Awarding Body	**Arden University, Berlin**
Programme Name	M. Sc. in Data Science
Module Name	Programming for Data Science
Assessment Title	Exploratory Data Analysis
Student Number	24152182
Tutor Name	Ahmed Hassan
Word Count	~ 2,850

Table of Contents

1. Introduction

This report examines and analyses a dataset, and conclusively stating with evidence what factors contributed towards the subjects of the study changing their occupation. The dataset (Career data_PDA_4053.xls) contains multiple features about each individual, including demographic factors like gender, age, geographic mobility, as well as details about their professional background including experience, job satisfaction, salary, industry growth rate, and more. The target variable of this study is the feature called ‘Likely to Change Occupation’, which is also predetermined in the dataset.

This study investigates the correlation between this target feature in context of the other columns is the primary objective of this study.

This type of analysis is related to workforce churn studies, where the focus is on factors leading to transitions between occupations. It would help the organisation make decisions driven by substantial evidence, for instance identify patterns such as the relationship between salary brackets and job satisfaction and hence, less likelihood of subjects changing jobs, or the degree of influence demographic features have on job churn.

The approach for this particular study starts with forming a schema of the raw dataset in order to understand more about each feature - their data-types, the number of missing values they contain, the average if the feature is numeric, and the 3 mode values if it is not. This is followed by data clean-up - dealing with null values, encoding categorical data into numeric values to better prepare for machine learning models in the future scope of the study. Once processed, the dataset is analysed using a correlation matrix, calculation of metrics like Z-Score for identifying outliers, and information gain of each column. For a thorough analysis, a ‘grouped mean’ approach is also implemented, which will be discussed later in the report.

2. Data Loading and Initial Understanding

This study is performed with Python, and uses the following libraries and modules:

- **Pandas**: for file reading and creation, data manipulation, and creating a 'Data Frame' to work with. It provides high-level tools for structured data manipulation.
- **NumPy**: offers optimized numerical operations and array structures required for statistical preprocessing in this study.
- **Matplotlib** and **Seaborn**: for elementary and advanced data visualization respectively.
- **Scikit-Learn**: for calculation of information gain of each feature, in addition to machine-learning utilities, including feature-selection metrics such as mutual information.
- **SciPy**: specifically, the stats module, for calculation of Z-Score of each feature.
- **iPyWidgets**: used to generate interactive widgets directly in the notebook. In this study it was implemented in order to create a function capable of generating graphs between any column and target variable, depending on the user's requirements.

The analysis begins by importing the excel file and preparing a raw data-frame using Pandas library. The **.read_excel()** function is used to do this in one single clean line for excel files. Now for further analysis it is important to understand the data. In order to achieve this, the methodology implemented here is generation of a schema (or a blueprint) of the raw data, containing the column names, their respective data types, the number of missing values in each column and finally the average value of each column - if the column is numeric, otherwise the top 3 occurring values (modes) of the column. This gives a clearer insight into the data present in the columns.

There are other Pandas functions that can help further understand the data. For the purpose of this study, the functions used are:

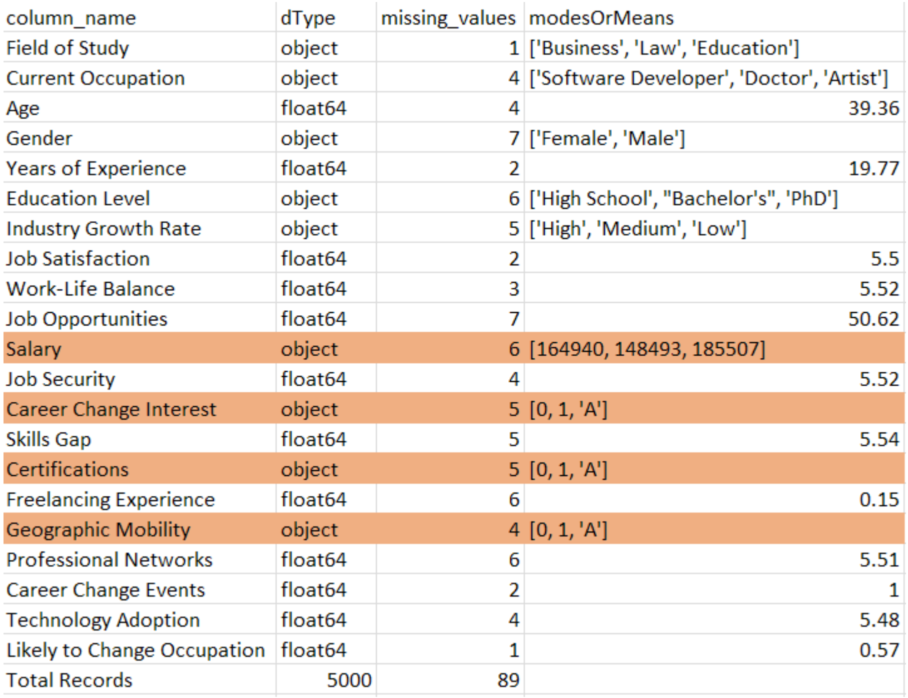
- **.info()**: generates something similar to the schema, showing columns, not-null counts, and data type of columns.
- **.describe()**: displays crucial summary about each column - mean, standard deviation, 1st, 2nd and 3rd quartiles, min and max values. It is notable that this only includes numeric features because of the nature of the calculations it performs.
- **.shape()**: displays the number of rows and columns, which in this dataset before processing is 5000 rows and 21 columns.
- **.head()**: returns the first 5 records of the dataset. This can be modified to display more records by passing the desired number as a parameter to the function.

