HOSPITAL READMISSION PREDICTION FOR DIABETIC PATIENTS

Github Repo : Final Assessment Task

By Kuldeep Yadav

PROBLEM & OBJECTIVE

PROBLEM

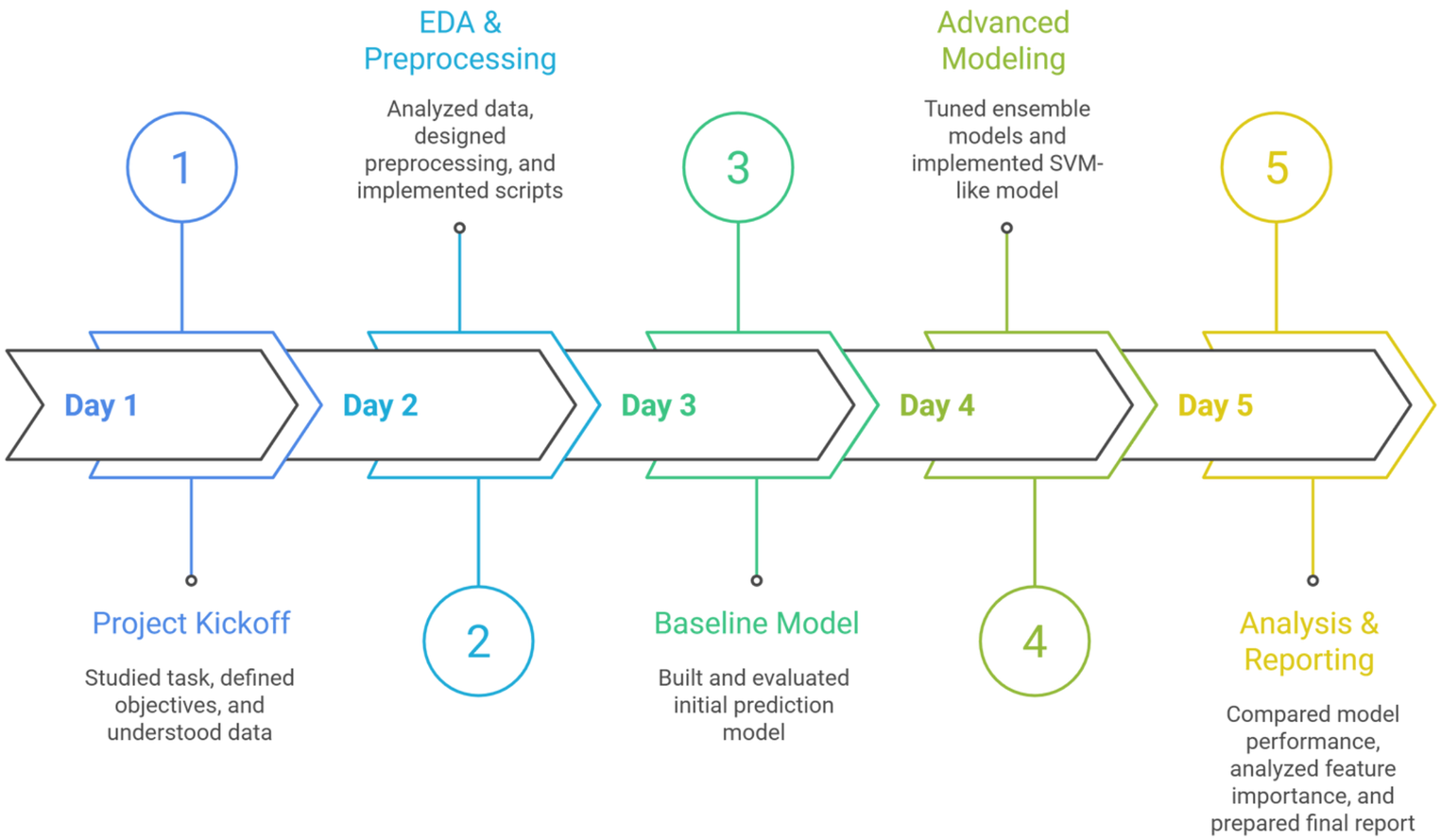
MediSure Hospitals want to use Machine Learning to predict whether a patient will be readmitted within 30 days of discharge.

