Iris Flower Classification Project

| **Name** | : Prateek Verma |
| --- | --- |
| **Registration Number** | : 12016050 |
| **Name** | : Tushar Vilekar |
| **Registration Number** | : 12010945 |
| **Programme** | : B. Tech - CSE |
| **Semester** | : Fifth |
| **Name of School** | : School of Computer Science And Engineering |
| **Date of Submission** | : 15th November 2022 |

**Submitted to**

**Md. Imran Hussain,**

**Department of Computer Science and Engineering,**

**LOVELY PROFESSIONAL UNIVERSITY,**

**PHAGWARA, PUNJAB**

****

# DECLARATION

**To whomsoever it may concern**

We, Prateek Verma (12016050) and Tushar Vilekar (1201045), hereby declare that the work done by us on “IRIS FLOWER CLASSIFICATION” is a record of original work towards the Project under the Subject “INT254: FUNDAMENTALS OF MACHINE LEARNING.”

| **Name** | **Registration Number** | **Signature** |
| --- | --- | --- |
| Prateek Verma | 12016050 |  |
| Tushar Vilekar | 12010945 |  |

# ACKNOWLEDGEMENT

We would like to thank our mentor and faculty, Md. Imran Hussain for his advice and input on this project and for guiding us in the right direction.

The allotment of the topic Iris Flower Classification for this project encouraged us to dive deeper into the world of machine learning and classification problems for various scenarios and study their advantages over each other in different applications.

# TABLE OF CONTENTS

[**DECLARATION**](#_g57h554gvkee) **2**

[**ACKNOWLEDGEMENT**](#_7rocvptjgdd) **3**

[**INTRODUCTION**](#_2x2xk8tgnl80) **5**

[**LIBRARIES**](#_wmje7t5gt823) **6**

[**TOOLS AND SOFTWARES**](#_4v83jyw83b63) **7**

[**PROPOSED MODULES**](#_91550diy9db0) **8**

[**DATA PROCESSING**](#_lvxc341k7dgw) **10**

[**MODEL TRAINING**](#_e8x1xlguuhfs) **14**

[**MODEL EVALUATION**](#_yqdm3ctwndlf) **15**

[**MODEL TRAINING SVM**](#_q1qizon5et1y) **16**

[**MODEL TESTING**](#_m6k85cnirf9w) **17**

[**CONCLUSION**](#_ohr58xfboj9t) **18**

[**BIBLIOGRAPHY**](#_ri18x2p0t05f) **19**

# INTRODUCTION

1. **Context**

This project has been done as partial fulfillment of the award of degree in Bachelor of Technology in Computer Science and Engineering at Lovely Professional University.

1. **Team Members**

**PRATEEK VERMA**

* Coding
* Reports
* Debugging

**TUSHAR VILEKAR**

* Coding
* Reports
* Debugging

# LIBRARIES

* **Pandas:** Pandas is a free library written for the Python programming language which is specifically intended for sorting and analyzing large data and helping break the large data down into humanly manageable chunks.
* **Matplotlib:** Matplotlib is a low-level graph plotting library in python that serves as a visualization utility. Matplotlib, created by John D. Hunter is an open-source library i.e, can be used for free for non-commercial purposes. Matplotlib is mostly written in python, and a few segments are written in C, Objective-C, and Javascript for Platform compatibility. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.
* **Numpy:** NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow use NumPy internally for the manipulation of Tensors.
* **Seaborn:** Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of the matplotlib library and is also closely integrated into the data structures from pandas.
* **Scikit-learn:** Unsupervised and supervised learning are both supported by the open-source machine learning package scikit-learn. Additionally, it offers a number of tools for data preparation, model selection, model assessment, and many other utilities.

