

SESSION 9: MACHINE LEARNING FOR TEXT ANALYSIS

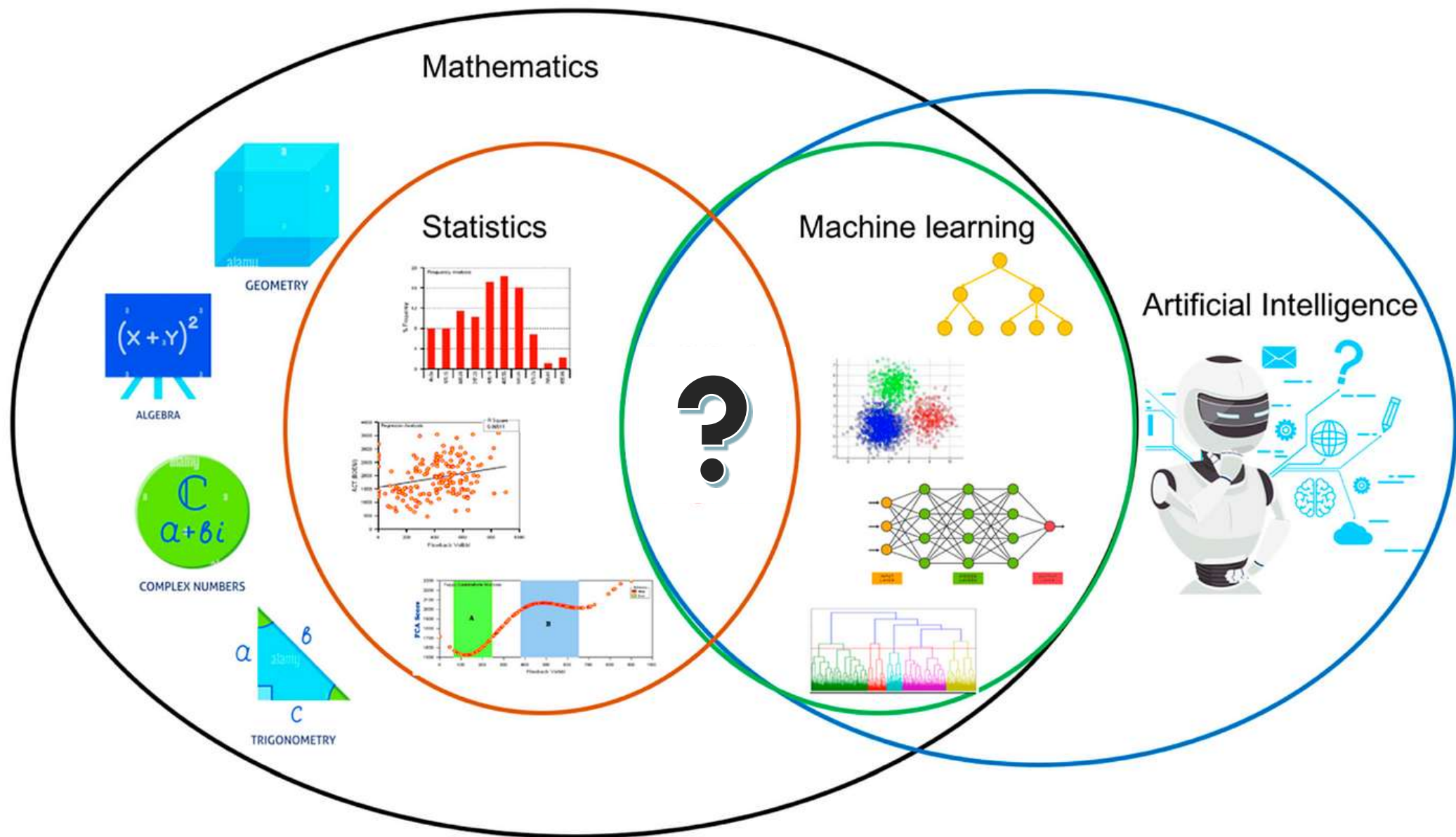
DR. SOFIA GIL-CLAVEL

- ❖ Recap on Text Analysis and Unsupervised Methods
- ❖ Unsupervised Methods for Text
- ❖ Recap on Statistics and Supervised Methods
- ❖ Supervised Methods for Text

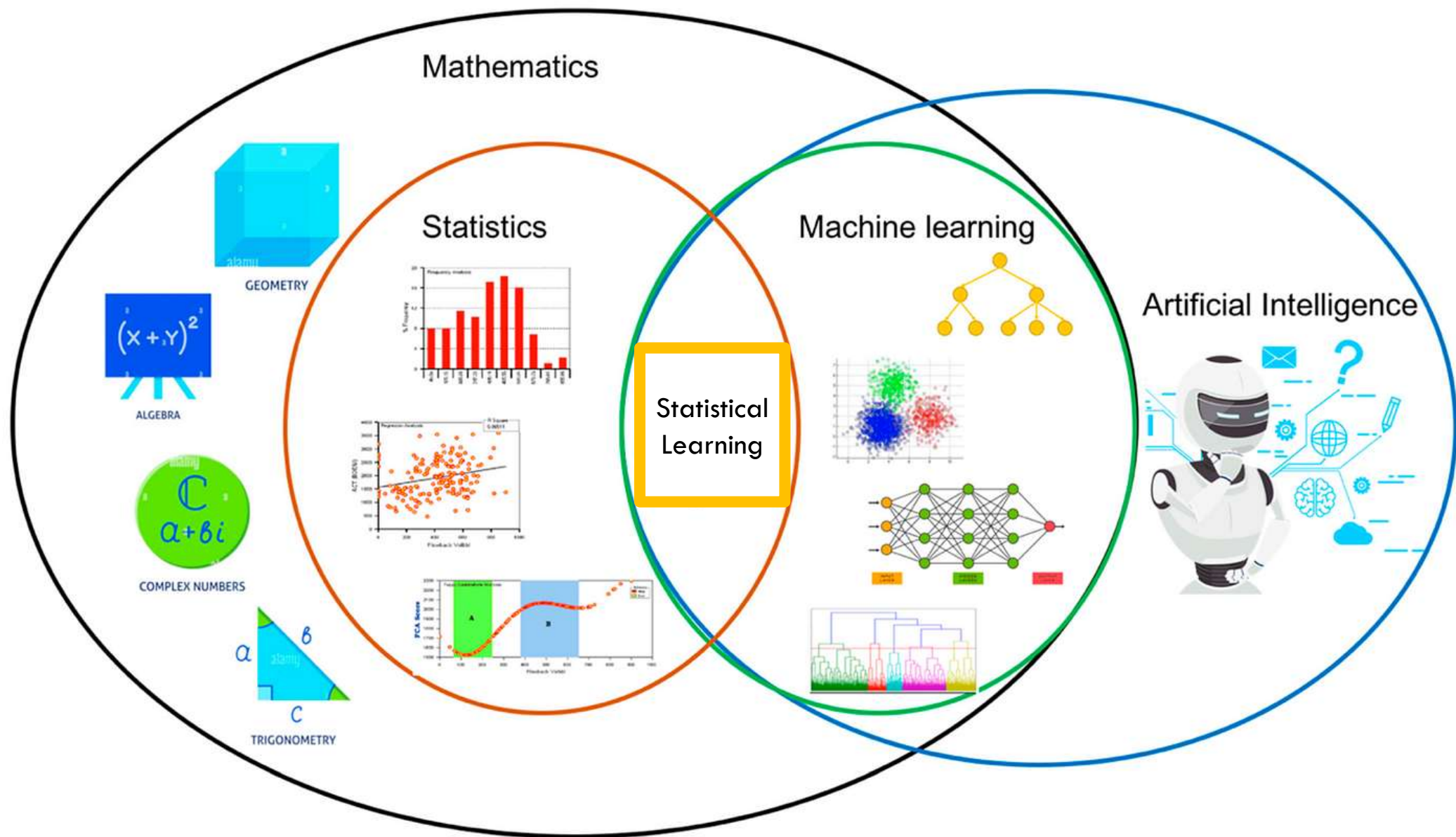
1. PRELIMINARIES FOR MACHINE LEARNING ON TEXT

1. Recap on Statistical Learning
2. Unsupervised vs. Supervised
3. From text to numbers

1.1 STATISTICAL LEARNING



Original source: <https://medium.com/stats-learning/differences-and-synergies-between-statistics-and-machine-learning-90cea85d4cf5>



Original source: <https://medium.com/stats-learning/differences-and-synergies-between-statistics-and-machine-learning-90cea85d4cf5>

WHY STATISTICAL LEARNING?

So far in these workshops we have learned to handle data frames in R.

data frame
name

columns

rows

variable

value/element

```
> Bikeshare
```

	bikers	season	day	holiday	weekday	workingday	temp	atemp	hum
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	16	1	1	0	6	0	0.24	0.288	0.81
2	40	1	1	0	6	0	0.22	0.273	0.8
3	32	1	1	0	6	0	0.22	0.273	0.8
4	13	1	1	0	6	0	0.24	0.288	0.75
5	1	1	1	0	6	0	0.24	0.288	0.75
6	1	1	1	0	6	0	0.24	0.258	0.75
7	2	1	1	0	6	0	0.22	0.273	0.8
8	3	1	1	0	6	0	0.2	0.258	0.86
9	8	1	1	0	6	0	0.24	0.288	0.75
10	14	1	1	0	6	0	0.32	0.348	0.76

i 8,655 more rows
i 3 more variables: windspeed <dbl>, casual <dbl>,
registered <dbl>
i Use `print(n = ...)` to see more rows

WHY STATISTICAL LEARNING?

