EXAMPLE 4.1

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('dog\_breed\_classification.csv')

df

x = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

Training the K-NN model on the Training set (k=3)

from sklearn.neighbors import KNeighborsClassifier

classifier\_knn\_3 = KNeighborsClassifier(n\_neighbors = 3, metric = 'euclidean')

classifier\_knn\_3.fit(x, y)

Training the K-NN model on the Training set (k=5)

from sklearn.neighbors import KNeighborsClassifier

classifier\_knn\_5 = KNeighborsClassifier(n\_neighbors = 5, metric = 'euclidean')

classifier\_knn\_5.fit(x, y)

predicting for Ear flap length=6.3 inches, Ear flap width=3.1 inches.

classifier\_knn\_3.predict([[6.3,3.1]])

classifier\_knn\_5.predict([[6.3,3.1]])

confusion matrix

y\_pred = classifier\_knn\_5.predict(x)

from sklearn.metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y, y\_pred)

cm

accuracy\_score(y, y\_pred)

EXAMPLE 4.2

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('salaries\_scores.csv')

df

x = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

Training the K-NN model on the Training set

from sklearn.neighbors import KNeighborsRegressor

regressor\_knn = KNeighborsRegressor(n\_neighbors=5, metric='euclidean')

regressor\_knn.fit(x,y)

Predicting the salary that should be fixed for a person with age 38 and score=8.2. Use K-NN with K=3

regressor\_knn.predict([[8.2, 38]])

EXAMPLE 4.3

import numpy as np

import pandas as pd

df = pd.read\_excel('laidoff.xlsx')

df

df['Laid off ?'] = df['Laid off ?'].replace({'N':0, 'Y':1})

x = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

x = sc.fit\_transform(x)

from sklearn.neighbors import KNeighborsClassifier

knn\_classifier = KNeighborsClassifier(n\_neighbors=5, metric='euclidean')

knn\_classifier.fit(x,y)

Predicting for age=48 and salary=142,000

knn\_classifier.predict([[48, 142]])

y\_pred = knn\_classifier.predict(x)

from sklearn.metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y, y\_pred)

print(cm)

accuracy\_score(y, y\_pred)

EXAMPLE 4.4

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('sales\_data\_puppies.csv')

df

encoding all the categorical variables

df['color'] = df['color'].replace({'Brown':1, 'White':2, 'Cream':3})

df['ear flap'] = df['ear flap'].replace({'Folded':1, 'Straight':2, 'Extra Long':3})

df['fur'] = df['fur'].replace({'Thick':0, 'Curly':1})

df['adopted'] = df['adopted'].replace({'Y':1, 'N':0})

df

x = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

from sklearn.naive\_bayes import GaussianNB

classifier\_nb = GaussianNB()

classifier\_nb.fit(x, y)

Predicting for (i) Brown colour, Extra Long Ear Flaps, Curly fur

print(classifier\_nb.predict([[1,3,1]]))

Predicting for (ii) Cream colour, Straight ear flaps, Thick fur

print(classifier\_nb.predict([[3,2,0]]))

y\_pred = classifier\_nb.predict(x)

from sklearn.metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y, y\_pred)

cm

accuracy\_score(y, y\_pred)

EXAMPLE 4.5

import pandas as pd

df = pd.read\_excel('car\_preference.xlsx')

df

encoding all the categorical variables

df['Engine cc'] = df['Engine cc'].replace({'Low':1, 'Medium':2, 'High':3})

df['Ground clearance'] = df['Ground clearance'].replace({'Low':1, 'Medium':2, 'High':3})

df['Preferred –Yes/No'] = df['Preferred –Yes/No'].replace({'Y':1, 'N':0})

df

x = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

from sklearn.naive\_bayes import GaussianNB

classifier\_nb = GaussianNB()

classifier\_nb.fit(x, y)

Predicting for : High engine cc, low clearance, 4 seater

print(classifier\_nb.predict([[3,4,1]]))

y\_pred = classifier\_nb.predict(x)

from sklearn.metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y, y\_pred)

cm

accuracy\_score(y, y\_pred)

EXAMPLE 4.8

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('sales\_data\_puppies.csv')

df

encoding all the categorical variables

df['color'] = df['color'].replace({'Brown':1, 'White':2, 'Cream':3})

df['ear flap'] = df['ear flap'].replace({'Folded':1, 'Straight':2, 'Extra Long':3})

df['fur'] = df['fur'].replace({'Thick':0, 'Curly':1})

df['adopted'] = df['adopted'].replace({'Y':1, 'N':0})

df

x = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

from sklearn.tree import DecisionTreeClassifier

classifier\_dt = DecisionTreeClassifier(criterion = 'entropy', random\_state=0)

classifier\_dt.fit(x, y)

Predicting for i) White colour, Extra Long Ears, Thick fur

print(classifier\_dt.predict([[2,3,0]]))

Predicting for (ii) Brown colour, Folded ears, Thick fur

print(classifier\_dt.predict([[1,1,0]]))

y\_pred = classifier\_dt.predict(x)

from sklearn.metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y, y\_pred)

cm

accuracy\_score(y, y\_pred)