Contents

[1. What is data science 8](#_Toc28721386)

[2. Types of data 8](#_Toc28721387)

[Structured Data 8](#_Toc28721388)

[Nominal level data 8](#_Toc28721389)

[Ordinal level data 8](#_Toc28721390)

[Interval level data (Continuous data) 8](#_Toc28721391)

[Ratio level data (Discrete data) 8](#_Toc28721392)

[Semi Structured Data 8](#_Toc28721393)

[Unstructured Data 9](#_Toc28721394)

[3. What is Statistics and it’s concepts 9](#_Toc28721395)

[Collection 9](#_Toc28721396)

[Organizing 9](#_Toc28721397)

[Analyzing/Summarizing 9](#_Toc28721398)

[Population/Universe 9](#_Toc28721399)

[Sample 9](#_Toc28721400)

[Numerical variable analysis 10](#_Toc28721401)

[Measure of central tendency 10](#_Toc28721402)

[Symmetrical Data 11](#_Toc28721403)

[Drawbacks of Measure of central tendency 12](#_Toc28721404)

[Measure of dispersion 12](#_Toc28721405)

[Interpret the data 16](#_Toc28721406)

[4. Python 16](#_Toc28721407)

[Introduction to Python 16](#_Toc28721408)

[How to create the object 17](#_Toc28721409)

[Data Types 17](#_Toc28721410)

[Numbers 17](#_Toc28721411)

[String 17](#_Toc28721412)

[Data structures 18](#_Toc28721413)

[List 18](#_Toc28721414)

[Tuple 19](#_Toc28721415)

[Dictionary 20](#_Toc28721416)

[Sets 20](#_Toc28721417)

[Conditional statements 21](#_Toc28721418)

[if conditional statement 21](#_Toc28721419)

[if-else conditional statement 21](#_Toc28721420)

[nested if 22](#_Toc28721421)

[nested if-else conditional statement (if...elif..else) 22](#_Toc28721422)

[Looping statements 22](#_Toc28721423)

[For Loop 22](#_Toc28721424)

[While Loop 23](#_Toc28721425)

[5. Probability 23](#_Toc28721426)

[Events are of 2 types: 23](#_Toc28721427)

[Simple event 23](#_Toc28721428)

[Rare event 23](#_Toc28721429)

[Classification of Probability (3 Types) 24](#_Toc28721430)

[Relative frequency 24](#_Toc28721431)

[Classic approach 24](#_Toc28721432)

[Subjective approach 24](#_Toc28721433)

[Two rules in Probability 25](#_Toc28721434)

[Addition Rule 25](#_Toc28721435)

[Multiplication rule 25](#_Toc28721436)

[Probability Distribution 25](#_Toc28721437)

[Discrete Distribution (At one particular point) 26](#_Toc28721438)

[Continuous Distribution (In Particular range) 30](#_Toc28721439)

[6. Outlier Detection (Identify Outliers) 33](#_Toc28721440)

[Boxplot (5 point number summary) 33](#_Toc28721441)

[qqplot: quantile-quantile plot 35](#_Toc28721442)

[z-score 36](#_Toc28721443)

[IQR (Inter Quartile Range)widely used 37](#_Toc28721444)

[Outlier Treatment 39](#_Toc28721445)

[IQR method for outlier treatment 39](#_Toc28721446)

[7. To check if data is normally distributed 40](#_Toc28721447)

[dist plot 40](#_Toc28721448)

[qq plot 41](#_Toc28721449)

[z-score 42](#_Toc28721450)

[Jarquebera test (JB test) 43](#_Toc28721451)

[How do we make the data Normally distributed? 44](#_Toc28721452)

[Variable Transformation Technique 44](#_Toc28721453)

[Data standardization (Making the data Normal distributed and MinMaxScaler) 46](#_Toc28721454)

[8. How to check missing values ? 47](#_Toc28721455)

[Missing Value Treatment 48](#_Toc28721456)

[Treat missing Value (for Categorical Variables) 49](#_Toc28721457)

[We treat missing values first then we treat outliers and then check if the data is normally distributed 50](#_Toc28721458)

[9. Sampling Techniques 50](#_Toc28721459)

[Sampling techniques are of 2 types 52](#_Toc28721460)

[Non Probability Method (not used much) 52](#_Toc28721461)

[Judgement sampling 52](#_Toc28721462)

[Convenience sampling 53](#_Toc28721463)

[Quota sampling 53](#_Toc28721464)

[Snowball sampling 53](#_Toc28721465)

[Probability Method (most widely used) 53](#_Toc28721466)

[Simple Random sampling (SRS) or Random sampling method 53](#_Toc28721467)

[Stratified sampling 55](#_Toc28721468)

[Systematics/Sequential sampling 56](#_Toc28721469)

[Cluster sampling 56](#_Toc28721470)

[10. Hypothesis Testing 56](#_Toc28721471)

[Steps/components for hypothesis testing 57](#_Toc28721472)

[Define the hypothesis 57](#_Toc28721473)

[Null hypothesis (, Read "H zero") 57](#_Toc28721474)

[Alternative hypothesis (, read "H one", or Research Hypothesis) 58](#_Toc28721475)

[11. EDA (Exploratory Data Analysis) 59](#_Toc28721476)

[Uni Variate 59](#_Toc28721477)

[Bi Variate 59](#_Toc28721478)

[Multi Variate 59](#_Toc28721479)

[12. Variable Reduction or Dimensionality reduction or Feature Importance 60](#_Toc28721480)

[Variable Reduction Techniques 60](#_Toc28721481)

[Variable reduction technique (wrt continuous variable). 61](#_Toc28721482)

[Classification model 61](#_Toc28721483)

[Regression model 61](#_Toc28721484)

[VIF (Variance influence factor or multi collinearity) 62](#_Toc28721485)

[Information Value 63](#_Toc28721486)

[Weight of events 63](#_Toc28721487)

[Binning 64](#_Toc28721488)

[PCA 65](#_Toc28721489)

[When to apply MinMax scalar vs Standard scalar 65](#_Toc28721490)

[Steps to perform PCA 66](#_Toc28721491)

[Chi-Square 66](#_Toc28721492)

[Pearson correlation ( r ) 67](#_Toc28721493)

[Formula for Chi-square 67](#_Toc28721494)

[Correlation (Pair Wise correlation) 69](#_Toc28721495)

[Pearson correlation (r) 69](#_Toc28721496)

[Excel Formula 70](#_Toc28721497)

[Python Code 71](#_Toc28721498)

[10. Simple Linear Regression 73](#_Toc28721499)

[Assumptions of linear regression 74](#_Toc28721500)

[Excel Formula 74](#_Toc28721501)

[Interception of coefficient 77](#_Toc28721502)

[OLS --> Ordinary Least squared estimation 77](#_Toc28721503)

[Using Python 79](#_Toc28721504)

[11. Multi Linear Regression 81](#_Toc28721505)

[Pairwise Correlation: 82](#_Toc28721506)

[VIF (Variance influence factor or multi collinearity) 82](#_Toc28721507)

[Variable reduction technique (w.r.t. continuous variables) 83](#_Toc28721508)

[Data requirements for multiple linear regression 85](#_Toc28721509)

[Step-wise Regression: Finalize the model (4 parameters) 85](#_Toc28721510)

[F-Static 85](#_Toc28721511)

[Adj. R-squared 86](#_Toc28721512)

[t-Value 86](#_Toc28721513)

[AIC 87](#_Toc28721514)

[Gradient Descent 88](#_Toc28721515)

[12. Regularization 93](#_Toc28721516)

[Ridge Regularization (L2 Regularization) 93](#_Toc28721517)

[Lasso Regularization(L1 Regularization) 94](#_Toc28721518)

[Cross Validation method 94](#_Toc28721519)

[Elastic Regularization 96](#_Toc28721520)

[13. Multiplication Rule in Probability 96](#_Toc28721521)

[13. Classification of Machine Learning 98](#_Toc28721522)

[Supervised Learning - 98](#_Toc28721523)

[Regression Algorithm (Numerical data, 3 Types) 98](#_Toc28721524)

[Classification Algorithm (Categorical data, 10 Types) 99](#_Toc28721525)

[Unsupervised Learning 99](#_Toc28721526)

[Reinforcement Learning 99](#_Toc28721527)

[14. Naïve Bayes 99](#_Toc28721528)

[Dummy Coding/One hot encoding 100](#_Toc28721529)

[Label Encoding 101](#_Toc28721530)

[Confusion Matrix or classification Matrix 104](#_Toc28721531)

[Model Metrices 104](#_Toc28721532)

[ROC 106](#_Toc28721533)

[15. Clustering Algorithms (K-Means) 107](#_Toc28721534)

[Euclidian distance 109](#_Toc28721535)

[Manhattan distance 109](#_Toc28721536)

[K-Means 110](#_Toc28721537)

[Minimum or maximum linkage method 110](#_Toc28721538)

[Cluster Schema 111](#_Toc28721539)

[16. Decision Tree 115](#_Toc28721540)

[Rules to choose the best split variable 117](#_Toc28721541)

[Gini 117](#_Toc28721542)

[Information Gain (Entropy) 117](#_Toc28721543)

[Pruning 119](#_Toc28721544)

[Methods of Pruning 119](#_Toc28721545)

[Hyper Parameter Tuning 119](#_Toc28721546)

[Limitations of Decision tree 119](#_Toc28721547)

[17. KNN 119](#_Toc28721548)

[Difference b/w K-Mean and KNN 120](#_Toc28721549)

[K-Means (Recap) 120](#_Toc28721550)

[KNN = K-Mean + Predictions 121](#_Toc28721551)

[Brut Method 122](#_Toc28721552)

[Brute force 122](#_Toc28721553)

[(Kd-Tree or ball-tree). 122](#_Toc28721554)

[18. Ensembling 123](#_Toc28721555)

[Injective or cognitive method 123](#_Toc28721556)

[Average Probability Method 124](#_Toc28721557)

[Voting Method 124](#_Toc28721558)

[Bagging Method 124](#_Toc28721559)

[oob\_score 125](#_Toc28721560)

[Drawbacks of bagging 126](#_Toc28721561)

[Random forest 126](#_Toc28721562)

[Boosting Method 128](#_Toc28721563)

[19. Logistic Regression (Most trusted Algo) 128](#_Toc28721564)

[Sigmoid Function 129](#_Toc28721565)

[Activation function of regression 131](#_Toc28721566)

[Activation function of classification 131](#_Toc28721567)

[Event Ratio 133](#_Toc28721568)

[Sign test 134](#_Toc28721569)

[20. Deep Learning (Neural Network) 136](#_Toc28721570)

[Neural network 136](#_Toc28721571)

[Some Important algorithms to focus to get job in Indian Industry 143](#_Toc28721572)

[21. SVM (Support Vector Machine) 143](#_Toc28721573)

[The best classifier (Hyper Plane) 144](#_Toc28721574)

[Kernel 144](#_Toc28721575)

[22. Timeseries 145](#_Toc28721576)

[Secular Trend (T) 146](#_Toc28721577)

[Seasonality Pattern (S) 146](#_Toc28721578)

[Cyclic Pattern( C ) 147](#_Toc28721579)

[Irregularity Component (I) 147](#_Toc28721580)

[23. NLTK 149](#_Toc28721581)

[Tokenization 149](#_Toc28721582)

[Sentence tokenization 150](#_Toc28721583)

[Word tokenization 150](#_Toc28721584)

[stop words 150](#_Toc28721585)

[Lemmatization or Normalization or stemming 150](#_Toc28721586)

[24. TextBlob 150](#_Toc28721587)

[Bag of words 151](#_Toc28721588)

[TF-IDF (Other name is word2vec) 151](#_Toc28721589)

[25. Sentimental Analysis 152](#_Toc28721590)

[26. Agglomerative Clustering 152](#_Toc28721591)

[Dendogram 153](#_Toc28721592)

[27. CNN (Image Classification) 153](#_Toc28721593)

[Step1: Pooling 154](#_Toc28721594)

[Step2: Standardization of data: (Flattening the data) 154](#_Toc28721595)

[Step3: Convolution 155](#_Toc28721596)

[Step4: Max Polling: 156](#_Toc28721597)

[Step5: Drop Out 156](#_Toc28721598)

[28. Model Deployment 157](#_Toc28721599)

[29. Imbalance data classification 158](#_Toc28721600)

[Over sampling (99% we use this) 158](#_Toc28721601)

[Under sampling 159](#_Toc28721602)

[SMOTE technique 160](#_Toc28721603)

[30. Steps of model building 160](#_Toc28721604)

[For Regression Problems 160](#_Toc28721605)

[For Classification Problems 161](#_Toc28721606)

[31. Telecom Domain 162](#_Toc28721607)

[32. Retail Business 165](#_Toc28721608)

[33. Classification Project 166](#_Toc28721609)

[34. Important Links 167](#_Toc28721610)

# What is data science

Art of making decisions by using historical data.

# Types of data

* Structured Data:
* Semi Structured Data
* Un structured Data

## Structured Data

This data consists of observations ad variables. Mysql, Oracle, SQL Server, Teradata, MS Excel, csv, text file. It’s like tables

|  |  |  |  |
| --- | --- | --- | --- |
| Structured Data | | | |
| Categorical data  (Non-Numerical/Qualitative/Non-Parametric) | | Numeric data  (Numerical/ Quantitative/Parametric) | |
| Nominal Data | Ordinal Data | Interval Data  (Continuous) | Ratio Data  (Discrete) |

### Nominal level data

Classifying the categorical level of data. Not assigning the Rank. E.g. EMP ID

### Ordinal level data

Classifying the categorical level of data based on particular property or characteristics. . Assigning the Rank. E.g. Class Rank

### Interval level data (Continuous data)

It is continuous data to measure the particular range of values. The ideal range is -∞ to +∞

e.g. Temperature in a day/month. Height, weight, BMI, Credit card bill, support calls

### Ratio level data (Discrete data)

The data which measures whole or natural numbers. It can be +ve or -ve. E.g age, salary.

## Semi Structured Data

Xml, HTML, JSON(90%). 90% of the websites store data into JSON format

## Unstructured Data

The data in the form of pdf, word, audio, video, images etc

# What is Statistics and it’s concepts

Statistics consists of performing following steps

* Collection
* Organizing
* Analyzing/Summarizing
* Interpret the data

## Collection

Collecting the structured data which consists of both numerical and categorical data

## Organizing

We can’t perform same analysis on entire dataset. It should be different for numerical and categorical data

## Analyzing/Summarizing

### Population/Universe

Collection of group of entities are called population. The description portion of the population is called **parameter**. Parameter can not be constant.

#### Descriptive statistics

To describe or analyze about population data is called descriptive statistics

### Sample

It is subset of population. One or few of the entities are called sample. The description portion of the sample is called **statistic/constant**.

Sample can be subset of population data but population data can not be subset of sample data.

#### Inferential statistics

To describe or analyze about sample data is called inferential statistics

Analyzing/Summary

Inferential statistics

(for sample data)

Descriptive Statistics

(for population data)

Measure of variation/dispersion

Measure of central tendency

Range

Variance

Standard Deviation

Mean

Median

Mode

## Numerical variable analysis

### Measure of central tendency

#### Mean

What is Mean: When random data is used in calculation then it’s mean.

What is Average: When sequential data is used in calculation then it’s average.

Hence average is special case of mean.

Mean = Sum(Individual data point) / count

Population Mean (µ) =

Sample Mean () =

#### Median

Middle point of a given data is called Median. The data should be sorted (ascending/descending)

Odd data series median = element on position

e.g series = {1,3,5,6,7}

median = 5

Even data series median =

e.g series = {1,3,5,6,7,9}

median = (5+6)/2 = 5.5

#### Mode

The number or the data point which is repeating or occurring most number of times is called Mode.

e.g. series = {10,15,15,15,15,10,3,7,9}

Mode = 15

##### No-Mode

If none of the data points are repeating then its No-Mode data

##### Uni-Mode

If one data points are repeating then it is Uni-Mode data

##### Bi-Mode

If any two numbers which are repeating the most and equal number of times then

It is Bi-Mode data

##### Tri-Mode

If any three numbers which are repeating the most and equal number of times then

It is Tri-Mode data

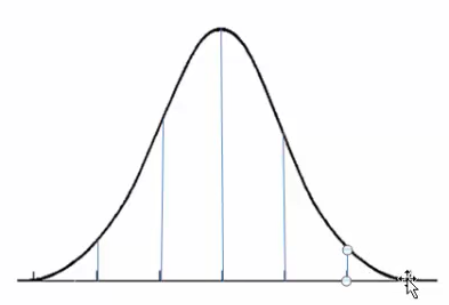
##### Multi-Mode

If more than three numbers are repeating the most and equal number of times then

its Multi-Mode data

### Symmetrical Data

If Mean = Median = Mode then it is called Symmetrical data



Bell Curve

Which never touches the x-axis

AUC : Area under the curve

100%: for symmetrical data

### Drawbacks of Measure of central tendency

By using measure of central tendency, we are able to calculate the middle values of a given data series but we are not able to calculate min, max and how each data point is far away from other data point. To overcome these issues we are going to use measure of dispersion/variability.

### Measure of dispersion

#### Range

Difference between max and min values

R = Max - Min

#### Variance

How each and every data point is variating or deviating from its mean value. Deviation can’t be negative but it can be square.

#### Standard Deviation

Square root of variance is standard deviation

#### Summary Statistics

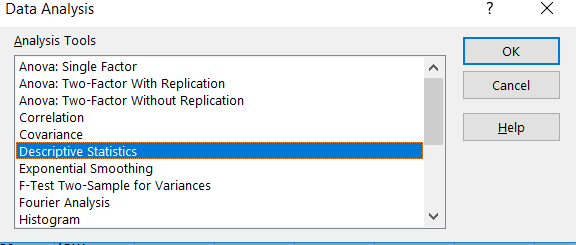
It consists of following (for Numerical data):

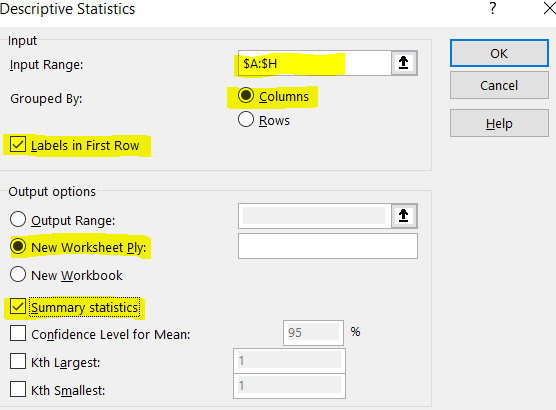
* + - Min
    - Max
    - Range
    - Mean
    - Median
    - Mode
    - Variance
    - Std. deviation
    - Count

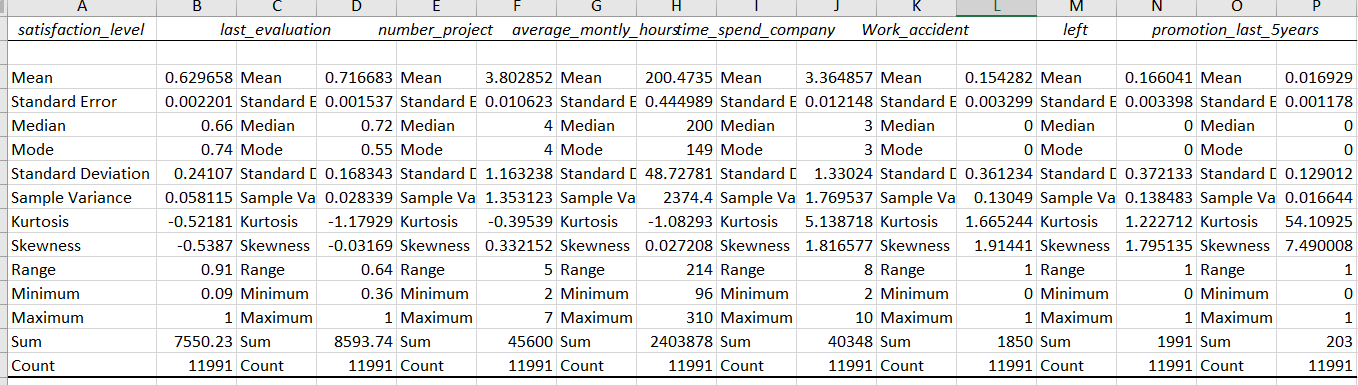
#### Summary Statistics (Numeric data) (Using excel)

**To enable** **data analysis button in Excel (in Data tab)**

* + - * + Click the File tab, click Options, and then click the Add-Ins category.
        + In the Manage box, select **Excel** Add-ins and then click Go.
        + In the Add-Ins available box, select the **Analysis** ToolPak check box, and then click OK.





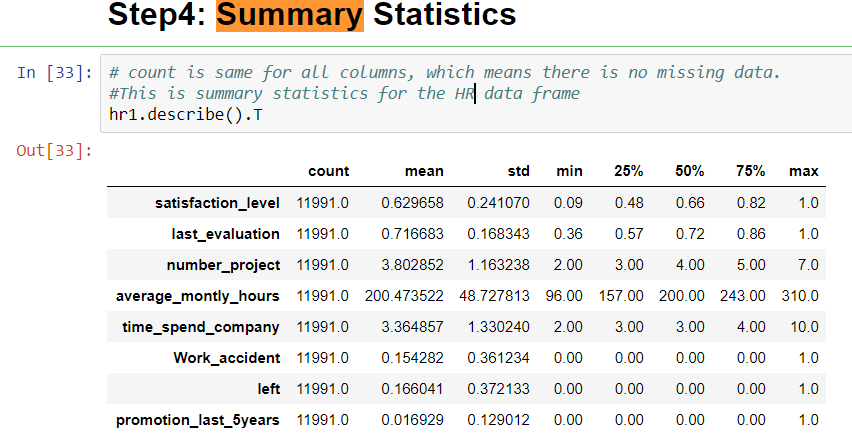


#### Summary Statistics (Using python)

##### Numeric data

1. For numerical data (1 column of a DF)--> df["col\_name"].describe()
2. For numerical data (all columns of a DF)--> df.describe().T
3. For categorical data--> df["col\_name"].value\_counts()/len(df)\*100

It is produced for numeric data



##### Categorical variable analysis

* + - * It can be analyzed using frequency distribution
      * For categorical data--> df["col\_name"].value\_counts()/len(df)\*100
      * If a variable can be treated as categorical variable but has numeric continuous data.or nominal data we should apply Binning or Bucketing

**Binning or Bucketing Nominal or continuous data**

|  |
| --- |
| telcoBill['SERVICE\_ACTIVE\_DT1'] = pd.cut(telcoBill['SERVICE\_ACTIVE\_DT'], 8)  telcoBill['ACT\_ACTIVE\_DT1'] = pd.cut(telcoBill['ACT\_ACTIVE\_DT'], 8) |
|  |
|  |

## Interpret the data

# Python

The process of converting to bits to bytes is called Kernel. The process of converting PVC to PVM is called Kernel

Kernel

Output

PVM

PVC

Input

Python is case sensitive language.

## Introduction to Python

1. It is a general purpose programming language
2. It is human readable format
3. We are using Juputer NB for data science
4. Ctrl+ to increase the font in Jupyter NB
5. shift + enter to execute this
6. It is a case sensitive language
7. ipynb --> Interactive Python notebook
8. Cluster is for parallel execution
9. Upload is to import any file to the library

## How to create the object

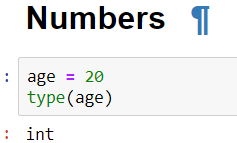
var must start with (A-Z)(a-z)(\_ follwed by 0-9 or \_ or A-Z or a-z)

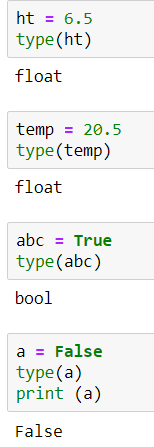
## Data Types

In python we have 2 data types

### Numbers

* 1. int --> data point which does not have any decimal values
  2. float --> data point which has decimal values
  3. boolean --> It has true or false only
  4. complex --> data point which are like 1j,2j,0j. Not real number and are define with number followd by j (or i)





### String

it should be define with single/double quotes

|  |
| --- |
| print(54/5) # single slash prints the float value  print(22//7) # double slash prints the Int value  ht = 6.5  type(ht)  abc = True  type(abc)  a = 0j  type(a)  a = "Hello Word"  print (type(a))  print(a[0])  # displays 1st letter from the rightside  print(a[-1])  #displays 10 times of abc  a\*10  abc1="data"  abc1+" " + abc1  abc1+"data" |

## Data structures

1. List --> Collection of numbers/strings. Mutable. It must be specied in [] square bracket
2. Tuples --> Collection of numbers/strings. Immutable. It must be specied in () square bracket
3. Dictionary --> Define with {} & also called user defined formats.Consists of Keys and Values
4. Set --> Normal set (Mutable), Frozen set(Immutable).