So far in these workshops we have learned to handle data frames in R. This is very convenient, as we can analyze these data frames using statistical symbolic language!

➤ **Dependent (Y)**: Also known as response variable.

➤ **Independent (X)**: Also known as input, predictor, feature, or just variable.

```
> Bikeshare
# A tibble: 8,645 × 12
```

	bikers	season	day	holiday	weekday	workingday	temp	atemp	hum
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	16	1	1	0	6	0	0.24	0.288	0.81
2	40	1	1	0	6	0	0.22	0.273	0.8
3	32	1	1	0	6	0	0.22	0.273	0.8
4	13	1	1	0	6	0	0.24	0.288	0.75
5	1	1	1	0	6	0	0.24	0.288	0.75
6	1	1	1	0	6	0	0.24	0.258	0.75
7	2	1	1	0	6	0	0.22	0.273	0.8
8	3	1	1	0	6	0	0.2	0.258	0.86
9	8	1	1	0	6	0	0.24	0.288	0.75
10	14	1	1	0	6	0	0.32	0.348	0.76

```
# 8,645 more rows
# 3 more variables: windspeed <dbl>, casual <dbl>
#   registered <dbl>
# Use `print(n = ...)` to see more rows
```

dependent variable (Y)

independent variables (X)

WHY STATISTICAL LEARNING?


So far in these workshops we have learned to handle data frames in R. This is very convenient, as we can analyze these data frames using statistical symbolic language!

Symbolic
Language

➤ **Dependent** (Y): Also known as response variable.

➤ **Independent** (X): Also known as input, predictor, feature, or just variable.

$$Y \sim X1 + X2 + X3$$

 was built to
understand symbolic
language!

1.2 UNSUPERVISED VS. SUPERVISED

SUPERVISED STATISTICAL LEARNING

In supervised learning, for each observation of the predictor measurement(s) X_i there is an associated response measurement Y_i .

$$Y_i \sim X1_i + X2_i + X3_i$$

i refers to the data frame row

We wish to fit a model that relates the response to the predictors, with the aim of:

- Prediction: accurately predicting the response for future observations.
- Inference: better understanding the relationship between the response and the predictors.

Many classical statistical learning methods such as linear regression and logistic regression, as well as more modern approaches such as GAM, boosting, and support vector machines, operate in the supervised learning domain.

UNSUPERVISED STATISTICAL LEARNING

Unsupervised learning describes the somewhat more challenging situation in which for every observation $i = 1, \dots, n$, we observe a vector of measurements X_i but no associated response Y_i .

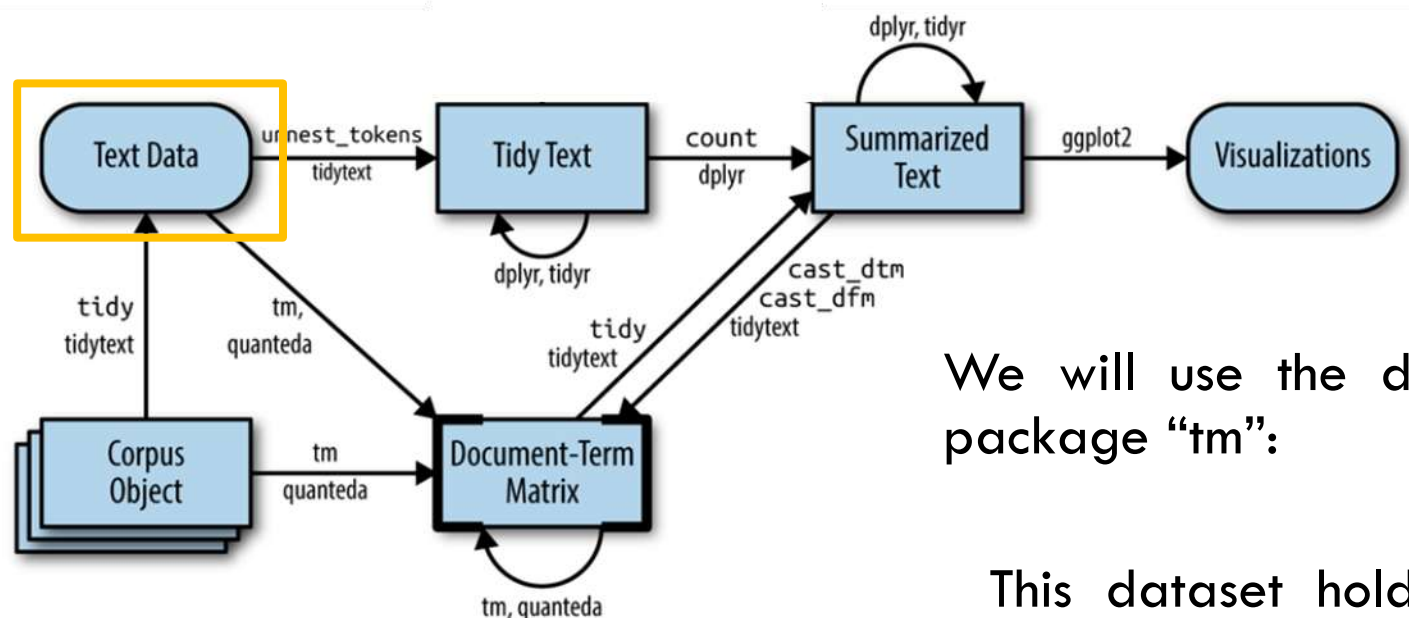
$$\text{No } Y_i \sim X1_i + X2_i + X3_i$$

We are not interested in prediction, because we do not have an associated response variable Y . The goal is to discover interesting things about the measurements on $X1, X2, \dots, Xp$. Is there an informative way to visualize the data? Can we discover subgroups among the variables or among the observations?

Unsupervised learning is often performed as part of an exploratory data analysis.

1.2 FROM TEXT TO NUMBERS

FROM TEXT TO NUMBERS

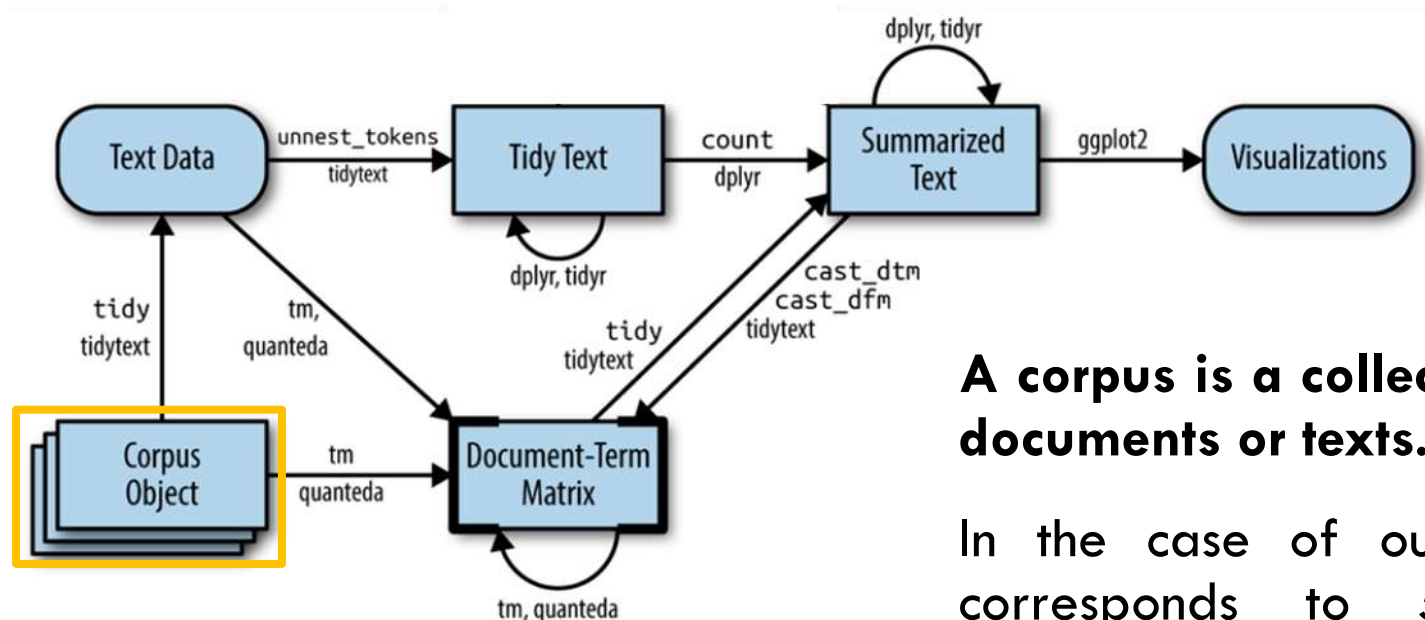


We will use the data “acq” from the package “tm”:

This dataset holds **50 news articles** with additional meta information from the Reuters-21578 data set. **All documents belong to the topic acq dealing with corporate acquisitions.**

```
library(tm)
data("acq")
```

FROM TEXT TO NUMBERS



A corpus is a collection of unstructured documents or texts.

