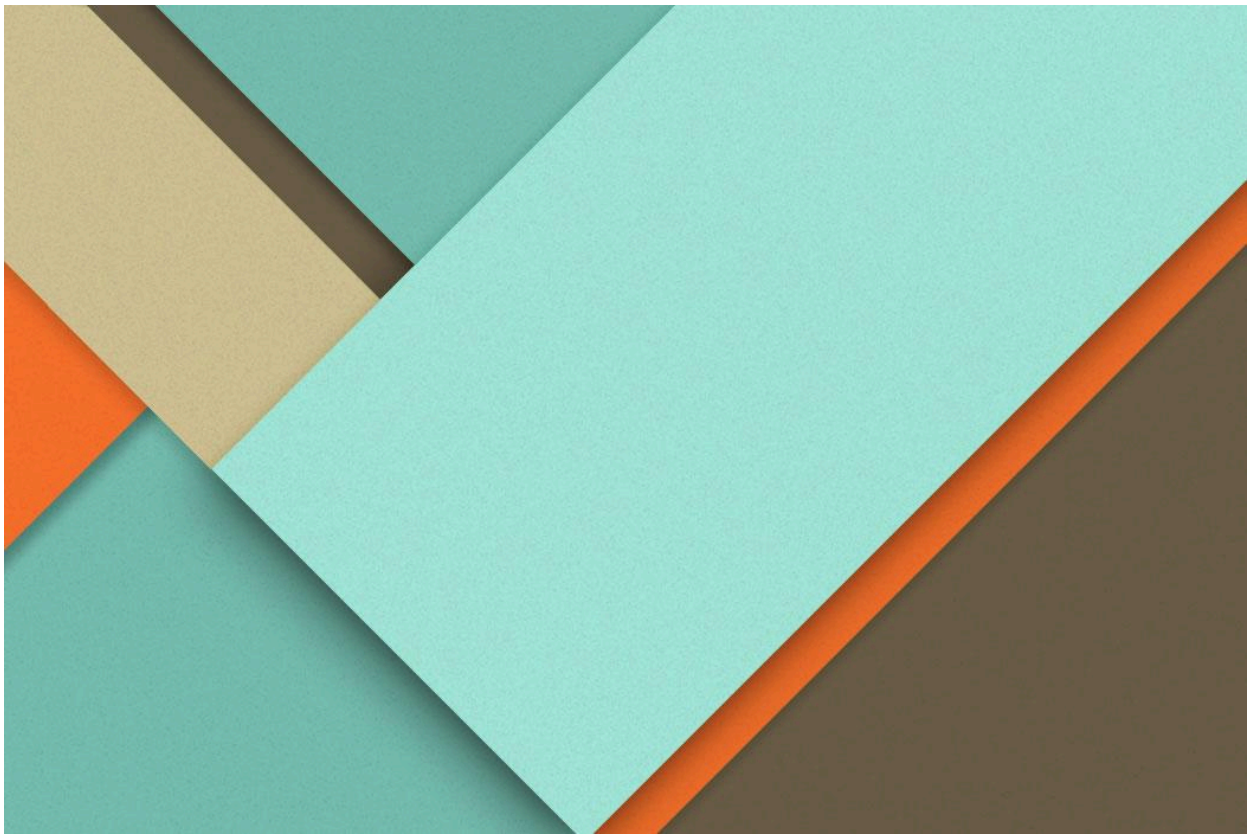


Data Quality Project Report

09.12.2024

—

Group 65

Assigned dataset: MOMA (Artists and Artwork)

Members

Sebastian Ballesteros - 10945208

Alberto Capoti - 11018580

Felipe Abadia Bermeo - 10992214

Set up choices

Libraries and Tools

For this project, we utilized several Python libraries to streamline data profiling, cleaning, and quality assessment:

- **Pandas**: For data manipulation and analysis.
- **ProfileReport (pandas-profiling)**: To generate an in-depth data profiling report for visual representation of the dataset.
- **NumPy**: For handling numerical computations during transformations and imputations.
- **Matplotlib/Seaborn**: For correlation analysis and exploratory data visualization.

