

**DATA ANALYTICS FOR BUSINESS DECISION MAKING**

**DATA 1202 - DATA ANALYSIS TOOLS ANALYTICS**

**Project – Report**

**Group 2**

**Section: 3**

**Professor: Sk. Md. Mizanur Rahman, Ph.D.**

**Date: 16-08-2024**

**Submitted by:**

1. Abhinav Mudakuzha Gopal - 100913045
2. Akanksha Nagpal – 100910075
3. Bhavana Dasari – 100965020
4. Gayathri Chinka – 100965099
5. Mohammad Pathari - 100962527

**Table of Contents**

[Information about the Dataset 1](#_Toc174723686)

[Dataset Overview 1](#_Toc174723687)

[Missing values 2](#_Toc174723688)

[Duplicates 2](#_Toc174723689)

[Data types and Unique values 2](#_Toc174723690)

[Handling Missing values 3](#_Toc174723691)

[Data Cleaning 3](#_Toc174723692)

[Data Preprocessing 4](#_Toc174723693)

[Basic Statistics 5](#_Toc174723694)

[Exploratory Data Analysis (EDA) 6](#_Toc174723695)

[MODELS 10](#_Toc174723696)

[Logistic Regression 10](#_Toc174723697)

[Proper split of dataset 11](#_Toc174723698)

[Building the classifier 11](#_Toc174723699)

[Training the classifier 11](#_Toc174723700)

[Testing the classifiers 12](#_Toc174723701)

[Support vector machine 13](#_Toc174723702)

[Proper split of dataset 13](#_Toc174723703)

[Building the classifier 13](#_Toc174723704)

[Training the classifier 13](#_Toc174723705)

[Testing the classifiers 14](#_Toc174723706)

[Random Forest 15](#_Toc174723707)

[Proper split of dataset 15](#_Toc174723708)

[Building the classifier 15](#_Toc174723709)

[Training the classifier 16](#_Toc174723710)

[Testing the classifiers 16](#_Toc174723711)

[Naïve Bayes 17](#_Toc174723712)

[Proper split of dataset 17](#_Toc174723713)

[Building the classifier 17](#_Toc174723714)

[Training the classifier 17](#_Toc174723715)

[Testing the classifiers 18](#_Toc174723716)

[Neural Networks 19](#_Toc174723717)

[Proper split of dataset 19](#_Toc174723718)

[Building the classifier 20](#_Toc174723719)

[Training the classifier 20](#_Toc174723720)

[Testing the classifiers 21](#_Toc174723721)

[Neural Networks using MLP Classifier with 3 hidden layers 22](#_Toc174723722)

[Proper split of dataset 22](#_Toc174723723)

[Building the classifier 22](#_Toc174723724)

[Training the classifier 22](#_Toc174723725)

[Testing the classifiers 23](#_Toc174723726)

[Comparison 24](#_Toc174723728)

[Conclusion 25](#_Toc174723729)

[GitHub Repository link 25](#_Toc174723730)

[References 25](#_Toc174723731)

# Information about the Dataset

## Dataset Overview

**Chosen Dataset:** Netflix

**Source : [Kaggle]** <https://www.kaggle.com/datasets/shivamb/netflix-shows>

**Problem statement:** ***The objective of this project is to create a recommendation engine that accurately determines whether a given piece of content on Netflix is a Movie or a TV show.*** This prediction is based on several factors, such as the duration of the content, the number of seasons (for TV shows), the release year, and the year the content was added to the platform.

**Introduction:**

The Netflix dataset in this study has 8807 entries that include content added from 2008 to 2021. It has 12 characteristics, both categorized (e.g., content kind, director, cast) and numerical (e.g., release date, runtime). The major objective was to predict the content type (movie or TV show) using these features.

We performed significant preprocessing to resolve missing values, with the 'director' and duration' features having around 29.91% and 37.05% missing data, respectively. The dataset was additionally modified by removing the year from the date\_added column and encoding categorical factors such as rating and listed\_in.

