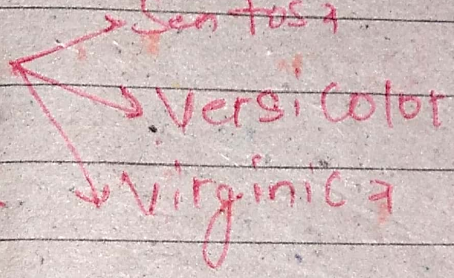
Types of classifier :-

i) Artificial Neural Network
ii) Support Vector Machine (SVM)
iii) Lions
iv) Tigers
v) Bears
vi) Oh my!

But we start with Decision tree - Because it is easy to read & understand.



Sepal length | Sepal Width | Petal length | Petal Width

features

Goal

1. Import dataset (of this Iris flower)
2. Train a classifier
3. Predict label for new flowers
4. Visualize the tree.

IMPORT DATASET

We can import iris dataset like this —

```
from sklearn.datasets import load_iris
iris = load_iris()
```

This dataset include both the table from wikip. & meta data

```
print iris.feature_names
print iris.target_names
```

→ The feature & example themselves contain in the data variable

The metadata tell us the name of diffent types of features and the name of different types of flowers.

```
print iris.data[0]
```

feature?

→ Printing first entry of dataset we will find the measurment of this flower

['sepal length (cm)', 'sepal wi(cm)', 'Petal L', 'Petal W']

['Sentosa', 'versicolor', 'virginica']

Output:— [5.1 3.5 1.4 0.2]

No target variable is labels.

```
print iris.target[0]
```

↓

The target valiable contains the label

output = 0 means its Sentosa

for entire dataset

```
For i in range(len(iris.target)):
    print("Example %d: label %d %s, feature %s"
          %(i, iris.target[i], iris.data[i]))
```

CLASSIFIER

↳ Example used to "test" the classifier's accuracy.
Not part of the training data.

This is used to test accuracy of classifier

```
from sklearn import tree
import numpy as np
from sklearn.datasets import load_iris
iris = load_iris()
test_idx = [0, 50, 100]
```

for Removing 1st example of each dataset

(0 → Sentosa, 50 → Versicolor, 100 → Verginical)

```
# training data
train_target = np.delete(iris.target, test_idx)
train_data = np.delete(iris.data, test_idx, axis=0)
```

↳ It will have majority of data

```
# testing data
test_target = iris.target[test_idx]
test_data = iris.data[test_idx]
```

↳ It will be have only the data which has removed.

```
# creating decisiontree classifier & train it on our training data.
clf = tree.DecisionTreeClassifier()
clf = clf.fit(train_data, train_target)
```

PREDICT THE LABEL FOR NEW FLOWER

```
# Now use the tree to classify our testing data
print test_target
```

output = [0, 1, 2]
↳ label of each type

```
# Now we see what the tree predict on this testing data
print clf.predict(test_data)
```

[features of our testing data]

output = [0 1 2]

Means predicted label matey our testing label. mean

visualize the tree

```
from StringIO import StringIO
import pydot
dot_data = StringIO()
tree.export_graphviz(clf, out_file=dot_data,
                     feature_names = iris.feature_names,
                     class_names = iris.target_names
                     filled = True, rounded = True,
                     impurity = False)

graph = pydot.graph_from_dot_data(dot_data.getvalue())
graph.write_pdf('iris.pdf')
```

Now lets use this tree to classify an example from our testing data.

```
Print test_data[0], test_target[0]
```

output → [5.1 3.5 1.4 0.2] 0

in this way we can test our data by its value we get diff labels

features → [5.1 3.5 1.4 0.2]

