Nama : Deni Purwanto

NPM : 41155050210017

Kelas : INF A

Mata Kuliah : Machine Learning

1.0. Lakukan praktek dari https://youtu.be/lcjq7-2zMSA?si=f4jWJR6lY8y0BZKl dan buat screen shot hasil run dengan nama anda pada hasil run tersebut. Praktek tersebut yaitu:

1.1. Sample dataset

```
[2]: import pandas as pd

     print("Nama : Deni Purwanto")

     pizza = {'diameter': [6, 8, 10, 14, 18],
              'harga': [7, 9, 13, 17.5, 18]}

     pizza_df = pd.DataFrame(pizza)
     pizza_df
```

```
Nama : Deni Purwanto
```

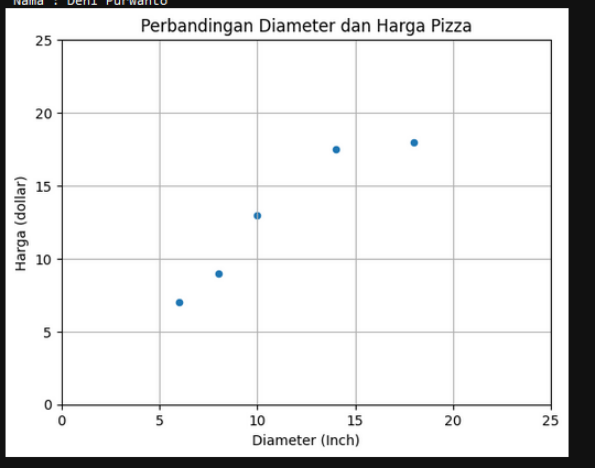
[2]:

	diameter	harga
0	6	7.0
1	8	9.0
2	10	13.0
3	14	17.5
4	18	18.0

1.2. Visualisasi dataset

```
[3]: import matplotlib.pyplot as plt

pizza_df.plot(kind='scatter', x='diameter', y='harga')

print("Nama : Deni Purwanto")

plt.title('Perbandingan Diameter dan Harga Pizza')
plt.xlabel('Diameter (Inch)')
plt.ylabel('Harga (dollar)')
plt.xlim(0, 25)
plt.ylim(0, 25)
plt.grid(True)
plt.show()
```

```
Nama : Deni Purwanto
```



1.3. Transformasi dataset

```
[5]: import numpy as np

print("Nama : Deni Purwanto")

X = np.array(pizza_df['diameter'])
y = np.array(pizza_df['harga'])

print(f'X: {X}')
print(f'y: {y}')
```

```
Nama : Deni Purwanto
X: [ 6  8 10 14 18]
y: [ 7.   9.  13.  17.5 18. ]
```

```
[6]: X = X.reshape(-1, 1)
X.shape
```

```
[6]: (5, 1)
```

```
[7]: X
```

```
[7]: array([[ 6],
       [ 8],
       [10],
       [14],
       [18]])
```

1.4. Training Simple Linear Regression Model

```
[9]: from sklearn.linear_model import LinearRegression

     print("Nama : Deni Purwanto")

     model = LinearRegression()
     model.fit(X, y)
```

```
Nama : Deni Purwanto
```

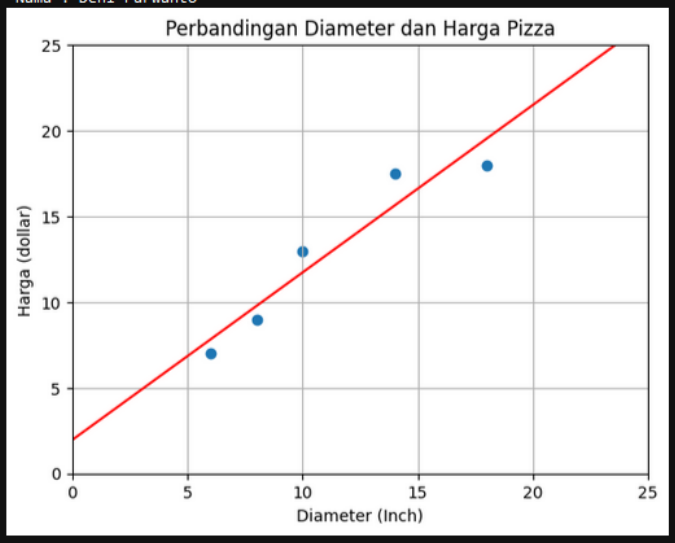
```
[9]: LinearRegression
     LinearRegression()
```

