Problem statement : Audience movie rating prediction:

Columns:

- 'Movie_title' : Title of the movie
- 'Movie_info' : Storyline/plot of the movie
- 'Critics_consensus': context/ text based review by critics
- 'Rating': R/ PG/ NR.. restrictiveness to the movie
- 'Genre': Combination of genres as action, comedy
- 'Directors': Directors of the movie
- 'Writers': Story is written by
- 'Cast': Main actors who took part in the movie
- 'In_theaters_date': Theatre release date
  'On_streaming_date': OTT release date
- 'Runtime_in_minutes': Duration of movie
- 'Studio_name': studio produces
- 'Tomatometer_status' : movie categorized as fresh, rotten..
- 'Tomatometer_rating': Critic rating in numbers
- 'Tomatometer_count': Number of critics rated
- 'Audience_rating': Target variable, movie rating by audience

Data Analysis:

- Audience rating is ***discrete*** variable from 0-100, instead of continuous variable
- There are duplicates in the data (with respect to movie title) but they're having different theatre dates which justifies duplication.
- There is missing data in columns as : genre, audience rating
  They are removed since they've too low count
- The missing data in runtime has been imputed by median to avoid the impact of outliers
- There is no theater date for some movies, which had an OTT date. Hence, equated the theatre date to stream date (took only theatre year as feature due to multi collinearity with stream date and avoiding too granular info on date level)
- Identified and removed the outliers in Runtime using ***Z-score***
- Transformed the genre and rating columns after cleaning
- For genre, identified unique genres and used ***one-hot encoding*** type encoding for the genre as : if a movie had action, comedy – labelled as 1,1
- Since there is order in the data for rating, tomatometer status, I've used ***ordinal encoding*** instead of label encoding:
  ['Rotten', 'Certified Fresh', 'Fresh'] —> increased movie recommendation
  ['PG', 'R', 'NR', 'G', 'PG-13', 'NC17'] —-> increased restriction on audience
- For scaling,
  I've used ***min-max scaler*** for - Tomatometer count, theatre year (since there is bound)
  For runtime I've used ***standardscaler*** to avoid impact of outliers (there is skewness in data)

- The dataset had too many class imbalances which is impacting the model performance heavily. Hence I’ve used ***Synthetic Minority Oversampling Technique*** for oversampling the minority classes. This has improved my model performance.
- For the feature selection:
  Manual picking – I’ve used ***VIF*** score, ***perm_importance*** for identifying the necessity of feature compared to others
  PCA – I’ve also tried PCA technique for reducing dimensionality

Data preprocessing (pipeline):

- Removing duplicates
- Removing some missing data
- Removing outliers (z-score)
- Imputing missing data for relevant fields as runtime, theatre date
- Transforming the data as genre, rating (for relevant values)
- Adding new relevant columns as theatre year..
- Encoding categorical values (one - hot encoding, ordinal encoding)
- Scaling the data (standard scaler , min-max scaler)
- Handling Imbalances (SMOTE)
- Feature selection (PCA, manual selection)

Model analysis :

Without semantic analysis :

- This problem can be interpreted as classification task – into 101 classes
  Or regression task (which is rounded and taken as classification result)
  Due to class imbalances present in the data and too many classes
  I’ve regression as core and it resulted in better results than classification
- ***Random forest regressor*** predicted best compared to others
- For feed forwards neural networks, regression based predictions (linear) are better than classification (softmax)

With Semantic analysis:

- I’ve taken movie info which conveys about the storyline for inference
- Rest of the text columns have following problems:
  Critics consensus : Lots of missing values (50%), relevant with critic rating (numerical)
  Director, cast, writer, studio : Lots of unique values ranging from 45%-75%, hence encoding based on class uniqueness will increase dimensionality with low pattern
- I’ve used sentence transformers*(all-MiniLM-L6-v2*) for encoding the movie info
- Then with the rest of the features, I’ve used PCA to reduce dimensionality
- Initially the model was being overfitted which has been controlled with early stopping, dropout layers.
- The final semantic model performed slightly better than the above feed forward neural networks but still they aren’t greater than optimal model (Random forest)
- This infers that movie info here have slightly lower correspondence than expected

Model:

- Regression model (using random forest, xgboost, Feedforward NN(linear))
- Classification model (Feedforward NN(soft max))
- Semantic model (sentence transformer: all-MiniLM-L6-v2,Feedforward NN & xgboost)

Metrics:

- Since we’re calculating 101 class identification exact accuracy won’t depict the exact model result. Hence used MAE for analyzing.
- Also due to more classes, imbalances tolerance based accuracy is also used : tolerance_3_accuracy = ((np.abs(y_pred - y_test) <= 3).sum() / len(y_test)) * 100

Analysis :

- The model that performed best compared to others is random forest regressor (using manual feature selection)
  Metrics :
  - ➔ MAE: 7.974329931972789
  - ➔ Rounded Accuracy: 0.14301658163265307
  - ➔ Tolerance Accuracy (±3): 35.76%
  - ➔ Tolerance Accuracy (±5): 48.37%

Contact details

Lashyanth
lashi7941@gmail.com