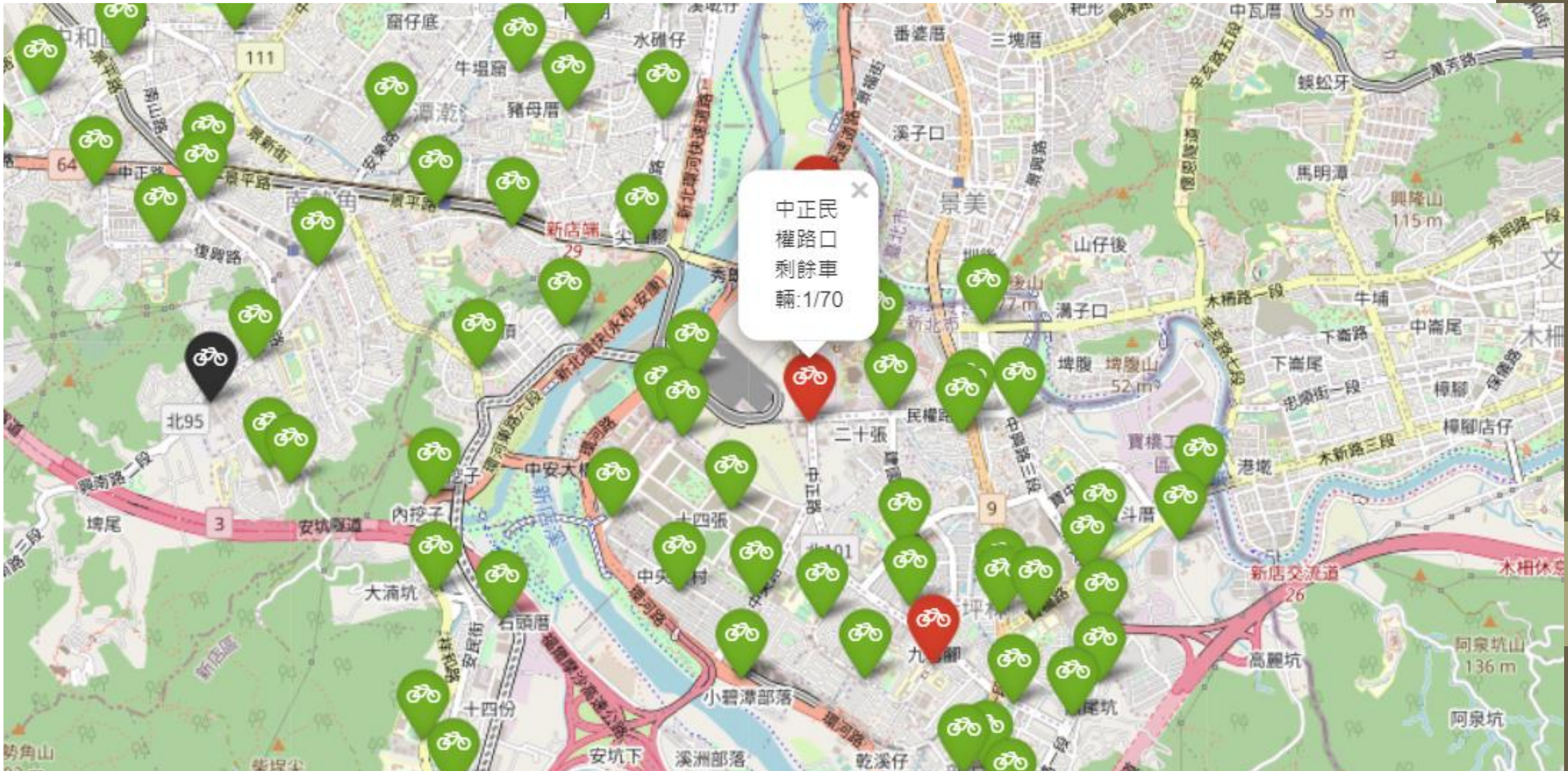
結訓評量

Python機器學習與深度學習實作

考題一

- Python基礎語法考題:
  運用地圖套件，對政府公開資料進行分析，並將分析資訊顯示在地圖上。



中正民
權路口
剩餘車
輛:1/70

考題二

- 機器學習起手式四步驟為何?
  請以SVM(支持向量機)為例，處理經典IRIS分類問題，來展示程式及成果

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

#1. 載入模型
from sklearn.svm import SVC
iris = load_iris()
X = iris.data
Y = iris.target

X = X[:,2:]
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,random_state=9487)

#2. 建立模型
clf = SVC()
#3. 訓練模型
clf.fit(x_train, y_train)

#4. 使用模型來做預測
y_predict = clf.predict(x_test)
plt.scatter(x_test[:,0], x_test[:,1], c=y_predict-y_test)
```

考題三

- 請以線性回歸的機器學習手法處理UCI(爾灣加州大學)的渦輪發電系統的電力預測問題。
  (資料來源: https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant)

```
%matplotlib inline

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import datasets, linear_model
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics

data = pd.read_excel('Folds5x2_pp.xlsx')
X = data[['AT', 'V', 'AP', 'RH']]
y = data[['PE']]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)

linreg = LinearRegression()
linreg.fit(X_train, y_train)

y_pred = linreg.predict(X_test)

# MSE
print("MSE: ", metrics.mean_squared_error(y_test, y_pred))
# RMSE
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

考題四

- 請以Keras深度學習架構處理手寫數字辨識問題。請將程式碼及完成訓練的效能列出來。

```
from tensorflow.keras import models
from tensorflow.keras import layers
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical

#- 備資料
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

#- 整理資料
train_images = train_images.reshape((60000, 28 * 28))  #reshape 是 NumPy 陣列的 method
train_images = train_images.astype('float32') / 255

test_images = test_images.reshape((10000, 28 * 28))
test_images = test_images.astype('float32') / 255


#- 建立layer, model
network = models.Sequential()
network.add(layers.Dense(512, activation='relu', input_shape=(28 * 28,)))
network.add(layers.Dense(10, activation='softmax'))

network.compile(optimizer='rmsprop',
                loss='categorical_crossentropy',
                metrics=['accuracy'])
```

```
#- 準備標籤
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)

#- training
network.fit(train_images, train_labels, epochs=5, batch_size=128)

#- testing
test_loss, test_acc = network.evaluate(test_images, test_labels)
print('test_acc:', test_acc)
```