1.5. Visualisasi Simple Linear Regression Model | Penjelasan persamaan garis linear

```
[10]: X_vis = np.array([0, 25]).reshape(-1, 1)
      y_vis = model.predict(X_vis)
```

```
[11]: plt.scatter(X, y)
      plt.plot(X_vis, y_vis, '-r')

      print("Nama : Deni Purwanto")

      plt.title('Perbandingan Diameter dan Harga Pizza')
      plt.xlabel('Diameter (Inch)')
      plt.ylabel('Harga (dollar)')
      plt.xlim(0, 25)
      plt.ylim(0, 25)
      plt.grid(True)
      plt.show()
```

```
Nama : Deni Purwanto
```




```
[12]: print(f'intercept: {model.intercept_}')
      print(f'slope: {model.coef_}')
```

```
intercept: 1.965517241379315
slope: [0.9762931]
```

1.6. Kalkulasi nilai slope

```
[13]: print("Nama : Deni Purwanto")

      print(f'X:\n{X}\n')
      print(f'X:flatten:{X.flatten()}\n')
      print(f'y:{y}')

      Nama : Deni Purwanto
      X:
      [[ 6]
       [ 8]
       [10]
       [14]
       [18]]

      X:flatten:[ 6  8 10 14 18]

      y:[ 7.   9.  13.  17.5 18. ]

[14]: variance_x = np.var(X.flatten(), ddof=1)
      print(f'variance: {variance_x}')

      variance: 23.2

[15]: np.cov(X.flatten(), y)

[15]: array([[23.2 , 22.65],
             [22.65, 24.3 ]])

[18]: convariance_xy = np.cov(X.flatten(), y)[0][1]
      print(f'convariance: {convariance_xy}')

      convariance: 22.650000000000002

[19]: slope = convariance_xy / variance_x
      print(f'slope: {slope}')

      slope: 0.976293103448276
```

1.7. Kalkukasi nilai intercept

```
[20]: intercept = np.mean(y) - slope * np.mean(X)

      print("Nama : Deni Purwanto")

      print(f'intercept: {intercept}')

      Nama : Deni Purwanto
      intercept: 1.9655172413793096

[22]: print(f'intercept: {model.intercept_}')
      print(f'slope: {model.coef_}')

      intercept: 1.965517241379315
      slope: [0.9762931]
```

1.8. Prediksi harga pizza dengan Simple Linear Regression Model

```
[23]: print("Nama : Deni Purwanto")
      diameter_pizza = np.array([12, 20, 23]).reshape(-1, 1)
      diameter_pizza

      Nama : Deni Purwanto
[23]: array([[12],
             [20],
             [23]])

[24]: prediksi_harga = model.predict(diameter_pizza)
      prediksi_harga

[24]: array([13.68103448, 21.49137931, 24.42025862])

[25]: for dmtr, hrg in zip(diameter_pizza, prediksi_harga):
          print(f'Diameter: {dmtr} prediksi harga: {hrg}')

      Diameter: [12] prediksi harga: 13.681034482758621
      Diameter: [20] prediksi harga: 21.491379310344826
      Diameter: [23] prediksi harga: 24.42025862068965
```

1.9. Evaluasi model dengan Coefficient of Determination | R Squared

```
[32]: X_train = np.array([6, 8, 10, 14, 18]).reshape(-1, 1)
      y_train = np.array([7, 9, 13, 17.5, 18])

      X_test = np.array([8, 9, 11, 16, 12]).reshape(-1, 1)
      y_test = np.array([11, 8.5, 15, 18, 11])

[33]: model = LinearRegression()
      model.fit(X_train, y_train)

[33]: LinearRegression
      LinearRegression()

[35]: from sklearn.metrics import r2_score

      print("Nama : Deni Purwanto")

      y_pred = model.predict(X_test)
      r_squared = r2_score(y_test, y_pred)
      print(f'R-squared: {r_squared}')

      Nama : Deni Purwanto
      R-squared: 0.6620052929422553
```

1.10. Kalkulasi nilai R Squared | Coefficient of Determination

```
[36]: ss_res = sum([(y_i - model.predict(x_i.reshape(-1, 1))[0])**2
                  for x_i, y_i in zip(X_test, y_test)])

      print("Nama : Deni Purwanto")
      print(f'ss_res:{ss_res}')

      Nama : Deni Purwanto
      ss_res:19.1980993608799

[37]: mean_y = np.mean(y_test)
      ss_tot = sum([(y_i - mean_y)**2 for y_i in y_test])

      print(f'ss_tot: {ss_tot}')

      ss_tot: 56.8
```

