

DATA SCIENCE LIFECYCLE FROM DATA TO IMPACT

Afshin Ashofteh

[🔗Link!](#)

Nova Information Management School (Nova IMS)
Nova University of Lisbon

Email: aashofteh@novaims.unl.pt

Webpage: <https://novaresearch.unl.pt/en/persons/afshin-ashofteh>

25 February | 2.30 pm (GMT) | online

[E-mail](#) | [Academic HP](#) | [ResearchGate](#) | [Discussion Group](#) | [YouTube](#)

ABSTRACT:

This talk explores how data science and AI bring the impact of new data sources into our modern world through technologies like LLMs, AI agents, data spaces, web platforms, etc. Many professionals need new skills to automate, analyze, and optimize complex systems. Adopting these technologies requires updated knowledge, better methodologies, and improvements in quality, security, privacy, and legal frameworks. It also demands a diverse skill set for data scientists and AI specialists. This talk introduces a model integrating data science, data engineering, software development, Artificial Intelligence, and essential technical and soft skills for data professionals.

OUTLINE:

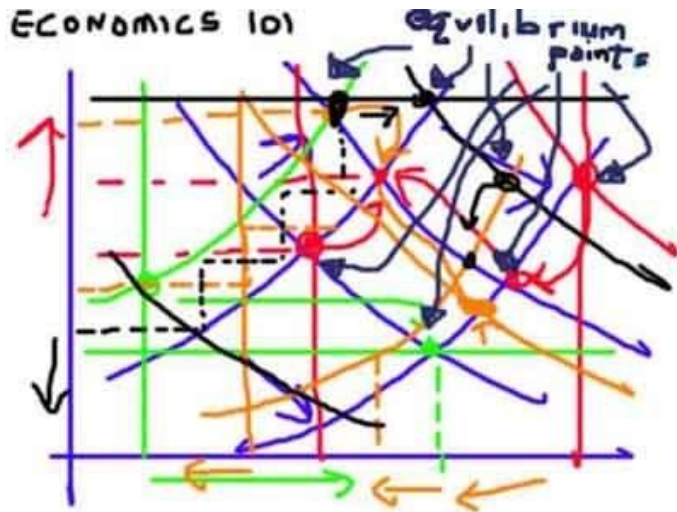
- Data Science Lifecycle
- New Technologies
- Examples



TWO KINDS OF NOBELISTS IN FINANCIAL MODELING!

Academia

Specialists in the academy who have not practiced what they teach!



Industry

Specialists in the industry who offer explanations absent of any rigorous academic theory!

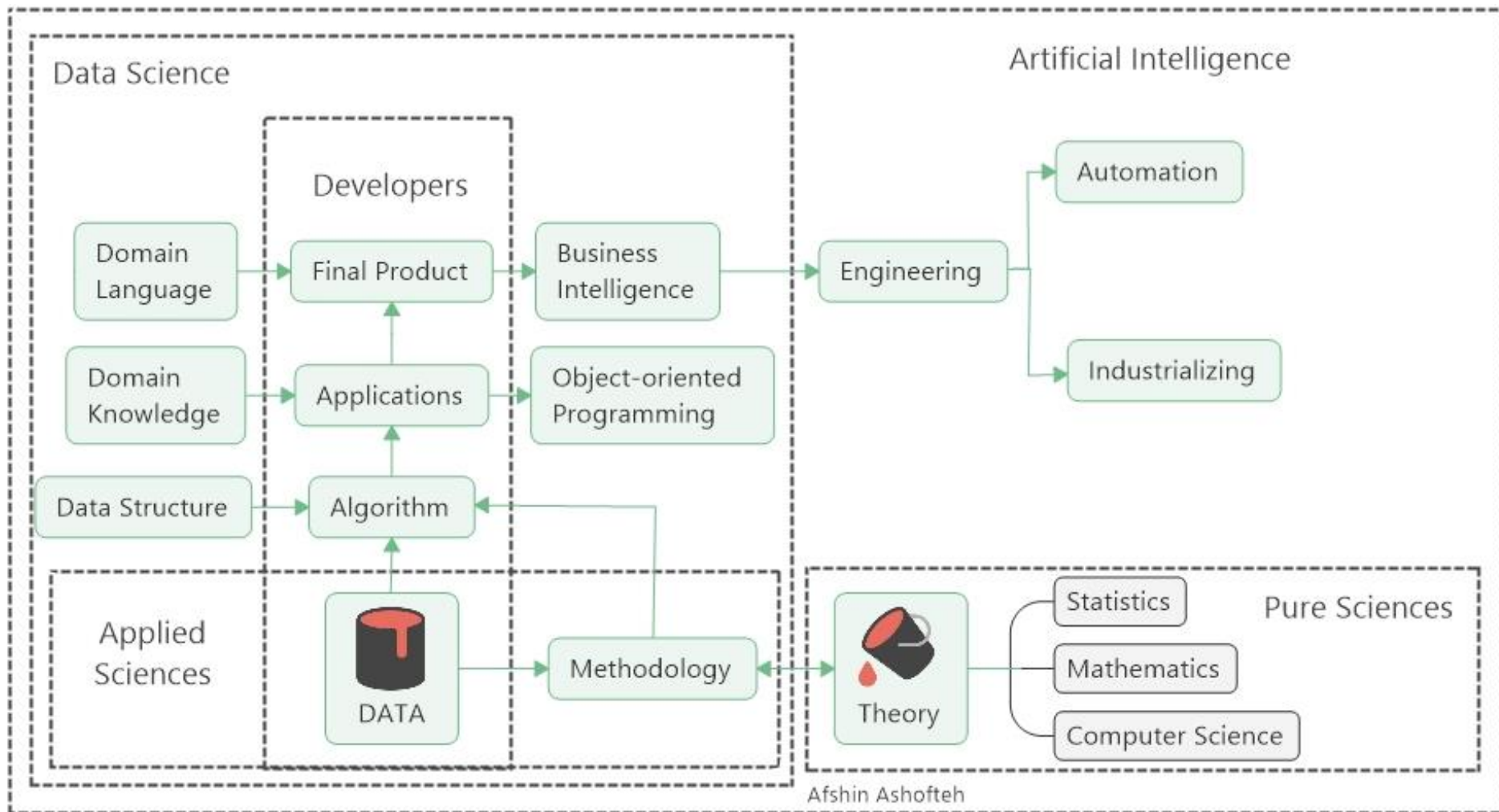


- Contain extremely elegant mathematics!
- Describes a world that does not exist!

- Good fundamental analysts!
- Misuse mathematical tools to describe actual observations!

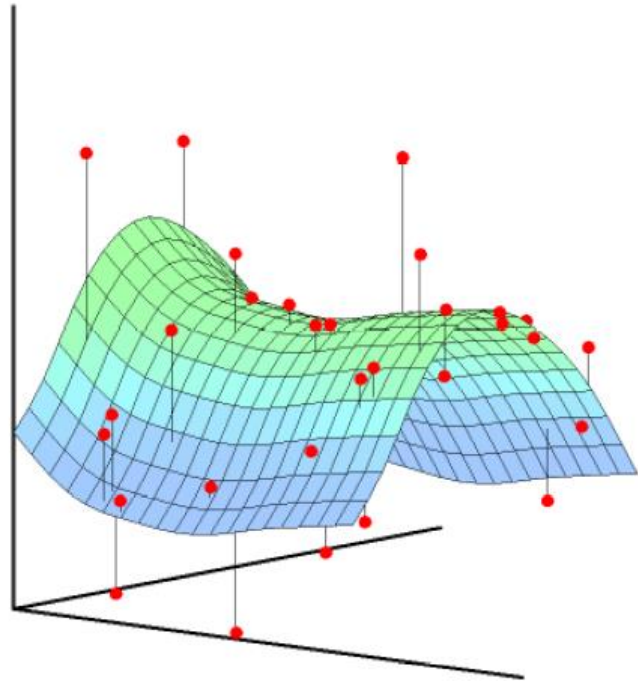
(OVERFITTING to PURE KNOWLEDGE)

(OVERFITTING to PRACTICAL KNOWLEDGE)



Ashofteh, A., & Bravo, J. M. (2021). Data Science Training for Official Statistics: a New Scientific Paradigm of Information and Knowledge Development in National Statistical Systems. *Statistical Journal of the IAOS*, 37(3), 771 – 789.

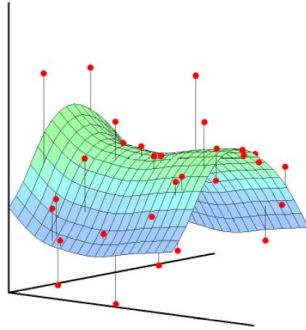


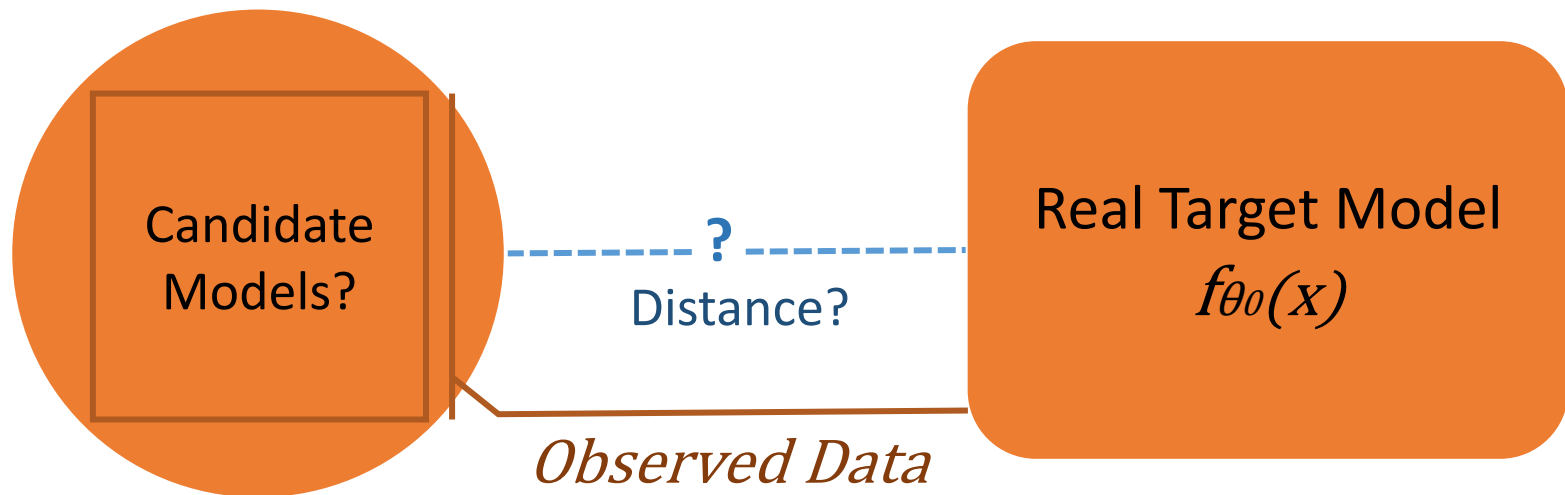
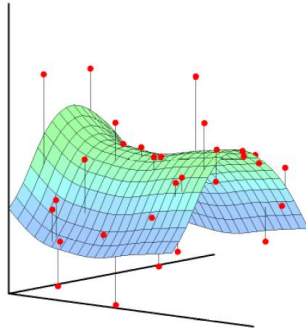


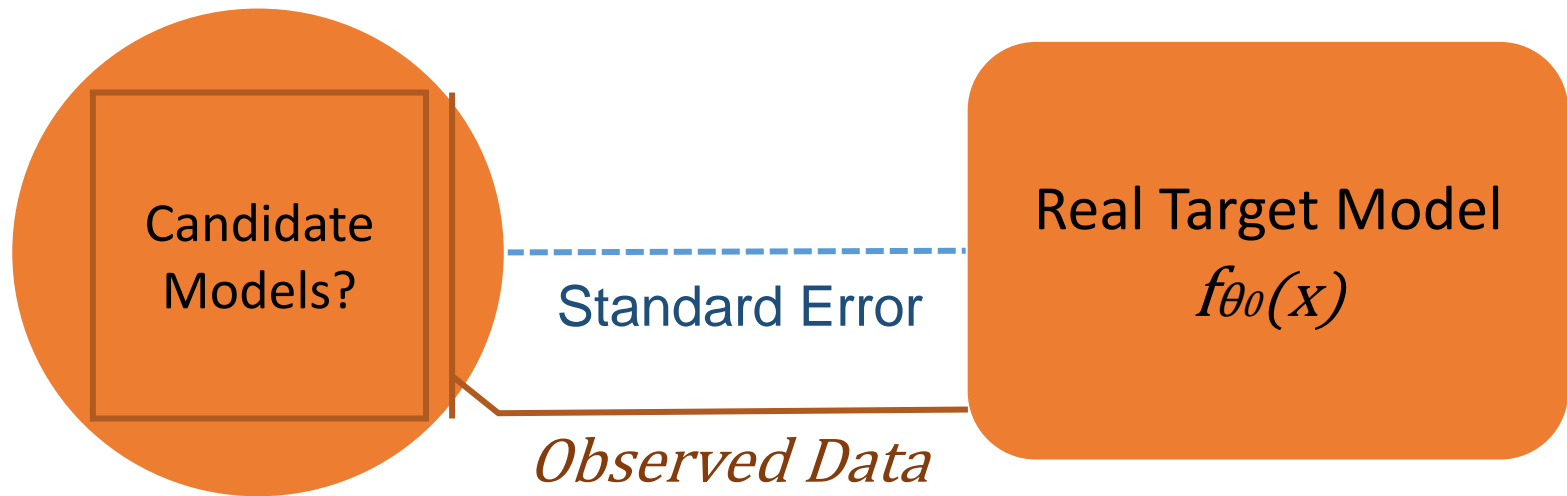
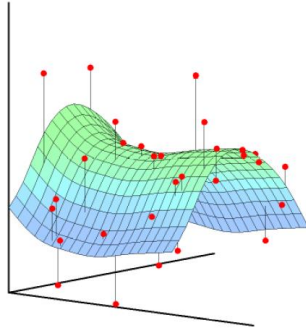
Observed Data

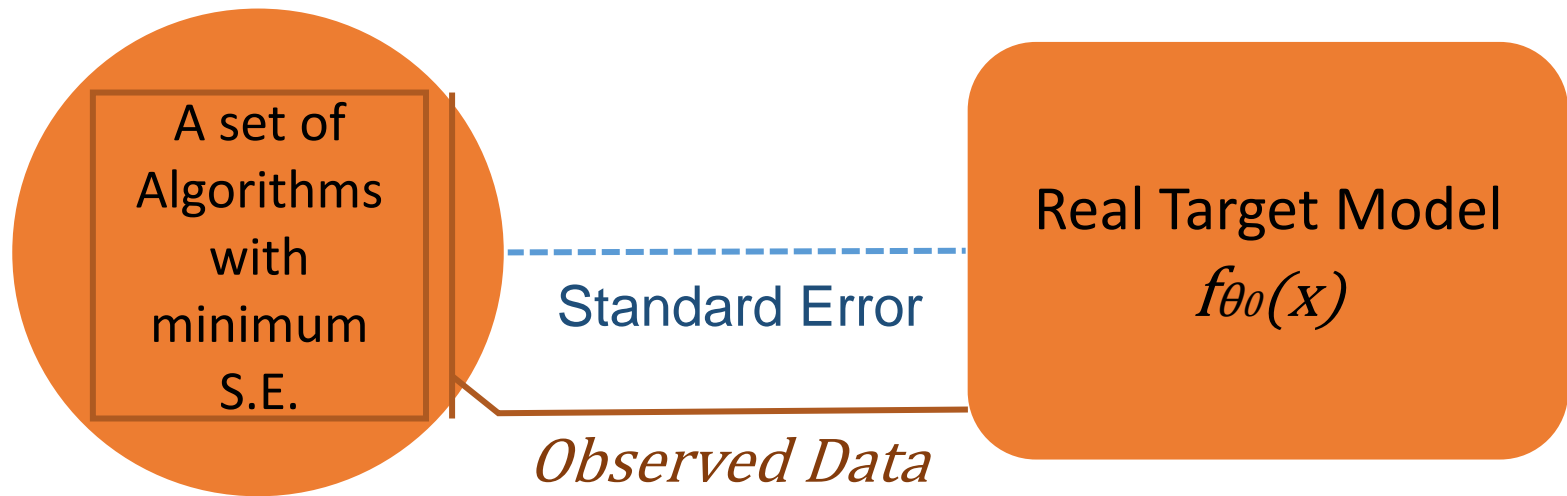
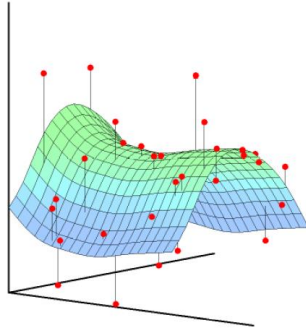
Real Target Model

