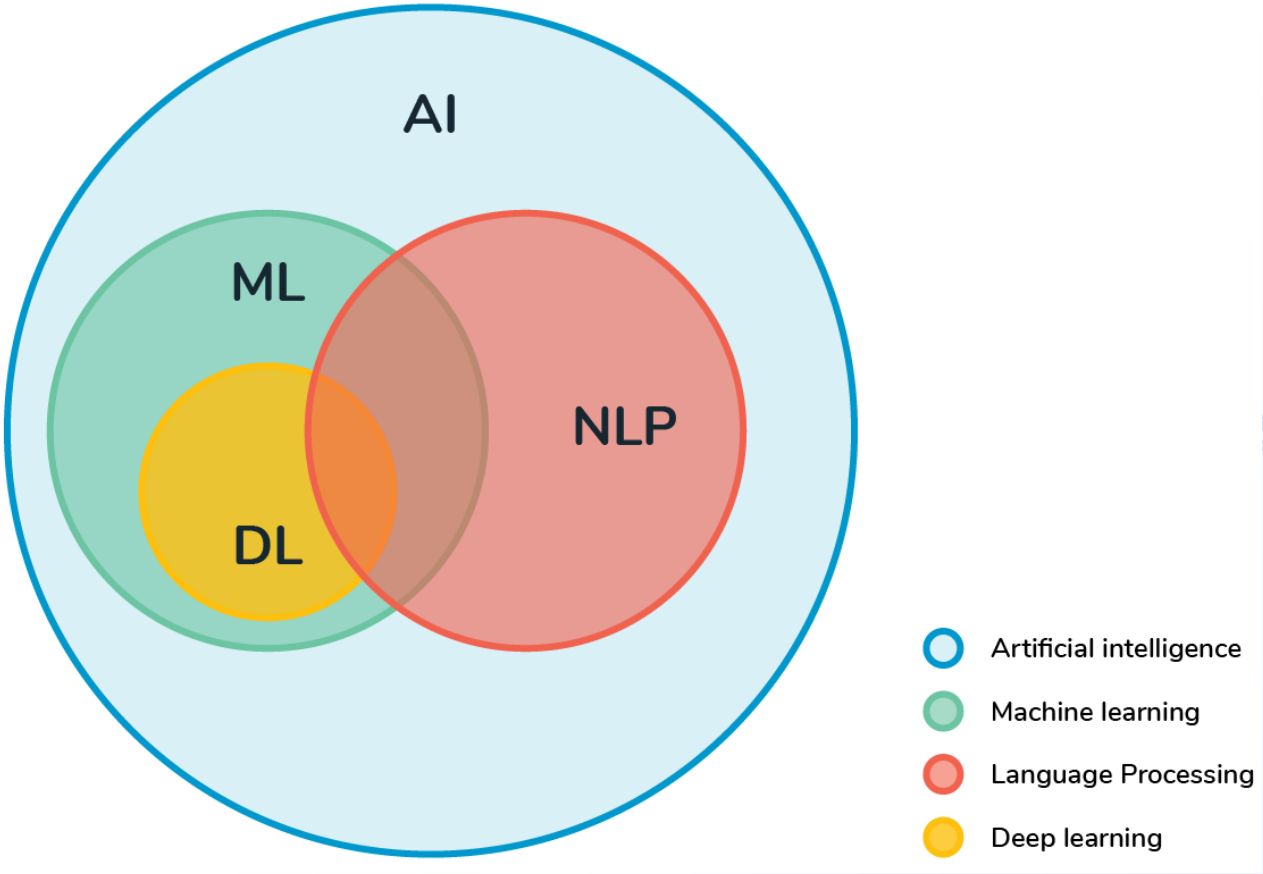
Procesamiento de Lenguaje Natural

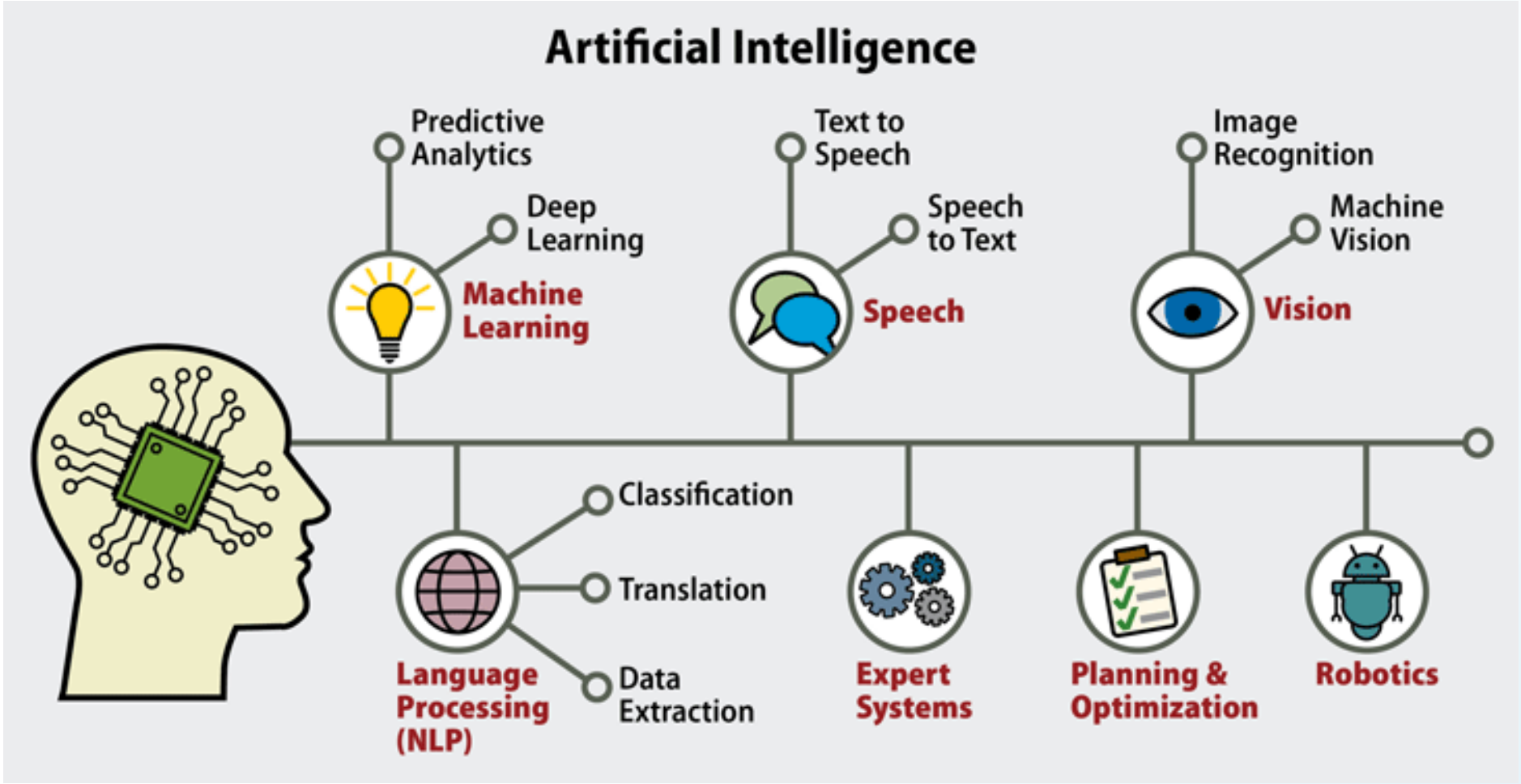
Contenido

- Conceptos de Inteligencia Artificial / Machine Learning
- Conceptos de Procesamiento de Lenguaje Natural
- Conceptos de Análisis de Sentimientos


