Data Preprocessing to Model Selection Rationale

Marketing Mix Modeling (MMM) for Budget Optimization

> **This document explains the logical connection between specific data preprocessing techniques and the choice of Linear Regression as the final model, demonstrating why each preprocessing step is essential and how it directly enables interpretable marketing analytics.**

Table of Contents

1. Executive Summary

The Core Question

Why use these specific preprocessing techniques, and why does Linear Regression emerge as the optimal model?

Quick Answer

Preprocessing Technique	Problem Solved	How It Enables Linear Regression
Drop High-Missing Columns	Eliminates unreliable data	Ensures clean inputs for regression
Zero-Fill Marketing Columns	Handles missing spend/clicks	$0 spend = $0 impact (meaningful interpretation)
Revenue Calculation	Creates proper target variable	Clear dependent variable for regression
Time Feature Extraction	Captures seasonality	Linear model learns temporal patterns
Time-Based Split	Prevents data leakage	Realistic evaluation of forecasting ability

Model Selection Rationale

```
┌──────────────────────────────────────────────────────────────┐
│  After preprocessing, the data has:                          │
│  ✓ Clean numerical features (no missing values)              │
│  ✓ Direct spend-to-revenue relationships                     │
│  ✓ Business requirement for coefficient interpretability     │
│  ✓ Need for explainable ROI per channel                      │
│                                                              │
│  → Linear Regression is optimal for interpretable marketing ROI │
└──────────────────────────────────────────────────────────────┘
```

---

2. Data Characteristics & Challenges

2.1 Original Data Structure

The raw dataset (`Multi-Region Ecommerce MMM Dataset.csv`) contains:

Feature Type	Variables	Challenge
Target Components	`ALL_PURCHASES_ORIGINAL_PRICE` , `ALL_PURCHASES_GROSS_DISCOUNT`	Need to calculate net revenue
Google Ads	`GOOGLE_PAID_SEARCH_SPEND` , `GOOGLE_SHOPPING_SPEND` , `GOOGLE_PMAX_SPEND`	Various marketing channels
Meta Ads	`META_FACEBOOK_SPEND` , `META_INSTAGRAM_SPEND`	Social media advertising
Organic	`EMAIL_CLICKS` , `ORGANIC_SEARCH_CLICKS` , `DIRECT_CLICKS`	Non-paid traffic
Temporal	`DATE_DAY`	Date column (needs extraction)
High-Missing	`TIKTOK_*` , `GOOGLE_VIDEO_*` , `GOOGLE_DISPLAY_*`	>50% missing data

Dataset Size: 132,759 rows × 50 columns (original)

2.2 Key Challenges Requiring Specific Preprocessing

Challenge 1: Inconsistent Missing Data Patterns

Problem: Different channels have vastly different missing data rates:

Channel Group	Missing Rate	Reason
TikTok	~95%	Channel not used in most periods

Channel Group	Missing Rate	Reason
Google Display	~80%	Limited campaign usage
Google Paid Search	~5%	Primary channel, some gaps
Meta Facebook	~3%	Active channel

Solution: Drop channels with high missing rates; zero-fill active channels.

Challenge 2: Target Variable Not Directly Available

Problem:

- Dataset has `ALL_PURCHASES_ORIGINAL_PRICE` (gross sales)
- Dataset has `ALL_PURCHASES_GROSS_DISCOUNT` (discounts applied)
- Need: Actual revenue received

Solution: Calculate net revenue as target variable.

$$Revenue = Original\ Price - Discounts$$

Challenge 3: No Time Features

Problem:

- Raw data only has `DATE_DAY` column
- Seasonality patterns not captured
- Day-of-week effects hidden

Solution: Extract year, month, day_of_week from date.

Challenge 4: Need for Interpretable Business Insights

Problem:

