Project Name: Predicting IMDb Scores

Project Description: Develop a machine learning model to predict the IMDb scores of movies available on Films based on their genre, premiere date, runtime, and language. The model aims to accurately estimate the popularity of movies to assist users in discovering highly rated films that align with their preferences.

Phase 2 : Innovation

Description : Consider exploring advanced regression techniques like Gradient Boosting or Neural Networks for improved prediction accuracy.

IMDb Score Prediction Using Neural Network

1. Collect and prepare data

Gather a dataset of movies with their IMDb scores. Preprocess the data by cleaning the text, removing stop words, and stemming or lemmatizing the words.

2. Choose a neural network architecture.

There are many different neural network architectures that can be used for regression tasks, such as predicting IMDb scores. Some popular architectures include feedforward neural networks, convolutional neural networks, and recurrent neural networks.

3. Train neural network model.

Feed the prepared data to the model and allow it to learn the relationships between the features and the target variable (IMDb score).

4. Evaluate the model.

Evaluate the performance of the model on a held-out test set. This will give an idea of how well about the model will generalize to new data.

5. Deploy the model.

Once you are satisfied with the performance of the model, it can deploy it to production. This may involve saving the model to a file or deploying it to a cloud-based platform.

Here is an example of how to train a simple feedforward neural network to predict IMDb scores using the Keras library in Python:

import keras

# Load the IMDb dataset

(x\_train, y\_train), (x\_test, y\_test) = keras.datasets.imdb.load\_data(num\_words=10000)

# Preprocess the data

x\_train = keras.preprocessing.sequence.pad\_sequences(x\_train, maxlen=500)

x\_test = keras.preprocessing.sequence.pad\_sequences(x\_test, maxlen=500)

# Define the neural network architecture

model = keras.Sequential([

keras.layers.Embedding(input\_dim=10000, output\_dim=128),

keras.layers.LSTM(128),

keras.layers.Dense(1, activation='linear')

])

# Compile the model

model.compile(loss='mse', optimizer='adam')

# Train the model

model.fit(x\_train, y\_train, epochs=10)

# Evaluate the model on the test set

loss, mse = model.evaluate(x\_test, y\_test)

print('Test loss:', loss)

print('Test MSE:', mse)

# Make a prediction on a new movie

new\_movie = ['This is a great movie!']

new\_movie = keras.preprocessing.text.Tokenizer(num\_words=10000).texts\_to\_sequences(new\_movie)

new\_movie = keras.preprocessing.sequence.pad\_sequences(new\_movie, maxlen=500)

prediction = model.predict(new\_movie)

print('Predicted IMDb score:', prediction[0])