

PORTFOLIO ASSOCIATE DATA SCIENTIST

SUPERVISED LEARNING

Disusun Oleh:

Drs. Herwin, M.Pd.

**FAKULTAS ILMU KEOLAHRAGAAN
UNIVERSITAS NEGERI YOGYAKARTA**

18 DESEMBER 2021

Data Understanding

- 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)

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

Load Library

#library data evaluasi

- from sklearn.metrics import classification_report
- from sklearn.metrics import confusion_matrix

#library export modelling

- import pickle

Load Dataset

- import pandas as pd
- df = pd.read_csv('Dataset12_Social_Advertising.csv')
- df.head(5)

	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)

Validation Dataset

#deskriptif of dataset

- df.describe().T

	count	mean	std	min	25%	50%	75%	max
User ID	400.0	1.569154e+07	71658.321581	15566689.0	15626763.75	15694341.5	15750363.0	15815236.0
Age	400.0	3.765500e+01	10.482877	18.0	29.75	37.0	46.0	60.0
Estimated Salary	400.0	6.974250e+04	34096.960282	15000.0	43000.00	70000.0	88000.0	150000.0
Purchased	400.0	3.575000e-01	0.479864	0.0	0.00	0.0	1.0	1.0

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)`

	User ID	Gender	Age	Estimated Salary	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

Objective Dataset

#drop user ID

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

	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
```

Objective Dataset

#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

Konstruksi Prediktor Dataset

- `df.head(5)`

	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

Konstruksi 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)`

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

Labelling of Dataset

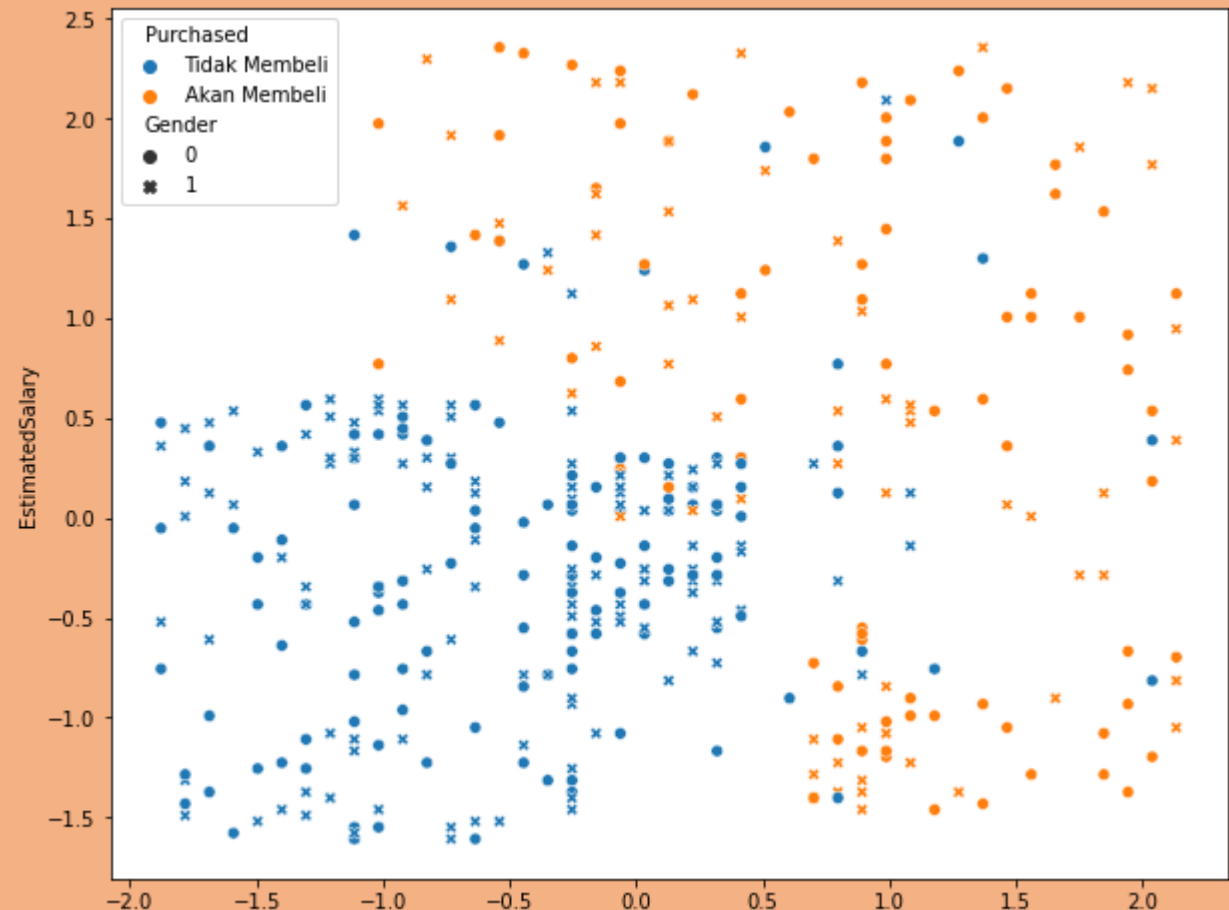
- `df['Purchased'].unique()`
- `array([0, 1])`
- `df['Purchased'] = df['Purchased'].map({0:'Tidak Membeli', 1:'Akan Membeli'})`
- `df.head(5)`

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

Visualization of Dataset

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



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)`

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_lr = LogisticRegression()`
- `model_lr.fit(x_train, y_train)`
- `with open('model_lr.pkl', 'wb') as file:`
`pickle.dump(model_lr, file)`

Evaluation of Modelling

- `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

Evaluation of Modelling

- `y_predict = model_svm.predict(x_test)`
- `print(f"Classification Report \n {classification_report(y_test, y_predict)})"`

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

Evaluation of Modelling

- `print(f"Confusion Matrix Report \n {confusion_matrix(y_test, y_predict)}")`

```
Confusion Matrix Report
```

```
[[ 29  3]
```

```
[ 7 41]]
```

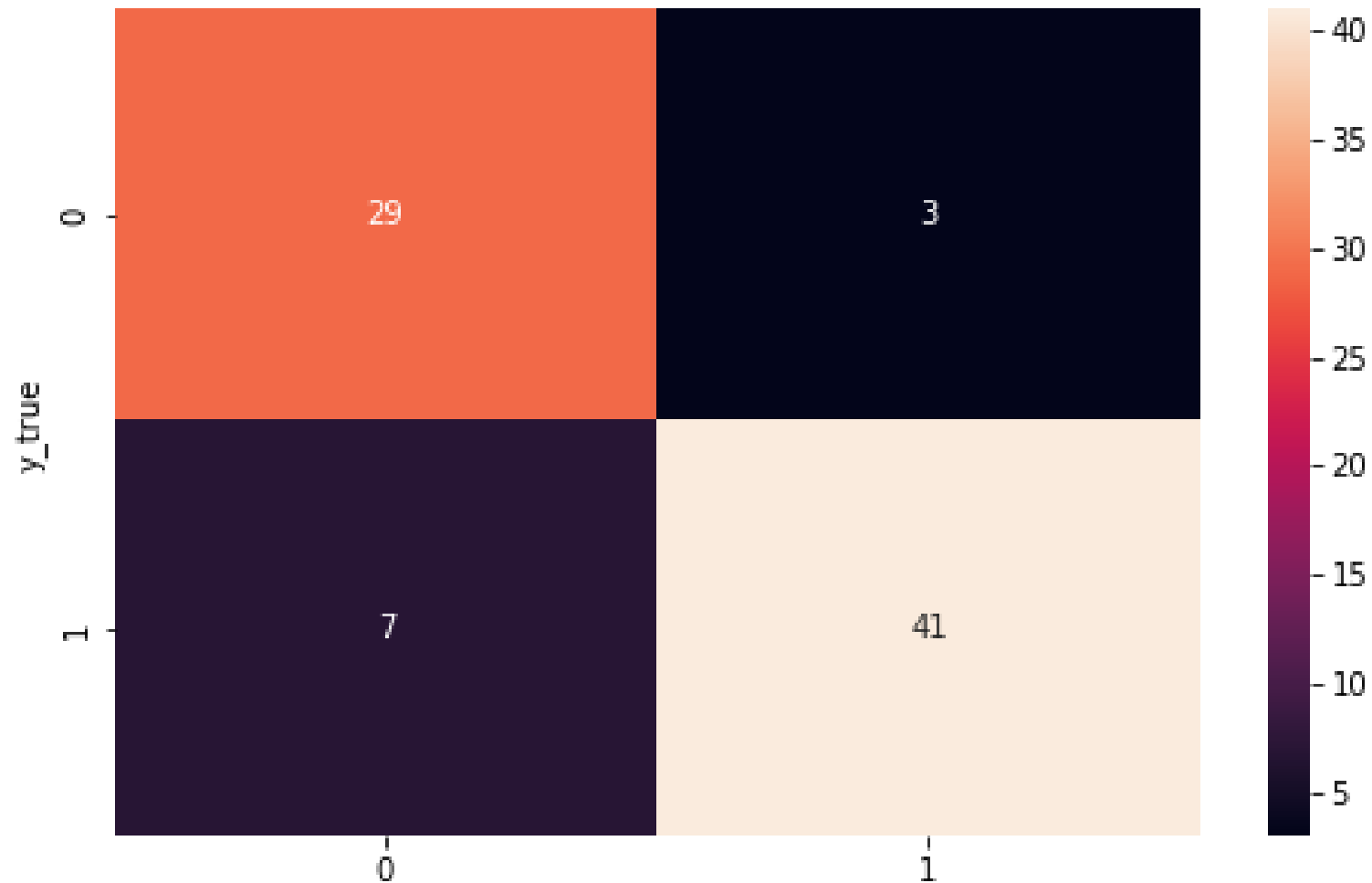
Evaluation of Modelling

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()`

Evaluation of Modelling

Visualisasi *Confusion Matrix*



TERIMA KASIH