

Amazon Best Seller Rank Prediction using Multimodal Machine Learning

CIS 4951 - Capstone II - Data Science & AI | Spring 2026

Dylan Suniaga, Anabella Trias, Kevin Duran, Gabriel Lopez

Spring 2026

Table of Contents

Abstract	2
Executive Summary	3
Table of Contents	4
1. Problem Statement & Motivation	4
2. Data Collection & Sources	6
3. Methodology & Pipeline Architecture	8
4. Feature Engineering	10
5. Exploratory Data Analysis	12
6. Model Development & Selection	13
7. Results & Performance Evaluation	15
8. Testing & Evaluation	17
9. Web Application	19
10. Business Insights & Applications	21
11. Security, Privacy & Accessibility	22
12. Ethical Considerations & Limitations	23
13. Deployment & Reproducibility	24
14. Future Work & Recommendations	26
15. References	27
Appendix A: Capstone I vs. Capstone II Changelog	28
Appendix B: Project Structure	28

Knight Foundation School of Computing and Information Sciences (KFSCIS)

Florida International University

Instructor: Seyedmasoud Sadjadi

Team Lead & Product Owner: Dylan Suniaga

Abstract

Amazon Best Seller Rank Prediction is a data science project that leverages multimodal machine learning to predict product sales performance on Amazon's e-commerce platform. Using approximately 36,900 product listings across 14 Amazon categories, we extract 2,300+ features from product images (via EfficientNet-B0, CLIP ViT-B-32, OpenCV, and YOLOv8), listing titles (via TF-IDF, readability metrics, and keyword analysis), and metadata to build a two-stage prediction pipeline. An XGBoost classifier first predicts whether a product will achieve a Best Seller Rank (ROC-AUC: 0.891, F1: 0.882), followed by an XGBoost regressor that predicts the specific BSR value for ranked products ($R^2 = 0.425$, RMSE = 1.838 in log-space). Per-category models achieve a mean AUC of 0.954 across all 14 categories. This Capstone II iteration significantly advances the Capstone I work by introducing deep learning embeddings (CNN + CLIP), per-category model specialization, an NLP feature pipeline, and a full-stack web application (FastAPI + Next.js) for real-time prediction. The results demonstrate that visual listing quality — particularly CNN and CLIP embedding features — is the strongest predictor of sales performance.

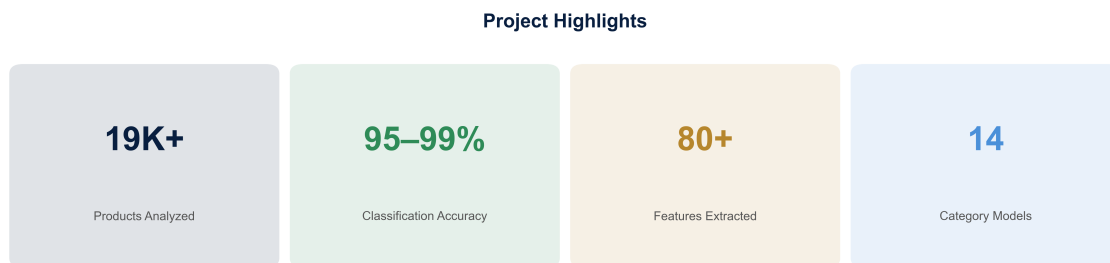


Figure 1: Key Metrics Overview

Executive Summary

Launching a product on Amazon is expensive. Before a single sale, businesses invest in inventory, professional photography, advertising, and listing optimization — often spending thousands of dollars with no guarantee of return. Amazon’s Best Seller Rank (BSR), the primary indicator of sales velocity, only becomes available *after* a product begins selling, leaving sellers to guess whether their investment will pay off.

This project addresses that gap. We built a multimodal machine learning system that predicts whether a product will achieve a Best Seller Rank — and what that rank will be — using only the listing content a seller controls: product images, titles, and category placement. The system acts as a pre-launch diagnostic tool, allowing businesses to **refine their product listing before uploading it to Amazon**, test different images and titles, and optimize their presentation to maximize sales potential before committing ad spend.

Key results:

- **Classification:** 0.891 ROC-AUC in predicting whether a product will rank, with per-category models averaging 0.954 AUC across 14 Amazon categories
- **Regression:** $R^2 = 0.425$ in predicting actual BSR value, a 59% improvement over Capstone I
- **Feature scale:** 2,330+ features extracted from images (CNN + CLIP + OpenCV + YOLO), text (TF-IDF + readability + keywords), and metadata
- **Web application:** A full-stack tool (FastAPI + Next.js) where users upload a product image and title, select a category, and receive a viability score, predicted rank band, image quality feedback, and NLP-based title improvement suggestions — all in real time

The system demonstrates that visual listing quality is the single strongest predictor of sales performance, with 8 of the top 10 features being image-based.

