IMDb Rating Prediction System

SENG 474 Data Mining - University of Victoria
Nolan Kurylo, Kahvi Patel, Jayden Chan, Ahnaf Ahmed

MOVIE RATING: 9.9/10

Problem Description

IMDb is a website and database platform that stores information regarding every movie in history, including a rating on a 0-10 scale.

The IMDb Rating Prediction System proposes an extension of the IMDb platform to predict the ratings of movies that do not currently exist (...yet). By examining various attributes that describe the production of previous movies, a user can supply attributes about a movie that are real (or hypothetical), and receive an estimated rating value from the Prediction System.

IMDb

With a focus on Decision Trees and Deep Learning, we attempt to analyze the rating predictability of various regression and classification models.

Dataset & Preprocessing

Our dataset is sourced from Kaggle and contains all movies in the IMDb database with > 100 votes. Our label vector is weighted_avg_vote" which assigns each movie a rating.
The `top_actor`, `top_actor_gender` and `actor_age` features were added manually.

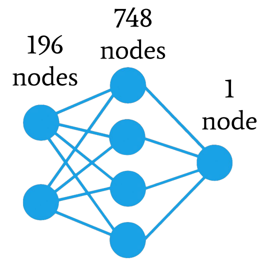
```
year                          Int64
genre                        object
duration                    float64
country                      object
language                     object
director                     object
writer                       object
production_company           object
budget                      float64
top_actor                    object
top_actor_gender             object
weighted_average_vote       float64
height                      float64
divorces                    float64
actor_age_at_release       category
```

After dropping irrelevant columns, there remain 10 categorical features and 5 continuous ones. The continuous features are *scaled* and the categorical ones are *encoded*. The encoding process is demonstrated below. In most cases, we only encode the top 10 most frequently occuring categories.

title	genre
Lost River	Drama, Fantasy, Mystery
Un ponte per Terabithia	Drama, Family, Fantasy
Please Give	Comedy, Drama
The Law of Enclosures	Drama
Suburban Mayhem	Comedy, Drama, Thriller

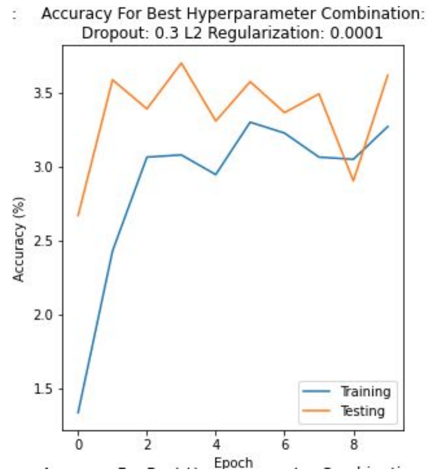
title	other_genre	Family	Mystery	Thriller	Comedy	Fantasy	Drama
Lost River	0	0	1	0	0	1	1
Un ponte per Terabithia	0	1	0	0	0	1	1
Please Give	0	0	0	0	1	0	1
The Law of Enclosures	0	0	0	0	0	0	1
Suburban Mayhem	0	0	0	1	1	0	1

