Mean temperature and change int max,min temperatures

# -\*- coding: utf-8 -\*-

"""

Created on Mon Jun 14 18:33:11 2021

@author: Mr.BeHappy

"""

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn import metrics

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

dataset=pd.read\_csv('weather.csv')

dataset.plot(x='MeanTemp',y='change',style='8')

plt.title('2nd check')

plt.xlabel('MeanTemp')

plt.ylabel('change')

plt.show();

x=dataset['MeanTemp'].values.reshape(-1,1)

y=dataset['change'].values.reshape(-1,1)

print(type(x))

n=np.size(x);

m\_x,m\_y=np.mean(x),np.mean(y)

ss\_xy=np.sum(x\*y)-n\*(m\_x\*m\_y)

ss\_xx=np.sum(x\*x)-n\*m\_x\*m\_x

b0\_1=ss\_xy/ss\_xx

b0\_0=m\_y-b0\_1\*m\_x

np.append(x,70)

np.append(y,b0\_0+b0\_1\*70)

y\_pred=b0\_0+b0\_1\*x

print("intercept:",b0\_0)

print("slope: ",b0\_1)

plt.scatter(x,y)

plt.plot(x,y\_pred,color='r',marker='o')

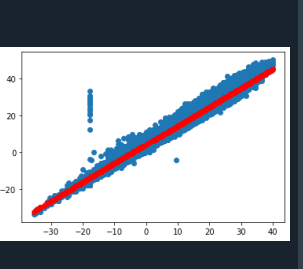
from sklearn.metrics import r2\_score

r2=r2\_score(y,y\_pred)

print("r2 =",r2)

r=r2\*\*0.5

print ("r=",r)



<class 'numpy.ndarray'>

intercept: 4.228344641952948

slope: 1.0180770080234096

r2 = 0.9390544924832133

r= 0.9690482405346048

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////

IRIS

# -\*- coding: utf-8 -\*-

"""

Created on Tue Jun 15 12:13:32 2021

@author: Mr.BeHappy

"""

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

leaves=pd.read\_csv('C:/Users/rajar/Documents/.summercoding/ML/to train\iris/Iris.csv')

data=leaves

leaves.head()

from sklearn.preprocessing import LabelEncoder,OneHotEncoder

from sklearn.compose import ColumnTransformer

le=LabelEncoder()

data['Species']=le.fit\_transform(data['Species']);

columnTransformer=ColumnTransformer([('encoder',OneHotEncoder(),[3])],remainder='passthrough')

data=np.array(columnTransformer.fit\_transform(data), dtype = np.float64)

X=data[:,:-1]

Y=data[:,-1]

#extracting targets

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

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,Y,test\_size=0.3,random\_state=0)

lin\_reg=LinearRegression()

lin\_reg.fit(X\_train,y\_train)

y\_pred=lin\_reg.predict(X\_test)

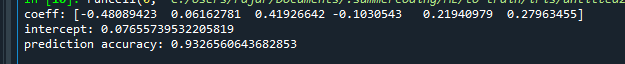
print("coeff:",lin\_reg.coef\_)

print("intercept:",lin\_reg.intercept\_)

from sklearn.metrics import r2\_score

score=r2\_score(y\_pred,y\_test)

print('prediction accuracy:',score)



#profit prediction

# -\*- coding: utf-8 -\*-

"""

Created on Tue Jun 15 11:18:42 2021

@author: Mr.BeHappy

"""

import pandas as pd;

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import OneHotEncoder

companies=pd.read\_csv('C:/Users/rajar/Documents/.summercoding/ML/to train/profit prediction/1000\_companies.csv')

data=companies

companies.head()

x=companies['R&D Spend'].values.reshape(-1,1)

y=companies['State'].values.reshape(-1,1)

from sklearn.preprocessing import LabelEncoder

from sklearn.compose import ColumnTransformer

le=LabelEncoder()

data['State']=le.fit\_transform(data['State'])

columnTransformer = ColumnTransformer([('encoder',OneHotEncoder(),[3])],remainder='passthrough')

data = np.array(columnTransformer.fit\_transform(data), dtype = np.float64)

#extracting features

X=data[:,:-1]

#extracting targets

Y=data[:,-1]

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

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,Y,test\_size=0.3,random\_state=0)

lin\_reg=LinearRegression()

lin\_reg.fit(X\_train,y\_train)

y\_pred=lin\_reg.predict(X\_test)

print("coeff:",lin\_reg.coef\_)

print("intercept:",lin\_reg.intercept\_)

from sklearn.metrics import r2\_score

score=r2\_score(y\_pred,y\_test)

print('prediction accuracy:',score)

import statsmodels.api as sm

X = sm.add\_constant(X)

model= sm.OLS(Y, X).fit()

model.summary()

