

Project Name: E-Learning vs. Traditional Coaching: An Exploratory Data Analysis and Predictive Modeling Approach

Institution Name: Vigor Council

Guidance Under: Dr. B.P. Sharma

Intern Name: Kriti Khurana & Kritika Mittal

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_excel('Survey Form Responses.xlsx')
```

```
df.head()
```

	Timestamp	Name \
0	2024-05-23 10:52:09.341	Kriti khurana
1	2024-05-23 11:00:12.108	Sweety
2	2024-05-23 11:00:41.276	Naveen
3	2024-05-23 11:01:48.760	Rishub Srivastava
4	2024-05-23 11:04:30.412	Aditya

	Age in years (enter numeric values only)	Gender \
0	24	Female
1	23	Female
2	23	Male
3	24	Male
4	25	Male

	You belong to which district of Delhi?	Current Profession \
0	North-West	Student
1	New Delhi	Working Professional
2	North-West	Student
3	North-West	Working Professional
4	New Delhi	Working Professional

	Highest Level of Education	Background of Education \
0	Graduate	Commerce
1	Post Graduate	Commerce
2	Post Graduate	Commerce
3	Graduate	Science
4	Post Graduate	Science

When do you prefer to do a job \ What is your family income per annum

0	After Graduation	Above 15LPA
1	After Graduation	9LPA-11LPA
2	After Post Graduation	9LPA-11LPA
3	After Graduation	Above 15LPA
4	After Post Graduation	12LPA-15LPA

Is the online course value for money rather than offline course \

0	yes
1	yes
2	no
3	yes
4	yes

Considering the course fees and value obtained, which platform do you believe offers the best return on investment \

0	online
1	online
2	online
3	offline
4	offline

What specific aspects of online platform made you think it is the best platform \

0	Any time access
1	All of the above
2	No travelling Expense
3	I don't like online platforms
4	All of the above

What specific aspects of offline platform made you think it is the best platform \

0	Personal Touch
1	Classroom study with peers

2 Personal Touch

3 All of the above

4 All of the above

On a scale of 1 to 5, how would you rate the diversity and depth of course offerings across the online platform \

0 3

1 4

2 5

3 3

4 4

On a scale of 1 to 5, how do the instructors on online platforms compared to offline platforms in terms of expertise and teaching effectiveness \

0 3

1 4

2 5

3 5

4 3

Which platform according to you, offers the most interactive and engaging teaching methods \

0 offline

1 offline

2 offline

3 offline

4 offline

On a scale of 1 to 5, how user-friendly and intuitive are the online learning platforms \

0 4

1	4
2	4
3	2
4	3

Which platform according to you, offers the best balance of structured learning materials, hands-on projects, and interactive elements \

0	offline
1	offline
2	offline
3	offline
4	offline

Which platform according to you, provide superior student support services or mentorship opportunities

0	offline
1	offline
2	offline
3	offline
4	offline

[5 rows x 35 columns]

df.columns

Index(['Timestamp', 'Name', 'Age in years (enter numeric values only)',
 'Gender', 'You belong to which district of Delhi?',
 'Current Profession', 'Highest Level of Education',
 'Background of Education', 'When do you prefer to do a job ',
 'What is your family income per annum',
 'Have you ever joined coaching (whether online or offline)',
 'Are you seeking coaching from some institution',
 'Does distance matter before preferring a coaching institute?',
 'For what exams are/have you seeking coaching (you can choose

```

more than 1)',
    'Does price matter of coaching class for your preferred course
    ',
    'Which institute do you prefer the most for your offline
    coaching ',
    'Which online platforms have you tried? ',
    'What days you prefer for coaching',
    'How many hours can you study continuously in a day',
    'Your most preferred mode of study',
    'Do you feel that your selected online platform has met your
    learning objectives and exceeded your expectations',
    'On a scale of 1 to 5, how flexible are the course schedules
    and pacing options of online platforms in comparison to offline',
    'Which platform according to you, offers the most convenient
    and accessible learning experience for your needs',
    'Which online platform do you feel have best quality content',
    'On a scale of 1 to 5, how likely are you to recommend your
    preferred online platform to a friend or colleague',
    'Is the online course value for money rather than offline
    course',
    'Considering the course fees and value obtained, which platform
    do you believe offers the best return on investment',
    'What specific aspects of online platform made you think it is
    the best platform ',
    'What specific aspects of offline platform made you think it is
    the best platform',
    'On a scale of 1 to 5, how would you rate the diversity and
    depth of course offerings across the online platform',
    'On a scale of 1 to 5, how do the instructors on online
    platforms compared to offline platforms in terms of expertise and
    teaching effectiveness',
    'Which platform according to you, offers the most interactive
    and engaging teaching methods',
    'On a scale of 1 to 5, how user-friendly and intuitive are the
    online learning platforms',
    'Which platform according to you, offers the best balance of
    structured learning materials, hands-on projects, and interactive
    elements',
    'Which platform according to you, provide superior student
    support services or mentorship opportunities'],
    dtype='object')

df=df.drop('Timestamp',axis=1)

df.isnull().sum()

Name
194
Age in years (enter numeric values only)
0

```

Gender

0

You belong to which district of Delhi?

0

Current Profession

0

Highest Level of Education

0

Background of Education

0

When do you prefer to do a job

0

What is your family income per annum

0

Have you ever joined coaching (whether online or offline)

0

Are you seeking coaching from some institution

0

Does distance matter before preferring a coaching institute?

0

For what exams are/have you seeking coaching (you can choose more than 1)

0

Does price matter of coaching class for your preferred course

0

Which institute do you prefer the most for your offline coaching

0

Which online platforms have you tried?

0

What days you prefer for coaching

0

How many hours can you study continuously in a day

0

Your most preferred mode of study

0

Do you feel that your selected online platform has met your learning objectives and exceeded your expectations

0

On a scale of 1 to 5, how flexible are the course schedules and pacing options of online platforms in comparison to offline

0

Which platform according to you, offers the most convenient and accessible learning experience for your needs

0

Which online platform do you feel have best quality content

0

On a scale of 1 to 5, how likely are you to recommend your preferred online platform to a friend or colleague

0

```

Is the online course value for money rather than offline course
0
Considering the course fees and value obtained, which platform do you
believe offers the best return on investment
0
What specific aspects of online platform made you think it is the best
platform
0
What specific aspects of offline platform made you think it is the
best platform
0
On a scale of 1 to 5, how would you rate the diversity and depth of
course offerings across the online platform
0
On a scale of 1 to 5, how do the instructors on online platforms
compared to offline platforms in terms of expertise and teaching
effectiveness
0
Which platform according to you, offers the most interactive and
engaging teaching methods
0
On a scale of 1 to 5, how user-friendly and intuitive are the online
learning platforms
0
Which platform according to you, offers the best balance of structured
learning materials, hands-on projects, and interactive elements
0
Which platform according to you, provide superior student support
services or mentorship opportunities
3
dtype: int64

df=df.drop('Name',axis=1)

new_column=['Age','Gender','Area','Profession','Education','Stream','J
ob_pref','Family_income','Hx_coaching','current_coaching','Distance_ma
tter','Exams_coaching','Price_matter','offline_institute','online_plat
form','coaching_days','study_hours','mode_of_study','onlineplatform_me
t_expectations','online_platform_flexible','most_convienent_platform',
'best_quality_content','recommend_onlineplatform_friend','online_value
_for_money_than_offline','best_return_on_investment_platform','best_as
pects_online','best_aspects_offline','online_platform_diversity','inst
ructors_online_vs_offline','most_interactive_platform','user_friendly_
online','best_learning_material_platform','superior_support_service_pl
atform']

df.columns=new_column

df.columns

```

```
Index(['Age', 'Gender', 'Area', 'Profession', 'Education', 'Stream',
      'Job_pref', 'Family_income', 'Hx_coaching', 'current_coaching',
      'Distance_matter', 'Exams_coaching', 'Price_matter',
      'offline_institute', 'online_platform', 'coaching_days',
      'study_hours',
      'mode_of_study', 'onlineplatform_met_expectations',
      'online_platform_flexible', 'most_convienent_platform',
      'best_quality_content', 'recommend_onlineplatform_friend',
      'online_value_for_money_than_offline',
      'best_return_on_investment_platform', 'best_aspects_online',
      'best_aspects_offline', 'online_platform_diversity',
      'instructors_online_vs_offline', 'most_interactive_platform',
      'user_friendly_online', 'best_learning_material_platform',
      'superior_support_service_platform'],
      dtype='object')
```

