Intern Name : Faraj Momin

Intern ID : IP -4039

Domain : Machine Learning

Task 1: Placement Prediction Model

1. Importing the necessary Python Modules / Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
```

Reading the data **set**

```
train_data = pd.read_excel('01 Train Data.xlsx')
```

```
train_data.head()
```

	First Name	Email ID	Quantity	Price Tier	Ticket Type	Attendee #	Group	Order Type
0	ANIKET	aniket@xyz.com	1	NaN	Art of Resume Building	2.213855e+09	NaN	Free Order
1	Dhanshree	dhanshree@xyz.com	1	NaN	Art of Resume Building	2.213859e+09	NaN	Free Order
2	Dhiraj	dhiraj@xyz.com	1	NaN	Art of Resume Building	2.213862e+09	NaN	Free Order
3	Pooja	pooja@xyz.com	1	NaN	Art of Resume Building	2.213988e+09	NaN	Free Order
4	Aayush	aayush@xyz.com	1	NaN	Art of Resume Building	2.214567e+09	NaN	Free Order

5 rows × 23 columns

```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4894 entries, 0 to 4893
Data columns (total 23 columns):
 #   Column                                                  Non-Null Count  Dtype
---  ------                                                  --------------  -----
 0   First Name                                              4894 non-null   object
 1   Email ID                                                4894 non-null   object
 2   Quantity                                                4894 non-null   int64
 3   Price Tier                                              0 non-null      float64
 4   Ticket Type                                             4894 non-null   object
 5   Attendee #                                              4490 non-null   float64
 6   Group                                                   0 non-null      float64
 7   Order Type                                              4894 non-null   object
 8   Currency                                                4490 non-null   object
 9   Total Paid                                              4894 non-null   int64
```

```
 10  Fees Paid                                                          4490 non-null   float64
 11  Eventbrite Fees                                                    4894 non-null   int64
 12  Eventbrite Payment Processing                                      4894 non-null   int64
 13  Attendee Status                                                    4894 non-null   object
 14  College Name                                                       4879 non-null   object
 15  How did you come to know about this event?                         2678 non-null   object
 16  Specify in "Others" (how did you come to know about this event)    89 non-null     object
 17  Designation                                                        4894 non-null   object
 18  Year of Graduation                                                 1032 non-null   object
 19  CGPA                                                               4894 non-null   float64
 20  Speaking Skills                                                    4894 non-null   int64
 21  ML Knowledge                                                       4894 non-null   int64
 22  Placement Status                                                   1098 non-null   object
dtypes: float64(5), int64(6), object(12)
memory usage: 879.5+ KB
```

3. Dropping Blank **columns**

```
train_df = train_data.copy()
```

```
train_df.head()
```

	First Name	Email ID	Quantity	Price Tier	Ticket Type	Attendee #	Group	Order Type	Currency	Total Paid	...	Attendee Status	College Name
0	ANIKET	aniket@xyz.com	1	NaN	Art of Resume Building	2.213855e+09	NaN	Free Order	USD	0	...	Attending	D Y PATIL INSTITUTE OF MCA AND MANAGEMENT AKUR...
1	Dhanshree	dhanshree@xyz.com	1	NaN	Art of Resume Building	2.213859e+09	NaN	Free Order	USD	0	...	Attending	AP SHAH INSTITUTE OF TECHNOLOGY
2	Dhiraj	dhiraj@xyz.com	1	NaN	Art of Resume Building	2.213862e+09	NaN	Free Order	USD	0	...	Attending	Don Bosco College of Engineering Fatorda Goa
3	Pooja	pooja@xyz.com	1	NaN	Art of Resume Building	2.213988e+09	NaN	Free Order	USD	0	...	Attending	Pillai College of Engineering New Panvel
4	Aayush	aayush@xyz.com	1	NaN	Art of Resume Building	2.214567e+09	NaN	Free Order	USD	0	...	Attending	St Xavier's College

5 rows × 23 columns

```
columns = ['Price Tier', 'Group']
train_df = train_df.drop(columns, axis=1)
```

```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4894 entries, 0 to 4893
Data columns (total 21 columns):
 #   Column                                                             Non-Null Count  Dtype
---  ------                                                             --------------  -----
 0   First Name                                                         4894 non-null   object
 1   Email ID                                                           4894 non-null   object
 2   Quantity                                                           4894 non-null   int64
 3   Ticket Type                                                        4894 non-null   object
 4   Attendee #                                                         4490 non-null   float64
 5   Order Type                                                         4894 non-null   object
 6   Currency                                                           4490 non-null   object
 7   Total Paid                                                         4894 non-null   int64
 8   Fees Paid                                                          4490 non-null   float64
 9   Eventbrite Fees                                                    4894 non-null   int64
 10  Eventbrite Payment Processing                                      4894 non-null   int64
 11  Attendee Status                                                    4894 non-null   object
 12  College Name                                                       4879 non-null   object
 13  How did you come to know about this event?                         2678 non-null   object
 14  Specify in "Others" (how did you come to know about this event)    89 non-null     object
```

```
 15  Designation                                4894 non-null   object
 16  Year of Graduation                         1032 non-null   object
 17  CGPA                                       4894 non-null   float64
 18  Speaking Skills                            4894 non-null   int64
 19  ML Knowledge                               4894 non-null   int64
 20  Placement Status                           1098 non-null   object
dtypes: float64(3), int64(6), object(12)
memory usage: 803.0+ KB
```

4. Dropping Duplicate Values from dataset

```
train_df['Email ID'].value_counts()
```

```
Email ID
shubham@xyz.com       58
abhishek@xyz.com      42
patel@xyz.com         42
vaishnavi@xyz.com     41
omkar@xyz.com         36
                      ..
sowmyasree@xyz.com     1
sana@xyz.com           1
kanan@xyz.com          1
aishwary@xyz.com       1
utkarsha@xyz.com       1
Name: count, Length: 1987, dtype: int64
```

```
duplicates = train_df[train_df.duplicated(subset='Email ID', keep=False)]
duplicates
```

	First Name	Email ID	Quantity	Ticket Type	Attendee #	Order Type	Currency	Total Paid	Fees Paid	Eventbrite Fees	...	Attendee Status	Col
0	ANIKET	aniket@xyz.com	1	Art of Resume Building	2.213855e+09	Free Order	USD	0	0.0	0	...	Attending	INS MAN
1	Dhanshree	dhanshree@xyz.com	1	Art of Resume Building	2.213859e+09	Free Order	USD	0	0.0	0	...	Attending	INS TEC
2	Dhiraj	dhiraj@xyz.com	1	Art of Resume Building	2.213862e+09	Free Order	USD	0	0.0	0	...	Attending	F
3	Pooja	pooja@xyz.com	1	Art of Resume Building	2.213988e+09	Free Order	USD	0	0.0	0	...	Attending	Pilla
4	Aayush	aayush@xyz.com	1	Art of Resume Building	2.214567e+09	Free Order	USD	0	0.0	0	...	Attending	
...	...	...	...	...	...	...	...	...	...	...	...	...	
4889	Saif ali	saifali@xyz.com	1	Data Visualization using Power BI	1.949319e+09	Free Order	USD	0	0.0	0	...	Attending	mit
4890	Lankesh	lankesh@xyz.com	1	Data Visualization using Power BI	1.949328e+09	Free Order	USD	0	0.0	0	...	Attending	
4891	Sanjay	sanjay@xyz.com	1	Data Visualization using Power BI	1.949328e+09	Free Order	USD	0	0.0	0	...	Attending	wil
4892	Sushmita	sushmita@xyz.com	1	Data Visualization using Power BI	1.949330e+09	Free Order	USD	0	0.0	0	...	Attending	e a
4893	Vipul	vipul@xyz.com	1	Data Visualization using Power BI	1.949332e+09	Free Order	USD	0	0.0	0	...	Attending	dk

3585 rows × 21 columns

```
# Remove duplicates and keep only the first occurrence
cleaned_df = train_df.drop_duplicates(subset='Email ID', keep='first')

cleaned_df
```

	First Name	Email ID	Quantity	Ticket Type	Attendee #	Order Type	Curr
0	ANIKET	aniket@xyz.com	1	Art of Resume Building	2.213855e+09	Free Order	
1	Dhanshree	dhanshree@xyz.com	1	Art of Resume Building	2.213859e+09	Free Order	
2	Dhiraj	dhiraj@xyz.com	1	Art of Resume Building	2.213862e+09	Free Order	
3	Pooja	pooja@xyz.com	1	Art of Resume Building	2.213988e+09	Free Order	
4	Aayush	aayush@xyz.com	1	Art of Resume Building	2.214567e+09	Free Order	
...	...	...	...	...	...	...	
4829	Kevin	kevin@xyz.com	1	Transformation with DevOps: The Easy Way	2.231408e+09	Free Order	
4831	Hemil	hemil@xyz.com	1	Transformation with DevOps: The Easy Way	2.231440e+09	Free Order	
4832	Shitij	shitij@xyz.com	1	Transformation with DevOps: The Easy Way	2.231448e+09	Free Order	
4834	dirgh	dirgh@xyz.com	1	Transformation with DevOps: The Easy Way	2.231489e+09	Free Order	
4836	UTkarsha	utkarsha@xyz.com	1	Transformation with DevOps: The Easy Way	2.231554e+09	Free Order	

1987 rows × 21 columns

