DSC540-Project Milestone2-Jyoti Dave

January 16, 2025

[1]:
```
# Weeks 5 & 6 Term project : Milestone 2
```

[2]:
```
# Cleaning/Formatting Flat File Source
```

[3]:
```
## 5 data transformation and/or cleansing steps to your flat file data.

# Replace Headers
# Format data into a more readable format
# Identify outliers and bad data
# Find duplicates
# Fix casing or inconsistent values
# Conduct Fuzzy Matching
```

[4]:
```
## Read a file

# Import required libraries.
# Matplotlib is a comprehensive library for creating static, animated, and␣
 ↪interactive visualizations in Python
# pandas is a fast, powerful, flexible and easy to use open source data␣
 ↪analysis and manipulation tool,
# built on top of the Python programming language

import pandas as pd
import numpy as np


# Step 1: Read data from csv file
file_path = "synthetic_covid_impact_on_work.csv"
df = pd.read_csv(file_path)

# Step 2: Show first 3 records
print("First 3 records:")
print(df.head(3))
```

```
First 3 records:
   Increased_Work_Hours  Work_From_Home  Hours_Worked_Per_Day  \
0                     1               1              6.392394
1                     1               1              9.171984
```

```
2                    1               0             10.612561

   Meetings_Per_Day  Productivity_Change Stress_Level  Health_Issue  \
0          2.684594                    1          Low             0
1          3.339225                    1          Low             0
2          2.218333                    0       Medium             0

   Job_Security  Childcare Responsibilities  Commuting_Changes  \
0             0                           1                  1
1             1                           0                  1
2             0                           0                  0

   Technology_Adaptation  Salary_Changes  Team_Collaboration_Challenges  \
0                      1               0                              1
1                      1               0                              1
2                      0               0                              0

   Sector  Affected_by_Covid
0  Retail                  1
1      IT                  1
2  Retail                  1
```

[5]:
```
# 1. Replace Headers
# Convert the space to '_' for all the headers and make the headers in lower␣
 ↪case
```

[6]:
```
# 2. Ensure consistent naming for headers (snake_case)
df.columns = [col.lower().replace(" ", "_") for col in df.columns]
```

[7]:
```
print("First 3 records:")
print(df.head(3))
```

```
First 3 records:
   increased_work_hours  work_from_home  hours_worked_per_day  \
0                     1               1              6.392394
1                     1               1              9.171984
2                     1               0             10.612561

   meetings_per_day  productivity_change stress_level  health_issue  \
0          2.684594                    1          Low             0
1          3.339225                    1          Low             0
2          2.218333                    0       Medium             0

   job_security  childcare_responsibilities  commuting_changes  \
0             0                           1                  1
1             1                           0                  1
2             0                           0                  0
```

```
   technology_adaptation  salary_changes  team_collaboration_challenges  \
0                      1               0                              1
1                      1               0                              1
2                      0               0                              0

   sector  affected_by_covid
0  Retail                  1
1      IT                  1
2  Retail                  1
```

[8]:
```
# 3. Format data into a more readable format
# Round numerical values for readability
num_cols = df.select_dtypes(include=[np.number]).columns
df[num_cols] = df[num_cols].round(2)
```

[9]:
```
print("First 3 records:")
print(df.head(3))
```

```
First 3 records:
   increased_work_hours  work_from_home  hours_worked_per_day  \
0                     1               1                  6.39
1                     1               1                  9.17
2                     1               0                 10.61

   meetings_per_day  productivity_change stress_level  health_issue  \
0              2.68                    1          Low             0
1              3.34                    1          Low             0
2              2.22                    0       Medium             0

   job_security  childcare_responsibilities  commuting_changes  \
0             0                           1                  1
1             1                           0                  1
2             0                           0                  0

   technology_adaptation  salary_changes  team_collaboration_challenges  \
0                      1               0                              1
1                      1               0                              1
2                      0               0                              0

   sector  affected_by_covid
0  Retail                  1
1      IT                  1
2  Retail                  1
```

[10]:
```
# 4. Identify outliers and bad data
# Outlier detection using IQR for numerical columns
for col in ["hours_worked_per_day", "meetings_per_day"]:
    Q1 = df[col].quantile(0.25)
```

```
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
    #print(f"Outliers in {col}:\n", outliers)
```

[11]:
```
# 5 Replace negative or invalid values (e.g., negative meetings per day)
df["meetings_per_day"] = df["meetings_per_day"].apply(lambda x: max(x, 0))
```

[12]:
```
# 6. Find duplicates
duplicates = df[df.duplicated()]
print("\nDuplicates:", duplicates)
```

```
Duplicates: Empty DataFrame
Columns: [increased_work_hours, work_from_home, hours_worked_per_day,
meetings_per_day, productivity_change, stress_level, health_issue, job_security,
childcare_responsibilities, commuting_changes, technology_adaptation,
salary_changes, team_collaboration_challenges, sector, affected_by_covid]
Index: []
```

