In [1]:
```
# This is an excerpt and translated version of my course assignment. The origina
# https://github.com/HhQqYy0926/Big-Data-Analysis-in-Business-Project
```

Penguin Dataset Analysis

In [2]:
```
import pandas as pd
import numpy as np
data=pd.read_csv("./penguins_size.csv")
```

This dataset contains three different species of penguins from three islands in the Palmer Archipelago, namely Adelie, Chinstrap, and Gentoo. Among them, attributes include the length and depth of the upper ridge of the bird's beak, the length of the fins, weight, and gender.

In [3]:
```
data = data.dropna()
mapping = {'Adelie': 0, 'Chinstrap': 1, 'Gentoo': 2}
data['species'] = data['species'].map(mapping) #Convert species category charact
data = data.drop(['sex','island'], axis=1)
X = data[['flipper_length_mm','culmen_depth_mm']]
y = data['species']
```

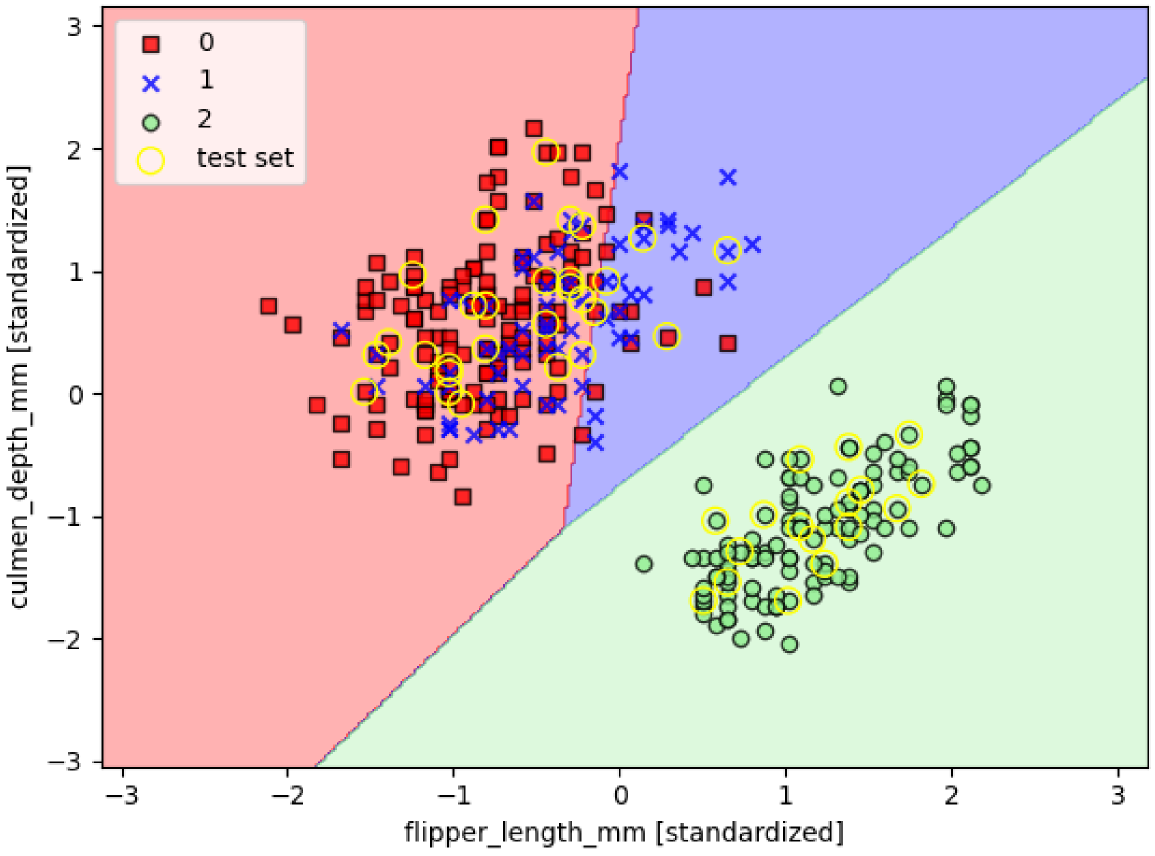
In [4]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=1, stratify=y)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
X_combined_std = np.vstack((X_train_std, X_test_std))
y_combined = np.hstack((y_train, y_test))
```

In [5]:
```
# Visualization
from matplotlib.colors import ListedColormap
import matplotlib.pyplot as plt
def plot_decision_regions(X, y, classifier, test_idx=None, resolution=0.02):
    markers = ('s', 'x', 'o', '^', 'v')
    colors = ('red', 'blue', 'lightgreen', 'gray', 'cyan')
    cmap = ListedColormap(colors[:len(np.unique(y))])
    x1_min, x1_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    x2_min, x2_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx1, xx2 = np.meshgrid(np.arange(x1_min, x1_max, resolution),
                           np.arange(x2_min, x2_max, resolution))
    Z = classifier.predict(np.array([xx1.ravel(), xx2.ravel()]).T)
    Z = Z.reshape(xx1.shape)
    plt.contourf(xx1, xx2, Z, alpha=0.3, cmap=cmap)
    plt.xlim(xx1.min(), xx1.max())
    plt.ylim(xx2.min(), xx2.max())
    for idx, cl in enumerate(np.unique(y)):
        plt.scatter(x=X[y == cl, 0],
                    y=X[y == cl, 1],
                    alpha=0.8,
                    c=colors[idx],
                    marker=markers[idx],
                          label=cl,
                          edgecolor='black')
    if test_idx:
        X_test, y_test = X[test_idx, :], y[test_idx]
        plt.scatter(X_test[:, 0],
                          X_test[:, 1],
                          c='none',
                          edgecolor='yellow',
                          alpha=1.0,
                          linewidth=1,
                          marker='o',
                          s=100,
                          label='test set')
```

In [6]:

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(C=100.0, random_state=1)
lr.fit(X_train_std, y_train)
plot_decision_regions(X_combined_std, y_combined,
                      classifier=lr, test_idx=range(105, 150))
plt.xlabel('flipper_length_mm [standardized]')
plt.ylabel('culmen_depth_mm [standardized]')
plt.legend(loc='upper left')
plt.tight_layout()
plt.show()
```

```
/tmp/ipykernel_3801/1232439212.py:18: UserWarning: You passed a edgecolor/edgecol
ors ('black') for an unfilled marker ('x').  Matplotlib is ignoring the edgecolor
in favor of the facecolor.  This behavior may change in the future.
  plt.scatter(x=X[y == cl, 0],
```



It can be seen that Gentoo's category distinction is better, but Adelie and Chinstra's category distinction is more vague.