```
df.shape
```

```
(271, 33)
```

```
df.head()
```

	Age	Gender	Area	Profession	Education
Stream \					
0	24	Female	North-West	Student	Graduate
Commerce					
1	23	Female	New Delhi	Working Professional	Post Graduate
Commerce					
2	23	Male	North-West	Student	Post Graduate
Commerce					
3	24	Male	North-West	Working Professional	Graduate
Science					
4	25	Male	New Delhi	Working Professional	Post Graduate
Science					

	Job_pref	Family_income	Hx_coaching
current_coaching ... \			
0	After Graduation	Above 15LPA	Yes
No ...			
1	After Graduation	9LPA-11LPA	No
No ...			
2	After Post Graduation	9LPA-11LPA	Yes
Yes ...			
3	After Graduation	Above 15LPA	Yes
No ...			
4	After Post Graduation	12LPA-15LPA	Yes
Yes ...			

	online_value_for_money_than_offline
best_return_on_investment_platform \	

0		yes
online		
1		yes
online		
2		no
online		
3		yes
offline		
4		yes
offline		

	best_aspects_online	best_aspects_offline \
0	Any time access	Personal Touch
1	All of the above	Classroom study with peers
2	No travelling Expense	Personal Touch
3	I don't like online platforms	All of the above
4	All of the above	All of the above

	online_platform_diversity	instructors_online_vs_offline \
0	3	3
1	4	4
2	5	5
3	3	5
4	4	3

	most_interactive_platform	user_friendly_online \
0	offline	4
1	offline	4
2	offline	4
3	offline	2
4	offline	3

	best_learning_material_platform	superior_support_service_platform
0	offline	offline
1	offline	offline
2	offline	offline
3	offline	offline
4	offline	offline

[5 rows x 33 columns]

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 271 entries, 0 to 270

Data columns (total 33 columns):

#	Column	Non-Null Count	Dtype
---	-----	-----	-----
0	Age	271 non-null	int64
1	Gender	271 non-null	object

2	Area	271	non-null	object
3	Profession	271	non-null	object
4	Education	271	non-null	object
5	Stream	271	non-null	object
6	Job_pref	271	non-null	object
7	Family_income	271	non-null	object
8	Hx_coaching	271	non-null	object
9	current_coaching	271	non-null	object
10	Distance_matter	271	non-null	object
11	Exams_coaching	271	non-null	object
12	Price_matter	271	non-null	object
13	offline_institute	271	non-null	object
14	online_platform	271	non-null	object
15	coaching_days	271	non-null	object
16	study_hours	271	non-null	object
17	mode_of_study	271	non-null	object
18	onlineplatform_met_expectations	271	non-null	object
19	online_platform_flexible	271	non-null	int64
20	most_convenient_platform	271	non-null	object
21	best_quality_content	271	non-null	object
22	recommend_onlineplatform_friend	271	non-null	int64
23	online_value_for_money_than_offline	271	non-null	object
24	best_return_on_investment_platform	271	non-null	object
25	best_aspects_online	271	non-null	object
26	best_aspects_offline	271	non-null	object
27	online_platform_diversity	271	non-null	int64
28	instructors_online_vs_offline	271	non-null	int64
29	most_interactive_platform	271	non-null	object
30	user_friendly_online	271	non-null	int64
31	best_learning_material_platform	271	non-null	object
32	superior_support_service_platform	268	non-null	object

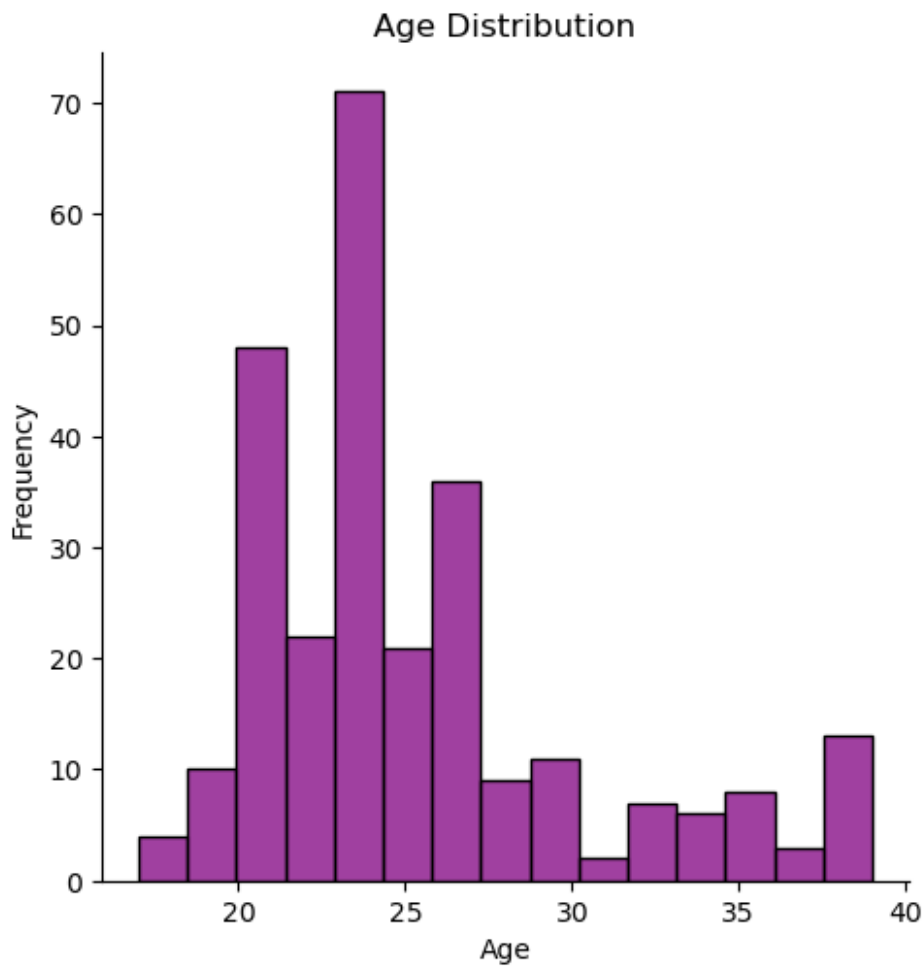
dtypes: int64(6), object(27)
memory usage: 70.0+ KB

```
# Show labels in bar chart
def ShowLabels(ax):
    for data in ax.containers: ax.bar_label(data)
```

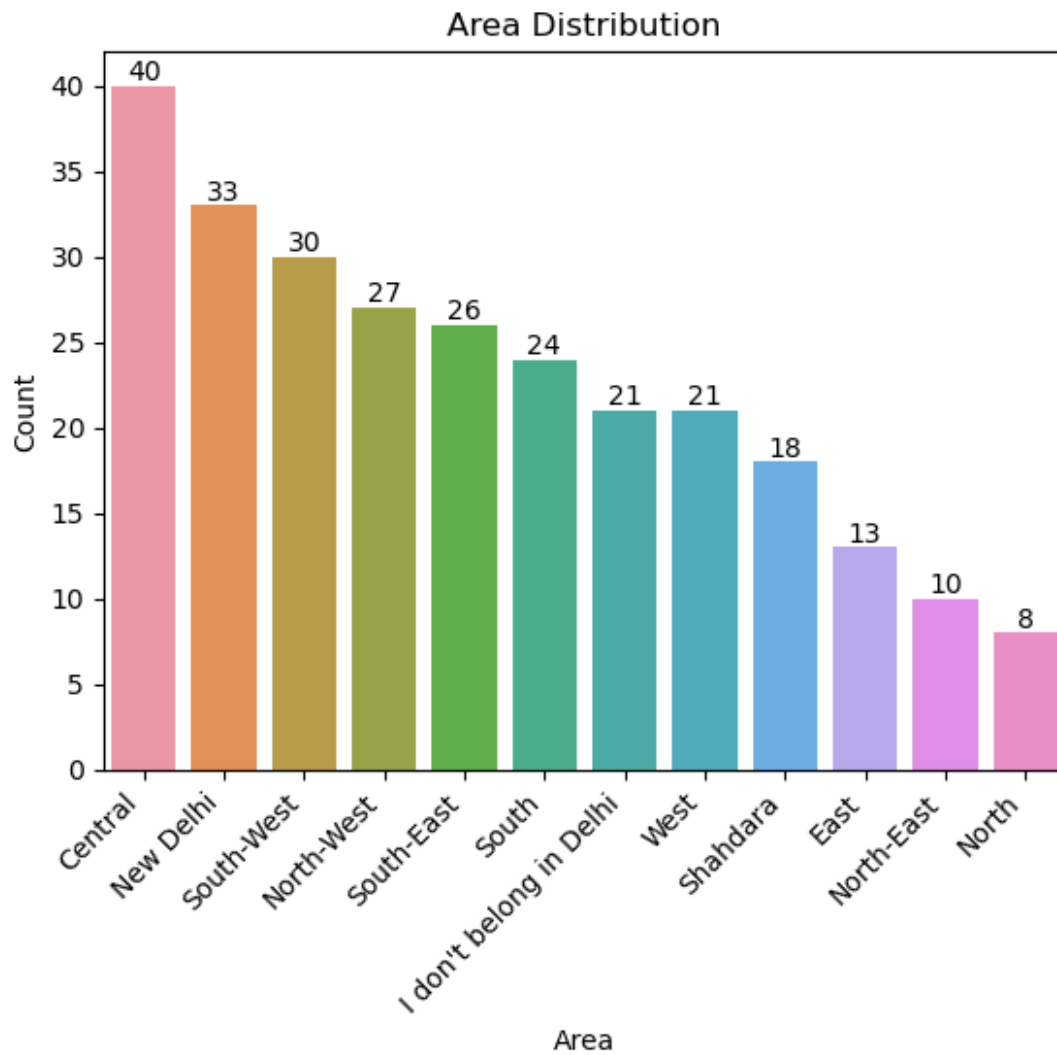
Exploratory Data Analysis (EDA)

