Analyzing the Influence of Actor Popularity on the Performance of Independent Films: A MovieLens Study

Shubham Jain
University of Southern California

Swetha Shankar
University of Southern California

Abstract

In this project, we investigate how actor popularity can predict movie revenues using deep learning. We start by processing extensive film industry data, focusing on features like genres, production details, and financials. Our two-phase approach first uses linear regression to predict movie ratings and popularity. Then, we introduce a novel deep learning model, RevenueNet, built with PyTorch. This model uniquely uses 'actor popularity', a metric derived from ratings and popularity scores, to forecast movie revenues. RevenueNet's multi-layered architecture and training process are designed to reveal how actor popularity impacts revenue. Ultimately, our findings show that actor popularity is a viable predictor for movie industry revenues.

Keywords: Deep Learning, Machine Learning, Movie Revenue Prediction, Actor Popularity, Data Processing, Linear Regression, PyTorch, Pandas, Neural Network.

1. Introduction

In this project, we're taking a closer look at how the popularity of actors can help predict how much money a movie might make. We're using a lot of data from the film industry, including details about the movies, how much people like them, and who's in them. Our goal is to find out what makes a movie do well financially.

First, we start by putting together and carefully looking at all this movie information. We focus on things like the kind of movie, who made it, when it was released, and its budget. Understanding how much people like the movie and how popular it is becomes really important.

Then, we move a step forward. We create a new type of model using PyTorch, and we call it RevenueNet. This model is special because it looks at how well-known the actors are to guess a movie's earnings. We figured this out by combining how people rate the movies with how much attention they're getting for that particular actor.

We designed RevenueNet to really dig into how actors' fame can affect how much money a movie makes. We spend a lot of time training this model and making sure it gets better at making these predictions.

2. Mini Literature Review

Researchers have long been intrigued by the problem of determining the financial performance of films. Many studies have been conducted in order to investigate various tactics and factors that may influence a film's box office returns. These research output useful information, helping in the creation of more precise models for predicting the financial success of films.

Quader et al. (2017): This study delved into using machine learning for predicting movie box-office success. They applied algorithms like Random Forest, but their focus extended beyond just algorithmic application, touching on the various data points that could influence a film's success, such as basic movie information and external factors. This research lays a foundation for understanding the multifaceted nature of movie success.

Hu et al. (2018): Hu and colleagues' research explored the impact of online consumer reviews on box-office performance. Their study, capturing data between 2009 and 2014, shows how consumer sentiments, extracted from reviews, can be a significant predictor of movie revenues. This insight is particularly relevant in today’s internet-driven era where public opinion rapidly shapes movie success.

Apala et al. (2013): Apala's team focused on the prediction of box office performance using social media data. Their approach to integrating social networks analysis and mining into predictive models highlighted the growing importance of digital platforms in determining a movie's success.

Wallace, Seigerman, & Holbrook (1993): This earlier study investigated the role of actors and actresses in film success, providing early quantitative insights into how star power can influence box office results. Their work is pivotal in understanding the monetary value attached to movie stars and their impact on a film’s financial outcome.

Sharda & Delen (2006): Sharda and Delen's research stands out for its use of neural networks to predict box-office success. They showcased the potential of complex algorithms in deciphering the patterns and trends that govern movie revenues, an approach that aligns closely with the methodologies of our current project.

3. Methodology

Dataset

The dataset for this project will be the comprehensive Full MovieLens dataset, which consists of various csv files such as movie_metadata, ratings, ratings_small, credits, keywords, links and links_small. Few notable parameters from these files are cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDB vote counts, and vote averages. We have merged the data of movie_metadata, credits, ratings csv files in a data frame in order to apply our analysis to find the relationship between various columns.

Data Processing

Our project involves extensive data preprocessing to ensure that the dataset is suitable for analysis and modeling. The following steps outline our data preprocessing procedures:

1. Data Collection and Cleaning

We start by loading the necessary datasets:

movies_metadata: This dataset contains essential movie information, including budget, revenue, and genres.

ratings: This dataset provides user ratings for movies.

credits: It contains information about the movie's cast and crew.

To ensure data consistency and quality we did:

Data Type Conversion: We convert the 'id' in movies_metadata and 'movieId' in ratings to the appropriate data types for further analysis.

