Week 9: Dealing with Missing Data

Data Science Certificate Program

Ryerson University

Bora Caglayan

26 Mar, 2015

Outline

- A Review of Model Design (based on week 3)
- A Data Extraction Case Study
- Data Quality
- Data Preprocessing
- Complex Datasets

Review of Model Design

Definition of the Problem

Informal Definition:

A computer program is said to learn from experience ***E*** *with respect to some class of tasks* ***T*** *and performance measure* ***P****, if its performance at tasks in* ***T****, as measured by* ***P****, improves with experience* ***E****.*

Initially we need to define ***T, E*** *and* ***P.***

Definition of the Problem

Example

- Task (T): Classify a tweet that has not been published as going to get retweets or not.
- Experience (E): A corpus of tweets for an account where some have retweets and some do not.
- Performance (P): Classification accuracy, the number of tweets predicted correctly out of all tweets considered as a percentage.

Sample Problems

We start from informal definitions.

- Find faulty pieces of electronics.
- Approve/reject credit requests.
- Predict temperature in the weekend.

Sample Problems

Informal Definition

- Find faulty pieces of electronics. What are the task, experience and the performance criterion?

- Approve/reject credit requests.

- Predict temperature in the weekend.

More formal

Task: Classify an electronic equipment as faulty or not.

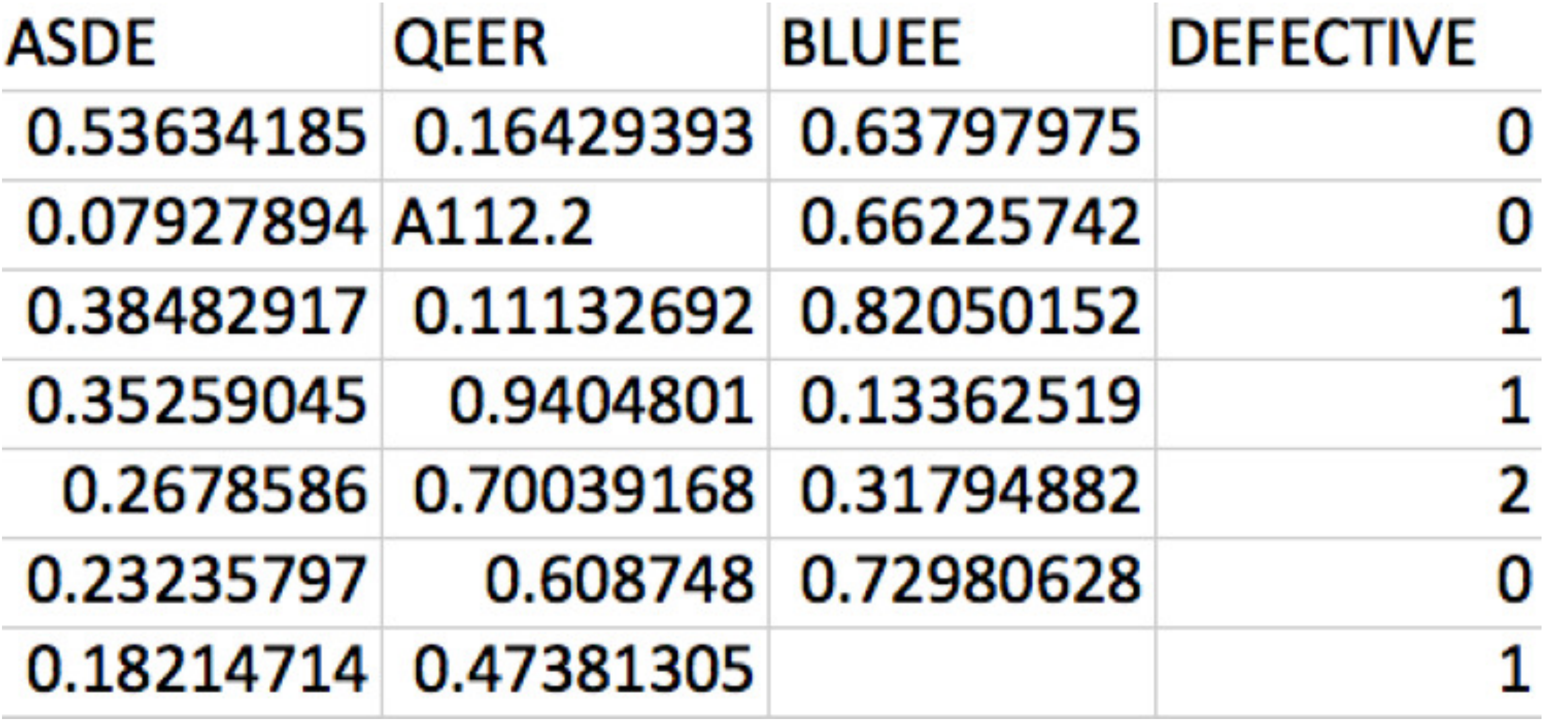
Experience: Numeric attributes based on diagnostic results and the classes of the electronics manufactured previously.

Performance Criteria: False negatives are much more costly than false positives so we introduce a custom performance criterion.

