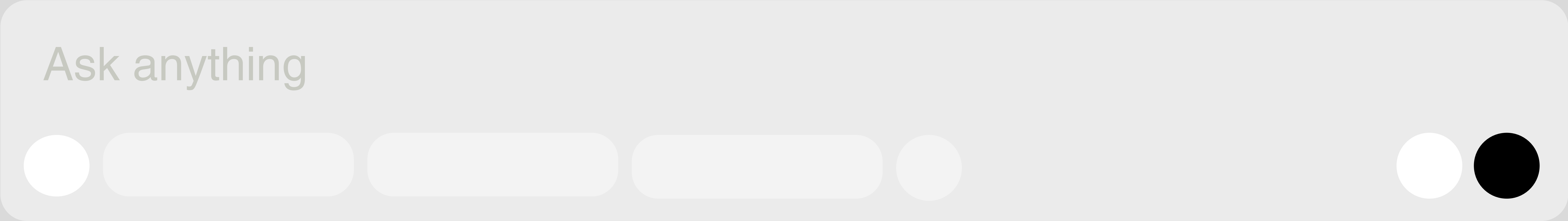
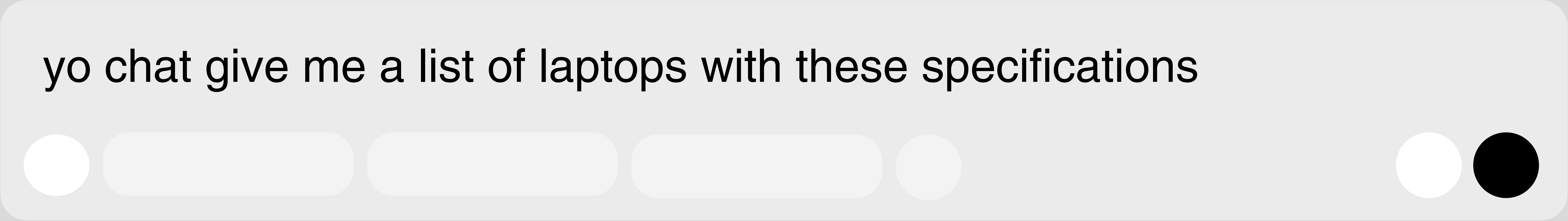
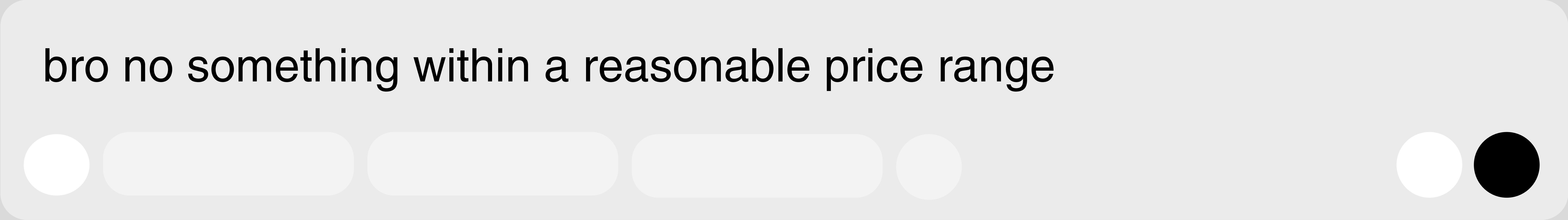
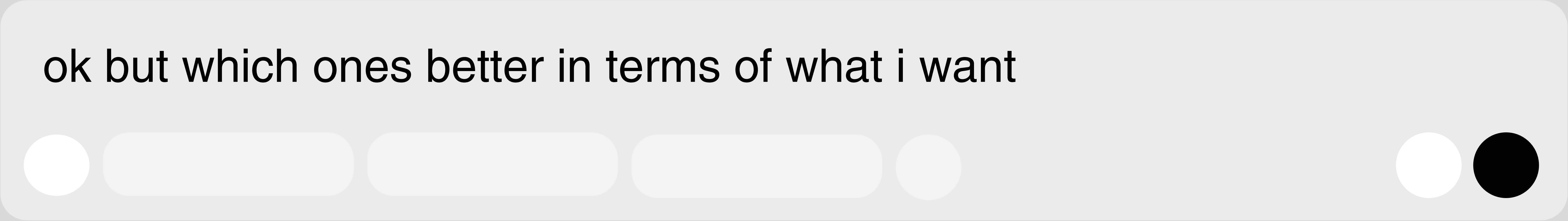

Ask anything



yo chat give me a list of laptops with these specifications



bro no something within a reasonable price range



ok but which ones better in terms of what i want

we’ve got
a *better*
solution
than
chatgpt

TECH GENIE
AT YOUR SERVICE

PC PART PICKER 3000

AN OVERVIEW

Ever wish you had a genie who could instantly tell you the price of your dream laptop and show you the best match in the market?

