CS-513-A Knowledge Discovery and Data Mining

Multi-classification and visualization of an iris plant based on four features

Final Project Report

Shlok Arun Khetan
1046485

Name: Shlok Khetan
CWID: 10446585

Contents

Name: Shlok Khetan
CWID: 10446585

Motivation

Knowledge Discovery and data mining has been a very interesting and insightful subject which has taught me the fundamentals of data mining and the R language. This is the report of my final project in which I have taken an iris dataset and used two algorithms, ANN(Artificial Neural Network) and RF(Random Forest) to classify them into three categories (***Iris setosa*, *Iris virginica* or *Iris versicolor*).**

Platform and language used:

Language used : Python 3.7

Platform used: Google collab notebook, jupyter notebook

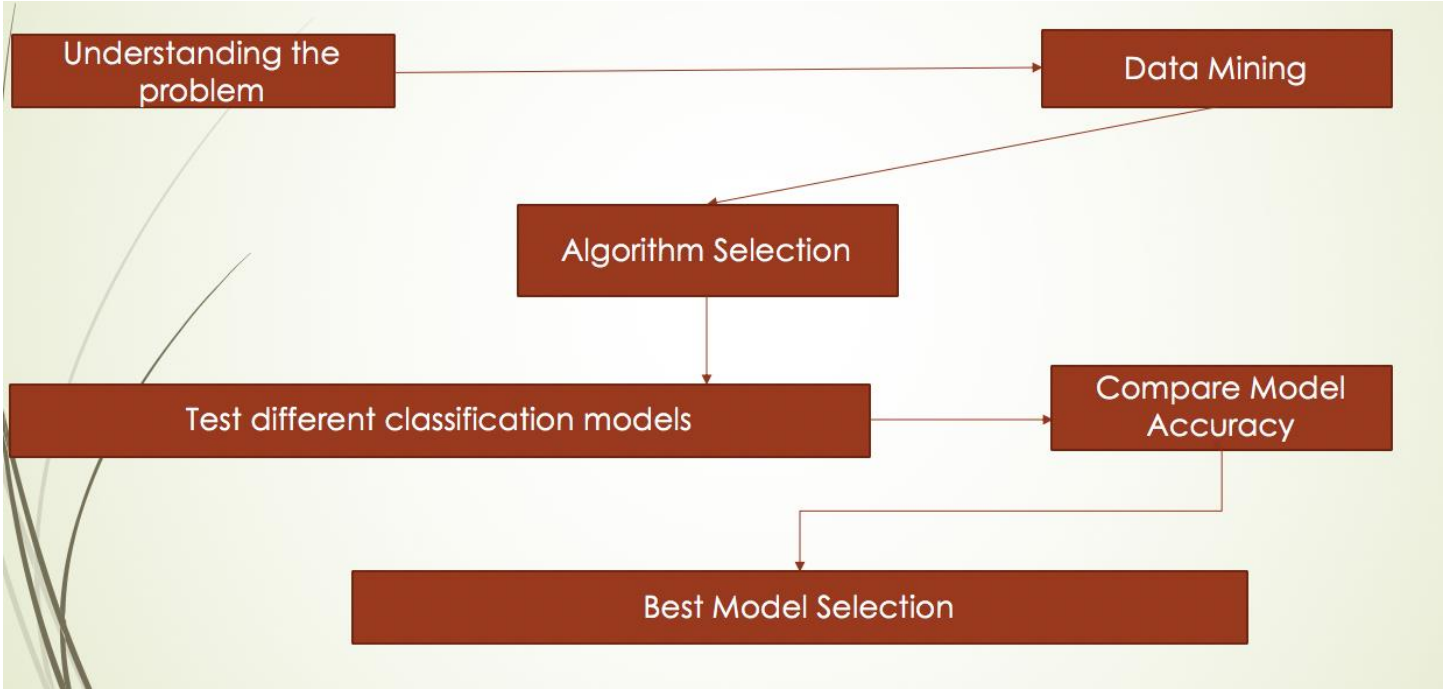
Goal:

My goal is to develop a model capable of classifying an iris plant based on four features. This is a multi-class classification where each sample can belong to ONE of 3 classes (*Iris setosa*, *Iris virginica* or *Iris versicolor*). The artificial neural network(ANN) will have 4 input neurons (flower dimensions) and 3 output neurons (scores). my loss function will compare the target label (ground truth) to the corresponding output score.

