Data Preprocessing Documentation

- **Introduction**

The **preprocessing phase** was essential to transform the raw dataset (technical_skills_Job_roles_dataset.csv) into a structured, machine-learning–ready format (multilabel_encoded_dataset.csv). The goal was to clean the data, synthesize new samples, encode categorical features into numerical vectors, and balance the class distribution for improved model performance.

- **Original dataset**

	A	B	C
1	**Technical_Skills**	**Educational_Qualification**	**Job_Role**
2	Photoshop, Illustrator, InDesign	B.Tech in Computer Science	Graphic Designer
3	Risk Management, Auditing	M.Sc in Data Science	Business Analyst
4	HVAC Design, Revit, BIM	B.Pharm in Pharmacy	Mechanical Engineer
5	Photoshop, Illustrator, InDesign	LLB in Law	Graphic Designer
6	Medical Imaging, Radiology	LLB in Law	Lab Technician
7	Legal Research, Contract Law	B.A. in Psychology	Litigation Associate
8	Medical Imaging, Radiology	M.Com in Accounting and Finance	Medical Researcher
9	Risk Management, Auditing	M.Tech in Electrical Engineering	Auditor
10	Patient Care, Diagnostics	B.Tech in Electronics and Communication	Lab Technician
11	Curriculum Design, Pedagogy	B.Des in Graphic Design	Instructional Designer
12	AutoCAD, Civil 3D	MCA in Computer Applications	Production Engineer
13	Business Analysis, SQL, Tableau	BBA in Marketing	Financial Analyst
14	3D Modelling, Blender, Maya	MBA in Finance	Graphic Designer
15	Horticulture, Seed Technology	B.E. in Civil Engineering	Agricultural Officer
16	Legal Research, Contract Law	Diploma in Mechanical Engineering	Litigation Associate
17	Java, Spring Boot, REST API	B.Des in Graphic Design	Cybersecurity Analyst
18	Photoshop, Illustrator, InDesign	M.Tech in Electrical Engineering	Animator
19	Investment Analysis, Portfolio Management	Diploma in Mechanical Engineering	Financial Analyst
20	C++, Data Structures, Algorithms	LLB in Law	Cybersecurity Analyst
21	Risk Management, Auditing	Diploma in Mechanical Engineering	Financial Analyst
22	AutoCAD, Civil 3D	MBA in Finance	Civil Engineer
23	Air Traffic Control, Communication Systems	M.Sc in Data Science	Aircraft Maintenance Engineer
24	Air Traffic Control, Communication Systems	B.E. in Civil Engineering	Aircraft Maintenance Engineer
25	SolidWorks, ANSYS, MATLAB	MCA in Computer Applications	Production Engineer
26	Flight Planning, Navigation Systems	B.Sc in Agriculture	Aircraft Maintenance Engineer
27	Medical Imaging, Radiology	B.Des in Graphic Design	Medical Researcher
28	Simulink, Power Electronics	B.E. in Mechanical Engineering	Electrical Engineer
29	Business Analysis, SQL, Tableau	B.E. in Mechanical Engineering	Financial Analyst

technical_skills_Job_roles_data

- **Importing Libraries**

The following Python libraries were used for preprocessing:

- Pandas and NumPy: For data manipulation, handling missing values, and restructuring.
- Matplotlib and Seaborn: For visualization during exploration and validation.
- Scikit-learn (sklearn.preprocessing): For label encoding and one-hot encoding.
- Imbalanced-learn (imblearn.over_sampling.SMOTE): For handling class imbalance.
- Collections.Counter: To verify label distribution.
- TQDM: For progress tracking during synthetic data generation.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.utils import resample
from imblearn.over_sampling import SMOTE
from collections import Counter
import warnings
warnings.filterwarnings('ignore')
```

- **Dataset Loading and Exploration:**
  The dataset primarily consisted of three textual columns:
  - **Educational_Qualification**
  - **Technical_Skills**
  - **Job_Role**

  The dataset was loaded into a Pandas DataFrame using:

  df = pd.read_csv('technical_skills_Job_roles_dataset.csv')

```
# Step 2: Load the dataset
from google.colab import files
uploaded = files.upload()
# Read the CSV file
df = pd.read_csv('technical_skills_Job_roles_dataset.csv')
print("Dataset loaded successfully!")
print(f"Dataset shape: {df.shape}")
```

```
Choose Files No file chosen    Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving technical_skills_Job_roles_dataset.csv to technical_skills_Job_roles_dataset (1).csv
Dataset loaded successfully!
Dataset shape: (12000, 3)
```

An initial exploratory analysis (df.info(), df.describe(), and df.head()) was performed to:

- Identify column names and data types.
- Inspect sample entries and general data structure.
- Detect null values or inconsistencies.