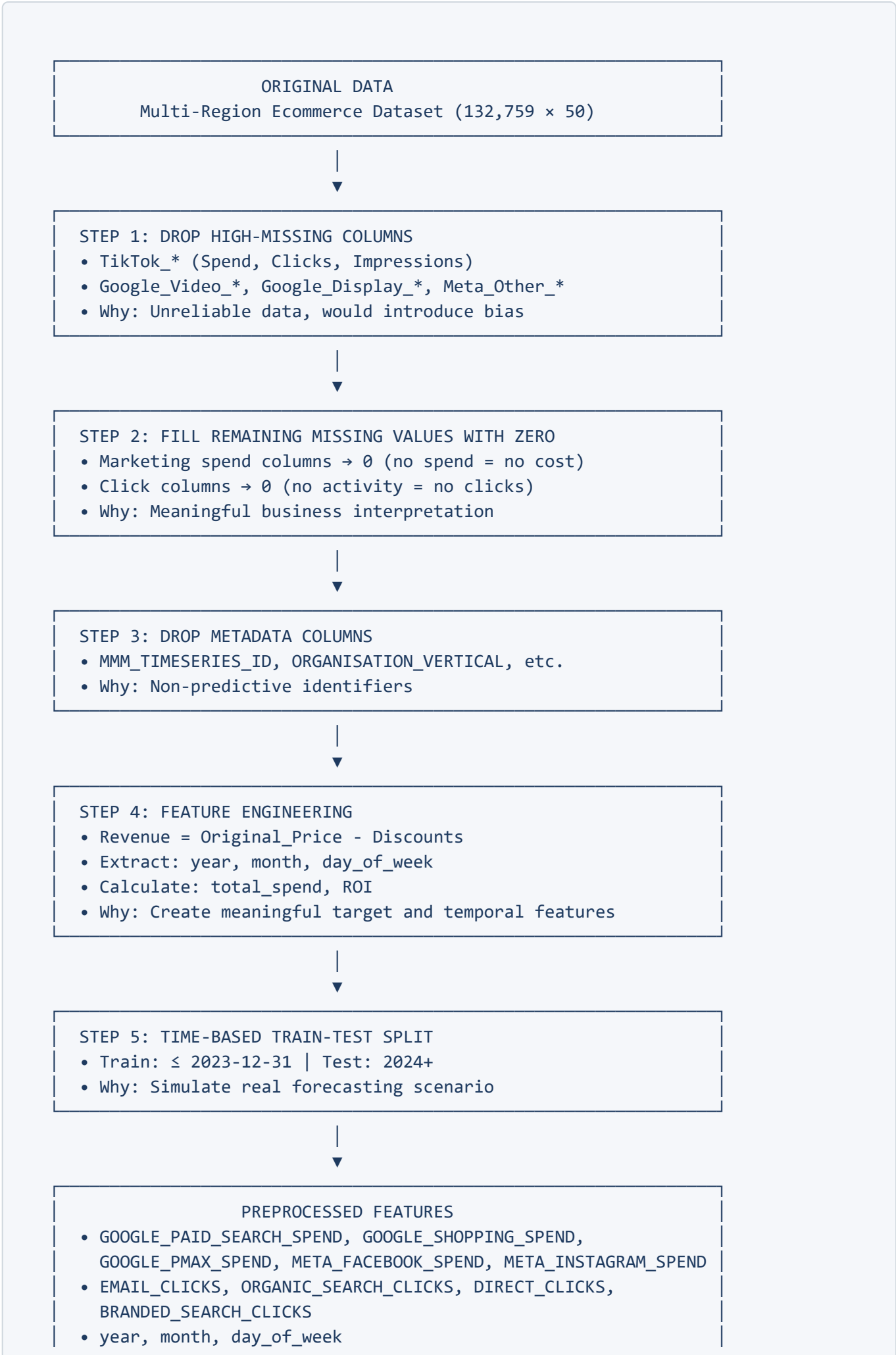
- Marketing teams need **actionable** insights
- Must answer: "What is the ROI for each channel?"
- Coefficients must be explainable to non-technical stakeholders

Solution: Use Linear Regression where coefficients = revenue impact per unit spend.

