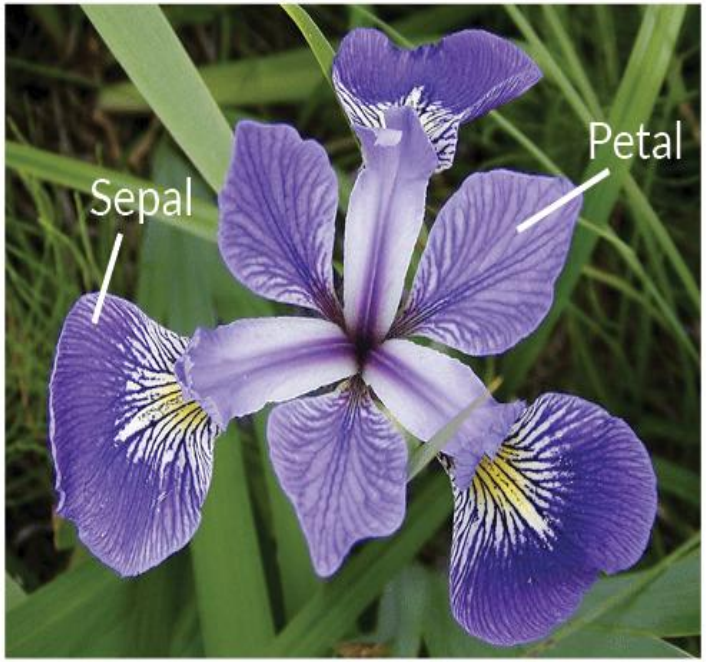
Your First Machine Learning Project

Prerequisite:
What is classification?

By
Dr. S Rao Chintalapudi(Subhash)
Professor & HoD,
Department of CSE (AI & ML),
CMR Technical Campus (Autonomous)
hod.aiml@cmrtc.ac.in

Classification of iris flowers



Iris Versicolor

Iris Setosa

Iris Virginica

Problem Statement

Given Sepal, Petal lengths and widths predict the class of iris

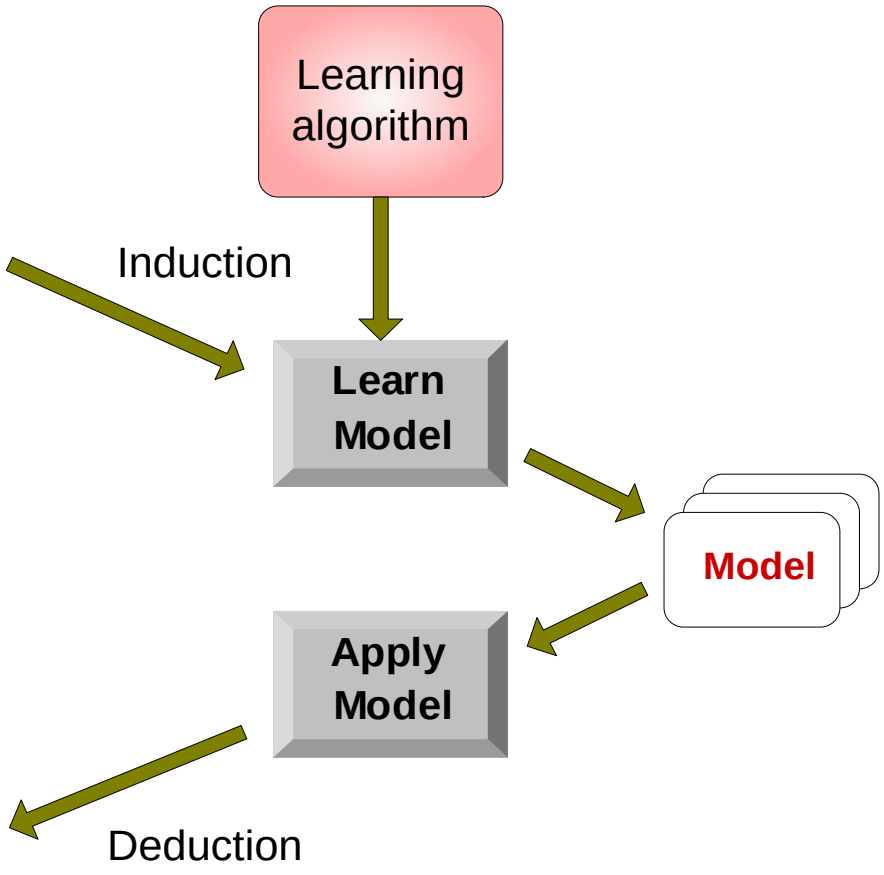
Building a classification model

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Classification Techniques

- Base Classifiers
  - Decision Tree based Methods
  - Rule-based Methods
  - Nearest-neighbor
  - Neural Networks
  - Deep Learning
  - Naïve Bayes and Bayesian Belief Networks
  - Support Vector Machines

- Ensemble Classifiers
  - Boosting, Bagging, Random Forests

Dataset Description

- 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
- 150 records – 50 per class
- Dataset Link : iris.csv

What is a CSV file?

Import required packages

```
import numpy as np
import pandas as pd
```

Load the iris dataset

```
import numpy as np
import pandas as pd
iris=pd.read_csv(“D:\mlproject\iris.csv”)
iris
```

View iris dataset

```
[14] iris
```

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	iris_setosa
1	4.9	3.0	1.4	0.2	iris_setosa
2	4.7	3.2	1.3	0.2	iris_setosa
3	4.6	3.1	1.5	0.2	iris_setosa
4	5.0	3.6	1.4	0.2	iris_setosa
5	5.4	3.9	1.7	0.4	iris_setosa
6	4.6	3.4	1.4	0.3	iris_setosa
7	5.0	3.4	1.5	0.2	iris_setosa
8	4.4	2.9	1.4	0.2	iris_setosa
9	4.9	3.1	1.5	0.1	iris_setosa
10	5.4	3.7	1.5	0.2	iris_setosa
11	4.8	3.4	1.6	0.2	iris_setosa

Splitting the dataset

```
x=iris.drop(columns=['class'])
y=iris['class']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test
  _size=.2)
```

Build the model

```
from sklearn import neighbors
classifier=neighbors.KNeighborsClassifier()
```

Train the model

```
classifier.fit(x_train,y_train)
```

Make Predictions

```
predictions=classifier.predict(x_test)
```

Compute Accuracy

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,predictions))
```

Output

```
[50] from sklearn import neighbors
     classifier=neighbors.KNeighborsClassifier()
```

```
[51] classifier.fit(x_train,y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

```
[53] predictions=classifier.predict(x_test)
```

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,predictions))
```

```
0.9666666666666667
```

Full code

```
from sklearn import datasets
iris=datasets.load_iris()
x=iris.data
y=iris.target
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.2)
from sklearn import tree
classifier=tree.DecisionTreeClassifier()
classifier.fit(x_train,y_train)
predictions=classifier.predict(x_test)
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,predictions))
```

THAT'S IT …
YOU MADE IT…