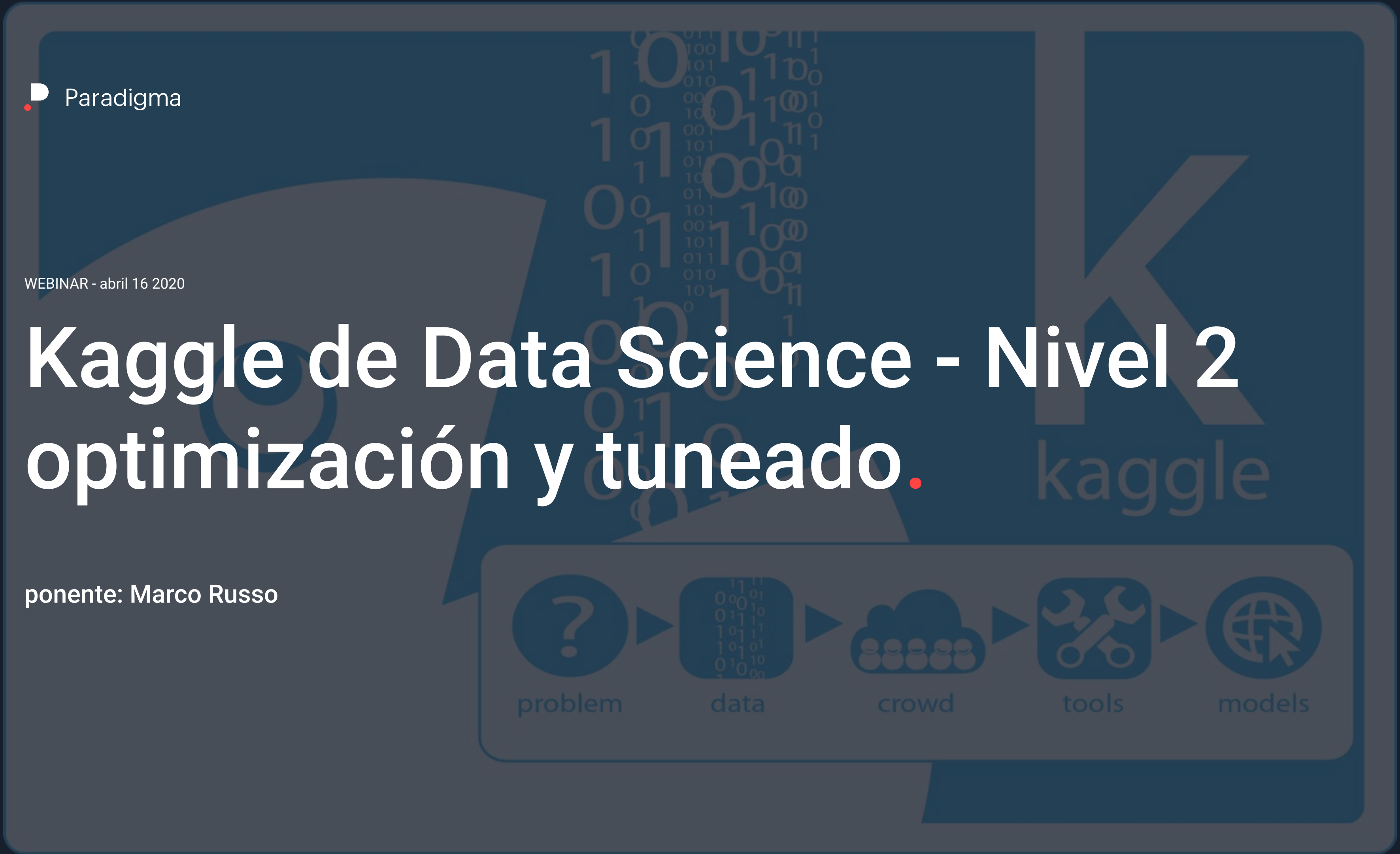
Paradigma

WEBINAR - abril 16 2020

Kaggle de Data Science - Nivel 2 optimización y tuneado.

ponente: Marco Russo



Quién soy.

- **Consultor en Data** en Paradigma Digital, con más de 7 años como docente para importantes escuelas de negocios y profesor colaborador en la UOC.
- **Especializado** en data mining, optimización de modelos y machine learning en área del Marketing, Retail y Banca-Finanzas entre otras. Además de especialista en analítica digital, SEO y PPC en digital marketing y visualización de datos - BI.
- **Apasionado** de IoT, datos y robótica, dedico el tiempo con mi familia y a mi deporte favorito, bici de carretera.

Marco Russo (aka marcusRB)

@rb_marcus

github.com/marcusRB

marcusRB

Qué vamos a ver.

1. Organización del entorno de trabajo
2. Flujo de trabajo
3. Tips
4. Optimización de los modelos
5. Demo

01.01

Introducción

Nivel 1.

Webinar y repo - nivel 1.

https://bit.ly/34HCEGz

01.02

Introducción

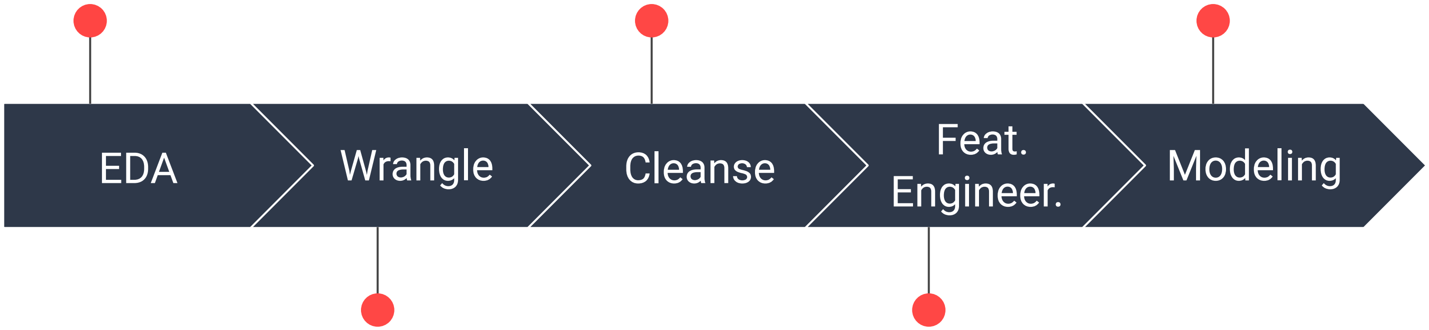
Flujo de trabajo.