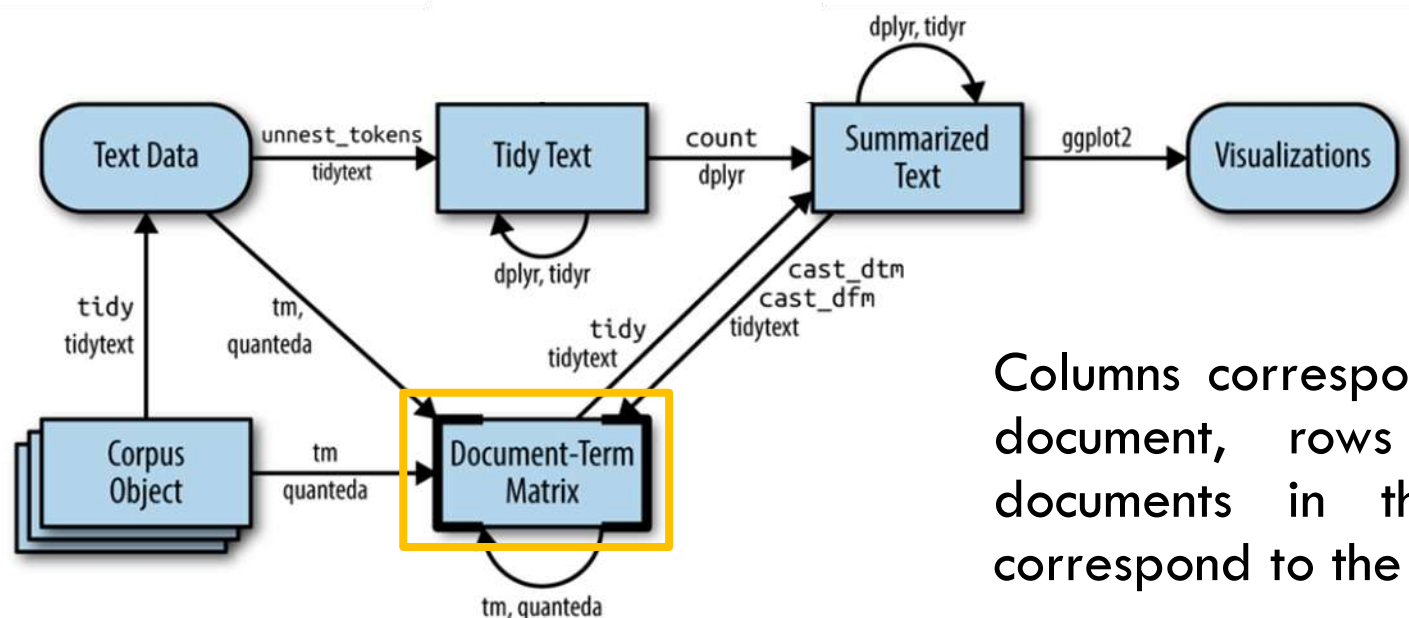
In the case of our data, the corpus corresponds to 50 articles about corporate acquisition.

Name	Type	Value
acq	list [50] (S3: VCorpus, Corpus)	List of length 50
10	list [2] (S3: PlainTextDocument, TextDocument)	List of length 2
content	character [1]	'Computer Terminal Systems Inc said\nit
meta	list [15] (S3: TextDocumentMeta)	List of length 15
12	list [2] (S3: PlainTextDocument, TextDocument)	List of length 2
content	character [1]	'Ohio Mattress Co said its first\nquarter,
meta	list [15] (S3: TextDocumentMeta)	List of length 15

Source: <https://www.tidytextmining.com/topicmodeling#topicmodeling>

Also: https://rpubs.com/Argaadya/topic_lda

FROM TEXT TO NUMBERS



Columns correspond to the terms in the document, rows correspond to the documents in the corpus and cells correspond to the weights of the terms.

TermDocumentMatrix {tm}

R Documentation

Term-Document Matrix

Description

Constructs or coerces to a term-document matrix or a document-term matrix.

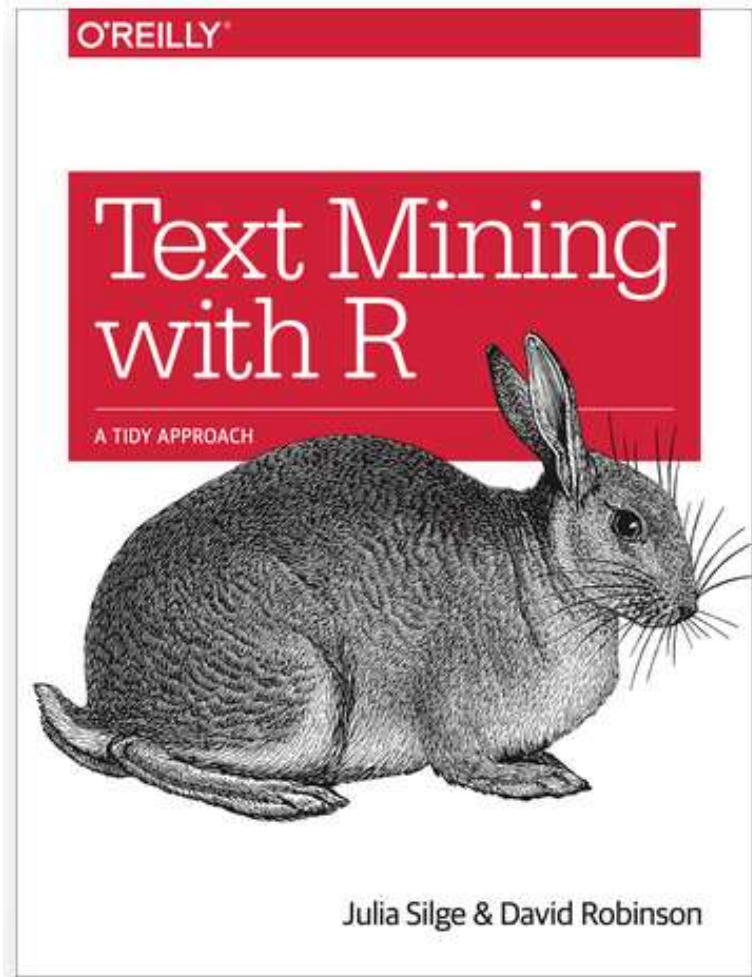
```
<<DocumentTermMatrix (documents: 50, terms: 1959)>>
Non-/sparse entries: 2615/95335
Sparsity           : 97%
Maximal term length: 21
Weighting          : term frequency (tf)
Sample            :
```

Docs	american	analysts	could	courier	express	hutton	rmj	shearson	value	viacom
110	12	5	5	0	10	1	0	6	4	0
302	0	2	0	0	0	0	0	0	0	0
331	0	0	0	0	0	0	8	0	0	0
362	12	8	2	0	10	1	0	12	2	0
372	0	3	0	11	0	10	0	0	0	0
393	0	0	0	0	0	0	0	0	6	8
448	0	0	0	0	0	0	0	0	0	0
45	0	0	5	0	0	0	0	0	0	0
496	0	0	0	0	0	0	0	0	1	7
504	0	0	0	0	0	0	0	0	0	0

2. UNSUPERVISED METHODS FOR TEXT

- 1. Topic Modelling
- 3. Latent Dirichlet allocation (LDA)

❖ TIDYTEXT: [HTTPS://WWW.TIDYTEXTMINING.COM/](https://www.tidytextmining.com/)



We developed the `tidytext` (Silge and Robinson 2016) R package because we were familiar with many methods for data wrangling and visualization, but couldn't easily apply these same methods to text. We found that using tidy data principles can make many text mining tasks easier, more effective, and consistent with tools already in wide use. Treating text as data frames of individual words allows us to manipulate, summarize, and visualize the characteristics of text easily and integrate natural language processing into effective workflows we were already using.

2.1 TOPIC MODELING

WHAT IS TOPIC MODELING?

Topic modeling is a collection of text-mining techniques that uses statistical and machine learning models to **automatically discover hidden abstract topics** in a collection of documents.

Topic modeling is also an amalgamation of a set of **unsupervised techniques** that's capable of detecting word and phrase patterns within documents and **automatically cluster word groups** and similar expressions helping in best representing a set of documents.

Source: <https://www.scaler.com/topics/nlp/topic-modelling-in-natural-language-processing/>

ARE PCA, TOPIC MODELING, AND CLUSTERING THE SAME?

All three algorithms, PCA, topic modeling, and clustering are unsupervised and used to **simplify the data set with a small number of summaries**, the major difference lies in how they are used:

- PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance. Since output representations are unrelated, they can also be used as **input to clustering**.
- Topic modeling and PCA both can be used for dimensionality reduction, but LDA provides **better accuracies and explainability in terms of text**.
- Clustering looks to find homogeneous subgroups among the observations by **maximizing the distance** between clusters, clusters are not known in advance.

