

SULTAN & DANIYAL SAEED

56189 & 53937

BS/DS-5_1

Documentation

Netflix Movies & TV Shows

Data Mining Project



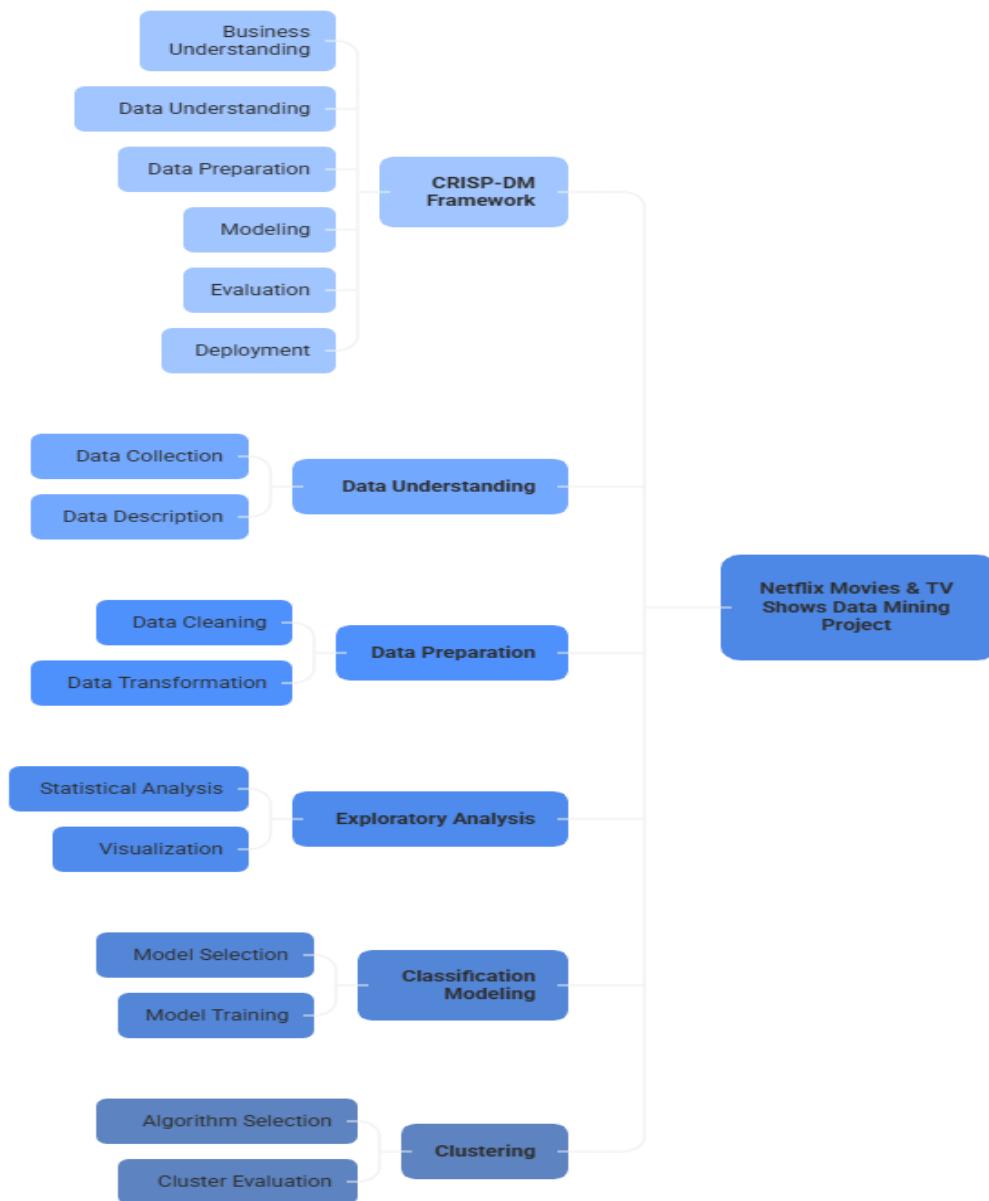
1. Introduction

This document provides complete documentation for the Netflix Movies & TV Shows Data Mining Project. The project follows the CRISP-DM (Cross-Industry Standard Process for Data Mining)

framework and includes data understanding, data preparation, exploratory analysis, classification modeling, clustering, and the development of a recommendation system.

The objective is to extract insights from the Netflix dataset and build machine learning models that support better content discovery and personalization.

Netflix Movies & TV Shows Data Mining Project



2. Dataset Overview

The dataset used in this project is the **Netflix Movies & TV Shows dataset**, publicly available on Kaggle.

