

# ML Project

# Final Presentation

**Analysing The Effect of Different Filmmaking choices  
on IMDB rating**

- *Madhav Rangaiah (2019251)*
- *Ishaan Jindal (2019246)*
- *Parimal Shivale (2019068)*

# Problem Statement:

We will study the effect of different parameters of a movie such as the cast, metascore, the actors, etc. to predict how well it will do and provide a rating based on it.

## Dataset:

Initially we had the dataset downloaded from kaggle. But later we realised that we would need additional column pertaining to poster. So we used the omdb API to get these details for each movie and added those details to our modified dataset.

Dataset: [Link](#)

Dataset Dimensions = 1000 X 12

Target class :- Rating

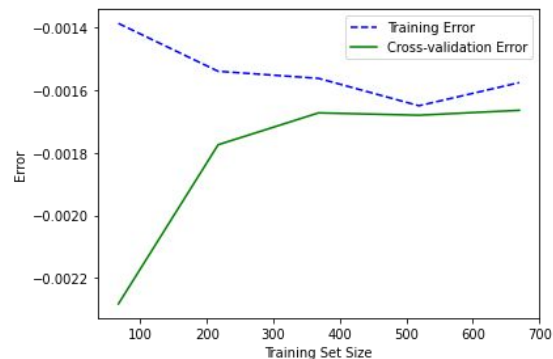


# Progress Before Interim Review

**Preprocessing:** We used target encoding on nominal categorical values for lesser number of features and then normalized the numerical values from 0 to 1.

**Baseline Model:** Our basic approach was to take all the features, excluding poster and summary of the movie, and fit them in a linear regression model to predict the IMDB rating of a movie.

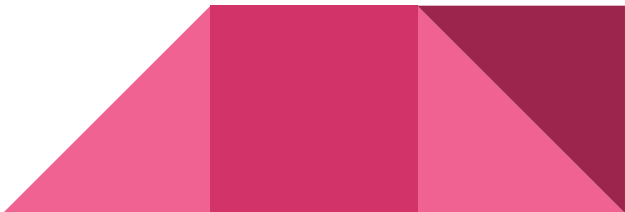
**Error Analysis:** From our learning curve we can see that the model is training correctly and we get an  $R^2$  score of 0.861



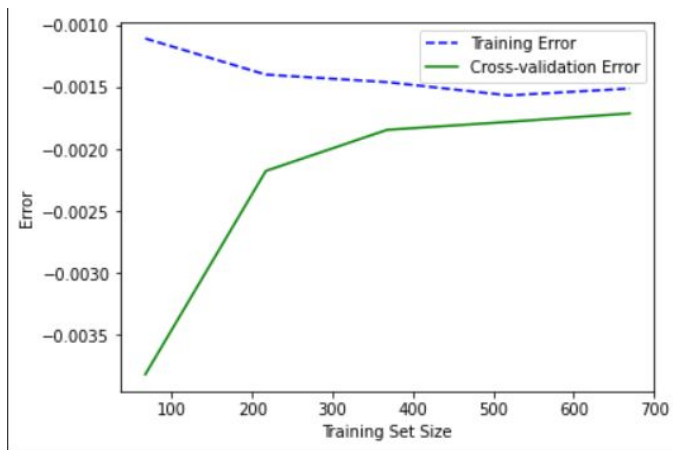
# Progress after Interim Review

**Improving baseline model:** We tried techniques such as K-Fold Cross Validation, selecting K best features, L2 regularization to improve our R2 score. We also tried other regression models such as Decision Tree Regressor, SVR, KNN regressor, and also tried stacking these models together. In the end, Linear Regression with K-Fold and L2 Regularization gave the best R2 score (0.923).

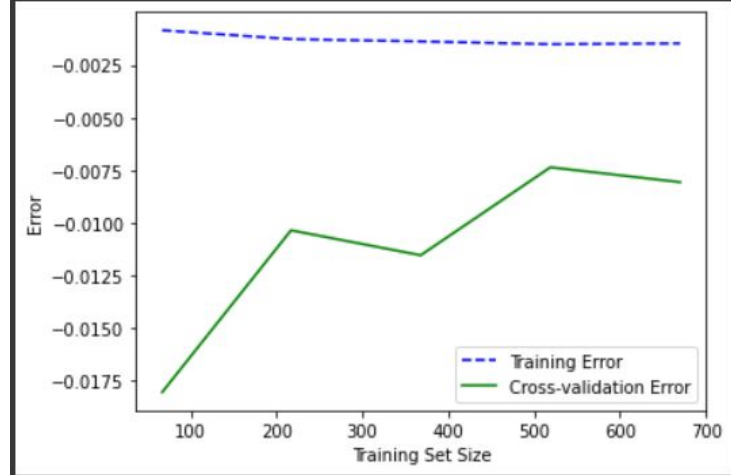
**Polynomial Regression:** To further improve our accuracy, we extended our linear model with 2° and 3° features. On adding 3° features, our model was overfitting so we added L2 regularization and got an R2 score of 0.925.