Data Preparation Techniques

The project involved working with two datasets:

1. **Artwork Dataset**: Focused on cleaning and improving data quality.
2. **Artists Dataset**: Used as a reference for domain definitions and validation checks.

The decision to include both datasets ensured that we could make use of cross-references to improve the accuracy and consistency of the Artwork Dataset. Initial steps included:

- Mounting the Google Drive to access datasets.
- Importing libraries and configuring the environment.

IMPORTANT NOTE:

The pipeline described in the first part of the following report was performed without considering the objectives of the data analysis, i.e. the quality improvements and decisions were made in order to achieve a better general data quality in all fields and not only in a ML oriented one.

Pipeline implementation

Our pipeline implementation consisted of mainly two steps: data profiling and data cleaning. In both steps we perform a quality assessment. During the former step this assessment helps us identify areas of poor quality where we should focus our cleaning while the latter shows the improvement of the quality of the dataset after our cleaning is performed.

Data Profiling and Data Quality Assessment

Data Profiling and Exploration

After importing the datasets, an extensive profiling phase was conducted to:

- Understand the structure of the data.
- Identify potential issues such as missing values or inconsistent formats.

Using **ProfileReport**, we generated visual summaries to explore edge cases in columns such as:

- **Dates**: Variations in format.
- **Dimensions**: Inconsistencies in representation.
- **Medium**: Variability in entries.

The reason to use **ProfileReport** was due to its simplicity and its user-friendly outcome. The output of this report helped us understand many key points about the data we were working with. The next two figures represent the main information about our dataset and give a glimpse about its quality.

Dataset statistics

Number of variables	21
Number of observations	130262
Missing cells	839439
Missing cells (%)	30.7%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	20.9 MiB
Average record size in memory	168.0 B

Figure 1. Overall information about our dataset.

