Application of Machine Learning in Television Rating Prediction

Fedor Chursin (220904)

DISCOVER YOUR WORLD

Applied Data Science & AI

Friday 20 January 2023

Breda University OF APPLIED SCIENCES

Application of Machine Learning in Television Rating Prediction

Fedor Chursin

Breda University of Applied Sciences

Applied Data Science & Artificial Intelligence

Bhushan Nitin

January 20, 2023

Abstract

This report explores the application of machine learning in television rating prediction. The first section of the report covers exploratory data. The second section focuses on the various machine-learning techniques that can be used to predict television ratings, including linear regression, and decision trees. The final section of the report discusses the ethical considerations that must be considered when using machine learning for television rating prediction. Overall, this report highlights the potential of machine learning to improve the accuracy of television rating predictions and its potential to inform programming decisions in the media industry.

Table of Contents

Application of Machine Learning in Television Rating Prediction

Nowadays one of the most used metrics for each TV show is its rating. Usually, it represents the feedback from the audience. Based on this metric company that produces the show can decide whether they proceed and make new episodes or stop the production, also show's rating influence the price that the production company can demand from the advertiser to put their product in the show. For every production house, it would be very useful to know the rating of the show beforehand so they can either make changes to increase the score or just stop the production.

One of the biggest production companies in the Benelux called "Banijay" hired me to create a machine learning model that will predict ratings of the TV show "OP1". Banijay's main specialisation is the production and distribution of shows they made, and it is essential for them to know the show's popularity in advance. Based on this data they can set a reasonable price for the show and convince broadcasting companies to buy it.

The goal of this project is to develop the optimal model for forecasting show scores based on available data, ultimately resulting in increased revenue for the production company.

1 Exploratory Data Analysis

The main goal of this project was to create a machine-learning model that will predict the ratings of future shows. To create such a model, we need data and to increase the accuracy of the model we need cleaned and well-organised data. That is why cleaning and exploring data were crucial aspects of this project. We were given three different datasets:

- Content Data - this dataset included such info as the show's id and length, the broadcast's date, the show's hosts, keywords, summary and title.
- Rating Data - this data collection gave us an insight into the show's ratings by providing its target groups, broadcast and rating type, and coverage.
- Twitter Data - this set mainly had information about tweets related to episodes of the show, which allowed us to get live feedback rather than raw numbers.

I divided my exploratory data analysis into three phases to increase my efficiency and understand the data better. This approach helped me to get ready for modelling in advance by finding new patterns in data and investigating correlations between different metrics. Further, I will elaborate on each phase of my EDA.

1.1 First Phase - Cleaning, Exploring and Transforming Data:

I mostly worked with the content and rating data at the beginning of my analysis. Like in any other exploratory data analysis, cleaning all the data collections was the first step. Fortunately, the data provided by Banijay was well structured and I only had to drop NaN values and remove duplicate columns. After making these changes to both datasets, I only had to change the "Date" column type from "Object" to "Date_time".

As I used Twitter API to scrape the data from this social media, the outcome data was well formatted and clean. I only had to change the data format in the "created_at"

column to make merging possible and drop all the tweets that were a reference to a different tweet to better understand the engagement rate.

After performing these changes, my data was ready to be merged, so I proceeded to the next part of my EDA.

1.2 Second Phase - Merging Data:

The second part of the data analysis was the essential one as without merged data I could not make a proper analysis and could not create an ML model. This phase also consists of two different tasks.

At first, I had to merge the rating data with the content data. This merge was surprisingly hard because of the amount of data we had to deal with. After trying several algorithms for matching data to prepare it for the merge I came up with the most efficient one. The illustration in Fig. n shows peace of code with the implementation of the algorithm. It iterates through each element in the data collection and applies a constant time operation (pandas loc method which run-time is O(1)) the run-time of the whole algorithm results in O(n) where n is the number of observations in the dataset. After running the function, I still had to clear the data as it returns a “Series data frame” then I was finally able to merge it.

Merging Twitter data was easier as I could match two datasets by the data column and just use the merge method from the pandas. After merging these datasets, I calculated the Twitter engagement rate for each show.

