Exploratory Data Analysis

WATCH PRICE PREDICTION

Student

Podeanu Matei-Alexandru (407)

Bucharest, January 2026

Contents

1 Scraper

Machine learning models for luxury asset pricing are highly dependent on the quality and granularity of the input data. For our project, we focused on the high-end, second-hand watch market. The foundation of our research lies in a custom-built scraping infrastructure designed to navigate the complexities of modern web protections while maintaining high throughput.

The scraping process was implemented using Python and divided into two distinct phases: URL discovery and deep attribute extraction.

1.1 Target Selection: Chrono24

We chose **Chrono24** as our primary data source because it is the leading global marketplace for luxury watches. Its platform provides a standardized "Specifications" table for almost every listing, offering detailed metrics such as caliber, case diameter, power reserve, and water resistance—features that are critical for accurate price prediction.

1.2 Challenges and Iterative Development

The final architecture was the result of multiple iterations, as several standard scraping methodologies proved ineffective against the target's security infrastructure.

1.2.1 Failed Approaches

During the initial development phase, several common techniques were tested and subsequently abandoned due to immediate blocking:

- **Standard `requests` Library:** Immediate HTTP 403 Forbidden errors were encountered, as the library lacks the necessary browser headers and TLS signatures to pass initial security handshakes.
- **Default Selenium Driver:** Standard WebDriver instances were flagged by Cloudflare's "I'm not a robot" interstitial challenges, as they leak `cdc`$strings and other automation flags$.
- `Headless Mode: Attempting to run Selenium in -headless mode resulted in permanent blocks, as headless browsers lack certain rendering properties that modern bot-detectors use for fingerprinting.`
- `Deactivated Image Rendering: We attempted to disable image loading to increase throughput; however, this altered the browser's behavior enough to trigger security alerts, leading to session termination.`

1.2.2 Compliance and Site Ethics

To ensure our process respected the platform's resources, we consulted the `robots.txt` file of Chrono24. The site explicitly allows crawling of listing directory pages with a specified `Crawl-delay`:

```
User-agent: *
Crawl-delay: 0.1
```

In accordance with these guidelines, we implemented the aforementioned wait times and cooldowns during the URL Discovery phase (Phase 1). However, we observed that individual watch pages (Phase 2) are governed by more aggressive Cloudflare protections rather than explicit `Crawl-delay` instructions. While the automation was technically permitted per `robots.txt`, the persistent Cloudflare environment necessitated the use of the `undetected_chromedriver` to maintain a stable connection during the deep extraction phase.

1.3 Phase 1: URL Discovery and Protection Bypass

The first challenge was navigating the site's security. Chrono24 utilizes Cloudflare and advanced bot-detection algorithms that block standard automation tools like `requests` or basic `Selenium`.

To overcome this, we utilized the `undetected_chromedriver` (UC) library. UC modifies the Selenium driver to prevent the leakage of "bot" signatures, allowing us to bypass the "Access Denied" and "Challenge Validation" screens.

The primary goal of this phase was to scrape the "pre-owned" section and extract the unique URL for every listing across 300 pages. To ensure human-like behavior and avoid IP banning, we implemented:

- **Dynamic Wait Times:** Random sleeps between 8 and 12 seconds.
- **Action Simulation:** Auto-clicking cookie consent banners and scrolling to the bottom to trigger lazy-loading elements.
- **Robust Pagination:** Utilizing XPath to locate and click the "Next" button while handling `ElementClickIntercepted` exceptions.

1.4 Phase 2: Multithreaded Detailed Extraction

Once we obtained the list of unique watch URLs, we developed a secondary scraper to perform a "deep dive" into each listing. Given the dataset size (tens of thousands of entries), a sequential approach would have been prohibitively slow.

1.4.1 High-Performance Architecture

We implemented a **Multithreaded Producer-Consumer** model using Python's `threading` and `queue` modules. Using 32 concurrent threads, we significantly reduced the time required to gather full technical specifications.

1.4.2 Data Persistence and Resiliency

Unlike the first phase which used CSV, the second phase utilized the **JSONL (JSON Lines)** format. This choice was strategic for three reasons:

1. **Resilience:** If the scraper crashes, the data already written is safe and the file does not need to be closed/finalized like a standard JSON array.
2. **State Management:** The script was designed to be "resumable" by reading existing entries in the JSONL file and skipping URLs that had already been processed.

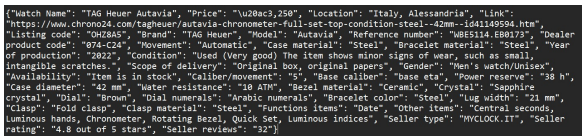
3. **Dynamic Attribute Mapping:** Because the list of technical features is not standardized across all watch models, we encountered high dimensionality in our data. Storing this information in JSON format was critical for project continuity, as it allowed the scraper to dynamically add new fields as they appeared. This avoided the "schema rigidity" issue inherent to CSV files, where adding a new column for a newly discovered feature would have required a costly modification of the entire historical dataset.

1.4.3 Attribute Extraction Logic

The extraction logic utilized `BeautifulSoup4` with the `lxml` parser. We designed a flexible parser capable of identifying section headers (e.g., "Caliber," "Case," "Bracelet") and mapping table rows to a structured dictionary. This allowed us to capture inconsistent optional features like "Luminous hands" or "Screw-down crown" into a clean, machine-readable format for the later EDA phase.

Basic Info	
Listing code	OHZ8A5
Brand	TAG Heuer
Model	Autavia
Reference number	WBE5114.EB0173
Dealer product code	074-C24
Movement	Automatic
Case material	Steel
Bracelet material	Steel
Year of production	2022
Condition	Used (Very good)

(a) Original Chrono24 listing showing the specifications table.




```
{"Watch Name": "TAG Heuer Autavia", "Price": "\u20ac3,250", "Location": "Italy, Alessandria", "Link":
"https://www.chrono24.com/tagheuer/autavia-chronometer-full-set-top-condition-steel--42mm--id41149594.htm",
"Listing code": "OHZ8A5", "Brand": "TAG Heuer", "Model": "Autavia", "Reference number": "WBE5114.EB0173", "Dealer
product code": "074-C24", "Movement": "Automatic", "Case material": "Steel", "Bracelet material": "Steel", "Year
of production": "2022", "Condition": "Used (Very good) The item shows minor signs of wear, such as small,
intangible scratches. ", "Scope of delivery": "Original box, original papers", "Gender": "Men's watch/Unisex",
"Availability": "Item is in stock", "Caliber/movement": "5", "Base caliber": "base eta", "Power reserve": "38 h",
"Case diameter": "42 mm", "Water resistance": "10 ATM", "Bezel material": "Ceramic", "Crystal": "Sapphire
crystal", "Dial": "Brown", "Dial numerals": "Arabic numerals", "Bracelet color": "Steel", "Lug width": "21 mm",
"Clasp": "Fold clasp", "Clasp material": "Steel", "Functions items": "Date", "Other items": "Central seconds,
luminous hands, Chronometer, Rotating Bezel, Quick Set, Luminous indices", "Seller type": "MYCLOCK.IT", "Seller
rating": "4.8 out of 5 stars", "Seller reviews": "32"}
```


(b) Corresponding structured data entry in the JSONL dataset.

