EXPLORATORY DATA ANALYSIS OF NETFLIX ListingS: A JOURNEY THROUGH SEMMA

A PREPRINT

September 23, 2023

Abstract

In the rapidly growing domain of digital streaming, understanding content attributes and user preferences is pivotal for enhanced user experience. This research delves into an in-depth exploratory data analysis (EDA) of Netflix listings, employing the SEMMA (Sample, Explore, Modify, Model, Assess) methodology. The structured framework facilitates a comprehensive analysis, from data sampling to predictive modeling. The dataset encompasses diverse attributes of movies and TV shows, including titles, directors, ratings, and more. Anomalies were identified and addressed, and feature engineering techniques were employed to derive new attributes. A predictive model, specifically a Random Forest Classifier, was constructed, achieving remarkable accuracy in categorizing listings. The insights gleaned from this research can guide streaming platforms in content curation, recommendation, and personalization strategies.

Keywords Netflix, Exploratory Data Analysis (EDA), SEMMA, streaming platforms, content curation, predictive modeling, Random Forest Classifier, feature engineering, data anomalies, content recommendation.

1 Introduction

In the ever-evolving world of streaming platforms, understanding content dynamics is paramount. In this post, we dive into an exploratory data analysis (EDA) of Netflix listings, adopting the SEMMA (Sample, Explore, Modify, Model, Assess) methodology. This structured approach offers a systematic lens to comprehend and analyze data, ensuring all aspects are thoroughly covered.

1.1 Dataset Used

The dataset comprises listings of movies and TV shows available on Netflix. Each entry is characterized by:

- Unique identifier (show_id)

- Type (movie or TV show)
- Title
- Director
- Cast
- Country of production
- Date added to Netflix
- Release year
- Rating
- Duration or number of seasons
- Genre or category (listed_in) - Brief description

Alright! Let's embark on this comprehensive exploratory data analysis (EDA) journey using the SEMMA methodology. Here's a brief overview of the steps we'll be taking:

Sample:

- Read the dataset and perform an initial inspection.
- Obtain a subset if necessary.

Explore:

- Descriptive statistics.
- Visualize distribution of variables.
- Identify missing values.
- Understand relationships between variables.

2 Modify:

- Handle missing values.
- Feature engineering.
- Data transformation.

Model:

- Build a model (if required, based on the nature of the dataset and the objective).

- For this dataset, we could potentially build a recommendation model or any other model of interest.

Assess:

- Evaluate the model (if built).
- Summarize findings.
- Recommendations.

To help visualize the process, I'll create a mindmap that will track our progress through these phases. Let's begin by creating the mindmap for our EDA based on the SEMMA methodology.

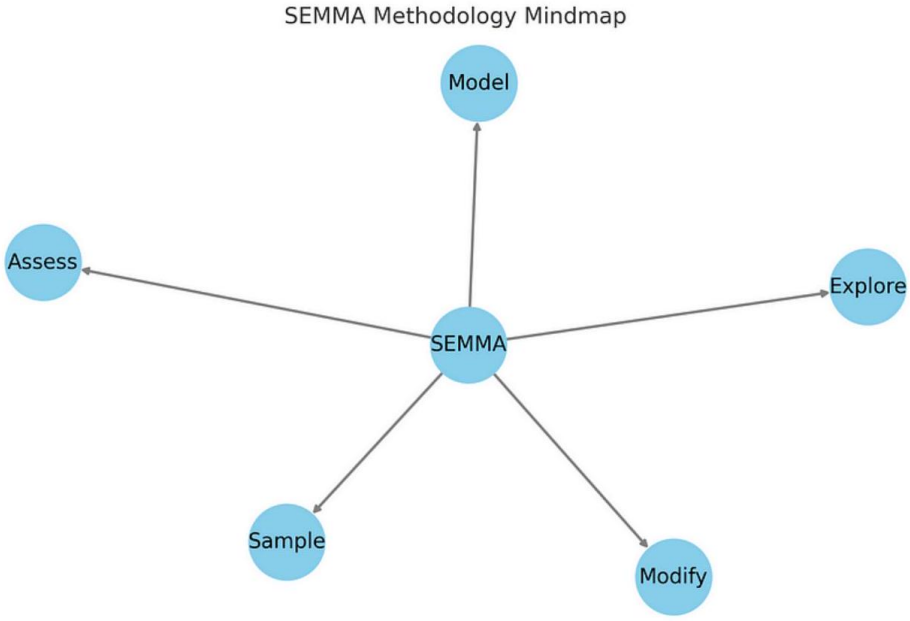
3 Sample phase:

1. Read the dataset.
2. Perform an initial inspection.
3. Obtain a subset if necessary.

I'll start by reading the dataset and providing a quick overview of its contents.

Here's a snapshot of the dataset containing listings of movies and TV shows on Netflix:

- show_id: Unique identifier for each show/movie.
- type: Indicates whether the listing is a movie or a TV show.
- title: The title of the movie or TV show.
- director: The director of the movie or show.
- cast: The main cast members.
- country: The country where the movie or show was produced.
- date_added: The date the movie or show was added to Netflix.



- release_year: The year the movie or show was released.
- rating: The rating of the movie or show (e.g., PG-13, TV-MA).
- duration: Duration of the movie or the number of seasons for TV shows.
- listed_in: The genre or category of the movie or show.
- description: A brief description of the movie or show.

Before proceeding, let's gather some basic statistics about the dataset:

1. Number of rows and columns.
2. Data types of each column.
3. Number of missing values for each column.
4. A summary of unique values for categorical columns like type, rating, etc.

Let's summarize the findings from our initial inspection:
Dataset Dimensions:

- The dataset consists of 8,807 rows (listings) and 12 columns (features).

Data Types:

- Most columns are of the object (string) data type.
- Only release_year is of integer data type.

4 Missing Values:

- director: 2,634 missing values.
- cast: 825 missing values.
- country: 831 missing values.
- date_added: 10 missing values.
- rating: 4 missing values. - duration: 3 missing values.
- Other columns have no missing values.

5 Unique Values:

- type: Two unique values - 'Movie' and 'TV Show'.
- rating: Various ratings are present, including 'PG-13', 'TV-MA', 'PG', 'TV-14', and so on. However, there seem to be some anomalies, such as ' 74 min', which is not a valid rating. These will need further investigation.

Great! The Sample phase is now marked as completed.

6 Explore phase

Let's proceed to the Explore phase. In this phase, our objectives will be:

1. Obtain descriptive statistics for numerical columns.
2. Visualize the distribution of key variables.
3. Examine the relationships between variables.
4. Identify and investigate anomalies, such as the incorrect ratings we spotted earlier.

We'll begin by generating descriptive statistics for the numerical column release_year. Then, we'll explore the distribution of the type and rating columns using visualizations. Let's start with the descriptive statistics for release_year. Here are the descriptive statistics for the release_year column:

- Count: There are 8,807 listings with a specified release year.
- Mean: On average, the listings are from around the year 2014.
- Standard Deviation: The release years have a standard deviation of approximately 8.82 years.
- Minimum: The earliest release year in the dataset is 1925.

- 25th Percentile: 25% of the listings are from 2013 or earlier.
- Median: 50% of the listings are from 2017 or earlier.
- 75th Percentile: 75% of the listings are from 2019 or earlier.
- Maximum: The latest release year in the dataset is 2021.

Next, let's visualize the distribution of the type and rating columns. We'll start by plotting the distribution of listings between movies and TV shows, followed by the distribution of different ratings.