Then comes the end of the EDA phase where I merged the data. After I was able to visualise and analyse the data I got.

1.3 Third Phase - Visualising and Analysing Data:

In the last part of my analysis, I managed to visualise data and explore new patterns in the data that were not identified before:

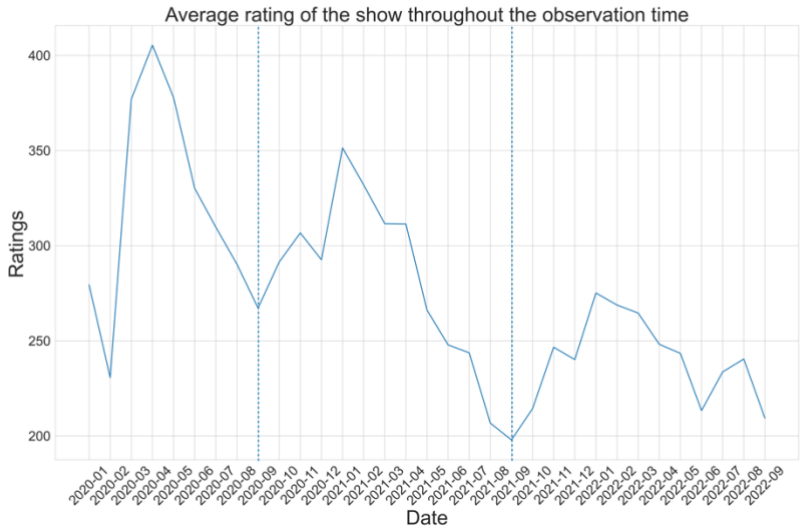


Figure 1 Average rating of the show throughout the observation period

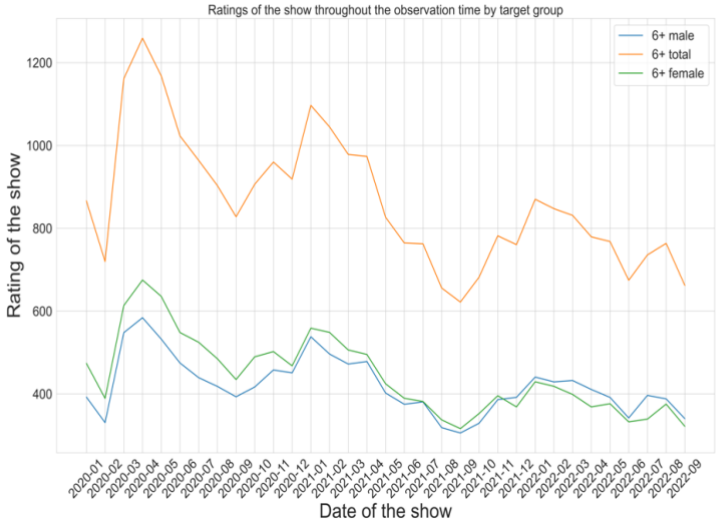
1. I decided to begin by visualising the average rating throughout the observation period. I found an interesting pattern: at the beginning of each year there is a spike in ratings and then throughout the rest of the year ratings tend to decrease. Fig 1 depicts my findings split into three timelines of increase and decrease rounds. Because of this pattern, I decided to use a quarter of the year when the show is broadcasted as one of the features of the model.

2. After finding such a trend in the data, I plot the same metrics for each target group to find out which age group is watching this show and if there is a gap in the show's popularity between the metrics for the male and female audience.

My findings were:

Figure 2 Ratings of the show throughout the observation time by target group

Figure 3 Ratings of the show throughout the observation time by target group



a. As you can see from Fig. 2 this show is most popular among people who are over 50 years old. Representatives of the 35-49-year-old target group still watch it, but the popularity of this show tends to be zero for all other

age groups. I included all the age groups in the model's features as this type of data has a high impact on the rating of the show.

b. From Fig. 3 it is clear that OP1 covers both topics as interesting for female and male audiences. That is why there is no gap in ratings for these two target groups.

