Software Engineering Tools Lab

Assignment No-1

(Module 1- Introduction to OSS)

Batch – T8

Name: Sayali Yogesh Desai

PRN: 2020BTECS00206

---

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

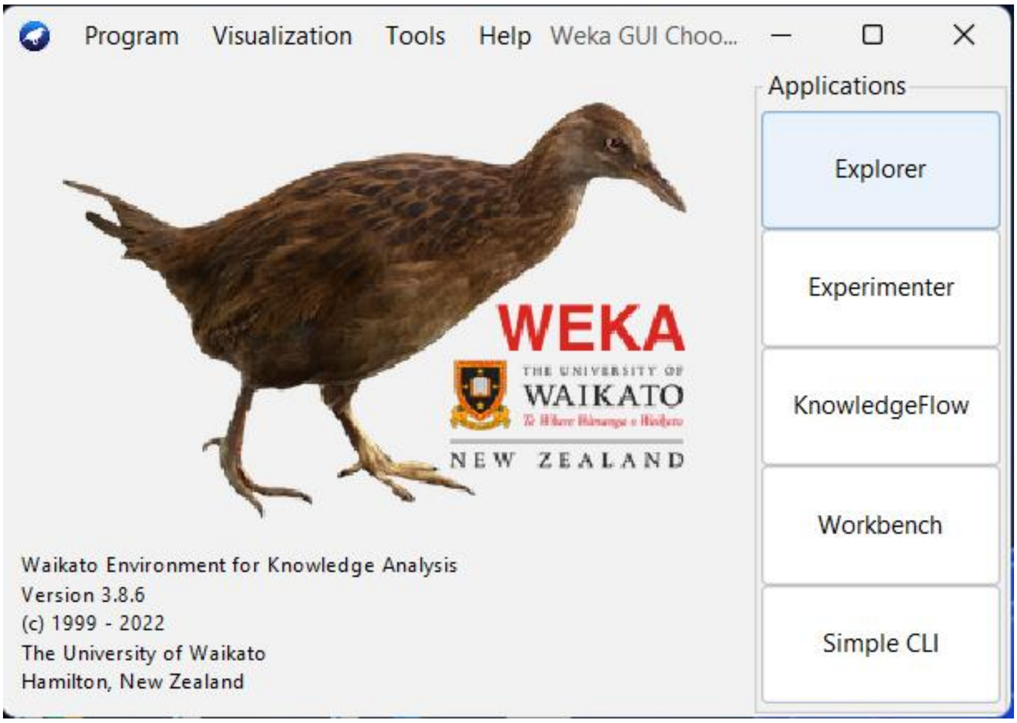
https://drive.google.com/file/d/1A3Fxsfzm6BSfhFZGDrjI47RTe45bSgYP/view

Note-provide screen shots for every task

Create a report which will illustrate the details of tasks performed

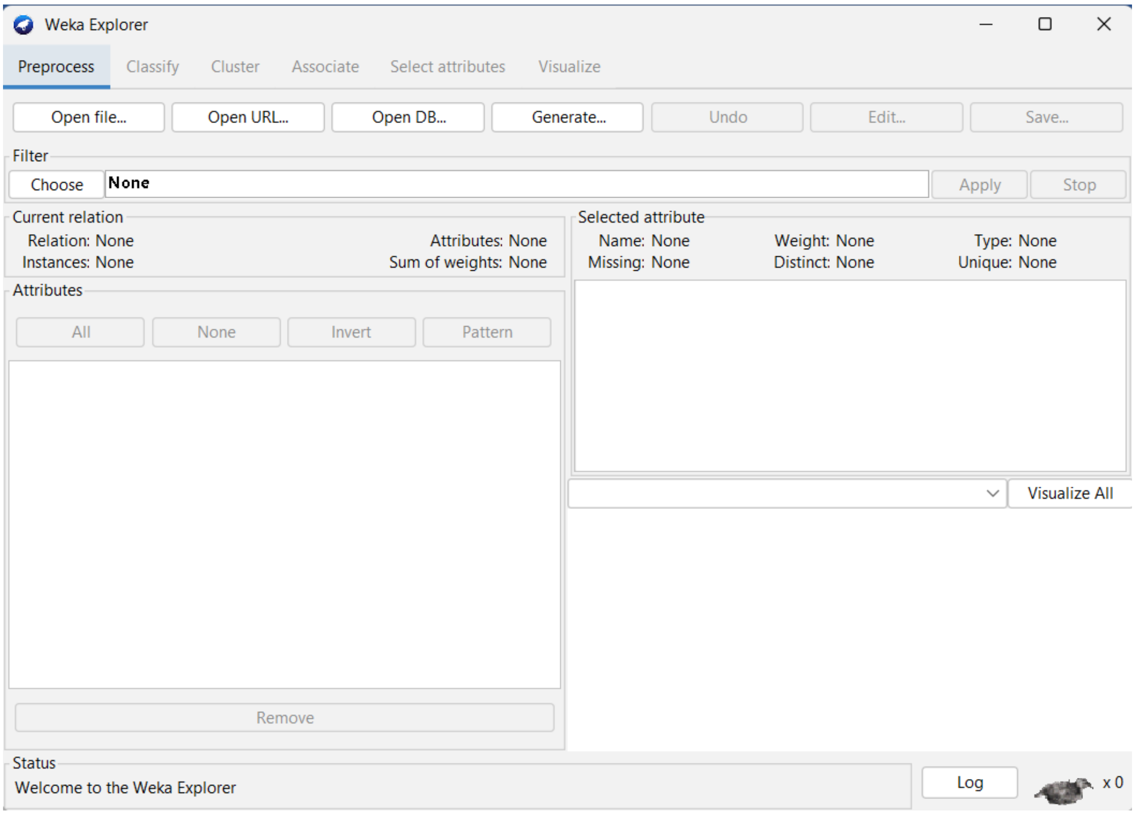
(For e.g to perform pre-processing of data provide details of navigation and selection of appropriate parameters)

- Open Weka



Waikato Environment for Knowledge Analysis
Version 3.8.6
(c) 1999 - 2022
The University of Waikato
Hamilton, New Zealand

