**PREDICTING IMDB SCORE**

## Introduction to IMDB:



**What is IMDb?**

IMDb is a platform that provides a vast collection of data related to films, [**TV series**](https://www.careerguide.com/career/post-tag/tv-shows), and other media content. It offers users the ability to explore details about movies and shows, such as cast and crew information, release dates, plot summaries, trivia, user reviews, ratings, and much more. Whether you’re a casual moviegoer or a dedicated film enthusiast, IMDb offers a wealth of information to satisfy your curiosity.

IMDb was founded by Col Needham in Bristol, England. It began as a hobby project by Needham to catalog his personal [**movie collection**](https://www.careerguide.com/career/film-industry). Over time, it gained popularity and attracted a community of movie fans who contributed to its growth. In 1998, Amazon acquired IMDb, which allowed the platform to further expand its offerings and resources.

As technology advanced, IMDb evolved from a simple database into a multifaceted website and mobile app, offering not only information but also features like user reviews, recommendations, and a professional version for industry insiders. The platform’s journey from a small passion project to a global entertainment hub reflects its enduring impact on the way we consume and engage with movies and TV shows.

## Key Features of IMDb

#### Ratings and Reviews

1. **User Ratings**: IMDb allows users to rate movies and TV shows on a scale of 1 to 10. These ratings provide an overview of audience opinions and preferences.
2. **Reviews**: Users can write and read reviews to share their thoughts and insights about specific titles. This feature encourages discussions and helps others make informed viewing choices.

#### Cast and Crew Information

1. **Detailed Cast Lists**: IMDb provides comprehensive lists of actors, directors, writers, and other crew members associated with movies and TV shows.
2. **Filmography**: Each actor and crew member’s IMDb [**profile showcases**](https://www.careerguide.com/career/profile-building-service) their filmography, allowing users to explore their body of work.

#### Trivia and Fun Facts

1. **Trivia Sections**: Movie and TV show pages include trivia sections that offer interesting behind-the-scenes information, Easter eggs, and lesser-known facts.
2. **Connections**: IMDb highlights connections between actors, characters, and movies. For example, you can discover which actors have appeared in multiple movies together.

## IMDbPro: The Professional Version

#### Benefits for Industry Professionals

* 1. **Contact Information**: IMDbPro provides access to contact details of industry insiders, including actors, directors, producers, agents, and executives. This facilitates [**direct communication**](https://www.careerguide.com/career/communication-skills) and collaboration.
  2. **Casting Information**: IMDbPro offers insights into casting details for upcoming roles, enabling actors and casting directors to find suitable talent for projects.
  3. **Box Office Data**: IMDbPro provides box office data, including opening weekend grosses and historical performance, helping professionals gauge the success of different titles.

#### Advanced Features and Tools

* 1. **Starmeter Rankings**: IMDbPro subscribers can see their “Starmeter” rankings, which reflect their popularity and visibility within the [**entertainment industry**](https://www.careerguide.com/career/new-age-career-options/top-10-jobs-in-entertainment-industry).
  2. **Enhanced Profiles**: IMDbPro users have the option to add additional information and media to their profiles, making them more informative and engaging for potential collaborators.
  3. **Messaging**: The platform enables direct messaging between professionals, streamlining communication for networking and collaboration purposes.

## User-Generated Content on IMDb

#### User Reviews and Ratings

1. **Personal Reviews**: Users can write detailed reviews for movies and TV shows, sharing their thoughts, [**analyses, and recommendations**](https://www.careerguide.com/career/career-options/a-career-guide-for-a-financial-analyst) with the community.
2. **Star Ratings**: IMDb allows users to rate movies and TV shows on a scale of 1 to 10 stars, contributing to the overall rating and helping others gauge audience reception.

#### Contributions to Data Accuracy

1. **Edit and Correct Information**: Users have the ability to edit and correct certain information on IMDb. This collaborative effort enhances the accuracy and completeness of the database.
2. **Trivia and Quotes**: Enthusiasts can add interesting trivia, memorable quotes, and connections between movies, contributing to the platform’s depth of information.

#### Community and Forums

1. **Message Boards (Formerly)**: IMDb used to host message boards where users could engage in discussions about movies, actors, and various topics related to the entertainment industry.
2. **Community Contributions**: Users can participate in discussions, [**ask questions**](https://www.careerguide.com/career/career-advice/questions-to-ask-yourself-before-choosing-a-career), and share their expertise within the IMDb community, connecting with others who share similar interests.

