COMMERCE.AI

COMMUNITY EDITION

AI-Powered Commerce

Building the products and services of the future with Commerce.AI

Andy Pandharikar | Frederik Bussler

Preface

The future of commerce is here. AI-powered commerce is enabling businesses to offer new experiences and value to consumers. AI will play a critical role in the future of product and service innovation, transforming how brands engage with customers across multiple touchpoints.

Commerce.AI introduces you to the latest advances in AI and how it's being used to power new products and services across a variety of industries. You will learn about the latest advancements in AI, including deep learning, **generative adversarial networks** (or **GANs**), **natural language processing** (or **NLP**), and computer vision.

The book begins with an overview of different applications of AI for product and service innovation, including market opportunity identification, creating product ideas, and industry trend forecasting. You will then explore AI for innovation use cases across a number of industries, from consumer electronics to luxury goods. Finally, you'll learn how to use Commerce.AI's core features to empower your product and service teams to create innovative products and services that meet the needs of your customers.

Who this audiobook is for

This book will guide you through the process of product and service innovation, no matter your pre-existing skillset. Whether you're an AI developer, a product manager, an analyst, or a consumer insights professional, this book will teach you everything you need to use the power of AI for innovation.

What this audiobook covers

Chapter 1, *Improving Market Opportunity Identification*, focuses on using AI for market opportunity identification and how it can be used to better understand customers' interests in addition to their needs and desires. This is an important step for any company looking to refine its value proposition, modify customer experience, or create new products or services.

Chapter 2, *Creating Product Ideas*, explains how to leverage AI and machine learning to find product ideas. We cover the concepts of NLP and large language models to generate product ideas. We explain how you can use AI to analyze market data to create relevant and in-demand product ideas.

Chapter 3, *Understanding How to Predict Industry-Wide Trends Using Big Data*, explains how to use advanced machine learning and data science techniques to predict future trends. It explains what big data is, how it's structured, and how you can leverage it to see into the future. This is vital to creating successful products or services because it helps you understand what consumers will want in the future.

Chapter 4, *Applying AI for Innovation – Luxury Goods Deep Dive*, explores how AI is used for innovation in the luxury goods industry. It explores how luxury goods companies are using this technology to create personalization, improve customer experience, and develop new products.

Chapter 5, *Applying AI for Innovation – Wireless Networking Deep Dive*, looks at how AI is used in the wireless networking industry. It explores how AI is applied to optimize KPIs such as star ratings, bestseller rankings, product sentiment, and conversion.

Chapter 6, *Applying AI for Innovation – Consumer Electronics Deep Dive*, analyzes how AI is used for innovation in the consumer electronics industry. It explores how this technology is applied to understand and optimize product positioning, brand research, idea generation, insights extraction, and more.

Chapter 7, *Applying AI for Innovation – Restaurants Deep Dive*, explains how data is a critical tool for restaurant innovation teams. You'll learn how to use data and AI to inform your restaurant strategy.

Chapter 8, *Applying AI for Innovation – Consumer Goods Deep Dive*, lays out how AI is being utilized in the consumer goods industry. It examines how AI can be used for consumer goods market intelligence, generating content, analyzing sentiment, and so much more.

Chapter 9, *Delivering Insights with Product AI*, is an exploration of Commerce.AI's Product AI features, which empower product innovation teams to research, develop, launch, and track winning products.

Chapter 10, *Delivering Insights with Service AI*, explores Commerce.AI's Service AI features, which enable service innovation teams to build next-generation experiences and reputation by truly understanding their customers and the rest of the market.

Chapter 11, *Delivering Insights with Market AI*, explores Commerce.AI's Market AI features, which enable teams to uncover facets, use cases, and topics to determine changing customer behavior, trending products, emerging brands, and new product opportunities.

Chapter 12, *Delivering Insights with Voice Surveys*, dives into Commerce.AI voice surveys, which use speech recognition and natural language understanding. You'll learn how to gain unparalleled insights into what consumers are thinking.

To get the most out of this audiobook

You will need a stable Internet connection and a Python-compatible IDE, on either Windows, macOS, or Linux. All code examples have been tested using Google Colaboratory. However, they will work with offline IDEs and other environments as well.

Software/hardware covered in the book	Operating system requirements
Commerce.AI	Any web browser
Python pandas and other libraries	Windows, macOS, or Linux
GPT-J	Windows, macOS, or Linux
AI21 Studio	Windows, macOS, or Linux

Note

If you are using the digital version of this book, we advise you to type the code yourself or access the code from the book's GitHub repository (a link is available in the next section). Doing so will help you avoid any potential errors related to the copying and pasting of code.

During or after reading the book, we encourage you to sign up for a free trial of Commerce.AI at https://www.commerce.ai/contact.

Download the example code files

You can download the example code files for this book from GitHub at https://github.com/PacktPublishing/AI-Powered-Commerce. If there's an update to the code, it will be updated in the GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at https://github.com/PacktPublishing/. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: https://static.packt-cdn.com/downloads/9781803248981_ColorImages.pdf.

Conventions used

There are a number of text conventions used throughout this document.

`Code in text`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Take a look at more positive reviews, with a positive `polarity` and fairly low `subjectivity`."

