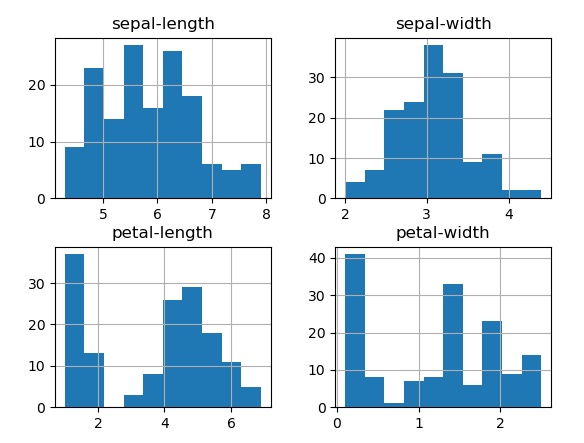
**Assignment -1 CS-805 Course Machine Learning Name: Robert Akinie**

**\*Answers in Bold**

1. **Assingment\_1a:**
2. Copy and Paste the snapshots of the histogram plots?



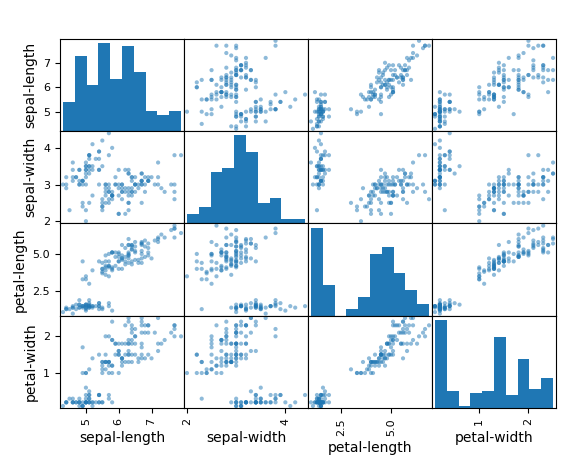
1. What is the total size of the whole dataset in terms of [Matrix dataset] = X\*Y?

**[150, 5]**

1. What is the command to print, if you want to see the first 10 rows of the dataset?

**print(dataset.head(10))**

1. Copy and paste the snapshot of the scatter matrix?



1. What is the Mean Value of Sepal Width Column of the Dataset?

**3.054 (4 s.f)**

1. What is the Max Value of the Petal-width Column in the Dataset?

**2.500(4 s.f)**

1. Out of the 6 different algorithms mentioned in the project, which one has higher accuracy?

**Support Vector Machine**

1. How many classes are present in the Dataset?

**Three**

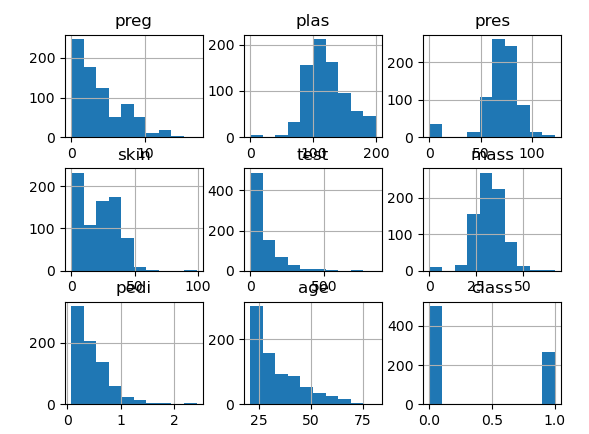
1. What is the difference you observe between univariate and bi-variate plots?

