Data preparation phase - Overview

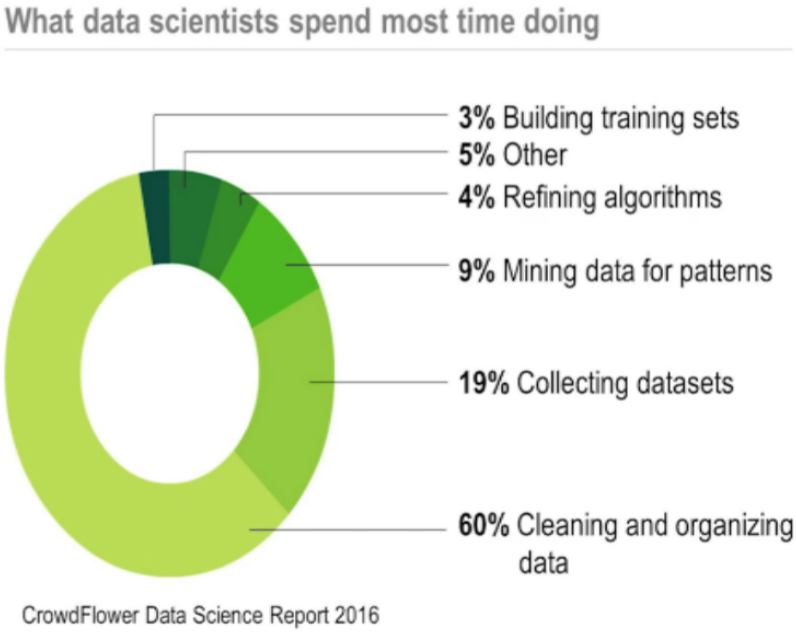
- refered to as "data wrangling", or "data munging" or "data janitor work".
- Includes everything from list verification to removing commas in the data, and debugging databases.
- Cleaning data is very time consuming

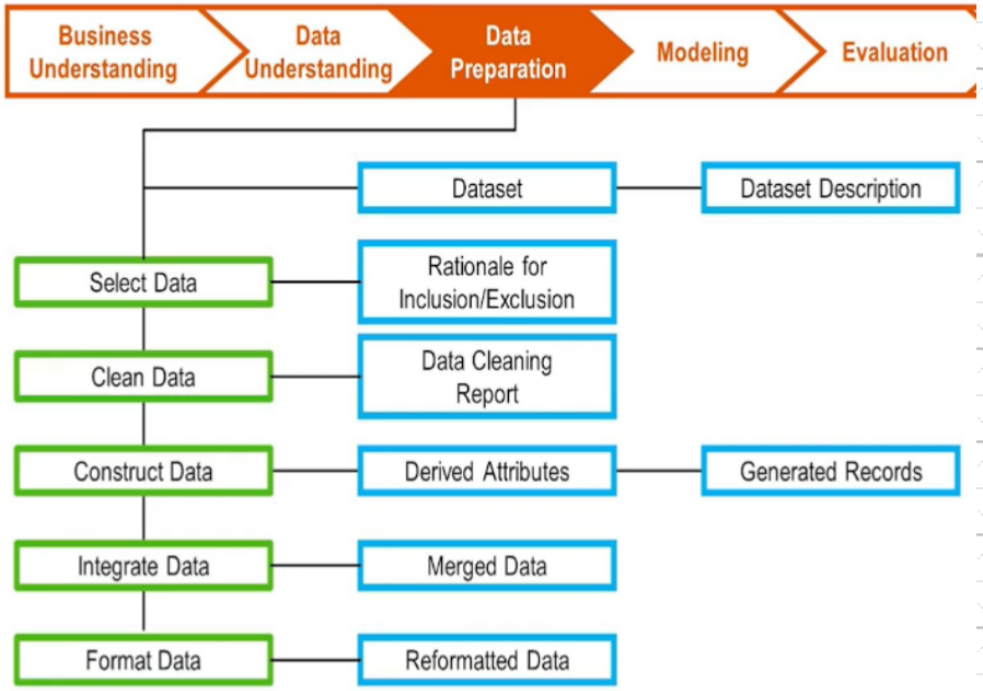
Introduction to data preparation

The chart below shows that 3 out of every 5 data scientists spend most time during their working day cleaning and organizing data.

A New York Times article reported that data scientists spend from **50% to 80%** of their time mired in the more mundane task of collecting and preparing unruly digital data before it can be explored for useful nuggets.

