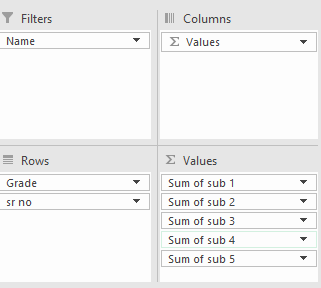
**PRACTICAL-1**

**Aim:- Use MS-Excel to create pivot table & apply statistical measures to it..**

**Data:**



**Pivot table:**



**PRACTICAL-2**

**AIM:** **Use the Table Created in Above Practical to Generate Different Charts.**

**Data:**



**Output:**

**PRACTICAL-4**

**AIM:** **Use python libraries to generate chart from data stored in Excel.**

**Code:**

import pandas as pd

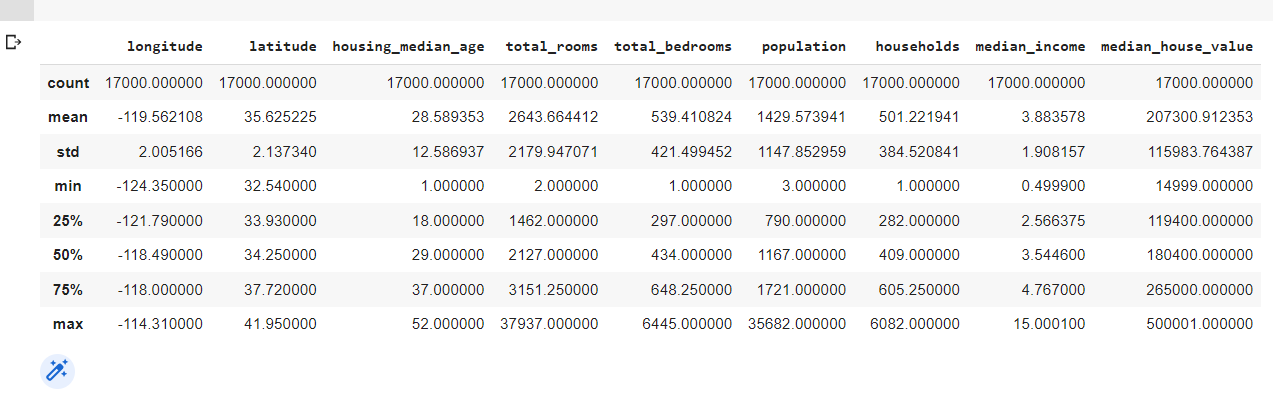
from matplotlib import pyplot as plt

raw\_data="/content/sample\_data/california\_housing\_train.csv"

df = pd.read\_csv(raw\_data)

df.describe()

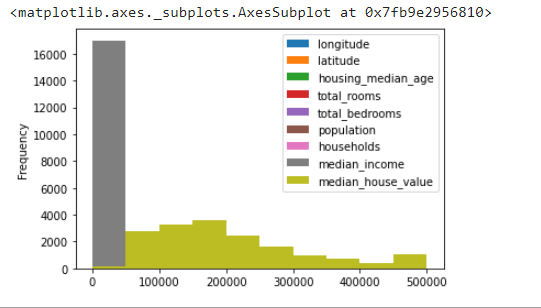
**Output:**

****

**Code:**

df.plot.hist()

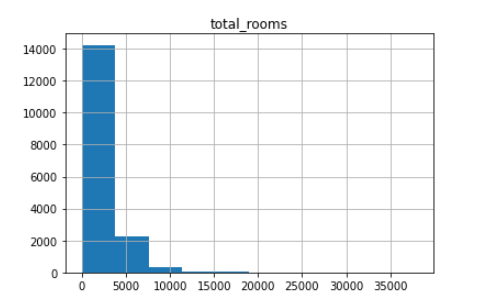
**Output:**

****

**Code:**

df.hist(column='total\_rooms');

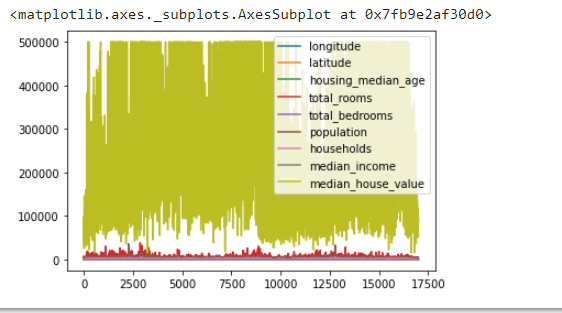
**Output:**

****

**Code:**

df.plot.line()