A block of code is set as follows:

```
s = df['Reviews']

df['Reviews'] = df['Reviews'].astype(str)

df = df[df['Reviews'] == s]

df[['polarity', 'subjectivity']] = df['Reviews'].apply(lambda   Text:
pd.Series(TextBlob(Text).sentiment))
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
s = df['Reviews']

df['Reviews'] = df['Reviews'].astype(str)

df = df[df['Reviews'] == s]

df[['polarity', 'subjectivity']] = df['Reviews'].apply(lambda   Text:
pd.Series(TextBlob(Text).sentiment))
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: "Even positive reviews complain about the range, such as one review that says simply **Good short range**."

Chapter 1

Figures

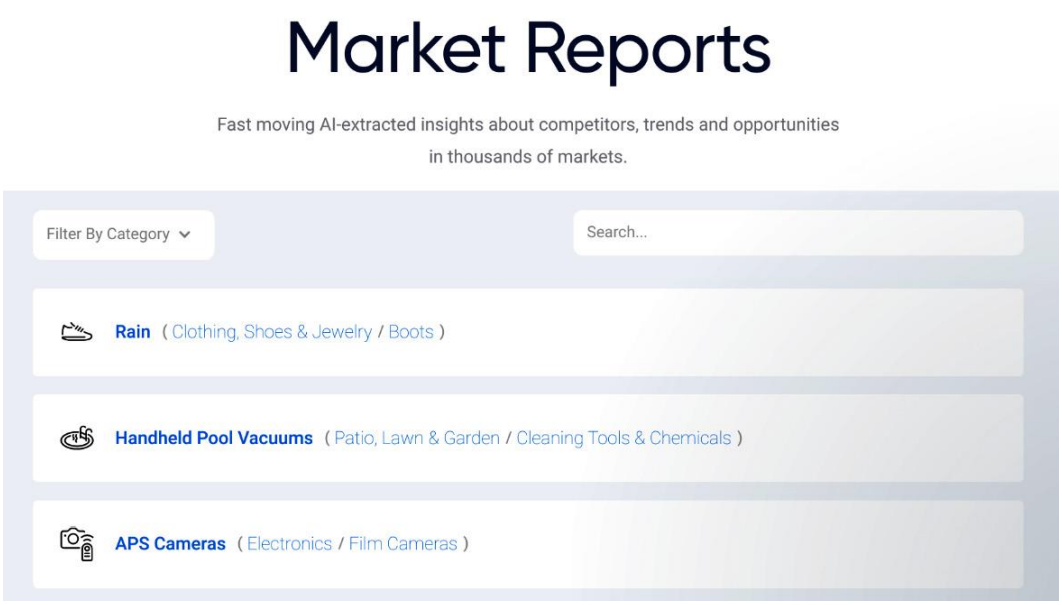

Figure 1.1 – A sample of Commerce.AI's AI-generated market reports

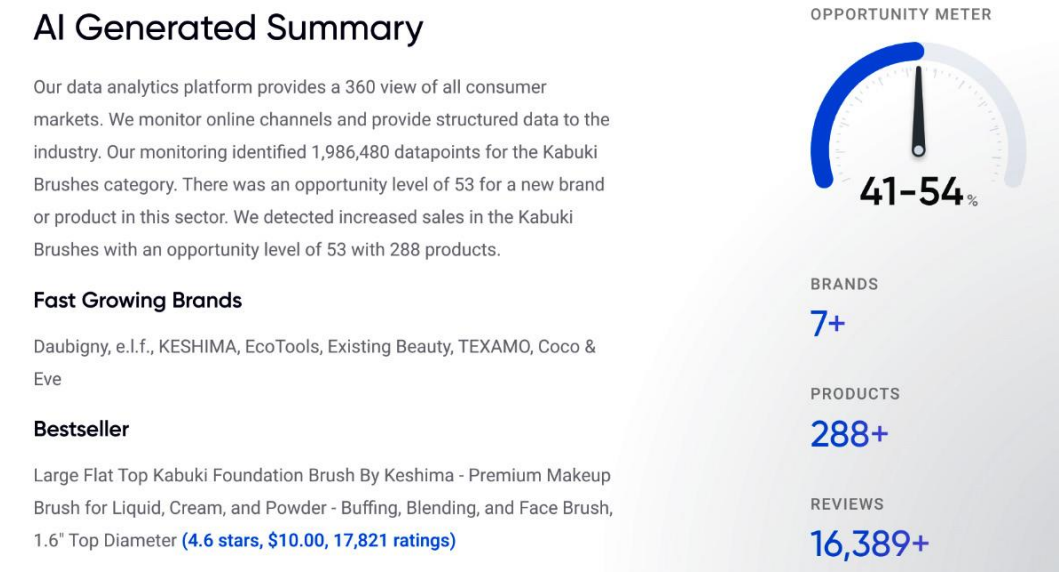

Figure 1.2 – A sample AI-generated market report

Chapter 2

Figures

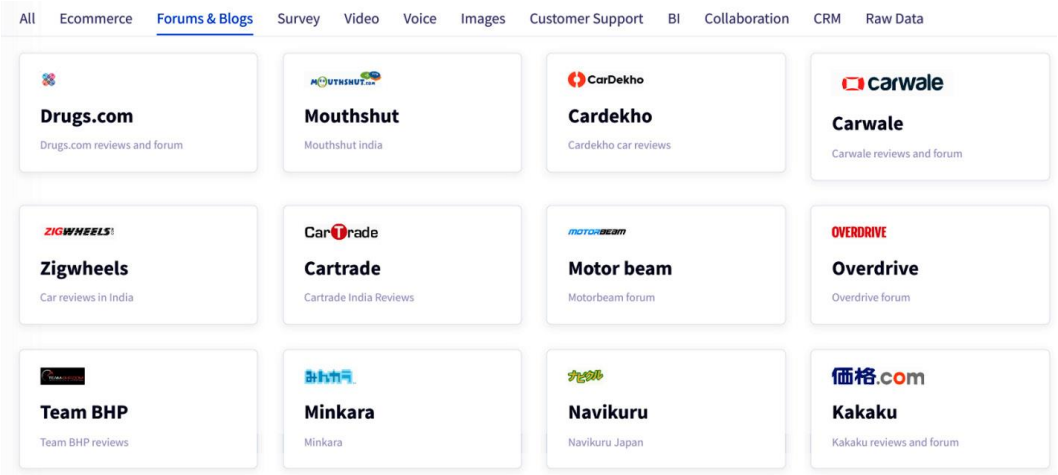

Figure 2.1 – A sample of customer data sources

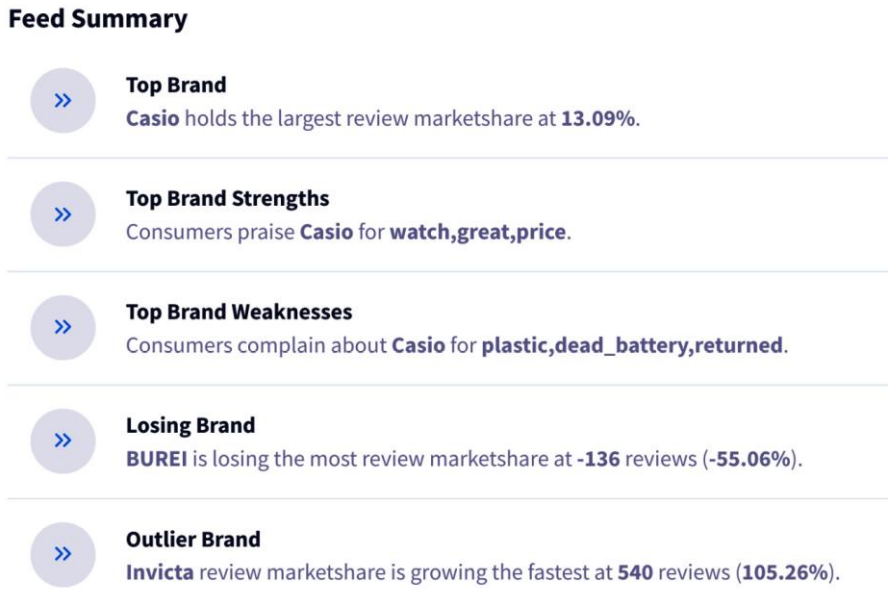

Figure 2.2 – A sample of organized information about watch brands

Figure 2.3 – Generating product ideas based on a customer wishlist

Figure 2.4 – A generative design by Emmanuel Touraine, CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)

Chapter 3

Technical requirements

You can download the latest code samples for this chapter from this book's official GitHub repository at https://github.com/PacktPublishing/AI-Powered-Commerce/tree/main/Chapter03.

Figures

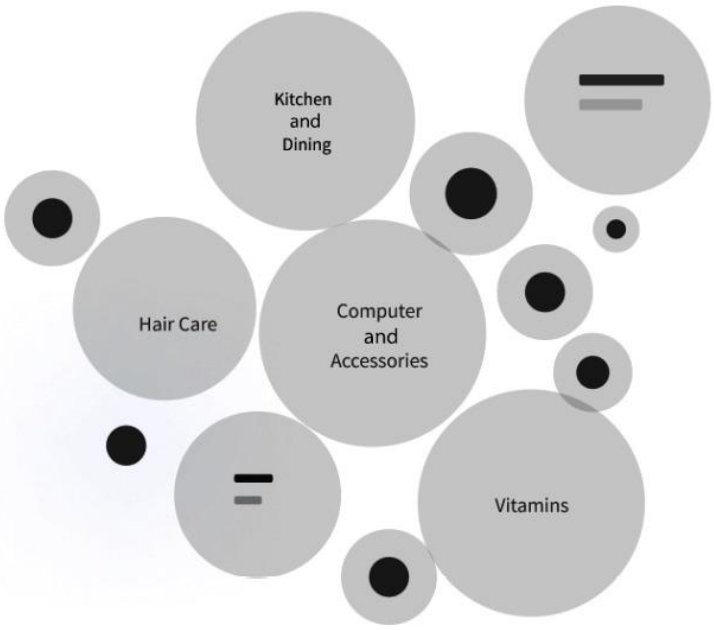

Figure 3.1 – Trending market segments

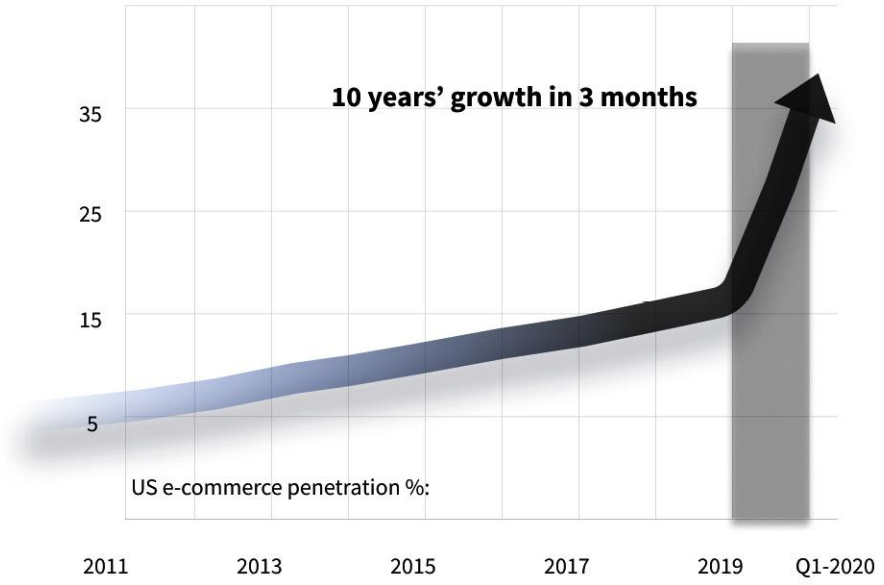

Figure 3.2 – US e-commerce penetration growth

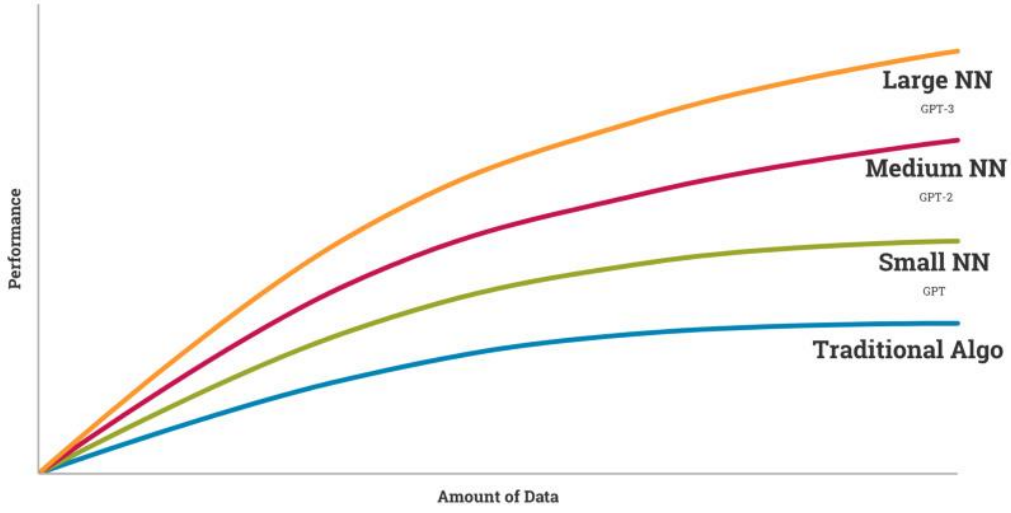

Figure 3.3 – ImageNet accuracy versus model size

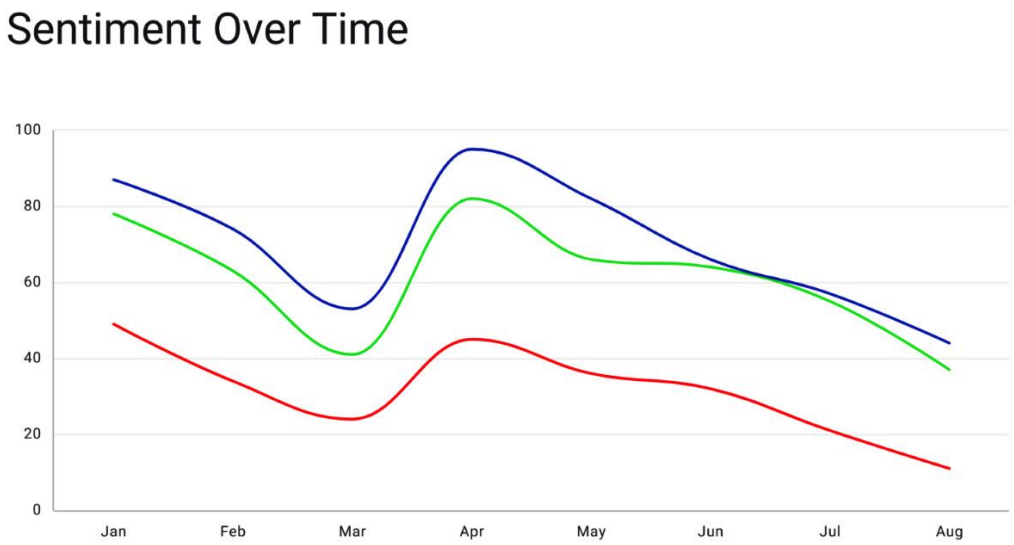

Figure 3.6 – Sentiment of Yeezy sneaker reviews over time

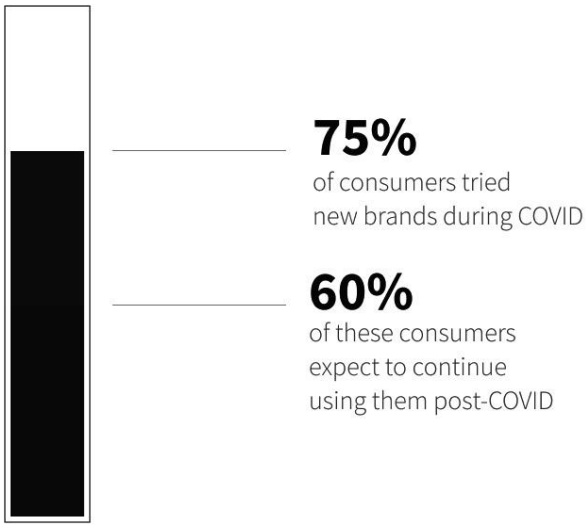

Figure 3.7 – Consumer statistics

Figure 3.8 – Map of major commerce data sources

Example: Forecasting demand for Adidas Yeezy sneakers

Let's walk through a practical example of forecasting demand for Adidas Yeezy sneakers.

1. First, we'll import the libraries that we need, which are Python's pandas for data manipulation and Facebook's Prophet, an AI forecasting library:

```
import pandas as pd
from fbprophet import Prophet
```

2. Next, we'll import our data – 5 years of worldwide search history for the term `Yeezy`, retrieved using Google Trends (https://trends.google.com/):

```
df = pd.read_csv("multiTimeline.csv")
```

Here's what the associated data looks like:

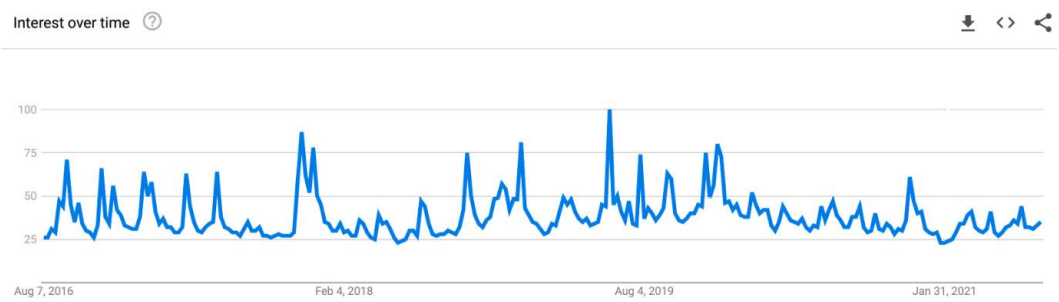

Figure 3.4 – Google Trends interest over time for the search term Yeezy

3. The preceding graph shows us search trends for the term `Yeezy`, which we now have to turn into a format that we can forecast from. Prophet requires that the datetime column is named `ds` and that the observation column is named `y`, so we'll rename both columns:

```
df = df.rename(columns = {"Week": "ds", "yeezy:
  (Worldwide)": "y"})
```

4. Building an out-of-the-box model is just two lines of code, where we first instantiate the model and then fit it to the data:

```
m = Prophet()
m.fit(df)
```

5. Now that we've built a forecasting model, it's time to make a forecast. We'll make an empty DataFrame to store forecasted values, and then fill that DataFrame with the predicted values:

```
future = m.make_future_dataframe(periods=52, freq='W')
forecast = m.predict(future)
```

6. Now, we can plot our forecast with a single line of code:

```
fig1 = m.plot(forecast)
```

With that, we've successfully predicted demand for a given product. Here's what the resulting forecast graph looks like:

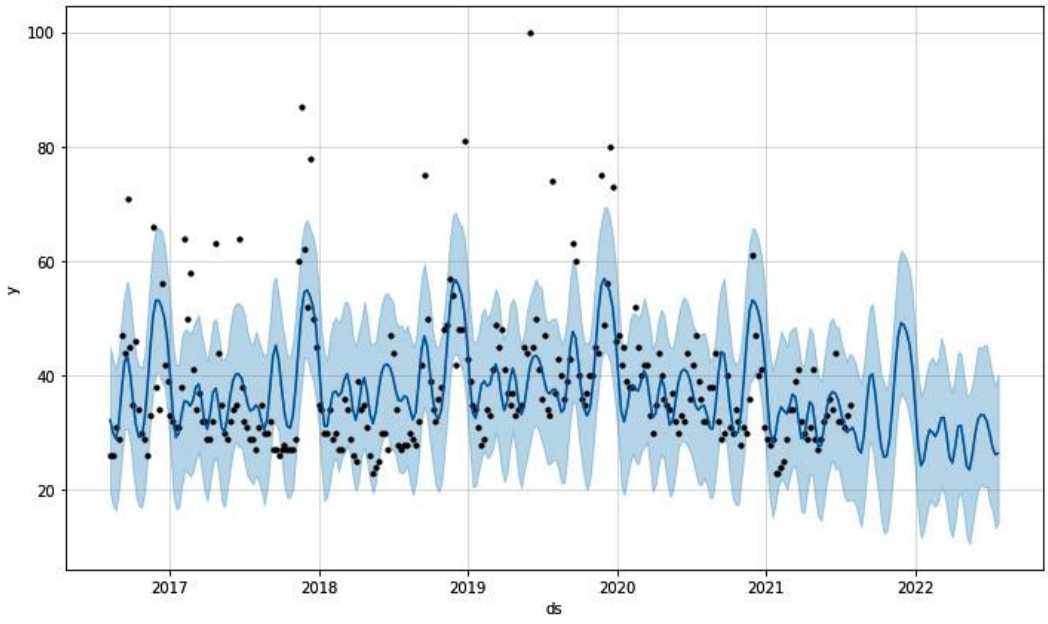

Figure 3.5 – A Facebook Prophet 1-year forecast of search demand for Yeezy

The preceding graph shows a 1-year forecast of Google Search interest for the product `Yeezy`. The same steps could be replicated for any product and any timespan or scaled to forecast demand across entire product lines or even market segments.

Code

Code 3.1 – Implementing the bert-base-multilingual-uncased-sentiment model concerning sentiment analysis

```
from transformers import pipeline

st = f"I like Yeezy"

seq = pipeline(task="text-classification",  model='nlptown/bert-base-multilingual-uncased-sentiment')

print(f"Result: { seq(st) }")
```

Code 3.2 – Sentiment classification library by TextBlob

```
from textblob import TextBlob

text = "I just bought Yeezys and am absolutely in love!"

blob = TextBlob(text)

print(blob.sentiment)
```

Chapter 4

Technical requirements

You can download the latest code samples for this chapter from this book's official GitHub repository at https://github.com/PacktPublishing/AI-Powered-Commerce/tree/main/Chapter04.

Links

The luxury market: https://www.statista.com/study/61582/in-depth-luxury/

Code

Code 4.1 – Installing GPT-J

