Absolutely! Here’s a **comprehensive, production-quality documentation** for your **Modeling Notebook**, ready to copy into your report file or project docs. This is written for both technical and business audiences, includes standard best practices, and captures the full modeling workflow, logic, findings, and recommendations.

# **📑 Modeling Notebook Summary: Healthcare Fraud Detection**

## **1. Overview**

This notebook covers the full modeling phase of our provider-level healthcare fraud detection project, leveraging engineered features from previous steps. The goal is to **identify providers likely to commit fraud**, prioritizing models and metrics that maximize the catch rate (recall) for the fraud class.

## **2. Modeling Workflow & Logic**

### **2.1. Model Selection**

We systematically tested a range of industry-standard machine learning models:

* **Logistic Regression (Spark ML):** Interpretable baseline for tabular data.
* **Decision Tree (Spark ML):** Baseline tree for non-linear relationships.
* **Random Forest (Spark ML):** Ensemble method known for strong tabular performance.
* **XGBoost (scikit-learn API):** Powerful boosting method, industry gold standard for tabular, imbalanced problems.

Each model was run on the same engineered feature set for fair comparison. For Random Forest, we conducted grid search cross-validation for hyperparameter tuning (numTrees, maxDepth).

### **2.2. Handling Class Imbalance**

* **No undersampling or oversampling** (to preserve all available data).
* **Class weighting:**
  + For Logistic Regression and XGBoost, we used built-in parameters (class\_weight='balanced' or scale\_pos\_weight) to account for the significant class imbalance (only ~9% of providers labeled as fraud).
  + Decision Tree and Random Forest used default Spark settings, which do not automatically apply class weighting, but we compared their performance.

### **2.3. Evaluation Metrics**

All models were evaluated using:

* **AUC-ROC (Area Under the ROC Curve):** Main measure of model discrimination.
* **Confusion Matrix:** For visualizing the trade-off between true/false positives/negatives.
* **Recall (Fraud class):** Priority metric—to minimize undetected fraud.
* **Precision (Fraud class):** To limit unnecessary investigations.
* **F1-score:** Harmonic mean of precision and recall.
* **Accuracy:** Reported, but not the main focus due to class imbalance.

All results and artifacts were tracked and logged with **MLflow** for reproducibility.

## **3. Key Findings & Statistics**

### **3.1. Model Performance Table**

| **Model** | **AUC-ROC** | **Fraud Recall** | **Fraud Precision** | **Fraud F1** | **Accuracy** |
| --- | --- | --- | --- | --- | --- |
| Logistic Regression | 0.955 | 0.55 | 0.79 | 0.65 | 0.94 |
| Decision Tree | 0.654 | 0.54 | 0.47 | 0.51 | 0.90 |
| Random Forest | 0.956 | 0.57 | 0.76 | 0.66 | 0.94 |
| RF (Grid Search) | 0.954 | 0.56 | 0.80 | 0.66 | 0.95 |
| XGBoost | 0.949 | 0.76 | 0.53 | 0.62 | 0.91 |

### **3.2. Key Observations**

* **XGBoost achieved the highest recall (0.76) for fraud class**, making it best suited for business needs where catching as many fraud cases as possible is paramount.
* **Random Forest (tuned) and Logistic Regression** delivered strong, balanced performance, with slightly higher precision (fewer false alarms).
* **Decision Tree underperformed**—expected for single trees on complex, high-dimensional data.
* **All top models (AUC ~0.95) are production-grade** and generalize well, as confirmed by cross-validation and validation set metrics.
* **Class weighting proved effective**—particularly for XGBoost—at handling severe imbalance.

## **4. Conclusions & Recommendations**

* **XGBoost is recommended as the primary model** for deployment, due to its superior fraud recall (catch rate). In high-risk settings like insurance fraud, recall is typically prioritized over precision.
* **Random Forest (with grid search tuning) is recommended as a robust secondary model**—with slightly better precision, it’s well-suited for scenarios where limiting manual reviews is more important.
* **Logistic Regression offers interpretable, competitive results**—a strong baseline or alternative for settings requiring maximum explainability.

All models and results are **fully tracked in MLflow** for reproducibility, auditability, and future monitoring.

## **5. What’s Missing / Future Enhancements**

* **Interpretability:** Next, apply SHAP or feature importance analysis to explain model decisions and support business/ethical transparency.
* **Post-deployment monitoring:** Implement a process to monitor recall, precision, and drift after deployment.
* **Further ensemble testing:** Although not necessary for current performance, advanced ensembling (stacking/voting) can be explored for further incremental gains.
* **Automated retraining pipeline:** For production, set up regular retraining as more labeled data becomes available.

## **6. Summary Statement**

**This modeling notebook demonstrates a rigorous, auditable, and business-aligned workflow for healthcare fraud detection.  
 XGBoost was selected for its high recall, aligning with the goal of minimizing undetected fraud.  
 All artifacts, code, and results are production-ready and support future extension for interpretability, deployment, or advanced modeling.**

**End of Modeling Notebook Documentation**

*Save this in your docs or as a final markdown cell in your notebook. It covers what you did, why, findings, stats, and recommendations per data science best practices.*