

```
In [17]: import keras
from sklearn.neural_network import MLPClassifier
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
import pydot
import graphviz
from IPython.display import SVG
from keras.utils.vis_utils import model_to_dot
from keras.utils import plot_model
import seaborn as sns
import pandas as pd
import numpy as np
```

```
In [18]: df = sns.load_dataset("iris")
df
```

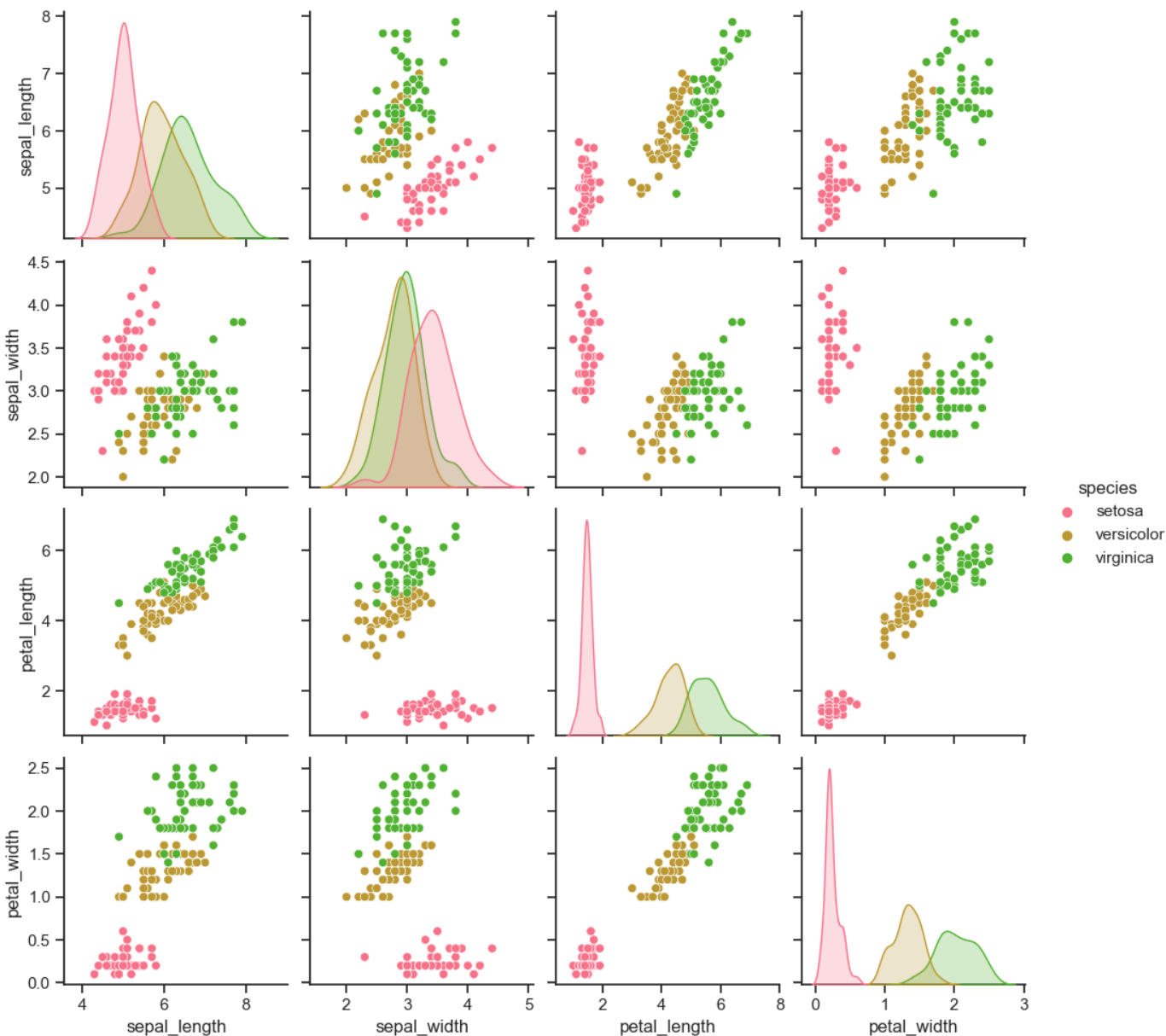
```
Out[18]:
```

	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
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
In [19]: sns.set(style="ticks")
sns.set_palette("husl")
sns.pairplot(df.iloc[:,0:6], hue="species")
```

```
Out[19]: <seaborn.axisgrid.PairGrid at 0x241f8ef67c0>
```



```
In [20]: #Splitting the data into training and test
X = df.iloc[:,0:4].values
y = df.iloc[:,4].values

from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
y1 = encoder.fit_transform(y)
Y = pd.get_dummies(y1).values

