Iris Flower Classification Using Machine Learning

Project source code:

https://colab.research.google.com/drive/1_5wadeK0QDBYfMhJo8bc3OtGjtKTC1Ml

somagani sai chandra

Department of computer science and engineering

Amrita Vishwa Vidyapeetham,Chennai campus..

https://www.linkedin.com/in/sai-chandra-somagani-59944a217/

ch.en.u4cse20167@ch.students.amrita.edu

DR.sangapu sreenivasa chakravarthi

Department of computer science engineering

ss.chakravarthi@ch.amrita.edu

ABSTRACT

Classification is a supervised machine learning approach which is used to predict group membership for data instances. Neural networks are being developed to simplify the difficulty of classification.This model focuses on Iris flower categorization using Neural Network. Utilizing the Scikit Learn Tool Kit will simplify classification. This project primarily focuses on utilising Scikit Learn to classify datasets. The issue is the identification of Iris flower species (setosa, versicolor, and verginica) based on the measures of the sepal and petal length and width. The iris flower dataset may be used to train a variety of machine learning algorithms to create classification models, and we can then select the model with the highest accuracy to more accurately predict the species of iris flower. Identifying patterns from the Iris data set's classification would includeevaluating the Iris flower's sepal and petal sizes and how the classification of Iris flowers was formed based on an analysis of the pattern. This pattern and categorization can be used to more accurately anticipate future years' unobserved data. The application of artificial neural networks to issues with pattern categorization, function approximation, optimization, and associative memories has proved fruitful. The intention is to simulate the probabilities of class membership, depending on the characteristics of the flower. In order to forecast the species of iris flower by input of the unseen data using what it has learned from the trained data, we will train our model with data using machine learning in this project.

INTRODUCTION

In order to train and anticipate an outcome using algorithms, a machine must be fed with sufficient data. The machine will grow more effective the more processed or usable data it receives. It learns the data and creates the prediction model when the data is complex. It is said that the more data, the better the model, and the greater the accuracy. Machine learning can be done in many different methods, including supervised learning, unsupervised learning, and reinforcement learning.

In order to train and anticipate an outcome using algorithms, a machine must be fed with sufficient data. The machine will grow more effective the more processed or usable data it receives. It learns the data and creates the prediction model when the data is complex. It is said that the more data, the better the model, and the greater the accuracy. Machine learning can be done in many different methods, including supervised learning, unsupervised learning, and reinforcement learning.

One of the main data mining techniques is classification, which divides data into predetermined categories. Since the classes are established prior to looking at the data, it falls under the category of supervised learning. It is necessary to have some understanding of the data in order to apply all categorization techniques. Usually, data knowledge aids in the discovery of previously unidentified patterns. Building a function that outputs two or more classes from the given feature is the goal of pattern classification.

The goal of mining the iris data set is to find patterns by analysing the iris plant's sepal and petal sizes, and to determine the class of iris plants by studying the patterns. Future years will allow for the individual identification of different blooms utilising classification and pattern recognition. Unmistakably, it is said that a classification model would be the kind of relationship that is being mined from the iris information.

LITERATURE SURVEY

I previously read some research papers

- Zainab Iqbal used gaussian naïve bayes to classify iris flower.he created scatter matrix and scatter plot it give uh detailed analysis of iris dataset.he used this with python and he achived 95% accuracy and this is efficient for supervised classification.

- Joylin Priya used KNN,SVM and logistic regression.she applied cross validationto maximise accuracy.she found that svm has highest accuracy

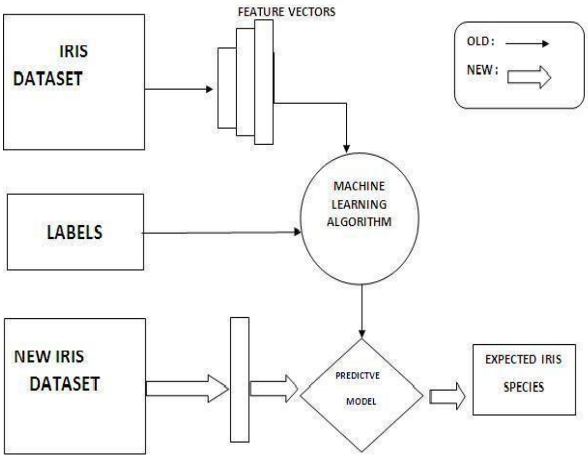
- Rathee showed us feed forward neural system which is based on features of iris data which gave him 98.3% accuracy and when he used multilayer perceptionon iris data he got 98.82% accuracy.