```
cleaned_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1987 entries, 0 to 4836
Data columns (total 21 columns):
 #   Column                                                                Non-Null Count  Dtype
---  ------                                                                --------------  -----
 0   First Name                                                            1987 non-null   object
 1   Email ID                                                              1987 non-null   object
 2   Quantity                                                              1987 non-null   int64
 3   Ticket Type                                                           1987 non-null   object
 4   Attendee #                                                            1765 non-null   float64
 5   Order Type                                                            1987 non-null   object
 6   Currency                                                              1765 non-null   object
 7   Total Paid                                                            1987 non-null   int64
 8   Fees Paid                                                             1765 non-null   float64
 9   Eventbrite Fees                                                       1987 non-null   int64
 10  Eventbrite Payment Processing                                         1987 non-null   int64
 11  Attendee Status                                                       1987 non-null   object
 12  College Name                                                          1977 non-null   object
 13  How did you come to know about this event?                            1222 non-null   object
 14  Specify in "Others" (how did you come to know about this event)       47 non-null     object
 15  Designation                                                           1987 non-null   object
 16  Year of Graduation                                                    280 non-null    object
 17  CGPA                                                                  1987 non-null   float64
```

```
 18  Speaking Skills                                    1987 non-null   int64
 19  ML Knowledge                                       1987 non-null   int64
 20  Placement Status                                   666 non-null    object
dtypes: float64(3), int64(6), object(12)
memory usage: 341.5+ KB
```

```
cleaned_df.describe()
```

	Quantity	Attendee #	Total Paid	Fees Paid	Eventbrite Fees	Eventbrite Payment Processing	CGPA	
count	1987.0	1.765000e+03	1987.0	1765.0	1987.0	1987.0	1987.000000	198
mean	1.0	2.496110e+09	0.0	0.0	0.0	0.0	8.027177	
std	0.0	7.963343e+08	0.0	0.0	0.0	0.0	1.004961	
min	1.0	1.937670e+09	0.0	0.0	0.0	0.0	6.200000	
25%	1.0	1.986366e+09	0.0	0.0	0.0	0.0	7.200000	
50%	1.0	2.220114e+09	0.0	0.0	0.0	0.0	7.900000	
75%	1.0	2.355403e+09	0.0	0.0	0.0	0.0	8.900000	

5. Dropping Unnecessary columns

```
cleaned_df.Quantity.value_counts()
```

```
Quantity
1    1987
Name: count, dtype: int64
```

```
cleaned_df['Ticket Type'].value_counts()
```

```
Ticket Type
Internship Program(IP) Success Conclave        339
Art of Resume Building                         337
Product Design & Full Stack                    258
Data Visualization using Power BI              222
Hello ML and DL                                115
Talk on Skill and Employability Enhancement    109
IS DATA SCIENCE FOR YOU?                       101
KYC - Know Your CCPC                            73
Skill and Employability Enhancement             71
IAC - Q&A                                       67
Artificial Intelligence                         62
Product Marketing                               62
The SDLC & their transformations                52
The Agile Ways of Working                       51
Transformation with DevOps: The Easy Way        44
RPA: A Boon or A Bane                           24
Name: count, dtype: int64
```

```
cleaned_df['Attendee #'].value_counts()
```

```
Attendee #
2.213855e+09    1
4.075904e+09    1
4.076058e+09    1
4.076054e+09    1
4.076055e+09    1
                ..
2.069050e+09    1
2.069025e+09    1
2.067939e+09    1
2.067783e+09    1
2.231554e+09    1
Name: count, Length: 1765, dtype: int64
```

```
cleaned_df['Order Type'].value_counts()
```

```
Order Type
Free Order    1986
Other            1
Name: count, dtype: int64
```

```
cleaned_df['Currency'].value_counts()
```

```
Currency
USD    1765
Name: count, dtype: int64
```

```
cleaned_df['Total Paid'].value_counts()
```

```
Total Paid
0    1987
Name: count, dtype: int64
```

```
cleaned_df['Fees Paid'].value_counts()
```

```
Fees Paid
0.0    1765
Name: count, dtype: int64
```

```
cleaned_df['Eventbrite Fees'].value_counts()
```

```
Eventbrite Fees
0    1987
Name: count, dtype: int64
```

```
cleaned_df['Eventbrite Payment Processing'].value_counts()
```

```
Eventbrite Payment Processing
0    1987
Name: count, dtype: int64
```

```
cleaned_df['Attendee Status'].value_counts()
```

```
Attendee Status
Attending    1982
NAN             5
Name: count, dtype: int64
```

drop some unecessary columns

```
columns = ['Quantity','Attendee #','Order Type','Currency','Total Paid','Fees Paid','Eventbrite Fees',
           'Eventbrite Payment Processing','Attendee Status']
cleaned_df = cleaned_df.drop(columns, axis=1)
```

```
cleaned_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1987 entries, 0 to 4836
Data columns (total 12 columns):
 #   Column                                                                Non-Null Count  Dtype
---  ------                                                                --------------  -----
 0   First Name                                                            1987 non-null   object
 1   Email ID                                                              1987 non-null   object
 2   Ticket Type                                                           1987 non-null   object
 3   College Name                                                          1977 non-null   object
 4   How did you come to know about this event?                            1222 non-null   object
 5   Specify in "Others" (how did you come to know about this event)  47 non-null     object
 6   Designation                                                           1987 non-null   object
 7   Year of Graduation                                                    280 non-null    object
 8   CGPA                                                                  1987 non-null   float64
 9   Speaking Skills                                                       1987 non-null   int64
 10  ML Knowledge                                                          1987 non-null   int64
 11  Placement Status                                                      666 non-null    object
dtypes: float64(1), int64(2), object(9)
memory usage: 201.8+ KB
```

```
cleaned_df['Ticket Type'].value_counts()
```

```
Ticket Type
Internship Program(IP) Success Conclave        339
Art of Resume Building                         337
Product Design & Full Stack                    258
Data Visualization using Power BI              222
Hello ML and DL                                115
Talk on Skill and Employability Enhancement    109
IS DATA SCIENCE FOR YOU?                       101
KYC - Know Your CCPC                            73
Skill and Employability Enhancement             71
IAC - Q&A                                       67
Artificial Intelligence                         62
Product Marketing                               62
The SDLC & their transformations                52
The Agile Ways of Working                       51
Transformation with DevOps: The Easy Way        44
RPA: A Boon or A Bane                           24
Name: count, dtype: int64
```

```
cleaned_df['College Name'].value_counts()
```

```
College Name
vidyalankar institute of technology, mumbai                                        94
priyadarshini college of engineering, nagpur                                       93
government polytechnic gandhinagar                                                91
ld college of engineering, ahmedabad, gujarat                                      87
wilson college                                                                     82
b. k. birla college of arts, science & commerce (autonomous), kalyan               78
g h raisoni institut of engineering and technology pune                            72
kle society's college of bca, rls institute, belagavi                              72
mit academy of engineering ,alandi                                                 71
dkte society's textile and engineering institute ichalkaranji                      64
vishwakarma institute of technology, pune                                          63
a. c. patil college of engineering                                                 62
new horizon institute of technology and management                                 58
symbiosis institute of technology, pune                                            53
don bosco college of engineering fatorda goa                                       51
st.francis institute of technology                                                 50
adhiyamaan college of engineering                                                  49
st xavier's college                                                                49
pillai college of engineering new panvel                                           48
silicon institute of technology                                                    47
late g. n. sapkal college of engineering                                           47
chhattisgarh swami vivekananda technical university teaching department bhilai     38
Priyadarshini college of engineering, Nagpur                                       32
A. C. Patil College of Engineering                                                 28
s.i.e.s. graduate school of technology, nerul, navi mumbai                         25
Vidyalankar Institute of Technology, Mumbai                                        25
lokmanya tilak college of engineering koparkhairane navi mumbai                    25
B. K. Birla College of Arts, Science & Commerce (Autonomous), Kalyan               25
thakur institute of management studies, career development & research - [timscdr]  24
Wilson college                                                                     23
Vishwakarma Institute of Technology, Pune                                          23
KLE Society's College of BCA, RLS Institute, Belagavi                              22
G H Raisoni institut of engineering and technology pune                            22
LD College of engineering, Ahmedabad, Gujarat                                      22
DKTE Society's Textile And Engineering Institute Ichalkaranji                      21
MIT Academy Of Engineering ,Alandi                                                 20
GOVERNMENT POLYTECHNIC GANDHINAGAR                                                 20
Symbiosis Institute of Technology, Pune                                            19
St.Francis Institute of Technology                                                 17
Late G. N. Sapkal College Of Engineering                                           17
Don Bosco College of Engineering Fatorda Goa                                       17
New horizon institute of Technology and Management                                 15
St Xavier's College                                                                15
Silicon Institute of Technology                                                    13
Pillai College of Engineering New Panvel                                           13
CHHATTISGARH SWAMI VIVEKANANDA TECHNICAL UNIVERSITY TEACHING DEPARTMENT BHILAI     13
ADHIYAMAAN COLLEGE OF ENGINEERING                                                  11
LOKMANYA TILAK COLLEGE OF ENGINEERING KOPARKHAIRANE NAVI MUMBAI                    11
ap shah institute of technology                                                     9
d y patil institute of mca and management akurdi pune                               8
S.I.E.S. Graduate School Of Technology, Nerul, Navi Mumbai                          8
THAKUR INSTITUTE OF MANAGEMENT STUDIES, CAREER DEVELOPMENT & RESEARCH - [TIMSCDR]   6
AP SHAH INSTITUTE OF TECHNOLOGY                                                     4
D Y PATIL INSTITUTE OF MCA AND MANAGEMENT AKURDI PUNE                               3
na                                                                                  2
Name: count, dtype: int64
```