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights.
New York Times. STEVE LOHR.
AUG. 17, 2014





Data prep Phase - This phase covers all the activities to construct the final analytical dataset from the initial raw data.

Data preparation tasks are likely to be performed multiple times and not in any prescribed order.

Tasks Include:

- Table
- Record
- Attribute selection
- transformations
- Data cleaning for the modeling tool

Select Data task - Decides on the data that can be used for analysis.
The criteria to choose data includes the:
- Relevance to the data mining goals
- Data quality
- technical constraints.

And this covers Attribute selection as well as the selection of records in a table.

The cleam data task - raises the data quality to the level that's required for the seleted analysis

The construct Data task - Includes the data preparation operations, such as the production of derived attributes, entire new records, or transformed values for existing attributes.

The integrate Data task combines information from multiple tables or records to create new records or values

Format Data task - produces the transformations that are primarily syntactic modifications made to the data, that don't actually change its meaning, but might be required by the modeling tool.

Data Preparation-Phase 3: Outputs

- Dataset
  - This is the dataset (or datasets) produced by the Data Preparation phase, which will be used for modeling or the major analysis work of the project.
- Dataset description
  - Describes the dataset (or datasets) that will be used for the modeling or the major analysis work of the project.

Phase 3.1: Select Data

- Tasks
  - Decide on the data to be used for analysis
  - Criteria include relevance to the data science goals and quality and technical constraints such as limits on →

data volume or data types.

- Note that data selection covers selection of attributes (columns) as well as selection of records (rows) in a table

- Output - Rationale for inclusion/exclusion
- List the data to be included/excluded and the reasons for these decisions.

Phase 3.2 - Clean Data

- Task
- Raise the data quality to the level required by the selected analysis techniques.
- This may involve selection of clean subsets of the data, the insertion of suitable defaults or more ambitious techniques such as the estimation missing data by modeling.
- Output - Data cleaning report
- Describe what decisions and actions were taken to address the data quality problems reported during the *Verify Data Quality* task of the Data Understanding phase.

Phase 3.3 - Construct Data

- Task
  - This task includes constructive data preparation operations such as the production of derived attributes, entire new records, or transformed values for existing attributes
- Output - Derived Attributes
  - Derived attributes are new attributes that are constructed from one or more existing attributes in the same record. Examples: area = length * width.
- Output - Generated records
  - Describe the creation of completely new records

Phase 3.4 - Integrate Data

- Task
  - These are methods whereby information is combined from multiple tables or records to create new records or values.
- Output - Merged Data
  - Merging data tables refers to joining together two or

more tables that have different information about the same objects

- Merged data also covers aggregations.

↳ Aggregation refers to operation where new values are computed by summarizing together info from multiple records.

Example:

An aggregation converts a table of customer purchases, where there are multiple purchases per customer with one record for each purchase.

And you aggregate this into a new table where there is **one** record for each customer, and that will include fields such as:

- The number of purchases made
- The average purchase amount
- % of orders that have been charged to a credit card

Phase 3.5 - Format Data

- Task
  - Formatting transformations refer to primarily syntactic modifications made to the data that do not change its meaning, but might be required by the modeling tool.
- Output - Reformatted data
  - Some tools have requirements on the order of the attributes, such as the first field being a unique identifier for each record or the last field being the outcome field the model is to predict.

Example -

- Some tools have requirements on the order of the attributes, such as the first field being the unique identifier for each record, or the last field, the target field, the thing that were trying to predict. It might be important to change the order of the records in the dataset, perhaps the modeling tool requires that the records be sor→

ted according to the value of the outcome attribute.

A common situation is that the records of the dataset are initially ordered in some way but the modeling tool itself needs them to be in a fairly randomized order.

Example

when using neural networks it's generally best for the records to be presented in a random order. And some data science tools, machine Learning tools do this randomizing for you automatically. Additionally, there are purely syntactic changes made to satisfy the requirements of the specific modeling tod