### List

List is mutable. It is defined with [] Different operation we can perform on list

1. Insert
2. Update
3. Delete
4. Concatenate/Append
5. Length
6. Pop

|  |
| --- |
| # list can be defined in 2 ways  # type1  a = 50,110,"aus","tara"  my\_list1 = list[a] # check this way of defining the list  #**TypeError**: 'type' object is not subscriptable  # type2  my\_list = [20,25,45,"abc","India",60.05]  type(my\_list)  my\_list  len(my\_list)  my\_list[3]  my\_list[2:5]  my\_list[-2]  my\_list.index(45)  # to edit any element in a position  my\_list[2]=5000  del my\_list[4] # to delete object at postion 6  my\_list  my\_list.append(70)  my\_list  my\_list.pop() # deletes element in the last position  my\_list.insert(2,3) # insert at postion 2 the value of 3. At a time only one position can be inserted  my\_list  my\_list4 = [100,340,[234,56]]  my\_list4[-1][0] # access element of list within the list  my\_list4 = [10,20,3,60,1000,'india','sab']  my\_list4.index('india')  my\_list5 = [10,50,70,80,1000,'india','UAE']  len(my\_list5)  my\_list5.index(10)  del my\_list5[1] |

### Tuple

Tuples are immutable. It is defined with () By default ML o/p is stored in the form of Tuple. It is collection of numbers and strings. Different operation we can perform on Tuple

1. cant delete any element
2. cant update
3. cant insert
4. cancatenate 2 diff tuples is possible

|  |
| --- |
| #  my\_tuple1 = (10,50,70,80,'india','uae')  type(my\_tuple1)  del my\_tuple1[3] # no delete operation possible  my\_tuple1[2] = 500 # no insert operation possible  # concatenate is possible  my\_tuple2 = (50,100,150)  my\_tuple1 + my\_tuple2  my\_tuple3 = (50,100,(160,200),300) # nested tuple  my\_tuple3[2][0] # access nested tuple element  my\_tuple.insert(2,5) # no insert possible |

### Dictionary

It is define with {}. It is also called as user defined formats. In this we have 2 components. Keys and Values are separated by : (colon) Before : are keys , after : are values Keys:Values Keys and Values can be numbers or string. Key should be unique. One dictionary position is separated from other with comma (,)

1. Add
2. Update
3. Delete
4. can not Concatenate 2 different dict

|  |
| --- |
| #  my\_dict1 = {"name":"tara","age":30,"salary":100000}  type(my\_dict1)  # element can be extacted using key name and not by index  my\_dict1["name"]  my\_dict1["age"] = 40 # element can be updated using key name  my\_dict1  del my\_dict1["salary"] # element can be deleted using key name  my\_dict1  my\_dict1["loc"] = "NJ" # element can be added using key name  my\_dict1.keys() # display all the keys of dictionary  my\_dict1.values() # display all the keys of dictionary  std1= {"Name":"Tara", "age":30,"sal":50000}  type(std1)  std1[0] # it doesnot support index like List or Tuple  std2= {"Name1":"Tara", "Name1":30,"sal1":50000} # if we have same key then latest key will be taken |

### Sets

Combination of number. It is in {}. It holds only unique values. No partricular ordering is followed. Eevn if duplicate value is added, it will effectively not add any duplicate value.

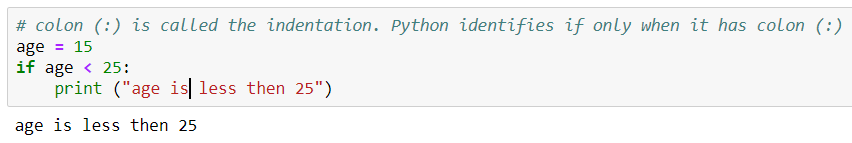
Two types of set 1.Normal set (Mutable) --> Union, Intersection, difference 2.Frozen set(Immutable)

|  |
| --- |
| #  abc1 = {10,20,50,40,60,70}  type(abc1)  abc2 = {160,100,140,80,50}  print("abc1 set values are ",abc1, "abc2 set values are ", abc2)  abc1|abc2 # unique values in data set, i.e. union w/o duplicates  abc1.union(abc2) # unique values in data set, i.e. union w/o duplicates  abc1&abc2 # intersection of two sets  abc1.intersection(abc2) # intersection of two sets  abc1-abc2 # abc1 minus intersection of two sets  abc1^abc2 # abc1 union abc2 minus intersection of two sets  # frozen set  abc3 = frozenset({10,50,70})  type(abc3)  abc3.add(100) # insert is not possible as it is immutable  abc2.add(200)  abc4 = {10, 20, 20}  abc4 # contains only unique values {10, 20} |

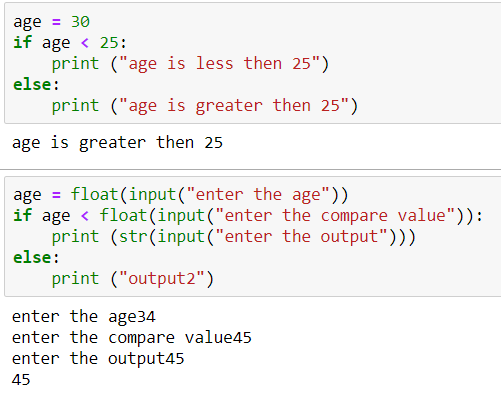
## Conditional statements

1. if
2. if-else
3. nested if
4. nested if-else

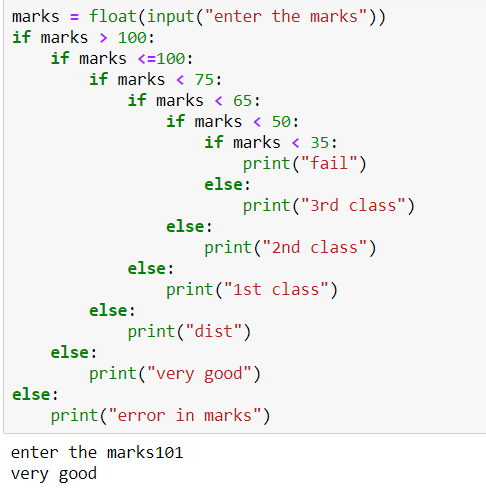
### if conditional statement



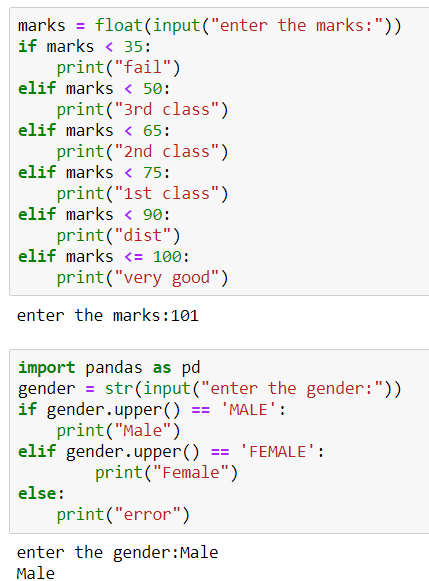
### if-else conditional statement



### nested if



### nested if-else conditional statement (if...elif..else)



## Looping statements

### For Loop

### While Loop

# Probability

Chances of occurring or happening of an event is called Probability.

**Event:** Any resulted outcome of a process or procedure is called event.

## Events are of 2 types:

### Simple event

* 1. Any resultant outcome is not possible to breakdown further. e.g. Tossed coin resulting in head/tail

### Rare event

* 1. The probability of happening an event is very rare. e.g winning a lottery event in a row of 10 times is rare event.

Probability is denoted as P

A,B,C,D are events

P(A) --> Probability of occurring an event A

--> Probability of NOT occurring an event A

P(A) +

Total range of probability value is 0-1

1 Certain to happen

0.75 Likely to happen

0.50 50% chances

0.25 Unlikely to happen

0.05 Unusual to happen

0 Impossible to happen

## Classification of Probability (3 Types)

1. Relative frequency
2. Classic approach
3. Subjective approach

### Relative frequency

Finding out the similar event probability value. We are just finding out how many times particular event outcome is occurring.

Conducting same event multiple times --> Trial

If trial = 100

P(S) = number of times particular even is occurring /total num of particular events is happening

**Drawback:** we are not giving the equal weights to each and every outcome of an event

### Classic approach

Each and every outcome of an event are having equally likely chances to happen

We have 100 multiple choice questions. Each question consists of 4 possible outcomes. Out of that 1 is correct answer. What is the probability to choose the wrong answer

3/4 is the probability.

Relative freq calculates overall sample set data whereas classic approach works at each and every individual event level data.

### Subjective approach

There is no formula/no rule. Subject to perception

There are 2 types of events

1. Independent events (disjoint/mutually exclusive/Non overlapping)

If one event is not influencing another event: Rolling two dices separately

1. Dependent Events (overlapping)

If one event is influencing another event : Playing cricket vs rain

Machine generated alternative text:



## Two rules in Probability

### Addition Rule

* 1. Combining more than one event into a single event is called addition.
  2. SetA + SetB --> Common portion is dependent, but uncommon portion is independent

If both events are dependent events

P(A U B) = P(A) + P(B) - P(A ∩ B) ----------------> **dependent events**

If both events are **independent events**

P(A U B) = P(A) + P(B) ----------------> **independent events**

### Multiplication rule

Out of scope at this moment (sir will teach later)

Contingency table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Allergy Drug | Seldane | placebo | Control |  |
| Headache | 49 | 49 | 24 | 122 |
| No Headache | 732 | 616 | 602 | 1950 |
| total | 781 | 665 | 626 | 2072 |

## Probability Distribution

How probability value will change at one particular point or in a particular range. In order to understand that we are going to use **distribution technique**

1. **Discrete Distribution** (At one particular point)
   1. Binomial Distribution
   2. Poisson Distribution
   3. Multinomial Distribution
   4. Hypergeometric Distribution
   5. Uniform Distribution

1. **Continuous Distribution** (At particular range)
   1. Normal Distribution
   2. T- Distribution
   3. Chi- Distribution
   4. Exponential Distribution
   5. F - Distribution
   6. Uniform Distribution

### Discrete Distribution (At one particular point)

#### Bi-nomial Distribution

* + - Nominal Data: Classifying the categorical level of data is called nominal data.
    - Bi-nomial - --> Classifying the two categorical level of data is called bi-nomial data.
    - Properties/Criteria to use binomial distribution :
      * Number of trials(events) are fixed
      * Each trail having 2 possible outcomes --> Success and Failure or Yes/No, True/False, Win/Loose
      * Each and every event is independent to other event
      * The probability of success is the same in every one of the trials
        + Probability(Success) = p
        + Probability(Failure) = q
        + p+q = 1

Machine generated alternative text:
the couq of 
of ways em 
n! 
(x) 
(n -x)!x! 
ms cart of 
mrrber of ways evert can 
TWO 
THS is the probabiEty 
of *Eeess Er x thü. 
TN' is 
the x trials. 

Machine generated alternative text:
Notations for Binomial Distribution and the Mass Formula: 
nCx pxqn-x 
Where: 
O P is the probability of success on any trail. 
O q = I-P - the probability of failure 
O n - the number of trails/experiments 
O x- the number of successes, it can take the values 0, 1, 2, 3 
o 
nCx = n!/x!(n-x) and denotes the number of combinations of n elements 
taken x at a time. 
Assuming what the nCx means, we can write the above formula in this way: 
P(X) : 
pxqn x 

* µ = n\*p
* n --> num of observation
* P probability of success
* **If µ > 7, though the data is a binomially distributed data it behaves like a Poisson distribution**
* **Excel function:**

=**BINOMDIST function**. = BINOM.DIST(number\_success,trials,probability\_success,)

Machine generated alternative text:
BINOM.DlST(number_s, trials, probability_s, cumulative) 

* **Drawbacks binomial distributed:** 
  + While calculating the binomial distributed we never measure the time interval, region, space.

**Examples of binomial distribution problems:**

* The number of defective/non-defective products in a production run.
* Yes/No Survey (such as asking 150 people if they watch ABC news).
* Vote counts for a candidate in an election.
* The number of successful sales calls.
* The number of male/female workers in a company

#### Poisson Distribution

* + To measure probability of a given number of events occurring in a fixed interval of time or space
  + Poisson is a scientist who found the technique
  + **Properties**:
    - Each and every time interval is independent with other time interval
    - Probability of success is directly proportional to the size of the interval
    - Probability of more than one success is close to 0 in very small size of the interval
    - The experiment results in outcomes that can be classified as successes or failures.
    - The average number of successes (μ) that occurs in a specified region is known.
    - The probability that a success will occur is proportional to the size of the region.
    - The probability that a success will occur in an extremely small region is virtually zero.

Note that the specified region could take many forms. For instance, it could be a length, an area, a volume, a period of time, etc.

* *e*: A constant equal to approximately 2.71828. (Actually, *e* is the base of the natural logarithm system.)
* μ: The mean number of successes that occur in a specified region.
* *x*: The actual number of successes that occur in a specified region.
* P(*x*; μ): The **Poisson probability** that exactly *x* successes occur in a Poisson experiment, when the mean number of successes is μ.

The Poisson distribution is popular for modelling the *number of times an event occurs in an interval of time or space*.

**Example**

The Poisson distribution may be useful to model events such as

* The number of meteorites greater than 1 meter diameter that strike Earth in a year
* The number of patients arriving in an emergency room between 10 and 11 pm
* The number of photons hitting a detector in a particular time interval

**Problem statement1:** Pearson has conducted a research. To identify type mistakes. On and avg for every 100 pages there are 1.5 typo mistakes. For new text book (100 pages book) what is the probability to get 0 mistakes

µ= 1.5

x = 0

**Problem statement2:** Pearson has conducted a research. To identify type mistakes. On and avg for every 100 pages there are 1.5 typo mistakes. For new text book (400 pages book) what is the probability to get 3 mistakes

µ= 1.5 \* 400/100 = 6

x = 3

f(x) = e^-6

On a avg 12 cars crossing the bridge in one minutes, what is the prob to cross 16 or more

µ= 12

x = 16

**Cumulative Poisson Example**

**Problem statement3:** Suppose the average number of lions seen on a 1-day safari is 5. What is the probability that tourists will see fewer than four lions on the next 1-day safari?

*Solution:* This is a Poisson experiment in which we know the following:

* μ = 5; since 5 lions are seen per safari, on average.
* x = 0, 1, 2, or 3; since we want to find the likelihood that tourists will see fewer than 4 lions; that is, we want the probability that they will see 0, 1, 2, or 3 lions.
* e = 2.71828; since *e* is a constant equal to approximately 2.71828.

To solve this problem, we need to find the probability that tourists will see 0, 1, 2, or 3 lions. Thus, we need to calculate the sum of four probabilities: P(0; 5) + P(1; 5) + P(2; 5) + P(3; 5). To compute this sum, we use the Poisson formula:

P(x < 3, 5) = P(0; 5) + P(1; 5) + P(2; 5) + P(3; 5)

#### Multinomial Distribution

* + Classifying more than 2 categorical level of the data. Eg. Good, not good, poor
  + Questions: Suppose that two chess players had played numerous games and it was determined that Player A would win is 0.40, the probability that player B would win is 0.35 and the probability that the game would end in a draw is 0.25. The multinomial distribution can be used to answer questions such as " if these two chess players played 12 games, player B would win 2 games and remaining 3 games would be drawn? The following formula gives the probability of obtaining a specific set of outcomes when there are three possible outcomes for each event.
  + Formula

Where n =

#### Hypergeometric Distribution

* 1. It is used to calculate probabilities when sampling without replacement
  2. **Properties**:
     1. A sample of size n is randomly selected without replacement from a population of N items
     2. In the population k items can be classified as successes and N-k items can be classified as failures
     3. The mean of the distribution is = n \* k/N
     4. The variance is n \* k \* (N - k) \* (N-n)/[N^2 \* (N-1)]

Variance =

* + 1. N = The number is items in population
    2. n: The number is items in sample
    3. K = The number is items in population that are classified as successes
    4. x: The number is items in sample that are classified as successes
    5. kCx = the number of combination of k things, taken x at a time
    6. h(x;N,n,k): hypergeometric probability the probability that an n-trial hyper geometric experiment results in exactly x success when the population consists of N items, k of which are classified as successes
  1. Formula:

Machine generated alternative text:



1. **Example**: Suppose we select 5 cards from an ordinary deck of playing cards. What is the prob of obtaining 2 or fewer hearts
2. Sol: This hypergeometric experiment
   1. N=52, since there are 52 cards in deck
   2. K = 13, since there are 13 hearts in deck
   3. n=5, since we randomly select 5 cards from deck
   4. X = 0 to 2 since selection includes 0,1 or 2 hearts

#### Uniform Distribution

(not taught in the class)

### Continuous Distribution (In Particular range)

How probability values are changing is particular range

**Continuous distribution helps to identify this**

#### Normal Distribution (central limit theorem/Gaussian distribution)

1. Whenever we have Mean = median = mode, then the continuous distributed data is called normally distributed data.
2. If Mean = median = mode, then the data is called symmetrical data. It looks like bell shaped curve. It never touches the x-axis. The area under the curve is 100% of data.
3. AUC = 100% or 1

In order to perform normal distribution, we required at least 30 observations. To check the normal distribution the prerequisites are

* **Data should not have missing value**
* **Data should not have outliers**

Mean=Median=Mode

Machine generated alternative text:
s" m rwtrical 

Mean > Median --> Then is it called right skewed data

Machine generated alternative text:
Mode 
Mean 
Right-Skewed (Positive Skewness) 

Mean < Median --> Then is it called left skewed data

Machine generated alternative text:
Mod e 
Mean 
Left-Skewed (Negative Skewuess) 

**Why do we make the data Normally distributed:** To control the variability of a data and To make the data into uniform ranges we need normally distributed data.

How each and every data point is variating/deviating for its mean value.

**Other name of Normal distribution is central limit theorem.**

Machine generated alternative text:
Areas under the normal curve that lie between I, 2, and 3 
standard deviations on each side of the mean 
68.3% of data 
95.5% of data 
99.7% of data 
-3SD 
-2SD 
-ISD 
+ ISD 
+ 2SD 
+ 3SD 

Machine generated alternative text:
g-36 
g +26 
g 36 

Any data point are within the range of µ-3sigma to µ+are called normally distributed data point

Z = (X - µ)/  
(Z = standardd normal distribution value or score, X = indivudial data point, µ is mean, is standard deviation)

If **-3 ≤ Z ≥ +3** then data is called normally distributed data

Normal distribution f(x) =

Any data point which are beyond the range of µ-36 to µ+36 are called outliers (extreme values)

# Outlier Detection (Identify Outliers)

**Outliers detection is done by using following 4 methods. 1st 3 are to check whether we have outliers. IQR is to also find how many outliers**

1. Boxplot (5 point number summary)
2. Qqplot (quantile-quantile plot)
3. z-score
4. **IQR --> (Inter Quartile Range) --> widely used**

### Boxplot (5 point number summary)

* 1. It is also called the 5 point number summary.
     1. Min
     2. 25%
     3. 50%
     4. 75%
     5. Max

If width of boxplot increases then we have outlier in the data

* Extreme outliers are marked with an asterisk (\*) on the boxplot.
* Mild outliers are marked with a circle (O) on the boxplot.

**Drawback**: we can know if outliers are present but we cant determine how many outliers are present

|  |
| --- |
| import matplotlib.pyplot as plt  plt.boxplot(fb1["TAX\_AMOUNT"])  import seaborn as sns  sns.boxplot(fb1["TAX\_AMOUNT"]) |
|  |

### qqplot: quantile-quantile plot

* 1. 0 -->
  2. -1 to 1 --> µ - to µ + (68%)
  3. -2 to 2 --> µ - 2 to µ + 2 (95%)
  4. -3 to 3 --> µ - 3 to µ + 3 (99.7%)
  5. -4 to 4 --> µ - 4 to µ + 4 (99.7999%)
  6. When red line and blue line coincides then we say it is normally distributed data
  7. A point which is lying far away are outliers
  8. **Drawback**: Drawback of qqplot: by using the qqplot we can know if outliers are present but we cant determine how many outliers are present

|  |
| --- |
| from statsmodels.graphics.gofplots import qqplot  qqplot(fb1["TAX\_AMOUNT"],line="s")  # qqplot(telcoBill\_treat\_outlier["TAX\_AMOUNT"],line="s") |
|  |

### z-score

* 1. **z = (x-**µ**)/**

#### StandardScaler to check the outliers

1. z = (x-mu)/sigma
2. Excel functions used AVERAGE, STDEV.P, STANDARDIZE
3. StandardScaler (x-mu)/sigma
4. ss.fit\_transform(fb) --> it transform the data into range of -3 to +3
5. upper outlier and lower outlier
6. values < - 3 are lower side outlier
7. values > 3 are upper side outlier
8. **Drawback** : We still dont know what is the cutoff for lower/upper side outlier and how many outluers are present. We can only know that we have outlier

* **Excel functions used**

AVERAGE

STDEV.P

STANDARDIZE

|  |
| --- |
| from scipy import stats  stats.ttest\_ind(hr1[hr1.left==1]['satisfaction\_level'],hr1[hr1.left==0]['satisfaction\_level'])  # statistic is Z-score value(level of confidence), pvalue |
|  |
| from sklearn.preprocessing import StandardScaler  ss = StandardScaler()  telcoBill\_num = telcoBill\_treat\_outlier[['DATA\_USAGE','DATA\_USAGE\_UNITS','NRC\_AMOUNT','RC\_AMOUNT','TAX\_AMOUNT','NEW\_CHRGS','CURRENT\_CHRGS','UNITCREDIT\_AMOUNT','DISCOUNT','ADJ\_AMOUNT','OTHERS','OTHERS\_UNITS']]  fb1 = pd.DataFrame(ss.fit\_transform(telcoBill\_num),columns=telcoBill\_num.columns)  fb1.min()  fb1.max() |
|  |

### IQR (Inter Quartile Range)widely used

# IQR : Inter Quartile Range:

# **IQR = Q3 - Q1 or 75% - 25%**

1. There are 4 quartiles: Q1 (25%) ,Q2 (50%) ,Q3 (75%) ,Q4 (100%)

#The values to detect outlier points will be

2. L.O. (lower outliers)= Q1 - 3 \* IQR (this is practically)

3. U.O. (upper outliers)= Q3 + 3 \* IQR (this is practically)

4. Lower cutoff value = Q1 - 1.5 \* IQR (this is theoretically)

5. Upper cutoff value = Q3 + 1.5 \* IQR (this is theoretically)

* Outliers are of 2 types
  + LO
  + UO
* L.O. (lower outliers)= Q1 - 3 \* IQR (this is practically)
* U.O. (upper outliers)= Q3 + 3 \* IQR (this is practically)
* iqr = Q3-Q1
* Compute the outer fences OF1 or LO = Q1 - 3 \* IQR and OF2 or UO = Q3 + 3 \* IQR.
* Extreme outliers (Wild outliers) are observations that are beyond one of the outer fences OF1 (LO) or OF2(UO).
* Mild outliers are observations that are between an inner and outer fence (b/w LO and UO)