```
!pip install gptj

from GPTJ.Basic_api import SimpleCompletion
```

Code 4.2 – Prompt

```
prompt = "Recommend a fashion item based on a list of clothes
item.\n##\nWearing: Floral skirt\nFashion item: Try a black-and-
white polkadot shirt for the classic floral and dot
combo!\n##\nWearing: Blue denim jacket\nFashion item: Try golden
yellow eye-shadow for a warm blue and gold glow!\n##\nWearing:
Red lipstick\nFashion item: Pair your red lipstick with a red
dress and red heels for a triple threat!\n##\nWearing: " + item +
"\nFashion item:"
```

Code 4.3

```
temperature = 0.4

top_probability = 1.0

max_length = 15

item = "Orange shorts"
```

Code 4.4

```
query = SimpleCompletion(prompt, length=max_length,
  t=temperature, top=top_probability)
Query = query.simple_completion()
lines = Query.splitlines()
results = []
```

Figures

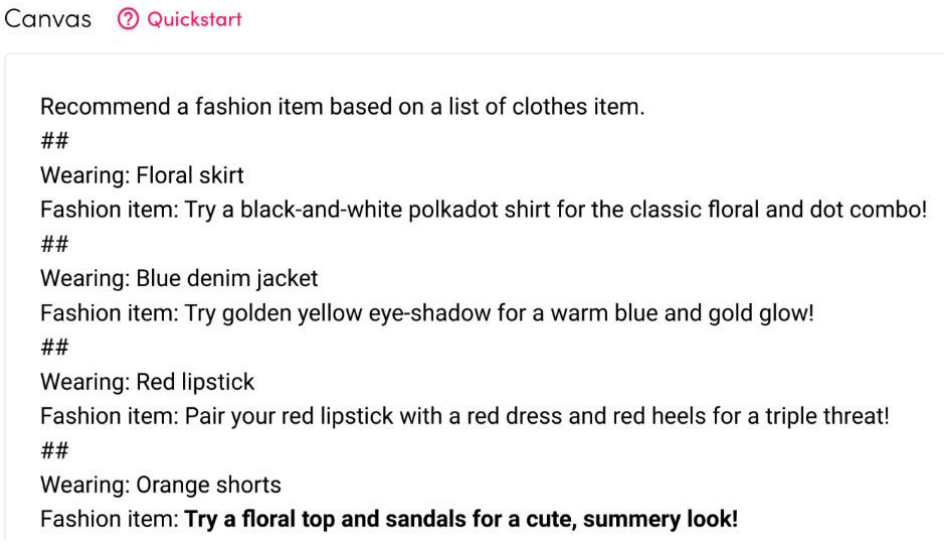

Figure 4.1 – The AI21 Studio canvas as a fashion adviser

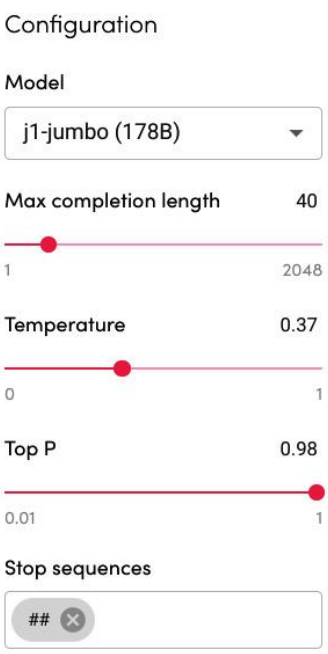

Figure 4.2 – The AI21 Studio Configuration panel

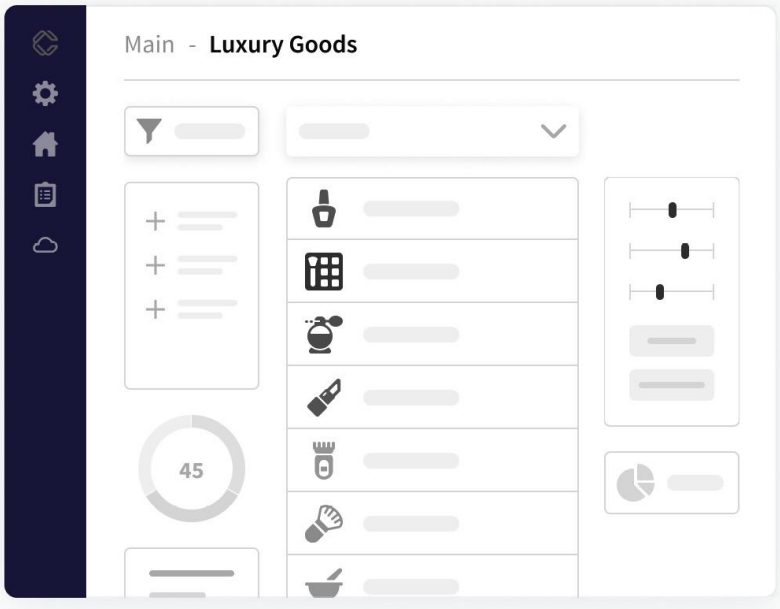

Figure 4.3 – A mock-up of a Commerce-dot-AI Luxury Goods dashboard

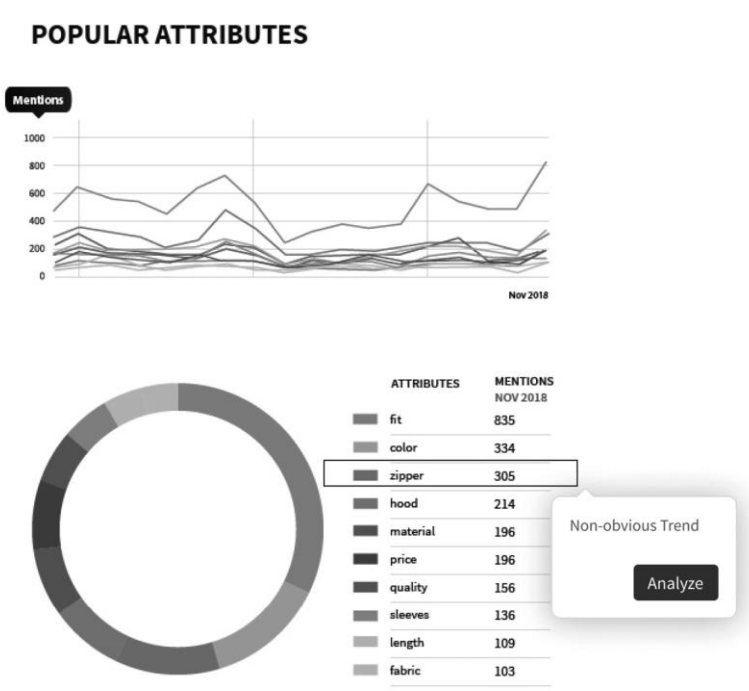

Figure 4.4 – An example of luxury brand attribute analysis

Figure 4.5 – A mock-up forecast of luxury brand trends

Chapter 5

Technical requirements

You can download the latest code samples for this chapter from this book's official GitHub repository at https://github.com/PacktPublishing/AI-Powered-Commerce/tree/main/Chapter05.

Figures

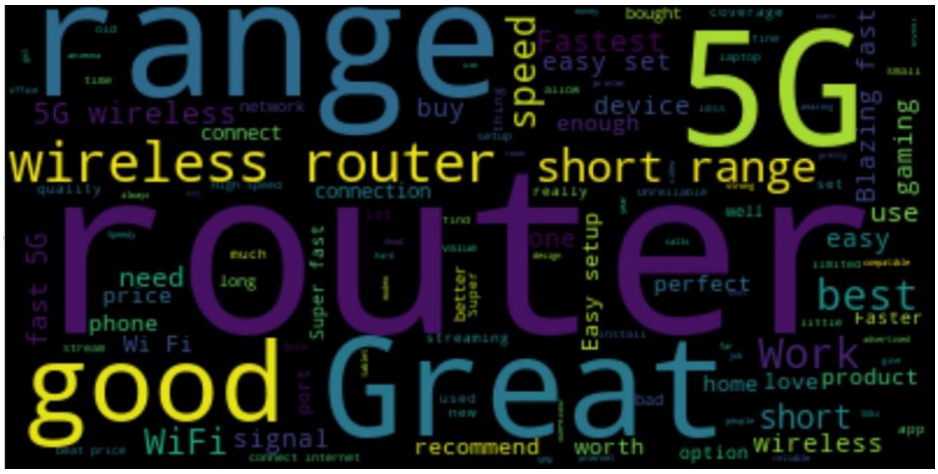

Figure 5.1 – A word cloud of 5G wireless router reviews

	Reviews	polarity	subjectivity
14	Short range and not worth the money	-0.075000	0.200000
21	This router has a limited range	-0.071429	0.142857
27	Range is not very far	-0.038462	0.769231
32	Biggest con - hard to set up	-0.291667	0.541667
57	The router's range is limited	-0.071429	0.142857

Figure 5.2 – A snippet of negative wireless router product reviews

163	Fastest router available	0.400	0.4000
179	Best router, good for gaming	0.850	0.4500
185	Good short range	0.350	0.4500
190	Best option for gaming	1.000	0.3000
193	This is the best router!	1.000	0.3000

Figure 5.3 – A snippet of positive wireless router product reviews

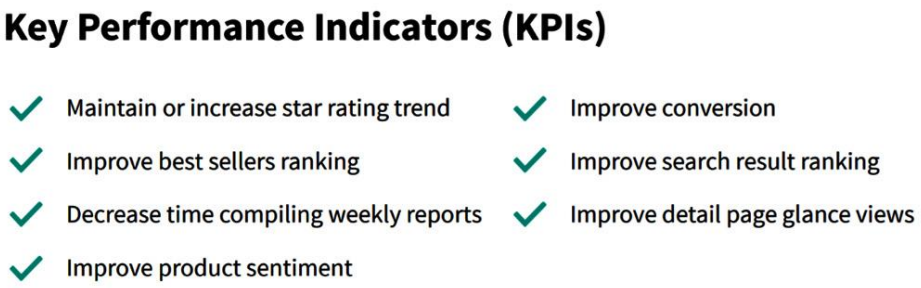

Figure 5.4 – Important KPIs for product teams

Code

Code 5.1 – Import dependencies