Source: <https://www.scaler.com/topics/nlp/topic-modelling-in-natural-language-processing/>

METHODS FOR DIMENSIONALITY REDUCTION AND CLUSTERING

Here we can also perform Dimensionality Reduction! But in this case, there are other well-known methods:

- Non-Negative Matrix Factorization
- LDA – Latent Dirichlet Allocation
- LSA – Latent Semantic Allocation
- PLSA – Probabilistic Latent Semantic Analysis
- Ida2vec – Deep Learning Model
- tBERT – Topic BERT

Source: <https://www.scaler.com/topics/nlp/topic-modelling-in-natural-language-processing/>

METHODS FOR DIMENSIONALITY REDUCTION AND CLUSTERING

Here we can also perform Dimensionality Reduction! But in this case, there are other well-known methods:

- Non-Negative Matrix Factorization
- LDA – Latent Dirichlet Allocation
- LSA – Latent Semantic Allocation
- PLSA – Probabilistic Latent Semantic Analysis
- Ida2vec – Deep Learning Model
- tBERT – Topic BERT



We will focus on this one!

Source: <https://www.scaler.com/topics/nlp/topic-modelling-in-natural-language-processing/>

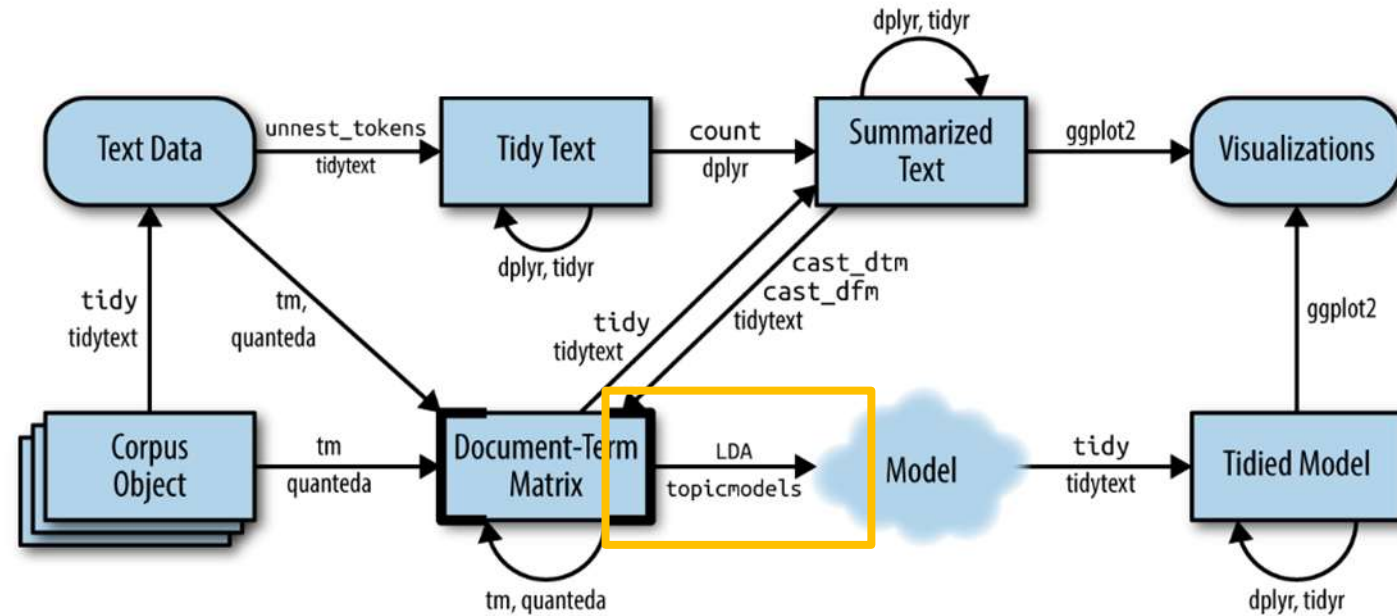
2.2 LATENT DIRICHLET ALLOCATION

LATENT DIRICHLET ALLOCATION

Latent Dirichlet allocation is one of the most common algorithms for topic modeling. Without diving into the math behind the model, we can understand it as being guided by two principles.

- **Every document is a mixture of topics.** We imagine that each document may contain words from several topics in particular proportions. For example, in a two-topic model we could say “Document 1 is 90% topic A and 10% topic B, while Document 2 is 30% topic A and 70% topic B.”
- **Every topic is a mixture of words.** For example, we could imagine a two-topic model of American news, with one topic for “politics” and one for “entertainment.” The most common words in the politics topic might be “President”, “Congress”, and “government”, while the entertainment topic may be made up of words such as “movies”, “television”, and “actor”. Importantly, words can be shared between topics; a word like “budget” might appear in both equally.

LATENT DIRICHLET ALLOCATION



LDA is a mathematical method for estimating both at the same time: finding the mixture of words that is associated with each topic, while also determining the mixture of topics that describes each document.

FitLdaModel {textmineR} R Documentation

Fit a Latent Dirichlet Allocation topic model

Description

Fit a Latent Dirichlet Allocation topic model using collapsed Gibbs sampling.

MAKING SENSE OF THE RESULTS

Some important attributes acquired from the LDA Model:

- **phi** : Posterior per-topic-per-word probabilities
- **theta** : Posterior per-document-per-topic probabilities
- **alpha** : Prior per-document-per-topic probabilities
- **beta** : Prior per-document-per-topic probabilities
- **coherence** : The probabilistic coherence of each topic

MAKING SENSE OF THE RESULTS

Some important attributes acquired from the LDA Model:

- **phi** : Posterior per-topic-per-word probabilities
- **theta** : Posterior per-document-per-topic probabilities
- **alpha** : Prior per-document-per-topic probabilities
- **beta** : Prior per-document-per-topic probabilities
- **coherence** : The probabilistic coherence of each topic

Let's check the top
ten words per
cluster!

```
GetTopTerms(lda_news$phi, 10) %>%  
as.data.frame()
```

MAKING SENSE OF THE RESULTS

LDA doesn't specifically inform us about what each topic is about. By looking at the representative words of each topic, we as the human will give meaning to each topic.



3. PRELIMINARIES FOR SUPERVISED LEARNING

- 1. Recap on Statistics
- 2. Regression vs. Classification
- 3. Prediction vs. Classification

3.1 RECAP ON STATISTICS

SOME NOTATION

```
> Bikeshare
# A tibble: 8,645 x 12
```

	bikers	season	day	holiday	weekday	workingday	temp	atemp	hum
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	16	1	1	0	6	0	0.24	0.288	0.81
2	40	1	1	0	6	0	0.22	0.273	0.8
3	32	1	1	0	6	0	0.22	0.273	0.8
4	13	1	1	0	6	0	0.24	0.288	0.75
5	1	1	1	0	6	0	0.24	0.288	0.75
6	1	1	1	0	6	0	0.24	0.258	0.75
7	2	1	1	0	6	0	0.22	0.273	0.8
8	3	1	1	0	6	0	0.2	0.258	0.86
9	8	1	1	0	6	0	0.24	0.288	0.75
10	14	1	1	0	6	0	0.32	0.348	0.76

```
# 18,455 more rows
# 3 more variables: windspeed <dbl>, casual <dbl>
# registered <dbl>
# Use `print(n = ...)` to see more rows
```

$$Y_i \sim X1_i + X2_i + X3_i \sim f(X_i)$$

dependent
variable (Y)

independent
variables (X)

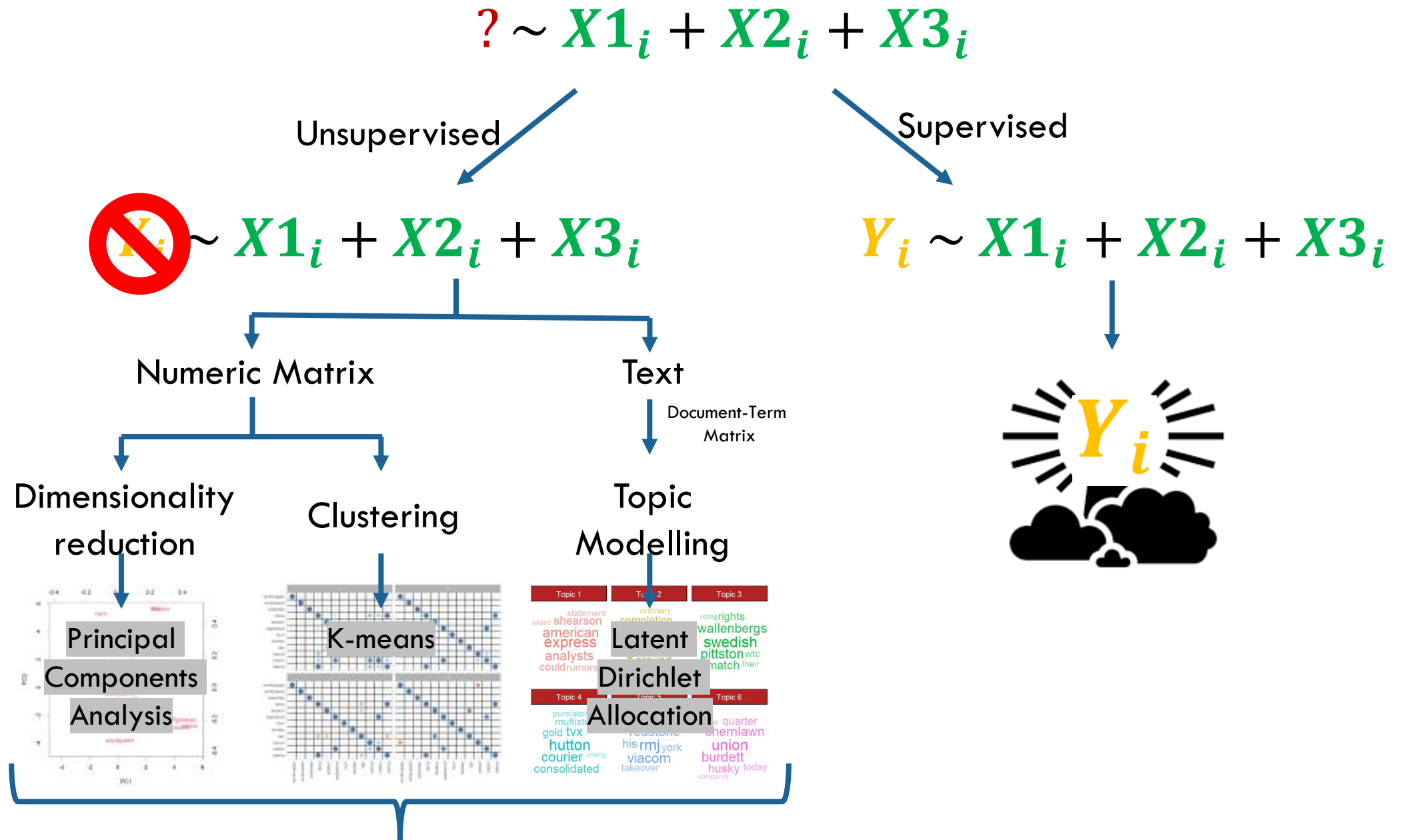
This value comes
from our data.