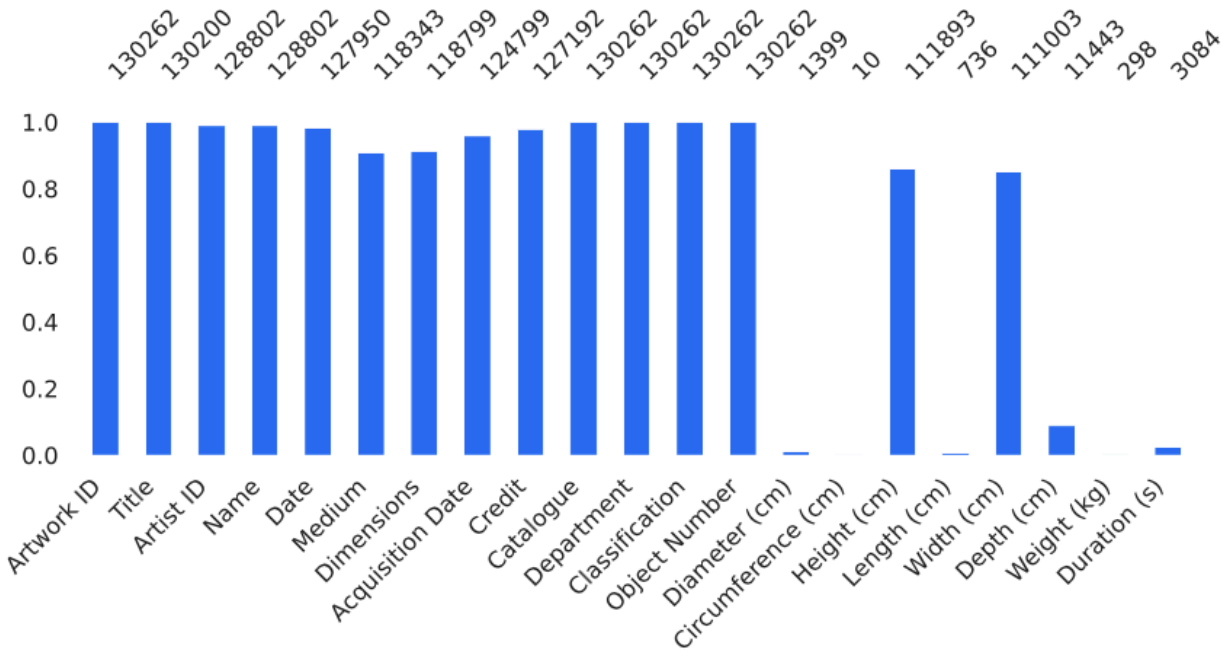


Figure 2. A simple visualization of level of nullity by column.

For example, the following *word cloud* of the **Dates** column suggests there are different variations in format.

Figure 3. *Word Cloud* representation of column Date.

These summaries helped pinpoint problematic columns requiring cleaning and standardization. In addition, it helped us identify null value formats that could be used when importing the data such as n.d. (no date), unidentified, unassigned, etc.

Exploration of Edge Cases

Further exploration was conducted to:

- Investigate columns with diverse formats or irregularities.
- Identify well-balanced columns with minimal missing values, which could serve as potential target variables for future analysis.

This exploration was executed by using the **pandas** built-in functions.

Correlation Analysis

Correlation matrices were generated to:

- Assess relationships between numeric variables.
- Aid in identifying patterns that could guide data transformations and imputations.

Data Quality Assessment

The quality assessment was divided into general and specific checks:

1. **General Assessment**:
    - **Uniqueness**: Identified duplicate entries.
    - **Completeness**: Assessed missing values on a per-column basis.
2. **Specific Assessments**:
    - **Accuracy**: Verified the correctness of data values by cross-referencing with the Artists Dataset. This involved calculating the proportion of accurate values in fields such as artist names.
    - **Consistency**:
        - **Films**: Ensured all films had a duration specified in seconds.
        - **Dates**: Verified logical consistency in the dates (e.g. acquisition dates after artwork date)
    - **Duplicates:**
        - Verification of duplicates based on the title of the artwork. We found that there were 27% of duplicates in the date - based only on the title. Once a deeper analysis of these duplicates was performed, it was evident that these were indeed duplicates since they shared all the other values.
        - We decided to use the `duplicated()` built-in function of *pandas* rather than other methods (i.e. sound-wise methods) to perform this analysis since it gave robust results and succeeded at identifying duplicates. More information about deduplication is given later.

Evaluation Metrics

- **Accuracy** : Number of accurate values / Total number of values
- **Consistency**: Number of consistent tuples / Total number of tuples
- **Duplication (overall)**: Number of duplicates / Total number of tuples
- **Uniqueness (column-wise)**: Number of unique values / Total number of tuples
- **Completeness (column-wise)**: Number of available tuples / Total number of tuples