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=0)
```

```
In [21]: len(X_test)
```

```
Out[21]: 30
```

```
In [22]: model = Sequential()

model.add(Dense(4,input_shape=(4,),activation='relu'))
#model.add(Dense(8,activation='tanh'))
#model.add(Dense(6,activation='tanh'))
model.add(Dense(3,activation='softmax'))

model.compile(Adam(lr=0.04),'categorical_crossentropy',metrics=['accuracy'])
```

```
model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_2 (Dense)	(None, 4)	20
dense_3 (Dense)	(None, 3)	15

=====
Total params: 35
Trainable params: 35
Non-trainable params: 0

C:\Users\HP\anaconda3\lib\site-packages\keras\optimizers\optimizer_v2\adam.py:114: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
super().__init__(name, **kwargs)

```
In [23]: model.fit(X_train,y_train,epochs=100)
```

Epoch 1/100
4/4 [=====] - 0s 2ms/step - loss: 4.6554 - accuracy: 0.3667
Epoch 2/100
4/4 [=====] - 0s 1ms/step - loss: 2.5857 - accuracy: 0.3667
Epoch 3/100
4/4 [=====] - 0s 3ms/step - loss: 1.3916 - accuracy: 0.3667
Epoch 4/100
4/4 [=====] - 0s 3ms/step - loss: 1.0950 - accuracy: 0.3750
Epoch 5/100
4/4 [=====] - 0s 3ms/step - loss: 1.2383 - accuracy: 0.3083
Epoch 6/100
4/4 [=====] - 0s 3ms/step - loss: 1.2499 - accuracy: 0.3083
Epoch 7/100
4/4 [=====] - 0s 1ms/step - loss: 1.2252 - accuracy: 0.3083
Epoch 8/100
4/4 [=====] - 0s 2ms/step - loss: 1.1797 - accuracy: 0.3083
Epoch 9/100
4/4 [=====] - 0s 4ms/step - loss: 1.1471 - accuracy: 0.3083
Epoch 10/100
4/4 [=====] - 0s 3ms/step - loss: 1.1199 - accuracy: 0.3083
Epoch 11/100
4/4 [=====] - 0s 1ms/step - loss: 1.1071 - accuracy: 0.3083
Epoch 12/100
4/4 [=====] - 0s 1ms/step - loss: 1.1010 - accuracy: 0.3000
Epoch 13/100
4/4 [=====] - 0s 2ms/step - loss: 1.0983 - accuracy: 0.3667
Epoch 14/100
4/4 [=====] - 0s 2ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 15/100
4/4 [=====] - 0s 3ms/step - loss: 1.0974 - accuracy: 0.3667
Epoch 16/100
4/4 [=====] - 0s 3ms/step - loss: 1.0977 - accuracy: 0.3667
Epoch 17/100
4/4 [=====] - 0s 2ms/step - loss: 1.0977 - accuracy: 0.3667
Epoch 18/100
4/4 [=====] - 0s 1ms/step - loss: 1.0974 - accuracy: 0.3667
Epoch 19/100
4/4 [=====] - 0s 1ms/step - loss: 1.0973 - accuracy: 0.3667
Epoch 20/100
4/4 [=====] - 0s 3ms/step - loss: 1.0968 - accuracy: 0.3667
Epoch 21/100
4/4 [=====] - 0s 1ms/step - loss: 1.0967 - accuracy: 0.3667
Epoch 22/100
4/4 [=====] - 0s 1ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 23/100
4/4 [=====] - 0s 3ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 24/100
4/4 [=====] - 0s 1ms/step - loss: 1.0965 - accuracy: 0.3667
Epoch 25/100
4/4 [=====] - 0s 2ms/step - loss: 1.0965 - accuracy: 0.3667
Epoch 26/100
4/4 [=====] - 0s 2ms/step - loss: 1.0968 - accuracy: 0.3667
Epoch 27/100
4/4 [=====] - 0s 3ms/step - loss: 1.0961 - accuracy: 0.3667
Epoch 28/100
4/4 [=====] - 0s 2ms/step - loss: 1.0962 - accuracy: 0.3667
Epoch 29/100
4/4 [=====] - 0s 4ms/step - loss: 1.0963 - accuracy: 0.3667
Epoch 30/100
4/4 [=====] - 0s 2ms/step - loss: 1.0963 - accuracy: 0.3667
Epoch 31/100
4/4 [=====] - 0s 1ms/step - loss: 1.0962 - accuracy: 0.3667
Epoch 32/100
4/4 [=====] - 0s 2ms/step - loss: 1.0962 - accuracy: 0.3667

