Evaluasi Model Machine Learning Menggunakan Dataset Student Score

Data Series 15.0 AI Machine Learning Dibimbing

Muhammad Aziz Habiburrahim

Table of Contents

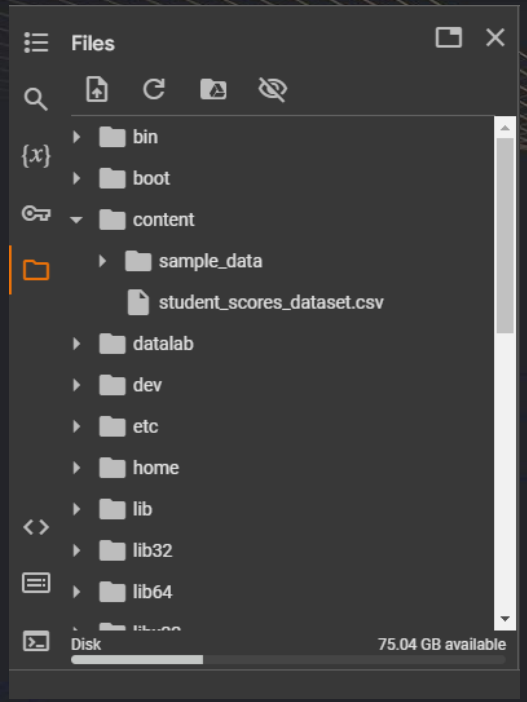
01 IMPORT LIBRARY AND RESOURCE

Pertama-tama, terlebih dahulu melakukan kode import library meliputi pandas, numpy, matplotlib.pyplot, seaborn dan kode import resource meliputi sklearn.model_selection, sklearn.linear_model, sklearn.tree, sklearn.ensemble, sklearn.metrics, google.colab.

```
[27] # import libraries and resources
     import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression
     from sklearn.tree import DecisionTreeRegressor
     from sklearn.ensemble import RandomForestRegressor
     from sklearn.metrics import mean_squared_error, r2_score
     from google.colab import sheets
```

01 IMPORT LIBRARY AND RESOURCE

Setelah import, lakukan kode untuk mendeteksi dataset yang akan dipakai menggunakan pd.read_csv. Pastikan sudah mengupload file CSV pada bagian content.




```
[18] # read the dataset using pandas
     data = pd.read_csv('/content/student_scores_dataset.csv')
```

ibimbing

02 EXPLORATORY DATA ANALYSIS

Setelah dataset terbaca oleh system, lakukan pencarian informasi data yang diperlukan.

```
[29] # general information about the dataset
     data.info()
     data.describe()
```

```
[24] # this displays the top 100 rows of the data
     data.head(100)
```

```
[31] # convert data to spreadsheets
     sheet = sheets.InteractiveSheet(df=data)
```

