|  |  |
| --- | --- |
| Project Title | **Finance & Accounting Courses - Udemy (13K+ course)** |
| language | Machine learning, python, SQL, Excel |
| Tools | VS code, Jupyter notebook |
| Domain | Data Analyst |
| Project Difficulties level | Advance |

Dataset : Dataset is available in the given link. You can download it at your convenience.

[Click](https://drive.google.com/file/d/1rqn_eoGU0aN8QC-xK_s4xZ99pgwV1koF/view?usp=sharing) [here](https://drive.google.com/file/d/1rqn_eoGU0aN8QC-xK_s4xZ99pgwV1koF/view?usp=sharing) [to](https://drive.google.com/file/d/1rqn_eoGU0aN8QC-xK_s4xZ99pgwV1koF/view?usp=sharing) [download](https://drive.google.com/file/d/1rqn_eoGU0aN8QC-xK_s4xZ99pgwV1koF/view?usp=sharing) [data](https://drive.google.com/file/d/1rqn_eoGU0aN8QC-xK_s4xZ99pgwV1koF/view?usp=sharing) [set](https://drive.google.com/file/d/1rqn_eoGU0aN8QC-xK_s4xZ99pgwV1koF/view?usp=sharing)

# About Dataset Context

**A compilation of all the development related courses ( 13 thousand courses) which are available on Udemy's website. Under the development category, there are courses from Finance, Accounting, Book Keeping, Compliance,**

**Cryptocurrence, Blockchain, Economics, Investing & Trading, Taxes and much more each having multiple courses under it's domain.**

**All the details can be found on** [**Udemy's**](https://www.udemy.com/?utm_source=adwords-brand&utm_medium=udemyads&utm_campaign=Brand-Udemy_la.EN_cc.INDIA&utm_term=_._ag_78279294239_._ad_450776424635_._de_c_._dm__._pl__._ti_kwd-310556426868_._li_1007785_._pd__._&utm_term=_._pd__._kw_udemy_._&matchtype=e&gclid=EAIaIQobChMIgaPGkarj6wIViSQrCh1weAXPEAAYASAAEgIhuvD_BwE)[**website**](https://www.udemy.com/?utm_source=adwords-brand&utm_medium=udemyads&utm_campaign=Brand-Udemy_la.EN_cc.INDIA&utm_term=_._ag_78279294239_._ad_450776424635_._de_c_._dm__._pl__._ti_kwd-310556426868_._li_1007785_._pd__._&utm_term=_._pd__._kw_udemy_._&matchtype=e&gclid=EAIaIQobChMIgaPGkarj6wIViSQrCh1weAXPEAAYASAAEgIhuvD_BwE) **as well!**

# Content

**Here, I have extracted data related to 10k courses which come under the development category on Udemy's website.**

**The 17 columns in the dataset can be used to gain insights related to:**

* **id : The course ID of that particular course.**
* **title : Shows the unique names of the courses available under the development category on Udemy.**
* **url: Gives the URL of the course.**
* **is\_paid : Returns a boolean value displaying true if the course is paid and false if otherwise.**
* **num\_subscribers : Shows the number of people who have subscribed that course.**
* **avg\_rating : Shows the average rating of the course.**
* **avg rating recent : Reflects the recent changes in the average rating.**
* **num\_reviews : Gives us an idea related to the number of ratings that a course has received.**
* **num\_ published\_lectures : Shows the number of lectures the course offers.**
* **num\_ published\_ practice\_tests : Gives an idea of the number of practice tests that a course offers.**
* **created : The time of creation of the course.**
* **published\_time : Time of publishing the course.**
* **discounted\_ price\_amount : The discounted price which a certain course is being offered at.**
* **discounted\_ price\_currency : The currency corresponding to the discounted price which a certain course is being offered at.**
* **price\_ detail\_amount : The original price of a particular course.**
* **price\_ detail\_currency : The currency corresponding to the price detail amount for a course.**

**Example: You can get the basic idea how you can create a project from here**

|  |
| --- |
| **Machine Learning Project: Finance & Accounting Courses Analysis**  This project involves analyzing and predicting course performance on a learning platform. Specifically, you will use data related to finance and accounting courses and build a machine learning model to predict **num\_subscribers** or **avg\_rating** based on features such as course title, price, and number of lectures. This project will focus on understanding relationships in the data and predicting course success.  **Steps in the Project:**   1. **Problem Statement**:    1. Predict the number of subscribers or average rating of finance and accounting courses using the available data. 2. **Dataset**:    1. The dataset contains information such as course ID, title, URL, subscription details, pricing, and course content metrics.   ○ Columns: id, title, url, is\_paid, num\_subscribers, avg\_rating, avg\_rating\_recent, rating, num\_reviews, is\_wishlisted, num\_published\_lectures, num\_published\_practice\_tests, created, published\_time, |

|  |
| --- |
| discount\_price\_\_amount, discount\_price\_\_currency, discount\_price\_\_price\_string, price\_detail\_\_amount, price\_detail\_\_currency, price\_detail\_\_price\_string  **Step-by-Step Project Implementation:**  **Step 1: Import Libraries**  First, import the necessary libraries for data manipulation, visualization, and machine learning.  import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns from sklearn.model\_selection import train\_test\_split from sklearn.ensemble import RandomForestRegressor from sklearn.metrics import mean\_squared\_error, r2\_score  **Step 2: Load Dataset**  Load the dataset using pandas. Make sure to inspect the dataset to understand the structure and types of values present.  # Example of loading the dataset df = pd.read\_csv('finance\_accounting\_courses.csv') |