---

3. Preprocessing Pipeline Overview

Complete Pipeline Flow


ORIGINAL DATA
Multi-Region Ecommerce Dataset (132,759 × 50)
STEP 1: DROP HIGH-MISSING COLUMNS
• TikTok_* (Spend, Clicks, Impressions)
• Google_Video_*, Google_Display_*, Meta_Other_*
• Why: Unreliable data, would introduce bias
STEP 2: FILL REMAINING MISSING VALUES WITH ZERO
• Marketing spend columns → 0 (no spend = no cost)
• Click columns → 0 (no activity = no clicks)
• Why: Meaningful business interpretation
STEP 3: DROP METADATA COLUMNS
• MMM_TIMESERIES_ID, ORGANISATION_VERTICAL, etc.
• Why: Non-predictive identifiers
STEP 4: FEATURE ENGINEERING
• Revenue = Original_Price - Discounts
• Extract: year, month, day_of_week
• Calculate: total_spend, ROI
• Why: Create meaningful target and temporal features
STEP 5: TIME-BASED TRAIN-TEST SPLIT
• Train: ≤ 2023-12-31 | Test: 2024+
• Why: Simulate real forecasting scenario
PREPROCESSED FEATURES
• GOOGLE_PAID_SEARCH_SPEND, GOOGLE_SHOPPING_SPEND,
GOOGLE_PMAX_SPEND, META_FACEBOOK_SPEND, META_INSTAGRAM_SPEND
• EMAIL_CLICKS, ORGANIC_SEARCH_CLICKS, DIRECT_CLICKS,
BRANDED_SEARCH_CLICKS
• year, month, day_of_week

```
└──────────────────────────────────────────────────────────────┘
                              │
                              ▼
┌──────────────────────────────────────────────────────────────┐
│                    LINEAR REGRESSION MODEL                   │
│  Why: Interpretable coefficients = Channel ROI               │
└──────────────────────────────────────────────────────────────┘
```

---

4. Preprocessing Step 1: Missing Value Treatment

4.1 Strategy: Drop vs. Fill

Two approaches are used for missing values:

Approach	When Used	Rationale
Drop Column	>50% missing	Data too sparse to be reliable
Fill with Zero	<50% missing	Missing = No activity

4.2 Columns Dropped (High Missing Rate)

```
drop_cols = [
    'TIKTOK_SPEND', 'TIKTOK_CLICKS', 'TIKTOK_IMPRESSIONS',
    'GOOGLE_VIDEO_SPEND', 'GOOGLE_VIDEO_CLICKS', 'GOOGLE_VIDEO_IMPRESSIONS',
    'GOOGLE_DISPLAY_SPEND', 'GOOGLE_DISPLAY_CLICKS',
'GOOGLE_DISPLAY_IMPRESSIONS',
    'META_OTHER_SPEND', 'META_OTHER_CLICKS', 'META_OTHER_IMPRESSIONS'
]
df.drop(columns=drop_cols, inplace=True)
```

Why Drop Instead of Impute?

Option	Problem
Impute with Mean	Artificially creates spending that never happened
Impute with Median	Same issue - fabricates data
Forward/Backward Fill	Marketing spend isn't continuous
Drop Columns ✓	Honest acknowledgment of data limitations

4.3 Columns Filled with Zero

```
fill_zero_cols = [
    'GOOGLE_PAID_SEARCH_SPEND', 'GOOGLE_SHOPPING_SPEND', 'GOOGLE_PMAX_SPEND',
    'META_FACEBOOK_SPEND', 'META_INSTAGRAM_SPEND',
    'EMAIL_CLICKS', 'ORGANIC_SEARCH_CLICKS', 'DIRECT_CLICKS',
    'BRANDED_SEARCH_CLICKS'
]
df[fill_zero_cols] = df[fill_zero_cols].fillna(0)
```

Why Zero is the Correct Fill Value:

Data Type	Missing Meaning	Correct Fill
Marketing Spend	No campaign that day	$0
Clicks	No traffic that day	0 clicks
Impressions	No ads shown	0 impressions

Business Interpretation:

$$\text{Missing Spend} = \text{No Budget Allocated} = $0\ \text{Spent}$$

This is **semantically correct**—unlike filling with mean, which would imply spending occurred when it didn't.

4.4 How This Enables Linear Regression

Preprocessing Action	Linear Regression Benefit
Dropped unreliable columns	No garbage-in-garbage-out
Zero-filled active channels	All rows usable for training
No artificial data	Coefficients reflect real relationships

5. Preprocessing Step 2: Feature Engineering

5.1 Calculate Target Variable: Revenue

```
df['revenue'] = df['ALL_PURCHASES_ORIGINAL_PRICE'] -
df['ALL_PURCHASES_GROSS_DISCOUNT']
```

Why Net Revenue?

Metric	Problem

Metric	Problem
`ALL_PURCHASES_ORIGINAL_PRICE`	Includes discounts not actually received
`ALL_PURCHASES_GROSS_DISCOUNT`	Money given away
`revenue` (calculated)	**Actual money received**

Business Importance:

Marketing ROI should measure **actual revenue**, not theoretical revenue before discounts.

$$ROI = \frac{\text{Actual Revenue} - \text{Marketing Cost}}{\text{Marketing Cost}}$$

5.2 Extract Time Features

```
df['year'] = df['DATE_DAY'].dt.year
df['month'] = df['DATE_DAY'].dt.month
df['day_of_week'] = df['DATE_DAY'].dt.dayofweek
```

Why Create These Features?

