

Date:02.02.2022

Third Year B. Tech., Sem VI 2021-22

Software Engineering Tools Lab

Assignment Submission

PRN No: 2019BTECS00064

Full name: Kunal Santosh Kadam

Batch: T2

Assignment: 1

Title of assignment: Introduction to OSS

1. Weka is a GUI workbench that empowers data wranglers to assemble machine learning pipelines, train models, and run predictions without having to write code. Using Weka tool perform below tasks such as data pre-processing, data classification (use any appropriate ML algorithm) and data visualization efficiently on given dataset.

Use the Iris dataset given –

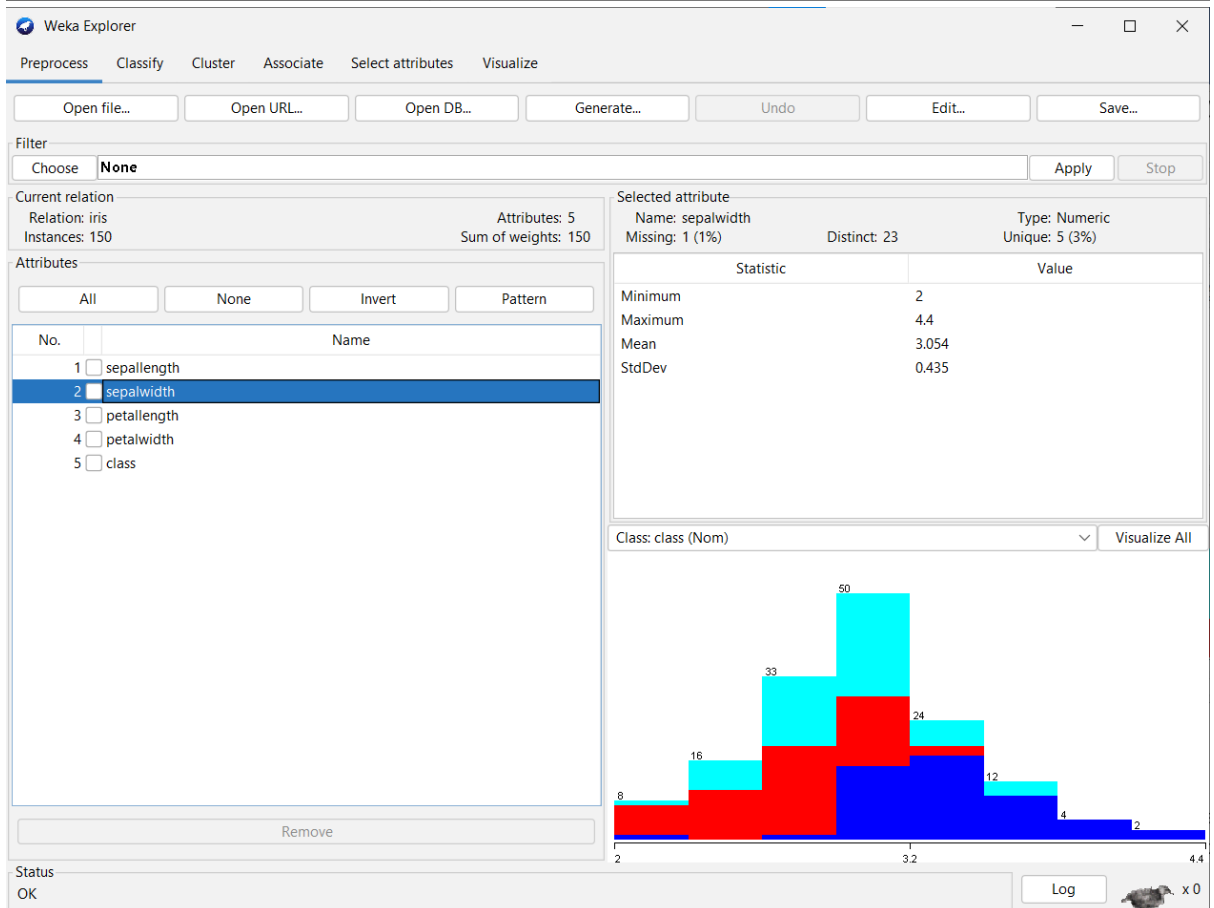
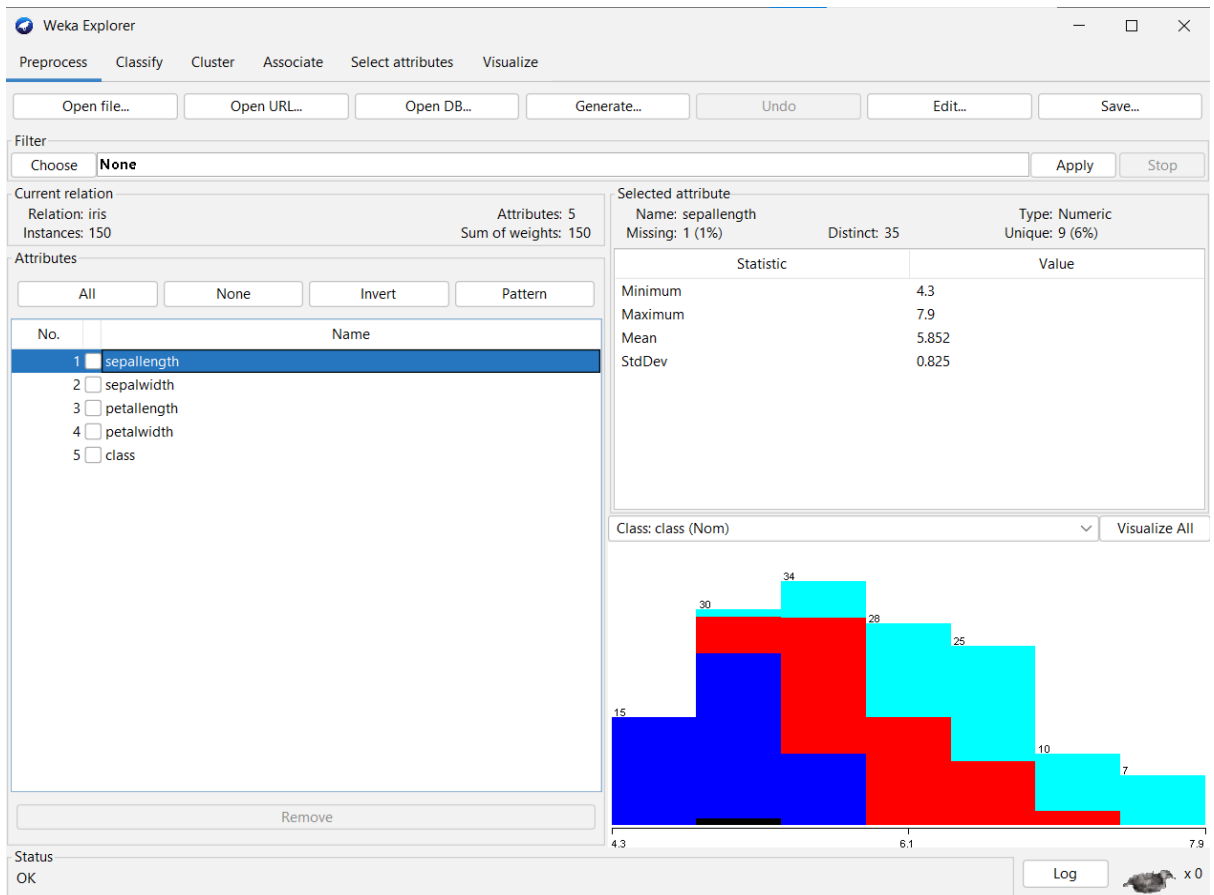
<https://drive.google.com/file/d/1A3Fxsfzm6BSfhFZGDrjI47RTe45bSgYP/view>