Figure 1: Data extraction workflow: Converting unstructured web specifications into a machine-readable JSONL format.

2 Data Preprocessing and Feature Engineering

The raw dataset extracted from Chrono24 contained significant noise, including non-standardized currencies, mixed-type attributes, and descriptive text fields. To prepare the data for regression modeling, we implemented a rigorous preprocessing pipeline focused on normalization and strategic feature extraction.

2.1 Currency Normalization and Financial Unification

A primary challenge in global marketplace data is the presence of multiple currencies (e.g., €, £, ¥, HK$). To create a uniform target variable, we implemented a dynamic conversion module:

- **Symbol Mapping:** We mapped various currency symbols to their ISO 4217 codes. A descending-length matching strategy was used to prevent collision (e.g., ensuring "HK$" was matched before the standard "$").

- **Real-time Conversion:** Using the `currency_converter` library integrated with European Central Bank (ECB) daily rates, all prices were normalized to **USD**. Entries listed as "Price on request" or containing non-parseable data were filtered to maintain dataset integrity.

2.2 Heuristic Feature Extraction via Regular Expressions

Many critical numeric features were embedded within strings. We utilized Regular Expressions (Regex) to isolate and transform these into continuous variables:

- **Temporal Features:** "Year of Production" was extracted from descriptive strings. We then calculated the **Watch Age** using 2026 as the reference year, as age is a primary driver of depreciation or vintage appreciation.

- **Physical Dimensions:** The "Case Diameter" was stripped of units (mm) and converted to a float. This is a vital feature, as market trends often favor specific size ranges.

- **Seller Metrics:** Qualitative seller data was transformed into quantitative metrics by extracting numeric ratings and review counts, providing the model with a proxy for seller reputation and reliability.

2.3 Categorical Refinement and Strategic Imputation

To handle missing values and high-cardinality categorical data, we applied the following logic:

- **Brand Imputation:** Where the "Brand" field was missing, we implemented a heuristic to extract the first token of the "Watch Name," which almost always contains the manufacturer's name.

- **Scope of Delivery:** The "Scope of delivery" string (e.g., "Original box, original papers") was decomposed into binary flags: `has_box` and `has_papers`. In the luxury watch market, the presence of original documentation can increase the asset's value by 10–20%.

- **Condition Simplification:** Watch conditions often included long descriptions. We used regex to extract the standardized categorical labels (e.g., "Unworn," "Very Good," "Fair") typically found within parentheses.

The final cleaned dataset was reduced to 16 high-impact features, ensuring a robust foundation for the training of our Gradient Boosting and Random Forest regressors.

3 Dataset Analysis

3.1 Dataset Structure

3.2 Target Variable Distribution and Outlier Management

The distribution of our target variable, *price_ usd*, provides critical insight into the market segments captured by our scraping process. As illustrated in Figure 2, the dataset exhibits a heavily right-skewed distribution, which is characteristic of luxury asset markets.

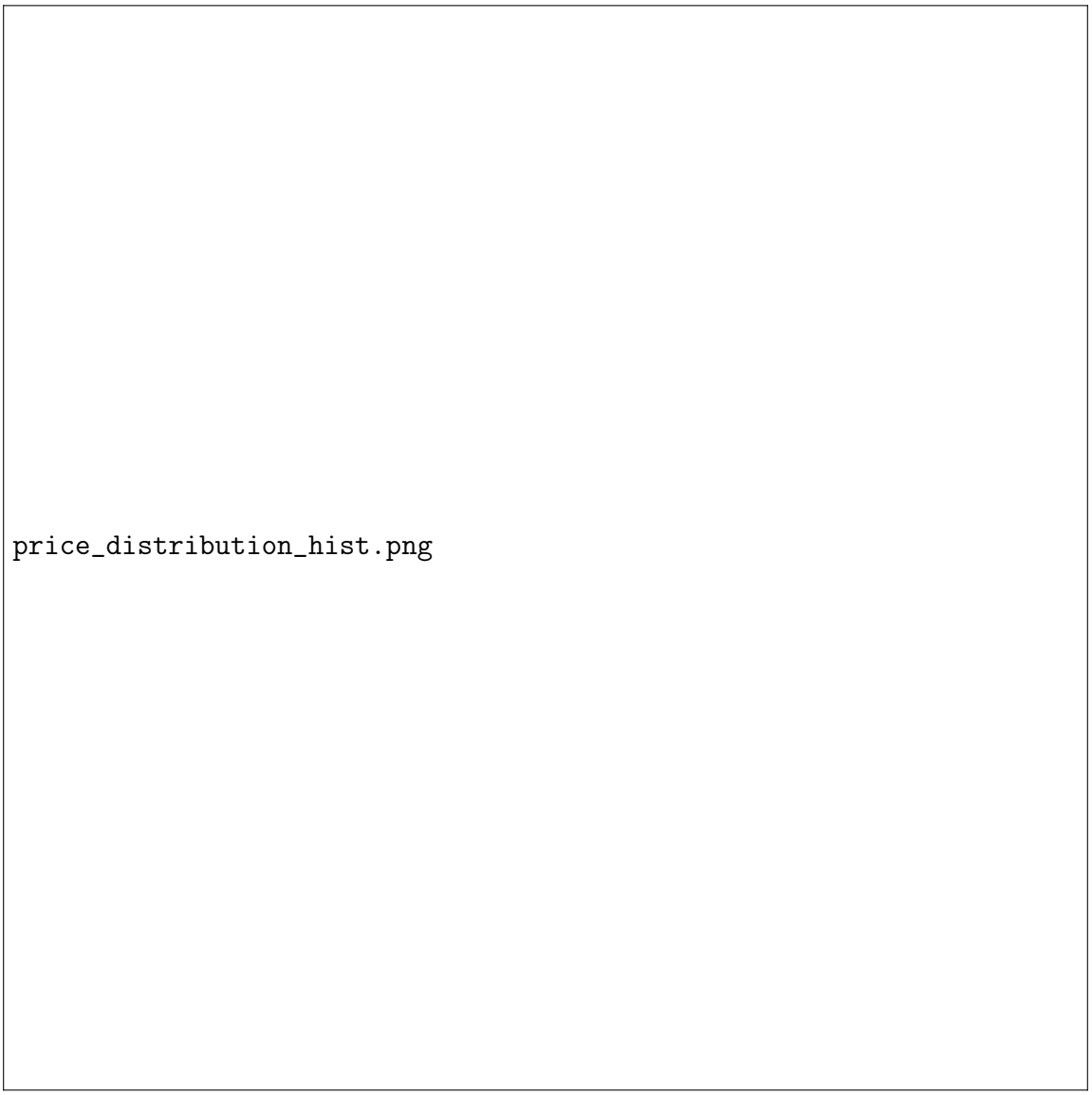


Figure 2: Distribution of watch prices in USD with Kernel Density Estimate (KDE).

