Machine Learning 2017 HW2 Report

Binary Classification

B03901156 EE3 Yu Xuan Huang

1. 請說明你實作的 generative model，其訓練方式和準確率為何？

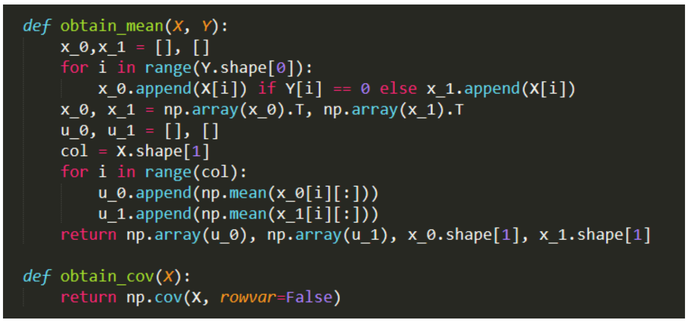
ANS. Generative model 主要是根據上課投影片實作 Gaussian distribution model, 其訓練方式:

1. Estimate n_1, n_2, u1, u2 及 covariance
2. 根據

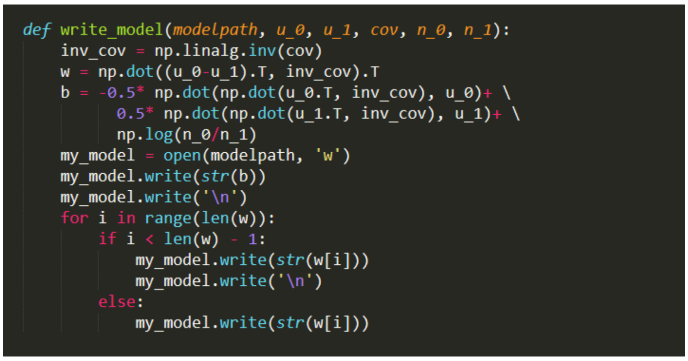
.

$$z = \underbrace{(\mu^1-\mu^2)^T\Sigma^{-1}}_{w^T} x \underbrace{-\frac{1}{2}(\mu^1)^T\Sigma^{-1}\mu^1+\frac{1}{2}(\mu^2)^T\Sigma^{-1}\mu^2+ln\frac{N_1}{N_2}}_{b}$$

求出 w (weight) 及 b (bias)



```
def obtain_mean(X, Y):
    x_0,x_1 = [], []
    for i in range(Y.shape[0]):
        x_0.append(X[i]) if Y[i] == 0 else x_1.append(X[i])
    x_0, x_1 = np.array(x_0).T, np.array(x_1).T
    u_0, u_1 = [], []
    col = X.shape[1]
    for i in range(col):
        u_0.append(np.mean(x_0[i][:]))
        u_1.append(np.mean(x_1[i][:]))
    return np.array(u_0), np.array(u_1), x_0.shape[1], x_1.shape[1]

def obtain_cov(X):
    return np.cov(X, rowvar=False)
```



```
def write_model(modelpath, u_0, u_1, cov, n_0, n_1):
    inv_cov = np.linalg.inv(cov)
    w = np.dot((u_0-u_1).T, inv_cov).T
    b = -0.5* np.dot(np.dot(u_0.T, inv_cov), u_0)+ \
        0.5* np.dot(np.dot(u_1.T, inv_cov), u_1)+ \
        np.log(n_0/n_1)
    my_model = open(modelpath, 'w')
    my_model.write(str(b))
    my_model.write('\n')
    for i in range(len(w)):
        if i < len(w) - 1:
            my_model.write(str(w[i]))
            my_model.write('\n')
        else:
            my_model.write(str(w[i]))
```

