

Aprendizaje Automático

I. Introducción / Probabilidad

Juan F. Döppler

UNSAM - 18 de septiembre de 2023

Contenidos de la materia

- Introducción al Análisis de Datos.
 - Probabilidad y Estadística (y algo de Bayes)
 - Análisis exploratorio de datos
 - Modelos de un solo parámetro
 - Regresión Lineal
- Machine Learning “clásico”.
 - Modelos lineales para regresión y clasificación.
 - Support Vector Machine
 - Árboles de Decisión
 - Métodos de Ensemble (RandomForests, ...)
 - Otros algoritmos
- Redes Neuronales y Aprendizaje Profundo
 - Multi-Layer Perceptron,
 - Convolutional Neural Networks.
 - Otra arquitecturas (RNN, AE, GAN, ...).

Presentaciones!

Un cronograma aproximado

Semana	Clase	Fecha	Tema	Notebook
1	1	Lu 18/09/2023	Introducción a la materia - Probabilidad	01_Probabilidad.ipynb
	2	Ju 21/09/2023	AED y preprocesamiento	
2	3	Lu 25/09/2023	Modelos lineales	
	4	Ju 28/09/2023	Underfitting, overfitting, cross validation	
3	5	Lu 02/10/2023	Regularización, Bias variance	
	6	Ju 05/10/2023	Clasificación, regresión logística	
4	7	Lu 09/10/2023	Métricas	
	8	Ju 12/10/2023	SVM	
5		Lu 16/10/2023	FERIADO	
	9	Ju 19/10/2023	Árboles de decisión	
6	10	Lu 23/10/2023	Métodos de ensemble	
	11	Ju 26/10/2023	Random forest	
7	12	Lu 30/10/2023	Boosting	
	13	Ju 02/11/2023	Aprendizaje no supervisado	
8	14	Lu 06/11/2023	Redes neuronales 1	
	15	Ju 09/11/2023	Redes neuronales 2	
9	16	Lu 13/11/2023	AE y VAE	
	17	Ju 16/11/2023	GAN	
10		Lu 20/11/2023	FERIADO	
	18	Ju 23/11/2023	CNNs	
11	19	Lu 27/11/2023	Transfer learning	
	20	Ju 30/11/2023	RNNs	
12	21	Lu 04/12/2023	Parcial	
	22	Ju 07/12/2023	Recuperatorio	

Padrinos



**Escuela de
Ciencia y Tecnología**
ECyT_UNSAM



Inteligencia Artificial Interdisciplinaria
VICERRECTORADO

"Sponsors"



Machine
Learning



Code hosting



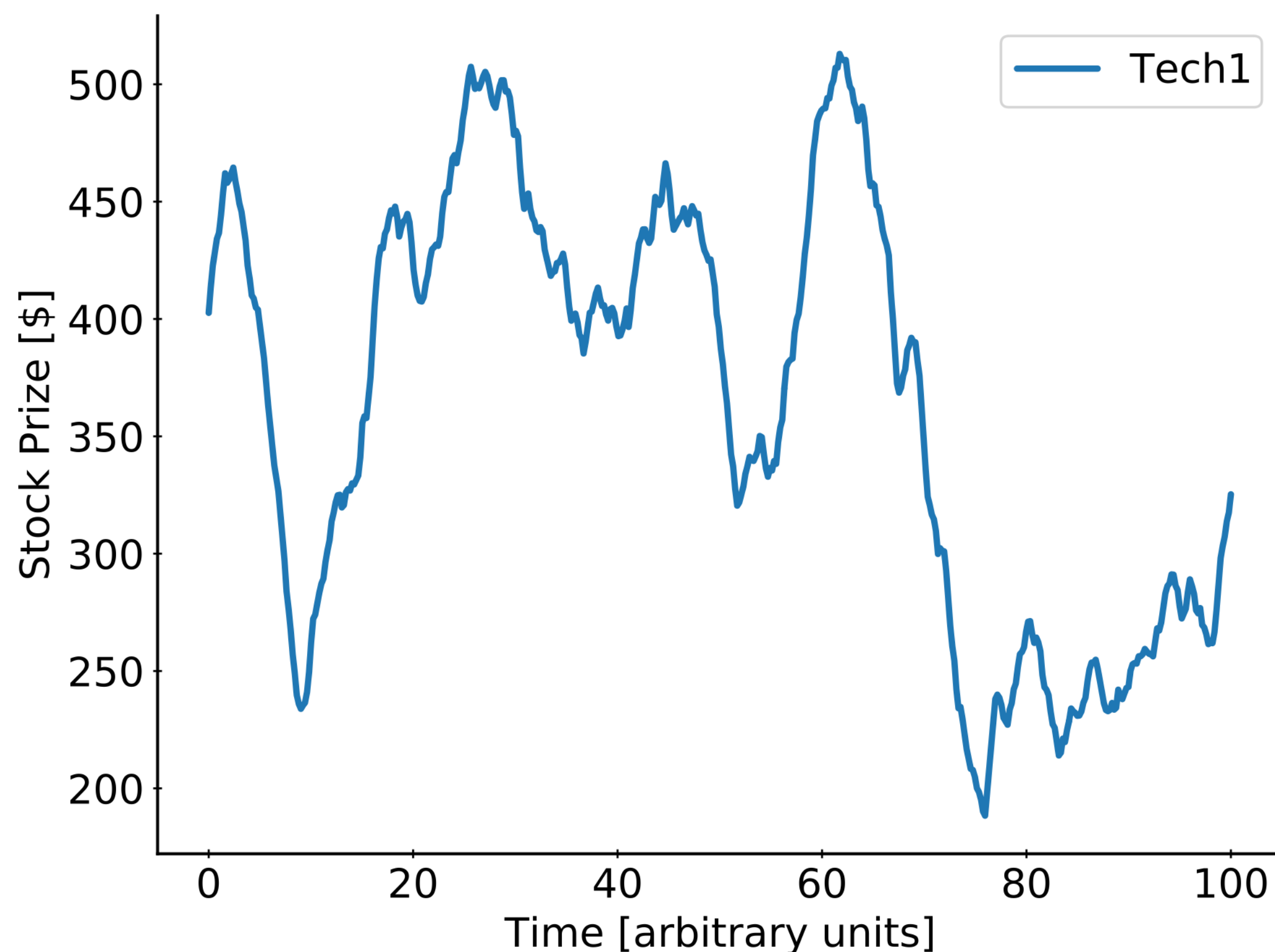
GitHub



Running
environment

Predicciones a partir de datos

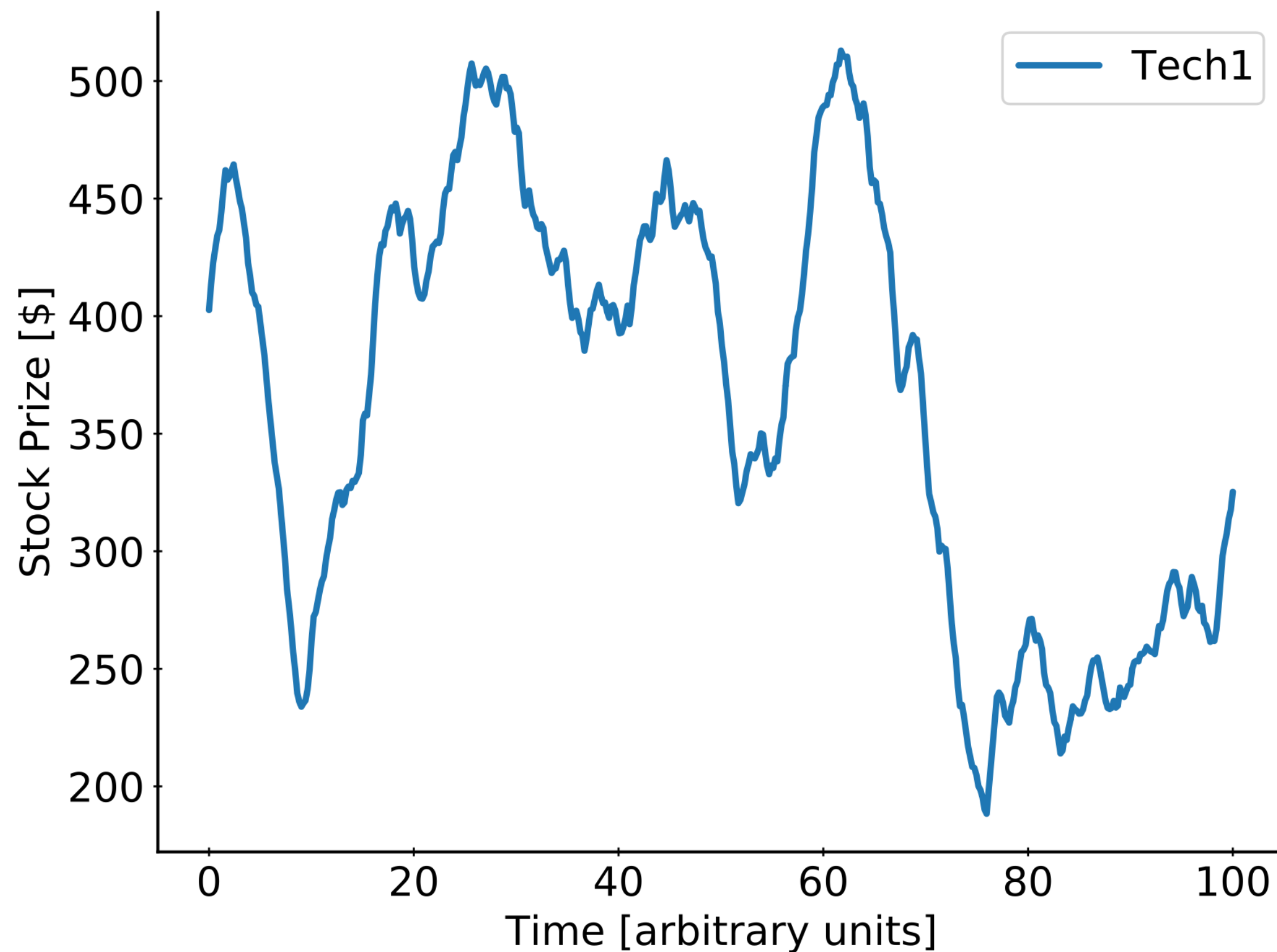
Un problema de juguete



- Basado en los valores observados de la acción de una compañía a lo largo del tiempo, ¿podemos predecir si el precio va a subir o bajar?
- ¿Y si hubiera otros “observables” como el precio de otras compañías?
- ¿Cómo hacemos eso?

Predicciones a partir de datos

Un problema de juguete



- Primero, hay que decidir si queremos:

1. predecir el *valor* de la acción, para determinar si el precio va a subir o bajar.