Flujo de trabajo.

Check feature one by one new patterns and insights

Check missing values and impute / remove them.

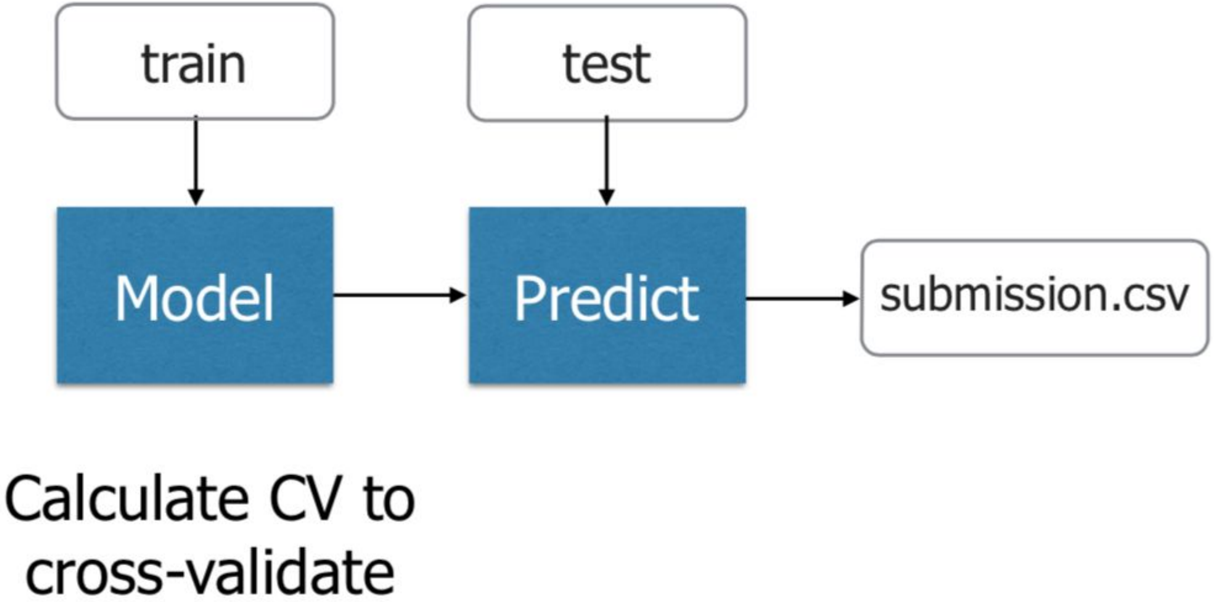
Apply selected features to ML models.



Prepare features for the final model.

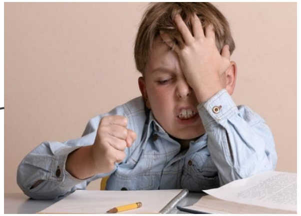
Create new features or check cluster / groups to select the best ones.

Build prediction model.



Cómo funciona.

- “Train” a **model** on lots and lots of data
    - Start with poor predictions
    - Make little tweaks to improve
    - Like child doing homework!
- Infer predictions on new data

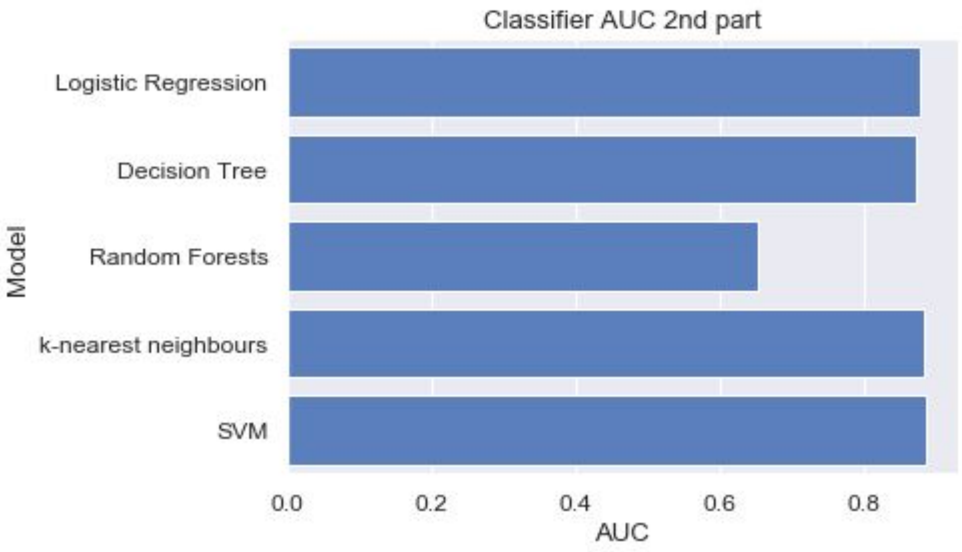


Training

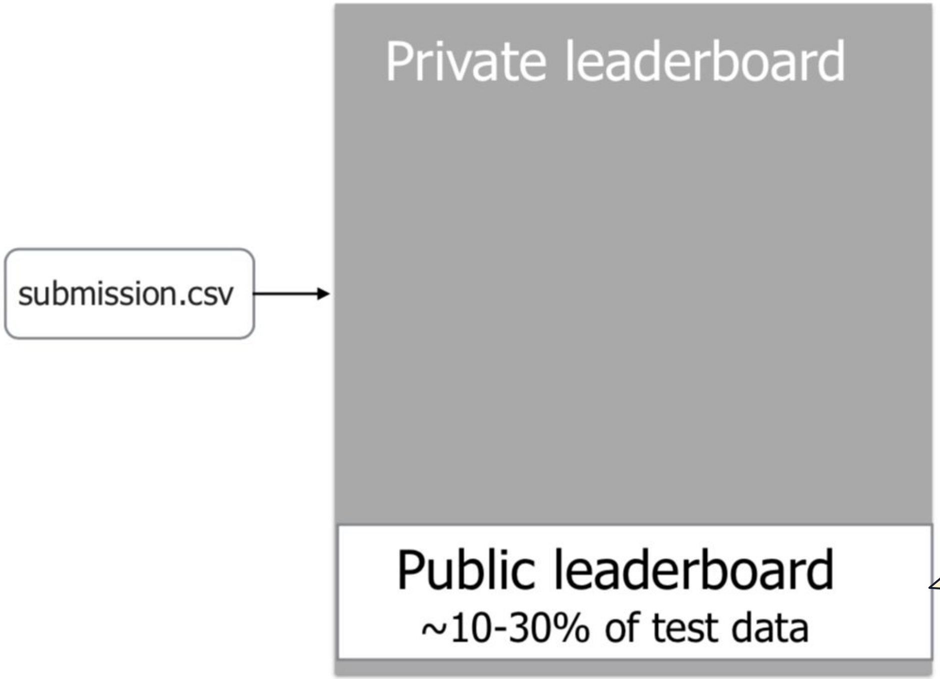
Inference

Mejora continua.

