

Model-based predictive control for continuous success planning in movie production

Authors:

Suman Kalyan, Data Scientist

ML Consultant

Angshuman Patra, SVP & BU Head

Accion Labs

Sujay Kumar, SVP of Business Development

Accion Labs

Abraham Addanki, Data Scientist

Ness Digital Engineering

Shashank Sahoo, Lead Data Scientist

Ness Digital Engineering

Abstract. *The movie industry across the globe is a multi-billion-dollar business for stakeholders. Perhaps the single most financially dominating recipe among all flavors of media entertainments available to the audience. A plethora of carefully coordinated onerous efforts encompassing acting, direction, scriptwriting, casting, editing, and production goes into making and releasing a movie. Financiers and producers reel through several financial losses if the movie does not perform well at the box office. Given the complicated composition of time, money, imagination, creativity, and risk, it becomes imperative to have some measurable, quantifiable, and controllable parameters to predict movie performance on box-office during various stages of making through leveraging insights from plot-summaries, trailers, posters, teasers, social-buzz, cast/crew-selection etc.*

Can specialized Artificial Intelligence techniques help from historical data across these modalities to best predict the success of a movie and be the foundation for creating a sustainable architecture for Model-based predictive control? This paper proposes a Model-based predictive control to create a closed-loop feedback system that will enable superior production planning. Model-based predictive control has seen success in the Industrial automation domain and has been historically used for establishing feedback control loops for industrial automation processes. The paper also presents experimental results on the foundation needed for a model-based predictive control: A hybrid, yet comprehensive RNN/ LSTM/CNN-based neural network architecture that can predict the success of the content much earlier to optimize production cost. It leads to a useful foundation that enables content creators to manage especially the pre-production part of the journey much better by making the right decisions that enhance the probability of success.

Keywords. *CNN, LSTM, RNN, Movies, Posters, Videos, Genres, Explainability, SHAP, Grad-CAM, Multimodal architecture, Model-based predictive control*

Introduction

The financial success of a movie depends on many different factors and the production and release process goes through a number of steps. Each step of the process generates data that today is not always used to the full extent. A simple high-level illustration of the movie lifecycle and the associated datasets and attributes are listed in Figure 1:

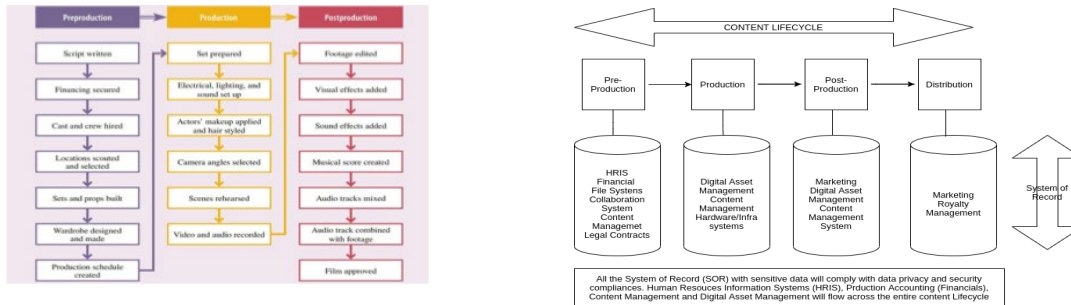


Figure 1: (Image on the left: Source Credit: [India shoots website](#))

Each of the phases, as described above, is defined, critical, and can be expensive. For example, Movie distribution is a finance-intensive [8] activity that combines a lot of core data from marketing, advertisement, scheduling, location, etc. with behavioral information like understanding the audience's [3] pulse, socio-political climate, and others. to provide insights on a successful launch strategy of the movie.

Multiple studies and research have been carried out in the recent past to understand these factors through various quantitative approaches like deep learning. Most of these studies focus on estimating the success probability of contents. Today, movie success prediction systems rely extensively on post-production and post-release features such as teasers, trailer clips, movie reviews, etc. to evaluate movie performance at the box office.