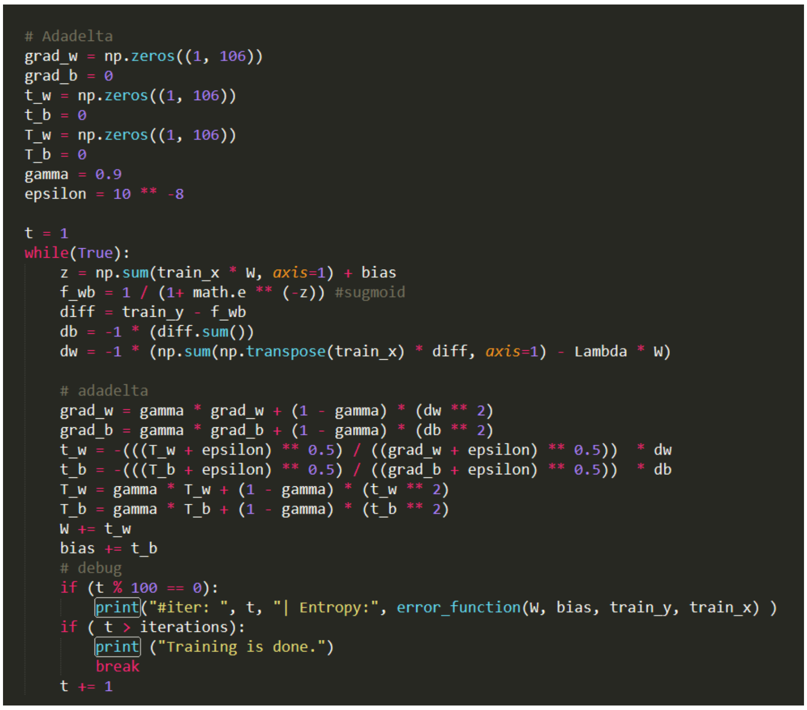
準確率: 和 discriminative model 相比，準確度較低(0.76723)，且 kaggle 上 public 和 private case 準確率皆較 discriminative model 遜色，推測其因可能為此機率分布模型不適用於這次作業的 data 類型，改用其他的機率分布模型應可改善此結果。

2. 請說明你實作的 discriminative model，其訓練方式和準確率為何？

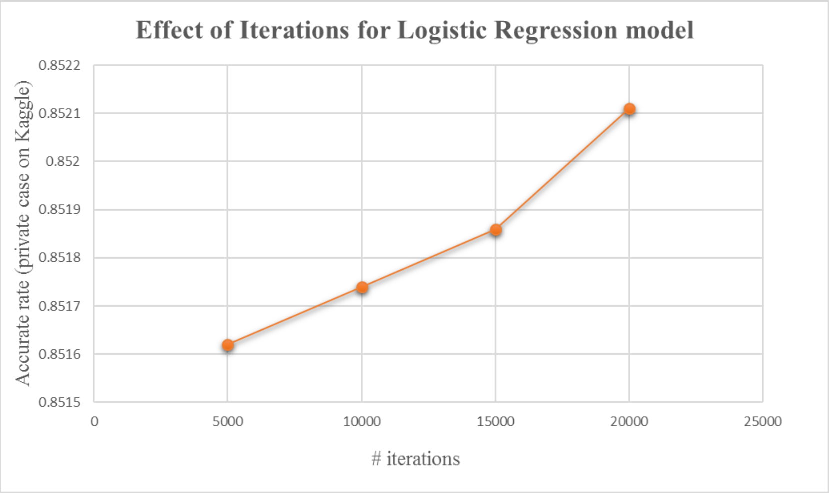
Ans. Discriminative model 主要是 Logistic regression 的實作，其訓練方式為

1. Read data: 先將 X_train 及 Y_train data 讀進 numpy 的 array 中
2. Gradient descent: 訓練過程中將 w 初始值設為 1, b 初始值設為 0 並利用 numpy 提供的矩陣運算算出 gradient
3. Update parameters: 利用 regularization, adadelta 更新參數, #iterations=20000

準確率(learning rate = 1e-8):