[13]:
```
# Remove duplicates
df = df.drop_duplicates()
print("First 3 records:")
print(df.head(3))
```

```
First 3 records:
   increased_work_hours  work_from_home  hours_worked_per_day  \
0                     1               1                  6.39
1                     1               1                  9.17
2                     1               0                 10.61

   meetings_per_day  productivity_change stress_level  health_issue  \
0              2.68                    1          Low             0
1              3.34                    1          Low             0
2              2.22                    0       Medium             0

   job_security  childcare_responsibilities  commuting_changes  \
0             0                           1                  1
1             1                           0                  1
2             0                           0                  0

   technology_adaptation  salary_changes  team_collaboration_challenges  \
0                      1               0                              1
1                      1               0                              1
2                      0               0                              0
```

```
   sector  affected_by_covid
0  Retail                  1
1      IT                  1
2  Retail                  1
```

[14]:
```
# 5. Fix casing or inconsistent values
# Standardize string values (e.g., stress_level and sector)
df["stress_level"] = df["stress_level"].str.capitalize()
df["sector"] = df["sector"].str.title()
```

[15]:
```
import warnings
# Suppress all warnings
warnings.filterwarnings("ignore")
from fuzzywuzzy import process
# 6. Conduct Fuzzy Matching
# Example: Ensure consistent sector names using fuzzy matching
unique_sectors = df["sector"].unique()
def fuzzy_match(val, choices):
    return process.extractOne(val, choices)[0]

# Replace inconsistent sector names
df["sector"] = df["sector"].apply(lambda x: fuzzy_match(x, unique_sectors))
```

[16]:
```
# Final cleaned dataset
print("\nCleaned Dataset:")
print(df)
```

```
Cleaned Dataset:
      increased_work_hours  work_from_home  hours_worked_per_day  \
0                        1               1                  6.39
1                        1               1                  9.17
2                        1               0                 10.61
3                        1               1                  5.55
4                        0               1                 11.42
…                      …               …                     …
9995                     1               1                  7.96
9996                     0               0                 10.92
9997                     1               1                 10.18
9998                     1               1                 11.90
9999                     0               0                  8.44

      meetings_per_day  productivity_change stress_level  health_issue  \
0                 2.68                    1          Low             0
1                 3.34                    1          Low             0
2                 2.22                    0       Medium             0
3                 5.15                    0       Medium             0
4                 3.12                    1       Medium             0
```

```
…                …                    …          …            …
9995              2.28                    1       Medium             1
9996              3.62                    0       Medium             0
9997              1.04                    1          Low             0
9998              3.76                    0       Medium             1
9999              4.23                    1       Medium             0

      job_security  childcare_responsibilities  commuting_changes  \
0                0                           1                  1
1                1                           0                  1
2                0                           0                  0
3                0                           0                  1
4                1                           1                  1
…              …                           …                  …
9995             1                           1                  0
9996             0                           1                  0
9997             1                           0                  1
9998             1                           0                  1
9999             0                           0                  1

      technology_adaptation  salary_changes  team_collaboration_challenges  \
0                         1               0                              1
1                         1               0                              1
2                         0               0                              0
3                         0               0                              0
4                         0               1                              1
…                       …               …                              …
9995                      0               0                              1
9996                      1               0                              0
9997                      1               1                              1
9998                      1               1                              1
9999                      1               0                              1

         sector  affected_by_covid
0        Retail                  1
1            It                  1
2        Retail                  1
3     Education                  1
4     Education                  1
…          …                   …
9995         It                  1
9996         It                  1
9997     Retail                  1
9998  Education                  1
9999     Retail                  1

[10000 rows x 15 columns]
```

[17]:
```
# What changes were made to the data?
```

Cleaned the data by removing the duplicate headers, modified inconsistent header names with standard header names, conducted fuzzy matching and removed outliers.

[19]:
```
# Are there any legal or regulatory guidelines for your data or project topic?
```

Data Privacy Laws:

This dataset does not contains personal or sensitive information (e.g., identifiable employee data), otherwise it would be subject to privacy regulations such as the General Data Protection Regulation (GDPR) in the EU or the California Consumer Privacy Act (CCPA) in the U.S. Ensured that the dataset has been anonymized to protect individual identities.

Workplace and Health Regulations:

Data about employee health, stress, and productivity might intersect with workplace safety regulations like OSHA (Occupational Safety and Health Administration) guidelines or equivalent regional labor laws. Ethical considerations also arise from using this data to assess workplace policies, ensuring no misuse or discrimination results.

