師 大 附 中

數理資優班

專研組別:資訊

任課老師:李啟龍老師

學生班級:1612

學生座號:07

學生姓名:李宣毅

日期:2023/10/26

課程內容:Python 基本(1)

```
[ ]  num1=float(input())
     num2=float(input())
     print("%.2f+  %.2f=  %.2f"%(num1,num2,num1+num2))
     print("%.2f-  %.2f=  %.2f"%(num1,num2,num1-num2))
```

```
[ ]  F=float(input())
     C=(F-32)*5/9
     print("華氏溫度%.2f轉換攝氏溫度後是%.2f"%(F,C))
```

```
[ ]  candy=int(input())
     kid=int(input())
     print("每位可以分得%d顆，還剩下%d顆糖"%(candy//kid,candy%kid))
```

```
[ ]  list1=[1,2,3,4]
     print(sum(list1))
     list2=sum([1,2,3,4])
     print(list2)
```

```
[ ]  num=int(input())
     panduan=(num>=100  and  num<1000)
     print("%d是三位數的判斷為%s"%(num,panduan))
```

```
[ ]  num=int(input()
     if(num%2==0):
            print(f"數字{num}是偶數")
     if(num%2==1):
            print(f"數字{num}是奇數")
```

```
[ ]  purple=int(input())
     if(purple>=8):
            print("今天的紫外線指數過量，請小心防曬!")
     else:
            print("今日的紫外線指數未過量")
```

這堂課主要講解了有關 print 的幾種寫法、list 的基本程式語言還有 if/else 的用法

# 日期:2023/11/3

課程內容:Python 基本(2)

```
word="happy"
for  x  in  word:
        print(x)
```

```
for  a  in  range(5):
    print("%d.loop  is  fun!"%(a+1))
```

```
1.loop is fun!
2.loop is fun!
3.loop is fun!
4.loop is fun!
5.loop is fun!
```

```
[ ]  n=0
     for  a  in  range(10):
             n=n+a+1
             print(f"第{a+1}輪={n}")
     print(f"最後為:{n}")
```

這部分說明了迴圈的基本概念 例如 range 如果是(1,10) 迴圈就會分別跑 1~9，而 range 後面接單詞的話， 就會分別代表每一個字。

```
[ ]  i=1
     sum=0
     a=0
     while(i<=100):
             i+=a
             sum+=i
             a+=1
             print(f"i={i},+i={sum}")
```

```
[ ]  num=int(input("input?"))
     while  True:
             if  (num%7==0):
                     break
             print(num)
             num+=1
     print("剛都是非7的倍數")
```

這部份則是說明 while 在後面接的條件 如果成立 就會持續執行 若 while 裡有執行 break 則會跳出迴圈。

```
[ ] a=int(input())
    b=int(input())
    for  x  in  range(a,b+1):
            if(x%3!=0):
                    continue
            print(x,end="  ")
```

```
    for  a  in  range(1,10):
            for  b  in  range(1,10):
                    print("{0}*{1}={2:2d}".format(a,b,a*b),end="  ")
            print()
```

```
1*1= 1  1*2= 2  1*3= 3  1*4= 4  1*5= 5  1*6= 6  1*7= 7  1*8= 8  1*9= 9
2*1= 2  2*2= 4  2*3= 6  2*4= 8  2*5=10  2*6=12  2*7=14  2*8=16  2*9=18
3*1= 3  3*2= 6  3*3= 9  3*4=12  3*5=15  3*6=18  3*7=21  3*8=24  3*9=27
4*1= 4  4*2= 8  4*3=12  4*4=16  4*5=20  4*6=24  4*7=28  4*8=32  4*9=36
5*1= 5  5*2=10  5*3=15  5*4=20  5*5=25  5*6=30  5*7=35  5*8=40  5*9=45
6*1= 6  6*2=12  6*3=18  6*4=24  6*5=30  6*6=36  6*7=42  6*8=48  6*9=54
7*1= 7  7*2=14  7*3=21  7*4=28  7*5=35  7*6=42  7*7=49  7*8=56  7*9=63
8*1= 8  8*2=16  8*3=24  8*4=32  8*5=40  8*6=48  8*7=56  8*8=64  8*9=72
9*1= 9  9*2=18  9*3=27  9*4=36  9*5=45  9*6=54  9*7=63  9*8=72  9*9=81
```

```
    num=int(input())
    salary=[]
    x=0
    for  i  in  range(1,num+1):
            a=int(input(f"第{i}筆資料?"))
            x+=a
            salary.append(a)
    print(f"all={x},average={x/num:2f}")
    for  j  in  salary:
            print(f"{j}元")
```

