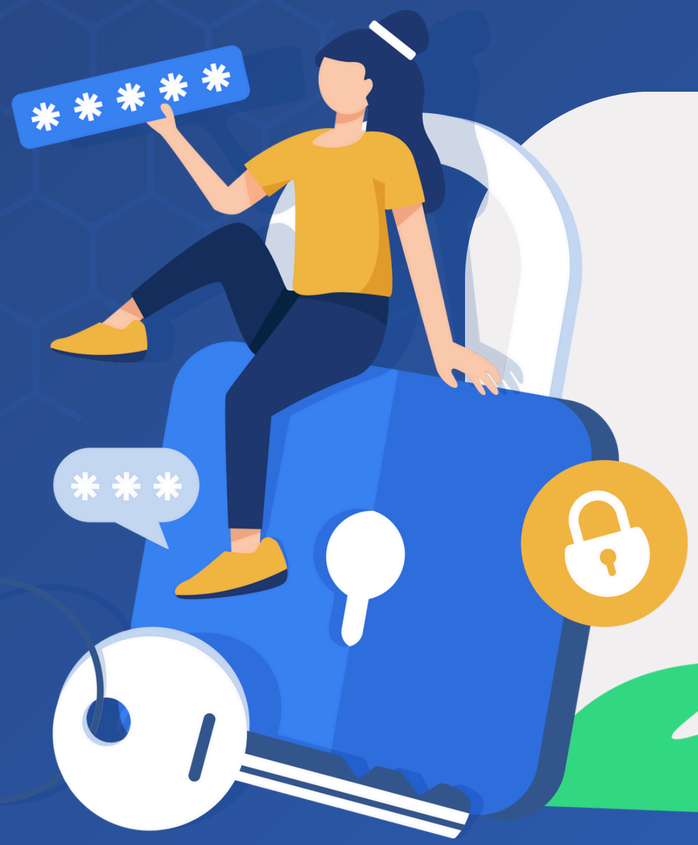


# EXPLORATORY DATA ANALYSIS

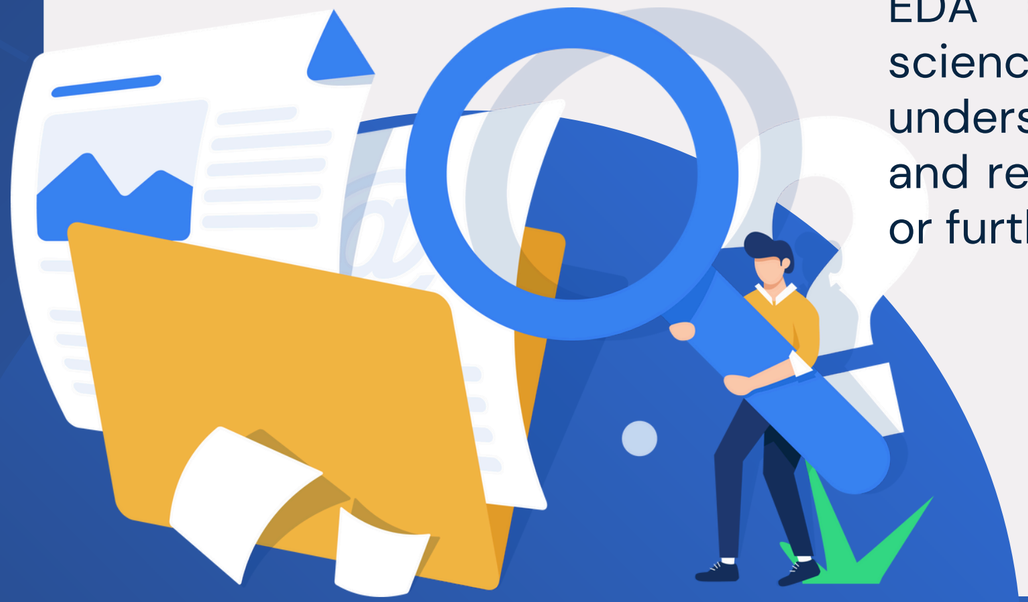
DATA SCIENCE PORTOFOLIO
BY AZHAR FRENOTAMA

# INTRODUCTION

I am a student of SMK Telkom Purwokerto majoring in Software and Game Development (PPLG) with a deep interest in the world of technology, especially in programming, design, and video editing. Since childhood, I have an interest in gadgets and games, which motivates me to continue learning and developing in this field. At school, I also learned programming and joined extracurricular graphic design to develop my skills to a higher level.