**Output:**

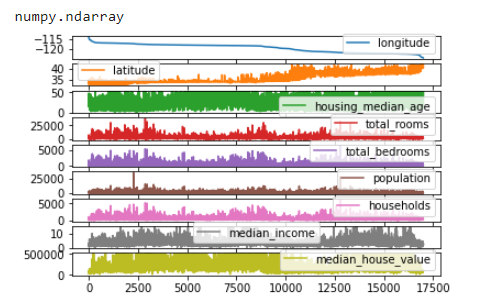
****

**Code:**

axes = df.plot.line(subplots=True)

type(axes)

**Output:**

****

**PRACTICAL-5**

**AIM:** **Perform Multiple Linear Regression on data.**

**Code:**

# Import Libary

import numpy as np

import pandas as pd

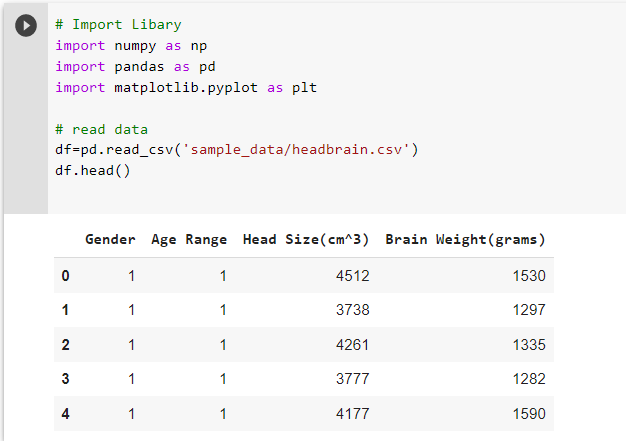
import matplotlib.pyplot as plt

# read data

df=pd.read\_csv('sample\_data/headbrain.csv')

df.head()

**Output:**

****

**Code:**

# Declare dependent variable(Y) and independent variable(X)

X=df['Head Size(cm^3)'].values

Y = df['Brain Weight(grams)'].values

np.corrcoef(X, Y)

**Output:**

array([[1. , 0.79956971], [0.79956971, 1. ]])

**Code:**

# Plot the Input Data

plt.scatter(X, Y, c='green', label='Data points')

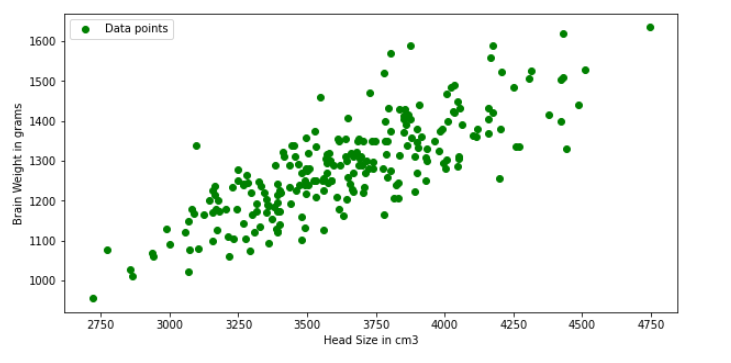
plt.xlabel('Head Size in cm3')

plt.ylabel('Brain Weight in grams')

plt.legend()

plt.show()

**Output:**



**Code:**

# Mean X and Y

mean\_x = np.mean(X)

mean\_y = np.mean(Y)

# Total number of values

n = len(X)

# Using the formula to calculate theta1 and theta2

numer = 0

denom = 0

for i in range(n):

    numer += (X[i] - mean\_x) \* (Y[i] - mean\_y)

    denom += (X[i] - mean\_x) \*\* 2

b1 = numer / denom

b0 = mean\_y - (b1 \* mean\_x)

# Printing coefficients

print("coefficients for regression",b1, b0)

**Output:**

coefficients for regression 0.26342933948939945 325.57342104944223

**Code:**

# Plotting Values and Regression Line

%matplotlib inline

plt.rcParams['figure.figsize'] = (10.0, 5.0)

max\_x = np.max(X) + 100

min\_x = np.min(X) - 100

y = b0 + b1 \* X

# Ploting Line

plt.plot(X, y, color='blue', label='Regression Line')

# Ploting Scatter Points

plt.scatter(X, Y, c='green', label='Scatter data')