**Univariate plots show data of one variable at a time, whilst bivariate displays the comparison between two variables in question**

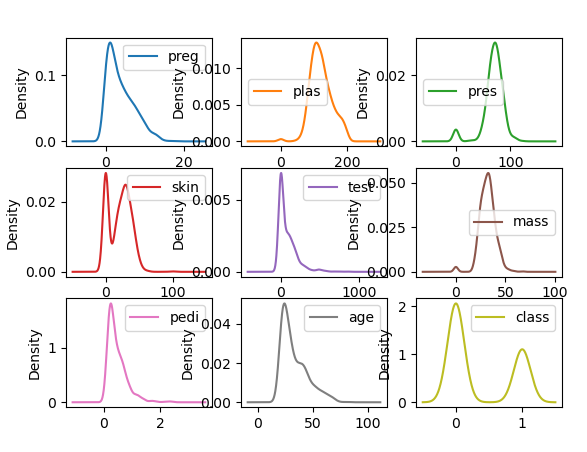
1. How is the training and Test dataset distributed in this Project?

**80% of dataset was allocated to training dataset, and remaining 20% is to test dataset**

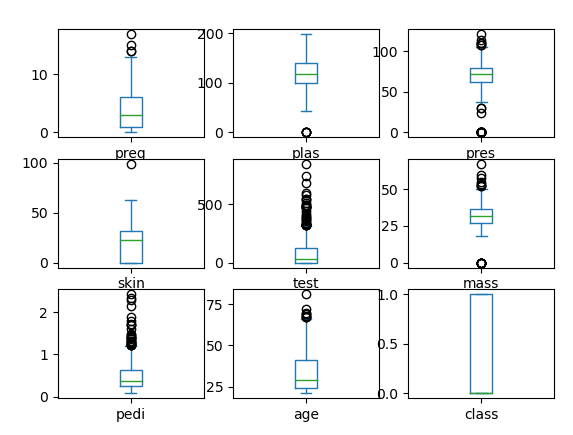
1. Visualize the data using the steps mentioned in the below link and paste the plots here. :--
2. Histogram plots



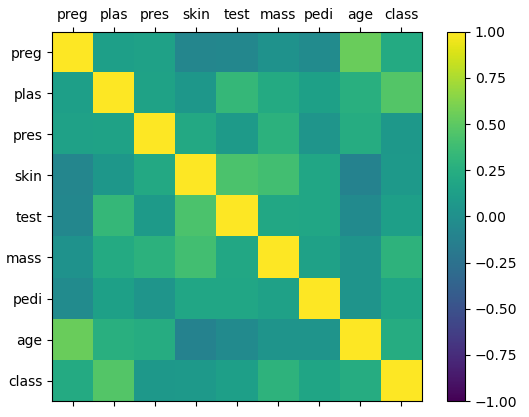
1. Density plots



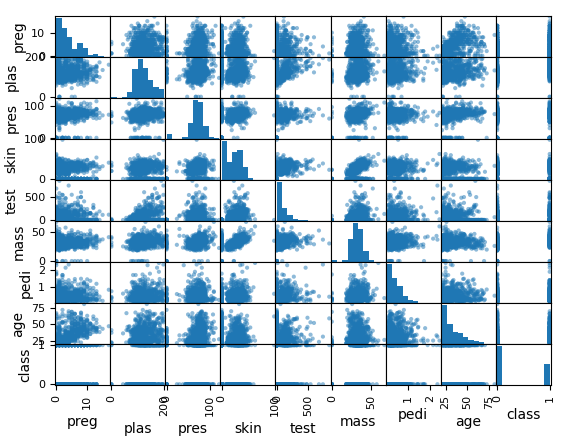
1. Box and Whisker Plots



1. Correlation matrix plot



1. Scatter plot matrix



1. Read through this link, and produce the

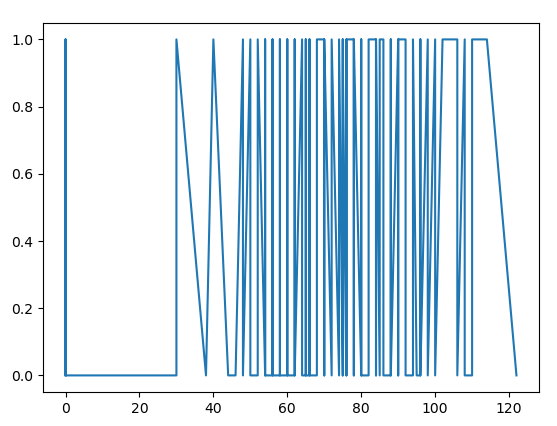
<https://machinelearningmastery.com/data-visualization-methods-in-python/>

