ibimbing

DIGITAL SKILL FAIR 32.0

# TITANIC DATA ANALYST

**By Arinda C**

COMPUTER SCIENCE

**2024**

# About Me

Hi, I'm Arinda, a Computer Science student from Binus University with a strong interest in data. This time, I would like to share the results of my Titanic data analysis, which I created during the Digital Skill Fair 32.0 program by Dibimbing.id.

I hope you find it useful.

# TABLE OF CONTENS

# Introduction

The Titanic tragedy serves as a somber reminder of the devastating consequences of inadequate safety measures and the impact of social class on survival. On its maiden voyage in 1912, the Titanic struck an iceberg, leading to the loss of over 1,500 lives.

Analysis of the event reveals that women, children, and first-class passengers had higher survival rates due to prioritized rescue efforts, while many second- and third-class passengers faced greater challenges in reaching safety. The disaster highlighted critical shortcomings in maritime safety regulations, ultimately leading to significant reforms in lifeboat requirements and emergency protocols.

# Project Phases

**01.**

**Passenger safety identification**

**02.**

**Develop Predictive models**

**03.**

**We aim to gain insights from the data**

# The Process of Data Modelling

### EDA

Analyzing and visualizing data to identify patterns.

### Feature Engineering

Selecting and creating relevant features for the model.

### Data Preprocessing

Cleaning and preparing data, including handling missing values.

### Model Evaluation

Test, evaluate the model and draw conclusions.

In [3]:
```
# Memuat data atau membaca data
df = pd.read_csv("train.csv")
df.shape # (rows, columns)
```

Out[3]: (891, 12)

In [4]:
```
# Melihat data 5 baris dari atas
df.head()
```

Out[4]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | NaN | S |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.1000 | C123 | S |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | 8.0500 | NaN | S |

In the Titanic dataset, there are 891 rows and 12 columns (variables).

# Data Preprocessing

### Melihat Missing Value

In [5]:
```
# Mengecek missing value
df.isna().sum()
```

Out[5]:

| | 0 |
|---|---|
| PassengerId | 0 |
| Survived | 0 |
| Pclass | 0 |
| Name | 0 |
| Sex | 0 |
| Age | 177 |
| SibSp | 0 |
| Parch | 0 |
| Ticket | 0 |
| Fare | 0 |
| Cabin | 687 |
| Embarked | 2 |

In [9]:
```
df.isna().sum()
```

Out[9]:

| | 0 |
|---|---|
| Survived | 0 |
| Pclass | 0 |
| Sex | 0 |
| Age | 0 |
| SibSp | 0 |
| Parch | 0 |
| Fare | 0 |
| Embarked | 0 |

