

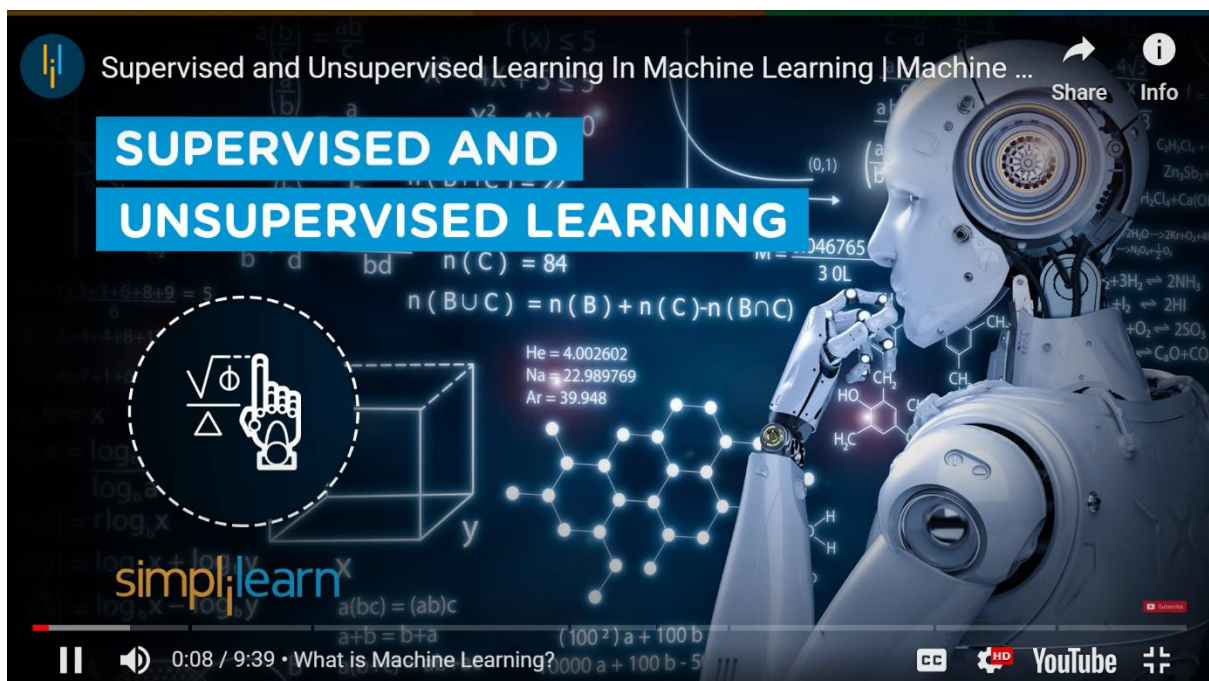
Preparation Phase

Prior Knowledge

Date	11 October 2022
Team ID	PNT2022TMID21553
Project Name	Project – Car Resale Value Prediction

Prior Knowledge – Car Resale Value Prediction:

1. Supervised and Unsupervised learning:



What is Machine Learning?

What is Machine Learning?

Machine Learning is the science of making computers learn and act like humans by feeding data and information without being explicitly programmed!

