Engineering Graduate Salary Prediction

Exploratory Data Analysis

Arifa Eva Celinia Candra

Brief Description

India has a total 6,214 Engineering and Technology Institutions in which around 2.9 million students are enrolled. Every year, on an average 1.5 million students get their degree in engineering, but due to lack of skill required to perform technical jobs less than 20% get employment in their core domain.

A relevant question is what determines the salary and the jobs these engineers are offered right after graduation. Various factors such as college grades, candidate skills, the proximity of the college to industrial hubs, the specialization one have, market conditions for specific industries determine this. On the basis of these various factors, your objective is to determine the salary of an engineering graduate in India.

Data Source:
https://www.kaggle.com/datasets/manishkc06/engineering-graduate-salary-prediction

Data Description [1]

ID: A unique ID to identify a candidate
Salary: Annual CTC offered to the candidate (in INR)
Gender: Candidate's gender
DOB: Date of birth of the candidate
10percentage: Overall marks obtained in grade 10 examinations
10board: The school board whose curriculum the candidate followed in grade 10
12graduation: Year of graduation - senior year high school
12percentage: Overall marks obtained in grade 12 examinations
12board: The school board whose curriculum the candidate followed
CollegeID: Unique ID identifying the university/college which the candidate attended for her/his undergraduate
CollegeTier: Each college has been annotated as 1 or 2. The annotations have been computed from the average AMCAT scores obtained by the students in the college/university. Colleges with an average score above a threshold are tagged as 1 and
others as 2.
Degree: Degree obtained/pursued by the candidate
Specialization: Specialization pursued by the candidate
CollegeGPA: Aggregate GPA at graduation
CollegeCityID: A unique ID to identify the city in which the college is located in.
CollegeCityTier: The tier of the city in which the college is located in. This is annotated based on the population of the cities.
CollegeState: Name of the state in which the college is located

Data Description [2]

GraduationYear: Year of graduation (Bachelor's degree)
English: Scores in AMCAT English section
Logical: Score in AMCAT Logical ability section
Quant: Score in AMCAT's Quantitative ability section
Domain: Scores in AMCAT's domain module
ComputerProgramming: Score in AMCAT's Computer programming section
ElectronicsAndSemicon: Score in AMCAT's Electronics & Semiconductor Engineering section
ComputerScience: Score in AMCAT's Computer Science section
MechanicalEngg: Score in AMCAT's Mechanical Engineering section
ElectricalEngg: Score in AMCAT's Electrical Engineering section
TelecomEngg: Score in AMCAT's Telecommunication Engineering section
CivilEngg: Score in AMCAT's Civil Engineering section
conscientiousness: Scores in one of the sections of AMCAT's personality test
agreeableness: Scores in one of the sections of AMCAT's personality test
extraversion: Scores in one of the sections of AMCAT's personality test
nueroticism: Scores in one of the sections of AMCAT's personality test
openess_to_experience: Scores in one of the sections of AMCAT's personality test

Data Cleaning

[1] Reading and Understanding Our Data

- Read the data into pandas data frame and load the brief look of the data.

	ID	Gender	DOB	10percentage	10board	12graduation	12percentage	12board	CollegeID	CollegeTier	...	MechanicalEngg	ElectricalEngg	Telecor
0	604399	f	1990-10-22	87.80	cbse	2009	84.00	cbse	6920	1	...	-1	-1	
1	988334	m	1990-05-15	57.00	cbse	2010	64.50	cbse	6624	2	...	-1	-1	
2	301647	m	1989-08-21	77.33	maharashtra state board,pune	2007	85.17	amravati divisional board	9084	2	...	-1	-1	

- Find more information about the features and data types.

  We can see that there's no null data. Hence, we won't deal with missing values.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2998 entries, 0 to 2997
Data columns (total 34 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   ID                     2998 non-null   int64
 1   Gender                 2998 non-null   object
 2   DOB                    2998 non-null   object
 3   10percentage           2998 non-null   float64
 4   10board                2998 non-null   object
```

- Use the describe() function to show the count, mean, max, of the features attribute.

  Add median and range to describe() table

	ID	10percentage	12graduation	12percentage	CollegeID	CollegeTier	collegeGPA	CollegeCityID	CollegeCityTier	GraduationYear	...	M
count	2.998000e+03	2998.000000	2998.000000	2998.000000	2998.000000	2998.000000	2998.000000	2998.000000	2998.000000	2998.000000	...	
mean	6.648926e+05	77.666264	2008.080720	74.341061	5210.210807	1.924616	71.509857	5210.210807	0.296197	2011.939960	...	
std	3.648951e+05	10.002785	1.631814	11.120299	4776.609877	0.264053	8.122462	4776.609877	0.456655	36.780582	...	
25%	3.334648e+05	71.140000	2007.000000	66.000000	526.250000	2.000000	66.530000	526.250000	0.000000	2012.000000	...	
median	6.396945e+05	78.965000	2008.000000	74.000000	4027.500000	2.000000	71.800000	4027.500000	0.000000	2013.000000	...	
75%	9.951770e+05	85.600000	2009.000000	82.600000	8822.250000	2.000000	76.300000	8822.250000	1.000000	2014.000000	...	
min	1.124400e+04	43.000000	1998.000000	40.000000	2.000000	1.000000	6.630000	2.000000	0.000000	0.000000	...	
max	1.297877e+06	97.760000	2012.000000	98.700000	18409.000000	2.000000	99.930000	18409.000000	1.000000	2017.000000	...	
range	1.286633e+06	54.760000	14.000000	58.700000	18407.000000	1.000000	93.300000	18407.000000	1.000000	2017.000000	...	

[2] Handling the Duplicates

- Check whether there are any duplicates in our data and if there’s any, remove the duplicates.

  There’s no duplicate data, so we don’t remove any duplicates

```
#Check if there is any duplicate in our dataframe
df['ID'].duplicated().sum()
```

```
0
```

```
df.ID.is_unique
```

```
True
```

```
df.index.is_unique
```

```
True
```

[3] Feature Selection

- Get rid of some irrelevant variables that do not affect Salary

