

UNIT-4

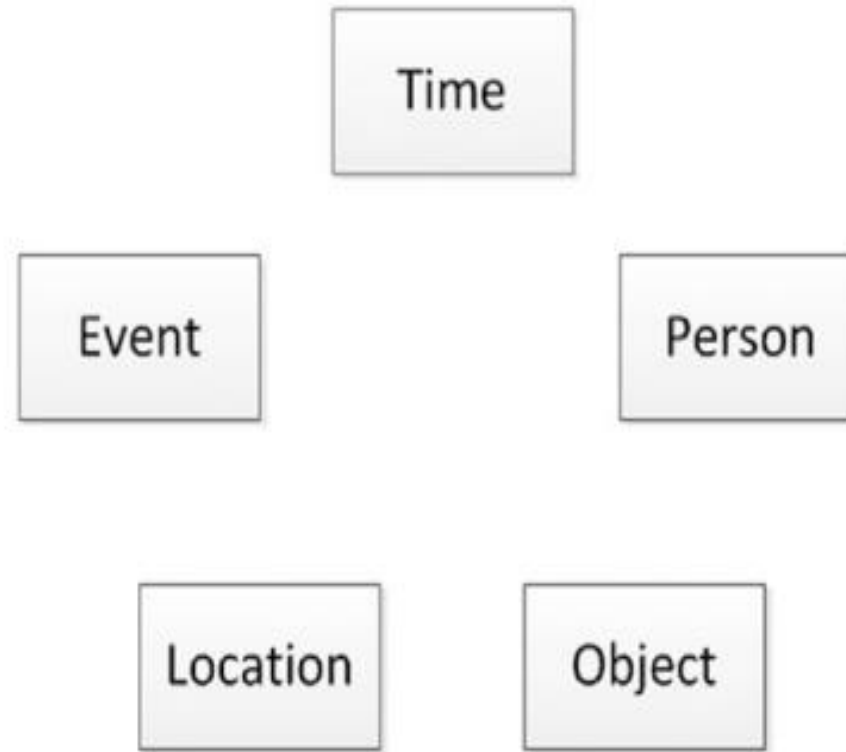
DATASCIENCE

PROCESS SUPERSTEP

- ▶ The Process super step adapts the assess results of the retrieve versions of the data sources into a highly structured data vault that will form the basic data structure for the rest of the data science steps.
- ▶ It basically contains all the processing chains for building the data vault.

PROCESS SUPERSTEP

- ▶ The Process superstep is the amalgamation process that pipes your data sources into five main categories of data .



. *Five categories of data*

Using only these five hubs in your data vault, and with good modeling, you can describe most activities of your customers.

DATA VAULT

- ▶ Data vault modeling is a database modeling method designed by Dan Linstedt.
- ▶ The data structure is designed to be responsible for long-term historical storage of data from multiple operational systems.

HUBS

- ▶ Each hub represents a core business concept, such as they represent Customer Id/Product Number/Vehicle identification number (VIN).
- ▶ Users will use a business key to get information about a Hub.
- ▶ The business key may have a combination of business concept ID and sequence ID, load date, and other metadata information.

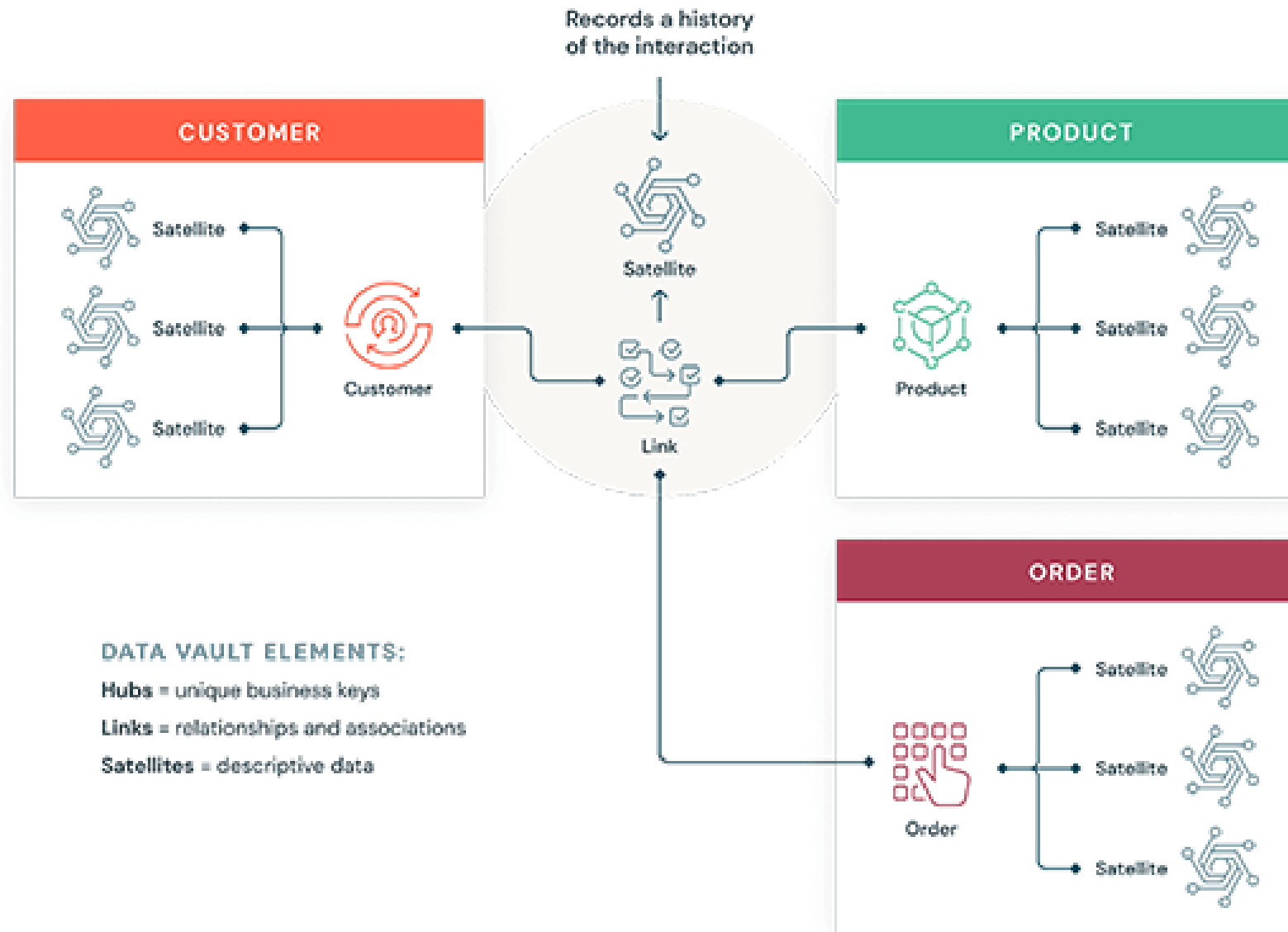
LINKS

- ▶ Links represent the relationship between Hub entities.

SATELLITES

- ▶ Data vault satellites stores the chronological descriptive and characteristics for a specific section of business data.
- ▶ Using hub and links we get model structure but no chronological characteristics.
- ▶ Satellites consist of characteristics and metadata linking them to their specific hub.

Data vault modeling



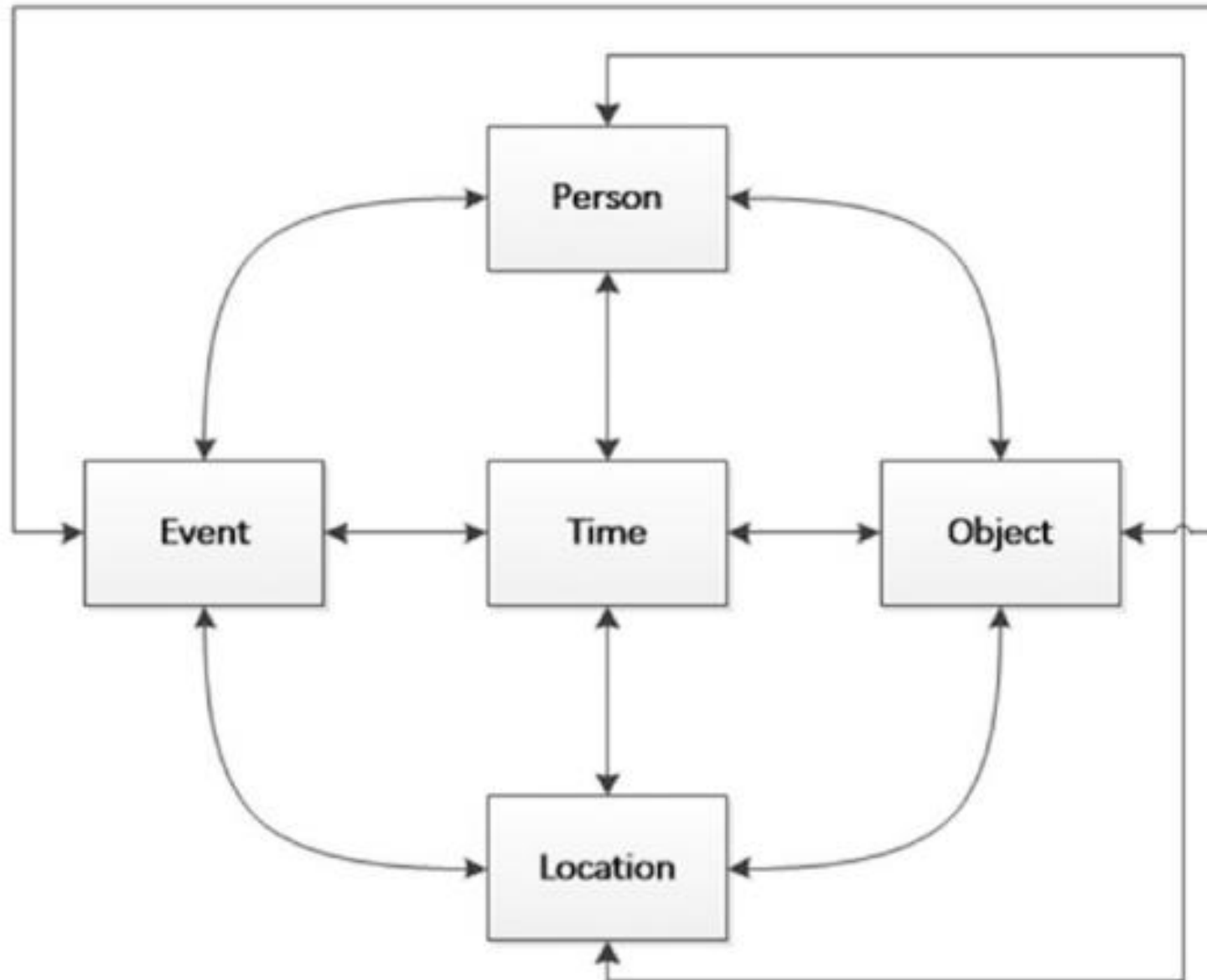
REFERENCED SATELLITES

- ▶ Reference satellites are referenced from satellites that can be used by other satellites to prevent redundant storage of reference characteristics.
- ▶ Typical reference satellites are
 - Standard codes: These are codes such as ISO 3166 for country codes, ISO 4217 for currencies, and ISO 8601 for time zones.

TIME-PERSON-OBJECT-LOCATION- EVENT DATA VAULT

- ▶ The data vault we use is based on the Time-Person-Object-Location-Event (T-P-O-L-E) design principle.

PROCESS SUPERSTEP



Time Section

- ▶ The time section contains the complete data structure for all data entities related to recording the time at which everything occurred.

Time Hub

The time hub consists of the following fields:

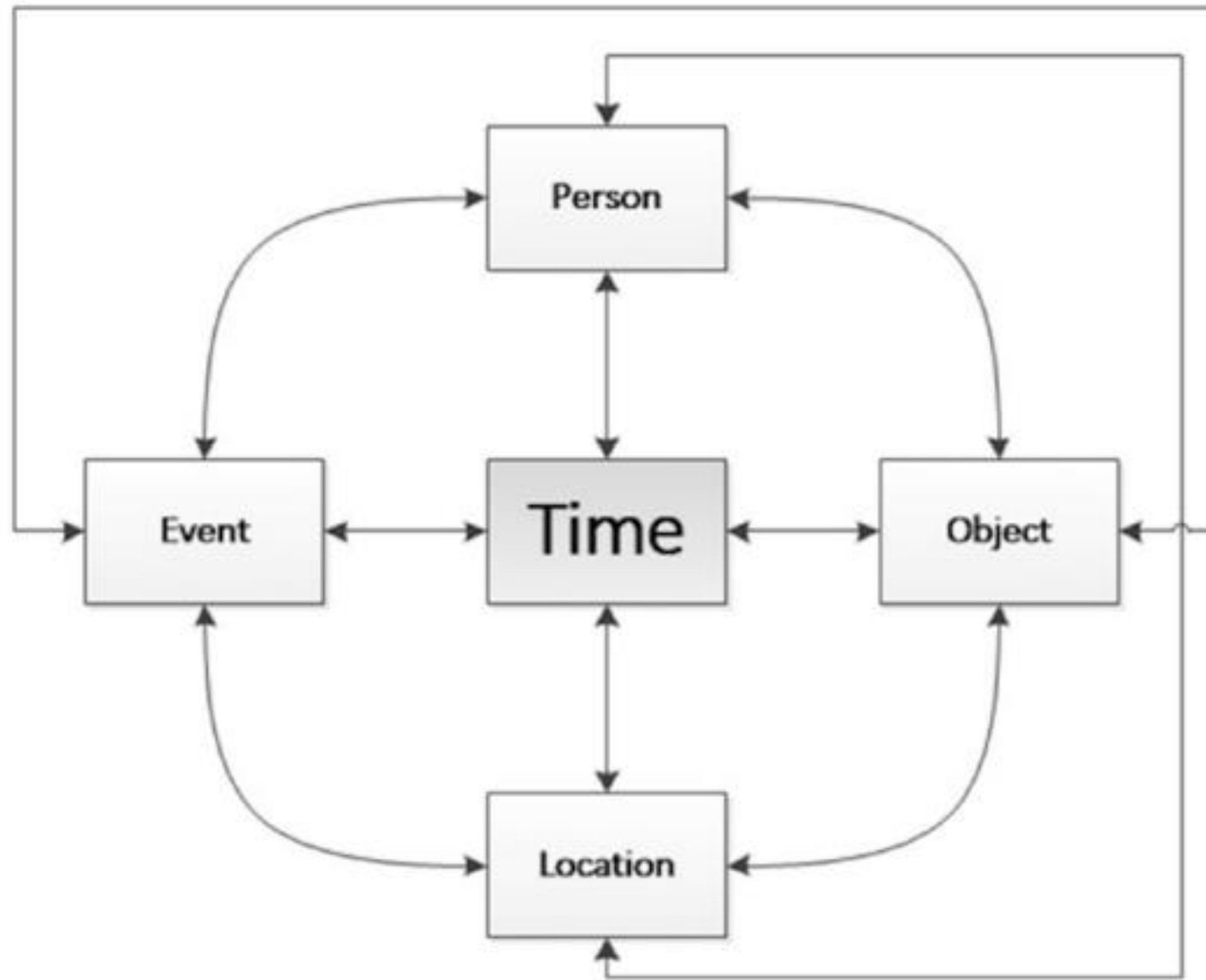
```
CREATE TABLE [Hub-Time] (  
    IDNumber      VARCHAR (100) PRIMARY KEY,  
    IDTimeNumber  Integer,  
    ZoneBaseKey   VARCHAR (100),  
  
    DateTimeKey   VARCHAR (100),  
    DateTimeValue DATETIME  
);
```

varchar means character data that is varying. Also known as Variable Character, it is an indeterminate length string data type.

It can hold numbers, letters and special characters.

Time Links

The time links link the time hub to the other hubs.



