LAPORAN

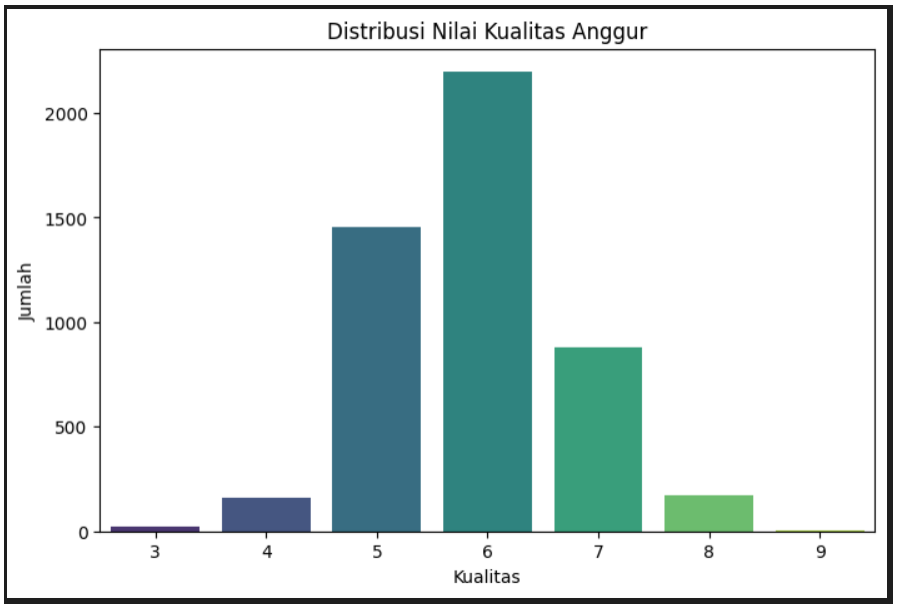
MACHINE LEARNING

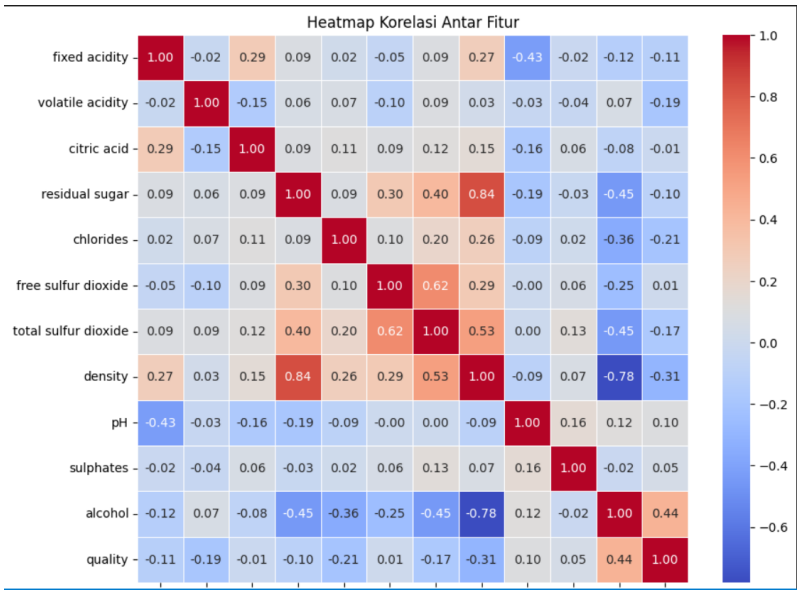
WEEK 14

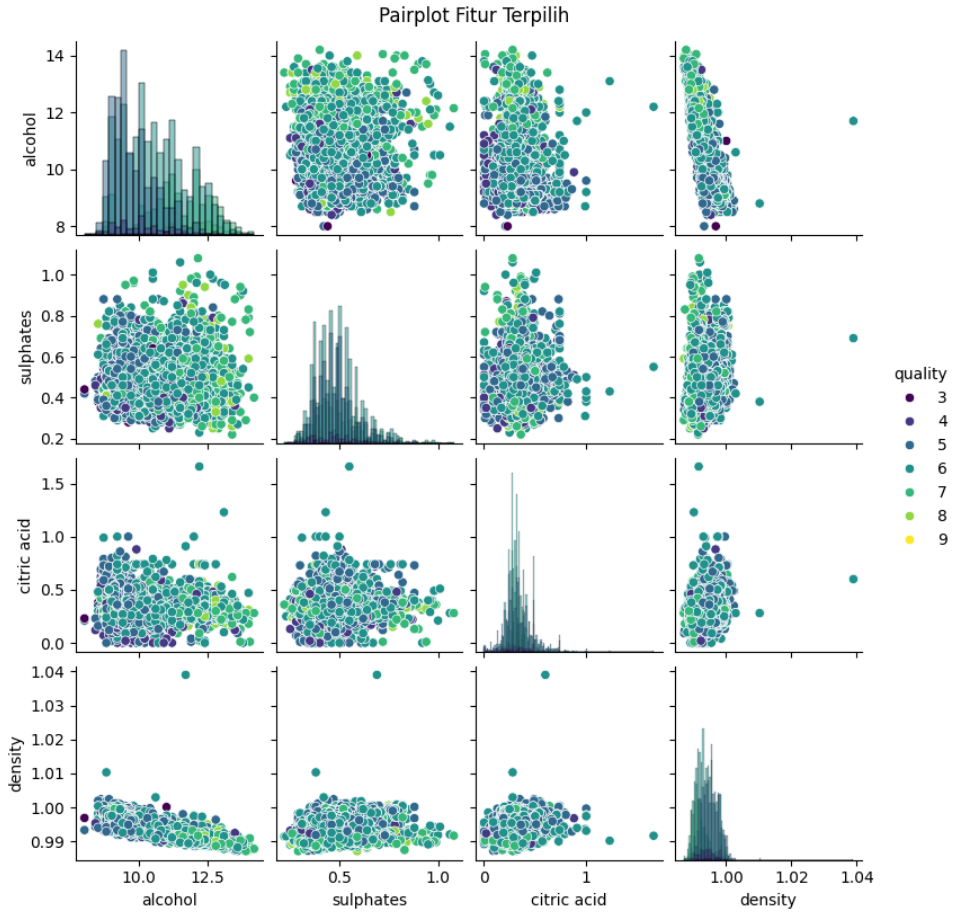
Bidirectional RNN model dataset Wine Quality

Hidden Markov Model (HMM) adalah pengembangan dari Markov Model, di mana keadaan (state) sistem tidak dapat diamati secara langsung (tersembunyi) tetapi hanya dapat diinferensi melalui observasi.

Exploratory Data Analysis (EDA)








```python
# Define the Bidirectional RNN model
class BiRNNModel(nn.Module):
    def __init__(self, input_size, hidden_size, output_size, pooling_type='max'):
        super(BiRNNModel, self).__init__()
        self.rnn = nn.RNN(input_size, hidden_size, batch_first=True, bidirectional=True)
        self.fc = nn.Linear(hidden_size * 2, 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, 250,350]
optimizers = {'SGD': optim.SGD, 'RMSProp': optim.RMSprop, 'Adam': optim.Adam}

best_accuracy = 0
best_config = {}
```

implementasi model Bidirectional Recurrent Neural Network (BiRNN) menggunakan PyTorch. Kelas BiRNNModel dirancang untuk memanfaatkan informasi dari kedua arah dalam data sekuensial, yaitu maju (forward) dan mundur (backward), melalui parameter bidirectional=True pada nn.RNN. Pada konstruktor (__init__), model menerima parameter seperti input_size (ukuran input), hidden_size (ukuran hidden state), output_size (ukuran output), dan pooling_type (jenis pooling). Karena model bersifat bidirectional, ukuran hidden state dalam lapisan fully connected (nn.Linear) dikalikan dua untuk menggabungkan informasi dari kedua arah.

