Data Management Algorithm for Decision Making
02360003

Home Assignment 3 - Dry Part

Ido Tausi - 214008997
Afek Nahum - 214392706
Or Mutay - 206918633

Problem 1: Handling Missing Values

1.

The difference between the different types is that Missing at random means that the likelihood of a value being missing depends only on observed variables and not on the missing value itself, that's compared to Missing completely at random, which means that the likelihood of a value being missing is unrelated to any observed or unobserved variables in the database. Both of these differ from Missing not at random, which states that the likelihood of a value being missing depends on itself or on unobserved factors.

Examples - Income Survey:

Missing at random: Younger participants are more likely to leave the income field empty due to their age, not because of the income itself.

Missing completely at random: Some participants skip the income question due to a random bug in the survey form, unrelated to any participant characteristics.

Missing not at random: High-income individuals may not disclose their income due to privacy concerns.

2.

```
Accuracy using KNN: 1.0
Accuracy using Iterative Imputer: 1.0
Accuracy using Mean Imputation: 1.0
```

All three imputation methods (KNN, Iterative Imputer, and Mean Imputation) achieved perfect accuracy (1.0) on the Iris dataset. This result is likely because the dataset is simple, with highly separable classes, and the missing data had minimal impact overall. Since the missing values were limited to one feature (sepal length (cm)), the other features likely compensated for any imputation differences. None of the methods outperformed the others, as the dataset's simplicity allowed the model to perform well regardless of the imputation method.

We selected the "Titanic - Machine Learning from Disaster" dataset.

```
Accuracy using KNN: 0.7821229050279329
Accuracy using Iterative Imputer: 0.7877094972067039
Accuracy using Mean Imputation: 0.8044692737430168
```

The accuracy varied across the three imputation methods. Mean Imputation achieved the highest accuracy (0.804), followed by Iterative Imputer (0.788), and KNN (0.782). This suggests that in the Titanic dataset, the missing values in the Age feature did not significantly disrupt the relationships between the features and the target, allowing the simpler Mean

Imputation to perform well. The lower performance of KNN and Iterative Imputer might be due
to noise introduced by their more complex methods, which could not provide a significant advantage for this dataset.
Overall, the results indicate that for this particular problem, simpler imputation can be as effective or better than advanced methods, depending on the dataset's complexity and the nature of the missing data.

3.

```
Dataset: Iris, Method: KNN, Accuracy: 0.7821229050279329
Dataset: Iris, Method: Iterative Imputer, Accuracy: 0.7653631284916201
Dataset: Iris, Method: Mean Imputation, Accuracy: 0.7541899441340782

Dataset: Titanic, Method: KNN, Accuracy: 0.7821229050279329
Dataset: Titanic, Method: Iterative Imputer, Accuracy:
0.7653631284916201
Dataset: Titanic, Method: Mean Imputation, Accuracy: 0.7541899441340782
```

With $MCAR$ missing values, the accuracies decreased for both datasets compared to the $MNAR$ scenario. This happened because the missing values were distributed randomly, making it harder for imputation methods to predict them accurately. In the Iris and Titanic datasets, KNN performed best because it uses relationships between nearby data points, while Mean Imputation performed the worst as it doesn’t account for feature relationships. The randomness of $MCAR$ disrupts patterns in the data, reducing the effectiveness of all methods.

4.

Check for Missing Completely at Random (MCAR):

- Analyze patterns in missing data using visualizations (for example heatmaps).
- Randomly shuffle the data and check if the missingness pattern changes. If no patterns emerge, the data is likely MCAR.

Check for Missing at Random (MAR):

- Analyze relationships between missingness and observed variables.
- Use a simple statistical model (for example logistic regression) where missingness is the target and observed variables are predictors. If relationships exist, the data is likely MAR.

Check for Missing Not at Random (MNAR):

- Estimate the missing values (for example using mean or median imputation).
- Compare the distribution of observed and imputed values. If distributions differ significantly, the data might be MNAR.

Problem 2: Data inconsistency

1.

- $A \rightarrow B,\ B \rightarrow C \overset{\substack{axiom\ of \\ transitivity}}{\Longrightarrow} A \rightarrow C$
- $\underbrace{A \rightarrow B}_{\text{נתון}},\ \underbrace{A \rightarrow C}_{\text{מסעיף קודם}} \overset{union}{\Longrightarrow} A \rightarrow BC$
- $A \rightarrow C \overset{\substack{axion\ 2\ from \\ lecture}}{\Longrightarrow} AB \rightarrow C$
- $A \rightarrow C \overset{\substack{axion\ 2\ from \\ lecture}}{\Longrightarrow} AD \rightarrow C$

2.