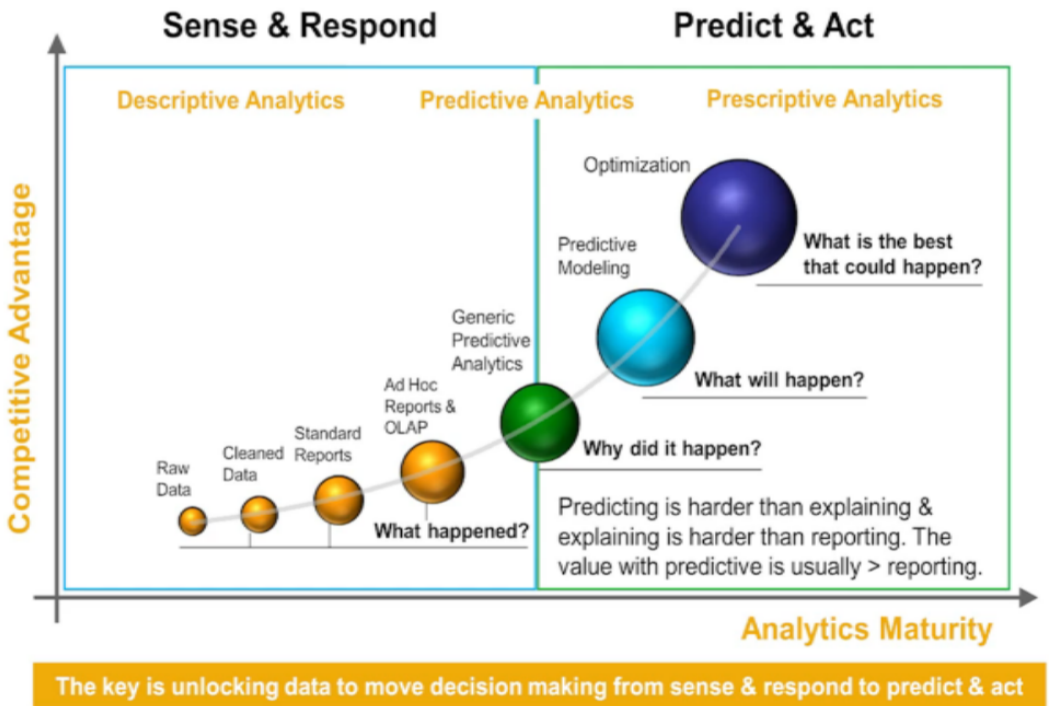
Example

Removing commas from within text fields in comma-delimited data-files is really importa-nt when you're doing text analysis

Predictive Modeling Methodology -

Predictive modeling encompasses a variety of techniques that analyze current and historical facts to make predictions about future or otherwise unknown events. The output of a predictive model is a score or a probability of the targeted event occuring in a specified time frame in the future. Although, most often the unknown event of interest is in the future, predictive analytics can also be applied to any tipe of unknown, whether in the past, present or future

Introduction



Example

- identifying suspects after a crime has been commited or credit card fraud as it occurs.

Descriptive analytics uses data vizualization to provide insight into the past and answer the question "What has happened?".

Descriptive analytics are useful for company reports giving total stock inventory, average dollars spent per customer, and Year-over-Year changes in sales.

Predictive analytics uses statistical models and forecasting techniques to understand the future and answer: "What could happen?".

Predictive analytics analyses patterns in historical data and develops models that predict the probability of events happening in the future.

Prescriptive analytics, which use optimization and simulation algorithims enelde us to answer: "What should we do"→

Prescriptive analytics predicts not only what will happen, but also why it will happen, providing recommendations regarding actions that will take advantage of the predictions. Prescriptive analytics uses a combination of techniques and tools such as:

- Business rules
- Algorithims
- Optimization
- Machine learning
- Mathematical modeling and processing

Use predictive analytics to solve a variety of business challenges

Common use cases for machine learning and data analysis across industries

There are two phases to the predictive modeling process:

Model build

(The training phase)

Predictive models are built or "trained" on historic data with a known outcome

The input variables are called explanatory or independent variables

For model building the outcome, which is called target or dependent variable, is known.

Model apply

(The applying phase)

Once the model has been built, it is applied on to more, more recent data, which has an unknown outcome (because the outcome is in the future)

The model calculates the score or probability of the target category occuring

Eg. Calculates the probability of a customer responding to a market campaign

Building the model – Training phase

Historic Data

Key	Explanatory Variables		Target
Name	**City**	**Age**	**Churner**
Mike	Miami	42	Yes
Jerry	New York	32	No
Bryan	Orlando	18	No
Patricia	Miami	45	Yes
Elodie	Phoenix	35	No
Remy	Chicago	72	Yes

When training a model, the "target" is a flag indicating the known outcome in the past.

This example has a target that is binary – yes or no – so this is a "classification" model.

