UNIT - 1

Introduction to Data science, benefits and uses, facets of data, data science Process in brief, big data ecosystem and data science.

Data science Process: overview, defining goals and creating Project charters, retrieving data, cleaning, integrating and transforming data, exploratory analysis, model building, Presenting findings and building applications on top of them.

Introduction to data science:

Data science involved using methods to analyze massive amounts of data extract the knowledge it contains. You can think the relationship between the big data and data science as being like the relationship blw the crude oil and oil refinery. Data science and big data evolved from statistics and traditional data management but are now considered to distinct disciplines.

the characteristics of big data are often referred to as the three Vs:

* volume → How much data is there?
* variety → How diverse are different types of data?
* velocity → At what speed is new data generated?

The often these characteristics are complemented with a fourth V is veracity → How accurate is the data.

Data science is an evolutionary extension of statistics capable of dealing with the massive amount of data produced today. It adds to methods from computer science to the repertoire of statistics.

The main things that sets a data scientist apart from a statistician are the ability to work with big data experience in machine learning, computing and algorithm building.

Tools:

The tools tend to differ too, more frequently mentioning the ability to use..

→ Hadoop
→ Pig
→ Spark
→ R,
→ Python and Java.

for instance, almost every popular nosql data base has a Python specific API.

As the amount of data continues to grow and the need to leverage it becomes more important, every data scientist will come across big data Projects throughout the career.

Benefits and uses:

Data science are used almost everywhere in commercial and non-commercial settings. The no. of use cases is vast and, examples we'll Provide throughout this book only scratch the surface of the possibilities.

Financial institutions use data science to Predict stock markets, determine their risk of lending money and learn how to attract new clients for their services.

— Government Organisation, Communication Headquarters
— Non Government Organisations, Universities.

At the time the atleast 50% of trades world wide are performed automatically by machines based on algorithms developed by quants, as data scientists who work on trading algorithms are often called, with the help of big data and data science techniques.

Benefits of data science:

→ Informed decision making:

Data science helps organizations make data driven decisions, minimizing guesswork and maximizing efficiency.

→ Trend Prediction:

By analyzing past data, it defines future trends, helping business stay ahead in competative markets.

→ Automation of Tasks:

ML models automate repetative tasks, saving time and reducing human error.

→ Improved customer experiences:

DS personalize user experiences, such as Product decommendations or targeted advertisements.

→ Efficiency and optimization:

It streamlines operations by identifying inefficiencies and suggesting improvements.

→ Problem solving.

It uncovers hidden patterns or issues in complex data sets, leading to innovative solutions.

Uses of Data science:

*) Data science uses:

1) Business
2) Healthcare
3) Finance
4) Ecommerce
5) Social media
6) Transportation
7) Government and Policy
8) Entertainment
9) Education
10) Environmental science.

Facets of Data:

In, data science you will come across many different types of data, and each of them tends to require different tools and techniques.

The main categories of data are these:

* structured
* unstructured
* Natural language
* Machine-generated
* Graph-based
* Audio, video and Images
* streaming.

1) Structured data:

Structured data is data that depends on a data model and resides in a fixed field within a record. structured Query language (SQL) is the preferred way to manage and query data that resides the database. You may also come across structured data that might give you a hard time storing it in a traditional relational database.

Fig: An Excel table example of structured data

1.	Indicator ID	Dimension list	Time frame	numeric value	confidence into.
2	21439083O	Total (age-adjusted)	2008	74.6%.	73.8%.
3	21439083I	Aged 18-44 years	2008	59.4%.	58.0%.
4	21439083Z	Aged-18-24 years	2008	37.4%.	39.6%.
5	21439083S	Aged 55-64 years	2008	91.5%.	90.4%.
6	21439083H	Male (age-adjusted)	2008	72.2%.	71.1%.

2) Unstructured data:

→ Unstructured data is data that is not easy to fit into a data model because the content is context-specific or varying.

→ Although email contains structured elements such as sender, title and body text, it's a challenge to find the no. of people who have written in email complaint about a specific employee because so many ways to exit to refer to a person.

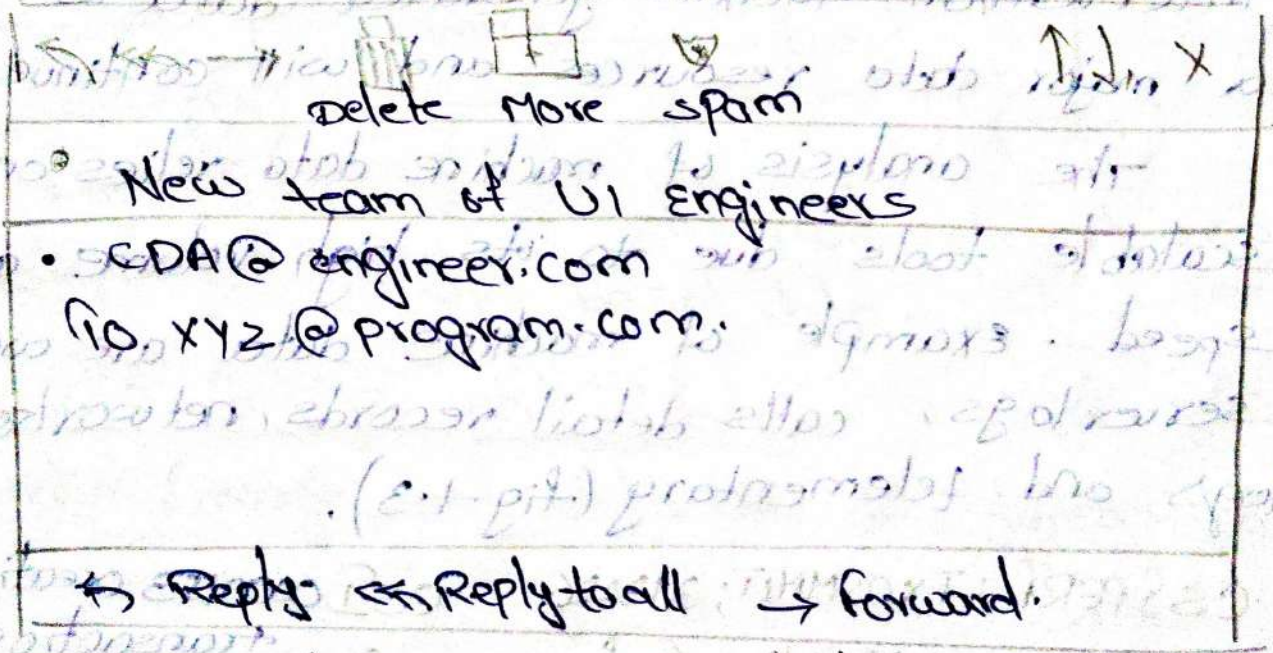


Fig: Email is simultaneously an example of unstructured data and natural language data.

Natural language data:

Natural language is a special type of unstructured data., it's challenging to Process because it requires knowledge of specific data science techniques and linguistics.

Machine generated data:

Machine generated data is information that's automatically created by computer, process, application or other machine without human intervention. Machine generated data is becoming a major data resources and will continue to do so.

The analysis of machine data relies on highly scalable tools due to its high volume and speed. example of machine data are web server logs, calls detail records, network event logs and telementary (fig-1.3).

CS[PERF:TXCOMMIT;313236 2014-11-20 11:36:13, Info 69), objectname (6)" (null)"
2014-11-28 11:36:13 Info result 0x000000, handle @0x4e4

CS[000153 creating NT transaction.
CS[000154 " "
CS[00015 @2014/11/28: 10:36:13.471

fig: example of machine generated data.

Graph-based or network data:

"Graph data" can be a confusing team because any data can be shown in graph. "Graph" in this case points to mathematical graph theory. a graph is a mathematical structure to model pair-wise relationships b/w objects. Graph based data is a natural way to represent social n/w's.

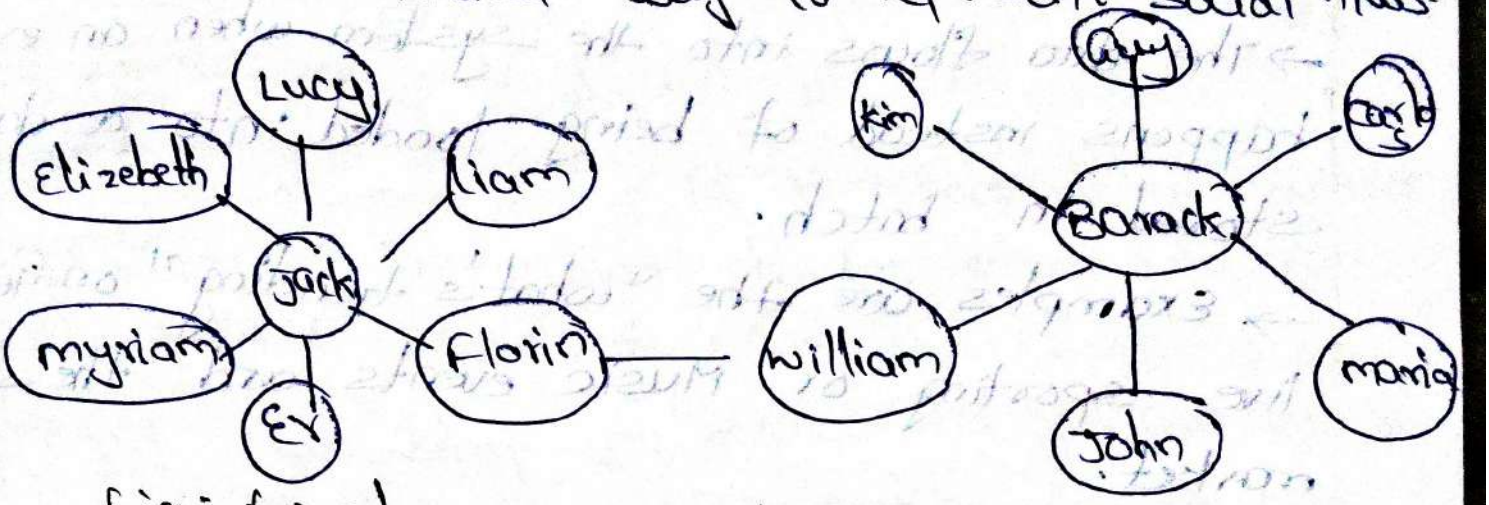


Fig: friends in a social media network are an example of graph based data.

Audio, image and video:

Audio, image and video are data types that pose specific challenges to a data scientist.

→ Tasks that are trivial of humans, such as recognizing objects in pictures, turn out to be challenging for computers.

→ MLBAM (major league Baseball Advanced Media) announced in 2014 that they will increase video captured to approximately 7 TB per game purpose in live.

the learning algorithms takes in data as it's produced by the computer game, its streaming data.

Streaming data:

while streaming data can take almost any of the previous forms, it has an extra property.

→ the data flows into the system when an event happens instead of being loaded into a data stored in batch.

→ Examples are the "what's trending" on Twitter, live sporting or Music events and the stock market.

II → Data Science Process in Brief:

The data science process typically consits of six steps, as you can see in the mind map in fig. we will introduce them briefly.

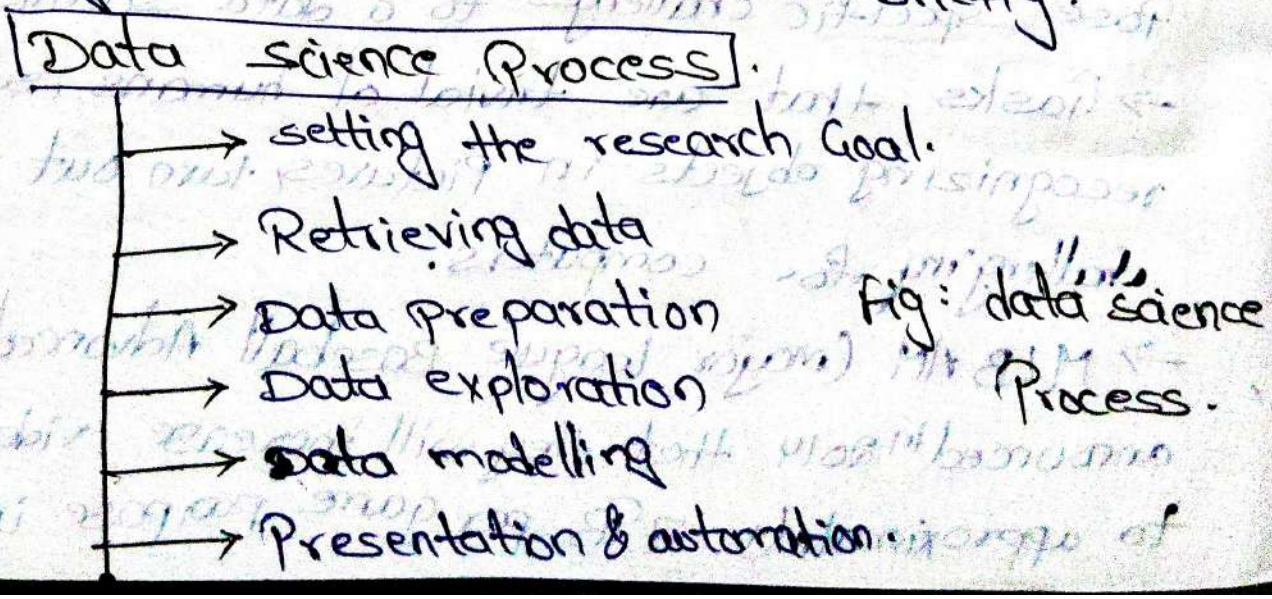


fig: data Science Process.

1) Setting the research goal:

→ Data science is mostly applied in the context of an organization.

→ Throughout this book, the data science process will be applied to bigger case studies and you will get an idea of different possible research goals.

2) Retrieving Data:

→ The second step is to collect data. You have stated in the project charter which data you need and where you can find it.

→ In this step you ensure that you can use the data in your program, which means checking the existence of, quality and access to the data.

3) Data Preparation:

Data collection is an error-prone process; in this phase you enhance the quality of the data and prepare it for use in subsequent steps.

→ this phase consists of three subphases like---

1) Data cleansing
2) Data integration
3) Data transformation.

4) Data exploration:

→ Data exploration is concerned with building a deeper understanding of your data.
→ To acheive this you mainly use descriptive statistics, visual techniques, and simple modeling.
→ This step often goes by the abbreviation EDA, for exploratory Data Analysis.

5) Data Modeling (or) Model Building:

→ In this phase you use models, domain knowledge, and insights about the data you found in the previous steps to answer the research question.
→ Building a model is an iterative process that involves selecting the variable for the model, executing the model and model diagnostics.

6) Presentation and Automation:

→ finally, you present the results to your business
⇒ these results can take many forms, ranging from presentations to research reports.

11→ Big data ecosystem and Data science:

Currently many big data tools and frameworks exist, and it's easy to get lost because new technologies appear rapidly. It's much easier once you realize that the big data ecosystem can be grouped into technologies that have similar goals and functionalities.

Let's look at the different group of tools in this diagram and see what each does. We will start with distributed file systems.

Distributed file systems:

A distributed file systems is similar to normal file system, except that it runs on multiple servers at once. Actions such as storing, reading and deleting files and adding security to files are the core of every file systems including distributed one. Advantages of distributed file systems:

→ they can store files larger than any one computer disk
→ the system scales easily: you are no longer bound by the memory or storage restrictions of single servers.

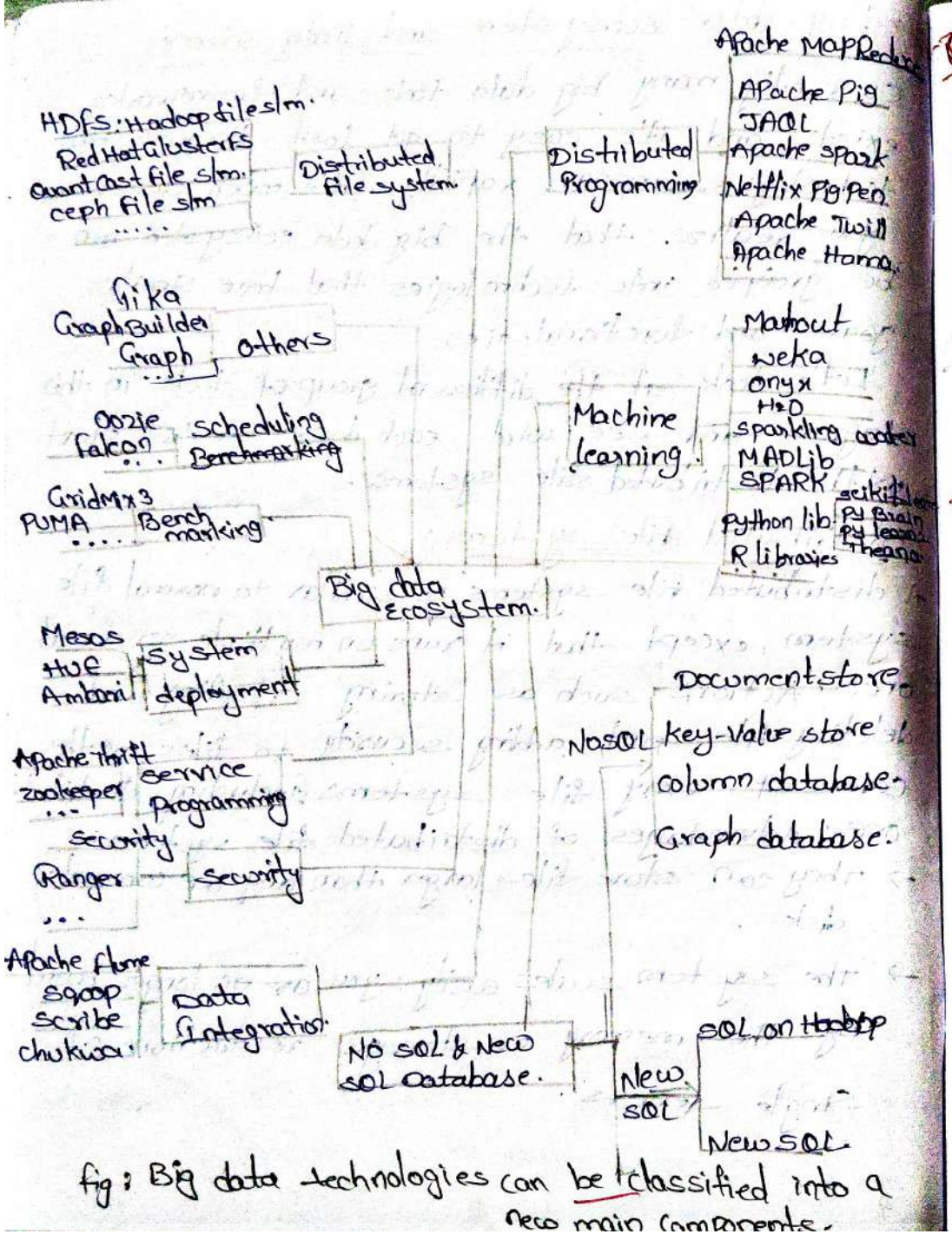


fig: Big data technologies can be classified into a few main components.

(2) Distributed Programming framework:

Once you have the data stored on the distributed files system, you want to exploit it. One important aspects of working on a distributed hard disk is that you won't move your data to your program, but rather you will move your program to the data. when you start from scratch with a normal general purpose programming language such as C, Python and Java.

(3) Data Integration framework:

once you have a DFS in place, you need to add data. you need to move data from one to another and this where the data integration framework such as Apache sqoop and Apache flume excel.

→ The process is similar to an extract, transform and load process in a traditional data warehouses

4) Machine learning framework:

When you have the data in place, it's time to extract the coveted insights. This is where you rely on the fields of machine learning, statistics and applied mathematics.

In this framework machine learning libraries are to, Python is

→ PyBrain for neural networks
→ NLTK or natural language Toolkit
→ Pylearn 2
→ Tensor Flow.

5) NoSQL data bases:
If you need to store huge amount of data, you require software that's specialized in managing and querying this data. Traditionally this has been the playing field of relational databases such as Oracle SQL, MySQL, sybase IQ and others.
→ Many different types of databases have arisen, but they can be categorized into the following types:

* column database → Data stored in columns.
- Document stores → Document stores no longer use tables.
- streaming data → Data is collected, transformed and aggregated.
- key-value Re:stores → Data is stored in table.
- SQL on Hadoop
- New SQL.
- Graph-databases.

5) scheduling tools:
Scheduling tools help you automate repetitive tasks and trigger jobs based on events such as adding a new file to a folder. You can use them, for instance, to start a Map Reduce task whenever a new dataset is available in dictionary.

7) Benchmarking tools:
This class of tool was developed to optimize your big data installation by Providing standardized Profiling suites. For example, if you can gain 10% on a cluster of 100 servers, you save the cost of 10 servers.

8) System Deployment:
Setting up a big data infrastructure is not easy task and assisting engineers in deploying new applications into a big data clusters is where system deployment tools shine. This is not core task of data scientists.

9) Service programming:
Data scientists sometimes need to expose their models through services. The best known example is the REST service; Rest stands for representational state transfer. It's often used to feed websites with data.

10) Security:

Big data security tools allow you to have central and fine-grained control over access the data. In this book we don't describe how to setup security on big data because this is a job for the security expert.

IV) Data Science Process:

Overview:

The typical data science process consists of six steps through which you will iterate, as shown in figure.