#### Mild (50%-75%) vs. Extreme Outliers (Beyond 75%)

* Extreme outliers are data points that are more extreme than Q1 - 3 \* IQR or Q3 + 3 \* IQR.
* Extreme outliers are marked with an asterisk (\*) on the boxplot.
* Mild outliers are data points that are more extreme than than Q1 - 1.5 \* IQR or Q3 + 1.5 \* IQR, but are not extreme outliers.
* Mild outliers are marked with a circle (O) on the boxplot.
* B/c of Mild outliers we have data NOT normally distributed hence we apply variable transformation technique
  + square
  + sqrt
  + Log
  + Inverse

|  |
| --- |
| # how to check how many records are below LO and above UO  def IQR\_inter\_quantile\_range(df):  return df.quantile(0.75)-df.quantile(0.25)  def LO\_Lower\_cutoff\_point(df,IQR):  return df.quantile(0.25)-3\*IQR  def UO\_Upper\_cutoff\_point(df,IQR):  return df.quantile(0.75)+3\*IQR  def MIN(df):  return df.min()  def MAX(df):  return df.max()  col = ['DATA\_USAGE','DATA\_USAGE\_UNITS','NRC\_AMOUNT','RC\_AMOUNT','TAX\_AMOUNT','NEW\_CHRGS','CURRENT\_CHRGS','UNITCREDIT\_AMOUNT','DISCOUNT','ADJ\_AMOUNT','OTHERS','OTHERS\_UNITS']  for i in col:  iqr = round(IQR\_inter\_quantile\_range(telcoBill[i]),2)  lo = round(LO\_Lower\_cutoff\_point(telcoBill[i],iqr),2)  uo = round(UO\_Upper\_cutoff\_point(telcoBill[i],iqr),2)  min = MIN(telcoBill[i])  max = MAX(telcoBill[i])  print('# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print("# ", i)  print(" \* % of rec below lower outlier: ",np.sum(np.where(telcoBill[i] < lo,1,0))/len(telcoBill)\*100)  print(" \* % of rec above upper outlier: ",np.sum(np.where(telcoBill[i] > uo,1,0))/len(telcoBill)\*100)  print(" \* Lower cut-off point = ",lo,"\*\*\*\*\*\*\*Upper cut-off point = ",uo)  print(" \* Min= ",min,"\*\*\*\*\*\*\*Max = ",max)  print(" \* IQR= ",iqr) |
|  |

## Outlier Treatment

Outlier treatment is done using IQR method

If there are any upperside outliers those data points are going to be replaced by UO values

If there are any lowerside outliers those data points are going to be replaced by LO values

### IQR method for outlier treatment

|  |
| --- |
| telcoBill\_treat\_outlier = telcoBill  col = ['DATA\_USAGE','DATA\_USAGE\_UNITS','NRC\_AMOUNT','RC\_AMOUNT','TAX\_AMOUNT','NEW\_CHRGS','CURRENT\_CHRGS','UNITCREDIT\_AMOUNT','DISCOUNT','ADJ\_AMOUNT','OTHERS','OTHERS\_UNITS']  for i in col:  iqr = round(IQR\_inter\_quantile\_range(telcoBill\_treat\_outlier[i]),2)  lo = round(LO\_Lower\_cutoff\_point(telcoBill\_treat\_outlier[i],iqr),2)  uo = round(UO\_Upper\_cutoff\_point(telcoBill\_treat\_outlier[i],iqr),2)  telcoBill\_treat\_outlier[i] = np.where(telcoBill\_treat\_outlier[i] > uo,uo,telcoBill\_treat\_outlier[i])  telcoBill\_treat\_outlier[i] = np.where(telcoBill\_treat\_outlier[i] < lo,lo,telcoBill\_treat\_outlier[i]) |
|  |

# To check if data is normally distributed

To Check if my data is Normally distributed or not, we have 4 methods

### dist plot

|  |
| --- |
| import seaborn as sns  # how Work\_satisfaction has impacted satisfaction level of employees  # lesser satisfaction for employess with lower workload  sns.distplot(hr1[hr1.Work\_accident==1]['satisfaction\_level'],color='r')  sns.distplot(hr1[hr1.Work\_accident==0]['satisfaction\_level'],color='g') |
|  |

### qq plot

* 1. 0 --> sigma
  2. -1 to 1 --> mu - sigma to mu + sigma(68%)
  3. -2 to 2 --> mu - 2sigma to mu + 2sigma (95%)
  4. -3 to 3 --> mu - 3sigma to mu + 3sigma (99.7%)
  5. -4 to 4 --> mu - 4sigma to mu + 4sigma (99.7999%)
  6. When red line and blue line conside then we say it is normally distributed data
  7. A point wich is lying far away are outliers
  8. **Drawback**: Drawback of qqplot: by using the qqplot we can know if outliers are present but we cant determine how many outliers are present

|  |
| --- |
| from statsmodels.graphics.gofplots import qqplot  qqplot(fb1["TAX\_AMOUNT"],line="s")  # qqplot(telcoBill\_treat\_outlier["TAX\_AMOUNT"],line="s") |
|  |

### z-score

* 1. StandardScaler to check the outliers
  2. z = (x-mu)/sigma
  3. Excel functions used AVERAGE, STDEV.P, STANDARDIZE
  4. StandardScaler (x-mu)/sigma
  5. ss.fit\_transform(fb) --> it transform the data into range of -3 to +3
  6. df.min() values < - 3 are lower side outlier
  7. df.max() values > 3 are upper side outlier
  8. **Drawback** : We still don’t know what is the cutoff for lower/upper side outlier and how many outliers are present. We can only know that we have outlier

|  |
| --- |
| from scipy import stats  stats.ttest\_ind(hr1[hr1.left==1]['satisfaction\_level'],hr1[hr1.left==0]['satisfaction\_level'])  # statistic is Z-score value(level of confidence), pvalue |
|  |
| from sklearn.preprocessing import StandardScaler  ss = StandardScaler()  telcoBill\_num = telcoBill\_treat\_outlier[['DATA\_USAGE','DATA\_USAGE\_UNITS','NRC\_AMOUNT','RC\_AMOUNT','TAX\_AMOUNT','NEW\_CHRGS','CURRENT\_CHRGS','UNITCREDIT\_AMOUNT','DISCOUNT','ADJ\_AMOUNT','OTHERS','OTHERS\_UNITS']]  fb1 = pd.DataFrame(ss.fit\_transform(telcoBill\_num),columns=telcoBill\_num.columns)  fb1.min()  fb1.max() |
|  |

### Jarquebera test (JB test)

* J.B = n/6(s^2 + 1/4(k-3)^2)

J.B =

* s= skewness
* k = kurtosis
* n = number of observations
* In general if
  + k=3 (bell shaped) then data is normally distributed data, bell shaped curve
  + k < 3 (mono scratic data)
  + k > 3 (lepo scratic)
* if JB test is close to 0 then data is normally distributed data
* if skew is -ve then left skewed data
* if skew is +ve then right skewed data

|  |
| --- |
| col = ['DATA\_USAGE','DATA\_USAGE\_UNITS','NRC\_AMOUNT','RC\_AMOUNT','TAX\_AMOUNT','NEW\_CHRGS','CURRENT\_CHRGS','UNITCREDIT\_AMOUNT','DISCOUNT','ADJ\_AMOUNT','OTHERS','OTHERS\_UNITS']  for i in col:  s = telcoBill[i].skew()  k = telcoBill[i].kurtosis()  print('# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  # print("# ", i)  if s < 0:  if k == 3:  print("# ", i," -- left skewed , bell shaped data")  elif k < 3:  print("# ", i," -- left skewed , mono scratic data")  else:  print("# ", i," -- left skewed , leposcratic data")  else:  if k == 3:  print("# ", i," -- right skewed , bell shaped data")  elif k < 3:  print("# ", i," -- right skewed , mono scratic data")  else:  print("# ", i," -- right skewed , leposcratic data") |
|  |

## How do we make the data Normally distributed?

### Variable Transformation Technique

**If in case data is not normally distributed we need to go for variable transformation technique**

#### square

most preferred, if data is right or left skewed data

#### sqrt

if no -ve then this is used,

#### inverse

if no 0 then this is 3rd preferred

#### log

if no -ve and 0 then this is the preferred (most preferred is when data is non-linear data, sin curve kind of) right or left skewed data

**Limitations**:

a. If there at any negative value in the data we can not directly apply sqrt and log to the data

b. If there at any 0 value in the data we can not directly apply inverse and log to the data

c. square transformation does not have any limitation. Hence if data is not normally distributed the most preferred transformation is square

1. If your data is normally distributed 99.99% your data is linear in nature.
2. If data is non-linear data, to make it linear in nature or normally distributed, the most preferred Transformation is log transformation of the data
3. If data is right or left skewed data the best possible transformations are square and log

**To standardize the data (or to make it uniform) we are going to use the method minmaxscalar**

#### minmaxscalar [0 to 1] = (x-min(x)) /(max(x) - min(x))

1. This can be applied to even binary data, its not going to change the end result (yes/no, True/False etc)

|  |
| --- |
| a = [10,20,30,40,50]  min = np.min(a)  max = np.max(a)  diff = max-min  minmaxscalar\_10 = (a[0] - np.min(a))/diff  # similaryly do for a[1] and others  Or  from sklearn.preprocessing import MinMaxScaler  mn = MinMaxScaler()  fb1 = pd.DataFrame(mn.fit\_transform(fb),columns=fb.columns) |
|  |

## Data standardization (Making the data Normal distributed and MinMaxScaler)

When do we use Normal distribution methods Vs MinMaxScaler

MinMaxScaler is used in non-banking industries where each varaible is not so important

1. Normal distribution is required in traditional/predictive models
   1. Linear Regression (simple, Multiple)
   2. Logistics Regression
   3. Decison trees
   4. Random forest
   5. SVM (support vector machines)
2. MinMaxScaler is used in non-traditional models

**General flow of model building**

Machine generated alternative text:
st Mdxd 
EDA 

# How to check missing values ?

|  |
| --- |
| def identify\_missing(df):  df1 = **df.isnull().sum()/len(df)\*100**  df = df1[df1 > 0]  print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Columns with Missing values and % of missing values \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  print (df)  return (df)    missing\_df = pd.DataFrame(identify\_missing(telcoBill\_wo\_dup),columns = ['missing\_percentage'])  missing*.shape*  missing*.head()*  missing*.index* |
|  |

|  |
| --- |
| df.info() |
|  |

## Missing Value Treatment

1. Missing values :the records or data points which has not been populated for a particular col/variable. These are called missing values or null values
2. It can be for both numerical as well as categorical variable
   1. **Numerical -->**
      1. if variable have more than 95% missing values then discard the variable
      2. This is the 1st variable reduction
      3. If missing values are greater than 50% less than 95%, then create a flag(column) when it is missing 1 otherwise 0 and then drop the original variable. This means we convert the variable into binary variable
      4. If variable is important variable then we divide it into bins/groups. The reason behind making into bins is to take any decision based on bins/groups (we create 10 groups in decile, 4 groups in quantile)
         1. 0-10%
         2. 10-20%
         3. 20-30%
         4. 30-40%
         5. …
         6. 90-100%
      5. If missing values are 10% to 50% then we replace with median value. Median is not affected by outliers, Even mode is not affected. But there are chances that outlier are in mod

1. If missing values are 1% - 10% then we replace with mode value. If we use median instead of mode, then kurtosis will increase and we don't have bell shaped

1. If missing values are less then 1% then we replace with mean value. If we use median or mode instead of mean, then kurtosis will increase and we don't have bell shaped

|  |
| --- |
| Placeholder for code |
| Placeholder for code |

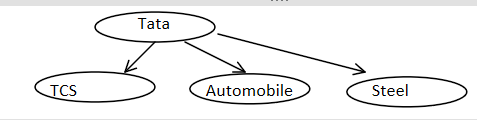
1. **categorical variable**
   1. if variable have more than 95% missing values then discard the variable
   2. This is the 1st variable reduction
   3. If missing values are greater than 50% less than 95%, then 1st we create the frequency count percentage of missing and non-missing data. Then create bins/group of data (max 4 groups are created generally)
   4. If missing values are 10% to 50% then we replace with "missing" indicator
   5. If missing values are < 10% then we replace with mode value

|  |
| --- |
| Treat missing Value (for Categorical Variables) cols = ['PARENT\_ID', 'BILL\_COMPANY', 'PACKAGE\_ID', 'PRICE\_PLAN','CUSTOMER\_SEGMENT']  for i in cols:  for j in missing\_categorical.index:  if i == j:  if missing\_categorical.loc[j,:].item() < 10:  # print (missing\_categorical.loc[j,:])  mode = telcoBill\_wo\_dup[i].mode()  # print (mode[0])  telcoBill\_wo\_dup[i] = np.where(telcoBill\_wo\_dup[i].isnull() ,mode[0],telcoBill\_wo\_dup[i])  elif missing\_categorical.loc[j,:].item() >= 10 and missing\_categorical.loc[j,:].item() <= 50 :  telcoBill\_wo\_dup[i] = np.where(telcoBill\_wo\_dup[i].isnull() ,"missing",telcoBill\_wo\_dup[i]) |
|  |

### We treat missing values first then we treat outliers and then check if the data is normally distributed

# Sampling Techniques

A collection or group of entities is called as population



Sample : One of the entity or few of the entities from the population data

Sample data should represents the population data. Sample proportion and population proportion should be equal to each other

 Contingency table

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Population** | **Sample** |
| Gender |  | 1000 |  |
| Male | 40% | 400 | 39 |
| Female | 35% | 350 | 36 |
| Others | 25% | 250 | 23 |

Machine generated alternative text:
Business Objective 
Data requirement 
High Volume 
Yes, do the sampling 
Missing 
Outlier 
EDA 
Train 
No 
Missing 
Outlier 
Test 

Sample size should be < one Million

Again sampling is done and divide data into 2 sets : 80% of the volume in train, 20% test data set

Biasness: When we consider only one set of data from population then it creates bias, which does not represent the population data.

We always do analysis on sample data, never on population data.

2 types data sources:

1. Primary Data sources (most reliable)
2. Secondary data sources

Machine generated alternative text:
Population 
Sample 
(size of sample and Technique 
Collecting the sample data 
Analysis 

## Sampling techniques are of 2 types

Machine generated alternative text:
Sampling Technique 
Non- Probability Method 
Judgement 
Quota 
Convenience 
Snowball 
Probability Method 
Simple Random 
Stratified Random 
Systematic/sequential 
Cluster 

## Non Probability Method (not used much)

### Judgement sampling

* + 1. Depends exclusively on judgement of investigator
    2. Sample selected which investigator thinks to be most typical of the universe

### Convenience sampling

* + 1. Convenient sample units selected
    2. Selected neither by probability nor by judgement

### Quota sampling

* + 1. Quotas set up according to some specified characteristics
    2. Within the quota selection depends on personal judgement

### Snowball sampling

* + 1. It may be extremely difficult or cost prohibitive to locate respondents in these situations
    2. Snowball sampling relies on referrals from initials subjects to generate additional subjects.

## Probability Method (most widely used)

### Simple Random sampling (SRS) or Random sampling method

* + 1. 90-95% of the time is it used
    2. Random sample will pick the samples based on seed number. Seed number is nothing but some specific number
    3. **SRS are of 2 types**

#### Sample without replacement (used mostly in ML)

* + - * 1. Whenever we are splitting the data into **training and test we need to always use sample without replacement**
        2. Data should be **unique** data at row level then only we can apply SRS
        3. Data should not have any **missing** value and outliers
        4. To compare the mean of the 2 different samples/populations we use T-Test
        5. T-Test : To compare two sample we use T-Test. Other name is Student T-Test
        6. T-Test normal distribution score = f(x) =

n= no of observations

µ= mean

X = individual data point,

* + - * 1. Comparing more than 2 mean of the sample / populations we can us ANOVA (Analysis of variance)
        2. T-test is to compare 2 sample/population, ANOVA is to compare more than 2 samples/population

Machine generated alternative text:
Business Objective 
Data requirement 
Size > 1 
Yes, do the sampling 
Remove duplicates 
Clean Data 
(Missing, Outlier, EDA) 
SRS Sampling 
Train (80%) 
Remove duplicates 
Clean Data 
(Missing, Outlier, EDA) 
Test (20%) 

#### Sample with replacement (used mostly in Deep Learning)

In this sample can have duplicates because as we pull data from population for sample, next data pull will have same population because whatever we pulled in as first records is replaced back to population hence there are chances of duplicates

* Why we divide data into train and test: To tune the model we use train data. To Validate the model we use the test data.
* size of train data > size of test data (70:30 or 80:20) But In pharma and healthcare (90:10). In research 50:50

**Train Data -->**  To tune the model we use train data

**Test Data -->** To Validate the model we use the test data.

**Holdout Data -->** After training and test, we use hold-out data on the model we got to validate it. If accuracy is the same then model is good o/w rebuild the model

|  |
| --- |
| import numpy as np  import pandas as pd  fb=pd.read\_csv("C:\\ksr\\data science\\DS\_batch1\\datasets\\Fiberbits.csv")  fb.shape  fb1=fb.drop\_duplicates()  fb1.shape  fb2=fb1.sample(10000) # this will NOT have duplicates  fb2.shape  fb2=fb2.drop\_duplicates()  fb2=fb1.sample(10000,replace=True) # this will have duplicates  fb2.shape  fb2=fb2.drop\_duplicates()  fb2.shape  from sklearn.model\_selection import train\_test\_split  train,test=train\_test\_split(fb1,test\_size=0.2,random\_state=1234)  print(train.shape,test.shape)  print(train["income"].mean(),test["income"].mean())  import matplotlib.pyplot as plt  plt.boxplot(fb1["income"])  plt.show()  %matplotlib inline |

### Stratified sampling

* 1. When **data volume is very high** then we use Stratified sampling
  2. Sampling based on particular categorical variable
     1. Eg. Suppose we have college student data. When we do sampling based on engineering branch of data

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Population** | **Sample** |
| Branch |  | 1000 |  |
| ECE | 40% | 400 | 39 |
| EEE | 35% | 350 | 36 |
| CS | 25% | 250 | 23 |

1. Imbalanced data: We use Stratified sampling incase of imbalanced data.
2. CDR call data record, fraud data

|  |
| --- |
| from sklearn.model\_selection import train\_test\_split  train,test=train\_test\_split(fb1,test\_size=0.2,random\_state=1234,stratify=fb1["active\_cust"])  print(train.shape,test.shape) |

Machine generated alternative text:
Train Data 
Business Objective 
Data requirement 
Size > 1 
Yes 
Stratified mpling 
Test Data 
Remove duplicates 
Clean Data 
(Missing, Outlier, EDA) 
SRS Sampling 
Train (80%) 
Remove duplicates 
Clean Data 
(Missing, Outlier, EDA) 
Test (20%) 

### Systematics/Sequential sampling

* 1. Never used in data science
  2. Used in election polls and political surveys
  3. 500 houses in village, Take a sample of say N houses. Go to 1st house, then skip x houses to again get the sample. This is called Systematic/sequential sampling
  4. Applying the process of applying results of sample data to population is called **inferential statistics/Extrapolation**

### Cluster sampling

* 1. xxxx

# Hypothesis Testing

What is hypothesis: A premise or claim that we want to test (<https://www.youtube.com/watch?v=VK-rnA3-41c>)

Hypothesis testing is a statistical technique, which is used to compare two datasets or a sample from a dataset

It is a statistical inference method so in the end of the test you'll draw a conclusion -- you 'll infer something - about the characteristics of what you are comparing

e.g. 3 movies release this weekend. Which combination of movie is good to watch

1. Bahubali
2. Joker
3. WAR

1 vs 2 , 2 vs 3 , 3 vs 1

It is the backbone for ML. Entire data science works on assumptions

**Prerequisites to perform hypothesis testing**

1. No Missing values
2. No Outliers
3. Data should be normally distributed. If data is not normally distributed then use variable transformation technique (square, SQRT, Long, Inverse) to make it normally distributed
4. **Confidence intervals (alpha α) or variable significance or** 
   1. µ - to µ + (68%)
   2. µ - 2 to µ + 2 (95%)
   3. µ - 3 to µ + 3 (99.7%)
5. **Deviation or error : (P-Value or**
   1. α
   2. Ideal or acceptable error is 5%
   3. P-Value is a statistical value. It measures the probability of deviation (probability distribution) of data

## Steps/components for hypothesis testing

### Define the hypothesis

* + 1. Define the requirement (eg. Which combination of movie to watch)

### Null hypothesis (, Read "H zero")

* + 1. This states that all things remain equal. No phenomena is observed or there is no relationship between what you are comparing
    2. Currently accepted value of a parameter (<https://www.youtube.com/watch?v=VK-rnA3-41c>)
    3. In the movie example, all the movies are good
    4. We always need to reject the Null Hypothesis
    5. If P ≤ 0.05 then we are rejecting the Null Hypothesis
    6. If P ≥ 0.05 then we are accepting the Null Hypothesis
    7. Which means if p-value is more we should reject that error
    8. If assumptions and Null hypothesis are the same still we are rejecting then it is called type-1 error

Tesco example:

|  |  |  |  |
| --- | --- | --- | --- |
| **Assumption** | **Null hypothesis** | **Reject** | **Error** |
| TRUE | TRUE | Reject | Type-1 |
| TRUE | FALSE | Accept | Type-II |
|  |  |  |  |
| Married and Pregnant | Married and Pregnant | Dad rejected the gift | Type-1 |
| Married and Pregnant | Not Married and Pregnant | Dad accepting | Type-II |

1. If assumptions and Null hypothesis are different still we are accepting then it is called type-2 error

### Alternative hypothesis (, read "H one", or Research Hypothesis)

* 1. States the opposite of Null Hypothesis. That there was some change, or observed relationship b/w what you are comparing
  2. It involves the claim to be tested
  3. P = 0.05
  4. Difference b/w Null hypothesis and Alternative hypothesis
     1. It is believed that a candy machine makes chocolate bar that are on avg 5 g. A worker claims that the machine after maintenance no longer makes 5 g bars. Write
  5. Possible outcomes of Hypothesis testing is
     1. Reject Null Hypothesis
     2. Fail to reject Null Hypothesis

1. Set the P-Value. It should always be equal to 0.05

|  |
| --- |
| from scipy import stats  stats.ttest\_ind(hr1[hr1.left==1]['satisfaction\_level'],hr1[hr1.left==0]['satisfaction\_level'])  # statistic is Z-score value(level of confidence), pvalue  #Ttest\_indResult(statistic=-40.98502015841721, pvalue=0.0) |
|  |

# EDA (Exploratory Data Analysis)

**3 Types of EDA**

## Uni Variate

## Bi Variate

## Multi Variate

EDA is always done of historical data. It is root cause analysis of what happened in historical data.