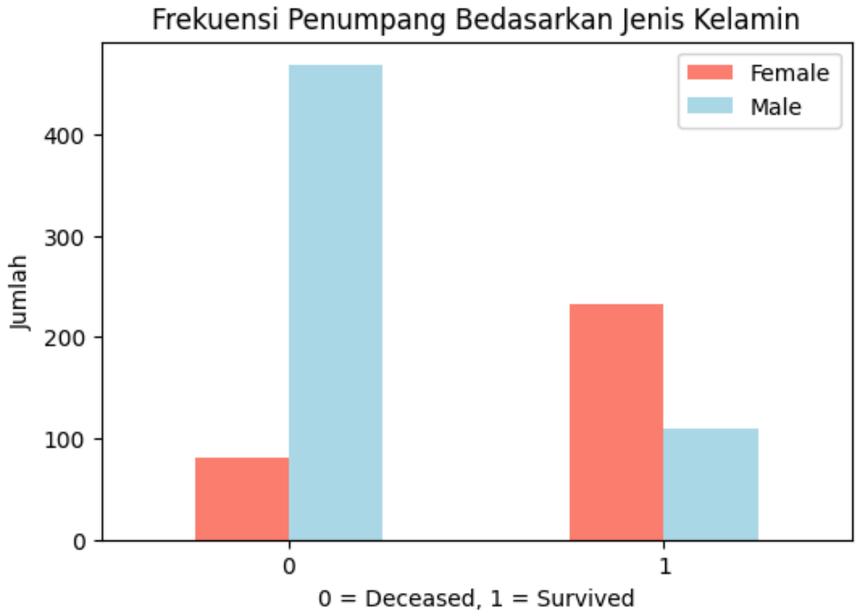
dtype: int64

- At this stage, data cleaning is performed by removing NaN values, filling "age" with the median, and "embarked" with the mode.
- Since the "embarked" variable is categorical, mode is the appropriate choice. Meanwhile, the median is chosen for the "age" variable (numerical data) as it is more robust to outliers compared to the mean.

```
In [13]: # Membandingkan kolom penumpang sel
         pd.crosstab(df.Survived, df.Sex)
```

Out[13]:

| Sex | female | male |
|---|---|---|
| **Survived** | | |
| **0** | 81 | 468 |
| **1** | 233 | 109 |




## Visualize

The number of male passengers who survived is 109, while the number of female passengers who survived is 233. The total number of deceased passengers, both male and female, is 549.

# Feature Engineering

In [15]:
```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
df = df.apply(le.fit_transform)
```

- Label Encoding: Converts categorical variables into numerical values.
- The code `df.apply(le.fit_transform)` transforms all categorical features in the DataFrame `df` to numeric, making them usable for machine learning models.

This step helps in preparing the data for model training.

Out[20]:

| | Pclass | Sex | Age | SibSp | Parch | Fare | Embarked |
|---|---|---|---|---|---|---|---|
| **0** | 2 | 1 | 28 | 1 | 0 | 18 | 2 |
| **1** | 0 | 0 | 51 | 1 | 0 | 207 | 0 |
| **2** | 2 | 0 | 34 | 0 | 0 | 41 | 2 |
| **3** | 0 | 0 | 47 | 1 | 0 | 189 | 2 |
| **4** | 2 | 1 | 47 | 0 | 0 | 43 | 2 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **886** | 1 | 1 | 35 | 0 | 0 | 85 | 2 |
| **887** | 0 | 0 | 24 | 0 | 0 | 153 | 2 |
| **888** | 2 | 0 | 36 | 1 | 2 | 131 | 2 |
| **889** | 0 | 1 | 34 | 0 | 0 | 153 | 0 |
| **890** | 2 | 1 | 42 | 0 | 0 | 30 | 1 |

891 rows × 7 columns

Out[21]:

| | Survived |
|---|---|
| **0** | 0 |
| **1** | 1 |
| **2** | 1 |
| **3** | 1 |
| **4** | 0 |
| **...** | ... |
| **886** | 0 |
| **887** | 1 |
| **888** | 0 |
| **889** | 1 |
| **890** | 0 |

891 rows × 1 columns

In [22]:

```
x_train = np.array(x[0:int(0.80*len(x))])
y_train = np.array(y[0:int(0.80*len(y))])
x_test = np.array(x[int(0.80*len(x)):])
y_test = np.array(y[int(0.80*len(y)):])
len(x_train), len(y_train), len(x_test), len(y_test)
```

Out[22]: (712, 712, 179, 179)

## Splitting Data Training and Testing

- Before modeling, the next step is to split the training data into 80% and the testing data into 20%.