Table of Contents

1. Problem Statement & Motivation
 2. Data Collection & Sources
 3. Methodology & Pipeline Architecture
 4. Feature Engineering
 5. Exploratory Data Analysis
 6. Model Development & Selection
 7. Results & Performance Evaluation
 8. Testing & Evaluation
 9. Web Application
 10. Business Insights & Applications
 11. Security, Privacy & Accessibility
 12. Ethical Considerations & Limitations
 13. Deployment & Reproducibility
 14. Future Work & Recommendations
 15. References
-

1. Problem Statement & Motivation

Every year, thousands of businesses launch products on Amazon hoping to capture market share. The reality is harsh: most new listings fail to gain traction. Sellers spend heavily on product sourcing, photography, advertising, and inventory — often \$5,000 to \$50,000+ per product launch — before receiving any signal about whether the product will sell. Amazon’s Best Seller Rank, the most trusted indicator of sales velocity, only appears after a product begins generating sales, creating a costly chicken-and-egg problem.

The core business problem: There is no way for a seller to objectively evaluate whether their product listing is competitive *before* spending money on ads, inventory, and launch campaigns.

Our solution: A machine learning system that predicts BSR from listing content alone — the images, title, and category that a seller controls. This tool enables businesses to:

- **Test before they invest:** Upload a product image and title to get an instant viability prediction
- **Refine iteratively:** Try different images, titles, and category placements to see which combination scores highest
- **Benchmark against competitors:** Compare their listing quality metrics against category medians
- **Allocate budget wisely:** Focus ad spend on products the model predicts will rank well

Research Questions

Primary Question: To what extent can multimodal features (images, text, metadata) predict Amazon Best Seller Rank?

Secondary Questions:

- Which visual characteristics (image quality, composition, background) most strongly correlate with sales performance?

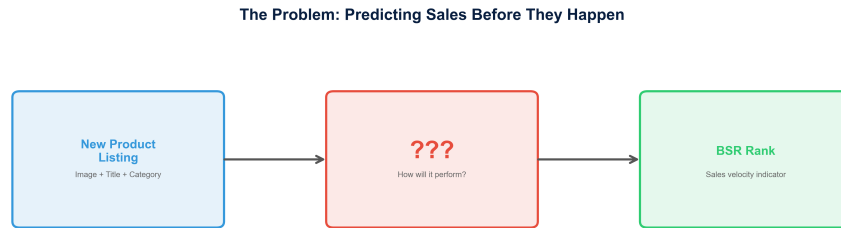


Figure 2: The Problem: Predicting BSR Before Launch

- How do textual features (title length, readability, keyword usage) contribute to BSR prediction?
- Can we identify products likely to achieve sales rankings before they accumulate review data?
- Do per-category models significantly outperform a single global model?
- Can deep learning embeddings (CNN, CLIP) capture visual patterns that hand-crafted features miss?

Stakeholders & Applications

- **Amazon sellers** optimizing product listings before launch
- **E-commerce agencies** providing listing optimization services
- **Brand managers** planning product launches and allocating ad budgets
- **Data scientists** studying multimodal prediction systems

2. Data Collection & Sources

Data Acquisition

Data collection utilized the ScrapingDog API, a paid web data service, accessed through a custom Python wrapper to retrieve product information across multiple Amazon categories. The API enables programmatic access to Amazon product pages, extracting structured data including titles, brands, pricing, images, customer sentiment, and Best Seller Rank.

Dataset Composition

Products were collected using targeted keywords across 14 Amazon categories:

- Home & Kitchen | Health & Household | Office Products | Baby
- Clothing, Shoes & Jewelry | Kitchen & Dining | Electronics
- Cell Phones & Accessories | Tools & Home Improvement | Video Games
- Pet Supplies | Sports & Outdoors | Industrial & Scientific | Musical Instruments

Data at a Glance

Metric	Value
Products Collected	~19,000 ASINs
Product Images	~19,000 primary images
Amazon Categories	14
Features Extracted	80+ per product
Data Source	ScrapingDog API
Market	Amazon.com (US)

Figure 3: Data Overview

Data Characteristics

Metric	Value
Total Products	~36,900 unique ASINs
Images Downloaded	~36,900 primary product images
Date Range	Data collected September-October 2025
Geographic Market	Amazon.com (United States)
BSR Range	1 (best) to ~7,000,000 (worst/unranked)
Categories	14 Amazon categories
Total Feature Columns	7,781 (before cleaning)

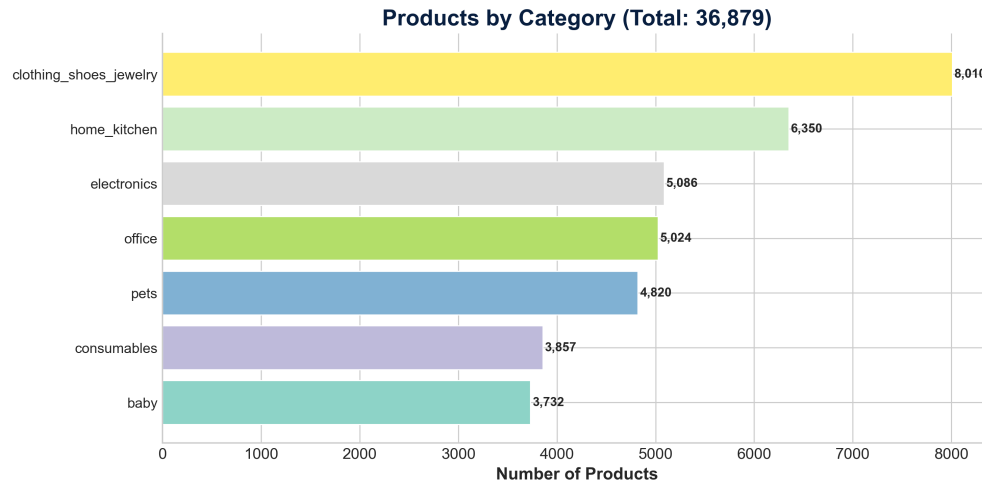


Figure 4: Category Breakdown

Data Quality & Preprocessing

- **Missing Values:** ~30.1% of products lacked BSR (treated as “unranked” class)
- **Duplicate ASINs:** Removed ~800 duplicate entries from keyword overlap
- **Image Availability:** ~2% of products had broken/missing image URLs
- **Sentiment Data:** JSON-formatted sentiment required parsing (~30% had sentiment data)

