Contents

[25-07-2019 3](#_Toc15370762)

[15-07-2019 3](#_Toc15370763)

[URL’s 3](#_Toc15370764)

[Collection of Machine Learning repositories in GitHub 3](#_Toc15370765)

[Math 3](#_Toc15370766)

[Theory 3](#_Toc15370767)

[Supervised Learning 3](#_Toc15370768)

[K-Nearest neighbor (KNN) 3](#_Toc15370769)

[Decision tree 4](#_Toc15370770)

[Naïve Bayes 4](#_Toc15370771)

[Logical regression 4](#_Toc15370772)

[Unsupervised 4](#_Toc15370773)

[Artificial neural networks (ANN) 4](#_Toc15370774)

[Deep learning 4](#_Toc15370775)

[Year 2016 4](#_Toc15370776)

[Understanding entire cycle 4](#_Toc15370777)

[Tools 5](#_Toc15370778)

[More details 5](#_Toc15370779)

[Types of machine learning 5](#_Toc15370780)

[Linear Algebra 5](#_Toc15370781)

[Statistics for ML 6](#_Toc15370782)

[Measurement of variability 6](#_Toc15370783)

[How to calculate IQR 7](#_Toc15370784)

[Variance and standard deviation 8](#_Toc15370785)

[Z-Score 9](#_Toc15370786)

[Frequency table 10](#_Toc15370787)

[Contingency table 10](#_Toc15370788)

[Scatter plot 11](#_Toc15370789)

[Probability 12](#_Toc15370790)

# 25-07-2019

Data sources

<http://archive.ics.uci.edu/ml/index.php>

<https://www.kaggle.com/datasets>

<https://www.data.gov/>

# 15-07-2019

Python appears best for ML. There are several algorithms in several libraries in Python. For typical cases you need to select such library and algorithm and apply it on data. Most of such cases are trivial in nature. That’s it.

## URL’s

High level tutorial: <http://www.tutorialspoint.com/machine_learning>

Low level tutorial: <https://www.tutorialspoint.com/machine_learning_with_python/>

## Collection of Machine Learning repositories in GitHub

<https://github.com/ujjwalkarn/Machine-Learning-Tutorials>

<https://medium.com/machine-learning-in-practice/over-200-of-the-best-machine-learning-nlp-and-python-tutorials-2018-edition-dd8cf53cb7dc>

## Math

<https://www.analyticsvidhya.com/blog/2017/05/comprehensive-guide-to-linear-algebra/>

## Theory

Machine learning can be Supervised (data and pattern given), Non-supervised (Only data given, machine finds out pattern), Reinforced (result is correct or incorrect information is given to machine), Deep learning and reinforced deep learning.

## Supervised Learning

You have leverage of giving more data with results and check. Thereafter you give data and machine give probable results. Broadly this is of two types.1) regression: where there is some linear relationship and 2) classified: where data lies in some patters or patches or categories. Say for example cancer or no cancer cells. You plot them, they both cover different areas. So data in one area corresponds to cancer and in other, no cancer. So by seeing the new data and its area the machine can say cancer or no cancer.

This is for small set of data where you can train machine from data and based on that training of machine you can get results on new data given by you. In short machine can predict Y if X is given. In supervised learning There are results and you feed the result to model to get better outcome. For example, you traced spam mails and found the words in it. Then you apply the model on some mail and found out if that mail is spam. You fed the mail to model telling it that it is spam. The model learned from outcome. Goal is to predict the output from input.

Following algorithms are there for Supervised Learning.

### K-Nearest neighbor (KNN)

This is used for classification of objects. Say, there are some objects in the machine, and they are plotted or classified as red, green and blue on a canvas. Now if unknown data is given to the machine, its is first potted to the same canvas. Then its average distance from all other points is calculated and found out which color neighbors are in most vicinity. If the unknown point has mostly blue points near it, then it is classified as blue. Most ML libraries provide KNN.

### Decision tree

### Naïve Bayes

Say in a basket there are many fruits. Based on color, diameter and shape you can say it’s an apple. You provide training data for various fruits, then you give an unknown fruit and machine will tell probably which fruit it is based on training data.

### Logical regression

Say many data points are plotted. You can draw a line separating two sets of data points. Now unknown data point you can classify based on which side of line it lies.

One python toolset is scikit-learn. Several ready-to-use algorithms are available.