|  |
| --- |
| # View the first few rows df.head()  **Step 3: Data Preprocessing**   1. **Handle Missing Values**   Identify missing values and either fill them or remove rows that contain null values.  # Check for missing values df.isnull().sum()  # Fill or drop missing values df = df.dropna()   1. **Handle Categorical Variables**   Convert columns like is\_paid, discount\_price\_\_currency, price\_detail\_\_currency into numerical format using **Label Encoding** or **One-Hot Encoding**.  # Convert categorical columns to numerical using One-Hot Encoding df = pd.get\_dummies(df, columns=['is\_paid',  'discount\_price\_\_currency', 'price\_detail\_\_currency'], |

|  |
| --- |
| drop\_first=True)  **c. Feature Selection**  We will predict **num\_subscribers** based on the available features, so we separate the independent and dependent variables.  # Select features (independent variables) and target (dependent variable)  X = df[['avg\_rating', 'num\_reviews', 'num\_published\_lectures',  'discount\_price\_\_amount', 'price\_detail\_\_amount']] y = df['num\_subscribers']  **Step 4: Train-Test Split**  Split the data into training and testing sets for model evaluation.  # Split the dataset into training and testing sets X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  **Step 5: Train the Model**  For this project, we'll use a **Random Forest Regressor**, which is a good starting point for regression tasks and can capture complex relationships in the data. |

|  |
| --- |
| # Initialize and train the Random Forest Regressor model = RandomForestRegressor(n\_estimators=100, random\_state=42) model.fit(X\_train, y\_train)  **Step 6: Make Predictions**  Once the model is trained, make predictions using the test set.  # Predicting the number of subscribers using the test set y\_pred = model.predict(X\_test)  **Step 7: Evaluate the Model**  Evaluate the model's performance using common metrics like **Mean Squared Error (MSE)** and **R-squared (R²)**.  # Calculate Mean Squared Error mse = mean\_squared\_error(y\_test, y\_pred) print(f"Mean Squared Error: {mse}")  # Calculate R-squared r2 = r2\_score(y\_test, y\_pred) print(f"R-squared: {r2}") |

|  |
| --- |
| **Step 8: Visualize Results**  You can visualize the actual vs predicted values to see how well the model performs.  plt.scatter(y\_test, y\_pred) plt.xlabel("Actual Number of Subscribers") plt.ylabel("Predicted Number of Subscribers") plt.title("Actual vs Predicted Subscribers") plt.show()  **Project Summary:**   * **Problem**: Predict the number of subscribers for finance and accounting courses based on course characteristics. * **Steps**:   ○ Data loading and preprocessing (handling missing values, converting categorical features).  ○ Feature selection (selecting key features like ratings, reviews, and price).  ○ Model training using **Random Forest Regressor**.  ○ Evaluation of model using **MSE** and **R²**.  ○ Visualization of actual vs predicted subscribers.  **Key Points:**   * This project focuses on **predictive modeling** using regression techniques. * **Feature selection** is critical to improve model performance. * **Random Forest** is a flexible and powerful model that works well for beginners |
| and advanced use cases.  This beginner project is an excellent introduction to predictive analytics in a real-world scenario with online course data. You can expand on this by trying different algorithms (like **Linear Regression** or **XGBoost**) or by analyzing the **avg\_rating** as a target variable instead of subscribers. |

**Example: You can get the basic idea how you can create a project from here**

Sample code and output

|  |
| --- |
| import numpy as np *# linear algebra* import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)* import matplotlib.pyplot as plt import seaborn as sns import plotly.express as px import plotly.graph\_objs as go from plotly.subplots import make\_subplots from sklearn.impute import SimpleImputer  import warnings warnings.filterwarnings('ignore') |

|  |
| --- |
| Dataset Content  Here, I have extracted data related to 10k courses which come under the development category on Udemy's website. The 17 columns in the dataset can be used to gain insights related to:   * **id**: The unique identifier assigned to each course in the dataset. * **title**: The title or name of the course as listed on Udemy's platform. * **url**: The URL of the course on Udemy's website. * **is\_paid**: A boolean value indicating whether the course is paid (True) or free   (False).   * **num\_subscribers**: The number of individuals who have subscribed to the course. * **avg\_rating**: The average rating of the course based on user reviews. * **avg\_rating\_recent**: The recent changes in the average rating of the course. * **num\_reviews**: The total number of reviews or ratings that the course has received. * **is\_wishlisted**: Indicates whether the course is wishlisted by users (True) or not (False). * **num\_published\_lectures**: The total number of lectures available in the course. * **num\_published\_practice\_tests**: The number of practice tests included in the course. * **created**: The timestamp indicating the creation time of the course. * **published\_time**: The timestamp indicating the time when the course was published. * **discounted\_price\_amount**: The discounted price at which the course is being |

offered.

●

**discounted\_price\_currency**

:

The

currency

corresponding

to

the

discounted

price

of

the

course.

●

**discounted\_price\_price\_string**

:

A

formatted

string

representing

the

discounted

price

of

the

course.

●

**price\_detail\_amount**

:

The

original

price

of

the

course

before

any

discounts.

●

**price\_detail\_currency**

:

The

currency

corresponding

to

the

original

price

of

the

course.

●

**price\_detail\_price\_string**

:

A

formatted

string

representing

the

original

price

of

the

course.