```
df1 = df.drop(['ID', 'DOB', 'CollegeID', '12graduation' ,'GraduationYear','10board', '12board' , 'CollegeState','CollegeCityID',
              'CollegeCityTier'], axis = 1)
```

- Check category counts to make sure all categories have reasonable representation

Aggregate some variables in Specialization feature into 1 category because they only have very few counts

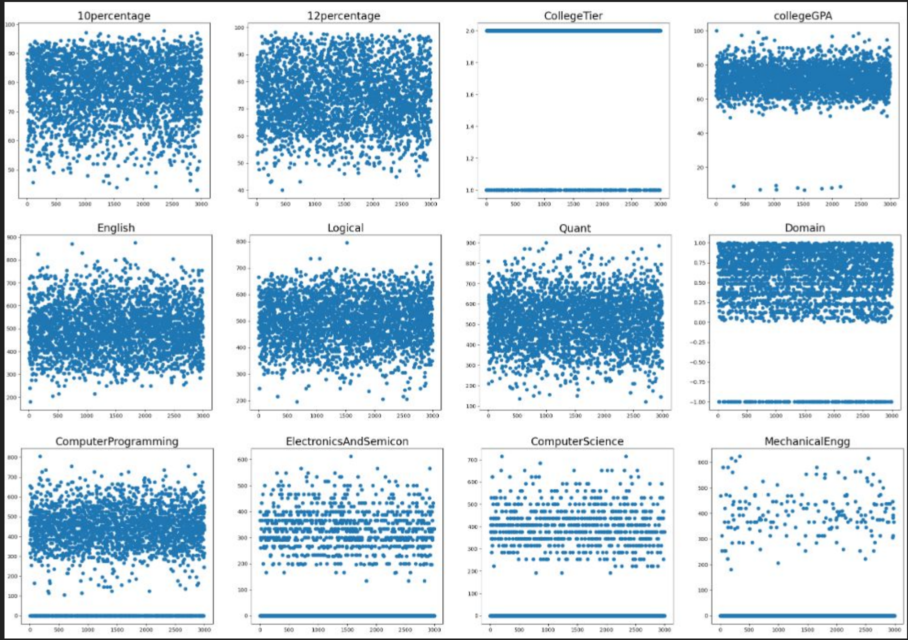
```
electronics and communication engineering      670
computer science & engineering                 557
information technology                         506
computer engineering                           415
computer application                           201
mechanical engineering                         155
electronics and electrical engineering         148
electronics & telecommunications                89
electrical engineering                          63
electronics & instrumentation eng               24
instrumentation and control engineering         18
information science engineering                 18
electronics and instrumentation engineering     18
civil engineering                               15
electronics engineering                         13
biotechnology                                   12
other                                           10
industrial & production engineering              8
chemical engineering                             7
applied electronics and instrumentation          5
mechanical and automation                        5
telecommunication engineering                    4
automobile/automotive engineering                4
computer science and technology                  4
aeronautical engineering                         3
instrumentation engineering                      3
electronics and computer engineering             3
mechatronics                                     3
metallurgical engineering                        2
industrial engineering                           2
biomedical engineering                           2
information & communication technology           1
electronics                                      1
embedded systems technology                      1
industrial & management engineering              1
electrical and power engineering                 1
computer and communication engineering           1
mechanical & production engineering              1
control and instrumentation engineering          1
ceramic engineering                              1
computer networking                              1
information science                              1
Name: Specialization, dtype: int64
```

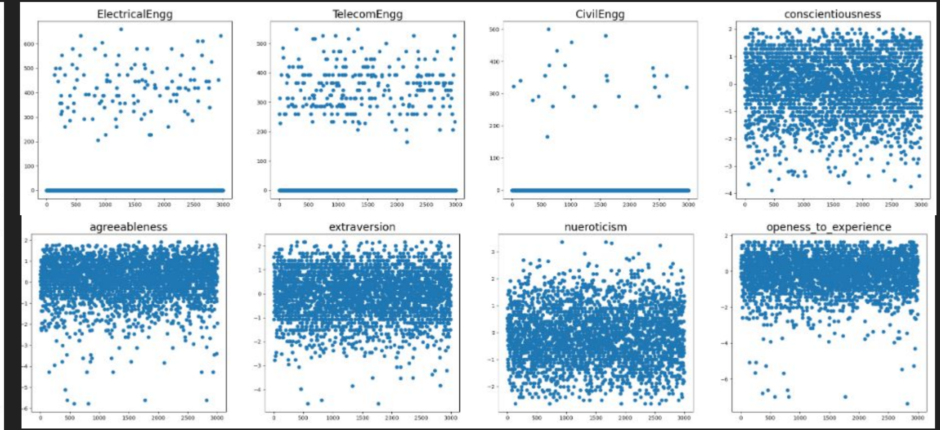
```
electronics and communication engineering      670
computer science & engineering                 557
information technology                         506
computer engineering                           415
computer application                           201
mechanical engineering                         155
electronics and electrical engineering         148
electronics & telecommunications                89
Others                                          76
electrical engineering                          63
electronics & instrumentation eng               24
electronics and instrumentation engineering     18
instrumentation and control engineering         18
information science engineering                 18
civil engineering                               15
electronics engineering                         13
biotechnology                                   12
Name: Specialization, dtype: int64
```

[3] Handling The Outliers

- Plot every predictor variable using scatter plot function to visually detect the outliers.

  We found outliers in variables: collegeGPA, Domain, ComputerProgramming, ElectronicsAndSemicon, ComputerScience, MechanicalEngg, ElectricalEngg, TelecommEngg, and CivilEngg.

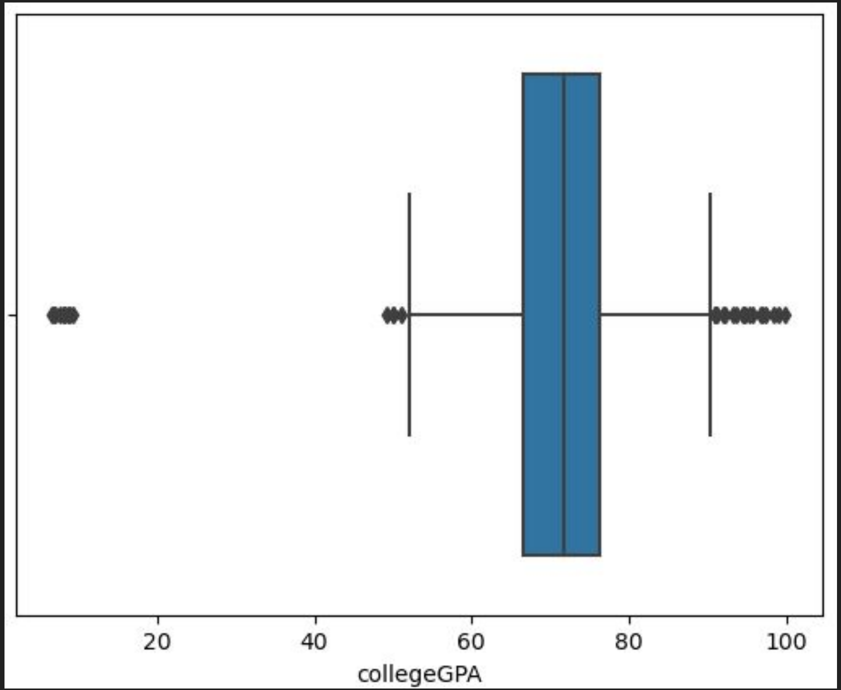


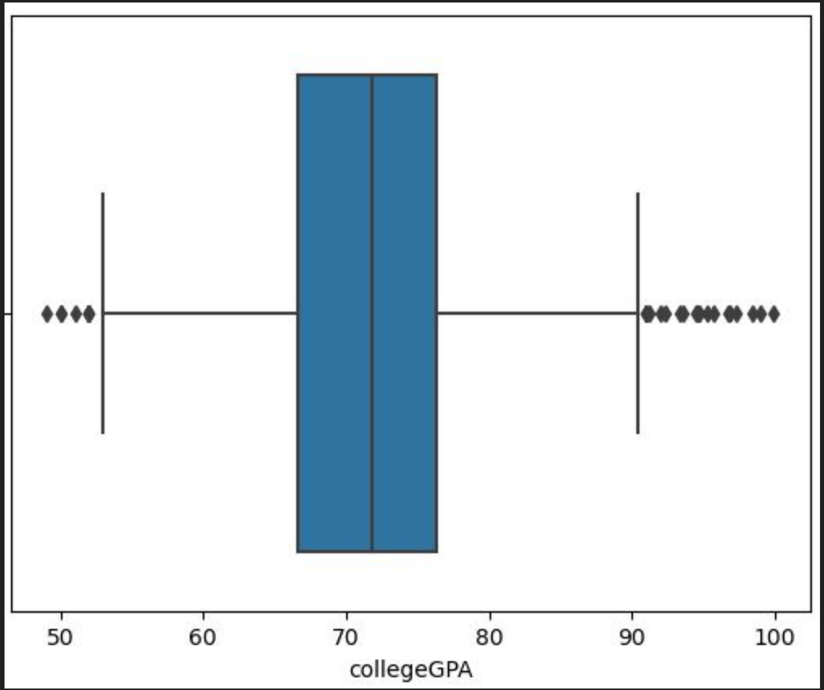


[3.1] Handling The Outliers

- Using uni-variate analysis (box plot) to get rid of outliers in collegeGPA variable.

  Drop collegeGPA less than 20

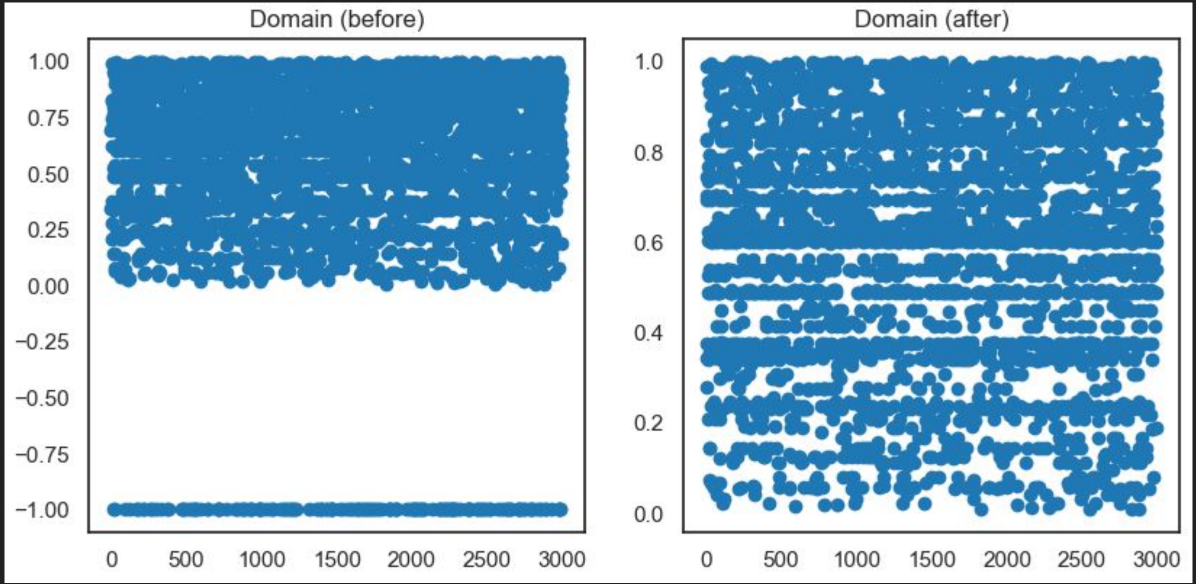




[3.2] Handling The Outliers

- We found out that in variables Domain, ComputerProgramming, ElectronicsAndSemicon, ComputerScience, MechanicalEngg, ElectricalEngg, TelecommEngg, and CivilEngg all data outliers are in -1 values. To handle these outliers, we will replace -1 with nan first, and then fill the nan values with mean values