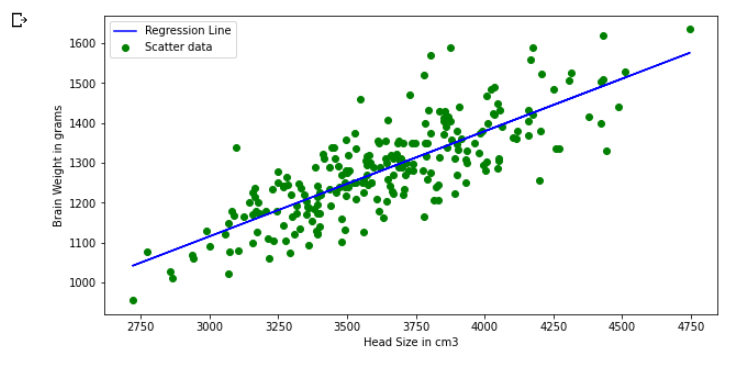
plt.xlabel('Head Size in cm3')

plt.ylabel('Brain Weight in grams')

plt.legend()

plt.show()

**Output:**



**PRACTICAL-5**

**AIM:** **Perform the Logistic Regression on a dataset and Interpret the regression table.**

**Output:**   
 

**PRACTICAL-8**

**AIM:** **Use a dataset & apply K means clustering to get insights from data.**

**Code:**

from sklearn.cluster

import KMeans import pandas as pd

from sklearn.preprocessing

import MinMaxScaler from matplotlib

import pyplot as plt

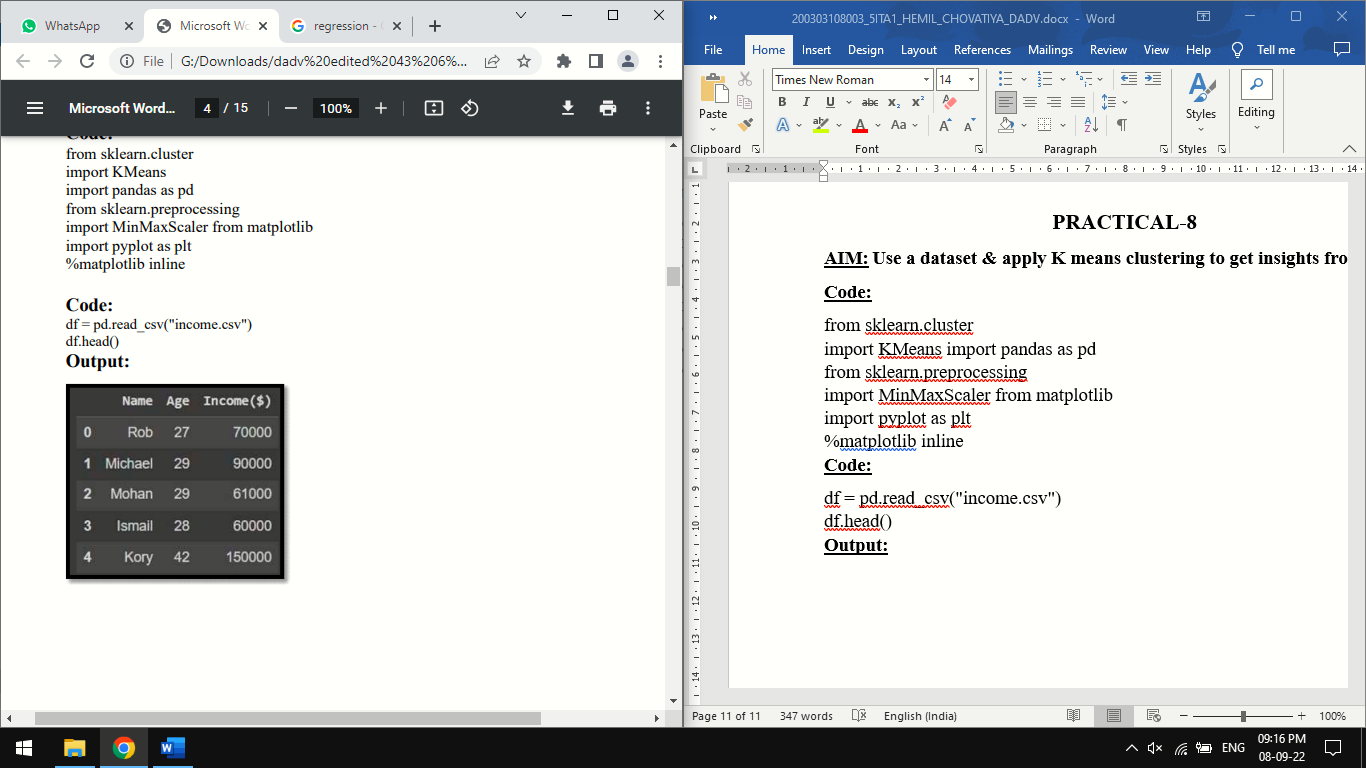
%matplotlib inline

**Code:**

df = pd.read\_csv("income.csv")

df.head()

