

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

In [2]: df=pd.read_csv('Sheet1.csv')

DATA DESCRIPTION

In [3]: df.head()

Out[3]:
   Unnamed: 0   ID  Salary  DOJ  DOL  Designation JobCity  Gender  DOB  10percentage  ComputerScience  MechanicalEngg  ElectricalEngg  TelecomEngg  CivilEng  conscientiousness  agreeableness  extraversion  neuroticism  openness_to_experience
0    train  203097  420000.0  6/1/2012 0000  present  senior quality engineer  Bangalore  f  2/19/90 0000  84.3  -1  -1  -1  -1  -1  0.9737  0.8128  0.5269  1.35490  -0.4455
1    train  579905  500000.0  9/1/13 0000  present  assistant manager  Indore  m  10/4/89 0000  85.4  -1  -1  -1  -1  -1  0.7335  0.3789  1.2396  -0.10760  0.8637
2    train  810601  325000.0  6/1/14 0000  present  systems engineer  Chennai  f  8/2/92 0000  85.0  -1  -1  -1  -1  -1  0.2718  1.7109  0.1637  -0.86820  0.6721
3    train  267447  1100000.0  7/1/14 0000  present  senior software engineer  Gurgaon  m  12/5/89 0000  85.6  -1  -1  -1  -1  -1  0.0464  0.3448  -0.3440  -0.40780  -0.9194
4    train  343523  2000000.0  3/1/14 0000  get  Manesar  m  2/27/91 0000  78.0  -1  -1  -1  -1  -1  -0.8810  -0.2793  -1.0697  0.09163  -0.1295

5 rows x 39 columns

In [4]: df.shape
Out[4]: (3998, 39)

In [5]: df.describe()

Out[5]:
      ID  Salary  10percentage  12graduation  12percentage  CollegeID  CollegeTier  collegeGPA  CollegeCityID  CollegeCityTier  ComputerScience  MechanicalEngg  ElectricalEngg  TelecomEngg  CivilEng  conscientiousness  agreeableness  extraversion  neuroticism  openness_to_experience
count  3.998000e+03  3.998000e+03  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000  3998.000000
mean  6.63745e+05  3.706998e+05  77.525443  2008.087544  74.466366  5156.851426  1.925713  71.466171  5156.851426  0.300400  90.742371  22.974737  16.478739  31.851176  2.683842  -0.037831  0.146496  0.002763  -0.169033  -0.138110
std  3.632182e+05  2.127375e+05  9.800162  1.653599  10.999933  4802.261482  0.262270  8.167330  4802.261482  0.458489  175.273083  98.123311  87.585634  104.852845  36.658505  1.028666  0.941782  0.951471  0.007580  1.008075
min  1.124400e+04  3.500000e+04  43.000000  195.000000  40.000000  2.000000  1.000000  6.450000  2.000000  0.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000
25%  3.342842e+05  1.800000e+05  71.680000  2007.000000  66.000000  494.000000  2.000000  66.407500  494.000000  0.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000
50%  6.396000e+05  3.000000e+05  79.150000  2008.000000  74.400000  3879.000000  2.000000  71.720000  3879.000000  0.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000
75%  9.904800e+05  3.700000e+05  85.670000  2009.000000  82.600000  8818.000000  2.000000  76.527500  8818.000000  1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000
max  1.298275e+06  4.000000e+06  97.760000  2013.000000  98.700000  18409.000000  2.000000  99.930000  18409.000000  1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000

8 rows x 27 columns

In [6]: df.info()

RangeIndex: 3998 entries, 0 to 3997
Data columns (total 39 columns):
 # Column          Non-Null Count  Dtype  
--- 
 0   Unnamed: 0       3998 non-null   object 
 1   ID              3998 non-null   int64  
 2   Salary           3998 non-null   float64
 3   DOJ             3998 non-null   object 
 4   DOL             3998 non-null   object 
 5   Designation     3998 non-null   object 
 6   JobCity          3998 non-null   object 
 7   Gender           3998 non-null   object 
 8   DOB              3998 non-null   object 
 9   10percentage    3998 non-null   float64
 10  IBoard            3998 non-null   float64
 11  12graduation     3998 non-null   float64
 12  12percentage    3998 non-null   float64
 13  IBoard            3998 non-null   object 
 14  CollegeID        3998 non-null   int64  
 15  CollegeTier      3998 non-null   int64  
 16  Degree            3998 non-null   object 
 17  Specialization   3998 non-null   object 
 18  collegeGPA        3998 non-null   float64
 19  CollegeCityID    3998 non-null   int64  
 20  CollegeCityTier  3998 non-null   int64  
 21  CollegeState      3998 non-null   object 
 22  EducationYear     3998 non-null   int64  
 23  English           3998 non-null   int64  
 24  Logical            3998 non-null   int64  
 25  Quant             3998 non-null   int64  
 26  Domain             3998 non-null   float64
 27  ComputerProgramming 3998 non-null   int64  
 28  ElectronicsDesign 3998 non-null   int64  
 29  ComputerScience    3998 non-null   int64  
 30  MechanicalEngg    3998 non-null   int64  
 31  ElectricalEngg    3998 non-null   int64  
 32  TelecommEngg      3998 non-null   int64  
 33  Agreeableness     3998 non-null   float64
 34  Conscientiousness  3998 non-null   float64
 35  Extraversion        3998 non-null   float64
 36  Neuroticism        3998 non-null   float64
 37  OpennessToExperience 3998 non-null   float64
 38  conscientiousness  3998 non-null   float64
 39  agreeableness     3998 non-null   float64
dtypes: float64(18), int64(17), object(12)
memory usage: 1.2e+06
```

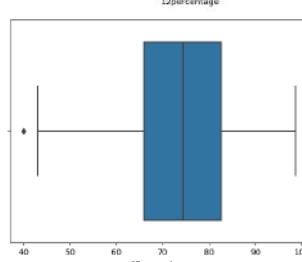
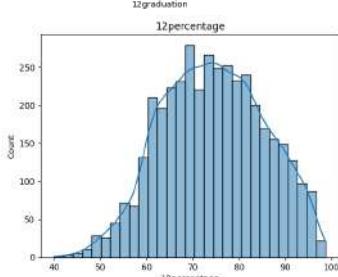
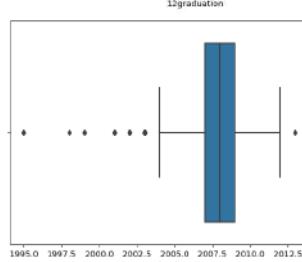
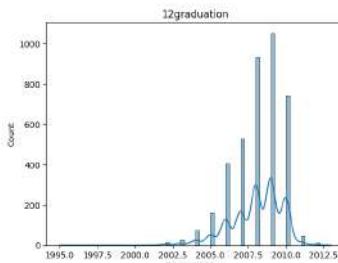
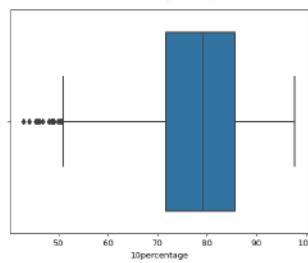
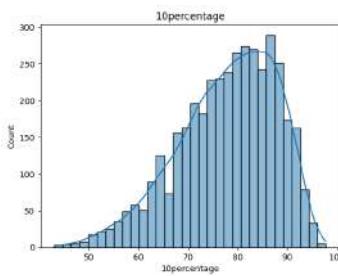
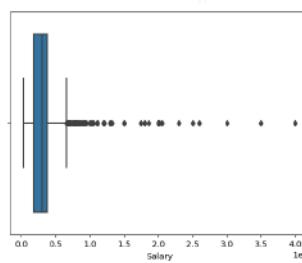
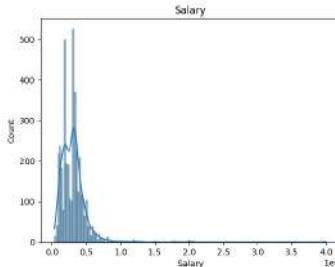
```
In [7]: df.isnull().sum()

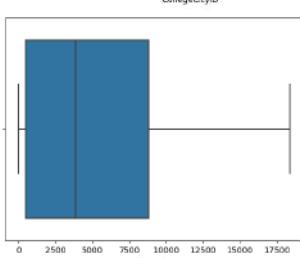
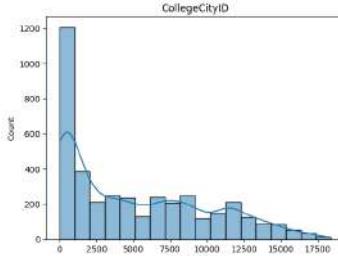
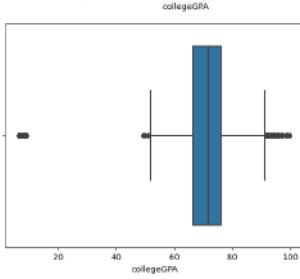
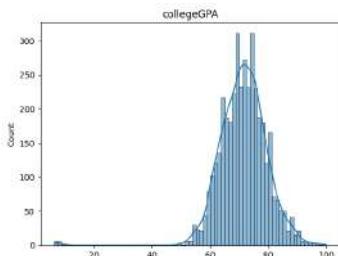
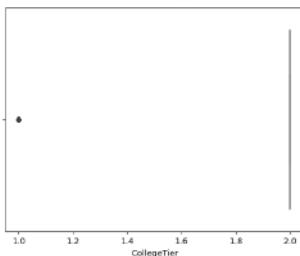
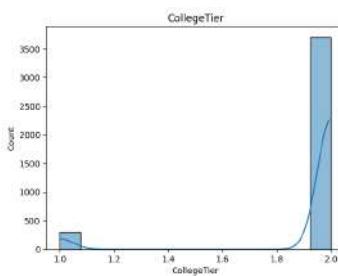
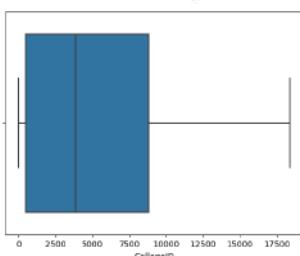
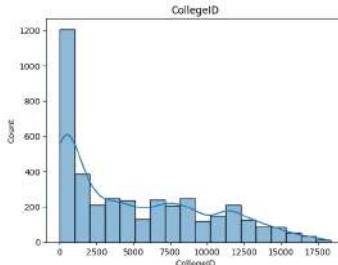
Out[7]:
   Unnamed: 0      0
   ID              0
   Salary           0
   DOJ             0
   DOL             0
   Designation     0
   JobCity          0
   Gender           0
   DOB              0
   10percentage    0
   IBoard            0
   12graduation     0
   12percentage    0
   IBoard            0
   CollegeID        0
   CollegeTier      0
   Degree            0
   Specialization   0
   collegeGPA        0
   CollegeCityID    0
   CollegeCityTier  0
   CollegeState      0
   EducationYear     0
   English           0
   Logical            0
   Quant             0
   Domain             0
   ComputerProgramming 0
   ElectronicsDesign 0
   ComputerScience    0
   MechanicalEngg    0
   ElectricalEngg    0
   TelecommEngg      0
   Agreeableness     0
   Conscientiousness 0
   extraversion       0
   neuroticism        0
   openness_to_experience 0
dtype: int64
```

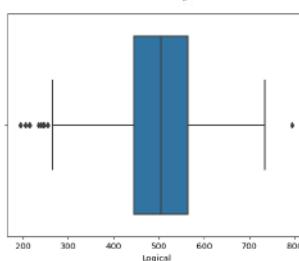
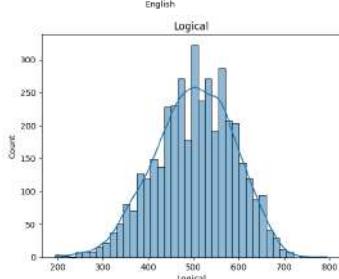
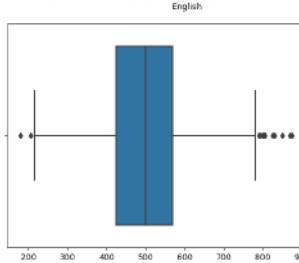
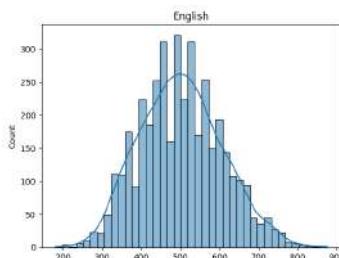
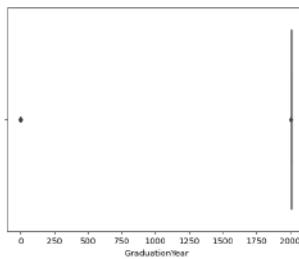
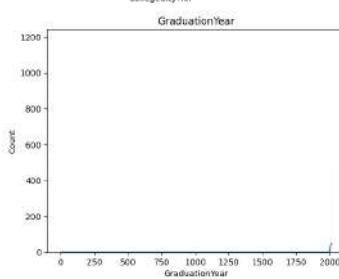
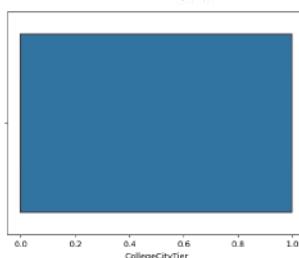
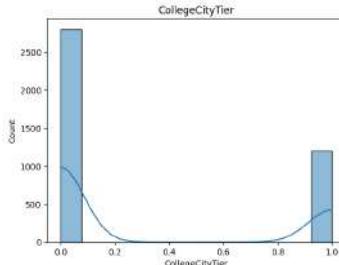
UNIVARIATE ANALYSIS

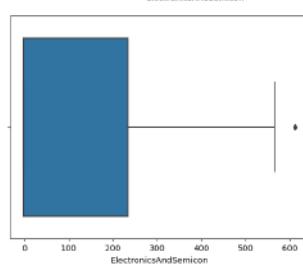
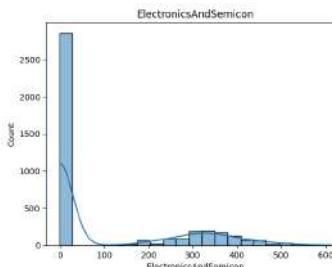
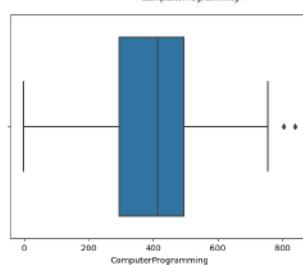
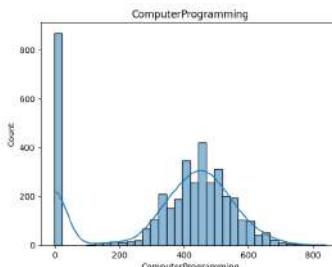
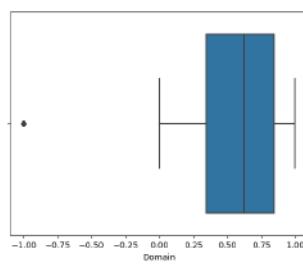
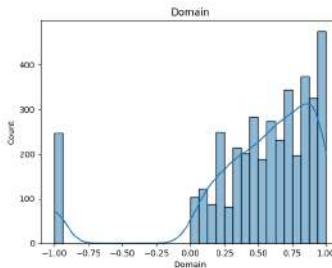
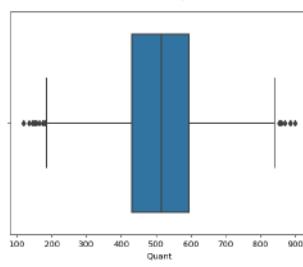
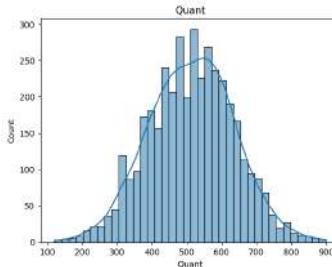
```
In [8]: numerical_cols=df.select_dtypes(include=['float64','int64']).columns
```

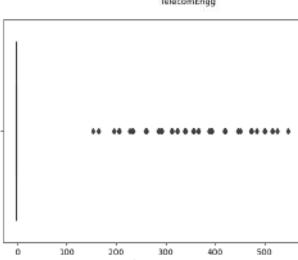
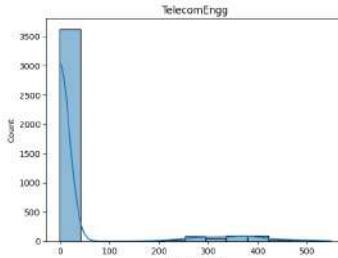
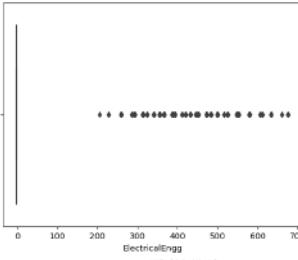
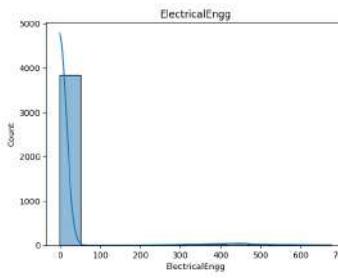
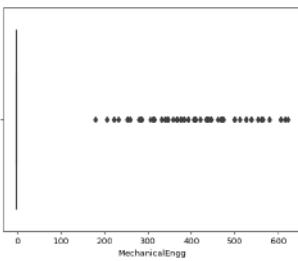
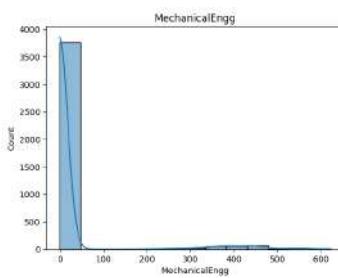
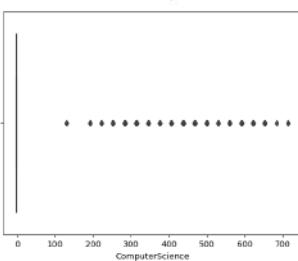
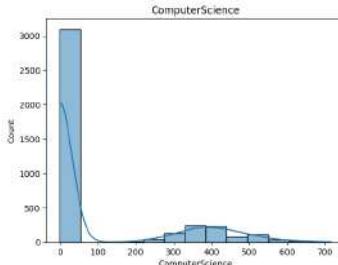
```
In [9]: #Boxplots and histograms for numerical cols
import seaborn as sns
for col in numerical_cols:
    sns.histplot(df[col], kde=True)
    plt.show()
    sns.boxplot(data=df, ecol)
    plt.show()
```

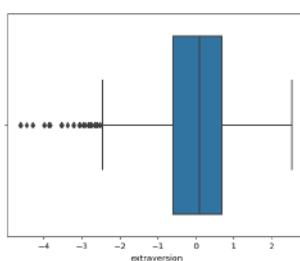
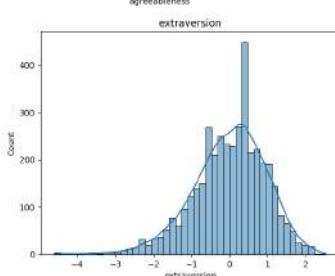
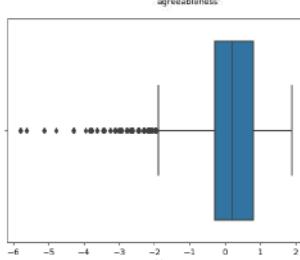
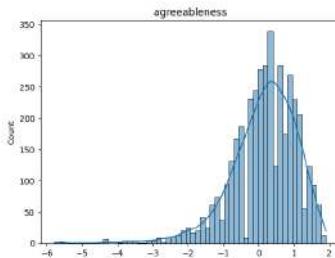
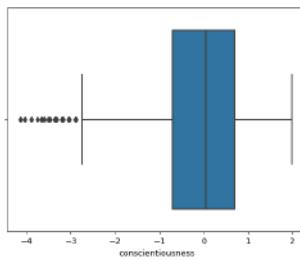
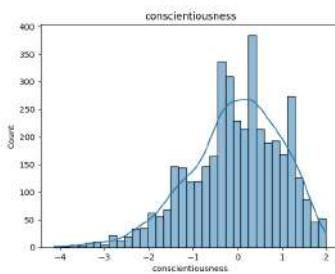
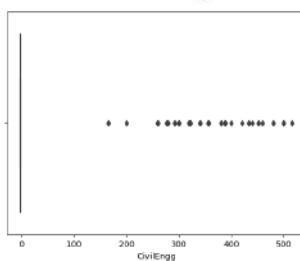
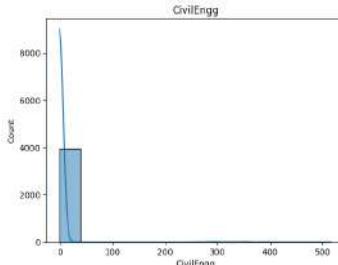


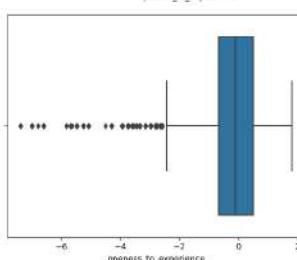
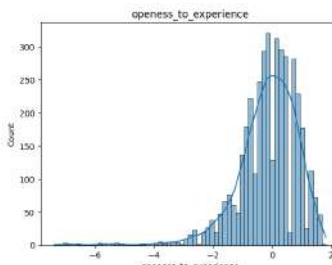
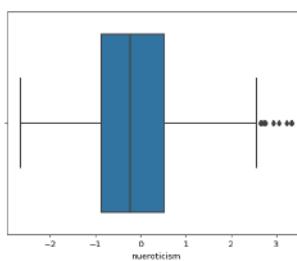
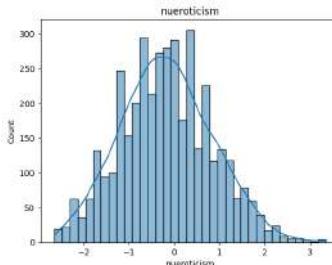










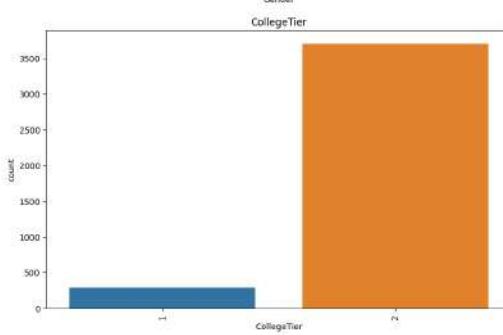
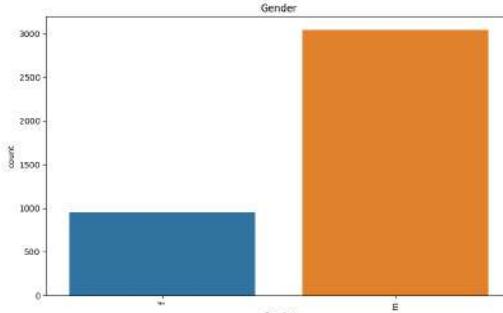


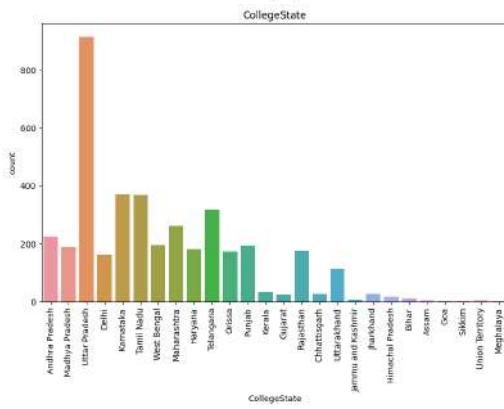
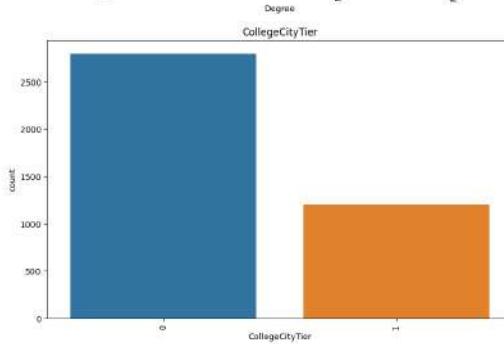
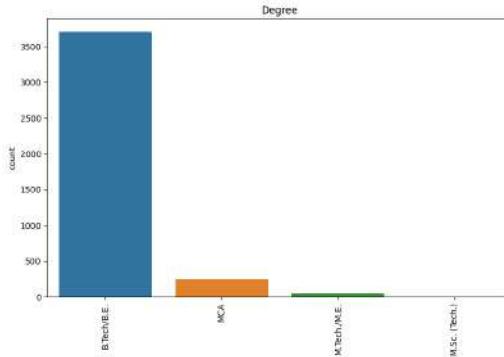
```
In [18]: #Removal of outliers
Q1=df.quantile(0.25)
Q3=df.quantile(0.75)
IQR=Q3-Q1
low_lim=Q1-(1.5*IQR)
up_lim=Q3+(1.5*IQR)
df_clean=df[~((df<low_lim) | (df>up_lim)).any(axis=1)]
cnt_outliers=df.shape[0]-df_clean.shape[0]
print("Number of outliers removed: ",cnt_outliers)

Number of outliers removed: 2255
C:\Users\deell\AppData\Local\Temp\ipykernel_18092\2112488016.py:7: FutureWarning: Automatic reindexing on DataFrame vs Series comparisons is deprecated and will raise ValueError in a future version. Do 'left, right = left.align(right, axis=1, copy=False)' before e.g. 'left == right'
df_clean=df[~((df<low_lim) | (df>up_lim)).any(axis=1)]
```

```
In [19]: categorical_cols=df[['Gender','CollegeTier','Degree','CollegeCityTier','CollegeState']]
```