## Unsupervised

Entire information is derived on provided data. There is no scope to provide new data. Machine infers based on huge provided data and shows the interesting results. There is no feedback.

If data is huge say bigdata in gigabytes or terabytes, you need machine to analyze the data and tell you what distinct categories or patterns there in data are. For example, Voters data X. Each voter has certain features (AI terminology) such as age, geography, education area, interests etc. You can ask machine what distinct age group exists? What education or geography exists? In which geography which age group is more, or education is more and so on. Suppose or requirement is We are provided voters data and whether they can switch there vote in favor of somebody if certain condition is fulfilled for them. This data can be provided to a clustering algorithm which creates clusters based on similar conditions of vote switching. In one cluster similar conditions are there. Now for each cluster a message for voter can be associated and election campaign can be proceeded with such messages. One such algorithm is K-means. This finds out unique ‘K’ such clusters and center of a cluster is mean of all values.

## Artificial neural networks (ANN)

Idea comes from neural network of human brains. Several architectures are developed. For a specific requirement you need to use one architecture. Deep learning uses ANN.

## Deep learning

# Year 2016

## Understanding entire cycle

Problem Definition 🡪 Data Collection 🡪 Data Exploration and preparation 🡪 Model Development 🡪 Model Deployment.

Define problem like who are prospective buyers, what items are likely to sell, which geographic area has more prospects, how do I sell more products to existing customer and so on.

Take sales data of one or many store, take data from customer.

Data is messy. Prepare it and shape it and put missing values. This is data wrangling.

Choose machine learning algorithm and prepare a model. Feed a subset of data to model to train the model. Test the model with other subsets and rest of data and evaluate it.

If you want to reuse the model then deploy it.

Machine learning is “learn by experience”. Machine learns from data. This is also called predictive analysis. Get data, analyze, put it to algorithm and get the predictive analysis. Must know the domain. More data better predictions. Data quality should be good.

## Tools

R Studio, ML Studio

## More details

Machine Learning is “Learn from data” and not from programming. Output is predictions.

Example:

Problem: Suppose you have a large dataset of age, gender, education and income. The problem is find out the income if others parameters are given.

Solution: Select a generic algorithm and create a model with age, gender, education as input and income as output (prediction). Train the model with data. A trained model can do prediction of income once other parameters are supplied. There is no programming. The machine learnt from data. This must be iterative. At first train with small subset of data and then validate with another set of data. The algorithm actually discovers relationships between various parameters. The algorithm learnt from data and got trained.

In above example there is a target variable and its value is available. This type of machine learning is called “**Supervised machine learning**”. Machine learning without a target variable is unsupervised machine learning. Recommendations is another type of machine learning.

## Types of machine learning

#### Classifications

Output is a classified value. Spam detection. Input is email out is “spam” “not spam”.

#### Regression

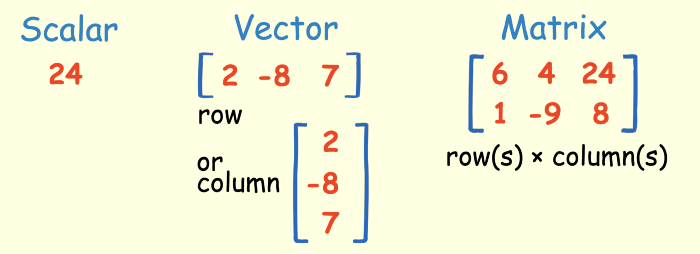
Output is continuous value. Property prices. Input is locality, price, built up area, type such as residence / office etc. and output is property price.

#### Clustering

Input is varied data and output is groups.

# Linear Algebra

It is math of matrices and vectors. Many problems in business and technology can be expressed in terms of vectors and matrices.



Learn about matrix operations: Add, Subtract, multiply, Inverse and transpose. Matrix and vectors are used to solve multiple equations having many variables.

<https://towardsdatascience.com/linear-algebra-for-deep-learning-f21d7e7d7f23>

# Statistics for ML

<http://makemeanalyst.com/basic-statistics-for-data-analysis>

**Mean**: (Sum of all values) / count

**Median**: It is middle / central value of data. Say sample is: 1, 5, 4, 3, 6 -> 1, 3, 4, 5, 6 = Median is 4. If there are two values then average of two values is median, in case of even number of counts. Remember, to calculate median, it is not (max – min ) / 2. It is central value of distribution. Basically, you take out the **middle position** of a distribution, if more than two then take mean of two. It is not the arithmetic middle but the positional middle. Say for distribution 2, 5, 7, 9, 22, 100, 200, 1000, 10,000, the arithmetic middle is (10000 – 2) /2 which is not median. Positional middle is 22 because on right of 22 there are 4 items and on left also 4 items. Hence median is 22. The value of an item is only used to make placement of that item and not to calculate the median.