For 迴圈的延伸

例如可以做 99 乘法表(更大的也可以) 或是驗證一串數字是否為某數的倍數等等。

```
[ ]  dict1={"apple":"蘋果","ball":"球","cat":"貓"}
     print(dict1)
     choice="0"
     while  True:
         choice=input("-1為停止，1為查詢，2為增加，3為刪減")
         if(choice=="-1"):
             break
         elif(choice=="1"):
             key=input("英文？")
             print(dict1.get(key,"查無此詞語"))
         elif(choice=="2"):
             key=input("新的英文單字？")
             key2=input("中文翻譯？")
             dict[key]=key2
         elif(choice=="3"):
             key=input("要刪的英文單字")
```

這裡是有關 dict 的程式碼 結合了 if 和 while 後 就會像上面的程式

如果我輸入了 1 2 3 就會執行上面文字敘述的程式碼

如果輸入了-1 就會跳出 while 迴圈停止運作

# 日期:2023/11/10

課程內容:機器學習和 Visual Studio Code(簡稱 VS code)的環境準備  還有 corelab 的登入準備。


#Corelab 主畫面

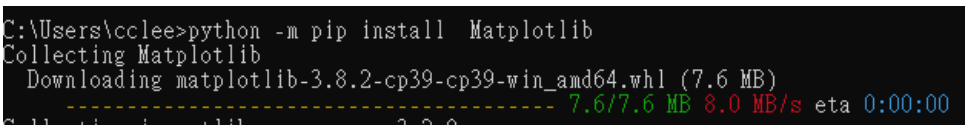
機器學習:

機器學習的介紹呢  老師為我們講解了一些有關於目前機器學習的狀況。目前機器學習分成一般類型和混合類型，底下又有分成很多種學習，主要的學習是監督式學習，用來監測像是圖片辨識，或是數值預測，還有翻譯等等，接下來講了監督式學習和其他的相異之處。之後又說了一些機器學習更深入的知識。

VS code 環境準備:

做好基礎的安裝和 Python 準備後，

老師帶我們在 VS code 裡安裝了一些以後基本上都會用到的模組 像是 pandas sk-learn 等等




(這是安裝模組的過程)

# 日期:2023/11/17and2023/11/24

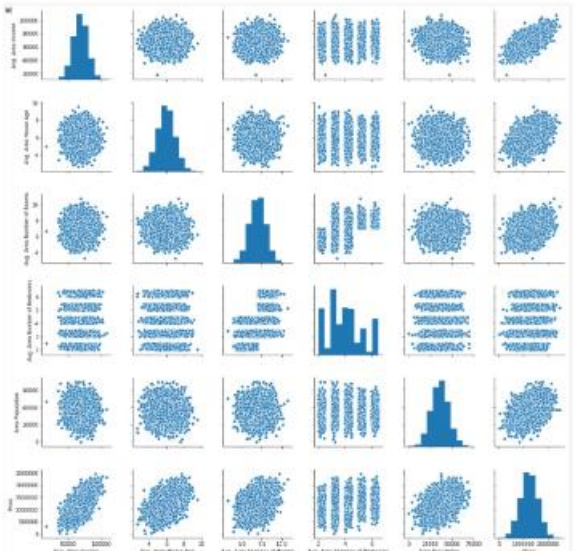
課程內容: 房價預測 & 鐵達尼號生存預測(1):