```
In [20]: #Countplots for categorical columns
for col in categorical_cols:
    plt.figure(figsize=(10,6))
    plt.xticks(rotation=90)
    plt.title(col)
    sns.countplot(data=df,x=col)
    plt.show()
```

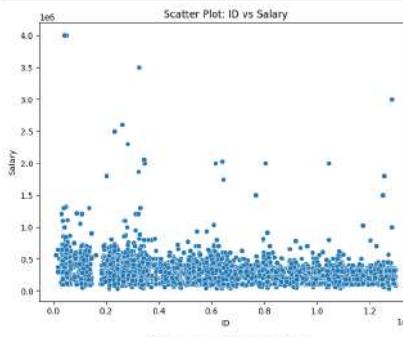


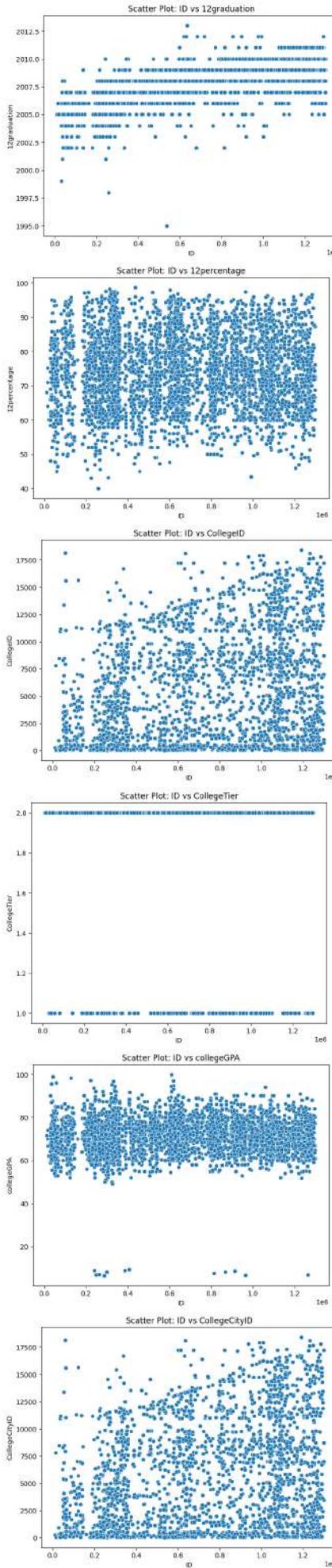


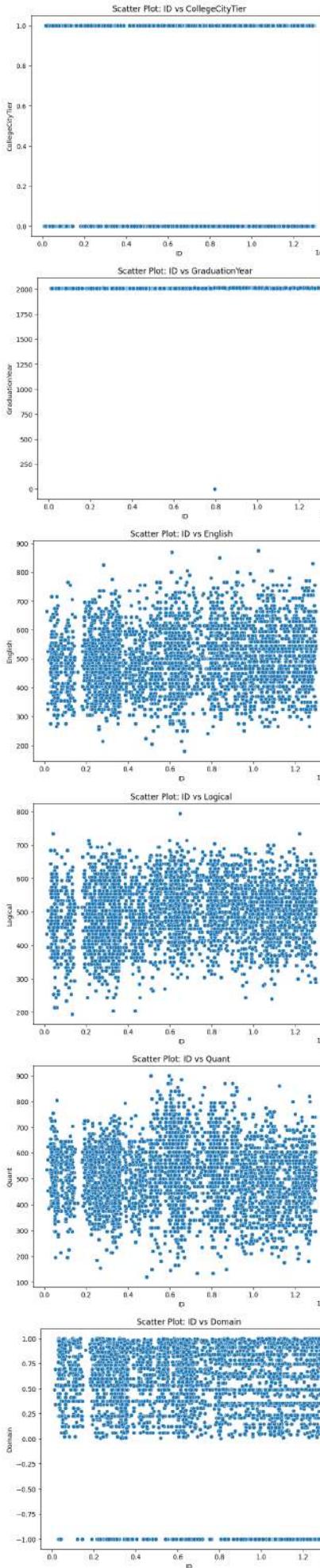
BIVARIATE ANALYSIS

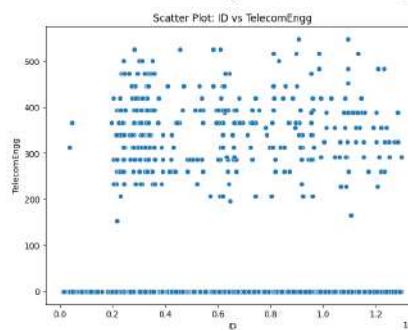
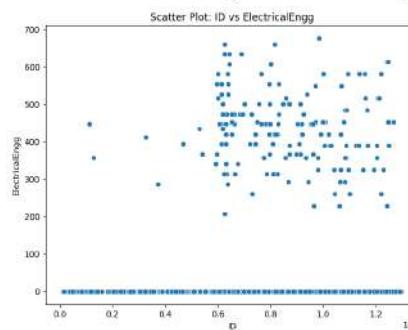
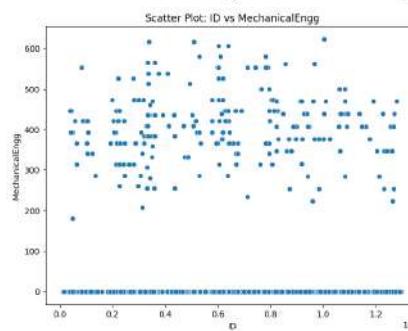
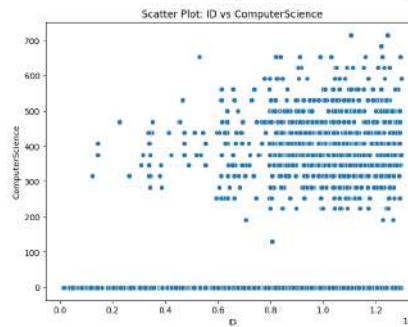
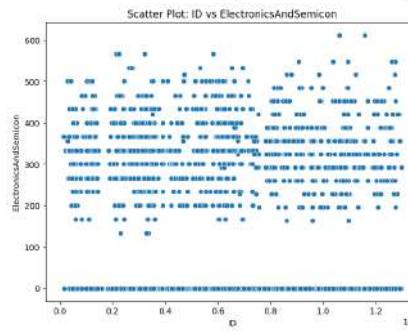
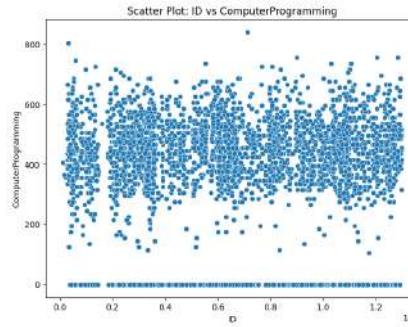
Numerical vs Numerical

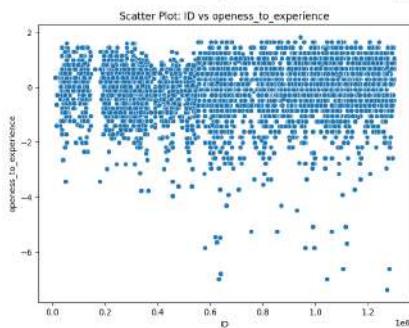
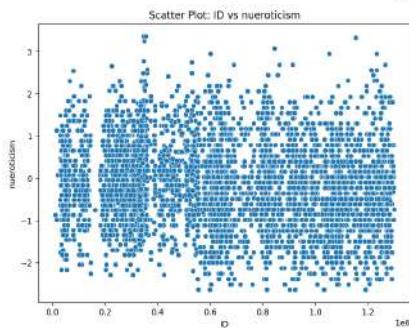
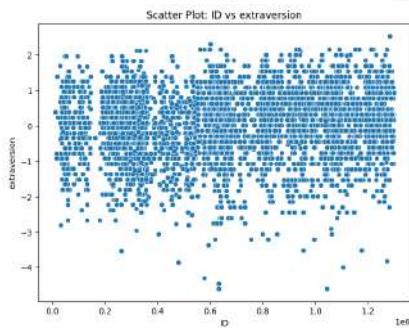
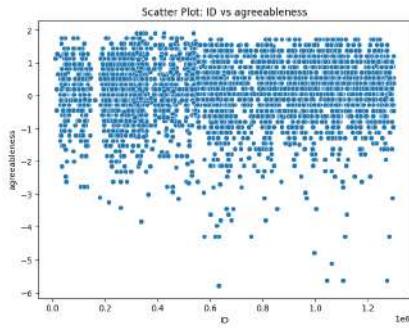
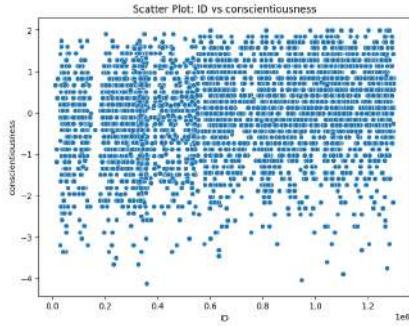
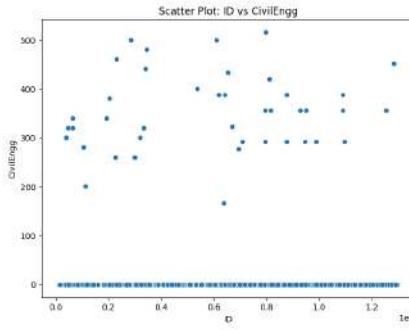
```
In [15]: #Scatter plot
for col1 in numerical_cols:
    for col2 in numerical_cols:
        if col1!=col2:
            plt.figure(figsize=(8,5))
            plt.scatterplot(data=df,x=col1,y=col2)
            plt.title(f"Scatter Plot: {col1} vs {col2}")
            plt.xlabel(col1)
            plt.ylabel(col2)
            plt.show()
```

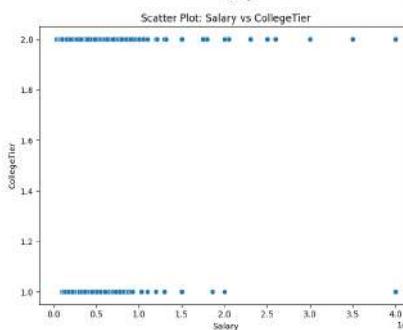
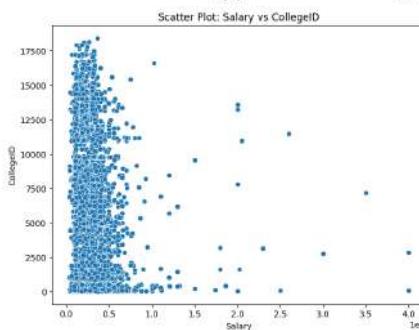
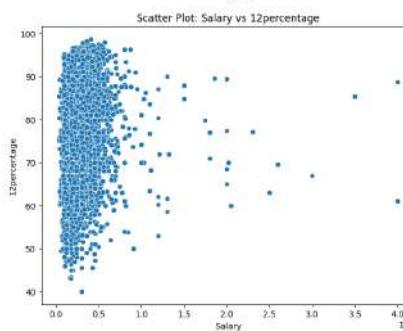
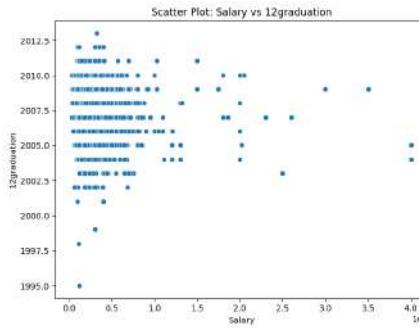
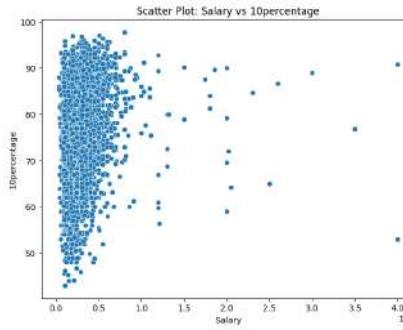
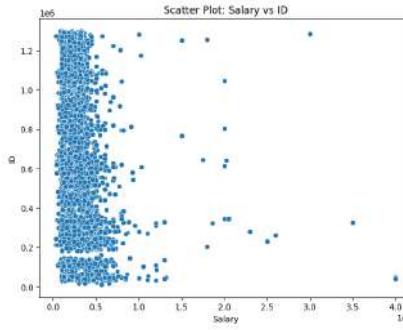


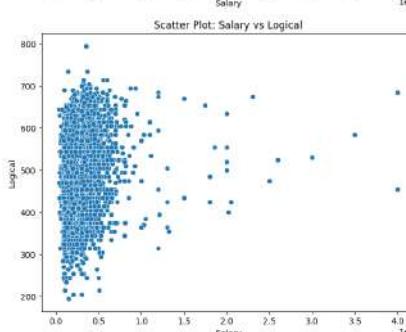
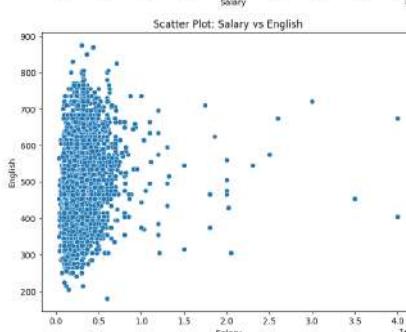
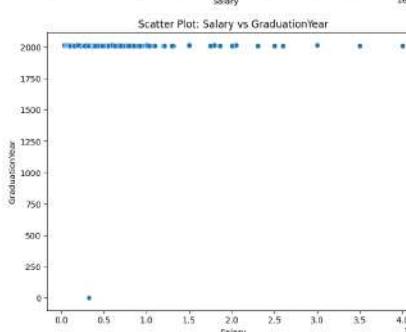
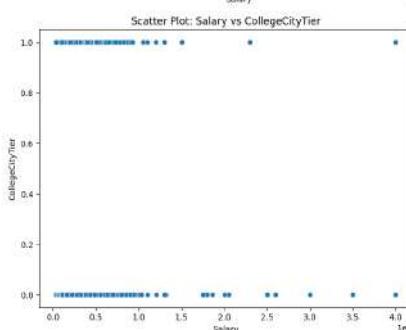
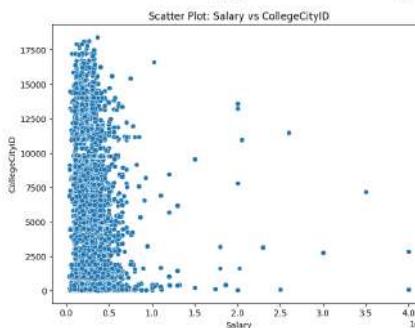
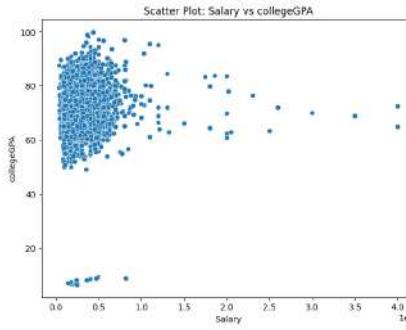


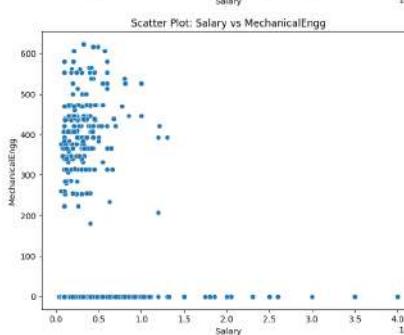
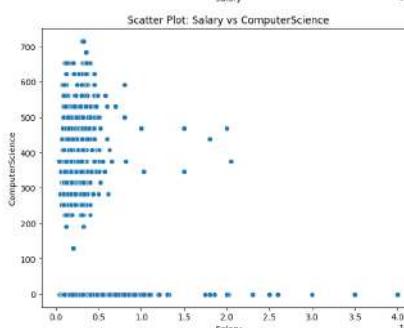
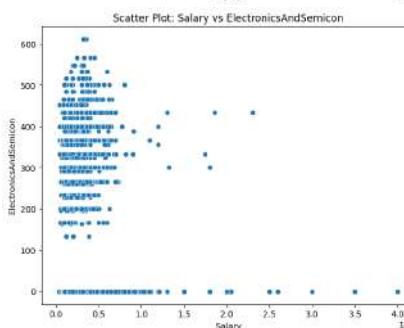
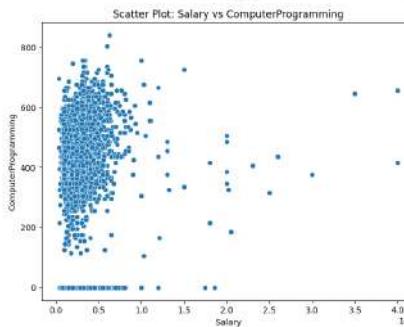
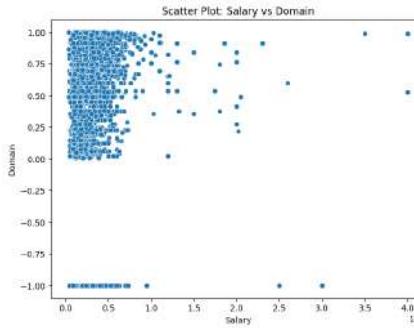
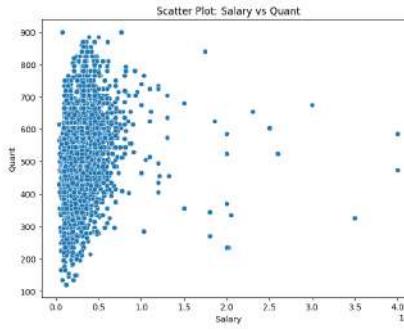


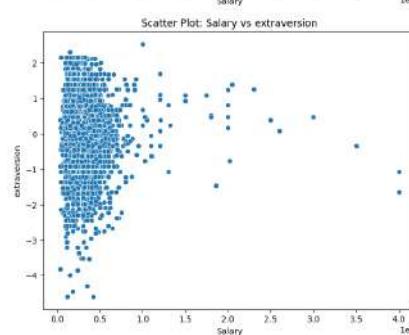
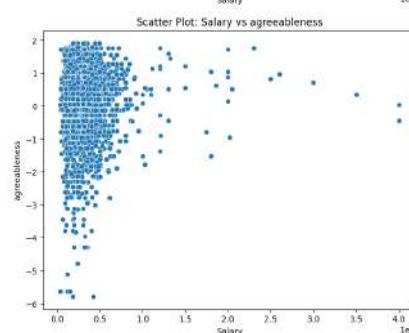
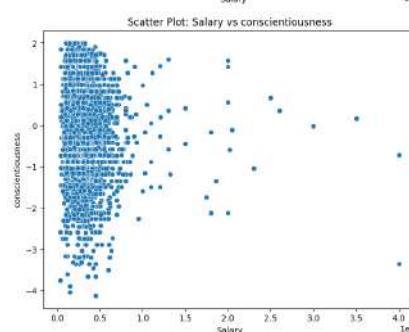
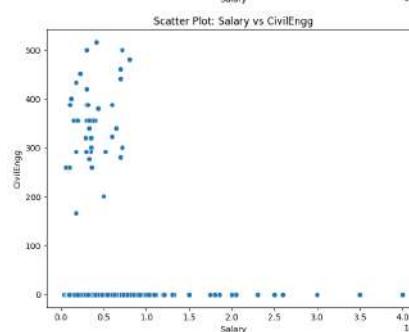
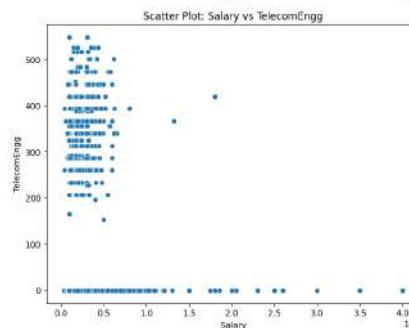
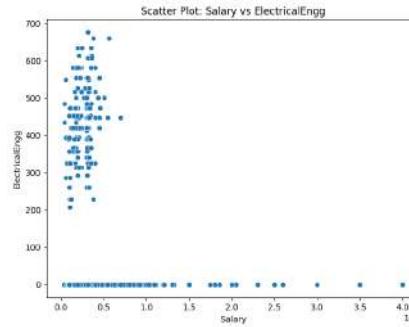


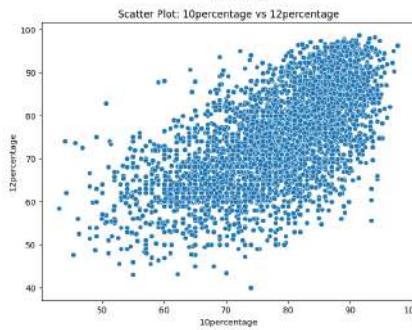
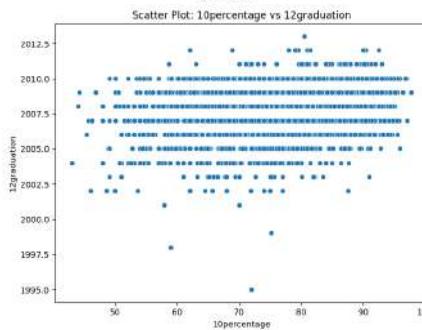
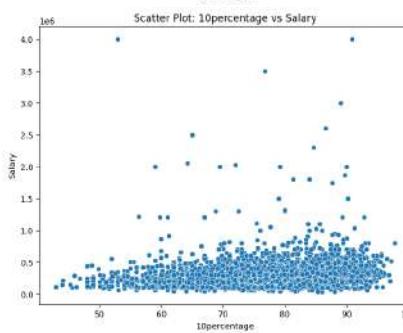
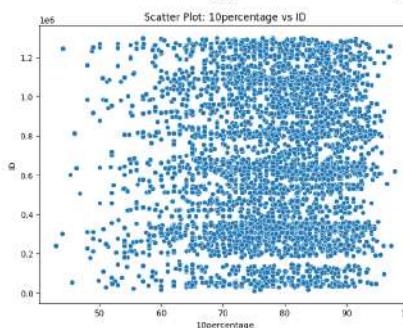
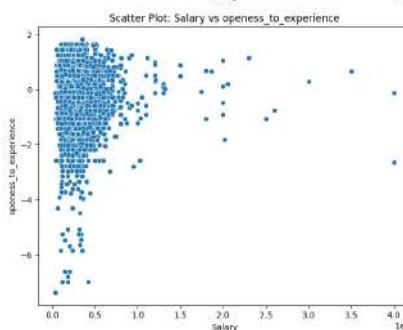
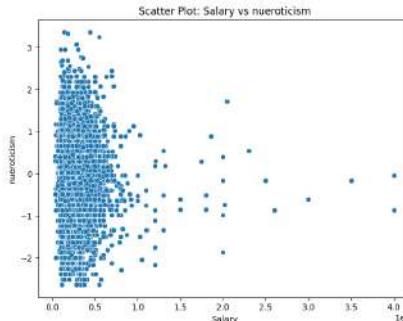


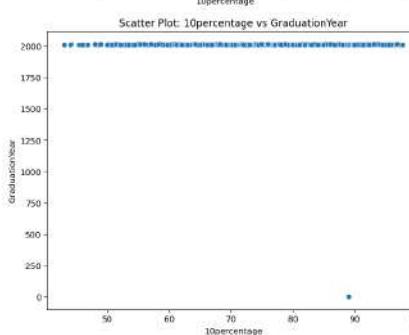
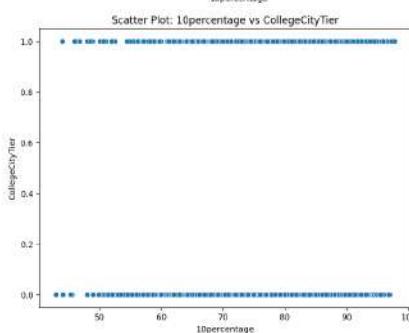
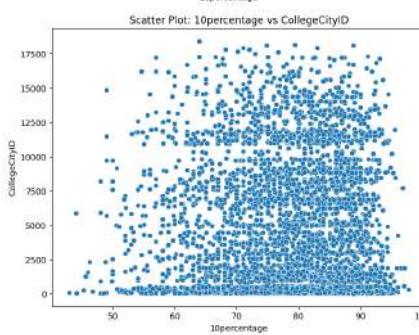
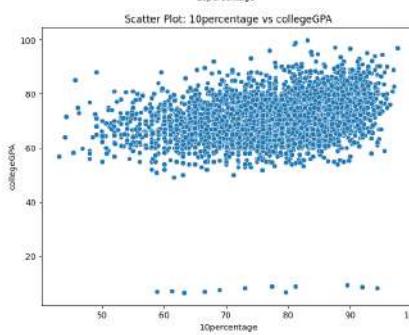
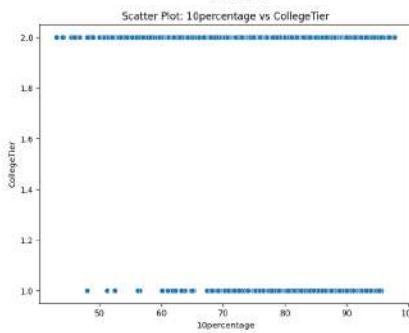
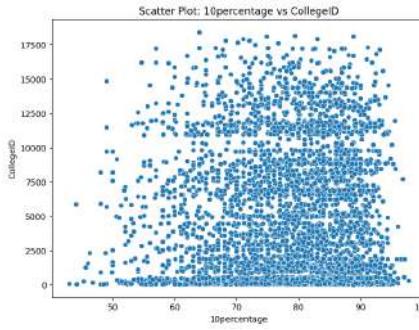


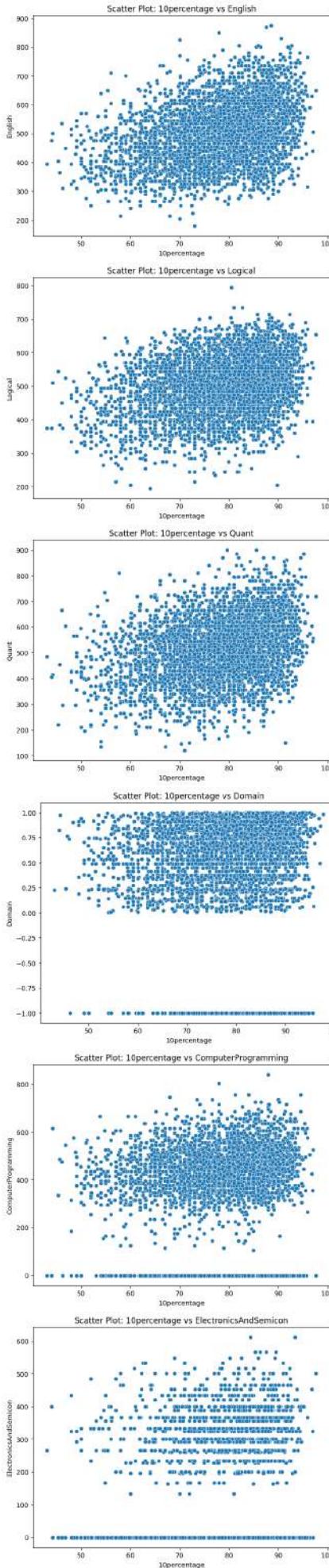


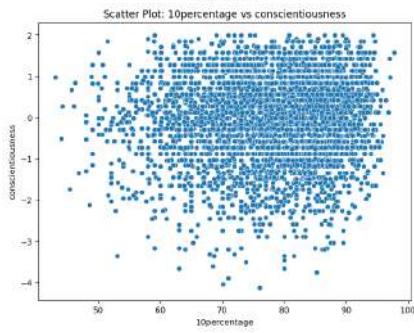
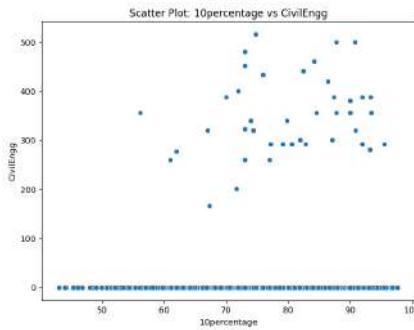
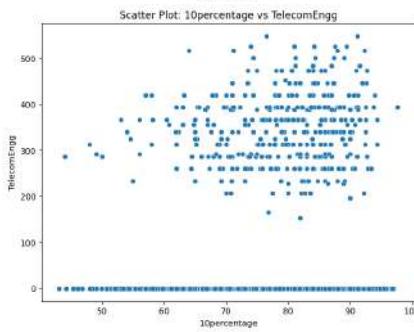
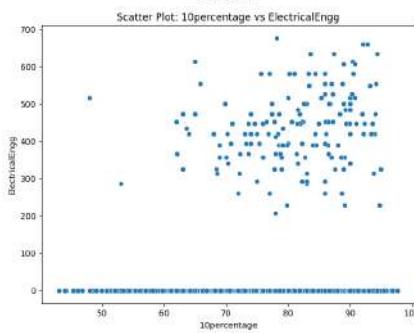
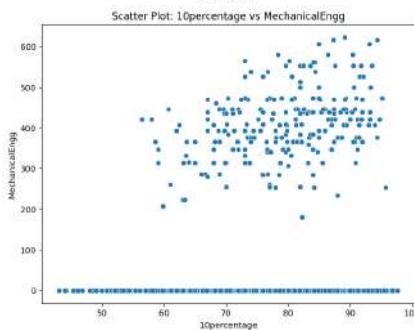
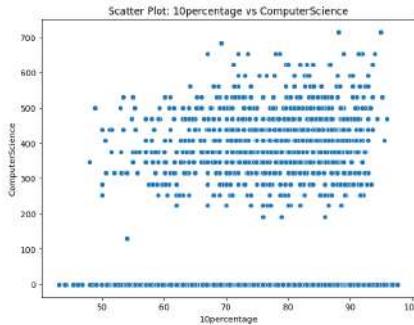


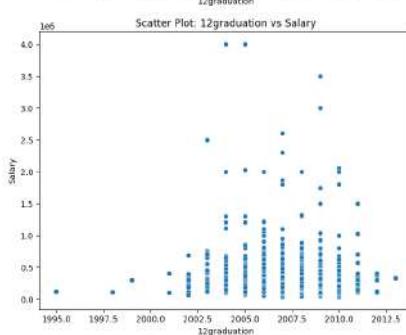
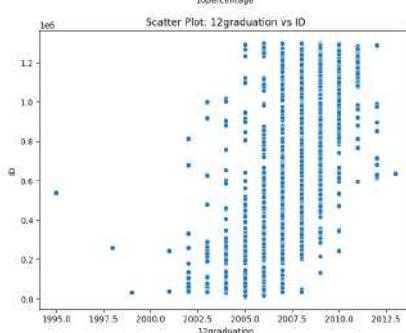
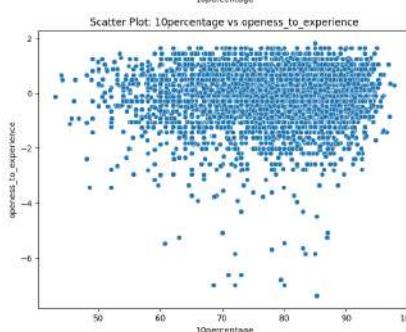
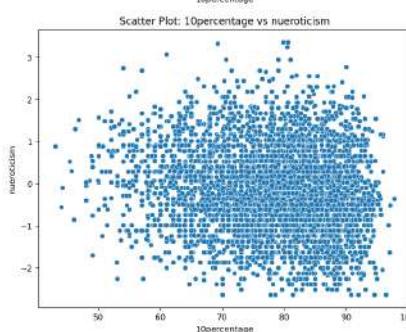
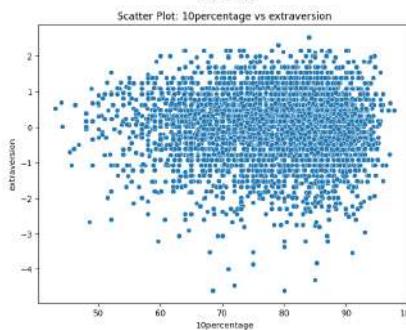
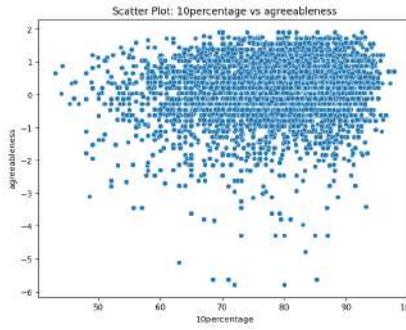


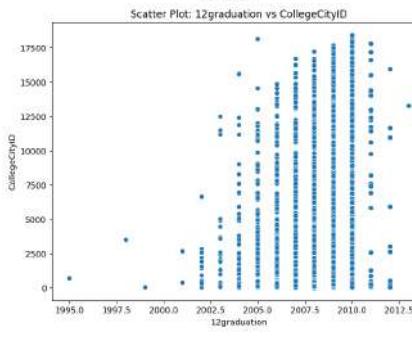
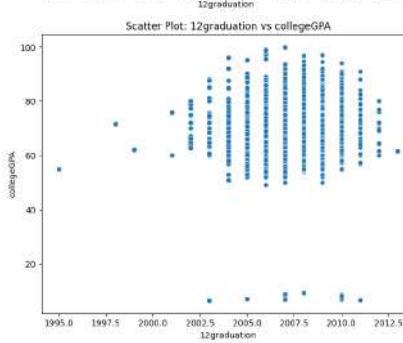
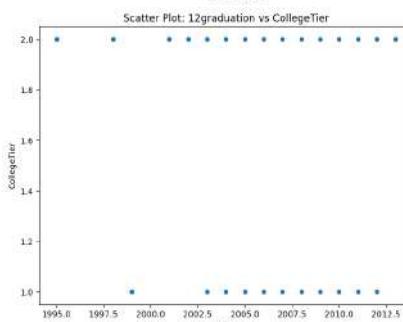
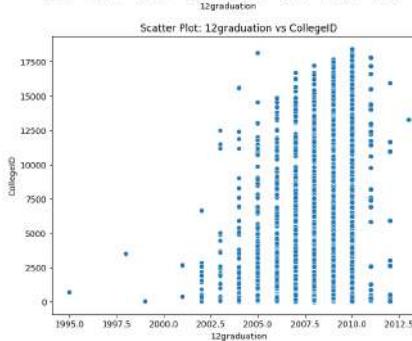
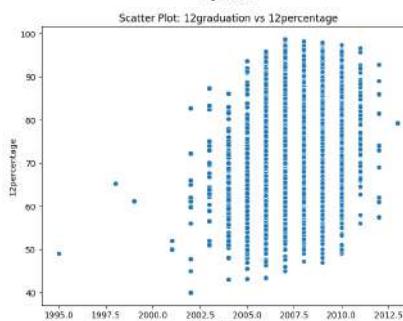
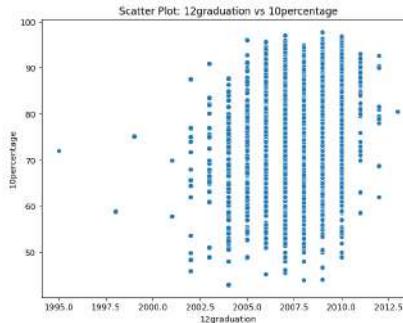


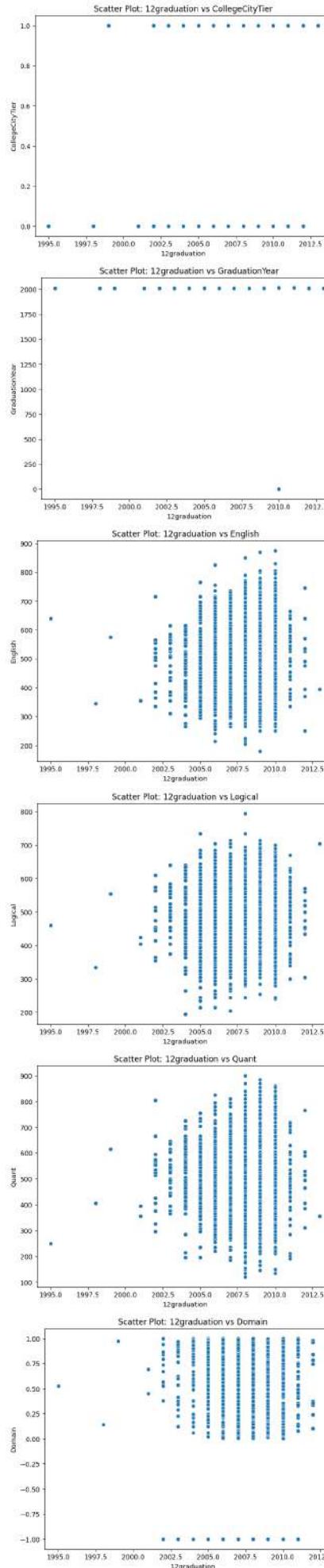


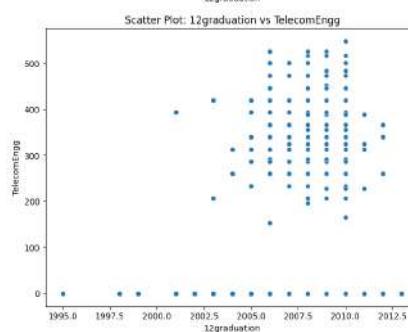
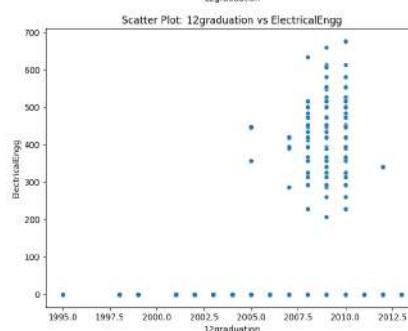
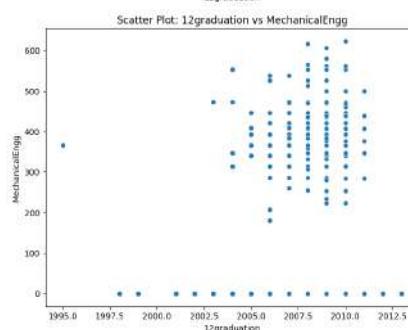
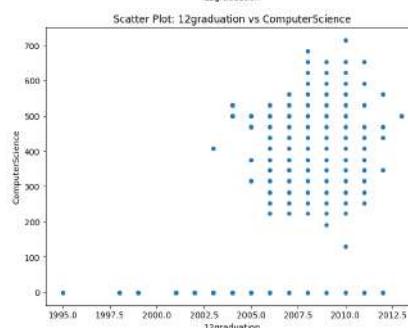
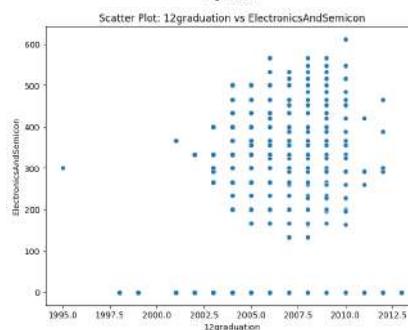
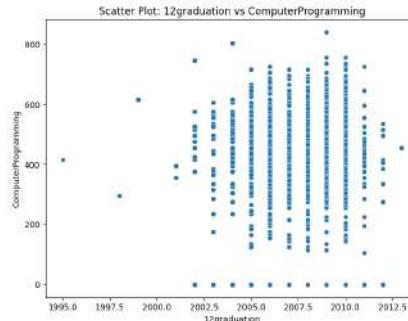


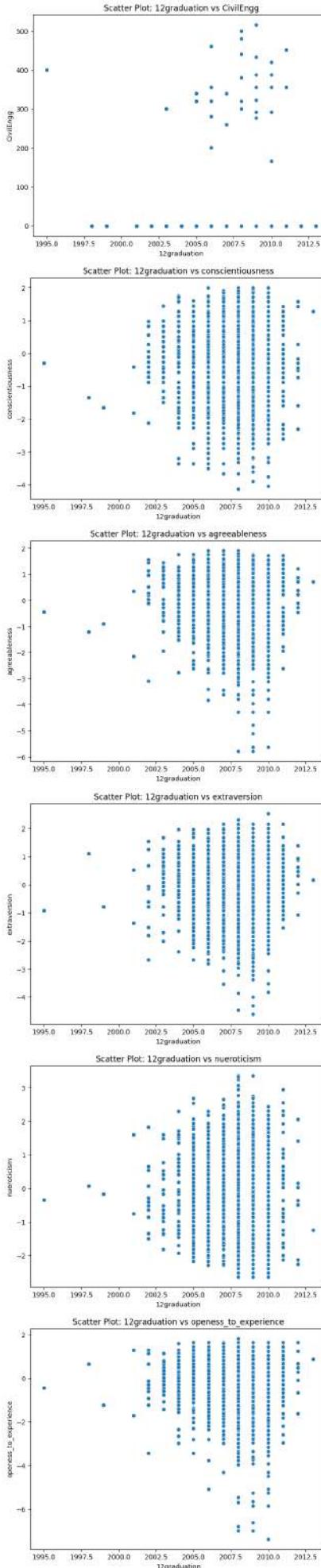


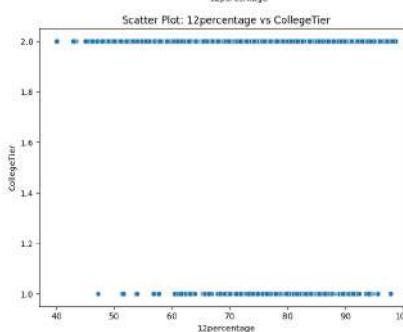
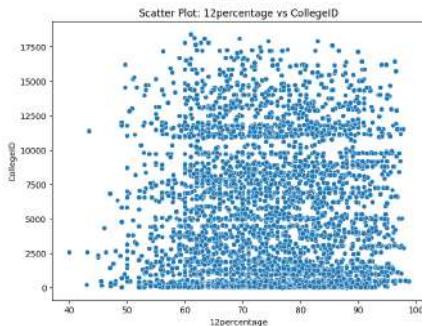
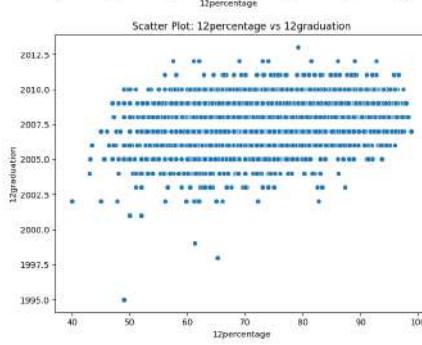
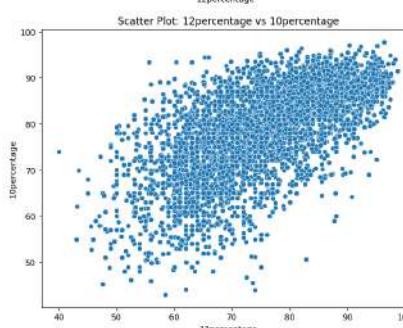
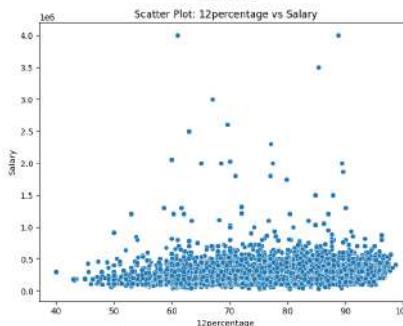
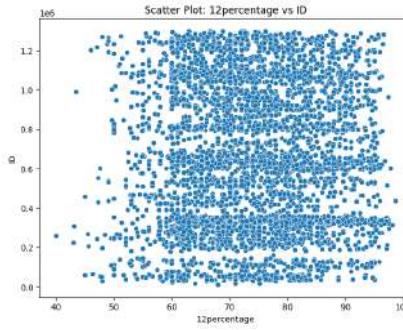


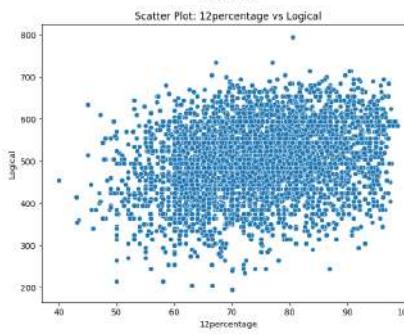
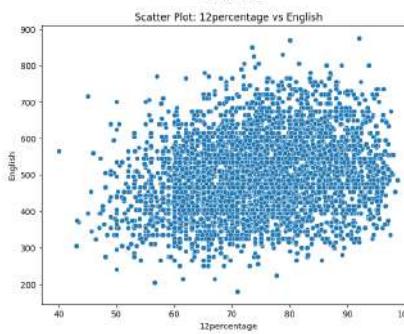
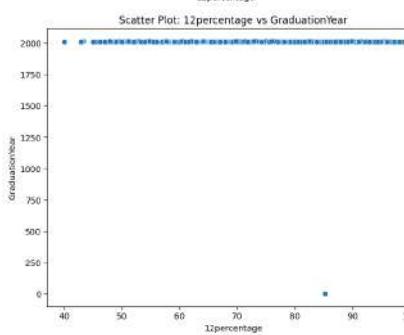
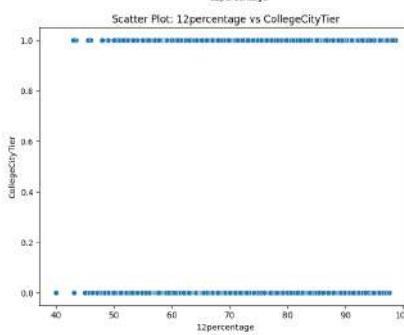
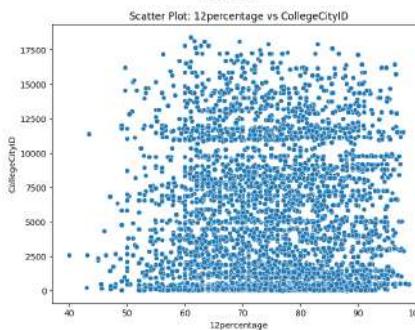
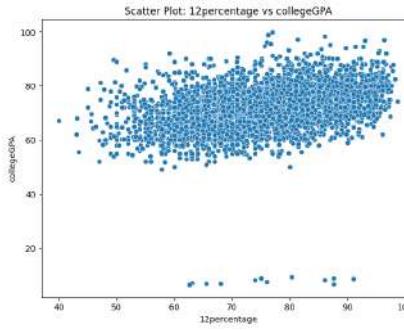


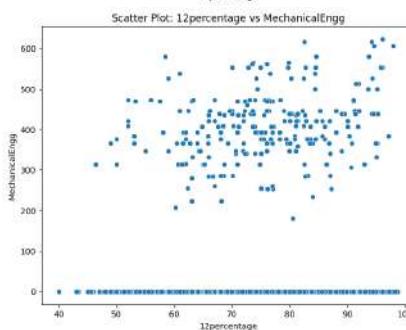
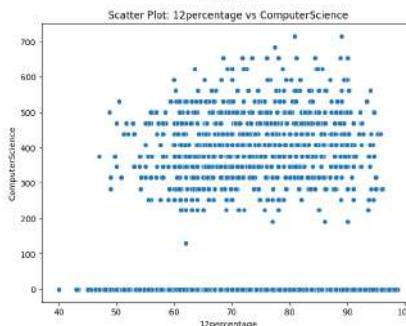
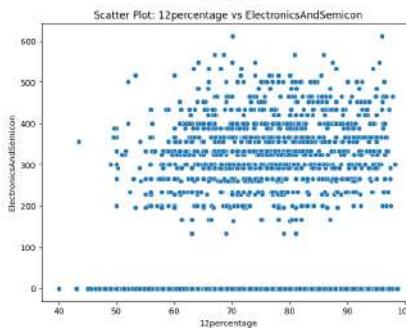
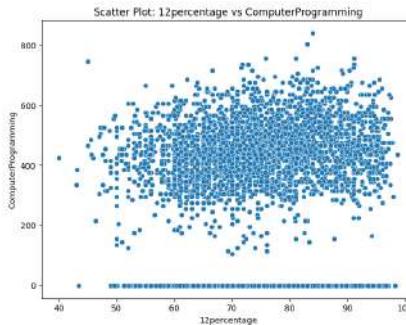
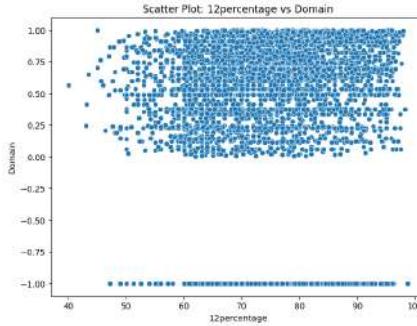
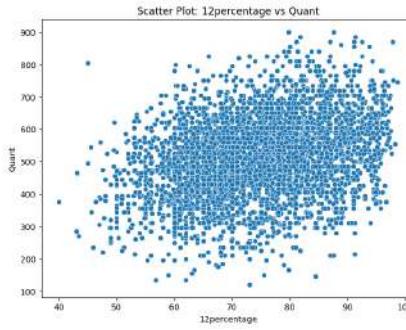


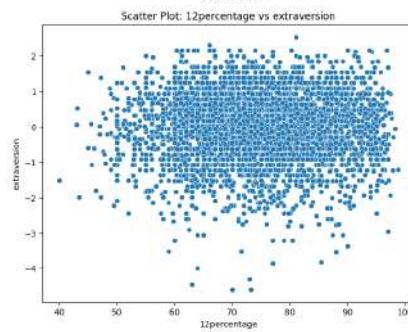
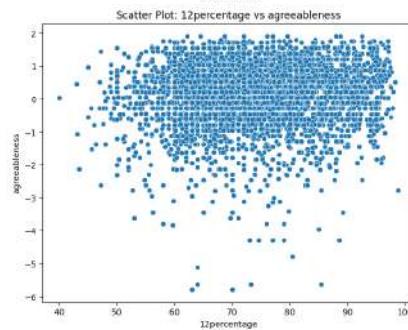
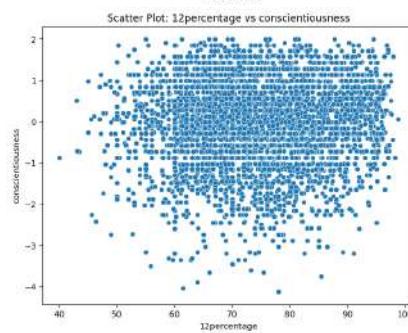
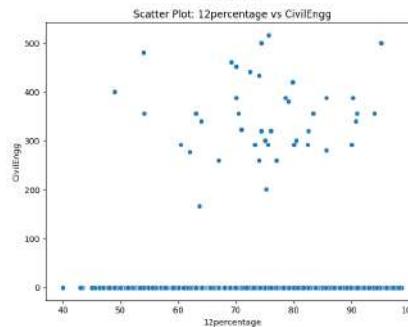
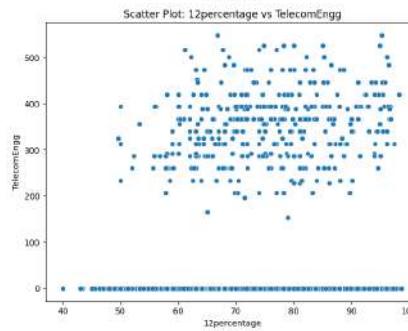
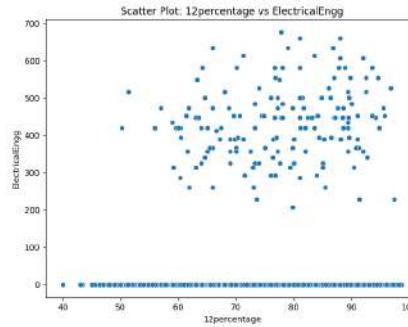


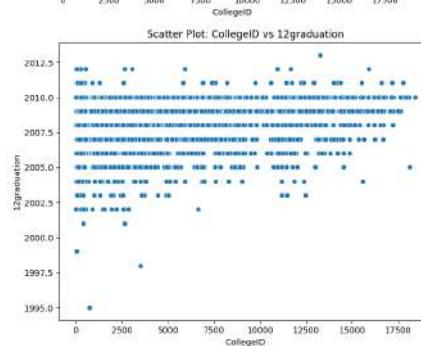
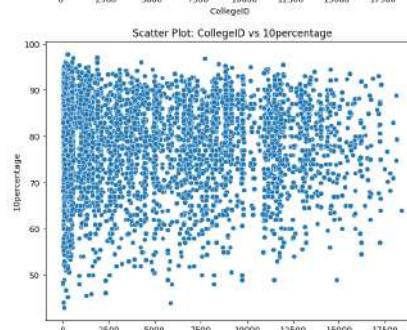
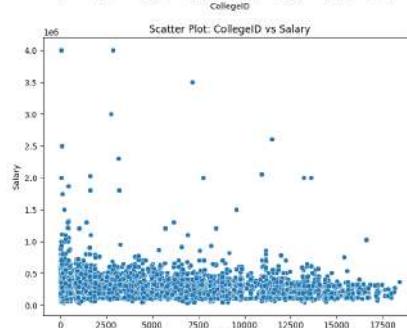
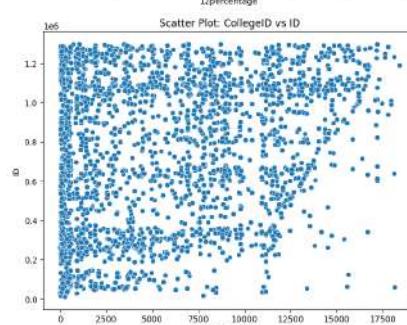
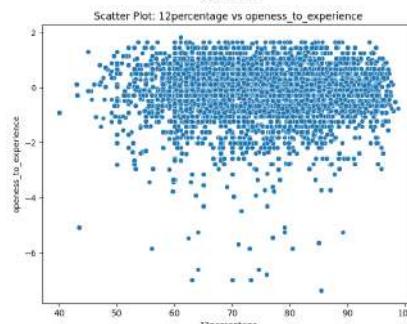
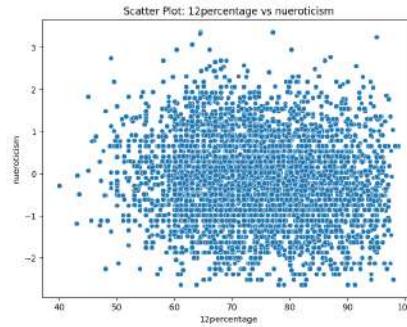


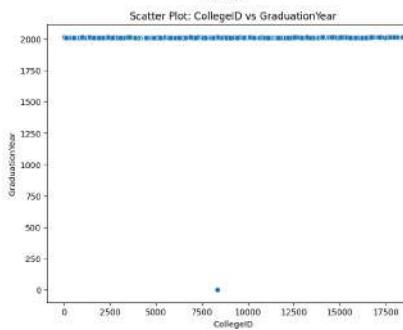
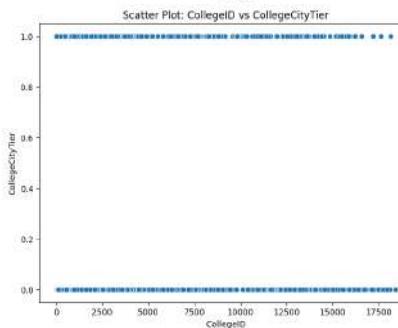
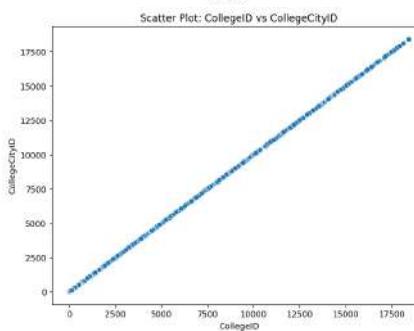
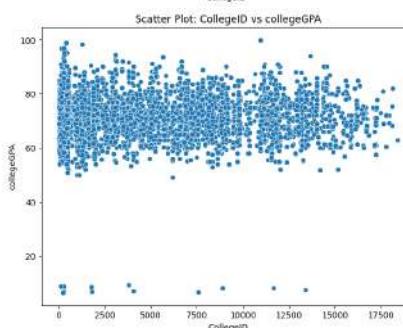
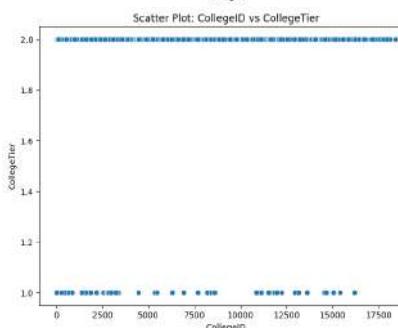
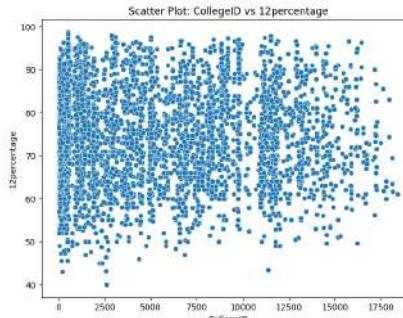


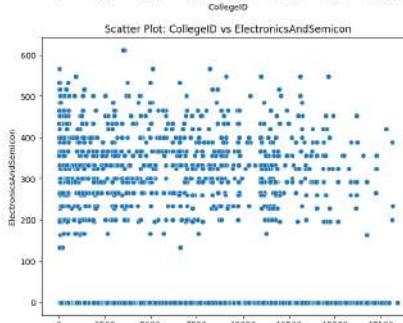
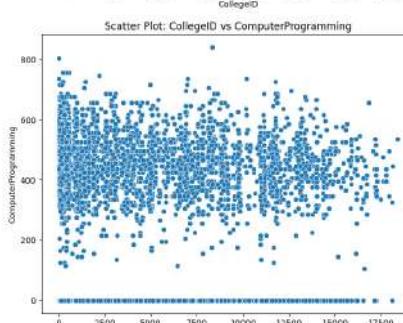
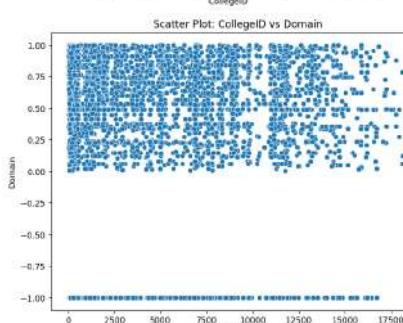
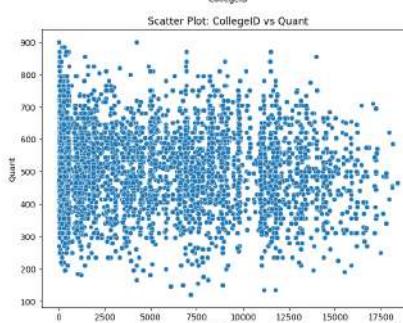
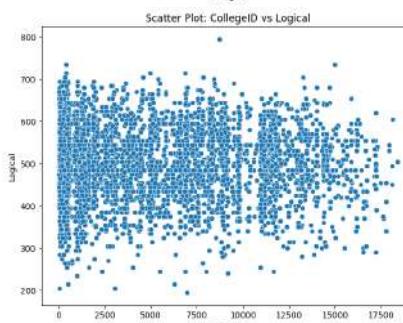
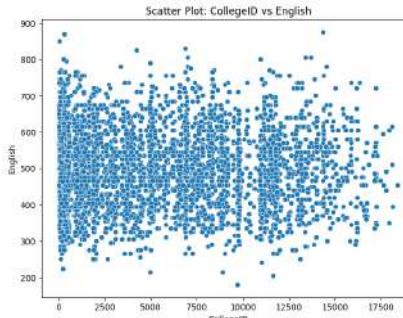