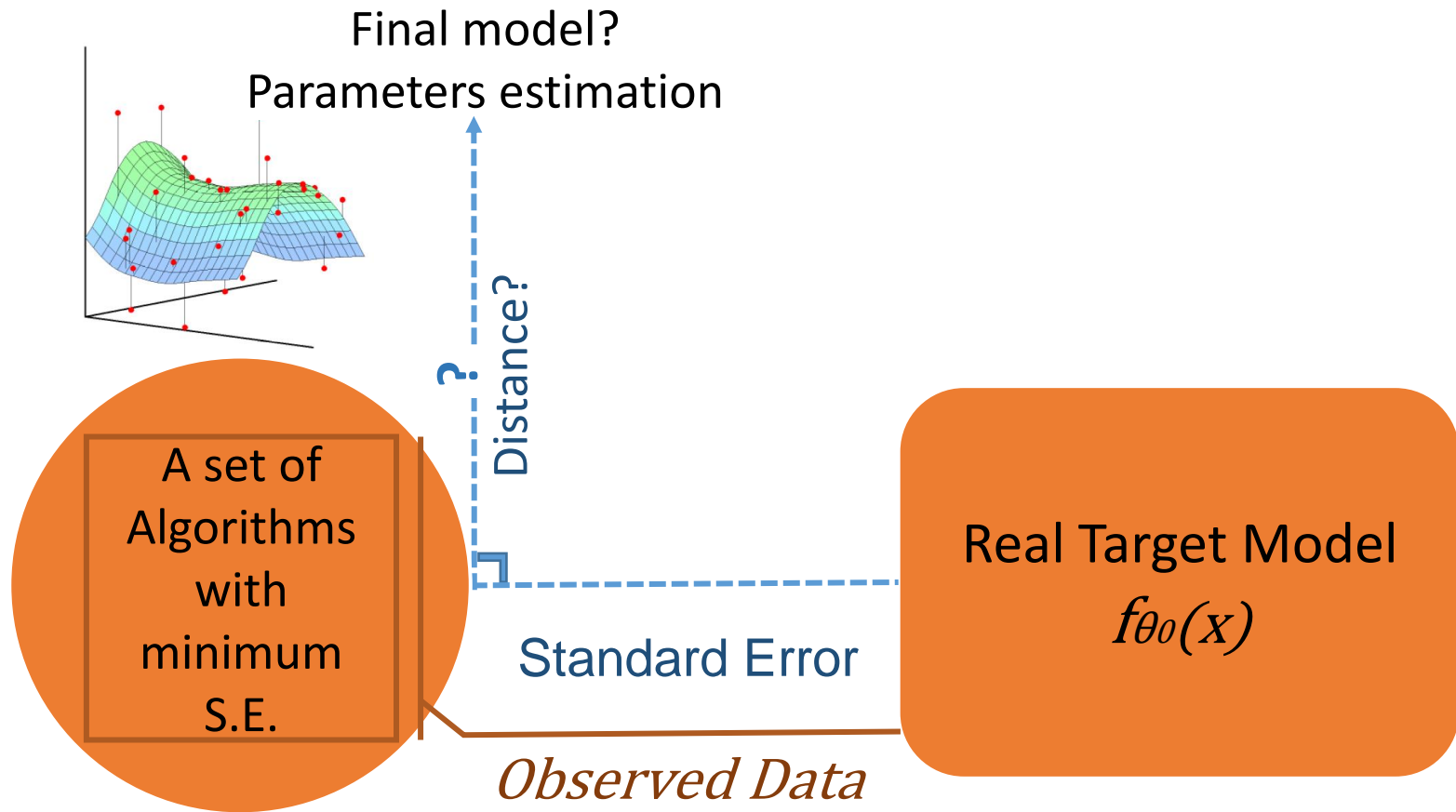
$$f_{\theta_0}(x)$$

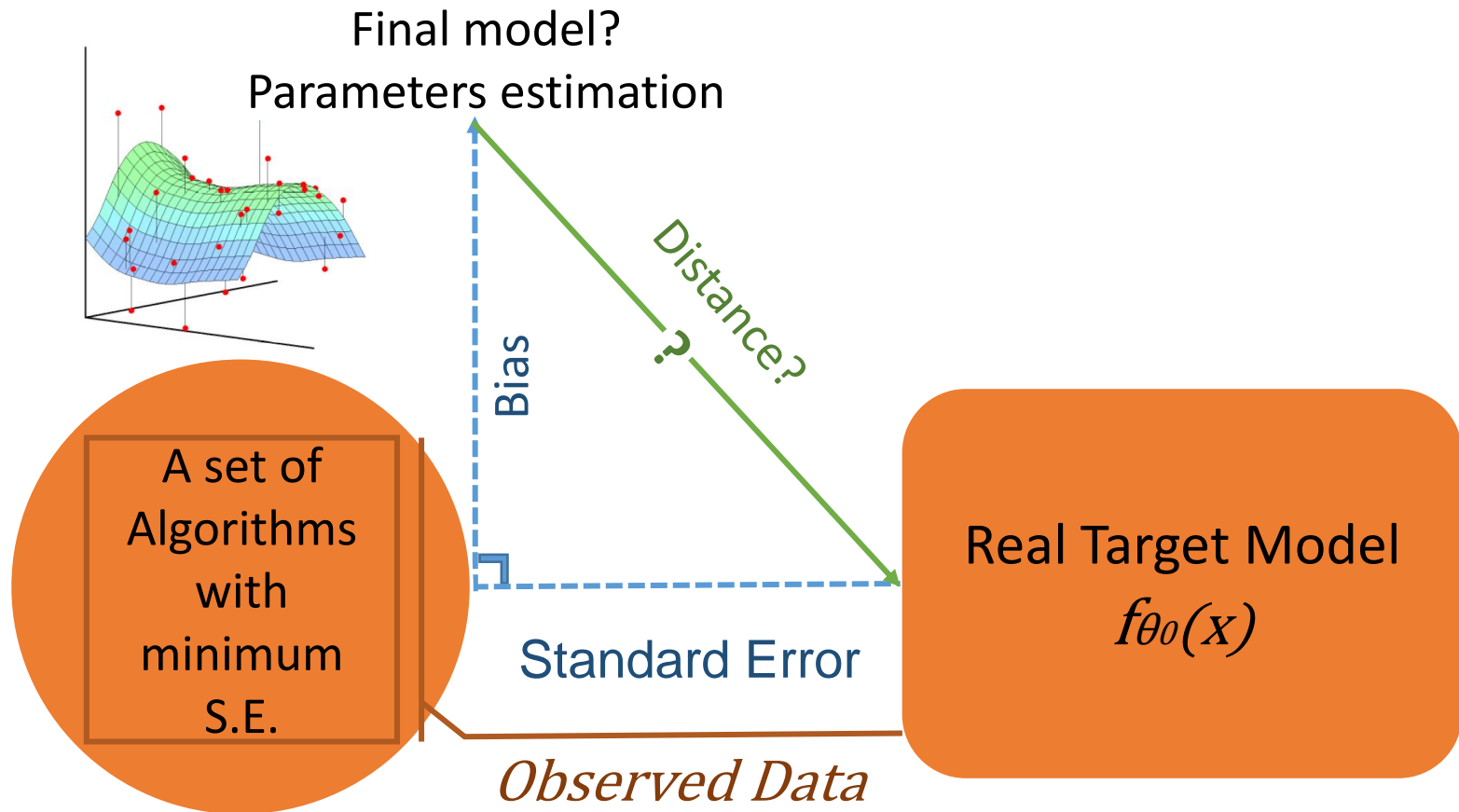


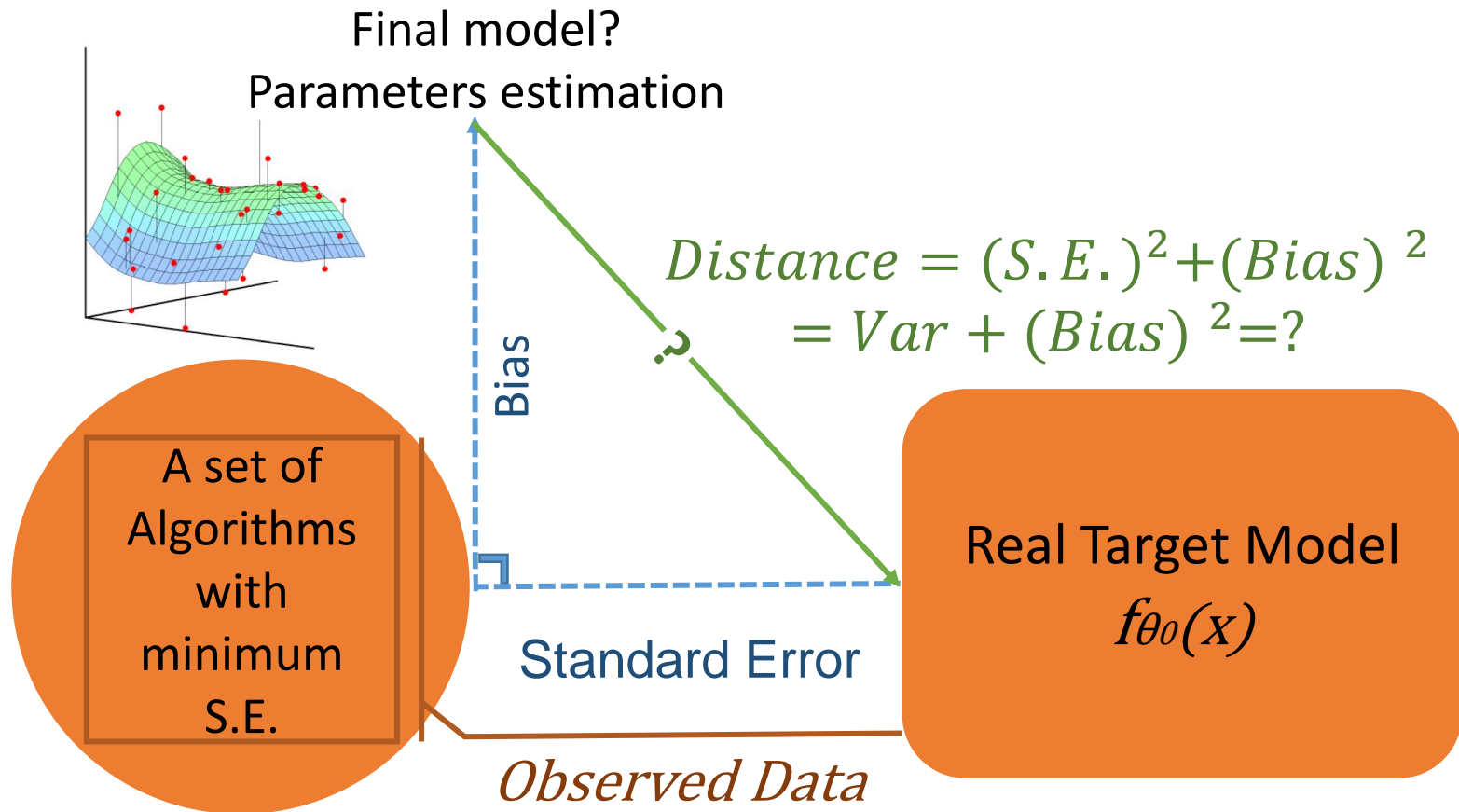


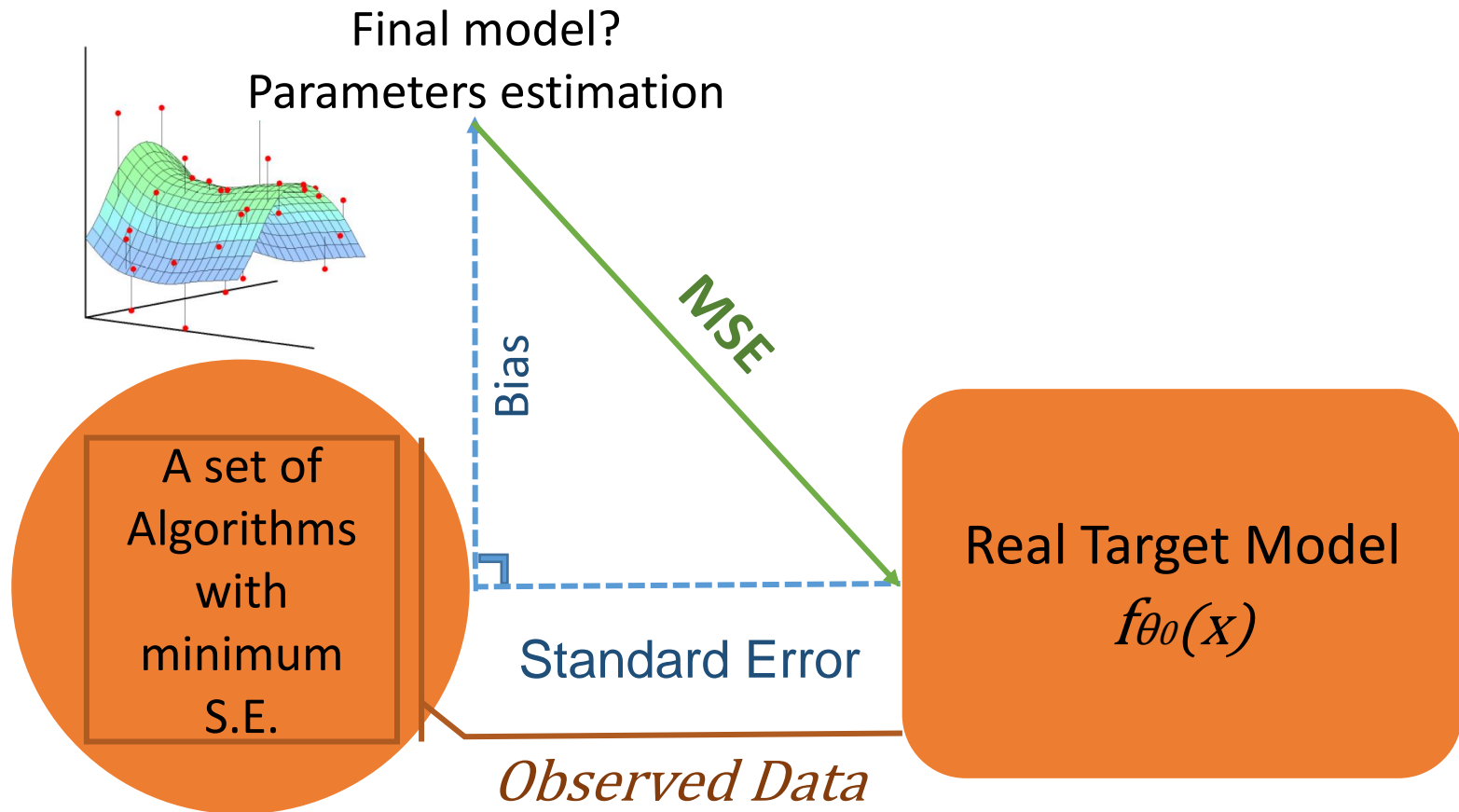


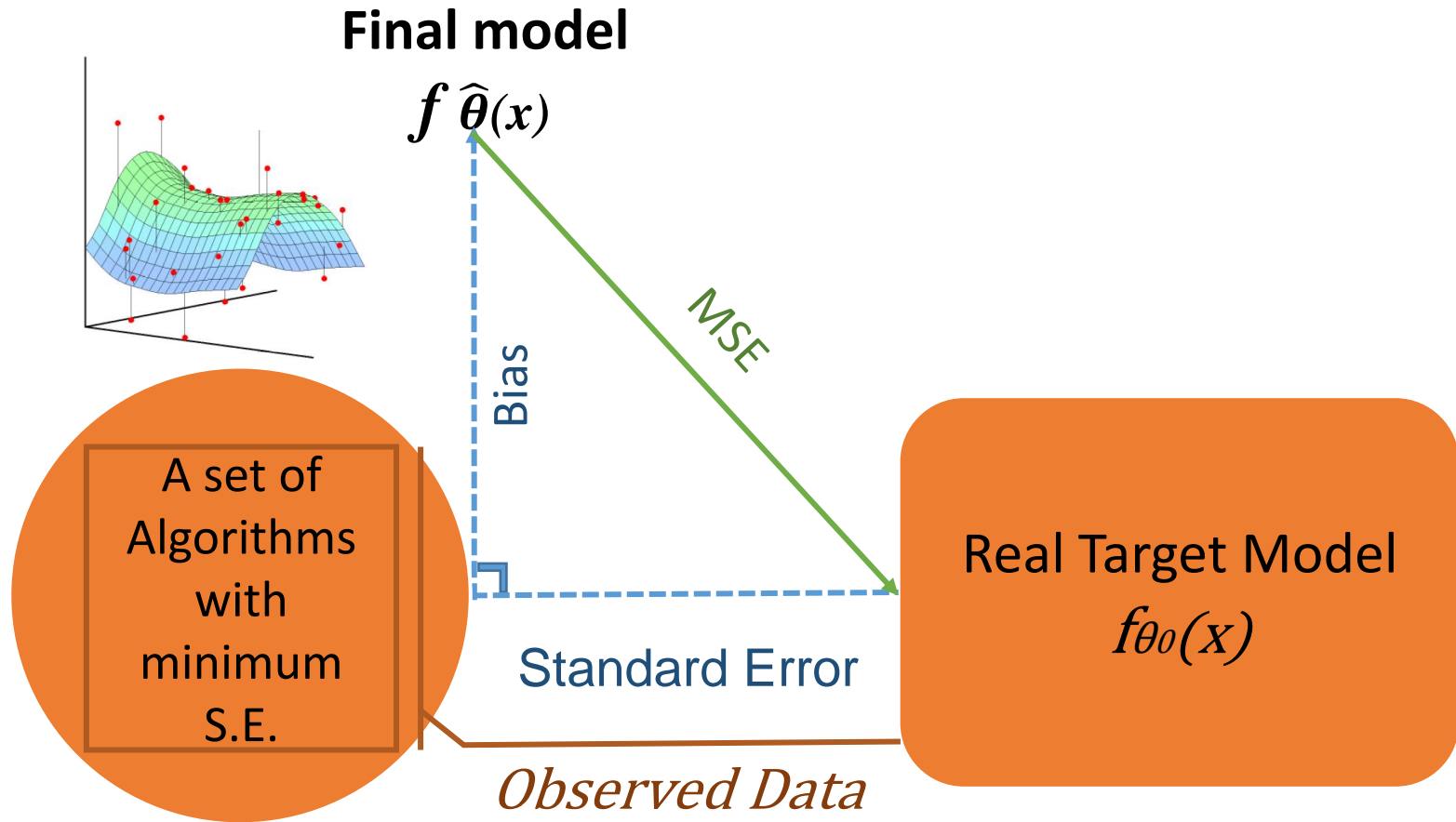


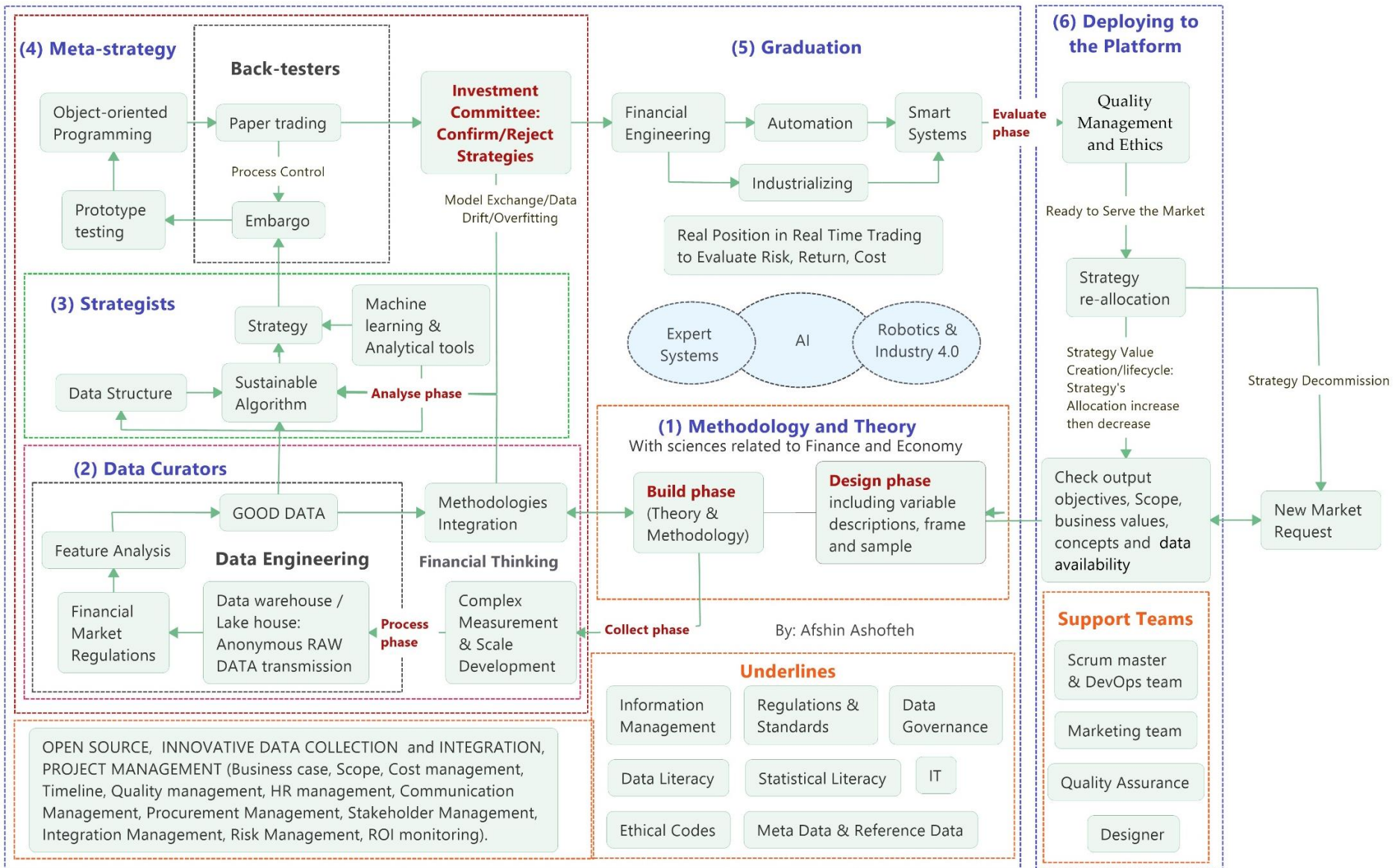






