SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIT – IV – Internet of Things – SCSA5301

DATA ANALYTICS AND IOT PLATFORM

Big Data Analytics - Apache Hadoop - Using Hadoop Map Reduce for Batch Data Analysis - Apache Storm - Data Visualization - Visualization tools for IoT.

1. Big Data Analytics

Big Data Analytics is "the process of examining large data sets containing a variety of data types – i.e., Big Data – to uncover hidden patterns, unknown correlations, market trends, customer preferences, and other useful information." Companies and enterprises that implement Big Data Analytics often reap several business benefits, including more effective marketing campaigns, the discovery of new revenue opportunities, improved customer service delivery, more efficient operations, and competitive advantages. Companies implement Big Data Analytics because they want to make more informed business decisions. Big Data Analytics gives analytics professionals, such as data scientists and predictive modelers, the ability to analyze Big Data from multiple and varied sources, including transactional data

Characteristics

Big Data is often defined in terms of **"3V's"** i.e.

Volume - the amount of data generated, stored and analysed. The amount of data stored determines the level of insight that can be obtained from that data;

Variety - type and nature of data. Historically data was structured and from a single source - in which case it would fit readily into 'columns' and 'rows'. Increasingly data is sourced from a variety of sources with many different formats;

Velocity - the speed at which data is generated and processed. Where historically data could reasonably be expected to be uploaded via a daily 'batch' process now data is measured in thousands or even millions of transactions per minute.

In addition, other **"V's"** may be added including:

Variability - Variations in the data sets. For example is a temperature measured in degrees Celsius, Fahrenheit or Kelvin;

Veracity - Quality of the captured data. Where decisions are being made on data you need to be sure that the data is correct.

Analytics

Analytics is the scientific process of discovering and communicating the meaningful patterns which can be found in data.

It is concerned with turning raw data into insight for making better decisions. Analytics relies on the application of statistics, computer programming, and operations research in order to quantify and gain insight to the meanings of data. It is especially useful in areas which record a lot of data or information.

Types

"**Descriptive**: A set of techniques for reviewing and examining the data set(s) to understand the data and analyze business performance.

Diagnostic: A set of techniques for determine what has happened and why

Predictive: A set of techniques that analyze current and historical data to determine what is most likely to (not) happen

Prescriptive: A set of techniques for computationally developing and analyzing alternatives that can become courses of action – either tactical or strategic – that may discover the unexpected

Decisive: A set of techniques for visualizing information and recommending courses of action to facilitate human decision-making when presented with a set of alternatives

Challenges for IoT Big Data

Some of the key challenges for IoT Big Data, which have a bearing on the design of architectures suitable for service delivery include

1. **The number of IoT devices**: With forecasted growth in the number of connected "things" expected into the billions world-wide there will be masses of devices which may be a data source, and which may be subject to third party control;
2. **The variety of IoT device**s: There will be enormous variety in the devices which may provide data, even in the case of similar devices e.g. an electronic thermostat. Data from any individual device manufacturer or model may be quite dissimilar from that of

nominally identical devices in such areas as field names, units, and data structures;

3. **Intelligence of IoT devices:** IoT devices have more and more compute resources and integrate several technologies like Graphics Processing Unit (GPU) and Solid State Drive (SSD) storage. Simple sensors are evolving to autonomous systems which will be able to manage their own analytics and be part of large analytics networks;

4. **Risk of IoT device malfunction:** With a great number of IoT devices and manufacturers it is reasonable to assume there will be many occasions where IoT devices malfunction in various ways. In the mos t drastic situations devices will fail completely but it should be expected that more subtle malfunctions will occur which might result in aberrations of data coming from those devices, or a failure of the device to perform a required control function;

5. **Update frequency**: Though some devices (e.g. remote sensors) will produce data reports at a low frequency there may be substantial quantities of data streaming from more sophisticated Internet connected things such as cars;

6. **Historical data**: It is expected that many Big Data insights will derive from historical data recorded from IoT devices. This may be processed alone to derive analytics/ intelligence or considered alongside current data particularly to enable smart monitoring and control;

7. **Context data**: Much IoT data will make more sense when put in context with other data. Context data might be generally "static" (or at least with a slow update period) such as geographical data, or could be more dynamic e.g. weather forecast data. Another important source of context data can be information gathered from the mobile networks themselves e.g. mobile user location or usage dynamics;

8. **Privacy issues:** With so many expected IoT devices acquiring data there could be a substantial risk relating to the disclosure of data which is considered personal to end users. When IoT data is stored in a Big Data system and made available to third parties there is a need to implement strong safeguards to ensure end users remain in control of their personal information. Mobile Network Operators (MNOs) are in a strong position to help users remain in control of their data and to make data available in the best way via consent, aggregation or anonymisation.

General Architecture for IoT Big Data

Context Data Layer

This functional unit is concerned with obtaining external non IoT data ("Context data") which is either available to the third party application or used during the processing of IoT data e.g. "mashing up" IoT data with context data. The Context Data Layer is also able to communicate with the external data sources, e.g. to start and stop data feeds.

Examples of context data might include geographic/ mapping information, weather forecasts, schedules
e.g. for transportation, or information generated from mobile networks/ users. This allows IoT data to be associated with further context data e.g. a moisture sensor reports the current moisture level whilst a weather forecast for the same geographical area identifies whether rain is predicted - allowing a decision to be made as to whether to water a crop.

Context data might be received in different ways e.g. via Hypertext Transfer Protocol (HTTP) based APIs which request data from external servers, information received within an email, via batch file by making an outgoing File Transfer Protocol (FTP) request or by a batch file being deposited via FTP, or data received using removable media. This unit is principally concerned with implementing the relevant adapters in order to receive the various types of context data.

The diagram below shows the general architecture for delivery of IoT Big Data services. This is explained in the following narrative.

