Importing the Data

In [1]:
```
pip install irdatacleaning
```

```
Requirement already satisfied: irdatacleaning in c:\users\pc\anaconda3\lib\site-packages
(2022.1.19)
Requirement already satisfied: scikit-learn in c:\users\pc\anaconda3\lib\site-packages
(from irdatacleaning) (0.24.2)
Requirement already satisfied: numpy in c:\users\pc\anaconda3\lib\site-packages (from ir
datacleaning) (1.20.3)
Requirement already satisfied: pandas in c:\users\pc\anaconda3\lib\site-packages (from i
rdatacleaning) (1.3.4)
Requirement already satisfied: matplotlib in c:\users\pc\anaconda3\lib\site-packages (fr
om irdatacleaning) (3.4.3)
Requirement already satisfied: IslanderDataPreprocessing in c:\users\pc\anaconda3\lib\si
te-packages (from irdatacleaning) (2022.1.18)
Requirement already satisfied: opencv-python in c:\users\pc\anaconda3\lib\site-packages
(from irdatacleaning) (4.6.0.66)
Requirement already satisfied: kaggle in c:\users\pc\anaconda3\lib\site-packages (from I
slanderDataPreprocessing->irdatacleaning) (1.5.12)
Requirement already satisfied: urllib3 in c:\users\pc\anaconda3\lib\site-packages (from
kaggle->IslanderDataPreprocessing->irdatacleaning) (1.26.7)
Requirement already satisfied: certifi in c:\users\pc\anaconda3\lib\site-packages (from
kaggle->IslanderDataPreprocessing->irdatacleaning) (2021.10.8)
Requirement already satisfied: tqdm in c:\users\pc\anaconda3\lib\site-packages (from kag
gle->IslanderDataPreprocessing->irdatacleaning) (4.62.3)
Requirement already satisfied: python-slugify in c:\users\pc\anaconda3\lib\site-packages
(from kaggle->IslanderDataPreprocessing->irdatacleaning) (5.0.2)
Requirement already satisfied: six>=1.10 in c:\users\pc\anaconda3\lib\site-packages (fro
m kaggle->IslanderDataPreprocessing->irdatacleaning) (1.16.0)
Requirement already satisfied: requests in c:\users\pc\anaconda3\lib\site-packages (from
kaggle->IslanderDataPreprocessing->irdatacleaning) (2.26.0)
Requirement already satisfied: python-dateutil in c:\users\pc\anaconda3\lib\site-package
s (from kaggle->IslanderDataPreprocessing->irdatacleaning) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\pc\anaconda3\lib\site-packages
(from matplotlib->irdatacleaning) (0.10.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\pc\anaconda3\lib\site-packag
es (from matplotlib->irdatacleaning) (3.0.4)
Requirement already satisfied: pillow>=6.2.0 in c:\users\pc\anaconda3\lib\site-packages
(from matplotlib->irdatacleaning) (8.4.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\pc\anaconda3\lib\site-packa
ges (from matplotlib->irdatacleaning) (1.3.1)
Requirement already satisfied: pytz>=2017.3 in c:\users\pc\anaconda3\lib\site-packages
(from pandas->irdatacleaning) (2021.3)
Requirement already satisfied: text-unidecode>=1.3 in c:\users\pc\anaconda3\lib\site-pac
kages (from python-slugify->kaggle->IslanderDataPreprocessing->irdatacleaning) (1.3)
Requirement already satisfied: idna<4,>=2.5 in c:\users\pc\anaconda3\lib\site-packages
(from requests->kaggle->IslanderDataPreprocessing->irdatacleaning) (3.2)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\pc\anaconda3\lib\si
te-packages (from requests->kaggle->IslanderDataPreprocessing->irdatacleaning) (2.0.4)
Requirement already satisfied: joblib>=0.11 in c:\users\pc\anaconda3\lib\site-packages
```

```
(from scikit-learn->irdatacleaning) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\pc\anaconda3\lib\site-pa
ckages (from scikit-learn->irdatacleaning) (2.2.0)
Requirement already satisfied: scipy>=0.19.1 in c:\users\pc\anaconda3\lib\site-packages
(from scikit-learn->irdatacleaning) (1.7.1)
Requirement already satisfied: colorama in c:\users\pc\anaconda3\lib\site-packages (from
tqdm->kaggle->IslanderDataPreprocessing->irdatacleaning) (0.4.4)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import pandas as pd
import numpy as np
import irdatacleaning
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
cd "E:\Cube_Statistica\Professional-Training-for-Data-Science-Team-
Statistics"
```

```
E:\Cube_Statistica\Professional-Training-for-Data-Science-Team-Statistics
```

In [4]:
```
import os

cwd = os.getcwd()
print(cwd)
```

```
E:\Cube_Statistica\Professional-Training-for-Data-Science-Team-Statistics
```

In [5]:
```
raw_data = pd.read_csv("Data/EDA.csv", encoding= "unicode_escape")
```

In [6]:
```
raw_data.head()
```

Out[6]:

	Unnamed: 0	Timestamp	Student ID	Do you understand that this is a paid course	Course Fee Email	Date sent	Payment Receipt Sent	How and where did you hear about this course	Which Country are you currently residing in	W Ci cur res
0	1	8/25/2022 14:35	1	Yes	Sent	8/26/2022	Yes	Through a friend	Pakistan	K

	Unnamed: 0	Timestamp	Student ID	Do you understand that this is a paid course	Course Fee Email	Date sent	Payment Receipt Sent	How and where did you hear about this course	Which Country are you currently residing in	W Ci cur res
1	2	8/22/2022 5:25	2	No	Sent	8/23/2022	NaN	Friend	Pakistan	K
2	3	8/22/2022 5:26	3	No	Sent	8/23/2022	NaN	WhatsApp group	Pakistan	L
3	4	8/22/2022 5:27	4	No	Sent	8/23/2022	NaN	Whatsapp group	Pakistan	Pes
4	5	8/22/2022 5:31	5	Yes	Sent	8/23/2022	NaN	Through a collegue	Pakistan	K