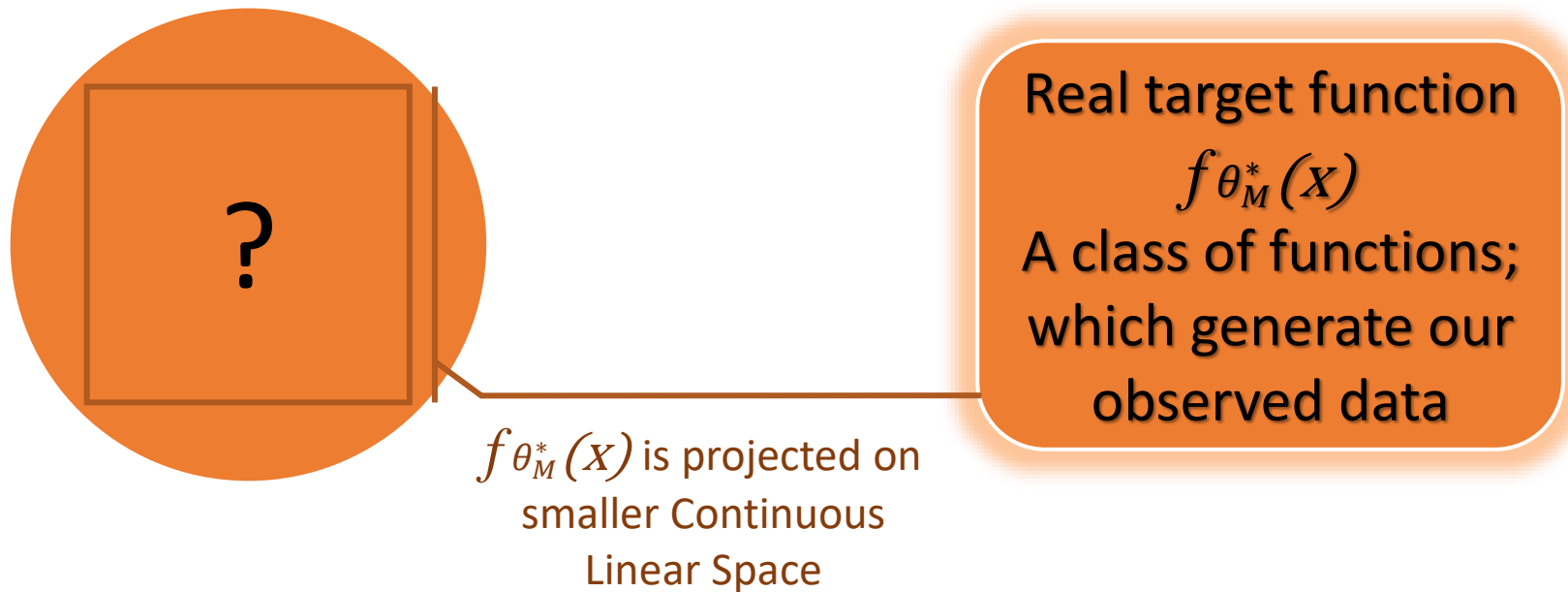


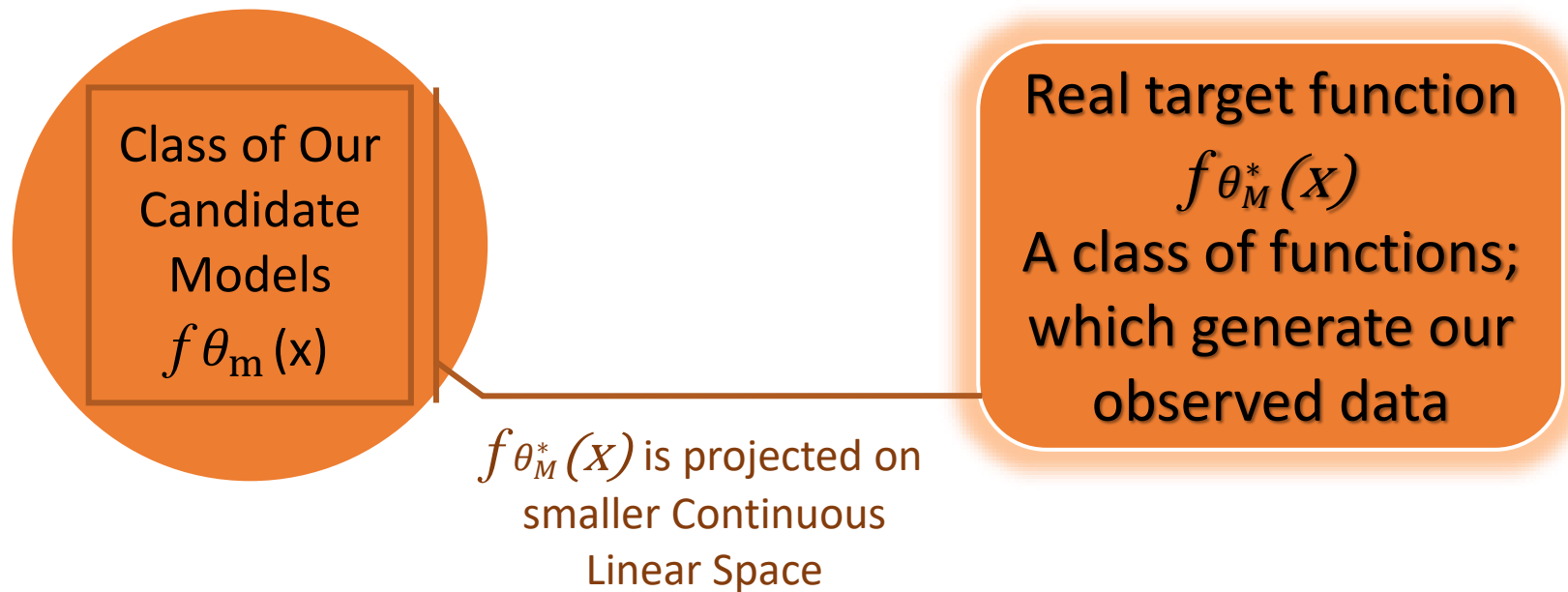


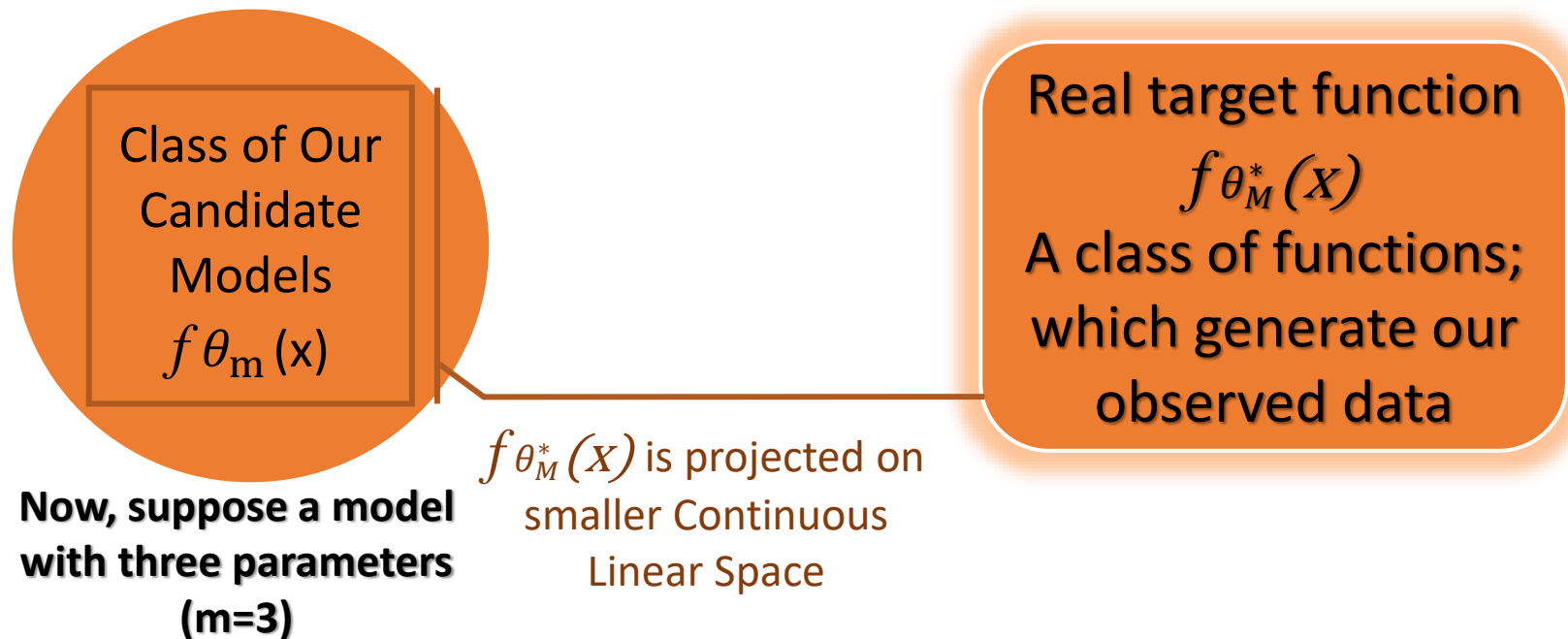


Paper: https://doi.org/10.1007/978-3-031-39864-3_2

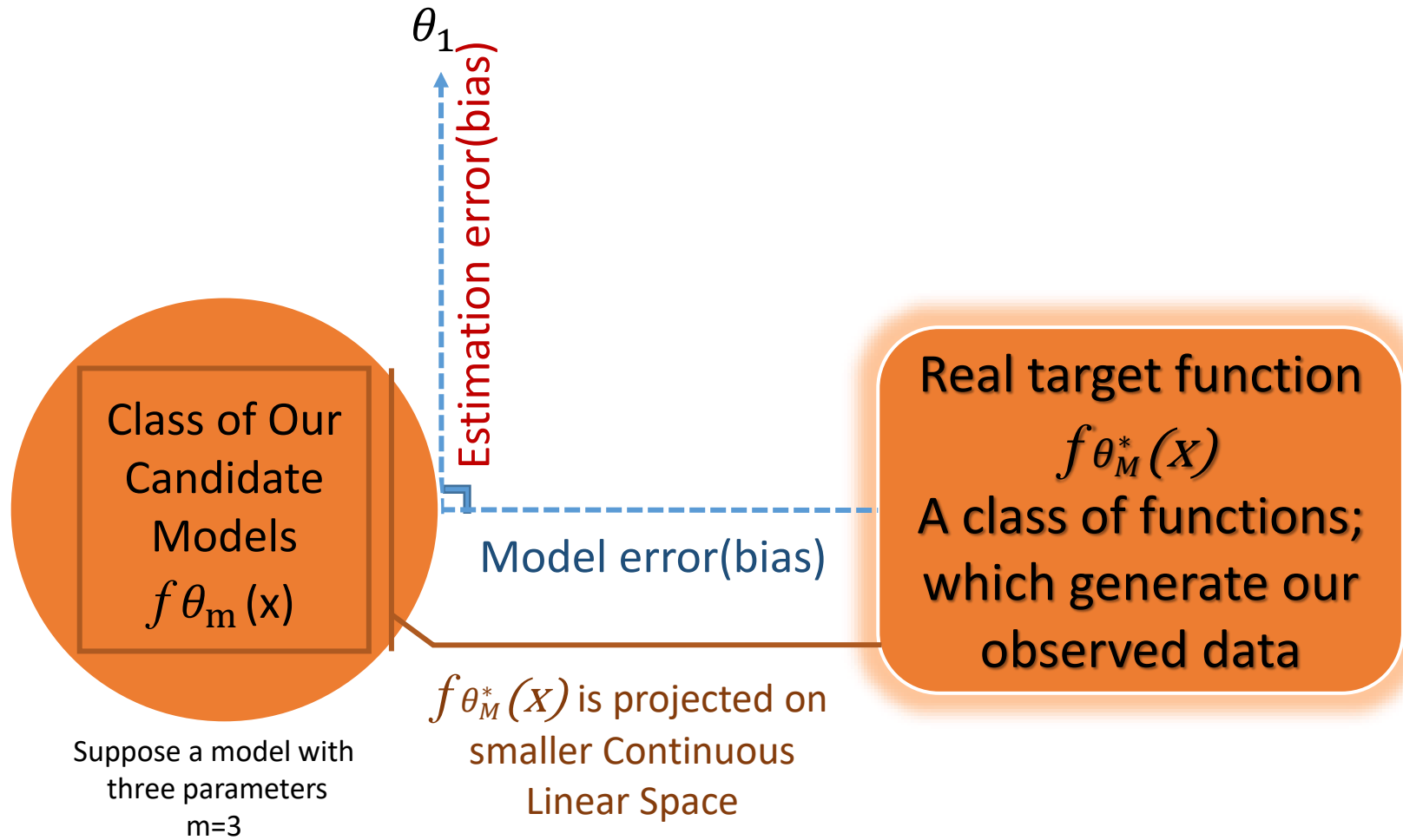




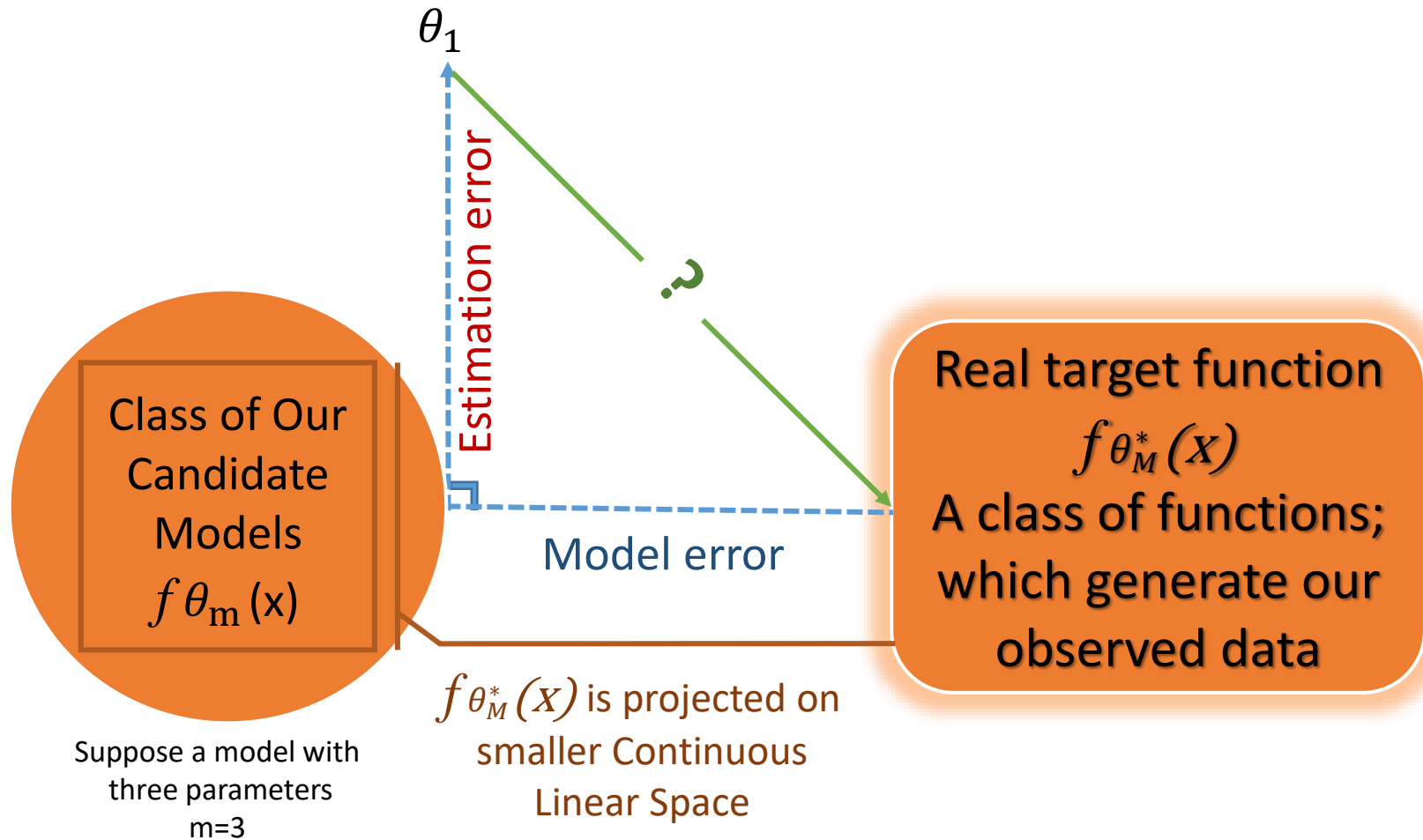