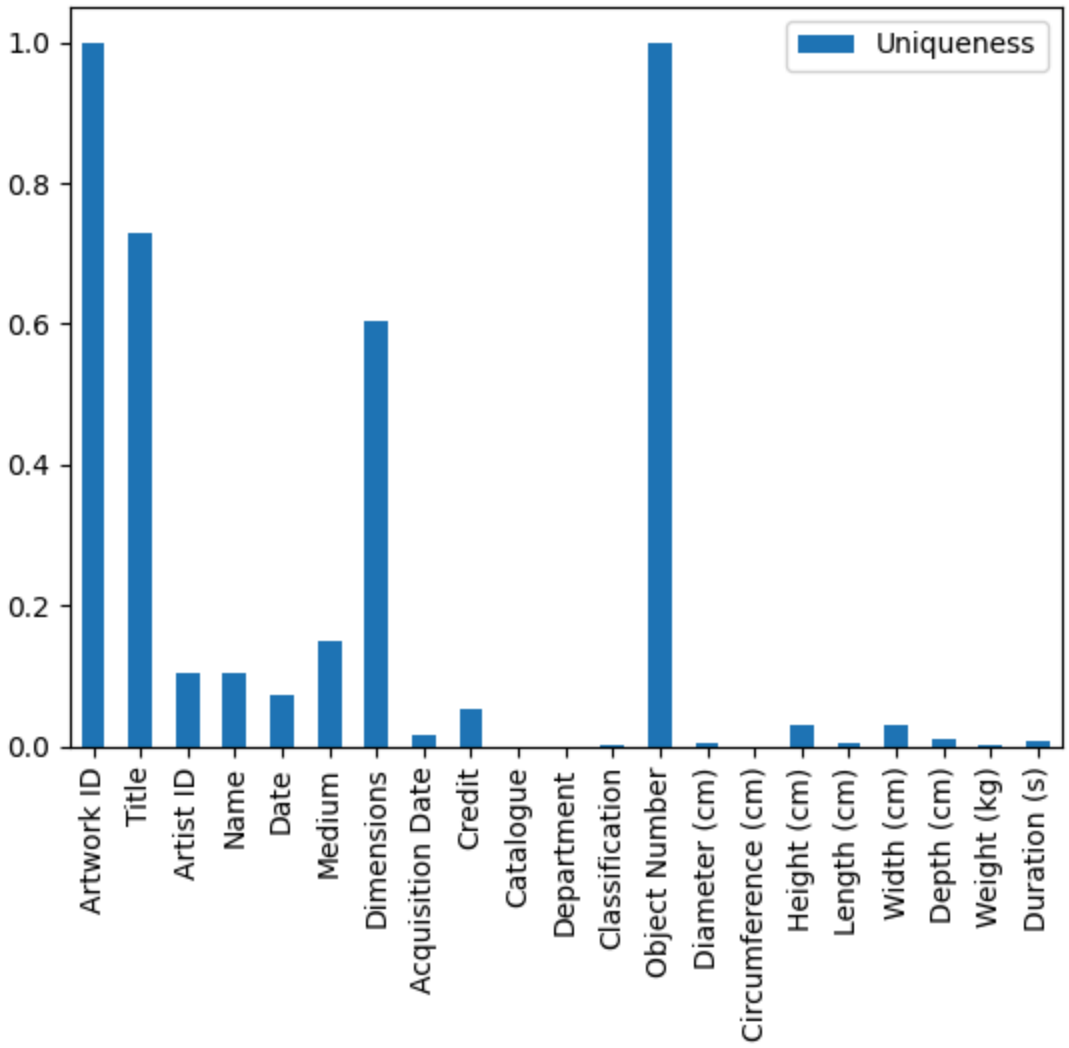


Figure 4. Uniqueness assessment per column.

As seen in the previous figure, we can see that there's roughly 70% of uniqueness across artwork titles and 10% of uniqueness when considering artists' names.

Data Cleaning

Data Transformation (Wrangling)

1. **Dropping of useless columns or columns with high level of null values**

    - Dropped columns such as **Diameter**, **Circumference**, **Length** due to their low completeness. However, we kept **Duration** since it may be useful for video or film types.
    - Similarly, we dropped Object Number due to its low relevance.
2. **Standardization of Date Formats**:
    - Extracted relevant year information from dates with inconsistent formats. Two new columns (Start year, End year) were added in place of the original Date column.
3. **Standardization of Department and Classification of artwork**:
    - We applied standardization across all departments and classification. For example, “&” were changed to “and” to ensure a coherent format.
4. **Standardization of Acquisition Date formats**
    - The most common format in the column was kept (YYYY-MM-DD). For values that had other formats like YYYY-MM, it was added a “-01” for the day.
5. **Artists involved in the artwork**
    - Some values in the column "Artist ID" consist of lists of the artists that collaborated to do the artwork. To fix this and obtain more relevant information, 3 new columns were created at the place of the original column: “Num. Artists” stores the count of artists involved, “Main artist” keep only the first artist in the list, “Secondary artists” keep the remaining artists in a list.