The second would be to also analyze and predict the tensor with the RF(Random Forest classifier)

Name: Shlok Khetan
CWID: 10446585

Road Map:



1)Understanding the Problem Statement: The goal is to classify the flowers based on input flower dimensions.

2)Data Mining: We discover patterns in our dataset and try to devise and decide on our algorithms to reach our goal.

3)Algorithm Selection: After much deliberation, I decided on using two algorithms ANN and Random Forest because of their superior classification capabilities.

4)Test different classification models: Both the algorithms are implemented using python data fuctions importing the respective nn ad RF classifiers.

5)Compare Model Accuracy: After the model is implemented we compare the accuracy which is the percentage of times our model has been able to predict the correct answer in a given situation.

Name: Shlok Khetan
CWID: 10446585

6)Best Model Selection: The model with the best accuracy percentage is selected, which in my case was the Artificial Neural Network with an accuracy of 96.67% as compared to Random Forest which is 95.55%.

Data Understanding:

NOTE: Multi-class classifications usually involve converting the target vector to a one_hot encoded matrix. That is, if 5 labels show up as

```
tensor([0,2,1,0,1])
```

then we would encode them as:

```
tensor([[1, 0, 0],
        [0, 0, 1],
        [0, 1, 0],
        [1, 0, 0],
        [0, 1, 0]])
```

This is easily accomplished with **`torch.nn.functional.one_hot()`**.
However, my loss function **`torch.nn.CrossEntropyLoss()`** takes care of this for us.

The Iris flower data set or Fisher's Iris data set is a multivariate data set presented by the British analyst and scientist Ronald Fisher in his 1936 paper The utilization of numerous estimations in ordered issues for instance of direct discriminant analysis. It is here and there called Anderson's Iris data set since Edgar Anderson gathered the data to evaluate the morphologic variety of Iris flowers of three related species. Two of the three species were gathered in the Gaspé Peninsula "all from a similar field, and singled out that day and estimated simultaneously by a similar individual with a similar mechanical assembly".

The data set comprises of 50 examples from every one of three types of (Iris setosa, Iris virginica and Iris versicolor). Four highlights were estimated from each example: the length and the width of the

sepals and petals, in centimetres. In view of the mix of these four highlights, Fisher built up a straight discriminant model to separate the species from one another.

Loading the iris data set

I have loaded the iris dataset through a link and using the function

```
#Load dataset
iris = datasets.load_iris()
```

which gives a snapshot

	sepal length	sepal width	petal length	petal width	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