3. For each episode directors of the show choose their host team. Hosts of the show create the atmosphere and energy viewers get from watching the episode. It was clear to me that I should track the correlation between the hosts and the popularity of the show. Indeed, there is a huge difference in the show's rating depending on the host team. Fig. 4 shows the top 5 of the best-rated host teams. Therefore, I also included host teams as a feature for my future model.

```
The top 1 hosts team is Pauw, Jeroen, Ekiz, Fidan with the average rating of 451.1538566904605.
The top 2 hosts team is Napel, Carrie ten, Groenhuijsen, Charles, Sijtsma, Welmoed with the average rating of 450.89120318235297.
The top 3 hosts team is Ekiz, Fidan, Pauw, Jeroen with the average rating of 443.36266281100325.
The top 4 hosts team is Ostiana, Giovanca, Brink, Tijs van den, Fikse, Margje with the average rating of 442.91306596862745.
The top 5 hosts team is Sijtsma, Welmoed, Groenhuijsen, Charles with the average rating of 419.1786952978723.
```

Figure 4 Top 5 of the best-rated host teams

4. Merging Twitter data gave an engagement rate for each episode, but no correlation was found between a high rating and a high engagement rate. Fig. 5 shows the correlation between engagement rate and episode rating for each target group, but since no strong correlation existed, engagement rate was not

included as a feature in the model.

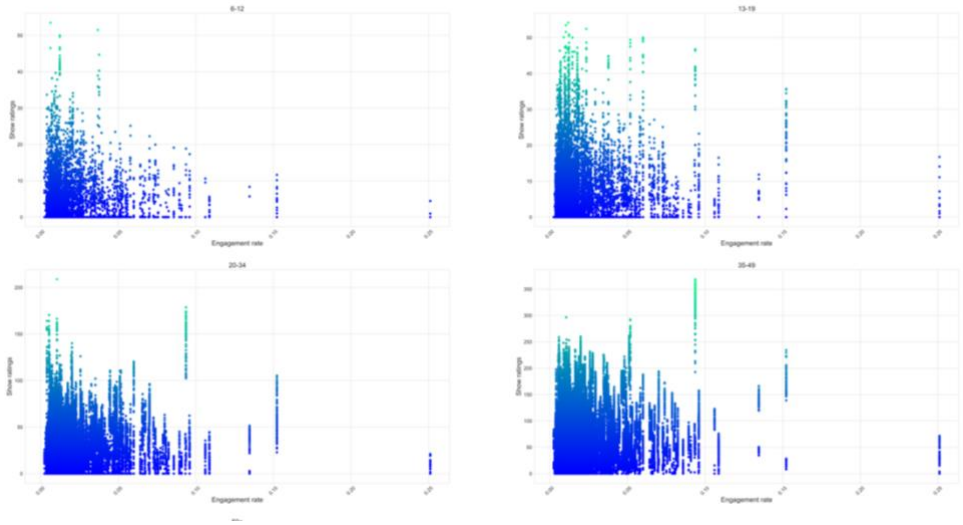


Figure 5 Correlation Between Engagement Rate and Ratings

This exploratory data analysis allowed me not only to clean, organise and prepare data but also made it possible to discover new patterns in data that made my model more accurate.

2 Building Machine Learning Model to Predict Ratings of the TV Shows

The main aim of this project is to create a machine-learning model that will forecast ratings of TV shows, based on the features of the show. There are several types of ML models that can be used for this task, but at first, I had to decide whether I want to use Supervised or Unsupervised learning.

2.1 Supervised vs Unsupervised Learning

The main difference between supervised and unsupervised learning is that Supervised models use labelled data and interact with the Data Scientist who builds the model, meanwhile, Unsupervised Learning uses unlabelled data and tries to find patterns in the data and learn from them without any human interaction, also they do not explain how they give a result or decide. These two types of learning models have different approaches, and they are used for various purposes. Fig. 6 shows the difference between the approaches of these algorithms. Supervised ones specialise in Regression and Classification tasks while Unsupervised algorithms specialise in Clustering and Association problems.

