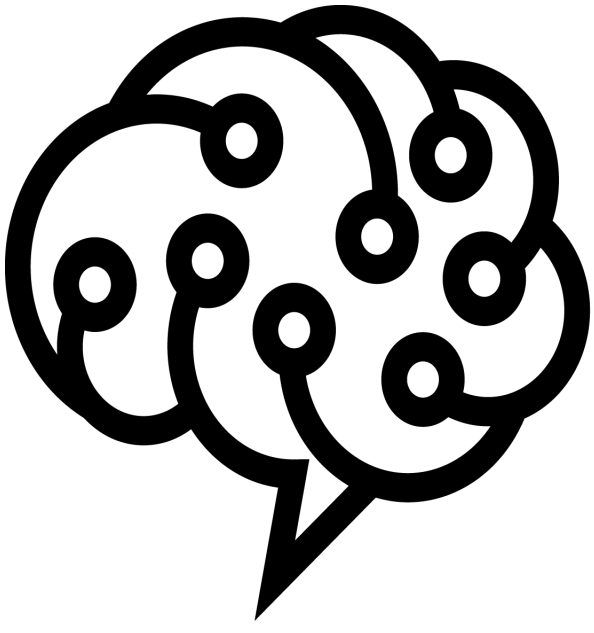
~Inspiring Lives~



school of ai

Raleigh, NC

Jubeen Shah

jnshah2@ncsu.edu

Some text collected by Sudha Mantri (Dean, Leesburg School of AI, VA, USA)

The Dawn

What is School of AI?

An international school dedicated to studying, teaching, and creating Artificial Intelligence to help solve the world's most difficult problems.