The authors of this paper carried out several literature studies/surveys across various machine learning methods. These studies focused on the movie outcomes [1] based on content popularity. One observation was that datasets used to build baseline foundation models took into consideration either structured [4] or unstructured [1,2,6,7] data monotonically. While this data choice helped with critical analysis of the movie outcomes, we think the results missed predictions across hybrid data sources and under-valued the complex data interplay across hybrid data sources and under-valued the complex data interplay across process steps resembling an actual movie production process. Furthermore, an interesting work [5] leverages linguistic features to predict violence in movies both at utterance and movie level. This however comes with certain limitations as the approach does not account for modifications in post-production (e.g., an actor delivering a line with a threatening tone).

In a typical movie production process, a lot of data is generated and available during the pre/production/post-production process for analysis/ analytics:

- Content (plot, videos, promos, audio, music score, etc)
- 'Buzz' (combination of social chatter, impressions, editorial mention, etc.)
- Reviews
- Budget
- Distribution
- Production Cast & Crew
- Genre
- Release Date

We believe the richness of data (structured, semi-structured, and unstructured) across the movie-making lifecycle and the iterative nature of the production process make it a perfect candidate for a data-driven continuous prediction modeling, that can act as a guide to decision making at every stage of the content lifecycle. For this, we took inspiration from how model-based predictive controls are being used in the Manufacturing/Industrial sector, yielding a more automated and efficient way to control the outcome across a process.

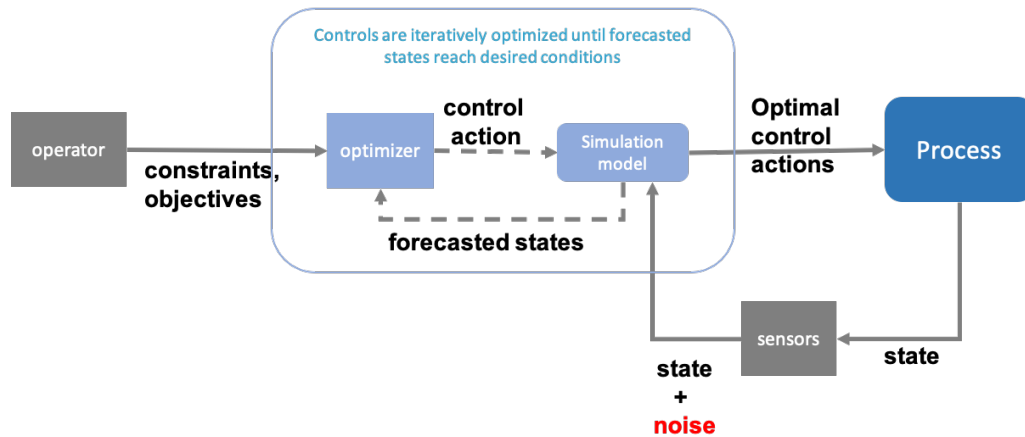


Figure 2: AI for Industrial process control (Model-based predictive control)^[24]

An approach is known as Model-Based Predictive Controls (MBPC), used in manufacturing (Figure 2), uses a continuous monitoring and control loop that incorporates predictions at each stage of the production process and uses that data to guide control actions in the upcoming steps of the process.

We can look at movie production as a set of process steps:

- **Process:** The production process
- **Objectives:** Maximize on outcome variables (IMDB score, vote average, movie collections in the first 2 weeks, number of impressions, etc)
- **Constraints:** On input variables like budget, cast, crew, etc.
- **Control Action:** Changes to input variables to move it closer to desired outcomes(with the constraints added)
- **State:** Predicted outcome (IMDB score, vote average movie collections in the first 2 weeks, number of impressions, etc)
- **Simulation Model:** ML model that can provide inferences on the outcome variables
- **Optimizer:** A mathematical solver that can iteratively determine the “optimal” control action based on outputs from the simulation model

Applying MBPC to the above process can allow production houses to run their production processes more efficiently by operating within a tighter bound of the constraints than would be possible with conventional planning and management.

