| 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 here to download data set

**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 website** 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**:
    - Predict the number of subscribers or average rating of finance and accounting courses using the available data.
2. **Dataset**:
    - 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

#### a. 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()
```

#### b. 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 ($R^2$)**.

```
# 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 **$R^2$**.
  - 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.

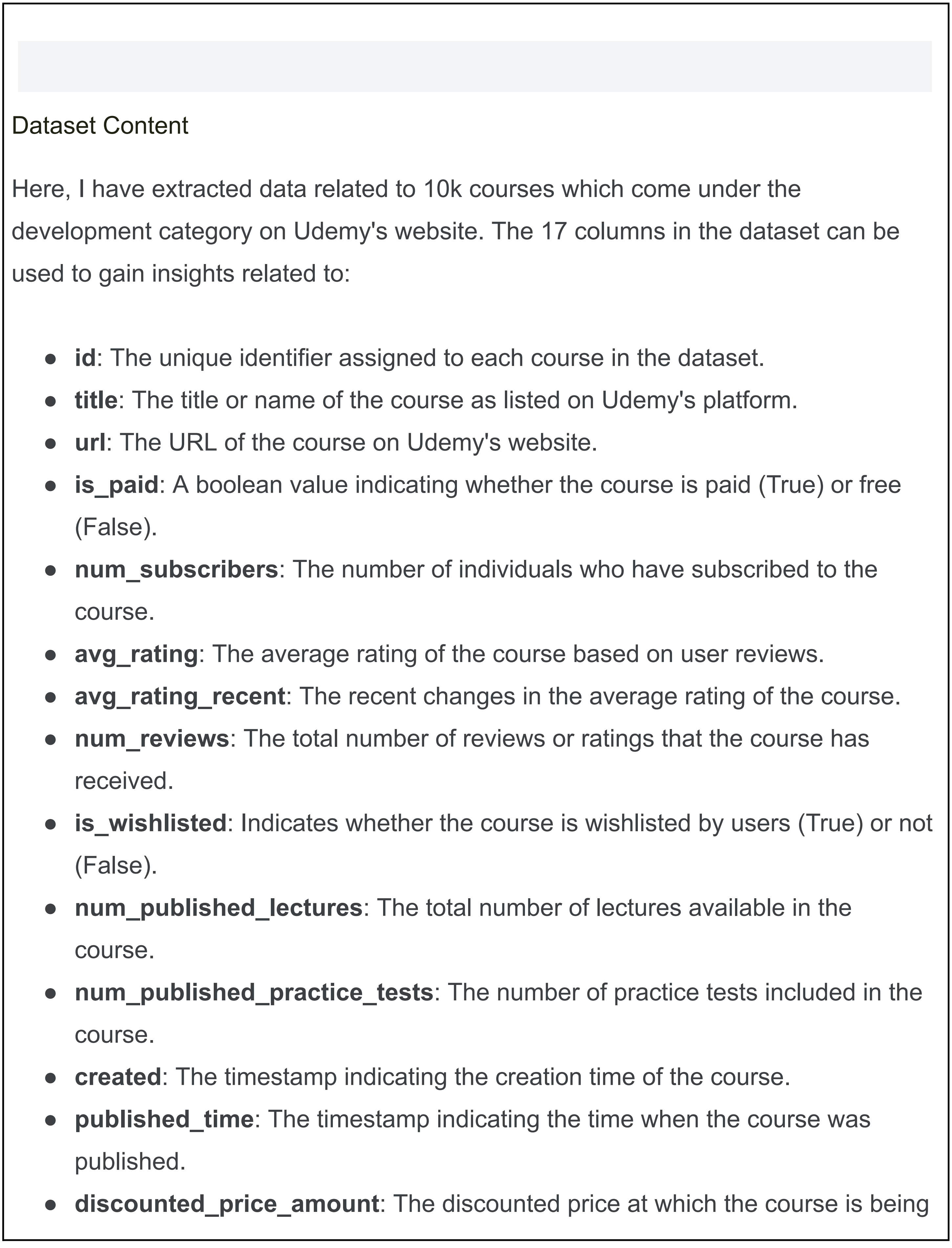
This beginner project is an excellent introduction to predictive analytics in a real-world scenario with online course data. You can expand on this by trying different algorithms (like **Linear Regression** or **XGBoost**) or by analyzing the **avg_rating** as a target variable instead of subscribers.

**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]:
```

| | id | title | url | is_p | num_ | avg_ | avg_r | rati | num_ | is_w | num_publis | num_published_ | crea | publi | discount_pr | discount_pric | discount_price_ | price_det | price_deta | price_detail_ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

|  |  |  |  | aid | subscribers | rating | ating_recent | ng | reviews | ishlisted | hed_lectures | practice_tests | ted | shed_time | ice__amount | e__currency | _price_string | ail__amount | il__currency | _price_string |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 762616 | TheCompleteSQLBoot | /course/the-complete-sql-bootca | True | 295509 | 4.66019 | 4.67874 | 4.67874 | 78006 | False | 84 | 0 | 2016-02-14T22:57:4 | 2016-04-06T05:16:1 | 455.0 | INR | ₹455 | 8640.0 | INR | ₹8,640 |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | camp2020:GofromZerot... | mp/ | | | | | | | | | | 8Z | 1Z | | | | | | |
| 1 | 937678 | Tablea | /course/ta | True | 209070 | 4.5895 | 4.60015 | 4.6001 | 54581 | False | 78 | 0 | 2016-0 | 2016-0 | 455.0 | INR | ₹455 | 8640.0 | INR | ₹8,640 |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | u2020A-Z:Hands-OnTableauTrain | bleau10/ | | | 6 | | 5 | | | | | 8-22T12:10:18Z | 8-23T16:59:49Z | | | | | | |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ingfo... | | | | | | | | | | | | | | | | | | |
| 2 | 1361790 | PMPExamPrepSeminar- | /course/pmp-pmbok6-35-pdus/ | True | 155282 | 4.59491 | 4.59326 | 4.59326 | 52653 | False | 292 | 2 | 2017-09-26T16:32:48 | 2017-11-14T23:58:14 | 455.0 | INR | ₹455 | 8640.0 | INR | ₹8,640 |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PMBOKGuide6 | | | | | | | | | | | Z | Z | | | | | | |
| 3 | 648826 | TheCompleteFina | /course/the-complete-financ | True | 245860 | 4.54407 | 4.53772 | 4.53772 | 46447 | False | 338 | 0 | 2015-10-23T13:3 | 2016-01-21T01:3 | 455.0 | INR | ₹455 | 8640.0 | INR | ₹8,640 |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ncialAnalystCourse2020 | ial-analyst-course/ | | | | | | | | | | 4:35Z | 8:48Z | | | | | | |
| 4 | 6379 | AnEn | /course | True | 3748 | 4.47 | 4.471 | 4.47 | 4163 | Fals | 83 | 0 | 2015 | 2016 | 455.0 | INR | ₹455 | 8640.0 | INR | ₹8,640 |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 30 | tireMBAin1Course:AwardWinningB | /an-eentire-mba-in-1-courseaward-winning.... | | 36 | 080 | 73 | 173 | 0 | e | | | -10-12T06:39:46Z | -01-11T21:39:33Z | | | | | | |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | usine... | | | | | | | | | | | | | | | | | | |

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  
---  ------                       --------------  -----  
 0   id                           13608 non-null  int64  
 1   title                        13608 non-null  object 
 2   is_paid                      13608 non-null  bool   
 3   num_subscribers              13608 non-null  int64  
 4   rating                       13608 non-null  float64
 5   num_reviews                  13608 non-null  int64  
 6   is_wishlisted                13608 non-null  bool   
 7   num_published_lectures       13608 non-null  int64  
 8   num_published_practice_tests 13608 non-null  int64  
 9   created                      13608 non-null  object 
 10  published_time               13608 non-null  object 
 11  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_subscribers | rating | num_reviews | num_published_lectures | num_published_practice_tests | discount_price__amount | price_detail__amount |
|---|---|---|---|---|---|---|---|---|
| count | 1.360800e+04 | 13608.000000 | 13608.000000 | 13608.000000 | 13608.000000 | 13608.000000 | 12205.000000 | 13111.000000 |
| mean | 1.681721e+06 | 2847.010435 | 3.912242 | 243.169827 | 32.224794 | 0.110523 | 493.943794 | 4646.992602 |
| std | 9.539271e+05 | 9437.865634 | 1.039237 | 1580.965895 | 42.766911 | 0.623501 | 267.827260 | 3109.101019 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| min | 2.762000e+03 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 455.000000 | 1280.000000 |
| 25% | 8.580862e+05 | 62.000000 | 3.787315 | 7.000000 | 12.000000 | 0.000000 | 455.000000 | 1600.000000 |
| 50% | 1.623421e+06 | 533.000000 | 4.181735 | 24.000000 | 21.000000 | 0.000000 | 455.000000 | 3200.000000 |
| 75% | 2.503720e+06 | 2279.500000 | 4.452105 | 87.000000 | 37.000000 | 0.000000 | 455.000000 | 8640.000000 |
| max | 3.486006e+06 | 374836.000000 | 5.000000 | 78006.000000 | 699.000000 | 6.000000 | 3200.000000 | 12800.000000 |

## 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()
```

024advanced 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

4.64.855.25.4RatingTop 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()
```

0204060sales fundamentalsservice desk and itil fundamentalsdata analytics using elasticsearch kibanahands onpmp 6th edition exam preparation 2020advanced financial accounting with tally erp and gst

5556575859Discount 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()
```

050100150project 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

155.6155.8156156.2156.4Price ($)Most Expensive CoursesPrice ($)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_published_practice_testsdiscount_price__amountprice_detail__amountDiscount_Percentage00.5M1M1.5M2M2.5M3M3.5M

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()
```

00.5M1M1.5M2M2.5M3M3.5M0100n200n300n400n050k100k150k200k250k300k350k020μ40μ01234500.20.40.60.8010k20k30k40k50k60k70k80k050μ100μ150μ200μ0100200300400500600000.0050.010.01501234560 0.511.525101520253035400 0.20.420406080100120140160 00.020.0410203040506000.050.1

idnum_subscribersratingnum_reviewsnum_published_lecturesnum_published_practice_testsdiscount_price__amountprice_detail__amountDiscount_PercentageNormal Distribution of Numeric Featuresidnum_subscribersratingnum_reviewsnum_published_lecturesnum_published_practice_testsdiscount_price__amountprice_detail__amountDiscount_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]:
```

| | Courses | Num of Reviews |
|---|---|---|

| category | | |
|---|---|---|
| Business | 1450 | 216.850345 |
| Communication | 71 | 575.915493 |
| Data Science | 99 | 264.242424 |
| Data Visualization | 129 | 1497.589147 |
| Database | 46 | 2757.652174 |
| Finance | 1119 | 250.382484 |

| | | |
|---|---|---|
| Leadership | 74 | 543.554054 |
| Management | 523 | 244.900574 |
| Other | 8685 | 191.098446 |
| Project Management | 374 | 624.807487 |
| Sales/Marketing | 516 | 160.048450 |
| Spreadsheet | 263 | 299.372624 |
| Writing | 259 | 402.930502 |

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 | Communication | 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 |

| | | |
|---|---|---|
| 10 | Sales/Marketing | 15.134143 |
| 11 | Spreadsheet | 18.748882 |
| 12 | 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 | Free | Paid |
|---|---|---|---|
| 0 | Business | 7 | 1443 |
| 1 | Communication | 0 | 71 |
| 2 | Data Science | 0 | 99 |
| 3 | Data Visualization | 0 | 129 |
| 4 | Database | 0 | 46 |

| | | | |
|---|---|---|---|
| 5 | Finance | 84 | 1035 |
| 6 | Leadership | 0 | 74 |
| 7 | Management | 0 | 523 |
| 8 | Other | 394 | 8291 |
| 9 | Project Management | 0 | 374 |
| 10 | Sales/Marketing | 1 | 515 |
| 11 | Spreadsheet | 10 | 253 |

|  |  |  |  |
|---|---|---|---|
| 12 | Writing | 0 | 259 |

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/MarketingSpreadsheetWriting010002000300040005000600070008000

TypeFreePaidCounts of Paid and Free Courses by CategoryCategoryCount of Courses

In [25]:

```
df.to_csv('Cleaned_Udemy_data.csv')
```

Reference link