	Model	Score_1st
4	Decision Tree	99.99
1	KNN	98.27
0	Support Vector Machines	97.69
2	Logistic Regression	97.08
3	Random Forest	93.86



Submit.



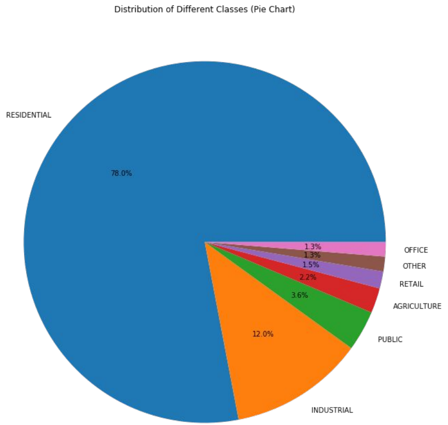
Casi siempre no coincide con **nuestro resultado de testing**!!!

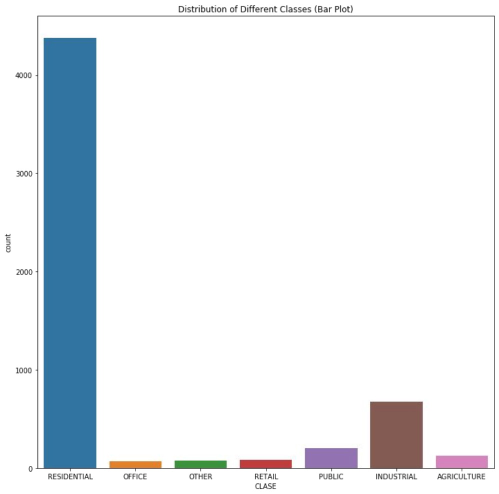
01.03

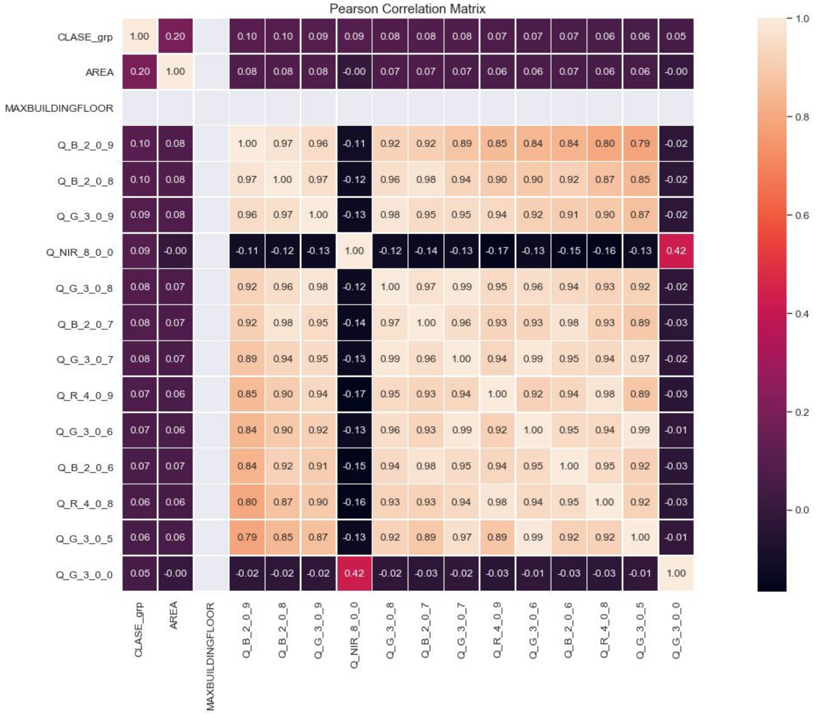
Introducción

Tips.

EDA y correlaciones.







Features.

Feature Engineering

Step1.

Feature Selection

choose efficient features
and
abandon useless features

Step2.

Feature Extraction
(Dimensional Reduction)