**Dataset Composition:**

* **No of rows:** 8807
* **No of columns:** 12

**Feature Description:**

**Categorical variables:**

1. show\_id: Unique id for each row
2. type: Type of content (Movie/TV Show)
3. title: Name of Movie or TV Show
4. director: Director of movie or TV Show
5. cast: Cast of the film or TV Show
6. country: Country from where content was originated
7. date\_added: Date the content was added to Netflix (2012-12-06 and Jan 02, 2021)
8. rating: content rating classification (TV-MA/PG-13 etc.,)
9. duration: Duration of movie or TV show (90 Min, 2 Seasons)
10. listed\_in: Content category (Documentaries/International TV Shows, TV Dramas, TV Mysteries etc.,)
11. description: Summary of content

**Numerical variables:**

1. release\_year: Release year of content

**Target Variable:** We have chosen ‘type’ as the dependent variable. It is encoded with 0 for Movie, 1 for TV Show with a new column name called ‘type\_encoded’.

Missing values

Features such as Director, cast, country, date\_added, rating and duration has missing values.

A screenshot of a computer

Description automatically generated A screenshot of a computer code

Description automatically generated

Duplicates

We have dropped duplicates.

Data types and Unique values

Dataset has object (11) and int64(1) datatypes with following number of unique values.

A screenshot of a computer code

Description automatically generated A screenshot of a computer

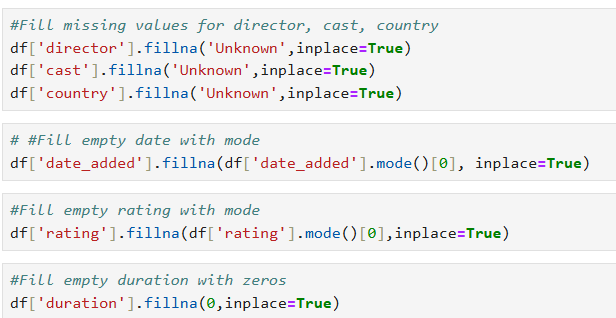
Description automatically generated

Handling Missing values

We resolved missing values by using appropriate placeholders.

* Missing entries in category features such as director, cast, and description were replaced with 'Unknown'.
* Missing numerical feature duration values were replaced with '0'.

In addition, missing values in the date\_added and rating columns was replaced with the corresponding column's mode.



Data Cleaning

We converted the date formats to a uniform Year-Month-Day format. For example, 'August 15, 2021' was changed to '2021-08-15'. Additionally, the data type was changed to 'datetime' to ensure the proper handling of date values.

A screen shot of a computer code

Description automatically generated

Data Preprocessing

To train our model we did some feature engneering on some of the columns.

* Extracted year and month for our analysis and prediction.

A screenshot of a computer program

Description automatically generated

* Removed strings in duration column and make two separate columns for duration and no of seasons in integer.

A screenshot of a computer program

Description automatically generated

* Additionally, We encoded type, rating and listed\_in columns.

A screenshot of a computer program

Description automatically generated

* After dropping encoded values, top 5 rows,

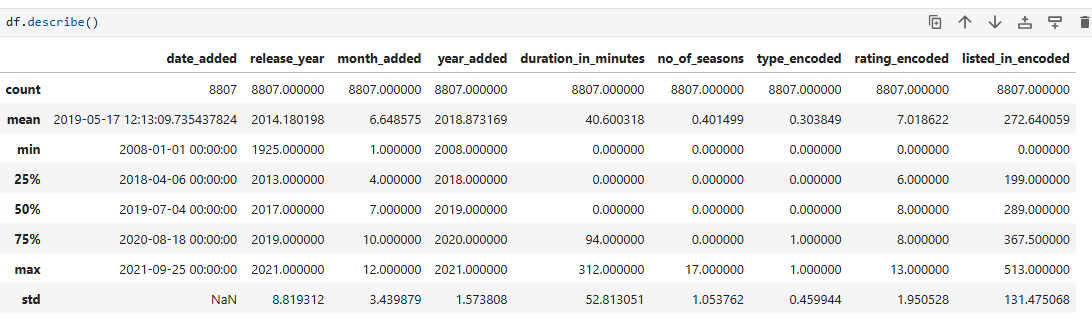
A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Basic Statistics

Understanding the basics statistics helps us to analyze data thoroughly and make informative decisions based on statistical proof. **Mean, median, mode and standard deviation** are descriptive statistics that assist describe and explain the dataset's properties.  
Distribution measures such as **range, quartiles, and IQR** help you understand the spread and variability of the data.



