**3.1.** **Mô tả tập dữ liệu**

Bộ dữ liệu về hoa Iris được thu thập bởi Edgar Anderson – 1 nhà thực vật học người mỹ - trên bán đảo Gaspe (là một bán đảo dọc theo bờ biển phía nam của sông Saint Lawrence ở Quebec , Canada , mở rộng vào vịnh Saint Lawrence) Sau đó bộ dữ liệu trên được Ronald Aylmer Fisher – 1 nhà thống kê người Anh đồng thời là 1 nhà di truyền học rút gọn lại, bỏ đi các thuộc tính mang tính chất mô tả,không đo đạt được giá trị theo số học (như thời điểm gieo trồng, thời điểm nở hoa, màu sắc hoa, tốc độ tăng trưởng, điều kiện ánh sáng, đất nước và độ ẩm,…) Bộ dữ liệu sau khi được rút gọn bao gồm 5 thuộc tính : 4 thuộc tính kiểu số: chiều dài đài hoa, chiều rộng đài hoa, chiều dài cánh hoa, chiều rộng cánh hoa (đơn vị là cm) 1 thuộc tính còn lại là tên của loài hoa Iris (có 3 loài tất cả : Iris Setosa, Iris Versicolour, Iris Virginica)



Tỉ lệ phân chia loài hoa

**3.2 Làm sạch dữ liệu:**

Thiếu giá trị: khi xảy ra sự thiếu thông tin ở 1 thuộc tính nào đó trong 1 bản ghi của bộ dữ liệu thu thập, khi mà tính đảm bảo số bản ghi chia đều cho 3 loài trên đã có ( 33,3%) và số lượng bản ghi thiếu là ít ta có thể áp dụng phương pháp loại bỏ.(thực tế trong bộ dữ liệu trên, không tồn tại giá trị thuộc tính bị thiếu hay không được điền).mặt khác khi cần phải điền vào các giá trị thiếu, ta có thể áp dụng phương pháp điền tự động, thay thế các giá trị thiếu bằng giá trị trung bình đã được thống kê ở trên Nhiễu dữ liệu: khi xuất hiện 1 giá trị bất ngờ nào đó, đột nhiên vượt quá các giá trị biên đã được thống kê, ta có thể sửa lại giá trị đó thành các giá trị ở vùng biên theo bảng thống kê ở trên

**3.3. Chọn lọc dữ liệu:**

Tích hợp và dư thừa dữ liệu: do dữ liệu trên thu thập từ 1 nguồn duy nhất nên việc tích hợp là không cần thiết, các thuộc tính của dữ liệu là độc lập nhau, không có mối quan hệ tương quan nào, các thuộc tính đã được rút gọn chọn lọc nên không cần việc phân tích dư thừa dữ liệu

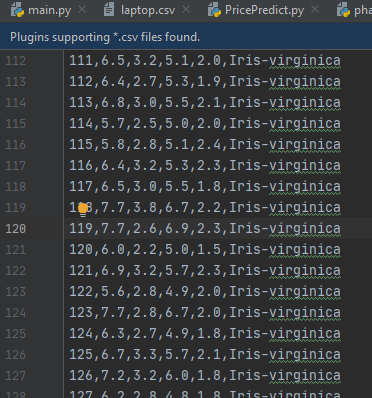
Biến đổi dữ liệu:

Biến đổi dữ liệu: ta sử dụng phương pháp chuẩn hóa dữ liệu về khoảng giá trị [0,1], phần này sẽ được nói rõ hơn ở phần sau tùy vào phương pháp khai phá dữ liệu được lựa chọn.

**3.4. Rút gọn dữ liệu:**

:Như đã nói từ đầu, bộ dữ liệu đã được loại bỏ đi các thuộc tính mang tính chất mô tả, chỉ giữ lại 1 thuộc tính mô tả duy nhất là tên loài Iris cụ thể.các thuộc tính mang tính số học còn lại vẫn đảm bảo được tính phân loại (đã được Ronald Aylmer Fisher – 1 nhà di truyền học người Anh – chọn lọc)

**3.5 Dữ liệu:**

****

Hình 3.5.1: Dữ liệu hoa

**3.6 Mô tả về kỹ thuật học máy:**

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import seaborn as sns # data visualisation and plotting

import matplotlib.pyplot as plt # data plotting

import warnings

# Seaborn default configuration

sns.set\_style("darkgrid")

# set the custom size for my graphs

sns.set(rc={'figure.figsize':(8.7,6.27)})

# filter all warnings

warnings.filterwarnings('ignore')

# set max column to 999 for displaying in pandas

pd.options.display.max\_columns=999

data = pd.read\_csv('Iris-1.csv')

data.head()

data.info()

data.describe()

data['Species'].value\_counts()

rows, col = data.shape

print("Rows : %s, column : %s" % (rows, col))

#Trực quan hóa dữ liệu

# snsdata = data.drop(['Id'], axis=1)

# g = sns.pairplot(snsdata, hue='Species', markers='x')

# g = g.map\_upper(plt.scatter)

# g = g.map\_lower(sns.kdeplot)

#

# sns.violinplot(x='SepalLengthCm', y='Species', data=data, inner='stick', palette='autumn')

# plt.show()

# sns.violinplot(x='SepalWidthCm', y='Species', data=data, inner='stick', palette='autumn')

# plt.show()

# sns.violinplot(x='PetalLengthCm', y='Species', data=data, inner='stick', palette='autumn')

# plt.show()

# sns.violinplot(x='PetalWidthCm', y='Species', data=data, inner='stick', palette='autumn')

# plt.show()

#Multivariate Linear Regression Model(Mô hình hồi quy tuyến tính đa biến)

mapping = {

'Iris-setosa' : 1,

'Iris-versicolor' : 2,

'Iris-virginica' : 3

}

X = data.drop(['Id', 'Species'], axis=1).values # Input Feature Values

y = data.Species.replace(mapping).values.reshape(rows,1) # Output values

X = np.hstack(((np.ones((rows,1))), X))# Adding one more column for bias

np.random.seed(0) # Let's set the zero for time being

theta = np.random.randn(1,5) # Setting values of theta randomly

print("Theta : %s" % (theta))

iteration = 10000

learning\_rate = 0.003 # If you are going by formula, this is actually alpha.