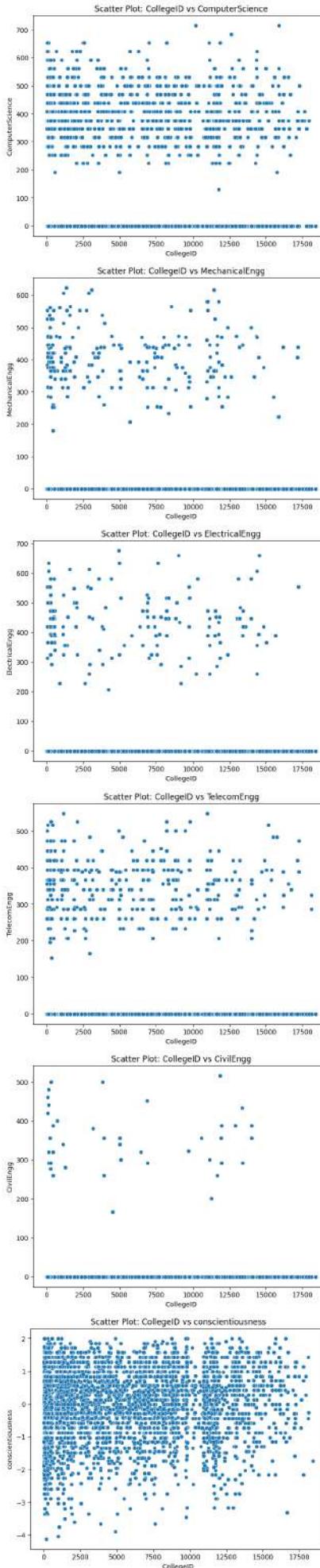


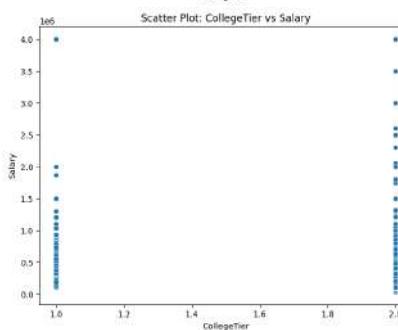
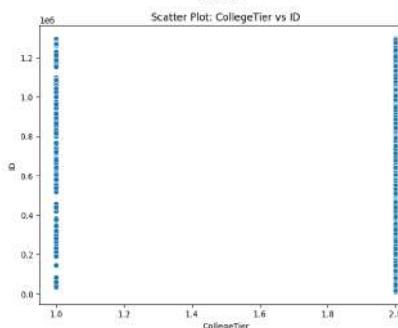
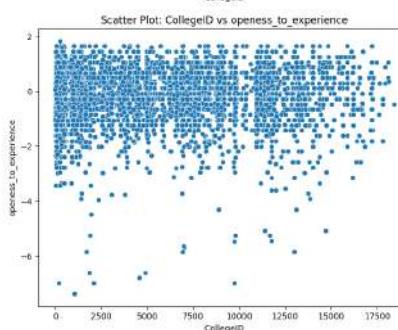
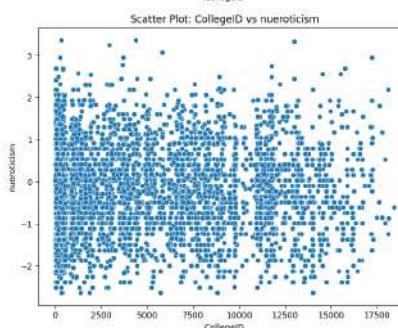
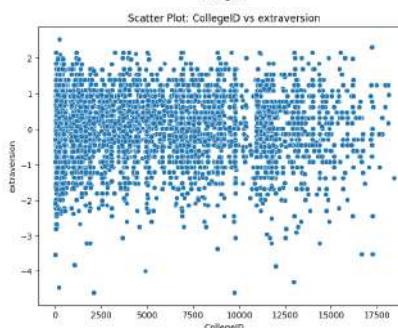
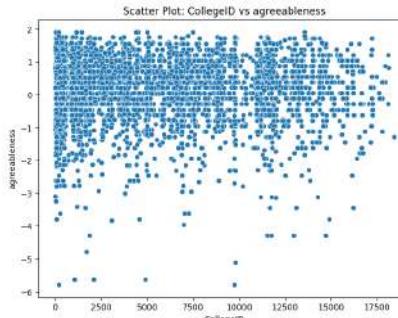


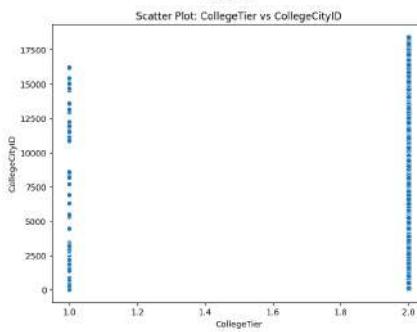
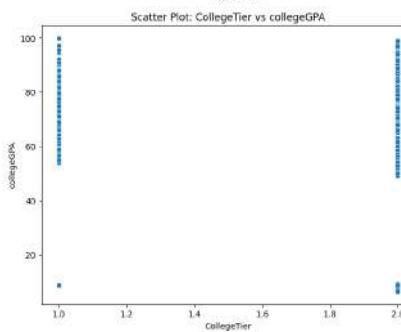
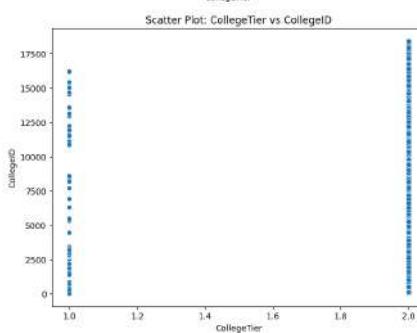
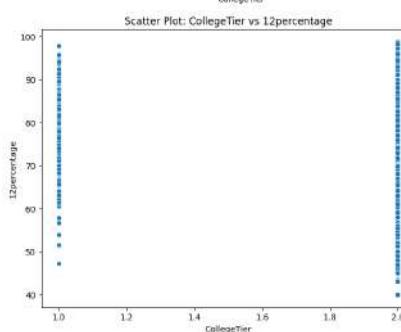
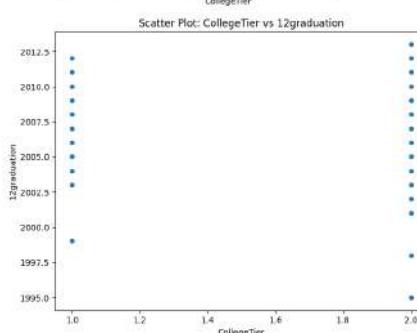
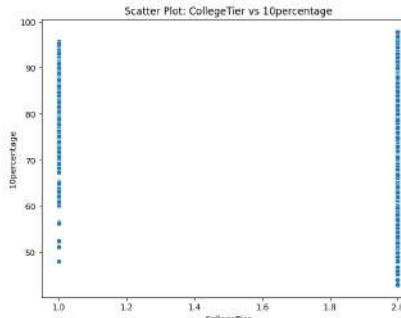


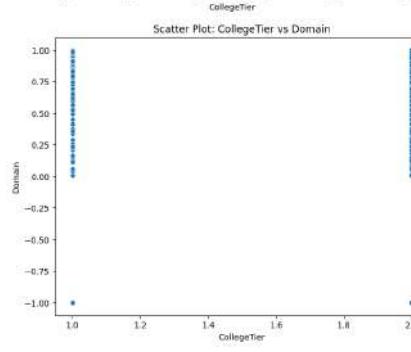
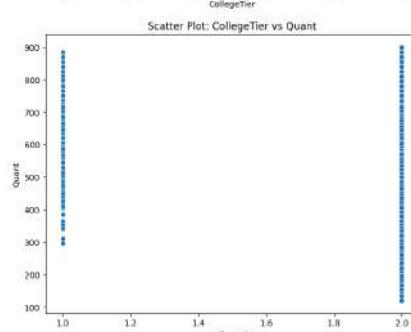
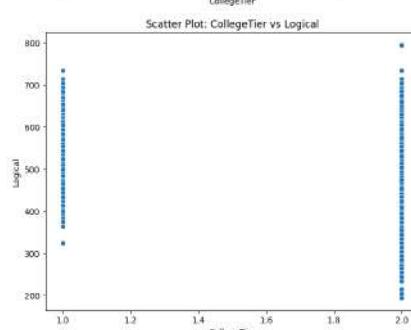
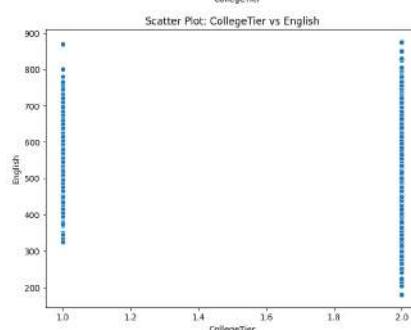
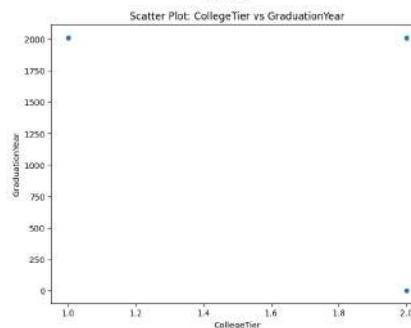
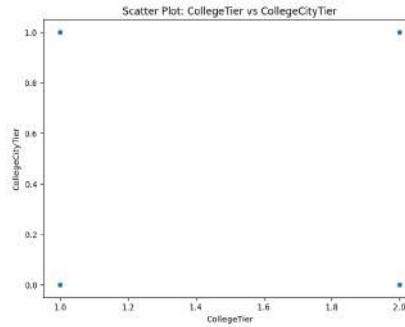


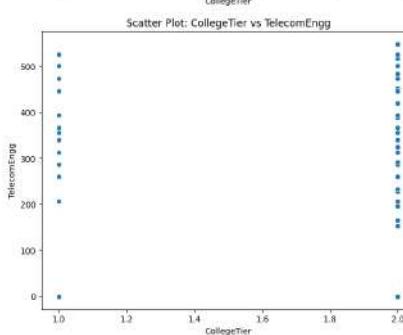
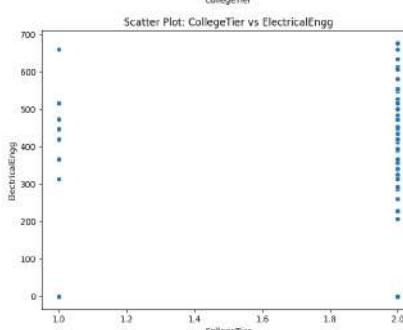
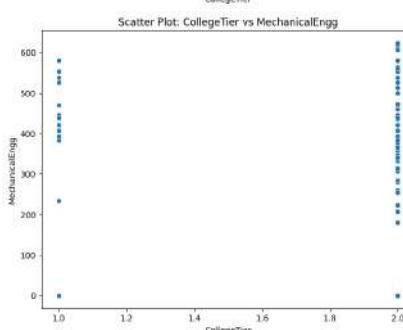
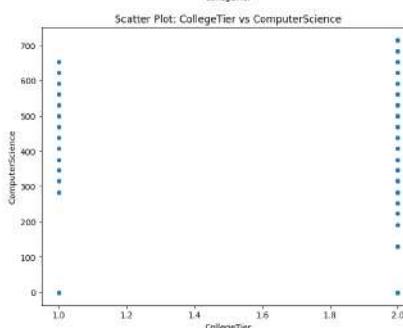
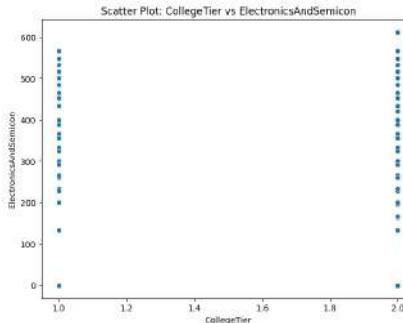
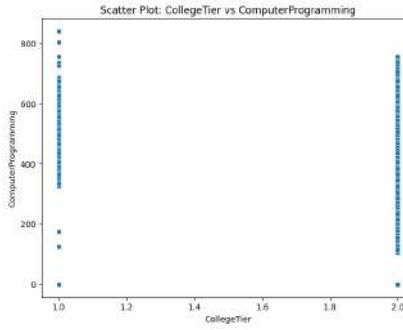


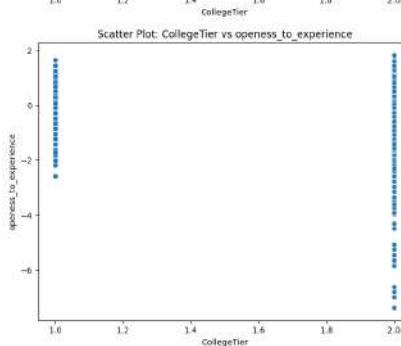
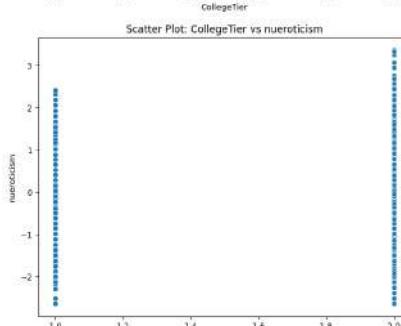
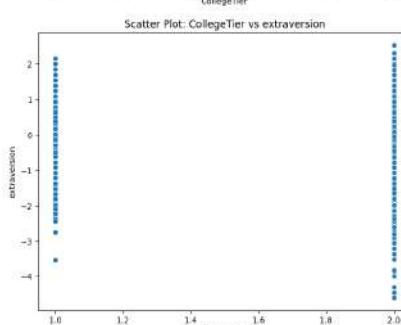
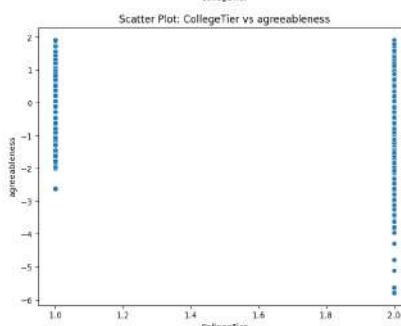
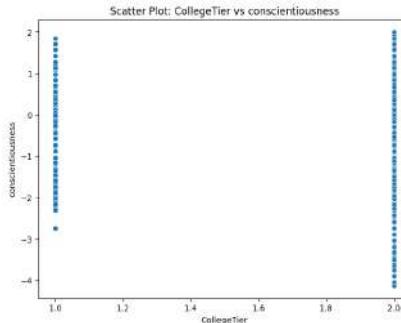
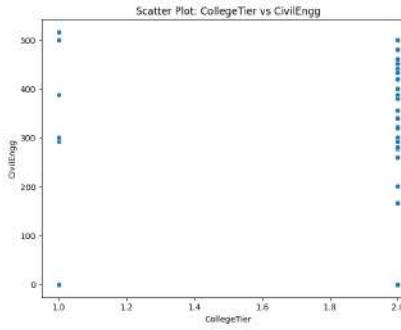


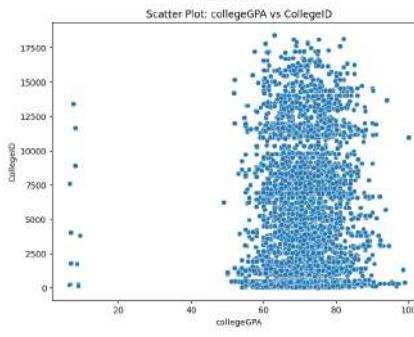
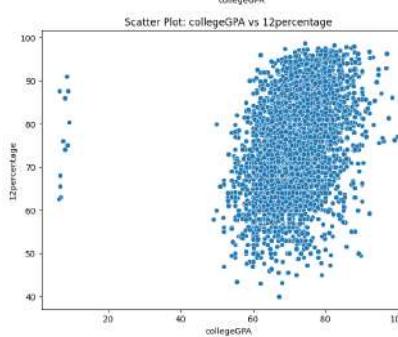
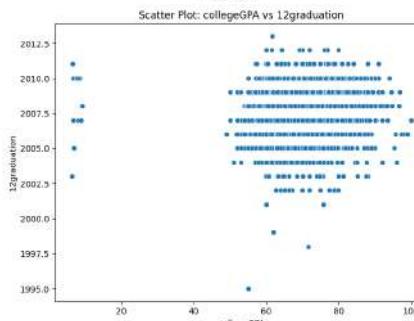
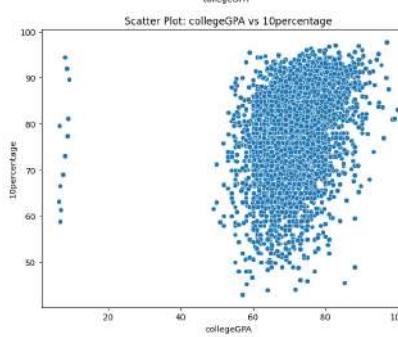
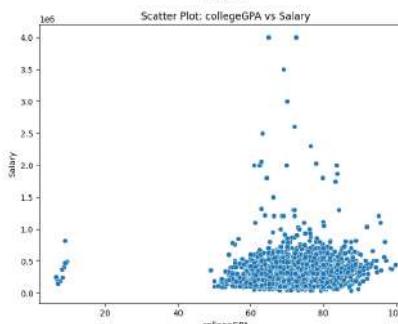
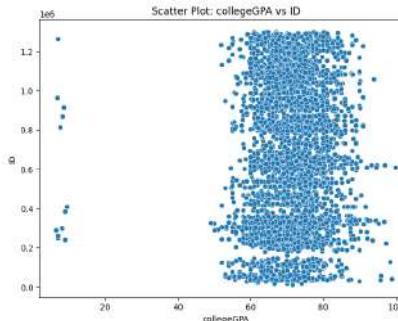


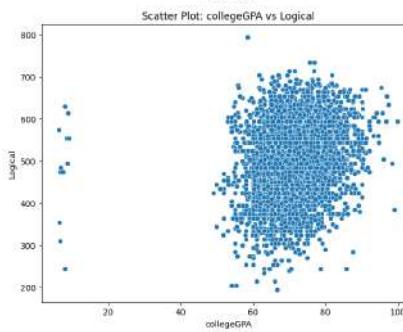
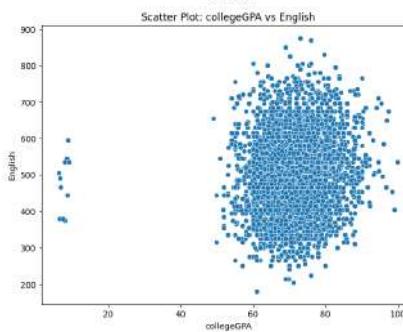
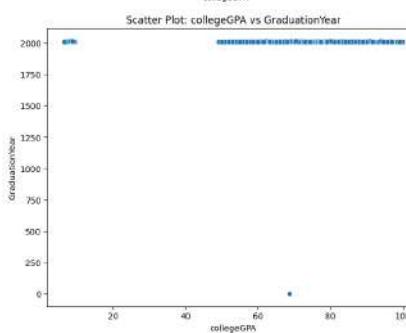
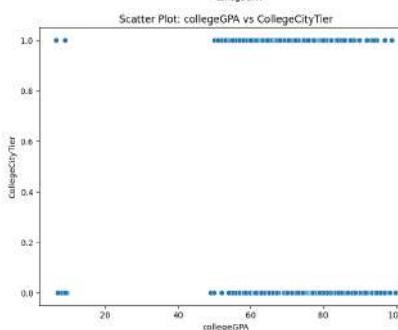
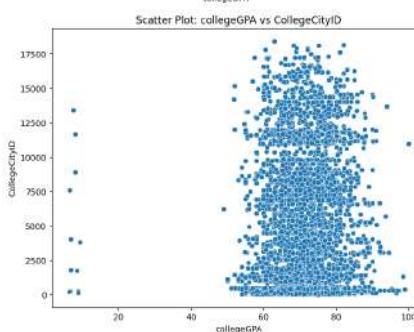
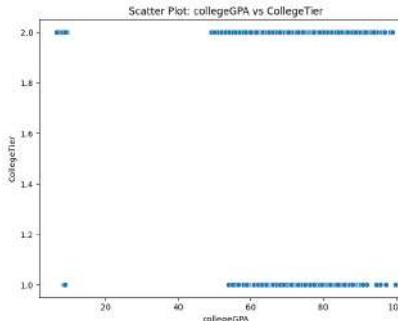


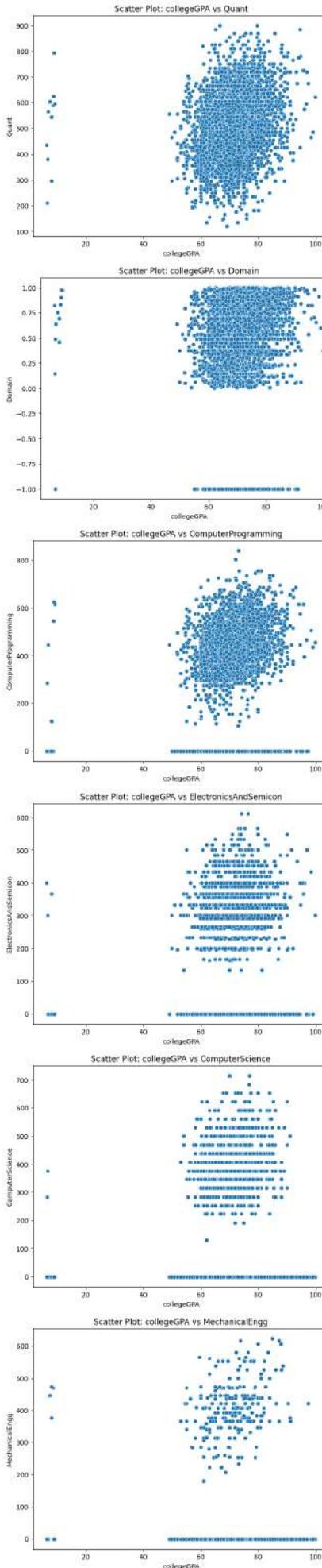


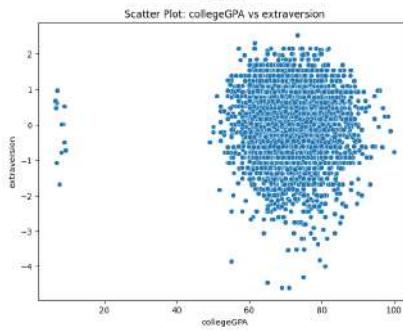
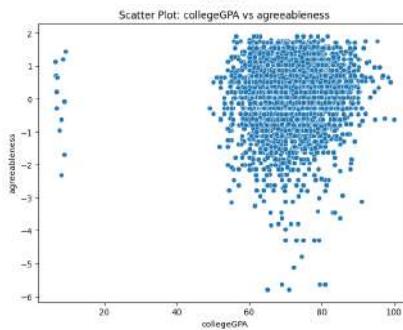
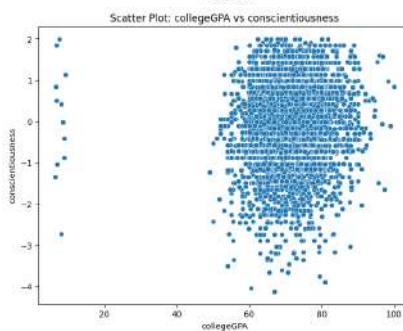
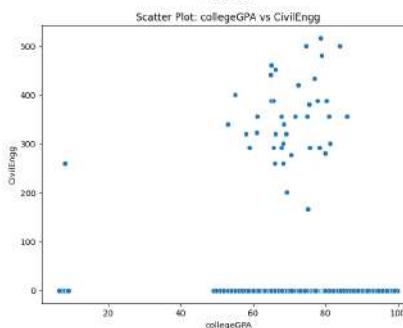
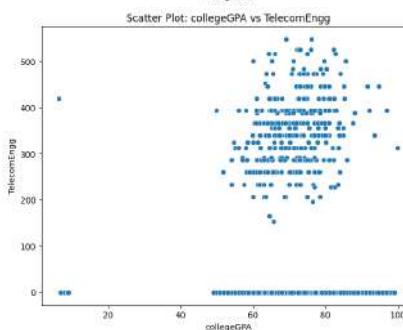
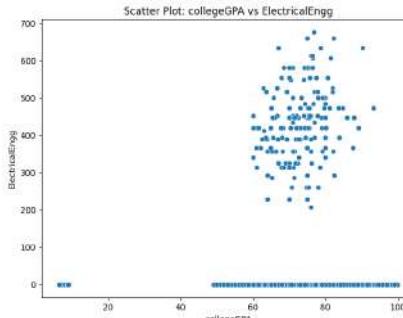


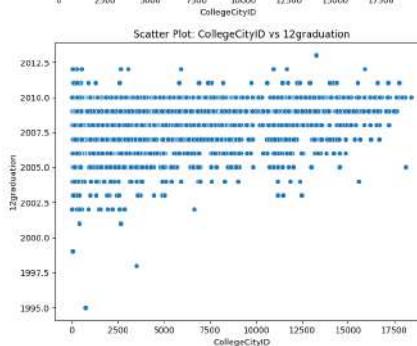
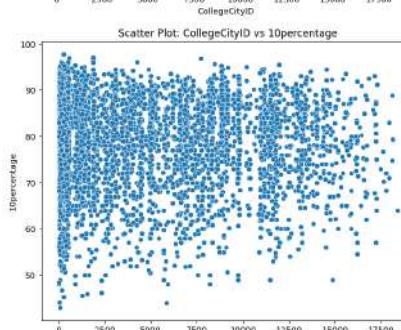
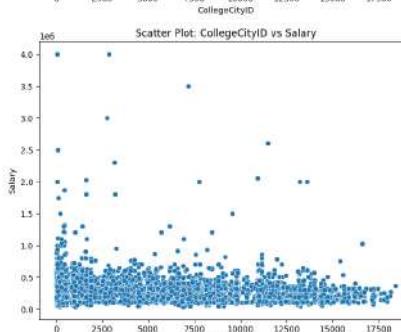
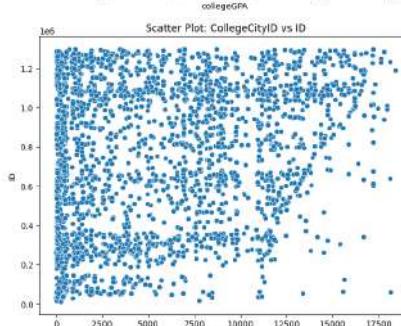
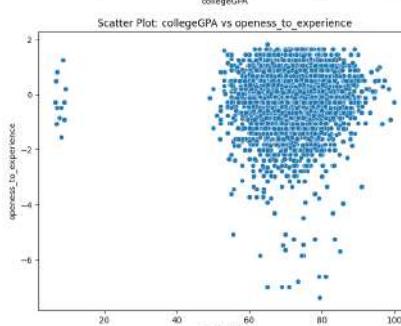
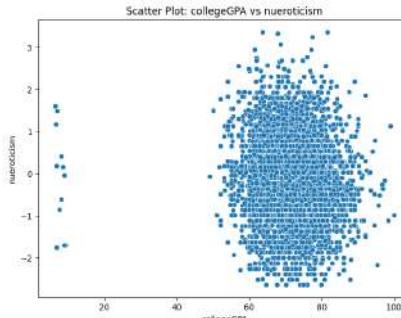


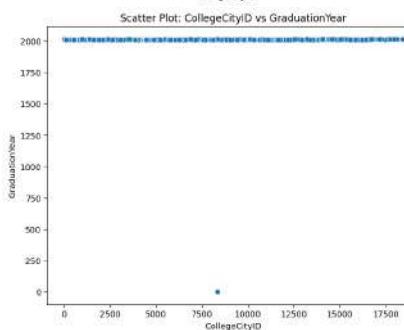
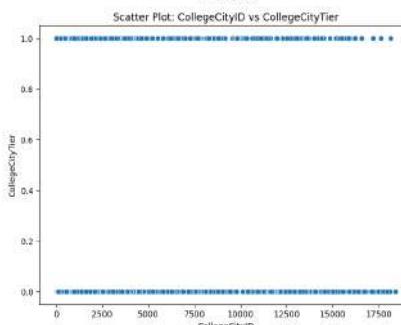
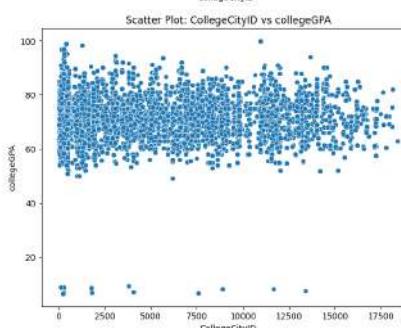
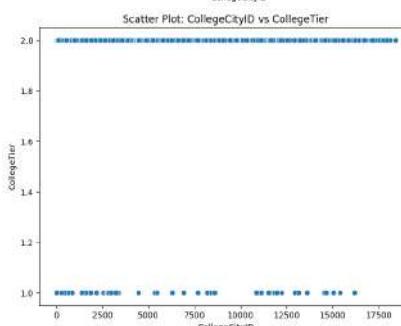
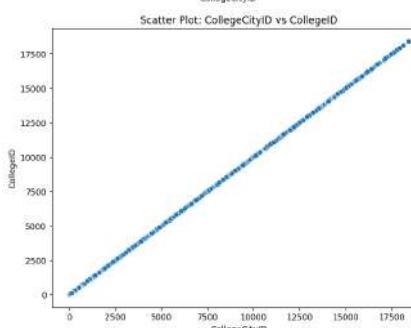
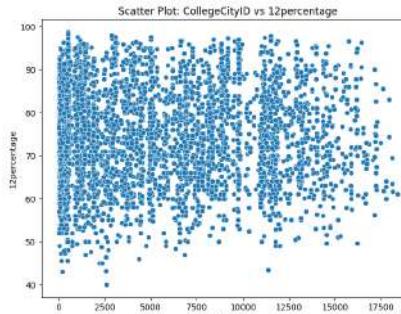


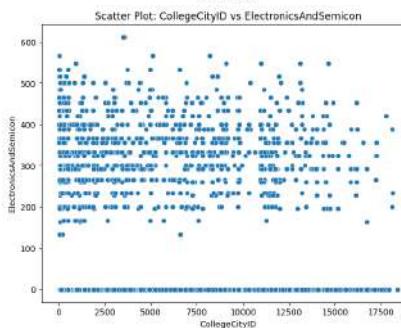
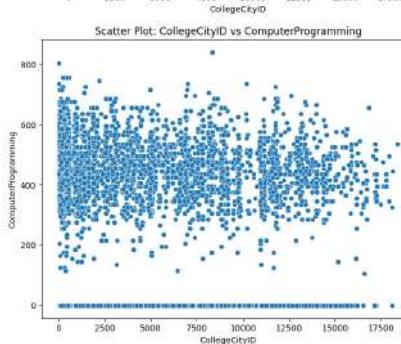
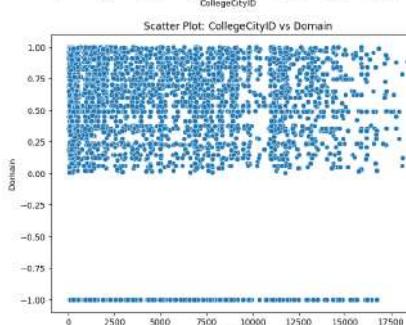
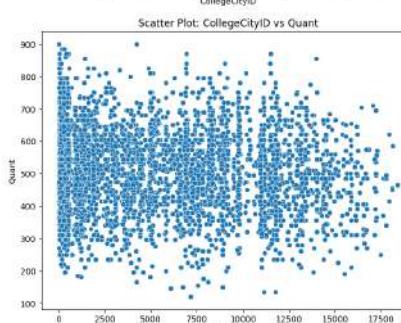
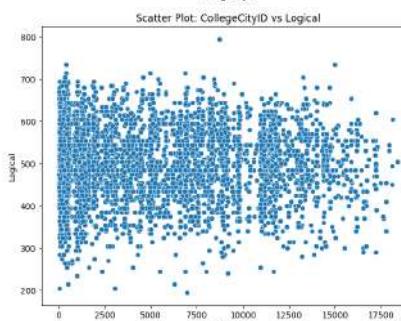
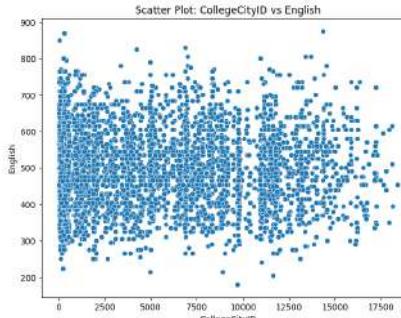


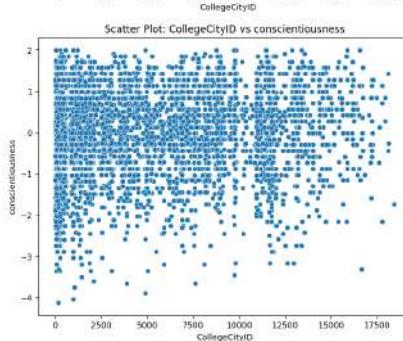
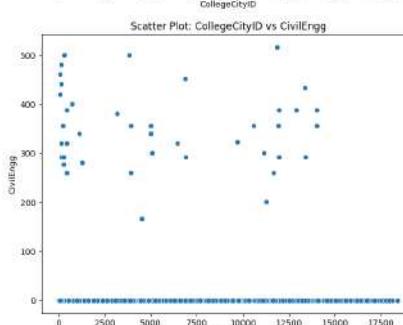
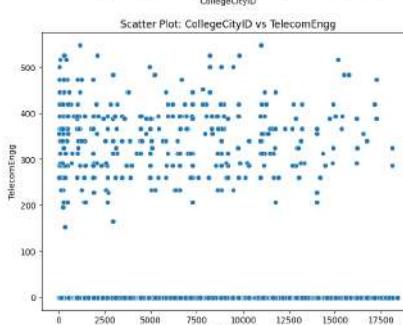
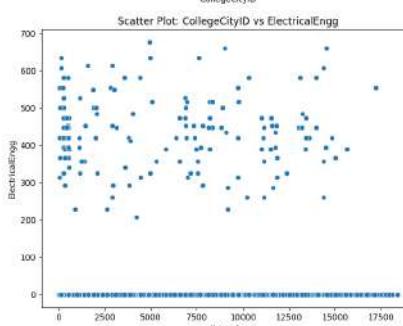
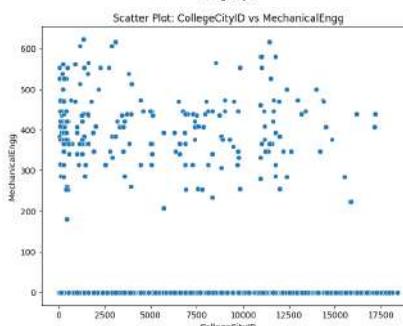
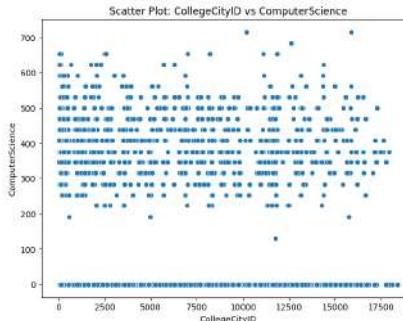


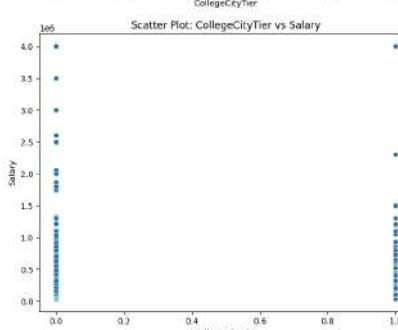
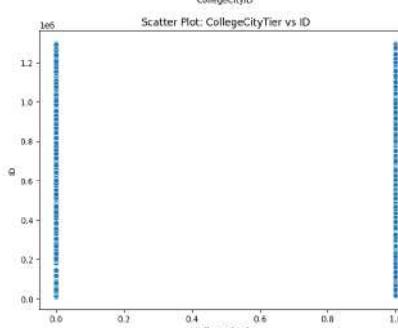
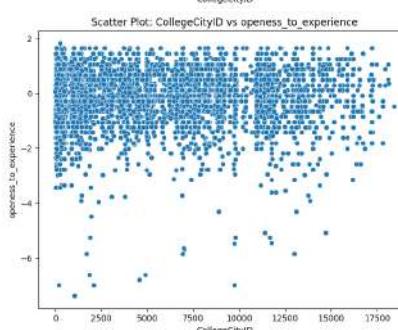
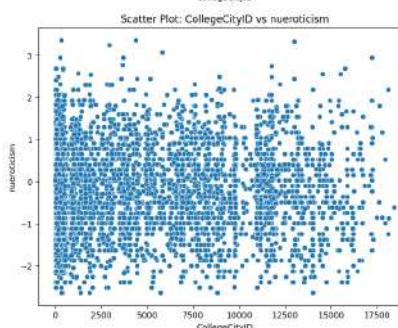
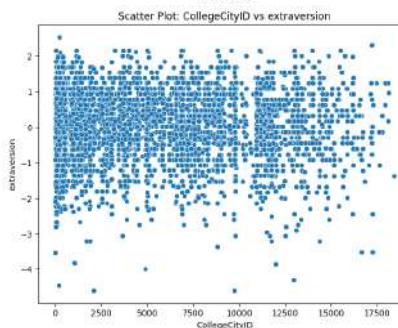
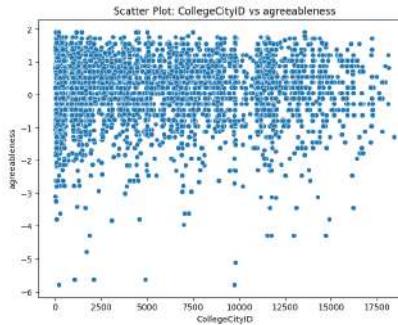


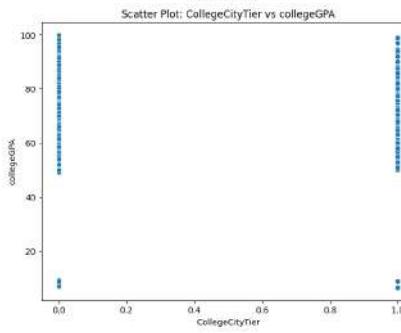
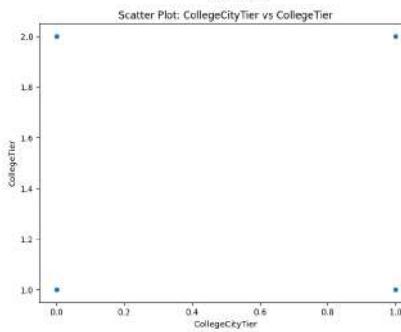
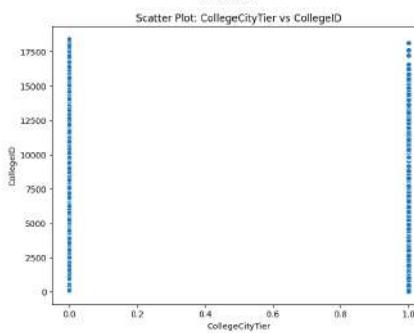
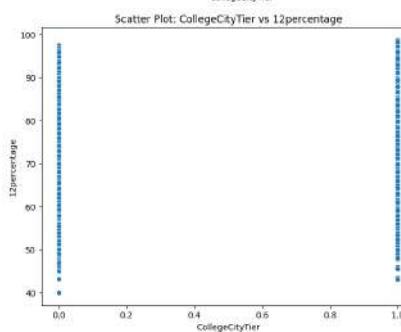
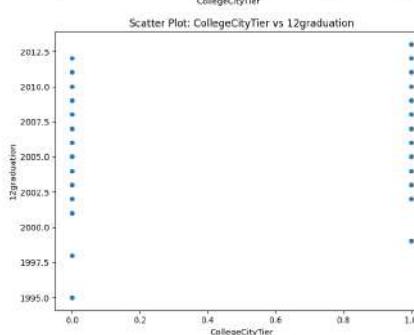
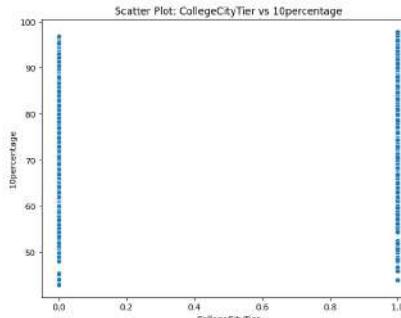


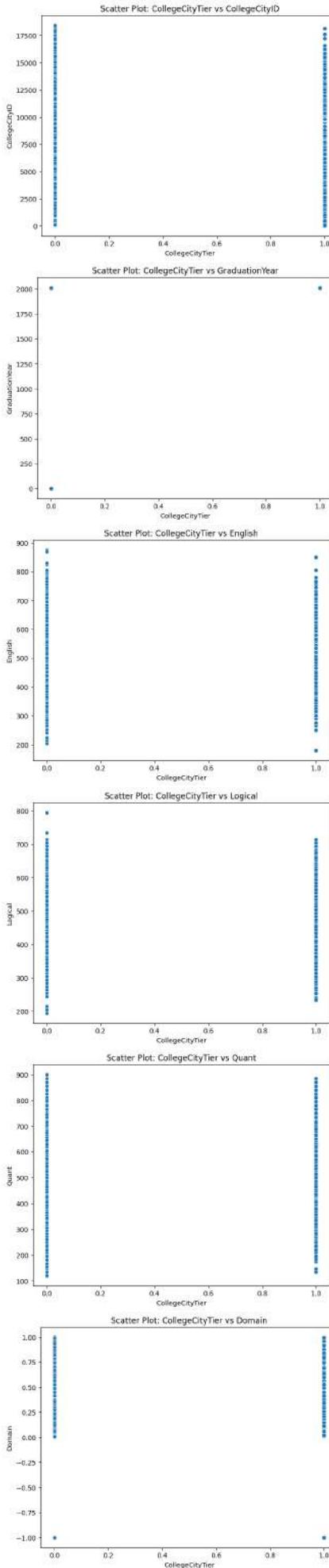


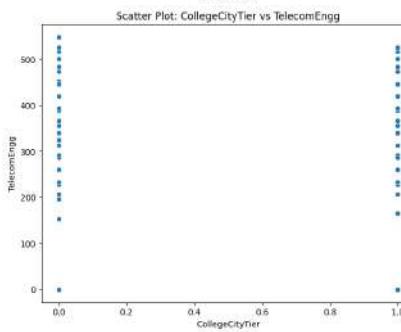
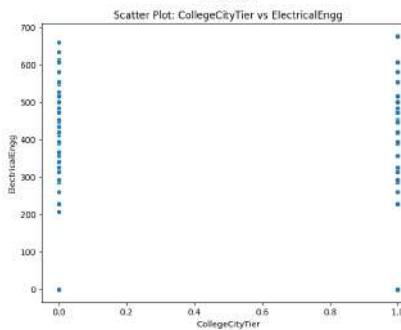
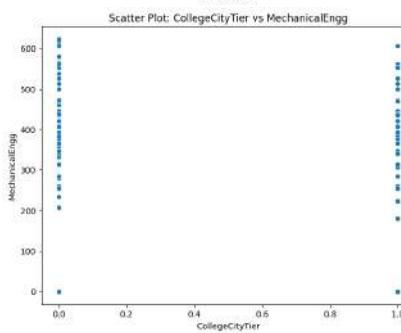
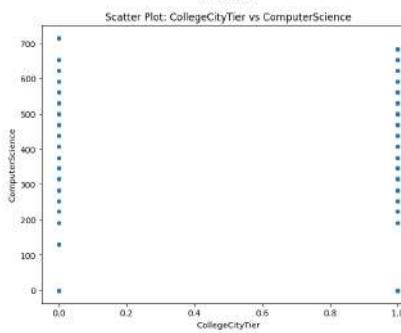
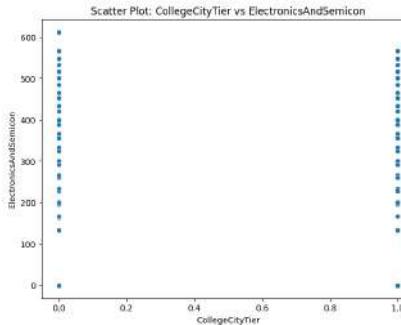
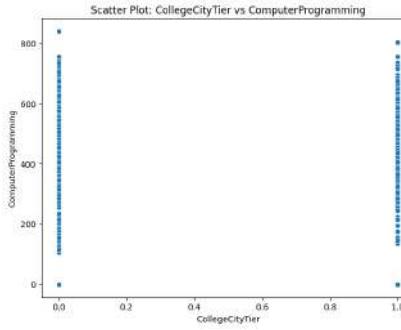


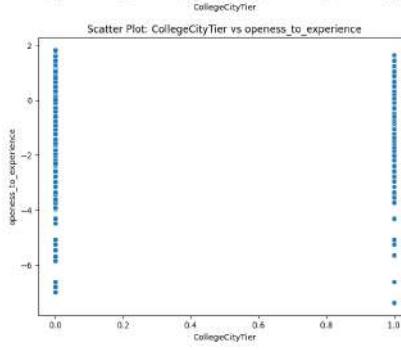
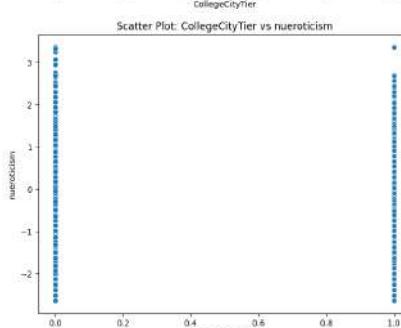
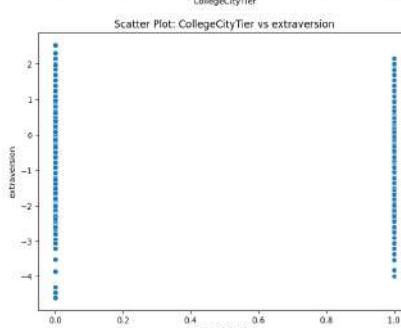
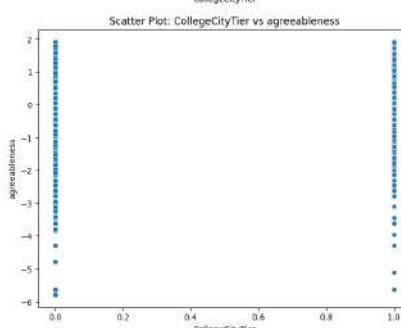
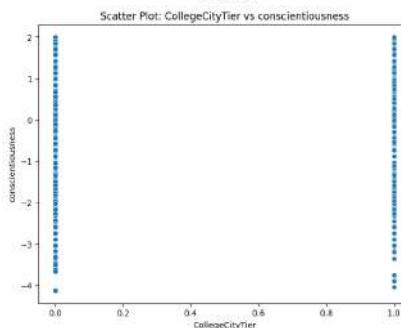
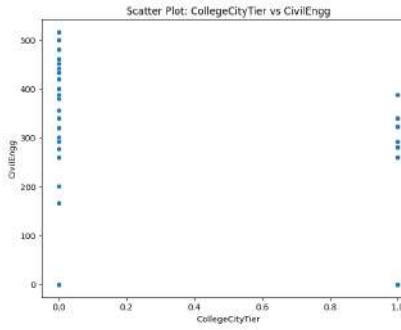


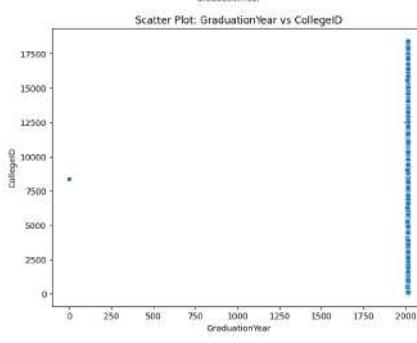
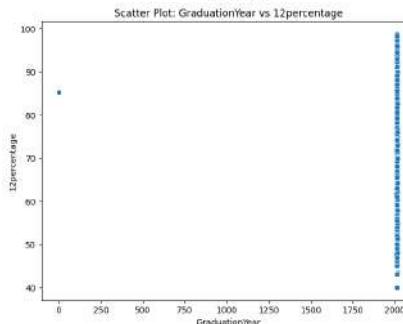
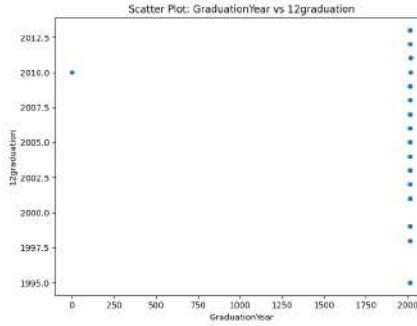
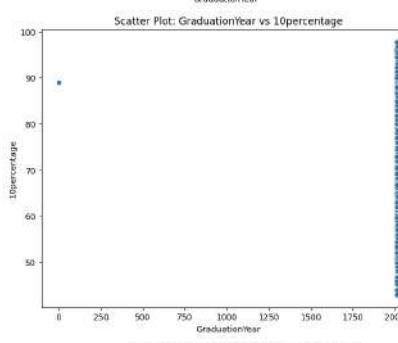
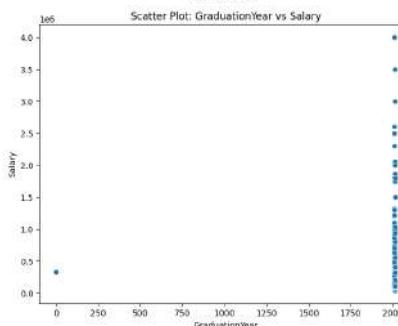
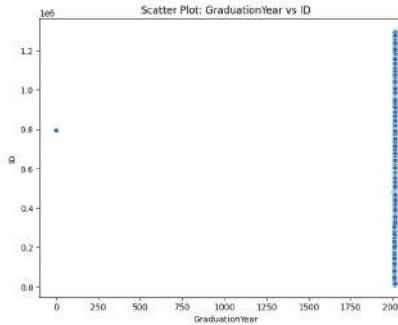


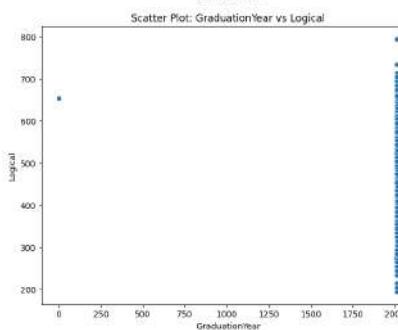
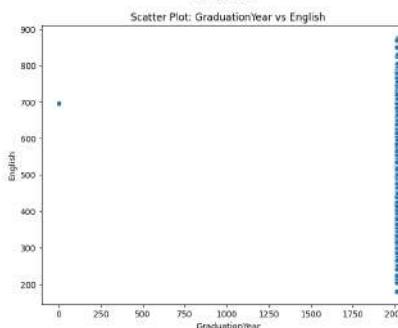
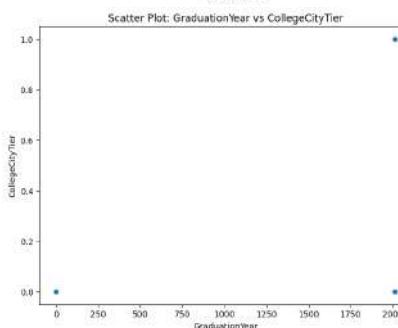
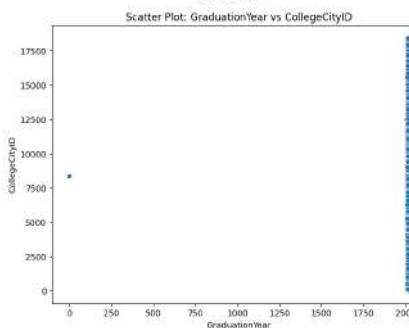
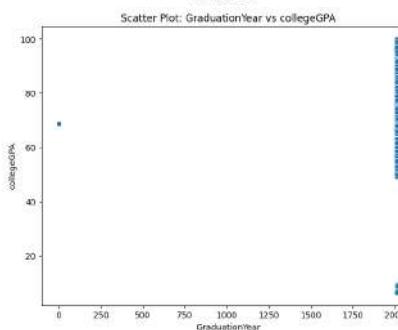
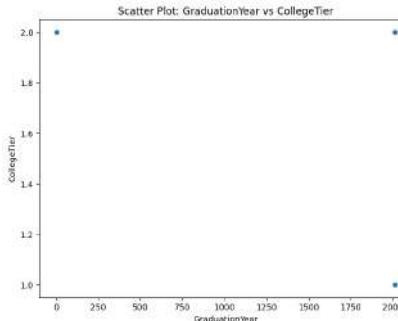


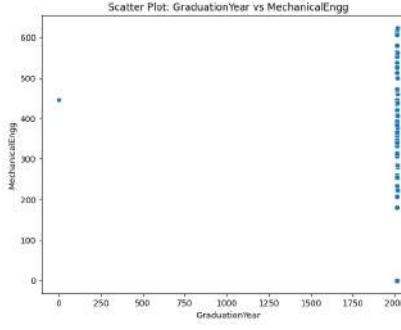
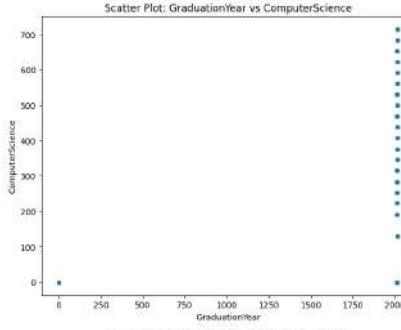
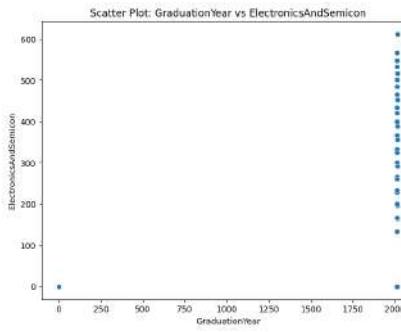
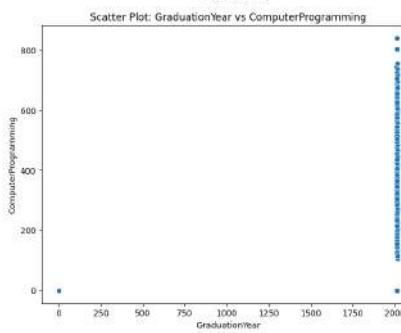
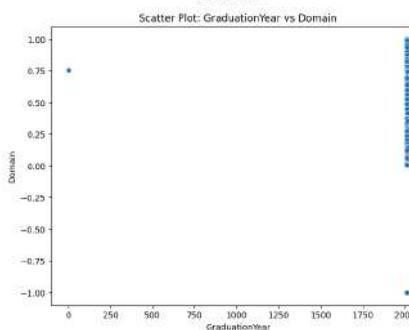
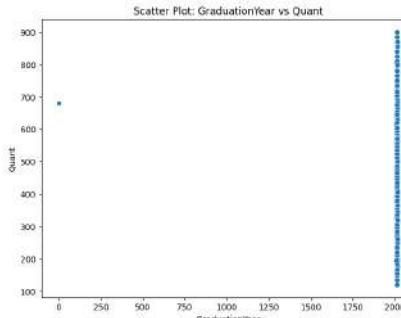


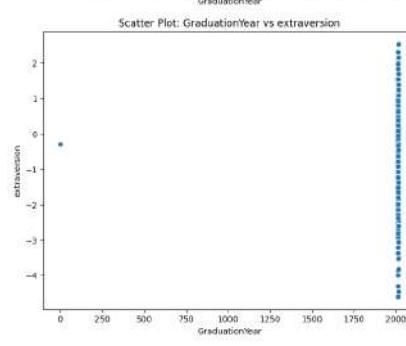
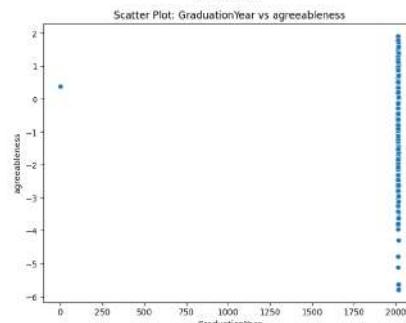
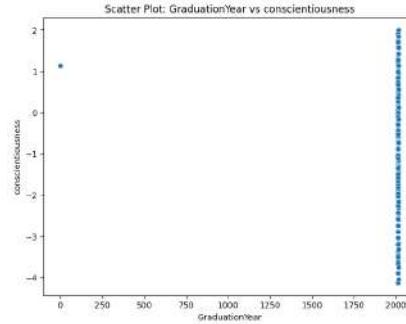
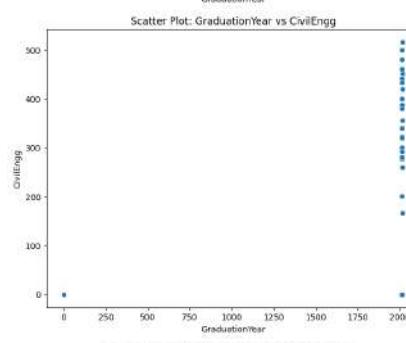
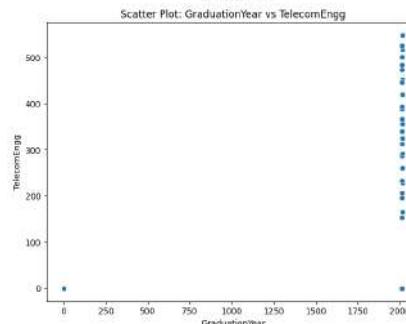
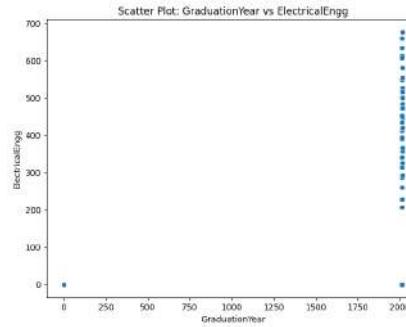


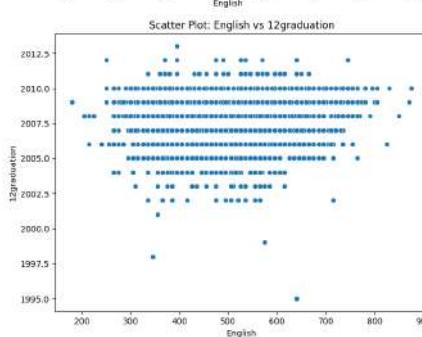
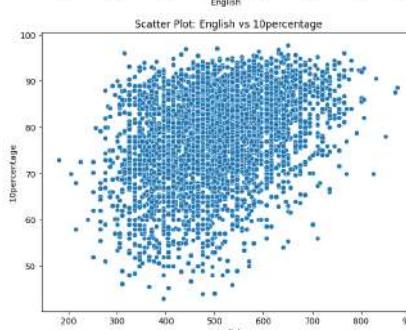
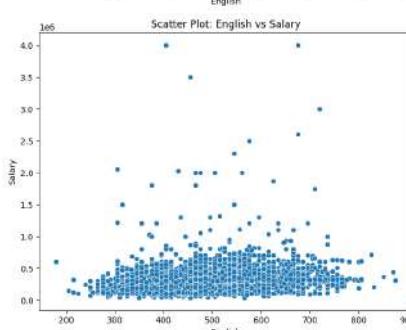
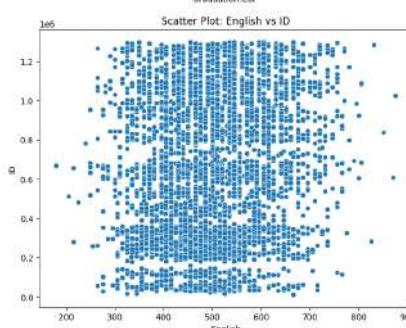
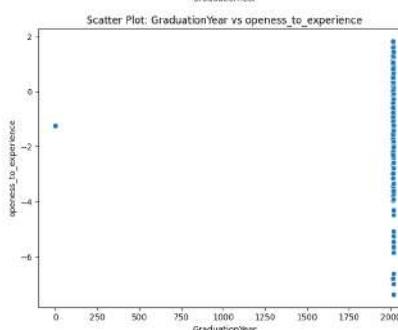
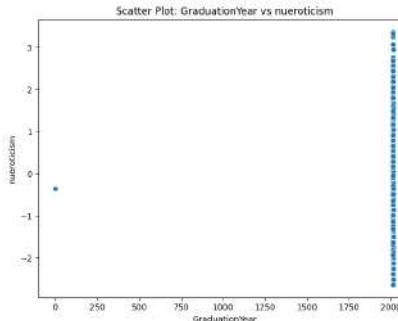


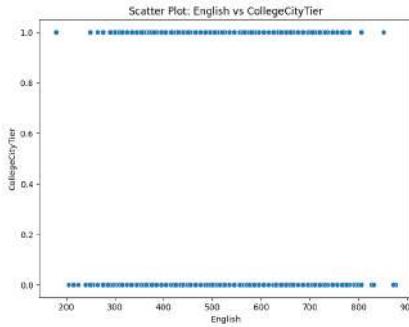
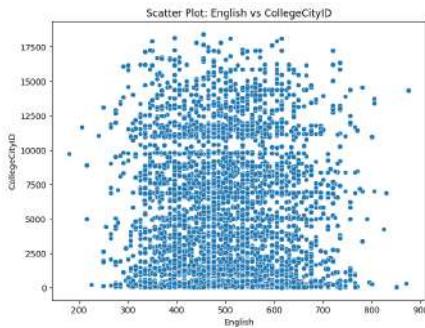
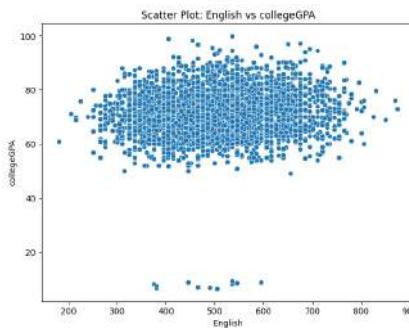
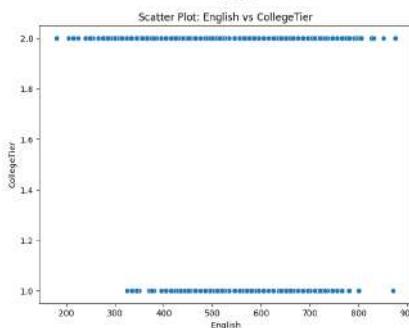
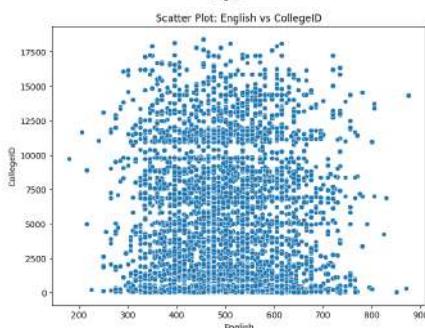