- Borovinskiy used three neural systems procedures by making them connected.base model and neural system is 98% .he then used coordinated grouping and characterization showed 98.66% prescion.

→not only these many people researched aboud iris dataset and implementing in different type of models like svm,naïve bayes,knn,…so on .different strategies used in every study.the only problem statement is classifying and recognition of iris flower species based on its features.

In my method I classify the dataset by determining patterns after knowing about features of iris flowers and then I predict the processing of the patterns to form the iris flower class.

BLOCK DIAGRAM



➔ First we need to collect the iris data set .the one which I downloaded is iris.csv .there are three samples of flowers that are iris setosa,iris versicolour,iris virgincia.

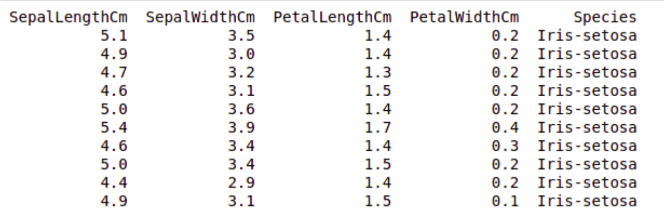
➔ Then we need to preprocess the data then visualize of data,then we need to train the iris data , then we to evaluate the data and at last we need to test the data.

DATASET

➔ The Iris flower data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper.

➔ The data set includes 50 samples from each of the three Iris species (Iris Setosa, Iris virginica, and Iris versicolor). From each sample, the length and width of the sepals and petals, both in centimetres, were measured.

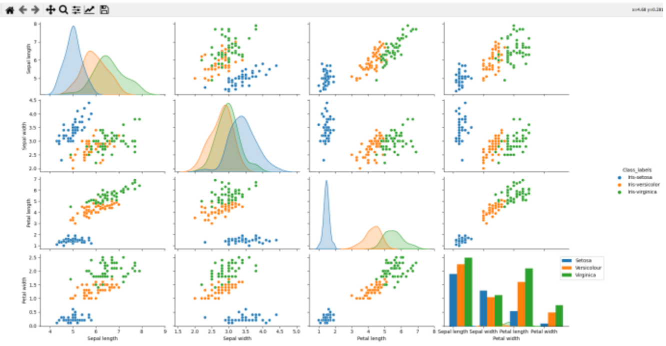
➔ This dataset became a typical test case for many statistical classification techniques in machine learning such as support vector machines.



```
SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
          5.1           3.5            1.4           0.2  Iris-setosa
          4.9           3.0            1.4           0.2  Iris-setosa
          4.7           3.2            1.3           0.2  Iris-setosa
          4.6           3.1            1.5           0.2  Iris-setosa
          5.0           3.6            1.4           0.2  Iris-setosa
          5.4           3.9            1.7           0.4  Iris-setosa
          4.6           3.4            1.4           0.3  Iris-setosa
          5.0           3.4            1.5           0.2  Iris-setosa
          4.4           2.9            1.4           0.2  Iris-setosa
          4.9           3.1            1.5           0.1  Iris-setosa
```

- The collection includes 150 records with the following 5 attributes: class, petal length, petal width, sepal length, and sepal width (Species).
- There are 4 features in iris flower classification are sepal length, sepal width, petal length, petal width.

VISUALIZATION



MODEL USED→ SUPPORT VECTOR MACHINE

A supervised machine learning algorithm called a support vector machine, commonly referred to as a support vector network, analyses data for regression and classification. SVMs are among the most reliable methods for classifying data. Support vector machines (SVMs) are huge yet adaptable supervised machine learning techniques used for outlier detection, regression, and classification. SVMs are frequently employed in classification issues and are extremely effective in large dimensional spaces. Because they only use a portion of the training points in the decision function, SVMs are well-liked and memory-efficient algorithms.

