# DPDM Report

## Data Extraction and Cleaning on Coursera

Aakhya Singh-22030242002
Uttkarsh Adhikari-22030242056
Shivam Akulwar-22030242048
Saloni Jain-22030242074
Ankit Mishra-22030242079

14th August, 2022

# Index

**S.No Title**

# Data Extraction and Cleaning on Coursera

## 1. Introduction:

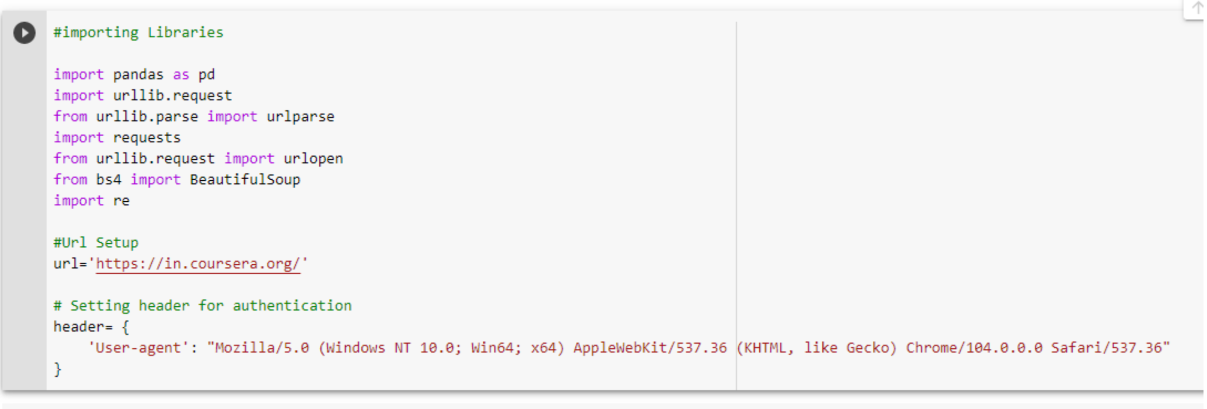
Coursera is a global online learning platform that offers anyone, anywhere, access to online courses and degrees from leading universities and companies.In this report we will be scraping, cleaning, and exploring data. The process of data extraction, cleaning, exploration, and tokenization is implemented in the code below:-

## 2. Datasets and Packages used:

### 2.1. Datasets

- Data was scraped using the BeautifulSoup module and data was extracted from Coursera's sitemap.
- The dataset consists of 712 entries and 17 attributes (like Category,Description,Domain,Enrolled,Instructor name, offered by etc) of the various courses offered.
- Some columns of the dataset had NaN values and we were able to clean it using various data cleaning techniques.

### 2.2. Packages



```
#importing Libraries

import pandas as pd
import urllib.request
from urllib.parse import urlparse
import requests
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

#Url Setup
url='https://in.coursera.org/'

# Setting header for authentication
header= {
    'User-agent': "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36"
}
```

Fig. 1. Packages Used

For this project, we have primarily used the following packages:

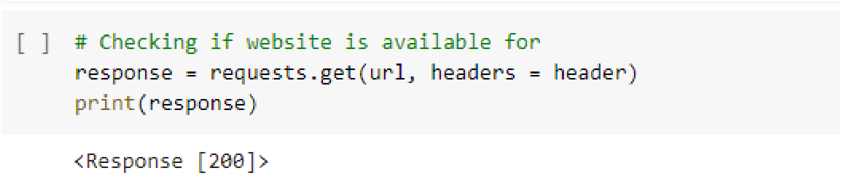
- **Pandas** for analyzing the data.

- The Matplotlib is a package that helps in visualizing and plotting the data. It is for creating static,animated and interactive visualization in python.
- Parsing the HTML and XML data with **BeautifulSoup.** This works by creating a soup object from extracting  the data. The soup object contains the parsed data.
- **Pandas**  is a data wrangling package that helps in performing tasks such as cleaning , visualizing, inspection and normalization.
- The Seaborn module is built on top of matplotlib and helps create more attractive visualizations.

# 3. Process

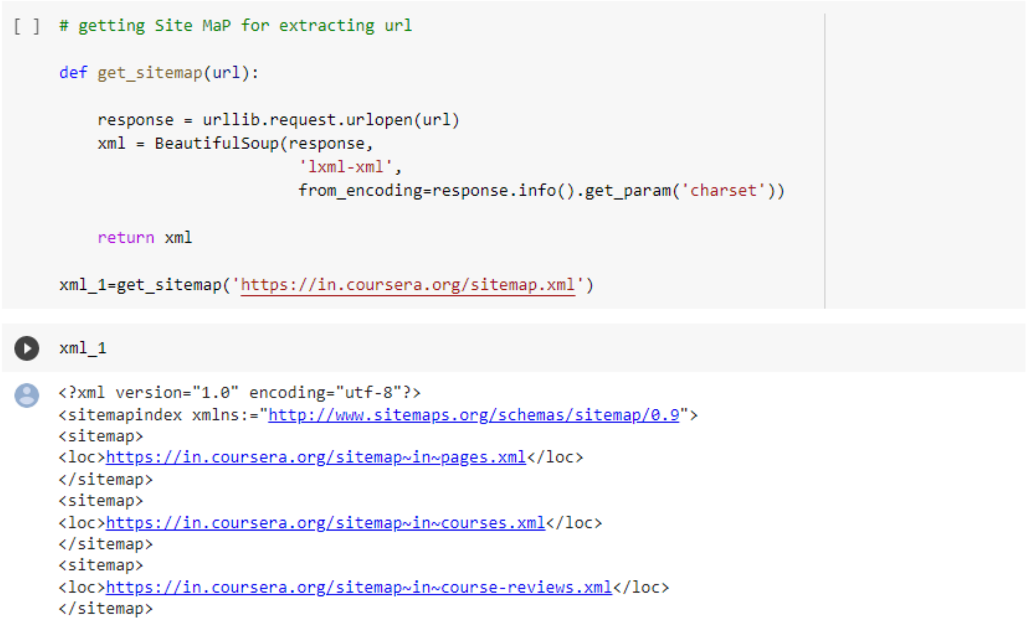
## 3.1. Data Extraction

Data Extraction is defined as the process of collecting data from various sources for the purpose of storing that data, transforming , and feeding it to another system for subsequent analysis.



```
[ ] # Checking if website is available for
    response = requests.get(url, headers = header)
    print(response)

    <Response [200]>
```

Fig. 2. Checking response from website

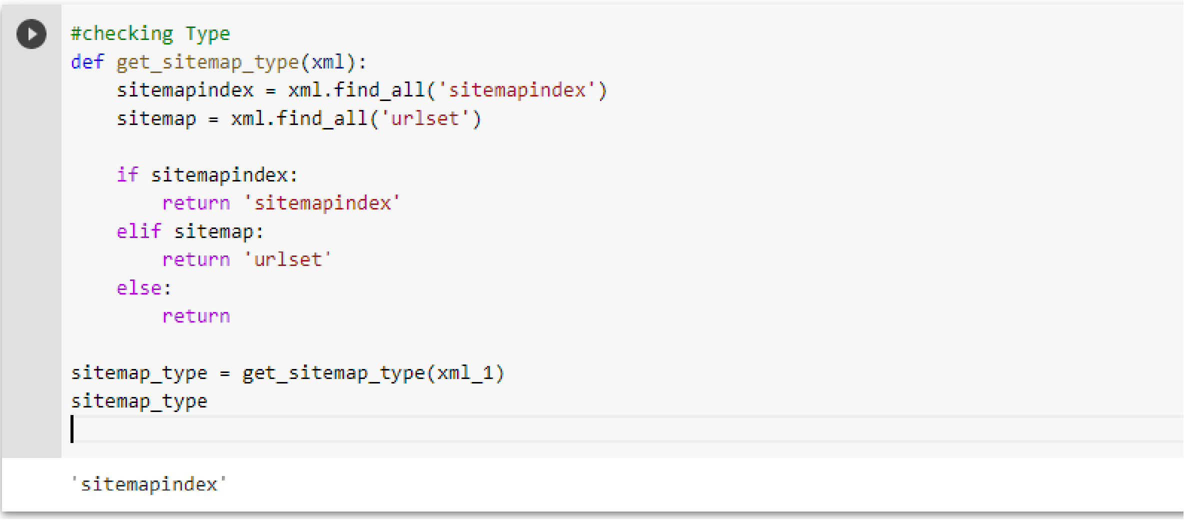


```
[ ] # getting Site MaP for extracting url

    def get_sitemap(url):

        response = urllib.request.urlopen(url)
        xml = BeautifulSoup(response,
                            'lxml-xml',
                            from_encoding=response.info().get_param('charset'))

        return xml

    xml_1=get_sitemap('https://in.coursera.org/sitemap.xml')
```

```
xml_1
```

```
<?xml version="1.0" encoding="utf-8"?>
<sitemapindex xmlns:="http://www.sitemaps.org/schemas/sitemap/0.9">
<sitemap>
<loc>https://in.coursera.org/sitemap~in~pages.xml</loc>
</sitemap>
<sitemap>
<loc>https://in.coursera.org/sitemap~in~courses.xml</loc>
</sitemap>
<sitemap>
<loc>https://in.coursera.org/sitemap~in~course-reviews.xml</loc>
</sitemap>
```