1. By using EDA analysis we can understand the data
2. Understand the business importance of a given problem
3. Drawing the insights based on the given data
4. Creating the new variables or flags with the help of hypothesis testing
5. Summarizing the historical data
6. Prerequisites (Steps)

* Load the data into puthon
* Check # of records and variables
* Check for any Duplicates at row level
* Understand the variables
* Write down the generic description of each and every variable
* Produce the summary statistics of both Numerical and categorical variable
* Missing value
* Outliers
* master datasets

Machine generated alternative text:
Train Data 
Business Objective 
Data Gathering 
Master Data 
Size > 1 M 
Stratified Sampling 
Test Data 
Remove duplicates 
Clean Data 
(Missing, Outlier, Check Normal 
distribution, EDA: Hypothesis 
testing) 
SRS Sampling 
Train (80%) 
Remove duplicates 
Clean Data 
(Missing, Outlier, Check Normal 
distribution, EDA: Hypothesis testing) 
Test (20%) 

# Variable Reduction or Dimensionality reduction or Feature Importance

## Variable Reduction Techniques

* Any parameter with> 95% missing we discard that (this is 1st variable reduction technique
* Low variance (all the values are similar in nature) close to 0 variance. Identical variables
* Correlation (Pair wise correlation)
* VIF (variance influence factor) or Multicollinearity
* p-value
* Stepwise regression
* z-value
* Information value
* Extra tree classifier
* Random forest classifier
* PCA (Principal Component Analysis)

## Variable reduction technique (wrt continuous variable).

if variables are less than 30, then follow below step and drop columns one by one

1. Missing value % (> 95% missing value then drop the variable)
2. Low variance (if less than 1 then drop)
3. Pairwise correlation (< 0.5 then drop the variable)
4. VIF (> 2 then drop)
5. P-Value (> 0.05 then drop)
6. T-Value (< 5 then drop)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Drop if | | |
| Metrics | Variables < 30 (**Filter Methods**) | Variables < 100 (**Wrapper Methods**) | 100 < Variables < 500 (**Embedded Methods**) |
| High Missing value % | > 95% | > 95% | > 95% |
| Low variance | < 1 | < 1 | < 1 |
| Low Pairwise correlation | < 0.5 | < 0.5 |  |
| High VIF | > 2 | > 2 |  |
| High P-Value | > 0.05 | > 0.05 | > 0.05 |
| Low T-Value | < 5 | < 5 | < 5 |
| Stepwise Regression | NA |  |  |
| Lasso | NA |  |  |
| PCA |  |  | < 0.5 |

## Classification model

* Information value
* Feature Importance (gini, entropy)
* Pairwise correlation
* VIF
* P-Value
* Chi-Square
* RF-variable, Rf feature\_importance\_ or AUC score or ExtraTree feature\_importance\_
* z-Value (T-Value or Step Wise)
* AUC/ROC

## Regression model

* Pairwise correlation
* Stepwise regression
* P-Value
* VIF
* PCA
* Lasso
* F-stat/ATC
* Finalize the variable

Any automated tool (Knime/IBM Watson/IQ) for variable reduction uses below variable reduction techniques

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | AUC | KS (K-stat) | IV (Information Value) | Feature Importance |
| x1 |  |  |  |  |
| x2 |  |  |  |  |
| x3 |  |  |  |  |
| x4 |  |  |  |  |

## VIF (Variance influence factor or multi collinearity)

**VIF (Variance influence factor or multi collinearity)**

How one independent variable is influencing the other independent variable or how one independent variable is influenced by the other independent variable.

Suppose we have 4 variables x1,x2,x3,x4

|  |  |  |
| --- | --- | --- |
|  | ~ influence |  |
| x1 | x1~x2 | x1:x2:  = 0.05  r = 0.05 |
| x2 | x1~x2+x3 | x1:x2+x3:  = 0.7  r = 0.7 |
| x3 |  |  |
| x4 |  |  |

VIF =

If VIF is high then we will drop that independent variable

If = 1 which means variance of independent variable is explaining the variance of dependent variable which mean both are same then VIF = infinity then we are going to drop the independent variable

|  |  |
| --- | --- |
|  | Pairwise Correlation |
| X1-y | 0.8 |
| X2-y | 0.3 |
| X1-X2 | 0.7 |

Here we are going to drop X2 as it has pairwise correlation as 0.2 which is less than 0.5

## Information Value

Information value will give us the predicting power of each and every independent variable

* It works only for linear data
* Works for both categorical as well as numerical variable
* It works only for classification problems (dependent variable is binary)

* The best IV value ranges from 0.02 to 0.5
* If 0.02 <IV < 0.1 --> Low Predicting power
* If 0.1 <IV < 0.3 --> Medium Predicting power
* If 0.3 <IV < 0.5 --> Strong Predicting power
* If 0.5 <IV --> Suspicious variable,
* If IV < 0.02 --> not useful for analysis

### Weight of events

WoE = LN()

It can be positive or negative

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable | non-Events | vents | %of NonEvents | %of Events | WOE |
| 3 | 1053 | 2315 | 0.024987542 | 0.040011061 | -0.47078 |
| 4 | 38378 | 53669 | 0.91070454 | 0.927582571 | -0.01836 |
| 5 | 2650 | 1868 | 0.062884127 | 0.032285383 | 0.666679 |
| 6 | 60 | 7 | 0.001423792 | 0.000120984 | 2.465422 |
|  | 42141 | 57859 |  |  |  |

In excel use LN formula

WOE =LN(F2/G2)

'WOE - 
- np. "percent _ nonevents"] 
abc 1 
active_cust 
N um_complaints 
1053 
38378 
2660 
1 
2315 
0.024988 
53660 
og1071Y 
1868 
0.001424 
percent_events 
0040011 
0927583 
0032285 
0000121 
_0470779 
-0.018363 
0.666679 
2.465422 

### Binning

**Binning is done for Numerical Variables**

#### Fine class binning

* + **Before we do binning, data must not have missing value or outliers**
  + Divide the data into 10 groups (decile). On top of that we calculate WOE. If it is not in ASC or DESC order then we go for course Binning.
  + The process of creating WOE or Bins is also called **Fine class Binning**
  + It can be derived thru WOE
  + If WOE is in ascending or descending order then our data is linear in nature

#### Course Binning (Optimal Binning)

* + Wherever WOE breaks (not following Asc or Desc), we club two bins in which it breaks and recalculate WOE to get optimal Binning and Course Binning

Fine Class Binning 
Optimal Binning or Course Binning 
Bins 
WOE 
0.3 
0.5 
1.5 
0.9 
Bins 
WOE 
0.3 
0.5 
1.2 

**Information Value (IV) = (**percent\_nonevents - percent\_events)\*WoE

In classification Models we have 2 types of models

1. Equation based models
2. Tree based models

Data in classification Models (before we build the model)

* Numerical (discrete/continuous)
  + Divide the variables into bins (1 to 10)
  + Convert it into binary format
* categorical --> Dummy coding

## PCA

**Principal component analysis (dimensionality reduction technique)**

1. When we have small train dataset then model will always be overfit.
2. Multi collinearity (one independent variable is influencing other independent variable)
3. A set of independent variable can influence another independent variable

If we have very large number of variables then we use PCA.

Except PCA, if we apply any other variable reduction technique, magnitude, direction and name of the variable does not change.

In PCA, variable name changes. If data set has n independent variable, it's going to create n components.

* Data should not have missing values
* No outliers

### When to apply MinMax scalar vs Standard scalar

|  |  |
| --- | --- |
| MinMax scalar | Standard scalar |
| Gradient Descent | Linear Regression (simple, Multi) |
| Regularization | Logistics Regression |
| NN (ANN, DNN, KNN) | Decision Tree |
| SVM |  |
| Boosting |  |
| Bagging |  |
| random forest |  |
| Naïve Bayes |  |

### Steps to perform PCA

1. **Standardization of data** (z = (x-µ)/σ), standard scalar (for ML), minmax scalar for deep learning
2. Units of each variable should be same
3. Calculate **co-variance matrix** (if we have n variables then it will be nxn matrix (how one variable is influencing other variable, unit bound should be same)

Sample covariance

|  |  |  |
| --- | --- | --- |
|  | xa | xb |
| xa | 100 | 50 |
| xb | 150 | 120 |

(Xa,Xa) and (Xb,Xb) are variance. (Xa,Xb) and (Xb,Xa) are covariance.

1. Find **Eigen value.** If we have nxn matrix, we will get n eigen values and for each eigen value we will have 2

Eigen value is nothing but vectors

Suppose we have A = nXn matrix

Det(A-λI) = 0

= (1-

= 16+12

Look for integer values ±1,±2,±4,±8,±16

Say   
After solving the range values of

Using

Whichever variable has highest variance that will be 1st variable to be considered

We require PSA and linear regression

## Chi-Square

Chi Square test: To check the independence b/w two categorical variables. To check the correlation b/w two categorical variables, we are going to use Chi Square.

Two check the correlation b/w two numerical variables we use Pearson correlation.

### Pearson correlation ( r )

At this point we are going to use the correlation b/w 2 numerical variables. So finding out the relation b/w 2 numerical variables is called correlation.

**Pre-requisites**

* Relationship b/w the 2 variables is linear in nature.
* Either directly or inversely proportional)
* The 2 variables must be continuous or discrete variable.
* They must be normally distributed

r= cov(x,y)/std(x) \* std(y) =

r--> coefficient of correlation

If two variables are categorical variables (binary and numeric) then we are going to use the Point bi-serial correlation. This is nothing but pairwise correlation.

S = Variance of age

p= event

q = non-event

### Formula for Chi-square

Observe is

If Chi square is

Greater than 100% which means one variable is independent of other variable.

Less than 100% which means one variable is dependent on other variable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | very | Pretty | low | Total |
| Above | 272 | 294 | 49 | 615 |
| Avg | 454 | 835 | 131 | 1420 |
| Below | 185 | 527 | 208 | 920 |
|  | 911 | 1656 | 388 | 2955 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 189.599 | 344.6497 | 80.75127 |  |
|  | 437.7733 | 795.7766 | 186.4501 |  |
|  | 283.6277 | 515.5736 | 120.7986 |  |
|  |  |  |  |  |
| Sum of all is Chi Square | 35.81204 | 7.44349 | 12.48455 | 172 |
|  | 0.601469 | 1.933295 | 16.4908 |  |
|  | 34.29648 | 0.253237 | 62.94836 |  |

**Comparing Two categorical variables**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | var1 | var2 | var3 | var4 | Total |
| Above | 752 | 2815 | 1471 | 169 | 5207 |
| Avg | 5246 | 13770 | 7038 | 1160 | 27214 |
| Below | 853 | 6617 | 4792 | 528 | 12790 |
|  | 6851 | 23202 | 13301 | 1857 | 45211 |
|  |  |  |  |  |  |
|  | 789.0371 | 2672.2 | 1531.891 | 213.8727 |  |
|  | 4123.844 | 13966.05 | 8006.313 | 1117.79 |  |
|  | 1938.119 | 6563.747 | 3762.796 | 525.3374 |  |
|  |  |  |  |  |  |
| Sum of all is Chi Square | 1.738509 | 7.631154 | 2.420322 | 9.414759 |  |
|  | 305.3544 | 2.752161 | 117.1113 | 1.593944 |  |
|  | 607.539 | 0.432047 | 281.5087 | 0.013495 |  |
|  | 914.632 | 10.81536 | 401.0404 | 11.0222 | 1337.51 |

## Correlation (Pair Wise correlation)

Variance =

**Variance and standard deviation can never be -ve.**

Co-Variance --> How two random variables are variating with each other

* In order to calculate the co-variance 2 variable must have same units
* Its unit bound
* It can be -ve or +ve
* If co-variance is +ve --> Two variables are directly proportional
* If co-variance is -ve --> Two variables are inversely proportional

Sample covariance

Population covariance =

**Correlation** --> Finding out the relation b/w the 2 variables (2 numerical, 2 categorical, 1 numerical and 1 categorical variables)

### Pearson correlation (r)

At this point we are going to use the correlation b/w 2 numerical variables. So finding out the relation b/w 2 numerical variables is called correlation.

**Pre-requisites**

* Relationship b/w the 2 variables is linear in nature.
* Either directly or inversely proportional
* The 2 variables must be continuous or discrete variable.
* They must be normally distributed

r=

r--> coefficient of correlation

Correlation --> Data is standardize and Correlation will be unit free (all the variables having same unit)

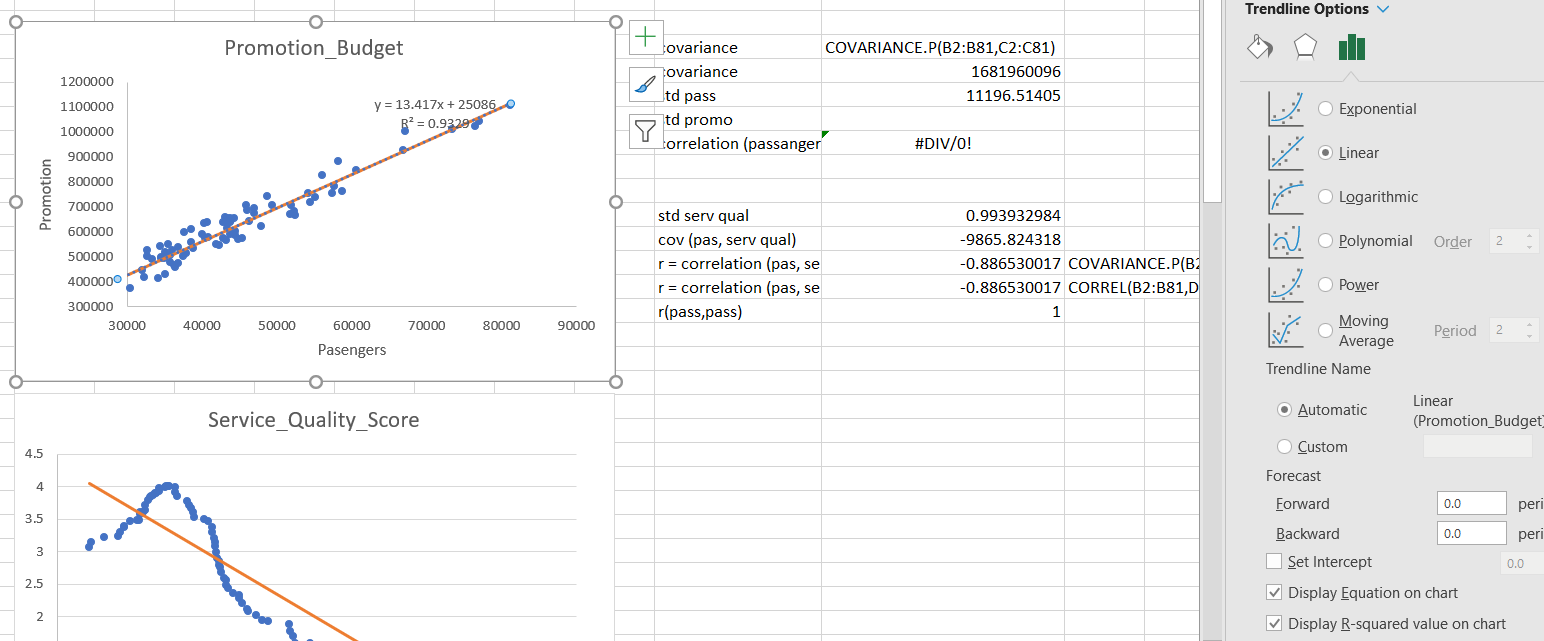
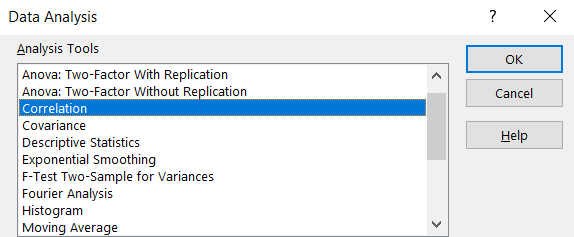
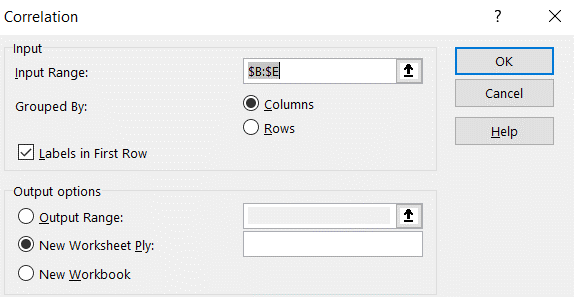
* Correlation is unit free
* Range of correlation is -1 ≤ r ≤ +1

|  |  |
| --- | --- |
| Value of r | correlation b/w the 2 variables |
| 0 | No linear relationship |
| 0 to 0.25 | Negligible +ve |
| 0.25 to 0.5 | Weak +ve |
| 0.5 to 0.75 | Moderate +ve |
| 0.75 to 1 | Strong +ve |
| +1 | Exact +ve (directly proportional) |
| -0 to -0.25 | Negligible -ve |
| -0.25 to -0.5 | Weak -ve |
| -0.5 to -0.75 | Moderate -ve |
| -0.75 to -1 | Strong -ve |
| -1 | Exact -ve (inversely proportional) |

Drawbacks of correlation -->

* To perform we should have atleast 30 observations (b/c data should be normally distributed)
* We can rely on coefficient of correlation ( r ) of small dataset
* Outliers could affect the coefficient of correlation ( r )
* By using the correlation, we can not measure the cause and effect
* To measure the cause and effect, we are going to use regression method

### Excel Formula

* + COVARIANCE.P(B2:B81,C2:C81)
  + CORREL(B2:B81,D2:D81)
  + STDEV.P(C2:C81)
* Scatter plot--> to check the linear relation b/w 2 variables
* 
* Go to Data—Data Analysis
* 
* 
* Passengers promotion_ Budget 
  Service 
  Passengers 
  Promotion_Budget 
  0.965851028 
  -0.886530017 
  0.491946803 
  -0.843382312 
  O. 531448944 
  Quality_Score 
  _0.443437013 
* As Promotion\_Budget increases then Passengers increases

### Python Code

|  |
| --- |
| # check covarianve  abc = np.cov(air["Passengers"],air["Promotion\_Budget"])  abc  array([[1.26948787e+08, 1.70325073e+09],  [1.70325073e+09, 2.44967415e+10]])  sns.heatmap(abc,annot=True,square=True) # annotation by default is false, by default is rectangle, we can make it square  air["Passengers"].var()  126948786.63227847  air["Promotion\_Budget"].var()  24496741542.053165  abc\_r = np.corrcoef(air["Passengers"],air["Promotion\_Budget"])  abc\_r  array([[1., 0.96585103], [0.96585103, 1.]]) |
|  |

# Simple Linear Regression

**Regression:**

(simple linear regression with only 2 variables, 1 dependent and 1 independent variable)

To measure the cause and effect b/w two variables we use regression.

In Regression we do not give equal importance to two variables (independent and dependent variables)

X = Independent variable (feature) --> Cause

Y = Dependent variable (Label) --> effect

x = temp (cause)

Y = ice-cream sales (effect)

**y = mx + c**

y= label or dependent variable

c= constant

m= slope

x=feature or independent variable

Convert mathematical equation into statistical equation

y=

x=feature or independent variable

y= label or dependent variable

Machine generated alternative text:
Y 
y error 
slope 
Predicted value 
intercept 

## Assumptions of linear regression

* Two variables must be numerical variables or continuous variables
* Dependent variable must be normally distributed (in case of multi linear regression)
* For simple linear regression both the variables must be normally distributed
* Two variables must be linear in natures
* Two variables must have some variance (should not have 0 variance)
* Variance of error (should be constant. Error should be normally distributed

*Predicted equation*

Actual value y =

## Excel Formula

In AirPassengers data set

Promotion\_Budget is cause (independent variable) and Passengers is effect (dependent variable)

Take Promotion\_Budget and Passengers into new excel (say AirPassenger\_Regression)

Keep Independent variable first and then keep dependent variable

Machine generated alternative text:
Promotion Budget 
517356 
646086 
638330 
506492 
609658 
476084 
635978 
495152 
429800 
Passengers 
37824 
43936 
42896 
35792 
38624 
35744 
40752 
34592 
35136 

* Check scatter plot (compare 2 variables and check normally distributed and linear in nature)

Machine generated alternative text:
55000 
9tmoo 
Promotional Budget 
1100000 
130moo 

On trend line

Machine generated alternative text:
o 
Exponential 
• Linear 
o 
Logarithmic 
o 
Polynomial 
o 
Power 
Moving 
Average 
Trendline Name 
@ Automatic 
o 
Custom 
Forecast 
Fomard 
BacQvard 
C] Set Intercept 
Order 
Period 
Linear (Passengers) 
periods 
periods 
Dis 
uation on cha 
Display R-squared value on chart 

In Excel

|  |  |
| --- | --- |
| B0 = INTERCEPT(dependent var(y), independent var(x)) | 1259.605832 |
| B1 = SLOPE(dependent var(y), independent var(x)) | 0.069529685 |

Even if no promotional\_budget is allocated, avg passenger is 1259

With every increase in promotional\_budget increase in passenger is 0.069529685

Machine generated alternative text:
= 0.0695x + 1259.6 
= 0.9329 
300000 500000 700000 900000 1100000130moo 
Promotional Budget 

Machine generated alternative text:
Format Trendline 
Trendline Options 
o 
Exponential 
Linear 
o 
Logarithmic 
o 
Polynomial 
o 
Power 
Moving 
Average 
Trendline Name 
@ Automatic 
o 
Custom 
Forecast 
Fomard 
BacQvard 
C] Set Intercept 
Order 
Period 
Linear (Passengers) 
periods 
periods 
@ Display Equation on chart 
@ Display R-squared value on chart 

## Interception of coefficient

Slope

* Estimated Y changes by for each 1 unit increase in X
  + If

Y-Intercept

* Average value of Y when X = 0
  + If

## OLS --> Ordinary Least squared estimation

SSE : (Sum of squared error)

*Predicted equation*

Actual value y =

SSR (sum of squared regressor) =

SST (Total sum of square) =

SST = SSR + SSE

SSR --> Model is able to predict

SSE --> Model is not able to predict

0 ≤ 1

Using data analysis tab

Machine generated alternative text:
c 
Promotion Budget Passengers 
517356: 
646086' 
638330: 
506492: 
609658' 
476084: 
635978\ 
495152} 
429800: 
613326\ 
4927581 
600726: 
456960\ 
586096' 
704802 : 
536970: 
742308' 
500234: 
570682: 
826420' 
761040: 
37824 
43936 
42896 
35792 
38624 
35744 
40752 
34592 
35136 
43328 
34960 
44464 
36464 
44464 
51888 
36800 
48688 
37456 
56032 
58800 
Regression 
Input 
Input Y Range: 
Input Range: 
Labels 
Confidence Level: 
Output options 
C) Output Range: 
New Worksheet ely: 
O New Workbook 
Residuals 
Residuals 
Standardized Residuals 
Normal Probability 
Normal Probability Plots 
S8S1:S8S81 
SAS1:SAS81 
Constant is Zero 
Residual Plots 
Line Fit Plots 
x 
Cancel 
Help 

Machine generated alternative text:
SUMMARY OUTPUT 
Regression Statistics 
Multiple R 
R Square 
Adjusted R Square 
Standard Error 
Observations 
o. 965851028 
0.932868209 
0.932007545 
2937.951609 
80 
coefficient of correlation strong correlation b/w x and Y 
93% of the time variance of x and y is same 
used in multiple linear regression 
Total error of given data 
num of observation 
sqrt(SSE/(n 
SSE = sum of square error 
n = num of observations 
k = num of independent 
variable 
2937.951609 

Machine generated alternative text:
ANOVA 
Regression 
Residual 
Total 
Analysis of variance 
MS 
MSR 
9355692490.83145 
MSE - 
8631559.65536607 
F (it is generally used in multi linear regression) 
1083.893626 
Significance F 
(P value) 
1.6573E-47 
(SSR) 
1 
78 (SSE) 
79 (SST) 
= 9355692490.83145 
= 673261653.118554 
= 10028954143.95 

F (it is generally used in multi linear regression) : To measure the goodness fit of the model we use F static. We can not use the standalone F-static value. Higher F value, better the model is

F-static = MSR/MSE

MSR = Mean square regressor

MSE = Mean square error

Machine generated alternative text:
Intercept 
Promotion Budget 
Coefficien 
1259.606 
0.06953 
Standard Error 
1361.070807 
0.002111917 
t Stat (z-score) 
0.925452097 
32.92253978 
p_ value 
0.357585257 
1.6573E-47 
Lower 95% 
-1450.077995 
0.065325181 
pper 95% 
3969.29 
0.073734 
Lower 95.0% 
-1450.077995 
0.065325181 
Upper 95.0% 
3969.289659 
0.07373419 

Check for following to select independent variables (The key check points we need to check)

* P-Value
* Coefficients of an equation
* F-Static
* Adj-Rsquare Value
* Rsquare value
* Standard error

## Using Python

# if we have more independent variable then use + to add more variables. Before tilda dependent variable

|  |
| --- |
| either use scikitlearn or statsmodels.formula.api for regression  \* Ordinary Least squared estimation (OLS)  import statsmodels.formula.api as sm  model = sm.ols(formula="Passengers~Promotion\_Budget",data=air).fit() \  # if we have more independent variable then use + to add more variables. Before tilda dependent variable  model.summary() |

Machine generated alternative text:
import statsmodels . formula. api as sm 
, data—air) . fit() \ 
model : 
# if we have more independent variable then use + to add more variables. 
model. summary( ) 
Before tiLda dependent variable 

Machine generated alternative text:
In 
either use scikitlearn or statsmodels.formula.api for regression 
• Ordinary Least squared estimation (OLS) 
import statsmodels. formula. api as sm 
model 
sm. 01s (formula= 'V PassengersæPromotion_Budget" , data-air ) . fit() 
model. summary( ) 
OLS Regression Results 
Dep. Variable: 
Date: 
Time: 
No. Observations: 
Dt Residuals: 
Of Model: 
Covariance Type: 
Intercept 
Promotion_Budget 
Passengers 
OLS 
Least Squares 
sat, 02 Nov 201g 
11 15:57 
nonrobust 
R-squared: 
Adj. R-squared: 
F -statistic: 
Prob (F-statistic): 
Log-Likelihood: 
AIC: 
BIC: 
coef 
125g6058 
00695 
std err 
1361 071 
0 002 
0 925 
32 923 
p>ltl 
0 358 
0 000 
0 933 
1_66e-47 
-751.34 
1507 
1511 
[0.025 
-1450078 
0065 
0.975] 
290 
0 074 
Omnibus: 
Prob(Omnibus): 
Skew: 
Kurtosis: 
Warnings: 
26 624 
0 000 
-0 128 
1 77g 
Durbin-Watson: 
Jarque-Bera (JB): 
Prob(JB): 
Cond. No. 
1 831 
5188 
0.0747 
2_67e+06 
[1] Standard Errors assume that the covariance matrix ot the errors is correctly specified. 
[2] The condition number is large, 2.67e+O& This might indicate that there are 
strong multicollinearity or other numerical problems. 