After obtaining an initial understanding of the dataset, the analysis proceeds to the next stage – pre-processing and clean-up.

3. Data Pre-processing

It was noted that via the preliminary schema (*Figure 1*) there were 89 null values in total across all columns. However, the schema also revealed anomalies in a few columns. Firstly, the **Salary** column - which by logic should be numeric, had the datatype ‘object’, indicating impurities. Secondly, the columns **Career Change Interests**, **Certifications**, and **Geographic Mobility** also had datatypes ‘object’, but when looking at the dataset and what the columns represent, all three of them are columns that should be having binary values (0 and 1). But since they are of datatype ‘object’, the code for schema generation treated them as non-numeric and gave the 3 mode values for them. This revealed the source of the issue - there were a few values which had neither 0 nor 1, but the alphabet ‘A’. Fortunately, all four anomalies are treatable with the same approach - by using the Pandas function **.to_numeric()** on the columns. In the parameter of this function, it was important to specify the “error” parameter as **error=‘coerce’**. In essence, it tries to convert all the values in the column into numeric values, for instance a string “5000” becomes an integer 5000. If that is not possible, however, the error=’coerce’ intervenes and converts the value to **NaN**.

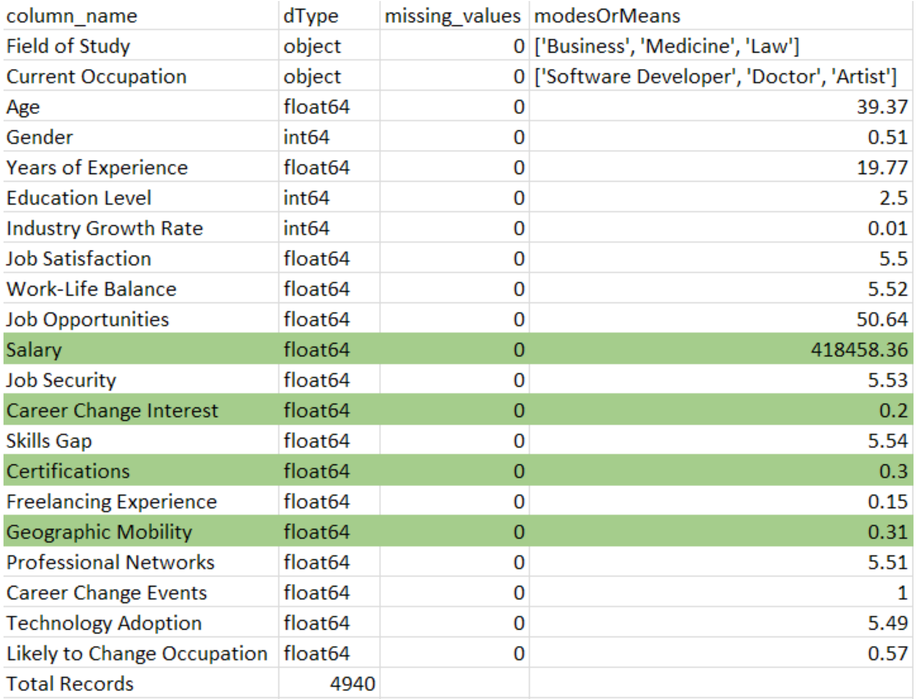
After performing this operation, the null values increased slightly, from 89 to 94. Since the null values made up only 1.88% of the total record count of 5000, the records containing them were deleted (using **.dropna(how = ‘any’)**) as this would not influence the distribution, instead of trying to fill them. The final step involved resetting the indexes of the data-frame as Pandas maintains the indexes before dropping records. This is done using the **.reset_index(drop = True)** function of the library.



