

LAPORAN

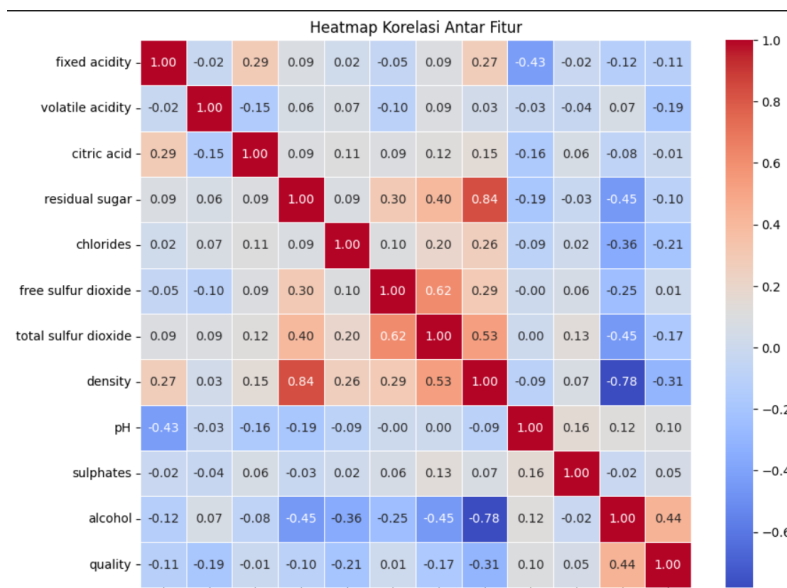
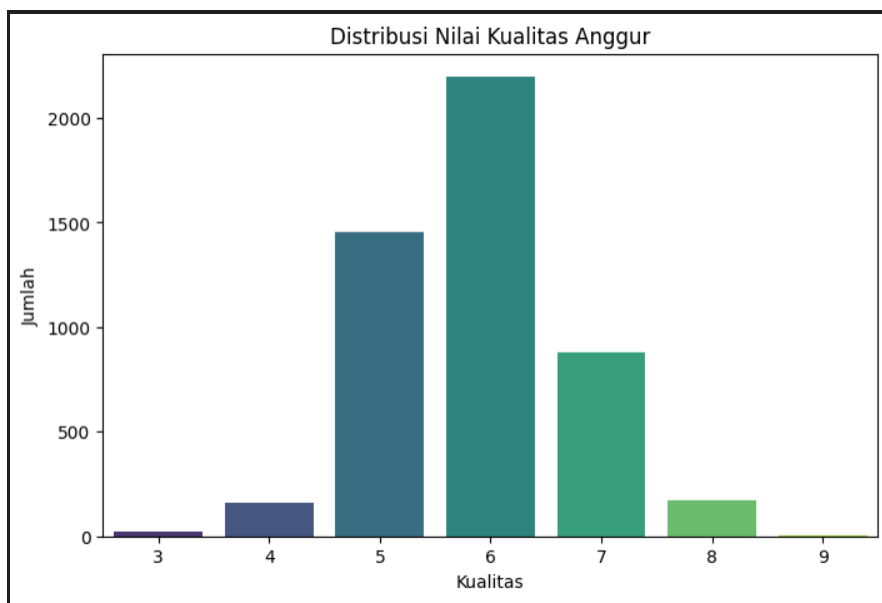
MACHINE LEARNING

