IS-5960-04: MRP

Employability Analytics Project

Week 6 Deliverable:

Verifying Data Integrity

Group Name: Team 16

Group Members:

Ananya Chowdary Bheemaneni

Maneesha Kakarla

Bala Krishna Kalavakunta

Laya Kalva

Manohar Kancharla

Sai Venkata Sriram Chowdary Karicheti

Revised Problem Statement

The goal of our project is to develop a Power BI dashboard that provides data-driven insights to career advisors regarding Business Analyst job trends across different U.S. states. This dashboard will:

- Provide real-time job market insights, helping advisors guide job seekers.
- Highlight skill gaps and salary benchmarks to enhance employability.
- Support data validation to ensure accurate and reliable insights.

The challenge includes data integration from multiple sources, ensuring data integrity, and validating data fields for correctness and completeness.

Mapping Action Components to Data Fields

Action Component	Dashboard Module	Relevant Data Fields
Job Market Trends	Labor Market Insights	title, location, companyName, sector
Salary Benchmarking	Compensation Analysis	title, companyName, location, experienceLevel, contractType
Candidate Interest	Application Analytics	job title, applicationsCount, contractType, experienceLevel
Career Path Insights	Work Experience Trends	workType, experienceLevel, sector, companyName
Employer Demand	Hiring Patterns	job title, employer, contractType, location, applicationsCount

Data Cleaning and Validation Process

Data Overview

Dataset: Business_analyst_job_listings_linkedin.csv

Total Records (Before Cleaning): 921 rows

Total Columns: 10

Issues Identified:

- Missing values in companyName (10 records)
- Non-numeric values in applications count
- Date format inconsistencies in published at
- Duplicate records (201 records)

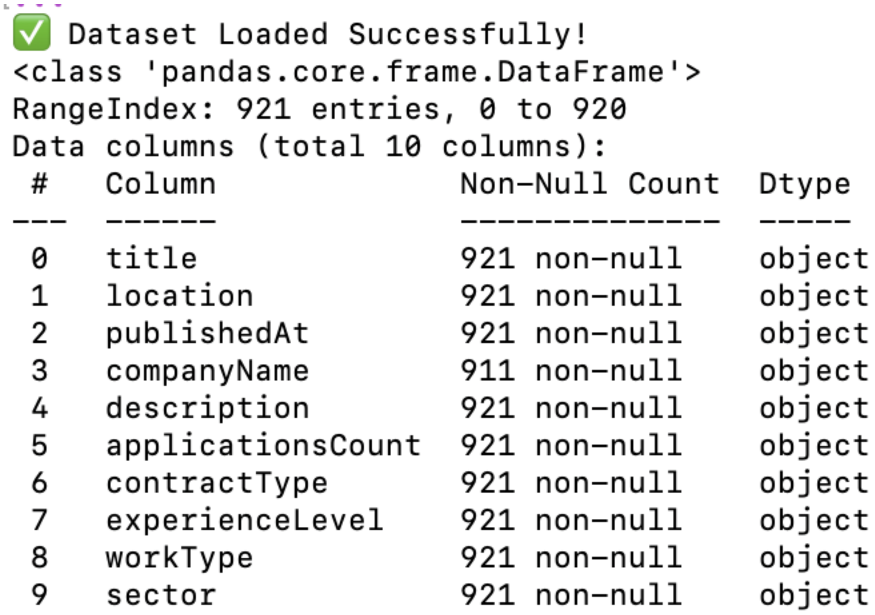
Data Cleaning Steps

Below are the Python scripts which we have executed for data cleaning.

Step 1: Load the Dataset

```
import pandas as pd
file_path =
"/Users/ananya/Documents/TEAM16_MRP/Business_analyst_job_listings_linkedin.csv"
df = pd.read_csv(file_path, dtype=str)
```

Output:



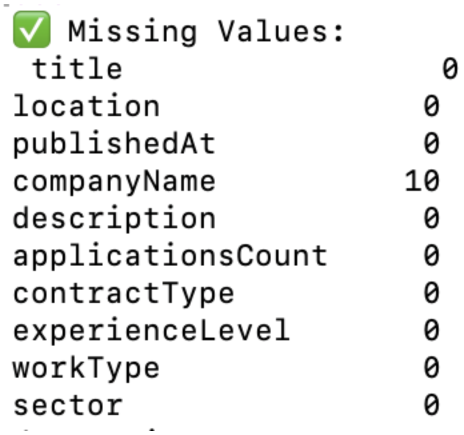
```
✅ Dataset Loaded Successfully!
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 921 entries, 0 to 920
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   title              921 non-null    object
 1   location           921 non-null    object
 2   publishedAt        921 non-null    object
 3   companyName        911 non-null    object
 4   description        921 non-null    object
 5   applicationsCount  921 non-null    object
 6   contractType       921 non-null    object
 7   experienceLevel    921 non-null    object
 8   workType           921 non-null    object
 9   sector             921 non-null    object
```

Step 2: Check for Missing Values

Before Cleaning:

print("✅ Missing Values:\n", df.isnull().sum())

Output:

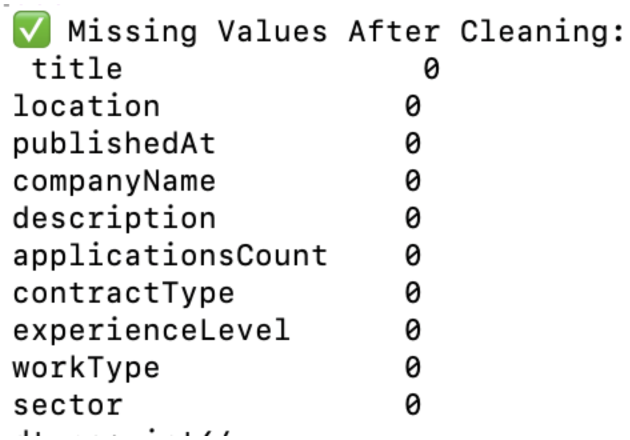


```
✅ Missing Values:
 title                 0
location              0
publishedAt           0
companyName          10
description           0
applicationsCount     0
contractType          0
experienceLevel       0
workType              0
sector                0
```