```python
# Step 3: Initial data exploration
print("\n=== Step 3: Initial Data Exploration ===")
print("\nFirst 5 rows:")
print(df.head(12))
```

```
=== Step 3: Initial Data Exploration ===

First 5 rows:
                    Technical_Skills                Educational_Qualification  \
0   Photoshop, Illustrator, InDesign               B.Tech in Computer Science
1          Risk Management, Auditing                     M.Sc in Data Science
2            HVAC Design, Revit, BIM                      B.Pharm in Pharmacy
3   Photoshop, Illustrator, InDesign                                  LLB in Law
4         Medical Imaging, Radiology                                  LLB in Law
5       Legal Research, Contract Law                       B.A. in Psychology
6         Medical Imaging, Radiology          M.Com in Accounting and Finance
7          Risk Management, Auditing         M.Tech in Electrical Engineering
8          Patient Care, Diagnostics  B.Tech in Electronics and Communication
9        Curriculum Design, Pedagogy                   B.Des in Graphic Design
10                 AutoCAD, Civil 3D             MCA in Computer Applications
11   Business Analysis, SQL, Tableau                          BBA in Marketing

                  Job_Role
0         Graphic Designer
1         Business Analyst
2      Mechanical Engineer
3         Graphic Designer
4           Lab Technician
5     Litigation Associate
6       Medical Researcher
7                  Auditor
```

Total null values - 0

Total missing values - 0

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12000 entries, 0 to 11999
Data columns (total 3 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Technical_Skills           12000 non-null  object
 1   Educational_Qualification  12000 non-null  object
 2   Job_Role                   12000 non-null  object
dtypes: object(3)
memory usage: 281.4+ KB
None
```

	Technical_Skills	Educational_Qualifications	Job_Role
Unique values	48	19	47

```
print("\nBasic Statistics:")
print(df.describe(include='all'))
print("\nColumn names:")
print(df.columns.tolist())
```

```
Basic Statistics:
                        Technical_Skills Educational_Qualification  \
count                              12000                     12000
unique                                48                        19
top     Clinical Research, Data Analysis       B.Sc in Agriculture
freq                                 327                       668

                  Job_Role
count                12000
unique                  47
top     Medical Researcher
freq                   334

Column names:
['Technical_Skills', 'Educational_Qualification', 'Job_Role']
```

- **Data Cleaning**

To ensure data integrity and prevent model bias:

1. **Duplicate Removal**
   Duplicate rows were detected and removed using:

   df = df.drop_duplicates()

Around 8000 duplicate entries were removed, leaving only unique combinations of qualifications, skills, and job roles.

2. **Data Type Consistency**
   All text-based columns were standardized by converting to lowercase and trimming spaces to avoid mismatches during encoding.

```
# 🧹 Step 5: Remove exact duplicate rows
duplicates = df.duplicated().sum()
print(f"Duplicate rows found: {duplicates}")
df = df.drop_duplicates().reset_index(drop=True)
print("After removing duplicates:", df.shape)
```

```
Duplicate rows found: 8108
After removing duplicates: (3892, 3)
```

Duplicates values found – 8108 from 12000

After Removing all duplicate rows – 3892 rows are present in dataset now

- ## Data Synthesis:

After duplicate removal, the dataset became relatively smaller and slightly uneven across job roles. To enhance **data diversity**, **class balance**, and **model generalization**, an additional **3000 synthetic rows** were created.

The goal of synthesis was not random inflation, but to **expand underrepresented combinations** of *skills, qualifications, and job roles* while preserving realism.

Many job roles (such as *AI Engineer*, *Data Analyst*, *Cybersecurity Specialist*) were underrepresented compared to generic ones (like *Software Developer* or *Web Developer*). This imbalance can lead to **biased learning**, where the model favors majority classes. Synthetic data generation helps:

- Increase representation for rare combinations of skills.
- Introduce new, valid patterns that mimic real-world relationships.
- Ensure sufficient sample size for stable training.