Epoch 33/100
4/4 [=====] - 0s 3ms/step - loss: 1.0962 - accuracy: 0.3667
Epoch 34/100
4/4 [=====] - 0s 2ms/step - loss: 1.0965 - accuracy: 0.3667
Epoch 35/100
4/4 [=====] - 0s 3ms/step - loss: 1.0962 - accuracy: 0.3667
Epoch 36/100
4/4 [=====] - 0s 1ms/step - loss: 1.0962 - accuracy: 0.3667
Epoch 37/100
4/4 [=====] - 0s 3ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 38/100
4/4 [=====] - 0s 3ms/step - loss: 1.0969 - accuracy: 0.3667
Epoch 39/100
4/4 [=====] - 0s 1ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 40/100
4/4 [=====] - 0s 2ms/step - loss: 1.0962 - accuracy: 0.3667
Epoch 41/100
4/4 [=====] - 0s 3ms/step - loss: 1.0961 - accuracy: 0.3667
Epoch 42/100
4/4 [=====] - 0s 1ms/step - loss: 1.0965 - accuracy: 0.3667
Epoch 43/100
4/4 [=====] - 0s 3ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 44/100
4/4 [=====] - 0s 1ms/step - loss: 1.0966 - accuracy: 0.3667
Epoch 45/100
4/4 [=====] - 0s 1ms/step - loss: 1.0962 - accuracy: 0.3667
Epoch 46/100
4/4 [=====] - 0s 3ms/step - loss: 1.0963 - accuracy: 0.3667
Epoch 47/100
4/4 [=====] - 0s 4ms/step - loss: 1.0962 - accuracy: 0.3667
Epoch 48/100
4/4 [=====] - 0s 3ms/step - loss: 1.0966 - accuracy: 0.3667
Epoch 49/100
4/4 [=====] - 0s 2ms/step - loss: 1.0961 - accuracy: 0.3667
Epoch 50/100
4/4 [=====] - 0s 2ms/step - loss: 1.0968 - accuracy: 0.3667
Epoch 51/100
4/4 [=====] - 0s 1ms/step - loss: 1.0961 - accuracy: 0.3667
Epoch 52/100
4/4 [=====] - 0s 3ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 53/100
4/4 [=====] - 0s 2ms/step - loss: 1.0970 - accuracy: 0.3667
Epoch 54/100
4/4 [=====] - 0s 2ms/step - loss: 1.0963 - accuracy: 0.3667
Epoch 55/100
4/4 [=====] - 0s 1ms/step - loss: 1.0962 - accuracy: 0.3667
Epoch 56/100
4/4 [=====] - 0s 3ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 57/100
4/4 [=====] - 0s 1ms/step - loss: 1.0963 - accuracy: 0.3667
Epoch 58/100
4/4 [=====] - 0s 3ms/step - loss: 1.0968 - accuracy: 0.3667
Epoch 59/100
4/4 [=====] - 0s 1ms/step - loss: 1.0969 - accuracy: 0.3667
Epoch 60/100
4/4 [=====] - 0s 1ms/step - loss: 1.0967 - accuracy: 0.3667
Epoch 61/100
4/4 [=====] - 0s 3ms/step - loss: 1.0967 - accuracy: 0.3667
Epoch 62/100
4/4 [=====] - 0s 2ms/step - loss: 1.0968 - accuracy: 0.3667
Epoch 63/100
4/4 [=====] - 0s 2ms/step - loss: 1.0970 - accuracy: 0.3667
Epoch 64/100
4/4 [=====] - 0s 1ms/step - loss: 1.0964 - accuracy: 0.3667