$$(1+\beta^2) * \frac{(Precision * Recall)}{(\beta^2 * Precision + Recall))}$$

Sample Problems

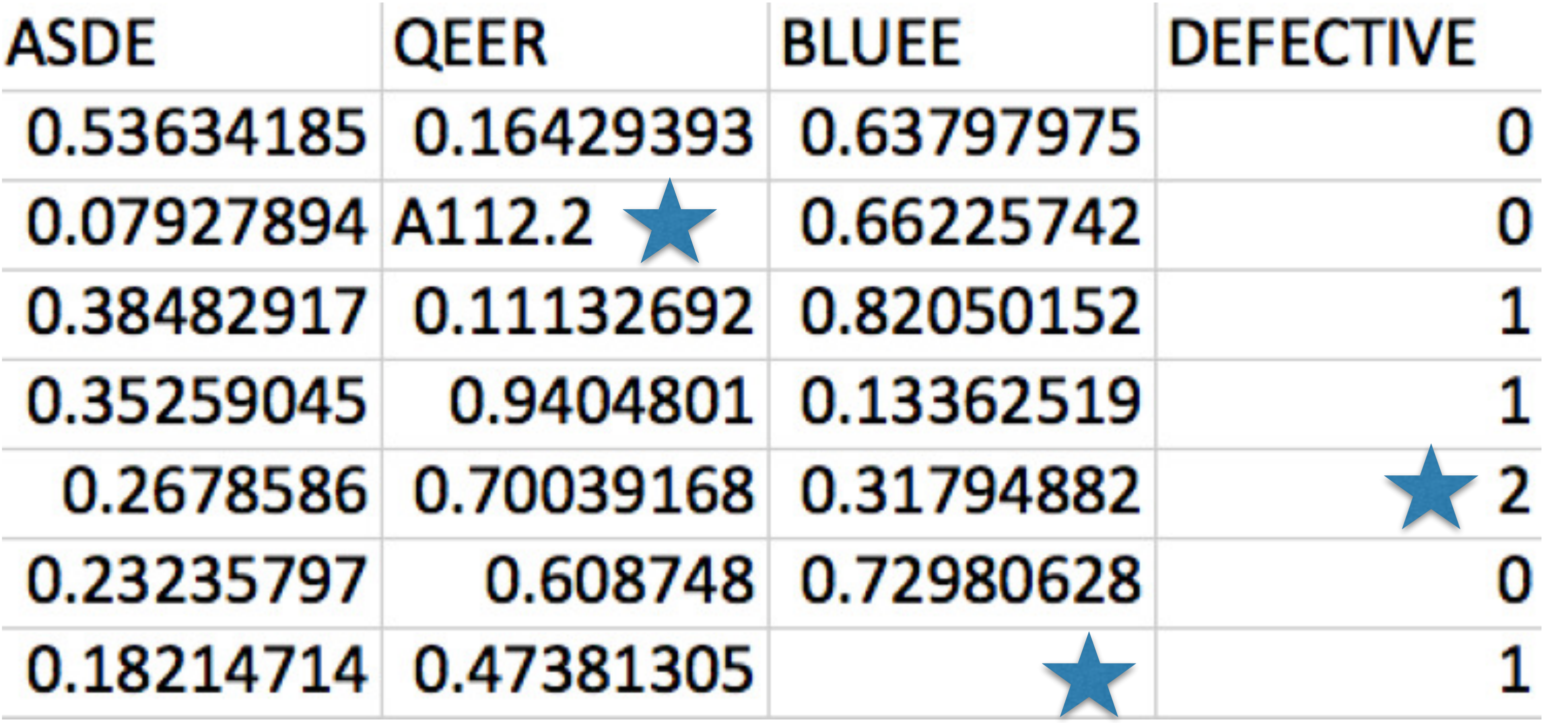
Here is the mockup data (experience) for chip classification



ASDE	QEER	BLUEE	DEFECTIVE
0.53634185	0.16429393	0.63797975	0
0.07927894	A112.2	0.66225742	0
0.38482917	0.11132692	0.82050152	1
0.35259045	0.9404801	0.13362519	1
0.2678586	0.70039168	0.31794882	2
0.23235797	0.608748	0.72980628	0
0.18214714	0.47381305		1

Sample Problems

Here is the mockup data (experience) for chip classification



ASDE	QEER	BLUEE	DEFECTIVE
0.53634185	0.16429393	0.63797975	0
0.07927894	A112.2 ★	0.66225742	0
0.38482917	0.11132692	0.82050152	1
0.35259045	0.9404801	0.13362519	1
0.2678586	0.70039168	0.31794882	★ 2
0.23235797	0.608748	0.72980628	0
0.18214714	0.47381305	★	1