* R-squared --> 93% of the time variance is same for dependent (y) and independent variable (x)
* F\_static to measure goodness of the model . We will use this to compare 2 or more models
* Covariance Type --> Nonrobust --> it is equation model. It does not change dynamically. Based on equation only will it change
* All ML models are robust models. Since above are equation-based model hence it is non-robust model
* t-z-score
* JB-Test : to check whether a variable is normally distributed of not. But in regression models JB test will help us to understand whether variance of error is constant or not(normally distributed or not).
* Based on JB P value we can decide whether it is significant or not.
  + If P-Value ≤ 0.05 then variance of error is constant. It is also called as **Homoscedasticity**
  + If P-Value > 0.05 then it is called **Heteroscedasticity**. Variance of error is not constant

* **Durbin-Watson** --> To measure the auto correlation (serial correlation) b/w the individual data points of an error variable.
  + If DW < 2 positive correlation
  + If DW > 2 negative correlation
  + If DW = 2 no correlation

e.g 10,20,30,40 how 10 is influencing 20. how 20 is influencing 30. If adjacent data points are influencing each other then they have auto or serial correlation.

So for DW, it should not have auto correlation. In time-series analysis we use auto-correlation

Formula

Omnibus

How much model is able to predict correctly?

# Multi Linear Regression

When we have more than two variables, we sue Multi Linear Regression

**Drawback of simple Linear regression**

* Outcome depends on multiple factors, but in simple linear regression we cannot predict such outcomes as it considers single factor

**Equation for Multi Linear Regression**

Y =

*or intercept*

..

x1=feature or independent variable one

y= label or dependent variable

**Pre-requisites of MLR**

* Each and every variable must be continuous variable
* Dependent variable must be normally distributed
* Every independent variable must be independent with the other independent variables
  + correlation b/w two independent variable should be weak
  + correlation b/w dependent and independent variable should be strong
* The variable which has less influence wrt dependent variable should be dropped
* Checking the correlation b/w dependent and independent variable is called **pairwise correlation**
* The relation b/w dependent and independent variable **(pairwise correlation) should be 0.5**

  Correlation b/w

* x1 & x2 weak
* x1 : y strong ≥ 0.5

**If we want to use multilinear regression below variable reduction technique should be applied:**

1. Missing value % (drop if 95% missing values)
2. Low variance. If 2 independent variables have variance less than 1 then drop one of the variables
3. **Pair wise correlation**, (y, Xn) should be > 0.5. If it is < 0.5 then drop the variable)
4. **VIF Variance Influence Factor or multi collinearity**. If VIF is ≥ 2 we are going to drop the variable

## Pairwise Correlation:

* Checking the correlation b/w dependent and independent variable is called **pairwise correlation**
* The relation b/w dependent and independent variable **(pairwise correlation) should be 0.5**

  Correlation b/w

* x1 & x2 weak
* x1 : y strong ≥ 0.5

|  |  |
| --- | --- |
|  | **Pairwise Correlation** |
| X1-y | 0.8 |
| X2-y | 0.3 |
| X1-X2 | 0.7 |

Here we are going to drop X2 as it has pairwise correlation as 0.2 which is less than 0.5

## VIF (Variance influence factor or multi collinearity)

How one independent variable is influencing the other independent variable or how one independent variable is influenced by the other independent variable.

Suppose we have 4 variables x1,x2,x3,x4

|  |  |  |
| --- | --- | --- |
|  | ~ influence |  |
| x1 | x1~x2 | x1:x2:  = 0.05  r = 0.05 |
| x2 | x1~x2+x3 | x1:x2+x3:  = 0.7  r = 0.7 |
| x3 |  |  |
| x4 |  |  |

VIF =

If VIF is high then we will drop that independent variable

how variance of independent variable is explaining the variance of dependent variable .

If = 1 , means both are same then VIF = infinity then we are going to drop the independent variable

## Variable reduction technique (w.r.t. continuous variables)

if variables are less than 30, then follow below step and drop columns one by one

1. Missing value % (> 95% missing value then drop the variable)
2. Low variance (if less than 1 then drop)
3. Pairwise correlation (< 0.5 then drop the variable)
4. VIF (> 2 then drop)
5. P-Value (> 0.05 then drop)
6. T-Value (< 5 then drop)

|  |  |
| --- | --- |
| **Metric** | **Drop if** |
| High Missing value % | > 95% |
| High VIF | > 2 |
| High p-Value | > 0.05 |
| Low t-Value | < 5 |
| Low variance | < 1 |
| Low Pairwise correlation | < 0.5 |

|  |  |
| --- | --- |
| Number of Variables < 30  **(Filter Methods)** | Drop if |
| Missing value % | > 95% |
| Low variance | < 1 |
| Pairwise correlation | < 0.5 |
| VIF | > 2 |
| P-Value | > 0.05 |
| T-Value | < 5 |

|  |  |
| --- | --- |
| Number of Variables < 100  **(Wrapper Methods)** | Drop if |
| Missing value % | > 95% |
| Low variance | < 1 |
| Pairwise correlation | < 0.5 |
| VIF | > 2 |
| P-Value | > 0.05 |
| T-Value | < 5 |
| Stepwise Regression |  |
| Lasso |  |

|  |  |
| --- | --- |
| 100 < Number of Variables < 500  **(Embedded Methods)** | Drop if |
| Missing value % | > 95% |
| Low variance | < 1 |
| PCA | < 0.5 |
| P-Value | > 0.05 |
| T-Value | < 5 |

## Data requirements for multiple linear regression

* Atleast we need to have 400 records/observations
* In final model or data we must have 3 independent variables
* If after all the reduction we got less than 3 independent variables then we go for simple linear regression

Drop Inter\_metro\_flight\_ratio as P<|t| (0.350) > 0.05

We also need to check F-static value. 495

**Higher F value, better the model is**

Machine generated alternative text:
In 
[26] : 
import 
model = 
# here 
modell 
statsmodels.formula.api as sm 
Budget+Service Quality Score+lnter metro flight ratio", data=airl) 
just for examp,we are using atL 3 columns but ideatLy promotion Budget would be dropped as it has high VIF 
outC26]: 
modell = model. fit() 
. summary 
OLS Regression Results 
Dep. Variable: 
Model: 
Method: 
Date: 
Time: 
No. Observations: 
Df Residuals: 
Of Model: 
Covariance Type: 
Passengers 
OLS 
Least Squares 
Sun, 
R-squared: 
Adj. R-squared: 
F-statistic: 
Prob (F-statistic): 
Log-Likelihood: 
10 Nov 2019 
80 
76 
3 
nonrobust 
coef 
1821e+04 
0.0555 
-28020708 
-20034508 
0.951 
0.949 
4956 
871+50 
-73845 
1485. 
1494. 
Intercept 
Promotion_Budget 
Service_Quality_Score 
Inter_metro_flight_ratio 
std err 
3542.694 
0.004 
530.382 
2129.095 
AIC: 
BIC: 
5.424 
15.476 
-5.283 
-0.941 
2.312 
2.759 
0.252 
p>ltl 
0.000 
0.000 
0.000 
0.350 
[0.025 
1 _22e+04 
0.048 
-3858.419 
-6243.912 
0.975] 
2.63e+04 
0.063 
-1745.723 
2237.010 
Omnibus: 
Prob(Omnibus): 
Skew: 
Kurtosis: 
6.902 
0.032 
-0.051 
2.096 
Durbin-Watson: 
Jarque-Bera (dB): 
Prob(JB): 
Cond. No. 
8.22e+06 

## Step-wise Regression: Finalize the model (4 parameters)

### F-Static

To measure the goodness fit of a model. Higher it is better the model is

F-Static = MSR/MSE = mean square regressor/mean square error

MSR = SSR/k = (sum of squared regressor)/k

k = independent variables

n num of observations

SSR is sum of squared regressor

SSE is Sum of squared error

MSE = SSE/(n-k-1)

F-Static = (SSR/k ) \* (n-k-1)/SSE

**F-Static = SSR/SSE \* (n-k-1)/k**

**F-Static = O{ (n-k-1)/k}**

**O is omnibus = SSR/SSE**

### Adj. R-squared

: Variance  explained of independent variable wrt to percentage of Variance  explained of dependent variable (coefficient of determination or

R squared). It is the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

0 ≤ 1

SST = SSR + SSE

SST is Total Sum of Square

As num of independent variable are increasing value reaches close to 1. As value increases, the variance of the model also increases, then it leads to increase in standard error. As standard error increases the accuracy of the model decrease. To overcome this issue, we are going to use adj-

adj-

adj-

### t-Value

It is z-score value

Z = (X - µ)/

### AIC

**Akaiye Information Criteria**

How much amount of information we are going to loose from this model

AIC = 2K - 2log(L)

K = independent variable

L = log likelihood or maximum likelihood

**Check all the above for each iteration (stepwise regression) to validate if we need to remove any variable.**

**Iteration-1:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | t-value | p-value | F-Static | AIC | adj-R^2 |
| x1 | abs(20) | 0.05 | 750 | 400 | 0.95 |
| x2 | abs(-8) | 0.05 |
| x3 | abs(-30) | 0.05 |
| x4 | abs(4) | 0.05 |
| x5 | abs(25) | 0.05 |

Here t-value of x4 is min. Hence we drop x4 and iterate building the model

**Iretation-2 (after dropping x4 )**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | t-value | F-Static | AIC | adj-R-squared | p-value |
| x1 | abs(20) | ≥ 750    if F-static increases then drop  (should increase for next iteration or else revert back removal of x4) | ≤400    if AIC decreases then drop  (should decrease for next iteration or else revert back removal of x4) | ~0.95     if adj-R^2 is close to earlier value then don’t drop  (should change minimally for next iteration or else revert back removal of x4) | 0.05 |
| x2 | abs(-8) | 0.05 |
| x3 | abs(-30) | 0.05 |
| x5 | abs(25) | 0.05 |

Machine generated alternative text:
import 
model = 
# here 
# here 
statsmodels.formula.api as sm 
sm. Budget+Service Quality Score" , 
data=airl) 
just for examp,we are using aLL 3 columns but ideatLy promotion Budget would be dropped as it has high VIF 
we are now dropping Inter metro_f[ight ratio 
outC27]: 
modell = model. fit() 
modell. summary() 
OLS Regression Results 
Dep. Variable: 
Model: 
Method: 
Date: 
Time: 
No. Observations: 
Df Residuals: 
Of Model: 
Covariance Type: 
Passengers 
OLS 
Least Squares 
sun, 10 Nov 2019 
R-squared: 
Adj. R-squared: 
F-statistic: 
Prob (F-statistic): 
Log-Likelihood: 
0.951 
0.950 
744.0 
438+51 
-738.91 
1484. 
1491. 
Intercept 
Promotion_Budget 
Service_Quality_Score 
09:5862 
80 
2 
nonrobust 
coef 
1853e+04 
0.0544 
-2807.3095 
std err 
3464.796 
0.003 
529.958 
AIC: 
BIC: 
5.348 
16.063 
-5.297 
2.331 
2.913 
0.233 
p>ltl 
0.000 
0.000 
0.000 
[0.025 
1.16e+04 
0.048 
-3862.592 
0.975] 
2.54e+04 
0.061 
-1752.028 
Omnibus: 
Prob(Omnibus): 
Skew: 
Kurtosis: 
7.728 
0.021 
-0.043 
2.069 
Durbin-Watson: 
Jarque-Bera (dB): 
Prob(JB): 
Cond. No. 
7.97e+06 

## Gradient Descent

In order to reduce the error or residuals of the model we are going to apply machine learning models

w = weights

Internally the weights value are -3 ≤w ≤+3

It assumes the data is normally distributed data

Error or residuals

Machine generated alternative text:



y -

m= num of observations

For each record we are going to get the error

**Quadratic cost function or loss function**

Summation of MSE =

Our goal is to minimize the mean squared error (MSE). To minimize MSE we can modify weights.

**Feed forwarding:** The process of multiplying i/p data with weights to get

**Backward propagation:** Checking error to minimize the MSE we will give weights to the i/p data.

To further improve the model we use **gradient descent** apart from above 4 factors

The Backpropagation **algorithm** looks for the minimum value of the error function in weight space using a technique called the **delta rule or gradient descent.** The weights that minimize the error function is then considered to be a solution to the learning problem

Machine generated alternative text:



By default, value will be initially taken as 1

Why initially if one of the i/p weight is 0 , the contribution of weight will be 0. So there are higher chances that error will increase. To reduce some error we are adding the bias term

Error needs to decrease that’s the intent of gradient descent

To reduce the error inconsistency we introduce the term **alpha α (learning rate)**

0 ≤ α ≤ 1

When error is inconsistent, then to decrease the error in the model we are adding the learning rate, α

Iterate the w1 and w0 values till the point we get the difference b/w two MSE values as 0.01

That point is called the **tolerance point**

**Derivation of above equations**

Say z = 3

z=

=

=

=

Optimizing the

Do the same equation for beta 0 and beta 1

RMSE: root mean squared error

Machine generated alternative text:
Train Data 
Model Building 
MSE 
RMSE 
Business Objective 
Data Gathering 
Data processing 
EDA 
ariable Reduction 
Pairwise VIF 
R-Value 
-Value 
Diff > 
Test Data 
MSE 
RMSE 
Regularization 
Gradient Descent 
Improve 

# Regularization

If model is performing well on training data but not on test data set then we use regularization

* Ridge**(L2 Regularization)**: **to understand ridge and Lasso equations**
* Lasso**(L1 Regularization)** --> Minimizing the abs value of coefficient to 0. It is one of the variable reduction techniques, to remove the variables having abs coefficient value as 0. In general Lasso is applicable when we have more than 1lac records. But atleast 10K records are required to apply the Lasso
* Elastic Net

To reduce the overfitting we use regularization

## Ridge Regularization (L2 Regularization)

If we would like to remove the overfitting the we can use ridge

SSE =

Gaussian marker missing model

SSE = we ass the weights)

SSE =

Gradient descent: we iterate the equation based on alpha value to get the desired tolerance

To remove the insignificant variables we are adding the weights to the coefficients. The coeff values which are close to 0 the impact of those variables is very less then we can drop those variables from the data. In order to perform this we use Lasso regression

If p = 2 --> Ridge

If p = 1 --> Lasso

If p = 0 --> Elastic Net

Types of Derivates

* Normal derivates
* Partial derivates
* Chain rule derivates

## Lasso Regularization(L1 Regularization)

If we would like to remove the insignificant variables then we can use Lasso

Minimizing the abs value of coefficient to 0. It is one of the variable reduction techniques, to remove the variables having abs coefficient value as 0. In general Lasso is applicable when we have more than 1lac records. But atleast 10K records are required to apply the Lasso

Weights are between -3 to +3 as data is linear regularized. This is the initial value Linear regression starts with.

Lasso chooses 100 alpha values

If difference in score or accuracy b/w train and test is more than 10% then we apply Ridge or Lasso regularization

### Cross Validation method

To check for the best alpha value

Gradient descent: we iterate the equation based on alpha value to get the desired tolerance

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Total num 1000 Rec | sample | Model | Train | Test | alpha=2 | alpha=3 | alpha=100th value |
| 100 Rec | S1 | m1 | S1-S9 | S10 | on test data build m1 | on test data build m1 | on test data build m1 |
| 100 Rec | S2 | m2 | S1-S8,S10 | S9 | on test data build m2 | on test data build m2 | on test data build m2 |
| 100 Rec | S3 | m3 |  | S8 | on test data build m3 | on test data build m3 | on test data build m3 |
| 100 Rec | S4 | m4 |  | S7 | on test data build m4 | on test data build m4 | on test data build m4 |
| 100 Rec | S5 | m5 |  | S6 | on test data build m5 | on test data build m5 | on test data build m5 |
| 100 Rec | S6 | m6 |  | S5 | on test data build m6 | on test data build m6 | on test data build m6 |
| 100 Rec | S7 | m7 |  | S4 | on test data build m7 | on test data build m7 | on test data build m7 |
| 100 Rec | S8 | m8 |  | S3 | on test data build m8 | on test data build m8 | on test data build m8 |
| 100 Rec | S9 | m9 |  | S2 | on test data build m9 | on test data build m9 | on test data build m9 |
| 100 Rec | S10 | m10 | S2-S10 | S1 | on test data build m10 | on test data build m10 | on test data build m10 |
|  |  |  |  |  | MSE1 = Calculate AVG of test MSE when alpha=2 | MSE2 = Calculate AVG of test MSE when alpha=3 | Calculate AVG of test MSE when alpha=100th value |
|  |  |  |  |  | - should be less than or equal to tolerance value, One we get this we can stop the iteration |  |  |

## Elastic Regularization

It is combination of Ridge and Lasso. We generally never use this. While building the model should not have any insignificant variables as well as it's not overfitting issues.

# Multiplication Rule in Probability

P(A and B) = P(event A occurs in the 1st trial and event occurs in the 2nd trial)

Occurrence of B purely depends on Occurrence of A.

P(TandC)

Machine generated alternative text:



P(both correct) = P(T and C) = 1/2 \* 1/5 = 1/10

P(B|A) represents the Probability of event B occurring after it is assumed that event A has already occurred (read B|A as B given A)

**Multiplication rule formula:**

* **Independent events**
  + If 2 events are independent then P( A and B) = P(A) \* P(B).
  + Sample with replacement
  + Total count does not change
* **Dependent events**
  + If 2 events are dependent then P( A and B) = P(A) \* P(B|A).
  + Sample without replacement
  + Total count changes, it will decrease

**Small sample from large population:**

If sample size is not more than 5% of the of the population, treat the selection as being independent (even if the selections are made w/o replacement, so they are technically dependent).

Machine generated alternative text:
Train Data 
Business Objective 
Data gathering 
Master Data 
Size > 1 M 
Yes 
No 
Sampling size > 5% of 
Further process 
Test Data 

If sample size is less than 5% of population then all the variables are independent variables, there is no dependent variable hence we can’t apply ML. If it is less than 5% then we apply special type of ML called clustering algorithm.

Classification Algorithm (90% of the problems are based on classification algorithm)

Machine generated alternative text:
Feeding Information to 
Machine 
Machine Learning 
Process the data with 
the help of algorithm 
Outcome 

# Classification of Machine Learning

## Supervised Learning -

* + Objective is predefined
    - **When Objective in Numerical format**

### Regression Algorithm (Numerical data, 3 Types)

* + - * 1. Simple Linear Regression
        2. Multi Linear Regression
        3. Polynomial Regression
        4. Optimization techniques of Regression Algorithms

Gradient Descent

VGD (Vanish gradient descent)

SGD (stochastic gradient descent) (batchwise or step by step),

Stepwise regression

* + - * 1. Regularization techniques of Regression Algorithms

Ridge

Lasso

ElasticNet

* + - **When Objective in Categorical format**

### Classification Algorithm (Categorical data, 10 Types)

* + - * 1. Naïve Bayes
        2. Logistics Regression
        3. Decision Tree
        4. KNN (Kth Nearest Neighbor)
        5. Ensembling techniques

Bagging

Random Forest

Boosting

Ada Boosting

Gradient Boosting

XG Boosting

* + - * 1. Neural Network

Shallow

ANN (Artificial Neural N/W)

DNN (Deep Neural N/W)

* + - * 1. SVM (Support Vector Machine)
        2. CNN (Convolution Neural N/W), Image processing
        3. RNN (Recurring Neural N/w), text data processing
        4. NLP (Natural Language Processing, Neural Network)

## Unsupervised Learning

## Reinforcement Learning

# Naïve Bayes

**Properties:**

* It is a classification algorithm (supervise learning method)
* It is a benchmark model or base model. Mostly used in POC. It's kind of dirty model
* POC or POV Proof of Concept or Proof of Value.
* Dependent variable should be a binary variable
* Data might consist of numerical as well as categorical variable
* **For numerical variables (discrete or continuous variable)**
  + Data must be normally distributed
  + If data is not normally distributed, we need to apply variable transformation techniques. 4 diff transformation techniques
    - Log
    - Square root
    - Inverse
    - Square
  + Data should not have missing value and outliers
  + If data is normally distributed, 99% chances are that variables are linear in nature. Increasing or decreasing trend
* **For categorical variables**
  + Never consider PII data, e.g name, email, phone number, SSN, DOB
  + One of the drawback of ML is it does not accept the categorical variables. So we need to convert categorical variables into numerical format

## Dummy Coding/One hot encoding

* + - * Converting the categorical data into binary format (1 or 0)
      * N columns: If we have 3 values for a categorical variable, then it will create 3 new columns and remove original column.
        + e.g. Gender will be removed, instead G-m, G-f,G-O will be created

|  |  |  |  |
| --- | --- | --- | --- |
| Gender | G-m | G-f | G-o |
| Male | 1 | 0 | 0 |
| Female | 0 | 1 | 0 |
| Others | 0 | 0 | 1 |

* + - * N-1 columns: If we have 3 values for a categorical variable, then it will create 2 new columns and remove original column.
        + e.g. Gender will be removed, instead G-m, G-f will be created

|  |  |  |
| --- | --- | --- |
| Gender | G-m | G-f |
| Male | 1 | 0 |
| Female | 0 | 1 |
| Others | 0 | 0 |

* + - * + **Can be done on all the categorical variables at a time**

Machine generated alternative text:
In 
outC11]: 
In [16]: 
In [15]: 
outC15] : 
bankl[ ' marital ' ] . value_counts() 
married 
single 
di vorced 
27214 
12798 
5287 
Name: marital, dtype: int64 
bankl[ 'education ' ] . value counts() 
secondary 
tertiary 
primary 
unknown 
23282 
13381 
6851 
1857 
Name: education, dtype: 
# bankl.nunique 
i nt64 
create N variable (Dummy Coding/One hot encoding) 
# create N variable (Dummy Coding/One hot encoding) 
bank2=pd . get dummies(bankl) 
print (bank2. shape 
bank2 . head() 
(45211, 7) 
marital divorced 
marital married 
marital_single 
education_primary 
education_secondary 
education_tertiary 
education unknown Machine generated alternative text:
create N-l variable (Dummy Coding/One hot encoding) 
# create N-l variable (Dummy Coding/One hot encoding) 
outC18] : 
bank3 = 
print (bank3. shape) 
bank3 . head() 
(45211, S) 
marital_married marital_single education_secondary 
education_tertiary 
education unknown 

## Label Encoding

**Done on one variable at a time.**

Machine generated alternative text:
from sklearn . preprocessing import Label Encoder 
1b = LabelEncoder() 
# = pd. 
bankl[ "mari tal_flag"] 
1b. "marital " ] ) 

Machine generated alternative text:
pd. crosstab(bankl[ "marital " , bankl[" marital_flag" ] ) 
outC21]: 
marital_flag 
marital 
divorced 
married 
single 
5207 
27214 
12790 

|  |
| --- |
| * Formula w.r.t. one Independent variable   + P(c|x) =   P(c|x) = Posterior Probability  P(x|c) = Likelihood Probability  P ( c ) = Class prior Probability  P(x) = Predictor Prior Probability  The Naïve Bayes applies for sample w/o replacement |

What is the probability of not playing when weather is sunny

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Weather | No | Yes | Total | Probability |
| Overcast | 0 | 4 | 4 | 1 |
| Rainy | 3 | 2 | 5 | 0.4 |
| Sunny | 2 | 3 | 5 | 0.6 |
| Grand Total | 5 | 9 | 14 | 0.24 |

P(y|x) = P(y) P(x|y)/P(x)

P(No|sunny) = P(No) P(sunny|No)/P(sunny)

P(No) = 5/14

P(sunny) = 5/14

P(sunny|No) = 2/5

P(No|sunny) = P(No) P(sunny|No)/P(sunny) = 0.4

Play when raining

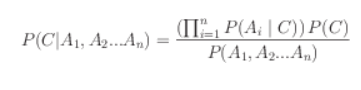
P(Yes|Rainy) = P(Yes) P(Rainy|Yes)/P(Rainy) = (9/14)\*(2/9)/(5/14) = 2/5 = 0.4

P(Yes) = 9/14

P(Rainy) = 5/14

P(Rainy|Yes) = 2/9

* Naïve Bayes works very well with categorical variable rather than numeric variable
* Convert continuous variable into categorical format
* The Naïve Bayes applies for sample w/o replacement
* What if we have more than one independent variables?
  + Say Age convert into bins
    - Grp1= 0-25%
    - Grp2 = 25%-50%
    - Grp3 = 50%-75%
    - Grp4 = 75%-100%

* Formula w.r.t. two Independent variable
* 

|  |
| --- |
| from sklearn.naive\_bayes import GaussianNB  gb = GaussianNB()  gb.fit(x\_train,y\_train)  # var\_smoothing: Variance smoothing, how much variance in the given dataset  # prior probablities |

GaussianNB(priors=None, var\_smoothing=1e-09)

Portion of the largest variance of all features that is added to variances for calculation stability

To make variables more stable and one variable to not influence the other variable, we are adding variance to each and every variable. Variance smoothing and epsilon are the same thing.

