**Open-source machine learning packages**

Some of top listed machine learning packages are

* ***Apache Mahout:*** It provides way to host machine learning project. It's mainly work with Apache project name spark.
* ***Core ML Tools:*** This is a ML framework makes ML available for application programming interface.
* ***Cortex:*** Cortex makes way to use Python, TensorFlow, PyTorch, Scikit-learn and others building machine learning model. It offers to use Docker for hosting environment using cortex.yml file.
* ***GoLearn:*** It’s a ML library based on Go language by Google. Data loading & handling is more simpler than other in GoLearn.
* ***Gradio:*** It givesthe feature of customized UI for model training and prediction based service. Gradio support real-time interaction with models.
* ***H2O:*** In memory machine learning platform prepared for business purpose. Java, Python, R and Scala are allowed to interact with H2O.
* ***Oryx:*** Oryx use Hahoop distribution these are Apache Kafka, Apache Spark for machine learning model provides real-time feature.
* ***PyTorch Lightning:*** Wrapper of pyTorch that make it easy to use is a third party library.
* ***Scikit-learn:*** Makes math and science work more easier leveraging existing python packages like Numpy, SciPy and Matplotlib.
* ***Shogun:*** I is written in C++, but Java, Python, C#, Ruby, R & Matlab can also use it.
* ***Spark MLlib:*** Java is the primary language to work with MLLib is a library forApache Hadoop and Spark. It is faster and scalable.
* ***Weka:*** It has GUI feature for creating and training models without writing code.

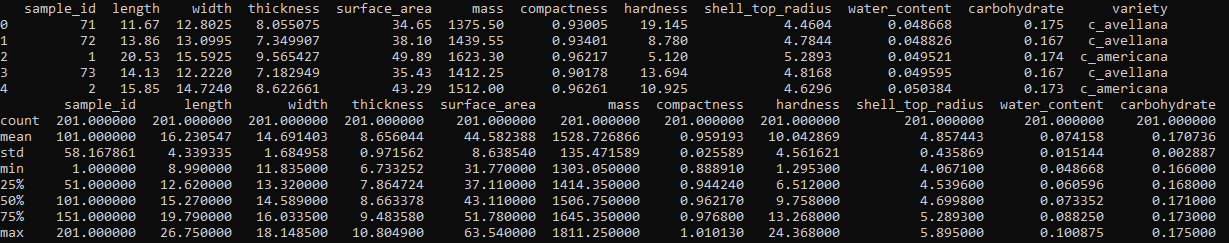
Here I select to use `**Scikit-Learn`** for this classification task. Scikit-Learn is powerful and robust library for machine learning in python. It is simple and has efficient tools for math and science work.

