**Машин сургалт**

(Лаборатори №2)

Э.Номундарь 16B1SEAS2768

ХШУИС, МКУТ-ийн оюутан, namuna91nomuna@gmail.com

**1. УДИРТГАЛ**

Python хэлний numpy, pandas сангийн талаар судалж зарим функцүүдийн талаар тодорхойлж бичнэ.Төслийг зааврын дагуу хийж гүйцэтгэж ажиллуулж туршиж үзнэ.

**2. ЗОРИЛГО**

Python хэлний талаар судалж numpy, pandas сангийн талаар судалж, өгөгдсөн төслийн зааврын дагуу хийж гүйцэтгэн тайлан бичнэ. Үүний тулд дараах зорилтуудыг тавьж ажилласан:

1. Numpy, pandas санг судлах

2.Төслийн зааврын дагуу алхам алхамаар гүйцэтгэх

**3. ОНОЛЫН СУДАЛГАА**

3.1numpy

Numpy нь Python хэл дээрх шинжлэх ухааны тооцоололд зориулсан үндсэн сан юм [2]. Үүнд :

1. N хэмжээст хүснэгтэн объект
2. Хүснэгтийн хэлбэр хөрвүүлэх, тооцоолол хийх зэрэг төвөгтэй функцүүд
3. Шугаман алгебр, Фурегийн хувиргалт, санамсаргүй тоо гэх зэрэг боломжууд багтдаг.

Үүнээс гадна ерөнхий төрөлтэй өгөгдлийн олон хэмжээст агуулагч хэлбэрээр ашиглан төрөл бүрийн өгөгдлийн сантай ажиллах боломжтой.

3.2 pandas

1. Pandas сангийн read функцийг ашиглан csv файлаас өгөгдлийн уншина. Өгөгдлийг уншихдаа файлын замыг зааж эсвэл url-ийг зааж өгч өгөгдлийг уншина.

2. pandas-ийн shape функц нь өгөгдлийн хэмжээсийг буюу хэдэн мөр баганын хэмжээтэй байгааг буцаадаг функц.

3. Pandas-ийн head функцийг ашиглан параметрээр дамжуулах утгын хэмжээгээр өгөгдлийн эхний хэдэн мөрийг авахыг заана.

4.describe функц нь баганын хэмжээ, хүснэгтийн хамгийн их, бага утга мөн дундаж утга зэргийг буцаадаг.

5.groupby функц нь параметрт дамжуулсан аргументтэй баганын утгуудыг бүлэглэнэ.

6.plot функц нь өгөдлийн тархалтыг дүрсэлдэг бөгөөд праметртэй хэлбэр, layout зэрэг утгуудыг авдаг.

7.hist функц нь өгөгдлийг багана тус бүртээ хэрхэн тархалттай байгааг графикаар дүрсэлдэг.

8.scatter\_matrix функцийг ашиглан өгөгдлийг багана тус бүрийн хувьд ямар тархалттай байгааг матриц хэлбэрээр мөн графикаар дүрсэлнэ.

4. ХЭРЭГЖҮҮЛЭЛТ

Сургалтын өгөгдлийг интернэт эх сурвалжаас уншсан бөгөөд pandas-ийн read функцийг ашиглан url-ийг зааж өгч мөн баганын нэрүүдийг өгсөнөөр уншиж ирсэн өгөгдлүүдийг багана тус бүрт харгалзуулан уншиж байна. Баганын тоог дутуу өгвөл алдаа зааж байсан бол илүү тодорхойлсон үед Nan буюу хоосон утгаар дүүргэж байна.