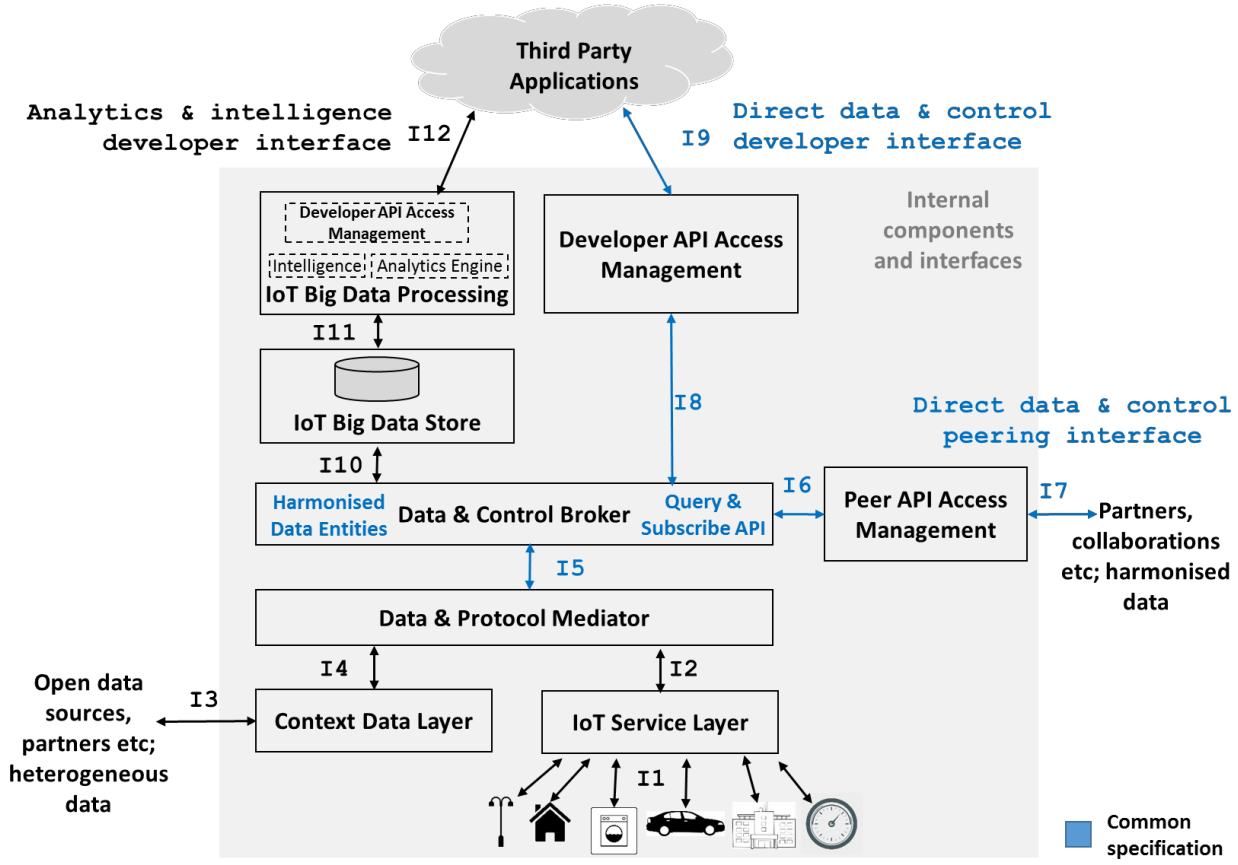


Figure 1: General Architecture for IoT Big Data

IoT Service Layer

The IoT service layer is concerned with handling the device specific interactions required for obtaining data from IoT devices and sending control commands (where

relevant) to those IoT devices. Therefore this layer is required to handle bi-directional communications both to IoT devices and to the upper layers of the architecture.

The IoT Service Layer is expected to handle the lower level interactions with IoT devices. Those devices might be connected using a variety of protocols and low level communication technologies including (but not limited to) oneM2M [3], Hypercat [4], Constrained Application Protocol (CoAP), MQ Telemetry Transport (MQTT), Real Time Streaming Protocol (RTSP), or device specific interfaces such as JavaScript Object Notation (JSON)/Extensible Markup Language (XML) over HTTP.

The IoT Service Layer is expected to handle authentication and security aspects regarding the interfacing with IoT devices.

Data and Protocol Mediator

The Data and Protocol Mediator is responsible for ingesting information from IoT devices as well as other external sources ("context data"). It ensures that data is transformed to the Harmonised Entity Definition before being stored and published by the Data & Control Broker. The harmonisation process itself may be partially implemented in the 'Context Data Layer' function or the 'IoT Service Layer' function but the Data & Protocol Mediator will ensure that harmonisation is complete before data is exposed to the higher level parts of the architecture.

The harmonisation process includes:

- Conversion of payload encoding e.g. converting between an XML format payload of the IoT or context data and the JSON based structures defined in the Harmonised Entity Definitions;
- Mapping of the data structures and data fields between the lower level IoT device structures and fields and the Harmonised Entity Definitions e.g. the IoT device might store a temperature value in a field named 'temp' whereas the Harmonised Entity Definition in a field named 'currentTemperature';
- Unit conversion from the values and ranges of the lower level IoT devices to the Harmonised Entity Definition e.g.
    - The Harmonised Entity Definition might represent a switch as the Boolean true or

false value whereas the IoT device could represent as the integer 1 or 0 respectively;

- The Harmonised Entity Definition might represent a temperature in degrees Centigrade as a double precision float whereas the IoT device might record in degrees Fahrenheit.

- Data quality verification e.g. identifying a situation where an IoT sensor is apparently faulty such as delivering a sensor reading which is outside of the expected range. For example an outdoor temperature sensor transmitting a value which is significantly outside of the normal weather temperature range;

- Combining ("mash up") or linking relevant context data with IoT data e.g. associating a specific IoT sensor with its geographic location or weather forecast data to form a richer entity definition;
- Cross referencing related entity data e.g. different sensors with say the car to which they belong.

The Data & Protocol Mediator will also enable control requests to be processed - performing broadly a 'reverse' process compared with data harmonisation:

- Verifying the control request to make sure that the request is valid e.g.

Refers to a valid IoT device;

- The control action is relevant to that IoT device e.g. a fixed position sensor cannot be asked to move to a different position;
- The control action is valid according to the current state of the device/ system (which should be maintained by the control broker);
- The parameter values supplied in the request are valid both in terms of individual parameter range and in combination.

- Transforming the high level control request into the equivalent device specific request payload

e.g. generating an XML format payload if that is required by the IoT device;

- Mapping of the data structures and data fields in the control request between the high level structures and fields of the Harmonised Entity Definitions and the lower level IoT device structures and fields;

Data & Control Broker

The Data & Control Broker is responsible for enabling third party application access to harmonised data entities through a query and subscribe API, allowing applications to gain access to such data in a standard way. The broker may store data in the short to medium term, coming from multiple devices and data sources via the Data and Protocol Mediator. This function also transforms control requests coming from the application layer to be passed onwards to the Data & Protocol Mediator.

The control process itself may be partially implemented in the 'IoT Service Layer' function but the Data & Control Broker in collaboration with the Data & Protocol Mediator will ensure responsibility for providing third party application access to control services in a consistent (harmonised) and controlled way. Control brokering will perform broadly a 'reverse' process compared with data harmonisation, receiving high level control requests from the third party application - normally formatted as a JSON based request communicated over HTTPS, and adapting this through the Data & Protocol Mediator and IoT Service Layer.

The Data & Control Broker is expected to have access to a data store which may act as a short to medium term buffer space for control actions or a short to medium term store for harmonised data entities. The expected use of this is:

- Retention of current instances of harmonised data entities processed from IoT devices and external sources (context data);
- Storage of control requests and any status information relevant to the request;
- Storage of a window of historical harmonised data entities that may be queried directly via the third party application. Note that it is expected that such a data store would be for short to medium term harmonised data entities, whereas longer term storage of harmonised data entities would be provided in the "IoT Big Data Store";
- Storage of any results of Analytics and Intelligence results which become additional context data that can be queried or mashed up with other IoT data or external data sources.

It should be noted that the Data & Control Broker has the option of using its own internal database for data storage or the defined IoT Big Data Store function defined in this

architecture i.e. some of the logically separate elements of the defined architecture may be physically implemented together.

Peer API Access Management

The Peer API Access Management function is responsible for interfacing with its peers in other organisations to receive and publish additional relevant harmonised IoT and context data. The policies applied to these trusted interfaces may be different to those applied to the main developer interface provided by the Developer API Access Management function. For example, organisations may be willing to share certain sensitive data with each other but require this sensitive data to be anonymised before being offered to third party developers. See sections 4.2.6 and 4.2.7 on I6 and I7 interfaces for more details.

Developer API Access Management

The Developer API Access Management function controls access to harmonised data entities, covering both IoT and context data, as well as control services to third party applications. It implements authentication, authorisation and access control using industry standards to ensure privacy and security around the harmonised data. This function is mainly concerned with managing the privacy and security aspects of the data access by external parties. It is expected that this layer does not perform any actual IoT and context data processing or storage but is ensuring that data and control services from lower layers of the architecture are delivered in a properly managed way. It is assumed that any data processing/ complex queries/ analytics/ intelligence is the responsibility of the third party application.

