LAPORAN

MACHINE LEARNING

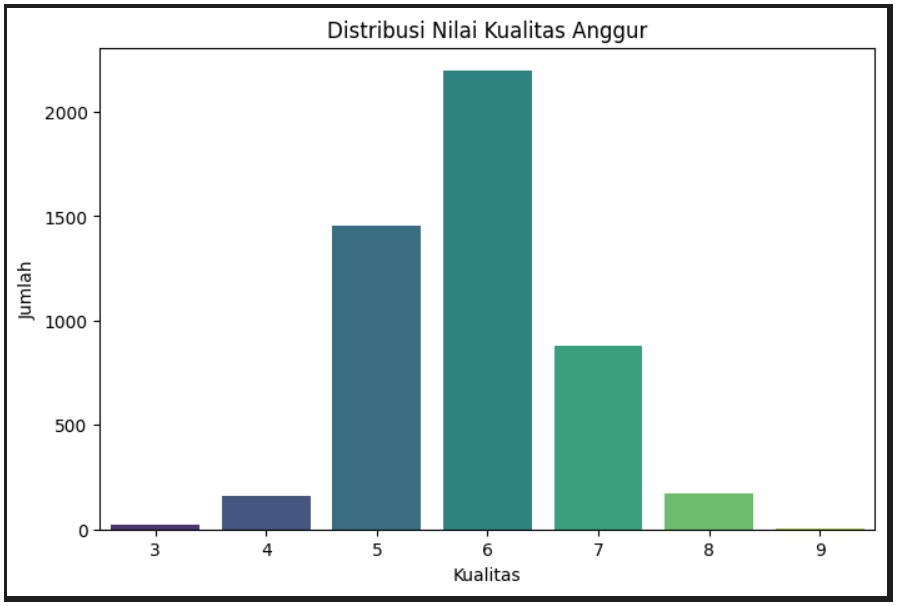
WEEK 14

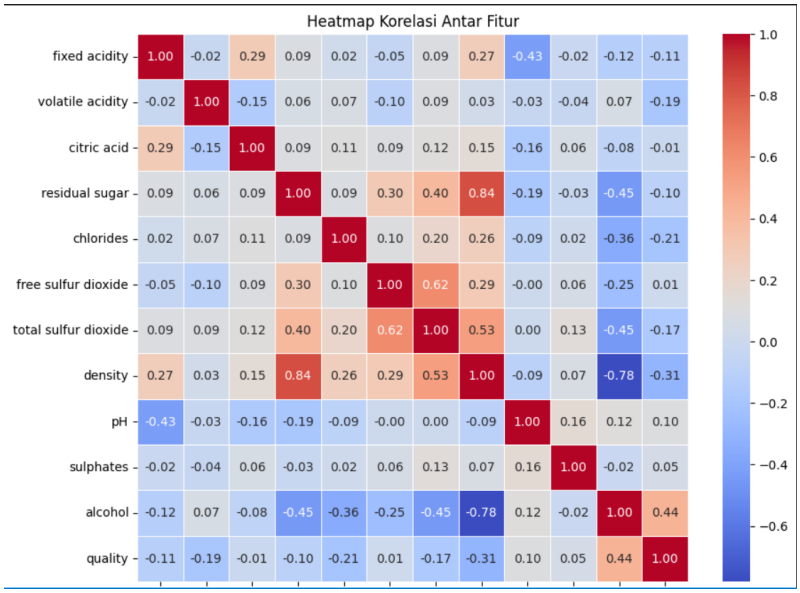
RNN Model dataset Wine Quality

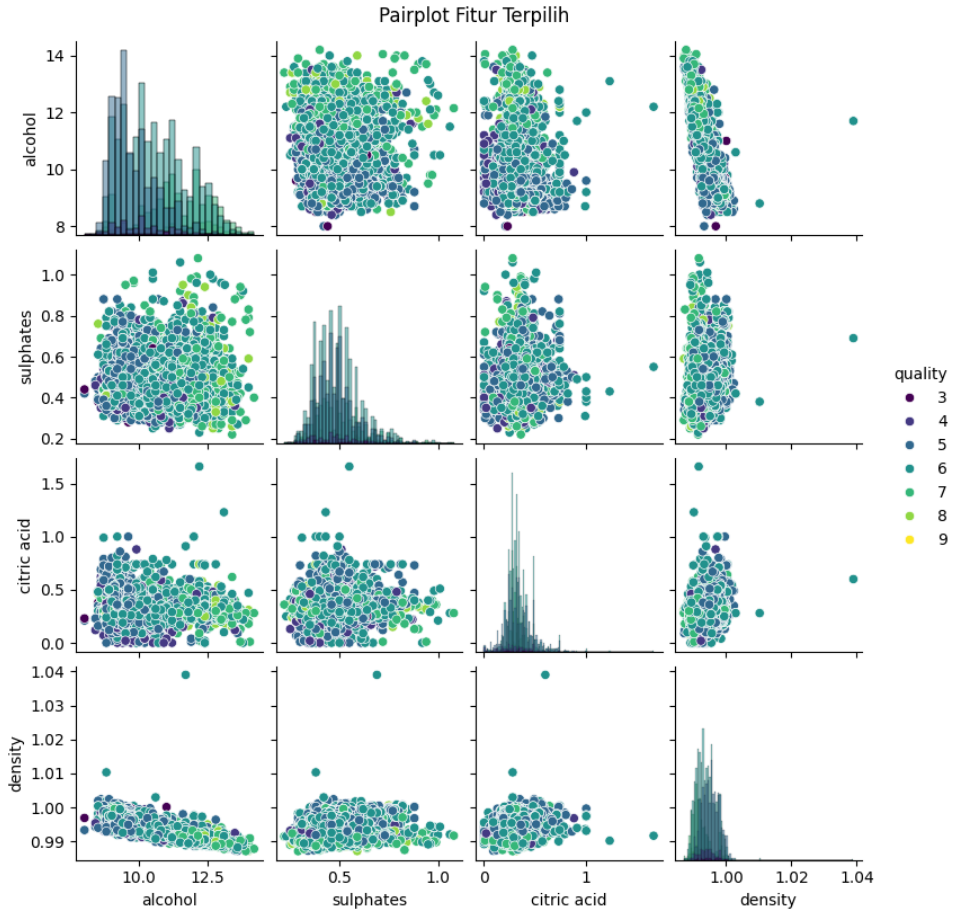
Recurrent Neural Network (RNN) adalah arsitektur jaringan saraf yang dirancang khusus untuk menangani data sekuensial, seperti teks, audio, dan data deret waktu. RNN berbeda dari jaringan saraf feedforward karena memiliki memori internal yang memungkinkan informasi dari langkah sebelumnya dipertimbangkan saat memproses data saat ini.

```
# Load and preprocess the dataset
file_path = 'winequality-white.csv'
data = pd.read_csv(file_path, delimiter=';')

# Exploratory Data Analysis (EDA)
plt.figure(figsize=(8, 5))
sns.countplot(data=data, x='quality', palette='viridis')
plt.title('Distribusi Nilai Kualitas Anggur')
plt.xlabel('Kualitas')
plt.ylabel('Jumlah')
plt.show()

plt.figure(figsize=(10, 8))
correlation_matrix = data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Heatmap Korelasi Antar Fitur')
plt.show()

selected_features = ['alcohol', 'sulphates', 'citric acid', 'density', 'quality']
sns.pairplot(data[selected_features], hue='quality', palette='viridis', diag_kind='hist', height=2)
plt.suptitle('Pairplot Fitur Terpilih', y=1.02)
plt.show()
```

Exploratory Data Analysis (EDA)