column_name	dType	missing_values	modesOrMeans
Field of Study	object	1	['Business', 'Law', 'Education']
Current Occupation	object	4	['Software Developer', 'Doctor', 'Artist']
Age	float64	4	39.36
Gender	object	7	['Female', 'Male']
Years of Experience	float64	2	19.77
Education Level	object	6	['High School', "Bachelor's", 'PhD']
Industry Growth Rate	object	5	['High', 'Medium', 'Low']
Job Satisfaction	float64	2	5.5
Work-Life Balance	float64	3	5.52
Job Opportunities	float64	7	50.62
Salary	object	6	[164940, 148493, 185507]
Job Security	float64	4	5.52
Career Change Interest	object	5	[0, 1, 'A']
Skills Gap	float64	5	5.54
Certifications	object	5	[0, 1, 'A']
Freelancing Experience	float64	6	0.15
Geographic Mobility	object	4	[0, 1, 'A']
Professional Networks	float64	6	5.51
Career Change Events	float64	2	1
Technology Adoption	float64	4	5.48
Likely to Change Occupation	float64	1	0.57
Total Records	5000	89	

Figure 1 Schema of Raw Data

Post this, it was decided that columns with categorical values, specifically for **Gender** and **Education Level**, could be replaced with integers. Performing this step, called **encoding**, specifically manual label mapping, leads to the data being better suited towards use by machine learning models, as the models may not be able to identify the relationship between strings ‘high’, ‘medium’ and ‘low’ for example, but they would understand the relationship between integers 1, 0 and -1. This concludes data cleaning. A new schema of the processed data is generated (*Figure 2*) using the same custom function, and as expected the datatypes are correctly defined and the null values have been removed.



column_name	dType	missing_values	modesOrMeans
Field of Study	object	0	['Business', 'Medicine', 'Law']
Current Occupation	object	0	['Software Developer', 'Doctor', 'Artist']
Age	float64	0	39.37
Gender	int64	0	0.51
Years of Experience	float64	0	19.77
Education Level	int64	0	2.5
Industry Growth Rate	int64	0	0.01
Job Satisfaction	float64	0	5.5
Work-Life Balance	float64	0	5.52
Job Opportunities	float64	0	50.64
Salary	float64	0	418458.36
Job Security	float64	0	5.53
Career Change Interest	float64	0	0.2
Skills Gap	float64	0	5.54
Certifications	float64	0	0.3
Freelancing Experience	float64	0	0.15
Geographic Mobility	float64	0	0.31
Professional Networks	float64	0	5.51
Career Change Events	float64	0	1
Technology Adoption	float64	0	5.49
Likely to Change Occupation	float64	0	0.57
Total Records	4940		

Figure 2 Schema of Cleaned Data

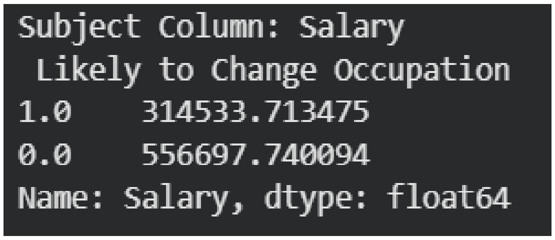
4. Exploratory Data Analysis

In the context of this dataset, it is important to be able to determine the impact a change in any feature will have on the target variable. This is what in the end helps empower data driven decision making. Correlation analysis assumes linear relationships, which is suitable here given the numeric or ordinal nature of most variables. To do this, a correlation matrix has been created using Pandas method **.corr()**.