```
cleaned_df['Designation'].value_counts()
```

```
Designation
Students                    1847
Intern                        28
Asst. Professor               17
Engineering                   14
Data Science and Analyst      13
HoD                            9
Computer Engineer              9
B.Tech                         4
CSE                            4
Artificial intelligence        3
IT                             3
Professor                      3
Web Development                3
BE IT                          3
Civil Engineering              3
Principal                      3
Mechanical Engineer            2
Software Engineer              2
BE CS                          2
IT Engineering                 1
MTech                          1
Research Scholar               1
Computer applications          1
Django                         1
Cyber Security                 1
```

```
MCA                          1
PGDM Marketing student       1
SPC                          1
Aerospace engineer           1
Electrical Engineering       1
BE                           1
Director T& P                1
Interior designer            1
Sr. Manager                  1
Name: count, dtype: int64
```

```
cleaned_df['Year of Graduation'].value_counts()
```

```
Year of Graduation
2021                        79
2022                        62
2023                        45
Second year                 32
Third year                  25
2020                        16
Final Year                   5
Fourth Year                  5
Bachelor of Engineering      2
Pursuing                     2
2019                         1
AISSMS                       1
2016                         1
B.Tech                       1
Friend                       1
1999                         1
2024                         1
Name: count, dtype: int64
```

```
cleaned_df['Source of Event Information'] = cleaned_df['How did you come to know about this event?'] + ' ' + cleaned_df['Specify in "Othe
```

```
cleaned_df.head()
```

	First Name	Email ID	Ticket Type	College Name	How did you come to know about this event?	Specify in "Others" (how did you come to know about this event)	Designa
0	ANIKET	aniket@xyz.com	Art of Resume Building	D Y PATIL INSTITUTE OF MCA AND MANAGEMENT AKUR...	Email	NaN	Stu
1	Dhanshree	dhanshree@xyz.com	Art of Resume Building	AP SHAH INSTITUTE OF TECHNOLOGY	Others	College	Stu
2	Dhiraj	dhiraj@xyz.com	Art of Resume Building	Don Bosco College of Engineering Fatorda Goa	Email	NaN	Stu
3	Pooja	pooja@xyz.com	Art of Resume	Pillai College of Engineering	Email	NaN	Stu

```
columns = ['How did you come to know about this event?','Specify in "Others" (how did you come to know about this event)']
cleaned_df = cleaned_df.drop(columns, axis=1)
```

```
cleaned_df.head()
```

	First Name	Email ID	Ticket Type	College Name	Designation	Year of Graduation	CGP
0	ANIKET	aniket@xyz.com	Art of Resume Building	D Y PATIL INSTITUTE OF MCA AND MANAGEMENT AKUR...	Students	NaN	6.
1	Dhanshree	dhanshree@xyz.com	Art of Resume Building	AP SHAH INSTITUTE OF TECHNOLOGY	Students	NaN	8.
2	Dhiraj	dhiraj@xyz.com	Art of Resume	Don Bosco College of	Students	NaN	6.

```
cleaned_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1987 entries, 0 to 4836
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   First Name                  1987 non-null   object 
 1   Email ID                    1987 non-null   object 
 2   Ticket Type                 1987 non-null   object 
 3   College Name                1977 non-null   object 
 4   Designation                 1987 non-null   object 
 5   Year of Graduation          280 non-null    object 
 6   CGPA                        1987 non-null   float64
 7   Speaking Skills             1987 non-null   int64  
 8   ML Knowledge                1987 non-null   int64  
 9   Placement Status            666 non-null    object 
 10  Source of Event Information  1222 non-null   object 
dtypes: float64(1), int64(2), object(8)
memory usage: 186.3+ KB
```

6. Visualizing the cleaned data for finding some useful analytics using the 'Plotly' library

```
cleaned_df['First Name'].value_counts()
```

```
First Name
ANIKET        1
Rahulsing     1
Daneshwari    1
Kausar        1
Sonica        1
             ..
Nishant       1
Arti          1
Barkha        1
Dhruvika      1
UTkarsha      1
Name: count, Length: 1987, dtype: int64
```

```
cleaned_df['Email ID'].value_counts()
```

```
Email ID
aniket@xyz.com        1
rahulsing@xyz.com     1
daneshwari@xyz.com    1
kausar@xyz.com        1
sonica@xyz.com        1
                     ..
nishant@xyz.com       1
arti@xyz.com          1
barkha@xyz.com        1
dhruvika@xyz.com      1
utkarsha@xyz.com      1
Name: count, Length: 1987, dtype: int64
```

```
cleaned_df['Ticket Type'].value_counts()
```

```
Ticket Type
Internship Program(IP) Success Conclave        339
Art of Resume Building                         337
Product Design & Full Stack                    258
Data Visualization using Power BI              222
Hello ML and DL                                115
Talk on Skill and Employability Enhancement    109
IS DATA SCIENCE FOR YOU?                       101
KYC - Know Your CCPC                            73
Skill and Employability Enhancement             71
```

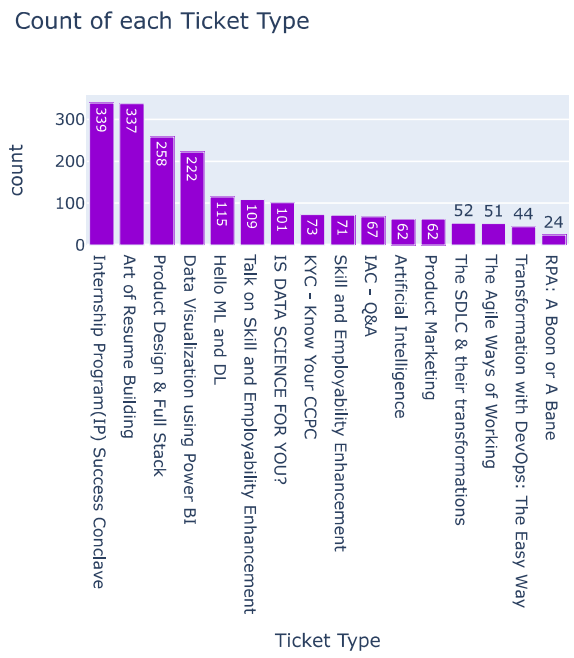
```
IAC - Q&A                                   67
Artificial Intelligence                     62
Product Marketing                           62
The SDLC & their transformations            52
The Agile Ways of Working                   51
Transformation with DevOps: The Easy Way    44
RPA: A Boon or A Bane                       24
Name: count, dtype: int64
```

```
ticket_type = cleaned_df['Ticket Type'].value_counts().keys()
count = cleaned_df['Ticket Type'].value_counts().values
labels = [str(val) for val in count]

fig = px.bar(x=ticket_type, y=count, text=labels, title='Count of each Ticket Type', template='plotly', color_discrete_sequence=['darkvio

# Set the labels for the y-axis and x-axis
fig.update_yaxes(title_text='count')
fig.update_xaxes(title_text='Ticket Type')

