Analysis Report

Detected Libraries & Frameworks

- Data handling & math: numpy, pandas
- Visualization: matplotlib, seaborn
- Preprocessing: StandardScaler, PCA, train_test_split
- Modeling: LinearRegression (scikit-learn)
- Evaluation: mean_squared_error (MSE), r2_score (R^2)
- Deep Learning reference: keras/tensorflow (import detected, but not clearly used in training snippet)

Data-related Findings

- The dataset may be loaded via a different method (e.g., load_dataset()) or inside a custom function.
- Target variable is y_ratings (movie ratings), indicating this is a regression task.

Preprocessing Steps

- Scaling — StandardScaler.fit_transform used to normalize feature values.
- Dimensionality reduction — PCA applied to reduce dimensionality before modeling.
- Train-test split — train_test_split(..., random_state=42) ensures reproducibility.

Models & Training

The primary model used is Linear Regression from scikit-learn.

Example training snippet:

```
X_train, X_test, y_train, y_test = train_test_split(
    X_pca, y_ratings, test_size=0.2, random_state=42
)

model = LinearRegression()
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)

print('MSE:', mean_squared_error(y_test, y_pred))
print('R²:', r2_score(y_test, y_pred))
```

🔹 Idea of the model: PCA reduces correlated features into orthogonal components. Linear Regression then learns a weighted combination of these components to predict movie ratings.

Evaluation

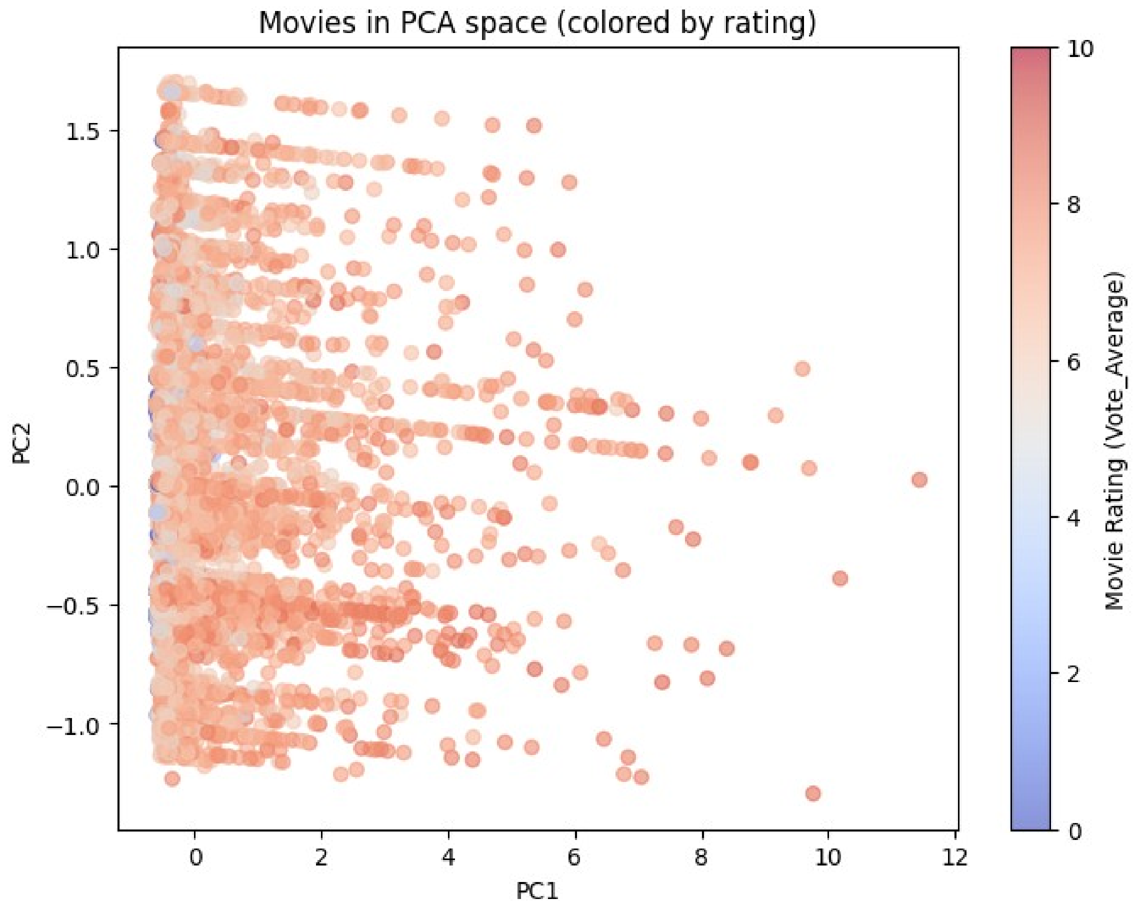
- Metrics explicitly printed: Mean Squared Error (MSE) and R^2 score.
- From the extracted plots:
  - PCA scatter plot shows data distribution in reduced feature space.
  - Predicted vs Actual scatter shows predictions against true ratings.