*url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"*

*names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']*

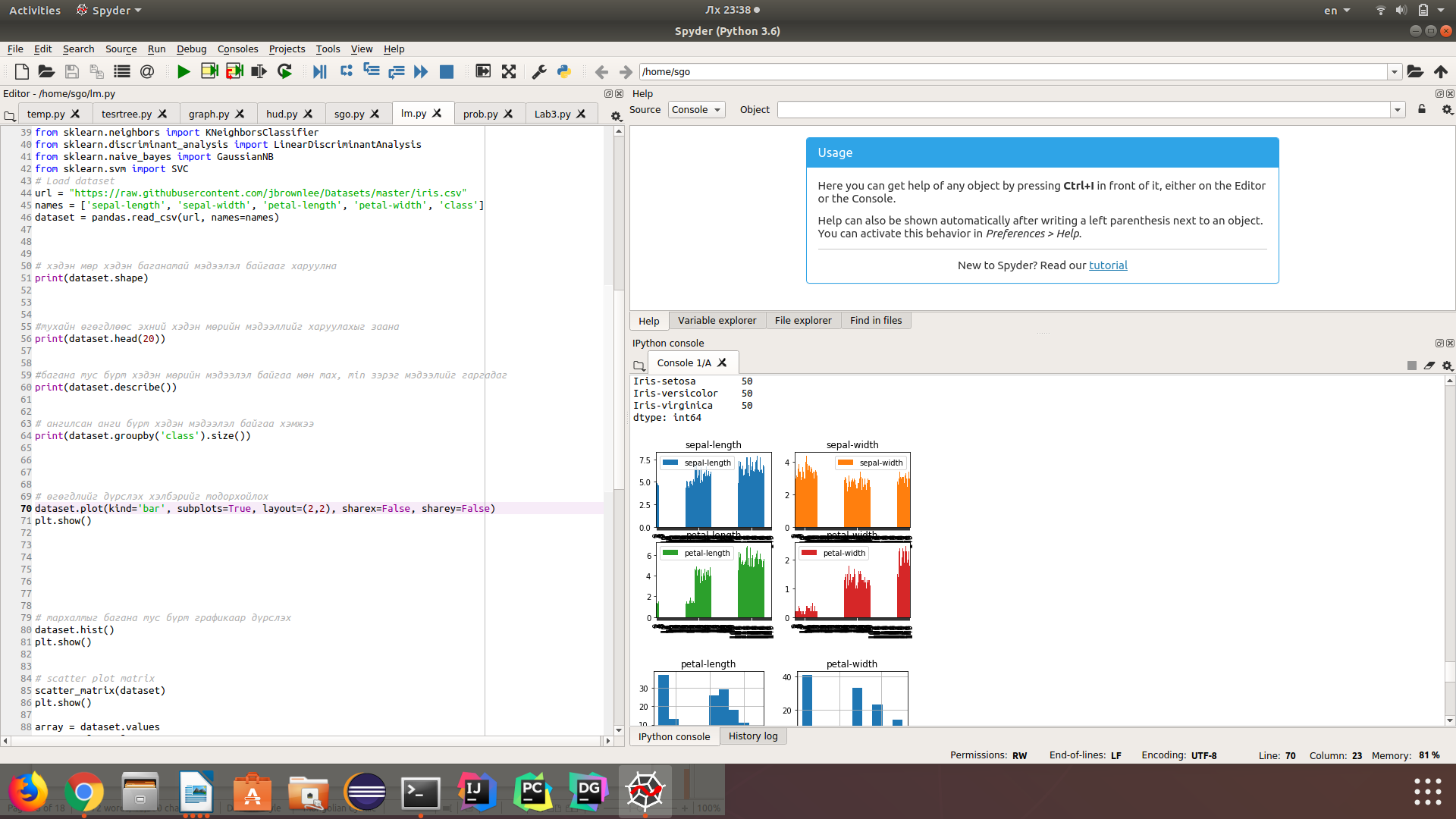
*dataset = pandas.read\_csv(url, names=names)*

Өгөгдлийг дүрслэх хэлбэрийг тодорхойлохдооо

# өгөгдлийг дүрслэх хэлбэрийг тодорхойлох

dataset.plot(kind='bar', subplots=True, layout=(2,2), sharex=False, sharey=False)

