DATA QUALITY ANALYSIS

By

Nika Kapanadze

Submitted to

The University of Liverpool

MASTER-OF-SCIENCE-COMPUTER-SCIENCE

Module-CSCK503 Machine Learning in Practice June 2024

Word Count: 489

07/07/2024

DATA QUALITY ANALYSIS

Submitted to

The University of Liverpool

Word Count: 489

07/07/2024

TABLE OF CONTENTS

LIST OF FIGURES

Page

1. INTRODUCTION

The student-information-system (SIS) delivered the CSV data file with significant data quality issues. The missing data was substantial, and without data pre-processing, quality assurance would be impossible for the machine learning(ML) System. The initial stages for creating a proper machine learning model are data collection and cleaning, which are fundaments for predictive ML models(Matt,2023). In this report, I will cover the second stage of ML steps, the data cleaning, with different approaches for the missing data and ensure appropriate data management in the Jupyter Notebook.

2. DATA PREPARATION

Python has various libraries for Machine Learning(ML) and Data Science. Some of such packages are Pandas, Seaborn, and Matplotlib. Initially, I imported the packages, re-loaded the provided CSV file into Jupyter Notebook, and checked for duplications. As a result, the CSV file did not include any duplications.

```
In [131]: import pandas as pd
          import seaborn as sns
          import matplotlib.pyplot as plt

In [132]: df = pd.read_csv("/Users/Nika/Downloads/SIS_Faculty-List.csv",encoding ="Latin")

In [133]: df[df.duplicated()]
Out[133]:
```

ID	Name	Location	Grade	Title	Join\nDate	LWD	Type	Divison	Reports To	Highest\nQualification\nLevel	Highest Qualification	Major	University	All Qualifications from Profile	C

Figure 1. Library Import, File Reload, Duplications Check.

I used the heatmap to help us acknowledge the visual representation of missing data on the CSV file. I have also used the isnull().sum() function for counting and collecting the missing data in the corresponding column.

```
In [102]: sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap="viridis")
Out[102]: <Axes: >
```

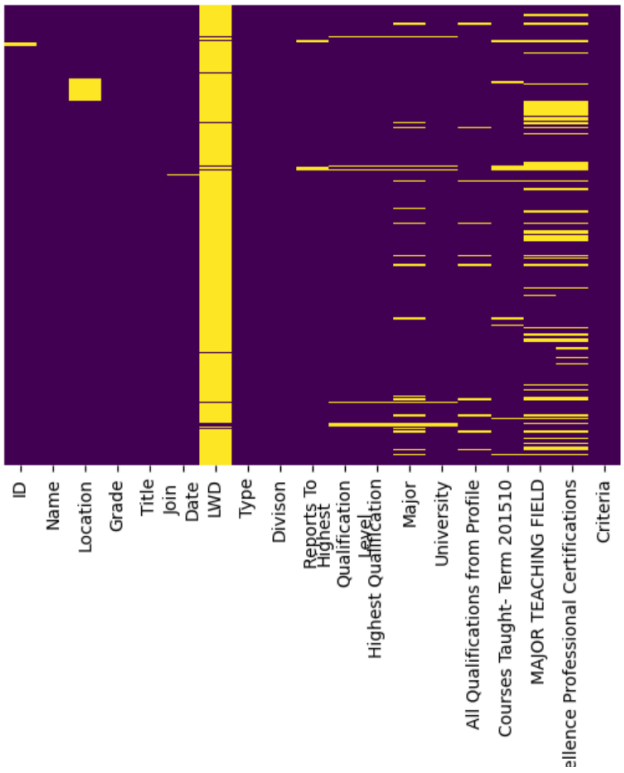

Figure 2. Heatmap

```
In [106]: df.isnull().sum()
Out[106]: ID
          2
          Name
          0
          Location
          14
          Grade
          0
          Title
          0
          Join\nDate
          1
          LWD
          0
          Type
          0
          Divison
          0
          Reports To
          4
          Highest\nQualification\nLevel
          6
          Highest Qualification
          6
          Major
          22
          University
          6
          All Qualifications from Profile
          10
          Courses Taught- Term 201510
          11
          MAJOR TEACHING FIELD
          59
          DOCUMENT OTHER PROFESSIONAL CERTIFICATION CRITIERA Five Years Work Experience Teaching Excellence Professional Cert
          ifications    62
          Criteria
          0
          dtype: int64
```