**Insights**

* The earliest entry is in 2008-01-01 and recent was 2021-09-25 according to our dataset
* Most of the movies were added in between 2019 - 2020.
* Movies added were released from 1925 to 2021 latest.
* Movie with highest of duration 312 minutes and seasons with 17.

# Exploratory Data Analysis (EDA)

As part of our EDA, we used data visualization approaches to obtain insights into the information and identify the correlations among variables.

* We used histograms to explore the distribution of year\_added in relation to type\_encoded.
* Pie chart was used to show the proportions of various sorts of material uploaded to the site.
* Pair plots were created to investigate correlations between many numerical variables at once.
* We used heatmaps to demonstrate the numerical features correlation matrix.
* We presented scatter plots of year\_added versus duration\_in\_minutes and release\_year vs rating\_encoded.

A screenshot of a computer screen

Description automatically generated

* **Insights :**

We can see a slow start for Netflix over several years. Things begin to pick up in 2015 and then there is a rapid increase from 2016.It looks like content additions have slowed down in 2020, likely due to the COVID-19 pandemic.

A screen shot of a computer screen

Description automatically generated

* **Insights**

Movies occupy the content with 69.6% and TV shows with 30.4%

A screenshot of a computer screen

Description automatically generated

A group of blue and orange bars

Description automatically generated

**Insights**

* We can see that the release\_year vs. listed\_in\_encoded plots have significant clustering, implying that listed\_in\_encoded is impacted by release year indicating that this feature pair may be useful for classification models.
* KDE charts demonstrate that movies are the most popular content type, with a larger density of blue (indicating movies). This is telling us that movies are more more in the dataset, which might be useful in categorization models.

A screenshot of a computer screen

Description automatically generated

**Insights**

* Most of them have weak negative correlation and weak positive correlation.For example, -0.29 for no\_of\_seasons vs duration\_in\_minutes and 0.32 year\_added vs duration\_in\_minutes
* These observations indicating the most the features are not influencing each other.But we have to further investigate how they are impacting our target variable.

A screenshot of a computer screen

Description automatically generated

**Insights**

* Graph 1 shows there is a weak relation initally between year\_added and duration\_in\_minutes w.r.t to type\_encoded but it went strong as the years are increasing.
* Graph 2 tells that recent content are more listed in ratings rather than early ones(1940-2000)

# MODELS

# Logistic Regression

We used logistic regression since our dependent variable is binary. Its purpose is to describe data and explain the connection between one dependent binary variable and one or more nominal, ordinal independent variables.

A diagram of a hexagon with arrows

Description automatically generated

## Proper split of dataset

A screen shot of a computer code

Description automatically generated

## Building the classifier

A computer screen shot of a logistic regression

Description automatically generated

## Training the classifier

A close up of a sign

Description automatically generated

## Testing the classifiers

A computer screen shot of a computer code

Description automatically generated

A screenshot of a graph

Description automatically generated

**Insights**

* Total Predictions for Movie: 1342. Actual values: 1227
* Total Predictions for TV Show: 420. Actual values: 535
* Precision of 0.97 for TV shows, indicating 97% accuracy in predictions.
* Recall of 0.99 for movies, showing 99% success in identifying actual movies.
* Overall accuracy is 92%. F1-score is 0.95 for movies and 0.85 for TV shows, reflecting slightly better performance with movies.

# Support vector machine

It is a supervised machine learning task in which we attempt to identify a hyperplane that best divides the two classes. Both algorithms seek the optimum hyperplane, but the fundamental distinction is that logistic regression employs a probabilistic approach, whereas support vector machines rely on statistical methods.