```
#Insight 1- Age Distribution
sns.displot(df['Age'], kind='hist', color='purple', legend=True)
plt.title('Age Distribution')
plt.xlabel("Age")
plt.ylabel('Frequency')
plt.savefig('Insight1.png')
plt.show()
```

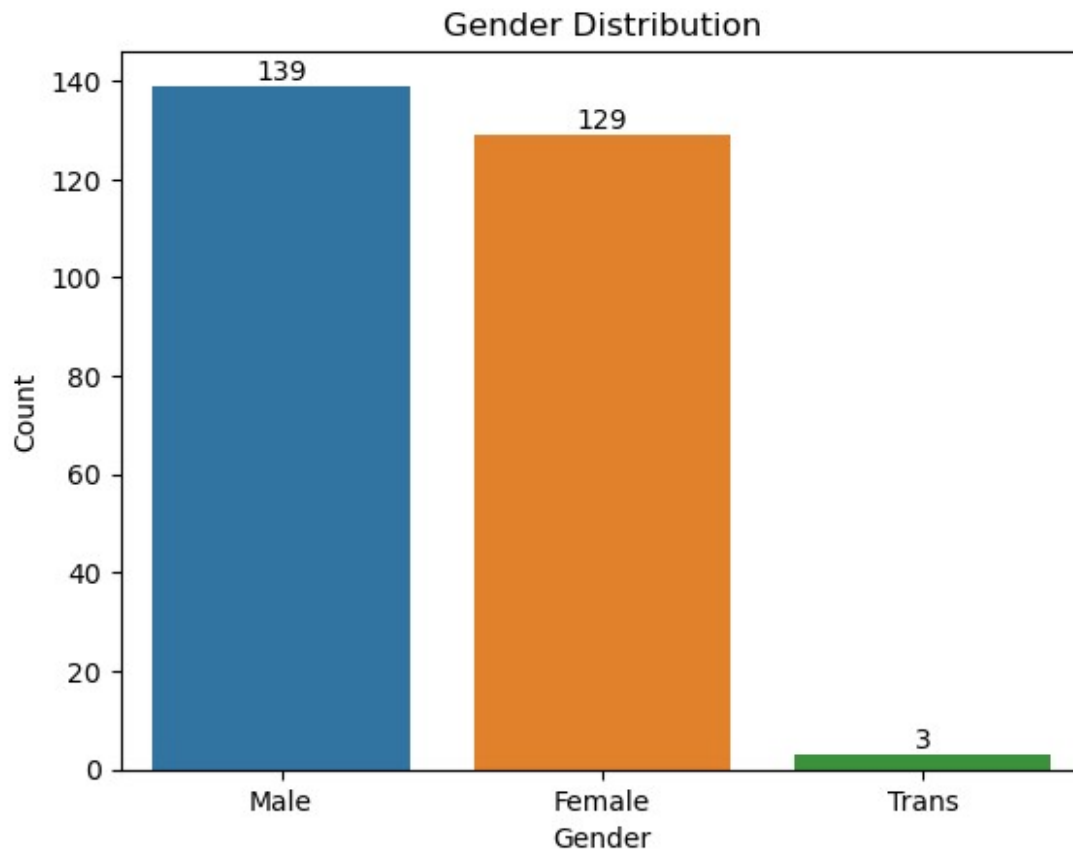
```
C:\Users\KRITI\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118:
UserWarning: The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)
```



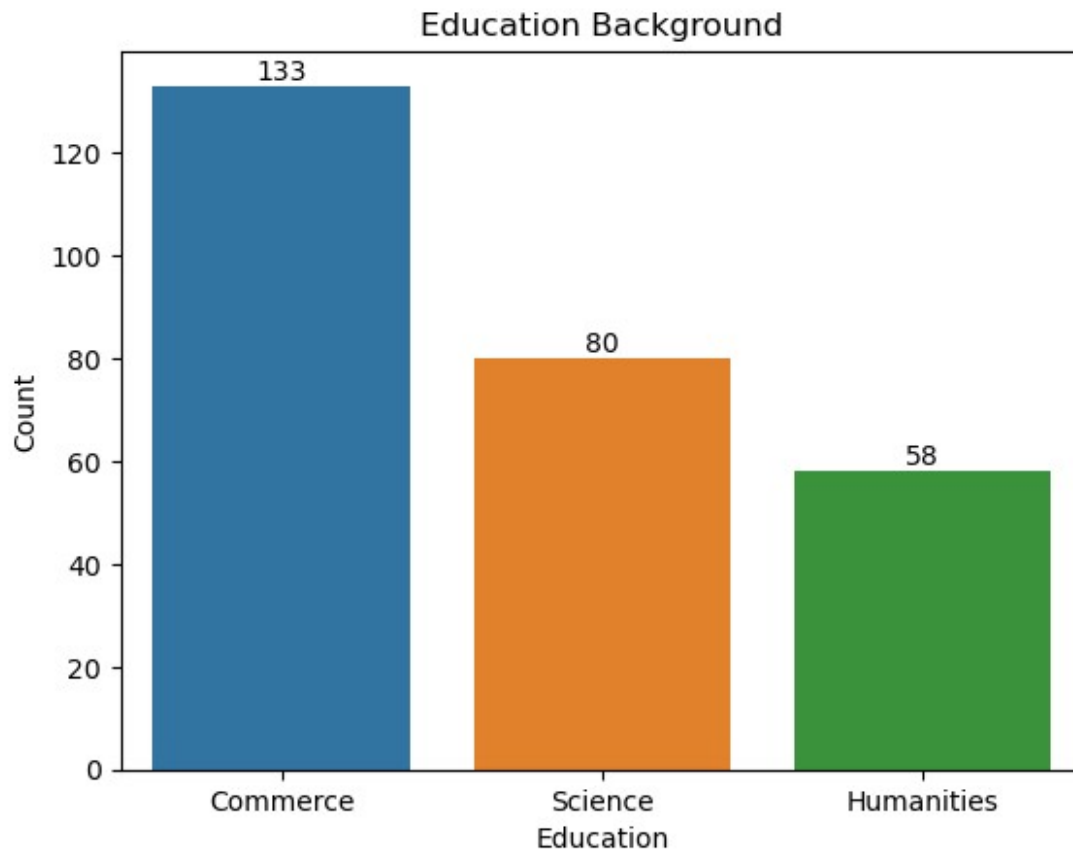
```
#Insight 2- Area Distribution
ax=sns.countplot(x='Area',data=df,order =
df['Area'].value_counts().index)
plt.title('Area Distribution')
plt.xlabel("Area")
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
ShowLabels(ax)
plt.savefig('Insight2.png')
plt.show()
```



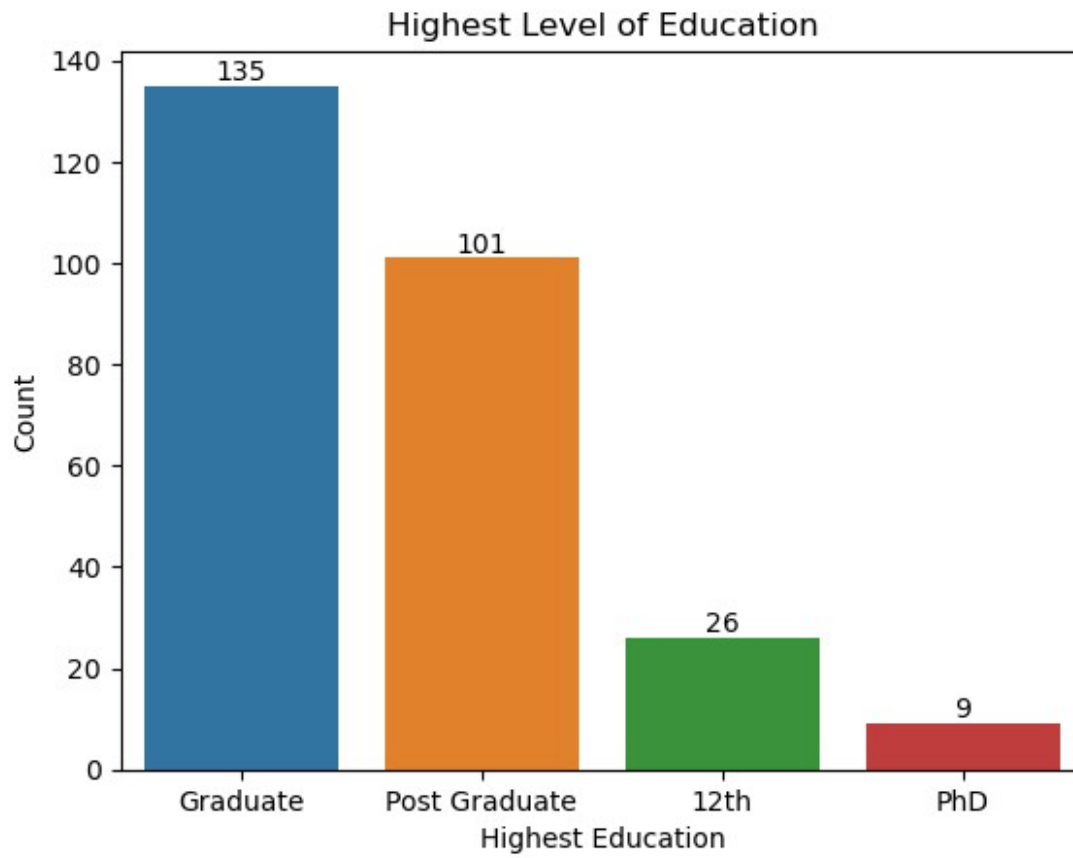
```
#Insight 3- Gender Distribution  
ax=sns.countplot(x='Gender',data=df, order =  
df['Gender'].value_counts().index)  
plt.title('Gender Distribution')  
plt.xlabel('Gender')  
plt.ylabel('Count')  
ShowLabels(ax)  
plt.savefig('Insight3.png')  
plt.show()
```



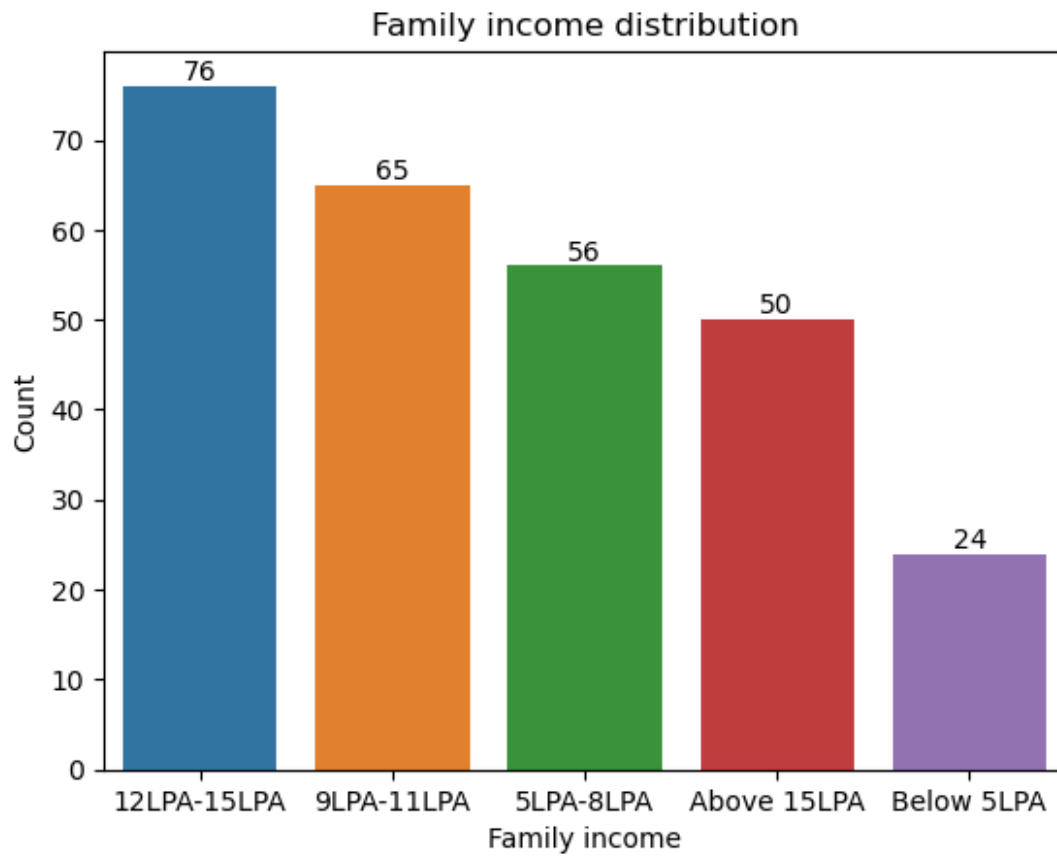
```
#Insight 4 - Education Background  
ax=sns.countplot(x='Stream', data=df,order =  
df['Stream'].value_counts().index)  
plt.title('Education Background')  
plt.xlabel("Education")  
plt.ylabel('Count')  
ShowLabels(ax)  
plt.savefig('Insight4.png')  
plt.show()
```



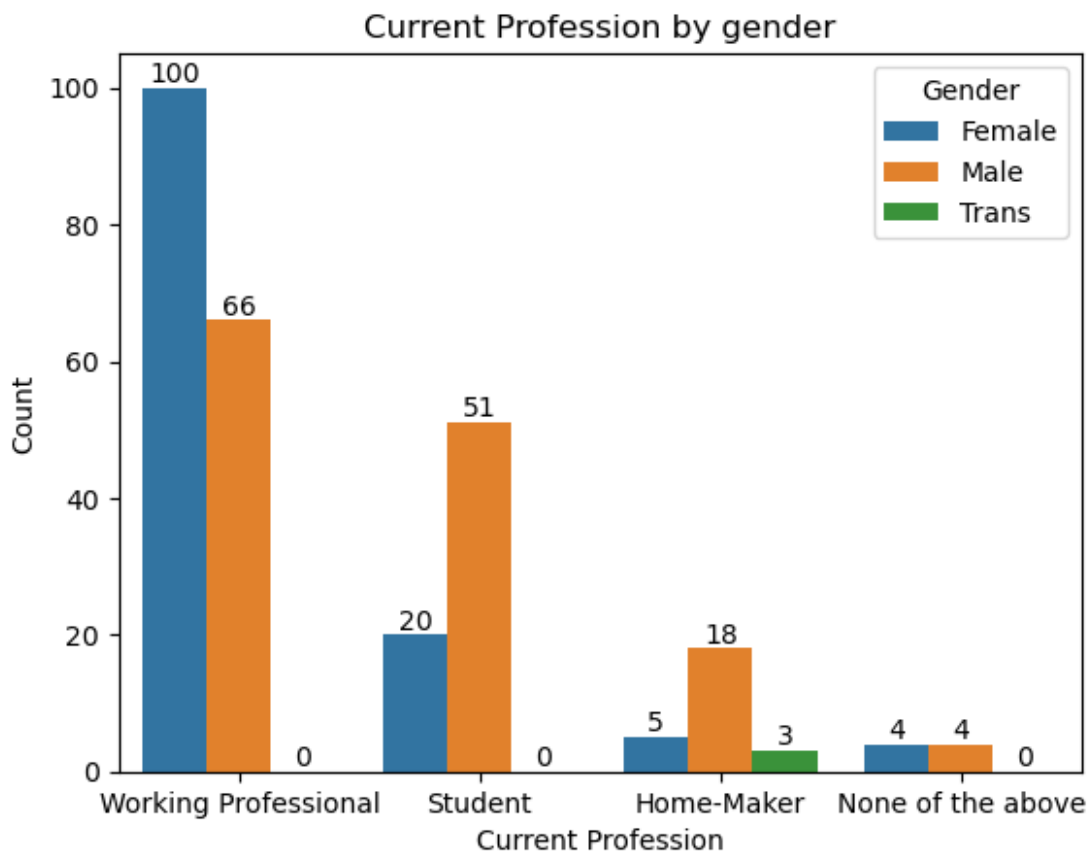
```
#Insight 5- Highest level of education  
ax=sns.countplot(x='Education',data=df,order =  
df['Education'].value_counts().index)  
plt.title('Highest Level of Education')  
plt.xlabel("Highest Education")  
plt.ylabel('Count')  
ShowLabels(ax)  
plt.savefig('Insight5.png')  
plt.show()
```