Time links

Time-Person Link

- ▶ This connects date-time values within the person hub to the time hub.
- ▶ Dates such as birthdays, marriage anniversaries, and the date of reading this book can be recorded as separate links in the data vault.
- ▶ The normal format is BirthdayOn, MarriedOn, or ReadBookOn. The format is simply a pair of keys between the time and person hubs.

Time-Object Link

- ▶ This connects date-time values within the object hub to the time hub. Dates such as those on which you bought a car, sold a car, and read this book can be recorded as separate links in the data vault.
- ▶ The normal format is BoughtCarOn, SoldCarOn, or ReadBookOn. The format is simply a pair of keys between the time and object hubs.

Time-Location Link

- ▶ This connects date-time values in the location hub to the time hub. Dates such as moved to post code SW1, moved from post code SW1, and read book at post code SW1 can be recorded as separate links in the data vault.
- ▶ The normal format is MovedToPostCode, MovedFromPostCode, or ReadBookAtPostCode. The format is simply a pair of keys between the time and location hubs.

Time-Event Link

- ▶ This connects date-time values in the event hub with the time hub. Dates such as those on which you have moved house and changed vehicles can be recorded as separate links in the data vault.
- ▶ The normal format is MoveHouse or ChangeVehicle. The format is simply a pair of keys between the time and event hubs.

Time Satellites

- ▶ Time satellites are the part of the vault that stores the following fields:

```
CREATE TABLE [Satellite-Time-<Time Zone>] (  
    IDZoneNumber    VARCHAR (100) PRIMARY KEY,  
    IDTimeNumber    INTEGER,  
    ZoneBaseKey     VARCHAR (100),  
    DateTimeKey     VARCHAR (100),  
    UTCDateTimeValue DATETIME,  
    Zone            VARCHAR (100),  
    DateTimeValue   DATETIME  
);
```

Time satellites enable you to work more easily with international business patterns. You can move between time zones to look at such patterns as “In the morning . . . ” or “At lunchtime . . . ”

These capabilities will be used during the Transform superstep, to discover patterns and behaviors around the world.

Person Section

- ▶ The person section contains the complete data structure for all data entities related to recording the person involved.

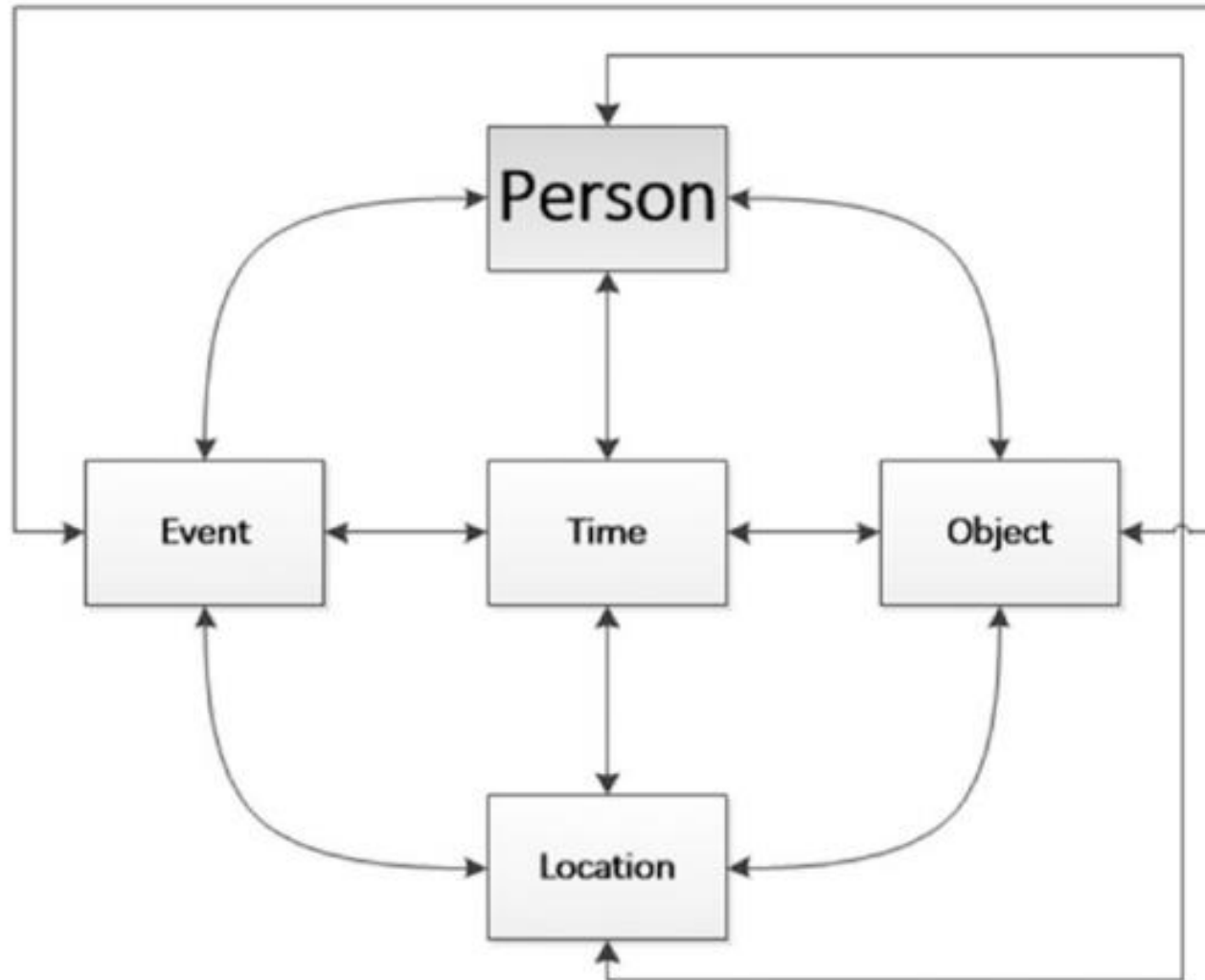
Person Hub

- ▶ The person hub consists of a series of fields that supports a “real” person. The person hub consists of the following fields:

```
CREATE TABLE [Hub-Person] (  
    IDPersonNumber    INTEGER,  
    FirstName         VARCHAR (200),  
    SecondName        VARCHAR (200),  
    LastName          VARCHAR (200),  
    Gender            VARCHAR (20),  
    TimeZone          VARCHAR (100),  
    BirthDateKey      VARCHAR (100),  
    BirthDate         DATETIME  
);
```

Person Links

- ▶ This links the person hub to the other hubs.



Person links

Person-Time Link

- ▶ This link joins the person to the time hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Person-Time] (  
    IDPersonNumber    INTEGER,  
    IDTimeNumber      INTEGER,  
    ValidDate         DATETIME  
);
```

Person-Object Link

- ▶ This link joins the person to the object hub to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Person-Object] (  
    IDPersonNumber    INTEGER,  
    IDObjectNumber    INTEGER,  
    ValidDate         DATETIME  
);
```

Person-Location Link

This link joins the person to the location hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Person-Time] (  
    IDPersonNumber    INTEGER,  
    IDLocationNumber  INTEGER,  
    ValidDate         DATETIME  
);
```

Person-Event Link

This link joins the person to the event hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Person-Time] (  
    IDPersonNumber    INTEGER,  
    IDEventNumber     INTEGER,  
    ValidDate         DATETIME  
);
```

Person Satellites

- ▶ The person satellites are the part of the vault that stores the temporal attributes and descriptive attributes of the data. The satellite is of the following format


```
CREATE TABLE [Satellite-Person-Gender] (  
  PersonSatelliteID VARCHAR (100),
```

```
  IDPersonNumber INTEGER,  
  FirstName VARCHAR (200),  
  SecondName VARCHAR (200),  
  LastName VARCHAR (200),  
  BirthDateKey VARCHAR (20),  
  Gender VARCHAR (10),  
  );
```

Object Section

- ▶ The object section contains the complete data structure for all data entities related to recording the object involved.

Object Hub

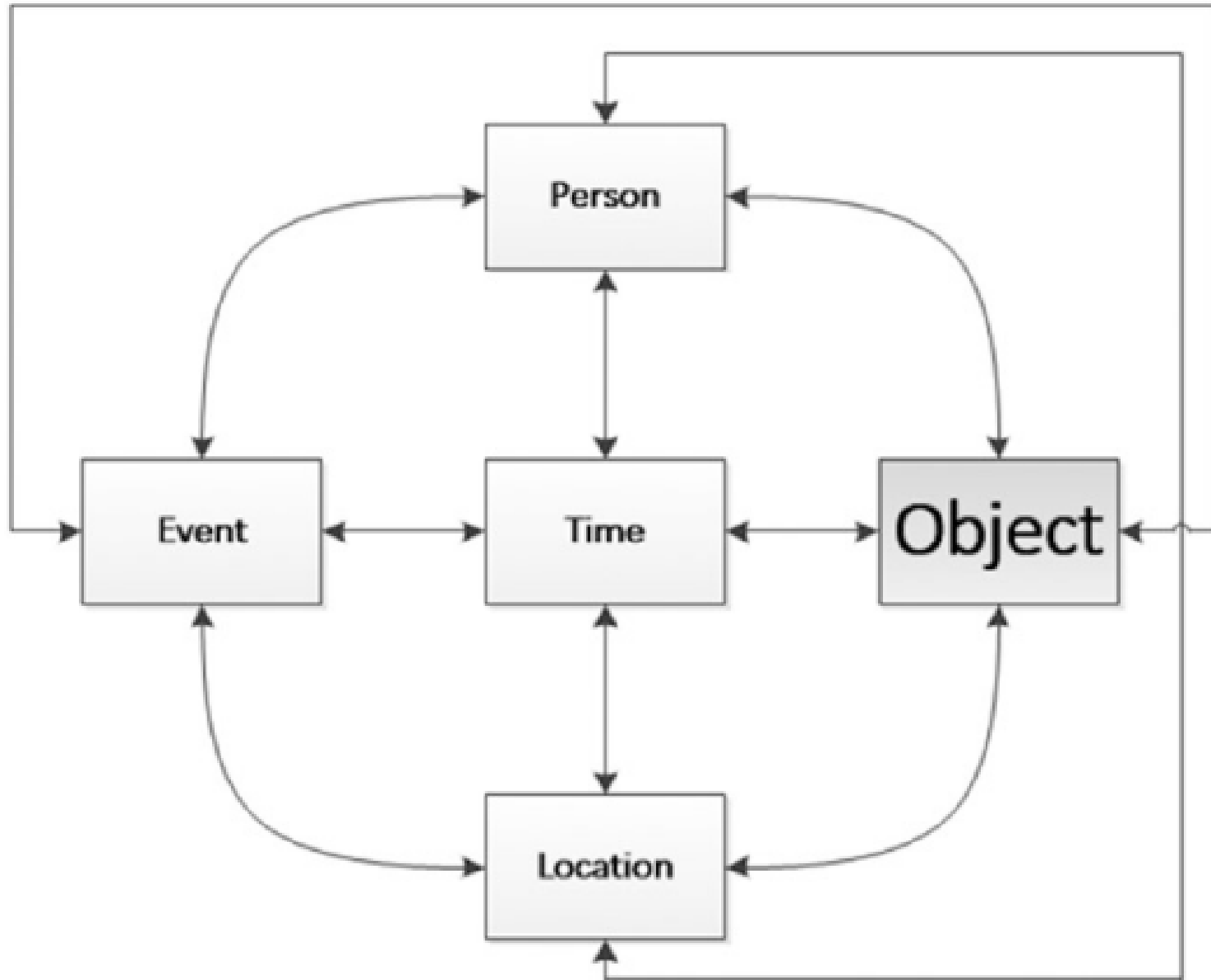
The object hub consists of a series of fields that supports a “real” object. The object hub consists of the following fields:

```
CREATE TABLE [Hub-Object-Species] (  
  IDObjectNumber INTEGER,  
  ObjectBaseKey VARCHAR (100),  
  ObjectNumber VARCHAR (100),  
  ObjectValue VARCHAR (200),  
);
```

This structure enables you as a data scientist to categorize the objects in the business environment.

Object Links

- ▶ These link the object hub to the other hubs.



Object-Time Link

This link joins the object to the time hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Object-Time] (  
    IDObjectNumber    INTEGER,  
    IDTimeNumber      INTEGER,  
    ValidDate         DATETIME  
);
```

Object-Person Link

This link joins the object to the person hub to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Object-Person] (  
    IDObjectNumber    INTEGER,  
    IDPersonNumber    INTEGER,  
    ValidDate         DATETIME  
);
```

Object-Location Link

This link joins the object to the location hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Object-Location] (  
    IDObjectNumber    INTEGER,  
    IDLocationNumber  INTEGER,  
    ValidDate         DATETIME  
);
```

Object-Event Link

This link joins the object to the event hub to describe the relationships between the two hubs.

Object Satellites

Object satellites are the part of the vault that stores and provisions the detailed characteristics of objects. The typical object satellite has the following data fields:

```
CREATE TABLE [Satellite-Object-Make-Model] (  
  IDObjectNumber INTEGER,  
  ObjectSatelliteID VARCHAR (200),  
  ObjectType VARCHAR (200),  
  ObjectKey VARCHAR (200),  
  ObjectUUID VARCHAR (200).
```

```
  Make VARCHAR (200),  
  Model VARCHAR (200)  
);
```

Location Section

- ▶ The location section contains the complete data structure for all data entities related to recording the location involved.

Location Hub

The location hub consists of a series of fields that supports a GPS location. The location hub consists of the following fields:

```
CREATE TABLE [Hub-Location] (  
    IDLocationNumber INTEGER,  
    ObjectBaseKey  VARCHAR (200),  
    LocationNumber INTEGER,  
    LocationName   VARCHAR (200),  
    Longitude      DECIMAL (9, 6),  
    Latitude       DECIMAL (9, 6)  
);
```

The location hub enables you to link any location, address, or geospatial information to the rest of the data vault.

Location Links

- ▶ The location links join the location hub to the other hubs.

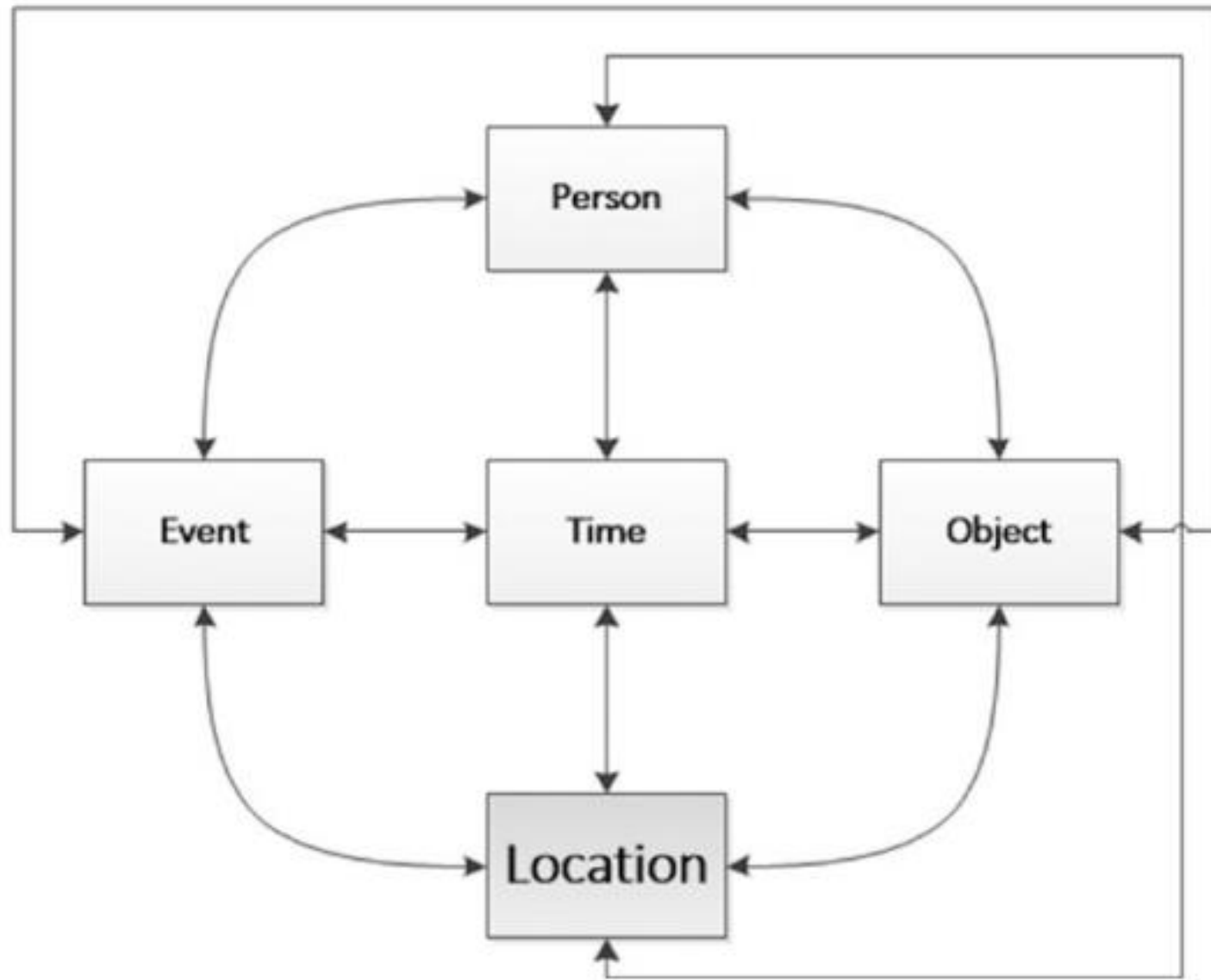


Figure 9.6 Location links

Location-Time Link

The link joins the location to the time hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Location-Time] (  
    IDLocationNumber    INTEGER,  
    IDTimeNumber        INTEGER,  
    ValidDate           DATETIME  
);
```

These links support business actions such as ArrivedAtShopAtDateTime or ShopOpensAtTime.

