**Introduction to Machine Learning in Engineering Science**

**National Cheng Kung University** Department of Engineering Science

Instructor: Chi-Hua Yu

**HW 2**

**Programming, Due 9:00, Tuesday, October 18th, 2021**

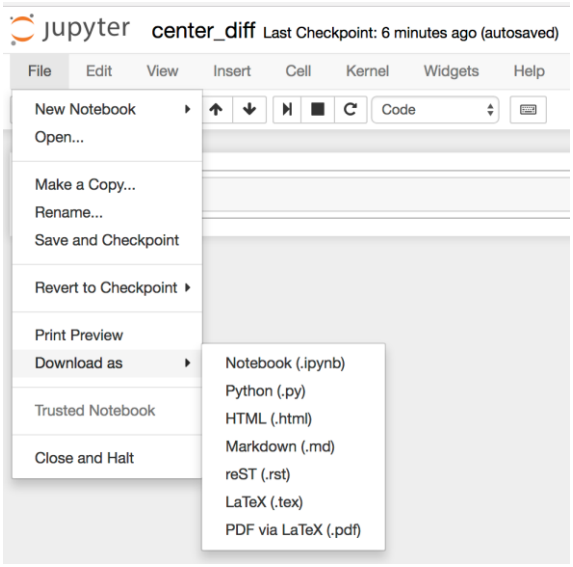
**Late submission within 24 hours: score*0.9.**

**Late submission before post of solution: score*0.8 (the solution will usually be posted within a week); no late submission after the post of solution**

---

Lab Submission Procedure (請仔細閱讀)

1. You should submit your Jupyter notebook and Python script (`*.py`, in Jupyter, click File, Download as, Python (`*.py`)).




2. Name a folder using your student id and lab number (e.g., n96081494_HW2), put all the python scripts into the folder and zip the folder (e.g., n96081494_ HW2.zip).

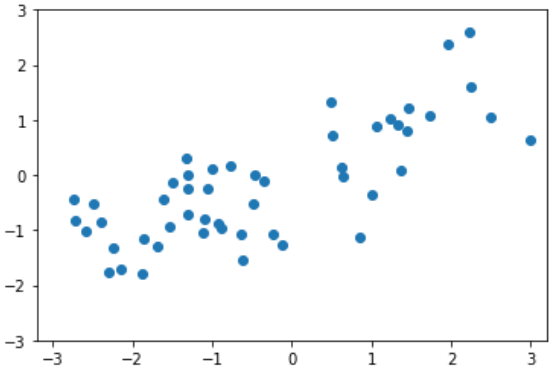
3. Submit your lab directly through the course website.

**Total 100%**

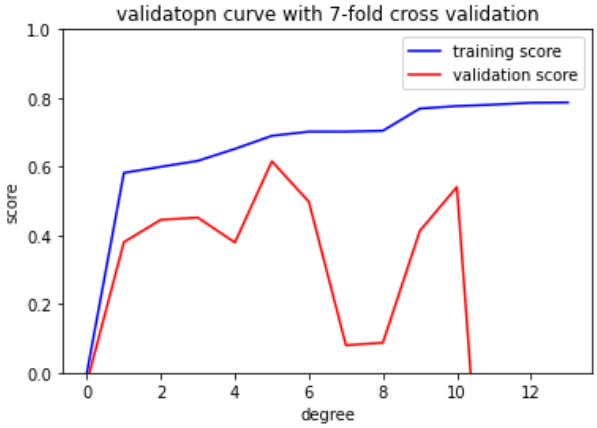
**1. (40%)** Name your Jupyter notebook `validationCurve` and Python script `validationCurve.py`. Incorporate the following `make_wave` function into your program.

```
def make_wave(n_samples=100):
    rnd = np.random.RandomState(41)
    x = rnd.uniform(-3, 3, size=n_samples)
    y_no_noise = (np.sin(4 * x) + x)
    y = (y_no_noise + rnd.normal(size=len(x))) / 2
    return x.reshape(-1, 1), y
```

(a) **(5%)** Create a wave dataset with 50 samples using `make_wave` and plot the X vs. y. Below is a sample plot:

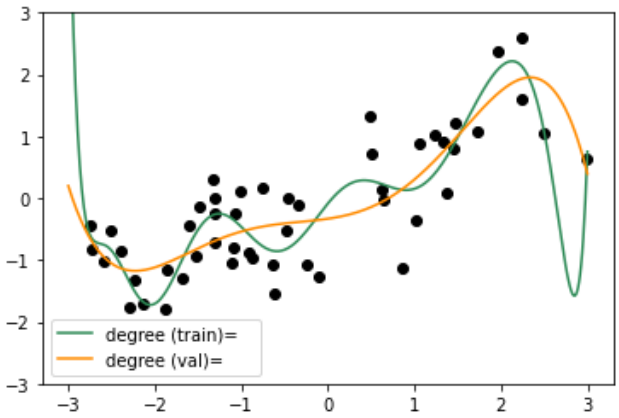


(b) **(15%)** Plot the training and validation curves using a polynomial regression model with 7-fold cross validation. Below is the sample plot:
(`degree` ranges from 0 to 13.)