# Show the plot
fig.show()
```




```
cleaned_df['College Name'].value_counts()
```

```
College Name
vidyalankar institute of technology, mumbai                                94
priyadarshini college of engineering, nagpur                               93
government polytechnic gandhinagar                                        91
ld college of engineering, ahmedabad, gujarat                              87
wilson college                                                             82
b. k. birla college of arts, science & commerce (autonomous), kalyan       78
g h raisoni institut of engineering and technology pune                    72
kle society's college of bca, rls institute, belagavi                      72
mit academy of engineering ,alandi                                         71
dkte society's textile and engineering institute ichalkaranji              64
vishwakarma institute of technology, pune                                  63
a. c. patil college of engineering                                         62
new horizon institute of technology and management                         58
symbiosis institute of technology, pune                                    53
don bosco college of engineering fatorda goa                               51
st.francis institute of technology                                         50
adhiyamaan college of engineering                                          49
st xavier's college                                                        49
pillai college of engineering new panvel                                   48
silicon institute of technology                                            47
late g. n. sapkal college of engineering                                   47
chhattisgarh swami vivekananda technical university teaching department bhilai    38
Priyadarshini college of engineering, Nagpur                               32
A. C. Patil College of Engineering                                         28
s.i.e.s. graduate school of technology, nerul, navi mumbai                 25
Vidyalankar Institute of Technology, Mumbai                                25
lokmanya tilak college of engineering koparkhairane navi mumbai            25
B. K. Birla College of Arts, Science & Commerce (Autonomous), Kalyan       25
thakur institute of management studies, career development & research - [timscdr]    24
Wilson college                                                             23
Vishwakarma Institute of Technology, Pune                                  23
```

```
KLE Society's College of BCA, RLS Institute, Belagavi                              22
G H Raisoni institut of engineering and technology pune                            22
LD College of engineering, Ahmedabad, Gujarat                                     22
DKTE Society's Textile And Engineering Institute Ichalkaranji                      21
MIT Academy Of Engineering ,Alandi                                                 20
GOVERNMENT POLYTECHNIC GANDHINAGAR                                                 20
Symbiosis Institute of Technology, Pune                                            19
St.Francis Institute of Technology                                                 17
Late G. N. Sapkal College Of Engineering                                           17
Don Bosco College of Engineering Fatorda Goa                                       17
New horizon institute of Technology and Management                                 15
St Xavier's College                                                                15
Silicon Institute of Technology                                                    13
Pillai College of Engineering New Panvel                                           13
CHHATTISGARH SWAMI VIVEKANANDA TECHNICAL UNIVERSITY TEACHING DEPARTMENT BHILAI     13
ADHIYAMAAN COLLEGE OF ENGINEERING                                                  11
LOKMANYA TILAK COLLEGE OF ENGINEERING KOPARKHAIRANE NAVI MUMBAI                    11
ap shah institute of technology                                                     9
d y patil institute of mca and management akurdi pune                               8
S.I.E.S. Graduate School Of Technology, Nerul, Navi Mumbai                          8
THAKUR INSTITUTE OF MANAGEMENT STUDIES, CAREER DEVELOPMENT & RESEARCH - [TIMSCDR]    6
AP SHAH INSTITUTE OF TECHNOLOGY                                                     4
D Y PATIL INSTITUTE OF MCA AND MANAGEMENT AKURDI PUNE                               3
na                                                                                  2
Name: count, dtype: int64
```

```
college = cleaned_df['College Name'].value_counts().keys()
count = cleaned_df['College Name'].value_counts().values
labels = [str(val) for val in count]

fig = px.bar(y=college, x=count, text=labels, title='Count of each College', template='plotly', color_discrete_sequence=['yellow'])

# Set the labels for the y-axis and x-axis
fig.update_xaxes(title_text='count')
fig.update_yaxes(title_text='College')

fig.update_layout(width=1000, height=1300)

# Show the plot
fig.show()
```

Count of each College

College	Count
na	
D Y PATIL INSTITUTE OF MCA AND MANAGEMENT AKURDI PUNE	
AP SHAH INSTITUTE OF TECHNOLOGY	
THAKUR INSTITUTE OF MANAGEMENT STUDIES, CAREER DEVELOPMENT & RESEARCH - [TIMSCDR]	
S.I.E.S. Graduate School Of Technology, Nerul, Navi Mumbai	
d y patil institute of mca and management akurdi pune	
ap shah institute of technology	
LOKMANYA TILAK COLLEGE OF ENGINEERING KOPARKHAIRANE NAVI MUMBAI	
ADHIYAMAAN COLLEGE OF ENGINEERING	
CHHATTISGARH SWAMI VIVEKANANDA TECHNICAL UNIVERSITY TEACHING DEPARTMENT BHILAI	
Pillai College of Engineering New Panvel	
Silicon Institute of Technology	
St Xavier's College	
New horizon institute of Technology and Management	
Don Bosco College of Engineering Fatorda Goa	
Late G. N. Sapkal College Of Engineering	
St.Francis Institute of Technology	
Symbiosis Institute of Technology, Pune	
GOVERNMENT POLYTECHNIC GANDHINAGAR	
MIT Academy Of Engineering ,Alandi	
DKTE Society's Textile And Engineering Institute Ichalkaranji	
LD College of engineering, Ahmedabad, Gujarat	
G H Raisoni institut of engineering and technology pune	
KLE Society's College of BCA, RLS Institute, Belagavi	
Vishwakarma Institute of Technology, Pune	
Wilson college	
thakur institute of management studies, career development & research - [timscdr]	
B. K. Birla College of Arts, Science & Commerce (Autonomous), Kalyan	
lokmanya tilak college of engineering koparkhairane navi mumbai	
Vidyalankar Institute of Technology, Mumbai	
s.i.e.s. graduate school of technology, nerul, navi mumbai	
A. C. Patil College of Engineering	
Priyadarshini college of engineering, Nagpur	
chhattisgarh swami vivekananda technical university teaching department bhilai	
late g. n. sapkal college of engineering	
silicon institute of technology	
pillai college of engineering new panvel	
st xavier's college	
adhiyamaan college of engineering	
st.francis institute of technology	
don bosco college of engineering fatorda goa	
symbiosis institute of technology, pune	
new horizon institute of technology and management	
a. c. patil college of engineering	
vishwakarma institute of technology, pune	
dkte society's textile and engineering institute ichalkaranji	
mit academy of engineering ,alandi	
kle society's college of bca, rls institute, belagavi	
g h raisoni institut of engineering and technology pune	
b. k. birla college of arts, science & commerce (autonomous), kalyan	
wilson college	
ld college of engineering, ahmedabad, gujarat	
government polytechnic gandhinagar	
priyadarshini college of engineering, nagpur	
vidyalankar institute of technology, mumbai	
0	

```
cleaned_df['Designation'].value_counts()
```

```
Designation
Students                    1847
Intern                        28
Asst. Professor               17
Engineering                   14
Data Science and Analyst      13
HoD                            9
Computer Engineer              9
B.Tech                         4
CSE                            4
Artificial intelligence        3
```

```
IT                          3
Professor                   3
Web Development             3
BE IT                       3
Civil Engineering           3
Principal                   3
Mechanical Engineer         2
Software Engineer           2
BE CS                       2
IT Engineering              1
MTech                       1
Research Scholar            1
Computer applications       1
Django                      1
Cyber Security              1
MCA                         1
PGDM Marketing student      1
SPC                         1
Aerospace engineer          1
Electrical Engineering      1
BE                          1
Director T& P               1
Interior designer           1
Sr. Manager                 1
Name: count, dtype: int64
```

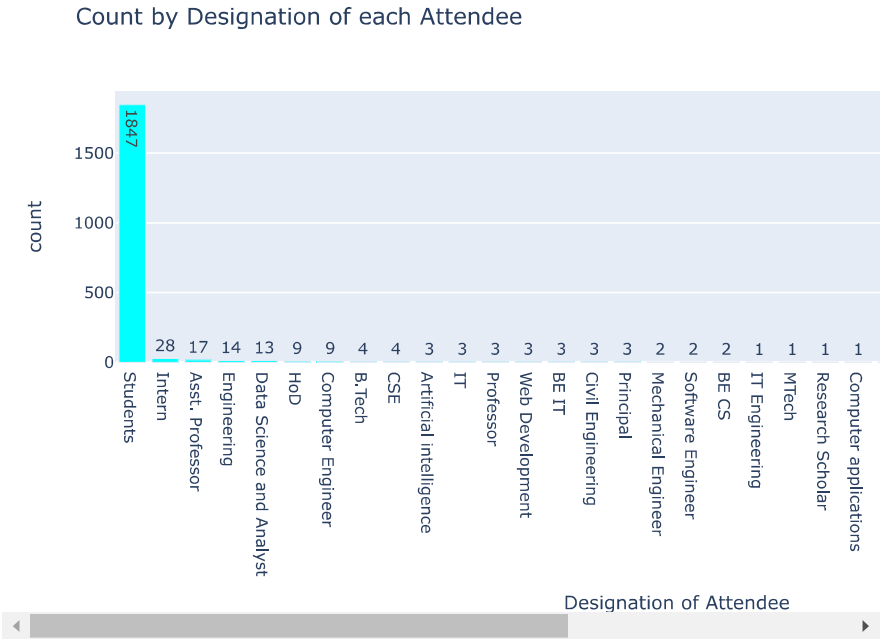
```
designation = cleaned_df['Designation'].value_counts().keys()
count = cleaned_df['Designation'].value_counts().values
labels = [str(val) for val in count]

fig = px.bar(x=designation, y=count, text=labels, title='Count by Designation of each Attendee', template='plotly', color_discrete_sequ