Location-Person Link

This link joins the location to the person hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Location-Person] (  
    IDLocationNumber    INTEGER,  
    IDPersonNumber      INTEGER,  
    ValidDate           DATETIME  
);
```

These links support such business actions as `ManagerAtShop` or `SecurityAtShop`.

Location-Object Link

This link joins the location to the object hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Location-Object] (  
    IDLocationNumber    INTEGER,  
    IDObjectNumber      INTEGER,  
    ValidDate           DATETIME  
);
```

These links support such business actions as ShopDeliveryVan or RackAtShop.

Location-Event Link

This link joins the location to the event hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Location-Event] (  
    IDLocationNumber    INTEGER,  
    IDEventNumber       INTEGER,  
    ValidDate           DATETIME  
);
```

These links support such business actions as ShopOpened or PostCodeDeliveryStarted.

Location Satellites

- ▶ The location satellites are the part of the vault that stores and provisions the detailed characteristics of where entities are located. The typical location satellite has the following data fields

```
CREATE TABLE [Satellite-Location-PostCode] (  
  IDLocationNumber INTEGER,  
  LocationSatelliteID VARCHAR (200),  
  LocationType VARCHAR (200),  
  LocationKey VARCHAR (200),  
  LocationUUID VARCHAR (200),  
  CountryCode VARCHAR (20),  
  PostCode VARCHAR (200)  
);
```

The location satellites will also hold additional characteristics that are related only to your specific customer.

They may split their business areas into their own regions, e.g., Europe, Middle-East, and China.

Event Section

- ▶ The event section contains the complete data structure for all data entities related to recording the event that occurred.

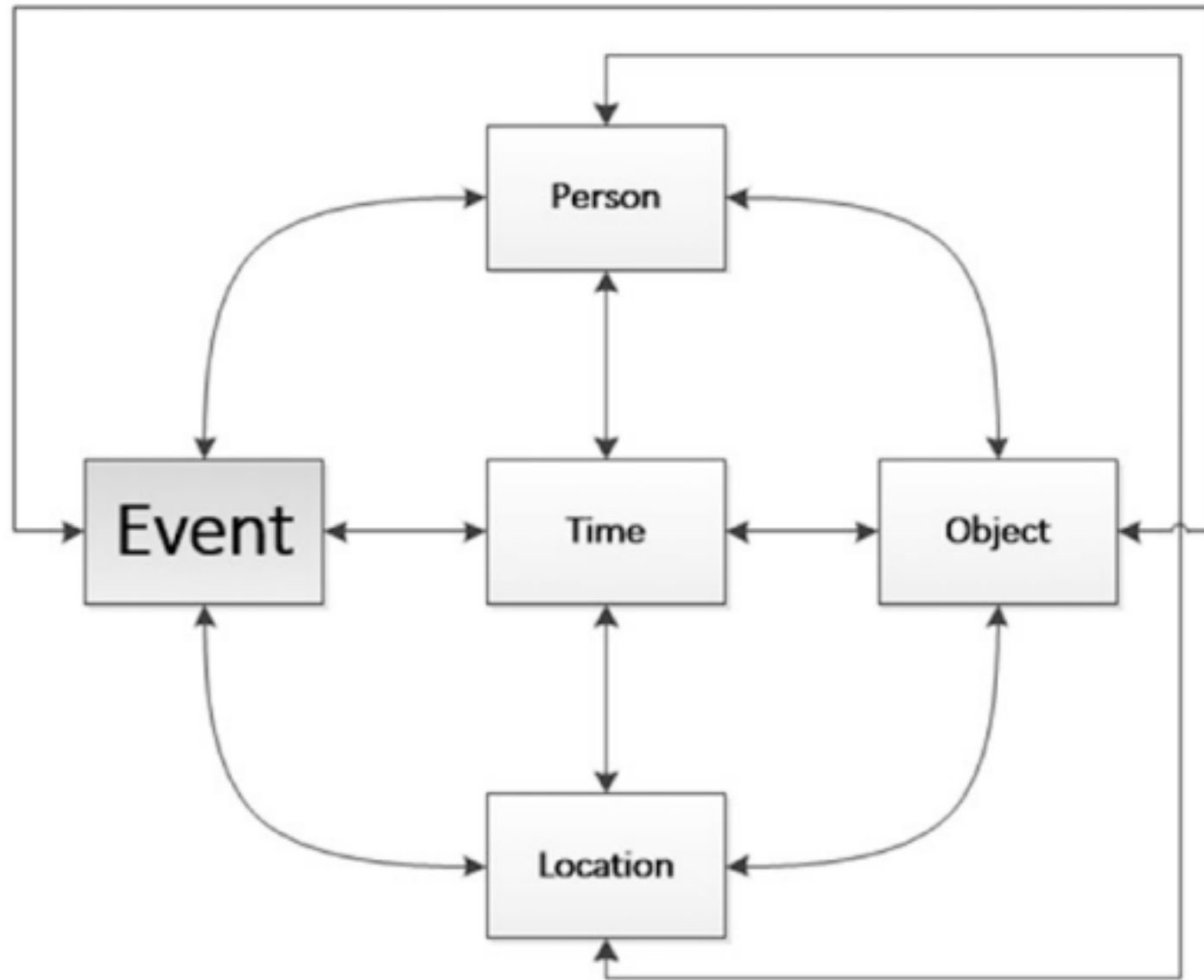
Event Hub

The event hub consists of a series of fields that supports events that happens in the real world. The event hub consists of the following fields:

```
CREATE TABLE [Hub-Event] (  
    IDEventNumber    INTEGER,  
    EventType        VARCHAR (200),  
    EventDescription VARCHAR (200)  
);
```

Event Links

- ▶ Event links join the event hub to the other hubs .



Event-Time Link

This link joins the event to the time hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Event-Time] (  
    IDEventNumber    INTEGER,  
    IDTimeNumber     INTEGER,  
    ValidDate        DATETIME  
);
```

These links support such business actions as `DeliveryDueAt` or `DeliveredAt`.

Event-Person Link

This link joins the event to the person hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Event-Person] (  
    IDEventNumber    INTEGER,  
    IDPersonNumber   INTEGER,  
    ValidDate        DATETIME  
);
```

These links support such business actions as `ManagerAppointAs` or `StaffMemberJoins`.

Event-Object Link

This link joins the event to the object hub, to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Event-Object] (  
    IDEventNumber    INTEGER,  
    IDObjectNumber   INTEGER,  
    ValidDate        DATETIME  
);
```

These links support such business actions as `VehicleBuy`, `VehicleSell`, or `ItemInStock`.

Event-Location Link

The link joins the event to the location hub to describe the relationships between the two hubs. The link consists of the following fields:

```
CREATE TABLE [Link-Event-Location] (  
    IDEventNumber    INTEGER,  
    IDTimeNumber     INTEGER,  
    ValidDate        DATETIME  
);
```

These links support such business actions as `DeliveredAtPostCode` or `PickupFromGPS`.

Event Satellites

The event satellites are the part of the vault that stores the details related to all the events that occur within the systems you will analyze with your data science. I suggest that you keep to one type of event per satellite. This enables future expansion and easier long-term maintenance of the data vault.

FEW IMPORTANT TERMS

- ▶ Local Time
- ▶ Coordinated Universal Time

LOCAL TIME

- ▶ The approved local time is agreed by the specific country, by approving the use of a specific time zone for that country.
- ▶ The general format for local time is hh:mm:ss

LOCAL TIME

- ▶ There is a Simple Network Time Protocol (SNTP) that you can use automatically to synchronize your system's time with that of a remote server.
- ▶ The SNTP can be used to update the clock on a machine with a remote server. This keeps your machine's time accurate, by synchronizing with servers that are known to have accurate times


```
from datetime import datetime
now_date = datetime.now()
print('Date:',str(now_date.strftime("%Y-%m-%d %H:%M:%S (%Z)
(%z)"))))
```

The results is 2023-10-03 9:10:58 () ()

The **strftime()** function is used to convert date and time objects to their string representation.

%Z: Replaced by the timezone name or abbreviation, or by no bytes if no timezone information exists.

The reason for the two empty brackets is that the date time is in a local time setting

COORDINATED UNIVERSAL TIME(UTC)

- ▶ **Coordinated Universal Time** or **UTC** is the primary time standard by which the world regulates clocks and time.
- ▶ The valid format is hh:mm:ss±hh:mm or hh:mm:ss±hh

Combining Date and Time

- ▶ When using date and time in one field, ISO supports the format YYYY-MM-DDThh:mm:ss.
- ▶ These date and time combinations are regularly found in international companies' financials or logistics shipping and on my smartphone.

Day of the Week

Number	Name
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday

```
now_date_local=datetime.now()
```

```
now_date=now_date_local.replace(tzinfo=timezone('Europe/Lon  
don'))
```

```
print('Weekday:',str(now_date.strftime("%w")))
```

Event

- ▶ This structure records any specific event or action that is discovered in the data sources. An event is any action that occurs within the data sources. Events are recorded using three main data entities: Event Type, Event Group, and Event Code.

Explicit Event

- ▶ This type of event is stated in the data source clearly and with full details. There is clear data to show that the specific action was performed.
- ▶ Example: A security card with number 1234 was used to open door A.
- ▶ You are reading Chapter 9 of Practical Data Science.

Implicit Event

- ▶ This type of event is formulated from characteristics of the data in the source systems plus a series of insights on the data relationships.

Implicit Event

- ▶ A security card with number 8884.1 was used to open door X.
- ▶ • A security card with number 8884.1 was issued to Mr. Vermeulen.
- ▶ • Room 302 is fitted with a security reader marked door X.
These three events would imply that Mr. Vermeulen entered room 302 as an event.

Data Science Process(5 why's technique)

- ▶ Data science is at its core about curiosity and inquisitiveness. This core is rooted in the 5 Whys. The 5 Whys is a technique used in the analysis phase of data science.

Data Science Process(5 why's technique)

- ▶ **Benefits of the 5 Whys:** The 5 Whys assist the data scientist to identify the root cause of a problem and determine the relationship between different root causes of the same problem.
- ▶ It is one of the simplest investigative tools—easy to complete without intense statistical analysis.

When Are the 5 Whys Most Useful?

- ▶ The 5 Whys are most useful for finding solutions to problems that involve human factors or interactions that generate multilayered data problems.
- ▶ In day-to-day business life, they can be used in real-world businesses to find the root causes of issues.

How to Complete the 5 Whys

- ▶ Write down the specific problem.
- ▶ This will help you to formalize the problem and describe it completely.
- ▶ It also helps the data science team to focus on the same problem.
- ▶ Ask why the problem occurred and write the answer below the problem.
- ▶ If the answer you provided doesn't identify the root cause of the problem that you wrote down first, ask why again, and write down that answer.
- ▶ Loop back to the preceding step until you and your customer are in agreement that the problem's root cause is identified.

Example of 5 why's technique

- ▶ Problem Statement: Customers are unhappy because they are being shipped products that don't meet their specifications.
- ▶ Define 5 why's technique for the above problem statement.

Example of 5 why's technique

- ▶ 1. Why are customers being shipped bad products?
- ▶ Because manufacturing built the products to a specification that is different from what the customer and the salesperson agreed to.

Example of 5 why's technique

2. Why did manufacturing build the products to a different specification than that of sales?

- Because the salesperson accelerates work on the shop floor by calling the head of manufacturing directly to begin work. An error occurred when the specifications were being communicated or written down.

Example of 5 why's technique

3. Why does the salesperson call the head of manufacturing directly to start work instead of following the procedure established by the company?

- Because the “start work” form requires the sales director’s approval before work can begin and slows the manufacturing process.

Example of 5 why's technique

4. Why does the form contain an approval for the sales director?

- Because the sales director must be continually updated on sales for discussions with the CEO, as my retailer customer was a top ten key account

FishBorne Diagrams

- ▶ The fishbone diagram or Ishikawa diagram is a useful tool to find where each data fits into data vault.
- ▶ This is a cause-and-effect diagram that helps managers to track down the reasons for imperfections, variations, defects, or failures.



FishBorne Diagrams

- ▶ The diagram looks just like a fish's skeleton with the problem at its head and the causes for the problem feeding into the spine.
- ▶ Once all the causes that underlie the problem have been identified, managers can start looking for solutions to ensure that the problem doesn't become a recurring one. It can also be used in product development.