**Mode**: It is most frequent value. 1,2, 3, 3, 3, 4, 5. Mode is 3.

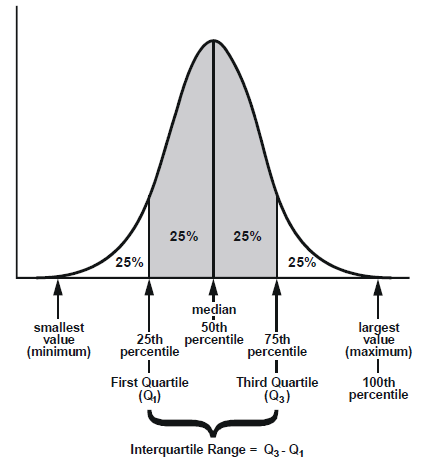
Mean, Median and Mode, three M’s are also called central tendency of a distribution.

## Measurement of variability

Mean, Median and Mode of two distributions can be same, but the distributions may be quite different. Range, Interquartile range (IQR) and Box plot shows the variability of a distribution.

**Range**: Difference between highest and lowest value.

**IQR**: It equally divides the distribution in 4 equal parts called quartiles, 25% each.



IQR is Q3 – Q1. IRQ shows the spread of a distribution in a range. Q2 is middle, thus the Median. IQR is used to find out how spread-out the values are. This is a numerical value.

Statistics assumes that your values are clustered around some central value. IQR tells how your middle values are ‘spread out’. It also tells which values are too far from middle values. These too far values are called outliers, because they lie beyond the expected range.

### ****How to calculate IQR****

Step 1: Order from low to high

Step 2: Find the median or in other words Q2

Step 3: Find Q1 by looking the median of the left side of Q2

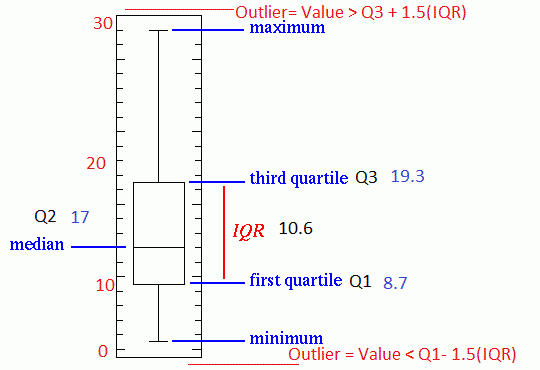
Steps 4: Find Q3 by looking the median of the right of Q2

Steps 5: Subtract Q1 from Q3 to get IQR.

IQR concentrates on central distribution of values, where generally most of the values lie.

**Box Plot (Box and whisker plot)**: A graph which explains the center and variability of a distribution. It is also helpful in detection of outliers.

Below is sample box plot



Length of the box is IQR. Outliers are those values which are more than 1.5IQR from either end of box. Outliers are values which we don’t expect. Statistics say that 1 percent of data may be outlier. To calculate outlier, get IQR and outliers are Q1 – 1.5IQR or Q3 + 1.5IQR

The box in the box plot shows the middle spread of your distribution.

# Variance and standard deviation

They are also variability of a distribution. A population is entire population or say total distribution. Sample is a small set taken from population. The formulas for variance and standard deviation are bit different for **sample** and **population**.

Standard deviation is sigma **σ,** It is how spread out numbers are. This is square root of variance.

Variance = **Average** of **Squared** difference from mean

Example

600mm, 470mm, 170mm, 430mm and 300mm

Step 1: Get mean

(600 + 470 + 170 + 430 + 300) / 5 = 394

Step 2: Square the difference from mean, Add them, divide by total number, this gets you variance.

Variance = (SQR(600 – 394) + SQR(470 – 394) + SQR(170 – 394) + SQR( 430 – 394) + SQR(300 – 394)) / 5

= (42436 + 5776 + 50176 + 1296 + 8836) / 5 = **21704**

**σ = SQRT(21704 ) = 147**

SD is useful. It is interesting to see how many values are within one SD from mean.