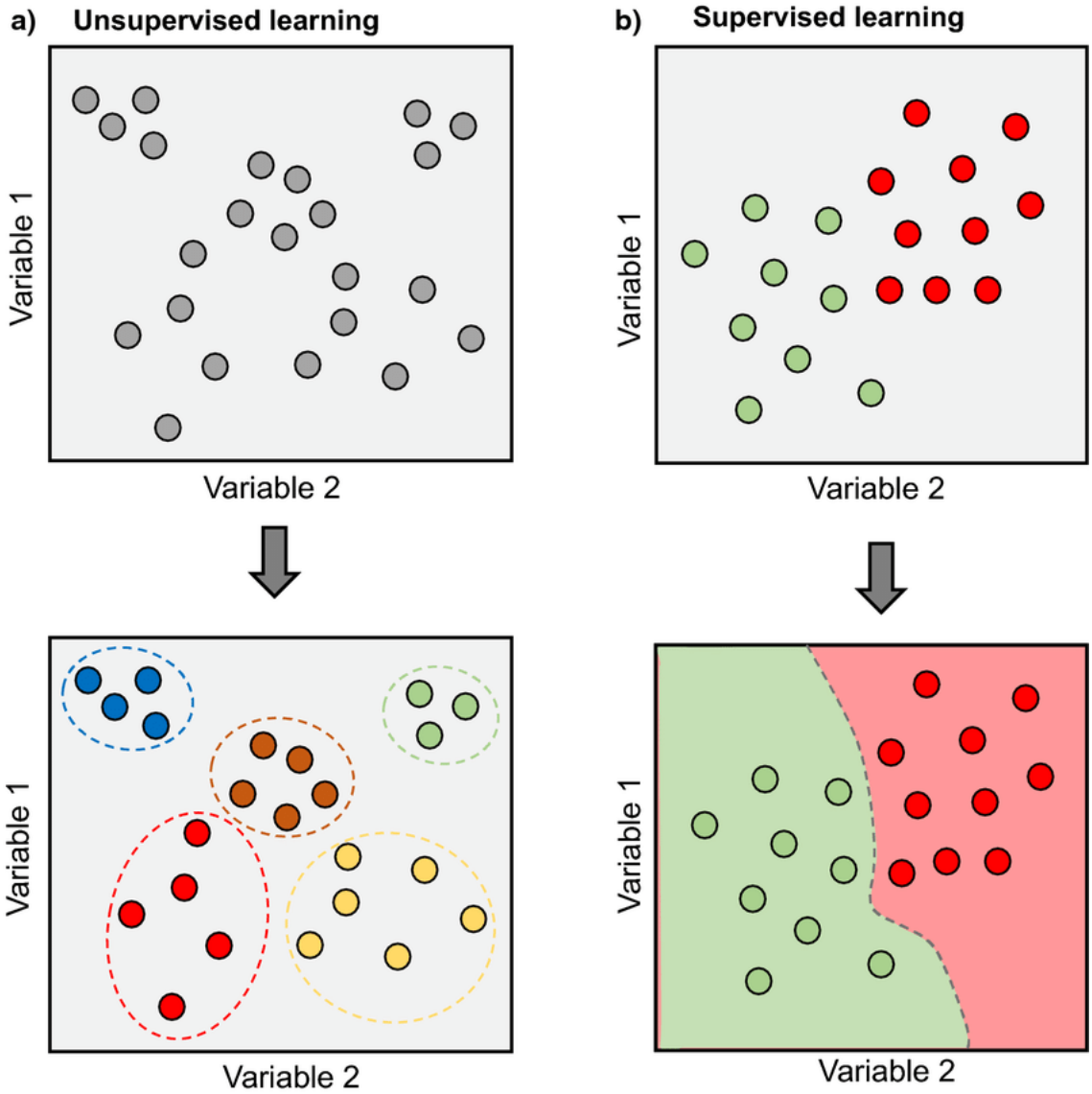
I decided to use Supervised learning due to several reasons:

1. As we are dealing with a continuous variable it would be better to use Regression and Classification instead of Clustering. Such an approach will increase the accuracy of the model.

2. All our data is labelled which perfectly suits Supervised learning algorithms.

3. As my model will be used by a production house, I would like to avoid using a “Black Box” algorithm. The production team will be able to understand what influences the score of the show and how to change it to increase its popularity.