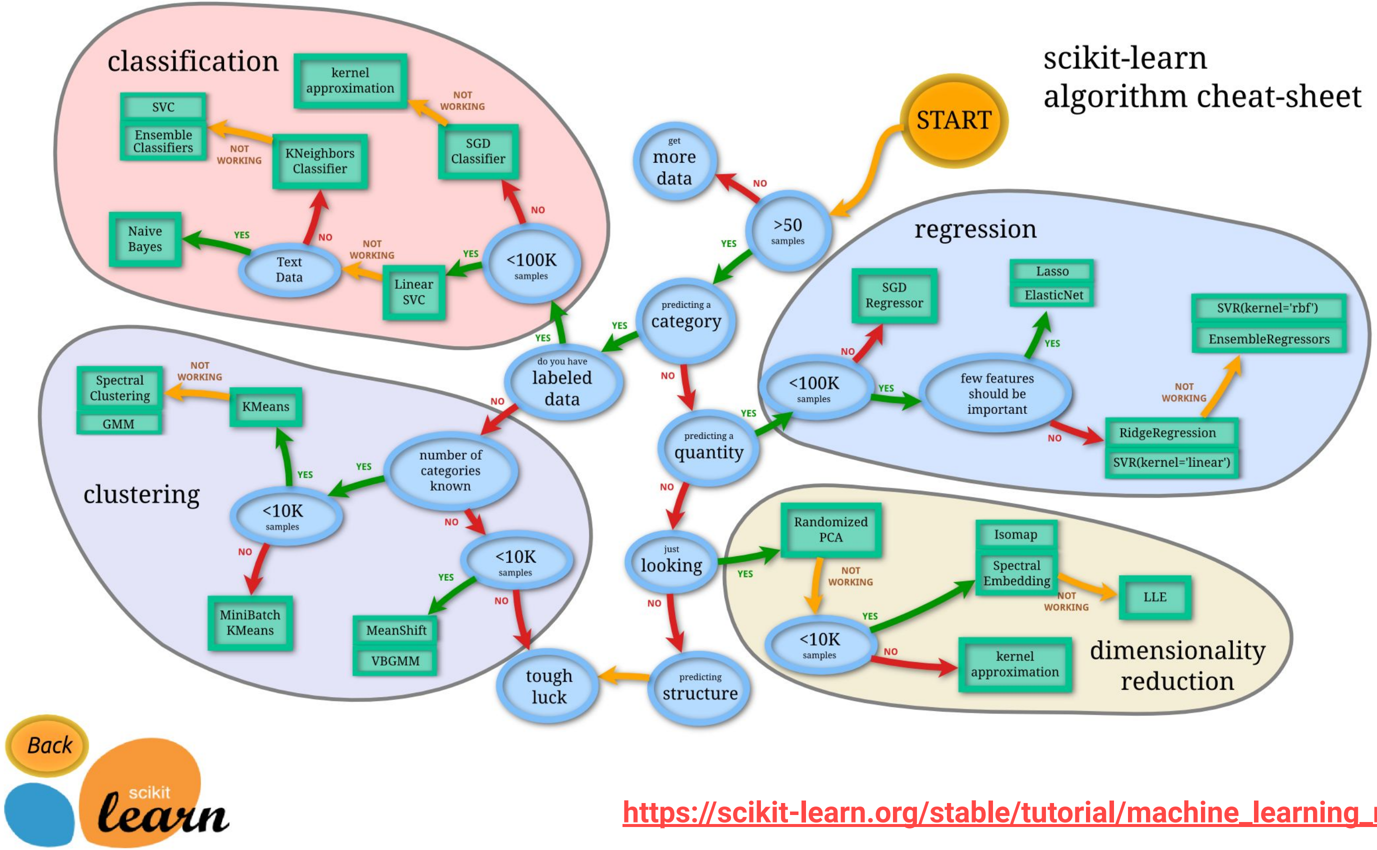
- PCA
  - SVD
- LDA

01.04

Introducción

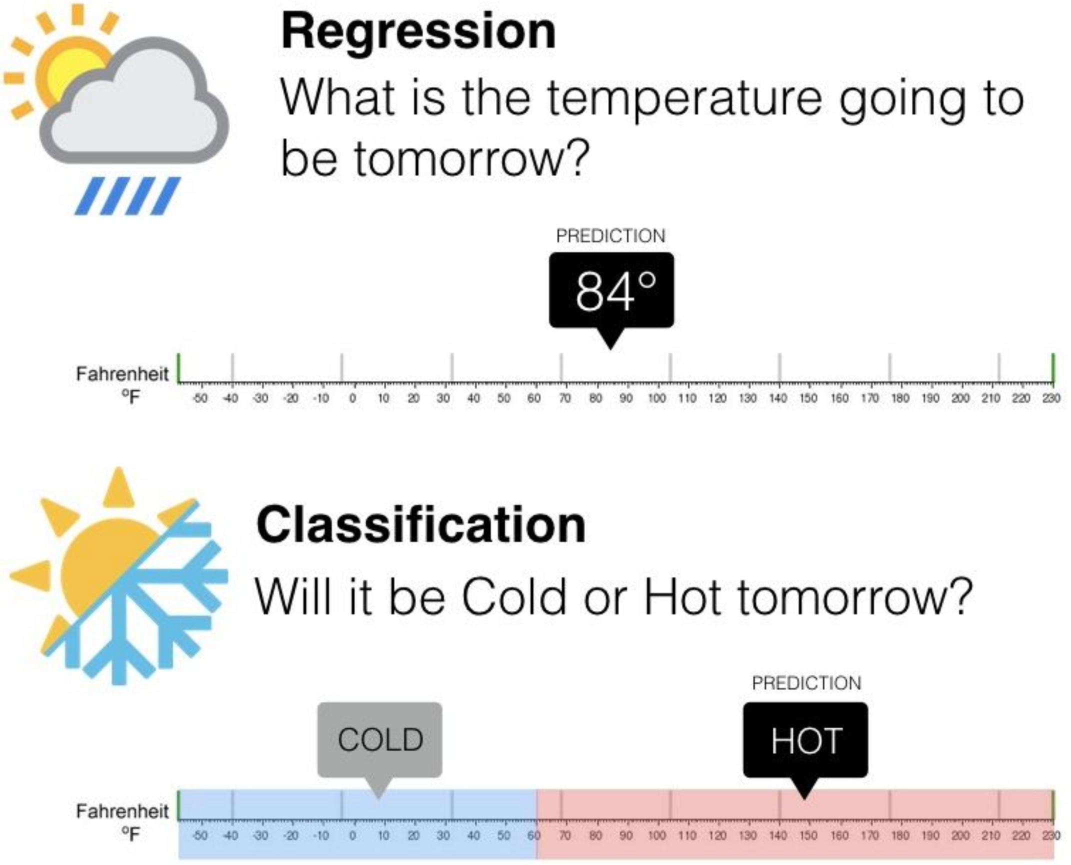
Optimización de modelos.

Eliges el estimador.

