Career Guidance Project

```
#Importing Database and reading the csv file
import pandas as pd
import numpy as py
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
data =pd.read_csv("career_dataset_student.csv")

print(data.head())
```

```
   EntryID First Name Last Name  Gender Date of Birth  \
0        1      Riley    Martin  Female    03-08-1995
1        2    Cameron   Jackson  Female    15-04-1976
2        3     Skylar    Martin    Male    11-11-1992
3        4      Rowan     Davis    Male    10-09-1984
4        5      Rowan  Gonzalez  Female    24-08-1993

                        Email  Phone Number        Location  \
0     Riley.Martin.@email.com  874-181-5824         Chicago
1  Cameron.Jackson.@email.com  885-476-8589    Rhode Island
2    Skylar.Martin.@email.com  634-634-6837       Hong Kong
3      Rowan.Davis.@email.com  604-631-1668         Chicago
4   Rowan.Gonzalez.@email.com  835-256-7470   Fontainebleau

                    Major                        Institution  \
0   Information Technology  University of California, Berkeley
1  Business Administration             Harvard Business School
2                  Nursing                             INSEAD
3                  Finance              London Business School
4               Psychology                             INSEAD

  Graduation Date   GPA             Company               Job Title  \
0      19-08-2020  7.59                Uber   Information Technology
1      16-03-2021  7.32  McKinsey & Company  Business Administration
2      11-02-2017  7.37           Accenture                 Nursing
3      09-06-2021  7.81       Goldman Sachs                 Finance
4      06-10-2021  7.54           Coca-Cola               Psychology

  Job Start Date        Skill   Experience   Salary
0     18-11-2017      Beginner           9  1258594
1     13-01-2016      Advanced           4  4707572
2     08-03-2017      Advanced           8  2720362
3     21-09-2018      Advanced           6   589047
4     17-07-2017  Intermediate           9  2486789
```

Data Preprocessing

- Load the Data with the help of pandas library
- print the data using function
- Visualize the correlation map to understand the relation with columns
- Check for nul values and if data contains any value remove them
- Additionally,inspect for dulpilcate value sand remove them if present

```
# about the dataset
dataset = data.info()
```

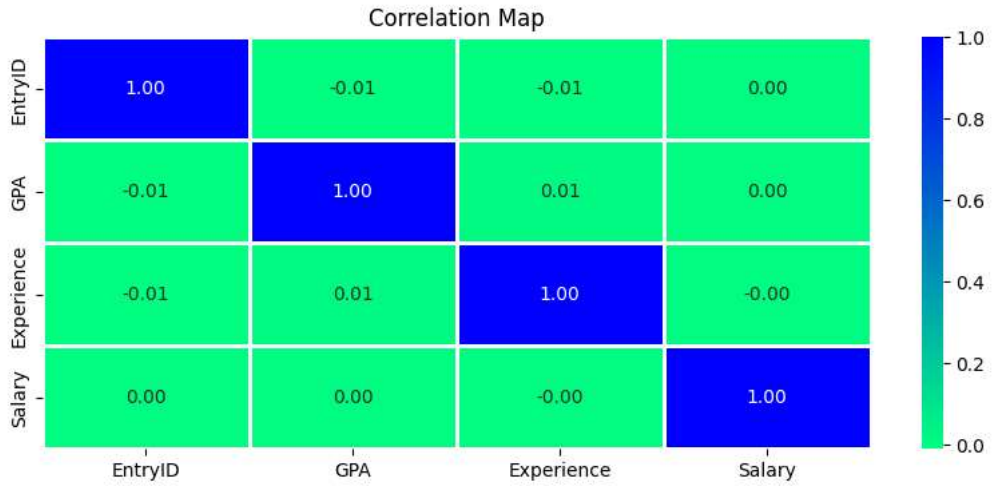
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22456 entries, 0 to 22455
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   EntryID          22456 non-null  int64
 1   First Name       22456 non-null  object
 2   Last Name        22456 non-null  object
 3   Gender           22456 non-null  object
 4   Date of Birth    22456 non-null  object
 5   Email            22456 non-null  object
 6   Phone Number     22456 non-null  object
 7   Location         22456 non-null  object
 8   Major            22456 non-null  object
 9   Institution      22456 non-null  object
 10  Graduation Date  22456 non-null  object
```