It contains metadata for Netflix titles, including:

- Type (Movie or TV Show)
- Title
- Director
- Cast
- Country of origin
- Date added
- Release year
- Rating
- Duration
- Genres (listed_in)
- Description

These attributes support multiple data mining tasks such as classification, clustering, trend analysis, and recommendation modeling.

2. CRISP-DM Workflow

This project is structured according to the CRISP-DM methodology:

- **Business Understanding**

Identify goals such as classifying content type, grouping similar content, and providing recommendations.

- **Data Understanding**

Explore dataset structure, missing values, feature types, and initial visualization of distributions.

- **Data Preparation**

Includes:

- Handling missing values
- Cleaning and transforming duration
- Encoding categorical variables
- Converting textual descriptions using TF-IDF
- Splitting the dataset for training/testing
- **Modeling**

Three types of models were created:

1. Classification Models

Predict whether a title is a Movie or TV Show

using: Logistic Regression Random Forest Classifier Gradient Boosting Classifier

2. Clustering Model

K-Means clustering applied on TF-IDF vectors of descriptions to group similar content.

3. Recommendation Model

A content-based recommender using cosine similarity.

• **Evaluation**

Metrics used for model evaluation:

- Accuracy
- Precision
- Recall
- F1-score
- Silhouette score (for clustering)
- **Deployment**

Trained models (TF-IDF vectorizer and KMeans model) were saved using joblib for future integration into applications.

4. Exploratory Data Analysis (EDA)

The EDA phase focuses on understanding patterns and trends in the dataset.

Key analyses performed:

- **Distribution of content types (Movies vs TV Shows)**
Count plots and pie charts show that Netflix has more Movies than TV Shows.
- **Top countries producing Netflix content**
Using value_counts and bar plots to find leading countries.
- **Release year trends**
A histogram of release years shows how content production changes over time.
- **Genre distribution**
The “listed_in” column is split to analyze the most frequent genres.
- **Missing values analysis**
Missing fields in director, cast, rating, and country were identified and handled.

EDA helps shape the preprocessing and modeling strategies.

5. Data Cleaning and Preparation

Data preparation included:

- **Handling Missing Values**
Columns like director, cast, country, and rating contained missing entries.
These were filled using placeholder values or left as-is depending on relevance.
- **Duration Cleaning**
Movie durations (e.g., “120 min”) were converted into integer minutes. TV
show seasons were handled separately if needed.
- **Encoding Target Variable** type (Movie/TV Show) was converted into numerical form using
LabelEncoder.
- **Feature Engineering**

Categorical columns were prepared for modeling using:

- OneHotEncoder (for nominal features)

- StandardScaler (for numerical features)
- **Train-Test Split**

The dataset was split into training and testing sets to train and evaluate classification models.

6. Classification Models

Three classifiers were trained to predict whether a title is a **Movie or TV Show**:

1. Logistic Regression

Simple and interpretable linear model.

2. Random Forest Classifier

Ensemble of decision trees, robust and effective.

3. Gradient Boosting Classifier

Boosting-based model that often provides high accuracy.

Evaluation Metrics

Each model was evaluated using:

- Accuracy
- Precision
- Recall
- F1 Score
- Confusion Matrix
- Classification Report

The model with the highest **F1 score** was selected as the best model.

7. Clustering Analysis

Clustering was performed using **K-Means** on TF-IDF vectors derived from textual descriptions.

Steps:

1. Convert descriptions into numerical vectors using **TF-IDF**.
2. Reduce dimensionality using **TruncatedSVD** for visualization.
3. Apply **K-Means** clustering.
4. Interpret clusters by analyzing top terms from each cluster.

Evaluation

Clustering was evaluated using **Silhouette Score**, which measures how well data points fit into clusters.

8. Recommendation System

A **content-based recommendation system** was implemented.

How it works:

1. Convert descriptions into TF-IDF vectors.
2. Compute **cosine similarity** between all pairs of titles.
3. When a user inputs a title, return the top N most similar titles.

This allows Netflix-like personalized recommendations based solely on content similarity.

9. Model Saving (Deployment)

To prepare for deployment, the following models were saved using

```
joblib: joblib.dump(tfidf_model, "tfidf.pkl") joblib.dump(kmeans_model,  
"kmeans.pkl")
```

These files can later be loaded into:

- Web apps
- APIs
- Recommendation engines
- Dashboards

This step simulates the deployment phase of CRISP-DM.

10. Conclusion

This project demonstrates how data mining techniques can extract meaningful insights from entertainment data. Using:

- EDA
- Classification
- Clustering
- Recommendation modeling

it becomes possible to enhance user experiences on platforms like Netflix by improving search, categorization, and personalized recommendations.

The project provides a complete end-to-end machine learning workflow and shows how modern data mining techniques support real-world applications.

Data Mining Workflow for Entertainment Data