```
!pip install wordcloud
from os import path
from PIL import Image
from wordcloud import WordCloud, STOPWORDS
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
!pip install textblob
from textblob import TextBlob
```

Code 5.2 – Review product data

```
df = pd.read_csv('Reviews.csv')

document = df['Reviews'].to_string()
```

Code 5.3 – Generating a word cloud

```
wordcloud = WordCloud().generate(document)

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis("off")
```

Code 5.4 – We can now use `TextBlob` to analyze review sentiment and see whether we can find more meaningful insights.

```
s = df['Reviews']

df['Reviews'] = df['Reviews'].astype(str)

df = df[df['Reviews'] == s]

df[['polarity', 'subjectivity']] = df['Reviews'].apply(lambda   Text:
pd.Series(TextBlob(Text).sentiment))
```

Code 5.5 – Sorting product reviews (negative reviews)

```
df[df['polarity'] < 0]
```

Code 5.6 – sorting positive product reviews

```
df[(df['polarity'] > 0.2) & (df['subjectivity'] < 0.5)]
```

Chapter 6

Figures

Figure 6.1 – A product positioning chart mockup comparing sentiment and number of reviews

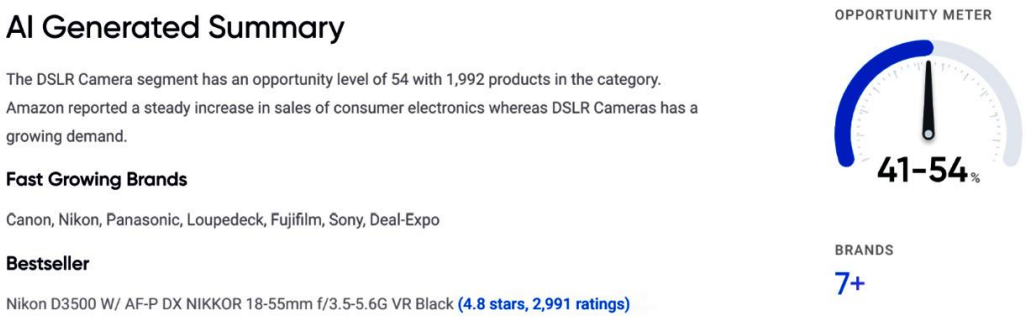

Figure 6.2 – A snippet of an AI-generated market report on DSLR cameras

Figure 6.3 – A mockup of the blank slate Commerce.AI dashboard

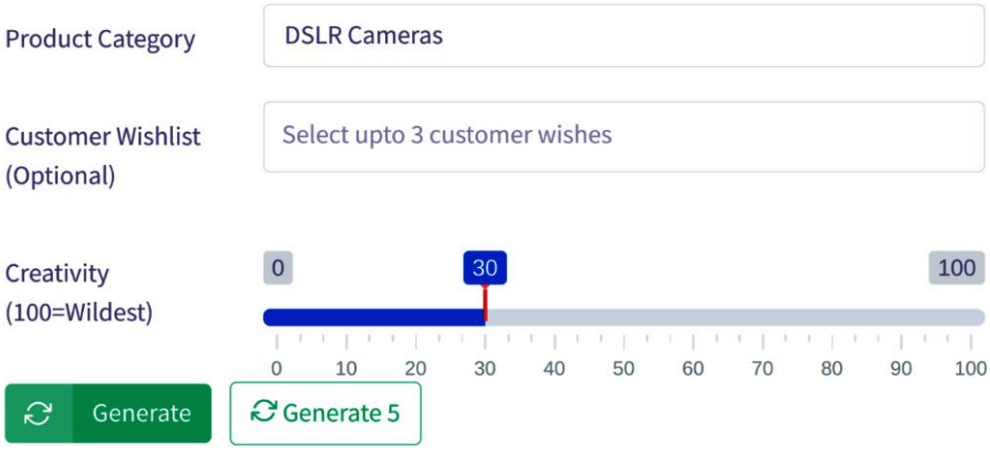

Figure 6.4 – A product positioning chart mockup comparing sentiment and number of reviews

Figure 6.5 – A snippet of AI-generated DSLR product ideas

Links

https://www.statista.com/markets/418/topic/485/consumer-electronics/#overview

Chapter 7

Figures

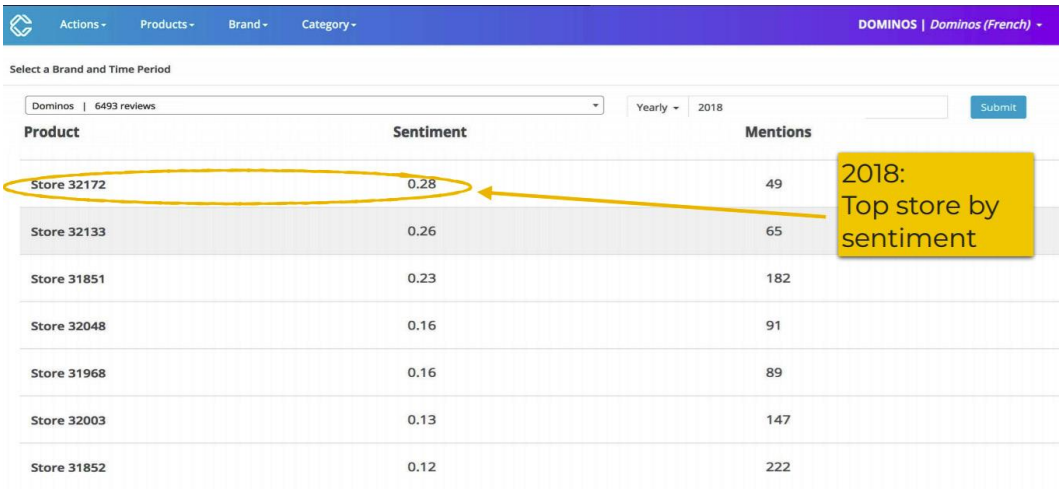

Figure 7.1 – Leaderboard of top stores by sentiment in Commerce.AI

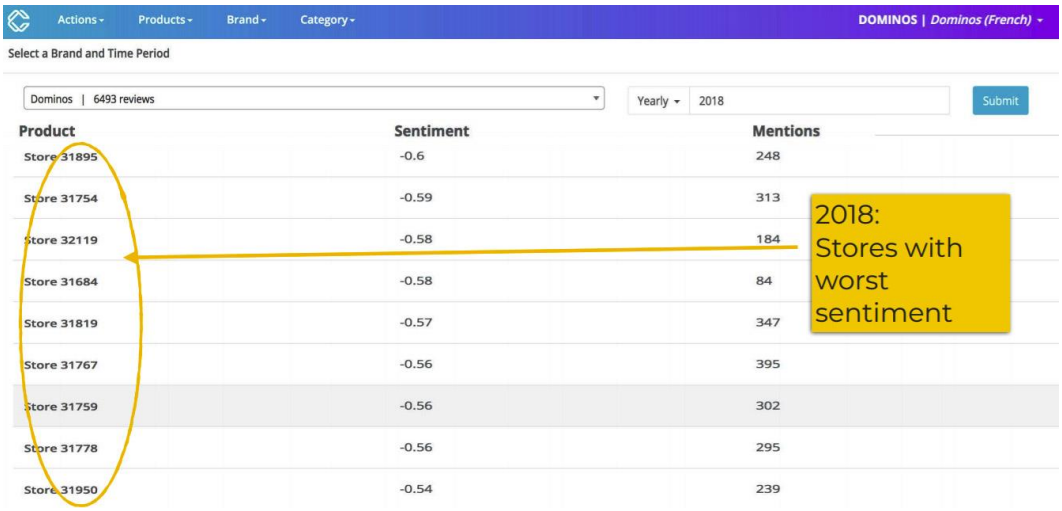

Figure 7.2 – Leaderboard of worst stores by sentiment in Commerce.AI

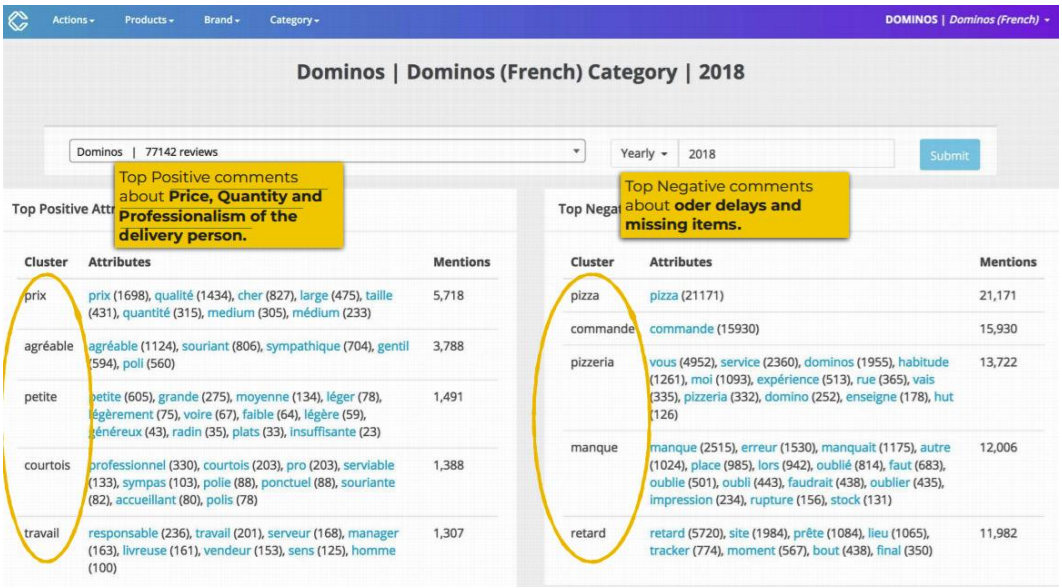

Figure 7.3 – Analyzing store attributes in Commerce.AI

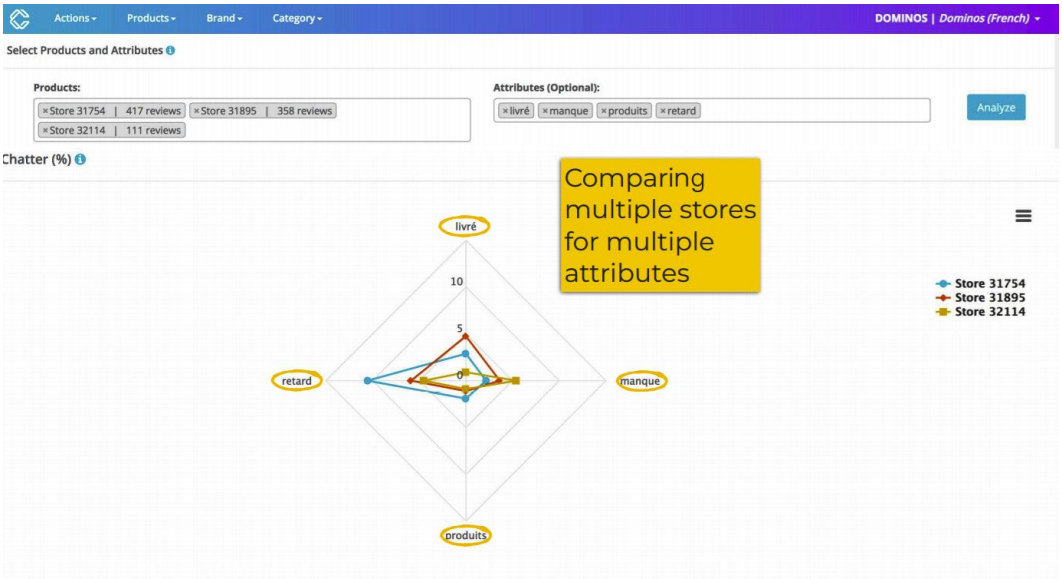

Figure 7.4 – Comparing multiple stores for multiple attributes

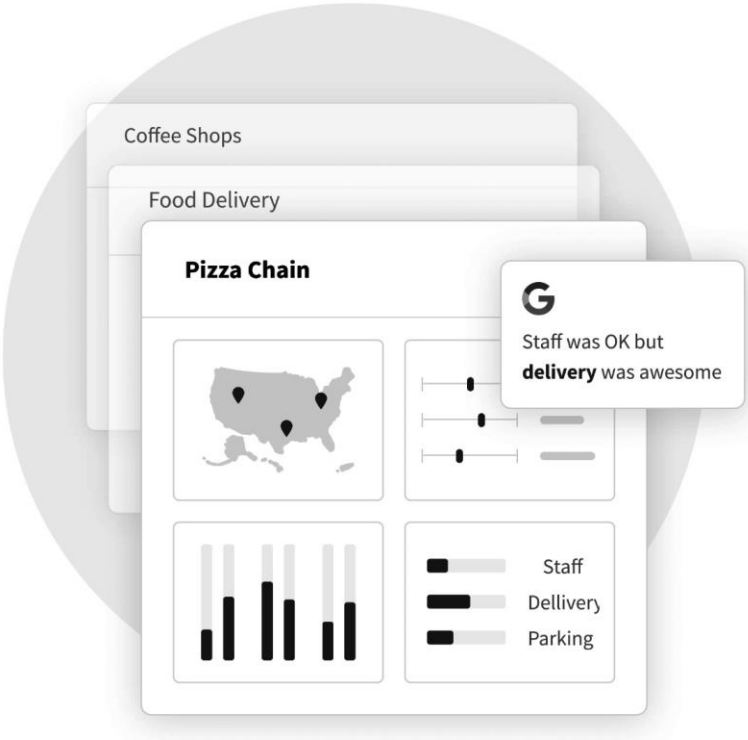

Fig 7.5 – A Commerce.AI mockup of a high-level consumer insights dashboard

Chapter 8

Technical requirements

You can download the latest code samples for this chapter from the book's GitHub repository:

https://github.com/PacktPublishing/AI-Powered-Commerce/tree/main/Chapter08

Code

Code 8.1 – Install GPT-J and import the required library:

```
!pip install gptj

from GPTJ.Basic_api import SimpleCompletion
```

Code 8.2 – Next, we define this task by providing `prompt`, which includes examples of product descriptions being generated from a product name and features:

```
prompt = "Write one sentence descriptions for products based on a list
of features.\n##\nProduct: Sundef\nFeatures:\n- Sunscreen for
athletes\n- Unique formula to prevent burning eyes\n- Can be worn on
the body and on the face\nOne sentence description: Sundef face & body
sunscreen for athletes keeps your skin protected without hurting your
eyes, so you can keep your head in the game.\n##\nProduct: " + product
+ "\nFeatures:\n" + features + "\nOne sentence description:"
```

Code 8.3 – We'll want to pass a number of parameters, primarily `temperature` (or randomness), `max_length` (or the maximum output size of the model), and `product` (or what the user types in, such as `SlimWallet`):

```
temperature = 0.4

top_probability = 1.0

max_length = 5

product = "SlimWallet"
```

Code 8.4 – Finally, we can now pass the `prompt` variable and the parameters to the model to create a recommendation. We'll also grab just the first line of text generated, in case the model goes overboard:

```
query = SimpleCompletion(prompt, length=max_length,  t=temperature,
top=top_probability)

Query = query.simple_completion()

lines = Query.splitlines()

