Big Data Science (CSCI 7952)

Amin Kamranfar

University of Colorado Denver

04/025/2021

Contents

[Problem Statement and Background 3](#_Toc70298650)

[Statement of the problem and data characteristics 3](#_Toc70298651)

[Informal measures 4](#_Toc70298652)

[Background and summary of related work 5](#_Toc70298653)

[Placeholders for the in-progress items 6](#_Toc70298654)

[Methods 6](#_Toc70298655)

[Exploring methods 6](#_Toc70298656)

[Parametrized effect on performance 7](#_Toc70298657)

[Failing methods and rationale 7](#_Toc70298658)

[Results 8](#_Toc70298659)

[Comparing the baseline solution and primary solution 8](#_Toc70298660)

[Univariate Plots 9](#_Toc70298661)

[Multivariate Plots 10](#_Toc70298662)

[Visualizations and graphs 11](#_Toc70298663)

[Tools 12](#_Toc70298664)

[Describing the tools 12](#_Toc70298665)

[Employing the tools and their features 12](#_Toc70298666)

[Lesson Learned 14](#_Toc70298667)

[High-level summary 14](#_Toc70298668)

# Problem Statement and Background

## Statement of the problem and data characteristics

One of the interesting topics in data science field is analyzing different types of data according to their attributes. The iris dataset is one of the oldest datasets that has many enthralling features for data visualization, machine learning, and classification. The iris dataset has three different iris species: iris Setosa, iris Versicolour, and iris Virginica. Each species attribute information has been divided into diverse sepal and petal length. Figure 1 shows the iris flowers.

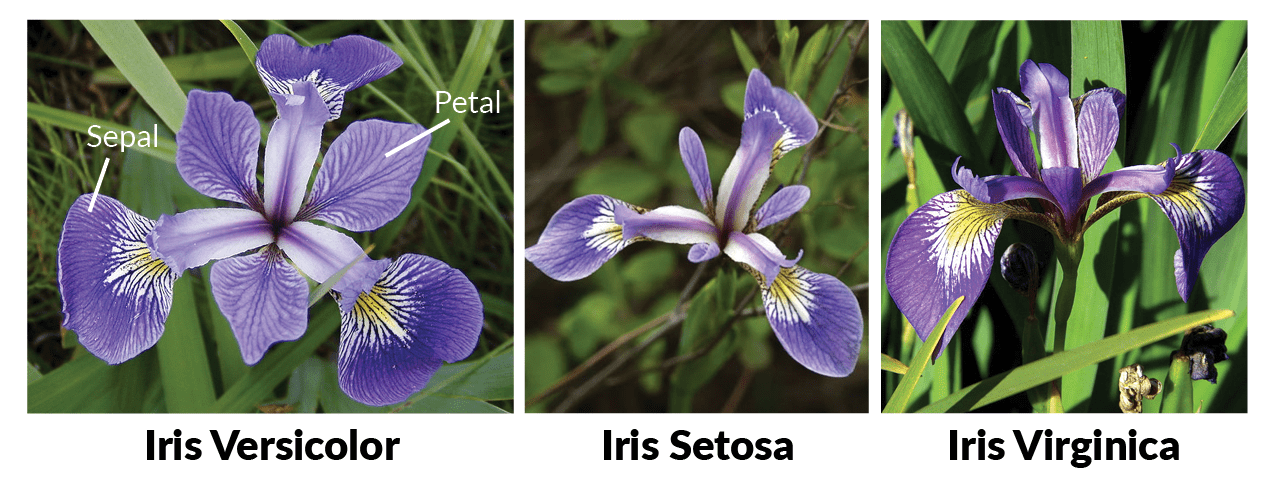


Figure 1. Iris Flower

## Informal measures

The "Iris flower data set" or "Fisher's Iris data set" is used to make quick predictions. The dataset consists of a set of 150 records and 5 attributes. These features are "Petal Length", "Petal Width", "Sepal Length", "Sepal Width" and tags. The labels in this dataset actually determine which species of iris each specimen is with its sepals and petals. Species in this dataset include "Iris Setosa", "Iris Virginica" and "Iris Versicolor". The data source is the file iris\_flowers.csv. It contains the data for this example in comma-separated values (CSV) format. The number of columns is 5, and the number of rows is 150.

The variables are:

* **sepal\_length**: Sepal length, in centimeters, used as input.
* **sepal\_width**: Sepal width, in centimeters, used as input.
* **petal\_length**: Petal length, in centimeters, used as input.
* **petal\_width**: Petal width, in centimeters, used as input.
* **class**: Iris Setosa, Versicolor, or Virginica, used as the target.