1. The first step of this process is setting a research goal. The main purpose here is making sure all the stakeholders understand the what, how, and why of the project etc...
2. The second phase is data retrieval. You want to have data available for analysis, so this step includes finding suitable data and getting access to the data from the owner.
3) Now that you have the raw data, it's time to prepare it. this includes ~~transforma~~ transforming the data from a raw form into data that's directly usable in your models. If you have successfully completed this step, you can progress to data visualization and modeling.
4) The fourth step is data exploration. the goal of this step is to gain a deep understanding of the data. the insights you gain from this phase will enable you to start modeling.
5) Finally, we get to the sexiest part: Model building It is now that you attempt to gain the insight & make the predictions stated in your project charter.
6) The last step of the data science model is presenting your results and automating the analysis. if needed. one goal of a project is to change a process and/or make better decisions. certain projects require you to perform the business process over and over again. so automating the project will save time.

Data science Process:

- Define research goal.
- 1. Setting the research goal — create project charter.
- 2. Retrieving data — internal data / external data — Data Retrieval, Data ownership.
- 3. Data preparation
  - data cleansing: errors from data entry, physically impossible values, missing values, outliers, spaces, types, -- errors against codebook, aggregated data.
  - data transformation: extrapolating data, derived measures, creating dummies, Reducing no. of variables:, Merging/joining data sets, combining data, set operators, creating views.
- 4. Data exploration: Simple graphs, combined graphs, link and brush, Non graphical techniques.
- 5. Data Modeling: Model and variable selection, Model execution, Model diagnostic and model comparison.
- 6. Presentation & automation: Presenting data, Automating data analysis

Fig: the six steps of the data science Process.

Defining goals and creating a Project charter:

A Projects starts by understanding the 'what', the why and the how of your Project in figure.

Data science Process

- 1. setting the research goal — define research goal, create project charter
- 2. Retrieving the data
- 3. Data Preparation
- 4. Data exploration
- 5. Data Modeling
- 6. Presentation and automation.

Fig: setting the research goal.

Spend time understanding the goals and context of your research:

An essential outcome is the research goal that states the purpose of your assignment in a clear and focussed manner. Understanding the business goals and context is critical for Project success. Don't skim over this phase lightly. Many data scientists fail here: despite their mathematical wit and scientific brilliance, they never seem to grasp the business goals and context.

1) Create a project charter:

Clients like to know upfront what they are paying for, so after you have a good understanding of the business problem, try to get a formal agreement on the deliverables.

→ A project charter requires teamwork, and your input covers at least the following:

- A clear research goal.
- The project mission and context.
- How you are going to perform your analysis.
- What resources you expect to use.
- Deliverable and a measure success
- A timeline.

2) Retrieving Data: (2)

The next step in data science is to retrieve the required data in figure. Sometimes you need to go into the field and design a data collection process yourself, but most of time you won't be involved in this step. In this data more and more organizations are making even high quality data freely available for public and commercial use.

Data science Process:

- setting the research goal.
- Retrieving data
  - Internal data
  - External data
- Data retrieval
- Data ownership.
- → Data preparation
- → Data exploration
- → Data modeling
- → Presentation and automation.

(a) start with data stored within the company.

Your first act should be to assess the relevance and quality of the data that's readily available within your company. Most companies have a Program for maintaining key data and so much of cleaning work may already done. This data stored in official data repositories such as databases, data marts, data warehouses and data lakes maintained by team of IT Professionals. Getting access to the data may take time and involve company politics.

(b) Don't be afraid to shop around:

Many companies specialize in collecting valuable information. Other Companies Provide data so that you, in turn, can enrich their service and ecosystem. such is case with Twitter, linkedin and facebook.

The below figure explain to shows only a small selection from the growing number of open data Providers.

open data site	Description.
Data.gov	the home of the US Gounment's open data.
https://open-data.europa.eu/	the home of the European commision's open data.
Freebase.org	An open database that retrieves its information from sites like wikipedia
Data.worldbank.org	open data initiative from the world bank.
Aiddata.org	open data for international development.
Open.fda.org	open data from the US food and Drug administration.

fig: A list of open data providers that should get you started.

3.4.4 (c) Cleaning, Integrating and transforming Data:

The data received from the data retrieval phase is likely to be "a diamond in the rough." you task now is to sanitize and Prepare it for use in the modeling and reporting phase.

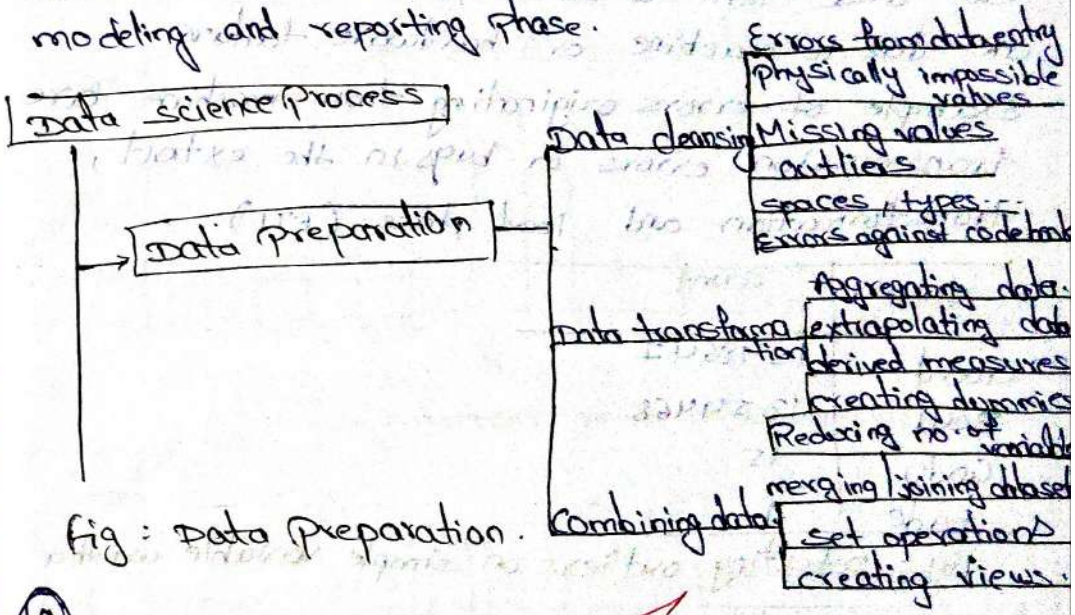


fig: Data Preparation.

(a) Cleansing Data:

Data cleansing is a subprocess of the data science Process that focuses on removing errors in your data so your data becomes a true and consistent representation of the process it originates from.

b) Data entry errors:

Data collection and data entry are error-prone processes. But data collected by machines or computers is not free from error either. Errors can arise from human sloppiness, whereas others are due to machine or hardware failure. An example of errors originating from machines are transmission errors or bugs in the extract, transformation and load phase (ETL).

value	count
Good	1598647
Bad	1354468
Godo	15
Bade	1

fig: Detecting outliers on simple variable with a frequency table.

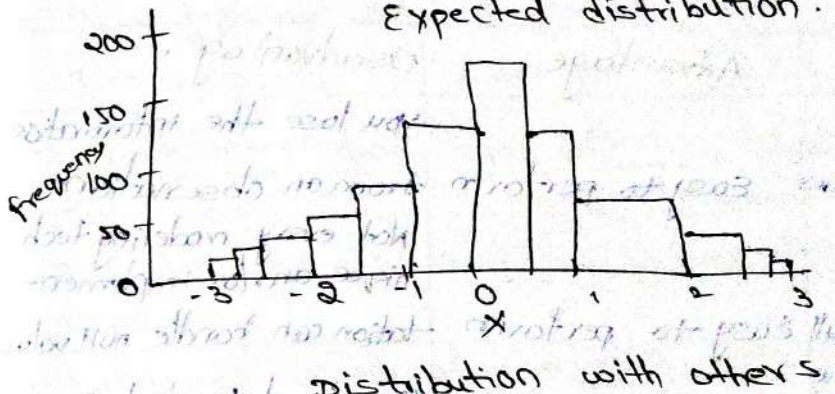
Most errors of this type are easy to fix with simple assignment and if-then-else rule.

```
if x == "Godo":
    x = "Good"
if x == "Bade":
    x = "Bad"
```

c) Outliers:

→ An outlier is an observation that seems to be distant from other observations or more specifically one observation that follows different logic.

→ The easiest way to find outliers is to use a plot or table with min and max values; shown in fig-



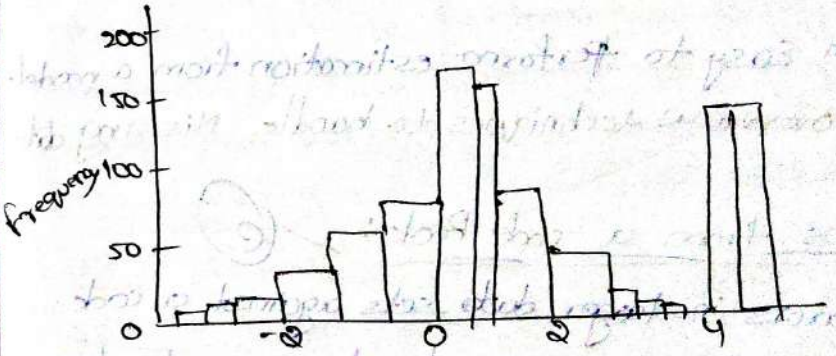


fig: Distribution plots are helpful in detecting outliers and helping you understand the variable.

3) Dealing with Missing values: (d)

Missing values are not necessarily wrong, but you still need to handle separately certain modeling techniques can't handle missing values.

common techniques data scientist use are listed in table.

Technique	Advantage	disadvantage.
omit the values	Easy to perform	you lose the information from an observation
set value to null	Easy to perform	Not every modeling technique and/or implementation can handle null values
impute a static value such as 0 or the mean	Easy to perform	Can lead to false estimation from a model

fig: An overview techniques to handle Missing data

4) Deviations from a code Book: (e)

Detecting errors in larger data sets against a code book or against standarized values can be done with help of set operations. A code book is a description, of your data a form of metadata.

(f) Different levels of Aggregation:

→ Different levels of aggregation is similar to having different types of measurements.

→ An example of it would be a data set containing data per week versus one containing data per work week.

→ this type of data is generally easy to detect and summarizing, the data set will fix it.

→ Combining data from different data sources:

Data comes from several different places, in this substep we focus on integrating these different sources.

• Different ways of combining data:

The data can be perform two operations to combine information from different data sets. the first operation is joining, second operation is appending or stacking.

1) Joining tables:

Joining tables allows you to combine the information from one observation found in one table with the information that you find in another table.

→ Joining the tables allows to combine the information so that you can use it for your model as shown in figure.

Client	Item	Month
John Doe	coca-cola	January
Jackie Qi	pepsi-cola	January

Client	Region
JohnDoe	NY
JackieQi	NC

Client	Item	Month	Region
John Doe	coca-cola	January	NY
Jackie Qi	pepsi-cola	January	NC.

fig: joining two tables on the Item and Region keys.

2) Appending tables:

Appending or stacking tables is effectively adding observations from one table to another table.

→ figure shows an example of appending tables.

client	Item	Month
JohnDoe	cocal-cola	January
Jackie Qi	pepsi-cola	January

client	Item	Month.
JohnDoe	zero-cola	February
JackieQi	Maxi-cola	February

client	Item	Month.
John Doe	cocal-cola	January
Jackie Qi	pepsi-cola	January
John Doe	zero-cola	February
Jackie Qi	Maxi-cola	February

Aggregated Measures:

Data enrichment can also be done by adding calculated information to the table, such as the total no. of sales or what percentage of total stock has been sold on a certain region shown in fig.

Product class	product	sales in $	sales t-1 in $	Growth	sales by product/cls	Rank sale
A	B	X	y	(x-y)/y	Ax	Nx
sport	sport1	95	98	-3.06%	215	2
sport	sport2	120	132	-9.09%	215	1
shoes	shoes1	10	6	66.67%	10	3

fig: Growth, sales by Product class, and rank sales are example of derived aggregate measures.

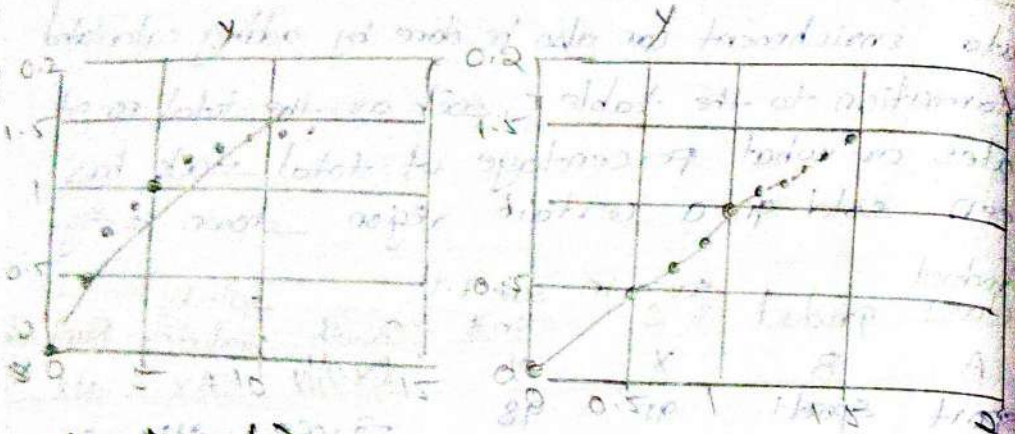
⇒ Transforming Data:

certain models require their data to be in a certain shape. Now you've clean and integrated the data.

⇒ Transforming Data:

Relationship b/w an input variable and an output variable are not always linear. Take for instance, a relationship of the form $y = ae^{bx}$.

X	1	2	3	4	5	6	7	8	9	10
log(x)	0.00	0.43	0.68	0.86	1.00	1.11	1.21	1.29	1.37	1.43
y	0.00	0.44	0.69	0.87	1.02	1.11	1.24	1.32	1.38	1.46.



• Y → linear (y)

fig: Transforming x to log x makes the relationship b/w x and y linear (right) compared with non-log x (left)

Euclidean Distance:

Euclidean distance b/w two points in a two-dimensional plane is calculated using a similar formula:

$$\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Three dimensions we get

$$\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

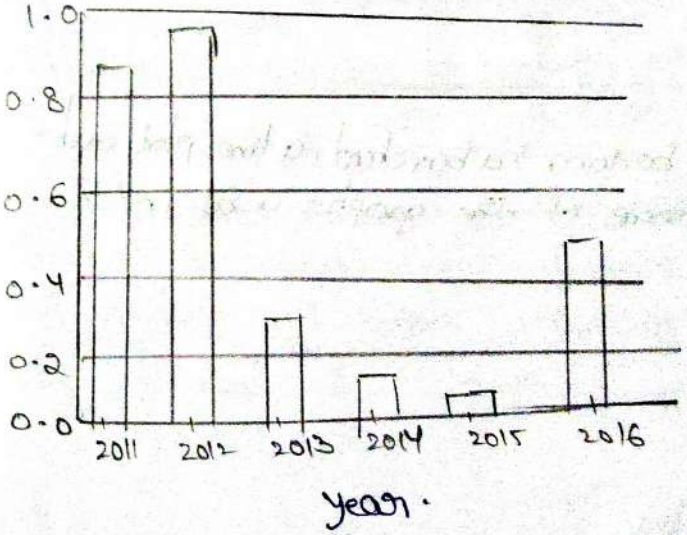
4. Exploratory Data Analysis:

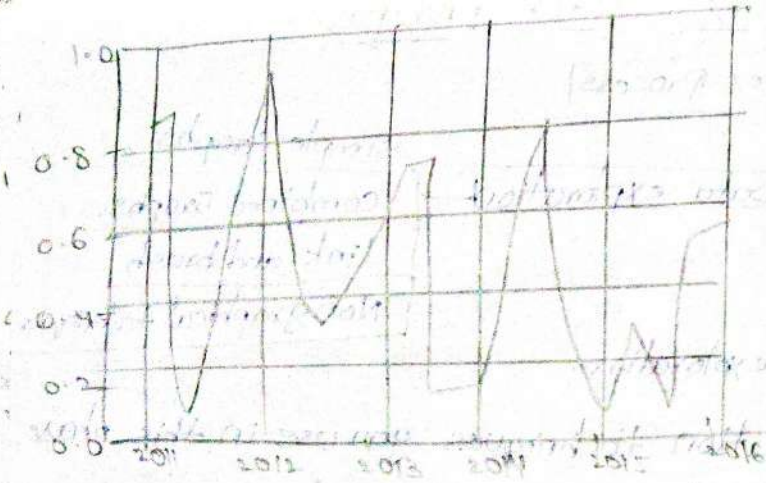
Data science Process

4. Data exploration → Simple Graphs
   - Combined Graphs
   - link and brush
   - Nongraphical techniques

fig: Data exploration.

The visualization techniques you use in this phase range from simple line graphs or histograms, as shown in figure to more complex diagrams such as Sankey and network graphs. It's worth spending time on his website, though most of his examples are more useful for data Presentation than data exploration.





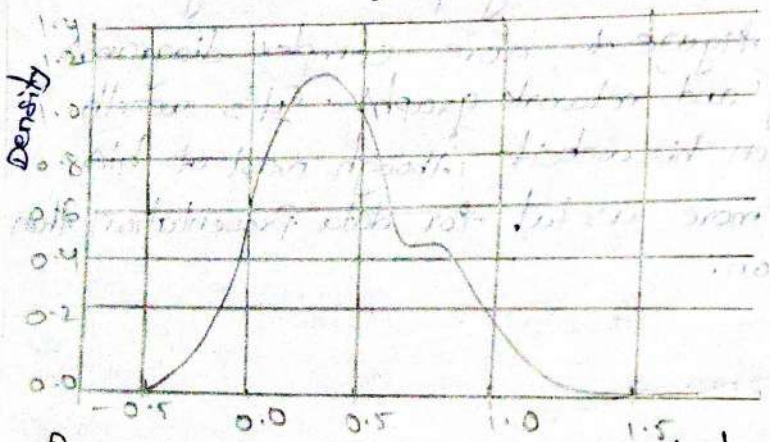


fig: from top to bottom, a barchart, a line plot and a distribution are some of the graphs used in exploratory analysis.

Two important graphs are the histogram shown in fig and the boxplot shown in figure below.

In a histogram variable is cut into discrete categories and the number of occurences in each category are summed up and shown in fig graph.

It can show maximum, minimum, median and other characterizing measures at the same time.

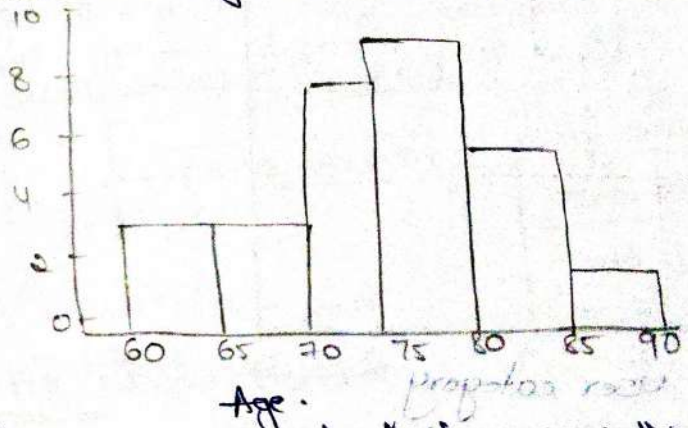


fig: Example of histogram: the no. of people in the age-groups of 5year intervals.

The techniques we described in this phase are mainly visual, but in practice they are certainly not limited to visualization techniques. Tabulation, clustering and other modeling techniques can also be a part of exploratory analysis.

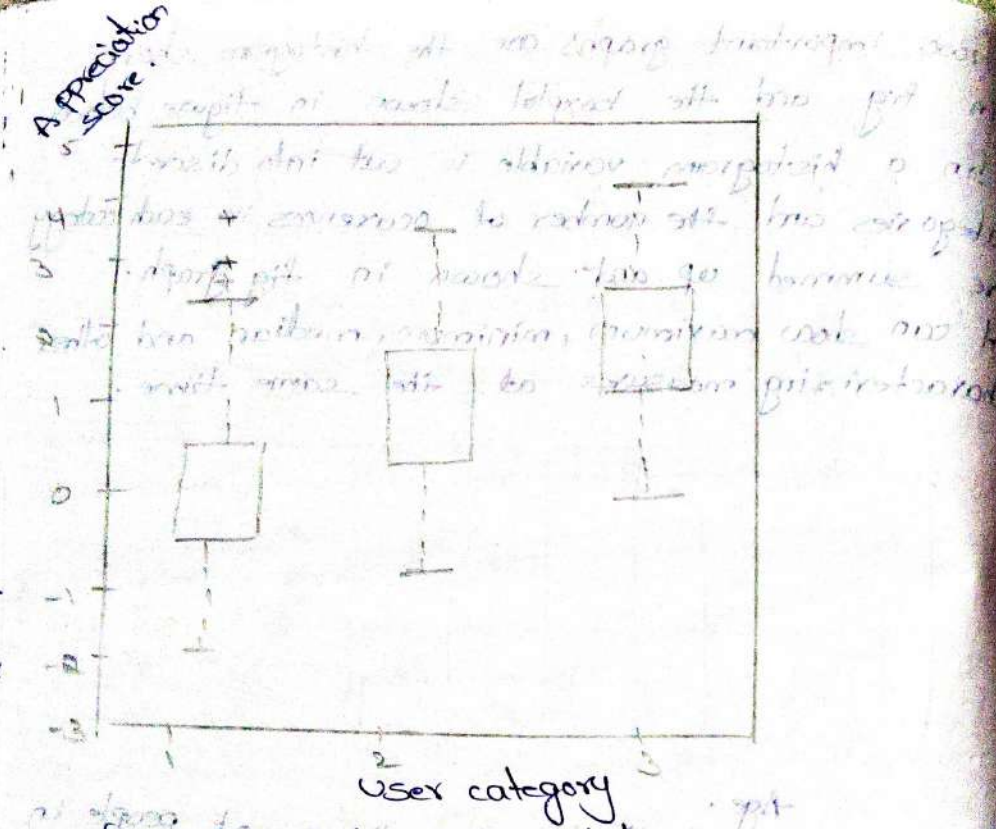


fig: Example of boxplot.