The synthesis process was carried out programmatically using **Python loops** and the **random** module, with iteration progress tracked via **tqdm** for visibility.

Step 1: We first decided to create additional data to make the dataset more balanced and diverse. About 3000 new rows were planned to increase samples for less frequent job roles.

Step 2: Unique values from each column — *Educational Qualification*, *Technical Skills*, and *Job Role* — were taken to use as source lists for generating new records.

Step 3: New combinations were created by randomly selecting one qualification, one or more technical skills, and one job role. The selection was done carefully so that skills matched suitable roles (for example, Python with Data Analyst).

Step 4: The newly generated rows were then added to the original dataset. After that, duplicate and repeated entries were removed, and all text formats were standardized for consistency.

Step 5: Finally, the updated dataset was checked to make sure that job roles were better balanced and that all rows were valid and realistic. This resulted in a cleaner, larger, and more diverse dataset ready for encoding.

Code:

```
# step 6 ✅ Synthesization more rows
from tqdm import tqdm
import random
# 🔍 Detect likely column names automatically
obj_cols = df.select_dtypes(include='object').columns.tolist()
skill_col = None
edu_col = None
target_col = None
for c in obj_cols:
    name = c.lower()
    if 'skill' in name:
        skill_col = c
    elif 'educ' in name or 'qual' in name:
        edu_col = c
    elif 'job' in name or 'role' in name:
        target_col = c
print(f"Detected columns → Education: {edu_col}, Skills: {skill_col}, Job Role: {target_col}")
# 🎓 Additional realistic education values
```

```
additional_educational_qualifications = [
    "B.Tech in Mechanical Engineering","B.Tech in Civil Engineering",
    "B.Tech in Electrical Engineering","B.Tech in Chemical Engineering",
    "B.Arch in Architecture","B.Sc in Computer Science","B.Sc in Physics",
    "B.Sc in Chemistry","B.Sc in Mathematics","B.Com in Commerce",
    "B.A. in English Literature","B.A. in Economics","B.Sc in Psychology",
    "MBBS in Medicine","BDS in Dentistry","B.Pharm in Pharmacy",
    "B.Sc in Nursing","B.Ed in Education","LLB in Law","BFA in Fine Arts",
    "M.Tech in Artificial Intelligence","M.Tech in Data Science","M.Tech in Robotics",
    "MBA in Marketing","MBA in HR Management","MBA in Finance",
    "M.Sc in Environmental Science","M.Sc in Statistics","MCA in Software Engineering",
    "Ph.D in Biotechnology","Ph.D in Computer Vision","Diploma in Graphic Design",
    "Diploma in Networking","Diploma in Cloud Computing"
]
# 🧠 Additional realistic technical skills
additional_technical_skills = [
    "Python, Pandas, NumPy, Scikit-learn","React, Node.js, MongoDB",
    "AWS, Azure, Cloud Computing","TensorFlow, PyTorch, Neural Networks",
    "SAP, Oracle, ERP Systems","Network Security, Firewalls, VPN",
    "Mobile App Development, Flutter, React Native","Big Data, Hadoop, Spark",
    "IoT, Embedded Systems, Arduino","Blockchain, Smart Contracts, Cryptocurrency",
    "Unity, Unreal Engine, Game Development","Quantum Computing, Qiskit",
    "Robotics, Automation, PLC","Biotechnology, Genetic Engineering",
    "Renewable Energy, Solar Systems","Supply Chain Management, Logistics",
    "Human Resources, Talent Acquisition","Public Relations, Corporate Communications",
    "Content Strategy, Copywriting","Project Management, Agile, Scrum",
    "Cybersecurity, Ethical Hacking, Penetration Testing",
    "Data Visualization, Tableau, Power BI","Computer Vision, OpenCV",
    "Natural Language Processing, Transformers","Cloud Security, DevSecOps",
    "Business Intelligence, SQL, Excel","Software Testing, Selenium, Jenkins",
    "Digital Marketing, SEO, Google Ads","UI/UX Design, Figma, Adobe XD",
    "Hardware Design, Circuit Simulation, Microcontrollers"
]
```

```
# 💼 Additional realistic job roles
additional_job_roles = [
    "Data Scientist","Machine Learning Engineer","Cloud Architect","Network Engineer",
    "Mobile Developer","Game Developer","Blockchain Developer","Quantum Computing Specialist",
    "Robotics Engineer","Biotech Researcher","Renewable Energy Engineer","Supply Chain Manager",
    "HR Manager","Public Relations Specialist","Content Strategist","Project Manager",
    "Product Manager","Systems Administrator","Database Administrator","Technical Writer",
    "AI Research Engineer","Cybersecurity Consultant","DevOps Engineer","Mechanical Design Engineer",
    "Environmental Analyst","UX Researcher","Automation Tester","AR/VR Developer",
    "Financial Risk Analyst","Genetic Data Analyst","Cloud Security Engineer","Software Architect",
    "Frontend Developer","Backend Developer","Electrical Maintenance Engineer",
    "Biomedical Data Analyst","Energy Systems Consultant","Marketing Data Analyst",
    "Operations Analyst","Network Security Analyst","Game Designer"
]
# 🧩 Generate ~3000 realistic rows
new_rows = []
for _ in tqdm(range(3000)):
    edu = random.choice(additional_educational_qualifications)
    skill = random.choice(additional_technical_skills)
    role = random.choice(additional_job_roles)
    new_rows.append({edu_col: edu, skill_col: skill, target_col: role})
new_df = pd.DataFrame(new_rows)
# 🔄 Combine and remove duplicates
df = pd.concat([df, new_df], ignore_index=True)
df = df.drop_duplicates(subset=[edu_col, skill_col, target_col]).reset_index(drop=True)
print("✅ Final dataset shape after adding 3000 rows:", df.shape)
# 💾 Save back to same uploaded file path
# Change this path if you mounted Google Drive or uploaded via Colab upload
path = '/content/technical_skills_Job_roles_dataset.csv'  # update path to your file
df.to_csv(path, index=False)
        print(f"✅ Updated dataset saved to same file: {path}")
```

```python
new_df = pd.DataFrame(new_rows)

# 🔄 Combine and remove duplicates
df = pd.concat([df, new_df], ignore_index=True)
df = df.drop_duplicates(subset=[edu_col, skill_col, target_col]).reset_index(drop=True)

print("✅ Final dataset shape after adding 3000 rows:", df.shape)

# 💾 Save back to same uploaded file path
# Change this path if you mounted Google Drive or uploaded via Colab upload
path = '/content/technical_skills_Job_roles_dataset.csv'  # update path to your file
df.to_csv(path, index=False)

print(f"✅ Updated dataset saved to same file: {path}")
```

```
Detected columns → Education: Educational_Qualification, Skills: Technical_Skills, Job Role: Job_Role
100%|██████████| 3000/3000 [00:00<00:00, 238633.62it/s]✅ Final dataset shape after adding 3000 rows: (6782, 3)
✅ Updated dataset saved to same file: /content/technical_skills_Job_roles_dataset.csv
```