results = []
```

Figures

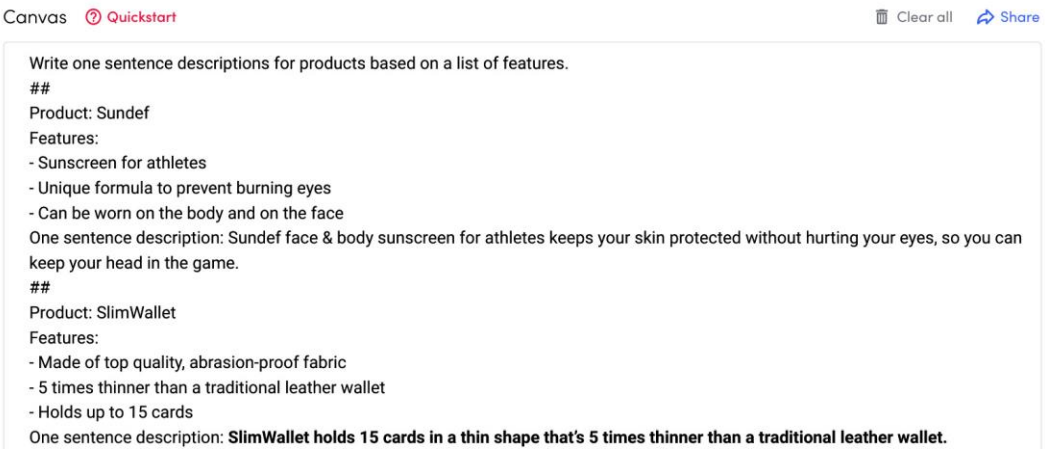

Figure 8.1 – The AI21 Studio canvas for product description generation

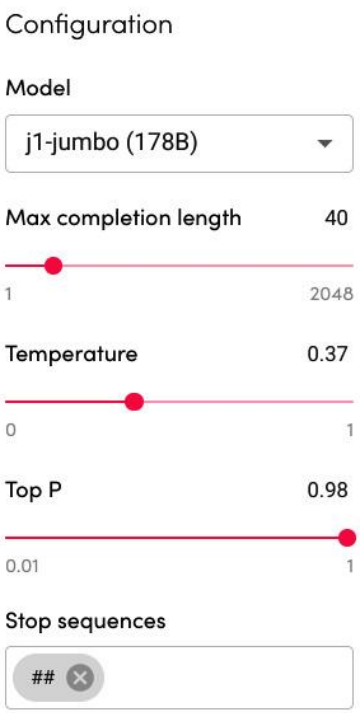

Figure 8.2 – The AI21 Studio Configuration panel

In *Figure 8.3*, we provide AI21 Studio with a prompt that extracts **Areas for improvement** from a product review:

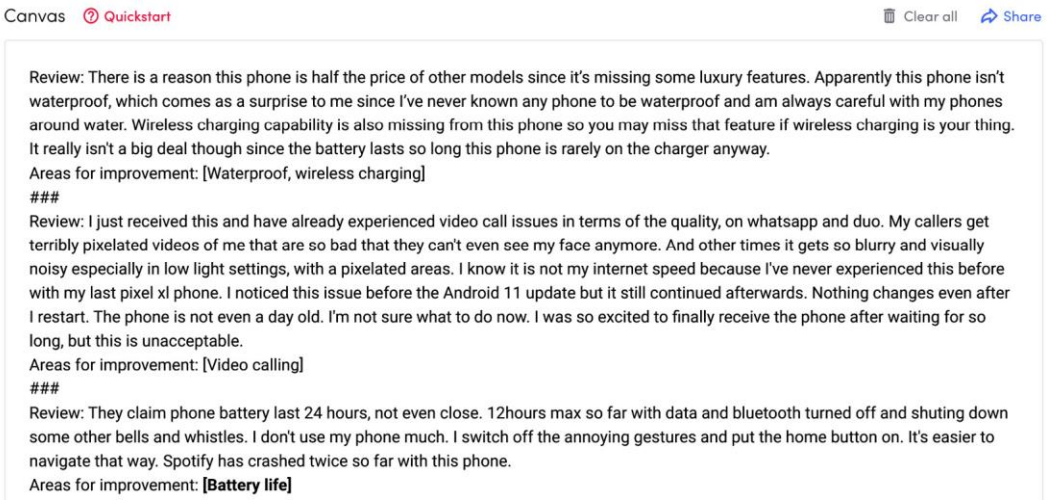

Figure 8.3 – The AI21 Studio canvas for product review analysis

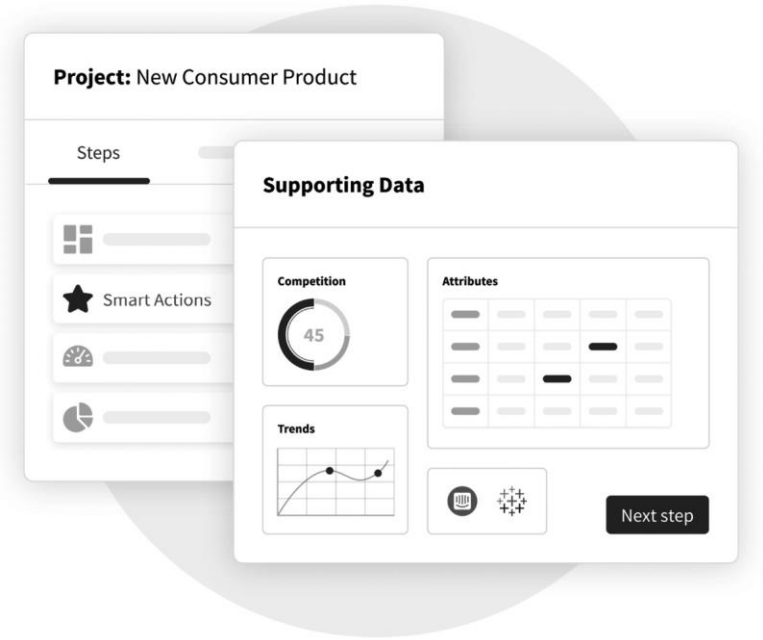

Figure 8.4 – A Commerce.AI mockup for measuring product attributes and trends

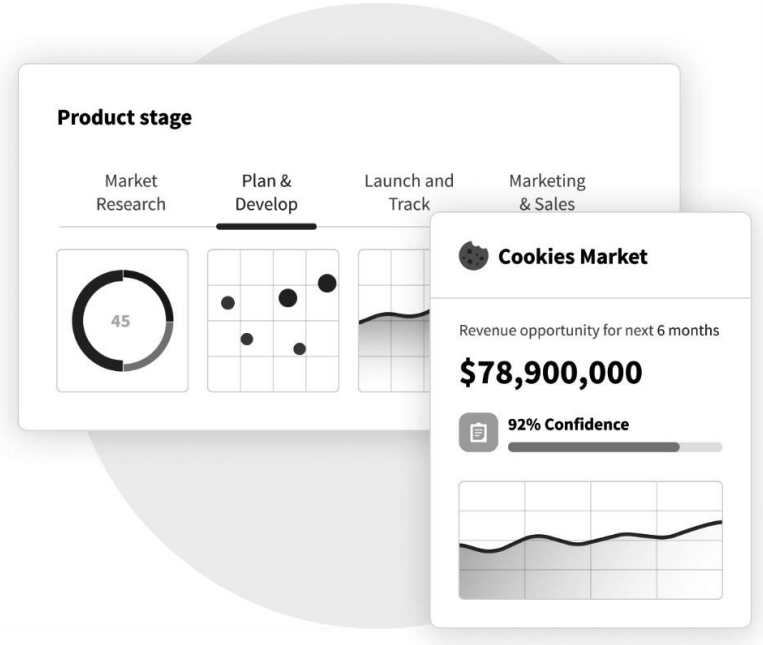

Figure 8.5 – A Commerce.AI mockup for measuring revenue opportunity

Figure 8.6 – A Commerce.AI mockup for analyzing personas and segments

Figure 8.7 – A Commerce.AI mockup for analyzing customer journeys

GENERATED IDEAS

	INDEX	SCORE	SAVE	PRODUCT IDEA
	10	★★★★★	Copy	A high-protein, healthy alternative to traditional chocolate peanut butter bars, with the same creamy and crunchy consistency.
	9	★★★★★	Copy	A healthy oils potato chip without vegetable oils.
	8	★★★★★	Copy	A no-sugar, healthy-fat based yogurt.