Who is the director?

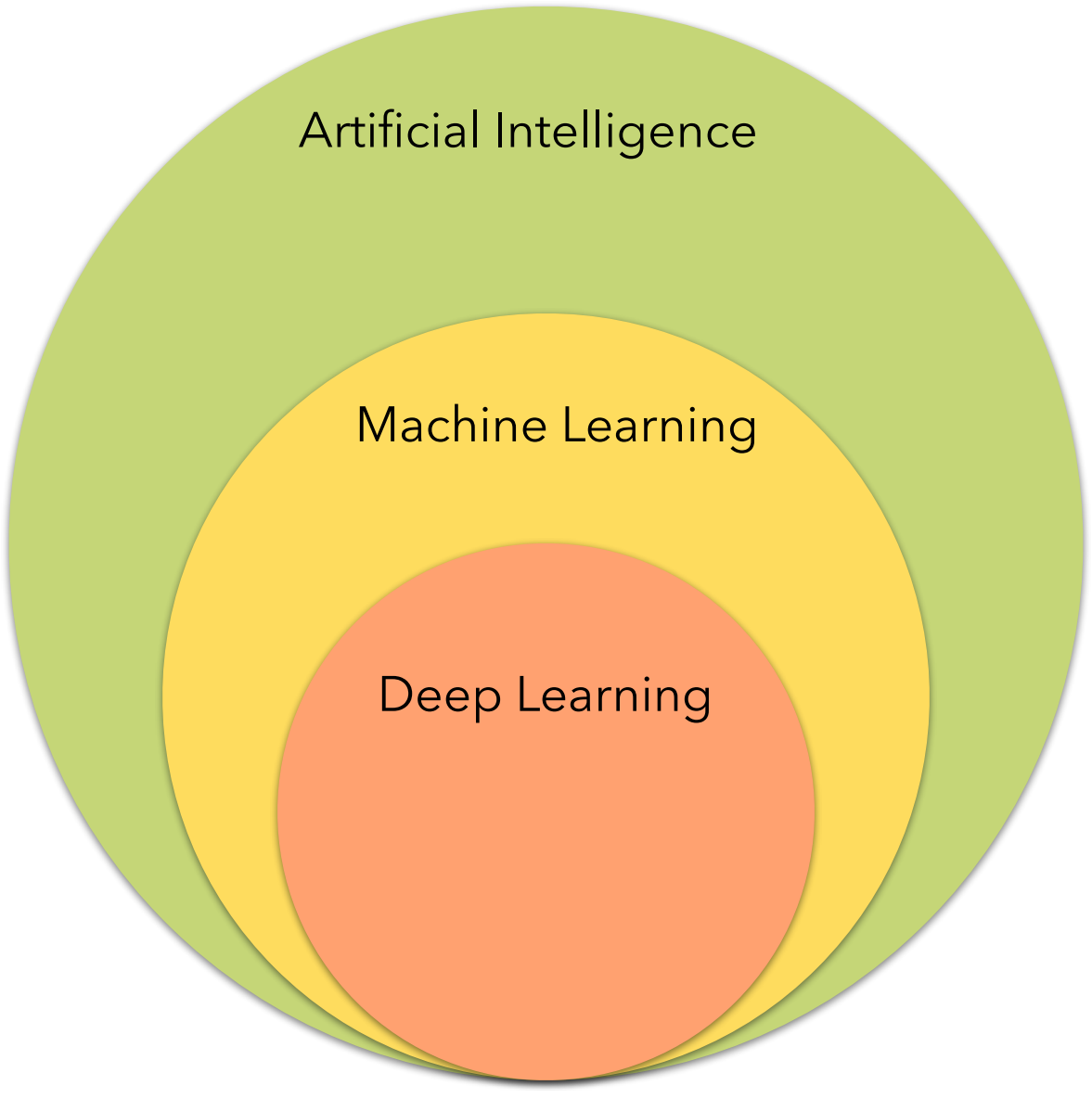
Siraj Raval - He lives to serve all Wizards [the shared moniker of the community]. Inspiring, educating, and guiding them along their journey to help them maximize their positive impact in the world using AI technology.