The above example is for population. For sample the N should be N -1, instead of 5 it should be 4.

|  |  |  |
| --- | --- | --- |
| The "**Population** Standard Deviation": |  | square root of [ (1/N) times Sigma i=1 to N of (xi - mu)^2 ] |
| The "**Sample** Standard Deviation**":** |  | square root of [ (1/(N-1)) times Sigma i=1 to N of (xi - xbar)^2 ] |

Why square? Otherwise negative and positive will cancel. It has been found useful if done in that way of doing square then square root.

Larger the SD more the variability of data.

# Z-Score

Also called standardized score. It is how much Standard deviation an element is from the mean. That means how much difference is between the element’s value and mean value in terms of standard deviation.

Z-score = ((value of element) – mean) / (Standard deviation)

Why Z-score is useful?

Say there are two populations of exams Maths and Chemistry. Two persons A and B sat for above two exams. A scored 85 and B scored 75, who performed well?

In Maths Mean = 80, SD = 10

In Chemistry Mean = 60, SD = 5

Z- score (A) = (85 – 80) / 10 = 1.5

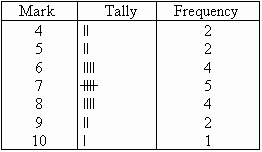
Z-score (B) = (75 – 60) / 5 = 3

So, B has performed better than A instead of A scoring more marks.

# Frequency table

This is number of times a value occurs in a distribution.

is created on single variable. The table has two columns 1) Element value 2) No of times it occurred. Example:



Remember that frequency table is count.

# Contingency table

This is the relationship between two variables. We can say that frequency tables of two variables shown simultaneously is contingency table. This is sort of crosstab, a crosstab means rows, columns and some sort of aggregating information in cells. An individual cell is aggregated result of multiple cells. In case of contingency table, the aggregation function used is count.

To create a contingency table These are steps:

Say for example two variables player type and height. Player type can be Basketball (B) and Football (F). Height can be tall (Y) and short (N). Here is the data:

Type Height

B Y

F N

B N

B Y

F N

B Y

F Y

F N

F N

B Y

B Y

F N

F N

|  |  |  |
| --- | --- | --- |
|  | Y | N |
| B | ~~IIII~~ | I |
| F | I | ~~IIII I~~ |

=

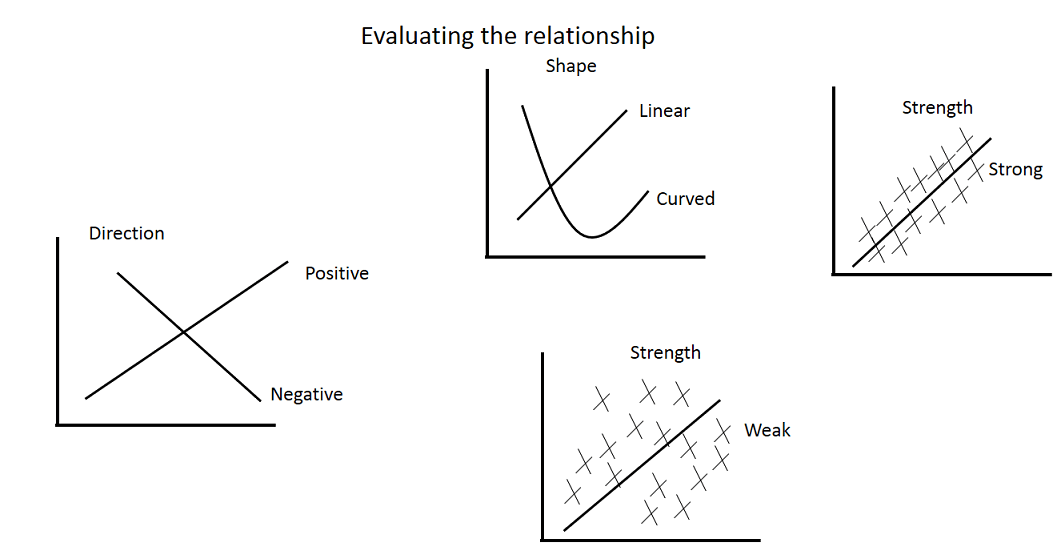
|  |  |  |  |
| --- | --- | --- | --- |
|  | Y | N | Total |
| B | 5 | 1 | 6 |
| F | 1 | 6 | 7 |
| Total | 6 | 7 | 13 |

The right-hand side is contingency table.

