Tugas Pertemuan 7_Andrean Yonathan_Institut Teknologi Sepuluh Nopember

Latihan(1)

import library yg dibutuhkan

In [1]:
```
# import pandas
import pandas as pd
# import numpy
import numpy as np
# import library SelectKBest
from sklearn.feature_selection import SelectKBest
# import library chi kuadrat/squared
from sklearn.feature_selection import chi2
```

In [2]:
```
# load dataset
data = pd.read_csv('Iris.csv')
data
```

Out[2]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

Latihan(2)

buat dataframe tanpa kolom 'Id' yang ditampung dalam variabel bernama df1, lalu tampilkan

In [3]:
```
# Menghilangkan kolom Id
df1 = data.drop(columns = 'Id', axis = 1)
# lalu tampilkan
df1
```

Out[3]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

Latihan(3)

- Buat variabel independent columns dan target kedalam variabel X dan y

In [4]:
```
#independent columns --> SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm
X = df1.iloc[:, 0:4]
# target columns --> species
y = df1.iloc[:, -1]
```

Latihan(4)

- Aplikasikan library **SelectKBest** untuk mengekstrak fitur terbaik dari dataset

In [5]:
```
#Apply SelectKBest class to extract

bestfeature = SelectKBest(score_func=chi2, k=4)
fit = bestfeature.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
```

Latihan(5)

- lihat hasil score seleksi feature

In [6]:
```
#gabungkan 2 dataframe tersebut untuk visualisasi yang lebih bagus

featureScores = pd.concat([dfcolumns, dfscores],axis=1)
featureScores.columns = ['Field', 'Score']
print(featureScores.nlargest(10,'Score'))
```

```
           Field       Score
2  PetalLengthCm  116.169847
3   PetalWidthCm   67.244828
0  SepalLengthCm   10.817821
1   SepalWidthCm    3.594499
```

Latihan(6)

buat dataframe tanpa kolom 'Id' yang ditampung dalam variabel bernama df2, lalu tampilkan

In [7]:
```
data = pd.read_csv('Iris.csv')

# Menghilangkan kolom Id
df2 = data.drop(columns = 'Id', axis = 1)
# lalu tampilkan
df2
```

Out[7]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

Latihan(7)

- Buat variabel independent columns dan target kedalam variabel A dan b

In [8]:
```
#independent columns --> SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm
A = df2.iloc[:, :4]
# target columns --> species
b = df2.iloc[:, -1]
```

Latihan(8)

Tujuan dari **ExtraTreesClassifier** adalah untuk menyesuaikan sejumlah pohon keputusan acak ke data. dan dalam hal ini adalah dari pembelajaran ensemble. Khususnya, pemisahan acak dari semua pengamatan dilakukan untuk memastikan bahwa model tidak terlalu cocok dengan data.

- Aplikasikan library **ExtraTreesClassifier** untuk mengekstrak fitur terbaik dari dataset

In [9]:
```
# Import library ExtraTreesClassifier
from sklearn.ensemble import ExtraTreesClassifier
# Import library matplotlib
import matplotlib.pyplot as plt

# fit model ExtraTreesClassifier
model = ExtraTreesClassifier()
model.fit(A,b)
```

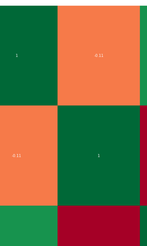
Out[9]:
```
ExtraTreesClassifier()
```

Latihan(9)

- visualisasikan hasil dari model ExtraTreesClassifier

In [10]:
```
print(model.feature_importances_)
feat_importance = pd.Series(model.feature_importances_, index=A.columns)
feat_importance.nlargest(10).plot(kind='barh')
plt.show()
```

```
[0.10360325 0.06302444 0.37559577 0.45777654]
```



In [11]:
```
import pandas as pd
import numpy as np
import seaborn as sns

data = pd.read_csv('Iris.csv')
df3= data.iloc[:,1:]
df3
```

Out[11]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

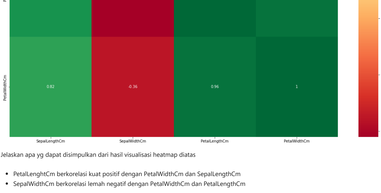
Latihan(10)

- Buat variabel independent columns dan target kedalam variabel K dan J
- hitung korelasi setiap fitur
- visualisasikan hasil dari Matriks Korelasi dengan Heatmap

In [12]:
```
#independent columns --> SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm
K = df3.iloc[:, :4]
# target columns --> species
J = df3.iloc[:, -1]

# mendapatkan korelasi di setiap fitur dalam dataset
corrmat = df3.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(20,20))

# plot heatmap
h = sns.heatmap(df3[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```



Jelaskan apa yg dapat disimpulkan dari hasil visualisasi heatmap diatas

- PetalLenghtCm berkorelasi kuat positif dengan PetalWidthCm dan SepalLengthCm
- SepalWidthCm berkorelasi lemah negatif dengan PetalWidthCm dan PetalLengthCm