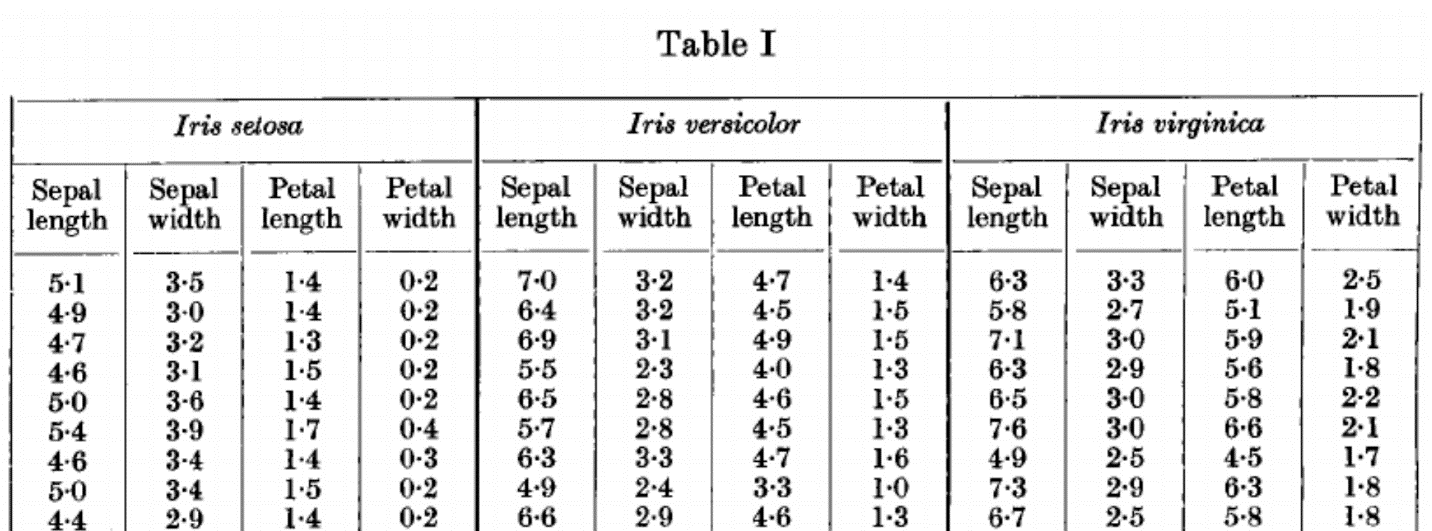


Figure 2. Morphological Measures of Iris Flowers

## Background and summary of related work

The Iris flower dataset is also known as the Fisher’s Iris dataset has been introduced by the British statistician and biologist in his 1936 paper. He collected the data to quantify the morphologic variation of iris flower. According to the article published by Pinto et. al. “Many methods have been introduced for iris dataset. Each and every method uses different strategy. This review consists of some prominent solution for iris species. Deeptam Dutta et. al. proposed a method on training Artificial Neural Networks. In this paper, IRIS bloom classification is done by utilizing Neural System. The issue concerns the recognizable proof of IRIS bloom species on the premise of bloom quality estimations. Characterization of IRIS informational collection would find designs from analyzing petal and sepal size of the IRIS blossom along with how the forecast was produced using breaking down the example to frame the class of IRIS bloom. Poojitha A et. al. reviewed collection of Iris flower using neural networks. Machine learning is subpart of the computer science. Existing iris bloom dataset is preloaded in MATLAB and is utilized for bunching into three unique species. The dataset is grouped utilizing the k implies calculation and neural system bunching instrument in MATLAB”. Viashali Arya et. al. giving most attention on effective neural fuzzy method for classification. Shashidhar T et. al. planned documentation of iris flower using classification. Patrick S. et. al. focused on statistical analysis of IRIS flower dataset.

# Placeholders for the in-progress items

* Update April/05/2021: Evaluate Some Algorithms
* Update April/07/2021: Training strategy
* Update April/10/2021: Variations of solutions
* Update April/14/2021: Building Model
* Update April/20/2021: Train and test on the same dataset
* Update April/24/2021: Compare the performance
* Update April/29/2021: Testing and conclusion
* Update May/01/2021: Lesson learned and drafting final report
* Update May/04/2021: Presentation preparation and recording

# 

# Methods

## Exploring methods

The goal of our method is to select the top classification model which achieves the best results on iris flower species. We can use different learning models which are created on three machine learning algorithms: K- Nearest Neighbor (KNN) classifier, Logistic Regression, Support Vector Machine (SVM). For addressing these approaches four iris dataset topographies are used in the train and test datasets. Those algorithms are applied with scikit-learn tool kit built on Python.