## Corporate Social Responsibility

|  |  |
| --- | --- |
| **Impact** | **Description** |
| **Cultural Barometer** | The IMDb Top 250 list, showcasing the highest-rated movies as voted by users, has become a cultural barometer of cinematic excellence. It reflects the collective preferences of a global audience and informs [**discussions about must-watch films.**](https://www.careerguide.com/career/online-career-counselling/8-movies-a-child-must-watch-with-parents) |
| **Memorable Quotes and References** | IMDb compiles memorable quotes from movies and TV shows, which often become part of everyday [**language and references**](https://institute.careerguide.com/importance-of-body-language-and-ways-to-improve/) in conversations. |
| **Cultural Conversations** | IMDb ratings and reviews often influence cultural conversations and debates about a movie’s quality, themes, and impact. |
| **Movie Recommendations** | The list guides viewers toward acclaimed movies they might have missed, leading to a resurgence of interest in classic and lesser-known films. |

## DATA:

## 1.IMDB DATA

## 2. YOUTUBE DATA

## 3.FACEBOOK DATA

## MACHINE COD ALGORITHM MODEL:

## 1: Prepare data set

## 2: Check Minority

## 3: If needed apply SMOTE algorithm until the minority class becomes equal to the size of it’s closest class

## 4: Classification

## 5: Accuracy ←− 0

## 6: while True do

## 7: Resample Data

## 8: Call (Classifier)

## 9: if % of correctly classified Instance >Previous Accuracy Measure then

## 10: Accuracy ←− % of correctly classif ied Instance

## 11: else

## 12: Break

## 13: end if

## 14: end while=0

# Step 1 : Importing required libraries

In [1]:

*# Python Libraries*

import numpy as np

*# numneric calculation*

import pandas as pd

*# Libraries for Visualization*

import plotly.express as px

import matplotlib.pyplot as plt

*# Library for splitting the data in Train and Test*

from sklearn.model\_selection import train\_test\_split

*# Library required for the Linear Regression Algorithm*

from sklearn.linear\_model import LinearRegression

*# Library for the metric required to evaluate the model*

from sklearn.metrics import mean\_absolute\_error

from sklearn.metrics import mean\_squared\_error

%matplotlib inline

*# allow to plot the charts inline*

In [2]:

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

/kaggle/input/student-scores/student\_scores.csv

/kaggle/input/market-share-of-pharma-company-imaginary/Growth\_Rate\_Medical\_Mojo.csv

/kaggle/input/market-share-of-pharma-company-imaginary/Market\_Share.csv

In [3]:

df = pd.read\_csv('/kaggle/input/student-scores/student\_scores.csv')

df.head()

Out[3]:

|  | Hours | Scores |
| --- | --- | --- |
| 0 | 2.5 | 21 |
| 1 | 5.1 | 47 |
| 2 | 3.2 | 27 |
| 3 | 8.5 | 75 |
| 4 | 3.5 | 30 |

In [4]:

df.sample(6)

Out[4]:

|  | Hours | Scores |
| --- | --- | --- |
| 16 | 2.5 | 30 |
| 24 | 7.8 | 86 |
| 7 | 5.5 | 60 |
| 14 | 1.1 | 17 |
| 21 | 4.8 | 54 |
| 18 | 6.1 | 67 |

# Step 3 : Understanding the data\*\*

In [5]:

df.dtypes

Out[5]:

Hours float64

Scores int64

dtype: object

In [6]:

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 25 entries, 0 to 24

Data columns (total 2 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Hours 25 non-null float64

1 Scores 25 non-null int64

dtypes: float64(1), int64(1)

memory usage: 528.0 bytes

In [7]:

df.shape

Out[7]:

(25, 2)

In [8]:

df.isnull().sum()

Out[8]:

Hours 0

Scores 0

dtype: int64

# Step 4 : Exploring the relationship in the dataset

In [9]:

fig = px.scatter(x = df['Hours'], y = df['Scores'], labels = {'x' : 'Hours Studies', 'y' : 'Scores Obtained'})

fig.show()

24682030405060708090100

Hours StudiesScores Obtained

In [10]:

plt.scatter(df.Hours, df.Scores)

Out[10]:

<matplotlib.collections.PathCollection at 0x7f24ee813490>

# Step 5 : Splitting the dataset into Train and Test data

In [11]:

feature = df['Hours'].values

target = df['Scores'].values

In [12]:

*# Reshaping the features and target*

feature = feature.reshape(-1,1)

target = target.reshape(-1,1)

In [13]:

plt.scatter(feature, target)

Out[13]:

<matplotlib.collections.PathCollection at 0x7f24e679a690>

In [14]:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(feature, target, test\_size = 0.2, random\_state = 0)

print("X-Train :",X\_train.shape)

print("X-Test :",X\_test.shape)

print("Y-Train :",y\_test.shape)

print("Y-Test :",y\_test.shape)

X-Train : (20, 1)

X-Test : (5, 1)

Y-Train : (5, 1)

Y-Test : (5, 1)

# Step 6 : Implementing the Linear Regression Algorithm

In [15]:

*# Instantiating the object of the class*

lr = LinearRegression()

*# Fitting the model*

lr.fit(X\_train, y\_train)

Out[15]:

LinearRegression()

#### Plotting the Regression Line with Actual vs Predicted values

In [16]:

plt.figure(figsize = (15,8))

plt.scatter(X\_train, y\_train)

plt.plot(X\_train, lr.predict(X\_train), color = 'green')

plt.title('Hours vs Scores')

plt.xlabel('Number of hours studied')

plt.ylabel('Scored Obtained')

plt.show()

### Predicting the values

In [17]:

pred\_vals = lr.predict(X\_test)

pred\_vals

Out[17]:

array([[16.88414476],

[33.73226078],

[75.357018 ],

[26.79480124],

[60.49103328]])

Creating a dataframe with Actual and Predicted Values

In [18]:

df\_ac\_vs\_pr = pd.DataFrame(pred\_vals, index = range(0,5), columns = ['Predicted'])

df\_ac\_vs\_pr['Actual'] = y\_test

In [19]:

df\_ac\_vs\_pr

Out[19]:

|  | Predicted | Actual |
| --- | --- | --- |
| 0 | 16.884145 | 20 |
| 1 | 33.732261 | 27 |
| 2 | 75.357018 | 69 |
| 3 | 26.794801 | 30 |
| 4 | 60.491033 | 62 |

# Step 7 : Evaluating the performance of the model

In [20]:

print('Mean Absolute Error =', mean\_absolute\_error(y\_test, pred\_vals))

print("Mean Squared Error= ", mean\_squared\_error(y\_test, pred\_vals))

print("Root Mean Squared Error= ", np.sqrt(mean\_squared\_error(y\_test, pred\_vals)))

Mean Absolute Error = 4.183859899002982

Mean Squared Error= 21.598769307217456

Root Mean Squared Error= 4.647447612100373

# Step 8 : Finding the solution of question asked in the problem statement

What will be predicted score if a student studies for 9.25 hrs/ day?

In [21]:

*# Defining variable 'y' with given data*

y = np.array(9.25)

y = y.reshape(-1, 1)

*# Predicting on the basis of the value in 'y'*

pred\_y = lr.predict(y)

pred\_y

Out[21]:

array([[93.69173249]])

In [22]:

print('The score obtained after studying for **{}** hours = **{}**'.format(y[0][0], pred\_y[0][0]))

The score obtained after studying for 9.25 hours = 93.69173248737539

## Conclusion

The [**Internet Movie Database (IMDb)**](https://simple.wikipedia.org/wiki/Internet_Movie_Database) stands as a remarkable testament to the power of information and community within the realm of entertainment. From its humble origins as a personal project to becoming a global phenomenon, IMDb has redefined how we explore, appreciate, and engage with movies and TV shows.

IMDb’s influence on the entertainment industry is undeniable. It serves as an essential resource for filmmakers, actors, and industry professionals, offering a comprehensive platform for networking, research, and project exploration. The IMDbPro subscription further caters to the needs of those working behind the scenes, providing tools to navigate the intricate web of the industry.

Algorithm 1 Algorithm for developing the model

1: Prepare data set

2: Check Minority

3: If needed apply SMOTE algorithm until the minority class

becomes equal to the size of it’s closest class

4: Classiﬁcation

5: Accuracy ←− 0

6: while True do

7: Resample Data

8: Call (Classiﬁer)

9: if % of correctly classiﬁed Instance >Previous Accu-

racy Measure then

10: Accuracy ←− %of correctly classified

Instance

11: else

12: Break

13: end if

1. 14: end while=0