# Set the labels for the y-axis and x-axis
fig.update_yaxes(title_text='count')
fig.update_xaxes(title_text='Designation of Attendee')

fig.update_layout(width=1000, height=500)

# Show the plot
fig.show()
```




```
cleaned_df['Year of Graduation'].value_counts()
```

```
Year of Graduation
2021                      79
2022                      62
2023                      45
Second year               32
Third year                25
2020                      16
Final Year                 5
Fourth Year                5
Bachelor of Engineering    2
Pursuing                   2
2019                       1
AISSMS                     1
2016                       1
```

```
B.Tech                      1
Friend                      1
1999                        1
2024                        1
Name: count, dtype: int64
```

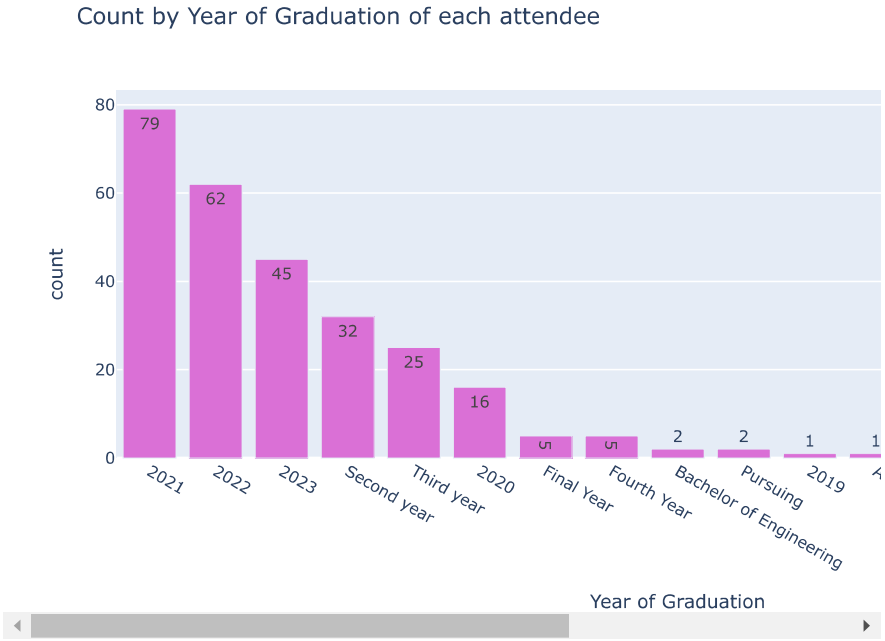
```
yog = cleaned_df['Year of Graduation'].value_counts().keys()
count = cleaned_df['Year of Graduation'].value_counts().values
labels = [str(val) for val in count]

fig = px.bar(x=yog, y=count, text=labels, title='Count by Year of Graduation of each attendee', template='plotly', color_discrete_sequer

# Set the labels for the y-axis and x-axis
fig.update_yaxes(title_text='count')
fig.update_xaxes(title_text='Year of Graduation')

fig.update_layout(width=1000, height=500)

# Show the plot
fig.show()
```




```
cleaned_df['Source of Event Information'].value_counts().head(25)
```

```
Source of Event Information
Whatsapp                                                481
Email                                                   229
SPOC/ College Professor                                 127
Cloud Counselage Website                                 58
Others                                                   41
Others College                                           34
LinkedIn                                                 24
Whatsapp | SPOC/ College Professor                       21
Facebook                                                 20
Instagram                                                19
Youtube                                                  18
Whatsapp | Email                                         15
Friend/ Classmate                                        15
Email | Cloud Counselage Website                         12
Youtube | Whatsapp                                        8
Friend/ Classmate | SPOC/ College Professor               6
Youtube | Whatsapp | Email                               6
Whatsapp | Cloud Counselage Website                      6
Email | SPOC/ College Professor                           5
Telegram                                                 4
Others Friends                                           4
Youtube | Facebook | Instagram | Whatsapp | Email       3
Instagram | Whatsapp                                     2
Whatsapp | Friend/ Classmate                            2
Instagram | LinkedIn | Email | Cloud Counselage Website  2
Name: count, dtype: int64
```

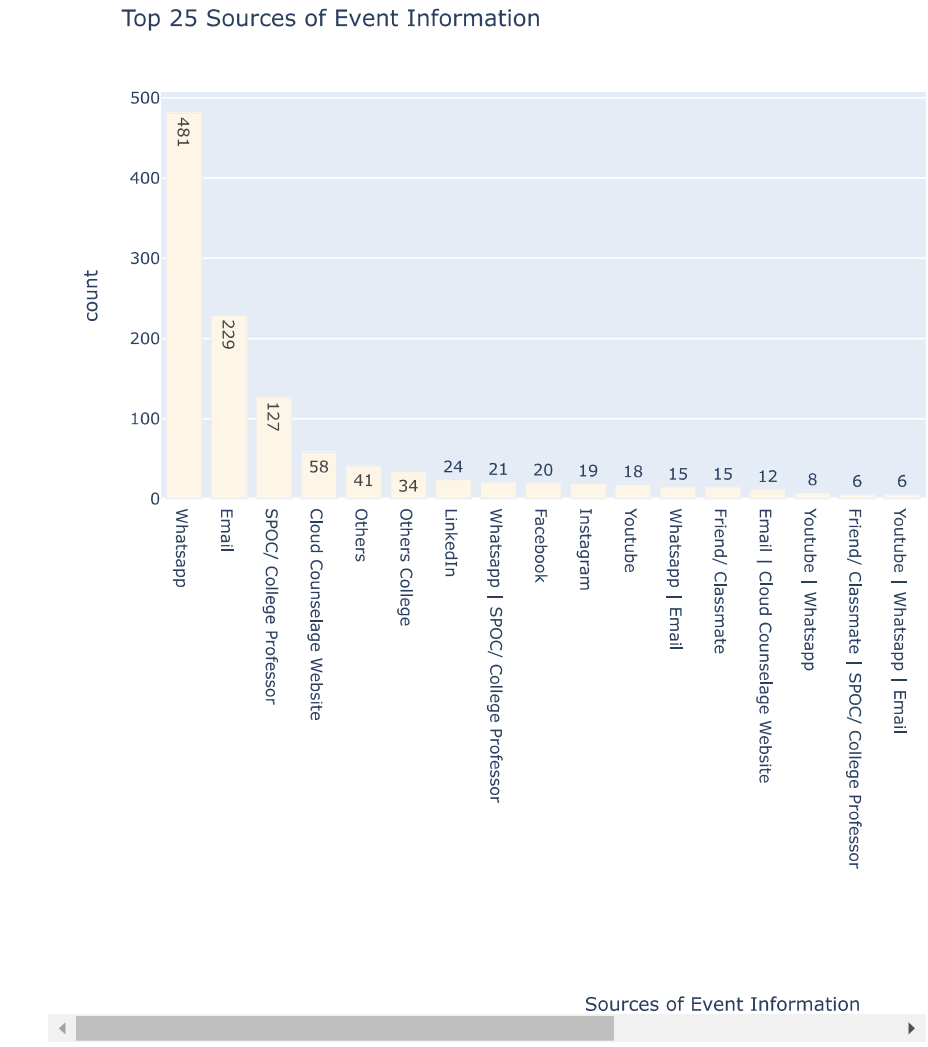
```
source = cleaned_df['Source of Event Information'].value_counts().head(25).keys()
count = cleaned_df['Source of Event Information'].value_counts().head(25).values
labels = [str(val) for val in count]

fig = px.bar(x=source, y=count, text=labels, title='Top 25 Sources of Event Information', template='plotly', color_discrete_sequence=['

# Set the labels for the y-axis and x-axis
fig.update_yaxes(title_text='count')
fig.update_xaxes(title_text='Sources of Event Information')

fig.update_layout(width=1000, height=800)

# Show the plot
fig.show()
```




```
cleaned_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1987 entries, 0 to 4836
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   First Name                   1987 non-null   object
 1   Email ID                     1987 non-null   object
 2   Ticket Type                  1987 non-null   object
 3   College Name                 1977 non-null   object
 4   Designation                  1987 non-null   object
 5   Year of Graduation           280 non-null    object
 6   CGPA                         1987 non-null   float64
 7   Speaking Skills              1987 non-null   int64
 8   ML Knowledge                 1987 non-null   int64
 9   Placement Status             666 non-null    object
 10  Source of Event Information  1222 non-null   object
dtypes: float64(1), int64(2), object(8)
memory usage: 186.3+ KB
```

7. Cleaning and preparing the final training and testing data

