



Hands-On

Hands-On ini digunakan pada kegiatan Microcredential Associate Data Scientist 2021

Pertemuan 7

Pertemuan 7 (tujuh) pada Microcredential Associate Data Scientist 2021 menyampaikan materi mengenai Menentukan Objek atau Memilih Data atau Seleksi Fitur

iris setosa



petal sepal

iris versicolor



petal sepal

iris virginica



petal sepal

Info dataset: Dataset ini berisi ukuran/measures

3 spesies iris

Seleksi Univariat

Uji statistik dapat digunakan utk memilih fitur-fitur tsb yang memiliki relasi paling kuat dengan variabel output

Scikit-learn API menyediakan kelas **SelectKBest** untuk mengekstrak fitur terbaik dari dataset yang diberikan. Metode SelectKBest memilih fitur sesuai dengan k skor tertinggi. Dengan

mengubah parameter 'score_func' kita dapat menerapkan metode untuk data klasifikasi dan regresi. Memilih fitur terbaik adalah proses penting ketika kami menyiapkan kumpulan data besar untuk pelatihan. Ini membantu kami menghilangkan bagian data yang kurang penting dan mengurangi waktu pelatihan.

chi-kuadrat ini dapat digunakan untuk memilih `n_fitur` fitur dengan nilai tertinggi untuk uji statistik chi-kuadrat dari X, yang harus berisi hanya fitur non-negatif seperti boolean atau frekuensi (misalnya, jumlah istilah dalam klasifikasi dokumen), relatif terhadap kelas.

Ingatlah bahwa uji chi-kuadrat mengukur ketergantungan antara variabel stokastik, jadi menggunakan fungsi ini "menyingkirkan" fitur-fitur yang paling mungkin tidak bergantung pada kelas dan oleh karena itu tidak relevan untuk klasifikasi.

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.head()
```

```
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

Latihan(2)

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

```
In [3]: # Menghilangkan kolom Id
data = data.drop('Id',1)
```

```
In [9]: # lalu tampilkan
data.head()

data['Species'].unique()
```

```
Out[9]: array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

```
In [11]: # Import Label encoder
from sklearn import preprocessing

# Label_encoder object knows how to understand word Labels.
label_encoder = preprocessing.LabelEncoder()

# Encode Labels in column 'Species'.
data['Species'] = label_encoder.fit_transform(data['Species'])

data['Species'].unique()
```

```
Out[11]: array([0, 1, 2])
```

```
In [13]: data.head()
```

```
Out[13]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Latihan(3)

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

```
In [33]: #independent columns --> SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm
X = data.iloc[:,0:20]
# target columns --> species
y = data.iloc[:, -1]
```

Latihan(4)

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

```
In [17]: #Apply SelectKBest class to extract
bestfeature = SelectKBest(score_func=chi2, k=4)
fit = bestfeature.fit(X,y)
datascores = pd.DataFrame(fit.scores_)
datacolumns = pd.DataFrame(X.columns)
```

Latihan(5)

- lihat hasil score seleksi feature

```
In [19]: #gabungkan 2 dataframe tersebut untuk visualisasi yang lebih bagus

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

	Field	Score
2	PetalLengthCm	116.169847
4	Species	100.000000
3	PetalWidthCm	67.244828
0	SepalLengthCm	10.817821
1	SepalWidthCm	3.594499

Feature Importance (FT)

FT berfungsi memberi skor untuk setiap fitur data, semakin tinggi skor semakin penting atau relevan fitur tersebut terhadap variabel output

FT merupakan kelas inbuilt yang dilengkapi dengan Pengklasifikasi Berbasis Pohon (Tree Based Classifier), kita akan menggunakan Pengklasifikasi Pohon Ekstra untuk mengekstraksi 10 fitur teratas untuk kumpulan data

Latihan(6)

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