```
df2 = df2.replace(-1,np.nan)
cols_with_nan = [col for col in df2.columns if df2.isna().sum()[col]>0]
for col in cols_with_nan:
    df2[col] = df2[col].fillna(df2[col].mean())
```

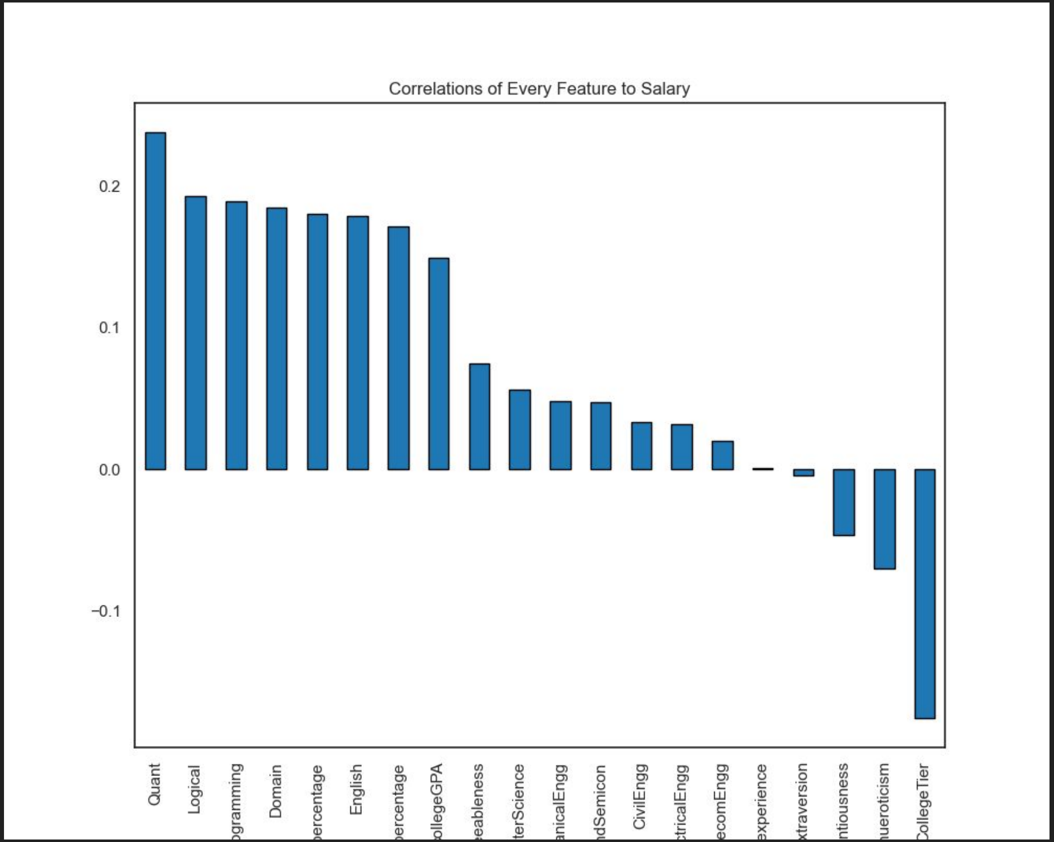


Exploratory Data Analysis

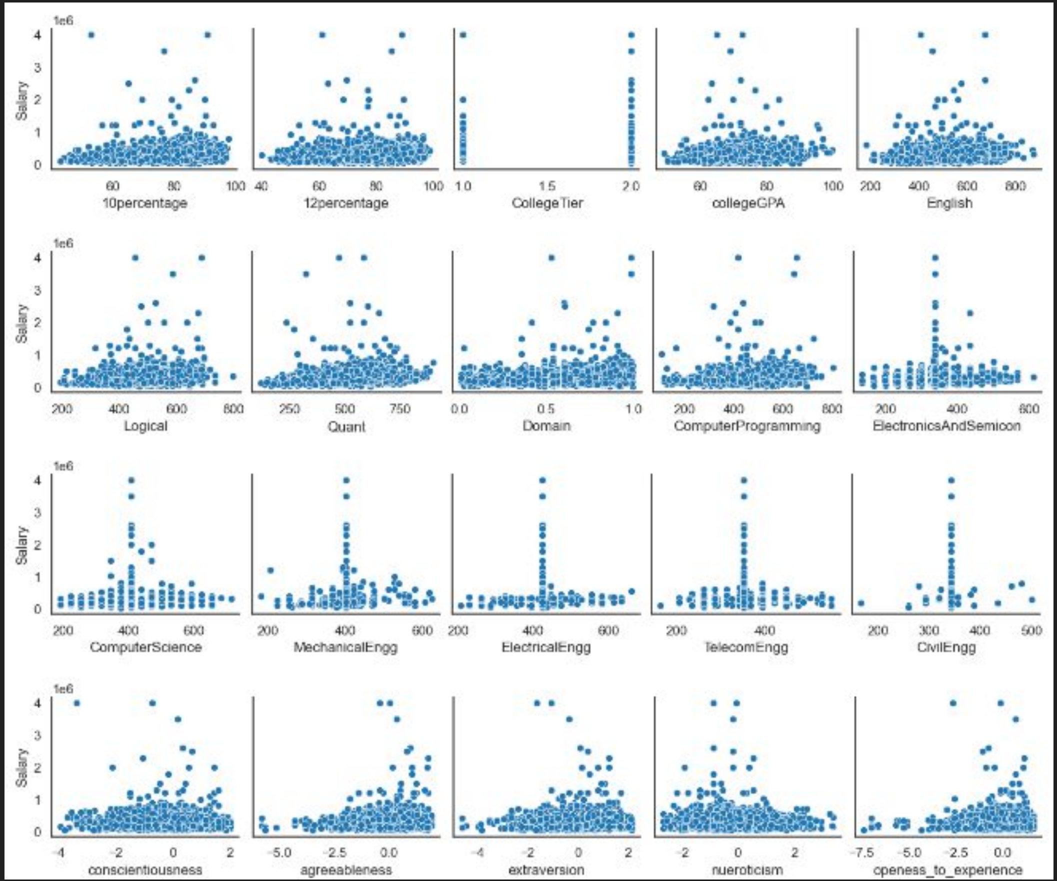
[1] Looking for Correlations

- Calculate correlations of every predictor variable to target variable (Salary)