The Developer API Access Management function access control for the harmonised data, it is expected to perform the following:

- Be responsible for presenting data & control services to third party applications via a RESTful based API over http[3]. This interface shall use JSON based encoding of data using the Harmonised Entity Definitions for both data and control and use the NGSIv2 interface to support entity retrieval/ simple queries;

- Implement API access control (i.e. application level security) to ensure only known/ approved applications have access to IoT and context data and control services. Access control should be provided on a per application basis allowing granularity over which application should be able to access what IoT and context data and control functions;
- Implement any usage policies against applications accessing the APIs e.g. applying IP address based access rules or throttling rules to ensure there is relevant fair usage of the platform;
- Provide access to a publish/ subscribe service so that IoT and context data can be pushed to the third party application server as new data is received;
- Log API usage information e.g. number of API calls made by an application, number and type of entity data retrieved, number and type of control requests received.

IoT Big Data Store

The provision of Big Data Analytics and Intelligence is dependent on having access to the relevant mass of data from which the various insights can be obtained. This function provides data storage for this massive data and it may also provide short to medium term storage capabilities for use by the Data & Control Broker, depending on the specific implementation.

For IoT Big Data usage it is considered that the Data Store must be able to handle a data set greater than 50TB in size. For small scale deployments/ prototypes a Relational Database such as MySQL may support IoT data storage. However realistically a NoSQL or 'graph' database is considered more suitable for commercial 'Big Data' deployment particularly because the graph data model is richer and more versatile.

"Big Data" databases address needs such as:

- The need to store vast amounts of data (orders of magnitude higher than Relational Databases reasonably work to);
- Insights are obtained when exploring ad-hoc relationships between data;
- Data is arriving at such a rate that it is impossible to maintain an indexing process;
- Data are not tightly constrained into fixed table/ column formats.

The "Big Data" database is expected to be used to store the harmonised data entities received from the IoT devices and/or the external data sources. As it is expected there could be many millions of IoT devices generating data frequently, the required storage space may be vast (i.e. of the order of many terabytes to many petabytes of data). It is expected the "Big Data" database could be implemented using products such as Apache Cassandra, Apache Hadoop, MongoDB, Neo4j, Titan or DynamoDB. To achieve high performance the database component may employ substantial quantities of memory to hold copies of data that is persistently stored on "hard disk".[4]

IoT Big Data Processing

The processing of stored IoT data to perform analytics and intelligence is identified as the responsibility of the IoT Big Data Processing function. The IoT Big Data Processing function also provides related Developer API Access Management to control access to the intelligence and analytics by implementing authentication, authorisation and access control to ensure privacy and security. A broad division is made between analytics and intelligence. In practice both analytics and intelligence will be processing subsets of the mass of IoT data retained in the IoT Big Data Store. The main difference is

- Analytics - principally involves relatively conventional methods (by human analysts and normal programming techniques) of exploring links and statistical relationships between data and then the analytics engine will produce its output based on the execution of a defined process;
- Intelligence - encompassing machine learning / artificial intelligence, it would be expected that algorithms 'adapt' to the observed data and the match between predicted and desired outcomes.

The outputs from Analytics and Intelligence are expected to be in a wide range of different formats, many of which will not conform to a uniform 'API' based approach e.g. the generation of a PDF report or the generation of a data set to be FTP'd to the third party developer's platform.

Relevant products for Analytics & Intelligence provision include:

- **Apache Spark**

Apache Spark[15] is a powerful data processing system based upon Cassandra or Hadoop[16] for the data storage component and provides several powerful tools for building applications around it such as an SQL interface, graph data library and a job server.

Spark is not a complete solution out of the box, but does provide a powerful big data platform with great performance. Spark is considered the leading solution for high performance Big Data analytics.

A Spark solution could be delivered over a RESTful interface or a websockets connection (for better notification and real time services). More usually however developers would use the standard programming interfaces available to Java, Python, Scala and R programming languages.

- **Apache TinkerPop3 + Titan + Elastic Search + Gremlin**

Titan provides Casandra backends, integration with Elastic search[17], Apache Lucene[18] / Solr[19], Spark and others which allows it to support Geo searches, full text searches, graph traversals and regular 'SQLesque' queries making it ideal for the IoT Big Data project.

Apache TinkerPop3[20] is a graph computing framework which is seeing a rapid adoption in data driven applications. Many projects are seeking to incorporate the TinkerPop specification into their interfaces for interoperability in graph databases and servers. There are several implementations of graph databases which expose a Gremlin[21] querying interface which makes it easier to query the graph database. Two such databases are Titan and Google Cayley.

- **Apache Mahout**

Mahout[22] is designed for the development of high performance and scalable machine learning applications. It builds for example on top of Apache Spark / Hadoop and supports a range of machine learning algorithms. Uses include

Collaborative filtering – mines user behaviour and makes product recommendations (e.g. Amazon recommendations);
Clustering – takes items in a particular class (such as web pages or newspaper articles) and

organizes them into naturally occurring groups, such that items belonging to the same group are similar to each other;
Classification – learns from existing categorizations and then assigns unclassified items to the best category;
Frequent itemset mining – analyses items in a group (e.g. items in a shopping cart or terms in a query session) and then identifies which items typically appear together.

- **Tensorflow**

Another open source set of tools for machine learning - developed originally by the Google Brain Team to support advances in search ranking algorithms as well as other Google research activities.

Tensorflow[23] could be used for example in IoT applications such as developing high accuracy automated number plate recognition algorithms based on images captured from CCTV cameras. This can then be applied in the IoT Big Data system to applications such as security, congestion or traffic planning.

Tensorflow can also be coupled with Apache Spark which is used to obtain the select the data from the IoT Big Data store to use with the tensorflow algorithms.

2. Big Data Analytical Tools Classification

- Data Storage and Management
- Data Cleaning
- Data Mining
- Data Analysis

Data Storage and Management

Hadoop

Apache Hadoop[6] is a highly scalable storage platform designed to process very large data sets across hundreds to thousands of computing nodes that operate in parallel. It provides a very cost effective storage solution for large data volumes with no particular format

requirements. MapReduce [6] is the programming paradigm that allows for this massive scalability, is at the heart of Hadoop. The term MapReduce refers to two separate and distinct tasks that Hadoop programs perform. Hadoop has two main components - HDFS and YARN.

HDFS[7] – the Hadoop Distributed File System is a distributed file system designed to run on commodity hardware. It differs from other distributed file systems in that HDFS is highly fault- tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets.

YARN[8] - YARN is a large-scale, distributed operating system for big data applications that runs on top of HDFS. It provides a framework for job scheduling and cluster resource management.

Cassandra

Cassandra[5] is a scalable database for large scale data storage from the Apache foundation and is used by many of the world"s leading tech companies including github, Netflix, Apple and Instagram. The largest known deployment of Cassandra contains 75000 nodes (cloud servers) and stores over 10PB (Petabytes) of data. Cassandra is a NoSQL data store, which provides a robust means of storing data which spans many nodes, however it does not provide a very powerful query interface; it's highly inefficient to query on anything other than Cassandra's equivalent of a 'primary key'. Several solutions can be combined with Cassandra to provide a more powerful query interface. Apache Spark is one of the most powerful of these.

Cloudera

Cloudera is essentially a brand name for Hadoop with some extra services stuck on. They can help your business build an enterprise data hub, to allow people in your organization better access to the data you

are storing. While it does have an open source element, Cloudera is mostly and enterprise solution to help businesses manage their Hadoop ecosystem. Essentially, they do a lot of the hard work of administering Hadoop for you. They will also deliver a certain amount of data security, which is highly important if you're storing any sensitive or personal data.