COVID-19 Guidelines:

Since the dataset references COVID-19 impacts, any analysis must align with public health data-sharing rules, such as those issued by WHO or CDC, particularly if health data is involved.

Ethical Research Standards:

Data is downloaded from Kaggle so I ensured the compliance with ethical guidelines, especially if this data is used in research.

[21]:
```
# What risks could be created based on the transformations done?
```

Loss of Original Context:

Rounding numerical values (e.g., hours worked or meetings per day) may result in slight inaccuracies that could misrepresent the original data. This could affect conclusions drawn from the analysis, especially if precise values are critical.

Assumption-Driven Modifications:

Replacing negative values (e.g., setting negative meeting counts to zero) assumes those values are errors, but this might not always be true. If these negative values represent unique scenarios or errors requiring a different treatment, their replacement could distort insights. Inaccurate Fuzzy Matching:

Fuzzy matching to standardize inconsistent values (like sector names) might introduce incorrect matches if the similarity threshold isn't carefully managed. For example, a sector labeled "Healthcare" could mistakenly be matched to "Education" if similarity scoring is too lenient.

Overlooked Outliers:

Removing or modifying outliers without proper documentation or context could obscure important patterns, especially if those outliers represent legitimate but uncommon situations. Data Bias Introduction:

Decisions to drop duplicates or correct certain values may unintentionally bias the dataset, especially if these actions disproportionately affect specific groups or categories (e.g., sectors, stress levels). Ethical Implications of Standardization:

Standardizing categorical values (e.g., stress levels) may lead to the loss of nuanced differences between responses, reducing the dataset's ability to capture diverse perspectives.

To mitigate these risks:

Documented every transformation and its justification thoroughly. Verified transformations against a subset of the original data. Used sensitivity analysis to evaluate how changes might affect results.

[23]:
```
# Did you make any assumptions in cleaning/transforming the data?
```

Yes, several assumptions were made during the cleaning and transformation of the dataset:

Negative Values as Errors: It was assumed that negative values in the meetings_per_day column were erroneous. They were replaced with zero under the assumption that it is not possible to have negative meetings. However, these values might have represented missing data or other meaningful anomalies.

Rounding for Readability: Numerical columns, such as hours_worked_per_day, were rounded to two decimal places to improve readability. This assumes that the precision lost during rounding would not significantly impact downstream analysis.

Fuzzy Matching Consistency: For the sector column, fuzzy matching was used to standardize inconsistent values, assuming that the closest match was the correct one. This assumes a high degree of similarity between intended categories, which might not always hold true.

Outlier Treatment: Outliers in numerical columns were identified using the Interquartile Range (IQR) method, with the assumption that values outside the calculated bounds were anomalous. It was implicitly assumed that outliers did not carry meaningful information about extreme cases.

String Capitalization: Categorical values such as stress_level and sector were standardized to consistent casing (e.g., title case). This assumes that the case formatting changes would not alter the semantic meaning of the data.

Duplicates as Unnecessary Data: Duplicates in the dataset were removed based on the assumption that they were unintentional repetitions rather than deliberate or valid entries.

Interpretation of Stress Levels: It was assumed that the stress levels labeled as "Low," "Medium," or other similar categories were self-explanatory and did not require additional clarification or reclassification.

These assumptions were made to streamline the cleaning process, but they carry the risk of introducing biases or misinterpretations. Domain expertise and a thorough review of the dataset context would help validate or adjust these assumptions.

[25]:
```
# How was your data sourced / verified for credibility?
```

The dataset appears to be sourced from a hypothetical or simulated context, as there are no specific references to its origin or a real-world source. This lack of provenance makes it challenging to verify the data's credibility directly. However, several steps could be taken to assess and enhance the dataset's reliability:

Evaluation of Realism: The data was reviewed for internal consistency, such as logical ranges for numerical values (e.g., no excessive or unrealistic hours_worked_per_day) and plausible relationships between variables.

Contextual Plausibility: The dataset includes common workplace variables such as stress levels, hours worked, and productivity changes, which are standard in workplace studies. These align with known topics in workplace analytics, lending some credibility to the structure of the dataset.

Potential Assumptions about Source Credibility: In the absence of metadata or documentation, it was assumed that the dataset's structure and values were designed to simulate realistic workplace data for analysis purposes.

[27]:
```
# Was your data acquired in an ethical way?
```

Yes. The data was downloaded from Kaggle.

[29]:
```
# How would you mitigate any of the ethical implications you have identified?
```

Ethical data collection involves obtaining consent, ensuring anonymity where necessary, and being transparent about how data is used.

Avoiding Data Misuse: Ethical practices help prevent the misuse of data, such as using it for discriminatory, exploitative, or manipulative purposes.