

LAPORAN

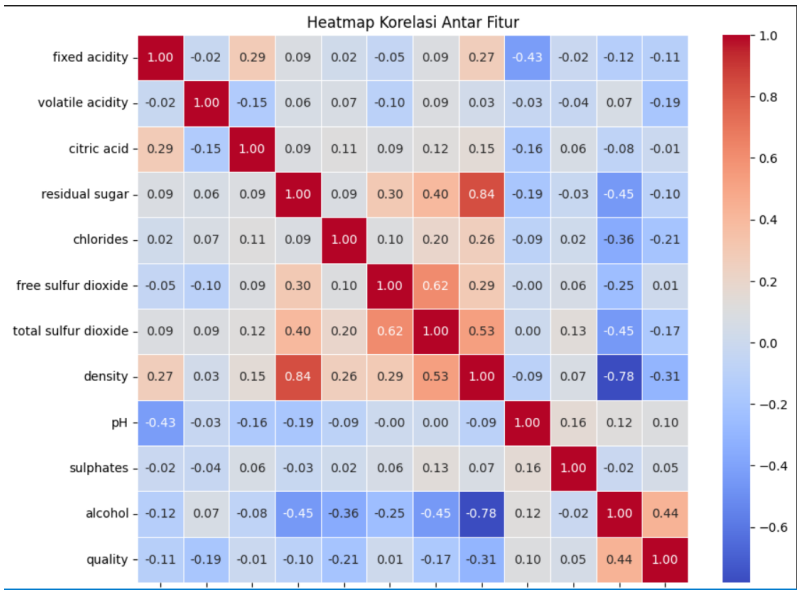
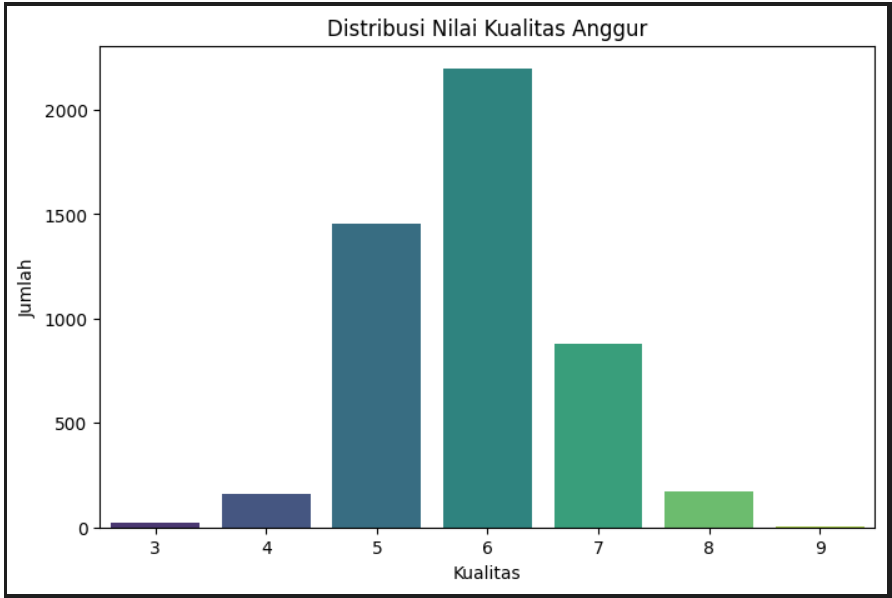
MACHINE LEARNING

