|  |
| --- |
| **Pertemuan 4 – Teknik Praproses Data** |
|  |
| **Tujuan pembelajaran** |
| * Mahasiswa mampu menggunakan library scikit-learn untuk praproses data. * Mahasiswa mampu menerapkan beberapa teknik praproses data seperti pembersihan data, normalisasi data, penggabungan data, reduksi data, dan pembagian data. |

Data Tabular – Visualisasi Iris Datataset

|  |
| --- |
| # lib manipulasi data  import pandas as pd  import numpy as np    # lib visualisasi data  import seaborn as sns  import matplotlib.pyplot as plt    # lib preprocessing  from sklearn.preprocessing import MinMaxScaler  from sklearn.preprocessing import StandardScaler  from sklearn.model\_selection import train\_test\_split |
|  |
| # load dataset  dataset = pd.read\_csv("../dataset/dataset\_iris.csv") |
|  |
| # Set features and Labels  x = dataset[["sepal\_length","sepal\_width","petal\_length","petal\_width"]].values  y = dataset["species"].values |

**MinMax - Scaler**

|  |
| --- |
| # process normalization data  scaler = MinMaxScaler(feature\_range=(0,1))  scaled = scaler.fit\_transform(x) |
|  |
| # results normalization data  minmax = pd.concat([    pd.DataFrame(scaled, columns=["sepal\_length","sepal\_width","petal\_length","petal\_width"]),    pd.DataFrame(y, columns=["species"])  ],axis=1) |
| # results normalization data  print(minmax.tail()) |
|  |
| sepal\_length sepal\_width petal\_length petal\_width species  145 0.666667 0.416667 0.711864 0.916667 virginica  146 0.555556 0.208333 0.677966 0.750000 virginica  147 0.611111 0.416667 0.711864 0.791667 virginica  148 0.527778 0.583333 0.745763 0.916667 virginica  149 0.444444 0.416667 0.694915 0.708333 virginica |

**Standard - Scaler**

|  |
| --- |
| # process normalization data  scaled = StandardScaler().fit\_transform(x) |
|  |
| # results normalization data  standard = pd.concat([    pd.DataFrame(scaled, columns=["sepal\_length","sepal\_width","petal\_length","petal\_width"]),    pd.DataFrame(y, columns=["species"])  ],axis=1) |
|  |
| # results normalization data  print(standard.tail()) |
|  |
| sepal\_length sepal\_width petal\_length petal\_width species  145 1.038005 -0.124958 0.819624 1.447956 virginica  146 0.553333 -1.281972 0.705893 0.922064 virginica  147 0.795669 -0.124958 0.819624 1.053537 virginica  148 0.432165 0.800654 0.933356 1.447956 virginica  149 0.068662 -0.124958 0.762759 0.790591 virginica |

|  |  |
| --- | --- |
| # function boxplot  def boxplot(dataset, title):      # create boxplot with seaborn    fig, ax = plt.subplots(figsize=(8,4))    sns.boxplot(data=dataset, x="species", y="petal\_length", hue="species")      # set labels    ax.set\_title(title, fontsize=12)    ax.set\_xlabel("", fontsize=12)    ax.set\_ylabel("", fontsize=12)    ax.grid(True)      # return values    plt.tight\_layout()    plt.show() | |
|  | |
| # call function boxplot  boxplot(minmax, "Boxplot with min-max scaler")  boxplot(standard, "Boxplot with standard scaler") | |
|  | |
|  |  |
| Gambar x. Output program | |

|  |  |
| --- | --- |
| # function heatmap  def heatmap(dataset, title):      # create heatmap with seaborn    fig, ax = plt.subplots(figsize=(8,4))    sns.heatmap(      data=dataset.corr(numeric\_only=True), vmin=-1, vmax=1, cmap="viridis", annot=True)      # set labels    ax.set\_title(title, fontsize=12)    ax.grid(False)      # show plot    plt.tight\_layout()    plt.show() | |
|  | |
| # call function heatmap  heatmap(minmax, "Heatmap with min-max scaler")  heatmap(standard, "Heatmap with standard scaler") | |
|  | |
|  |  |
| Gambar x. Output program | |

|  |  |
| --- | --- |
| # function scatter plot  def scatter(data, x, y, title):      # create scatter plots    fig, ax = plt.subplots(figsize = (8,4))    sns.scatterplot(data=data, x=x, y=y, hue="species")      # set labels    ax.set\_title(title, fontsize=12)    ax.set\_xlabel("", fontsize=12)    ax.set\_ylabel("", fontsize=12)    ax.grid(True)      # return values    plt.tight\_layout()    plt.show() | |
|  | |
| # call func scatterplot  scatter(minmax, "petal\_length", "petal\_width", "Heatmap with min-max scaler")  scatter(standard, "petal\_length", "petal\_width", "Heatmap with standard scaler") | |
|  | |
|  |  |
| Gambar x. Output program | |

Data Time Series– Visualisasi Cryptocurrency dan Stock Price

|  |
| --- |
| # lib manipulasi data  import pandas as pd  import numpy as np    # lib visualisasi data  import seaborn as sns  import matplotlib.pyplot as plt    # lib preprocessing  from sklearn.preprocessing import MinMaxScaler  from sklearn.preprocessing import StandardScaler  from sklearn.model\_selection import train\_test\_split |
|  |
| # func load dataset  def load\_dataset(df):      # load dataset    dataset = pd.read\_csv("../../dataset/"+df, parse\_dates=['Date'])      # set feature    dataset = dataset[["Date", "Open", "High", "Low", "Close"]]      # set index    dataset = dataset.set\_index("Date")      # return values    return dataset |
|  |

|  |
| --- |
|  |
|  |
|  |
|  |