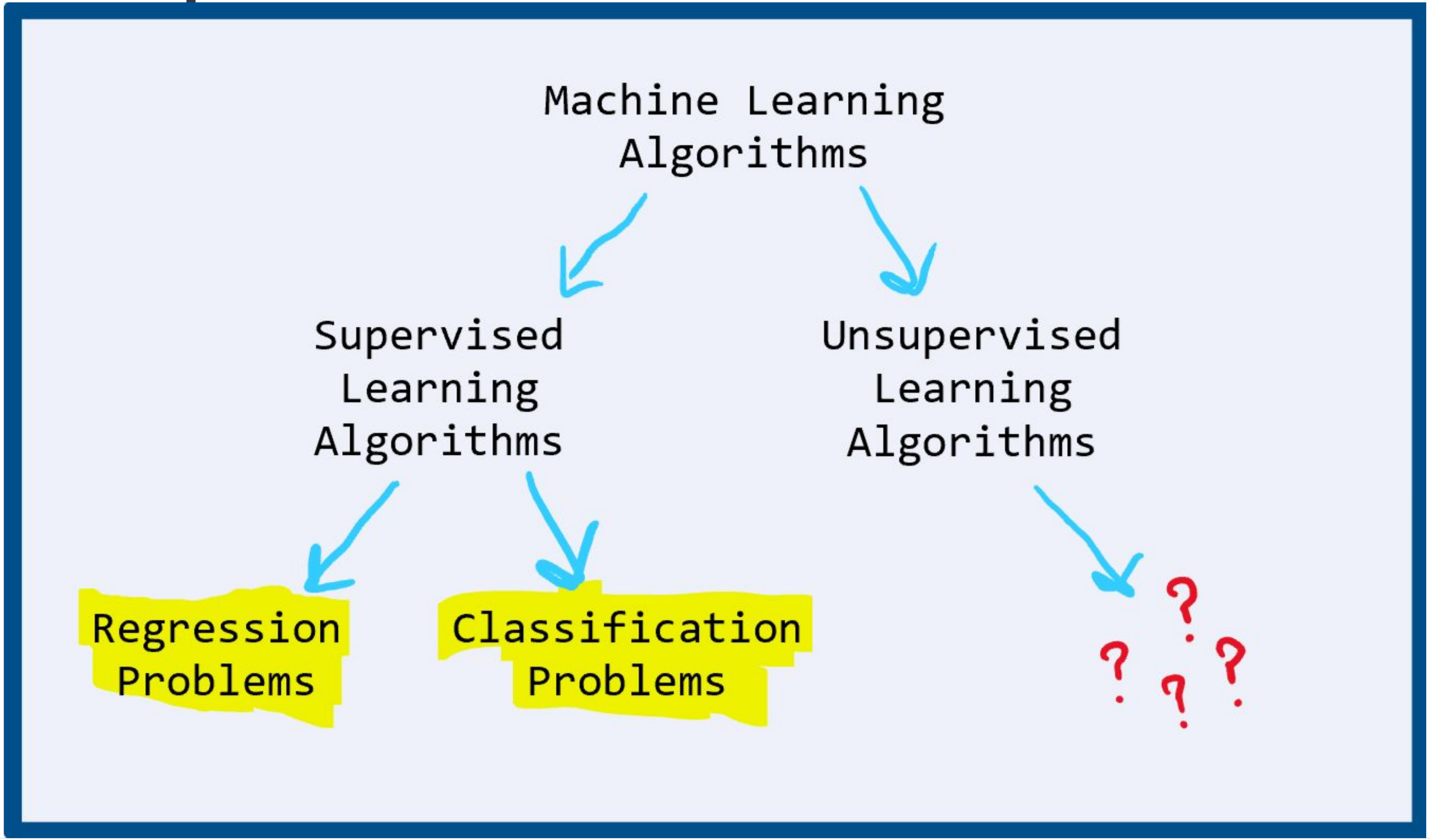


https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

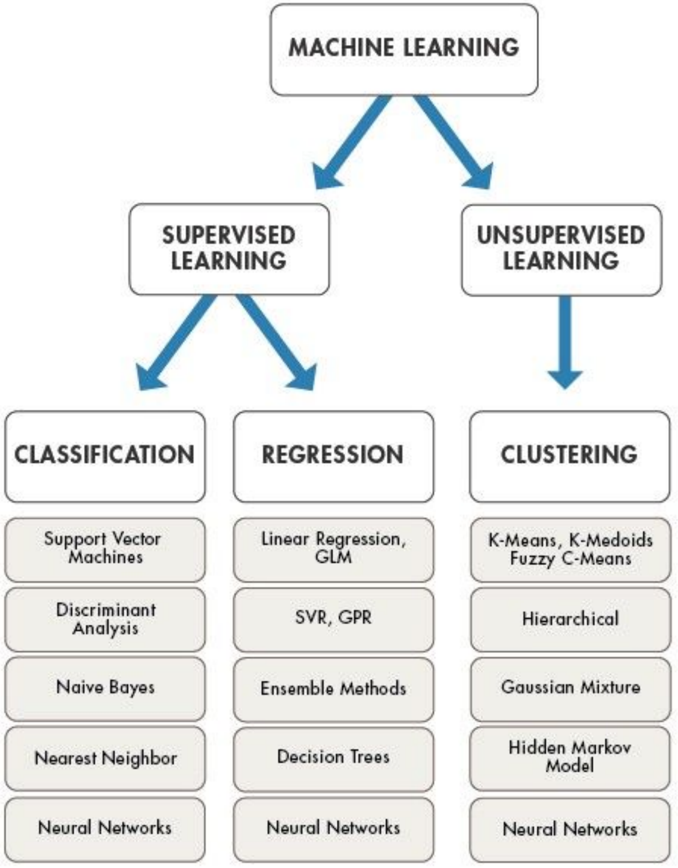
Los más frecuentes son.

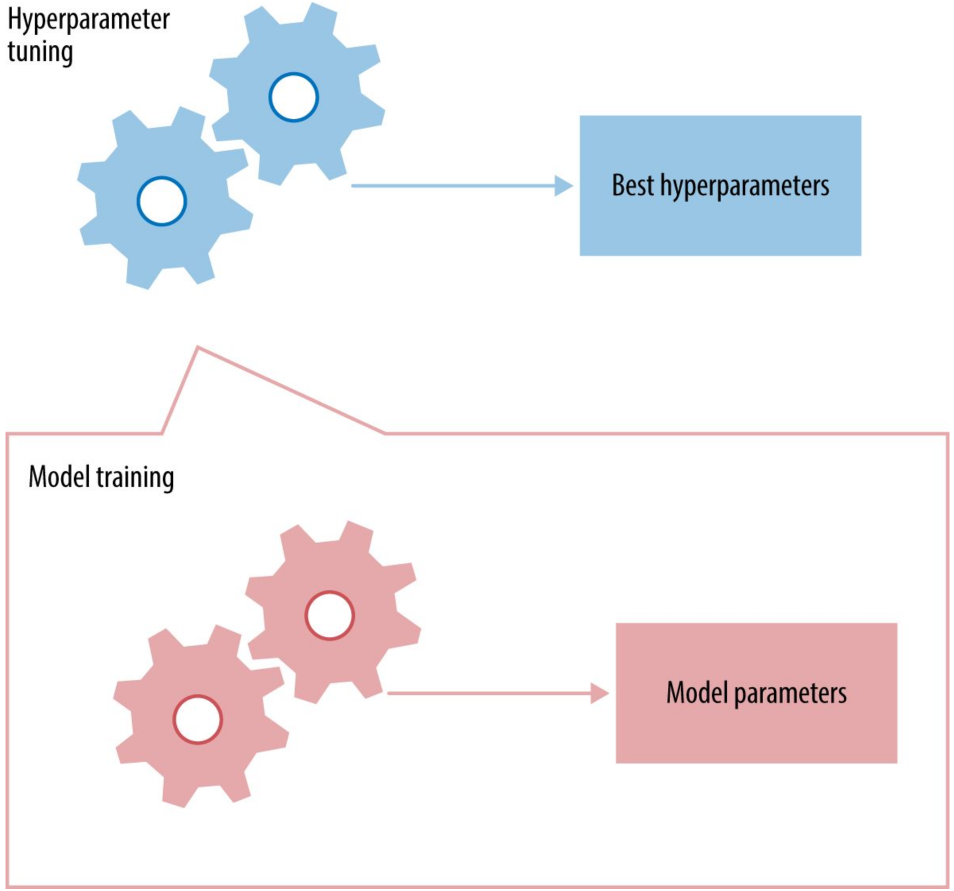


Depende del dataset.