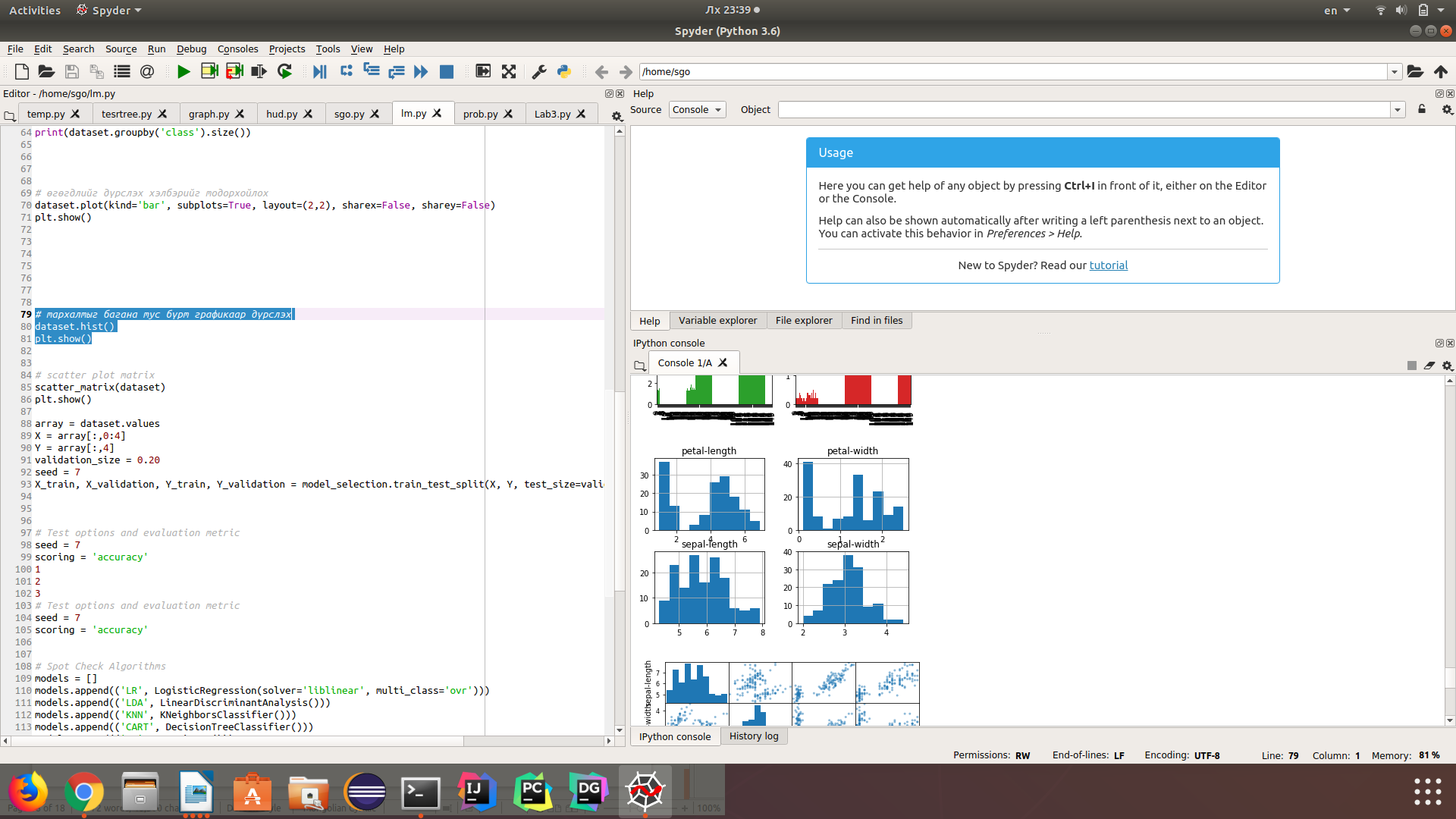