BUSINESS IMPORTANCE

- Reduce preventable readmissions and penalties
- Allocate follow-up resources to highest-impact patients
- Improve care continuity and patient outcomes

GOALS & TIMELINE

Hospital Readmission Prediction Project Timeline



DATASET OVERVIEW

EDA

- Loaded the diabetic encounters dataset and mapped the target to binary (readmit <30 days = 1, else 0). Replaced "?" with NaN.
- Major missingness: weight (mostly missing), max_glu_serum and A1Cresult (~80% missing) — handled as special/missing categories.
- Target is highly imbalanced (few readmitted cases).
- Many numeric features (num_lab_procedures, num_medications, number_outpatient/number_emergency/number_inpatient) are right-skewed with outliers: time_in_hospital is compact with a few extremes.
- Individual medication flags show near-zero linear correlation with readmission (insulin is the only weak positive signal).
- Correlation matrix and PCA/pairplots show weak linear signals and no clear class separation — non-linear effects or interactions likely drive readmission.

PREPROCESSING

1. Missing & Irrelevant Data Handling
   - Replaced '?' with NaN.
   - Dropped identifiers (encounter_id, patient_nbr) and high-null columns (weight).
   - Removed constant-value columns (examide, citoglipton).
   - Retained clinically relevant columns (max_glu_serum, AlCresult) and imputed missing values with 'None'.

2. Missing Value Treatment
   - Filled categorical missing values using appropriate strategies:
     - race → Mode
     - max_glu_serum, AlCresult → 'None'
     - payer_code, medical_specialty, diag_1-3 → 'Unknown'

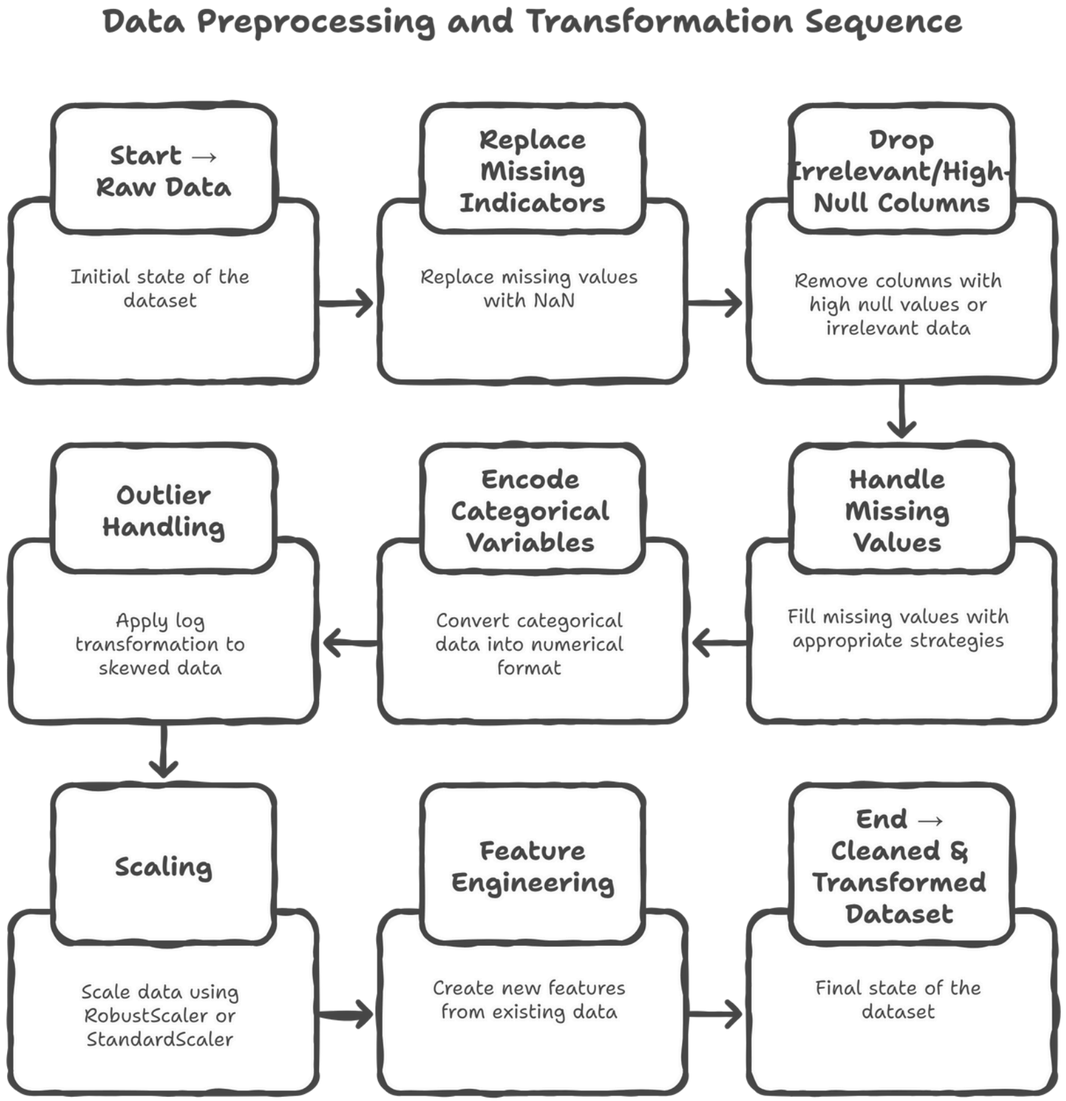
3. Encoding Categorical Variables
   - Target Encoding: readmitted → {'NO', '>30': 0, '<30': 1}
   - Label Encoding: gender, race, medical_specialty, payer_code, max_glu_serum, AlCresult.
   - Medication-related features encoded ordinally (No < Down < Steady < Up).

