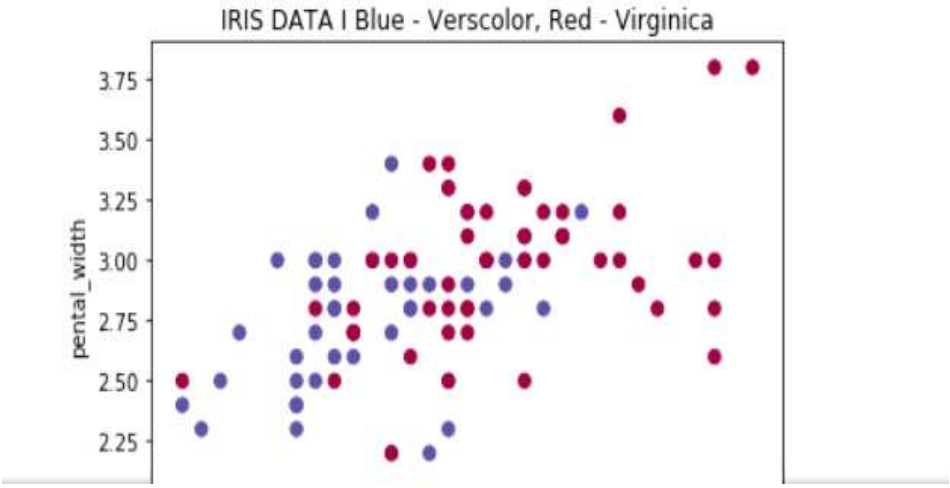
인공지능 11주차 과제

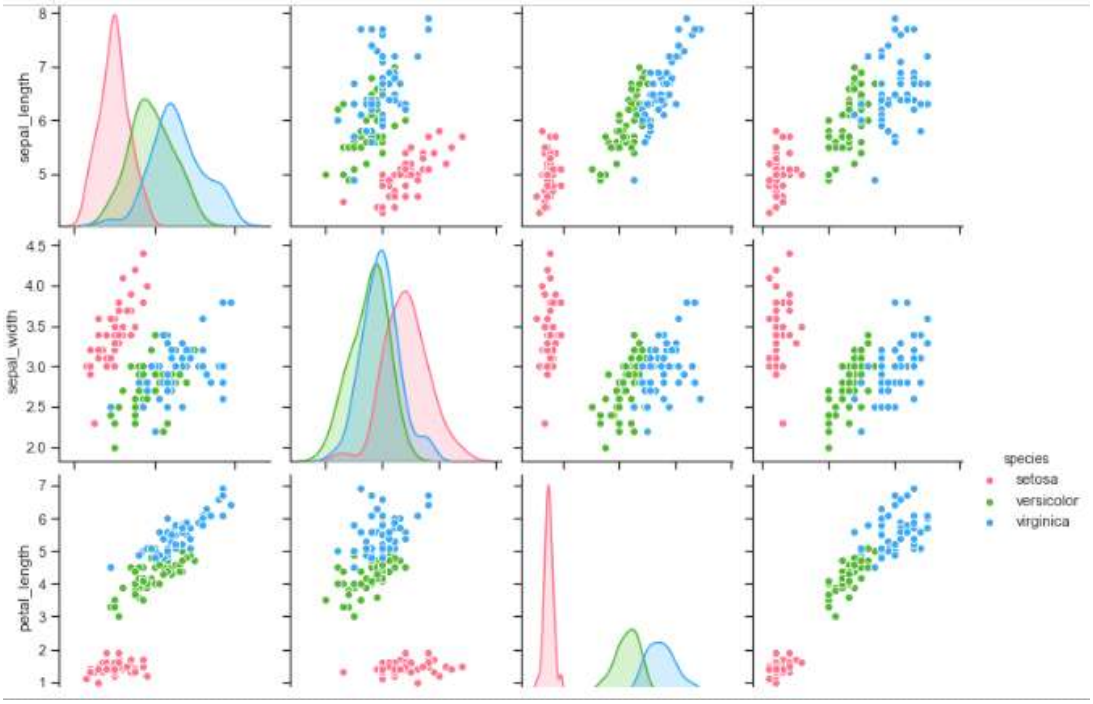
심리학과 2015011022
이명섭

```
import os
import pandas as pd
import matplotlib.pyplot as plt

os.chdir(r'C:₩Users₩myung₩Downloads')
iris = pd.read_csv('iris.csv')

print(iris.head())

iris.loc[iris['species']=='virginica','species']=0
iris.loc[iris['species']=='versicolor','species']=1
iris.loc[iris['species']=='setosa','species']=2
iris = iris[iris['species']!=2]

X = iris[['sepal_length', 'sepal_width']].values.T
Y = iris[['species']].values.T
Y = Y.astype('uint8')

plt.scatter(X[0, :], X[1,:], c=Y[0,:], s=40, cmap=plt.cm.Spectral);
plt.title("IRIS DATA I Blue - Verscolor, Red - Virginica")
plt.xlabel('pental_length')
plt.ylabel('pental_width')
plt.show()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa



```
import seaborn as sns
import pandas as pd
import numpy as np

sns.set(style="ticks", color_codes=True)
iris = sns.load_dataset("iris")
g = sns.pairplot(iris, hue="species", palette="husl")
```



```
from sklearn.preprocessing import LabelEncoder

X = iris.iloc[:,0:4].values
y = iris.iloc[:,4].values

encoder = LabelEncoder()
y1 = encoder.fit_transform(y)
Y = pd.get_dummies(y1).values
print(Y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,random_state=1)
X_train.shape, X_test.shape, y_train.shape, y_test.shape

from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam

model = Sequential()

model.add(Dense(64,input_shape=(4,),activation = 'relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(3,activation='softmax'))

model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])

model.summary()
```

```
Using TensorFlow backend.
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 64)                320
_________________________________________________________________
dense_2 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_3 (Dense)              (None, 3)                 195
=================================================================
Total params: 4,675
Trainable params: 4,675
Non-trainable params: 0
```