- Select Explorer



Filter: None

Current relation

Relation: None Attributes: None
Instances: None Sum of weights: None

Selected attribute

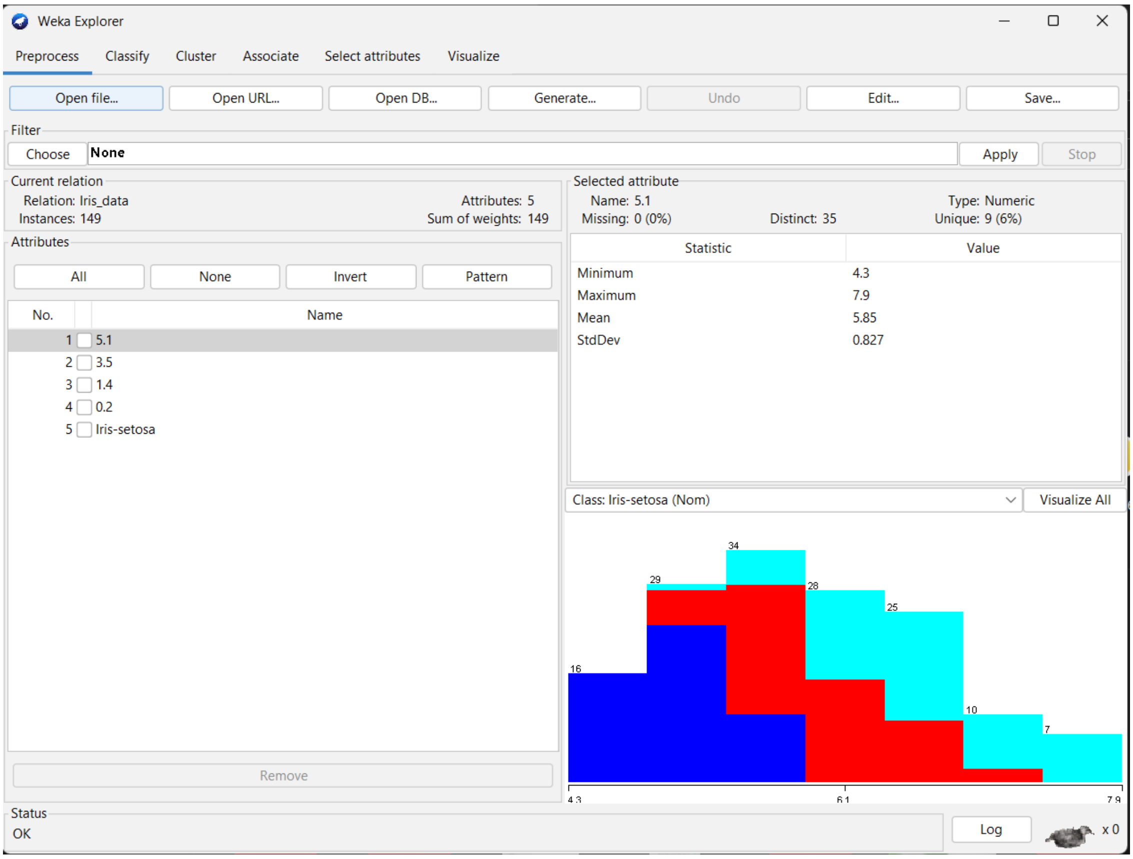
Name: None Weight: None Type: None
Missing: None Distinct: None Unique: None

Attributes

Status

Welcome to the Weka Explorer

- Open iris_data.csv file



Filter: None

Current relation

Relation: Iris_data Attributes: 5
Instances: 149 Sum of weights: 149

Selected attribute

Name: 5.1 Type: Numeric
Missing: 0 (0%) Distinct: 35 Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.85
StdDev	0.827

Attributes

No.		Name
1	□	5.1
2	□	3.5
3	□	1.4
4	□	0.2
5	□	Iris-setosa

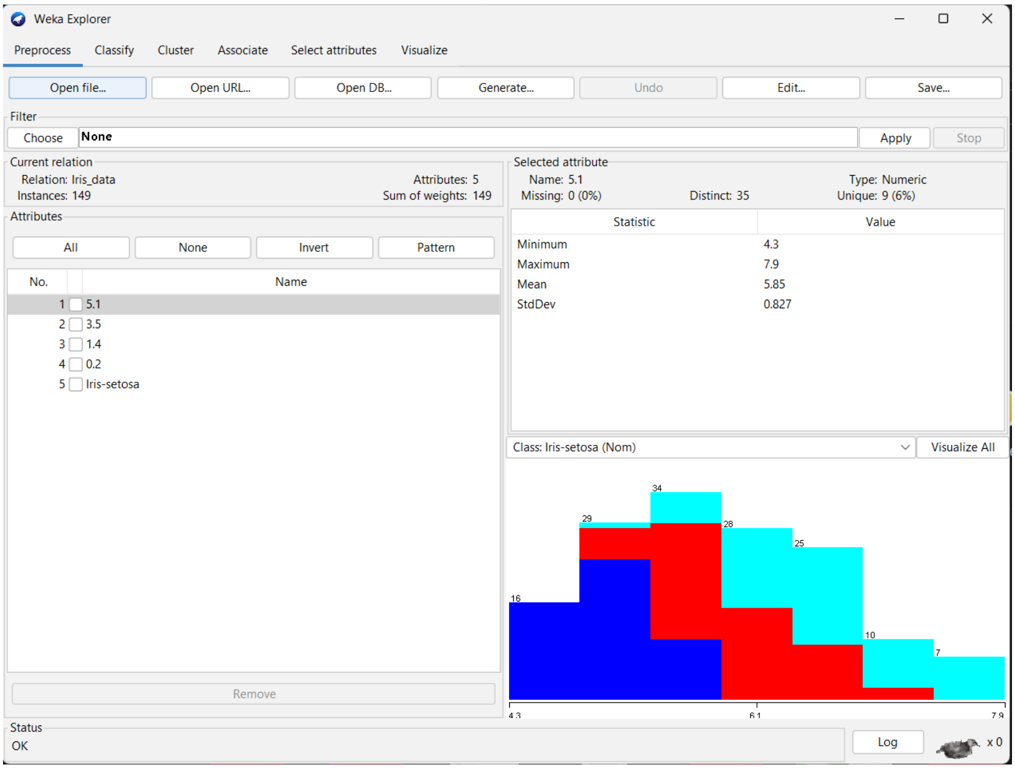
Class: Iris-setosa (Nom)