Who are the deans?

Deans are guardians of our mission - "To offer a world-class AI education to anyone on Earth for free. Our doors are open to all those who wish to learn. We are a learning community that spans almost every country dedicated to teaching our students how to make a positive impact in the world using AI technology, whether that's through employment or entrepreneurship"

What is Artificial Intelligence?

Artificial intelligence is an ability to apply knowledge to real world problems demonstrated by machines. Stuart Russel in his book Artificial Intelligence : A modern approach stated that there are four organized structured definition of AI – *Thinking Humanly, Thinking Rationally, Acting Humanly, and Acting Rationally*. Humans are addictively fascinated with defining and categorizing everything. The benefit I see in that exercise is it makes communication easy (please note I said communication not understanding). There are many definitions for Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL). We can say that right now, AI field is still in its infancy and nomenclature is fluid. There is tremendous opportunity in the field.
Machine learning on the other hand is a subset of artificial intelligence. According to SAS *"Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention."*[1]



[1] https://www.sas.com/en_us/insights/analytics/machine-learning.html

Difference between AI, ML, and DL

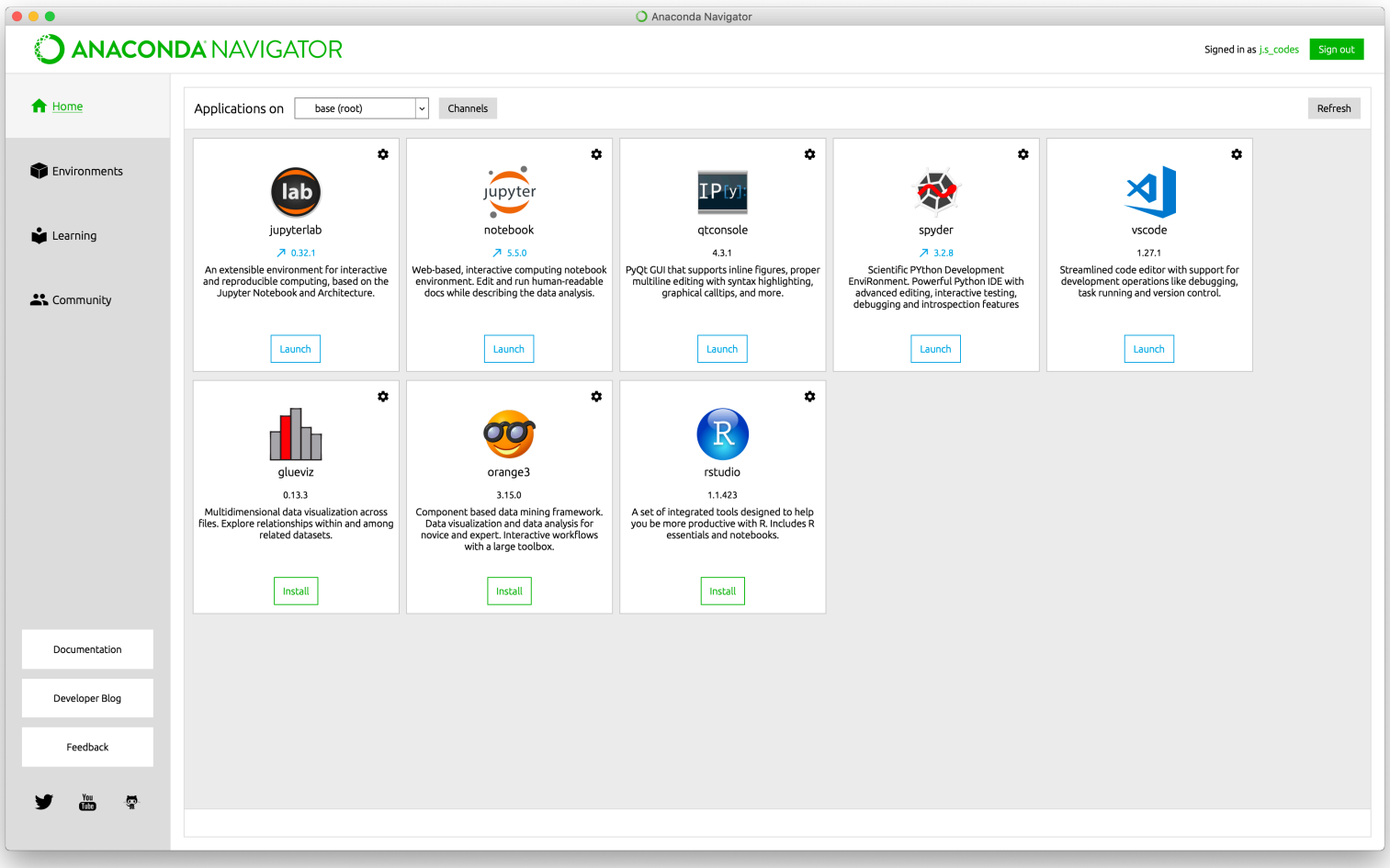
	Artificial Intelligence (AI)	Machine Learning (ML)	Deep Learning (DL)
Definition	Intelligent machines that are meant to imitate/think/act more like human beings	System learns things without being programmed to do so. Logistic Regression. System is able to make predictions or take decisions based on the past data. Model being trained based on features. This can be done on smaller amounts of data.	System thinks and learns like human brain using artificial neural networks. Initial dataset is small with less testing time. Performance improves with more data and continual learning. Concept of weights, bias, variance. Problem solved in an end to end method. Best features are selected by the system
Progress	Era of Weak AI. Machines in their infancy. Any code or technique or algorithm that enables machines to mimic develop and demonstrate human cognition and behavior.	To move from weak AI to strong AI machines need to learn the ways of humans. Techniques and process of ML help machines in this endeavor Machines learn to predict outcomes on the go by recognizing patterns in the input data.	Machines demonstrate ability to learn deeply by drawing meaningful inferences from large data sets. DL requires Artificial Neural Networks (ANN) in layers which are like biological neural networks in humans. ANNs connect and communicate with each other to make sense of vast input data