Each control lever can then be used to optimize and achieve the desired outcome. For example, choosing a different poster or changing a few members from the production crew, or refining the plot of the movie.

Constraints can vary based on the stage of a production process. e.g: constraints on budget, options for supporting actors, the better quality of the poster. Accurate model predictions can provide early warnings of potential issues in content quality and the Control Actions are used as a corrective mechanism to move closer to the desired higher quality.

Model-Based Predictive Control Architecture

Model-based predictive control has been used in the industrial automation domain for creating a closed feedback loop control system for the effective management of industrial processes. The key features of this system include a system model, a dynamic optimizer, a way of handling constraints, and a cost function. Predictive controls for handling linear systems have been far more successful compared to nonlinear systems particularly when the expectations are to provide near real-time updates to the control variables. For nonlinear systems, the demands on the optimizer are far greater. Figure 3 below illustrates the proposal for a model-based predictive control architecture for a Media Industry that can derive value in using this approach for a more effective data-driven approach in the creation of more successful content.

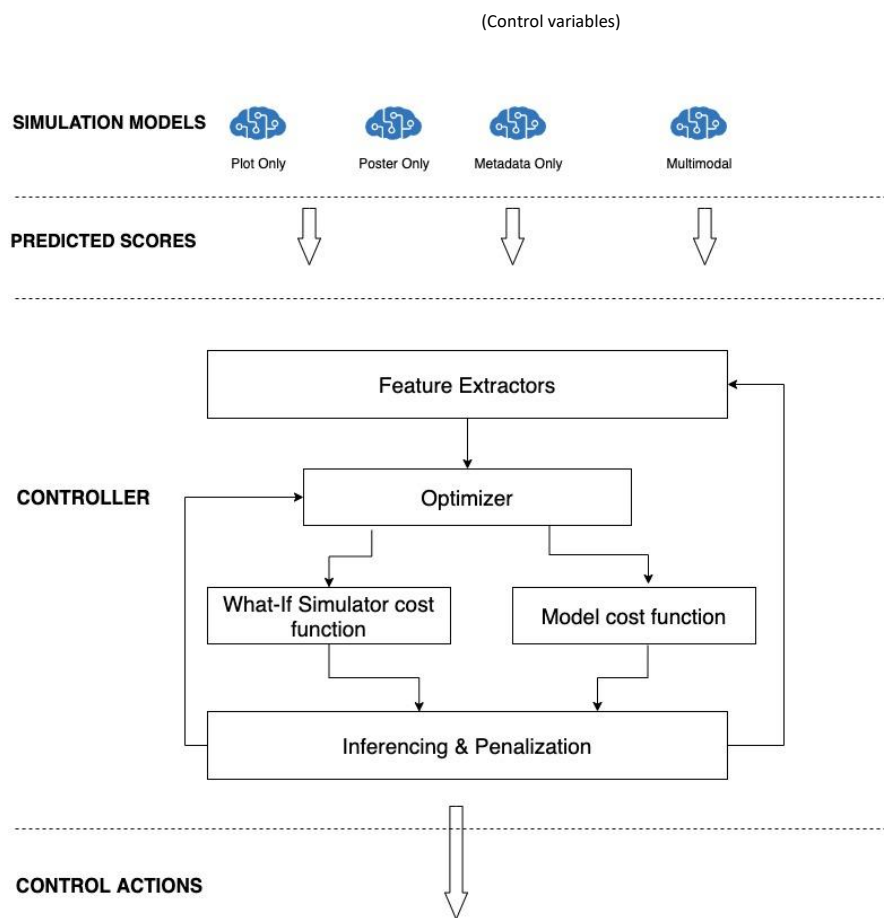


Figure 3: Proposal for a Model-based predictive control architecture for Media industry production process

The various components of the Model-Based Predictive Control Architecture are as follows:

- a. **Inputs:** Based on the content lifecycle stage of the pre/ post-production process the inputs could vary. At the early stage, it could just be the plot of a movie or at later stages include more features including audio, promotional video, etc. The input space is essentially the **control variables** that will be optimized by the Optimizer.