After Cleaning:

df["companyName"] = df["companyName"].fillna("Unknown") print("✅ Missing Values After Cleaning:\n", df.isnull().sum())

Output:



```
✅ Missing Values After Cleaning:
 title                0
location             0
publishedAt          0
companyName          0
description          0
applicationsCount    0
contractType         0
experienceLevel      0
workType             0
sector               0
```

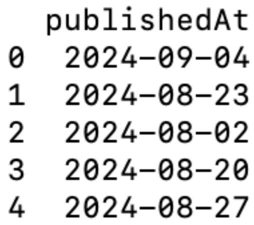
Step 3: Convert applicationsCount to Numeric

```
import re
def extract_number(val):
    match = re.search(r'\d+', str(val))
    return int(match.group()) if match else None
df["applicationsCount"] = df["applicationsCount"].apply(extract_number)
```

Step 4: Convert publishedAt to Standard Date Format

```
df["publishedAt"] = pd.to_datetime(df["publishedAt"], format="%m/%d/%y", errors="coerce")
```

Output:



```
  publishedAt
0  2024-09-04
1  2024-08-23
2  2024-08-02
3  2024-08-20
4  2024-08-27
```

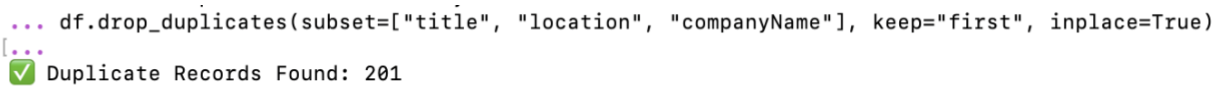
Step 5: Check and Fix Duplicate Job Listings

Before fixing:

```
duplicates = df[df.duplicated(subset=["title", "location", "companyName"], keep=False)]

print("✅ Duplicate Records Found:", len(duplicates))
```

Output:



```
... df.drop_duplicates(subset=["title", "location", "companyName"], keep="first", inplace=True)
[...
✅ Duplicate Records Found: 201
```

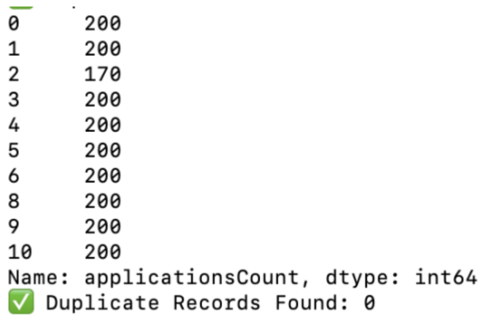
After fixing:

df.drop_duplicates(subset=["title", "location", "companyName"], keep="first", inplace=True)

duplicates = df[df.duplicated(subset=["title", "location", "companyName"], keep=False)]

print("✅ Duplicate Records Found:", len(duplicates))

Output:



```
0     200
1     200
2     170
3     200
4     200
5     200
6     200
8     200
9     200
10    200
Name: applicationsCount, dtype: int64
✅ Duplicate Records Found: 0
```

Step 6: Save the Cleaned Dataset

cleaned_file_path =
"/Users/ananya/Documents/TEAM16_MRP/CLEANED_Business_analyst_job_listings.csv"
df.to_csv(cleaned_file_path, index=False)
print(f"✅ Cleaned dataset saved at: {cleaned_file_path}")

Manual Adjustments Made in Data Cleaning

Alongside automated cleaning using Python, we made a few manual adjustments to ensure data accuracy and integrity.

- Handling missing company name values was necessary because some job listings did not have a company name. Instead of removing these rows, we filled the missing values with "Unknown" to retain useful job listings.

- Fixing applications count values required addressing text-based estimates such as “Over 200 applicants”. While we extracted numbers using code, we manually reviewed edge cases where text conversion failed to ensure all values were accurate.
- Checking date formatting in the published date column was needed because some dates had incorrect formats or invalid values, such as “12/35/24”. We manually checked unique values and removed or corrected any invalid dates before final conversion.
- Standardizing contract type values was required due to inconsistencies in formatting, such as extra spaces or capitalization mismatches, like “FULL TIME” instead of “Full Time”. We manually verified and standardized them for consistency.
- Validating duplicate removals was done by manually checking the dataset before deleting duplicate job listings to ensure that important listings were not mistakenly removed.

AI Usage & External Resources Consulted

AI Prompts Used:

- Write a Python script to clean a dataset with missing values and convert date formats.
- How to validate data integrity in Pandas?

External References:

- *pandas documentation — pandas 2.2.3 documentation*. https://pandas.pydata.org/docs/
- *Newest “pandas” questions*. Stack Overflow. https://stackoverflow.com/questions/tagged/pandas

Final Dataset Summary

Step	**Action Taken**
Load Data	Read CSV file
Fix Multi-line Descriptions	Replaced \n with spaces
Handle Missing Values	Filled missing companyName values
Convert applicationsCount	Extracted numeric values

Convert publishedAt	Standardized to YYYY-MM-DD
Check Missing Values	Confirmed no null values remain
Check Date Range	Verified dates are valid
Check and Remove Duplicates	Removed 201 duplicate job listings
Save Cleaned Dataset	Exported cleaned data to CSV