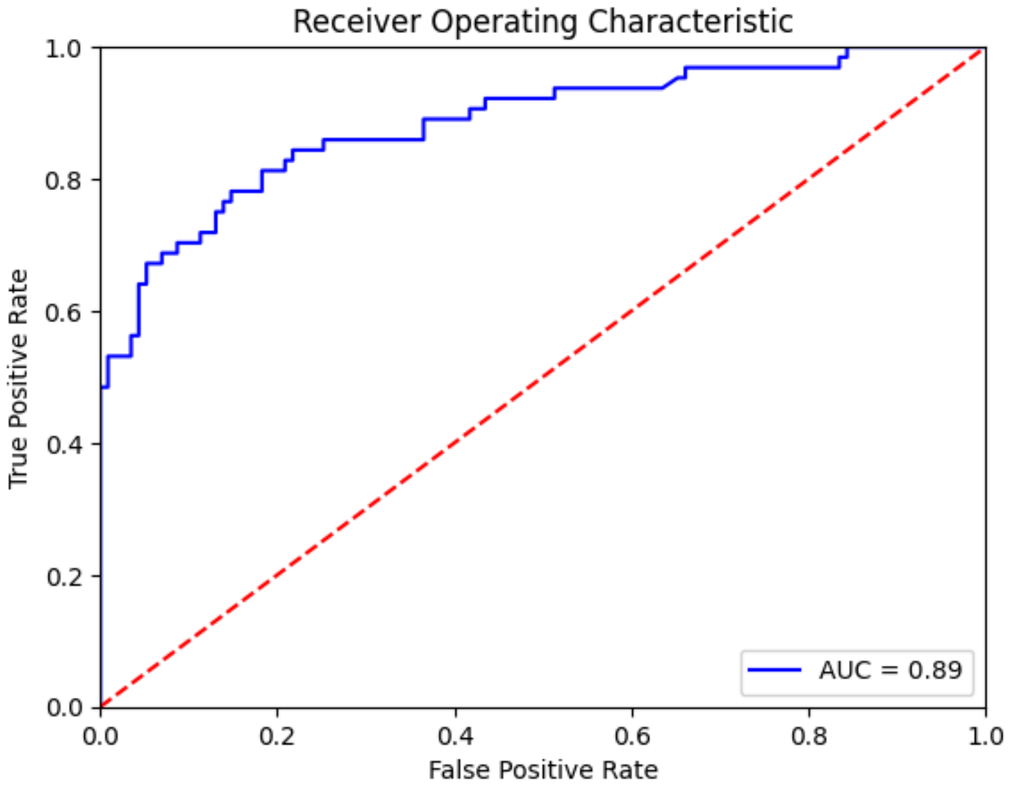
# Accuracy

In [26]:
```
from sklearn.metrics import accuracy_score
print("Logistic Regression is %f percent accurate" % (accuracy_score(LogReg_pred, y_test)*100))
print("KNN is %f percent accurate" % (accuracy_score(KNN_pred, y_test)*100))
print("Naive Bayes is %f percent accurate" % (accuracy_score(NB_pred, y_test)*100))
print("SVM is %f percent accurate" % (accuracy_score(SVM_pred, y_test)*100))
print("Decision Trees is %f percent accurate" % (accuracy_score(DT_pred, y_test)*100))
print("Random Forests is %f percent accurate" % (accuracy_score(RF_pred, y_test)*100))
```

```
Logistic Regression is 82.681564 percent accurate
KNN is 70.391061 percent accurate
Naive Bayes is 81.005587 percent accurate
SVM is 71.508380 percent accurate
Decision Trees is 75.418994 percent accurate
Random Forests is 81.005587 percent accurate
```