## Parametrized effect on performance

K-means clustering is a clustering algorithm that aims to find the local maximum. In this algorithm, the desired number of clusters must be selected first. Since there are three categories for the Iris dataset, the algorithm is programmed to put the data into three clusters by passing the n\_clusters parameter to the KMeans model. Now, three points (inputs) are randomly assigned to three categories. Based on the "centroid" distance between the two points, the next input is assigned to the corresponding cluster. The category centers are now re-selected for all clusters.

Each centripetal of a cluster is a set of attribute values ​​that determine the result groups. Centrifugal feature weights testing can be qualitatively used to interpret what type of iris each data set contains. In this regard, the KMeans model is imported from the sklearn library and predicts the results by fitting the features.

## Failing methods and rationale

According to figure 3, which is displaying the cluster result with three clusters but bad initialization. We can see that some of the samples change their class compare to the other result. We will get different cluster results whenever we are running a random initializing number. Therefore, a random initialization number is significant for an accurate cluster outcome. Nevertheless, we have no idea what possibly will be a decent initialization number. In this regard, by obtaining different machine learning systems, the researchers will select GA (Genetic Algorithm) to acquire the initialization point.



Figure 3. Clustering of Iris dataset with bad initialization

# Results

## Comparing the baseline solution and primary solution

We now have a basic idea about the data. We need to extend that with some visualizations. We are going to look at two types of plots:

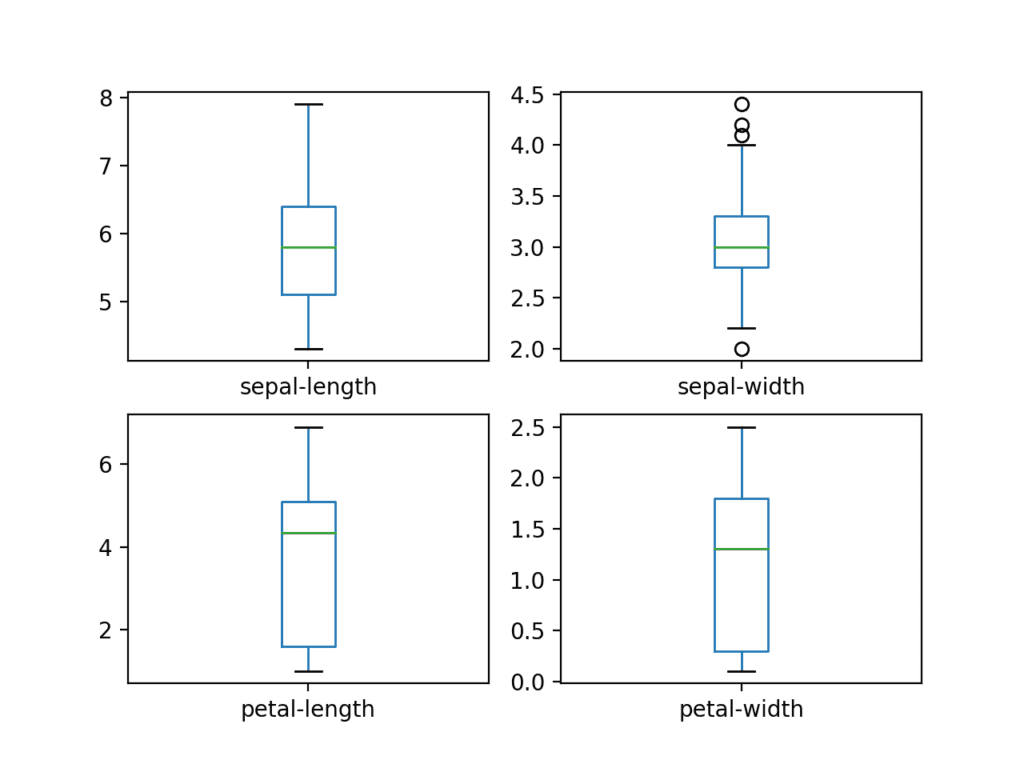
1. Univariate plots to better understand each attribute.
2. Multivariate plots to better understand the relationships between attributes.