Each user category has a distribution of the appreciation each has for a certain picture on a photography website.

5) Building the models

With clean data in space and a good understanding of the content you are ready to build models with the goal of making better predictions, classifying objects, organizing an understanding of the system that you are modeling. The below diagram shows the component of model building.

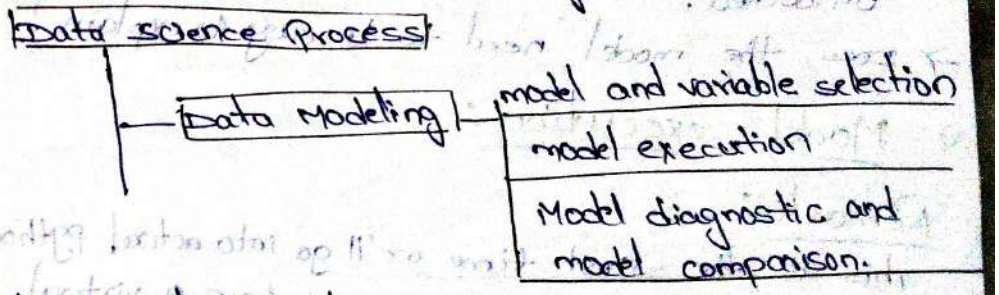


fig: Data Modeling:

1) Model and variable selection:

You will need to select the variables you want to include in your model and a modeling technique.

→ your findings from the exploratory analysis should already give a fair idea of what variables will help you construct a good model.

Other factors of model and variable selection

→ Must the model be moved to a Production environment and if so, would it be easy to implement?

→ How difficult is the maintainance on the model? how long will it remain relevant if left untouched?

→ Does the model need to be easy to explain?

2) Model execution:

Remark:

This is the first time we'll go into actual python code execution so make sure you have a virtual env up and running.

→ Likely most programming languages, such as Python, already have libraries such as statsmodels or scikit-learn. These Packages use several of the most popular techniques.

Listing 2.1 Executing a linear Prediction model on semi random data

```
import statsmodel.api as sm                    | Import required
import numpy as np                             | python modules
Predictors = np.random.random(1000).reshape(500,2).
target = Predictors.dot(np.array([0.4,0.6])) + np.random.
lmRegmodel = sm.OLS(target, Prediction)        | random(500).
result = lmRegModel.fit()                      | fits linear
                                               | regression
result.summary()  ← shows model fit statistics | on data.
```

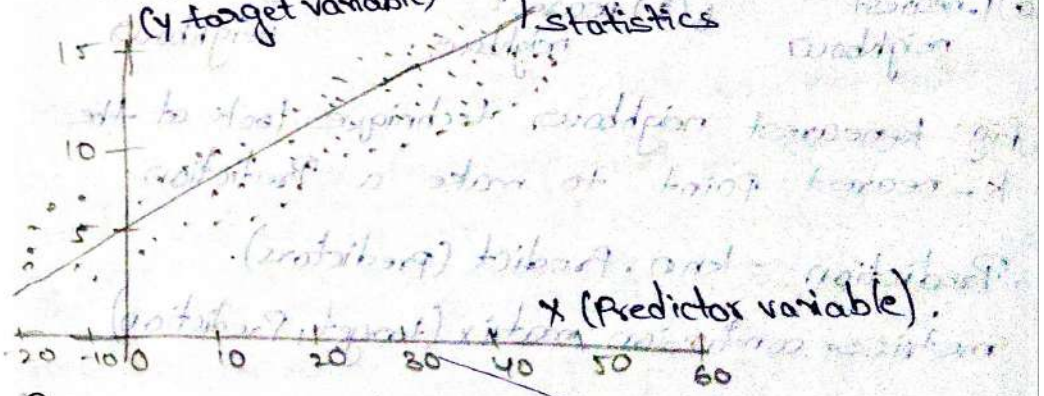


Fig: Linear regression tries to fit a line while minimizing the distance to each point.

Let's ignore most of the output we got here and focus on the most important Parts.

→ Model fit

→ Predictor variables have a coefficient.

→ Predictor significance.

As shown in below figure, k-nearest neighbours looks at labeled points nearby an unlabeled point and based on this makes a Prediction of what the label should be.

a) 1-nearest neighbour b) (a) nearest neighbours. (c) 3-nearest neighbours.

fig: k-nearest neighbours techniques look at the k-nearest point to make a Prediction.

Prediction = knn.Predict(Predictors).

metrics.confusion_matrix(target, Prediction).

```
In [45]: metrics.confusion_matrix(target, Prediction)

Out[45]: array([[17, 33,  0],
                [ 9, 405,  1],
                [ 0, 30,  5]])
```

0 1 2 ← Predicted value.

0 1 2 ← Actual value.

number of correctly Predicted cases.

3) Model diagnostics and Model comparison:

you will be building multiple models from which you then choose the best one based on multiple criteria. working with a holdout sample helps you pick the best-performing model.

→ The error measure used in example is the mean square error.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

fig: formula for mean square error.

MSE is a simple measure: check for every prediction how far it was from the truth, square the error, and add up the error of every prediction.

The below figure compares the performance of two models to predict the order size from the Price. the first model is size = 23 * Price and the second model is size = 10.

	n	size	Price	Predicted model-1	Predicted model-2	error model-1	error model-2
	1	10	3				
	2	15	5				
	3	18	6				
80% train	4	14	5				
	...	....					
	800	9	3				
	801	12	4	12	10	0	2
	802	13	4	12	10	1	3
	...						
	999	21	7	21	10	0	11
20% test	1000	10	4	12	10	-2	0
					Total	5861	110225.

Fig : A holdout sample helps you compare models and ensures that you can generalize results to data the model has not yet seen.

→ Many models make strong assumptions, such as independence of the i/ps and you have to verify that these assumptions are indeed met. This is called model diagnostics.

⇒ Presenting findings and building applications on top of them :

After you are successfully analysed the data and built a well-performing model, you are ready to present your findings to the world.

→ this is exciting part; all your hours at hood work have paid off and you can explain what you found to the stakeholders.

Data science Process

1. Setting the research goal.
2. Retrieving data
3. Data preparation
4. Data exploration
5. Data modeling
6. Presentation and automation

Presenting data
Automating data analysis.

fig : Presentation and automation.

Sometimes people get so excited about your work that you will need to repeat it over and over again because they value the Predictions of your models or the insights that you Produced. Sometimes it's sufficient that you implement only the model scoring; other times you might build an application that automatically updates reports. Excel spreadsheets, or power point Presentations. The last stage of the data science Process is where your soft skills will be most useful, and yes, they are extremely important. we recommend you find dedicated books and other information on the subject work through them. If you have done this right, you now have a working model and satisfied stakeholders.

UNIT - II:

Syllabus:

Applications of Machine learning in Data science, role of ML in DS, Python tools like sklearn, Modelling Process for feature engineering, Model selection, validation and Prediction, types of ML, semi-supervised learning. Handling large data: Problems and general techniques for handling large data, Programming tips for dealing large data, case studies on DS Projects for Predicting malicious URLs for building recommender syste

7) Applications of Machine learning in data science:

Regression and classification are primary importance to a data scientist. To acheive these goals, one of the main tools of a data science used in machine learning. This uses of regression and automatic classification are wide ranging, such as following.

* finding oil fields, gold mines, or archeological sites based on existing sites.
* finding place names or persons in text (classification).
* Identifying people based on pictures or voice recordings. (classification).
* Recognizing birds based on their whistle. (classification).

1. Although the following paper is often cited as the source of this quote, it's not present in a 1967 reprint of that paper. The authors were unable to verify or find the exact source of this quote. See Author L. samuel, "Some studies in ML using the Game of checkers".

* Identifying portable customers (regression & classification).
* Proactively Identifying car parts that are likely to fail. (regression).
* Identifying tumors and diseases. (classi)
* Predicting the amount of money a person will spend on product x (regression).
* Predicting the number of eruptions of a volcano in a period (regression).
* Predicting your company's yearly revenue. (Regression).
* Predicting which team will win the champions league in soccer (classification).

Ocasionally data scientists build a model that Provide insight to the underlying processes of phenomenon. When the goal of a model is not Prediction but interpretation, it's called root cause analysis. Here few examples:

* Discovering what causes diabetes.
* Determining the causes of traffic jams

a) where ml is used in the datascience Process

Although, ml is mainly linked to the datamodeling step of the data science Process it can be used at almost every step. To refresh your memory from previous chapters, the data science Process is shown in fig: 3.1.

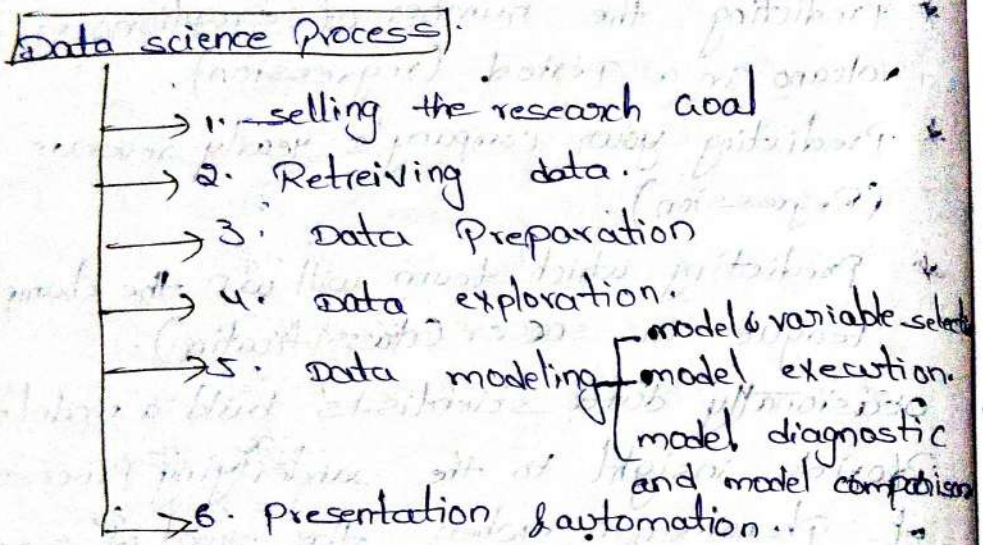


fig: the data science Process.

The data modeling phase can't start until you have qualitative raw data you can understand. But Prior to the data Preparation phase can benefit from the use of ML.

ML is also useful when exploring data. Alg can root out underlying Patterns in data where they did be difficult to find with only charts.

② Python tools like sklearn:

Python has an overwhelming number of Packages that can be used in ML setting. The python machine learning ecosystem can be divided into three main types of Packages as shown in figure.

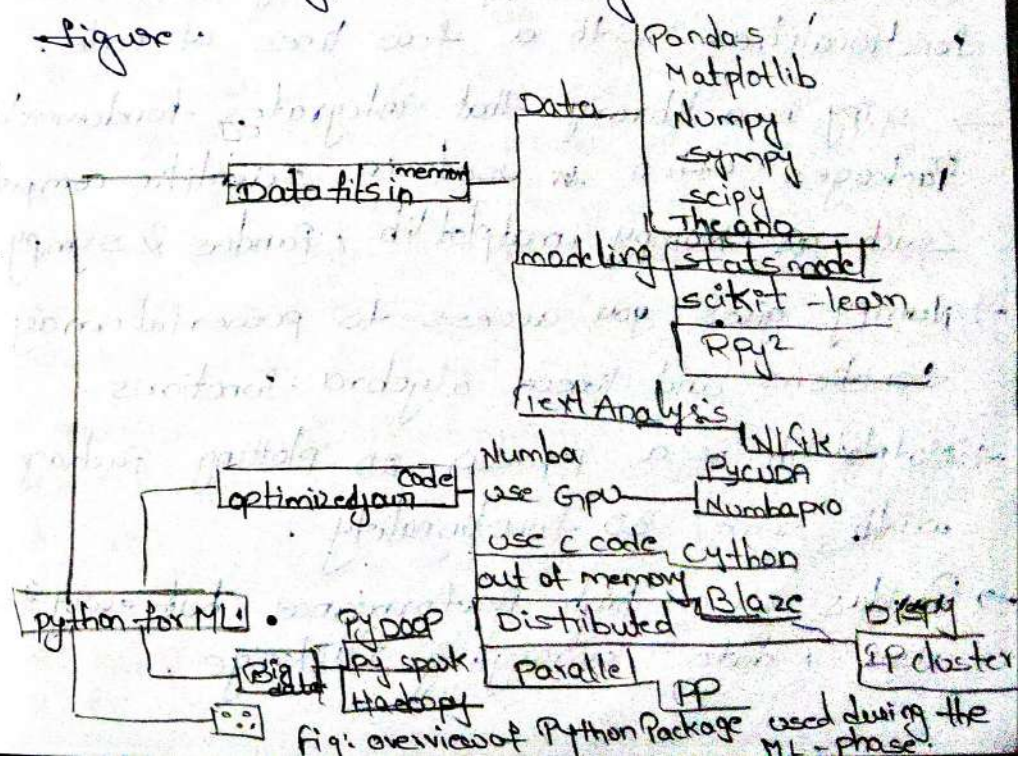


fig: overview of Python Package used during the ML-phase.

the first type of packages as shown in fig(1) mainly used in simple tasks when data fits into memory. the second type used to optimize your code when you have finished prototyping and run into speed or memory issues. The third type is specific to using python with big data technologies:

→ Packages for working with data in Memory: when Prototying the following Packages can get you started by providing advanced functionalities with a few lines of code:

→ scipy is a library that integrates fundamental Packages often used in scientific compt. such as Nompy, matplotlib, Pandas & sympy.

→ Nempy gives you access to powerful array functions and linear algebra functions.

→ Matplotlib is a popular 2D-plotting Packages with some 3D functionality.

→ Pandas is a high performance, but easy to use, data wrangling Package.

→ sympy is a Package used for symbolic mathematics and computer algebra.

→ stats Models is a Package for statistical methods and algorithms.

→ scikit-learn is a library filled with ML algorithms.

→ RPy2 allows you to call R functions with in Python. R is a Popular open source statistic Program.

→ NLTK (Natural language toolkit) is a python toolkit with a focus on text analytics.

Optimizing operations:

once your applications moves into production, the libraries listed here can help you deliver the speed you need. sometimes this involves connecting to big data infrastructure such as Hadoop and spark.

* Numba and Numbapro.
* PyCUDA
* Cython or C for Python.
* Blaze
* Dispy and Ipcluster.
* PP
* Pydoop & Hadoopy
* Pyspark.

(3) Modeling Process of feature engineering:

The modeling phase consists of four steps:

1. feature Engineering and model selection.
2. Training the model.
3. Model validation and selection.
4. Applying the trained model to unseen data.

Before you find a good model, you will Probably iterate among the first 3 steps. The last step is not always present because sometimes the goal is not prediction but explanation.

It is possible to chain or combine multiple techniques. when you can chain multiple models, the output of the first becomes an input for the second model. when you combine multiple models you train them independently and combine their results. the last technique is also known as ensemble learning.

A model consists of constructs of information called features or predictors and a target or response variable. your model goal is to Predict the target variable for example tomorrow's high temperature.

(a) Feature engineering and selecting Model:

With engineering features, you must come up with and create possible predictors for the model. this is one of the most important steps in the Process, because a model recombines these features to achieve its Predictions. often you may need to consults an expert or the appropriate literature to come up with meaningful features.

certain features are the variables you get from a data set, as it case with the Provided data sets in our excercise and most school excercises. In several Projects we had to brig together more than 20 different data source. before we had the raw data we required.

when the initial feature are created, a model can be trained to the data.

⑦ Training the model:

with the right predictors in place and a modeling technique in mind, you can progress to model training. In this phase you present to your model data from which it can learn.

The most common modeling technique having industry-ready implementations in almost every programing language, including Python. These enable you to train your model by executing a few lines of code.

once a model trained, it is time to test whether it can be extrapolated to reality model validation.

⑧ Validating a Model:

Data science has many modeling techniques, and the question is which one is the right one to use. A good model has two properties: it good predictive power and it generalizes well to data it hasn't seen. To acheive this you define an error measure and validation strategy. Two common error measures in Machine learning are the classification error rate for classification Problems and the mean squared error for regression Problems. the classification error rate is the percentage of observation in the test data, set that your model mislabeled; lower is better.

Many validation strategies exist, including the following commonones:-

1. Dividing your data into a training set with X% of the observation and keeping the rest as a holdout data set.
2. k-fold cross validation.
3. Leave-1 out.

Another popular term in machine learning is regularizations. once you have constructed a good model, you can use it predict the features.

(d) Predicting new observations:

If you have implemented the first three steps successfully, you have now a performant model that generalizes to unseen data. the process of applying your model to new data is called model scoring. In fact, model scoring is something you implicitly did during validation, only now you don't know the correct outcome.

Model scoring involves two steps:
first, you prepare a data set that has feature exactly as defined by your model. this boils down to repeating the data preparations you did in step one of the modeling process but for a new data set. then you apply the model on this new data set, and this results in a prediction.

(4) Types of Machine learning:

Broad speaking, we can divide the different approaches to machine learning by the amount of human efforts that's required to coordinate them and how they use labeled data. data with a category or real-value number assigned to that presents the outcome of previous observations.

→ supervised learning technique attempts to discern results and learn by trying to find patterns in labeled data set. human interaction is required to label the data.

→ unsupervised learning technique don't rely on labeled data and attempt to find patterns in a data set without human interactions.

→ semi-supervised learning technique that need labeled data, therefore human interaction, to find patterns in the data set but they can still progress toward a result and learn even if passed unlabelled data as well.

Supervised learning:

As stated before, supervised learning is a technique that can only be applied on labeled data. An example implementation of this could be discerning digits from images. let's dive into a case study on number recognition.

Case study: Discerning digits from Images:

one of the many common approaches on the web to stopping computers from hacking into user accounts is the captcha check - a picture of text and numbers that one human user must decipher and enter into a form field before sending the from back to the web server.

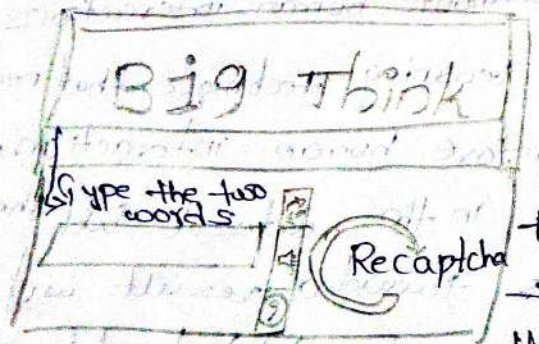


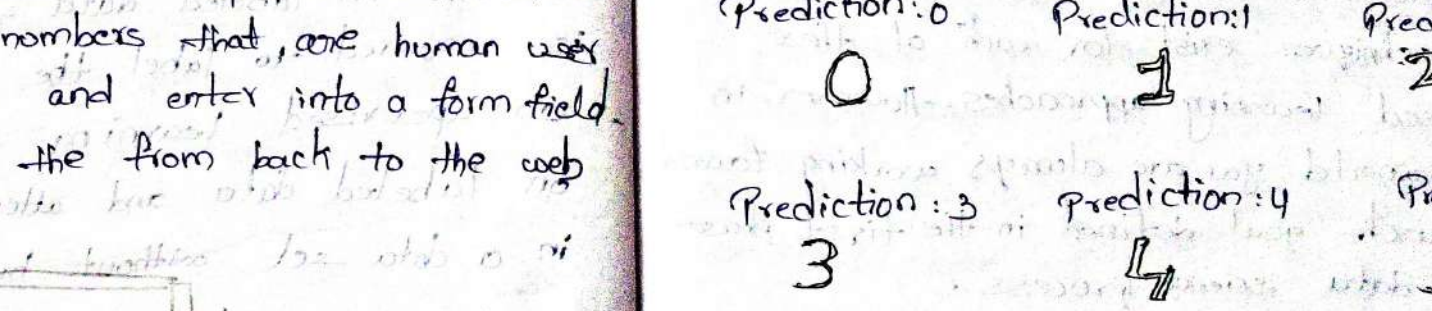
fig: A simple captcha control can be used to prevent automated spam being sent through an online web form.

With the help of the Naive Bayes classifier, a simple yet powerful algorithm to categorize observations into classes that explained in more details in the sidebar. you can regognize digits from textual images.

Our search goal is to let a computer recognize images of numbers.

The data we will be working on the MNIST data set, which is often used in the data science literature for teaching and benchmarking.

Prediction: 0 Prediction: 1 Prediction: 8.



Prediction: 3 Prediction: 4 Prediction: 3.

Fig: For each blurry image a number is Predicted: only the number 9 is misinterpreted as 8. then an ambigious number is predicted to be 3 but it could as well be 5; even to human eyes this is not clear.

Onsupervised learning:

It's generally true that most large data sets don't have labels on their data, so unless you sort through it all and give it labels.

* we can study the distribution of the data and infer new truths about the data in different parts of the distribution.
* we can study the structure and values in the data and infer new, more meaningful data and structure from it.

Many techniques exist for each of these unsupervised learning approaches. However, in the real-world you are always working toward the research goal defined in the first phase of the data science process.

