

Pipeline card

From analysis table to fitted model in eight steps, with the leakage checks.

```
X = df[features]; y = df["missed_anc4"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)

pre = ColumnTransformer([("num", StandardScaler(), num_cols), ("cat",
OneHotEncoder(handle_unknown="ignore"), cat_cols)])

pipe = Pipeline([("pre", pre), ("model", LogisticRegression(max_iter=1000))])

cross_val_score(pipe, X_train, y_train, cv=5, scoring="average_precision")

pipe.fit(X_train, y_train); scores = pipe.predict_proba(X_test)[:, 1]
```

Step	Rule
1 Build X and y	Every feature known on the morning of the decision, label rule fixed from the guideline
2 Split first	Stratified, or by time for routine data, test set locked
3 Preprocess inside the pipeline	Learned from training rows only
4 Baseline rule	The sticky note rule the model must beat
5 Cross validate inside training	Choose features and settings here
6 Fit the final pipeline	On the whole training set
7 Open the test set once	Report the six numbers
8 Leakage checks	Read the features aloud, drop each in turn, distrust near perfect

Based on the ERAS Africa Machine Learning in Data Analysis course, 2026, the WHO guidance on ethics and governance of artificial intelligence for health, TRIPOD+AI, PROBAST and the scikit-learn documentation. Free to copy and adapt with attribution.