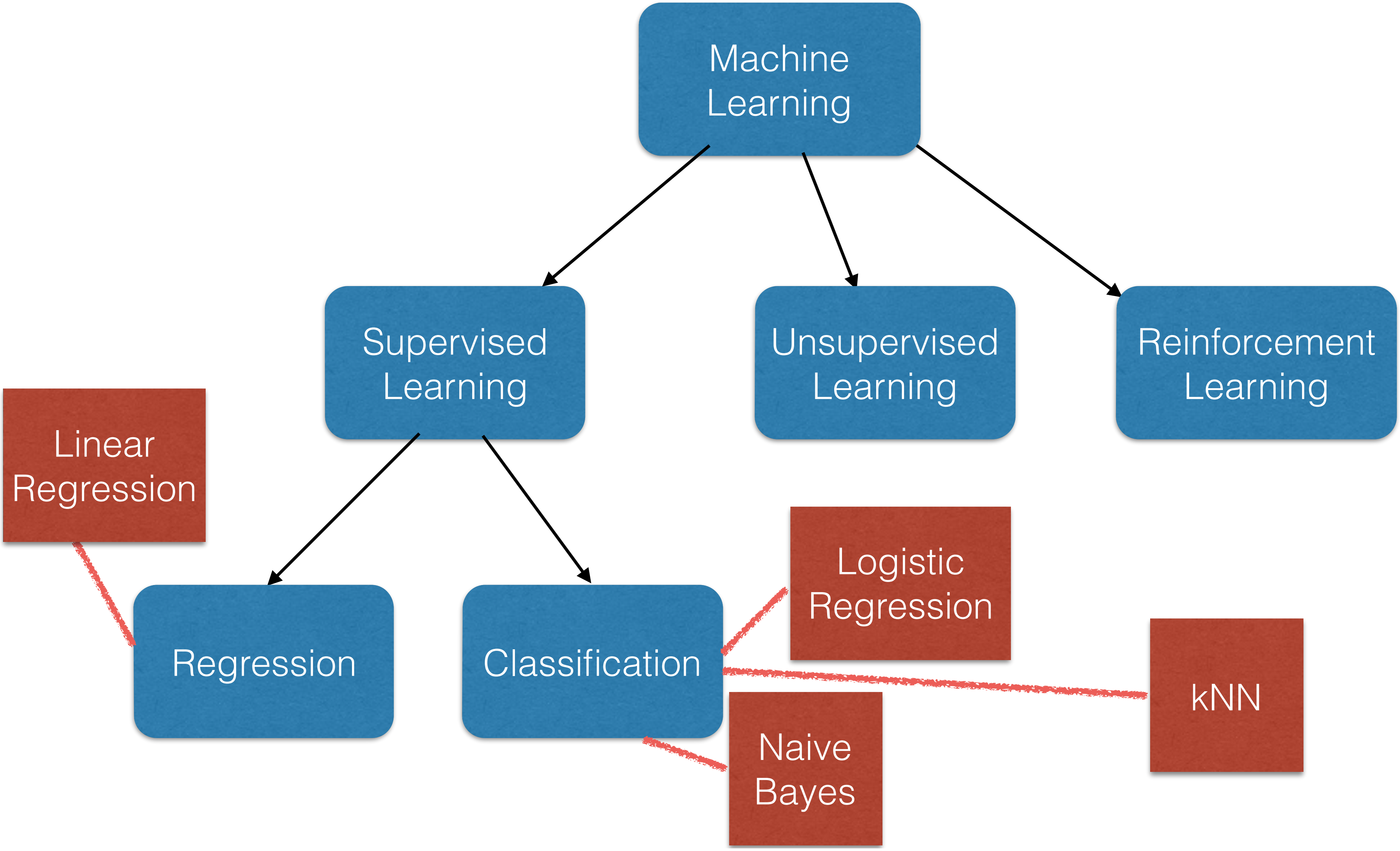
Sample Problems

Understand the data

- Get domain knowledge: What does bluee mean?
- What are the causes for the suspicious patterns?
- Patterns of attributes and classes?

Choose the machine learning model

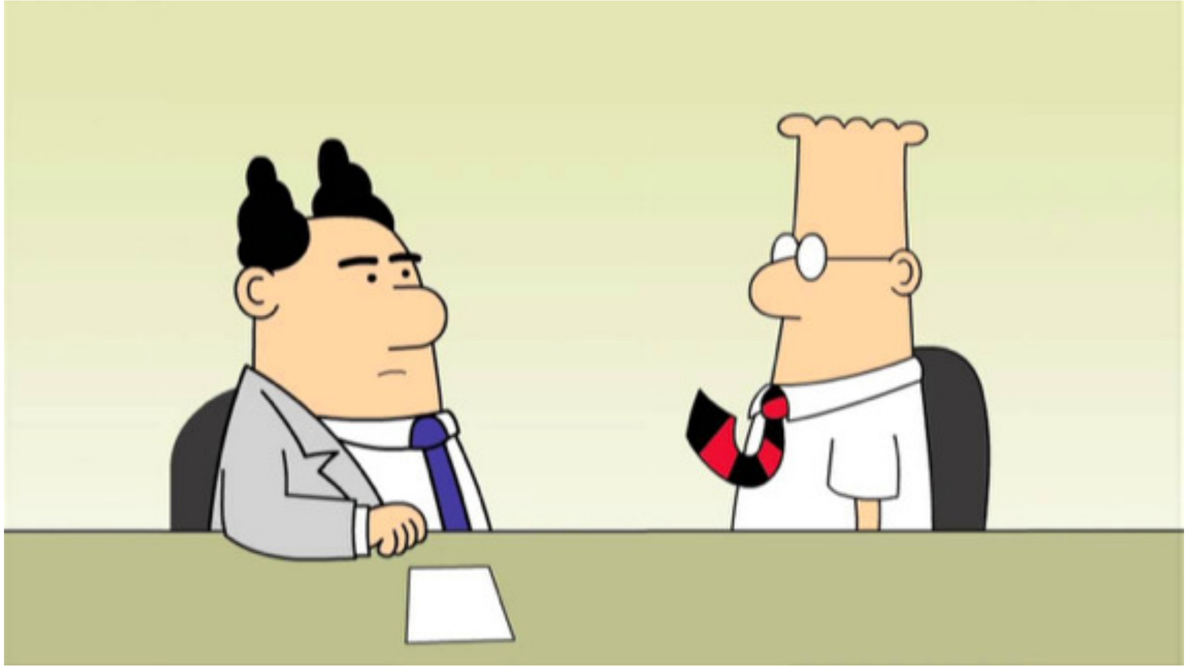
Machine Learning Problems



A Data Extraction Case Study

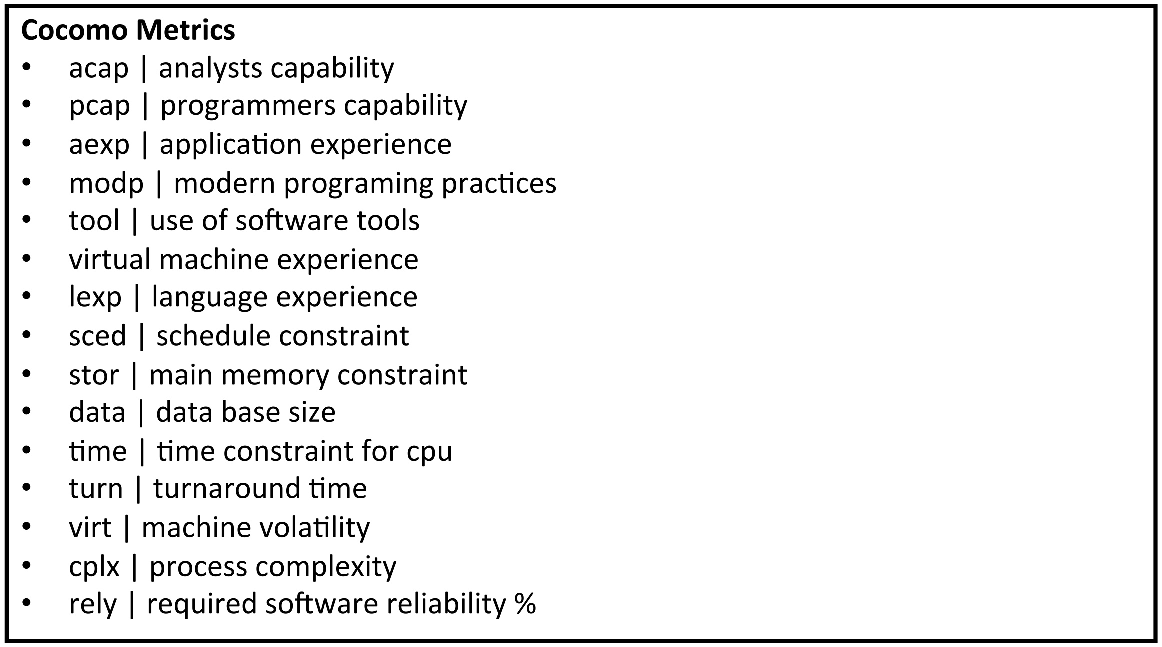
A Case Study

Informal Problem Definition: Build a model that predicts effort for software projects *(and it should be really good!)*.