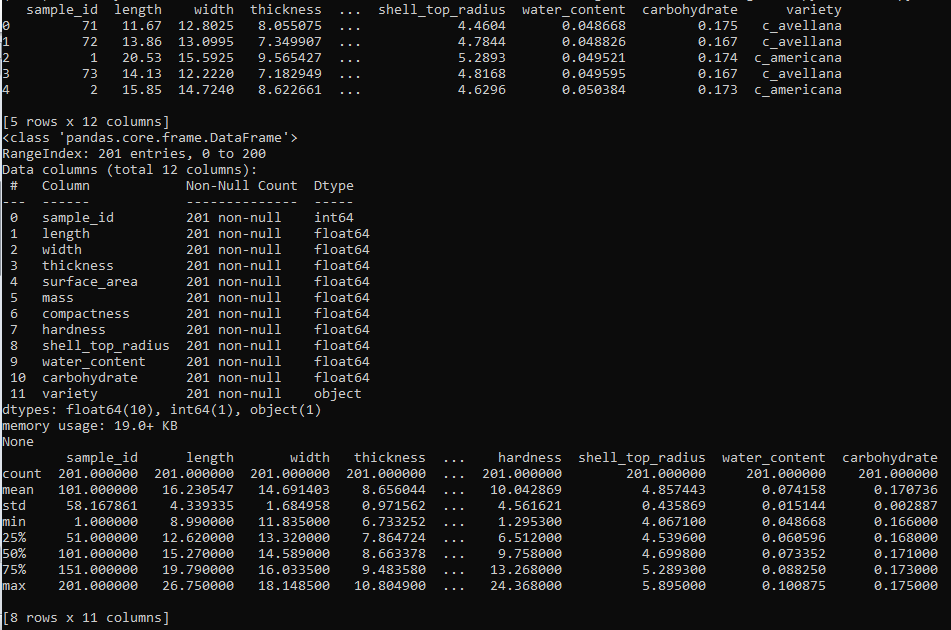
**Prepare datasets**

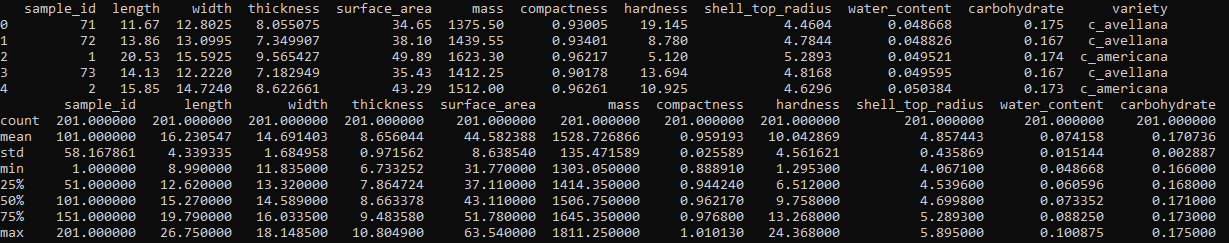
I found dataset and analyzing this data set I will prepare this for fitting in model. I’ll show step of code for analyzing and preparing this dataset for our model.

1. First load our dataset using panda and view sample of our dataset, its column information and numerical summery of numeric column like standard deviation, mean, min & max value.







