Webinar Series, 7 Nopember 2020
PROGRAM PASCASARJANA TERAPAN
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

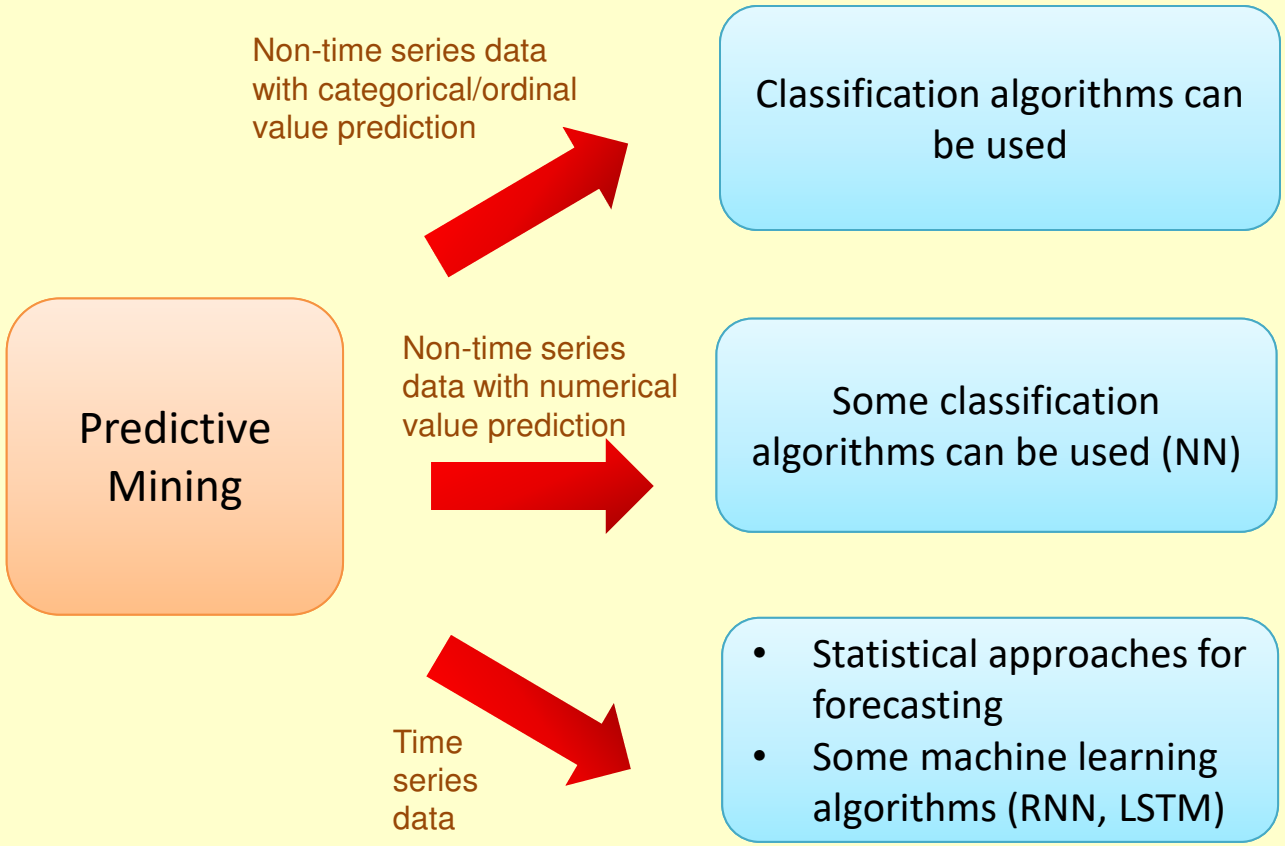
Workshop & Tutorial
Data Mining with Python

Predictive Mining

Ali Ridho Barakbah

Knowledge Engineering Laboratory
Department of Information and Computer Engineering
Politeknik Elektronika Negeri Surabaya

Predictive Mining



Predictive Modelling – (Linear Regression)

- Regression is a measuring tool used to determine whether there is a correlation between variables
- Regression analysis is more accurate in correlation analysis because the rate of change of a variable against other variables can be determined. So in regression, forecasting or estimating the value of the dependent variable on the independent variable is more accurate
- Linear regression is a regression where the independent variable (variable X) has the highest rank of one. For simple regression, i.e. linear regression which only involves 2 variables (variables X and Y)

Linear Regression from Y to X

$$Y = a + b * X$$

where:

Y = dependent variable

X = independent variable

a = intercept

b = slope (regression coefficient)

$$a = \frac{(\Sigma Y)(\Sigma X^2) - (\Sigma X)(\Sigma XY)}{(n)(\Sigma X^2) - (\Sigma X)^2}$$

$$b = \frac{(n)(\Sigma XY) - (\Sigma X)(\Sigma Y)}{(n)(\Sigma X^2) - (\Sigma X)^2}$$

Contoh

$$a = \frac{(\Sigma Y)(\Sigma X^2) - (\Sigma X)(\Sigma XY)}{(n)(\Sigma X^2) - (\Sigma X)^2} = \frac{(447 * 204) - (36 * 2344)}{(8 * 204) - (36 * 36)} = 20.25$$

$$b = \frac{(n)(\Sigma XY) - (\Sigma X)(\Sigma Y)}{(n)(\Sigma X^2) - (\Sigma X)^2} = \frac{(8 * 2344) - (36 * 447)}{(8 * 204) - (36 * 36)} = 7.9167$$

X	Y	X^2	XY
1	32	1	32
2	24	4	48
3	43	9	129
4	65	16	260
5	57	25	285
6	71	36	426
7	76	49	532
8	79	64	632
Sum 36	447	204	2344

$n = 8$

$Y = a + b X$

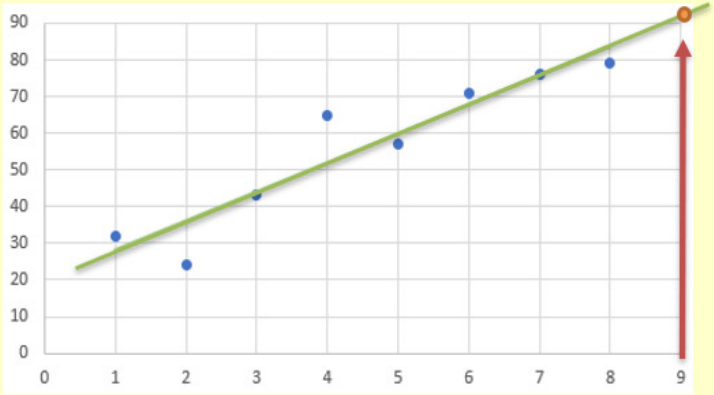
$= 20.25 + 7.9167 * X$

$x = 9$

$y = 20.25 + 7.9167 * 9$

$= 20.25 + 71.2503$

$= 91.5003$



Prediction Evaluation

$$\text{Mean Absolute Error (MAE)} = \frac{\sum_{t=1}^{N} \left| d_t - d_t' \right|}{N}$$

$$\text{Mean Squared Error (MSE)} = \frac{\sum_{t=1}^{N} \left(d_t - d_t' \right)^2}{N}$$

$$\text{Mean Absolute Percent Error (MAPE)} = \frac{100}{N} \sum_{t=1}^{N} \left[\left| \frac{d_t - d_t'}{d_t} \right| \right]$$

Predictive Mining dengan Linear Regression

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

dataset = pd.read_csv('ipm.csv')
dataset = dataset.dropna()
dataset = dataset.loc[(dataset['nama_provinsi']=='Prov. Jawa Timur')]
data = dataset[['nama_provinsi', 'tahun', 'ipm']]

avg_ipm = data.groupby('tahun')['ipm'].mean()

print('Rata-rata IPM per tahun\n', avg_ipm)

x=avg_ipm.index
y=avg_ipm.values

plt.scatter(x, y)
plt.plot(x, y)
plt.xlabel('Tahun')
plt.ylabel('Rata-Rata IPM')

linreg=LinearRegression()
x=np.array(x).reshape(-1,1)
linreg.fit(x, y)

IPM_2013=np.array(2013).reshape(-1,1)
pred_ipm=linreg.predict(IPM_2013)

print('\nPrediksi rata-rata IPM tahun 2013 =\n', pred_ipm.item())

plt.scatter(IPM_2013, pred_ipm, c='red')
pred_y=linreg.predict(x)
plt.plot(x, pred_y)
plt.show()

MSE=mean_squared_error(y,pred_y)
print('\nMSE = ', MSE)
```

```
Rata-rata IPM per tahun
 tahun
2004    66.860043
2005    67.476053
2006    68.420409
2007    69.092105
2008    69.582152
2009    70.144162
2010    70.712123
2011    71.291999
2012    71.873669
Name: ipm, dtype: float64

Prediksi rata-rata IPM tahun 2013 =
 72.58956525657891

MSE =  0.01632358744245256
```