```
In [29]: data2 = pd.read_csv('Iris.csv')
```

```
# Menghilangkan kolom Id
data2 = data2.drop('Id',1)
# lalu tampilkan
data2.head()
```

Out[29]:

	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

```
In [30]: data2['Species'].unique()
```

Out[30]: array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)

```
In [36]: # Import Label encoder
from sklearn import preprocessing

# Label_encoder object knows how to understand word Labels.
label_encoder = preprocessing.LabelEncoder()

# Encode Labels in column 'Species'.
data2['Species'] = label_encoder.fit_transform(data2['Species'])

data2['Species'].unique()
```

Out[36]: array([0, 1, 2], dtype=int64)

```
In [37]: data2.head()
```

Out[37]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Latihan(7)

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

```
In [38]: #independent columns --> SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm
A = data2.iloc[:,0:20]
# target columns --> species
b = data2.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 [39]: # 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[39]: ExtraTreesClassifier()
```

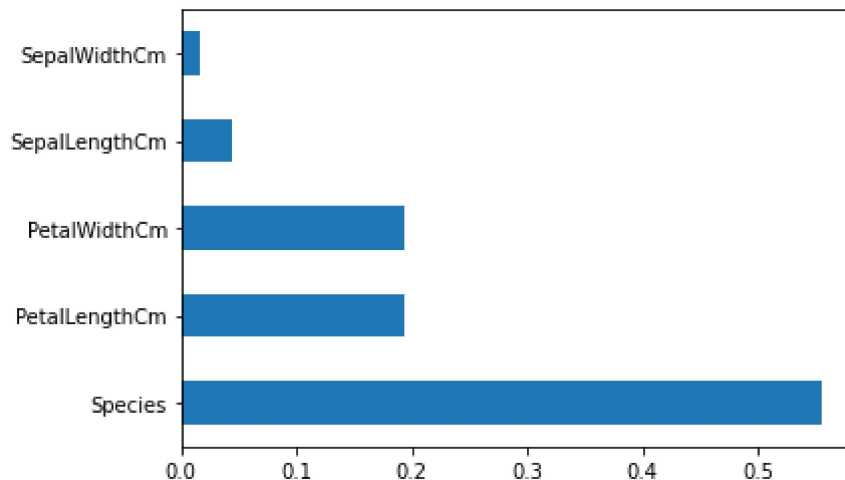
Latihan(9)

- visualisasikan hasil dari model ExtraTreesClassifier

```
In [40]: print(model.feature_importances_)
feat_importance = pd.Series(model.feature_importances_, index=A.columns)
feat_importance.nlargest(10).plot(kind='barh')
plt.show
```

```
[0.04412787 0.0151417 0.19322433 0.19265069 0.55485541]
```

```
Out[40]: <function matplotlib.pyplot.show(close=None, block=None)>
```



Matriks Korelasi dengan Heatmap

- Korelasi menyatakan bagaimana fitur terkait satu sama lain atau variabel target.
- Korelasi bisa positif (kenaikan satu nilai fitur meningkatkan nilai variabel target) atau negatif (kenaikan satu nilai fitur menurunkan nilai variabel target)
- Heatmap memudahkan untuk mengidentifikasi fitur mana yang paling terkait dengan variabel target, kami akan memplot peta panas fitur yang berkorelasi menggunakan seaborn library

