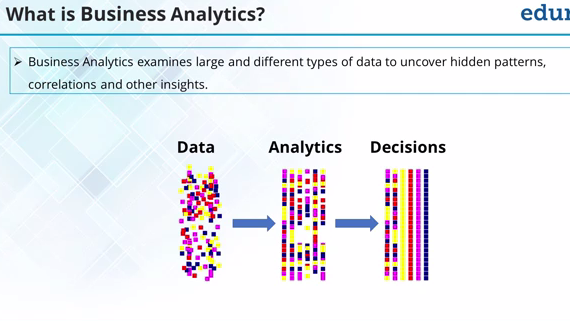
[Python vs R for machine learning](https://datascience.stackexchange.com/questions/326/python-vs-r-for-machine-learning)

Some real important differences to consider when you are choosing **R** or **Python** over one another:

* **Machine Learning** has 2 phases. Model Building and Prediction phase. Typically, model building is performed as a batch process and **predictions are done realtime**. The model building process is a compute intensive process while the prediction happens in a jiffy. Therefore, performance of an algorithm in Python or R doesn't really affect the turn-around time of the user. Python 1, R 1.
* **Production:** The real difference between Python and R comes in being production ready. Python, as such is a full fledged programming language and many organisations use it in their production systems. R is a statistical programming software favoured by many academia and due to the rise in data science and availability of libraries and being open source, the industry has started using R. Many of these organisations have their production systems either in Java, C++, C#, Python etc. So, ideally they would like to have the **prediction system** in the same language to reduce the latency and maintenance issues. Python 2, R 1.
* **Libraries:** Both the languages have enormous and reliable libraries. R has over 5000 libraries catering to many domains while Python has some incredible packages like **Pandas, NumPy, SciPy, Scikit Learn, Matplotlib**. Python 3, R 2.
* **Development:** Both the language are interpreted languages. Many say that python is easy to learn, it's almost like reading english (to put it on a lighter note) but R requires more initial studying effort. Also, both of them have good IDEs (Spyder etc for Python and RStudio for R). Python 4, R 2.
* **Speed:** R software initially had problems with large computations (say, like nxn matrix multiplications). But, this issue is addressed with the introduction of R by Revolution Analytics. They have re-written computation intensive operations in C which is blazingly fast. Python being a high level language is relatively slow. Python 4, R 3.
* **Visualizations:** In data science, we frequently tend to plot data to showcase patterns to users. Therefore, visualisations become an important criteria in choosing a software and R completely kills Python in this regard. Thanks to Hadley Wickham for an incredible ggplot2 package. R wins hands down. Python 4, R 4.
* **Dealing with Big Data:** One of the constraints of R is it stores the data in system memory (RAM). So, RAM capacity becomes a constraint when you are handling Big Data. Python does well, but I would say, as both R and Python have HDFS connectors, leveraging Hadoop infrastructure would give substantial performance improvement. So, Python 5, R 5.

So, both the languages are equally good. Therefore, depending upon your domain and the place you work, you have to smartly choose the right language. The technology world usually prefers using a single language. Business users (marketing analytics, retail analytics) usually go with statistical programming languages like R, since they frequently do quick prototyping and build visualisations (which is faster done in R than Python).

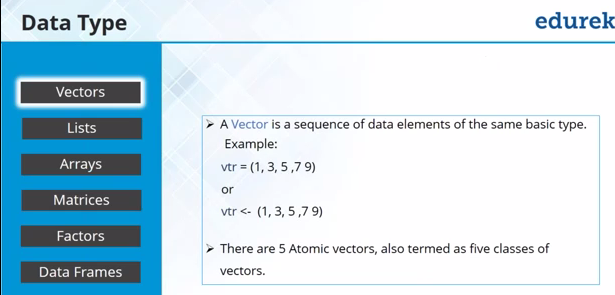


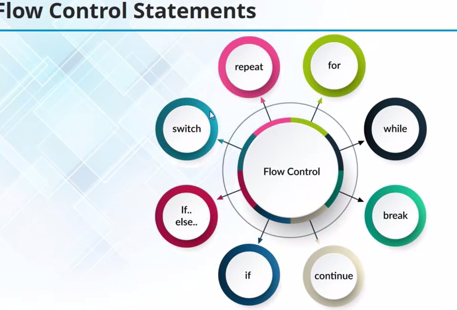


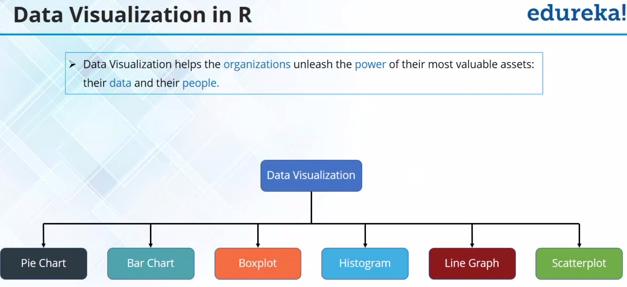
Why learn R ?