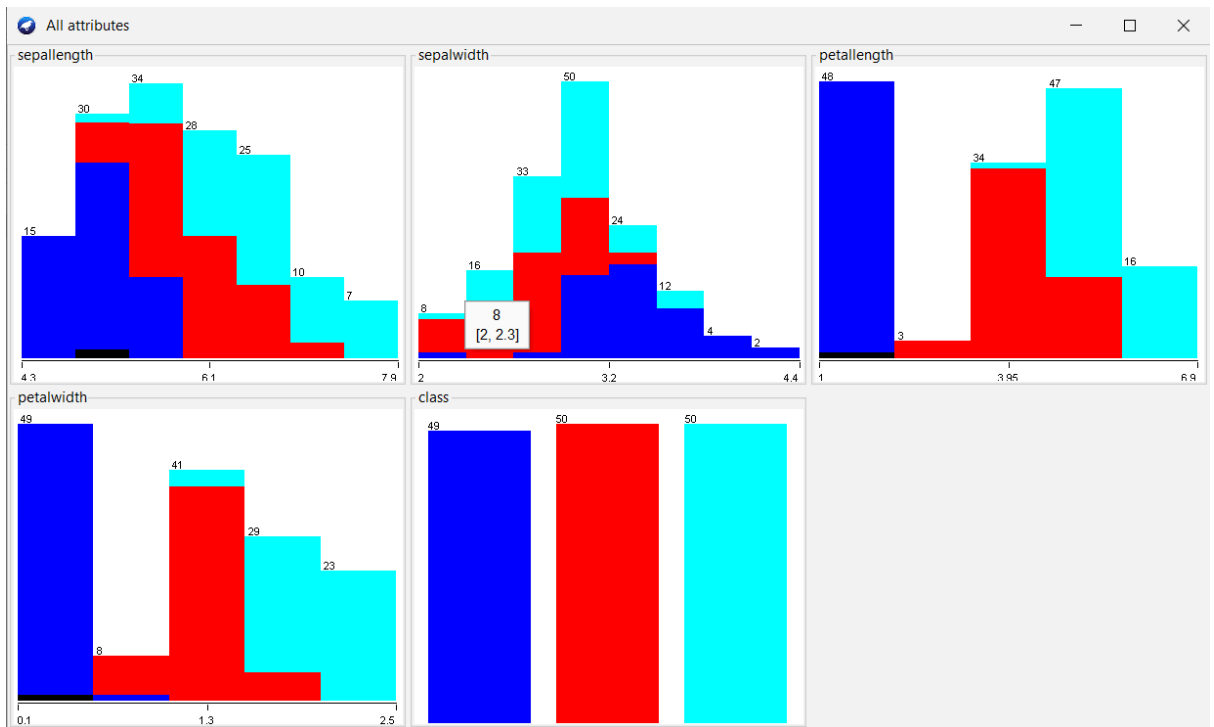
Note - provide screen shots for every task

Create a report which will illustrate the details of tasks performed (for e.g., to perform pre-processing of data provide details of navigation and selection of appropriate parameters)

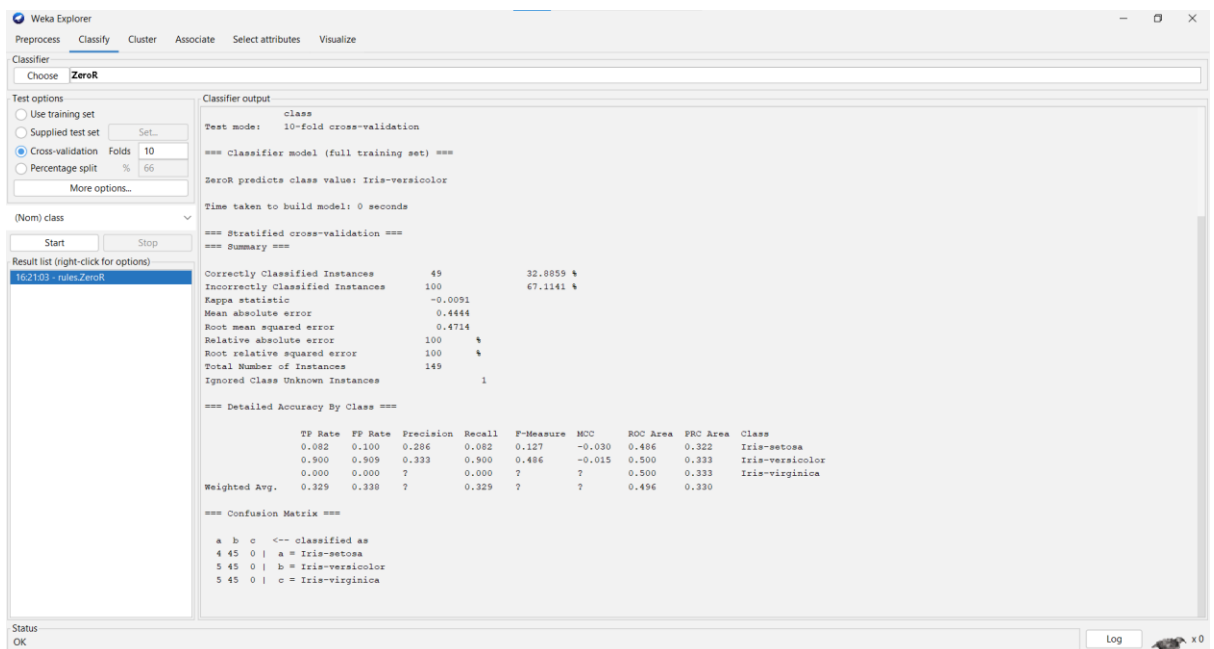
Ans:

Data Pre-processing





Data Classification:



Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier: Choose **Logistic** R 1.0E-8 -M -1 -num-decimal-places 4

Test options:
☐ Use training set
☐ Supplied test set Set...
☒ Cross-validation Folds **10**
☐ Percentage split % **66**
 More options...