After adding rows , dataset looks like

	A	B	C
1	**Technical_Skills**	**Educational_Qualification**	**Job_Role**
2	Photoshop, Illustrator, InDesign	B.Tech in Computer Science	Graphic Designer
3	Risk Management, Auditing	M.Sc in Data Science	Business Analyst
4	HVAC Design, Revit, BIM	B.Pharm in Pharmacy	Mechanical Engineer
5	Photoshop, Illustrator, InDesign	LLB in Law	Graphic Designer
6	Medical Imaging, Radiology	LLB in Law	Lab Technician
7	Legal Research, Contract Law	B.A. in Psychology	Litigation Associate
8	Medical Imaging, Radiology	M.Com in Accounting and Finance	Medical Researcher
9	Risk Management, Auditing	M.Tech in Electrical Engineering	Auditor
10	Patient Care, Diagnostics	B.Tech in Electronics and Communication	Lab Technician
11	Curriculum Design, Pedagogy	B.Des in Graphic Design	Instructional Designer
12	AutoCAD, Civil 3D	MCA in Computer Applications	Production Engineer
13	Business Analysis, SQL, Tableau	BBA in Marketing	Financial Analyst
14	3D Modelling, Blender, Maya	MBA in Finance	Graphic Designer
15	Horticulture, Seed Technology	B.E. in Civil Engineering	Agricultural Officer
16	Legal Research, Contract Law	Diploma in Mechanical Engineering	Litigation Associate
17	Java, Spring Boot, REST API	B.Des in Graphic Design	Cybersecurity Analyst
18	Photoshop, Illustrator, InDesign	M.Tech in Electrical Engineering	Animator
19	Investment Analysis, Portfolio Management	Diploma in Mechanical Engineering	Financial Analyst
20	C++, Data Structures, Algorithms	LLB in Law	Cybersecurity Analyst
21	Risk Management, Auditing	Diploma in Mechanical Engineering	Financial Analyst
22	AutoCAD, Civil 3D	MBA in Finance	Civil Engineer
23	Air Traffic Control, Communication Systems	M.Sc in Data Science	Aircraft Maintenance Engineer
24	Air Traffic Control, Communication Systems	B.E. in Civil Engineering	Aircraft Maintenance Engineer
25	SolidWorks, ANSYS, MATLAB	MCA in Computer Applications	Production Engineer
26	Flight Planning, Navigation Systems	B.Sc in Agriculture	Aircraft Maintenance Engineer
27	Medical Imaging, Radiology	B.Des in Graphic Design	Medical Researcher
28	Simulink, Power Electronics	B.E. in Mechanical Engineering	Electrical Engineer
29	Business Analysis, SQL, Tableau	B.E. in Mechanical Engineering	Financial Analyst