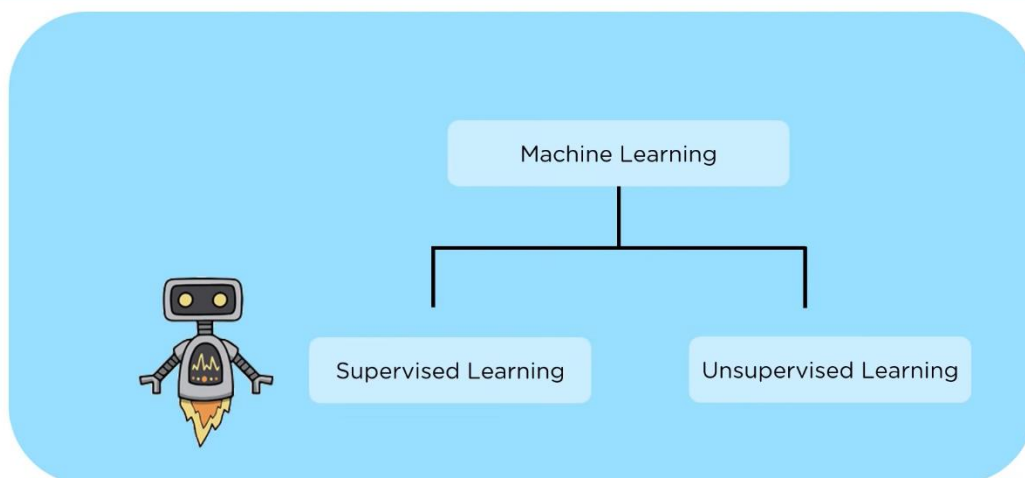


©Simplilearn. All rights reserved.

simplilearn

Types of Machine Learning:

Types of Machine Learning

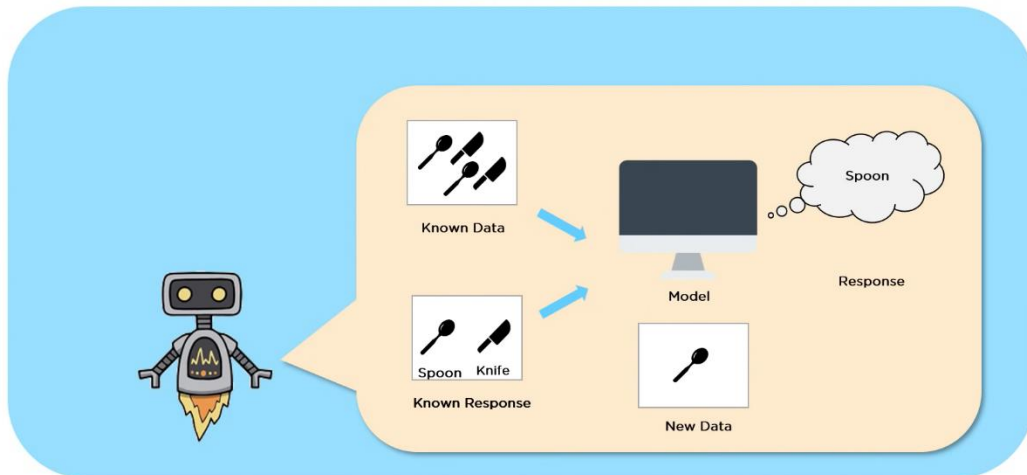


©Simplilearn. All rights reserved.

simplilearn

Supervised Learning:

Supervised Learning

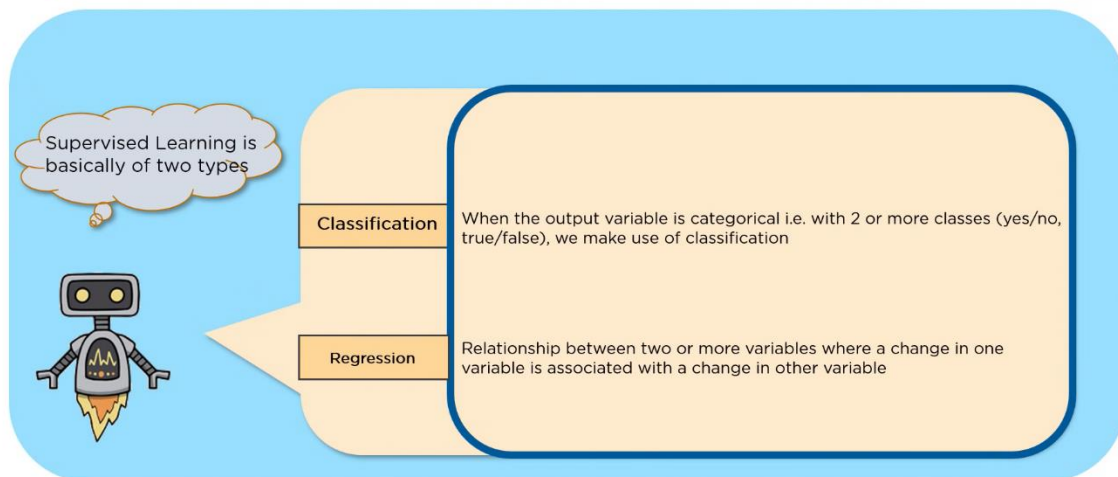


©Simplilearn. All rights reserved.