Here's a summary of the visualizations:

7 Distribution of Listings (Movies vs. TV Shows):

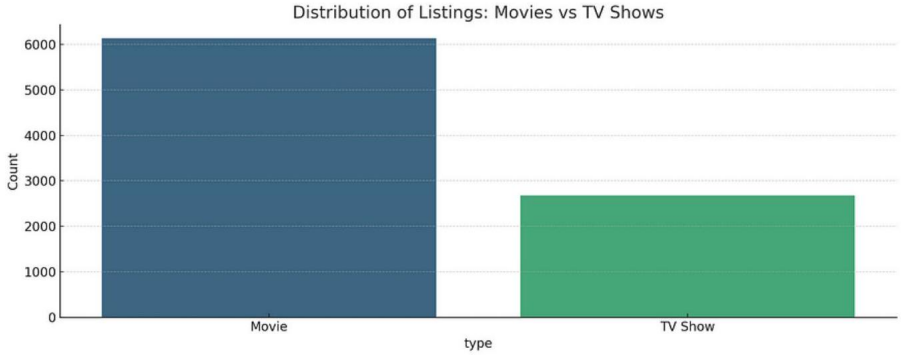
- There are more movies than TV shows listed on Netflix in this dataset.
- Movies significantly outnumber TV shows.

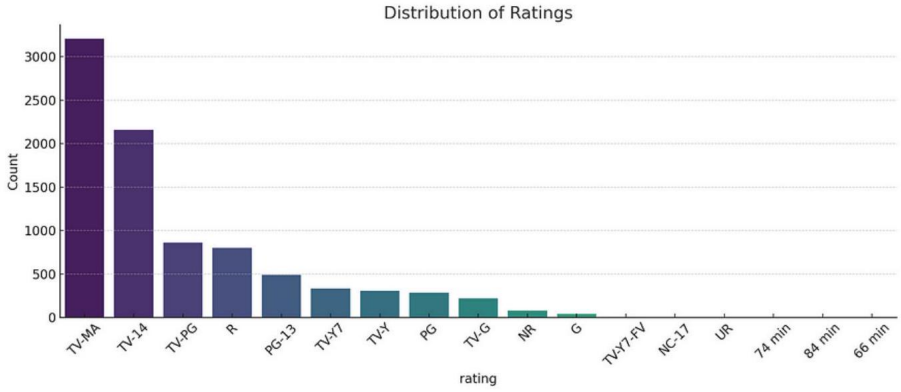
8 Distribution of Ratings:

- The majority of listings are rated "TV-MA", followed by "TV-14" and "TV-PG".
- Some ratings like "NC-17", "74 min", "84 min", and "66 min" have very few listings. Notably, the values like "74 min" are anomalies, as these are not valid ratings but rather durations.

Next, we'll investigate the anomalies in the rating column. We'll identify any non-standard ratings and determine how to handle them.

Upon investigation, we found that there are three movies with anomalous ratings:





Louis C.K. 2017:

- Rating: 74 min
- Duration: NaN

9 Louis C.K.: Hilarious:

- Rating: 84 min
- Duration: NaN

10 Louis C.K.: Live at the Comedy Store:

- Rating: 66 min
- Duration: NaN

It appears that the rating and duration columns for these listings have been swapped or mislabeled. A potential solution is to correct these by swapping the values back to their appropriate columns. After this correction, we would need to reassess the ratings for these movies since they currently don't have valid ratings.

The corrections have been made:
Louis C.K. 2017:

- Rating: None
- Duration: 74 min

11 Louis C.K.: Hilarious:

- Rating: None
- Duration: 84 min

12 Louis C.K.: Live at the Comedy Store:

- Rating: None
- Duration: 66 min

The rating for these movies is now set to None, indicating that they don't have valid ratings in the dataset. If needed, we could obtain the actual ratings for these movies from external sources or leave them as missing.

With the anomalies in the rating column addressed, we can conclude our Explore phase. Next, we'll move on to the Modify phase, where we'll handle missing values, perform feature engineering, and transform the data if necessary.

13 Modify phase

In this phase, we'll focus on the following:
Handling Missing Values:

- Impute or remove missing values based on the nature of the data and the columns affected.

