**Problem**

**x = [5,7,8,7,2,17,2,9,4,11,12,9,6]**

**y = [99,86,87,88,111,86,103,87,94,78,77,85,86]**

**The x array represents the age of each car.**

**The y array represents the speed of each car.**

1. **Draw scatter plot to predict future values.**
2. **Let us try to predict the speed of a 10 years old car.**

## Scatter Plot

* A scatter plot is a diagram where each value in the data set is represented by a dot.
* The Matplotlib module has a method for drawing scatter plots,
* it needs two arrays of the same length,
* one for the values of the x-axis, and one for the values of the y-axis:

x = [5,7,8,7,2,17,2,9,4,11,12,9,6]

y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

* The x array represents the age of each car.
* The y array represents the speed of each car.

#Import the necessary python library

import matplotlib.pyplot as plt  
  
x=[5,7,8,7,2,17,2,9,4,11,12,9,6]  
y=[99,86,87,88,111,86,103,87,94,78,77,85,86]  
  
plt.scatter(x,y)  
plt.show()



Step 3

## Predict Future Values

Now we can use the information we have gathered to predict future values.

Example: Let us try to predict the speed of a 10 years old car.

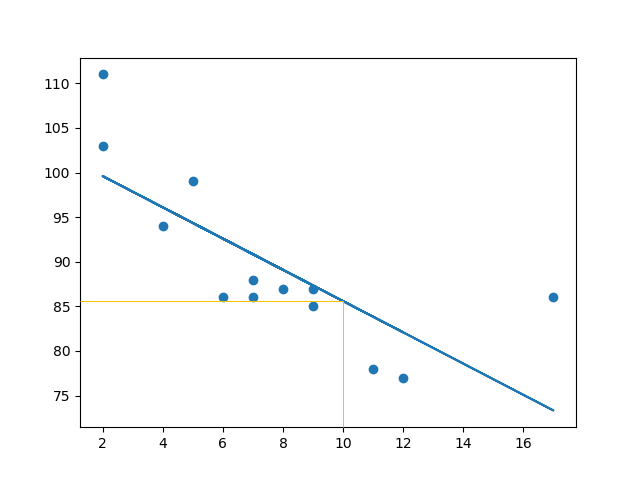
To do so, we need the  myfunc() function from the example above:

def myfunc(x):  
  return slope \* x + intercept

phase 3

Import scipy and draw the line of Linear Regression:

import matplotlib.pyplot as plt  
from scipy import stats  
  
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]  
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]  
  
slope, intercept, r, p, std\_err = stats.linregress(x, y)  
  
def myfunc(x):  
  return slope \* x + intercept  
  
mymodel = list(map(myfunc, x))  
  
plt.scatter(x, y)  
plt.plot(x, mymodel)  
plt.show()



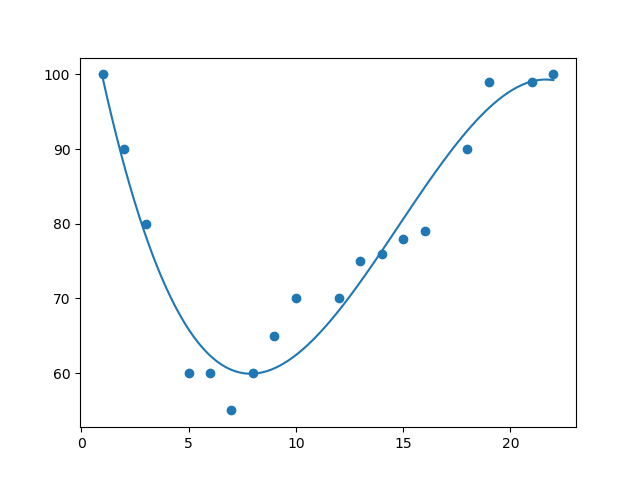
[Python Machine Learning Linear Regression (w3schools.com)](https://www.w3schools.com/python/python_ml_linear_regression.asp)

**Problem**

**In the example below, we have registered 18 cars as they were passing a certain tollbooth.We have registered the car's speed, and the time of day (hour) the passing occurred.The x-axis represents the hours of the day and the y-axis represents the speed:**

**Polynomial regression model**

Polynomial regression, like linear regression, uses the relationship between the variables x and y to find the best way to draw a line through the data points.

