Week 6 Verifying data integrity

Group: Team 9

Aravind Kamuni

Aravind Reddy Manda

Lokender Yadav Kanneboina

Prudhviraj Kancharla

Rajesh Kandukuri

Satish Kumar Kalla

Master's research project

Professor: Maria Weber

Date: 02/21/2025

Revised Problem Statement

Students and recent graduates receive inaccurate and untimely career guidance from educational institution career advisors who handle outdated and fragmented job market data. A centralized career analytics dashboard will integrate real-time job market data with salary metrics alongside skill requirement analysis and industry reports and career development pathways to close the guidance gap. Students will obtain better employment preparation thanks to data-driven guidance from advisors who use this dashboard.

Mapping "Action Components" to Data Fields

Action Component	Data Fields in Dataset
Identify job market trends	`Industry Sector`, `Job Demand Trends`, `Job Posting Sources`
Recommend in-demand skills	`Top Skills Required`
Guide students on certifications	`Certifications Preferred`
Provide salary expectations	`salary_in_usd`
Advise on career progression	`Career Progression Path`
Highlight remote work opportunities	`remote_ratio`
Assess job satisfaction and work balance	`Job Satisfaction Rating`, `Work-Life Balance Score`

Data Validation and Integrity Checks

To ensure the integrity of the dataset, we performed the following checks using Python:

1. **Referential Integrity Checks**:
    - Ensured that key fields like `job_title`, `work_year`, `salary_in_usd` contain valid values.
2. **Field-Level Integrity Checks**:
    - Checked for missing values.
    - Verified that `salary_in_usd` does not contain negative or zero values.
    - Ensured `remote_ratio` is within the valid range (0-100).
    - Validated categorical fields like `experience_level` and `employment_type` to ensure only expected values are present.
    - Identified duplicate job titles per work year to check uniqueness constraints.

Python Code Used for Integrity Checks and Cleaning:

```
import pandas as pd

# Load the dataset
file_path = "Enhanced_Cybersecurity_Job_Dataset.csv"
```

```
df = pd.read_csv(file_path)

# Display column names to confirm available fields
print("Dataset Columns:")
print(df.columns)

# Performing Comprehensive Data Integrity Checks

# Check for missing values in any column
missing_values = df.isnull().sum()

# Check for duplicate records
duplicate_rows = df.duplicated().sum()

# Check salary_in_usd for negative or zero values (invalid salaries)
invalid_salaries = df[df["salary_in_usd"] <= 0]

# Check remote_ratio for valid values (should be between 0 and 100)
invalid_remote_ratio = df[(df["remote_ratio"] < 0) |
(df["remote_ratio"] > 100)]

# Check experience levels for valid values
valid_experience_levels = ["EN", "MI", "SE", "EX"]  # Entry, Mid,
Senior, Executive
invalid_experience_levels =
df[~df["experience_level"].isin(valid_experience_levels)]

# Check employment types for valid values
valid_employment_types = ["FT", "PT", "CT", "FL"]  # Full-time, Part-
time, Contract, Freelance
invalid_employment_types =
df[~df["employment_type"].isin(valid_employment_types)]

# Check for valid salary currency codes (ISO 3-letter format)
invalid_salary_currency =
df[~df["salary_currency"].astype(str).str.match(r"^[A-Z]{3}$",
na=False)]

# Check for valid company location country codes (2-letter format)
invalid_company_location =
df[~df["company_location"].astype(str).str.match(r"^[A-Z]{2}$",
na=False)]

# Check for valid employee residence country codes (2-letter format)
invalid_employee_residence =
df[~df["employee_residence"].astype(str).str.match(r"^[A-Z]{2}$",
na=False)]

# Check that industry sectors belong to a predefined valid list
valid_industry_sectors = ["Finance", "Healthcare", "Government",
"Retail", "Technology", "Consulting", "Education"]
invalid_industry_sectors = df[~df["Industry
Sector"].isin(valid_industry_sectors)]

# Check that job demand trends contain only expected values
valid_demand_trends = ["Increasing", "Decreasing", "Stable"]
```

```
invalid_demand_trends = df[~df["Job Demand
Trends"].isin(valid_demand_trends)]

# Ensure work-life balance and job satisfaction scores are within range
(1-5)
invalid_work_life_balance = df[(df["Work-Life Balance Score"] < 1) |
(df["Work-Life Balance Score"] > 5)]
invalid_job_satisfaction = df[(df["Job Satisfaction Rating"] < 1) |
(df["Job Satisfaction Rating"] > 5)]

# Print final cleaning summary
print("Final Data Cleaning Summary:")
print(f"Total Missing Values: {missing_values.sum()}")
print(f"Total Duplicate Rows: {duplicate_rows}")
print(f"Invalid Salaries: {len(invalid_salaries)}")
print(f"Invalid Remote Ratio: {len(invalid_remote_ratio)}")
print(f"Invalid Experience Levels: {len(invalid_experience_levels)}")
print(f"Invalid Employment Types: {len(invalid_employment_types)}")
```

