**Practical No.3**

**1. Predict Canada’s per capita income in 2020. There is an exercise folder here on github at the same level as this notebook, download that and you will find the canada\_per\_capita\_income.csv file. Using this build a regression model and predict the per capita income of canadian citizens in year 2020**

**Implementation:**

**Program and Outputs:**

#Linear Regression Using Python

import numpy as np

from matplotlib import pyplot as plt

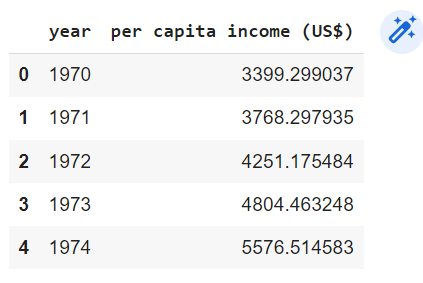
import pandas as pd

from sklearn import linear\_model

url='https://raw.githubusercontent.com/codebasics/py/master/ML/1\_linear\_reg/Exercise/canada\_per\_capita\_income.csv'

lr2=pd.read\_csv(url)

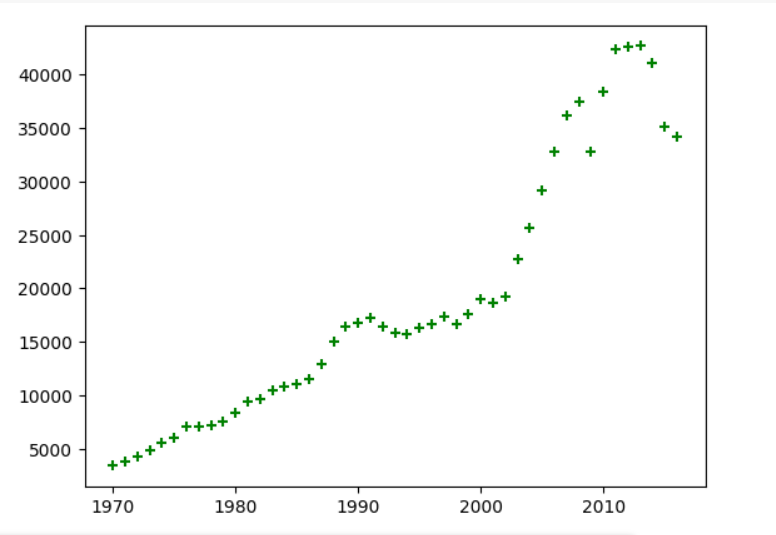
lr2.head()

****

lr2 = lr2.rename(columns={'per capita income (US$)': 'Income'})

plt.scatter(lr2.year,lr2.Income,color='g',marker='+')

plt.show()

****

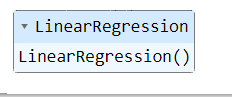
lr3=lr2.drop('Income',axis=1)

lr3.head()

****

rg1=linear\_model.LinearRegression()

rg1.fit(lr3,lr2.Income)

****

rg1.predict([[2020]])

****

rg1.coef\_

****

rg1.intercept\_

****

#y=mx+b

828.46507522\*2020+(-1632210.7578554575)

****

**2. Download employee retention dataset from here:**

**https://www.kaggle.com/giripujar/hr-analytics.**

**Now do some exploratory data analysis to figure out which variables have direct and clear impact on employee retention (i.e. whether they leave the company or continue to work)**

**Plot bar charts showing impact of employee salaries on retention**

**Plot bar charts showing correlation between department and employee retention**

**Now build logistic regression model using variables that were narrowed down in**

**step 1**

**Measure the accuracy of the model**

**Implementation:**

**Program And Output:**

#logitsic regression- Binary classification import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