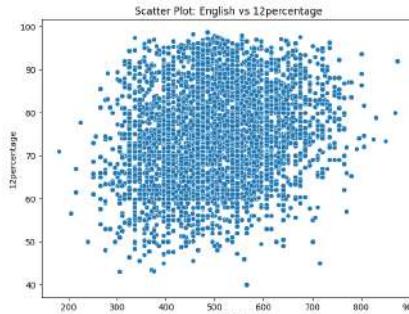


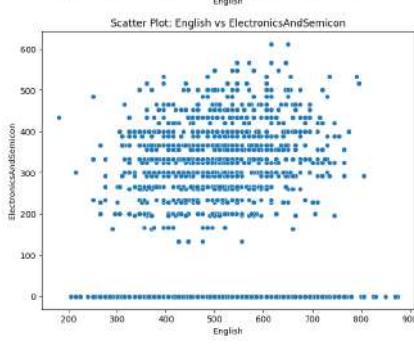
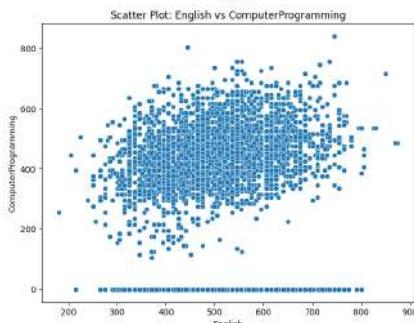
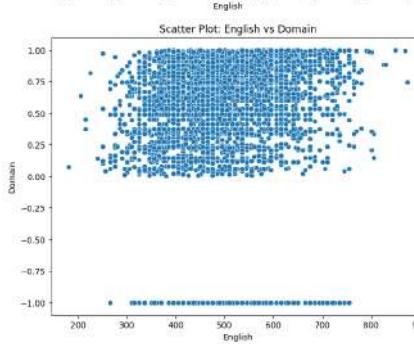
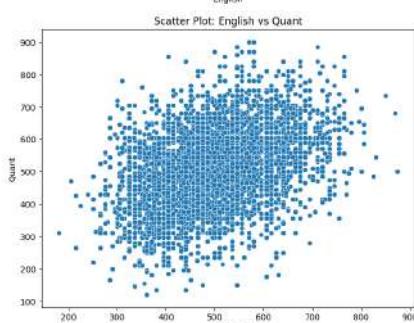
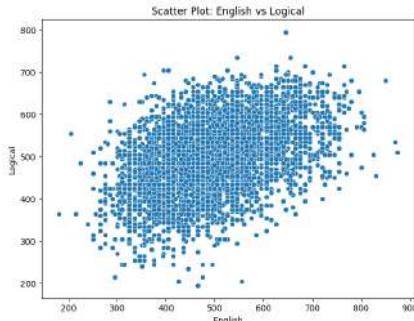
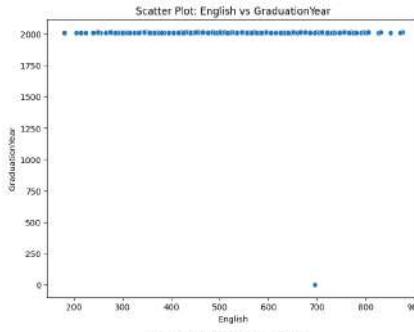


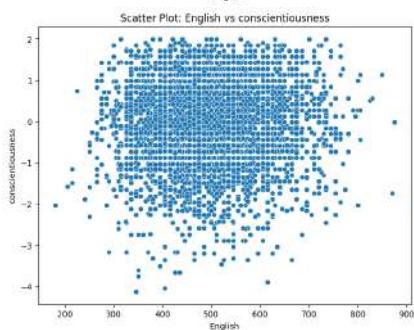
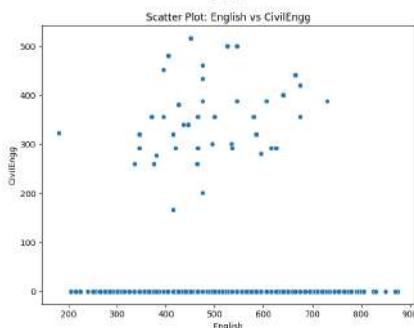
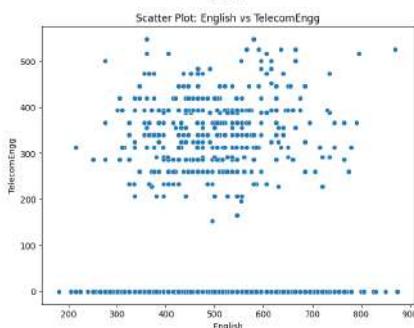
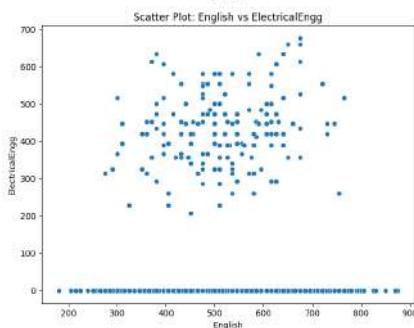
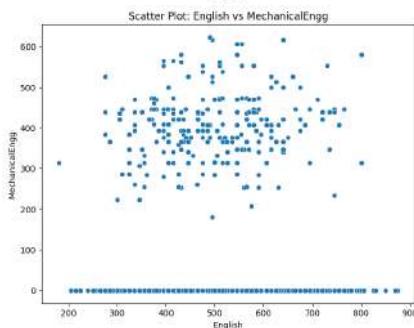
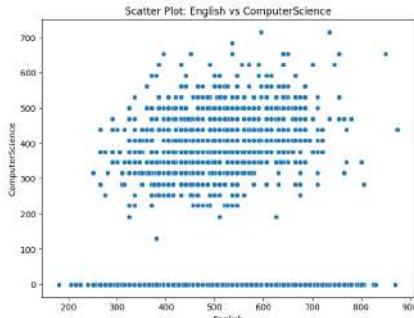


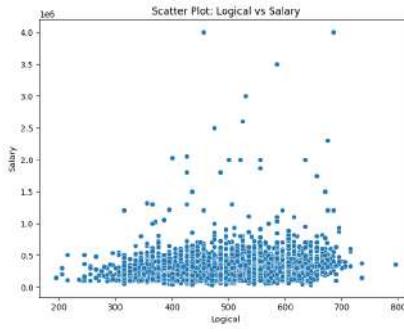
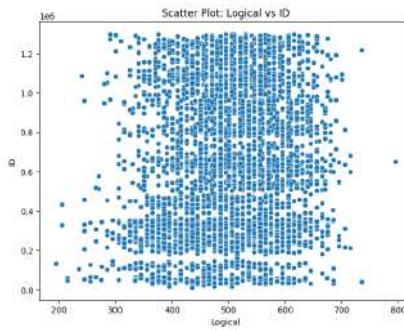
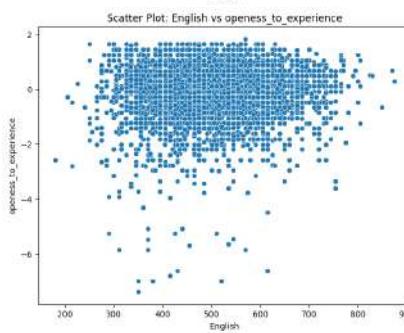
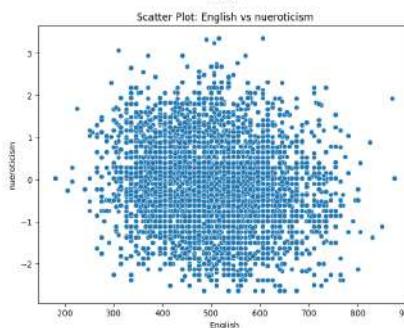
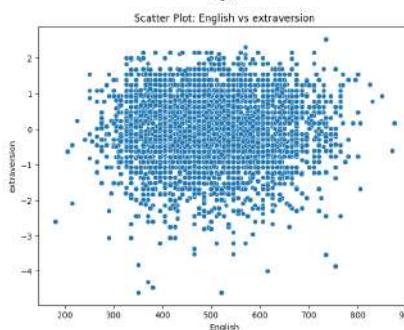
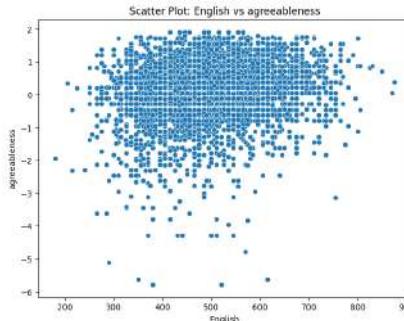


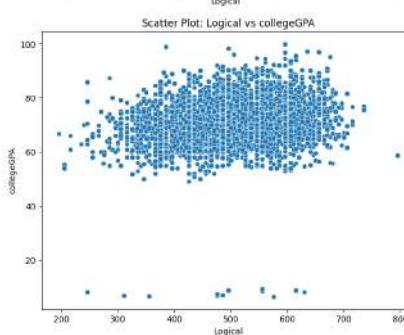
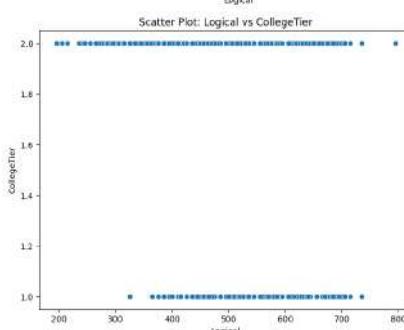
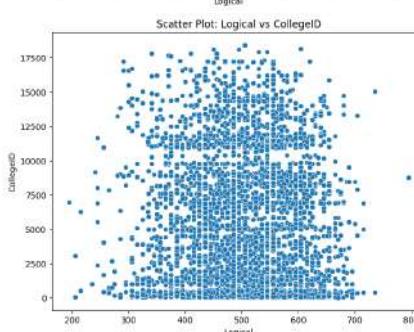
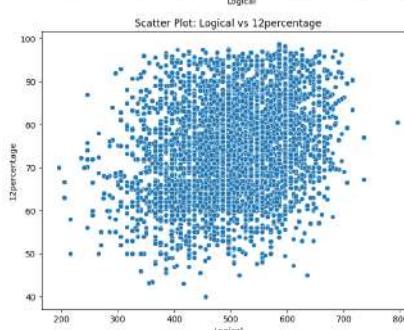
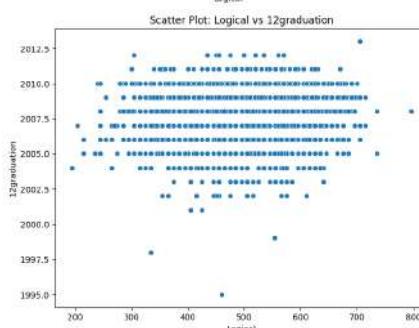
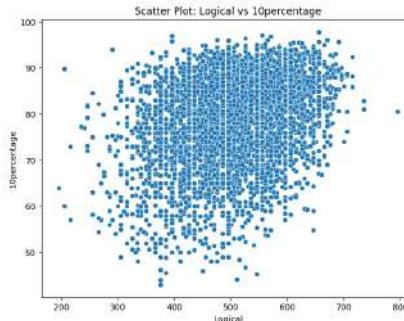


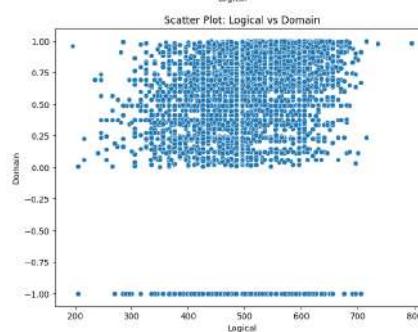
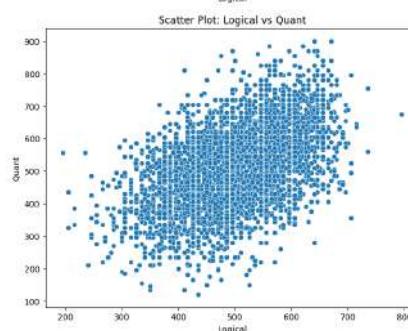
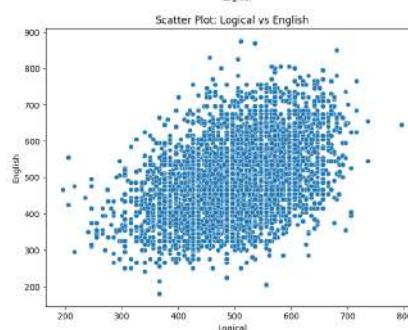
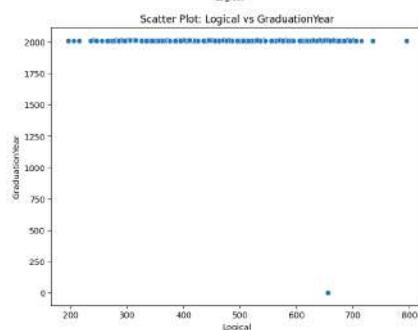
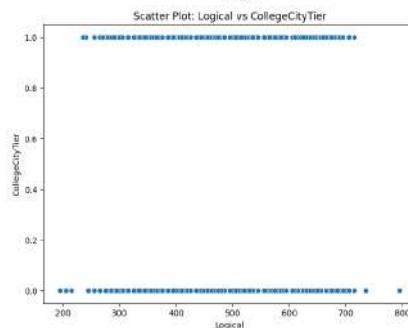
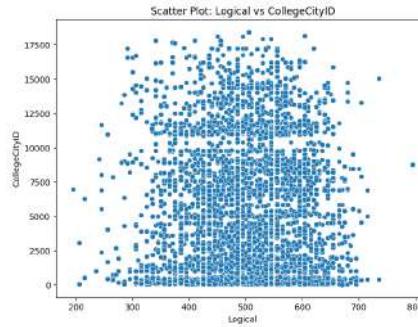


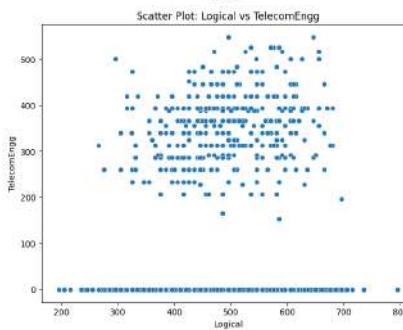
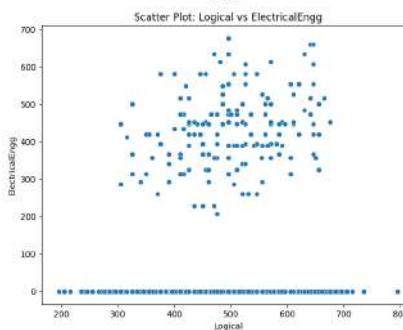
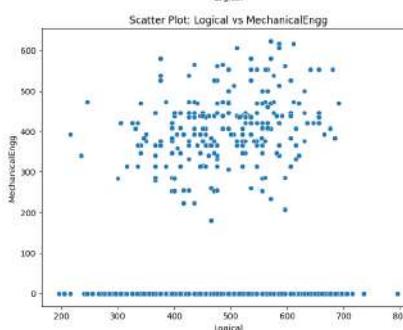
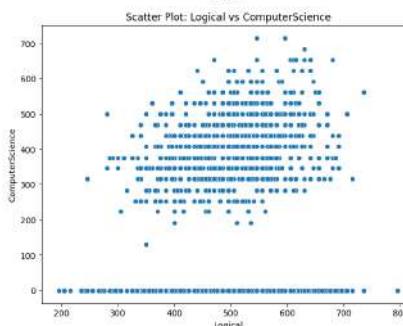
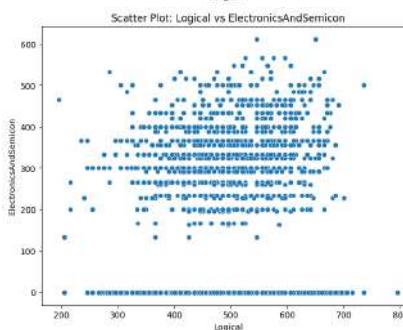
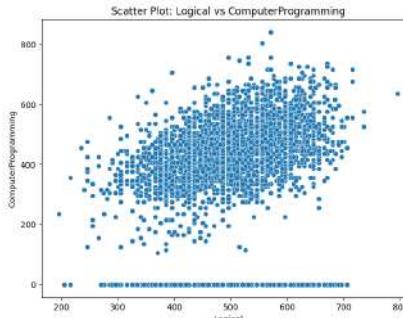


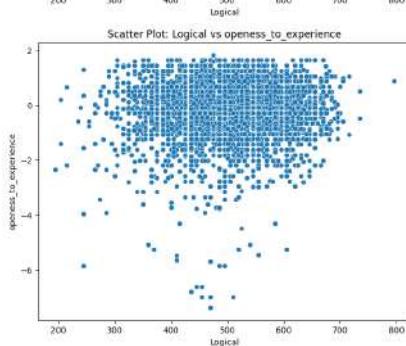
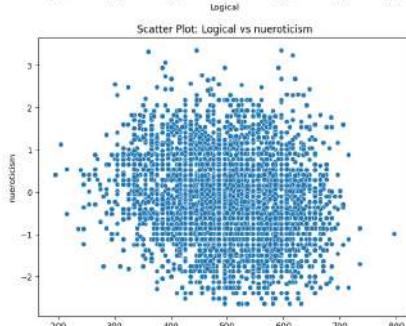
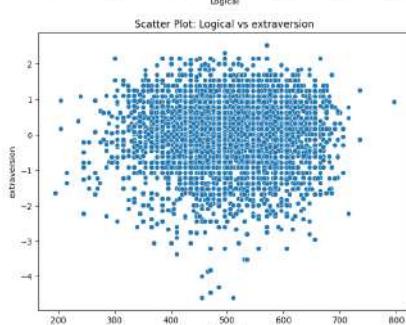
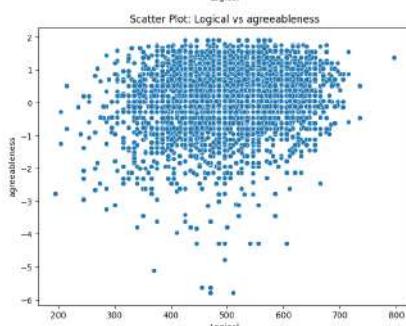
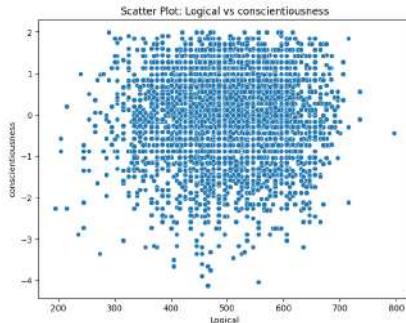
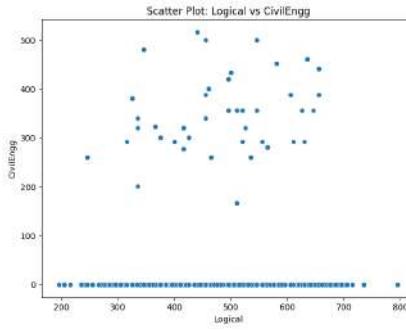


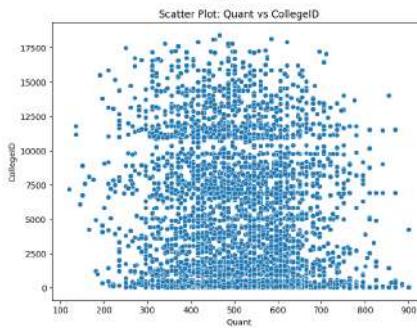
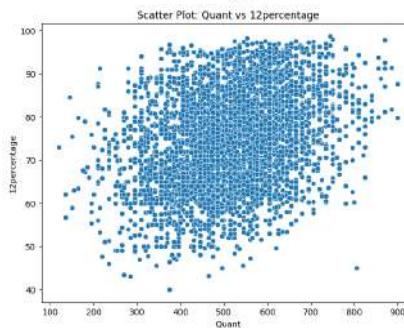
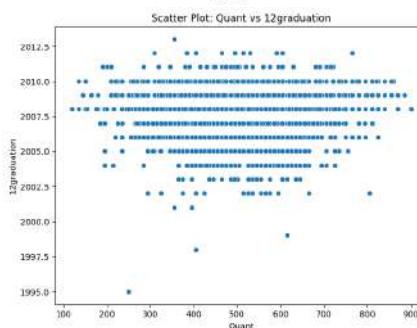
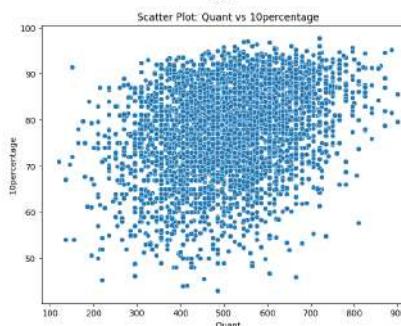
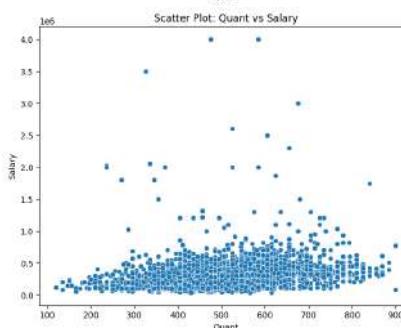
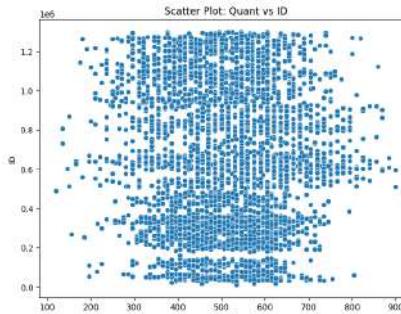


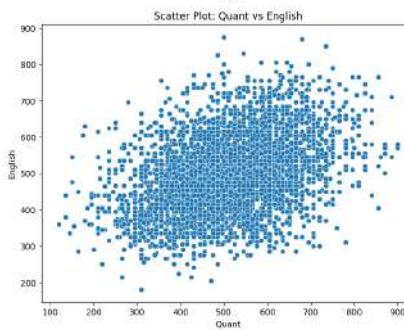
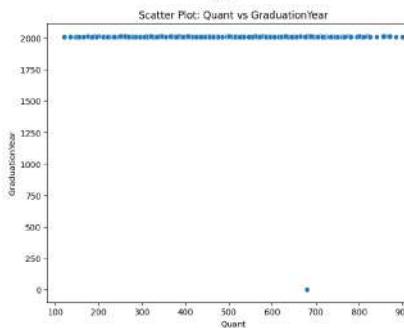
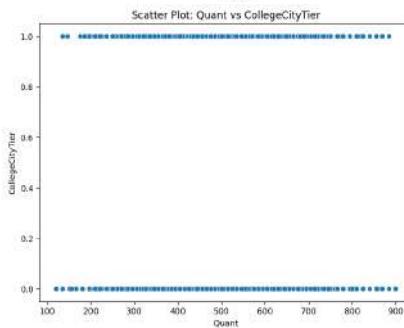
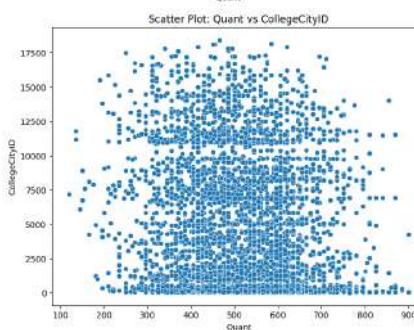
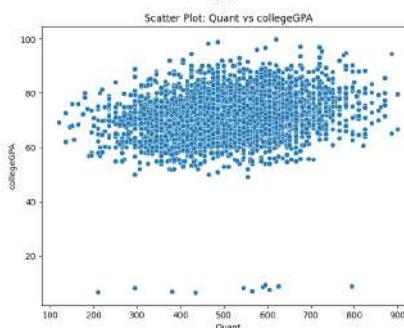
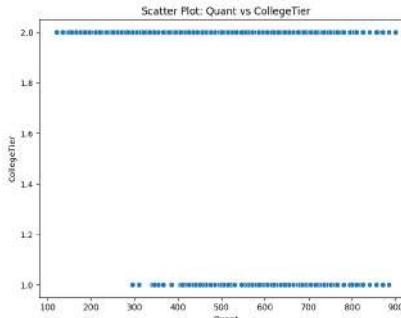


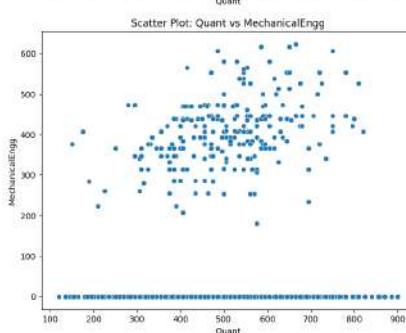
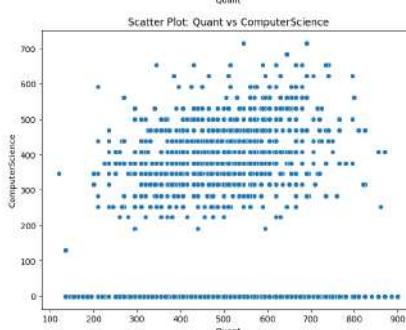
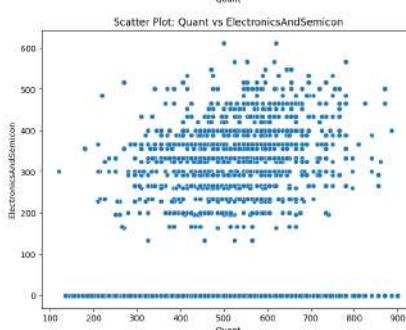
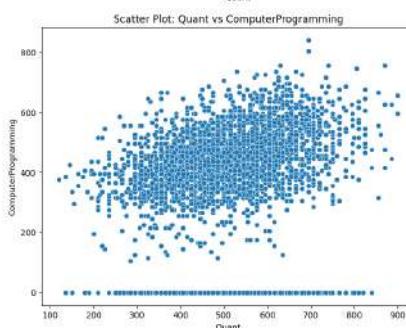
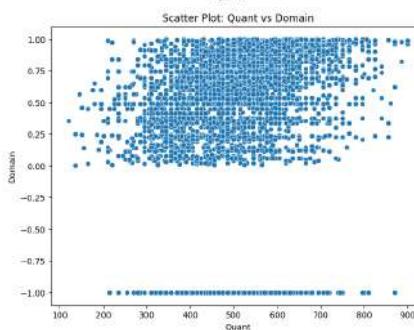
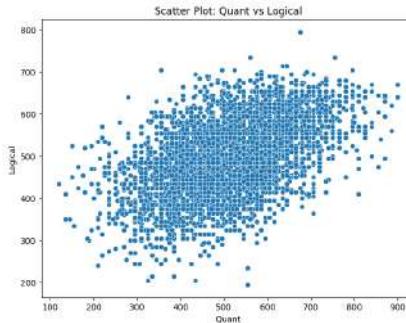


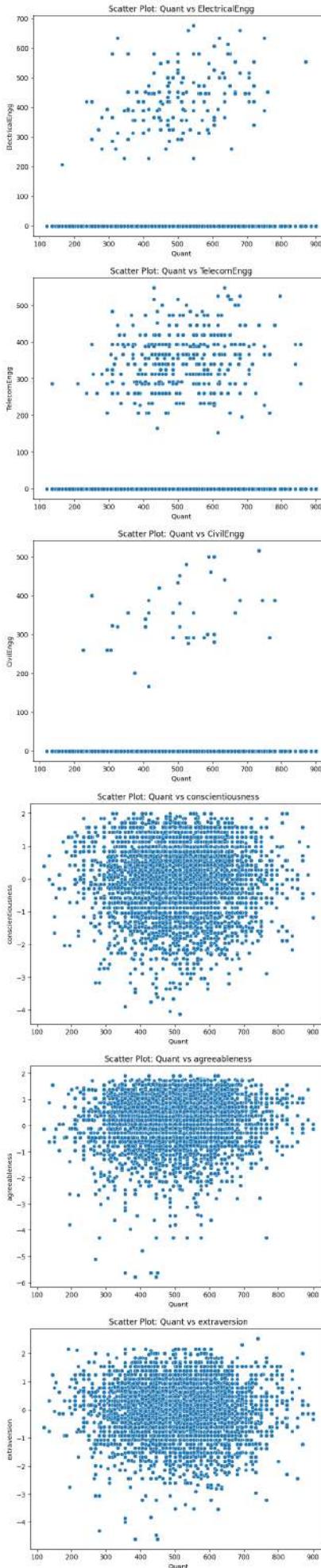


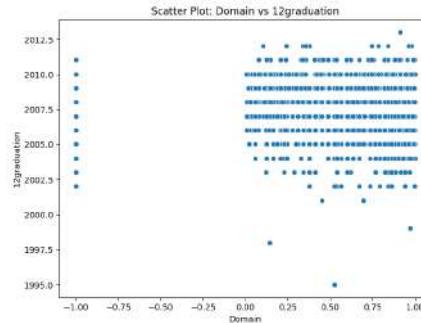
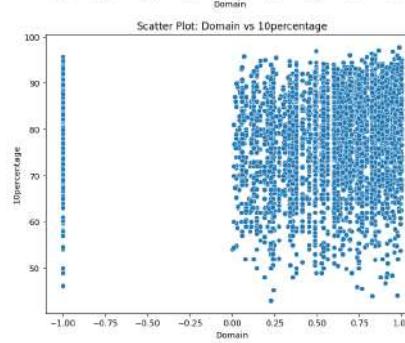
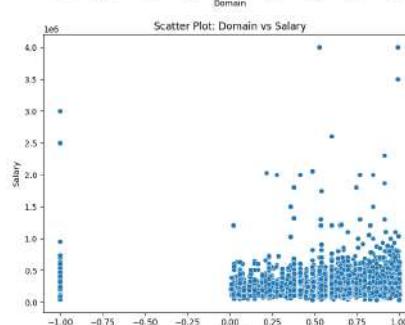
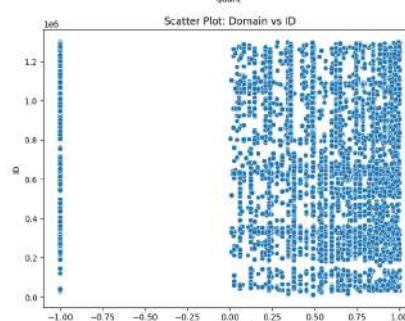
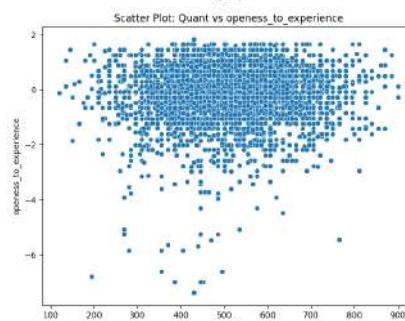
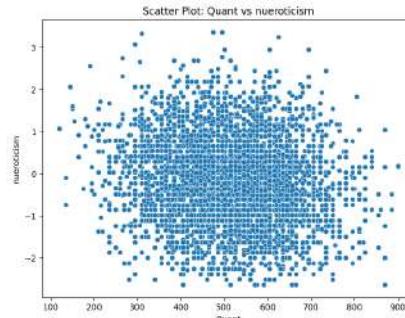


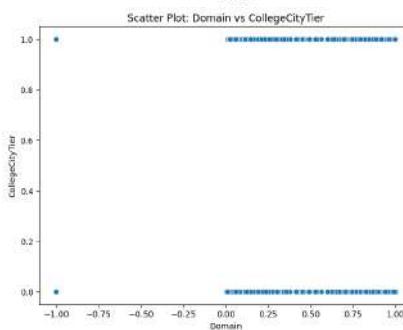
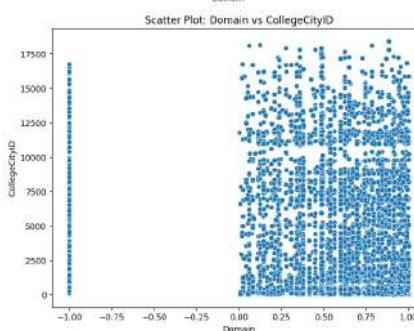
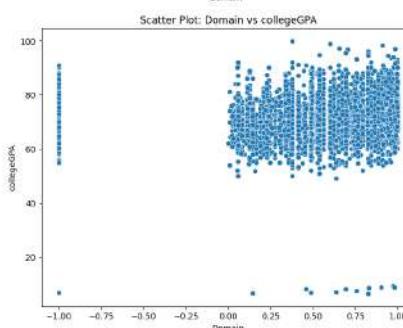
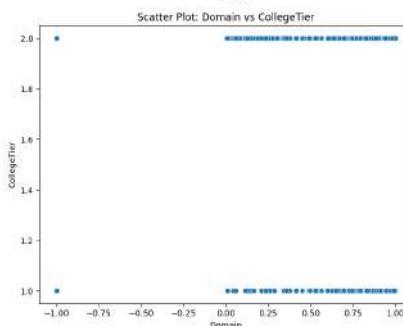
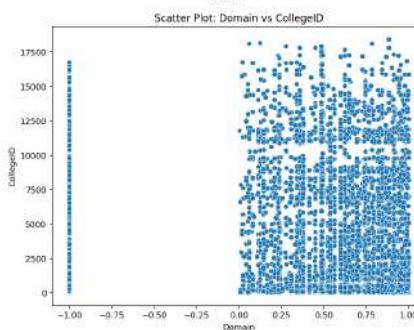
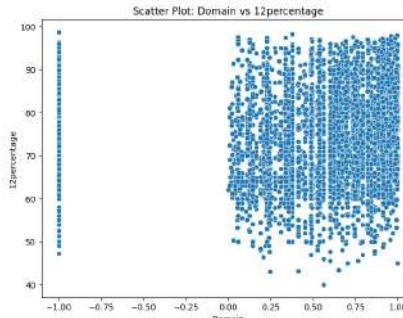


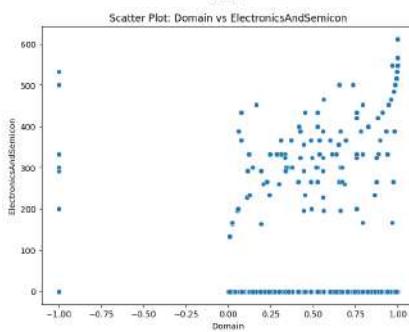
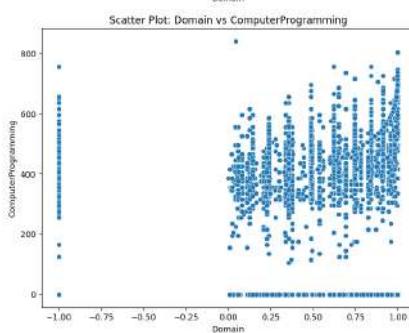
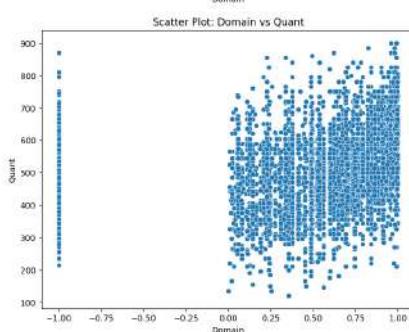
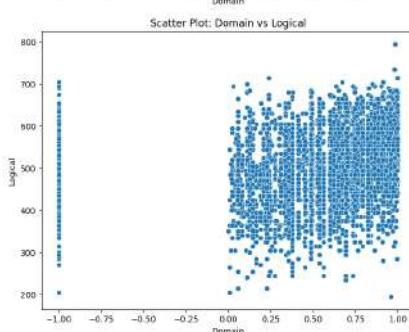
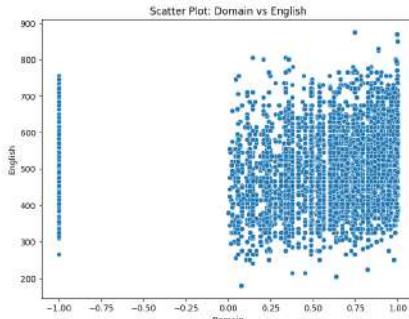
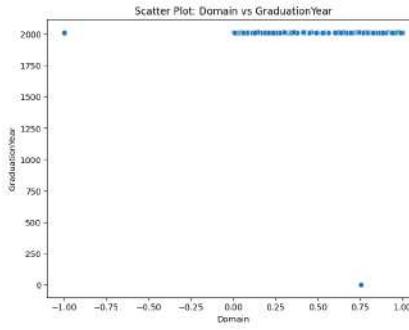


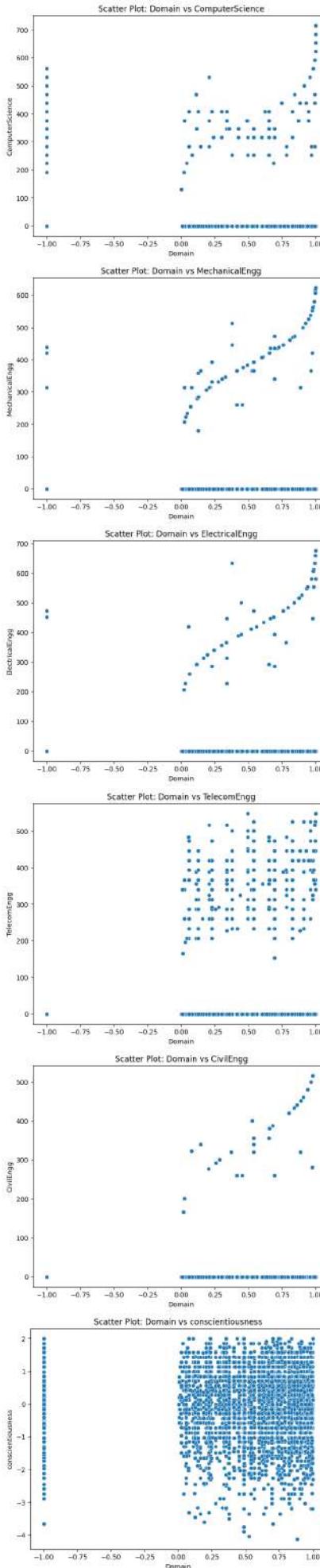


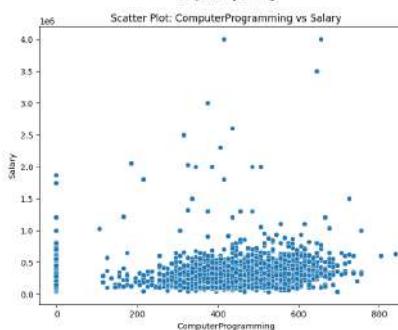
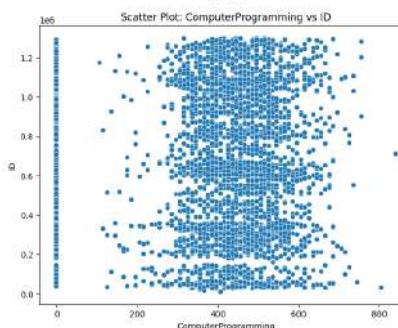
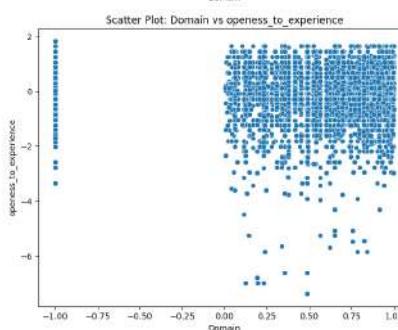
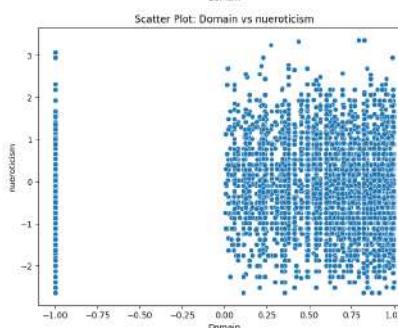
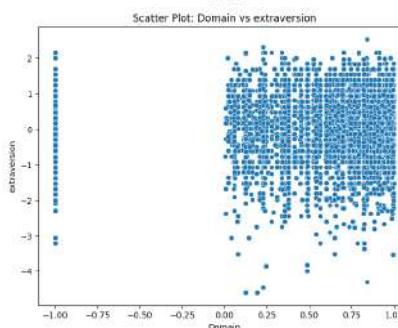
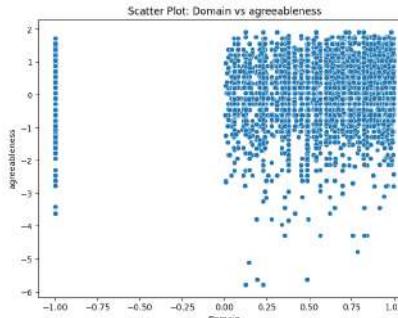


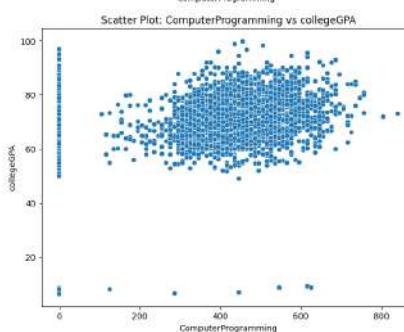
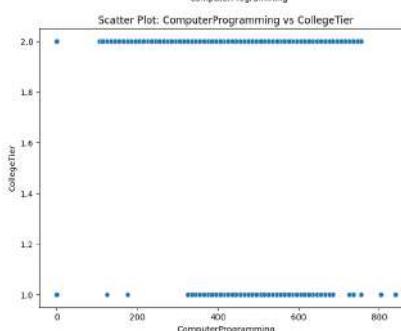
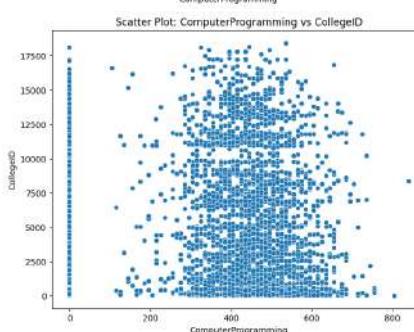
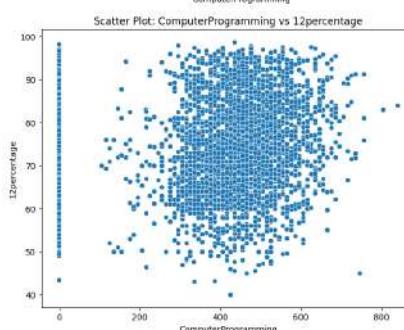
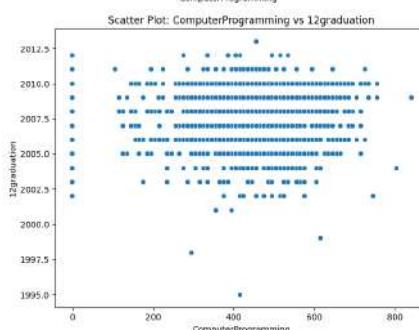
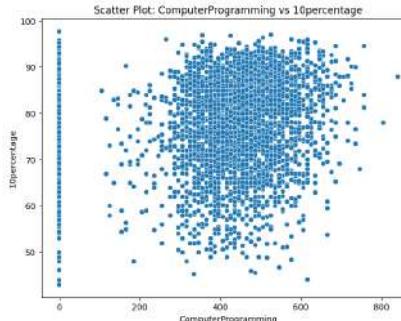


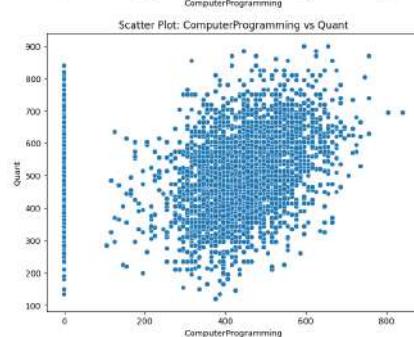
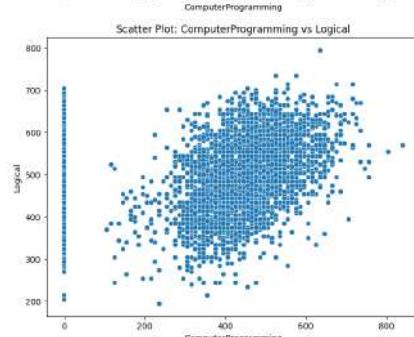
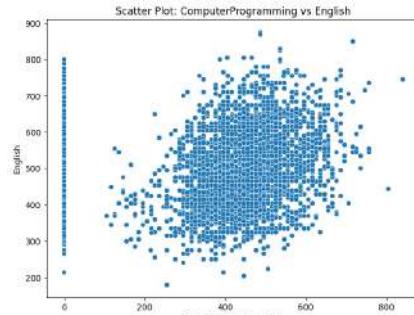
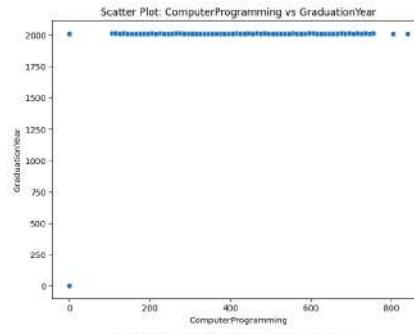
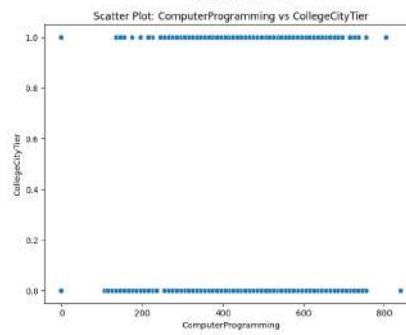
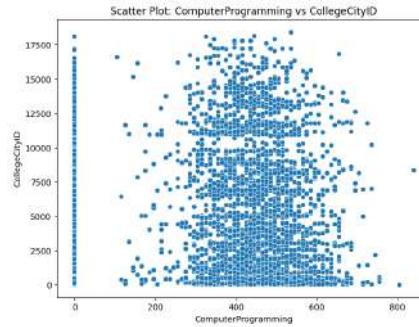


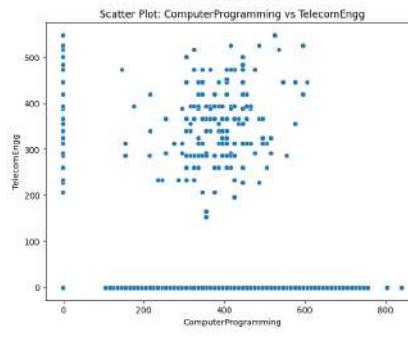
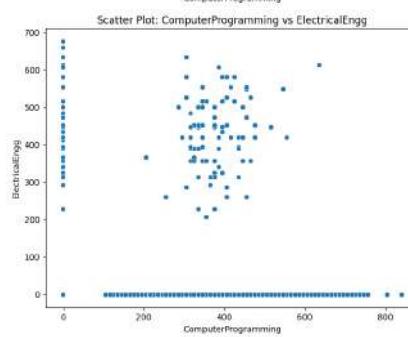
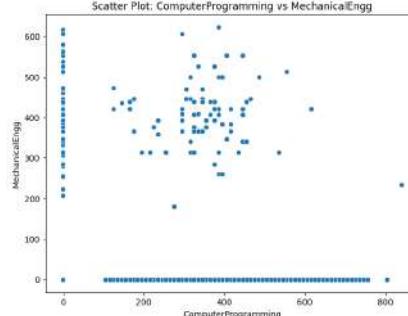
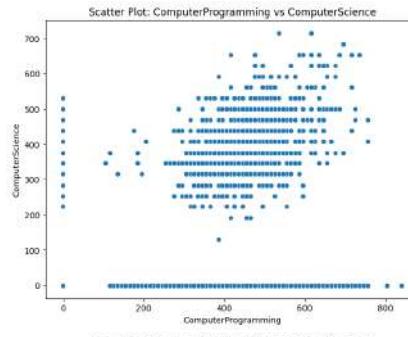
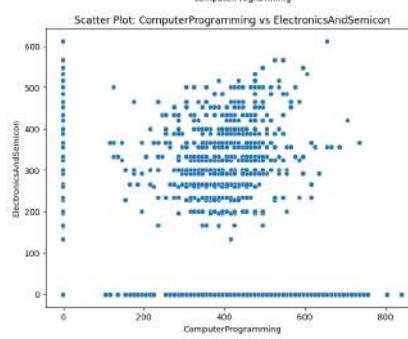
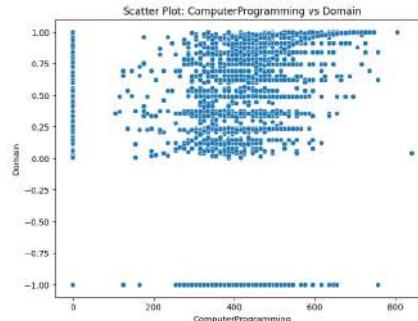


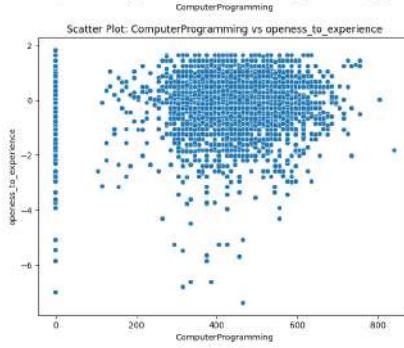
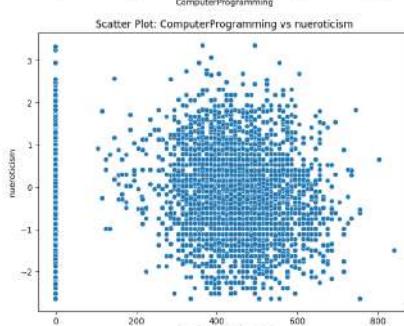
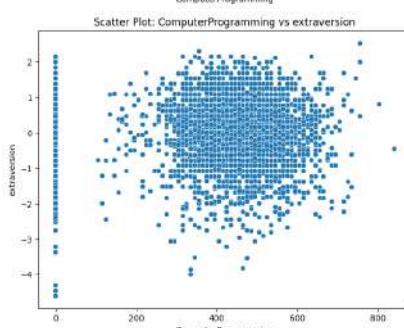
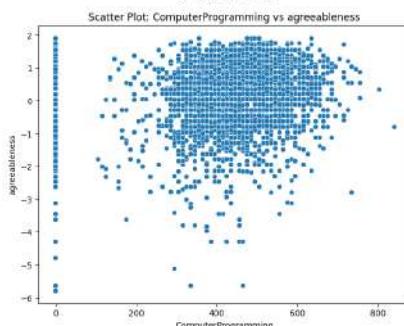
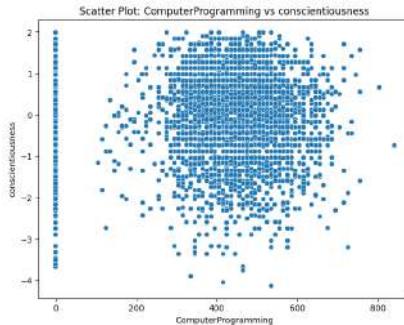
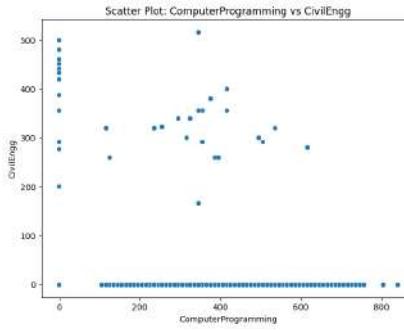


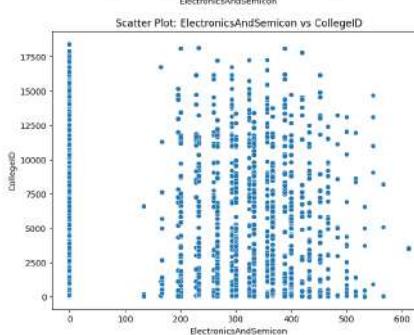
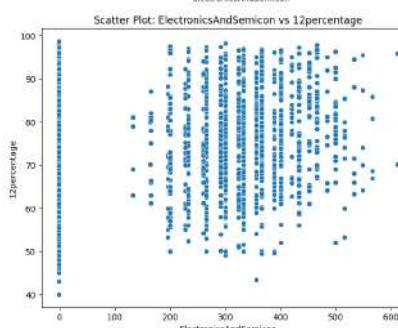
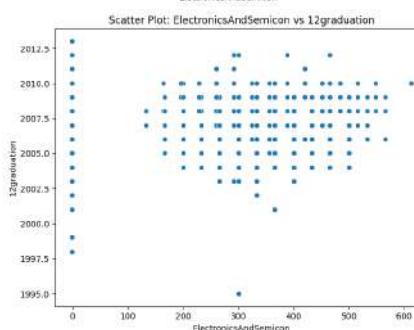
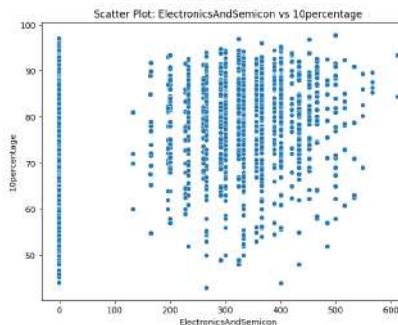
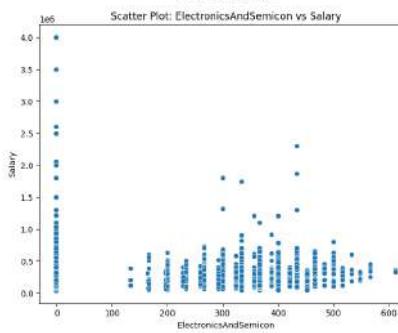
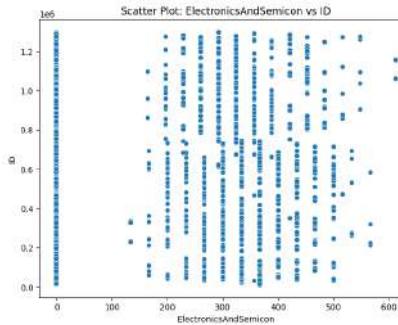


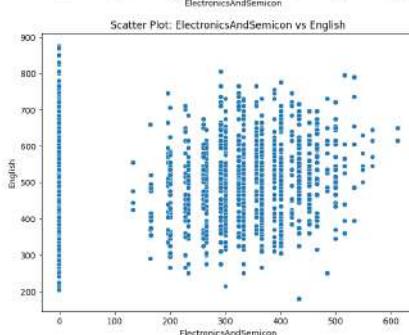
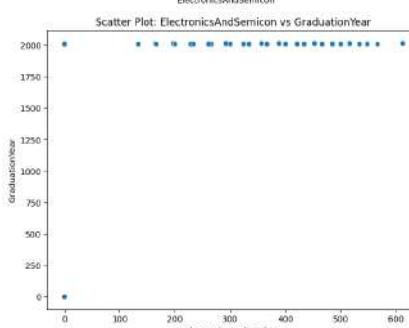
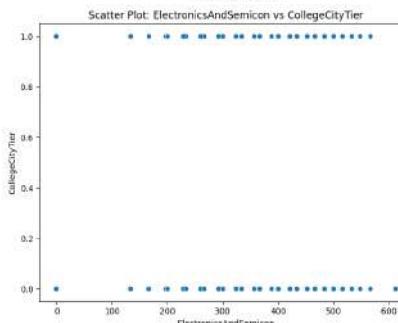
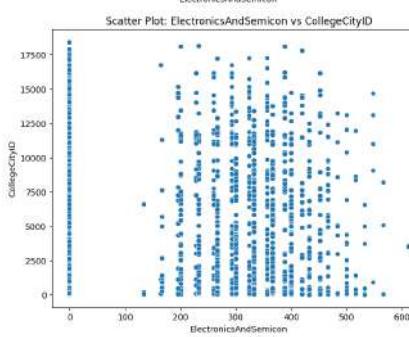
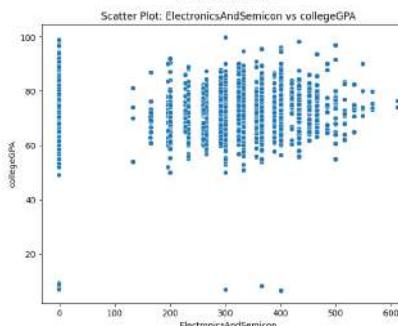
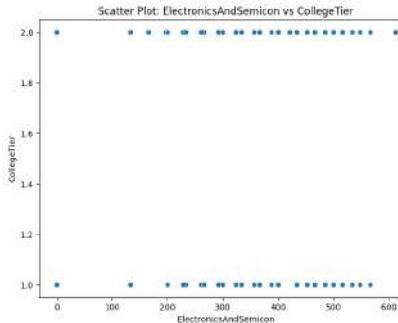


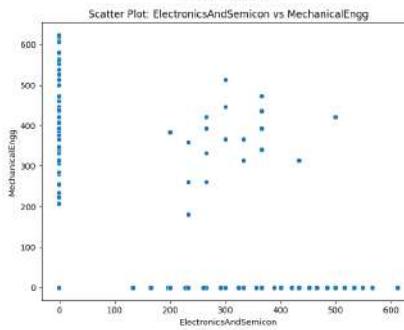
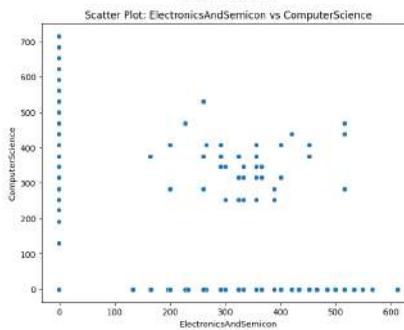
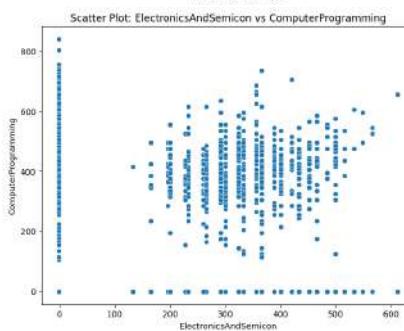
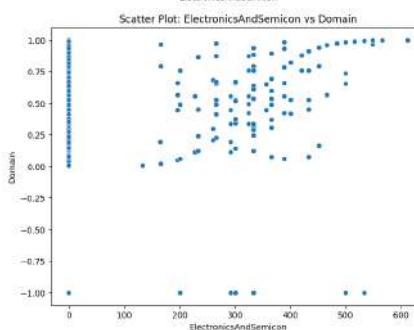
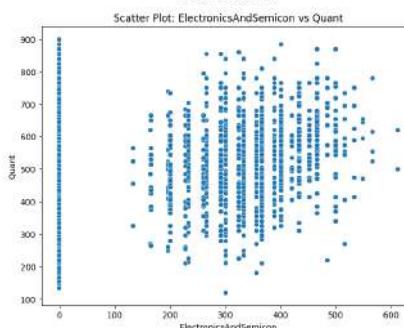
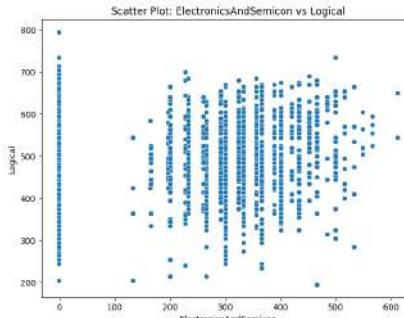


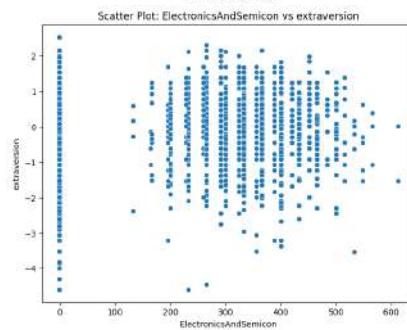
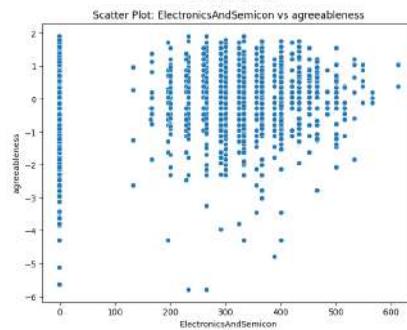
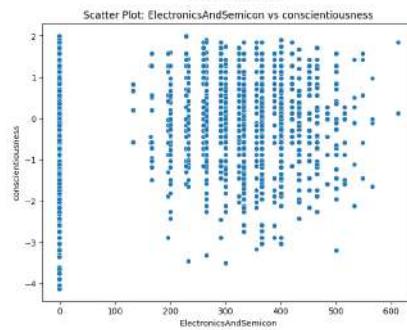
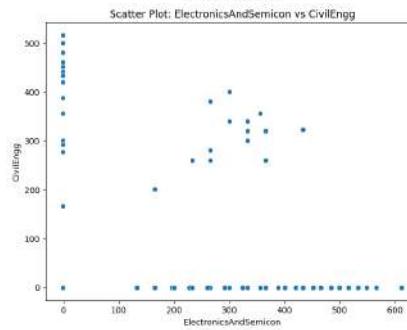
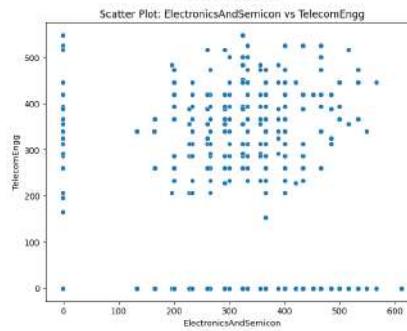