- b. **Deep learning model inferencing layer** that can provide inferencing for different modalities of input data. The model inferences in the form of **predicted success scores** are used to set the **lower setpoint of the feedback control loop**.
- c. **Feature extractors**: This layer extracts information from the multimodal inputs. The feature extractors are derived from the **weight's matrices of the** deep learning models and transform the input data to higher-level features and in the process compress the input information for further processing.
- d. **Optimizer**: The optimizer sets up the process of optimizing the control variables.
- e. **What-If simulator cost function with constraints**: This is a custom cost function that works on penalizing the non-influential custom features formed by the optimizer.
- f. **Model cost function with constraints**: This is typically the generalized regression cost function.
- g. **Inferencing and penalization**: This layer works on the actual inference and iterates through its deviation from the expected value and adjusts the bias based on the deviation by either reselecting important features from the feature set or by applying penalization on non-influential features.

At any stage of the content creation lifecycle, based on the input data available, the right model from the deep learning model inferencing layer can be used to predict the chances of success. A high target value (desired value) of 90% and above can be chosen as a **goal**. The closing of the gap between the predicted and the desired value is through the process of optimization using an optimization scheme like **convex optimization**. The **Optimizer** does multiple iterations to explore the input feature space based on the constraints to arrive at an optimal set of inputs that maximizes the predicted value and gets it closer to the desired value. The new set of inputs that result from the Optimization iterative process are the actions to be taken to enhance the chances of success of the content.

In order to understand intuition in a little more depth, think about "Point of Equilibrium"[19][20]. Which mathematically is a constant result of a differential equation or in simpler terms a constant point of convergence or fulcrum point where results seem almost constant. Hence when the **Optimizer** iterates over the input feature space based on the constraints, the point of equilibrium is where all the features converge to equilibrium and hence are optimal feature values for the convex optimizer constraint, This can famously be called the "Region Of Attraction"(ROA) [20]. This is derived from the Lyapunov Theory [21] for system stability analysis. So, in theory, we are applying the Lyapunov Theory on a complex neural network and identifying a scalar surrogate equilibrium point or points.

Suppose there is a function V and constant α such that

- V is positive definite
- $\dot{V}(z) \leq -\alpha V(z)$ for all z

Then there is an M such that every trajectory of $\dot{x} = f(x)$ satisfies

$$\|x(t)\| \leq M e^{-\alpha t/2} \|x(0)\| \text{ which is called the global exponential stability (GES)}$$

The idea is that $\dot{V} \leq -\alpha V$ gives a guaranteed minimum dissipation rate, proportional to the energy

Once the ROA has been sketched out, the architecture has the **What-If simulator cost** function whose main purpose is similar to a sigmoid function, but with a twist. The role of an activation function in a network as we know is simple - it decides whether a neuron should fire or not. Also, during the backpropagation, we need to derive the activation function to learn the weights of the network. Since we are dealing with nonlinear systems, working with a usual class of sigmoidal activations may not be particularly helpful.

Hence we would like to introduce the “Algebraic Sigmoidal” [20][22] activation function which is as follows :

$$F(x) = x/\sqrt{x^2 + 1} \text{ ----- (1)}$$

This Algebraic Sigmoidal is known as the most “musical” sigmoid function [22] and has been used in various analog electronics and music signal processing. We wanted to use this as our activation function with a combination of the Lyapunov Theory in order to make the evaluation of the gradients cheaper. As mentioned above the activation functions derivative (1) is as follows:

$$d/dx(x/\sqrt{x^2 + 1}) = 1/(\sqrt{x^2 + 1})^3 \text{ ----- (1)}$$

Which as we know is much faster for evaluation when compared to the other families of sigmoids.