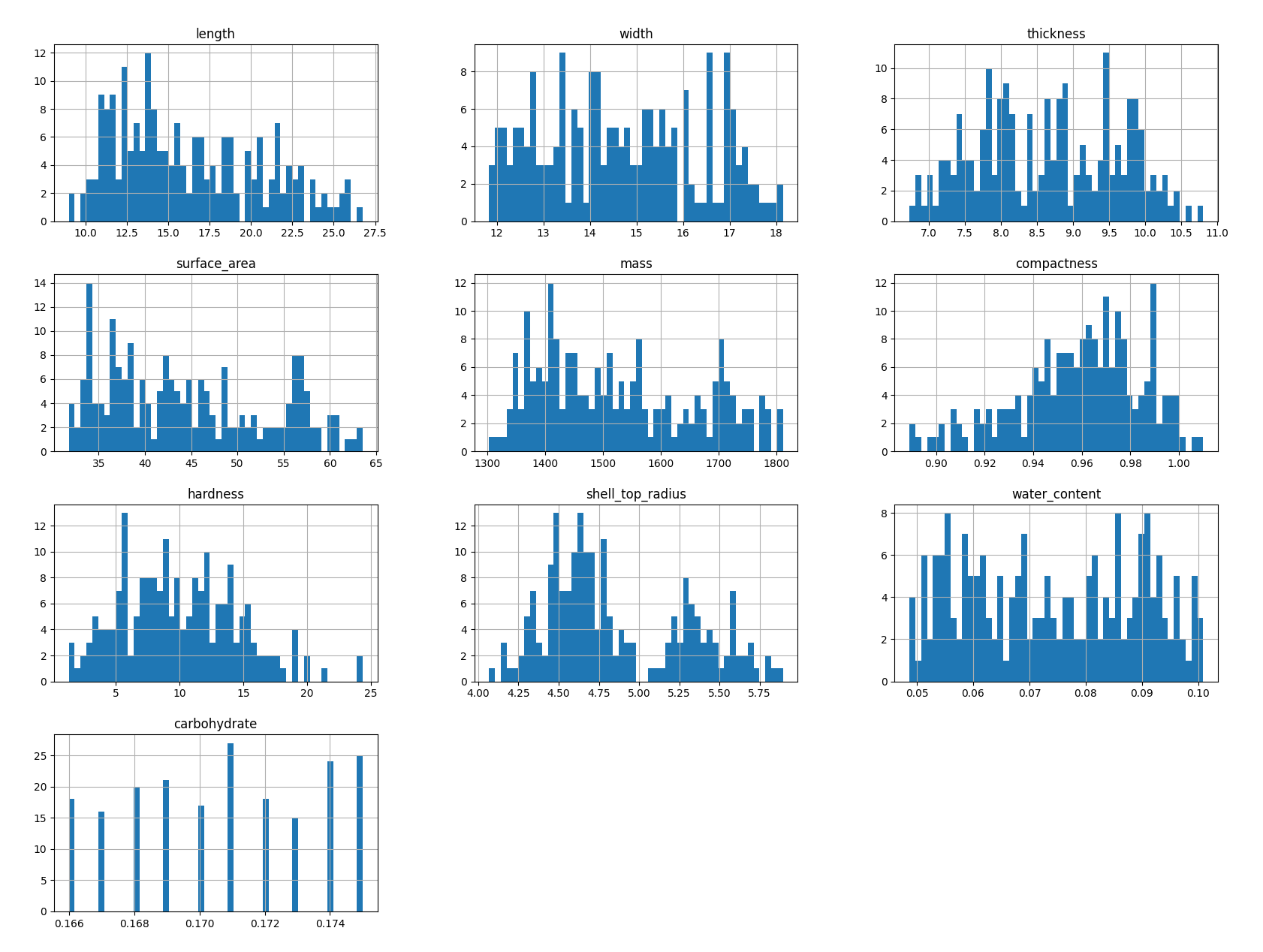


1. Here `sample\_id` is not the part of our data, first remove this.



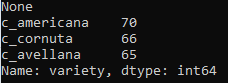
1. To get feel of the type of dataset lets plot a histogram for numerical values. Here I will use `matplotlib`.





1. `Variety` is the label of our dataset. It’s is an object type data. Let’s see how many category belongs to these label. For in depth



