Draft for Model Building Framework_For final presentation

Introduction

Based on the professor's guidance, our next steps in model development will focus on three key areas: **Model Framework Design**, **Model Optimization**, and **Model Explainability**. Below is a detailed breakdown of our proposed framework to guide the upcoming work.

Work Objectives

1.Refine the model based on three key aspects: Model Framework Design, Optimization, and Explainability**.**

2.define clear input and output elements **by next Tuesday** to support the UI team's development.

1. **Model Framework Design**
   a. **Goal**: Compare XGBoost and Random Forest (along with other strong classifiers) to select the best-performing model for predicting sales time.
   b. **Evaluation Metrics**:
      - **Classification Metrics**:
      - F1-Score: Balances precision and recall across all classes.
      - ROC-AUC: Measures the ability to distinguish between classes.
      - **Class-Specific Analysis**:
      - Focus on Recall and Precision for minority classes (e.g., slow sales).

2. **Model Optimization**
   - **Feature Engineering**:
   - Extract time-related features (e.g., sales duration, periodic patterns).
   - Apply log transformations for skewed numerical features (e.g., price).
   - Encode categorical variables using target encoding or one-hot encoding.
   - Generate interaction features to capture complex relationships (e.g., price × seller rating).
   - **Class Balancing**:
   - Use oversampling methods like SMOTE for underrepresented classes.
   - Experiment with undersampling majority classes.
   - Tune class weight parameters (e.g., scale_pos_weight in XGBoost).
   - **Hyperparameter Tuning**:
   - **For XGBoost**:
   - Learning Rate (eta): Optimize in the range of 0.01-0.1.
   - Tree Depth (max_depth): Adjust between 6-10 for balanced complexity.
   - Regularization (lambda, alpha): Fine-tune to avoid overfitting.
   - **Validation Strategy**:
   - Apply Stratified K-Fold Cross Validation to maintain consistent class distributions.
   - Evaluate model robustness using different train-test splits.

3. **Model Explainability**
   a. **Feature Importance Analysis**:
   - For **XGBoost**: Evaluate Gain (accuracy improvement), Weight (usage frequency), and Cover (sample impact).
   - For **Random Forest**: Assess Gini Importance or Permutation Importance.
   - Visualize feature importance with libraries like matplotlib or XGBoost's plot_importance.

   b. **SHAP Analysis**:
   - Use SHAP to provide consistent comparison of feature contributions across models.
   - Compare global feature impact between XGBoost and Random Forest.
   - Generate local explanations for specific predictions to evaluate interpretability.

   c. **Partial Dependence Plots (PDP)**:
   - Visualize how single features (e.g., price, feedback score) affect predictions globally.
   - Compare PDPs between XGBoost and Random Forest to detect consistent trends.

   d. **Interaction Effects**:
   - Leverage SHAP Interaction values to explore feature pair interactions.
   - Highlight any differences in how XGBoost and Random Forest handle interactions.

   e. **Tree Visualization**:
   - For Random Forest: Analyze decision paths from individual trees.
   - For XGBoost: Visualize key trees to understand split decisions.