The foundation for this architecture is based on the viability of different deep learning models for different data modalities at different stages of the content lifecycle. The next section on the experimental setup proves this concept by training deep learning models of different modalities, however, for a commercial-grade application, this does not constitute an exhaustive list of deep learning models that would be needed. This is only a representative list.

Experimental Set-Up

The research study and experimentation are performed corroborating with the objective to build a heterogeneous deep learning framework that can process different categories of input streams to predict the rating of movie content. This framework can be applied during the pre-production phase to predict the best chances of success quantitatively. The commercial success of a movie is measured to be deterministic in terms of audience ratings and financial returns. The value of such a framework during pre-production is self-evident in terms of providing a foundation for decision-making during the pre-production process related to cast, crew, movie posters, video content, the plot, etc.

The deep learning framework presented in this paper demonstrates the feasibility and mathematical formulation of this prediction approach by adding the following input data streams to predict the success of content [23].

1. Plot summaries of the movie
2. Movie poster(s)
3. Features from movie metadata (cast, crew, production company)

The approach implemented consolidates inputs from texts (plot-summaries), images (posters), and movie metadata after vectorizing them individually across parallel data pipelines. The paper also discusses how the framework can be extended to support more input parameters like audio, the movie video sequences, and features related to the capture of historical facial expressions of the audience watching the content. Furthermore, the levels of influencers can be controlled and tailored suiting the narrative, production cost, audience popularity, and social-media hype to ensure optimized return on investment.

To conduct this experiment, we configured a sophisticated network of distributed training clusters on AWS SageMaker to efficiently aggregate all datasets from an S3 bucket to get consumed by pre-processing pipelines for handling missing and inconsistent data which later gets vectorized in the feature designing phase. The data thus consolidated is used by the stacked deep learning model to extract patterns and learn various geometry of feature composition for optimizing prediction results.

Model Architecture

The deep learning model architecture is designed to implement the hybrid stacked design paradigm consisting of various vectors obtained from feature engineering of multiple inputs. The multi-modal nature of the design (Figure 3) includes model inputs from movie poster images, textual plot summaries, cast-crew-popularity metadata, and could accommodate screenshots of trailer/ teaser videos. Every input type is treated separately by infusing the images into a Convolution Neural Network (CNN), the texts into a Long Short-Term Memory (LSTMs), the movie metadata into an Artificial Neural Network (ANN), and finally the videos into a Recurrent Neural Network (RNN). The processed output from each neural net is condensed into multiple Fully Connected (FC) vector layers which later gets consolidated through a vector concatenation operation. The concatenated layer is parsed into the final FC layer followed by the 19-channel final output head. Each of the output channels has its own sigmoid activation function to compute the final movie success prediction score (Vote average).

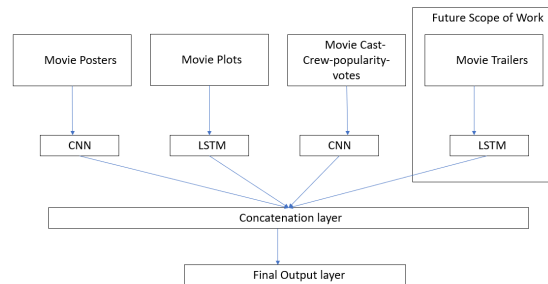


Figure 3. High-Level Diagram of Heterogeneous Model Architecture

The experimentation was done on a dataset of 5000+ odd movies which resulted in the following metrics as tabulated in Table 1:

	Metric	Hand crafted model	VGG16	VGG19	ResNet50	Algebraic Sigmoid Activation
Training	Size of training dataset	3357	3357	3357	3357	3357
	MSE	0.0017	0.0019	0.0026	0.0026	0.0045
	MAE	0.0302	0.0321	0.0395	0.0395	0.0541
	R2	0.8226	0.8015	0.8415	0.8415	0.5322
	Loss (MAPE)	5.5425	5.8404	6.5404	6.5404	5.8301
Cross validation	Size of validation dataset	1343	1343	1343	1343	1343
	MSE	0.0017	0.0019	0.0026	0.0026	0.0045
	MAE	0.0302	0.0321	0.0395	0.0395	0.0541
	R2	0.8226	0.8015	0.8415	0.8415	0.5322
	Loss (MAPE)	5.5425	5.8404	6.5404	6.5404	5.8301
Test	Size of Test dataset	343	343	343	343	343
	MSE	0.0012	0.0012	0.0023	0.0023	0.0051
	MAE	0.0264	0.0264	0.0351	0.0351	0.0558
	R2	0.8226	0.8015	0.828	0.828	0.5322
	Loss (MAPE)	4.1047	4.1047	6.13877	6.13877	10.3141
Hyperparameters	Learning Rate	1.00E-03	1.00E-03	1.00E-03	1.00E-03	1.00E-03
	Decay	1e3/100	1e3/100	1e3/100	1e3/100	1e3/100
	Optimizer	adam	adam	adam	adam	adam
	Batch Size	256	256	256	256	256

Table 1. Experimental results for prediction of Vote average using a multi modal architecture

All the experiments in the table above have been run over 50 epochs and the plots of metrics against the number of epochs are provided in the following table - Table 2:

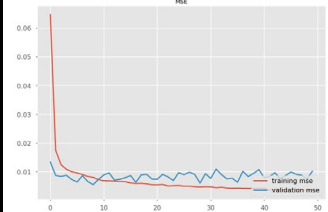
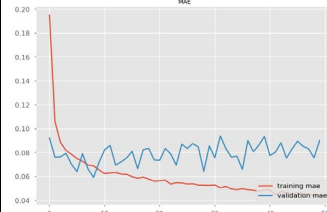
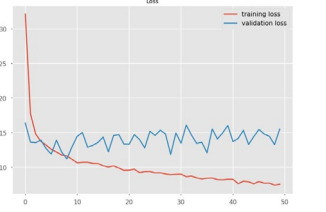
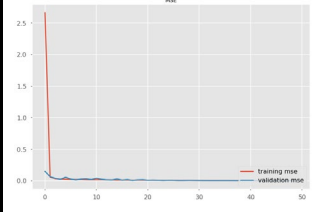
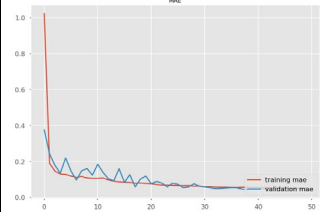
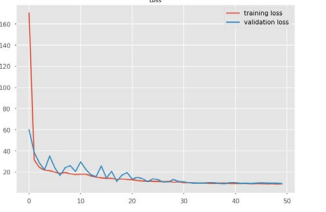
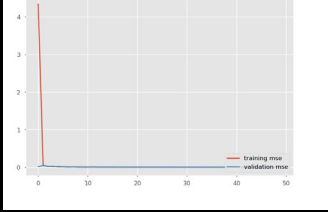
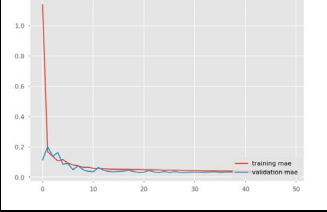
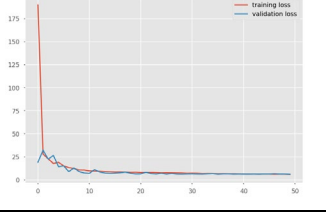
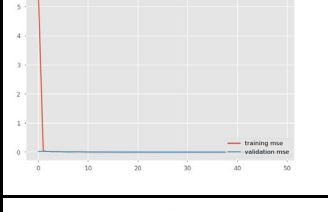
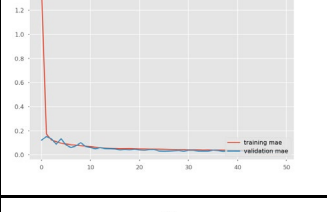
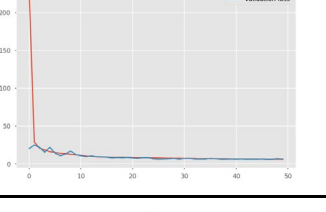
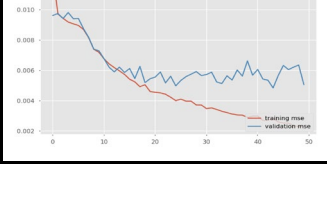
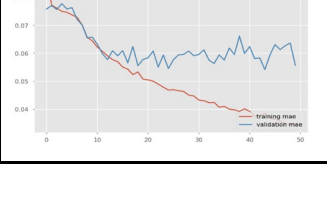
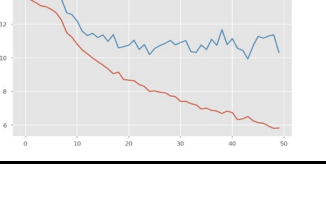
	MSE	MAE	LOSS (MAPE)
Handcrafted model			
VGG16			
VGG19			
ResNet50			
Algebraic sigmoidal activation			

