PREDICTING IMDb SCORES

PHASE-02(INNOVATION)

PROJECT OVERVIEW:

The goal of this project is to develop a predictive model for IMDb movie scores using advanced regression techniques. We will leverage machine learning algorithms such as Gradient Boosting and Neural Networks to achieve improved prediction accuracy.

PREDICTING IMDb SCORES USING DATA SCIENCE:

Predicting IMDb scores using data science involves using statistical and machine learning techniques to build a model that can estimate the rating (usually on a scale from 1 to 10) of a movie based on various features or factors. Here's a simplified step-by-step explanation of the process:

1. Data Collection: Gather a dataset that includes information about movies, such as cast, crew, budget, genre, release date, runtime, and more. IMDb provides datasets for this purpose.

2. Data Preprocessing: Clean and prepare the data by handling missing values, converting categorical features into numerical representations (e.g., one-hot encoding), and scaling/normalizing numerical features.

3. Feature Selection: Choose the most relevant features that may influence a movie's IMDb score. Feature selection helps improve the model's efficiency and accuracy.

4. Data Splitting: Divide the dataset into two parts: a training set and a testing/validation set. The training set is used to train the predictive

model, while the testing/validation set is used to evaluate the model's performance.

5. Model Selection: Select an appropriate machine learning model for regression tasks. Common choices include linear regression, decision trees, random forests, support vector machines, and neural networks. The choice of model can depend on the complexity of the problem and the dataset size.

6. Model Training: Train the selected model on the training data. During training, the model learns the relationships between the features and the IMDb scores.

7. Model Evaluation: Use the testing/validation set to assess the model's performance. Common evaluation metrics for regression tasks include

Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

8. Hyperparameter Tuning: Fine-tune the model by adjusting hyperparameters to optimize its performance. Techniques like cross-validation can help with this.

9. Prediction: Once the model is trained and evaluated satisfactorily, it can be used to predict IMDb scores for new, unseen movies.

10. Interpretation: Analyze the model to understand which features have the most impact on IMDb scores. This insight can be valuable for filmmakers and studios.

11. Deployment: If the model performs well and meets the desired criteria, it can be deployed in a

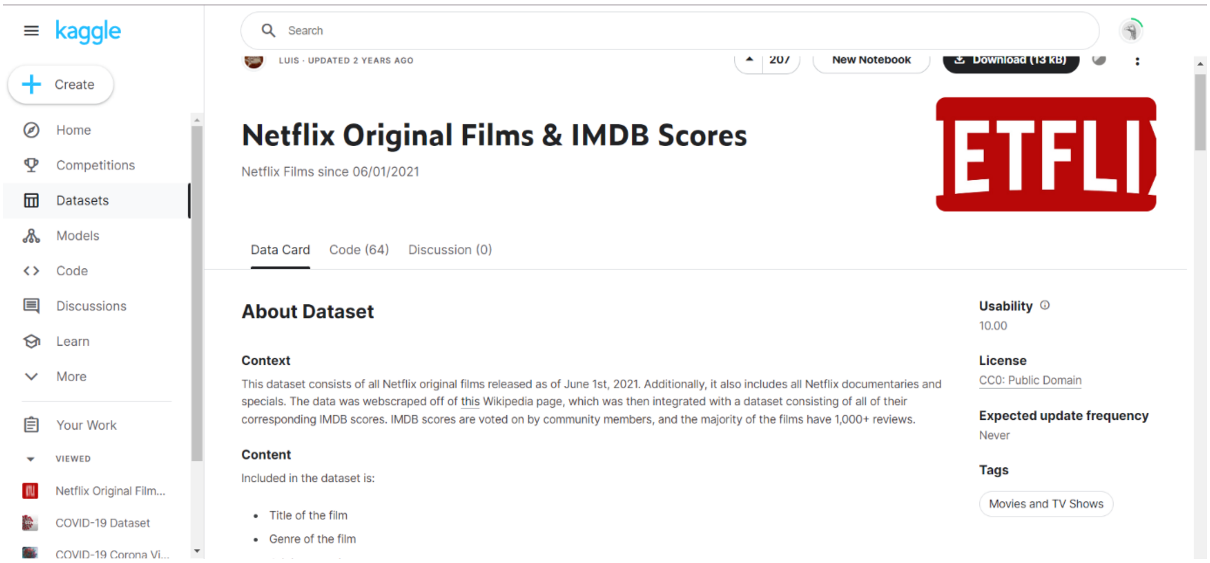
production environment to make real-time predictions.