A diagram of a technical support

Description automatically generated

## Proper split of dataset

A screen shot of a computer code

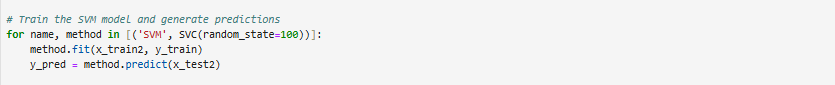
Description automatically generated

## Building the classifier

A computer screen shot of a logistic regression

Description automatically generated

## Training the classifier



## Testing the classifiers

A screen shot of a computer

Description automatically generated

A screenshot of a graph

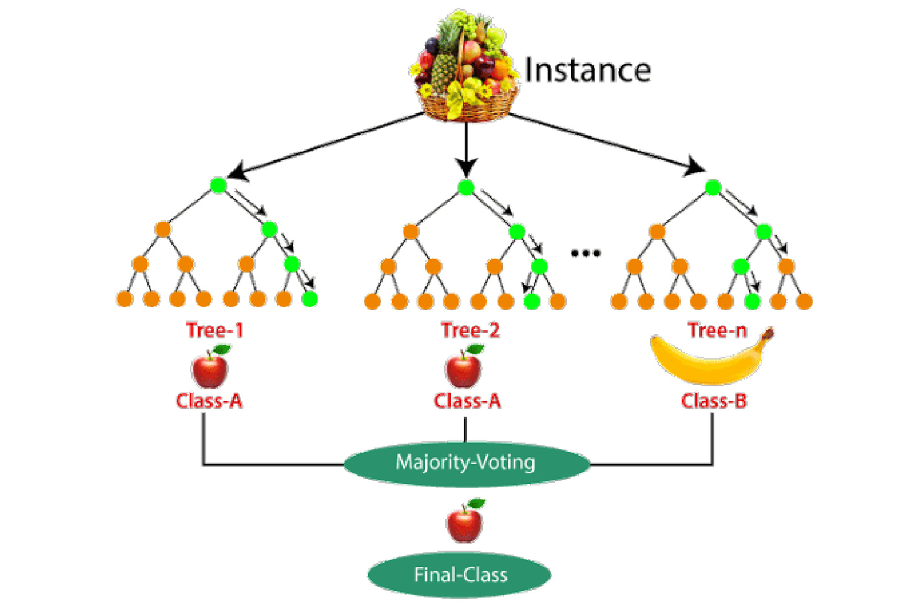
Description automatically generated

**Insights**

* Total Predictions for Movie: 1311. Actual values: 1227
* Total Predictions for TV Show: 451. Actual values: 535
* Precision of 0.95 for TV shows, indicating 95% accuracy in predictions.
* Recall of 0.98 for movies, showing 98% success in identifying actual movies.
* Overall accuracy is 93%. The model performs well with high F1-scores for both movies and TV shows, though it slightly favors movies in recall.

# Random Forest

Random Forest is a popular machine learning technique created by Leo Breiman and Adele Cutler that mixes the outputs of numerous decision trees to produce a single conclusion. Its ease of use and adaptability have boosted its popularity, since it can handle both classification and regression issues.



## Proper split of dataset

A close-up of a computer screen

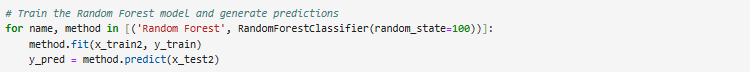
Description automatically generated

## Building the classifier

A computer screen shot of a logistic regression

Description automatically generated

## Training the classifier



## Testing the classifiers

A computer code with many colorful text

Description automatically generated with medium confidence

A screenshot of a graph

Description automatically generated

**Insights**

* Total Predictions for Movie: 1231. Actual values: 1227
* Total Predictions for TV Show: 531. Actual values: 535
* The model has precision and recall of 0.99 for both Movies and TV Shows, indicating excellent accuracy and effectives.
* The model's overall accuracy is 99%, reflecting strong performance in classification.
* With F1-scores of 0.99 for both classes, the model shows consistent and reliable performance across Movies and TV Shows

# Naïve Bayes

We used Naive Bayes Classifier, which can be extremely fast relative to other classification algorithms. It works on Bayes theorem of probability to predict the class of unknown data sets.