```
#Insight 6 - Family income distribution  
ax=sns.countplot(x='Family_income',data=df,order =  
df['Family_income'].value_counts().index)  
plt.title('Family income distribution')  
plt.xlabel("Family income")  
plt.ylabel('Count')  
ShowLabels(ax)  
plt.savefig('Insight6.png')  
plt.show()
```



```
#Insight 7 - Current Profession by gender  
ax=sns.countplot(x='Profession', data=df, hue='Gender',order =  
df['Profession'].value_counts().index)  
plt.title('Current Profession by gender')  
plt.xlabel("Current Profession")  
plt.ylabel('Count')  
ShowLabels(ax)  
plt.savefig('Insight7.png')  
plt.show()
```

#Insight 8 - job preference according to age

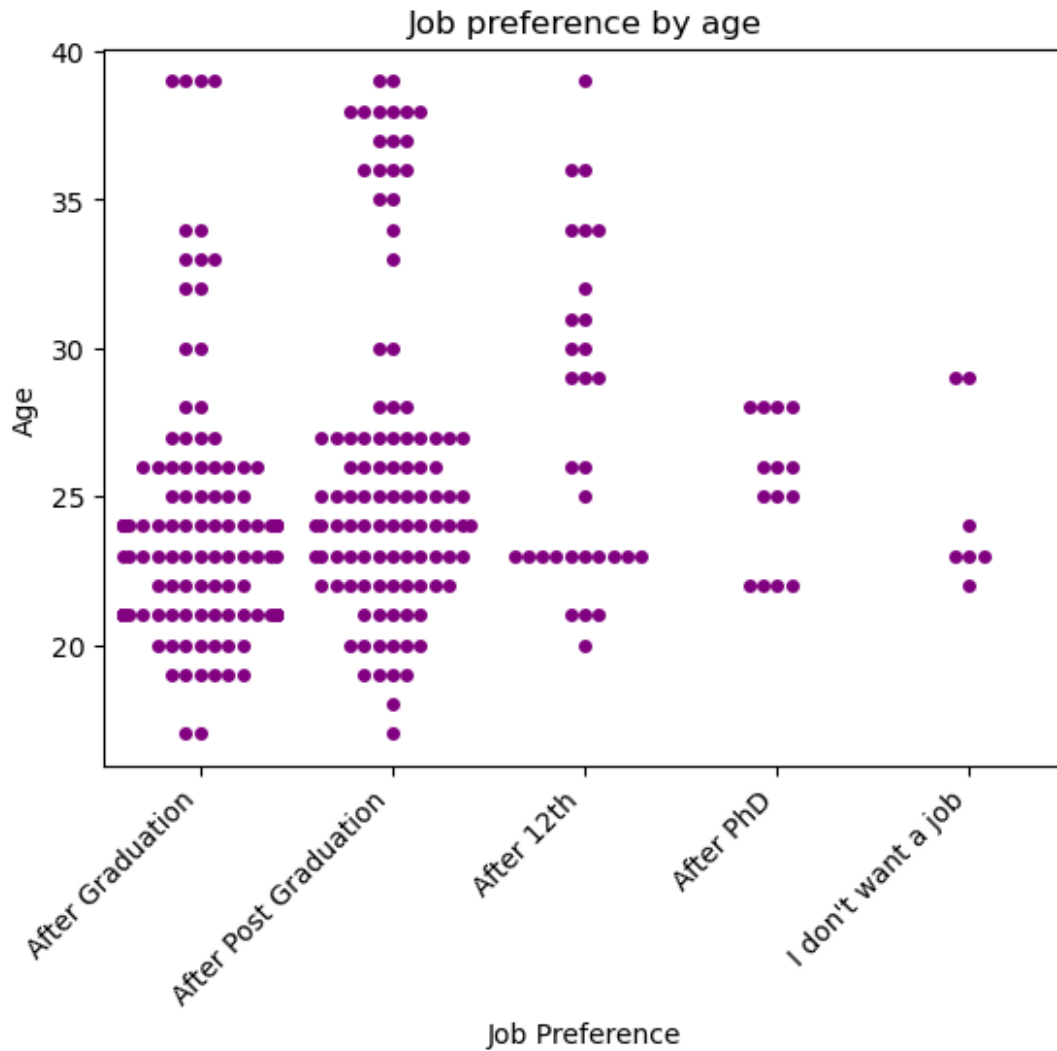
```
sns.swarmplot(x='Job_pref',y='Age',data=df, color='purple')
plt.title('Job preference by age')
plt.xlabel("Job Preference")
plt.ylabel('Age')
plt.xticks(rotation=45, ha='right')
plt.savefig('Insight8.png')
plt.show()
```

C:\Users\KRITI\anaconda3\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 16.7% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

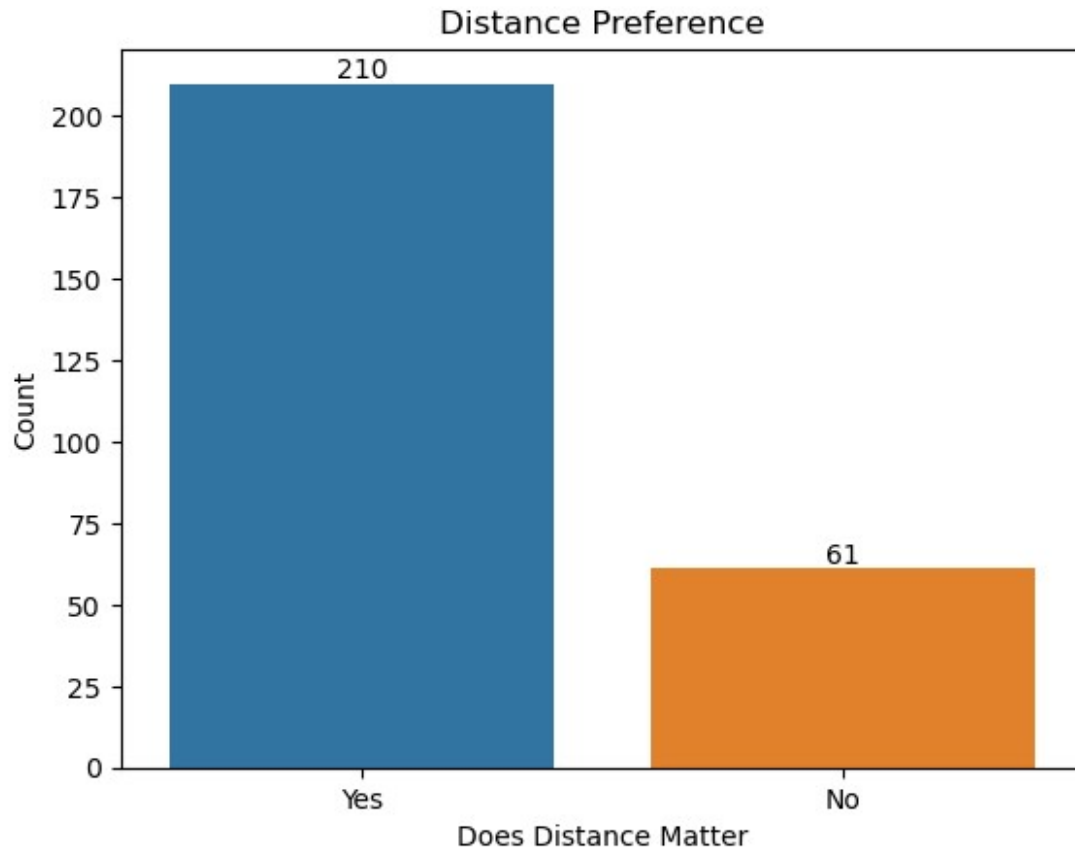
```
warnings.warn(msg, UserWarning)
```

C:\Users\KRITI\anaconda3\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 21.9% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.