(a)

$A \twoheadrightarrow B$

The dataset violates $A \twoheadrightarrow B$. We will give an example for that, showing a pair of tuples that does not satisfy the MVD conditions:
We'll choose: $t_1 = \{A:1,\ B:0,\ C:1\},\ t_2 = \{A:1,\ B:1,\ C:0\}$
Now, there are no pairs of tuples satisfying all 3 MVD conditions. That is because the first condition narrows down the options for t_3, t_4 to be only the third or fourth rows from R. However, none of the combinations of these 2 tuples will lead to a satisfaction of the rest of the MVD conditions. since if we'll choose
$t_3 = \{A:1,\ B:0,\ C:1\},\ t_4 = \{A:1,\ B:1,\ C:1\}$ the second condition is satisfied ($t_1[A] = t_3[A] = 0,\ t_2[A] = t_4[A] = 1$).
The third one is not satisfied since $t_3[C] \neq t_2[C]$.
If we choose the second option for t_3, t_4 it also doesn't work, since if
$t_3 = \{A:1,\ B:1,\ C:1\},\ t_4 = \{A:1,\ B:0,\ C:1\}$, the second condition is not satisfied, since $t_3[B] \neq t_1[B]$.
Therefore the MVD conditions for the pair (t_1, t_2) as we described are not satisfied, and thus the dataset violates $A \twoheadrightarrow B$.

$B \perp C \mid A$

This dataset violates $B \perp C \mid A$, we will give a counter example.
Given $A = 1$, we get $P[B = 1 \mid A = 1] = 0.5$
Now, assume $C = 1$, we get $P[B = 1 \mid A = 1, C = 1] = \frac{1}{3}$.
Therefore according to the definition we get a contradiction because:

$$0.5 = P[B = 1 \mid A = 1] \neq P[B = 1 \mid A = 1, C = 1] = \frac{1}{3}.$$

In conclusion, $B \perp C \mid A$ is violated in the dataset.

(b)

To satisfy both conditions we modified the data base

We removed all the rows from the table except the 2 rows of $[A:2,\ B:1,\ B:1]$. After that we added 2 rows of $[A:2,\ B:1,\ B:1]$, marked in yellow.

A	B	C
2	1	1
2	1	1
2	1	1
2	1	1

Now, we'll prove that both conditions are met:

$A \twoheadrightarrow B$:

The dataset satisfies this condition trivially - the same tuple repeats 4 times, so for each pair of tuples t_1 and t_2 from the dataset there is another pair t_3 and t_4 such that:

1. $t_1[A] = t_2[A] = t_3[A] = t_4[A] = 2$
2. $t_1[B] = t_3[B] = 1, t_2[B] = t_4[B] = 1$
3. $t_1[C] = t_3[C] = 1, t_2[c] = t_4[C] = 1$

Thus, the condition is satisfied by this database.

$B \perp C \mid A$:

This dataset satisfies this condition, given $A = 2$ (it's the only option) we get

$P[B = 1 \mid A = 2] = 1,\ P[C = 1 \mid A = 2] = 1.$

Assuming $B = 1$ (its the only option), we get $P[C = 1 \mid A = 2, B = 1] = 1.$

Assuming $C = 1$ (again, its the only option), we get $P[B = 1 \mid A = 2, C = 1] = 1.$

Therefore, if A is given, the knowledge of B doesn't give us any information on C, and the other way around, so $B \perp C \mid A$ is satisfied.

Problem3: Duplicate removal and outlier detection

1.

(a)

```
With Duplicates:
Accuracy: 0.98
Precision: 0.98
Recall: 0.98

Class-wise Accuracy:
setosa Accuracy: 1.00
versicolor Accuracy: 1.00
virginica Accuracy: 0.92

Without Duplicates:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00

Class-wise Accuracy:
setosa Accuracy: 1.00
versicolor Accuracy: 1.00
virginica Accuracy: 1.00
```

When duplicates are introduced into a dataset, they can influence a machine learning model's performance by altering the distribution and emphasis on certain data points.
In this experiment with the Iris dataset, the results showed minimal impact, as the dataset is balanced and clean. The model achieved nearly identical accuracy, precision, and recall with and without duplicates, except for a slight drop in class-wise accuracy for the virginica class when duplicates were present.