Table 2. Loss (MAPE), MSE, MAE plots for prediction of Vote average using a multi modal architecture

- Various industry-standard CNN models like VGG16, VGG19, and ResNet50 were used as a foundation (Poster (Images)) to conduct different experiments on the data for building a viable model
- The output of the model architecture was a numeric score (vote average)
- The metrics used for evaluating the models: MSE (mean squared error), MAE (Mean absolute error), R2
- The MAPE was used as the Loss function for the experiments
- Hand-coded model (Model architecture in the previous slide) performed best in terms of the selected metrics. In this problem statement, simpler architecture performed better compared to more complex models like VGG16, VGG19, ResNet50
- The last column used an algebraic activation function in the output layer to demonstrate the feasibility of using algebraic sigmoidal functions as a basis for building an optimizer in future work). The model not only converged at just 50 epochs but the mse,mae and loss were not fluctuating much. In the future, we can experiment with algebraic sigmoid functions throughout the network as a basis for building an Optimizer cost function (higher-order polynomials)

The simulation models using single stream (Plot only/ Poster only/ categorical variables only) is provided in the following table- Table 3 below (run only for 20 epochs). As it can be seen from the Loss, these are viable models to use as simulation models during various stages of the production process when only these modalities of data may be available.

Plot Only	Dataset Size	Epochs 20	Poster Only	Dataset Size	Epochs 20	Categorical Only	Dataset Size	Epochs 20				
Training MSE	3261	0.0014	Training MSE	3261	0.0067	Training MSE	3261	0.0014				
Training MAE	3261	0.0037	Training MAE	3261	0.0094	Training MAE	3261	0.0053				
Training Loss	3261	5.8558	Training Loss	3261	4.1282	Training Loss	3261	4.4738				
Test MSE	907	0.0025	Test MSE	907	0.0074	Test MSE	907	0.0064				
Test MAE	907	0.0057	Test MAE	907	0.0087	Test MAE	907	0.0087				
Test Loss	907	4.2569	Test Loss	907	2.3675	Test Loss	907	4.7778				
Validation MSE	363	0.0137	Validation MSE	363	0.0108	Validation MSE	363	0.0564				
Validation MAE	363	0.0976	Validation MAE	363	0.0868	Validation MAE	363	0.0734				
Validation R2	363	0.7317	Validation R2	363	0.6943	Validation R2	363	-0.84				
Validation Loss	363	4.7177	Validation Loss	363	1.6217	Validation Loss	363	5.8231				

Learning Rate	1e-3,0.001
Decay	1e-3/200,1e-5
Optimizer	Adam
Loss	Mean_absolute_percentage_error
Metrics	MAE,MSE

Table 3. Experimental results for prediction of Vote average using single data streams (Plot, Posters, metadata)

We would like to extend this proven theory of multiple influencing streams stacked together as input features and have the ability to select influencing features on-demand and have the ability to penalize non-influencing features weights.