12. Monitoring and Maintenance: Continuously monitor the model's performance and retrain it periodically to account for changes in movie preferences and trends.

Predicting IMDb scores using data science can provide valuable insights into the factors that influence a movie's rating and help filmmakers, studios, and movie enthusiasts make informed decisions.

SOURCE OF THE DATASET:

The dataset for this project is sourced from “Kaggle” and contains information about Netflix original films and their IMDb scores. It can be access in the dataset at the following link: [Netflix Original Films IMDb Scores Dataset]: https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores



Netflix Original Films & IMDB Scores

Netflix Films since 06/01/2021

About Dataset

Context

This dataset consists of all Netflix original films released as of June 1st, 2021. Additionally, it also includes all Netflix documentaries and specials. The data was webscraped off of this Wikipedia page, which was then integrated with a dataset consisting of all of their corresponding IMDB scores. IMDB scores are voted on by community members, and the majority of the films have 1,000+ reviews.

Content

Included in the dataset is:

- Title of the film
- Genre of the film

Usability

10.00

License

CC0: Public Domain

Expected update frequency

Never

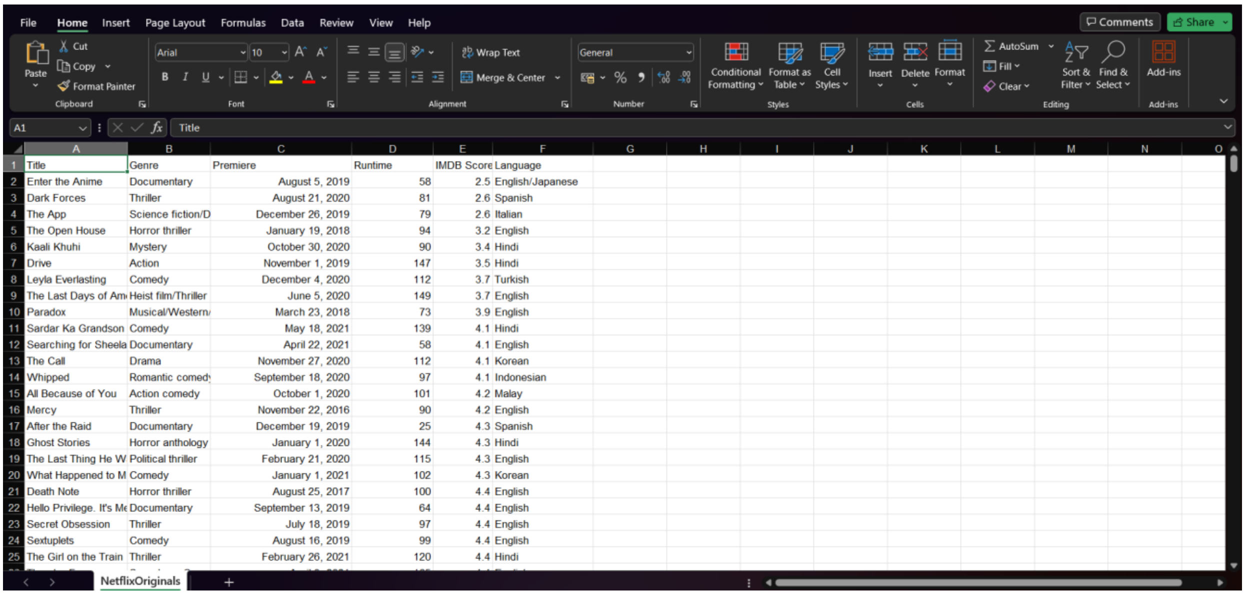
Tags

Movies and TV Shows

DETAILS ABOUT THE DATASET:

Included in the dataset is:

- Title of the film
- Genre of the film
- Original premiere date
- Runtime in minutes
- IMDB scores (as of 06/01/21)
- Languages currently available (as of 06/01/21)