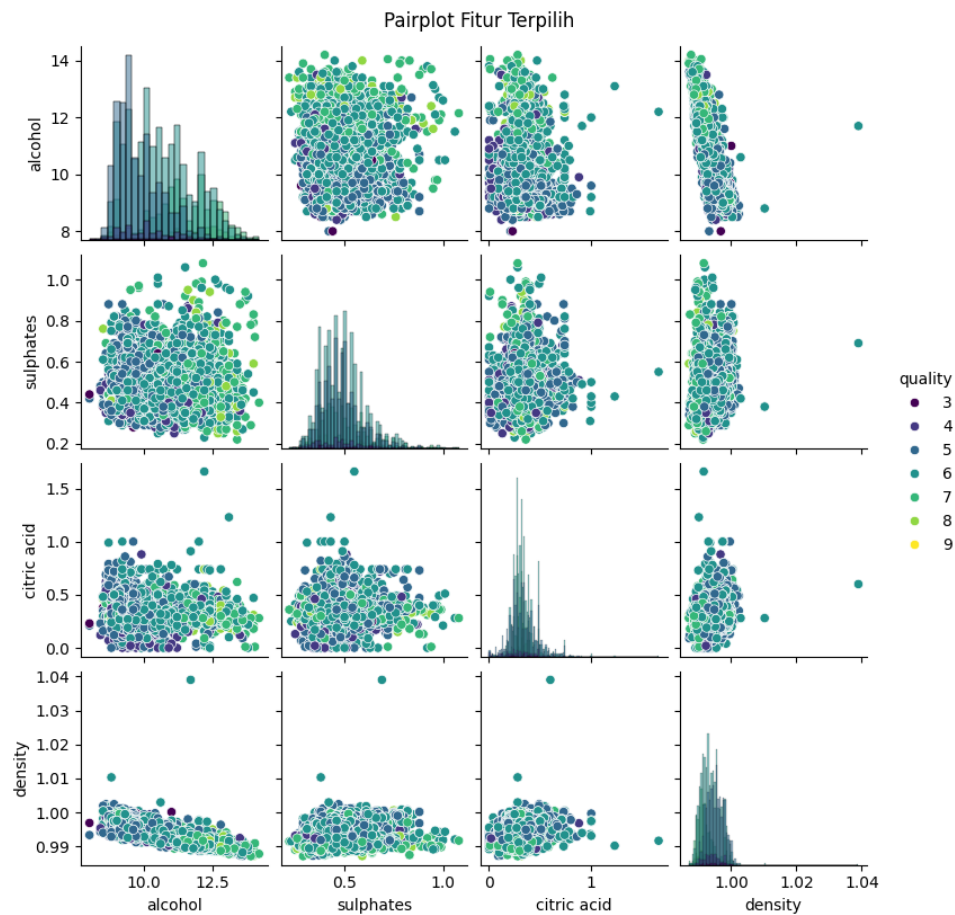
WEEK 14

Hidden Markov 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)





```
# Define the HMM with RNN-based emission probabilities
class RNNHMMModel(nn.Module):
    def __init__(self, input_size, hidden_size, output_size, pooling_type='max'):
        super(RNNHMMModel, 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 RNN-HMM (Hidden Markov Model berbasis RNN) menggunakan PyTorch. Kelas RNNHMMModel dirancang untuk menggabungkan prinsip Hidden Markov Model (HMM) dengan kekuatan pemrosesan sekuensial dari Recurrent Neural Network (RNN). Pada metode `__init__`, model memanfaatkan `nn.RNN` untuk memproses data sekuensial dan `nn.Linear` untuk memproyeksikan hidden state menjadi output akhir. Parameter seperti `input_size`, `hidden_size`, dan `output_size` memungkinkan fleksibilitas dalam mengatur ukuran input, unit tersembunyi, dan output sesuai kebutuhan. Model juga mendukung jenis pooling, yaitu max dan avg, yang ditentukan oleh parameter `pooling_type`.

Metode `forward` mendefinisikan bagaimana data mengalir melalui model. Data masukan diproses oleh lapisan RNN, menghasilkan hidden state untuk setiap langkah waktu. Hidden state kemudian diringkas menggunakan pooling—baik max pooling untuk mengambil nilai maksimum sepanjang dimensi waktu atau average pooling untuk menghitung rata-rata. Output dari pooling diteruskan ke lapisan fully connected untuk menghasilkan prediksi akhir. Bagian bawah kode mendefinisikan konfigurasi eksperimen dengan berbagai hyperparameter, seperti ukuran hidden state (`hidden_sizes`), jenis pooling (`pooling_types`), jumlah epoch (`epochs_list`), dan optimizer (`optimizers`).

Model ini dirancang untuk mengeksplorasi kombinasi hyperparameter yang optimal. Dengan menggabungkan kekuatan RNN untuk pemrosesan data sekuensial dan prinsip HMM, model ini cocok untuk tugas-tugas seperti pengenalan pola dalam data sekuensial, seperti teks atau sinyal waktu. Kombinasi pooling dan konfigurasi fleksibel meningkatkan kemampuan model untuk menyesuaikan dengan berbagai dataset dan skenario.

Output

