In [1]:

```
import sklearn
import numpy as np
from sklearn import datasets
from sklearn.datasets import make_classification
```

In [2]:

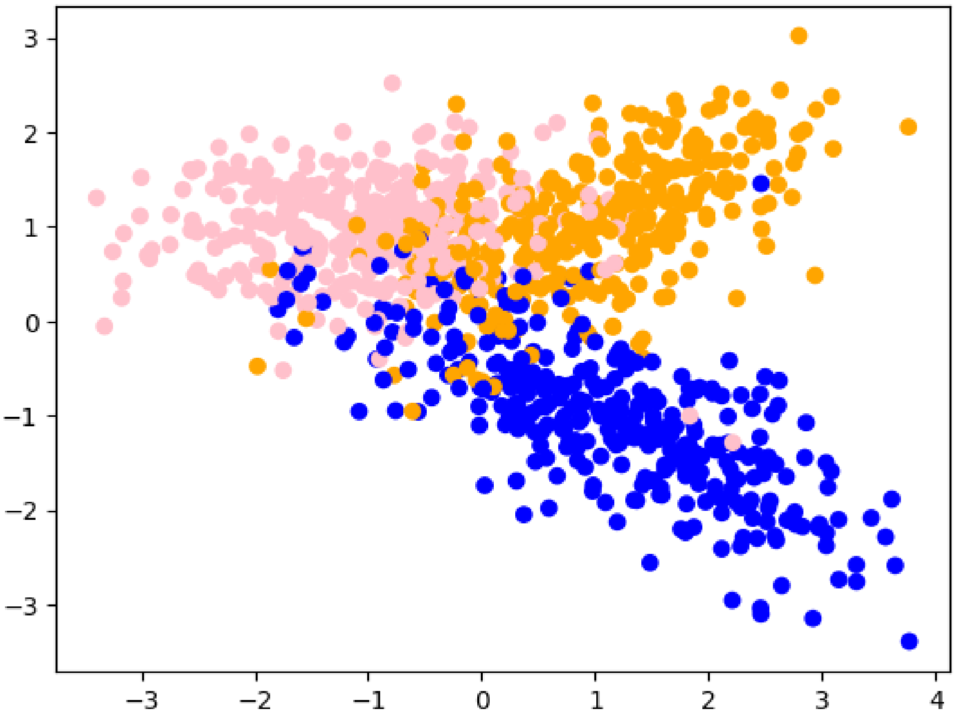
```
X, y = make_classification(n_samples=1000, n_features=2, n_classes=3,
                    n_redundant=0, n_informative=2,
                    n_clusters_per_class=1, random_state=0)
```

In [3]:

```
import matplotlib.pyplot as plt
colors = np.array(["blue", "orange", "pink"])
plt.scatter(X[:,0], X[:, 1], color=colors[y])
```

Out[3]:

```
<matplotlib.collections.PathCollection at 0x7f34566a0610>
```



In [4]:

```
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 [5]:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
x_train = scaler.transform(X_train)
x_test = scaler.transform(X_test)
```

In [7]:

```
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(8),
                    batch_size=32,
                    #learning_rate= 'adaptive', # for 'sgd'
                    learning_rate_init=0.001,
                    max_iter=1000, # no of epochs
                    early_stopping=True)
```

In [8]:

```
mlp.fit(x_train, y_train)
```

Out[8]:

```
▾                         MLPClassifier
MLPClassifier(batch_size=32, early_stopping=True, hidden_layer_sizes=8,
              max_iter=1000)
```

In [9]:

```
mlp.score(x_test, y_test)
```

Out[9]:

```
0.875
```

In [ ]:

```
parm_grid = [
    {'hidden_layer_sizes': [(8), (16), (8,16,8)]}
]
```