O

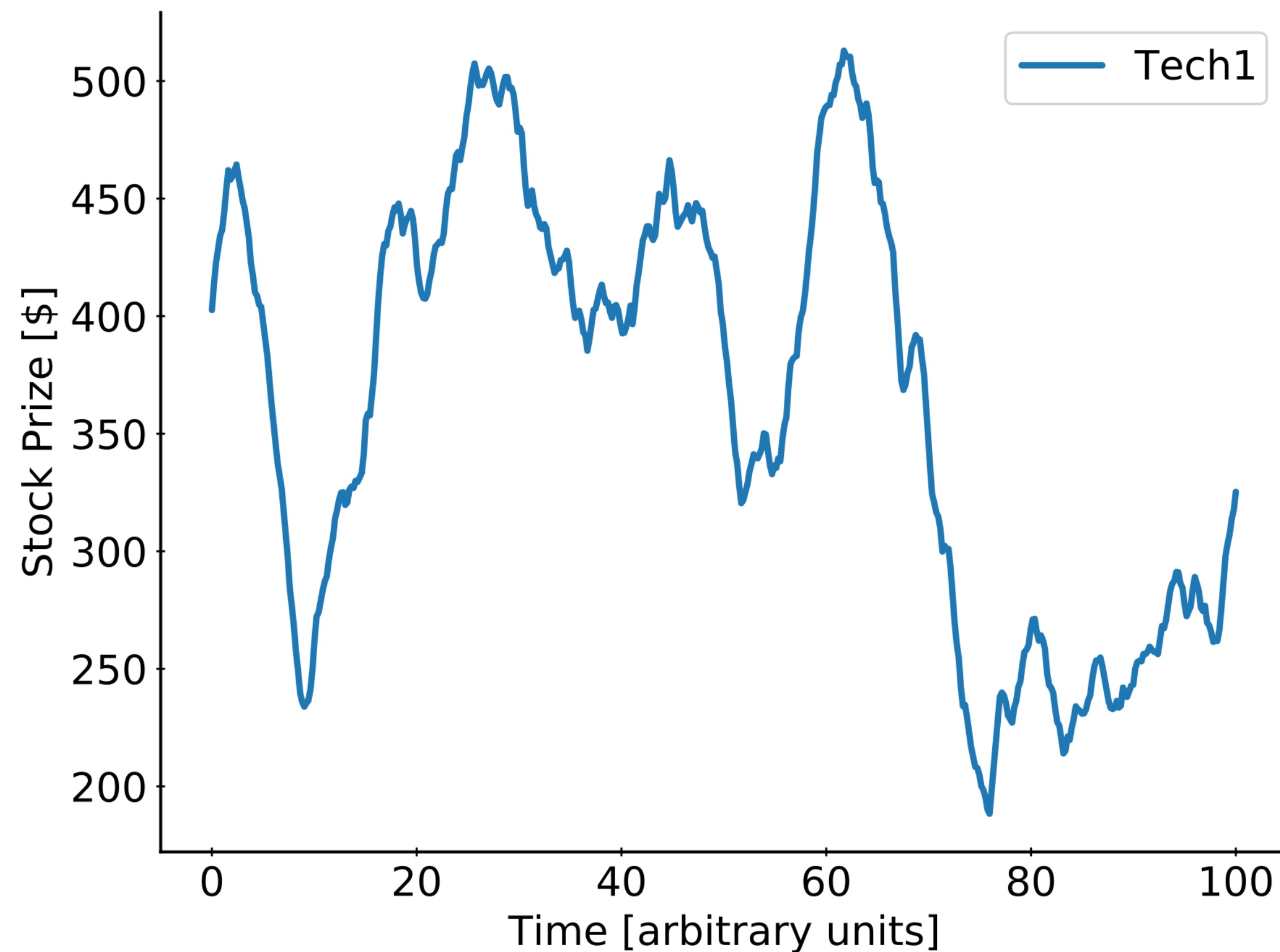
2. predecir directamente si la acción va a subir o bajar.

Regresión

Clasificación

Predicciones a partir de datos

Un problema de clasificación de juguete - dos abordajes posibles



- Imaginemos que le damos el problema a dos personas: Alice, la experta en ciencia de datos; y Bob, el gurú de machine learning.
- ¿Cómo se aproxima cada uno al problema?



Alice

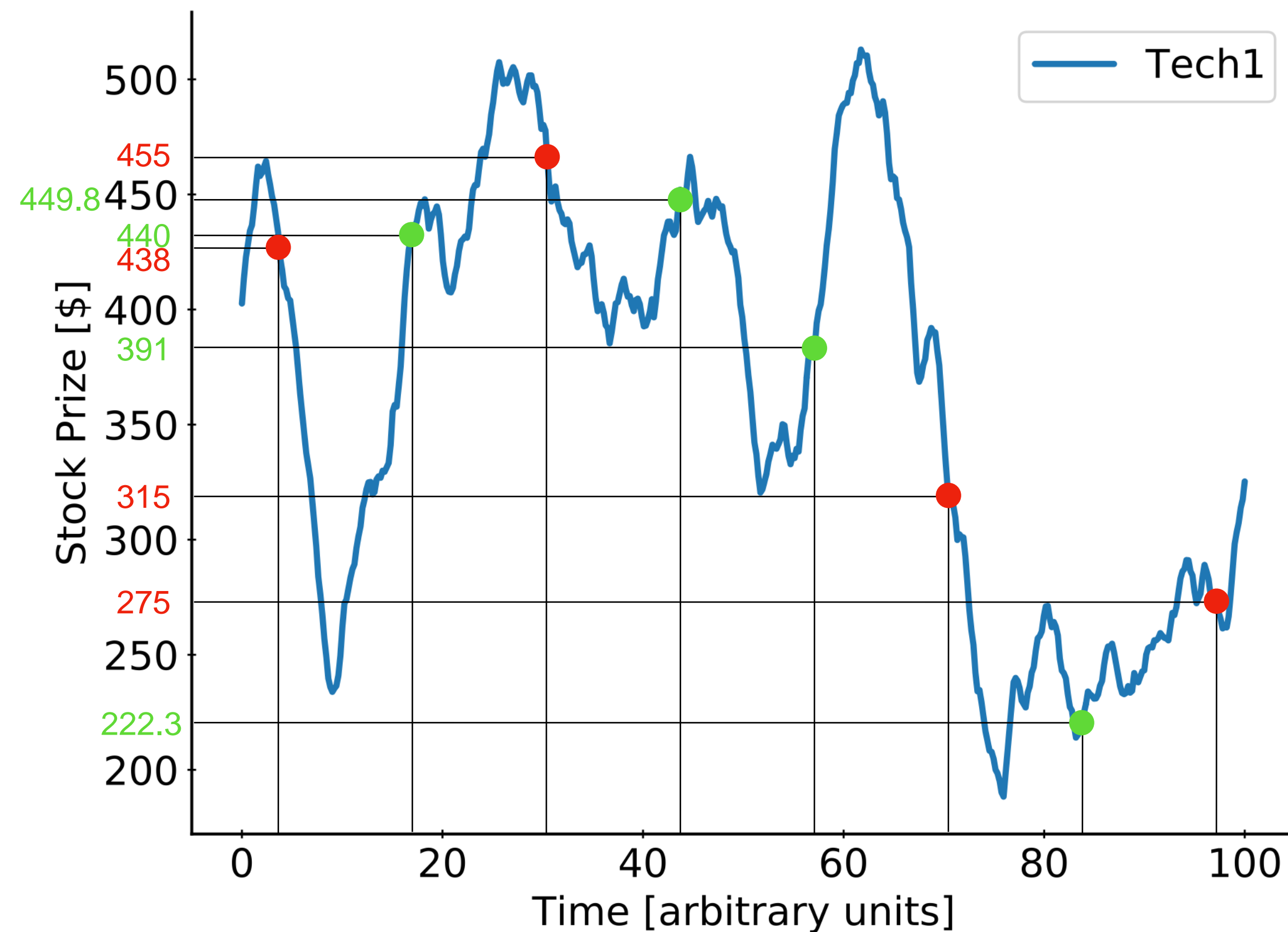


Bob

```
import numpy as np
import pandas as pd
... # other imports
import sklearn
```

