**Phase 1: Project Planning & Business Alignment**

**Objective:** Define how the project solves a real-world business problem.

1. **Business Goal**:
   * *Problem*: The bank wants to reduce marketing costs by targeting high-propensity customers.
   * *Success Metric*: Increase precision (reduce false positives) while maintaining recall.
2. **ROI Calculation**:
   * Assume each call costs 5,andaconvertedcustomerbrings5,*andaconvertedcustomerbrings*100 profit.
   * Calculate cost savings if the model reduces unnecessary calls by 20%.

**Phase 2: Advanced Preprocessing & Feature Engineering**

**Objective:** Prepare data to uncover hidden patterns and improve model performance.  
**Tools**: Python, Pandas, Scikit-learn, Feature-engine.  
**Steps**:

1. **Data Cleaning**:
   * Handle "unknown" values (e.g., job=unknown) using **MICE imputation**.
   * Detect outliers with **Isolation Forest** and cap/remove them.
2. **Feature Engineering**:
   * Create interaction terms: duration / campaign (call efficiency).
   * Derive customer\_value\_score = (balance / age) + loan\_status.
   * Use **Target Encoding** for high-cardinality categorical variables (e.g., job).
3. **EDA**:
   * Plot **SHAP summary charts** to pre-identify impactful features.
   * Analyze correlation between duration and y (target) for ethical bias (longer calls may coerce subscriptions).

**Phase 3: Model Development & Optimization**

**Objective:** Build, tune, and compare models to maximize business metrics.  
**Algorithms**:

* **Baseline**: Logistic Regression, Naive Bayes, KNN, Decision Tree, Random Forest, XGBoost.
* **Advanced**: Stacked Ensemble (XGBoost + Logistic Regression), LightGBM.

**Steps**:

1. **Hyperparameter Tuning**:
   * Use **Optuna** for Bayesian optimization (focus on precision/recall trade-off).
   * Example: Tune XGBoost’s max\_depth, learning\_rate, and scale\_pos\_weight.
2. **Validation Strategy**:
   * **Time-based split** (if timestamp exists) to simulate real-world performance.
   * Stratified K-Fold to handle class imbalance.
3. **Performance Metrics**:
   * Primary: **Precision** (cost reduction), **AUC-ROC** (overall performance).
   * Secondary: F1-Score, Recall.

**Phase 4: Business Impact & Fairness**

**Objective:** Link model results to actionable insights and ethical AI.  
**Tools**: SHAP, AIF360, Tableau.  
**Steps**:

1. **Cost-Benefit Analysis**:
   * Use the confusion matrix to calculate:
     + Savings = (Reduced False Positives \* $5/call)
     + Profit Gain = (True Positives \* $100/customer)
2. **Bias Audit**:
   * Check if the model underperforms for age > 60 or job=unemployed using **AIF360**.
   * Mitigate bias with **reweighting** or **adversarial debiasing**.

**Phase 5: Deployment & Scalability**

**Objective:** Showcase engineering skills for recruiters.  
**Tools**: Docker, FastAPI, MLflow, AWS.  
**Steps**:

1. **Model Serving**:
   * Wrap the best model in a **FastAPI** endpoint with endpoints for prediction and SHAP explanations.
   * Containerize with **Docker** and deploy on **AWS EC2**.
2. **MLOps**:
   * Track experiments with **MLflow** (metrics, parameters, artifacts).
   * Automate retraining with GitHub Actions or **Airflow**.

**Phase 6: Presentation & Recruiter Appeal**

**Objective:** Create a portfolio-ready deliverable.  
**Tools**: GitHub, LinkedIn, Streamlit.  
**Steps**:

1. **GitHub Structure**:

bash

Copy

/bank-marketing-project

├── data/ # Raw and processed data

├── notebooks/ # EDA, modeling Jupyter notebooks

├── src/ # Modular scripts (preprocess.py, train.py)

├── deployment/ # Dockerfile, FastAPI code

├── docs/ # Business report, presentation slides

└── README.md # Project summary, visuals, video demo link

1. **Visual Storytelling**:
   * Build a **Streamlit dashboard** showing real-time predictions and SHAP force plots.
   * Create a **1-page PDF** summarizing ROI impact and fairness findings.
2. **LinkedIn Post**:
   * Title: *“Reducing Bank Marketing Costs by 18% with XGBoost & Ethical AI”*
   * Content:
     + Problem, solution, tools (Python, AWS, SHAP).
     + **Metrics**: AUC, precision, cost savings.
     + Link to GitHub/Streamlit demo.

**Implementation Roadmap**

| **Week** | **Task** |
| --- | --- |
| 1-2 | Data cleaning, advanced EDA, feature engineering. |
| 3-4 | Model training, hyperparameter tuning, fairness audit. |
| 5 | ROI calculation, Docker deployment, FastAPI setup. |
| 6 | Streamlit dashboard, GitHub/docs polish, LinkedIn post. |

**Recruiter Keywords to Highlight**

* **Technical**: Hyperparameter tuning (Optuna), MLOps (MLflow, Docker), Feature Engineering (SHAP).
* **Business**: ROI optimization, cost-benefit analysis, bias mitigation.
* **Tools**: Python, AWS, XGBoost, Scikit-learn.

This plan balances technical depth with business storytelling, ensuring your project stands out to recruiters and solves real-world problems! 🚀