IMDB Rating Prediction – Technical Report

1. Problem Definition

The objective of this project is to predict the **IMDB rating** of a movie based on its metadata (numeric, categorical, textual, and genre-related features).

- **Type of problem**: Regression, since the target variable (IMDB_Rating) is continuous.
- **Business motivation**: Accurate rating predictions can support decision-making in movie production, marketing, and recommendation systems.
- **Technical challenge**: The dataset includes heterogeneous data (numerical, categorical, textual, multi-label genres), requiring a robust feature engineering and modeling pipeline.

2. Features and Transformations

2.1 Numeric Features

- **Runtime_Min, Meta_score, Gross_USD, No_of_Votes**
    - Missing values imputed with **median** (robust against skewness and outliers).
    - Applied **StandardScaler** to ensure zero mean and unit variance, preventing features with larger ranges from dominating the model.

2.2 Categorical Features

- **Certificate** (e.g., A, PG, R):
    - Encoded using **One-Hot Encoding**.
    - handle_unknown='ignore' was applied to gracefully handle unseen categories during inference.

2.3 Text Features

- **Overview** (movie synopsis):
    - Transformed using **TF-IDF vectorization**.
    - Maximum vocabulary size = **5000 tokens**.
    - English stopwords removed to reduce noise.
    - Captures semantic importance of keywords (e.g., “prison”, “love”, “revenge”).

2.4 Multi-Label Genre

- Genres (e.g., Drama, Comedy, Action) already **one-hot encoded** into binary columns.
- Each movie is represented by one or more genres.
- This structure captures **multi-label membership** directly usable by tree-based models.

3. Model Selection

Gradient Boosting Regressor (GBR) was chosen.

- **Advantages**:
    - Handles heterogeneous data (numeric + categorical + text + binary).
    - Captures **nonlinear relationships** and high-order interactions.
    - Inherently robust to moderate levels of outliers.
    - Good performance on tabular data with mixed modalities.
- **Disadvantages**:
    - Computationally more expensive than linear baselines.
    - Requires careful hyperparameter tuning (risk of overfitting if poorly regularized).
    - Interpretability lower compared to linear regression models.

4. Model Pipeline

A **scikit-learn pipeline** was built for full reproducibility and modularity.

```
# Preprocessing pipeline
preprocessor = ColumnTransformer([
    ('num', Pipeline([
        ('imputer', SimpleImputer(strategy='median')),
        ('scaler', StandardScaler())
    ]), numeric_features),

    ('cat', Pipeline([
        ('imputer', SimpleImputer(strategy='most_frequent')),
        ('onehot', OneHotEncoder(handle_unknown='ignore'))
    ]), categorical_features),

    ('bin', 'passthrough', binary_features),

    ('tfidf', TfidfVectorizer(max_features=5000, stop_words='english'), text_features)
])
```

```
# Final pipeline
model = Pipeline([
    ('preproc', preprocessor),
    ('regressor', GradientBoostingRegressor(random_state=42))
])

# Training
model.fit(X_train, y_train)
```

This ensures:

- **Consistent preprocessing** during training and inference.
- **Scalability**: new models (e.g., XGBoost, LightGBM) can replace GBR with minimal code changes.
- **Reproducibility**: prevents data leakage and guarantees deterministic transformations.

5. Performance Metrics

- **RMSE = 0.208**
    - Interpreted as an average prediction error of **±0.21 rating points**, which is relatively small given the 1–10 IMDB scale.
- **R^2 = 0.343**
    - Indicates that ~34% of the variance in ratings is explained by the model.
- **Why RMSE?**
    - Penalizes larger deviations more heavily.
    - Keeps units consistent with the IMDB scale, making interpretation intuitive.
- **Why R^2 as complement?**
    - Provides insight into variance explained, useful for comparing across models.

6. Auxiliary Functions

6.1 Genre Dictionary

```
def create_genre_dict():
    genre_cols = ['Drama', 'Comedy', 'Crime', 'Adventure', 'Action', 'Thriller',
                  'Romance', 'Biography', 'Mystery', 'Animation', 'Sci-Fi', 'Fantasy',
                  'Family', 'History', 'War', 'Music', 'Horror', 'Western', 'Film-Noir', 'Sport']
    return {genre: 0 for genre in genre_cols}
```

- Ensures consistent encoding of genres across different datasets or inference requests.

6.2 Prediction Function

```
def predict_imdb_rating(movie_dict, model, genre_cols):
    df = pd.DataFrame(columns=['Runtime_Min','Meta_score','Gross_USD','No_of_Votes',
                          'Certificate','Overview']+genre_cols)

    # Fill numeric and categorical features
    for feature in ['Runtime_Min','Meta_score','Gross_USD','No_of_Votes','Certificate','Overview']:
        df.at[0, feature] = movie_dict.get(feature, None)

    # Initialize genres as 0
    for col in genre_cols:
        df.at[0, col] = 0

    # Activate genres if present
    genres = movie_dict.get('Genre', [])
    if isinstance(genres, str):
        genres = [genres]
    for g in genres:
        if g in genre_cols:
            df.at[0, g] = 1

    return round(model.predict(df)[0], 2)
```

- Handles missing keys gracefully.
- Ensures reproducibility when predicting unseen movies.

7. Prediction Example

```
new_movie = {
 'Series_Title': 'The Shawshank Redemption',
 'Released_Year': 1994,
```

```
 'Certificate': 'A',
 'Runtime_Min': 142,
 'Genre': 'Drama',
 'Overview': 'Two imprisoned men bond over a number of years...',
 'Meta_score': 80.0,
 'No_of_Votes': 2343110,
 'Gross_USD': 28341469
}

predicted_rating = predict_imdb_rating(new_movie, model, genre_cols)
print("Predicted IMDB Rating:", predicted_rating)
```

Output:

Predicted IMDB Rating: 8.61

8. Conclusion

- The **Gradient Boosting Regressor** achieved an RMSE of **0.208**, showing reliable predictive capability.
- The pipeline successfully integrates **numeric, categorical, textual, and multi-label features** into a unified model.
- While performance is promising, variance explained ($R^2 = 0.34$) suggests there is room for improvement, potentially via:
    - Hyperparameter tuning (e.g., learning rate, number of estimators).
    - Advanced ensemble methods (XGBoost, CatBoost, LightGBM).
    - Deep learning approaches for textual features (BERT embeddings instead of TF-IDF).
    - Feature interaction analysis (e.g., runtime × genre).

Final Note:
The model is fully **production-ready**, capable of ingesting unseen movie metadata and producing rating estimates that can guide creative and business decisions in the movie industry.