A diagram of a mathematical equation

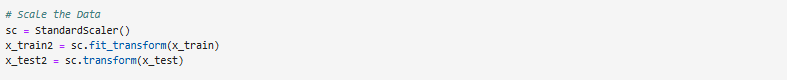
Description automatically generated

## Proper split of dataset

A screenshot of a computer

Description automatically generated

## Building the classifier



## Training the classifier



## Testing the classifiers

A computer screen shot of a computer code

Description automatically generated

A screenshot of a computer

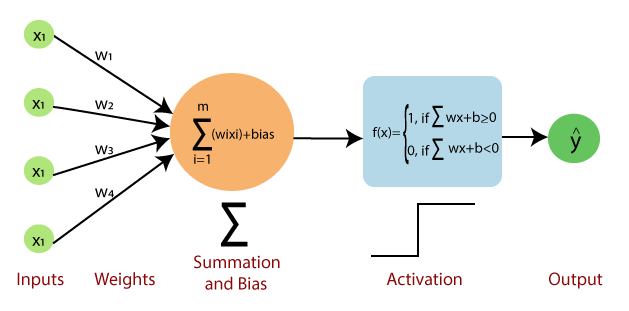
Description automatically generated

**Insights**

* Total Predictions for Movie: 1207. Actual values: 1227
* Total Predictions for TV Show :555. Actual values: 535
* The model shows strong accuracy in predicting movies with a precision of 0.92.
* TV show predictions are less reliable, with a precision of 0.78 and recall of 0.81.
* The model achieves 87% accuracy, performing well overall but with some misclassification.

# Neural Networks

ANN is a network of interconnected input and output units with weighted connections. During the learning phase, the network adjusts weights to predict the proper class from input data.



A diagram of a algorithm

Description automatically generated

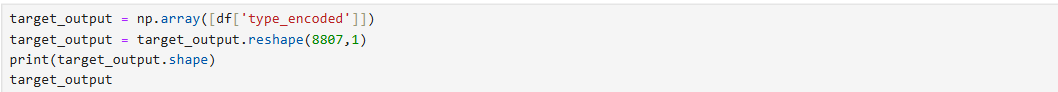
## Proper split of dataset



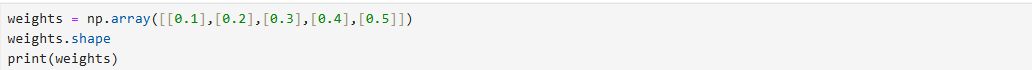
## Building the classifier

A close-up of a computer screen

Description automatically generated



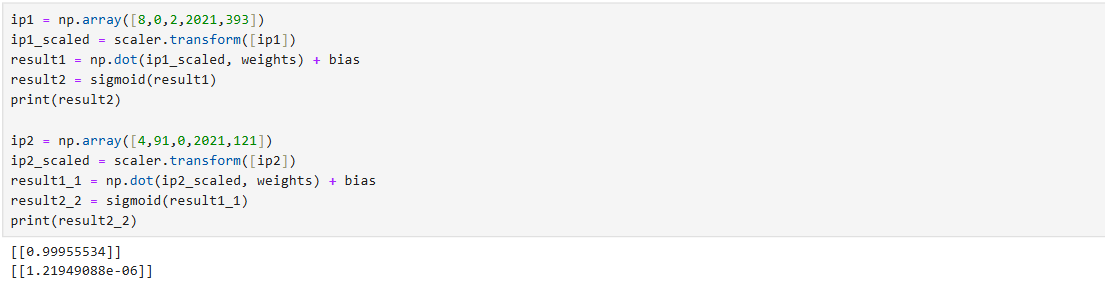
## Training the classifier



A screenshot of a computer program

Description automatically generated

## Testing the classifiers



A screenshot of a computer program

Description automatically generated

A screenshot of a graph

Description automatically generated

**Insights**

* Total Predictions for Movie: 1370. Actual values: 1227
* Total Predictions for TV Show: 392. Actual values: 535
* The model attained a precision of 1.00 for TV programs, indicating that it consistently accurately recognizes TV shows with no false positives.
* The model has a recall of 1.00 for movies, which means it accurately identifies all actual movie occurrences without missing any.
* The recall for TV programs is 0.73, showing that 27% of TV series are incorrectly categorized as movies, implying that the model has difficulty identifying TV shows.
* The model performs well overall, with an accuracy of 0.92; however, its F1-score for TV shows (0.85) is lower than that for movies (0.94), indicating space for development in TV show categorization.