We estimate the model (f) parameters
 $\hat{\beta}_i$ and use them to predict \hat{Y}_i

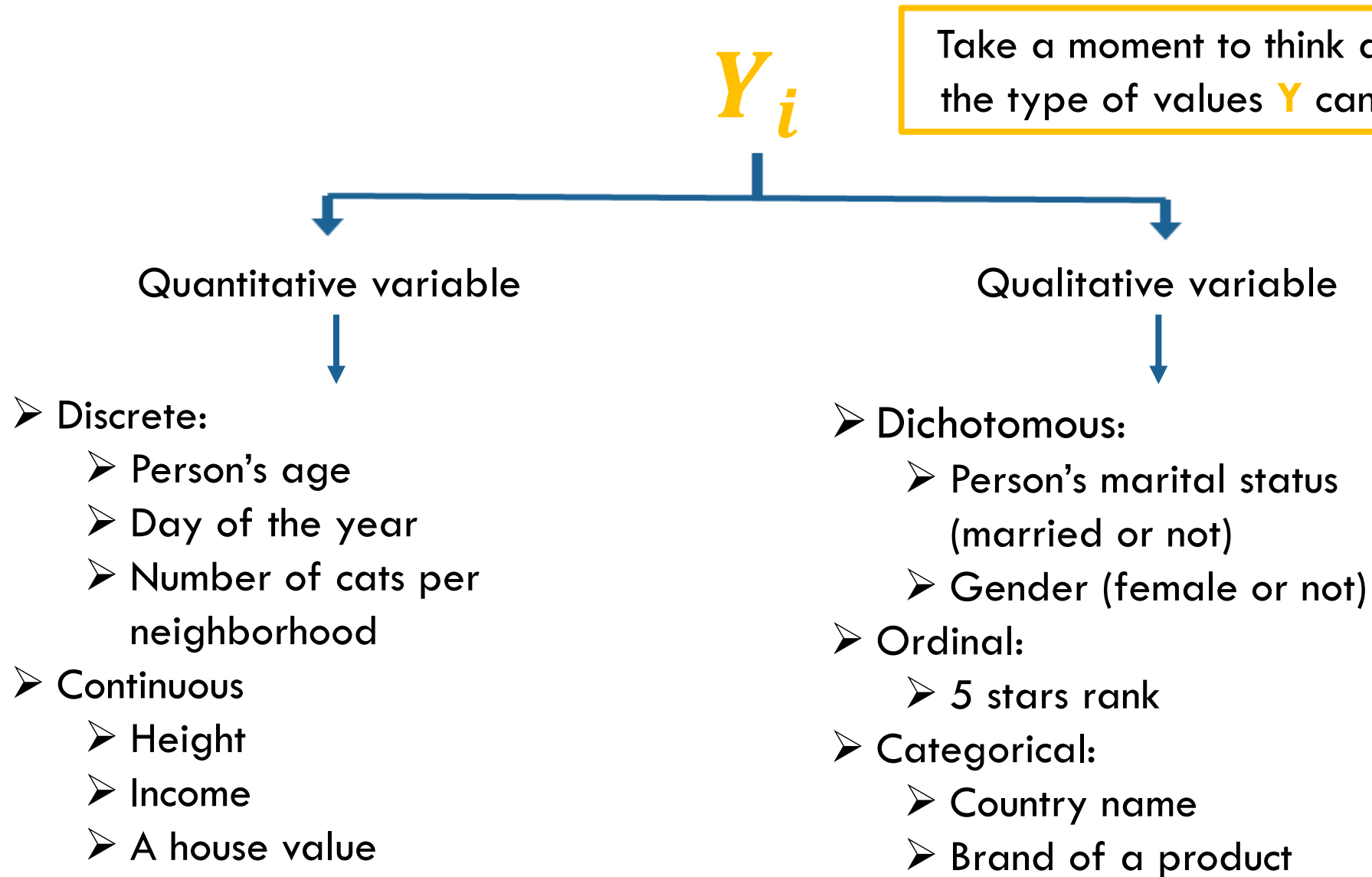
$$\hat{\beta}_1 X1_i + \hat{\beta}_2 X2_i + \hat{\beta}_3 X3_i = \hat{Y}_i$$

This value is the
prediction.

