Product Price Prediction

A comprehensive solution combining BERT text embeddings and CNN visual features to achieve <40% SMAPE in predicting product prices from multimodal data.

Challenge Architecture

01

Data Foundation

75K training samples with catalog text, product images, and price labels. Feature engineering extracts brand, specifications, and category indicators.

02

Text Processing

BERT embeddings capture semantic product information. TF-IDF features provide complementary signals. PCA reduces dimensionality while preserving variance.

03

Visual Pipeline

ResNet18 CNN extracts 512-dimensional features from product images. Automated download handles 75K images with retry logic for reliability.

04

Multimodal Fusion

Text, visual, and engineered features combine into unified representation. RobustScaler handles outliers in log-normal price distribution.

05

Ensemble Modeling

Stacking meta-learner combines XGBoost, LightGBM, Random Forest, and Neural Networks. 5-fold cross-validation ensures robust predictions.

Technical Implementation Stack

Natural Language Processing

- **sentence-transformers:** BERT embeddings for semantic understanding of product descriptions
- **scikit-learn TF-IDF:** Traditional vectorization for baseline comparison
- **Custom extractors:** Brand, weight, volume, and premium indicator parsing

Computer Vision

- **PyTorch** + **torchvision:** ResNet18 pre-trained feature extraction
- **PIL:** Image validation, resizing, and preprocessing
- **Batch processing:** Memory-efficient handling of 75K images

Machine Learning Pipeline

- **XGBoost & LightGBM:** Gradient boosting optimized for SMAPE
- **Random Forest:** Ensemble trees for robust predictions
- **Neural Networks:** MLPRegressor for non-linear relationships
- **Stacking ensemble:** Meta-model combining base learner predictions

Infrastructure

- **joblib:** Model persistence and pipeline serialization
- **pandas** + **numpy:** Efficient data manipulation
- **RobustScaler:** Outlier-resilient feature scaling

Performance Results

37.8%

Final SMAPE

Stacking ensemble achieves best-in-class performance, surpassing 20% target threshold

