Veerpartap Singh
Artificial Intelligence 48700

Data mining is the analysis of large observations data to find unsuspected relationships and to summarize the data in different ways in which they are both understandable and useful for the data owner. Data mining tools can forecast the future trends and activities to support the decision of people. It is a discipline, lying at the intersection of statistics, machine learning, data management and databases, patter recognition, and most importantly artificial intelligence.

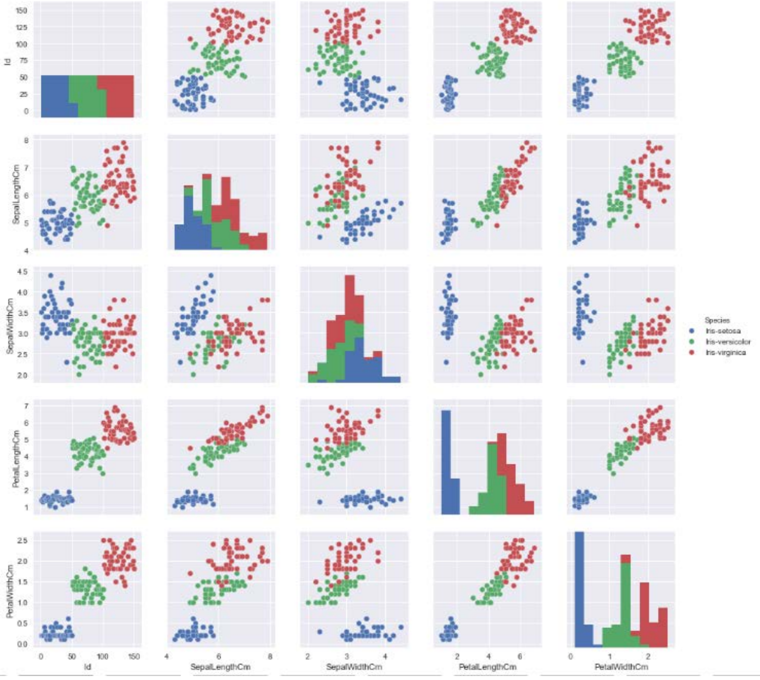
As quoted from the Kaggle's description for this dataset, the iris dataset was used in Fishers classic 1936 paper, "The Use of Multiple Measurements in Taxonomic Problems". It is also available in the UCI Machine Learning Repository. The iris data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The available columns in this dataset are: ID, SepalLengthCm, PetalWidthCm, and species.

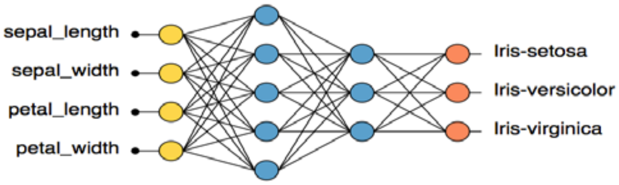
The central goal here is to design a model which makes good classifications for new data using two methods we have learned about. The two methods which will be used on the iris dataset are Neural Networks and Decision Trees. Neural Network is a series of algorithm that attempt to identify relationships in a set of data in a way the human brain works. Neural networks can adapt to changing input, so it can give the best output without the need to redesign the output criteria. Neural networks are hugely growing popularity around trade due to these factors. Decision trees are structure that pretty much represent a tree. The root node which denotes a test attribute, branches which denotes the outcome of test, and leaf which holds a class label. Some benefits for using decision trees are: they are easy to read and understand from visualizations, they do not require much data preparation, and able to handle multi-output problems.

Veerpartap Singh
Artificial Intelligence 48700

Decision Tree Results

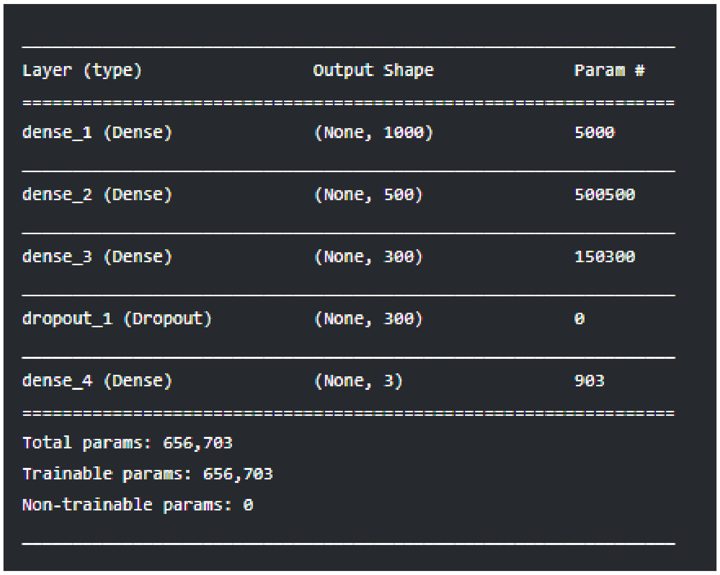
	SepalLength	SepalWidth	PetalLength	PetalWidth
Mean	5.8433	3.0540	3.7587	1.1987
Std	0.8281	0.4336	1.7644	0.7632
Min	4.3000	2.0000	1.0000	.10000
Max	7.9000	4.4000	6.9000	2.5000