```
Quant                      0.238025
Logical                    0.192844
ComputerProgramming        0.189329
Domain                     0.184818
10percentage               0.180528
English                    0.178810
12percentage               0.171857
collegeGPA                 0.149643
agreeableness              0.074807
ComputerScience            0.056355
MechanicalEngg             0.048106
ElectronicsAndSemicon      0.047189
CivilEngg                  0.033688
ElectricalEngg             0.031881
TelecomEngg                0.020219
openess_to_experience      0.000987
extraversion              -0.004129
conscientiousness         -0.046078
nueroticism               -0.069793
CollegeTier               -0.175449
Name: Salary, dtype: float64
```

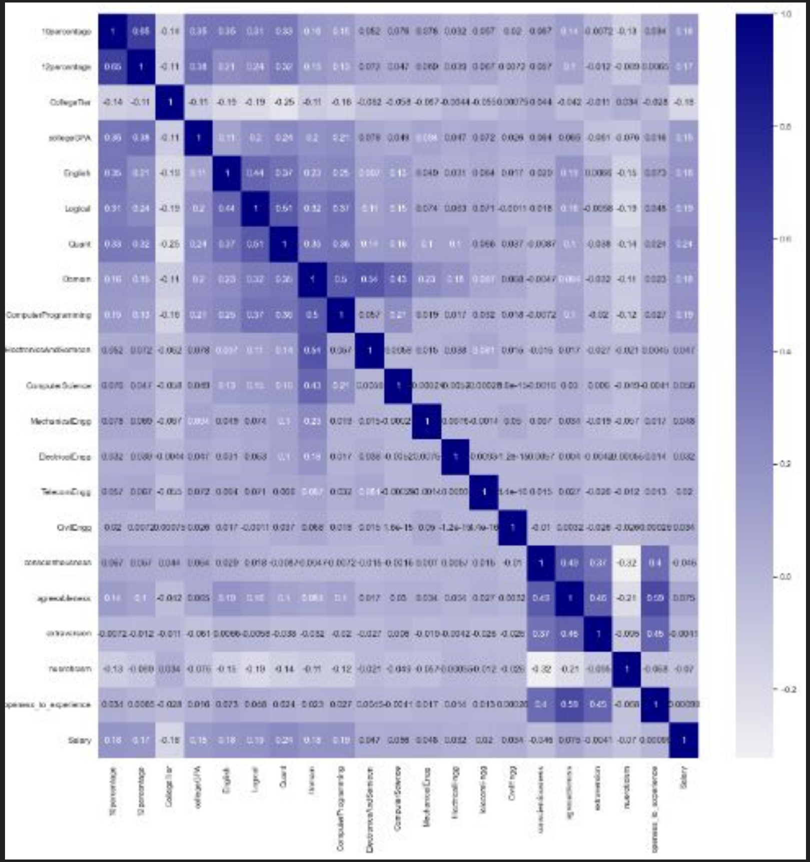


Visually Inspect the Correlations



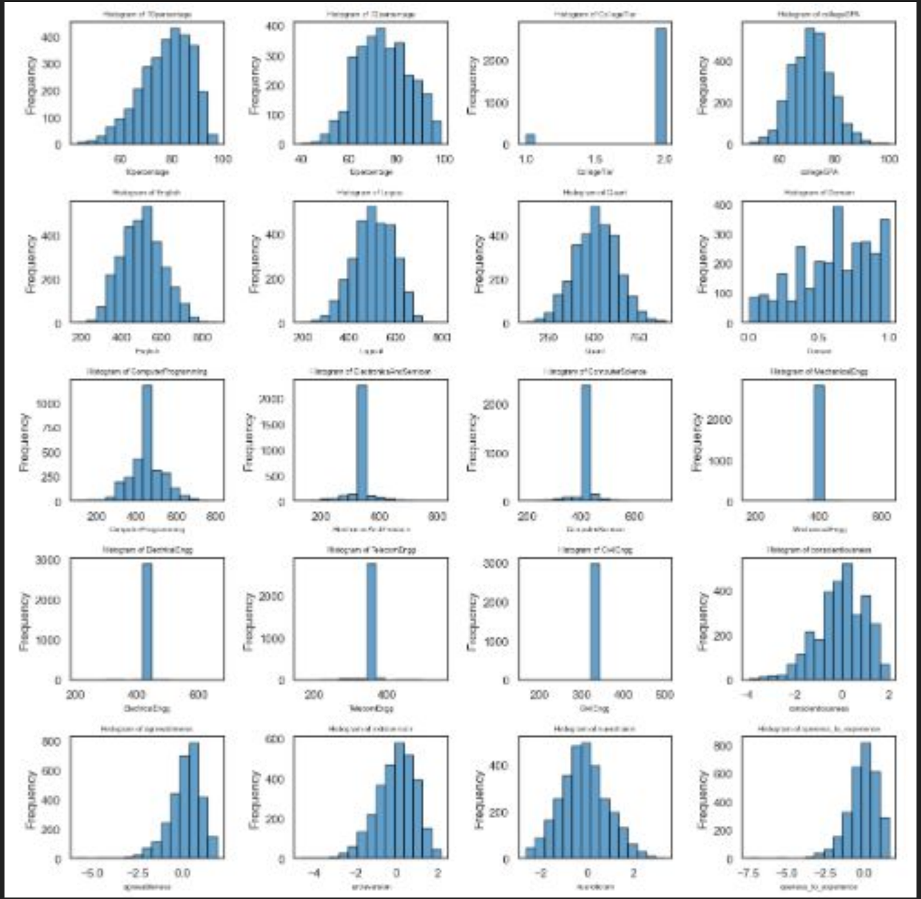
- Build relationship plots of every predictor variables to target variable (Salary) We can see that there are upward-curved relationships in the correlation plots between ‘Salary’ to ‘agreeableness’ and between ‘Salary’ to ‘openess_to_experience’. This suggests that we should do quadratic polynomial terms or transformations for these features later.

Construct a Correlation Heatmap



- Build a heatmap to see the relationships across the variables
  We can see that the relationship between ‘12percentage’ and ‘10percentage’ is the strongest.

[2] Skew Variables

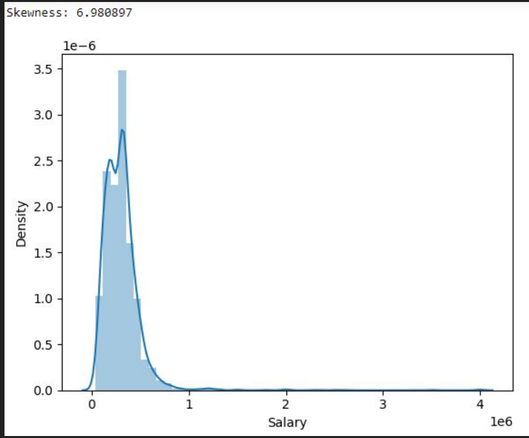


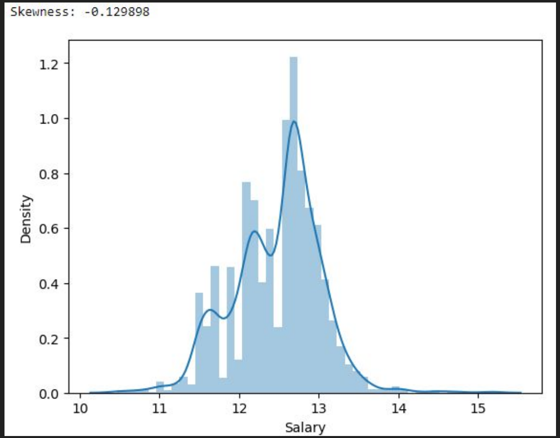
- Build distribution plots for every feature to visually inspect any variables with skewed distribution

Log Transformation

1) Visually inspect the effect of log transformation first.

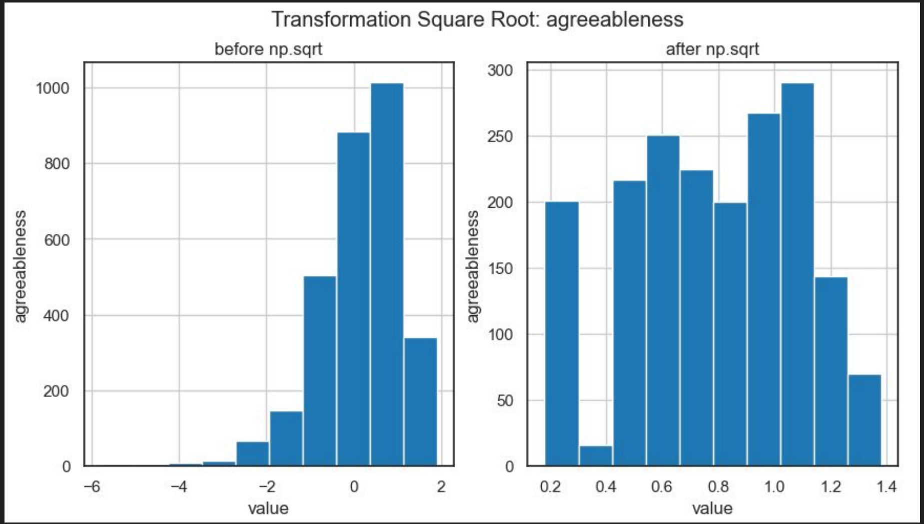
2) Apply log transformation



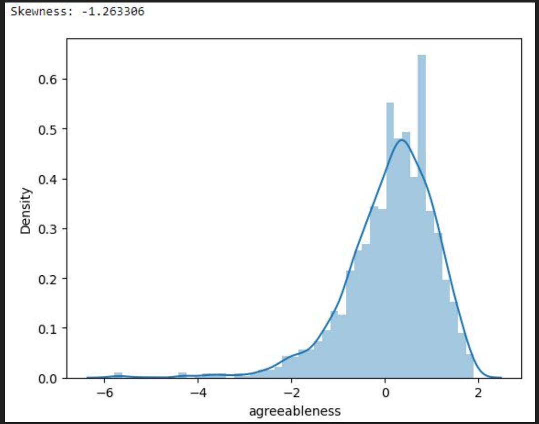


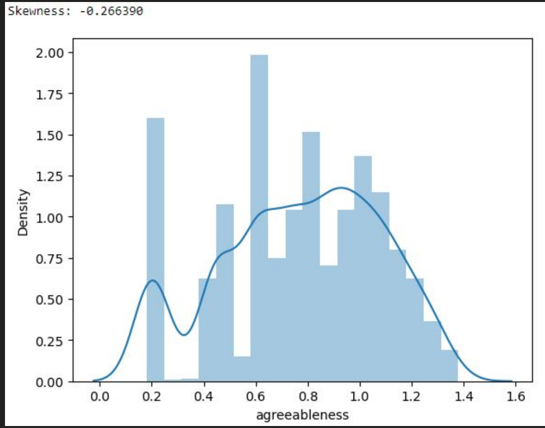
Square Root Transformation

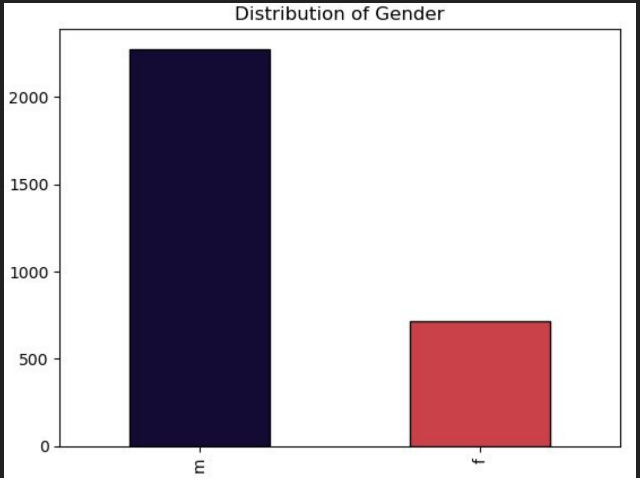
1) Visually inspect the effect of square root transformation first.



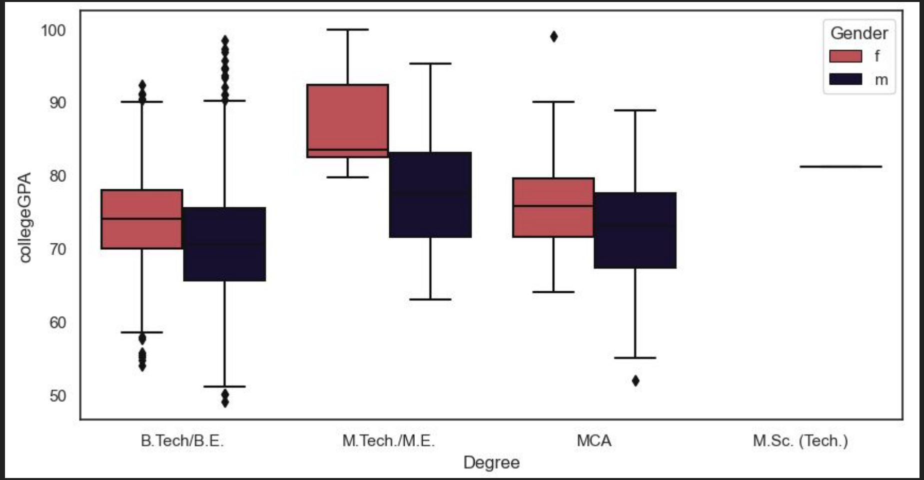
2) Apply square root transformation



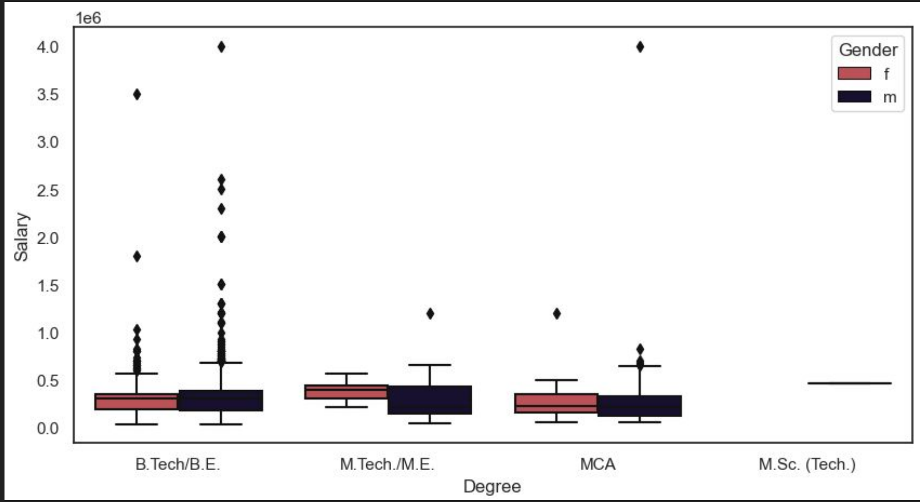




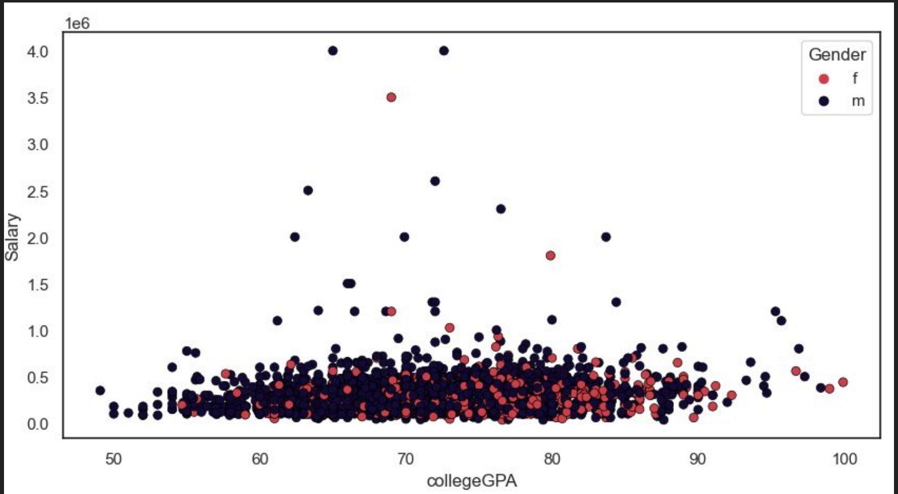
Distribution of Gender
2000
1500
1000
500
0
m
f

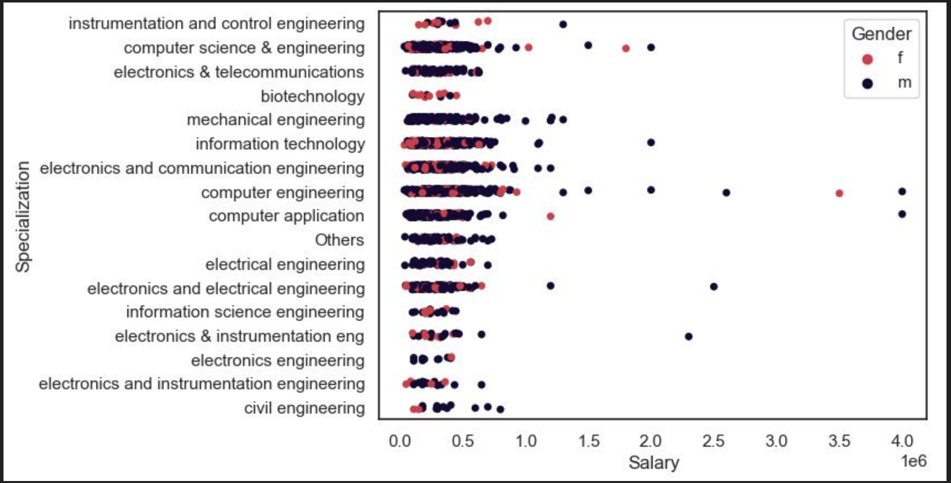


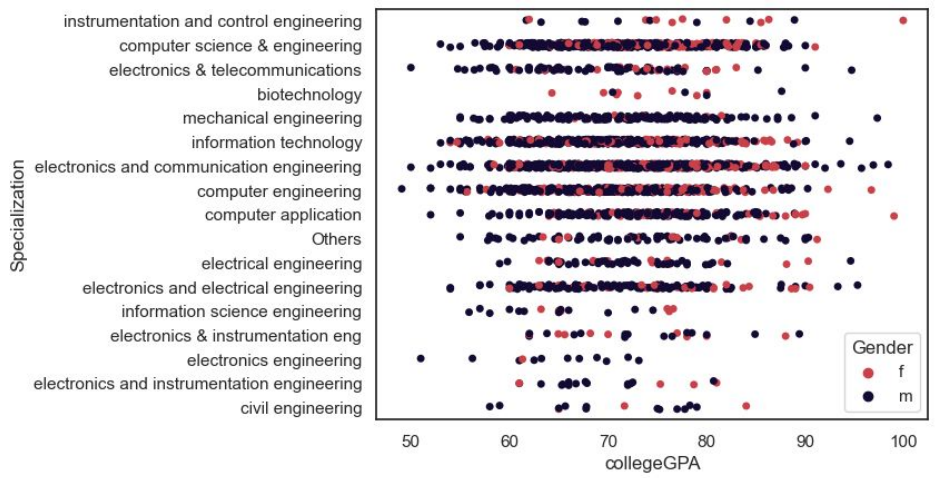
collegeGPA
100
90
80
70
60
50
Gender
f
m
B.Tech/B.E.
M.Tech./M.E.
MCA
M.Sc. (Tech.)
Degree

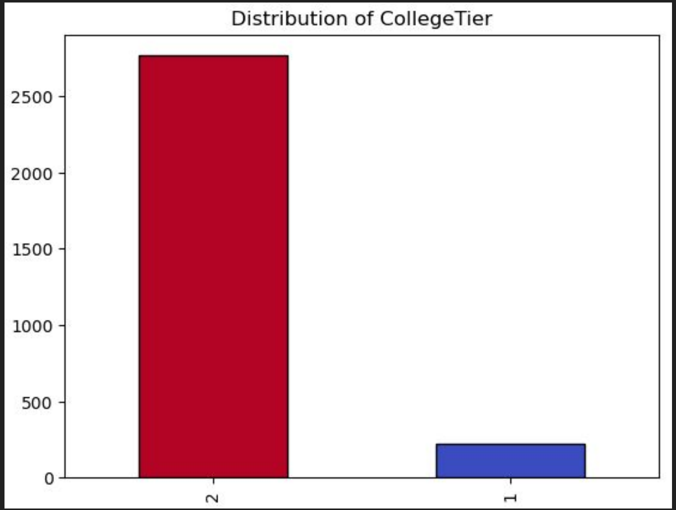


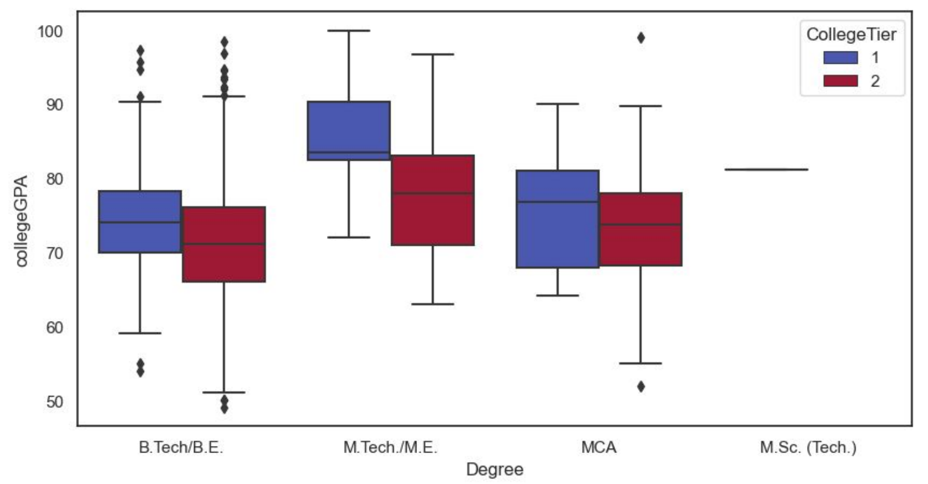
1e6
Salary
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Gender
f
m
B.Tech/B.E.
M.Tech./M.E.
MCA
M.Sc. (Tech.)
Degree

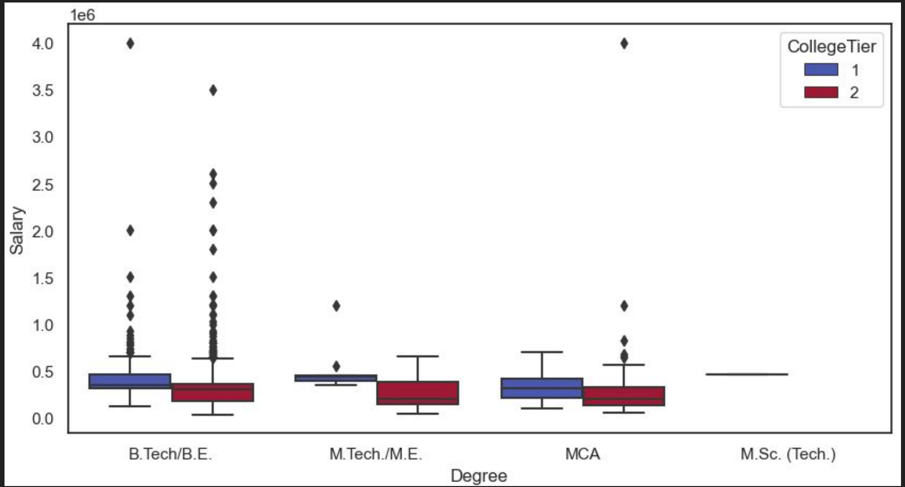

1e6
Gender
f
m
Salary
collegeGPA

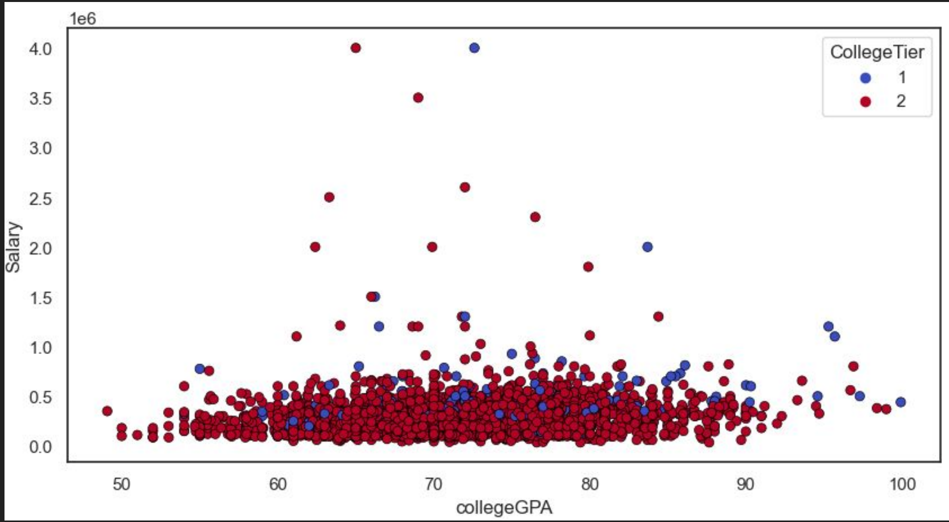

instrumentation and control engineering
computer science & engineering
electronics & telecommunications
biotechnology
mechanical engineering
information technology
electronics and communication engineering
computer engineering
computer application
Others
electrical engineering
electronics and electrical engineering
information science engineering
electronics & instrumentation eng
electronics engineering
electronics and instrumentation engineering
civil engineering
Specialization
Gender
f
m
Salary
1e6

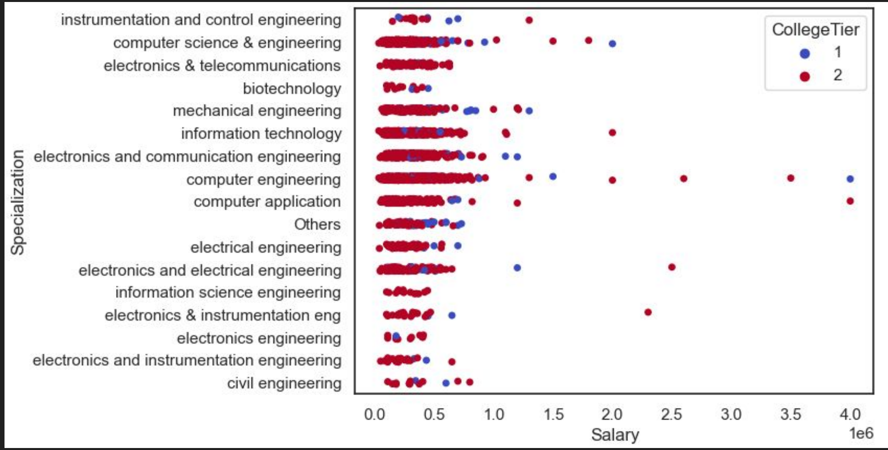

instrumentation and control engineering
computer science & engineering
electronics & telecommunications
biotechnology
mechanical engineering
information technology
electronics and communication engineering