Discerning a simplified latent structure from your data:

Not every thing can be measured. when you meet someone for the first time you might try to guess whether they like you based on their behaviour and how they respond.

the first type variable are known as observable variables and second type are known as latent variable. In our example, the emotional state of your new friend is a latent variable. It definitely influences by their judgement of you but its value is not clear...

* Latent variable can substitute for several existing variables already in the data set.
* Because latent variables are designed to target toward the defined research goal you lose ~~little~~ little key information by using them.

Case study: finding latent variables in a wi quality Data set:

In this short case study, you will use a technique know as Principal component Anal to find a latent variable in a data se that describes the quality of wine.

lets look at the main component at this example:

→ Data set: the university of california.

→ Principal component Analysis: A technique to find the latent variables in your data set while retraining as much information as follows:

→ Scikit-learn: we use this library because it already implements a PCA for us and is a easy to generate the scree plot:

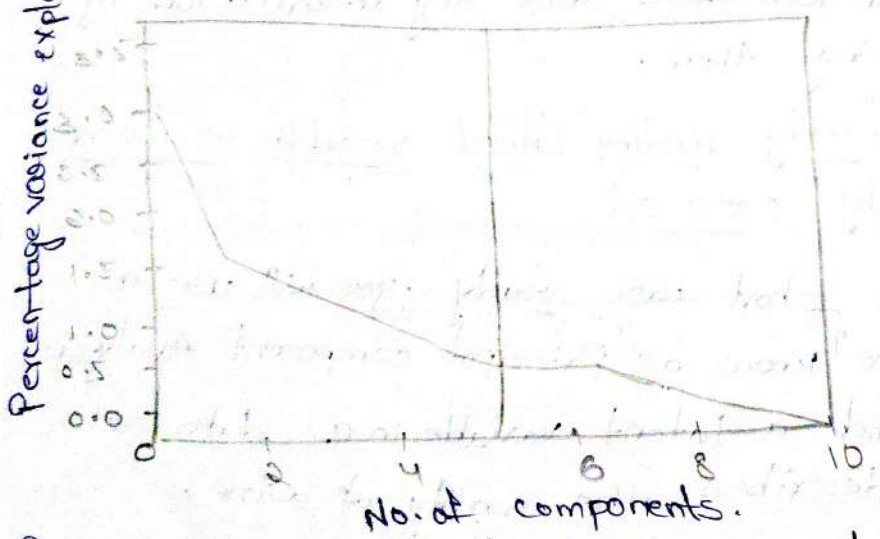


fig: PCA scree plot showing the marginal amount of information at every new variable PCA can crete.

The first variable can explain approximately 25% of the variance in the data, the second variable accounts for another 17%, the third approximately 13% and so on.

The two variables will capture aproximately 17% more or 42% total and so on.

Integrating the new variables:

With the initial decision made to reduce the data set from 11 original variables to 5 latent variables.

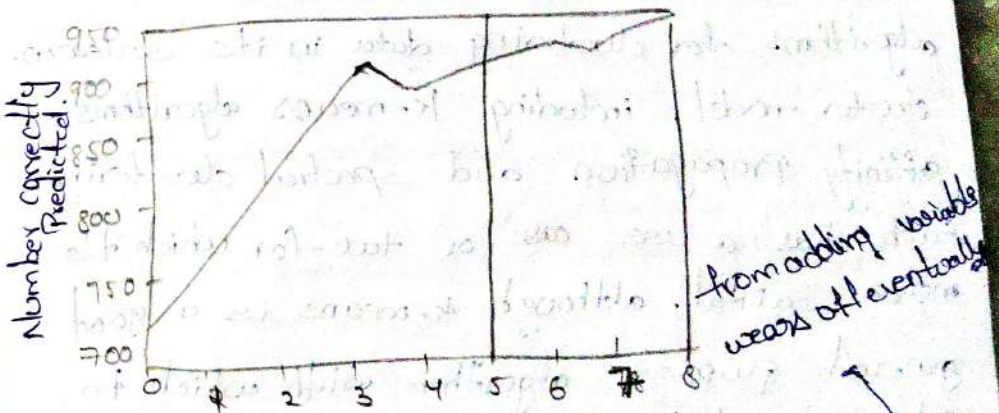


fig: the results plot shows that adding more latent variables to a model (x-axis) greatly increases Predictive power (y-axis) up to a point but then tails off. the gain in Predictive power

1 Grouping similar observations to gain insight from the distribution of your data:

The general technique we are describing here is known as clustering. In this process, we attempt to divide our data set into observation subsets, or clusters, where in observations should be similar to those in same cluster but differ greatly from the observation in other clusters as shown in figure.

Scikit-learn implements several common algorithms for clustering data in its sklearn.cluster model, including k-means algorithms, affinity Propagation and spectral clustering. Each has a use case or two for which it's more suited, although k-means is a good general purpose algorithm with which to get started.

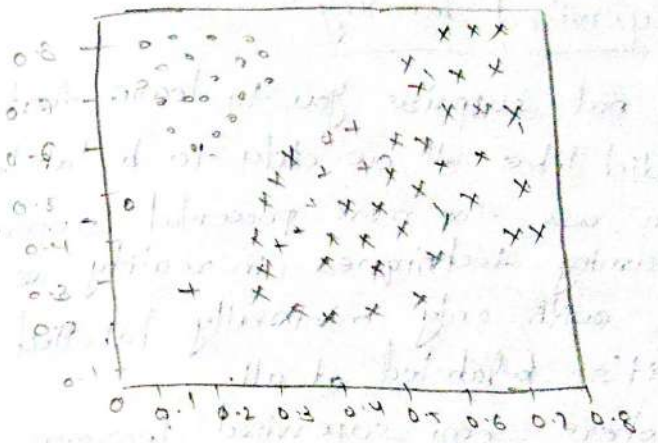


Fig: the goal of clustering is to divide a data set into "sufficiently distinct" subsets. In this plot for instance, the observation have been divided into three clusters.

this figure shows that even without using a label did find clusters that are similar to the official iris classification with a result of 134 (50+48+36) correct classification out of 150.

		C
cluster	target	
0	0	50
1	1	48
	2	14
2	1	2
	2	36

Fig: output of Iris classification

(5) Semi-supervised learning:

It should not surprise you to learn that while we did like all our data to be labeled so we can use the more powerful supervised machine learning techniques, in reality we often start with only minimally labelled data, if it's labeled at all.

This is where semi supervised learning techniques comes in hybrids of the two approaches we have already seen.

Take for example the plot in figure. In this case, the data has only two labeled observations, normally this is too few to make valid predictions.

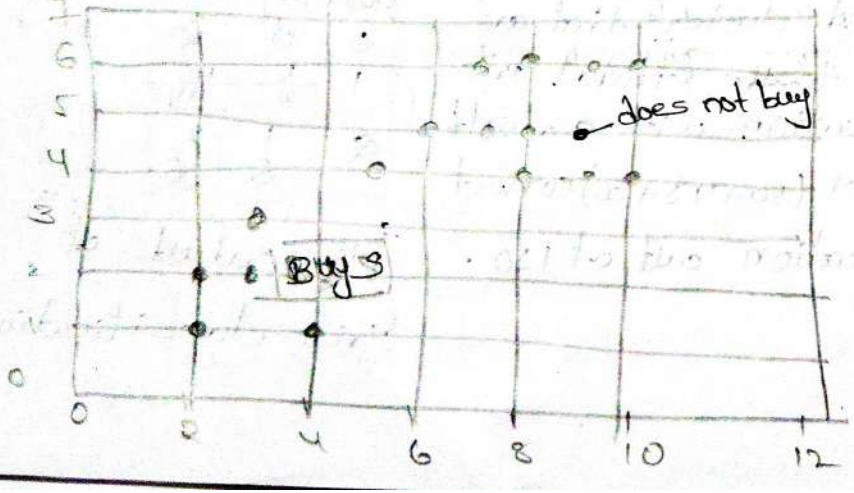


fig: This plot has only two label observations too few for supervised observations, but enough to start with an unsupervised a semi supervised approach.

A common semi-supervised learning technique is label propagation. In this technique, you start with a labeled data set and give same label to similar data points.

one special approach to semi-supervised learning worth mentioning here is active learning.

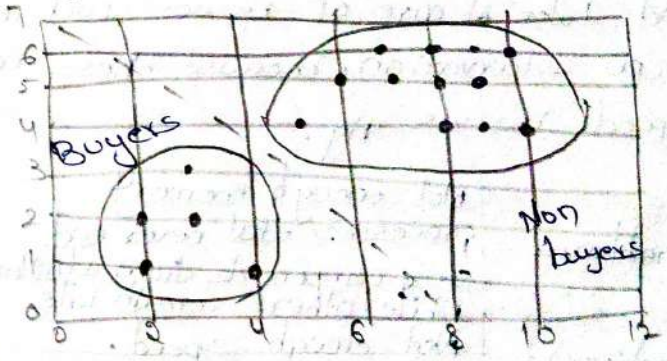


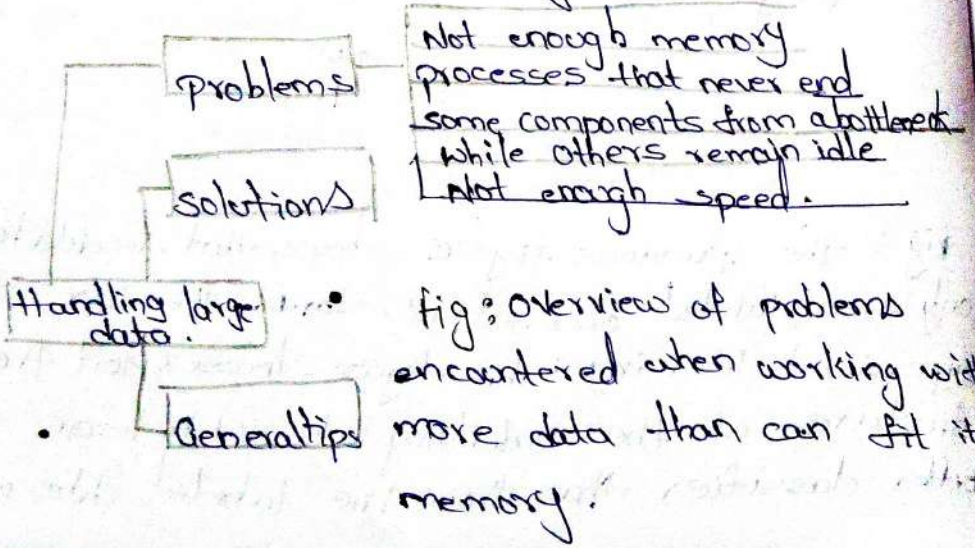
fig: The previous figure shows that the data has only two labeled observations, for too few for supervised learning. this figure shows exploit the structure of the underlying data set to learn better classifier than from the labeled data only.

Handling large data:

1) Problems and general techniques for handling large data:

The Problems techniques for handling large data

The large volume of data poses new challenges, such as overloaded memory and algorithms that never stop running. It forces you to adapt and expand your repetoire of techniques. But even when you can perform your analysis you should take care of issues such as I/O and CPU starvation, because these can cause speed issues. fig.



fig; Overview of problems encountered when working with more data than can fit in memory.

A computer only has a limited amount of RAM. When you try to squeeze more data into this memory than actually fits, the os will start swapping out memory blocks to disks which is far less efficient than having it all in memory.

You may need to deal with another limited resources: time. Although a computer may think you live for millions of years; in reality you won't. Certain algorithms don't take time into account, other algorithms can't end in a reasonal amount of time when they need to process only a few mega bytes of data.

A third thing you will observe when dealing with large data sets is that components of your computer can start to form a bottle neck while living other systems idle. Certain addressed with the instruction of solid state drives, but SSDs are still much more expensive than the slower and more widespread hard disk drive (HDD). technology.

General techniques for handling large data

Never ending algorithms, out-of memory errors and speed issues are the most common challenges you face when working with large data.

the solutions can be divided into three categories: using the correct algorithms, choosing the right data structure and using the right tools as shown in figure.

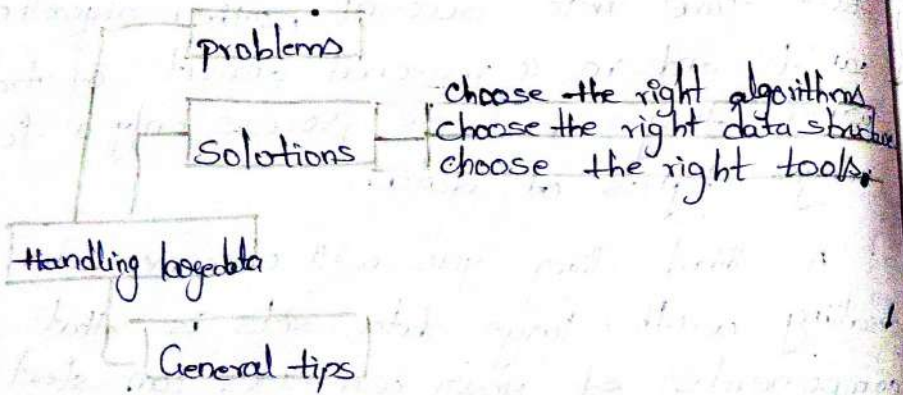


fig: over view of solutions for handling large data.

no clear one-to-one mapping exists b/w the Problems and solutions because many solutions address both lack of memory and computational Performance.

(a) choosing the right algorithms:

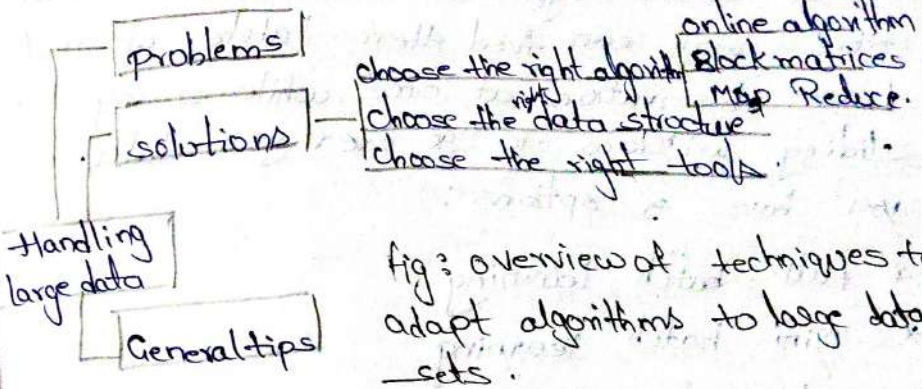


fig: overview of techniques to adapt algorithms to large data sets.

Online learning algorithms:

several but not all, machine learning algorithms can be trained using one observation at a time instead of taking all the data into memory. upon a arrival of a new data point, the model is trained and the observations can be forgotten; its effect is now incorporated into the models Parameters.

this "use and forget" way of working is the perfect solution for the memory problem as a single observation is unlikely to ever be big enough to fill up the memory of a modern day computer.

→ Most online algorithms can handle mini-batches this way can feed them batches of 10 to 1,000 observations at once while using sliding window to go over your data.

you have 3 options:

* full batch learning
* Mini batch learning
* online learning.

Diving large data Matrix into small ones

where as in the previous chapter we barely needed to deal with how exactly the algorithm estimates parameters, diving into this might sometimes help. By cutting the large data table into small matrices for instance, we can still do a linear regression.

the logic behind this matrix splitting and how a linear regression will be calculated with matrices can be found in the sidebar.

Block matrices and matrix formula of linear regression coefficient estimations:

certain algorithms can be translated into algorithms that use blocks of matrices instead of full matrices.

$$A+B = \begin{bmatrix} a_{1,1} & \cdots & a_{1,m} \\ \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,m} \end{bmatrix} + \begin{bmatrix} b_{1,1} & \cdots & b_{1,m} \\ \vdots & & \vdots \\ b_{n,1} & \cdots & b_{n,m} \end{bmatrix}$$

$$= \begin{bmatrix} a_{1,1} & \cdots & a_{1,m} \\ \vdots & & \vdots \\ a_{j,1} & \cdots & a_{j,m} \\ \hline a_{j+1,1} & \cdots & a_{j+1,m} \\ \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,m} \end{bmatrix} + \begin{bmatrix} b_{1,1} & \cdots & b_{1,m} \\ \vdots & & \vdots \\ b_{j,1} & \cdots & b_{j,m} \\ \hline b_{j+1,1} & \cdots & b_{j+1,m} \\ \vdots & & \vdots \\ b_{n,1} & \cdots & b_{n,m} \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$$

fig: Block matrices can be used to calculate the sum of the matrices A and B.

Map Reduce:

Map Reduce algorithms are easy to understand with analogy; Imagine that you were asked to count all the votes for the national elections.

Map Reduce follow a similar process to the second way of working, the first map values to a key and then do an aggregation on that key during the reduce phase. Have look at the following listening, pseudo code to get better feeling for this.

⑥ Choosing the right data structure:

Algorithms can make or break your program. but the way you store your data is of equal importance. Data structure have different storage requirements, but also influence the performance of CRUD (create, read, update and delete) and other operations on the data set.

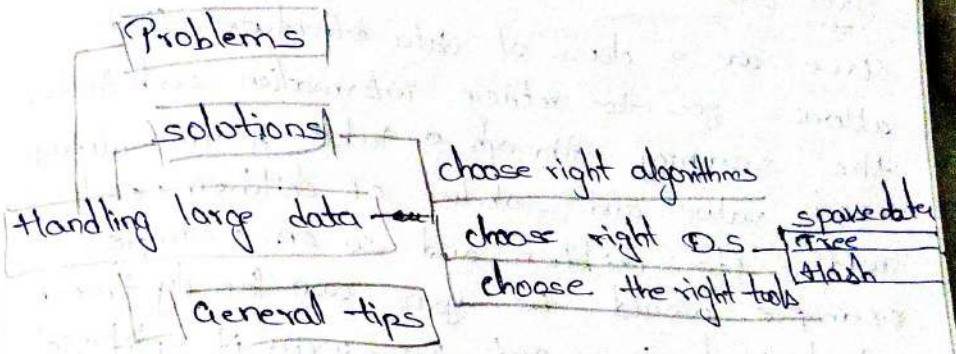


fig: overview of data structure often applied in data science when working with large data.

Sparse Data:

A sparse data set contains relatively little information compared to its entries

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

fig: Example of sparse matrix: almost everything is 0; other values are the exception in a sparse matrix.

Tree structures:

Tree are a class of data structure that allows you to retrieve information such faster the scanning through a table. A tree always root, value and subtree of children, each with its children, and so on. simple example would be your own family tree or a biological tree and the way it splits into branches, twigs and leaves.

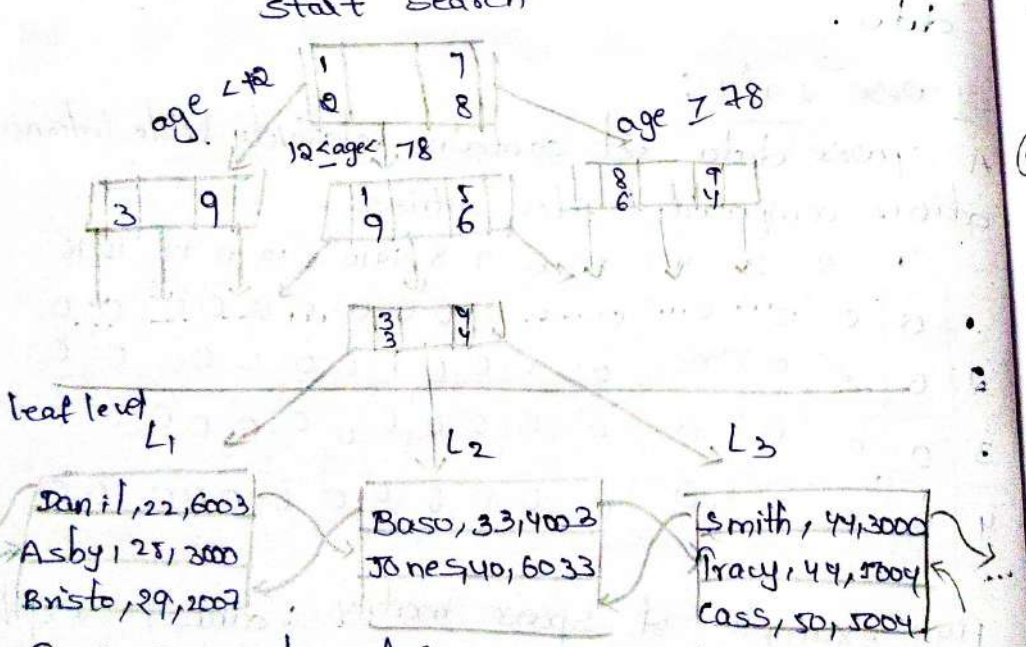


Fig: Example of tree data structure: decision rules such as age categories can be used to quick locate a person in a family tree.

Hash Tables:

Hash tables are data structures that calculate a key for every value in your data and the put keys in a bucket. this way you can quickly retrieve the information by looking in the right bucket when you encounter the data. Dictionaries in Python are a hash table implementation and they are close to relative of key value stores. you will encounter them in the last example of chapter when you build a reccommender system within the database.