Classification algorithm to predict probability of churner = yes

Model

Train the Model

PREDICTIVE MODEL EQUATION

Probability of Churn = ($Weight_1$ x City) + ($Weight_2$ x Age)

$Weight_1$ and $Weight_2$ are automatically calculated by the algorithm process to give the most accurate prediction of the known target

Identifies the key characteristics between a churner and a non churner

Using the model – Applying phase

Recent Data

Key	Explanatory Variables		Target
Name	**City**	**Age**	**Churn**
Marine	Miami	45	?
Julien	Miami	52	?
Fred	Orlando	20	?
Michelle	Boston	34	?
Nicolas	Phoenix	90	?

Recent data, with customers who have not yet made a decision to churn or remain.

Model

Apply the Model

PREDICTIVE MODEL EQUATION

Probability of Churn = ($Weight_1$ x City) + ($Weight_2$ x Age)

Name	City	Age	Probability
Marine	Miami	45	**0.8**
Julien	Miami	52	**0.9**
Fred	Orlando	20	**0.6**
Michelle	Boston	34	**0.5**
Nicolas	Phoenix	90	**0.4**

"Apply" the model onto new data to calculate the overall "score" or "probability" for each customer.

When we apply the model onto new data we don't know the target value →

Example moving data through time – Training phase

Imagine a Telco company on July 1st that wants to predict customer churn using a classification model

Training Phase – Build (Train) Model

Historical Data

Client ID	Age	Region	Jan # Trans	Feb # Trans	Mar # Trans
6571001	28	CA	3	4	3
0540112	42	IL	0	5	5
4489613	24	MO	7	8	10
9056724	33	MA	2	0	0
4465785	37	GA	16	14	11

Target

Churn Flag
Yes
No
No
Yes
No

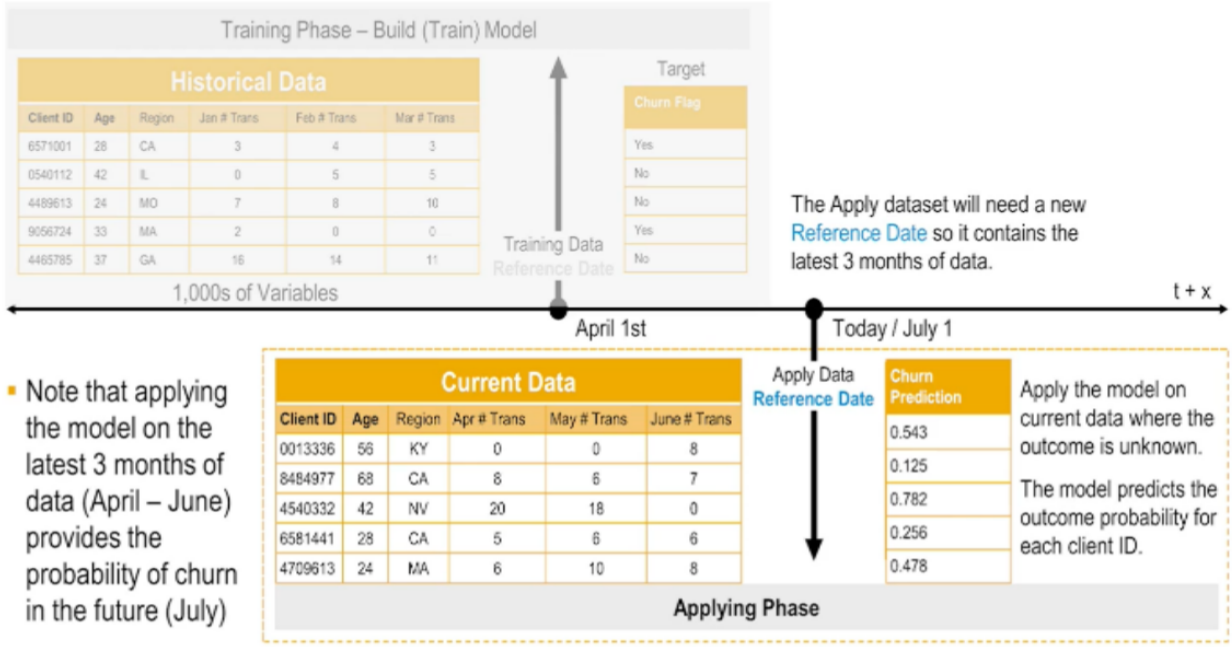
Train the model on historical data where the outcome is known (target).