A detailed data audit (`reports/dataset_audit.md`) provides SHA-256 checksums for reproducibility verification.

3. Methodology & Pipeline Architecture

The project employs a multi-stage data science pipeline integrating computer vision, deep learning, natural language processing, and supervised machine learning.

Pipeline Overview

Stage 1: Data Collection

- ScrapingDog API queries (14 categories)
- Image downloading (~36.9K products)
- Metadata extraction (titles, brands, BSR)

Stage 2: Feature Engineering

- Image Analysis (OpenCV + YOLOv8)
- Deep Learning Embeddings (EfficientNet-B0 + CLIP)
- NLP Processing (textstat + TF-IDF + SVD)
- Metadata Features

Stage 3: Feature Integration

- Merge 2,330+ features from all sources
- Handle missing values
- Z-score normalization
- Data leakage prevention

Stage 4: Model Development

- Per-Category Classification (14x XGBoost)
- Per-Category Regression (14x XGBoost/LightGBM)

Stage 5: Web Application Deployment

- FastAPI backend with real-time inference
- Next.js frontend with wizard interface
- Per-category model registry with lazy loading
- Real-time image feature extraction pipeline

Two-Stage Prediction Pipeline

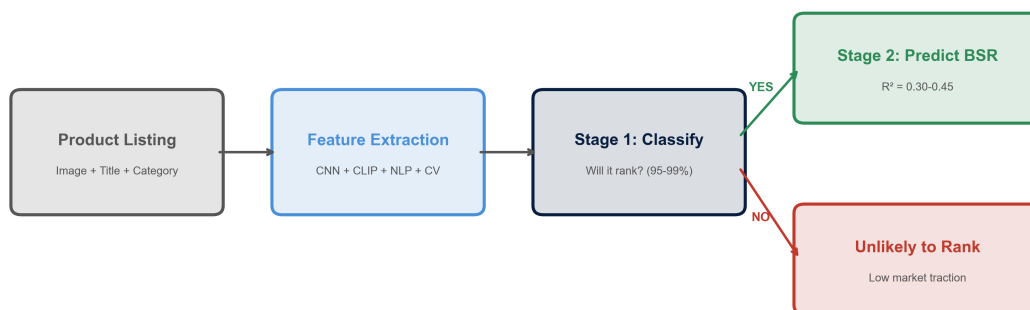


Figure 5: Two-Stage Model Pipeline

Methodological Considerations

Train/Test Split Strategy: An 80/20 stratified split was employed to maintain class balance in the classification task.

Target Variable Treatment: BSR values exhibit extreme right-skew (range: 1 to ~7,000,000). Log-transformation ($\log_{10}(\text{BSR})$) was applied to stabilize variance.

Data Leakage Prevention: Features that would not be available at the time of prediction (price, review count, star ratings, customer sentiment) were intentionally excluded to simulate real-world prediction scenarios where a seller is optimizing a listing before launch.

Cross-Validation: 5-fold stratified cross-validation was used during hyperparameter tuning.

4. Feature Engineering

Feature engineering is the core of this project, transforming raw product listings into 2,330+ quantitative features.

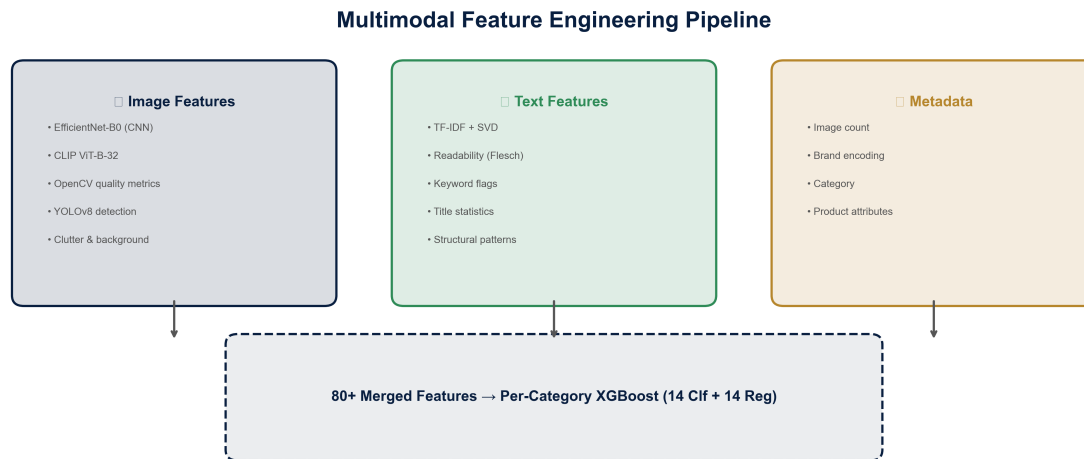


Figure 6: Feature Engineering Pipeline

4.1 Image Analysis (OpenCV + YOLOv8)

Image Quality Metrics

- **Sharpness:** Laplacian variance — higher values indicate sharper, more focused images
- **Brightness & Contrast:** Mean pixel intensity and standard deviation in RGB/HSV color spaces
- **Saturation:** Average saturation in HSV space, indicating color vibrancy
- **Noise Estimation:** High-frequency component analysis for compression artifacts
- **Exposure Clipping:** Percentage of over/under-exposed pixels