5 rows × 34 columns

DATA CLEANING

In [7]:
```
raw_data.columns
```

Out[7]:
```
Index(['Unnamed: 0', 'Timestamp', 'Student ID',
       'Do you understand that this is a paid course  ', 'Course Fee Email',
       'Date sent', 'Payment Receipt Sent',
       'How and where did you hear about this course ',
       'Which Country are you currently residing in ',
       'Which City are you currently residing in ', 'Gender', 'Age',
       'Are you currently attending University   College ',
       'Latest Degree Completed or in Progress ',
       'Name of University or College currently or previously attended ',
       'Discipline of Degree ',
       'Have you taken any foundational course in data science   econometrics   statisti
cs   computer science ',
       'What programming language s  are you most comfortable in  R  Python  Java  JS  e
tc    If none please type  None   ',
       'Prior Work Experience in Data Science or Machine Learning',
```

```
       'Primary Reason for taking this course ',
       'Do you understand that this is a paid course
',
       'Prior Formal Education  course teaching or using R  university assignments  codi
ng projects  course work  etc   in R ',
       'Prior Formal Education  course teaching or using Python  university assignments
coding projects  course work  etc   in Python ',
       'Prior Work Experience in R ', 'Prior Work Experience in Python ',
       'Personal Education or Knowledge  learning R on your own and or doing personal pr
ojects  etc   in R ',
       'Personal Education or Knowledge  learning Python on your own and or doing person
al projects  etc   in Python ',
       'Do you have a LinkedIn Account ',
       'Do you have any feedback  thoughts  or comments you would like to share ',
       'Date', 'Time', 'Qr_uni_col', 'Uni_col', 'Payment'],
      dtype='object')
```

In [8]:
```
raw_data.duplicated().sum()
```

Out[8]:
```
0
```

In [9]:
```
raw_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 247 entries, 0 to 246
Data columns (total 34 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Unnamed: 0
247 non-null    int64
 1   Timestamp
247 non-null    object
 2   Student ID
247 non-null    int64
 3   Do you understand that this is a paid course
233 non-null    object
 4   Course Fee Email
239 non-null    object
 5   Date sent
238 non-null    object
 6   Payment Receipt Sent
26 non-null     object
 7   How and where did you hear about this course
247 non-null    object
 8   Which Country are you currently residing in
246 non-null    object
 9   Which City are you currently residing in
246 non-null    object
 10  Gender
```

```
247 non-null    object
 11  Age
247 non-null    object
 12  Are you currently attending University   College
247 non-null    object
 13  Latest Degree Completed or in Progress
247 non-null    object
 14  Name of University or College currently or previously attended
247 non-null    object
 15  Discipline of Degree
247 non-null    object
 16  Have you taken any foundational course in data science   econometrics   statistics
computer science                            247 non-null    object
 17  What programming language s  are you most comfortable in  R  Python  Java  JS  etc
If none please type  None                  247 non-null    object
 18  Prior Work Experience in Data Science or Machine Learning
247 non-null    object
 19  Primary Reason for taking this course
247 non-null    object
 20  Do you understand that this is a paid course
247 non-null    object
 21  Prior Formal Education  course teaching or using R  university assignments  coding
projects  course work  etc   in R             247 non-null    object
 22  Prior Formal Education  course teaching or using Python  university assignments  co
ding projects  course work  etc   in Python   247 non-null    object
 23  Prior Work Experience in R
247 non-null    object
 24  Prior Work Experience in Python
247 non-null    object
 25  Personal Education or Knowledge  learning R on your own and or doing personal proje
cts  etc   in R                             247 non-null    object
 26  Personal Education or Knowledge  learning Python on your own and or doing personal
projects  etc   in Python                   247 non-null    object
 27  Do you have a LinkedIn Account
246 non-null    object
 28  Do you have any feedback  thoughts  or comments you would like to share
128 non-null    object
 29  Date
247 non-null    object
 30  Time
247 non-null    object
 31  Qr_uni_col
247 non-null    object
 32  Uni_col
247 non-null    object
 33  Payment
240 non-null    object
dtypes: int64(2), object(32)
memory usage: 65.7+ KB
```

In [10]:

```
raw_data.isnull().values.sum()
```

Out[10]: 381

RENAMING SOME COLUMNS

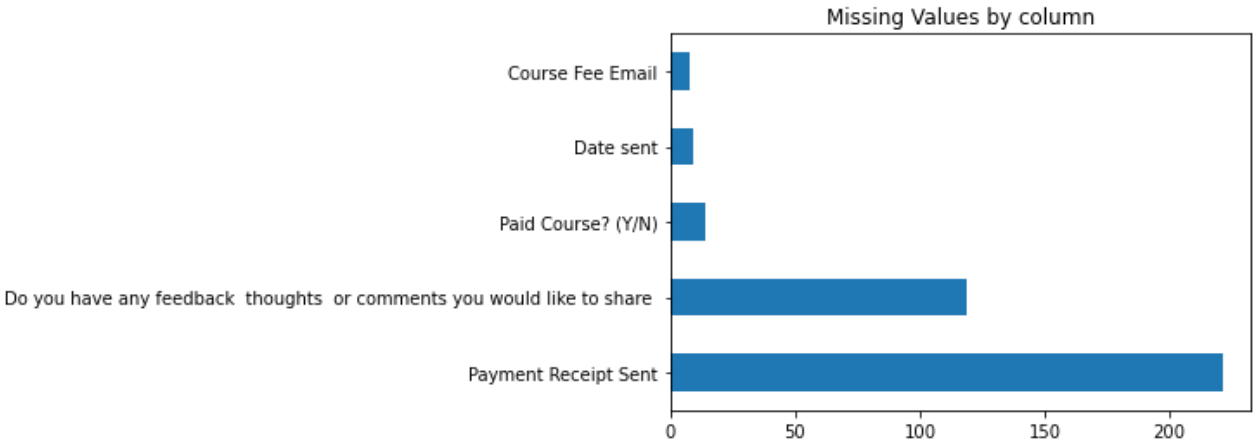
In [11]:
```
# create a dictionary
# key = old name
# value = new name
dictt = {'Do you understand that this is a paid course  ': "Paid Course? (Y/N)",
                'Which City are you currently residing in ': 'City',
        'Which Country are you currently residing in ': "Country",
        'Prior Formal Education  course teaching or using R  university assignments  coding projects  course work  etc   in R ': "Prior Formal Education in R",
        'Prior Formal Education  course teaching or using Python  university assignments  coding projects  course work  etc   in Python ': "Prior Formal Education in Python",
        'Personal Education or Knowledge  learning R on your own and or doing personal projects  etc   in R ': "Personal Education/Knowledge in R",
        'Personal Education or Knowledge  learning Python on your own and or doing personal projects  etc   in Python ': "Personal Education/Knowledge in Python",
        'Do you have a LinkedIn Account ': "LinkedIn?",
        'Name of University or College currently or previously attended ': "University Attended"
       }

# call rename () method
raw_data.rename(columns=dictt, inplace = True)
```

