CMPE 255 - Assignment 4: CRISP-DM, KDD, SEMMA - Report

Author: Shilpa Yelkur Ramakrishnaiah

SJSU ID: 019151782

CRISP-DM Project Report — Walmart Weekly Sales Forecasting

Critique → revise loop

- **Before/After** cells per phase and "**What changed after critiquing**" notes.
- Persona prompt: *"You are a world-renowned KDD authority and award-winning author…"*

0) Executive summary

- **Business goal:** Forecast **weekly sales** at the (Store, Dept) level so planning can reduce stockouts and improve staffing.
- **Dataset:** Walmart weekly sales (loaded from `data/train.csv`). Key columns used across phases: **`Date`**, **`Store`**, **`Dept`**, **`IsHoliday`**, **`Weekly_Sales`**.
- **Success criteria (from your Phase 1):**
  - Reduce **RMSE** ≥ **15%** vs. a manual/naive baseline.
  - Provide a model that **updates weekly** and is easy to run locally.
- **Approach:** Full CRISP-DM flow with a critique→revise loop. Baselines first (naive lag), then **Linear Regression** and a **Fast Random Forest** trained with **time-series cross-validation** and a **chronological hold-out**.

---

1) Business Understanding (Phase 1)

From your notebook's Phase 1 markdown:

- **Objective:** “Predict future weekly sales for Walmart stores.”
- **KPIs:** Improve RMSE ≥ **15%** vs last year’s manual baseline; a model that **refreshes weekly**.
- **Constraints:** Data is **weekly**; solution must **run locally**.

---

2) Data Understanding (Phase 2)

- **Load:** `df = pd.read_csv("data/train.csv")`
- **Core fields present:** `Date` (parsed), `Store` (cat), `Dept` (cat), `IsHoliday` (bool), `Weekly_Sales` (target).
- **Sanity checks:** dtypes, missingness, simple histograms for `Weekly_Sales`, categorical level counts for `Store/Dept`.
- **Temporal checks: weekly completeness** (no missing weeks per panel), duplicate (Store, Dept, Date) guards.
- **Outcome:** Clear picture of volume/quality, ready for time-aware prep.

---

3) Data Preparation (Phase 3)

Key transformations

- **Date parsing** → `Year`, `Month`, `Week`, `Quarter`.
- **Seasonality encodings:** `Month_sin/cos`, `Week_sin/cos`.
- **Lag/rolling features:** **`lag1`**, **`roll4`**, **`roll12`** (computed **within** each (Store, Dept) panel to avoid leakage).
- **Target:** `Weekly_Sales`.
- **Boolean passthrough:** `IsHoliday`.

Leakage control

- Lags/rolls are built using **only prior periods** (no forward info).
- NAs introduced by lagging handled appropriately (e.g., train masks).

The result is a tidy `df_prep` with a **panel-aware** feature set aligned to downstream pipelines.

---

4) Modeling (Phase 4)

Splits

- **Chronological hold-out**: last **~90 days** for **Test**, with a fallback to ensure ≥ 4 weeks of test data.
- **TimeSeriesSplit** (`tscv`) for **cross-validation** on the training window.

Baseline

- **Naive lag-1** forecast (where available) reported with **RMSE/MAE** → serves as the KPI reference (“improve ≥ 15%”).

Pipelines

- **Linear Regression** pipeline
  - Numerics: `SimpleImputer(median)` → `StandardScaler`
  - Categoricals: `SimpleImputer(most_frequent)` → `OneHotEncoder(handle_unknown="ignore")`

- **Fast Random Forest** pipeline
  - Numerics: `SimpleImputer(median)` (no scaler)
  - Categoricals: **same** one-hot pipeline
- **Bool features** (e.g., `IsHoliday`) **passthrough**.

Metrics (you compute and display)

- **CV:** `neg_root_mean_squared_error` and `neg_mean_absolute_error` with `TimeSeriesSplit`.
- **Hold-out: RMSE** and **MAE** on the **chronological test split**.
- Calculate **ΔRMSE_vs_Base_%** to show percent improvement vs. the naive baseline.

---

5) Evaluation (Phase 5)

- **Consistency:** Test split matches Phase 4 (last ~90 days).
- **KPIs:** Primary = **RMSE reduction** vs baseline; also MAE.
- **Error analysis suggestions (and some implemented):**
  - Residual histogram (already present).
  - **Breakdowns by Store/Dept** to identify underperforming segments.
  - Guardrails: minimum Test length (≥ 4 weeks), and checks to ensure no leakage warnings were tripped.

KDD Project Report — Telco Customer Churn

Critique → revise loop

- **Before/After** cells per phase and "**What changed after critiquing**" notes.
- Persona prompt: *"You are a world-renowned KDD authority and award-winning author…"*
- Revisions pushed you to: stratified CV with PR-AUC, strict schema & ID handling (`customerID`), probability calibration, and cost-sensitive thresholding with decision curves.