```
columns = ['College Name','CGPA','Speaking Skills','ML Knowledge','Placement Status']
train_data = cleaned_df[columns]
train_data['Placement Status'].value_counts()
```

```
Placement Status
Not placed    454
Placed        212
Name: count, dtype: int64
```

Label Encode 'Placed' as 1 and 'Not Placed' as 0 and 'Blanks' as 2

```
from sklearn.preprocessing import LabelEncoder
# Initialize LabelEncoder
label_encoder = LabelEncoder()
train_data['Placement Status'] = label_encoder.fit_transform(train_data['Placement Status'])
train_data['College Name'] = label_encoder.fit_transform(train_data['College Name'])
```

```
<ipython-input-49-0205de7169f7>:4: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus

<ipython-input-49-0205de7169f7>:5: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
```

```
train_data['Placement Status'] = train_data['Placement Status'].fillna(0)
```

```
<ipython-input-50-acbf9708c742>:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
```

```
train_data.head()
```

	College Name	CGPA	Speaking Skills	ML Knowledge	Placement Status
0	5	6.7	2	5	1
1	2	8.2	3	2	0
2	7	6.5	4	3	0
3	16	8.7	2	5	0
4	20	9.1	3	5	1

```
train_data.corr()
```

	College Name	CGPA	Speaking Skills	ML Knowledge	Placement Status
College Name	1.000000	0.015798	-0.023175	-0.022636	0.389209
CGPA	0.015798	1.000000	0.019610	0.010322	0.016035
Speaking Skills	-0.023175	0.019610	1.000000	0.884473	-0.052706
ML Knowledge	-0.022636	0.010322	0.884473	1.000000	-0.052940
Placement	0.389209	0.016035	-0.052706	-0.052940	1.000000

```
plt.figure(figsize=(10, 8))
sns.heatmap(train_data.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap with Placement Status')
plt.show()
```


```
train_data.to_excel('Final Cleaned Training Data.xlsx')
train_data.to_csv('Final Cleaned Training Data.csv')
```

we have our final cleaned training data ready Now, let's prepare our final cleaned testing data We will be following the given process for creating our final cleaned testing data

```
test_df = pd.read_excel('02 Test Data.xlsx')
test_df.head()
```

	First Name	Email ID	Quantity	Price Tier	Ticket Type	Attendee #	Group	Order Type
0	Sahil	sahil@xyz.com	1	NaN	Hello ML and DL	2.293940e+09	NaN	Free Order
1	Amrita	amrita@xyz.com	1	NaN	Hello ML and DL	2.293941e+09	NaN	Free Order
2	Mamta	mamta@xyz.com	1	NaN	Hello ML and DL	2.293941e+09	NaN	Free Order
3	Bhagyashri	bhagyashri@xyz.com	1	NaN	Hello ML and DL	2.293946e+09	NaN	Free Order
4	Divyanshu	divyanshu@xyz.com	1	NaN	Hello ML and DL	2.293956e+09	NaN	Free Order

5 rows × 23 columns

```
test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3796 entries, 0 to 3795
Data columns (total 23 columns):
 #   Column                                                                Non-Null Count  Dtype  
---  ------                                                                --------------  -----  
 0   First Name                                                            3796 non-null   object 
 1   Email ID                                                              3796 non-null   object 
 2   Quantity                                                              3796 non-null   int64  
 3   Price Tier                                                            0 non-null      float64
 4   Ticket Type                                                           3796 non-null   object 
 5   Attendee #                                                            3794 non-null   float64
 6   Group                                                                 0 non-null      float64
 7   Order Type                                                            3796 non-null   object 
 8   Currency                                                              3794 non-null   object 
 9   Total Paid                                                            3796 non-null   int64  
 10  Fees Paid                                                             3794 non-null   float64
 11  Eventbrite Fees                                                       3796 non-null   int64  
 12  Eventbrite Payment Processing                                         3796 non-null   int64  
 13  Attendee Status                                                       3796 non-null   object 
 14  College Name                                                          3795 non-null   object 
 15  How did you come to know about this event?                            1584 non-null   object 
 16  Specify in "Others" (how did you come to know about this event)       27 non-null     object 
 17  Designation                                                           3796 non-null   object 
 18  Year of Graduation                                                    1032 non-null   object 
 19  CGPA                                                                  3796 non-null   float64
 20  Speaking Skills                                                       3796 non-null   int64  
 21  ML Knowledge                                                          3796 non-null   int64  
 22  Placement Status                                                      0 non-null      float64
dtypes: float64(6), int64(6), object(11)
memory usage: 682.2+ KB
```

remove duplicates and redundant data(unnecessary columns)

```
duplicates = test_df[test_df.duplicated(subset='Email ID', keep=False)]
test_cleaned_df = test_df.drop_duplicates(subset='Email ID', keep='first')
test_cleaned_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2321 entries, 0 to 3695
Data columns (total 23 columns):
 #   Column                                                                Non-Null Count  Dtype  
---  ------                                                                --------------  -----  
 0   First Name                                                            2321 non-null   object 
 1   Email ID                                                              2321 non-null   object 
 2   Quantity                                                              2321 non-null   int64  
 3   Price Tier                                                            0 non-null      float64
 4   Ticket Type                                                           2321 non-null   object 
 5   Attendee #                                                            2321 non-null   float64
 6   Group                                                                 0 non-null      float64
 7   Order Type                                                            2321 non-null   object 
 8   Currency                                                              2321 non-null   object 
 9   Total Paid                                                            2321 non-null   int64  
 10  Fees Paid                                                             2321 non-null   float64
 11  Eventbrite Fees                                                       2321 non-null   int64  
 12  Eventbrite Payment Processing                                         2321 non-null   int64  
 13  Attendee Status                                                       2321 non-null   object 
 14  College Name                                                          2320 non-null   object 
 15  How did you come to know about this event?                            815 non-null    object 
 16  Specify in "Others" (how did you come to know about this event)       14 non-null     object 
 17  Designation                                                           2321 non-null   object 
 18  Year of Graduation                                                    489 non-null    object 
 19  CGPA                                                                  2321 non-null   float64
 20  Speaking Skills                                                       2321 non-null   int64  
 21  ML Knowledge                                                          2321 non-null   int64  
 22  Placement Status                                                      0 non-null      float64
dtypes: float64(6), int64(6), object(11)
memory usage: 435.2+ KB
```

```
columns = ['Quantity','Price Tier','Group','Attendee #','Order Type','Currency','Total Paid','Fees Paid','Eventbrite Fees',
           'Eventbrite Payment Processing','Attendee Status']
test_cleaned_df = test_cleaned_df.drop(columns, axis=1)
test_cleaned_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2321 entries, 0 to 3695
Data columns (total 12 columns):
 #   Column                                                                Non-Null Count  Dtype  
---  ------                                                                --------------  -----  
 0   First Name                                                            2321 non-null   object 
 1   Email ID                                                              2321 non-null   object 
 2   Ticket Type                                                           2321 non-null   object 
 3   College Name                                                          2320 non-null   object 
 4   How did you come to know about this event?                            815 non-null    object 
 5   Specify in "Others" (how did you come to know about this event)       14 non-null     object 
```

```
 6   Designation                2321 non-null   object
 7   Year of Graduation         489 non-null    object
 8   CGPA                       2321 non-null   float64
 9   Speaking Skills            2321 non-null   int64
 10  ML Knowledge               2321 non-null   int64
 11  Placement Status           0 non-null      float64
dtypes: float64(2), int64(2), object(8)
memory usage: 235.7+ KB
```

```
test_cleaned_df['Source of Event Information'] = test_cleaned_df['How did you come to know about this event?'] + ' ' + test_cleaned_df[
columns = ['How did you come to know about this event?','Specify in "Others" (how did you come to know about this event)']
test_cleaned_df = test_cleaned_df.drop(columns, axis=1)
test_cleaned_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2321 entries, 0 to 3695
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   First Name                  2321 non-null   object
 1   Email ID                    2321 non-null   object
 2   Ticket Type                 2321 non-null   object
 3   College Name                2320 non-null   object
 4   Designation                 2321 non-null   object
 5   Year of Graduation          489 non-null    object
 6   CGPA                        2321 non-null   float64
 7   Speaking Skills             2321 non-null   int64
 8   ML Knowledge                2321 non-null   int64
 9   Placement Status            0 non-null      float64
 10  Source of Event Information  815 non-null    object
dtypes: float64(2), int64(2), object(7)
memory usage: 217.6+ KB
```

```
columns = ['College Name','CGPA','Speaking Skills','ML Knowledge','Placement Status']
test_data = test_cleaned_df[columns]
```

Label Encode College Name as well

```
test_data['College Name'] = label_encoder.fit_transform(test_data['College Name'])
```

```
<ipython-input-61-475f80763ead>:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
```