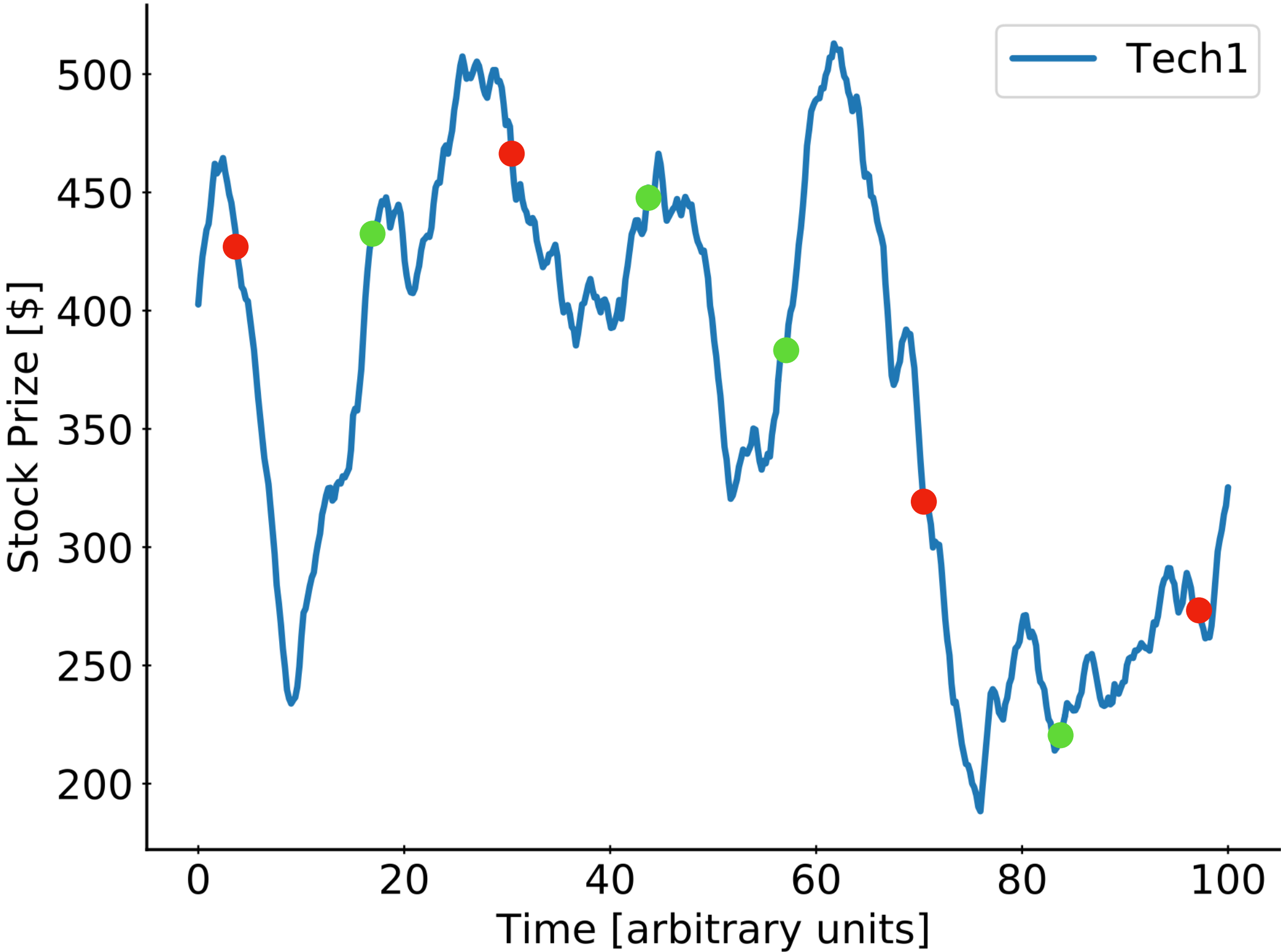
Predicciones a partir de datos

Un problema de clasificación de juguete - *feature* extraction



Predicciones a partir de datos

Un problema de clasificación de juguete - *feature* extraction



x_I : Feature

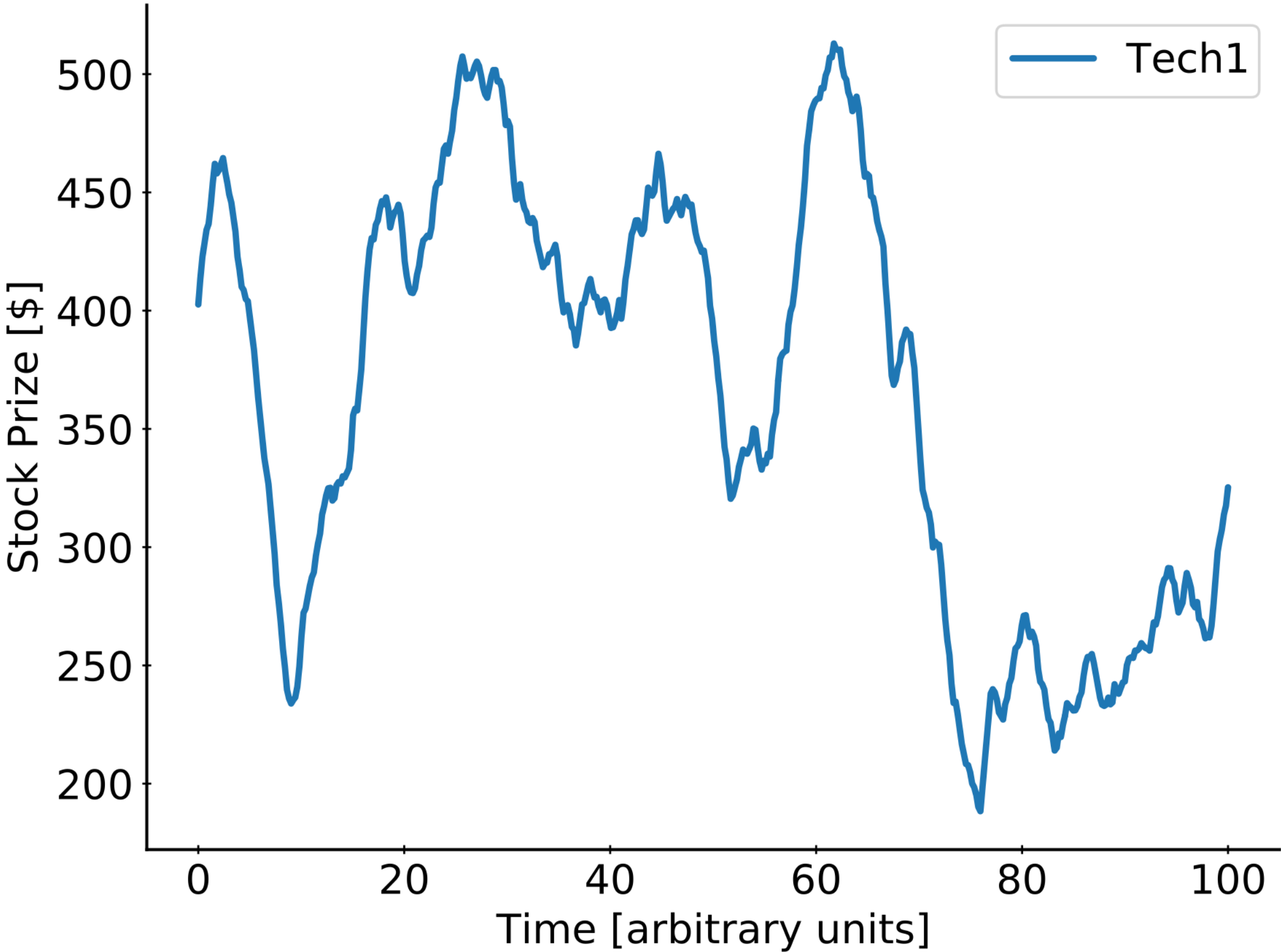
x_I	t(target)
455	0
449.8	1
440	1
438	0
391	1
315	0
275	0
222.3	1

Labels /
Etiquetas



Predicciones a partir de datos

Un problema de clasificación de juguete - *feature* extraction

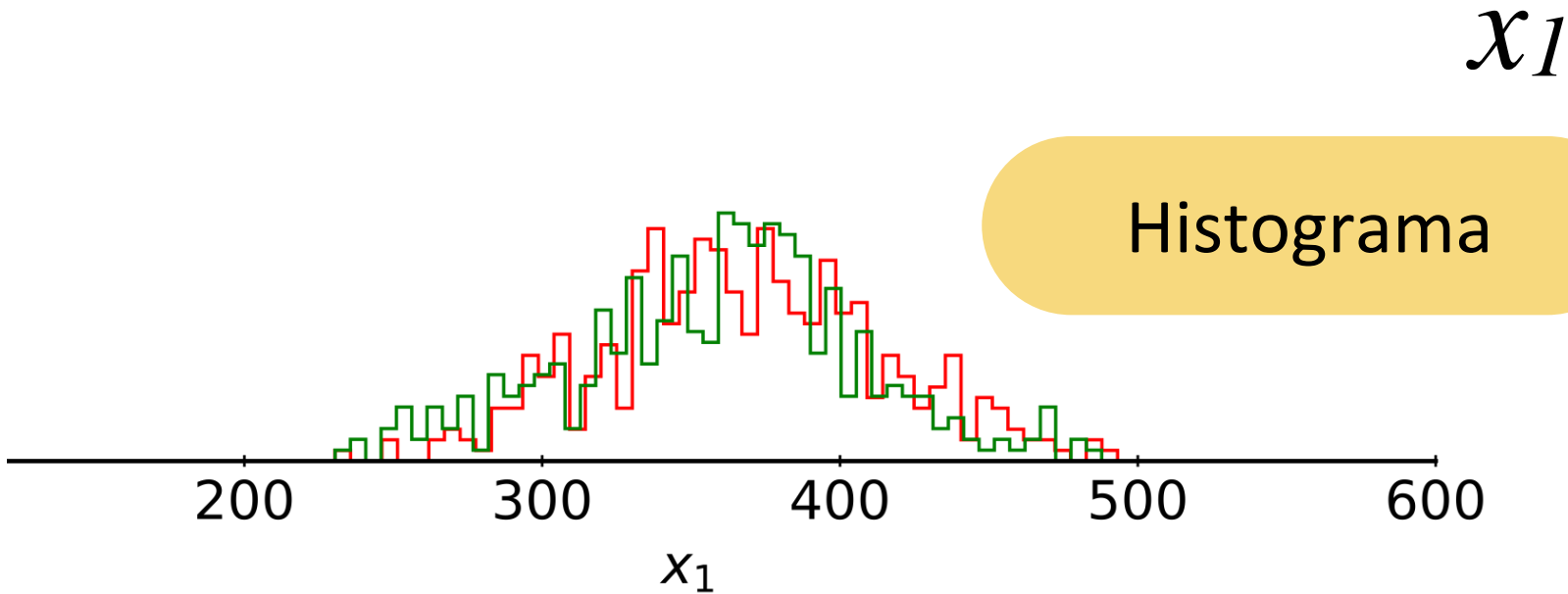


x_1 : Feature

x_1	t(target)
455	0
449.8	1
440	1
438	0
391	1
315	0
275	0
222.3	1



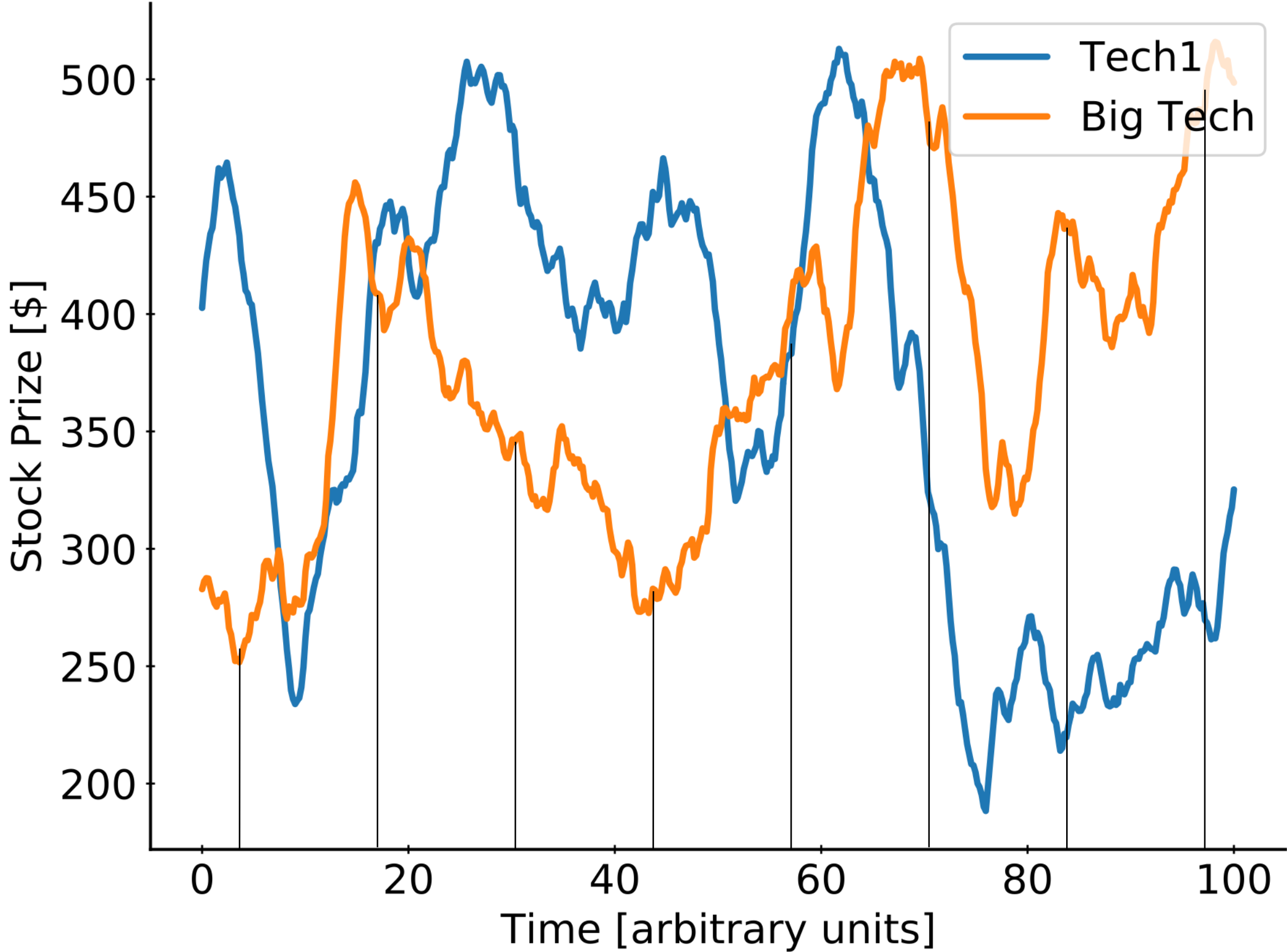
Labels /
Etiquetas



Predicciones a partir de datos

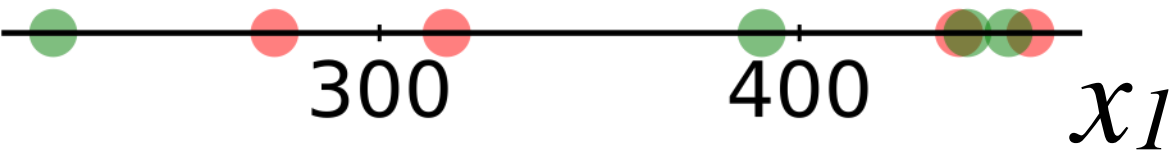
Un problema de clasificación de juguete - *feature selection*