* All classification algorithms by default will give the predicted probabilities values with an exception of SVM.
* By default each model will give us 2 probabilities values. Probabilities of 0 or events and probability of 1's or non-events.
* By default the probability value = 0,.5
* Anything which is < 0.5 is considered as 0
* And > 0.5 is considered as 1

## Confusion Matrix or classification Matrix

TP : True Positive : Actual and predicted both values are positive. **Correctly predicted as positive.**

TN: True Negative: : Actual and predicted both values are negative. **Correctly predicted as negative.**

FP: False Positive: Actual value is negative and predicted value is positive. **Wrongly predicted as positive.**

FN: False Negative: : Actual value is positive and predicted value is negative. **Wrongly predicted as negative.**

**0 is considered as negative**

**1 is considered as positive**

|  |  |  |
| --- | --- | --- |
|  | Predicted |  |
| Actual | 0 | 1 |
| 0 | TN | FP |
| 1 | FN | TP |

## Model Metrices

|  |  |
| --- | --- |
| **Model Metrices** | **Description** |
| **Accuracy =** | How many positives and negatives are predicted correctly |
| **Error or misclassification =** | How many positives and negatives are predicted Incorrectly |
| **Precision =** | How many predicted positives have been classified as positives |
| **Recall (or sensitivity) =** | How many actual positives have been classified as positives |
| **Specificity=** | How many actual negatives have been classified as negatives |
| **F1 Score =** | Harmonic mean of precision and recall is F1 Score Avg balanced accuracy of the model |
| ROC Curve (best range values 0-7-0.8) | Receiver Operating Characteristics  While making the TP how many mistakes we have identified.  default probability of predicted value is 0.5  The max diff b/w line and the curve is called is ROC point  False Positive = 1-Specificity = 1-TN/(FP+TN)  False Positive = FP/(FP+TN)    The ideal ROC value should be b/w 0.7 to 0.8  ROC is a curve, the diagonal line makes 45%  X axis is FP and Y-axis is TP    AUC is 100% (area under the curve)  The maximum value b/w the line and curve is ROC.  Machine generated alternative text: ROC  True Positive  45 degree  FP: False Positive |
| KS - Stats (best range values 25% -60%) |  |
| Gain Chart/lift curve |  |
| Rank Ordering |  |

We prefer Recall over Precision. Which means we always consider Recall as compared to Precision

## ROC

Receiver Operating Characteristics

While making the TP how many mistakes we have identified.

default probability of predicted value is 0.5

The max diff b/w line and the curve is called is ROC point

False Positive = 1-Specificity = 1-TN/(FP+TN)

False Positive = FP/(FP+TN)

The ideal ROC value should be b/w 0.7 to 0.8

ROC is a curve, the diagonal line makes 45%

X axis is FP and Y-axis is TP

AUC is 100% (area under the curve)

The maximum value b/w the line and curve is ROC.

Machine generated alternative text:
ROC 
True Positive 
45 degree 
FP: False Positive 

Churn Analysis

* Identify Reasons
* Predict
* Recommendation

Machine generated alternative text:
Business Objective 
Data gathering 
Master Data 
Clean Data 
EDA: Reasons 
Test 
Test Data 
Variable Reduction 
Model 
Measure Model Metrics 

# Clustering Algorithms (K-Means)

**Homogeneous data**: all the data is having similar nature

**Heterogeneous data**: all the data is having different nature

Sometimes though we have dependent variables (we know what we are drying to build),

Identify who is going to become delinquent customer (not paying bills for consecutive 3 months when bill is greater than 0).

Machine generated alternative text:
Train Data 
Business Objective 
Data Gathering 
Master Data 
Size > 1 M 
Stratified Sampling 
Test Data 
Remove duplicates 
Remove duplicates 
Clean Data 
(Missing, Outlier, Check Normal distribution, EDA(Univariate, 
ai-variate, Multi Variate analysis): Hypothesis testing) 
Clustering 
Clusterl 
Cluster2 
Train (80%) 
Variable Reduction technique 
Model Building 
Measure Model metrics 
Cluster3 
Test (20%) 
Measure Model metrics 
Recommendation 

Clustering is also called segmentation

If the data is categorical then clustering is easy but if data is numeric then actual clustering algorithm will work.

Machine generated alternative text:
Un-employed 3000 
Low salary 5000 
Low designation 1500 
10000 customers 
Employed-7000 
High salary 2000 
High designation 500 

## Euclidian distance

In order to calculate the distance b/w the two data points

Machine generated alternative text:



## Manhattan distance

Manhattan distance =

1. Non-Hierarchical clustering algorithm: K Means algorithm (mostly this is used)
2. Hierarchical clustering algorithm: Agglomerative algorithm

K-Mode(in NLP)

## K-Means

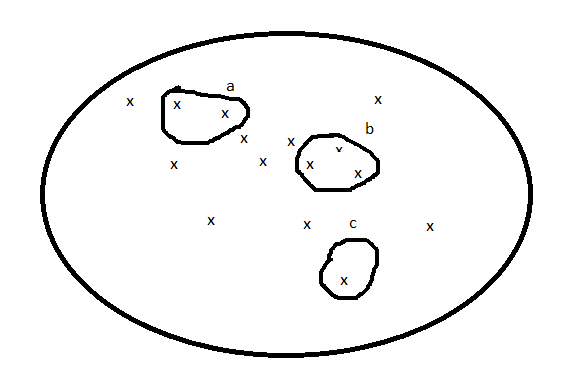
K is number of clusters. Means me mean value

Say K = 3

Select randomly one data point

## Minimum or maximum linkage method

whichever distance is min or max we assign that cluster to that row



Machine generated alternative text:
applied manhattan distance 
ew Centroids after Step 
Cluster Schema 
Clusterl Cluster2 Cluster3 
Y 
10 
1 
3 
2 
8 
Clusterl 
Distmean 
(2,10) 
12 
10 
10 
Stepl 
Cluster2 
Distmean 
(5,8) 
10 
Cluster3 
Distmean 
(1,2) Cluster 
10 
10 
(2,10) 
2,10) 
(8,4) 
(5,8) 
(7,5) 
(6,4) 
(6,6) 
(2,5) 
(1,2) 
(1.5,3. 

## Cluster Schema

Machine generated alternative text:
applied manhattan distance 
ew Centroids after Step 
Cluster Schema 
Clusterl Cluster2 Cluster3 
1 
2 
3 
4 
5 
6 
7 
8 
x 
2 
2 
8 
5 
7 
6 
1 
4 
Y 
10 
5 
4 
8 
5 
4 
2 
Clusterl 
Distmean 
(2,10) 
o 
5 
12 
5 
10 
10 
g 
3 
Step2 
Cluster2 
Distmean 
(6,6) 
8 
5 
4 
3 
2 
2 
g 
5 
Cluster3 
Distmean 
(1.5,3.5) 
7 
2 
7 
8 
7 
5 
2 
8 
Cluster 
1 
3 
2 
2 
2 
2 
3 
1 
(2,10) 
(4,9) 
,9.5) 
(8,4) 
(5,8) 
(7,5) 
(6,4) 
(6.5,5.25) 
6.5 
5.25 
(2,5) 
(1,2) 
(1.5,3.5 
If 2 values are same then take the 1st cluster 

Machine generated alternative text:
applied manhattan distance 
ew Centroids after Step 
1 
2 
3 
4 
5 
8.5 
6 
8.5 
7 
9.5 
8 
1.5 
x 
2 
2 
8 
5 
7 
6 
1 
4 
Y 
10 
5 
4 
8 
5 
4 
2 
Clusterl 
Distmean 
(3,9.5) 
1. 
5. 
10.5 
3. 
Step3 
Cluster2 
Distmean 
(6.5,5.25) 
9.25 
4.75 
2.75 
4.25 
0.75 
1.75 
8.75 
6.25 
Cluster3 
Distmean 
(1.5,3.5) 
7 
2 
7 
8 
7 
5 
2 
8 
Cluster Schema 
Cluster2 Cluster3 
5 
5 
5 
Cluster 
1 
3 
2 
1 
2 
2 
3 
1 
Clusterl 
(2,10) 
(5,8) 
(4,9) 
.6,9) 
3.666667 
9 
(8,4) 
(5,8) 
(7,5) 
(6,4) 
(7,4.3) 
7 
4.333333 
(2,5) 
(1,2) 
(1.5,3.5 

Machine generated alternative text:
applied manhattan distance 
New Centroids after Step4 
Clusterl 
Distmean 
(3.6,9) 
Step4 
Cluster2 
Distmean 
(7,4.3) 
10.7 
1 
2 
5.7 
3 
1.3 
4 
5.7 
5 
6 
1.3 
7 
8.3 
8 
7.7 
x 
2 
2 
8 
5 
7 
6 
1 
4 
Y 
10 
5 
4 
8 
5 
4 
2 
Cluster3 
Distmean 
(1.5,3.5) 
7 
2 
7 
8 
7 
5 
2 
8 
Cluster Schema 
Cluster2 Cluster3 
2.6 
5.6 
9.4 
2.4 
7.4 
7.4 
9.6 
0.4 
Cluster 
1 
3 
2 
1 
2 
2 
3 
1 
Clusterl 
(2,10) 
(5,8) 
(4,9) 
3.6,9) 
3.666667 
9 
(8,4) 
(5,8) 
(7,5) 
(6,4) 
7 
4.333333 
(2,5) 
(1,2) 
1.5 3.5 

Now since centroid is not changing we can stop after step4

Repeat steps until we get same centroid values in subsequent step, then we stop the step and that centroid is the final centroid

Datapoints

10

20

30

40

50

10-10/(50-10) = 0

20-10/40 = 0.25

30-10/40 = 0.5

40-10/40 = 0.75

50-10/40 = 1

Machine generated alternative text:
Business Objective 
Data Gathering 
Master Data : Very large number of variables 
Data Processing 
Missing and outliers 
Random Forest 
Important varables Data 
Pairwise correlation 
Model Building 

X1,x2,x3,x4,x5,y

(x1,y): Naïve Bayes --> ROC1

(x2,y): Naïve Bayes --> ROC2

(xn,y): Naïve Bayes --> ROCn

Discard any ROC less than 0.7 then discard those variables.

Range of ROC is (0.7-0.8)

Machine generated alternative text:
Business Objective 
Data Gathering 
Master Data : Very large number of variables 
Data Processing 
Missing and outliers 
Naive Bayes calculate ROC for each variable 
Important variables Data with ROC < 0.7 
Pairwise correlation 
Model Building 

For fast calculation, we convert the data into min-max scalar (0,1)

Machine generated alternative text:
Business Objective 
Data Gathering 
Master Data : Very large number of variables 
Data Processing 
Missing and outliers 
Converting O to 1 (minmax scalar: Unified data) 
1. 
2. 
3. 
4. 
5. 
Clustering 
Logistics Regression 
ANN 
KNN 
Naive Bayes 
SVM 
Variable Reduction: On Imp Variables do the EDA 
Random Forest 
correlation 
AUC 
ROC correlation 

Machine generated alternative text:
Test (20%) 
Train (80%) 
Variable Reduction technique 
Measure Model metrics 
Model Building 
Recommendation 
Measure Model metrics 

# Decision Tree

It is supervised learning algorithm. Decision tree divides the heterogeneous data into homogeneous data based on dependent variable.

1. What is segmentation
2. What is decision tree
3. Best splitting attribute
4. Building decision tree
5. Tree validation
6. Pruning
7. Prediction using the model

Clustering is unsupervised algorithm; it does not consider the label (dependent variable). It converts the heterogeneous data into homogeneous data based on independent variables.

**Supervised learning:**

* In Supervised learning when dependent variable is numerical then those algorithms are called Regression algorithms.
* Possibility to divide data into train and test
* In case of categorical dependent variables, the algorithm is classification algorithm

**Un-Supervised learning:**

* Not possible to divide into train and test
* No dependent variable

Decision tree is supervised learning algorithm. It divides the heterogeneous data into homogeneous data based on dependent variable. It is a treebased algorithm.

Example: customer's call data. Voice call (inbound, outbound), data-usage, sms, mms, churn. Predict churn and not churn customers.

Machine generated alternative text:
Parent or 
Child 
Node 
Leaf 
Node 
Leaf 
Node 
Root Node or 
depended variable 
Parent or 
Child 
Node 
Leaf 
Node 
Parent or 
Child 
Node 
Leaf 
Node 

Machine generated alternative text:
Gender 
Old Data 
Marital Status Ordered the product 
Gender 
Married 
Unmarried 
Married 
Married 
Married 
Married 
Unmarried 
Unmarried 
Married 
Married 
Married 
Unmarried 
Married 
Unmarried 
No 
Yes 
No 
No 
No 
No 
Yes 
Yes 
No 
No 
No 
No 
No 
Yes 
New Data 
Marital Status Ordered the product 
Married 
Unmarried 

Machine generated alternative text:
6 - Female 
3- No, 3 - Yes 
No-50%, Yes-50% 
3-Married 
6-Married 
6- No, O- Yes 
No-IOO%, Yes-O% 
14-Total 
10- No, 4-Yes 
No-71%, Yes-29% 
8-Male 
7 - No, 1 -Yes 
No-86%, Yes-14% 
2-Un-Married 
1 
No, 1 — Yes 
No-50%, Yes-50% 
3 
— Yes 
No-IOO%, Yes-O% 
3-Un-Married 
O No, 3-yes 
No-O%, Yes-IOO% 

* 1st node is **root node**. It always is the dependent variable
* Middle nodes are called **parent or child nodes**
* The last node which is not splitting further is called **leaf node**
* We can make the decision, based on leaf node. It should be **perfect node** or **strong node** or **Pure Node**. It must have to give clear cut information be +ve or -ve
* The node which does not give clear info. It is not pure node i.e does not give 100% positive or 100% negative info then we split it further. It is **confusion node** or **impure node.**

Which variable should be consider 1st for splitting the data for decision? How do we consider the most Important variable?

## Rules to choose the best split variable

### Gini

It works at node level

**Gini** = Entropy at the RootNode - Sum of weighted entropy at each child node

Gini = 1-

P = probability of success or failure (so I will be for success and failure)

**Collection of multiple decision tree is random forest**

### Information Gain (Entropy)

Maximum amount of information is considered as most significant variable for splitting.

* Each node should be pure(node can’t be divided further) or impure node (node can be divided further)
* It measures the uncertainty of a node. It is denoted with a s

S = (-p \* ) + (-q \* )

P = probability of success

q = probability of failure

**Entropy value ranges from 0 to 1**

If Entropy is ~ 0 then pure node,

if Entropy is ~1 then impure node

**Pure node Example**: Out of 100 customers, 100 buying, 0 not buying

P = 100/100 = 1

Q = 1/100 = 1

S = (-p \* ) + (-q \* ) =

= (-1 \* ) + (-0 \* ) = -1\* 0 + 0 = 0

**Impure node Example:** Out of 100 customers, 50 buying, 50 not buying

P = 50/100 = 0.5

Q = 50/100 = 0.5

S = (-p \* ) + (-q \* ) =

= (-0.5 \* ) + (-0.5 \* ) = -0.5\* (-1) + -0.5\* (-1) = 1

чан 
ноовпО 
= родпо 
= РИМ 
РВПО 

**Players will play = if (Outlook=Overcast) or ((Outlook=Rain) and Wind=Weak)) or ((Outlook=Sunny) and Wind=Normal))**

Decision Tree never gives probability value, it always give yes or no values

**Height of the tree is called depth of tree**

**Overfit**: If depth of the tree is high

**100% of the chances decision tree models are overfit.**

While building decision tree or random forest or clustering (KNN or Agglomerative), your categorical variable cannot be dummy coding it has to be **label encoding.**

## Pruning

The process of controlling the overfitting is called Pruning. If difference b/w train and test accuracy is more than 10% then it is called overfitting. Overfit means model is performing well on training dataset but not on the test dataset. By default most of the decision tree model are prone to overfitting.

### Methods of Pruning

Decision tree are overfit, how do I prune it or how do I reduce the overfitting

* max\_depth
* max\_features
* min\_impurity\_decrease (Max impurity to decrease (entropy to decrease))

## Hyper Parameter Tuning

To overcome overfitting, we use hyper parameter tuning.

**Hyper Parameter tuning** : Manually

**Hyper Parameter tuning :** By program

1. Grid Search
2. General Search
3. Random search

## Limitations of Decision tree

If dependent variable consists of more than 2 labels then we can't apply decision tree (as it can't calculate Entropy or Gini as they are based on probability of success or failure)

**Decision tree has KNN working internally**

# KNN

Kth nearest neighbor. KNN is a supervised learning. (as opposed to K-Mean, which is an unsupervised learning.

K-means To identify patterns of data (by default k = 8) or cluster/group of data we use K-Means. It is non-hierarchical algorithm. By using algorithm curve, we determine the optimal number of K.

K is nothing but groups so we need to consider random centroid to start with determining optimal centroid.

## Difference b/w K-Mean and KNN

|  |  |
| --- | --- |
| **K-Means** | **KNN** |
| Unsupervised Learning | Supervised Learning |
| We don't have dependent variable | We have the dependent variable |
| It has possibility of having Train and test data sets | It does not have the possibility of Train and test data sets |

Supervised Learning

* Classification
  + Binary
  + Multinomial
* Regression

Classification Algorithms

* Naïve Bayes (binary dependent variable, range of probability value is 0 to 1)
  + Range of probability value (0 to 1)
* Decision tree (0 or 1)
  + Binary Value (0 or 1)

KNN can be used for binomial as well as multinomial

## K-Means (Recap)

Machine generated alternative text:
applied manhattan distance 
New Centroids after Stepl 
Cluster Schema 
Clusterl Cluster2 Cluster3 
Y 
10 
1 
3 
6 
2 
8 
6 
Clusterl 
Distmean 
(2,10) 
12 
10 
10 
Stepl 
Cluster2 
Distmean 
(5,8) 
10 
Cluster3 
Distmean 
(1,2) Cluster 
10 
10 
(2,10) 
2,10 
(8,4) 
(5,8) 
(7,5) 
(6,4) 
6,6 
(2,5) 
(1,2) 
(1.5 3.5 

Iterate until you get non-changing centroids. Once we get fixed centroid values (3 centroid is K=3, 8 centroid if K=8)

Then find the distance of each point in a particular cluster from the centroid of corresponding cluster. Then rank the distance of each point from the centroid of corresponding cluster. Find K- nearest points based on Rank and then determine the dependent variable based on more numbers of dependent variable.

## KNN = K-Mean + Predictions

KNN to find the nearest K neighbors. --> **K-Mean + Predictions**

Always take odd K and not even number. By default, K value is 5.

Machine generated alternative text:
10 
5 YES 
10 NO 
5 YES 
7 YES 
5 NO 

Machine generated alternative text:
Length 
5.3 
5.1 
7.2 
5.4 
5.1 
5.4 
7.4 
6.1 
7.3 
6 
5.8 
6.3 
5.1 
6.3 
5.5 
Width 
3 
.7 
3 
.8 
3 
3 
.4 
3. 
3 
3 
.9 
2.8 
2.8 
2.9 
2.7 
2.8 
2.3 
2.5 
2.5 
2.4 
Species 
setosa 
setosa 
virgimca 
setosa 
setosa 
setosa 
virgimca 
versicolor 
virgimca 
versicolor 
virgimca 
versicolor 
versicolor 
versicolor 
versicolor 
2.5 
Length-Width 
Length 

**Predict the dependent variable (species) for flower of length=7 and width=3**

Machine generated alternative text:
sl No 
Length 
Width 
Manhattan distance from (7,3) Euclidian Distance Euclidian Rank Manhattan Rank Top 3 Nearest Species 
Top 3 Rank belongs to Virginica 
5.3 
5.1 
2.7 
2.7 
7.2 
0.2 
0.2 
5.4 
5.1 
2.2 
2.2 
5.4 
2.5 
2.5 
0.6 
0.2 
0.6 
Predict the 
6.1 
1.1 
0.2 
1.1 
dependent 
7.3 
0.1 
variable 
0.3 
5.8 
0.2 
6.3 
5.1 
0.5 
6.3 
0.5 
5.5 
0.6 
1 
2 
3 
4 
5 
6 
7 
8 
10 
11 
12 
13 
14 
15 
3 
.7 
3 
.8 
3 
3 
.4 
3. 
3 
3 
.9 
2.8 
2.8 
2.9 
2.7 
2.8 
2.3 
2.5 
2.5 
2.4 
Species 
setosa 
setosa 
virgimca 
setosa 
setosa 
setosa 
virgimca 
versicolor 
virgimca 
versicolor 
virgimca 
versicolor 
versicolor 
versicolor 
versicolor 
2.4 
2 
0.4 
1.3 
1.4 
1.4 
2.4 
1.2 
2.1 
0.7 
0.8 
0.4 
0.3 
0.9 
2.4 
2 
0.4 
1.3 
1.4 
1.4 
2.4 
1.2 
2.1 
1.838477631 
2.061552813 
0.2 
1.64924225 
1.923538406 
1.835755975 
0.447213595 
0.921954446 
0.316227766 
1.044030651 
1.216552506 
0.989949494 
1.96468827 
0.860232527 
1.615549442 
12 
1 
10 
13 
11 
3 
2 
7 
12 
1 virginica 
11 
14 
3 virginica 
2 virginica 
12 
10 
1 virginica 
2 virginica 
3 virginica 
7.4 
6 

Euclidian Distance = SQRT((C23-$F$22)^2 + (D23-$G$22)^2)

Manhattan Distance = ABS($F$22-C23) + ABS($G$22-D23)

Rank = RANK(H23,$H$23:$H$37,1) = RANK(point, range,1=asc,0=desc)

Manhattan distance =

Decision tree has KNN working internally

If dependent variable consists of more than 2 labels then we can't apply decision tree (as it can't calculate Entropy or Gini as they are based on probability of success or failure)

## Brut Method

KNN works based on brut method (clustering, forceful method)

* Binomial Dependent variable
  + Decision tree (Information gain (entropy) or Gini we can build decision tree model)
* Multinomial Dependent variable

### Brute force

* + - KNN (with Brute force: K-Mean clustering method) 99.99% this is used.