Learning Models

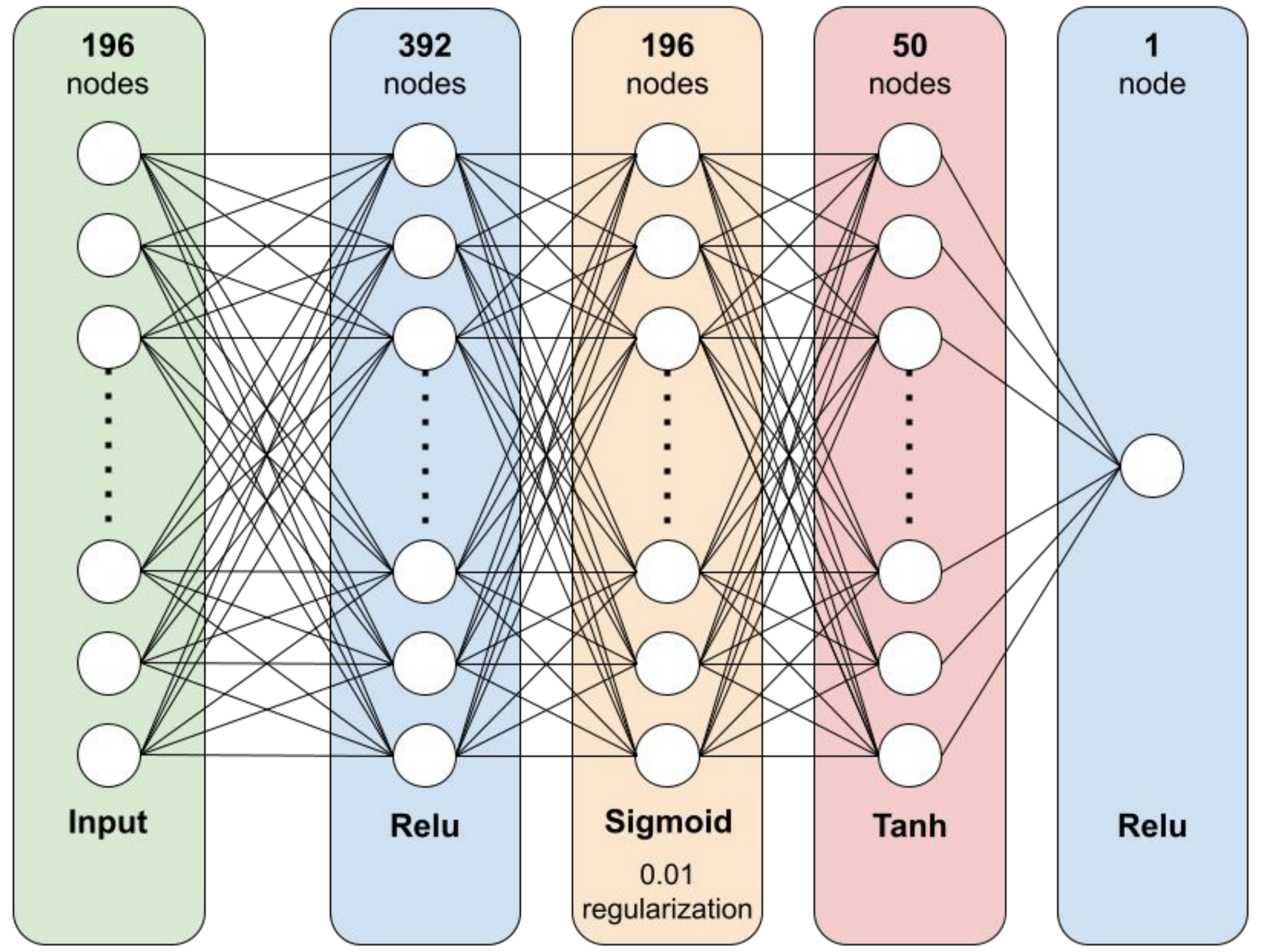


Simple Neural Network:
A fully-connected single-hidden layer neural network was outputted to a single regression neuron for the movie rating range from 0 to 10. The regressor was tuned on varying the number of neurons in the hidden layer with linear activation in the output layer using Trenn's formula[1]: $n_h = \sqrt{n_i n_o}$

The neural network was also designed for classification by spreading the 0 to 10 single decimal rating range across 100 classes, activated by SoftMax. The classifier involved the same structure as the regressor with its hidden layer tuned to ReLU activation and Normal initialization. Seen to the right, overfitting correction with Dropout and L2 regularization failed; the regressor is the preferred Simple model.