J = np.zeros(iteration) # 1 x 10000 maxtix

# Let's train our model to compute values of theta

for i in range(iteration):

J[i] = (1/(2 \* rows) \* np.sum((np.dot(X, theta.T) - y) \*\* 2 ))

theta -= ((learning\_rate/rows) \* np.dot((np.dot(X, theta.T) - y).reshape(1,rows), X))

prediction = np.round(np.dot(X, theta.T))

ax = plt.subplot(111)

ax.plot(np.arange(iteration), J)

ax.set\_ylim([0,0.15])

plt.ylabel("Cost Values", color="Green")

plt.xlabel("No. of Iterations", color="Green")

plt.title("Mean Squared Error vs Iterations")

plt.show()

ax = sns.lineplot(x=np.arange(iteration), y=J)

plt.show()

ax = plt.subplot(111)

ax.plot(np.arange(1, 151, 1), y, label='Orignal value', color='red')

ax.scatter(np.arange(1, 151, 1), prediction, label='Predicted Value')

plt.xlabel("Dataset size", color="Green")

plt.ylabel("Iris Flower (1-3)", color="Green")

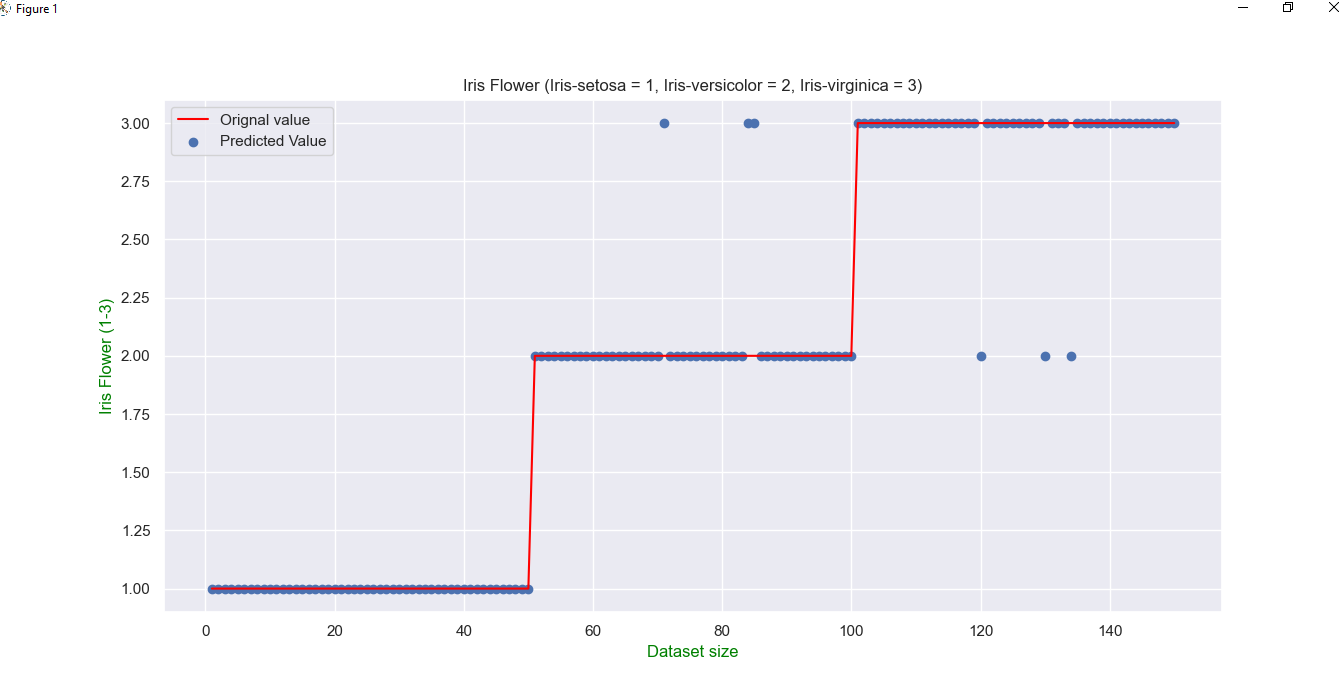
plt.title("Iris Flower (Iris-setosa = 1, Iris-versicolor = 2, Iris-virginica = 3)")

ax.legend()

plt.show()

accuracy = (sum(prediction == y)/float(len(y)) \* 100)[0]

print("The model predicted values of Iris dataset with an overall accuracy of %s" % (accuracy))



Hình 3.6.1 : Kết quả

**3.7 Kết luận:**

Chúng ta đã hoàn thành bài toán dự đoán hoa dựa vào các thuộc tính độ dài, rộng của đài hoa, cánh hoa. Qua đó ta cũng nhận thấy được tầm quan trọng cũng như các ứng dụng vô cùng cần thiết của hồi quy tuyến tính trong việc xử lí các bài toán liên quan tới dự đoán.