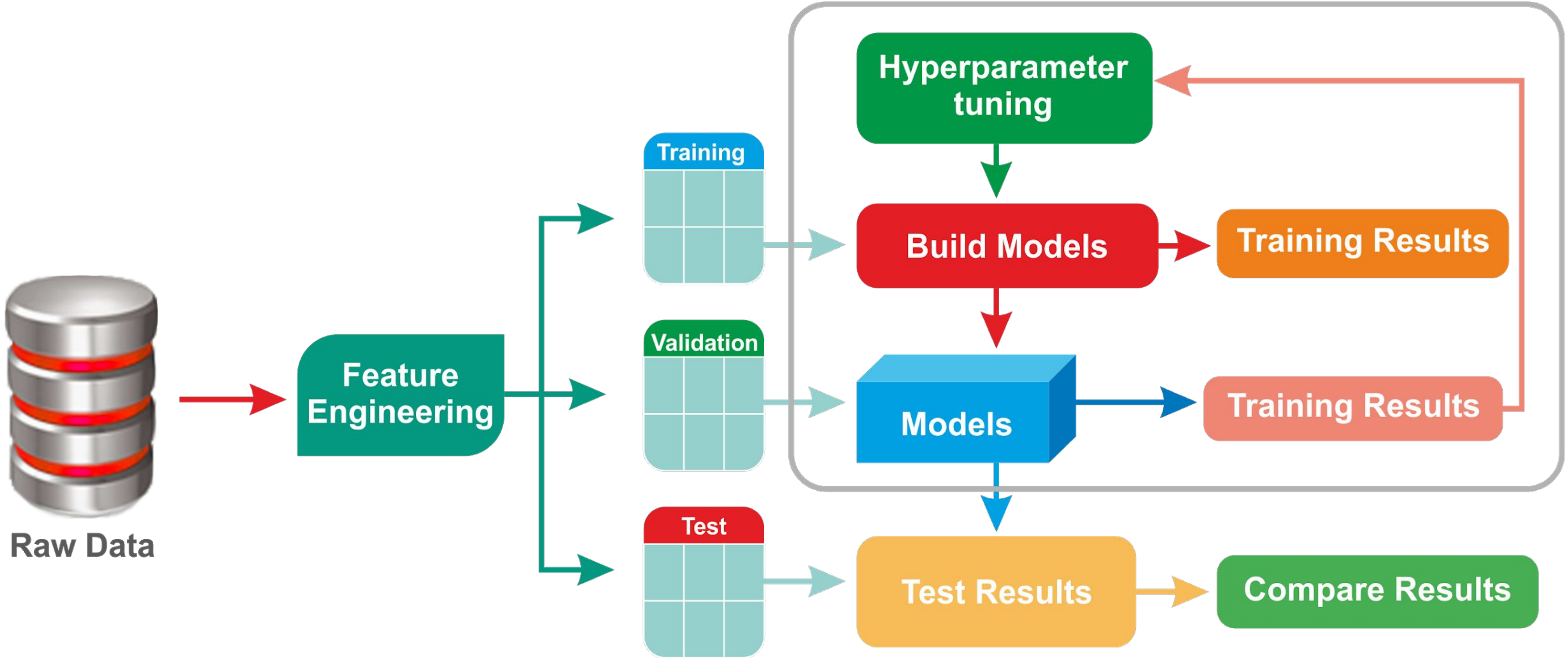


Algoritmos más utilizados.





Pipeline Optimización.



Construir un ensemble.