(c) selecting the right tools:

with the right class of algorithm and data structure in place, it's time to choose the right tools for the job. the right tools can be a Python library or at least a tool that's controlled from python as shown in fig:

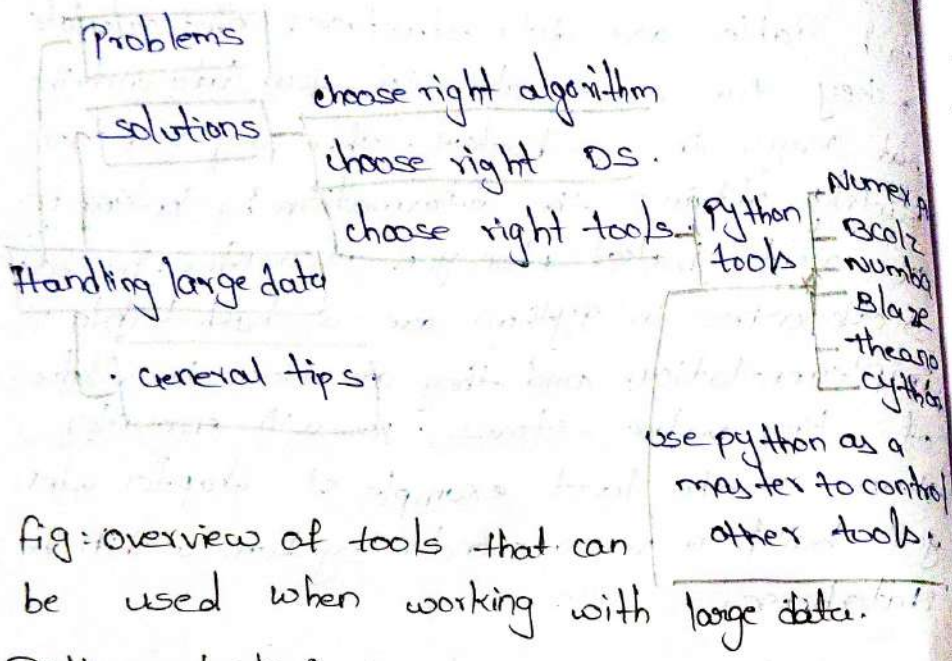


fig: overview of tools that can be used when working with large data.

Python tools:

Python has a number of libraries that can help you deal with large data. the range from smarter data structures over code optimize to just in time compilers. the following is a list of libraries we like to use when confronted with large data.

→ Cython
→ Numexpr
→ Numba
→ Bcolz
→ Blaze
→ Theano
→ Dask.

these libraries are mostly about using python itself for data processing.

use python as a Master to control other tools.

Most software and tool producers support a python interface to their software. this enables you to tap into specialized pieces of software with the ease and productivity that combines with python. this way python sets itself apart from other popular data science language such as R and SAS. you should take advantage of this luxury and exploit the power of specialized tool to the fullest extent possible.

② Programming tips for dealing with large data

The tricks that work in a general programming context still apply for data science. Several might be worded slightly differently, but the principles are essentially the same for all programmers.

You can divide the general tricks into 3 parts, as shown in fig:

→ Don't reinvent the wheel.

use tools and libraries developed by others.

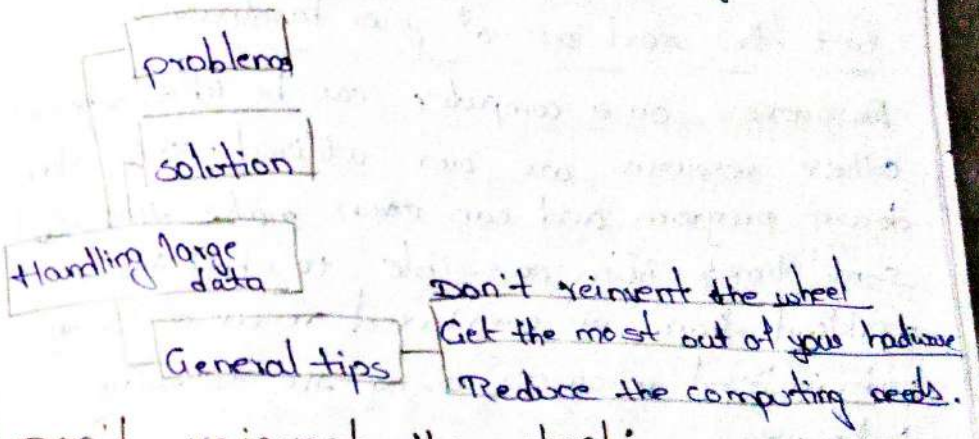
→ Get the most out of your hardware:

your machine is never used to its full potential with simple adaptions you can make it work harder.

→ Reduce computing need.

slim down your memory and processing need as much as possible.

fig: overview of general programming best practices when working with large data:



Don't reinvent the wheel:

"Don't repeate anyone" is probably even better than "don't repeat yourself". Add value with your action: make sure that they matter. solving a problem that has already been solved is a waste of time. As a data scientist, you have two large rules that can help you deal with large data and make you much more productive, to boot:

* Exploit the power of database:
* Use optimizing libraries.

Get the most out of your hardware:

Resources on a computer can be idle, whereas other resources are over-utilized. This slow down program and can even make them fail. Some times it's possible to shift the workload from an overtaxed resource to an underutilized resources using the following techniques.

* Feed the cpu compressed data.
* Make use of the GPU
* Use multiple threads.

Reducing your computing needs:

"working smart + hard = acheivements". This is also applies to the programs to write. The best way to avoid having large data, Problems is done removing as much of the work as possible up front and letting the computer work only on the part that can't be skipped.

the following list contains methods to help you acheive this:

* Profile your code and remediate slow pieces of code.
* Use compiled code whenever possible, certainly when loops are involved.
* otherwise, compile the code yourself.
* Avoid pulling data into memory.
* Use generators data into memory.
* use generators to avoid intermediate data storage.
* use as little data as possible.
* use your math skills to simplify calculations as much as possible.

(3) Case study on DS Projects for Predicting malicious URLs:

case study 1: Predicting malicious URLs:

The Internet is probably one of the greatest inventions of modren etimes. It has boosted in humanity development, but not every

one uses this great invention with honorable intensions. Many companys try to protect us from fraud by detecting malicious websites for us.

what will use

→ Data

→ The scikit learn library.

step-1: Defining the research goal:

the goal of our project is to detect whether certain URLs can be trusted or not. Because the data is so large we aim to do this is in a memory-friendly way. In the next step we will first look at what happens if we don't concern ourselves with memory (RAM issues).

step-2: Acquiring the URL Data:

start by downloading the data from http://sysnet.ucsd.edu/projects/url/ # datasets and place it in a folder. choose the data in SVM light format. SVMlight a text-based format with one observation per row.

(to save space, it leaves out the zeros.

Memory Error (Traceback (most recent call last)
<ipython-input-532-d1960508826e> <module> ()
5 print "there is %d files" % len(files)
6 x,y = load_svmlight_file(files[0], n_feature
---> 7 X.todense(). =3500000)

fig: memory error when trying to take a large data set into memory.

step-3: Data exploration:

to see if we can even apply our first trick we need to find out whether the data does indeed contain lots of zeros. we can check this with the following piece of code.

print "number of non-zero entries %.2.6f" % float((X.nnz)/(float(X.shape[0]).float(X.shape[1]))).

output:

number of non-zero entries 0.000033.

data that contains little information compared to zero is called sparse data. this can be saved more compactly if you store the data as [(0,0,1),(4,4,1)] instead of [[1,0,0,0,0], [0,0,0,0,0], [0,0,0,0,0], [0,0,0,0,0], [0,0,0,0,1]].

step-4: Model building:

Now that we are aware of the dimensions of our data, we can apply the same two tricks and add the third in the following listing let's find those harmful websites.

Now let's look at a second application of our techniques; this time you will build a recommender system inside a database.

	precision	recall	f1-score	support
-1	0.97	0.99	0.98	14045
1	0.97	0.94	0.96	5955
avg/total	0.97	0.97	0.97	20000

Fig: classification problem: can a website be

(4) For building Recommender system:

The reality most of data you work with it stored in a relational database, but most database are not suitable for data mining. But as shown in fig example, it possible to adopt our techniques so you can do a large part of the analysis inside the database itself.

Tools and Techniques needed:

→ MySQL database
→ MySQL database connection python library.

step-1: Research question:

Let's say your working in a video store and the manager ask you if its possible to use the information on what movies people rent to Predict what other movies they might like.

step-2: Data Preparation:

The data your boss collected is shown in table we will create this data ourselves for the sake of demonstration.

Customer	movie 1	movie 2	movie 3	...	movie 32
Jack Dani	1	0	0		1
Wihelmson	1	1	0		1
....					
Jane Dane	0	0	1		0
Xi Liu	0	0	0		1
Error Nazo	1	1	0		1
....					

fig: Excerpt from the client database and the movies customers rented.

step-3: model building:

To use hamming distance in the database we need to define it as a function.

step-4: Presentation and automation:

Now that we have it all set up, our application needs to perform two steps when confronted with a given customer.

* look for similar customers
* suggest movies the customer has yet to see based on what he or she has already viewed and the viewing history of the similar customers.

UNIT - III:

No SQL movement for handling Bigdata.

Syllabus:

Distributing data storage and Processing with Hadoop framework, case study on risk assessment for loan sanctioning, ACID Principle of relational data ~~sets~~ bases, CAP theorem, base principle of NoSQL databases, types of NoSQL database, case study on disease diagnosis and Profiling.

(1) Distributing data storage and Processing with Hadoop framework:

New big data technologies such as Hadoop and spark make it much easier to work with and control a cluster of computers. Hadoop can scale upto thousand of computers, creating a cluster with petabytes of storage. This enables business to grasp the value of Massive amount of data available.

Hadoop: a framework for storing and processing large data sets:

Apache Hadoop is a framework that simplifies working with a cluster of computers. It aims to be all of the following things and more.

* Reliable :- By automatically creating multiple copies of the data and redeploying processing logic in case of failure.
* fault tolerant : It detects faults and applies automatic recovery.
* scalable : Data and its processing are distributed over clusters of computers (horizontal scaling)
* Portable : Installable on all kinds of H/w and operating s/ms.

The core framework is composed of a distributin file s/m, a resource manager and a system to run distibuted programs. In Practice allows it you to work with the distibuted file s/m almost as easily as with the local file s/m of your compute But in the background, that data can be scattered among thousands of servers.

The Different components of Hadoop :

At the ~~has~~ heart of Hadoop we find
* A distributed file s/ms (HDFS)
* A method to execute programs on a massive scale (MAP Reduce).
* A s/m to manage the cluster resource (YARN)

on top of that, an ecos/m of applications arose the below fig, such as the databases Hive and HBase and frameworks for ML such as Mahout. Hive has a language based on the widely used SQL to interact with data stored inside the database.

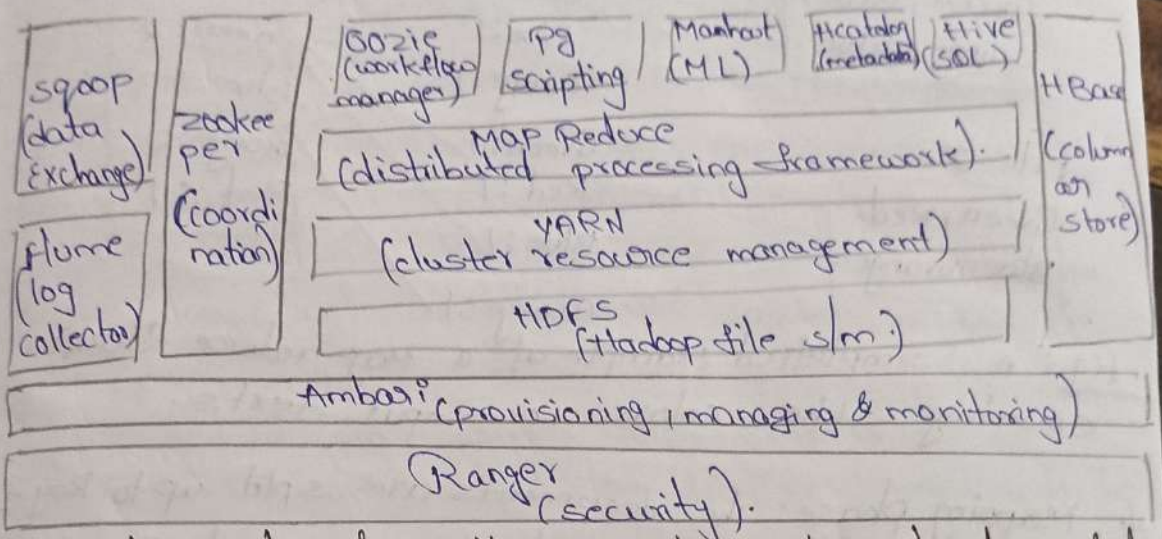


fig: A simple from the ecosystem of applications that arose around the Hadoop core framework.

-) Map Reduce : How Hadoop Acheives Parallelism:

Hadoop uses a Programming method called Map Reduce to acheive Parallelism. A Map Reduce algorithm splits up the data Process it in Parallel, and then sorts, combines and aggregate the results back together. Map Reduce is expensive when working with large data sets.

Lets see how map Reduce could work on a small fictitions examples. you are the director of a toy company. Every toy has two colors, and when a client orders a toy from the web Page the web Page puts an order file on Hadoop with the colors of toy. you will use a map reduce style algorithm to count the colors. first let's simplified version in fig.

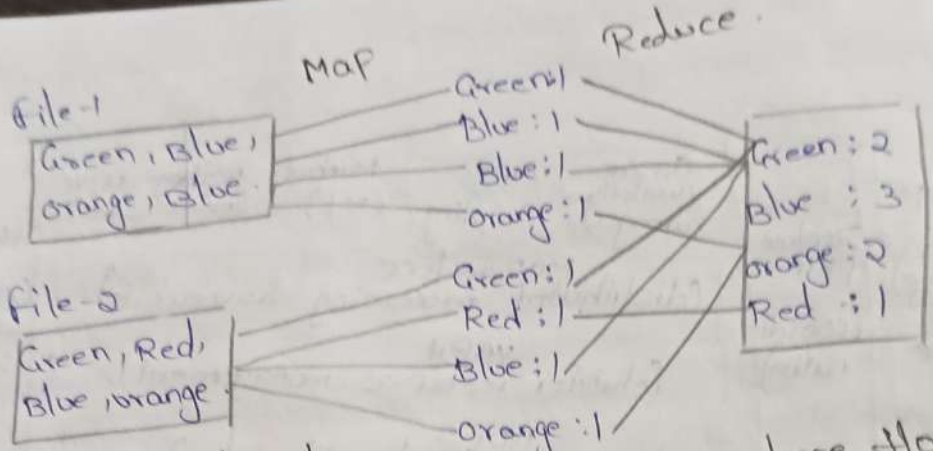
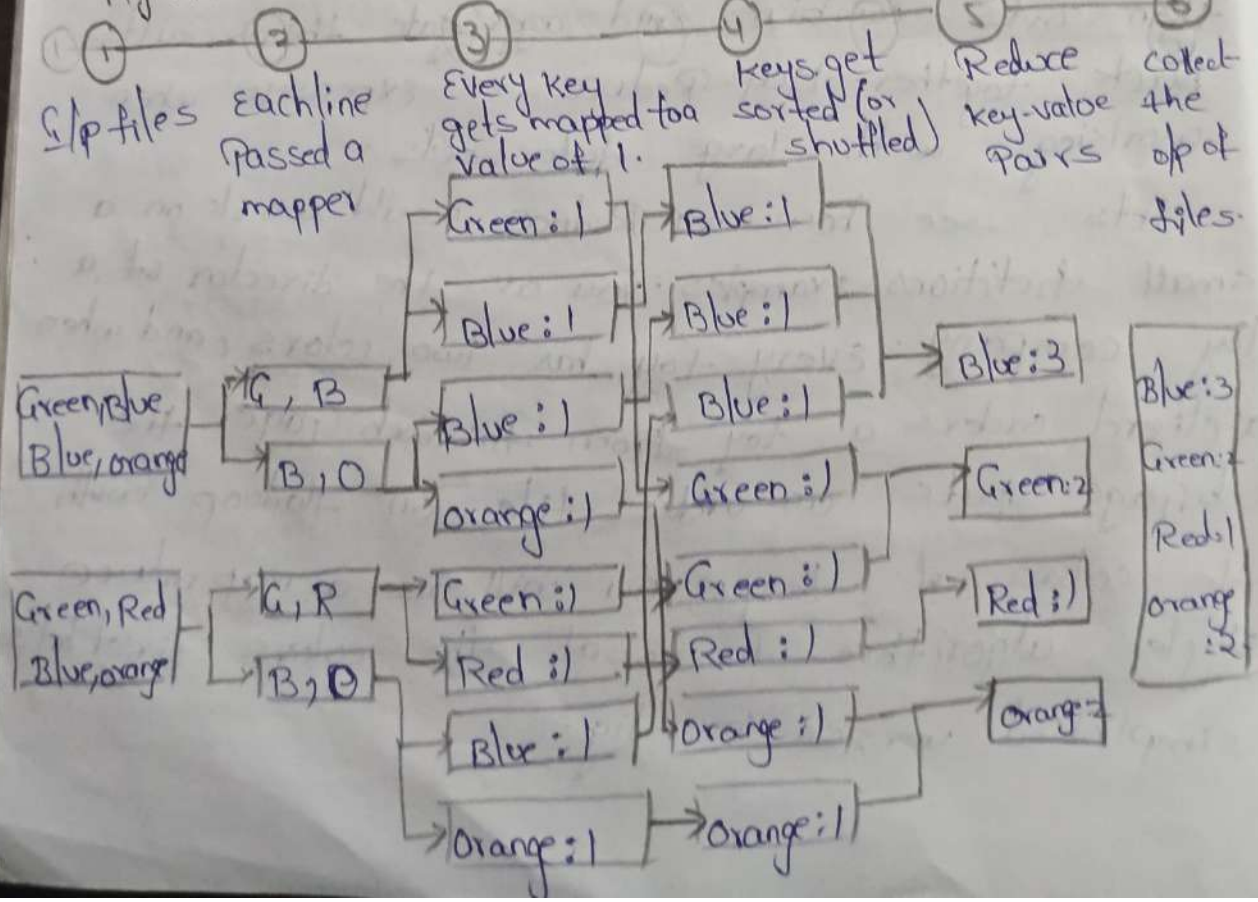


fig: A simplified example of a map reduce flow for counting the colors in input texts.

* Mapping phase: the documents are split up to key-value pair. until we reduce, we can have many duplicates.
* Reduce phase: It's not unlike SQL "group by". The different unique occurences are grouped together, and depending upon the reducing function, a different results can be created.

fig: An example of mapreduce flow for counting the colors in i/p text.



The whole process is described in the following 6 steps:

1. Reading the i/p files.
2. Passing each line to a mapper job.
3. The Mapper job parses the colors (keys) out of the file and outputs a file for each color with the no. of times it has been encountered (value). or more technically said, it map a key to a value.
4. The keys get shuffled and sorted to facilitate the aggregation.
5. The reduce phase sums the numbers of occurrences per color and output one file per key with the total no. of occurences for each.
6. The keys are collected in an output file.

Spark : replacing MapReduce for better performance:

Data scientists often do interactive analysis and rely on algorithms that are inherently iterative; it can take a while until an algorithm converges to a solution. It is a weak point of Map Reduce framework, to introduce the spark framework to over come it.

What is spark?

Spark is a cluster computing framework similar to Map Reduce. spark, however, doesnot handle the storge of files on the file s/m itself, nor does it handle the resource management.

Hadoop & and spark are thus complementary s/ms. For testing and development, you can even run spark on your local s/ms.

How does spark solve the problems of MapReduce?
sparks create a kind of shared RAM memory b/w the computer of your cluster. this allows the different workers to share variables and thus eliminates the need to write the intermediate result to disk. spark uses Resilient Distributed Database (RDD), Because it's an inmemory sln, it avoids costly disk operations.

The different components of spark ecosystem:
spark core provides a NoSQL environment well suited to interactive, exploratory analysis.

spark has four other large components, as listed in below fig:

1) spark streaming is a tool for real-time analysis
2) spark SQL provides a SQL interface to work with spark.
3) MLLib is a tool for ML inside the spark framework
4) Graph X is a graph database for spark.

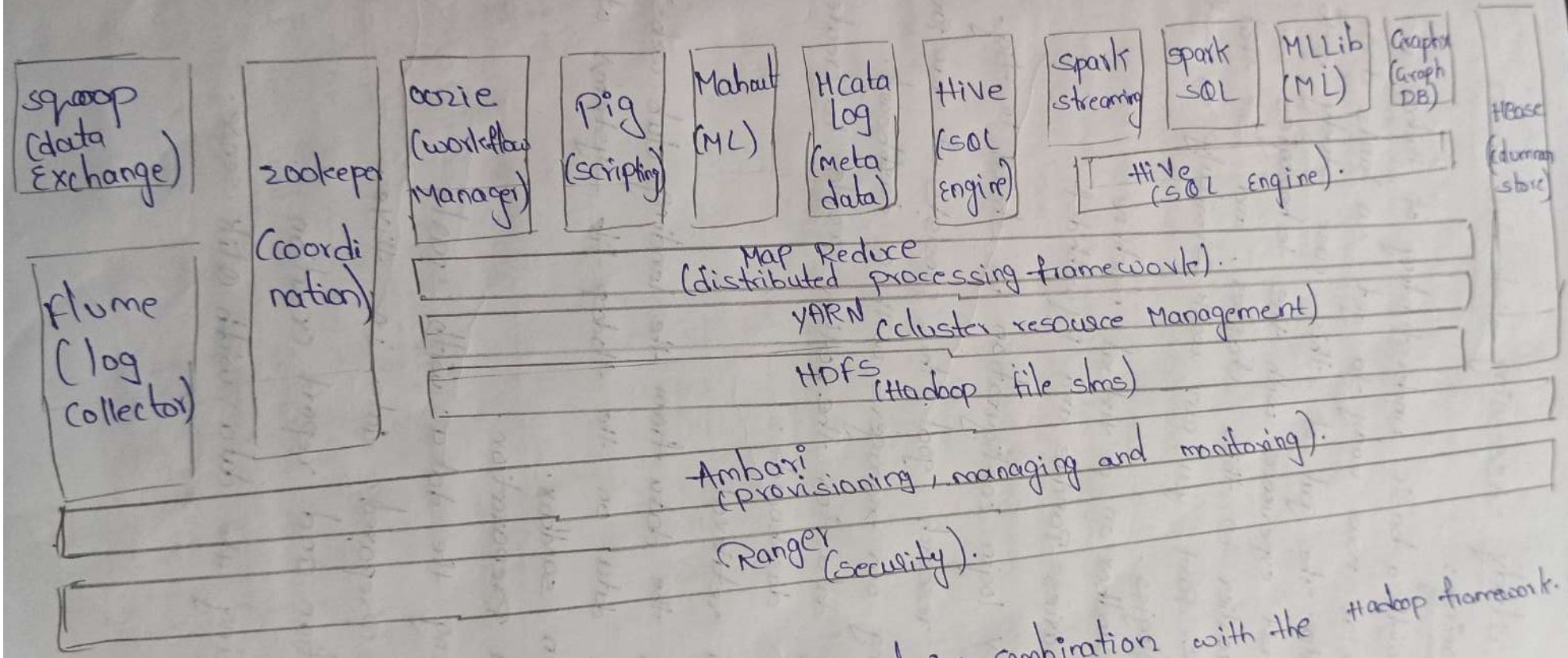


fig: the spark framework when used in combination with the Hadoop framework.