Figure 6 Difference between Supervised and Unsupervised Learning



2.2 Building Machine Learning Models

I chose two models that I will use to predict the rating of the TV shows: Linear Regression and Decision Tree. These two models are different one is numerical and the other one is categorical. I wanted to compare the outputs and find the advantages and disadvantages of each model.

2.2.1 Linear Regression

Linear regression is a linear approach for modelling the relationship between a target variable and one or more features. If there are two or more explanatory variables the model is called multiple linear regression.

For my model, I used 4 features they are: Twitter metrics, the date of the broadcast, the target group, and the hosts of the show. As the target group and host of the show are categorical variables and linear regression only works with the numerical variables, I had to encode this data. To encode this data, I had to create a binary column for each category and assign a 1 or 0 to the column, indicating the presence or absence of the category in the original data.

After encoding all the data, I proceeded with building the model. Models usually do not learn from the data, instead of learning they just recognize the pattern and operate well only with the data they have already seen. To avoid it I split my dataset into training and testing sets and compared key metrics of the model based on the training and test data. Fig. 7 shows the comparison of the metrics

	Training Set	Test Set	Difference
Correlation Coefficient	0.79	0.79	0.0
Mean Absolute Error	82.33	82.03	0.3
Mean Squared Error	26665.07	26546.75	118.32
Median Absolute Error	20.85	20.78	0.07
R2 Score	0.63	0.63	0

Figure 7

I also made a scatter plot to visualise the error in the predicted scores. From Fig. 8 it is clear that data is widely spread but the error in predictions is not greater than 25% of the rating.

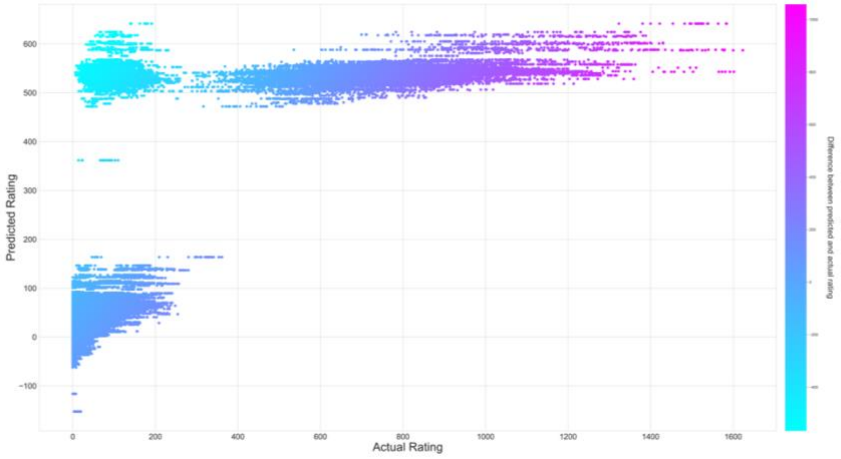


Figure 8 Linear Regression Visualisation

The model performs well on both seen and unseen data and accurately predicts show ratings based on hosts, target group, and broadcast date with an 80% correlation to actual ratings, meeting the set goal. Predictions may have a 20% error margin.

2.2.2 Decision Tree

A decision tree is a flowchart-like structure which represents possible outcomes for a decision-based condition. In our case, it represents a possible rating of the show based on the provided features of the show. To accurately compare the decision tree to the linear regression model I used the same features.

I created several Decision Tree models to achieve the best accuracy. My first model was a tree with a depth of 2. Fig. 9 represents the model I got.

