**KOREAN-DRAMA DATASET**

**MIDTERM PROJECT REPORT**

Submitted By:

1. Poojitha Rao Gadela.
2. Nawaz Shareef Shaik.

**TABLE OF CONTENTS**

1. **Introduction to the Dataset.**
2. **Data Cleaning & Preprocessing.**
3. **Exploratory Data Analysis (EDA) & Key Insights.**
4. **Visualization.**
5. **Feature Engineering.**
6. **Model Building & Evaluation.**
   1. Linear Regression.
   2. Logistic Regression.
   3. Decision Tree.
   4. Gradient Boosting Classifier.
7. **Conclusion**
8. **Recommendations for Future Work**
9. **References**
10. **Group Members and Contributions**

**1. Introduction to the Dataset.**

This project aims to analyze the Korean dramas (K-Dramas) dataset to identify patterns that influence viewer ratings and to build predictive models that can classify whether a show is highly rated. The dataset includes structured attributes like Title, Year of release, Number of Episodes, and Rating, as well as unstructured attributes such as Genre, Tags, and Actors. The analysis leverages popular Python libraries, including pandas for data manipulation, matplotlib and seaborn for visualization, and scikit-learn for machine learning.

## [Korean Dramas Dataset](https://www.kaggle.com/datasets/saikalbatyrbekova/korean-dramas-dataset-eda)

1. **Data Cleaning & Preprocessing.**

Data cleaning was a crucial first step due to missing and inconsistent values. We dropped rows where any of the key numeric columns (Rating, Number of Episodes, Year of release) were null. This ensured a robust input to our models without the risk of NaN-related computation issues.

A binary target variable, High Rating, was also engineered to represent whether a show was rated 9.0 or higher. This variable is central to our classification models. Categorical fields, such as Genre and Tags, were acknowledged but not used directly in modeling due to the complexity in encoding. Future work could transform these using NLP or multi-hot encoding techniques.

We also standardized numerical values using StandardScaler when appropriate, particularly for models sensitive to feature scaling.

1. **Exploratory Data Analysis (EDA).**

Exploratory Data Analysis (EDA) is a crucial step in the data science workflow that helps uncover patterns, detect anomalies, and form hypotheses using statistical graphics and data visualization techniques. In this project, EDA was used to derive insights from the K-Drama dataset and guide feature selection for modeling.

**Objectives of EDA**

* Understand data distribution and summary statistics
* Identify outliers and anomalies
* Detect relationships between features
* Generate hypotheses for model development

1. **Visualization.**

**1. Histograms**

Used to visualize the distribution of numerical variables such as ratings and a number of episodes.

* Purpose: Identify skewness, modality, and central tendency.
* Findings: The rating distribution was skewed toward higher values, with most shows rated above 8.0, indicating potential selection bias.

**2. Boxplots**

Used to compare distributions of ratings across different categorical groupings (e.g., episode count).

* Purpose: Detect outliers and compare median ratings.
* Findings: Dramas with 16 episodes showed a tighter rating range with a generally higher median rating.

A graph of a number of episodes

AI-generated content may be incorrect.

**3. Scatter Plots**

Used to explore relationships between release year and ratings.

* Purpose: Identify trends, clusters, and potential non-linear relationships.
* Findings: A cluster of highly rated dramas appeared in more recent years, suggesting either quality improvement or recency bias.

A chart with different colored dots

AI-generated content may be incorrect.

**4. Correlation Heatmap**

Although not used extensively in this project due to limited numeric features, these can reveal pairwise correlations between features.

* Potential Use: Could guide feature selection for multivariate models.

A screenshot of a computer screen

AI-generated content may be incorrect.

**Theoretical Justification for Visualizations**

* Histograms are grounded in descriptive statistics and offer a clear view of frequency distributions.
* Boxplots use quartile-based visual summaries to emphasize spread and outliers.
* Scatter plots visually capture the nature of bivariate relationships, critical for understanding linearity or clustering in feature space.
* Heatmaps with correlation matrices help in detecting multicollinearity and feature redundancy.

**Tools and Libraries**

* Pandas: For data manipulation and preprocessing
* Matplotlib & Seaborn: For creating high-level statistical plots
  + Seaborn provides convenient functions like sns.boxplot, sns.histplot, and sns.scatterplot for aesthetic and informative visualizations.