(2) Case study : Assessing risk when loan Sanctioning :

Enriched with a basic understanding of Hadoop and spark, we're now ready to get our hands dirty on big data. The goal of this case study is to have a first experience with the technologies we introduced earlier in this chapter, and see that for a large part you can work similarly as with other technologies.

→ Horton sandbox on a virtual machine.
→ Python libraries: Pandas and pywebhdfs.

The PuTTY login configuration is shown in fig.

steps 1 : The research goal. this consists of two parts:
* Providing our managers with a dashboard.
* preparing data for the other people to create their own dashboards.

step-2 : Data retrieval.
* Downloading the data from the lending club website.
* Putting the data on the Hadoop file systems of the Horton sandbox.

step-3 : Data preparation.
* Transforming the data with spark
* storing the prepared data in Hive.

step-4 : Exploration and report creation
* visualizing the data with Qlik sense.

step-1 : The research goal:

The lending club is an organization that connects people in need of loan with people who have money to invest. To acheive this, you will create a report for him that gives him insight into the average rating, risk and return for lending money to a certain person.

By going through this Project, you make the data accesible in a dashboard tool, thus enabling other people to explore it as well. Self-service Business Intelligence is often applied in data-driven organizations that don't have analyst to spare.

By the end of this case study, you will create a report.

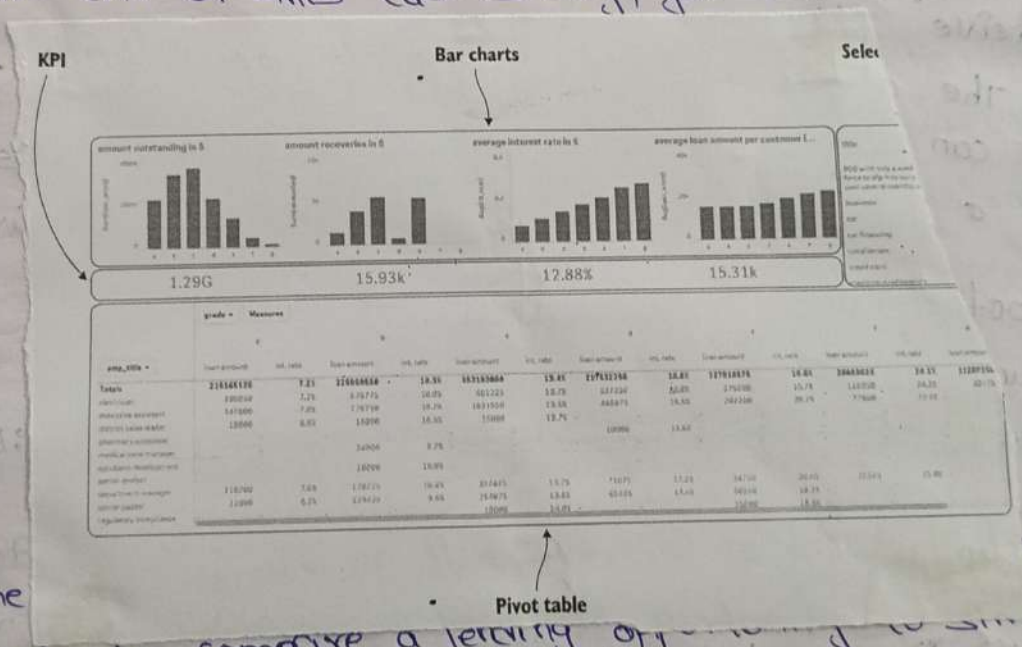


fig: The ... dashboard to compare a lending ... opportunities.

Step-2: Data Retrieval:

It's time to work with the Hadoop file system (or hdfs). First we will send commands through the command line line and then through the python scripting language with the help of pywebhdfs Package.

The Hadoop file sys is similar to the normal life file system, except that stores files and folders are stored over multiple servers and you don't know the physical address of each file. This is not unfamiliar if you have worked with tools such as drop box or Google drive. As a normal file system you can create, rename, and delete files and folders.

Using the command line to interact with the Hadoop file system:

Let's first retrieve the currently present list of directories and files in the Hadoop root folders using the command line. Type the command hadoop fs -ls / in PUTTY to acheive this.

The output of the Hadoop command is shown in fig. you can also add arguments such as hadoop fs -ls -R / to get a recursive list of all the files and subdirectories.

```
[root@sandbox ~]# hadoop fs -ls /
Found 20 items
drwxrwxrwx   - admin    hadoop          0 2015-07-14 14:15 /loan stats
-rw-r--r--   1 root     hadoop    1208345520 2015-7-14 14:47 /
drwxrwxrwx   - yarn     hadoop          0 2015-07-15 13:32 /app-logs
drwxr-xr-x   - hdfs     hdfs            0 2015-6-05  9:19 /apps
```

fig: output from the Hadoop list command: hadoop fs -ls / the hadoop root folder is listed.

The hadoops commands are very similar to our local file system commands (POSIX style) but starts with Hadoop fs and have a dash - before each command.

Goal	command Hadoop file slm	Local file slm command
Get a list of files and directions from a directory.	hadoop fs -ls URI	ls URI
create a directory	hadoop fs -mkdir URI	mkdir URI
Remove a directory.	hadoop fs -rm -r URI	rm -r URI
change the permission of files.	hadoop fs -chmod mode URI	chmod MODE URI
move or rename file.	hadoop fs -mv OLDURI NewURI	mv OLDURI NEWURI

there are two special commands you will use often. these are

→ upload files from the local file system to the distributed file system (hadoop fs -put local URI Remote URI)

→ Download a file from the distributed file system to the local file system (hadoop -get RemoteURI).

step-3: Data preparation:

Now we have downloaded the data for analysis, we will use spark to clean the data before we store it in Hive.

Data preparation in spark:

cleaning data is often an interactive exercise, because you spot a problem and fix the Roblem and you will likely do this do this a couple of times before you have clean and crisp data.

Sin spark is well suited as analysis because it does not save the data after each step and has a much better model than Hadoop for sharing data blw servers.

The transformation consists of four parts:

-> startup Pyspark and load the spark and Hive context.
-> Read and parse the .csv file
-> split the header line from the data.
-> clean the data.

step-4 : Data exploration & step-6 : Report building.

we will build an interactive report with Qlik sense to show to our manager. Qlik sense can be downloaded from http://www.qlik.com/try-or-buy/download-qlik-sense after subscribing to their websit

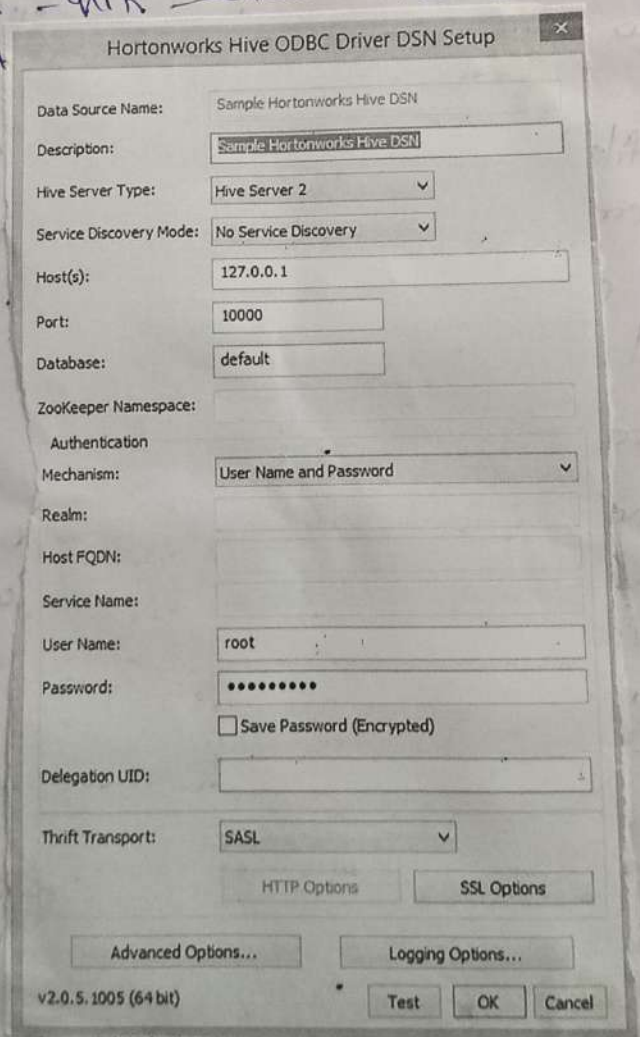


fig : window [illegible] configuration.

Qvf- Qlik can either take the data directly into memory or make a call every time to Hive. We have choosen this first method because it works faster.

this part has three steps:

1. Load data inside Qlik with an ODBC connection.
2. Create the report.
3. Explore data.

let's start with first step, loading data int Qlik.

step-1 : load data in Qlik Qlik.

when you start Qlik sense it will show you a welcome screen with the existing reports as in fig.

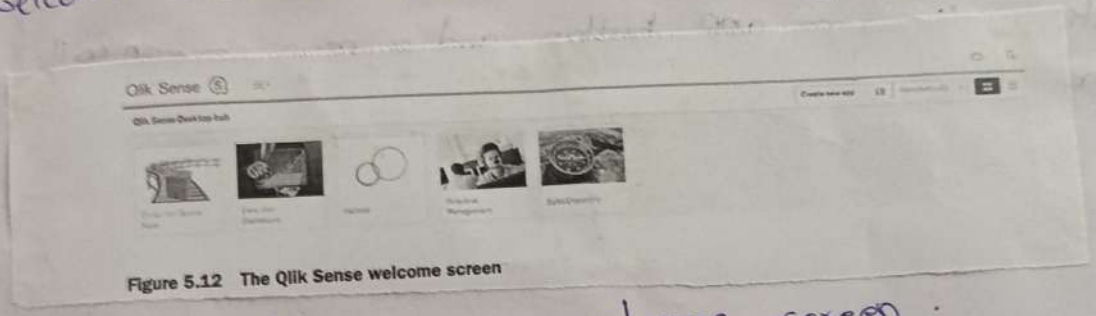


Figure 5.12 The Qlik Sense welcome screen

fig: the Qlik sense welcome screen.

Go start a new app, click on the create new app button on the right of the screen, as shown in fig: This opens up a new dialog box. enter "chapter 5" as the new name of our app.

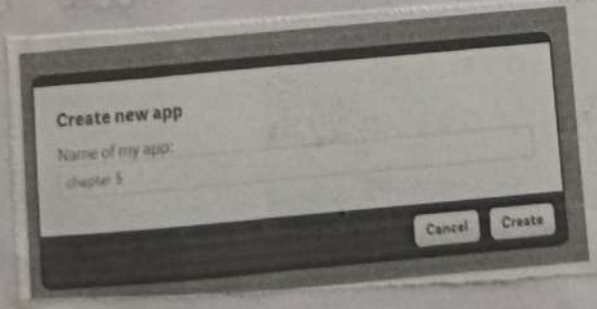


Figure 5.13 The Create new app message box

fig: the create new app message box.

A confirmation box appears if the app is created successfully.

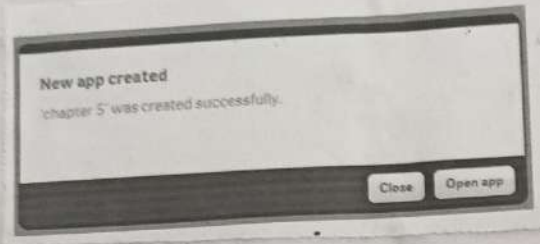


Figure 5.14 A box confirms that the app was created successfully.

fig: A box confirms that the app was created successfully.

click on the open app button and a new screen will prompt you to add data to the application.

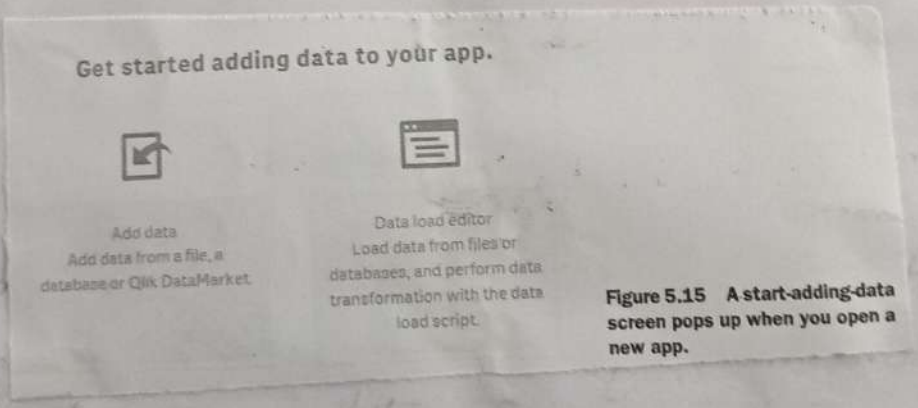


Figure 5.15 A start-adding-data screen pops up when you open a new app.

fig: A start adding data screen pops up when you open a new app.

click on the add data button and choose ODBC as a data source.

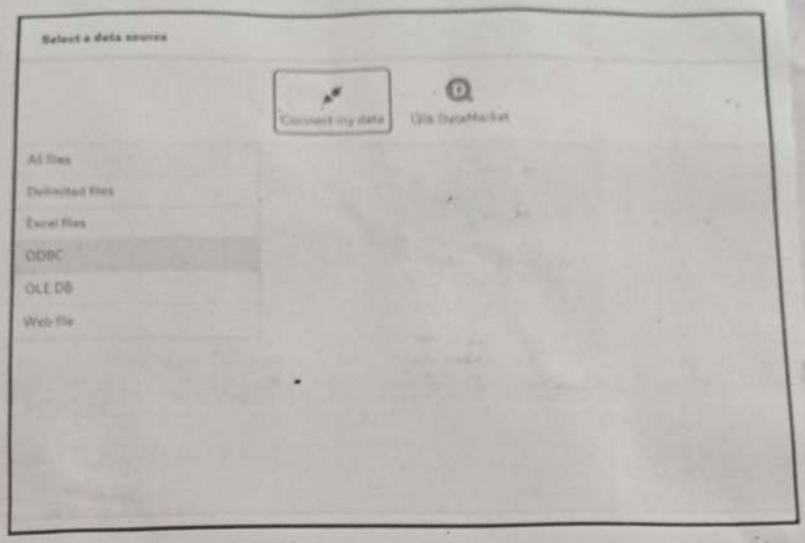


Figure 5.16 Choose ODBC as data source in the Select a data source screen

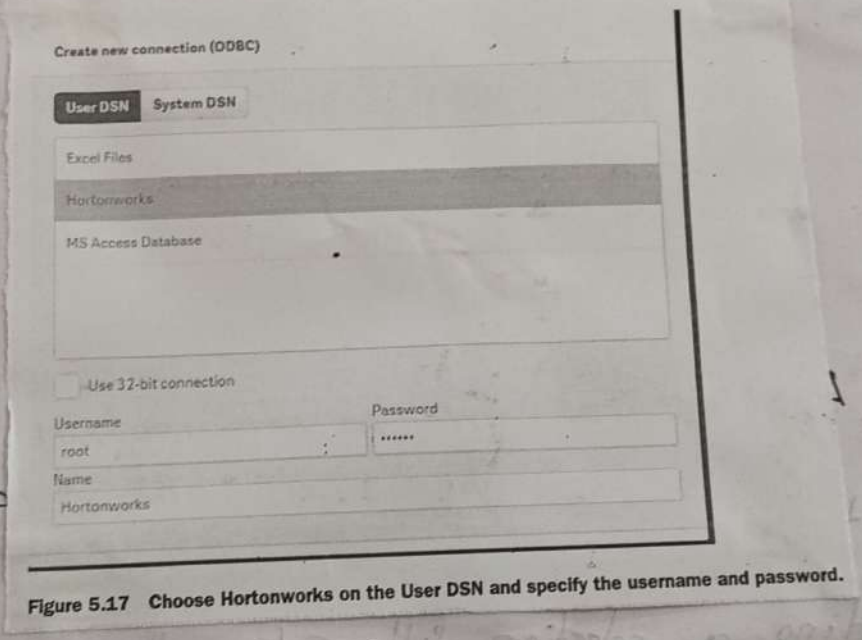


Figure 5.17 Choose Hortonworks on the User DSN and specify the username and password.

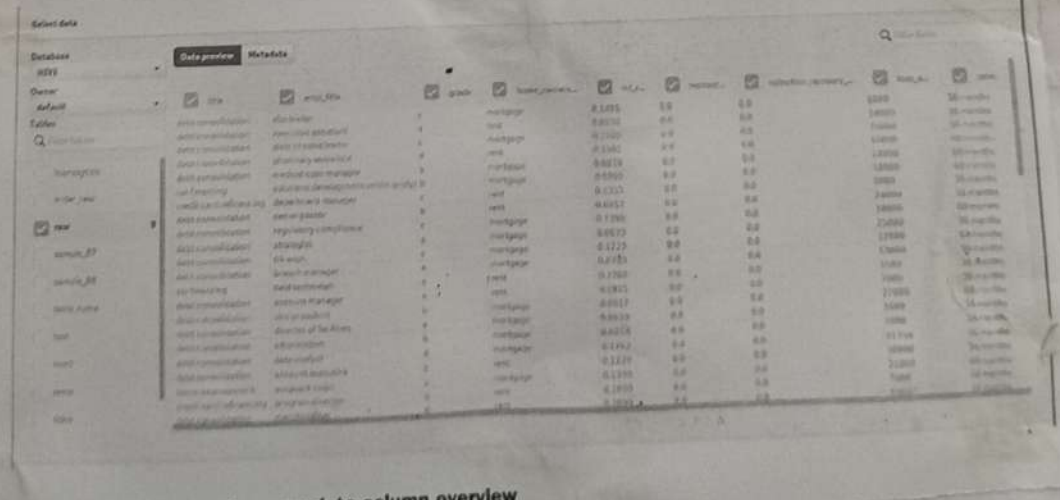


Figure 5.18 Hive interface raw data column overview

After step, it will take a few seconds to load the data in qlik.

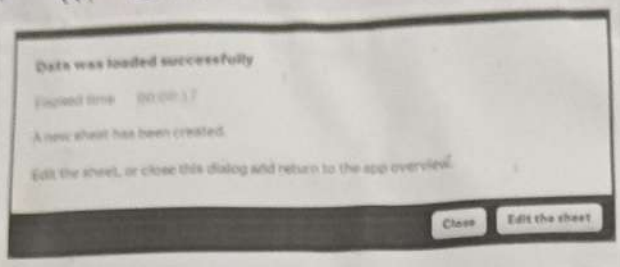


Figure 5.19 A confirmation that the data is loaded in Qlik

fig: A confirmation that the data is loaded in qlik.

step-2: create the report
choose the edit the sheet to start building the report. this will add the report editor.

Figure 5.20 An editor screen for reports opens

fig

step1: Adding a selection filter to the report.

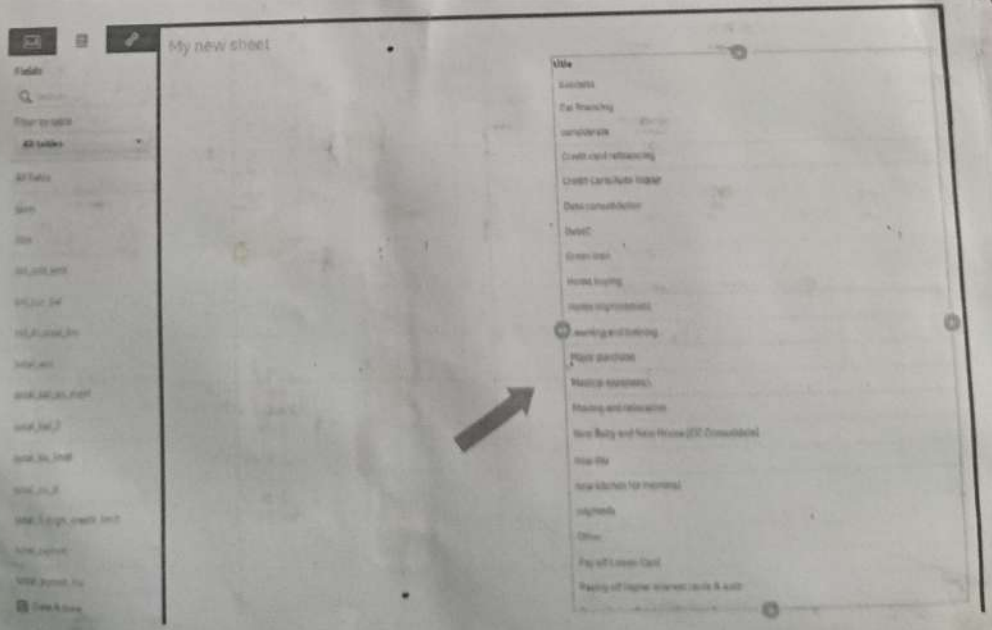


Figure 5.21 Drag the title from the left Fields pane to the report pane.

substep 2: Adding a KPI to the report:

the kpi chart shows an aggregated no. of population that's selected. Numbers such as average interest rate and the total no. of customers are shown in chart.