(Nom) class
 Start Stop

Result list (right-click for options):
 162103 - rules.ZeroR
162505 - functions.Logistic

Classifier output:

```
=====
sepalength      0.0004      11.7662
sepalwidth      1.6342968366014786E10      797.0118
petallength      0      0.0001
petalwidth      0      0

Time taken to build model: 0.07 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      145      97.3154 %
Incorrectly Classified Instances      4      2.6846 %
Kappa statistic      0.9597
Mean absolute error      0.0263
Root mean squared error      0.1395
Relative absolute error      5.9069 %
Root relative squared error      29.5849 %
Total Number of Instances      149
Ignored Class Unknown Instances      1

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC  ROC Area  PRC Area  Class
      -----  -
0.980   0.000   1.000   0.980   0.990   0.985   0.979   0.964   Iris-setosa
0.960   0.020   0.960   0.960   0.960   0.940   0.964   0.933   Iris-versicolor
0.980   0.020   0.961   0.980   0.970   0.955   0.974   0.937   Iris-virginica
Weighted Avg.   0.973   0.014   0.973   0.973   0.973   0.960   0.972   0.931

=== Confusion Matrix ===

  a  b  c  <-- classified as
48  1  0 | a = Iris-setosa
  0 48  2 | b = Iris-versicolor
  0  1 49 | c = Iris-virginica
```

Status: OK

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier: Choose **SimpleLogistic** -I 0 -M 500 -H 50 -W 0.0

Test options:
☐ Use training set
☐ Supplied test set Set...
☒ Cross-validation Folds **10**
☐ Percentage split % **66**
 More options...

(Nom) class
 Start Stop

Result list (right-click for options):
 162103 - rules.ZeroR
 162505 - functions.Logistic
162531 - functions.SimpleLogistic