Types (Just categorization)	1.Reactive machines (Say clothes are clumped on one side in a washing machine, then machine re-centers clothes). No past experience/memory for a new decision 2.Limited memory (Say neural network programmed to identify car). Machine may use short-lived or limited/past information 3.Theory of mind. Systems able to understand human emotions and adjust self-behavior according to the human needs (popularly known as user experience) 4.Self-awareness (self- fixing. Predicting which action is best for the humans – to make human life happy). Systems are aware of themselves and understand their own internal states. Systems predict other people's feelings/actions.	1.Supervised learning: Systems are able to predict future outcomes based on the past data (system learns from bunch of input images. Then it matches a new image). Requires both input and an output to be given to the model for it to be trained 2.Unsupervised learning: Systems are able to identify hidden patterns from the input data provided. By clustering etc., the patterns and similarities/ anomalies become apparent to the system 3.Reinforcement Learning: Systems are given no training. It may learn on the basis of reward/punishment (Yes/No answers)	DL uses different approaches, algorithms to achieve Deeper learning. 1. Multi Layer Perceptrons (MLPs) 2. Convolutional neural net (CNNs)
Examples	News generation – finds news articles based on news feeds, Google command – smart home devices. Amazon prime music, SAAVN, Philips Hue, OLA, NDTV, Zomata, TED, Goibibo, ESPN Cricinfo, Amazon Echo etc., KBS –DENDRAL, MYSIN, XCON/RI	Spam detection, Chats, Mail categories (All mail, Trash, important, spam), search engine result refining (weights - categorized) - Google search.	Sortie (translation) - neural network (different words and patterns) Chat boats – ask a question it gives answer. Specific tasks – IBM deep blue, Google DeepMind, Google AlphaGo. Convert a black and white image to, Automated library categorization of books – translation of handwritten/ printed data to digital form
Future is happening	Detect crimes before they happen, humanoid AI helpers	Increase efficiency in healthcare, better marketing techniques, Finance, sports, trading, trading.	Increase personalization – hyper intelligent systems, decision making.