### Univariate Plots

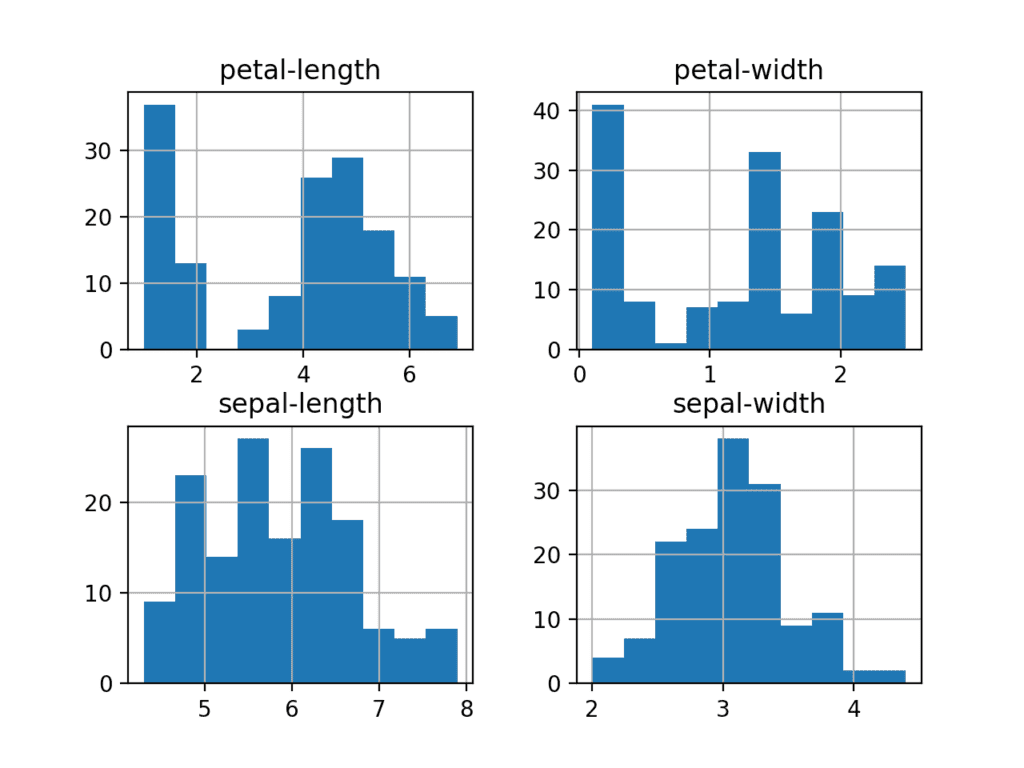
We start with some univariate plots, that is, plots of each individual variable.

Given that the input variables are numeric, we can create box and whisker plots of each.

|  |  |
| --- | --- |
|  | ...  # box and whisker plots  dataset.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False)  pyplot.show() |



We can also create a histogram of each input variable to get an idea of the distribution. It looks like perhaps two of the input variables have a Gaussian distribution. This is useful to note as we can use algorithms that can exploit this assumption.



### Multivariate Plots

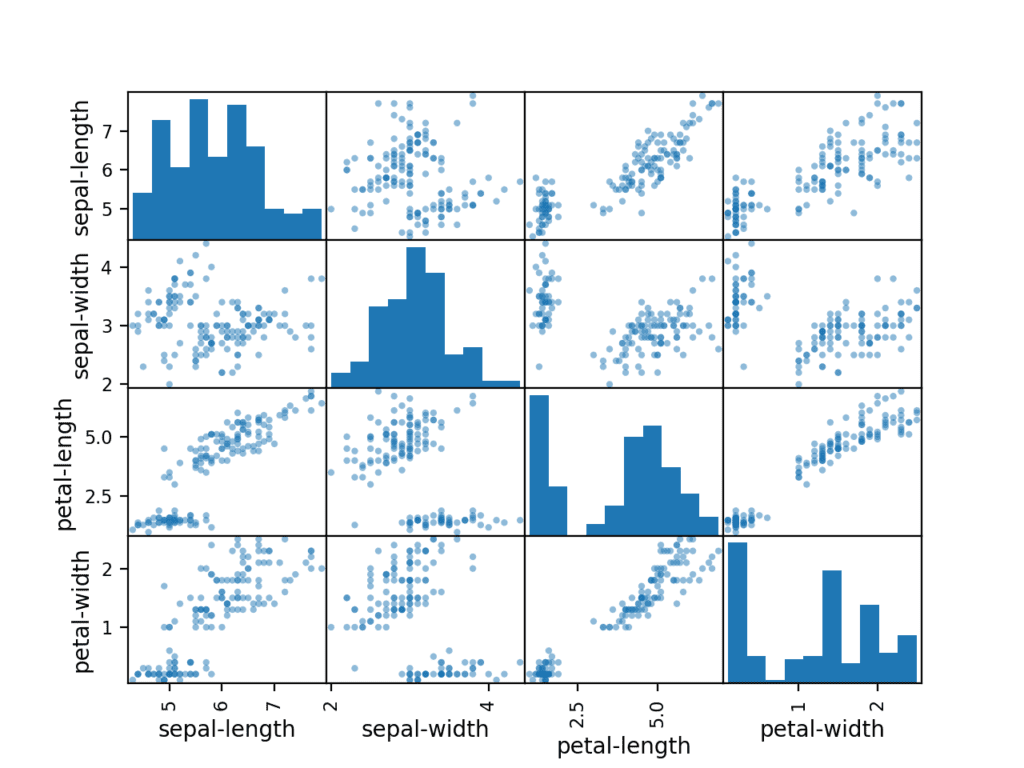
Now we can look at the interactions between the variables. First, let’s look at scatterplots of all pairs of attributes. This can be helpful to spot structured relationships between input variables.