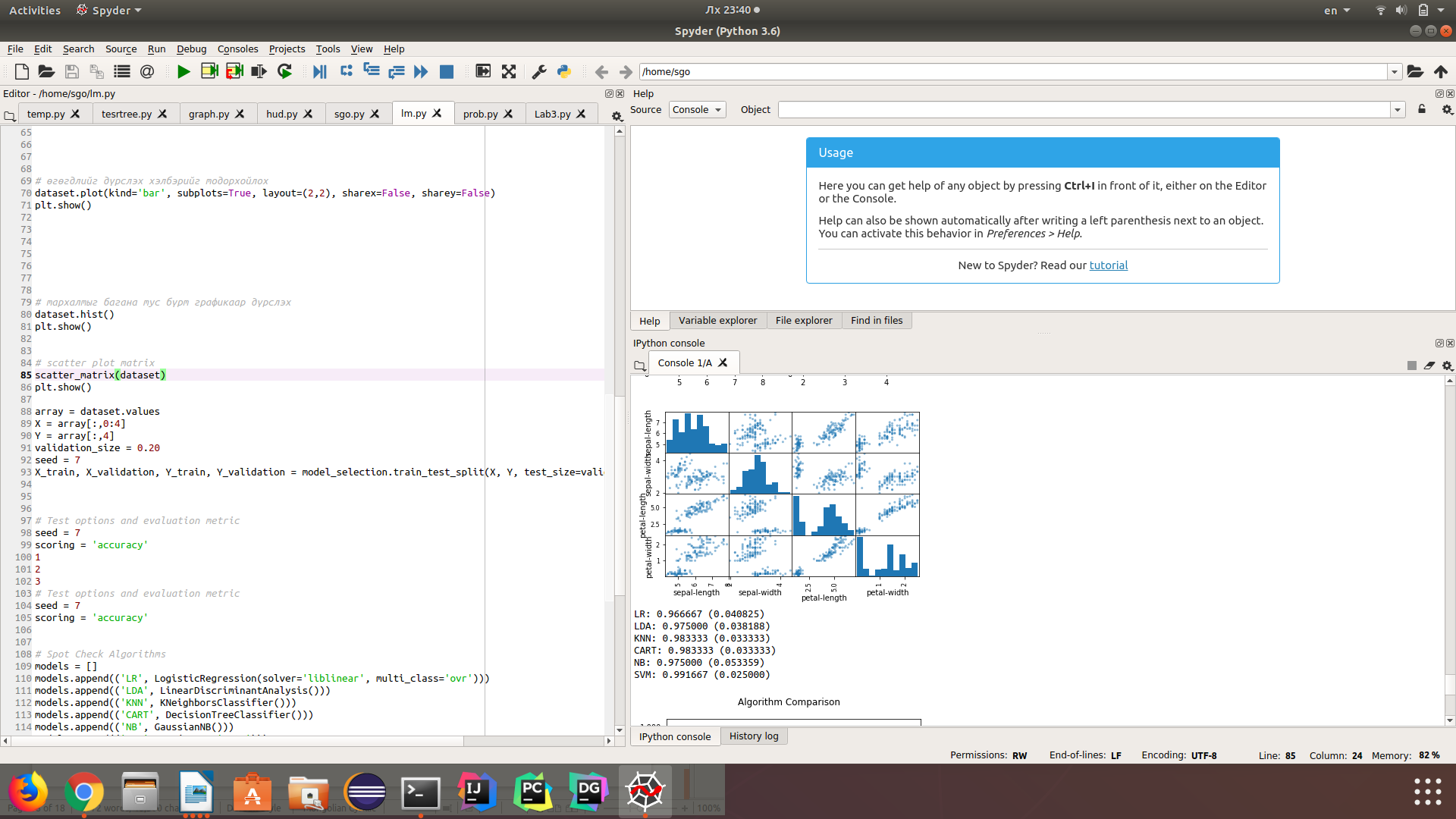
plt.show()