**Output:**



**Code:**

km.cluster\_centers\_

**Output:**

array([[3.40000000e+01, 8.05000000e+04],[3.82857143e+01, 1.50000000e+05], [3.29090909e+01, 5.61363636e+04]])

**Code:**

df1 = df[df.cluster==0]

df2 = df[df.cluster==1]

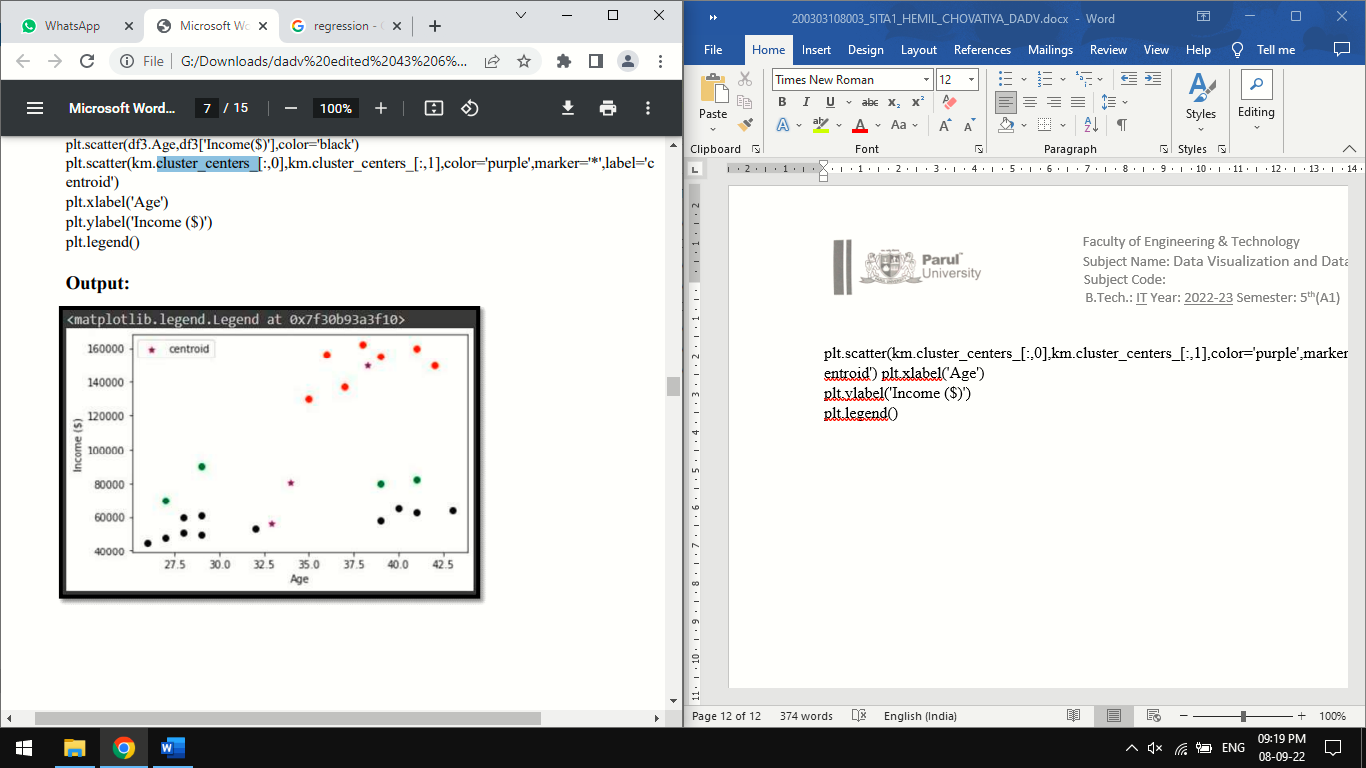
df3 = df[df.cluster==2]

plt.scatter(df1.Age,df1['Income($)'],color='green') plt.scatter(df2.Age,df2['Income($)'],color='red') plt.scatter(df3.Age,df3['Income($)'],color='black') plt.scatter(km.cluster\_centers\_[:,0],km.cluster\_centers\_[:,1],color='purple',marker='\*',label='c entroid') plt.xlabel('Age')

plt.ylabel('Income ($)')

plt.legend()