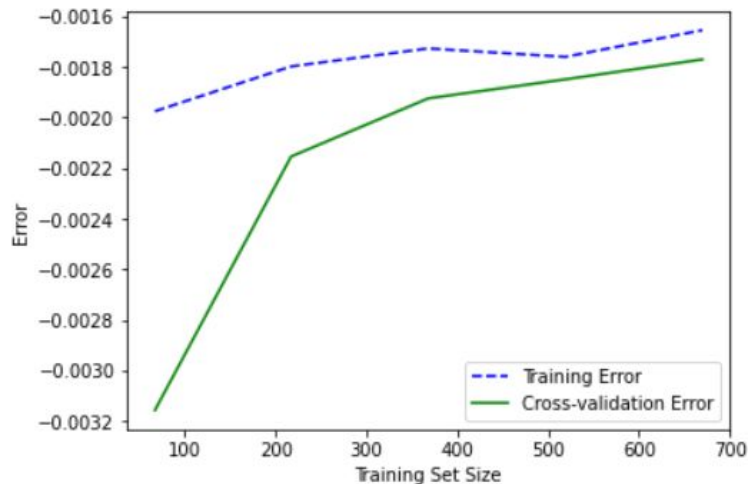
**Learning Curve for 2° features**



**Learning Curve for 3° features**



**Learning Curve for 3° features With L2 Regularization**

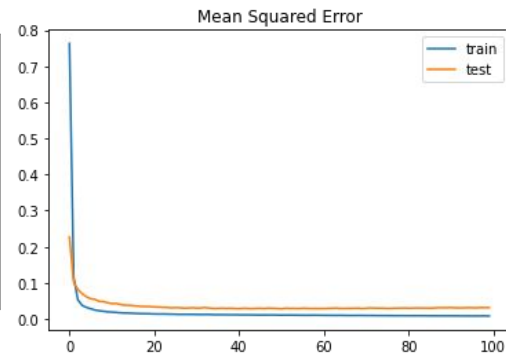
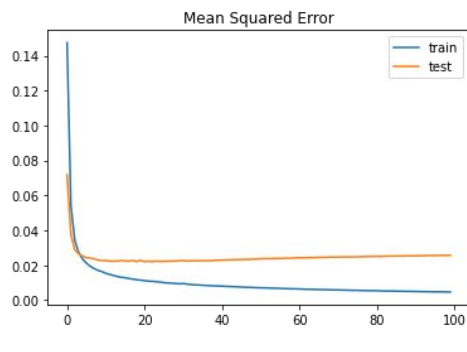
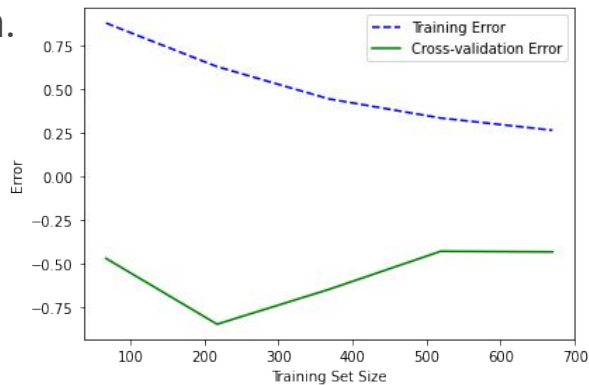


## Predicting Rating from Poster:

CNN as a classifier: Here we divided the rating of a movie into 10 categorical classes and trained the data as CNN. We first converted the poster images to NumPy arrays using the PIL library and did padding to make all the images of the same size (488 X 300 X 3). Then normalized image as input was passed to model with 4 hidden layers. We got an accuracy of about 45% which wasn't a very motivating score, so we discarded the model.



**Predicting Rating from Movie Description:** Using POS tagging we got all the nouns, adjectives and verbs from all descriptions and narrowed them down to the ones occurring more than 10 times. We got 155 words and applied one hot encoding to get 155 features for each movie. On applying linear regression, the model was underfitting even after L2 regularization.



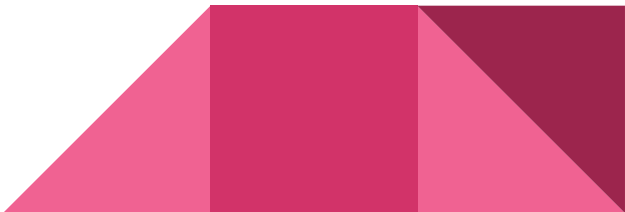
So we applied an ANN model with 3 hidden layers with 50 neurons and 'relu' activation function. The model was overfitting, so we reduced the hidden layers to 1 and increased the training size.

# Results and Analysis

For our linear models we got the following metrics:

Models	R2 Score	MSE
Linear Regression	0.8617	0.00198
Decision Tree Regressor	0.8615	0.00189
KNN Regressor	0.8229	0.00237
Support Vector Regressor	0.8918	0.00188
Ensemble learning	0.9215	0.00118
Linear Regression with K-Fold and L2	0.9237	0.00121
Linear Regression with 2 deg features	0.9246	0.00122
Linear Regression with 3 deg features	0.9253	0.00124

For the ANN model, the train MSE comes out to be 0.013 and the validation MSE comes out to be 0.024.





We can see which attributes have higher magnitude of weights in the models to see which of them affect the rating more. We can see the weights below:

```
Rank : 0.0
Runtime : -0.012465347760933332
Revenue : -0.055645006153890494
Year : -0.011532937561028892
Metascore : 0.09052975821972678
Title : 0.0
Votes : 0.21017495229359873
Director : 0.39808727943438316
Genre : 0.02317759828915819
Description : 0.0
Actors : 0.5546688618737018
```

```
government: 0.10486566948005929
been: 0.10413299944251776
escape: 0.1035074594966136
humanity: 0.10221482187509537
save: 0.0988212610501796
```



# Individual Contributions

**Madhav:** Pre-Processing, Baseline Linear Regression, Improving Regression techniques, ANN for Movie Description

**Ishaan:** Baseline Linear Regression, Polynomial Regression, K Means, CNN(To be delivered)

**Parimal:** EDA, Polynomial Regression