```
 11  GPA              22456 non-null  float64
 12  Company          22456 non-null  object
 13  Job Title        22456 non-null  object
 14  Job Start Date   22456 non-null  object
 15  Skill            22456 non-null  object
 16  Experience       22456 non-null  int64
 17  Salary           22456 non-null  int64
dtypes: float64(1), int64(3), object(14)
memory usage: 3.1+ MB
```

```
# understanding the stastics int he data set
data_stastics=data.describe().style.background_gradient(cmap='tab20c')
print(data_stastics)
```

```
<pandas.io.formats.style.Styler object at 0x79be072b5d80>
```

```
# correlation map
plt.figure(figsize=(10,4))
sns.heatmap(data.corr(),annot=True,cmap='winter_r',fmt='.2f',linewidths=1)
plt.title("Correlation Map")
plt.show()
```



Data Cleaning

- Removing the null values
- Removing the duplicate

```
#  null values
a=data.isna()
print(a)
```

```
       EntryID  First Name  Last Name  Gender  Date of Birth  Email  \
0        False       False      False   False          False  False
1        False       False      False   False          False  False
2        False       False      False   False          False  False
3        False       False      False   False          False  False
4        False       False      False   False          False  False
...        ...         ...        ...     ...            ...    ...
22451    False       False      False   False          False  False
22452    False       False      False   False          False  False
22453    False       False      False   False          False  False
22454    False       False      False   False          False  False
22455    False       False      False   False          False  False

       Phone Number  Location  Major  Institution  Graduation Date    GPA  \
0             False     False  False        False            False  False
1             False     False  False        False            False  False
2             False     False  False        False            False  False
3             False     False  False        False            False  False
4             False     False  False        False            False  False
...             ...       ...    ...          ...              ...    ...
22451         False     False  False        False            False  False
22452         False     False  False        False            False  False
22453         False     False  False        False            False  False
22454         False     False  False        False            False  False
22455         False     False  False        False            False  False
```

```
       Company  Job Title  Job Start Date  Skill   Experience  Salary
0        False      False           False   False       False   False
1        False      False           False   False       False   False
2        False      False           False   False       False   False
3        False      False           False   False       False   False
4        False      False           False   False       False   False
...        ...        ...             ...     ...         ...     ...
22451    False      False           False   False       False   False
22452    False      False           False   False       False   False
22453    False      False           False   False       False   False
22454    False      False           False   False       False   False
22455    False      False           False   False       False   False

[22456 rows x 18 columns]
```

```
# droping the values
b=data.dropna()
print(b.isna().sum().sum())
```

```
0
```

```
# Checking the duplicate value
duplicate_values= data.duplicated().sum()
print(duplicate_values)
```

```
0
```

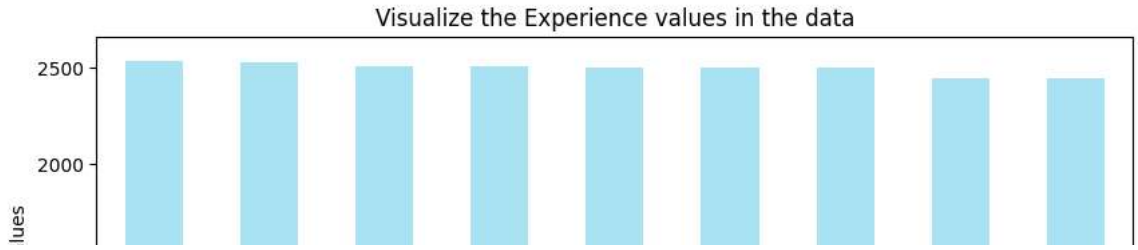
```
# remove the duplicate values and store
data = data.drop_duplicates()
after_remove_duplicates=data.duplicated().sum()
print(after_remove_duplicates)
```