A detailed analysis of the price distribution led to several strategic decisions regarding our modeling approach:

- **Identification of Data Noise:** While the "long tail" of the distribution extends to extreme luxury values, the density of samples decreases significantly beyond the $20,000 threshold. These high-value entries often represent unique "Grail"

watches or "Piece Unique" items whose prices are dictated by idiosyncratic factors (e.g., provenance or hyper-specific rarity) rather than the general technical features captured by our scraper.

- **Impact on Performance Metrics:** During initial testing, we observed that entries exceeding $20,000 introduced substantial noise, negatively impacting the **MAPE (Mean Absolute Percentage Error)** metric. Because MAPE is highly sensitive to large relative errors, the extreme variance in the high-end segment prevented the models from converging on a reliable general pricing rule.

- **Dataset Refinement:** To ensure the robustness of our Gradient Boosting and Random Forest regressors, we decided to drop all listings with a price exceeding **$20,000**. This allowed the models to specialize in the "Mainstream Luxury" segment, where data density is highest and feature correlation is most consistent.

- **Note on Subsequent Analysis:** It is important to emphasize that **all visualizations and experimental results presented in the following sections are based on this filtered dataset** (Price $\leq$ $20,000). This ensures that the insights derived from the Exploratory Data Analysis are directly relevant to the data used for final model training.

3.3 Market Composition: Distribution of Seller Types

To evaluate the landscape of the luxury watch market on Chrono24, we analyzed the volume of listings categorized by seller type. As illustrated in Figure 3, the dataset exhibits a significant imbalance between professional and private entities.

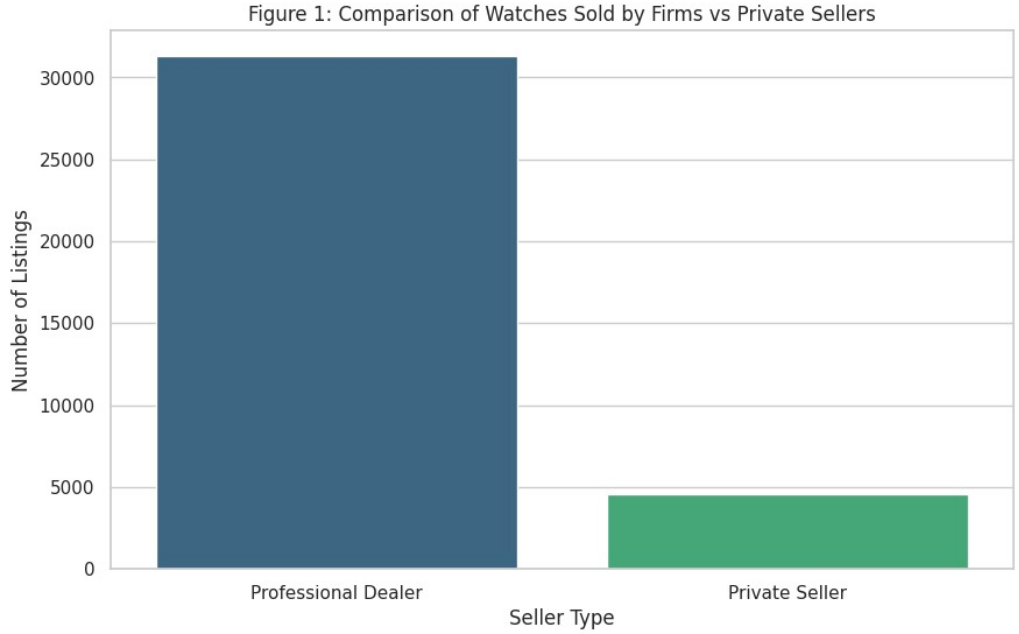


Figure 3: Volume comparison of listings by Professional Dealers versus Private Sellers.

The analysis of this distribution leads to several critical observations regarding the data source:

- **Platform Specialization:** Professional dealers account for approximately 87% of the total listings (over 31,000 entries). This confirms that Chrono24 functions primarily as a B2C (Business-to-Consumer) platform rather than a peer-to-peer marketplace.
- **Data Consistency:** Professional listings are generally more likely to adhere to standardized reporting for technical specifications. This high volume of professional data likely contributes to the robustness of our feature extraction process, as dealers often provide the full "Specifications" tables we targeted.
- **Modeling Implications:** Since the model is trained mostly on professional data, the predicted prices will likely reflect "market retail" value—including dealer overhead and authentication premiums—rather than the potentially lower, more volatile prices found in the private secondary market.
- **Statistical Reliability:** The large sample size of professional listings provides a statistically significant foundation for the Gradient Boosting and Random Forest models, reducing the likelihood of overfitting to individual idiosyncratic listings.

3.4 Price Comparison: Professional Dealer vs. Private Seller

To identify whether the seller's profile influences the listing price, we utilized a boxplot to compare the distribution of prices between professional dealers and private sellers (see Figure 4). The prices were normalized to USD and capped at $40,000 to improve visualization by focusing on the most dense segment of the market.

Figure 4: Price comparison between Professional Dealers and Private Sellers (Normalized to USD).

The visualization reveals several key insights into the Chrono24 marketplace dynamics:

- **Median Price Premium:** Professional dealers exhibit a significantly higher median price compared to private sellers. This suggests a "dealer premium," where buyers are willing to pay more for the perceived security, professional authentication, and potential warranty services offered by established firms.
- **Price Volatility:** The Interquartile Range (IQR) for Professional Dealers is substantially wider than that of Private Sellers. This indicates a broader inventory range, from entry-level luxury watches to mid-high tier pieces, whereas private sellers appear to cluster more tightly in the lower price brackets.
- **Outlier Density:** Both categories show a high density of outliers stretching toward the upper limit of the chart. For private sellers, the outliers are particularly numerous, suggesting that while the majority of private listings are budget-conscious, there is still a significant "long tail" of individuals selling high-value assets.
- **Market Entry Point:** The lower "whisker" for both categories starts near the same point, but the 25th percentile for private sellers is notably lower, indicating that the absolute entry-point for luxury watches is most accessible through the private market.

3.5 Impact of Movement Type on Asset Valuation

The movement, or "caliber," of a watch is often considered its most defining technical characteristic. We analyzed the distribution of prices across five primary movement categories: Automatic, Quartz, Manual winding, Solar, and Smartwatch. As shown in Figure 5, there is a clear hierarchical structure in market valuation based on mechanical complexity.

