**LAPORAN**

**PORTFOLIO ASSOCIATE DATA SCIENTIST**

**SUPERVISED LEARNING**

Disusun Oleh:

Drs. Herwin, M.Pd.

**FAKULTAS ILMU KEOLAHRAGAAN**

**UNIVERSITAS NEGERI YOGYAKARTA**

**18 DESEMBER 2021**

* Dataset ini merupakan informasi terkait pelanggan apakah ingin membeli atau tidak terhadap promo barang yang ditawarkan. Diambil beberapa informasi untuk menentukan pelanggan tersebut akan membeli atau tidak.  
  X1 : Gender  
  x2 : Age  
  X3 : Estimated Salary  
  Y : Purchased (0: Tidak Membeli, 1: Membeli)

**Data Understanding**

**Load Library**

**#library data manipulation**

* import pandas as pd
* import numpy as np

**#library data visualisasi**

* import matplotlib.pyplot as plt
* import seaborn as sns

**#library data modelling**

* from sklearn.preprocessing import StandardScaler
* from sklearn.model\_selection import train\_test\_split
* from sklearn.svm import SVC
* from sklearn.linear\_model import LogisticRegression

**#library data evaluasi**

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

**#library export modelling**

import pickle

* import pandas as pd
* df = pd.read\_csv('Dataset12\_Social\_Advertising.csv')
* df.head(5)

**Load Dataset**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **User ID** | **Gender** | **Age** | **EstimatedSalary** | **Purchased** |
| **0** | 15624510 | Male | 19 | 19000 | 0 |
| **1** | 15810944 | Male | 35 | 20000 | 0 |
| **2** | 15668575 | Female | 26 | 43000 | 0 |
| **3** | 15603246 | Female | 27 | 57000 | 0 |
| **4** | 15804002 | Male | 19 | 76000 | 0 |

**Validation Dataset**

#dimension of dataset

* df.shape

**(400, 5)**

#deskriptif of dataset

* df.describe().T

**Validation Dataset**



**Validation Dataset**

#information of dataset

* df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 400 entries, 0 to 399 Data columns (total 5 columns): # Column Non-Null Count Dtype --- ------ -------------- ----- 0 User ID 400 non-null int64

1 Gender 400 non-null object

2 Age 400 non-null int64

3 EstimatedSalary 400 non-null int64

4 Purchased 400 non-null int64

dtypes: int64(4), object(1) memory usage: 15.8+ KB

**Objective Dataset**

* df.head(5)



**#drop user ID**

* df = df.drop(['User ID'], axis=1)
* df.head(5)

**Objective Dataset**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Gender** | **Age** | **EstimatedSalary** | **Purchased** |
| **0** | Male | 19 | 19000 | 0 |
| **1** | Male | 35 | 20000 | 0 |
| **2** | Female | 26 | 43000 | 0 |
| **3** | Female | 27 | 57000 | 0 |
| **4** | Male | 19 | 76000 | 0 |

**Cleaning of Dataset**

* df.isnull().sum()

Gender 0

Age 0 EstimatedSalary 0 Purchased 0

dtype: int64

#option 1

# df = df.dropna()

# df.isnull().sum()

#option 2

* df['Gender'] = df['Gender'].fillna(df['Gender'].mode()[0])
* df['Age'] = df['Age'].fillna(int(df['Age'].mean()))
* df.isnull().sum()

**Objective Dataset**

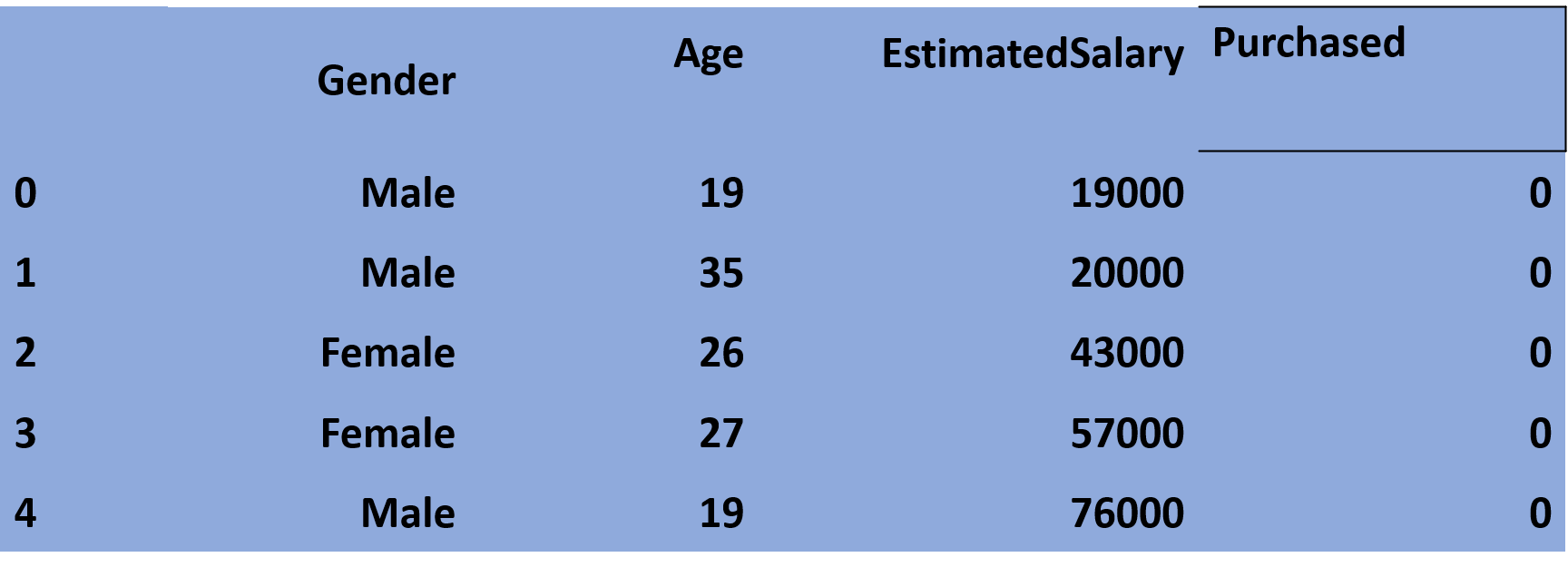
Gender 0

Age 0

EstimatedSalary 0

Purchased 0

dtype: int64

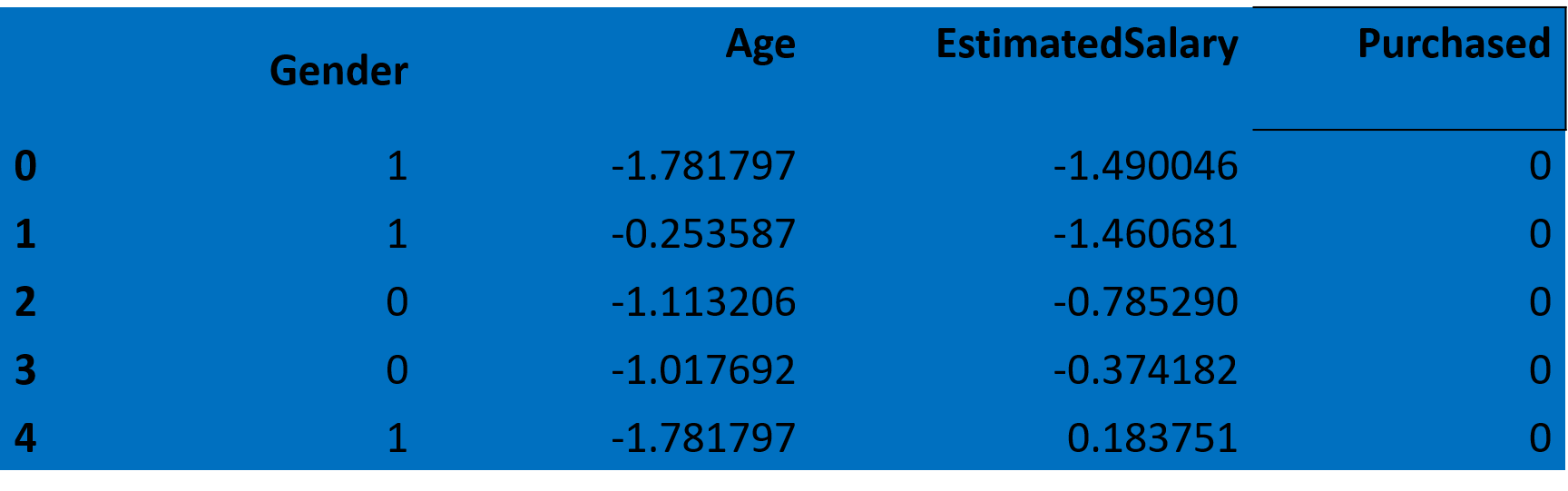


* df.head(5)

**Kontruksi Prediktor Dataset**

**Kontruksi Prediktor Dataset**

* df['Gender'] = df['Gender'].map({'Male': 1, 'Female':0})
* scaler = StandardScaler()
* df.iloc[:,1:3] = scaler.fit\_transform(df.iloc[:,1:3])
* df.head(5)

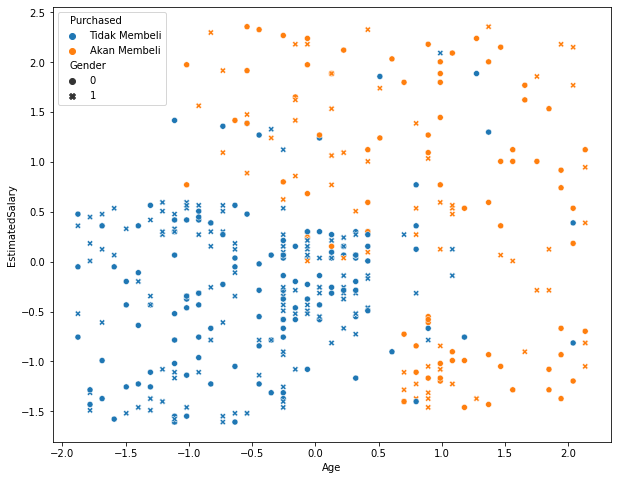


* df['Purchased'].unique()
* array([0, 1])
* df['Purchased'] = df['Purchased'].map({0:'Tidak Membeli', 1:'Akan Membeli'})
* df.head(5)

**Labelling of Dataset**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Gender** | **Age** | **EstimatedSalary** | **Purchased** |
| **0** | **1** | **-1.781797** | **-1.490046** | **Tidak Membeli** |
| **1** | **1** | **-0.253587** | **-1.460681** | **Tidak Membeli** |
| **2** | **0** | **-1.113206** | **-0.785290** | **Tidak Membeli** |
| **3** | **0** | **-1.017692** | **-0.374182** | **Tidak Membeli** |
| **4** | **1** | **-1.781797** | **0.183751** | **Tidak Membeli** |

* fig, axes = plt.subplots(figsize=(10,8))
* sns.scatterplot(data=df, x='Age', y='EstimatedSalary', hue='Purchased', style='Gender')
* plt.show()



**Visualization of Dataset**

**Model SVM**

**Modelling of Dataset**

* X = df[['Gender','Age','EstimatedSalary']]
* y = df['Purchased']
* x\_train, x\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size=0.2, random\_state=14)
* model\_svm = SVC(kernel='rbf')
* model\_svm.fit(x\_train, y\_train)
* with open('model\_svm.pkl', 'wb') as file:

pickle.dump(model\_svm, file)

**Model LogisticRegression**

* X = df[['Gender','Age','EstimatedSalary']]
* y = df['Purchased']
* x\_train, x\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size=0.2, random\_state=14)
* model\_lr = LogisticRegression()
* model\_lr.fit(x\_train, y\_train)
* with open('model\_lr.pkl', 'wb') as file:

pickle.dump(model\_lr, file)

* y\_predict = model\_svm.predict(x\_test)
* print(f"Classification Report \n {classification\_report(y\_test, y\_predict)}")

**Evaluation of Modelling**

**Classification Report**

**precision recall f1-score support**

**Akan Membeli 0.81 0.91 0.85 32**

**Tidak Membeli 0.93 0.85 0.89 48**

**accuracy 0.88 80**

**macro avg 0.87 0.88 0.87 80**

**weighted avg 0.88 0.88 0.88 80**

* y\_predict = model\_lr.predict(x\_test)
* print(f"Classification Report \n {classification\_report(y\_test, y\_predict)}")

**Classification Report**

**precision recall f1-score support**

**Akan Membeli 0.89 0.75 0.81 32**

**Tidak Membeli 0.85 0.94 0.89 48**

**accuracy 0.86 80**

**macro avg 0.87 0.84 0.85 80**

**weighted avg 0.86 0.86 0.86 80**

Visualisasi *Confusion Matrix*

**Evaluation of Modelling**

* print(f"Confusion Matrix Report \n {confusion\_matrix(y\_test, y\_predict)}")

**Confusion Matrix Report**

**[[ 29 3]**

**[ 7 41]]**

Visualisasi *Confusion Matrix*

* import seaborn as sns
* import matplotlib.pyplot as plt
* f, ax = plt.subplots(figsize=(8,5))
* sns.heatmap(confusion\_matrix(y\_test, y\_predict), annot=True, fmt=".0f", ax=ax)
* plt.xlabel("y\_head")
* plt.ylabel("y\_true")
* plt.show()