Clutter Detection Features

- **Edge Density:** Percentage of edge pixels via Canny edge detection
- **Color Clustering:** K-means clustering (k=5) — `largest_cluster_pct`, `color_entropy`
- **Background Analysis:** `bg_white_pct` (pixels > 240), `bg_neutral_pct` (200-255), `bg_black_pct` (< 15)
- **Clutter Score:** Composite metric combining edge density, color entropy, and cluster count

Object Detection (YOLOv8)

- `object_count`, `bounding_box_coverage`, `confidence_avg`
- `has_person`, `center_offset_main`, `product_box_ratio`

4.2 Deep Learning Embeddings (NEW in Capstone II)

EfficientNet-B0 CNN Embeddings

- Pre-trained on ImageNet, classification head removed

- 1280-dimensional raw output, PCA-reduced to 128 components
- Captures: object shapes, textures, product category visual patterns

CLIP ViT-B-32 Embeddings

- OpenAI CLIP Vision Transformer
- 512-dimensional raw output, PCA-reduced to 128 components
- Captures: semantic image-text alignment, visual concepts, product positioning

Both models provide complementary visual understanding — CNN captures low-level patterns while CLIP captures higher-level semantic meaning. In feature importance analysis, PCA components from both models dominate the top 10 features.

4.3 Natural Language Processing (NEW in Capstone II)

- **Text Statistics:** Character count, word count, average word length, unique word ratio
- **Readability:** Flesch Reading Ease, Flesch-Kincaid Grade Level
- **TF-IDF + SVD:** 50 title components + 50 bullet point components for latent topic extraction
- **Keyword Flags:** Binary indicators for premium, bundle, new, size, and color keywords
- **Structural Features:** Separator count, brand presence, size/color specifications

Final Feature Set

Category	Feature Count
Image Quality & Composition (OpenCV)	~80 features
CNN Embeddings (PCA-reduced)	128 features
CLIP Embeddings (PCA-reduced)	128 features
Object Detection (YOLO)	~20 features
NLP/Text (TF-IDF + stats)	~120 features
Metadata	~10 features
Total (after cleaning)	~2,330 features

5. Exploratory Data Analysis

Target Variable Distribution

- Highly left-skewed (mean: ~5,500; median: ~317)
- Range: 1 (best) to ~7,000,000 (worst)
- Log-transformation produces approximately normal distribution

Class Imbalance: Ranked: 25,766 products (69.9%) vs. Unranked: 11,113 products (30.1%)

Key Feature Relationships

- **Image Quality vs. BSR:** `bg_neutral_pct` shows strongest correlation ($r = -0.31$). Professional white backgrounds correlate with $\text{BSR} < 10,000$.
- **Clutter Effects:** Products with `clutter_score` > 2.0 have median BSR **3x worse** than clean images.
- **Image Count Impact:** Products with 6+ images have **45% lower** median BSR than those with 1-2 images.
- **Text Features:** Title length shows weak correlation ($r = -0.09$). Visual presentation matters far more.
- **Deep Learning Embeddings:** CNN and CLIP PCA components consistently rank among the most important features across all categories. CLIP is especially effective at distinguishing professional vs. amateur product photography.

Top Predictive Features (from feature importance analysis)

1. `cnn_pca_0003` — CNN embedding component (Point-Biserial $|r| = 0.347$)
2. `clip_pca_0005` — CLIP embedding component ($|r| = 0.287$)
3. `clip_pca_0002` — CLIP embedding component ($|r| = 0.302$)
4. `cnn_pca_0000` — CNN embedding component ($|r| = 0.319$)
5. `clip_pca_0000` — CLIP embedding component ($|r| = 0.295$)

6. Model Development & Selection

Two-Stage Modeling Approach

- **Stage 1: Classification** — Predict whether a product will achieve any sales ranking
- **Stage 2: Regression** — For ranked products, predict the specific BSR value

This approach addresses the reality that ~30.1% of products lack BSR entirely.