1. Line Plots for Variable “ Blood Pressure” Dataset, Hint:-- Sort the Column for better plot

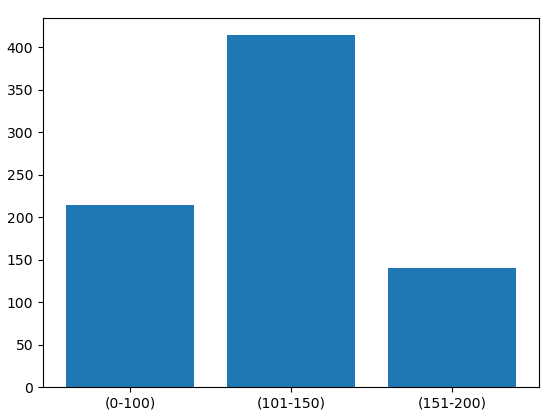
**\* '''comment: I was not sure what axis to plot the variable, and I could not quite make sense**

**or interpretation of the line plot so I plotted both axes, then one with default X-points. I placed one graph here because of space, but actual plots can be seen by executing attached code.**

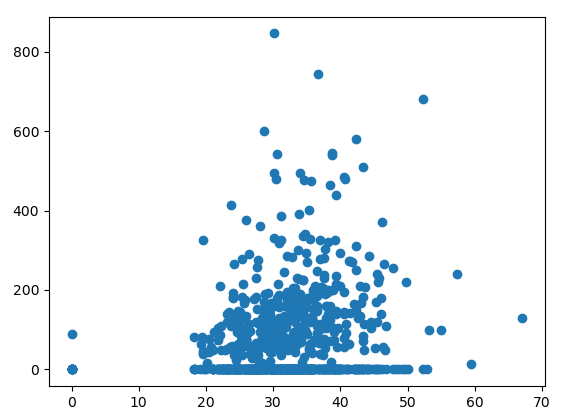
**'''**



1. Bar Plots for Variable “Glucose”, in three categorric values 0-100, 101-150, 151-200



1. Scatter plots of :two sets of variables : BMI Vs Insulin.



1. How many classes are present in the Dataset ?

**Two**

1. What is the command to change the ratio of training and Test dataset distributed in this Project ?

**train\_test\_split(arrays, \*test\_size, ...)**

**\*test\_size represent the ratio of the dataset distributed in training and test dataset**

1. What are the accuracies for ML model K-Nearest Neighbors and Support Vector Machines ?

**71.48(4 s.f) and 65.31(4 s.f) respectively**

1. What is the difference between Density Plots and Histograms ?

**Density plots are better at showing distribution of data, by smoothing out noise. Histograms have bins whose number can be altered. Histograms on the other hand, show in more detail which parts of the data have more weight.**

1. Write three advantages of Box-Whisker Plots and Correlation Matrix plot individually?

**Box-whisker plot**

1. Easy identification of data spread and location
2. It shows data outliers
3. It provides a summary of data, particularly key statistics.