PREPROCESSING

4. Outlier & Skewness Handling
   - Applied log transformation on key numeric columns (hospital time, labs, meds, etc.) to reduce skewness and handle outliers.

5. Feature Scaling
   - Standard Scaling for non-tree models.
   - Robust Scaling for tree-based models.

6. Feature Engineering
   - Discretized age into meaningful bins.
   - Grouped diagnosis codes (diag_1–3) by disease category.
   - Created new features: total_medications and num_med_changes.



ML MODEL

Pipeline Overview:

- Preprocessing → Train/Test Split → Optional SMOTE
- Scaling applied (Robust for tree models, Standard for non-tree)
- Model Training → Performance Evaluation

Evaluation Metrics & Tools:

- ROC-AUC
- Confusion Matrix
- Accuracy
- Classification report
- Visual analysis via ROC & CM plots

Models Tested:

- Logistic Regression (Baseline)
- Decision Tree (Baseline)
- Support Vector Classifier
- Random Forest (Tuned)
- XGBoost (Tuned)

Key Insights:

- RF & XGBoost tuning was time-intensive but improved performance

Reproducibility:

- Custom src/utils modules used for consistent preprocessing, model training, and evaluation

MODEL TRAINING

Data Loading: Import dataset using custom utility load_data()

Feature & Target Split: Separate predictors and target (readmitted)

Train-Test Split: Ensure unbiased evaluation

Pipeline Building:

Scaler: StandardScaler (non-tree) / RobustScaler (tree)

Optional SMOTE for class imbalance

Model: Logistic Regression / Random Forest / XGBoost / SVR

Training: Fit model with preprocessing handled automatically

Prediction: Generate classes and probabilities

Evaluation: Compute metrics via evaluate_model()

Model Saving: Export trained pipeline as .pkl for reuse

MODEL COMPARISON

Model	F1 Score	ROC-AUC	Precision	Recall	Accuracy	Confusion Matrix
Logistic Regression	0.2526	0.6409	0.8341	0.6356	0.6356	[[11685, 6384], [1032, 1253]]
Decision Tree Classifier	0.1679	0.5275	0.8108	0.7907	0.7907	[[15663, 2406], [1855, 430]]
Random Forest Classifier	0.0329	0.6445	0.8421	0.8875	0.8875	[[18025, 44], [2246, 39]]
Tuned Random Forest	0.2763	0.6742	0.8393	0.6971	0.6971	[[13012, 5057], [1108, 1177]]
XGB Classifier	0.0306	0.6701	0.8496	0.888	0.888	[[18038, 31], [2249, 36]]
Tuned XGB Classifier	0.107	0.6318	0.8338	0.8827	0.8827	[[17823, 246], [2142, 143]]

CONCLUSION

- Key outcome
  - Best model: Tuned Random Forest (best F1 ≈ 0.276, ROC-AUC ≈ 0.67) — improved minority detection but room to improve.
- Main predictive signals
  - Prior utilization (number_inpatient) and discharge destination (discharge_disposition_id) are the dominant predictors.
  - Secondary: num_lab_procedures, num_medications, time_in_hospital, age/payer/specialty.
  - Individual drug flags contribute very little — aggregate medication features work better.
- Limitations
  - Strong class imbalance and weak linear separability of features.
  - Many skewed numeric features and sparse medication columns → noise risk.
  - Need external / temporal validation before deployment.