MongoDB

MongoDB[9] is a hybrid open source and closed source database, where the core of the database is available freely on an open source license, although some features which may be required on larger commercial deployments are commercially supported add-ons. This model has made MongoDB arguably one of the most popular document oriented databases in use today. A 'document' in MongoDB is a 'binary' representation of a JSON document. This allows arbitrary JSON encoded data to be stored in the database and then queried using a rich JSON based querying interface.

2.1.4 Graph Databases

Other databases such as Neo4J[10] or Titan[11] are a powerful way for structuring data which allows for easily traversing relationships as well as retrieving attributes about a particular node. It is worth clarifying that a Graph Database works efficiently where there are ad-hoc relationships between data whereas a Relational Database is efficient for more structured relationships between data. The key strength of these systems is that they're very well adapted for traversing different data types to perform ad-hoc mash-ups.

Data Cleaning Tool

OpenRefine

OpenRefine (formerly GoogleRefine) is an open source tool that is dedicated to cleaning messy data. You can explore huge data sets easily and quickly even if the data is a little unstructured. As far as data softwares go, OpenRefine is pretty user-friendly. Though, a good knowledge of data cleaning principles certainly helps. The nice thing about OpenRefine is that it has a huge community with lots of contributors meaning that the software is constantly getting better and better.

Data Cleaner

DataCleaner recognises that data manipulation is a long and drawn out task. Data visualization tools can only read nicely structured, "clean" data sets. DataCleaner does the hard work for you and transforms messy semi-structured data sets into clean readable data sets that all of the visualization companies can read. DataCleaner also offers data warehousing and data management services. The company offers a 30- day free trial and then after that a monthly subscription fee.

Data Mining Tool

IBM SPSS Modeler

The IBM SPSS Modeler offers a whole suite of solutions dedicated to data mining. This includes text analysis, entity analytics, decision management and optimization. Their five products provide a range of advanced algorithms and techniques that include text analytics, entity analytics, decision management and optimization. SPSS Modeler is a heavy-duty solution that is well suited for the needs of big companies. It can run on virtually any type of database and you can integrate it with other IBM SPSS products such as SPSS collaboration and deployment services and the SPSS Analytic server.

Oracle data mining

Another big hitter in the data mining sphere is Oracle. As part of their Advanced Analytics Database option, Oracle data mining allows its users to discover insights, make predictions and leverage their Oracle data. You can build models to discover customer behavior, target best customers and develop profiles. The Oracle Data Miner GUI enables data analysts, business analysts and data scientists to work with data inside a database using a rather elegant drag and drop solution. It can also create SQL and PL/SQL scripts for automation, scheduling and deployment throughout the enterprise.

FramedData

If you‟re after a specific type of data mining there are a bunch of startups which specialize in helping businesses answer tough questions with data. If you‟re worried about user churn, we recommend FramedData, a startup which analyzes your

analytics and tell you which customers are about to abandon your product.

Data Analysis Tool

Qubole

Qubole simplifies, speeds and scales big data analytics workloads against data stored on AWS, Google, or Azure clouds. They take the hassle out of infrastructure wrangling. Once the IT policies are in place, any number of data analysts can be set free to collaboratively "click to query" with the power of Hive, Spark, Presto and many others in a growing list of data processing engines. Qubole is an enterprise level solution. They offer a free trial that you can sign up to at this page.The flexibility of the program really does set it apart from the rest as well as being the most accessible of the platforms.

BigML

BigML is attempting to simplify machine learning. They offer a powerful Machine Learning service with an easy-to-use interface for you to import your data and get predictions out of it. You can even use their models for predictive analytics. A good understanding of modeling is certainly helpful, but not essential, if you want to get the most from BigML. They have a free version of the tool that allows you to create tasks that are under 16mb as well as having a pay as you go plan and a virtual private cloud that meet enterprise-grade requirements.

Statwing

Statwing takes data analysis to a new level providing everything from beautiful visuals to complex analysis. They have a particularly cool blog post on NFL data! It's so simple to use that you can actually get started with Statwing in under 5 minutes. This allows you to use unlimited datasets of up to 50mb in size each. There are other enterprise plans that give you the ability to upload bigger datasets.

3 Data Exploration

Data exploration is an informative search used by data consumers to form true analysis from the information gathered. Often, data is gathered in a non-rigid or controlled manner in large bulks. For true analysis, this unorganized bulk of data needs to be narrowed down.

This is where data exploration is used to analyze the data and information from the data to form further analysis.

Data often converges in a central warehouse called a data warehouse. This data can come from various sources using various formats. Relevant data is needed for tasks such as statistical reporting, trend spotting and pattern spotting. Data exploration is the process of gathering such relevant data.

Below are the steps involved to understand, clean and prepare your data for building your predictive model:

1. Variable Identification
2. Univariate Analysis
3. Bi-variate Analysis
4. Missing values treatment
5. Outlier treatment
6. Variable transformation
7. Variable creation

Variable Identification

First, identify **Predictor** (Input) and **Target** (output) variables. Next, identify the data type and category of the variables. Let's understand this step more clearly by taking an example.

Example:- Suppose, we want to predict, whether the students will play cricket or not (refer below data set). Here you need to identify predictor variables, target variable, data type of variables and category of variables.

Student_ID	Gender	Prev_Exam_Marks	Height (cm)	Weight Caregory (kgs)	Play Cricket
S001	M	65	178	61	1
S002	F	75	174	56	0
S003	M	45	163	62	1
S004	M	57	175	70	0
S005	F	59	162	67	0

Below, the variables have been defined in different category:

Figure 2: Variable Identification

Univariate Analysis

At this stage, we explore variables one by one. Method to perform uni-variate analysis will depend on whether the variable type is categorical or continuous. Let's look at these methods and statistical measures for categorical and continuous variables individually:

Continuous Variables:- A **continuous variable** is a **variable** that has an infinite number of possible values. In other words, any value is possible for the **variable**.

Categorical Variables:- A categorical variable (sometimes called a nominal variable) is one that has two or more categories, but there is no intrinsic ordering to the categories. For example, gender is a categorical variable having two categories (male and female) and there is no intrinsic ordering to the categories. Hair color is also a categorical variable having a number of categories (blonde, brown, brunette, red, etc.) and again, there is no agreed way to order these from highest to lowest.

Bi-variate Analysis

Bi-variate Analysis finds out the relationship between two variables. Here, we look for association and disassociation between variables at a pre-defined significance level. We can perform bi-variate analysis for any combination of categorical and continuous variables. The combination can be: Categorical & Categorical, Categorical & Continuous and Continuous & Continuous. Different methods are used to tackle these combinations during analysis process.

Continuous & Continuous: While doing bi-variate analysis between two continuous variables, we should look at scatter plot. It is a nifty way to find out the relationship between two variables. The pattern of scatter plot indicates the relationship between variables. The relationship can be linear or non-linear.

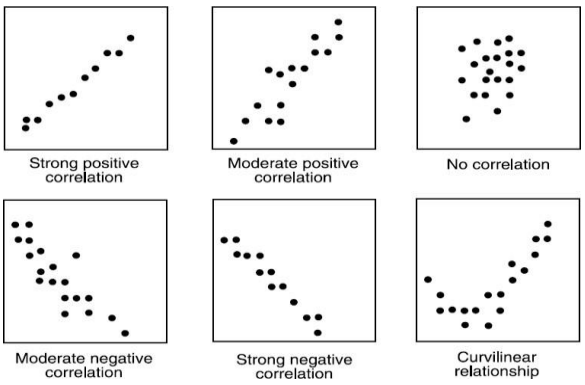


Figure 3: Scatter plot indicates Relationship

Scatter plot shows the relationship between two variable but does not indicates the strength of relationship amongst them. To find the strength of the relationship, we use Correlation. Correlation varies between -1 and +1.