PCA Diagram

PCA in Movie Rating Prediction

- **Why use PCA?**
  Movie datasets often have many features (e.g., genre indicators, cast, budget, duration, user reviews). Many of these are correlated (e.g., budget & revenue, director & actor collaborations).
  PCA compresses these into fewer **uncorrelated components** while retaining most of the variance (information).
- **How it helps here:**
  1. **Reduces dimensionality** → makes training faster and avoids overfitting.
  2. **Removes multicollinearity** → Linear Regression struggles if predictors are highly correlated, PCA fixes that.

3. **Visualization** → Scatter plots of PCA components let you see if movies form clusters (e.g., movie vs low-budget).



What the plot shows:

- The **x-axis** = true (actual) movie ratings.
- The **y-axis** = predicted ratings from the Linear Regression model.
- The **red dashed diagonal line** represents an "ideal" case where predicted = actual.
- **Significance:**
- If points lie **close to the diagonal**, it means predictions are accurate.

- In your plot, predictions seem to cluster around a narrow band (around rating 6–7), which suggests the model has limited variability and struggles to capture extremes (very low or very high ratings).
- This highlights both the **baseline predictive ability** and the **limitations** of Linear Regression on this dataset.
- **How to use in report:**
    - The Predicted vs Actual plot shows that the regression model mostly predicts average ratings (around 6–7), while struggling with movies at the extremes.
    - The clustering away from the diagonal indicates that while the model captures general trends, it lacks precision for outliers.