Classifier output:

```
=====
-34.62 +
[sepalength] * -0.38 +
[sepalwidth] * -3.89 +
[petallength] * 6.28 +
[petalwidth] * 10.82

Time taken to build model: 0.35 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      145      97.3154 %
Incorrectly Classified Instances      4      2.6846 %
Kappa statistic      0.9597
Mean absolute error      0.0332
Root mean squared error      0.1281
Relative absolute error      7.4706 %
Root relative squared error      27.1694 %
Total Number of Instances      149
Ignored Class Unknown Instances      1

=== Detailed Accuracy By Class ===

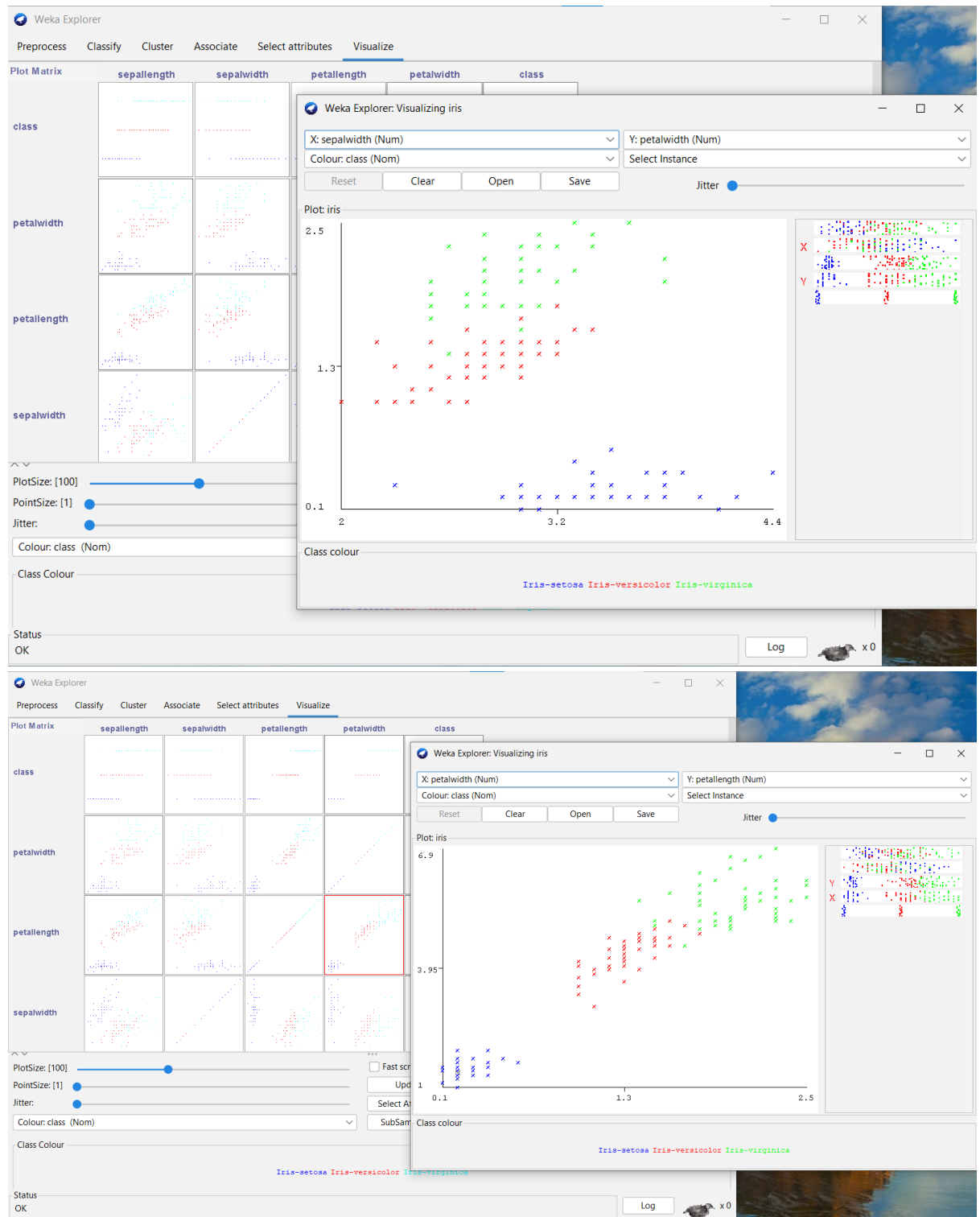