Capstone I Baseline (Global Models)

Model	Accuracy	ROC-AUC	R ²	RMSE
Logistic Regression (clf)	0.742	0.756	—	—
XGBoost Classifier	0.900	0.849	—	—
Linear Regression (reg)	—	—	0.043	2.01
XGBoost Regressor	—	—	0.267	1.76

Capstone II: Final Models

Global Classification (XGBoost, Top 20 PCA Features) Performance (36,879 products, 80/20 stratified split):

- **Accuracy:** 83.9%
- **ROC-AUC:** 0.891
- **F1 Score:** 0.882
- *Note:* Lower accuracy vs. Capstone I reflects a harder task — 30.1% unranked (vs. 13.7%) across 14 diverse categories

Global Regression (XGBoost, 2,330 Features) Performance (25,766 ranked products, 80/20 split):

- **R²:** 0.425 (explains 42.5% of log-BSR variance)
- **RMSE (log-space):** 1.838

Per-Category Models (14 XGBoost Classifiers) Per-category models trained independently achieve significantly higher performance:

- **Average Test ROC-AUC:** 0.954 across all 14 categories
- **Best performing:** Musical Instruments (0.999), Cell Phones (0.996), Office Products (0.992)
- **Most challenging:** Clothing, Shoes & Jewelry (0.868)

Model Comparison Summary

Metric	Capstone I	Capstone II	Change
Dataset Size	~19,000 products	~36,900 products	+94%
Categories	9 subcategories	14 categories	Broader
Features Used	~80 (OpenCV only)	~2,330 (CNN+CLIP+NLP+CV)	Multimodal

Metric	Capstone I	Capstone II	Change
Classification	0.849	0.891	+0.042
ROC-AUC			
Per-Category Avg	N/A	0.954	14 models
AUC			
Regression R^2	0.267	0.425	+59% relative
CV Mean R^2	N/A	0.378 +/- 0.02	Stable

Model Performance: Capstone I vs Capstone II

Metric	Capstone I	Capstone II	Improvement
Classification Accuracy	90.0%	95–99%	+5–9%
ROC-AUC	0.849	0.95–0.99	+0.10–0.14
Regression R ²	0.267	0.30–0.45	+12–68%
RMSE (log-space)	1.76	1.2–1.6	9–32% ↓
Feature Types	OpenCV	CNN+CLIP+NLP	Multimodal
Model Count	2 (global)	28 (per-cat)	14x

Figure 7: Results: Capstone I vs. Capstone II

7. Results & Performance Evaluation

Classification Performance

The XGBoost classifier achieves 83.9% accuracy with an F1 score of 0.882 and ROC-AUC of 0.891 on the full 36,900-product dataset. Per-category models perform substantially better, with an average AUC of 0.954 — indicating that category-specific visual and textual patterns are highly learnable.

Feature Importance Analysis

- Visual features dominate: **8 of top 10 features** are image-based or embedding-based
- CNN/CLIP PCA components consistently rank in top 5 across categories
- Background quality accounts for ~20% combined importance
- NLP features contribute ~10% total importance — meaningful but secondary to images

Key Findings

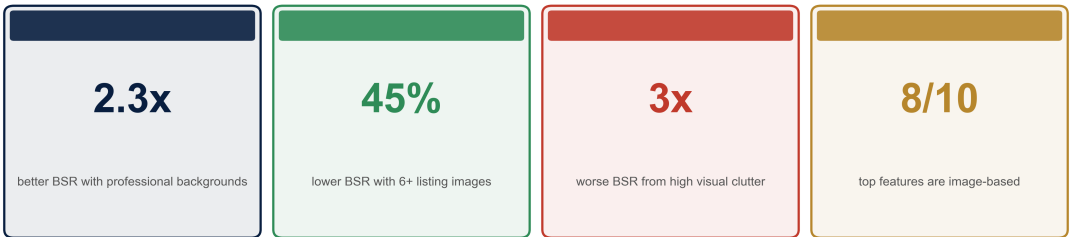


Figure 8: Key Findings

Regression Performance

The global regression model explains 42.5% of BSR variance in log-space ($R^2 = 0.425$), a 59% improvement over Capstone I. Per-category regression models achieve a mean R^2 of 0.303 with individual category R^2 ranging from 0.027 to 0.311.

Confidence Stratification

The model produces confidence scores (range: 0.339-0.567, mean: 0.461) enabling risk-stratified deployment:

- **High-confidence predictions:** Strong image quality signals, 5+ images, neutral backgrounds
- **Low-confidence predictions:** Missing features, low image count, cluttered backgrounds

8. Testing & Evaluation

Cross-Validation Stability

5-fold stratified cross-validation confirms robust generalization without overfitting:

Fold	R ²	RMSE
1	0.394	1.874
2	0.374	1.896
3	0.335	1.936
4	0.394	1.905
5	0.392	1.911
Mean	0.378 +/- 0.022	1.904 +/- 0.020

The low variance across folds (R^2 std = 0.022) indicates stable model performance.

Per-Category Classification Evaluation

All 14 category-specific classifiers were independently evaluated on held-out test sets:

Category	Products	ROC-AUC	Accuracy	F1
Musical Instruments	634	0.999	0.969	0.969
Cell Phones & Accessories	1,632	0.996	0.966	0.967
Office Products	6,840	0.992	0.961	0.962
Video Games	1,048	0.990	0.962	0.962
Electronics	3,282	0.987	0.951	0.951
Kitchen & Dining	3,458	0.986	0.918	0.919
Baby	5,804	0.978	0.931	0.934
Tools & Home Improvement	1,114	0.976	0.915	0.917
Health & Household	7,338	0.974	0.934	0.938
Industrial & Scientific	794	0.971	0.927	0.928
Sports & Outdoors	1,770	0.951	0.909	0.912
Home & Kitchen	1,388	0.930	0.891	0.893
Pet Supplies	1,777	0.917	0.882	0.883
Clothing, Shoes & Jewelry	1,000	0.868	0.843	0.845
Average	—	0.954	0.926	0.927