# TABLE OF CONTENT

# WHAT IS EDA?



EDA (Exploratory Data Analysis) in data science is the initial process in data analysis to understand the structure, patterns, anomalies, and relationships in a dataset before modeling or further analysis.

# PORTOFOLIO

In this portfolio, I showcase the Exploratory Data Analysis (EDA) process on the legendary Titanic dataset, which is often used as a benchmark in the data science world. The main focus of this analysis is to observe the data structure, handle missing values, and remove duplicate data that could compromise the accuracy of the analysis.

# GOAL

DATA OBSERVATION

HANDLING DUPICATES DATA

HANDLING MISSING VALUES

```
df = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/titanic.xlsx')
data = df.copy()
data.head()
```

| | survived | name | sex | age |
|---|---|---|---|---|
| 0 | 1 | Allen, Miss. Elisabeth Walton | female | 29.0000 |
| 1 | 1 | Allison, Master. Hudson Trevor | male | 0.9167 |
| 2 | 0 | Allison, Miss. Helen Loraine | female | 2.0000 |
| 3 | 0 | Allison, Mr. Hudson Joshua Creighton | male | 30.0000 |
| 4 | 0 | Allison, Mrs. Hudson J C (Bessie Waldo Daniels) | female | 25.0000 |

```
[ ] data.tail()
```

| | survived | name | sex | age |
|---|---|---|---|---|
| 495 | 1 | Mallet, Mrs. Albert (Antoinette Magnin) | female | 24.0 |
| 496 | 0 | Mangiavacchi, Mr. Serafino Emilio | male | NaN |
| 497 | 0 | Matthews, Mr. William John | male | 30.0 |
| 498 | 0 | Maybery, Mr. Frank Hubert | male | 40.0 |
| 499 | 0 | McCrae, Mr. Arthur Gordon | male | 32.0 |

```
[ ] data.sample(5)
```

| | survived | name | sex | age |
|---|---|---|---|---|
| 492 | 0 | Malachard, Mr. Noel | male | NaN |
| 86 | 1 | Daly, Mr. Peter Denis | male | 51.0 |
| 111 | 1 | Fortune, Miss. Alice Elizabeth | female | 24.0 |
| 425 | 0 | Givard, Mr. Hans Kristensen | male | 30.0 |
| 320 | 1 | Woolner, Mr. Hugh | male | NaN |

Observation:

1. Column name dan sex berupa string, sedangkan column survived dan age berupa numeric
2. Column survived mengeluarkan output (0 dan 1)
3. Column sex mengeluarkan dua output yaitu male dan female

# LOAD DATA

In the initial stage of exploring the Titanic data, a review was conducted using the head(), tail(), and sample() functions to understand the structure of the data.

The dataset consists of four main columns:

- survived (0 = not survived, 1 = survived),
- name (passenger name),
- sex (gender: male/female),
- age (age in decimal numbers).

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   survived  500 non-null    int64  
 1   name      500 non-null    object 
 2   sex       500 non-null    object 
 3   age       451 non-null    float64
dtypes: float64(1), int64(1), object(2)
memory usage: 15.8+ KB
```

Observation:

1. data memiliki 4 column dengan 500 baris
2. hanya column age yang memiliki missing values

# LOAD DATA

Through the data.info() function, it is known that the Titanic dataset consists of 500 rows and 4 columns, namely survived, name, sex, and age.

Key Findings:
All columns except age have complete data (500 entries).
The age column only has 451 non-null data, which means there are 49 missing values.

The data type of each column:

- survived: integer (int64)
- name and sex: object/string
- age: float64

```
[ ] data[numericals].describe()
```

| | survived | age |
|---|---|---|
| count | 500.000000 | 451.000000 |
| mean | 0.540000 | 35.917775 |
| std | 0.498897 | 14.766454 |
| min | 0.000000 | 0.666700 |
| 25% | 0.000000 | 24.000000 |
| 50% | 1.000000 | 35.000000 |
| 75% | 1.000000 | 47.000000 |
| max | 1.000000 | 80.000000 |

