

# BİL470

## PROJECT STATUS REPORT

Kaan Baykal 221101028  
Beril Aydın 231101002  
Anıl Özişler 211101081

## 1. Abstract

We are developing a machine learning system to predict IMDb ratings of movies/TV shows using metadata features such as genre, runtime, certificate, description, cast and directors. Data collection from Kaggle is complete; a thorough EDA has been conducted; and three modeling approaches have been selected (XGBoost, Random Forest, Ridge Regression). The next milestone is to finalize preprocessing pipelines, run 5-fold cross-validation with consistent splits, and report comparative metrics using RMSE as the primary measure (with MAE and  $R^2$  as secondary).

## 2. State of data collection

- **Source:** Kaggle — *TV and Movie Metadata with Genres and Ratings (IMDB)*  
URL: <https://www.kaggle.com/datasets/gayul4/tv-and-movie-metadata-with-genres-and-ratings-imdb>
- **Files:** Main CSV (UTF-8). Shape: [129,891 rows] × [9 columns]; size [~40 MB].
- **Integrity:** Duplicates detected: 0
- **Licensing/Use:** As stated on Kaggle, used for academic purposes in this course.

## 4. EDA(Exploratory Data Analysis)

Github Link For EDA Notebook: [Berilay6/Movie\\_IMDb\\_Prediction](#) (eda.ipynb)

### Key EDA Observations

- **Data Cleaning:**  
The dataset initially had major missing values (especially in *certificate*), and runtime/votes were text. After cleaning and type conversions, **61,227 valid rows** remained.
- **Target Variable (rating):**  
Ratings follow an almost normal distribution centered around **6.19** with standard deviation of **1.34**, meaning most films are rated between 5–7.
- **Numerical Features:**
  - **Runtime:** Average ≈84 minutes; films longer than 200 min are only **0.58%** (outliers).
  - **Votes:** Highly **right-skewed**—a few films get millions of votes; log transform improves visualization.

- **Categorical Features:**
  - **Top genres:** Action, Drama, and Comedy are dominant.
  - **Top directors:** A small group directs the majority of films.
  - **Genre impact:** *Documentary*, *Short*, and *Biography* have the highest ratings, while *Horror* and *Thriller* score lower.
- **Text Feature:**

Word cloud shows common description themes like *love*, *life*, *world*, *family*, *war*, suggesting narrative diversity.
- **Relationships with Rating:**
  - **Votes vs. Rating:** Slightly positive trend (popular films often have higher ratings), but **weak correlation** (+0.07).
  - **Runtime vs. Rating:** Very short movies (<60 min) tend to have lower ratings; overall **weak linear link** (−0.07).
- **Overall Insight:**

The dataset is clean and reliable; genres appear as the strongest categorical predictor, while numerical features show weak linear correlation—nonlinear or interaction models will likely perform better.

## 5. Preprocessing Decisions

This section is detailed with the information that was obtained by the previously made EDA(Exploratory Data Analysis). The primary goal is to convert the raw, mixed-type dataset into a fully numerical, clean, and structured format suitable for machine learning, while rigorously preventing data leakage.

1. **Handling of Missing Data (NaN Values):** The certificate column is noticed to have a great amount of missing values, with approximately %80 of its values being null. In contrast, the other columns were observed to have a manageable amount of null values.

**Decision:** Certificate column will be dropped and not be used as a feature in the model to prevent incorrect learning, introducing noise and systematic bias and risk of data corruption. For the other features, null values will be removed using listwise deletion.

2. **Data Type Conversion and Cleaning:** Runtime and votes columns were incorrectly loaded as object(string) types. This was due to non-numeric characters (“min”, “,”).

**Decision:** A cleaning function will be applied to clean data from non-numeric characters. Subsequently, the column will be converted to a numeric type. This conversion is mandatory for the models to interpret these features as continuous numerical variables.

- 3. Categorical Feature Encoding:** The dataset contains three distinct types of categorical features: genre(multi-label), director/stars(high-cardinality) and type(low-cardinality).

**Decision:** The genre column will be processed using Multi-hot Encoding. This strategy correctly represents the many-to-many relationship where one film can belong to multiple genres simultaneously. For director/star, the Top-K One-Hot Encoding strategy will be applied. Only the Top-K most frequent creators will be retained as unique features, while all others are grouped into a single "Other" category. This avoids the curse of dimensionality that would result from encoding thousands of unique names.