Error Detection and Correction

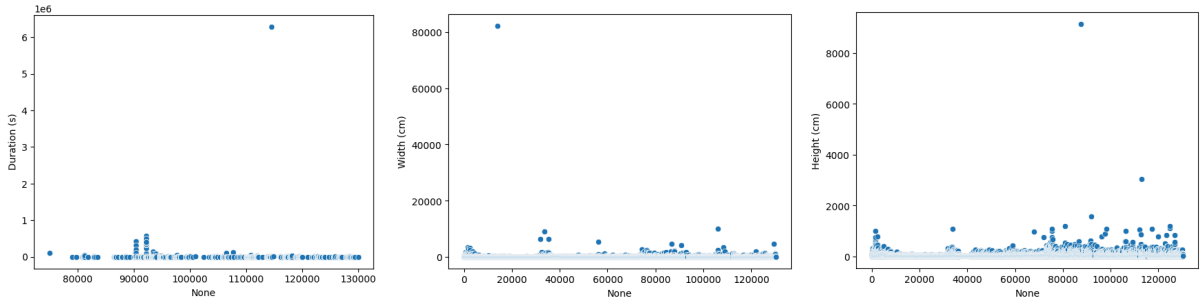
1. **Correction of 'Duration (s)':**
    - Imputed null values in the `Duration(s)` column for video/film objects by preprocessing information in the `Dimensions` column.
2. **Correction of 'Height' and 'Width':**
    - Null values in the `Height` and `Width` columns were imputed using values derived also from the `Dimensions` column.
3. **Handling Missing Values:**

- Performed targeted imputations and applied final transformations, such as standardizing data types for date columns.
- For dates, we decided to use single imputation and replace it with standard values rather than forward or backward propagation due to its effectiveness.
- The columns “Medium”, “Acquisition Date”, “Credit” and “Classification” were imputed as “Unspecified”.

The choice of using placeholders values to replace missing one (instead of using another more specific imputation technique) was adopted under the assumption that it’s better to let people interested in the content of the dataset know when a value is missing (through a placeholder value) rather then insert a “synthetic” value which could pass for the truth.

Correction of Outliers:

We analyzed the updated content of the 3 numerical column of Duration,Height and Width through a scatter plot and in all 3 of them was clear the presence of few values that were not only distant from all the others but were also infeasible (e.g. film duration of over 1.000.000 seconds ≅ 278 hours).



After discarding this values we proceed with a more complete analysis of the outliers:

- For the duration column analysis we used the Z-score technique (a statistical technique that assumes that the value distribution of the attribute analyzed follows a normal distribution and identifies as outliers values that are outside or at the edge

of it). We decide to use the median rather than the average to improve the robustness of the method.

- For the height and width column we opted for a multivariate analysis that we performed using a local distance-based outlier detection method such as the computation of the LOF (local outlier factor)

In both cases we decided to just pinpoint the probable outliers without removing them since most of them could just be strange, but not incorrect, values. In our opinion the decision over these values should be left to a specialist of the field from which the data come from.

Data Deduplication

As previously mentioned, we had already identified some duplicates and at this point we decided to drop them. Specifically, we utilized exact string similarity using the `duplicated()` function and `drop_duplicates()` to consolidate records using the title of the artwork as a subset. To further ensure the elimination of all duplicates we used the `Sorted Neighborhood` technique to reduce the searching space in the hunt for duplicates and a compare method that included string similarity measures for 3 columns (Name, Title and Medium) and exact matching for 4 others (Department, Main artist ID, Start year and End year). If two tuples respected all 7 comparisons they were assessed as duplicates and one of them was dropped.

Additional Data Quality Assessment

After completing the cleaning pipeline, a second data quality assessment was performed to validate:

1. Improvements in **accuracy** and **consistency** metrics.
2. Reduced proportion of missing and inconsistent values.
3. Enhanced uniqueness by eliminating duplicates.