computer engineering
computer application
Others
electrical engineering
electronics and electrical engineering
information science engineering
electronics & instrumentation eng
electronics engineering
electronics and instrumentation engineering
civil engineering
Specialization
Gender
f
m
collegeGPA

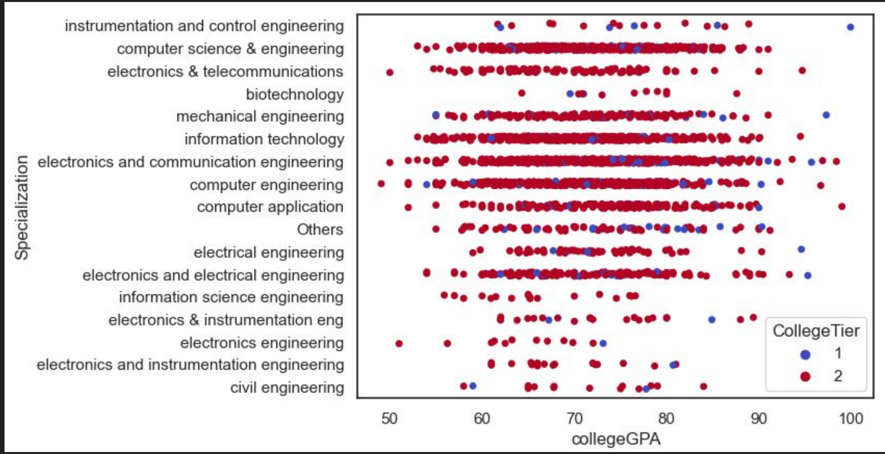

Distribution of CollegeTier
2500
2000
1500
1000
500
0
2
1

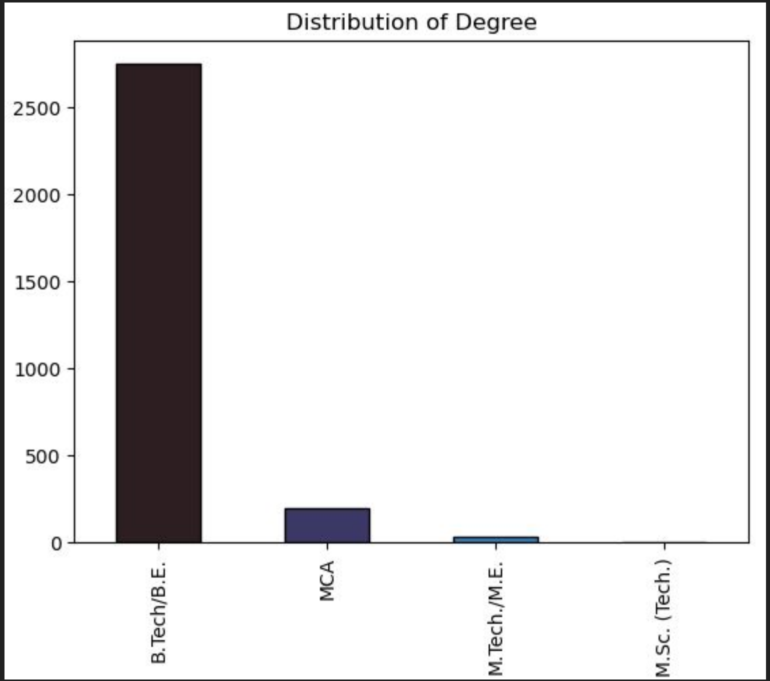

100
90
80
70
60
50
collegeGPA
CollegeTier
1
2
B.Tech/B.E.
M.Tech./M.E.
MCA
M.Sc. (Tech.)
Degree

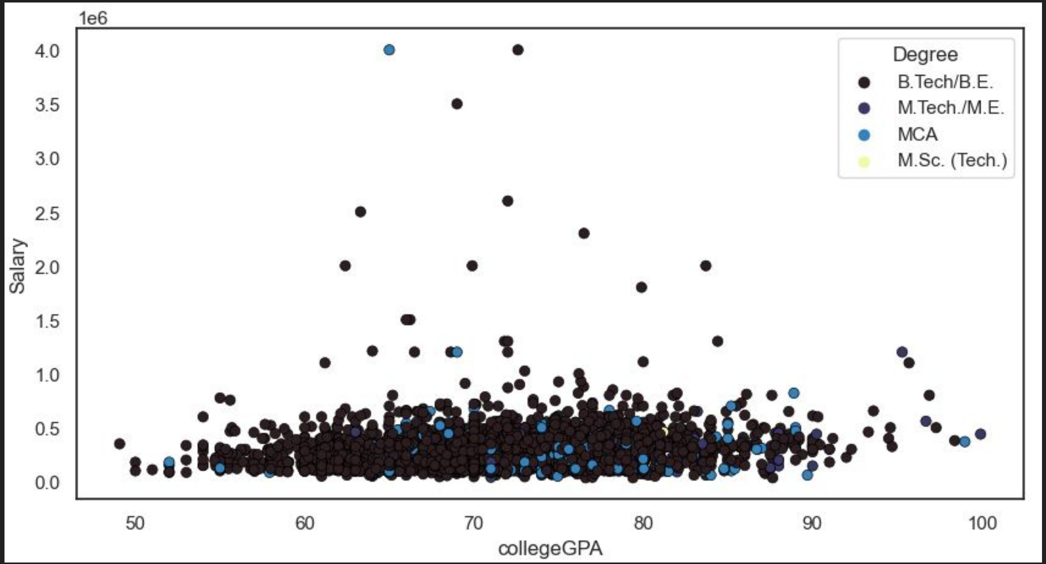

1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Salary
CollegeTier
1
2
B.Tech/B.E.
M.Tech./M.E.
MCA
M.Sc. (Tech.)
Degree

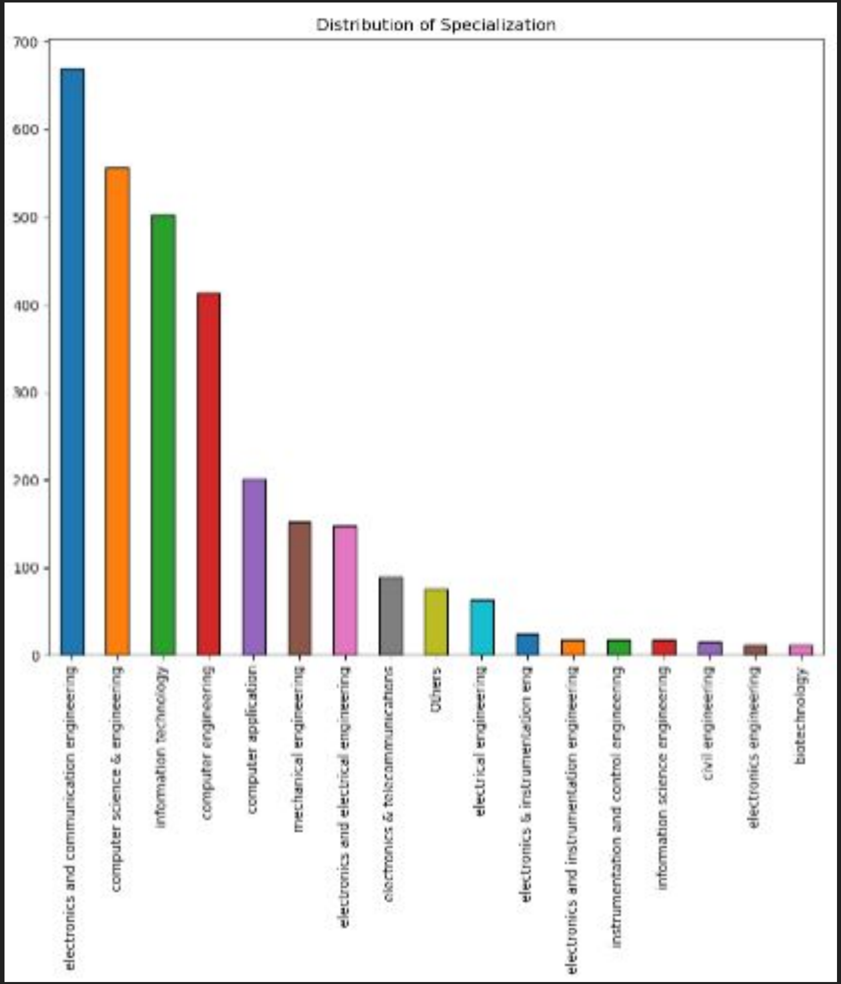

1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Salary
50
60
70
80
90
100
collegeGPA
CollegeTier
1
2

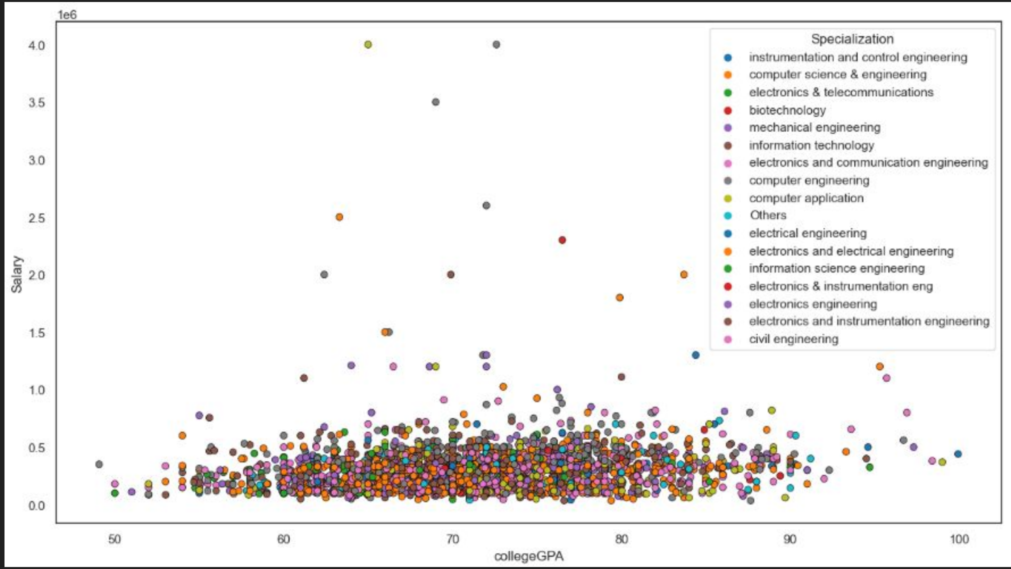

instrumentation and control engineering
computer science & engineering
electronics & telecommunications
biotechnology
mechanical engineering
information technology
electronics and communication engineering
computer engineering
computer application
Others
electrical engineering
electronics and electrical engineering
information science engineering
electronics & instrumentation eng
electronics engineering
electronics and instrumentation engineering
civil engineering
Specialization
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1e6
Salary
CollegeTier
1
2


instrumentation and control engineering
computer science & engineering
electronics & telecommunications
biotechnology
mechanical engineering
information technology
electronics and communication engineering
computer engineering
computer application
Others
electrical engineering
electronics and electrical engineering
information science engineering
electronics & instrumentation eng
electronics engineering
electronics and instrumentation engineering
civil engineering
Specialization
50
60
70
80
90
100
collegeGPA
CollegeTier
1
2


Distribution of Degree
2500
2000
1500
1000
500
0
B.Tech/B.E.
MCA
M.Tech./M.E.
M.Sc. (Tech.)


1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Salary
50
60
70
80
90
100
collegeGPA
Degree
B.Tech/B.E.
M.Tech./M.E.
MCA
M.Sc. (Tech.)


Distribution of Specialization
electronics and communication engineering
computer science & engineering
information technology
computer engineering
computer application
mechanical engineering
electronics and electrical engineering
electronics & telecommunications
Others
electrical engineering
electronics & instrumentation eng
electronics and instrumentation engineering
instrumentation and control engineering
information science engineering
civil engineering
electronics engineering
biotechnology


Specialization
instrumentation and control engineering
computer science & engineering
electronics & telecommunications
biotechnology
mechanical engineering
information technology
electronics and communication engineering
computer engineering
computer application
Others
electrical engineering
electronics and electrical engineering
information science engineering
electronics & instrumentation eng
electronics engineering
electronics and instrumentation engineering
civil engineering
Salary
collegeGPA
1e6

Feature Engineering

[1] Data Preprocessing

- We can see that Gender, Degree, and Specialization are 'object' variables, which aren't suitable for ML input. Hence, we need to process those variables.