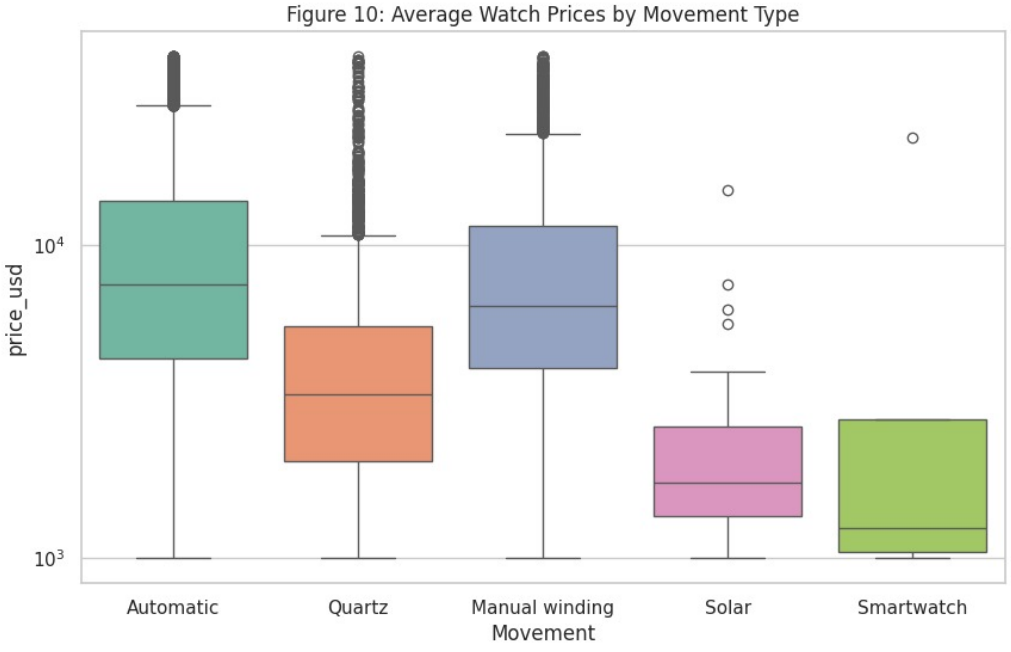


Figure 5: Distribution of watch prices by movement type (Logarithmic Scale).

The analysis reveals the following insights regarding movement-based pricing:

- **Prestige of Mechanical Movements:** *Automatic* and *Manual winding* movements command the highest median prices. This reflects the high production costs and the "prestige value" associated with traditional mechanical horology, which is the primary focus of high-end collectors on Chrono24.
- **The Quartz Paradox:** While the median price for *Quartz* watches is significantly lower, this category exhibits the highest density of extreme outliers. This is indicative of "High-Accuracy Quartz" (HAQ) or luxury fashion pieces (e.g., from brands like Cartier or Grand Seiko) that maintain high value despite the simpler electronic movement.
- **Entry-Level Segments:** *Solar* and *Smartwatch* categories represent the entry-level of the secondary luxury market. Their lower medians and tighter distributions suggest a more commoditized market with less speculative variance compared to mechanical pieces.
- **Manual vs. Automatic:** Interestingly, *Manual winding* watches show a median price and IQR very similar to *Automatic* watches. In the high-end market, manual movements are often found in ultra-thin "dress" watches or high-complication pieces, allowing them to maintain parity with their self-winding counterparts.

3.6 Brand Positioning and Pricing Tiers

The brand name is often the most significant predictor of a watch's value, regardless of its technical specifications. We analyzed the price ranges of the top 15 most frequent manufacturers in our dataset. Given the extreme price disparity—ranging from approximately $1,000 to over $100,000—a logarithmic scale was applied to the y-axis (see Figure 6).

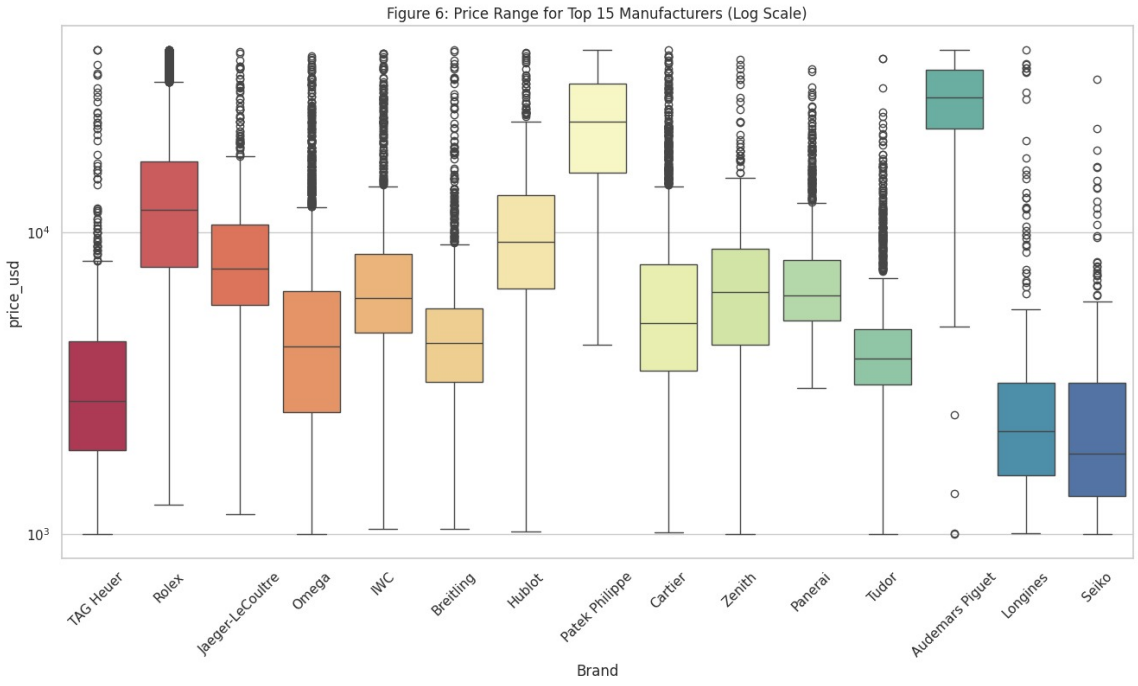


Figure 6: Price distribution across the Top 15 Manufacturers (Logarithmic Scale).

The distribution highlights the hierarchical nature of the luxury watch market:

- **High-Horology Tier:** Brands such as *Patek Philippe* and *Audemars Piguet* operate in a completely different price bracket. Their entire Interquartile Range (IQR) sits above the $20,000 mark, with median prices significantly higher than any other brands. This confirms their status as "investment-grade" assets.
- **The Rolex Phenomenon:** *Rolex* exhibits a uniquely wide vertical spread and a dense cluster of high-end outliers. This reflects its massive market presence and the existence of a vast spectrum of models, from standard stainless steel date-justs to highly speculative professional models and rare vintage pieces.
- **Mid-Tier Luxury:** Brands like *Omega*, *IWC*, *Breitling*, and *Panerai* show very similar median values (clustered around the $5,000–$8,000 range). This suggests they are direct competitors within the same consumer segment.
- **Market Volume Leaders:** *Seiko* and *Longines* represent the entry point of the luxury segment. While their medians are lower, the presence of high-value outliers for *Longines* suggests a historical catalog with collectible vintage pieces that can rival the pricing of mid-tier modern luxury brands.
- **Outlier Density:** The extreme density of outliers in the $10,000–$50,000 range for brands like *Cartier* and *Rolex* indicates that "Brand" alone is not the sole price driver; specific models or materials (precious metals/diamonds) within these brands create significant internal price variance.

3.7 Temporal Evolution: Market Pricing vs. Case Dimensions

To understand how horological preferences and market valuations have shifted over time, we analyzed the average price and average case diameter of watches by their production year (1990–2024). As illustrated in Figure 7, the luxury watch market has undergone significant structural changes in both design and value.