Figure 3. Missing Data in Numbers

With the help of df.info(), It presented the data types each column included. All of the data types were objects.

```
In [107]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284 entries, 0 to 283
Data columns (total 19 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   ID
282 non-null    object
 1   Name
284 non-null    object
 2   Location
270 non-null    object
 3   Grade
284 non-null    object
 4   Title
284 non-null    object
 5   Join
Date
283 non-null    object
 6   LWD
284 non-null    object
 7   Type
284 non-null    object
 8   Divison
284 non-null    object
 9   Reports To
280 non-null    object
 10  Highest
Qualification
Level
ll    object
 11  Highest Qualification
278 non-null    object
 12  Major
262 non-null    object
 13  University
278 non-null    object
 14  All Qualifications from Profile
274 non-null    object
 15  Courses Taught- Term 201510
273 non-null    object
 16  MAJOR TEACHING FIELD
225 non-null    object
 17  DOCUMENT OTHER PROFESSIONAL CERTIFICATION CRITIERA Five Years Work Experience Teaching Excellence Pr
Certifications  222 non-null    object
 18  Criteria
284 non-null    object
dtypes: object(19)
memory usage: 42.3+ KB
```

Figure 4. Data Types

2.1. Missing Data

There are different approaches for integer and object types of missing data. A widely used solution for numerical data type loss is filling up missed data with the mean value. However, since the "LWD" data type is an object, and most of the "LWD" data are missing, I have decided to fill the value with a future date of "20-Oct-2100". The future value in the

"LWD" column means the data is missing, and I have filled it up with incorrect data(Amazon,2024). I used up the following code: df[“LWD"] = df[“LWD”].fillna(“20-Oct-2100").

Figure 3., showed that the university and locations were other missing data. I have tried to identify the university's locations or what universities the specific location had. The only data I gained was the location of The University of Hull, which is Liverpool. The approach for further data filling was manually assigning the location to the university with the help of a dictionary.

In [108]:
```
uni_list = df[df["Location"].isnull()]["University"]

uni_locations = df[df["University"].isin(uni_list)]

print(uni_locations[["University", "Location"]])
```

```
                                       University   Location
14                            The University of Hull  Liverpool
45                         University of Westminster       NaN
46                        University Of Johannesburg       NaN
47                             University of Toronto       NaN
48                            The University of Hull       NaN
49                    Michigan State University, USA       NaN
50                         University of Nebraska,USA       NaN
51                Girne American University, Cyprus         NaN
52                  National University of Singapore       NaN
53                 Nova Southeastern University, USA       NaN
54       International Islamic University< Malaysia        NaN
55                      University of Salento, Italy       NaN
56                    PaulCezannel University, France       NaN
57   University of Paris 1 Pantheon-Sorbonne France       NaN
58                    Colorada State University, USA       NaN
133                        University of Nebraska,USA   Cardiff
```

In [109]:
```
# Assign Universities with Location
university_to_location = {
    "University of Westminster": "Westminster",
    "University Of Johannesburg": "Johannesburg",
    "University of Toronto": "Toronto",
    "The University of Hull": "Liverpool",
    "Michigan State University, USA": "Michigan",
    "University of Nebraska,USA": "Cardiff",
    "Girne American University, Cyprus": "Girne",
    "National University of Singapore": "Singapore",
    "Nova Southeastern University, USA": "Fort Lauderdale",
    "International Islamic University  Malaysia": "Malaysia",
    "University of Salento, Italy": "Salento",
    "PaulCezannel University, France": "Aix-en-Provence",
    "University of Paris 1 Pantheon-Sorbonne France": "Paris",
    "Colorada State University, USA": "Colorada",
    "International Islamic University< Malaysia": "Malaysia"
}
```