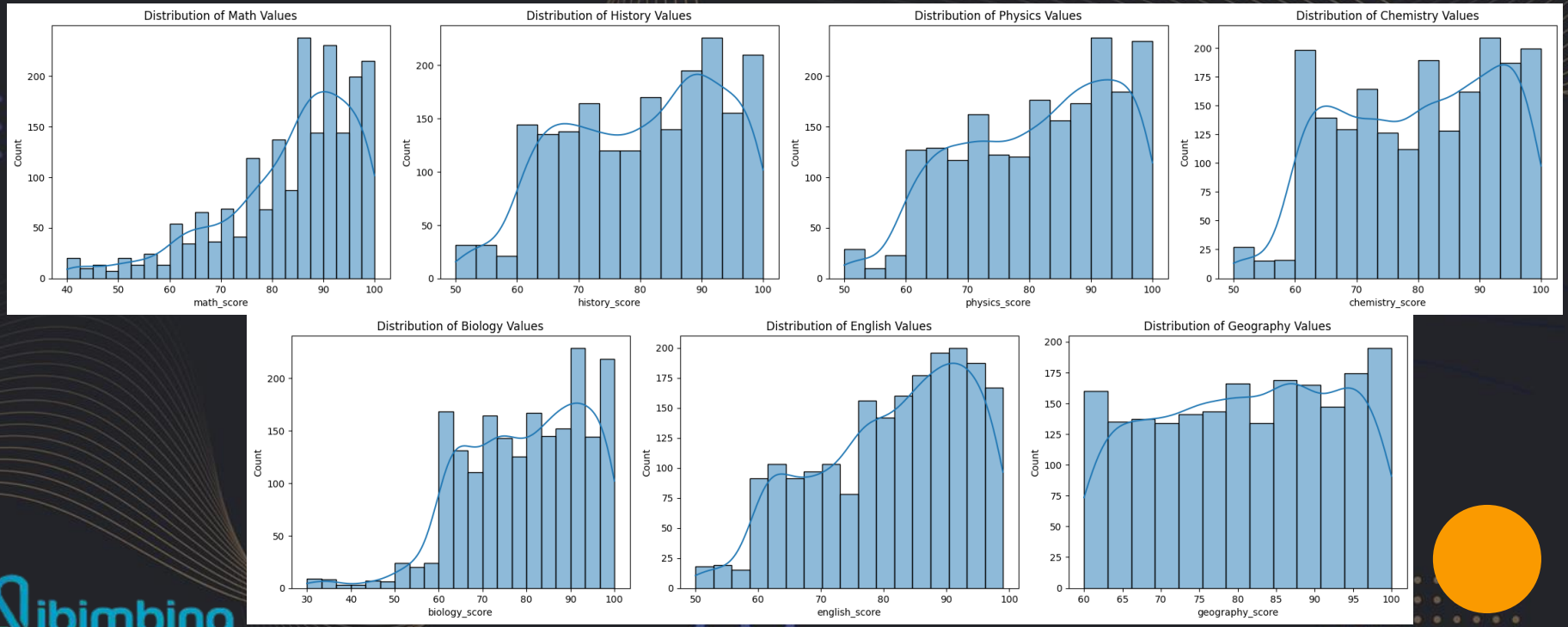
02 EXPLORATORY DATA ANALYSIS

Cek distribusi nilai dari tiap pelajaran yg terdiri dari Matematika, Sejarah, Fisika, Kimia, Biologi, Inggris dan Geografi. Distribusi menunjukkan interval nilai dengan banyaknya siswa memperoleh nilai tersebut.

```
[32] # check the distribution of math values
     sns.histplot(data['math_score'], kde=True)
     plt.title('Distribution of Math Values')
     plt.show()

[33] # check the distribution of history values
     sns.histplot(data['history_score'], kde=True)
     plt.title('Distribution of History Values')
     plt.show()

[34] # check the distribution of physics values
     sns.histplot(data['physics_score'], kde=True)
     plt.title('Distribution of Physics Values')
     plt.show()

[36] # check the distribution of chemistry values
     sns.histplot(data['chemistry_score'], kde=True)
     plt.title('Distribution of Chemistry Values')
     plt.show()

[37] # check the distribution of biology values
     sns.histplot(data['biology_score'], kde=True)
     plt.title('Distribution of Biology Values')
     plt.show()

[38] # check the distribution of english values
     sns.histplot(data['english_score'], kde=True)
     plt.title('Distribution of English Values')
     plt.show()

[39] # check the distribution of geography values
     sns.histplot(data['geography_score'], kde=True)
     plt.title('Distribution of Geography Values')
     plt.show()
```

ibimbing

02 EXPLORATORY DATA ANALYSIS

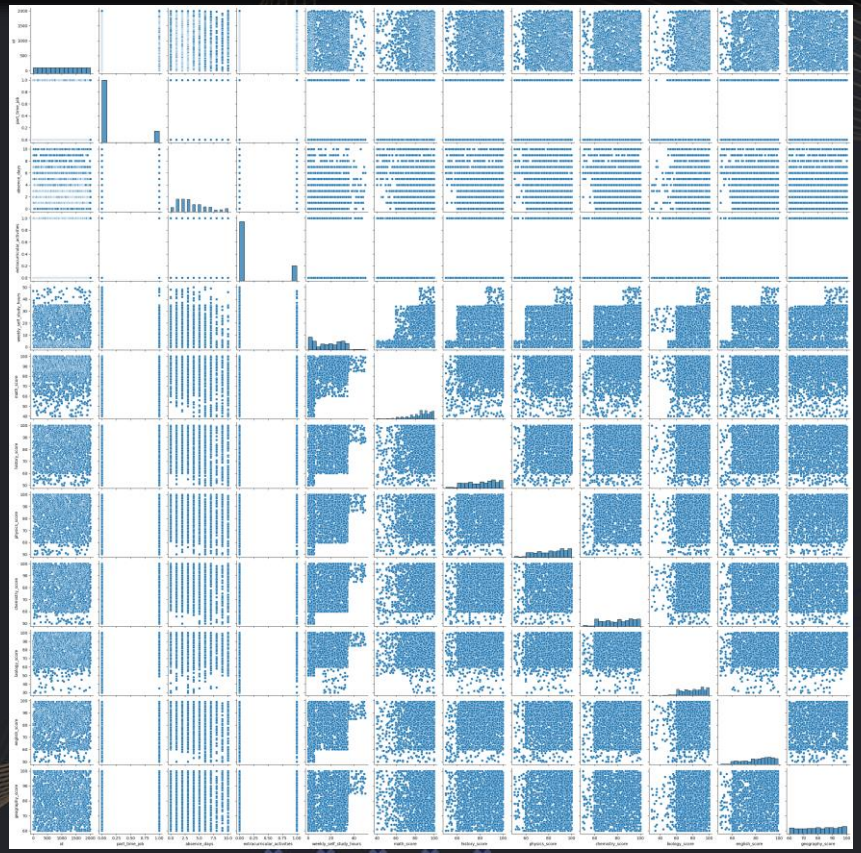


02 EXPLORATORY DATA ANALYSIS

Cek hubungan grafik antara dari dua setiap kolom menggunakan sns.pairplot yang berukuran 12 x 12.

```
[41] # check the relationship between features
     sns.pairplot(data)
     plt.show()
```