In [12]:
```
raw_data.columns
```

Out[12]:
```
Index(['Unnamed: 0', 'Timestamp', 'Student ID', 'Paid Course? (Y/N)',
       'Course Fee Email', 'Date sent', 'Payment Receipt Sent',
       'How and where did you hear about this course ', 'Country', 'City',
       'Gender', 'Age', 'Are you currently attending University   College ',
       'Latest Degree Completed or in Progress ', 'University Attended',
       'Discipline of Degree ',
       'Have you taken any foundational course in data science   econometrics   statistics   computer science ',
       'What programming language s  are you most comfortable in  R  Python  Java  JS  etc    If none please type  None   ',
       'Prior Work Experience in Data Science or Machine Learning',
```

```
       'Primary Reason for taking this course ',
       'Do you understand that this is a paid course
',
       'Prior Formal Education in R', 'Prior Formal Education in Python',
       'Prior Work Experience in R ', 'Prior Work Experience in Python ',
       'Personal Education/Knowledge in R',
       'Personal Education/Knowledge in Python', 'LinkedIn?',
       'Do you have any feedback  thoughts  or comments you would like to share ',
       'Date', 'Time', 'Qr_uni_col', 'Uni_col', 'Payment'],
      dtype='object')
```

In [13]:
```
# Visualizing Missing Data
x = raw_data.isnull().sum()
x.sort_values(ascending = False).head().plot(kind = "barh", stacked = True,
title = "Missing Values by column")
plt.show()
```



REMOVING INCONSISTENCY

In [14]:
```
import thefuzz
from thefuzz import fuzz, process
import chardet
```

```
C:\Users\pc\anaconda3\lib\site-packages\thefuzz\fuzz.py:11: UserWarning: Using slow pure
-python SequenceMatcher. Install python-Levenshtein to remove this warning
  warnings.warn('Using slow pure-python SequenceMatcher. Install python-Levenshtein to r
emove this warning')
```

In [15]:
```
inconsistent_data = irdatacleaning.InconsistentData(raw_data)
```

In [16]:
```
raw_data = inconsistent_data.data_white_space()
```

In [17]:
```
raw_data = inconsistent_data.changeing_column_cases()
```

In [18]:
```
raw_data.columns
```

Out[18]:
```
Index(['Unnamed: 0', 'Timestamp', 'Student Id', 'Paid Course? (Y/N)',
       'Course Fee Email', 'Date Sent', 'Payment Receipt Sent',
       'How And Where Did You Hear About This Course ', 'Country', 'City',
       'Gender', 'Age', 'Are You Currently Attending University   College ',
       'Latest Degree Completed Or In Progress ', 'University Attended',
       'Discipline Of Degree ',
       'Have You Taken Any Foundational Course In Data Science   Econometrics   Statisti
cs   Computer Science ',
       'What Programming Language S  Are You Most Comfortable In  R  Python  Java  Js  E
tc    If None Please Type  None   ',
       'Prior Work Experience In Data Science Or Machine Learning',
       'Primary Reason For Taking This Course ',
       'Do You Understand That This Is A Paid Course
',
       'Prior Formal Education In R', 'Prior Formal Education In Python',
       'Prior Work Experience In R ', 'Prior Work Experience In Python ',
       'Personal Education/Knowledge In R',
       'Personal Education/Knowledge In Python', 'Linkedin?',
       'Do You Have Any Feedback  Thoughts  Or Comments You Would Like To Share ',
       'Date', 'Time', 'Qr_Uni_Col', 'Uni_Col', 'Payment'],
      dtype='object')
```

IMPUTING MISSING DATA

In [19]:
```
raw_data["Paid Course? (Y/N)"] = raw_data["Paid Course?
(Y/N)"].replace(np.nan, "none" )
raw_data["Course Fee Email"] = raw_data["Course Fee Email"].replace(np.nan,
"Not Sent")
raw_data["Date Sent"]  = raw_data["Date Sent"].replace(np.nan, "-")
raw_data["Payment"]  = raw_data["Payment"].replace(np.nan, "Not paid")
raw_data["Payment Receipt Sent"]  = raw_data["Payment Receipt
Sent"].replace(np.nan, "not yet")
raw_data['Do You Have Any Feedback  Thoughts  Or Comments You Would Like To
Share '] = raw_data['Do You Have Any Feedback  Thoughts  Or Comments You
Would Like To Share '].replace(np.nan, "None")
raw_data['Linkedin?'] = raw_data['Linkedin?'].replace(np.nan, "not stated")
raw_data['Country']  = raw_data['Country'].replace(np.nan, "not stated")
raw_data['City']  = raw_data['City'].replace(np.nan, "not stated")
```

In [20]:
```
raw_data.isnull().values.sum() # checking any remaining missing data
```

Out[20]:
```
0
```

In [21]:
```
raw_data.columns
```

Out[21]:
```
Index(['Unnamed: 0', 'Timestamp', 'Student Id', 'Paid Course? (Y/N)',
       'Course Fee Email', 'Date Sent', 'Payment Receipt Sent',
       'How And Where Did You Hear About This Course ', 'Country', 'City',
       'Gender', 'Age', 'Are You Currently Attending University   College ',
       'Latest Degree Completed Or In Progress ', 'University Attended',
       'Discipline Of Degree ',
       'Have You Taken Any Foundational Course In Data Science   Econometrics   Statisti
cs   Computer Science ',
       'What Programming Language S  Are You Most Comfortable In  R  Python  Java  Js  E
tc    If None Please Type  None   ',
       'Prior Work Experience In Data Science Or Machine Learning',
       'Primary Reason For Taking This Course ',
       'Do You Understand That This Is A Paid Course
',
       'Prior Formal Education In R', 'Prior Formal Education In Python',
       'Prior Work Experience In R ', 'Prior Work Experience In Python ',
       'Personal Education/Knowledge In R',
       'Personal Education/Knowledge In Python', 'Linkedin?',
       'Do You Have Any Feedback  Thoughts  Or Comments You Would Like To Share ',
       'Date', 'Time', 'Qr_Uni_Col', 'Uni_Col', 'Payment'],
      dtype='object')
```

DATA RANGE CONSTRAINTS

In [22]:
```
raw_data.Age
```

Out[22]:
```
0            23
1            32
2            22
3            22
4            23
         ...
242          21
243    Under 17
244          21
245          21
246          21
Name: Age, Length: 247, dtype: object
```

In [23]:
```
# Since the age column has one category, we will replace it by the largest
number in that category i.e 16 to make it int dtype

raw_data.Age = raw_data.Age.replace("Under 17", 16)
raw_data[["Age"]] = raw_data[["Age"]].astype("int")
raw_data[["Age"]].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 247 entries, 0 to 246
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
```

```
 0   Age     247 non-null    int32
dtypes: int32(1)
memory usage: 1.1 KB
```