Metode forward menentukan aliran data melalui model. Data input diproses oleh lapisan RNN, menghasilkan output dari kedua arah. Pooling diterapkan untuk mereduksi dimensi output, dengan dua opsi: max pooling untuk mengambil nilai maksimum sepanjang dimensi waktu atau average pooling untuk menghitung rata-rata. Output dari pooling kemudian diteruskan ke lapisan fully connected untuk menghasilkan prediksi akhir.

Di bagian bawah kode, konfigurasi eksperimen mencakup:

1. Ukuran Hidden State: 16, 32, dan 64 untuk mengevaluasi kapasitas model.
2. Jenis Pooling: max dan avg untuk memilih mekanisme pooling terbaik.
3. Jumlah Epoch: 5, 50, 100, 250, dan 350 untuk menentukan durasi pelatihan optimal.
4. Optimizer: SGD, RMSProp, dan Adam untuk menentukan metode pembelajaran terbaik.

Model ini dirancang untuk mengeksplorasi kombinasi hyperparameter yang optimal, dengan menyimpan hasil terbaik dalam variabel best_accuracy dan konfigurasi terbaik di best_config. Dengan sifat bidirectional, BiRNN memiliki keunggulan dalam menangkap konteks penuh dari data sekuensial, menjadikannya sangat cocok untuk tugas-tugas seperti pemrosesan bahasa alami (NLP) atau pengenalan pola dalam data sekuensial yang kompleks. Kode ini memberikan fleksibilitas tinggi dalam eksperimen untuk meningkatkan performa model.