simplilearn

Types of Supervised Learning:

Types of Supervised Learning

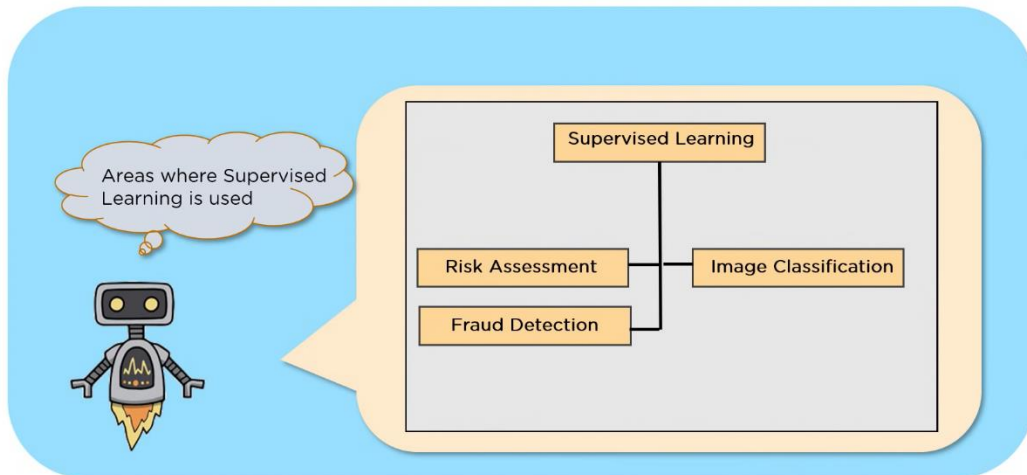


©Simplilearn. All rights reserved.

simplilearn

Applications of Supervised Learning:

Applications of Supervised Learning

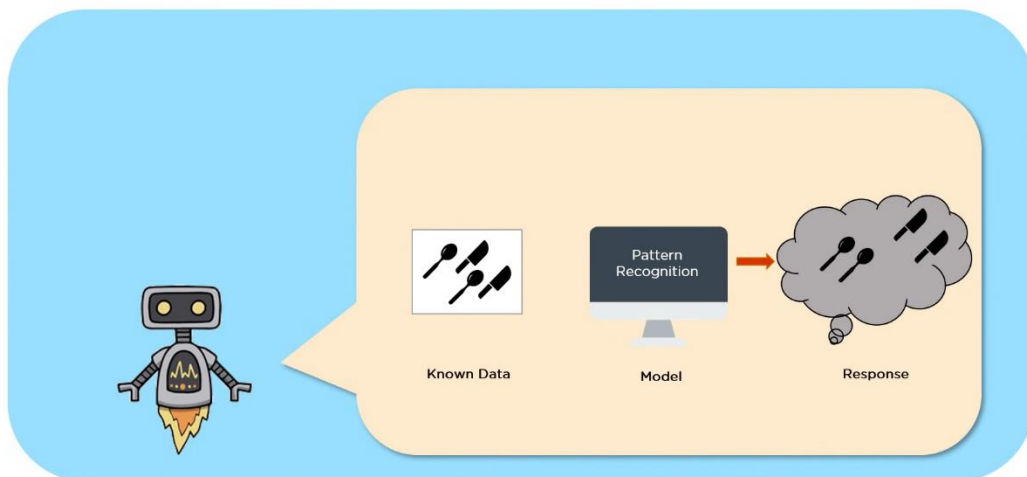


©Simplilearn. All rights reserved.

simplilearn

Unsupervised Learning:

Unsupervised Learning

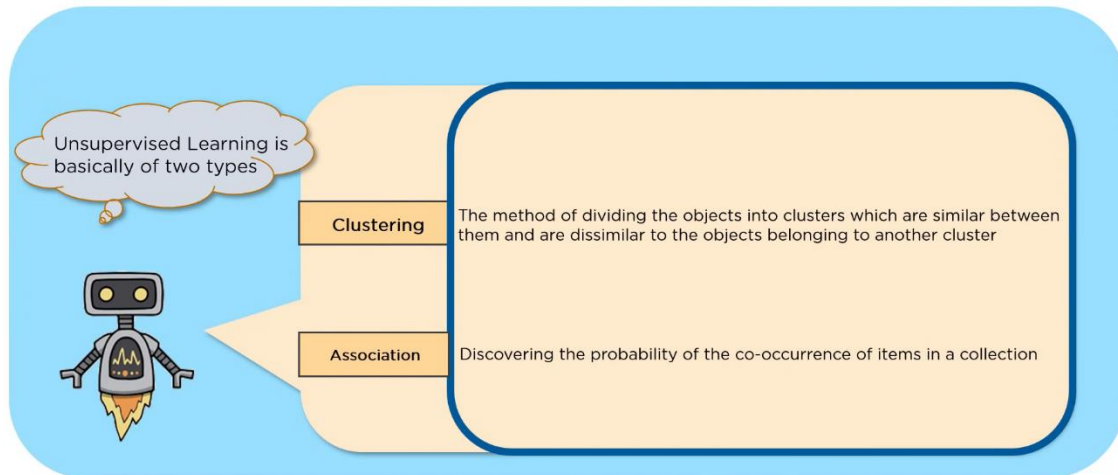


©Simplilearn. All rights reserved.

simplilearn

Types of Unsupervised Learning:

Types of Unsupervised Learning

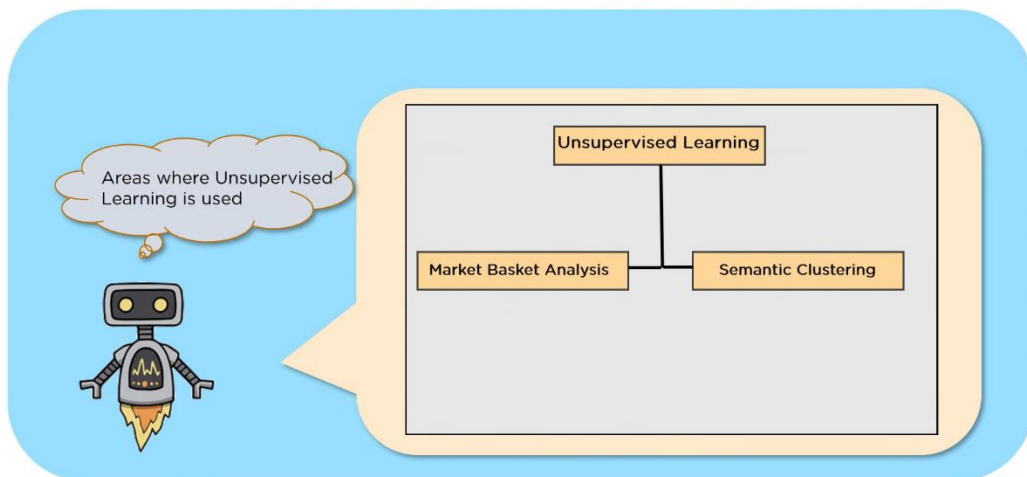


©Simplilearn. All rights reserved.

simplilearn

Applications of Unsupervised Learning:

Applications of Unsupervised Learning



©Simplilearn. All rights reserved.

simplilearn

2. Regression, Classification and Clustering:

Common ML Problems

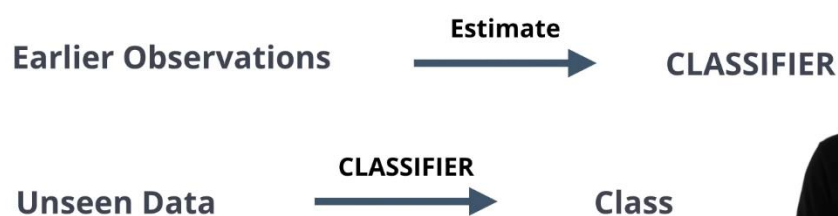
- Classification
- Regression
- Clustering



Classification Problem:

Classification Problem

Goal: predict category of new observation



Classification Applications:

Classification Applications

- **Medical Diagnosis** Sick and Not Sick
- **Animal Recognition** Dog, Cat and Horse

Important:

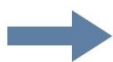
- Qualitative Output
- Predefined Classes



Regression:

Regression

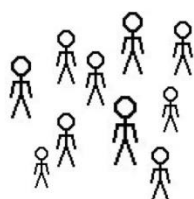
PREDICTORS



REGRESSION
FUNCTION



RESPONSE



- Relationship: **Height - Weight?**
- Linear?
- Predict: **Weight** → **Height**



Regression Model:

Regression Model

Fitting a **linear** function

$$\text{Height} \approx \beta_0 + \beta_1 \times \text{Weight}$$

- **Predictor:** Weight
- **Response:** Height
- **Coefficients:** β_0, β_1

Estimate on previous input-output

```
> lm(response ~ predictor)
```



Regression Applications:

Regression Applications

- Payments → Credit Scores
- Time → Subscriptions
- Grades → Landing a Job
- Quantitative Output
- Previous **input-output** observations



Clustering:

Clustering

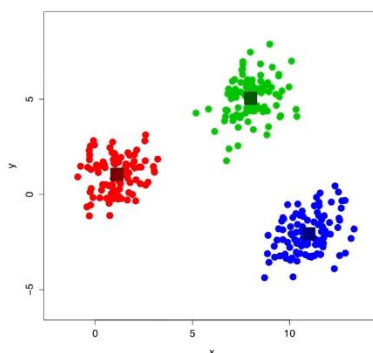
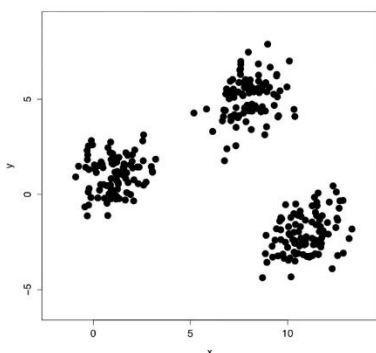
- **Clustering:** grouping objects in clusters
 - *Similar* within cluster
 - *Dissimilar* between clusters
- **Example:** Grouping similar animal photos
 - No labels
 - No **right** or **wrong**
 - Plenty possible clusterings



k-Means clustering:

k-Means

Cluster data in **k** clusters!



3. Random Forest Regressor:

Applications of Random Forest:

Application of Random Forest



Remote Sensing

Used in ETM devices to acquire images of the earth's surface.

Accuracy is higher and training time is less



Object Detection

Multiclass object detection is done using Random Forest algorithms

Provides better detection in complicated environments



Kinect

Random Forest is used in a game console called Kinect

Tracks body movements and recreates it in the game

©Simplilearn. All rights reserved.

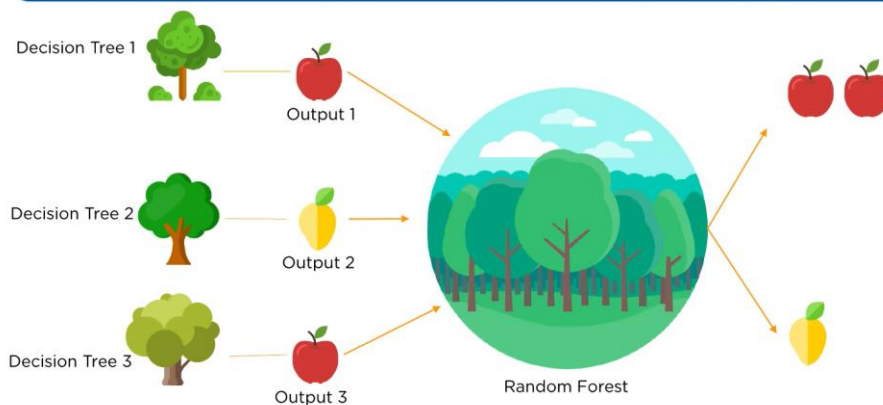
simplilearn

Random Forest definition:

What is Random Forest?

Random forest or Random Decision Forest is a method that operates by constructing multiple Decision Trees during training phase.