# TOOLS AND SOFTWARES

* **Google Colaboratory:** <https://colab.research.google.com/drive/1nxp9QORyBP8VCvkJZaHiaSgQ-0xDRvta?usp=sharing>
* **Google Docs:** <https://docs.google.com/document/d/1l-_HIUQGLqTh62Pw2aZiVAAEonTrSKzN4hs8Zezzl1k/edit?usp=sharing>
* **Microsoft Visual Studio Code**

# PROPOSED MODULES

**Datasets:**

* Iris Data Set: Perhaps the most well-known database in the pattern recognition literature is this one. Fisher's study is still often cited today and is a classic in the subject. Three classes of 50 occurrences each are included in the data set, each class referring to a different kind of iris plant. Two classes cannot be linearly separated from one another, although one class can be linearly separated from the other two.

Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:

* Iris Setosa
* Iris Versicolour
* Iris Virginica

**Classification Algorithms Used:**

* Logistic Regression: A technique known as logistic regression is used to forecast a binary outcome: either something occurs or it does not. The expressions for this include Yes/No, Pass/Fail, Alive/Dead, etc.

The binary outcome, which falls into one of two groups, is determined by analyzing independent factors. The dependent variable is usually categorical, although the independent variables might be either category or quantitative.

* Naive Bayes: Naive Bayes determines the likelihood that a data point falls into a particular category or not. It is possible to classify words or phrases in text analysis as either falling under a predetermined "tag" (classification) or not.
* KNN: A pattern recognition technique known as "k-nearest neighbors" (k-NN) uses training datasets to identify the k nearest relatives among incoming instances.

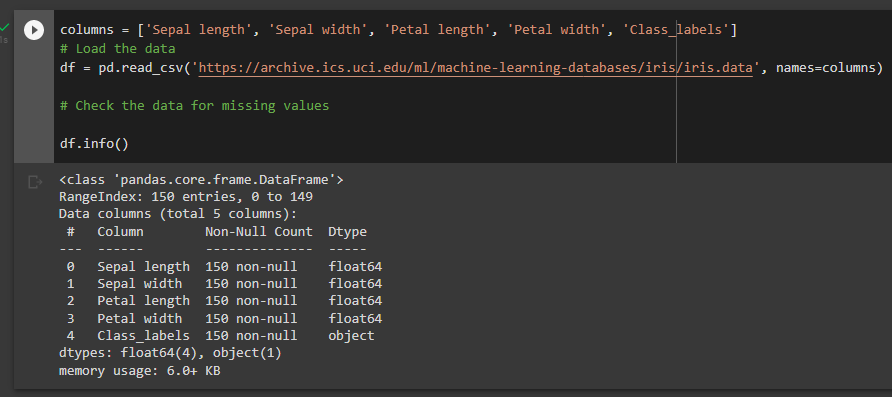
When using k-NN for classification, you figure out where to put the data according to its closest neighbor. If k = 1, it would be assigned to the class that is closest to 1. A majority survey of K's neighbors classifies it.

* Decision Tree: Because it can rank classes precisely, a decision tree is the ideal supervised learning technique for classification issues. Comparable to a flow chart, it divides data points into two similar categories at a time, starting with the "tree trunk" and moving through the "branches" and "leaves" until the categories are more closely related to one another. As a result, categories inside categories are created, enabling organic categorization with minimal human oversight.
* Support Vector Machines: A support vector machine (SVM) goes beyond X/Y prediction by using methods to train and classify data according to degrees of polarity.

# DATA PROCESSING

**Missing Data**

As claimed by the dataset provider, there are no missing values and this can be verified using pandas functions.



If the data had any corrupted or missing values use either of the two methods:

1. Deleting the Missing values
2. Imputing the Missing Values

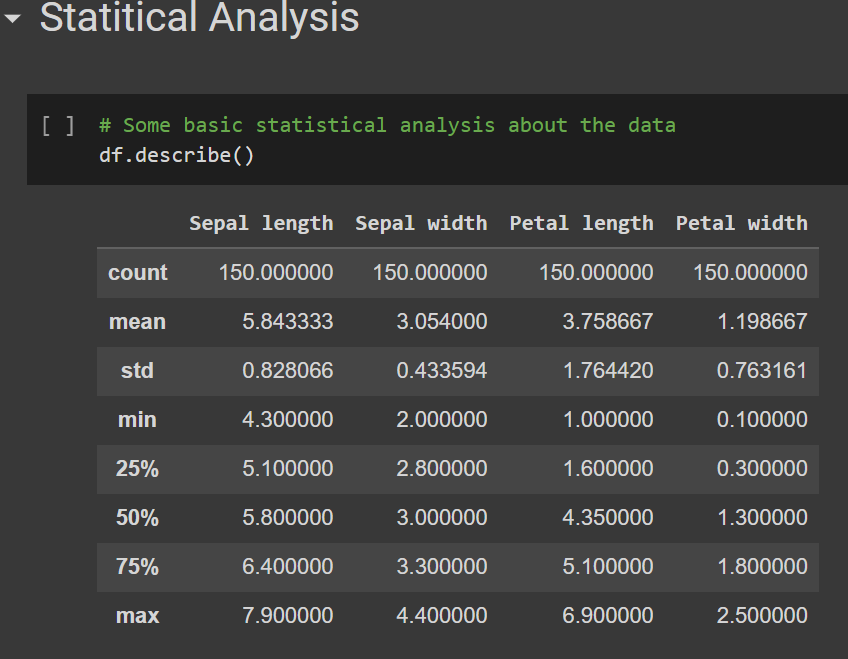
If the missing value is of the type Missing Not At Random (MNAR), then it should not be deleted.

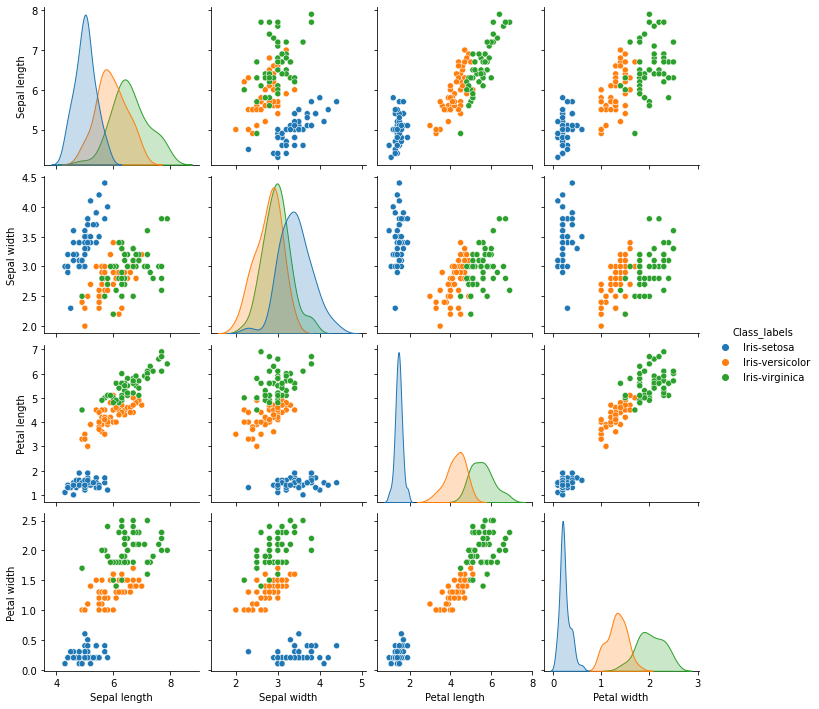
If the missing value is of type Missing At Random (MAR) or Missing Completely At Random (MCAR) then it can be deleted.