A Case Study

Data Model



Cocomo Metrics

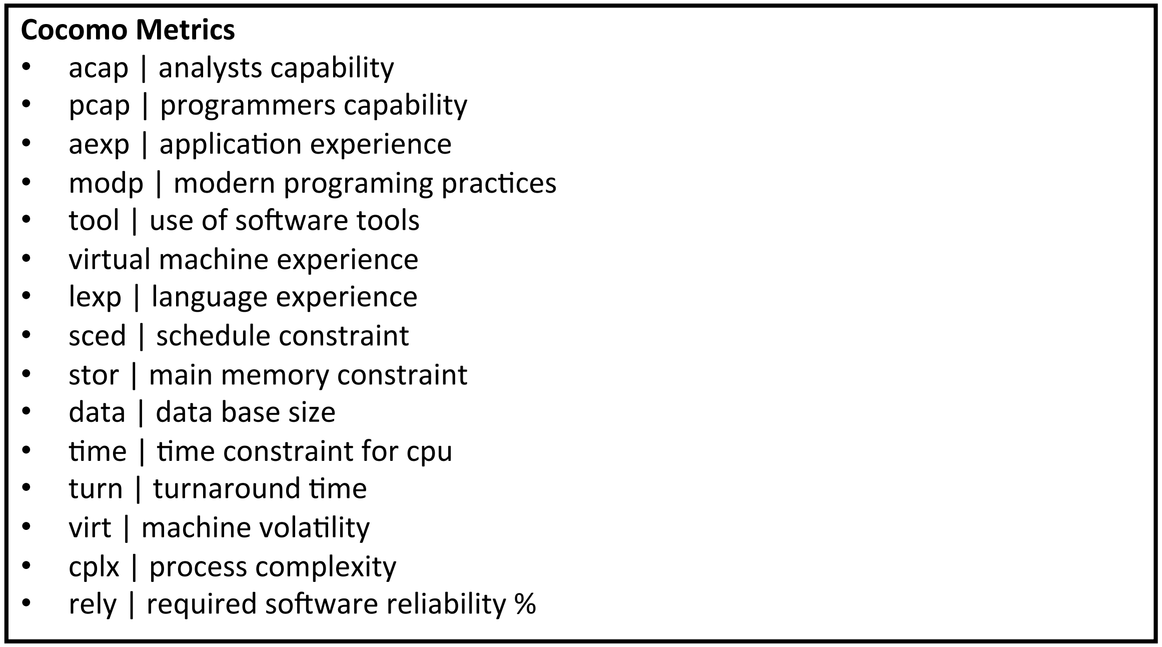
- acap | analysts capability
- pcap | programmers capability
- aexp | application experience
- modp | modern programing practices
- tool | use of software tools
- virtual machine experience
- lexp | language experience
- sced | schedule constraint
- stor | main memory constraint
- data | data base size
- time | time constraint for cpu
- turn | turnaround time
- virt | machine volatility
- cplx | process complexity
- rely | required software reliability %

Performance Measure

Pred(25): Percentage of instances within 25% error margin.

A Case Study

Data Model



Cocomo Metrics

- acap | analysts capability
- pcap | programmers capability
- aexp | application experience
- modp | modern programing practices
- tool | use of software tools
- virtual machine experience
- lexp | language experience
- sced | schedule constraint
- stor | main memory constraint
- data | data base size
- time | time constraint for cpu
- turn | turnaround time
- virt | machine volatility
- cplx | process complexity
- rely | required software reliability %

Possible Problems

Data is **not available** for your past projects.

Data is **incomplete** for some projects

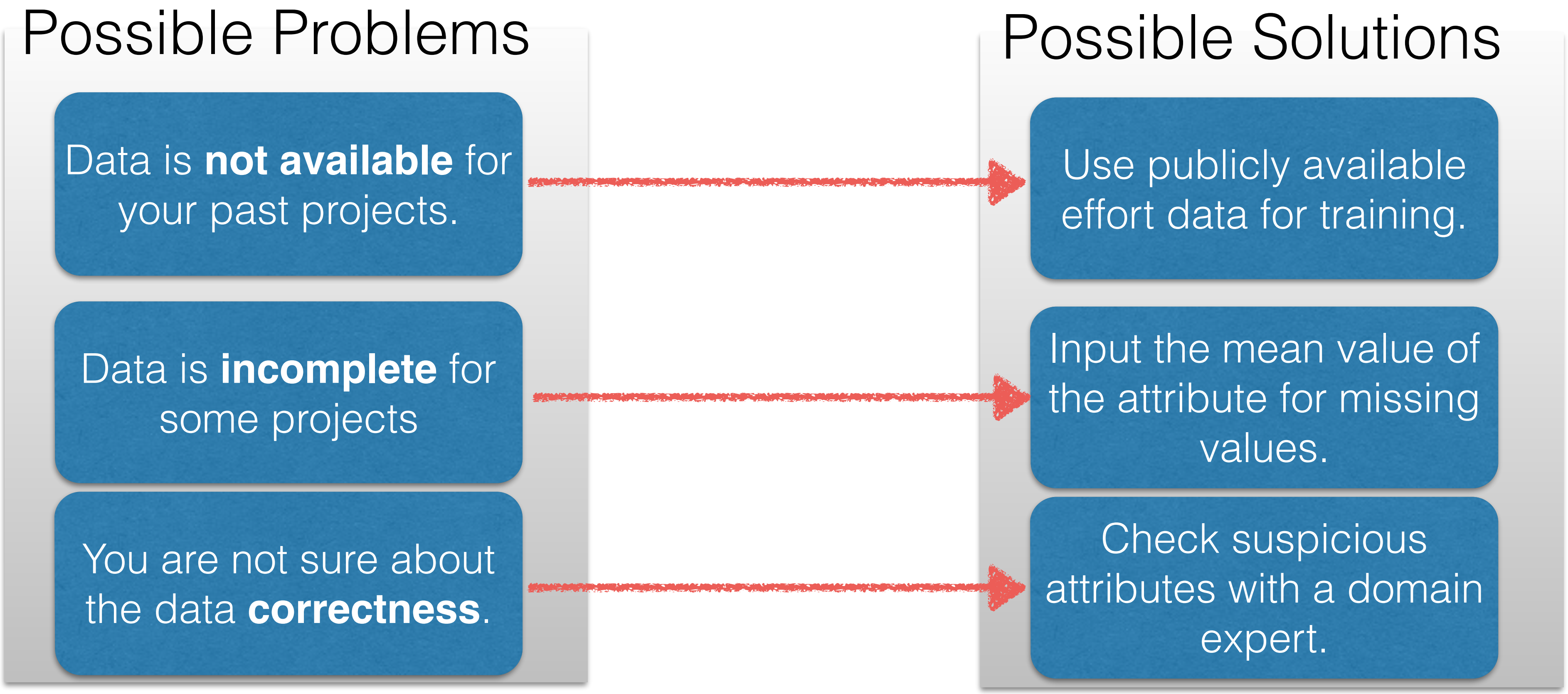
You are not sure about the data **correctness**.