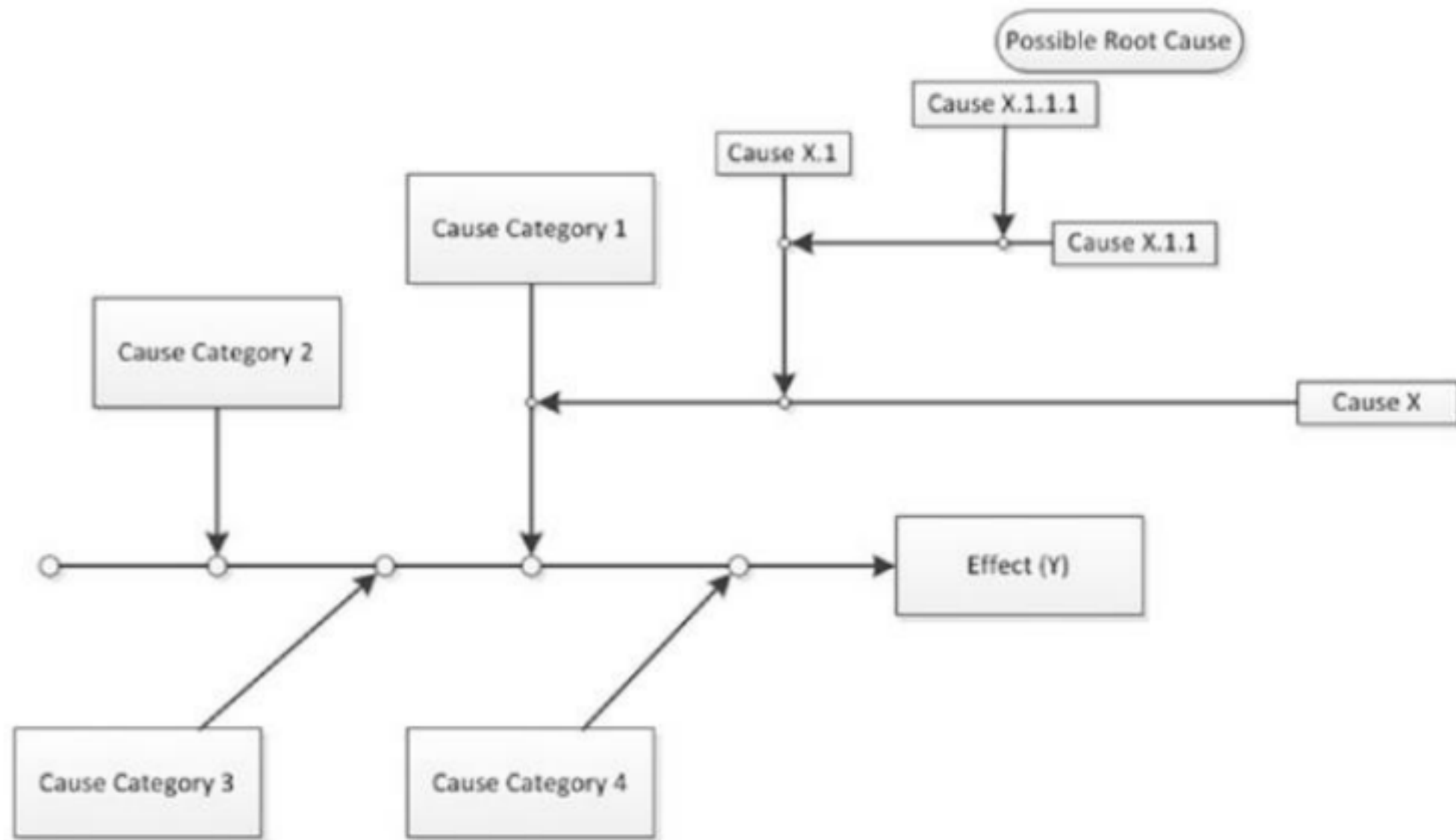


Figure 4.1-8. Fishbone diagram

What's good about fishborne diagrams?

- ▶ The fishbone diagram strives to pinpoint everything that's wrong with current market offerings so that you can develop an innovation that doesn't have these problems.
- ▶ Finally, the fishbone diagram is also a great way to look for and prevent quality problems before they ever arise.

Monte Carlo Simulation

- ▶ As a data scientist, Monte carlo simulation gives you an indication of how your model will react under real-life situations.
- ▶ It also gives the data scientist a tool to check complex systems, wherein the input parameters are high-volume or complex.

Monte Carlo Simulation

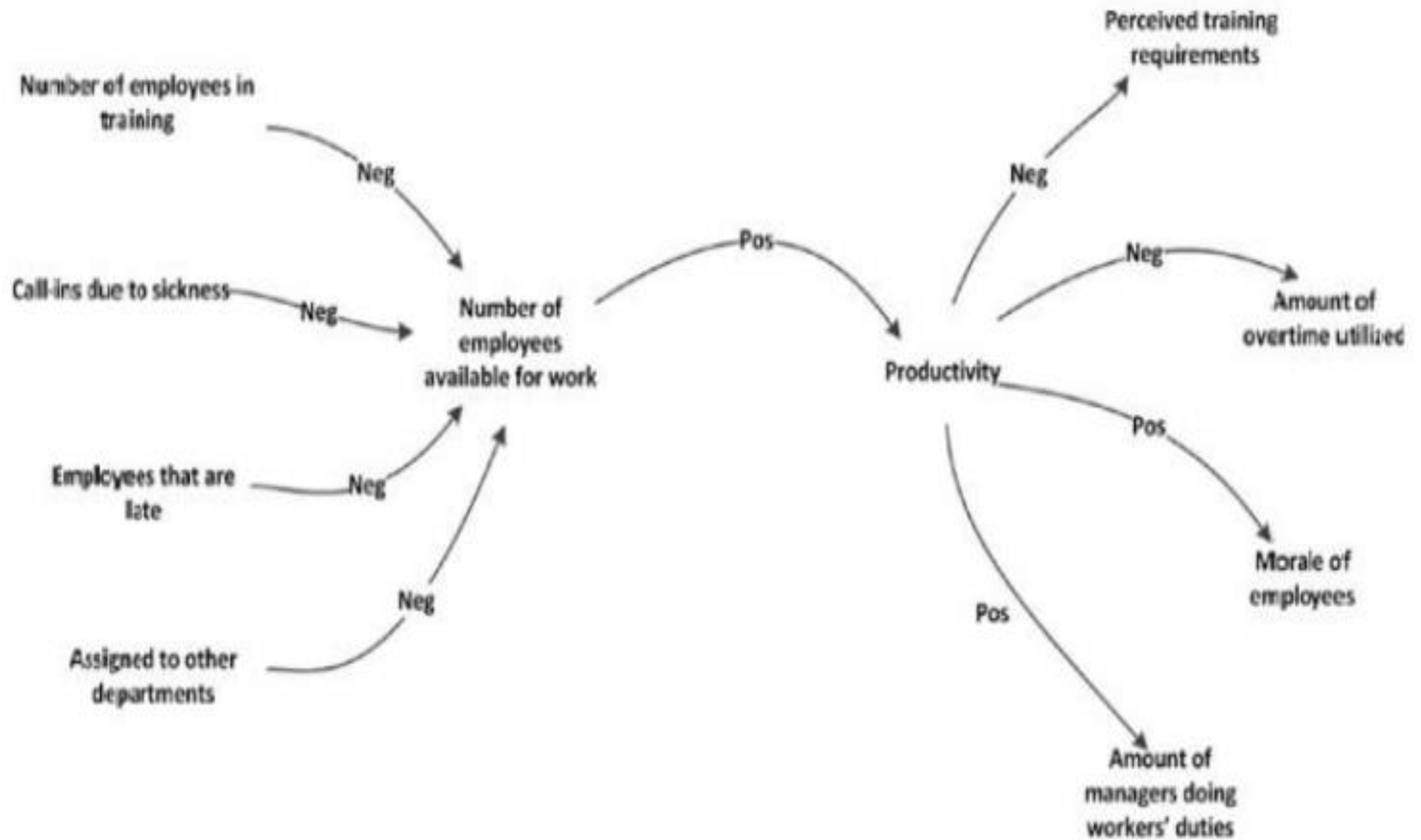
- ▶ Monte Carlo simulation technique performs analysis by building models of possible results, by substituting a range of values—a probability distribution—for parameters that have inherent uncertainty.
- ▶ It then calculates results over and over, each time using a different set of random values from the probability functions.

Causal Loop Diagrams

- ▶ A causal loop diagram (CLD) is a causal diagram that aids in visualizing how a number of variables in a system are interrelated and drive cause-and-effect processes.
- ▶ The diagram consists of a set of nodes and edges. Nodes represent the variables, and edges are the links that represent a connection or a relation between the two variables.

Example of Causal Loop Diagrams

- ▶ Example: The challenge is to keep the “Number of Employees Available to Work and Productivity” as high as possible.



Pareto Chart

- ▶ A Pareto chart is a bar graph. It is also called as Pareto diagram or Pareto analysis.
- ▶ The lengths of the bars represent frequency or cost (time or money), and are arranged with longest bars on the left and the shortest to the right. In this way the chart visually depicts which situations are more significant.

When to use Pareto Chart?

- ▶ When analysing data about the frequency of problems or causes in a process.
- ▶ When there are many problems or causes and you want to focus on the most significant.
- ▶ When analysing broad causes by looking at their specific components.
- ▶ When communicating with others about your data.

Example of Pareto Chart

Following Diagram shows how many customer complaints were received in each of five categories.



Pareto Chart

Correlation Analysis

- ▶ Feature development is performed between data items, to find relationships between data values.
- ▶ `import pandas as pd`
- ▶ `a = [[1, 2, 4], [5, 4.1, 9], [8, 3, 13], [4, 3, 19], [5, 6, 12], [5, 6, 11], [5, 6, 4.1], [4, 3, 6]]`
- ▶ `df = pd.DataFrame(data=a)`
- ▶ `cr=df.corr()`
- ▶ `print(cr)`

Pandas **dataframe.corr()** is used to find the pairwise correlation of all columns in the Pandas Dataframe in Python.

Any NaN values are automatically excluded. To ignore any non-numeric values, use the parameter `numeric_only = True`.

Forecasting

- ▶ Forecasting is the ability to project a possible future, by looking at historical data. The data vault enables these types of investigations, owing to the complete history it collects as it processes the source's systems data.

Forecasting:Example

- ▶ As a datascientist , you may need to answer such questions and do the forecasting as:

What should we buy?

What should we sell?

Where will our next business come from?

Combined Steps of the DataScience Process

- ▶ Step 1: It begins with a question.
- ▶ Step 2: Design a model, select prototype for the data and start a virtual simulation. Some statistics and mathematical solutions can be added to start a data science model. All questions must be related to customer's business, such a way that answer must provide an insight of business.

Combined Steps of the DataScience Process

- ▶ Step3: Formulate a hypothesis based on collected observation. Based on model process the observation and prove whether hypothesis is true or false.
- ▶ Step4: Compare the above result with the real-world observations and provide these results to real-life business.
- ▶ Step 5: Communicate the progress and intermediate results with customers and subject expert and involve them in the whole process to ensure that they are part of journey of discovery.

UNIT-4 PART-2

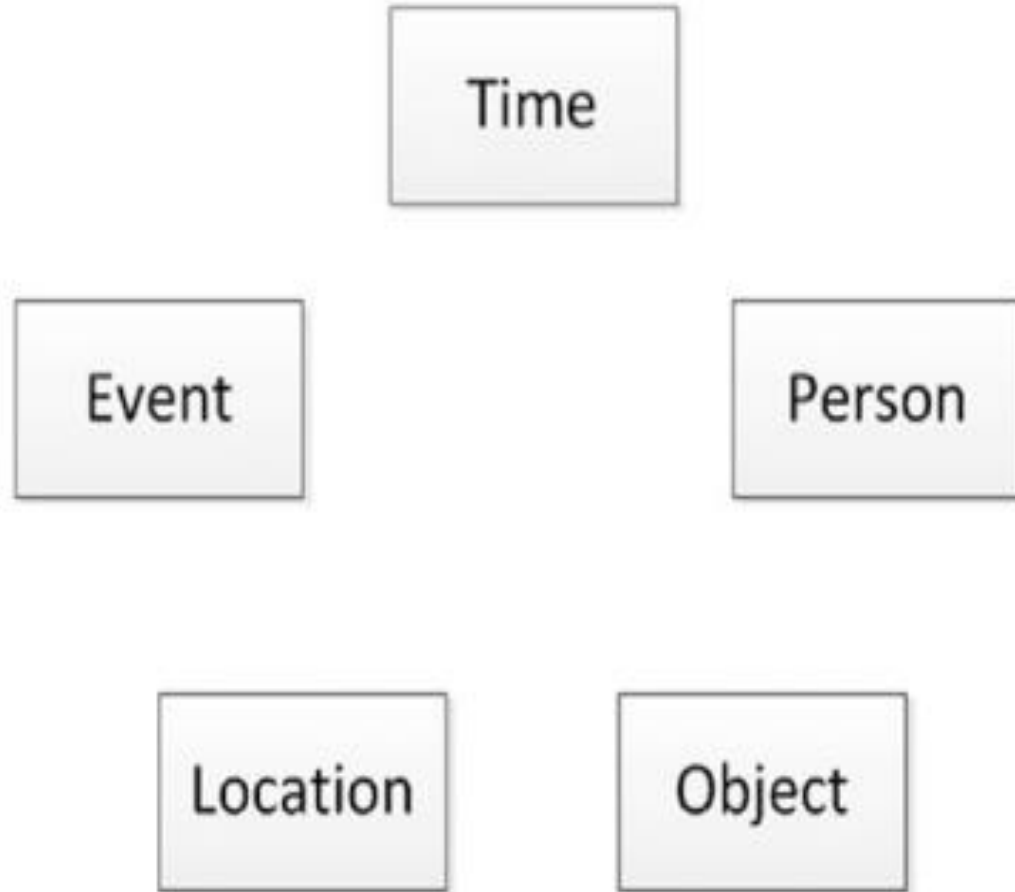
TRANSFORM SUPERSTEP

The Transform superstep allows you, as a data scientist, to take data from the data vault and formulate answers to questions raised by your investigations.

The transformation step is the data science process that converts results into insights.

The data warehouse is the only data structure delivered from the Transform step.

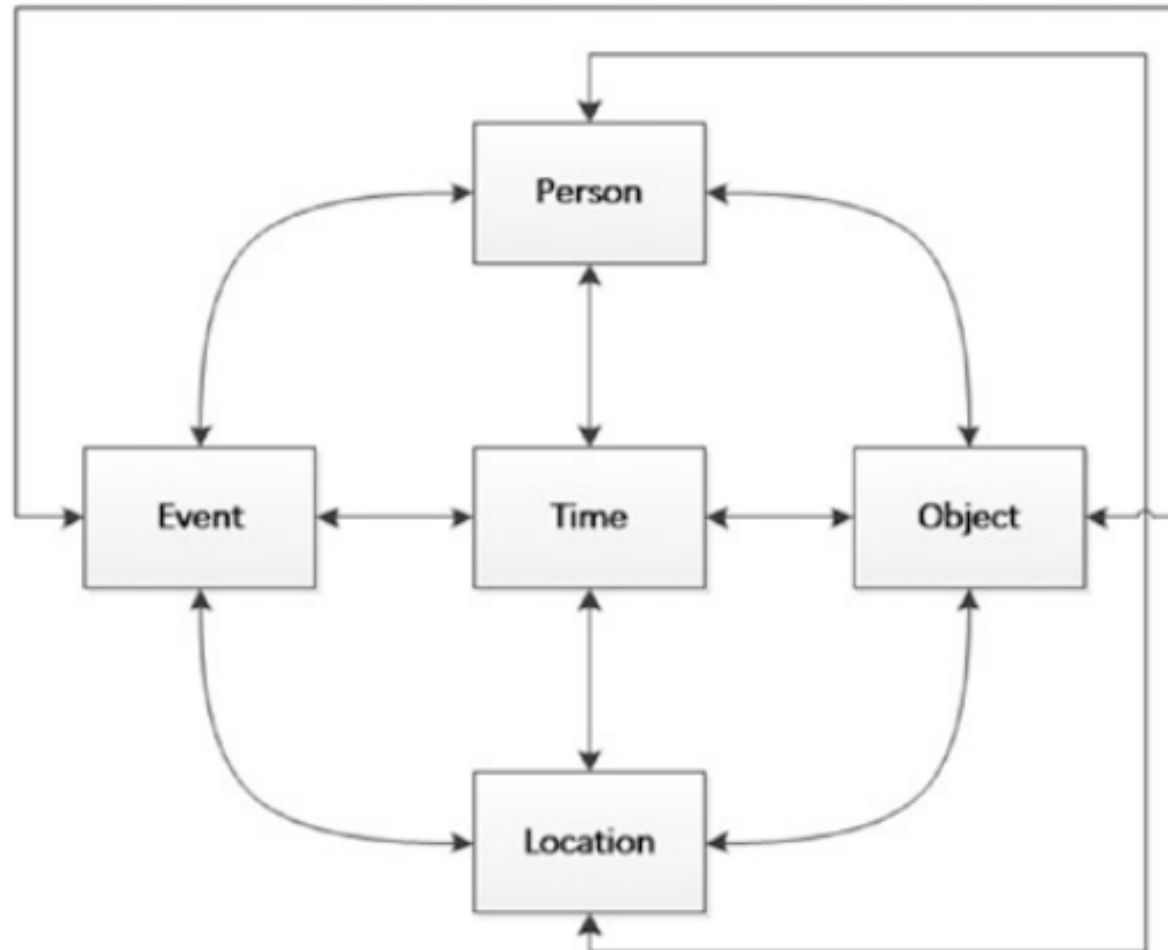
The transformations are tuned to work with the five dimensions of the data vault.