      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC  ROC Area  PRC Area  Class
      -----  -
1.000   0.000   1.000   1.000   1.000   1.000   0.993   0.975   Iris-setosa
0.940   0.010   0.979   0.940   0.959   0.940   0.992   0.986   Iris-versicolor
0.980   0.030   0.942   0.980   0.961   0.941   0.993   0.987   Iris-virginica
Weighted Avg.   0.973   0.014   0.974   0.973   0.973   0.960   0.993   0.982

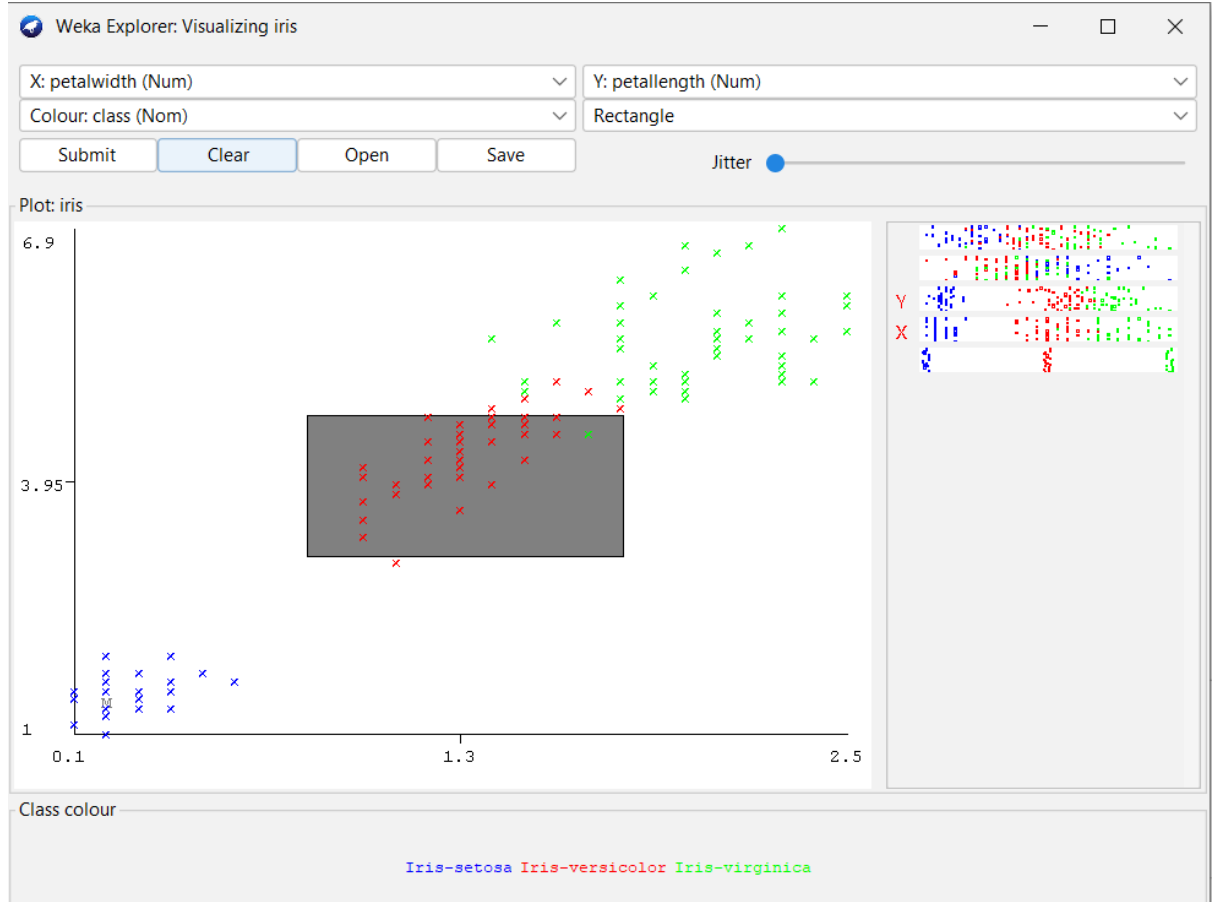
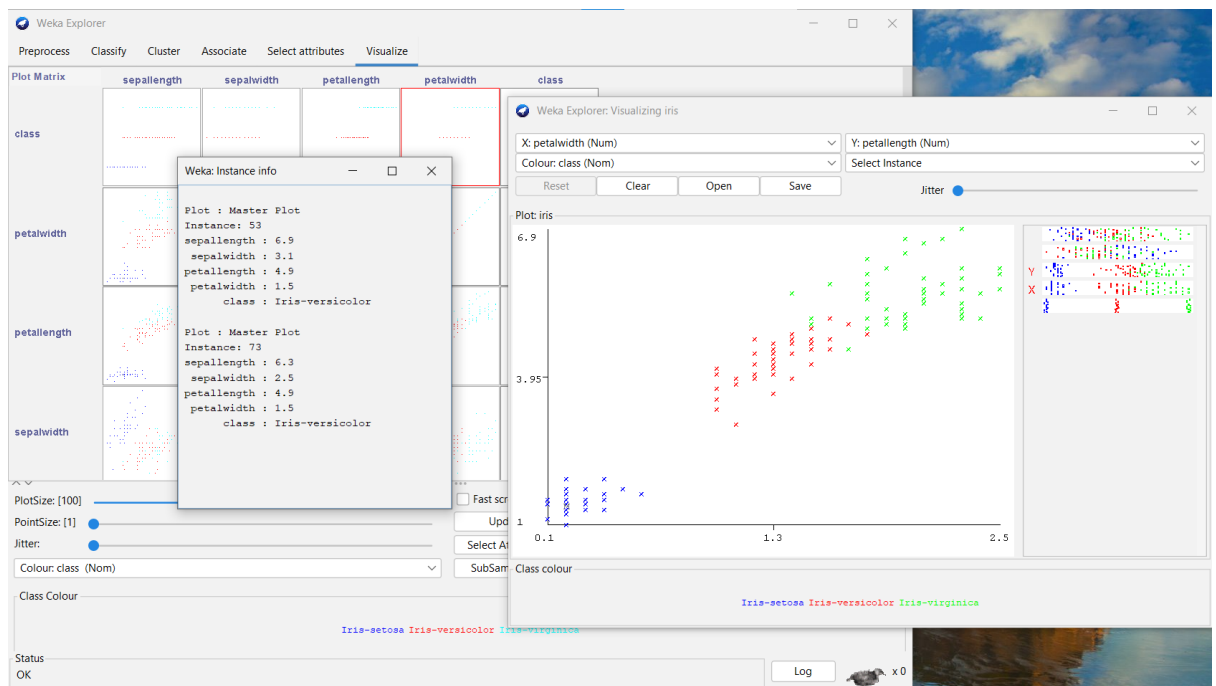
=== Confusion Matrix ===

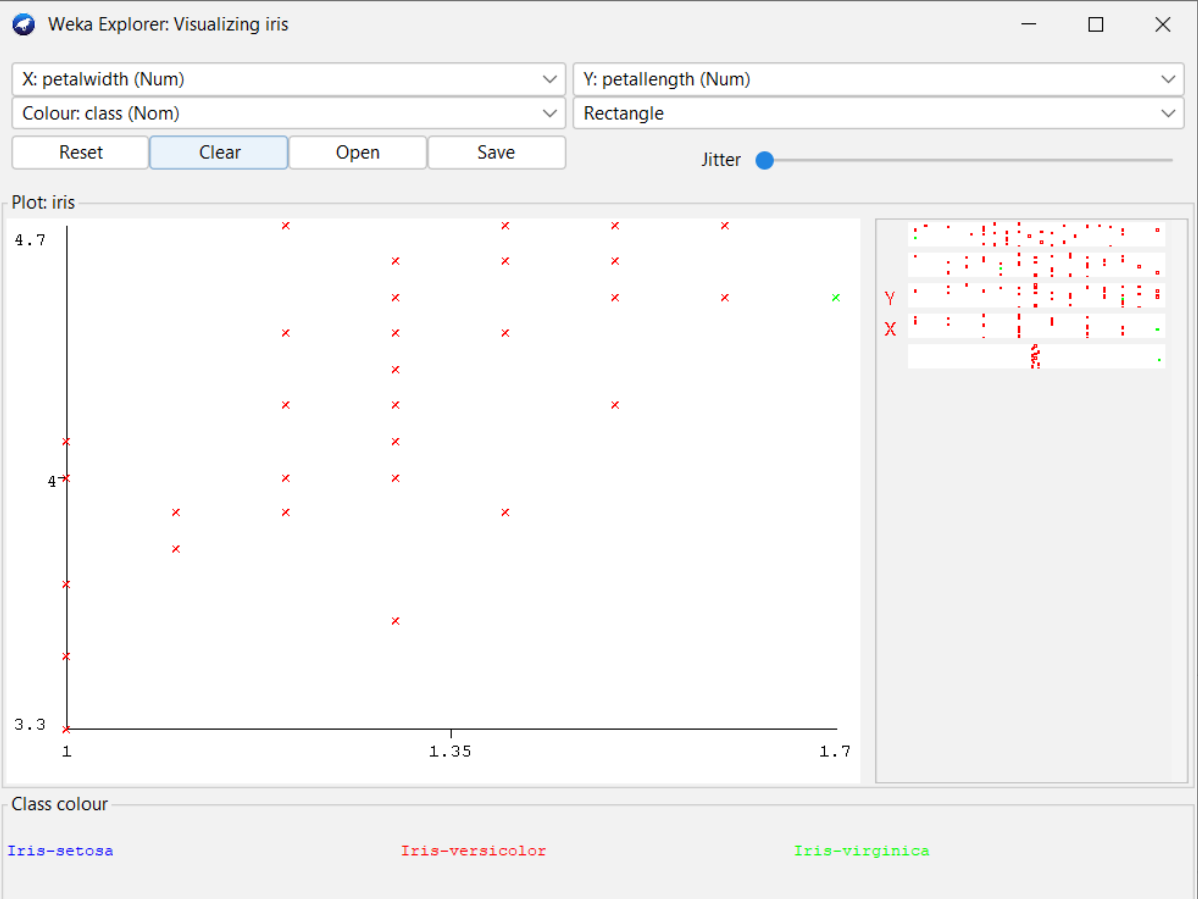
  a  b  c  <-- classified as
49  0  0 | a = Iris-setosa
  0 47  3 | b = Iris-versicolor
  0  1 49 | c = Iris-virginica
```