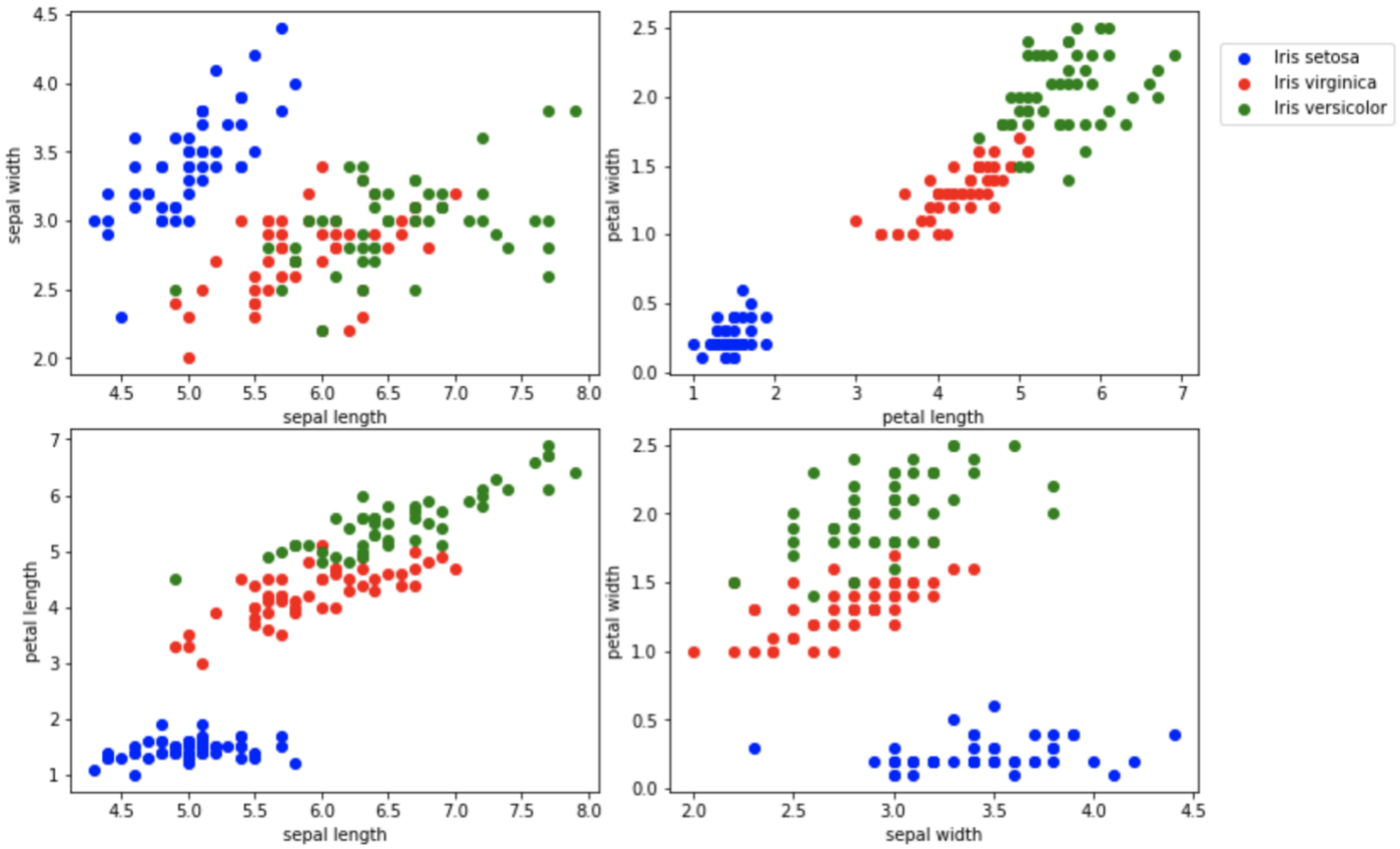
Name: Shlok Khetan
CWID: 10446585

Plotting the dataset

The iris dataset has 4 features. To get an idea how they correlate we can plot four different relationships among them.

We'll use the index positions of the columns to grab their names in pairs with plots = [(0,1),(2,3),(0,2),(1,3)].

Here (0,1) sets "sepal length (cm)" as x and "sepal width (cm)" as y.



Name: Shlok Khetan
CWID: 10446585

Artificial Neural Network (A.N.N.)

In Pattern recognition and statistical estimation, we use ANN for multi-class classification.

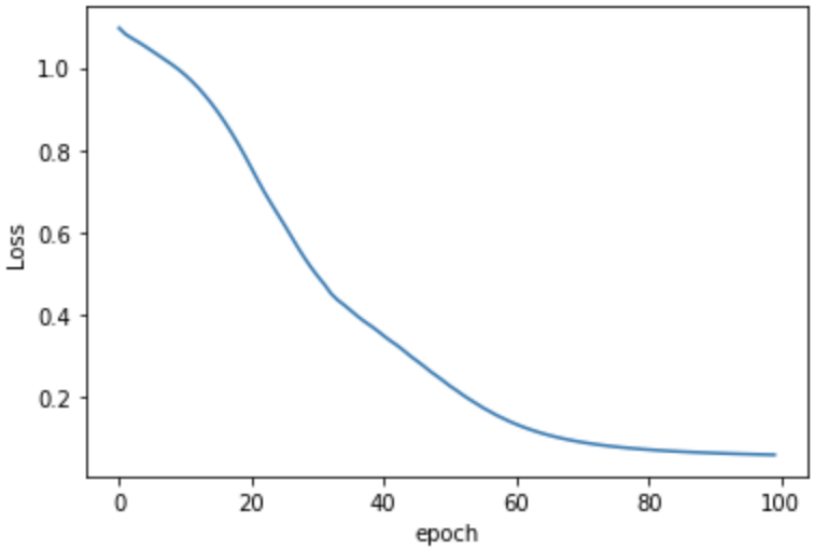
Artificial Neural Networks or ANN is a data processing algorithm that is motivated by the way the organic sensory system, for example, cerebrum process data. It is made out of huge number of exceptionally interconnected handling elements(neurons) working as one to take care of a particular issue

I have set a function to analyse the accuracy and also plot the loss function.