2.0. Lakukan praktek dari https://youtu.be/nWJUJenAyB8?si=BQDzWwrMnr8jtzpV dan buat screen shot hasil run dengan nama anda pada hasil run tersebut. Praktek tersebut yaitu:

2.1. Persiapan sample dataset

```
[6]: import pandas as pd

     print("Nama : Deni Purwanto")
     pizza = {'diameter': [6, 8, 10, 14, 18],
             'n_topping': [2, 1, 0, 2, 0],
             'harga': [7, 9, 13, 17.5, 18]}

     train_pizza_df = pd.DataFrame(pizza)
     train_pizza_df
```

```
Nama : Deni Purwanto
```

[6]:

	diameter	n_topping	harga
0	6	2	7.0
1	8	1	9.0
2	10	0	13.0
3	14	2	17.5
4	18	0	18.0

```
[7]: print("Nama : Deni Purwanto")
     pizza = {'diameter': [8, 9, 11, 16, 12],
             'n_topping': [2, 0, 2, 2, 0],
             'harga': [11, 8.5, 15, 18, 11]}

     train_pizza_df = pd.DataFrame(pizza)
     train_pizza_df
```

```
Nama : Deni Purwanto
```

[7]:

	diameter	n_topping	harga
0	8	2	11.0
1	9	0	8.5
2	11	2	15.0
3	16	2	18.0
4	12	0	11.0

2.2. Preprocessing dataset

```
[8]: import numpy as np

     print("Nama : Deni Purwanto")
     X_train = np.array(train_pizza_df[['diameter', 'n_topping']])
     y_train = np.array(train_pizza_df['harga'])

     print(f'X_train:\n{X_train}\n')
     print(f'y_train:\n{y_train}\n')
```

```
Nama : Deni Purwanto
X_train:
[[ 8  2]
 [ 9  0]
 [11  2]
 [16  2]
 [12  0]]

y_train:
[11.   8.5 15.  18.  11. ]
```

```
[13]: print("Nama : Deni Purwanto")
      X_test = np.array(test_pizza_df[['diameter', 'n_topping']])
      y_test = np.array(test_pizza_df['harga'])

      print(f'X_test:\n{X_test}\n')
      print(f'y_test:\n{y_test}\n')
```

```
Nama : Deni Purwanto
X_test:
[[ 8  2]
 [ 9  0]
 [11  2]
 [16  2]
 [12  0]]

y_test:
[11.   8.5 15.  18.  11. ]
```

2.3. Pengenalan Multiple Linear Regression | Apa itu Multiple Linear Regression?

```
[25]: from sklearn.linear_model import LinearRegression
      from sklearn.metrics import r2_score

      print("Nama : Deni Purwanto")
      model = LinearRegression()
      model.fit(X_train, y_train)
      y_pred = model.predict(X_test)

      print(f'r_squared: {r2_score(y_test, y_pred)}')
```

```
Nama : Deni Purwanto
r_squared: 0.9782495420956231
```

2.4. Pengenalan Polynomial Regression | Apa itu Polynomial Regression?

```
[26]: print("Nama : Deni Purwanto")
      X_train = np.array(train_pizza_df['diameter']).reshape(-1, 1)
      y_train = np.array(train_pizza_df['harga'])

      print(f'X_train:\n{X_train}\n')
      print(f'y_train:\n{y_train}')
```

```
Nama : Deni Purwanto
X_train:
[[ 8]
 [ 9]
 [11]
 [16]
 [12]]

y_train:
[11.   8.5 15.  18.  11. ]
```