Set a **Reference Date** to April 1st and train on 3 months historical data to predict what happens in the following month (April).

- Note that the target data time frame (April) occurs AFTER the historical data time frame (January to March)
- The model is trained to identify patterns in the data in the past to predict the target in the following or later months

Example moving data through time – Applying phase

Imagine a Telco company on July 1st that wants to predict customer churn using a classification model



- Note that applying the model on the latest 3 months of data (April – June) provides the probability of churn in the future (July)

Depending on the business requirements models may need to be applied every month, week, day or even minutes or seconds

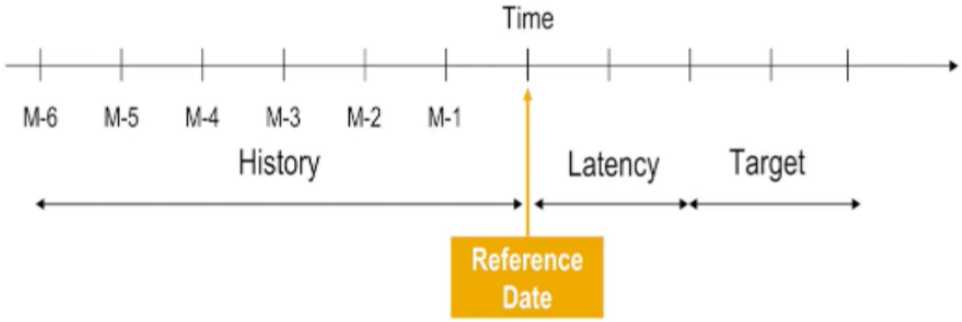
When the data-set changes the reference date also changes so any derived variables like a customer's age must be up-dated relative to the new reference data

Latency

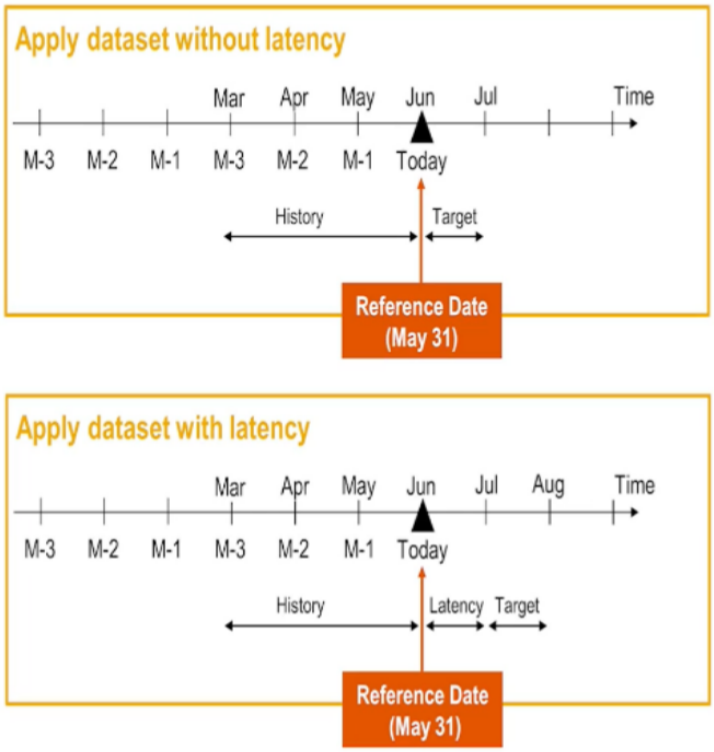
Sometimes when you design a model, you might want to build in a "latency" period.

Many datasets used for predictive modeling have the following structure:

- **Historic Data:** (in the past, compared to the reference date) with dynamic data computed in relation to the reference date. Usually short-term, mid-term, and long-term indicators.
- **Latency Period:** (starting after the reference state) a period where no data is collected. This is used to represent the time required by the business to collect new data, apply the model, produce the scores, and define the campaign. Not all predictive models require a latency period, although many churn models will.
- **Target:** (starting after the reference state + latency period) a period where the targeted behavior is observed.

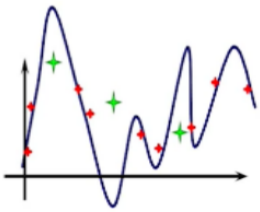


Why is latency needed?

