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

Authors: Suman Kalyan, Angshuman Patra, Sujay Kumar, Alex Addanki, Shashank Sahoo

Movie production is a complex multi-step process...

- Movie production is a complex, multi-step and lengthy process
- Although a large industry, ~65 B USD, the outcome are not predictable
- 7 out 10 movies/content do not give good returns on Investment

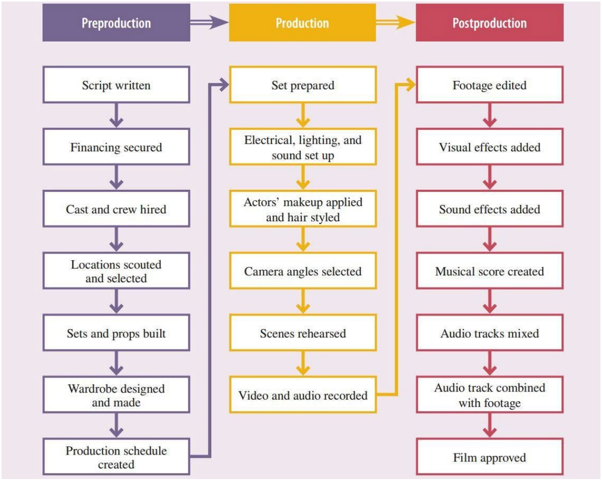
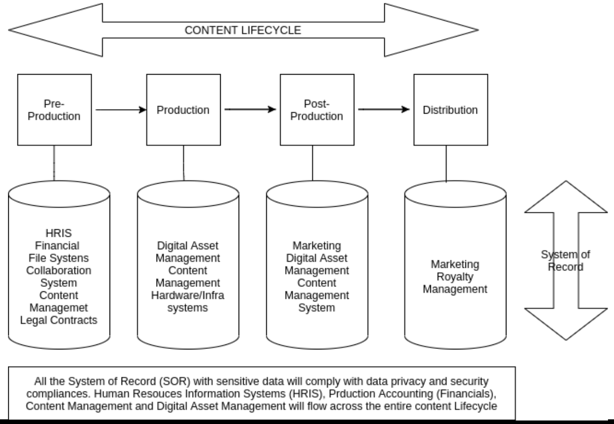


Figure 1: Source Credit: India Shoots website

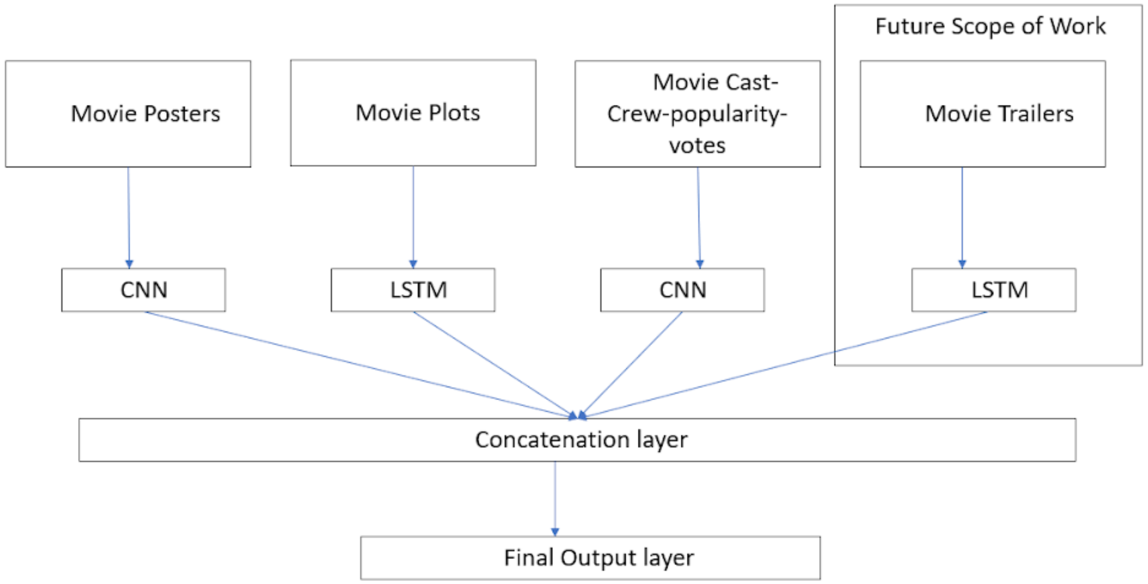


...each process step producing a ton of data.