. *Five categories of data*

DIMENSION CONSOLIDATION

- ▶ The data vault consists of five categories of data, with linked relationships and additional characteristics in satellite hubs.
- ▶ To perform dimension consolidation, there is need to start with a given relationship in the data vault and construct a sun model for that relationship.



T-P-O-L-E High-level design

Sun Model

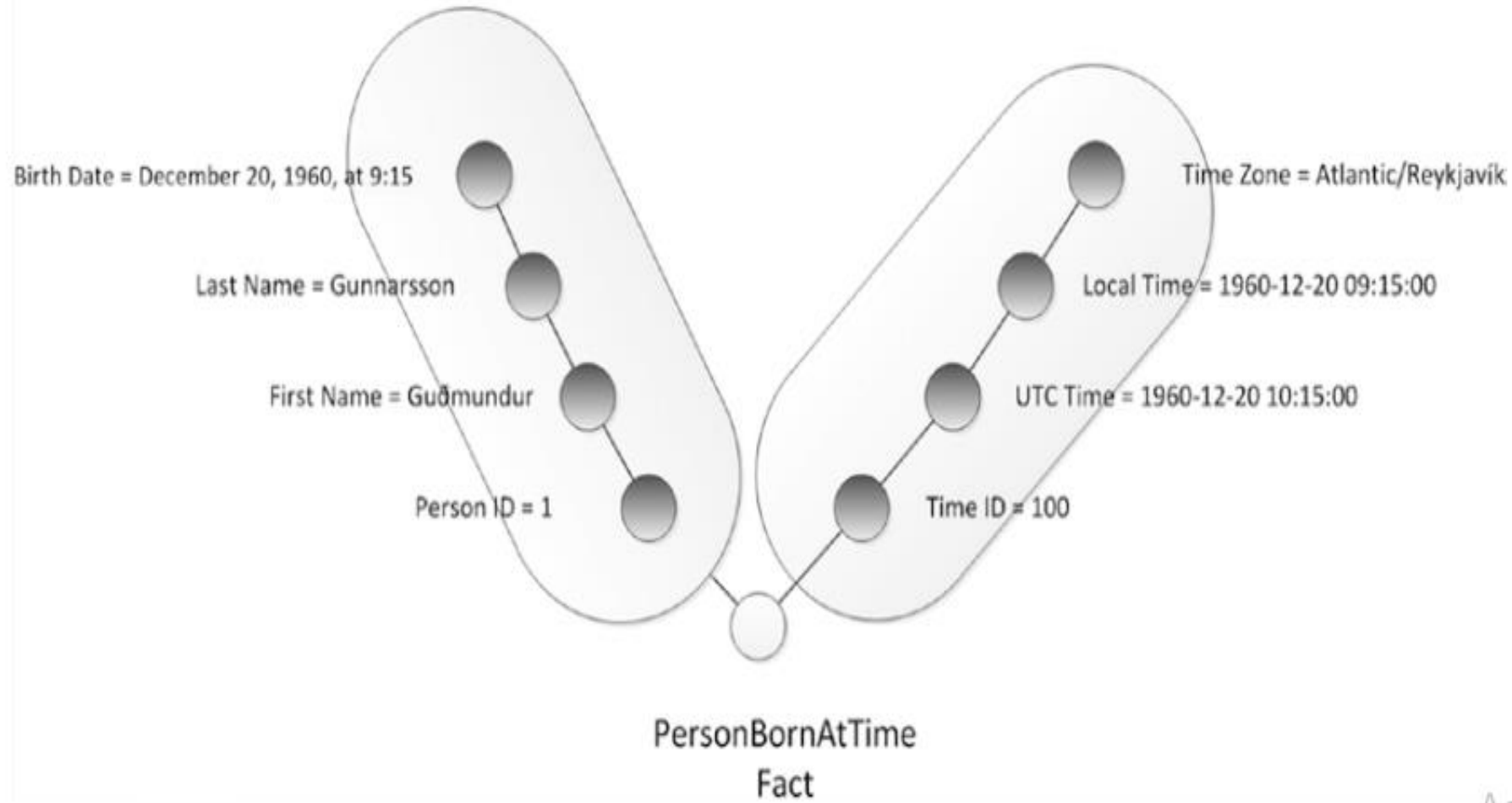
- ▶ The use of sun models is a technique that enables the data scientist to perform consistent dimension consolidation, by explaining the intended data relationship with the business, without exposing it to the technical details required to complete the transformation processing.

Person-to-Time Sun Model

- ▶ Person-to-Object Sun Model explains the relationship between the Person and Object categories in the data vault.
- ▶ Dimensions : Person and Time
- ▶ Fact: Person Born at Time

Person Dimension

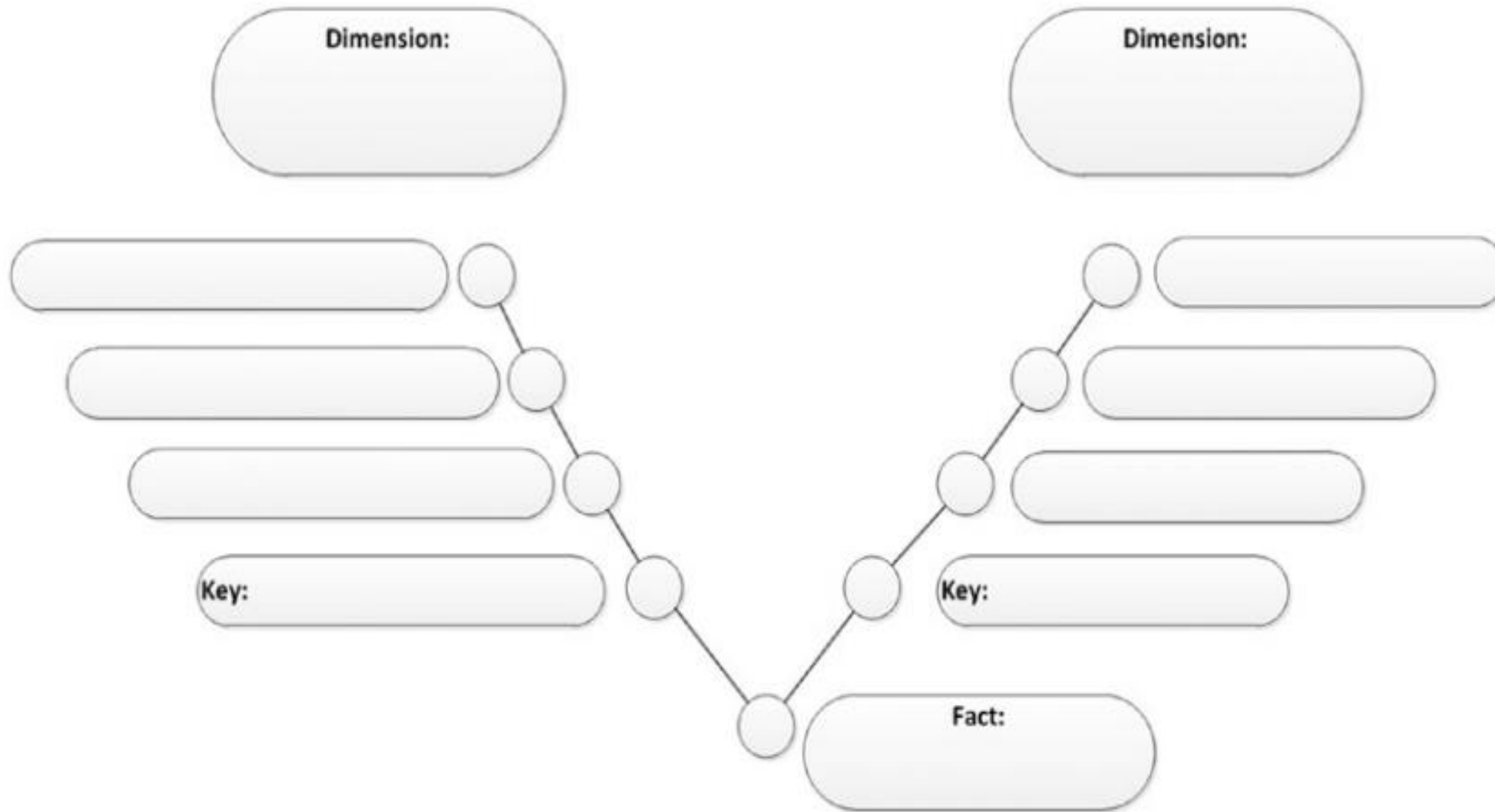
Time Dimension



Person-to-Time sun model (explained)

Activate
Go to Sett

Template Model



Template for a simple sun model

Person-to-Object Sun Model

- ▶ Person-to-Object Sun Model explains the relationship between the Person and Object categories in the data vault.
- ▶ The sun model is constructed to show all the characteristics from the two data vault hub categories.

Person

Birth Date = December 20, 1960, at 9:15

Last Name = Gunnarsson

First Name = Guomundur

Person ID = 1

Object

Common Name = Human

Species = Homo sapiens

Object ID = 888

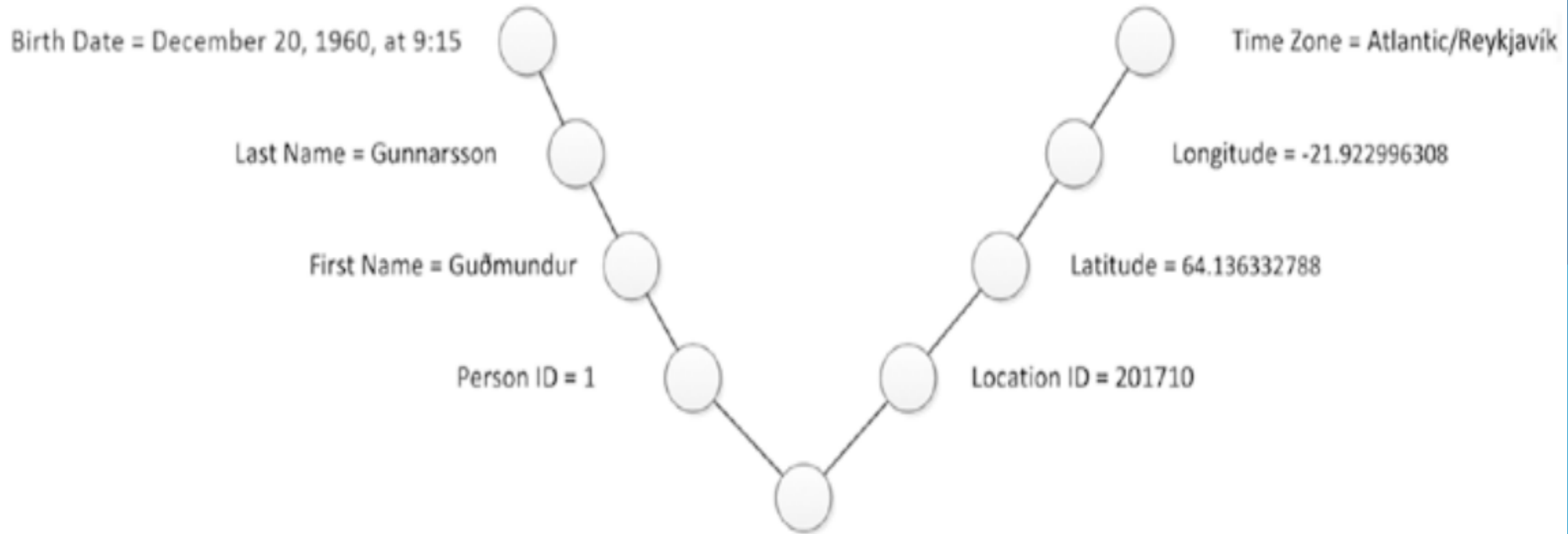
Sun model for the PersonIsSpecies fact

Person-to-Location Sun Model

- ▶ Person-to-Location Sun Model explains the relationship between the Person and Location categories in the data vault.

Person

Location

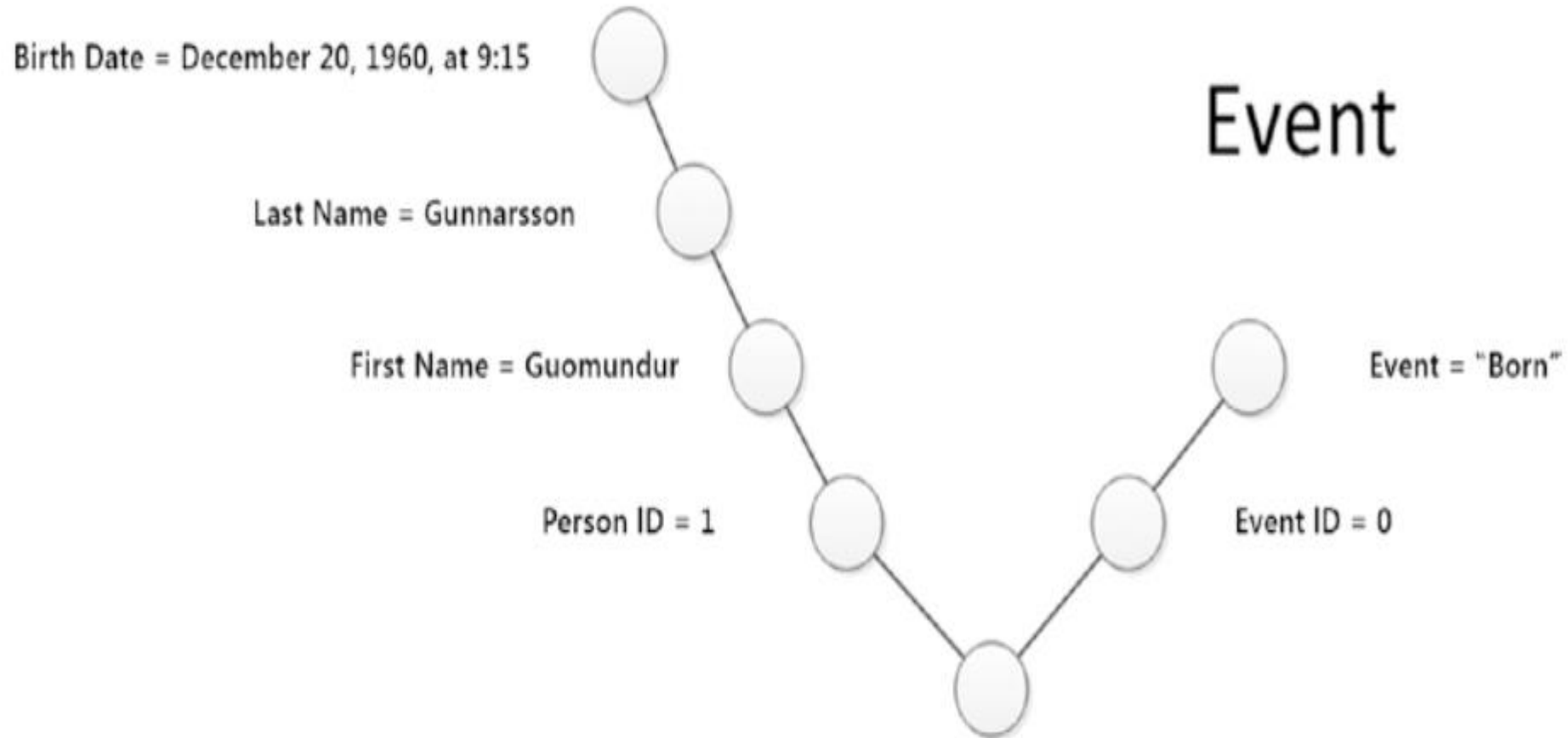


Sun model for PersonAtLocation fact

Person-to-Event Sun Model

- ▶ Person-to-Event Sun Model explains the relationship between the Person and Event categories in the data vault.