We selected the "Wine Quality" dataset

```
With Duplicates:
Accuracy: 0.60
Precision: 0.59
Recall: 0.60

Class-wise Accuracy:
Class 3 Accuracy: 0.00
Class 4 Accuracy: 0.67
Class 5 Accuracy: 0.67
Class 6 Accuracy: 0.55
Class 7 Accuracy: 0.52
Class 8 Accuracy: 0.00

Without Duplicates:
Accuracy: 0.60
Precision: 0.58
Recall: 0.60

Class-wise Accuracy:
Class 3 Accuracy: 0.00
Class 4 Accuracy: 0.00
Class 5 Accuracy: 0.66
Class 6 Accuracy: 0.55
Class 7 Accuracy: 0.67
Class 8 Accuracy: 0.00
```

For the Wine Quality dataset, duplicates had minimal impact on overall accuracy (0.60 with and without duplicates), but class-wise performance showed inconsistencies. Some classes, like 3 and 8, had zero accuracy in both cases due to the dataset's imbalance, while others, such as 5 and 7, saw slight changes in performance. This highlights that duplicates increase the influence of majority classes in imbalanced datasets, leading to skewed predictions and bad representation for minority classes.

(b)

Iris Dataset:

```
--- Experiment with Iris Dataset ---

Duplicate Percentage: 50% for Class 0(setosa)
Accuracy: 1.00
Precision: 1.00
Recall: 1.00

Class-wise Accuracy:
setosa Accuracy: 1.00
versicolor Accuracy: 1.00
virginica Accuracy: 0.93

Duplicate Percentage: 100% for Class 0(setosa)
Accuracy: 1.00
Precision: 1.00
Recall: 1.00

Class-wise Accuracy:
setosa Accuracy: 1.00
versicolor Accuracy: 1.00
virginica Accuracy: 0.93

Duplicate Percentage: 200% for Class 0(setosa)
Accuracy: 1.00
Precision: 1.00
Recall: 1.00

Class-wise Accuracy:
setosa Accuracy: 1.00
versicolor Accuracy: 1.00
virginica Accuracy: 0.93
```

The results for the Iris dataset show that duplicating the setosa class (Class 0) had no significant impact on overall accuracy, precision, or recall, as they remained at 1.00 for all duplication levels (50%, 100%, and 200%). Class-wise accuracy for setosa and versicolor remained perfect at 1.00, while virginica showed a slightly lower accuracy of 0.93 across all scenarios. This behavior suggests that the Iris dataset is well balanced and separable, meaning the classifier could already differentiate between classes effectively without being impacted by the additional weight from duplicates.

Wine Quality Dataset:

```
--- Experiment with Wine Quality Dataset ---

Duplicate Percentage: 50% for Class 5
Accuracy: 1.00
Precision: 0.99
Recall: 1.00

Class-wise Accuracy:
Class 3 Accuracy: 0.00
Class 4 Accuracy: 0.00
Class 5 Accuracy: 1.00
Class 6 Accuracy: 0.00
Class 7 Accuracy: 0.67
Class 8 Accuracy: 0.00

Duplicate Percentage: 100% for Class 5
Accuracy: 1.00
Precision: 1.00
Recall: 1.00

Class-wise Accuracy:
Class 3 Accuracy: 0.00
Class 4 Accuracy: 0.00
Class 5 Accuracy: 1.00
Class 6 Accuracy: 0.00
Class 7 Accuracy: 0.00
Class 8 Accuracy: 0.00

Duplicate Percentage: 200% for Class 5
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
```

For the Wine Quality dataset, duplicating Class 5 yielded different results due to the dataset's imbalanced nature and classes boundaries that are overlapping. While overall accuracy, precision, and recall reached 1.00, these metrics fail to reflect the poor class-wise performance for minority classes (Classes 3, 4, 6, and 8), which had zero accuracy across all duplication levels. Duplicating Class 5 increased its representation, leading to perfect accuracy for Class 5 but further ignores other classes. This imbalance caused the classifier to

focus disproportionately on the majority class (Class 5) while ignoring minority classes, as shown by the lack of predicted samples for Classes 3, 4, 6, and 8 in the 100% and 200% duplication scenarios.

2.

(a)

These outliers in the Blood Pressure variable, which serves as the outcome variable, can significantly impact the estimation of the ATE. The ATE measures the difference in the average outcomes between treated and untreated groups. If there are extremely high or low Blood Pressure values, these outliers can influence the mean calculation within each group, leading to a biased ATE estimate.

Outliers with low/high blood pressure in the treated groups can cause a decrease/increase the ATE, thus changing our estimation. For example, in the case of extremely high Blood Pressure readings in the treated group, these outliers might artificially inflate the group's average, making the treatment appear less effective than it truly is (under the assumption that the goal of the new drug is to decrease the blood pressure).

Similarly, outliers in the control group could lead to an underestimation or overestimation of the baseline blood pressure, further distorting the ATE.