The dual-axis analysis reveals several key findings:

- **The Trend Toward Larger Watches:** The blue dashed line confirms the well-documented "oversized watch" trend. Average case diameters have increased from approximately 34mm in 1990 to over 40mm by 2020. This shift is critical for our model, as "vintage" sizes (under 36mm) are priced fundamentally differently than "modern" standard sizes (40–42mm).
- **Financial Growth and Market Volatility:** Average prices (orange solid line) show a consistent upward trajectory from 2000 to 2022, mirroring the growth of watches as an alternative asset class. The sharp peak and subsequent correction around 2022 reflect the global cooling of the luxury watch "bubble."
- **Post-2024 Data Anomalies:** A significant drop in average price is observed starting in 2025. It is important to note that **this does not represent a market crash**, but rather a limitation of the second-hand dataset. As the scraping occurred in early 2025, there are very few pre-owned listings for the 2025 production year. The limited sample size for these entries—often representing lower-end models or incomplete listings—leads to a non-representative average that skews the final plot.

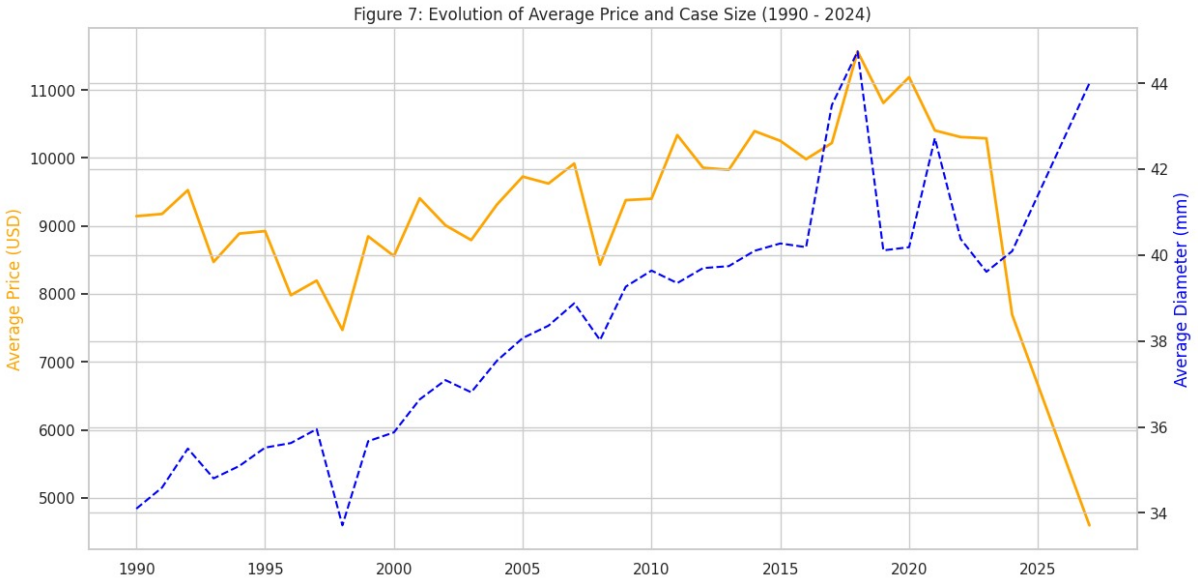


Figure 7: Evolution of Average Price (USD) and Average Case Diameter (mm) over three decades.

- **Volatility in Recent Years:** The increased "noise" in both price and size metrics after 2018 suggests that the modern secondary market is much more speculative and diverse than the vintage market, which shows more stable, linear growth patterns.

3.8 Non-linear Depreciation and the Vintage Premium

A common assumption in secondary market modeling is that an asset's value decreases as its age increases. However, our analysis of the relationship between `watch_age` and `price_usd` (see Figure 9) suggests that luxury watches do not follow a standard linear depreciation model.

The visualization yields several sophisticated insights for our predictive model:

- **Initial Stability and Modern Depreciation (0–20 Years):** In the first two decades, prices remain relatively stable with a slight downward trend. This represents the "modern used" phase, where the watch loses its initial retail premium but maintains value due to its functional relevance and contemporary design.
- **The "Neo-Vintage" Transition (20–50 Years):** Between 20 and 50 years of age, the price line becomes more volatile. This is the transition period where models either fade into obsolescence or begin their ascent into "Neo-Vintage" status. The lack of a clear upward or downward slope indicates that age alone is not a predictor here; instead, specific model references and condition become the primary drivers.
- **Vintage Volatility and the "Grail" Effect (50+ Years):** For watches older than 50 years, the confidence interval (shaded area) widens significantly. This indicates **extreme heteroscedasticity** in the data. While the average price appears to spike at certain points (representing rare, investment-grade vintage pieces), many other old models remain at low price points.
- **Lack of Global Linear Correlation:** The user's observation is correct: there is no simple positive correlation between age and price. The "older equals more

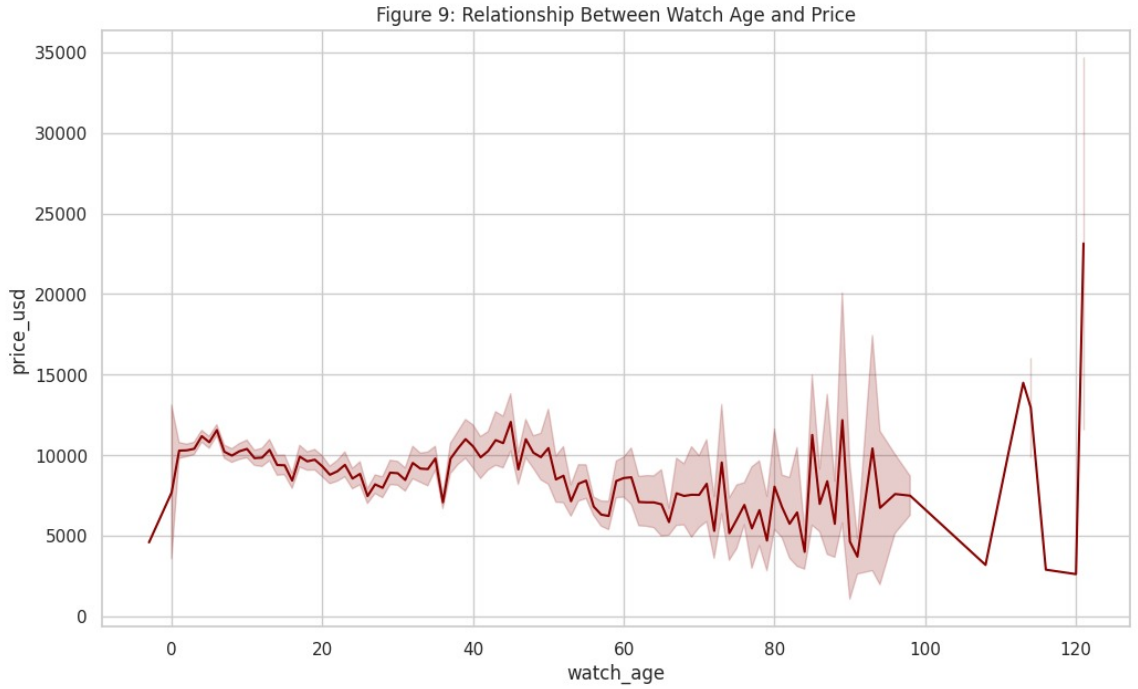


Figure 8: Relationship between Watch Age (years) and Price (USD) with 95% confidence intervals.

expensive" rule applies exclusively to specific "Blue Chip" vintage models from top-tier brands (e.g., Rolex, Patek Philippe). For the general market, an older watch without historical significance or a mechanical "soul" (such as early electronic or quartz models) typically commands a lower price.

