Lab 5.2 Classification (KNN&SVM) (13/5/2022)

ລະຫັດນັກສຶກສາ: **205Q0010.19**
ຊື່ ແລະ ນາມສະກຸນ: **ທ້າວ ນູຊ໋ອ ເຮີ 3CW1**

1. ຈາກຄຳສັ່ງຂອງ Classification_KNN_SVM, ຈົ່ງແຍກຊຸດຂໍ້ມູນອອກເປັນສອງພາກສ່ວນຄື: ຊຸດຮຽນ 80 ແລະ ຊຸດທົດສອບ 20 ?

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

2. ຈົ່ງສ້າງໂມເດອ KNeighborsClassifier ແລະ ທຳການປະມອນຜົນ (fit) ຊຸດຂໍ້ມູນ X_train

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

3. ຈົ່ງທົດສອບໂມເດອດ້ວຍການpredict(X_test).

```
y_pred = classifier.predict(X_test)
print(y_pred)
```

```
[0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0
 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 0 0 0 1 0 0 1
 0 0 0 0 1 1]
```

4. ຈົ່ງທຳການprocessing ດ້ວຍconfusion_matrix, ກຳນົດTP, TN, FP, FN

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[64  4]
 [ 3 29]]
```

5. ຈົ່ງສະແດງຜົນດ້ວຍການສົມທຽບຄ່າຈິງ ແລະ ຄ່າຄາດເດົາຂອງ y_test ດ້ວຍນຽบDataFrame

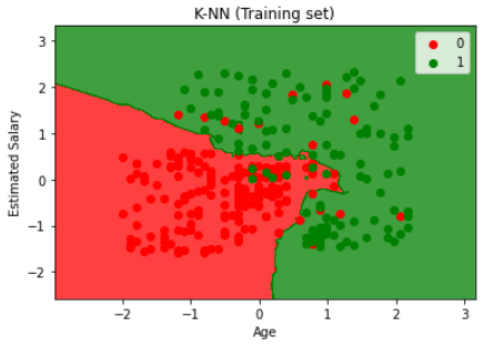
```
dx=pd.DataFrame({'y_true': y_test, 'y_pred': predsvc})
dx[dx.y_true != dx.y_pred]
```

	y_true	y_pred
9	0	1
15	0	1
31	1	0
53	0	1
81	0	1
85	1	0
95	1	0

6.ຈົ່ງສະແດງຂໍ້ມູນຊຸດຮຽน (X_train, y_train) ດ້ວยGraph ขึ้นพื้นฐานชุดคำสั่ง matplotlib.

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('K-NN (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

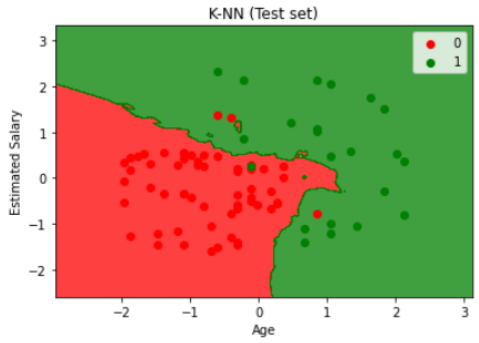
```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
```



7.ຈົ່ງສະແດງຂໍ້ມູນຊຸດຮຽນ (X_test, y_test) ດ້ວຍGraph ບົນພື້ນຖານຊຸດຄຳສັ່ງ matplotlib

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('K-NN (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
```



8. ຈົ່ງສ້າງໂມເດວ SVC ແລະ ທຳການປະມວນຜົນ (fit) ຊຸດຂໍ້ມູນ X_train

```
from sklearn.svm import SVC
```

```
svm_classifier=SVC()
```

```
svm_classifier.fit(X_train,y_train)
```

```
SVC()
```

9. ຈົ່ງທົດສອບໂມເດວດ້ວຍການpredict(X_test).

```
predsvc = svm_classifier.predict(X_test)
print(predsvc)
```

```
[0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0
 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 1 0 0 1
 0 0 0 0 1 1]
```

10. ຈົ່ງທຳການprocessing ດ້ວຍconfusion_matrix, ນຳນົດTP, TN, FP, FN

```
print(confusion_matrix(y_test, predsvc))
```

```
[[64  4]
 [ 3 29]]
```

11. ຈົ່ງສະແດງຜົນດ້ວຍການສົມທຽບຄ່າຈິງ ແລະ ຄ່າຄາດເດົາຂອງ y_test ດ້ວຍມູນDataFrame

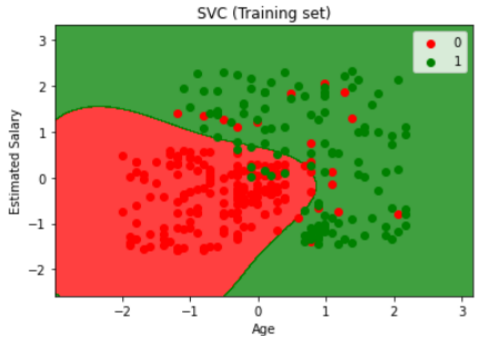
```
dx=pd.DataFrame({'y_true': y_test, 'y_pred': predsvc})
dx[dx.y_true != dx.y_pred]
```

	y_true	y_pred
9	0	1
15	0	1
31	1	0
53	0	1
81	0	1
85	1	0
95	1	0

12.ຈົ່ງສະແດງຂໍ້ມູນຊຸດຮຽນ (X_train, y_train) ດ້ວຍGraph ບົນພື້ນຖານຊຸດຄຳສັ່ງ matplotlib.

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, svm_classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('SVC (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
```



13. ຈົ່ງສະແດງຂໍ້ມູນຊຸດຮຽນ (X_test, y_tes) ດ້ວຍGraph ບົນພື້ນຖານຊຸດຄຳສັ່ງ matplotlib.

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, svm_classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('SVM (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D array with a single row if you intend to specify the same RGB or RGBA value for all points.
```