```
test_data.head()
```

	College Name	CGPA	Speaking Skills	ML Knowledge	Placement Status
0	49	7.8	3	3	NaN
1	41	9.1	3	3	NaN
2	27	6.9	2	2	NaN
3	53	8.4	4	4	NaN
4	39	6.7	5	5	NaN

8. Preparing training and validation datasets using the provided training data

```
columns = ['College Name','CGPA','Speaking Skills', 'ML Knowledge']

x_train = train_data[columns]
y_train = train_data['Placement Status']
```

9. Training and testing using different Classification and Regression Models

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier, BaggingClassifier, ExtraTreesClass
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from xgboost import XGBClassifier
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import accuracy_score, classification_report


# Split the train data into training and validation sets
X_train_split, X_val_split, y_train_split, y_val_split = train_test_split(x_train, y_train, test_size=0.2, random_state=0)
```

A. Training and testing using RandomForestClassifier

```
model_1 = RandomForestClassifier(n_estimators=100)
model_1.fit(X_train_split, y_train_split)
```

```
RandomForestClassifier()
```

```
# Predict on the validation set
y_val_pred_1 = model_1.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_1))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_1))
```

```
Validation Accuracy: 0.6457286432160804
Classification Report:
              precision    recall  f1-score   support

           0       0.44      0.40      0.42        95
           1       0.18      0.13      0.15        47
           2       0.76      0.83      0.80       256

    accuracy                           0.65       398
   macro avg       0.46      0.45      0.46       398
weighted avg       0.62      0.65      0.63       398
```

B. Training and testing using LogisticRegression

```
model_2 = LogisticRegression()
model_2.fit(X_train_split, y_train_split)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning:

lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
```

```
LogisticRegression()
```

```
y_val_pred_2 = model_2.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_2))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_2))
```

```
Validation Accuracy: 0.6984924623115578
Classification Report:
              precision    recall  f1-score   support

           0       0.57      0.38      0.46        95
           1       0.00      0.00      0.00        47
           2       0.72      0.95      0.82       256

    accuracy                           0.70       398
   macro avg       0.43      0.44      0.42       398
weighted avg       0.60      0.70      0.64       398
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cor

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cor

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cor
```

C. Training and Testing using KNeighborsClassifier

```
model_3 = KNeighborsClassifier()
model_3.fit(X_train_split, y_train_split)
```

```
KNeighborsClassifier()
```

```
y_val_pred_3 = model_3.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_3))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_3))
```

```
Validation Accuracy: 0.6708542713567839
Classification Report:
              precision    recall  f1-score   support

           0       0.46      0.44      0.45        95
           1       0.10      0.04      0.06        47
           2       0.78      0.87      0.82       256

    accuracy                           0.67       398
   macro avg       0.45      0.45      0.44       398
weighted avg       0.62      0.67      0.64       398
```

D. Training and Testing using Support Vector Machine (SVM)

```
model_4 = SVC()
model_4.fit(X_train_split, y_train_split)
```

```
SVC()
```

```
y_val_pred_4 = model_4.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_4))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_4))
```

```
Validation Accuracy: 0.7060301507537688
Classification Report:
              precision    recall  f1-score   support

           0       0.55      0.42      0.48        95
           1       0.00      0.00      0.00        47
           2       0.74      0.94      0.83       256

    accuracy                           0.71       398
   macro avg       0.43      0.45      0.44       398
weighted avg       0.61      0.71      0.65       398

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cor

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cor

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cor
```

E. Training and Testing using DecisionTreeClassifier

```
model_5 = DecisionTreeClassifier()
model_5.fit(X_train_split, y_train_split)
```

```
DecisionTreeClassifier()
```

```
y_val_pred_5 = model_5.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_5))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_5))
```

```
Validation Accuracy: 0.5678391959798995
Classification Report:
                precision    recall  f1-score   support

           0       0.37      0.43      0.40        95
           1       0.10      0.11      0.10        47
           2       0.76      0.70      0.73       256

    accuracy                           0.57       398
   macro avg       0.41      0.41      0.41       398
weighted avg       0.59      0.57      0.58       398
```

F. Training and Testing using Neural Networks (MLP Classifier)

```
model_6 =  MLPClassifier(hidden_layer_sizes=(100,), max_iter=300, random_state=0)
model_6.fit(X_train_split, y_train_split)
```

```
MLPClassifier(max_iter=300, random_state=0)
```

```
y_val_pred_6 = model_6.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_6))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_6))
```

```
Validation Accuracy: 0.7085427135678392
Classification Report:
                precision    recall  f1-score   support

           0       0.58      0.41      0.48        95
           1       0.00      0.00      0.00        47
           2       0.73      0.95      0.83       256

    accuracy                           0.71       398
   macro avg       0.44      0.45      0.44       398
weighted avg       0.61      0.71      0.65       398

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cor

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cor

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to cor
```

G. Training and Testing using XGBoost Classifier

```
model_7 =  XGBClassifier(use_label_encoder=False, eval_metric='mlogloss', random_state=0)
model_7.fit(X_train_split, y_train_split)
```

```
/usr/local/lib/python3.10/dist-packages/xgboost/core.py:158: UserWarning:

[10:03:44] WARNING: /workspace/src/learner.cc:740:
Parameters: { "use_label_encoder" } are not used.
```

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='mlogloss',
              feature_types=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None,
              n_jobs=None, num_parallel_tree=None, objective='multi:softprob', ...)
```

```
y_val_pred_7 = model_7.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_7))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_7))
```

```
Validation Accuracy: 0.6758793969849246
Classification Report:
              precision    recall  f1-score   support

           0       0.51      0.39      0.44        95
           1       0.14      0.09      0.11        47
           2       0.77      0.89      0.82       256

    accuracy                           0.68       398
   macro avg       0.47      0.46      0.46       398
weighted avg       0.63      0.68      0.65       398
```

H. Training and Testing using GradientBoostingClassifier

```
model_8 = GradientBoostingClassifier(n_estimators=100, random_state=0)
model_8.fit(X_train_split, y_train_split)
```

```
GradientBoostingClassifier(random_state=0)
```

```
y_val_pred_8 = model_8.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_8))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_8))
```

```
Validation Accuracy: 0.6909547738693468
Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.37      0.43        95
           1       0.05      0.02      0.03        47
           2       0.77      0.93      0.84       256

    accuracy                           0.69       398
   macro avg       0.45      0.44      0.44       398
weighted avg       0.63      0.69      0.65       398
```

I. Training and Testing using AdaBoostClassifier

```
model_9 = AdaBoostClassifier(n_estimators=100, random_state=0)
model_9.fit(X_train_split, y_train_split)
```

```
AdaBoostClassifier(n_estimators=100, random_state=0)
```

```
y_val_pred_9 = model_9.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_9))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_9))
```

```
Validation Accuracy: 0.7412060301507538
Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.60      0.60        95
```

```
              1       0.00      0.00      0.00        47
              2       0.80      0.93      0.86       256

       accuracy                           0.74       398
      macro avg       0.47      0.51      0.49       398
   weighted avg       0.66      0.74      0.70       398
```

J. Training and Testing using Bagging Classifier

```
model_10 = BaggingClassifier(n_estimators=100, random_state=0)
model_10.fit(X_train_split, y_train_split)
```

```
BaggingClassifier(n_estimators=100, random_state=0)
```

```
y_val_pred_10 = model_10.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_10))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_10))
```

```
Validation Accuracy: 0.628140703517588
Classification Report:
              precision    recall  f1-score   support

           0       0.41      0.40      0.41        95
           1       0.07      0.04      0.05        47
           2       0.76      0.82      0.79       256

    accuracy                           0.63       398
   macro avg       0.41      0.42      0.42       398
weighted avg       0.59      0.63      0.61       398
```

K. Training and Testing using ExtraTrees Classifier

```
model_11 = ExtraTreesClassifier(n_estimators=100, random_state=0)
model_11.fit(X_train_split, y_train_split)
```

```
ExtraTreesClassifier(random_state=0)
```

```
y_val_pred_11 = model_11.predict(X_val_split)
print("Validation Accuracy:", accuracy_score(y_val_split, y_val_pred_11))
print("Classification Report:\n", classification_report(y_val_split, y_val_pred_11))
```

```
Validation Accuracy: 0.6055276381909548
Classification Report:
              precision    recall  f1-score   support

           0       0.40      0.40      0.40        95
           1       0.06      0.04      0.05        47
           2       0.75      0.79      0.77       256

    accuracy                           0.61       398
   macro avg       0.40      0.41      0.41       398
weighted avg       0.58      0.61      0.59       398
```

L. Training and Testing using Linear Regression Model

```
model_12 = LinearRegression()
model_12.fit(X_train_split, y_train_split)
```

```
LinearRegression()
```

```
y_val_pred_12 = model_12.predict(X_val_split)
```

```
print("Validation MSE:", mean_squared_error(y_val_split, y_val_pred_12))
print("Validation R^2 Score: ", r2_score(y_val_split, y_val_pred_12))
```