FIXING DATA TYPES

In [24]:
```
raw_data["Date Sent"]
```

Out[24]:
```
0      8/26/2022
1      8/23/2022
2      8/23/2022
3      8/23/2022
4      8/23/2022
         ...
242    8/24/2022
243            -
244            -
245            -
246            -
Name: Date Sent, Length: 247, dtype: object
```

In [25]:
```
#Ensuring different columns have the correct data type


raw_data.Timestamp = raw_data.Timestamp.astype("datetime64[ns]")
raw_data["Date Sent"] = raw_data["Date Sent"].astype("object")
raw_data.Date = raw_data.Date.astype("datetime64[ns]")
raw_data.Time = raw_data.Time.astype("datetime64[ns]")
raw_data.Age = raw_data.Age.astype("int64")
```

WORDCLOUD

In [26]:
```
from os import path
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

In [27]:
```
comments = raw_data.iloc[:,28]
```

In [28]:
```
restricted = ["None", "No"]
comments = comments[~comments.isin(restricted)]
```

In [29]:
```
text = comments.values
# Creating word_cloud with text as argument in .generate() method
wordcloud = WordCloud(collocations = False, background_color =
'black').generate(str(text))
# Display the generated Word Cloud
```

```
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

PREPROCESSING

In [30]:
```
raw_data[["Payment"]] = np.where(raw_data[["Payment"]] == "Paid",1,0)
```

In [31]:
```
raw_data.loc[(raw_data['Country'] == 'PAKISTAN')
             | (raw_data['Country'] == 'PK')
             | (raw_data['Country'] == 'Pakistan')
             | (raw_data['Country'] == 'pakistan')
             | (raw_data['Country'] == 'Karachi')
             | (raw_data['Country'] == 'Lahore'), 'Country'] = 'Pakistan'

raw_data.loc[raw_data["Country"] == "not stated", "Country"] = "Canada"
#from original dataset, university is Ryerson
```

In [32]:
```
raw_data.loc[(raw_data['City'] == 'KARACHI')
             | (raw_data['City'] == 'Karachi')
             | (raw_data['City'] == 'Pakistan')
             | (raw_data['City'] == 'karachi'), "City"] = "Karachi"

raw_data.loc[(raw_data['City'] == 'Lahore')
             | (raw_data['City'] == 'LAHORE'), "City"] = "Lahore"

raw_data.loc[(raw_data['City'] == 'Abbottabad')
             | (raw_data['City'] == 'Abbottabad Pakistan'), "City"] =
"Abbottabad"

raw_data.loc[(raw_data['City'] == 'ISLAMABAD')
             | (raw_data['City'] == 'Islamabad')
```

```
            | (raw_data["City"] == "islamabad"), "City"] = "Islamabad"

raw_data.loc[raw_data["City"] == "Batkhela, Malakand", "City"] = "Malakand"

raw_data.loc[raw_data["City"] == "not stated", "City"] = "Toronto" # Had to
check the university. Ryerson is in Toronto so assumed city is toronto too.
```

In [33]:
```
raw_data.to_csv("data_cleaned.csv")
```

In [34]:
```
cleaned_data = pd.read_csv("data_cleaned.csv")
```

In [35]:
```
cleaned_data['Have You Taken Any Foundational Course In Data Science
Econometrics   Statistics   Computer Science '].unique()
```

Out[35]:
```
array(['Data Science', 'None of them', 'All of them', 'Computer Science',
       'Econometrics', 'Statisitics'], dtype=object)
```

In [36]:
```
# Create a series out of the Country column
countries = raw_data['Country']

# Get the counts of each category
country_counts = countries.value_counts()

# Create a mask for only categories that occur less than 3 times
mask = countries.isin(country_counts[country_counts < 3].index)

# Label all other categories as Other
countries[mask] = 'Other'

# Print the updated category counts
print(pd.value_counts(countries))
```

```
Pakistan    234
Other         7
Canada        6
Name: Country, dtype: int64
C:\Users\pc\AppData\Local\Temp/ipykernel_11876/2336189131.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  countries[mask] = 'Other'
```

In [37]:
```
countries = raw_data['Country']

# Get the counts of each category
country_counts = countries.value_counts()

# Create a mask for only categories that occur less than 3 times
mask = countries.isin(country_counts[country_counts < 3].index)

# Label all other categories as Other
countries[mask] = 'Other'

# Print the updated category counts
print(pd.value_counts(countries))
```

```
Pakistan    234
Other         7
Canada        6
Name: Country, dtype: int64
C:\Users\pc\AppData\Local\Temp/ipykernel_11876/4183312234.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  countries[mask] = 'Other'
```

In [38]:
```
cities = raw_data['City']

# Get the counts of each category
cities_counts = cities.value_counts()

# Create a mask for only categories that occur less than 3 times
mask = cities.isin(cities_counts[cities_counts < 10].index)

# Label all other categories as Other
cities[mask] = 'Other'

# Print the updated category counts
print(pd.value_counts(cities))
```

```
Karachi      179
Other         36
Islamabad     18
```

```
Lahore        14
Name: City, dtype: int64
C:\Users\pc\AppData\Local\Temp/ipykernel_11876/2430343131.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  cities[mask] = 'Other'
```

ONE HOT ENCODING ON CERTAIN FEATURES WITHOUT USING PANDAS DUMMIES FUNCTION

In [39]:
```
def one_hot_top_x(df, variable, top_x_labels):



    for label in top_x_labels:
        df[variable+"_"+label] = np.where(cleaned_data[variable] == label, 1, 0)
```

In [40]:
```
cleaned_data = pd.read_csv("data_cleaned.csv", usecols = ["Paid Course? (Y/N)", "Country", "City", "Gender",
                                                          "Age", "Uni_Col",
'Have You Taken Any Foundational Course In Data Science   Econometrics   Statistics   Computer Science ', \
                                                         'How And Where Did You Hear About This Course '])
```

In [41]:
```
cleaned_data.rename(columns = {'Have You Taken Any Foundational Course In Data Science   Econometrics   Statistics   Computer Science ': "DS_Foundations"}, inplace = True)
```

In [42]:
```
#Reducing the categories. If someone has taken any one course is said to have some knowledge about Data Science
cleaned_data["DS_Foundations"] = np.where(cleaned_data["DS_Foundations"] != "None of them", "Have", "Do Not Have")
```

In [43]:
```
cleaned_data["DS_Foundations"]
```

Out[43]:
```
0            Have
1            Have
2      Do Not Have
3            Have
4            Have
          ...
```

```
242           Have
243           Have
244           Have
245           Have
246    Do Not Have
Name: DS_Foundations, Length: 247, dtype: object
```