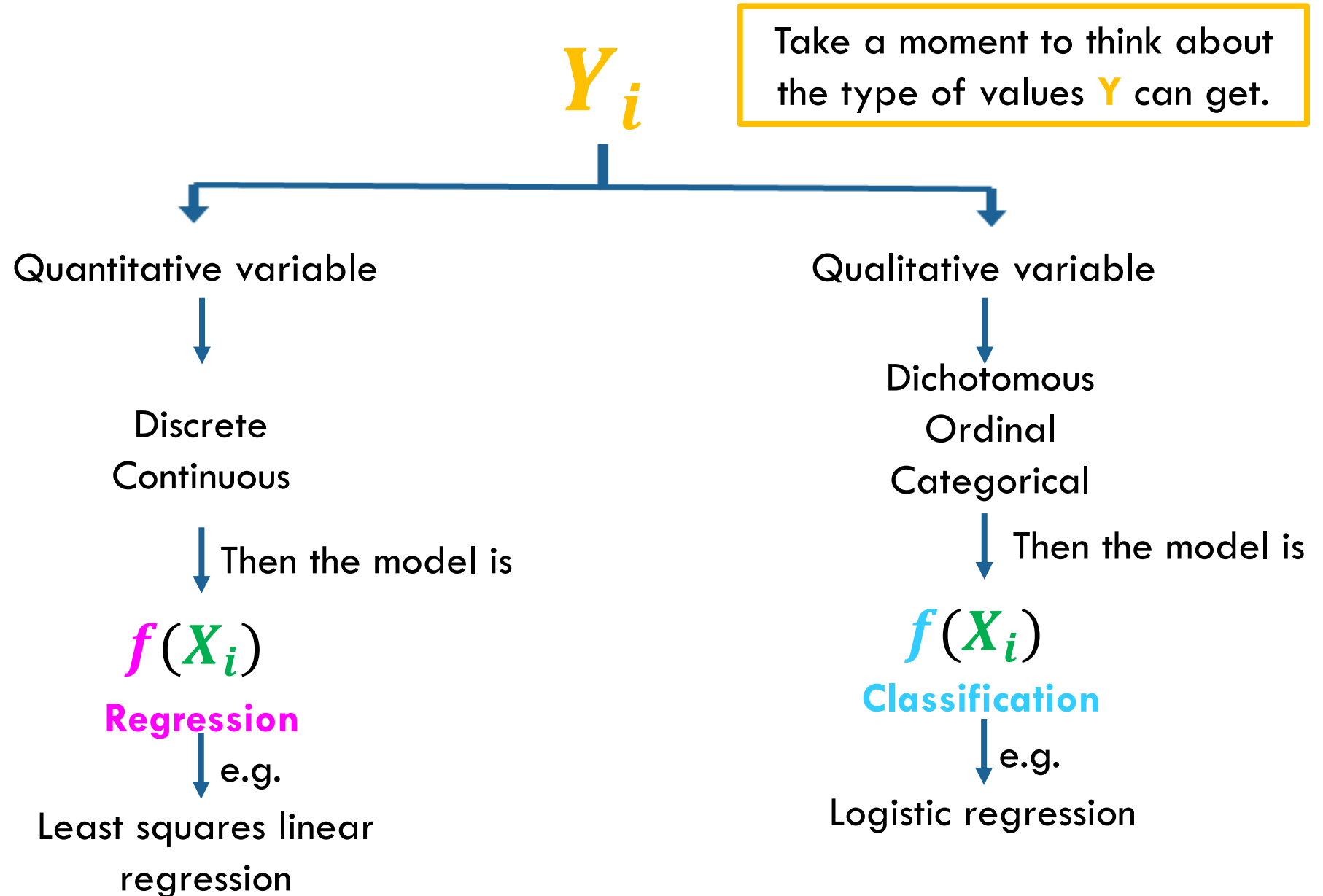
UNSUPERVISED VS. SUPERVISED



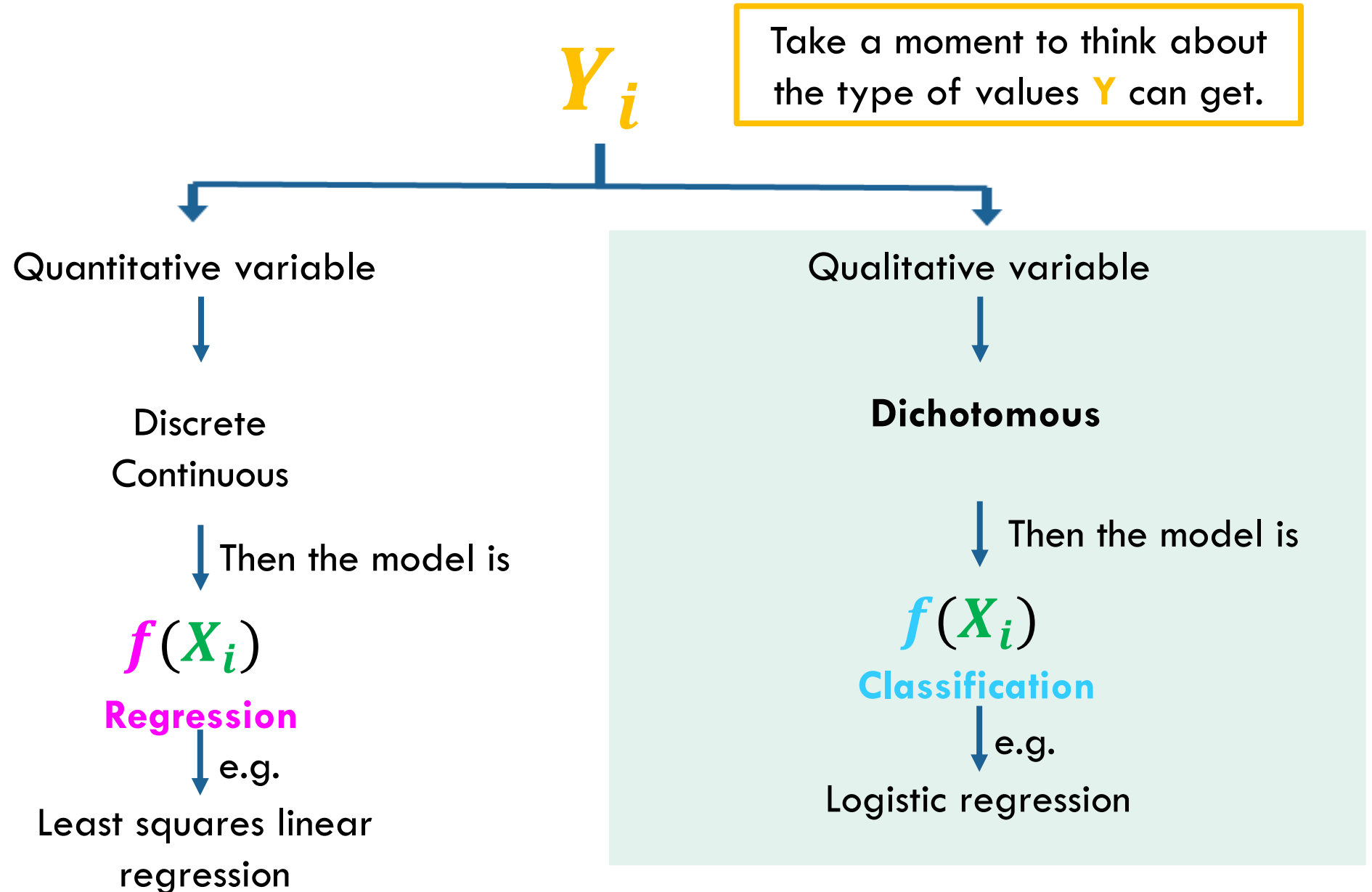
3.2 REGRESSION VS. CLASSIFICATION



REGRESSION VS. CLASSIFICATION



REGRESSION VS. CLASSIFICATION



3.3 PREDICTION VS. INFERENCE

PREDICTION VS. INFERENCE

$$Y_i \sim X1_i + X2_i + X3_i$$

↓ $f(X_i)$

$$\hat{Y}_i = \hat{\beta}_1 X1_i + \hat{\beta}_2 X2_i + \hat{\beta}_3 X3_i$$

Prediction

Predict likelihood someone gets
and allergic reaction to a
medicine.

The focus is on the predicted value:

$$\hat{Y}_i$$

We care about the **accuracy** of
the model.

Inference

Understand the **relationship between** income
and the variables years of education and
seniority.

The focus is on the coefficients:

$$\hat{\beta}_j$$

We care about the
interpretability of the model.

PREDICTION VS. INFERENCE

$$Y_i \sim X1_i + X2_i + X3_i$$

$$\downarrow f(X_i)$$

$$\hat{Y}_i = \hat{\beta}_1 X1_i + \hat{\beta}_2 X2_i + \hat{\beta}_3 X3_i$$

Prediction

Inference

Predict likelihood someone gets
and allergic reaction to a
medicine.

The focus is on the predicted value:

$$\hat{Y}_i$$

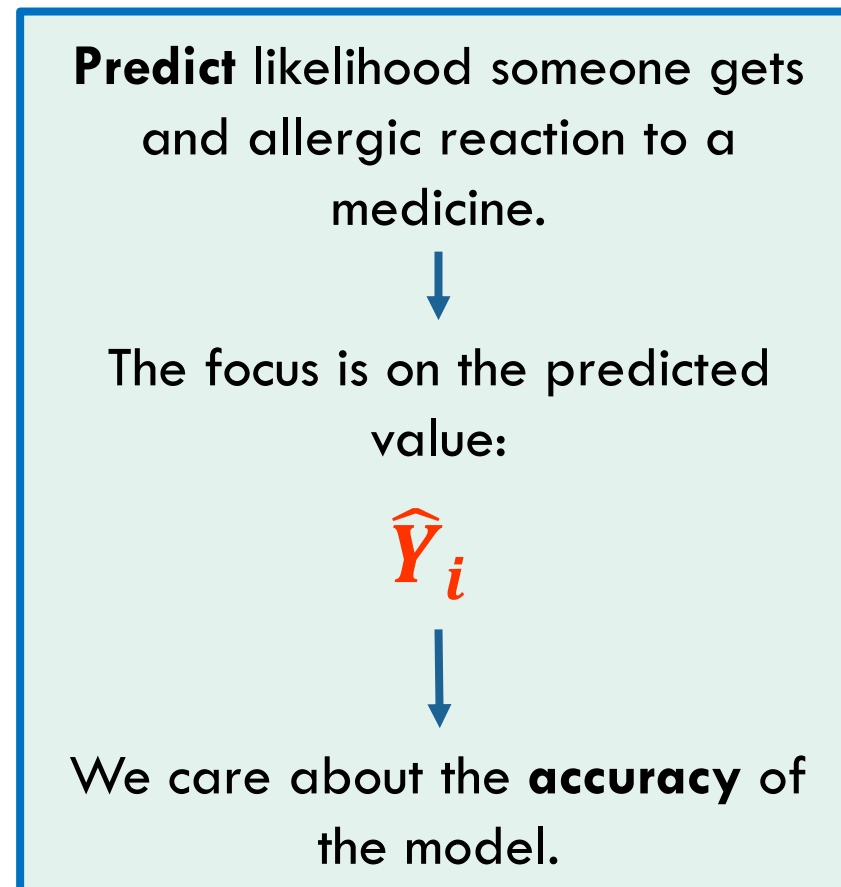
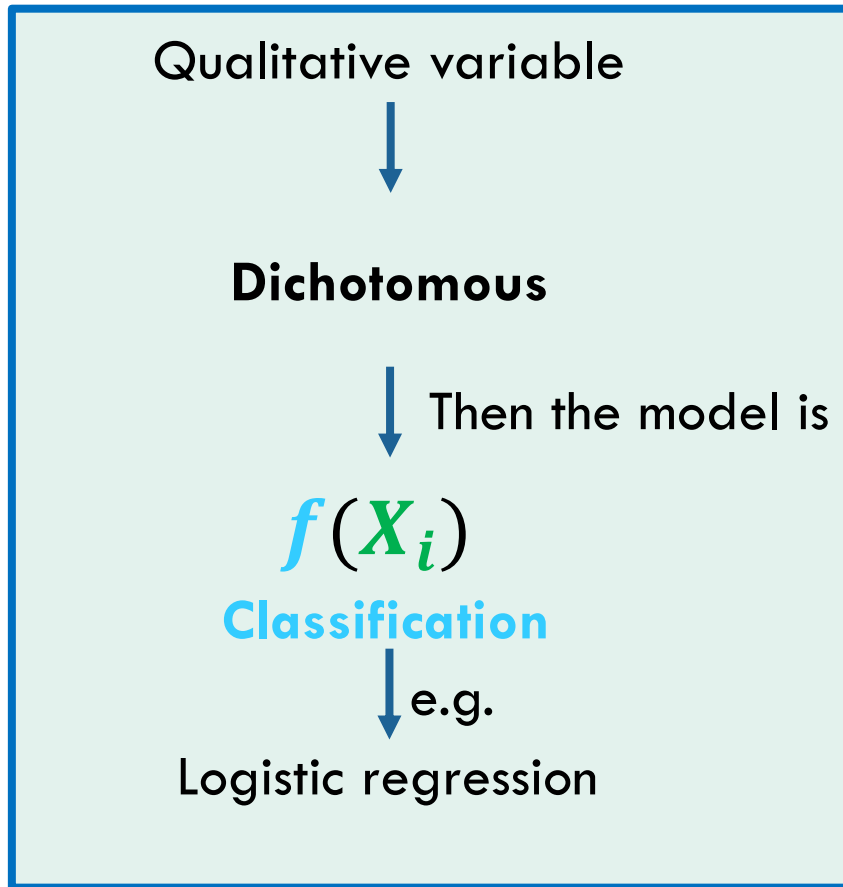
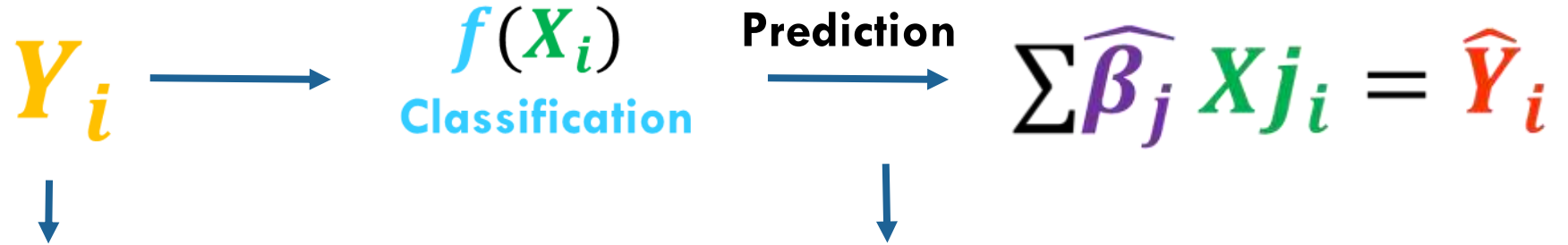
We care about the **accuracy** of
the model.