Tools

Anaconda

Version >5.0, python version 3.x

- About
  - Anaconda is a distribution of packages built for data science.
  - It comes with conda, a package and environment manager.
  - You can use conda to create environments for isolating your projects that use different versions of Python and/or different packages.
- Installation
  - For Windows
  - For MacOS
  - For Linux
- Click here if you're interested in the documentation
- To understand how to use anaconda please visit this site
- For environment creation and management please go through this link. Get a gist of the basics required for installation and management of environments.
  - If you prefer videos as a learning resource. This video provides a very good summary for you to follow.



Jupyter Notebooks

- This is a very interactive tool of learning and teaching python programming.
- Good thing is, it comes preinstalled with anaconda. So you don't have to install anything else.
- Example
  - Want to learn about markdown. Visit this site for information regarding formatting your Jupiter notebooks.
    - There are some magic keywords for you to use in the notebooks. I'll leave it for you to explore.

Windows

```
conda create -n tensorflow python=3.5
activate tensorflow
conda install pandas matplotlib jupyter notebook scipy scikit-learn
pip install tensorflow
```

Hello World ¶

Just a simple a hello world program, to make sure we have tensorflow properly installed

In [1]:
```
import tensorflow as tf

hello_constant = tf.constant("Hello World!")

with tf.Session() as sess:
    output = sess.run(hello_constant)
    print(output)
```

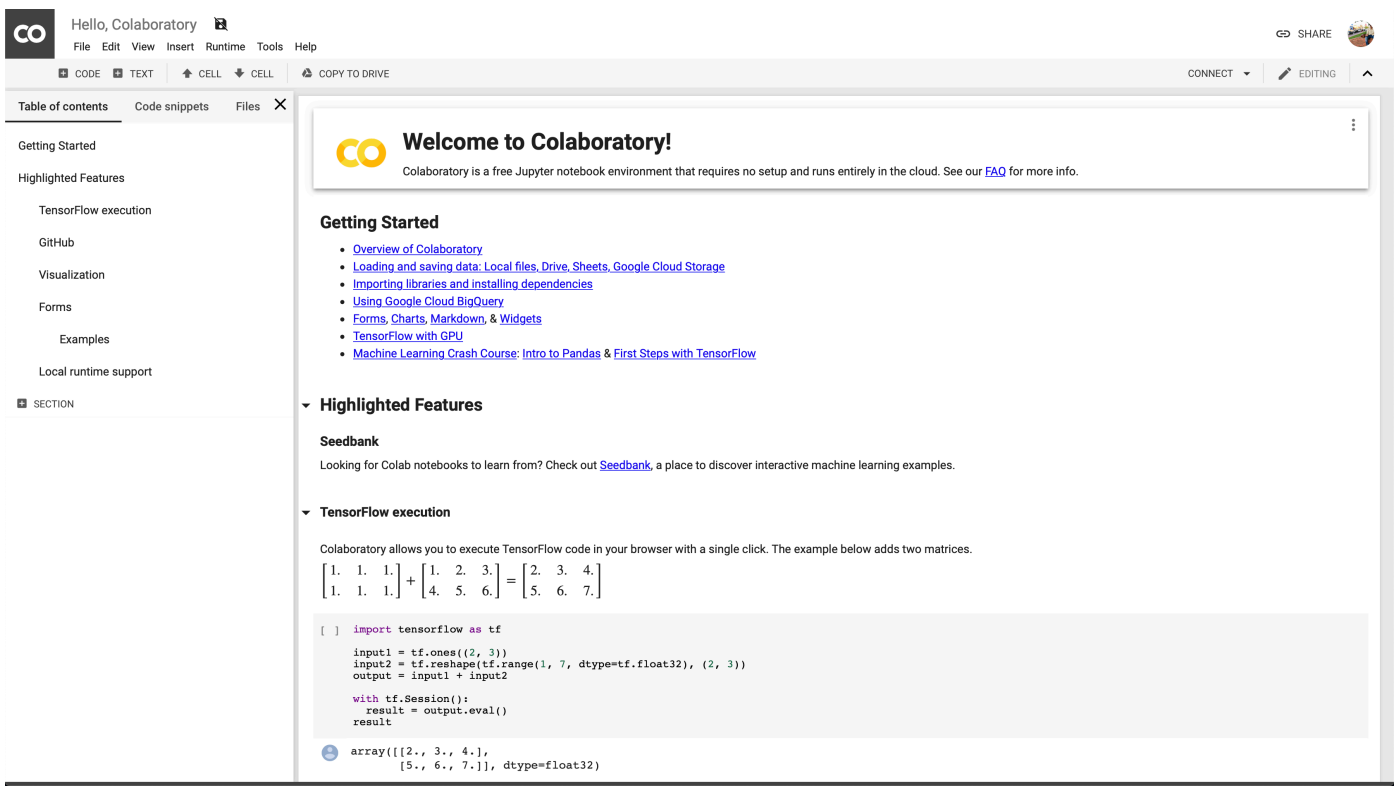
```
/anaconda3/envs/tensorflow/lib/python3.5/importlib/_bootstrap.py:222: RuntimeWarning: compiletime
version 3.6 of module 'tensorflow.python.framework.fast_tensor_util' does not match runtime versio
n 3.5
  return f(*args, **kwds)
b'Hello World!'
```