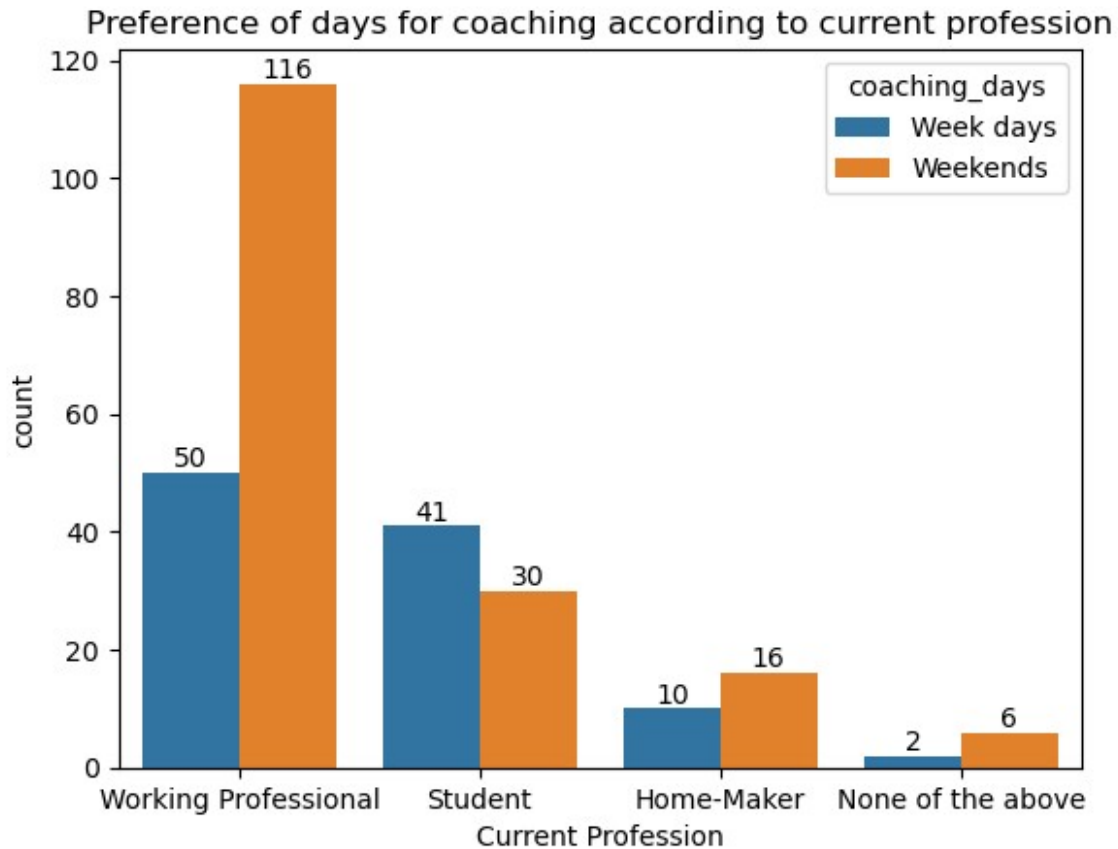
```
warnings.warn(msg, UserWarning)
```



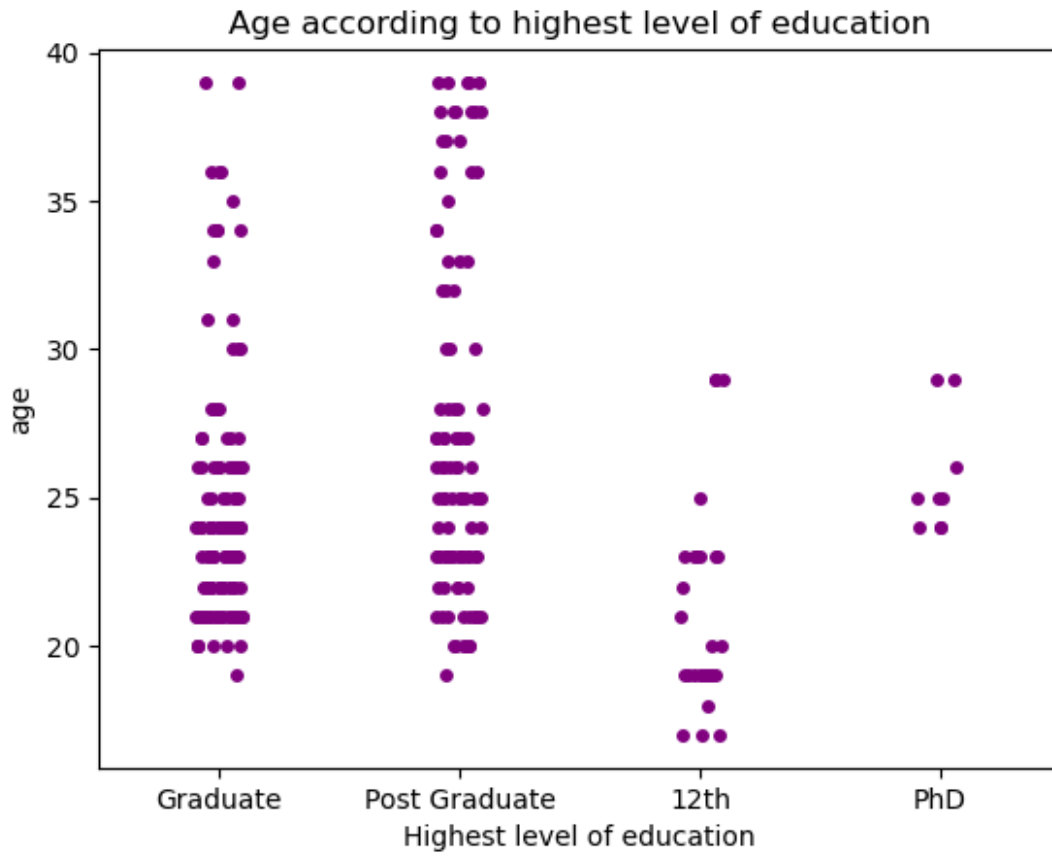
```
#Insight 9 - Does distance matter
ax=sns.countplot(x='Distance_matter',data=df,order =
df['Distance_matter'].value_counts().index)
plt.title('Distance Preference')
plt.xlabel("Does Distance Matter")
plt.ylabel('Count')
ShowLabels(ax)
plt.savefig('Insight9.png')
plt.show()
```



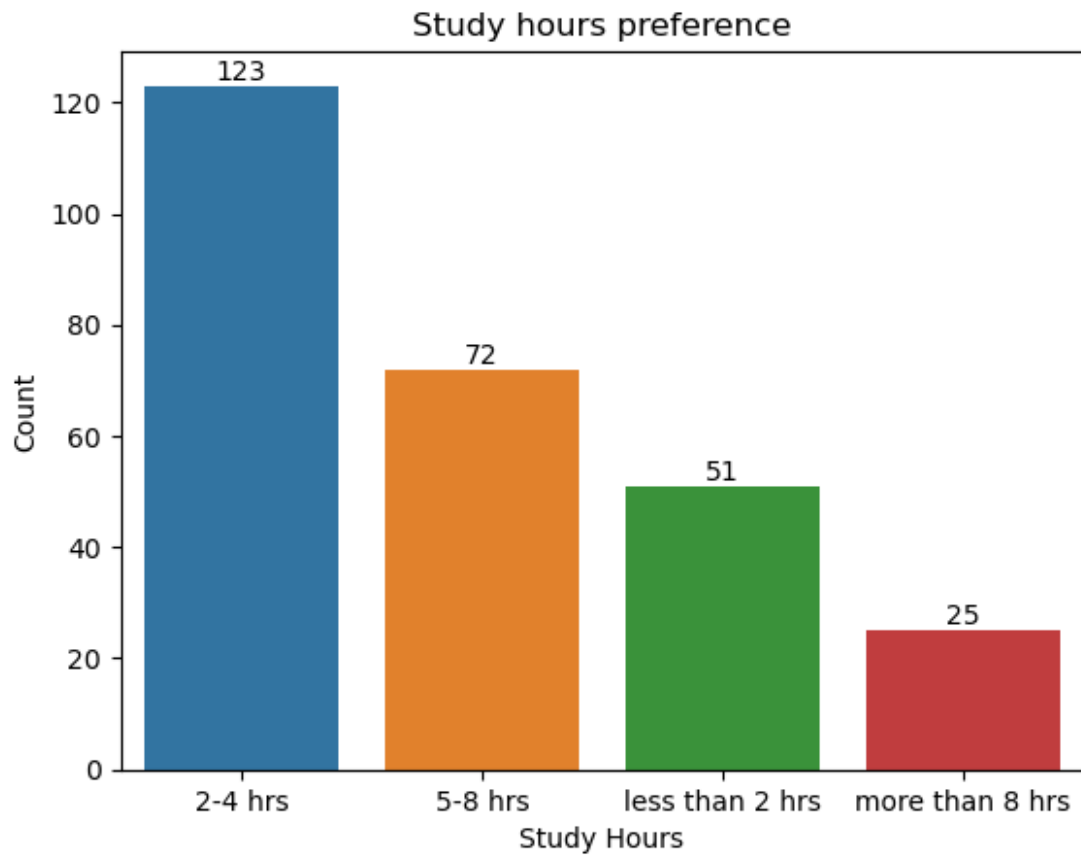
```
#Insight 10 - Preference of days for coaching according to current profession  
ax=sns.countplot(x='Profession',data=df,hue='coaching_days',order =  
df['Profession'].value_counts().index)  
plt.title('Preference of days for coaching according to current  
profession')  
plt.xlabel("Current Profession")  
plt.ylabel('count')  
ShowLabels(ax)  
plt.savefig('Insight10.png')  
plt.show()
```



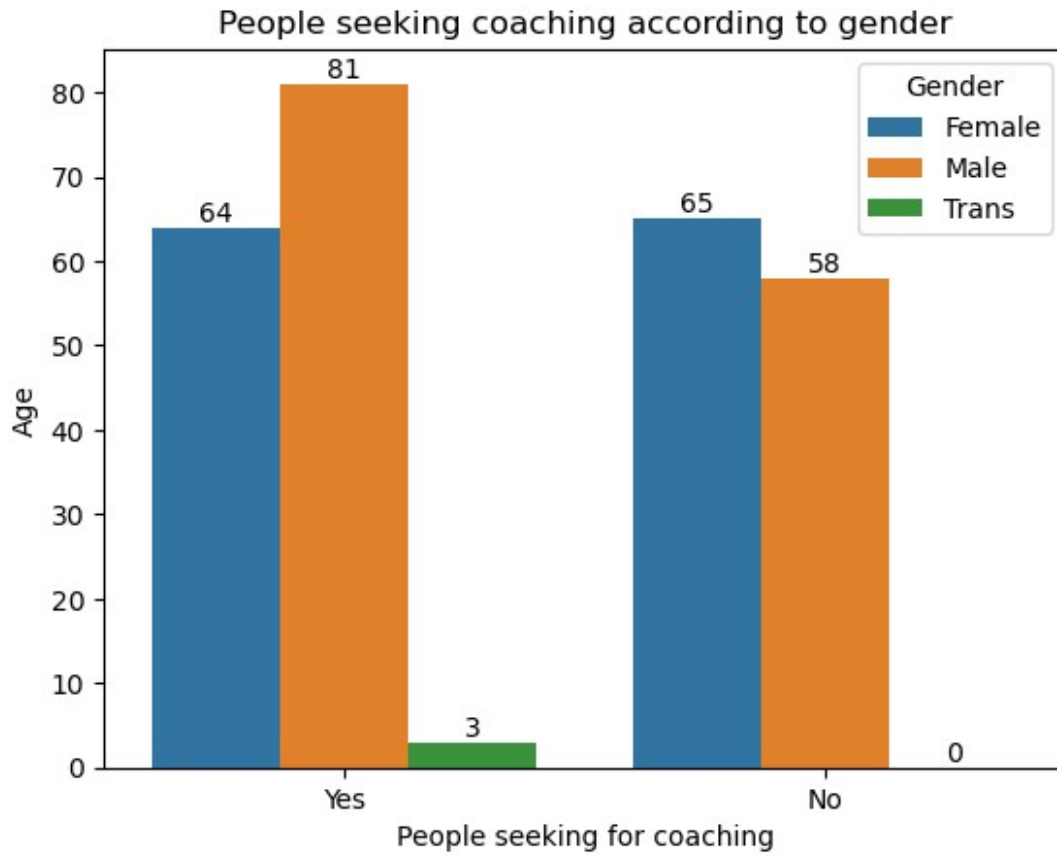
```
#Insight 11 - Age according to highest level of education
sns.stripplot(x='Education',y='Age',data=df, color='purple')
plt.title('Age according to highest level of education')
plt.xlabel("Highest level of education")
plt.ylabel('age')
plt.savefig('Insight11.png')
plt.show()
```