```
0
```

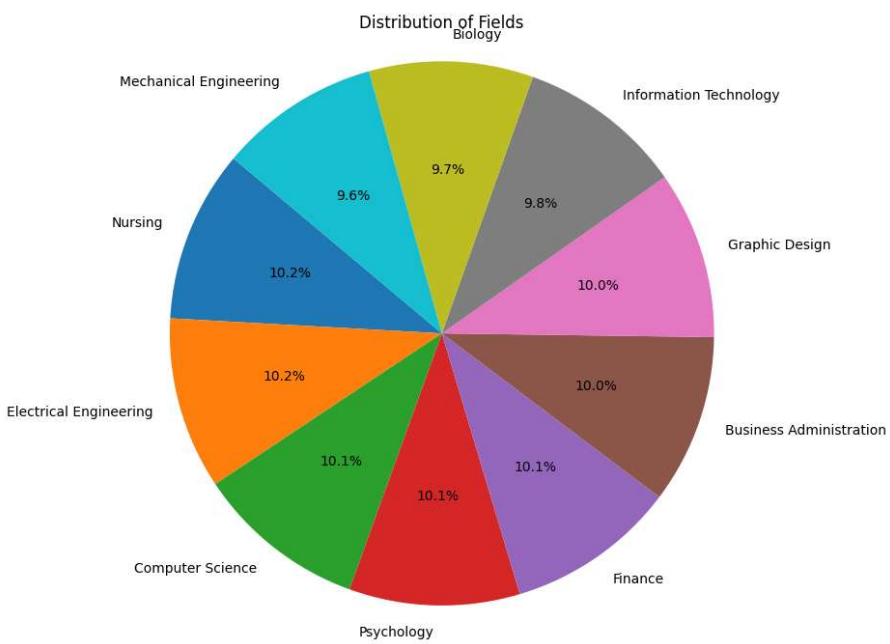
Data Analysis Process

- Created the bar graph to visulaoze the Experience of the people are there in the years.
- To visulize the the data of the people who has preferred the their role and on that basis there salary is there.
- From the distribution of the skill we can know how many people have their skill such as begginer,intermediate and andvanced.
- We are comparing the how many male and female have acquired which branched.
- We can visualize the salary chart know which company is paying more amount to their respective field.
- We can predict the on basis of the given data to what we have to acquire in future.