A lot of data is generated and available during 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

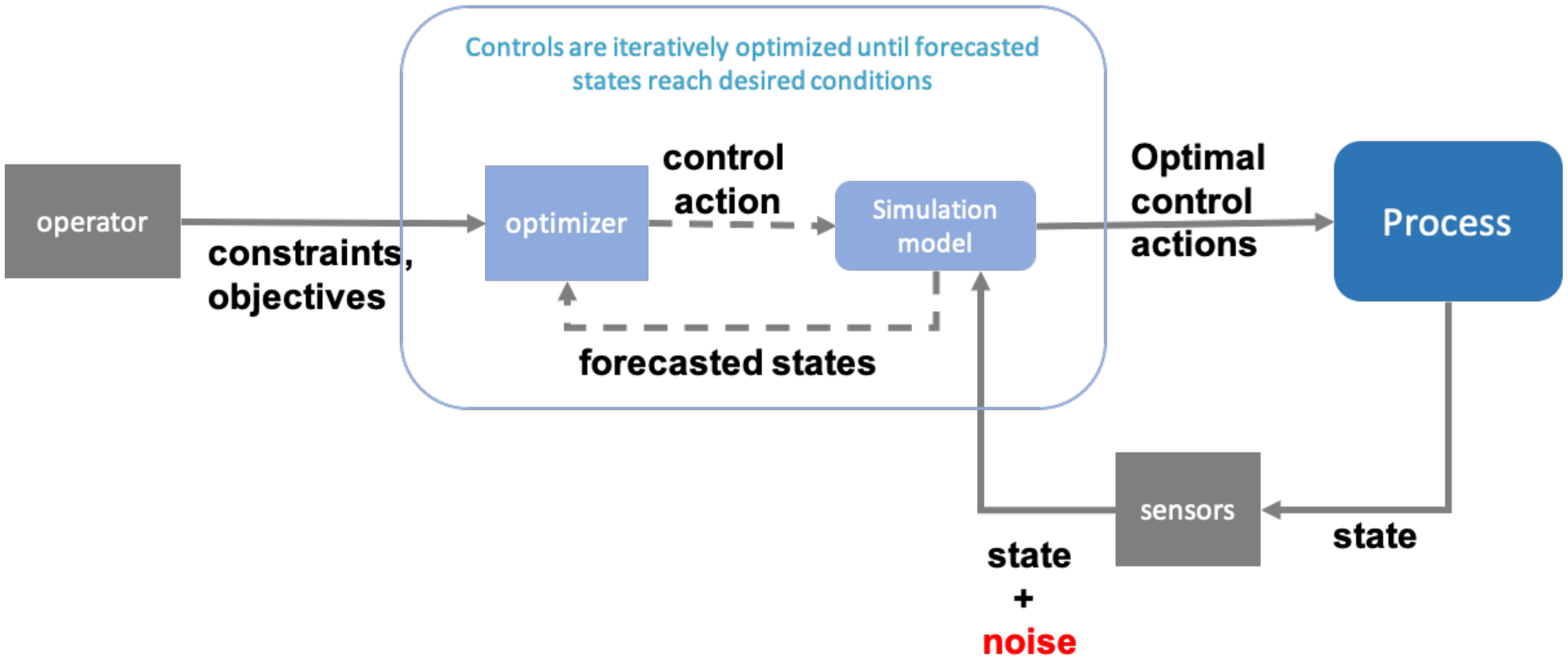
We showed previously that deep learning can help with performance prediction



- Data set size: 4500 Movies [9]
- Movie posters[10]
- Cast, Crew : from Kaggle
- Multi modal network
- Outcome variable: Vote Average

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.

