LAPORAN

MACHINE LEARNING

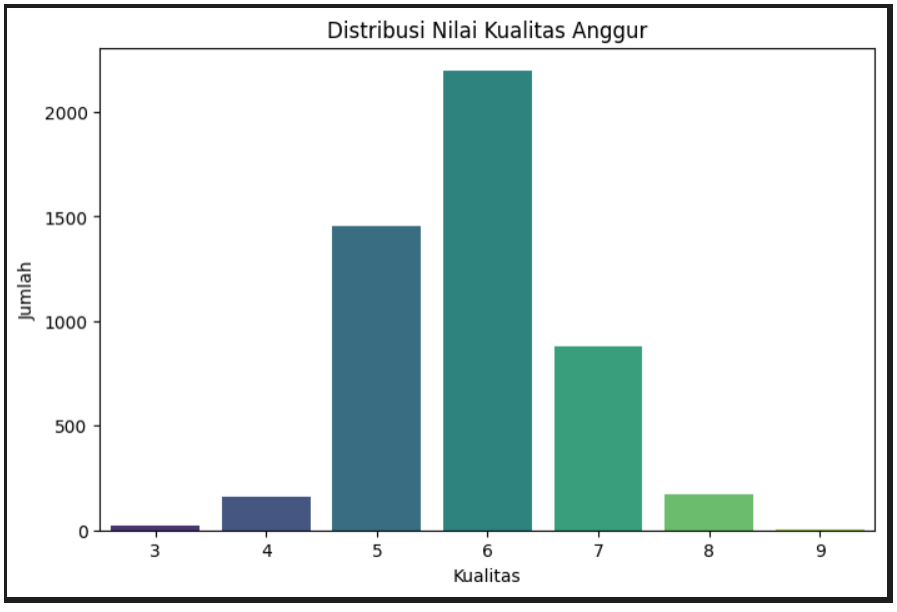
WEEK 14

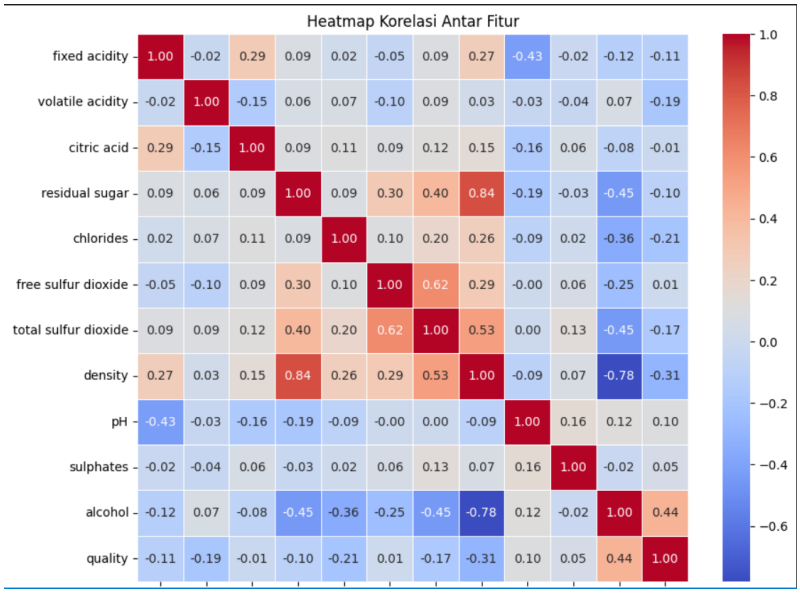
Deep RNN Model dataset Wine Quality

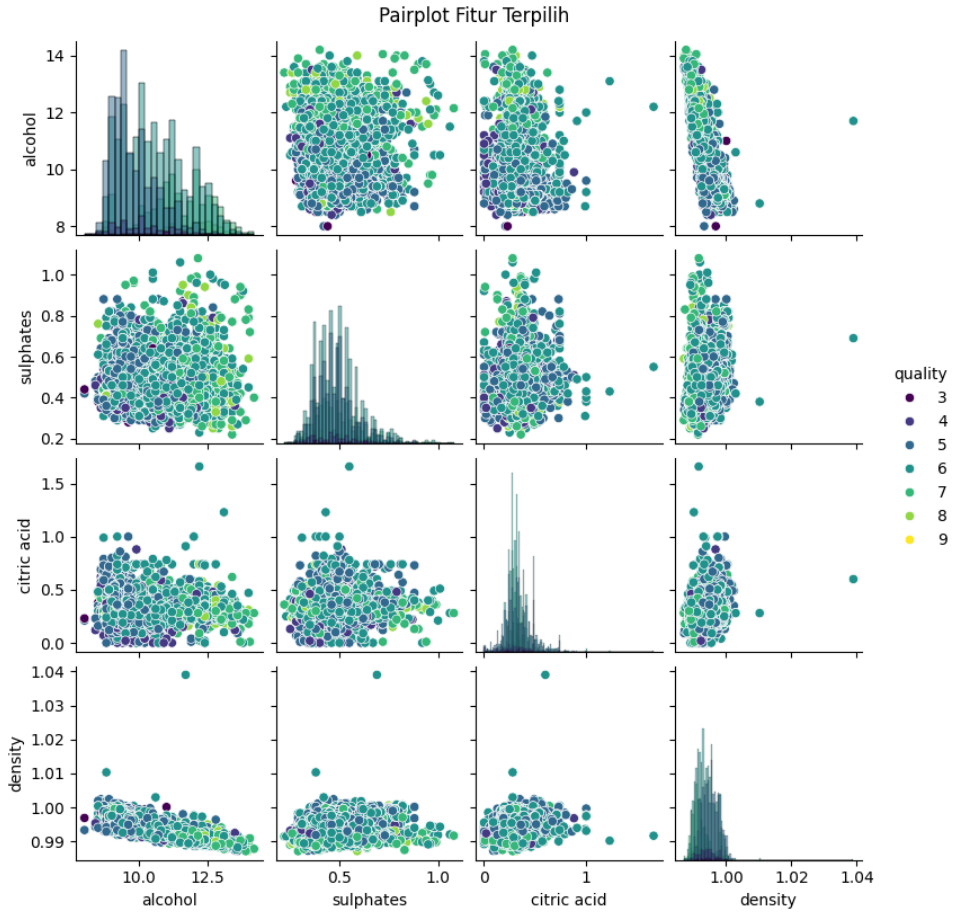
Deep RNN adalah pengembangan dari RNN standar, di mana lapisan rekuren digabungkan secara bertingkat (stacked). Artinya, output dari satu lapisan rekuren menjadi input untuk lapisan rekuren berikutnya. Hal ini meningkatkan kemampuan jaringan untuk menangkap hubungan kompleks dalam data sekuensial.

```
# 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 Deep RNN model
class DeepRNNModel(nn.Module):
    def __init__(self, input_size, hidden_size, output_size, num_layers=2, pooling_type='max'):
        super(DeepRNNModel, self).__init__()
        self.rnn = nn.RNN(input_size, hidden_size, num_layers=num_layers, 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 Deep Recurrent Neural Network (Deep RNN) menggunakan PyTorch. Kelas DeepRNNModel dirancang untuk memproses data sekuensial dengan lapisan rekuren bertingkat, memungkinkan model menangkap pola yang lebih kompleks dibandingkan RNN standar. Konstruktor (__init__) menerima beberapa parameter, termasuk ukuran input (input_size), ukuran unit