Epoch 65/100
4/4 [=====] - 0s 1ms/step - loss: 1.0965 - accuracy: 0.3667
Epoch 66/100
4/4 [=====] - 0s 2ms/step - loss: 1.0969 - accuracy: 0.3667
Epoch 67/100
4/4 [=====] - 0s 1ms/step - loss: 1.0972 - accuracy: 0.3667
Epoch 68/100
4/4 [=====] - 0s 3ms/step - loss: 1.0969 - accuracy: 0.3667
Epoch 69/100
4/4 [=====] - 0s 2ms/step - loss: 1.0970 - accuracy: 0.3667
Epoch 70/100
4/4 [=====] - 0s 2ms/step - loss: 1.0975 - accuracy: 0.3667
Epoch 71/100
4/4 [=====] - 0s 2ms/step - loss: 1.0963 - accuracy: 0.3667
Epoch 72/100
4/4 [=====] - 0s 1ms/step - loss: 1.0960 - accuracy: 0.3667
Epoch 73/100
4/4 [=====] - 0s 3ms/step - loss: 1.0970 - accuracy: 0.3667
Epoch 74/100
4/4 [=====] - 0s 1ms/step - loss: 1.0960 - accuracy: 0.3667
Epoch 75/100
4/4 [=====] - 0s 2ms/step - loss: 1.0961 - accuracy: 0.3667
Epoch 76/100
4/4 [=====] - 0s 1ms/step - loss: 1.0960 - accuracy: 0.3667
Epoch 77/100
4/4 [=====] - 0s 1ms/step - loss: 1.0965 - accuracy: 0.3667
Epoch 78/100
4/4 [=====] - 0s 2ms/step - loss: 1.0965 - accuracy: 0.3667
Epoch 79/100
4/4 [=====] - 0s 2ms/step - loss: 1.0966 - accuracy: 0.3667
Epoch 80/100
4/4 [=====] - 0s 1ms/step - loss: 1.0966 - accuracy: 0.3667
Epoch 81/100
4/4 [=====] - 0s 1ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 82/100
4/4 [=====] - 0s 1ms/step - loss: 1.0961 - accuracy: 0.3667
Epoch 83/100
4/4 [=====] - 0s 1ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 84/100
4/4 [=====] - 0s 2ms/step - loss: 1.0971 - accuracy: 0.3667
Epoch 85/100
4/4 [=====] - 0s 1ms/step - loss: 1.0966 - accuracy: 0.3667
Epoch 86/100
4/4 [=====] - 0s 1ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 87/100
4/4 [=====] - 0s 2ms/step - loss: 1.0967 - accuracy: 0.3667
Epoch 88/100
4/4 [=====] - 0s 1ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 89/100
4/4 [=====] - 0s 2ms/step - loss: 1.0969 - accuracy: 0.3667
Epoch 90/100
4/4 [=====] - 0s 1ms/step - loss: 1.0961 - accuracy: 0.3667
Epoch 91/100
4/4 [=====] - 0s 1ms/step - loss: 1.0963 - accuracy: 0.3667
Epoch 92/100
4/4 [=====] - 0s 1ms/step - loss: 1.0963 - accuracy: 0.3667
Epoch 93/100
4/4 [=====] - 0s 2ms/step - loss: 1.0964 - accuracy: 0.3667
Epoch 94/100
4/4 [=====] - 0s 2ms/step - loss: 1.0961 - accuracy: 0.3667
Epoch 95/100
4/4 [=====] - 0s 2ms/step - loss: 1.0966 - accuracy: 0.3667
Epoch 96/100
4/4 [=====] - 0s 2ms/step - loss: 1.0966 - accuracy: 0.3667

```
Epoch 97/100
4/4 [=====] - 0s 1ms/step - loss: 1.0962 - accuracy: 0.3667
Epoch 98/100
4/4 [=====] - 0s 2ms/step - loss: 1.0976 - accuracy: 0.3667
Epoch 99/100
4/4 [=====] - 0s 1ms/step - loss: 1.0963 - accuracy: 0.3667
Epoch 100/100
4/4 [=====] - 0s 1ms/step - loss: 1.0966 - accuracy: 0.3667
<keras.callbacks.History at 0x241f993dd30>
```

```
In [24]: y_pred = model.predict(X_test)
y_test_class = np.argmax(y_test,axis=1)
y_pred_class = np.argmax(y_pred,axis=1)

1/1 [=====] - 0s 42ms/step
```

[illegible]

```
In [26]: #Accuracy of the predicted values
from sklearn.metrics import classification_report, confusion_matrix
print(classification_report(y_test_class, y_pred_class))
print(confusion_matrix(y_test_class, y_pred_class))
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	11
1	0.00	0.00	0.00	13
2	0.20	1.00	0.33	6
accuracy			0.20	30
macro avg	0.07	0.33	0.11	30
weighted avg	0.04	0.20	0.07	30

```
[[ 0  0 11]
 [ 0  0 13]
 [ 0  0  6]]
```

```
C:\Users\HP\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1334: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\HP\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1334: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\HP\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1334: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
In [27]: X_test2 = np.array([[5, 3.9, 2, 0.5],[5,2.5,3,1],[8,3.5,6,2]])#setosa, versicolor, virgi
y_pred2 = model.predict(X_test2)
y_pred2
```

```
1/1 [=====] - 0s 25ms/step
Out[27]: array([[0.3232967 , 0.31389007, 0.3628132 ],
 [0.3232967 , 0.31389007, 0.3628132 ],
 [0.3232967 , 0.31389007, 0.3628132 ]], dtype=float32)
```

```
In [28]: print(np.argmax(y_pred2,axis=1))#0 = setosa 1 = versicolor 2 = virginica
[2 2 2]
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```