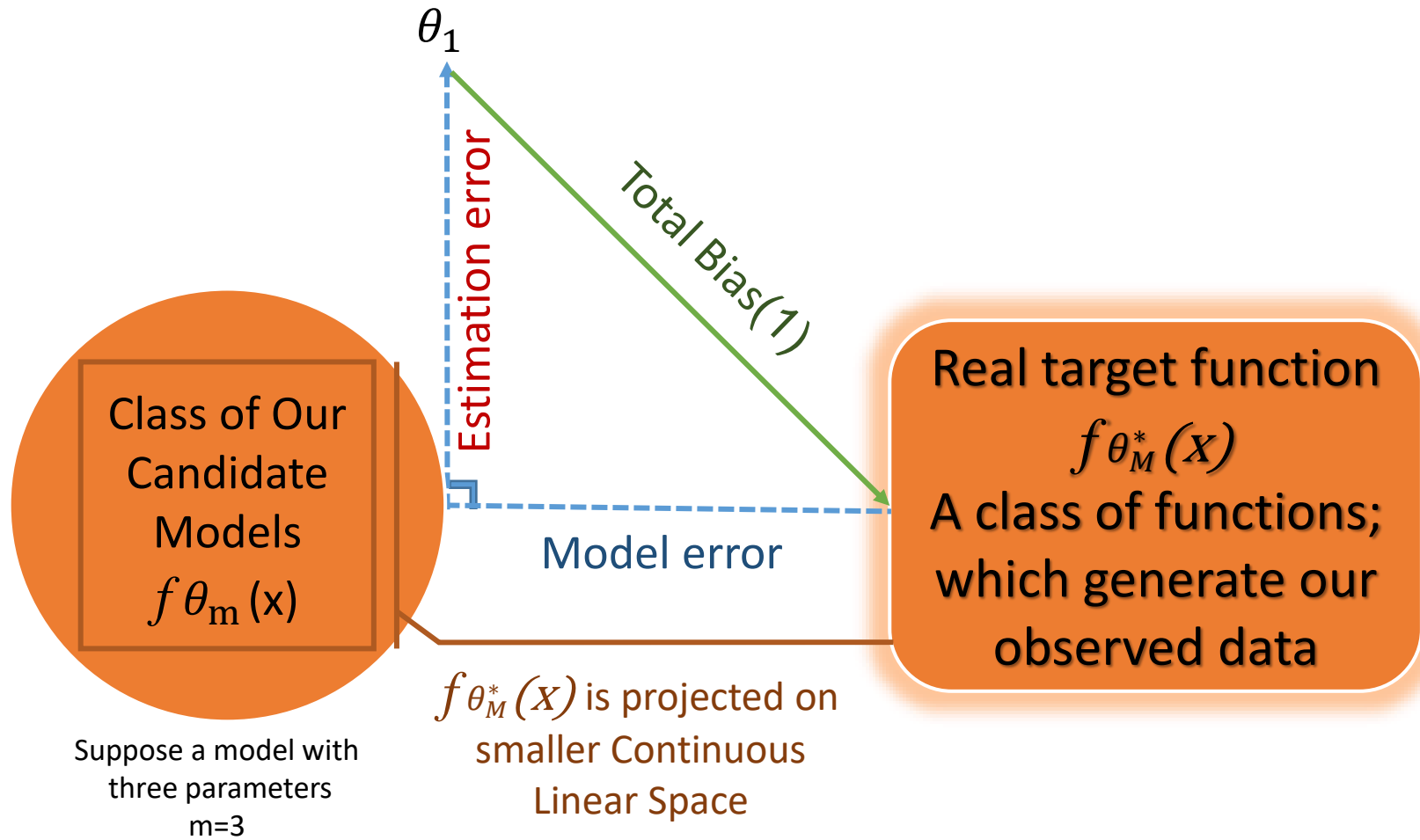
Our approximation for



Our approximation for

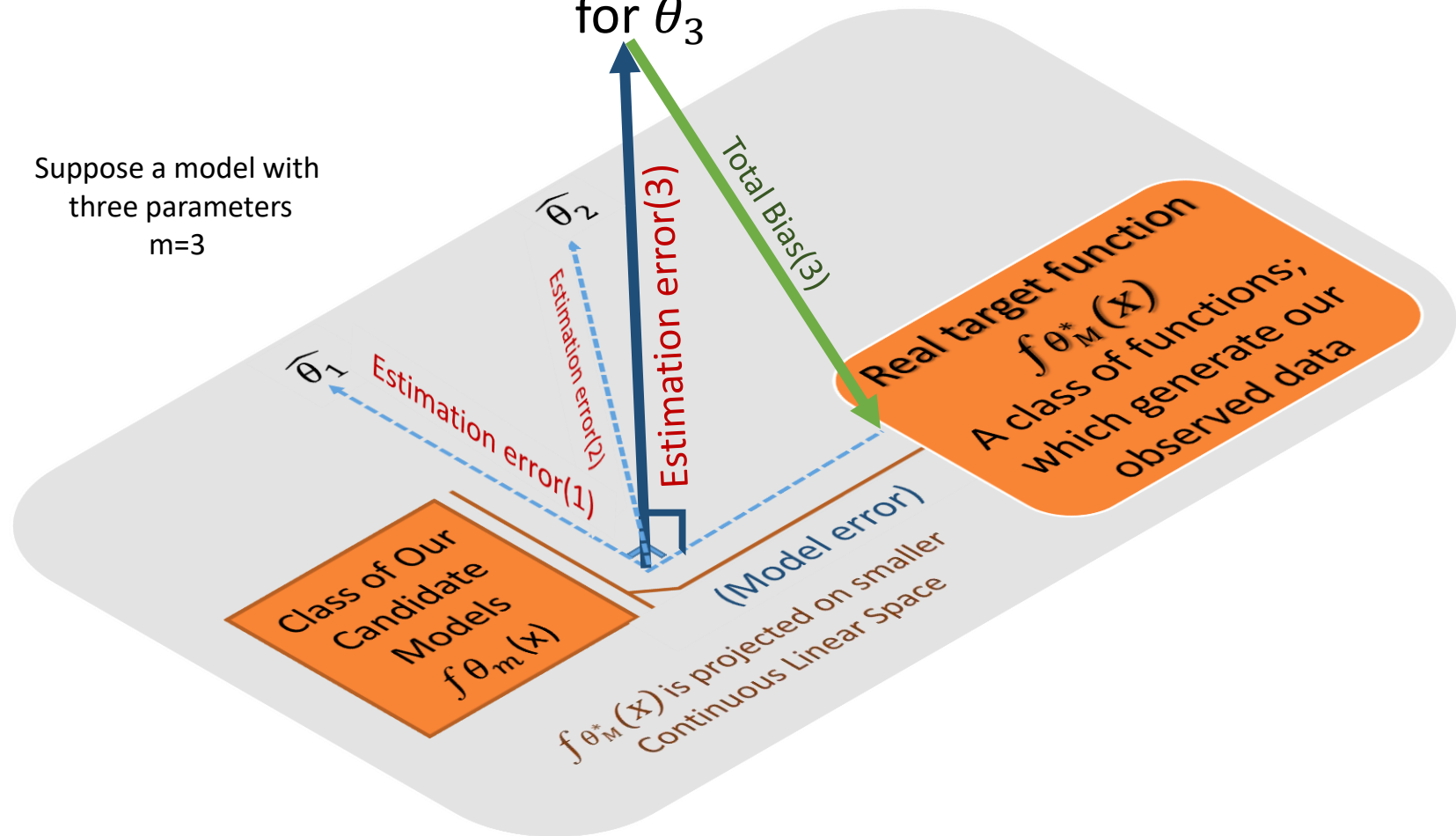


Our approximation for



Our approximation
 for θ_3

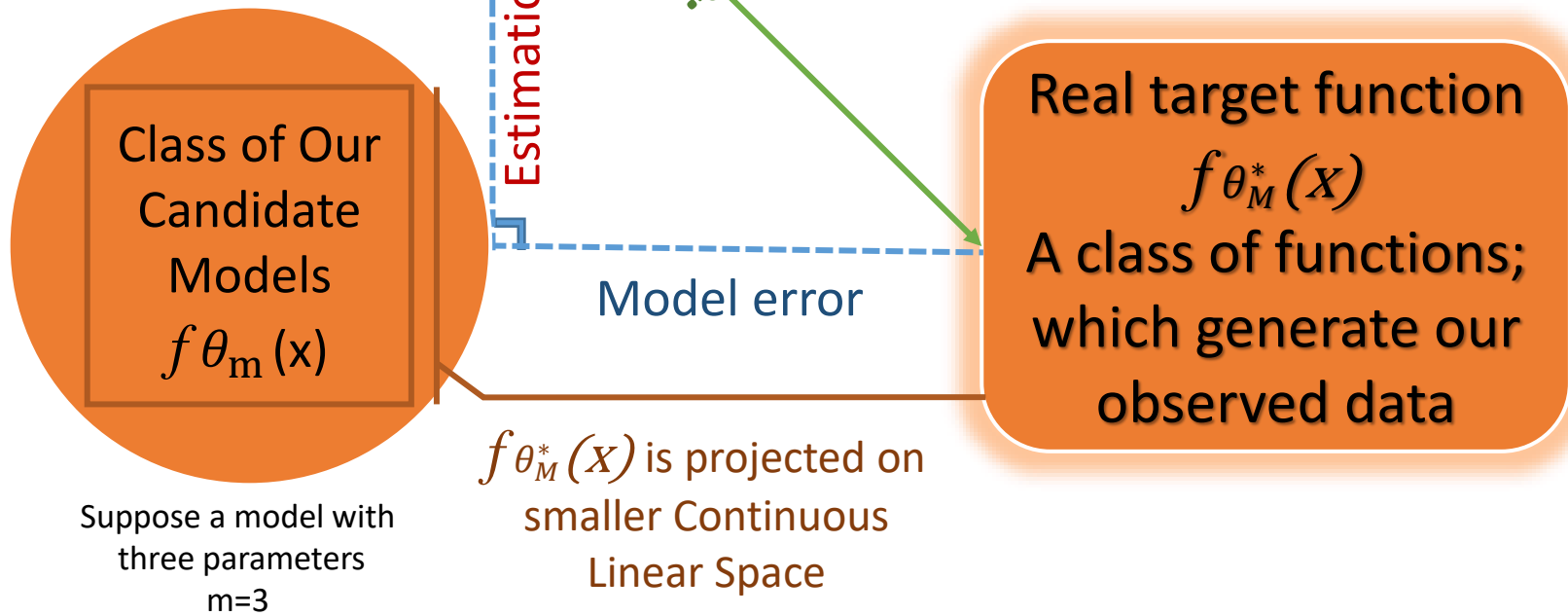
Suppose a model with
 three parameters
 $m=3$