2.5. Quadratic Polynomial Regression

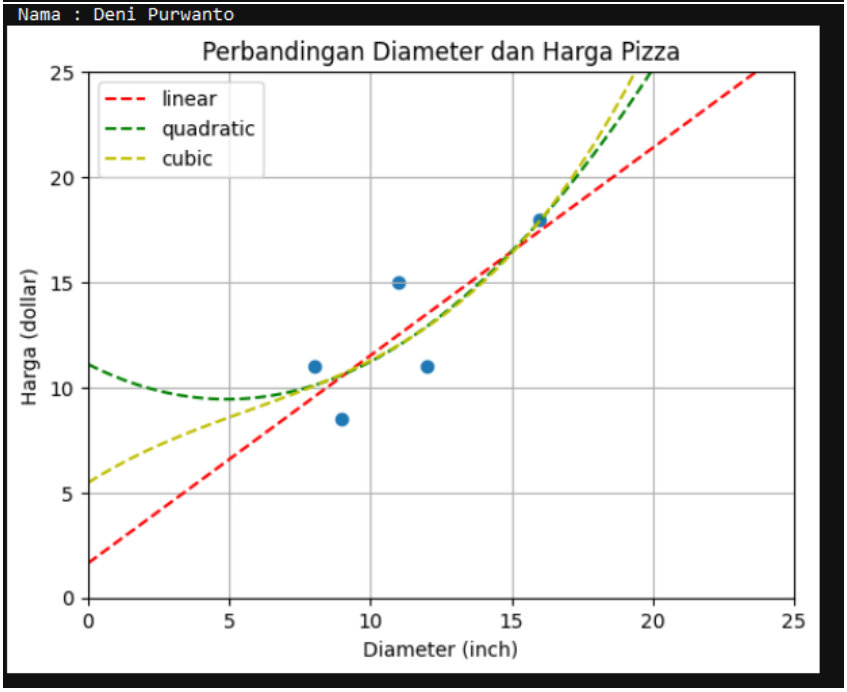
```
[27]: from sklearn.preprocessing import PolynomialFeatures

      print("Nama : Deni Purwanto")
      quadratic_feature = PolynomialFeatures(degree=2)
      X_train_quadratic = quadratic_feature.fit_transform(X_train)

      print(f'X_train_quadratic:\n{X_train_quadratic}\n')
```

```
Nama : Deni Purwanto
X_train_quadratic:
[[  1.   8.  64.]
 [  1.   9.  81.]
 [  1.  11. 121.]
 [  1.  16. 256.]
 [  1.  12. 144.]]
```

2.6. Linear Regression vs Quadratic Polynomial Regression vs Cubic Polynomial Regression

```
[37]: print("Nama : Deni Purwanto")
      # Training Set
      plt.scatter(X_train, y_train)
      # Linear
      model = LinearRegression()
      model.fit(X_train, y_train)
      X_vis = np.linspace(0, 25, 100).reshape(-1, 1)
      y_vis = model.predict(X_vis)
      plt.plot(X_vis, y_vis, '--r', label='linear')
      # Quadratic
      quadratic_feature = PolynomialFeatures (degree=2)
      X_train_quadratic = quadratic_feature.fit_transform(X_train)
      model = LinearRegression()
      model.fit(X_train_quadratic, y_train)
      X_vis_quadratic = quadratic_feature.transform(X_vis)
      y_vis = model.predict(X_vis_quadratic)
      plt.plot(X_vis, y_vis, '--g', label='quadratic')
      # Cubic
      PolynomialFeatures(degree=3)
      cubic_feature = PolynomialFeatures (degree=3)
      X_train_cubic = cubic_feature.fit_transform(X_train)
      model = LinearRegression()
      model.fit(X_train_cubic, y_train)
      X_vis_cubic = cubic_feature.transform(X_vis)
      y_vis = model.predict(X_vis_cubic)
      plt.plot(X_vis, y_vis, '--y', label='cubic')
      plt.title('Perbandingan Diameter dan Harga Pizza')
      plt.xlabel('Diameter (inch)')
      plt.ylabel('Harga (dollar)')
      plt.legend()
      plt.xlim(0, 25)
      plt.ylim(0, 25)
      plt.grid(True)
      plt.show()
```

```
Nama : Deni Purwanto
```



3.0. Lakukan praktek dari https://youtu.be/oe7DW4rSH1o?si=H-PZJ9rs9-Kab-Ln dan buat screen shot hasil run dengan nama anda pada hasil run tersebut. Praktek tersebut yaitu:

3.1. Formula dasar pembentuk Logistic Regression | Fungsi Sigmoid

3.2. Persiapan dataset | SMS Spam Collection Dataset

```
[47]: import pandas as pd

      print("Nama : Deni Purwanto")

      # Adjusting the file path to point to your dataset location
      df = pd.read_csv(r'C:\Users\ASUS\Desktop\Jupyter Projects\datasets\SMSSpamCollection',
                       sep='\t',
                       header=None,
                       names=['label', 'sms'])

      # Displaying the first few rows
      df.head()
```

```
Nama : Deni Purwanto
```

[47]:

	label	sms
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

```
[48]: df['label'].value_counts()
```

```
[48]: label
      ham     4825
      spam     747
      Name: count, dtype: int64
```