Fig. 3. Retrieving Sitemap



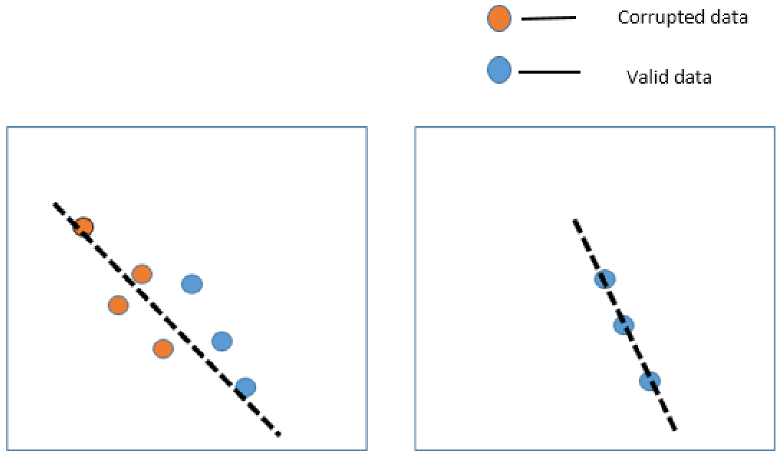
```
#checking Type
def get_sitemap_type(xml):
    sitemapindex = xml.find_all('sitemapindex')
    sitemap = xml.find_all('urlset')

    if sitemapindex:
        return 'sitemapindex'
    elif sitemap:
        return 'urlset'
    else:
        return

sitemap_type = get_sitemap_type(xml_1)
sitemap_type
```

```
'sitemapindex'
```

Fig.4. Checking Sitemap type

## 3.2. Data Cleaning

Data Extraction can be defined as the process of collecting data from various sources for the purpose of storing that data, transforming it, and feeding it to another system for subsequent analysis.

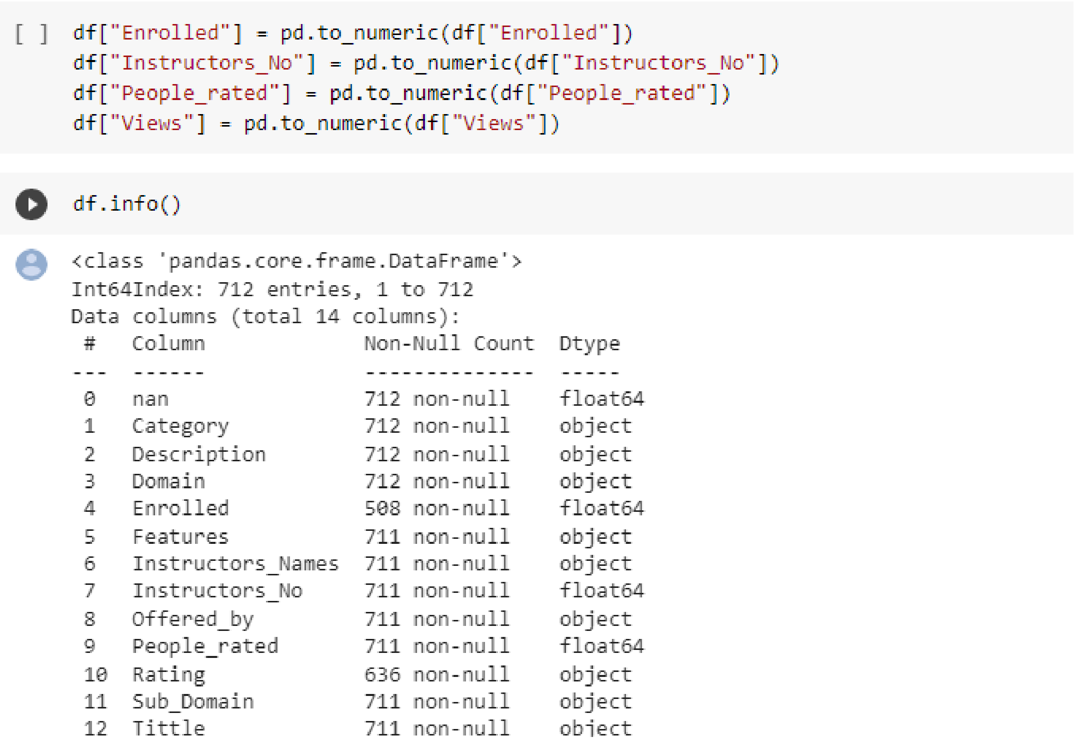


Incorrect model due to corrupted data

Correct model using cleaned data

Fig.5. Corrupted vs Valid Data

- **Converting the data types:** Values of analytical significance such as ‘Enrolled’ , ’Instructors_No’ , ’People_rated’ and ‘Views’ were converted into numeric types as shown below.



```
[ ]  df["Enrolled"] = pd.to_numeric(df["Enrolled"])
     df["Instructors_No"] = pd.to_numeric(df["Instructors_No"])
     df["People_rated"] = pd.to_numeric(df["People_rated"])
     df["Views"] = pd.to_numeric(df["Views"])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 1 to 712
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   nan               712 non-null    float64
 1   Category          712 non-null    object
 2   Description       712 non-null    object
 3   Domain            712 non-null    object
 4   Enrolled          508 non-null    float64
 5   Features          711 non-null    object
 6   Instructors_Names 711 non-null    object
 7   Instructors_No    711 non-null    float64
 8   Offered_by        711 non-null    object
 9   People_rated      711 non-null    float64
 10  Rating            636 non-null    object
 11  Sub_Domain        711 non-null    object
 12  Tittle            711 non-null    object
```

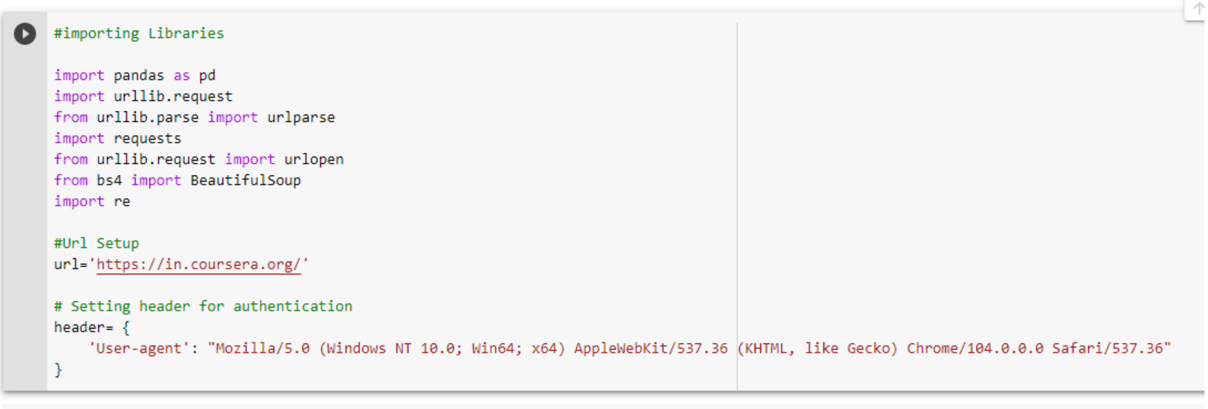
Fig. 6. Conversion to numerical data type

Then we have checked the skewness of the data by using **distplot** -( displays a histogram and shows the distribution of data)

- **Removal of NaN columns:** Columns having all null values were removed.