1. The style of coding is quite easy.
2. It’s open source. No need to pay any subscription charges.
3. Availability of instant access to over 7800 packages customized for various computation tasks.
4. The community support is overwhelming. There are numerous forums to help you out.
5. Get high performance computing experience ( require packages)
6. One of highly sought skill by analytics and data science companies.









### Useful R Packages

Out of ~7800 packages listed on [CRAN](https://cran.r-project.org/), I’ve listed some of the most powerful and commonly used packages in predictive modeling in this article. Since, I’ve already explained the method of installing packages, you can go ahead and install them now. Sooner or later you’ll need them.

**Importing Data:** R offers wide range of packages for importing data available in any format such as .txt, .csv, .json, .sql etc. To import large files of data quickly, it is advisable to install and use data.table, readr, RMySQL, sqldf, jsonlite.

**Data Visualization:** R has in built plotting commands as well. They are good to create simple graphs. But, becomes complex when it comes to creating advanced graphics. Hence, you should install ggplot2.

**Data Manipulation:** R has a fantastic collection of packages for data manipulation. These packages allows you to do basic & advanced computations quickly. These packages are dplyr, plyr, tidyr, lubridate, stringr. Check out this [complete tutorial](https://www.analyticsvidhya.com/blog/2015/12/faster-data-manipulation-7-packages/) on data manipulation packages in R.

**Modeling / Machine Learning:** For modeling, caret package in R is powerful enough to cater to every need for creating machine learning model. However, you can install packages algorithms wise such as randomForest, rpart, gbm etc

***Note:*** I’ve only mentioned the commonly used packages. You might like to check this interesting *[infographic](https://www.analyticsvidhya.com/blog/2015/08/list-r-packages-data-analysis/)* on complete list of useful R packages.

Till here, you became familiar with the basic work style in R and its associated components. From next section, we’ll begin with predictive modeling. But before you proceed. I want you to practice, what you’ve learnt till here.

## 1. Linear Regression

It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a \*X + b.

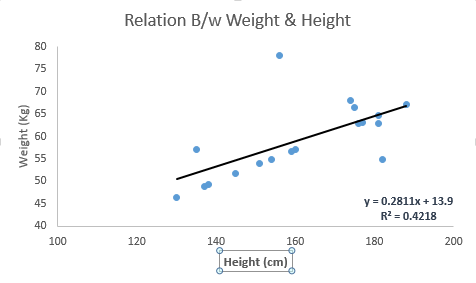
The best way to understand linear regression is to relive this experience of childhood. Let us say, you ask a child in fifth grade to arrange people in his class by increasing order of weight, without asking them their weights! What do you think the child will do? He / she would likely look (visually analyze) at the height and build of people and arrange them using a combination of these visible parameters. This is linear regression in real life! The child has actually figured out that height and build would be correlated to the weight by a relationship, which looks like the equation above.

In this equation:

* Y – Dependent Variable
* a – Slope
* X – Independent variable
* b – Intercept

These coefficients a and b are derived based on minimizing the sum of squared difference of distance between data points and regression line.

Look at the below example. Here we have identified the best fit line having linear equation **y=0.2811x+13.9**. Now using this equation, we can find the weight, knowing the height of a person.

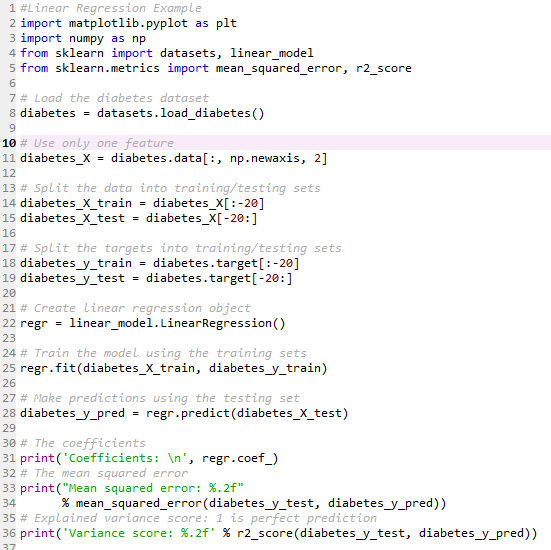
[](https://www.analyticsvidhya.com/wp-content/uploads/2015/08/Linear_Regression.png)

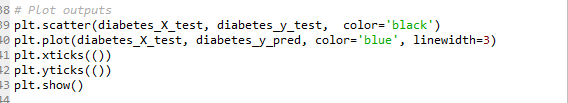
Linear Regression is of mainly two types: Simple Linear Regression and Multiple Linear Regression. Simple Linear Regression is characterized by one independent variable. And, Multiple Linear Regression(as the name suggests) is characterized by multiple (more than 1) independent variables. While finding best fit line, you can fit a polynomial or curvilinear regression. And these are known as polynomial or curvilinear regression.