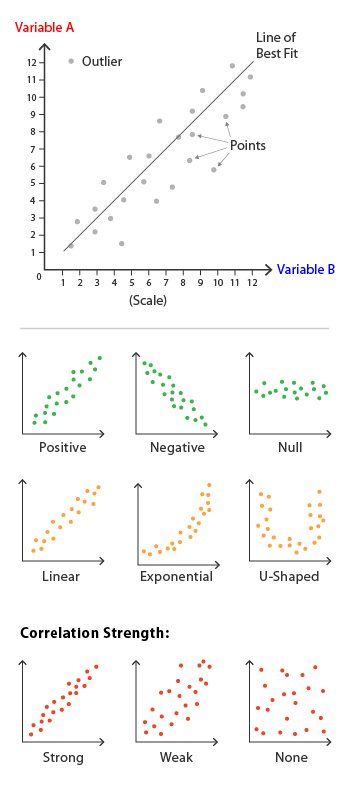
Basically, if you want to find out whether two variables have any relation, you create a contingency table for two variables. Then you run chi square test on that table. A high value shows relation and low values show no relation. If chi-square test is positive, then it is worth doing research in data otherwise it is not worth. Chi-square test is somewhat complex.

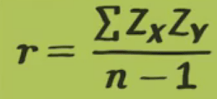
# Scatter plot

Contingency table is not for quantitative values. Scatter plot is for that. Independent variable is plotted at X-axis and dependent variable is plotted against Y-axis as custom.



In scatter plot points are plotted in cartesian co-ordinate denoting two variables. If you show the points by color / shape or size, then you have used a third variable. You can draw a line of best fit or trend line which shows the correlation. Using the trend line, you can estimate via interpolation. Too far points from trend line are called outliners.



The Pearson Correlation (r) is important measure as single number to find if there is a weak or strong co-relation and whether it is positive or negative. R’s value lies between -1 and 1. A (-1) means strong negative co-relation and vice-versa. 

# Linear regression

Relationships between two variables can be deterministic or statistical. Deterministic relationship is that where one variable can be calculated from other say, Fahrenheit and Celsius. Statistical relationship is like between weight and height where no accurate results are there but based on data predictions can be made.

The core idea is to find best fit line, which is the regression line. Error is the distance of a point from regression line.

# Probability

Probability lies between 0 and 1.

# Book Elements of statistical learning: Tough - left

Synonyms: Predictors or independent variables. Responses or dependent variables.

The outputs of a prediction task can be quantitative, we call it regression, if qualitative, we call it classification. The qualitative variables are mostly represented by numerical codes or also called as targets.

# Book Think stats

## Setup

Remove all Python installations. Download Anaconda and install it. It has everything you need.

Type **juypter notebook** at command prompt, it will start a server,you can browse at 8888 and open notebooks, create notebooks. In notebook you have many cells. You can run code in individual cell and check output.

## Brush up of Python

* Variable number of positional parameters to a function by \*

def myTest(\*args):

for a in args:

print(a)

myTest(1,2,3,4,5)

output

1 2 3 4 5

* Variable number of keyword arguments to a function by \*\*

def myTest(\*\*kargs):

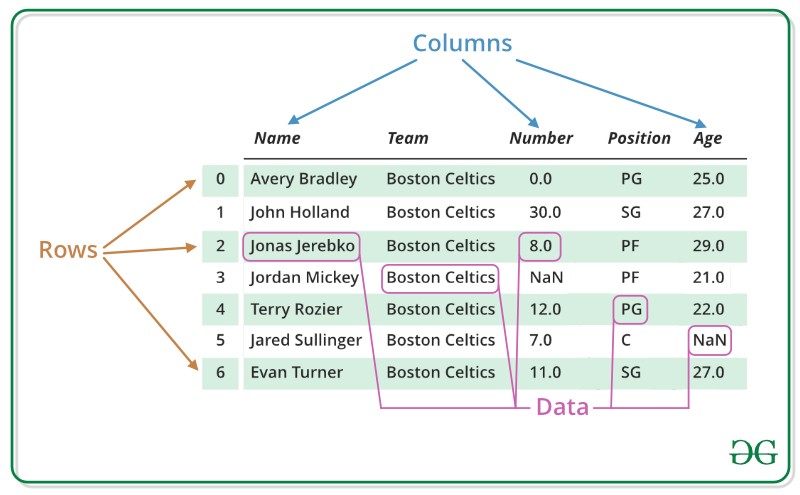
for a in kargs:

print(kargs[a])

myTest(x=1,y=2)

1 2

* dict(x=1,y=2) function creates a dictionary.
* Panda dataframe. It’s a two-dimensional data



* import pandas as pd

df = pd.DataFrame([1,2,3,4,5,6,7,8,9,11])

df.head()

head() method prints top 5 data

You can work on individual rows or columns.