In all 3 analyses the improvements made to the quality of data through the use of the implemented data cleaning pipeline were self-evident.

Data analysis

We decided to take as the goal of our data analysis the imputation of the categorical value Department. In order to do this we trained 4 different models of classification methods and compared them. Furthermore we performed the same analysis twice, once using the dataset that we had at the beginning of the pipeline (from now on referred to as dirty dataset) and once using the dataset at the end of it (clean dataset).

Data Preparation

In order to perform the analysis was mandatory a preparation of the datasets, both of them were modified to allow the execution of the classification methods.
Specifically we:

1. Converted the attributes Catalogue and Department from categorical to numerical
2. Extracted the year from the Acquisition date
3. Dropped the columns Artwork ID, Credit, Classification, Title, Main artist ID, Secondary artists ID, Name and Medium

This columns were dropped for 3 different reason:

- Their values were irrelevant to the process (e.g. Artwork ID)
- It was impossible to convert them from categorical to numerical without generating a huge amount of classes (e.g. Title)
- Using them would have made the classification process trivial due to the huge correlation between them and the target variable (e.g. Classification)

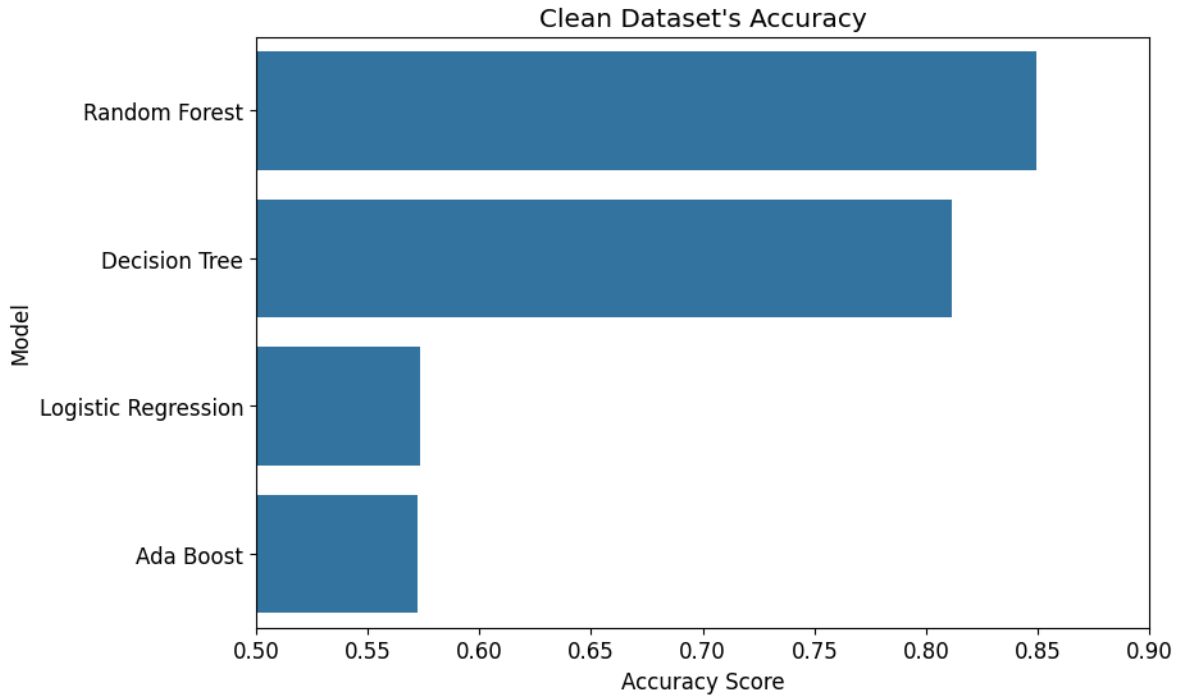
A similar preprocessing was performed on the dirty dataset with the fundamental difference that we were forced to drop all the dates column (whose importance would prove to be high later) due to their incorrect format that rendered impossible to extract from them a numeric value.