# scatter plot matrix

scatter\_matrix(dataset)

pyplot.show()

Note the diagonal grouping of some pairs of attributes. This suggests a high correlation and a predictable relationship.



## Visualizations and graphs

For the unsupervised learning algorithm, four properties of the iris are given to the model and the model predicts to which category each instance belongs. The sklearn library in Python is used to load the Iris dataset and the matplotlib library to visualize the data. The following figure is used to explore and visualize the dataset by length and width.

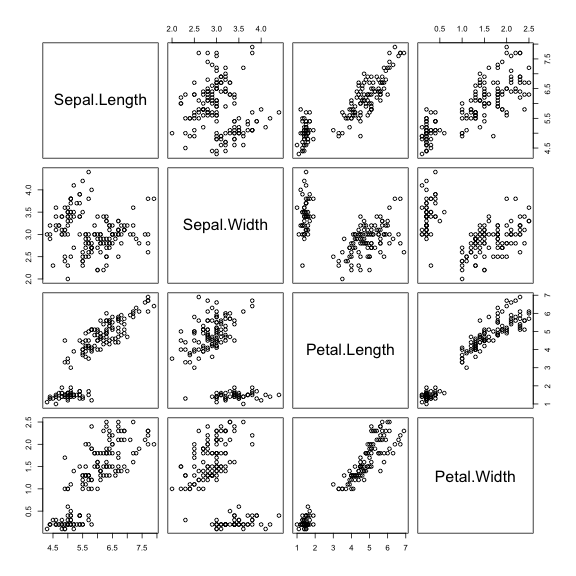


Figure 4 visual correlations

# Tools

## Describing the tools

Python is one of the best options for various data science application. We can combine the strength of multi-function programming language with running scripting languages like R or MATLAB. Although different languages such as C, Scala and R can be used in data mining, the best option for data mining is usually to use the Python programming language because its libraries are very useful. Python has a variety of libraries for working with data. For example, Scarpy is used to crawl the web and analyze existing data, or NumPy is used to build n-dimensional arrays and a variety of mathematical formulas such as Fourier transform or linear algebra functions. scikit-learn be contingent on two other Python packages, NumPy and SciPy.

## Employing the tools and their features

We have no idea which algorithms will be upright on this data structure or what outlines to practice. We yield a knowledge from the schemes that almost classes are partly linearly divisible in some scopes, so we are imagining usually accurate outcomes. We can analyze 6 different algorithms and they have different features.

* Logistic Regression (LR)
* Linear Discriminant Analysis (LDA)
* K-Nearest Neighbors (KNN).
* Classification and Regression Trees (CART).
* Gaussian Naive Bayes (NB).
* Support Vector Machines (SVM).

Following code will show and evaluate our models:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | ...  # 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())) |

In order to select the best model, we are supposed to compare the models to each other and choose the most effective one. Running the foregoing code, we would be able to produce the results:

LR: 0.951573 (0.023157)

LDA: 0.952974 (0.032941)

KNN: 0.956080 (0.032152)

CART: 0.946080 (0.032152)

NB: 0.937747 (0.045211)

SVM: 0.983974 (0.032083)

In this case, we would be able to scrutinize that it seems Support Vector Machines (SVM) has the principal appraised precision score at about 0.98 or 98%.

# Lesson Learned

## High-level summary

We have noticed that iris dataset comprises of two NumPy arrays: Scikit-learn which is responsible as X in scikit-learn and Y which is responsible for preferred outcome. The array X is a two-dimensional array of features, with one column per feature and one row per data point. We have divided dataset into a training set, in order to build a comprehensive model. According to desired model we can use test set to generalize new model for previously unseen data.