In

[2]:

df

=

pd

.

read\_csv(

'/kaggle/input/finance-accounting-courses-udemy-13k

course/udemy\_output\_All\_Finance\_\_Accounting\_p1\_p626.csv'

-

)

In

[3]:

df

.

head()

Out[3]:

i

d

ti

tl

e

ur

l

i

s

\_

p

n

u

m

\_

a

v

g

\_

a

v

g

\_r

r

a

t

i

n

u

m

\_

i

s

\_

w

nu

m\_

pu

blis

num

\_pu

blish

ed\_

c

r

e

a

p

u

b

li

dis

co

unt

\_pr

dis

cou

nt\_

pric

disc

oun

t\_pr

ice\_

pri

ce

\_d

et

pri

ce

\_d

eta

pric

e\_

det

ail\_

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | | | | |
|  |  |  |  |  | a i  d | s u b s cr ib er s | r  a  t i  n g | at in  g  \_r e c e  nt | n g | r  e v  i  e w  s | i  s h  li s t  e d | he  d\_l ect ure s | prac tice \_tes ts | t  e d | s h e d \_ ti  m  e | ice \_\_ am ou  nt | e\_ \_cu  rre  ncy | \_pri ce\_  strin  g | ail  \_\_ a  m  ou  nt | il\_  \_c  urr  en cy | \_pr ice \_st rin  g |
| 0 | 7 6 2 6 1 6 | T h e C  o  m  p  l  e  t  e  S  Q  L B o o  t | /c o  ur se  /t h  eco  m  pl et esq  lb o  ot  ca | T  r  u e | 2 9 5 5 0 9 | 4  .  6 6 0 1 9 | 4.  6 7 8 7 4 | 4  .  6 7 8 7 4 | 7 8 0 0 6 | F a  l  s e | 84 | 0 | 2   1. 1   6  -  0  2  -   1. 4   T  2  2  :  5  7  :  4 | 2  0 1  6  -  0  4  -  0 6  T  0  5  :  1  6  :  1 | 45 5.0 | IN  R | ₹45  5 | 86  40  .0 | IN  R | ₹8,  64  0 |
|  | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | c a  m  p 2  0 2  0  :  G  o fr o  m  Z e  r  o t. .. | m p/ |  |  |  |  |  |  |  |  |  | 8  Z | 1  Z |  |  |  |  |  |  |
| 1 | 9 3 7 6 7 8 | T a b  l  e a | /c  o  ur se  /t a | T  r  u e | 2 0 9 0 7 0 | 4  .  5 8 9 5 | 4.  6 0 0 1 5 | 4  .  6 0 0 1 | 5 4 5 8 1 | F a  l  s e | 78 | 0 | 2 0 1  6  -  0 | 2 0 1  6  -  0 | 45 5.0 | IN  R | ₹45  5 | 86  40  .0 | IN  R | ₹8,  64  0 |
|  | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | u 2 0 2  0  A  -  Z  :  H  a n d s  -  O  n T a b  l  e a u T  r  a  i  n | bl  e a u 1  0/ |  |  | 6 |  | 5 |  |  |  |  | 8  -  2 2  T  1  2  :  1  0  :  1  8  Z | 8  -  2 3  T  1  6  :  5  9  :  4  9  Z |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | i  n g  f  o  ..  . |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 3 6 1 7 9 0 | P  M  P  E  x a  m  P  r  e p S e  m  i  n a  r - | /c o  ur se  /p m pp  m  b ok  6-  3  5p d us  / | T  r  u e | 1 5 5 2 8 2 | 4  .  5 9 4 9 1 | 4.  5 9 3 2 6 | 4  .  5 9 3 2 6 | 5 2 6 5 3 | F a  l  s e | 29  2 | 2 | 2  0  1  7  -  0  9  -  2 6  T  1  6  :  3  2  :  4 8 | 2  0  1  7  -  1  1  -  1 4  T  2  3  :  5  8  :  1 4 | 45 5.0 | IN  R | ₹45  5 | 86  40  .0 | IN  R | ₹8,  64  0 |
|  | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | P  M  B  O  K G  u  i  d e 6 |  |  |  |  |  |  |  |  |  |  | Z | Z |  |  |  |  |  |  |
| 3 | 6 4 8 8 2 6 | T h e C  o  m  p  l  e  t  e F  i  n a | /c  o  ur se  /t h  eco  m  pl et e-  fin  a nc | T  r  u e | 2 4 5 8 6 0 | 4  .  5 4 4 0 7 | 4.  5 3 7 7 2 | 4  .  5 3 7 7 2 | 4 6 4 4 7 | F a  l  s e | 33  8 | 0 | 2 0 1  5  -  1  0  -  2 3  T  1  3  :  3 | 2 0 1  6  -  0  1  -  2 1  T  0  1  :  3 | 45 5.0 | IN  R | ₹45  5 | 86  40  .0 | IN  R | ₹8,  64  0 |
|  | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | n c  i  a l  A n a l  y s  t  C o u  r  s e 2 0 2 0 | ial  -a n  al  ys t-  co ur se  / |  |  |  |  |  |  |  |  |  | 4  :  3 5  Z | 8  :  4 8  Z |  |  |  |  |  |  |
| 4 | 6 3 7 9 | A n E n | /c  o  ur se | T  r  u e | 3 7 4 8 | 4  .  4 7 | 4.  4 7 1 | 4  .  4 7 | 4 1 6 3 | F a  l  s | 83 | 0 | 2 0 1 5 | 2 0 1 6 | 45 5.0 | IN  R | ₹45  5 | 86  40  .0 | IN  R | ₹8,  64  0 |
|  | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | | | | |
|  |  | 3  0 | ti r  e  M  B  A  i  n 1 C  o u r  s e  :  A w  a  r  d  W  i  n n  i  n g  B | /a ne  nti re  -  m  b  ain  -1  -c o  ur se a w  ar dwi n  ni  n  g.  .. |  | 3  6 | 0  8 0 | 7  3 | 1  7 3 | 0 | e |  |  | -  1  0  -  1  2  T  0  6  :  3  9  :  4 6  Z | -  0  1  -  1  1  T  2  1  :  3  9  :  3 3  Z |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | |

u

s

i

n

e

..