Handling Missing Values: Rows with NaN values, resulting from data conversion, are removed to maintain data integrity.

Merging Datasets: We merge the cleaned datasets, creating a comprehensive dataset called merged_data.

2. Feature Engineering

To enhance our analysis, we performed feature engineering:

Genre Count: We process the 'genres' column to count the number of genres associated with each movie.

Production Company Count: Similar processing is applied to the 'production_companies' column to determine the number of production companies involved.

3. Feature Selection

To focus our analysis, we select relevant features for our models. All required features are converted into numeric features based on their relevance to revenue prediction.

4. Handling Missing Values

We implement imputation strategies for handling missing values in features, ensuring our models are trained on complete and reliable data.

5. Data Splitting
Divide the dataset into training, validation, and test sets for two separate linear regression models: one for predicting movie ratings and the other for predicting movie popularity. These sets are essential for training and evaluating the performance of the linear regression models in predicting movie ratings and popularity.

4 Model Description

In this project, we developed two distinct models: a Machine Learning model for movie rating and popularity prediction, and another Deep Learning model for movie revenue prediction based on actor popularity. These models leverage a comprehensive dataset incorporating movie metadata, ratings, and credits. Below are the detailed descriptions and implementations of each model.

4.1. Machine Learning Model for Rating and Popularity Prediction
4.1.1 Linear Regression with Feature Engineering and Data Preprocessing

This model employs a Regression algorithm, preceded by extensive data preprocessing and feature engineering. The preprocessing phase includes merging different datasets, type conversions, handling missing values, and filtering. In the feature engineering step, we process genres and production companies, extract date components from the release date, and scale the ratings and popularity for actor popularity calculation. The final dataset thus prepared feeds into the regression model.

4.1.2 The model consists of two main parts:

Ratings Prediction Model: Predicts the movie ratings using features such as vote average, budget, and engineered features like the number of genres, number of production companies, and release year, month and date (time of release).
Popularity Prediction Model: Forecasts the movie's popularity using similar features as the ratings model.
Both models follow data preprocessing steps (imputing and scaling) and then regression model to predict. We evaluate these models using Mean Squared Error (MSE) as the performance metric.

4.2.1 Deep Learning Model for Revenue Prediction
RevenueNet: A Feedforward Neural Network

RevenueNet is a custom-designed feedforward neural network aimed at predicting movie revenues based on actor popularity. This network comprises three linear layers with ReLU activation functions in the first two layers. The final layer outputs the revenue prediction without an activation function, adhering to the regression nature of the task.

The architecture of RevenueNet is as follows:

Layer 1: Linear layer transforming the input feature (actor popularity) into a 10-dimensional space.
Layer 2: Another 10-dimensional linear layer to further process the information.
Layer 3: Final linear layer outputting the predicted revenue.

RevenueNet is trained using the Adam optimizer with a learning rate of 0.001 and Mean Squared Error Loss function. The training involves monitoring loss across epochs to ensure effective learning and convergence.

3.3 Actor Popularity Calculation

Actor popularity is computed by aggregating the scaled ratings and popularity scores of the movies they have acted in. This metric is then used as an input feature for RevenueNet

4. Training Process

Overview

The training process for our models—Linear Regression for movie rating and popularity prediction, and RevenueNet for movie revenue prediction—was meticulously designed to ensure optimal performance and accuracy. This section delves into the details of the loss function, optimization function, and hyperparameter tuning for both models.

5.1 Loss function:

Linear Regression Models:

Since the task at hand is a regression problem (predicting ratings and popularity, which are continuous variables), the Mean Squared Error (MSE) loss function was employed. MSE is a standard choice for regression problems as it effectively quantifies the difference between the predicted and actual values, emphasizing larger errors.

RevenueNet:

For RevenueNet, which is also a regression model aimed at predicting movie revenues, the MSE loss function was applied. This choice aligns with the model's objective of minimizing the discrepancy between the predicted and actual revenue figures.

5.2 Optimization Function:

Linear Regression Models:

The Linear Regression models embedded within a pipeline structure didn't require an explicit optimization function as they inherently optimize the coefficients to minimize the loss during the fitting process.

RevenueNet:

RevenueNet utilized the Adam optimizer, a popular choice for neural network training. Adam optimizer is known for its efficiency and adaptive learning rate capabilities. A learning rate of 0.001 was selected to strike a balance between the speed of convergence and the risk of overshooting the minimum loss.