### (Kd-Tree or ball-tree).

* + - KNN (with Kd-Tree or ball-tree). Decision tree will behave as KNN when we have more than 2 labels in dependent variable. Not used much as it is memory intensive
    - Kd-Tree(based on Entropy) or
    - ball-tree (based on Gini).

Machine generated alternative text:
Decision Treel Decision Tree2 Decision Tree3 
Good 
Bad 
Average 

# Ensembling

Combining the multiple models into a single model is ensembling.

The reasons to combine multiple models is

* To get the better performance or get the better results of the model

All the models ensembled are either binary or multinomial models. Can’t be combination.

Some data is captured in one , some in other hence average of all will be the best model

By measuring standalone accuracy, we can't measure the accuracy of final model

Machine generated alternative text:
Data 
KNN 
N aive 
Bayes 
Decision 
Tree 

## Injective or cognitive method

Use predicted probability values from each model (DT-pred, KNN-Pred, NB-Pred ) to build one more iteration of model building KNN, DT and NB.

Build model (KNN,DT,NB using NB-Pred, Probability Value of DT, Probability Value of KNN and y)

Machine generated alternative text:
xl 
DT-pred 
KNN-Pred NB-Pred 
0.6 
0.5 
0.5 
0.2 
Probability Value of DT 
0.5 
0.6 
0.5 
0.7 
Injective or cognitive method 
Probability Value of KNN 
0.8 
0.5 Build model using NB- 
Pred, Probability Value of DT, 
0.6 
Probability Value of KNN and y) 
0.5 

## Average Probability Method

Machine generated alternative text:
xl 
DT-pred 
KNN-Pred NB-Pred 
0.6 
0.5 
0.5 
0.2 
Probability Value of DT 
0.5 
0.6 
0.5 
0.7 
Average Method 
Probability Value of KNN 
Calculate average of probability 
0.8 
0.5 methods of DT,KNN, NB (NB-Pred+ 
Probability Value of DT+ Probability 
0.6 
0.5 Value of KNN)/3 

Calculate average of probability methods of DT,KNN, NB (NB-Pred+ Probability Value of DT+ Probability Value of KNN)/3

## Voting Method

Use predicted values (0,1) from each model (DT-pred, KNN-Pred, NB-Pred ) to determine the predicted value.

Build model (KNN,DT,NB using NB-Pred, DT\_pred, KNN\_pred and y)

Machine generated alternative text:
xl 
DT-pred 
KNN-Pred NB-Pred 
0.6 
0.5 
0.5 
0.2 
Probability Value of DT 
0.5 
0.6 
0.5 
0.7 
Voting 
Probability Value of KNN 
0.8 
0.5 
0.6 Build model using NB- 
0.5 Pred, DT_pred, KNN_pred and y) 

## Bagging Method

Aggregation of boot straples is called Bagging.

Boot strap (Straples) : Sample with replacement one record at a time.

Population = 10000

10 samples with 10000 size each. **Sample size = population size**

33% is duplicate, 67% is unique. By default 10 samples are created. Each sample can have duplicates but atleast one record will be different from other sample. All the model can have duplicates but atleast one record will definitely different.

Machine generated alternative text:
Samplel 
Modell 
Same 
model 
(10K, 20 cols, 10 samples) 
Sample2 
Mode12 
Same 
model 
Avg of all 
models in 
bagging 
model 
Sample10 
Modeli0 
model 

By default bagging model uses **decision tree model** for model building of different sets it creates.

High, medium and low predicting power of samples.

Bagging will also overcome the overfitting issue b/c we take avg of all the models. It reduces variance of model. Because some sample will have less predicting power, some have medium and some have strong predicting power.

### oob\_score

 (out of bag score)

if you run bg.fit multiple time, accuracy is not going to be the same

in order to identify the optimal accuracy of the model, we are going to use the #**cross validation techniques**, which is nothing but oob\_score: out of bag score.

In oob\_score calculation, #we are going to create sample w/o replacement which means sample have unique records (no duplicates)

General tendency of bagging model is over fit b/c oob\_score and accuracy score difference is more than 10% (overfit model)

Avg of accuracy of all the models = oob\_score

* # BaggingClassifier(
* # base\_estimator=None, by default internally it will biuld decison tree model
* # bootstrap=True, sample with replacement at row level
* # bootstrap\_features=False, column level sampling we are not doing. if bootstrap\_features = True (random forest)
* # max\_features=1.0, atleast each sample one rec at a time
* # max\_samples=1.0, atleast each sample splits should be there
* # n\_estimators=10, by default 10 smaples are there
* # n\_jobs=None, one job at a time
* # oob\_score=False, out of bag score, optimal accuracy of the model, cross validation
* # random\_state=None,
* # verbose=0, buy default does not printbthe o/p, verbose=-1 (print the o/p)
* # warm\_start=False): False: does not support parallel executuon, True: Execute parallel

# change from the default Decision tree to KNN. When model is overfit we use **hyper parameter tuning (grid search and random search)**

### Drawbacks of bagging

* Overfitting as some of the variables are highly influencing which will influence each model and hence will cause overfitting

### Random forest

* It is specific case of Bagging algorithm
* Collection of decision tree model on random sample data is called Random forest model
* Random sample data will have different collection of columns with two sample having atleast one different column

#### Feature sampling Methods (max\_features)

* + Sqrt ( when num of variable > 20 less than 100)
  + Log ( when num of variable < 20)
  + (Number of col)/3 --> Not available in python

#### Feature importance Methods

* + Gini or Entropy: Variable importance will be calculated : For each var it will calculate Gini or Entropy(information gain). Var1..Var2..Var10 should have atleast one different variable.

E.g. 100 columns with 1 dependent variable, 99 dependent variable. Sqrt(99) ~ 10. Hence each variable set will have {10 (independent var) +1(dependent var)} = 11 variables.

Machine generated alternative text:
Samplel 
10K, 100 var 
varl (11 var) 
Two var sets with 
one unique 
Data 
(10K, 100 cols, 10 samples) 
Sample2 
10K, 100 var 
Var2(11 var) 
Two var sets 
with 
one unique var 
Modell 
Mode12 
Same 
Same 
model 
model 
Median 
or MOD 
of all 
models in 
Sample10 
10K, 100 var 
varlO(11 var) 
Two var sets 
with A!lg@étone 
unique var 
Mode110 
Same 
model 

##### Practical approach (Variable reduction technique (dimensionality reduction)

( Feature importance in Classification model (practical approach) this is also called Variable reduction technique (dimensionality reduction)

* Rf feature\_importance\_ or AUC score or ExtraTree feature\_importance\_
* Pairwise correlation
* VIF
* P-Value
* Finalize the variable

**Feature importance in Classification model (standard approach).**

**This is also called Variable reduction technique (dimensionality reduction)**

* Information value (gini, entropy)
* Pairwise correlation
* VIF
* P-Value
* Chi-Square
* RF-variable
* Finalize the variable

**Feature importance in Regression model (standard approach)**

* Pairwise correlation
* Stepwise regression
* P-Value
* VIF
* PCA
* Lasso
* F-stat/ATC
* Finalize the variable

## Boosting Method

* This method is to improve the accuracy or performance of model. **This is used rarely in practical scenario.** B/c in this data is manipulated manually which is not possible in real world scenario where we want to automate the process.
* By default under the hood decision tree runs
* **All the algorithms work (Random forest, Decision Tree, Regression, Naïve Bayes, KNN, K-Mean) at column level. Only Boosting algo works at row level**
* **Boosting is 3 types**
  + ADA Boosting
  + Gradient Boosting (it has correlation with logistic regression hence once logistic regression is covered then this can easily be understood)
  + XGBoost --> Combination of ADA boosting and Gradient Boosting

# Logistic Regression (Most trusted Algo)

Heteroscedasticity

To overcome the issues with LinearRegression (or limitations of LinearRegression) we use logictic regression.

If we have binary dependent variables then

If we have y = 0 and 1 (binary values) 🡪 Then linear regression will be violated

If we have y = 0 and 1 (binary values) 🡪Then linear regression will be violated

If dependent variable is binary, then linear regression does not work as it works on numeric (continuous or discrete ) dependent variable (range -∞ to +∞).

If dependent variable is binary, when we are trying to fit the linear regression models, it violates some of the assumptions of regressions such as

* Dependent variable is not normally distributed
* Variance of error is not constant
* Data is not linear in nature

If dependent variable is binary,

* we measure the probability values, rather than the numerical value.

## Sigmoid Function

If dependent variable is binary, and we try to fit the linear regression model, it does not obey the rules of probability i.e Range of probability.

To overcome that problem, we are going to use the sigmoid function.

Sigmoid function p( y|x) =

Machine generated alternative text:
Buy 
S-Curve 
Sigmoid function p( ylx) = 

Odds Ratio =

P = probability of success

q = probability of failure

Z =

To measure of ratio of probability of success and probability of failure

In order to make data normally distributed, we apply variable transformation technique, say log over here.

log(odds) = log(

If accuracy of the model is very low then the model is called bias model

Odds Ratio =

P = probability of success

q = probability of failure

Z =

To measure of ratio of probability of success and probability of failure

In order to make data normally distributed, we apply variable transformation technique, say log over here.

log(odds) = log(

If p = 0.5

Log(

This is the reason the line is a straight line, as above equation is of straight line

Machine generated alternative text:
Productivity O 
Productivity I 
20 
22 
24 
26 
28 
30 
32 

Antidote of log is exponential

log(odds) is predicted value

1-p =

P = 1 -

P = =

P =

Range of (range -∞ to +∞)

## Activation function of regression

Summation of MSE =

Activation function of regression is OLS = (Ordinary least square equation) = SST = SSR + SSE

## Activation function of classification

Cross entropy (or binary cross entropy) = -y\* log(p) - (1-y) \* log(1-p)

Machine generated alternative text:
Tree Based 
Activation function is 
Entropy and Gini 
Classification 
Linear Equation 
Activation function is 
Sigmoid 
Equation 
Non-Linear Equation, 
Activation function is 
tan-h, Rel U, softmax 

**Formula for logistic regression**

P(y|x) = =

P(y|x)

Here we will convert probability values(P(y|x)) to binary values. If P(y|x) > 0.5 then 1 otherwise 0

**Below is the explanation why we use 0.5 as the threshold or default probability value.**

If

Then

P(y|x) =

If

Then

P(y|x) =

When there is no predicted value to predict then we use default probability value as 0.5

**Understanding**

Logistics regression formula P(y|x) =

SGD (stochastic gradient descent) is to overcome overfitting. It works on sample dataset, whereas gradient descent works of full dataset.

Gradient Descent works on linear data and population. SGD works on non-linear data and sample data

Use below formula then use  **in gradient descent to get new**

Practically

By default

**Iteration0**

P(y|x) =

X = 50,

**Iteration1:**

X = 50,

Accuracy =0.78 (say)

Build the model taking these

**Iteration2:**

Iterate until difference b/w two accuracies are 0.0001. This is called tolerance

SGD (stochastic gradient descent) is to overcome overfitting. It works on sample (without replacement)- dataset, whereas gradient descent works of full dataset.

After we get

Non-linear function (**Exponential is non-linear function**) of linear input is called logistic regression.

Regression Internally it apply regularization techniques (Ridge and Lasso). **Only Logistic Regression in the world which never ever overfits.**

## Event Ratio

Event Ratio or event rate or target ratio =

* **6% < event rate < 20% --> good to build**
* event rate < 0.6% Imbalanced data (fraud detection in Banking or data dealing with wholesale or enterprise data)
* event rate >20% Random Model

**T-test in regression is similar to Event Ratio in classification**

Can you explain one the project you recently worked, Business objective, approach, solution and final recommendation

## Sign test

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Coef.** | sign test (for dependent variable negative in nature) | sign test (for dependent variable positive in nature) |
| **income** | positive | Inversely proportional | directly proportional |
| **months\_on\_network** | positive | Inversely proportional | directly proportional |
| **Num\_complaints** | negative | directly proportional | Inversely proportional |
| **number\_plan\_changes** | negative | directly proportional | Inversely proportional |
| **relocated** | negative | directly proportional | Inversely proportional |
| **monthly\_bill** | negative | directly proportional | Inversely proportional |
| **technical\_issues\_per\_month** | negative | directly proportional | Inversely proportional |
| **Speed\_test\_result** | positive | Inversely proportional | directly proportional |

In [281: 
out [28]: 
import as SI 
model - Sm. 
, data=train) . fit( ) 
model. summary2( ) 
Optimization terninated successfully. 
Current function value: 0.494581 
Iteratfcns 9 
Lwit Pseudo R-squared: 
0.274 
773274024 
77910_8586 
oate-_ 
No. Observations: 
Dt Residuals 
NO 
2019-12-14 11:01 
78662 
Cc*t. 
-17.5921 
0,0016 
00273 
-06567 
-02050 
-3.1205 
-00022 
43795 
022 
AIC: 
BIC: 
Log-Like•nood: 
LL.NWI: 
LLR pan lue: 
Intercept 
montns on networ* 
Num_complaints 
n ges 
relocated 
Std .Err. 
0,3358 
0,0001 
00336 
0086 
.0442 
.0002 
00027 
-52.380 
17,8788 
246133 
-195363 
-239689 
-70.5240 
-12,3882 
-474772 
n 4846 
P>lzl 
o.oocn 
o, 0000 
00000 
00000 
00000 
o. 0000 
a 0000 
00000 
10.025 
-18,2502 
0,0014 
00251 
-07226 
-02217 
-3.2072 
0025 
-0,3951 
02179 
0.9751 
-16,9339 
0,0018 
-05909 
-01882 
-3.0338 
-0,0018 
-o, 3638 
02284 

Based on sign test also we are going to drop the variables. If we feel that data is not correct. It should be directly proportional but outcome is showing its inversely proportional and vice versa

If we are predicting negative nature of dependent variable then relation b/ two variables (dependent vs independent) are inversely proportional

If we are predicting positive nature of dependent variable then relation b/ two variables (dependent vs independent) are directly proportional

**Sign test:**

**If we are predicting negative nature of dependent variable then relation b/ two variable is inversely proportional**

**If we are predicting positive nature of dependent variable then relation b/ two variable is directly proportional**

**if predicting dependent variable which is positive in nature:**

* if Coef is positive then the independent variable is directly proportional to dependent variable.
* Whereas if Coef is negative then the independent variable is inversely proportional to dependent variable

**if predicting dependent variable which is negative in nature:**

* if Coef is positive then the independent variable is inversely proportional to dependent variable.
* Whereas if Coef is negative then the independent variable is directly proportional to dependent variable

**Based on sign test also we are going to drop the variables. If we feel that data is not correct. It should be directly proportional but outcome is showing its inversely proportional and vice versa**

If we have nonlinear data then what will happen if we build logistic regression. It can’t e built. This is **drawback** the of logistic regression. Even if we apply log to non-linear data, if we don’t get non-linear data then we apply **deep learning** algorithms

# accuracy score is 57%, this is underfitting or bias model where accuracy is less

# on top of non-linear data if you build linear models it will always underfit and there will be biasness in the model

Machine generated alternative text:
In [33): 
In [34): 
out[34 
" pred " ]=np. where( " ] S, 1, 8) 
from sklearn.metrics import accuracy _ score 
accuracy_score pred"]) 
accurac score is 57%, this is underfitting or bias model where accuracy is Less 
on to of non-Linea data if you build Linear models it wilt always underfit and there Witt be biasness in the model 
e. 5798319327731 

To overcome these problem we are going to use deep learning methodologies

**Reasons to drop variable**

* Multi collinearity issues : one variable is influencing the other independent variable
* Over fitting
* One variable is not important and not contributing to dependent variable

**Moving average**

Build simple moving average method:

When simple model does not work or data is very complex then only we go for deep learning

# Deep Learning (Neural Network)

If we have nonlinear data then what will happen if we build logistic regression. It can’t e built. This is **drawback** the of logistic regression. Even if we apply log to non-linear data, if we don’t get non-linear data then we apply **deep learning** algorithms

## Neural network

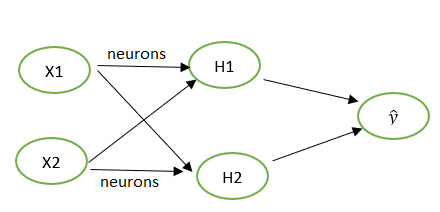
Machine generated alternative text:



Using X1, X2 (lower set in figure) build H1 model (logistic regression model)

Using X1, X2 (upper set in figure) build H2 model (logistic regression model)

Then using H1 and H2, build model to get y (logistic regression model)



Machine generated alternative text:
sample1X1 
mple2X1 
mple2X2 
samp e 
opulationH1 
probability values 
ationH2 
probability values 

Each node excluding the i/p node is called neurons. Here 1 hidden layer (also called **dense**) and 2 nodes or 3 neurons (H1,H2. )

Here 3 layers (1 input layer, 1 hidden layer and 1 o/p layer)

H1 and H2 is hidden layer. Above example where only one hidden layer(only one neuron layer) is called **Shallow neural network**

**Shallow neural network : when hidden layer =1**

**ANN : if Hidden layer is less than 3 greater than 1**

**Deep Neural Network : if Hidden layer is more than 3**

**MNN is if hidden layer is > 100**

Neural Network algorithm uses regularized Logistic Regression model.

In industry we use Neural Network for image process. No One uses it for structured data

Machine generated alternative text:



If accuracy of the model is very low then the model is called bias model

**There is no variable reduction technique in Deep Learning**

Step to perform neural network or components of neural network

1. Assigning random weights
2. Feed Forwarding
3. Error calculation
4. Back propagation
5. Finalize the model

Machine generated alternative text:
1 bias 
WI 
w2 
w3 

**Activation function**

* **Tanh (-1 to 1) =**
* **RelU**
* **Softmax --> for Image data**

**Main components/hyper parameter**

* Activation function
* Optimizer, cost func and training
* Regularization
* Tuning
* Classification vs regression tasks

**DNN basic architecture**

* Convolutions
* Recurrent
* Attention mechanism

**Application example: Relation extraction**

Back propagation

GANs & adversarial training

Bayesian Deep Learning

Generative models

Unsupervised/Pretraining

Machine learning

Input --> Feature Extraction --> Classification --> Output

Deep learning

Input --> Feature Extraction +Classification --> Output

Accuracy and what is happening inside is more in ML. DL is slow.

Each node is called neurons

Input 
TanH or RelU 
Neurons 
Hidden 
Tan 
(dense) 
Logistic egression 
Output 
or Relu ivation Function 
Sigmoid Activation Func 

**Combination of Input, Hidden and Output layer is called Perceptron or Sequential**

4+2 = 6 neurons (H1,H2,H3,H4,Y1,Y2)

[3\*4] + [4\*2] = 20 weights (non w0) (3 Weights of H1,H2,H3,H4 AND 4 weights on Y1,Y2)

1\*6 = 6 biases (w0 for H1,H2,H3,H4,Y1,Y2)

The input activation function for Neural N/W are

1. Tanh
2. RelU

The output activation function for Neural N/W is Sigmoid

Hidden Layers are having activation function are either Tanh or RelU

RelU

* When Data is non-linear the activation function is Tanh or RelU.
* Compared to TanH, RelU is having fast execution.
* The range of Relu is 0 to ∞.
* The full name is RelU is Regularized Linear Units or Rectified Linear Units
* By default all the deep learning algo will take an activation function as RelU

Tanh:

* Range of TanH activation function is -1 to +1

•1 to TanH activation 
Function 
-1 

Machine generated alternative text:
sample1X1 
mple2X1 
mple2X2 
samp e 
opulationH1 
probability values 
ationH2 
probability values 

H1 and H2 build on sample without replacement data. H1 and H2 samples can have overlapping data

ело 
w11 
w21 
w22 
w20 
Feed ForwardinB 

H1 = w10 + w11X1+w12X2+w13X3

H2 = w20 + w21X1+w22X2+w23X3

Error = y-

We check error at over all data and at each node level. Fix one node and check other nodes.

Checking the error from right to left is called back propagation

SGD is also called mini batch GD (gradient descent)

Building the model from left to right is feed forwarding

Unfortunately all the DL are overfit and are not there in production

uopeSed0Jd 
p. emhoe€ 
I apou_l 
p.e,wod paad 
papqeo aldwes 

As num of hidden layer increases, there is higher chance that model is overfit. There is no direct method to find the optimal number of hidden layer, we will have to go by trial and error method

Very rarely we use neural network in the industry. 80-90% of the problems are structured data. Very few projects in India are semi or un-structured data

Keras is wrapper of tensor flow

## Some Important algorithms to focus to get job in Indian Industry

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. Random forest
5. Clustering (K\_mean)

# SVM (Support Vector Machine)

The line which separates good vs bad in linear regression. If we are not able to separate then it’s a non-linear data

SVM is all about finding an optimal line to separate good vs bad.

* It is used only in NLP, only if data is small
* Execution time is very high
* Less than 1% people are using it in industry
* Main objective is to find the optimal line

* The decision boundary with largest margin
* SVM the large margin classifier
* Kernel trick

## The best classifier (Hyper Plane)

* The decision boundary which separates the classes
* Classification algo are all about finding the decision boundary
* A good classifier is the one which generalizes well. It should work well for both test and train data
* It need not be a straight line
* p=0.5, then log(loss function) = log() = log() = log(1) = 0 = B0+B1x (equation of straight line)
* The line which has maximum or best margin distance b/w the two different data points (1 and 0) is called **best classifier**
* Vector is combination of direction and magnitude
* We call data points as vector
* We calculate Euclidean distance b/w two data points, wherever we have min distance, those data points are called support vectors

1. SVC(C=1.0, = 1/alpha
2. cache\_size=200, --> 200mb
3. class\_weight=None,
4. coef0=0.0,
5. decision\_function\_shape='ovr', -> one over rest
6. degree=3, when kernel=polynomial then only this works. By default its 3 but it will not work for any other kernel
7. gamma='auto\_deprecated', 1/(num of feature) = each independent variable it will add the weight of gamma, to reduce the error.default in latest versions gamma value is 0
8. kernel='rbf', --> rbf and polynomial (non-linear data) will give the same result. Simple linear data the sigmoid funct and will behave as logistic regression.
9. max\_iter=-1, it takes 1000 iteration
10. probability=True, (by default it gives 0 or 1 unless we specify for probability values
11. random\_state=None,
12. shrinking=True,
13. tol=0.001,
14. verbose=False)

## Kernel

Types of Kernel

1. Linear
2. RBF
3. Polynomial
4. Sigmoid
5. Gaussian

Application:

* **kernel=linear -->if we use NLP data then**
* **kernel=rbf or polynomial --> for non-linear data**
* **kernel=sigmoid or rbf --> for linear data**
* **Polynomial Kernel** --> Best for image processing. Non Linear data
* **Sigmoid Kernel** --> Very similar to Neural network. Linear data, binary data or categorical data
* **Gaussian Kernel** --> No prior knowledge on data. Numerical data
* **Linear Kernel** --> Text classification
* **Laplace Radial Basis Function (RBF)** --> No prior knowledge on data. Non Linear data

If data is in **Hyper Plane,** we need to provide kernel as ploy or rbf

1. Classifier is a generic name, its actually called Hyper Plane.
   1. In 3-D system hyper planes are 2-D planes. In 2-D space, hyper planes are 1-D lines
2. SVM algo makes use of the nearest training examples to derive the classifier with max margin
3. Each data point is considered as a p-dimensional vector (a list of p numbers)
4. SVM uses vector algebra and mathematical optimization to find the optimal hyperplane that has maximum margin

* **Hard Margin** : If 0's are classified as 0s and 1s and 1s then it's called hard margin. If accuracy is 100%.
* **Soft Margin (slack)** --> If accuracy is not 100% or there are misclassifications then its soft margin. The notation is ξ

* SVM does not output probability. It directly classifies which class the new data point belongs to
* For a new point calculate

Non-Linear decision boundary

Kernel trick: to convert non-linear data into linear by apply higher order function e.g. square

* It increases num of dimensions or num of variables
* We are not dropping the original variable; we keep both transformed as well as original variable

# Timeseries

Difference b/w predict and forecast

Predict

* Never measured w.r.t. time
* We use random/stratified sampling

Forecast

* Related w.r.t. time
* We use sequential or systematic sampling
* Moving average method

|  |  |  |  |
| --- | --- | --- | --- |
|  | moving avg of 2 | moving avg of 3 |  |
| 20 |  |  |  |
| 30 | 25 |  |  |
| 40 | 35 | 30 |  |
| 50 | 45 | 40 |  |
| 60 | 55 | 50 |  |
| 70 | 65 | 60 |  |

For moving avg calculation of week, we calculate moving avg of 7

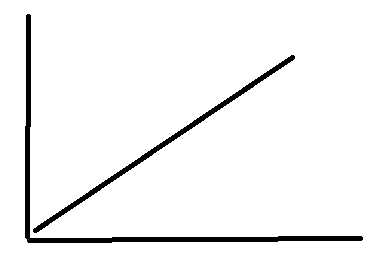
For year, moving avg of 12.