AI
ML
NLP
DL
Artificial intelligence
Machine learning
Language Processing
Deep learning

“

La Inteligencia Artificial (IA) es la rama de las ciencias de la computación que se enfoca en brindar a las máquinas o computadoras la capacidad de pensar de manera tan inteligente como los humanos y, en algunos casos, mejor que los humanos, aprendiendo de una gran cantidad de datos.



https://www.datamation.com/artificial-intelligence/what-is-artificial-intelligence.html

Machine Learning

- El aprendizaje automático es un subcampo de la inteligencia artificial que utiliza algoritmos para aprender automáticamente cómo realizar una tarea determinada sin estar programado explícitamente con reglas.

Aplicaciones Machine Learning

Voice assistants (Alexa, Siri, Google Home etc)

Recommendation Engines

Sentiment Analysis

Face detection and Recognition

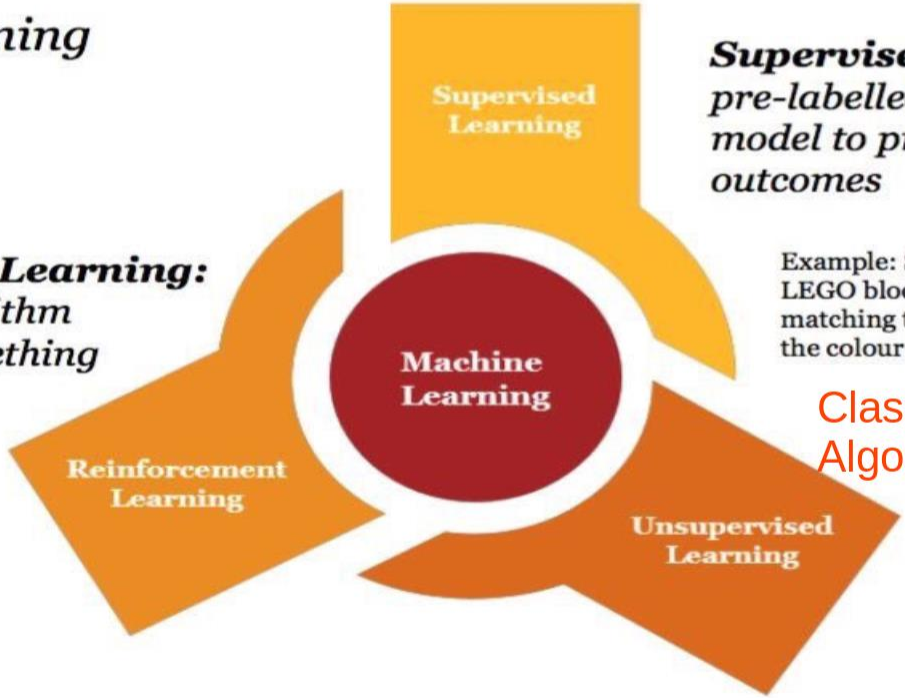
Detecting credit card Fraud

Text summarization algorithms

Machine Learning : 3 Types of Learning

Background

Types of machine learning



Supervised Learning:
pre-labelled data trains a model to predict new outcomes

Example: Sorting LEGO blocks by matching them with the colour of the bags

Classification & Regression Algorithms

Reinforcement Learning:
feedback to algorithm when it does something right or wrong

Rewards & Recommendations algorithms

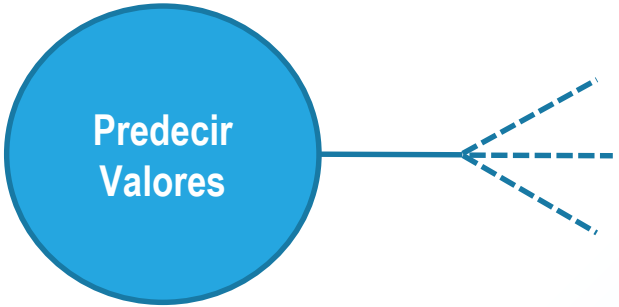
Example:
Child gets feedback 'on the job' when it does something right or wrong

Unsupervised Learning:
Non-labelled data self organises to predict new outcomes (e.g. clustering)

Clustering Algorithms

Aprendizaje Supervisado

Regresión
Predecir resultados futuros estimando relaciones entre variables



Clasificación
Identificar a qué categorías pertenecen nuevos valores

Aplicaciones

Estimar demanda de productos

Predecir gustos de clientes

Predecir ventas de una tienda

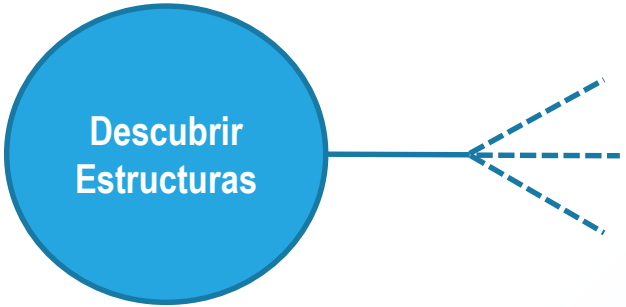
Predecir fraude en uso de Tarjeta de Crédito