```
Data columns (total 24 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Gender                 2989 non-null   object
 1   10percentage           2989 non-null   float64
 2   12percentage           2989 non-null   float64
 3   CollegeTier            2989 non-null   int64
 4   Degree                 2989 non-null   object
 5   Specialization         2989 non-null   object
 6   collegeGPA             2989 non-null   float64
 7   English                2989 non-null   int64
 8   Logical                2989 non-null   int64
 9   Quant                  2989 non-null   int64
 10  Domain                 2989 non-null   float64
 11  ComputerProgramming    2989 non-null   float64
 12  ElectronicsAndSemicon  2989 non-null   float64
 13  ComputerScience        2989 non-null   float64
 14  MechanicalEngg         2989 non-null   float64
 15  ElectricalEngg         2989 non-null   float64
 16  TelecomEngg            2989 non-null   float64
 17  CivilEngg              2989 non-null   float64
 18  conscientiousness      2989 non-null   float64
 19  agreeableness          2989 non-null   float64
 20  extraversion           2989 non-null   float64
 21  nueroticism            2989 non-null   float64
 22  openess_to_experience  2989 non-null   float64
 23  Salary                 2989 non-null   int64
dtypes: float64(16), int64(5), object(3)
memory usage: 583.8+ KB
```

```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df4 = df2.copy()
df4.Gender = le.fit_transform(df4.Gender)
df4.Degree = le.fit_transform(df4.Degree)
df4.Specialization = le.fit_transform(df4.Specialization)
```

	Gender	10percentage	12percentage	CollegeTier	Degree	Specialization
0	0	87.80	84.00	1	0	15
1	1	57.00	64.50	2	0	5
2	1	77.33	85.17	2	0	8
3	1	84.30	86.00	1	0	5
4	0	82.00	75.00	2	0	1

[2] Feature Scaling

- Normalize our data