Check this video for more details: <https://youtu.be/mVXqQCViEfo>

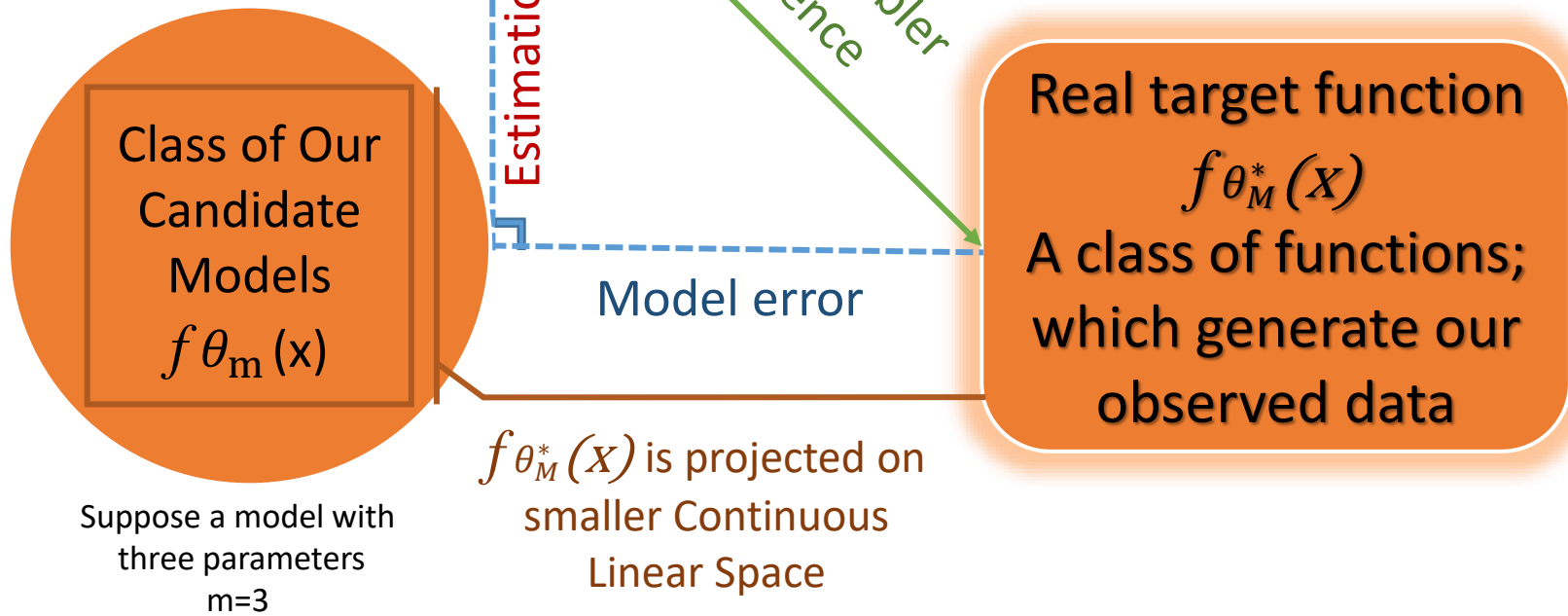
Class of suggested model

$$f(\widehat{\theta}_1, \widehat{\theta}_2, \widehat{\theta}_3)(x)$$



Class of suggested model

$$f(\widehat{\theta}_1, \widehat{\theta}_2, \widehat{\theta}_3)(x)$$



**Kullback-Leibler
Divergence**

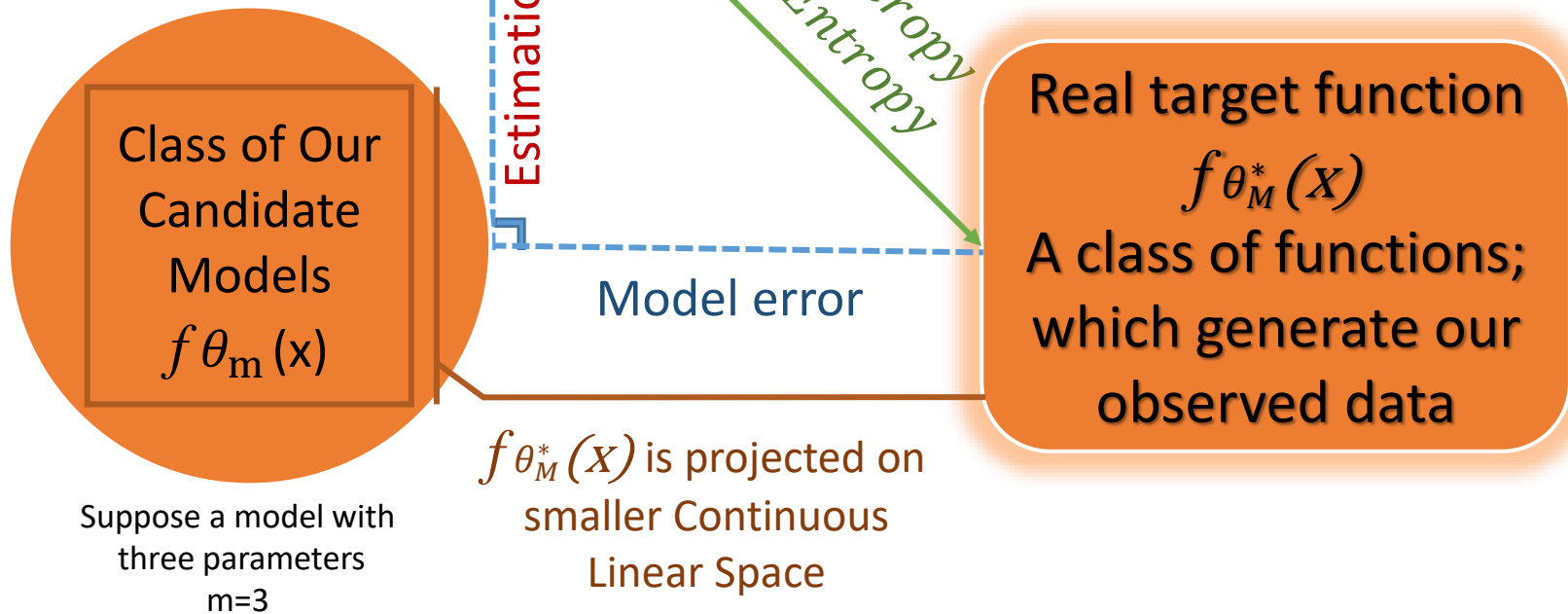


$$\begin{aligned}
 KL(\theta^*, \theta) &= \int \log \left\{ \frac{f_M^*(x)}{f_m(x)} \right\} f_M^*(x) dx \\
 &= \int \{ \log f_M^*(x) - \log f_m(x) \} f_M^*(x) dx \\
 &= \int \log f_M^*(x) f_M^*(x) dx - \int \log f_m(x) f_M^*(x) dx \\
 &= H(\theta^*, \theta^*) - H(\theta, \theta^*)
 \end{aligned}$$

$KL(\theta^, \theta) = \text{Shanon Entropy} - \text{Cross Entropy}$*

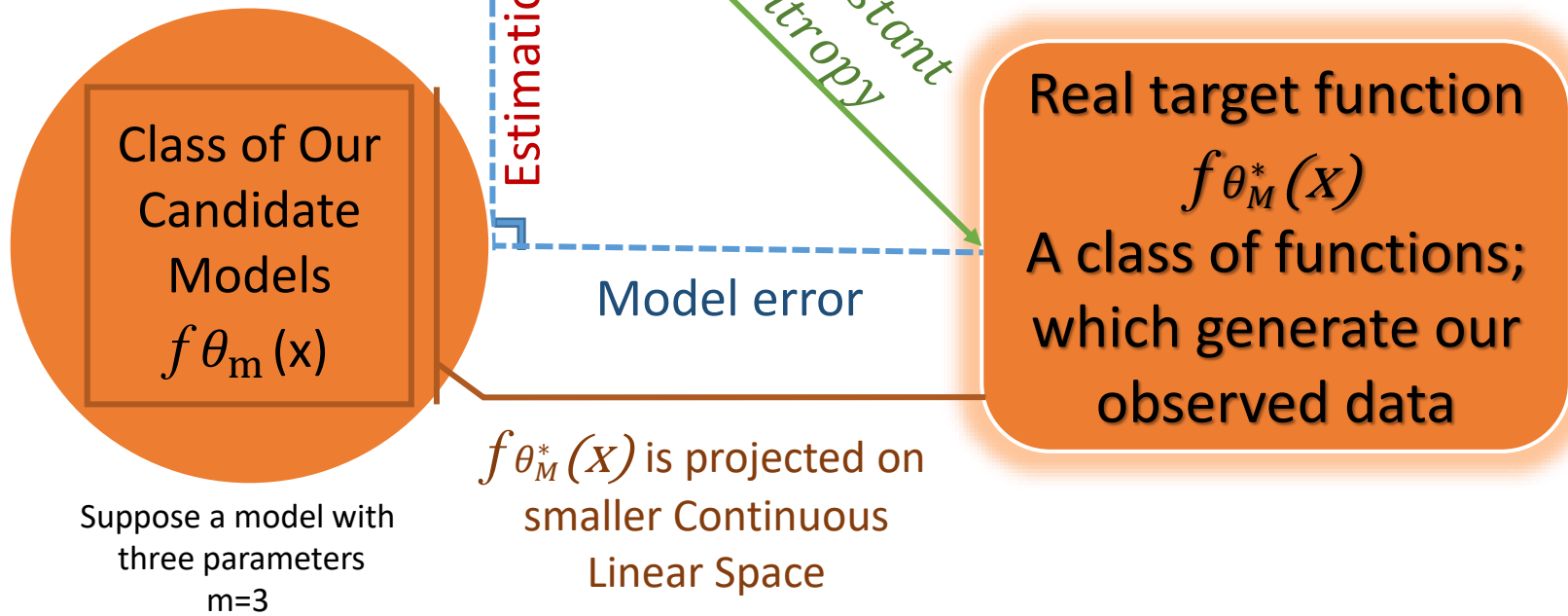
Class of suggested model

$$f(\widehat{\theta}_1, \widehat{\theta}_2, \widehat{\theta}_3)(x)$$



Class of suggested model

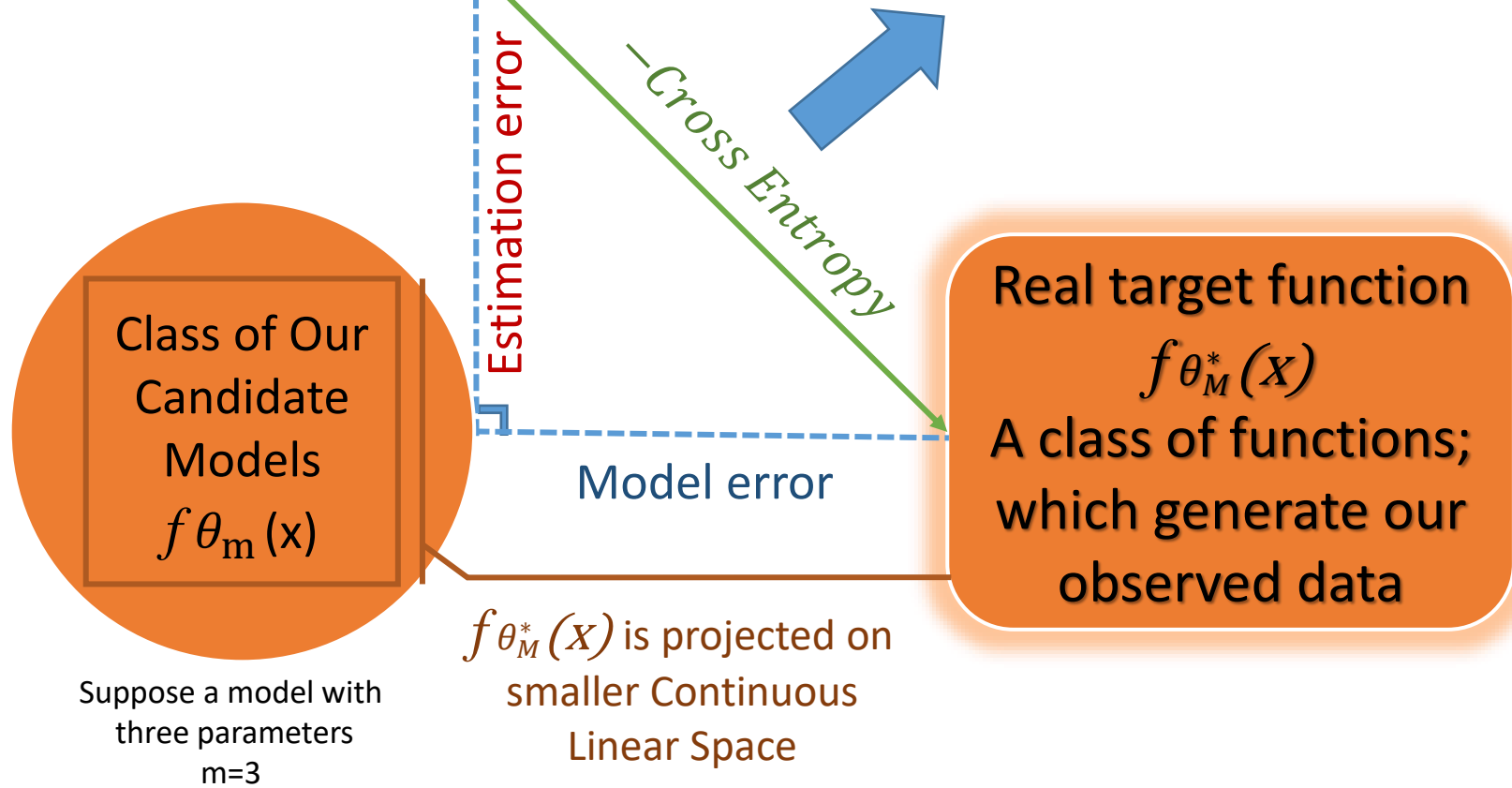
$$f(\widehat{\theta}_1, \widehat{\theta}_2, \widehat{\theta}_3)(x)$$



Class of suggested model

$$f(\widehat{\theta}_1, \widehat{\theta}_2, \widehat{\theta}_3)(x)$$

Minimum Distance = Maximum Cross Entropy



**MAXIMIZING THE CROSS ENTROPY BY USING
 OBSERVED DATA**

$$Cross\ Entropy = \int \log f_m(x) \cdot f_M^*(x) dx$$

X is a Random Variable ... We cannot use Riemann Integral ...
 We should solve it Statistically ... In Statistics we have:

$$\int xf(x)dx = E(X) = \bar{x} = Average = \sum_{i=1}^n x_i \cdot p(x_i)$$

and **Strong law of Large numbers**

$$x_1, x_2, \dots, x_n \text{ iid } \bar{x} \rightarrow \mu \text{ as } n \rightarrow \infty$$

Solution: calculate Average of observed data

$$\text{CrossEntropy} \approx E_{\theta_0}(\log f_m(x)) = \sum_{i=1}^n f_m(x) \cdot \log f_m(x) \quad ; \quad x_i \sim f_M^*(x)$$

**Strong law of
Large numbers**

**Kullback-Leibler
Divergence**



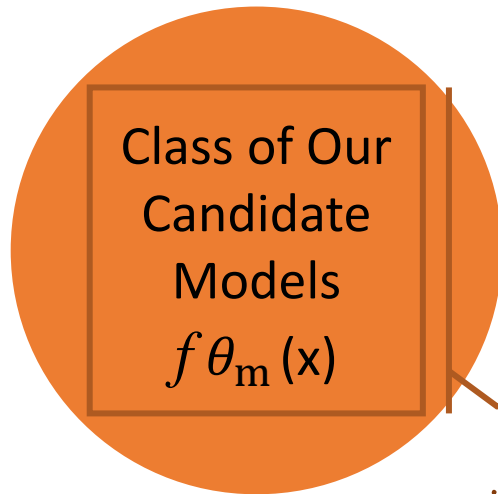
Class of suggested model

$$f(\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3)(x)$$

Minimum Distance = Maximum $\sum_{i=1}^n P_i \log P_i$
 = **Maximum Likelihood**

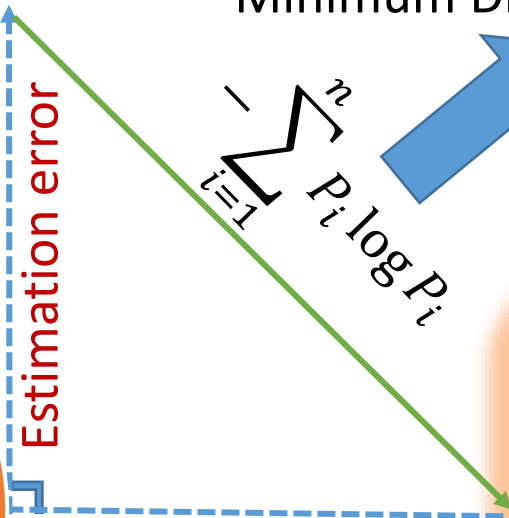
$$\sum_{i=1}^n P_i \log P_i$$

If we have the same dimension
 i.e. $M=m$



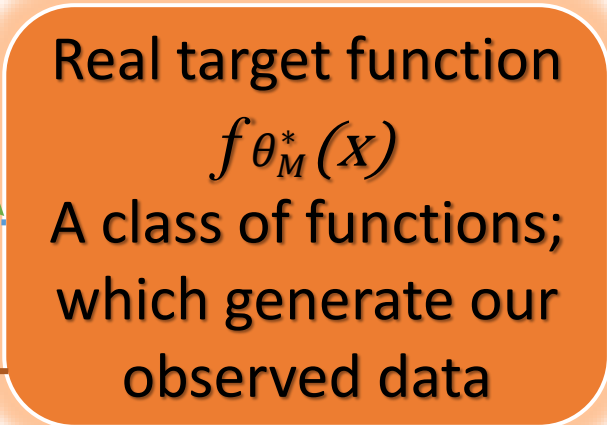
Class of Our
 Candidate
 Models
 $f_{\theta_m}(x)$