.

Dropping

Unnecessary

Columns

We

have

dropped

the

following

columns

from

the

DataFrame:

●

avg\_rating

●

avg\_rating\_recent

●

discount\_price\_\_price\_string

●

price\_detail\_\_price\_string

●

discount\_price\_\_currency

●

price\_detail\_\_currency

●

url

These

columns

were

considered

unnecessary

for

our

analysis,

so

we

have

removed

them

from

the

DataFrame

using

the

drop

method.

In

[4]:

df

.

drop([

'avg\_rating'

,

'avg\_rating\_recent'

,

'discount\_price\_\_price\_string'

,

'price\_detail\_\_price\_string'

,

'discount\_price\_\_currency'

,

'price\_detail\_\_currency'

,

'url'

,

]

inplace

=

True

,

axis

=

1

)

## DataFrame Info

The DataFrame df contains information about its structure and data types. Here's a summary:

In [5]: df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 13608 entries, 0 to 13607 Data columns (total 13 columns):

# Column Non-Null Count Dtype

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

1. id 13608 non-null int64
2. title 13608 non-null object
3. is\_paid 13608 non-null bool
4. num\_subscribers 13608 non-null int64
5. rating 13608 non-null float64
6. num\_reviews 13608 non-null int64
7. is\_wishlisted 13608 non-null bool
8. num\_published\_lectures 13608 non-null int64
9. num\_published\_practice\_tests 13608 non-null int64
10. created 13608 non-null object
11. published\_time 13608 non-null object
12. discount\_price\_\_amount 12205 non-null float64 12 price\_detail\_\_amount 13111 non-null float64 dtypes: bool(2), float64(3), int64(5), object(3) memory usage: 1.2+ MB

## DataFrame Summary Statistics

The describe() method provides summary statistics for numeric columns in the DataFrame df. Here's a summary:

## DataFrame Summary Statistics

The describe() method provides summary statistics for numeric columns in the DataFrame df. Here's a summary:

* **count**: Number of non-null values in each column.
* **mean**: Mean of the values in each column.
* **std**: Standard deviation of the values in each column.
* **min**: Minimum value in each column.
* **25%**: 25th percentile (lower quartile) of the values in each column.
* **50%**: Median (50th percentile) of the values in each column.
* **75%**: 75th percentile (upper quartile) of the values in each column.
* **max**: Maximum value in each column.

This summary provides insight into the distribution and central tendency of numeric data in the DataFrame.

In [6]:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| df.describe()  Out[6]:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | id | num\_s  ubscri bers | rating | num\_  revie  ws | num\_publ ished\_lec tures | num\_publis hed\_practic e\_tests | discount\_ price\_\_a mount | price\_de tail\_\_am ount | | c o u  nt | 1.360  800e  +04 | 13608  .0000  00 | 1360  8.000  000 | 1360  8.000  000 | 13608.00  0000 | 13608.0000  00 | 12205.00  0000 | 13111.0  00000 | | m  e a n | 1.681  721e  +06 | 2847.  01043  5 | 3.912  242 | 243.1  6982  7 | 32.22479  4 | 0.110523 | 493.9437  94 | 4646.99  2602 | | st  d | 9.539  271e  +05 | 9437.  86563  4 | 1.039  237 | 1580.  9658  95 | 42.76691  1 | 0.623501 | 267.8272  60 | 3109.10  1019 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | m  in | 2.762  000e  +03 | 0.000  000 | 0.000  000 | 0.000  000 | 0.000000 | 0.000000 | 455.0000  00 | 1280.00  0000 | | 2  5  % | 8.580 862e  +05 | 62.00  0000 | 3.787  315 | 7.000  000 | 12.00000  0 | 0.000000 | 455.0000  00 | 1600.00  0000 | | 5 0  % | 1.623  421e  +06 | 533.0  00000 | 4.181  735 | 24.00  0000 | 21.00000  0 | 0.000000 | 455.0000  00 | 3200.00  0000 | | 7 5  % | 2.503 720e  +06 | 2279.  50000  0 | 4.452  105 | 87.00  0000 | 37.00000  0 | 0.000000 | 455.0000  00 | 8640.00  0000 | | m  a x | 3.486  006e  +06 | 37483  6.000  000 | 5.000  000 | 7800  6.000  000 | 699.0000  00 | 6.000000 | 3200.000  000 | 12800.0  00000 |   Data  Preprocessing  We  have  converted  the  following  columns  in  the  DataFrame: |

|  |
| --- |
| 1. **created**: Converted to datetime format using pd.to\_datetime. 2. **published\_time**: Converted to datetime format using pd.to\_datetime.   These columns contained datetime information, and we have converted them to datetime format for easier manipulation and analysis.  In [7]:  df['created'] = pd.to\_datetime(df['created'])  df['published\_time'] = pd.to\_datetime(df['published\_time'])  Currency Conversion  We have converted the currency from INR to USD in the following columns:   1. **discount\_price\_\_amount**: Multiplied the actual data by 1/82 to convert from INR to USD. 2. **price\_detail\_\_amount**: Multiplied the actual data by 1/82 to convert from INR to USD.   This conversion was done by multiplying the actual data by the conversion factor of 1/82.  In [8]: df['discount\_price\_\_amount'] = df['discount\_price\_\_amount'] \*  (1/82) df['price\_detail\_\_amount'] = df['price\_detail\_\_amount'] \*  (1/82) |