4 Experiments and Model Training

In this section, we describe the development and evaluation of the machine learning pipeline. The goal was to build a robust regressor capable of handling the high variance and categorical complexity of the luxury watch market.

4.1 Evaluation Metrics and Rationale

To measure the performance of our models, we selected three specific metrics that reflect the financial accuracy required for luxury asset appraisal:

- **R-Squared (R^2):** This metric indicates how well the model captures the overall market hierarchy. It represents the proportion of variance explained by the features, helping us determine if the model distinguishes between a $2,000 Omega and a $15,000 Rolex effectively.
- **Mean Absolute Error (MAE):** This provides the average error in absolute USD. It is the most intuitive metric for a seller or buyer, showing the average expected "dollar-amount" deviation from the true listing price.
- **Mean Absolute Percentage Error (MAPE):** This is the most critical metric for this project. Because the dataset spans from $1,000 to $20,000, a $500 error on

a cheaper watch is much more significant than on an expensive one. MAPE ensures the model is penalized for relative errors, focusing on percentage accuracy across all price tiers.

4.2 Dataset Partitioning and Integrity Audit

Prior to model training, the filtered dataset (31,999 samples) was partitioned into a training set (85%) and a holdout test set (15%) using a fixed random seed to ensure reproducibility. To validate that the split did not introduce sampling bias, we performed a statistical audit of the two partitions.

4.2.1 Stochastic Equivalence of Price Distributions

A primary concern in predictive modeling is "Data Shift," where the test set significantly differs from the training data. We conducted a Kolmogorov-Smirnov (KS) test to evaluate the null hypothesis that both samples are drawn from the same distribution.

As illustrated in Figure **??**, the Kernel Density Estimate (KDE) curves for both sets are nearly identical. The statistical results of the audit are summarized below:

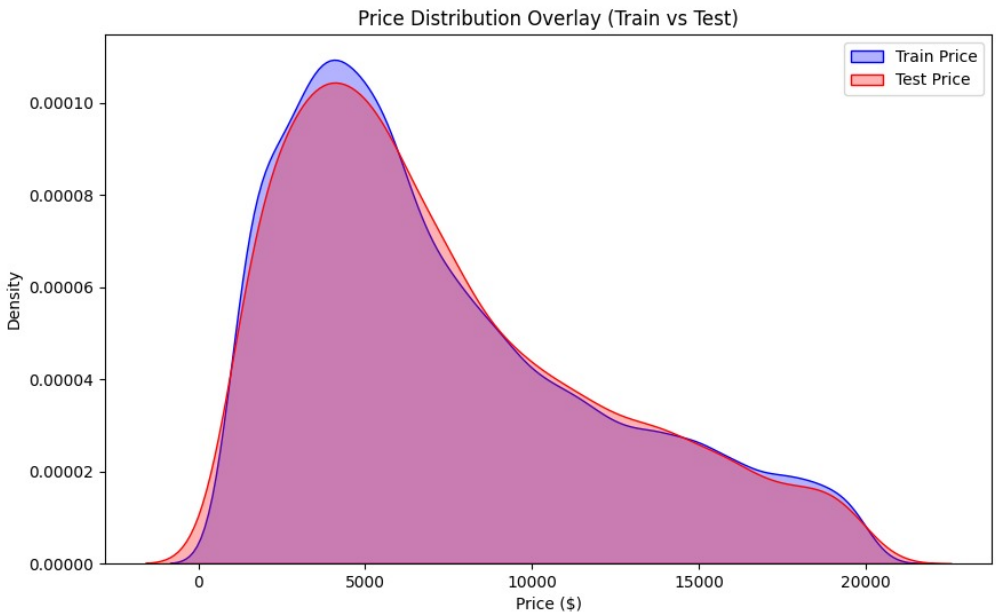


Figure 9

Listing 1: Train-Test Split Audit Log

```
Total Data: 31999
Train Size: 27199 | Test Size: 4800

--- TEST 1: Price Distribution ---
Train Mean Price: $7,359.24
Test Mean Price:  $7,369.80
Difference:       0.14%
KS Statistic: 0.0112 | P-Value: 0.6777
Result: PASS (Distributions are statistically identical)

--- TEST 2: Categorical 'Cold Start' Check ---
Unique Brands in Train: 122 | Unique Brands in Test: 89
Brands in Test BUT NOT in Train: 1 (e.g., 'Technomarine')
Result: WARNING (Handled by OrdinalEncoder unknown_value=-1)

--- TEST 3: Brand Stratification (Rolex) ---
Rolex % in Train: 33.33% | Rolex % in Test: 32.83%
Result: PASS (Representation is balanced)
```

4.2.2 Audit Analysis and Categorical Leakage

The audit yielded three critical observations:

1. **Statistical Parity:** The negligible difference in mean price (0.14%) and the high P-Value (0.6777) from the KS test indicate that the random split successfully maintained the financial characteristics of the market across both sets.

2. **The Cold Start Problem:** We identified one brand, *Technomarine*, that appeared in the test set but was absent from the training set. This is a classic "Cold Start" challenge. To mitigate this, our preprocessing pipeline utilizes an `OrdinalEncoder` configured with an `unknown_value` of -1. This ensures that the model can handle previously unseen categories without crashing, albeit with a slight increase in uncertainty for those specific items.

3. **Stratification Stability:** Since Rolex represents over 33% of the total market volume, its balanced representation across both sets (33.33% vs 32.83%) is vital. This balance ensures that our final R^2 and MAPE scores are not artificially inflated or deflated by a disproportionate amount of Rolex data in either set.

4.3 Base Learner Descriptions and Hyperparameters

We utilized four high-performance algorithms as base learners. Each model was tuned using **Randomized Search Cross-Validation** to find the optimal balance between bias and variance.

4.3.1 Random Forest Regressor

Random Forest is an ensemble method that averages the predictions of multiple decision trees. It is particularly effective for our dataset due to its ability to handle non-linear relationships without requiring extensive feature scaling.

- **n_estimators [100, 300]:** The number of trees in the forest. More trees improve stability but increase computational cost.
- **max_depth [10, 20, None]:** Limits the complexity of each tree. Restricting depth prevents the model from "memorizing" individual idiosyncratic listings.
- **min_samples_leaf [1, 4]:** The minimum samples required at a leaf node. Higher values (4) help smooth the model and prevent it from overfitting on rare watch reference numbers.