Status

OK

Pre-processing

- For adding pre-process filter, click on filter and select filter



Filter: None

Current relation
Relation: Iris_data	Attributes: 5
Instances: 149	Sum of weights: 149

Selected attribute
Name: 5.1	Type: Numeric
Missing: 0 (0%)	Distinct: 35	Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.85
StdDev	0.827

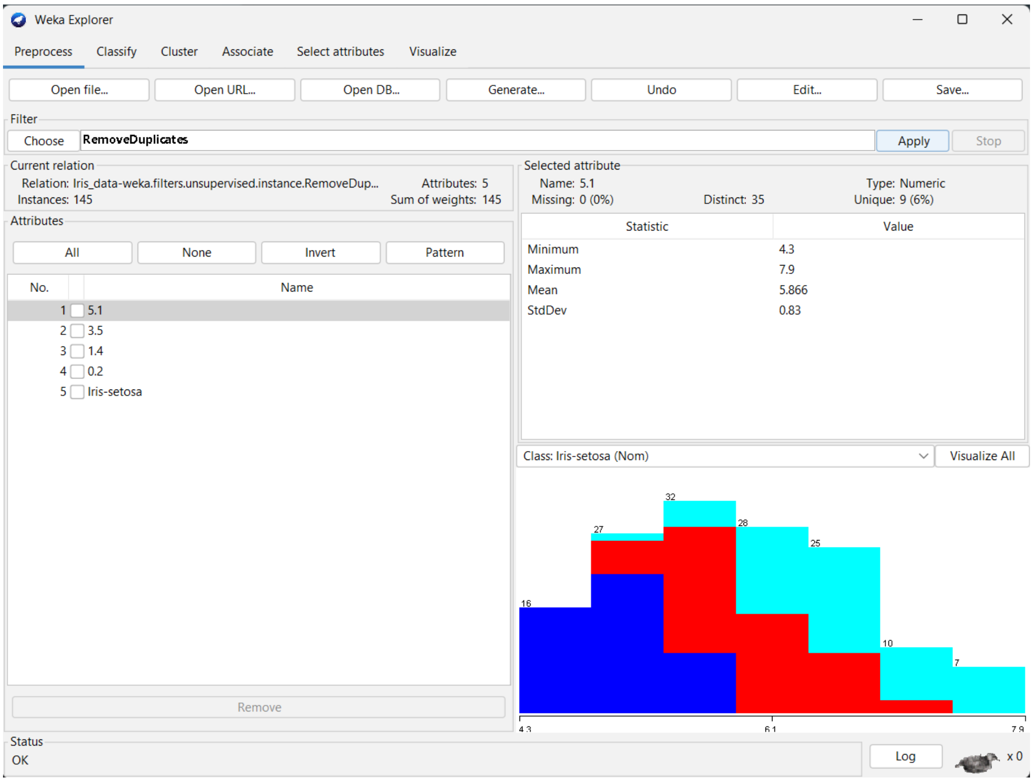
Attributes

No.	Name
1	5.1
2	3.5
3	1.4
4	0.2
5	Iris-setosa

Class: Iris-setosa (Nom)

Status
OK

- After Applying



Filter: RemoveDuplicates

Current relation
Relation: Iris_data-weka.filters.unsupervised.instance.RemoveDup...	Attributes: 5
Instances: 145	Sum of weights: 145

Selected attribute
Name: 5.1	Type: Numeric
Missing: 0 (0%)	Distinct: 35	Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.866
StdDev	0.83

Attributes

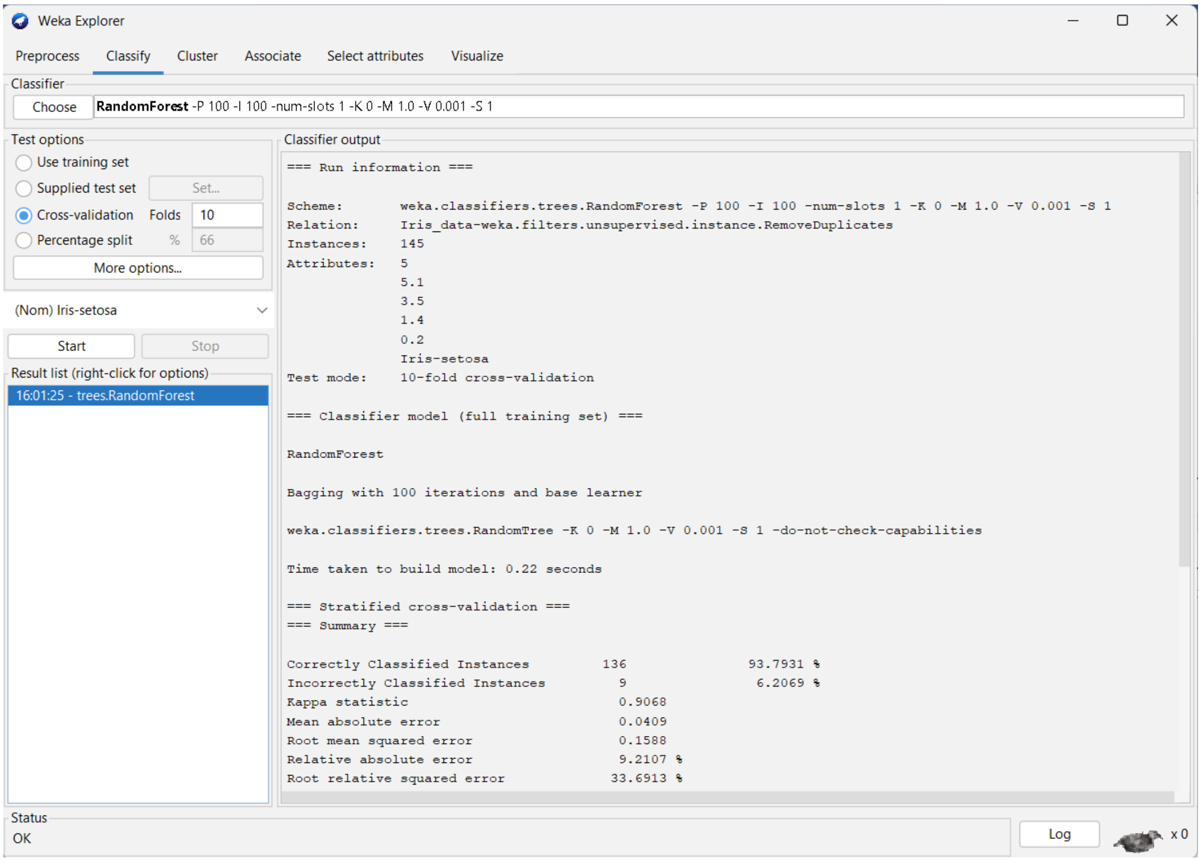
No.	Name
1	5.1
2	3.5
3	1.4
4	0.2
5	Iris-setosa