|  |
| --- |
| Discount Percentage Calculation  We have calculated the discount percentage in the DataFrame using the following steps:  1. **Discount Percentage Calculation**:   * We subtracted the discounted price (discount\_price\_\_amount) from the total price (price\_detail\_\_amount) to get the discount amount. * Divided the discount amount by the total price and multiplied by 100 to get the discount percentage. * Subtracted the discount percentage from 100 to get the percentage of the discount.   In [9]:  df['Discount\_Percentage'] = ((df['price\_detail\_\_amount'] df['discount\_price\_\_amount']) / df['price\_detail\_\_amount']) \*  100  df['Discount\_Percentage'] = 100 - df['Discount\_Percentage']  1. **Imputation of Missing Values**:  ● Missing values were imputed in each column using an appropriate method, such as mean, median, or mode.  These calculations were performed to analyze the extent of discounts offered on the |

|  |
| --- |
| courses and to handle any missing values in the DataFrame.  In [10]:  imputer = SimpleImputer(strategy='median') columns\_to\_impute = ['discount\_price\_\_amount', 'Discount\_Percentage', 'price\_detail\_\_amount'] df[columns\_to\_impute] =  imputer.fit\_transform(df[columns\_to\_impute])  Title Column Cleaning  We have cleaned the 'title' column in the DataFrame using the following steps:   1. **Convert to Lowercase**:    * All text in the 'title' column has been converted to lowercase using the str.lower() method. 2. **Remove Leading and Trailing Whitespace**:    * Any leading and trailing whitespace in the 'title' column has been removed using the str.strip() method. 3. **Replace Multiple Whitespaces with Single Whitespace**:    * Any multiple whitespaces in the 'title' column have been replaced with a single whitespace using the replace() method with the regular expression pattern r'\s+'. 4. **Remove Non-Alphanumeric Characters**:    * Any non-alphanumeric characters in the 'title' column have been removed using the replace() method with the regular expression pattern |

|  |
| --- |
| r'[^\w\s]'.  These cleaning steps were performed to standardize the format of the 'title' column and improve consistency in the data.  In [11]: df['title'] = df['title'].str.lower() df['title'] = df['title'].str.strip().replace(r'\s+', ' ', regex=True) df['title'] = df['title'].str.replace(r'[^\w\s]', '', regex=True)  Feature Extraction : Categorizing our Courses  In this step, we have categorized each course into its specific category using the categorize\_title function. Here are the categories we have identified:   * **Database**: Courses related to SQL, MySQL, or database management. * **Data Visualization**: Courses related to Tableau, Power BI, or data visualization techniques. * **Spreadsheet**: Courses related to Excel or spreadsheet management. * **Project Management**: Courses related to Agile, Scrum, PMP, or project management methodologies. * **Finance**: Courses related to financial management, finance, or accounting. * **Business**: Courses related to MBA, business, or enterprise management. * **Writing**: Courses related to writing, editorial skills, or content creation. * **Sales/Marketing**: Courses related to sales or marketing techniques. |

|  |
| --- |
| * **Data Science**: Courses related to data science, analytics, or machine learning. * **Management**: Courses related to general management principles. * **Leadership**: Courses related to leadership skills. * **Communication**: Courses related to communication skills.   Any courses that do not fit into these specific categories are categorized as 'Other'.  This categorization process helps in organizing and analyzing the courses based on their content and subject matter.  In [12]:  def categorize\_title(title):  title\_lower = title.lower() if 'sql' **in** title\_lower **or** 'mysql' **in** title\_lower **or** 'database' **in** title\_lower: return 'Database'  elif 'tableau' **in** title\_lower **or** 'power bi' **in** title\_lower  **or** 'data viz' **in** title\_lower: return 'Data Visualization'  elif 'excel' **in** title\_lower **or** 'spreadsheet' **in**  title\_lower: return 'Spreadsheet'  elif any(kw **in** title\_lower for kw **in** ['agile', 'scrum', 'pmp', 'project management']): return 'Project Management'  elif any(kw **in** title\_lower for kw **in** ['financial', 'finance', 'accounting']): return 'Finance' |

|  |
| --- |
| elif 'mba' **in** title\_lower **or** 'business' **in** title\_lower **or** 'enterprise' **in** title\_lower: return 'Business'  elif any(kw **in** title\_lower for kw **in** ['write', 'writing', 'editorial']): return 'Writing'  elif 'sale' **in** title\_lower **or** 'marketing' **in** title\_lower: return 'Sales/Marketing'  elif any(kw **in** title\_lower for kw **in** ['data science', 'analytics', 'machine learning']): return 'Data Science'  elif 'management' **in** title\_lower: return 'Management'  elif 'leadership' **in** title\_lower: return 'Leadership'  elif 'communication' **in** title\_lower:  return 'Communication'  else:  return 'Other'  df['category'] = df['title'].apply(categorize\_title)  In [13]:  df['category'].value\_counts() |

|  |
| --- |
| Out[13]: category  Other 8685  Business 1450  Finance 1119  Management 523  Sales/Marketing 516  Project Management 374  Spreadsheet 263  Writing 259  Data Visualization 129  Data Science 99  Leadership 74  Communication 71  Database 46  Name: count, dtype: int64  Top 10 Courses by Rating  We have created a bar chart to visualize the top 10 courses by rating:   * **Title**: Course Title * **Rating**: Rating   The chart provides a visual representation of the top-rated courses based on their |

