HW #1 Due: 3/14/2022

呂昀星 110598056

1. In the lecture, we mentioned that different algorithms have different problems. Use your own words to explain the shortcomings of each of the following methods:
   * Neural networks (particularly CNN)
   * C4.5 decision tree
   * Adaboost
   * Neural networks (particularly CNN)

缺點:

1. 需要調整參數
2. 需要大量樣本來訓練
3. 需要龐大運算(cpu)能力
4. 訓練過程如黑盒子 無法解釋其意義
5. 其中cnn 多用於影像辨識效果較佳, 不適合用於其他類型資料
   * C4.5 decision tree

缺點:

1. 對於各類別樣本數不一樣的數據, 決策數中信息增益的結果偏向那些具有更多數值的特徵
2. 對連續性資料欄位較難預測
   * Adaboost

缺點:

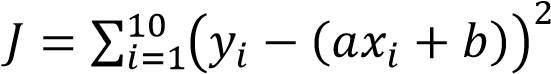
1. 對於異常樣本資料敏感, 在迭代中可能獲得較高權重,影像最終預測準確性
2. Suppose that we want to use a machine learning method to predict the rent of a house in a city. The inputs to the model include the size of the house, the built year, attached utilities, etc., and the model output is the monthly rent.
   * Suppose that a supervised-learning model is to be used. Based on the above description, between a classification model and a regression model, which one is more suitable? Explain.

由於最終output是租金, 租金屬於數值連續型資料,並不是分類或分群型資料,因此regression為較適合模型

* + Can the problem also be effectively solved by an unsupervised-learning algorithm?

可以, 可以用聚類分析

1. Consult any statistics textbook to find the closed form of the linear regression problem given in the lecture notes, i.e., find equations for *a* and *b* to minimize

.

given (𝑥𝑖, 𝑦𝑖), 1 ≤ 𝑖 ≤ 𝑁.

用最小平方法求出a,b使得J最小

簡化後

二元一次聯立方程組 解出a, b

Where

1. UC Irvine has a large repository for various kinds of data. In this problem, you are asked to use the iris dataset ([https://archive.ics.uci.edu/ml/datasets/Iris)](https://archive.ics.uci.edu/ml/datasets/Iris) to perform the experiments. Use the k-NN classifier for the classification task with *k* = 7. To begin one trial, randomly draw 70% of the instances for training and the rest for testing. Repeat the trials 10 times and compute the average accuracy. Note: you can directly import iris dataset by using sklearn without downloading from the UC Irvine repository.

acc=0.97

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

total=0

for i in range(10):

iris = datasets.load\_iris()

data\_x=iris.data

data\_y=iris.target

x\_train,x\_test,y\_train,y\_test=train\_test\_split(data\_x,data\_y,test\_size=0.3)

knn=KNeighborsClassifier(n\_neighbors=7)

knn.fit(x\_train,y\_train)

acc=knn.score(x\_test,y\_test)

print('Acc=%.2f'%acc)

total+=acc

avg=total/10

print('avg acc=%.2f'%avg)

1. Repeat problem 4, but use 60% of the data as the training set, 20% as the validation set, and the rest 20% as test set. Vary *k* from 3 to 11 and use the validation set to determine the best value of *k*. The value of *k* must be determined based on average of 10 trials. Once *k* is known, find the average accuracy of 10 trials based on the best K.

acc:0.97, k=9

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

train\_ratio = 0.6

validation\_ratio = 0.2

test\_ratio = 0.2

iris = datasets.load\_iris()

data\_x=iris.data

data\_y=iris.target

k=0

highest=0

for j in range(3,12):

total=0

for i in range(10):

x\_train,x\_test,y\_train,y\_test=train\_test\_split(data\_x,data\_y,test\_size=1-train\_ratio)

x\_train,x\_val,y\_train,y\_val=train\_test\_split(data\_x,data\_y,test\_size=test\_ratio/(test\_ratio+validation\_ratio))

knn=KNeighborsClassifier(n\_neighbors=j)

knn.fit(x\_train,y\_train)

acc=knn.score(x\_val,y\_val)

total+=acc

avg=total/10

if (avg>highest):

highest=avg

k=j

total\_test=0

for i in range(10):

x\_train,x\_test,y\_train,y\_test=train\_test\_split(data\_x,data\_y,test\_size=1-train\_ratio)

x\_train,x\_val,y\_train,y\_val=train\_test\_split(data\_x,data\_y,test\_size=test\_ratio/(test\_ratio+validation\_ratio))

knn=KNeighborsClassifier(n\_neighbors=k)

knn.fit(x\_train,y\_train)

acc=knn.score(x\_test,y\_test)

total\_test+=acc

avg\_test=total\_test/10

print('acc:%.2f, k=%d'%(avg\_test,k))