Class: Iris-setosa (Nom)

Status
OK

Classification

- Select the classify option and select appropriate method from filter
- I have chosen Random Forest under Trees
- After that we can adjust the options and hit start to see result
- I have selected cross validation with 10 folds



RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) Iris-setosa

16:01:25 - trees.RandomForest

```
=== Run information ===

Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Relation:     Iris_data-weka.filters.unsupervised.instance.RemoveDuplicates
Instances:    145
Attributes:   5
              5.1
              3.5
              1.4
              0.2
              Iris-setosa
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.22 seconds

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

Correctly Classified Instances         136               93.7931 %
Incorrectly Classified Instances         9                6.2069 %
Kappa statistic                          0.9068
Mean absolute error                      0.0409
Root mean squared error                  0.1588
Relative absolute error                  9.2107 %
Root relative squared error             33.6913 %
```

Status
OK

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

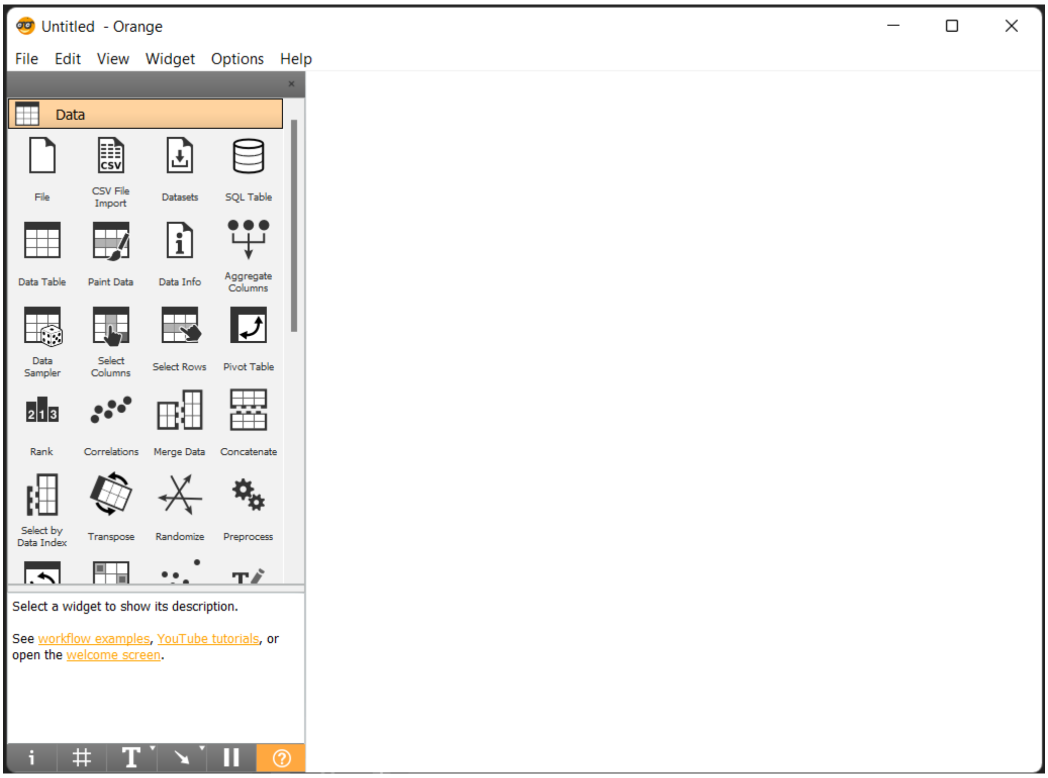
d. Show FreeViz

Use dataset

https://drive.google.com/file/d/1m6sKI1Dap0XK6Bw1edUd5PohwpPwXnd9/view

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

- Install and Open Orange



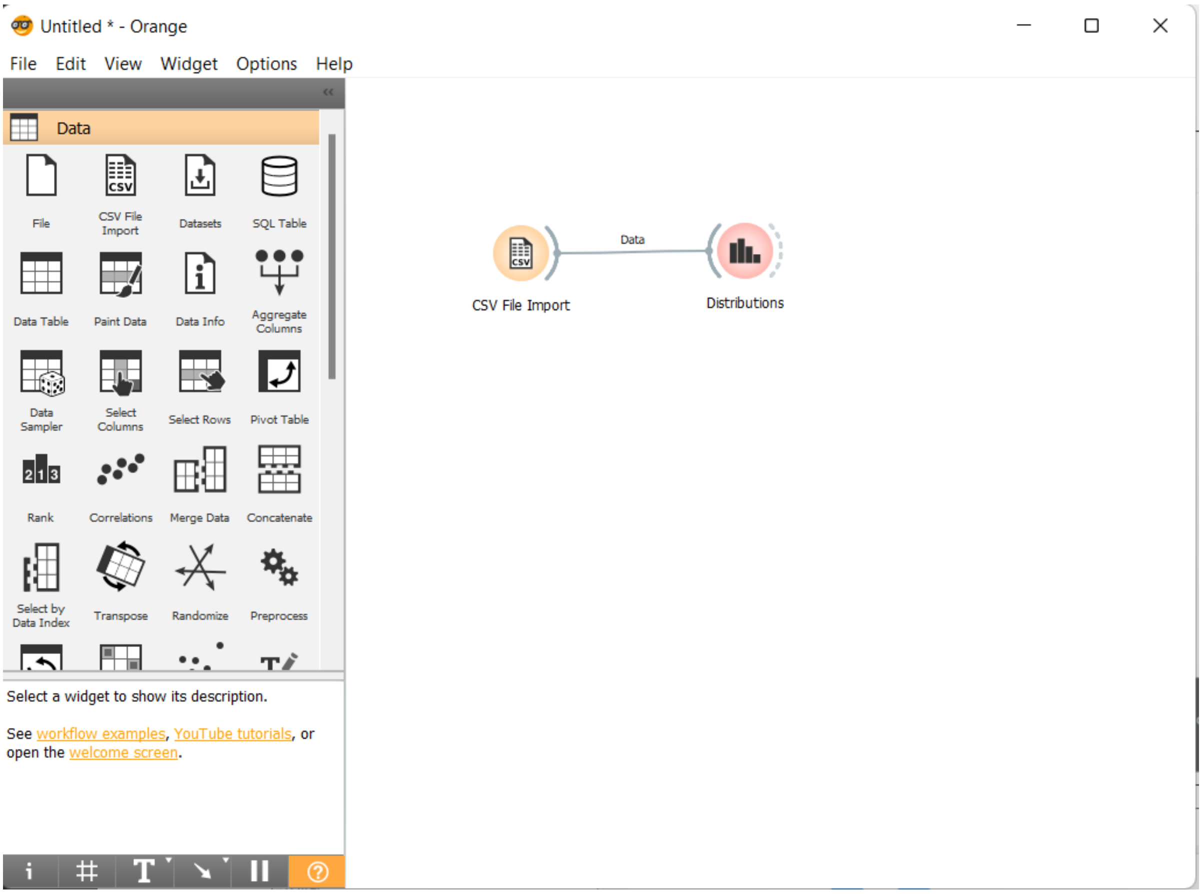
Select a widget to show its description.