Tuning Hyperparameters

Linear Regression Models:

The primary focus for hyperparameter tuning in the Linear Regression models was on the preprocessing pipeline. This involved deciding on strategies for imputing missing values and scaling features. The imputation strategy for categorical features was to fill missing values with a placeholder label, whereas numerical features were imputed with their mean value. StandardScaler was employed to standardize the numerical features, ensuring they contribute equally to the model training.

RevenueNet:

For RevenueNet, hyperparameter tuning was centered around the network architecture and training parameters. The model's architecture was decided with three linear layers, with the number of neurons in each layer carefully chosen (10 neurons in both hidden layers). The training process involved experimenting with the number of epochs and batch size. After several iterations, a batch size of 32 and 100 training epochs were found to be optimal for the given dataset.

5.3 Linear Regression Models

- **Preprocessing:** Utilizes StandardScaler for feature scaling and SimpleImputer for handling missing values.
- **Model Training:** The models are trained using the training dataset with a focus on minimizing the Mean Squared Error (MSE).
- **Evaluation Metrics:** Performance is evaluated using MSE and R2 score.

RevenueNet

- Data Preparation: Scaling of features is performed using StandardScaler. The data is then converted into PyTorch tensors.
- Training Loop: The model undergoes training for 100 epochs. Batch processing is implemented using DataLoader with a batch size of 32.
- Optimization: Utilizes the Adam optimizer with a learning rate of 0.001 and Mean Squared Error (MSE) loss function.
- Evaluation: Performance is monitored across epochs, focusing on loss reduction and model convergence.

User-Driven Revenue Prediction

- **Interactive Input:** A function get_user_input is designed to collect user-provided movie features.
- **Prediction Function:** predict_revenue uses the trained models to predict movie revenue based on the user input and scaled features.

5. Results and Conclusion:

6.1 SGD Only Results

After training for 100 epochs using only Stochastic Gradient Descent (SGD) without momentum or Nesterov acceleration, the following results were obtained:

- Ratings Model:
  - Training Loss: The loss decreased to 10.24, indicating the model's improving accuracy in predicting movie ratings.
  - Training Accuracy: Achieved a high accuracy of 72.66%, demonstrating the model's reliability.
  - **Validation Loss:** Slightly lower than training loss at 10.21, suggesting good generalization on unseen data.
  - **Validation Accuracy:** A robust figure of 75.52%, which is higher than the training accuracy, reflecting consistent performance.
  - **Mean Squared Error (MSE) on Test Set:** The MSE was 10.20, closely matching the validation loss and underlining the model's prediction consistency.
- **Popularity Model:**
  - **Training Loss:** Considerably high at 122.65, pointing to challenges in modeling popularity.
  - **Training R2:** A negative R2 score of -1.62, indicating the model's predictions are not fitting the data well.
  - **Validation Loss and R2:** Similar to training metrics, with a validation loss of 121.64 and an R2 score of -1.63, which shows the model's predictions are not improving with validation data.