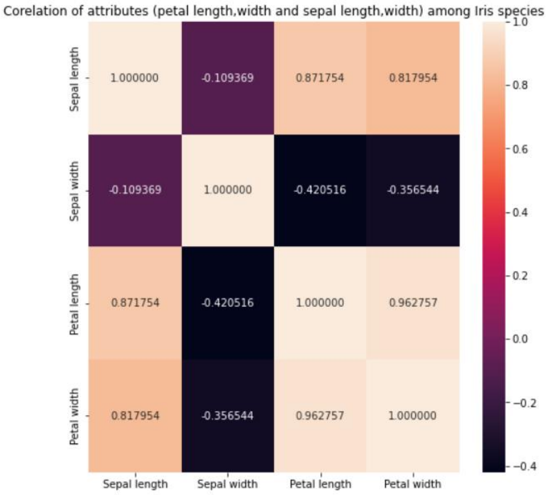
It uses libsvm as the foundation for its C-support vector classification implementation. Scikit-learn makes use of the sklearn.svm.SVC module. This class manages the one-vs-one system for multiclass support.

CORRELATION MATRIX

A correlation matrix is simply a table which displays the correlation coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values in a table. It is a powerful tool to summarize a large dataset and to identify and visualize patterns in the given data.

A correlation matrix consists of rows and columns that show the variables. Each cell in a table contains the correlation coefficient.

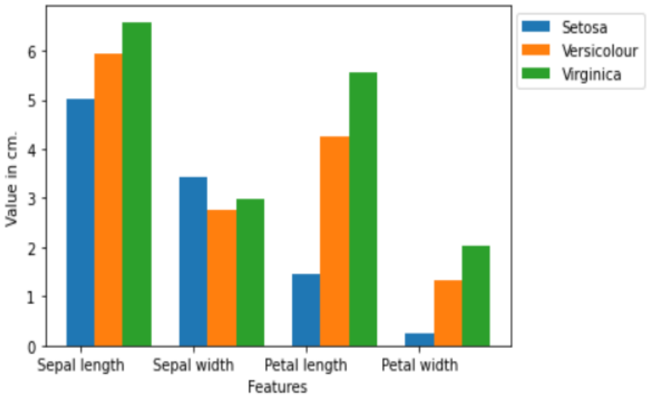
In addition, the correlation matrix is frequently utilized in conjunction with other types of statistical analysis. For instance, it may be helpful in the analysis of multiple linear regression models. Remember that the models contain several independent variables. In multiple linear regression, the correlation matrix determines the correlation coefficients between the independent variables in a model.



Features

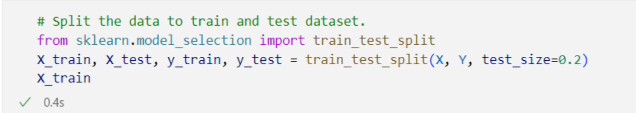
The iris dataset contains three classes of flowers, Versicolor, Setosa, Virginica, and each class contains 4 features, 'Sepal length', 'Sepal width', 'Petal length', 'Petal width'. The aim of the iris flower classification is to predict flowers based on their specific features.

- **Sepal length**
- **Sepal width**
- **Petal length**
- **Petal width**



TRAINING DATA

When we will understand what the dataset is about, we can start training our model based on the algorithms. First, we have to train our model with some of the samples. Here, we will be using scikit-learn library method called ‘train_test_split’ which divides our data set into a ratio of 80:20, in which 80% data will be using for training and 20% data will be using for testing. This process can be done by the following code:



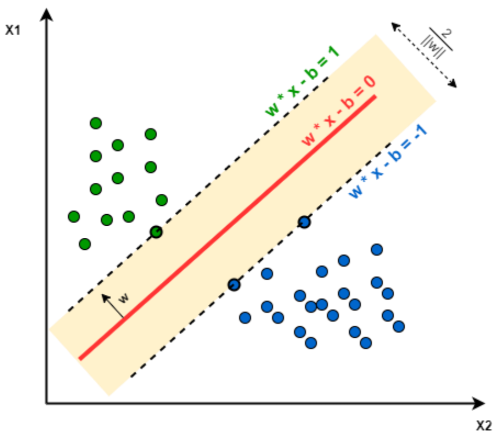
```
# Split the data to train and test dataset.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
X_train
```

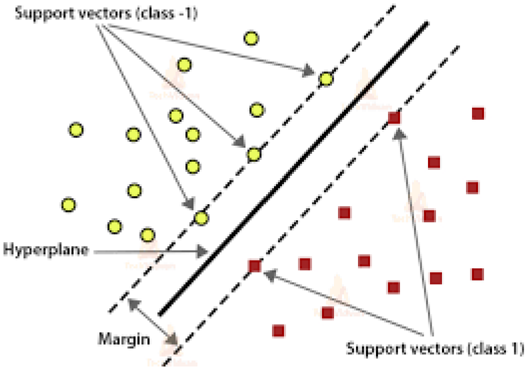
✓ 0.4s

SVM method.