We deliver three core ML functions:

- **Descriptive:** K-Means Clustering to segment computers based on price, RAM, and other specs. PCA is a way of showing the clusters of the K-Means
- **Predictive:** LightGBM Regression to estimate **prices** from user inputs
- **Prescriptive:** K-Nearest Neighbors (KNN) to recommend similar listings with ranked similarity

DATA COLLECTION & PREP

RAW DATA

- **CSV file:** 8,064 marketplace listings (rows) x **135 raw Spanish-language columns**
- Encoded in **UTF-8-SIG with mixed metrics, units, and labels; .CSV file with 135 columns.**
- **No scraping or APIs; data ingested directly via pandas.read_csv.**

CLEANED DATA

1. Dropped Duplicates → df.duplicated().sum()

2. Standardized Column Names with custom slugify function

- → removed accents, lowercase, dropped stopwords (e.g., **Pantalla_Tamaño → pantalla_tamano)**

3. Dropped Unnamed Columns:

- df.drop(columns=['unnamed_0'])
- Full null or >70% null columns

4. Price Normalization

- Parsed "Precio_Rango" (e.g., "1.026,53 € – 2.287,17 €") into:
- `precio_min`, `precio_max`, and `precio_mean`
- **Dropped original string after parsing**

5. Numerical Extraction

- Created functions to **extract float from strings (e.g., RAM, CPU speed)**
- Remove thousands separators.
- Apply apply_cleaning_to_column() across many dirty fields

6. Standardized Screen Resolution

- Used regex to convert inconsistent resolution strings to “WIDTHXHEIGHT”

E.g., "4K (3.840 x 2.160)" → "3840x2160"

7. Offers Cleaning

- Convert strings like “200 ofertas” to 200.0 (float) for numeric ops.

HANDLING MISSING DATA

- **Used** df.isnull().sum() **and** missingno heatmaps
- **Aware of** Missing-Not-At-Random (MNAR) issues (e.g., screens missing in desktops). To solve, we handled it by isolating category-specific structures and then:

STRATEGY?

| 70% missing: dropped

- **30–70%:** conditional imputation or dropped
- **<30%:** imputed by product category using mean/mode

FEATURE ENGINEERING & SELECTION

MEAN PRICE: Extracted from raw price range string

```
def process_price_range(price_str)
```

Volume (cm^3)= height x width x depth

Category Mapping: Mapped devices to **English Classes (Ultrabook, Tower, All-in-One)**

FEATURE ENGINEERING

HALL OF FAME

FEATURE ENGINEERING & SELECTION

1

ONE-HOT ENCODING

for low-cardinality categorical fields.

2

ORDINAL ENCODING

for ordered features *like processor generation*

CATEGORICAL HANDLING

3

PCA & CORRELATION ANALYSIS

PCA to retain features explaining 90%+ variance + Removed highly correlated variables (Pearsons).

4

FINAL MATRIX

Final feature matrix optimized for model performance & interpretability.

MODEL TRAINING & VALIDATION

TECHNICAL APPROACH FOR SOLVING FUNCTIONALITIES

DESCRIPTIVE

K-MEANS CLUSTERING

- We used K-Means to segment the marketplace into natural product clusters
- Input features included normalized price, RAM, storage, and GPU type
- We validated cluster count using PCA + visual separation
- **Helps users explore differences across product segments (Ultrabook vs Desktop)**

PREDICTIVE

LIGHTGBM REGRESSION

- Chosen for **speed, accuracy, and native handling of missing values**
- Input: Engineered features like RAM, CPU model, GPU, brand, etc.
- Target: precio_mean (average of price range)
- Applied log-transform to the target for numerical stability
- RMSE ≈ 162 EUR, $R^2 \approx 0.89$
- Outputs price prediction + feature importance chart

