Predicting Movie Ratings using Linear Regression Models

Ria Rajasekar

Introduction:

My team's goal was to make a prediction model that can predict the rating of a movie given data about it. We did this by making three different models: Linear Regression, KNN, and Random Forest. I was in charge of the Linear Regression experiments. Linear Regression uses training data with inputs and outputs to come up with some function f, which predicts the output f(x) given some input x. Our dataset[1] pulls data from the IMDB website, and has the following features of 10179 movies: title, release date, rating, genre, description, status, language, budget, revenue, and country.

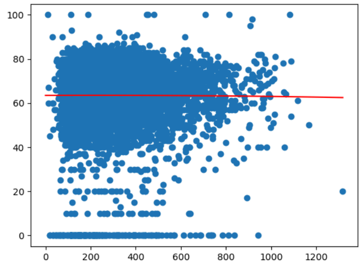
Overview of Methods:

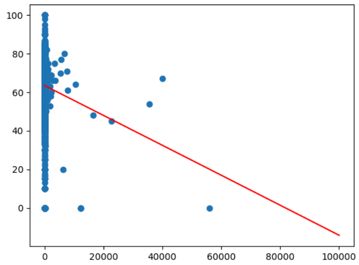
The key for linear regression was finding a way to characterize text features as numbers. To do this, we built off the idea of using its "linguistic aspects," or the structure of the text itself (Chambua and Niu, 2020). This characterization follows the assumptions that movies with common ratings will have text features that follow common linguistic patterns. We numericized linguistic aspects using Scrabble, which scores the text features based on its letters, giving a larger score to more 'unique' letters and longer text.

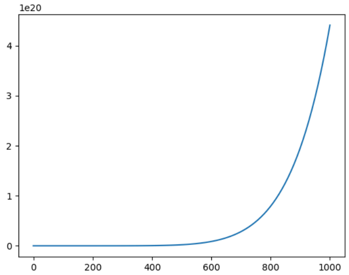
From here, we used this characterization method for the description of each movie along with their other text features. We followed the model from Lecture 13 page 20[2], and used those equations to get our MLE estimators for θ and σ^2. MLE estimation maximizes the likelihood of our prediction, and our estimators determine our prediction function $y = f(x) = \theta X$, along with the probability distribution of each prediction $f_{\theta,\sigma}(y, x)$. We did this for three different combinations of predictors: just the description, just the ratio of revenue over budget, and every feature. We weighted each predictor equally. For the model using every feature, we did not use the same equation from class but rather used a python library SKLearn (Buitinck et al., 2013) which has its own linear regression module; the matrix multiplication was giving us complications otherwise. SKLearn gets a function by minimizing the residual sum of squares between the line and each point. The tradeoff here is that this library does not get us the variance or distribution of each prediction in the model.

Results:

We made a total of 7 different models[3]. We validated our models using 5-fold cross validation: training our model on 4/5ths of the data, or our train data, and getting our mean squared error (MSE) using the other 1/5th of the data, or our test data. From there, we could see that the models which had the lowest MSE were the 2nd-degree models using description and earnings along with the model using all of our features as predictors, which is a 9th-degree model.







Top: Quadratic Model of Description vs Rating, with MSE = 183 and Variance = 183.
Middle: Quadratic Model of Earnings vs Rating, with MSE = 182 and Variance = 183.
Bottom: 9th-degree Model of All Features vs Rating[4], with MSE = 140.

Each of these models has a fairly high MSE of over 100, showing that on average these models will be off of the true rating by greater than 10. They also each have an extremely high variance, meaning that the model is not too sure of its prediction: there are a large range of values it could actually be. Thus, the assumptions we made when choosing the model likely aren't true in reality: the linguistic aspects of each text feature is not informative to the rating, and/or the rating does not have a linear relationship with the predictors we chose.