* Dataframe can have millions of rows of data, only computer memory is limitation. I connected pandas with csv and with sql anywhere database. By giving raw sql command I was able to get results of sql in dataframe. Remember to give host=’kushserver’ in connection string if server is on a different computer.
* I connected to SQL anywhere and fetched SQL query data directly in dataframe. Multi years dataframes were concatenated successfully. Pivoting was also done.

import pandas as pd

import sqlanydb

from string import Template

mydf1 = pd.read\_csv('mydata/test.txt')

sql = '''select item, brand, model, qty, price, "date"

from bill\_memo key join bill\_memo\_product

key join inv\_main

key join product

order by "date" '''

def getconn(db):

return sqlanydb.connect(uid='dba', pwd='sql', eng='server', dbn=db, host='kushserver' )

df1 = pd.read\_sql\_query(sql, getconn('capi2016'))

df1['month'] = pd.to\_datetime(df1['date']).dt.strftime('%Y-%m')

df2 = pd.read\_sql\_query(sql, getconn('capi2018'))

df2['month'] = pd.to\_datetime(df2['date']).dt.strftime('%Y-%m')

df3 = pd.read\_sql\_query(sql, getconn('capi2018'))

df3['month'] = pd.to\_datetime(df3['date']).dt.strftime('%Y-%m')

df4 = pd.read\_sql\_query(sql, getconn('capi2019'))

df4['month'] = pd.to\_datetime(df4['date']).dt.strftime('%Y-%m')

df = [df1,df2,df3,df4]

df = pd.concat(df)

p = df.pivot\_table(index='item', columns='month', values='qty', aggfunc=sum, fill\_value=0)

p.head(10000)

Another sample of pivot

import pandas as pd

import sqlanydb

from string import Template

mydf1 = pd.read\_csv('mydata/test.txt')

sql = '''select item, brand, model, qty, price, "date"

from bill\_memo key join bill\_memo\_product

key join inv\_main

key join product

order by "date" '''

def getconn(db):

return sqlanydb.connect(uid='dba', pwd='sql', eng='server', dbn=db, host='kushserver' )

def get\_amount(row):

return float(row['qty']) \* float(row['price'])

pd.options.display.float\_format = '{:,.0f}'.format

dfa = pd.read\_sql\_query(sql, getconn('capi2013'))

dfa['month'] = pd.to\_datetime(dfa['date']).dt.month

dfa['year'] = pd.to\_datetime(dfa['date']).dt.year

dfa['amount'] = dfa.apply(get\_amount, axis=1)

dfb = pd.read\_sql\_query(sql, getconn('capi2014'))

dfb['month'] = pd.to\_datetime(dfb['date']).dt.month

dfb['year'] = pd.to\_datetime(dfb['date']).dt.year

dfb['amount'] = dfb.apply(get\_amount, axis=1)

dfc = pd.read\_sql\_query(sql, getconn('capi2015'))

dfc['month'] = pd.to\_datetime(dfc['date']).dt.month

dfc['year'] = pd.to\_datetime(dfc['date']).dt.year

dfc['amount'] = dfc.apply(get\_amount, axis=1)

df1 = pd.read\_sql\_query(sql, getconn('capi2016'))

df1['month'] = pd.to\_datetime(df1['date']).dt.month

df1['year'] = pd.to\_datetime(df1['date']).dt.year

df1['amount'] = df1.apply(get\_amount, axis=1)

df2 = pd.read\_sql\_query(sql, getconn('capi2017'))

df2['month'] = pd.to\_datetime(df2['date']).dt.strftime('%Y-%m')

df2['year'] = pd.to\_datetime(df2['date']).dt.year

df2['month'] = pd.to\_datetime(df2['date']).dt.month

df2['amount'] = df2.apply(get\_amount, axis=1)