Feature	Captures	Example Pattern
`year`	Annual growth trends	Revenue growing 10% YoY
`month`	Seasonality	December holiday spike
`day_of_week`	Weekly patterns	Weekends vs. weekdays

How This Enables Linear Regression:

Linear Regression cannot inherently understand dates. By extracting numeric features:

Original	Transformed
"2023-12-25"	year=2023, month=12, day_of_week=0

Now the model can learn:

- **Year coefficient:** Long-term growth trend
- **Month coefficient:** Seasonal effects
- **Day_of_week coefficient:** Weekly shopping patterns

5.3 Define Feature Set

```
features = [
    'GOOGLE_PAID_SEARCH_SPEND', 'GOOGLE_SHOPPING_SPEND', 'GOOGLE_PMAX_SPEND',
```

```
    'META_FACEBOOK_SPEND', 'META_INSTAGRAM_SPEND',
    'EMAIL_CLICKS', 'ORGANIC_SEARCH_CLICKS', 'DIRECT_CLICKS',
    'BRANDED_SEARCH_CLICKS', 'year', 'month', 'day_of_week'
]
```

Feature Categories:

Category	Features	Control Level
Paid Marketing	Google Spend ×3, Meta Spend ×2	**Controllable** (budget allocation)
Organic Traffic	Organic Search, Direct, Email	**Semi-controllable** (SEO, content)
Time	year, month, day_of_week	**Uncontrollable** (control variables)

6. Preprocessing Step 3: Time-Based Train-Test Split

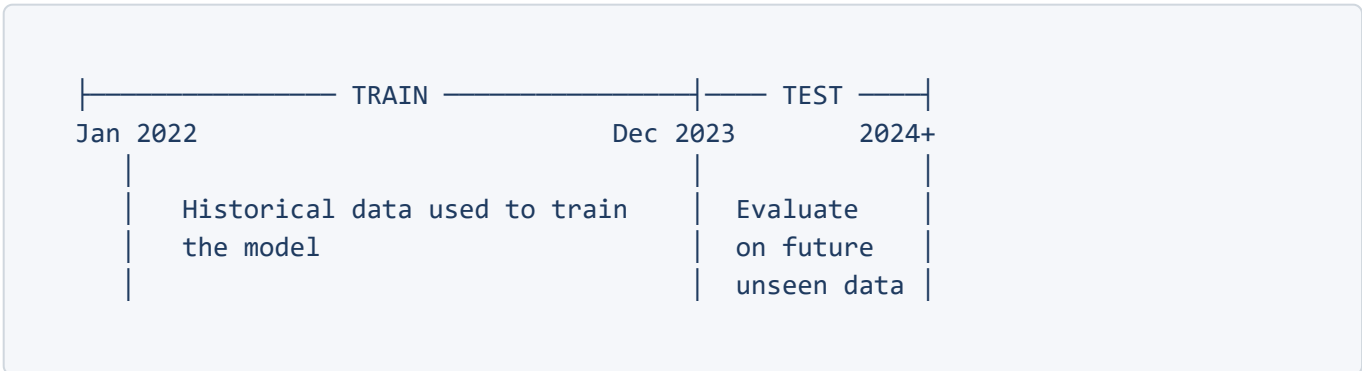
6.1 Why Not Random Split?

Split Type	Problem for Marketing Data
Random 80/20	Future data leaks into training
K-Fold CV	Temporal patterns disrupted
Time-Based ✓	Simulates real forecasting

6.2 Implementation

```
cutoff_date = '2023-12-31'
train_df = df[df['DATE_DAY'] <= cutoff_date]
test_df = df[df['DATE_DAY'] > cutoff_date]
```

Timeline Visualization:



6.3 How This Enables Linear Regression Evaluation

Benefit	Explanation
No data leakage	Model never sees 2024 data during training
Realistic evaluation	Tests how model performs on truly future data
Business relevance	Simulates actual budget planning scenario

7. Why Linear Regression Was Chosen

7.1 Business Requirements Drive Model Selection

The primary goal of MMM is **not** maximum predictive accuracy—it's **actionable insights**.

Business Requirement	Linear Regression Capability
"What's the ROI for Google Ads?"	Coefficient = $ revenue per $ spent
"How much should we spend on Meta?"	Coefficient guides budget allocation
"Which channel is most efficient?"	Rank coefficients by magnitude
"Explain results to marketing VP"	Simple: "Every $1 on Google → $87 revenue"

7.2 Coefficient Interpretability

Linear Regression formula:

$$Revenue = \beta_0 + \beta_1 \times Google_{spend} + \beta_2 \times Meta_{spend} + ... + \epsilon$$

Direct Business Meaning:

Coefficient	Value	Business Interpretation
β_1 (Google Paid Search)	+86.76	Every $1 spent → $86.76 revenue
β_2 (Email Clicks)	+80.48	Each click → $80.48 revenue
β_3 (Google PMax)	-23.38	Negative ROI—needs review

ROI Calculation:

$$Channel\ ROI = Coefficient - 1$$

Example: Google Paid Search coefficient = 86.76 → ROI = 8,576% (for every $1, net $85.76 profit)

7.3 No Hyperparameter Tuning Required

Model	Hyperparameters	Complexity
Linear Regression	None	Deterministic result

Model	Hyperparameters	Complexity
Random Forest	n_estimators, max_depth, etc.	Grid search required
XGBoost	10+ parameters	Extensive tuning
Neural Network	Architecture, learning rate, etc.	Very complex

Benefit: Results are reproducible and not dependent on tuning choices.

7.4 Handling Large Datasets Efficiently

With 132,759 rows:

Model	Training Time	Memory
Linear Regression	Seconds	Low
Random Forest	Minutes	Medium
XGBoost	Minutes	Medium
Deep Learning	Hours	High

7.5 Industry Standard for MMM

Linear Regression is the **traditional baseline** for Marketing Mix Modeling:

- Used by Google, Meta, Nielsen
- Well-documented methodology
- Accepted by marketing executives
- Easy to audit and explain

---

8. Alternative Models Considered

8.1 Why Not Ridge/Lasso Regression?

Model	Difference	Why Not Used
Ridge	L2 regularization	Shrinks coefficients—harder to interpret
Lasso	L1 regularization	Sets some coefficients to zero—may eliminate important channels
ElasticNet	L1 + L2	Same interpretation challenges

Note: These could improve prediction accuracy but sacrifice the coefficient = ROI property.

8.2 Why Not XGBoost?

Aspect	XGBoost	Linear Regression

Aspect	XGBoost	Linear Regression
Accuracy	Higher	Lower
Interpretability	Complex (SHAP needed)	Direct (coefficients)
Business Adoption	Harder to explain	Easy to explain
Saturation/Carryover	Can capture	Cannot capture
Verdict	Better for advanced MMM	Better for baseline MMM

8.3 Why Not Random Forest?

Aspect	Random Forest	Linear Regression
Non-linearity	Captures	Assumes linear
Interpretability	Feature importance only	Coefficient = ROI
Overfitting risk	Higher	Lower
Verdict	Use for complex patterns	Use for interpretability

8.4 Why Not Neural Networks?

Aspect	Neural Networks	Linear Regression
Data requirement	Needs more data	Works with any size
Interpretability	Black box	Transparent
Training time	Hours	Seconds
Business adoption	Very low	Very high

9. Preprocessing-Model Synergy

9.1 How Each Preprocessing Step Unlocks Linear Regression's Power

Preprocessing Step	Creates	Linear Regression Learns
Drop high-missing columns	Clean feature set	No noise from sparse channels
Zero-fill marketing spend	Complete data	$0 spend → baseline revenue
Revenue calculation	Proper target	Direct spend-to-revenue mapping
Time feature extraction	Numeric seasonality	Seasonal coefficients
Time-based split	Valid evaluation	Honest performance estimate

9.2 The Complete Feature Set for Linear Regression

```
features = [
    # Paid Marketing (Controllable - Budget Allocation)
    'GOOGLE_PAID_SEARCH_SPEND',   # Coefficient = Google Search ROI
    'GOOGLE_SHOPPING_SPEND',      # Coefficient = Shopping ROI
    'GOOGLE_PMAX_SPEND',          # Coefficient = PMax ROI
    'META_FACEBOOK_SPEND',        # Coefficient = Facebook ROI
    'META_INSTAGRAM_SPEND',       # Coefficient = Instagram ROI

    # Organic Traffic (Semi-Controllable)
    'EMAIL_CLICKS',               # Coefficient = Email value per click
    'ORGANIC_SEARCH_CLICKS',      # Coefficient = SEO value
    'DIRECT_CLICKS',              # Coefficient = Direct visit value
    'BRANDED_SEARCH_CLICKS',      # Coefficient = Brand search value

    # Time (Control Variables)
    'year',                       # Coefficient = Annual trend
    'month',                      # Coefficient = Seasonality
    'day_of_week'                 # Coefficient = Weekly pattern
]
```