Suppose a model with
 three parameters
 $m=3$

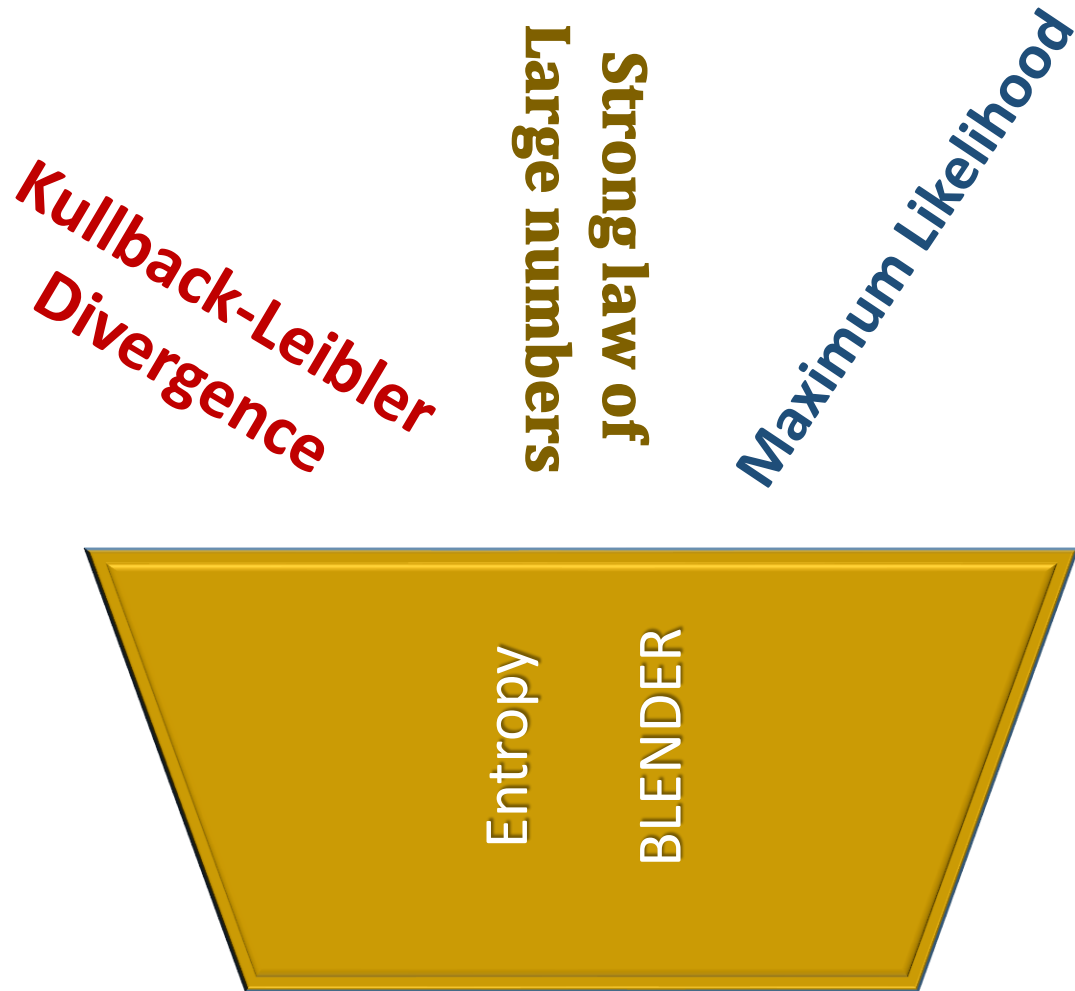


Model error

$f_{\theta_M^*}(x)$ is projected on
 smaller Continuous
 Linear Space



Real target function
 $f_{\theta_M^*}(x)$
 A class of functions;
 which generate our
 observed data



*Maximum Cross Entropy = **Maximum Likelihood***

But we don't have $f_M^*(x)$ and it is unknown!

Again Statistics helps:

Central Limit theorem

x_1, x_2, \dots, x_n iid ; $0 < \text{Var}(x) < \infty$; $\bar{X} \sim N(\mu_x, \frac{\sigma^2}{n})$ as $n \rightarrow \infty$

Solution:

Use *C.L.T.* and estimate *parameters of Normal Distribution* from *observed data* to calculate not only

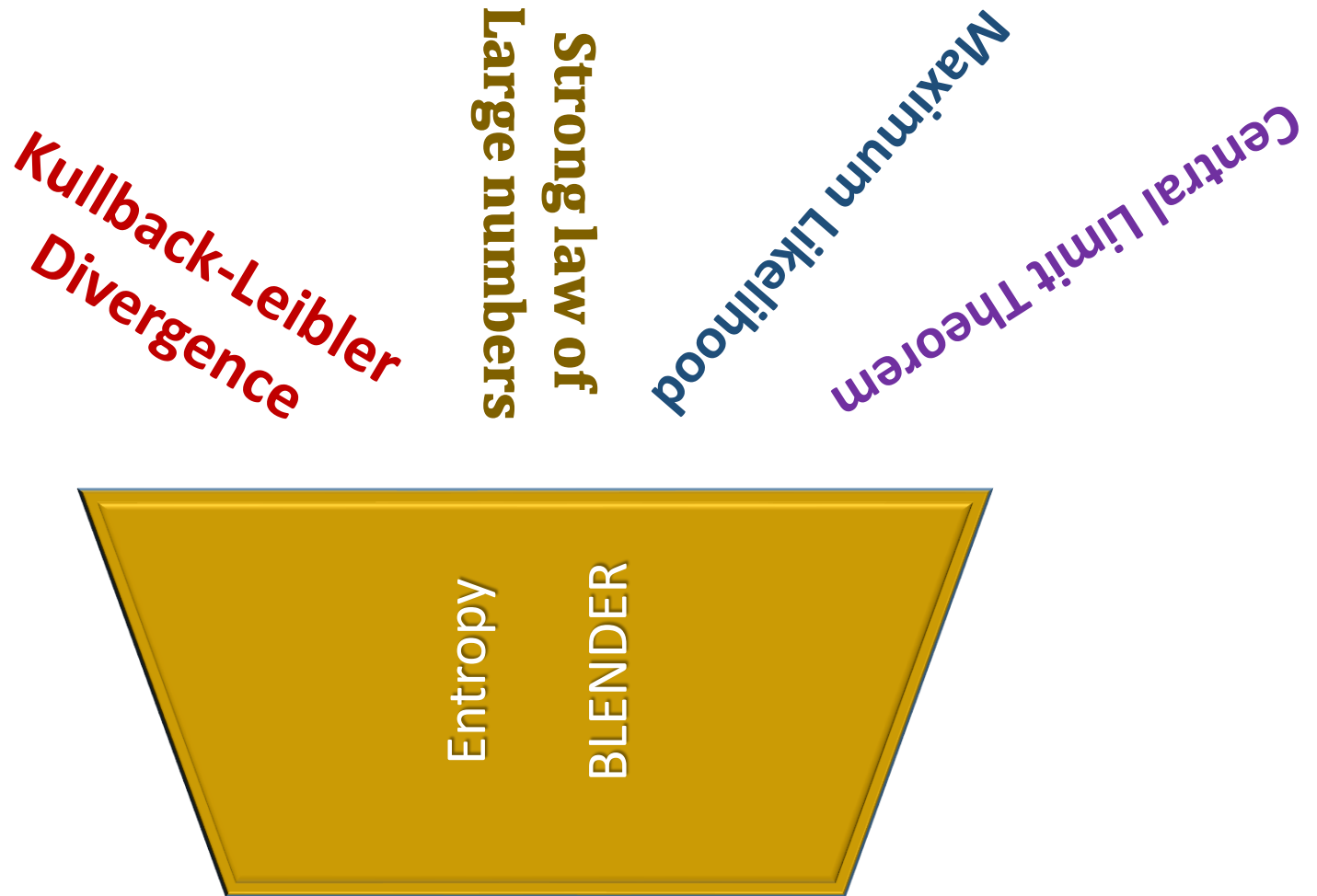
Cross Entropy by Maximum Likelihood but also

Confidence Interval for Cross Entropy!

Even we can have a good **Outliers Detection!**

CrossEntropy

$$\approx E_{\theta_0}(\log f_m(x)) = \sum_{i=1}^n f_m(x) \cdot \log f_m(x) \quad ; \quad x_i \sim N\left(\hat{\mu}_x, \frac{\hat{\sigma}^2}{n}\right)$$



We used these theorems:

- Kullback-Leibler Divergence
- Min K.L.D. = Max Cross Entropy = Max Likelihood
- Strong Law of Large Numbers
- Central Limit Theorem

K.L.D. = Kullback-Leibler
Divergence

We need these assumptions:

- Class of candidate models should be Absolutely **Continuous** and **Smooth** to have second order derivative in all points
- For using average, we should **remove outliers**.
- To know about dimensions ($m=M$), we try to guess m by looking at **Histogram**.

It is a very brief review of a personal concept, which shows how I usually attack the curve fitting or modeling problems. First, we must know linear algebra. Why?

Because computer engines work only by **COMPUTATIONAL LINEAR ALGEBRA**. It is essential to make a function linear and put machines to work. Software such as R and Python are using linear algebra packages such as LAPACK and BLAS. They load these linear algebra packages for their computation. Even SAS does so. This is why we prefer **LINEAR SPACE** in comparison with the **NORM SPACE**. Additionally, in linear space, we prefer **ADDITIVE MODELS**:

$$y \approx f(x_1) + f(x_2) + \dots + f(x_n),$$

why? Because simply they are easy to compute!

We can make multivariate analyses, Clustering, and Classification by using these additive univariate models without the need to reinvent the wheel.

Then, in linear space, we can use the tools of vector space, and we have angle θ as follows:

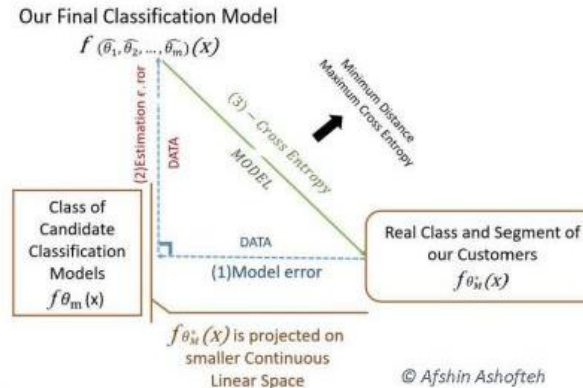
$$\hat{\theta} = \operatorname{argmax} \log L(\theta) = \operatorname{argmax} \log \prod_{i=1}^n f(x_i|\theta).$$

Why these tools and angles are important?

Because it gives us the ability to define the norm, vector, and direction.

For instance, we can define an inner product. $\|X\| \|Y\| \cos(\theta)$ and $\langle X, Y \rangle = \sum x_i y_i$, which is important to measure and minimize the errors and defining the **DISTANCE MEASURES**. In this vector space, we are able to fit an approximate line, surface, or high dimension solution to our data! Isn't that amazing? Most of the time we need our outcome function to be smooth because simply we have many tools in mathematics for optimizing **SMOOTH FUNCTIONS**! We will be able to take derivatives for optimization and use averages in our theories. Therefore, smooth functions have good behavior in converging to an appropriate outcome in the modeling stage. If we want to have a smooth function, then it is necessary to solve the problem of **MISSING VALUES**, which make our function discrete in the middle of the distribution, and **OUTLIERS**, which make our function discrete in extreme values. With a class of smooth functions and a huge amount of good data, we are able to find the best model and the best estimation of parameters, only by using the data.

This is the methodology to solve the modeling problems! First, we will define the problem, clean the data set, use the data to make a suggested class of smooth functions and finally use the computational power to choose the best model with the best estimations of parameters. ■