Output:

```
Dataset Columns:
Index(['work_year', 'experience_level', 'employment_type', 'job_title',
       'salary', 'salary_currency', 'salary_in_usd', 'employee_residence',
       'remote_ratio', 'company_location', 'company_size', 'Industry Sector',
       'Top Skills Required', 'Certifications Preferred',
       'Educational Requirements', 'Job Demand Trends',
       'Career Progression Path', 'Job Posting Sources',
       'Work-Life Balance Score', 'Job Satisfaction Rating'],
      dtype='object')

Comprehensive Data Integrity Validation Results:
Missing Values (per column): {'work_year': 0, 'experience_level': 0, 'employment_type': 0, 'job_title': 0, 'salary': 0, 'salar
y_currency': 0, 'salary_in_usd': 0, 'employee_residence': 0, 'remote_ratio': 0, 'company_location': 0, 'company_size': 0, 'Ind
ustry Sector': 0, 'Top Skills Required': 0, 'Certifications Preferred': 0, 'Educational Requirements': 0, 'Job Demand Trends':
 0, 'Career Progression Path': 0, 'Job Posting Sources': 0, 'Work-Life Balance Score': 0, 'Job Satisfaction Rating': 0}
Duplicate Rows: 0
Invalid Salaries (<= 0): 0
Invalid Remote Ratio Values: 0
Invalid Experience Levels: 0
Invalid Employment Types: 0
Invalid Salary Currency Codes: 0
Invalid Company Location Codes: 0
Invalid Employee Residence Codes: 0
Invalid Industry Sectors: 0
Invalid Job Demand Trends: 0
Invalid Work-Life Balance Scores: 0
Invalid Job Satisfaction Ratings: 0

Final Data Cleaning Summary:
Total Missing Values Before Cleaning: 0
Total Missing Values After Cleaning: 0
Total Duplicate Rows Removed: 0
Final Duplicate Rows After Cleaning: 0
```

Results of the Data Validation Checks:

- No missing values were found in the dataset.
- No duplicate records were found.
- No invalid salaries (zero or negative values) were found.
- All `remote_ratio` values were within the valid range (0-100).
- All categorical fields contained valid values.
- No duplicate job titles within the same year were detected.

AI and External Tools Used:

We utilized generative AI to assist in structuring the validation script. The following prompt was used:

Prompt Used: "Write a comprehensive Python script to check data integrity for a cybersecurity job dataset. Ensure missing values, duplicates, salary validation, and categorical consistency are verified. Also, include data cleaning steps."

The generated code was reviewed and refined to meet the project requirements accurately.

Conclusion:

The system verifies that the dataset meets required quality standards to be used properly within the career analytics dashboard. Structured validation enables career advisors to supply students with data-based insights which guide their career selection decisions. Further analytics and visualization can proceed with the dataset after it passes quality standards.

References

- Kaggle Dataset: https://www.kaggle.com/datasets/dannyrevaldo/salary-cyber-security-jobs
- Pandas Documentation: https://pandas.pydata.org/docs/
- Python Regular Expressions: https://docs.python.org/3/library/re.html
- ISO Currency Codes: https://www.iso.org/iso-4217-currency-codes.html
- Country Codes Reference: https://www.iso.org/iso-3166-country-codes.html
- Industry Trends and Data Validation Methods from Research Articles and Online Data Science Communities