Tugas Pertemuan 10 Data Sains dan Analisis

Link source code: https://drive.google.com/drive/folders/1q_QkjjIfpi8p43CBAt-4Bngk2Oh3RNpC?usp=sharing

1. Regresi Linear Sederhana

Source Code

```
import matplotlib.pyplot as plt
import pandas as pd
import pylab as pl
import numpy as np
import pandas as pd
%matplotlib inline

#membaca data
df = pd.read_csv("vgsales.csv")
df.head()

#eksplorasi data
df.describe()

cdf = df[['NA_Sales', 'EU_Sales', 'JP_Sales',
'Other_Sales','Global_Sales']]
cdf.head(10)

#visualisasi data histogram
viz = cdf[['NA_Sales', 'EU_Sales', 'JP_Sales',
'Other_Sales','Global_Sales']]
viz.hist()
plt.show()

#visualisasi data Other Sales vs Global Sales
plt.scatter(cdf.Other_Sales, cdf.Global_Sales,  color='blue')
plt.xlabel("Other_Sales")
plt.ylabel("Global_Sales")
plt.show()

#visualisasi data EU Sales vs Global Sales
plt.scatter(cdf.EU_Sales, cdf.Global_Sales,  color='blue')
plt.xlabel("EU_Sales")
plt.ylabel("Global_Sales")
plt.show()

#membuat dataset train dan test
msk = np.random.rand(len(df)) < 0.8
train = cdf[msk]
test = cdf[~msk]

#visualisasi data train Other Sales vs Global Sales
```

```
plt.scatter(train.EU_Sales, train.Global_Sales,  color='blue')
plt.xlabel("EU_Sales")
plt.ylabel("Global_Sales")
plt.show()
from sklearn import linear_model
regr = linear_model.LinearRegression()
train_x = np.asanyarray(train[['EU_Sales']])
train_y = np.asanyarray(train[['Global_Sales']])
regr.fit (train_x, train_y)
# The coefficients
print ('Coefficients: ', regr.coef_)
print ('Intercept: ',regr.intercept_)

#visualisasi data train EU Sales vs Global Sales dengan garis
plt.scatter(train.EU_Sales, train.Global_Sales,  color='blue')
plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], '-r')
plt.xlabel("EU_Sales")
plt.ylabel("Global_Sales")

#evaluasi
from sklearn.metrics import r2_score

test_x = np.asanyarray(test[['EU_Sales']])
test_y = np.asanyarray(test[['Global_Sales']])
test_y_ = regr.predict(test_x)

print("Mean absolute error: %.2f" % np.mean(np.absolute(test_y_ -
test_y)))
print("Residual sum of squares (MSE): %.2f" % np.mean((test_y_ - test_y)
** 2))
print("R2-score: %.2f" % r2_score(test_y_ , test_y) )
```

Output

```
Mean absolute error: 0.23
Residual sum of squares (MSE): 0.22
R2-score: 0.93
```

Output diatas menunjukkan performa model regresi linear dengan cukup baik. Mean absolute error sebesar 0.23 menunjukkan rata-rata selisih antara nilai prediksi dan nilai sebenarnya pada data testing sebesar 0.23. Residual sum of squares (MSE) sebesar 0.22 menunjukkan besarnya error total model pada data testing. R2-score sebesar 0.93 menunjukkan seberapa baik model dapat menjelaskan variansi pada data testing. Semakin mendekati 1, semakin baik performa model. Oleh karena itu, dapat disimpulkan bahwa model yang digunakan memiliki performa yang baik.

2. Regresi Linear Jamak

Souce Code

```
import matplotlib.pyplot as plt
import pandas as pd
import pylab as pl
import numpy as np
import pandas as pd
%matplotlib inline

df = pd.read_csv("vgsales.csv")

#membaca dataset
df.head()

#eksplorasi data
df.describe()

cdf = df[['NA_Sales', 'EU_Sales', 'JP_Sales',
'Other_Sales','Global_Sales']]
cdf.head(10)

#visualisasi data histogram
viz = cdf[['NA_Sales', 'EU_Sales', 'JP_Sales',
'Other_Sales','Global_Sales']]
viz.hist()
plt.show()

#visualisasi data Other Sales vs Global Sales
plt.scatter(cdf.Other_Sales, cdf.Global_Sales,  color='blue')
plt.xlabel("Other_Sales")
plt.ylabel("Global_Sales")
plt.show()

#visualisasi data EU Sales vs Global Sales
plt.scatter(cdf.EU_Sales, cdf.Global_Sales,  color='blue')
plt.xlabel("EU_Sales")
plt.ylabel("Global_Sales")
plt.show()

#visualisasi data NA Sales vs Global Sales
plt.scatter(cdf.NA_Sales, cdf.Global_Sales,  color='blue')
plt.xlabel("NA_Sales")
plt.ylabel("Global_Sales")
plt.show()

#membuat dataset train dan test
msk = np.random.rand(len(df)) < 0.8
train = cdf[msk]
```

```
test = cdf[~msk]

#visualisasi data train Other Sales vs Global Sales
plt.scatter(train.Other_Sales, train.Global_Sales,  color='blue')
plt.xlabel("Other_Sales")
plt.ylabel("Global_Sales")
plt.show()

#visualisasi data train EU Sales vs Global Sales
plt.scatter(train.EU_Sales, train.Global_Sales,  color='blue')
plt.xlabel("EU_Sales")
plt.ylabel("Global_Sales")
plt.show()

#visualisasi data train NA Sales vs Global Sales
plt.scatter(train.NA_Sales, train.Global_Sales,  color='blue')
plt.xlabel("NA_Sales")
plt.ylabel("Global_Sales")
plt.show()

from sklearn import linear_model
regr = linear_model.LinearRegression()
train_x = np.asanyarray(train[['EU_Sales']])
train_y = np.asanyarray(train[['Global_Sales']])
regr.fit (train_x, train_y)

#The coefficients
print ('Coefficients: ', regr.coef_)
print ('Intercept: ',regr.intercept_)

#visualisasi data train EU Sales vs Global Sales dengan garis
plt.scatter(train.EU_Sales, train.Global_Sales,  color='blue')
plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], '-r')
plt.xlabel("EU_Sales")
plt.ylabel("Global_Sales")

#evaluasi
from sklearn.metrics import r2_score

test_x = np.asanyarray(test[['EU_Sales']])
test_y = np.asanyarray(test[['Global_Sales']])
test_y_ = regr.predict(test_x)

print("Mean absolute error: %.2f" % np.mean(np.absolute(test_y_ -
test_y)))
print("Residual sum of squares (MSE): %.2f" % np.mean((test_y_ - test_y)
** 2))
print("R2-score: %.2f" % r2_score(test_y_ , test_y) )
```