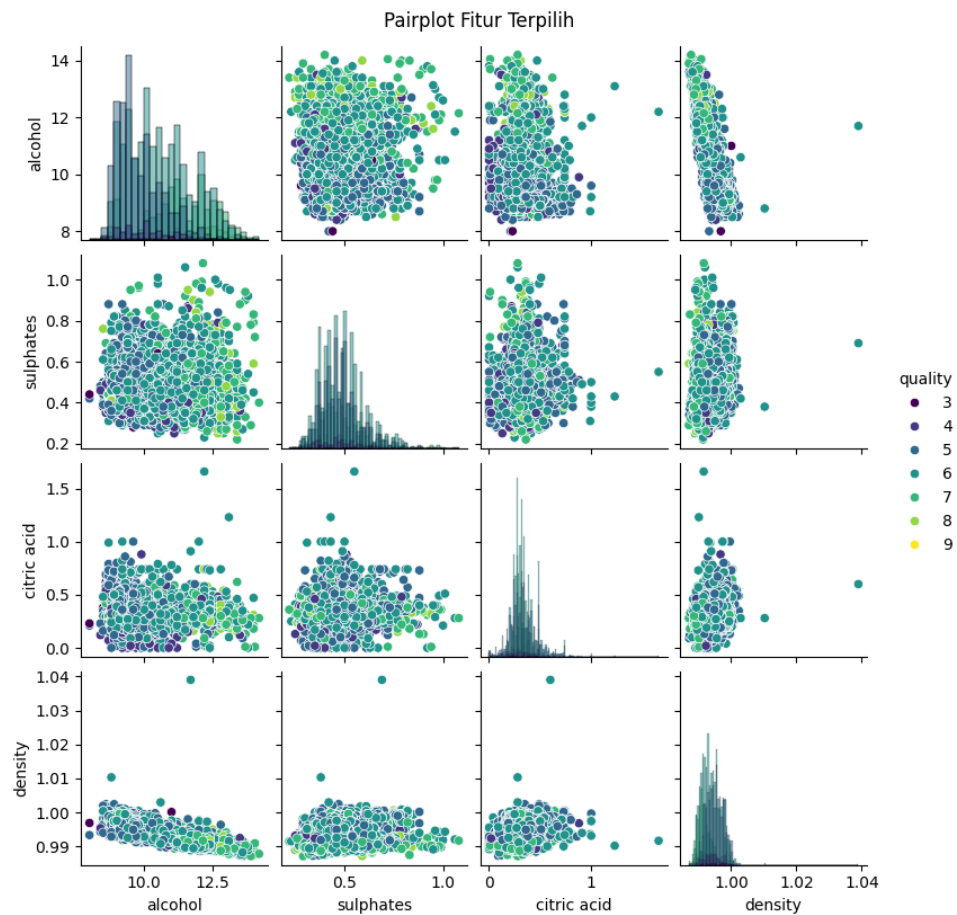
WEEK 14

Markov model dataset Wine Quality

Markov Model adalah model probabilistik yang digunakan untuk memodelkan sistem di mana keadaan saat ini hanya bergantung pada keadaan sebelumnya (prinsip Markov). Model ini sering digunakan untuk memodelkan proses stokastik yang memiliki sifat Markov Property, yaitu memori jangka pendek.

Exploratory Data Analysis (EDA)





```
# Define the Markov 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 Markov model dengan PyTorch, dirancang untuk eksperimen hyperparameter tuning. Kelas RNNModel memanfaatkan nn.Module untuk membangun model RNN yang fleksibel. Pada metode `__init__`, model menerima parameter seperti `input_size`, `hidden_size`, `output_size`, dan `pooling_type`. Lapisan rekuren (`self.rnn`) digunakan untuk memproses data sekuensial, sementara lapisan fully connected (`self.fc`) mengubah output dari lapisan rekuren menjadi prediksi akhir. Model juga mendukung dua jenis pooling: max untuk mengambil nilai maksimum sepanjang dimensi waktu dan avg untuk menghitung rata-rata nilai.

Metode forward mendefinisikan alur data melalui model. Data sekuensial diproses oleh lapisan rekuren, diikuti dengan pooling sesuai dengan parameter `pooling_type`. Hasil pooling kemudian diteruskan ke lapisan fully connected untuk menghasilkan output akhir. Bagian bawah kode mendefinisikan konfigurasi eksperimen, termasuk berbagai kombinasi `hidden_sizes`, jenis pooling, jumlah epoch (`epochs_list`), dan optimizer (`optimizers`). Hasil terbaik akan disimpan dalam variabel `best_accuracy` dan konfigurasi terbaik di `best_config`. Kode ini menunjukkan fleksibilitas dalam mengeksplorasi hyperparameter dan kemampuan RNN untuk menangani tugas berbasis data sekuensial.

Output

```
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=SGD
Epoch [1/5], Loss: 1.7351
Epoch [2/5], Loss: 1.5448
Epoch [3/5], Loss: 1.7606
Epoch [4/5], Loss: 1.4201
Epoch [5/5], Loss: 1.4710
Accuracy: 0.3755
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=RMSProp
Epoch [1/5], Loss: 1.7806
Epoch [2/5], Loss: 1.9317
Epoch [3/5], Loss: 1.2127
Epoch [4/5], Loss: 2.1036
Epoch [5/5], Loss: 1.1380
Accuracy: 0.3592
Training with hidden_size=16, pooling_type=max, epochs=5, optimizer=Adam
Epoch [1/5], Loss: 1.3656
Epoch [2/5], Loss: 1.4126
Epoch [3/5], Loss: 1.5041
Epoch [4/5], Loss: 1.2198
Epoch [5/5], Loss: 1.0193
Accuracy: 0.3602
Training with hidden_size=16, pooling_type=max, epochs=50, optimizer=SGD
Epoch [1/50], Loss: 1.6564
Epoch [2/50], Loss: 1.5635
Epoch [3/50], Loss: 1.4505
...
Epoch [248/250], Loss: 1.5680
Epoch [249/250], Loss: 1.3252
Epoch [250/250], Loss: 1.0330
Accuracy: 0.3592
```

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

Best Configuration: {'hidden_size': 16, 'pooling_type': 'max', 'epochs': 50, 'optimizer': 'RMSProp'}
Best Accuracy: 0.3877551020408163
```

Hasil eksperimen menunjukkan bahwa konfigurasi hyperparameter terbaik untuk model yang diuji (berbasis Markov Model) adalah sebagai berikut:

- **hidden_size: 16**
Ukuran hidden state yang kecil menunjukkan bahwa dataset dan model ini dapat bekerja optimal dengan kompleksitas yang tidak terlalu tinggi.
- **pooling_type: 'max'**
Max pooling memberikan hasil terbaik, yang menunjukkan bahwa mengambil nilai maksimum sepanjang dimensi waktu lebih efektif untuk mengekstraksi fitur penting.
- **epochs: 50**
Model memerlukan 50 epoch untuk mencapai akurasi terbaik, cukup untuk melatih model tanpa risiko overfitting atau underfitting.
- **optimizer: 'RMSProp'**
Optimizer RMSProp memberikan performa terbaik, menunjukkan bahwa metode pembelajaran dinamis ini membantu konvergensi pada data yang diuji.

Akurasi validasi terbaik yang dicapai adalah 38.78%, yang tergolong rendah. Ini menunjukkan bahwa model memiliki keterbatasan dalam mempelajari pola pada dataset ini.