See workflow examples, YouTube tutorials, or open the welcome screen.

Data Distribution

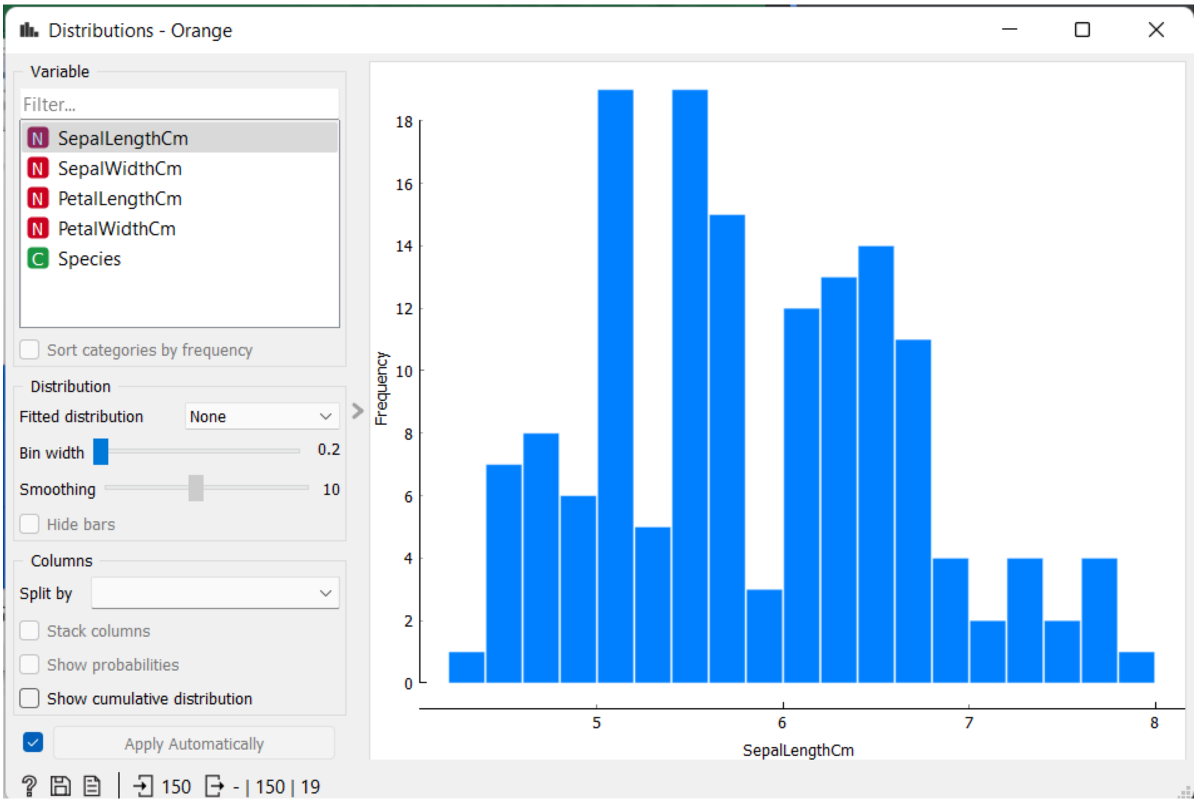
Select CSV import and choose dataset file

Select CSV import icon and then drag and search distribution.



Select a widget to show its description.

See workflow examples, YouTube tutorials, or open the welcome screen.



□ Sort categories by frequency

Fitted distribution: None

Bin width: 0.2

Smoothing: 10

□ Hide bars

Split by:

□ Stack columns

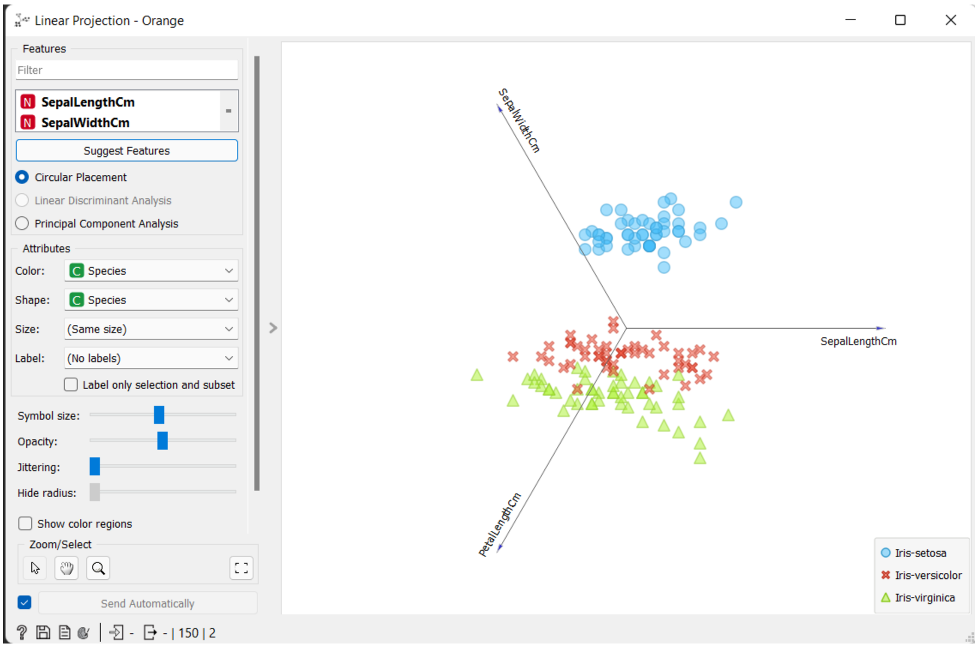
□ Show probabilities

□ Show cumulative distribution

☑ Apply Automatically

Linear Projection

Select Import CSV file and drag and search projection



Linear Projection - Orange

Features

Filter

SepalLengthCm

SepalWidthCm

Suggest Features

● Circular Placement

○ Linear Discriminant Analysis

○ Principal Component Analysis

Attributes

Color: Species

Shape: Species

Size: (Same size)

Label: (No labels)

□ Label only selection and subset

Symbol size:

Opacity:

Jittering:

Hide radius:

□ Show color regions

Zoom/Select

☑ Send Automatically

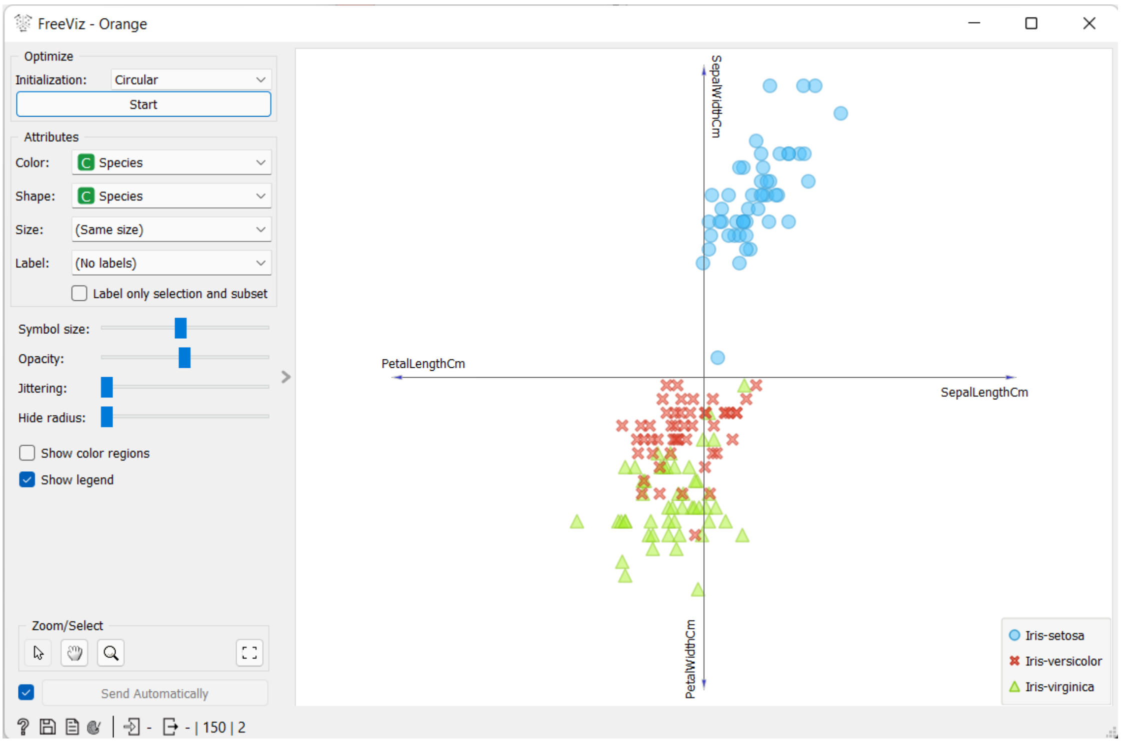
150 | 2

FreeViz

Drag from Import CSV and select Select Column

Select target variable

Drag from Select Column and search FreeViz



FreeViz - Orange

Optimize

Initialization: Circular

Start

Attributes

Color: Species

Shape: Species

Size: (Same size)

Label: (No labels)

□ Label only selection and subset

Symbol size:

Opacity:

Jittering:

Hide radius:

□ Show color regions

☑ Show legend

Zoom/Select

☑ Send Automatically

150 | 2

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

1. Free Software:

- Free software (or libre software is computer software distributed under terms that allow users to run the software for any purpose as well as to study, change, and distribute it and any adapted versions.
- Free software is a matter of liberty, not price; all users are legally free to do what they want with their copies of a free software (including profiting from them) regardless of how much is paid to obtain the program.
- Computer programs are deemed "free" if they give end-users (not just the developer) ultimate control over the software and, subsequently, over their devices.
- The right to study and modify a computer program entails that source code— the preferred format for making changes—be made available to users of that program.
- While this is often called "access to source code" or "public availability", the Free Software Foundation (FSF) recommends against thinking in those terms, because it might give the impression that users have an obligation (as opposed to a right) to give non-users a copy of the program.

2. Open-source Software:

- Open-source software is computer software whose source code is available openly on the internet and programmers can modify it to add new features and capabilities without any cost.
- Here the software is developed and tested through open collaboration.
- This software is managed by an open-source community of developers.
- It provides community support as well as commercial support if available for maintenance.
- We can get it for free of cost.
- This software also sometimes comes with a license and sometimes does not.
- This license provides some rights to users.
  i. Software can be used for any purpose
  ii. Allows studying how the software works
  iii. Freedom to modify and improve the program
  iv. No restrictions on redistributions Some examples of Open source software includes Android, Ubuntu, Firefox, Open Office etc.

3. Proprietary Software:

- Proprietary software is computer software where the source codes are publicly not available only the company that has created can modify it.
- Here the software is developed and tested by the individual or organization by which it is owned not by the public.
- This software is managed by a closed team of individuals or groups that developed it.