|  |
| --- |
| ratings.  Insights from the Top 10 Courses by Rating Bar Chart   * **Course Title:** exec 901 introduction to management with slimf it method ■ **Rating:** 5.4 * **Course Title:** are you ready to start your own preschool   ■ **Rating:** Around 5.2   * **Course Title:** energizing your powerful entrepreneurial mindset ■ **Rating:** 5 * **Course Title:** persuasion psychology influence close the deal   ■ **Rating:** Close to 5   * **Course Title:** make money on youtube without making videos 2020 edition ■ **Rating:** Just above 4.8 * **Course Title:** emotional intelligence training for increased sales ■ **Rating:** Around 4.8 * **Course Title:** shipping address and shipping cost for ecommerce series ■ **Rating:** Around 4.8 * **Course Title:** new raise funds for your innovative business with eu grants ■ **Rating:** 4.6 * **Course Title:** price action the complete price action masterclass az ■ **Rating:** 4.6 * **Course Title:** advanced upwork interviews a simple way to earn highpay ■ **Rating:** 4.6   Note: Some ratings are approximated as the exact values are not visible on the bar chart. |

In

[14]:

top\_rated

=

df

.

nlargest(

10

,

'rating'

)

.

sort\_values(

'rating'

,

ascending

=

True

)

fig\_rating

=

px

.

bar(top\_rated,

x

=

'rating'

,

y

=

'title'

,

orientation

=

'h'

,

title

=

'Top

10

Courses

by

Rating'

,

color

=

'rating'

,

labels

=

{

'title'

:

'Course

Title'

,

'rating'

:

'Rating'

})

fig\_rating

.

show()

024

advanced

upwork

interviews

a

simple

way

to

earn

highpayprice

action

the

complete

price

action

masterclass

aznew

raise

funds

for

your

innovative

business

with

eu

grantsshipping

address

and

shipping

cost

for

ecommerce

seriesemotional

intelligence

training

for

increased

salesmake

money

on

youtube

without

making

videos

2020

editionpersuasion

psychology

influence

close

the

dealenergizing

your

powerful

entrepreneurial

mindsetare

you

ready

to

start

your

own

preschoolexec

901

introduction

to

management

with

slimf

it

method

4.64.855.25.4

RatingTop

10

Courses

by

RatingRatingCourse

Title

Top

10

Courses

by

Number

of

Subscribers

We

have

compiled

a

bar

chart

to

showcase

the

top

10

courses

based

on

the

number

of

subscribers:

|  |
| --- |
| * **Title**: Course Title * **Subscribers**: Number of Subscribers   This chart offers a clear depiction of the most popular courses, distinguished by their subscriber count. It illustrates which courses have garnered the most significant following and suggests a correlation between the number of subscribers and the perceived value or demand for these courses.  Insights from the Top 10 Courses by Number of Subscribers Bar Chart   * **Course Title:** an entire mba in 1 course award winning business school prof   ■ **Number of Subscribers:** Around 350k   * **Course Title:** the complete sql bootcamp 2020 go from zero to hero   ■ **Number of Subscribers:** Around 300k   * **Course Title:** stock market investing for beginners   ■ **Number of Subscribers:** Around 250k   * **Course Title:** the complete financial analyst course 2020   ■ **Number of Subscribers:** Between 200k and 250k   * **Course Title:** deep learning prerequisites the numpy stack in python v2   ■ **Number of Subscribers:** Around 200k   * **Course Title:** tableau 2020 az handson tableau training for data science   ■ **Number of Subscribers:** Between 150k and 200k   * **Course Title:** the complete presentation and public speaking/speech course ■ **Number of Subscribers:** Around 150k * **Course Title:** pmp exam prep seminar pmbok guide 6   ■ **Number of Subscribers:** Around 100k   * **Course Title:** introduction to finance accounting modeling and valuation |

■ **Number of Subscribers:** Around 100k

Note: Some numbers are approximations as the exact values were not clear from the bar chart, and the course titles are as read from the OCR results.

In [15]: top\_subscribers\_desc = df.nlargest(10,

'num\_subscribers').sort\_values('num\_subscribers', ascending=True)

fig\_subscribers\_desc = px.bar(top\_subscribers\_desc, x='num\_subscribers', y='title', orientation='h', title='Top 10 Courses by Number

of Subscribers', color='num\_subscribers', labels={'title': 'Course Title',

'num\_subscribers': 'Number of Subscribers'}) fig\_subscribers\_desc.show()

0100k200k300kintroduction to finance accounting modeling and valuationpmp exam prep seminar pmbok guide 6the complete presentation and public speakingspeech coursethe complete financial analyst training investing coursetableau 2020 az handson tableau training for data sciencedeep learning prerequisites the numpy stack in python v2the complete financial analyst course 2020stock market investing for

|  |
| --- |
| beginnersthe complete sql bootcamp 2020 go from zero to heroan entire mba in 1 courseaward winning business school prof  150k200k250k300k350kNumber of SubscribersTop 10 Courses by Number of SubscribersNumber of SubscribersCourse Title  Top 5 Courses by Discount Percentage  The bar chart illustrates the top 5 courses offering the highest discounts. Each course is listed along with the discount percentage provided.   * **Course Title**: advanced financial accounting with tally erp and gst   ■ **Discount Percentage**: 59%   * **Course Title**: pmp 6th edition exam preparation 2020   ■ **Discount Percentage**: 58%   * **Course Title**: data analytics using elasticsearch kibanahands on   ■ **Discount Percentage**: 57%   * **Course Title**: service desk and itil fundamentals   ■ **Discount Percentage**: Approximately between 54.86% ● **Course Title**: sales fundamentals  ■ **Discount Percentage**: Approximately 54.68%  The courses are likely organized from highest to lowest discount, making it easy to see which courses offer the best deals. This type of visualization is helpful for students looking for educational opportunities at reduced prices.  In [16]: top\_subscribers\_desc = df.nlargest(5,  'Discount\_Percentage').sort\_values('Discount\_Percentage', ascending=True) |

fig\_subscribers\_desc

=

px

.