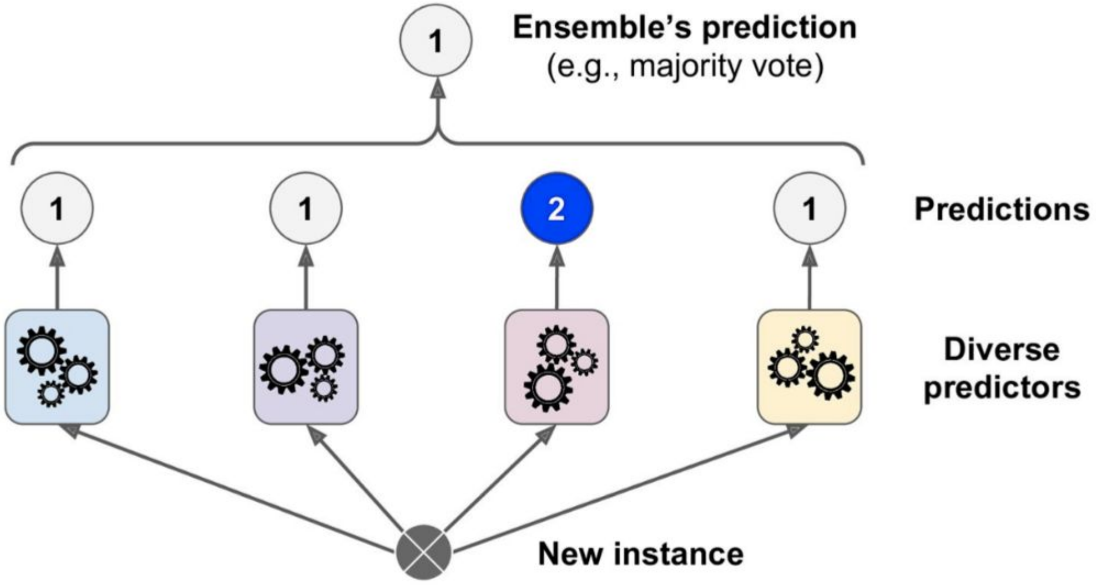
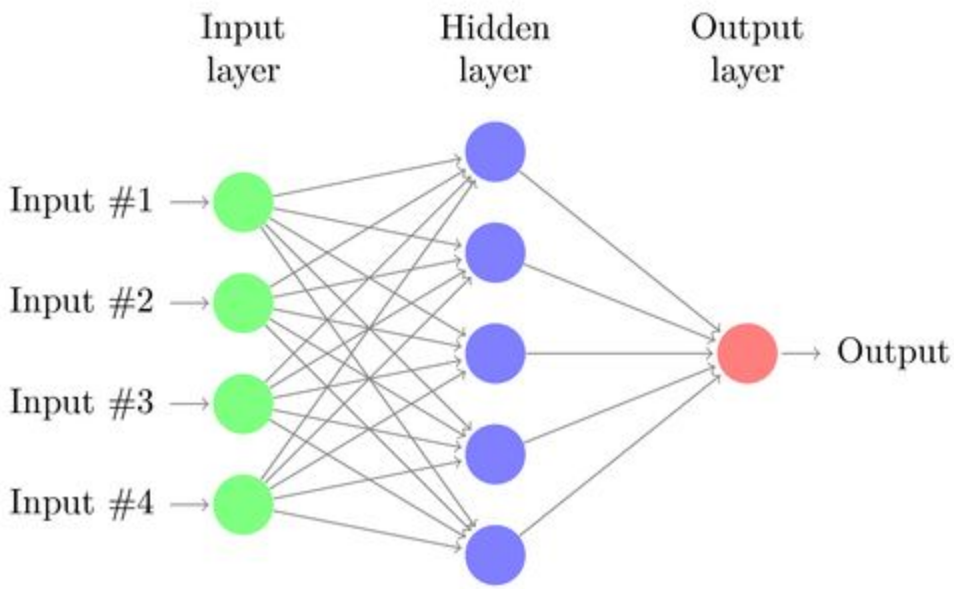
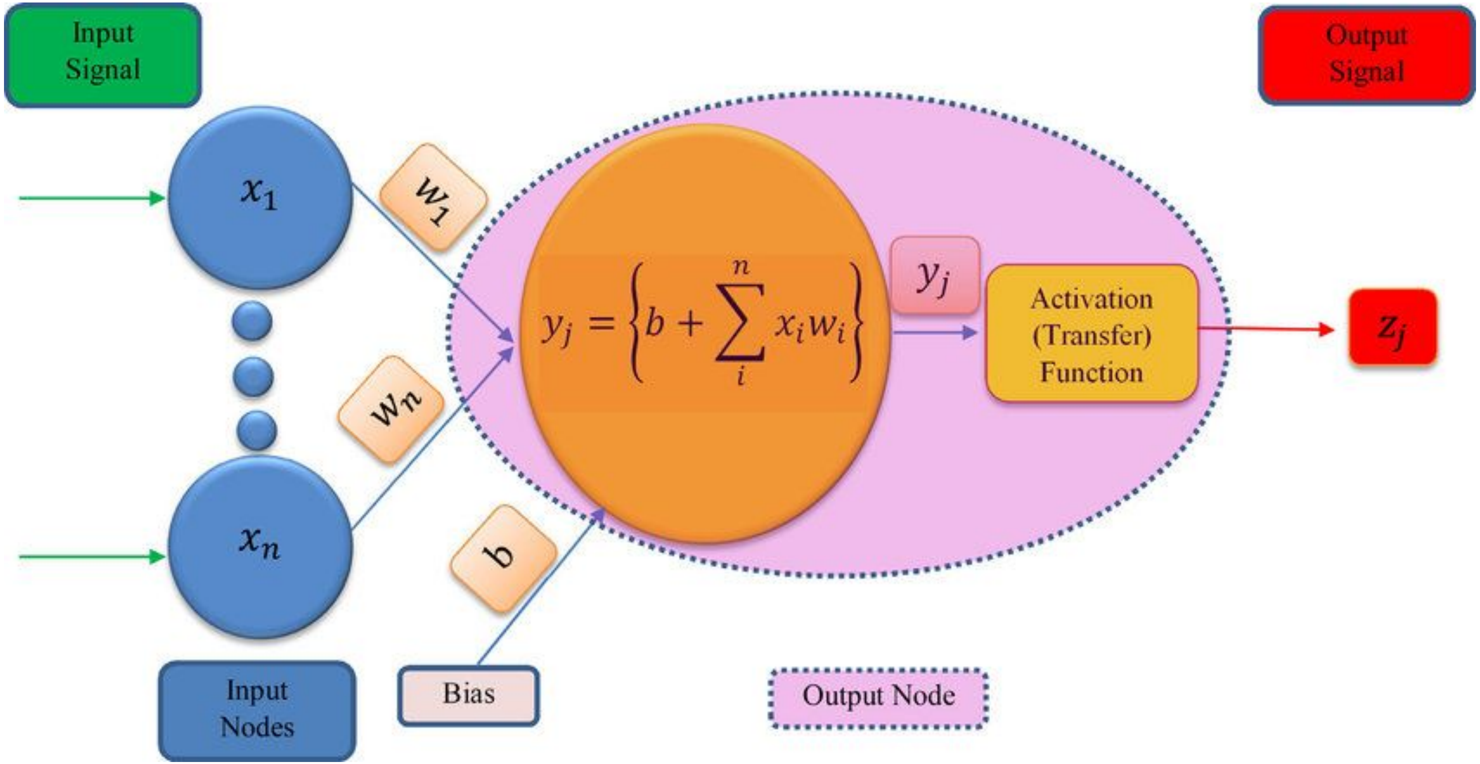


Figure 7-2. Hard voting classifier predictions

Construir un modelo ANN.



Construir un modelo ANN.



Métricas.

		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Métricas.