02 EXPLORATORY DATA ANALYSIS



03 FEATURE ENGINEERING

Cek duplikat data pada dataset student score, lakukan kode untuk mengecek duplikat menggunakan drop_cuplicates.

```
[42] # check duplicated data
     print("Number of Duplicate Data:", data.duplicated().sum())
     data = data.drop_duplicates()
```

```
Number of Duplicate Data: 0
```

ibimbing

03 FEATURE ENGINEERING

Cek nilai yang bermasalah pada dataset, lakukan kode untuk megecek nilai yang bermasalah menggunakan data.isnull.

```
[43] # check missing value handling
     print("Check Missing Values:")
     print(data.isnull().sum())
```

```
Check Missing Values:
id                          0
first_name                  0
last_name                   0
email                       0
gender                      0
part_time_job               0
absence_days                0
extracurricular_activities  0
weekly_self_study_hours     0
career_aspiration           0
math_score                  0
history_score               0
physics_score               0
chemistry_score             0
biology_score               0
english_score               0
geography_score             0
dtype: int64
```

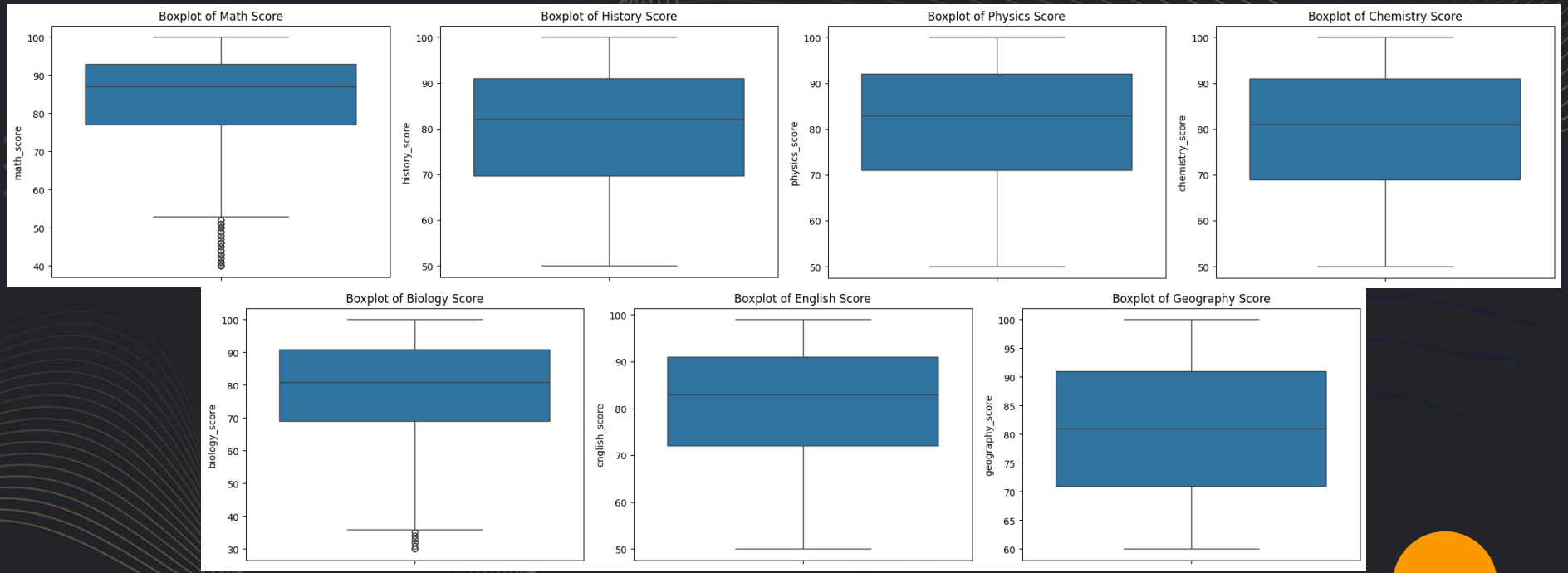
03 FEATURE ENGINEERING

Cek nilai setiap pelajaran menggunakan visualisasi boxplot dengan kode sns.boxplot untuk melihat rentang nilai rata rata.

```
[44] # visualisasi boxplot math score
     sns.boxplot(data['math_score'])
     plt.title('Boxplot of Math Score')
     plt.show()
[45] # visualisasi boxplot history score
     sns.boxplot(data['history_score'])
     plt.title('Boxplot of History Score')
     plt.show()
[46] # visualisasi boxplot physics score
     sns.boxplot(data['physics_score'])
     plt.title('Boxplot of Physics Score')
     plt.show()
[47] # visualisasi boxplot chemistry score
     sns.boxplot(data['chemistry_score'])
     plt.title('Boxplot of Chemistry Score')
     plt.show()
[48] # visualisasi boxplot biology score
     sns.boxplot(data['biology_score'])
     plt.title('Boxplot of Biology Score')
     plt.show()
[49] # visualisasi boxplot english score
     sns.boxplot(data['english_score'])
     plt.title('Boxplot of English Score')
     plt.show()
[50] # visualisasi boxplot geography score
     sns.boxplot(data['geography_score'])
     plt.title('Boxplot of Geography Score')
     plt.show()
```

ibimbing

03 FEATURE ENGINEERING



04 MACHINE LEARNING MODELLING

Lakukan split data atau variabel baru untuk melakukan testing model dari setiap pelajaran. Pada kode di bawah merupakan kode untuk pelajaran matematika, untuk pelajaran lainnya silahkan disesuaikan

```
[51] # assume the 'weekly_self_study_hours' column as a feature and 'math_score' as a target