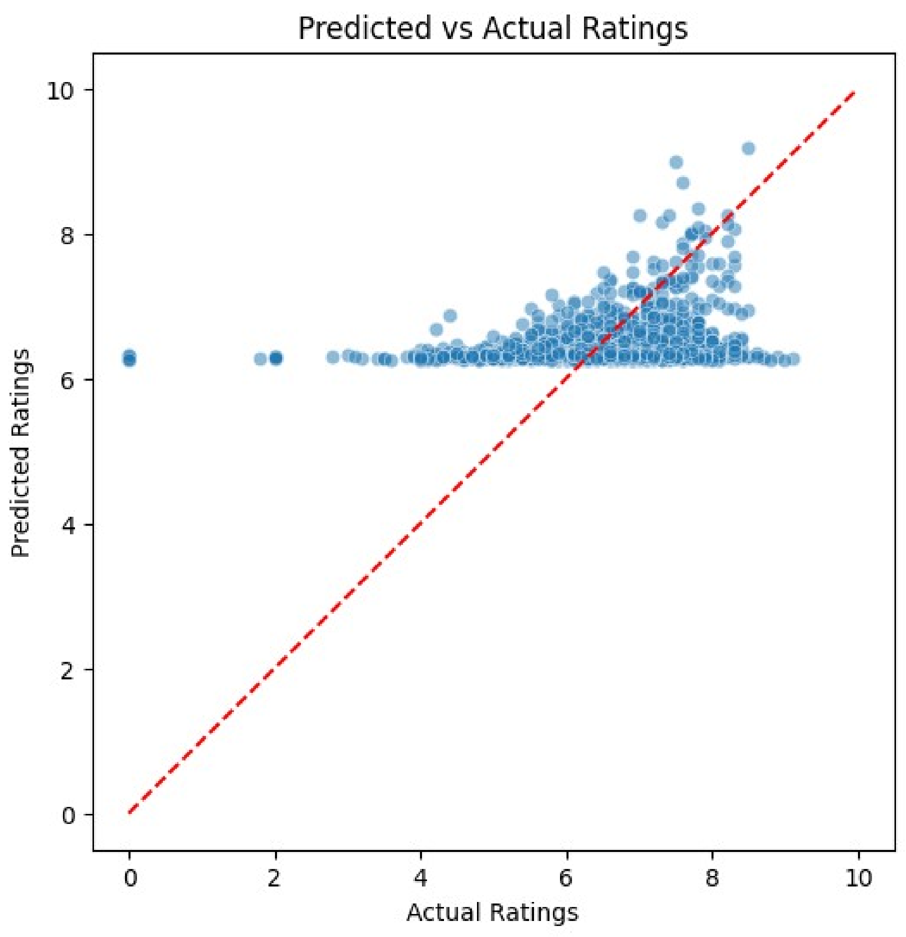
Linear Regression

- **Why use Linear Regression?**
  It's one of the simplest models to map numeric features → target (here, `ratings`).
  It assumes:

Rating=w1·PC1+w2·PC2+...+wk·PCk+b

- **How it helps here:**

1. Provides **baseline prediction** for movie ratings.
2. **Interpretability** → coefficients show which PCA components (and thus which original features) most strongly influence ratings.
3. Works well if relationships are roughly **linear**.


Predicted vs Actual Ratings
Predicted Ratings
Actual Ratings
0
2
4
6
8
10

What the plot shows:

- Movies are projected into **two principal components (PC1 and PC2)**.
- The **color gradient** represents movie ratings (blue = low rating, red = high rating).

  Significance:

- PCA compresses many original features into just two axes, allowing visualization of patterns.
- Here, the movies spread along PC1, but ratings (colors) appear fairly mixed — meaning ratings are not easily separable just by the top two components.
- This suggests that while PCA reduces dimensionality, predicting ratings requires more complex interactions across multiple components.

  How to use in report:

  - The PCA plot visualizes movies in reduced 2D feature space, with colors indicating ratings.
  - The distribution shows that ratings do not cluster strongly in PCA space, implying that ratings depend on a combination of multiple hidden factors.
  - This supports the use of regression to combine PCA components for prediction.

PCA+Linear Regression

1. **Input Data:** Movie features (e.g., budget, cast, reviews).
2. **Scaling:** StandardScaler makes all features comparable.
3. **PCA:** Reduces dataset to fewer uncorrelated features (PCs).
4. **Linear Regression:** Fits a model to predict ratings from these PCs.

5. **Evaluation:** Outputs metrics like **MSE** and **R^2**, and plots (Predicted vs Actual ratings).

In short:

- **PCA** helps simplify and clean up messy, high-dimensional movie data.
- **Linear Regression** then provides a straightforward way to predict ratings.
- Together, they form a neat pipeline: **"Compress features → Predict ratings → Evaluate."**

Recommendations & Next Steps

- Model robustness — Compare Ridge and Lasso regression; try tree-based models (Random Forest, XGBoost, LightGBM).
- Evaluation — Add more metrics (MAE, RMSE) and perform cross-validation.
- Data considerations — Verify dataset quality, missing values, balance of rating distribution.
- Reproducibility — Save model coefficients, add requirements.txt, fix random seeds.

Limitations of this Static Analysis

This analysis was conducted without executing the notebook.
- Metrics and plots included are based on extracted outputs, not freshly computed values.
- Any runtime errors, warnings, or updated results are not visible here.
- To confirm accuracy and obtain latest results, run the notebook end-to-end.