1.線性回歸-pizza 價格

id	直徑(cm)	價格
1	15	180
2	18	210
3	21	259
4	24	270
5	30	360
6	36	409
7	39	474
8	42	555
9	45	576
10	47	600

線性回歸基礎步驟主要包括：

1.導入數據集，採用列表的形式定義直徑和價格兩列數據。
2.調用 Scikit-learn 機器學習包中線性回歸模型。
3.調用 fit()函數對直徑和價格進行訓練。
4.調用 predice()函數對數據集進行預測。
5.對線性回歸算法進行評價。
6.可視化分析並繪製相關圖形，直觀的呈現算法模型的結果

求 pizza 在直徑 22cm、40 cm、50 cm 價格為何?

2.邏輯回歸

```
from sklearn import datasets
import numpy as np
from sklearn.model_selection import train_test_split

iris = datasets.load_iris() # 載入數據集
X = iris.data[:, [2, 3]]
y = iris.target # 標籤已經轉換成 0，1，2 了
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=0) # 抽出 30%當測試

# 特徵縮放
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(X_train) # 估算每個特徵的平均值和標準差
sc.mean_ # 查看特徵的平均值
sc.scale_ # 查看特徵的標準差
X_train_std = sc.transform(X_train)
# 注意：這裡我們要用同樣的參數來標準化測試集，使得測試集和訓練集之間
有可比性
X_test_std = sc.transform(X_test)

# 訓練感知機模型
from sklearn.linear_model import Perceptron
# n_iter：可以理解成梯度下降中迭代的次數
# eta0：可以理解成梯度下降中的學習率
# random_state：设置隨機種子的，為了每次迭代都有相同的訓練集順序
ppn = Perceptron(n_iter=40, eta0=0.1, random_state=0)
ppn.fit(X_train_std, y_train)

# 分類測試集，這將返回一個測試結果的數組
y_pred = ppn.predict(X_test_std)
```

請參考以上程式碼，求出準確度有多少?

3. K Nearest Neighbor

使用鳶尾花資料集進行 KNN 分類演算法

資料載入方法如下

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
import numpy as np

iris = datasets.load_iris()
```

鳶尾花資料集
https://zh.wikipedia.org/wiki/%E5%AE%89%E5%BE%B7%E6%A3%AE%E9%B8%A2%E5%B0%BE%E8%8A%B1%E5%8D%89%E6%95%B0%E6%8D%AE%E9%9B%86

1. 資料切割調用 train_test_split(data,label,size)函數

2. KNN 調用函數為 KNeighborsClassifier()

3. fit()函數用來訓練

4. predict()函數用來預測