The Decision of the majority of the trees is chosen by the random forest as the final decision



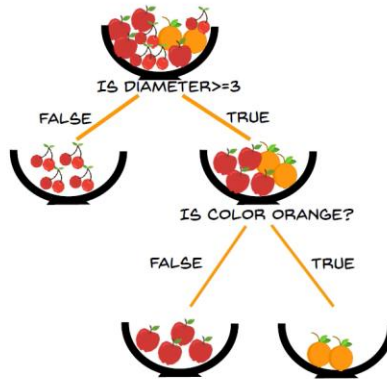
©Simplilearn. All rights reserved.

simplilearn

Decision Tree definition:

Decision Tree

Decision Tree is a tree shaped diagram used to determine a course of action. Each branch of the tree represents a possible decision, occurrence or reaction

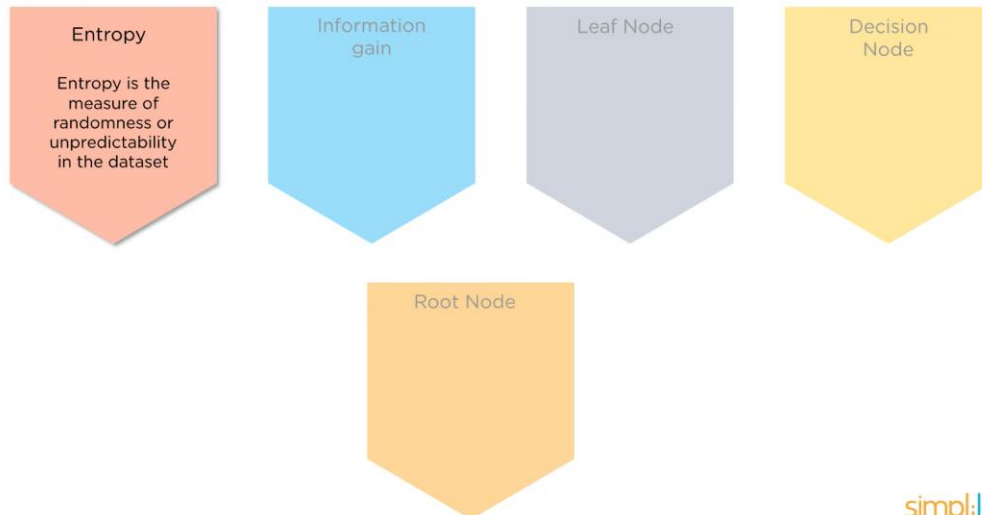


©Simplilearn. All rights reserved.

simplilearn

Important terms in decision tree:

Decision Tree- Important Terms

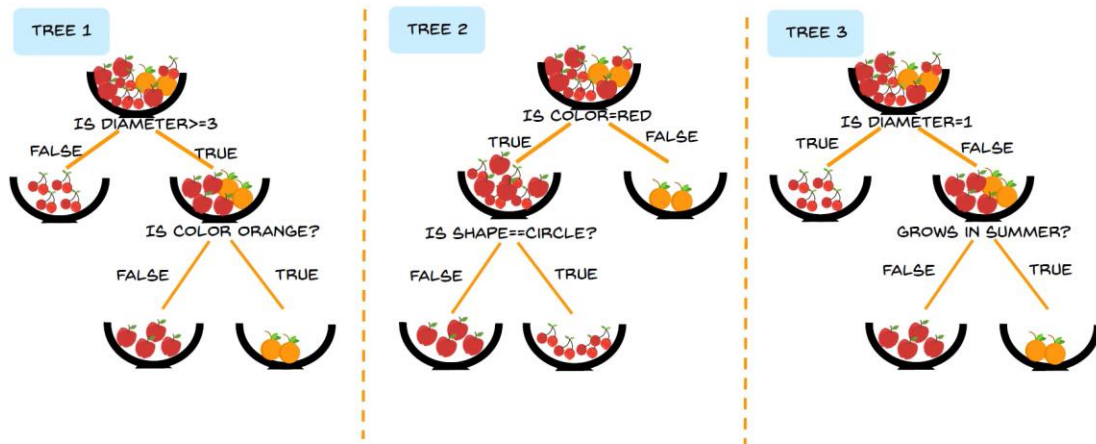


©Simplilearn. All rights reserved.

simplilearn

Random Forest Working:

How does a Random Forest work?