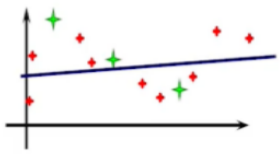


latency Periods can give the business time to convince the customer to stay

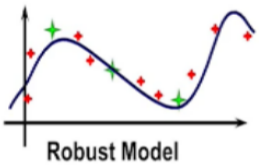
Model fitting



Over-Fit Model/Low Robustness
(No Training Error, High Test Error)



Under-Fit Model/High Robustness
(High Training Error = High Test Error)



Robust Model
(Low Training Error ≈ Low Test Error)

Dont overfit - accurate on training data but not on "real data".

Overfitting occurs when a model is too complex

A model that's been overfited over-reacts to minor fluctuations in the data

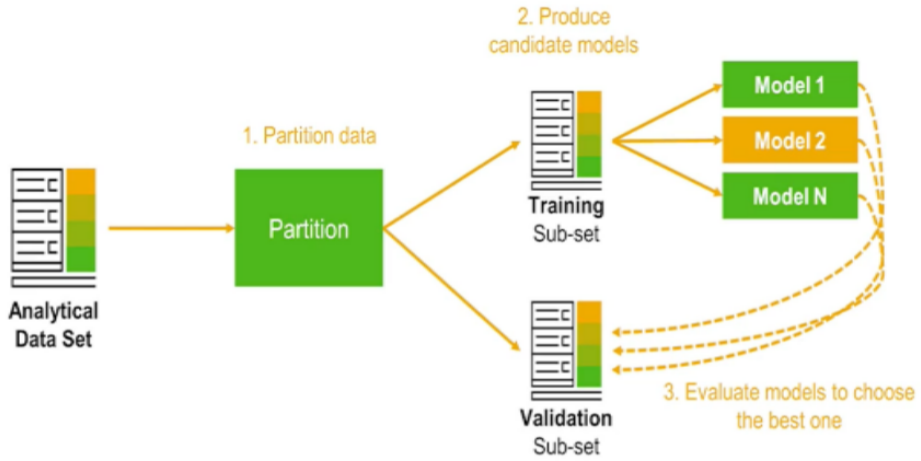
Aditional techniques for avoiding overfitting -

- Cross - validation
- Regularization
- Early stopping
- Pruning decision trees

Dont Underfit

Hold-out sample

- During the Model Build phase, a "hold-out" sample is created.
- This is a sample of observations withheld from the model learning, so that the model's ability to predict future probabilities can be estimated by its ability to predict the data in the hold-out sample.



a hold out sample is data that's separated before training to test and validate the model and its robustness after training

Data manipulation -

- Most data mining activities will require the data to be "prepared" before analysis
- Data manipulation is often driven by domain knowledge.
- This is a process where database tables are merged and aggregated, new variables and transformations are created to try and improve model quality, IF/THEN conditions are created, filters are applied etc...

- The first step is to identify the entity for analysis
  - An entity is the object targeted by the planned analytical task
  - It may be a customer, product, store, etc..., and it is usually identified by a unique identifier

- The entity defines the "granularity" of the analysis

items of significance to an enterprise
are data entities

customer material product

Entity makes business sense
It can be characterized in terms of attributes
Can be associated with predictive metrics related to the
tasks you want to perform

Defining the entity is quite dificult

Feature engineering -

A feature is an attribute or property that is shared by all the entities on which analysis or predictions are to be done

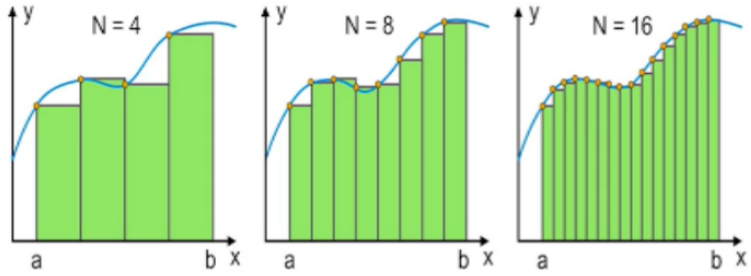
- In feature engineering, new features are created to extract more information from existing features.
- These new features may have an improved ability to explain the variance in the training data and improve model accuracy
- Feature engineering is highly influenced by business understanding

Main goals of feature engineering

- preparing the proper input dataset that's compatible with the machine learning algorithm requirements.
- try and improve the model performance

Binning

- Binning is one of the fundamental feature engineering techniques.
- The original data values which fall into a given small interval, a bin, are replaced by a value representative of that interval, often the central value.



