

Table of Contents

- [1 sns - pairplot](#)
- [2 LinearRegression](#)
- [3 Gaussian Naive Bayes \(GaussianNB\)](#)
- [4 iris dimensionality](#)
- [5 Gaussian mixture model \(GMM\)](#)

In [1]:

```
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
```

In [2]:

```
import numpy as np
```

sns - pairplot

In [3]:

```
iris = sns.load_dataset('iris')

iris.head()
```

Out[3]:

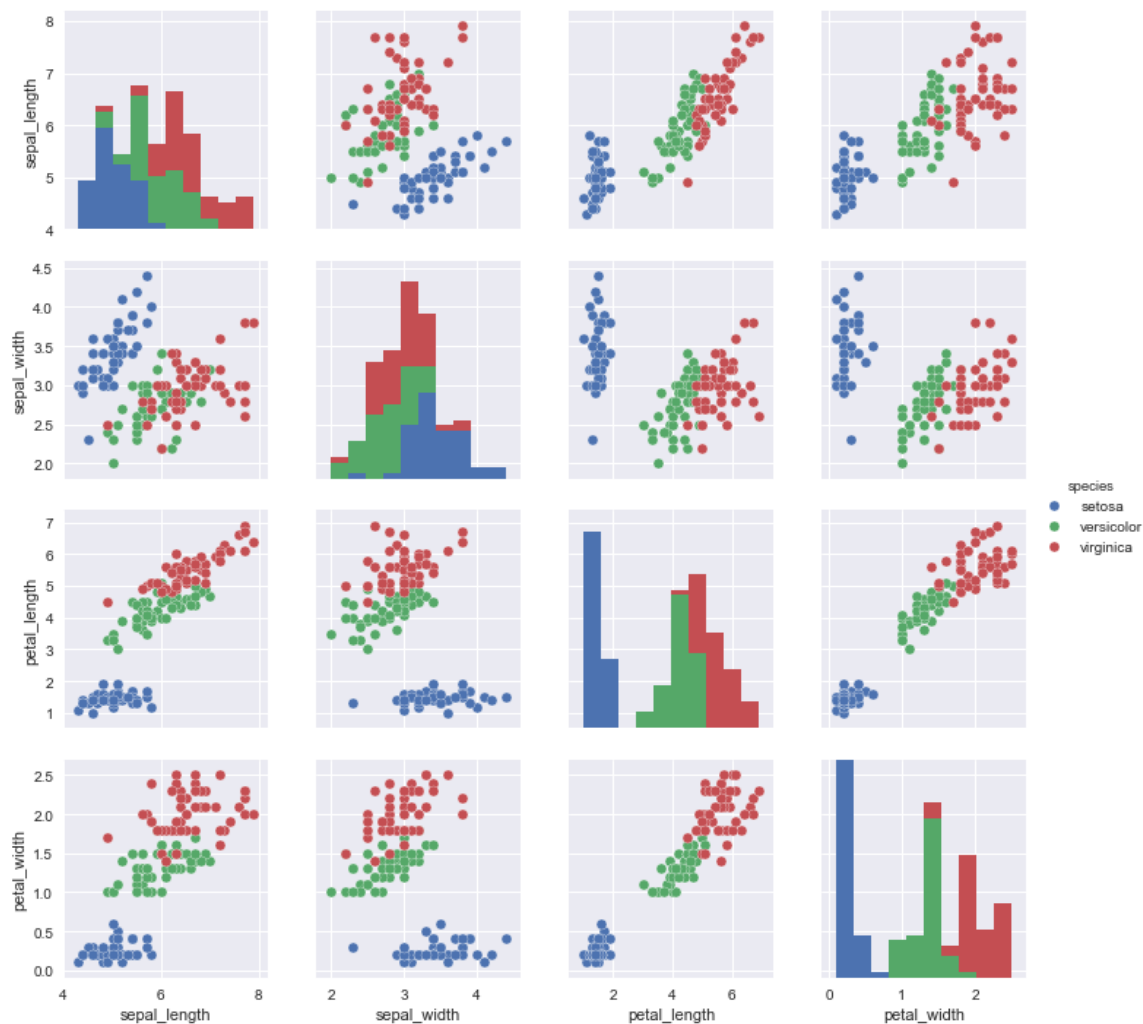
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [4]:

```
sns.set()  
  
sns.pairplot(iris, hue='species', size=2.5)
```