Understand the **relationship between** income
and the variables years of education and
seniority.

The focus is on the coefficients:

$$\hat{\beta}_j$$

We care about the
interpretability of the model.



4. TEXT CLASSIFICATION

THE DATA

```
> complaints
# A tibble: 117,214 × 18
  date_received product          sub_product issue sub_issue consumer_complaint_narrative
  <date>         <chr>          <chr>      <chr> <chr>      <chr>
1 2019-09-24      Debt collection I do not k... Atte... Debt is ... "transworld systems i...
2 2019-10-25      Credit reporting, ... Credit rep... Inco... Informat... "I would like to requ...
3 2019-11-08      Debt collection I do not k... Comm... Frequent... "Over the past 2 week...
4 2019-09-08      Money transfer, vi... Domestic (... Frau... NA          "I was sold access to...
5 2019-09-24      Debt collection Medical de... Atte... Debt is ... "While checking my cr...
6 2019-11-20      Credit reporting, ... Credit rep... Inco... Account ... "I would like the cre...
7 2019-09-19      Credit reporting, ... Credit rep... Inco... Personal... "MY NAME IS XXXX XXXX...
8 2019-11-22      Credit reporting, ... Credit rep... Inco... Personal... "Today XX/XX/XXXX wen...
9 2019-04-17      Credit reporting, ... Credit rep... Prob... Their in... "XXXX is reporting in...
10 2019-10-24      Credit reporting, ... Credit rep... Prob... Their in... "Please reverse the l...
# i 117,204 more rows
# i abbreviated name: 'consumer_complaint_narrative'
# i 12 more variables: company_public_response <chr>, company <chr>, state <chr>,
#   zip_code <chr>, tags <chr>, consumer_consent_provided <chr>, submitted_via <chr>,
#   date_sent_to_company <date>, company_response_to_consumer <chr>,
#   timely_response <chr>, consumer_disputed <chr>, complaint_id <dbl>
# i Use `print(n = ...)` to see more rows
```

Download from here:

<https://github.com/EmilHvitfeldt/smltar/blob/master/data/complaints.csv.gz>

THE PIPELINE

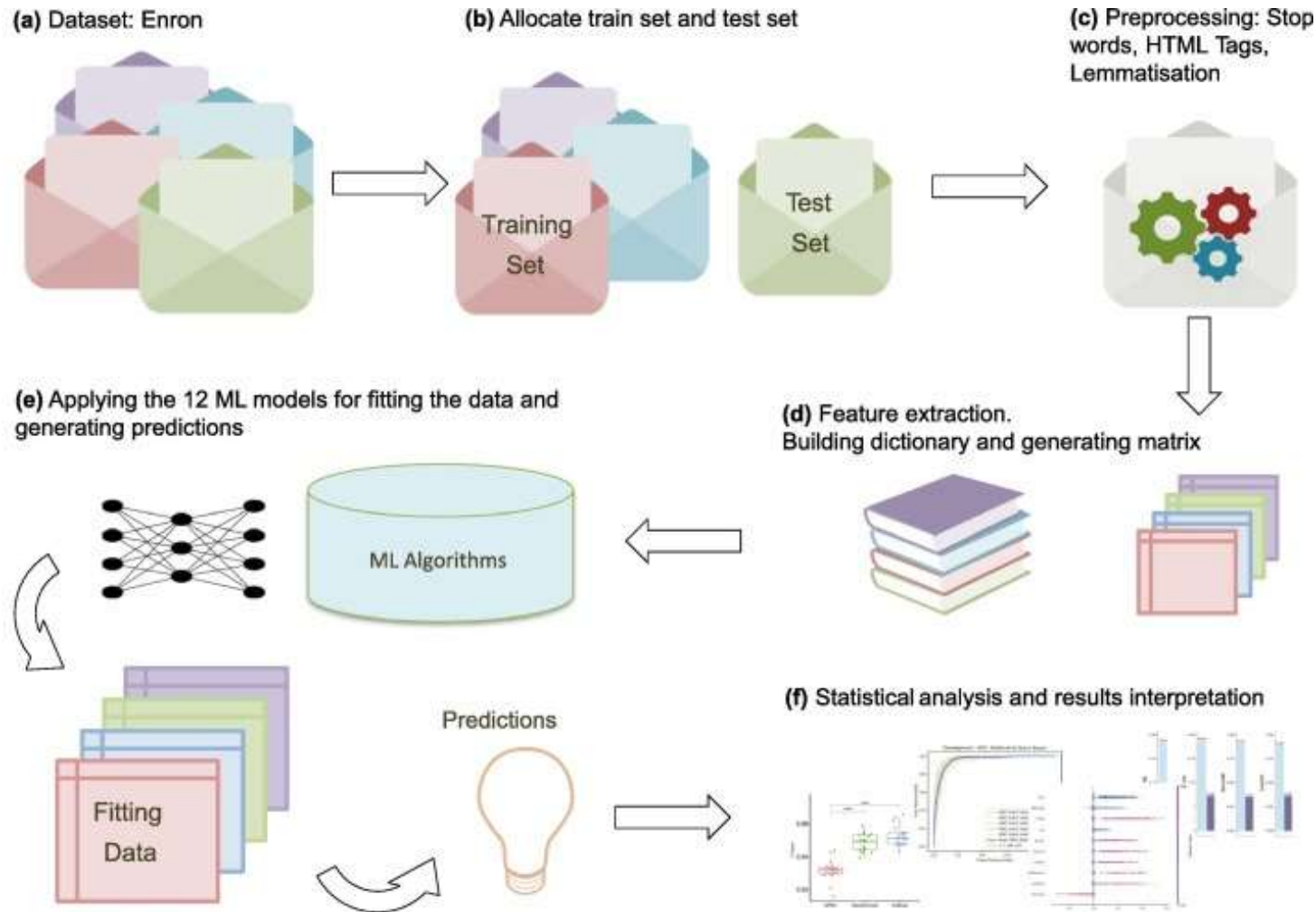
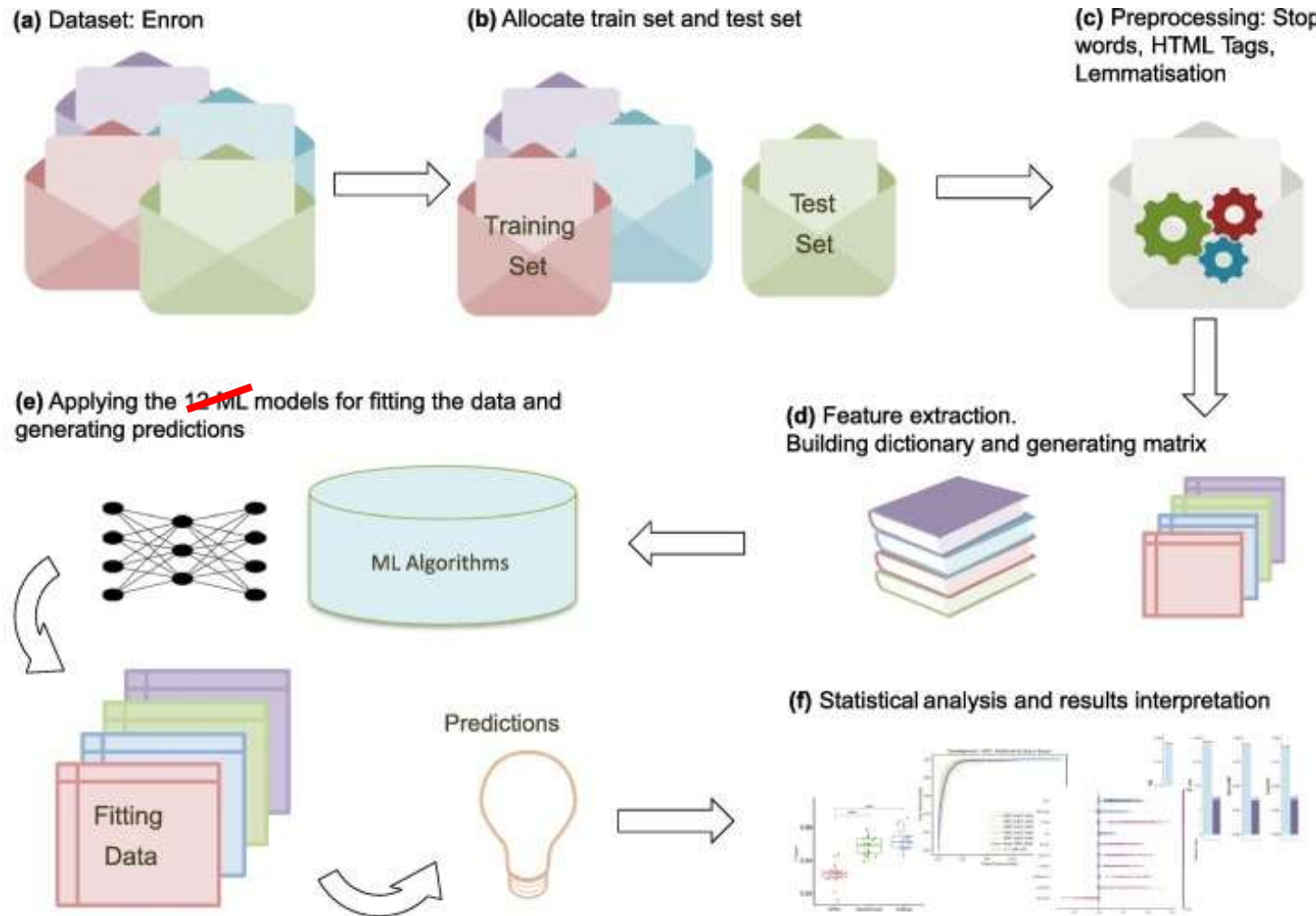