In [44]:
```
#Since there are many categories in each variable of the dataset, I have
taken atmost 5 categories and hot_encoded them
foundations = [x for x in
cleaned_data["DS_Foundations"].value_counts().sort_values(ascending=False).he


foundations
```

Out[44]:
```
['Have', 'Do Not Have']
```

In [45]:
```
one_hot_top_x(cleaned_data, "DS_Foundations", foundations)
cleaned_data.head()
```

Out[45]:

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	City	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_H
0	Yes	Through a friend	Pakistan	Karachi	Male	23	Have	University of Illinois	
1	No	Friend	Pakistan	Karachi	Female	32	Have	NED	
2	No	WhatsApp group	Pakistan	Lahore	Male	22	Do Not Have	Univeristy of agriculture, Faislabad	
3	No	Whatsapp group	Pakistan	Peshawar	Male	22	Have	IOBM	
4	Yes	Through a collegue	Pakistan	Karachi	Male	23	Have	IBA	

In [46]:
```
top_referral = [x for x in cleaned_data['How And Where Did You Hear About
This Course '].value_counts().sort_values(ascending=False).head(4).index]
top_referral
```

Out[46]:
```
['Facebook', 'Friend', 'LinkedIn', 'Whatsapp']
```

In [47]:
```
one_hot_top_x(cleaned_data, 'How And Where Did You Hear About This Course '
```

```
, top_referral)
cleaned_data.head()
```

Out[47]:

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	City	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_H
0	Yes	Through a friend	Pakistan	Karachi	Male	23	Have	University of Illinois	
1	No	Friend	Pakistan	Karachi	Female	32	Have	NED	
2	No	WhatsApp group	Pakistan	Lahore	Male	22	Do Not Have	Univeristy of agriculture, Faislabad	
3	No	Whatsapp group	Pakistan	Peshawar	Male	22	Have	IOBM	
4	Yes	Through a collegue	Pakistan	Karachi	Male	23	Have	IBA	

In [48]:
```
cleaned_data.iloc[:,-1]
```

Out[48]:
```
0      0
1      0
2      0
3      0
4      0
      ..
242    0
243    0
244    0
245    0
246    0
Name: How And Where Did You Hear About This Course _Whatsapp, Length: 247, dtype: int32
```

In [49]:
```
cleaned_data
```

Out[49]:

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	City	Gender	Age	DS_Foundations	Uni_Col	DS_Foundat
0	Yes	Through a friend	Pakistan	Karachi	Male	23	Have	University of Illinois	
1	No	Friend	Pakistan	Karachi	Female	32	Have	NED	
2	No	WhatsApp group	Pakistan	Lahore	Male	22	Do Not Have	Univeristy of agriculture, Faislabad	
3	No	Whatsapp group	Pakistan	Peshawar	Male	22	Have	IOBM	
4	Yes	Through a collegue	Pakistan	Karachi	Male	23	Have	IBA	
...	...	...	...	...	...	...	...	...	
242	Yes	By a friend	Pakistan	Layyah	Female	21	Have	QU	
243	none	Faizan	Canada	Toronto	Male	16	Have	university of ottawa	
244	none	From a friend	Pakistan	Karachi	Male	21	Have	IBA	
245	none	One of my friend recommended me to do this cou...	Pakistan	Karachi	Male	21	Have	IBA	
246	none	Friend	Pakistan	Karachi	Male	21	Do Not Have	SZABIST	

247 rows × 14 columns

In [50]:
```
top_4 = [x for x in
cleaned_data.City.value_counts().sort_values(ascending=False).head(4).index]


top_4
```

Out[50]:
```
['Karachi', 'Islamabad', 'Lahore', 'Peshawar']
```

In [51]:
```
one_hot_top_x(cleaned_data, "City" , top_4)
cleaned_data.head()
```

Out[51]:

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	City	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_H
0	Yes	Through a friend	Pakistan	Karachi	Male	23	Have	University of Illinois	
1	No	Friend	Pakistan	Karachi	Female	32	Have	NED	
2	No	WhatsApp group	Pakistan	Lahore	Male	22	Do Not Have	Univeristy of agriculture, Faislabad	
3	No	Whatsapp group	Pakistan	Peshawar	Male	22	Have	IOBM	
4	Yes	Through a collegue	Pakistan	Karachi	Male	23	Have	IBA	

In [52]:
```
cleaned_data.drop("City", axis = 1)
```

Out[52]:

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_Have
0	Yes	Through a friend	Pakistan	Male	23	Have	University of Illinois	1
1	No	Friend	Pakistan	Female	32	Have	NED	1
2	No	WhatsApp group	Pakistan	Male	22	Do Not Have	Univeristy of agriculture, Faislabad	0
3	No	Whatsapp group	Pakistan	Male	22	Have	IOBM	1
4	Yes	Through a collegue	Pakistan	Male	23	Have	IBA	1
...	...	...	...	...	...	...	...	...
242	Yes	By a friend	Pakistan	Female	21	Have	QU	1

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_Have
243	none	Faizan	Canada	Male	16	Have	university of ottawa	1
244	none	From a friend	Pakistan	Male	21	Have	IBA	1
245	none	One of my friend recommended me to do this cou...	Pakistan	Male	21	Have	IBA	1
246	none	Friend	Pakistan	Male	21	Do Not Have	SZABIST	0

247 rows × 17 columns

In [53]:
```
top_3 = [x for x in
cleaned_data.Country.value_counts().sort_values(ascending=False).head().index


top_3
```

Out[53]:
```
['Pakistan', 'Canada', 'Saudi Arabia', 'United States', 'Bangladesh']
```

In [54]:
```
one_hot_top_x(cleaned_data, "Country" , top_3)
cleaned_data.head()
```

Out[54]:

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	City	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_H
0	Yes	Through a friend	Pakistan	Karachi	Male	23	Have	University of Illinois	
1	No	Friend	Pakistan	Karachi	Female	32	Have	NED	
2	No	WhatsApp group	Pakistan	Lahore	Male	22	Do Not Have	Univeristy of agriculture, Faislabad	

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	City	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_H
3	No	Whatsapp group	Pakistan	Peshawar	Male	22	Have	IOBM	
4	Yes	Through a collegue	Pakistan	Karachi	Male	23	Have	IBA	

5 rows × 23 columns

In [55]:
```
top_G = [x for x in
cleaned_data.Gender.value_counts().sort_values(ascending=False).head(2).index


top_G
```