**Output:**



scaler = MinMaxScaler()

scaler.fit(df[['Income($)']])

df['Income($)'] = scaler.transform(df[['Income($)']])

scaler.fit(df[['Age']]) df['Age'] = scaler.transform(df[['Age']]) plt.scatter(df.Age,df['Income($)'])

**Output:**

array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2],

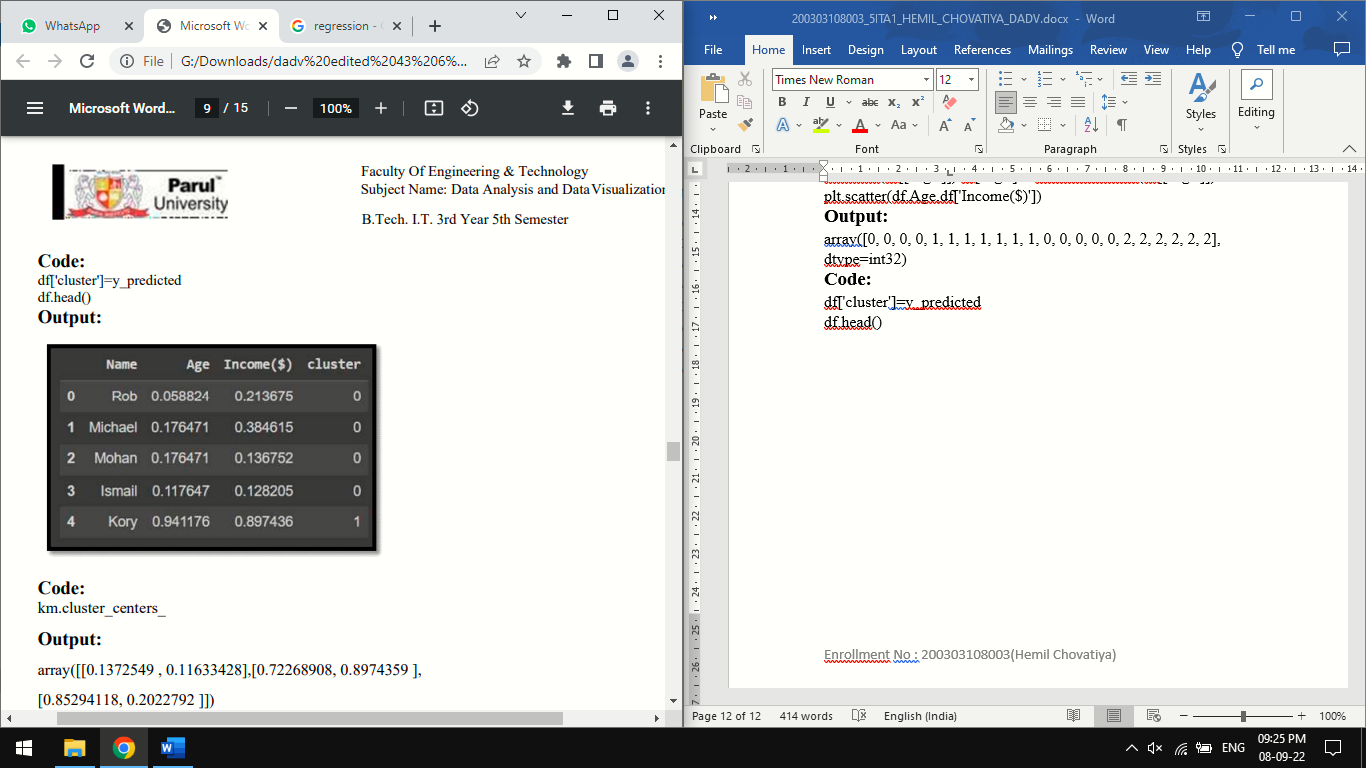
dtype=int32)

**Code:**

df['cluster']=y\_predicted

df.head()