Figure 8.8 – AI-generated product ideas around healthy snacks

Links

https://nijianmo.github.io/amazon/index.html

Chapter 9

Links

https://www.commerce.ai/reports

Figures

Figure 9.1 – A mockup of the Commerce.AI Market signals dashboard

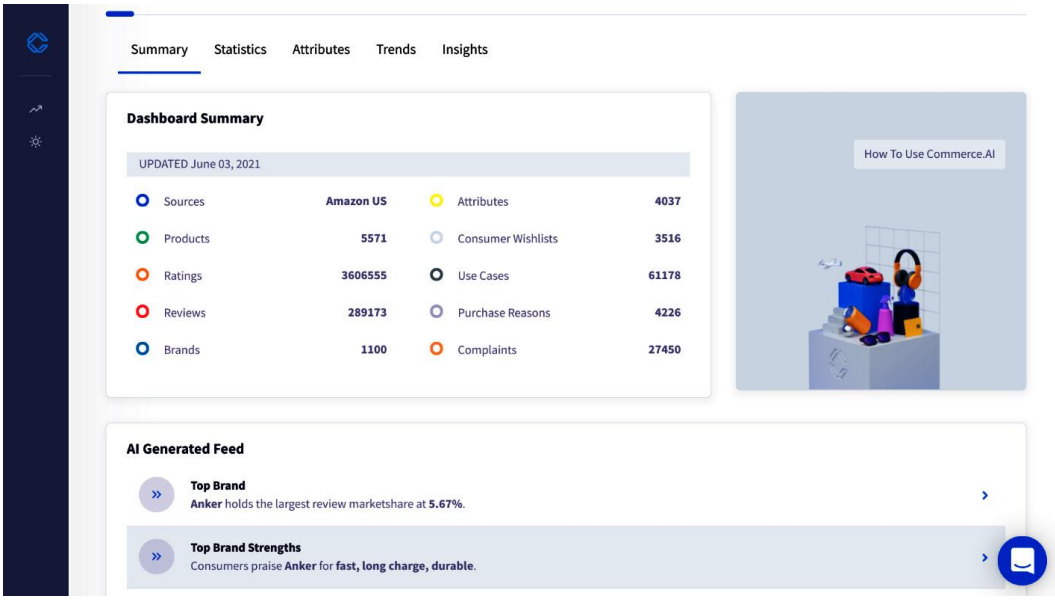

Figure 9.2 – Commerce.AI Dashboard Summary for wall chargers

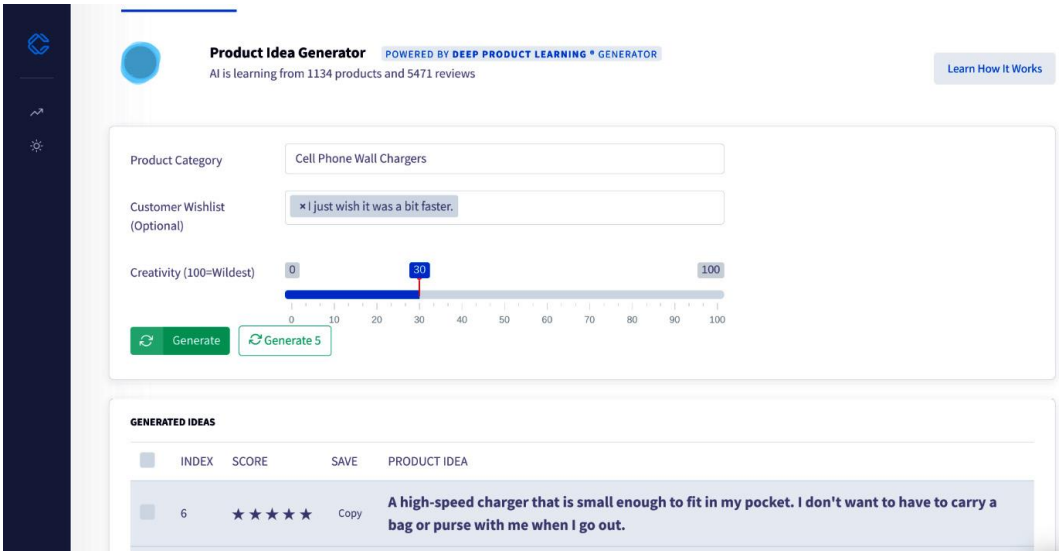

Figure 9.3 – The Commerce.AI Product Idea Generator

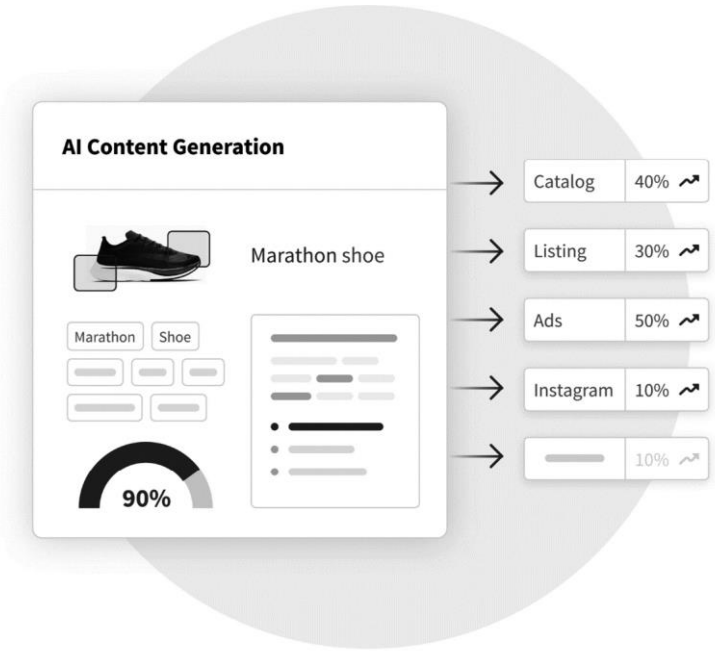

Figure 9.4 – A mockup showing Commerce.AI's AI Content Generation features

Chapter 10

Figures

Figure 10.1 – A mockup of Commerce.AI analysis for a telecom service

Chapter 11

Figures

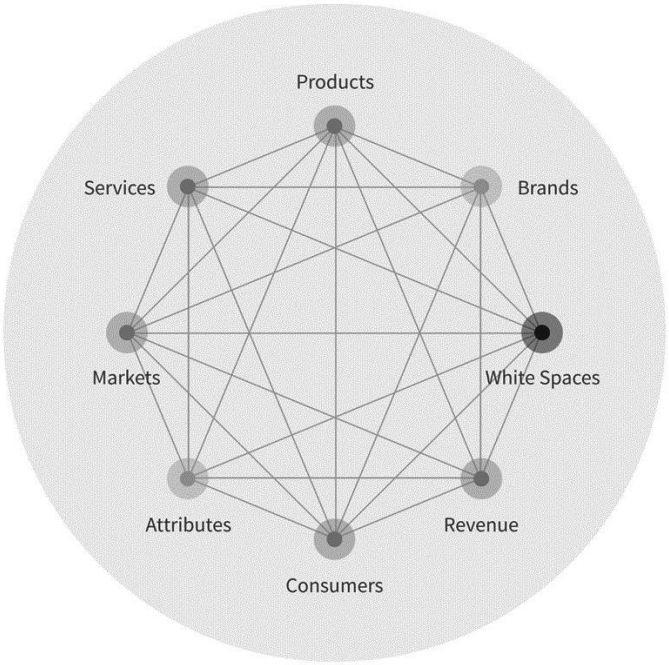

Figure 11.1 – A mockup of Commerce.AI white spaces analysis

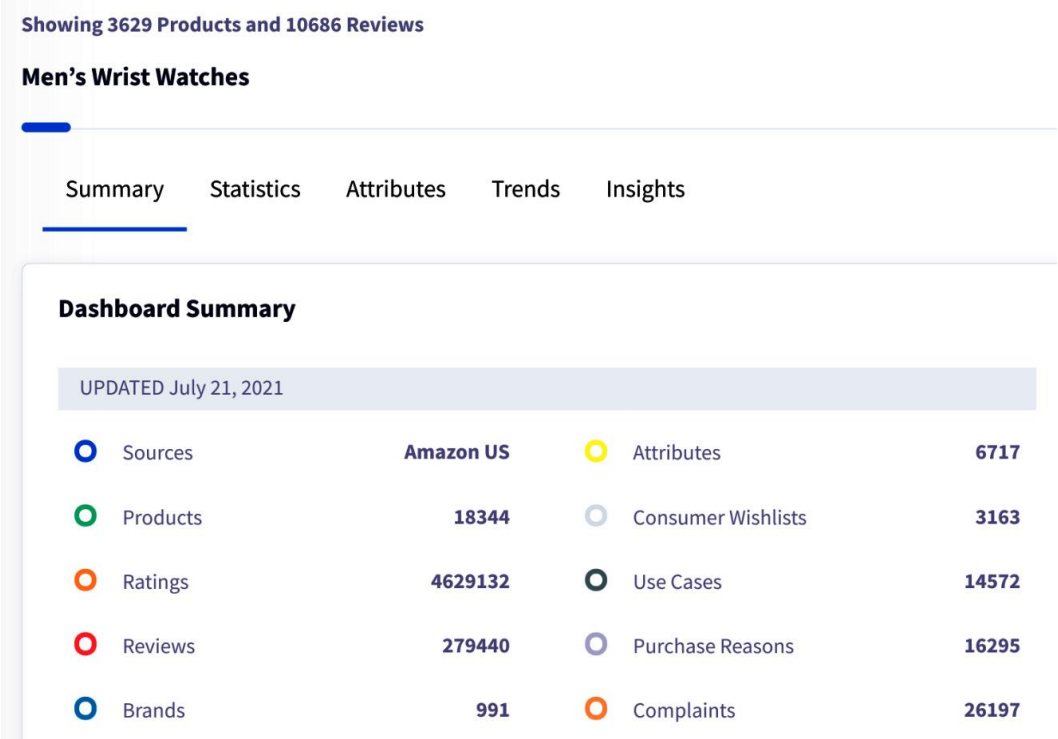

Figure 11.2 – A screenshot of Commerce.AI's market overview for men's wristwatches

Links

https://thejournal.com/articles/2021/07/01/virtual-reality-headsets-see-explosive-growth.aspx

Chapter 12

Figures

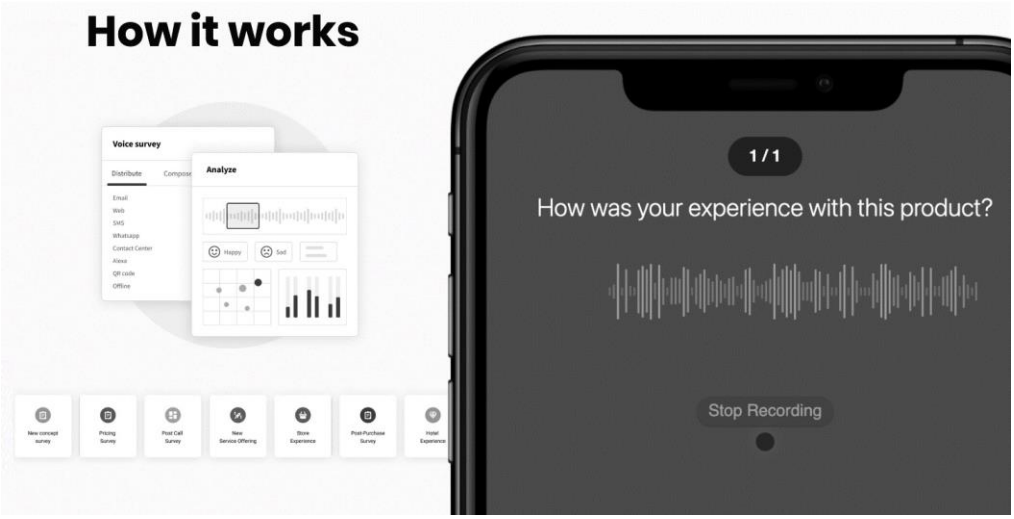

Figure 12.1 – A mock-up of a Commerce.AI voice survey