```
In [41]: import pandas as pd
import numpy as np
import seaborn as sns

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

```
Out[41]:
```

	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

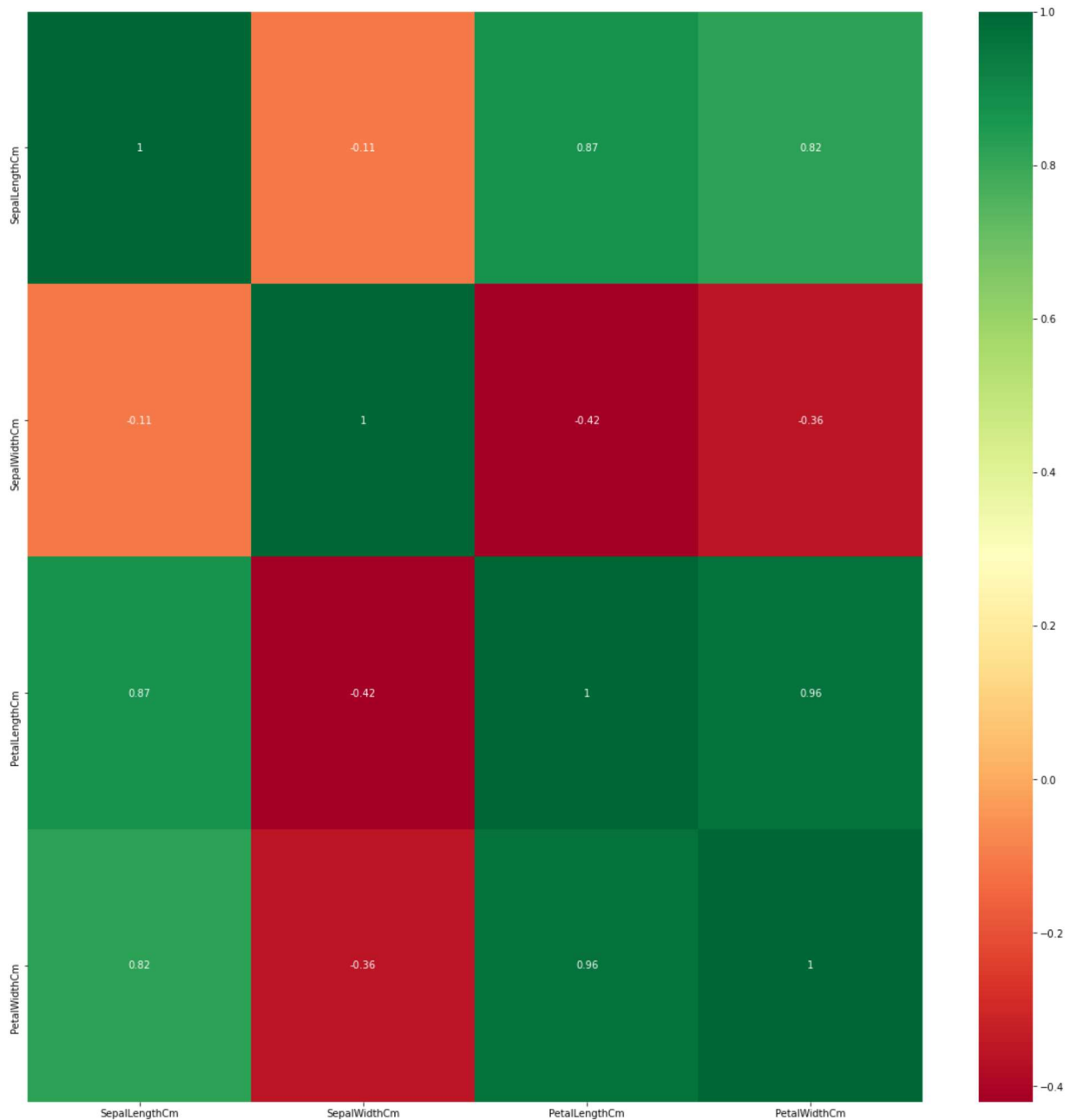
Latihan(10)

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


```
In [43]: #independent columns --> SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm
X = data.iloc[:,0:20]
# target columns --> species
y = data.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

```
In [ ]: ### Kesimpulan  
* lihat pada baris terakhir yaitu species, korelasi antara species dengan fitur  
* lain dimana ada relasi kuat dengan variabel PetalLengthCm dan diikuti oleh  
* SepalLengthCm and PetalWidthCm.  
* sedangkan utk var SepalWidthCm berkorelasi lemah dengan Species
```