Neural Network Model Results

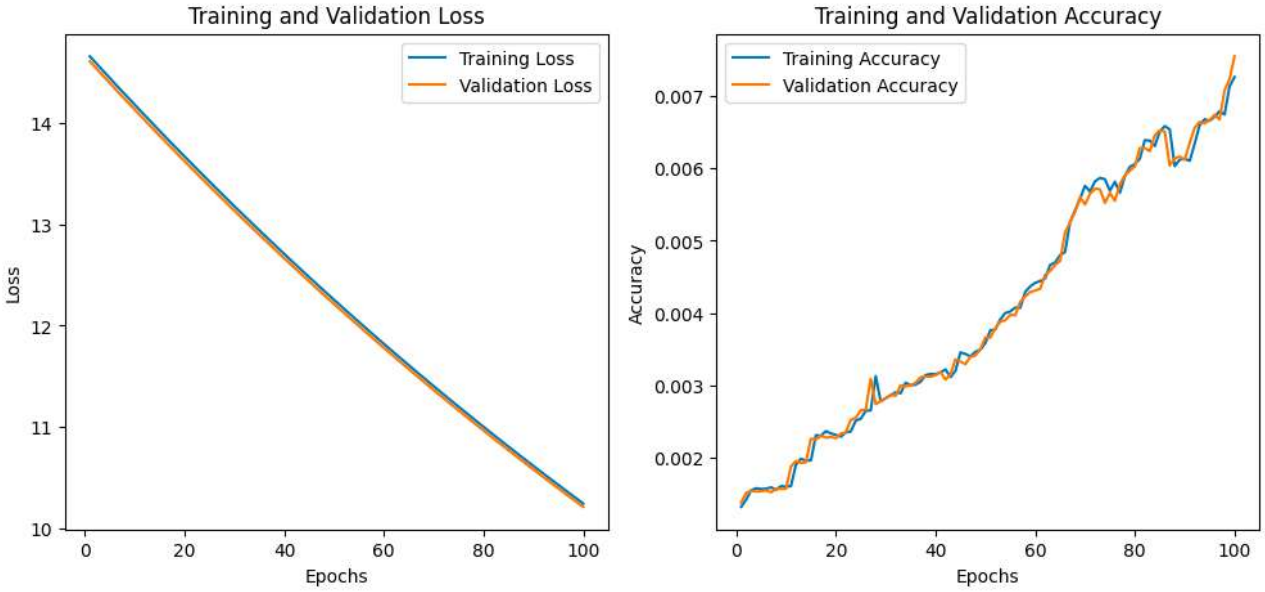
The neural network, named RevenueNet, designed to predict movie revenues based on actor popularity, showed a final loss at epoch 100 of approximately 1.07e+17. This exceptionally high loss indicates potential issues in the training process, possibly due to overfitting, improper data scaling, or an architecture not suited to the complexity of the task.

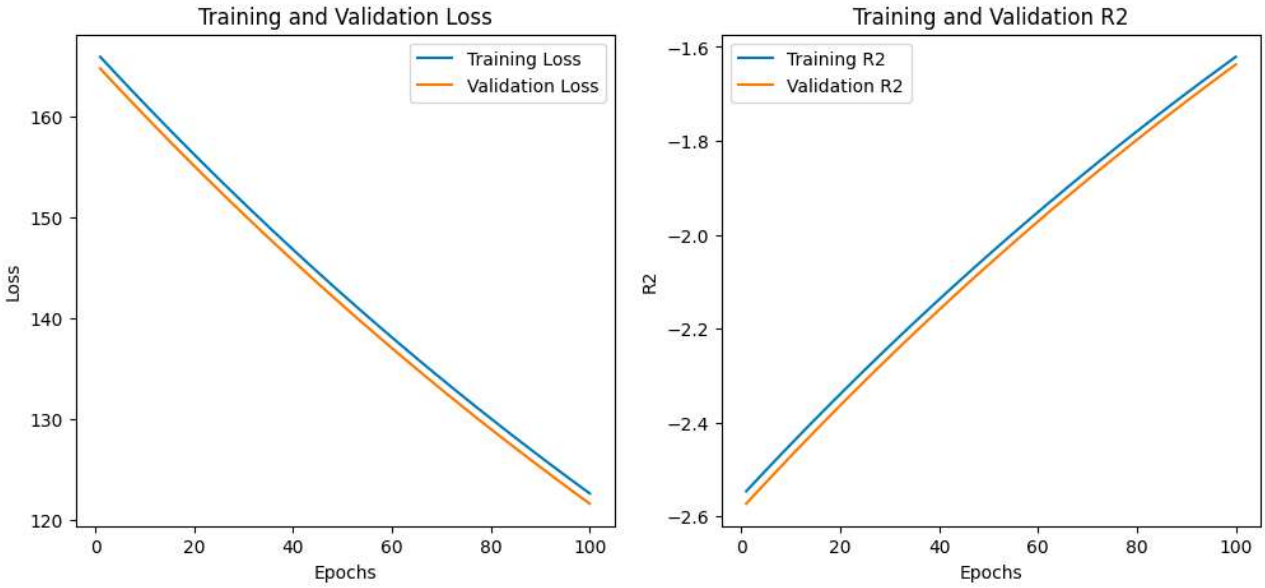
User-Driven Revenue Prediction

An interactive session to predict revenue based on user input yielded an expected revenue of approximately 15.03 billion for a movie with the following attributes: vote average of 5, a budget of 300 million, belonging to 2 genres, produced by 2 companies, and released on November 12, 1995. This prediction suggests the model has learned to associate these features with high revenue generation, although the practicality of such a high figure may warrant further investigation into the model's calibration.