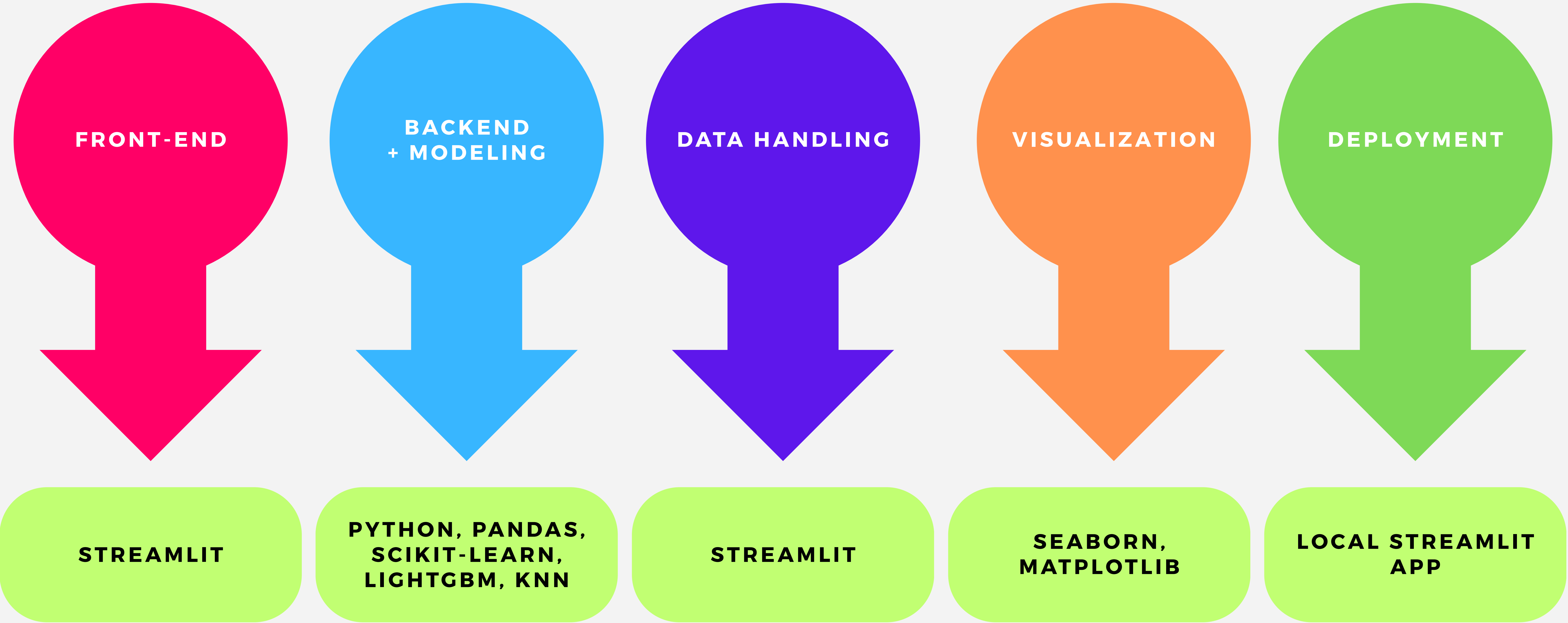
PRESCRIPTIVE

K-NEAREST NEIGHBORS

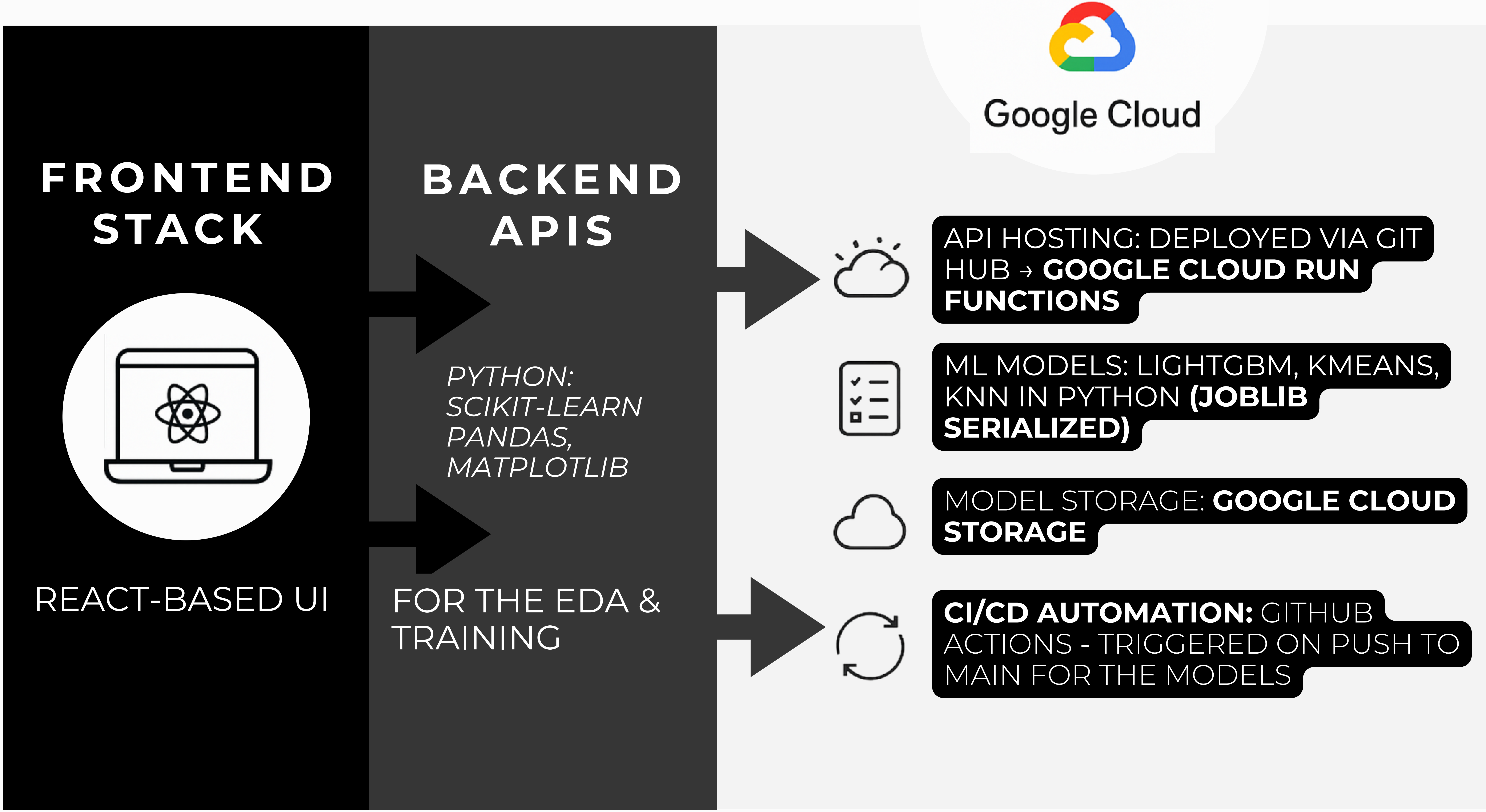
- **Recommend similar real-world laptop listings**
- Scaled user input and listing data using StandardScaler
- Used cosine similarity to match user config to **closest products**
- Returned **top-k results sorted by similarity, and included:**
- Predicted price
- Real listing price
- Side-by-side specs comparison

APP ARCHITECTURE & DEPLOYMENT

ARCHITECTURE



FRONT-END	BACKEND + MODELING	DATA HANDLING	VISUALIZATION	DEPLOYMENT
STREAMLIT	PYTHON, PANDAS, SCIKIT-LEARN, LIGHTGBM, KNN	STREAMLIT	SEABORN, MATPLOTLIB	LOCAL STREAMLIT APP