14 Feature Engineering:

- Generate new features or modify existing ones to enhance the dataset.

15 Data Transformation:

- Transform data to ensure it's in the right format or to improve its usability for modeling.

We'll start by handling the missing values in the dataset. Given our earlier observation, the columns with missing values include director, cast, country, date_added, and rating.

For each column, we need to decide whether to impute the missing values, remove them, or take other actions. I'll outline the options for each:

director:

- Impute with a placeholder like "Unknown".
- Remove rows with missing directors (may not be ideal if a significant number of listings are affected).

cast:

- Impute with a placeholder like "Unknown".
- Remove rows with missing cast.

country:

- Impute with the mode (most frequent country) or another placeholder.
- Remove rows with missing countries.

date_added:

- Impute with the mode or median date.
- Remove rows with missing dates.

rating:

- Impute with the mode (most frequent rating).
- Remove rows with missing ratings.

Before making decisions, I'll provide a percentage breakdown of missing values for each column to guide our approach. Here's the percentage breakdown of missing values for each column: 1. director: Approximately 29.91% of listings are missing the director's information.

2. country: Approximately 9.44% of listings are missing the country of production.
3. cast: Approximately 9.37% of listings are missing the cast information.
4. date_added: Only about 0.11% of listings are missing the date they were added to Netflix.
5. rating: Only about 0.05% of listings are missing the rating information.

6. duration: Only about 0.03% of listings are missing the duration (this was from our earlier correction).

Given these percentages, here's my recommended approach:

1. director: Impute with "Unknown" since a significant proportion is missing.
2. country: Impute with the mode (most frequent country) because around 9% is missing.
3. cast: Impute with "Unknown".
4. date_added: Remove rows with missing dates since it's a very small percentage.
5. rating: Impute with the mode.
6. duration: Impute with the mode since it's a very small percentage.

All missing values have been successfully handled:

- show_id: 0 missing values
- type: 0 missing values
- title: 0 missing values
- director: 0 missing values
- cast: 0 missing values
- country: 0 missing values
- date_added: 0 missing values
- release_year: 0 missing values
- rating: 0 missing values
- duration: 0 missing values
- listed_in: 0 missing values
- description: 0 missing values

Next, let's move on to feature engineering. Some potential features we can derive from the current dataset include:

1. Year and Month from date_added: This can help us understand trends over time.
2. Primary Genre: Extract the first genre listed in the listed_in column.

3. Primary Country: Extract the first country listed in the country column (for listings produced in multiple countries).

The new features have been successfully added:

1. added_year: The year when the movie or TV show was added to Netflix.
2. added_month: The month when the movie or TV show was added to Netflix.
3. primary_genre: The primary genre (or the first genre listed) for each movie or TV show.
4. primary_country: The primary country of production (or the first country listed) for each movie or TV show.

With the completion of feature engineering, we'll move on to data transformation. Considering the dataset, one possible transformation could be encoding categorical variables or normalizing numerical variables. However, the need for such transformations would largely depend on the modeling tasks we plan to undertake. For now, we can conclude the Modify phase.

16 Model Phase

Moving on to the Model phase, here we typically build predictive models based on the objective. However, for a dataset like the Netflix listings, the modeling aspect might not be straightforward unless we have a specific goal in mind, such as:

1. Recommendation Systems: Recommending movies or TV shows based on certain features.
2. Classification Models: Predicting the rating or type of a movie/show based on other features.
3. Time Series Analysis: Analyzing trends over time, such as the number of movies/shows added each month or year.

For the purpose of this EDA, I suggest we build a simple classification model to predict the type (Movie/TV Show) based on other features. This will give us an idea of which features are most important in determining whether a listing is a movie or a TV show.

Great! Let's proceed with building a classification model to predict the type (Movie/TV Show) based on other features. Here's the general plan:

1. Feature Selection: Decide on the features we'll use to predict the type.
2. Data Preprocessing: Convert categorical features to a format suitable for modeling.