**Output:**



**Code:**

km.cluster\_centers\_

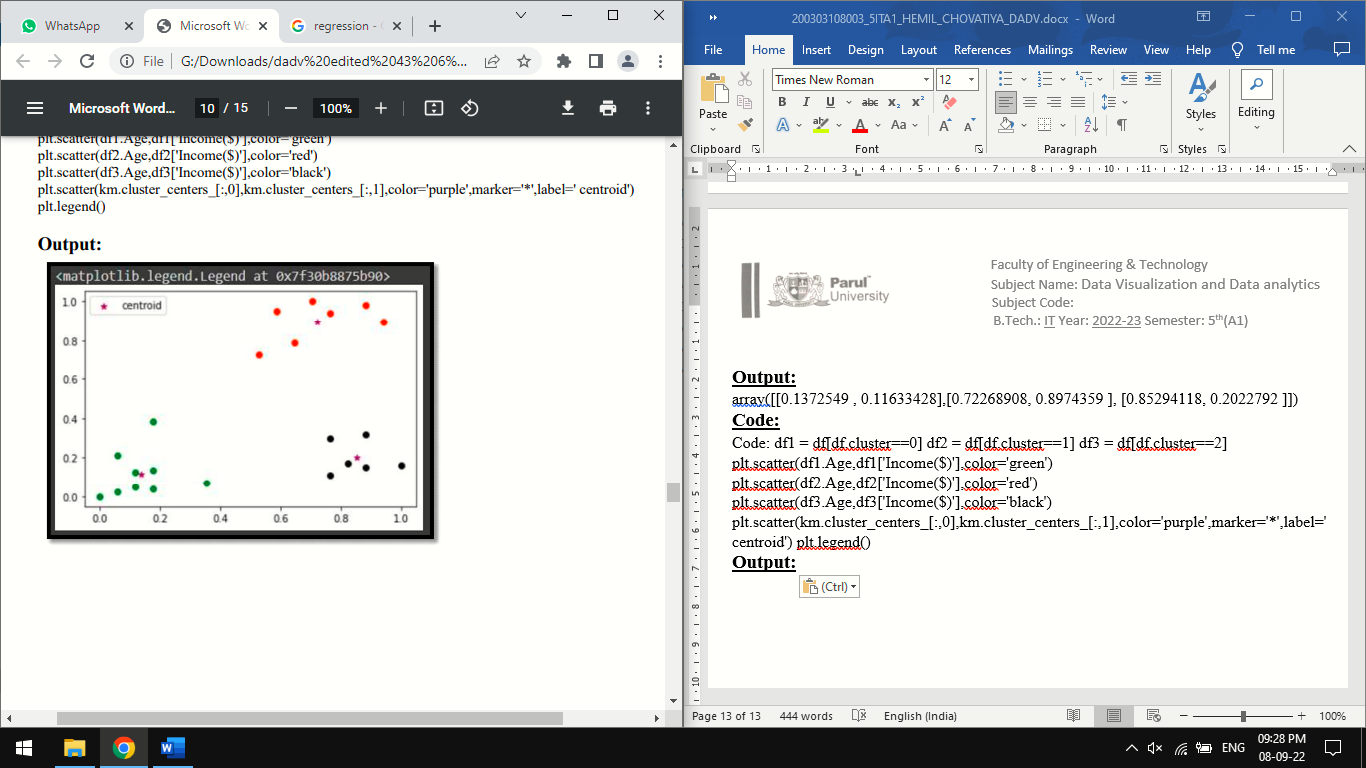
**Output:**

array([[0.1372549 , 0.11633428],[0.72268908, 0.8974359 ], [0.85294118, 0.2022792 ]])

**Code:**

Code: df1 = df[df.cluster==0] df2 = df[df.cluster==1] df3 = df[df.cluster==2] plt.scatter(df1.Age,df1['Income($)'],color='green') plt.scatter(df2.Age,df2['Income($)'],color='red') plt.scatter(df3.Age,df3['Income($)'],color='black') plt.scatter(km.cluster\_centers\_[:,0],km.cluster\_centers\_[:,1],color='purple',marker='\*',label=' centroid') plt.legend()

**Output:**



**PRACTICAL-9**

**AIM:** **Study about the tools like Orange, Tableau, Weka etc. tool for data Visualization.**

**Theory:**

Orange Tool: Orange supports a flexible domain for developers, analysts, and data mining specialists. Python, a new generation scripting language and programming environment, where our data mining scripts may be easy but powerful. Orange employs a component-based approach for fast prototyping. We can implement our analysis technique simply like putting the LEGO bricks, or even utilize an existing algorithm. What are Orange components for scripting Orange widgets for visual programming?. Widgets utilize a specially designed communication mechanism for passing objects like classifiers, regressors, attribute lists, and data sets permitting to build easily rather complex data mining schemes that use modern approaches and techniques.

Orange core objects and Python modules incorporate numerous data mining tasks that are far from data preprocessing for evaluation and modeling. The operating principle of Orange is cover techniques and perspective in data mining and machine learning. For example, Orange's top-down induction of decision tree is a technique build of numerous components of which anyone can be prototyped in python and used in place of the original one. Orange widgets are not simply graphical objects that give a graphical interface for a specific strategy in Orange, but it includes an adaptable signaling mechanism that is for communication and exchange of objects like data sets, classification models, learners, objects that store the results of the assessment. All these ideas are significant and together recognize Orange from other data mining structures.