```
#Experience
import matplotlib.pyplot as plt
import pandas as pd
data =pd.read_csv("career_dataset_student.csv")
plt.figure(figsize=(10,5))
data['Experience'].value_counts().sort_values(ascending=False).plot(kind='bar',color=['#A9E2F3'])
plt.title("Visualize the Experience values in the data")
plt.ylabel("Count of the values")
plt.xlabel("Experience")
plt.show()
```

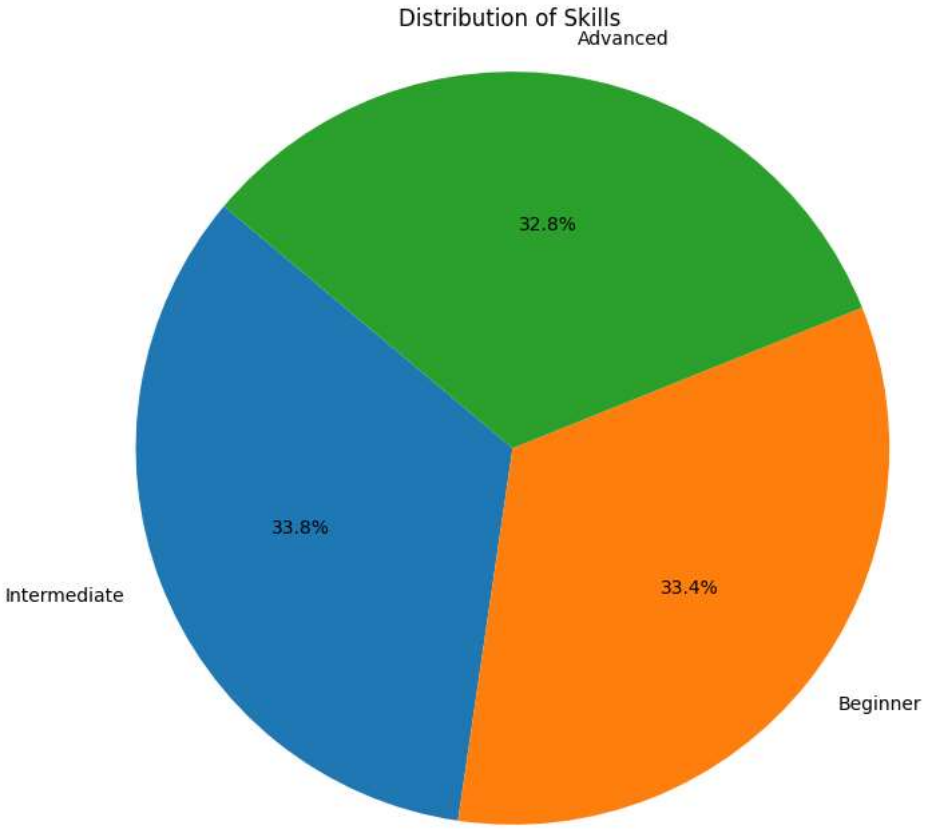



```
# Distribution Degree and Salary
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('career_dataset_student.csv')

# Assuming your CSV has columns 'Major' and 'Salary'
major_counts = df['Major'].value_counts()
average_salary = df.groupby('Major')['Salary'].mean()

# Create a pie chart for the 'Major' distribution
plt.figure(figsize=(8, 8))
plt.pie(major_counts, labels=major_counts.index, autopct='%1.1f%%', startangle=140)
plt.title("Distribution of Fields")
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```

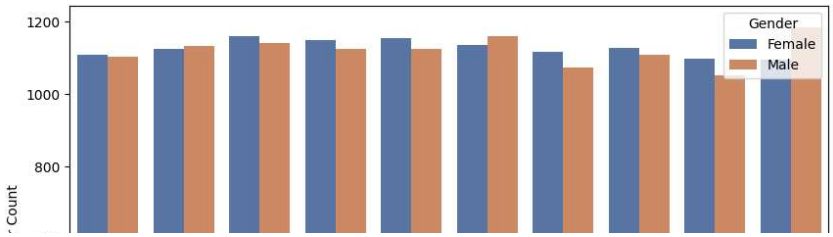



```
#Distribution of Skills
import pandas as pd
import matplotlib.pyplot as plt
# Load the CSV data into a DataFrame
df = pd.read_csv('career_dataset_student.csv')

# Assuming your CSV has columns 'Major' and 'Salary'
major_counts = df['Skill '].value_counts()
average_salary = df.groupby('Skill ')['Salary'].mean()

# Create a pie chart for the 'Major' distribution
plt.figure(figsize=(8, 8))
```

