PREDICTING IMDB RATING

PHASE 1 : DATA ANALYSIS

* Crawled imdb data is loaded into pandas dataframe df
* Dataset’s head is observed.
* All the columns present in dataset are printed, to analyse them.
* Any null value is checked using isna().sum()
* Observation: Genre 4, Writer 3, Star 4, Star 5 are all null. Thus they are unrequired features.
* Datatypes of each column is then checked. Observation: Most of them are object. They need to be cleaned and converted to int or float
* Then, Genre 1, Genre 2 and Genre 3’s value counts are observed.
* It’s observed that many genre is being repeated in these three feature columns.
* To do: I applied one-hot encoding in these columns and then all common genre are being merged to a single column by iterating through them. I’ve directed them as algorithm in python notebook
* Next, Writer columns are analysed.
* Observation: We got too many unique writers name including all 2 Writer columns. I’ve omitted Writer names for model training.
* We can use top 20-30 writers to train model(I’ve omitted this apptoach here)
* I observed similar thing in Star 1, Star 2 and Star 3. Again we have too many unique Star values. So, i ignore Star list as of now.
* Plot Keywords List is also ignored because of too many values. A different approach could be used for this.
* Next, censor board rating is analysed. It has 7 unique rating and 1 banned movie.
* I’ve applied one-hot encoding for censor board rating. And this can be a good feature for Imdb rating prediction.
* Next, Budget, Gross USA, Cumulative worlwide Gross are important Summary details. These columns needs to be cleaned. As we will see in PHASE 2.
* Number of reviewers are also important feature. We will clean them in PHASE 2.
* Length of the movie also determines an important factor. People may prefer less duration movie, etc.
* We can divide length into intervals of 1h,2h,3h,4h,5h

PHASE 2: DATA CLEANING

* Genre: As stated in PHASE1, I’ve encoded unique genre in Genre 1, Genre 2 and Genre 3. Then all the unique genres are merged and we get set of features named “new\_<genre\_name>”
* Budget: I’ve used regex to remove “(estimated)”, “$”, and “,” signs from Budget column. As Budget indicated simple integer value, extracting just the integers is an advantage.
* Gross USA: Gross USA column contains date. I’ve removed date using regex and also “$” and “,” are removed.
* Cumulative worlwide Gross: Similar approach as Gross USA is applied to remove date ftom this column and get a simple integer value
* Number of Reviewers: Unwanted “,” are removed to get simple number of reviewers
* Length of the movie: All movies between 1h to 2h are grouped as 1h, 2h to 3h are grouped as 2h, 3h to 4h are grouped as 3h, 4h to 5h are grouped as 4h and 5h to 6h are grouped as 5h.
* Year: Year of release is kept as it is

PHASE 3: TRAINING MODEL

* I’ve splited dataset into training-test-validation set as 70-15-15 percentage.
* Validation set is important while training
* 1st Approach: I have used simple linear regression on training data set.
* Result: I got 1.2%(approx) MAPE in test data.
* Although there’s no sign of overfitting in the model. I simply tried using regression to check the model. But it resulted in increase in error rate. MAPE was 1.7%(approx).
* 2nd Approach: Using Neural Network
* I’ve always observed NN to outperform many ML algorithm.
* Although we don’t have many dataset in this example but seeing how NN works is no harm.
* NN Architechture:
* Input layer: (?,38) where 38 is number of features
* 2 hidden layers with 16 and 8 nodes respectively.
* And a linear output layer.
* Total params: 769
* loss: mean\_absolute\_percentage\_error
* metrics is same as loss: mape
* Result: And as expected NN gave an error of 0.6%(approax)
* Thus, NN outperformed normal Linear Regression
* To Do: NN could be used further with some hyperparamter training to get less error
* Also Linear regression with further hyperparameter tuning and including L1, L2 regularization could be used.
* To further improve the model, features like story summary, writer, star, etc. Could be used