Predecir fallos en dispositivos

Diagnósticos médicos

Aprendizaje No Supervisado

Clustering
Separar en grupos con características similares



Asociaciones
Detectar relaciones entre las características de una instancia

Aplicaciones

Agrupar clientes

Reconocimiento de Formas

Clasificación de Documentos

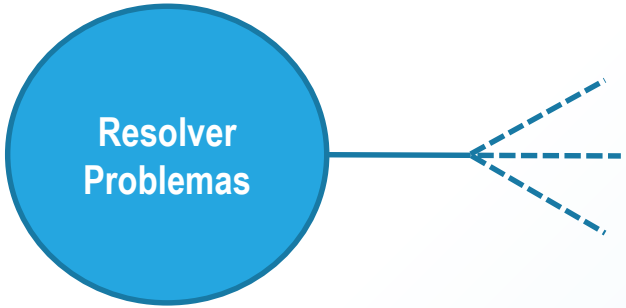
Relacionar productos de la compra

Relacionar gustos de entretenimientos de clientes

Relacionar productos contratados por cliente

Aprendizaje por Refuerzo

Reinforcement Learning
Crea la mejor estrategia para obtener la mayor recompensa



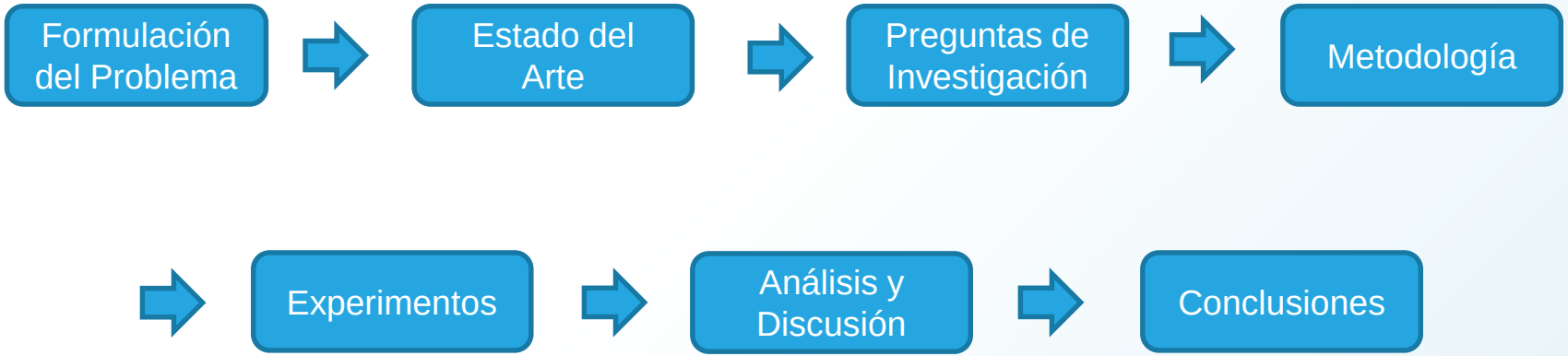
Aplicaciones

Juegos

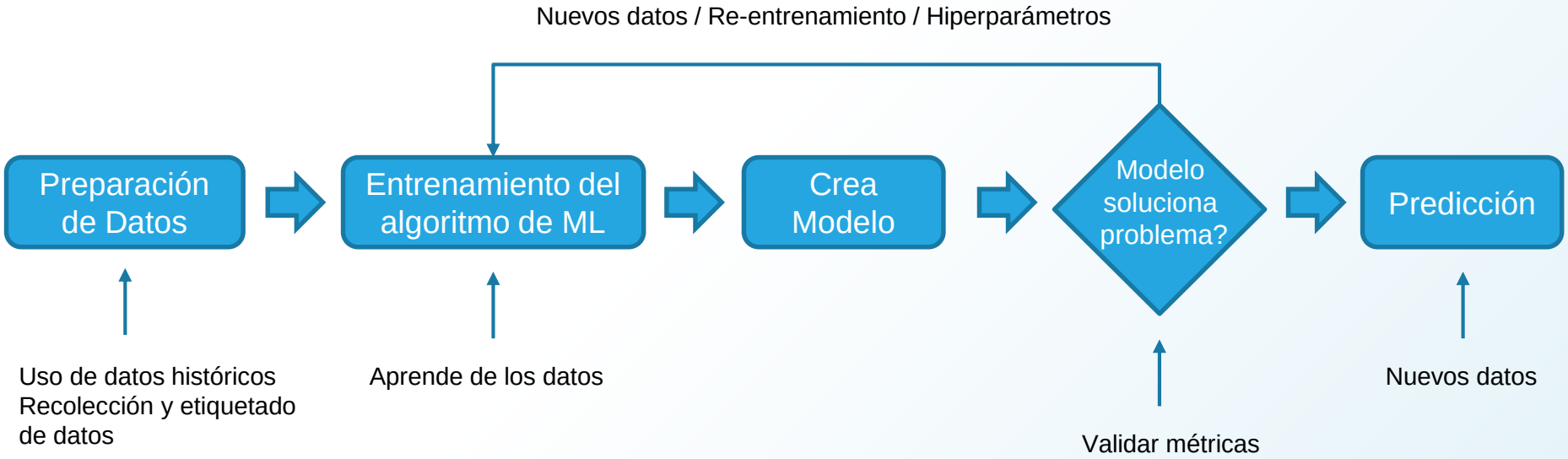
Control de Dispositivos (motores)

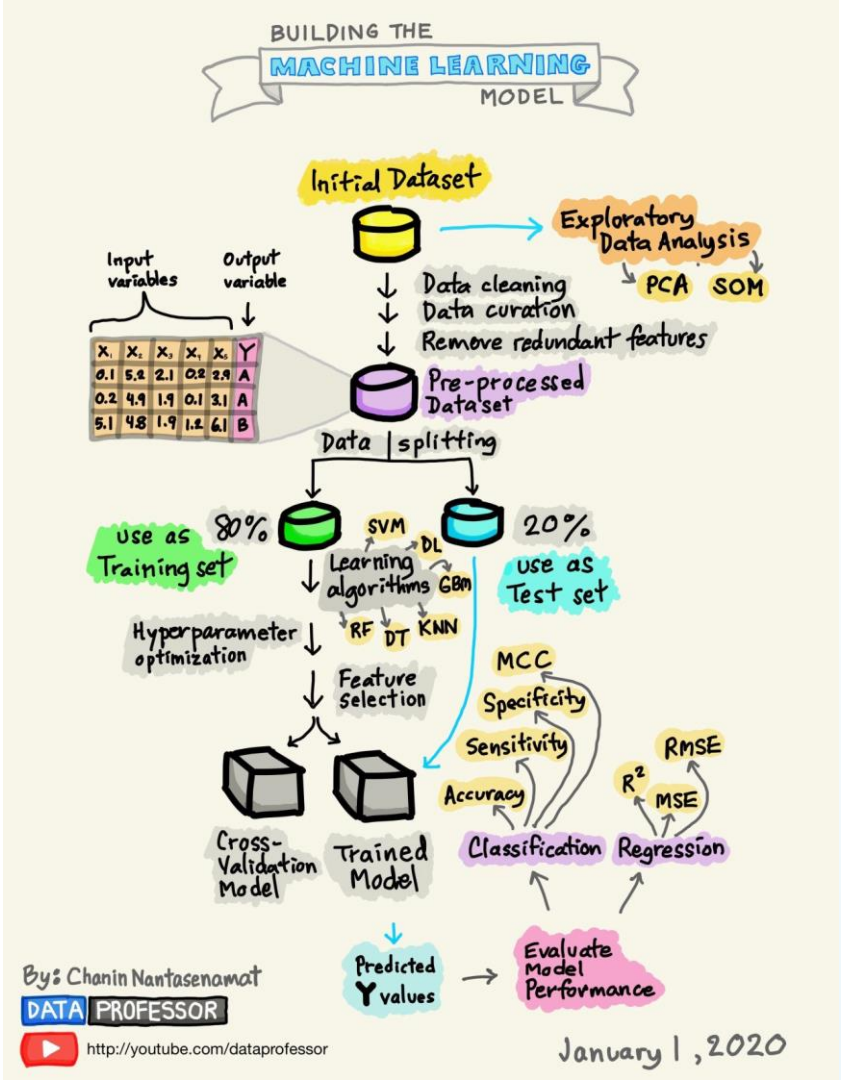
Robótica (control de movimiento)