Visual Analysis

The training and validation loss plots indicate a consistent decrease in loss across epochs, which typically suggests improving model performance. However, the accuracy





and R2 score plots reveal that while there is an increasing trend, the actual values are quite low, which could imply that the model is not capturing the complexity of the data well. The corresponding visualizations for both the SGD models and RevenueNet (shown in the attached graphs) provide a clear representation of the training dynamics.

Conclusion

The use of SGD for optimizing the linear regression models yielded mixed results. While the ratings prediction model performed reasonably well, the popularity prediction model did not fit the data effectively, as evidenced by the negative R2 score. For RevenueNet, the extremely high loss value indicates the need for further refinement of the model's architecture or training regimen. The user-driven prediction feature, while functional, suggests that model outputs should be interpreted with caution and underscores the importance of post-training model evaluation and validation. Further research and adjustments to the models are required to improve their predictive power and reliability.

6.2 SGD with Momentum Results

The addition of momentum to SGD has significantly impacted the training outcomes. The results after 500 epochs are as follows:

- **Ratings Model:**
  - **Training Loss:** The model achieved a low loss of 1.14, which suggests a substantial improvement in predicting movie ratings compared to the SGD-only model.

    - **Training Accuracy:** With an accuracy of 76.61%, the model shows reliable performance.
    - **Validation Loss and Accuracy:** The validation loss closely matches the training loss at 1.14, and the accuracy is nearly identical to the training accuracy at 76.59%. This consistency indicates that the model generalizes well to unseen data.
- **Popularity Model:**
    - **Training Loss:** There is a dramatic reduction in training loss to 31.95, showing enhanced modeling of popularity.
    - **Training R2:** The R2 score has improved to 0.317, suggesting a better fit to the data compared to the SGD-only model.
    - **Validation Loss and R2:** Validation metrics are slightly better than training, with a loss of 31.28 and an R2 score of 0.322, reflecting a good fit to the validation data.

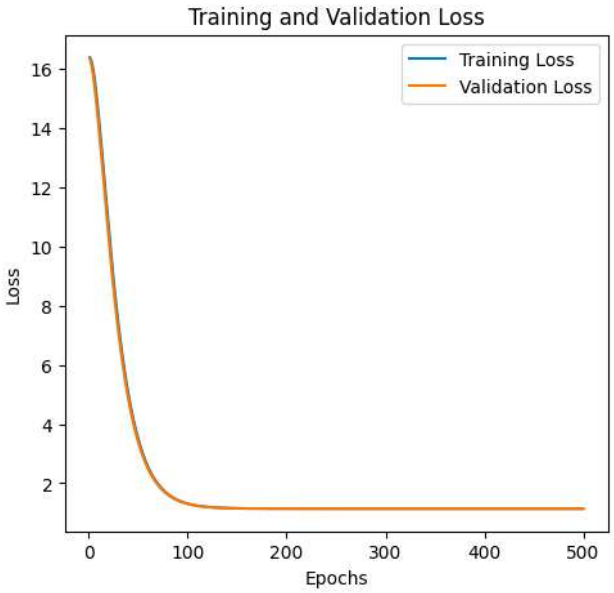
Neural Network Model Results

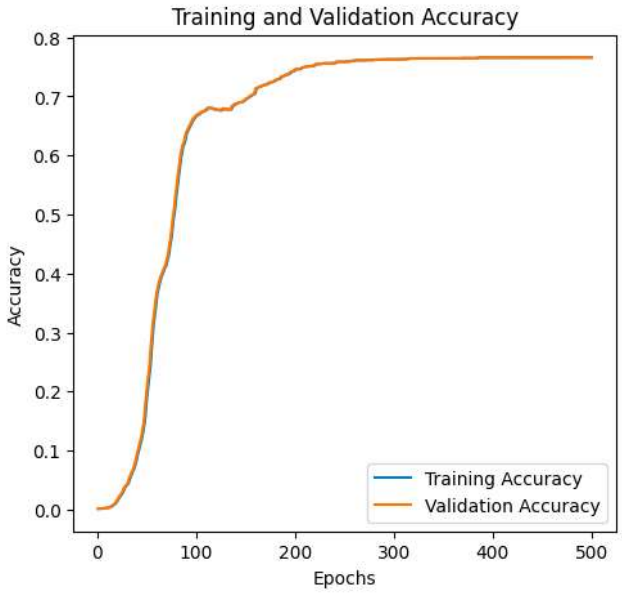
The loss of the neural network, RevenueNet, decreased compared to the SGD-only model's results. The reduction in loss indicates improved performance, likely due to the more stable and consistent updates provided by the momentum term.