- -1: perfect negative linear correlation
- +1:perfect positive linear correlation and
- 0: No correlation

Correlation can be derived using following formula:

Correlation = Covariance(X,Y) / SQRT(Var(X)* Var(Y))

Various tools have function or functionality to identify correlation between variables. In Excel, function CORREL() is used to return the correlation between two variables and SAS uses procedure PROC CORR to identify the correlation. These function returns Pearson Correlation value to identify the relationship between two variables:

X	65	72	78	65	72	70	65	68
Y	72	69	79	69	84	75	60	73

Metrics	Formula	Value
Co-Variance (X,Y)	=COVAR(E6:L6,E7:L7)	18.77
Variance (X)	=VAR.P(E6:L6)	18.48
Variance (Y)	=VAR.P(E7:L7)	45.23
Correlation	=G10/SQRT(G11*G12)	0.65

Figure 4: Correlation

Categorical & Categorical: To find the relationship between two categorical variables, we can use following methods:

- **Two-way table:** We can start analyzing the relationship by creating a two-way table of count and count%. The rows represents the category of one variable and the columns represent the categories of the other variable. We show count or count% of observations available in each combination of row and column categories.
- **Stacked Column Chart:** This method is more of a visual form of Two-way table.

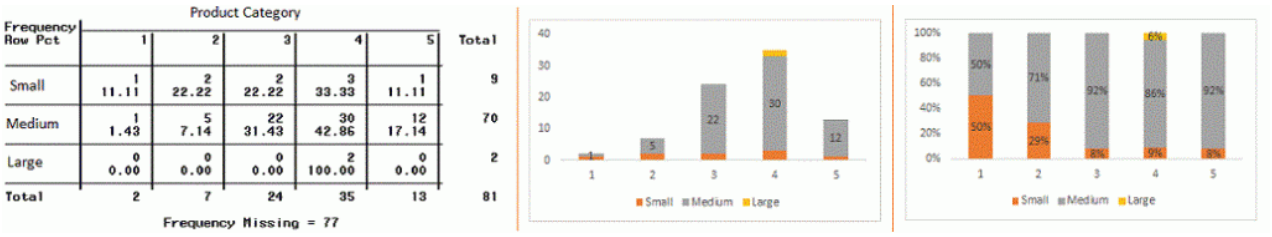


Figure 5: Two-way table and Stacked Column Chart

- **Chi-Square Test:** This test is used to derive the statistical significance of relationship between the variables. Also, it tests whether the evidence in the sample is strong enough to generalize that the relationship for a larger population as well. Chi-square is based on the difference between the expected

and observed frequencies in one or more categories in the two-way table. It returns probability for the computed chi-square distribution with the degree of freedom.

Categorical & Continuous: While exploring relation between categorical and continuous variables, we can draw box plots for each level of categorical variables. If levels are small in number, it will not show the statistical significance. To look at the statistical significance we can perform Z-test, T-test or ANOVA.

- **Z-Test/ T-Test:-** Either test assess whether mean of two groups are statistically different from each other or not.

Example: Suppose, we want to test the effect of five different exercises. For this, we recruit 20 men and assign one type of exercise to 4 men (5 groups). Their weights are recorded after a few weeks. We need to find out whether the effect of these exercises on them is significantly different or not. This can be done by comparing the weights of the 5 groups of 4 men each. Till here, we have understood the first three stages of Data Exploration, Variable Identification, Uni-Variate and Bi-Variate analysis. We also looked at various statistical and visual methods to identify the relationship between variables. Now, we will look at the methods of Missing values Treatment. More importantly, we will also look at why missing values occur in our data and why treating them is necessary.

Missing Value Treatment

Missing data in the training data set can reduce the power / fit of a model or can lead to a biased model because we have not analysed the behavior and relationship with other variables correctly. It can lead to wrong prediction or classification

Name	Weight	Gender	Play Cricket/ Not
Mr. Amit	58	M	Y
Mr. Anil	61	M	Y
Miss Swati	58	F	N
Miss Richa	55		Y
Mr. Steve	55	M	N
Miss Reena	64	F	Y
Miss Rashmi	57		Y
Mr. Kunal	57	M	N

Gender	#Students	#Play Cricket	%Play Cricket
F	2	1	50%
M	4	2	50%
Missing	2	2	100%

Name	Weight	Gender	Play Cricket/ Not
Mr. Amit	58	M	Y
Mr. Anil	61	M	Y
Miss Swati	58	F	N
Miss Richa	55	F	Y
Mr. Steve	55	M	N
Miss Reena	64	F	Y
Miss Rashmi	57	F	Y
Mr. Kunal	57	M	N

Gender	#Students	#Play Cricket	%Play Cricket
F	4	3	75%
M	4	2	50%

Figure 6: Missing Value Treatment

Notice the missing values in the image shown above: In the left scenario, we have not treated missing values. The inference from this data set is that the chances of playing cricket by males is higher than females. On the other hand, if you look at the second table, which shows data after treatment of missing values (based on gender), we can see that females have higher chances of playing cricket compared to males. Now, let's identify the reasons for occurrence of these missing values. They may occur at two stages:

1. **Data Extraction**: It is possible that there are problems with extraction process. In such cases, we should double-check for correct data with data guardians. Some hashing procedures can also be used to make sure data extraction is correct. Errors at data extraction stage are typically easy to find and can be corrected easily as well.
2. **Data collection**: These errors occur at time of data collection and are harder to correct. They can be categorized in four types:
   - **Missing completely at random:** This is a case when the probability of missing variable is same for all observations. For example: respondents of data collection process decide that they will declare their earning after tossing a fair coin. If an head occurs, respondent declares his / her earnings & vice versa. Here each observation has equal chance of missing value.
   - **Missing at random:** This is a case when variable is missing at random and missing ratio varies for different values / level of other input variables. For example: We are collecting data for age and female has higher missing value compare to male.
   - **Missing that depends on unobserved predictors:** This is a case when the missing values are not random and are related to the unobserved input variable. For example: In a medical study, if a particular diagnostic causes discomfort, then there is higher chance of drop out from the study. This missing value is not at random unless we have included "discomfort" as an input variable for all patients.
   - **Missing that depends on the missing value itself:** This is a case when the probability of missing value is directly correlated with missing value itself. For example: People with higher or lower income are likely to provide non-response to their earning.

Methods to treat missing values

1. **Deletion:** It is of two types: List Wise Deletion and Pair Wise Deletion.

- In list wise deletion, we delete observations where any of the variable is missing. Simplicity is one of the major advantage of this method, but this method reduces the power of model because it reduces the sample size.

- In pair wise deletion, we perform analysis with all cases in which the variables of interest are present. Advantage of this method is, it keeps as many cases available for analysis. One of the disadvantage of this method, it uses different sample size for different variable

List wise deletion

Gender	Manpower	Sales
M	25	343
F	.	280
M	33	332
M	.	272
F	25	.
M	29	326
	26	259
M	32	297

Pair wise deletion

Gender	Manpower	Sales
M	25	343
F	.	280
M	33	332
M	.	272
F	25	.
M	29	326
	26	259
M	32	297

Figure 7: List Wise Deletion and Pair Wise Deletion

- Deletion methods are used when the nature of missing data is "**Missing completely at random**" else non random missing values can bias the model output.

2. **Mean/ Mode/ Median Imputation:** Imputation is a method to fill in the missing values with estimated ones. The objective is to employ known relationships that can be identified in the valid values of the data set to assist in estimating the missing values. Mean / Mode / Median imputation is one of the most frequently used methods. It consists of replacing the missing data for a given attribute by the mean or median (quantitative

attribute) or mode (qualitative attribute) of all known values of that variable. It can be of two types:-

Generalized Imputation: In this case, we calculate the mean or median for all non missing values of that variable then replace missing value with mean or median. Like in above table, variable "**Manpower"** is missing so we take average of all non missing values of "**Manpower" (28.33)** and then replace missing value with it.