Heatmap Korelasi Antar Fitur
fixed acidity 1.00 -0.02 0.29 0.09 0.02 -0.05 0.09 0.27 -0.43 -0.02 -0.12 -0.11
volatile acidity -0.02 1.00 -0.15 0.06 0.07 -0.10 0.09 0.03 -0.03 -0.04 0.07 -0.19
citric acid 0.29 -0.15 1.00 0.09 0.11 0.09 0.12 0.15 -0.16 0.06 -0.08 -0.01
residual sugar 0.09 0.06 0.09 1.00 0.09 0.30 0.40 0.84 -0.19 -0.03 -0.45 -0.10
chlorides 0.02 0.07 0.11 0.09 1.00 0.10 0.20 0.26 -0.09 0.02 -0.36 -0.21
free sulfur dioxide -0.05 -0.10 0.09 0.30 0.10 1.00 0.62 0.29 -0.00 0.06 -0.25 0.01
total sulfur dioxide 0.09 0.09 0.12 0.40 0.20 0.62 1.00 0.53 0.00 0.13 -0.45 -0.17
density 0.27 0.03 0.15 0.84 0.26 0.29 0.53 1.00 -0.09 0.07 -0.78 -0.31
pH -0.43 -0.03 -0.16 -0.19 -0.09 -0.00 0.00 -0.09 1.00 0.16 0.12 0.10
sulphates -0.02 -0.04 0.06 -0.03 0.02 0.06 0.13 0.07 0.16 1.00 -0.02 0.05
alcohol -0.12 0.07 -0.08 -0.45 -0.36 -0.25 -0.45 -0.78 0.12 -0.02 1.00 0.44
quality -0.11 -0.19 -0.01 -0.10 -0.21 0.01 -0.17 -0.31 0.10 0.05 0.44 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


Pairplot Fitur Terpilih
alcohol
sulphates
citric acid
density
quality
3
4
5
6
7
8
9

```python
# Define the RNN model
class RNNModel(nn.Module):
    def __init__(self, input_size, hidden_size, output_size, pooling_type='max'):
        super(RNNModel, self).__init__()
        self.rnn = nn.RNN(input_size, hidden_size, batch_first=True)
        self.fc = nn.Linear(hidden_size, output_size)
        self.pooling_type = pooling_type

    def forward(self, x):
        out, _ = self.rnn(x)
        if self.pooling_type == 'max':
            out = torch.max(out, dim=1)[0]
        elif self.pooling_type == 'avg':
            out = torch.mean(out, dim=1)
        out = self.fc(out)
        return out

# Experiment configurations
hidden_sizes = [16, 32, 64]
pooling_types = ['max', 'avg']
epochs_list = [5, 50, 100, 250]
optimizers = {'SGD': optim.SGD, 'RMSProp': optim.RMSprop, 'Adam': optim.Adam}

best_accuracy = 0
best_config = {}
```

implementasi model Recurrent Neural Network (RNN) dalam PyTorch, dengan fitur tambahan pooling dan konfigurasi eksperimen untuk melakukan hyperparameter tuning. Kelas RNNModel mendefinisikan arsitektur dasar RNN, yang terdiri dari lapisan rekuren (nn.RNN) dan lapisan fully connected (nn.Linear) untuk menghasilkan output akhir. Pada konstruktor (__init__), model menerima parameter seperti ukuran input (input_size), ukuran unit tersembunyi (hidden_size), ukuran output (output_size), dan jenis pooling (pooling_type), yang dapat berupa 'max' atau 'avg'. Metode forward mengatur jalannya data melalui model. Output dari lapisan rekuren diproses dengan pooling—max pooling untuk mengambil nilai maksimum di sepanjang dimensi waktu atau average pooling untuk mengambil rata-rata. Output hasil pooling kemudian diteruskan ke lapisan fully connected untuk prediksi akhir.

Di bagian bawah kode, terdapat konfigurasi eksperimen dengan berbagai kombinasi hyperparameter. Parameter yang diuji meliputi ukuran unit tersembunyi (hidden_sizes), jenis pooling (pooling_types), jumlah epoch (epochs_list), dan optimizer (optimizers). Model ini dirancang untuk mengeksplorasi kombinasi hyperparameter terbaik, dengan menyimpan akurasi validasi tertinggi di variabel best_accuracy dan konfigurasi terbaik di best_config. Kode ini menunjukkan fleksibilitas model RNN untuk menangani berbagai tugas data sekuensial melalui eksperimen sistematis.