Ciclo de Vida – Trabajo de Investigación

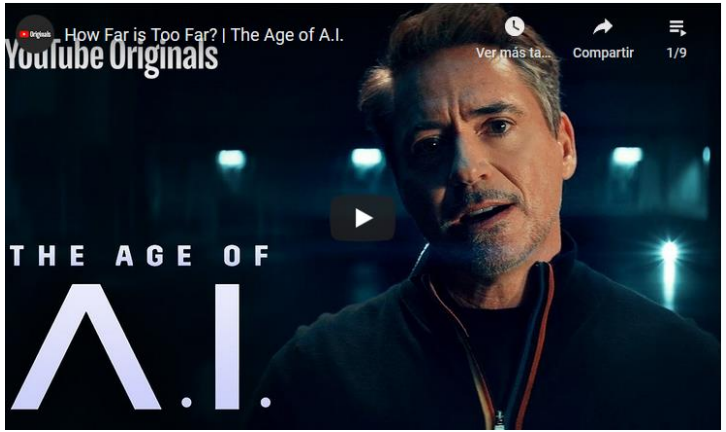


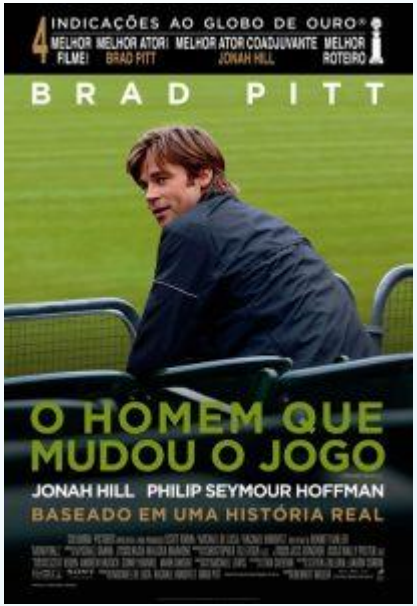
Ciclo de Vida – Experimentos en Machine Learning




BUILDING THE
MACHINE LEARNING
MODEL
Initial Dataset
Exploratory Data Analysis
PCA
SOM
Input variables
Output variable
Data cleaning
Data curation
Remove redundant features
Pre-processed Data set
Data splitting
80%
use as Training set
SVM
DL
Learning algorithms
GBM
RF
DT
KNN
20%
use as Test set
Hyperparameter optimization
Feature selection
MCC
Specificity
Sensitivity
Accuracy
RMSE
R^2
MSE
Classification
Regression
Cross-Validation Model
Trained Model
Predicted Y values
Evaluate Model Performance
By: Chanin Nantasenamat
DATA PROFESSOR
http://youtube.com/dataprofessor
January 1, 2020

Videos


How Far is Too Far? | The Age of A.I.
YouTube Originals
Ver más ta...
Compartir
1/9
THE AGE OF
A.I.


4 INDICAÇÕES AO GLOBO DE OURO
MELHOR FILME
MELHOR ATOR BRAD PITT
MELHOR ATOR COADJUVANTE JONAH HILL
MELHOR ROTEIRO
BRAD PITT
O HOMEM QUE MUDOU O JOGO
JONAH HILL PHILIP SEYMOUR HOFFMAN
BASEADO EM UMA HISTÓRIA REAL

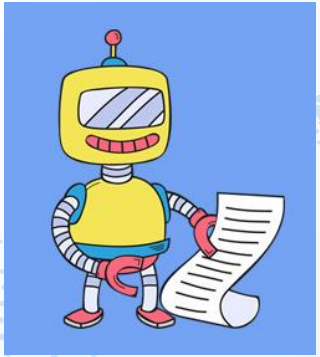

CINEMAMBIENTE 2018
AUDIENCE AWARD
"A part-scary, part-spellbinding documentary."
THE NEW YORK TIMES
FROM THE DIRECTOR OF "WHO KILLED THE ELECTRIC CAR?"
DO YOU TRUST THIS COMPUTER?

1.Definición

Qué es Procesamiento de Lenguaje Natural?

Lenguaje Natural

- Comunicación por seres humanos
- Ejemplos: Inglés, Español, Portugués
- Procesamiento de Lenguaje Natural es cualquier manipulación computacional del lenguaje natural



Tecnologías basadas en PLN

- **Predicción de texto / Correctores Ortográficos (MS Word/otros editores de texto)**
- **Búsqueda en la web (Google, Bing, Yahoo)**
- **Clasificadores de Spam (Todos los servicios de email)**
- **Traducción automática (Google Translate)**
- **Asistentes virtuales (Siri, Alexa, Cortana)**
- **Análisis de sentimientos en tweets y blogs**

https://www.youtube.com/watch?v=MZ8rJfg2mLs

Técnicas

- Segmentación de Oraciones
- Segmentación de Palabras (Tokenización)
- Limpieza de datos (stopwords)
- Análisis Lexicográfico (stemming, lematizing)
- Etiquetado Gramatical (POS)
- Chunking
- Reconocimiento de Entidades

NLTK

- Instalación:

```
import nltk
nltk.download('book')
```

- Carga de textos:

```
from nltk.book import *
text9
```

```
<Text: The Man Who Was Thursday by G . K . Chesterton 1908>
```

https://www.nltk.org/book/ch01.html

Contador de Vocabulario

```
print('Cantiddad de símbolos:',len(text3))
```

```
Cantiddad de símbolos: 44764
```

```
print('Cantiddad de palabras',len(set(text3)))
```

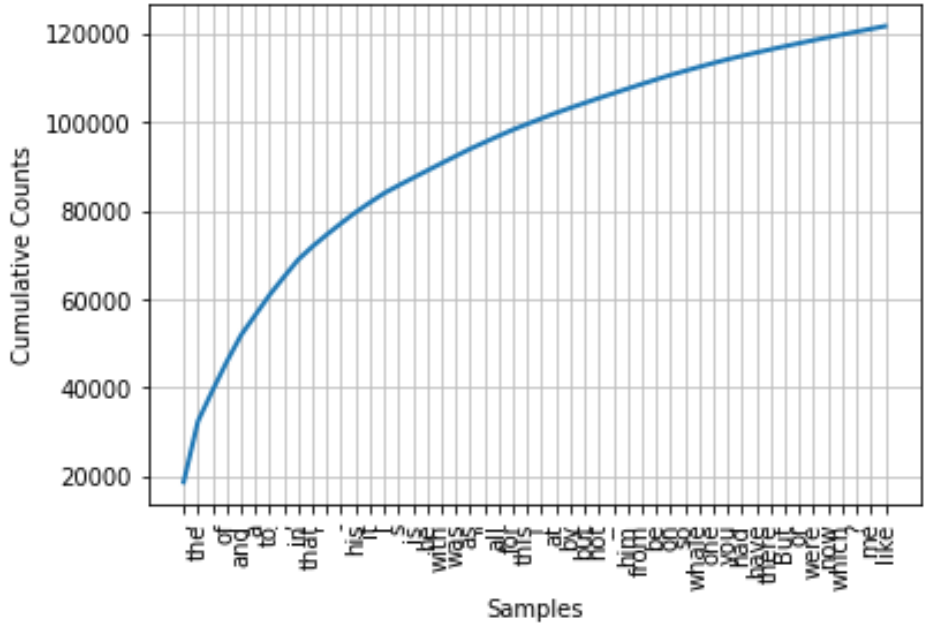
```
Cantiddad de palabras 2789
```

```
text3.count("smote")
```