bar(top\_subscribers\_desc,

x

=

'Discount\_Percentage'

,

y

=

'title'

,

orientation

=

'h'

,

title

=

'Top

5

Courses

by

Discount

Percentage'

,

color

=

'Discount\_Percentage'

,

labels

=

{

'title'

:

'Course

Title'

,

'Discount\_Percentage'

:

'Discount

Percentage'

})

fig\_subscribers\_desc

.

show()

0204060

sales

fundamentalsservice

desk

and

itil

fundamentalsdata

analytics

using

elasticsearch

kibanahands

onpmp

6

th

edition

exam

preparation

2020

advanced

financial

accounting

with

tally

erp

and

gst

5556575859

Discount

PercentageTop

5

Courses

by

Discount

PercentageDiscount

PercentageCourse

Title

Most

Expensive

Courses

We

have

curated

a

bar

chart

to

display

the

most

expensive

courses

available:

●

**Course**

**Title**

:

Diversity

Inclusion

Building

a

Grassroots

Foundation

■

**Price**

:

156.4

●

**Course**

**Title**

:

Modern

Marketing

with

Seth

Godin

■

**Price**

:

Just

below

156.4

●

**Course**

**Title**

:

Learn

to

Trade

the

News

■

**Price**

:

156.2

|  |
| --- |
| * **Course Title**: Ally Up Using Allyship to Advance Diversity Inclusion ■ **Price**: Just above 156 * **Course Title**: Presentation Skills Give More Powerful Memorable Talks ■ **Price**: 156 * **Course Title**: Think Like a Leader with Brian Tracy   ■ **Price**: Slightly below 156   * **Course Title**: Speak Like a Pro Public Speaking for Professionals   ■ **Price**: Approximately 155.8   * **Course Title**: Mastering Agile Scrum Project Management ■ **Price**: 155.8 * **Course Title**: Seth Godin's Freelancer Course   ■ **Price**: Just above 155.6   * **Course Title**: Project Management Professional PMP 35 Contact Hours ■ **Price**: 155.6   The chart effectively underlines the high value of these educational programs, potentially linked to the specialized expertise they offer or the professional accreditation they provide. This visual guide is instrumental for learners in pinpointing which courses might require a higher financial commitment.  In [17]: most\_expensive = df.nlargest(10,  'price\_detail\_\_amount').sort\_values('price\_detail\_\_amount', ascending=True) fig\_expensive = px.bar(most\_expensive, x='price\_detail\_\_amount', y='title', orientation='h', title='Most Expensive Courses',  color='price\_detail\_\_amount', |

labels

=

{

'title'

:

'Course

Title'

,

'price\_detail\_\_amount'

:

'Price

(

$)'

})

fig\_expensive

.

show()

050100150

project

management

professional

pmp

35

contact

hoursseth

godins

freelancer

coursemastering

agile

scrum

project

managementspeak

like

a

pro

public

speaking

for

professionalsthink

like

a

leader

with

brian

tracypresentation

skills

give

more

powerful

memorable

talksally

up

using

allyship

to

advance

diversity

inclusionlearn

to

trade

the

newsmodern

marketing

with

seth

godindiversity

inclusion

building

a

grassroots

foundation

155.6155.8156156.2156.4

Price

(

$)Most

Expensive

CoursesPrice

$)Course

(

Title

Outliers

Detection

In

[18]:

numeric\_columns

=

df

.

select\_dtypes(include

=

[

'float64'

,

'int64'

])

.

columns

fig

=

px

.

box(df,

y

=

numeric\_columns,

title

=

'Outliers

Detection

in

Numeric

Features'

)

fig

.

update\_traces(boxmean

=

'sd'

)

*#*

*Shows*

*the*

*standard*

*deviation*

*within*

*the*

*box*

fig

.

show()

|  |
| --- |
| idnum\_subscribersratingnum\_reviewsnum\_published\_lecturesnum\_publish ed\_practice\_testsdiscount\_price\_\_amountprice\_detail\_\_amountDiscount\_P ercentage00.5M1M1.5M2M2.5M3M3.5M  Outliers Detection in Numeric Featuresvariablevalue  Normal Distribution  In [19]:  numeric\_columns = df.select\_dtypes(include=['float64',  'int64']).columns rows = len(numeric\_columns)  fig = make\_subplots(rows=rows, cols=1, subplot\_titles=numeric\_columns)  for i, column **in** enumerate(numeric\_columns, start=1):  fig.add\_trace( go.Histogram(x=df[column], nbinsx=30, name=column,  histnorm='probability density'), row=i, col=1  )  *# Update layout* fig.update\_layout(height=200\*rows, width=900, |

title\_text="Normal Distribution of Numeric Features") fig.show()

00.5M1M1.5M2M2.5M3M3.5M0100n200n300n400n050k100k150k200k250 k300k350k020μ40μ01234500.20.40.60.8010k20k30k40k50k60k70k80k0

50μ100μ150μ200μ010020030040050060000.0050.010.015012345600.51

1.5251015202530354000.20.42040608010012014016000.020.04102030 40506000.050.1

idnum\_subscribersratingnum\_reviewsnum\_published\_lecturesnum\_publish ed\_practice\_testsdiscount\_price\_\_amountprice\_detail\_\_amountDiscount\_P ercentageNormal Distribution of Numeric

Featuresidnum\_subscribersratingnum\_reviewsnum\_published\_lecturesnu m\_published\_practice\_testsdiscount\_price\_\_amountprice\_detail\_\_amount

Discount\_Percentage

## Growth Analysis

Assuming 'published\_time' is your timestamp for when the course was published, you could see how many courses are added per year.