Adding a KPI to a report takes four steps:

1. choose a chart - choose KPI as the chart and place it on the report screen; resize and position to your linking.
2. Add a measure - click the add measure button inside the chart and select int_rate.
3. choose an aggregation method - Avg (int_rate).
4. format the chart - on the right pane, fill in average interest rate as label.

Average Interest Rate

0.13

fig: An example of a KPI chart.

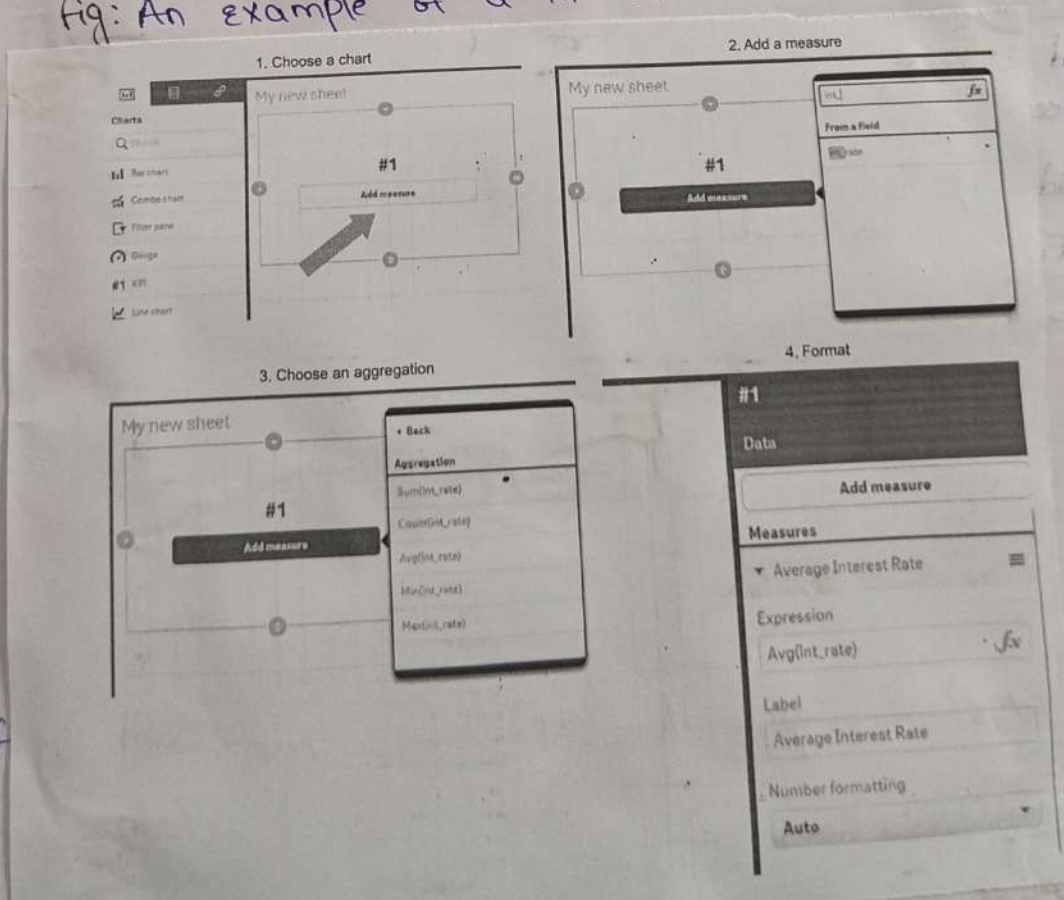


Figure 5.23 The four steps to add a KPI chart to a Qlik report

* Average interest rate
* Total loan amount
* Average loan amount
* Total recoveries.

Substep-3 ; Adding bar charts to our report:

Next we will add four bar charts to the report. These will show the different numbers for each risk grade.

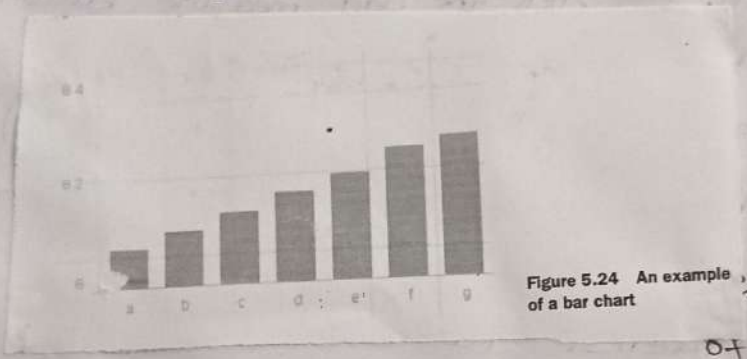


Figure 5.24 An example of a bar chart

; An example of a bar chart.

Adding bar chart to a report takes five steps.

1. choose a chart
2.
3.
4.
5.

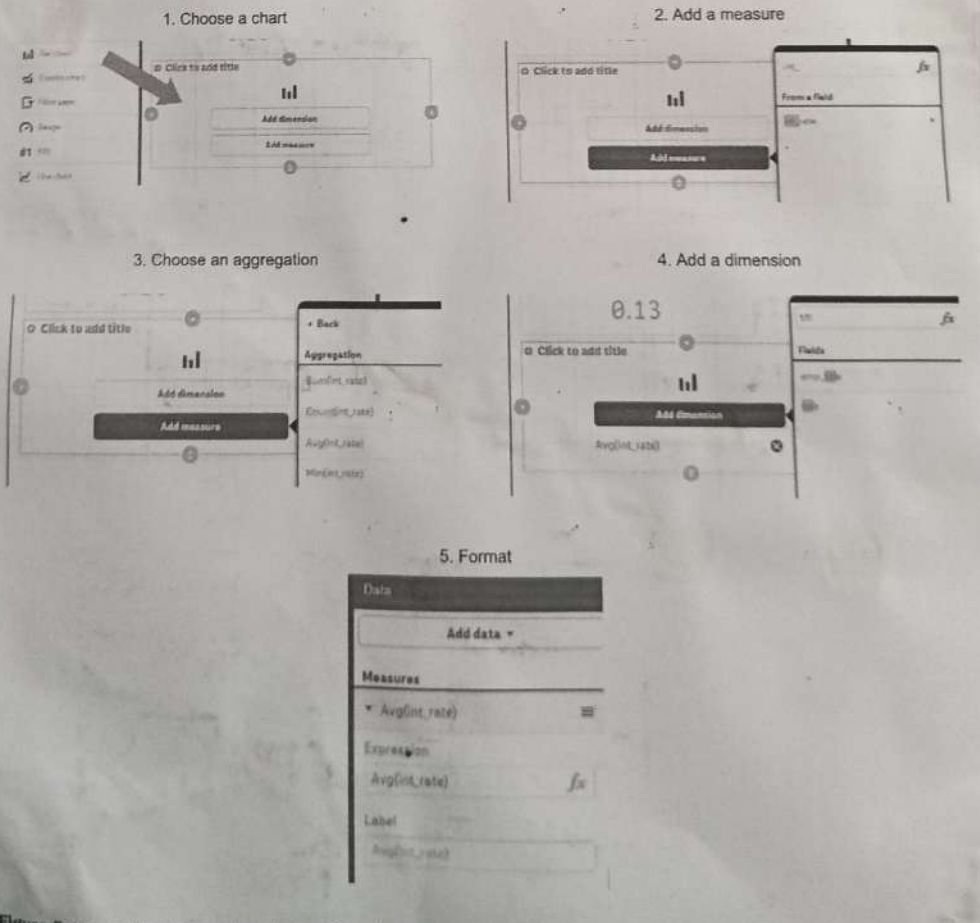


Figure 5.25 Adding a bar chart takes five steps.

Repeat this procedure for the following dimension and measure combinations:

→ Average interest rate per grade.
→ Average load amount per grade
→ Total loan amount per grade.
→ Total recoveries per grade.

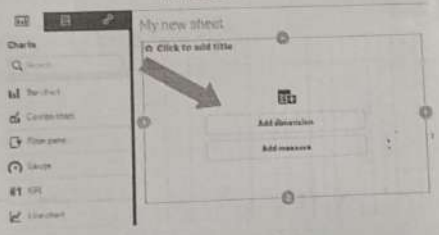
substep-4 : Adding a cross-table to the report:

average interest rate per job title / risk grade

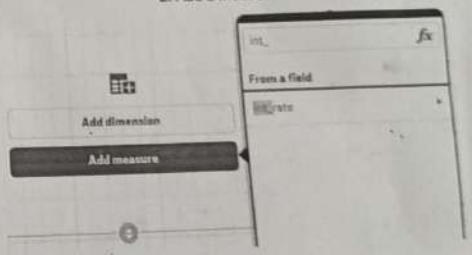
emp_title	a	b	c	d
electrician	0.072455172	0.099825	0.13674545	0.16769333
executive assistant	0.069928571	0.1023193	0.13615811	0.16489691
district sales leader	0.0697	0.1049	0.1269	
pharmacy associate				0.1561
medical case manager		0.0818		
solutions development senior analyst		0.0999		
	0.075866667	0.10196667	0.132177	0.17185

Figure 5.26 An example of a pivot table, showing the average interest rate paid per job title/risk grade combination

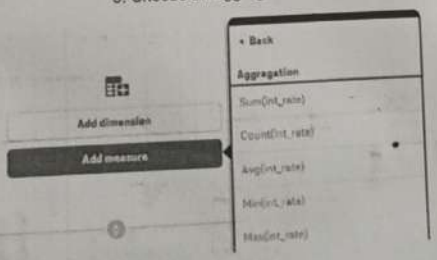
1. Choose a chart



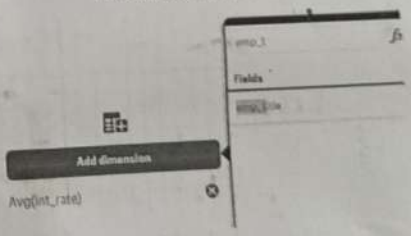
2. Add a measure



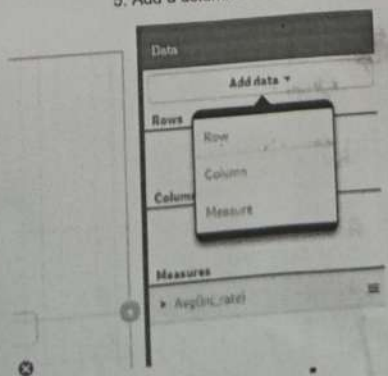
3. Choose an aggregation



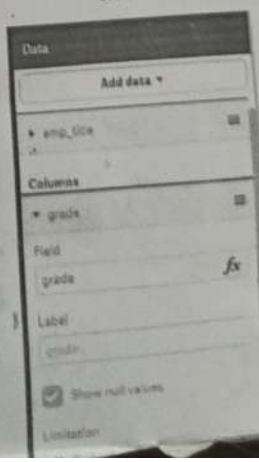
4. Add a row dimension



5. Add a column dimension



6. Format



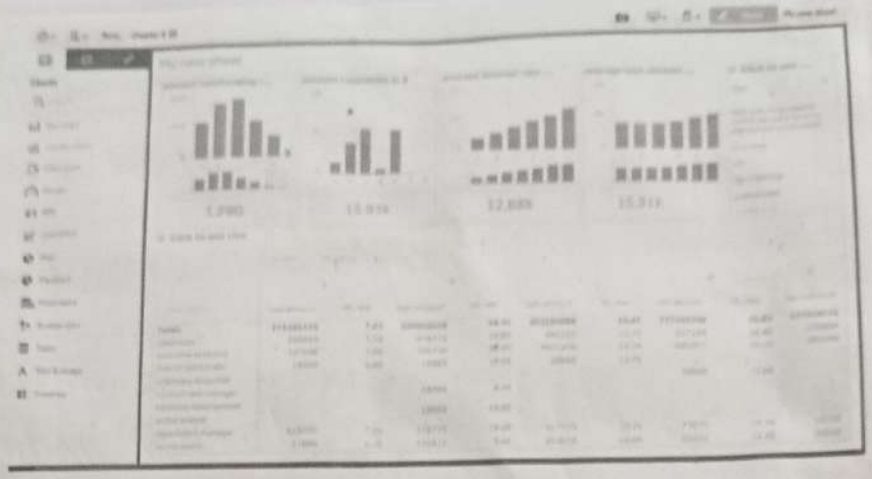


Figure 5.28 The end result in edit mode

Fig: End of the result in edit mode.

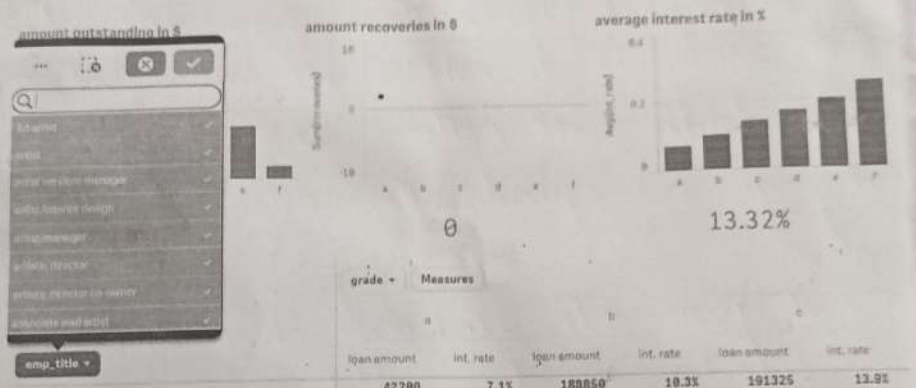


Figure 5.30 When we select artists, we see that they pay an average interest rate of 13.32% for a loan.

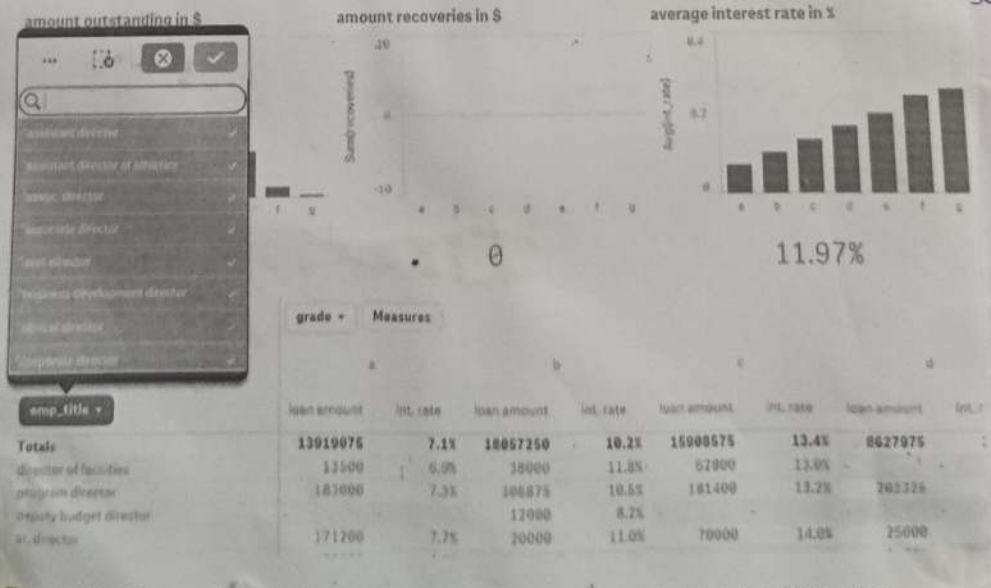


Figure 5.29 When we sel... directors, we can see that they pay an average rate of 11.97% for a loan.

(3) ACID - Principle of relational database.

The main aspects of a traditional relational database can be summarized by the concept ACID.

→ Aut

→ Atomicity :- The "all or nothing" principle. If a record is put into a database, it's put in completely or not at all. If, for instance, a power failure occurs in the middle of a database write action, you would not end up with half a record; it would not be there at all.

→ Consistency :- This important principle maintains the integrity of the data. No entry that makes it into database will ever be in conflict with predefined rules, such as lacking a required field or a being numerical instead of text.

→ Isolation : When something is changed in the database, nothing can happen on this exact same data at exactly the same moment. Isolation is a scale going from low isolation to high.

→ Durability :- If data has entered the database, it should survive permanently. Physical damage to the hard disc will destroy records, but power outages and software crashes should not.

ACID applies to all relational database and certain NoSQL databases, such as graph database Neo4j. For most other NOSQL database another Principle applies: BASE to understand BASE and why it applies to most NOSQL database we need to look at the CAP theorem.

(4) CAP Theorem:-

Once a database get spread out over different servers, it difficult to follow the ACID Principle because of the consistency ACID promise; the CAP theorem points out why this becomes Problematic.

* Partition tolerant — the database can handle a network partition or network failure.
* Available — As long as a node you are connecting to is up and running and you can connect to it, the node will respond, even if the connection blw the different database nodes is lost.
* consistent — No matter how which node you connect to you will always see the next same data.

for a single node database it's easy to see how it's always available and consistent.

* Availability — As long as the node is up, it's available. that's all the CAP availability promises.
* Consistent — There is no second node, so nothing an inconsistent.

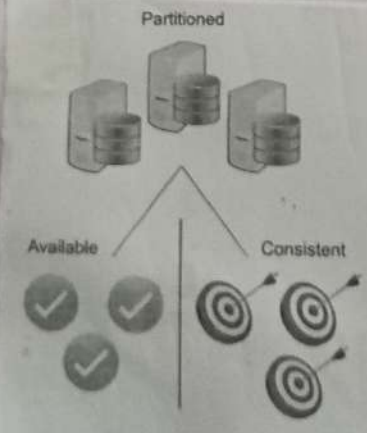


Figure 6.2 CAP Theorem: when partitioning your database, you need to choose between availability and consistency.

In the first case, Fritz and Freddy both buy the octopus coffee table, because the last-known stock number for both nodes is "one" and both nodes are allowed to sell it as shown in fig:

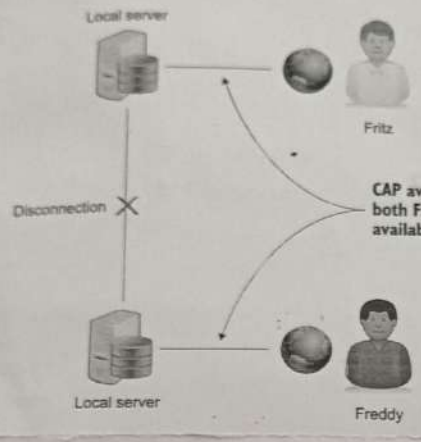


Figure 6.3 CAP Theorem: if nodes get disconnected, you can choose to remain available, but the data could become inconsistent.

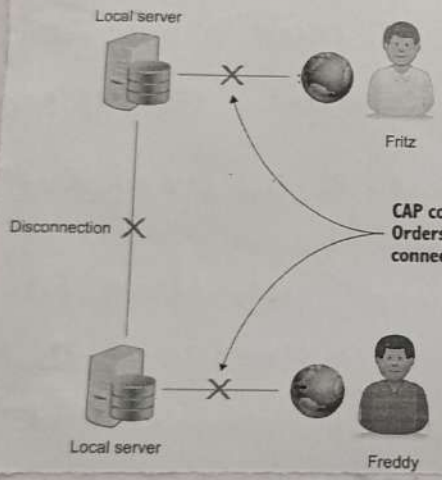


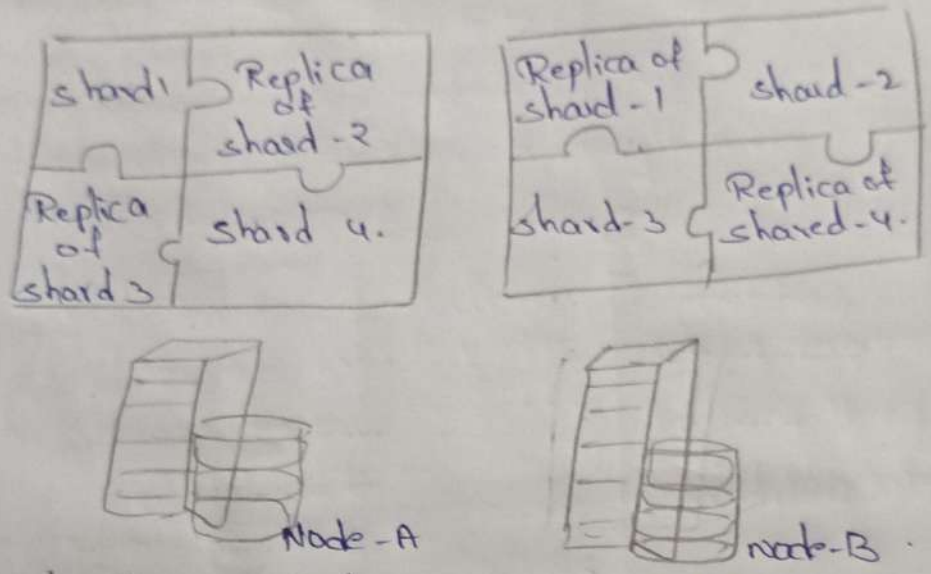
Figure 6.4 CAP Theorem: if nodes get disconnected, you can choose to remain consistent by stopping access to the databases until connections are restored.

