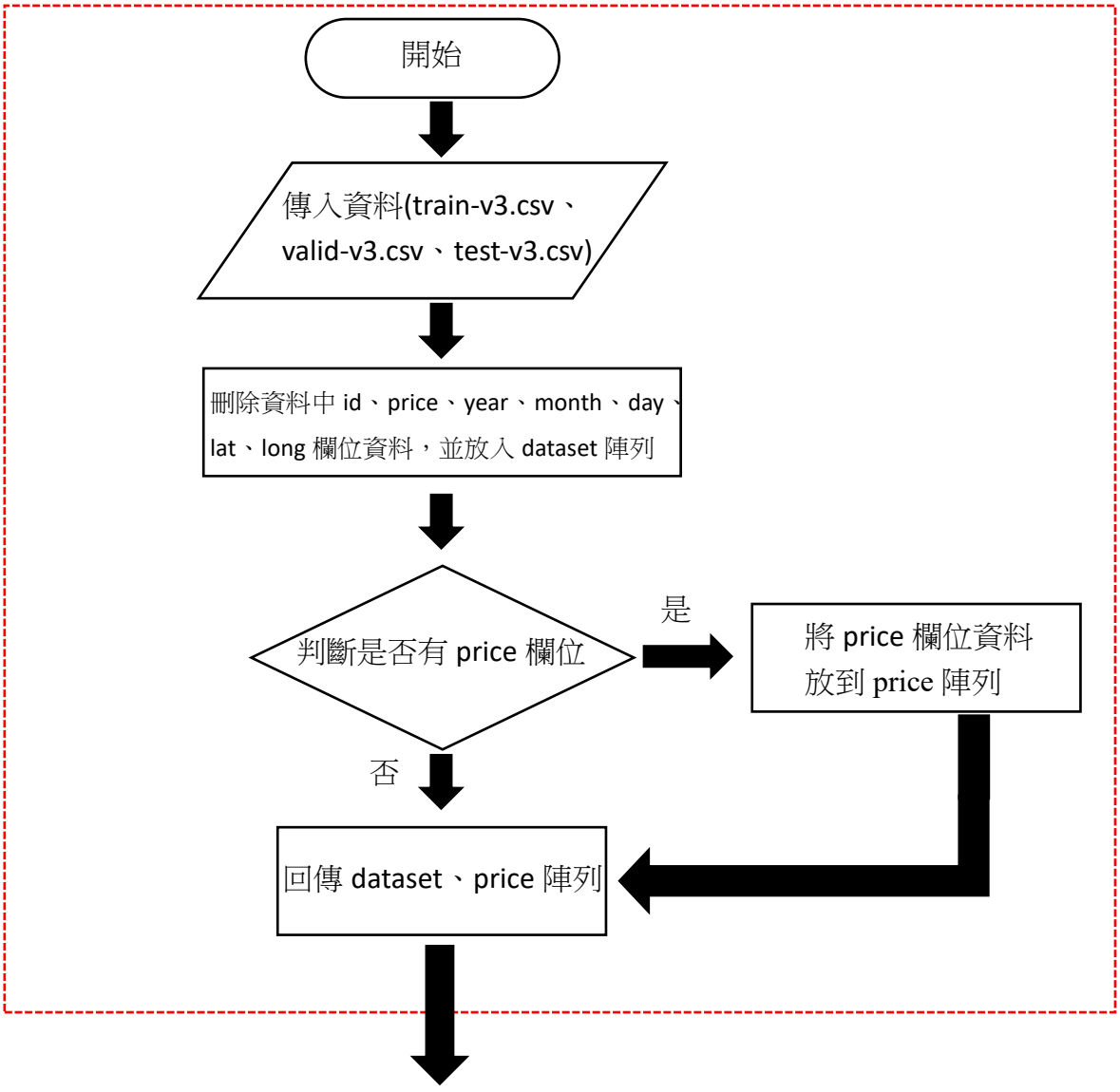
Machine Learning – Regression

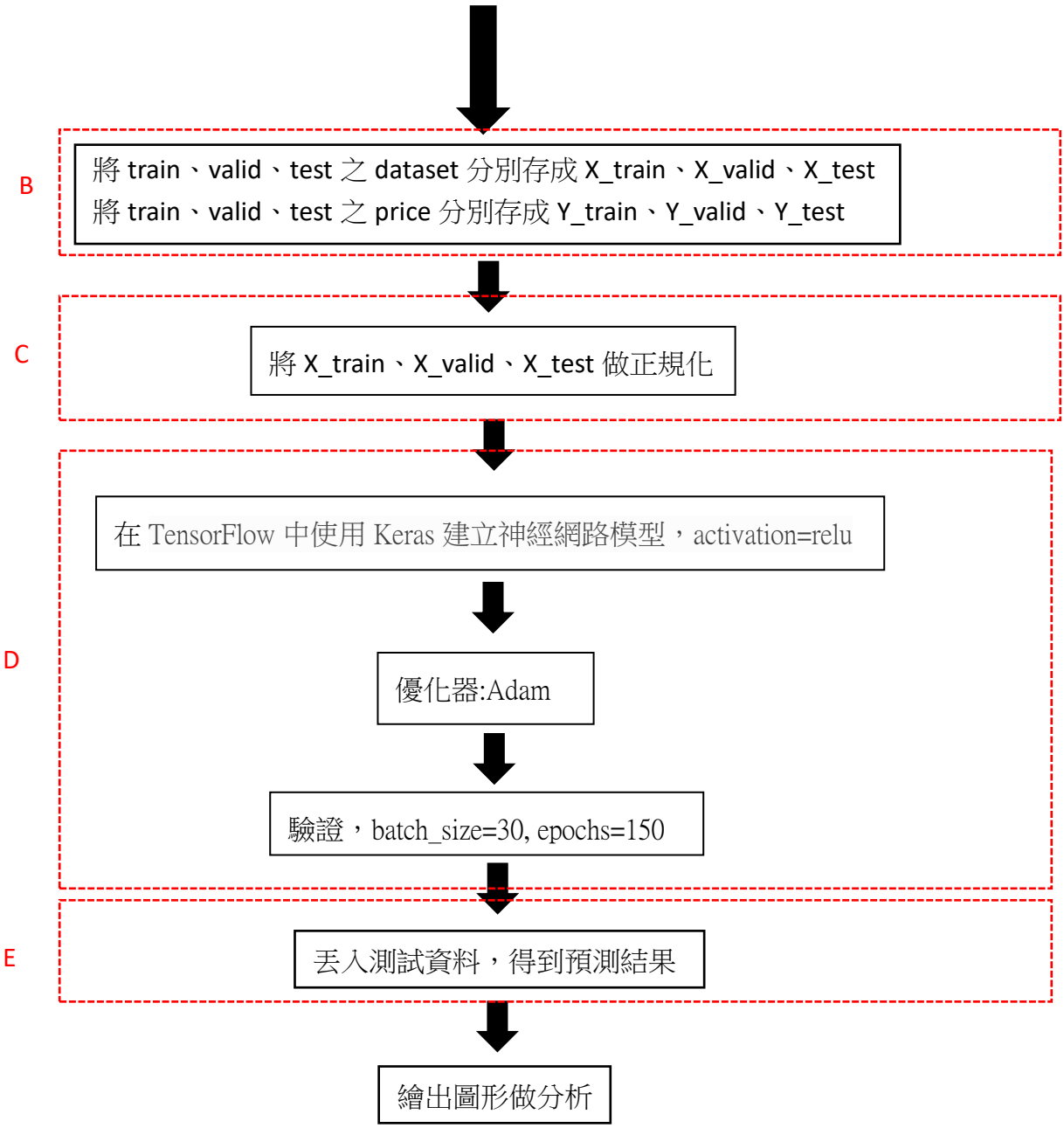
指導老師:廖元甫 教授
姓名:劉宸睿
學號:109368127

1. 做法說明:

1. 繪圖比較房價與房屋參數關係
2. 剔除與價錢較無關係的無用參數
3. 將有用參數做正規化
4. 在 TensorFlow 中使用 Keras 建立神經網路模型
5. 將正規化過的 train data 丟入神經網路中 train
6. 利用正規化過的驗證 data 驗證訓練出的神經網路，檢查 loss，確定無 overfitting
7. 將正規畫過的測試資料丟入神經網路得出預測數值
8. **程式方塊圖與寫法:**

A




B
將 train、valid、test 之 dataset 分別存成 X_train、X_valid、X_test
將 train、valid、test 之 price 分別存成 Y_train、Y_valid、Y_test
C
將 X_train、X_valid、X_test 做正規化
D
在 TensorFlow 中使用 Keras 建立神經網路模型，activation=relu
優化器:Adam
驗證，batch_size=30, epochs=150
E
丟入測試資料，得到預測結果
繪出圖形做分析

A:

```
def Data(path):
    data = pd.read_csv(path)
    price = np.array([])
    zip = pd.get_dummies(data['zipcode'])
    data = data.join(zip)
    data=data.drop(columns=['id', 'zipcode', 'sale_yr', 'sale_month', 'sale_day', 'lat', 'long'])
    dataset=np.array(data)
    if "price" in data.columns:
        price = dataset[:, 0]
        data=data.drop(columns=['price'])  #刪價錢，放到Y
        dataset=np.array(data)
    return dataset, price
```

B:

```
X_train, Y_train = Data('./train-v3.csv')
X_valid, Y_valid = Data('./valid-v3.csv')
X_test, Y_test = Data('./test-v3.csv')
```

C:

```
def normalize(train,valid,test):
    tmp=train
    mean=tmp.mean(axis=0)     #算train平均
    std=tmp.std(axis=0)       #算train標準差

    train=(train-mean)/std
    valid=(valid-mean)/std
    test=(test-mean)/std

    return train,valid,test
```

