# Irisデータセットを使ってkNNを体験する

## ライブラリのインポート

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from mlxtend.plotting import plot_decision_regions
```

## ハイパーパラメータとデータセットの準備

```
k = 5
seed = 1
d1 = 0
d2 = 1

iris = datasets.load_iris()
x = iris.data[:, [d1, d2]]
y = iris.target

x_train, x_test, y_train, y_test =\
    train_test_split(x, y, test_size=0.3, random_state=seed)

sc = StandardScaler()
sc.fit(x_train)
x_train_std = sc.transform(x_train)
x_test_std = sc.transform(x_test)
```

## kNNの学習

```
knn = KNeighborsClassifier(n_neighbors=k)
knn.fit(x_train_std, y_train)
```

## 分類制度の算出

```
train_acc = accuracy_score(y_train, knn.predict(x_train_std))
test_acc = accuracy_score(y_test, knn.predict(x_test_std))

print("k={}, features=({}, {})".format(k, d1, d2))
print("Training accuracy: {:.4%}".format(train_acc))
print("Test_accuracy:     {:.4%}".format(test_acc))
```

# 識別境界面をプロット

## Trainデータセットを使った場合

```
plt.title("Using train_dataset")
plt.xlabel("sepal length [standardized]")
plt.ylabel("sepal width [standardized]")
plot_decision_regions(x_train_std, y_train, clf=knn)
```

## Testデータセットを使った場合

```
plt.title("Using test_dataset")
plt.xlabel("sepal length [standardized]")
plt.ylabel("sepal width [standardized]")
plot_decision_regions(x_test_std, y_test, clf=knn)
```