x_1 : Feature



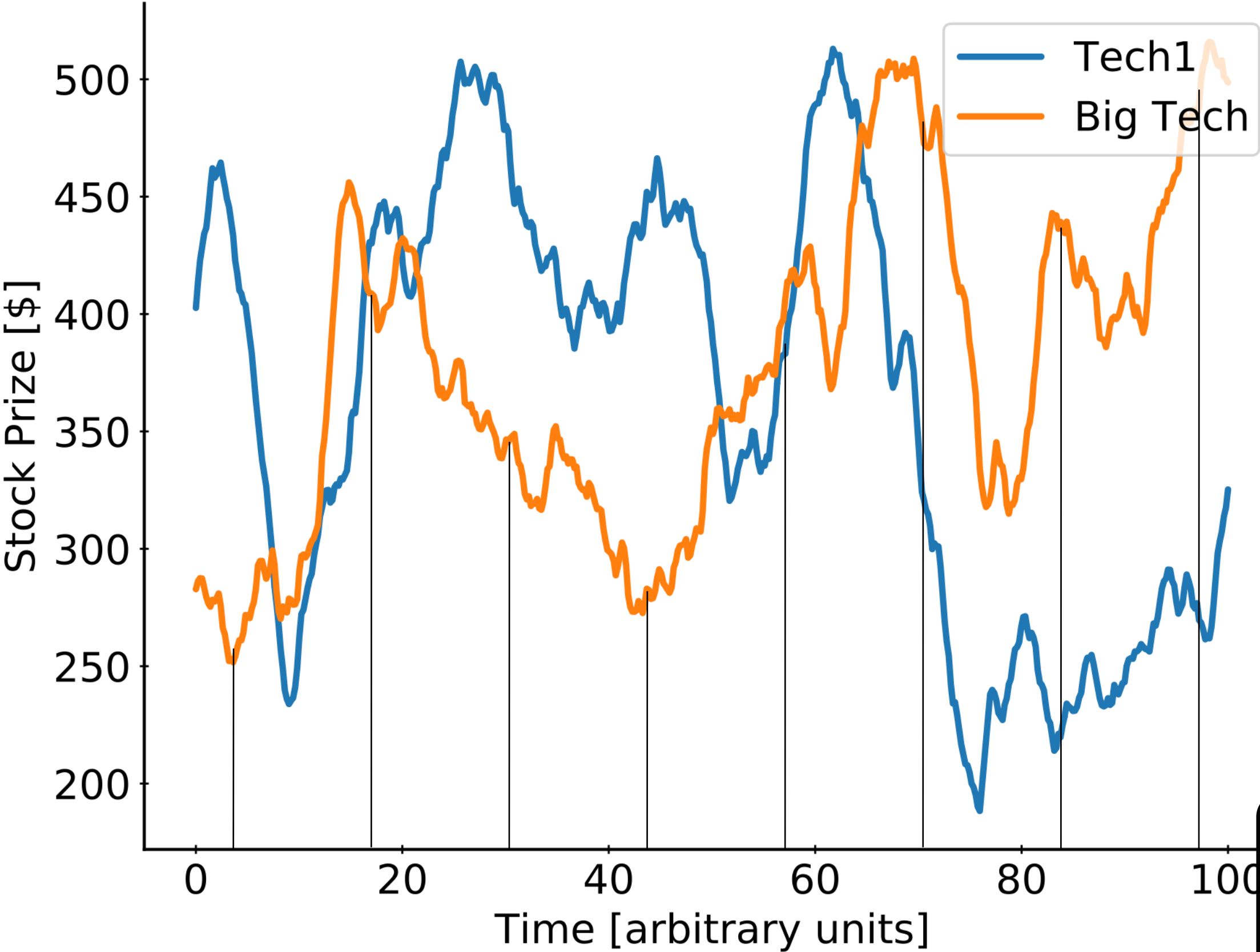
x_1	t(target)
455	0
449.8	1
440	1
438	0
391	1
315	0
275	0
222.3	1

Labels /
Etiquetas



Predicciones a partir de datos

Un problema de clasificación de juguete - *feature selection*



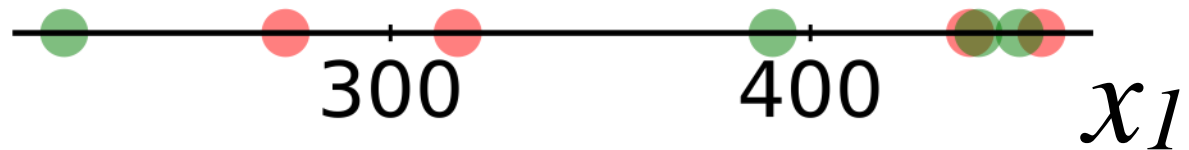
Features

x_1	x_2	t(target)
455	260	0
449.8	425	1
440	320	1
438	280	0
391	387	1
315	470	0
275	440	0
222.3	500	1



Labels

(referidas a Tech1)



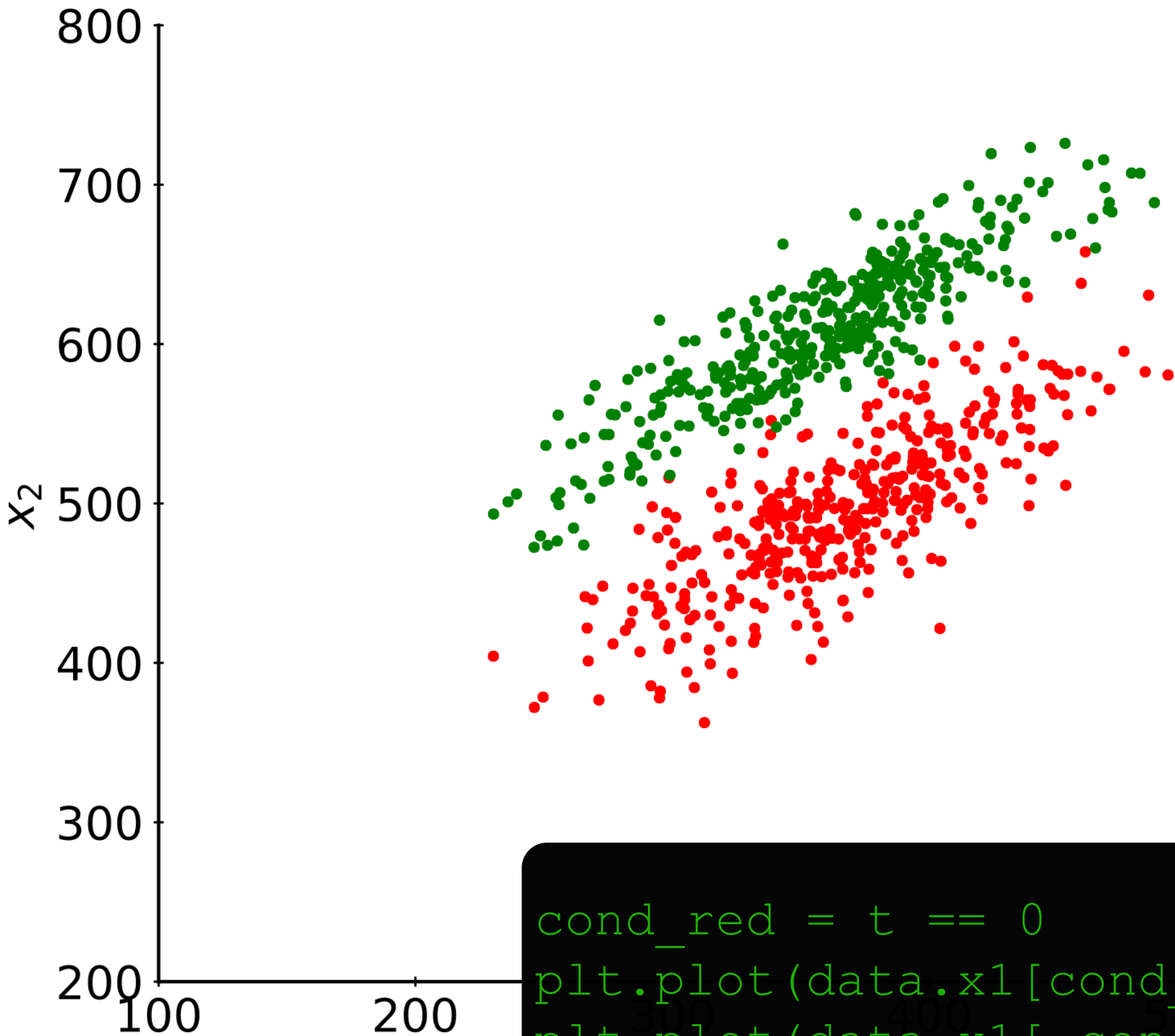
```
datafull = pd.read_csv('tech1_features.csv')
data = datafull.drop('target', inplace=False)
t = datafull.target
```

Predicciones a partir de datos

Un problema de clasificación de juguete - *features* multidimensionales

Features

x_1	x_2	t(arget)
455	260	0
449.8	425	1
440	320	1
438	280	0
391	387	1
315	470	0
275	440	0
222.3	500	1
....



```
cond_red = t == 0
plt.plot(data.x1[cond_red], data.x2[cond_red], '.r')
plt.plot(data.x1[~cond_red], data.x2[~cond_red], '.g')
```



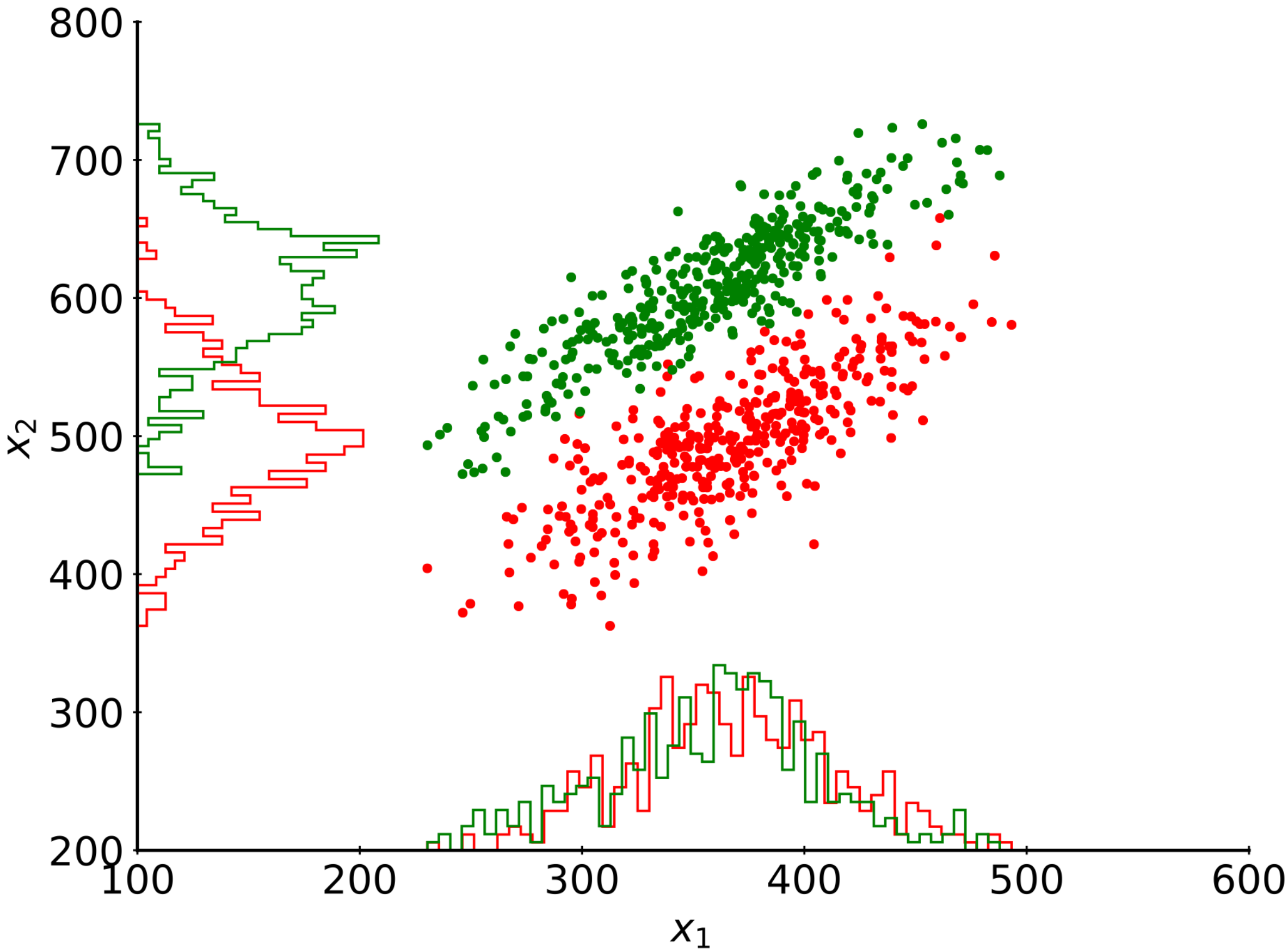
Usando un set de features extendido, el problema se vuelve mucho más fácil

Predicciones a partir de datos

Un problema de clasificación de juguete - *features* multidimensionales

Features

x_1	x_2	t(target)
455	260	0
449.8	425	1
440	320	1
438	280	0
391	387	1
315	470	0
275	440	0
222.3	500	1
....