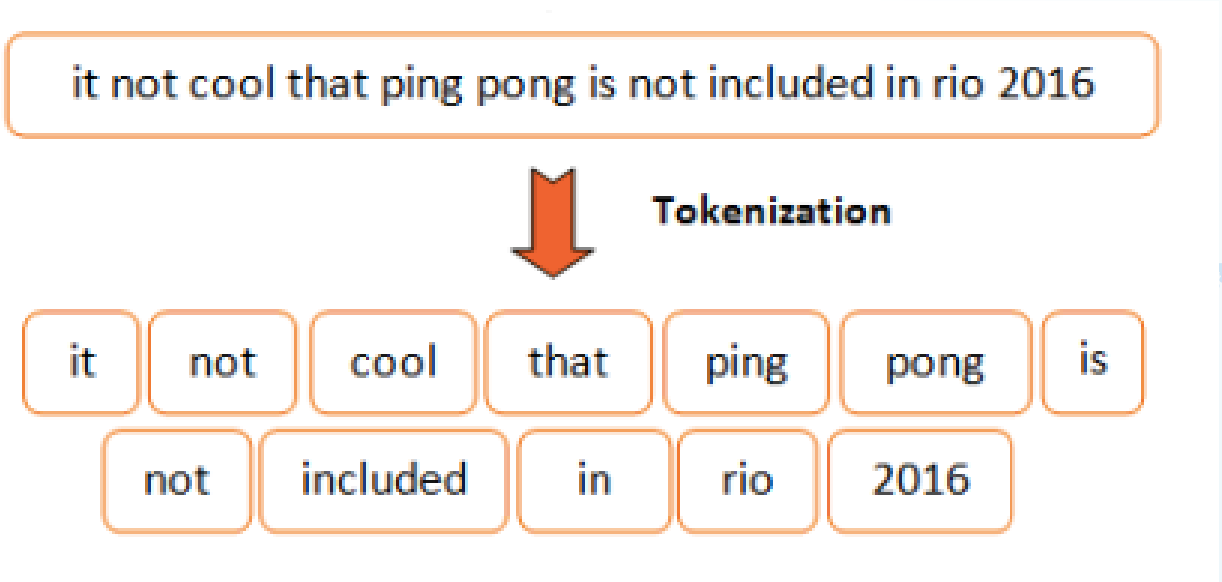
```
5
```

Distribución de Frequencia

```
fdist = FreqDist(text1)
fdist.most_common(50)
fdist.plot(50, cumulative=True)
```



Tokenización



Tokenization

```
text = word_tokenize("And now for something completely different")
print(nltk.pos_tag(text))
```

```
Tokens: ['And', 'now', 'for', 'something', 'completely', 'different']
```

Stopwords y Signos de Puntuación

Stop Words: remover palabras comunes pero que no proveen utilidad al descubrimiento del contexto (el, la, de, los, y, etc...)

StopWords

```
from nltk.corpus import stopwords

def removeStopword(texto):
    stopwordSpanish = stopwords.words('spanish')
    textNew= [w.lower() for w in texto if w not in stopwordSpanish]
    print(textNew)
```

```
['A', 'punto', 'de', 'cumplirse', 'una', 'semana', 'desde', 'que', 'Nicolás', 'Maduro', 'pusiera',
'en', 'marcha', 'las', 'nuevas', 'medidas', 'económicas', 'para', 'Venezuela', ',']
['a', 'punto', 'cumplirse', 'semana', 'nicolás', 'maduro', 'pusiera', 'marcha', 'nuevas',
'medidas', 'económicas', 'venezuela', ',', 'valor', 'bolívar', ',', 'fijado', 'tasa', '60',
'unidades']
```

Stemming - Lemmatization

campo --> camp
casita --> cas
vendedor --> vend
panadería --> pan

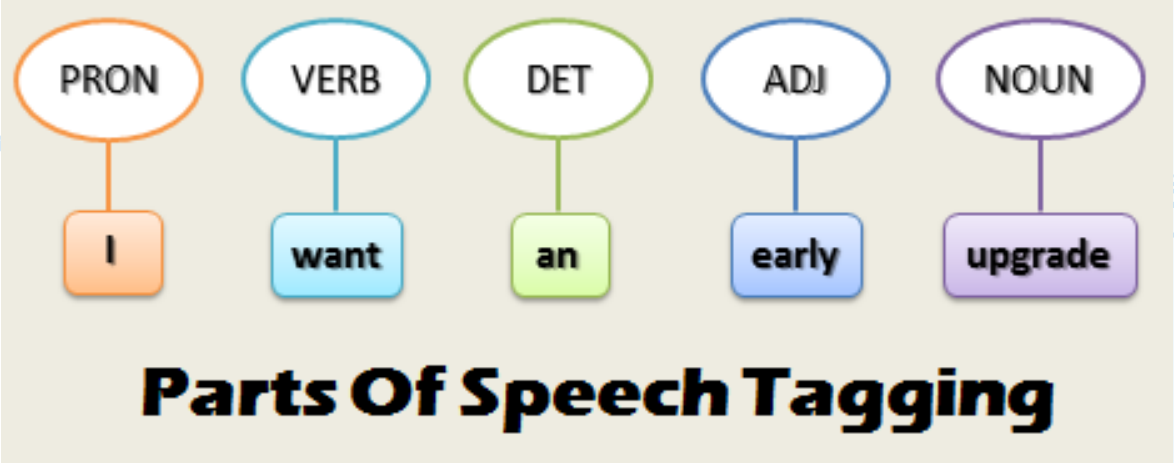
comiendo --> comer
limones --> limón
corruptas --> corruptos
nueces --> nuez

Stemming y Lemmatization

```
tokens = word_tokenize(raw)
porter = nltk.PorterStemmer()
lancaster = nltk.LancasterStemmer()
print('PorterStemmer',[porter.stem(t) for t in tokens])
print('LancasterStemmer',[lancaster.stem(t) for t in tokens])
wnl = nltk.WordNetLemmatizer()
print('WordNetLematizer',[wnl.lemmatize(t) for t in tokens])
```

```
Porter Stemmer ['la', 'familia', 'suelen', 'escond', 'a', 'lo', 'niño', 'afectado', 'o', 'lo', 'áislan', 'con', 'lo', 'animal']
Lancaster Stemmer ['las', 'familia', 'suel', 'escond', 'a', 'los', 'niño', 'afectado', 'o', 'los', 'aísl', 'con', 'los', 'anim']
Lemmatization ['Las', 'familias', 'suelen', 'esconder', 'a', 'los', 'niños', 'afectados', 'o', 'los', 'áislan', 'con', 'los', 'animales']
```

Etiquetado Gramatical Tagging Words



Etiquetador (tagger)

```
text = word_tokenize("And now for something completely different")
nltk.pos_tag(text)
```