0) Executive summary

- **Objective:** Predict **customer churn** so retention can prioritize outreach and incentives.
- **Dataset:** Telco Customer Churn (Kaggle/UCI style). Target `Churn ∈ {Yes, No}` → `{1, 0}`. `customerID` is an identifier and must not be modeled.
- **Business success criteria:**
    1. Lift in top deciles (e.g., top 10% captures a disproportionate share of churners).
    2. High **PR-AUC** (primary), stable **ROC-AUC** (secondary).
    3. **Calibrated** probabilities for cost/benefit thresholding and budget planning.
- **Champion model: Calibrated Random Forest** selected by **Stratified 5-fold CV** optimizing PR-AUC; **Calibrated Logistic Regression** retained as interpretable baseline.
- **Validation:** Hold-out **test** with bootstrap CIs, calibration reliability, decision curves, decile lift, and subgroup robustness.

KDD methodology

KDD = **Selection** → **Preprocessing** → **Transformation** → **Data Mining** → **Interpretation/Evaluation**.

1) Selection

Goal: Choose relevant data, define target, and ensure splits reflect the real population.

- **Loaded** raw Telco data; coerced types; trimmed whitespace.
- **Canonicalized target:** `Churn` → binary `y` (Yes=1, No=0); reported churn prevalence.
- **Identifier handling:** dropped/ignored `customerID`.
- **Sanity on numeric quirks:** `TotalCharges` coerced to numeric with `errors='coerce'` (e.g., tenure==0 rows → NA then impute or treat explicitly).
- **Duplicates:** checked for duplicate `customerID` rows; reported counts.
- **Initial EDA:** histograms / crosstabs on impactful variables (tenure, contract type, payment method, tech support).
- **Split:** Train/Validation/Test with **Stratified** sampling on the target (to preserve churn rate).

Acceptance gates

- Target prevalence within ±1–2 percentage points across splits.
- No heavy categorical drift across splits (flag relative shifts > 20% on major levels).
- Numeric KS tests vs. population: p > 0.05 (warn otherwise).
- Dupes on `customerID` = 0 (or documented and removed).

---

2) Preprocessing

Goal: Clean and standardize data to a consistent, model-ready shape.

- **Missingness audit** and imputation plan:
  - **Numerics:** median impute.
  - **Categoricals:** most-frequent/“Unknown” fallback (explicitly allowed at inference).
- **Type fixes:**
  - `SeniorCitizen` treated as **categorical** (even if stored numeric).
  - `TotalCharges` coerced to float; tenure==0 rows handled gracefully.
- **Categorical hygiene:**
  - Trimmed whitespace; normalized values; merged extremely rare levels into `__rare__` to stabilize one-hots.
- **Schema consistency:** kept a **feature list** to enforce at inference; unknown categories allowed without crashing.

---

3) Transformation

Goal: Encode/scale features so models learn signal efficiently.

- **ColumnTransformer** inside a **Pipeline**:
  - **Categoricals:** `OneHotEncoder(handle_unknown='ignore')` (and optional K-fold **target encoding** for specific high-cardinality columns if needed; fitted in CV space only).
  - **Numerics:** `RobustScaler` or `StandardScaler` post-imputation (you kept it version-compatible).
- **Variance filter:** removed near-zero variance features after encoding.
- **Feature documentation:** emitted feature order/metadata (optional `features.json`).

---

4) Data Mining

Goal: Train/tune models with the right objective and guardrails.

- **CV design: StratifiedKFold (k=5)** with **PR-AUC** scorer (best for imbalanced churn).
- **Candidates:**
  - **Logistic Regression** (elastic-net grid over C & l1_ratio; solver set based on sklearn version).
  - **Random Forest** (n_estimators, max_depth, min_samples_leaf, max_features; class_weight balanced_subsample).
- **Selection rule:** choose champion by **CV PR-AUC** (ties broken by stability/variance and interpretability).
- **Calibration:** `CalibratedClassifierCV` (isotonic) to get reliable **probabilities**.
- **Interpretability:** coefficients (LR) & permutation importances (RF).

---

5) Interpretation / Evaluation

Goal: Validate on untouched test; explain results for decisions.

- **Hold-out test** metrics:
  - **PR-AUC** (primary) and **ROC-AUC** with **bootstrap 95% CIs**.
  - **Calibration**: Reliability curve, Brier score, ECE.
  - **Confusion matrices** and classification report at:
    - 0.50,
    - **cost-optimized threshold** (minimizes expected FP/FN campaign cost),
    - **F2-optimal** threshold (recall-favored).

- **Lift analysis:** decile table showing responders captured in top X%.
- **Decision curve analysis:** net benefit vs. treat-all / treat-none across thresholds.
- **Subgroup robustness:** PR-AUC by key segments (contract type, payment method, tenure bands).

SEMMA Project Report — Bank Marketing Response Modeling

Critique loop (how it was integrated)

For each phase, I captured **BEFORE** code, then used a persona prompt like:

> "You are a world-renowned SEMMA authority and award-winning author. Audit my SEMMA **{phase}** section. Identify methodological issues, data leakage, validation flaws, or weak business ties. Provide concrete code diffs and acceptance gates. Be ruthlessly practical."

I then incorporated the critique into **REVISED** code and documented **"What changed and why"**.