Person



Sun model for PersonBorn fact

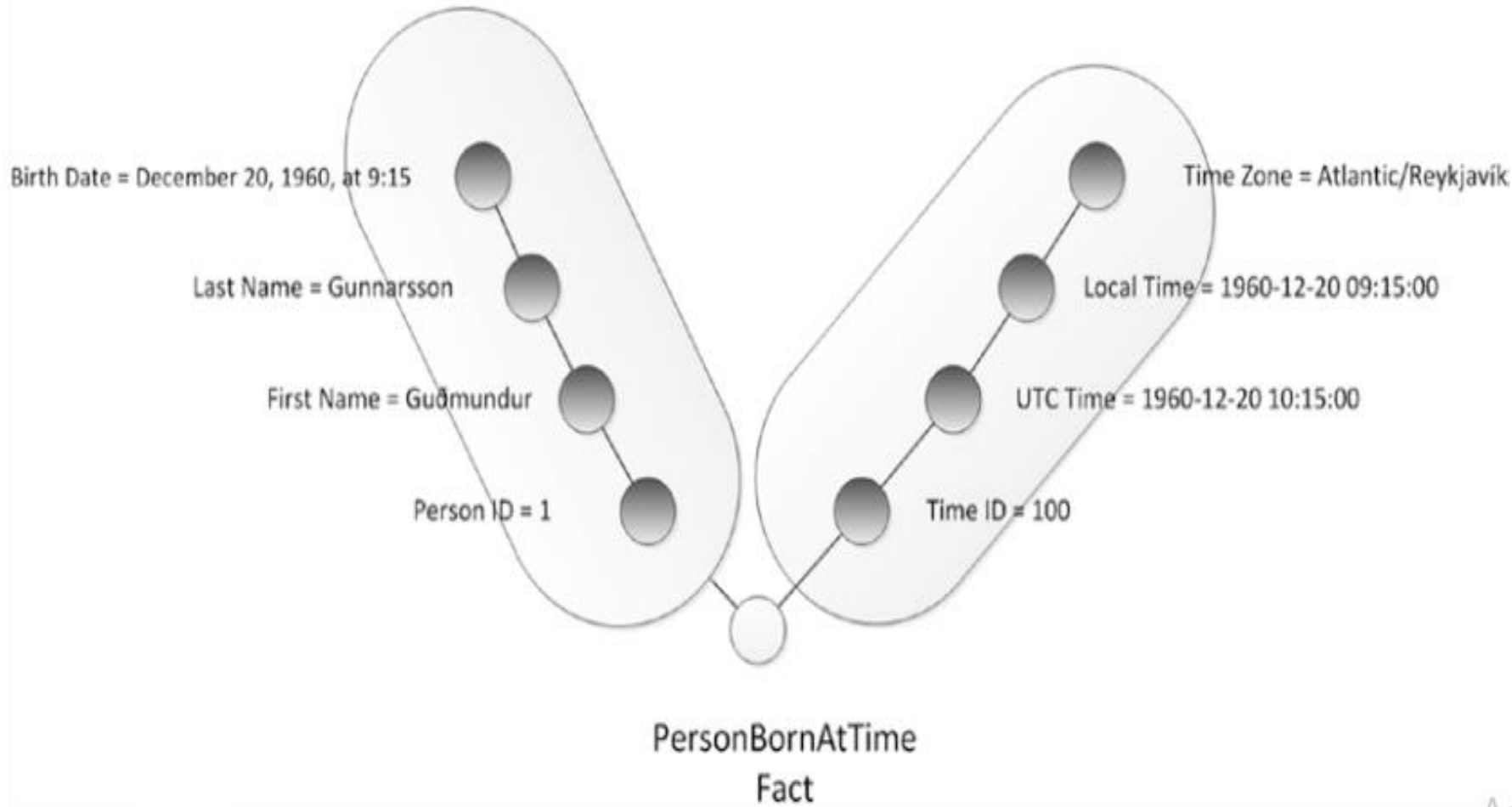
Activate

EXAMPLE OF DIMENSION CONSOLIDATION USING SUN MODEL (TRANSFORM STEP)

- ▶ You must build three items: dimension Person, dimension Time, and fact PersonBornAtTime.

Person Dimension

Time Dimension



Person-to-Time sun model (explained)

Activate
Go to Sett

```
import sys
import os
from datetime import datetime
from pytz import timezone
import pandas as pd
import sqlite3 as :
import uuid
pd.options.mode.chained_assignment = None
#####
if sys.platform == 'linux' or sys.platform == ' Darwin':
    Base=os.path.expanduser('~') + '/VKHCG'
else:
    Base='C:/VKHCG'
print('#####')
print('Working Base :',Base, ' using ', sys.platform)
print('#####')
#####
Company='01-Vermeulen'
#####
sDataBaseDir=Base + '/' + Company + '/04-Transform/SOLite'
```

pytz module allows for date-time conversion and timezone calculations so that your python applications can keep track of dates and times, while staying accurate to the timezone of a particular location.

```
if not os.path.exists(sDataBaseDir):
```

```
    os.makedirs(sDataBaseDir)
```

```
sDatabaseName=sDataBaseDir + '/Vermeulen.db'
```

```
conn1 = sq.connect(sDatabaseName)
```

```
#####
```

```
sDataWarehouseDir=Base + '/99-DW'
```

```
if not os.path.exists(sDataWarehouseDir):
```

```
    os.makedirs(sDataWarehouseDir)
```

```
sDatabaseName=sDataWarehouseDir + '/datawarehouse.db'
```

```
conn2 = sq.connect(sDatabaseName)
```

```
print('\n#####')
```

```
print('Time Dimension')
```

```
BirthZone = 'Atlantic/Reykjavik'
```

```
BirthDateUTC = datetime(1960,12,20,10,15,0)
```

```
BirthDateZoneUTC=BirthDateUTC.replace(tzinfo=timezone('UTC'))
```

```
BirthDateZoneStr=BirthDateZoneUTC.strftime("%Y-%m-%d %H:%M:%S")
```

```
BirthDateZoneUTCStr=BirthDateZoneUTC.strftime("%Y-%m-%d %H:%M:%S (%Z)  
(%z)")
```

```
BirthDate = BirthDateZoneUTC.astimezone(timezone(BirthZone))
```

```
BirthDateStr=BirthDate.strftime("%Y-%m-%d %H:%M:%S (%Z) (%z)")
```

```
BirthDateLocal=BirthDate.strftime("%Y-%m-%d %H:%M:%S")
```

```
#####
```

```
IDTimeNumber=str(uuid.uuid4())
```

```
TimeLine=[('TimeID', [IDTimeNumber]),
```

```
('UTCDate', [BirthDateZoneStr]),
```

```
('LocalTime', [BirthDateLocal]),
```

```
('TimeZone', [BirthZone])]
```

```
TimeFrame = pd.DataFrame.from_items(TimeLine)
#####
DimTime=TimeFrame
DimTimeIndex=DimTime.set_index(['TimeID'],inplace=False)
sTable = 'Dim-Time'
print('\n#####')
print('Storing :',sDatabaseName,'\n Table:',sTable)
print('\n#####')
DimTimeIndex.to_sql(sTable, conn1, if_exists="replace")
DimTimeIndex.to_sql(sTable, conn2, if_exists="replace")

print('\n#####')
print('Dimension Person')
print('\n#####')
FirstName = 'Guðmundur'
LastName = 'Gunnarsson'
#####
IDPersonNumber=str(uuid.uuid4())
```

`Pandas.DataFrame.to_sql`

Write records stored in a dataframe to a SQL database.

```
PersonLine=[('PersonID', [IDPersonNumber]),  
( 'FirstName', [FirstName]),  
( 'LastName', [LastName]),  
( 'Zone', ['UTC']),  
( 'DateTimeValue', [BirthDateZoneStr])]  
PersonFrame = pd.DataFrame.from_items(PersonLine)  
#####  
DimPerson=PersonFrame  
DimPersonIndex=DimPerson.set_index(['PersonID'],inplace=False)
```

Pandas `set_index` is a method to set a list, series or data frame as index of a data frame.

If a data frame is made out of two or more data frames then later index can be changed using this method.


```
sTable = 'Dim-Person'
print('\n#####')
print('Storing :',sDatabaseName,'\n Table:',sTable)
print('\n#####')
DimPersonIndex.to_sql(sTable, conn1, if_exists="replace")
DimPersonIndex.to_sql(sTable, conn2, if_exists="replace")
print('\n#####')
print('Fact - Person - time')
print('\n#####')
IDFactNumber=str(uuid.uuid4())
PersonTimeLine=[('IDNumber', [IDFactNumber]),
('IDPersonNumber', [IDPersonNumber]),
('IDTimeNumber', [IDTimeNumber])]
PersonTimeFrame = pd.DataFrame.from_items(PersonTimeLine)
#####
FctPersonTime=PersonTimeFrame
FctPersonTimeIndex=FctPersonTime.set_index(['IDNumber'],inplace=False)
#####
```

.....

```
sTable = 'Fact-Person-Time'
```

```
print("\n#####")
```

```
print('Storing:',sDatabaseName,'\n Table:',sTable)
```

```
print("\n#####")
```

```
FctPersonTimeIndex.to_sql(sTable, conn1, if_exists="replace")
```

```
FctPersonTimeIndex.to_sql(sTable, conn2, if_exists="replace")
```


TRANSFORMING WITH DATASCIENCE

- ▶ Transform your data into insights.

Steps of Data Exploration and Preparation

- ▶ Missing Value Treatment
- ▶ We must describe the missing value treatment in the transformation. The missing value treatment must be acceptable by the business community.

Steps of Data Exploration and Preparation

- ▶ Why Missing Value Treatment Is Required
- ▶ It can lead to wrong prediction or classification.

Steps of Data Exploration and Preparation

Why Data Has Missing Values

- ▶ Data fields were renamed during upgrades
- ▶ Mappings were incomplete during the migration processes from old systems to new systems
- ▶ Wrong table name was provided during loading
- ▶ Data was not available

Methods to treat missing values

- ▶ Outlier Detection and Treatment
- ▶ Expected “Yes” or “No” but found some “N/A”s, or you expected number ranges between 1 and 10 but got 11, 12, and 13 also.
- ▶ These out-of-order items are the outliers.

Elliptic Envelope

- ▶ A function called `EllipticEnvelope` is one of the more common techniques used to detect outliers in a Gaussian distributed data set.
- ▶ `EllipticEnvelope(support_fraction=1., contamination=0.261)`. The `support_fraction` is the portion of the complete population you want to use to determine the border between inliers and outliers.

Elliptic Envelope

- ▶ The contamination is the indication of what portion of the population could be outliers, hence, the amount of contamination of the data set, i.e., the proportion of outliers in the data set.

Isolation Forest

- ▶ One efficient way of performing outlier detection in high-dimensional data sets is to use random forests. The ensemble.IsolationForest tool “isolates” observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.

Novelty Detection

- ▶ Novelty detection simply performs an evaluation in which we add one more observation to a data set.
- ▶ Is the new observation so different from the others that we can doubt that it is regular? (I.e., does it come from the same distribution?)
- ▶ The `sklearn.svm.OneClassSVM` tool is a good example of this unsupervised outlier detection technique.

Local Outlier Factor

- ▶ An efficient way to perform outlier detection on moderately high-dimensional data sets is to use the local outlier factor (LOF) algorithm. The neighbors.
- ▶ LocalOutlierFactor algorithm computes a score (called a local outlier factor) reflecting the degree of abnormality of the observations.

What Is Feature Engineering?

- ▶ Feature engineering is your core technique to determine the important data characteristics in the data lake and ensure they get the correct treatment through the steps of processing.

Common Feature Extraction Techniques

▶ Binning

- ▶ Binning technique is used to reduce the complexity of data sets, to enable the data scientist to evaluate the data with an organized grouping technique.

Common Feature Extraction Techniques

- ▶ Averaging
- ▶ The use of averaging enables you to reduce the amount of records you require to report any activity that demands a more indicative, rather than a precise, total.

Example: Create a model that enables you to calculate the average position for ten sample points

```
import numpy as np
import pandas as pd
#Create two series to model the latitude and longitude ranges.
LatitudeData = pd.Series(np.array(range(-90,91,1)))
LongitudeData = pd.Series(np.array(range(-14.20,14.21,1)))
#Select 10 samples for each range:
LatitudeSet=LatitudeData.sample(10)
LongitudeSet=LongitudeData.sample(10)
#Calculate the average of each data set
LatitudeAverage = np.average(LatitudeSet)
LongitudeAverage = np.average(LongitudeSet)
#See the results
```

Example: Create a model that enables you to calculate the average position for ten sample points

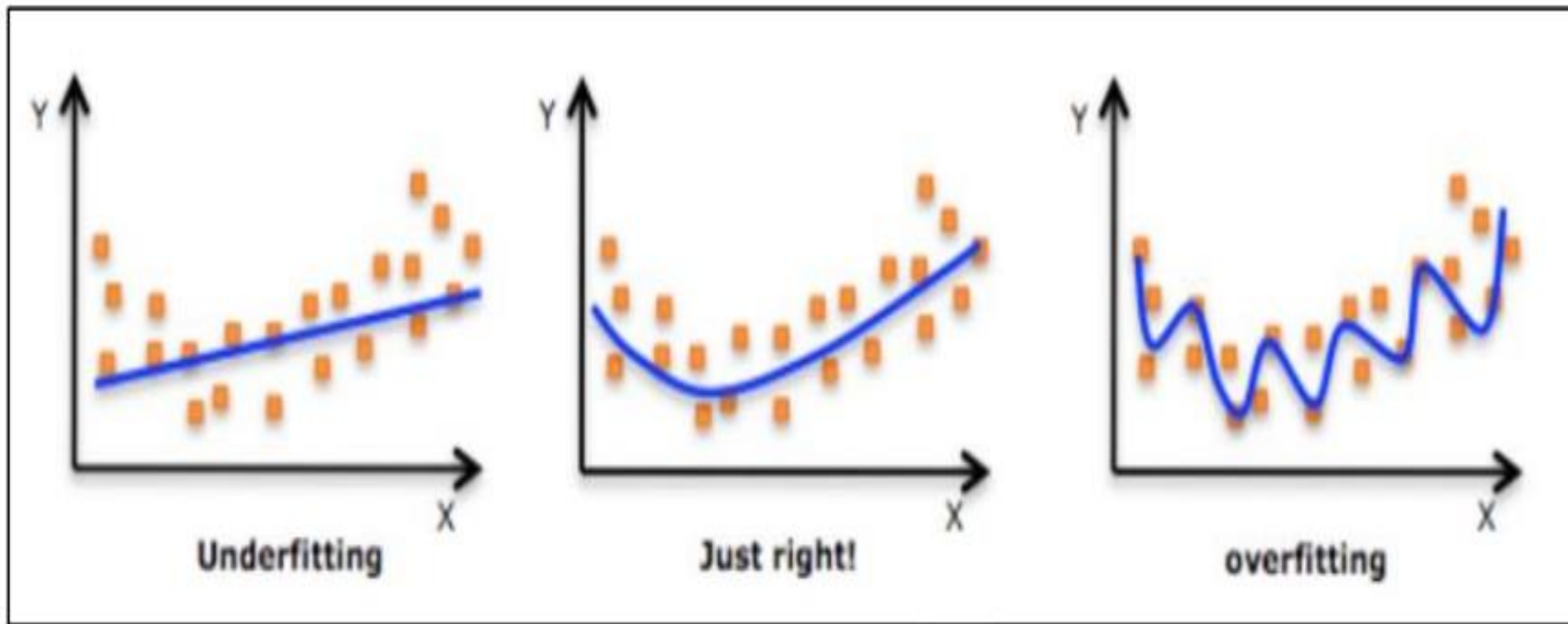
```
print('Latitude')
print(LatitudeSet)
print('Latitude (Avg):',LatitudeAverage)
print('#####')
print('Longitude')
print(LongitudeSet)
print('Longitude (Avg):', LongitudeAverage)
```

OVERFITTING AND UNDERFITTING

- ▶ They refer to the deficiencies that the model's performance might suffer from.
- ▶ Overfitting occurs when the model or the algorithm fits the data too well. When a model gets trained with so much of data, it starts learning from the noise and inaccurate data entries in our data set.
- ▶ But the problem then occurred is, the model will not be able to categorize the data correctly, and this happens because of too much of details and noise.