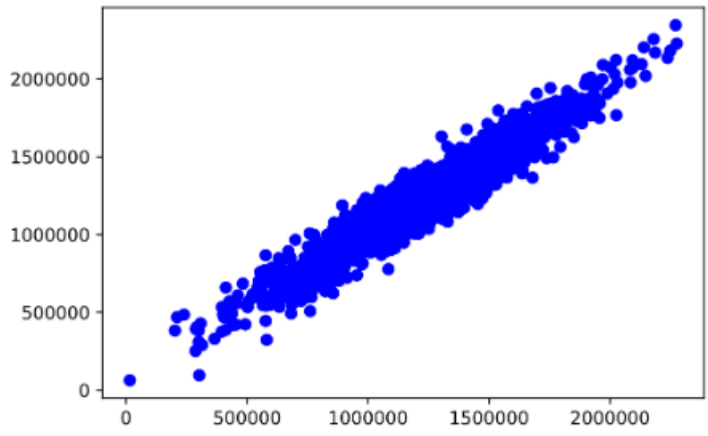
這堂課呢，主要是老師講解+帶我們寫有關預測的程式

房價預測:

老師一開始先告訴我們會用到那些程式庫，所以要先呼叫。之後鐵達尼號也會需要。接著看各種數據間的關係，也許能從中看到特定關係，然後從我們有的數據中切出拿來練習的和測試模型的。觀察結果是不是跟我們一開始想的一樣。

```
#import modules
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#%matplotlibinline
import seaborn as sns
```
呼叫程式庫


原本數據互相的關係圖



預測後的關係圖

# 鐵達尼號生存預測:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | NaN | S |
| 1 | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 2 | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S |
| 3 | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.1000 | C123 | S |
| 4 | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | 8.0500 | NaN | S |

一開始像房價預測一樣，先準備好數據(圖表)，然後觀察數據後，我們發現有很多項很可能沒用，或是無

法變成模型能使用的，像是 Name 和 Ticket，所以我們必須把他們從圖表上提出(丟掉)。

```
df.drop(['Name','Ticket'],axis=1,inplace=True)
```

然後我們觀察其他像和有沒有活下來的關係圖，有些看起來沒有關係，但也有些我們能看出一些端倪，例如存活者都稍微年輕一點、票都比較貴一點。下一步我們去看圖表上的空值，Cabin 的空值特別多，可能就不重要，或是沒辦法判斷出甚麼，我們就丟掉了

```
PassengerId      0
Survived         0
Pclass           0
Sex              0
Age            177
SibSp            0
Parch            0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

後來->