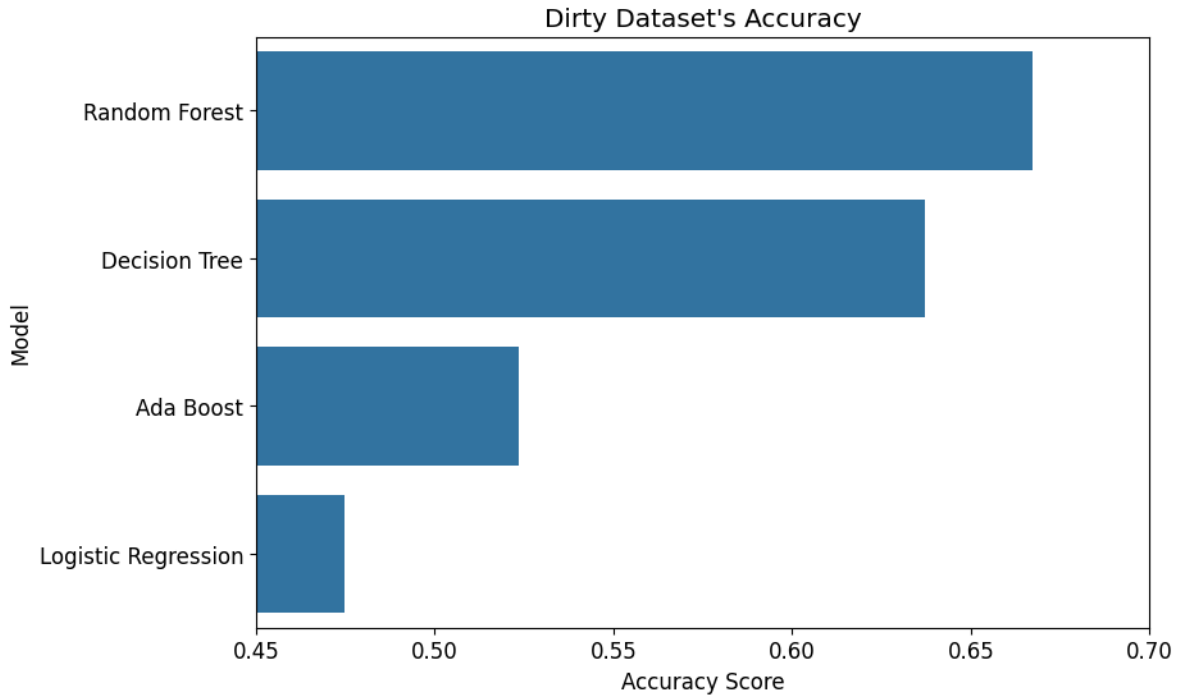
Additionally, during the data preprocessing phase, we decided to drop the "classification" feature because it was highly correlated with the target variable, which would have made the analysis trivial and resulted in perfect accuracy. By removing this feature, we ensured that the models were tested on meaningful data.

