Customer Review Analysis: Student Practice Guide

From Reviews to Business Decisions

Today's Mission

You're a product manager at a laptop company. Customer reviews are pouring in. Your job: Figure out what to fix, what to promote, and who needs help.

Setup Code

```python
import pandas as pd

# Load the reviews
df = pd.read_csv("reviews_demo.csv")

# Check what we have
print(f"Total reviews: {len(df)}")
print(f"Columns: {list(df.columns)}")
print(f"Average rating: {df['rating'].mean():.2f}")
```

Part 1: Basic Sentiment Analysis

Step 1: Create Sentiment Labels

```python
def label_sentiment(rating):
    if rating >= 4:
        return "positive"
    elif rating <= 2:
        return "negative"
    else:
        return "neutral"

df["sentiment"] = df["rating"].apply(label_sentiment)
```

Check Your Work

```python
print(df["sentiment"].value_counts())
```

Questions to Answer:

- What % of reviews are positive? _____
- What % are negative? _____
- Is this good or concerning? _____

Part 2: The Hidden Truth

Investigate: Rating vs. Recommendation

```python
```

```python
# Create a crosstab
ct = pd.crosstab(
    df["sentiment"],
    df["recommend"],
    normalize="index"
).round(2)

print(ct)
```

Discovery Questions:

1. What % of positive reviews DON'T recommend? _____
2. Why might someone give 4 stars but not recommend? _____
3. What business action does this suggest? _____

Part 3: Aspect-Based Analysis

Find What Customers Talk About

```python
# Helper function
def contains_keywords(text, keywords):
    text = str(text).lower()
    return any(k in text for k in keywords)

# TASK: Complete these keyword lists
battery_keywords = ["battery", "charge", "___", "___"]
screen_keywords = ["screen", "display", "___", "___"]
price_keywords = ["price", "expensive", "___", "___"]

# Flag mentions
df["mentions_battery"] = df["review_text"].apply(
    lambda x: contains_keywords(x, battery_keywords)
)
```

Analyze Each Aspect

```python
def analyze_aspect(df, column, aspect_name):
    # Get reviews mentioning this aspect
    subset = df[df[column] == True]

    # Calculate sentiment breakdown
    if len(subset) > 0:
        sentiment_pcts = subset["sentiment"].value_counts(normalize=True)

        print(f"\n{aspect_name}:")
        print(f"  Total mentions: {len(subset)}")
        print(f"  Positive: {sentiment_pcts.get('positive', 0):.1%}")
        print(f"  Negative: {sentiment_pcts.get('negative', 0):.1%}")

# Run analysis
analyze_aspect(df, "mentions_battery", "Battery Life")
```

Fill in Your Findings:

Aspect	# Mentions	% Positive	% Negative	Action Needed
Battery	_____	_____	_____	_____
Screen	_____	_____	_____	_____
Price	_____	_____	_____	_____

Part 4: Business Decisions

The Decision Matrix

Based on your analysis, categorize each aspect:

HIGH PRIORITY FIX (Many mentions, mostly negative):

- Aspect: _____
- Why: _____

MARKETING OPPORTUNITY (Many mentions, mostly positive):

- Aspect: _____
- Why: _____

MONITOR (Few mentions, mixed sentiment):

- Aspect: _____
- Why: _____

Part 5: Find Specific Customers to Help

Identify At-Risk Customers

```python
# Find very unhappy battery complainers
unhappy_battery = df[
    (df["mentions_battery"] == True) &
    (df["rating"] <= 2)
]

print(f"Customers needing immediate outreach: {len(unhappy_battery)}")

# View a sample
if len(unhappy_battery) > 0:
    print("\nSample review:")
    print(unhappy_battery["review_text"].iloc[0][:200])
```

Customer Success Actions:

1. How many customers need immediate outreach? _____
2. What would you offer them? _____
3. Draft a 2-sentence apology email:

Part 6: Competitive Intelligence

Compare to Industry Benchmarks

Industry averages for laptops:

- Battery complaints: 15%
- Screen complaints: 8%
- Price complaints: 25%

Your Analysis:

- Battery: ___% (vs 15% industry)
- Screen: ___% (vs 8% industry)
- Price: ___% (vs 25% industry)

Where are you winning? _____ Where are you losing? _____

Part 7: ROI Calculation

Estimate the Business Impact

```python
# Calculate potential impact
total_reviews = len(df)
negative_reviews = len(df[df["sentiment"] == "negative"])
return_rate = negative_reviews / total_reviews

print(f"Current return rate estimate: {return_rate:.1%}")

# If we fix the top complaint
top_complaint_pct = 0.40  # 40% of complaints about battery
potential_reduction = return_rate * top_complaint_pct * 0.60  # 60% fix rate

print(f"Potential return rate reduction: {potential_reduction:.1%}")

# Money saved
monthly_sales = 1000
return_cost = 75  # dollars per return
monthly_savings = monthly_sales * potential_reduction * return_cost

print(f"Estimated monthly savings: ${monthly_savings:,.0f}")
```

Business Case:

- Current problem costs: $_____/month
- Fix would save: $_____/month
- Break-even time if fix costs $50,000: _____ months

Part 8: Create Your Executive Summary

One-Page Report Template

EXECUTIVE SUMMARY - PRODUCT REVIEWS ANALYSIS

Overall Health

- Total Reviews Analyzed: _____
- Average Rating: _____
- Recommendation Rate: _____%

Top 3 Issues (Priority Order)

1. _____ (___% negative mentions)
2. _____ (___% negative mentions)
3. _____ (___% negative mentions)

Top 3 Strengths

1. _____ (___% positive mentions)
2. _____ (___% positive mentions)
3. _____ (___% positive mentions)

Recommended Actions

1. **Engineering:** Fix _____ issue
2. **Marketing:** Promote _____ in ads

3. **Customer Success:** Contact _____ at-risk customers

Expected Impact

- Return rate reduction: ____%
- Monthly savings: $_____
- Customer satisfaction increase: ___ points

Challenge Extensions

Easy: Add More Aspects

Add analysis for:

- Keyboard quality
- Weight/portability
- Customer service

Medium: Trending Analysis

```
python

# If reviews have dates
df["review_date"] = pd.to_datetime(df["review_date"])
monthly_sentiment = df.groupby(
    df["review_date"].dt.to_period("M")
)["sentiment"].value_counts(normalize=True)
```

Hard: Competitor Comparison

Load competitor reviews and compare:

- Who has fewer battery complaints?
- Who has better sentiment?
- What's their secret?

Key Formulas

Aspect Priority Score

```
Priority = (# Mentions) × (% Negative) × (Business Impact)
```

ROI Calculation

```
ROI = (Monthly Savings × 12) / Fix Cost
```

Customer Lifetime Value Impact

```
CLV Change = (Retention Rate Change) × (Avg Customer Value)
```

Remember

You just did what Amazon does with millions of reviews daily. Every insight you found could save thousands of dollars or prevent customer churn. This skill is worth $100K+ in product management roles.

Your homework: Apply this to real reviews from Amazon, Yelp, or app stores!