Similar case Imputation: In this case, we calculate average for gender "**Male"** (29.75) and "**Female**" (25) individually of non missing values then replace the missing value based on gender. For "**Male**", we will replace missing values of manpower with 29.75 and for "**Female**" with 25.

3. **Prediction Model**: Prediction model is one of the sophisticated method for handling missing data. Here, we create a predictive model to estimate values that will substitute the missing data. In this case, we divide our data set into two sets: One set with no missing values for the variable and another one with missing values. First data set become training data set of the model while second data set with missing values is test data set and variable with missing values is treated as target variable. Next, we create a model to predict target variable based on other attributes of the training data set and populate missing values of test data set.We can use regression, ANOVA, Logistic regression and various modeling technique to perform this. There are 2 drawbacks for this approach:

1. The model estimated values are usually more well-behaved than the true values
2. If there are no relationships with attributes in the data set and the attribute with missing values, then the model will not be precise for estimating missing values.

KNN Imputation: In this method of imputation, the missing values of an attribute are imputed using the given number of attributes that are most similar to the attribute whose values are missing. The similarity of two attributes is determined using a distance function. It is also known to have certain advantage & disadvantages.

Techniques of Outlier Detection and Treatment

Outlier is a commonly used terminology by analysts and data scientists as it needs close attention else it can result in wildly wrong estimations. Simply speaking, Outlier is an

observation that appears far away and diverges from an overall pattern in a sample.

Let's take an example, we do customer profiling and find out that the average annual income of customers is $0.8 million. But, there are two customers having annual income of $4 and $4.2 million. These two customers annual income is much higher than rest of the population. These two observations will be seen as Outliers.

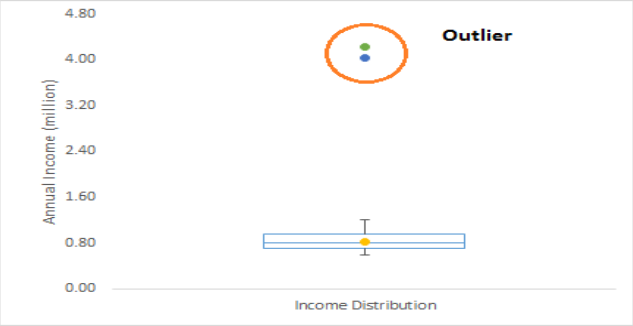


Figure 8: outlier

Types of Outliers

Outlier can be of two types: **Univariate** and **Multivariate**. Above, we have discussed the example of univariate outlier. These outliers can be found when we look at distribution of a single variable. Multi-

variate outliers are outliers in an n-dimensional space. In order to find them, you have to look at distributions in multi-dimensions.

Let us understand this with an example. Let us say we are understanding the relationship between height and weight. Below, we have univariate and bivariate distribution for Height, Weight. Take a look at the box plot. We do not have any outlier (above and below 1.5*IQR, most common method). Now look at the scatter plot. Here, we have two values below and one above the average in a specific segment of weight and height.

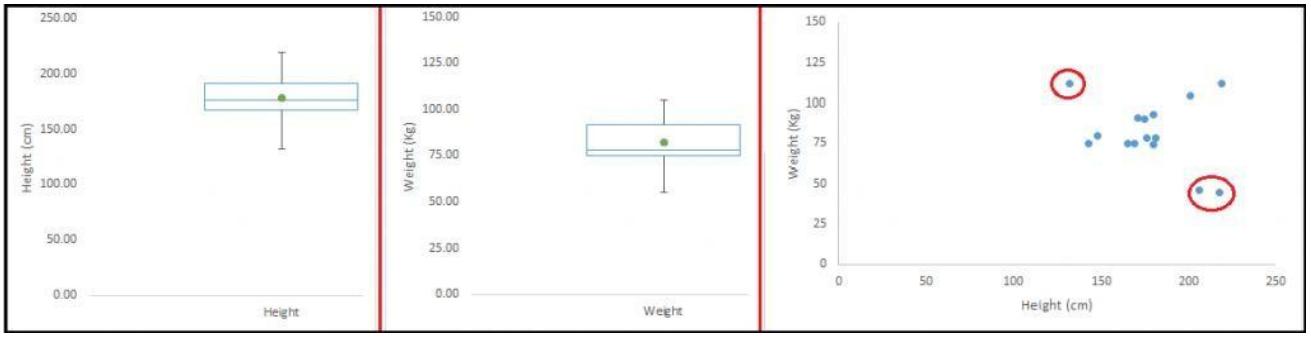


Figure 9: Types of outliers

- **Data Entry Errors:-** Human errors such as errors caused during data collection, recording, or entry can cause outliers in data. For example: Annual income of a customer is $100,000. Accidentally, the data entry operator puts an additional zero in the figure. Now the income becomes $1,000,000 which is 10 times higher. Evidently, this will be the outlier value when compared with rest of the population.
- **Measurement Error:** It is the most common source of outliers. This is caused when the measurement instrument used turns out to be faulty. For example: There are 10 weighing machines. 9 of them are correct, 1 is faulty. Weight measured by people on the faulty machine will be higher / lower than the rest of people in the group. The weights measured on faulty machine can lead to outliers.
- **Experimental Error:** Another cause of outliers is experimental error. For example: In a 100m sprint of 7 runners, one runner missed out on concentrating on the „Go" call which caused him to start late. Hence, this caused the runner"s run time to be more than other runners. His total run time can be an outlier.

 Intentional Outlier*:* This is commonly found in self-reported measures that involves sensitive data. For example: Teens would typically under report the amount of alcohol that they consume. Only a fraction of them would report actual value. Here actual values might look like outliers because rest of the teens are under reporting the consumption.

 Data Processing Error: Whenever we perform data mining, we extract data from multiple sources. It is possible that some manipulation or extraction errors may lead to outliers in the dataset.

 Sampling error: For instance, we have to measure the height of athletes. By mistake, we include a few basketball players in the sample. This inclusion is likely to cause outliers in the dataset.

 Natural Outlier: When an outlier is not artificial (due to error), it is a natural outlier. For instance: In my last assignment with one of the renowned insurance company, I noticed that the performance of top 50 financial advisors was far higher than rest of the population. Surprisingly, it was not due to any error. Hence, whenever we perform any data mining activity with advisors, we used to treat this segment separately.

Detect Outliers

Most commonly used method to detect outliers is visualization. We use various visualization methods, like **Box-plot**, **Histogram**, **Scatter Plot** (above, we have used box plot and scatter plot for visualization). Some analysts also various thumb rules to detect outliers. Some of them are:

- Any value, which is beyond the range of -1.5 x IQR to 1.5 x IQR
- Use capping methods. Any value which out of range of 5th and 95th percentile can be considered as outlier
- Data points, three or more standard deviation away from mean are considered outlier
- Outlier detection is merely a special case of the examination of data for influential data points and it also depends on the business understanding

Remove Outliers

Most of the ways to deal with outliers are similar to the methods of missing values like deleting observations, transforming them, binning them, treat them as a separate group, imputing values and other statistical methods. Here, we will discuss the common techniques used to deal with outliers:

Deleting observations: We delete outlier values if it is due to data entry error, data processing error or outlier observations are very small in numbers. We can also use trimming at both ends to remove outliers.