Out[4]:

<seaborn.axisgrid.PairGrid at 0xae1dc50>



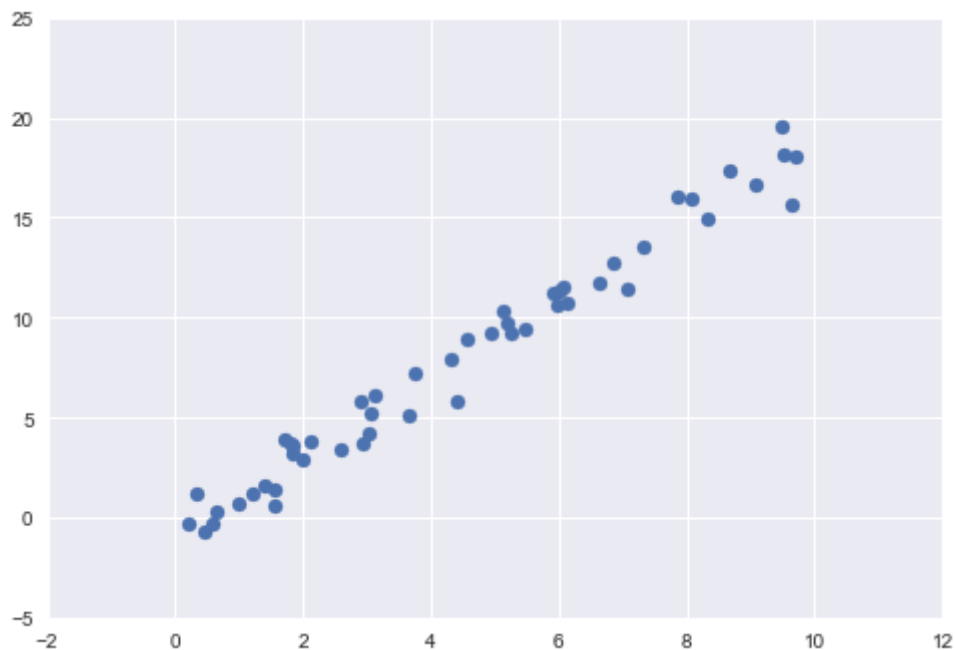
LinearRegression

In [5]:

```
rng = np.random.RandomState(42)

x = 10* rng.rand(50)
y = 2*x -1 + rng.randn(50)

plt.scatter(x, y)
plt.xlim(-2, 12)
plt.ylim(-5, 25)
# plt.xlim(xmax=12)
# plt.ylim(ymax=25)
plt.show()
```



In [6]:

```
from sklearn.linear_model import LinearRegression
```

In [7]:

```
model = LinearRegression(fit_intercept=True)
model
```

Out[7]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [8]:

```
X = x[:, np.newaxis]
print(X.shape)
# X
```

```
(50, 1)
```

In [9]:

```
model.fit(X, y)
```

Out[9]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [10]:

```
model.coef_
```

Out[10]:

```
array([ 1.9776566])
```

In [11]:

```
model.intercept_
```

Out[11]:

```
-0.90331072553111635
```

In [12]:

```
xfit = np.linspace(-1, 11)  
Xfit = xfit[:, np.newaxis]  
yfit = model.predict(Xfit)
```

In [13]:

```
xfit.shape
```

Out[13]:

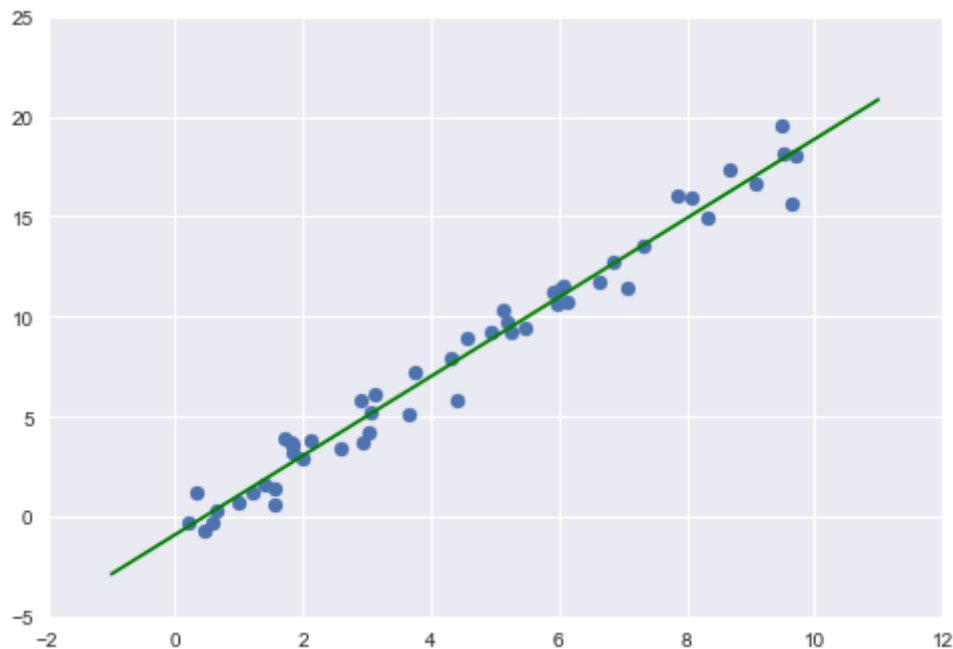
```
(50,)
```

In [14]:

```
plt.scatter(x, y)
plt.plot(xfit, yfit, color='g')

plt.xlim(-2, 12)
plt.ylim(-5, 25)

plt.show()
```



Gaussian Naive Bayes (GaussianNB)

In [15]:

```
from sklearn.cross_validation import train_test_split
```

C:\Program Files\Anaconda3\lib\site-packages\sklearn\cross_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of the model_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are different from that of this module. This module will be removed in 0.20.

"This module will be removed in 0.20.", DeprecationWarning)

In [16]:

```
from sklearn.model_selection import train_test_split
```

In [17]:

```
X_iris = iris.drop('species', axis=1)
y_iris = iris['species']

# Xtrain, Xtest, ytrain, ytest = train_test_split(X_iris, y_iris, random_state=1, test_size=0.5)
Xtrain, Xtest, ytrain, ytest = train_test_split(X_iris, y_iris, random_state=1)
```

In [18]:

```
from sklearn.naive_bayes import GaussianNB
```

In [19]:

```
model = GaussianNB()

model.fit(Xtrain, ytrain)
y_predict = model.predict(Xtest)
```

In [20]:

```
from sklearn.metrics import accuracy_score
```

In [21]:

```
accuracy_score(ytest, y_predict)
```

Out[21]:

0.97368421052631582

iris dimensionality

In [22]:

```
from sklearn.decomposition import PCA

model = PCA(n_components=2)
model.fit(X_iris)
X_2D = model.transform(X_iris)
```

In [23]:

```
iris['PCA1'] = X_2D[:, 0]
iris['PCA2'] = X_2D[:, 1]

sns.lmplot('PCA1', 'PCA2', hue='species', data=iris, fit_reg=False, size=8)
```

Out[23]:

<seaborn.axisgrid.FacetGrid at 0xdc839b0>



Gaussian mixture model (GMM)

In [24]:

```
# from sklearn.mixture import GMM
```

In [28]:

```
# model = GMM(n_components=3, covariance_type='full')
# model.fit(X_iris)

# y_gmm = model.predict(X_iris)
```