

Step 2: Advanced Modeling & MLflow Experiment Tracking

University: Tek-Up
Group: ING-4-SDIA

Tutor: Haythem Ghazouani

1 Introduction

Building a single model is rarely sufficient. In this tutorial, we explore the **Ensemble Learning** strategies discussed in the course notes ([Ensemble-Learning.pdf](#)). We will compare simple models with advanced Voting, Stacking, and Boosting.

2 Modeling Strategies

Strategy: 1. Simple Baselines

Start with classical models like **Logistic Regression** or **Decision Trees**. These provide a performance baseline. If an advanced ensemble doesn't beat these, the complexity is not justified.

Strategy: 2. Voting Classifier

Combines multiple models (e.g., Random Forest + XGBoost) and averages their predictions. **Soft Voting** uses probabilities, which often yields better results.

Strategy: 3. Stacking Classifier

Trains a set of base-models and uses their predictions as features for a **Meta-Learner** (usually Logistic Regression). This allows the model to learn which base-model is most reliable for different segments of data.

Strategy: 4. Boosting (XGBoost/LightGBM)

Sequential learning where each new model fixes the errors of the previous ones. This is the industry standard for high-performance tabular ML.

3 Tracking with MLflow

Every strategy mentioned above must be tracked as a separate "Run" in **MLflow**.

```
1 import mlflow
2 from sklearn.ensemble import VotingClassifier
3
4 mlflow.set_experiment("Bank_Churn_Ensembles")
5
6 # Define the ensemble
7 voting_model = VotingClassifier(estimators=[('rf', rf), ('xgb',
8     xgb)], voting='soft')
9
10 with mlflow.start_run(run_name="Voting_RF_XGB"):
11     pipeline.fit(X_train, y_train)
12     mlflow.log_metric("accuracy", accuracy_score(y_test, y_pred))
13     mlflow.sklearn.log_model(pipeline, "model")
```

4 Selecting the Best Model

After running your script, launch the UI:

```
1 mlflow ui
```

Sort your experiments by **F1-Score** or **Accuracy**. The model with the highest performance while maintaining low complexity will be selected for Step 3: Deployment.

5 Exercise

1. Implement a **StackingClassifier** using at least three different base-models.
2. Use MLflow to identify which model handles the minority class (`Churn=1`) the best.
3. **Question:** Why might a Stacking model overfit more easily than a simple Voting model?