📊 Project Report: Model Evaluation for Customer Subscription Prediction

# 🔍 Objective

The objective of this project was to build a classification model to predict whether a customer would subscribe to a term deposit based on data from a direct marketing campaign of a Portuguese banking institution. The dataset presents class imbalance, with relatively few customers subscribing compared to the majority who do not.

# 📊 Model Performance Overview (Test Set)

|  |  |  |
| --- | --- | --- |
| Metric | Value | Interpretation |
| Accuracy | 0.6329 | The model correctly predicted ~63.3% of all test instances. However, in imbalanced datasets, accuracy can be misleading. |
| Precision | 0.1998 | Only ~20% of the positive predictions were correct. This means many customers were wrongly predicted to subscribe. |
| Recall | 0.7827 | The model correctly identified ~78% of actual subscribers — a high capture rate of positives. |
| F1 Score | 0.3183 | A balance between precision and recall. The moderate score reflects the high recall but low precision. |
| ROC AUC | 0.7440 | The model has a good ability to distinguish between subscribers and non-subscribers. Anything above 0.70 is considered respectable. |

# ⚖️ Summary of Trade-Offs

|  |  |
| --- | --- |
| Metric | Verdict |
| Accuracy | Moderate – might mislead in imbalanced data |
| Precision | Low – many false positives |
| Recall | High – captures most subscribers |
| F1 Score | Moderate – trade-off is visible |
| ROC AUC | Good – solid classification ability |

# 🧠 Suggestions for Improvement with more time

* Tune the classification threshold away from the default 0.5 to find a better balance between precision and recall.
* Experiment with different classifiers such as XGBoost or Gradient Boosted Trees, and use `class\_weight='balanced'` in models like Random Forest or Logistic Regression.
* Remove low-impact or noisy features and perform correlation analysis or use tree-based feature importance.
* Use probability calibration methods like Platt scaling or isotonic regression to improve the model’s confidence scores.
* Combine models (e.g., stacking, bagging) to improve generalization and stability.

# 🏆 Best Performing Model: Random Forest (Validation)

A tuned Random Forest classifier was ultimately the top-performing model with the following metrics:

|  |  |  |
| --- | --- | --- |
| Metric | Value | Verdict |
| Accuracy | 0.942 | Excellent |
| Precision | 0.9266 | Very High – low false positives |
| Recall | 0.960 | Very High – catches nearly all positives |
| F1 Score | 0.943 | Excellent balance |
| ROC AUC | 0.9874 | Outstanding separability |

This model achieved both high recall and high precision, indicating it correctly identified most subscribers while minimizing false alarms — ideal in real-world applications where contacting uninterested customers can be costly.

# 📁 Output Files Generated

* Predictions CSV: data/output/test\_predictions.csv
* ROC Curve: reports/figures/test\_model\_roc\_curve.png
* Confusion Matrix: reports/figures/confusion\_matrix.png
* Performance JSON (recommended to add): data/output/test\_metrics.json

# ✅ Conclusion

The initial model served well to highlight key issues around class imbalance and metric trade-offs. After tuning and model comparison, the Random Forest classifier emerged as the best option, offering both performance and reliability. With further threshold tuning and calibration, this model can be confidently deployed in production settings.