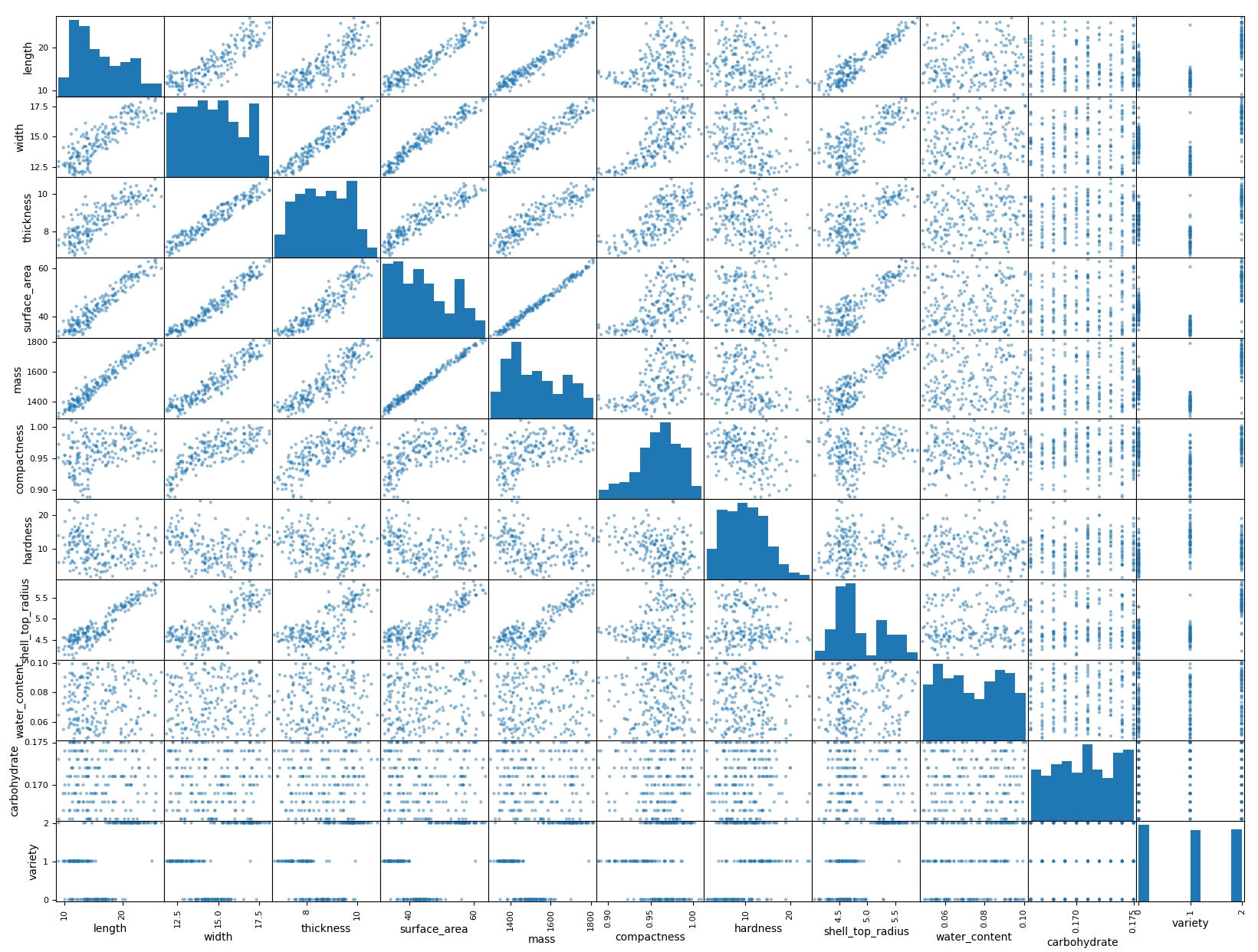


1. Let’s have a sight on the correlation of the attributes of the datasets. As we are to find standard correlation coefficient then all data should be numerical. Then let’s convert the target label object of `variety` to numerical attributes.



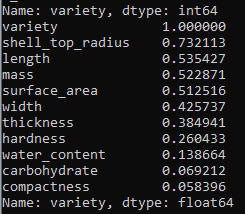
Then create the correlation matrix. We may also have a sight on graph.





And have sight on numerical correlation of `variety`.





1. Let’s split our dataset into training set and test set. First separate our feature and prediction on X Y.



1. In few cases machine learning algorithms don’t perform well when input numerical attributes have very different scales. So let range these feature X.



1. Then obtain train set and test set from them



Now our dataset is ready to feed on our model.

**Classification models**

**Let’s apply our dataset in `KNN` (K Nearest Neighbor) model first.**



*Now understand how KNN model work.* KNN is used to solve both classification and regression problem. Here is the step KNN follow to solve problems.

1. Load the data.
2. Initialize k the number we chose our neighbor to compare our test value (query example) in particular.
   1. Calculate the distance between the query example and the current example
   2. Add the distance and the index in an order list
3. Sort (ascending) the ordered list by distance.
4. Select the k entries from top of list
5. Return the Mode of the k labels as it is classification. Otherwise mean in regression term.

**Let’s apply our dataset in another mode. I select `Random Forest` for now.**



*Now understand how ` Random Forest` model work.* Random forest is based on `Decision Tree` model, where more than one tree is created using Bagging (Bootstrapping+ Aggregations). Let’s see the steps it follows.

1. From dataset of K records model takes n number of records randomly for T times.
2. T number of times decision tree is created with their selected records.
3. Each decision tree generates an output.
4. Mode of all output is the required labels as it is classification. Otherwise mean of output, in regression term.

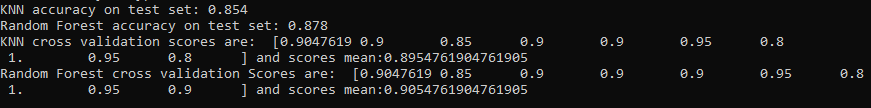
**Estimate future performance**

Let’s test our model with test set, then we will see 10 fold cross validation.



Now see the result of 10 fold cross validation for KNN and Random Forest model.





**Report analysis**

From our test set and mostly from 10 fold cross validation; result is not same but very similar. For KNN cross validation score is 0.895 and random forest score is 0.905. Random Forest gives better result than KNN in validation score. In our model of Random Forest n\_estimator=50, means it create 50 Decision Tree model in time of bagging and use Gini Impurity for measurement of split quality.

On the other hand our model KNN uses 5 neighbors for estimating mode of related feature.