tersembunyi (hidden_size), ukuran output (output_size), jumlah lapisan rekuren (num_layers), dan jenis pooling (pooling_type). Lapisan rekuren menggunakan nn.RNN dengan beberapa lapisan (num_layers) untuk memperkuat kemampuan menangkap pola jangka panjang. Lapisan fully connected (nn.Linear) digunakan untuk memetakan output terakhir dari lapisan rekuren menjadi prediksi akhir.

Metode forward menentukan alur data melalui model. Data masukan diproses oleh lapisan rekuren, dan hasilnya dilanjutkan ke mekanisme pooling. Dua jenis pooling didukung: max untuk mengambil nilai maksimum sepanjang dimensi waktu, dan avg untuk menghitung rata-rata. Output hasil pooling kemudian diproses oleh lapisan fully connected untuk menghasilkan prediksi. Bagian bawah kode mendefinisikan konfigurasi eksperimen dengan parameter seperti ukuran unit tersembunyi, jenis pooling, jumlah epoch, dan optimizer yang akan diuji. Model ini dirancang untuk mengeksplorasi kombinasi hyperparameter terbaik, dengan menyimpan akurasi validasi tertinggi di best_accuracy dan konfigurasi terbaik di best_config. Implementasi ini menunjukkan fleksibilitas dan kekuatan Deep RNN dalam menangani tugas data sekuensial yang lebih kompleks.

Output