0) Executive summary

- **Objective:** Predict which prospects will respond ("y = yes") to a telemarketing offer so the team can **prioritize calls** and **improve ROI**.
- **Dataset:** UCI Bank Marketing ("bank-additional-full.csv"; semicolon-delimited). Target `y ∈ {yes,no}` mapped to {1, 0}.
  Leakage note: classic leak features such as `duration` (call length) must be **excluded** from training and inference because they are known only *after* the call.
- **Business success criteria:**
  1. **Improve PR-AUC** vs. naive baseline (positive rate).
  2. Provide **calibrated probabilities** to support budget-aware thresholding.
  3. Demonstrate **lift** in top deciles (e.g., top 10% captures a disproportionate share of responders).
- **Model:** Calibrated Random Forest (champion) selected via stratified CV optimizing **PR-AUC**, with isotonic calibration on validation folds. Logistic Regression retained as an interpretable baseline.

- **Validation:** Hold-out test set with **bootstrap CIs** for ROC-AUC/PR-AUC, **decision curve analysis** (net benefit), **calibration reliability** (Brier/ECE), and **decile lift**.

---

1) Sample

Goal: Create training/validation/test splits that **preserve class balance and key feature distributions**.

What was done

- Loaded `bank-additional-full.csv` (semicolon-delimited), mapped `y` to {0,1}.
- **Stratified split** (Train/Valid/Test; e.g., 60/20/20) to preserve the positive rate.
- **Representativeness checks**:
    - **Target rate parity** across splits within a small tolerance.
    - **Categorical proportion drift** vs. full population (flag large deviations).
    - **Numeric distribution differences** using **KS tests**.
    - Quick **bar charts/histograms** per split for high-impact features.

Acceptance gates

- Absolute difference in target rate between Train and Test ≤ 1.5 pp.
- For high-cardinality categoricals, any level shift ≤ 20% relative difference (warn otherwise).
- Numeric KS p-value > 0.05 (warn if ≤ 0.05).
  Outcome: Splits **PASS** with a few benign WARNs documented.

---

2) Explore

Goal: Understand quality, signal, and risks (esp. **leakage**).

Highlights

- **Missingness audit:** overall ≤ few percent; imputable with median (nums) / most-frequent (“unknown”) (cats).
- **Univariate** summaries for top drivers; **bivariate** checks with target.

- **Correlation / Mutual Information:** identify redundancy; prevent double-counting.
- **Leakage screening:** `duration` (call length) and sometimes `poutcome` can leak post-contact information.
  → **Decision:** Drop `duration` (and drop/limit post-campaign info depending on your cohort definition).

---

3) Modify

Goal: Prepare features robustly and **encode** them in a way that generalizes.

Pipeline (sklearn `ColumnTransformer` inside a `Pipeline`)

- **Categorical:**
  - Rare-level merging (e.g., levels <1% → `__rare__`).
  - One-Hot Encoding (handle unknown at inference).
  - Optional **K-fold target encoding** for specific high-cardinality features (fit only on Train folds to avoid leakage).
- **Numeric:**
  - Median impute → **RobustScaler**.
  - **Winsorization** for flagged heavy-tailed features (documented thresholds).
- **Leak control:** `duration` explicitly **excluded** from feature list.
- **Validation checks:** "unknown rate" by column, post-transform dimensionality, transform latency.

Artifacts: `preproc.joblib` (usually bundled with the model), `features.json` (optional list of training columns for schema checks).

---

4) Model

Goal: Train competing models, perform **stratified CV** with PR-AUC scoring, calibrate probabilities, and select a champion.

Candidates & search

- **Logistic Regression** (elastic-net path over `C`, `l1_ratio`).
- **Random Forest** (n_estimators, max_depth, min_samples_leaf, max_features, bootstrap).

- Cross-validation: **StratifiedKFold (k=5)**; scorer = **PR-AUC**.
- **Calibration:** `CalibratedClassifierCV` (isotonic) on CV splits.

Selection

- Chosen **champion** = Calibrated Random Forest (better PR-AUC + lift; stable across folds).
- **Baseline** = Calibrated Logistic Regression (interpretable; used for sanity checks).

Thresholding

- **Cost-sensitive threshold**: Choose `t*` that maximizes expected **net benefit** based on campaign economics (e.g., call cost vs. conversion value).
- Also report **F2-optimal** threshold when recall is prioritized.

---

5) Assess

Goal: Present **reliable performance** on the hold-out test set and demonstrate **business value**.

Metrics (Test set)

- **ROC-AUC** and **PR-AUC** with **bootstrap 95% CIs** (e.g., 1,000 resamples).
- **Calibration quality**: Brier score, ECE, reliability curve.
- **Confusion matrix** at `t*` and at 0.50; precision/recall with **Wilson CIs**.
- **Permutation importance** on Test for sanity (ensure drivers make business sense).

Business framing

- **Lift table** (deciles): share of responders captured by top X% targeted.
- **Decision curve analysis**: net benefit vs. treat-all / treat-none across thresholds.
- **Subgroup robustness**: PR-AUC ± CI for key subsegments (month, contact channel, age bands).