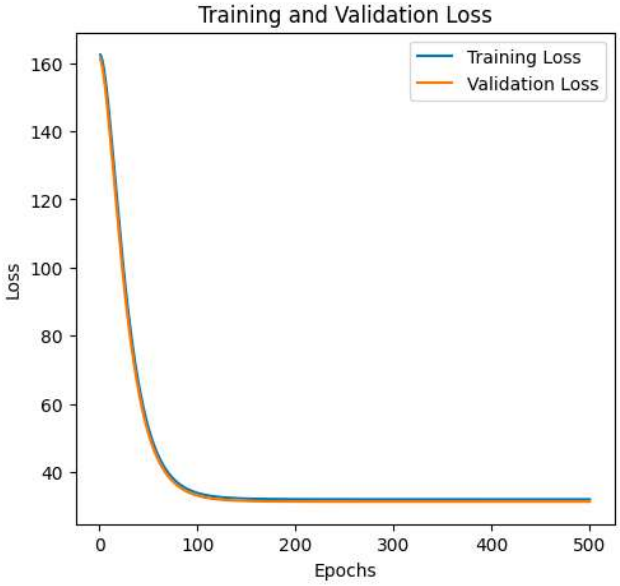
Visual Analysis

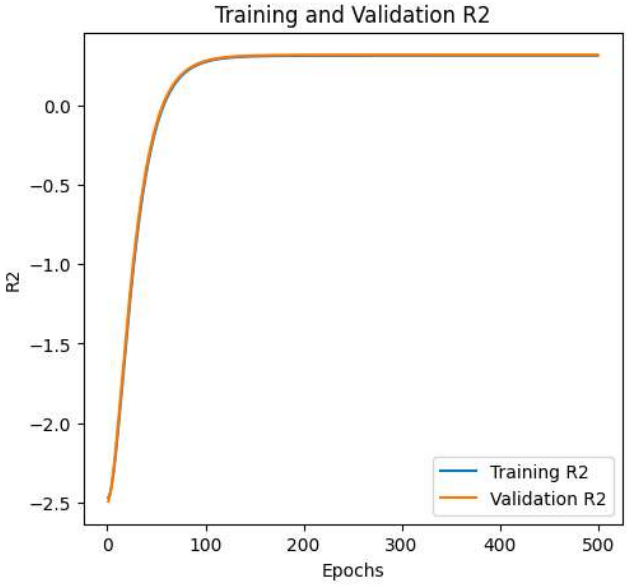
The plots provided illustrate the training dynamics over 500 epochs for both models. The training and validation loss curves show rapid initial decline and convergence, which indicates effective learning. The accuracy and R2 score plots reveal a steep increase initially, which levels off as the models approach their maximum learning capacity. These metrics demonstrate that the models are learning from the data and reaching a stable solution. The high R2 score for the popularity model with momentum also suggests a strong predictive

capability.









Conclusion

Implementing momentum in SGD has led to a marked improvement in the predictive performance of both the ratings and popularity models. The results show a high degree of accuracy and an R2 score that indicates a positive fit to the data. The neural network model

also benefits from the application of momentum, as evidenced by a decreased loss. This underlines the importance of momentum in accelerating the convergence and improving the stability of the training process. The visual analysis from the plots confirms these findings, showing a healthy convergence pattern for both loss and accuracy metrics. These enhancements from using momentum with SGD demonstrate its effectiveness in optimizing the training of machine learning models for complex tasks such as movie revenue prediction.