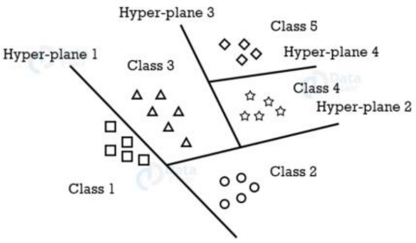
A support vector machine abbreviated as SVM was first introduced in the 1960’s and late an improvised in the 1990’s. SVM is supervised learning machine learning classification algorithm that has become extremely popular nowadays owing to its extremely efficient results so SVM is implemented in a slightly differently than other machine learning algorithms it is capable of performing classification and regression and outlier detection as well.

Svm algorithm first finds the points which are close to the line from both classes.and these points are called as support vectors which added by a base vector.what it actually does is it find the distance between the line and support vectors.and this distance is called margin.in svm the main goal is to maximize the margin and also known as optimal hyperplane.





Svm is mainly used for binary classification.but for multiclass classification,it seperates the data for binary classification and it utilizes the same data by breaking down multi-classification problems into multiple binary classification problems.



The main algorithm of svm is to divide dataset into number of classes to find maximum marginal hyperplane.

→support vector machine first generate hyperplane iteratively seperates class in best way.

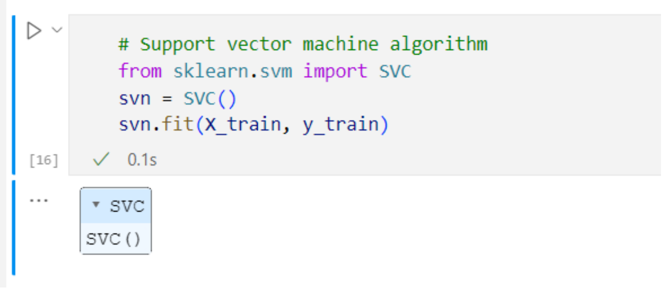
→ After that it will choose the hyperplane that segregate the classes correctly.

We used svm in scikit-learn .it provides three classes which are

- SVC
- NuSVC
- LinearSVC

These three classes perform **multi class classification.**

In my project I used two classes which is SVC, LinearSVC i.e support vector classifier, linear support vector classifier



```
# Support vector machine algorithm
from sklearn.svm import SVC
svn = SVC()
svn.fit(X_train, y_train)
```

[16] ✓ 0.1s

```
SVC
SVC()
```

SVC CLASS

- I have used two methods in support vector classifier which are
  1. **Fit**
  2. **Predict**

LinearSVC CLASS

- I have used three methods in linear svc class that are
  1. **Fit**
  2. **Predict**
  3. **Score**

SVC CLASS

SVC.FIT()

→PARAMETRES PASSED ARE X_train and y_train.

X: {array-like, sparse matrix} of shape (n_samples, n_features) or (n_samples, n_samples).Training vectors, where n_samples is the number of samples and n_features is the number of features. For kernel=”precomputed”, the expected shape of X is (n_samples, n_samples).

Y array-like of shape (n_samples,)
Target values (class labels in classification, real numbers in regression).

(n_samples,n_outputs) True labels for X.
sample_weightarray-like of shape (n_samples,),

sample_weightarray-like of shape (n_samples,), default=None
Per-sample weights. Rescale C per sample. Higher weights force the classifier to put more emphasis on these points.

SVC.PREDICT

Parameters:→ passed are X_test(last 30 rows)
X{array-like, sparse matrix} of shape (n_samples, n_features) or (n_samples_test, n_samples_train)
For kernel=”precomputed”, the expected shape of X is (n_samples_test, n_samples_train).