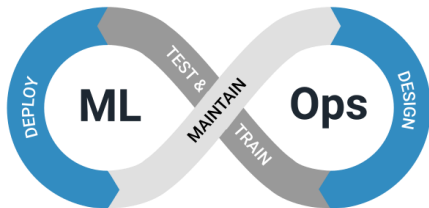
THE ERA OF TECHNOLOGY



databricks

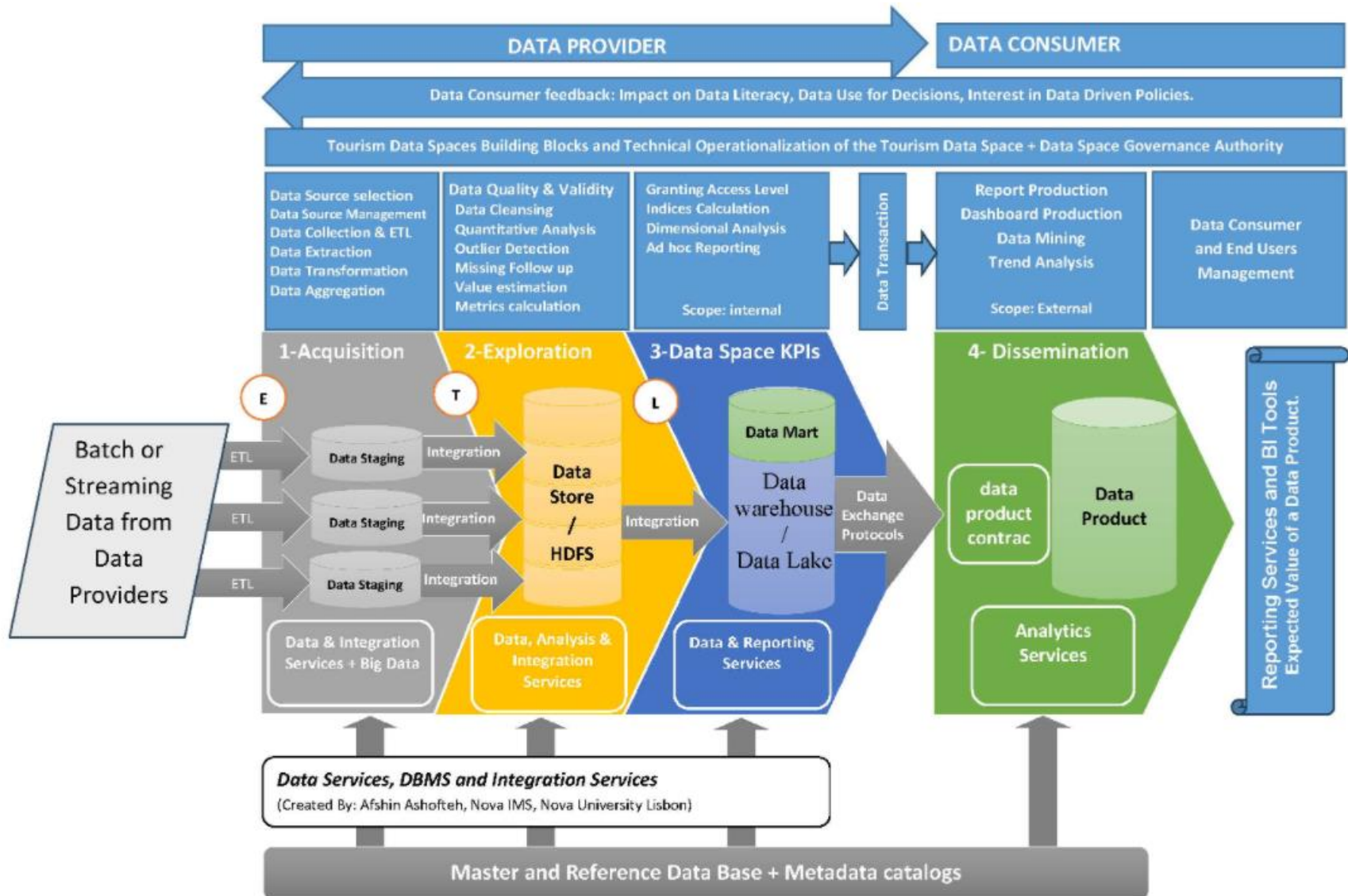


Power BI

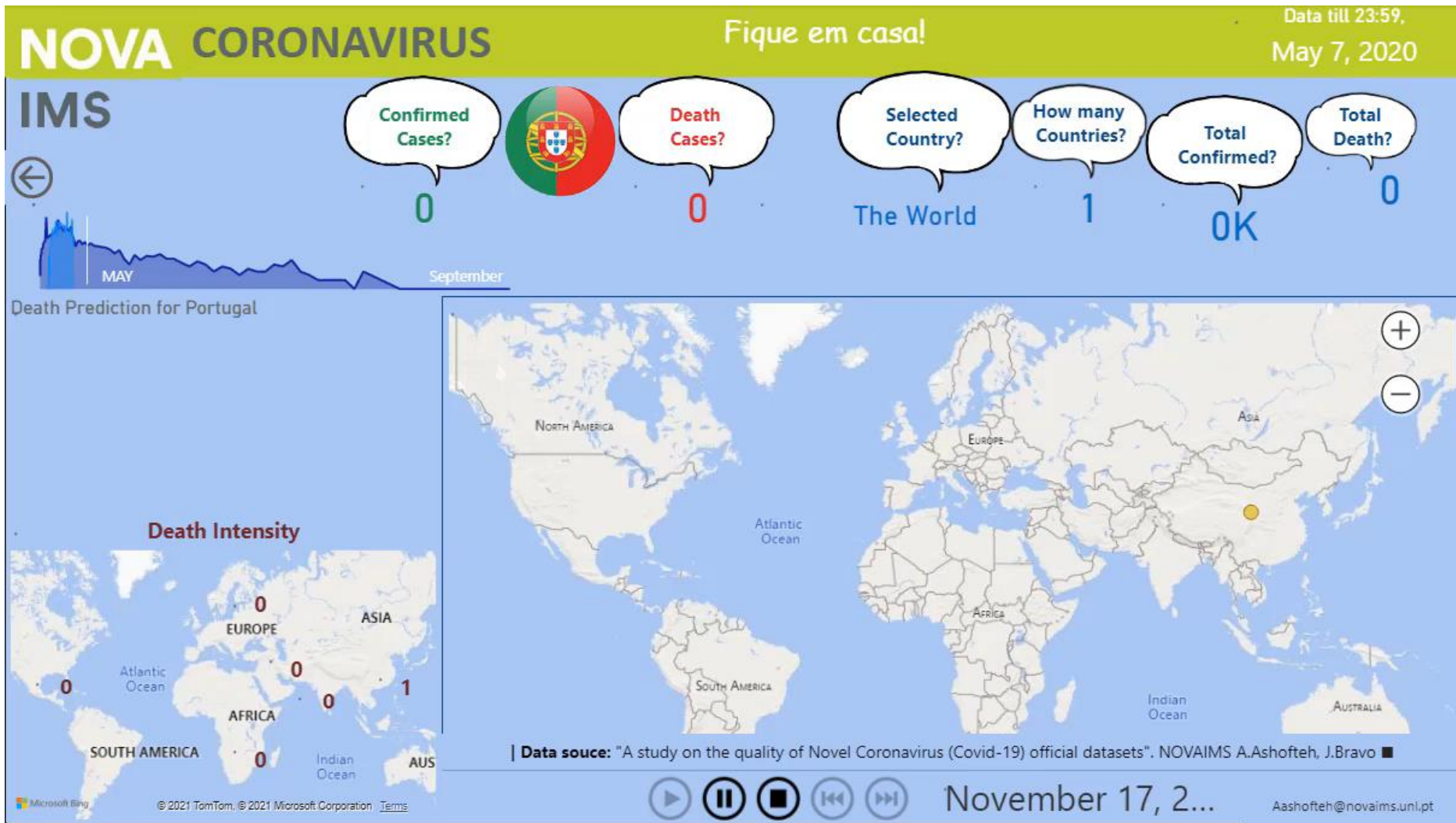


Paper1: <https://doi.org/10.1109/COMPSAC61105.2024.00101>

Paper2: <https://doi.org/10.1109/MITP.2025.3532129>



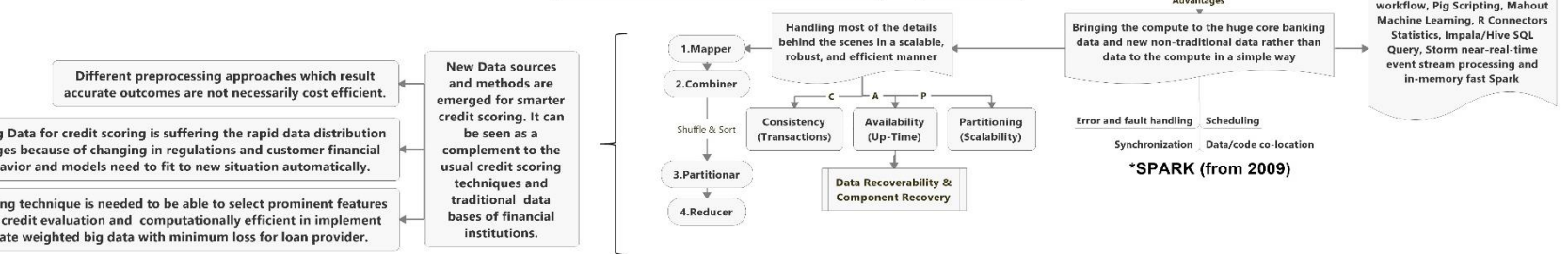
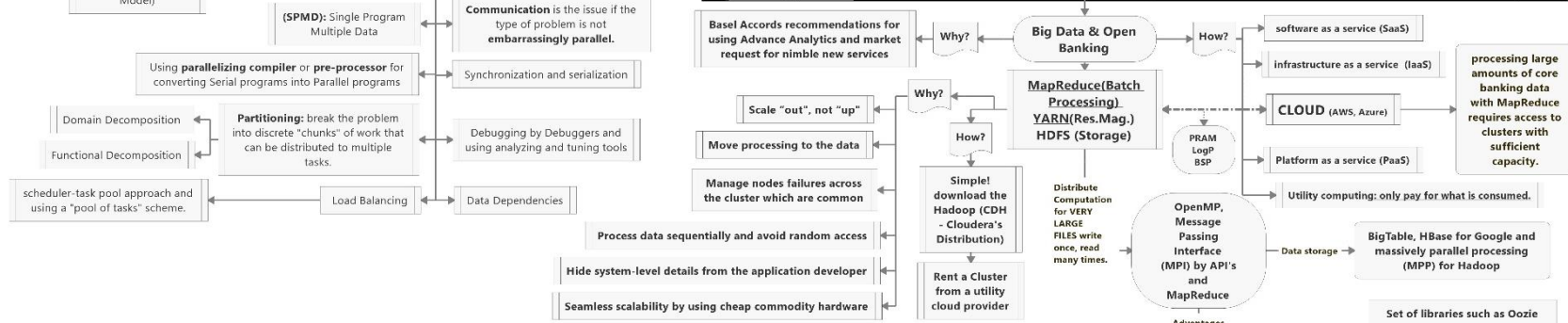
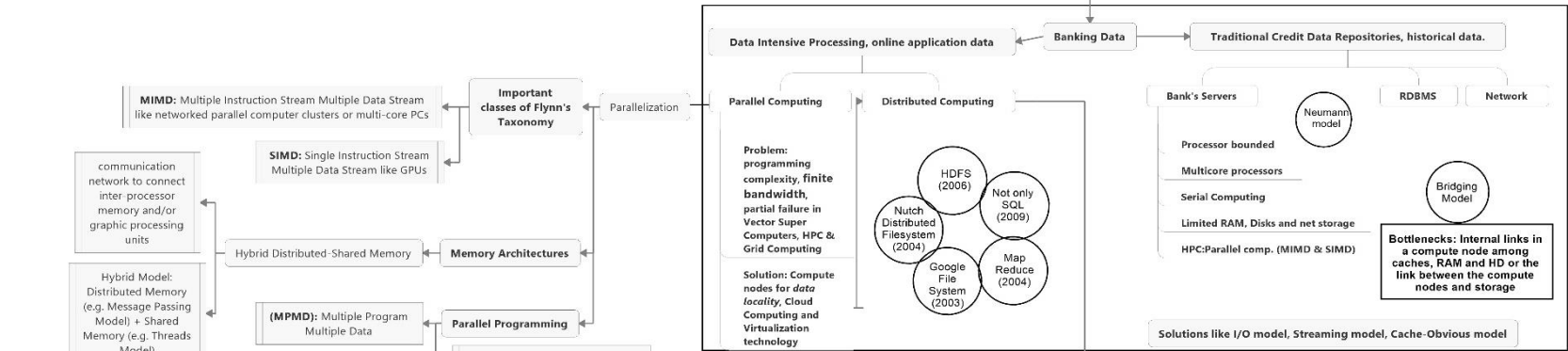
<https://doi.org/10.3233/SJI-200674>



THE ERA OF BIG DATA

Paper:
<https://doi.org/10.1016/j.eswa.2021.114835>

Video of presentation:
<https://youtu.be/tWu3HgZ9TtA>



Chapter in Statistical Modeling and Simulation for Experimental Design and Machine Learning Applications (SimStat 2019): https://doi.org/10.1007/978-3-031-40055-1_14



databricks



```
import sklearn.metrics as metrics
import pandas as pd
from plotnine import *
from plotnine.data import meat
from mizani.breaks import date_breaks
from mizani.formatters import date_format
from pyspark.ml import Pipeline
from pyspark.ml.feature import StandardScaler, StringIndexer, OneHotEncoder, Imputer, VectorAssembler
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
import mlflow
import mlflow.spark
from pyspark.mllib.evaluation import BinaryClassificationMetrics
from pyspark.ml.linalg import Vectors

# setting the parameters
maxIter = 10

## we start with mlflow.start_run() which essentially start tracking what we are doing in this notebook in databricks
with mlflow.start_run():
    labelCol = "default_loan"
    indexers = list(map(lambda c: StringIndexer(inputCol=c, outputCol=c+"_idx", handleInvalid = "keep"), categoricals))
    ohes = list(map(lambda c: OneHotEncoder(inputCol=c + "_idx", outputCol=c+"_class"), categoricals))
    imputers = Imputer(inputCols = numerics, outputCols = numerics)
    featureCols = list(map(lambda c: c+"_class", categoricals)) + numerics
    model_matrix_stages = indexers + ohes + \
        [imputers] + \
        [VectorAssembler(inputCols=featureCols, outputCol="features"), \
         StringIndexer(inputCol=labelCol, outputCol="label")]

    scaler = StandardScaler(inputCol="features",
                            outputCol="scaledFeatures",
                            withStd=True,
                            withMean=True)
```

codeocean.com/capsule/0503126/tree/v1

Published Spark Code: A Novel Conservative Approach for Online Credit Scoring (Afshin Ashofteh & Jorge Bravo)

Core Files

- metadata 752 B
- environment 1.41 KB
- code 6.99 MB
 - AA_DFW_ALL.parquet 6.66 MB
 - mlruns 256.42 KB
 - 4-paper-phi+CRI train and t... 73.44 KB
 - environment.yml 4.93 KB
 - LICENSE 1.04 KB
 - README.md 1.57 KB
 - requirements.txt 6.44 KB
 - run 353 B
- data Manage Datasets 263.17 MB
 - LICENSE 6.4 KB
 - paper_train1.csv 201.57 MB
 - paper_valid1.csv 61.59 MB
 - .gitignore 7 B
- results 1019.18 KB

A Conservative Approach for Online Credit Scoring

Journal: Expert Systems with Application- 2021

Afshin Ashofteh, Jorge Bravo

Table of contents

- [General info](#)
- [Data Source](#)
- [Technologies](#)
- [Contact](#)

General info

This capsule is related to a novel method of machine learning for Big Data, which is discussed in a manuscript in Expert Systems with Applications. It is appropriate for the default prediction of high-risk branches or customers and online banking. This study uses the Kruskal-Wallis non-parametric statistic to form a conservative credit-scoring model and to study the impact of modeling performance on the benefit of the credit provider. This is the first study that develops an online non-parametric credit scoring system, which is able to reselect effective features automatically for continued credit evaluation and weigh them out by their level of contribution with a good diagnostic ability. We have implemented this new methodology on Ridge, Lasso, Elastic-net Regressions, Random forest classifier, and Linear support vector machine.

Data source

loan.csv - data is from Lending Club includes all funded loans from 2012 to 2017. Each loan includes applicant information provided by the applicant as well as current loan status (Current, Late, Fully Paid, etc.) and latest payment information.

Technologies

Project is created with:

- Python version: 3.8.1
- PySpark version: 3.0.2

Reproducible Run

or launch a cloud workstation

Timeline

- Feb 28, 2021
Published Version 1.0
Currently viewing
- Author ran Feb 28, 2021 00:58:40
 - Published Result
 - 4-paper-phi+CRI tr... 737.58 KB
 - output 265.04 KB
- Afshin Ashofteh committed Feb 28, 2021
- Version 1.0
- Feb 28, 2021
Created capsule

Afshin Ashofteh, Jorge Bravo (2021) Spark Code: A Novel Conservative Approach for Online Credit Scoring [Source Code]. <https://doi.org/10.24433/CO.1963899.v1>

THE ERA OF COMPLEX MODELS

Ensemble models

DELMS for Time Series Modeling

Dynamic Ensemble Learning with an Intelligent Model Selection Strategy

- We use direct mapping from the time series of different countries, combining the intractable time series algorithms and predicting the ensemble model as the final output.

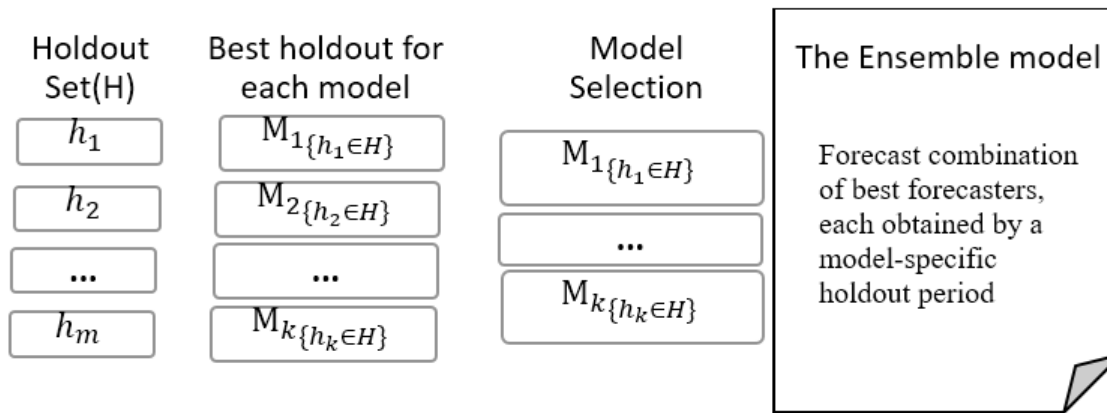


Figure 1 - Proposed strategy of ensemble learning.