- We have to pay to get this software and its commercial support is available for maintenance.
- The company gives a valid and authenticated license to the users to use this software.
- But this license put some restrictions on users also like.
  1. the number of installations of this software into computers
  2. Restrictions on sharing of software illegally
  3. Time period up to which software will operate
  4. Number of features allowed to use

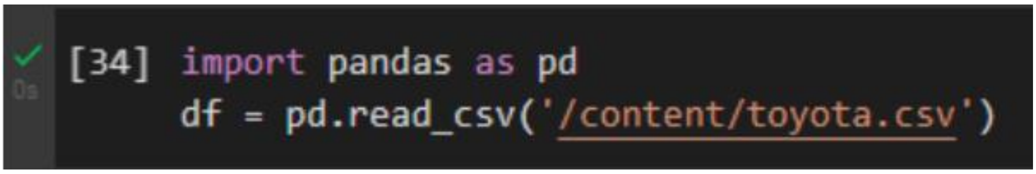
4. Using Anaconda Python create Histogram, Scatter plot and Bar plot for the dataset given below.

Dataset- https://drive.google.com/file/d/1i11BZFe8Xj9kNq7eeE9KOa_Iz1KhEdXJ/view

a. Scatter plot- Scatter plot of Price Vs Age

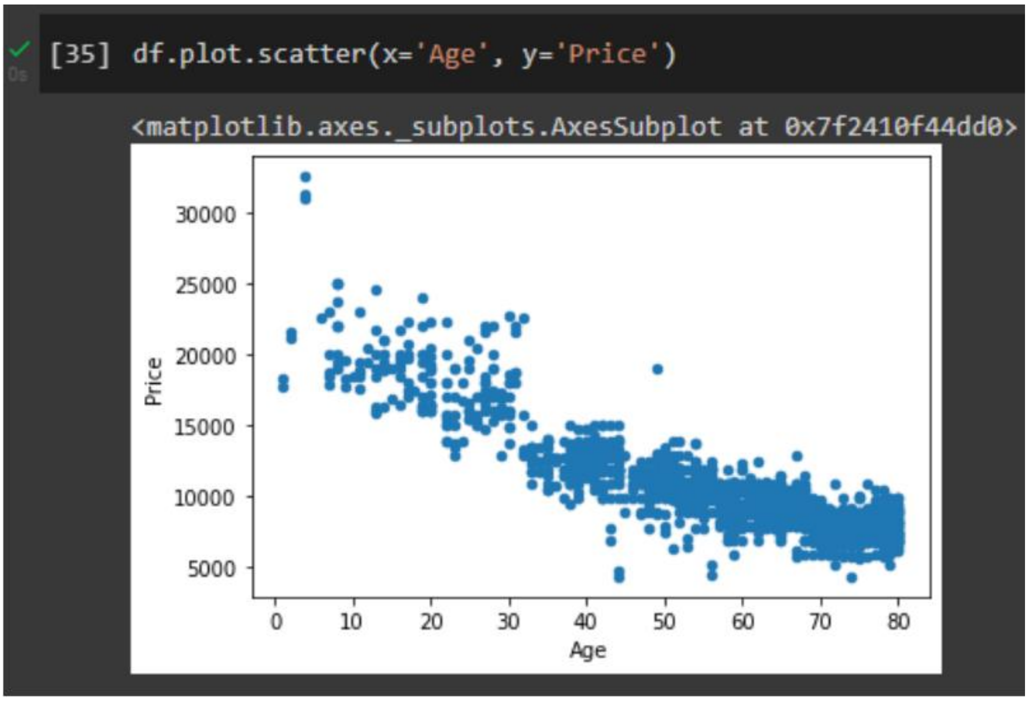
b. Histogram- for Kilometre and CC

c. Bar plot- Bar plot for different fuel types



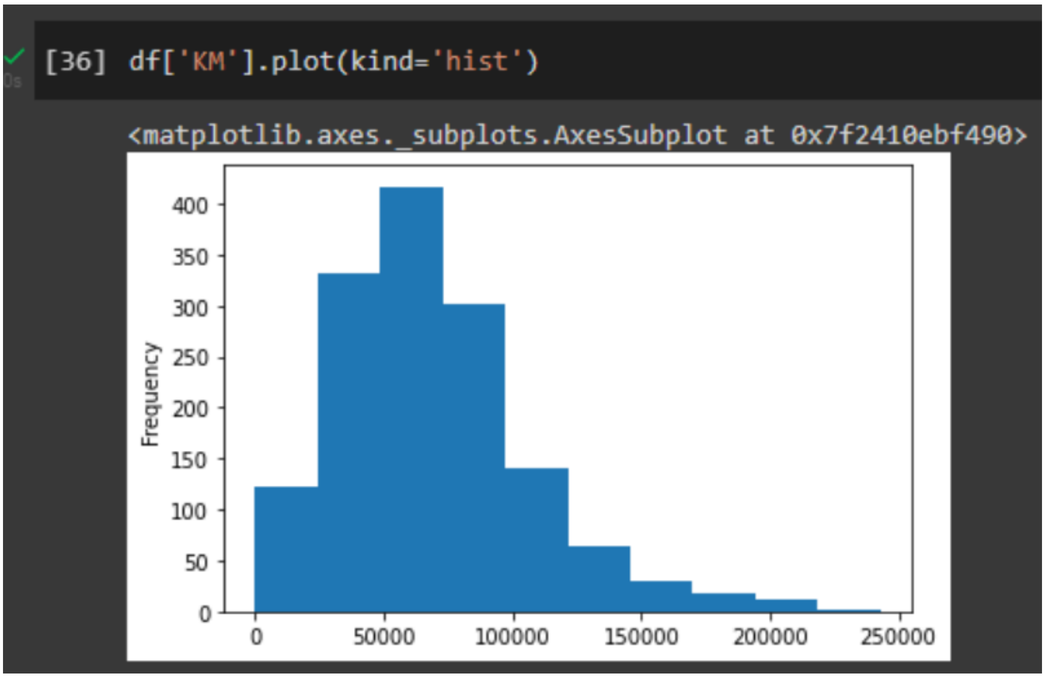
```
[34] import pandas as pd
     df = pd.read_csv('/content/toyota.csv')
```