## 3.3. Data Visualization

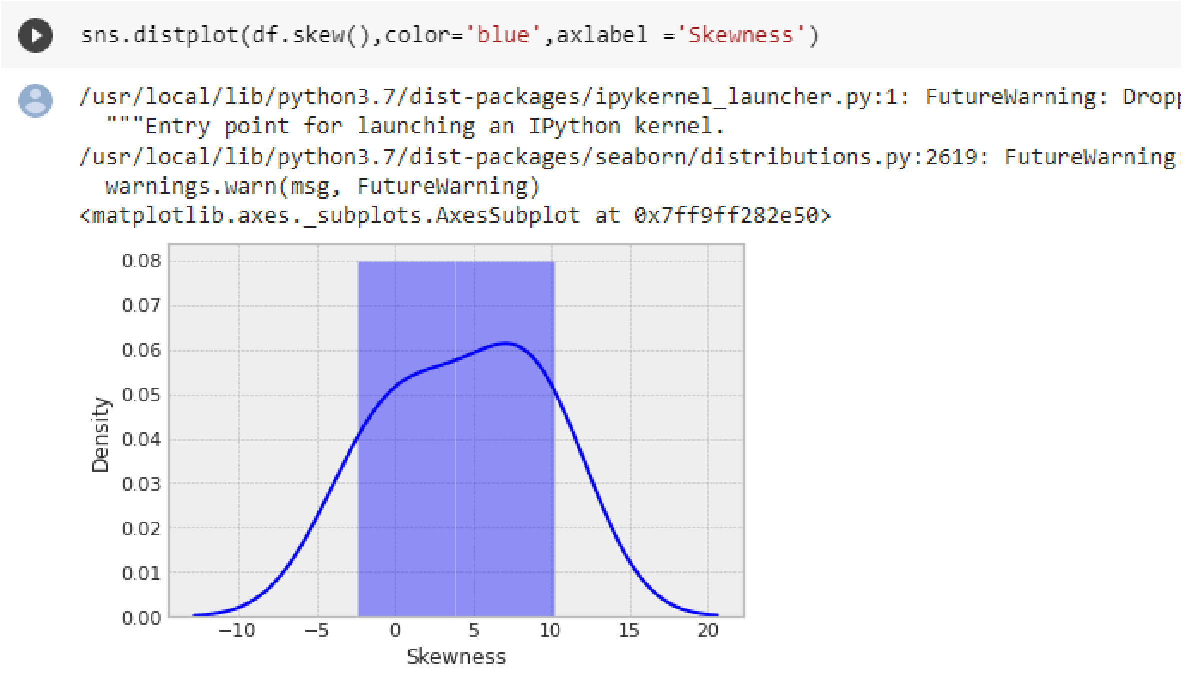
Graphical representation of data and information. It helps in analyzing massive amounts of data and making data- driven decisions. Helps in knowing distribution of variables in the data and find if any relationship exists between different variables.



```
#importing Libraries

import pandas as pd
import urllib.request
from urllib.parse import urlparse
import requests
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re

#Url Setup
url='https://in.coursera.org/'

# Setting header for authentication
header= {
    'User-agent': "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36"
}
```

Fig. 7. URL Setup

As we know, Skewness measures the deviation of a random variable's given distribution from the normal distribution, which is symmetrical on both sides.



```
sns.distplot(df.skew(),color='blue',axlabel ='Skewness')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropp
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7ff9ff282e50>
```

Fig. 8. Skewness of data

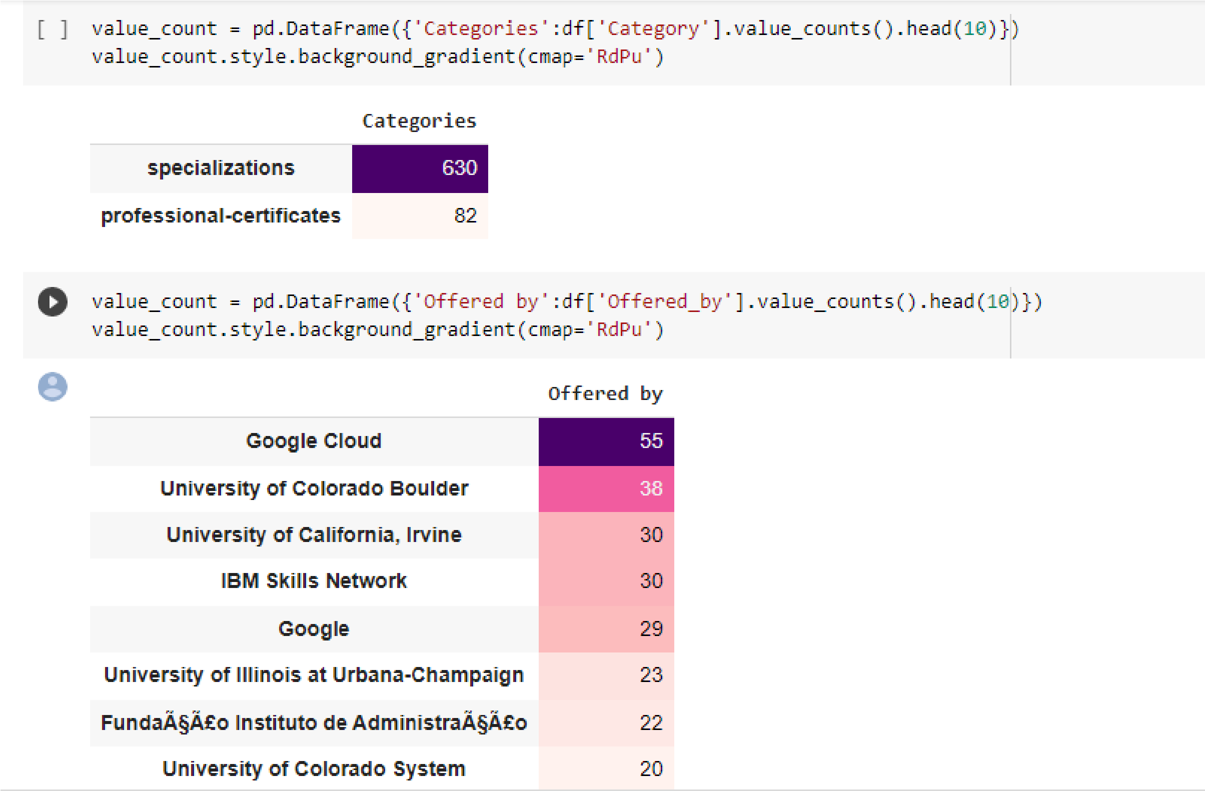
# 4. Analysis

## 4.1. Univariate Analysis

### Why Univariate Analysis?

These analyses help in understanding the location/ position of observations in data variables, its distribution and dispersion.

In the figures below, it is evident that ‘Specialization’ certification courses are preferred by customers more than ‘Professional’ certification courses.. Courses offered by “Google cloud” are the highest in number, whereas “University of Michigan” offered very few courses.



```
[ ]  value_count = pd.DataFrame({'Categories':df['Category'].value_counts().head(10)})
     value_count.style.background_gradient(cmap='RdPu')
```

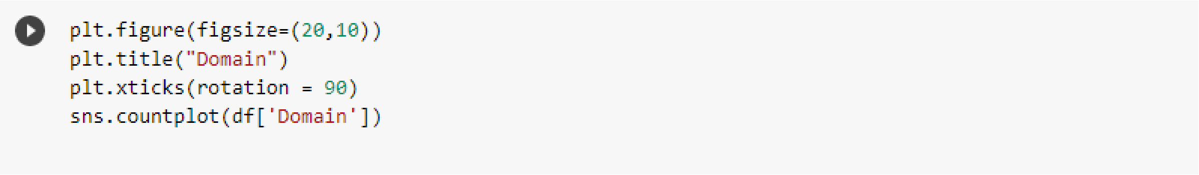
| | Categories |
|---|---|
| specializations | 630 |
| professional-certificates | 82 |

```
value_count = pd.DataFrame({'Offered by':df['Offered_by'].value_counts().head(10)})
value_count.style.background_gradient(cmap='RdPu')
```

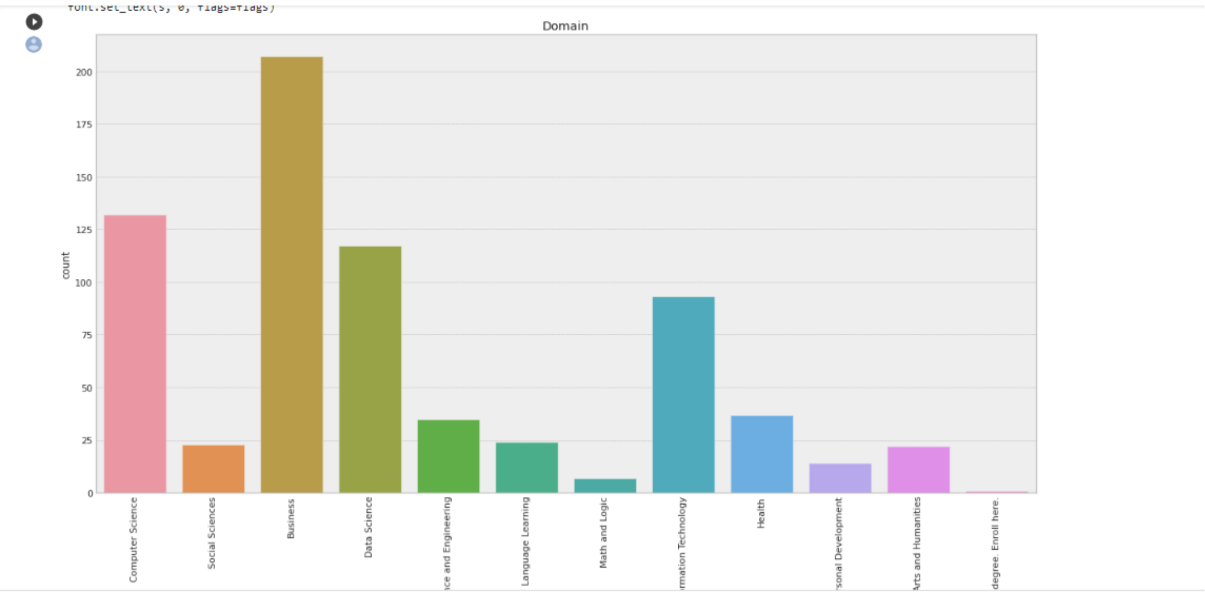
| | Offered by |
|---|---|
| Google Cloud | 55 |
| University of Colorado Boulder | 38 |
| University of California, Irvine | 30 |
| IBM Skills Network | 30 |
| Google | 29 |
| University of Illinois at Urbana-Champaign | 23 |
| FundaÃ§Ã£o Instituto de AdministraÃ§Ã£o | 22 |
| University of Colorado System | 20 |

Fig. 9. Counting values of attributes

Now for analyzing total count in the domain, we have used a Bar chart where we put Domain on the x-axis and Total count on y-axis. We can conclude that Business has the highest total count of 207 and Math and logic having the lowest count of approximately 5.



```
plt.figure(figsize=(20,10))
plt.title("Domain")
plt.xticks(rotation = 90)
sns.countplot(df['Domain'])
```

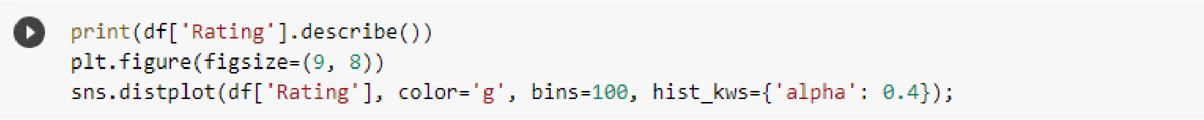


```
font.set_text(s, 0, flags=flags)
```

Fig. 10. Courses offered in various domains

The number of values in ‘Domain’, ’Category’ and ‘Offered By’ were counted and it can be inferred from the bar graph that the ‘Business’ domain has the count in terms of course domains selected by customers whereas the ‘Personal development’ domain has only 14 in total.The figure above depicts the top ten values in the ‘Domain’ attribute.

By the below graph,we can conclude that rating between 4.5 and 5 is the symmetric distribution having the highest density values.



```
print(df['Rating'].describe())
plt.figure(figsize=(9, 8))
sns.distplot(df['Rating'], color='g', bins=100, hist_kws={'alpha': 0.4});
```

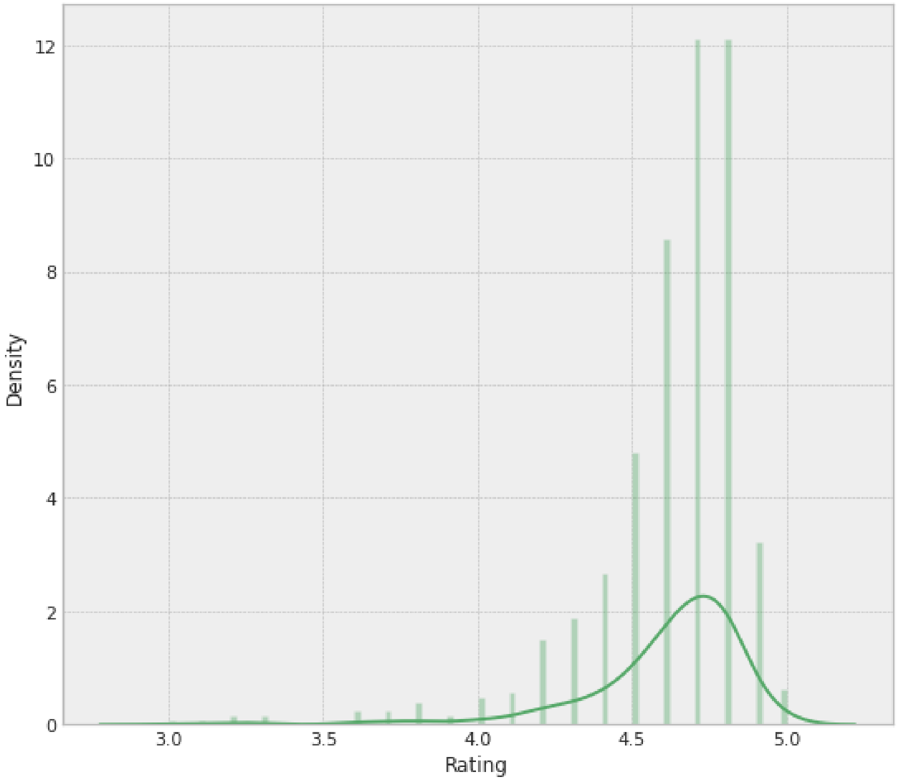
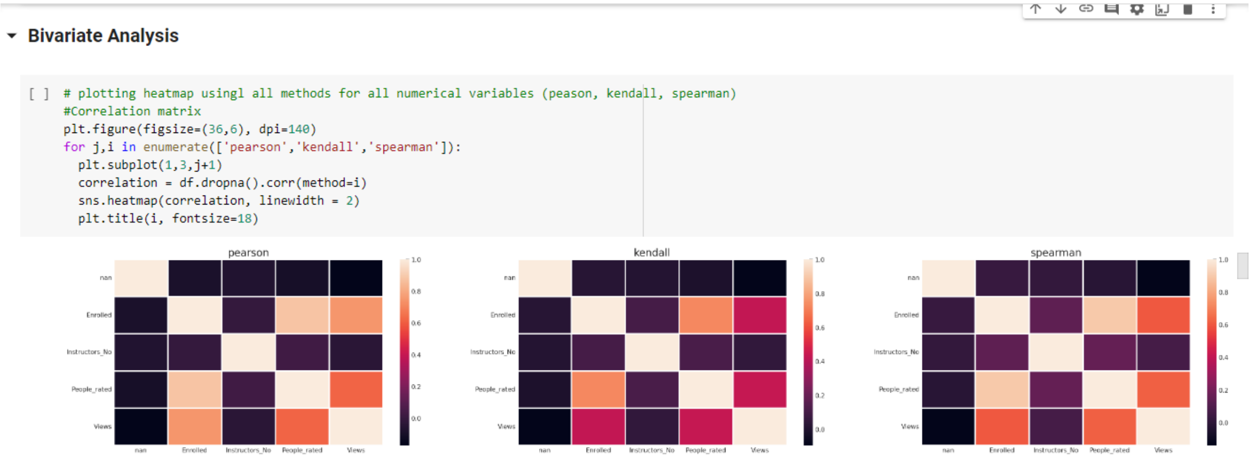


Fig. 11. Concentration of courses based on ratings

- Correlation matrix:

As we know,correlation matrix is the statistical method of defining the relationship and dependence among variables.



```
# plotting heatmap usingl all methods for all numerical variables (peason, kendall, spearman)
#Correlation matrix
plt.figure(figsize=(36,6), dpi=140)
for j,i in enumerate(['pearson','kendall','spearman']):
  plt.subplot(1,3,j+1)
  correlation = df.dropna().corr(method=i)
  sns.heatmap(correlation, linewidth = 2)
  plt.title(i, fontsize=18)
```

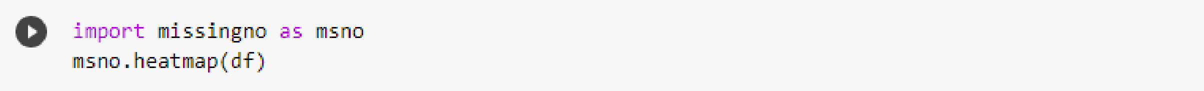
Fig. 12. Correlation matrix

## 4.2. Bivariate Analysis

Used to find the correlation between two variables through scatterplot and more information can be seen by hover method by pointing out .

- **HeatMap**

In a dataset with large number of variables, it is better to plot a Correlation Heatmap to understand the relation between the variables



```
import missingno as msno
msno.heatmap(df)
```

In the below Heatmap, we can see that a darker blue color represents a higher positive correlation whereas a darker orange color represents a higher negative correlation and white color represents no correlation.

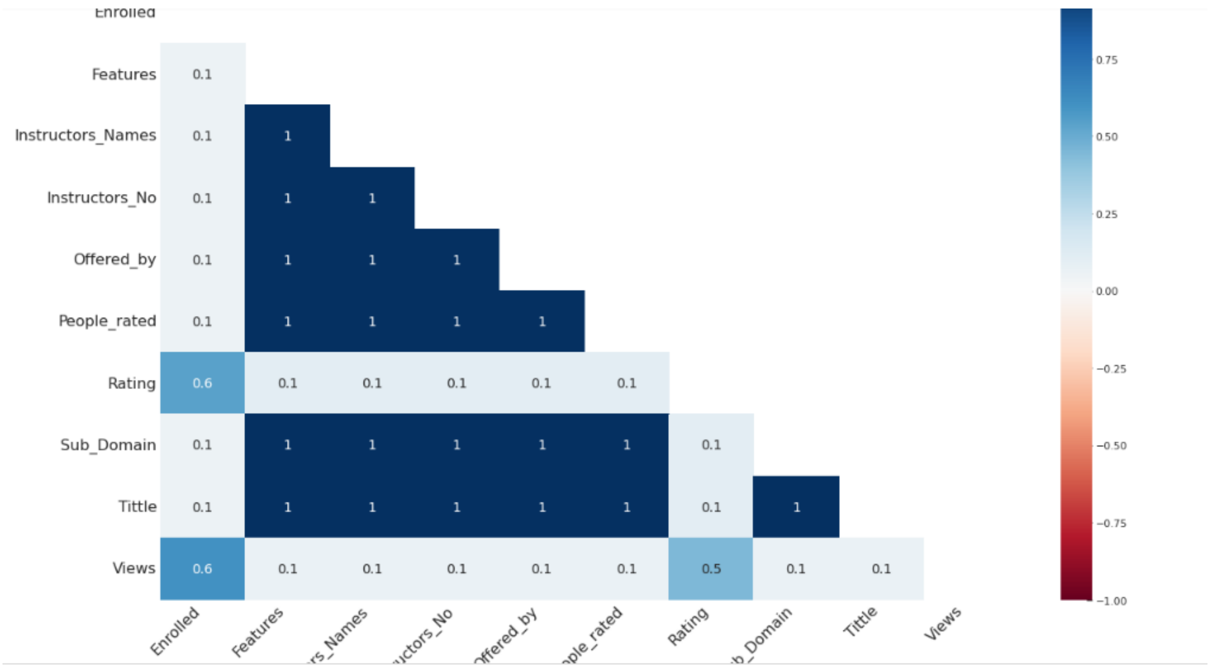


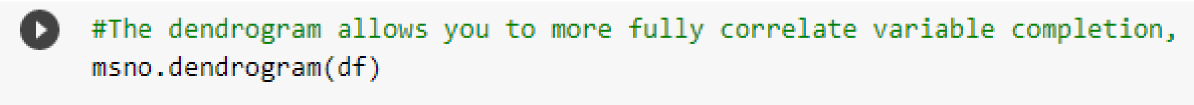
Fig. 13. Heatmap of DataFrame

Some Variables with High Positive Correlation with each others are-

- Features with Instructor_Names,Instructor_No,Offered_By,People_rated,Sub_Domain and title.
- Instructor_Names with Instructor_No,Offered_By,People_rated,Sub_Domain and title.
- Instructor_No with Offered_By,People_rated,Sub_Domain and title.
- Offered_By with People_rated,Sub_Domain and title.
- People_rated with Sub_Domain and title.
- Sub_Domain with Title.

- **Dendrogram**

The dendrogram allows you to more fully correlate variable completion, revealing trends deeper than the pairwise ones visible in the correlation heatmap.



```
#The dendrogram allows you to more fully correlate variable completion,
msno.dendrogram(df)
```

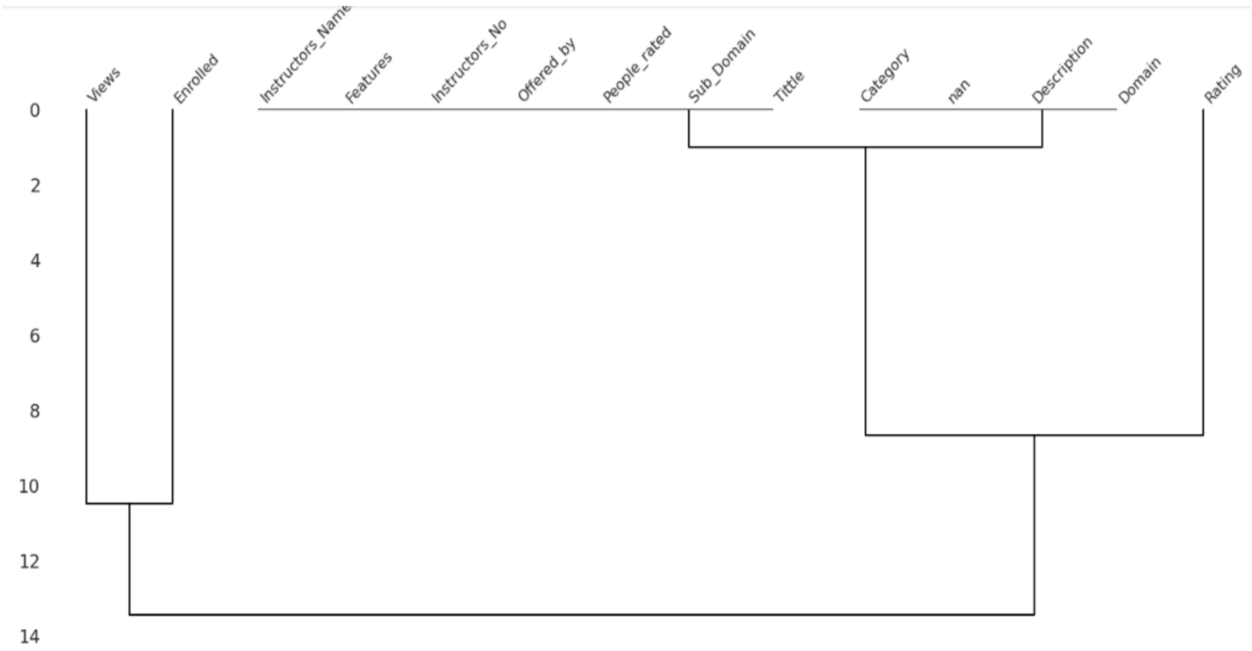
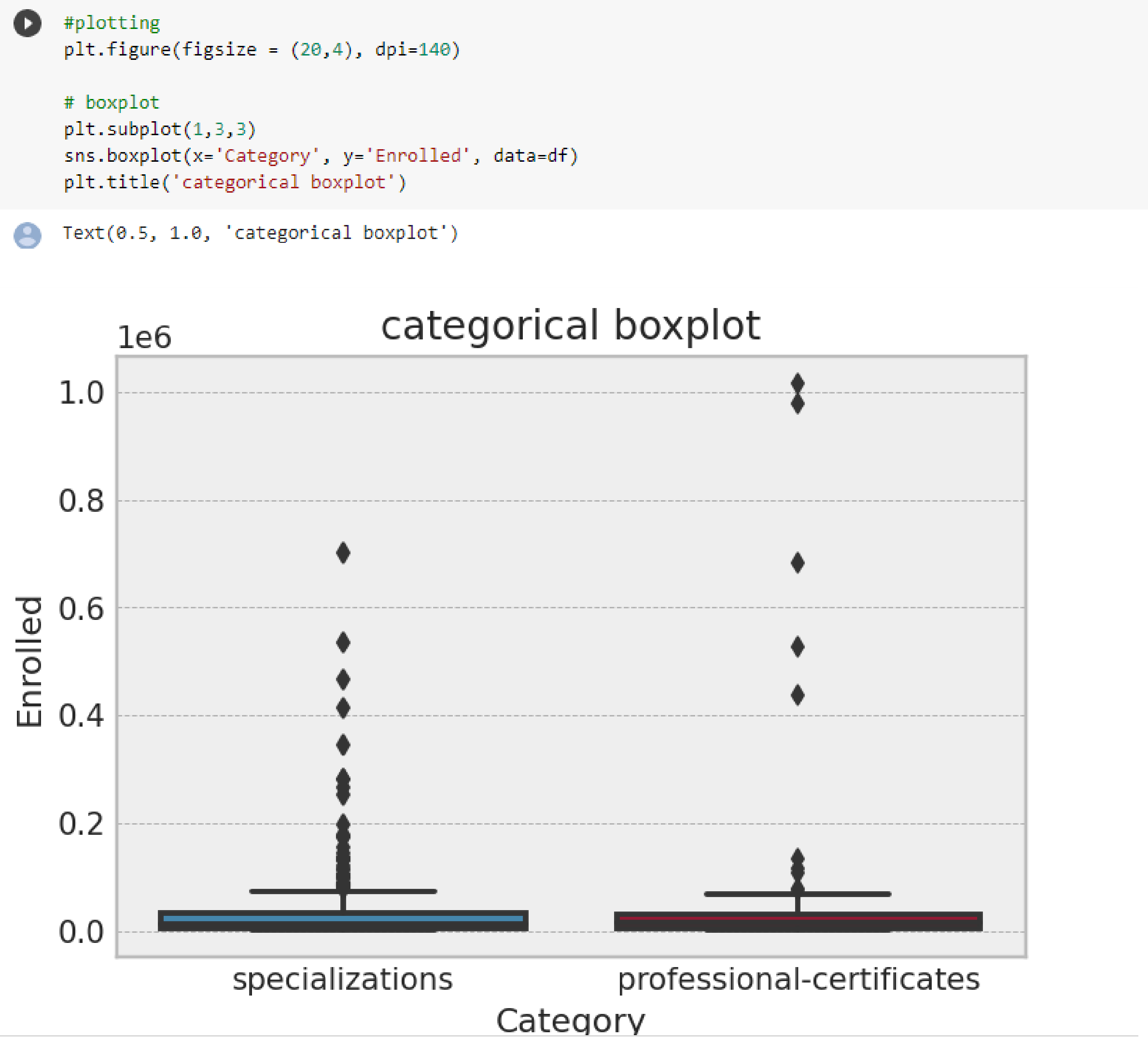


Fig. 14. Dendrogram

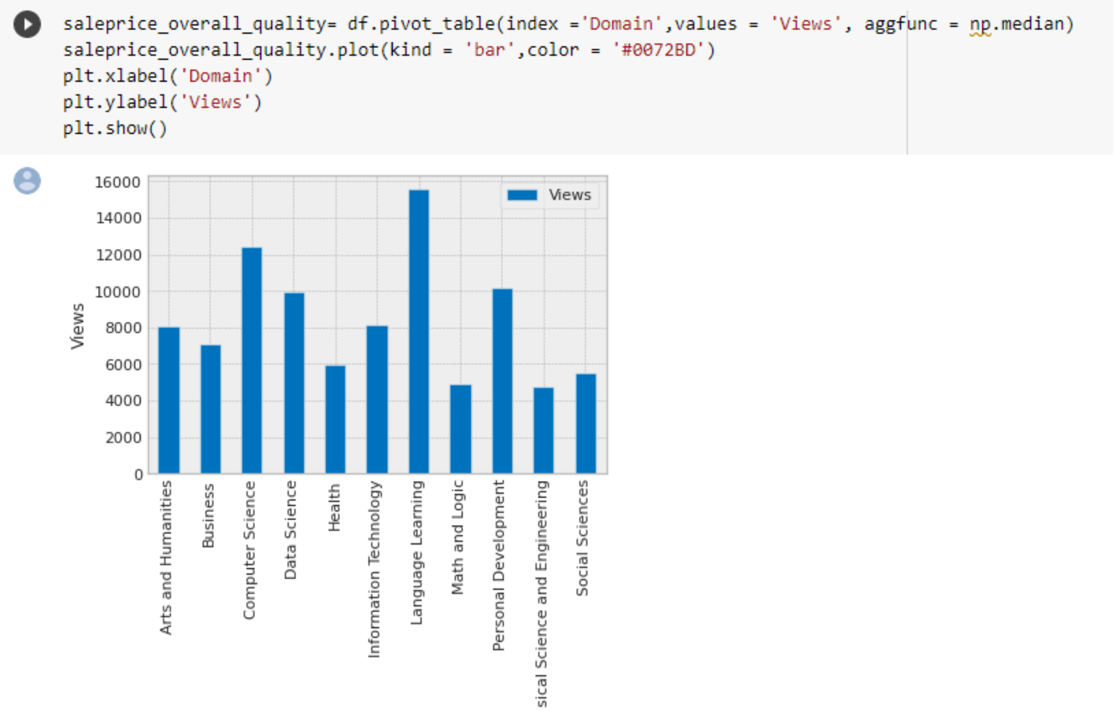
- **Categorical boxplot:**



```
#plotting
plt.figure(figsize = (20,4), dpi=140)

# boxplot
plt.subplot(1,3,3)
sns.boxplot(x='Category', y='Enrolled', data=df)
plt.title('categorical boxplot')
```

```
Text(0.5, 1.0, 'categorical boxplot')
```

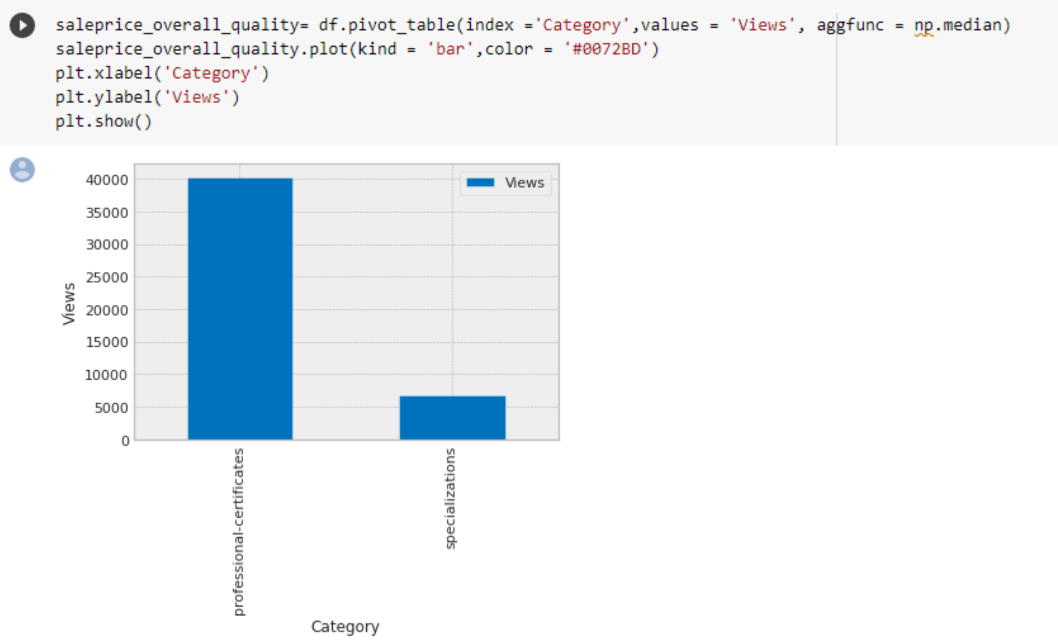
Fig. 15. Box Plotting categorical variables



```
saleprice_overall_quality= df.pivot_table(index ='Domain',values = 'Views', aggfunc = np.median)
saleprice_overall_quality.plot(kind = 'bar',color = '#0072BD')
plt.xlabel('Domain')
plt.ylabel('Views')
plt.show()
```

Fig. 16. Bar Graph

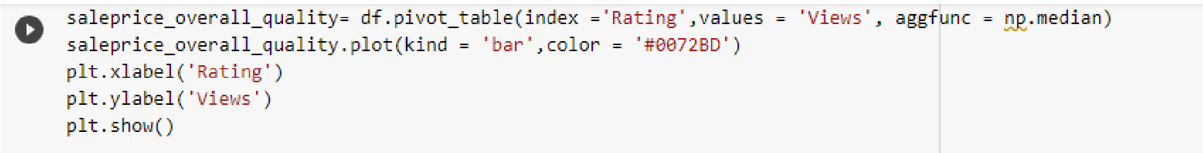
Fig. 16 depicts the number of views in courses from different domains. Language learning has the highest number of views.



```
saleprice_overall_quality= df.pivot_table(index ='Category',values = 'Views', aggfunc = np.median)
saleprice_overall_quality.plot(kind = 'bar',color = '#0072BD')
plt.xlabel('Category')
plt.ylabel('Views')
plt.show()
```

Fig. 17. Bivariate analysis of Category Vs Views

Fig. 17 shows a comparison between the two different certification options and the customer views in each. Professional certifications have pulled in more customers than specialization certifications.



```
saleprice_overall_quality= df.pivot_table(index ='Rating',values = 'Views', aggfunc = np.median)
saleprice_overall_quality.plot(kind = 'bar',color = '#0072BD')
plt.xlabel('Rating')
plt.ylabel('Views')
plt.show()
```

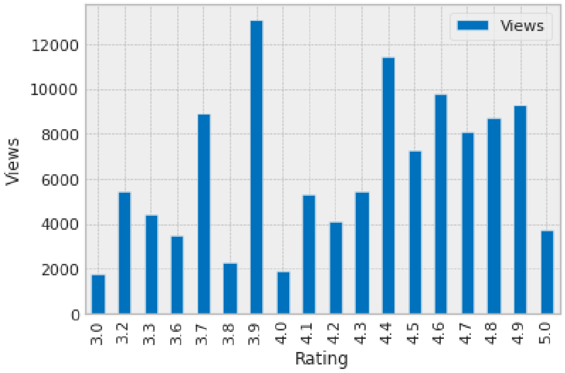


Fig. 18. Rating Vs Views

Fig. 18 depicts the number of views in each bracket of ratings given to the course. It can be seen that courses with rating 3.9 got the highest customer views.

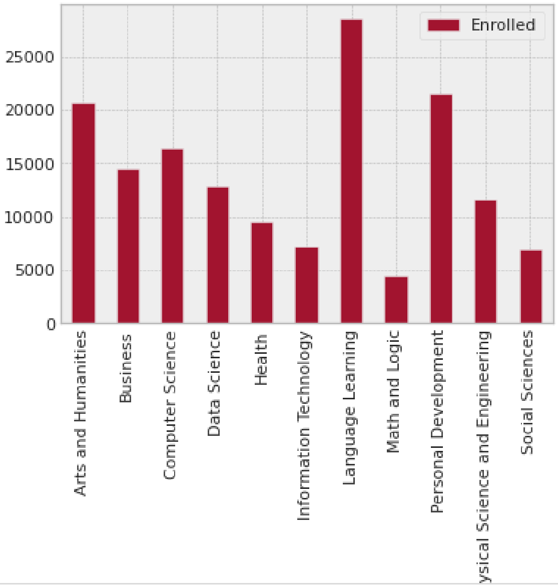
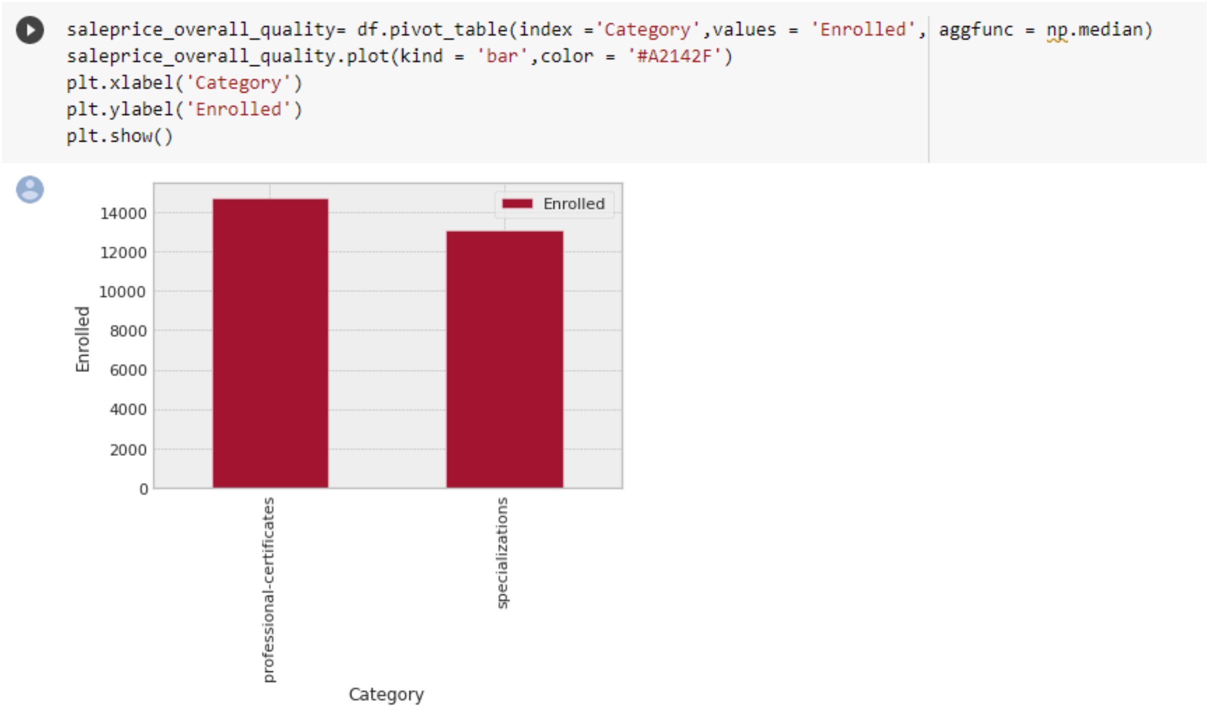


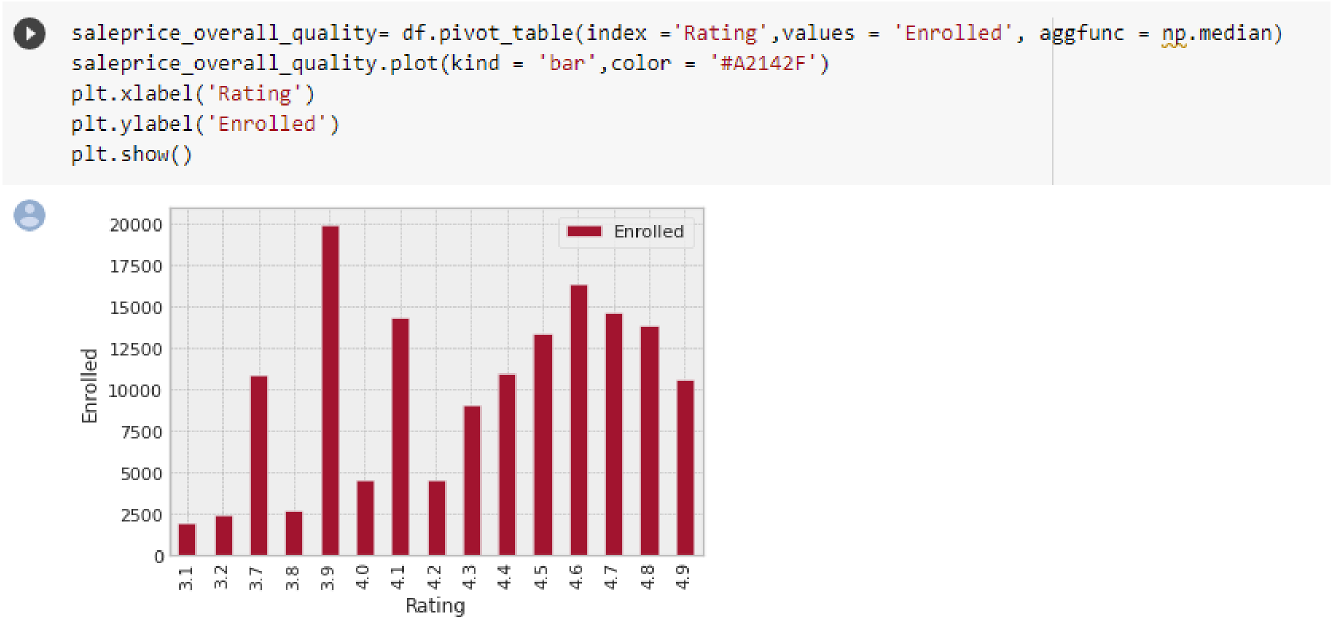
Fig. 19. Enrolled Vs Domain

Figure 19 depicts the number of customers that enrolled in a particular course from the different Domains. Most courses were from the Language Learning domain. This also shows that the course with most views also had the highest number of enrolled customers.



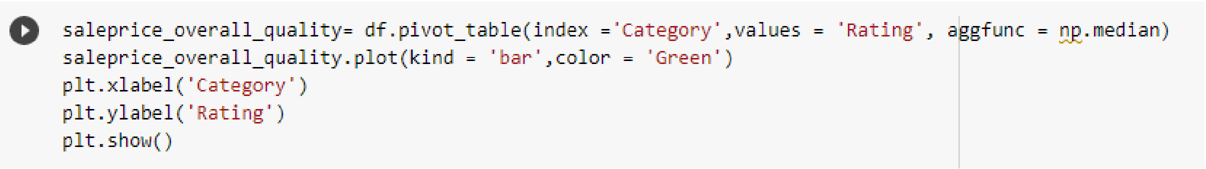
```
saleprice_overall_quality= df.pivot_table(index ='Category',values = 'Enrolled', aggfunc = np.median)
saleprice_overall_quality.plot(kind = 'bar',color = '#A2142F')
plt.xlabel('Category')
plt.ylabel('Enrolled')
plt.show()
```

Figure 20 helps us analyze the most customers are enrolled in Professional certification courses.



```
saleprice_overall_quality= df.pivot_table(index ='Rating',values = 'Enrolled', aggfunc = np.median)
saleprice_overall_quality.plot(kind = 'bar',color = '#A2142F')
plt.xlabel('Rating')
plt.ylabel('Enrolled')
plt.show()
```

Fig. 21. Ratings vs Enrolled customers



```
saleprice_overall_quality= df.pivot_table(index ='Category',values = 'Rating', aggfunc = np.median)
saleprice_overall_quality.plot(kind = 'bar',color = 'Green')
plt.xlabel('Category')
plt.ylabel('Rating')
plt.show()
```

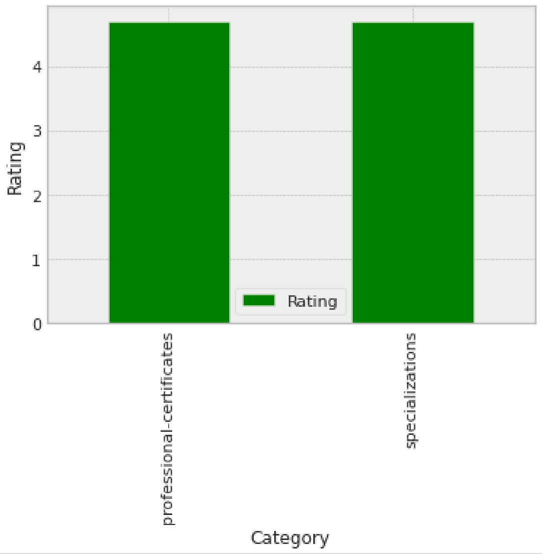
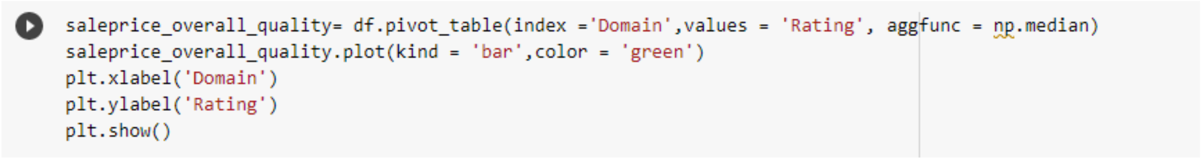


Fig. 22. Category vs Ratings

We create a pivot table for the ‘Category’ which consists of  Domain and Rating . Then we plot both the variables on the  graph where  ‘Domain’ lies on the x-axis and ratings on the y-axis.We can conclude that almost every domain  got a rating more than 4.



```
saleprice_overall_quality= df.pivot_table(index ='Domain',values = 'Rating', aggfunc = np.median)
saleprice_overall_quality.plot(kind = 'bar',color = 'green')
plt.xlabel('Domain')
plt.ylabel('Rating')
plt.show()
```

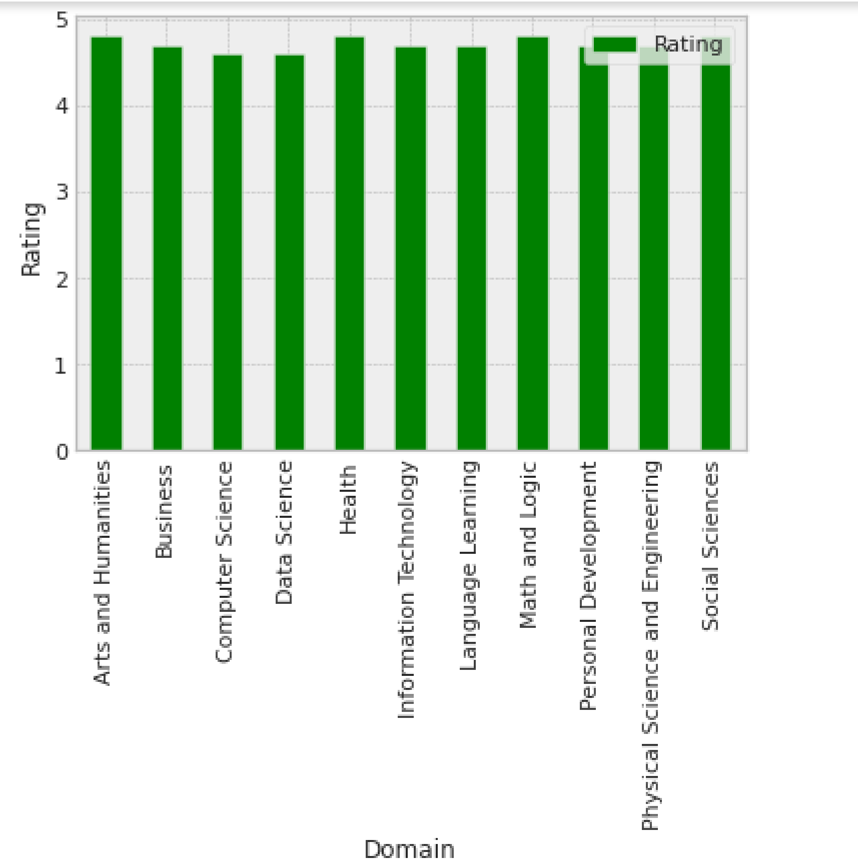


Fig. 23. Domain vs Ratings

Next we  have created a line graph based on ratings and enrollment . As we can see, we plot the variable 'Rating' on x-axis and 'Enrolled' on y-axis. We can conclude that 20000 enrolled people given the courses rating as 3.9 which is the highest out of all. As we move towards ratings greater than 4.5,we can see the fall in the enrolled users.

## 5. Conclusion

From the analysis performed, several insights could be drawn. Most courses having higher views had a higher number of enrolled customers. It is quite obvious that the customer preference towards a particular course was based on the ratings. Professional certification courses were preferred over specialization courses and more customers enrolled into Language Learning courses.

In this project of experiential learning, we have learned how to get unstructured data from websites and social media and transform unstructured data into structured data that can be easily represented as graphs. On datasets that were taken from Coursera, data cleaning and analysis has been done. We have learned numerous methods for obtaining data in raw format, cleaning the raw data, and visualizing structured data as graphs and charts.

## Resources:

- www.coursera.org
- Assignment Link