Out[55]:
```
['Male', 'Female']
```

In [56]:
```
one_hot_top_x(cleaned_data, "Gender" , top_G)
cleaned_data.head()
```

Out[56]:

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	City	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_H
0	Yes	Through a friend	Pakistan	Karachi	Male	23	Have	University of Illinois	
1	No	Friend	Pakistan	Karachi	Female	32	Have	NED	
2	No	WhatsApp group	Pakistan	Lahore	Male	22	Do Not Have	Univeristy of agriculture, Faislabad	
3	No	Whatsapp group	Pakistan	Peshawar	Male	22	Have	IOBM	
4	Yes	Through a collegue	Pakistan	Karachi	Male	23	Have	IBA	

5 rows × 25 columns

In [57]:
```
top_uni = [x for x in
```

```
cleaned_data.Uni_Col.value_counts().sort_values(ascending=False).head().index


top_uni
```

Out[57]: ['IBA', 'IQRA', 'SYED', 'NED', 'QU']

In [58]:
```
one_hot_top_x(cleaned_data, "Uni_Col" , top_uni)
cleaned_data.head()
```

Out[58]:

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	City	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_H
0	Yes	Through a friend	Pakistan	Karachi	Male	23	Have	University of Illinois	
1	No	Friend	Pakistan	Karachi	Female	32	Have	NED	
2	No	WhatsApp group	Pakistan	Lahore	Male	22	Do Not Have	Univeristy of agriculture, Faislabad	
3	No	Whatsapp group	Pakistan	Peshawar	Male	22	Have	IOBM	
4	Yes	Through a collegue	Pakistan	Karachi	Male	23	Have	IBA	

5 rows × 30 columns

In [59]:
```
Is_this_paid = [x for x in cleaned_data["Paid Course? (Y/N)"].value_counts().sort_values(ascending=False).head(2).index]
Is_this_paid
```

Out[59]: ['No', 'Yes']

In [60]:
```
one_hot_top_x(cleaned_data, "Paid Course? (Y/N)" , Is_this_paid)
cleaned_data.head()
```

Out[60]:

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	City	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_H

	Paid Course? (Y/N)	How And Where Did You Hear About This Course	Country	City	Gender	Age	DS_Foundations	Uni_Col	DS_Foundations_H
0	Yes	Through a friend	Pakistan	Karachi	Male	23	Have	University of Illinois	
1	No	Friend	Pakistan	Karachi	Female	32	Have	NED	
2	No	WhatsApp group	Pakistan	Lahore	Male	22	Do Not Have	Univeristy of agriculture, Faislabad	
3	No	Whatsapp group	Pakistan	Peshawar	Male	22	Have	IOBM	
4	Yes	Through a collegue	Pakistan	Karachi	Male	23	Have	IBA	

5 rows × 32 columns

In [61]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [62]:
```
tv = TfidfVectorizer(max_features=5, stop_words="english")
```

In [63]:
```
tv.fit(raw_data['How And Where Did You Hear About This Course '])
train_tv_transformed = tv.transform(raw_data['How And Where Did You Hear
About This Course '])
```

In [64]:
```
train_tv_df = pd.DataFrame(train_tv_transformed.toarray(),
columns=tv.get_feature_names()).add_prefix("Referred_via_")
raw_data = pd.concat([raw_data, train_tv_df], axis = 1, sort = False)
```

In [65]:
```
examine_row = train_tv_df.iloc[0]
print(examine_row.sort_values(ascending = False))
```

```
Referred_via_friend      1.0
Referred_via_facebook    0.0
Referred_via_group       0.0
Referred_via_linkedin    0.0
Referred_via_whatsapp    0.0
Name: 0, dtype: float64
```

In [66]:
```
cleaned_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 247 entries, 0 to 246
Data columns (total 32 columns):
 #   Column                                                Non-Null Count  Dtype
---  ------                                                --------------  -----
 0   Paid Course? (Y/N)                                    247 non-null    object
 1   How And Where Did You Hear About This Course          247 non-null    object
 2   Country                                               247 non-null    object
 3   City                                                  247 non-null    object
 4   Gender                                                247 non-null    object
 5   Age                                                   247 non-null    int64
 6   DS_Foundations                                        247 non-null    object
 7   Uni_Col                                               247 non-null    object
 8   DS_Foundations_Have                                   247 non-null    int32
 9   DS_Foundations_Do Not Have                            247 non-null    int32
 10  How And Where Did You Hear About This Course _Facebook  247 non-null  int32
 11  How And Where Did You Hear About This Course _Friend    247 non-null  int32
 12  How And Where Did You Hear About This Course _LinkedIn  247 non-null  int32
 13  How And Where Did You Hear About This Course _Whatsapp  247 non-null  int32
 14  City_Karachi                                          247 non-null    int32
 15  City_Islamabad                                        247 non-null    int32
 16  City_Lahore                                           247 non-null    int32
 17  City_Peshawar                                         247 non-null    int32
 18  Country_Pakistan                                      247 non-null    int32
 19  Country_Canada                                        247 non-null    int32
 20  Country_Saudi Arabia                                  247 non-null    int32
 21  Country_United States                                 247 non-null    int32
 22  Country_Bangladesh                                    247 non-null    int32
 23  Gender_Male                                           247 non-null    int32
 24  Gender_Female                                         247 non-null    int32
 25  Uni_Col_IBA                                           247 non-null    int32
 26  Uni_Col_IQRA                                          247 non-null    int32
 27  Uni_Col_SYED                                          247 non-null    int32
 28  Uni_Col_NED                                           247 non-null    int32
 29  Uni_Col_QU                                            247 non-null    int32
 30  Paid Course? (Y/N)_No                                 247 non-null    int32
 31  Paid Course? (Y/N)_Yes                                247 non-null    int32
dtypes: int32(24), int64(1), object(7)
memory usage: 38.7+ KB
```

In [67]:
```
#After one_hot_encoding, the original features were dropped to prevent
multiollinearity and noise
cleaned_data.drop(cleaned_data.iloc[:,:7], axis = 1, inplace = True)
```