```
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=SGD
Epoch [1/5], Loss: 1.6271
Epoch [2/5], Loss: 1.6925
Epoch [3/5], Loss: 2.0466
Epoch [4/5], Loss: 1.4471
Epoch [5/5], Loss: 1.5759
Accuracy: 0.3714
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=RMSProp
Epoch [1/5], Loss: 0.9787
Epoch [2/5], Loss: 1.1915
Epoch [3/5], Loss: 1.4631
Epoch [4/5], Loss: 1.5429
Epoch [5/5], Loss: 1.1901
Accuracy: 0.3673
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=Adam
Epoch [1/5], Loss: 1.5414
Epoch [2/5], Loss: 1.2320
Epoch [3/5], Loss: 1.6357
Epoch [4/5], Loss: 1.1019
Epoch [5/5], Loss: 1.0550
Accuracy: 0.3541
Training with hidden_size=16, pooling_type=max, epochs=50, optimizer=SGD
Epoch [1/50], Loss: 1.7447
Epoch [2/50], Loss: 1.6394
Epoch [3/50], Loss: 1.6132
...
Epoch [248/250], Loss: 2.2459
Epoch [249/250], Loss: 1.1847
Epoch [250/250], Loss: 1.1915
Accuracy: 0.3724
```

```
# 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': 250, 'optimizer': 'Adam'}
Best Accuracy: 0.386734693877551
```

Hasil eksperimen menunjukkan konfigurasi hyperparameter terbaik untuk model RNN-HMM adalah:

- **hidden_size: 32**
Ukuran hidden state sebesar 32 memungkinkan model menangkap pola yang cukup kompleks dalam data sekuensial tanpa menyebabkan overfitting atau computational overhead yang terlalu besar.
- **pooling_type: 'avg'**
Average pooling memberikan hasil terbaik, yang menunjukkan bahwa rata-rata nilai hidden state sepanjang dimensi waktu lebih efektif dalam mengekstraksi informasi penting dibandingkan dengan max pooling.
- **epochs: 250**
Model membutuhkan 250 epoch untuk mencapai performa terbaik, yang menunjukkan bahwa pelatihan yang lebih lama diperlukan untuk menangkap pola secara optimal.
- **optimizer: 'Adam'**
Optimizer Adam memberikan hasil terbaik, menunjukkan bahwa mekanisme pembelajaran dinamis Adam cocok untuk data dan arsitektur ini.

Akurasi validasi terbaik yang dicapai adalah 38.67%, yang tergolong rendah. Hal ini menunjukkan bahwa model masih memiliki keterbatasan dalam menangkap pola pada dataset ini.

Kesimpulan

Eksperimen ini menunjukkan bahwa model RNN-HMM dengan konfigurasi `hidden_size = 32`, `pooling_type = 'avg'`, `epochs = 250`, dan optimizer Adam memberikan performa terbaik dengan akurasi validasi sebesar 38.67%. Namun, akurasi ini masih jauh dari memadai, menunjukkan bahwa pendekatan ini tidak cukup kuat untuk menangkap pola dalam dataset yang diuji.