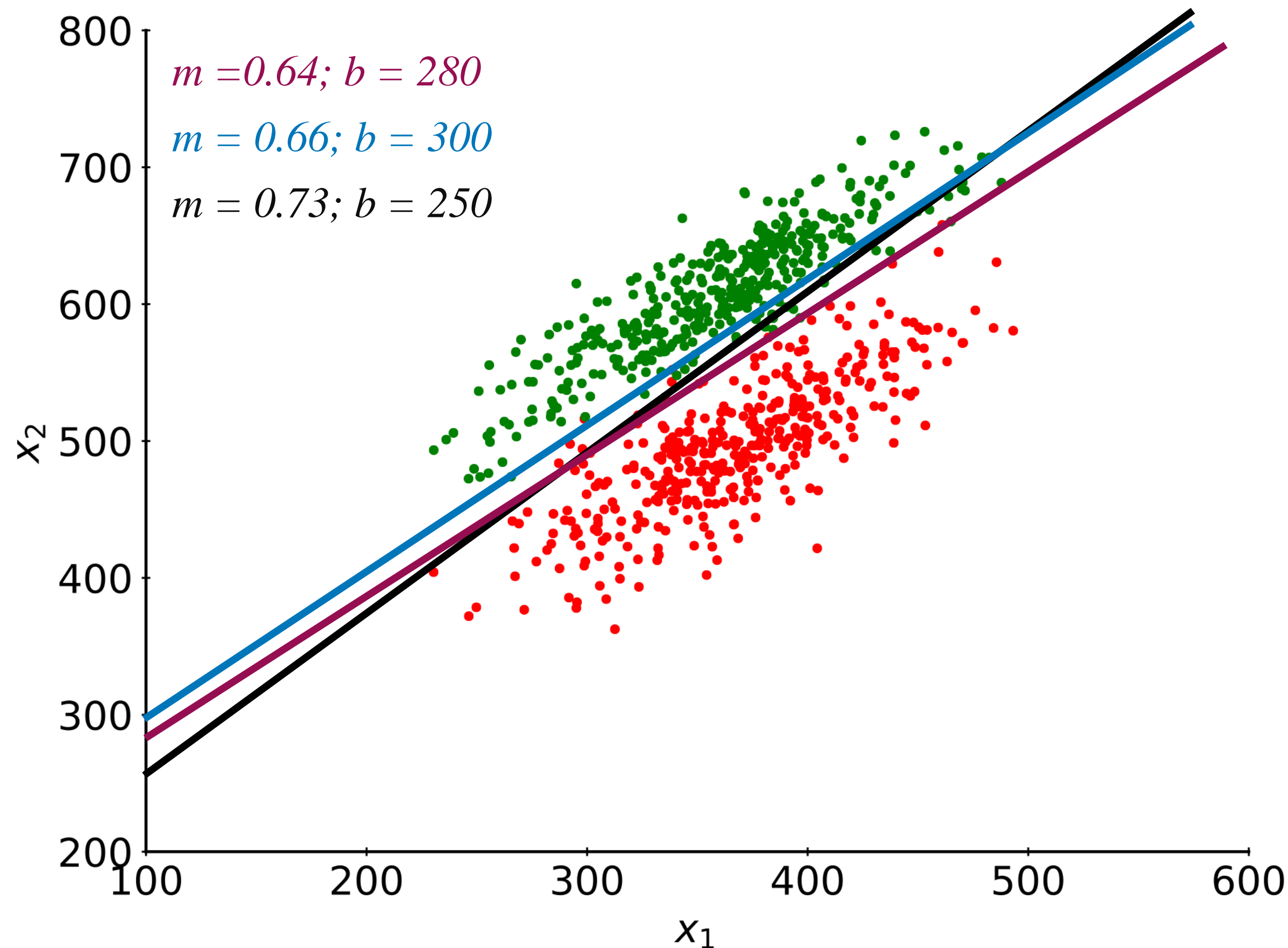


Usando un set de features extendido, el problema se vuelve mucho más fácil

Se reduce el problema a encontrar la mejor curva que separa ambos clusters.

Predicciones a partir de datos

Un problema de clasificación de juguete - modelos lineales



¿Cuál es la mejor curva que separa estos dos conjuntos?

- Primero, ¿qué significa “mejor”?
- Una familia importante de modelos usa combinaciones lineales de los features (es decir, una recta en el espacio de los features).



Modelos lineales

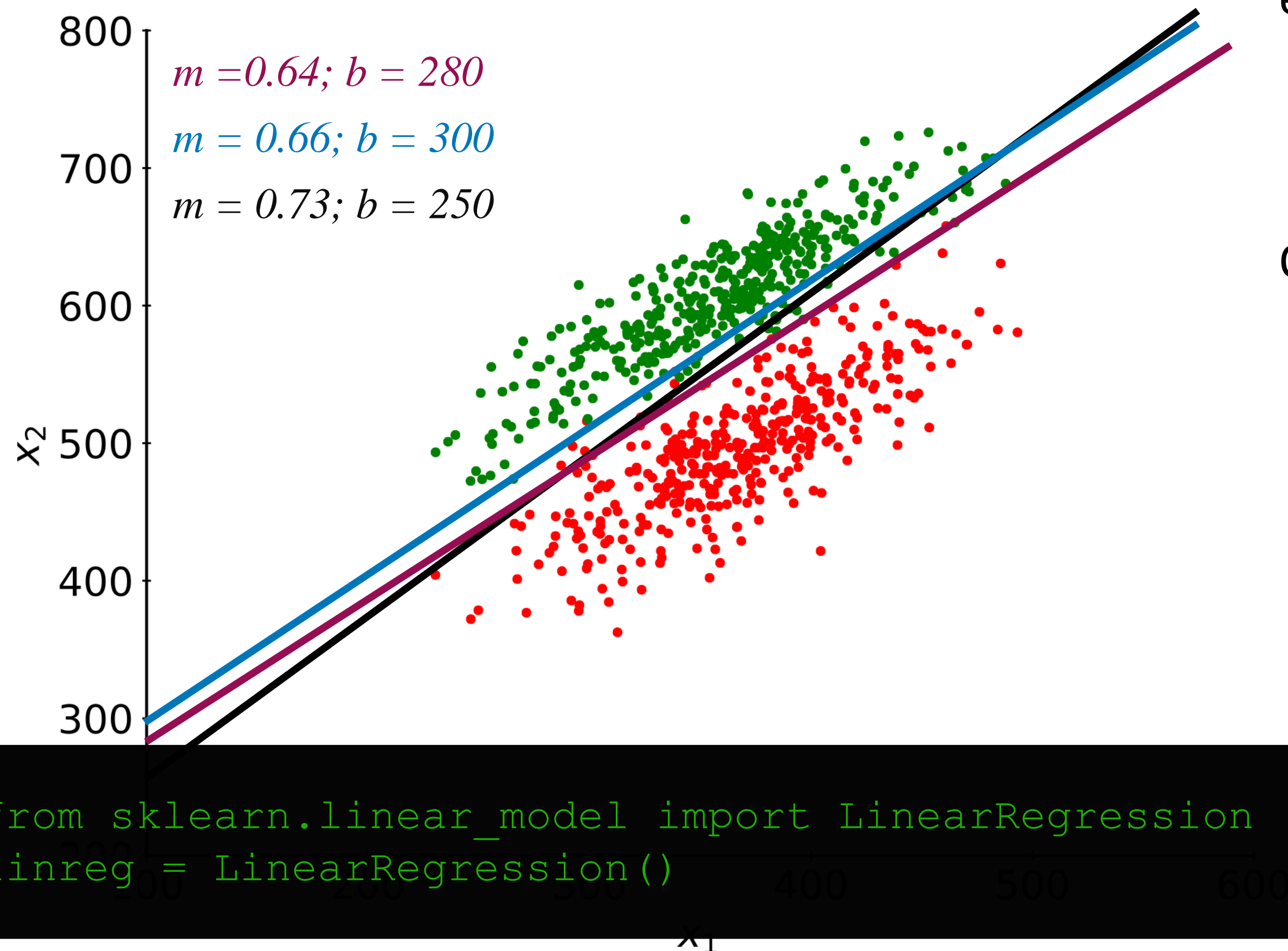
Ok, pero ¿qué línea recta?

Todas se pueden escribir con la expresión:

$$x_2 = \underset{\text{slope}}{m} \cdot x_1 + \underset{\text{intercept}}{b}$$

Predicciones a partir de datos

Un problema de clasificación de juguete - modelos lineales



Todas se pueden escribir con la expresión:

$$x_2 = m \cdot x_1 + b$$



Que es equivalente a:

$$w_1 \cdot x_1 + w_2 \cdot x_2 + w_0 = 0$$

Pesos / weights

Bias

Parámetros del modelo

Entrenamiento

Predicciones a partir de datos

Aparte: hiperparámetros

Ahora bien, ¿por qué funciones lineales de los features, en lugar de, por ejemplo, esto?

$$w_0 + w_1 \cdot x_1^2 + w_2 \cdot x_2^2 = 0 \quad ?$$

o esto
$$w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_1^2 + w_4 \cdot x_2^2 + w_5 \cdot x_1 x_2 = 0 \quad ?$$

¿Y por qué exponentes 2, en lugar de, digamos, cualquier número natural M?

¿Cómo cambiaría esto nuestro modelo? ¿Lo haría mejor? ¿Más fácil o rápido?