# тархалтыг багана тус бүрт графикаар дүрслэх

dataset.hist() scatter\_matrix(dataset)

plt.show()

5. ДҮГНЭЛТ

Энэ Лабораторийн хүрээнд numpy, pandas зэрэг санг ашиглан сургалтын өгөгдлийг хэрхэн унших, өгөгдлийн хэмжээ, бүлэг ангилалыг хэрхэн мэдэх мөн сургалтын өгөгдлйин тархалтыг графикаар дүрслэх аргыг мэдлээ. Мөн тархалтын алгоритмуудыг туршиж үзэн тэдгээрийн үр дүнг графикаар дүрслэсэн.Машин сургалтыг хэрэгжүүлж буй хэд хэдэн алгоритмуудыг танилтын хувь болон алдаагаар нь үнэлэхэд миний бодлоор KNN алгоритм танилтын хувь болон алдааны үзүүлэлт нь хамгийн дундаж танилттай байна гэж үзлээ.

6**. АШИГЛАСАН МАТЕРИАЛ**

1.<https://machinelearningmastery.com/machine-learning-in-python-step-by-step/>

2.<http://cs231n.github.io/python-numpy-tutorial/>

7. **ХАВСРАЛТ**

#pythonii version heden bitiinh

# zereg uzuuleltiig delgetsleh

import sys

print('Python: {}'.format(sys.version))

# scipy versioniigg hevleh

import scipy

print('scipy: {}'.format(scipy.\_\_version\_\_))

#numpy versioniigg hevleh

import numpy

print('numpy: {}'.format(numpy.\_\_version\_\_))

#matplotlib versioniigg hevleh

import matplotlib

print('matplotlib: {}'.format(matplotlib.\_\_version\_\_))

#pandas versioniigg hevleh

import pandas

print('pandas: {}'.format(pandas.\_\_version\_\_))

#scikit-learn versioniigg hevleh

import sklearn

print('sklearn: {}'.format(sklearn.\_\_version\_\_))

# Load libraries

import pandas

from pandas.plotting import scatter\_matrix

import matplotlib.pyplot as plt

from sklearn import model\_selection

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

from sklearn.naive\_bayes import GaussianNB

from sklearn.svm import SVC

# Load dataset

url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"

names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']

dataset = pandas.read\_csv(url, names=names)

# хэдэн мөр хэдэн баганатай мэдээлэл байгааг харуулна

print(dataset.shape)

#тухайн өгөгдлөөс эхний хэдэн мөрийн мэдээллийг харуулахыг заана

print(dataset.head(20))

#багана тус бүрт хэдэн мөрийн мэдээлэл байгаа мөн max, min зэрэг мэдээлийг гаргадаг

print(dataset.describe())

# ангилсан анги бүрт хэдэн мэдээлэл байгаа хэмжээ

print(dataset.groupby('class').size())

# өгөгдлийг дүрслэх хэлбэрийг тодорхойлох

dataset.plot(kind='bar', subplots=True, layout=(2,2), sharex=False, sharey=False)

plt.show()

# тархалтыг багана тус бүрт графикаар дүрслэх

dataset.hist()

plt.show()

# scatter plot matrix

scatter\_matrix(dataset)

plt.show()

array = dataset.values

X = array[:,0:4]

Y = array[:,4]

validation\_size = 0.20

seed = 7

X\_train, X\_validation, Y\_train, Y\_validation = model\_selection.train\_test\_split(X, Y, test\_size=validation\_size, random\_state=seed)

# Test options and evaluation metric

seed = 7

scoring = 'accuracy'

1

2

3

# Test options and evaluation metric

seed = 7

scoring = 'accuracy'

# Spot Check Algorithms

models = []

models.append(('LR', LogisticRegression(solver='liblinear', multi\_class='ovr')))

models.append(('LDA', LinearDiscriminantAnalysis()))

models.append(('KNN', KNeighborsClassifier()))

models.append(('CART', DecisionTreeClassifier()))

models.append(('NB', GaussianNB()))

models.append(('SVM', SVC(gamma='auto')))

# evaluate each model in turn

results = []

names = []

for name, model in models:

kfold = model\_selection.KFold(n\_splits=10, random\_state=seed)

cv\_results = model\_selection.cross\_val\_score(model, X\_train, Y\_train, cv=kfold, scoring=scoring)

results.append(cv\_results)

names.append(name)

msg = "%s: %f (%f)" % (name, cv\_results.mean(), cv\_results.std())

print(msg)

# Compare Algorithms

fig = plt.figure()

fig.suptitle('Algorithm Comparison')

ax = fig.add\_subplot(111)

plt.boxplot(results)

ax.set\_xticklabels(names)

plt.show()

# Make predictions on validation dataset

knn = KNeighborsClassifier()

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

predictions = knn.predict(X\_validation)

print(accuracy\_score(Y\_validation, predictions))

print(confusion\_matrix(Y\_validation, predictions))

print(classification\_report(Y\_validation, predictions))