Performance Measure

Pred(25): Percentage of instances within 25% error margin.

A Case Study

Data extraction might require a lot of pragmatic solutions.



Possible Problems

Data is **not available** for your past projects.

Data is **incomplete** for some projects

You are not sure about the data **correctness**.

Possible Solutions

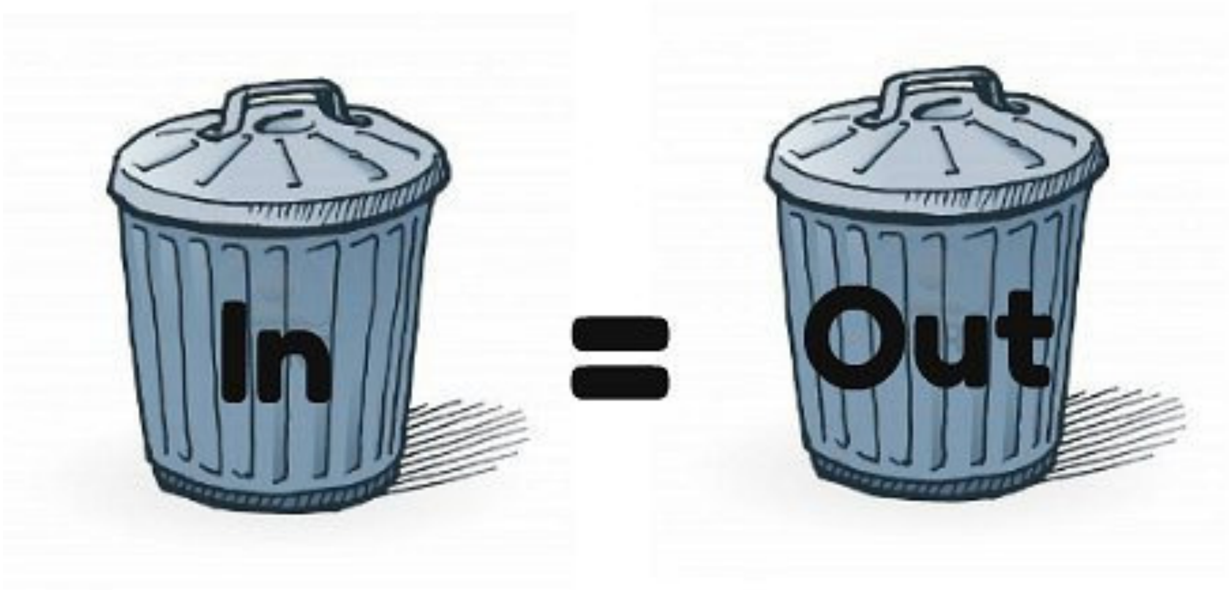
Use publicly available effort data for training.

Input the mean value of the attribute for missing values.