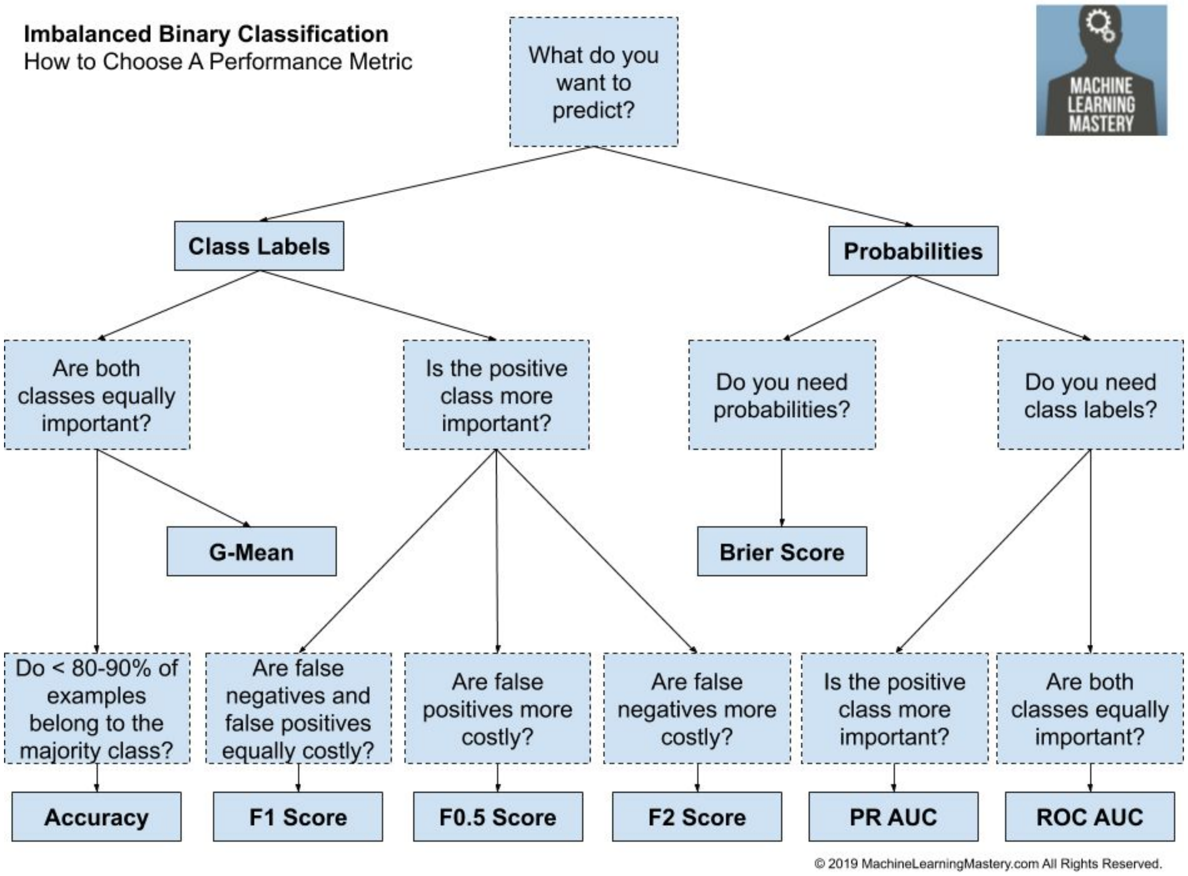
Table 3

Measures for multi-class classification based on a generalization of the measures of Table 1 for many classes C_i: tp_i are true positive for C_i, and fp_i – false positive, fn_i – false negative, and tn_i – true negative counts respectively. μ and M indices represent micro- and macro-averaging.

Measure	Formula	Evaluation focus
Average Accuracy	$\frac{\sum_{i=1}^{l} \frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$	The average per-class effectiveness of a classifier
Error Rate	$\frac{\sum_{i=1}^{l} \frac{fp_i+fn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$	The average per-class classification error
Precision$_\mu$	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i+fp_i)}$	Agreement of the data class labels with those of a classifiers if calculated from sums of per-text decisions
Recall$_\mu$	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i+fn_i)}$	Effectiveness of a classifier to identify class labels if calculated from sums of per-text decisions
Fscore$_\mu$	$\frac{(\beta^2+1)Precision_\mu Recall_\mu}{\beta^2 Precision_\mu+Recall_\mu}$	Relations between data's positive labels and those given by a classifier based on sums of per-text decisions
Precision$_M$	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fp_i}}{l}$	An average per-class agreement of the data class labels with those of a classifiers
Recall$_M$	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fn_i}}{l}$	An average per-class effectiveness of a classifier to identify class labels
Fscore$_M$	$\frac{(\beta^2+1)Precision_M Recall_M}{\beta^2 Precision_M+Recall_M}$	Relations between data's positive labels and those given by a classifier based on a per-class average

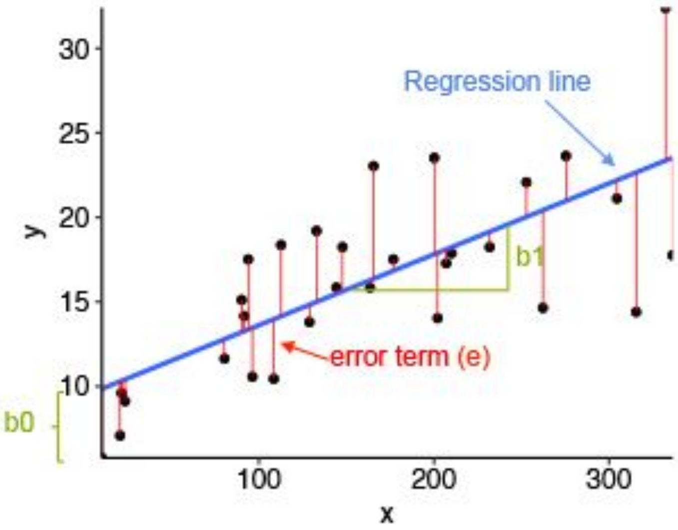
créditos Machine Learning Mastery

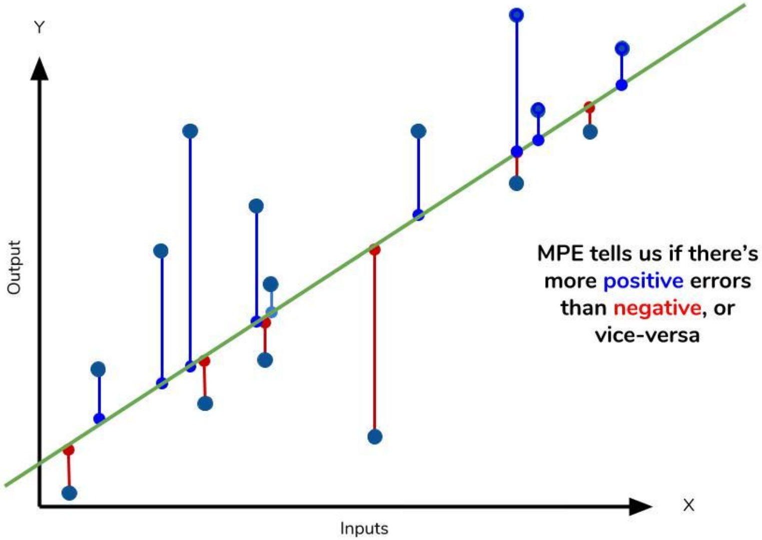
Métricas.



créditos Machine Learning Mastery

Métricas.





créditos Machine Learning Mastery


DEMO TIME.

¡Muchas Gracias!