df3 = pd.read\_sql\_query(sql, getconn('capi2018'))

df3['month'] = pd.to\_datetime(df3['date']).dt.strftime('%Y-%m')

df3['year'] = pd.to\_datetime(df3['date']).dt.year

df3['month'] = pd.to\_datetime(df3['date']).dt.month

df3['amount'] = df3.apply(get\_amount, axis=1)

df4 = pd.read\_sql\_query(sql, getconn('capi2019'))

df4['month'] = pd.to\_datetime(df4['date']).dt.strftime('%Y-%m')

df4['year'] = pd.to\_datetime(df4['date']).dt.year

df4['month'] = pd.to\_datetime(df4['date']).dt.month

df4['amount'] = df4.apply(get\_amount, axis=1)

df = [dfa,dfb,dfc,df1,df2,df3,df4]

df = pd.concat(df)

p = df.pivot\_table( index='item', columns='year', values='amount', aggfunc=sum, fill\_value=0)

p.to\_excel('c:/sales\_amount.xlsx')

#p.plot(kind='line', x='item', y='year')

p.head(10000)

Pivot is multiple cells aggregated to single cell. It takes three parameters ***index, columns, values***. All three parameters are column names. Values columns are aggregated for index (rows) and columns.

To show multiple items in row or column in pivot, use format like: (see year and month inside year, Item and brand inside item)

p = df.pivot\_table(index=['item','brand'], columns=['year','month'], values='qty', aggfunc=sum, fill\_value=0)

* Many dataframes can be combined to a single dataframe and pivoted to a new dataframe, as in above code.
* DataFrames can be stored as various file types with different formats inside, like CSV, EXCEL, JSON and many more. Ex: df.to\_excel(‘c:/test.xlsx’)

# Some Calculus

## Product rule

Example:

f(x) = h(x) \* g(x)

f’(x) = (h’(x) \* g(x)) + (h(x) \* g’(x))

<https://www.math.hmc.edu/calculus/tutorials/prodrule/>

Product rule is applicable for product of two functions of x say h(x) \* g(x).

## Chain rule

Chain rule applies for nested functions that is function inside a function.

Example:

f(x) = g(h(x))

Step 1:

Let h(x) = z; => f(x) = g(z)

f’(x) = g’(z) \* h(x)

f(x) = (5 – 6x) ^ 5;

let z = (5-6x);

f(x) = z^5

df/dz = **5z^4**

dz/dx = **-6**

* df/dx = f’(x) = product of above two = -30(5-6x) ^ 4

<https://www.khanacademy.org/math/ap-calculus-ab/ab-differentiation-2-new/ab-3-1a/a>

# Coursera course

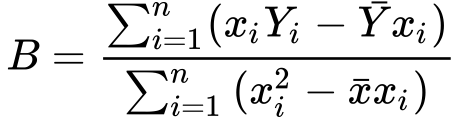
Linear regression in more details.

You can plot two variables, dependent(x) and dependent (y) on x and y axis. If all the points are in a straight line it is a deterministic relationship, otherwise it is a statistical relationship. In such case a best fit line can be drawn for all the lines. This problem of creating best fit line statistically is called linear regression problem. Best fit line is that line which minimizes the error that is square of difference between predicted value and actual value for all points. The formula thus arrived is something like equation of a line in form y = a + bx. In fact, you get the value of a and b by feeding points data to a formula (a computer program). The formula returns a and b for best fit line. Now give a value of x you can get value of y. Getting the formula may require calculus differentiation. You may not go in details of the formula and can apply it.

Programmatically line of best fit is







Explanation is here <https://towardsdatascience.com/linear-regression-derivation-d362ea3884c2>

Check the mean values of x and y with bar at the top.

## Theory

Training data set is scattered on the x-y plane. Purpose is to make a straight line (defined by h\_\theta(x)*hθ*​(*x*)) which passes through these scattered data points in best fit manner. The best possible line will be, that the average squared vertical distances of the scattered points from the line will be the least. This average squared vertical distance is the cost function. Goal is to minimize this cost function.

## Gradient Descent

I did partial differentiation using chaining and arrived at minimization of cost function.

<https://www.coursera.org/learn/machine-learning/supplement/aEN5G/gradient-descent-for-multiple-variables>

* Do feature scaling such that -1 <= value <= +1
* In linear regression we try to fit in straight line. But it is possible also to have some other polynomial equation except of straight line. There are some algorithms which tell you which equation to use for best fit.
* Feature scaling is useful only for iterative methods like Gradient Descent. For direct mathematical calculation of minimization of cost function by calculus and matrix algebra this feature scaling is not required.