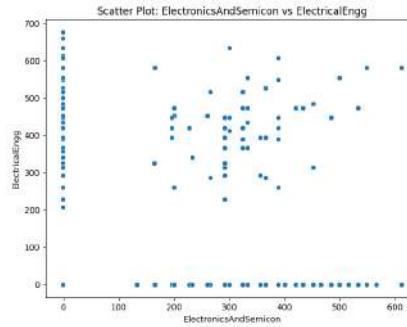


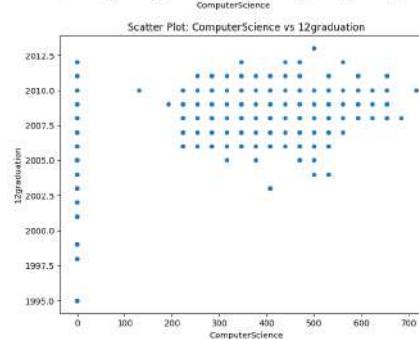
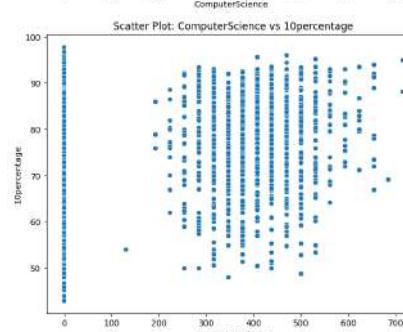
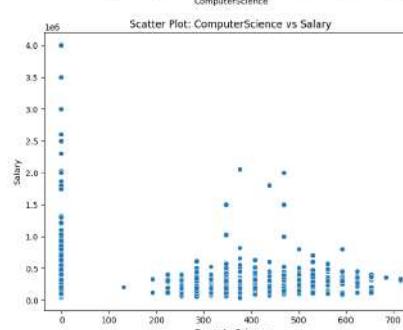
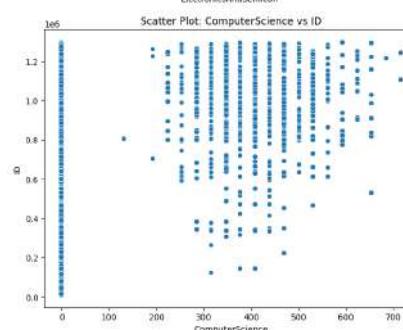
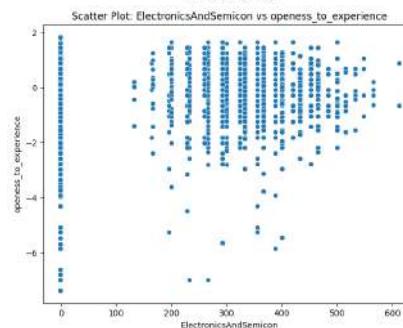
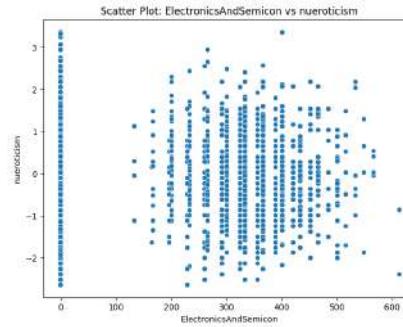


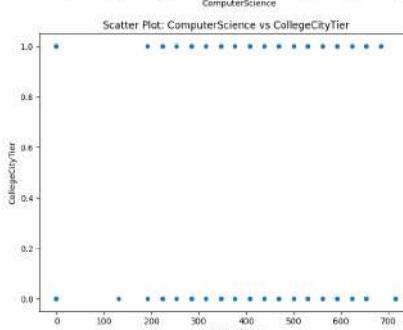
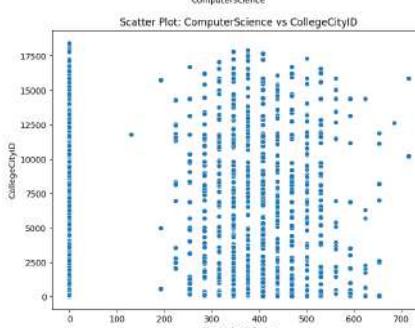
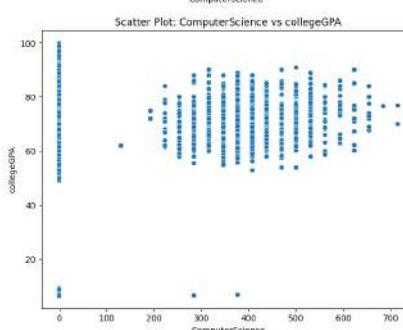
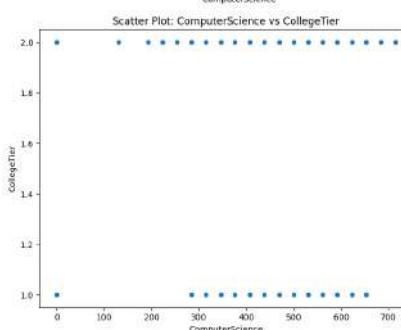
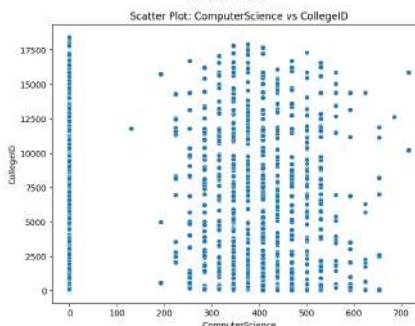
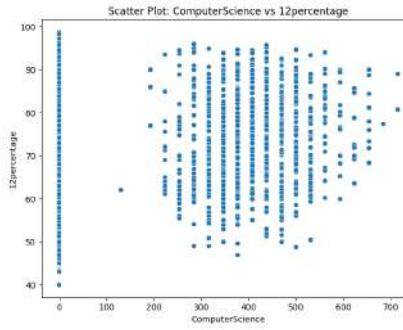


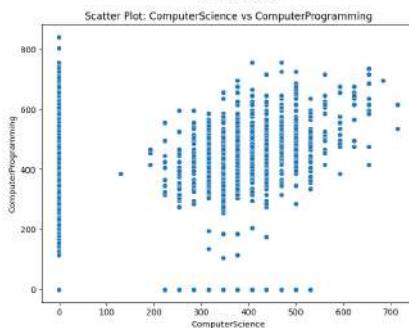
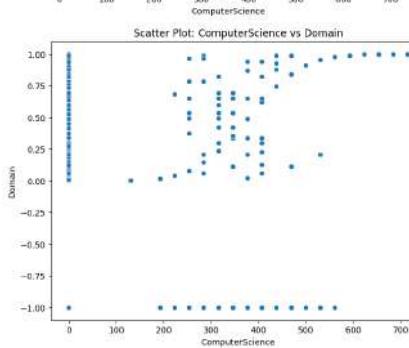
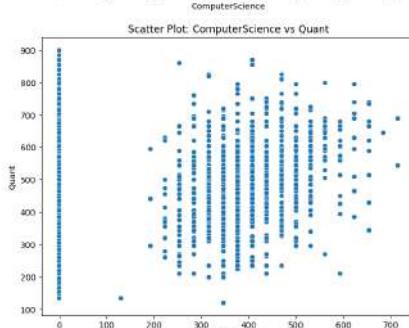
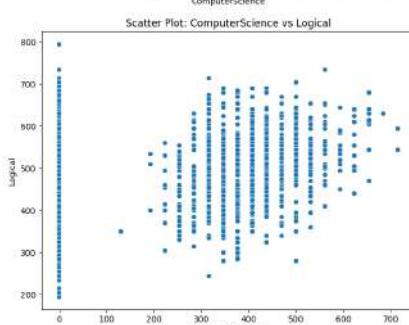
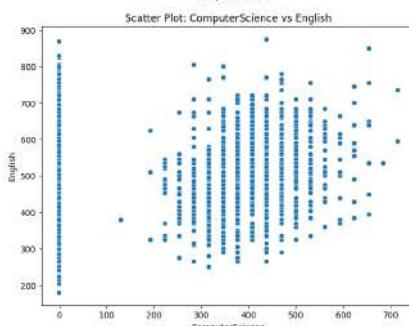
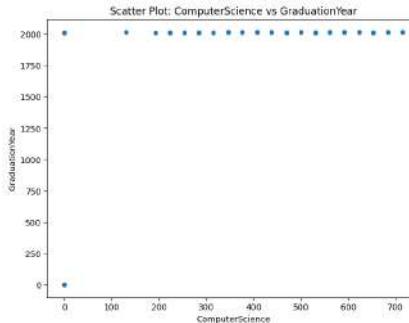


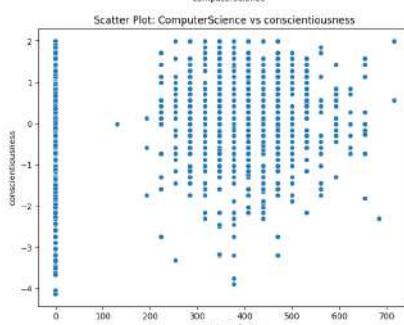
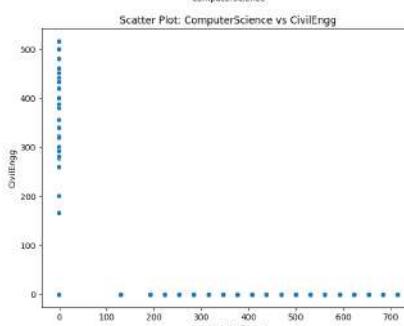
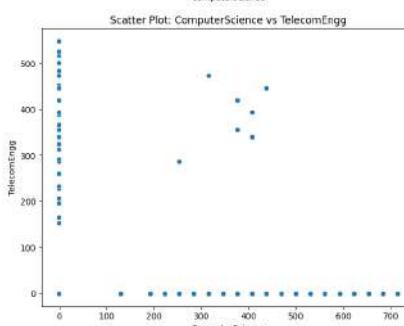
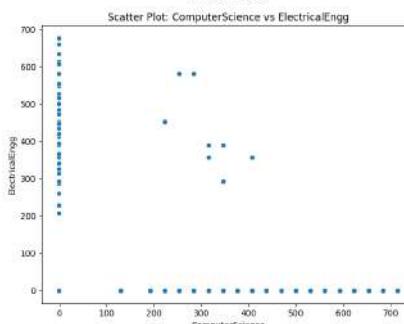
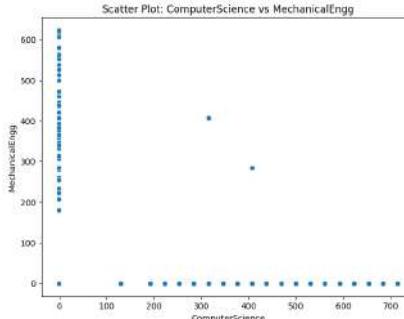
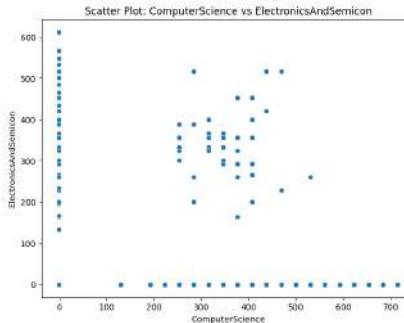


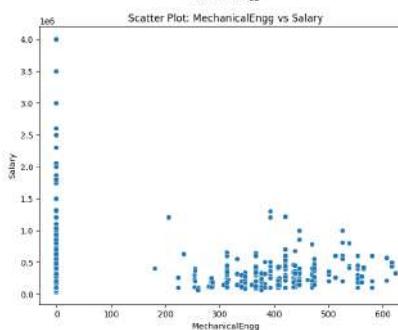
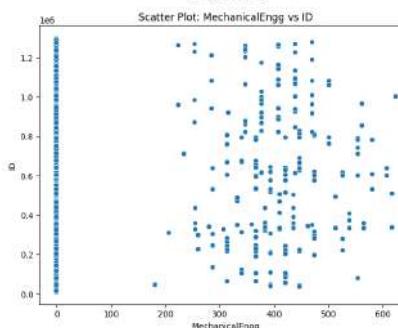
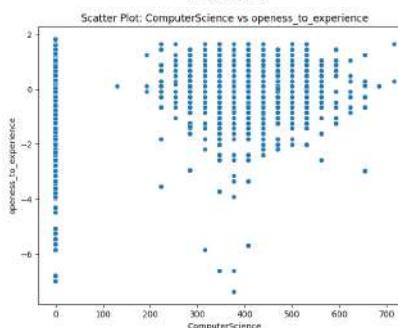
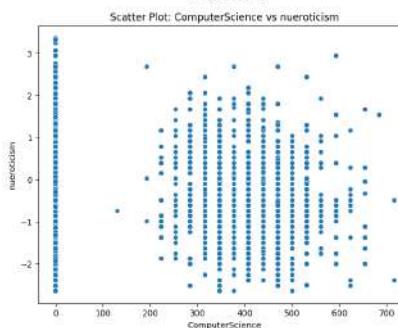
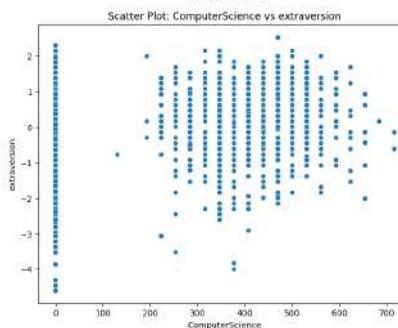
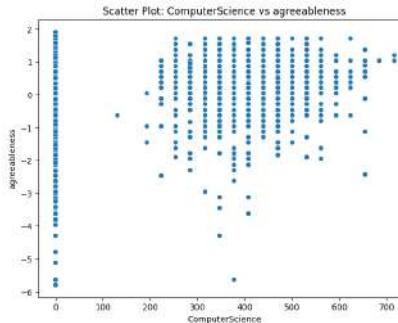


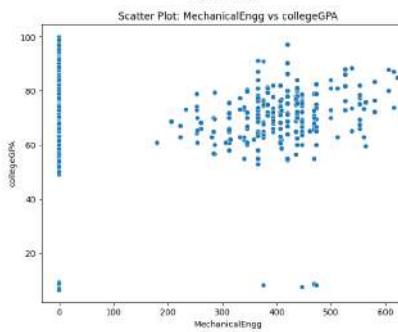
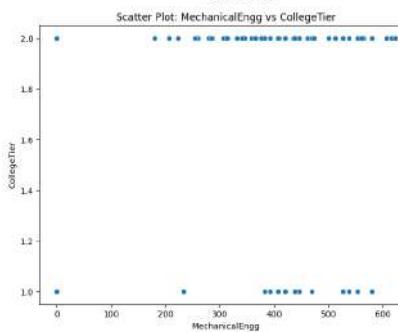
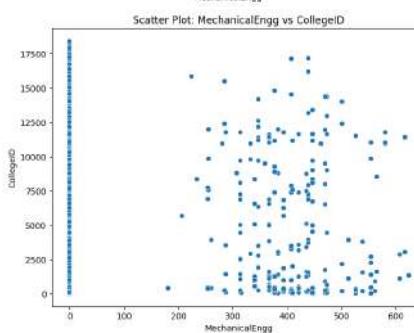
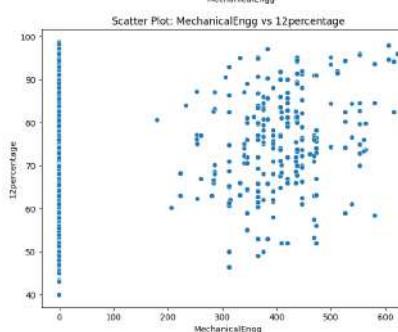
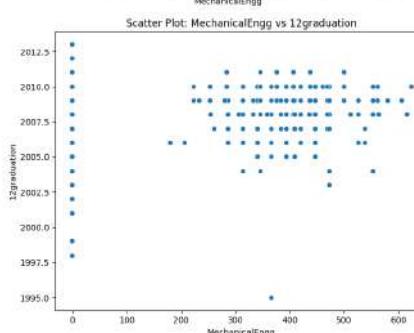
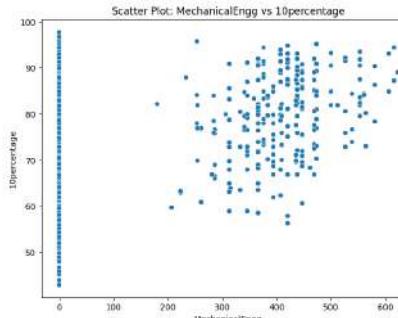


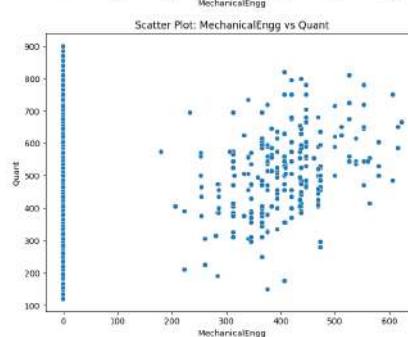
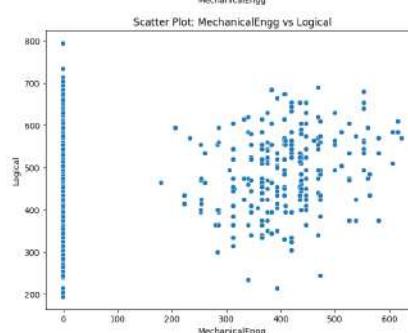
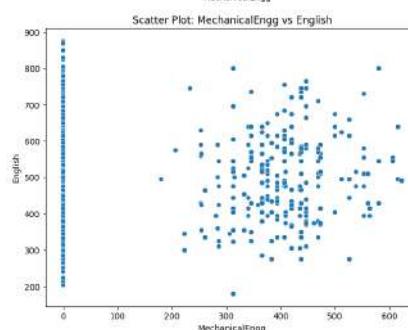
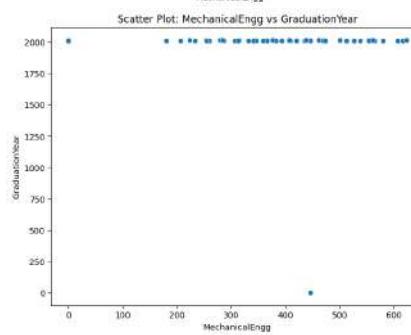
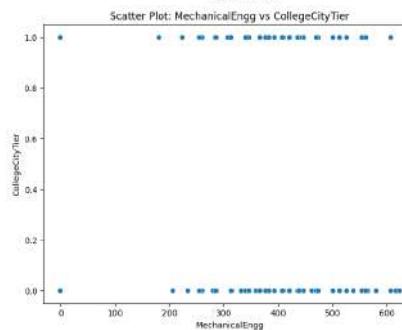
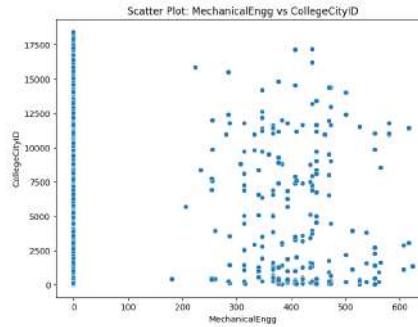


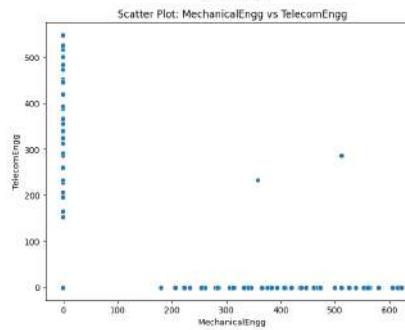
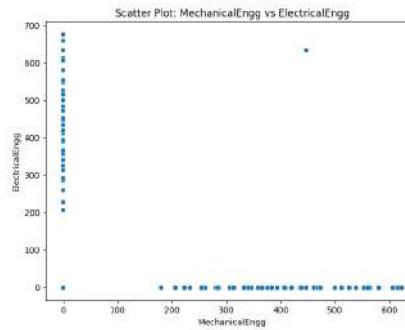
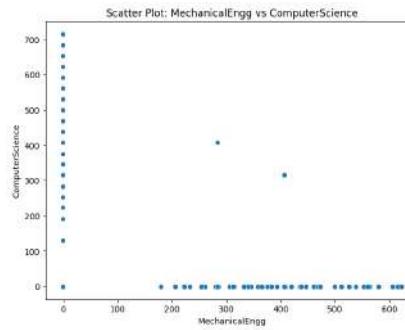
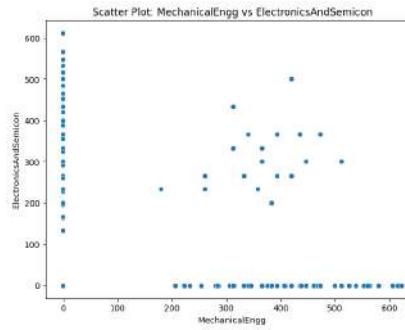
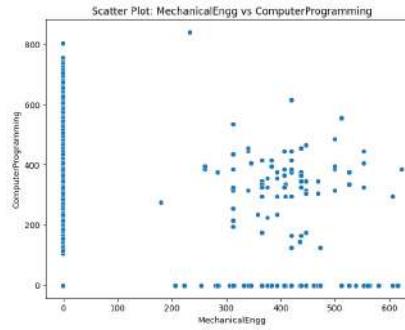
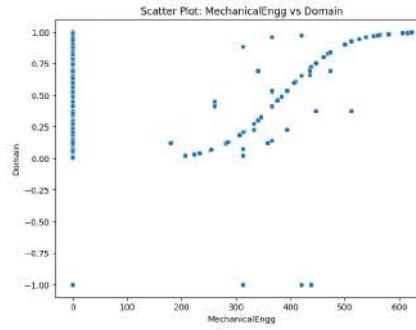


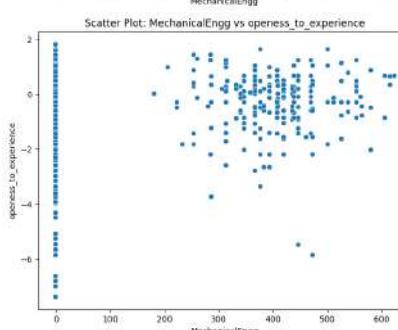
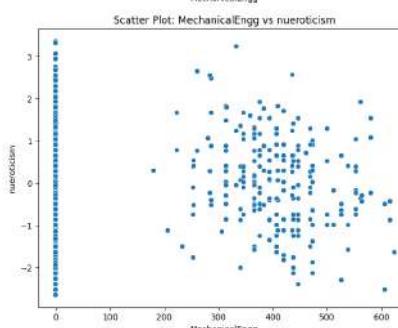
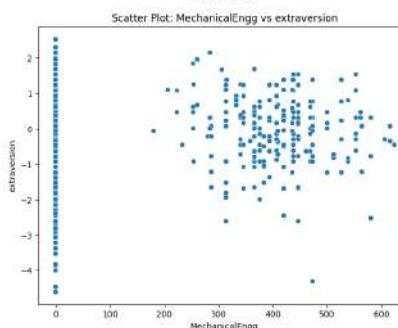
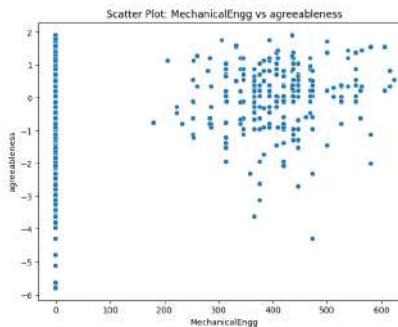
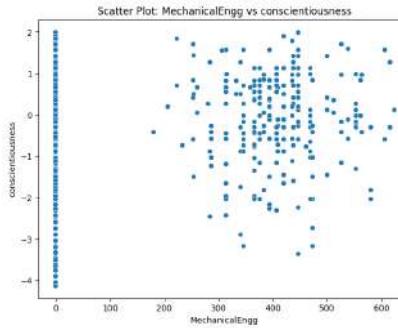
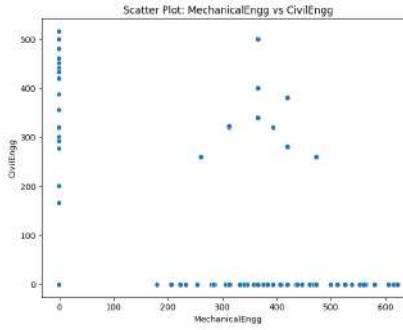


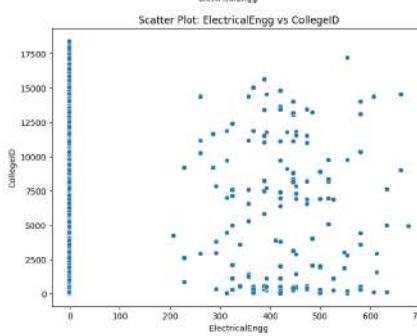
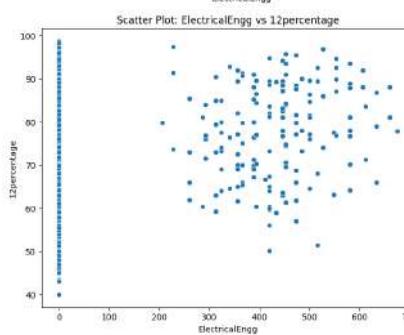
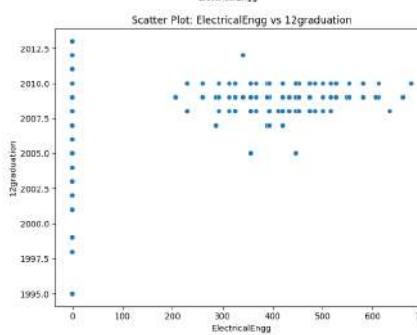
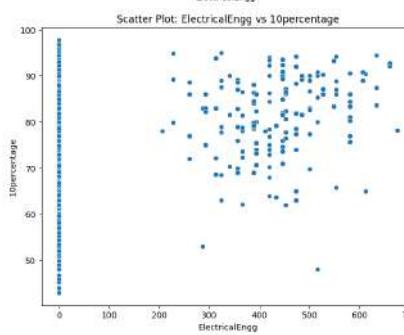
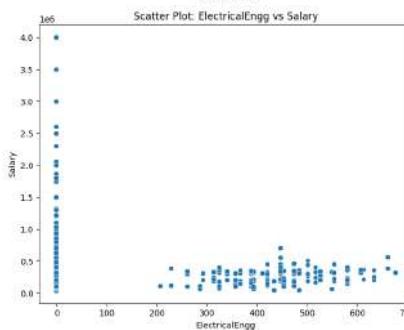
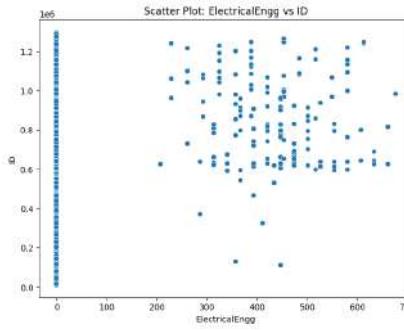


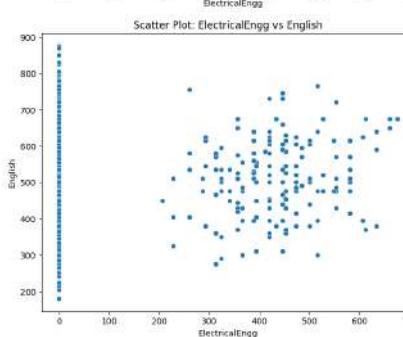
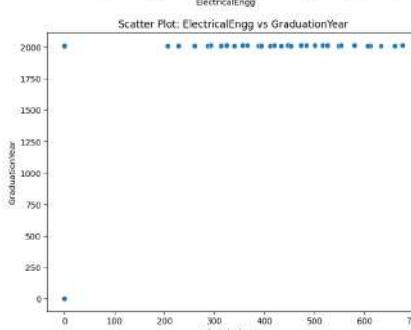
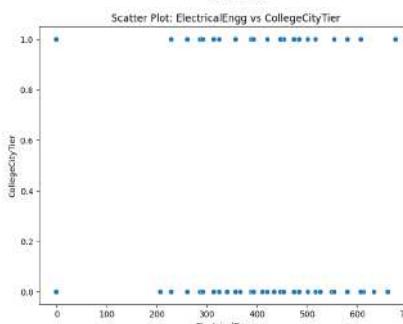
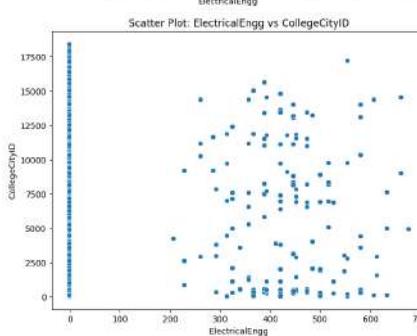
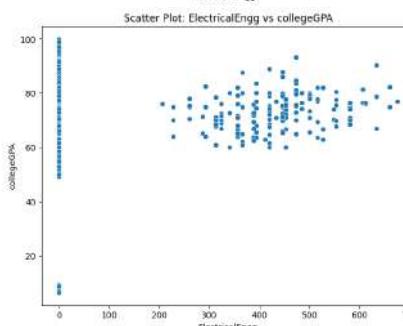
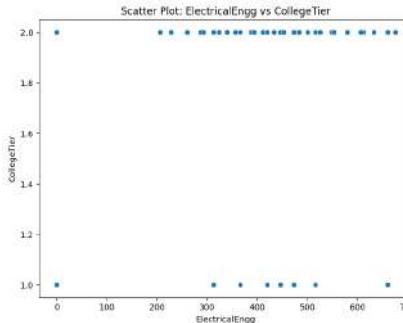


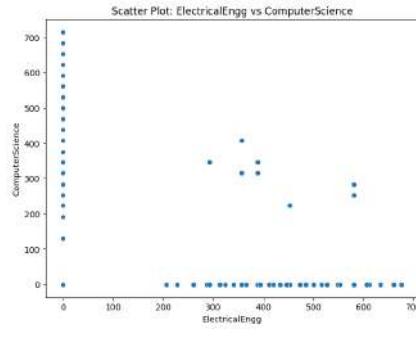
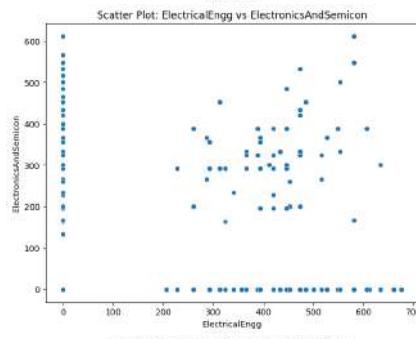
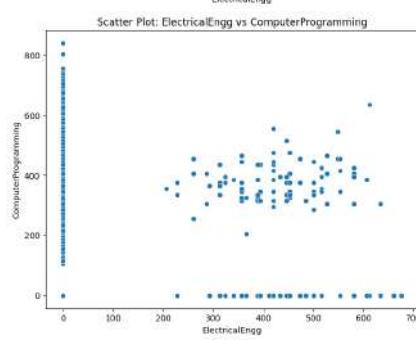
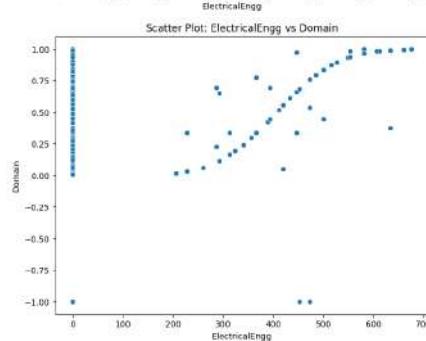
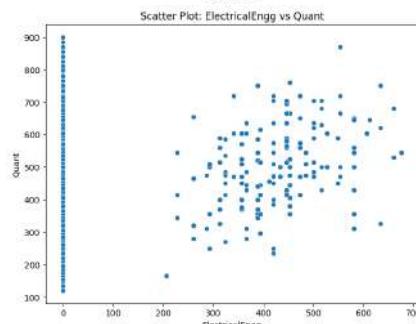
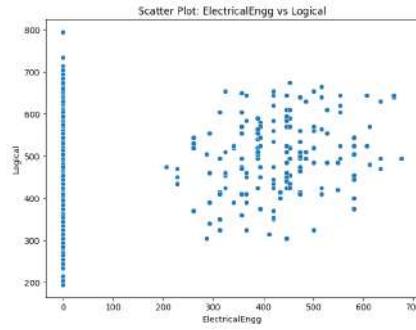


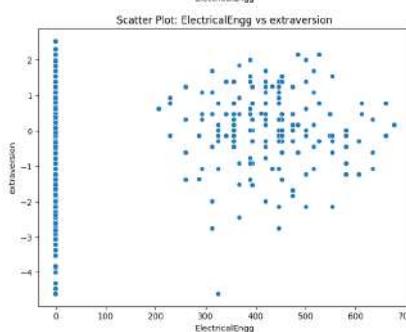
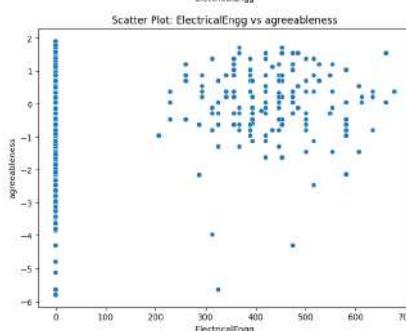
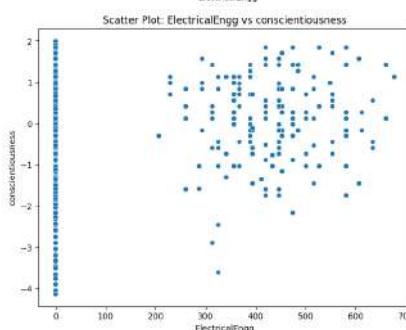
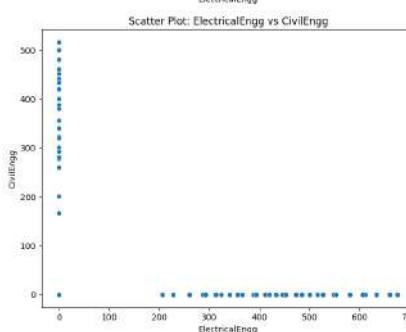
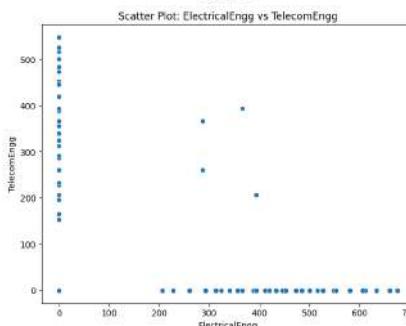
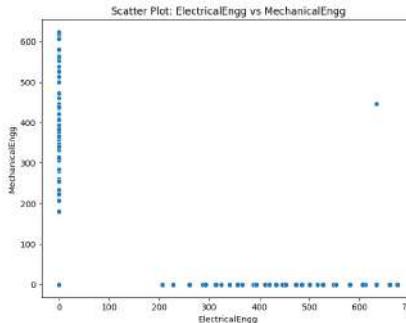


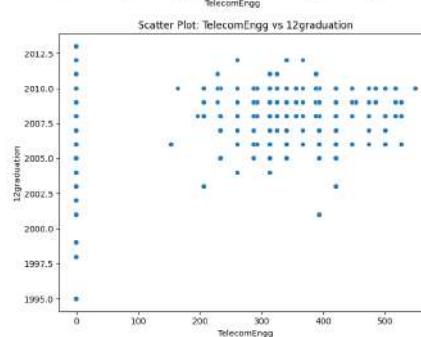
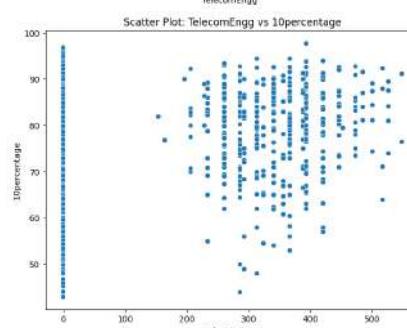
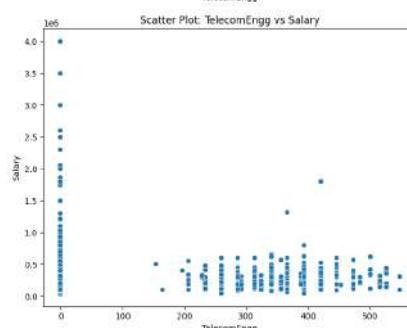
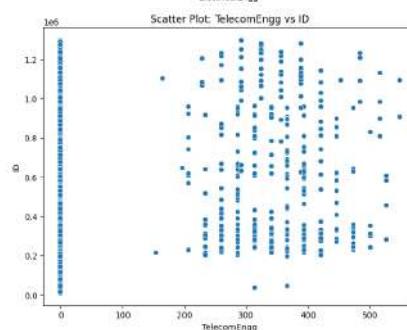
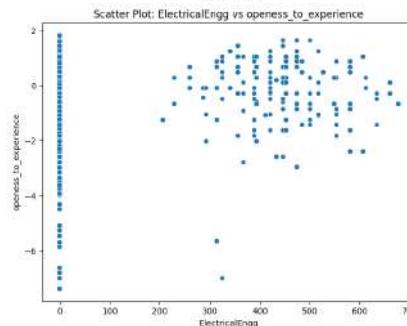
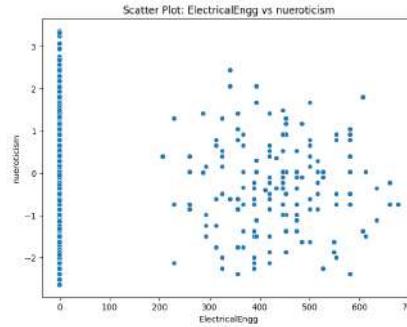


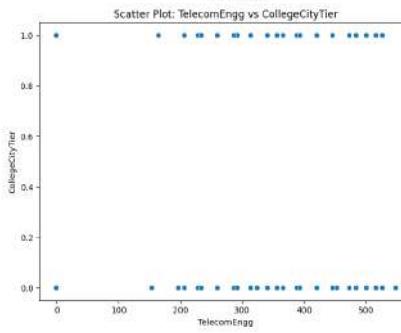
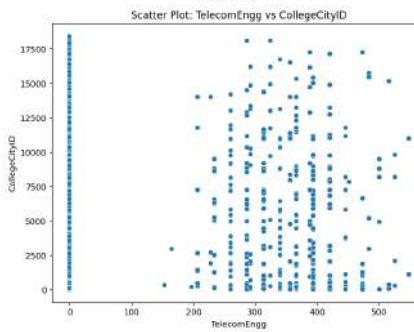
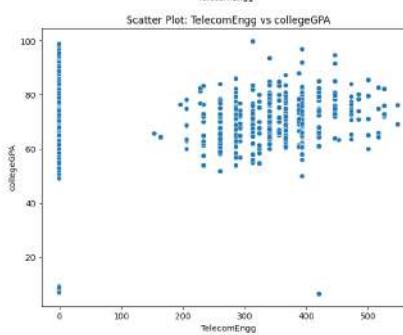
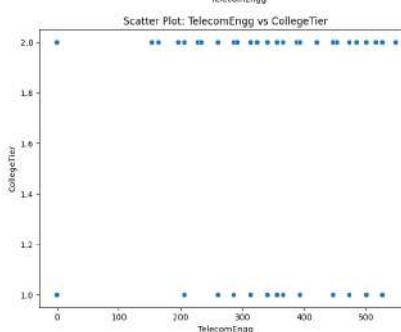
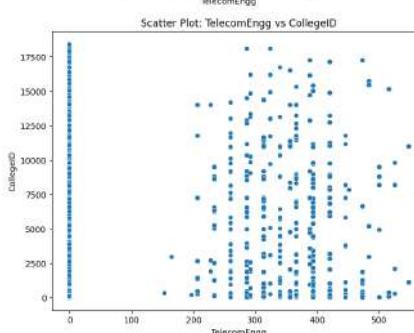
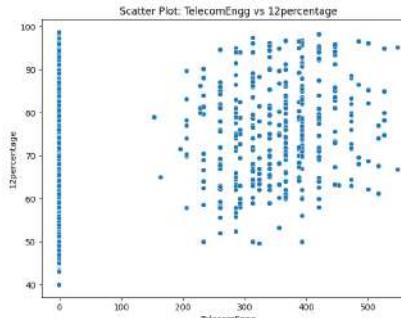


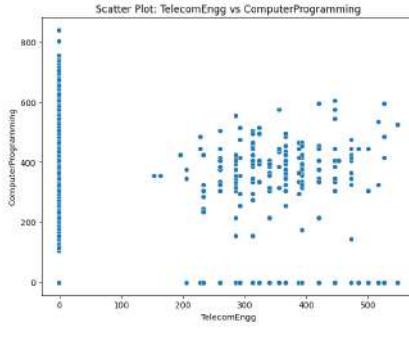
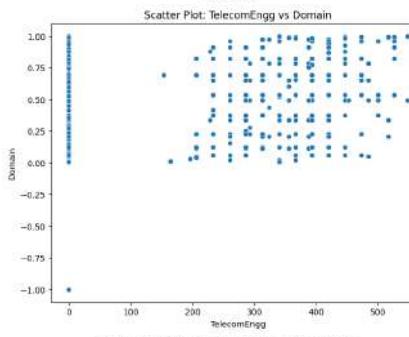
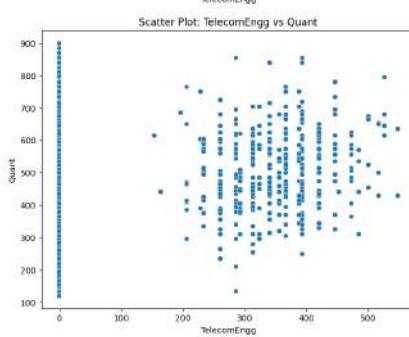
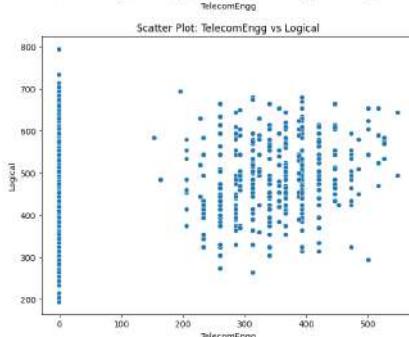
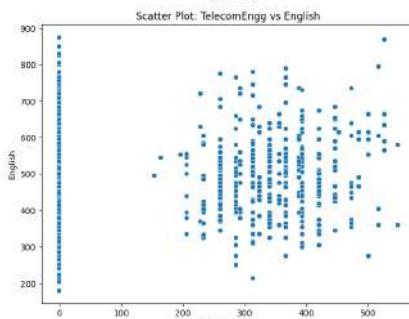
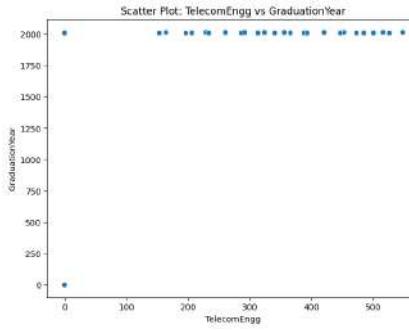


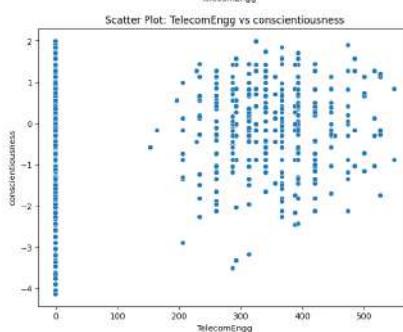
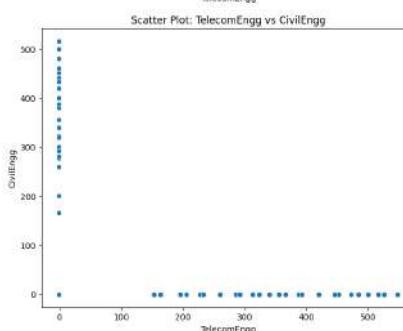
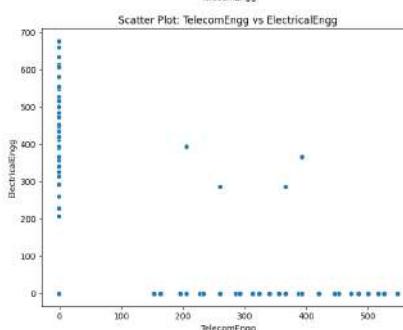
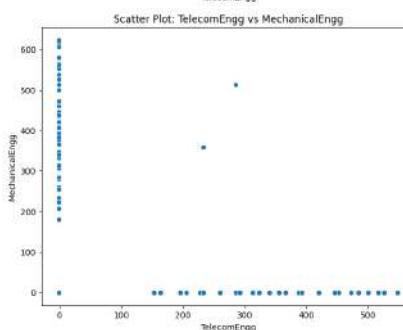
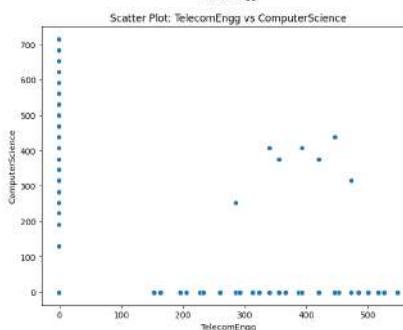
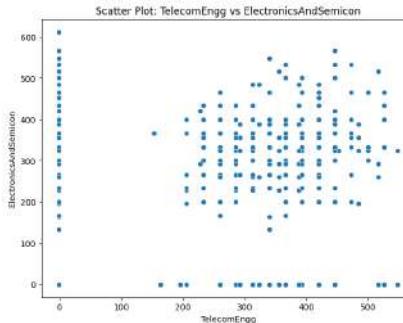


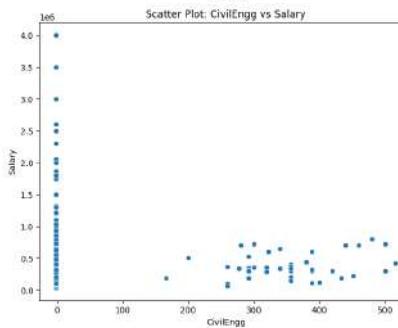
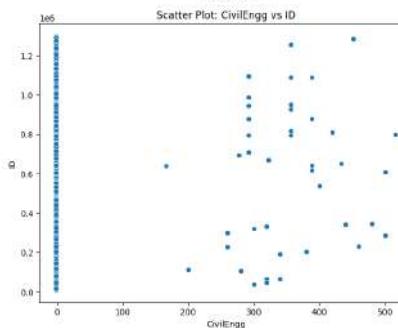
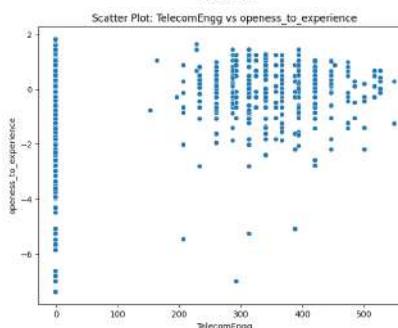
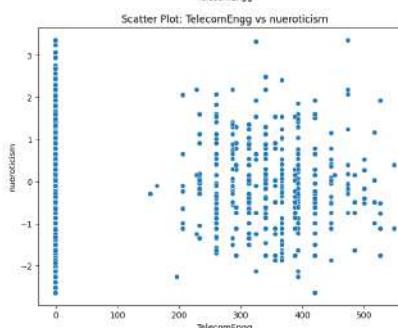
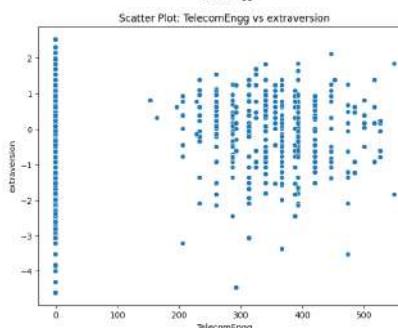
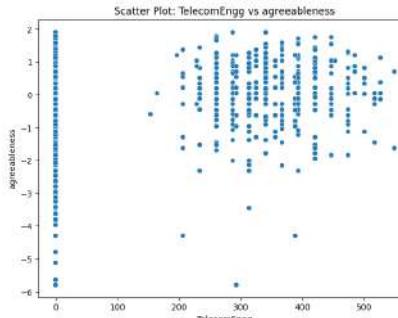


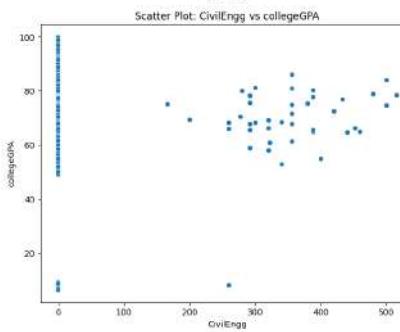
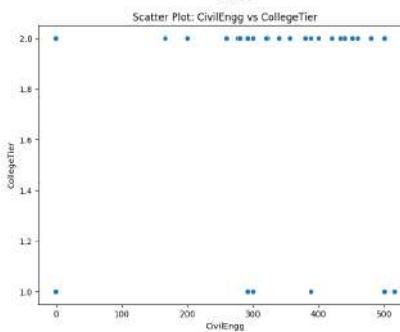
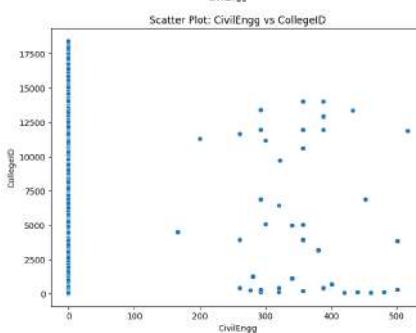
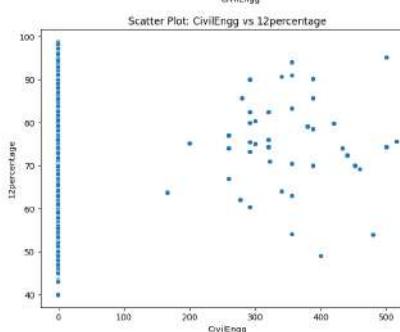
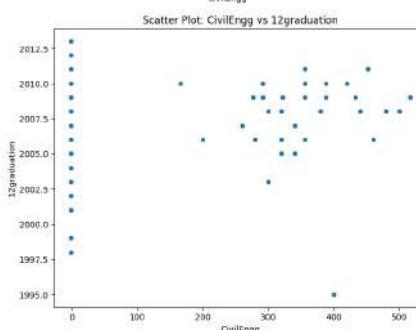
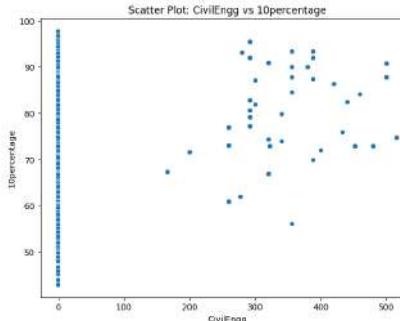


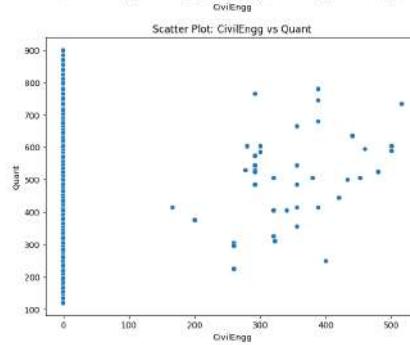
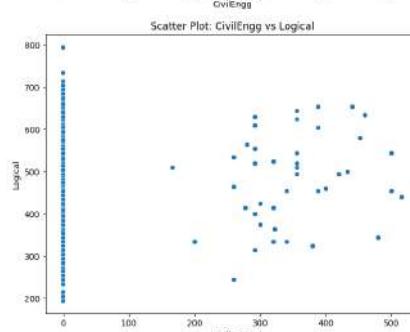
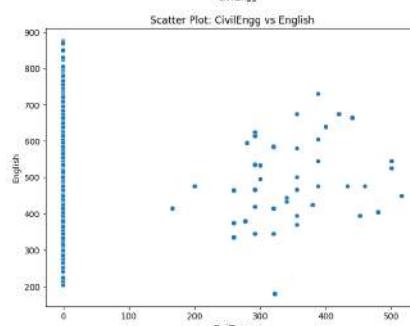
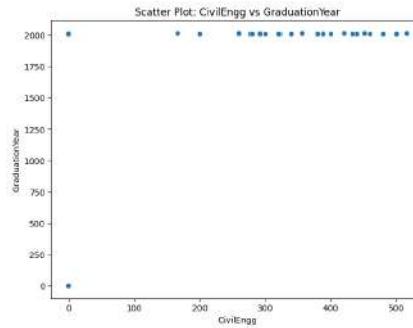
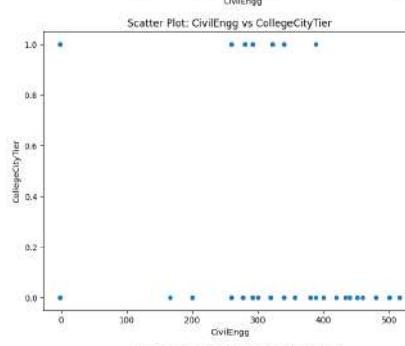
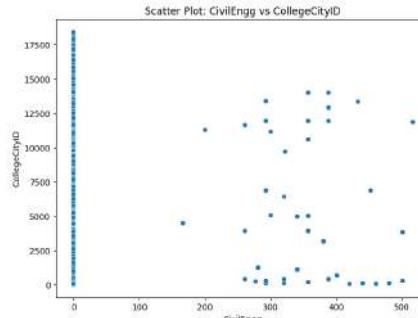


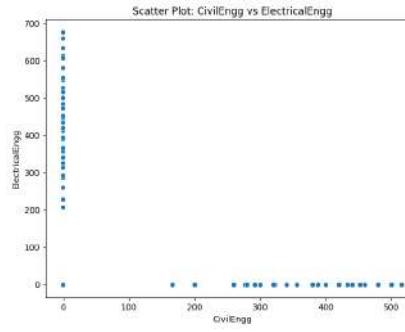
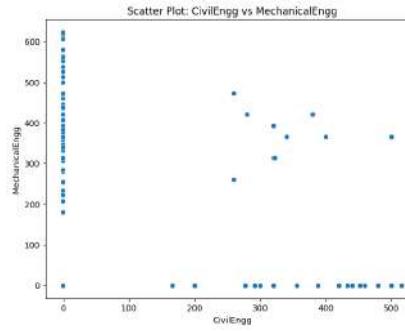
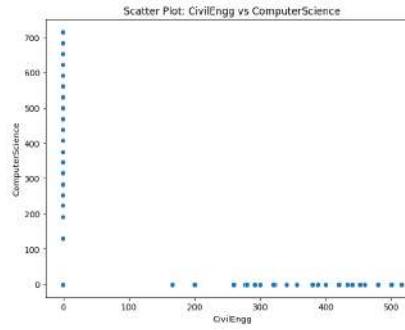
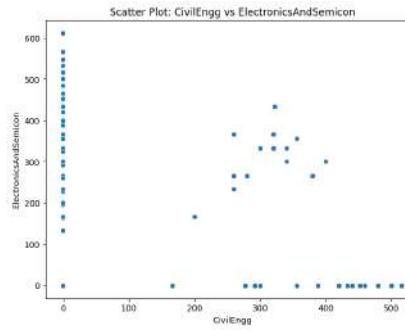
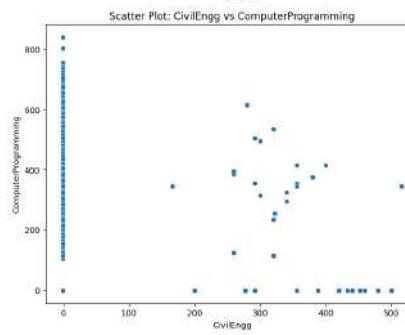
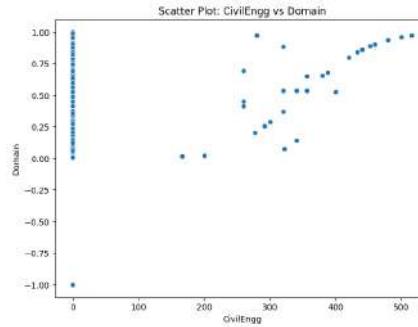


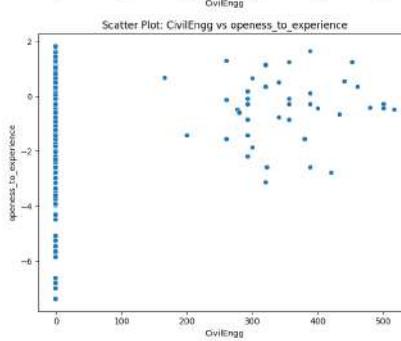
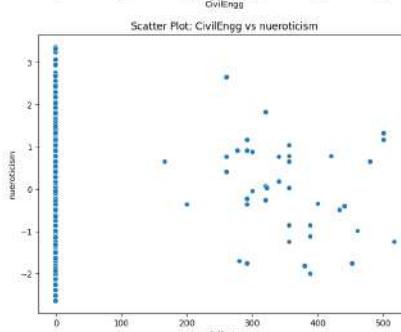
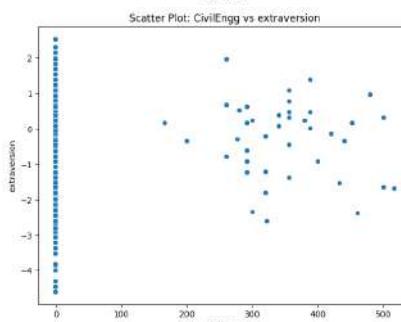
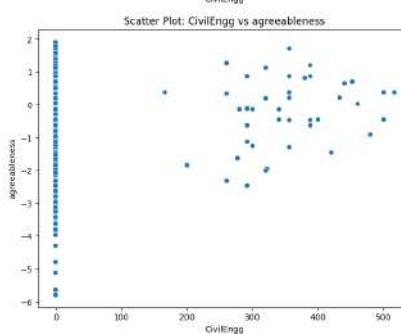
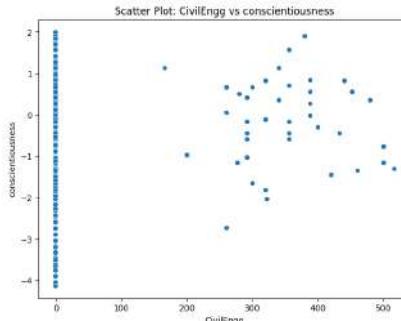
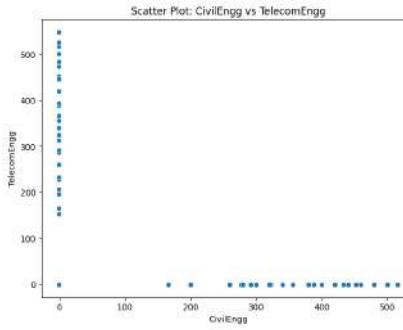


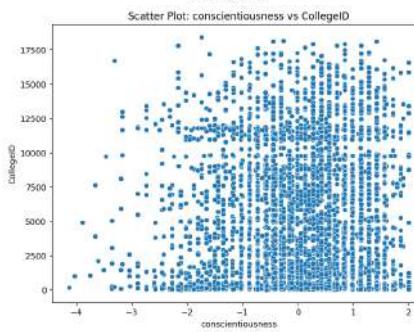
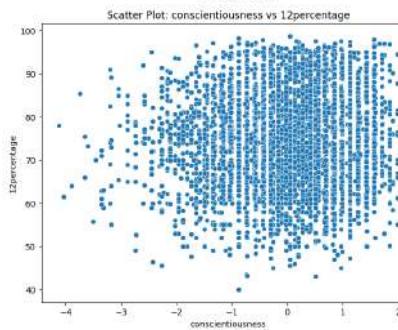