©Simplilearn. All rights reserved.

simplilearn

Random Forest using python with iris dataset:

```
jupyter Simplilearn_LearningTree_ (unsaved changes) Logout
File Edit View Insert Cell Kernel Help Trusted Python 3
In [2]: # Loading the library with the iris dataset
from sklearn.datasets import load_iris
# Loading scikit's random forest classifier library
from sklearn.ensemble import RandomForestClassifier
# Loading pandas
import pandas as pd
# Loading numpy
import numpy as np
# Setting random seed
np.random.seed(0)

In [ ]: # Creating an object called iris with the iris data
irisI= load_iris()
# print(iris)
# Creating a dataframe with the four feature variables
df= pd.DataFrame(iris.data, columns=iris.feature_names)
# Viewing the top 5 rows
df.head()
```


Installation - Prerequisite



Prerequisite

virtualenv



Virtual Python Environment builder



pip install virtualenv



Sudo apt-get install virtualenv



I'm learning Flask!

edureka!

Python Certification Training

www.edureka.co/python



Flask Application:

Flask - Application



Test Installation

Use this simple code, save it as Hello.py

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World'

if __name__ == '__main__':
    app.run()
```



I'm learning Flask!

edureka!

Python Certification Training

www.edureka.co/python



Flask - Application



App.run(host, port, options)

All these parameters are optional

Sl.no	Parameter	Description
1	host	Hostname to listen on. Defaults to 127.0.0.1 (localhost). Set to '0.0.0.0' to have server available externally
2	port	Defaults to 5000
3	debug	Defaults to false. If set to true, provides a debug information
3	options	To be forwarded to underlying Werkzeug server.

Python hello.py



Variable rules in Flask:

Flask – Variable Rules



It is possible to build a URL dynamically!

How?

By adding variable parts to the rule parameter

Consider the example

```
from flask import Flask
app = Flask(__name__)

@app.route('/hello/<name>')
def hello_name(name):
    return 'Hello %s!' % name

if __name__ == '__main__':
    app.run(debug = True)
```

http://localhost:5000/hello/Edureka



Flask – Variable Rules



More rules can be constructed using these converters

Sl.no	Parameter	Description
1	int	Accepts Integer
2	Float	For Floating point value
3	Path	Accepts slashes used as directory separator character

Visit the URL: <http://localhost:5000/blog/11>

Browser Output: **Blog number 11**

<http://localhost:5000/rev/1.1> → **Revision Number 1.100000**

```
from flask import Flask
app = Flask(__name__)

@app.route('/blog/<int:postID>')
def show_blog(postID):
    return 'Blog Number %d' % postID

@app.route('/rev/<float:revNo>')
def revision(revNo):
    return 'Revision Number %f' % revNo

if __name__ == '__main__':
    app.run()
```

Run the code

edureka!

Python Certification Training

www.edureka.co/python



Flask – HTTP methods:

Flask – HTTP Methods



HTTP Protocols are the foundation for data communication in WWW

Sl.no	Method	Description
1	GET	Sends data in unencrypted form to server
2	HEAD	Same as GET, but without response body
3	POST	Used to send HTML form data to server.
4	PUT	Replaces all current representations of target resource with uploaded content
5	DELETE	Removes all current representations of target resource given by URL

edureka!

Python Certification Training

www.edureka.co/python



Flask uses jinja2 template

Flask – Templates



Flask uses jinja2 template engine

```
<!doctype html>
<html>
  <body>

    <h1>Hello {{ name }}!</h1>

  </body>
</html>
```

```
from flask import Flask, render_template
app = Flask(__name__)

@app.route('/hello/<user>')
def hello_name(user):
    return render_template('hello.html', name = user)

if __name__ == '__main__':
    app.run(debug = True)
```

Flask will try to find the HTML file in the templates folder, in the same folder in which this script is present.

The **Jinja2** template engine uses the following delimiters for escaping from HTML

- {% ... %} for Statements
- {{ ... }} for Expressions to print to the template output
- {# ... #} for Comments not included in the template output
- # ... ## for Line Statements