OVERFITTING AND UNDERFITTING

- ▶ Underfitting occurs when the model or the algorithm cannot capture the underlying trend of the data.
- ▶ Intuitively, underfitting occurs when the model or the algorithm does not fit the data well enough. It is often a result of an excessively simple model. It destroys the accuracy of our model.



Overfitting & Underfitting

Polynomial Features

- ▶ The polynomial formula is the following:
- ▶ $(a_1x + b_1)(a_2x + b_2) = a_1a_2x^2 + (a_1b_2 + a_2b_1)x + b_1b_2$.
- ▶ The polynomial feature extraction can use a chain of polynomial formulas to create a hyperplane that will subdivide any data sets into the correct cluster groups. The higher the polynomial complexity, the more precise the result that can be achieved.

Example:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import Ridge
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline

def f(x):
    """ function to approximate by polynomial interpolation """
    return x * np.sin(x)

# generate points used to plot
x_plot = np.linspace(0, 10, 100)
# generate points and keep a subset of them
x = np.linspace(0, 10, 100)
rng = np.random.RandomState(0)
rng.shuffle(x)
x = np.sort(x[:20])
y = f(x)
# create matrix versions of these arrays
X = x[:, np.newaxis]
```

`sklearn.linear_model.Ridge` is the module used to solve a regression model.

Sklearn's `PolynomialFeatures` is a tool that can help you transform your input data into a polynomial form, making it suitable for polynomial regression.

It can help you capture nonlinear relationships between variables and improve the performance of your models.

The Scikit-learn pipeline is a tool that links all steps of data manipulation together to create a pipeline. It will shorten your code and make it easier to read and adjust.

```
colors = ['teal', 'yellowgreen', 'gold']
```

```
lw = 2
```

```
plt.plot(x_plot, f(x_plot), color='cornflowerblue', linewidth=lw, label="Ground Truth")
```

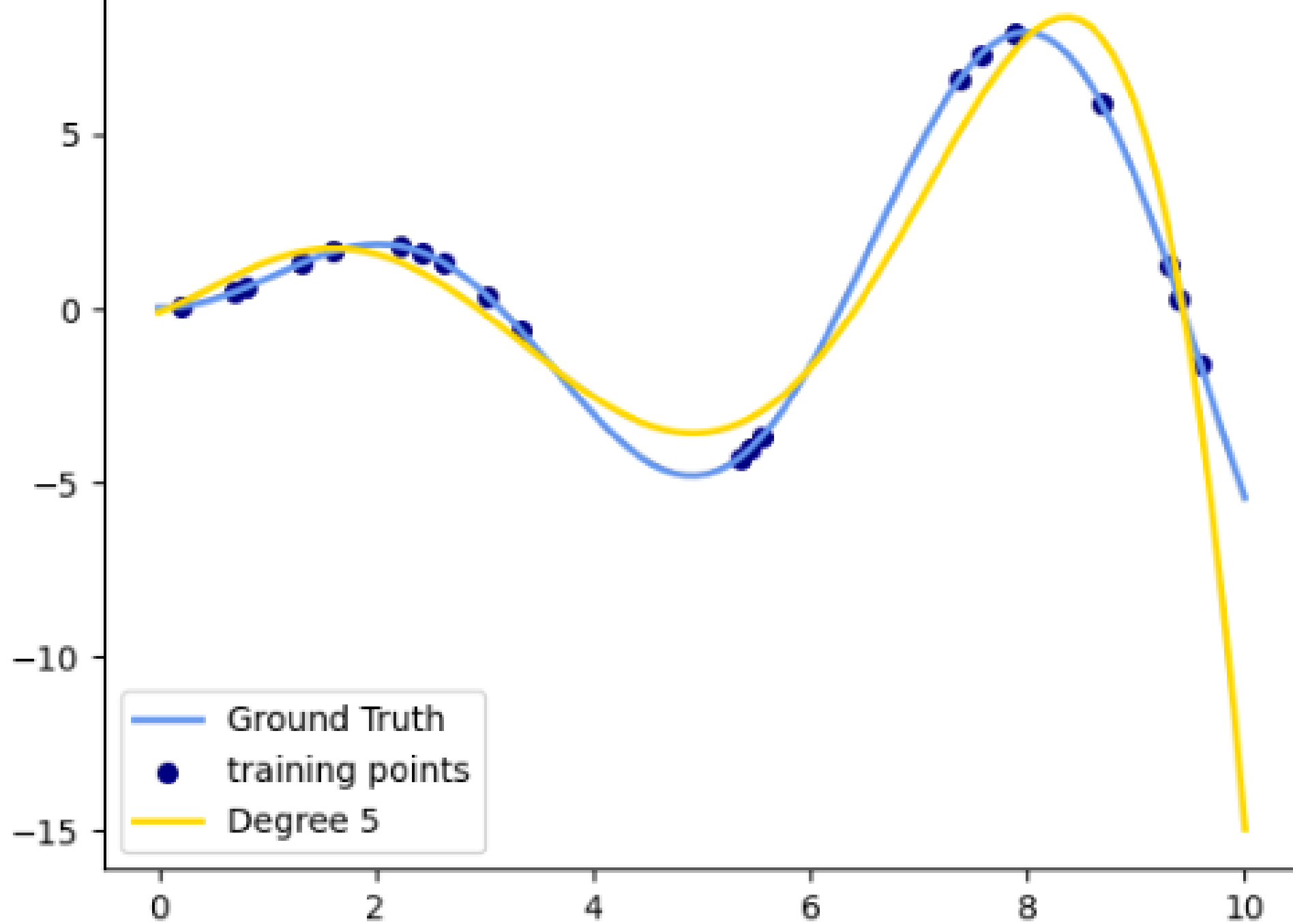
```
plt.scatter(x, y, color='navy', s=30, marker='o', label="training points")
```

```
for count, degree in enumerate([3, 4, 5]):
```

```
model = make_pipeline(PolynomialFeatures(degree), Ridge())  
model.fit(X, y)  
y_plot = model.predict(X_plot)  
plt.plot(x_plot, y_plot, color=colors[count], linewidth=lw, label="Degree %d" %  
degree)  
plt.legend(loc='lower left')  
plt.show()
```


A legend is an area describing the elements of the graph.

The attribute `Loc` in `legend()` is used to specify the location of the legend.



Common Data-Fitting Issue

- ▶ These higher order polynomial formulas are, however, more prone to overfitting, while lower order formulas are more likely to underfit.

HYPOTHESIS TESTING

- ▶ Hypothesis testing is a statistical test to check if a hypothesis is true based on the available data. Based on testing, data scientists choose to accept or reject (not accept) the hypothesis.

T-TEST

- ▶ The t-test is one of many tests used for the purpose of hypothesis testing in statistics.
- ▶ A t-test is a popular statistical test to make inferences about single means or inferences about two means or variances, to check if the two groups' means are statistically different from each other.
- ▶ The One Sample t Test determines whether the sample mean is statistically different from a known or hypothesised population mean. The One Sample t Test is a parametric test.

H0: Mean age of given sample is 30.

H1: Mean age of given sample is not 30

#pip3 install scipy

#pip3 install numpy

```
from scipy.stats import ttest_1samp
```

```
import numpy as np
```

```
ages = np.genfromtxt('ages.csv')
```

```
print(ages)
```

```
ages_mean = np.mean(ages)
```

```
print("Mean age:",ages_mean)
```

```
print("Test 1: m=30")
```

```
tset, pval = ttest_1samp(ages, 30)
```

```
print('p-values - ',pval)
```

```
if pval< 0.05:
```

```
    print("we reject null hypothesis")
```

```
else:
```

```
    print("we fail to reject null hypothesis")
```

`scipy.stats.ttest_1samp`

This is a test for the null hypothesis that the expected value (mean) of a sample of independent observations a is equal to the given population mean.

The `genfromtxt()` function is used to load data in a program from a text file. It takes multiple argument values to clean the data of the text file. It also has the ability to deal with missing or null values through the processes of filtering, removing, and replacing.

p-value

- ▶ A p-value measures the probability of obtaining the observed results, assuming that the null hypothesis is true.
- ▶ The lower the p-value, the greater the statistical significance of the observed difference. A p-value of 0.05 or lower is generally considered statistically significant.


```
[20. 30. 25. 13. 16. 17. 34. 35. 38. 43. 45. 48. 49. 50. 51. 54. 55. 56.  
59. 61. 62. 18. 22. 29.]
```

Mean age: 38.75

Test 1: m=30

p-values = 0.01333239479255858

we reject null hypothesis

Chi-Square Test

- ▶ A chi-square test is used to check if two variables are significantly different from each other. These variables are categorical.

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Chi-Square Test

- ▶ Usually, it is a comparison of two statistical data sets. This test was introduced by **Karl Pearson** in 1900 for categorical data analysis and distribution.

```
import numpy as np
import pandas as pd
import scipy.stats as stats
np.random.seed(10)
stud_grade = np.random.choice(a=["O","A","B","C","D"],
p=[0.20, 0.20, 0.20, 0.20, 0.20], size=100)
stud_gen = np.random.choice(a=["Male","Female"], p=[0.5, 0.5], size=100)
mscpart1 = pd.DataFrame({"Grades":stud_grade, "Gender":stud_gen})
print(mscpart1)
stud_tab = pd.crosstab(mscpart1.Grades, mscpart1.Gender, margins=True)
stud_tab.columns = ["Male", "Female", "row_totals"]
stud_tab.index = ["O", "A", "B", "C", "D", "col_totals"]
observed = stud_tab.iloc[0:5, 0:2 ]
print(observed)
expected = np.outer(stud_tab["row_totals"][0:5],
stud_tab.loc["col_totals"][0:2]) / 100
print(expected)
chi_squared_stat = (((observed-expected)**2)/expected).sum().sum()
print('Calculated : ',chi_squared_stat)
crit = stats.chi2.ppf(q=0.95, df=4)
```

NumPy's random.seed() function initializes the random number generator with the specified seed value.

With the help of **choice()** method, we can get the random samples of one dimensional array and return the random samples of numpy array.

A *crosstab* is a table showing the relationship between two or more variables. Where the table only shows the relationship between two categorical variables.

pandas.DataFrame.iloc: Purely integer-location based indexing for selection by position.

Pandas **DataFrame.loc** attribute access a group of rows and columns by label(s) or a boolean array in the given [Pandas DataFrame](#).

The method `norm.ppf()` accepts a percentage and returns a standard deviation multiplier for the value that percentage occurs at.

```

→
      Grades  Gender
0         C  Female
1         O  Female
2         C   Male
3         C   Male
4         B  Female
..      ...    ...
95        B   Male
96        D  Female
97        B  Female
98        A   Male
99        B   Male

```