Status: OK

Data Visualization







2. Orange is an easy-to-use data visualization tool with a large toolkit. In spite of being a GUI-based beginner-friendly tool, you mustn't mistake it for a light-weight one. It can do statistical distributions and box plots as well as decision trees, hierarchical clustering and linear projections.
 - a. Install orange
 - b. Show data distribution
 - c. Show linear projection
 - d. Show Free Viz

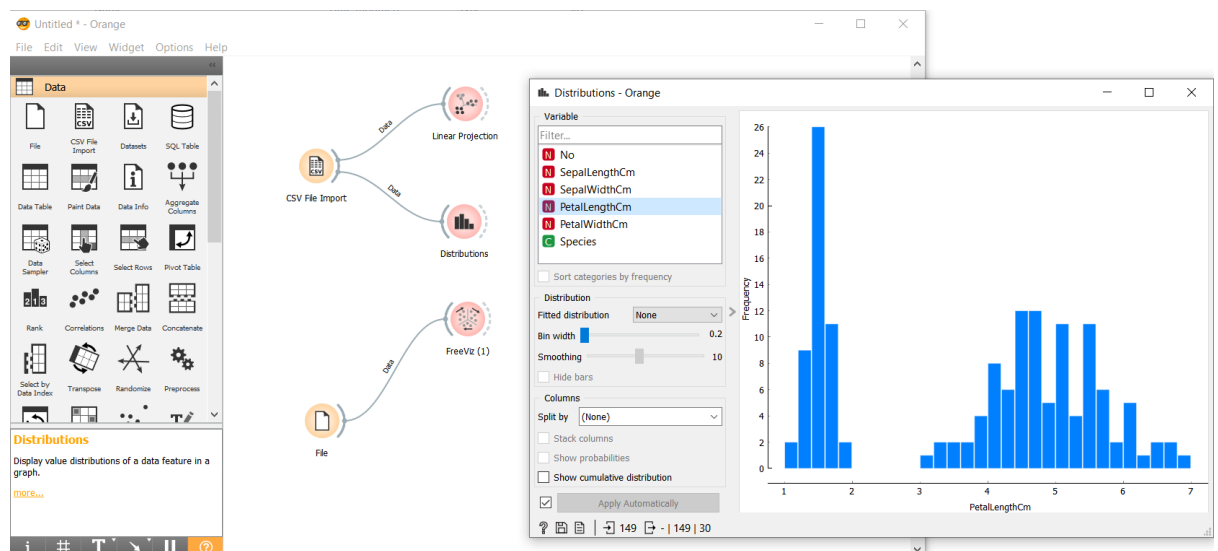
Use dataset -

<https://drive.google.com/file/d/1m6sKI1Dap0XK6Bw1edUd5PohwPpWxnd9/view>

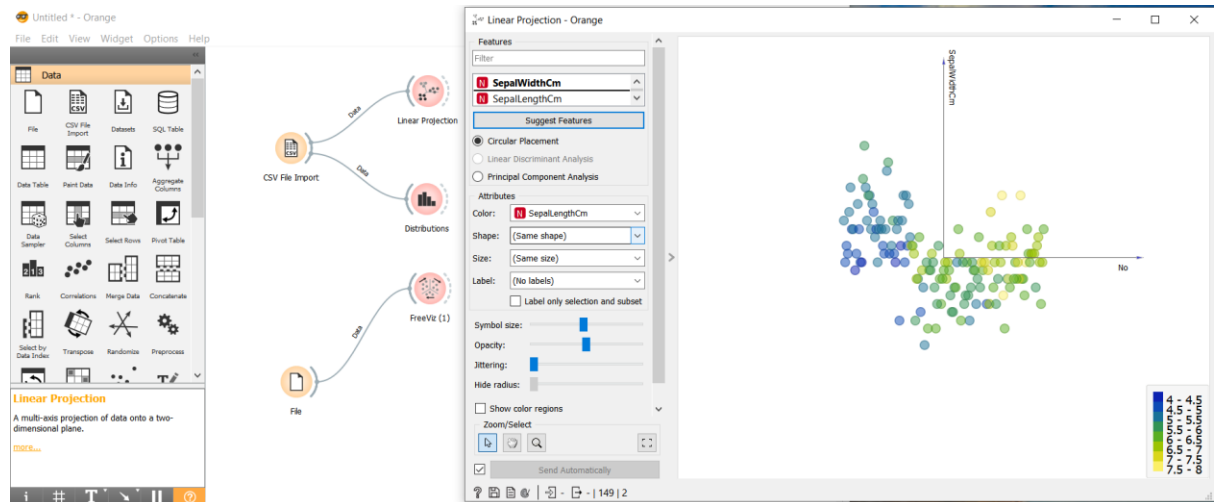
Create a report for this task and upload screenshots for the same.

Ans:

Data Distribution

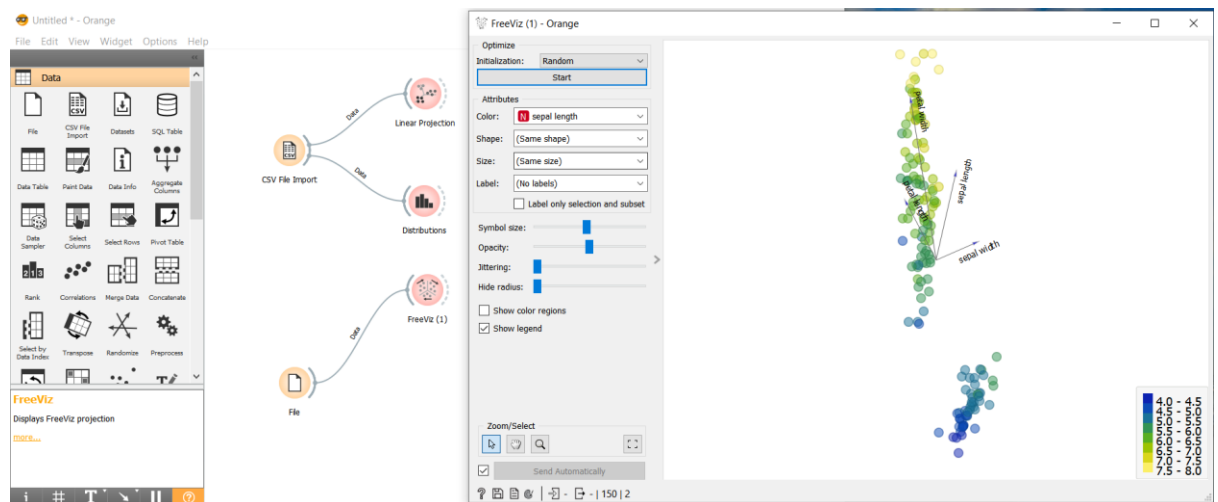


Linear Projection

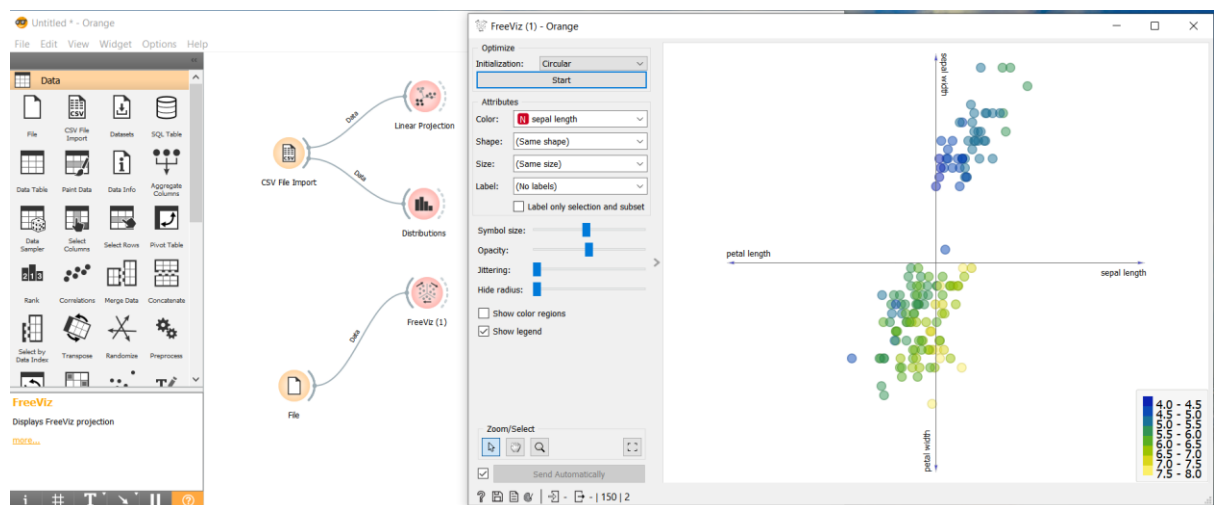


Free Viz

a. Random



b. Circular



3. Differentiate in between free software, Open-source software and proprietary software with respect to its properties.

Ans:

Parameter of Comparison	Free Software	Open-Source Software	Proprietary Software
Definition	“Free software” means software that respects users’ freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software.	Open-Source software refers to software that contains a source code with license which is open to free use, modification and distribution.	Proprietary software refers to the type of software that contains a licensed source code and is copyrighted for use.
Collaboration	The term “free software” is sometimes misunderstood—it has nothing to do with price. It is about freedom	Open-Source software is developed for open collaboration.	Proprietary software is not meant for open collaboration, but only for the creator and users who have paid for it.
Access	Software freedom translates to social freedom.	Open-Source software has open access, that is, can be accessed by anyone.	Proprietary software can be accessed only by those who developed it and those who have paid for it.