Hiperparámetros

Predicciones a partir de datos

Un problema de clasificación de juguete - evaluación

La performance del modelo tiene que ser evaluada (idealmente en datos nunca antes vistos) antes de usarse en la vida real:

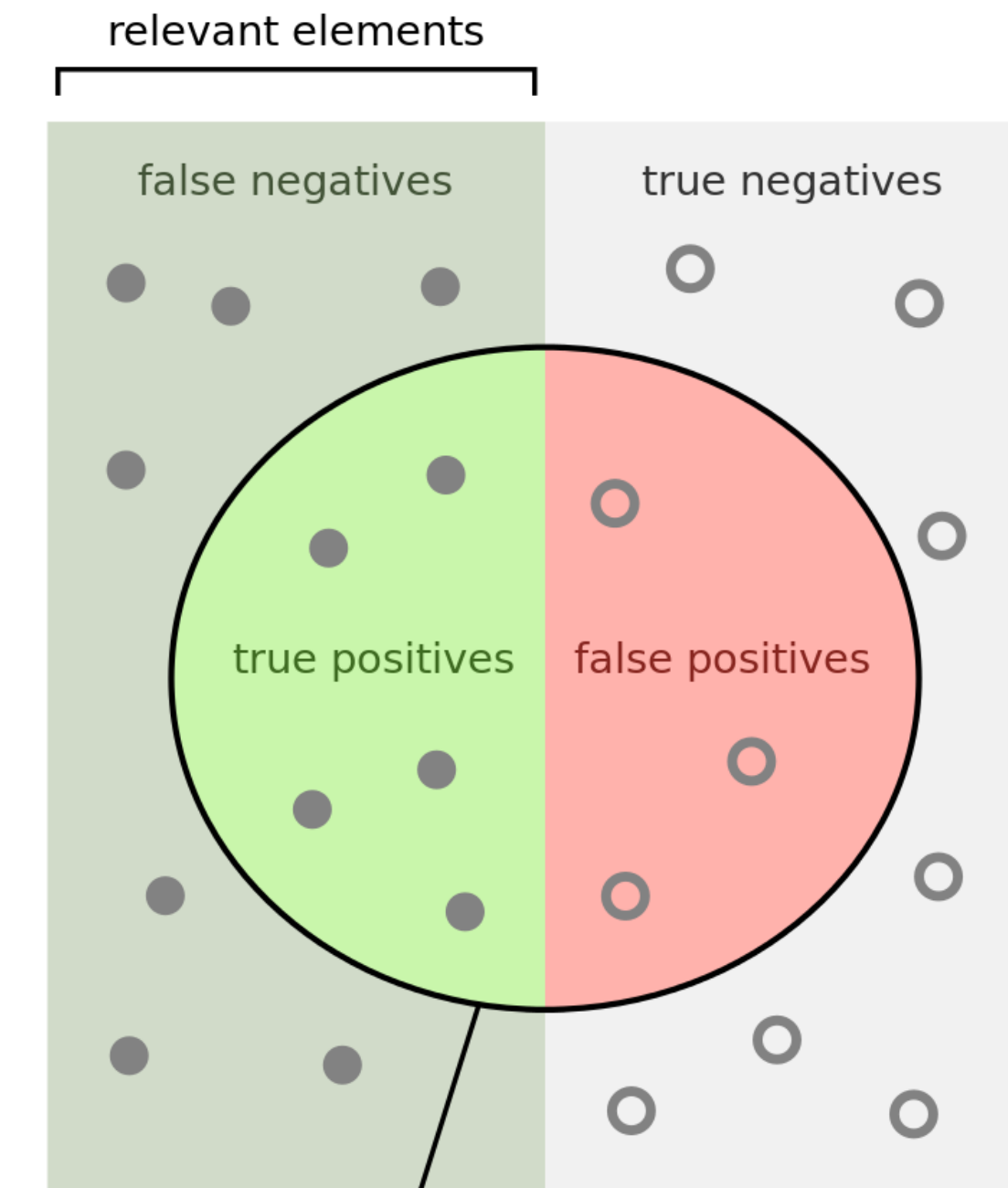
- fracción de veces que el modelo predice un cambio en el valor de la acción correctamente.
- fracción de veces que ocurre un aumento que es predicho por el algoritmo.
- la fracción de veces que se predice un aumento que se produce.

Necesitamos otras métricas para los modelos de regression problems.

Accuracy / Exactitud

Recall / Exhaustividad

Precision



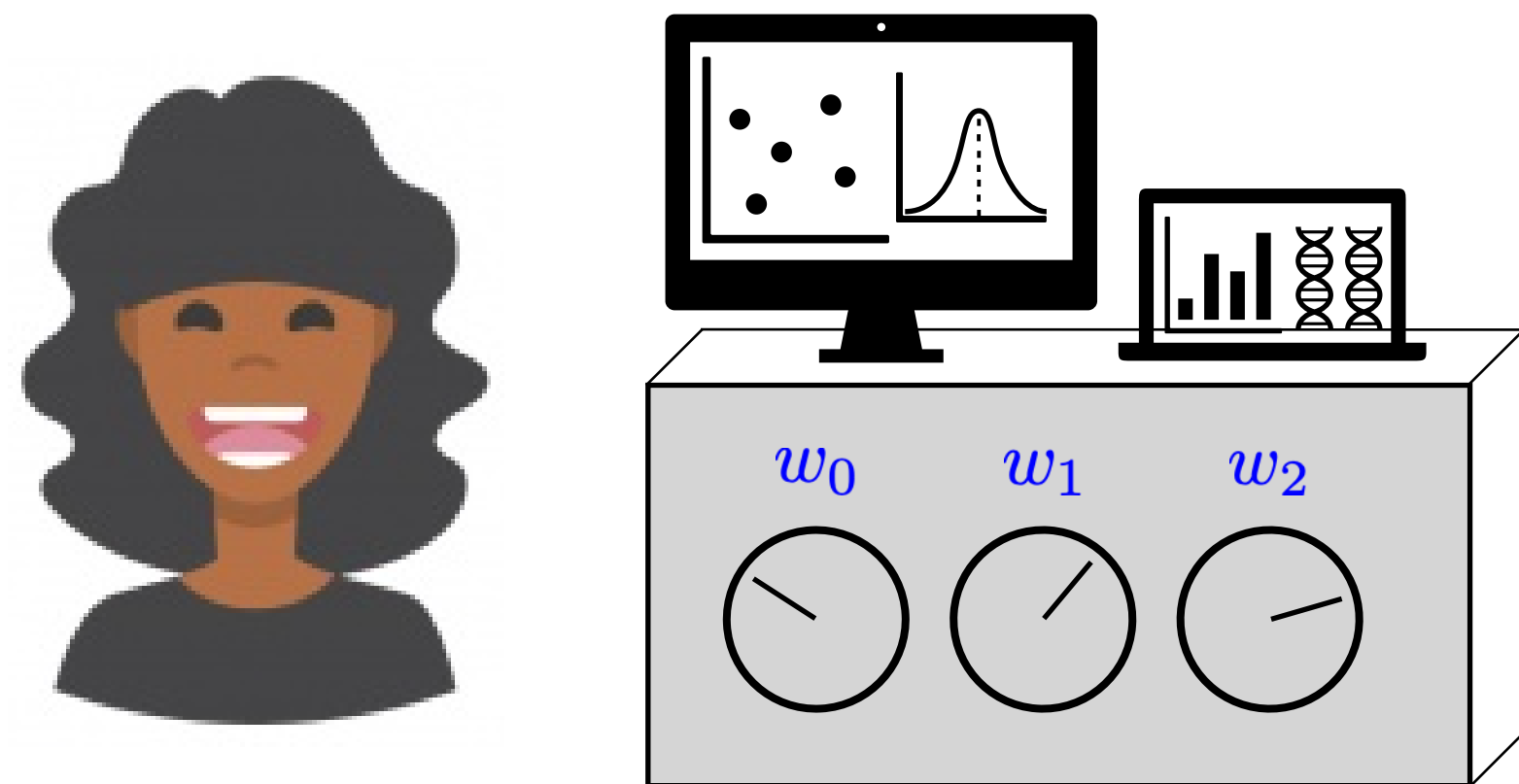
```
# Training evaluation
y = linreg.predict(data)
some_metric(y, t)

# Test evaluation
data_test, y_test = get_test_data()
y_test = linreg.predict(data_test)
some_metric(y_test, t_test)
```

Predicciones a partir de datos

Un problema de clasificación de juguete - interpretabilidad

$$w_1 \cdot x_1 + w_2 \cdot x_2 + w_0 = 0$$



La mayor ventaja de un modelo simple es su interpretabilidad:

- Alice entiende y como afecta al resultado cada elemento.
- Puede explorar los efectos de cada feature en sus predicciones sobre el precio de una acción.
- En particular, puede describir como cada feature afecta la predicción de forma separada del resto.
- Si necesita, puede expandir su modelo incluyendo nuevos features..

Predicciones a partir de datos

Un problema de clasificación de juguete - modelos de Aprendizaje Automático

Bob tiene una idea diferente:

- En lugar de considerar solamente dos o tres precios de acciones, descargó datos de cientos de compañías, algunas que ni siquiera son empresas tecnológicas.
- Además, incluyó datos de encuestas, índices de confianza, y todo tipo de cosas.
- Su algoritmo, a pesar de estar relacionado con los modelos lineales, es altamente no lineal y por lo tanto mucho más difícil de entrenar.
- El modelo es muy difícil / imposible de interpretar..