Linearsvc()

Fit method linearsvc.fit

Parameters
X{array-like, sparse matrix} of shape (n_samples, n_features)
Training vector, where n_samples is the number of samples and n_features is the number of features.
yarray-like of shape (n_samples,)
Target vector relative to X.
sample_weightarray-like of shape (n_samples,), default=None
Array of weights that are assigned to individual samples. If not provided, then each sample is given unit weight.

Linearsvc.predict

Parameters(X_train)

X{array-like, sparse matrix} of shape (n_samples, n_features)
The data matrix for which we want to get the predictions.

Third method linearsvc.score

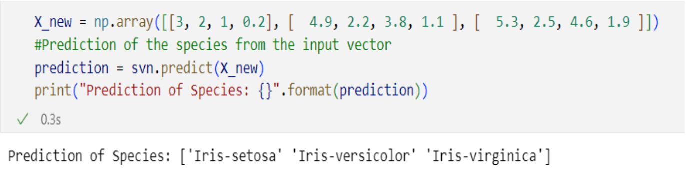
Parameters
X array-like of shape (n_samples, n_features) test samples.

Y array -like of shape (n_samples,) or

default=None Sample weights.

Testing the model

Here we take some random values based on the average plot to see if the model can predict accurately.



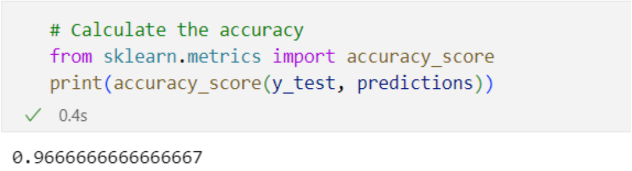
```
X_new = np.array([[3, 2, 1, 0.2], [  4.9, 2.2, 3.8, 1.1 ], [  5.3, 2.5, 4.6, 1.9 ]])
#Prediction of the species from the input vector
prediction = svm.predict(X_new)
print("Prediction of Species: {}".format(prediction))
```

✓ 0.3s

```
Prediction of Species: ['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']
```

I gave the data od three new iris species and I found the 1 spiece is iris setosa ,the second one is iris versicolor
Third oneis iris virgincia.

ACCURACY
96.67%



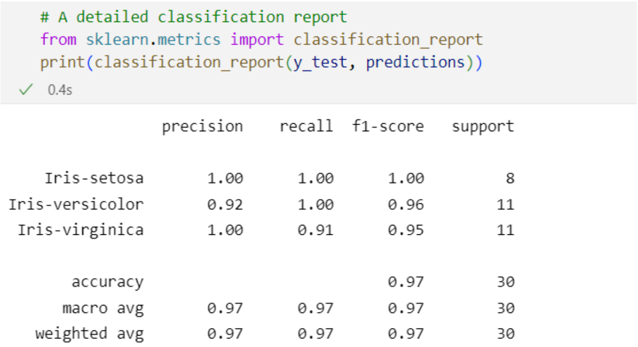
```
# Calculate the accuracy
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, predictions))
```

✓ 0.4s

```
0.9666666666666667
```

I have used svm method to classify iris flowers and I got the accuracy of 96.67%.

CLASSIFICATION REPORT



```
# A detailed classification report
from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))
```

✓ 0.4s

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00         8
Iris-versicolor       0.92      1.00      0.96        11
 Iris-virginica       1.00      0.91      0.95        11

       accuracy                           0.97        30
      macro avg       0.97      0.97      0.97        30
   weighted avg       0.97      0.97      0.97        30
```

CONCLUSION

In this project, I learned to train our own supervised machine learning model using Iris Flower Classification Project with Machine Learning. Through this project, we learned about machine learning, data analysis, data visualization, model creation, etc. The primary goal of supervised learning is to build a model that “generalizes”. Here in this project we make predictions on unseen data which is the data not used to train the model hence the machine learning model built should accurately predicts the species of future flowers rather than accurately predicting the label of starting index.

REFERENCES

https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

https://corporatefinanceinstitute.com/resources/excel/correlation-matrix/

https://www.tutorialspoint.com/scikit_learn/scikit_learn_support_vector_machines.htm

https://vitalflux.com/svm-classifier-scikit-learn-code-examples/

https://www.researchgate.net/publication/221918432_Iris_Recognition_System_Using_Support_Vector_Machines/figures?lo=1