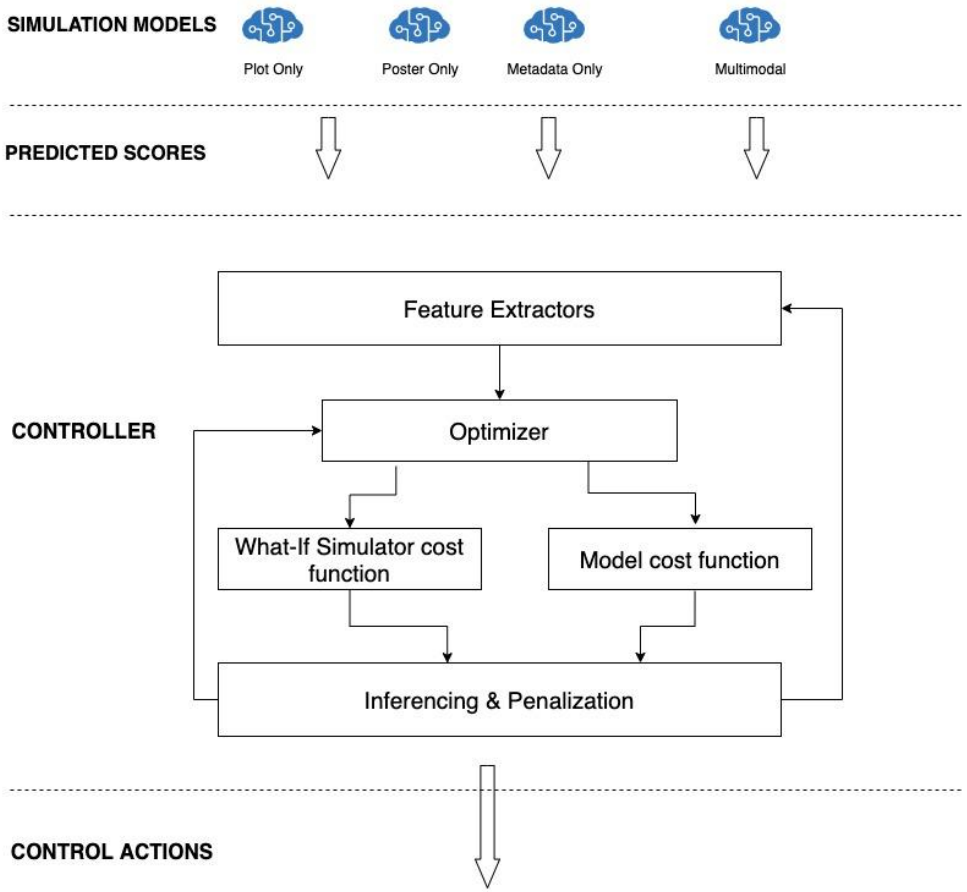
Manufacturing industry uses continuous monitoring for decision making



Schematic for an AI based Industrial process control

Typically manufacturing processes have a continuous monitoring and control loop that looks at predictions from the **simulation model** at each stage of the production process and uses that data to guide **control actions** in the upcoming steps of the process.
This approach is know as Model Based Predictive Controls (**MBPC**)

Our MBPC approach defined



The various components of the model-based predictive control architecture are as follows:

a. **<u>Simulation Models</u>**: Models that can be used as the starting point for the MBPC approach and support different modalities of data

b. **<u>Controller:</u>** Computes the control actions based on the predicted scores and the closing the gap with target scores

c. **<u>Control Actions:</u>** The actions that can be taken (with constrains considered)

Our experimental setup for MBPC approach in movie production

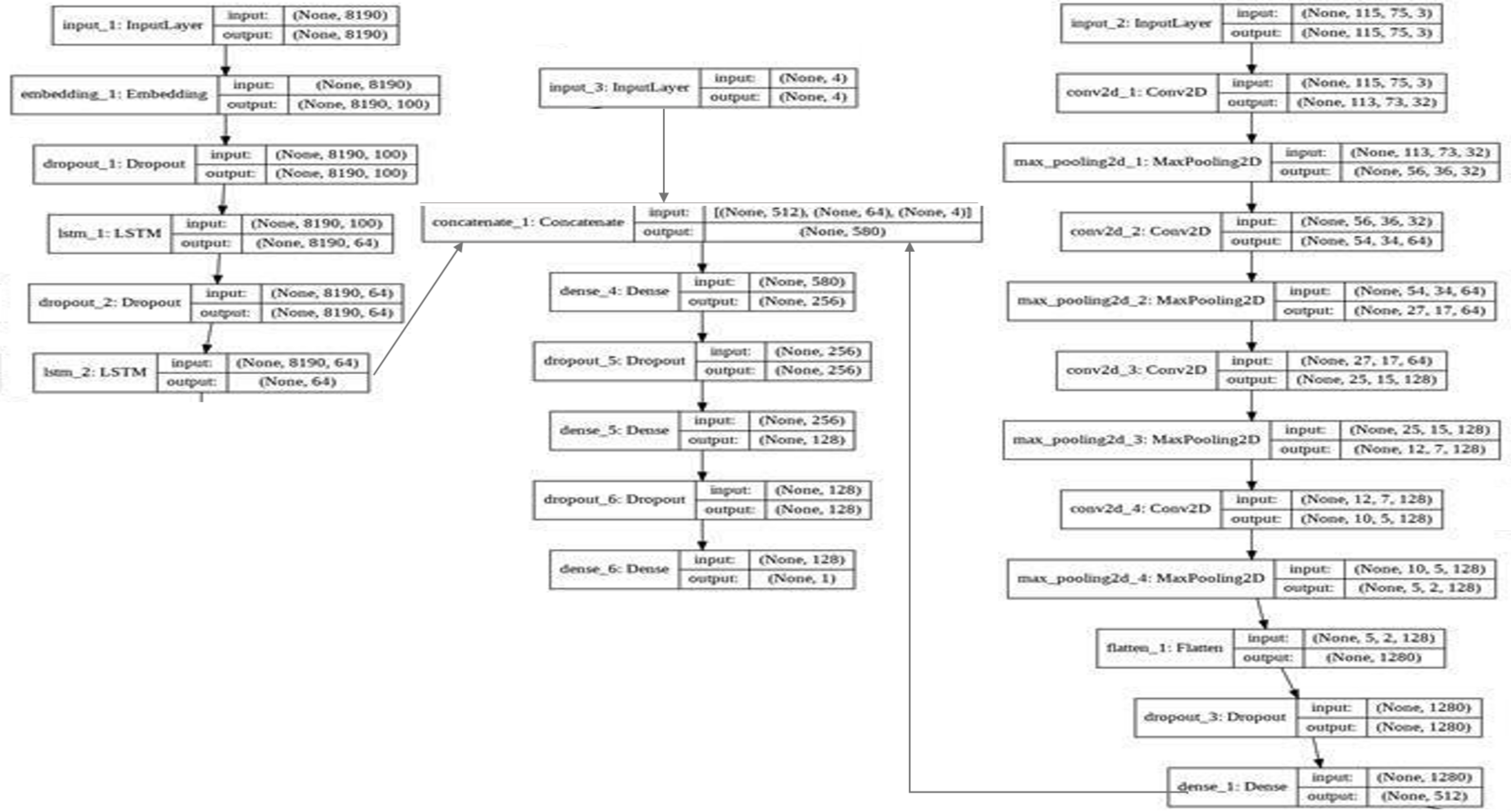
- The researchers used 3 data sources for a corpus of 4500 movies and created a datastore on Amazon S3 bucket for running experiments
  - Movie plots
  - Posters
  - Movie metadata(Genre, Cast, crew, budget, Producer, Directors, etc from Kaggle)
- The choice of data was primarily driven by what was available in the public domain (Kaggle, Wiki, TMDB)
- AWS Sagemaker notebooks for running the experiments
- Frameworks Used: Tensorflow 2.x, Keras, GLOVE, VGG16/19, ResNet
- Source code: Python 3.x (GIT repository:https://github.com/alex-addanki/MovieContentController.git)
- **Current setup and challenges**: Given the limited data availability, the best we could do within a limited time and budget was to validate the viability of **Simulation Models** and to do initial validation of using algebraic sigmoidal activation functions as a basis for building a polynomial Optimizer in future work.