Output

```
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=SGD
Epoch [1/5], Loss: 1.7279
Epoch [2/5], Loss: 1.6351
Epoch [3/5], Loss: 1.5742
Epoch [4/5], Loss: 1.4095
Epoch [5/5], Loss: 1.2997
Accuracy: 0.3551
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=RMSProp
Epoch [1/5], Loss: 1.1387
Epoch [2/5], Loss: 1.1893
Epoch [3/5], Loss: 1.5533
Epoch [4/5], Loss: 1.3104
Epoch [5/5], Loss: 1.2724
Accuracy: 0.3612
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=Adam
Epoch [1/5], Loss: 1.7142
Epoch [2/5], Loss: 0.9761
Epoch [3/5], Loss: 1.1023
Epoch [4/5], Loss: 1.5235
Epoch [5/5], Loss: 1.3187
Accuracy: 0.3735
Training with hidden_size=16, pooling_type=max, epochs=50, optimizer=SGD
Epoch [1/50], Loss: 1.7019
Epoch [2/50], Loss: 1.5994
Epoch [3/50], Loss: 1.5077
...
Epoch [348/350], Loss: 1.3223
Epoch [349/350], Loss: 0.9110
Epoch [350/350], Loss: 1.3972
Accuracy: 0.3653
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

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

```
Best Configuration: {'hidden_size': 32, 'pooling_type': 'avg', 'epochs': 350, 'optimizer': 'Adam'}
Best Accuracy: 0.386734693877551
```

Hasil eksperimen menunjukkan bahwa konfigurasi terbaik untuk model Bidirectional RNN adalah:

- hidden_size: 32
  Ukuran hidden state sebesar 32 memberikan keseimbangan antara kapasitas model dan kebutuhan komputasi. Ukuran ini cukup untuk menangkap pola dalam dataset tanpa overfitting.
- pooling_type: 'avg'
  Average pooling menghasilkan hasil terbaik, menunjukkan bahwa rata-rata hidden state sepanjang dimensi waktu lebih efektif dibandingkan dengan max pooling.
- epochs: 350
  Model memerlukan 350 epoch untuk mencapai performa terbaik, yang menunjukkan kebutuhan pelatihan yang lebih panjang untuk menyerap pola dalam dataset.

- optimizer: 'Adam'
  Optimizer Adam memberikan performa terbaik, menunjukkan bahwa metode pembelajaran adaptif ini cocok untuk data dan arsitektur yang digunakan.

Akurasi validasi terbaik yang dicapai adalah 38.67%, yang menunjukkan bahwa model memiliki keterbatasan dalam memahami pola dalam dataset.

Analisis Hasil

1. **Kinerja Model yang Rendah:**
   - Meskipun menggunakan model Bidirectional RNN, akurasi validasi sebesar 38.67% masih tergolong rendah. Hal ini mengindikasikan bahwa model kesulitan menangkap hubungan kompleks dalam dataset atau bahwa dataset memerlukan representasi yang lebih baik.
2. **Pengaruh Pooling:**
   - Average pooling menghasilkan hasil terbaik. Ini menunjukkan bahwa pengambilan rata-rata hidden state lebih baik dalam menyaring informasi relevan dibandingkan max pooling, yang hanya mempertimbangkan nilai maksimum.
3. **Jumlah Epoch yang Tinggi:**
   - Dengan 350 epoch, model membutuhkan waktu pelatihan yang panjang untuk mencapai konvergensi. Ini menunjukkan bahwa pola dalam dataset sulit dipelajari, sehingga membutuhkan banyak iterasi.
4. **Kinerja Optimizer Adam:**
   - Adam unggul dibandingkan optimizer lain seperti SGD dan RMSProp. Optimizer ini mampu menyesuaikan learning rate selama pelatihan, yang membantu mempercepat konvergensi.

Kesimpulan

Eksperimen menunjukkan bahwa konfigurasi dengan hidden_size = 32, pooling_type = 'avg', epochs = 350, dan optimizer Adam memberikan performa terbaik dengan akurasi validasi 38.67%. Meskipun menggunakan Bidirectional RNN yang dirancang untuk menangkap pola dalam kedua arah (forward dan backward), hasilnya masih tidak memadai untuk dataset ini.

Rekomendasi untuk Peningkatan:

1. **Gunakan Model yang Lebih Kompleks:**
   Cobalah arsitektur seperti **Bidirectional LSTM** atau **GRU**, yang lebih efektif dalam menangkap hubungan jangka panjang dibandingkan RNN.
2. **Perbaikan Preprocessing Data:**
   Pastikan data telah di-normalisasi dan lakukan eksplorasi fitur tambahan untuk meningkatkan representasi data.

3. **Tuning Hyperparameter Lebih Lanjut:**
   Eksplorasi ukuran batch, learning rate, atau penggunaan dropout untuk mengurangi overfitting.
4. **Penambahan Fitur Eksternal:**
   Jika dataset memiliki informasi tambahan yang dapat digunakan, seperti embedding atau metadata, masukkan ke dalam model.

Hasil ini menunjukkan pentingnya mengombinasikan arsitektur yang lebih canggih dan preprocessing yang optimal untuk mencapai performa yang lebih tinggi. Bidirectional RNN memberikan hasil yang lebih baik dibandingkan RNN biasa, tetapi belum cukup untuk menangani dataset dengan pola yang kompleks.