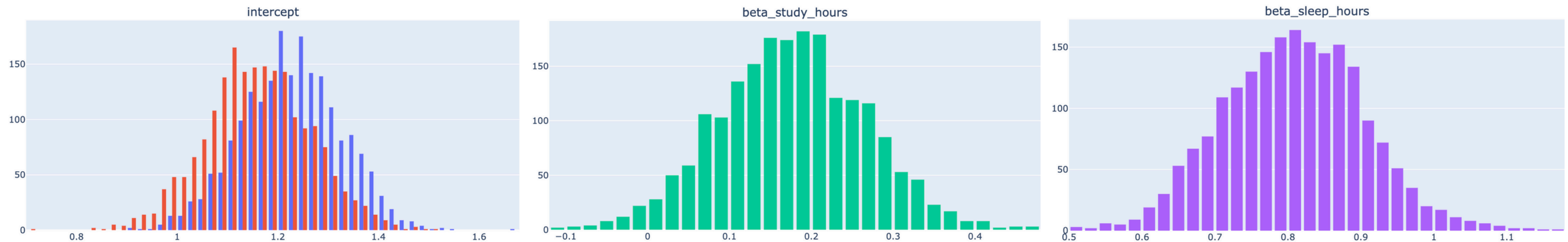
- Available Outputs:
 - Decision Boundaries (Binary & Multi-Class)
 - ROC Curves
 - Feature Importance
 - Correlation Heatmaps
 - 3D Feature Space



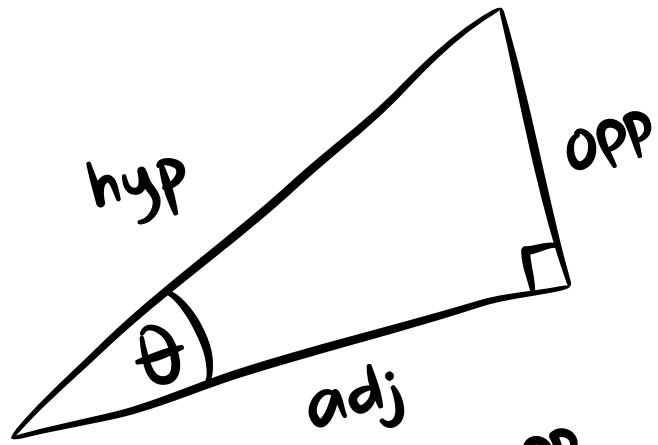
KEY INSIGHTS

- Feature Impact: Study Hours and Attendance are strong predictors
- Class Imbalance: SMOTE boosts performance, especially for imbalanced real data.
- Uncertainty: Bayesian models provide robust estimates (e.g., posterior distributions).

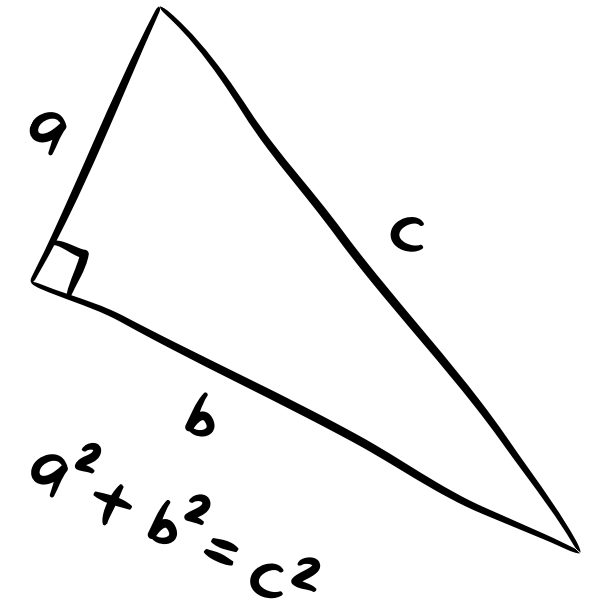
Bayesian Posterior Distributions



SUMMARY



$$\sin(\theta) = \frac{\text{opp}}{\text{hyp}}$$



1. Achievement: Best model (Simulated Multi-Class L2) achieved 0.995 accuracy
2. Key Takeaway: Logistic regression, enhanced by SMOTE and polynomial features, provides actionable insights for education.
3. Future Work: Explore ensemble methods (e.g., random forests), SVMs, or advanced neural networks.

$$y - y_1 = m(x - x_1)$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Thank You