	A	B	C	D	E	F
1	Title	Genre	Premiere	Runtime	IMDB Score	Language
2	Enter the Anime	Documentary	August 5, 2019	58	2.5	English/Japanese
3	Dark Forces	Thriller	August 21, 2020	81	2.6	Spanish
4	The App	Science fiction/D	December 26, 2019	79	2.6	Italian
5	The Open House	Horror thriller	January 19, 2018	94	3.2	English
6	Kaali Khuhi	Mystery	October 30, 2020	90	3.4	Hindi
7	Drive	Action	November 1, 2019	147	3.5	Hindi
8	Leyla Everlasting	Comedy	December 4, 2020	112	3.7	Turkish
9	The Last Days of Am	Heist film/Thriller	June 5, 2020	149	3.7	English
10	Paradox	Musical/Western/	March 23, 2018	73	3.9	English
11	Sardar Ka Grandson	Comedy	May 18, 2021	139	4.1	Hindi
12	Searching for Sheela	Documentary	April 22, 2021	58	4.1	English
13	The Call	Drama	November 27, 2020	112	4.1	Korean
14	Whipped	Romantic comed	September 18, 2020	97	4.1	Indonesian
15	All Because of You	Action comedy	October 1, 2020	101	4.2	Malay
16	Mercy	Thriller	November 22, 2016	90	4.2	English
17	After the Raid	Documentary	December 19, 2019	25	4.3	Spanish
18	Ghost Stories	Horror anthology	January 1, 2020	144	4.3	Hindi
19	The Last Thing He W	Political thriller	February 21, 2020	115	4.3	English
20	What Happened to M	Comedy	January 1, 2021	102	4.3	Korean
21	Death Note	Horror thriller	August 25, 2017	100	4.4	English
22	Hello Privilege. It's Me	Documentary	September 13, 2019	64	4.4	English
23	Secret Obsession	Thriller	July 18, 2019	97	4.4	English
24	Sextuplets	Comedy	August 16, 2019	99	4.4	English
25	The Girl on the Train	Thriller	February 26, 2021	120	4.4	Hindi

The target variable is the IMDb score,which is Continuous variable.

Libraries used in credit card fraud detection:

We use the following libraries and frameworks in credit card fraud detection project.

1. Python – 3.x

2. Numpy – 1.19.2

3. Scikit-learn – 0.24.1

4. Matplotlib – 3.3.4

5. Imblearn – 0.8.0

How to download the libraries:

Download python 3.x

This document describes how to install Python 3.6 or 3.8 on Ubuntu Linux machines.To see

which version of Python 3 you have installed, open a command prompt and run

```
$ python3 –version
```

If you are using Ubuntu 16.10 or newer, then you can easily install Python 3.6 with the

following

commands:

```
$ sudo apt-get update

$ sudo apt-get install python3.6
```

If you’re using another version of Ubuntu (e.g. the latest LTS release) or you want to use a

more current

Python, we recommend using the deadsnakes PPA to install Python 3.8:

```
$ sudo apt-get install software-properties-common
```

```
$ sudo add-apt-repository ppa:deadsnakes/ppa
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install python3.8
```

If you are using other Linux distribution, chances are you already have Python 3 pre-installed

as well. If not, use your distribution's package manager. For example on Fedora, you would

use dnf:

```
$ sudo dnf install python3
```

Download numpy 1.19.2

Installed Pythons found by C:\WINDOWS\py.exe Launcher for Windows

I use Python 3.6 for ESA SNAP snappy, so

```
PS C:\> cd $env:USERPROFILE\.snap\snap-python\

PS C:\Users\XXXXX\.snap\snap-python> py -3.6

Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)]

on win32

Type “help”, “copyright”, “credits” or “license” for more information.
```

```
>>> import snappy
```

Numpy installation:

To install numpy on a Python.org 2 version for use with snappy:

```
PS C:\> py -3.6 -m pip install numpy
```

```
Collecting numpy

Downloading

51986f6d0c5c2/numpy-1.19.5-cp36-cp36m-win_amd64.whl (13.2MB)
```

```
100%
|████████████████████████████
████| 13.2MB 3.9MB/s

Installing collected packages: numpy

The script f2py.exe is installed in
‘D:\Python36\Scripts’ which is not on PATH.


Consider adding this directory to PATH or, if you
prefer to suppress this warning, use –no-

warn-script location.


Successfully installed numpy-1.19.5

You are using pip version 18.1, however version
21.0.1 is available.

You should consider upgrading via the ‘python -m
pip install –upgrade pip’ command.
```

Note that using py -M.N doesn’t adjust the PATH

Download matplotlib

Matplotlib releases are available as wheel packages for macOS, Windows and Linux on PyPI.

Install it

using pip:

```
Python -m pip install -U pip

Python -m pip install -U matplotlib
```

If this command results in Matplotlib being compiled from source and there’s trouble with

the compilation, you can add –prefer-binary to select the newest version of Matplotlib for

which there is a precompiled wheel for that.