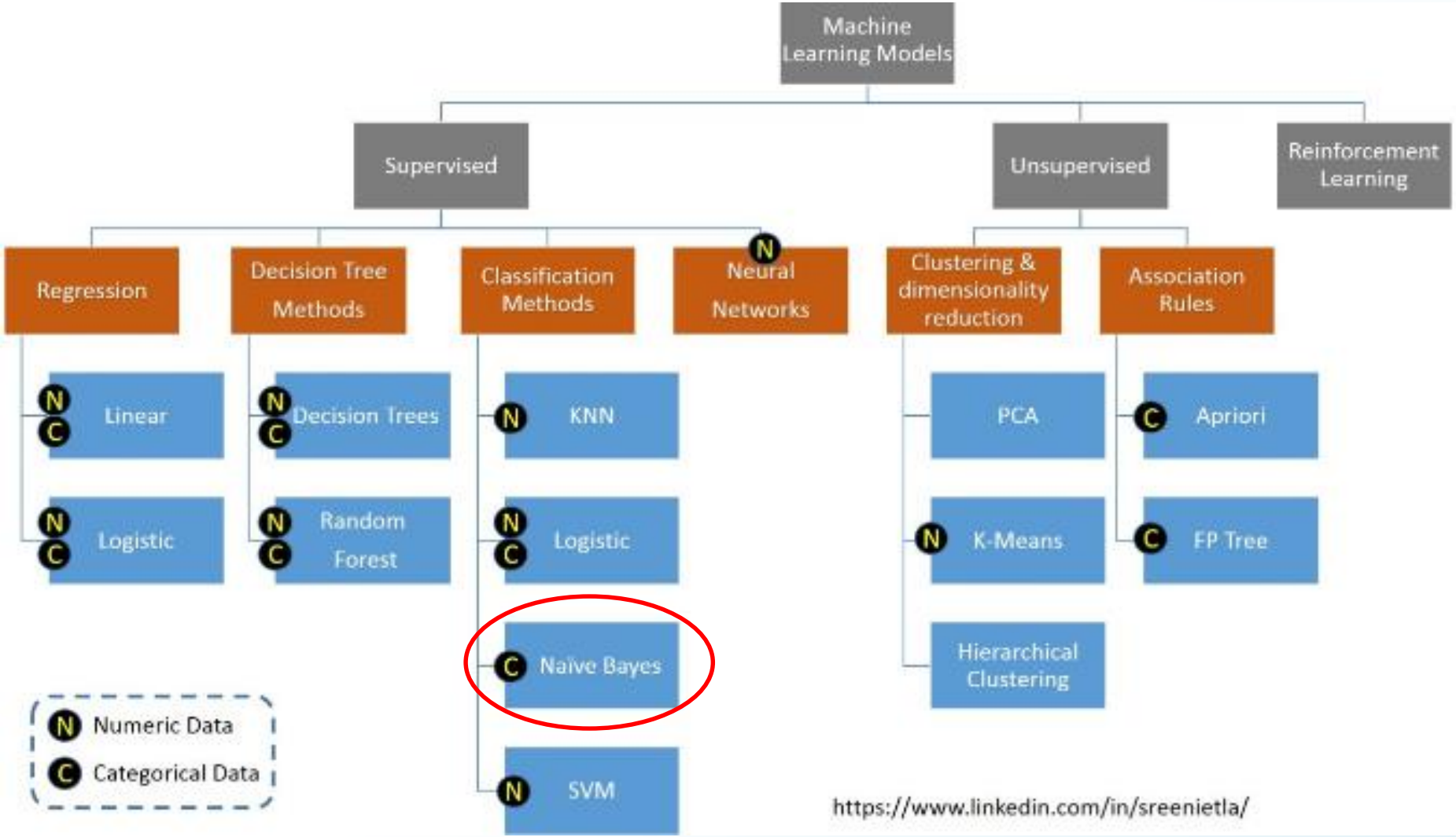
```
[('And', 'CC'),
 ('now', 'RB'),
 ('for', 'IN'),
 ('something', 'NN'),
 ('completely', 'RB'),
 ('different', 'JJ')]
```

```
nltk.corpus.brown.tagged_words(tagset='universal')
```

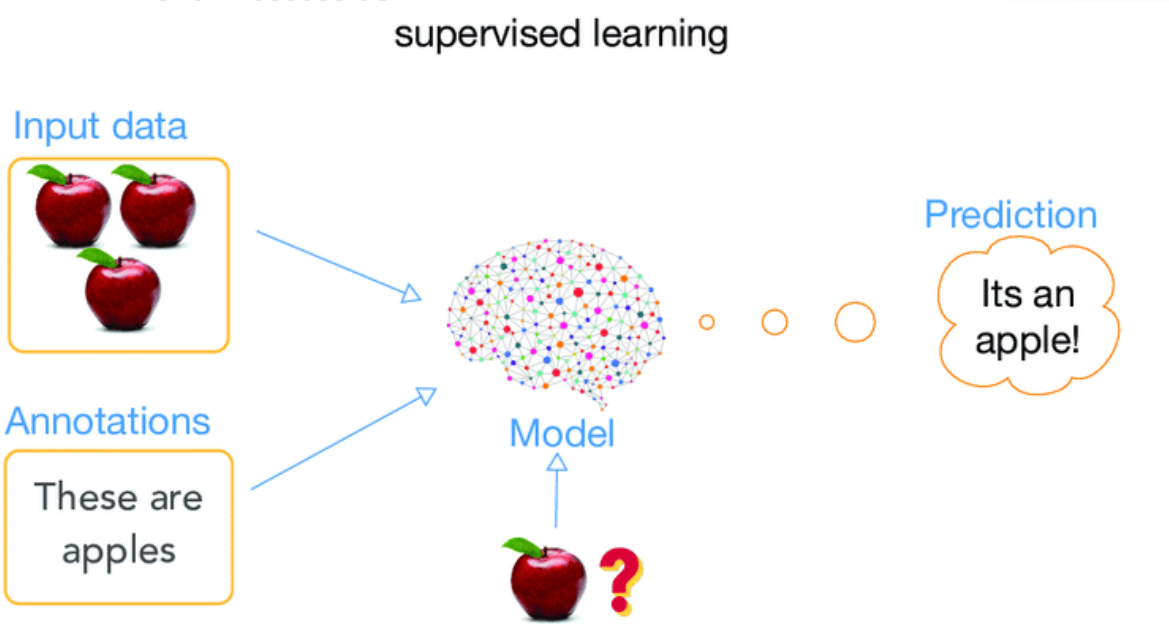
```
[('The', 'DET'), ('Fulton', 'NOUN'), ...]
```

2. Clasificación Supervisada

Y su uso para el Análisis de Sentimientos


Machine Learning Models
Supervised
Unsupervised
Reinforcement Learning
Regression
Decision Tree Methods
Classification Methods
Neural Networks
Clustering & dimensionality reduction
Association Rules
N C Linear
N C Logistic
N C Decision Trees
N C Random Forest
N KNN
N C Logistic
C Naive Bayes
N SVM
PCA
N K-Means
Hierarchical Clustering
C Apriori
C FP Tree
N Numeric Data
C Categorical Data
https://www.linkedin.com/in/sreenietla/