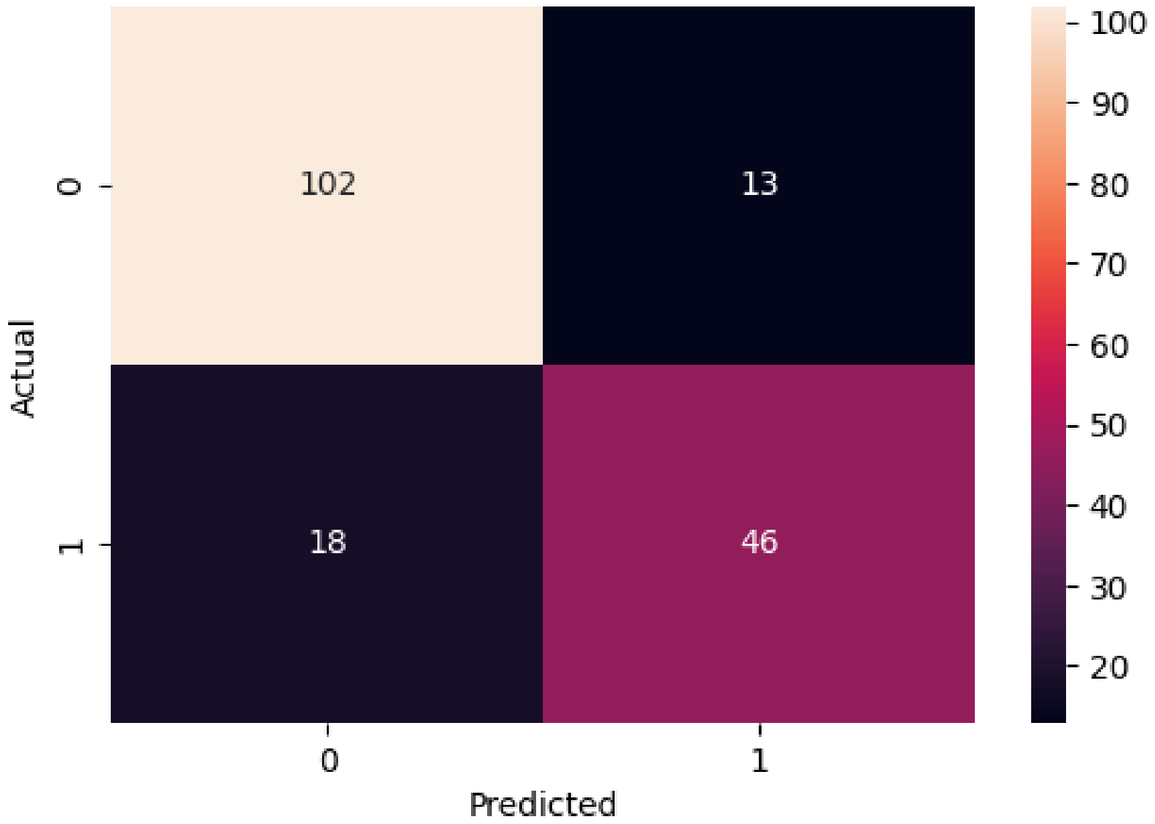
- The models used for evaluation include Logistic Regression, KNN, Naive Bayes, SVM, Decision Trees, and Random Forests.
- Among these six models, the one with the highest accuracy is Logistic Regression, with an accuracy of 82.68%.

# Model Evaluation




According to Metx (1978), the interpretation of the Area Under Curve (AUC) values is classified into five different categories:

- an accuracy level between 0.50 – 0.60 is considered very weak, 0.60 – 0.70 is weak, 0.70 – 0.80 is moderate, 0.80 – 0.90 is high, and 0.90 – 1.00 is very high.
- In the AUC graph results, an AUC of 0.89 was obtained, indicating that the accuracy level using the logistic regression model is very high.




## Confussion Matrix

- 102 passengers were correctly predicted to have died, and 46 passengers were correctly predicted to have survived. The total correct predictions amount to 102 + 46 = 148 passengers out of a total of 179 passengers.

# Classification Report

In [31]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, LogReg_pred))
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.85 | 0.89 | 0.87 | 115 |
| 1 | 0.78 | 0.72 | 0.75 | 64 |
| accuracy | | | 0.83 | 179 |
| macro avg | 0.81 | 0.80 | 0.81 | 179 |
| weighted avg | 0.82 | 0.83 | 0.83 | 179 |

Conclusion:

- The model is better at identifying deceased passengers (89%) compared to survivors (72%).
- This indicates that the model tends to be "conservative" in predicting survival, as it is less likely to predict that someone will survive.
- There may be important factors not yet included in the model that could improve the accuracy of passenger survival predictions.

# Conclusion

To improve the Titanic passenger safety prediction model, several steps can be taken.

- First, add new features like family size.
- Second, normalize numerical data to enhance model performance.
- Third, optimize model settings through hyperparameter tuning.
- Fourth, address data imbalance to achieve more accurate predictions.
- Fifth, use k-fold cross-validation techniques to ensure more stable results.
- Sixth, pay attention to outliers in the Fare variable to make the model more robust.
- Lastly, consider exploring other models like Gradient Boosting or XGBoost for better outcomes.

# Thank You

**Any question? Please contact me at:**


https://www.linkedin.com/in/arindacintalestari/

arindaclstr@gmail.com