Model Architecture for hand-coded model

Input _1
Unstructured Text: Plot

Input _3
Numeric: Metadata

Input _2
Images: posters



Multi modal architecture:

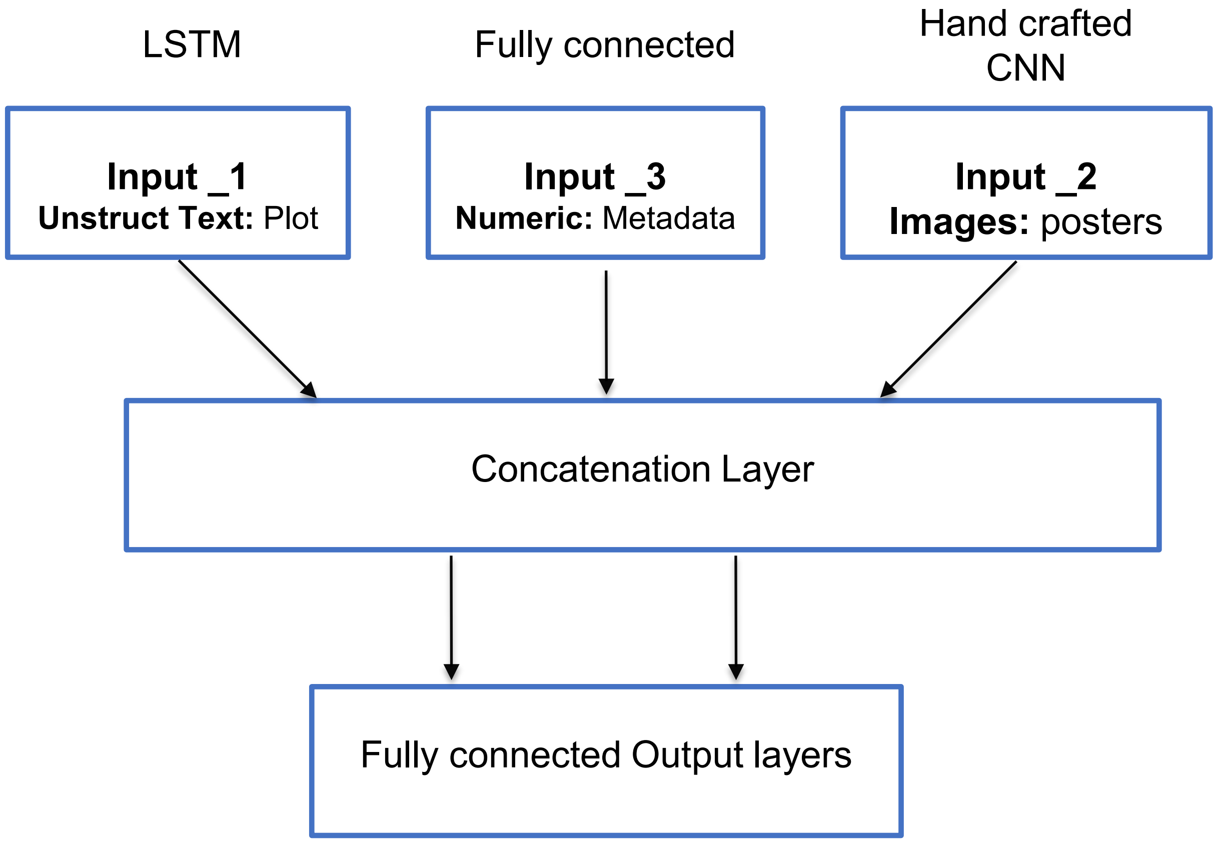
The target/outcome variable is a numeric variable.