Transforming and binning values: Transforming variables can also eliminate outliers. Natural log of a value reduces the variation caused by extreme values. Binning is also a form of variable transformation. Decision Tree algorithm allows to deal with outliers well due to binning of variable. We can also use the process of assigning weights to different observations.

Imputing: Like imputation of missing values, we can also impute outliers. We can use mean, median, mode imputation methods. Before imputing values, we should analyse if it is natural outlier or artificial. If it is artificial, we can go with imputing values. We can also use statistical model to predict values of outlier observation and after that we can impute it with predicted values.

Treat separately: If there are significant number of outliers, we should treat them separately in the statistical model. One of the approach is to treat both groups as two different groups and build individual model for both groups and then combine the output.

Feature engineering is the science (and art) of extracting more information from existing data. You are not adding any new data here, but you are actually making the data you already have more useful. For example, let's say you are trying to predict foot fall in a shopping mall based on dates. If you try and use the dates directly, you may not be able to extract meaningful insights from the data. This is because the foot fall is less affected by the day of the month than it is by the day of the week. Now this information about day of week is implicit in your data. You need to bring it out to make your model better.

Variable Transformation

In data modelling, transformation refers to the replacement of a variable by a function. For instance, replacing a variable x by the square / cube root or logarithm x is a transformation. In other words, transformation is a process that changes the distribution or relationship of a variable with others.

Let"s look at the situations when variable transformation is useful.

Methods of Variable Transformation

There are various methods used to transform variables. As discussed, some of them include square root, cube root, logarithmic, binning, reciprocal and many others. Let"s look at these methods in detail by highlighting the pros and cons of these transformation methods.

- **Logarithm:** Log of a variable is a common transformation method used to change the shape of distribution of the variable on a distribution plot. It is generally used for reducing right skewness of variables. Though, It can"t be applied to zero or negative values as well.

- **Square / Cube root:** The square and cube root of a variable has a sound effect on variable distribution. However, it is not as significant as logarithmic transformation. Cube root has its own advantage. It can be applied to negative values including zero. Square root can be applied to positive values including zero.

- **Binning:** It is used to categorize variables. It is performed on original values, percentile or frequency. Decision of categorization technique is based on business understanding. For example, we can

categorize income in three categories, namely: High, Average and Low. We can also perform co-variate binning which depends on the value of more than one variables.

Feature / Variable Creation

Feature / Variable creation is a process to generate a new variables / features based on existing variable(s). For example, say, we have date(dd-mm-yy) as an input variable in a data set. We can generate new variables like day, month, year, week, weekday that may have better relationship with target variable. This step is used to highlight the hidden relationship in a variable:

Emp_Code	Gender	Date	New_Day	New_Month	New_Year
A001	Male	21-Sep-11	21	9	2011
A002	Female	27-Feb-13	27	2	2013
A003	Female	14-Nov-12	14	11	2012
A004	Male	07-Apr-13	7	4	2013
A005	Female	21-Jan-11	21	1	2011
A006	Male	26-Apr-13	26	4	2013
A007	Male	15-Mar-12	15	3	2012

Figure 10: Creating derived variables

There are various techniques to create new features. Let's look at the some of the commonly used methods:

- **Creating derived variables:** This refers to creating new variables from existing variable(s) using set of functions or different methods. Let's look at it through "**Titanic – Kaggle competition**". In this data set, variable age has missing values. To predict missing values, we used the salutation (Master, Mr, Miss, Mrs) of name as a new variable. How do we decide which variable to create? Honestly, this depends on business understanding of the analyst, his curiosity and the set of hypothesis he might have about the problem. Methods such as taking log of variables, binning variables and other methods of variable transformation can also be used to create new variables.

- **Creating dummy variables:** One of the most common application of dummy variable is to convert categorical variable into numerical variables. Dummy

variables are also called Indicator Variables. It is useful to take categorical variable as a predictor in statistical models. Categorical variable can take values 0 and 1. Let's take a variable „gender". We can produce two variables, namely, "**Var_Male**" with values 1 (Male) and 0 (No male) and "**Var_Female**" with values 1 (Female) and 0 (No Female). We can also create dummy variables for more than two classes of a categorical variables with n or n-1 dummy variables.

Emp_Code	Gender	Var_Male	Var_Female
A001	Male	1	0
A002	Female	0	1
A003	Female	0	1
A004	Male	1	0
A005	Female	0	1
A006	Male	1	0
A007	Male	1	0

Figure 11: Creating dummy variables

4 Data Visualization

Data visualization is the visual and interactive exploration and graphic representation of data of any size, type (structured and unstructured) or origin. Visualizations help people see things that were not obvious to them before. Even when data volumes are very large, patterns can be spotted quickly and easily. Visualizations convey information in a universal manner and make it simple to share ideas with others.

It's a way to get fast insights through visual exploration, robust reporting and flexible information sharing. It helps a wide variety of users to make sense of the increasing amount of data within your organization. It's a way to present big data in a way business users can quickly understand and use. Data visualization brings the story of your data to life.

Data visualization can be used for different purposes.

Some examples:

- To create and share meaningful reports with anyone anywhere
- To forecast and quickly identify opportunities and anticipate future trends

- To optimize corporate processes & to drive innovation
- To give anyone in the organization the power to visually explore and analyze all available data.

Data visualization was created to visually explore and analyze data quickly. It"s designed for anyone in your organization who wants to use and derive insights from data regardless of analytic skill level – from influencers, decision makers and analysts to statisticians and data scientists. It also offers IT an easy way to protect and manage data integrity and security. The amount of data will continue to grow while often time and resources to interpret the data continue to decrease. Data visualization will become one of the few tools able to help us win that challenge.
Data visualization helps uncover insights buried in your data and discover trends within your business and the market that affect your bottom line. Insights in your data can provide competitive advantage and the opportunity to differentiate.

Data visualization lets you read your market intelligently, compare your overall position with the industry trend, define the most appreciated features of your products and adapt development accordingly, combine information about sales with consumer preferences and much more.
Data visualization allows you to spot market trends and grow your business Data visualization allows you now your market"s dynamics like never before Knowing your customer better leads to more effective sales and marketing actions and enhances customer experience. Data visualization allows you to know your customers" needs and act on it. Data visualization provides information that is easy to understand and to share. Company KPIs are always under control. Data from a variety of internal and external sources is channeled into one single, shared source of information.

Big Data visualization tool must be able to deal with semi-structured and unstructured data because big data usually have this type of format. It is realized that to cope with such huge amount of data there is need for immense parallelization, which is a challenge in visualization. The challenge in parallelization algorithm is to break down the problem into such independent task that they can run independently. The task of big data visualization is to recognize interesting patterns and correlations. We need to carefully choose the dimensions of data to be visualized, if we reduce

dimensions to make our visualization low then we may end up losing interesting patterns but if we use all the dimensions we may end up having visualization too dense to be useful to the users. For example: “Given the conventional displays (1.3 million pixels), visualizing every data point can lead to over-plotting, overlapping and may overwhelm user"s perceptual and cognitive capacities

Due to vast volume and high magnitude of big data it becomes difficult to visualize. Most of the current visualization tool have low performance in scalability, functionality and response time
.Methods have been proposed which not only visualizes data but processes at the same time. These methods use Hadoop and storage solution and R programming language as compiler environment in the model

Visualization Tools

Various tools have emerged to help us out from the above pointed problems. The most important feature that a visual- ization must have is that it should be interactive, which means that user should be able to interact with the visualization. Visualization must display relevant information when hovered over it, zoom in and out panel should be there, visualization should adapt itself at runtime if we select subset or superset of data. We reviewed some of the most popular visualization tools.