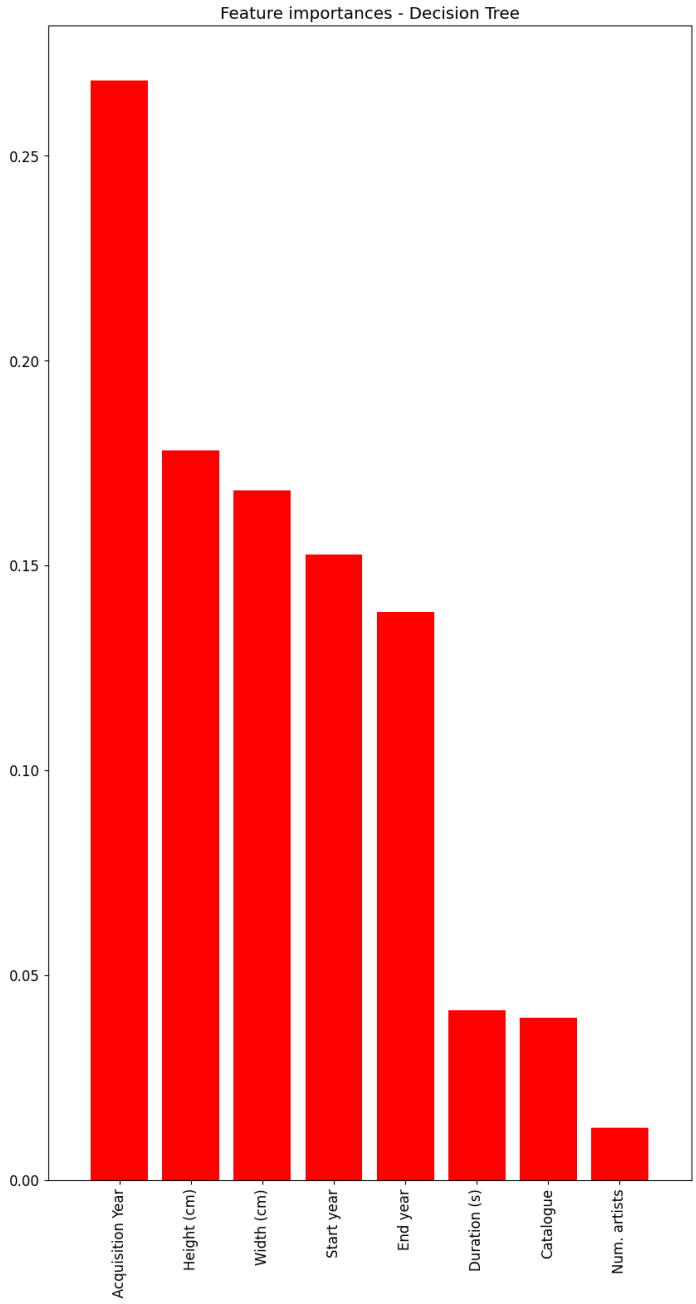
Results

We implemented and evaluated four classification models—Logistic Regression, Decision Tree, Random Forest, and AdaBoost—on both a clean and a dirty dataset to assess their performance in terms of accuracy, precision, recall, and F1 score. Logistic Regression, a linear model, applies a logistic function to the input features. The Decision Tree, a non-linear model, splits the data based on feature values to make predictions, offering high interpretability. Random Forest, an ensemble of decision trees, improves predictive performance through bagging, while AdaBoost combines weak learners (Decision Trees) by iteratively reweighting misclassified samples. The accuracy values are shown in the figures below while the other metrics can be found in the code. On the clean dataset, Random Forest performed the best, achieving an accuracy of 0.847, followed by Decision Tree with 0.809, AdaBoost with 0.636, and Logistic Regression with 0.565. However, when applied to the dirty dataset, all models experienced a significant drop in performance. Random Forest still performed the best, with an accuracy of 0.667, while Logistic Regression had the lowest accuracy of 0.475. The Decision Tree and Random Forest models showed more stability across both datasets, while Logistic Regression and AdaBoost were more sensitive to the quality of the data. Overall, the results indicate that data cleaning has a substantial positive impact on model performance, with Random Forest proving to be the most robust model in this analysis.

Finally, we included a figure to show the features that contributed most to the analysis. In this case, we noted that Acquisition Year (which was not included in the dirty dataset), Height and Width were the most important and relevant features for the models to determine a classification output.


Clean Dataset's Accuracy
Random Forest
Decision Tree
Logistic Regression
Ada Boost
Model
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
Accuracy Score


Dirty Dataset's Accuracy
Random Forest
Decision Tree
Ada Boost
Logistic Regression
Model
0.45
0.50
0.55
0.60
0.65
0.70
Accuracy Score


Feature importances - Decision Tree
0.25
0.20
0.15
0.10
0.05
0.00
Acquisition Year
Height (cm)
Width (cm)
Start year
End year
Duration (s)
Catalogue
Num. artists