3.3. Pembagian training dan testing set

```
[50]: from sklearn.preprocessing import LabelBinarizer

      print("Nama : Deni Purwanto")
      X = df['sms'].values
      y = df['label'].values

      lb = LabelBinarizer()
      y = lb.fit_transform(y).ravel()
      lb.classes_
```

```
Nama : Deni Purwanto
[50]: array(['ham', 'spam'], dtype='<U4')
```

```
[51]: from sklearn.model_selection import train_test_split

      print("Nama : Deni Purwanto")
      X_train, X_test, y_train, y_test = train_test_split(X,
                                                          y,
                                                          test_size = 0.25,
                                                          random_state = 0)
      print(X_train, '\n')
      print(y_train)
```

```
Nama : Deni Purwanto
['Its going good...no problem..but still need little experience to understand american customer voice...'
 'U have a secret admirer. REVEAL who thinks U R So special. Call 09065174042. To opt out Reply REVEAL STOP. 1.50 per msg recd. Cust care 07821230901'
 'Ok...' ...
 "For ur chance to win a £250 cash every wk TXT: ACTION to 80608. T's&C's www.movietrivia.tv custcare 08712405022, 1x150p/wk"
 'R U &SAM P IN EACHOTHER. IF WE MEET WE CAN GO 2 MY HOUSE'
 'Mm feeling sleepy. today itself i shall get that dear']

[0 1 0 ... 1 0 0]
```

3.4. Feature extraction dengan TF-IDF

```
[54]: from sklearn.feature_extraction.text import TfidfVectorizer

      print("Nama : Deni Purwanto")
      vectorizer = TfidfVectorizer(stop_words='english')

      X_train_tfidf = vectorizer.fit_transform(X_train)
      X_test_tfidf = vectorizer.transform(X_test)

      print(X_train_tfidf)
```

```
Nama : Deni Purwanto
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 32656 stored elements and shape (4179, 7287)>
  Coords        Values
  (0, 2997)     0.23173982975834367
  (0, 3007)     0.21421364306658514
  (0, 5123)     0.308974289326673
  (0, 4453)     0.2297719954323795
  (0, 3926)     0.3126721340000456
  (0, 2554)     0.3825278811525034
  (0, 6739)     0.3546359942830148
  (0, 900)      0.4114867709157148
  (0, 2006)     0.2898082580285881
  (0, 6903)     0.3591386422223876
  (1, 5642)     0.24344998442301355
  (1, 799)      0.25048918791028574
  (1, 5441)     0.5009783758205715
  (1, 6472)     0.24039776602646504
  (1, 6013)     0.20089911182610476
  (1, 216)      0.28902673040368515
  (1, 4677)     0.24039776602646504
  (1, 5394)     0.16464655071448758
  (1, 6131)     0.16142609035094446
  (1, 532)      0.20186022353306565
  (1, 4358)     0.17341410292348694
  (1, 5301)     0.2711077935907125
  (1, 2003)     0.2711077935907125
  (1, 1548)     0.18167737976542422
  (1, 36)       0.28902673040368515
  :     :
```

3.5. Binary Classification dengan Logistic Regression

```
[55]: from sklearn.linear_model import LogisticRegression

      print("Nama : Deni Purwanto")

      model = LogisticRegression()
      model.fit(X_train_tfidf, y_train)
      y_pred = model.predict(X_test_tfidf)

      for pred, sms in zip(y_pred[:5], X_test[:5]):
          print(f'PRED: {pred} - SMS: {sms}\n')

      Nama : Deni Purwanto
      PRED: 0 - SMS: Storming msg: Wen u lift d phne, u say "HELLO" Do u knw wt is d real meaning of HELLO?? . . . It's d name of a girl..! . . . Yes.. And u knw
      who is dat girl?? "Margaret Hello" She is d girlfrnd f Grahmbell who invnted telphone... . . . . Moral:One can 4get d name of a person, bt not his girlfrnd
      ... G o o d n i g h t . . .@

      PRED: 0 - SMS: <Forwarded from 448712404000>Please CALL 08712404000 immediately as there is an urgent message waiting for you.

      PRED: 0 - SMS: And also I've sorta blown him off a couple times recently so id rather not text him out of the blue looking for weed

      PRED: 0 - SMS: Sir Goodmorning, Once free call me.

      PRED: 0 - SMS: All will come alive.better correct any good looking figure there itself..
```