Normal equation model of linear regression is good for features less than 10,000. Above that gradient descent method is good. Normal equation deals with matrix, and handling matrix 10,000 \* 10,000 may be costly at time of inverse.

## Points to note

* Remember m= number of training examples / data; n = number of features.
* In octave there is pinv and inv functions. Pinv is pseudo inverse of a matrix. This does the inverse of even non-invertible matrices. In solving cost function problems through matrix algebra sometimes problem comes of non-invertibility. This generally happens when m <= n, that means there are too less training examples. This also happens when there are redundant features, that is when one features is linear function of another feature.

## Matlab

It is math lab. Created online account [capitalch@gmail.com](mailto:capitalch@gmail.com) / S…3

* In matlab you can evaluate mathematical expressions. If ; is not given at the end, the value of expression is displayed.
* Many matlab functions can return multiple outputs inside square brackets []. You can slice and dice your vectors.
* MatLab in short is matrices laboratory.
* Use in place operation by using ‘.’ A dot. Say instead of y=x^2 + 2\*x + 3, you should y=x.^2 + 2\*x + 3, if x is a vector and you want to do plot(x,y). If you don’t do .^2, it will perform matrix operation. The ‘.’ ascertains element wise operation.
* c=v’; means you are calculating transpose of v and storing it into c.
* plot(x,y) plots two vectors x and y. A third input to plot function determines the format of graph. You can customize the plot. You can label x, y axis, put legends, annotations and title of graph.
* Multiple plots can be in same figure. You use command **hold on**
* To create a matrix do [1,2,3;4,5,6;7,8,9]. See the separation by ; for a column.
* You can extract sub matrices from a big matrix.
* Can create matrix with 0’s, 1’s or random values or identity matrix with one line of code.
* You can concatenate two matrices vertically or horizontally. Z=[A,B] horizontal concatenation. Z = [A;B] vertical concatenation.
* Use length(v) and size(A) function to get length of vector or size of matrix.
* A \* B does matrix multiplication.
* Can use reshape function to reshape vector to matrix, change dimension of matrix and many more.
* Many statistical functions are included.
* MatLab contains relational operators, if- elseif – else – end, for k=1:n … end, while balance < 1000 … end, function statements. Anonymous functions and passing function parameters are there.
* Debugging is possible by breakpoints.

## Notes

* They recommend to learn Octave language for prototyping ML programs and them convert it to C, Java, Python or R.

## Octave

Not equal to ~=

And &&

Or ||

Matrix:

X = [ 1 2 3; 4 5 6] # just space separated. No comma required.

X = 1:0.1:2 creates arrow vector starting with 1, interval 0.1 till 2.

X = ones(1,3) 1X3 matrix of one’s

X = zeros(1,3) 1X3 matrix of zeros

W= rand(3,3) 3X3 matrix of randon numbers

I = eye(4) 4X4 identity matrix

pwd current directory

cd change directory

who shows variables in memory

whos also shows size of variables

save hello.mat v saves variable v in file

load x.dat loads x.dat file in memory

clear deletes all variables

A = B + C concatenates B and C matrices

* A \* B is matrix multiplication
* A .\* B is in place element wise multiplication
* A .^ 2 element wise square
* V + 1 adds 1 to all elements of V
* A’ is transpose of A
* plot(t,y) plots graph taking elements from t and y matrices
* Hold on allows one plot over other
* You can save your plots in form of png files through print command
* You can use comma to put multi commands in same line

For loop

For i=1:100, … end;

If elseif else end

Functions can be defined in a file. Call the function (if file is in current folder). The addpath(‘…’) will extend the search folders. Multiple values can be returned from a function.

## Logical regression

In linear regression you plot the points and get best fit line. You get the value of theta in straight line function of y = theta0 + theta1x. Now for each value of x you can predict y which can be any numerical value.

In some cases, the output is either 0 or 1. Say tumor size > 1cm is malignant and vice-versa. So whatever in input, the output is always between 0 and 1. We can also say if output >=0.5 consider it malignant. Function for y’s value always between 0 and 1 for whatever value of x is y = 1/ (1 + e^ -x). The logical regression uses that approach.

## Neural network (23-08-2019)