Tableau

Tableau is interactive data visualization tool which is focused on Business Intelligence. Tableau provides very wide range of visualization options. It provides option to create custom visualization. It is fast and flexible. It supports mostly all the data format and connection to various servers right from the Amazon Aurora to Cloudera Hadoop and Salesforce. User interface is intuitive, wide variety of charts are available. For simple calculations and statistics one does not require any coding skills but for heavy analytics we can run models in R and then import the results into Tableau. This requires quite a bit of programming skill based upon the task we need to perform.

Some other important big data visualization problems are as follows

Visual noise: Most of the objects in dataset are too relative to each other. It becomes very difficult to separate them.

Information loss: To increase the response time we can reduce data set visibility, but this leads to information loss.

Large image perception: Even after achieving desired me- chanical output we are limited by our physical perception.

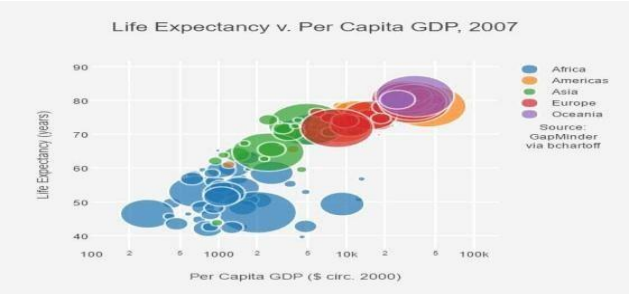
Microsoft Power BI

Power BI is a powerful cloud-base business analytics service. Visualization are interactive and rich. Power BI consists of 3 elements, Power BI Desktop, Service(SaaS), Apps. Every service is available to us that is why it makes Power BI flexible and persuasive. With more than 60 types of source integration you can start creating visualization in matter of minutes. Power BI combines the familiar Microsoft tools like Office, SharePoint and SQL Server. The feature that it distinguishes from other tools is that you can use natural language to query the data. You don't require programming skills for this tool but there is option available to run your R script. You can merge multiple data sources and create models, which comes in handy. Fig. 4 represents 3 visualizations in 3 coordinates, i.e. left, bottom and right. Left represents profit by county and market, bottom represents profit by region and right coordinate represents all over sales and profit.

Figure 12: Microsoft Power BI

Plotly

Plotly is also known as Plot.ly is build using python and Django framework. The actions it can perform are analyzing and visualizing data. It is free for users but with limited features, for all the features we need to buy the professional membership. It creates charts and dashboards online but can be used as offline service inside Ipython notebook, jupyter notebook and panda. Different variety of charts are available like statistical chart, scientific charts, 3D charts, multiple axes, dashboards etc. Plotly uses a tool called "Web Plot Digitizer(WPD)" which automatically grabs the data from the static image .Plotly on premises service is



also available, it is like plot.ly cloud but you host data on your private cloud behind your own firewall. This for those wo have concern about the privacy of their data. Python, R, MATLAB and Julia APIs are available for the same.

Figure 13: Plotly

Gephi

Gephi is open-source network analysis tool writ- ten in Java and OpenGL. It is used to handle very large and complex datasets. The network analysis includes

- Social Network Analysis
- Link Analysis
- Biological Network Analysis

With its dynamic data exploration Gephi stands out rest of its competition for graph analysis. No programming skills are required to run thin tools but a good knowledge

in graphs is necessary. It uses GPU 3D render engine to accelerate the performance and give real time analysis

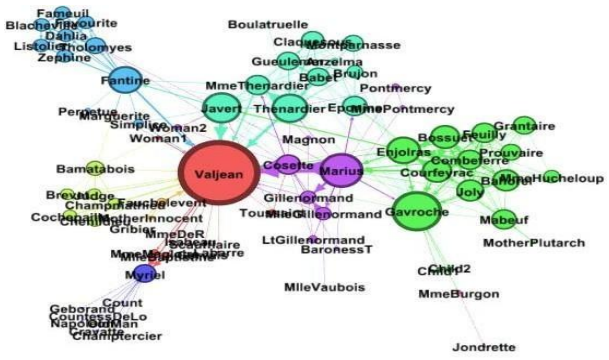


Figure 14: Gephi

Excel 2016

Microsoft Excel is a spreadsheet developed by Microsoft. It can not only be used for Big Data and statistical analysis but it is also a powerful visualization tool. Using power query excel can connect to most of the services like HDFS, SaaS etc and is capable of managing Semi- Structured data. Combined with visualization techniques like ”Conditional Formatting” and interactive graphs makes Excel 2016 a good contender in the ocean of Big Data visualization tools.

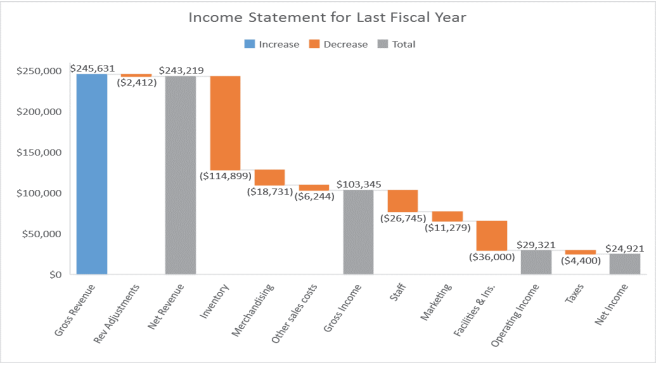


Figure 15: Excel

Oracle Visual Analyzer

Introduced in 2015, this web-based tool within the Oracle Business Intelligence Cloud Service claimed a spot at the Magic Quadrant Business Intelligence and Analytics Platform report by Gartner. Interactive visuals and highly advanced analysis clubbed with a customizable dashboard are some of the key features of Oracle Visual Analyzer. Being highly scalable, this data visualization tool is very suitable for enterprises with large-scale deployments where deep insights and well curated reports are essential.

Every bit of data carries a story with it and these data visualization tools are the gateway to fathom the story it tries to tell us. It helps us to understand about the current statistics and the future trends of the market.

Datawrapper

Datawrapper is a data visualization tool that"s gaining popularity fast, especially among media companies which use it for presenting statistics and creating charts. It has an easy to navigate user interface where you can easily upload a csv file to create maps, charts and visualizations that can be quickly added to reports. Although the tool is primarily aimed at journalists, it"s flexibility should accommodate a host of applications apart from media usage.

Google Chart

Google is an obvious benchmark and well known for the user-friendliness offered by its products and Google chart is not an exception. It is one of the easiest tools for visualizing huge data sets. Google chart holds a wide range of chart gallery, from a simple line graph to complex hierarchical tree-like structure and you can use any of them that fits your requirement. Moreover, the most important part while designing a chart is customization and with Google charts, it"s fairly Spartan. You can always ask for some technical help if you want to dig deep. It renders the chart in HTML5/SVG format and it is cross- browser compatible. Added to this, it also has adopted VML for supporting old IE browsers and that"s also cross-platform compatible, portable to iOS and the new release of Android. The chart data can be easily exported to PNG format.

Qlikview

Qlik is one of the major players in the data analytics space with their Qlikview tool which is also one of the biggest competitors of Tableau. Qlikview boasts over 40,000 customers spanning across over 100 countries. Qlik is particularly known for its highly customizable setup and a host of features that help create the visualizations much faster. However, the available options could mean there would be a learning curve to get accustomed with the tool so as to use it to its full potential.

Apart from its data visualization prowess, Qlikview also offers analytics, business intelligence and enterprise reporting features. The clean and clutter-free user experience is one of the notable aspects of Qlikview. Qliksense is a sister package of Qlikview which is often used alongside the former to aid in data exploration and discovery. Another advantage of using Qlikview is the strong community of users and resources which will help you get started with the tool.