Install an official release

Matplotlib releases are available as wheel packages for macOS, Windows and Linux on PyPI.

Install it

using pip:

```
Python -m pip install -U pip

Python -m pip install -U matplotlib
```

If this command results in Matplotlib being compiled from source and there’s trouble with

the

compilation, you can add –prefer-binary to select the newest version of Matplotlib for which

there is a

precompiled wheel for your OS and Python.

Third-party distributions

Various third-parties provide Matplotlib for their environments.

Conda packages

Matplotlib is available both via the anaconda main channel

Conda install matplotlib

As well as via the conda-forge community channel

```
Conda install -c conda-forge matplotlib
```

Downloading Imblearn 0.8.0

Imbalanced-learn

Imbalanced-learn is a python package offering a number of re-sampling techniques

commonly used in datasets showing strong between-class imbalance. It is compatible with

scikit-learn and is part of scikit learn-contrib projects.

Documentation

Installation documentation, API documentation, and examples can be found on the

documentation.

Dependencies

Imbalanced-learn is tested to work under Python 3.6+. The dependency requirements are

based on the

last scikit-learn release:

Scipy(>=0.19.1)

Numpy(>=1.13.3)

Scikit-learn(>=0.23)

Joblib(>=0.11)

Keras 2 (optional)

Tensorflow (optional)

Additionally, to run the examples, you need matplotlib(>=2.0.0) and pandas(>=0.22).

How to Train/Test:

Train/Test is a method to measure the accuracy of your model .It is called Train/Test

because you split the data set into two sets: a training set and a testing set.

80% for training, and 20% for testing.

You train the model using the training set.

You test the model using the testing set.

Train the model means create the model.

Test the model means test the accuracy of the model.

Split Into Train/Test

The training set should be a random selection of 80% of the original data.

The testing set should be the remaining 20%.

```
train_x = x[:80]

train_y = y[:80]

test_x = x[80:]

test_y = y[80:]
```

Display the Training Set

Display the same scatter plot with the training set.

Example

```
plt.scatter (train_x, train_y)
plt.show()
```

6.EXTRA EXPLANATION

ALGORITHM USED IN PREDICTING IMDb SCORE:

Here is a list of algorithms used for predicting IMDb scores:

1. Linear Regression
2. Decision Trees and Random Forests
3. Gradient Boosting Algorithms (e.g., XGBoost, LightGBM, CatBoost)
4. Support Vector Machines (SVM)
5. Neural Networks
6. K-Nearest Neighbors (KNN)
7. Ridge and Lasso Regression
8. Elastic Net Regression

9. Polynomial Regression

10. Time Series Models (e.g., ARIMA, LSTM)

METRICS USED FOR PREDICTING IMDb SCORE:

Metrics commonly used to assess the accuracy of IMDb score predictions include:

1. Mean Absolute Error (MAE): Measures the average absolute difference between the predicted IMDb scores and the actual scores. It provides a sense of how far off, on average, the predictions are.

2. Mean Squared Error (MSE): Measures the average of the squared differences between predictions and actual IMDb scores. MSE penalizes larger errors more heavily than MAE.

3. Root Mean Squared Error (RMSE): The square root of the MSE, which provides an interpretable metric in the same units as IMDb scores. It's a common choice for reporting prediction accuracy.

4. R-squared (R2): Measures the proportion of the variance in IMDb scores that can be explained by the model. A higher R-squared indicates a better fit of the model to the data.

5. Mean Absolute Percentage Error (MAPE): Measures the percentage difference between predicted and actual scores, making it useful for understanding prediction accuracy in relative terms.

6. Coefficient of Determination (CD): Similar to R-squared, it quantifies the goodness of fit, with values closer to 1 indicating a better fit.

7. Precision, Recall, F1-score: If IMDb scores are converted into classes (e.g., "Good," "Average," "Bad"), classification metrics like precision, recall, and F1-score can be used to evaluate how well the model classifies movies.

8. Spearman's Rank Correlation Coefficient: Assesses the correlation between the predicted and actual IMDb scores while considering the rank order of movies rather than their absolute values.

9. Kendall's Tau: Another rank-based correlation metric used to measure the association between predicted and actual IMDb scores while accounting for concordant and discordant pairs.

10. Percentage of Correctly Predicted Cases: A straightforward metric that calculates the

percentage of correctly predicted IMDb scores or classes.