technical_skills_Job_roles_data

- **Multi-Label One-Hot Encoding**

After cleaning, categorical textual columns were transformed into numerical binary vectors (0 and 1).

This was achieved using multi-label one-hot encoding, where each skill or job role was represented as a vector indicating the presence (1) or absence (0) of each possible label.

Example:

Job_Role	Python	Java	SQL	...
Data Analyst	1	0	1	...

Step 1: After cleaning and synthesizing the dataset, we needed to convert the text columns (like *Technical Skills* and *Job Role*) into numerical form so that machine learning models could understand them.

Step 2: One-Hot Encoding was used for this purpose. In this method, each unique skill or job role is represented as a separate column, and the presence of that category in a record is marked with 1, while absence is marked with 0.

Step 3: Since each candidate or record could have multiple skills, a multi-label one-hot encoding approach was applied. This means a single row could have several columns marked as 1 if the person had multiple skills.

Step 4: The encoding was performed using the MultiLabelBinarizer function from the *scikit-learn* library, which automatically converts a list of skills into binary vectors for every row in the dataset.

Step 5: The final output after one-hot encoding was a fully numerical dataset containing 0s and 1s, where each column represented a specific skill or role. This encoded dataset was then ready to be used for model training and further processing.

```python
# Function to split comma-separated values into lists
def split_multilabel(column):
    return column.str.split(',')

# Columns to encode
columns_to_encode = ['Technical_Skills', 'Educational_Qualification', 'Job_Role']

# Initialize MultiLabelBinarizer
mlb = MultiLabelBinarizer()

# List to store encoded dataframes
encoded_dfs = []

for col in columns_to_encode:
    # Split multi-label values
    splitted = split_multilabel(df[col])
    # Encode using MultiLabelBinarizer
    encoded = pd.DataFrame(mlb.fit_transform(splitted),
                           columns=[f"{col}_{cls}" for cls in mlb.classes_])
    encoded_dfs.append(encoded)

# Combine all encoded columns
df_encoded = pd.concat(encoded_dfs, axis=1)

# Save to new CSV file
df_encoded.to_csv("encoded_dataset.csv", index=False)

print("Encoded dataset saved as 'encoded_dataset.csv'")
```

```
Encoded dataset saved as 'encoded_dataset.csv'
```

Encoded dataset looks like:

	A	B	C	D	E	F	G	H	I	J	K
1	Technical_Skills_ adobe xd	Technical_Skills_ after effects	Technical_Skills_ agile	Technical_Skills_ algorithms	Technical_Skills_ ansys	Technical_Skills_ arbitration	Technical_Skills_ arduino	Technical_	Technical_	Technical_	Technical_
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	1	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	1	0	0
22	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	1	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0

multilabel_encoded_dataset

- **Handling Class Imbalance:**

Since some job roles were overrepresented, SMOTE (Synthetic Minority Oversampling Technique) was applied using the imblearn library to balance the dataset.