FRONTEND STACK

REACT-BASED UI

BACKEND APIS

PYTHON: SCIKIT-LEARN PANDAS, MATPLOTLIB

FOR THE EDA & TRAINING

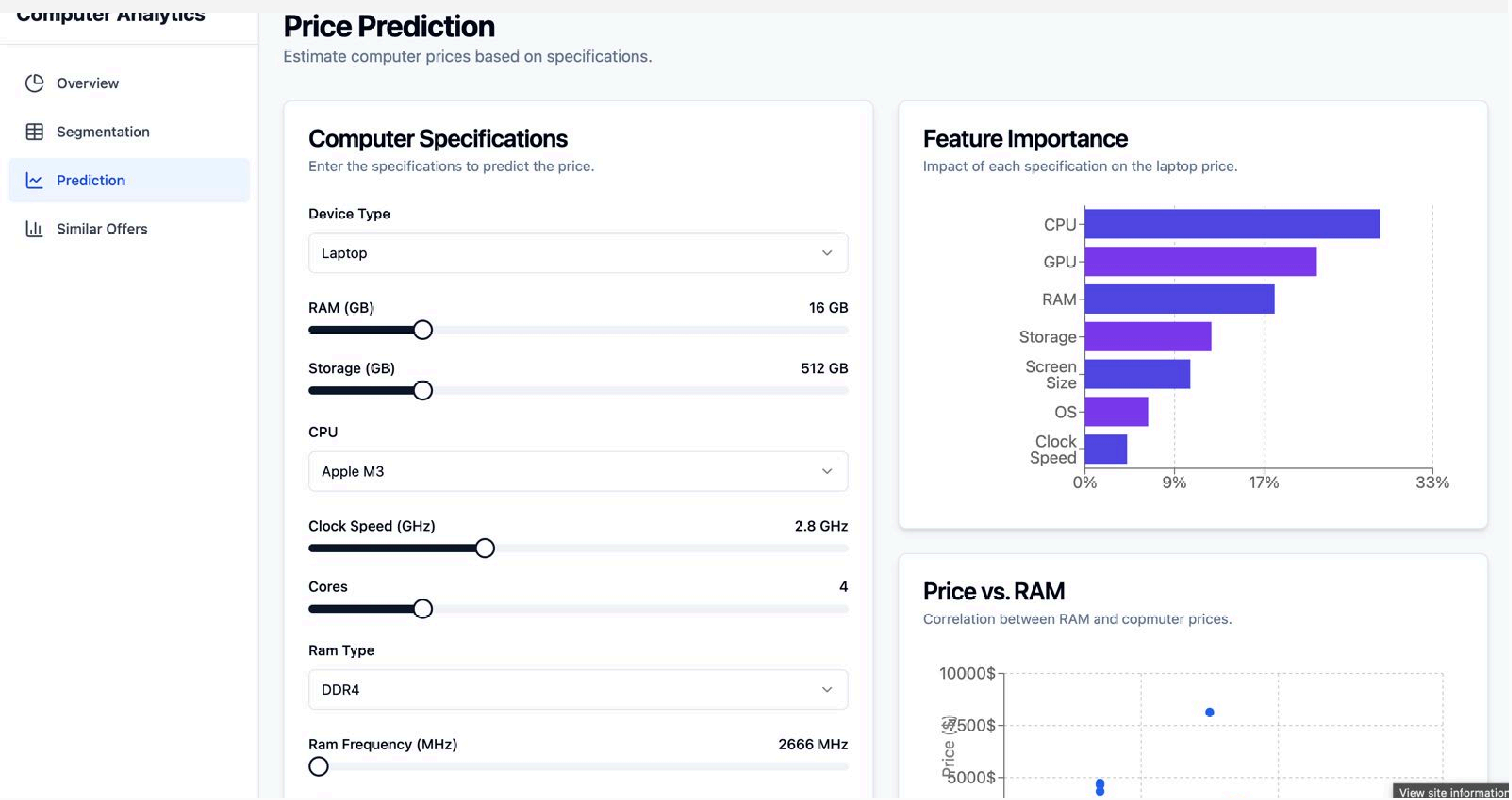
Google Cloud

API HOSTING: DEPLOYED VIA GIT HUB → **GOOGLE CLOUD RUN FUNCTIONS**

ML MODELS: LIGHTGBM, KMEANS, KNN IN PYTHON **(JOBLIB SERIALIZED)**

MODEL STORAGE: **GOOGLE CLOUD STORAGE**

CI/CD AUTOMATION: GITHUB ACTIONS - TRIGGERED ON PUSH TO MAIN FOR THE MODELS



Computer Analytics

Overview

Segmentation

Prediction

Similar Offers

Price Prediction

Estimate computer prices based on specifications.

Computer Specifications

Enter the specifications to predict the price.