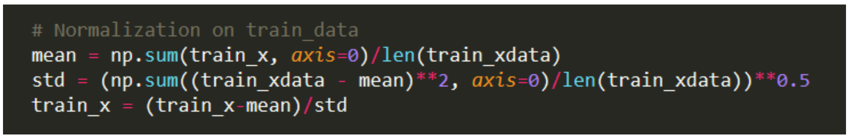


```
# Adadelta
grad_w = np.zeros((1, 106))
grad_b = 0
t_w = np.zeros((1, 106))
t_b = 0
T_w = np.zeros((1, 106))
T_b = 0
gamma = 0.9
epsilon = 10 ** -8

t = 1
while(True):
    z = np.sum(train_x * W, axis=1) + bias
    f_wb = 1 / (1+ math.e ** (-z)) #sugmoid
    diff = train_y - f_wb
    db = -1 * (diff.sum())
    dw = -1 * (np.sum(np.transpose(train_x) * diff, axis=1) - Lambda * W)

    # adadelta
    grad_w = gamma * grad_w + (1 - gamma) * (dw ** 2)
    grad_b = gamma * grad_b + (1 - gamma) * (db ** 2)
    t_w = -(((T_w + epsilon) ** 0.5) / ((grad_w + epsilon) ** 0.5))  * dw
    t_b = -(((T_b + epsilon) ** 0.5) / ((grad_b + epsilon) ** 0.5))  * db
    T_w = gamma * T_w + (1 - gamma) * (t_w ** 2)
    T_b = gamma * T_b + (1 - gamma) * (t_b ** 2)
    W += t_w
    bias += t_b
    # debug
    if (t % 100 == 0):
        print("#iter: ", t, "| Entropy:", error_function(W, bias, train_y, train_x) )
    if ( t > iterations):
        print ("Training is done.")
        break
    t += 1
```

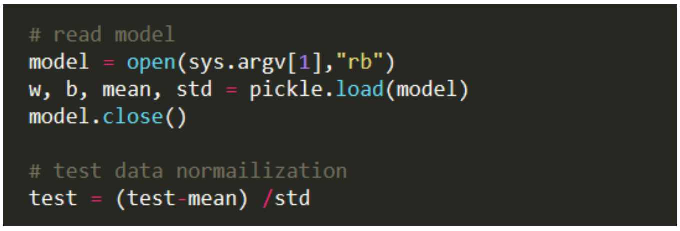


3. 請實作輸入特徵標準化(feature normalization)，並討論其對於你的模型準確率的影響。

Ans. Implementation:



```
# Normalization on train_data
mean = np.sum(train_x, axis=0)/len(train_xdata)
std = (np.sum((train_xdata - mean)**2, axis=0)/len(train_xdata))**0.5
train_x = (train_x-mean)/std
```



```
# read model
model = open(sys.argv[1],"rb")
w, b, mean, std = pickle.load(model)
model.close()

# test data normailization
test = (test-mean) /std
```

以實作 logistic regression 的 feature normalization 為例 (#iterations = 5000 , learning rate = 1e-8)，有 feature normalization 的 model 其在 kaggle 上 private case 的準確率為 0.85162，而無 normalization 的則為 0.25918，兩者相距甚遠；推測其因可能為 feature normalization 可避免 sigmoid function 產生 overflow 的問題，進而增加其準確率

4. 請實作 logistic regression 的正規化(regularization)，並討論其對於你的模型準確率的影響。

Ans. 下圖所呈現的即是在不同 Lambda 值下實作 logistic regression regularization 的準確率 (based on the accurate rate of private cases on kaggle). 由下圖觀察得知當 Lambda 值≥1 時，regularization 的確能有效提升 model 的準確率，然而 Lambda 值再增大對於準確率卻無更進一步的提升(大致以 Lambda=1 做分野)

Lambda with learning rate = 1e-8, #iterations = 5000	Accurate rate
0	0.85162
0.01	0.85162
0.1	0.85162
1	0.85198
10	0.85198

5. 請討論你認為哪個 attribute 對結果影響最大？

Ans. 在 train/test data 前，我認為 education 應該是影響最劇的 attribute，但是根據實驗的結果，capital-gain 才是最具指標性的 attribute；當去除 capital-gain 這項 feature 進行 model training 時，準確度將明顯的下降。