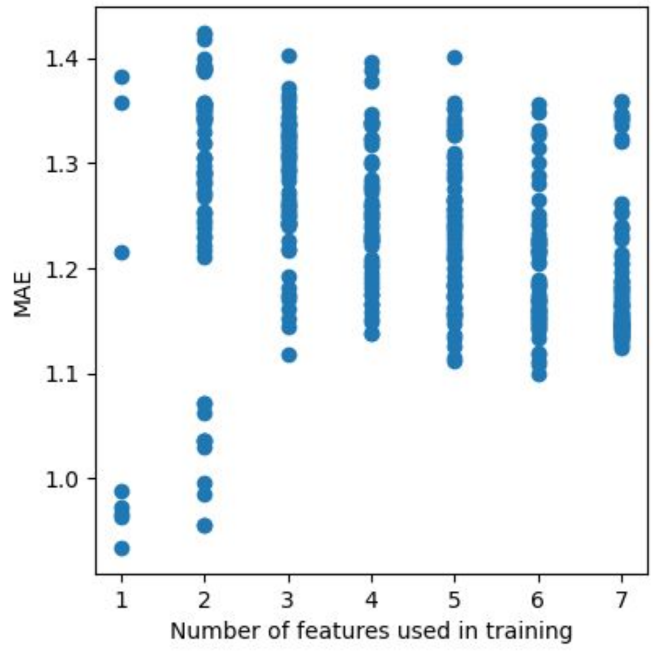


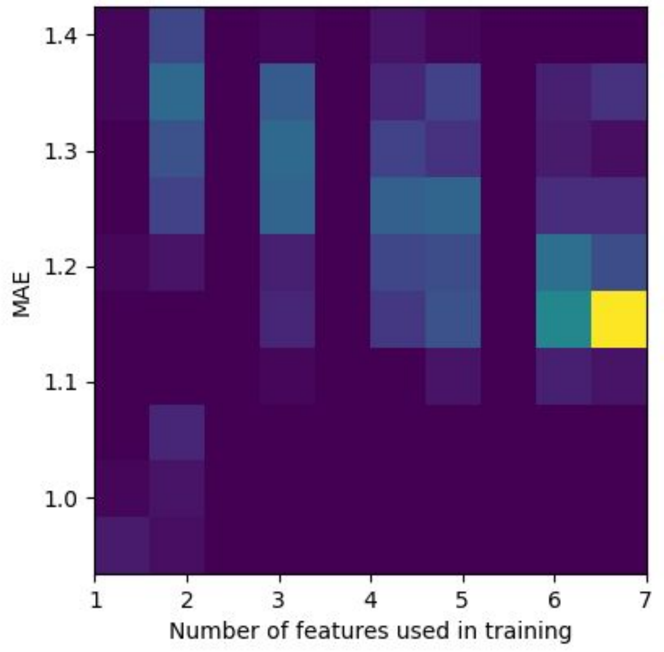
Deep Neural Network:
A fully connected deep neural neural network with 3 hidden layers and varied activation functions that leads into one regression neuron. The final model was a ReLU layer followed by sigmoid, tanh, and another ReLU layer with a regularizer of 0.01 on the Sigmoid layer. From the chart on the right it can be seen that upon approaching an MAE of 0.736 very quickly, the model evens out for the rest of the epochs.



Decision Tree:
A simple decision tree was constructed using an adapted version of the hand-built ID3 inference algorithm developed during assignment one. Using the three categorical features with the least unique values, the resulting decision tree produced an MAE of 0.95.`

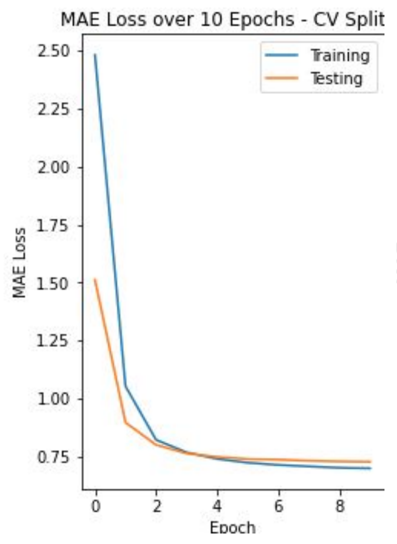
Random Forest:
500 unique random forests were generated, with each forest having 100 trees. The features used for each tree were selected randomly.
The resulting MAEs for each number of selected features are shown in the plot to the right:

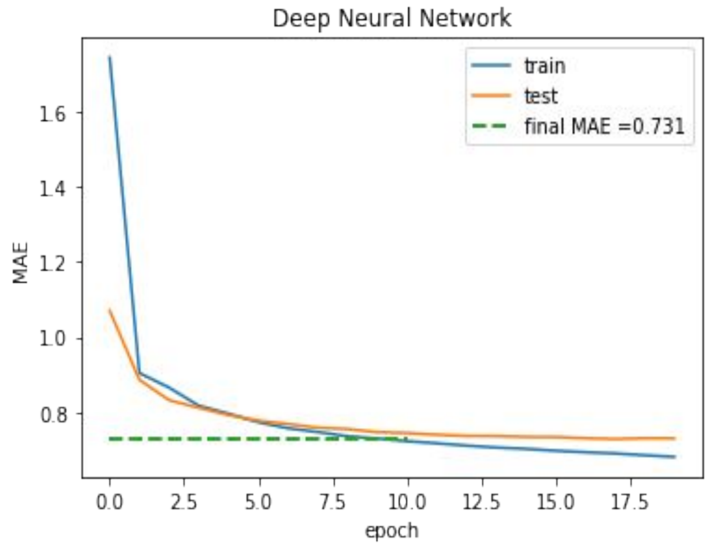




Results

Neural Networks
The Simple NN (left) is preferred to the Deep NN (right) for its better MAE. It also makes more sense to use the simpler model as a simpler structure minimizes complexity and maximizes efficiency.





Model	Train MAE	Test MAE
Simple NN	0.71	0.72
Deep NN	0.68	0.73
Decision Tree	N/A	0.95
Random Forest	N/A	0.96

Decision Trees
The best decision tree MAE was given by the hand-written ID3 algorithm using the three features with the least unique values (`top_actor_gender`, `height`, and `divorces`). This suggests that features with many unique values are not well suited for decision trees.

Random Forests
All of the top 15 results from the random forest tests were achieved using either 1 or 2 features for training only. This reaffirms the conclusion that high-cardinality features are not well suited for use with decision trees. The random forest procedure did not appear to improve upon ordinary decision trees, with a best MAE of 0.96 vs 0.95.

Conclusion

- ➔ Training a Neural Network on mostly categorical data is difficult
- ➔ Choosing a different approach could benefit us (like XGBoost)
- ➔ You can only optimize a Neural Network so much
- ➔ The diversity/cardinality of the categorical features was detrimental to the decision tree performance
- ➔ Predicting human behaviour is NP Hard

References

[1] K. Sheela and S. Deepa, "Review on Methods to Fix Number of Hidden Neurons in Neural Networks", Mathematical Problems in Engineering, vol. 2013, pp. 1-11, 2013. [Accessed: 13- Aug-2021]