```
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=SGD
Epoch [1/5], Loss: 1.6931
Epoch [2/5], Loss: 1.9043
Epoch [3/5], Loss: 1.5013
Epoch [4/5], Loss: 1.3685
Epoch [5/5], Loss: 1.8519
Accuracy: 0.3378
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=RMSProp
Epoch [1/5], Loss: 1.3762
Epoch [2/5], Loss: 1.5349
Epoch [3/5], Loss: 1.5601
Epoch [4/5], Loss: 1.2270
Epoch [5/5], Loss: 1.4294
Accuracy: 0.3561
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=Adam
Epoch [1/5], Loss: 0.9689
Epoch [2/5], Loss: 1.4398
Epoch [3/5], Loss: 1.8197
Epoch [4/5], Loss: 1.3763
Epoch [5/5], Loss: 1.6969
Accuracy: 0.3439
Training with hidden_size=16, pooling_type=max, epochs=50, optimizer=SGD
Epoch [1/50], Loss: 1.7442
Epoch [2/50], Loss: 1.5949
Epoch [3/50], Loss: 1.8347
...
Epoch [248/250], Loss: 1.5971
Epoch [249/250], Loss: 1.4867
Epoch [250/250], Loss: 1.3545
Accuracy: 0.3622
```

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

```
Best Configuration: {'hidden_size': 64, 'pooling_type': 'avg', 'epochs': 250, 'optimizer': 'RMSProp'}
Best Accuracy: 0.4091836734693878
```

Analisis Hasil

Hasil eksperimen menunjukkan bahwa konfigurasi hyperparameter terbaik untuk model Deep RNN adalah:

- hidden_size: 64
  Ukuran unit tersembunyi sebesar 64 menghasilkan representasi fitur yang cukup baik, memungkinkan model menangkap pola kompleks dalam data sekuensial.
- pooling_type: 'avg'
  Jenis pooling terbaik adalah average pooling, menunjukkan bahwa menghitung rata-rata dari output rekuren sepanjang dimensi waktu lebih efektif daripada hanya mengambil nilai maksimum (max pooling) dalam konteks dataset ini.
- epochs: 250
  Jumlah epoch yang cukup besar diperlukan untuk mencapai akurasi terbaik. Hal ini menunjukkan bahwa model memerlukan pelatihan yang lebih lama untuk mencapai konvergensi pada dataset tertentu.
- optimizer: 'RMSProp'
  Optimizer RMSProp memberikan performa terbaik dibandingkan alternatif seperti

SGD atau Adam, kemungkinan karena kemampuannya dalam menangani gradien yang tidak stabil pada data sekuensial.

Namun, akurasi validasi terbaik hanya mencapai 40.91%, yang menunjukkan bahwa performa model masih belum memadai untuk menangani dataset. Hal ini mungkin disebabkan oleh beberapa faktor:

1. Keterbatasan Arsitektur Deep RNN:
    - Meskipun memiliki beberapa lapisan rekuren, RNN standar masih mengalami kesulitan dalam menangkap dependensi jangka panjang karena masalah vanishing gradient.
2. Dataset yang Kompleks:
    - Jika dataset memiliki pola yang sangat kompleks, RNN mungkin tidak cukup untuk menangkapnya. Model seperti LSTM atau GRU lebih cocok untuk menangani data seperti ini.
3. Overfitting atau Underfitting:
    - Dengan 250 epoch, ada kemungkinan model mengalami overfitting jika dataset pelatihan kecil atau underfitting jika arsitektur tidak cukup kompleks.

Kesimpulan

Eksperimen ini menunjukkan bahwa kombinasi `hidden_size` = 64, `pooling_type` = 'avg', `epochs` = 250, dan `optimizer` = 'RMSProp' menghasilkan akurasi validasi terbaik sebesar 40.91%. Namun, performa ini masih tergolong rendah, yang mengindikasikan bahwa arsitektur Deep RNN standar kurang optimal untuk dataset ini.