```
Age            177
Embarked         2
PassengerId      0
Survived         0
Pclass           0
Sex              0
SibSp            0
Parch            0
Fare             0
dtype: int64
```

年齡的空格呢，還算好處理，我們就把男生的中位數填入男生的空值，女生的填女生的空值。

然後 Embarked 因為只有兩個空值，就填最多的 S 就好。這樣就沒有空值了。再把性別(female and male)數值化，去除一個，done。

接下來是切割數據和訓練模型，最後觀察結果--

| | Predict not Survived | Predict Survived |
|---|---|---|
| True not Survived | 146 | 16 |
| True Survived | 29 | 77 |

Predict not Survived 為預測未存活

Predict Survived 為預測有存活

True not Survived 為真的沒存活

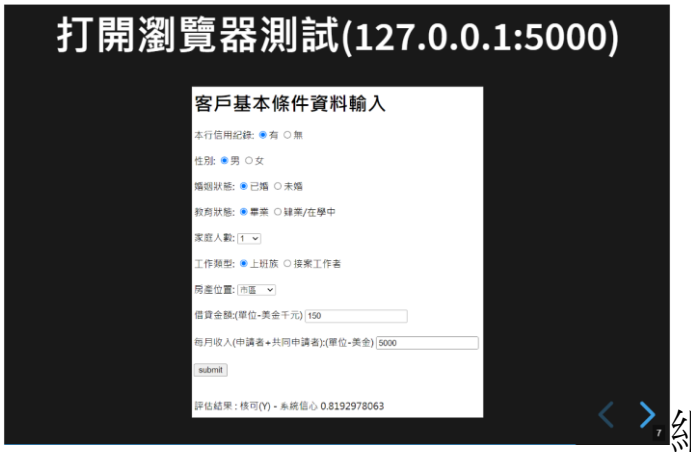
True Survived 為真的有存活

所以左上格和右下格式模型成功預測正確的。

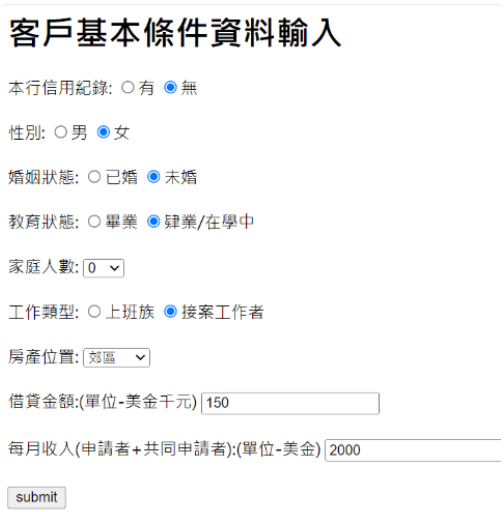
日期:2023/12/7and2023/12/14

課程內容: 銀行放款評估+使用者介面

這部份呢 老師提供給我們完成好的模型和程式碼了，所以我們大部分時間都在聽老師解釋其中程式碼的原理，和如何製作一個網站來輸入測資。



網站長這樣(核可)



另一種結果(拒絕)

# 日期:2023/12/21and2023/12/28

課程內容: 手寫數字辨識，動物分類

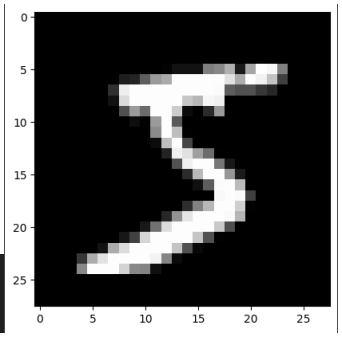
數字辨識:

一開始先準備好模型和資料切割

```
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense

import numpy as np
import matplotlib.pyplot as plt
import keras.utils as image_utils
import PIL.ImageOps

(X_train, y_train), (X_valid, y_valid) = mnist.load_data()
```

然後我們從中觀察了一筆資料

```
plt.imshow(X_train[0], cmap="gray")
y_train[0]
```




接著把資料轉換成以 0 和 1 組成的型態

```
X_train_1D = X_train.reshape(60000, 784)
X_valid_1D = X_valid.reshape(10000, 784)

X_train_1D_normal = X_train_1D / 255
X_valid_1D_normal = X_valid_1D / 255

X_train.min()
X_train.max()
X_train_1D_normal.min()
X_train_1D_normal.max()

num_categories = 10
y_train
y_train_category = image_utils.to_categorical(y_train, num_categories)
y_valid_category = image_utils.to_categorical(y_valid, num_categories)
# 測試看看 y_train_category[0]
y_train[0]
y_train_category[0]


model = Sequential()
model.add(Dense(units=512, activation="relu", input_shape=(784,)))
model.add(Dense(units=512, activation="relu"))
model.add(Dense(units=10, activation="softmax"))
model.summary()
```

這樣就可以測試裡面的數字了