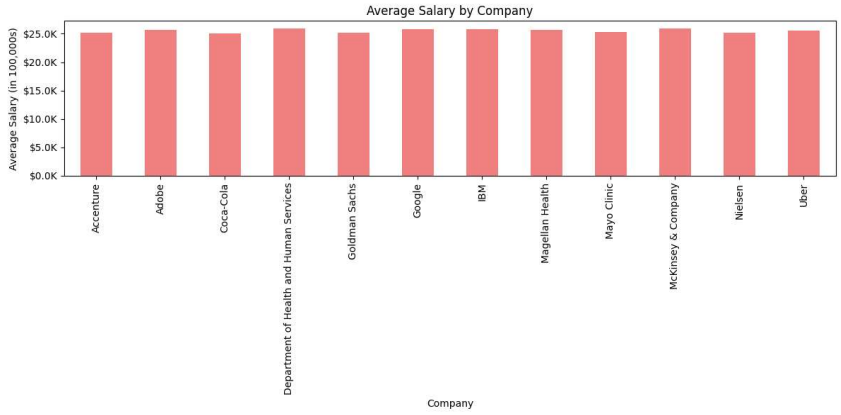
```
plt.pie(major_counts, labels=major_counts.index, autopct='%1.1f%%', startangle=140)
plt.title("Distribution of Skills ")
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```




```
# Comparing the male and female on basis of field selection
import seaborn as sns
plt.figure(figsize=(10,6))
sns.countplot(data=data,x='Major',hue='Gender',palette='deep')
plt.xticks(rotation=90)
plt.ylabel("Gender Count")
plt.show()
```




```
# Company Salary based distribution
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('career_dataset_student.csv')
from matplotlib.ticker import FuncFormatter
# Assuming your CSV has columns 'Company' and 'Salary'
company_salaries = df.groupby('Company')['Salary'].mean()

# Create a custom y-axis formatter
def salary_formatter(x, pos):
    return f'${x/100000:.1f}K'

# Create the bar chart
plt.figure(figsize=(12, 6))
ax = company_salaries.plot(kind='bar', color='lightcoral')
ax.yaxis.set_major_formatter(FuncFormatter(salary_formatter))
plt.title("Average Salary by Company")
plt.xlabel("Company")
plt.ylabel("Average Salary (in 100,000s)")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



Observations:

- We observed that the if you are taking Computer Science as the career option you are highest paid job in google and IBM.
- The pie chart illustrate the skills that are hired more for the job title you preferred
- Significant portion of the paid high salary from this data model.
- From the double bar graph of male and female we can obserce that graphic designer are more preferred by the female and Computer Science field is more prefferable by the Male.
- From skill pie chart we observed how many people are going in which field and which field is more prefferable for further studies.
- The Experience graph is shown how many people are experienced in their field sector.

**Machine Learning Model **

- In this model we have use the decision tree model in this.

- Using the data Experience ,GPA and the Salary we have find the mean absoulte error , root squared error and root mean squared error.
- We have plotted the scatter plot of it.

```
# Fitting the Machine learning model using the Decision Tree

import pandas as pd
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error

# Load the dataset
data = pd.read_csv('career_dataset_student.csv')
# feature selection
X = data[['Experience', 'GPA']]
y = data['Salary']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Create the Decision Tree Regressor model
model = DecisionTreeRegressor(max_depth=None, min_samples_leaf=1, random_state=42)

# Fit the model to the training data
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5  # Calculate RMSE as the square root of MSE

print(f"Mean Absolute Error: {mae}")
print(f"Mean Squared Error: {mse}")
print(f"Root Mean Squared Error: {rmse}")
plt.scatter(y_test, y_pred, label='Scatter Plot', color='blue')

# Add MAE, MSE, and RMSE as text annotations on the plot
plt.text(0.1, 0.9, f"MAE: {mae:.2f}", transform=plt.gca().transAxes)
plt.text(0.1, 0.85, f"MSE: {mse:.2f}", transform=plt.gca().transAxes)
plt.text(0.1, 0.80, f"RMSE: {rmse:.2f}", transform=plt.gca().transAxes)

# Set labels, title, legend, etc. as needed
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.legend()
plt.show()
```

```
Mean Absolute Error: 1274558.11023497
Mean Squared Error: 2256140770485.467
Root Mean Squared Error: 1502045.5287658449
```