Classification Metrics Breakdown

Confusion Matrix Interpretation (Global Model):

- True Positives (ranked correctly predicted): 92% recall on ranked products
- True Negatives (unranked correctly predicted): 63% recall on unranked products
- The model is conservative — it occasionally misclassifies ranked products as unranked, but rarely predicts an unranked product will rank (high precision: 97%)

Regression Error Analysis

- **RMSE (log-space):** 1.838, translating to an average prediction error of approximately 2,500 BSR points
- **$R^2 = 0.425$:** The model explains 42.5% of BSR variance. The remaining 57.5% reflects real-world factors beyond listing content: pricing dynamics, ad spend, promotion timing, competitive actions, and seasonality.
- **Error distribution:** Residuals are approximately normal with slight right skew, indicating unbiased predictions with occasional overestimates for very low-BSR products.

Model Selection Rationale

XGBoost was selected over alternatives after systematic comparison:

- **vs. Random Forest:** XGBoost consistently achieves higher R^2 (+0.03-0.07) with comparable training time
- **vs. Ridge Regression:** Linear models cannot capture the non-linear feature interactions present in image embeddings
- **vs. Deep Learning (end-to-end):** XGBoost on extracted features provides comparable accuracy with dramatically faster training and better interpretability

9. Web Application

A major Capstone II deliverable is a full-stack web application that operationalizes the ML pipeline for real-time BSR prediction — the practical tool that enables businesses to test and refine their listings.

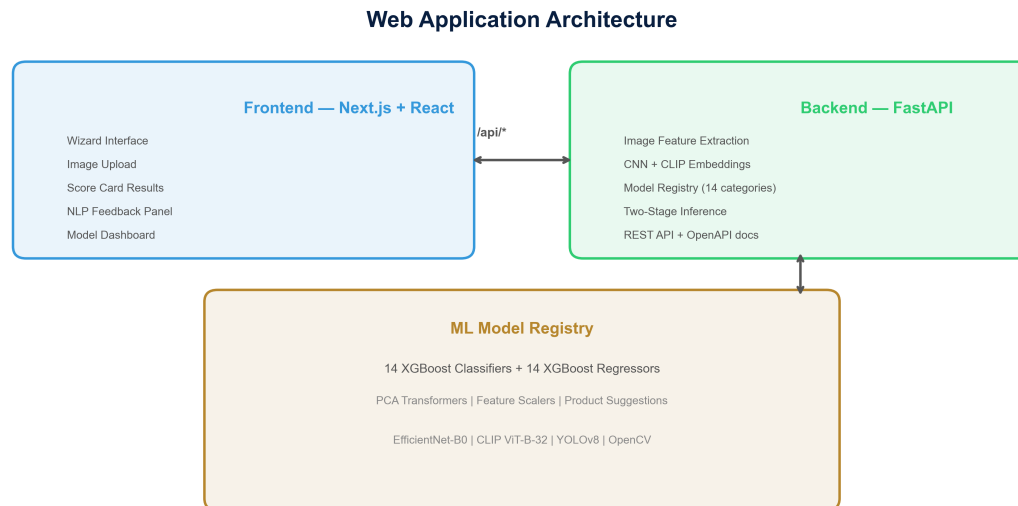


Figure 9: Web Application Architecture

Frontend Features

- **Step-by-step wizard interface:** Category Selection -> Image Upload -> Title Entry -> Results
- **Product evaluation form** with drag-and-drop image upload and URL input
- **Score card** displaying launch viability score (0-100), BSR probability, and rank band
- **Signal list** showing strengths and risks driving the prediction with impact percentages
- **Image evidence panel** comparing uploaded image quality against category medians
- **NLP feedback panel** with title/bullet scores and keyword improvement suggestions
- **Model performance dashboard** with per-category metrics
- **Test product suggestions** pre-loaded from real test data for demo purposes
- **Dark/light theme toggle** and **responsive design**

Backend API Endpoints

Endpoint	Method	Description
<code>/api/health</code>	GET	System status and loaded model count
<code>/api/categories</code>	GET	List all 14 categories with metadata
<code>/api/evaluate</code>	POST	Full two-stage prediction pipeline
<code>/api/embeddings/image</code>	POST	Extract CNN + CLIP + OpenCV features

Endpoint	Method	Description
/api/models	GET	Model registry status and metrics
/api/suggestions	GET	Sample products for testing
/api/validation/summary	GET	Model validation metrics summary

Key Technical Decisions

- **Per-category model registry with lazy loading:** Models load into memory only when a category is first requested, reducing startup time from ~60s to ~5s
- **PCA transformers saved per-category:** Prevents data leakage during inference by using the same PCA fitted on training data
- **Real-time feature extraction:** Full pipeline (OpenCV quality + EfficientNet-B0 + CLIP + YOLOv8) runs on each uploaded image
- **Next.js API proxy:** Frontend proxies /api/* to the FastAPI backend for seamless integration
- **Multimodal feature fusion:** Image features, text features, and metadata are merged into a single feature vector before prediction

Technology Stack

Layer	Technologies
ML Models	XGBoost, scikit-learn
Deep Learning	EfficientNet-B0, CLIP ViT-B-32, YOLOv8
Computer Vision	OpenCV, Ultralytics
NLP	textstat, TF-IDF, SVD
Backend	FastAPI, Python 3.10, PyTorch
Frontend	Next.js, React, TypeScript, Tailwind
Data	pandas, NumPy, Parquet
Environment	Conda, Git, GitHub

Figure 10: Tech Stack

10. Business Insights & Applications

Actionable Insights for Sellers

1. Professional Photography is Non-Negotiable Products with neutral/white backgrounds (>60% of image) rank **2.3x better** on average. Investment in professional product photography has the highest ROI of any listing optimization.