```python
def predict_number(image_path):
    if "http" in image_path:
        image_path = image_utils.get_file(origin=image_path)
    test_image = image_utils.load_img(
        image_path, color_mode="grayscale", target_size=(28, 28)
    )
    test_image_inverted = PIL.ImageOps.invert(test_image)
    f, axarr = plt.subplots(1, 2)
    axarr[0].imshow(test_image, cmap="gray")
    axarr[1].imshow(test_image_inverted, cmap="gray")

    test_image_array = image_utils.img_to_array(test_image_inverted)
    test_image_array_1D = test_image_array.reshape(1, 784)
    test_image_array_1D_normal = test_image_array_1D / 255
    return np.argmax(model.predict(test_image_array_1D_normal))
```

然後導入程式後，程式就也可以檢測其他數字了。

動物分類:

一開始先載入模型

```python
from keras.applications import VGG16

model = VGG16(weights="imagenet")
model.summary()
```

接著是圖片

```python
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

def show_image(image_path):
    image = mpimg.imread(image_path)
    print(image.shape)
    plt.imshow(image)

show_image("happy_dog.jpg")
```


然後把圖片轉換格式，讓圖片變成可閱讀+預測的樣子

```python
from keras.preprocessing import image as image_utils
from keras.applications.vgg16 import preprocess_input

def load_and_process_image(image_path):
    print("Original image shape: ", mpimg.imread(image_path).shape)
    image_s = image_utils.load_img(image_path, target_size=(224, 224)
    image_s_array = image_utils.img_to_array(image_s)
    print("image_s_array:", image_s_array.shape)
    image_s_array_reshape = image_s_array.reshape(1, 224, 224, 3)
    image_forVGG16 = preprocess_input(image_s_array_reshape)
    print("Processed image shape: ", image_forVGG16.shape)
    return image_forVGG16

load_and_process_image("brown_bear.jpg")


from keras.applications.vgg16 import decode_predictions

def readable_prediction(image_path):
    show_image(image_path)
    image = load_and_process_image(image_path)
    predictions = model.predict(image)
    print("Predicted: ", decode_predictions(predictions, top=3))
    #print("Predictions:",predictions)
```

```
Original image shape:  (900, 1200, 3)
image_s_array: (224, 224, 3)
Processed image shape:  (1, 224, 224, 3)
```

然後之後呢，我們試了幾張圖片

```
✓ readable_prediction("brown_bear.jpg") ···

(900, 1200, 3)
Original image shape:  (900, 1200, 3)
image_s_array: (224, 224, 3)
Processed image shape:  (1, 224, 224, 3)
1/1 [==============================] - 0s 159ms/step
Predicted:  [[('n02132136', 'brown_bear', 0.98256314), ('n02133161', 'American_black_bear', 0.01
```


Brown_bear 即為他預測的物種，後面為她確定的程

度，最高是 1，可得知模型十分確定他為棕熊

由於我們要執行新的模型，我們先把剛剛的模型凍結住。

```
base_model.trainable = False
```

接著開始堆疊我們的模型，然後產生更多資料來訓練模型，設定訓練的資訊和驗證資料來源後，就開始訓練模型了~

```
import keras

inputs = keras.Input(shape=(224, 224, 3))
x = base_model(inputs, training=False)
x_afterPooling = keras.layers.GlobalAveragePooling2D()(x)
outputs = keras.layers.Dense(1)(x_afterPooling)
custom_model = keras.Model(inputs, outputs)
custom_model.summary()

custom_model.compile(
    loss=keras.losses.BinaryCrossentropy(from_logits=True),
    metrics=[keras.metrics.BinaryAccuracy()])


datagen = image_utils.ImageDataGenerator(
    samplewise_center=True,
    rotation_range=10,
    zoom_range=0.1,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=False,
)


train_data = datagen.flow_from_directory(
    "IsBoOrNot/IsBoOrNot/train",
    target_size=(224, 224),
    color_mode="rgb",
    class_mode="binary",
    batch_size=8,
)

valid_data = datagen.flow_from_directory(
    "IsBoOrNot/IsBoOrNot/valid",
    target_size=(224, 224),
    color_mode="rgb",
    class_mode="binary",
    batch_size=8,
)
```