Vote average (based on votes polled by the audience)

The model architecture supports three distinct inputs:

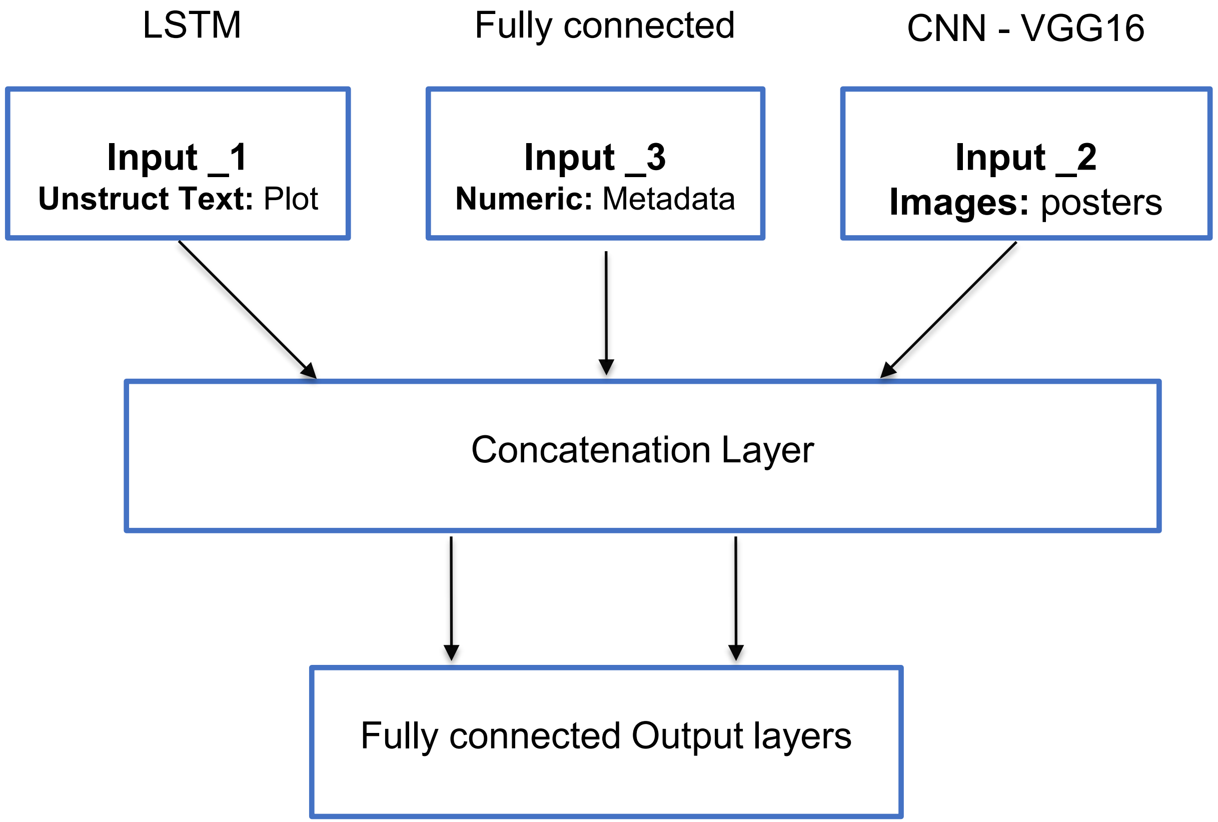
a. **Input_1**: Movie plot (unstructured text)
b. **Input_2:** Poster (Images)
c. **Input_3**: Numerical features based on cast, crew, production companies

Experimental results for the Simulation model (hand crafted)