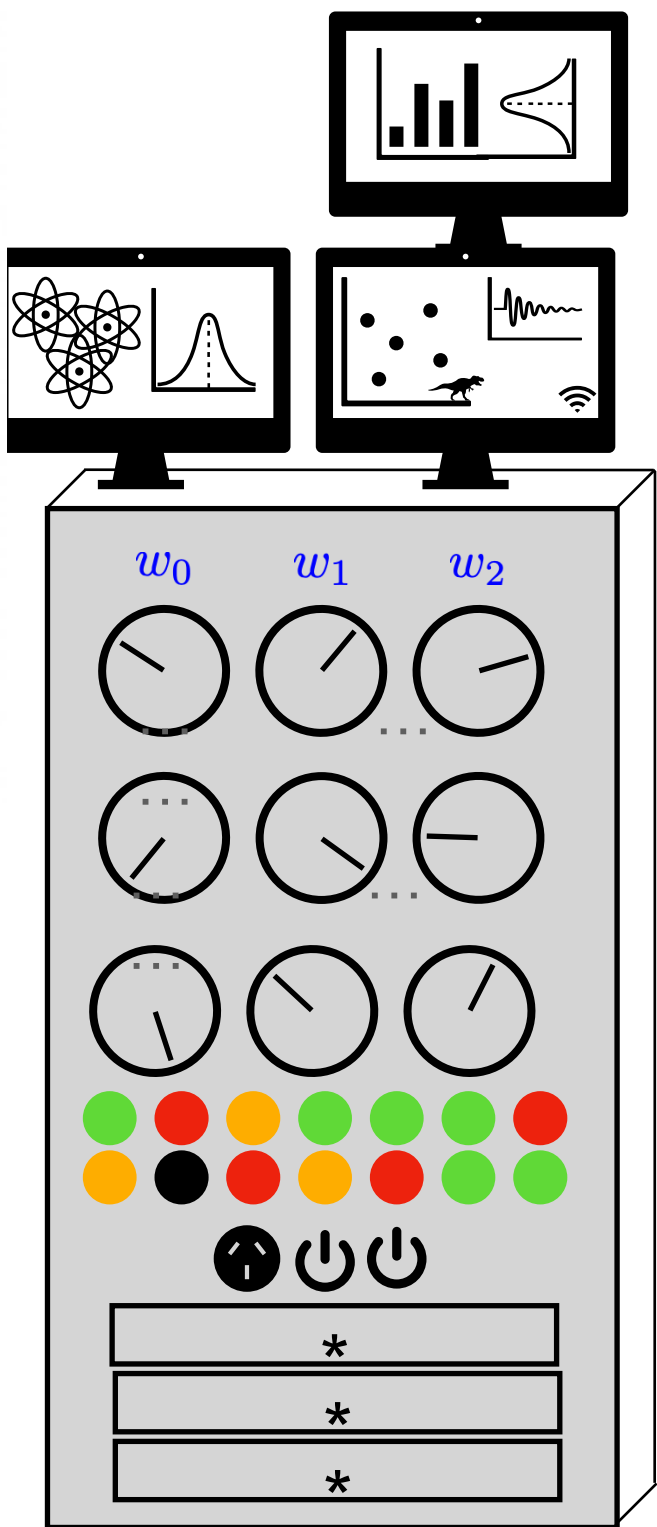
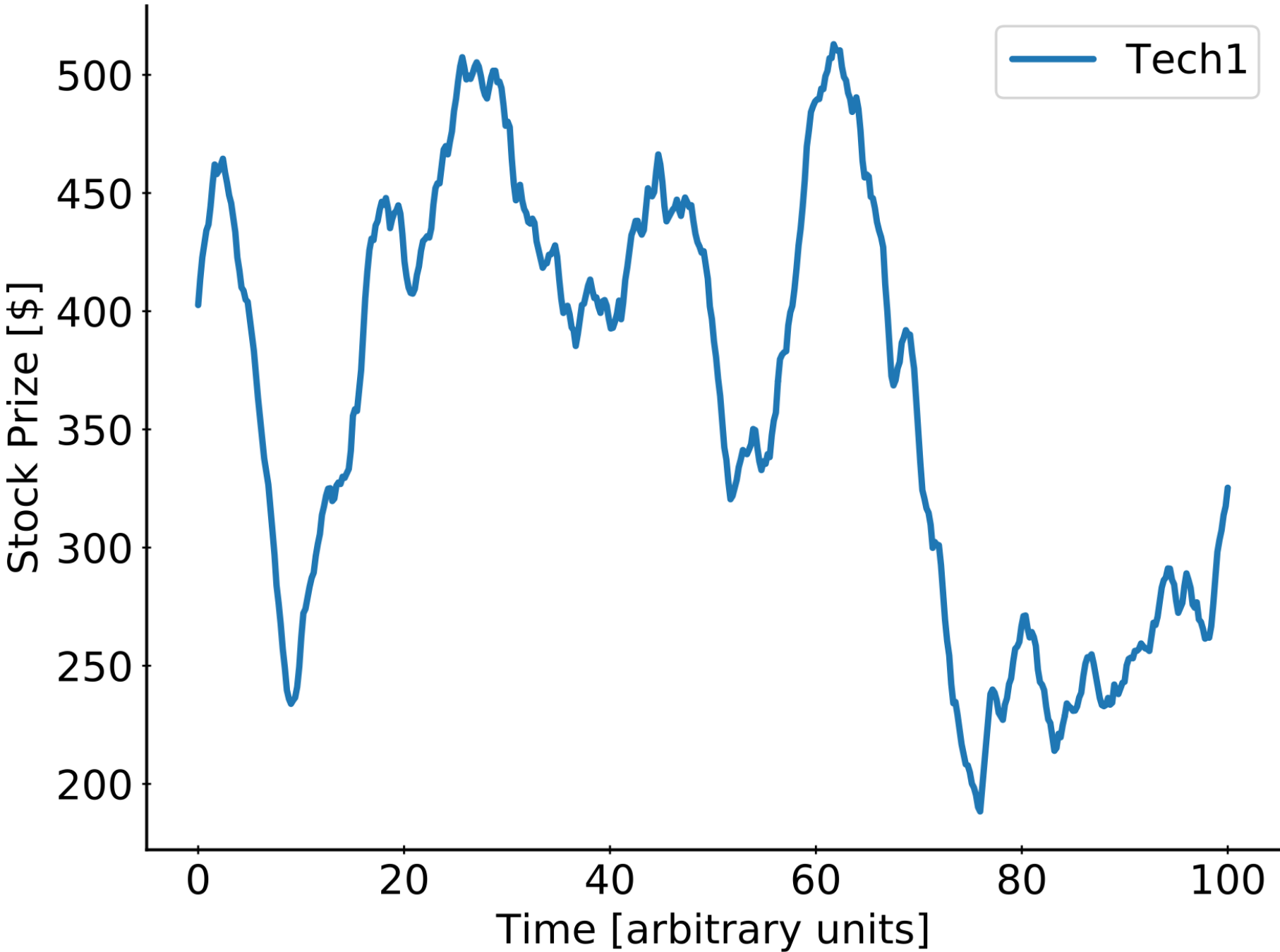
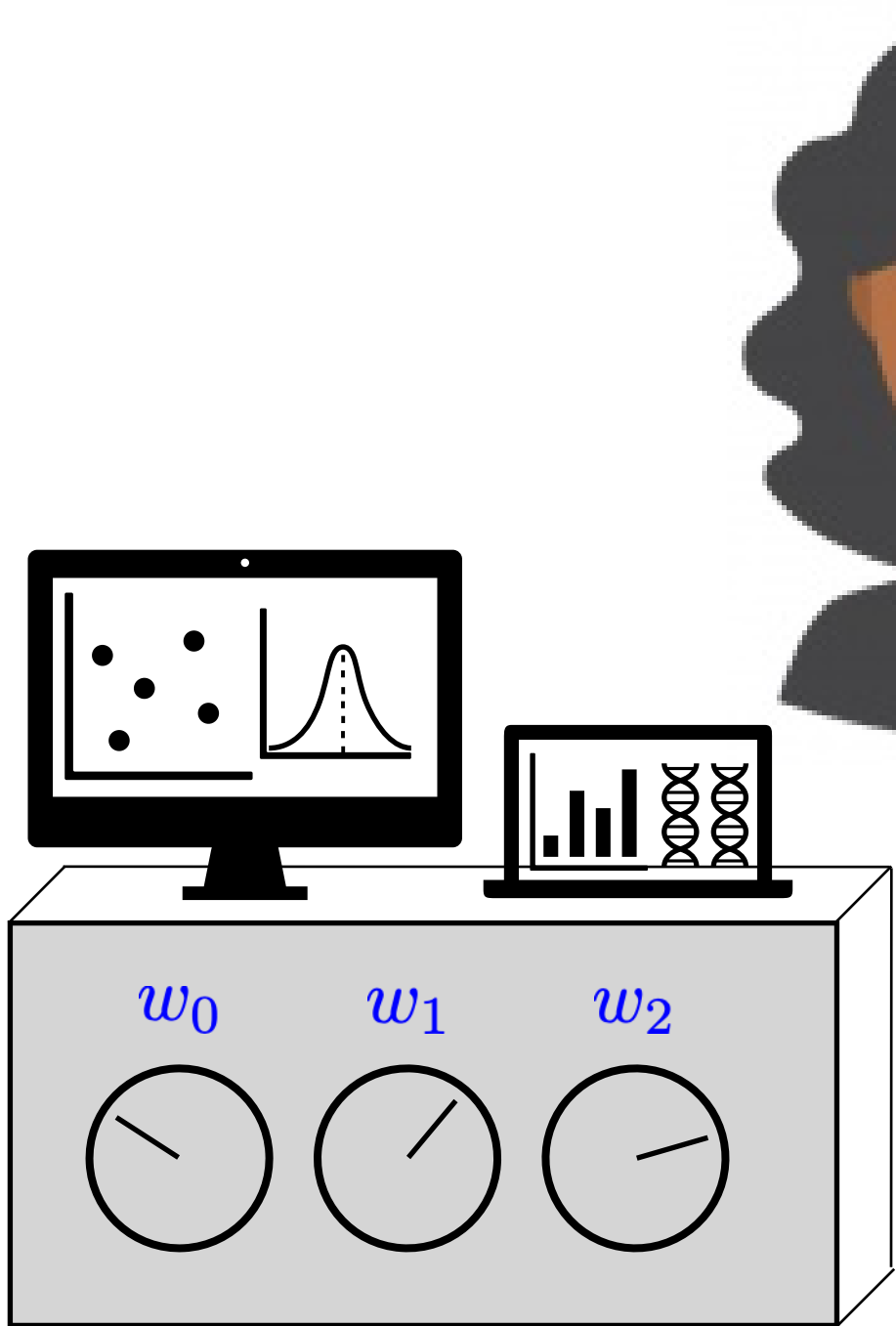
Data Curation
Pre-processing

Nonlinear programming

```
# Training  
model.fit(bob_data_processed, t)  
  
< Magic happens ... >
```


Predicciones a partir de datos

Un problema de clasificación de juguete - resultados



~89%

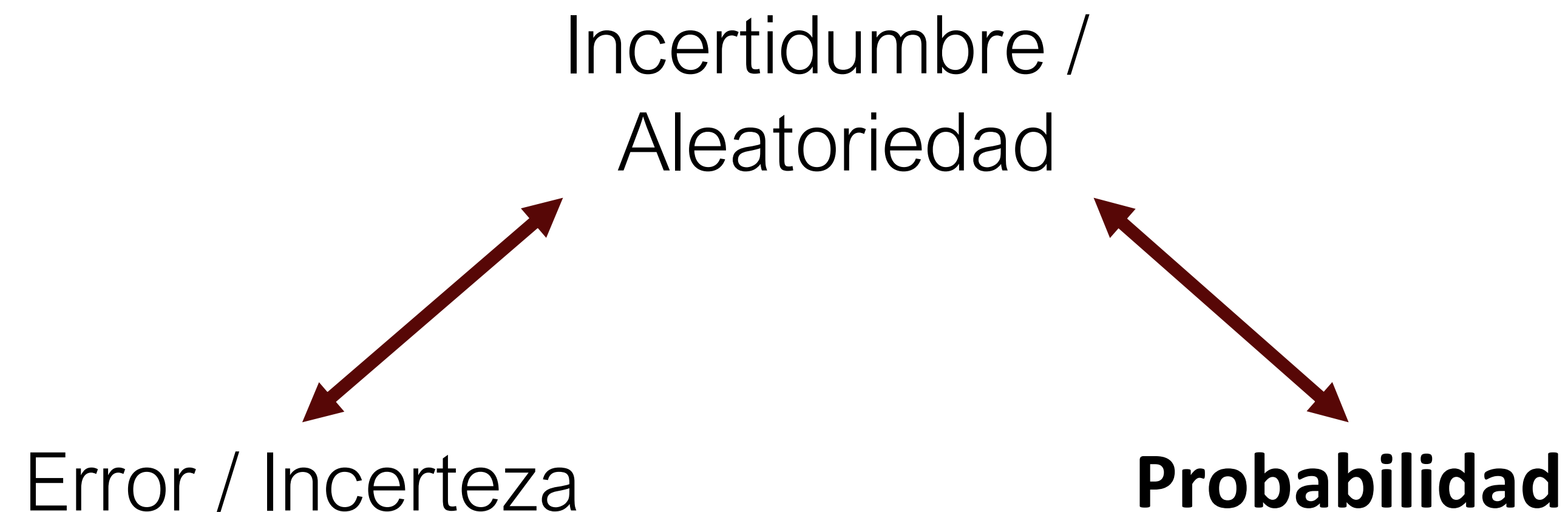
Accuracy

>99%

Probabilidad

¿Por qué tenemos que ver este tema?

- ¿Qué pasa si quiero describir con mi **modelo** eventos que no son deterministas (es decir, que tienen ciertas incertidumbre)?
- ¿Existe realmente alguna posibilidad de hacer un modelo determinista de los datos? Si reconocemos la existencia de incerteza, la aleatoriedad siempre está presente.



- Clásica (Laplace). Principio de indiferencia. Las probabilidades se reparten de forma igual entre todos los posibles resultados, suponiendo que pueden ser considerados iguales.
- Frecuentista (empírica). La probabilidad de un evento es igual a su frecuencia relativa (en relación con la cantidad de intentos posibles), cuando el número de ensayos aumenta.
- Bayesiana. También llamada probabilidad epistémica. Entiende a la probabilidad como un grado de incertidumbre.