Conclusion

The primary reason for the viability of the framework (as shown in Figure 1) in the media industry is the very nature of the multi-step content creation process. While previous works by previous authors were focused on predicting the success of contents, our approach takes an overall process optimization view of the content production process. The approach outlined in this document can lead to the development of a virtual assistant to the content production process providing optimization inputs along the way that are based on near real-time data.

The challenge that we ran into is the unavailability of public data across the process chain. We firmly believe that if we can collaborate with media organizations, we will be able to operationalize this proposed method in the media space.

Future Work/ Research Direction

- Train individual models using larger industries datasets: The future work/research direction will be

to apply this framework to actual datasets modalities built across the content lifecycle. This will require industry participation and sharing of data to successfully implement and measure the success of the framework.

- Build a generalized model (Similar to GPT and BERT): Some deep learning models based on modalities available in the public domain (like video promotion of contents, teasers, posters, etc.)
- Train the individual models using other algebraic sigmoidal functions: DNN ReLU as a composition of piecewise affine and Alexandre Megretski variation of sigmoid function [20]. The experimental results were demonstrated using algebraic sigmoidal functions in the output layer. Future work needs to expand this to convert all the activation functions of the network to algebraic functions and build a higher-order polynomial optimizer for a Model-Based Predictive Control.
- Explainability framework to explain the factors that cause the predicted scores to move up and down based on feature importance.

References

- [1] Prediction of a Movies Success from Plot Summaries Using Deep Learning Models 2019
- [2] Movie-scope Large-scale Analysis of Movies using Multiple Modalities 2019
- [3] Improving Box Office Result Predictions for Movies Using Consumer-Centric Models
- [4] Movie Success Rate Prediction Using Robust Classifier 2019
- [5] Violence Rating Prediction from Movie Scripts
- [6] Predicting and ranking box office revenue of movies based on big data
- [7] Predicting the performance of TV series through textual and network analysis: The case of Big Bang Theory
- [8] Annie Mueller, "Investopedia: Why movies cost so much to make", April 28th, 2020
- [9] TMDB 5000 Movies dataset, Kaggle
- [10] TMDB, Movie Poster
- [11] Jeffrey Pennington, Richard Socher, Christopher D. Manning, "Glove: Global Vectors for Word Representation." NLP Stanford
- [12] Bharat Raman, Kaggle, "Predicting Movie ratings through Machine Learning."
- [13] Jeffrey Pennington, Richard Socher, Christopher D. Manning, "Glove: Global Vectors for Word Representation." NLP Stanford
- [14] Kaylie Leung, Medium publication, "learning paradigms in neural-networks"
- [15] Delving Deep into Rectifiers Surpassing Human-Level Performance on ImageNet Classification Authors: Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun
- [16] What size net gives valid generalization- Author Eric B Baum, David Haussler
- [17] Comprehending AI model decisions with SHAPley intuitions - Part 1, Author Shashank Sahoo (Medium Blog)
- [18] SHAP (SHapley Additive exPlanations), Christopher Molnar, Github Page
- [19] Pensions Investment Mortgage Insurance, Thought Leadership blog
- [20] Linear Algebra and Functional Analysis for Signals and Systems, Author: Alexandre Megretski, John Wyatt, September 15th, 2009
- [21] Lyapunov Stability Theory, Author: ScienceDirect Journal
- [22] Characterization of a Class of Sigmoid Functions with Applications to Neural Networks, Authors: Anil Ravindran Menon, Kishan Mehrotra, Chilukuri K. Mohan, Sanjay Ranka
- [23] S. Kalyan, A. Tirkey, A. Patra, S. Kumar, P. Singh, and A. Addanki, "Deep Learning Approach to Predicting the Success of Content," SMPTE 2020 Annual Technical Conference and Exhibition, 2020, pp. 1-12, DOI: 10.5594/M001914.

[24]<https://www.innovyze.com/en-us/blog/ai-for-industrial-process-control-part-2-model-predictive-control-deep-dive>