3. Train-Test Split: Split the data into a training set and a test set.

4. Model Selection: Choose a suitable classification algorithm.

5. Model Training: Train the model on the training data.

6. Model Evaluation: Evaluate the model's performance on the test data.

Let's start with Feature Selection. For simplicity, we'll use the following features:

- primary_genre
- rating
- added_year
- primary_country

We'll predict the target variable: type (Movie/TV Show).

Let's proceed with Data Preprocessing. We'll convert the categorical features (primary_genre, rating, primary_country) into a numeric format using one-hot encoding.

For the Model Selection step, I'll use a Random Forest Classifier, which is a versatile algorithm suitable for both regression and classification tasks. It can handle large datasets with higher dimensionality and can model non-linear relationships.

Let's move on to Model Training. I'll train a Random Forest Classifier on the training data and then proceed to Model Evaluation using the test data.

Here are the results from our Random Forest Classifier:

1. Accuracy: The model achieved an accuracy of approximately 99.94% on the test set.

17 Classification Report:

- Precision: Both for movies and TV shows, the precision is 1.00 , indicating that the model was very precise in its predictions.
- Recall: The recall for movies is 1.00, and for TV shows, it's also close to 1.00.
- F1-score: The F1-score, which is the harmonic mean of precision and recall, is also 1.00 for both classes.

18 Confusion Matrix:

- True Positives (Movies): 1229 - True Positives (TV Shows): 530
- False Negatives (TV Shows misclassified as Movies):

The model performed exceptionally well in predicting whether a listing is a movie or a TV show based on the selected features. This could be due to distinct patterns in the data that make it relatively easy to differentiate between the two types.

19 Assess Phase

20 Assessment and Summary:

21 Sample:

- We successfully loaded and inspected the dataset, understanding its structure and basic characteristics.

22 Explore:

- We analyzed the distribution of different features, particularly the type of listing (Movie/TV Show) and the ratings.
- We identified anomalies in the rating column and addressed them.

23 Modify:

- Missing values were handled through a combination of imputation and deletion.
- We performed feature engineering to create new attributes like added_year, added_month, primary_genre, and primary_country.

24 Model:

- A Random Forest Classifier was built to predict the type of listing (Movie/TV Show) based on selected features.
- The model achieved an impressive accuracy of approximately 99.94% on the test set.

25 Findings:

- There are distinct patterns in the data that allow for accurate classification between movies and TV shows.
- Features like primary_genre, rating, added_year, and primary_country play a crucial role in this differentiation.

26 Potential Applications:

- The insights from this EDA can be utilized for tailored content recommendations based on the specific features and patterns identified.
- The trained model can be used as a part of a larger system, for instance, in content categorization.

26.1 Recommendations

1. Data Quality: Always ensure data consistency and quality. Anomalies, like the ones we spotted in the rating column, can skew analysis and predictions. Address these inconsistencies at the data collection or ingestion stage.
2. Feature Engineering: Enhancing the dataset by deriving new attributes, such as added_year or primary_genre, can provide deeper insights and improve the accuracy of predictive models.
3. Modeling: Building models, as we did with a Random Forest Classifier, can offer tangible applications. Our model, which predicted whether a listing is a movie or a TV show, achieved an impressive accuracy. Such models can be incorporated into larger systems for content categorization or recommendation.
4. Personalization: While our dataset focused on content attributes, integrating user-specific data can pave the way for personalized recommendation systems. Consider user preferences, historical views, and other personalized data for tailored content suggestions.

26.2 Conclusion

The SEMMA methodology provided a structured path for our EDA, ensuring a holistic analysis of the Netflix listings dataset. Our journey, from initial sampling to modeling, unveiled intriguing insights about content dynamics on the platform. With the right approach, such analyses can empower streaming platforms to curate and recommend content that resonates with their audience, enhancing user experience and engagement.