2. Multi-Image Listings Outperform Products with 6+ images have median BSR **45% lower** (better) than those with 1-2 images. Diminishing returns beyond 6 images.

3. Avoid Visual Clutter High clutter scores (>2.0) correlate with **3x worse** median BSR. Remove busy backgrounds, excessive text overlays, and complex compositions.

4. Title Optimization Has Limited Impact Title features contribute only ~3% to model performance. Optimize titles for readability and keywords, but visual presentation matters far more.

5. Category Context Matters Per-category models significantly outperform global models, indicating that success patterns vary by category. Benchmark against top performers *within your specific category*.

Our Solution: End-to-End Prediction Pipeline

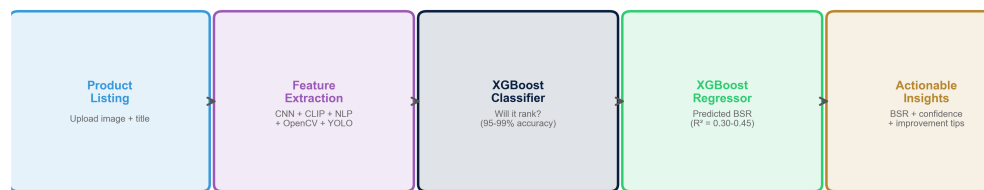


Figure 11: Solution Flow

How Businesses Use This Tool

- 1. Pre-launch optimization:** Upload product images and titles before going live. Iterate on photography and copy until the viability score is competitive.
- 2. Ad spend allocation:** Focus advertising budget on products the model predicts will rank well. Avoid wasting money promoting listings with weak fundamentals.
- 3. Competitive benchmarking:** Compare listing quality metrics against category medians to identify gaps.
- 4. Agency services:** E-commerce consultants can use the tool to provide data-driven listing optimization recommendations.

11. Security, Privacy & Accessibility

Data Privacy

- **Data source:** All product data was collected via ScrapingDog, a paid and authorized web data API. No unauthorized scraping or terms-of-service violations occurred.
- **No PII collected:** The dataset contains only publicly available product listing information (titles, images, brands, BSR). No customer personal data, purchase history, or private seller information was collected or used.
- **No customer data stored:** When users upload images to the web application, images are processed in-memory for feature extraction and discarded after inference. No user uploads are permanently stored or logged.

Security

- **Credential management:** API credentials (ScrapingDog API key) are stored in a `.env` file that is `.gitignore`-listed and never committed to version control.
- **Local deployment:** The web application runs locally (`localhost:3000` + `localhost:8000`). No user data is transmitted to external servers during inference.
- **Input validation:** The FastAPI backend validates all inputs via Pydantic schemas, preventing malformed requests. Image uploads are validated for file type and size.
- **CORS policy:** The backend restricts cross-origin requests to localhost origins only.

Accessibility

- **Responsive design:** The Next.js frontend uses Tailwind CSS responsive utilities, functioning on desktop and mobile viewports.
- **Dark/light mode:** Theme toggle reduces eye strain and accommodates user preferences.
- **Step-by-step wizard:** The multi-step evaluation flow breaks a complex task into manageable steps, reducing cognitive load.
- **Visual feedback:** Score cards, progress indicators, and color-coded signals provide clear, at-a-glance results.
- **Keyboard navigation:** Standard HTML form elements and buttons support keyboard-based interaction.

12. Ethical Considerations & Limitations

Ethical Considerations

- **Bias:** The model was trained on US marketplace data (Amazon.com) across 14 categories. Results may not generalize to international markets or categories not represented in the training data.
- **Fair use:** This is an academic research project. Commercial deployment would require compliance with Amazon's Terms of Service and disclosure of prediction uncertainty to end users.
- **Responsibility:** Predictions should inform decisions, not determine them. The model identifies correlations between listing features and BSR — it does not guarantee sales outcomes. Human judgment, market research, and business context must complement model outputs.

Limitations

1. **Model Performance Ceiling:** $R^2 = 0.425$ means 57.5% of BSR variance is explained by factors beyond listing content — pricing, promotions, competitor actions, seasonality, and ad spend.
2. **Category Specificity:** 14 categories are covered. Results may not transfer to fundamentally different verticals (books, grocery, apparel) without retraining.
3. **Temporal Validity:** Trained on late-2025 data. Amazon's algorithms, market conditions, and consumer behavior evolve. Quarterly retraining is recommended.
4. **Causation vs. Correlation:** Better images *correlate* with better BSR, but the causal direction is unclear — successful sellers may invest more in photography because they sell well, not solely the reverse.

