**Project Title: DiverseTalent\_Applications**

**Introduction to the Dataset**

The "Student Career Choices" dataset is a synthetic dataset created to explore how students from diverse backgrounds pursue data-related roles. It contains information about 10,000 applicants with different educational backgrounds, technical skills, experience levels, and application statuses for various data-related roles.

**Overview of the Dataset:**

* **Total Rows:** 10,000
* **Total Columns:** 12

**Columns in the Dataset**

1. **Age**: Age of the applicant (ranging from 18 to 60).
2. **Gender**: Gender of the applicant (Male, Female, Non-Binary).
3. **Region**: Geographic region where the applicant is located (e.g., North America, Europe, Asia, etc.).
4. **Educational\_Background**: Field of study (e.g., Engineering, Business, Arts, Science, Commerce).
5. **Technical\_Skills**: List of technical skills possessed by the applicant (e.g., Python, SQL, Tableau).
6. **Experience\_Level**: Level of experience (Fresher, Mid-level, Senior).
7. **Preferred\_Job\_Role**: The job role the applicant is interested in (e.g., Data Scientist, Data Analyst, Data Engineer, Business Analyst).
8. **Salary\_Expectation**: Expected salary in USD (Contains extreme values to mimic unrealistic expectations).
9. **Application\_Source**: The platform where the application was submitted (e.g., LinkedIn, Company Website, Referral, Job Portal).
10. **Application\_Status**: Status of the application (Accepted, Rejected). **Imbalanced** to reflect real-world scenarios.
11. **Interview\_Score**: Score obtained in the interview (scale of 1 to 10).
12. **Certifications\_Count**: Number of certifications held by the applicant (0 to 4).

**Business Questions to Answer:**

**Section 1: Data Understanding:**

Q1: What are the first 5 rows of the dataset?

df.head()

Q2: What is the shape of the dataset?

df.shape

Q3: What are the column names of the dataset?

df.columns

Q4: What is the data type of each column?

df.dtypes

Q5: Are there any missing (null) values in the dataset?

df.isnull().sum()

Q6: What is the percentage of missing values for each column?

percentage\_missing=df.isnull().sum()\*100/len(df)

percentage\_missing

Q7: What is the summary of the dataset?

df.info()

Q8: What are the basic statistics of numerical columns?

df.describe()

Q9: What is the distribution of the 'Application\_Status' column?

df['Application\_Status'].value\_counts()

Q10: What are the unique values in the 'Preferred\_Job\_Role' column?

df['Preferred\_Job\_Role'].unique()

**Section 2: Data Visualization and Analysis:**

Q11: How many duplicate rows are present in the dataset?

df.duplicated().sum()

Q12: Display the duplicate rows, if any.

df[df.duplicated()]

Q13: What is the distribution of the 'Gender' column?

df[‘Gender’].value\_counts()

Q14: What are the unique values in the 'Educational\_Background' column?

df['Educational\_Background'].unique()

Q15: Visualize the distribution of 'Experience\_Level'. (Countplot)

plt.figure(figsize = (10, 5))

sns.countplot(x = 'Experience\_Level', data = df)

plt.title("Distribution of Experience Level")

plt.xlabel("Experience\_Level")

plt.ylabel("Frequency")

plt.show()

Q16: Visualize the distribution of 'Application\_Status'. (Countplot)

plt.figure(figsize= (8,5))

sns.countplot(x = 'Application\_Status' ,data=df)

plt.title('Distribution of Application Status')

plt.xlabel("Application\_Status")

plt.ylabel("Frequency")

plt.show()

Q17: Find the correlation between numerical columns.

correlation\_matrix=df[numerical\_columns].corr()

print(correlation\_matrix)

Q18: Visualize the correlation matrix using a heatmap.

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

sns.heatmap(correlation\_matrix.corr(),annot=True)

Q19: Visualize the distribution of 'Interview\_Score'. (Histogram Plot)

sns.histplot(df.Interview\_score)

Q20: Plot a box plot to identify extreme values in 'Salary\_Expectation'.

sns.boxplot(df['Salary\_Expectation'])

**Section 3: Analysis Using np.where() Function:**

Q21: Create a new column 'High\_Salary' where the value is 'Yes' if the 'Salary\_Expectation' is above 100,000, otherwise 'No'.

df['High\_Salary']=np.where(df['Salary\_Expectation']<=100000,"Yes","No")

df['High\_Salary'].value\_counts()

Q22: Create a column 'Experienced' with value 'Yes' if 'Experience\_Level' is 'Senior', otherwise 'No'.

df['Experienced']=np.where(df['Experience\_Level']=="Senior","Yes","No")

Q23: Create a column 'Good\_Score' where the value is 'Yes' if 'Interview\_Score' is above 80, otherwise 'No'.

df['Good\_Score']=np.where(df['Interview\_Score']==80,"Yes","No")

Q24: Create a column 'Certified' where the value is 'Yes' if 'Certifications\_Count' is 2 or more, otherwise 'No'.

df['Certified']=np.where(df['Certifications\_Count']<=2,"Yes","No")

Q25: Create a column 'Technical\_Background' where the value is 'Yes' if 'Educational\_Background' is 'Computer Science' or 'Information Technology'. Otherwise, 'No'.

df['Techanical\_Background']=np.where(df['Educational\_Background']=="Computer Science","Yes" or "Information Technology","No")