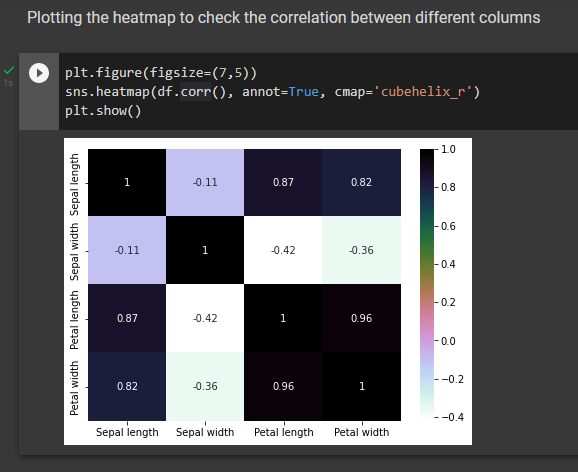
**Statistical Analysis**

df.describe() returns the statistical analysis of each column in the data frame (df). As we can see from the screenshot below, it returns a count of exactly 150 and parameters such as mean, standard deviation, minimum and maximum values, and 25, 50, and 75th percentile of the columns.

Using this data, we can further visualize the dataset using seaborn (sns.pairplot()) based on the class labels.





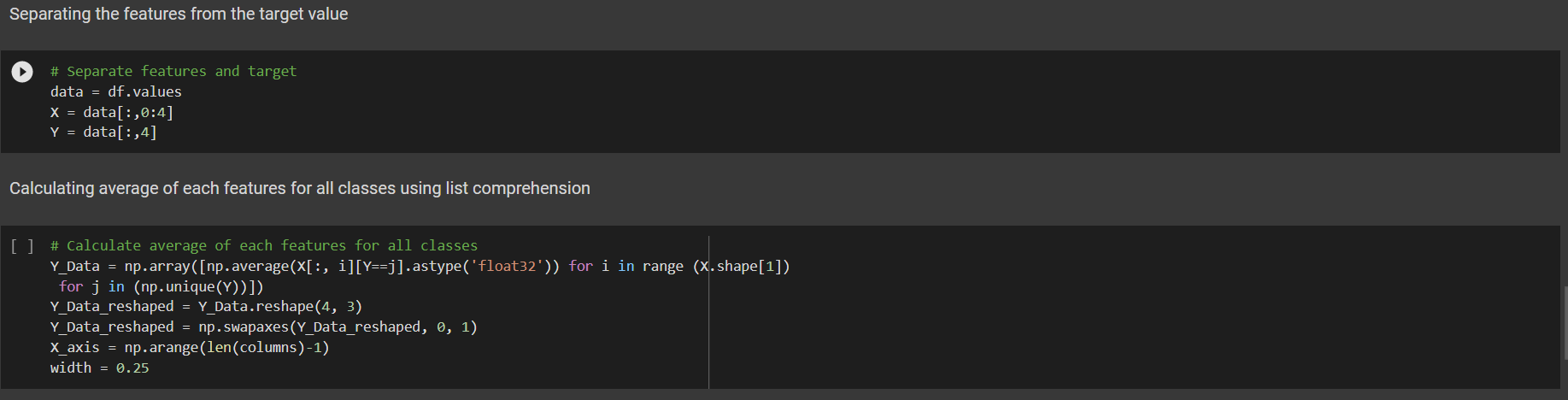


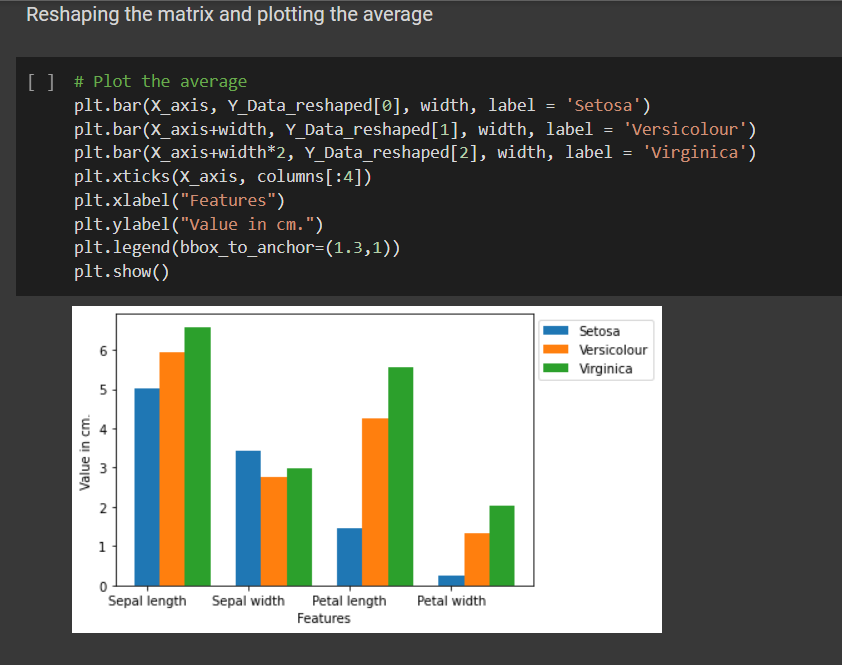
**Observations**

* As we can see from the pair plots above that iris-setosa can be classified based on multiple parameters.
* Iris virginica is the longest flower and iris setosa is the shortest flower.