**There are 4 components in time series**

## Secular Trend (T)

Gradual long term movement

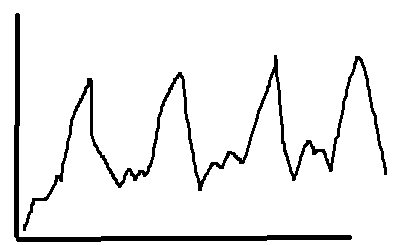
e.g Population growth in India



## Seasonality Pattern (S)

Results from events that are periodic and recurrent in nature

e.g Sales in a store in holiday season



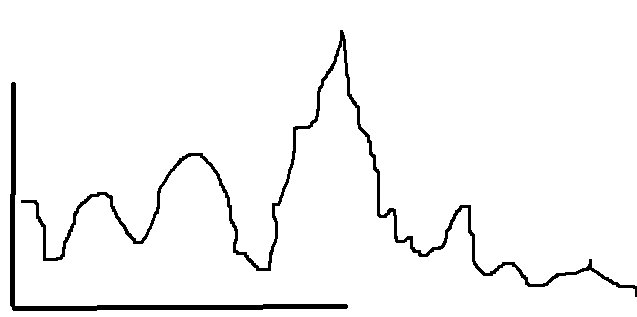
## Cyclic Pattern( C )

* + Results from events recurrent but not periodic in nature
  + Recession in US Economy



## Irregularity Component (I)

* + Disturbance or residual variation that remain after all the other behaviors have been accounted for
  + These are outliers (IQR) and missing values (impute the data by historical data last 3-4 periods)
  + Eg. Earthquake



**Steps**

1. Treat the irregularities
2. Then go for Secularity with help for moving average
3. If we identify and treat the seasonality and remove it then 99% of the chances cyclical pattern will also be treated automatically

**Equation**

O(t0 = T(t) + S(t)+ I(t)

O(t) = observed series

T(t) = Trend component

S(t) = Seasonal component

I(t) = Irregular component

**Objective:**

* **Step1:** Removing trend and seasonality factor and thus catching the random effect of the series
* **Step2:** We forecast the irregular component for the future ahead and then add the effect of trend and seasonality in order to get the original forecast
* Th process of removal of Trend component is called **de-trending**
* The process of removal of Seasonal component is called **de-seasonalization**
* We should have 30 periods of data to build the time series model

**De-Trending using Linear Regression**

* Linear detrend: Run a linear regression of X(t) on t and subtract that trend out
* **Advantage**: simple with well known properties

**De-trending using Moving average:**

* Calculate the moving avg

Machine generated alternative text:
Chart Title 
Passengers 
12-mvg-avg 

**3 kinds of data**

* **Timeseries data**
  + An order seq of values of a variable at equally spaced time intervals
  + e.g. Hourly temp reading, daily sales, monthly production
* **Cross Sectional data**
  + Similar nature of data. Higher chances to bias the model
  + Data on one or more var collected at the same point of time e.g census data
* **Panel Data**
  + A data set containing obs on multiple phenomena observed over multiple time periods is called panel data. e.g. avg speed of car b/w 9 am to 10 am over a week.

# NLTK

Natural Language Tool Kit

NLP Natural Language Processing

Nero Linguistic Process

Prop case:1st letter of the word is caps

## Tokenization

1st thing in NLTK is tokenization: The process of converting passage to paragraph, paragraph to sentences, sentences to words is called tokenization.

## Sentence tokenization

The process of converting paragraph to sentences

## Word tokenization

The process of converting sentences to words.

## stop words

The combination articles and prepositions are called as stop words. Total 179 stop words.

Importance of stop words. Only stop words can be used when we are performing **sentimental analysis**. Other than this usecase we never use stop words, we always remove stop words for other usecases.

e.g.

Positive sentence: X is a good boy

Negative sentence: X is not a good boy.

Highlighted one is stop word. If we remove it then both sentence becomes positive sentence

Stop words can be identified based on the words of a sentence and not at sentence level.

## Lemmatization or Normalization or stemming

Converting past tense and future tense into present tense. This also should be applied at word level and not at sentence level. If we use it at sentence level it will consider it letter by letter which is not we want.

EDA or summary statistics in text data is : count the freq of words

# TextBlob

<https://www.clips.uantwerpen.be/pages/mbsp-tags>

Parts of speech are also called named entity recognition or named entity information

[('European', 'JJ'),  
 ('authorities', 'NNS'),  
 ('fined', 'VBD'),  
 ('Google', 'NNP'),  
 ('a', 'DT'),  
 ('record', 'NN'),  
 ('$', '$'),  
 ('5.1', 'CD'),  
 ('billion', 'CD'),  
 ('on', 'IN'),  
 ('Wednesday', 'NNP'),  
 ('for', 'IN'),  
 ('abusing', 'VBG'),  
 ('its', 'PRP$'),  
 ('power', 'NN'),  
 ('in', 'IN'),  
 ('the', 'DT'),  
 ('mobile', 'JJ'),  
 ('phone', 'NN'),  
 ('market', 'NN'),  
 ('and', 'CC'),  
 ('ordered', 'VBD'),  
 ('the', 'DT'),  
 ('company', 'NN'),  
 ('to', 'TO'),  
 ('alter', 'VB'),  
 ('its', 'PRP$'),  
 ('practices', 'NNS')]

*How do we convert data into numerical format*

## Bag of words

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | How | are | you | doing | old |
| How are you doing | 1 | 1 | 1 | 1 | 0 |
| How old are you | 1 | 1 | 1 | 0 | 1 |

It takes the unique words from the data. The label them based on each sentence depending on if the word is present or not with 1 or 0 respectively.

## TF-IDF (Other name is word2vec)

TF-IDF : Term Frequency - Inverse Document Frequency

TF-IDF weight is a weight often used in information retrieval and text mining.

The wt is a statistical measure used to evaluate how important a word is to a document in a collection or corpus

Computed as log of the num of documents in the corpus divided by the num of doc where the specific term appears

TF =

IDF = log(

TF-IDF weight = TF \* IDF

TF-IDF value is always b/w 0 to 1

To normalize the data we use TF-IDF or word2vec.

TF-IDF is like MinMax scalar. Bag of words is like dummy coding.

# Sentimental Analysis

Polarity ranges from -1 to 1. This is sentiment score. If it is less than 0 the it is negative sentiment. If 0 then neutral. If greater than 0 then positive sentiment

# Subjectivity ranges from 0 to 1. If it is close to 0, there is no Subjectivity in data. If close to 1 then it has Subjectivity in data. This means if content is meaningful or not wrt sentiment

For text classification we can use only two algorithms

1. Naïve Bayes
2. SVM

# Agglomerative Clustering

If we have less than 4000 records then we cant build K-Means algorithm, we need to go for agglomerative algorithm.

There are 2 kinds of linkage

Single linkage --> Min distance from one data point to other data point

Complete linkage --> Max distance from one data point to other data point

## Dendogram

То Calculate the dusters we 
applied the Single linkage 
92 
95 
0.4 
0.22 
0.35 
0.26 
0.08 
0.45 
0.53 
0.38 
0.32 
0.19 
0.41 
0.3 
0.23 
0.22 
0.37 
0.34 
0.23 
0.23 
0.22 
0.37 
0.34 
0.15 
0.2 
0.14 
0.25 
0.15 
0.2 
0.14 
0.15 
0.28 0.29 
0.11 0.22 0.39 
рз,Р6Р4 95 
Р2,Р5 рз,Рб И 
рз,Рб 
0.15 
0.28 
0.29 
Р2,Р5 
рз,Рб 
0.22 
0.23 
0.22 
0.37 
0.15 
0.2 
0.15 
Р1 94 
0.22 
0.15 
AggIomative_c1ustering 
0.37 
Sheet2 k-means Sheet41 (9) 

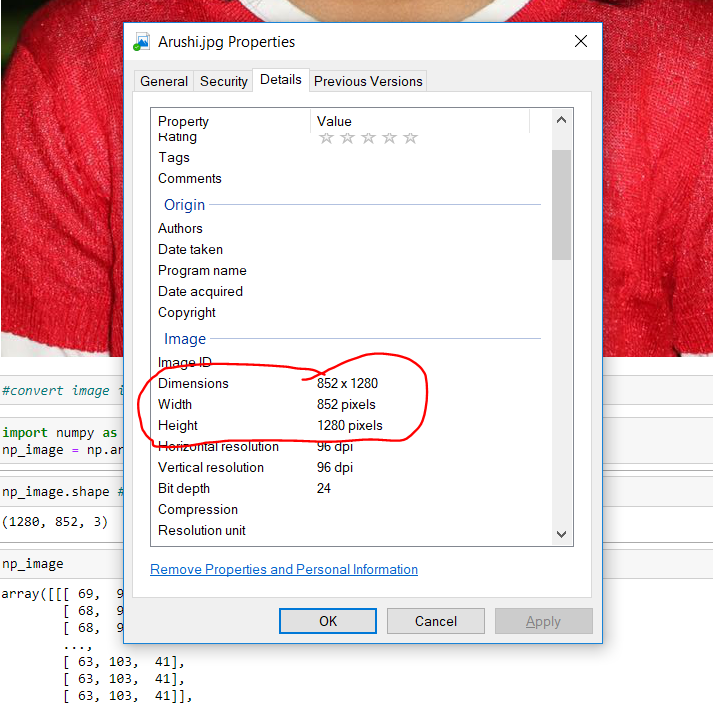
Drawback:

If one record is different (no relation ) it will increase number of clusters

# CNN (Image Classification)

By default every image is having 3 basic color: RGB

* If all 3 are 33.33% then its white color
* By default image pixels are 256\*256 (length and height)
* In image smallest dotted points are called pixels
* Say if we have an image which has 64\*64 size. Then total pixels are 1\*64\*64\*3 (height\*width\*RGB) = 12288 pixels



Step for Image classification

## Step1: Pooling

If processing multiple images, Image having max pixel, consider that as best image, and convert remaining image to have same pixel.

In the array of image having lesser pixel than best image, we add the 0s in the array for missing values.

## Step2: Standardization of data: (Flattening the data)

we need to normalize the data b/w 0 - 1, so we divide each value with 255 to

To normalize the image pixels, we need to divide each and every image value by 255

## Step3: Convolution

**Transpose the data:**

Image1-Transpose

Image2-Transpose

In (2eJ: 
out(2eJ: 
np_i mage 
for 
69 
68, 
each 
97 
96 , 
96, 
row 
46] ' 

In 123): 
out(231: 
np_imagel 
43, 
44, 
511, 

**Transpose**

|  |  |
| --- | --- |
| Image | Image1 |
| 69 | 43 |
| 97 | 44 |
| 46 | 48 |
| 68 | 45 |
| 96 | 46 |
| 45 | 48 |
| 68 | 48 |
| 96 | 49 |
| 45 | 51 |

**Take Convolutions**

Convolutions are 3\*3 matrix. Need to cover R, G,B value hence 3\*3 matrix.

If we have an image 1\*64\*64\*3 then Max num of convolutions will be 12288

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Image | Image1 |  |  |
| 0 | 0 | 69 | 43 | 0 | 0 |
| 0 | 0 | 97 | 44 | 0 | 0 |
| 0 | 0 | 46 | 48 | 0 | 0 |
| 0 | 0 | 68 | 45 | 0 | 0 |
| 0 | 0 | 96 | 46 | 0 | 0 |
| 0 | 0 | 45 | 48 | 0 | 0 |
| 0 | 0 | 68 | 48 | 0 | 0 |
| 0 | 0 | 96 | 49 | 0 | 0 |
| 0 | 0 | 45 | 51 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

Then out of each 3\*3 matrix, create 2\*2 matrix.

From each 3\*3 matrix we can create 20 matrix of 2\*2 matrix

From 2\*2 matrix we are taking --> Influence points

In image (shape say (209,50) first will be columns ie 209 columns and 2nd will be rows i.e 50 rows

Machine generated alternative text:



H1,H2…H7 are samples. At each hidden layer it checks if data is linear or not, it not then create next hidden layer.

## Step4: Max Polling:

Max influence point if data

## Step5: Drop Out

# Model Deployment

Model Deployment:

1. Different tools and approaches
2. It will serve 2 purpose
   1. Testing
   2. Production
   3. Very similar to Jupyter but can work on distributed envt
      1. Qubole: It can read data from distributed system
      2. Zeppelin --> Spark code can be written
      3. Sage maker

frontEnd can be

* PHP
* Nodejs
* Java
* #Net

MicroServices--> Web Services (API)

1. ReST API
   1. URL--> End Points--> These work with http protocol. GET,**POST**,DELETE,PUT
2. SOAP API

We should have model deployed in some server. That server should be able to

* take the request,
* invoke the model,
* give the result of model back to the application

Majority prefer Json file to communicate

There are several ways how e can deploy or host the server

* TomCat
* IBM Websphere
* Iracle weblogic
* JellyFish
* Python Flask (light weight web server)

Some useful webservices available in market

* Google Vision API
* Google Lense
* Google Maps
* IBM Watson API
* Azure Insights --> Vision APIs

Postman (GUI) where we can give the endpoint url and then get the response back

1. Once model is finalized
2. Serialize the model and dump the model in .pkl file (pickle file): Persisting the state of object into disk
3. In .pkl file we are passing lr,knn,gb value and columns (here lr=lr.fit(x\_train,y\_train). It’s a binary file
4. Load the pkl file and deserialize it (from hard disk to the memory, we got the information)
5. We do now do model.predict

# Imbalance data classification

Imbalance data classification: It is applicable for classification problem where dependent variable is binary.

Event Rate = Num of events/total number of records

If event rate is less than 6%, then that is called **imbalanced data.** And when we build classification on top of it, then it is called **imbalanced classification**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 0 | 1 |
| 0 | TP | FN | 25000 | 0 |
| 1 | FP | TN | 3000 | 0 |

Accuracy might not give correct picture, it might be biased towards positive or negatives . Like the example above.

To overcome this problem, we apply different methods.

## Over sampling (99% we use this)

This happen mostly in banking domain. Here we add actual event data (proxy data) to calculate event rate until it c=becomes more than 6%.

We do not take out non event data.

Suppose we have 1000 records. Out of which 10 are fraud and 990 are non-fraud.

Step1: 990

We calculate precision and recall, instead of accuracy

|  |
| --- |
| **Precision =** |
| **Recall (or sensitivity) =** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step1 | event | 10 |  |  |
|  | Total | 1000 |  |  |
|  | Event Rate | 1 |  |  |
|  |  |  |  |  |
| Step1 | Event + proxy Data | 20 | added 10 more records which are event |  |
|  | Total + proxy Data | 1010 |  |  |
|  | Event Rate | 1.980198 |  |  |
|  |  |  |  |  |
| Step3 | Event + proxy Data | 30 | added 10 more records which are event |  |
|  | Total + proxy Data | 1020 |  |  |
|  | Event Rate | 2.941176 |  |  |
|  |  |  |  |  |
| Step4 | Event + proxy Data | 40 | added 10 more records which are event |  |
|  | Total + proxy Data | 1030 |  |  |
|  | Event Rate | 3.883495 |  |  |
|  |  |  |  |  |
| Step4 | Event + proxy Data | 50 | added 10 more records which are event |  |
|  | Total + proxy Data | 1040 |  |  |
|  | Event Rate | 4.807692 |  |  |

Repeat steps till we get event rate more than 6%.

Pip install imblearn

From imblearn.under\_sampling import RandomOverSampler

In imblearn there is a bug. It considers 50% of events and 50% of event data which is not correct. There is no need to build the model in this case. That’s why we don’t use imblearn in practical scenarios.

## Under sampling

* 1. Here we are not going to change actual event data. We take out non event data.

|  |  |  |
| --- | --- | --- |
| Step1 | event | 10 |
|  | Non-fraud | 990 |
|  | Event Rate | 1.010101 |
|  |  |  |
| Step2 | event | 10 |
|  | Non-fraud - some % of total data  (say 10%) | 891 |
|  | Event Rate | 1.109878 |
|  |  |  |
| Step3 | event | 10 |
|  | Non-fraud - some % of total data  (say 10%) | 801.9 |
|  | Event Rate | 1.231679 |
|  |  |  |
| Step4 | event | 10 |
|  | Non-fraud - some % of total data  (say 10%) | 721.71 |
|  | Event Rate | 1.366662 |
|  |  |  |
| Step5 | event | 10 |
|  | Non-fraud - some % of total data  (say 10%) | 649.539 |
|  | Event Rate | 1.516211 |

Repeat steps till we get event rate more than 6%. We don’t prefer this as we loose information.

Pip install imblearn

From imblearn.under\_sampling import RandomUnderSampler

In imblearn there is a bug. It considers 50% of events and 50% of event data which is not correct. There is no need to build the model in this case. That’s why we don’t use imblearn in practical scenarios.

## SMOTE technique

# Steps of model building

## For Regression Problems

Who: is going to leave

When: TimeFrame

Why: What is the reason for churn

What:

How : to control

**Missing value**

Business Objective

**Outlier detection:**

Boxplot

Qqplot

IQR

Standard Scalar (-3 to +3)

Data Gathering Summary stats for all i/p files, then combine all I/P to create master data

S

Data processing

**Variable Reduction**

Pairwise

VIF

stepwise (T-Value)

Lasso

P-Value

F-stat/AIC (R-Value)

PCA

**Normality Check:**

Standard Scalar (-3 to +3)

Qqplot

histogram

distplot

JB

EDA (clean data)

Train Data

Test Data

**Model Building**

**(Linear Regression)**

MSE

RMSE

MAE

Check Accuracy

Diff > 10%

Overfit

Gradient Descent to increase accuracy

Regularization (Ridge)

Finalize Model

## For Classification Problems

Who: is going to leave

When: TimeFrame

Why: What is the reason for churn

What:

How : to control

**Missing value**

Business Objective

**Outlier detection:**

Boxplot

Qqplot

IQR

Standard Scalar (-3 to +3)

Data Gathering Summary stats for all i/p files, then combine all I/P to create master data

S

Data processing

**Variable Reduction**

Information Value

VIF

P-Value

z-Value (T-Value or Step Wise)

Feature Importance

AUC/ROC

**Normality Check:**

Standard Scalar (-3 to +3)

Qqplot

histogram

distplot

JB

EDA (clean data)

Train Data

Test Data

**Model Building**

**(Classification)**

MSE

RMSE

MAE

Check Accuracy

Diff > 10%

Overfit

Gradient Descent to increase accuracy

Regularization (Ridge)

Finalize Model

# Telecom Domain

Debt ratio: expenditure/salary

Different KPIs used in telecom industry

* Usage metrics
  + Voice calls
    - Inbound
      * ICU Inter connect charge units
    - outbound
  + Data usage
  + VAT (Value added top-up)
  + Roaming
  + SMS
  + MMS
* Billing
* Payment
* Network
* Complaints
* Demographics

VOLTE: Voice Over Long Term Evolution

PPP Public private partnership

NCTUE: [National Consumer Telecom & Utilities Exchange](https://www.nctue.com/)

CIBEL: Credit Information Bureau (India) Limited

Only India and Italy has pre-paid services

Credit reporting Bureau

* Experian
* Equifax
* TransUnion

Some of the derived variables used in telecom industry are

* ARPU: Avg Revenue per unit
* Avg Number of calls, 3 months vs 6 months vs 1 Year
* Avg data usage, 3 months vs 6 months vs 1 Year
* Avg bill amount, 3 months vs 6 months vs 1 Year
* Avg payment, 3 months vs 6 months vs 1 Year
* Avg complaints, 3 months vs 6 months vs 1 Year
* NRC charges (Non-Recurring Charges): N/w validity charges
* Tenure of the customer
* Offers, discount charges
* CDR call data recording

Main problem is

* customer churn
* Propensity: Probability to buy
* Network fraud

Machine generated alternative text:
$0.5 
Call conference 
USA 
$2 
$1 
India 
China 
Call conference 

To perform the analysis we aggregate the Tx level data to customer level data

Machine generated alternative text:
Who: is going to leave 
Business Objective 
When: TimeFrame 
Why: What is the reason for churn 
What: 
How : to control 

How many i20 cars are there in Bangalore, by using

Who: Who has i20 cars, avg income of ppl

When: When they bought it

Why: What is the reason

How

What is the total population of Bangalore: 1 crore

How many families have kids

Less than 18 year do not have cars say 20%

For 80% population is 80,00000

Total population --> remove kids population --> remove unemployed --> how many of the people have capacity to buy 4 wheeler --> some population has only 1 car --> say 20% of share of cars is Hundai --> say 30% of Hundai share is I20

# Retail Business

Aspect:

* Promotions
* Who are potential customers
* Knowing ur customer
* How does a business go about doing this

**Customer Life cycle phases**

Customer acquisition --> Item Upsell --> Service and Support --> Customer Retention

Upsell --> sell related product to a customer at the same time

Cross sell -> sell unrelated products to a customer at the same time

Predictive customer analytics

Data

1. Customer
2. Product
3. Age
4. Income

Estimate of project

|  |  |
| --- | --- |
| **High level Tasks** | **Weeks** |
| Understanding business problem | 1 -2 |
| Data gathering | 1 -2 |
| Master Data | 2 |
| Processing | 2 |
| EDA | 1 |
| Variable creation | 1 |
| Variable reduction | 2 - 3 |
| Model building | 2 - 3 |
| Model recommendation | 1 |
| Documentation | 1 |

# Classification Project

1. Business Objective
   1. Who is going to close the loan before their tenure period (Fore closure customer)
   2. Fix the population
   3. Historical data and profile level (demographic level information)
   4. Take at least 1 year of train data to capture seasonality. In practice they consider 3 years data
   5. Test data should be latest 3 periods
   6. From observation data or from snapshot period, we will get independent variable. Observation window
   7. From performance data we will derive the dependent variable. Performance window
   8. Train window (Observation window)
   9. Test Window (Performance window)

1. Machine generated alternative text:
   Sep'17 
   Observation Period 
   Historical Data 
   sep 18 
   Snapshot eriod 
   Preformance Period 
   Sep'19 
   Nov'19 

Business Objective

Data Gathering Summary stats for all i/p files, then combine all I/P to create master data

S

Exclusions

S

**Missing value**

**Outlier detection:**

Boxplot

Qqplot

IQR

Standard Scalar (-3 to +3)

Data processing

Clustering Algo

Cluster3

Cluster2

Cluster1

**Normality Check:**

Standard Scalar (-3 to +3)

Qqplot

histogram

distplot

JB

**Variable Reduction**

Information Value

VIF

P-Value

z-Value (T-Value or Step Wise)

Feature Importance

AUC/ROC

EDA (clean data)

Train Data

Test Data

**Model Building**

**Classification**

MSE

RMSE

MAE

Check Accuracy

Diff > 10%

Overfit

Gradient Descent to increase accuracy

Regularization (Ridge)

Finalize Model

Different Variable Reduction for classification problems are

1. Pairwise correlation

2. VIF

3. Information Value

4. P-Value

5. F-Stat/AIC

6. PCA

7. Feature Importance

8. AUC/ROC

9. z-Value (T-Value or Step Wise)

But in industry

1. Feature Importance

2. Pairwise

# Important Links

https://www.youtube.com/user/joshstarmer/videos