(5) Base Principles of NoSQL databases:

RBDMS follows the ACID Principles; NoSQL databases that don't follow ACID, such as a document stores and key-value stores, follow BASE. BASE is a set of much softer database promises:

Basically available: Availability is guaranteed in the CAP sense. Taking the web shop example, if a node is up and running, you can keep on shopping

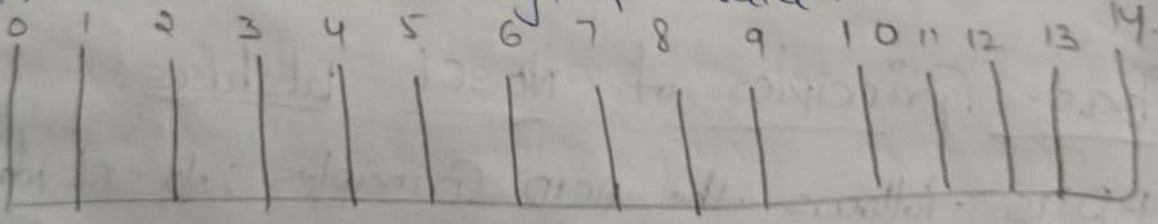
soft state: the state of a system might change over time. this corresponds to eventual consistency principle. the sbm will have to change to make the data.

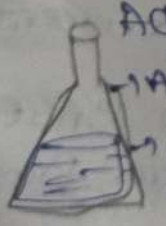


Eventually consistency: the database will become consistent over time.

—Acid - Provide a Consistent State
BASE - Provide high availability

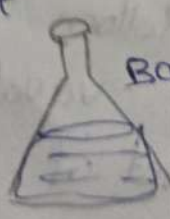
ACID VERSUS BASE:
The Base principles are some what contrived to fit acid and base from chemistry: an acid is a fluid with a low pH value. a base is the opposite and has a high pH value.





ACID
Atomicity
consistency
isolation
Durability

BASE



Basically available
soft state
Eventual consistency.

fig: Acid versus BASE: traditional relational database versus most NoSQL database.

(6) Types of NoSQL database

There are four big NoSQL types: key value store, document store, column oriented database, and graph database. and graph Each type solves a problem that can't be solved with relational databases. Actual implementations are often combinations of these. OrientDB, for example, is a multi-model database, combining NoSQL types.

person info table: represent person-specific information.

Name	Birth	Person-ID
JO S the Bass	11-12-1985	1
Fritz von Braun	27-1-1978	2
freddy	-	3
Delphine	16-9-1986	4

person ID	HobbyID
1	2
1	2
2	3
2	4
3	5
3	6
3	1

person hobby linking table: necessary because of the many to many relationship b/w hobbies & persons

Hobby ID	Hobby name	Hobby description
1	Archery	shooting arrow from a bow.
2	Conquering the world	looking for trouble with your neighbour.
3	Building things	Also known as construction
4	suting	catching waves
5	swordplay	fencing with swords
6	Lollygagging	hanging around doing nothing.

Hobby info table: represents hobby-specific information.

column-oriented database:

Traditional relational database are row-oriented, with each row having a row id and each field within the row stored together in a table.

RowID	Name	Birthday	Hobbies
1	Jos the Boss	11-12-1985	Archery, conquering the world
2	Fritz von Braun	27-1-1978	Building things
3	Freddy stark		swordplay, archery
4	Delphine thewisone	16-9-1986	

fig: Row oriented database layout. Every entity is represented by a single row, spread over multiple columns.

Row ID	name	Birthday	Hobbies
1	Jos the Boss	11-12-1985	Archery, conquering world
2	Fritz von Braun	27-1-1978	Building things
3	Freddy stark		swordplay, archery.
4	Delphine thewis one.	16-9-1986.	

Fig: Row oriented lookup: from top to bottom and for every entry, all columns are taken it memory.

Key-value stores:-

Key-value stores are the least complex of the NoSQL database, they are, as the name suggests a collection of key value pairs, as shown in fig:

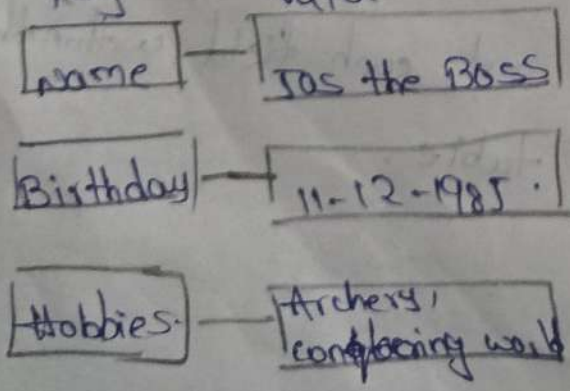


fig: key-value stores everything as a key and as value.

Document store:

Document store as are step up in complexity from key-value stores & a document store does assume a certain document structure that can be specified with the schema. Document stores appear the most natural among the noSQL database types because they are designed to store everyday documents as is and they allow for complex querying calculation on this often already aggregated data.

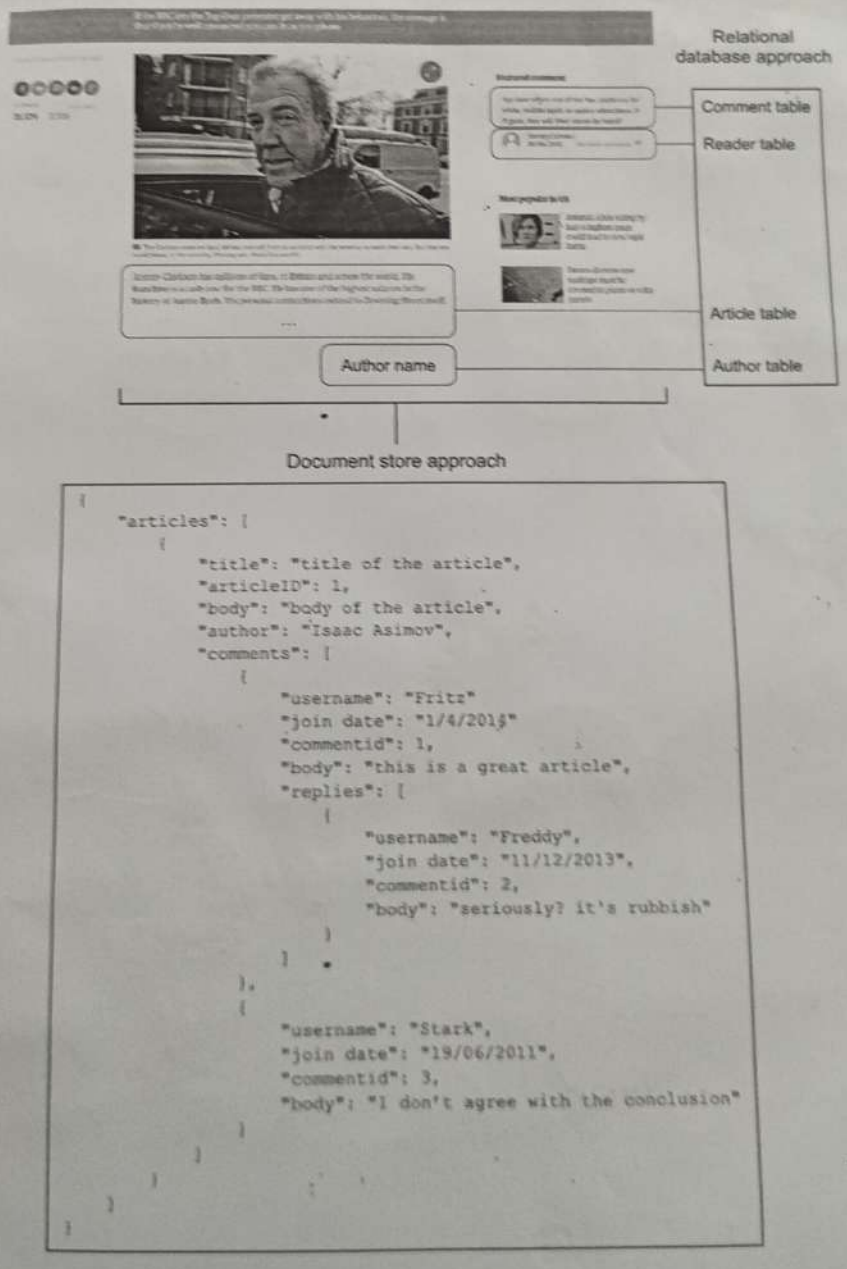


Figure 6.13 Document stores save documents as a whole, whereas an RDMS cuts up the article and saves it in several tables. The example was taken from the *Guardian* website.

Graph database :-

The last big nosql database type is the most complex one, geared toward storing relations to entities in an efficient manner. when the data is highly interconnected, such as for social network scientific paper citations or capital assest clusters, graph databases are the answer. Graph or n/w data has two main components.

1) Node :- the entities themselves. In a social n/w this could be people.

2) Edge - the relationship b/w two entities. The relationship is represented by a line and has its own properties.

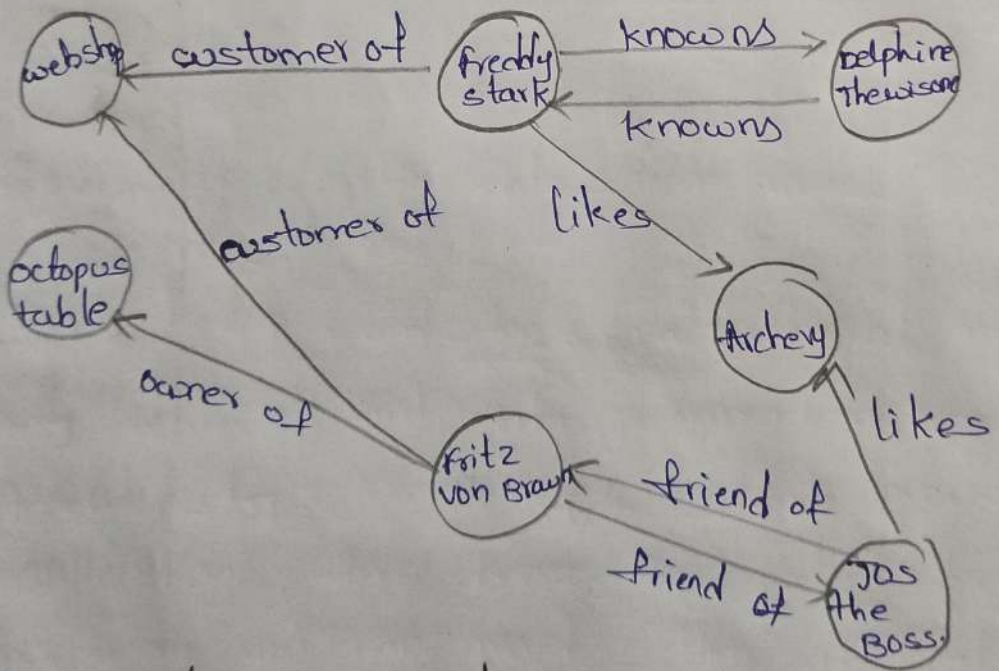


fig: Graph data example with four entity types and their relations without extra edge or node information.

(8) Case study on Disease diagnosis & Profiling.

In this case study, you will learn how to build such as search engine here, albeit using only a fraction of the medical, data that is freely accessible. To tackle the Problem you will use a modren NoSQL database called Elastic search to store the data, and the data science Process to work with the data and turn in into a resource that fast and easy to search. Here's how you will apply the Process.

1. Setting the research goal.
2. Data collection :- you will get your data from wikipedia. There are more sources out there, but for demonstration purpose a single one will do.
3. Data Preparation : the wikipedia data might not be perfect in it's current for mat. you will apply a few techniques to change this.
4. Data exploration :- you use case is special in that step-4 of the data science process is also the desired end result:
5. Data Modelling :- no real data modeling is applied in this chapter. Document term matrices are used for search are often the starting point for advanced topic modelling.
6. Presenting results :- To make the data searchable you did need a user interface such as a website were people can query and retrieve disease information.

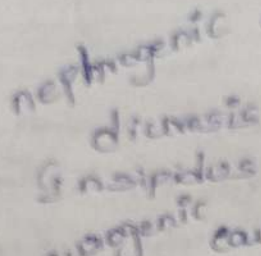


fig: A simple word cloud on non-weighted diabetes keywords.

step-1: setting the research goal:

Can you diagnose a disease by the end of this chapter, using nothing but your own home computer and the free software and data science process.

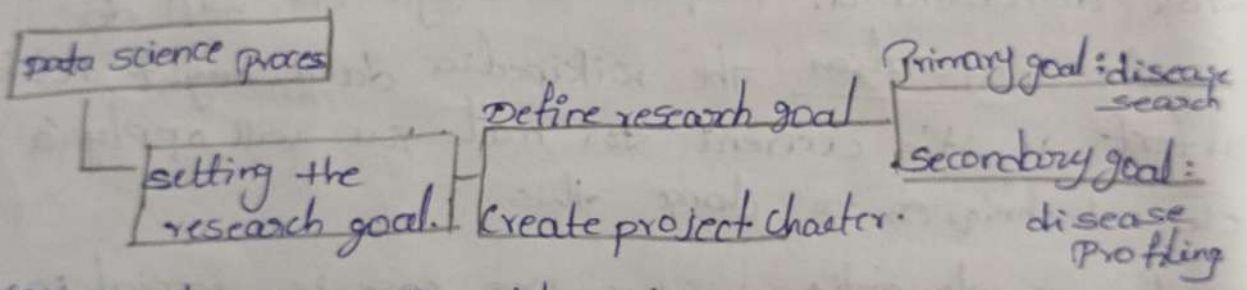


fig: step-1 in the data science process: setting the research goal.

step-2: step-2 and step-3: Data retrieval and preparation

Data retrieval and data preparation are two distinct steps in the data science process and even though this true for case study. this way you can avoid setting up local intermedia storage and immediately do data preparation while data is being retrieved. let's look at where we are in the data science process.

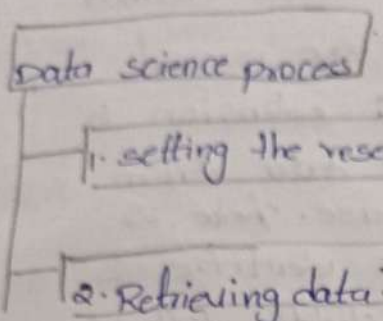


fig: Data science process step-2 : data retrieval. In this case there is no internal data, all data will be fetched from wikipedia.

Internal Data: you have no disease information lying around. If you currently work for a pharmaceutical company or hospital, you might be luckier.

External Data: All you can use for this case is external data. you have several possibilities, but you will go with wikipedia.

As shown in figure there are three distinct categories of data preparation consider:

→ Data cleansing: The data you will pull form wikipedia can be incomplete or erroneous. Even if you dumped on HTML tags on purpose, they did be unlikely to influence the results.

→ Data transformation:- you did not need to transform the data much at this point. you want to search it as is. But you will the distinction blw the Page title, disease name and Page body.

Combining data :- All the data is drawn from a single source in this case, so you have no real need to combine data. A possible extension to this excercise would be get disease data from another source and match the disease. there is no trivial task because no unique identifier is present and the names are often slightly different.

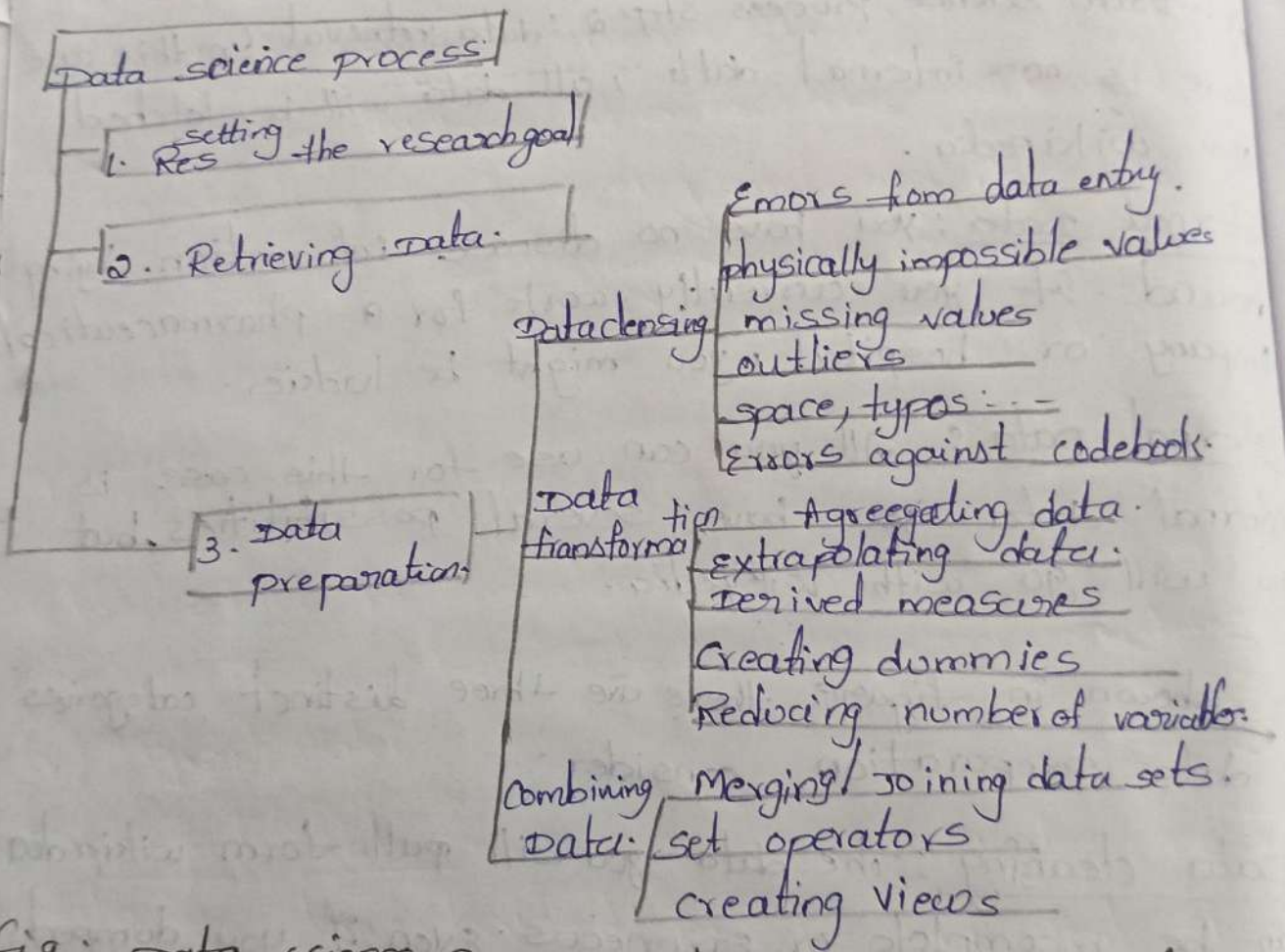


fig: Data science process step-3 : Data preparation.

step-4 : Data Exploration :-

Data exploration is what marks this case study, Because the primary goal of the a specific way of exploring the data query for disease symptoms.

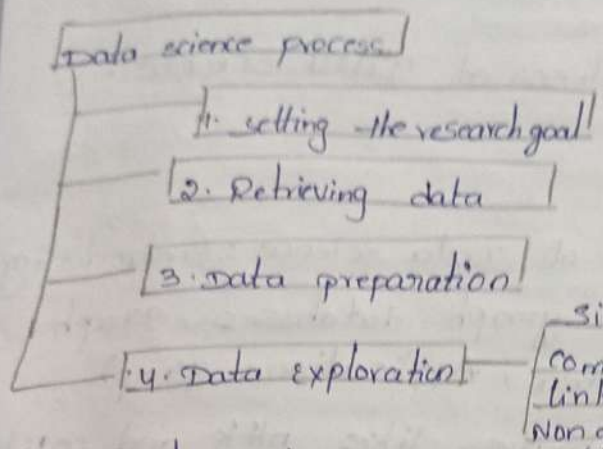


fig: Data science process step-4: data exploration.

step-6 : Presentation and Automation

your primary objective, disease diagnostics, turned into a self-service diagnostics tool by allowing a physician to query it via, for instance, a web application.

Elastic search for web application;-

As with any other database, it's bad practice to expose your elastic search REST API directly to the front end of web application. security can be added to elastic search directly using "sheild", an elastic search payable service.

The secondary objective, disease Profiling., you can also be taken to the level of a user interface; it's possible to let the search result Produce a word cloud that visually summarize the search resultt.

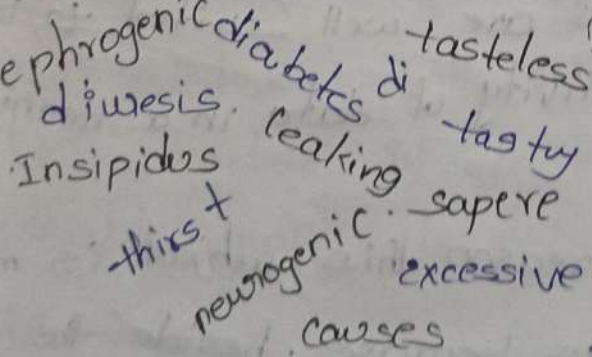


fig: unigram word cloud on non-weighted diabetes on keyword from elastic search.