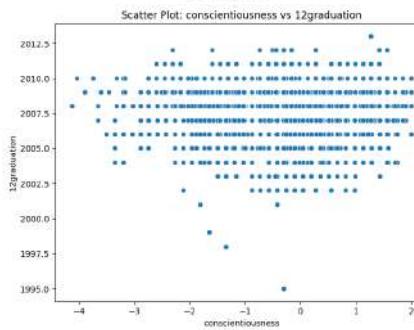
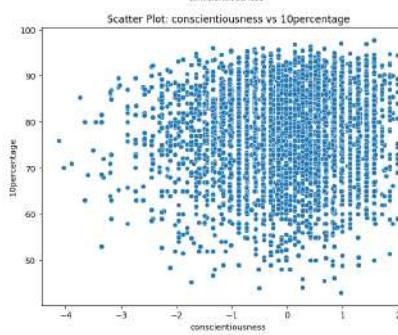
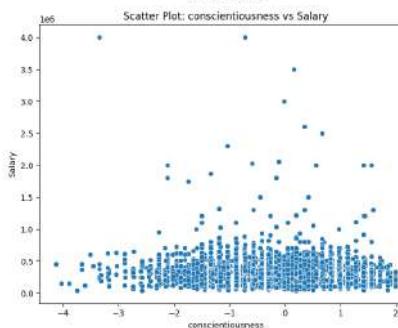
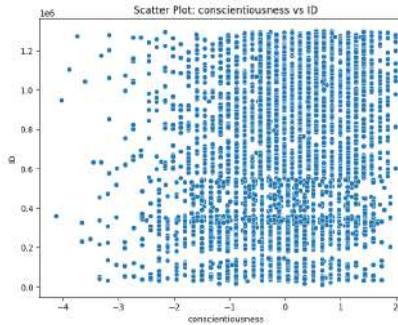


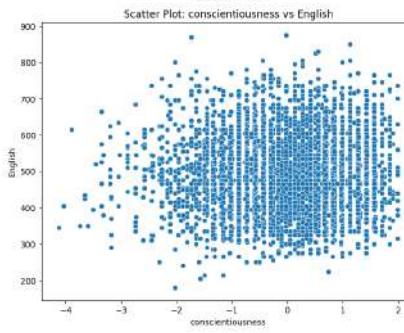
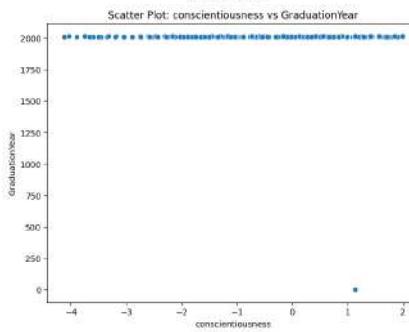
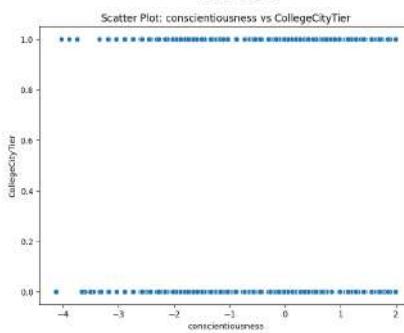
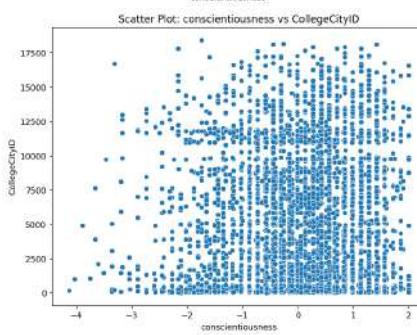
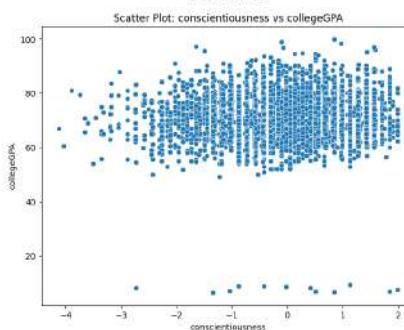
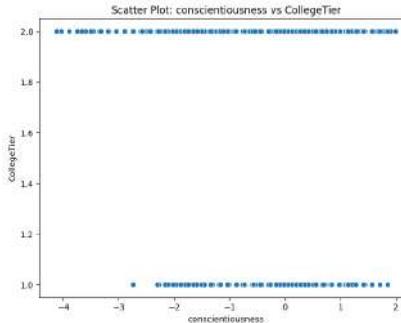


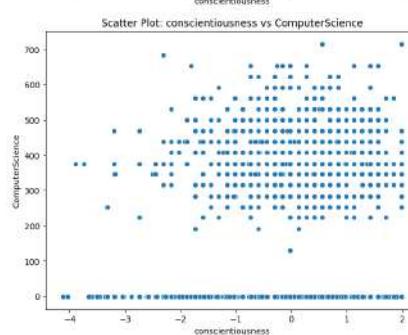
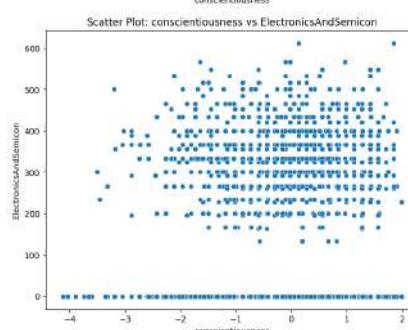
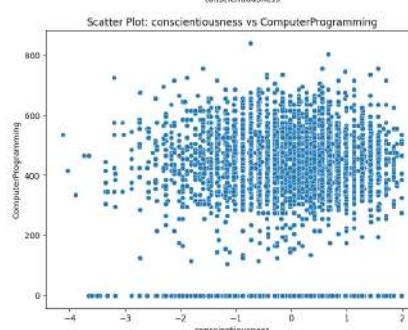
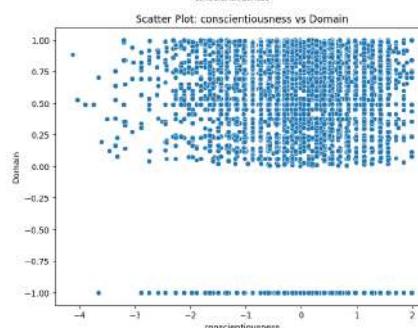
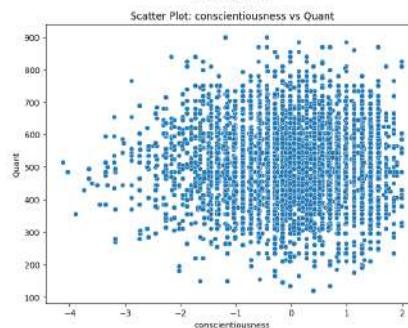
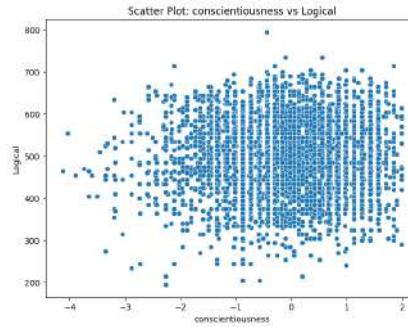


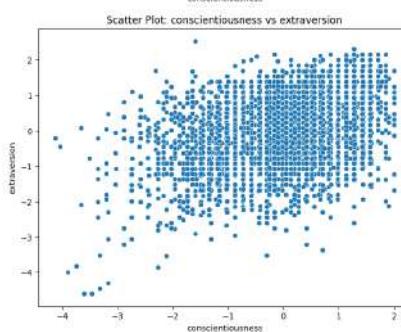
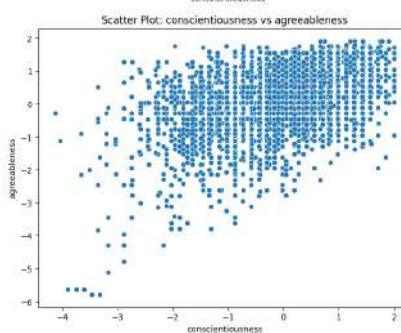
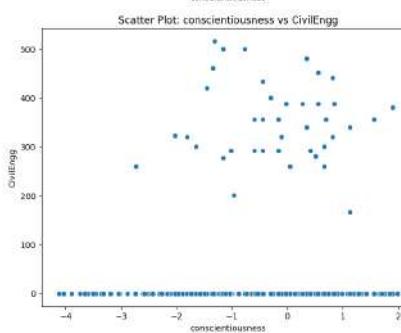
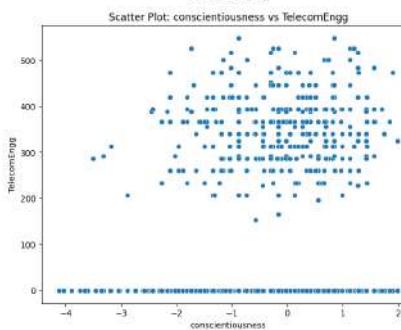
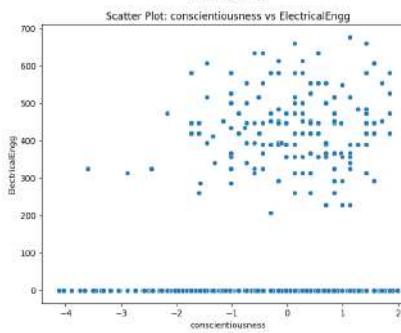
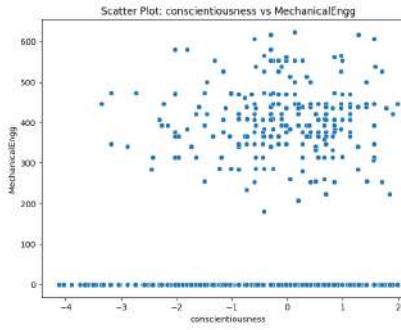


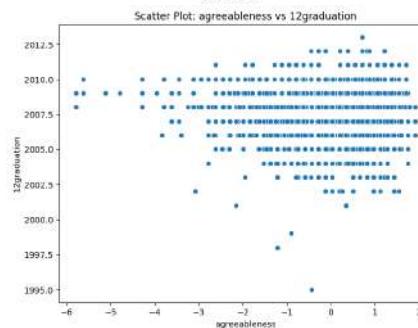
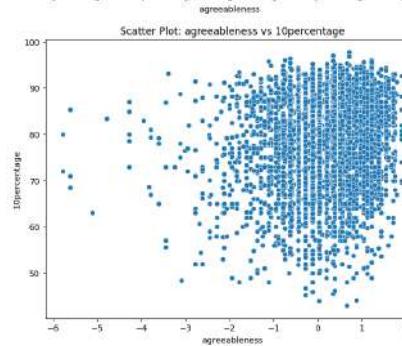
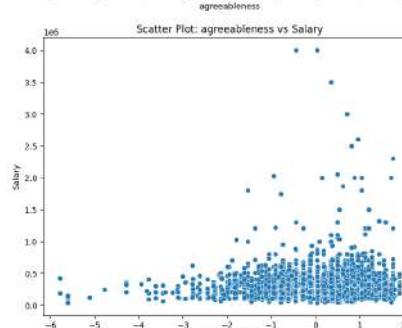
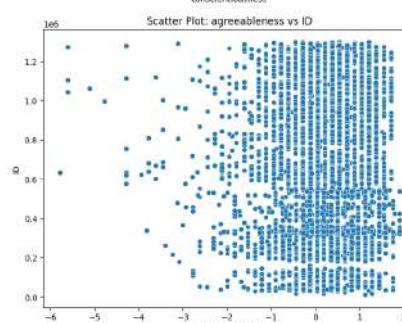
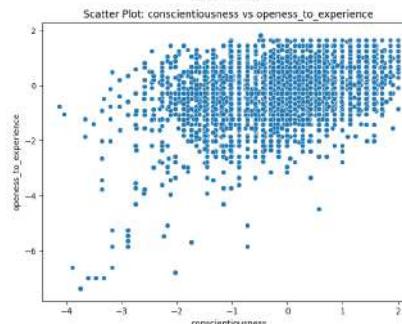
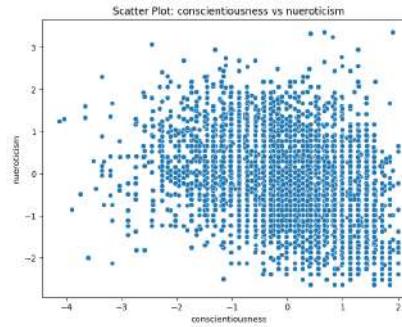


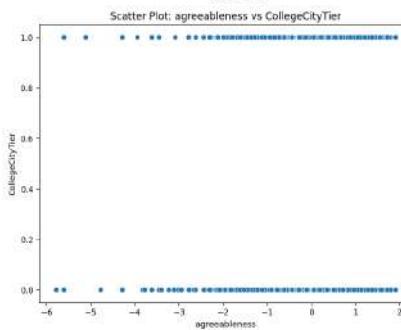
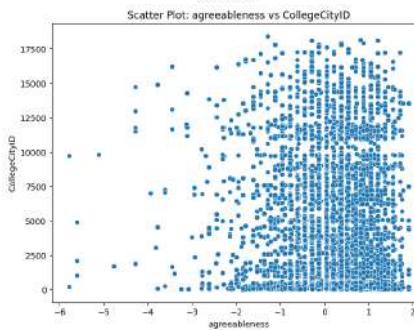
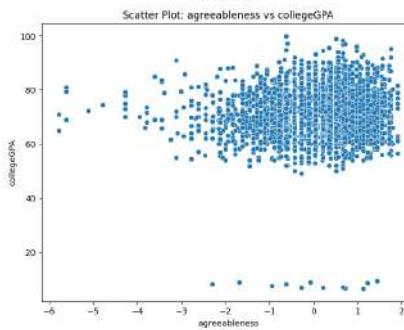
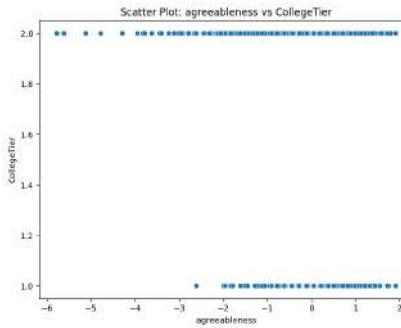
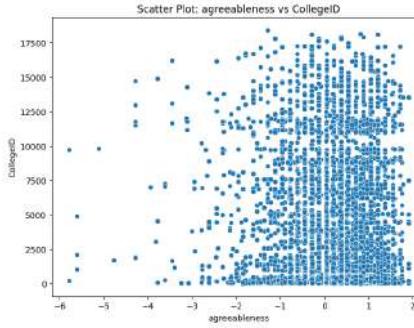
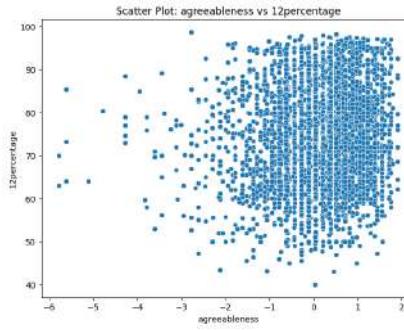


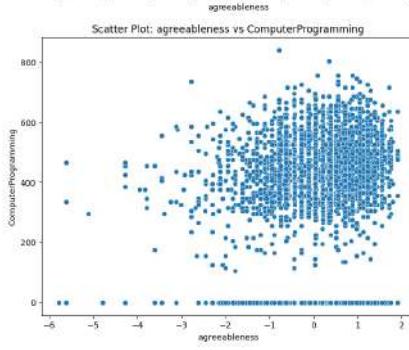
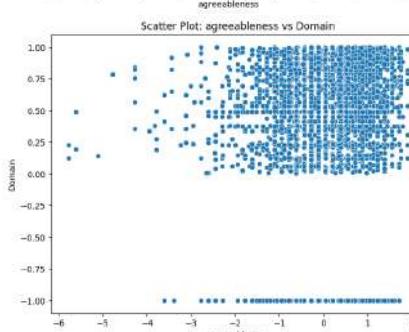
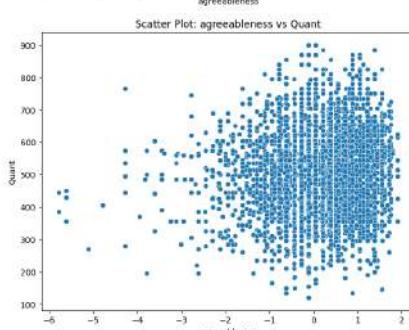
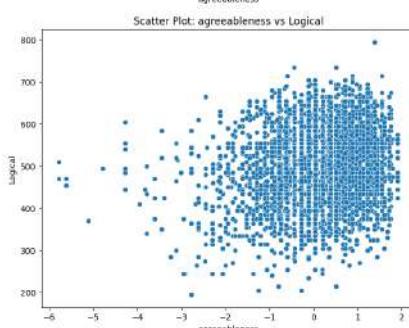
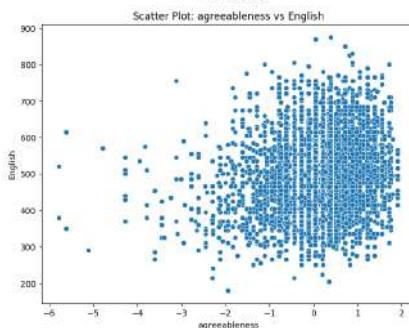
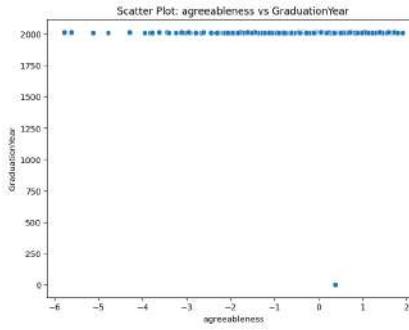


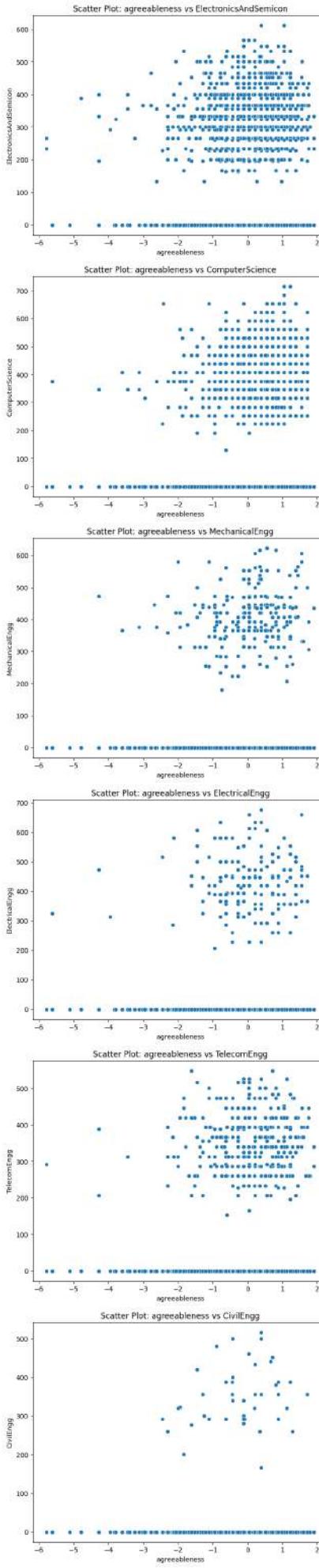


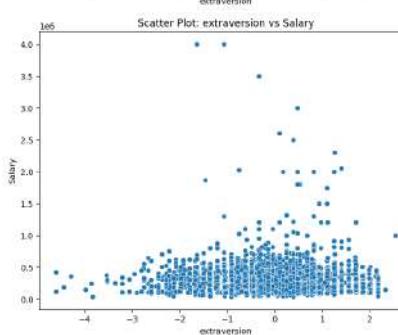
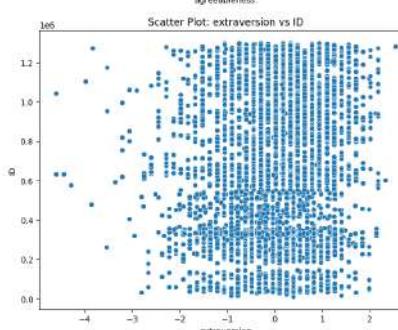
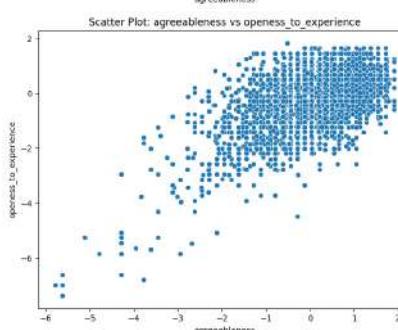
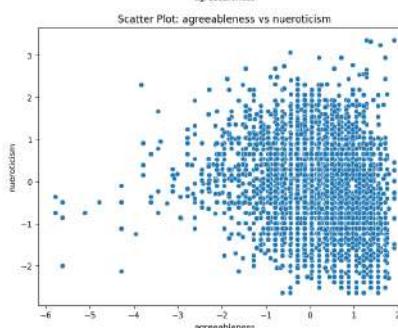
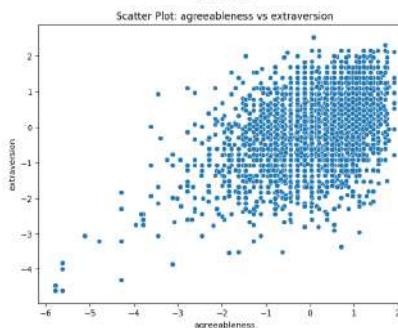
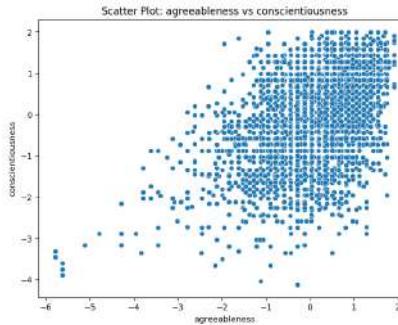


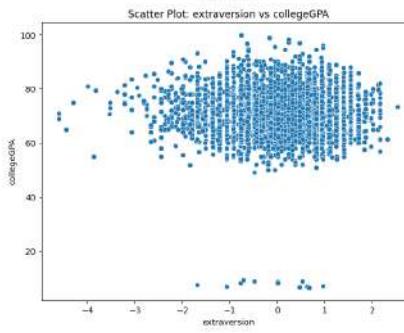
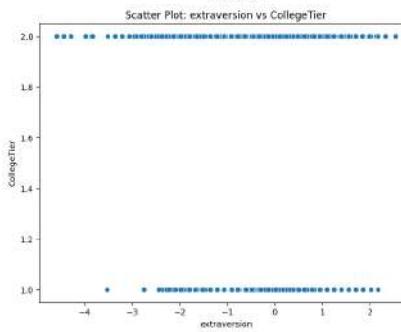
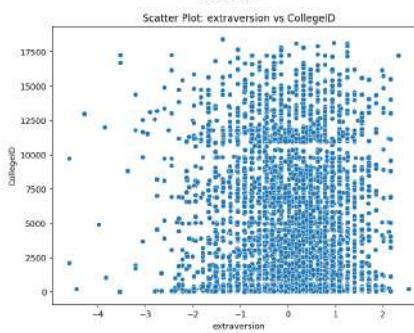
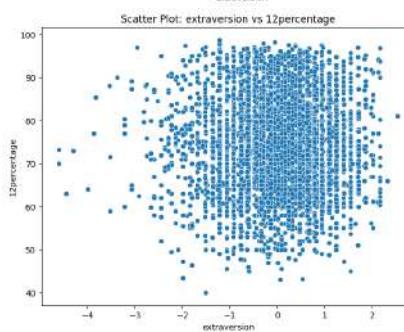
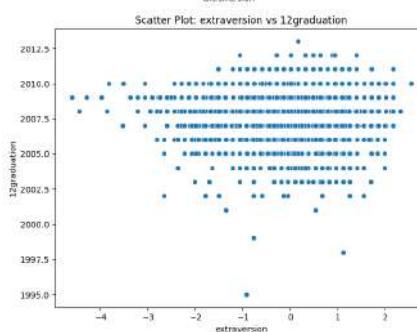
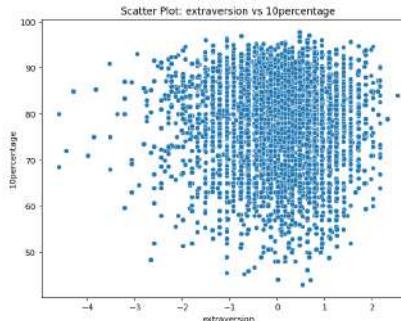


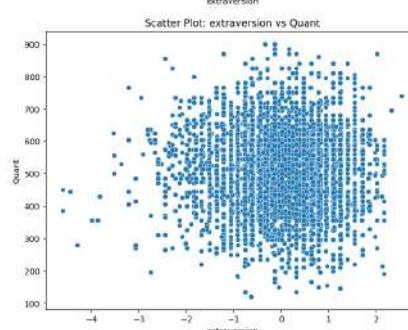
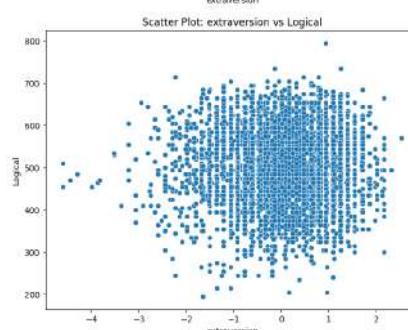
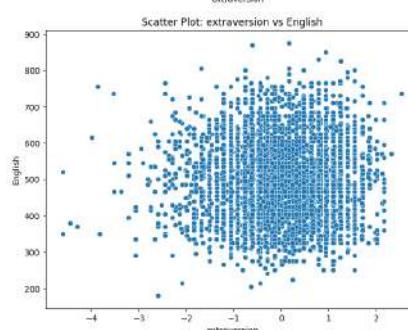
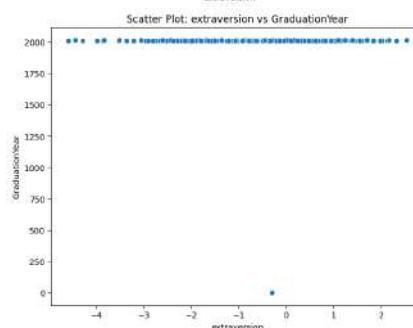
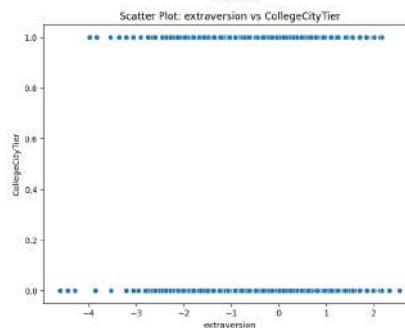
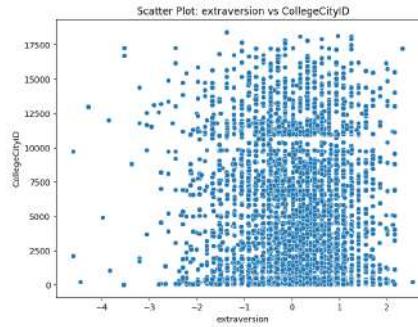


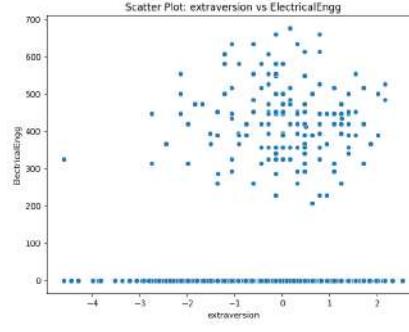
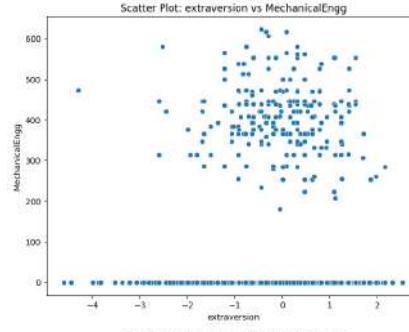
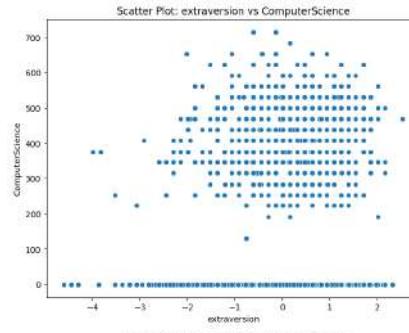
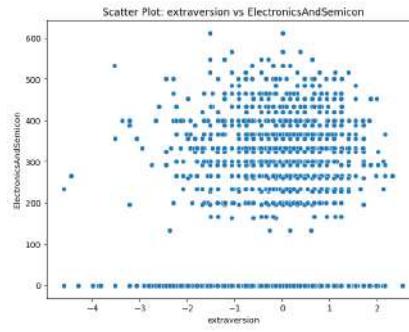
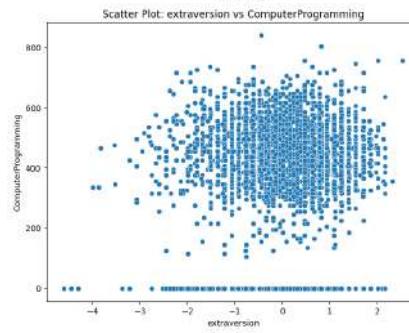
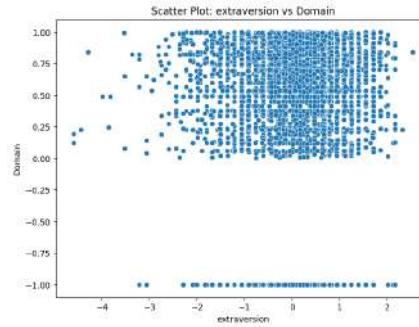


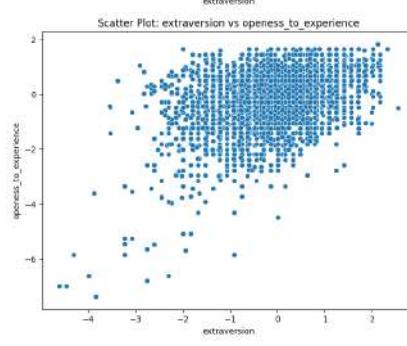
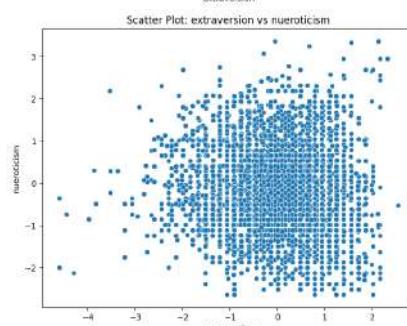
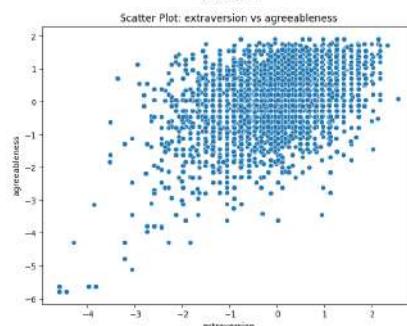
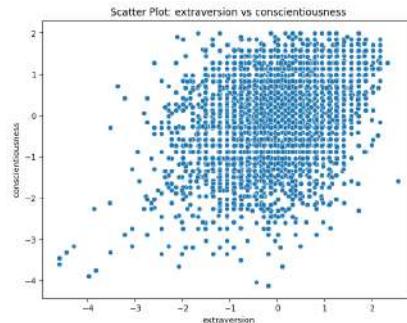
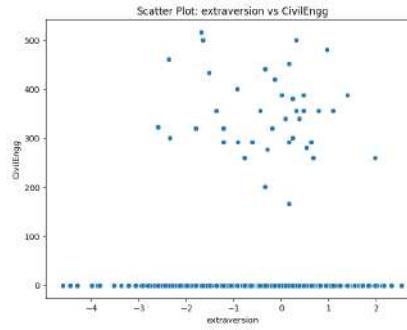
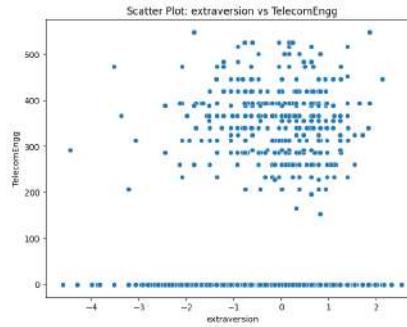


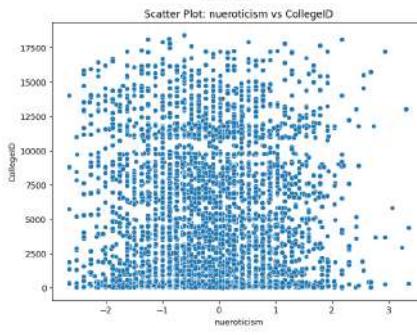
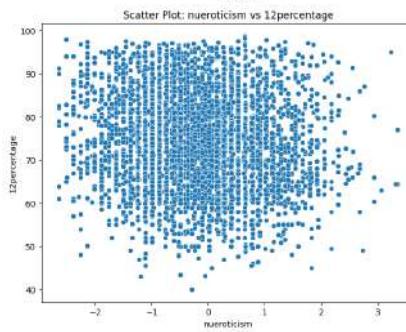
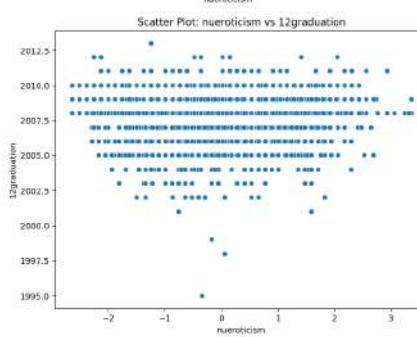
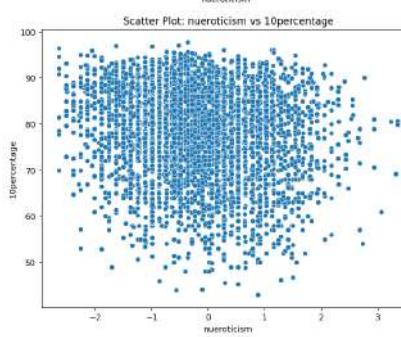
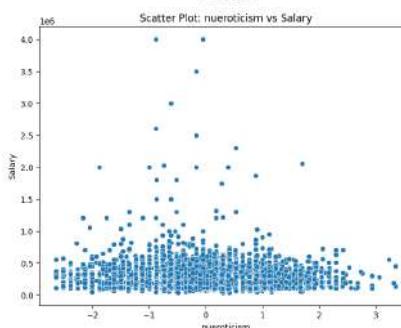
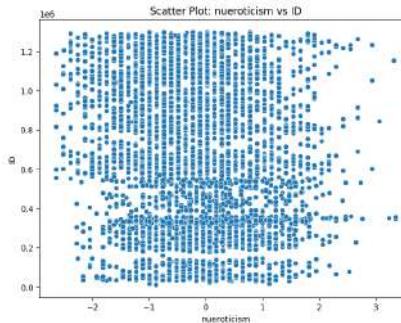


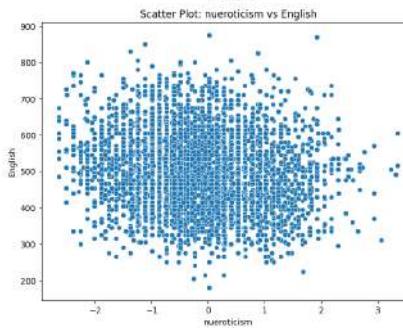
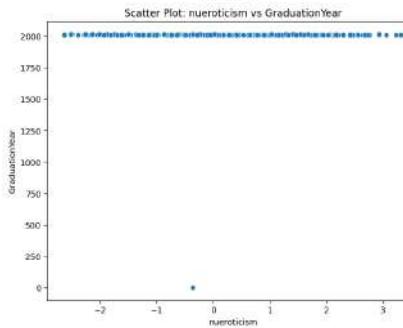
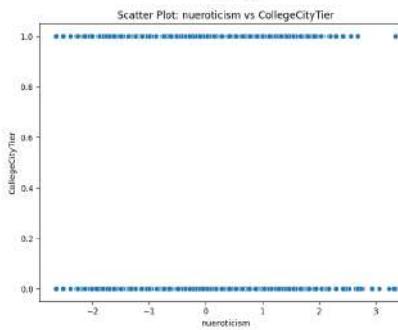
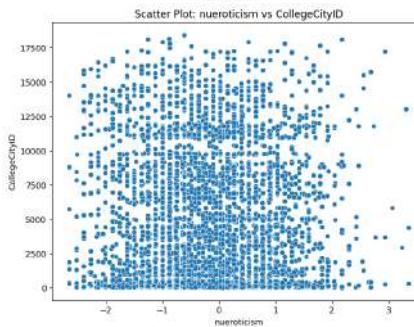
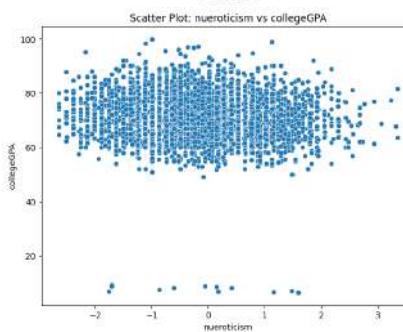
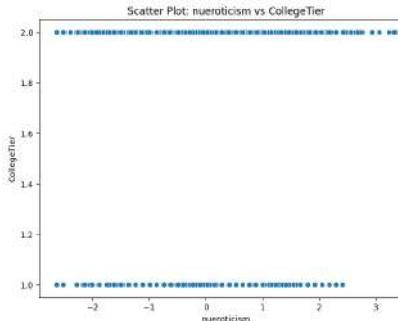


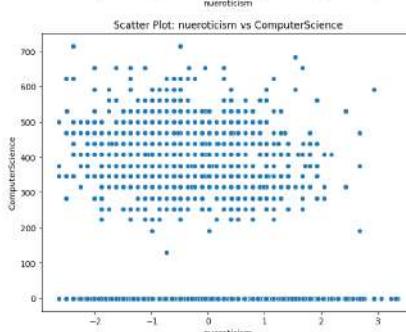
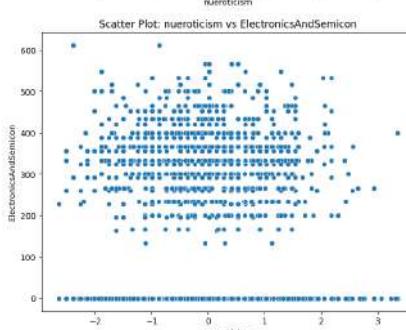
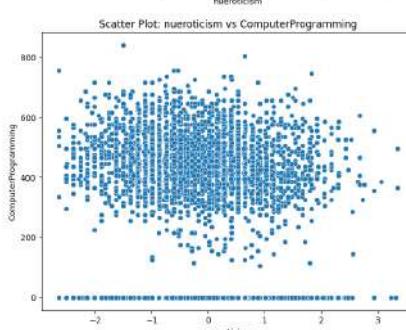
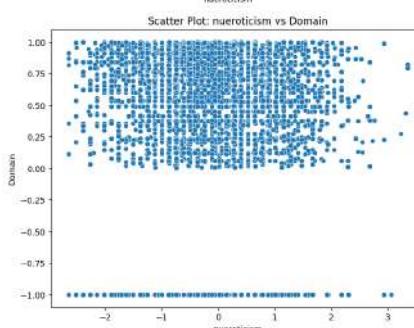
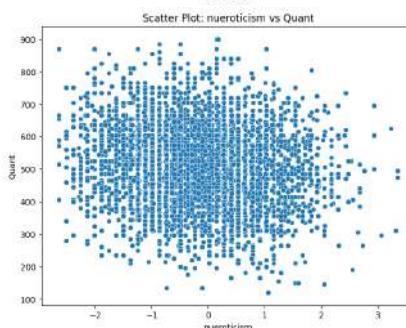
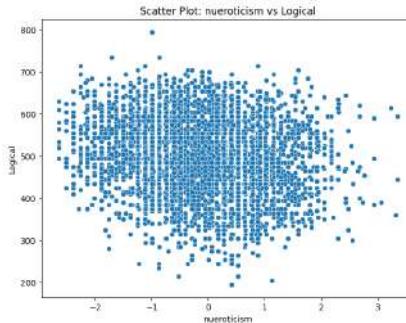


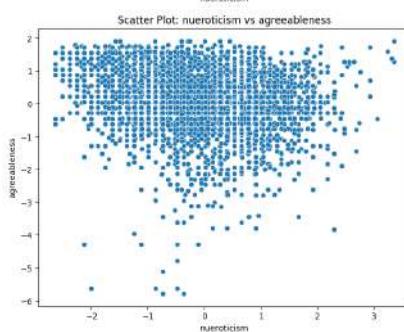
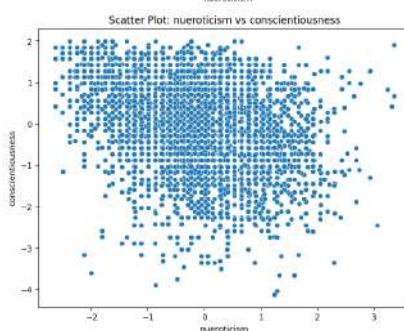
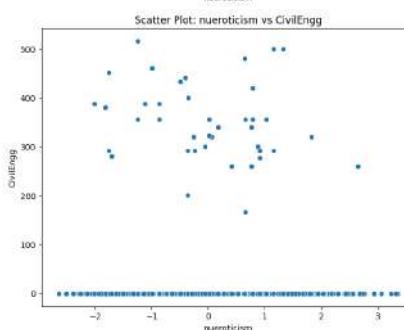
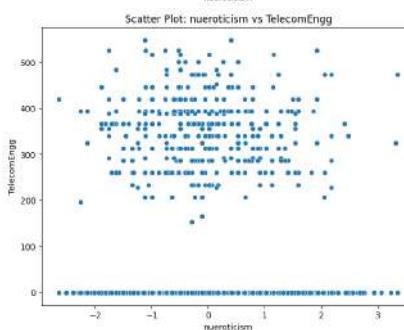
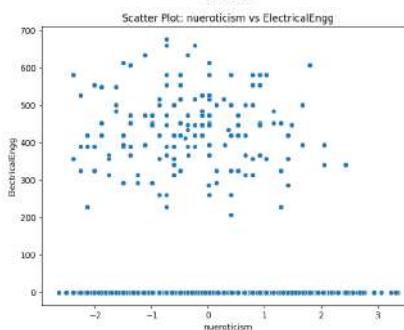
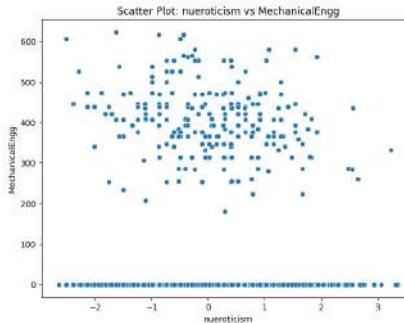


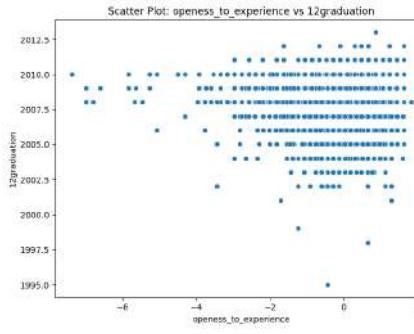
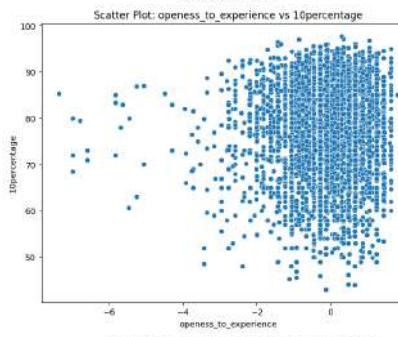
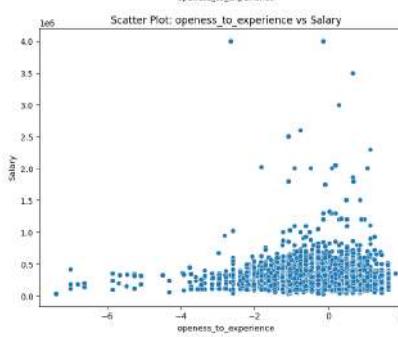
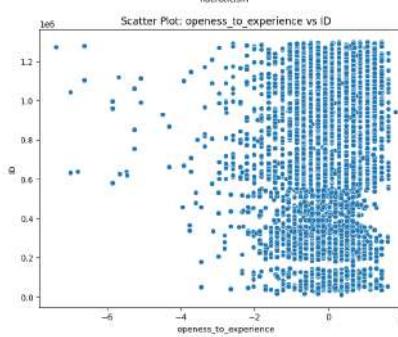
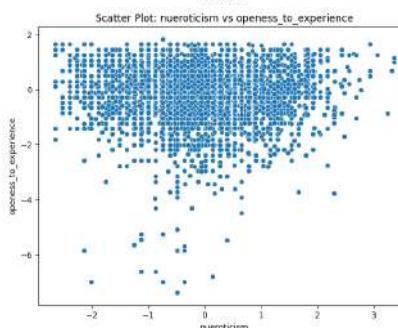
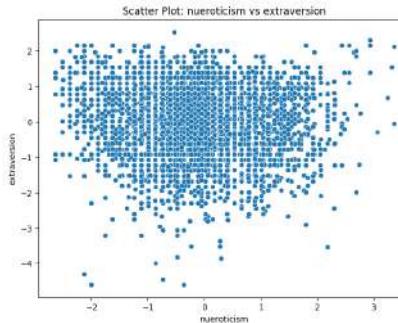


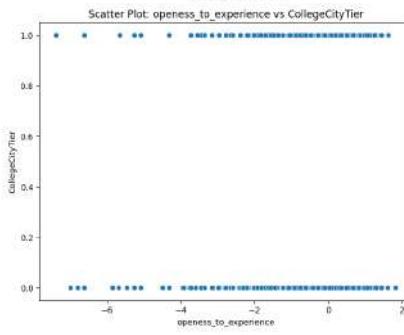
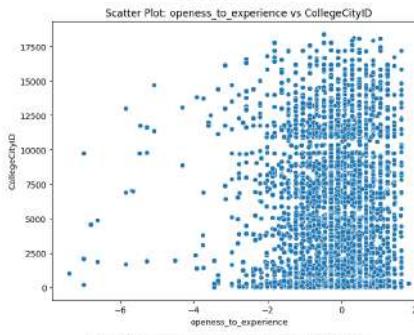
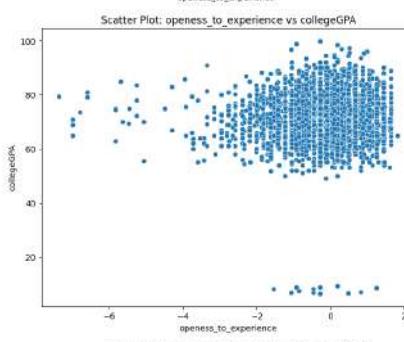
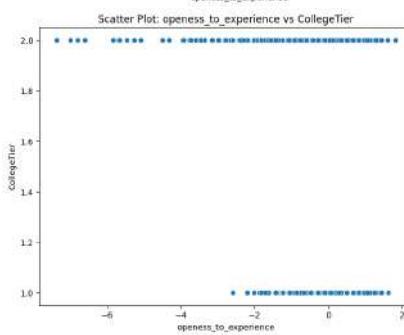
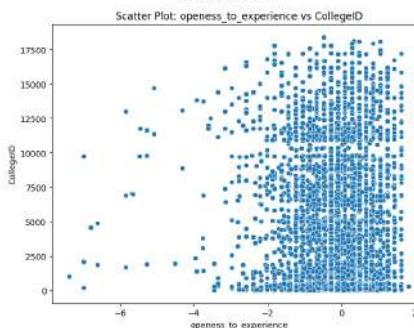
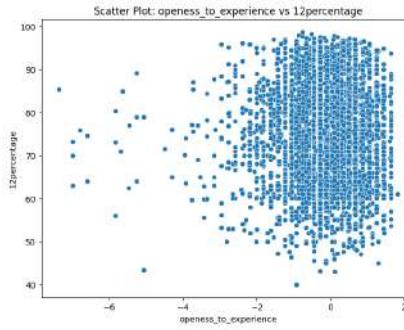


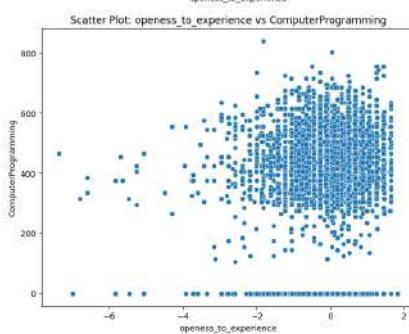
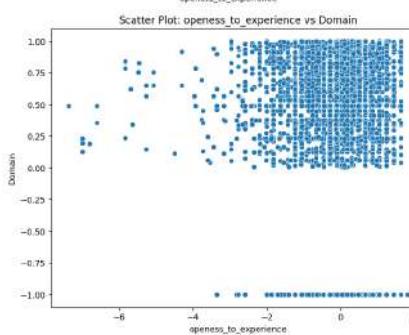
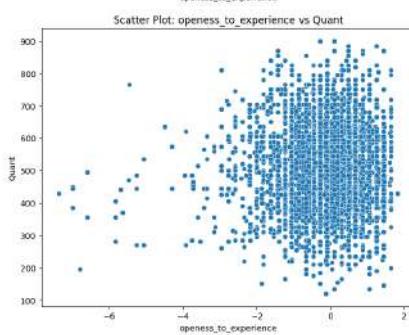
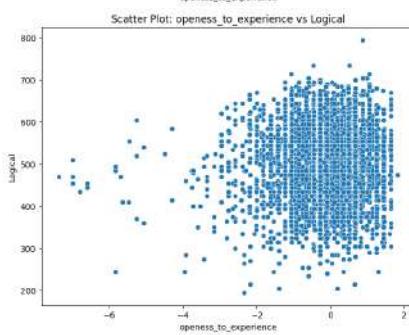
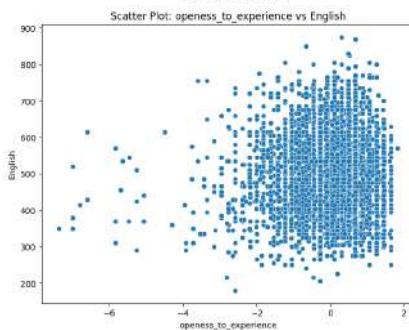
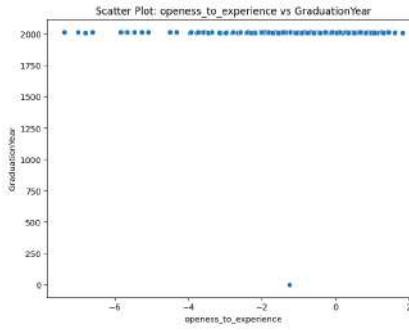


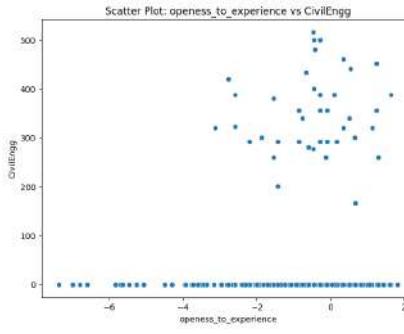
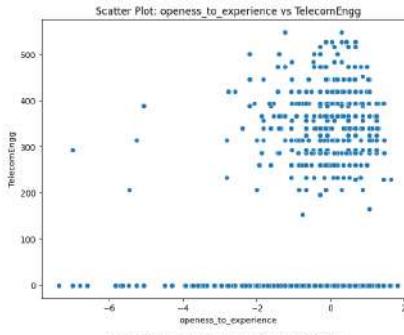
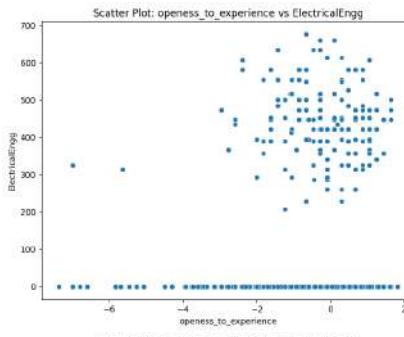
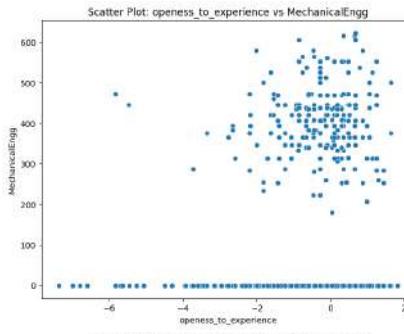
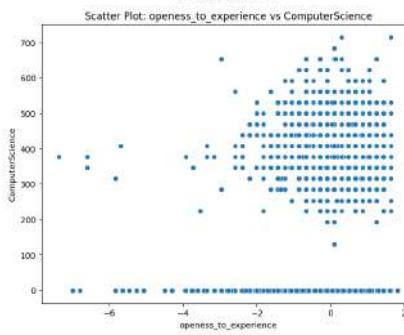
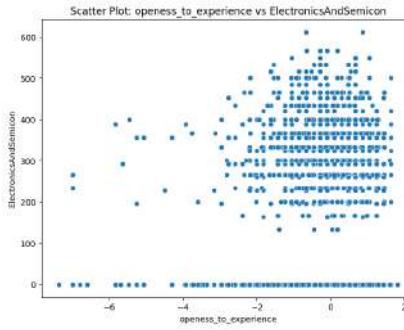


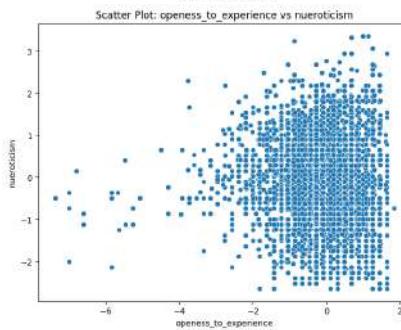
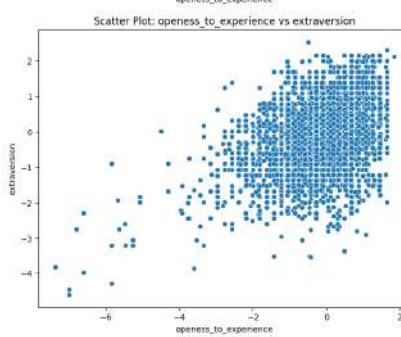
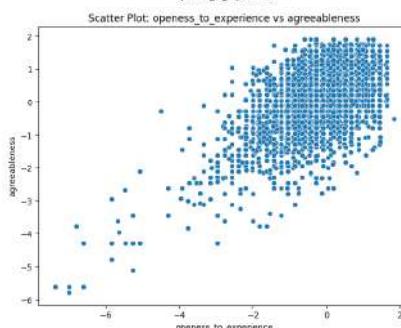
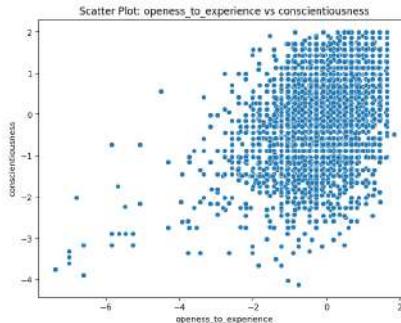






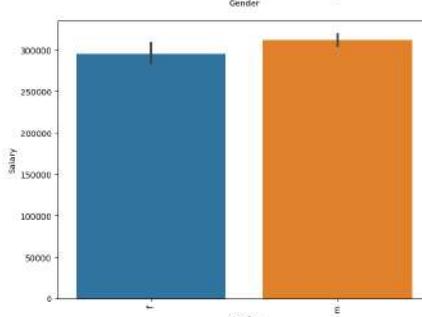
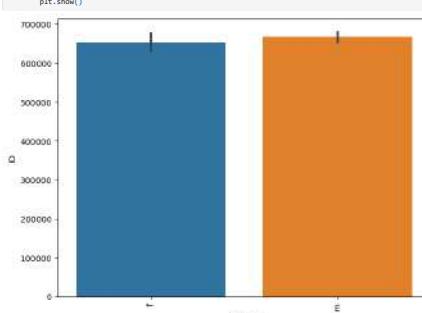


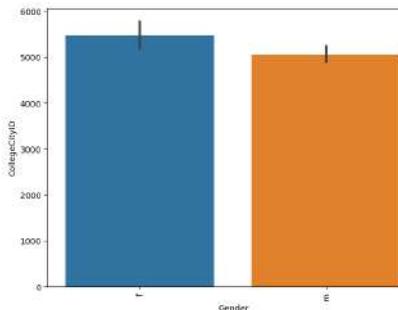
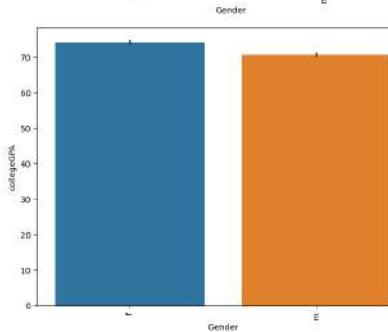
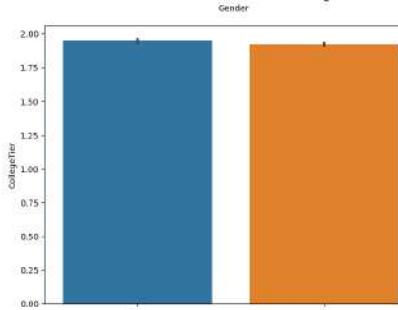
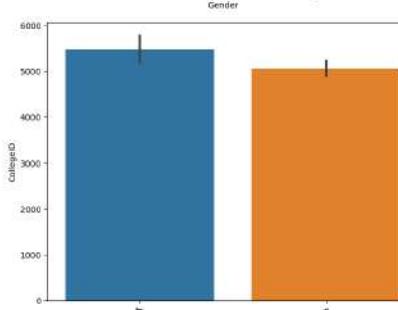
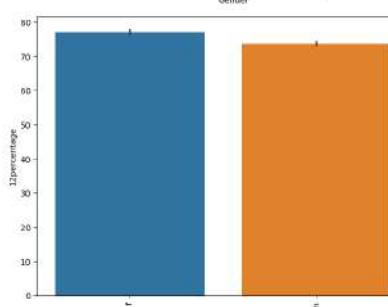
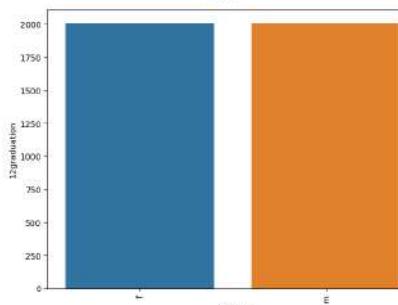
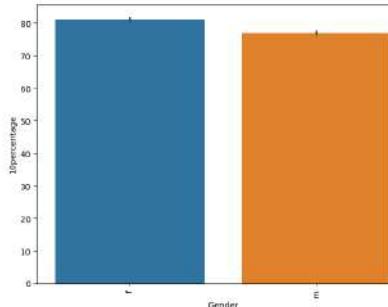


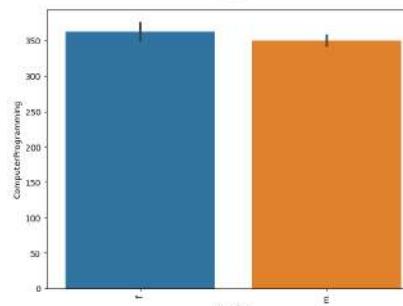
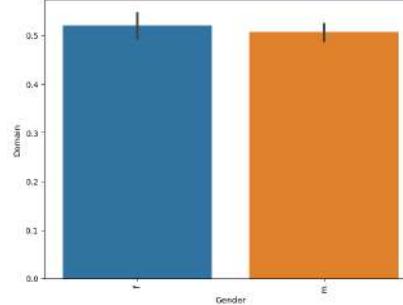