Flexibility	Free Software Is most Flexible, it can used by anyone.	Open-Source software is flexible, that is, it can be used, modified and distributed by anyone	Proprietary software has restricted flexibility, that is, there are restrictions on its usage.
Examples	The Free Software Directory maintains a large database of free- software packages. Some of the best-known examples include the Linux kernel, the BSD and Linux operating systems, the GNU Compiler Collection and C library; the MySQL relational database; the Apache web server; and the sendmail mail transport agent.	FreeBSD (Berklee Software Distribution), Android, LibreOffice, Ubuntu are a few examples of Open-Source software.	Windows, Microsoft, macOS, Adobe Photoshop , Adobe Flash Player are a few examples of Proprietary software.

4. Using Anaconda Python create Histogram, Scatter Plot and Bar plot for the dataset given below.

Dataset –

<https://drive.google.com/file/d/1i11BZFe8Xj9kNq7eeE9KOaIz1KhEdXJ/view>

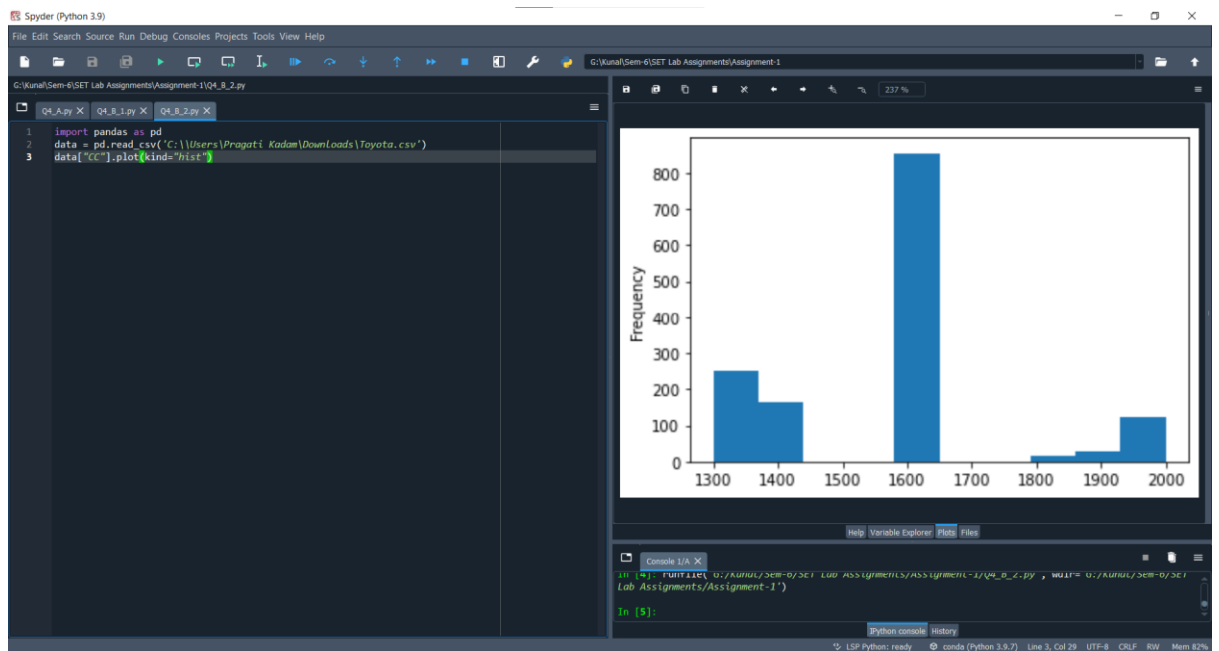
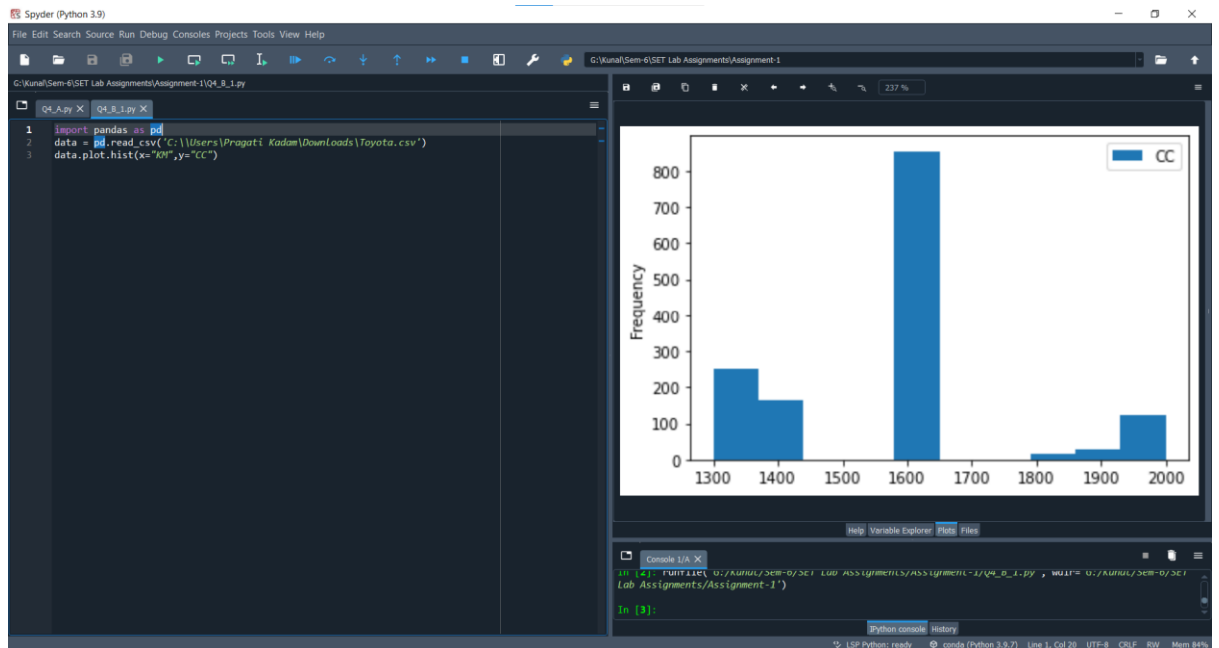
- a. Scatter plot – Scatter plot of Price Vs Age
- b. Histogram – for Kilometre and CC
- c. Bar plot – Bar plot for different fuel types