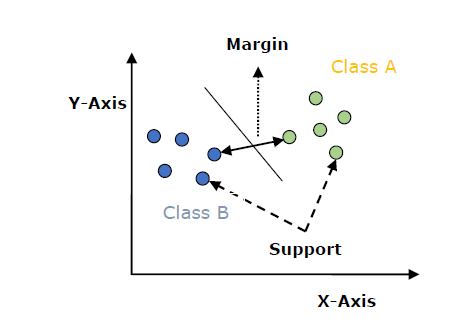


**Introduction to SVM**

* Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression
* extremely popular because of their ability to handle multiple continuous and categorical variables.

**Working of SVM**

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).



* **Support Vectors** − Datapoints that are closest to the hyperplane is called support vectors.
* **Hyperplane** − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.
* **Margin** − It may be defined as the gap between two lines on the closet data points of different classes.

## Implementing SVM in Python

**Phase 1**

## Scatter Plot

* A scatter plot is a diagram where each value in the data set is represented by a dot.
* The Matplotlib module has a method for drawing scatter plots,
* it needs two arrays of the same length,
* one for the values of the x-axis, and one for the values of the y-axis:

In the example below, we have registered 18 cars as they were passing a certain tollbooth.

We have registered the car's speed, and the time of day (hour) the passing occurred.

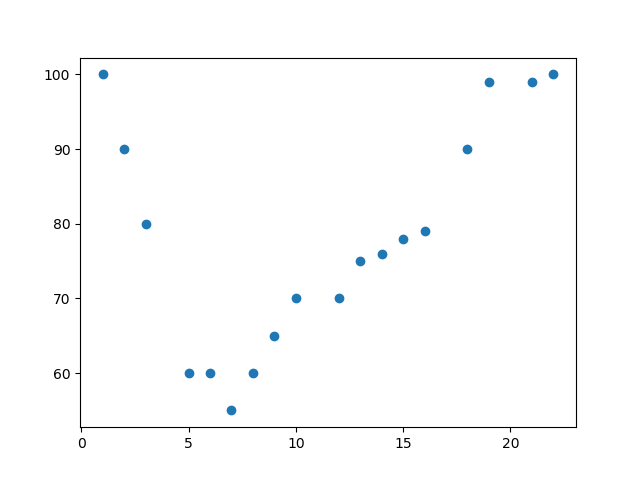
The x-axis represents the hours of the day and the y-axis represents the speed:

Phase 1

Import the necessary library from python

import matplotlib.pyplot as plt  
  
x = [1,2,3,5,6,7,8,9,10,12,13,14,15,16,18,19,21,22]  
y = [100,90,80,60,60,55,60,65,70,70,75,76,78,79,90,99,99,100]  
  
plt.scatter(x, y)  
plt.show()

### Result:



Phase 2

## Predict Future Values

Now we can use the information we have gathered to predict future values.

Example: Let us try to predict the speed of a car that passes the tollbooth at around the time 17:00:

To do so, we need the  mymodel array from the example above:

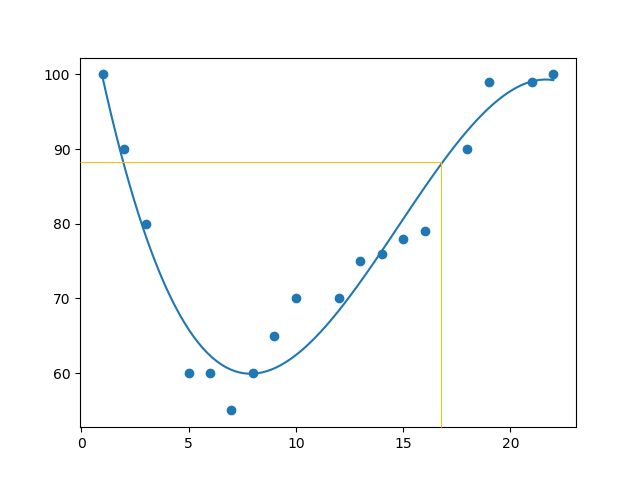
mymodel = numpy.poly1d(numpy.polyfit(x, y, 3))

phase 3

Predict the speed of a car passing at 17:00:

import numpy  
from sklearn.metrics import r2\_score  
  
x = [1,2,3,5,6,7,8,9,10,12,13,14,15,16,18,19,21,22]  
y = [100,90,80,60,60,55,60,65,70,70,75,76,78,79,90,99,99,100]  
  
mymodel = numpy.poly1d(numpy.polyfit(x, y, 3))  
  
speed = mymodel(17)  
print(speed)

The example predicted a speed to be 88.87, which we also could read from the diagram:



[Python Machine Learning Polynomial Regression (w3schools.com)](https://www.w3schools.com/python/python_ml_polynomial_regression.asp)