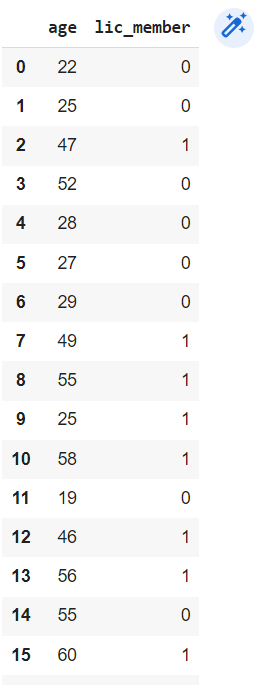
import pandas as pd

from sklearn import linear\_model

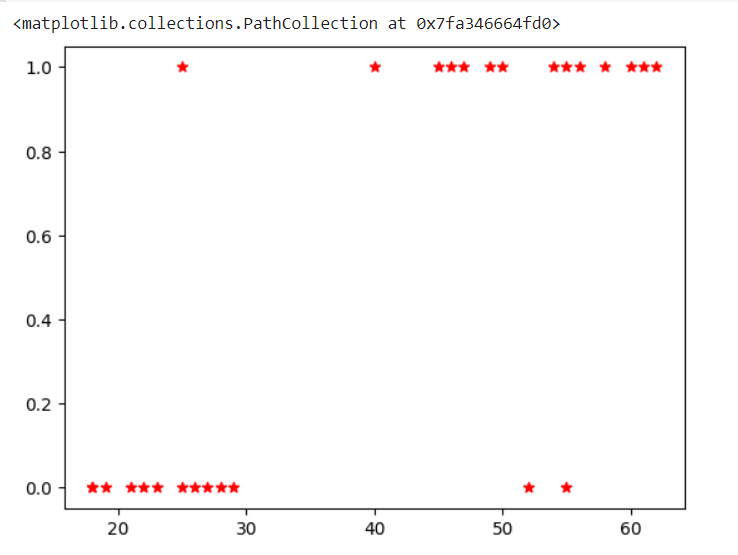
url='http://raw.githubusercontent.com/WamanParulekar/AIML/main/lic.csv'

lic = pd.read\_csv(url)

lic

****

plt.scatter(lic.age,lic.lic\_member,marker="\*",color='r')

****

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

x\_train, x\_test, y\_train, y\_test = train\_test\_split(lic[['age']],lic.lic\_member, train\_size=0.8)

len(x\_test)

****

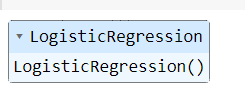
len(y\_test)

****

from sklearn.linear\_model import LogisticRegression

lr = LogisticRegression()

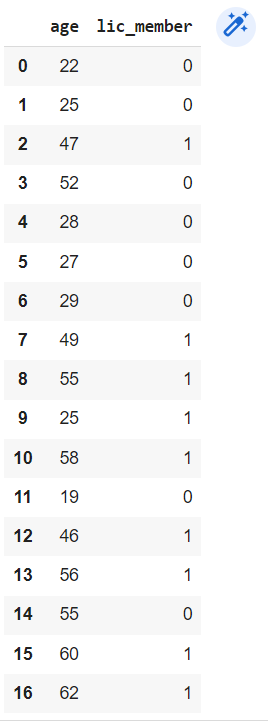
lr.fit(x\_train , y\_train)

****

lr.predict(x\_test)

****

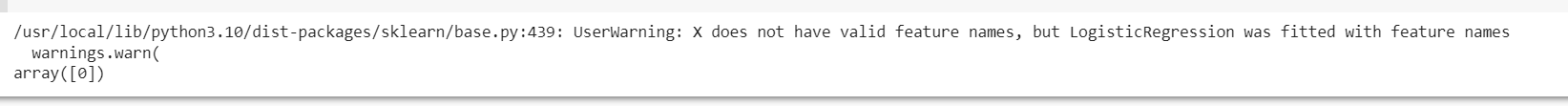
lic

****

lr.score(x\_test,y\_test)

****

lr.predict([[30]])

