📊 Model Evaluation Report

This report presents a comparison of four supervised machine learning classifiers—Support Vector Classifier (SVC), Logistic Regression, Random Forest, and XGBoost—based on their performance across multiple evaluation metrics: Accuracy, Precision, Recall, and F1-Score.

1. Support Vector Classifier (SVC)

Accuracy: 80.71%

Class 0:

Precision: 0.79

Recall: 0.84

F1-Score: 0.82

Class 1:

Precision: 0.82

Recall: 0.77

F1-Score: 0.80

📌 Remarks: SVC shows balanced performance but slightly underperforms compared to ensemble methods. No hyperparameters were tuned, which might leave room for optimization.

2. Logistic Regression

Accuracy: 74.73%

Class 0:

Precision: 0.74

Recall: 0.77

F1-Score: 0.76

Class 1:

Precision: 0.75

Recall: 0.73

F1-Score: 0.74

📌 Remarks: Logistic Regression performs the lowest among the models. It's a good baseline but lacks the complexity needed for better generalization on this dataset.

3. Random Forest

Accuracy: 89.70%

Class 0:

Precision: 0.90

Recall: 0.89

F1-Score: 0.90

Class 1:

Precision: 0.89

Recall: 0.90

F1-Score: 0.90

Best Hyperparameters:

{

'max_depth': None,

'min_samples_split': 2,

'n_estimators': 300

}

📌 Remarks: Excellent balance between precision and recall for both classes. The hyperparameter tuning significantly boosted its performance.

4. XGBoost

Accuracy: 89.13%

Class 0:

Precision: 0.88

Recall: 0.91

F1-Score: 0.89

Class 1:

Precision: 0.90

Recall: 0.87

F1-Score: 0.89

Best Hyperparameters:

```
{
  'learning_rate': 0.1,
  'max_depth': 7,
  'n_estimators': 300,
  'subsample': 0.8
}
```

📌 Remarks: Comparable to Random Forest. XGBoost benefits from advanced boosting techniques, slightly favoring recall on class 0 and precision on class 1.

✅ Conclusion

Model	Accuracy	Avg F1-Score	Remarks
Random Forest	0.8970	0.90	Best performer overall
XGBoost	0.8913	0.89	Very close to RF, efficient
SVC	0.8071	0.81	Moderate, no tuning applied
Logistic Reg.	0.7473	0.75	Weakest, good baseline

📎 Recommendation: Random Forest and XGBoost are highly recommended for deployment, especially after hyperparameter optimization.