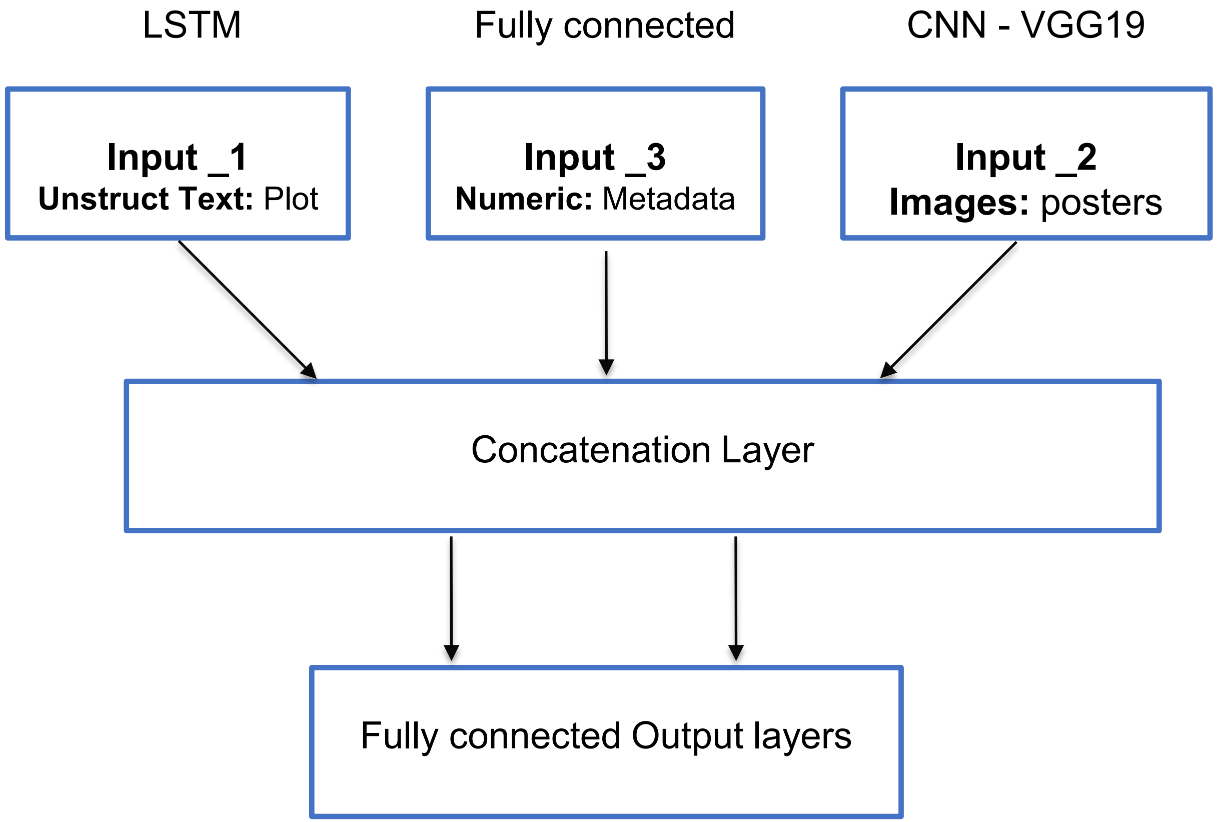
	Metric	Hand crafted model
Training	Size of training dataset	3357
	MSE	0.0017
	MAE	0.0302
	R2	0.8226
	Loss	5.5425
Cross validation	Size of validation dataset	1343
	MSE	0.0017
	MAE	0.0302
	R2	0.8226
	Loss	5.5425
Test	Size of Test dataset	343
	MSE	0.0012
	MAE	0.0264
	R2	0.8226
	Loss	4.1047
Hyperparameters	Learning Rate	1.00e-03
	Decay	1e3/100
	Optimizer	adam
	Batch Size	256

Experimental results for the Simulation model (VGG16)