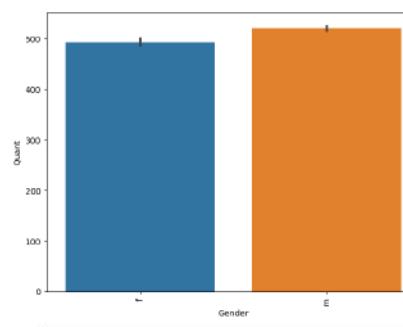
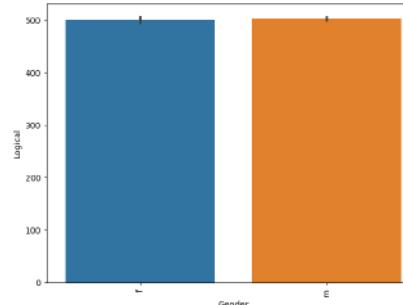
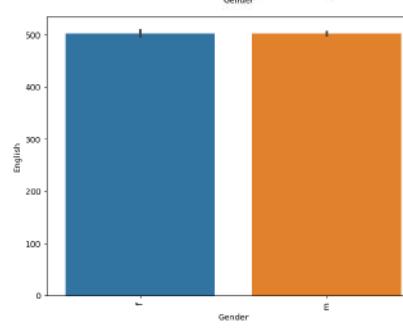
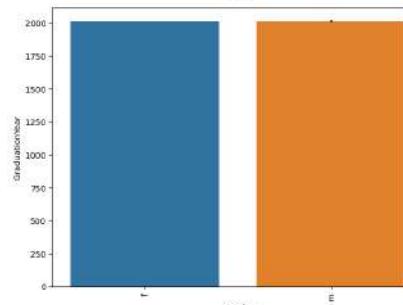
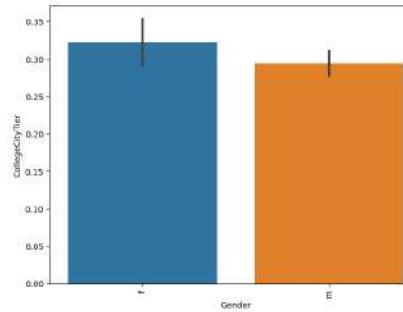


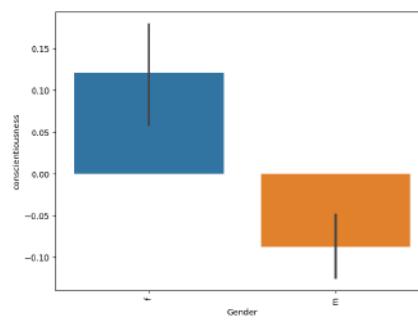
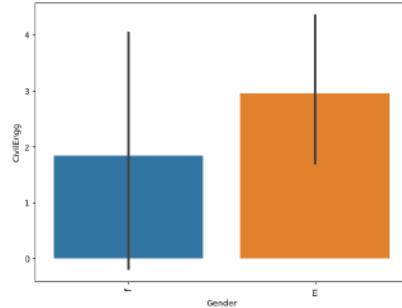
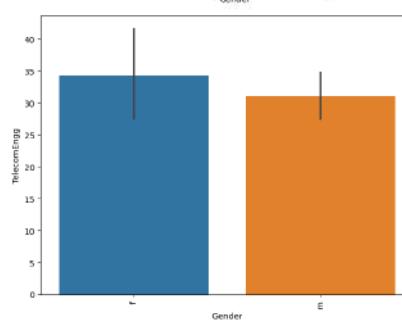
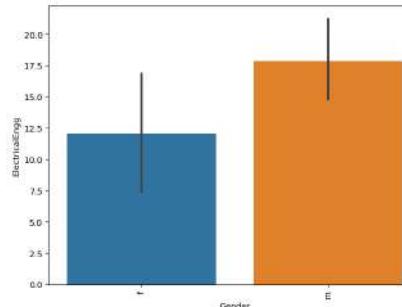
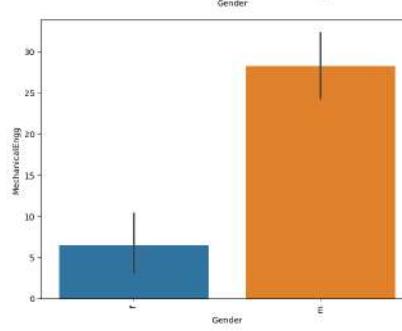
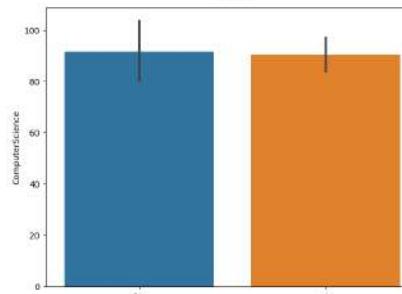
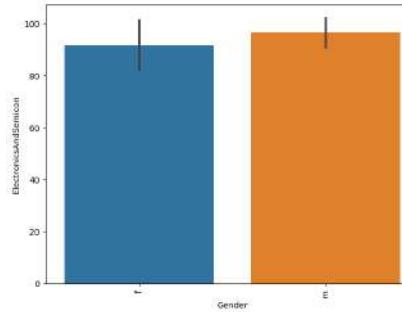
Categorical vs Numerical

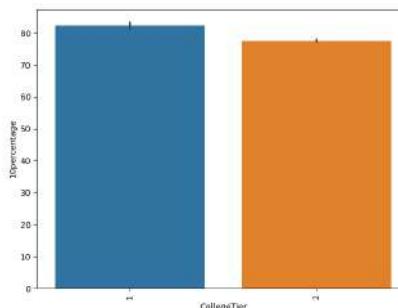
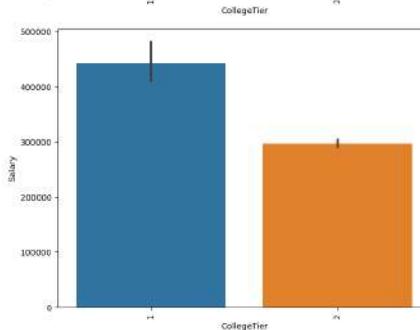
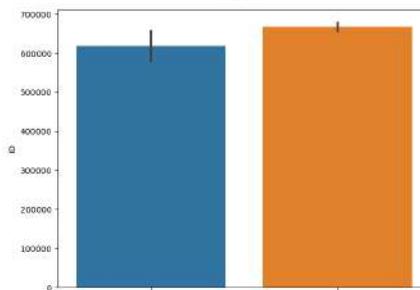
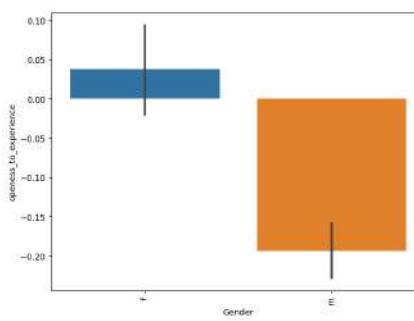
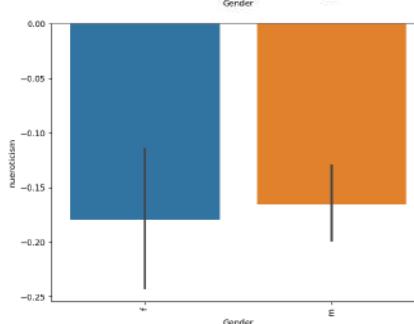
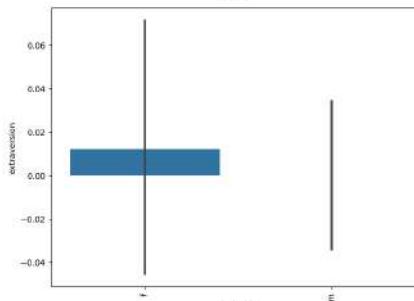
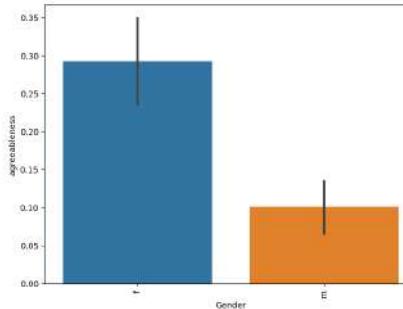
```
In [18]: #barplot
for cat_col in categorical_cols:
    for num_col in numerical_cols:
        plt.figure(figsize=(8, 6))
        sns.barplot(data=df, x=cat_col, y=num_col)
        plt.xlabel(cat_col)
        plt.xticks(rotation=90)
        plt.ylabel(num_col)
        plt.show()
```

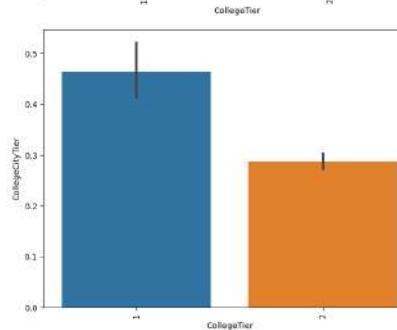
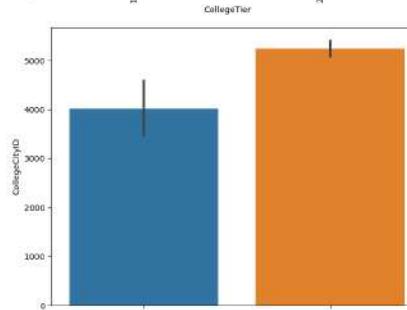
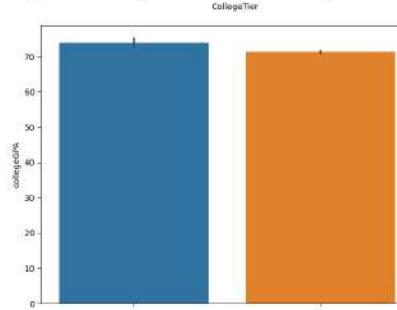
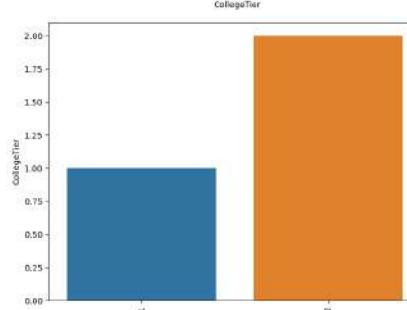
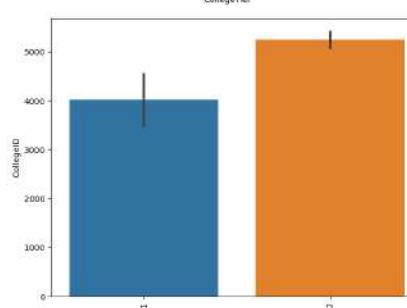
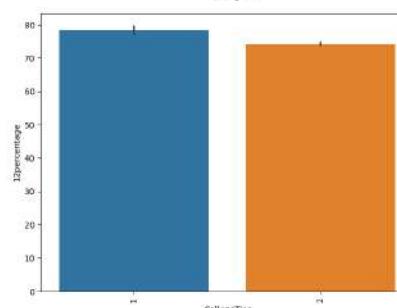
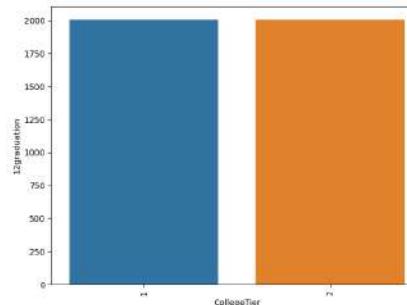


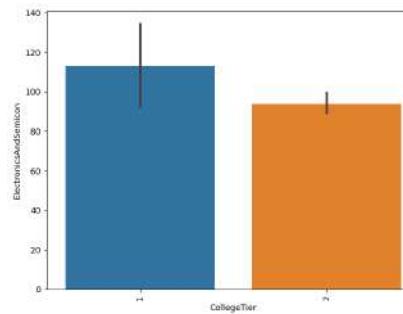
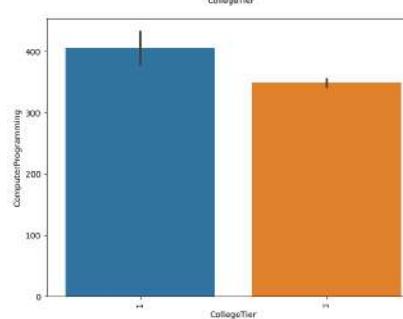
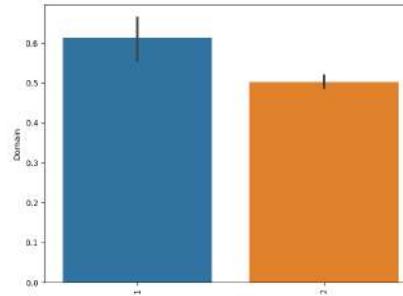
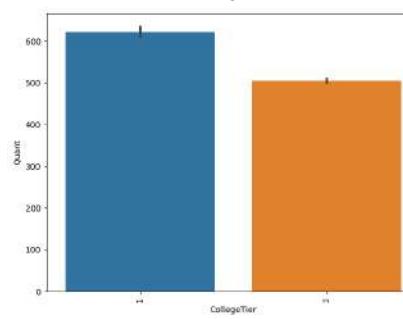
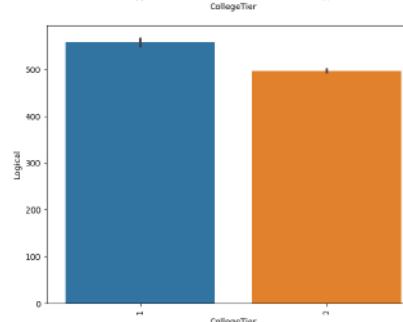
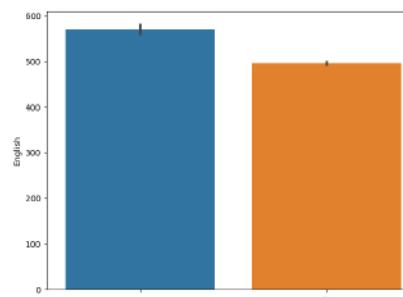
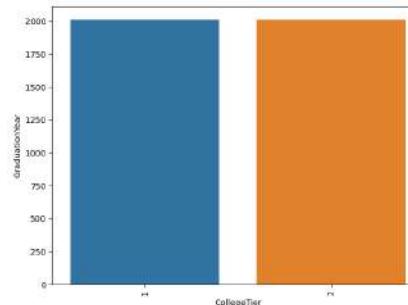


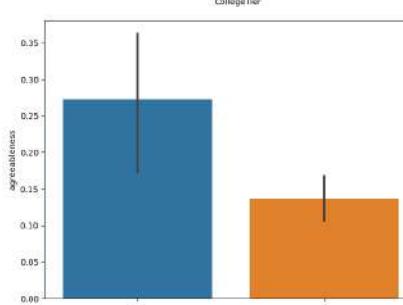
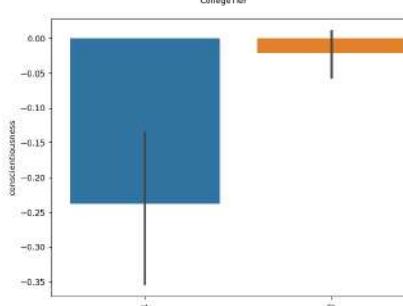
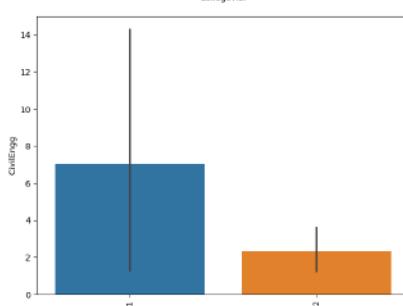
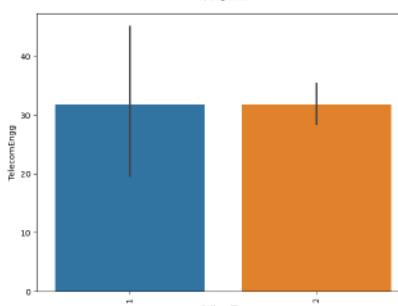
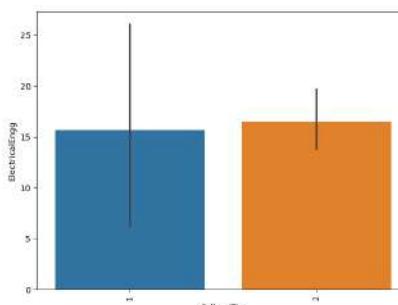
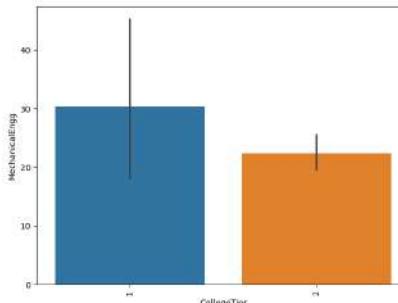
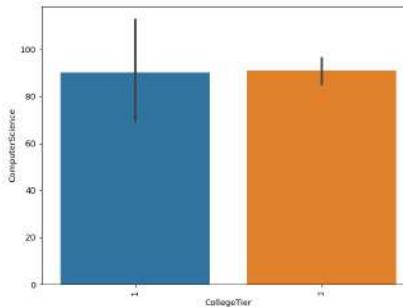


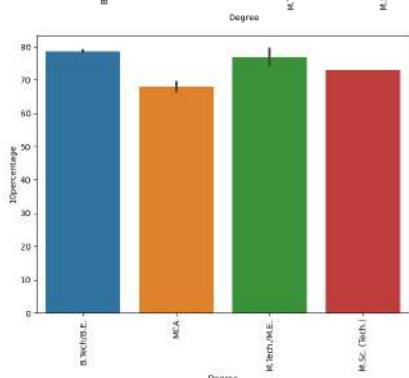
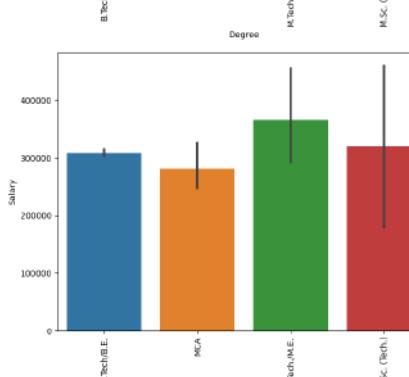
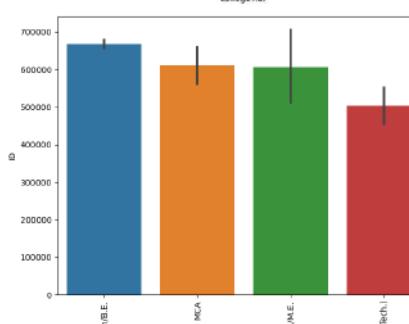
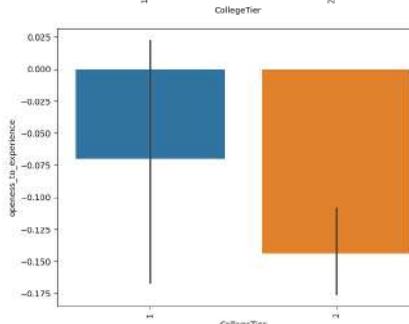
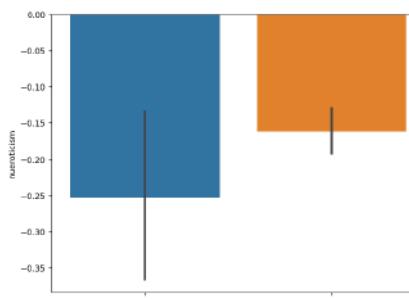
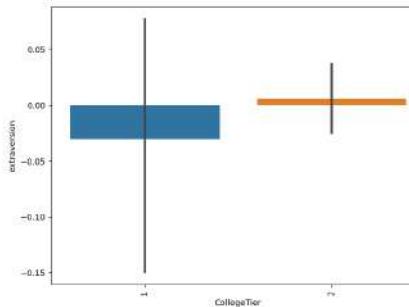


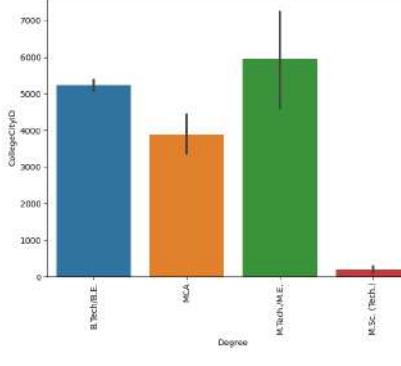
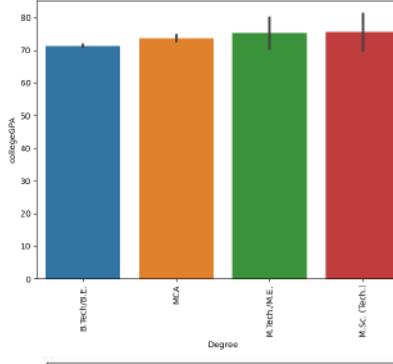
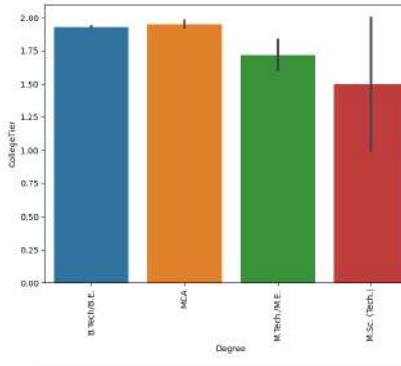
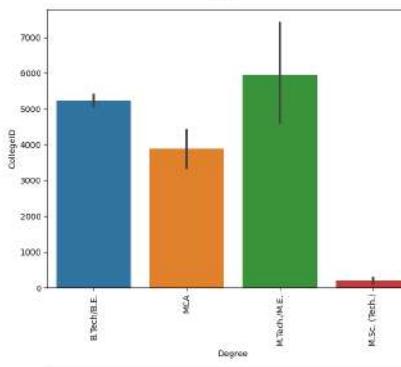
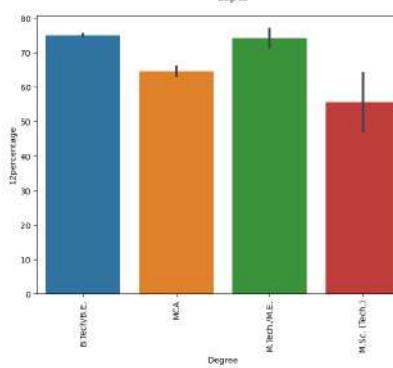
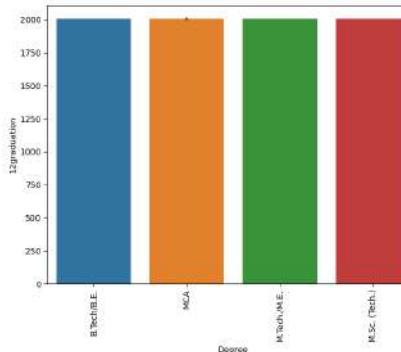


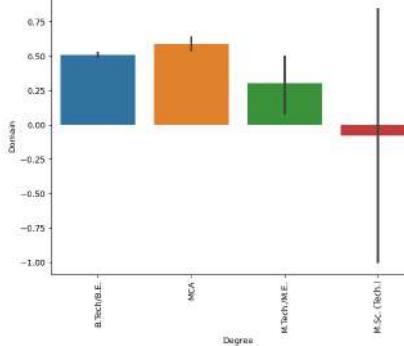
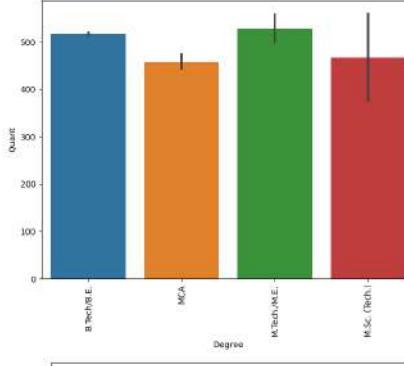
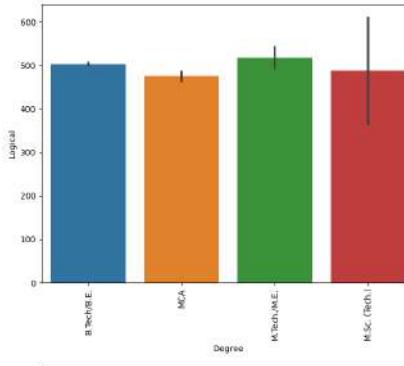
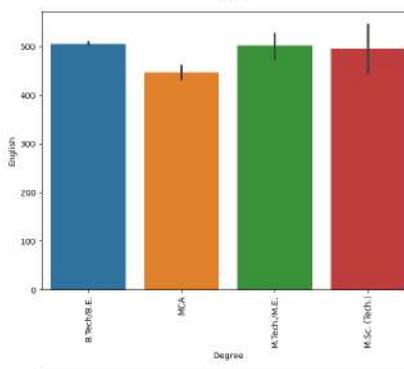
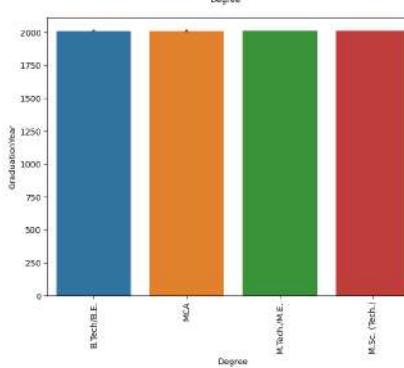
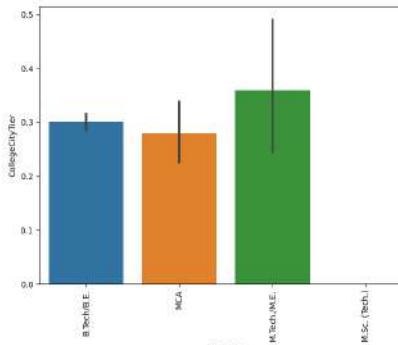


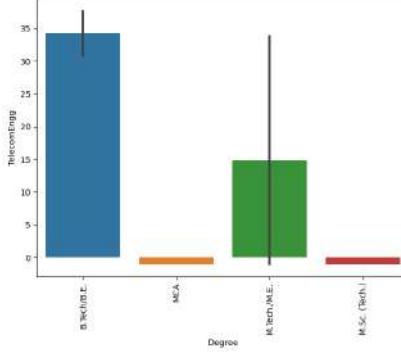
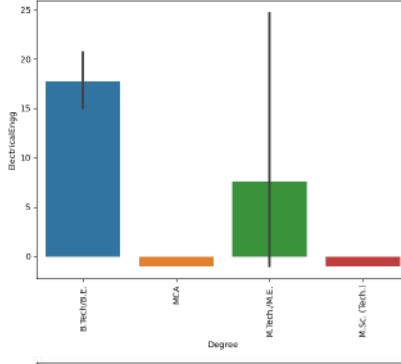
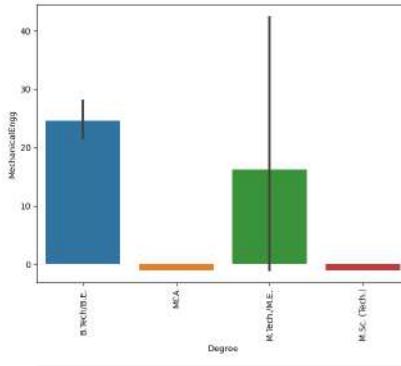
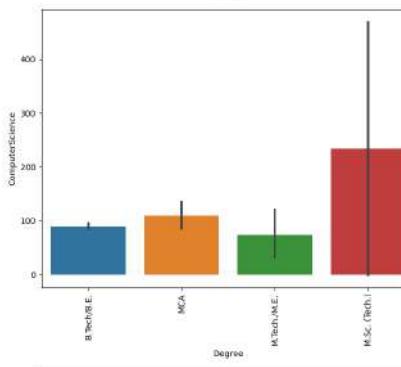
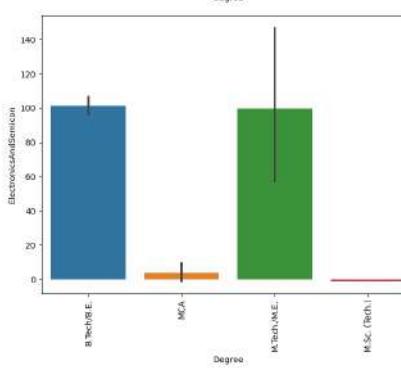
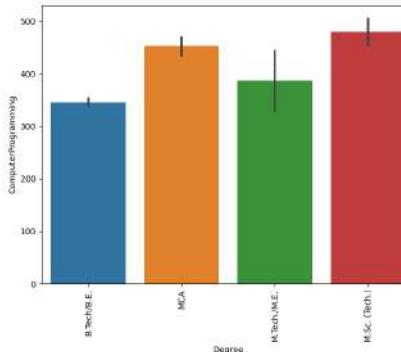


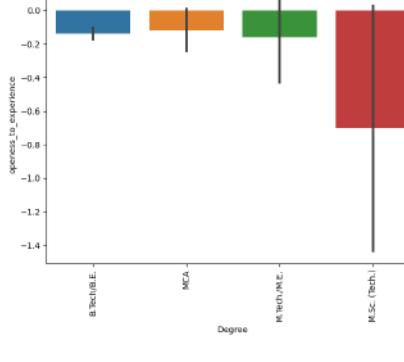
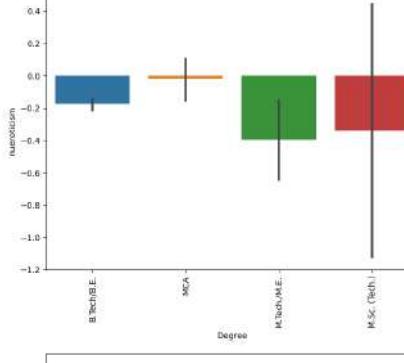
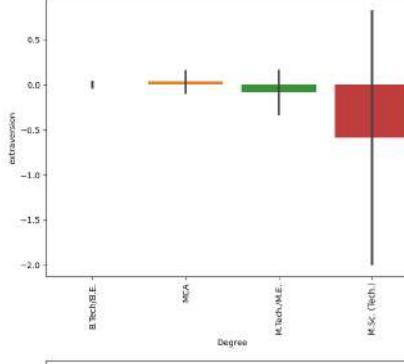
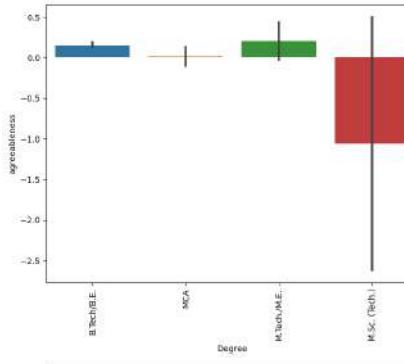
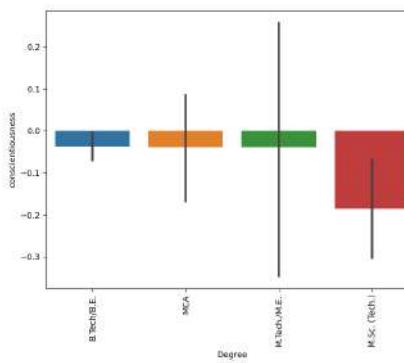
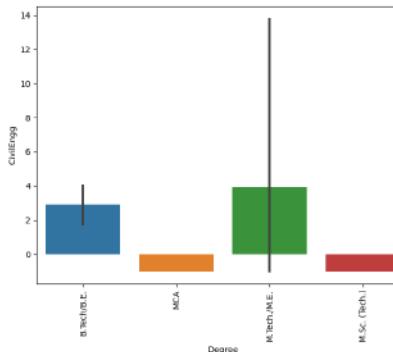


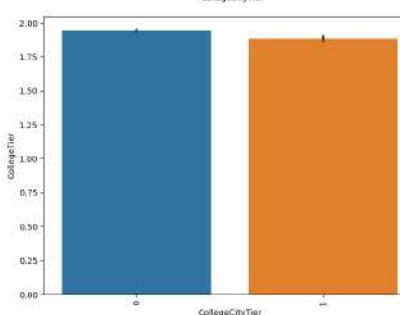
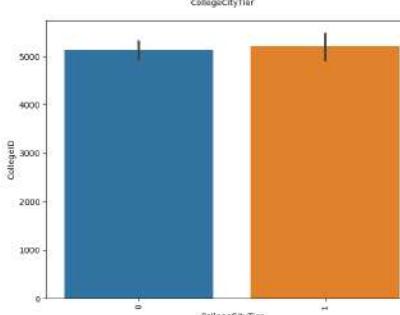
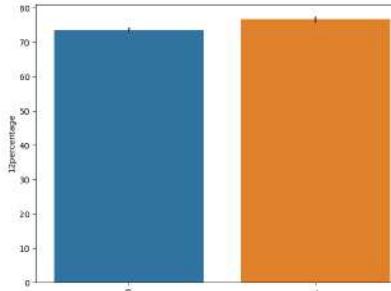
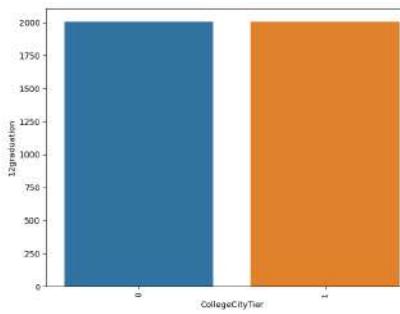
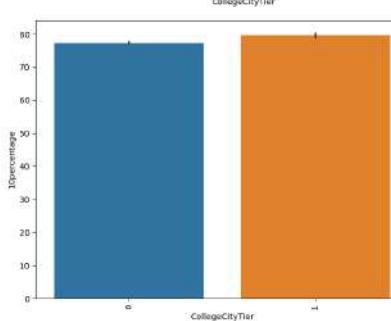
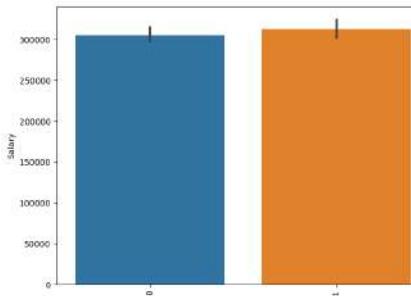
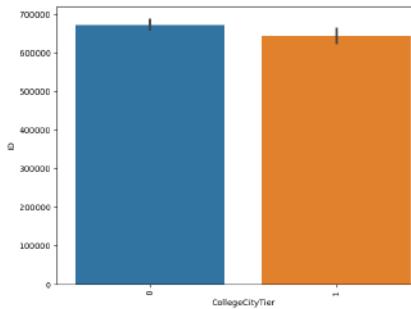


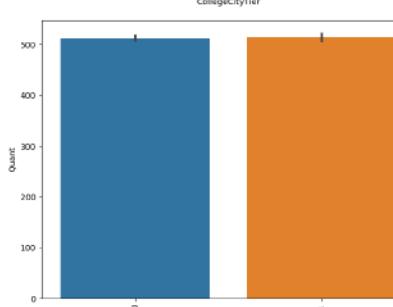
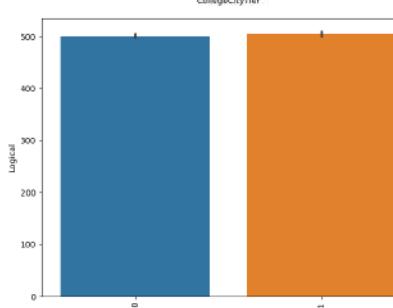
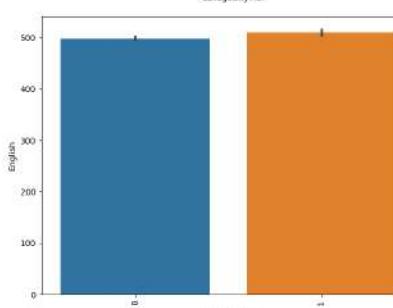
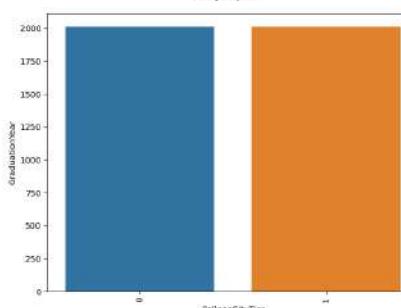
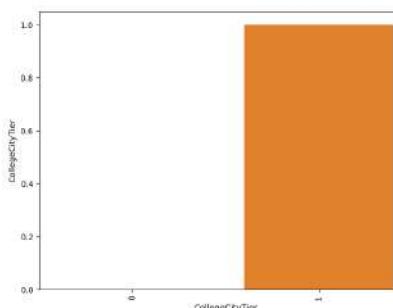
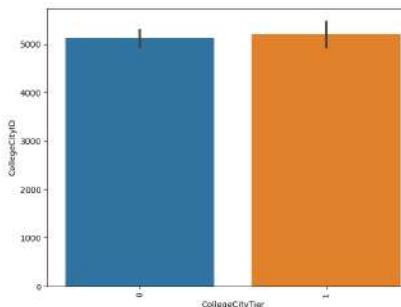
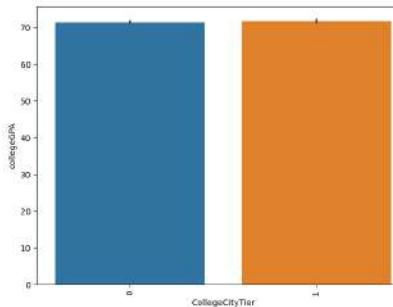


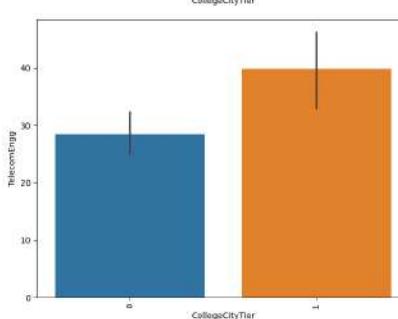
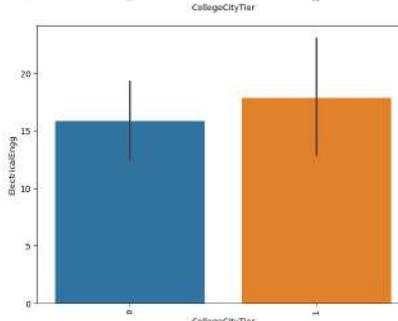
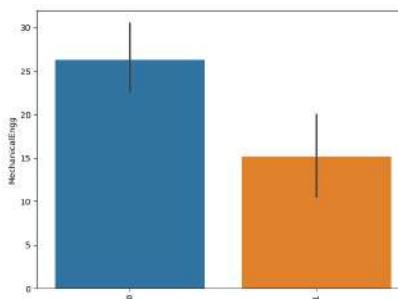
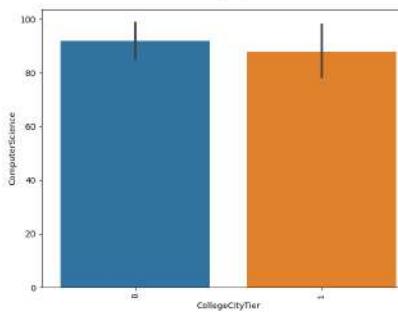
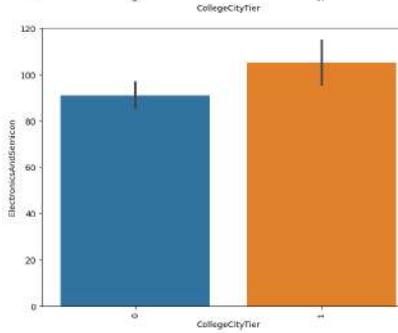
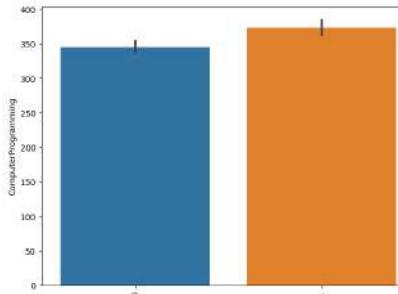
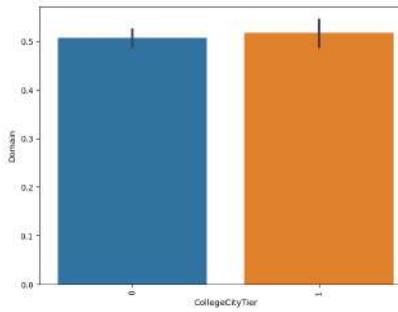


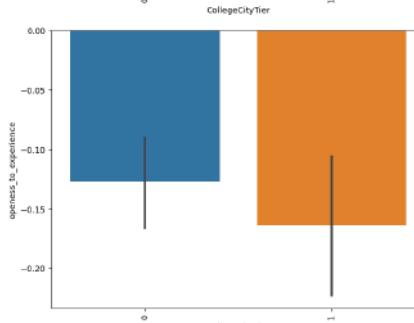
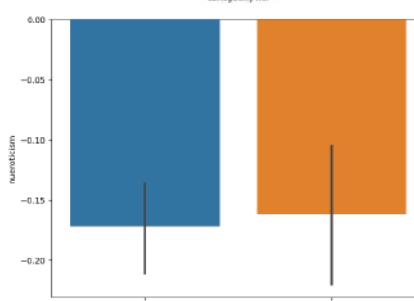
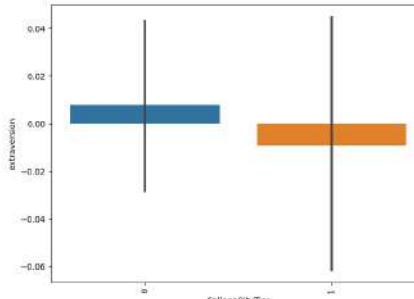
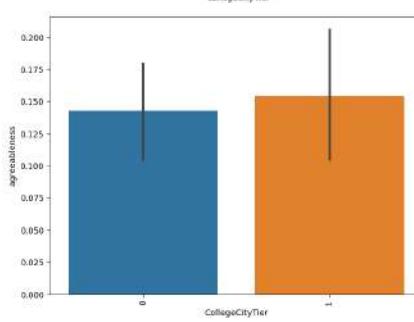
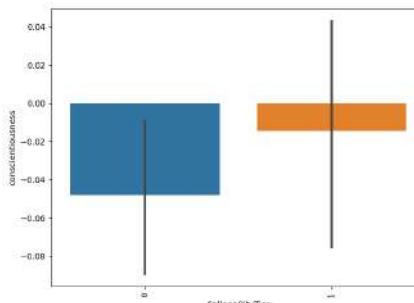
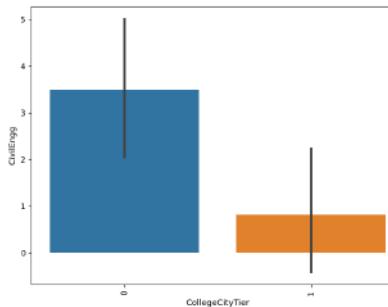


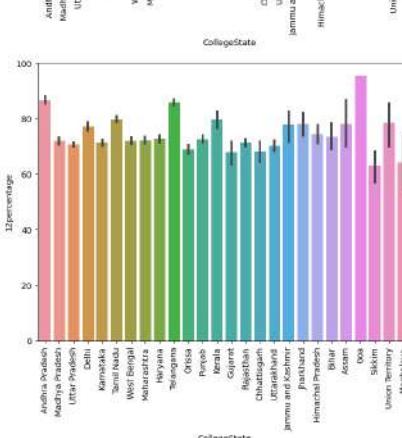
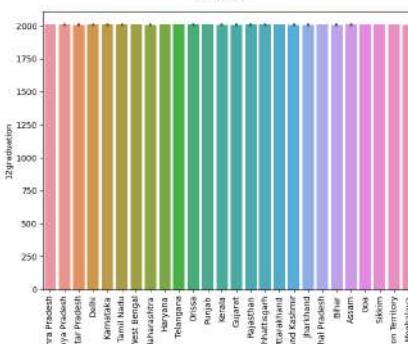
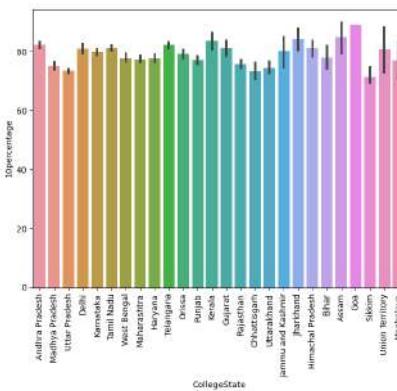
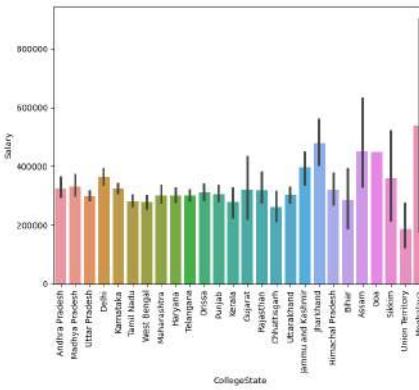
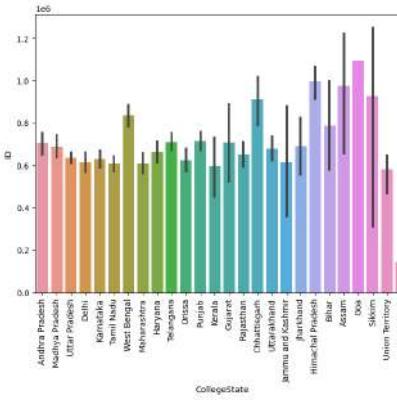


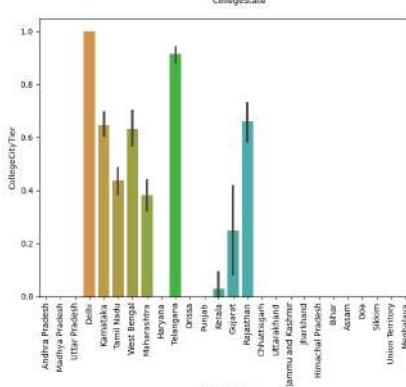
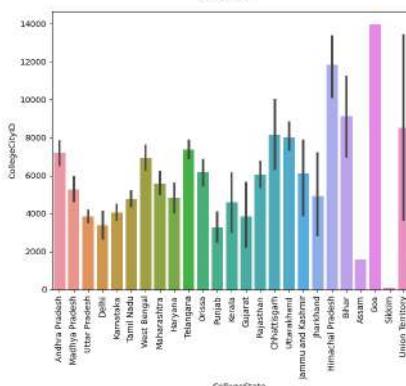
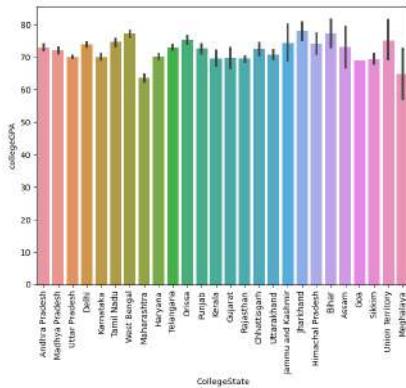
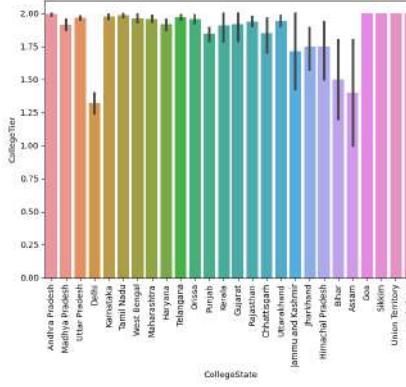
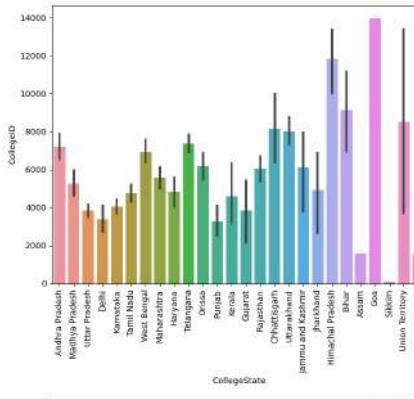


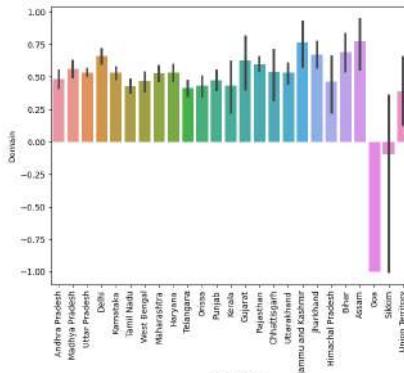
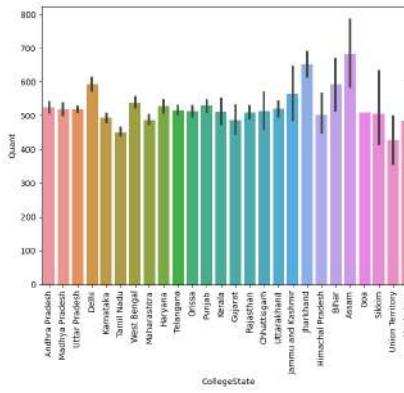
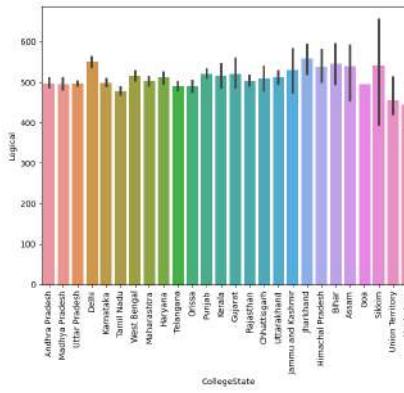
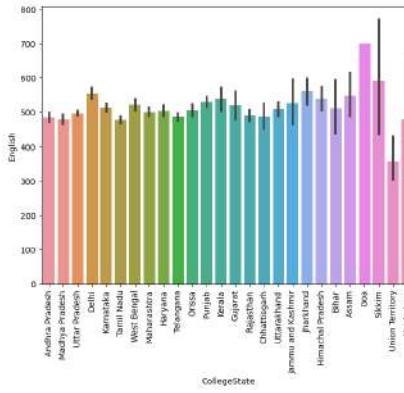
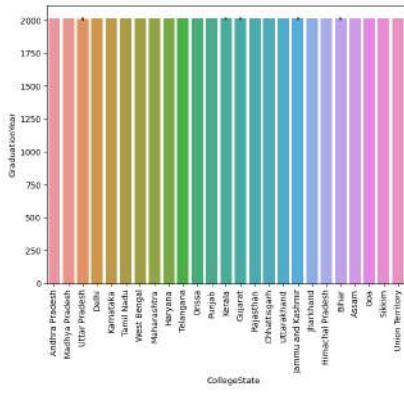


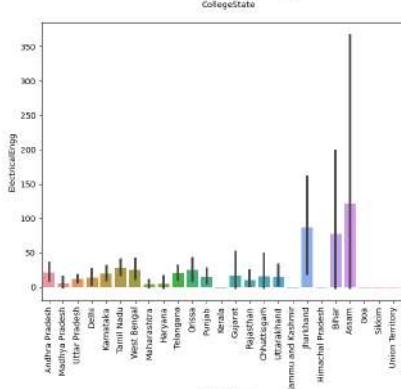
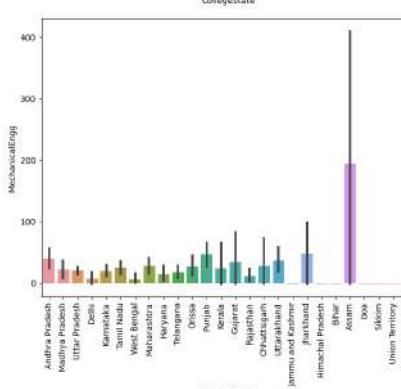
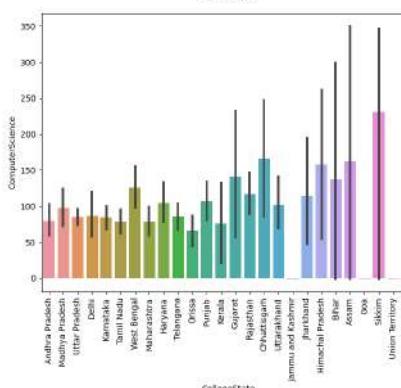
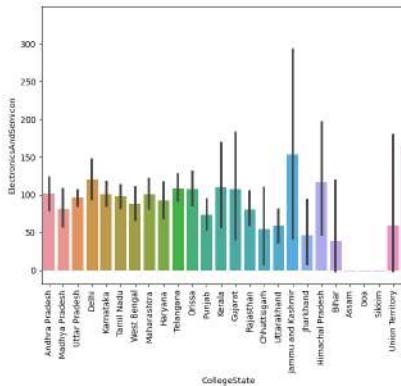
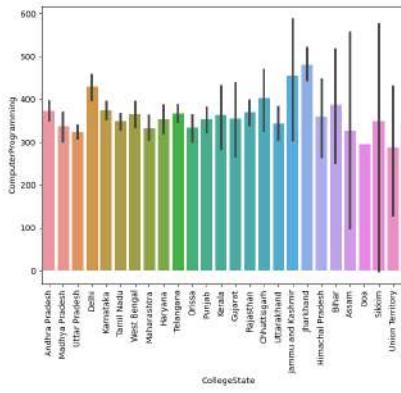


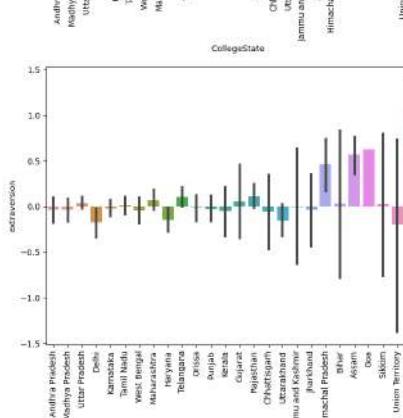
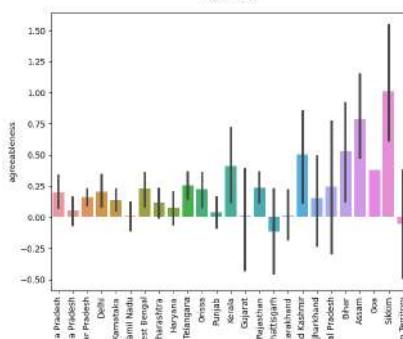
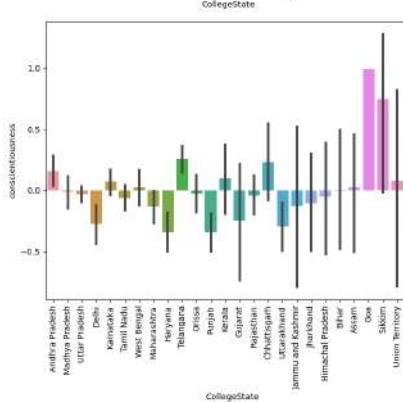
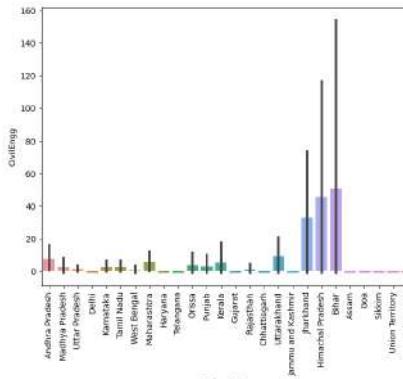
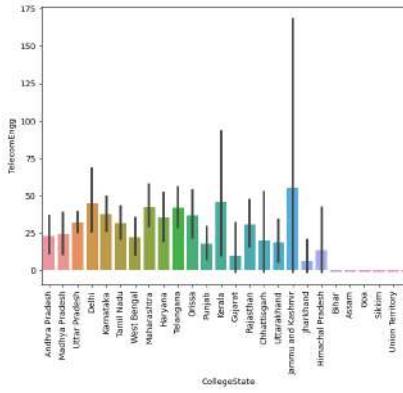


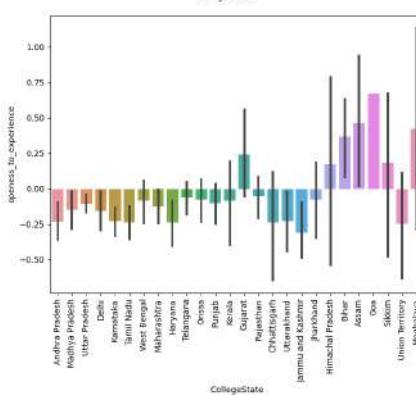
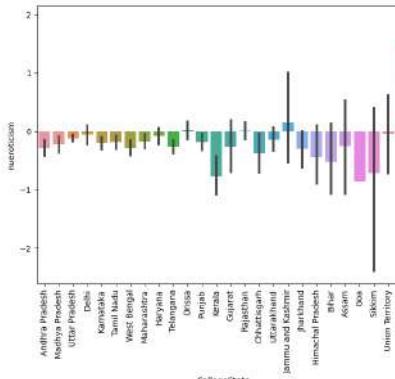








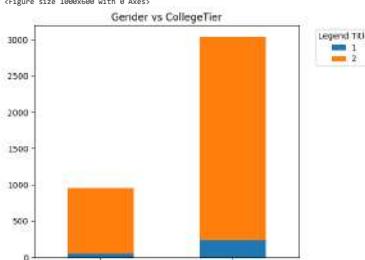




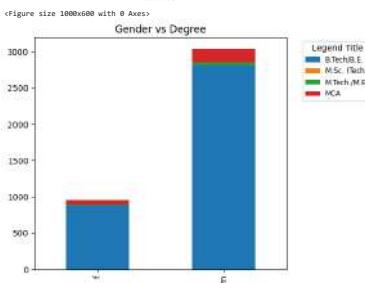
Categorical vs Categorical