In [68]:
```
cleaned_data.drop("Uni_Col", axis = 1, inplace = True)
```

In [69]:
```
raw_data.columns
```

Out[69]:
```
Index(['Unnamed: 0', 'Timestamp', 'Student Id', 'Paid Course? (Y/N)',
```

```
       'Course Fee Email', 'Date Sent', 'Payment Receipt Sent',
       'How And Where Did You Hear About This Course ', 'Country', 'City',
       'Gender', 'Age', 'Are You Currently Attending University   College ',
       'Latest Degree Completed Or In Progress ', 'University Attended',
       'Discipline Of Degree ',
       'Have You Taken Any Foundational Course In Data Science   Econometrics   Statisti
cs   Computer Science ',
       'What Programming Language S  Are You Most Comfortable In  R  Python  Java  Js  E
tc    If None Please Type  None   ',
       'Prior Work Experience In Data Science Or Machine Learning',
       'Primary Reason For Taking This Course ',
       'Do You Understand That This Is A Paid Course
',
       'Prior Formal Education In R', 'Prior Formal Education In Python',
       'Prior Work Experience In R ', 'Prior Work Experience In Python ',
       'Personal Education/Knowledge In R',
       'Personal Education/Knowledge In Python', 'Linkedin?',
       'Do You Have Any Feedback  Thoughts  Or Comments You Would Like To Share ',
       'Date', 'Time', 'Qr_Uni_Col', 'Uni_Col', 'Payment',
       'Referred_via_facebook', 'Referred_via_friend', 'Referred_via_group',
       'Referred_via_linkedin', 'Referred_via_whatsapp'],
      dtype='object')
```

In [70]:
```
raw_data = raw_data[["Course Fee Email", "Age", "Are You Currently
Attending University   College ", 'Latest Degree Completed Or In Progress
', "Payment"]]
```

In [71]:
```
raw_data
```

Out[71]:

	Course Fee Email	Age	Are You Currently Attending University College	Latest Degree Completed Or In Progress	Payment
0	Sent	23	No	Bachelors	1
1	Sent	32	No	Masters	0
2	Sent	22	No	Masters	0
3	Sent	22	Yes	Bachelors	0
4	Sent	23	Yes	Bachelors	0
...	...	...	...	...	...
242	Sent	21	No	Bachelors	1
243	Not Sent	16	No	Bachelors	0
244	Not Sent	21	Yes	H.S.C	0
245	Not Sent	21	Yes	Bachelors	0
246	Not Sent	21	Yes	Bachelors	0

247 rows × 5 columns

In [72]:
```
prepared_data = pd.concat([cleaned_data, raw_data], axis = 1)
```

In [73]:
```
prepared_data
```

Out[73]:

	DS_Foundations_Have	DS_Foundations_Do Not Have	How And Where Did You Hear About This Course _Facebook	How And Where Did You Hear About This Course _Friend	How And Where Did You Hear About This Course _LinkedIn	How And Where Did You Hear About This Course _Whatsapp	City_Karachi
0	1	0	0	0	0	0	1
1	1	0	0	1	0	0	1
2	0	1	0	0	0	0	0
3	1	0	0	0	0	0	0
4	1	0	0	0	0	0	1
...	...	...	...	...	...	...	...
242	1	0	0	0	0	0	0
243	1	0	0	0	0	0	0
244	1	0	0	0	0	0	1
245	1	0	0	0	0	0	1
246	0	1	0	1	0	0	1

247 rows × 29 columns

In [74]:
```
prepared_data.rename(columns = {'Latest Degree Completed Or In Progress ': "Latest Degree"}, inplace = True)
```

In [75]:
```
def one_hot_top_2x(df, variable, top_x_labels):


    for label in top_x_labels:
        df[variable+"_"+label] = np.where(prepared_data[variable] == label, 1, 0) #THe dataframe had to change i.e prepared_data
```

In [76]:
```
top_degree = [x for x in prepared_data["Latest
Degree"].value_counts().sort_values(ascending=False).head().index]
top_degree
```

Out[76]:
```
['Bachelors', 'Masters', 'H.S.C', 'PhD']
```

In [77]:
```
prepared_data.columns
```

Out[77]:
```
Index(['DS_Foundations_Have', 'DS_Foundations_Do Not Have',
       'How And Where Did You Hear About This Course _Facebook',
       'How And Where Did You Hear About This Course _Friend',
       'How And Where Did You Hear About This Course _LinkedIn',
       'How And Where Did You Hear About This Course _Whatsapp',
       'City_Karachi', 'City_Islamabad', 'City_Lahore', 'City_Peshawar',
       'Country_Pakistan', 'Country_Canada', 'Country_Saudi Arabia',
       'Country_United States', 'Country_Bangladesh', 'Gender_Male',
       'Gender_Female', 'Uni_Col_IBA', 'Uni_Col_IQRA', 'Uni_Col_SYED',
       'Uni_Col_NED', 'Uni_Col_QU', 'Paid Course? (Y/N)_No',
       'Paid Course? (Y/N)_Yes', 'Course Fee Email', 'Age',
       'Are You Currently Attending University   College ', 'Latest Degree',
       'Payment'],
      dtype='object')
```

In [78]:
```
one_hot_top_2x(prepared_data, 'Latest Degree', top_degree)
prepared_data.head()
```

Out[78]:

	DS_Foundations_Have	DS_Foundations_Do Not Have	How And Where Did You Hear About This Course _Facebook	How And Where Did You Hear About This Course _Friend	How And Where Did You Hear About This Course _LinkedIn	How And Where Did You Hear About This Course _Whatsapp	City_Karachi
0	1	0	0	0	0	0	1
1	1	0	0	1	0	0	1
2	0	1	0	0	0	0	0
3	1	0	0	0	0	0	0
4	1	0	0	0	0	0	1

5 rows × 33 columns

In [79]:
```
fee_email = [x for x in prepared_data["Course Fee
Email"].value_counts().sort_values(ascending=False).head(2).index]
```

```
fee_email
```

Out[79]:
```
['Sent', 'Not Sent']
```

In [80]:
```
one_hot_top_2x(prepared_data,"Course Fee Email", fee_email)
prepared_data.head()
```

Out[80]:

	DS_Foundations_Have	DS_Foundations_Do Not Have	How And Where Did You Hear About This Course _Facebook	How And Where Did You Hear About This Course _Friend	How And Where Did You Hear About This Course _LinkedIn	How And Where Did You Hear About This Course _Whatsapp	City_Karachi
0	1	0	0	0	0	0	1
1	1	0	0	1	0	0	1
2	0	1	0	0	0	0	0
3	1	0	0	0	0	0	0
4	1	0	0	0	0	0	1