Device Type: Laptop

RAM (GB): 16 GB

Storage (GB): 512 GB

CPU: Apple M3

Clock Speed (GHz): 2.8 GHz

Cores: 4

Ram Type: DDR4

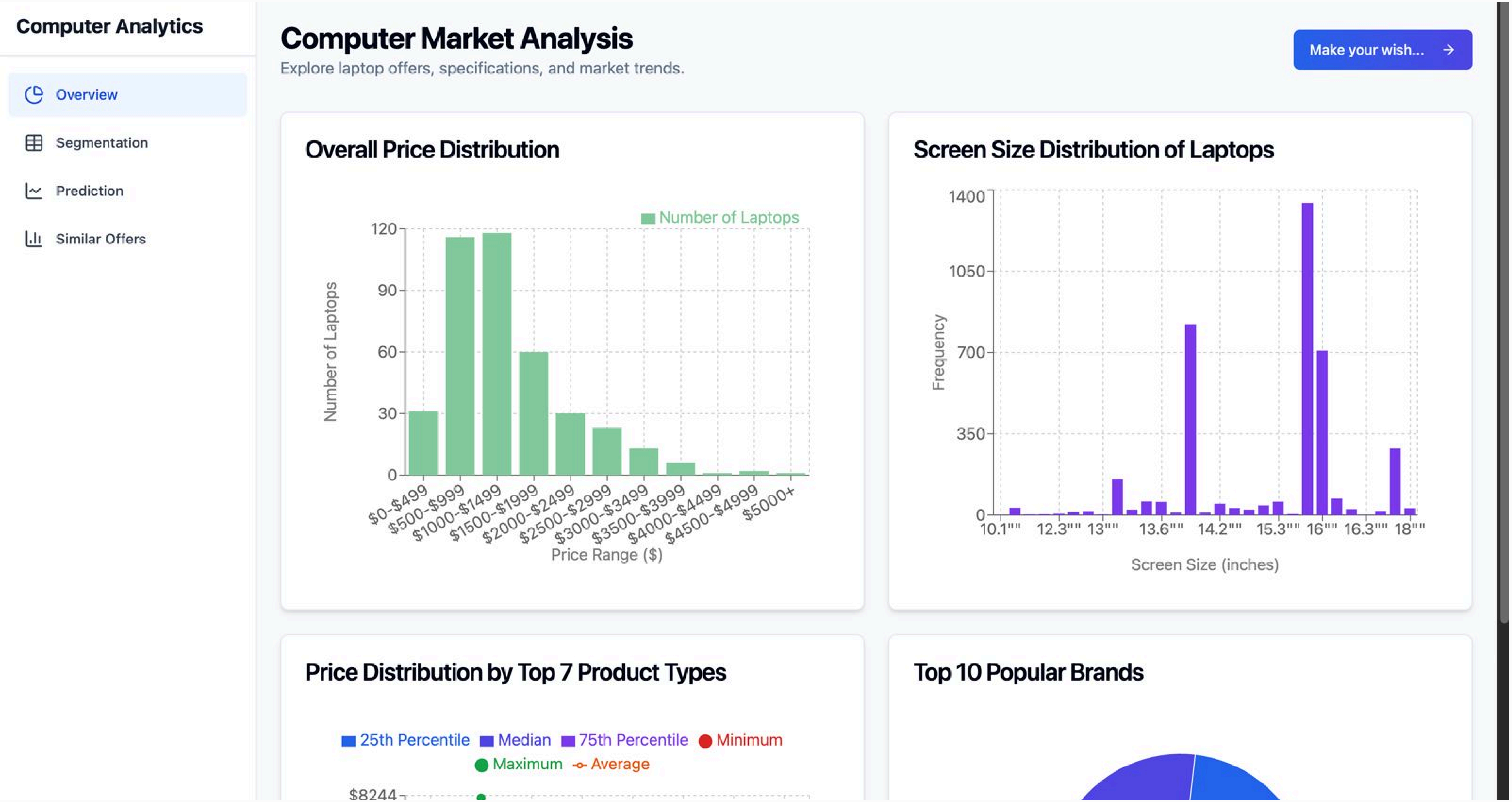
Ram Frequency (MHz): 2666 MHz

Feature Importance

Impact of each specification on the laptop price.

Price vs. RAM

Correlation between RAM and copmuter prices.