Bibliography:

Relevant Works:

Chambua and Niu. (2020) “Review text based rating prediction approaches: preference knowledge learning, representation and utilization,” Springer Nature. https://link.springer.com/article/10.1007/s10462-020-09873-y

Park, et al. (2016) "Predicting movie success with machine learning techniques: ways to improve accuracy," Springer Nature. https://link.springer.com/article/10.1007/s10796-016-9689-z

Siddique, et al. (2024) “Movies Rating Prediction using Supervised Machine Learning Techniques,” International Journal of Information Systems and Computer Technologies. https://journals.cfrit.com/index.php/ijisct/article/view/62/43

Zhang, et al. (2024) “Predicting popularity: Machine learning insights into movie team patterns and online ratings,” Issues in Information Systems. https://www.iacis.org/iis/2024/3_iis_2024_386-398.pdf

Jassim, M.A., Abd, D.H. & Omri, M.N. Machine learning-based new approach to films review. Soc. Netw. Anal. Min. 13, 40 (2023). https://doi.org/10.1007/s13278-023-01042-7

Tools Used:

Buitinck, et al. (2013) LinearRegression. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression.score

Hunter, et al. (2012) MatPlotLib. https://matplotlib.org/stable/

NumPy Developers. (2024) Numpy. https://numpy.org/doc/stable/index.html

Python Software Foundation License Version 2. (2025). SQLite3. https://docs.python.org/3/library/sqlite3.html

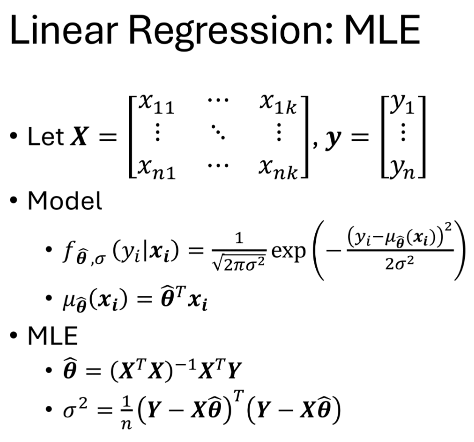
Python Software Foundation License Version 2. (2025). CSV. https://docs.python.org/3/library/csv.html

Mean_squared_error, scikit-learn. (2025). https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

Appendix:

[1] Link to dataset: https://www.kaggle.com/datasets/ashpalsingh1525/imdb-movies-dataset

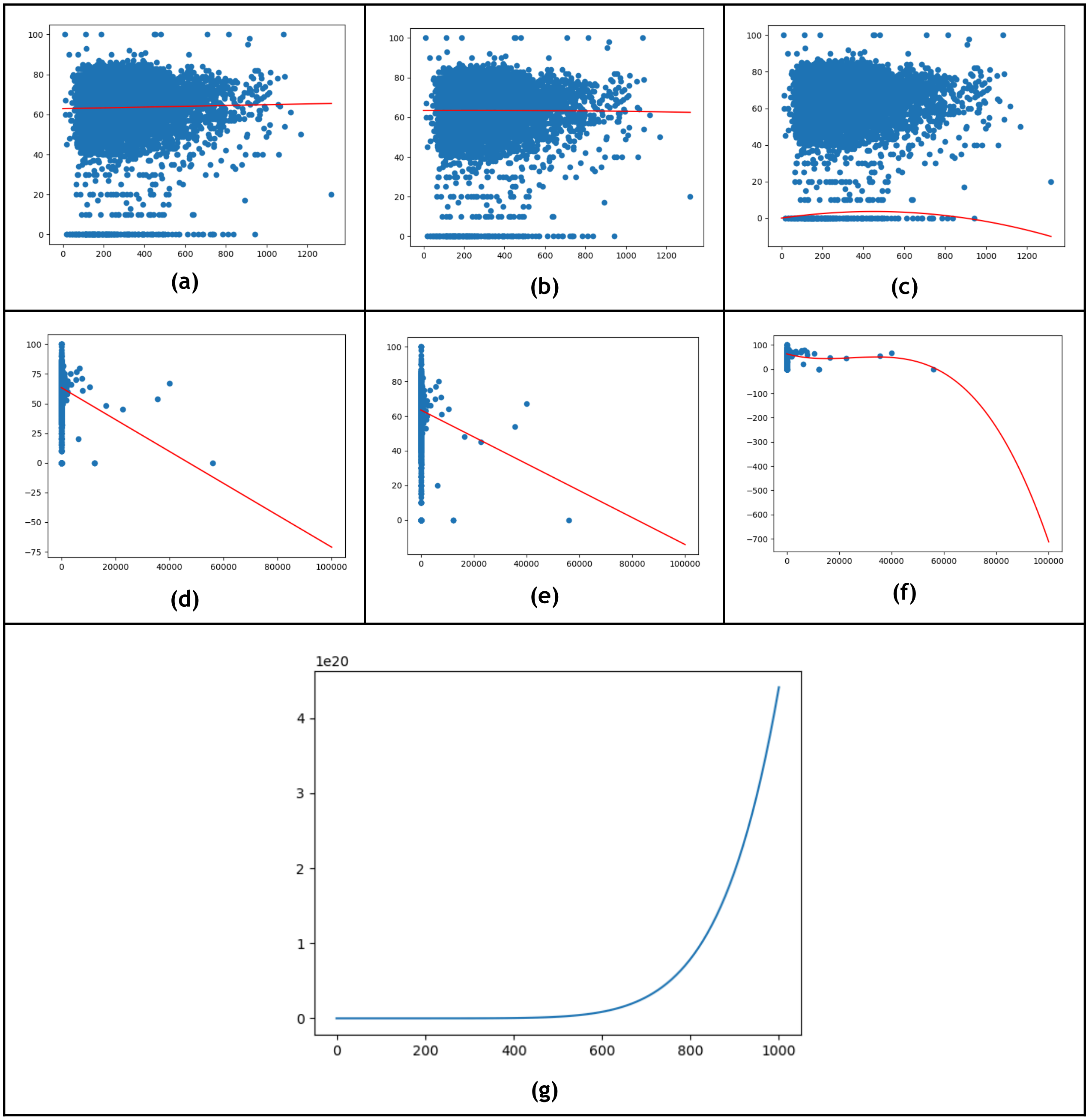
[2] Lecture 13, Page 30:



Linear Regression: MLE

- Let $\boldsymbol{X} = \begin{bmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nk} \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$
- Model
  - $f_{\widehat{\boldsymbol{\theta}},\sigma}(y_i|\boldsymbol{x_i}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(y_i - \mu_{\widehat{\boldsymbol{\theta}}}(\boldsymbol{x_i})\right)^2}{2\sigma^2}\right)$
  - $\mu_{\widehat{\boldsymbol{\theta}}}(\boldsymbol{x_i}) = \widehat{\boldsymbol{\theta}}^T \boldsymbol{x_i}$
- MLE
  - $\widehat{\boldsymbol{\theta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$
  - $\sigma^2 = \frac{1}{n}\left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\theta}}\right)^T\left(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\theta}}\right)$

[3] All results for linear regression:



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(a) Figure for Linear Model of Description vs Rating:
   (i) y= 0.0019901315756117866 x+ 62.9087778900077
   (ii) Variance = 182.9143169479819
   (iii) MSE = 184.0833071316481

(b) Figure for Quadratic Model of Description vs Rating:
   (i) y= -8.6055344506418e-07 x^2+ 0.0003881683320385976 x+ 63.479287738396856
   (ii) Variance = 183.17705277913035
   (iii) MSE = 183.3371494170967’

(c) Figure for Cubic Model of Description vs Rating:
   (i) y= -4.4810388687132364e-10 x^3+ -1.7200293602342293e-05 x^2+ 0.015797327139725148 x+ 0.015797327139725148
   (ii) Variance = 182.58721326417742
   (iii) MSE = 3875.812086354675

(d) Figure for Linear Model of Earnings vs Rating:
   (i) y= -0.0013465790915236824 x+ 63.53136987006687
   (ii) Variance = 182.30692704164582

    (iii) MSE = 185.69392164784702

(e) Figure for Quadratic Model of Earnings vs Rating:

    (i) y= -2.886181684734511e-11 x^2+ -0.0007745549542264585 x+ 63.52270160679331
    (ii) Variance = 182.94228947513284
    (iii) MSE = 182.21008352042645

(f) Figure for Cubic Model of Earnings vs Rating:

    (i) y= -1.888172889324754e-12 x^3+ 1.4119266597019597e-07 x^2+ -0.0030017269318347896 x+ 63.56054142237574
    (ii) Variance = 182.2176210439073
    (iii) MSE = 184.4823318584008

(g) Figure for Model of Multiple Feature vs Rating:

    (i) y= 2.80239321e-04x^9 + 1.85204458e-01x^8 + -2.93932856e+01x^7 + -1.14389190e-07x^6 + 1.94944783e-08x^5 + 2.04302982e-02x^4 + 4.65592585e-01x^3 + 1.41904649e-07x + 55.7336965409542
    (ii) MSE = 139.96186632162747

[4] For the model with all features, the visualization is simply to see the line of our model in 2D space. It is not representative of the actual model which is in 10D space due to our 9 predictors and 1 output.