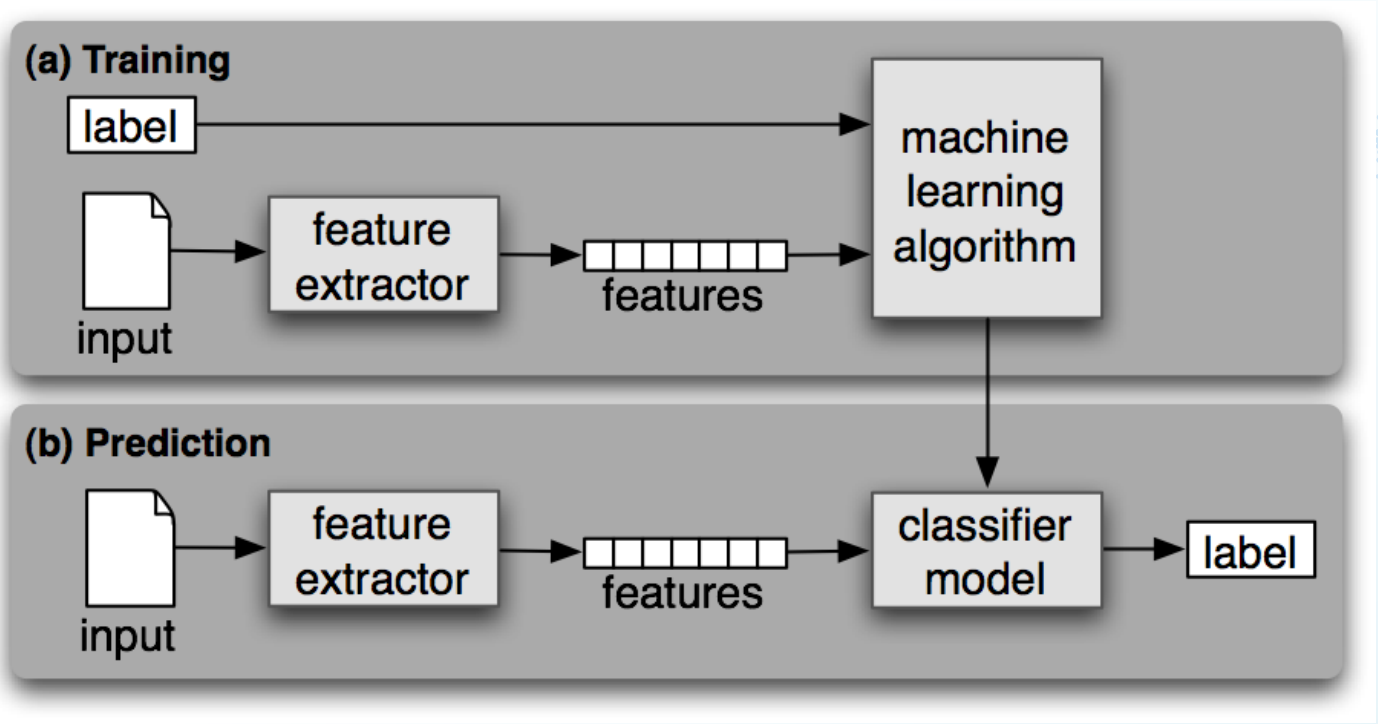
Aprendizaje Automático
Clasificación Supervisada



Análisis de Sentimientos

- This is a good book! Postive
- This is a awesome book! Postive
- This is a bad book! Negative
- This is a terrible book Negative

Aprendizaje Automático
Clasificación Supervisada



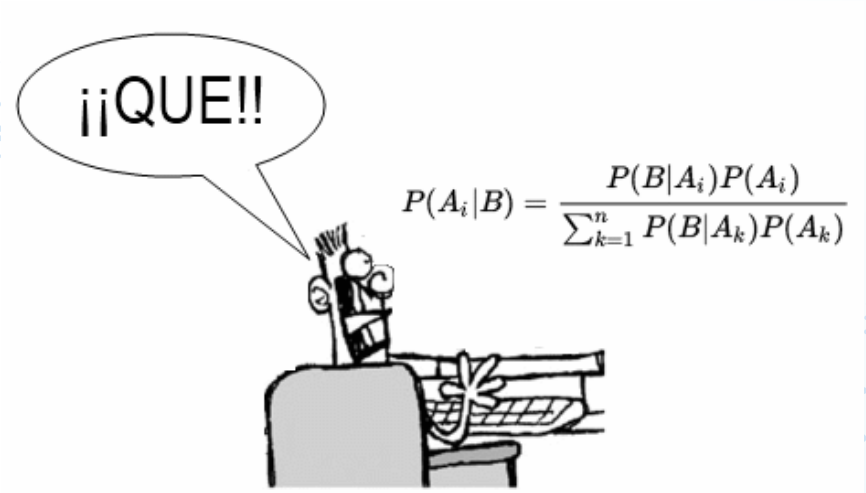
Aprendizaje Automático
Clasificación Supervisada

Características (features)

Técnicas

- Tokenización
- Lematización
- POS
- NER

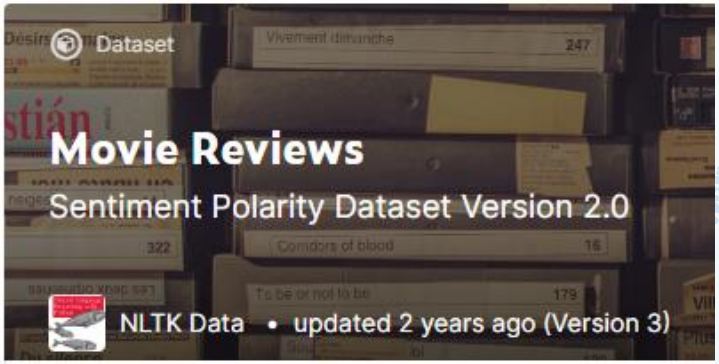
Naive Bayes



https://www.docirs.cl/entender_teorema_de_bayes_simple.asp

Movie Reviews

- 1000 archivos etiquetados como positivos y otros 1000 etiquetados como negativos.