```
data[categoricals].describe()
```

| | name | sex |
|---|---|---|
| count | 500 | 500 |
| unique | 499 | 2 |
| top | Eustis, Miss. Elizabeth Mussey | male |
| freq | 2 | 288 |

observation:

1. Column sex mempunyai 2 unique value

# STATISTICAL SUMMARY

Via the .describe() function for numeric data:

- The survived column has an average value of 0.54, meaning that about 54% of passengers survived.
- The age column has an average age of 35.91 years, with an age range from 0.6667 to 80 years.
- There are 451 valid age records, indicating 49 missing values.
- The age spread is quite wide with a standard deviation of 14.77, indicating the diversity of passenger ages.

From the .describe() result for categorical data:

- The sex column has 2 unique values: male and female, with the largest number being male (288 people).
- The name column has 499 unique names out of 500 rows of data, indicating one duplicate name.

# DATA DUPLICATION ANALYSIS AND HANDLING

```
[15] len(data.drop_duplicates()) / len(data)

0.998

duplicates = data[data.duplicated(keep=False)]

duplicate_counts = duplicates.groupby(list(data.columns)).size().reset_index(name='jumlah_duplikat')

sorted_duplicates = duplicate_counts.sort_values(by='jumlah_duplikat', ascending=False)

print("Baris yang terduplikasi:")
sorted_duplicates

Baris yang terduplikasi:
```

| | survived | name | sex | age | jumlah_duplikat |
|---|---|---|---|---|---|
| 0 | 1 | Eustis, Miss. Elizabeth Mussey | female | 54.0 | 2 |

```
[17] data = data.drop_duplicates()

len(data.drop_duplicates()) / len(data)

1.0
```

Findings:

- Before cleaning, about 99.8% of the data is unique, indicating the presence of 1 duplicate row.
- The detected duplicate row belongs to Eustis, Miss. Elizabeth Mussey with age 54.0 years and status survived = 1.
- The row appears 2 identical times in the entire column.

Action:

- Duplicates were removed using data.drop_duplicates().
- After cleaning, the proportion of unique data is 100%, indicating all rows are now unique.

# HANDLING MISSING VALUE

```
data.isna().sum()
```

| | 0 |
|---|---|
| survived | 0 |
| name | 0 |
| sex | 0 |
| age | 49 |

dtype: int64

```
total_rows = len(data)

for column in data.columns:
    missing_count = data[column].isna().sum()
    missing_percentage = (missing_count / total_rows) * 100
    print(f"Column '{column}' Has {missing_count} missing values ({missing_percentage:.2f}%)")
```

```
Column 'survived' Has 0 missing values (0.00%)
Column 'name' Has 0 missing values (0.00%)
Column 'sex' Has 0 missing values (0.00%)
Column 'age' Has 49 missing values (9.82%)
```

DataFrame with shape (499, 4)

```
data.isna().sum()
```

| | 0 |
|---|---|
| survived | 0 |
| name | 0 |
| sex | 0 |
| age | 0 |

dtype: int64

```
[ ]  for column in data.columns:
        if data[column].dtype == 'object':
            data[column].fillna(data[column].mode()[0], inplace=True)
        else:
            data[column].fillna(data[column].median(), inplace=True)
```

```
[ ] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 499 entries, 0 to 499
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   survived  499 non-null    int64
 1   name      499 non-null    object
 2   sex       499 non-null    object
 3   age       499 non-null    float64
dtypes: float64(1), int64(1), object(2)
memory usage: 19.5+ KB
```

Findings:

- The Titanic dataset has a total of 4 columns and 499 rows.
- Inspection results show that only the age column has missing values of 49 entries, equivalent to 9.82% of the total data.

Handling:

- For object type columns (such as name and sex), the blank value is filled with the mode (the most frequent value).
- For columns with numeric types (such as age), empty values are filled with the median (middle value).

Final Result:

- After the imputation process, all columns no longer have missing values.
- This is confirmed by data.isna().sum() which returns zero for all columns.

# THANK YOU

LinkedIn :www.linkedin.com/in/azhar-frenotama-421811334