Ans:

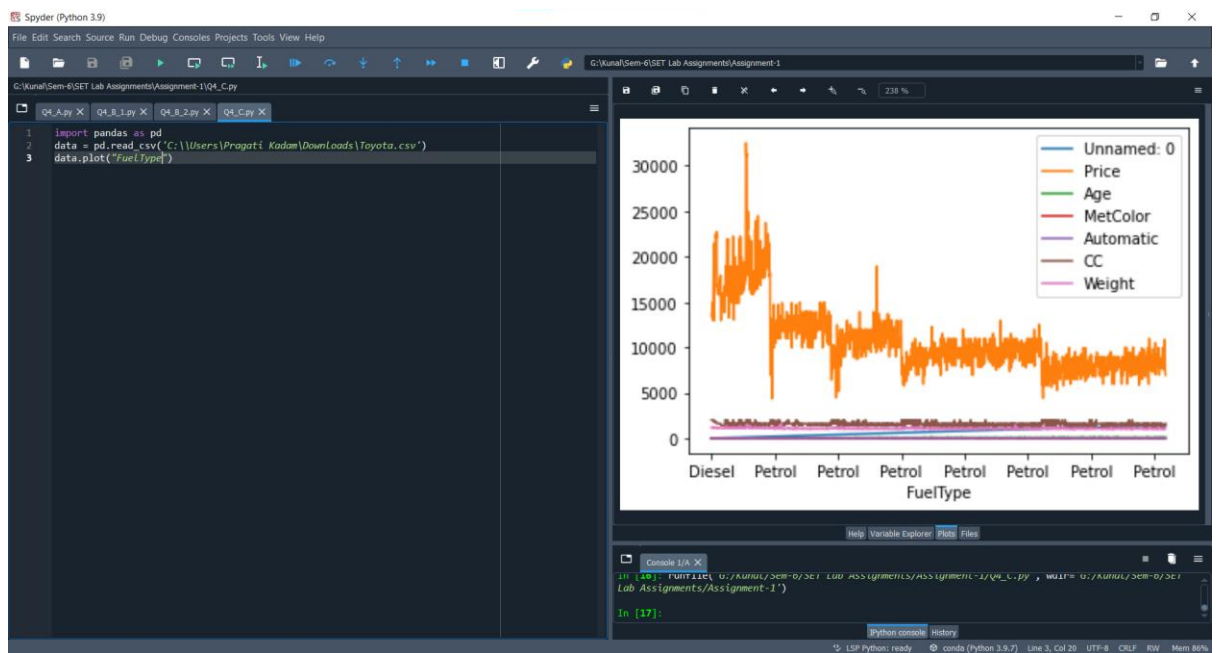
- a. Scatter Plot – Scatter Plot of Price Vs Age



b. Histogram for Kilometre and CC



c. Bar Plot – Bar plot for different fuel types



5. Enlist some examples along with its purpose and properties (at least 10) of FOSS and proprietary software with respect to database.

Ans:

FOSS:

Free and Open-source software (FOSS) is a software that can be classified as both free software and open-source software.

Examples: Ubuntu, VLC Player, Android System, etc.

Advantages of FOSS:

1. **Expansive Licensing:** Proprietary software licenses are usually quite restrictive in terms of use, number of users, type of machine and other. There is fee to own license of proprietary software. Open-source software are free to own there is no restriction on how we use the software, we can install it on unlimited machine.
2. **Transparency:** Open-source development is carried out openly. As software code is openly available anyone who finds some bug can fix it for others. As development process is carried out publicly its development process is transparent. Users can easily communicate with product developers to understand their product decisions and offer opinions for betterment of software.
3. **Source Code inspection:** As source code is openly available anyone can view code of the software for better understanding of how the application works.
4. **Source Code modification:** We can also modify code after inspecting the code. One can modify code and for himself and also, he can add those changes to main version of software so that changes will be available for everyone for use.
5. **Community:** Foundation of open-source projects is community; it includes developers of software and also users. Users in community can easily share their feedback with developers so that developers can improve the software.
6. **Redistribution rights:** Open-source licenses allow users to perform some changes in software and again redistribute the software without taking permission of the original product owner.

Proprietary Software:

This type of software requires licenses for their use. Company or organization that owns the software provides rights to use the software to customer. Users can only install software only on limited number of machines and cannot redistribute it.

1. Increased Functionality and Convenience

Proprietary systems are easier to use and learn, leading to faster work processes. Skype, for example, is used by organizations worldwide. It takes minutes to sign up for an account and make international phone calls or conduct video interviews online. On top of that, your customers, suppliers and employees may already have a Skype account, so they know how to use it.

Open-source programs are trickier to use and may lack user-friendly features, affecting productivity in the workplace. Unskilled end users may find it difficult to navigate them and take full advantage of what they have to offer. After all, there is a reason so few people use Linux.

2. Superior Customer Support

Open-source software can be difficult to install and set up. Customizing it isn't easier either. Plus, your staff may not be familiar with the program and may need additional training.

The average employee lacks the expertise to use open-source programs. Therefore, your team members may need help with most tasks. They will spend hours trying to figure things out instead of focusing on the tasks at hand.

Proprietary software is more accessible and includes technical support. Most companies offering these programs provide dedicated sources, 24/7 assistance, live chat and user manuals. The antivirus program Bitdefender, for example, offers online resources, technical

support around the clock and security-configuration services for enterprises. If your employees experience any issues, they can simply call or email the service provider.

3. Lower Maintenance Costs

As a small-business owner, you may prefer open-source software due to its low cost. Most programs are free or cost next to nothing. The downside is that you may end up paying a lot more for setup, maintenance and customization than you'd pay with proprietary software.

Some open-source programs are difficult to install and set up, so you may need to call an expert to do the job. In some cases, new hardware may be necessary to use the software. If your employees are not familiar with the program, they will need support and training, which may further increase the costs. Updating the software, testing new versions and applying patches isn't cheap either.

4. Stronger Competitive Advantages

Proprietary technology enables organizations to be more profitable, productive and innovative. This is particularly true for software-development companies, which often use custom programs at the core of their business model.

Even if you're not a software developer, you can still benefit from using proprietary systems. For example, you may hire a team to create software programs that integrate with your existing technology. This may improve work performance and productivity in your organization, streamline business processes and increase production. Furthermore, you may customize the program and add new features as your business changes.

5. Secure Financing for Your Business

Nearly one-third of start-ups close their doors because they run out of capital. Developing proprietary technology doesn't guarantee success, but it could make it easier to secure financing for your small business. Plus, you will be able to charge higher prices because no other company offers the same product as you do.