* Petal length and Petal width have a high correlation (0.96).

**Separating features and targets**

* Calculating the average for each feature for all classes
* Plotting the data





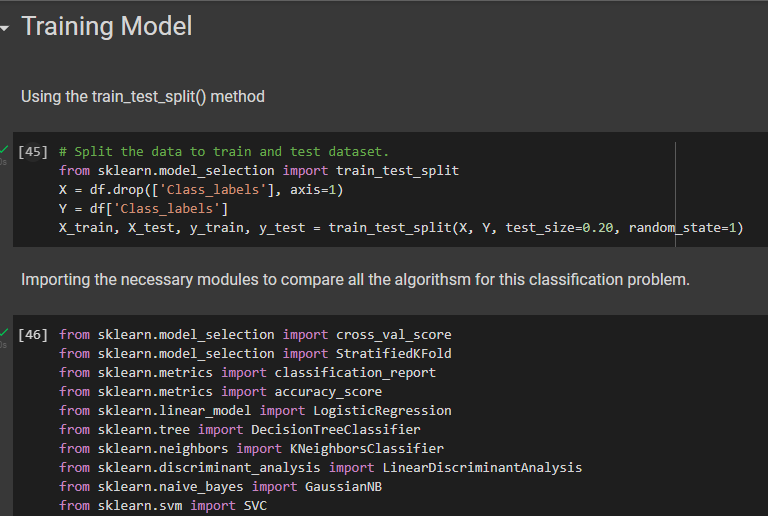
# MODEL TRAINING

**Assumption:**

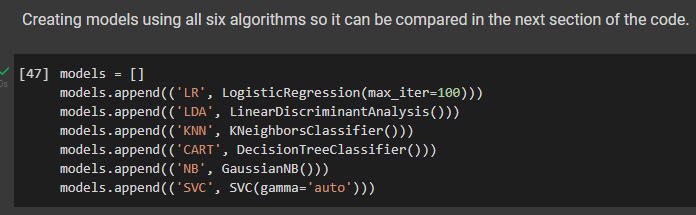
**Support Vector Machine will work best for this case.**

Since the data can be segregated by drawing a line (in the pairwise plots). SVM is the best approach for this since the data set is of the order of a hundred - two hundred and this is a classification problem. SVM best suits this problem and will be verified by trying out the same using all the previously listed methods.

The model is trained using the train\_test\_split() method followed by the fit() method.