****

**3. Using K nearest neighbors classification predict type of flower**

**given ‘sepal\_length’, ‘sepal\_width’,’petal\_length’, ‘petal\_width’ = 4.8,3.0,1.5,0.3**

**Implementation:**

**Program:**

#KNN classification

import pandas as pd

import numpy as np

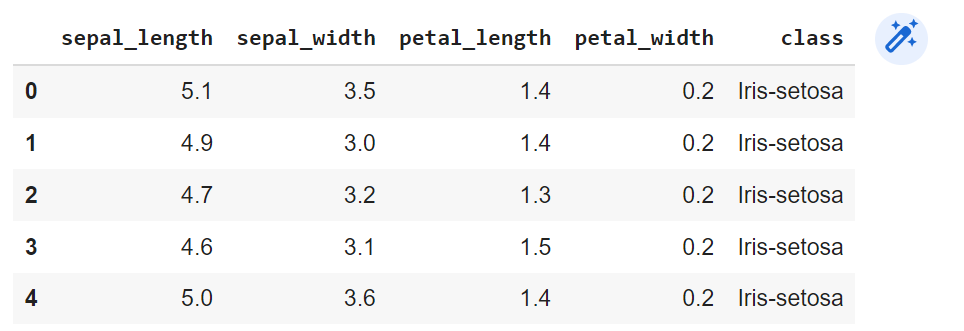
from sklearn import linear\_model

import matplotlib.pyplot as plt

url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

iris = pd.read\_csv(url, names=['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width', 'class'])

iris.head()



iris1 = iris[:50]

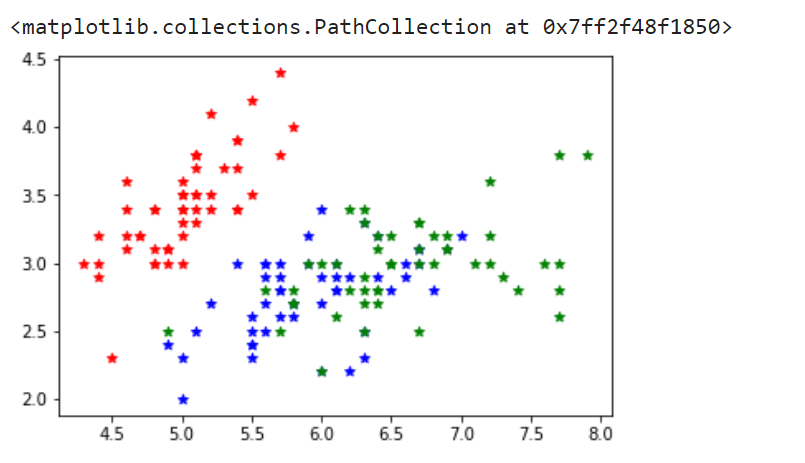
iris2 = iris[50:100]

iris3 = iris[100:]

plt.scatter(iris1.sepal\_length,iris1.sepal\_width,marker="\*",color='r')

plt.scatter(iris2.sepal\_length,iris2.sepal\_width,marker="\*",color='b')

plt.scatter(iris3.sepal\_length,iris3.sepal\_width,marker="\*",color='g')



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

X = iris.drop('class',axis=1)

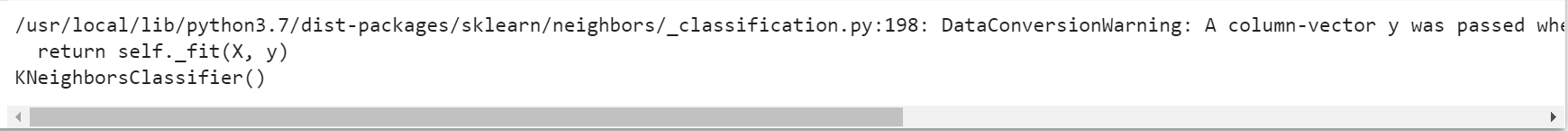
y = iris[['class']]

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,train\_size=0.8)

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()

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



knn.predict([[4.8,3.0,1.5,0.3]])