**Widgets:**

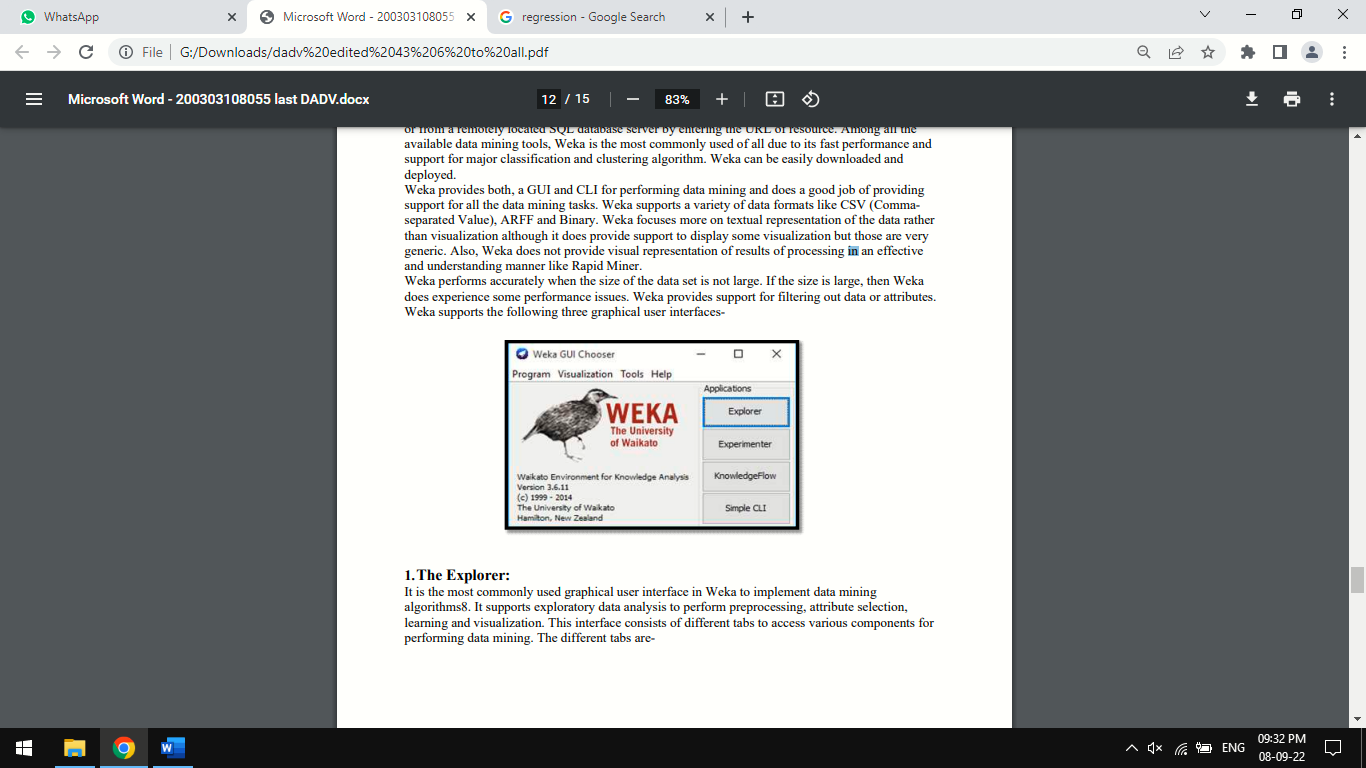
Orange widgets give us a graphical user interface to orange's data mining and machine learning techniques. They incorporate widgets for data entry and pre processing, classification, regression, association rules and clustering a set of widgets for model assessment and visualization of assessment results, and widgets for exporting the models into PMML.

**Scripting:**

If we want to access Orange objects, then we need to write our components and design our test schemes and machine learning applications through the script. Orange interfaces to Python, a model simple to use a scripting language with clear and powerful syntax and a broad set of additional libraries. Same as any scripting language, Python can be used to test a few ideas mutually or to develop more detailed scripts and programs.