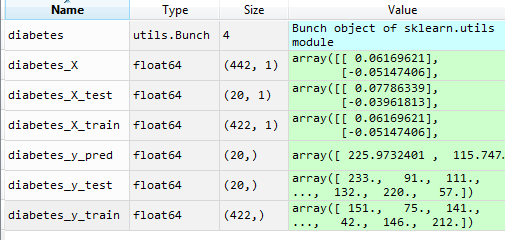
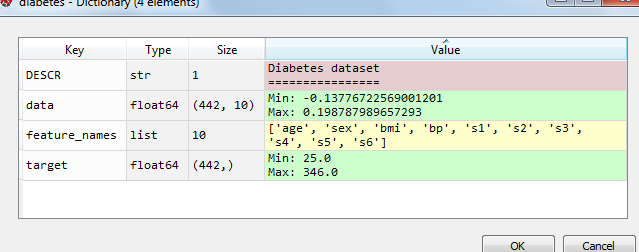
# Linear Regression Example

This example uses the only the first feature of the diabetes dataset, in order to illustrate a two-dimensional plot of this regression technique. The straight line can be seen in the plot, showing how linear regression attempts to draw a straight line that will best minimize the residual sum of squares between the observed responses in the dataset, and the responses predicted by the linear approximation.

The coefficients, the residual sum of squares and the variance score are also calculated.







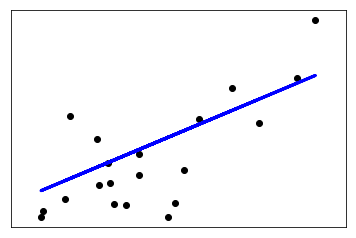
runfile('D:/Machine Learning/Projects/PythonFundamentals.py', wdir='D:/Machine Learning/Projects')

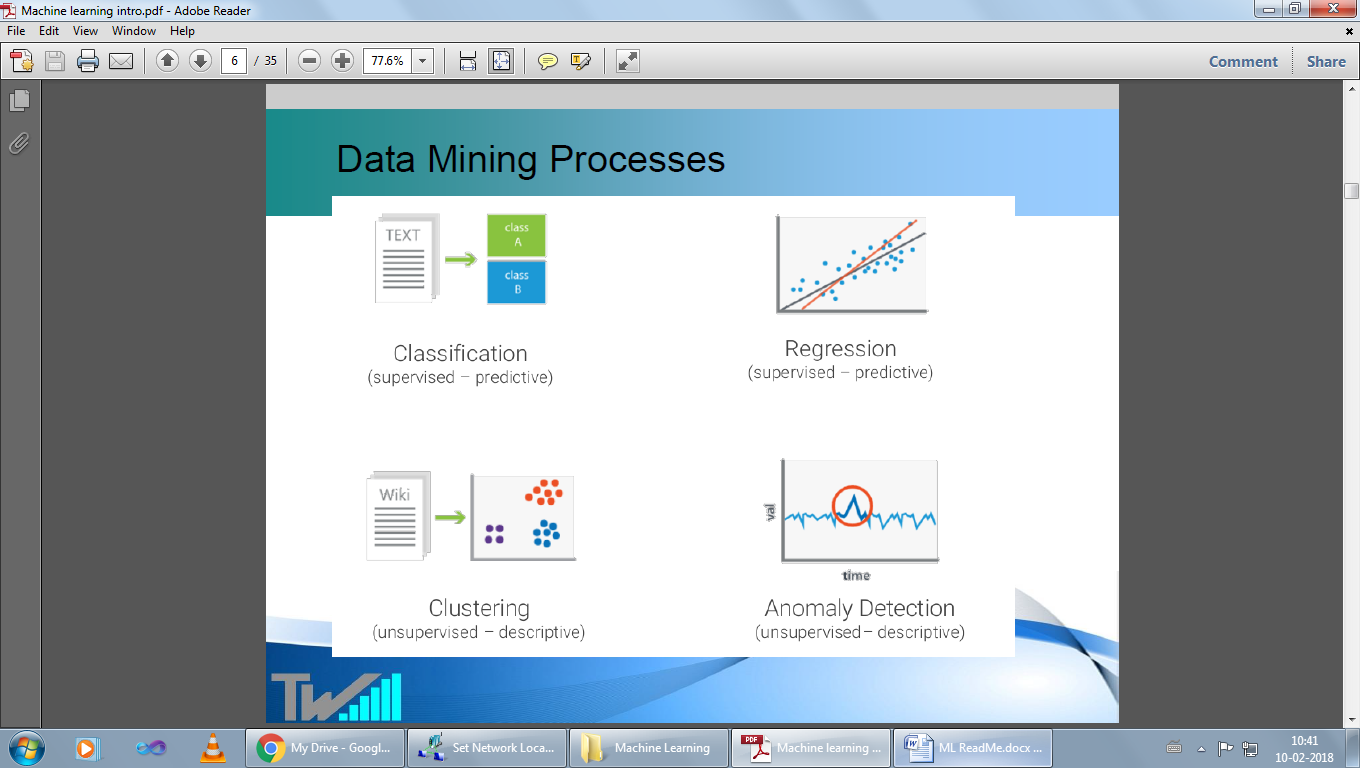
Coefficients:

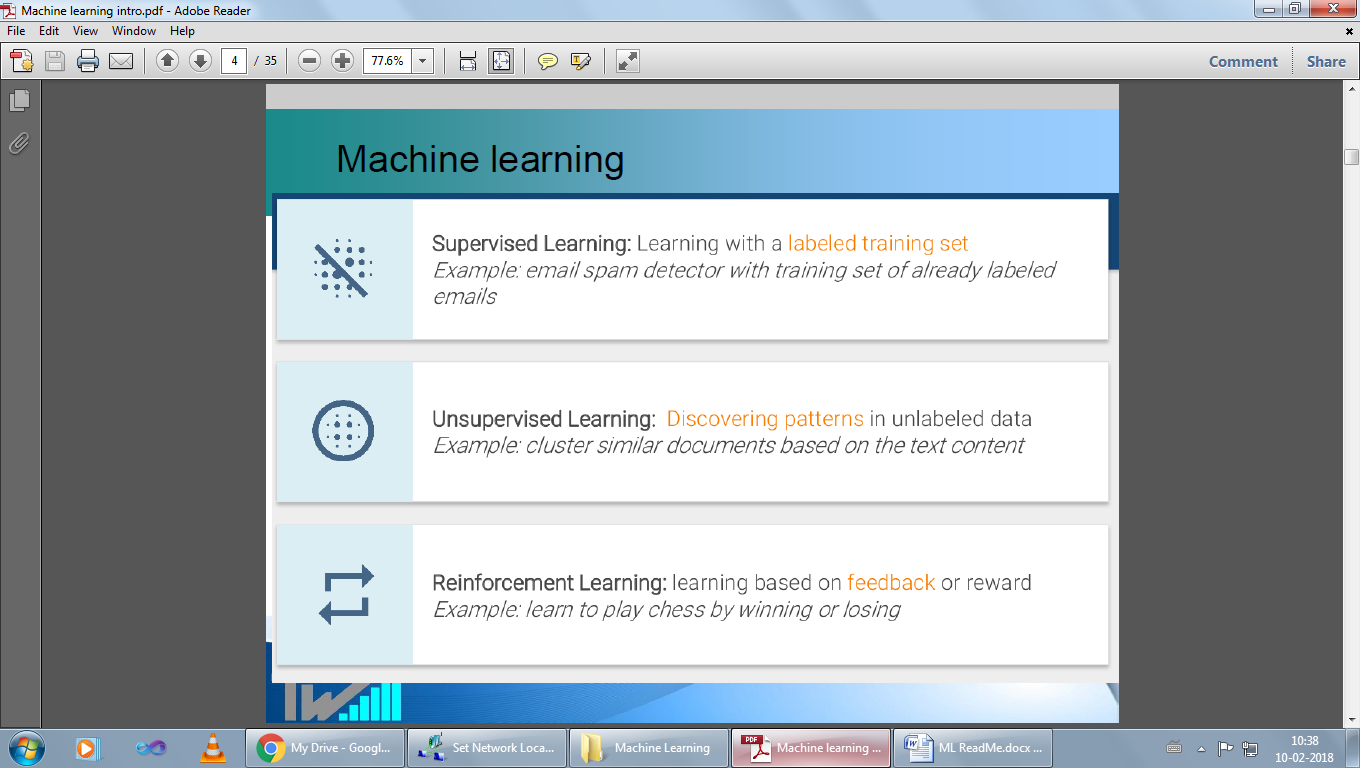
[ 938.23786125]

Mean squared error: 2548.07

Variance score: 0.47







Regression –for continuous values

Classification –for discrete values ( classes)

Binary classification examples

Email spam/Not Spam

Gaming Win/Loss

Sales Buying/Not Buying

Loans –Default vs Non Default

Fraud identification –Fraud vs Non Fraud