```
#Insight 12 - Hours of study
ax=sns.countplot(x='study_hours',data=df,order =
df['study_hours'].value_counts().index)
plt.title('Study hours preference')
plt.xlabel('Study Hours')
plt.ylabel('Count')
ShowLabels(ax)
plt.savefig('Insight12.png')
plt.show()
```



```
#Insight 13 - People seeking for coaching according gender  
ax=sns.countplot(x='current_coaching',data=df,hue='Gender',order =  
df['current_coaching'].value_counts().index)  
plt.title('People seeking coaching according to gender')  
plt.xlabel('People seeking for coaching')  
plt.ylabel('Age')  
ShowLabels(ax)  
plt.savefig('Insight13.png')  
plt.show()
```



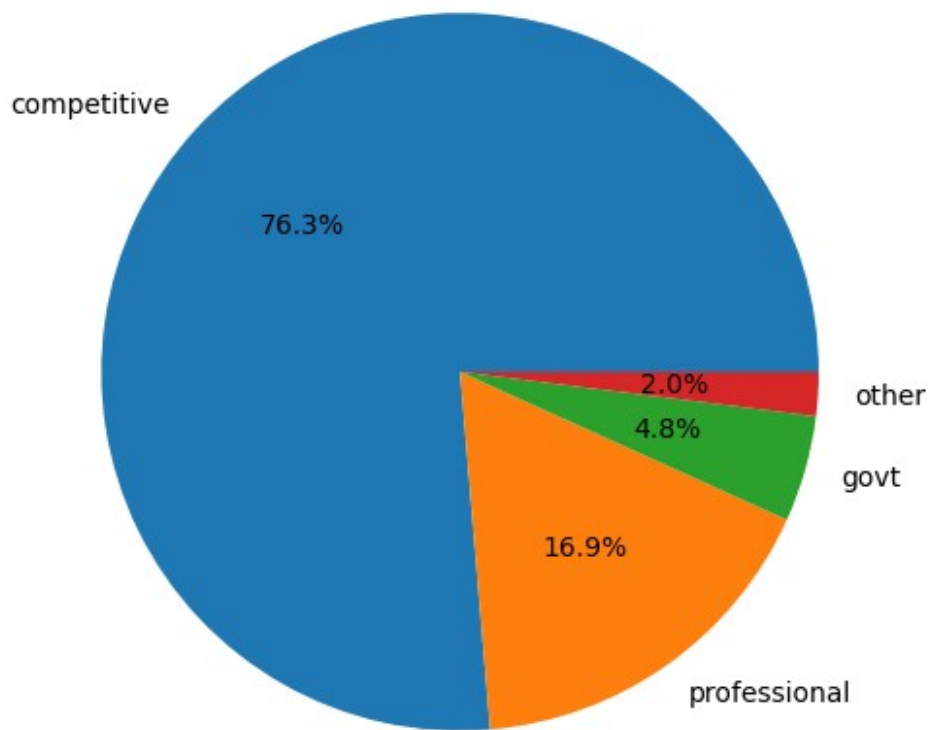
```
#Insight 14 - What exam people prefer according
competitive,professional,govt,other=df['Exams_coaching'].str.split(',')
, expand=True).count()

competitive,professional,govt,other
(271, 60, 17, 7)