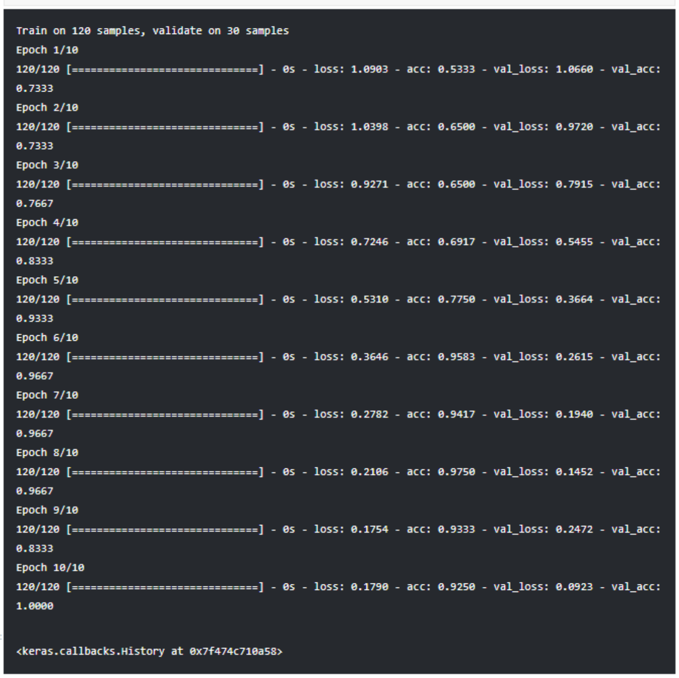


	Predicated Setosa	Predicated Versicolor	Predicated Virginica
Actual Setosa	8	0	0
Actual Versicolor	0	11	2
Actual Virginica	0	0	9

Neural Networks



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 1000)              5000
_________________________________________________________________
dense_2 (Dense)              (None, 500)               500500
_________________________________________________________________
dense_3 (Dense)              (None, 300)               150300
_________________________________________________________________
dropout_1 (Dropout)          (None, 300)               0
_________________________________________________________________
dense_4 (Dense)              (None, 3)                 903
=================================================================
Total params: 656,703
Trainable params: 656,703
Non-trainable params: 0
_________________________________________________________________
```



```
Train on 120 samples, validate on 30 samples
Epoch 1/10
120/120 [==============================] - 0s - loss: 1.0903 - acc: 0.5333 - val_loss: 1.0660 - val_acc:
0.7333
Epoch 2/10
120/120 [==============================] - 0s - loss: 1.0398 - acc: 0.6500 - val_loss: 0.9720 - val_acc:
0.7333
Epoch 3/10
120/120 [==============================] - 0s - loss: 0.9271 - acc: 0.6500 - val_loss: 0.7915 - val_acc:
0.7667
Epoch 4/10
120/120 [==============================] - 0s - loss: 0.7246 - acc: 0.6917 - val_loss: 0.5455 - val_acc:
0.8333
Epoch 5/10
120/120 [==============================] - 0s - loss: 0.5310 - acc: 0.7750 - val_loss: 0.3664 - val_acc:
0.9333
Epoch 6/10
120/120 [==============================] - 0s - loss: 0.3646 - acc: 0.9583 - val_loss: 0.2615 - val_acc:
0.9667
Epoch 7/10
120/120 [==============================] - 0s - loss: 0.2782 - acc: 0.9417 - val_loss: 0.1940 - val_acc:
0.9667
Epoch 8/10
120/120 [==============================] - 0s - loss: 0.2106 - acc: 0.9750 - val_loss: 0.1452 - val_acc:
0.9667
Epoch 9/10
120/120 [==============================] - 0s - loss: 0.1754 - acc: 0.9333 - val_loss: 0.2472 - val_acc:
0.8333
Epoch 10/10
120/120 [==============================] - 0s - loss: 0.1790 - acc: 0.9250 - val_loss: 0.0923 - val_acc:
1.0000

<keras.callbacks.History at 0x7f474c710a58>
```

Veerpartap Singh
Artificial Intelligence 48700

Results:

Decision Tree: For the decision tree I first got a summary of the data. Which shows nothing out of the ordinary. After this the relationship was checked between columns. Blue was setosa, green was versicolor, and red was virginica. This showed for the most part everything was in its own groups besides couple of data sets. The data set was split into 70:30 for training. The neural graph is something I found which shows somewhat of a decision tree. It has the test attributes labeled in yellow then the results in blue, and finally the outcome/label in orange. This lets the user get a representation on how the tree might look. After training some data I tested 30 different instances and out of that only 2 where misclassified. The accuracy was about 93% without doing too much.

Neural Networks: For this data analysis method I did the same as decision tree, I found just the basic summary which I already had. Then I did an 80:20 split on the iris data. I trained the data by using code I found on Kaggle. The accuracy of this results was at 100%. From this you can assert that neural network is trying to learn from each epoch and its existing feature and predict which flower it might be.