Key steps:

1. Separate features (X) and target labels (y).
2. Apply SMOTE:

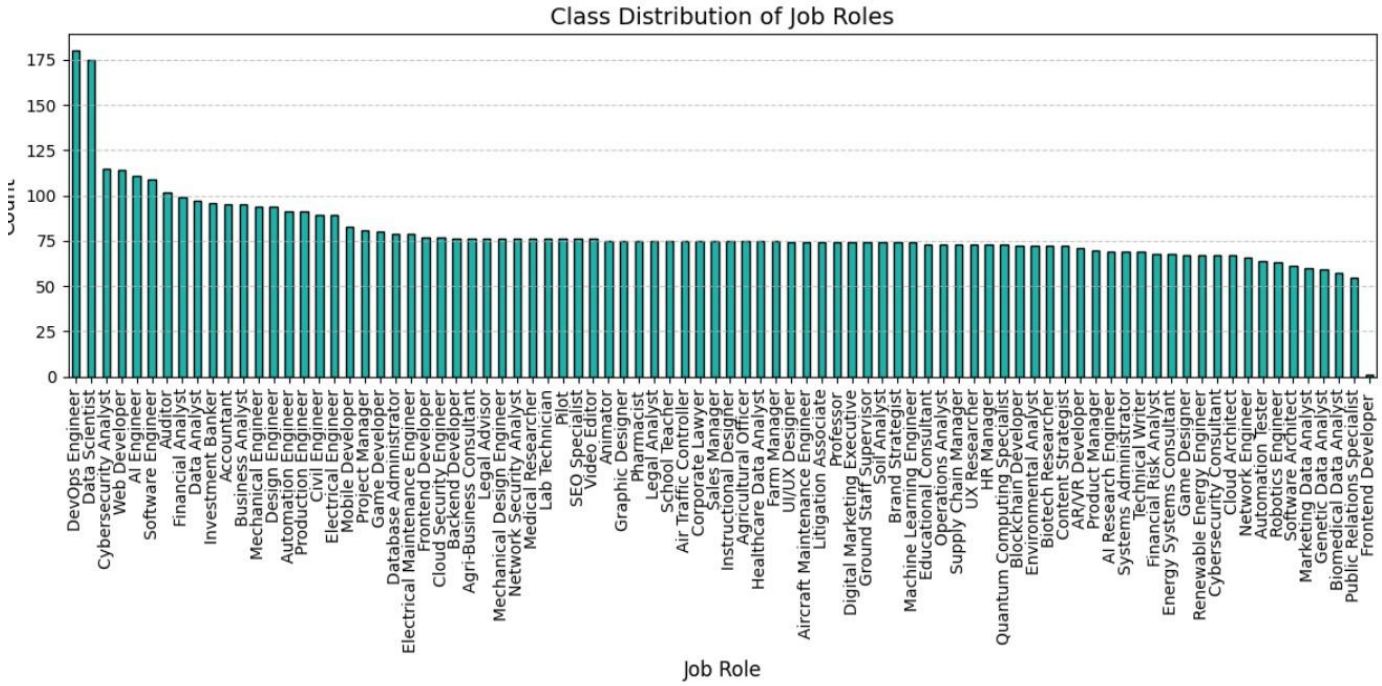
```
# 📊 STEP 3: Analyze Class Distribution
# ==============================
job_role_counts = df['Job_Role'].value_counts()

# Print top 10 job roles
print("\nTop 10 Job Roles:\n", job_role_counts.head(10))

    # Plot bar chart
    plt.figure(figsize=(12,6))
    job_role_counts.plot(kind='bar', color='lightseagreen', edgecolor='black')
    plt.title('Class Distribution of Job Roles', fontsize=14)
    plt.xlabel('Job Role', fontsize=12)
    plt.ylabel('Count', fontsize=12)
    plt.xticks(rotation=90)
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    plt.tight_layout()
    plt.show()
    # Imbalance ratio
    imbalance_ratio = job_role_counts.max() / job_role_counts.min()
    print(f"\nImbalance Ratio: {imbalance_ratio:.2f} (Higher means more imbalance)")
```

```
Top 10 Job Roles:
 Job_Role
DevOps Engineer          180
Data Scientist           175
Cybersecurity Analyst    115
Web Developer            114
AI Engineer              111
Software Engineer        109
Auditor                  102
Financial Analyst         99
Data Analyst              97
Investment Banker         96
Name: count, dtype: int64
```

- **Job_Role Distribution of dataset:**



- **Final Dataset Ready for ML Model**

The resulting encoded dataset (multilabel_encoded_dataset.csv) consisted of:

- **Binary encoded columns (0/1) representing skills and qualifications.**
- **Balanced class labels ensuring fair training distribution.**
- **Clean and structured data suitable for machine learning models.**
- **No duplicates or missing values.**