Output

```
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=SGD
Epoch [1/5], Loss: 1.9749
Epoch [2/5], Loss: 1.5103
Epoch [3/5], Loss: 1.4872
Epoch [4/5], Loss: 1.3768
Epoch [5/5], Loss: 1.5281
Accuracy: 0.3653
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=RMSProp
Epoch [1/5], Loss: 1.0548
Epoch [2/5], Loss: 1.1191
Epoch [3/5], Loss: 1.4042
Epoch [4/5], Loss: 1.6009
Epoch [5/5], Loss: 1.1519
Accuracy: 0.3694
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=Adam
Epoch [1/5], Loss: 1.4085
Epoch [2/5], Loss: 1.3744
Epoch [3/5], Loss: 1.1278
Epoch [4/5], Loss: 1.1036
Epoch [5/5], Loss: 1.3091
Accuracy: 0.3571
Training with hidden_size=16, pooling_type=max, epochs=50, optimizer=SGD
Epoch [1/50], Loss: 1.7885
Epoch [2/50], Loss: 1.4429
Epoch [3/50], Loss: 1.8175
...
Epoch [248/250], Loss: 1.2057
Epoch [249/250], Loss: 1.3741
Epoch [250/250], Loss: 1.0445
Accuracy: 0.3214
```

```python
# Display the best configuration and accuracy
print("Best Configuration:", best_config)
print("Best Accuracy:", best_accuracy)
```

```
Best Configuration: {'hidden_size': 64, 'pooling_type': 'max', 'epochs': 50, 'optimizer': 'RMSProp'}
Best Accuracy: 0.38571428571428573
```

Analisis Hasil

1. Akurasi Validasi Rendah (38.57%)
    - Akurasi validasi yang relatif rendah menunjukkan bahwa model belum cukup optimal dalam menangkap pola pada dataset. Hal ini mungkin disebabkan oleh:
        - Sifat sederhana dari arsitektur RNN standar yang kurang mampu menangani dependensi jangka panjang dalam data sekuensial.
        - Dataset yang mungkin memiliki pola kompleks sehingga memerlukan model yang lebih canggih seperti LSTM atau GRU.
        - Kombinasi hyperparameter yang meskipun terbaik dalam eksperimen, belum sepenuhnya optimal untuk dataset ini.
2. Konfigurasi Hyperparameter Terbaik
    - hidden_size = 64: Ukuran ini cukup besar untuk menangkap pola dalam data tanpa menyebabkan overfitting, namun mungkin masih belum cukup untuk pola yang sangat kompleks.

- pooling_type = 'max': Max pooling memberikan hasil terbaik, yang mengindikasikan bahwa memilih fitur maksimum dalam dimensi waktu membantu meningkatkan performa dibandingkan average pooling.
- epochs = 50: Epoch ini menunjukkan bahwa model mencapai konvergensi tanpa memerlukan pelatihan lebih lama, namun masih belum cukup untuk performa yang lebih tinggi.
- optimizer = 'RMSProp': Optimizer RMSProp lebih unggul dalam eksperimen ini, mungkin karena kemampuannya menyesuaikan langkah pembelajaran secara dinamis sesuai dengan gradien.

3. Keterbatasan RNN Standar
   - Masalah seperti vanishing gradient sering terjadi pada RNN standar, yang menyulitkan model untuk menangkap pola jangka panjang dalam data sekuensial.
   - Arsitektur yang sederhana tidak memiliki mekanisme memori eksplisit seperti yang dimiliki oleh LSTM atau GRU, yang membuatnya sulit untuk memahami konteks data yang lebih dalam**.**

Kesimpulan

Eksperimen ini menunjukkan bahwa konfigurasi dengan hidden_size = 64, pooling_type = 'max', epochs = 50, dan optimizer RMSProp memberikan akurasi validasi terbaik sebesar 38.57%. Namun, hasil ini menunjukkan bahwa arsitektur RNN standar mungkin tidak cukup kuat untuk menangani kompleksitas dataset.