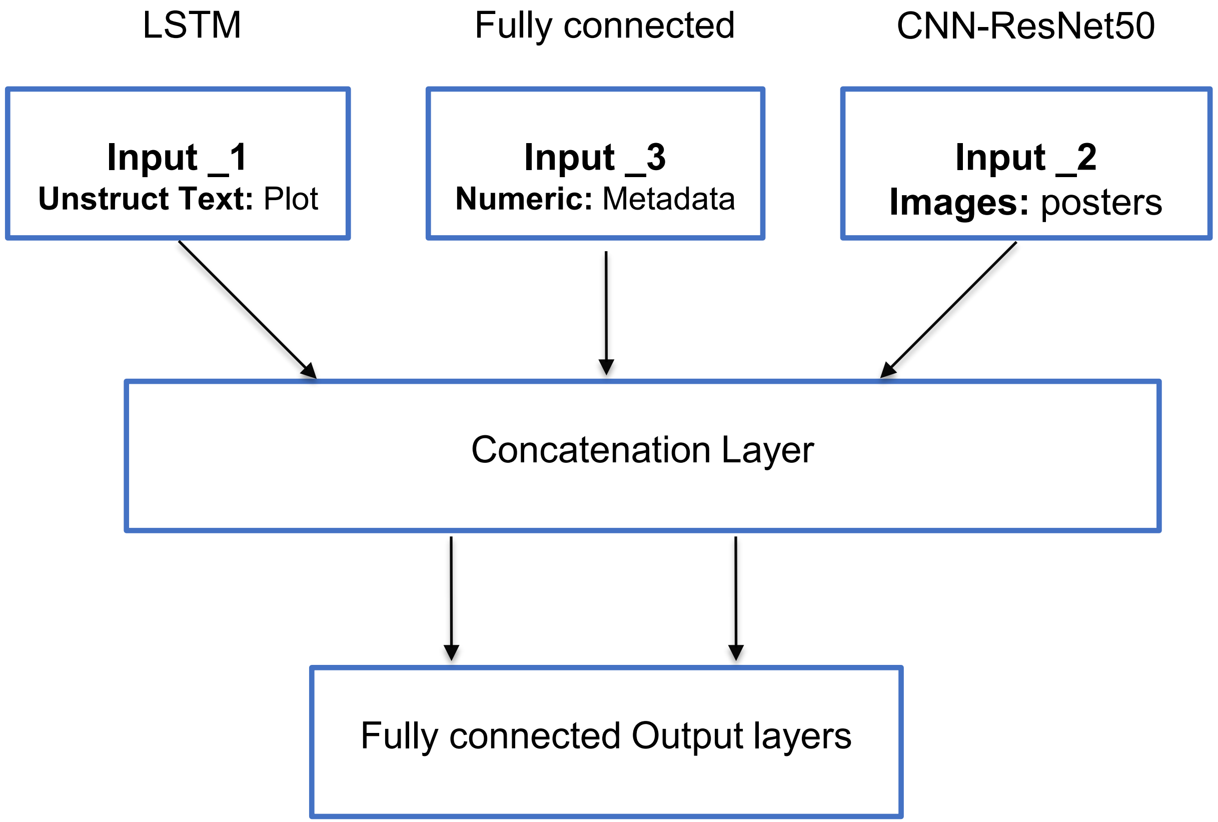
	Metric	VGG16
Training	Size of training dataset	3357
	MSE	0.0019
	MAE	0.0321
	R2	0.8015
	Loss	5.8404
Cross validation	Size of validation dataset	1343
	MSE	0.0019
	MAE	0.0321
	R2	0.8015
	Loss	5.8404
Test	Size of Test dataset	343
	MSE	0.0012
	MAE	0.0264
	R2	0.8015
	Loss	4.1047
Hyperparameters	Learning Rate	1.00e-03
	Decay	1e3/100
	Optimizer	adam
	Batch Size	256

Experimental results for the Simulation model (VGG19)