In [20]:

courses\_growth\_over\_time = df.set\_index('published\_time').resample('Y')['id'].count() fig = px.line(courses\_growth\_over\_time, title='Courses Growth Over Time') fig.update\_layout(xaxis\_title='Year', yaxis\_title='Number of

Courses'

)

fig

.

show()

2012201420162018202005001000150020002500

variableidCourses

Growth

Over

TimeYearNumber

of

Courses

Category

Analysis

Number

of

Courses

and

Average

Number

of

Reviews

by

Category

You

can

count

the

number

of

courses

and

calculate

the

average

number

of

reviews

per

category.

In

[21]:

courses\_and\_reviews\_by\_category

=

df

.

groupby(

'category'

)

.

agg({

'title'

:

'count'

,

'num\_reviews'

:

'mean'

})

courses\_and\_reviews\_by\_category

.

columns

=

[

'Courses'

,

'Num

of

Reviews'

]

courses\_and\_reviews\_by\_category

Out[21]:

Cour

ses

Num

of

Reviews

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | category |  |  | | Business | 1450 | 216.8503  45 | | Communicati on | 71 | 575.9154  93 | | Data Science | 99 | 264.2424  24 | | Data  Visualization | 129 | 1497.589  147 | | Database | 46 | 2757.652  174 | | Finance | 1119 | 250.3824  84 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Leadership | 74 | 543.5540  54 | | Management | 523 | 244.9005  74 | | Other | 8685 | 191.0984  46 | | Project  Management | 374 | 624.8074  87 | | Sales/Market ing | 516 | 160.0484  50 | | Spreadsheet | 263 | 299.3726  24 | | Writing | 259 | 402.9305  02 | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Discount Analysis  Calculate the average discount percentage offered per category.  In [22]:  average\_discount\_by\_category = df.groupby('category')['Discount\_Percentage'].mean() average\_discount\_by\_category\_df = average\_discount\_by\_category.reset\_index() average\_discount\_by\_category\_df.columns = ['Category',  'Discount Percentage'] average\_discount\_by\_category\_df  Out[22]:   |  |  |  | | --- | --- | --- | |  | Category | Discount  Percentage | | 0 | Business | 16.824443 | | 1 | Communicati on | 18.325329 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 2 | Data Science | 15.928668 | | 3 | Data  Visualization | 14.563425 | | 4 | Database | 12.963027 | | 5 | Finance | 18.200493 | | 6 | Leadership | 16.589793 | | 7 | Management | 17.208274 | | 8 | Other | 17.519021 | | 9 | Project  Management | 16.801650 | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1  0 | Sales/Market ing | 15.134143 | | 1  1 | Spreadsheet | 18.748882 | | 1  2 | Writing | 16.686187 |   Median Price of Courses by Paid/Free Status  You can find the median price of courses, separated by whether they are paid or free.  In [23]:  paid\_free\_by\_category = df.groupby(['category', 'is\_paid']).size().unstack(fill\_value=0)  *# Rename the columns for better readability* paid\_free\_by\_category.columns = ['Free', 'Paid']  *# Reset the index to make 'category' a regular column* paid\_free\_by\_category.reset\_index(inplace=True) paid\_free\_by\_category |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[23]:   |  |  |  |  | | --- | --- | --- | --- | |  | category | Fr ee | Pai d | | 0 | Business | 7 | 14  43 | | 1 | Communicati on | 0 | 71 | | 2 | Data Science | 0 | 99 | | 3 | Data  Visualization | 0 | 12  9 | | 4 | Database | 0 | 46 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | 5 | Finance | 84 | 10  35 | | 6 | Leadership | 0 | 74 | | 7 | Management | 0 | 52  3 | | 8 | Other | 39  4 | 82  91 | | 9 | Project  Management | 0 | 37  4 | | 1  0 | Sales/Market ing | 1 | 51  5 | | 1  1 | Spreadsheet | 10 | 25  3 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | 1  2 | Writing | 0 | 25  9 |   In [24]:  paid\_free\_melted = pd.melt(paid\_free\_by\_category, id\_vars='category', value\_vars=['Free', 'Paid'], var\_name='Type', value\_name='Count')  fig = px.bar(paid\_free\_melted, x='category', y='Count', color='Type', barmode='group', title='Counts of Paid and Free Courses by  Category', labels={'category': 'Category', 'Count': 'Count of  Courses', 'Type': 'Type'})  fig.update\_layout(xaxis\_title='Category', yaxis\_title='Count of Courses') fig.show()  BusinessCommunicationData ScienceData  VisualizationDatabaseFinanceLeadershipManagementOtherProject  ManagementSales/MarketingSpreadsheetWriting010002000300040005000  600070008000  TypeFreePaidCounts of Paid and Free Courses by CategoryCategoryCount |
| of Courses In [25]:   |  | | --- | | df.to\_csv('Cleaned\_Udemy\_data.csv') | |

[Reference](https://github.com/anthonyng2/udemy-the-complete-machine-learning-course-with-python) [link](https://github.com/anthonyng2/udemy-the-complete-machine-learning-course-with-python)