```
norm_data = MinMaxScaler().fit_transform(df2_num)
norm_data
```

```
array([[0.81811541, 0.74957411, 0.        , ..., 0.46516554, 0.8510643 ,
        0.10340479],
       [0.25566107, 0.41737649, 1.        , ..., 0.52859645, 0.78723948,
        0.01891551],
       [0.62691746, 0.76950596, 1.        , ..., 0.39242765, 0.78706182,
        0.0554855 ],
       ...,
       [0.88385683, 0.43543441, 1.        , ..., 0.45122175, 0.89230393,
        0.08827238],
       [0.83345508, 0.4286201 , 1.        , ..., 0.19641898, 0.879157  ,
        0.12484237],
       [0.62089116, 0.60477002, 1.        , ..., 0.63432574, 0.55319291,
        0.04161412]])
```

- Standardize our data

```
scaled_data = StandardScaler().fit_transform(df2_num)
scaled_data
```

```
array([[ 1.01404346,  0.86990384, -3.51336733, ...,  0.28809347,
         0.42693845,  0.65899772],
       [-2.06798607, -0.88360688,  0.28462723, ...,  0.6636889 ,
        -0.1436    , -0.91886628],
       [-0.03364645,  0.97511448,  0.28462723, ..., -0.14261169,
        -0.14518814, -0.23591022],
       ...,
       [ 1.37428068, -0.78828784,  0.28462723, ...,  0.20552765,
         0.79558491,  0.37639521],
       [ 1.09809881, -0.82425729,  0.28462723, ..., -1.30324405,
         0.67806272,  1.05935127],
       [-0.06666819,  0.10555301,  0.28462723, ...,  1.28974713,
        -2.23577286, -0.49496252]])
```

[3] Polynomial Features

- Separate our predictor variables from target variable.

```
X = df2.loc[:,['Gender', '10percentage', '12percentage', 'CollegeTier', 'Degree', 'Specialization',
               'collegeGPA', 'English', 'Logical', 'Quant', 'Domain', 'ComputerProgramming', 'ElectronicsAndSemicon',
               'ComputerScience', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg', 'CivilEngg', 'conscientiousness',
               'agreeableness', 'extraversion', 'nueroticism', 'openess_to_experience']]

Y = df2['Salary']
```

- Apply polynomial calculation

```
X2 = X.copy()

X2['agree2'] = X2['agreeableness'] ** 2
X2['opennes2'] = X2['openess_to_experience'] ** 2
```

[3.1] Polynomial Features in Scikit-Learn

```
from sklearn.preprocessing import PolynomialFeatures
```

```
#Instantiate and provide desired degree;
#Note: degree=2 also includes intercept, degree 1 terms, and cross-terms

pf = PolynomialFeatures(degree=2)
```

```
features = ['agreeableness', 'openess_to_experience']
pf.fit(df2[features])
```

```
▾ PolynomialFeatures
PolynomialFeatures()
```

```
pf.get_feature_names_out() #Must add input_features = features for appropriate name
```

```
array(['1', 'agreeableness', 'openess_to_experience', 'agreeableness^2',
       'agreeableness openess_to_experience', 'openess_to_experience^2'],
      dtype=object)
```