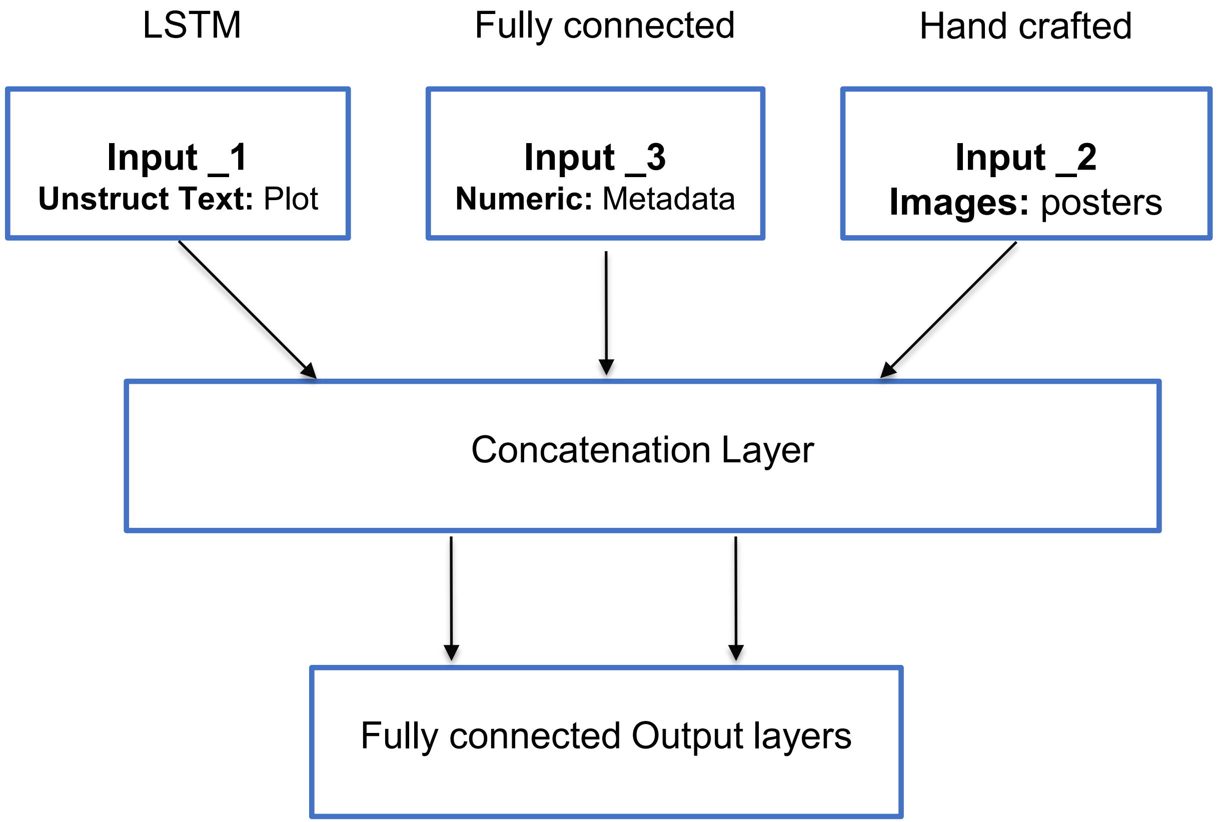
	Metric	VGG19
Training	Size of training dataset	3357
	MSE	0.0026
	MAE	0.0395
	R2	0.8415
	Loss	6.5404
Cross validation	Size of validation dataset	1343
	MSE	0.0026
	MAE	0.0395
	R2	0.8415
	Loss	6.5404
Test	Size of Test dataset	343
	MSE	0.0023
	MAE	0.0351
	R2	0.8280
	Loss	6.13877
Hyperparameters	Learning Rate	1.00e-03
	Decay	1e3/100
	Optimizer	adam
	Batch Size	256

Experimental results for the Simulation model (ResNet50)



	Metric	ResNet50
Training	Size of training dataset	3357
	MSE	0.0026
	MAE	0.0395
	R2	0.8415
	Loss	6.5404
Cross validation	Size of validation dataset	1343
	MSE	0.0026
	MAE	0.0395
	R2	0.8415
	Loss	6.5404
Test	Size of Test dataset	343
	MSE	0.0023
	MAE	0.0351
	R2	0.8280
	Loss	6.13877
Hyperparameters	Learning Rate	1.00e-03
	Decay	1e3/100
	Optimizer	adam
	Batch Size	256

Experimental results for the Simulation model (Algebraic sigmoid activations)



	Metric	Algebraic Sigmoid Activation
Training	Size of training dataset	3357
	MSE	0.0045
	MAE	0.0541
	R2	0.5322
	Loss	5.8301
Cross validation	Size of validation dataset	1343
	MSE	0.0045
	MAE	0.0541
	R2	0.5322
	Loss	5.8301
Test	Size of Test dataset	
	MSE	0.0051
	MAE	0.0558
	R2	0.5322
	Loss	10.3141
Hyperparameters	Learning Rate	1.00e-03
	Decay	1e3/100
	Optimizer	adam
	Batch Size	256

Summary: Results & Discussions

- Various Industry standard CNN models like VGG16, VGG19 and ResNet50 were used as a foundation (Input_2: 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)

Our Vision

- Move from point predictions to continuous prediction process with an Optimizer
- A potential use case for this can be building a virtual-assistant for key stakeholders in the movie production process that can assist in data driven decision making
- To realize the vision, we would love to partner with an organization to refine the approach with detailed data across the content lifecycle.

We plan to expand our work as we move forward

- Train individual models using larger and complete datasets
- Build a generalized model (Similar to GPT and BERT)
- Train the individual models using other algebraic sigmoidal functions
- Build Explainability framework

THANK YOU

Can continuous Model Based Predictive Controls (MBPC) be applied to movie productions?

We can look at the movie production process as set of process steps

- **Process**: 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, 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

- **MBPC** allows Media houses to run their **production processes more efficiently** by operating much closer to constraints than would be possible with conventional planning and pivoting rapidly when needed
- **Control actions** are optimized to achieve a desired outcome, such as changing the director, choosing a different poster or changing a few members from the production crew or changing the plot of the movie
- **Constraints** may vary based on stage of a production process. e.g: Constraints on Budget, options for supporting actors, better quality of 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 (MPC) operates by performing dynamic, real-time optimization to generate control actions that are compliant with user-specified constraints.