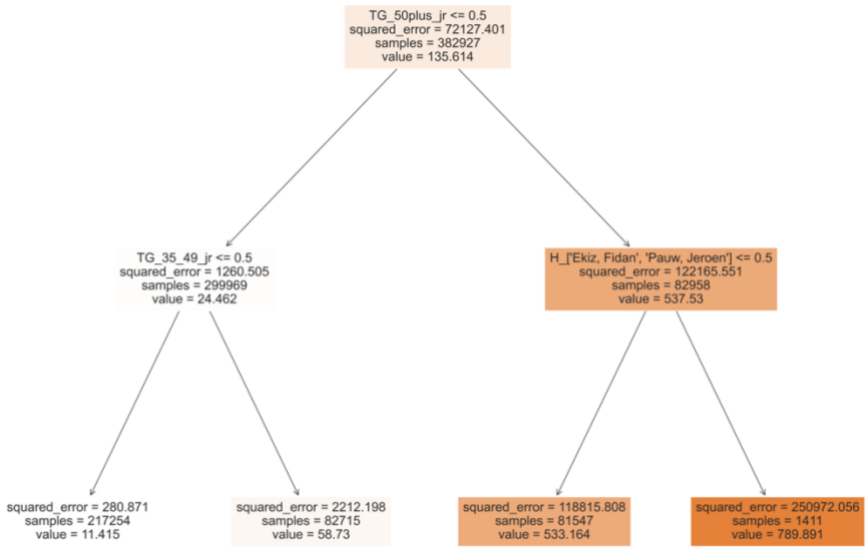


Figure 9 Decision Tree

Each leaf of the tree represents a decision that should be made, the number of samples left at this point of analysis, the current value of the target variable, and an error from the actual average target value. At this point, the decision tree was already as accurate as the linear regression model. I wanted to get the highest accuracy possible, so I used a hyperparameter tunning to get the pest hyperparameters for my decision tree. Fig. 10 shows the implementation of the hyperparameter tuning.

```
from sklearn.model_selection import GridSearchCV

parameters = {"splitter":["best","random"],"max_depth" : [2,4,8,16,32],"min_samples_leaf":[2,4]}

grid_search = GridSearchCV(clf, parameters, cv = 5)


grid_search.fit(X_train_tree,y_train_tree)

print("Best parameters: ", grid_search.best_params_)
print("Best score: ", grid_search.best_score_)
```

```
Best parameters:  {'max_depth': 32, 'min_samples_leaf': 2, 'splitter': 'best'}
Best score:  0.6654371832651973
```

```
tree_model_tunned = DecisionTreeRegressor(splitter="best",max_depth=32,min_samples_leaf=2)
tree_model_tunned.fit(X_train_tree,y_train_tree)

tree_tuned_predicts = tree_model_tunned.predict(X_test_tree)

tree_model_tunned.score(X_test_tree,y_test_tree)
```

```
0.6629375088928714
```

Figure 10 Hyperparameter tunning

After performing the hyperparameter tunning I got the best parameters for my decision tree which are a max depth of 32 and minimal leaf samples of 2. I trained another

model and assessed its performance. Fig. 11 represents a comparison of the two models' performance.

	Decision Tree	Tunned Decision Tree	Difference
Correlation Coefficient	0.79	0.81	0.02
Mean Absolute Error	80.47	75.07	0.3
Mean Squared Error	26785.33	24188.67	2596,66
Median Absolute Error	11.42	11.77	0.07
R2 Score	0.63	0.66	0.03

Figure 11

From the metrics, we can see that tunning indeed improved the performance of the model and made it more accurate than the linear regression model. Unfortunately, after tuning the model it is impossible to visualise it as there are 32 levels of the tree with 2 and more leaves each. In the case of decision trees, you either sacrifice the accuracy or the ease of explanation.

3 Result of Modelling

After a comparison of these two models, I concluded that both have advantages and disadvantages, and they should be used in different use cases:

1. The decision tree model is more accurate than the linear regression one, as the correlation between the features of the show is not completely linear. The decision tree uses both classification and regression, so it is easier to get the desired output from this model because we deal with categorical variables.

2. As I already mentioned with the decision tree you must choose whether you want to sacrifice the accuracy or the ease of the explanation of the model. Meanwhile, linear regression does not have such a problem, no matter how many features you have, it will always be easy to explain why your model gives a certain output and how the data influences it.

3. Linear regression scales better, it can handle new features like a new host team or target group and still provide accurate ratings due to its ability to work with a correlation of all features. Decision trees can have drastic differences in predicted rating as it is can be based primarily on one feature, requiring a new model to be created.

Overall, despite its slightly lower accuracy, I strongly recommend using the linear regression model for forecasting show ratings as it performs better in other aspects.