     X = data[['weekly_self_study_hours']]
     Ymath = data['math_score']

     # split data into training and testing
     X_train, X_test, Ymath_train, Ymath_test = train_test_split(X, Ymath, test_size=0.2, random_state=42)
```

04 MACHINE LEARNING MODELLING

Lakukan training tes model yang terdiri dari Linear Regression, Decision Tree Regressor dan Random Forest Regressor

```
[58] # linear regression
     lr = LinearRegression()
     lr.fit(X_train, Ymath_train)
     Ymath_pred_lr = lr.predict(X_test)

     # decision tree regressor
     dt = DecisionTreeRegressor(random_state=42)
     dt.fit(X_train, Ymath_train)
     Ymath_pred_dt = dt.predict(X_test)

     # random forest regressor
     rf = RandomForestRegressor(random_state=42)
     rf.fit(X_train, Ymath_train)
     Ymath_pred_rf = rf.predict(X_test)
```

04 MACHINE LEARNING MODELLING

Lakukan evaluasi model dengan skor MSE dan R2 menggunakan evaluate_model. Lalu pilihlah model terbaik dengan skor R2 tertinggi dan skor MSE terendah.

```
[60] # functions for model evaluation
     def evaluate_model(Ymath_test, Ymath_pred, model_name):
         mse = mean_squared_error(Ymath_test, Ymath_pred)
         r2 = r2_score(Ymath_test, Ymath_pred)
         print(f"{model_name} - MSE: {mse:.2f}, R2 Score: {r2:.2f}")

     evaluate_model(Ymath_test, Ymath_pred_lr, "Linear Regression")
     evaluate_model(Ymath_test, Ymath_pred_dt, "Decision Tree Regressor")
     evaluate_model(Ymath_test, Ymath_pred_rf, "Random Forest Regressor")
```

```
Linear Regression - MSE: 139.04, R2 Score: 0.16
Decision Tree Regressor - MSE: 144.23, R2 Score: 0.13
Random Forest Regressor - MSE: 144.83, R2 Score: 0.13
```

05 CONCLUSIONS

Dari ke 7 pelajaran, dapat disimpulkan bahwa model terbaik untuk Linear Regression ada 3, Random Forest Regressor ada 3, dan Decision Tree Regressor ada 1.

Math : The best model is Linear Regression with the lowest MSE (111.61) and highest R2 (0.07) Score.

History : The best model is Linear Regression with the lowest MSE (49.50) and highest R2 (0.04) Score.

Physics : The best model is Random Forest Regressor with the lowest MSE (144.33) and highest R2 (0.04) Score.

Chemistry : The best model is Random Forest Regressor with the lowest MSE (143.91) and highest R2 (0.07) Score.

Biology : The best model is Random Forest Regressor with the lowest MSE (172.19) and highest R2 (0.01) Score.

English : The best model is Linear Regression with the lowest MSE (129.57) and highest R2 (0.06) Score.

Geography : The best model is Decision Tree Regressor with the lowest MSE (130.56) and highest R2 (0.04) Score.

Thanks!

Do you have any questions?
azizhabibrahim@gmail.com