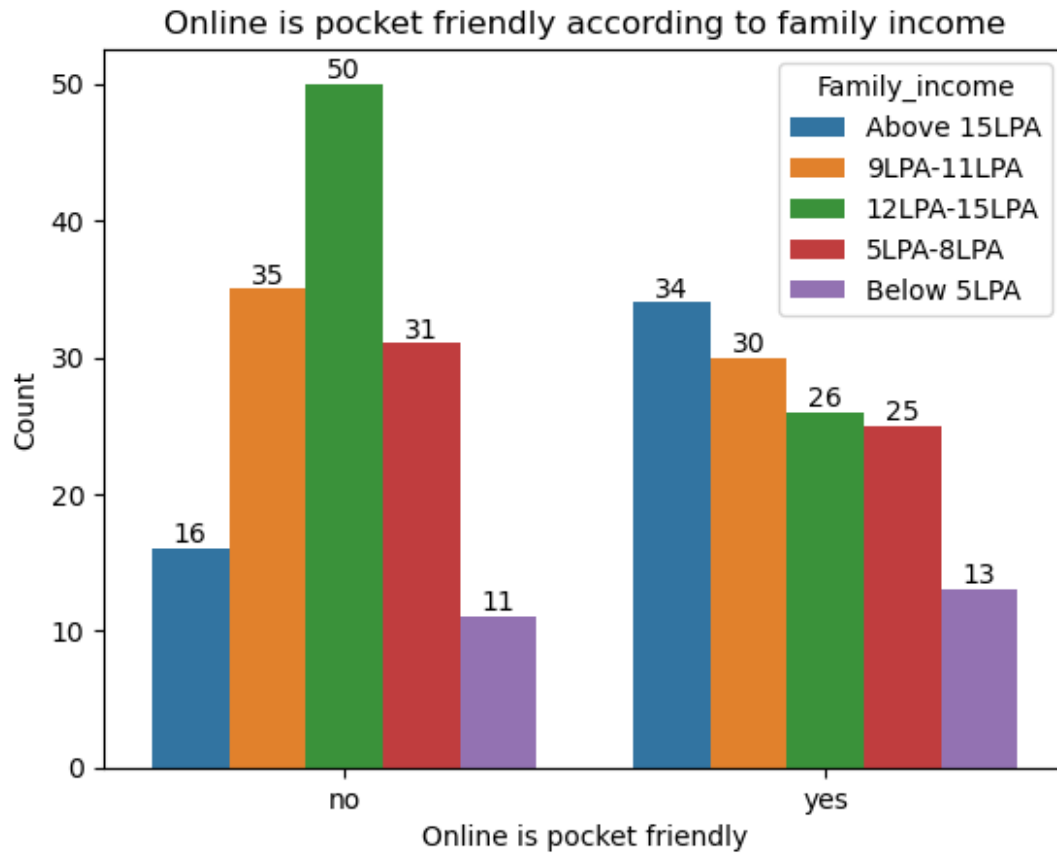
# Data
categories = ['competitive', 'professional', 'govt', 'other']
counts = [271, 60, 17, 7]

# Plot
plt.figure(figsize=(8, 6))
plt.pie(counts, labels=categories, autopct='%1.1f%%')
plt.title('Preference for Exam Coaching Categories')
plt.savefig('Insight14.png')
plt.show()
```

Preference for Exam Coaching Categories



```
#Insight 15 - value of online course rather than offline according to family income  
ax=sns.countplot(x='online_value_for_money_than_offline',data=df,hue='Family_income',order =  
df['online_value_for_money_than_offline'].value_counts().index)  
plt.title('Online is pocket friendly according to family income')  
plt.xlabel('Online is pocket friendly')  
plt.ylabel('Count')  
ShowLabels(ax)  
plt.savefig('Insight15.png')  
plt.show()
```

#Insight 16 - recommendation of online platform to your friend

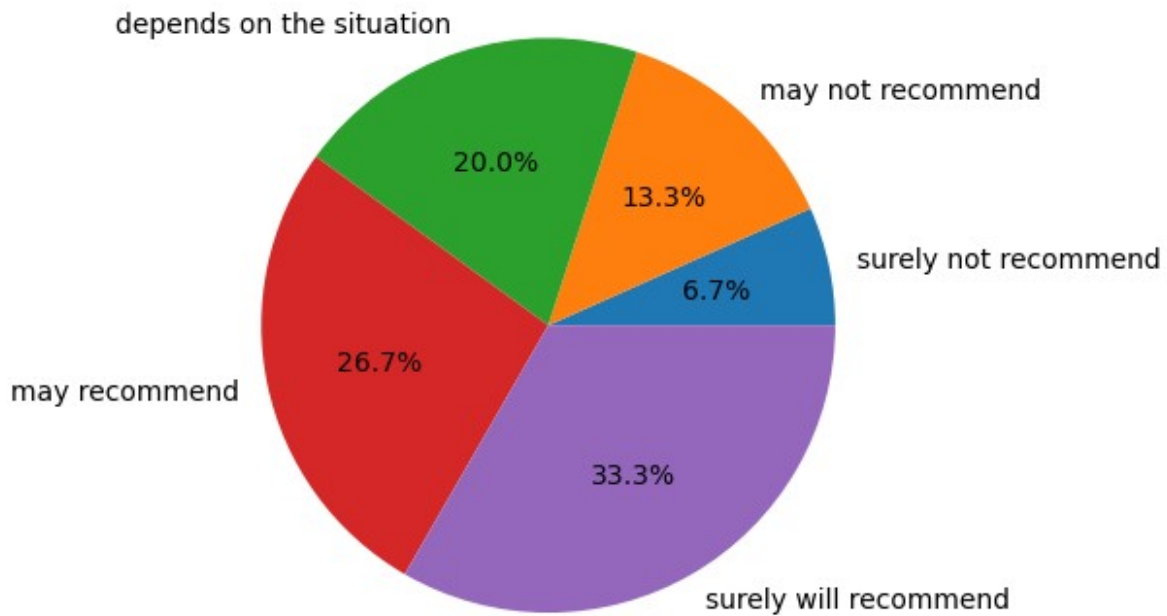
Data

```
categories = ['surely not recommend', 'may not recommend', 'depends on  
the situation', 'may recommend', 'surely will recommend']  
counts = (1,2,3,4,5)
```

Plot

```
plt.pie(counts, labels=categories, autopct='%1.1f%%')  
plt.title('recommendation of online platform to your friend')  
plt.savefig('Insight16.png')  
plt.show()
```

recommendation of online platform to your friend



#Insight 17 - how user friendly online platforms are

Data

```
categories = ['poor', 'below average', 'average', 'above average',  
'best']
```

```
counts = (1,2,3,4,5)
```

Plot

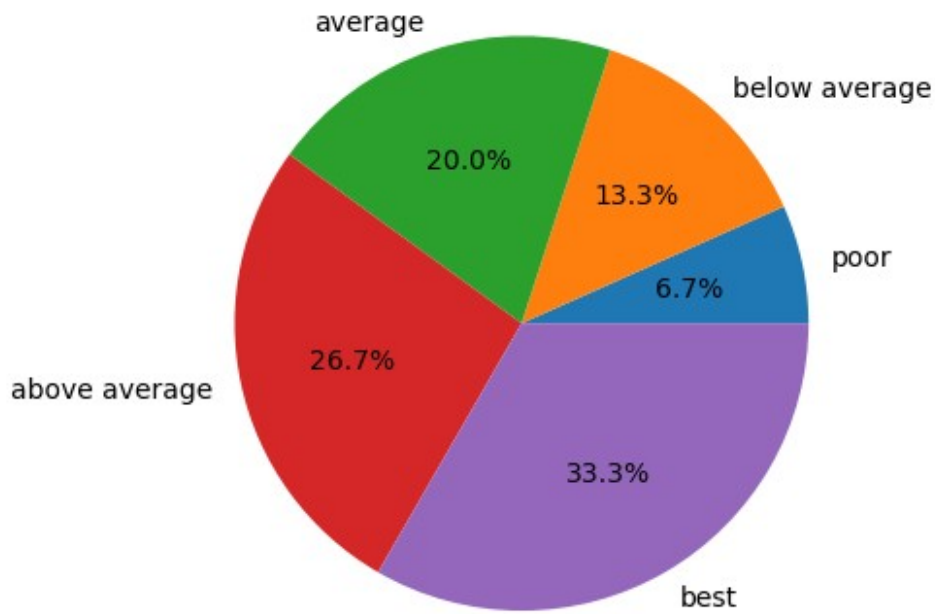
```
plt.pie(counts, labels=categories, autopct='%1.1f%%')
```