Computer Market Analysis

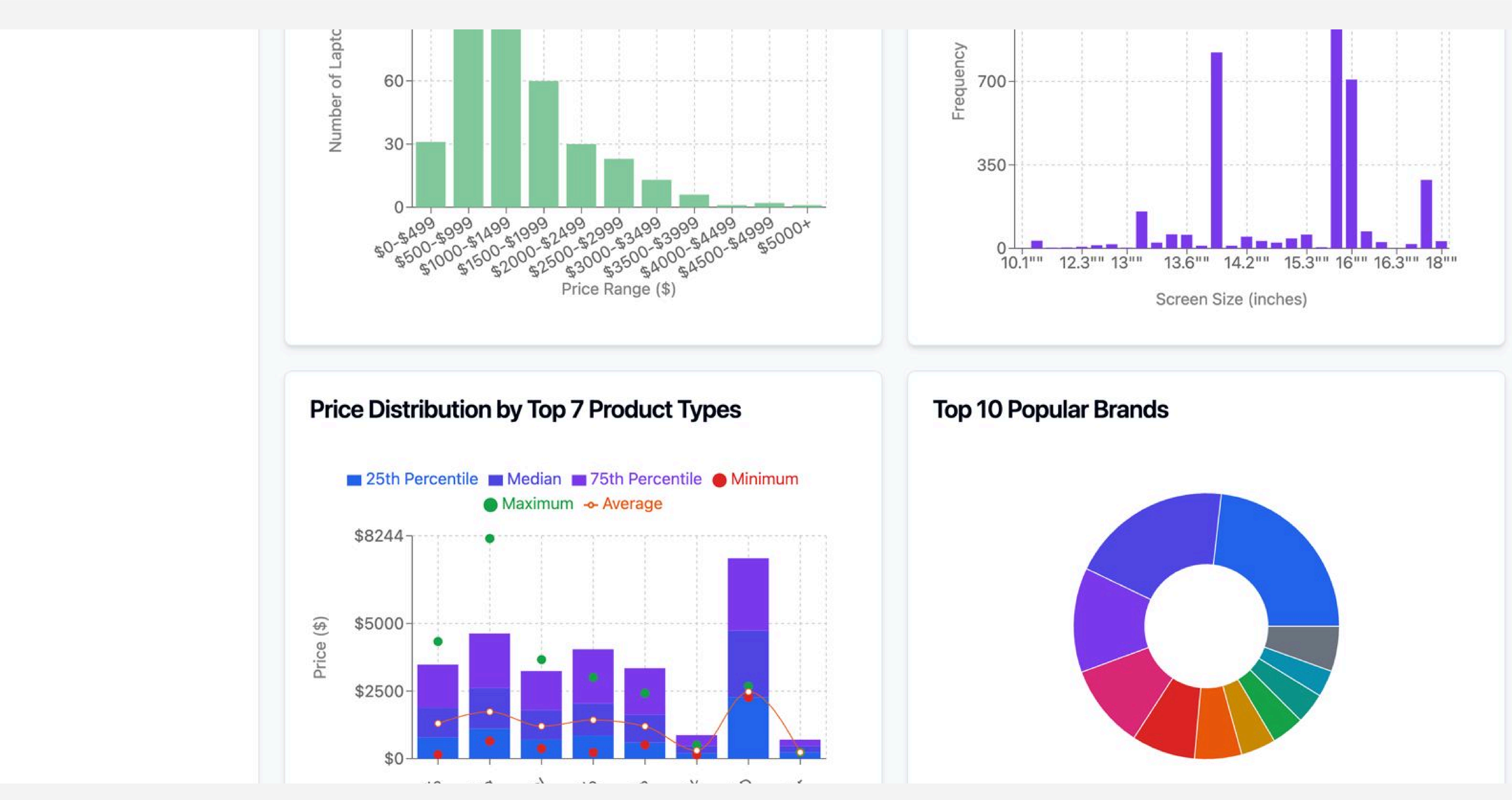
Explore laptop offers, specifications, and market trends.

Overall Price Distribution

Screen Size Distribution of Laptops

Price Distribution by Top 7 Product Types

Top 10 Popular Brands



Price Distribution by Top 7 Product Types

Top 10 Popular Brands

LIVE DEMONSTRATION

IMPROVEMENTS & NEXT STEPS

GENIE'S NEXT EVOLUTION

- **Live data integration** via APIs to keep listings up to date
- **Prediction confidence intervals** to show uncertainty
- **User-based personalization** using historical preferences
- **Model retraining** via feedback log ingestion
- **Multilingual toggle** to support Spanish/English UIs
- **Domain expansion** to peripherals, monitors, GPUs
- **Feature Feedback** to allow for constant improvements of model & the display of processed data.

THANK YOU