Fig. 1. **Pipeline diagram of the preprocessing and classification steps.** The pipeline shows the process applied for investigating the performance of 12 classifiers. (a) The Enron dataset is selected for the study; (b) the train and test sets are allocated (70% train set and 30% test set); (c) the preprocessing stage is then applied, including removal of stop words, HTML tag and lemmatisation; (d) the features are then extracted for generating a dictionary, based on the number of most occurring features; (e) finally, the 12 algorithms receive the matrices for fitting the data and predicting the classification outcomes; (f) comparative analysis is then performed to assess the statistical significance of the results and provide an accurate interpretation of the machine learning classification outcomes.



THE PIPELINE

Fig. 1. **Pipeline diagram of the preprocessing and classification steps.** The pipeline shows the process applied for investigating the performance of 12 classifiers. (a) The Enron dataset is selected for the study; (b) the train and test sets are allocated (70% train set and 30% test set); (c) the preprocessing stage is then applied, including removal of stop words, HTML tag and lemmatisation; (d) the features are then extracted for generating a dictionary, based on the number of most occurring features; (e) finally, the ~~12~~ algorithms receive the matrices for fitting the data and predicting the classification outcomes; (f) comparative analysis is then performed to assess the statistical significance of the results and provide an accurate interpretation of the machine learning classification outcomes.

Model

Naïve Bayes

The most common machine learning algorithm for text classification.

FOR THE PIPELINE



recipes

R-CMD-check passing codecov 95% CRAN 1.3.0
downloads 219K/month lifecycle stable

Introduction

With recipes, you can use `dplyr`-like pipeable sequences of feature engineering steps to get your data ready for modeling. For example, to create a recipe containing an outcome plus two numeric predictors and then center and scale (“normalize”) the predictors:



textrecipes

Introduction

`textrecipes` contain extra steps for the `recipes` package for preprocessing text data.

The `recipes` package that you can use to combine different feature engineering and preprocessing tasks into a single object and then apply these transformations to different data sets.



Join us!



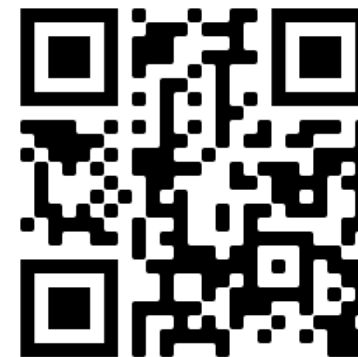
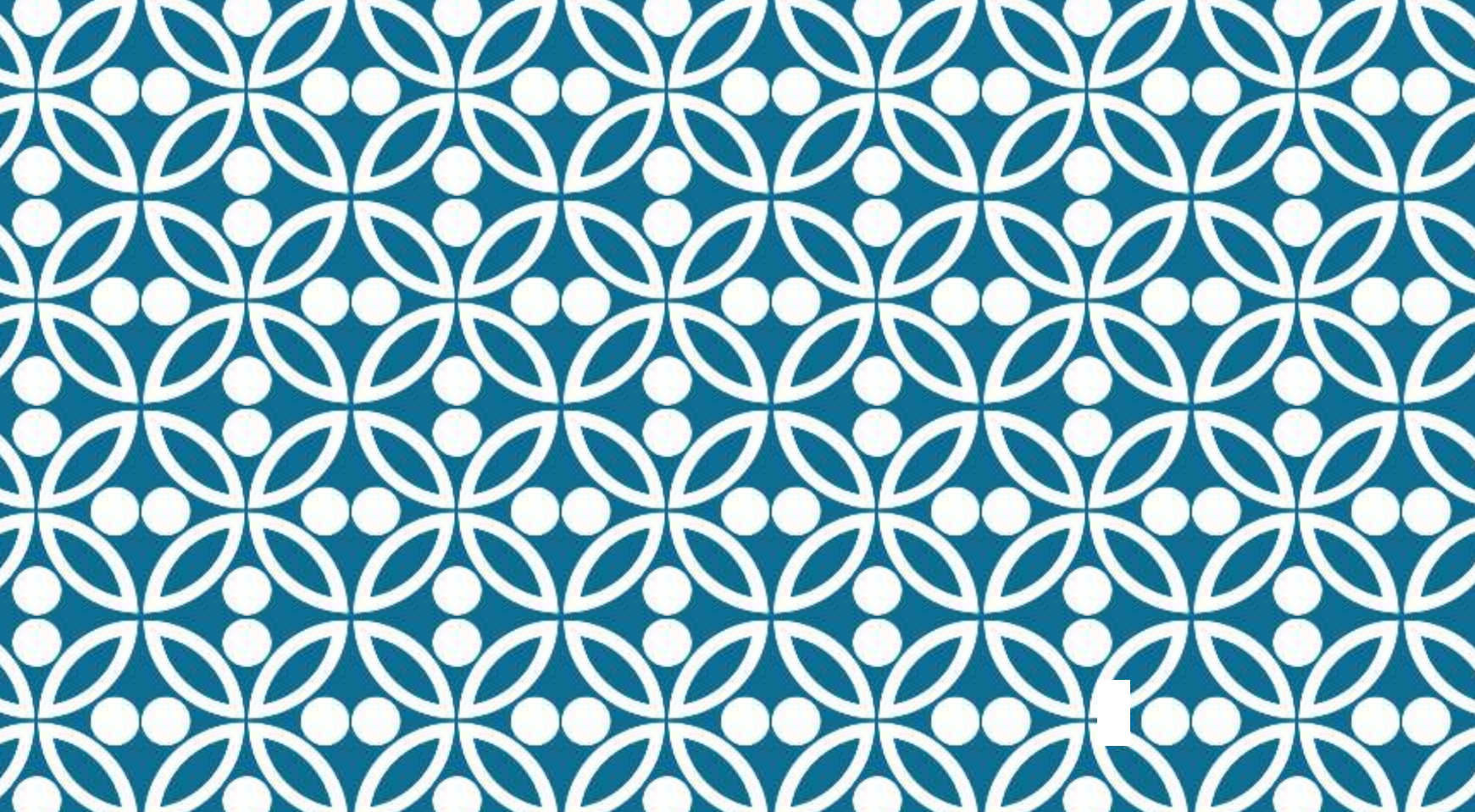
<https://forms.office.com/e/8Bgd2YsasJ>

All about the lab:

<https://societal-analytics.nl/>

Contact us at:

analytics-lab.fsw@vu.nl



<https://sofiagil.github.io/>

THANKS!

Dr. Sofia Gil-Clavel