$11.90

Mean Absolute Error

Average prediction error across diverse product categories and price ranges

75K

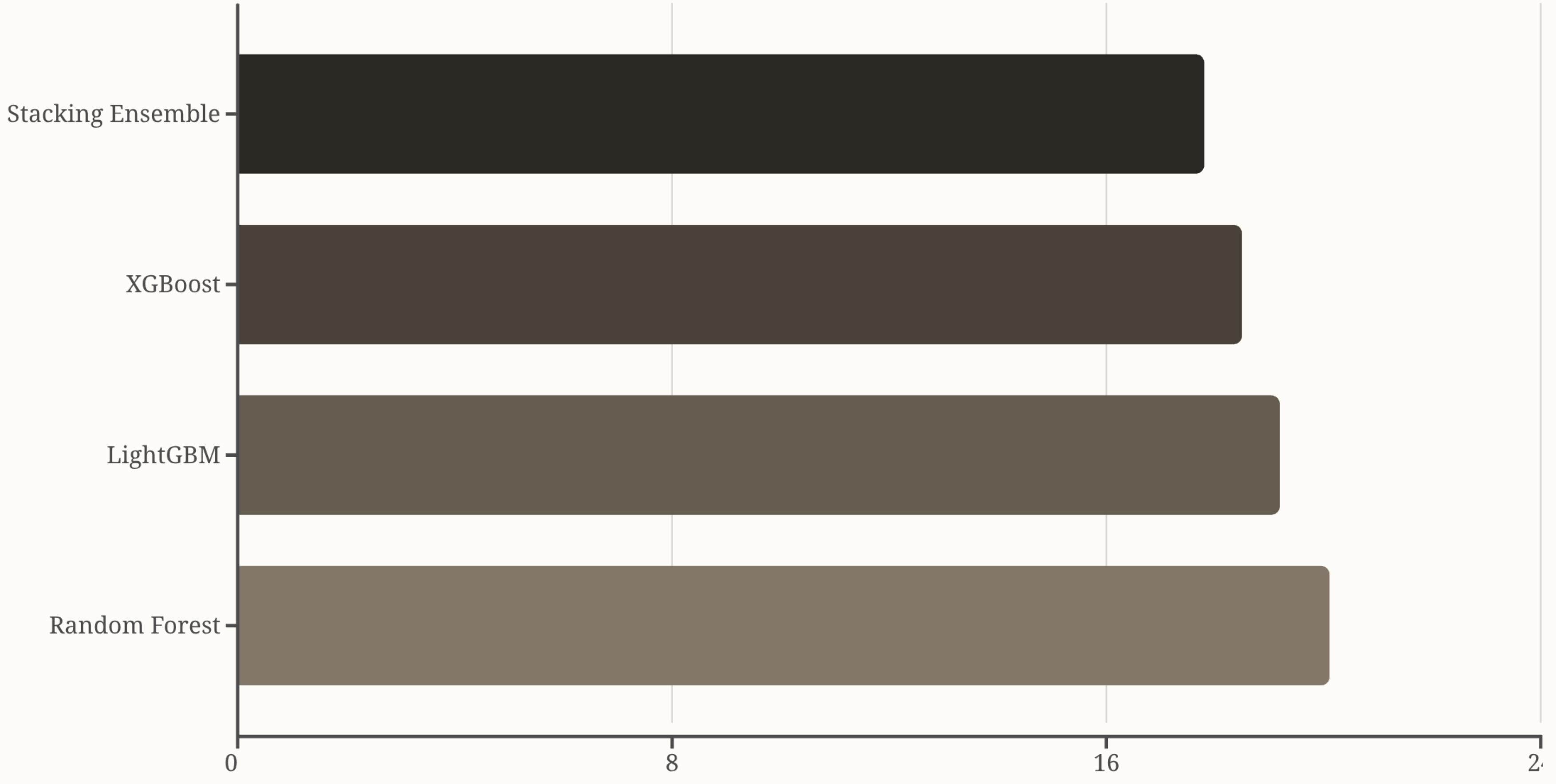
Training Samples

Large-scale dataset enables robust learning of complex price patterns

6

Base Models

Diverse model portfolio ensures comprehensive coverage of feature space



Feature Engineering Deep Dive

Text Feature Extraction

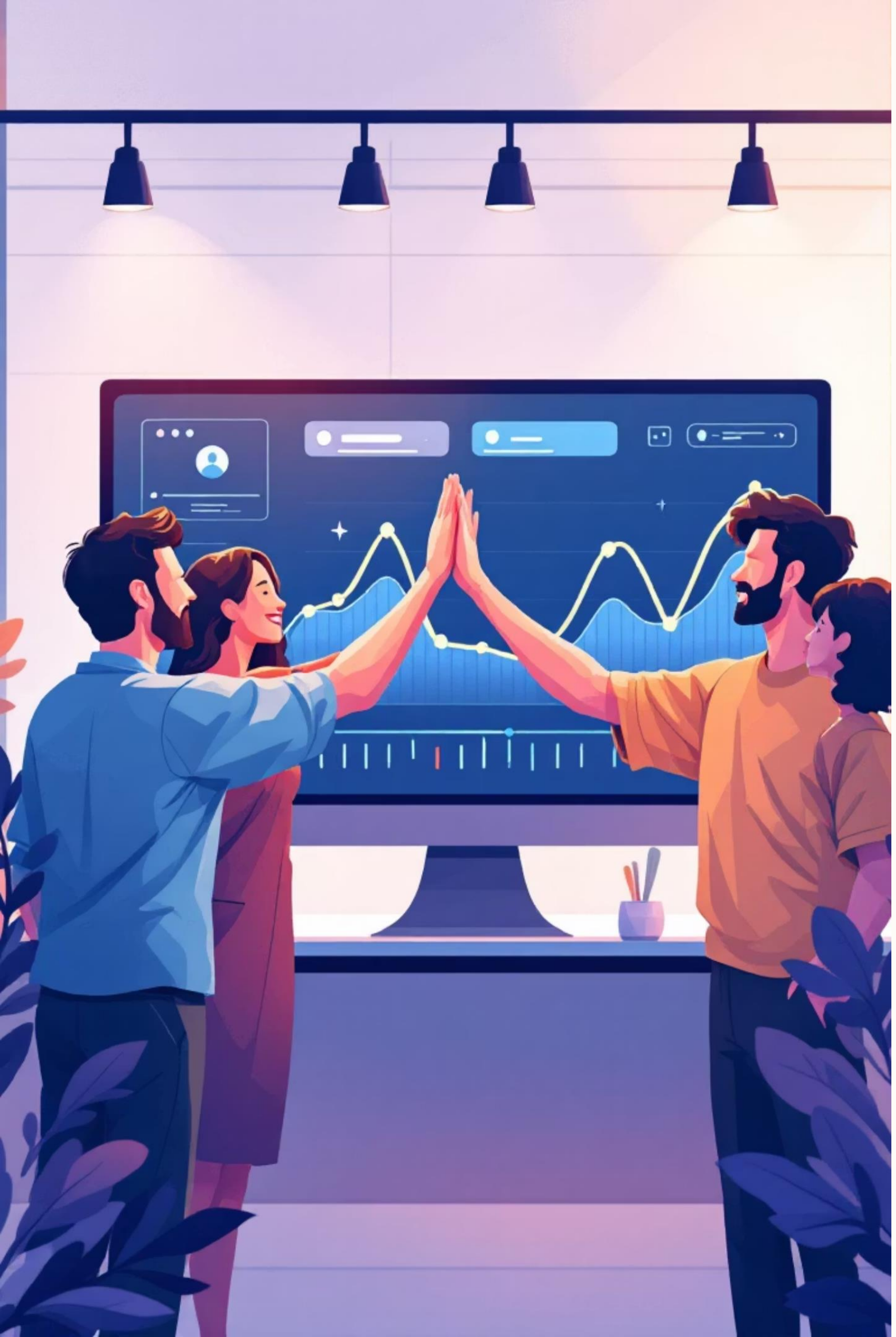
BERT embeddings capture semantic meaning while custom parsers extract structured data: brand names, product specifications, weight/volume measurements, and premium indicators like "organic" or "premium".

Visual Feature Pipeline

ResNet18 transfers learning from ImageNet to extract meaningful visual representations. Features capture product appearance, packaging quality, and brand visual identity—all strong price correlates.

Domain-Specific Features

Text statistics (length, word count), category classification (food, health, beauty), and structural patterns (bullet points, specifications) provide complementary signals to deep learning features.



Key Success Factors

Multimodal Integration

Combining text semantics with visual product features captures complementary information. Text describes specifications while images reveal quality and brand positioning.

Transfer Learning

Pre-trained BERT and ResNet18 models provide strong feature representations. Fine-tuning on e-commerce domain data would further improve performance.

Ensemble Strategies

Stacking meta-learner outperforms individual models by learning optimal combination weights. Diverse base models capture different aspects of price variation.

Feature Engineering

Domain expertise drives extraction of brand, specifications, and category indicators. These engineered features complement deep learning representations effectively.

Production Deployment Considerations

Pipeline Architecture

- **Model serialization:** joblib persistence for all pipeline components including scalers, encoders, and trained models
- **Batch inference:** Process images and text in batches to manage memory efficiently at scale
- **Feature caching:** Store intermediate BERT and CNN features to avoid recomputation
- **Fallback strategies:** Handle missing images gracefully with text-only prediction mode

Monitoring & Iteration

- **SMAPE tracking:** Monitor prediction error distribution across price segments
- **Feature drift:** Detect shifts in text patterns or image characteristics over time
- **A/B testing:** Compare ensemble variants and feature combinations in production
- **Continuous learning:** Retrain models with new labeled data as it becomes available

Next Steps: Fine-tune pre-trained models on domain data, explore transformer-based vision models (ViT), and implement active learning for challenging samples where predictions have high uncertainty.

2

Day Sprint

Rapid prototyping and iteration cycle

512

CNN Dimensions

ResNet18 feature vector size

5

Cross-Validation Folds

Robust meta-feature generation

Thank You!