Output

```
Mean absolute error: 0.24
Residual sum of squares (MSE): 0.28
R2-score: 0.92
```

Dari output tersebut, dapat disimpulkan bahwa model regresi linear jamak yang dibangun memiliki performa yang baik. Hal ini dapat dilihat dari nilai R-squared yang mencapai 0.92, yang artinya 92% variansi data dapat dijelaskan oleh model. Selain itu, nilai mean absolute error yang rendah yaitu sebesar 0.24 menunjukkan bahwa model memiliki ketepatan yang baik dalam memprediksi nilai target, sedangkan nilai residual sum of squares (MSE) yang rendah yaitu sebesar 0.28 menunjukkan bahwa model memiliki akurasi yang baik dalam memprediksi nilai target. Oleh karena itu, model ini dapat digunakan untuk memprediksi nilai target dengan akurasi yang cukup baik.

3. Multiple Linear Regression

Source Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#membaca data
df = pd.read_csv("winequality-red.csv")
df.head()

x = df[['density','pH', 'sulphates', 'alcohol']]
y = df['quality']

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3,
random_state = 100)

from sklearn.linear_model import LinearRegression

mlr = LinearRegression()
mlr.fit(x_train, y_train)

#Intercept and Coefficient
print("Intercept: ", mlr.intercept_)
print("Coefficients:")
list(zip(x, mlr.coef_))

#Prediction of test set
y_pred_mlr= mlr.predict(x_test)
#Predicted values
print("Prediction for test set: {}".format(y_pred_mlr))

#Actual value and the predicted value
```

```
mlr_diff = pd.DataFrame({'Actual value': y_test, 'Predicted value':
y_pred_mlr})
print(mlr_diff.head())

#Model Evaluation
from sklearn import metrics
meanAbErr = metrics.mean_absolute_error(y_test, y_pred_mlr)
meanSqErr = metrics.mean_squared_error(y_test, y_pred_mlr)
rootMeanSqErr = np.sqrt(metrics.mean_squared_error(y_test, y_pred_mlr))
print('R squared: {:.2f}'.format(mlr.score(x,y)*100))
print('Mean Absolute Error:', meanAbErr)
print('Mean Square Error:', meanSqErr)
print('Root Mean Square Error:', rootMeanSqErr)
```

Output

```
R squared: 28.21
Mean Absolute Error: 0.5341990702512317
Mean Square Error: 0.4787752352147102
Root Mean Square Error: 0.6919358606220017
```

R squared atau koefisien determinasi adalah pengukuran seberapa baik model dapat menjelaskan variasi dari data. Nilai R squared berkisar antara 0 dan 1, di mana semakin dekat dengan 1, semakin baik model dapat menjelaskan variasi data. Pada output tersebut, nilai R squared adalah 0.2821 yang menunjukkan bahwa model cukup mampu menjelaskan variasi dari data. Mean Absolute Error (MAE) adalah rata-rata dari selisih antara nilai prediksi dan nilai sebenarnya, di mana semua selisih dihitung sebagai nilai absolut dan kemudian dirata-ratakan. MAE mengukur seberapa dekat prediksi dengan nilai sebenarnya, dan semakin kecil nilainya yaitu sebesar 0,5341990, semakin baik performa model. Mean Square Error (MSE) adalah rata-rata dari kuadrat selisih antara nilai prediksi dan nilai sebenarnya. MSE juga mengukur seberapa dekat prediksi dengan nilai sebenarnya, tapi lebih sensitif terhadap perbedaan yang lebih besar antara prediksi dan nilai sebenarnya. Root Mean Square Error (RMSE) adalah akar dari MSE dan mengukur besarnya kesalahan prediksi dengan satuan yang sama dengan variabel target. Semakin kecil nilai RMSE, semakin baik performa model. Pada contoh tersebut, nilai RMSE adalah 0.6919 yang menunjukkan bahwa model cukup akurat dalam memprediksi target.