**Correlation Matrix**

1. It shows the level of correlation between various features
2. It shows direction of correlation
3. Estimating weight of a dependent variable

**Appendix**

Code for 1-10

# Load libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from pandas import read\_csv

from pandas.plotting import scatter\_matrix

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

from sklearn.model\_selection import cross\_val\_score

from sklearn.model\_selection import StratifiedKFold

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 = read\_csv(url, names=names)

'''Statistical Summary'''

#shape

print(dataset.shape)

# head

print(dataset.head(20))

# descriptions

print(dataset.describe())

# class distribution

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

'''Data Visualization'''

# box and whisker plots

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

pyplot.show()

# histograms

dataset.hist()

pyplot.show()

# scatter plot matrix

scatter\_matrix(dataset)

pyplot.show()

'''Evaluation of Algorithms and Predictions'''

# Split-out validation dataset

array = dataset.values

X = array[:,0:4]

y = array[:,4]

X\_train, X\_validation, Y\_train, Y\_validation = train\_test\_split(X, y, test\_size=0.20, random\_state=1, shuffle=True)

# 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 = StratifiedKFold(n\_splits=10, random\_state=1, shuffle=True)

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

results.append(cv\_results)

names.append(name)

print('%s: %f (%f)' % (name, cv\_results.mean(), cv\_results.std()))

# Compare Algorithms

plt.boxplot(results, labels=names)

plt.title('Algorithm Comparison')

plt.show()

# Make predictions on validation dataset

modelsvm = SVC(gamma='auto')

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

predictions = modelsvm.predict(X\_validation)

# Evaluate predictions

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

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

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

Code for 11-20

# Load libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from pandas import read\_csv

from pandas.plotting import scatter\_matrix

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

from sklearn.model\_selection import cross\_val\_score

from sklearn.model\_selection import StratifiedKFold

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/pima-indians-diabetes.csv"

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']

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

'''Evaluation of Algorithms and Prediction'''

# Split-out validation dataset

array = dataset.values

X = array[:,0:8]

y = array[:,8]

X\_train, X\_validation, Y\_train, Y\_validation = train\_test\_split(X, y, test\_size=0.20, random\_state=1, shuffle=True)

#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 algorithm in turn

results=[]

names2=[]

for name, model in models:

kfold = StratifiedKFold(n\_splits=10, random\_state=1, shuffle=True)

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

results.append(cv\_results)

names2.append(name)

print('%s: %f (%f)' % (name, cv\_results.mean(), cv\_results.std()))

# Compare Algorithms

plt.boxplot(results, labels=names2)

plt.title('Algorithm Comparison')

plt.show()

'''Visualizing the data'''

#Histogram plots

dataset.hist()

plt.show()

#Density plots

dataset.plot(kind='density', subplots=True, layout=(3,3), sharex=False)

plt.show()

#Box and Whisker plots

dataset.plot(kind='box', subplots=True, layout=(3,3), sharex=False)

plt.show()

#Correlation Matrix plot

correlations = dataset.corr()

fig = plt.figure()

ax = fig.add\_subplot(111)

cax = ax.matshow(correlations, vmin=-1, vmax=1)

fig.colorbar(cax)

ticks = np.arange(0,9,1)

ax.set\_xticks(ticks)

ax.set\_yticks(ticks)

ax.set\_xticklabels(names)

ax.set\_yticklabels(names)

plt.show()

#Scatterplot Matrix

scatter\_matrix(dataset)

plt.show()

#Line Plot

'''comment: I was not sure what axis to plot the variable, and I could not quite make sense

or interpretation of the line plot so I plotted both axes, then one with default X-points

'''

bloodpres = np.array(dataset.pres.to\_numpy())

plt.plot(np.sort(bloodpres), y)

plt.show()

plt.plot(y, np.sort(bloodpres))

plt.show()

plt.plot(np.sort(bloodpres))

plt.show()

#Bar plot

glucse = dataset.plas.to\_numpy()

x\_gluc = ["(0-100)" , "(101-150)" , "(151-200)"]

y\_gluc = [np.count\_nonzero(glucse<=100), np.count\_nonzero((100<glucse) & (glucse<=150)), np.count\_nonzero(glucse>150)]

plt.bar(x\_gluc,y\_gluc)

plt.show()

#Scatter plot

BMI = dataset.mass

Insulin = dataset.test

plt.scatter(BMI, Insulin)

plt.show()