3.6. Evaluation Metrics pada Binary Classification Task

3.7. Pengenalan Confusion Matrix

```
[56]: from sklearn.metrics import confusion_matrix

      print("Nama : Deni Purwanto")
      matrix = confusion_matrix(y_test, y_pred)
      matrix
```

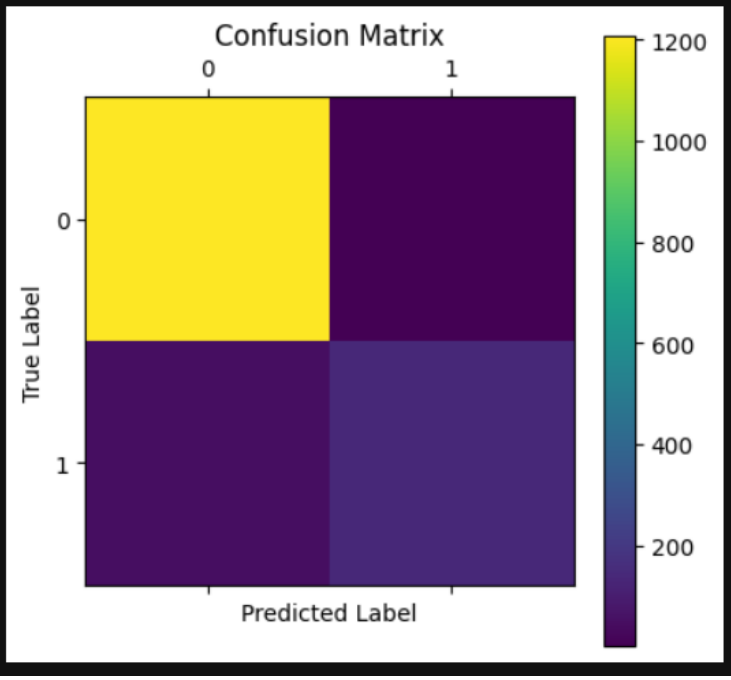
```
Nama : Deni Purwanto
[56]: array([[1207,    1],
             [  47,  138]])
```

```
[57]: tn, fp, fn, tp = matrix.ravel()

      print(f'TN: {tn}')
      print(f'FP: {fp}')
      print(f'FN: {fn}')
      print(f'TP: {tp}')
```

```
TN: 1207
FP: 1
FN: 47
TP: 138
```

```
[59]: import matplotlib.pyplot as plt

      plt.matshow(matrix)
      plt.colorbar()
      print("Nama : Deni Purwanto")
      plt.title('Confusion Matrix')
      plt.ylabel('True Label')
      plt.xlabel('Predicted Label')
      plt.show()
```

```
Nama : Deni Purwanto
```



3.8. Pengenalan Accuracy Score

```
[60]: from sklearn.metrics import accuracy_score

      print("Nama : Deni Purwanto")
      accuracy_score(y_test, y_pred)

      Nama : Deni Purwanto
[60]: 0.9655419956927495
```

3.9. Pengenalan Precision dan Recall

```
[61]: from sklearn.metrics import precision_score

      print("Nama : Deni Purwanto")
      precision_score(y_test, y_pred)

      Nama : Deni Purwanto
[61]: np.float64(0.9928057553956835)
```

```
[62]: from sklearn.metrics import recall_score

      print("Nama : Deni Purwanto")
      recall_score(y_test, y_pred)

      Nama : Deni Purwanto
[62]: np.float64(0.745945945945946)
```

3.10. Pengenalan F1 Score | F1 Measure

```
[63]: from sklearn.metrics import f1_score

      print("Nama : Deni Purwanto")
      f1_score(y_test, y_pred)

      Nama : Deni Purwanto
[63]: np.float64(0.8518518518518519)
```

3.11. Pengenalan ROC | Receiver Operating Characteristic

[64]:

```
from sklearn.metrics import roc_curve, auc
print("Nama : Deni Purwanto")
prob_estimates = model.predict_proba(X_test_tfidf)

fpr, tpr, threshhold = roc_curve (y_test, prob_estimates[:, 1])
nilai_auc = auc (fpr, tpr)

plt.plot(fpr, tpr, 'b', label=f'AUC={nilai_auc}')
plt.plot([0,1], [0,1], 'r--', label='Random Classifier')

plt.title('ROC: Receiver Operating Characteristic')
plt.xlabel('Fallout or False Positive Rate')
plt.ylabel('Recall or True Positive Rate')
plt.legend()
plt.show()
```

```
Nama : Deni Purwanto
```