Awesome!

Great work! Lets move to the next notebook!

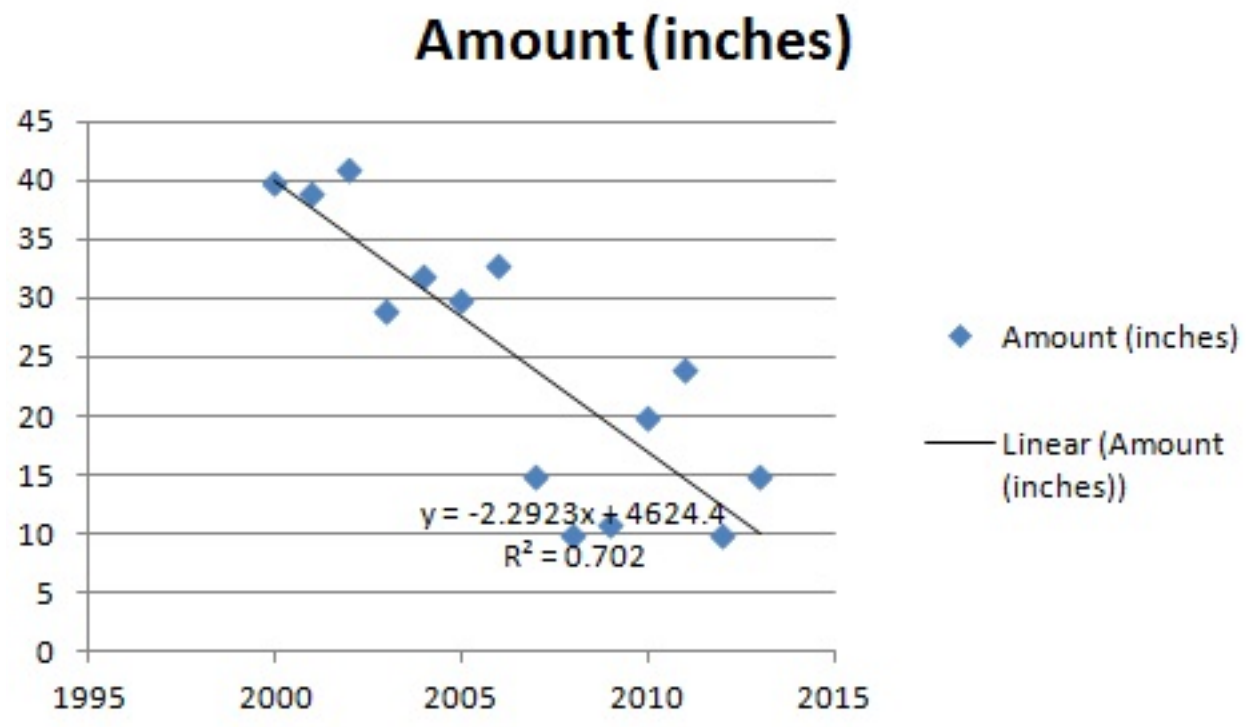
Colaboratory

- Alternatively if you would like to directly work with the jupyter notebooks from your web-browser you can use Google's resource
- https://colab.research.google.com/



Regression

Regression analysis is used in statistics to find relations between the data. More importantly they are used to find how strongly or weekly the data is correlated. For example, relationship between rash driving and number of road accidents by a driver is best studied through regression.



Src : Analytics Vidhya – https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

Regression Analysis: Introduction

In statistics it is important to analyze data that has been presented to you. However, that might not always be the case, if we're just staring at it. For example, look at the following data set.

In [167]:
```
import pandas as pd
from sklearn.linear_model import LinearRegression

# Assign the dataframe to this variable.

bmi_life_data = pd.read_csv("data.csv")
```

In [168]:
```
bmi_life_data.head()
```

Out[168]:

	Country	Life expectancy	BMI
0	Afghanistan	52.8	20.62058
1	Albania	76.8	26.44657
2	Algeria	75.5	24.59620
3	Andorra	84.6	27.63048
4	Angola	56.7	22.25083

In the notebook in the *demo* folder, you'll be working with data on the average life expectancy at birth and the average BMI for males across the world, using **Linear Regression**. The data comes from Gapminder[2]. The data file can be found under the "data.csv" demo folder. It includes three columns, containing the following data:

- Country – The country the person was born in.
- Life expectancy – The average life expectancy at birth for a person in that country.
- BMI – The mean BMI of males in that country.

[2] https://www.gapminder.org/

Building the model

For your linear regression model, you'll be using `scikit-learn`'s `LinearRegression` class. This class provides the function `fit()` to fit the model to your data.

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_values, y_values)
```

In the example above, the model variable is a linear regression model that has been fitted to the data x_values and y_values. Fitting the model means finding the best line that fits the training data. Let's make two predictions using the model's predict() function.

If you get the error:

> RuntimeWarning: numpy.dtype size changed, may indicate binary incompatibility. Expected 96, got 88
> return f(*args, **kwds)

It is a harmless. It is to do with the numpy version. You can either ignore it or go back to `1.14.5`. The issue Will be fixed in the upcoming 1.15.1. You can visit this page if you would like to learn more about it.

On a side note, we need to define our x and y values, that would be used to `train` the model on.

In [170]:
```
# if you look back, we had our data saved in bmi_life_data, we'll be using that for getting the x and the y values
x, y = bmi_life_data[['BMI']], bmi_life_data[['Life expectancy']]
# in python we can just have a comma between variables to assign value to them
```

```
x,y = 1,3
```

In [171]:
```
# for example
a,b = 1,3
print("a :" + str(a) + " b : " + str(b))
```