(b)

To mitigate the impact of outliers on ATE estimation, we can use statistical methods.

One approach is to use trimmed means, where a small percentage of the highest and lowest values in the Blood Pressure variable are excluded from the calculation.

Alternatively, we cap extreme values at a specified percentile based on domain knowledge, reducing their influence without discarding them entirely.

(c)

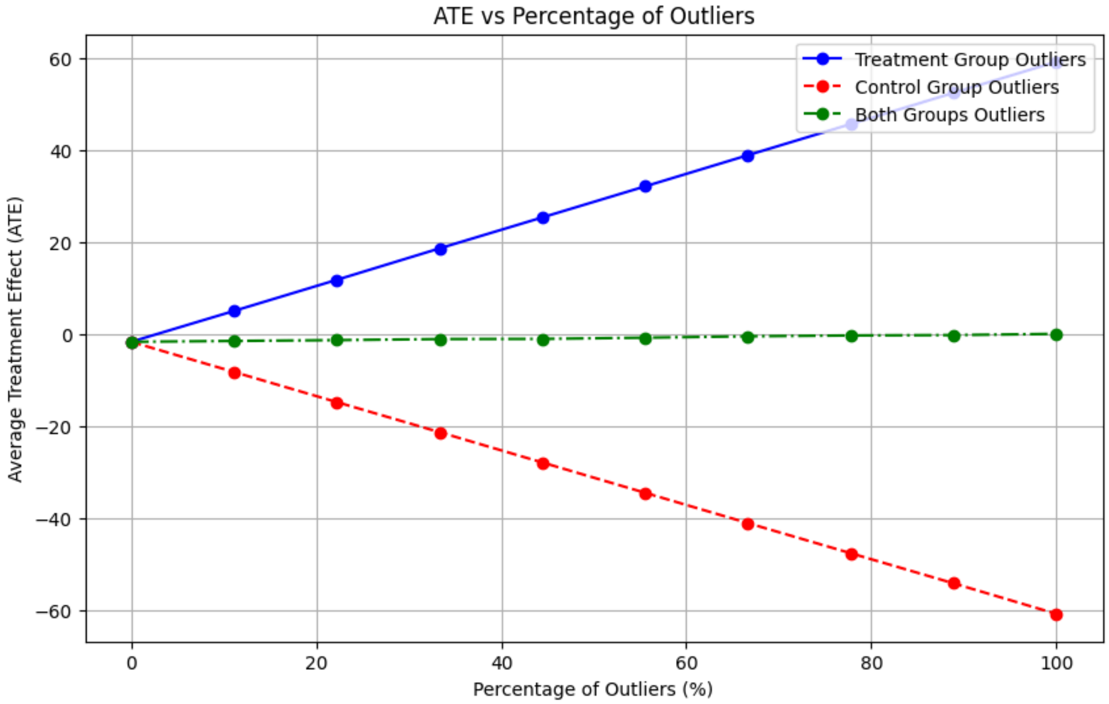
(a) If the outliers were present in the Age variable, the ATE estimation would be indirectly affected through the confounding adjustment process.

Extreme Age values could disproportionately influence the model used to estimate treatment probabilities or the baseline relationship between Age and Blood Pressure. This distortion could result in residual confounding, where the adjustment fails to fully account for Age’s effect, leading to a biased ATE.

Unlike Blood Pressure outliers, which directly affect the mean of the outcome variable, Age outliers impact how the treatment and control groups are adjusted, potentially causing inaccurate baseline estimates and an unreliable ATE.

(b) To mitigate the impact of outliers in the Age variable, the approach would need to focus on limiting the influence of extreme Age values during confounder adjustment. Instead of trimming or capping Blood Pressure values we should trim Age values at reasonable thresholds to help ensure that we are not too much influenced by outliers. Another method in this case can be using regression techniques to reduce the impact of extreme Age values on the adjustment process.

(d)



Under the assumption that the goal of the new drug is to decrease the blood pressure:

- Outliers in the Treatment Group: Outliers with high Blood Pressure in the treatment group inflate the group's mean, leading to an overestimated ATE. This makes the drug appear less effective at lowering Blood Pressure than it truly is.

- Outliers in the Control Group: Outliers with high Blood Pressure in the control group inflate its mean, resulting in a higher ATE. This exaggerates the drug's effectiveness by making the control group seem to have much higher Blood Pressure.

- Outliers in Both Groups: Outliers in both groups can cancel each other out if balanced, keeping the ATE stable.
  However, If there were any imbalances, it would have skew the results, making the drug's effect appear stronger or weaker depending on which group has more extreme outliers.