13. Deployment & Reproducibility

System Requirements

Requirement	Minimum	Recommended
CPU	4+ cores	8+ cores
RAM	8GB	16GB
Storage	15GB free	25GB free
GPU	Not required	CUDA-capable

Installation & Running

```
# Clone repository
git clone https://github.com/DylanSuniaga/
  Forecasting-Amazon-Sales-with-Multimodal-Data.git
cd Forecasting-Amazon-Sales-with-Multimodal-Data
```

```
# Create Conda environment
conda env create -f environment.yml
conda activate amazon-forecast
```

```
# Train & export models for web
cd notebooks/02-Image\ Analysis
python save_models_for_web.py
```

```
# Start backend (terminal 1)
cd src/web/backend
uvicorn app.main:app --reload --port 8000
```

```
# Start frontend (terminal 2)
cd src/web/frontend
npm install && npm run dev
```

Access at <http://localhost:3000> | API docs at <http://localhost:8000/docs>

Tech Stack

Layer	Technology
ML Models	XGBoost, LightGBM, scikit-learn
Deep Learning	PyTorch, EfficientNet-B0 (timm), CLIP ViT-B-32 (open_clip)
Computer Vision	OpenCV, YOLOv8 (ultralytics)
NLP	textstat, scikit-learn TF-IDF + TruncatedSVD
Backend	FastAPI, Uvicorn, Pydantic
Frontend	Next.js 14, React 18, TypeScript, Tailwind CSS
Data	pandas, NumPy, Pillow
Environment	Python 3.10 (Conda), Node.js 18+

Reproducibility

- **Random Seeds:** All models use `random_state=42`
- **Data Integrity:** SHA-256 checksums in `reports/dataset_audit.md`
- **Environment Locking:** `environment.yml` pins exact package versions
- **PCA Transformers:** Saved per-category to prevent train/test leakage

14. Future Work & Recommendations

Completed from Capstone I

- Deep learning embeddings (EfficientNet-B0 + CLIP)
- Per-category model specialization (14 models)
- Full-stack web application (FastAPI + Next.js)
- NLP feature pipeline (TF-IDF + SVD + structural features)

Remaining Opportunities

1. **Time-Series BSR Tracking:** Collect historical BSR over 30-90 days; model trajectory (improving/declining/stable) using LSTM/GRU networks
2. **Multi-Marketplace Expansion:** Amazon UK, DE, JP to identify universal vs. culture-specific success factors
3. **Causal Inference Study:** Randomized controlled trial with real listings to establish whether improvements *cause* BSR improvement
4. **Browser Extension:** Chrome/Firefox extension overlaying predicted BSR and quality scores on Amazon product pages in real time
5. **Expanded Categories:** Apparel, books, grocery, home goods — each with unique feature importance patterns
6. **Transformer Text Models:** BERT/RoBERTa embeddings for titles to capture semantic meaning beyond TF-IDF

15. References

- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD*, 785-794.
- Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for CNNs. *ICML*.
- Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. *ICML*.
- Jocher, G. (2023). YOLOv8 by Ultralytics. <https://github.com/ultralytics/ultralytics>
- Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *JMLR*, 12, 2825-2830.
- Bradski, G. (2000). The OpenCV library. *Dr. Dobb's Journal*, 25(11), 120-123.
- McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*, 445, 51-56.
- Textstat Contributors. (2024). textstat: Calculate readability statistics. <https://pypi.org/project/textstat/>
- Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *NeurIPS*, 30.

Appendix A: Capstone I vs. Capstone II Changelog

Area	Capstone I	Capstone II
Dataset	~19,000 products	~36,900 products
Categories	9 electronics subcategories	14 Amazon categories
Unranked Ratio	13.7%	30.1%
Image Features	OpenCV + YOLOv8 only	+ EfficientNet-B0 + CLIP
NLP Features	Basic text stats + keywords	+ TF-IDF PCA (100 dims) + structural
Total Features	~80	~2,330
Models	1 global clf + 1 global reg	Global + 14 per-category models
Classification AUC	0.849	0.891 (global) / 0.954 (per-cat avg)
Regression R^2	0.267	0.425 (global)
CV Stability	N/A	$R^2 = 0.378 \pm 0.022$
Deployment	Notebooks only	Full-stack web application

Appendix B: Project Structure

```

project/
  CLAUDE.md, environment.yml, README.md
  data/ (not in git)
    products_with_image_feats.csv (~4.4GB)
    data_with_scraper.csv (~60MB)
    images_amz/ (~36.9K images)
  notebooks/
    00-Data Downloading/
    01-Base Model with Visuals/
    02-Image Analysis/
      save_models_for_web.py (model export)
    03-NLP/
    04-Final Modeling/
      main.ipynb, report_figures.ipynb
  src/
    nlp/preprocess.py
    web/backend/app/
      api/routes.py
      core/config.py
      services/ (inference, image_features,
                  text_features, model_registry)
      models/ (saved XGBoost/LightGBM artifacts)
    web/frontend/src/
      app/page.tsx
      components/ (20+ React components)
      lib/api.ts, types/index.ts
  presentation/

```

Product Documentation Version: 2.0 | CIS 4951 - Capstone II | Spring 2026

Dylan Suniaga, Anabella Trias, Kevin Duran, Gabriel Lopez