Scatter plot – Price vs Age



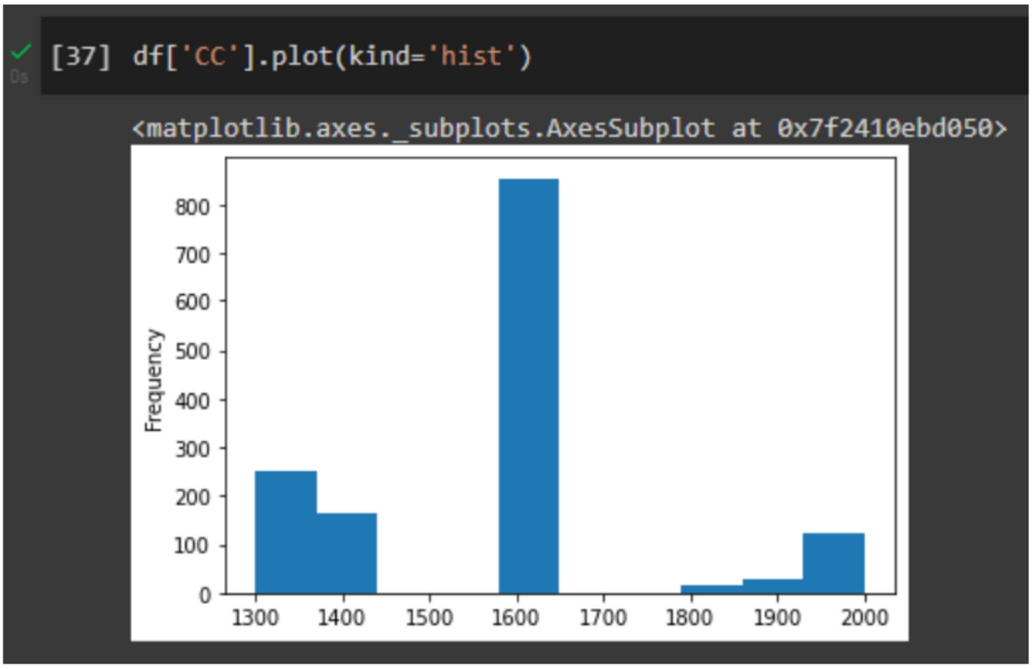
```
[35] df.plot.scatter(x='Age', y='Price')

<matplotlib.axes._subplots.AxesSubplot at 0x7f2410f44dd0>
```

Histogram – Kilometer and CC



```
[36] df['KM'].plot(kind='hist')
```

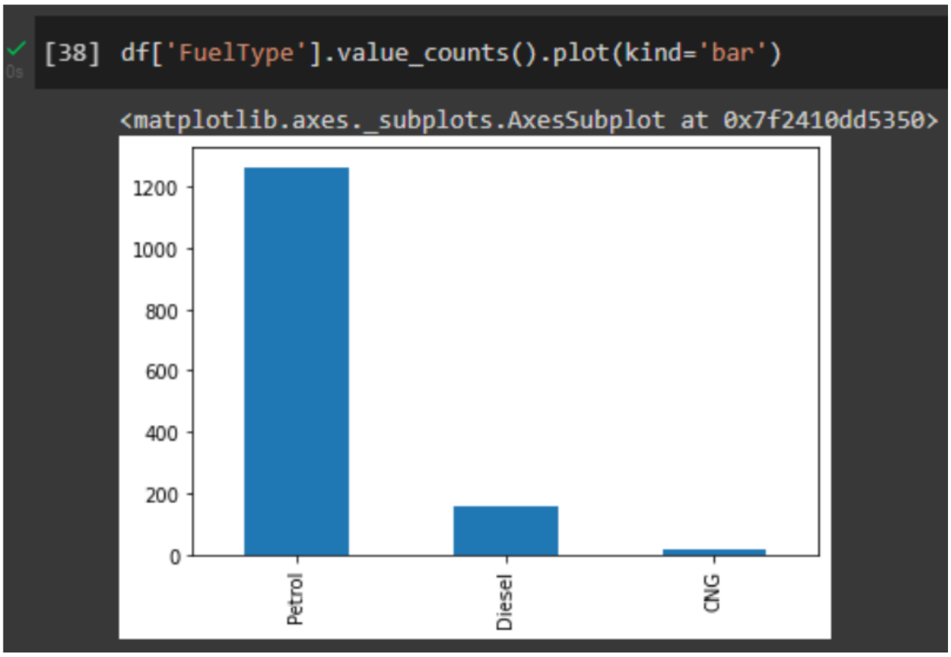
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2410ebf490>
```



```
[37] df['CC'].plot(kind='hist')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2410ebd050>
```

Bar Plot – Different Fuel Types



```
[38] df['FuelType'].value_counts().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2410dd5350>
```

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

1. PostgreSQL

This relational database software has been around since 1997 and is the top choice in communities like Ruby, Python, Go, etc.

2. MariaDB

MariaDB was created as a replacement for MySQL by the same person who developed MySQL.

3. CockroachDB

The idea behind "cockroach" is that it's an insect built for survival. No matter what happens — predators, floods, eternal darkness, rotting food, bombing, the cockroach finds a way to survive and multiply.

4. ClickHouse

It uses every hardware to its maximum potential to approach each query faster. The peak performance of processing a query usually remains more than two terabytes each second.

5. Neo4j

Support for transactional applications and graph analytics. Data transformation abilities for digesting large-scale tabular data into graphs. Specialized query language (Cypher) for querying the graph database Visualization and discovery features

6. Redis

When it comes to databases, it’s almost too easy to overlook the existence of Redis. That’s because Redis is an in-memory database and is mostly used in support functions like caching.

7. SQLite

SQLite is a lightweight C library that provided a relational database storage engine. Everything in this database lives in a single file (with a .sqlite extension) that you can put anywhere in your filesystem. And that’s all you need to use it! Yes, no “server” software to install and no service to connect to.

8. Cassandra

Cassandra belongs to what’s known as the “columnar” family of databases. The storage abstraction in Cassandra is a column rather than a row. The idea here is to store all the data in a column physically together on the disk, minimizing seek time.

9. Timescale

The timescale is a type of what’s called a “time series” database. It’s different from a traditional database in that time is the primary axis of concern, and the analytics and visualization of massive data sets is a top priority

10. CouchDB

It is a neat little database solution that sits quietly in a corner and has a small but dedicated following. It was created to deal with the problems of a net loss and eventual resolution of data, which happens to be a problem so messy that developers would instead switch jobs than deal with it.