```
plt.title('how user friendly online platforms are')
```

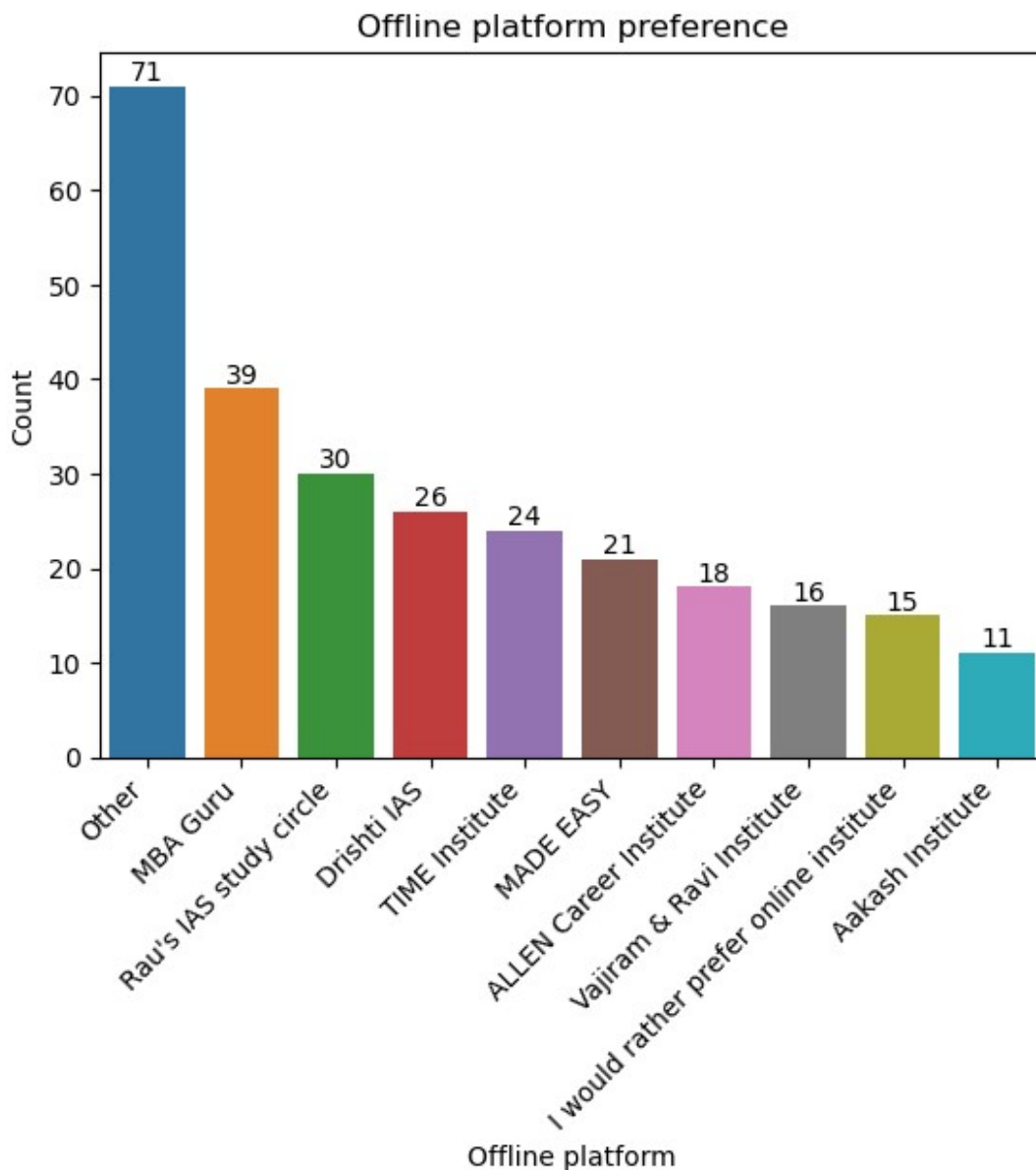
```
plt.savefig('Insight17.png')
```

```
plt.show()
```

how user friendly online platforms are



```
#Insight 18 - what offline platform do they prefer  
ax=sns.countplot(x='offline_institute',data=df,order =  
df['offline_institute'].value_counts().index)  
plt.title('Offline platform preference')  
plt.xlabel('Offline platform')  
plt.ylabel('Count')  
plt.xticks(rotation=45, ha='right')  
ShowLabels(ax)  
plt.savefig('Insight18.png')  
plt.show()
```



#Insight 19 - Top 5 most preferred online platform

Split the strings in the 'online_platform' column and then explode the resulting lists into separate rows

```
exploded_data = df['online_platform'].str.split(',').explode()
```

Count the occurrences of each platform

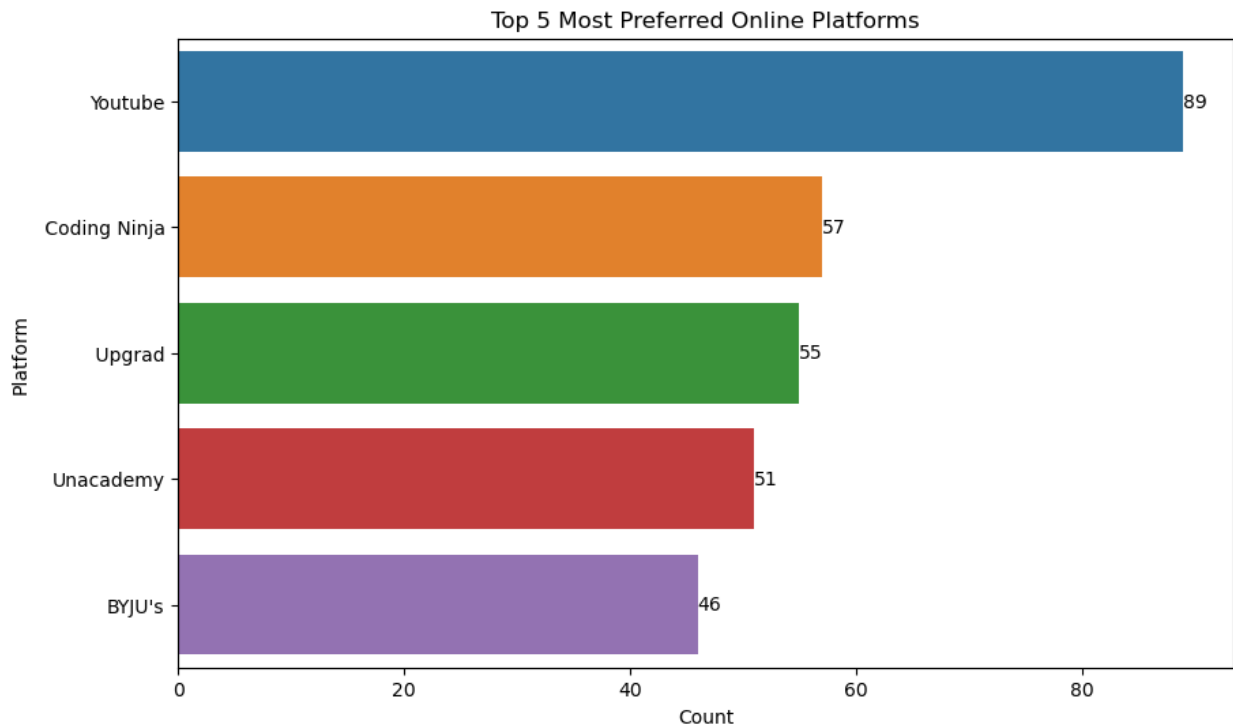
```
platform_counts = exploded_data.value_counts().head(5)
```

```
plt.figure(figsize=(10, 6))
```

```
ax=sns.countplot(y=exploded_data, order=platform_counts.index)
```

```
plt.title('Top 5 Most Preferred Online Platforms')
```

```
plt.xlabel('Count')
plt.ylabel('Platform')
ShowLabels(ax)
plt.savefig('Insight19.png')
plt.show()
```



Classification

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

# Select the target and features for the classification task
target_column = df['mode_of_study']

features = df.drop(columns=['mode_of_study'])

# Encode categorical variables
features_encoded = pd.get_dummies(features, drop_first=True)

# Encode target variable (assuming 'offline' is 0 and 'online' is 1)
target_encoded = target.map({'Offline': 0, 'Online': 1, 'offline': 0, 'online': 1})
```

```

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(features_encoded,
target_encoded, test_size=0.2, random_state=42)

# Initialize the models
logistic_regression_model = LogisticRegression(max_iter=1000)
svm_model = SVC()
random_forest_model = RandomForestClassifier()

# Train the models
logistic_regression_model.fit(X_train, y_train)
svm_model.fit(X_train, y_train)
random_forest_model.fit(X_train, y_train)

RandomForestClassifier()

# Predict on the test set
y_pred_lr = logistic_regression_model.predict(X_test)
y_pred_svm = svm_model.predict(X_test)
y_pred_rf = random_forest_model.predict(X_test)

# Evaluate the models
report_lr = classification_report(y_test, y_pred_lr)
report_svm = classification_report(y_test, y_pred_svm)
report_rf = classification_report(y_test, y_pred_rf)

print("Logistic Regression Report:\n", report_lr)
print("SVM Report:\n", report_svm)
print("Random Forest Report:\n", report_rf)

```

```

Logistic Regression Report:
              precision    recall  f1-score   support

      0       0.89        0.91        0.90         35
      1       0.84        0.80        0.82         20

 accuracy          0.87
 macro avg         0.87
weighted avg         0.87

```

```

SVM Report:
              precision    recall  f1-score   support

      0       0.64        1.00        0.78         35
      1       0.00        0.00        0.00         20

 accuracy          0.64
 macro avg         0.32
weighted avg         0.40

```

Random Forest Report:

	precision	recall	f1-score	support
0	0.79	0.94	0.86	35
1	0.85	0.55	0.67	20
accuracy			0.80	55
macro avg	0.82	0.75	0.76	55
weighted avg	0.81	0.80	0.79	55

```
C:\Users\KRITI\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1517: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
```

```
C:\Users\KRITI\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1517: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
```

```
C:\Users\KRITI\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1517: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
```

```
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
```

```
# Calculate confusion matrices
```

```
cm_lr = confusion_matrix(y_test, y_pred_lr)
cm_svm = confusion_matrix(y_test, y_pred_svm)
cm_rf = confusion_matrix(y_test, y_pred_rf)
```

```
# Plot confusion matrices
```

```
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(15, 5))
```

```
ConfusionMatrixDisplay(cm_lr, display_labels=['offline',
'online']).plot(ax=axes[0], cmap='Blues')
axes[0].set_title('Logistic Regression')
```

```
ConfusionMatrixDisplay(cm_svm, display_labels=['offline',
'online']).plot(ax=axes[1], cmap='Blues')
axes[1].set_title('SVM')
```

```
ConfusionMatrixDisplay(cm_rf, display_labels=['offline',
```

```

'online'])).plot(ax=axes[2], cmap='Blues')
axes[2].set_title('Random Forest')

plt.tight_layout()
plt.savefig('confusion.png')
plt.show()

```

