



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]: 1 # import pandas
2 import pandas as pd
3 # import numpy
4 import numpy as np
5 # import Library SelectKBest
6 from sklearn.feature_selection import SelectKBest
7 # import Library chi_kuadrat/squared
8 from sklearn.feature_selection import chi2
```

```
In [2]: 1 # Load dataset
2 data = pd.read_csv('Iris.csv')
3 data
```

```
Out[2]:
```

	Id	SepalLengthCm	SepalWidthCm	Petal.LengthCm	Petal.WidthCm	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 x 6 columns

Latihan(2)

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

```
In [3]: 1 # Menampilkan kolom Id
```

```
2 data = data.drop(columns=['Id'])
3 # lalu tampilkan
4 data.head(1)
```

```
Out[3]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa

Latihan(3)

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

```
In [4]: 1 #independent columns --> SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm
2 X = data[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
3 # target columns --> species
4 y = data['Species']
```

Latihan(4)

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

```
In [5]: 1 #Apply SelectKBest class to extract
2
3 bestfeature = SelectKBest(score_func=chi2, k=4)
4 fit = bestfeature.fit(X,y)
5 dfscores = pd.DataFrame(fit.scores_)
6 dfcolumns = pd.DataFrame(X.columns)
```

Latihan(5)

- lihat hasil score seleksi feature

```
In [6]: 1 #gabungkan 2 dataframe tersebut untuk visualisasi yang lebih bagus
2
3 featureScores = pd.concat([dfcolumns, dfscores],axis=1)
4 featureScores.columns = ['Field', 'Score']
5 print(featureScores.nlargest(10,'Score'))
```

	Field	Score
2	PetalLengthCm	116.169847
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 [7]: 1 data = pd.read_csv('Iris.csv')
2
3 # Menghilangkan kolom Id
4 df2 = data.drop(columns=['Id'])
5 # lalu tampilkan
6 df2.head(1)
```

```
Out[7]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa

Latihan(7)

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

```
In [8]: 1 #independent columns --> SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm
2 A = data[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
3 # target columns --> species
4 b = data['Species']
```

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]: 1 # Import Library ExtraTreesClassifier
2 from sklearn.ensemble import ExtraTreesClassifier
3 # Import library matplotlib
4 import matplotlib.pyplot as plt
5
6 # fit model ExtraTreesClassifier
7 model = ExtraTreesClassifier(n_estimators=100, random_state=0)
8 model.fit(A,b)
```

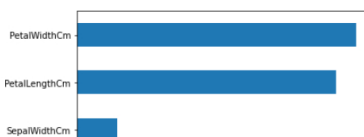
```
Out[9]: ExtraTreesClassifier(random_state=0)
```

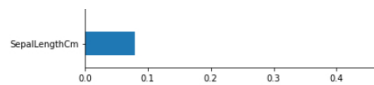
Latihan(9)

- visualisasikan hasil dari model ExtraTreesClassifier

```
In [10]: 1 print(model.feature_importances_)
2 feat_importance = pd.Series(model.feature_importances_, index=A.columns)
3 feat_importance.head(10).plot(kind='barh')
4 plt.show()
```

```
[0.08019955 0.06477472 0.41158323 0.4434425 ]
```





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 [11]: 1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4
5 data = pd.read_csv('Iris.csv')
6 df3 = data.iloc[:,1:]
7 df3
```

```
Out[11]:
```

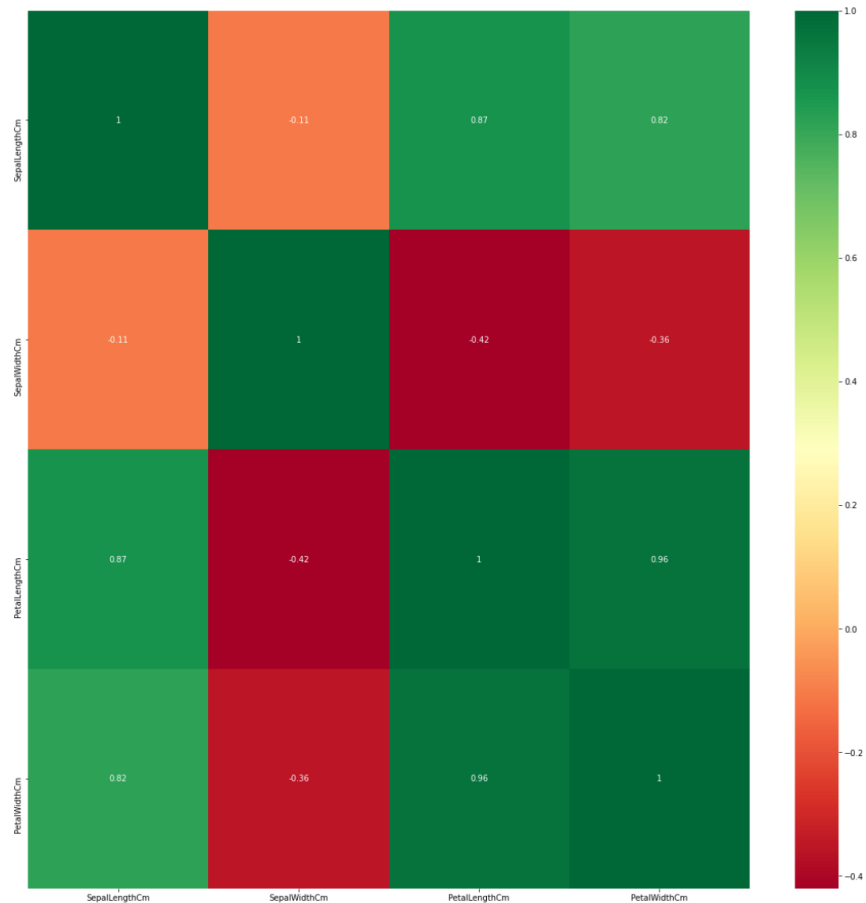
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	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 x 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]: 1 #independent columns --> SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm
2 K = df2.iloc[:, :-1]
3 # target columns --> species
4 j = df2.iloc[:, -1]
5
6 # mendapatkan korelasi di setiap fitur dalam dataset
7 corrmat = df3.corr()
8 top_corr_features = corrmat.index
9 plt.figure(figsize=(20,20))
10
11 # plot heatmap
12 h = sns.heatmap(df3[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```



```
In [ ]: 1
```

Jelaskan apa yg dapat disimpulkan dari hasil visualisasi heatmap diatas

Berdasarkan heatmap diatas korelasi tertinggi yang diperoleh adalah Petal.LengthCm dan Petal.WidthCm (0.96), yang berarti kedua fitur ini saling berkaitan satu sama lainnya. Kemudian terdapat dua fitur yang saling tidak berkorelasi yaitu 'Sepal.WidthCm' dan 'Sepal.LengthCm' (-0.11), yang berarti lebar dari sepal bunga tidak berpengaruh terhadap panjang sepal bunga.