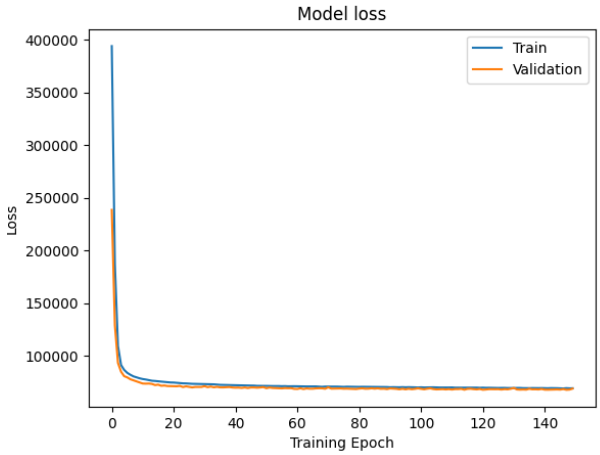
D:

```
model = keras.Sequential([
    keras.layers.Dense(40, input_dim=X_train.shape[1]),
    keras.layers.Dense(95, activation='relu'),

    keras.layers.Dense(40, activation='relu'),
    keras.layers.Dense(30, activation='relu'),
    keras.layers.Dense(2, activation='relu'),

    keras.layers.Dense(1)
])

model.compile(optimizer='adam',loss='mae')
history =model.fit(X_train, Y_train, batch_size=30, epochs=150, validation_data=(X_valid, Y_valid))
```

E:

```
Y_predict = model.predict(X_test)
```

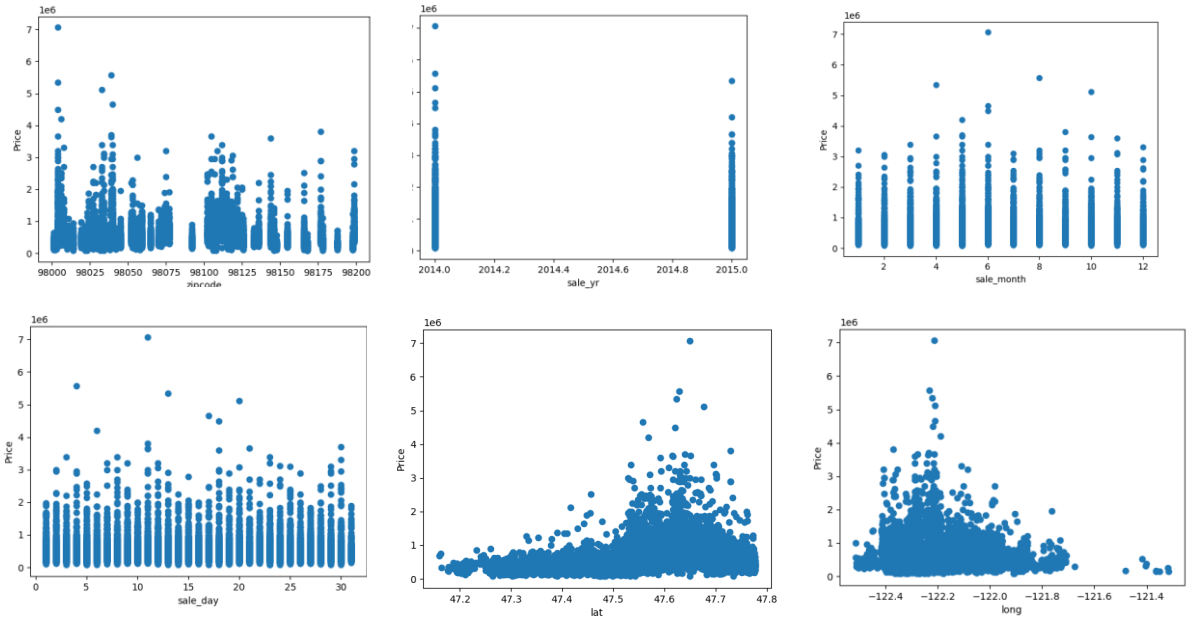
2. 畫圖做結果分析:



由上圖可知，訓練結果沒有 overfitting

3. 討論預測值誤差很大的，是怎麼回事？

剛開始，我經過畫出郵遞區號、出售年月日、經緯度對於房價關係圖認為它們 5 者對於房價是沒用的參數，於是直接將它剔除，訓練出來的 loss 約為 100000



後來我改用 one hot encode 的方法將郵遞區號轉換後再插回原始資料，並刪除原本的郵遞區號欄位，經過訓練發現提升到 68000 左右

```
433/433 [==============================] - 0s 835us/step - loss: 68101.0703 - val_loss: 67759.1875
Epoch 148/150
433/433 [==============================] - 0s 944us/step - loss: 68087.8438 - val_loss: 68247.5938
Epoch 149/150
433/433 [==============================] - 0s 871us/step - loss: 68306.5625 - val_loss: 67339.3750
Epoch 150/150
433/433 [==============================] - 0s 820us/step - loss: 68232.2891 - val_loss: 68266.8125
```

4.如何改進？

1. 適當的增加神經隱藏層數以及每層神經元數量
2. 增加隨機抓取數量以及疊帶次數