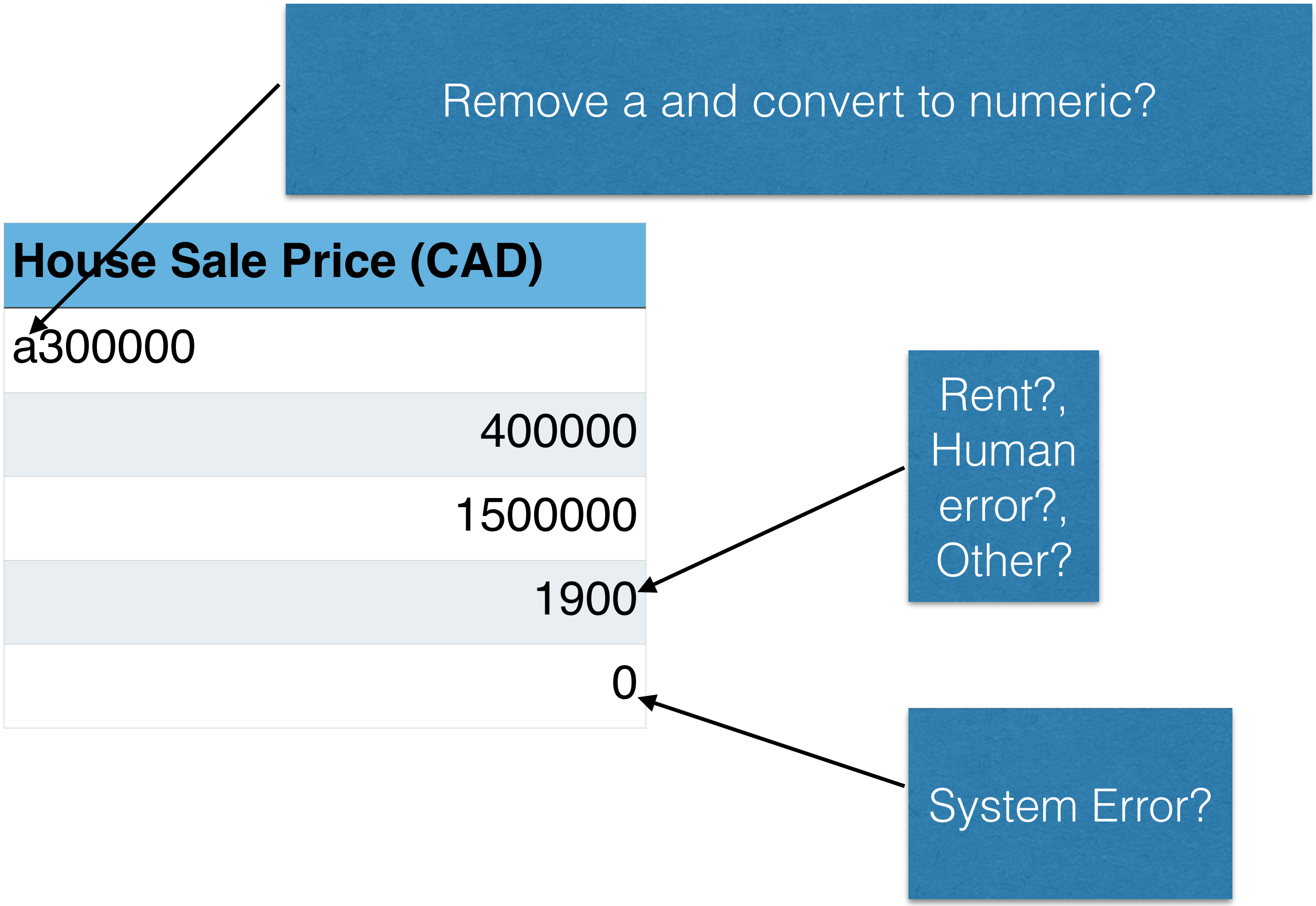
Check suspicious attributes with a domain expert.

Data Quality

Obvious Mistakes



Obvious Mistakes



House Sale Price (CAD)
a300000
400000
1500000
1900
0

Remove a and convert to numeric?

Rent?, Human error?, Other?

System Error?

Data Quality

- Generally, you have a problem if the data doesn't mean what you think it does, or should
  - Data not up to spec : garbage in, glitches, etc.
  - You don't understand the spec : complexity, lack of metadata.

- Data quality problems are expensive and pervasive
  - DQ problems cost hundreds of billion $$$ each year.
  - Resolving data quality problems is often the biggest effort in a data mining study.

Example

```
T.Das|97336o8327|24.95|Y|-|0.0|1000
Ted J.|973-360-8779|2000|N|M|NY|1000
```

- Can we interpret the data?
  - What do the fields mean?
  - What is the key? The measures?

- Data glitches
  - Typos, multiple formats, missing / default values

- Metadata and domain expertise
  - Field three is Revenue. In dollars or cents?
  - Field seven is Usage. Is it censored?
    - Field 4 is a censored flag. How to handle censored data?

Data Glitches

- Systemic changes to data which are external to the recorded process.
  - Changes in data layout / data types
    - Integer becomes string, fields swap positions, etc.
  - Changes in scale / format
    - Dollars vs. euros
  - Temporary reversion to defaults
    - Failure of a processing step
  - Missing and default values
    - Application programs do not handle NULL values well …
  - Gaps in time series
    - Especially when records represent incremental changes.

Conventional Definition of Data Quality

- Accuracy
  - The data was recorded correctly.
- Completeness
  - All relevant data was recorded.
- Uniqueness
  - Entities are recorded once.
- Timeliness
  - The data is kept up to date.
    - Special problems in federated data: time consistency.
- Consistency
  - The data agrees with itself.

Problems …

- Unmeasurable
  - Accuracy and completeness are extremely difficult, perhaps impossible to measure.
- Context independent
  - No accounting for what is important. E.g., if you are computing aggregates, you can tolerate a lot of inaccuracy.
- Incomplete
  - What about interpretability, accessibility, metadata, analysis, etc.
- Vague
  - The conventional definitions provide no guidance towards practical improvements of the data.

Missing Data

- Missing data - values, attributes, entire records, entire sections
- Missing values and defaults are indistinguishable
- Truncation/censoring - not aware, mechanisms not known
- Problem: Misleading results, bias.

Imputing Values to Missing Data

- In federated data, between 30%-70% of the data points will have at least one missing attribute - data wastage if we ignore all records with a missing value
- Remaining data is seriously biased
- Lack of confidence in results
- Understanding pattern of missing data unearths data integrity issues

Missing Value Imputation - 1

- Standalone imputation
  - Mean, median, other point estimates
  - Assume: Distribution of the missing values is the same as the non-missing values.
  - Does not take into account inter-relationships
  - Introduces bias
  - Convenient, easy to implement

Missing Value Imputation - 2

- Better imputation - use attribute relationships
- Assume : all prior attributes are populated
  - That is, monotonicity in missing values.

X1	X2	X3	X4	X5
1.0	20	3.5	4	.
1.1	18	4.0	2	.
1.9	22	2.2	.	.
0.9	15	.	.	.

Missing Value Imputation –3

- Regression method
  - Use linear regression, sweep left-to-right

    X3=a+b*X2+c*X1...

    X4=d+e*X3+f*X2+g*X1…
  - X3 in the second equation is estimated from the first equation if it is missing
- There are more advanced imputation methods for different scenarios…

Data Pre-processing

Definition of Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviours or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

Data goes through a series of steps during preprocessing:

1. Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.
2. Data Integration: Data with different representations are put together and conflicts within the data are resolved.
3. Data Transformation: Data is normalized, aggregated and generalized.
4. Data Reduction: This step aims to present a reduced representation of the data in a data warehouse.
5. Data Discretization: Involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

Sampling Techniques

Some Reasons for sampling:

- Design experiments
  - k-fold, splits
- Reduce data size for the experiments.
- Reduce class imbalance in the training dataset.
  - In this case equal number of random samples can be picked for each class to form the training data set.

Types of sampling

- Over sampling: number of instances is higher than the sampled population.
- Under sampling: number of instances is lower than the sampled population.