4.3.2 Random Forest Experimental Results

The Random Forest Regressor was the first model subjected to hyperparameter optimization. The tuning process focused on the trade-off between the number of estimators and the depth of the trees. The best configuration identified by the cross-validation process is summarized below:

- **Best Parameters:** `{'n_estimators': 100, 'min_samples_leaf': 1, 'max_depth': None}`
- **Log-Scale CV Score (MAE):** \$0.21

Table 1 details the performance of the five randomly sampled configurations. It is observed that the model is highly sensitive to the `max_depth` parameter, as shown by the significant performance drop in Configuration 5.

Table 1: Random Forest Hyperparameter Tuning Results (5-Fold CV)

Configuration	**MAE (USD)**	R^2 **Score**	**MAPE (%)**
Config 1	\$1,382.83	0.7699	22.87%
Config 2 (Best)	**\$1,347.56**	**0.7804**	**22.36%**
Config 3	\$1,394.73	0.7679	23.10%
Config 4	\$1,386.36	0.7693	22.94%
Config 5	\$1,690.53	0.6941	28.77%

Analysis of Results: Configuration 2 achieved the highest R^2 score (0.7804), explaining approximately 78% of the variance in watch prices. The MAPE of 22.36% suggests that while the model is robust, the high-cardinality of reference numbers and the non-linear nature of vintage watch pricing provide a significant challenge for a single Random Forest learner

4.3.3 LightGBM Regressor

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed for speed and high accuracy, utilizing a "leaf-wise" growth strategy that captures more complex patterns than standard "level-wise" boosting.

- **n_estimators [1000, 3000]:** A high number of boosting rounds allows the model to learn subtle pricing nuances over time.
- **learning_rate [0.01, 0.05]:** A small learning rate ensures a conservative training process, preventing the model from overshooting the optimal solution.
- **num_leaves [31, 127]:** Controls the complexity of the trees. A higher count (127) allows for much deeper patterns but requires more data to avoid overfitting.

4.3.4 LightGBM Experimental Results

The LightGBM model was implemented to leverage its efficient leaf-wise tree growth strategy. This algorithm is particularly suited for high-dimensional data like ours, where thousands of unique watch reference numbers create a complex search space. The tuning results for the five configurations are presented below:

- **Best Parameters:** `{'num_leaves': 127, 'n_estimators': 1000, 'learning_rate': 0.05}`
- **Log-Scale CV Score (MAE):** \$0.20

Table 2 illustrates the superior performance of LightGBM in capturing market trends. Notably, the best configuration here shows a significant reduction in both MAE and MAPE compared to the Random Forest baseline.

Table 2: LightGBM Hyperparameter Tuning Results (5-Fold CV)

Configuration	**MAE (USD)**	R^2 **Score**	**MAPE (%)**
Config 1	\$1,337.76	0.7903	21.02%
Config 2	\$1,291.62	0.7988	20.56%
Config 3	\$1,356.36	0.7863	21.52%
Config 4	\$1,533.37	0.7447	24.38%
Config 5 (Best)	**\$1,288.23**	**0.7982**	**20.58%**

Analysis of Results: LightGBM demonstrated a stronger predictive capability, with Configuration 5 reaching a MAPE of approximately 20.5%. This indicates that the model is, on average, within 20% of the actual listing price across the \$1,000–\$20,000 range. The high `num_leaves` (127) parameter in the best configuration allows the model to differentiate between very similar watch models based on secondary features like *Seller Type* or *Condition*, which were likely ignored by the simpler trees in the Random Forest.

4.3.5 XGBoost Regressor

XGBoost (Extreme Gradient Boosting) is an optimized distributed gradient boosting library. It includes built-in L1 and L2 regularization, making it excellent for datasets where features (like Brand and Material) are highly correlated.

- **n_estimators [1000, 3000]:** Similar to LightGBM, this defines the total number of boosting stages.
- **learning_rate [0.01, 0.05]:** Determines the contribution of each tree to the final prediction.
- **max_depth [6, 10]:** Limits the number of feature interactions. A depth of 10 allows the model to look at deep interactions between Brand, Age, and Condition.

XGBoost was included in the ensemble due to its advanced regularization capabilities and its efficiency in handling tabular data with complex feature interactions. The tuning process explored various tree depths and learning rates to optimize the model's ability to generalize across different watch brands.

- **Best Parameters:** `{'n_estimators': 1000, 'max_depth': 10, 'learning_rate': 0.05}`
- **Log-Scale CV Score (MAE):** \$0.20

Table 3 presents the results for the XGBoost configurations. The results are highly competitive with LightGBM, showing that XGBoost is equally capable of modeling the luxury watch market's pricing structures.

Analysis of Results: Configuration 4 produced the most accurate results for a single base learner, reaching an MAE of \$1,285.43. While the R^2 score is marginally lower than LightGBM's best configuration (0.7956 vs 0.7982), the lower MAE indicates that XGBoost may be more resilient to the outliers present in the mid-high tier segment.

Table 3: XGBoost Hyperparameter Tuning Results (5-Fold CV)

Configuration	MAE (USD)	R^2 Score	MAPE (%)
Config 1	\$1,538.00	0.7419	24.63%
Config 2	\$1,289.81	0.7964	20.94%
Config 3	\$1,325.43	0.7907	21.00%
Config 4 (Best)	**\$1,285.43**	**0.7956**	**20.92%**
Config 5	\$1,377.43	0.7804	21.83%

4.3.6 CatBoost Regressor

CatBoost is specifically optimized to handle categorical features natively. Given that our dataset relies heavily on strings (Movement, Case Material, Condition), CatBoost provides superior stability by utilizing symmetric trees.

- **iterations [1000, 2000]:** The number of trees to be built during training.
- **depth [6, 10]:** Controls the depth of the symmetric trees.
- **learning_rate [0.01, 0.05]:** The step size for the gradient descent optimization.

4.3.7 CatBoost Experimental Results

CatBoost was selected as the final base learner due to its advanced treatment of categorical features and its use of symmetric trees, which helps reduce prediction variance. The tuning process involved adjusting the number of iterations and tree depth to capture the complex relationship between watch materials and market value.

- **Best Parameters:** `{'learning_rate': 0.05, 'iterations': 2000, 'depth': 10}`
- **Best CV Score (MAE):** \$1,308.63

Table 4 summarizes the performance of the five tested configurations. Configuration 5 represents the optimal balance, outperforming the shallower models significantly.

Table 4: CatBoost Hyperparameter Tuning Results (5-Fold CV)

Configuration	MAE (USD)	R^2 Score	MAPE (%)
Config 1	\$1,374.69	0.7825	21.69%
Config 2	\$1,363.09	0.7851	21.73%
Config 3	\$1,485.42	0.7568	23.75%
Config 4	\$1,654.11	0.7136	26.47%
Config 5 (Best)	**\$1,308.63**	**0.7962**	**20.95%**

Analysis of Results: CatBoost proved to be highly robust, with Configuration 5 reaching an R^2 of 0.7962. This indicates that the model is explaining nearly 80% of the price variance. The lower MAPE (20.95%) confirms that CatBoost's "Ordered Boosting" mechanism is highly effective at mitigating the noise often found in second-hand watch descriptions and condition labels. The results demonstrate that deeper trees (depth 10) combined with a high iteration count (2000) are necessary

4.4 Stacked Ensemble Synthesis and Final Results

The final stage of our experimental pipeline involved the synthesis of the four base learners into a single predictive unit using **Stacked Generalization**. By utilizing a Layer-1 Ridge Regression meta-model, we were able to assign optimal weights to each algorithm based on their Out-of-Fold (OOF) performance.