```
In [27]: #stacked barplot
for col1 in categorical_cols:
    for col2 in categorical_cols:
        if col1 == col2:
            continue
        plt.figure(figsize=(10, 6))
        crossstab_data = pd.crosstab(df[col1], df[col2])
        crossstab_data.plot(kind='bar', stacked=True)
        plt.title(f'{col1} vs {col2}')
        plt.legend(loc='upper left', bbox_to_anchor=(1.05, 1), title='Legend Title', fontsize='small') # Adjust the position and size of the legend
        plt.title('Legend Title', fontweight='bold')
        plt.xticks(rotation=90)
        plt.tight_layout() # adjust layout to prevent overlapping
        plt.show()
```

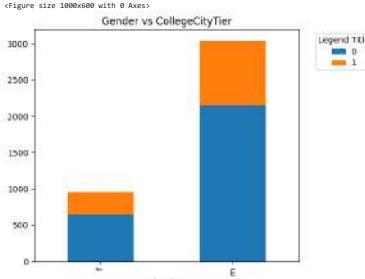
<Figure size 1000x600 with 0 Axes>



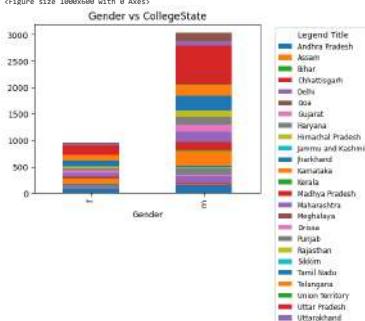
<Figure size 1000x600 with 0 Axes>



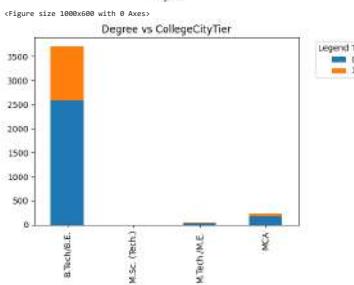
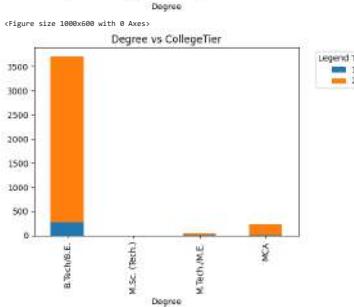
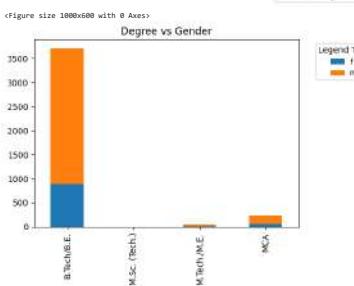
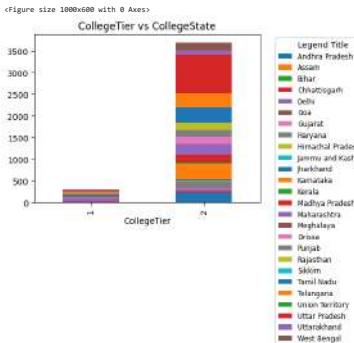
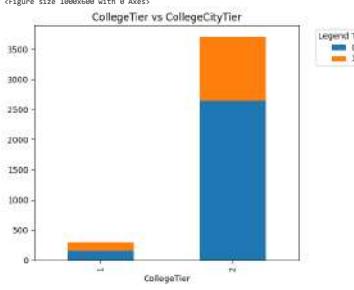
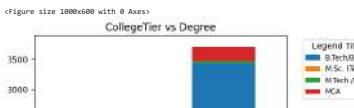
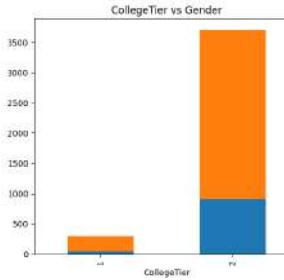
<Figure size 1000x600 with 0 Axes>

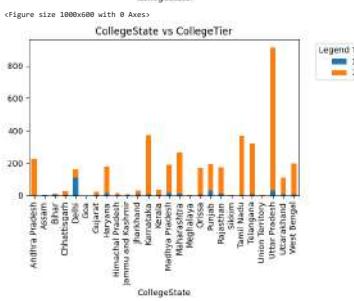
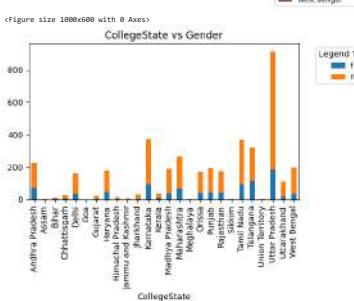
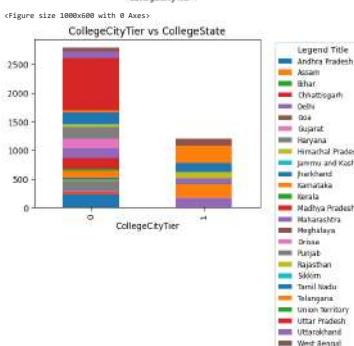
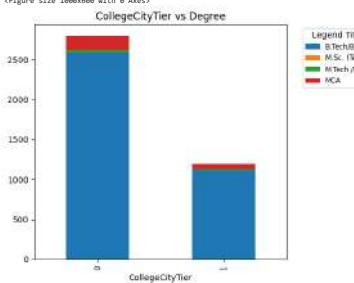
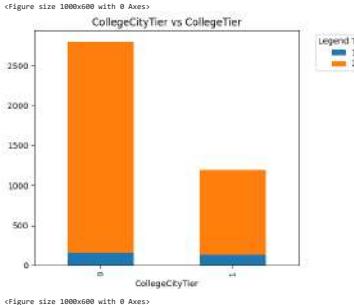
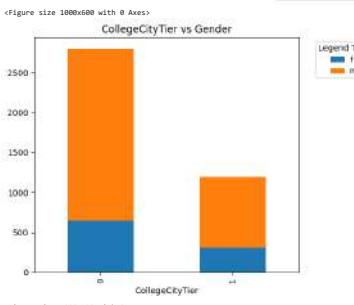
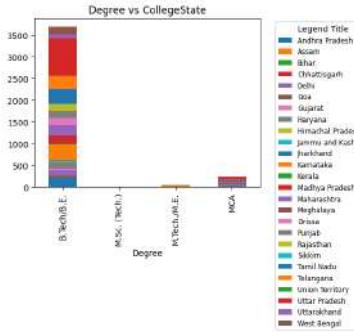


<Figure size 1000x600 with 0 Axes>



<Figure size 1000x600 with 0 Axes>






```
Summary Statistics for Salary in Claimed Roles:  
count      mean       std      min \n  
Desigination  
associate software engineer  46.0  307065.217391  98955.276476  70000.0  
hardware engineer           8.0  281250.000000  128473.176745  105000.0  
programmer analyst          135.0  353974.123358  55648.336987  90000.0  
software engineer            535.0  301215.213358  144399.578187  60000.0  
25%      50%      75%      max  
Desigination  
associate software engineer  300000.0  307500.0  330000.0  610000.0  
hardware engineer           175000.0  277500.0  400000.0  450000.0  
programmer analyst          300000.0  315000.0  355000.0  540000.0  
software engineer            255000.0  320000.0  400000.0  1500000.0  
Minimum Salary in Claimed Roles: 60000.0  
Maximum Salary in Claimed Roles: 1500000.0  
The claim is not supported by the data.
```

In []: