# Step 1: Amazon PPC 101 for AI Engineer

## ◆ Campaign Structure Basics

How we group campaigns:

- **Types:** Sponsored Products, Brands, Display
- **Campaign levels:** Portfolio → Campaign → Ad Group → Keyword → ASIN
- **Bid types:** down-only
- **Match types:** Exact, Phrase, Broad
- **Ad types:** Manual vs Automatic campaigns

---

## ◆ Key PPC Metrics

| Metric | Meaning | Typical AI use |
|---|---|---|
| Impressions | Times ad is shown | Detects reach & seasonality |
| Clicks | User interest | Engagement metric |
| CTR | Clicks / Impressions | Determines ad appeal |
| CPC | Cost per click | Input for bid optimization |
| Spend | Total ad spend | Used in ACOS and budget logic |
| Orders | Conversions | Core success metric |
| Revenue | Sales value | Used for ROAS |
| ACOS | Spend / Revenue | Used to optimize bids |
| ROAS | Revenue / Spend | Profitability metric |
| CVR | Orders / Clicks | Conversion health |
| TACOS | Total ad spend / Total revenue | Measures overall account health |

---

## ◆ Business Context

**East West Furniture's goals**:

- Large catalog (~12K listings)
- Focus: Dining Tables, Chairs, Benches, Dining Sets, Patio Table, Chairs, Sets, Bedroom Sets
- Goals: **Profitability + Ranking + Visibility**
- Budget: $50K–$100K/month
- High competition; ranking stability is key

---

## Step 2: Convert PPC Knowledge → AI Logic Framework

### Example: Bid Optimization Rules

```
If keyword has ≥ 1 orders and ACOS < target:
    Increase bid by 5%
If keyword has ≥ 2 orders and ACOS < target:
    Increase bid by 10%
If keyword has ≥ 3 orders and ACOS < target:
    Increase bid by 20%

Else if ACOS > 1.5× target and no sales for 14 days:
    Decrease bid by 10%
Else if main keyword (ranking keyword) and no sales:
    Hold bid steady for at least 30 days (ranking protection)
```

### Example: Negative Keyword Logic

```
If Clicks > 100 and Orders == 0:
    Add to negative keyword list
Unless keyword in "core_keyword_list"
```

### Example: Seasonal Adjustment

```
If month in [Nov, Dec]:
    Apply +25% bid multiplier to top 10% converting keywords
If off-season:
    Apply -15% bid reduction to low traffic, high CPC keywords
```

---

## ⚙️ Step 3: Data Map

Where the AI will get its data:

- **Amazon Ads API** — Campaign data, search term reports, keyword performance
- **Business Reports** — Organic sales, conversion rate, TACOS
- **Keyword ranking tools** (Helium10, DataDive, etc.) — Organic rank
- **Internal database** — product category, profit margin, seasonality tags

Example schema (simplified):

| Keyword | Clicks | Orders | Spend | Revenue | ACOS | Rank | Season | Type |
|---|---|---|---|---|---|---|---|---|
| drop leaf table | 1908 | 48 | 1028.21 | 14,267.01 | 7.21% | 18 | Q4 | main |
| small drop leaf table | 22 | 0 | 10.50 | 0 | — | 56 | Q2 | long-tail |

## Step 4: Define AI Modules:

### CORE AI MODULES

| Module | Purpose | Key Output |
|---|---|---|
| 1. Data Intelligence Engine | Collect & preprocess data | Clean, structured performance dataset |
| 2. Keyword Intelligence Engine | Keyword discovery, segmentation, negatives | Keyword classification & actions |
| 3. Bid Optimization Engine | Smart bid adjustments | Dynamic bid updates |
| 4. Budget Manager | Budget pacing & reallocation | Daily budget shifts |
| 5. Ranking & Seasonality Engine | Seasonal bid scaling, rank protection | Rank-aware bid updates |
| 6. Long-Tail Discovery Engine | Manage low-volume keywords | Discovery campaign management |
| 7. Profitability & Forecast Engine | Predict ROI, ACOS, seasonality | Forecasts & optimization triggers |
| 8. Listing Optimization Advisor | Detect listing issues (CVR drop) | Content optimization alerts |
| 9. Feedback & Learning Loop | Continuous model training | Smarter decision-making over time |

### ◆ 1. DATA INTELLIGENCE ENGINE

**(The foundation of the system)**

### Functions

- Pulls data from:
  - Amazon Ads API (Search Term Reports, Campaign data)
  - Business Reports (sessions, conversion, inventory)
  - Listing metadata (titles, bullet points, attributes)
  - Calendar/Seasonal datasets (Prime Day, Q4 events)
- Cleans, normalizes, and aggregates data daily

### Outputs

- Unified dataset by ASIN, keyword, and date

- Derived metrics: CTR, CVR, ACOS, ROAS, Impressions, Spend, Orders
- Feeds all other modules

---

## ◆ 2. KEYWORD INTELLIGENCE ENGINE

**(The brain of keyword lifecycle management)**

### Key Functions

1. **Keyword Extraction:**
   - Analyzes search term data
   - Promotes converting terms → manual exact
2. **Negative Keyword Reasoning:**
   - Adds negatives for irrelevant or wasteful search terms
   - NLP model detects semantic mismatch
3. **Keyword Segmentation:**
   - Classifies keywords into:
     - High Performer
     - Moderate
     - Strategic (Core)
     - Dormant (Long-tail)
     - Dead
     - High-Spend Waster
     - Seasonal
4. **Relevance Scoring (NLP):**
   - Matches keyword meaning to ASIN content using embeddings

### Sample Rule

```
if Orders == 0 and Relevance > 0.8 and Clicks < 5:
  segment = "Dormant Long-Tail"
elif Orders >= 3 and ACOS <= Target:
  segment = "High Performer"
```

### Output

- Keyword classification table with automated actions (keep, pause, reallocate, or move)

---

## ◆ 3. BID OPTIMIZATION ENGINE

**(The tactical adjustment layer)**

### Inputs

- Keyword segment
- Target ACOS
- Actual ACOS
- Rank signal
- Seasonal multiplier

### Logic

New Bid = Base Bid × Segment Weight × (TargetACOS / ActualACOS) × RankMultiplier × SeasonMultiplier

| Segment | Weight | Example Action |
|---|---|---|
| High Performer | 1.2 | Bid +20% |
| Moderate | 1.0 | Maintain |
| Strategic | 0.8 | -20% |
| Dormant | 0.5 | -50% |
| Dead | 0 | Pause |

### Features

- Learns optimal bids from historical CVR and CPC data
- Includes daily and weekly recalibration
- Adapts based on placement (Top of Search vs Product Page)

---

## ◆ 4. BUDGET MANAGER

**(The allocator & controller)**

### Functions

- Analyzes budget depletion time and performance
- Redistributes daily budget:
    - From low-ROAS campaigns → to high-performers

- Detects out-of-budget campaigns and boosts automatically
- Forecasts required daily pacing to hit monthly spend target

### Logic Example

```
If Campaign_ACOS < Target and Budget_Used > 90% by noon:
    Increase_Budget( +20% )
If Campaign_ACOS > 2×Target:
    Decrease_Budget( -30% )
```

### Outputs

- Budget reallocation report
- Pacing recommendations

---

## ◆ 5. RANKING & SEASONALITY ENGINE

**(The strategic layer)**

### Functions

- Tracks organic rank via keyword monitoring API
- Calculates **Rank Weight**:
- Rank Weight = 1 + ( (30 - Rank) / 100 ) if Rank > 30 else 1
- Uses event calendar + historical data to predict seasonal demand
- Prepares **Seasonal Multipliers**:
  - Prime Day: +30–50%
  - Black Friday/Cyber Monday: +60–80%
  - Off-season: -20–40%

### Logic Example

```
If Season = High_Demand:
    Bid = BaseBid × (1 + CVR_Lift%)
Else If Season = Low_Demand:
    Bid = BaseBid × (1 - CVR_Drop%)
```

### Outputs

- Rank-aware bid changes
- Seasonal ramp-up and cool-down scheduling

## 1. Shopping Season – Aggressive Bid Adjustment

**Goal:** Maximize visibility and sales velocity when demand and conversion rates peak.

### 🔍 Inputs AI Should Analyze

- Calendar events (Prime Day, Black Friday, Cyber Monday, Christmas, etc.)
- Historical seasonality patterns (YoY conversion lift, CPC spikes)
- Real-time conversion rate uplift (CVR ↑ vs baseline)
- Inventory level and availability
- Target TACOS or ACOS flexibility during season

### ⚙□ AI Logic

- Detect seasonal event or traffic spike → increase bids dynamically
- Bid uplift factor based on historical CVR increase:
- New Bid = Current Bid × (1 + Seasonal_CVR_Lift%)
- Example:
  If CVR typically increases by 40% during Black Friday → bid +40%
- Budget scaling rules:
  - Raise campaign daily budget up to +100% if out of budget before noon.
  - Monitor profitability → revert post-season.

### Extra Smart Layer:

- Use predictive modeling to identify **"ramp-up windows"** (3–5 days before event) and **"cool-down windows"** (2–3 days after event).

---

## 🌱 2. Off-Season – Bid & Spend Contraction

**Goal:** Preserve profitability and prevent waste when demand is low.

### 🔍 Inputs

- Historical off-season conversion decline
- Drop in CVR, CTR, or impression share
- Excess inventory or clearance situation

### ⚙□ AI Logic

- Detect off-season (based on category trends or conversion drop >30%)

- Gradually reduce bids or budgets:
- New Bid = Current Bid × (1 - Offseason_CVR_Drop%)
- Example:
  If CVR falls 30%, reduce bids by 30–40%.
- Prioritize only best-converting keywords and pause marginal ones.

### Optional:

- Launch **“defensive low-bid” campaigns** to keep minimal visibility but low CPC (rank protection).

---

## ◆ 6. LONG-TAIL DISCOVERY ENGINE

**(Manages rare but valuable long-tail keywords)**

### Functions

- Identifies low-volume, relevant long-tail terms
- Groups them into “Discovery Campaigns”
- Assigns ultra-low exploratory bids ($0.20–$0.50)
- Caps daily spend at 1–2% of total budget
- Monitors for new conversions to promote them

### Logic

```
If Clicks < 10 AND Relevance > 0.8:
   Bid = min(0.25 × Avg_CPC, $0.3)
If No Impressions > 60 days:
   Pause
If New Sale:
   Move to Manual Exact
```

### Output

- Controlled “lottery keyword” pool for discovery without waste

---

## ◆ 7. PROFITABILITY & FORECAST ENGINE

**(Financial intelligence core)**

### Functions

- Calculates **True Profit**:
- True Profit = Selling Price - (COGS + Fees + Ad Spend)
- True ACOS = Ad Spend / (Selling Price - COGS - Fees)
- Predicts:
    - Expected ACOS next 7–30 days
    - Seasonal sales uplift
    - CPC trend forecasting using regression models
- Suggests bid caps & profit-based limits

### Output

- Profit forecast per ASIN
- Smart bid ceilings to avoid overspending

---

## ◆ 8. LISTING OPTIMIZATION ADVISOR

**(AI feedback on conversion issues)**

### Functions

- Detects when CVR drops for important keywords
- Compares ad click CVR vs. listing CVR
- Flags listings needing optimization:
    - Images not matching search intent
    - Poor ratings (<3.5)
    - Missing color/variant keywords

### Example Trigger

If Keyword_CVR < 50% of Avg_CVR → Flag listing for optimization

### Output

- Listing health report for each ASIN

---

## ◆ 9. FEEDBACK & LEARNING LOOP

**(The self-improving engine)**

### Functions

- Retrains models weekly:
  - CTR prediction
  - CVR forecasting
  - Profit correlation
- Runs A/B bid experiments to learn best response curves
- Stores performance outcomes for continuous reinforcement learning

### Output

- Improved accuracy in bid predictions
- Evolving bid and keyword segmentation thresholds

---

## DECISION EXAMPLES

| Scenario | AI Action | Explanation |
|---|---|---|
| “black dining chair” – high importance, low sales | Move to Rank Campaign, reduce bid 25%, monitor rank | Strategic keyword protection |
| “modern oak dining chair set of 6” – 0 sales in 6 months | Move to Discovery campaign, $0.25 bid | Low-cost long-tail |
| “rustic dining table” – 10 orders, ACOS 5% | Increase bid 15% | High performer scaling |
| “wooden bench 72 inch” – 100 clicks, no sales | Add as negative | Wasteful term |

---

## 📊 SYSTEM DATA FLOW (Simplified)

[Amazon Ads Data]

↓

[Data Intelligence Engine]

↓

[Keyword Intelligence Engine] → [Long-Tail Engine] → [Negative Manager]

↓

[Bid Optimization Engine] ← [Ranking & Seasonality Engine] ← [Profit Engine]

↓

[Budget Manager]
↓
[Campaign Updates via Amazon Ads API]
↓
[Feedback & Learning Loop]

## 💼 EXECUTION FREQUENCY SUMMARY

| Process | Frequency |
|---|---|
| Keyword Segmentation | Daily |
| Bid Optimization | Daily |
| Budget Reallocation | Daily |
| Long-Tail Review | Weekly |
| Seasonality Forecast Update | Monthly |
| Learning Model Retrain | Weekly |

## Step 5: How to Test and Validate

Build **dashboards** showing:

- Before/After metrics (Spend, ACOS, Sales)
- List of paused, scaled, and added keywords
- Ranking improvement tracking (organic position trend)

This helps us monitor if AI decisions are good or harmful.

## 💬 Step 6: Continuous Feedback Loop

Schedule weekly syncs:

- Review results and logic

- Tune parameters or ML weighting
- Gradually let AI handle more autonomy

---

## 🚀 Final Output Example

“The system should automatically adjust bids, budgets, and keywords daily based on logic and real-time data, while protecting ranking keywords and optimizing profit.”