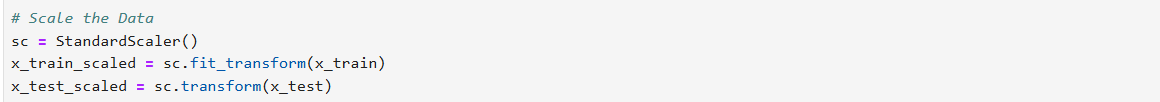
# Neural Networks using MLP Classifier with 3 hidden layers

## Proper split of dataset

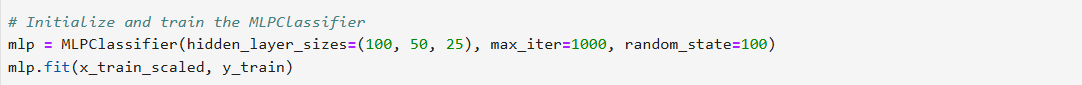
A screen shot of a computer

Description automatically generated

## Building the classifier



## Training the classifier



## Testing the classifiers

## A computer screen shot of text Description automatically generated

A screenshot of a graph

Description automatically generated

**Insights**

* Total Movie Predictions: 1234; Actual Values: 1227
* Total TV Show Predictions: 392. Actual value: 535.
* The precision for both classes is high, indicating that the model is good at minimizing false positives, particularly for class 0.
* Class 0 has a significantly greater recall, indicating that the model is better at detecting all instances of movies than TV shows, with just a few TV series misclassified.
* The model's accuracy is 97%, which means it successfully categorized 97% of the cases in the test set.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Logistic Regression** | **Support vector machine** | **Random Forest** | **Naïve Bayes** | **Neural Networks-single layer** | **Neural Networks-3 layer** |
| Precision | For Movie: 0.90  For TV show:0.97 | For Movie:0.92  For TV show:0.95 | For Movie:0.99  For TV show:0.99 | For Movie:0.92  For TV show:0.78 | For Movie:0.90  For TV show:1.00 | For Movie:0.97  For TV show:0.95 |
| Recall | For Movie:0.99  For TV show:0.76 | For Movie:0.98  For TV show:0.80 | For Movie:1.00  For TV show:0.99 | For Movie:0.90  For TV show:0.81 | For Movie:1.00  For TV show:0.73 | For Movie:0.98  For TV show:0.94 |
| F1-score | For Movie:0.95  For TV show:0.85 | For Movie:0.95  For TV show:0.87 | For Movie:1.00  For TV show:0.99 | For Movie:0.91  For TV show:0.80 | For Movie:0.94  For TV show:0.85 | For Movie:0.98  For TV show:0.94 |
| Accuracy | 0.92 | 0.93 | 0.99 | 0.87 | 0.92 | 0.97 |

# Comparison

**Insights:**

* Precision is good for all the models and random forest it the best. That means the models are predicitng more for movies.
* Recall is also higher for movies in all models. This is indicating that all models are performing better in predicitng acutal Class : movies.
* Random Forest stands out for its nearly flawless precision, recall, F1-scores, and greatest accuracy (0.99). This model is the most accurate for this classification.
* The Neural Networks - 3 Layers model comes in second, with good precision and recall in both classes, and an overall accuracy of 0.97.
* SVM outperforms Logistic Regression with higher accuracy and more balanced precision and recall.
* Naïve Bayes performs poorly with TV shows, resulting in a lower total accuracy (0.87). This suggests that it may not be the optimal solution for this classification, particularly with unbalanced classes.

# Conclusion

Random Forest stands out for its nearly flawless precision, recall, F1-scores, and greatest accuracy (0.99). This model is the most accurate for this classification.

# GitHub Repository link

https://github.com/BhavnaD21/DATA1202

# References

* Lecture notes.
* <https://www.youtube.com/watch?v=IHZwWFHWa-w>
* <https://www.youtube.com/watch?v=aircAruvnKk>
* <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>
* <https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/>
* <https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/>