```
feat_array = pf.transform(df2[features])
pd.DataFrame(feat_array, columns = pf.get_feature_names_out(input_features=features))
```

	1	agreeableness	openess_to_experience	agreeableness^2	agreeableness openess_to_experience	openess_to_experience^2
0	1.0	0.3789	0.2889	0.143565	0.109464	0.083463
1	1.0	0.0459	-0.2859	0.002107	-0.013123	0.081739
2	1.0	-0.1232	-0.2875	0.015178	0.035420	0.082656
3	1.0	0.2124	0.4805	0.045114	0.102058	0.230880
4	1.0	-0.7473	0.1864	0.558457	-0.139297	0.034745
...	...	...	...	...	...	...
2984	1.0	0.9688	0.0284	0.938573	0.027514	0.000807
2985	1.0	0.0328	0.5024	0.001076	0.016479	0.252406
2986	1.0	0.1888	0.6603	0.035645	0.124665	0.435996
2987	1.0	1.2808	0.5419	1.640449	0.694066	0.293656
2988	1.0	-1.9521	-2.3937	3.810694	4.672742	5.729800

2989 rows × 6 columns

[4] Getting Dummy Variables (One-Hot Encoding)

- We will create a new feature column for each category value, and fill these columns with 1s and 0s to indicate which category is present for each row. This method is called dummy variables or one-hot encoding. (Notice that before we have 24 columns, but now we have 44)

	Gender	10percentage	12percentage	CollegeTier	Degree	Specialization	collegeGPA	English	Logical	Quant	...	MechanicalEngg	ElectricalEngg
0	f	87.80	84.00	1	B.Tech/B.E.	instrumentation and control engineering	73.82	650	665	810	...	401.174863	423.336066
1	m	57.00	64.50	2	B.Tech/B.E.	computer science & engineering	65.00	440	435	210	...	401.174863	423.336066
2	m	77.33	85.17	2	B.Tech/B.E.	electronics & telecommunications	61.94	485	475	505	...	401.174863	423.336066
3	m	84.30	86.00	1	B.Tech/B.E.	computer science & engineering	80.40	675	620	635	...	401.174863	423.336066
4	f	82.00	75.00	2	B.Tech/B.E.	biotechnology	64.30	575	495	365	...	401.174863	423.336066

5 rows × 24 columns

```
pd.get_dummies(df2)
```

	10percentage	12percentage	CollegeTier	collegeGPA	English	Logical	Quant	Domain	ComputerProgramming	ElectronicsAndSemicon	...
0	87.80	84.00	1	73.82	650	665	810	0.694479	485.000000	366.000000	...
1	57.00	64.50	2	65.00	440	435	210	0.342315	365.000000	335.947917	...
2	77.33	85.17	2	61.94	485	475	505	0.824666	449.620837	400.000000	...
3	84.30	86.00	1	80.40	675	620	635	0.990009	655.000000	335.947917	...
4	82.00	75.00	2	64.30	575	495	365	0.278457	315.000000	335.947917	...
...	...	...	...	...	...	...	...	...	...	...	...
2993	75.00	73.00	2	70.00	505	485	445	0.538387	245.000000	333.000000	...
2994	84.00	77.00	2	75.20	345	585	395	0.190153	315.000000	335.947917	...
2995	91.40	65.56	2	73.19	385	425	485	0.600057	435.000000	335.947917	...
2996	88.64	65.16	2	74.81	465	645	505	0.901490	545.000000	335.947917	...
2997	77.00	75.50	2	69.30	370	390	285	0.486747	315.000000	335.947917	...

2989 rows × 44 columns

Result Example of One-Hot Encoding

	Degree
0	B.Tech/B.E.
1	B.Tech/B.E.
2	B.Tech/B.E.
3	B.Tech/B.E.
4	B.Tech/B.E.

Before One-Hot Encoding

	B.Tech/B.E.	M.Sc. (Tech.)	M.Tech./M.E.	MCA
0	1	0	0	0
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	0	0	0

After One-Hot Encoding

[5] Getting to Fancier Features

- We'll create features that capture where a feature value lies relative to the members of a category it belongs to. In particular, we'll calculate deviance of a row's feature value from the mean value of the category that row belongs to. This helps to capture information about a feature relative to the category's distribution,

```python
def add_deviation_feature(X, feature, category):

    #temp groupby object
    category_gb = X.groupby(category)[feature]

    #create category means and standard deviations for each observation
    category_mean = category_gb.transform(lambda x: x.mean())
    category_std = category_gb.transform(lambda x: x.std())

    #compute stds from category mean for each feature value,
    #add to X as new feature
    deviation_feature = (X[feature] - category_mean) / category_std
    X[feature + '_Dev_' + category] = deviation_feature
```

openess_to_experience_Dev_Degree 0.426733 (1st row) means that for this Degree, this was higher than the average #openess_to_experience for this Degree

#We can see in 2nd and 3rd rows (negatives), mean that they are below the average for that specific Degree

	Degree	openess_to_experience	openess_to_experience_Dev_Degree	collegeGPA	Specialization	collegeGPA_Dev_Specialization
0	B.Tech/B.E.	0.2889	0.426733	73.82	instrumentation and control engineering	-0.255174
1	B.Tech/B.E.	-0.2859	-0.143249	65.00	computer science & engineering	-1.014374
2	B.Tech/B.E.	-0.2875	-0.144835	61.94	electronics & telecommunications	-0.918264
3	B.Tech/B.E.	0.4805	0.616726	80.40	computer science & engineering	1.289844
4	B.Tech/B.E.	0.1864	0.325092	64.30	biotechnology	-1.698599
...	...	...	...	...	...	...

Hypothesis Testing

Hypothesis Testing 1

$H_0 : \mu_1 <= \mu_2$ The average salary of females are less than or equal to graduates from males.
$H_A : \mu_1 > \mu_2$ The average salary of females are greater than or equal to males.

Hypothesis Testing 2

$H_0 : \mu_1 - \mu_2 = 0$ There is no difference between the collegeGPA of graduates from CollegeTier 1 and CollegeTier 2.
$H_A : \mu_1 - \mu_2 ! = 0$ There is difference between the collegeGPA of graduates from CollegeTier 1 and CollegeTier 2.

Hypothesis Testing 3

$H_0 : \mu_1 = \mu_2 = \mu_3$ The mean Salary of graduates from every Degree are the same.
H_A : At least one of the Degrees's salary is not the same.

Conducting Hypothesis Testing 1

- Calculate the average salary for males and females

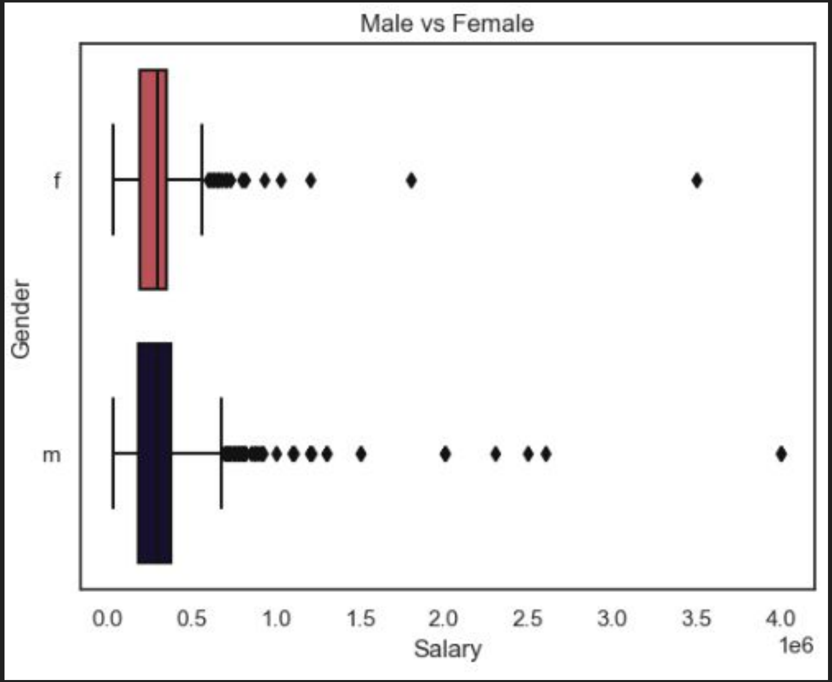
```
In [95]: male = df2.loc[df2.Gender == 'm']
         male_salary = male.Salary
         malesalary_mean = male_salary.mean()
         malesalary_mean

Out[95]: 309786.2796833773

In [96]: female = df2.loc[df2.Gender == 'f']
         female_salary = female.Salary
         femalesalary_mean = female_salary.mean()
         femalesalary_mean

Out[96]: 290139.86013986013
```

- Build a boxplot to visualize the distribution Salary by Gender



```python
alpha=0.05
t_val, p_value = stats.ttest_ind(male_salary, female_salary)
p_value_onetail = p_value/2
print("t_value = {} , p_value ={} , p_value_onetail = {}".format(t_val, p_value, p_value_onetail))
```

```
t_value = 2.159169682415574 , p_value =0.03091638149374018 , p_value_onetail = 0.01545819074687009
```

```python
# Enter your code and run the cell
if p_value <alpha:
    print("Conclusion: since p_value {} is less than alpha {} ". format (p_value_onetail,alpha))
    print("Reject the null hypothesis that the average salary of females are less than or equal to males.")

else:
    print("Conclusion: since p_value {} is greater than alpha {} ". format (p_value_onetail,alpha))
    print("Fail to reject the null hypothesis that the average salary of females are less than males.")
```

```
Conclusion: since p_value 0.01545819074687009 is less than alpha 0.05
Reject the null hypothesis that the average salary of females are less than or equal to males.
```

Suggestions

In getting rid of outliers, we can also try to conduct bi-variate analysis and Z-score analysis to make sure all the data outliers have been eliminated. In data exploratory analysis, the categories of AMCAT scores can be explored more in order to inspect the features that affect Salary. We can also do another feature engineering method, like PCA

Summary

Overall, in the terms of duplicate values and missing values, this data is in a very good quality because every feature doesn’t have any null-values. But in the term of outliers, there are a lot of outliers that we need to take care of from this data.

Thank You