5 rows × 35 columns

In [81]:
```
prepared_data.columns
```

Out[81]:
```
Index(['DS_Foundations_Have', 'DS_Foundations_Do Not Have',
       'How And Where Did You Hear About This Course _Facebook',
       'How And Where Did You Hear About This Course _Friend',
       'How And Where Did You Hear About This Course _LinkedIn',
       'How And Where Did You Hear About This Course _Whatsapp',
       'City_Karachi', 'City_Islamabad', 'City_Lahore', 'City_Peshawar',
       'Country_Pakistan', 'Country_Canada', 'Country_Saudi Arabia',
       'Country_United States', 'Country_Bangladesh', 'Gender_Male',
       'Gender_Female', 'Uni_Col_IBA', 'Uni_Col_IQRA', 'Uni_Col_SYED',
       'Uni_Col_NED', 'Uni_Col_QU', 'Paid Course? (Y/N)_No',
       'Paid Course? (Y/N)_Yes', 'Course Fee Email', 'Age',
       'Are You Currently Attending University   College ', 'Latest Degree',
       'Payment', 'Latest Degree_Bachelors', 'Latest Degree_Masters',
       'Latest Degree_H.S.C', 'Latest Degree_PhD', 'Course Fee Email_Sent',
       'Course Fee Email_Not Sent'],
      dtype='object')
```

In [82]:
```
prepared_data.rename(columns = {'Are You Currently Attending University
College ': "Currently in Uni/Col"}, inplace = True)
```

In [83]:
```
currently_attending = [x for x in prepared_data["Currently in Uni/Col"].value_counts().sort_values(ascending=False).head(2).index]
currently_attending
```

Out[83]:
```
['Yes', 'No']
```

In [84]:
```
one_hot_top_2x(prepared_data,"Currently in Uni/Col", currently_attending)
prepared_data.head()
```

Out[84]:

	DS_Foundations_Have	DS_Foundations_Do Not Have	How And Where Did You Hear About This Course _Facebook	How And Where Did You Hear About This Course _Friend	How And Where Did You Hear About This Course _LinkedIn	How And Where Did You Hear About This Course _Whatsapp	City_Karachi
0	1	0	0	0	0	0	1
1	1	0	0	1	0	0	1
2	0	1	0	0	0	0	0
3	1	0	0	0	0	0	0
4	1	0	0	0	0	0	1

5 rows × 37 columns

In [85]:
```
prepared_data = prepared_data.select_dtypes(exclude=['object'])
```

In [86]:
```
prepared_data.drop("Payment", axis = 1)
```

Out[86]:

	DS_Foundations_Have	DS_Foundations_Do Not Have	How And Where Did You Hear About This Course _Facebook	How And Where Did You Hear About This Course _Friend	How And Where Did You Hear About This Course _LinkedIn	How And Where Did You Hear About This Course _Whatsapp	City_Karachi
0	1	0	0	0	0	0	1
1	1	0	0	1	0	0	1
2	0	1	0	0	0	0	0
3	1	0	0	0	0	0	0

	DS_Foundations_Have	DS_Foundations_Do Not Have	How And Where Did You Hear About This Course _Facebook	How And Where Did You Hear About This Course _Friend	How And Where Did You Hear About This Course _LinkedIn	How And Where Did You Hear About This Course _Whatsapp	City_Karachi
4	1	0	0	0	0	0	1
...	...	...	...	...	...	...	...
242	1	0	0	0	0	0	0
243	1	0	0	0	0	0	0
244	1	0	0	0	0	0	1
245	1	0	0	0	0	0	1
246	0	1	0	1	0	0	1

247 rows × 33 columns

In [87]:
```
X = prepared_data #independent features
y = raw_data["Payment"] #target feature
```

In [88]:
```
# split X and y into training and testing set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=56)
```

KNN

In [89]:
```
from sklearn import neighbors, datasets, preprocessing
from sklearn.metrics import accuracy_score
scaler = preprocessing.StandardScaler().fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
knn = neighbors.KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
accuracy_score(y_test, y_pred)
```

Out[89]: 0.9634146341463414

RANDOM FOREST

In [90]:
```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators = 1100, min_samples_leaf = 0.05,
random_state = 343)
scaler = preprocessing.StandardScaler().fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
accuracy_score(y_test, y_pred)
```

Out[90]:
```
0.9390243902439024
```

In [91]:
```
from sklearn.utils.multiclass import unique_labels
from sklearn.metrics import confusion_matrix, classification_report
```

In [92]:
```
def plot(y_true, y_pred):
    labels = unique_labels(y_test)
    column = [f" Predicted{label}" for label in labels]
    indices = [f" Actual{label}" for label in labels]
    table = pd.DataFrame(confusion_matrix(y_true, y_pred), columns=column,
index = indices)

    return table
```

In [93]:
```
plot(y_test, y_pred) #I think the values of confusion matrix are wrong
```

Out[93]:

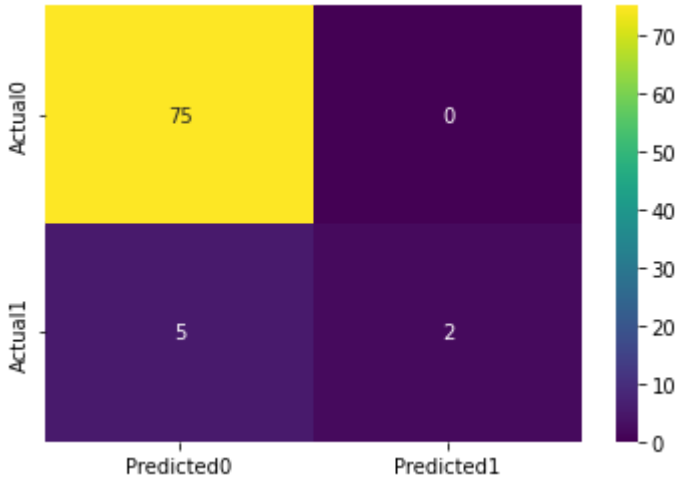
	Predicted0	Predicted1
Actual0	75	0
Actual1	5	2

In [94]:
```
def plot2(y_true, y_pred):
    labels = unique_labels(y_test)
    column = [f"Predicted{label}" for label in labels]
    indices = [f"Actual{label}" for label in labels]
    table = pd.DataFrame(confusion_matrix(y_true, y_pred), columns=column,
index = indices)
```

```
    return sns.heatmap(table, annot = True, fmt = "d", cmap = "viridis")
```

In [95]:
```
plot2(y_test, y_pred)
```

Out[95]:
```
<AxesSubplot:>
```



In [96]:
```
classification_report(y_test, y_pred)
```

Out[96]:
```
'              precision    recall  f1-score   support\n\n           0       0.94
1.00      0.97        75\n           1       1.00      0.29      0.44         7\n\n    a
ccuracy                           0.94        82\n   macro avg       0.97      0.64
0.71        82\nweighted avg       0.94      0.94      0.92        82\n'
```

In [97]:
```
prepared_data = pd.concat([prepared_data, raw_data["Payment"]], axis = 1)
```

In [98]:
```
prepared_data.to_csv("final_data.csv")
```