Feature Selection

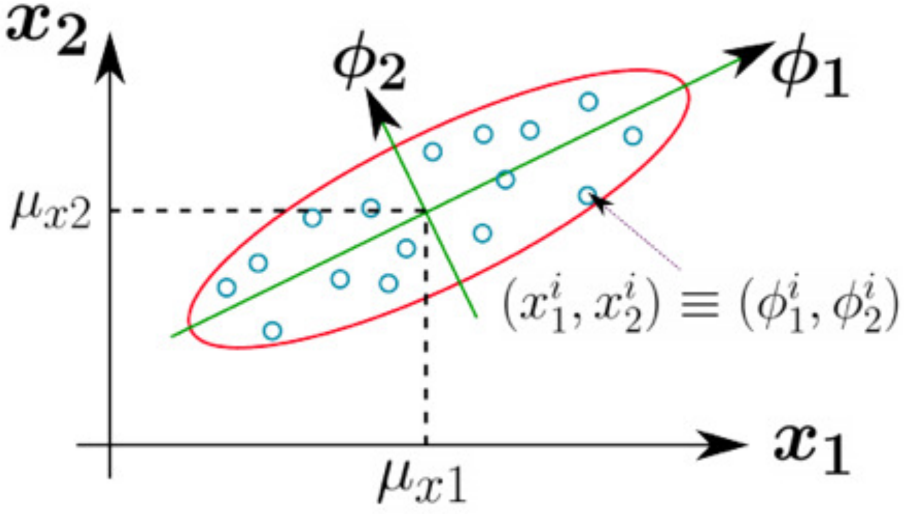
Reasons for Feature selection

- Reduce number of fitting.
- Reduce the chance for over-fitting.
- Reduce the computation cost.

Principal Component Analysis

Transformation of features into independent features.
Can be used for:

1. Orthogonalization
2. Feature reduction



Remember: The instances have the highest variance along the first PCA component.