4 Ethics Behind the Usage of Machine Learning in Ratings Forecast

Ethics is an essential part of every organisation nowadays. There are three main elements of an ethical organizational capacity:

4.1 Ethical Company

An ethical company should be diverse, transparent, and open to feedback. Banijay, with its worldwide offices and diverse team, strives to meet these standards and be inclusive and transparent in its business practices.

4.2 Ethical Process & Tools

Banijay and I are jointly responsible for ethical processes and tools in this project. I followed ethical frameworks and GDPR in my work, which is detailed in the report. Banijay also prioritizes ethics by regularly improving its ethics policy to achieve full ethical organizational capacity.

4.3 Ethical People

I am responsible for ethical behaviour while delivering the final product to Banijay. I used only anonymous or publicly available data and deleted any information that could lead to private data use without consent, in compliance with GDPR.

4.4 Problems in ethical practices and ways to improve it

During my research, I found a lack of diversity at Banijay's Amsterdam office. With mostly Dutch employees, this can lead to biased and one-sided productions. To improve inclusivity, Banijay should hire more diverse minorities in the region. Despite this issue, Banijay is an ethical company with transparent, trustworthy, and diverse practices.

5 Conclusion and Discussion

Recently Machine Learning has become a popular tool to forecast ratings of the show. Models can analyse a large amount of data with a wide variety of factors and find new patterns that can usually are not spotted by humans. This allows broadcasters, advertisers, and producers to improve their decision-making based on the outcome of the model.

The main advantage of the Machine Learning models is that they can deal with the almost infinite amount of data dedicated to any topic and still process, analyse and learn from it in a reasonable amount of time successfully. Meanwhile, analysts have limitations in data they can successfully work with. This obviously places ML models in a favourable position as these models have almost no limitations, can be multi-functional and work fast. Unfortunately, if you want to have such a model you will have to train it on the ideally clean, structured, transformed dataset. Models are sensible to any noise in data so even 10 rows of garbage data can ruin your forecasting model. While building a model, I encountered two issues that negatively impacted performance: Banijay's content data had overlapping shows which follow each other instead and the rating data had many zero values, which should have been NaN values instead. These issues significantly affect the model's accuracy and need to be addressed.

In conclusion, I would like to point out that Machine Learning in Television Rating Prediction is a powerful tool that can change the whole production and broadcasting sphere, but for it to happen there are some limitations that should be solved.

6 Reference List

3.1. Cross-validation: evaluating estimator performance. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/cross_validation.html

3.2. Tuning the hyper-parameters of an estimator. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/grid_search.html

Galarnyk, M. (2022, December 12). *Visualizing Decision Trees with Python (Scikit-learn, Graphviz, Matplotlib)*. Medium. https://towardsdatascience.com/visualizing-decision-trees-with-python-scikit-learn-graphviz-matplotlib-1c50b4aa68dc

Gogoll, J. (2021, April 21). *Ethics in the Software Development Process: from Codes of Conduct to Ethical Deliberation*. SpringerLink. https://link.springer.com/article/10.1007/s13347-021-00451-w?error=cookies_not_supported&code=5d3f0c25-8aab-4c74-809d-888107765b23

Holtz, Y. (n.d.). *Scatterplot with regression line in Matplotlib*. The Python Graph Gallery. https://www.python-graph-gallery.com/scatterplot-with-regression-fit-in-matplotlib/

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *Linear Regression*. SpringerLink. https://link.springer.com/chapter/10.1007/978-1-0716-1418-1_3?error=cookies_not_supported&code=9ec17153-4393-40e8-975a-321ab1490cad

Streefkerk, R. (2022, August 23). *APA format for academic papers and essays*. Scribbr. https://www.scribbr.com/apa-style/format/

Games

Leisure & Events

Tourism

Media

Data Science & AI

Hotel

Logistics

Built Environment

Facility

Mgr. Hopmansstraat 2
4817 JS Breda

P.O. Box 3917
4800 DX Breda
The Netherlands

PHONE
+31 76 533 22 03
E-MAIL
communications@buas.nl
WEBSITE
www.BUas.nl

DISCOVER YOUR WORLD

Breda University
OF APPLIED SCIENCES