```
a :1 b : 3
```

In [172]:
```
x.head()
```

Out[172]:

	BMI
0	20.62058
1	26.44657
2	24.59620
3	27.63048
4	22.25083

In [173]:
```
y.head()
```

Out[173]:

	Life expectancy
0	52.8
1	76.8
2	75.5
3	84.6
4	56.7

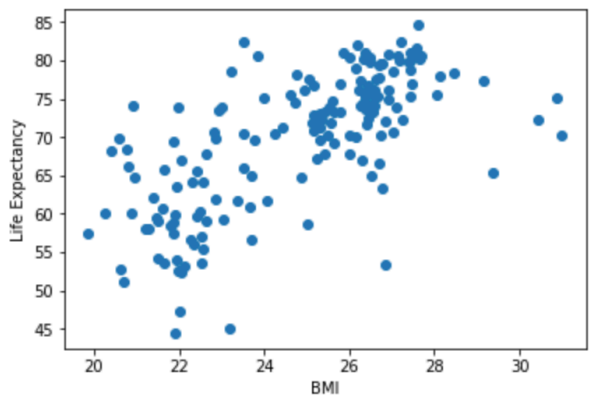
Fitting the model

Once we have set the x and the y values we can use the

model.fit(x_values, y_values)

To fit the model to those values

In [181]:
```
from matplotlib.pyplot import subplots, show
a = x
b = y
fig, ax = subplots()
ax.scatter(a, b)
ax.set_xlabel("BMI")
ax.set_ylabel("Life Expectancy")
show()
```



In [182]:
```
model.fit(x,y)
#Please notice that x and y are float64 data type, and the model.fit() expects x and y as numpy array
# Hence we pass them in '[]' --> model.fit([x],[y])
```

Out[182]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

Prediction

So now based on some other `country x` with the males having a average `BMI` rating of lets say `21.9802`
We can predict the average life expectancy using

model.prdict(Value)

In [183]:
```
lifeExpectancy = model.predict(21.9802)
for _ in lifeExpectancy:
    for j in lifeExpectancy: lifeExpectancy = j
print("Life Expectancy of country X is : " + str(lifeExpectancy[0]) + " years")
```

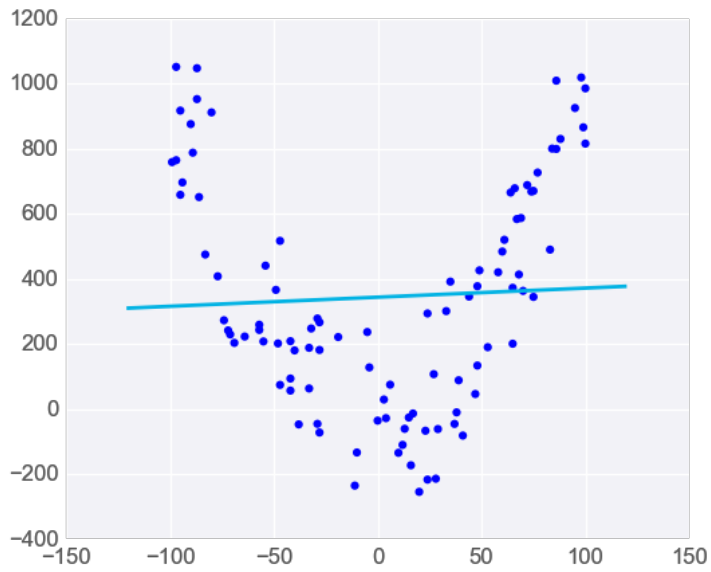
```
Life Expectancy of country X is : 62.58452010568185 years
```

That's awesome. Now you know how to use scikit learn for `linear regression`. Pretty impressive right for the first day at class?

What we saw is known as linear regression. Over time as I add more chapters to this book, I'll cover something known as

Linear regression cons

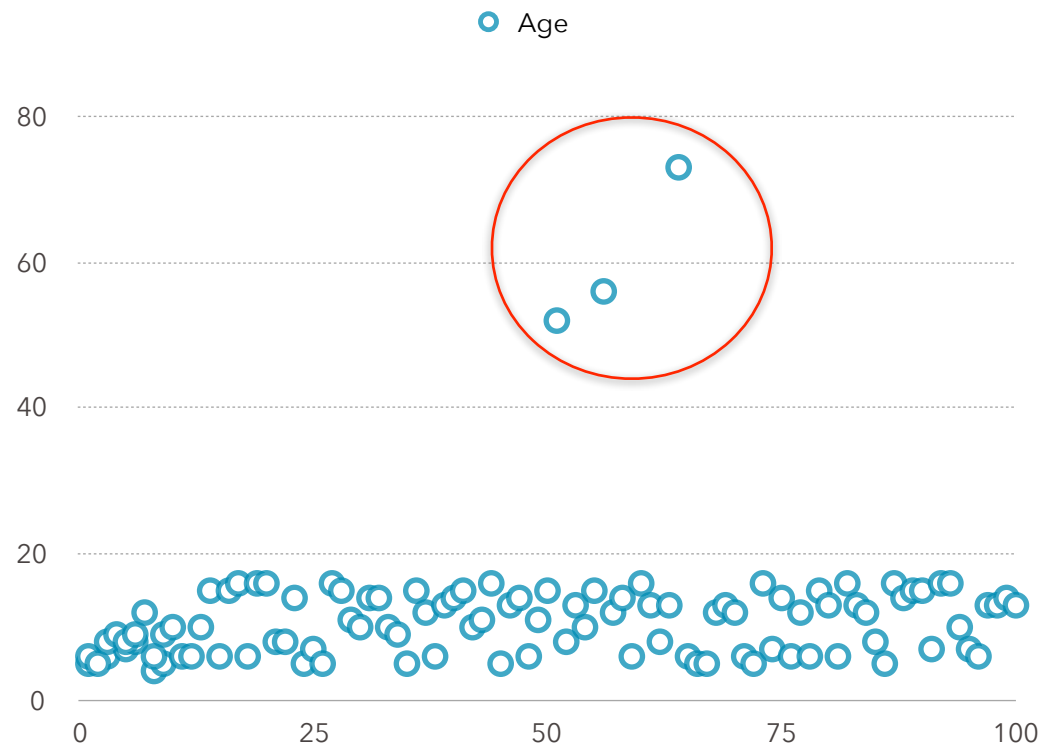
It's not always summer, when it comes to linear regression. As you can see in the diagram below. A single line is not able to fit the model. Hence, we can say that Linear Regression works best when the data provided is linear in nature. To solve this several methods are viable. We can add more features, we can transform the data, or if nothing else works, just use some other model.



Also, Linear Regression is very sensitive to outliers.

What are outliers? I hear you ask.
According to the engineering statistics handbook[3], An *outlier* is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. Before abnormal observations can be singled out, it is necessary to characterize normal observations.

What that means is, for example we have a dataset for the age of individuals in a class for elementary school students. It might be normal to find an age group from 5 to 15. If however, you have a datapoint that shows the age of a student to be 45, that would be an outlier.



However, the same might not be the case in a graduate or phD class. So linear regression takes a toll in such scenarios. We will take a look at such scenarios in the future.

[3] https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm

References

Code and demos