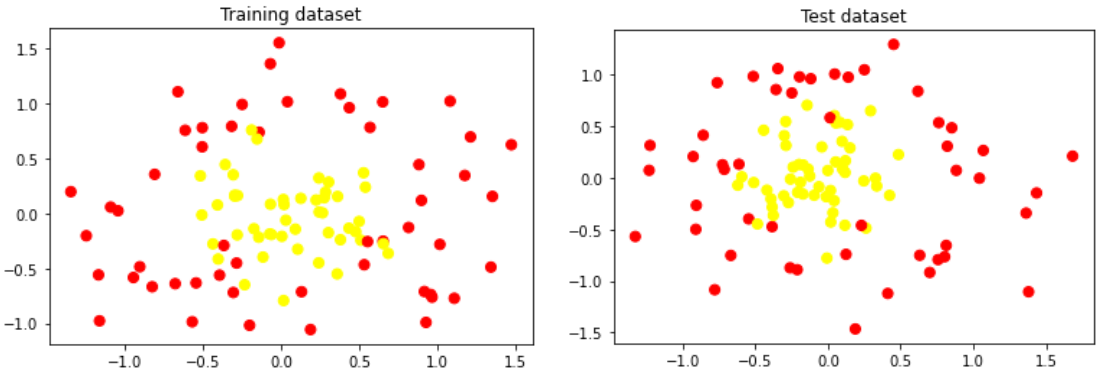

(c) **(20%)** Please find the degree with the highest score in the training and validation scores of the 7-fold cross-validation. Then plot the samples and the fitting line of the two degrees. Below is the sample plot:
(Please show the degrees in the figure.)

```
X_test = np.linspace(-3.0, 3.0, 500)[:, np.newaxis]
color_list= ['seagreen', 'darkorange']
```



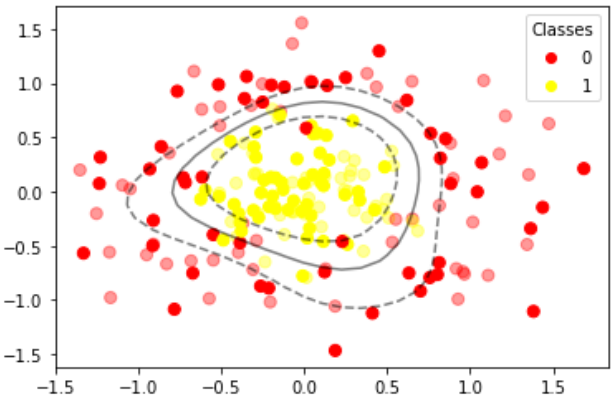
**2. (25%)** Name your Jupyter notebook `SVM` and Python script `SVM.py`. Using the following code to generate training and test dataset.

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_circles
from sklearn.model_selection import train_test_split

X, y = make_circles(200, factor=0.2, noise=0.3, random_state=1)
x_train, x_test, y_train, y_test = train_test_split(X, y,
random_state=0, test_size=0.5)

plt.title('Training dataset')
plt.scatter(x_train[:, 0], x_train[:, 1], c=y_train, s=50,
cmap='autumn')
plt.show()

plt.title('Test dataset')
plt.scatter(x_test[:, 0], x_test[:, 1], c=y_test, s=50, cmap='autumn')
plt.show()
```

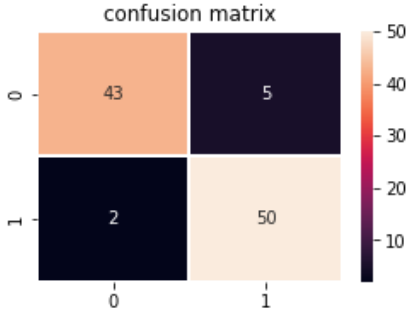



**(a) (15%)** Please find best hyperparameters of Support Vector Machine which can get the highest score in test dataset. Please plot the decision function for a 2D SVC.