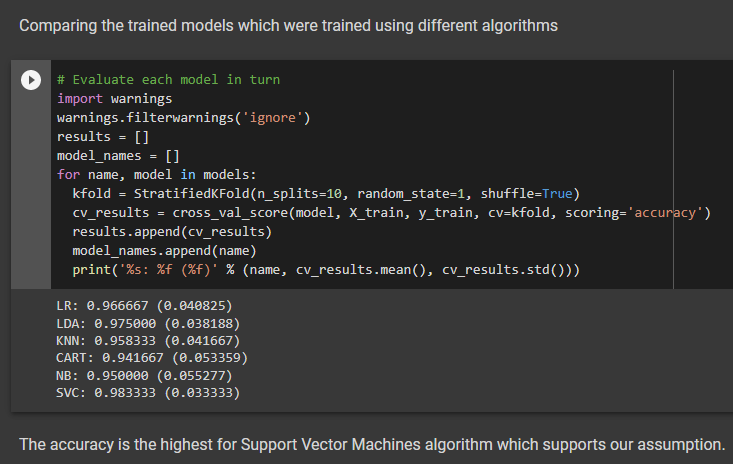


**Comparison**



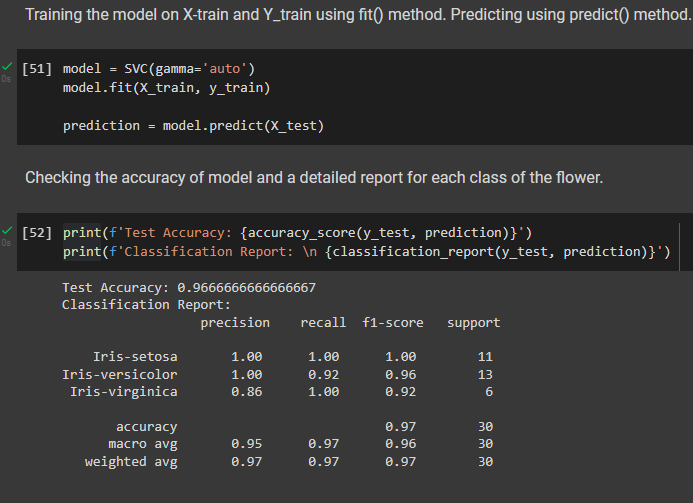
# MODEL EVALUATION

After the models are trained we evaluated the models of each algorithm using the StratifiedKFold() method with splits as 10 and random\_state=1.



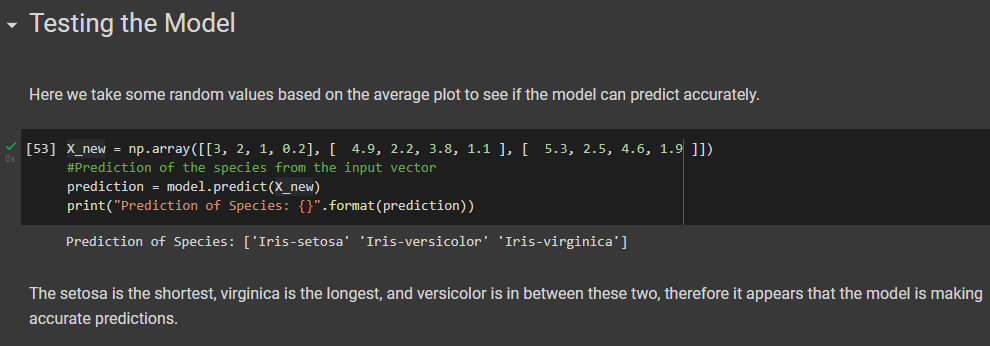
# MODEL TRAINING SVM

We tested the model by giving it some random values and based on the average plot, the results that we expected matched with the obtained result.



# MODEL TESTING

We tested the model by giving it a random value to predict and it predicted accurately.



# CONCLUSION

After validating our assumption that the Support Vector Machines algorithm will work best for the Iris Classification problem, we can successfully conclude the same and save the model in pickle format for future use cases.

# 

# BIBLIOGRAPHY

**Softwares**

* Google Collaboratory
* Google Docs
* Microsoft Word
* Visual Studio Code

**Libraries**

* Pandas (<https://pandas.pydata.org/>)
* Matplotlib (<https://matplotlib.org/>)
* Scikit-learn (<https://scikit-learn.org/>)
* seaborn (<https://seaborn.pydata.org/>)
* NumPy (<https://numpy.org/>)

**References**

* Stackoverflow
* Documentations of libraries
* Kaggle
* javaTpoint

**Datasets**

* Iris Dataset Link: <https://archive.ics.uci.edu/ml/datasets/iris>