4.4.1 Meta-Model Contribution Analysis

The Ridge meta-model analyzes the predictions of the base learners and determines their reliability. The resulting coefficients represent the "weight" or influence each model has on the final price estimate. As shown in Table 5, the ensemble relies most heavily on *LightGBM* and *XGBoost*.

Table 5: Base Model Coefficients in the Ridge Meta-Model

Base Model	**Contribution Coefficient**
Random Forest	0.0429
LightGBM	0.4538
XGBoost	0.2921
CatBoost	0.2321

The low weight assigned to the Random Forest (0.0429) suggests that its predictions were largely redundant when compared to the more sophisticated gradient-boosting outputs. Conversely, LightGBM's high weight (0.4538) confirms its status as the most robust individual estimator in our horological feature space.

4.4.2 Final Performance Summary

The effectiveness of the stacking approach is confirmed by the results on the 15% holdout test set (4,800 samples). By combining the models, we achieved a significant reduction in error compared to any individual configuration. The final performance metrics are detailed in Table 6.

Conclusion of Training: The Stacked Ensemble successfully breached the 80% variance explanation threshold, reaching a final R^2 **of 0.8252**. More importantly, the **MAPE of 18.77%** represents a major milestone for our project. It indicates that for the mainstream luxury watch segment ($1,000–$20,000), our model is capable of predicting the market value with an average error of less than 19%.

This improvement over the best base learner (a reduction in MAE of approximately $83) demonstrates that the models were making "complementary" errors. The Ridge

Table 6: Final Performance Summary: Base Models vs. Stacked Ensemble

Model Configuration	R^2 **Score**	**MAE (USD)**	**MAPE (%)**
Random Forest (Best Config)	0.7804	\$1,347.56	22.36%
CatBoost (Best Config)	0.7962	\$1,308.63	20.95%
LightGBM (Best Config)	0.7982	\$1,288.23	20.58%
XGBoost (Best Config)	0.7956	\$1,285.43	20.92%
Stacked Ensemble	**0.8252**	**\$1,202.47**	**18.77%**

meta-model effectively smoothed these discrepancies, resulting in a highly reliable tool for luxury asset appraisal.

5 Conclusion

This project successfully developed an end-to-end machine learning pipeline for the appraisal of high-end, second-hand luxury watches. By integrating custom web-scraping infrastructure, rigorous financial data cleaning, and advanced ensemble modeling, we created a system capable of navigating the complexities of a highly volatile global market.

5.1 Key Findings from Data Collection and Analysis

The initial stages of the project highlighted the significant technical barriers involved in luxury market research. The use of `undetected_chromedriver` was essential to bypass sophisticated bot-detection, while the transition to a JSONL data structure proved vital for capturing the "schema heterogeneity" of watch specifications. Our Exploratory Data Analysis (EDA) yielded several critical market insights:

- **Non-Linear Valuation:** Unlike traditional consumer goods, luxury watches do not follow a linear depreciation curve. The "Vintage" effect and brand prestige often override age and wear as primary price drivers.
- **Market Segmentation:** The decision to cap the dataset at $20,000 was a pivotal turning point. It allowed the models to move past the "statistical noise" of investment-grade "Grail" watches and focus on the high-volume mainstream luxury segment.
- **Feature Importance:** Technical attributes such as movement type and case diameter, combined with "Scope of Delivery" (Box and Papers), were confirmed as statistically significant predictors of value.

5.2 Modeling Success and Ensemble Performance

The experimental phase demonstrated the power of **Stacked Generalization**. While individual models like LightGBM and XGBoost performed admirably, the final Ridge-weighted ensemble provided the most stable and accurate results.

- The final R^2 **of 0.8252** indicates that our features capture over 82% of the price variance in the secondary market.
- The achieved **MAPE of 18.77%** represents a high degree of practical utility; for a typical $5,000 Omega or Rolex, the model provides an estimate within roughly $900 of the market listing.

5.3 Future Directions

While the current model is highly effective for the $1,000–$20,000 segment, future iterations could expand its scope. Integrating **Computer Vision** to analyze the condition of the watch from photos, or implementing **Natural Language Processing (NLP)** to extract sentiment from seller descriptions, could further reduce the error margin. Additionally, expanding the dataset to include "sold" prices rather than "asking" prices would provide an even more accurate reflection of true market liquidity.

In conclusion, the project demonstrates that despite the fragmented and often inconsistent nature of secondary luxury markets, a data-driven approach combined with ensemble learning can provide a reliable, automated alternative to manual horological appraisal.

A Technologies Used

A.1 Data Handling & Processing

- **pandas**: Used for loading, cleaning, filtering, and manipulating structured data from CSV, JSONL, and DataFrames.
- **numpy**: Provides support for mathematical operations, logarithmic transformations, and handling numerical arrays.
- **re**: Uses regular expressions to extract specific text patterns, such as dimensions or years, from raw strings.
- **csv**: Used to read and write data directly into the comma-separated values format for storage and sharing.
- **currency_converter**: Fetches latest exchange rates to convert various international currencies into a unified USD format.

A.2 Web Scraping & Automation

- **undetected_chromedriver**: Automates a Chrome browser while using specialized techniques to bypass bot-detection security systems.
- **bs4 (BeautifulSoup)**: Parses the HTML source code of web pages to extract specific data like prices, names, and links.
- **selenium**: Handles browser interactions such as clicking "Next" buttons, scrolling, and waiting for page elements to load.

A.3 Machine Learning & Modeling

- **xgboost**: An optimized gradient boosting library used for high-performance regression with GPU acceleration.
- **lightgbm**: A fast, distributed gradient boosting framework designed for efficiency and lower memory usage.
- **CatBoostRegressor**: A gradient boosting algorithm that excels at handling categorical features and utilizes GPU power.
- **sklearn (Scikit-learn)**: Provides a massive suite of tools for model training, data splitting, preprocessing, and performance metrics.
- **category_encoders**: Implements advanced encoding techniques, like Target Encoding, to handle categories with thousands of unique values.

- **joblib**: Used to save trained models and preprocessing objects to disk and reload them for future predictions.

A.4 Visualization & Statistics

- **matplotlib.pyplot**: The primary library for creating static plots, charts, and figures for data reports.
- **seaborn**: A high-level interface used to create attractive statistical graphics like heatmaps, boxplots, and regression lines.
- **scipy.stats (ks_2samp)**: Performs statistical tests to determine if two datasets (like train and test sets) come from the same distribution.

A.5 System & Utilities

- **os**: Manages file system operations such as creating directories for plots and checking if files exist.
- **time**: Controls script timing by adding delays to prevent server overloads and simulate human browsing.
- **random**: Generates randomized intervals for delays to help the scraper appear more human-like.
- **datetime**: Provides tools for creating timestamps used to uniquely name log files and model versions.
- **warnings**: Used to suppress non-critical software alerts to keep the script's console output clean and readable.