**Weka Tool:**

Weka is one of the very popular open source data mining tools developed at the University of Waikato in New Zealand in 1992. It is a Java based tool and can be used to implement various machine learning and data mining algorithms written in Java. The simplicity of using Weka has made it a landmark for machine learning and data mining implementation. Weka supports reading of files from several different databases and also allows importing the data from the internet, from web pages or from a remotely located SQL database server by entering the URL of resource. Among all the available data mining tools, Weka is the most commonly used of all due to its fast performance and support for major classification and clustering algorithm. Weka can be easily downloaded and deployed. Weka provides both, a GUI and CLI for performing data mining and does a good job of providing support for all the data mining tasks. Weka supports a variety of data formats like CSV (Commaseparated Value), ARFF and Binary. Weka focuses more on textual representation of the data rather than visualization although it does provide support to display some visualization but those are very generic. Also, Weka does not provide visual representation of results of processing in an effective and understanding manner like Rapid Miner. Weka performs accurately when the size of the data set is not large. If the size is large, then Weka does experience some performance issues. Weka provides support for filtering out data or attributes. Weka supports the following three graphical user interfaces



1.The Explorer:

It is the most commonly used graphical user interface in Weka to implement data mining algorithms8. It supports exploratory data analysis to perform preprocessing, attribute selection, learning and visualization. This interface consists of different tabs to access various components for performing data mining. The different tabs are-

A) **Preprocessing** Using this tab, we can load input data files and perform preprocessing on this data using filters.

B) **Classify** This tab is used to implement different classification and regression algorithms. We can do this by selecting a particular classifier from this tab. For example, the K-NN or Naïve Bayesian algorithm can be implemented by using this tab.

C) **Associate** This tab is used to find out all association rules between different attributes of the data and which can be used for further mining. For example, Association rule mining, etc. D) **Cluster** Using this tab, we can select a particular clustering algorithm to implement for our data set. Clustering algorithms like K-means can be implemented using this tab.

E) **Select attributes** This tab is used to select particular attributes from the data set useful for implementing the algorithm.

F) **Visualize** This tab is used to visualize the data whenever available or supported by a particular algorithm in the form of scatter plot matrix.

2. **The Experimenter**

This user interface provides experimental environment for testing and evaluating machine learning algorithms.

3. **The Knowledge**

Flow Knowledge flow is basically a component based interface similar to explorer. This interface is used for new process evaluations.

**Tableau Tool:**

Tableau is a powerful data visualization tool used in business intelligence and data analysis. Tableau Software was invented by Chris Stolte, Christian Chabot and Pat Hanrahan in January, 2003. The visualization provided by Tableau has completely enhanced the ability to gain more knowledge about the data we are working on and can be used to provide more accurate predictions. “The product queries relational databases, cubes, cloud databases, and spreadsheets and then generates a number of graph.Types that can be combined into dashboards which can be securely shared over a computer network or the internet”. Unlike Rapid Miner and Weka, Tableau does not implement data mining algorithms provides visualizations of the data. For this, Tableau provides integration with another popular statistical analysis tool R9, to provide support for data mining. “Tableau offers five main products namely Tableau Desktop, Tableau Server, Tableau Online,Tableau Reader and Tableau Public. Tableau Public and Tableau Reader are available freely, whereas Tableau Server and Tableau Desktop come with a free trial period afterwards which the user has to pay”. Tableau has made it possible to explore and present the data in a much simpler and beautiful manner. Working on projects using Tableau is less time consuming and easy to handle. Tableau uses a feature called Dashboard which is a collection of worksheets which can be easily imported from anywhere.

**PRACTICAL-10**

**AIM:** **Given a case study: Interactive Data Analytics with Power BI.**

**Theory:**

**The Client**:

The client is a data analytics specialist organization that uses artificial intelligence and data analytics to study professional services industry problems. The company is headquartered in London, England.

**Client's Requirement:**

The client was looking for a reliable and cost-effective service provider who could help them with the implementation of a Power BI solution for their business.

**The Challenges**:

The client had been facing several challenges before the implementation of the project. Some of the major challenges faced were:

1. The client wanted to create one dashboard that would give them all the information about their business that they needed in one, easy-to-access place.

2. They wanted to add some additional pages and visualizations to the report, as well as making it more concise, clear, and informative.

**The Solution**:

Flatworld Power BI development team meticulously studied the client's requirements and created the perfect reporting tool to accommodate all of them. To go above and beyond, Flatworld proposed multiple suggestions to improve the reports including shuffling reports around, adding more visual experiences, and other structural changes to make this dashboard both eye-catching and functional. Flatworld team worked with Power BI to enhance reports. The relationships in tables were made interactive, charts and graphs were redesigned, and the entire UI/UX was improved for an enhanced user experience.

**The Results:**

We estimated it would take 10 days to complete the first milestone of the project, but it ended up taking us only 7. The client was happy with the delivery and offered more Power BI work, all of which we will happily fulfill. The client was not only delighted to see the quality of the dashboard we created but also the cost-effectiveness of the entire project.