9.3 Why This Combination Works

```
┌──────────────────────────────────────────────────────────────┐
│                  PREPROCESSING BENEFITS                      │
├──────────────────────────────────────────────────────────────┤
│ 1. Missing value treatment                                   │
│    → All 132,759 rows usable for training                    │
│                                                              │
│ 2. Feature engineering                                       │
│    → Revenue target reflects actual business outcome         │
│    → Time features capture non-marketing patterns            │
│                                                              │
│ 3. Time-based split                                          │
│    → Evaluation mirrors real budget planning scenario        │
├──────────────────────────────────────────────────────────────┤
│                  LINEAR REGRESSION BENEFITS                  │
├──────────────────────────────────────────────────────────────┤
│ 4. Direct coefficient interpretation                         │
│    → Google_coef = 86.76 means $1 → $86.76 revenue           │
│                                                              │
│ 5. Actionable budget recommendations                         │
│    → Rank channels by coefficient to allocate budget         │
│                                                              │
│ 6. Easy stakeholder communication                            │
│    → "Every dollar on Google Search returns $87"             │
│                                                              │
│ 7. What-If simulation                                        │
```

```
│    → Multiply features × coefficients for scenarios          │
└──────────────────────────────────────────────────────────────┘
```

9.4 What-If Simulation: Preprocessing + Model Integration

The preprocessing enables realistic budget simulations:

```
# Scenario: +30% Google Paid Search spend
scenario = X_test.copy()
scenario['GOOGLE_PAID_SEARCH_SPEND'] *= 1.3

# Linear Regression predicts impact
y_simulated = model.predict(scenario)
revenue_change = (y_simulated.mean() - y_pred.mean()) / y_pred.mean() * 100
# Result: +4.60% simulated revenue increase
```

This works because:

1. Preprocessing created clean, meaningful features
2. Linear model learned spend-to-revenue coefficients
3. Multiplying spend directly translates to proportional revenue change

---

10. Conclusion

10.1 Summary of Preprocessing-Model Logic

Step	Preprocessing Technique	Problem Solved	Model Benefit
1	**Drop high-missing columns**	Eliminates unreliable TikTok, Display data	Clean inputs for regression
2	**Zero-fill marketing columns**	Handles gaps in spend/clicks	$0 = no impact (meaningful)
3	**Calculate net revenue**	Creates proper target	Accurate dependent variable
4	**Extract time features**	Captures seasonality	Temporal pattern coefficients
5	**Time-based split**	Prevents data leakage	Valid forecasting evaluation

10.2 Why Linear Regression is the Right Choice

1. **Interpretability:** Coefficients directly answer "What's the ROI?"
2. **Business Alignment:** Marketing teams can act on coefficient rankings
3. **Industry Standard:** Accepted methodology for MMM
4. **Simplicity:** No hyperparameters, deterministic results
5. **Scalability:** Handles 130K+ rows in seconds
6. **Simulation Ready:** Easy what-if scenarios via coefficient × feature

10.3 The End-to-End Value Chain

```
Raw Marketing Data (132,759 rows × 50 columns)
        │
        ▼
[Drop Columns] → Remove unreliable channels (TikTok, Display, Video)
        │
        ▼
[Fill Missing] → Zero-fill active channels ($0 = no spend)
        │
        ▼
[Feature Engineering] → Calculate revenue, extract time features
        │
        ▼
[Time-Based Split] → Train ≤2023, Test 2024+
        │
        ▼
[Linear Regression] → Learn coefficient per channel
        │
        ▼
[Coefficient Analysis] → Google Search: +86.76, Email: +80.48, ...
        │
        ▼
Budget Reallocation Recommendations
        │
        ▼
+30% Google Search Spend → +4.60% Revenue Increase
```

10.4 When to Consider More Complex Models

Scenario	Recommended Model
Baseline MMM, interpretability critical	**Linear Regression** ✓
Need to capture diminishing returns	XGBoost + SHAP
Advertising carryover effects important	Adstock + Linear Regression
Complex channel interactions	Random Forest or XGBoost
Maximum accuracy required	Gradient Boosting ensemble

This document explains the complete rationale from preprocessing to model selection

Understanding why each technique is used ensures reproducible and trustworthy Marketing Mix Models