```
plt.scatter(x_train[:, 0], x_train[:, 1], c=y_train,  s=50,
cmap='autumn', alpha=0.4)
scatter_test = plt.scatter(x_test[:, 0], x_test[:, 1], c=y_test,
s=50, cmap='autumn')
legend_test = plt.legend(*scatter_test.legend_elements(),
title="Classes")
plot_svc_decision_function(clf)from sklearn.model_selection import
```



**(b) (10%)** Please use hyperparameters your find to report confusion matrix and accuracy in test dataset.



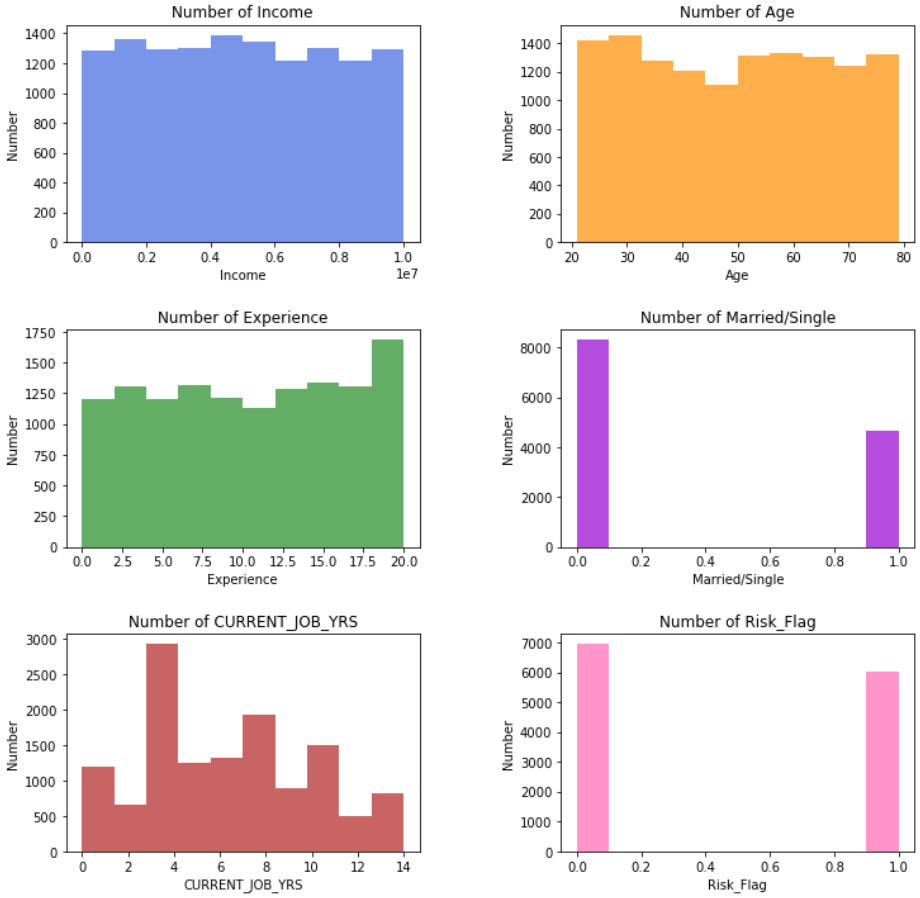

```
accuracy score: 0.93
```

**3. (35%)** Bank of LAiMM wants to predict who are possible defaulters for the consumer loans product. They have data about historic customer behavior based on what they have observed. Hence when they acquire new customers, they want to predict who is riskier and who is not. You are required to use the training dataset to identify patterns that predict "potential" defaulters. Name your Jupyter notebook `SVM_credit` and Python script `SVM_credit.py`.

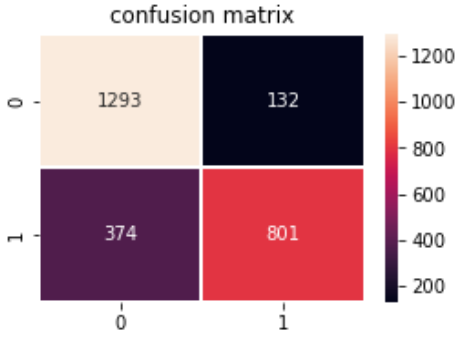
**(a) (25%)** Plot the statics of training dataset. You may need to process training data before feeding them into model.

This figure is an example, please plot the statistics of training data that are fed into the model.

```
color_list=
['royalblue','darkorange','forestgreen','darkviolet','firebrick','
hotpink']
```



**(b) (10%)** Use Support Vector Machine to identify possible defaulters, please report confusion matrix and accuracy.




```
accuracy score: 0.80538
```