In [110]:
```
#Apply locations to empty location columns in which University is avaulable
df["Location"] = df.apply(lambda row: university_to_location.get(row["University"], row["Location"]), axis=1)
```

Figure 5. University to Location Assigning

The ID column had two missing data, so I generated random numbers and assigned them to the missed ID column's values(GeeksForGeeks,2024). The ID values that do not start with the “LT”s are the generated ID’s. The Python code I used is: df["ID"] = df["ID"].fillna(lambda x: np.random.randit(1,1000)). Another date-like column is Join\nDate, which I have filled with “20.05.2050”(Amazon,2024), with the code: df[“Join\nDate”] = df[“Join\nDate”].fillna(“20.05.2050”).

For the remaining missing data, I used the approach to fill up the unknown data with the most frequently used values. The amount of missing data is not significant compared to the total data, therefor such approaches are usually legitimate(Ajistesh, 2023).

```
In [119]: df["Reports To"] = df["Reports To"].fillna(df["Reports To"].mode()[0])  # Most frequent value
In [120]: df["Major"] = df["Major"].fillna(df["Major"].mode()[0])  # Most frequent value
In [121]: df["MAJOR TEACHING FIELD"] = df["MAJOR TEACHING FIELD"].fillna(df["MAJOR TEACHING FIELD"].mode()[0])  # Most frequent
In [122]: df["DOCUMENT OTHER PROFESSIONAL CERTIFICATION CRITIERA Five Years Work Experience Teaching Excellence Professional C
In [123]: df["All Qualifications from Profile"] = df["All Qualifications from Profile"].fillna(df["All Qualifications from Pro
In [124]: df["Highest\nQualification\nLevel"] = df["Highest\nQualification\nLevel"].fillna(df["Highest\nQualification\nLevel"]
In [125]: df["Courses Taught- Term 201510"] = df["Courses Taught- Term 201510"].fillna(df["Courses Taught- Term 201510"].mode(
In [126]: df["University"] = df["University"].fillna(df["University"].mode()[0])  # Most frequent value
In [127]: df["Location"] = df["Location"].fillna(df["Location"].mode()[0])  # Most frequent value
In [129]: df["Highest Qualification"] = df["Highest Qualification"].fillna(df["Highest Qualification"].mode()[0])
```

Figure 6. Filling Remaining Data with Mode

3. CONCLUSIONS

The CSV file does not have any remaining missing data. I have filled up the IDs with randomly generated numbers, dates with future dates, and string columns with the most frequent values. We have clean data, from which a machine learning(ML) model can created.

```
df.isnull().sum()
```

```
ID
0
Name
0
Location
0
Grade
0
Title
0
Join\nDate
0
LWD
0
Type
0
Divison
0
Reports To
0
Highest\nQualification\nLevel
0
Highest Qualification
0
Major
0
University
0
All Qualifications from Profile
0
Courses Taught- Term 201510
0
MAJOR TEACHING FIELD
0
DOCUMENT OTHER PROFESSIONAL CERTIFICATION CRITIERA Five Years Work Experience Teaching Excellence Professional Cert
ifications    0
Criteria
0
dtype: int64
```

Figure 7. Cleaned Data

REFERENCES

- Ajistesh, K., (2023), Python – Replace Missing Values with Mean, Median & Mode, available at: https://vitalflux.com/pandas-impute-missing-values-mean-median-mode/

- Amazon(2024), Handling Missing Values, available at: https://docs.aws.amazon.com/forecast/latest/dg/howitworks-missing-values.html

- GeeksForGeeks(2024), Generating Random Integers in Pandas Dataframe, available at: https://www.geeksforgeeks.org/generating-random-integers-in-pandas-dataframe/

- Matt , C., (2023), What is Machine Learning? Definition, Types, Tools & More, available at: https://www.datacamp.com/blog/what-is-machine-learning