Binning illustration of numerical data

To learn more about data binning see:

Binning is a technique used to reduce the effects of minor observation errors

A bin is the original data value which falls into a given small interval, the bin is usually replaced by a value representative of that interval, and that's often the central value.

Binning "smooths" the input data, it might lower the chance of overfitting in small data-sets →

there are two general methods of dividing data into bins.

equal frequency binning

- basically, the bins have equal frequency

equal width binning

- bins have equal widths with the cut-off points for each calculated based on the maximum and minimum values of the attribute and the number of bins being created

Binning helps to improve model performance, and it does this by capturing non-linear behaviour of continuous variables, it minimizes the impact of outliers, and removes noise from large numbers of distinct values.

Grouped values are easier to display and understand

Improves model build speed - predictive algorithms build faster as the number of distinct values decreases

Merge tables -

Merge tables

	A
1	A_NUMBER
2	7809702612
3	6139214653
4	7809538328
5	7783183499
6	7788829560
7	6132919446

Table 1: A_NUMBER_FACT

	A	B
1	CUSTOMER_ID	A_NUMBER
2	1000172	2942930441
3	1000198	2502048322
4	1000210	2502164353
5	1000213	2502280241
6	1000258	2503072523
7	1000260	2503383993

Table 2: CUSTOMER_ID_LOOKUP

	A	B	C	D	E	F	G	H
1	CUSTOMER_ID	GENDER	AGE	ZIP_CODE	DISTRIBUTION_CHANNEL_ID	DEVICE_BRAND_NAME	DEVICE_MODEL_NAME	TENURE_MTHS
2	1000172	M	49	85364	KCLL001	Samsung	Galaxy S7 Edge	12
3	1000198	F	36	30032	PH00001	Apple	iPhone 7	12
4	1000210	F	24	11208	PDS0001	Samsung	Galaxy S7	12
5	1000213	F	33	10025	PDS0001	Samsung	Galaxy S7 Edge	12
6	1000258	F	55	10009	PDS0001	Huawei	Honor 8	12
7	1000260	F	36	48043	PH00001	OnePlus	3T	12
8	1000261	M	48	2155	WMN00001	Huawei	Honor 8	12

Table 3: CUSTOMER

	A	B	C	D	E	F	G	H	I
1	A_NUMBER	CUSTOMER_ID	GENDER	AGE	ZIP_CODE	DISTRIBUTION_CHANNEL_ID	DEVICE_BRAND_NAME	DEVICE_MODEL_NAME	TENURE_MTHS
2	7785583624	1000111	M	40	91706	SMO0001	OnePlus	3T	2
3	6133996086	1000112	M	62	49589	AUC0001	Google	Pixel	5
4	6136001282	1000113	F	53	11213	PDS0001	Apple	iPhone 7	5
5	6041380925	1000114	M	25	91325	WMN00001	Google	Pixel XL	13
6	7803611208	1000115	F	46	70560	SPR00001	Google	Pixel	1
7	6046152252	1000116	F	55	90650	WLM0001	Apple	iPhone 7	9
8	7803611669	1000117	M	21	90805	PDS0001	Apple	iPhone 7	1
9	6048081598	1000118	F	29	60623	SMO0001	Apple	iPhone 7	7
10	6048082683	1000119	F	57	23602	PH00001	Apple	iPhone 7	7
11	6047290312	1000120	M	32	92647	PDS0001	Apple	iPhone 7	8
12	6132266053	1000121	M	22	44187	WHP00001	Apple	iPhone 7	6

Merged Table

To learn more about merging tables, see:

https://365datascience.com/what-are-joins/

Left-outer Join.

A_NUMBER - unique line number associated to each account entity

merged

CUSTOMER_ID_LOOKUP - associates with the unique line number.

merge key

Customer_ID comes from the Customer_ID in table 3

Data aggregation

- When you know that there can be multiple entries for one individual in a transaction table, for example, you have to compute an aggregate to avoid creating duplicates in your dataset
- For example, when you are counting the number of purchases in a store - a customer can make several purchases within a month - so there will be multiple entries for that customer

Create variables such as the **number of purchases per week or per month**, **total amount spent on purchases per week, per month or per quarter**, **date of last and first purchase.**

is the process where raw data is gathered and expressed

There are a range of aggregation functions you can consider, for example:

Functions	Description	Returned Values
Count	computes the number of occurrences	number of occurrences
Sum	compute the sum	sum
Average	compute the mean	mean
Min	identifies the minimum value	minimum value
Max	identifies the maximum value	maximum value
Exists	checks if at least one event exists for the current reference	0 if no event has been found 1 if at least one event has been found
NotExists	checks if no event exists for the current reference	0 if at least one event has been found 1 if no event has been found
First	identifies the first occurrence note that this function needs a date column	value of the first chronological occurrence for the current reference
Last	identifies the last occurrence note that this function needs a date column	value of the last chronological occurrence for the current reference

Data Encoding

- Data encoding is an essential part of the data preparation process.
- The encoding process prepares missing values in the data, deals with outliers, and creates data bins or bands to transform raw data into a "mineable" source of information

Nominal variable -

- A nominal variable is a discrete (categorical), qualitative variable that characterizes, describes, or names an element of a population.

Examples:

- Hair Color
- Make of car
- Gender
- Postal code
- Residence city

the order of the categories does not matter

Ordinal value

· An ordinal variable is a discrete (categorical), qualitative variable that has order

· Examples:

- Gold, Silver, bronze
- Satispaction level
- Pain level

The order of the categories DOES matter.

Continuous Variable

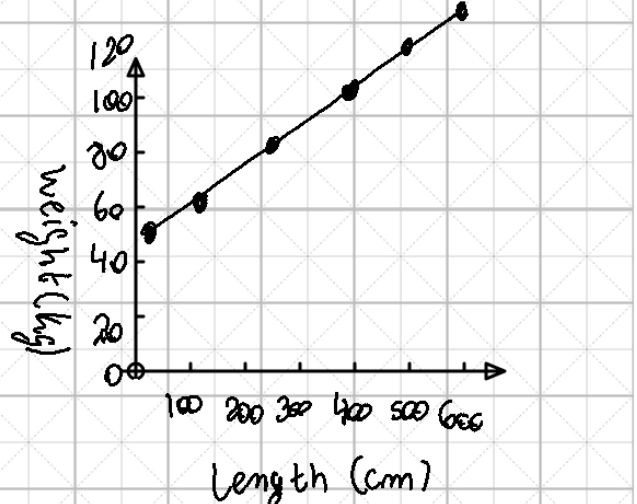
· A continuous variable is a quantitative variable.

· It is a real number that can take any value (with fractions/decimal places) between two specific numbers.

· It accommodates all basic math operations

Examples:

- Income ($)
- Age (years)
- Running time (minutes)
- Bank account balance ($)



Missing values

- A missing value is an empty cell in your dataset
- Missing values in a dataset can be due to error or simply not available
- They can be removed from the dataset, estimated or kept.
- The analysis could also be stopped so that further investigation of the missing values can be undertaken

Outliers

- For a continuous variable - An outlier is a single or low frequency occurence of the value of a variable.
- For a categorical variable (nominal or ordinary) - an outlier is a single or very low-frequency occurence of a category of a variable

Ex: 

Feature Selection

- IS the process of selecting a subset of relevant explanatory variables or predictors for use in data science model construction.
- it is also known as variable selection, attribute selection, or variable subset selection.
- Often, data contains many features that are either redundant or irrelevant, and can be removed without incurring much loss of information.
- Remember that domain knowledge can be the best selection criterion

Traditional approaches to data selection

- traditional approaches can be very time consuming, especially when there are 1000's of variables to analyse
- the most popular form of feature selection is stepwise regression this algorithim adds the best feature (or deletes the worst feature) in a series of iterative steps. The main control issue is deciding when to stop the algorithim
- Other automated selection processes are backward elimination and forward selection.

Backward Elimination

1. Backward elimination starts with all candidate features
2. Test the deletion of each feature using the chosen model comparison criterion, deleting the feature (if any) that improves the model the most by being deleted
3. Repeat this process until no further improvement is impossible

Forward selection

1. Forward selection starts with no features in the model
2. Test the addition of each feature using the chosen model comparison criterion
3. Add the feature (if any) that improves the model the most
4. Repeat this process until no other feature additions improve the model

Stepwise Regression

- This is a combination of backward elimination and forward selection
- At each stage in the process, after a new variable is added, a test is made to check if some variables can be deleted without appreciably increasing the error
- The procedure terminates when the measure is maximized, or when the available improvement falls below some critical value.
- One of the main issues with stepwise regression is that it is prone to overfitting the data.