```
custom_model.fit(
    train_data,
    steps_per_epoch=12,
    validation_data=valid_data,
    validation_steps=4,
    epochs=20,
)
```

```
Epoch 1/20
WARNING:tensorflow:From c:\Users\cclee\AppData\Local\Programs\Python\Python39\lib\site-packages\
...
Epoch 19/20
12/12 [==============================] - 9s 772ms/step - loss: 0.0024 - binary_accuracy: 1.0000
Epoch 20/20
12/12 [==============================] - 9s 771ms/step - loss: 5.2857e-04 - binary_accuracy: 1.0
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

# 接下來就是準備測試

```
def make_predictions(image_path):
    image = mpimg.imread(image_path)
    plt.imshow(image)
    image_resize = image_utils.load_img(image_path, target_size=(224, 224))
    image_resize_array = image_utils.img_to_array(image_resize)
    image_resize_array_reshape = image_resize_array.reshape(1, 224, 224, 3)
    image_resize_array_reshape_preprocessed = preprocess_input(
        image_resize_array_reshape
    )
    predictions = custom_model.predict(image_resize_array_reshape_preprocessed)
    return predictions
```

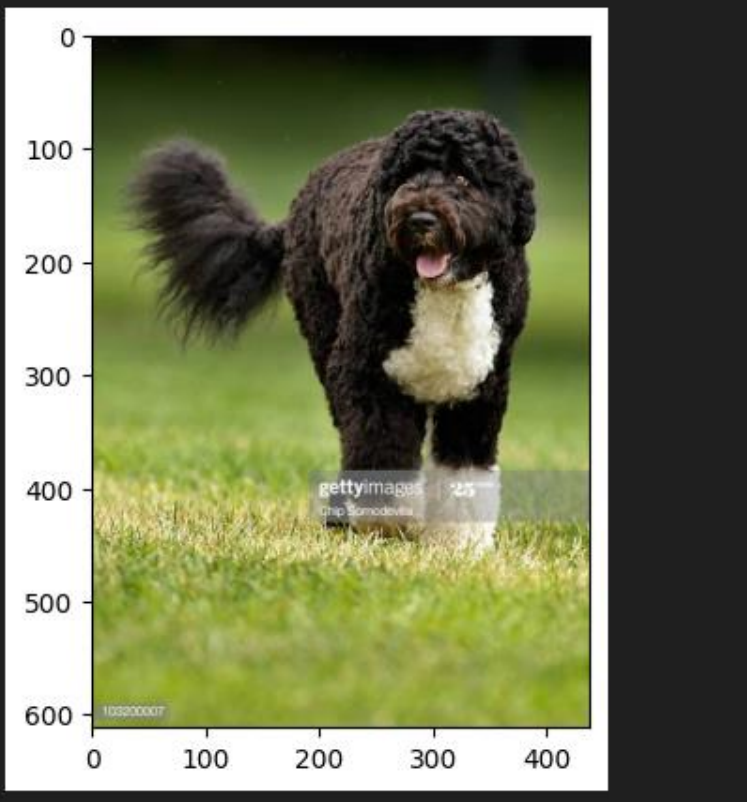
```
make_predictions("IsBoOrNot/IsBoOrNot/valid/bo/bo_20.jpg")
```

```
✓ make_predictions("IsBoOrNot/IsBoOrNot/valid/bo/bo_20.jpg") ···

1/1 [==============================] - 0s 111ms/step

array([[-13.911145]], dtype=float32)
```


把輸出會成可以容易判別的樣子後就變這樣:

```python
def tell_Bo(image_path):
    prediction = make_predictions(image_path)
    if prediction[0] < 0:
        print("It's Bo!")
    else:
        print("You are NOT Bo!")
```

```
✓ tell_Bo("IsBoOrNot/IsBoOrNot/valid/bo/bo_20.jpg")

1/1 [==============================] - 0s 110ms/step
It's Bo!
```




這樣就結束了。