Machine Learning - Assignment 02

Part 1
Support Vector Machine

February 11, 2019

Problem

Suppose there is a social networking site that has several business clients which can put their ads on this social network.
One of their clients is a Car Company that has just launched their brand new luxury SUV for an extremely high price.
The Company wants to know which of the users in the network are potential candidates for buying this brand new SUV.
They are provided with the information of 400 users which includes the user id, gender, age, and estimated salary. The last column tells if the user bought this SUV or not.
Model a SVM classifier with a linear kernel that predicts if a user is going to make the purchase or not based on two variables, age and estimated salary.

Use the following block of code for computing the confusion matrix:

---

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

You only need to make changes to the variables **y_pred** that denotes the predicted output of the classifier and **y_test** that denotes the actual output.

Use the following block of code to plot the graph of the Classification Model :

```
from matplotlib.colors import ListedColormap
import matplotlib.pyplot as plt
X_set, y_set = X_train, y_train
X_grid_0 = np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1,
step = 0.01)
X_grid_1 = np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1,
step = 0.01)
X1 , X2 = np.meshgrid(X_grid_0, X_grid_1)
X3 = classifier.predict(np.array([X1.ravel(),X2.ravel()]).T).reshape(X1.shape)
plt.contourf(X1, X2, X3, alpha = 0.50, cmap = ListedColormap(("yellow",
"cyan")))
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(("yellow", "cyan"))(i), label = j,edgecolors = "Black")
plt.title("Logistic Regression")
plt.xlabel("Age")
plt.ylabel("EstimatedSalary")
plt.legend()
plt.show()
```

You only need to make changes to the variables **X_train** that denotes the set of input variables, **y_train** that denotes the output variable and the **classifier.predict** which is a method that predicts the output **y** given a set of input variables **X**.