The next step was to find outliers. To achieve this, it is best to calculate the Z-Score of each value of each feature with respect to other values in that column. This value denotes how far the value is from the mean of the column. Following the **empirical rule**, we know that typically, values having a Z-Score more than +3 or less than -3 indicate that it is an outlier, because they are 3 times the standard deviation away from the mean, implying they are extreme values not fitting the normal distribution of data (Lehmann and Rüdiger, 2013)

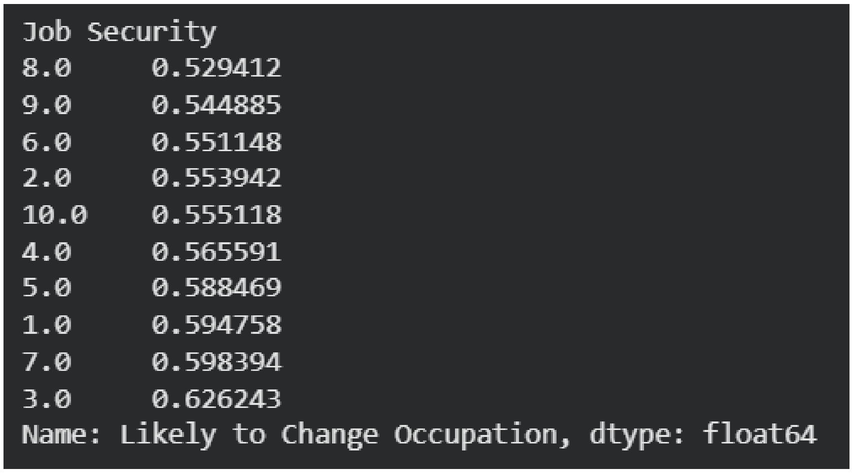
Another method that can provide hints about the relevance of each feature is calculating the information gain. Higher information gain – a metric of probability – indicates that knowing the value of the feature reduces uncertainty in the target variable, meaning it has practical predictive value (Kuzudisli et al., 2023). It is performed, in this study, using the **mututal_info_score()** method of the Scikit-Learn library in Python. Visualizing this on a horizontal bar chart can improve the understanding of the relevance of the columns in context of the target variable.

Additionally, grouping the data by 'Likely to Change Occupation' column and calculating mean from the resultant data of any specific column. This is the aforementioned 'grouped mean' approach. A sample output for the code is displayed in *Figure 3* for the **Salary** column. From this result we can see what was the average salary of people who are likely to change their job, and those who were not; and it can be concluded that there was a notable gap in the salary of both of these groups. An alternative approach involves reversing the grouping process – group by a specific column and calculate the mean of the 'Likely to Change Occupation' column, a sample output shown in *Figure 4*. In this case, it shows how likely it is that someone switches jobs, given a certain level of job security. It can therefore be interpreted that **Job Security** did not play a significant role in changing the target variable, as the minimum and maximum value of the result are not too far apart - minimum was 0.529 and maximum was 0.626. It should be noted that all values are sorted for the convenience of understanding, as is clear in *Figure 4*. These grouped mean comparisons directly support the study objective by revealing which feature levels distinguish individuals with higher or lower likelihood of occupation change.



```
Subject Column: Salary
 Likely to Change Occupation
1.0    314533.713475
0.0    556697.740094
Name: Salary, dtype: float64
```

Figure 3 Grouped by Likely to Change Occupation, mean of Salary



```
Job Security
8.0     0.529412
9.0     0.544885
6.0     0.551148
2.0     0.553942
10.0    0.555118
4.0     0.565591
5.0     0.588469
1.0     0.594758
7.0     0.598394
3.0     0.626243
Name: Likely to Change Occupation, dtype: float64
```

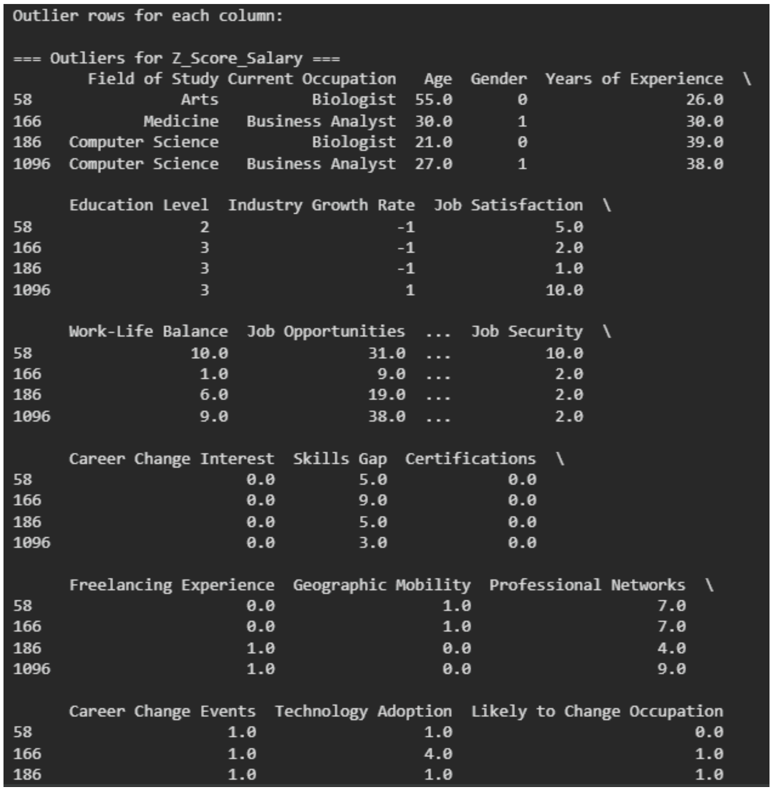
Figure 4 Grouped by Job Security, mean of Likely to Change Occupation