```
[100 rows x 2 columns]
```

	Male	Female
O	11	12
A	9	13
B	7	11
C	10	8
D	12	7

```

[[11.27 11.73]
 [10.78 11.22]
 [ 8.82  9.18]
 [ 8.82  9.18]
 [ 9.31  9.69]]

```

```

Calculated : 3.158915138993211
Table Value : 9.487729036781154
H0 is Rejected

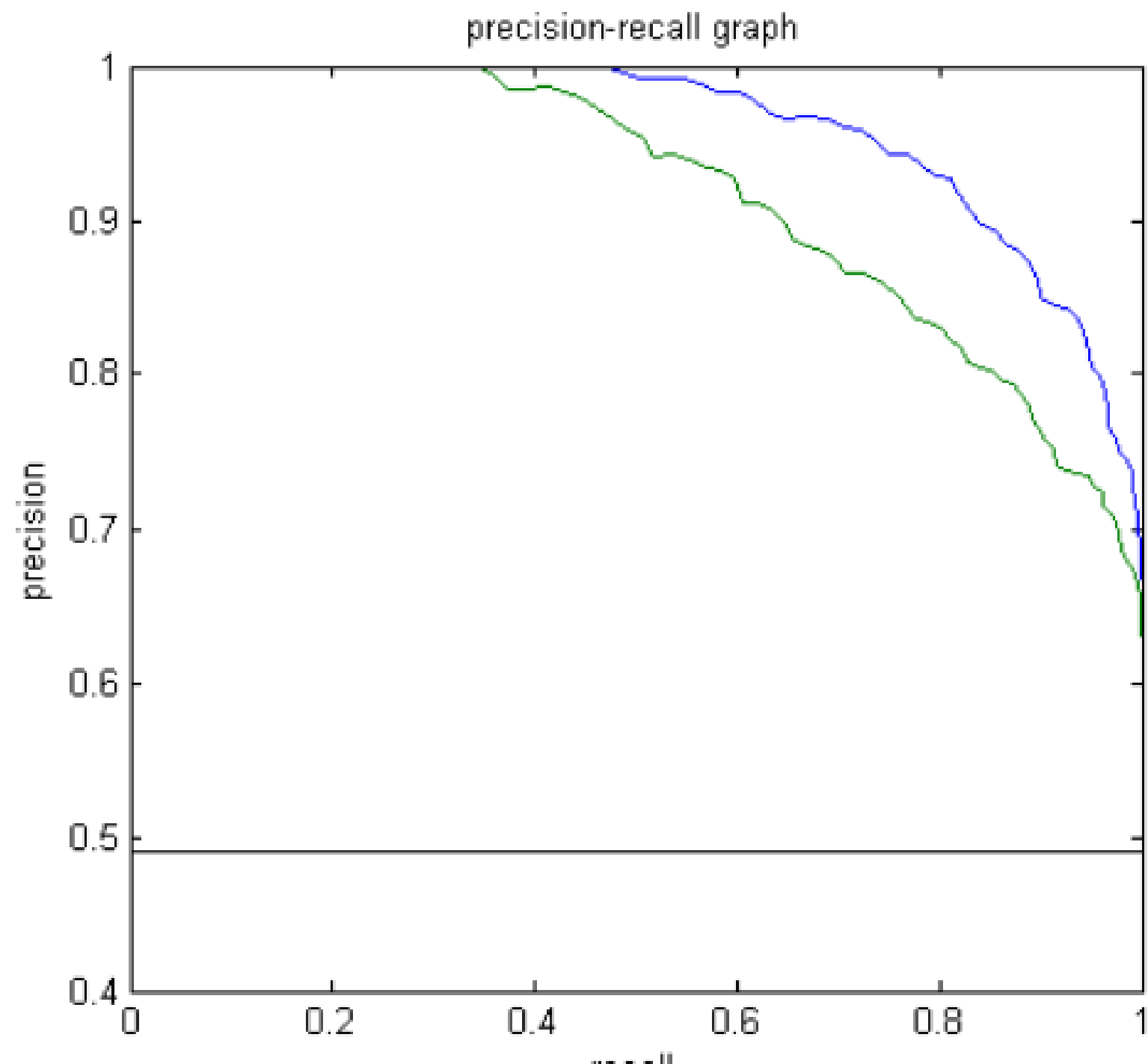
```

PRECISION -RECALL

- ▶ Precision-recall is a useful measure for successfully predicting when classes are extremely imbalanced.
- ▶ In information retrieval, • Precision is a measure of result relevancy.
- ▶ • Recall is a measure of how many truly relevant results are returned.

Precision-Recall Curve

- ▶ A PR curve is simply a graph with Precision values on the y-axis and Recall values on the x-axis. In other words, the PR curve contains $TP/(TP+FP)$ on the y-axis and $TP/(TP+FN)$ on the x-axis.
- ▶ It is important to note that Precision is also called the Positive Predictive Value (PPV).
- ▶ Recall is also called Sensitivity, Hit Rate or True Positive Rate (TPR).



PRECISION-RECALL CURVE

- ▶ A system with high recalls but low precision returns many results, but most of its predicted labels are incorrect when compared to the training labels.
- ▶ A system with high precision but low recall is just the opposite, returning very few results, but most of its predicted labels are correct when compared to the training labels.
- ▶ An ideal system with high precision and high recall will return many results, with all results labelled correctly.

- True Positive (TP): is the result that we get if we correctly predict the positive class.
- False Positive (FP): is the outcome that we get if we predict a negative class as a positive class.
- True Negative (TN): is the result that we get if we correctly predict the negative class.
- False Negative (FN): is the outcome that we get if we predict a positive class as a negative class.

FEW IMPORTANT TERMS

- ▶ PRECISION
- ▶ RECALL
- ▶ TRUE NEGATIVE RATE
- ▶ ACCURACY


PRECISION

- ▶ Precision (P) is defined as the number of true positives (Tp) over the number of true positives (Tp) plus the number of false positives (Fp).

$$P = \frac{Tp}{Tp + Fp}$$

RECALL

- ▶ Recall (R) is defined as the number of true positives (Tp) over the number of true positives (Tp) plus the number of false negatives (Fn)


$$R = \frac{Tp}{Tp + Fn}$$

TRUE NEGATIVE RATE

- ▶ The true negative rate (TNR) is the rate that indicates the recall of the negative items.

$$TNR = \frac{Tn}{Tn + Fp}$$

ACCURACY

$$A = \frac{Tp + Tn}{Tp + Fp + Tn + Fn}$$

Sensitivity & Specificity

- ▶ Sensitivity and specificity are statistical measures of the performance of a binary classification test, also known in statistics as a classification function.
- ▶ Sensitivity (also called the true positive rate, the recall, or probability of detection) measures the proportion of positives that are correctly identified as such (e.g., the percentage of sick people who are correctly identified as having the condition).

Sensitivity & Specificity

- ▶ Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified as such (e.g., the percentage of healthy people who are correctly identified as not having the condition).

F1-MEASURE

- ▶ The F1-score is a measure that combines precision and recall in the harmonic mean of precision and recall.

$$F1 = 2 * \frac{P * R}{P + R}$$

Harmonic mean of 1, 4, and 4, you would divide the number of observations by the reciprocal of each number.

$$3 / (1/1 + 1/4 + 1/4) = 2$$

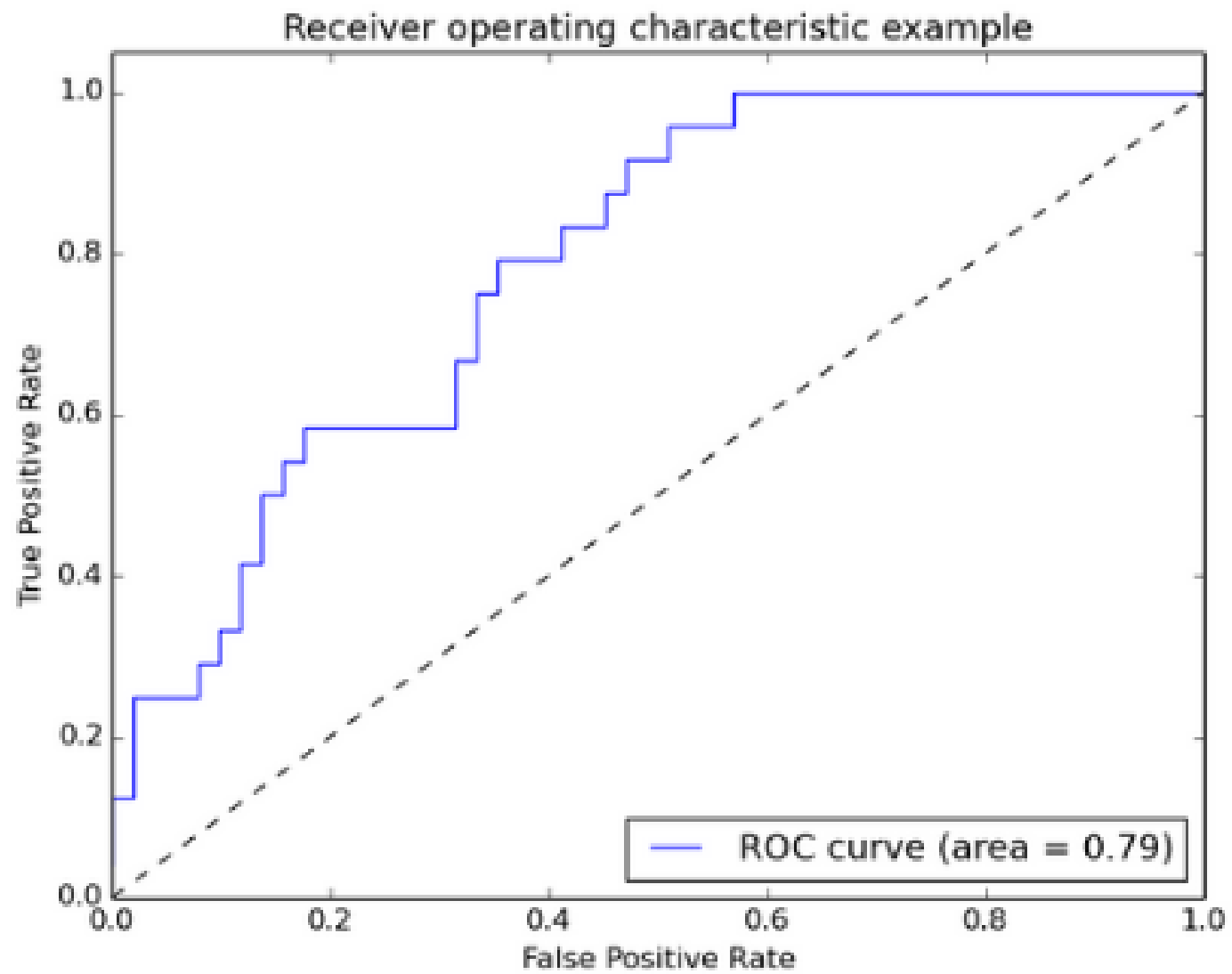
The harmonic mean has uses in finance and technical analysis of markets, among others.

Contt..

- ▶ The following sklearn functions are useful when calculating these measures:
- ▶ `sklearn.metrics.average_precision_score`
- ▶ `sklearn.metrics.recall_score`
- ▶ `sklearn.metrics.precision_score`
- ▶ `sklearn.metrics.f1_score`.

Receiver Operating Characteristic (ROC) Analysis Curves

- ▶ The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.
- ▶ The true positive rate is also known as sensitivity, recall, or probability of detection.
- ▶ You will find the ROC analysis curves useful for evaluating whether your classification or feature engineering is good enough to determine the value of the insights you are finding. This helps with repeatable results against a real-world data set.



PRECISION-RECALL VS ROC

- Precision Recall curve is used when there is imbalance class distribution.
- ROC-AUC curve is used when there is balanced class distribution in data.

Examples of imbalanced datasets

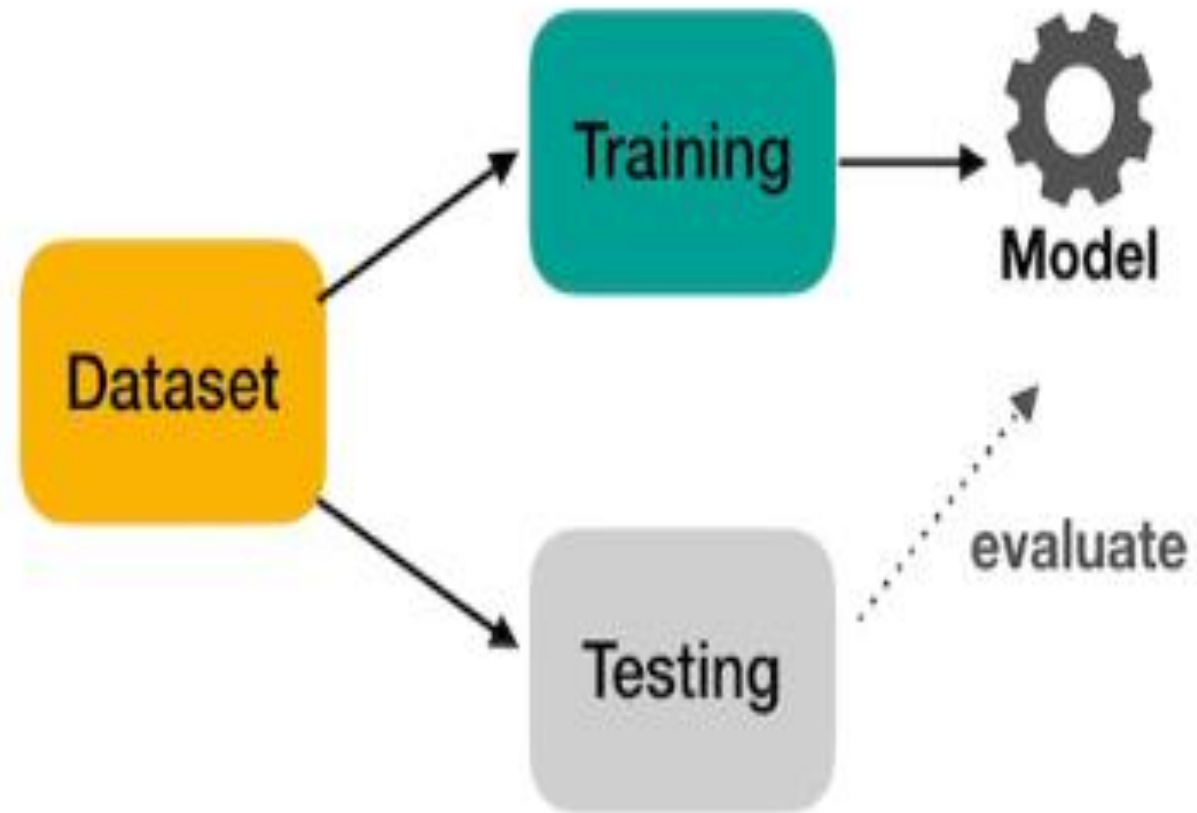
- ▶ Data sets to identify rare diseases in medical diagnostics etc.
- ▶ Total Observations = 1000
- ▶ Fraudulent Observations = 20
- ▶ Non-Fraudulent Observations = 980

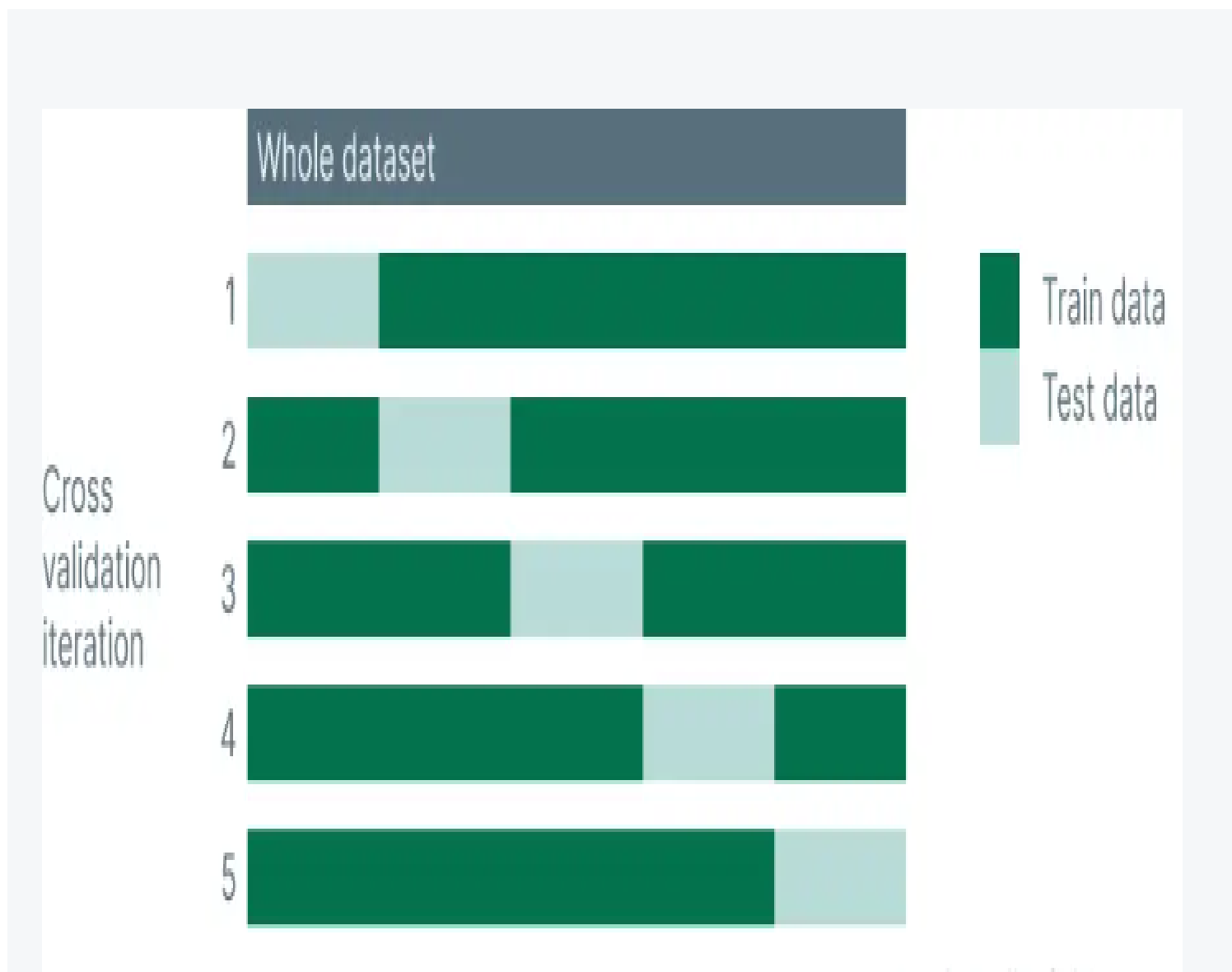
CROSS-VALIDATION TEST

- ▶ “Cross-Validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two parts, one was used to learn or train our model and the other was used to validate our model.”

NEED OF CROSS-VALIDATION TEST

- ▶ When the training dataset gives us good accuracy and whenever the new data comes then it's not able to give us good accuracy then, in that case, our model will be overfitted.
- ▶ To handle this type of problem, Cross-Validation comes into the picture.





STEPS OF CROSS-VALIDATION TESTING

- ▶ The number of folds is defined, by default this is 5
- ▶ The dataset is split up according to these folds, where each fold has a unique set of testing data.
- ▶ A model is trained and tested for each fold.
- ▶ Each fold returns a metric for its test data.
- ▶ The mean and standard deviation of these metrics can then be calculated to provide a single metric for the process.

Example:

```
import numpy as np
from sklearn.model_selection import cross_val_score
from sklearn import datasets, svm
import matplotlib.pyplot as plt
```

```
digits = datasets.load_digits()
X = digits.data
y = digits.target
```


Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

```
kernels=['linear', 'poly', 'rbf']  
for kernel in kernels:  
    svc = svm.SVC(kernel=kernel)  
    C_s = np.logspace(-15, 0, 15)  
    scores = list()  
    scores_std = list()  
    for C in C_s:  
        svc.C = C  
        this_scores = cross_val_score(svc, X, y, n_jobs=1)  
        scores.append(np.mean(this_scores))  
        scores_std.append(np.std(this_scores))
```

Cross_val_score is a function in the scikit-learn package which trains and tests a model over multiple folds of your dataset.

This cross validation method gives you a better understanding of model performance over the whole dataset instead of just a single train/test split.

You must plot your results.

```
Title="Kernel:>" + kernel
fig=plt.figure(1, figsize=(4.2, 6))
plt.clf()
fig.suptitle(Title, fontsize=20)
plt.semilogx(C_s, scores)
plt.semilogx(C_s, np.array(scores) + np.array(scores_std), 'b--')
plt.semilogx(C_s, np.array(scores) - np.array(scores_std), 'b--')
locs, labels = plt.yticks()
plt.yticks(locs, list(map(lambda x: "%0g" % x, locs)))
plt.ylabel('Cross-Validation Score')
plt.xlabel('Parameter C')
plt.ylim(0, 1.1)
plt.show()
```

Matplotlib.pyplot.semilogx() Function

This function is used to visualize data in a manner that the x-axis is converted to log format.

THANKS