The loss function

```
plt.plot(range(epochs), losses)
plt.ylabel('Loss')
plt.xlabel('epoch');
```

works through epochs where we observe that as the number of epochs increase the loss decreases



Name: Shlok Khetan
CWID: 10446585

Accuracy function – I have taken a test case of 30 iterations where it was unale to predict a single case incorrectly

```
 1. tensor([-0.3405,  7.3566,  1.3707])     1
 2. tensor([0.2747, 8.1488, 0.4138])        1
 3. tensor([ 11.9818,   6.1788, -19.1899])  0
 4. tensor([-2.0217,  7.9597,  4.2340])     1
 5. tensor([-6.1397,  7.9443, 11.0864])     2
 6. tensor([-10.2711,   8.3000,  18.0008])  2
 7. tensor([ 12.0381,   6.4263, -19.2827])  0
 8. tensor([ 12.9323,   6.4755, -20.7418])  0
 9. tensor([-5.7778,  8.2352, 10.5033])     2
10. tensor([-7.8917,  8.6034, 14.0678])     2
11. tensor([-8.7110,  8.5980, 15.4298])     2
12. tensor([ 11.6191,   5.8112, -18.6104])  0
13. tensor([-8.1081,  8.2234, 14.3879])     2
14. tensor([-2.0836,  7.7684,  4.3107])     1
15. tensor([-6.0867,  8.3830, 11.0509])     2
16. tensor([0.1330, 7.8598, 0.6288])        1
17. tensor([-4.0930,  7.7140,  7.6592])     2
18. tensor([ 13.1338,   6.5848, -21.0678])  0
19. tensor([-1.5696,  8.0146,  3.4673])     1
20. tensor([-6.2900,  8.9642, 11.4163])     2
21. tensor([ 12.3687,   6.2517, -19.8166])  0
22. tensor([ 13.8037,   7.0790, -22.1457])  0
23. tensor([-8.8527,  8.3090, 15.6448])     2
24. tensor([ 12.1811,   6.1210, -19.5152])  0
25. tensor([-5.8117,  7.5402, 10.5277])     2
26. tensor([-4.4569,  7.7801,  8.2799])     2
27. tensor([-1.4331,  7.7717,  3.2245])     1
28. tensor([ 0.5324,  7.5300, -0.0604])     1
29. tensor([-5.8277,  8.1491, 10.5915])     2
30. tensor([-5.2631,  7.7396,  9.6074])     2

29 out of 30 = 96.67% correct
```

Giving it an accuracy of 96.67%

Name: Shlok Khetan
CWID: 10446585

Random Forest Classifier:

To start with, Random Forest calculation is a directed grouping calculation. We can see it from its name, which is to make a forest by some way and make it arbitrary. There is an immediate connection between the quantity of trees in the forest and the outcomes it can get: the bigger the quantity of trees, the more precise the outcome. In any case, one thing to note is that making the forest isn't equivalent to developing the choice with data increase or addition file approach

The steps followed in implementing this were:

- Split dataset into training set and test set
- Import Random Forest Model
- Creating a Gaussian Classifier
  clf=RandomForestClassifier(n_estimators=100)
- Train the model using the training sets y_pred=clf.predict(X_test)
- Importing scikit-learn metrics module for accuracy calculation
- Modeling Accuracy

Accuracy obtained:

```
Accuracy: 0.9333333333333333
```

Name: Shlok Khetan
CWID: 10446585

Modeling Complications

As the sole member working on the whole project because of time zone complications made me really appreciate the role of a data scientist.

That is why I chose a dataset which was already relatively clean and a little easier to work with.

Learnt how to effectively use python notebooks for the first time as a challenge to learn something new for data science.

Conclusion

After studying different models on the dataset, it is observed that ANN

provides us the best results with the accuracy of about 96.67%, but that may change

if the dataset gets large enough as then the training of data for ANN would get tedious

Applications:

- With some changes to the code it can be used in the classification of mostly any data set by shifting and changing parameters.
- A website or an application can be used to launch the model for direct customer use
- The model can also be extended to predict the plant's status of other plant types, subject to availability of database.

Name: Shlok Khetan
CWID: 10446585