- 4. Numerical Feature Transformation and Scaling:** The distribution of the votes column was found to be highly right-skewed, and the runtime column was observed to contain significant outliers.

**Decision:** A Logarithmic Transformation will be applied to the votes column to normalize its distribution, make it symmetrical and reduce the disproportionate influence of extreme outlier values. All numerical features will be scaled before being fed into the Ridge Regression model. This is critical as L2 regularization is sensitive to feature scales; scaling ensures all features are centered around zero with unit variance, allowing the model to optimize its coefficients fairly.

- 5. Text Feature Vectorization:** The description column contains rich, unstructured text. The WordCloud analysis confirmed the presence of descriptive and thematic words that are relevant to a film's content.

**Decision:** Description column text will be converted into a numerical feature space using TF-IDF (Term Frequency-Inverse Document Frequency) Vectorization. This method is chosen over simpler word counts because it amplifies the importance of words that are descriptive of a specific movie (high IDF) while diminishing the weight of common, non-predictive words (low IDF).

## 6. Decided models

Decided Model	Reason
XGBoost	We select XGBoost because gradient boosting excels at capturing complex non-linear relationships and rich feature interactions that are common in mixed tabular+text problems. Its strong built-in regularization (L1/L2), shrinkage, and column/row subsampling help control overfitting while maintaining high accuracy. XGBoost is also efficient with sparse, high-dimensional inputs (e.g., TF-IDF features), making it a strong candidate for top performance.
Random Forest	We include Random Forest as a robust, low-sensitivity ensemble baseline. Bagging of decision trees provides stability against noise and outliers, typically requiring less delicate hyperparameter tuning than boosting methods. It also supports permutation feature importance for interpretability, offering a reliable benchmark to compare with more aggressive learners on non-linear patterns.
Ridge Regression	We use Ridge as a fast, regularized linear reference. The L2 penalty guards against overfitting in high-dimensional spaces (after encoding text and categorical variables), while producing interpretable coefficients that reveal global trends. This model serves as a sanity check to quantify how much variance can be explained without complex interactions

## 7. Evaluation Methods

We evaluate all models using **5-Fold Cross-Validation** with shared, fixed splits across models to ensure a fair, variance-reduced estimate of generalization performance. In every fold, all preprocessing steps—including imputation, categorical encoding, TF-IDF (and SVD where applicable), and scaling for linear models—are performed inside a single pipeline and fit only on the training portion, then applied to the validation portion, which prevents data leakage. We report **RMSE** as the primary metric, and **MAE** and **R<sup>2</sup>** as secondary metrics, providing per-fold scores and mean  $\pm$  standard deviation. For boosted trees (e.g., XGBoost), we use early stopping within each training fold by holding out a small internal validation split from the fold's training data to determine the optimal number of boosting rounds.

## 8. Brief Planned Methodology

- **Data Intake & Profiling:** Load the Kaggle CSV into a reproducible notebook; record shape, schema, types, missingness, correlations and duplicates. Preserve raw files in data/raw/ and document a data dictionary.
- **Preprocessing:** Convert types (e.g., runtime to int), drop exact duplicates, and eliminate the rows with null values except the certificate (we will not use the certificate column). Represent genre as multi-hot; encode type with one-hot; use top-K one-hot for high-cardinality director/star. Vectorize description with TF-IDF; for Random Forest and Ridge, reduce TF-IDF using Truncated SVD (~300 components).
- **Modeling:** Train three approaches on identical folds and feature space:
  1. **XGBoost Regressor** on sparse TF-IDF + encoded categoricals + numeric features.
  2. **Random Forest Regressor** on SVD-reduced TF-IDF + encoded categoricals + numeric features.
  3. **Ridge Regression (L2)** on SVD-reduced TF-IDF + one-hot categoricals + scaled numerics.
- **Hyperparameter Tuning:** Use small, well-bounded grids or randomized search per model. For boosted trees, apply early stopping using an internal validation split within each training fold.
- **Evaluation Protocol:** Use **5-Fold Cross-Validation** with shared, fixed splits. Primary metric: **RMSE**; secondary: **MAE** and **R<sup>2</sup>**. Report per-fold scores and mean  $\pm$  std. Prevent leakage by fitting all preprocessing only on training folds.
- **Error & Explainability:** Analyze residuals (error histograms, QQ plots), compare performance by strata (e.g., type, certificate, high/low votes), and inspect feature importance (XGBoost gain, RF permutation importance, Ridge coefficients).