As the final step of the analysis, an interactive graph generator was created and implemented. Its purpose is to generate graphs and charts between any column, and the target column ‘Likely to Change Occupation’. Furthermore, an option to choose the type of graph is also provided in order to provide anyone analyzing this dataset with multiple options to choose from, depending on what they may be looking for. The Python library at work here is iPyWidgets, which helped create the dropdowns for selecting the type of graph and the column for which the data is to be visualized. The options for the type of graph included scatter plot, histogram, box plot, count-plot, and bar chart. This tool enhances exploratory analysis by allowing quick and efficient understanding without writing repetitive plotting code.

5. **Findings and Discussion**

Upon analysis of this data using the correlation matrix, it was observed that the highest correlation with the target feature ‘Likely to Change Occupation’ was present in the ‘Job Satisfaction’ feature, with an inverse relationship of -0.597, followed by the column ‘Career Change Interest’ with a direct relationship of 0.431. All the other remaining columns had more than 10 times less impact.

When discussing outliers, the Z-Score of each feature was calculated, which showed that only the **Salary** column had values with the Z-Score more than 3 and hence was the only column with outliers, **four** to be exact, displayed in *Figure 5*. These can be removed at the time of implementing a machine learning model in order to normalize the data, but for the purpose of this study, data was left as intact as possible.



```
Outlier rows for each column:

=== Outliers for Z_Score_Salary ===
        Field of Study Current Occupation   Age  Gender  Years of Experience  \
58                Arts          Biologist  55.0       0                 26.0
166           Medicine   Business Analyst  30.0       1                 30.0
186   Computer Science          Biologist  21.0       0                 39.0
1096  Computer Science   Business Analyst  27.0       1                 38.0

      Education Level  Industry Growth Rate  Job Satisfaction  \
58                  2                    -1               5.0
166                 3                    -1               2.0
186                 3                    -1               1.0
1096                3                     1              10.0

      Work-Life Balance  Job Opportunities  ...  Job Security  \
58                 10.0               31.0  ...          10.0
166                 1.0                9.0  ...           2.0
186                 6.0               19.0  ...           2.0
1096                9.0               38.0  ...           2.0

      Career Change Interest  Skills Gap  Certifications  \
58                       0.0         5.0             0.0
166                      0.0         9.0             0.0
186                      0.0         5.0             0.0
1096                     0.0         3.0             0.0

      Freelancing Experience  Geographic Mobility  Professional Networks  \
58                       0.0                  1.0                    7.0
166                      0.0                  1.0                    7.0
186                      1.0                  0.0                    4.0
1096                     1.0                  0.0                    9.0

      Career Change Events  Technology Adoption  Likely to Change Occupation
58                     1.0                  1.0                          0.0
166                    1.0                  4.0                          1.0
186                    1.0                  1.0                          1.0
```

Figure 5 Records with Outliers

Additionally, plotting the results of the information gain calculated on a horizontal bar chart (using **Matplotlib**) gives a clear visual understanding, and it was observed that **Salary**, **Job Satisfaction** and **Career Change Interests** were the 3 columns that had the most effect on a subject changing their profession, in that order, with their values being 0.676, 0.317 and 0.129 respectively. **Job Opportunities** also had some degree of effect with a mutual information score of 0.011, but as for the other factors, there was no notable impact. These columns therefore imply potential scope of improvement, or where the employees may be experiencing dissatisfaction, leading them to change occupations.