¿Cuál es la probabilidad de obtener un as al tirar un dado no cargado?

$$P(A, B) = P(A|B) P(B) = P(B|A) P(A)$$

Regla del
producto

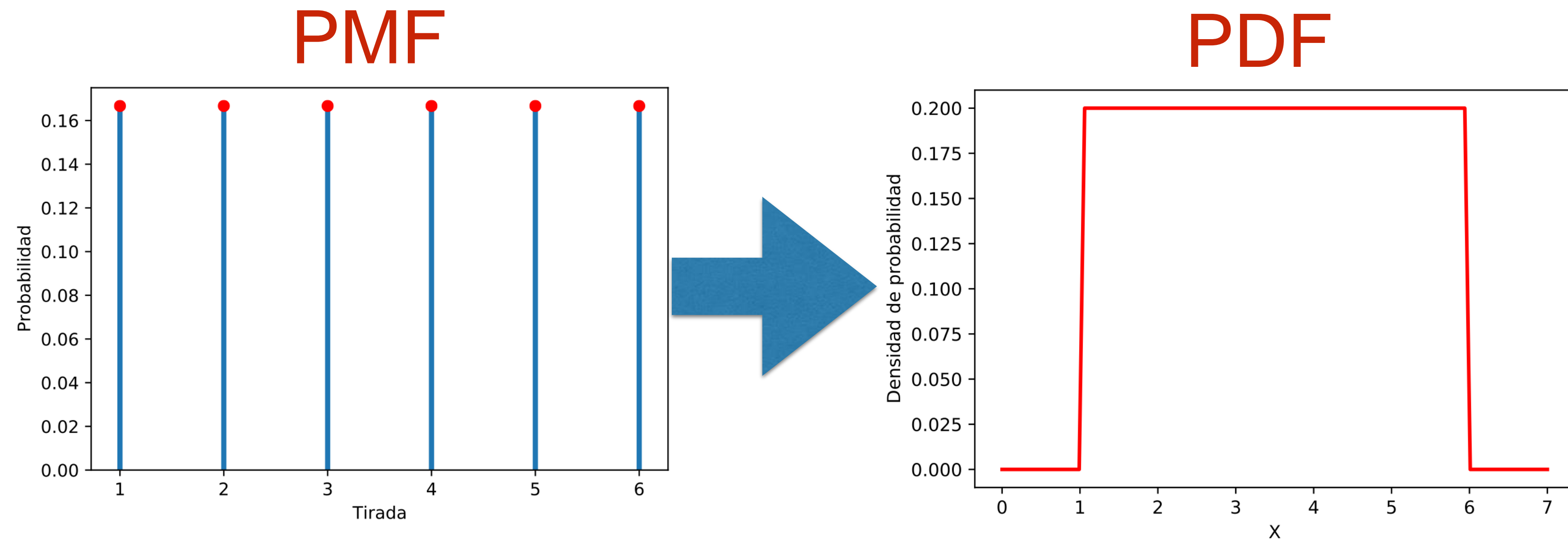
Teorema de
Bayes

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

Regla de la
probabilidad total

$$P(A|B) = \frac{P(B|A) P(A)}{P(B|A) P(A) + P(B|\bar{A}) P(\bar{A})}$$

Funciones de densidad de probabilidad (PDF)



$$\forall i, f(x_i) \geq 0$$

$$\sum_i f(x_i) = 1$$

$$f(x) \geq 0 \quad \int_{-\infty}^{\infty} f(x) d(x) = 1$$

$$P(x \in (a, b)) = \int_a^b f(x) dx$$

Si en lugar de la ley de probabilidad, tuvieramos una muestra (ilo que ocurre a menudo!)

$$\{x_i\} \quad i = 1 \dots N$$

$$\mathbb{E}_f[x] \longrightarrow \bar{X} = \frac{1}{N} \sum_{i=1}^N x_i$$

Ley de los grandes números

$$\text{var}_f[x] \longrightarrow \bar{S} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{X})^2$$

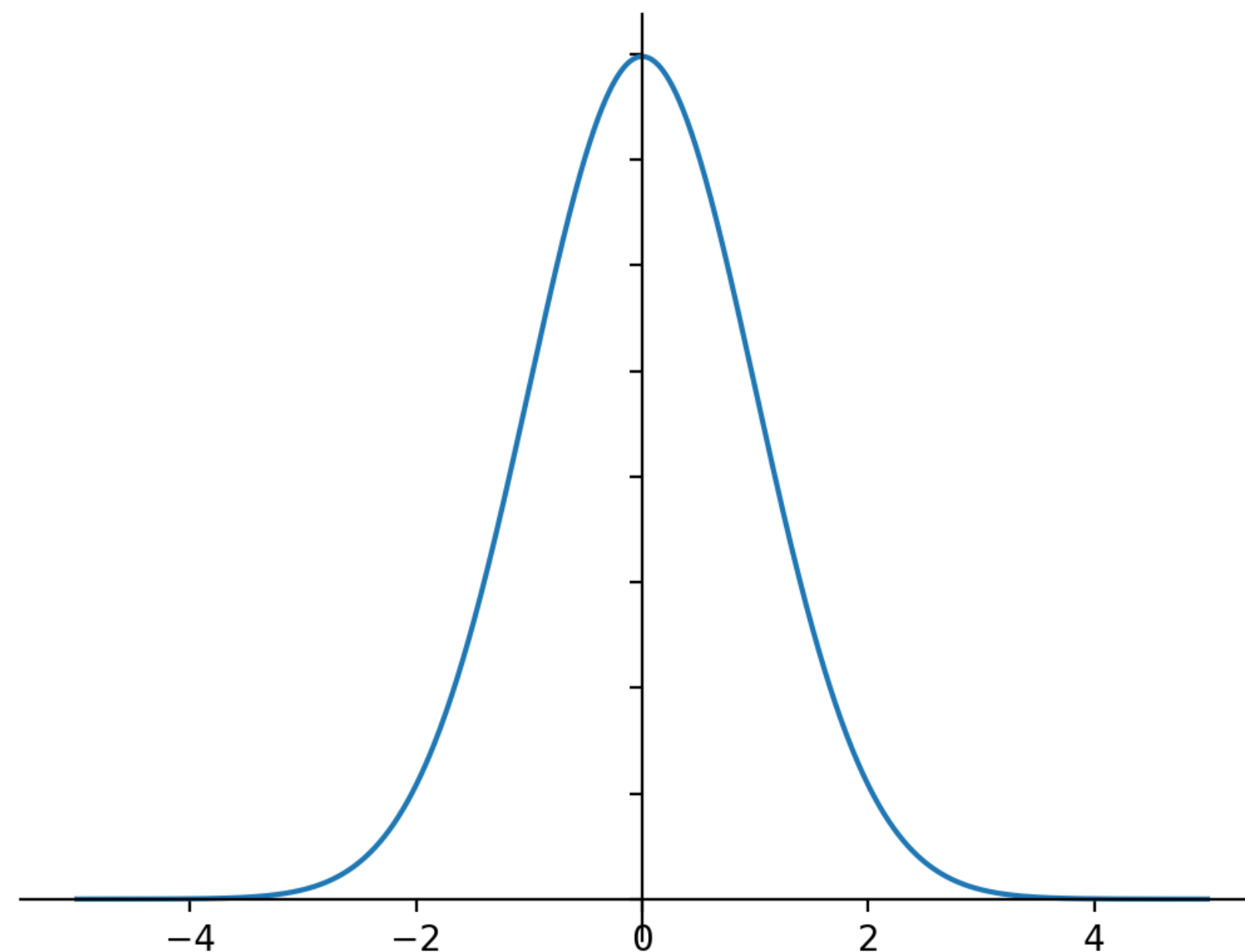
Distribución Normal / Gaussiana

En una sola dimensión

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2} (x - \mu)^2 \right\}$$

$$\mathbb{E}[x] = \mu$$

$$\text{var}[x] = \sigma^2$$



Distribución de Bernoulli

- Tengo un proceso que tiene una probabilidad μ de éxito, y solo dos resultados (éxito=1; fracaso=0). Ejemplo: tirar una moneda.
- El espacio de muestreo, entonces, es $S = \{0, 1\}$, y llamemos a la variable aleatoria X .
- ¿Podemos escribir la PMF de este proceso?

$$P(X = 1) = f(1) = ??? \quad P(X = 0) = f(0) = ???$$

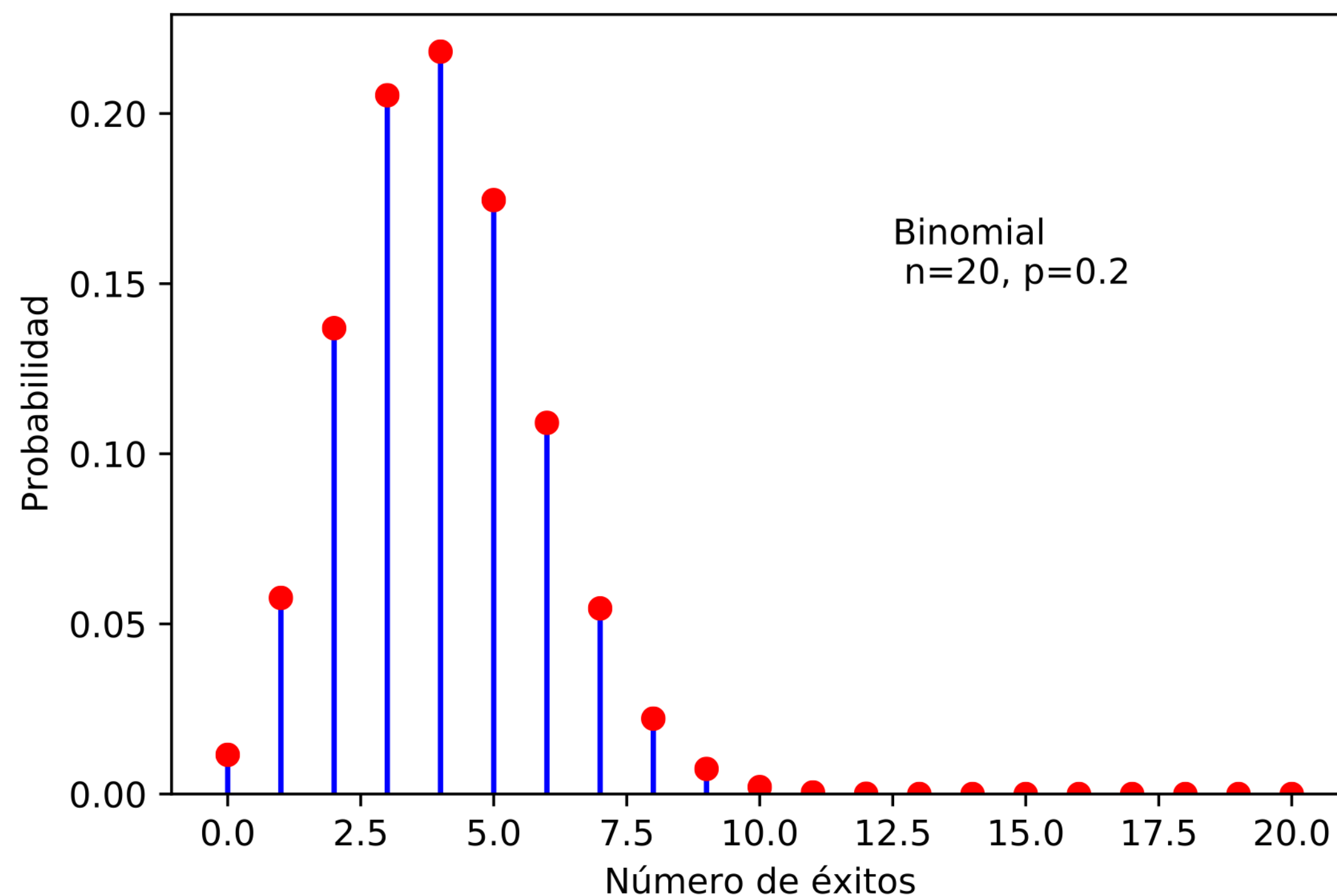
$$f(X) = \mu^X \cdot (1 - \mu)^{(1-X)}$$

Distribución
de Bernoulli

¿Cuál es el valor de expectación? ¿Y la varianza?

Distribución Binomial

- Describe el resultado de n experimentos con una variable de Bernoulli de probabilidad μ



$$p(k|n, \mu) = \binom{n}{k} \mu^k (1 - \mu)^{n-k}$$

¡A los notebooks!

`01_Probabilidad.ipynb`