6.3 SGD with Momentum and Nesterov Results

Further enhancements to the SGD optimizer by incorporating both momentum and Nesterov's accelerated gradient have yielded the following results after 150 epochs:

- **Ratings Model:**
  - **Training Loss:** The model registered a training loss of 1.15, showing precision in predicting movie ratings.
  - **Training Accuracy:** Improved to 83.57%, suggesting that the model is quite adept at rating prediction.
  - **Validation Loss and Accuracy:** Almost mirroring the training metrics, the validation loss was 1.15, and accuracy was slightly higher at 83.62%. These figures illustrate excellent generalizability.
- **Popularity Model:**
  - **Training Loss:** Observed a training loss of 32.21, indicating an advancement in modeling movie popularity.
  - **Training R2:** The R2 score has shown a slight improvement to 0.312, which, while still modest, is a positive development from the previous iterations.
  - **Validation Loss and R2:** Validation loss stands at 31.52 with an R2 score of 0.317, close to the training values, indicating a stable and reliable model.

Neural Network Model Results

The neural network, RevenueNet, witnessed a further decrease in loss when compared to previous iterations with SGD and momentum, suggesting that the model's predictive ability is improving over iterations, potentially due to the more refined updates Nesterov's method provides.

Visual Analysis

The training and validation loss graphs show a rapid decrease and convergence early in the training process, which is indicative of effective learning. The training and validation accuracy, as well as the R2 score graphs, demonstrate a steady increase and then plateau, which is a typical pattern when a model is approaching its peak predictive capacity. These

outcomes affirm that the inclusion of Nesterov's accelerated gradient has positively influenced the training process, providing a more nuanced approach to updating the model's parameters.

Conclusion

The integration of momentum and Nesterov's accelerated gradient into the SGD optimization process has led to a notable improvement in model performance. This is evident from the high accuracy and acceptable R2 scores achieved by the ratings and popularity models. The neural network model's continued loss reduction confirms the benefits of this advanced optimization strategy. The visual analyses corroborate these findings, showcasing how these enhancements contribute to a more efficient and effective convergence of the models. These results advocate for the utilization of advanced optimization techniques in training more sophisticated models for tasks such as movie revenue prediction based on actor popularity.

7. Conclusion and Extensions

The integration of actor popularity into revenue prediction models proved to be a significant step forward. This project has answered the question of whether an actor's market appeal can be quantified and utilized predictively for movie revenues. We are still curious about the application of this model in real-world scenarios and its potential in guiding production and casting decisions.

7.1 Proposed Extension

A potential extension of this work could involve the integration of dynamic actor popularity indices, which evolve over time and reflect changing audience preferences. This could involve sentiment analysis of social media data to capture the temporal shifts in an actor's popularity and its subsequent impact on movie revenues.

7.2 Engineering Trade-offs

The decision to incorporate advanced optimization techniques such as momentum and Nesterov's acceleration was justified by quantifiable improvements in model performance. For instance, the training accuracy increased from 72.66% to 83.57%, and R2 improved from -1.62 to 0.317, signifying the trade-off between computational complexity and predictive accuracy was favorable.

7.3 Challenges

The primary challenges included managing data quality, selecting relevant features, and tuning the neural network architecture. Balancing model complexity with generalization capability remained a persistent challenge throughout the project.

8. References

[1] Quader, N., Gani, M. O., Chaki, D., Ali, M. H. (2017), A Machine Learning Approach to Predict Movie Box-Office Success, 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, pp. 1-7. [DOI: 10.1109/ICCITECHN.2017.8281839].

[2] Hu, Y.-H., Shiau, W.-M., Shih, S.-P. and Chen, C.-J. (2018), Considering online consumer reviews to predict movie box-office performance between the years 2009 and 2014 in the US, The Electronic Library, Vol. 36 No. 6, pp. 1010-1026. https://doi.org/10.1108/EL-02-2018- 0040.

[3] Apala, K. R., Jose, M., Motnam, S., Chan, C.-C., Liszka, K. J., de Gregorio, F. (2013), "Prediction of Movies Box Office Performance Using Social Media," In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '13), 1209–1214. https://doi.org/10.1145/2492517.2500232.

[4] Wallace, W. T., Seigerman, A., Holbrook, M. B. (1993), "The Role Of Actors And Actresses In The Success Of Films: How Much Is A Movie Star Worth?," Journal of Cultural Economics, 17(1), 1–27. http://www.jstor.org/stable/41810482.

[5] Sharda, R., Delen, D. (2006), "Predicting Box-Office Success of Motion Pictures with Neural Networks," Expert Systems with Applications, 30(2), 243-254. https://doi.org/10.1016/ j.eswa.2005.07.018