The grouped mean analysis showed interesting results, highlighting the correlation between columns once again, for instance, age played an insignificant role in the likelihood of changing professions as the ones least likely to do so are aged 51 years and 28 years old, and the overall distribution was observed to be random. Another way of using the grouped mean method that was implemented was to swap both factors, that is treating the ‘Likely to Change Occupation’ column as the basis for grouping. This method led to the conclusion of what the average value was for any column, divided into two categories – of the people likely to switch their jobs and the ones who were not.

6. Recommendations and Reflection

After pre-processing and analyzing it can be stated that the dataset has high integrity, and needed removal of less than 2% of the records until it could be classified as cleaned data. The dataset primarily had ordinal and binary values, and once the relevant columns were encoded to be numeric, the dataset has high potential for being eligible for machine learning models, for example linear regression / logistical regression.

As identified, the relevant features **Salary**, **Career Change Interest**, and **Job Satisfaction** can be targeted to predict the likelihood of individuals switching their jobs. This can further help employers or organisations make data driven decisions, and upkeep or improve the fields that can help them preserve employees.

Similar kinds of studies can be performed for organisations which can quantify the output of their employees and their experiences, in order to retain long term employees. This study itself can also be improved upon by implementing multi-column correlation, in other words how multiple features interact with one another and their final impact on the decision of an individual to switch professions.

A certain limitation of the data set for this kind of study is that it can be difficult to ascertain accuracy, since the decision to switch occupation is in the end a personal decision, and therefore can be affected by unquantifiable reasons like family situation or personal interests and motivations (Caliendo et al., 2014). Records like these are anomalies that can hamper the overall accuracy, for example, someone earning the highest salary among the population may yet choose to change their profession for personal reasons, but it would lead a ‘mean salary’ based machine learning model to believe that salary is less relevant than it actually is for a regular subject of the study, therefore skewing the results.

7. Conclusion

In summary, this study is based on a job churn dataset, featuring demographic and professional details of individuals, and whether or not they are likely to change their occupation. The data was studied deeply, and after developing an understanding, it went through the pre-processing steps, which included using the Pandas library to ensure the columns had the right data types, removal of all missing values, encoding the categorical and ordinal data into integers in order for better understanding by the machine learning algorithms, exporting the processed data as well as generation of a schema to cross-verify that there are no more anomalies or inconsistencies.

After primary clean up, the Pandas data-frame was then used to find and understand trends and correlations between each feature and the target column ‘Likely to Change Occupation’ using the .corr() method. Furthermore, information gain was also calculated between the features to determine the certainty of what the likelihood of changing jobs would be given we know the value of any other column. A code snippet was also written to highlight the outliers in all numeric columns by calculating their Z-Scores. Additionally, a grouped mean approach was taken to quantify which groups of data in each column had what kind of relationship with the target feature.

Finally, a custom Python function was created to let the user plot graphs as per their requirements, with a dropdown menu created with the help of iPyWidgets library, providing choices between which feature to plot against the target column, as well as which type of graph. This was meant to boost any future analyst’s understanding of the data without them having to write the code manually for each graph they want to plot.

8. References

Caliendo, M., Mahlstedt, R., Mitnik, O. A., (2014), Unobservable, but Unimportant? The Influence of Personality Traits (and Other Usually Unobserved Variables) for the Evaluation of Labor Market Policies

Kuzudisli, C., Bakir-Gungor, B., Bulut, N., Qaqish, B. and Yousef, M., (2023), Review of feature selection approaches based on grouping of features, PeerJ, Vol 11

Lehmann, Rüdiger, (2013), Journal of Surveying Engineering, Vol. 139 Issue 4, p157-165

NumPy, v2.3, available from: https://numpy.org/doc/stable/reference/routines.ma.html

Pandas, v2.3, available from: https://pandas.pydata.org/docs/reference/index.html

SciPy, v1.16.2, available from: https://docs.scipy.org/doc/scipy/reference/

iPyWidgets,v8.1.8,available from: https://ipywidgets.readthedocs.io/en/latest/how-to/index.html

Matplotlib, v3.10, available from: https://matplotlib.org/stable/api/index.html

Seaborn, v0.13, available from: https://seaborn.pydata.org/api.html

SciKit-Learn, v1.7.2, available from: https://scikit-learn.org/stable/api/index.html

9. Appendix

Source Code (GitHub): https://github.com/LazyShinigami/programming-assessment-24152182