The marginal posterior distribution across all models is

$$p(\Delta|y) = \sum_{k=1}^K p(\Delta|y, M_k)p(M_k|y)$$

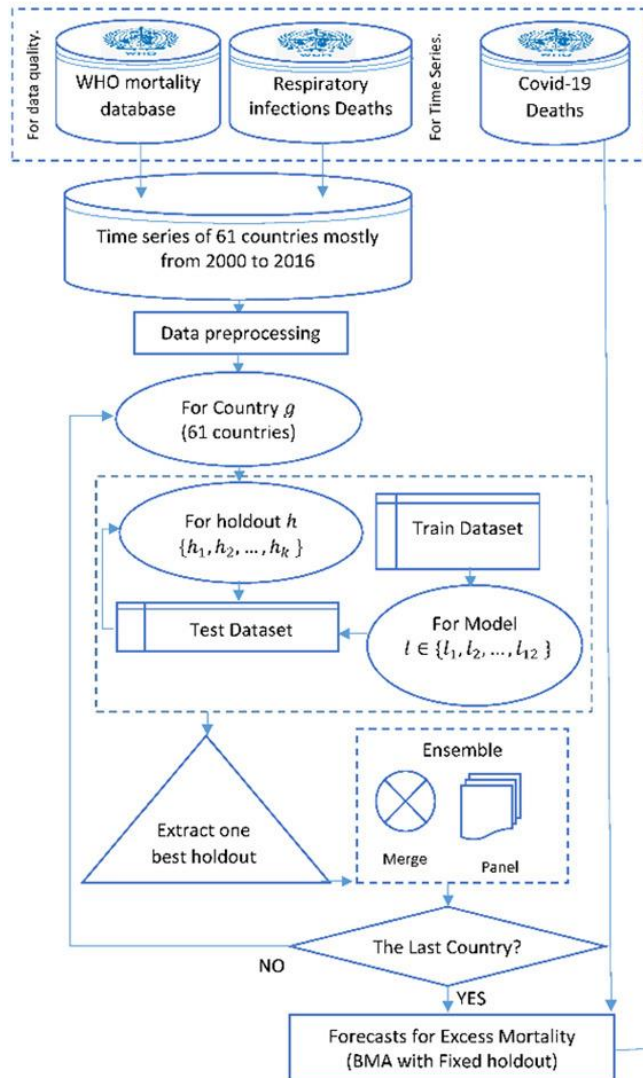
The weight assigned to each model M_k is given by its posterior probability

$$p(M_k|y) = \frac{p(y|M_k)p(M_k)}{\sum_{l=1}^K p(y|M_l)p(M_l)}$$

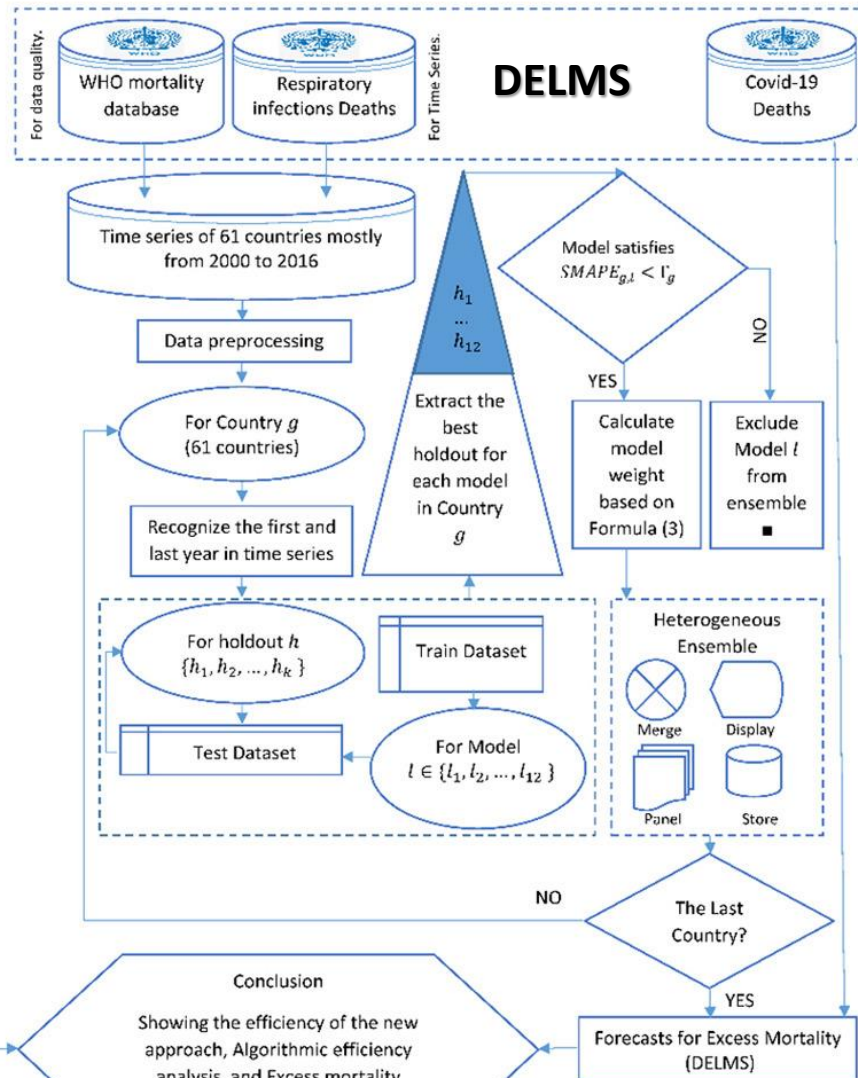
Publication: <https://doi.org/10.1016/j.asoc.2022.109422>

Video of presentation: <https://youtu.be/7xabpWH4aFs>

(1) Ensemble with fix holdout



(2) Ensemble with Dynamic holdouts and model selection (DELMS)



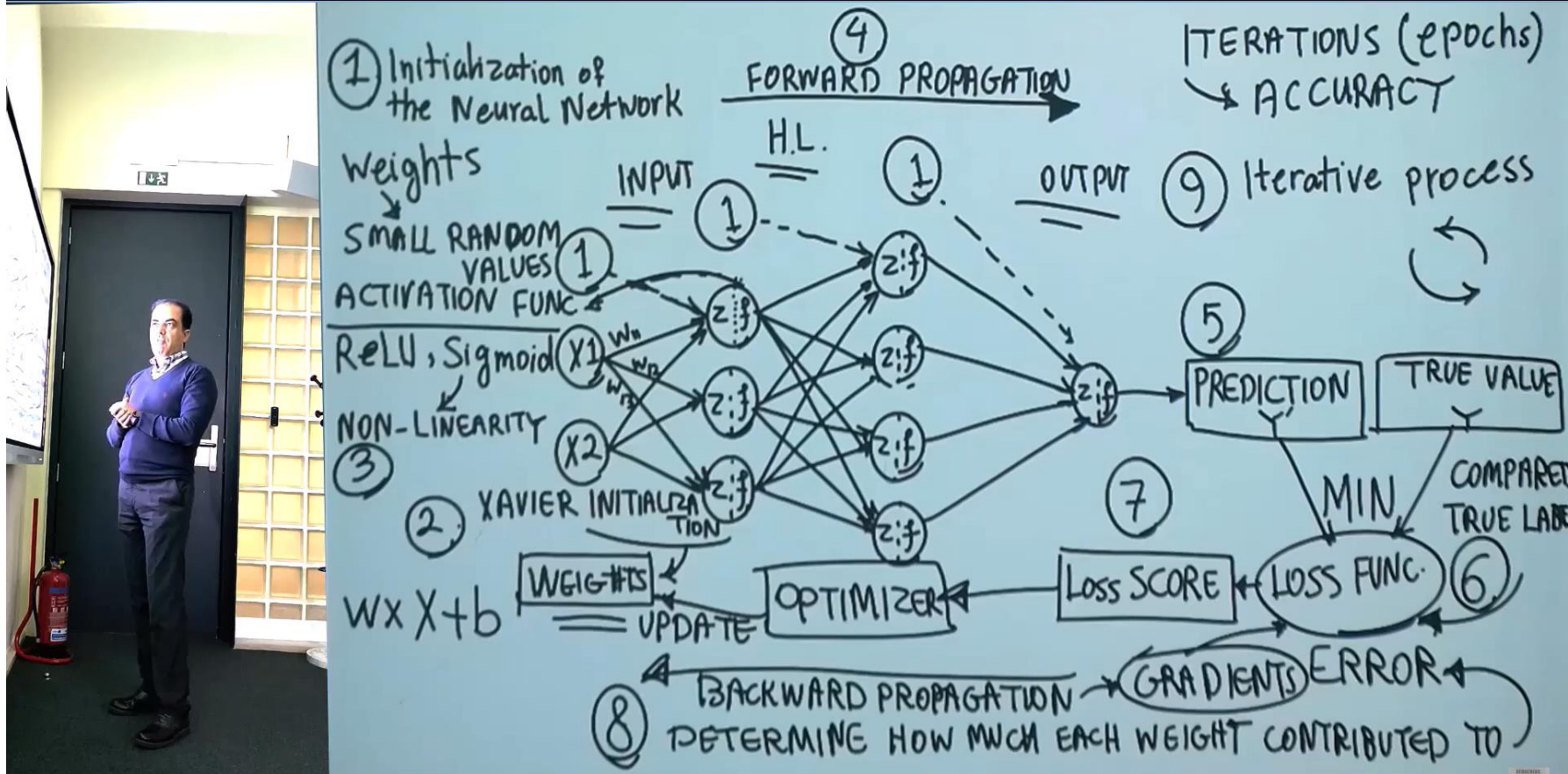
Conclusion
 Showing the efficiency of the new approach, Algorithmic efficiency analysis, and Excess mortality analysis. ■

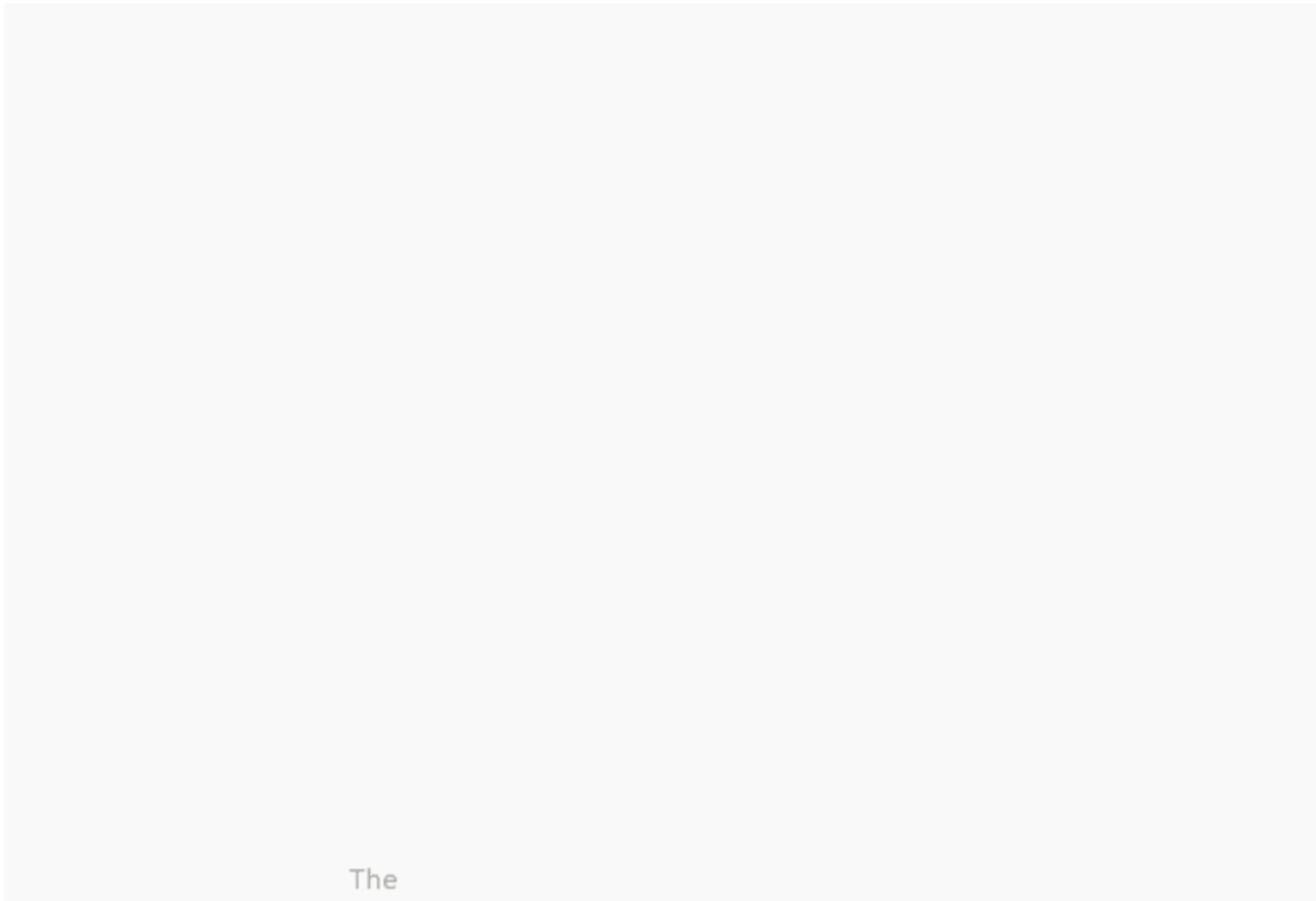
THE ERA OF DEEP LEARNING

Large Language Models (LLMs)

<https://youtu.be/vLxgDQmM57Y>

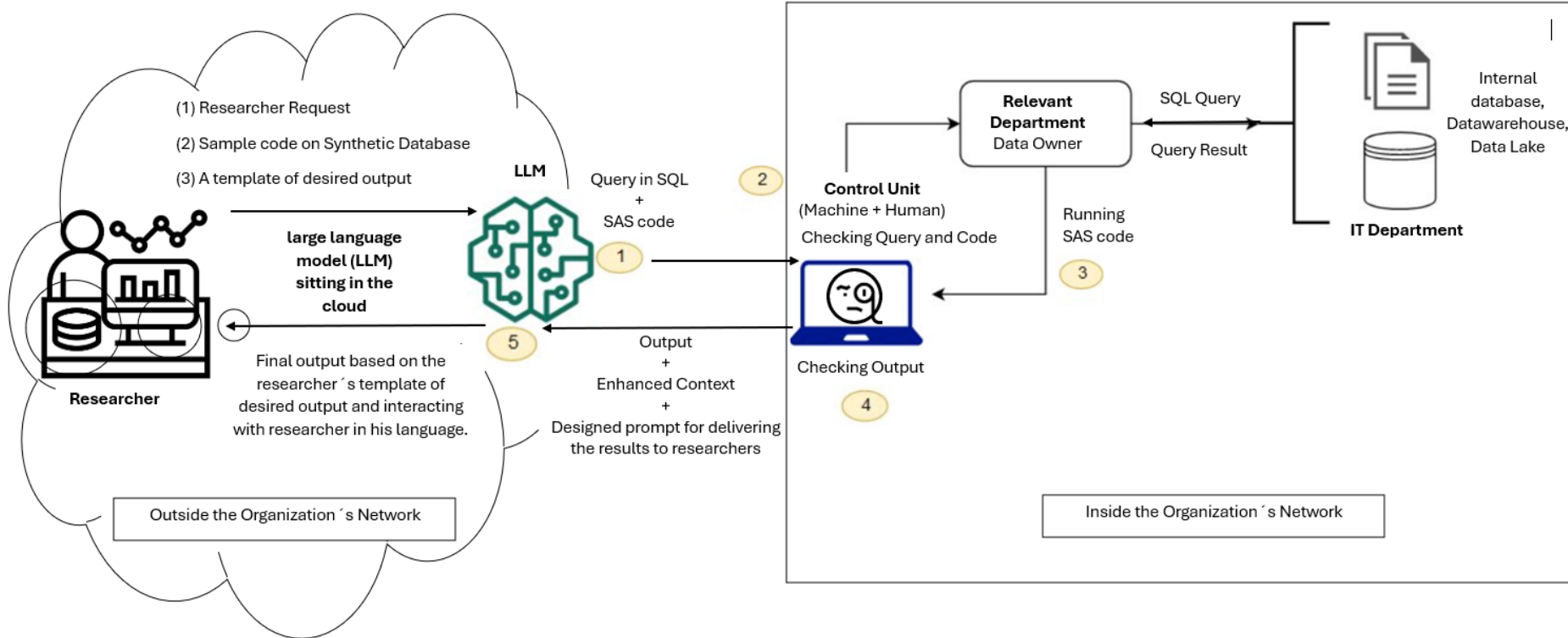
DEEP LEARNING - FORWARD & BACKWARD PROPAGATION





The

Saferoom LLM Agent



"SMARTTEST AI"



Data Science Discussion Group

<https://www.linkedin.com/groups/12420006/>



Data Science YouTube

<https://www.youtube.com/channel/UCTOuxIhJxcxNOntTpamJeAA>

<https://novaresearch.unl.pt/en/publications/a-conservative-approach-for-online-credit-scoring-2>

<https://novaresearch.unl.pt/en/publications/an-ensemble-learning-strategy-for-panel-time-series-forecasting-o>

E-mail | Academic HP | ResearchGate

THANK YOU!

Afshin Ashofteh; PhD, PGDip, MBA, MSc

[E-mail](#) | [Academic HP](#) | [Discussion Group](#) | [YouTube](#)

Address: Campus de Campolide, 1070-312 Lisboa, Portugal

Acreditações e Certificações



UNIGIS



A3ES



iSchools



Computing Accreditation Commission



Instituto Superior de Estatística e Gestão da Informação
Universidade Nova de Lisboa