Principal Component Analysis

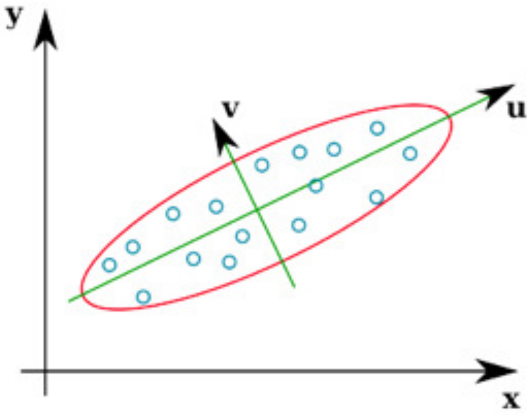


Figure 1: PCA for Data Representation

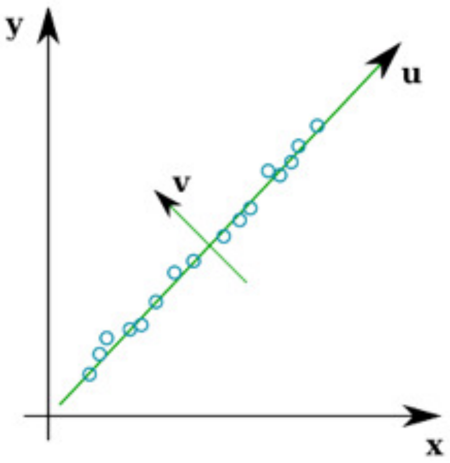


Figure 2: PCA for Dimension Reduction

Complex Datasets

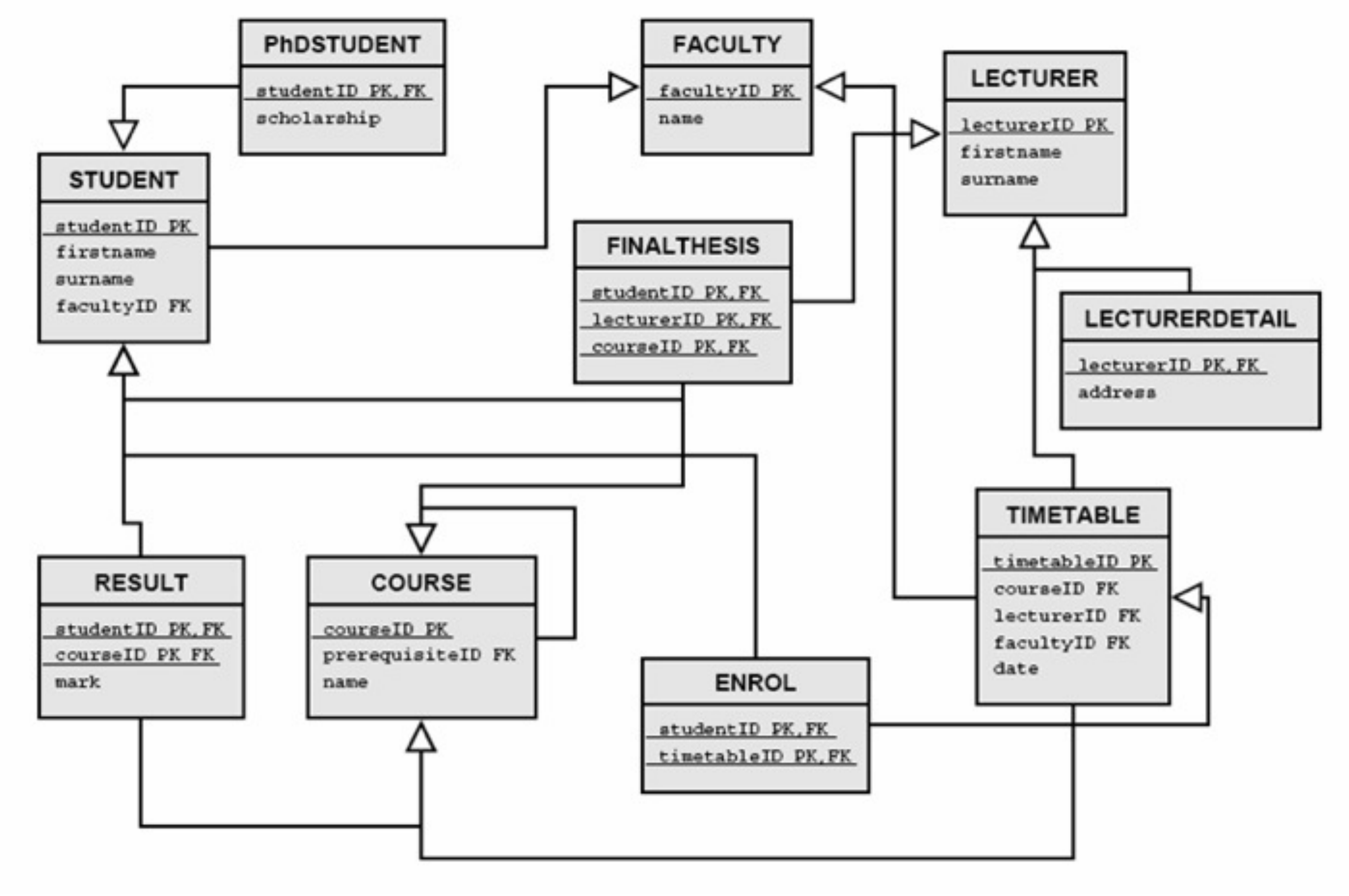
Complex Datasets

Complexity may emerge if one or more of the following is true:

- Many attributes (images, videos…)
- Many instances
- Dataset with many different dimensions
- Data requires a lot of domain expertise to process (medical data…)

Relational Data

Informal Problem Definition: Predict drop-out probability for students.



PhDSTUDENT
studentID PK, FK
scholarship

FACULTY
facultyID PK
name

LECTURER
lecturerID PK
firstname
surname

STUDENT
studentID PK
firstname
surname
facultyID FK

FINALTHESIS
studentID PK, FK
lecturerID PK, FK
courseID PK, FK

LECTURERDETAIL
lecturerID PK, FK
address

TIMETABLE
timetableID PK
courseID FK
lecturerID FK
facultyID FK
date

RESULT
studentID PK, FK
courseID PK FK
mark

COURSE
courseID PK
prerequisiteID FK
name

ENROL
studentID PK, FK
timetableID PK, FK

More on this on the next 2 weeks

Network Data

Informal Problem Definition: Rank the web pages based on their importance.

Week 9 Application Part: Data Pre-processing with R

Preparation

```r
library("RSQLite")
```

Connect to an in memory SQLite Database

In memory database is stored in memory. Deleted when connection is closed.

```
sqlite    <- dbDriver("SQLite")
m_con <- dbConnect(sqlite, dbname=":memory:")
```

Connect to a SQLite database file

Sqlite stores files into. Sqlite do not have advanced capabilities of other sql databases such as concurrency, user rights etc.. However, it is light weight, easy to share and scales pretty well.

```r
sqlite    <- dbDriver("SQLite")
m_con = dbConnect(sqlite, dbname="my_db.sqlite3")
```

Export data frame to table

```
dbSendQuery(m_con, "DROP TABLE IF EXISTS mtcars")
dbWriteTable(m_con, "mtcars", mtcars)
```

Query sqlite database and populate data frame

```
dbReadTable(m_con, "mtcars")
```

```
                     mpg cyl  disp  hp drat    wt  qsec vs
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1
Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1
Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1
Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0
```

Form advanced queries

```r
dbGetQuery(m_con,
"select * from mtcars where wt > 3")
```

```
             row_names  mpg cyl  disp  hp drat    wt  qsec
1       Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44
2    Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02
3              Valiant 18.1   6 225.0 105 2.76 3.460 20.22
4           Duster 360 14.3   8 360.0 245 3.21 3.570 15.84
5            Merc 240D 24.4   4 146.7  62 3.69 3.190 20.00
6             Merc 230 22.8   4 140.8  95 3.92 3.150 22.90
7             Merc 280 19.2   6 167.6 123 3.92 3.440 18.30
8            Merc 280C 17.8   6 167.6 123 3.92 3.440 18.90
9           Merc 450SE 16.4   8 275.8 180 3.07 4.070 17.40
10          Merc 450SL 17.3   8 275.8 180 3.07 3.730 17.60
11         Merc 450SLC 15.2   8 275.8 180 3.07 3.780 18.00
12  Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98
13 Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82
14   Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42
```

Sampling

An example of over-sampling:

```r
classes <- unique(iris$Species)
samples <- iris[0,]
for (c in classes){
    samples<- rbind(
        iris[sample(nrow(iris[iris$Species==c,]),
                    100, replace=T), ],
        samples
    )
}
nrow(samples)
```

```
[1] 300
```

PCA Example

```
cor(iris[,c(1,2,3,4)])
```

```
             Sepal.Length Sepal.Width Petal.Length Petal.Wi
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000
```

PCA Example

```
pca <- prcomp(iris[,c(1,2,3,4)])
cor(pca$x)
```

```
               PC1           PC2           PC3           PC4
PC1   1.000000e+00 -1.466480e-16 -1.103991e-15  2.006781e-15
PC2  -1.466480e-16  1.000000e+00  1.717045e-16 -5.192161e-16
PC3  -1.103991e-15  1.717045e-16  1.000000e+00 -1.478809e-15
PC4   2.006781e-15 -5.192161e-16 -1.478809e-15  1.000000e+00
```

PCA Example

```
summary(pca)
```

```
Importance of components:
                          PC1     PC2    PC3     PC4
Standard deviation     2.0563 0.49262 0.2797 0.15439
Proportion of Variance 0.9246 0.05307 0.0171 0.00521
Cumulative Proportion  0.9246 0.97769 0.9948 1.00000
```

Lab Preparation

```r
library("RWeka") # rweka (embedded Weka software)

diabetes <- read.arff(system.file("arff", "diabetes.arff",
                                  package = "RWeka"))
```

Lab Problems:

Lab questions will be distributed before the class.

1. Save diabetes dataset to an sqlite database
2. Load the diabetes dataset from the sqlite database.
3. (optional) Run a query on the newly formed database
4. Process numeric columns of the diabetes dataset with PCA. Check the correlations before and after the operation
5. Run logistic regression on the diabetes dataset before and after applying PCA. Check the difference.