```
Validation MSE: 0.5850991746157537
Validation R^2 Score:  0.1854061044338151
```

M. Training and Testing using Ridge Regression Model

```
model_13 = Ridge()
model_13.fit(X_train_split, y_train_split)
```

```
Ridge()
```

```
y_val_pred_13 = model_13.predict(X_val_split)
print("Validation MSE:", mean_squared_error(y_val_split, y_val_pred_13))
print("Validation R^2 Score: ", r2_score(y_val_split, y_val_pred_13))
```

```
Validation MSE: 0.5850873135632789
Validation R^2 Score:  0.18542261777268165
```

N. Training and Testing using Lasso Regression Model

```
model_15 = DecisionTreeRegressor(random_state=0)
model_15.fit(X_train_split, y_train_split)
```

```
DecisionTreeRegressor(random_state=0)
```

```
y_val_pred_15 = model_15.predict(X_val_split)
print("Validation MSE:", mean_squared_error(y_val_split, y_val_pred_15))
print("Validation R^2 Score: ", r2_score(y_val_split, y_val_pred_15))
```

```
Validation MSE: 1.1364810161920715
Validation R^2 Score:  -0.5822454352715296
```

After Training and testing using various Classification and Regression Models we have the following: - For Classification Models: -

```
print("\nValidation Accuracy Score for", model_1 ," is " ,accuracy_score(y_val_split, y_val_pred_1))
print("\nValidation Accuracy Score for", model_3 ," is " ,accuracy_score(y_val_split, y_val_pred_3))
print("\nValidation Accuracy Score for", model_4 ," is " ,accuracy_score(y_val_split, y_val_pred_4))
print("\nValidation Accuracy Score for", model_5 ," is " ,accuracy_score(y_val_split, y_val_pred_5))
print("\nValidation Accuracy Score for", model_6 ," is " ,accuracy_score(y_val_split, y_val_pred_6))
print("\nValidation Accuracy Score for", model_7," is ",accuracy_score(y_val_split, y_val_pred_7))
print("\nValidation Accuracy Score for", model_8 ," is ",accuracy_score(y_val_split, y_val_pred_8))
print("\nValidation Accuracy Score for", model_9 ," is ",accuracy_score(y_val_split, y_val_pred_9))
print("\nValidation Accuracy Score for", model_10 ," is ",accuracy_score(y_val_split, y_val_pred_10))
print("\nValidation Accuracy Score for", model_11 ," is ",accuracy_score(y_val_split, y_val_pred_11))
```

```
Validation Accuracy Score for RandomForestClassifier()  is  0.6457286432160804

Validation Accuracy Score for KNeighborsClassifier()  is  0.6708542713567839

Validation Accuracy Score for SVC()  is  0.7060301507537688

Validation Accuracy Score for DecisionTreeClassifier()  is  0.5678391959798995

Validation Accuracy Score for MLPClassifier(max_iter=300, random_state=0)  is  0.7085427135678392

Validation Accuracy Score for XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='mlogloss',
              feature_types=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None,
              n_jobs=None, num_parallel_tree=None, objective='multi:softprob', ...)  is  0.6758793969849246

Validation Accuracy Score for GradientBoostingClassifier(random_state=0)  is  0.6909547738693468

Validation Accuracy Score for AdaBoostClassifier(n_estimators=100, random_state=0)  is  0.7412060301507538

Validation Accuracy Score for BaggingClassifier(n_estimators=100, random_state=0)  is  0.628140703517588

Validation Accuracy Score for ExtraTreesClassifier(random_state=0)  is  0.6055276381909548
```

For Regression Models

```
print("\nValidation MSE for", model_2, " is ", mean_squared_error(y_val_split, y_val_pred_2))
print("R^2 for", model_2, " is ", r2_score(y_val_split, y_val_pred_2))

print("\nValidation MSE for", model_12, " is ", mean_squared_error(y_val_split, y_val_pred_12))
print("R^2 for", model_12, " is ", r2_score(y_val_split, y_val_pred_12))

print("\nValidation MSE for", model_13, " is ", mean_squared_error(y_val_split, y_val_pred_13))
print("R^2 for", model_13, " is ", r2_score(y_val_split, y_val_pred_13))

print("\nValidation MSE for", model_14, " is ", mean_squared_error(y_val_split, y_val_pred_14))
print("R^2 for", model_14, " is ", r2_score(y_val_split, y_val_pred_14))

print("\nValidation MSE for", model_15, " is ", mean_squared_error(y_val_split, y_val_pred_15))
print("R^2 for", model_15, " is ", r2_score(y_val_split, y_val_pred_15))
```

```
Validation MSE for LogisticRegression()  is  0.8517587939698492
R^2 for LogisticRegression()  is  -0.18584599699411974

Validation MSE for LinearRegression()  is  0.5850991746157537
R^2 for LinearRegression()  is  0.1854061044338151

Validation MSE for Ridge()  is  0.5850873135632789
R^2 for Ridge()  is  0.18542261777268165
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-96-cbbb4381a95e> in <cell line: 10>()
      8 print("R^2 for", model_13, " is ", r2_score(y_val_split, y_val_pred_13))
      9
---> 10 print("\nValidation MSE for", model_14, " is ", mean_squared_error(y_val_split, y_val_pred_14))
     11 print("R^2 for", model_14, " is ", r2_score(y_val_split, y_val_pred_14))
     12

NameError: name 'model_14' is not defined
```

From above we can conclude that Classifcation Models out-performed Regression Models

In Classification Models we have AdaBoost Classifier with an Validation Accuracy of 74.1% i.e. 0.74

10. Now, let's use the trained Model to predict the test data provided

```
test_data.head(10)
```

	College Name	CGPA	Speaking Skills	ML Knowledge	Placement Status
0	49	7.8	3	3	NaN
1	41	9.1	3	3	NaN
2	27	6.9	2	2	NaN
3	53	8.4	4	4	NaN
4	39	6.7	5	5	NaN
5	33	9.2	2	2	NaN
6	50	7.3	2	2	NaN
7	40	7.9	2	2	NaN
8	44	9.9	5	5	NaN
9	51	7.6	2	2	NaN

```
test_pred_para = test_data.drop(columns="Placement Status")
test_pred_placement = model_9.predict(test_pred_para)
test_data['Placement Status Predicted'] = test_pred_placement
```

```
<ipython-input-98-37c7e6aa2093>:3: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
```

```
test_data.head()
```

	College Name	CGPA	Speaking Skills	ML Knowledge	Placement Status	Placement Status Predicted
0	49	7.8	3	3	NaN	2
1	41	9.1	3	3	NaN	2
2	27	6.9	2	2	NaN	2
3	53	8.4	4	4	NaN	2
4	39	6.7	5	5	NaN	2

```
test_data['Placement Status Predicted'].value_counts()
```

```
Placement Status Predicted
2    2077
0     230
1      14
Name: count, dtype: int64
```

```
test_data.head()
```

	College Name	CGPA	Speaking Skills	ML Knowledge	Placement Status	Placement Status Predicted
0	49	7.8	3	3	NaN	2
1	41	9.1	3	3	NaN	2
2	27	6.9	2	2	NaN	2
3	53	8.4	4	4	NaN	2
4	39	6.7	5	5	NaN	2

Now let's add the Predicted placement status to the Cleaned test data

```
test_cleaned_df['Predicted Placement Status'] = test_pred_placement
test_cleaned_df.head(25)
```

	First Name	Email ID	Ticket Type	College Name	Designation	Year of Graduation	CGPA	Speaking Skills	ML Knowledge	Placement Status	Source of Event Information	Pr Pl
0	Sahil	sahil@xyz.com	Hello ML and DL	symbiosis institute of technology, pune	Students	NaN	7.8	3	3	NaN	Whatsapp	
1	Amrita	amrita@xyz.com	Hello ML and DL	mit academy of engineering ,alandi	Students	NaN	9.1	3	3	NaN	Whatsapp	
2	Mamta	mamta@xyz.com	Hello ML and DL	a. c. patil college of engineering	Students	NaN	6.9	2	2	NaN	Whatsapp	
3	Bhagyashri	bhagyashri@xyz.com	Hello ML and DL	wilson college	Students	NaN	8.4	4	4	NaN	Others	
4	Divyanshu	divyanshu@xyz.com	Hello ML and DL	ld college of engineering, ahmedabad, gujarat	Students	NaN	6.7	5	5	NaN	Whatsapp	
5	Aditya	aditya@xyz.com	Hello ML and	dkte society's textile and	Students	NaN	9.2	2	2	NaN	Email	

Let's Decode the Predicted Placement Status column:- 0 -- Not Placed 1 -- Placed 2 -- Blanks

```
pred_placement_status_decode_mapping = {
    0: 'Not Placed',
    1: 'Placed',
    2: ' '
}

test_cleaned_df['Predicted Placement Status'] = test_cleaned_df['Predicted Placement Status'].replace(pred_placement_status_decode_mapp
```