- Neural net in 11 lines **adding the link**
  - Reference to Andrew Trask
- Deep traffic
  - What is it about
- AI Flappy bird
  - Please note that you need to have anaconda installed on your PC for easy installation of dependencies.
    - It is perfectly fine if you don't understand what is going on

Tools

- Anaconda (Version >5.0, python version 3.x)
  - About
    - Anaconda is a distribution of packages built for data science.
    - It comes with conda, a package and environment manager.
    - You can use conda to create environments for isolating your projects that use different versions of Python and/or different packages.
  - Installation
    - For Windows
    - For MacOS
    - For Linux
  - Click here if you're interested in the documentation
  - To understand how to use anaconda please visit this site
  - For environment creation and management please go through this link. Get a gist of the basics required for installation and management of environments.
    - If you prefer videos. This provides a very good summary for you to follow.
- Jupiter Notebooks
  - This is a very interactive tool of learning and teaching python programming.
  - Good thing is, it comes preinstalled with anaconda. So you don't have to install anything else.
  - Example
    - Want to learn about markdown. Visit this site for information regarding formatting your Jupiter notebooks.
      - There are some magic keywords for you to use in the notebooks. I'll leave it for you to explore.

Additional reading before next meet-up

- Read the part 1 of the deep learning book found here
  - It will give you a brief idea about the math and machine learning concepts to get started with deep learning and being comfortable with it.
- You should use this cheat sheet for understanding any math notation.
- If you're uncomfortable with python programming I would recommend this course to get you started with python.
- Interested in deep learning and machine learning? Sci-kit learn is your go to library
- I recommend you to read up a bit on Perceptrons
  - These are the simplest forms of neural networks out there

- Lets *ascent* the mountain of AI space with gradient descent… ¡wait what!
    - A process by which Machine Learning algorithms learn to improve themselves based on the accuracy of their predictions
- Backpropogation
    - The process by which neural networks learn how to improve individual parameters.
- Need a library for scientific computing? Numpy has you covered.
- Tensorflow we'll get there.

If nothing go through these web pages and make sure you bookmark them for easy access.

Online Courses

- Python course
- Numpy, pandas, matplotlib

Books

- Deep learning book
- Neural Networks and DeepLearning
    - Free online book for you to learn more about deep learning

People

Siraj Raval

- School of AI
- LinkedIn
- YouTube
- Twitter
- Github

Jubeen Shah

- Facebook Group
- Meetup
- Instagram
- LinkedIn
- Github