1. **FEATURE ENGINEERING.**

Feature engineering in this project involved cleaning and transforming key data points to enhance model performance. Rows with missing values in **Rating**, **Year of release**, and **Number of Episodes** were removed. A binary target column, **High Rating**, was created to classify shows rated 9.0 or higher. Numerical features were scaled using **StandardScaler** to prepare for models sensitive to feature magnitudes. Although categorical fields like **Genre** and **Tags** were not used in this phase, they are recommended for future work using NLP or encoding techniques.

| **Feature** | **Type** | **Transformation** | **Used in Model** |
| --- | --- | --- | --- |
| Year of release | Numeric | StandardScaler | Yes |
| Number of Episodes | Numeric | StandardScaler | Yes |
| Rating | Numeric | Used to derive the target only | NA |
| High Rating (Target) | Binary (Derived) | Rating ≥ 9.0 | Yes |
| Genre / Tags / Actors | Categorical/Text | Not encoded (future scope) | NA |

1. **Model Building & Evaluation**

This section outlines the implementation and evaluation of four machine learning models applied to predict the rating category of K-Dramas. The goal was to classify shows as **Highly Rated (≥9.0)** or not, based on structured attributes.

**6.1 Linear Regression**

Though primarily used for regression tasks, Linear Regression was initially tested to explore the relationship between features like Year of Release and Number of Episodes with the actual Rating.

**Input Features**: Year of Release, Number of Episodes

**Target**: Rating (continuous)

**Evaluation Metric**: Mean Squared Error (MSE):0.1444, MAE: 0.259, MASE: 1.100

**Summary**:  
Linear Regression helped establish that the relationship between features and ratings wasn't purely linear, guiding the shift to classification models.

**6.2 Logistic Regression**

Logistic Regression was employed for binary classification using the engineered High Rating target variable.

**Input Features**: Year of Release, Number of Episodes

**Target**: High Rating (1 if rating ≥ 9.0)

**Preprocessing**: StandardScaler applied to input features

**Evaluation Metrics**:

1. Accuracy: ~30%

2. Precision: 0.058, Recall: 1.000, and F1-Score: 0.109

**Remarks**: Performed well due to its simplicity and interpretability. Balanced parameter (class weight='balanced') improved performance on imbalanced data.

**6.3 Decision Tree Classifier**

The Decision Tree model was tested for its ability to handle non-linear data and provide clear decision rules.

**Input Features**: Same as above

**Target**: High Rating

**Parameters**: max\_depth and Random-state tuned for best performance

**Evaluation Metrics**:

Accuracy: - 96%

**Summary**:  
Good interpretability but risk of overfitting on small datasets. Useful as a baseline for tree-based models.

**6.4 Gradient Boosting Classifier**

This ensemble model was chosen for its robustness and superior handling of complex, non-linear relationships.

**Input Features**: Year of Release, Number of Episodes

**Target**: High Rating

**Libraries Used**: Gradient Boosting Classifier from sklearn.ensemble

**Evaluation Metrics**:

Accuracy: ~80%

**Summary**:  
Offered high performance without overfitting. Slightly more computationally intensive but ideal for high-accuracy scenarios.

**Comparison Summary**

| **Model** | **Accuracy** | **Best Use Case** |
| --- | --- | --- |
| Linear Regression | ~12% | Exploring continuous Rating trends |
| Logistic Regression | ~30% | Fast, interpretable predictions |
| Decision Tree | ~96% | Rule-based decisions, interpretable |
| Gradient Boosting | ~80% | High accuracy, handles complexity |

1. **Conclusion**

From this analysis, we found that simple structural metadata like release year and episode count can offer strong predictive signals for show ratings. While Gradient Boosting provided powerful, accurate predictions, Logistic Regression proved equally viable for real-world deployment due to its speed and interpretability.

The dataset's bias toward highly rated shows suggests the need for more balanced data in future evaluations. Genre, cast popularity, and plot descriptions are likely untapped areas for improving model performance.

1. **Recommendations for Future Work**

* **Incorporate NLP:** scope to analyze Tags, Description, and Genre using embeddings or TF-IDF to extract content-based features.
* **Expand Feature Set:** Can include external data like actor social media influence, genre popularity, or viewer demographics.
* **Use Time-Based Splits:** Instead of random splits, use chronological train/test splits to simulate real-world predictions.
* **Deploy Clustering Models:** In the future, there is scope to apply GMM, DBSCAN, or hierarchical clustering to identify K-Drama archetypes or user segmentation.
* **Explore Model Stacking:** we can use ensemble techniques like stacking Gradient Boosting with Logistic Regression or SVM for improved predictions.

1. **Reference**

a)Netflix K-Drama Dataset (from Kaggle): # [Korean Dramas Dataset](https://www.kaggle.com/datasets/saikalbatyrbekova/korean-dramas-dataset-eda)

b) Python libraries: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn

1. **Group Members and Contributions**

|  |  |
| --- | --- |
| **Names** | **Points** |
| Poojitha Rao Gadela | 10 |
| Nawaz Shareef Shaik | 08 |