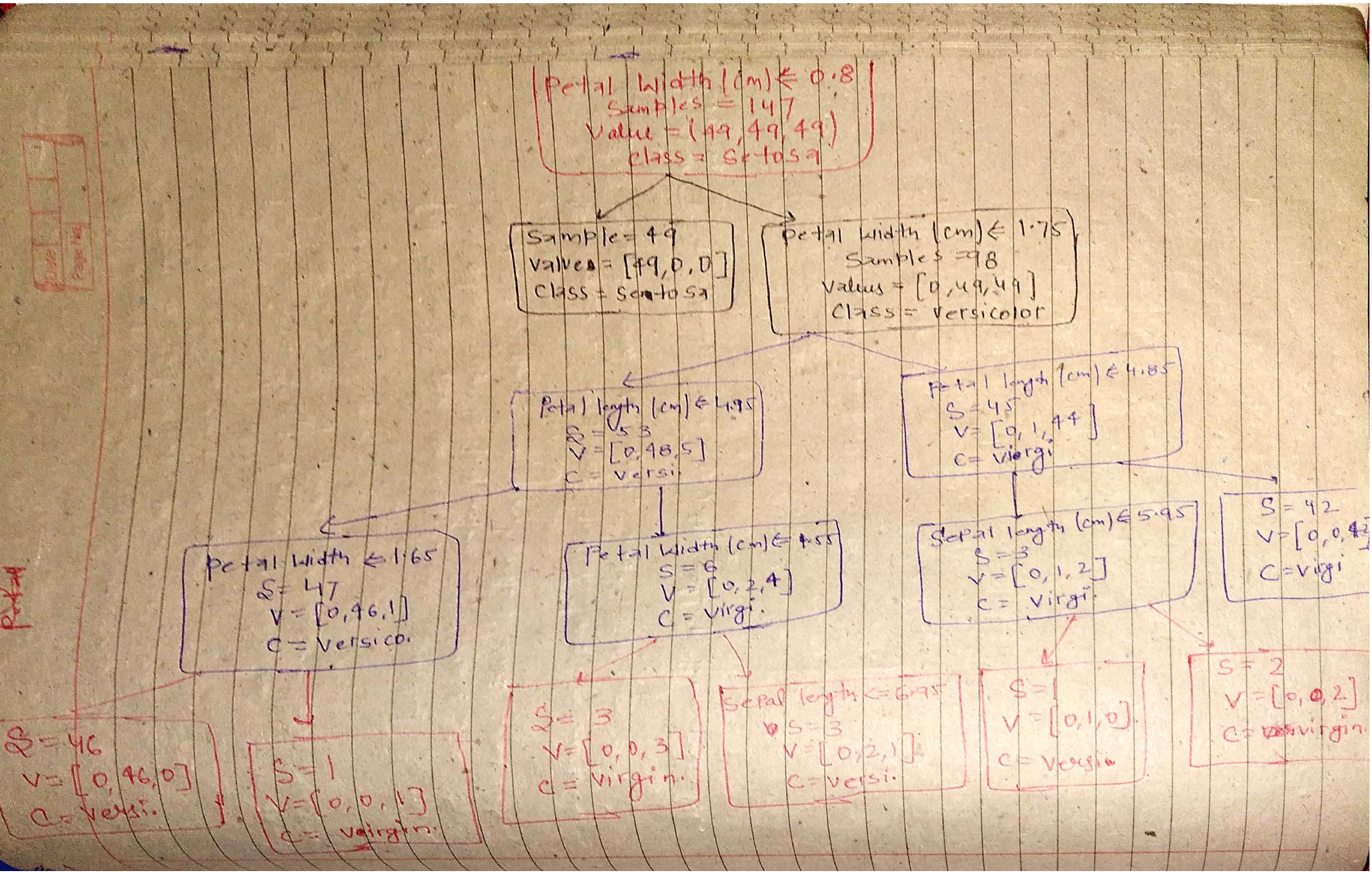
labels → 0

for 1st testing flowers

We can find feature name by looking at the metadata

```
print iris.feature_names, iris.target_names
```

[S, Ve, Vir] → [-(cm), -(cm), -(cm), -(cm)]


Petal Width (cm) ≤ 0.8
Samples = 147
Value = [49, 49, 49]
class = Setosa
Sample = 49
values = [49, 0, 0]
Class = Setosa
Petal width (cm) ≤ 1.75
Samples = 98
Values = [0, 49, 49]
Class = versicolor
Petal length (cm) ≤ 4.95
S = 5 5
V = [0, 48, 5]
C = versi.
Petal length (cm) ≤ 4.85
S = 45
V = [0, 1, 44]
C = virgi
Petal width ≤ 1.65
S = 47
V = [0, 46, 1]
C = Versico.
Petal width (cm) ≤ 4.55
S = 6
V = [0, 2, 4]
C = virgi.
Sepal length (cm) ≤ 5.95
S = 3
V = [0, 1, 2]
C = virgi.
S = 42
V = [0, 0, 42]
C = virgi
S = 46
V = [0, 46, 0]
C = versi.
S = 1
V = [0, 0, 1]
C = virgin.
S = 3
V = [0, 0, 3]
C = virgin.
Sepal length ≤ 6.95
S = 3
V = [0, 2, 1]
C = versi.
S = 1
V = [0, 1, 0]
C = Versio
S = 2
V = [0, 0, 2]
C = virgin.