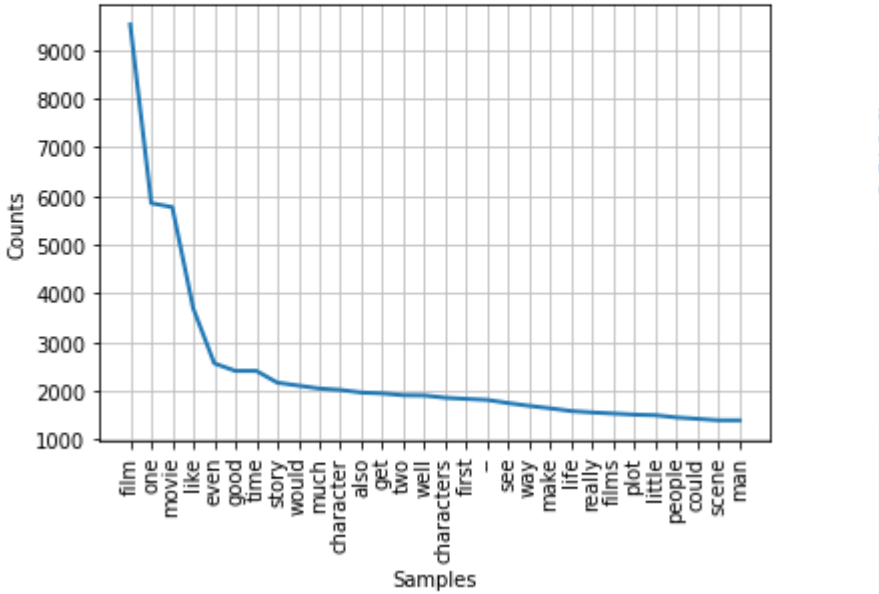
Movie Reviews

```
def Estadisticas():

    print ('total',len(movie_reviews.fileids()))
    print ('categorias',movie_reviews.categories())
    print ('total positivos',len(movie_reviews.fileids('pos')))
    print ('total negativos',len(movie_reviews.fileids('neg')))

    all_words = [word.lower() for word in movie_reviews.words()]
    all_words_frequency = FreqDist(all_words)
    print ('10 palabras más frecuentes',all_words_frequency.most_common(10))
    print ('cantidad de veces que se repite la palabra happy',all_words_frequency['happy'])
```

```
total 2000
categorias ['neg', 'pos']
total positivos 1000
total negativos 1000
10 palabras más frecuentes [(',', 77717), ('the', 76529), ('.', 65876), ('a', 38106),
('and', 35576), ('of', 34123), ('to', 31937), ("'", 30585), ('is', 25195), ('in', 21822)]
cantidad de veces que se repite la palabra happy 215
```

Movie Reviews

```
total positivos 1000
total negativos 1000
1583820
710578
10 palabras más frecuentes [('film', 9517), ('one', 5852),
cantidad de veces que se repite la palabra happy 215
```



Bag of Words Feature

- **B**ag **o**f **W**ords method converts text data into numbers.
- It does this by
  - Creating a **vocabulary** from the words in all documents
  - Calculating the **occurrences** of words:
    - **binary** (present or not)
    - **word counts**
    - **frequencies**

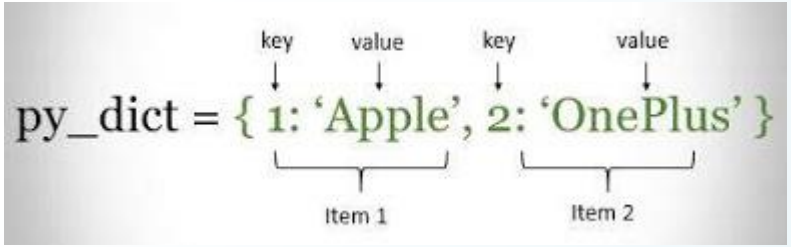
Bag of Words Feature

Simple example using **word counts**:

	a	cat	dog	happy	is	it	my	not	old	wolf
"It is a dog."	1	0	1	0	1	1	0	0	0	0
"my cat is old"	0	1	0	0	1	0	1	0	1	0
"It is not a dog, it a is wolf."	2	0	1	0	2	2	0	1	0	1

Bag of Words Feature

```
def bag_of_words(words):
    words_clean = []
    stopwords_english = stopwords.words('english')

    for word in words:
        word = word.lower()
        if word not in stopwords_english and word not in string.punctuation:
            words_clean.append(word)

    words_dictionary = dict([word, True] for word in words_clean)

    return words_dictionary
```



Bag of Words Feature

```
def DatosBOW():
    pos_reviews = []
    for fileid in movie_reviews.fileids('pos'):
        words = movie_reviews.words(fileid)
        pos_reviews.append(words)

    neg_reviews = []
    for fileid in movie_reviews.fileids('neg'):
        words = movie_reviews.words(fileid)
        neg_reviews.append(words)

    pos_reviews_set = []
    for words in pos_reviews:
        pos_reviews_set.append((bag_of_words(words), 'pos'))

    neg_reviews_set = []
    for words in neg_reviews:
        neg_reviews_set.append((bag_of_words(words), 'neg'))

    return pos_reviews_set,neg_reviews_set
```

```
POSITIVO [({'plot': True, 'two':
True, 'teen': True, 'couples':
True, 'go': True, 'church': True,
'party': True, 'drink': True,
'drive': True, 'get': True,
'accident': True, 'one': True,
'guys': True, 'dies': True,
'girlfriend': True, 'continues':
True, 'see': True, 'life': True,
'nightmares': True, 'deal': True,
'watch': True, 'movie': True,
'sorta': True, 'find': True,
'critique': True, 'mind': True,
'fuck': True, 'generation': True,
'touches': True, 'cool': True,
'idea': True, 'presents': True,
```

```
NEGATIVO [({'plot': True, 'two':
True, 'teen': True, 'couples':
True, 'go': True, 'church': True,
'party': True, 'drink': True,
'drive': True, 'get': True,
'accident': True, 'one': True,
'guys': True, 'dies': True,
'girlfriend': True, 'continues':
True, 'see': True, 'life': True,
'nightmares': True, 'deal': True,
'watch': True, 'movie': True,
'sorta': True, 'find': True,
'critique': True, 'mind': True,
'fuck': True, 'generation': True,
```

```
def clasificadorBOW(pos_reviews_set,neg_reviews_set):
    size = int(len(pos_reviews_set) * 0.1)
    test_set = pos_reviews_set[:size] + neg_reviews_set[:size]
    train_set = pos_reviews_set[size:] + neg_reviews_set[size:]
    print(len(test_set),  len(train_set))
    classifier = NaiveBayesClassifier.train(train_set)
    accuracy = classify.accuracy(classifier, test_set)
    print(accuracy)
    print (classifier.show_most_informative_features(10))
    return classifier
```

```python
def pruebaBOW(custom_review,classifier):
    custom_review_tokens = word_tokenize(custom_review)
    custom_review_set = bag_of_words(custom_review_tokens)
    print (custom_review)
    print ('positivo o negativo?',classifier.classify(custom_review_set))
    prob_result = classifier.prob_classify(custom_review_set)
    print ('probabilidad para negativo',prob_result.prob("neg"))
    print ('probabilidad para positivo',prob_result.prob("pos"))
```

```
I hated the film. It was a disaster. Poor direction, bad acting.
positivo o negativo? neg
probabilidad para negativo 0.8389515850310557
probabilidad para positivo 0.16104841496894456
It was a wonderful and amazing movie. I loved it. Best direction, good acting.
positivo o negativo? pos
probabilidad para negativo 0.06519158529235963
probabilidad para positivo 0.9348084147076396
```

Matriz de Confusión

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$ (sensitivity)

$$F\text{-}score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)}$$

$$Specificity = \frac{TN}{TN + FP}$$

https://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/

4. Herramientas Actuales

Qué puedo usar?

Herramientas actuales

ARTIFICIAL INTELLIGENCE

Amazon Transcribe

Automatic Speech Recognition (ASR) powered by deep learning

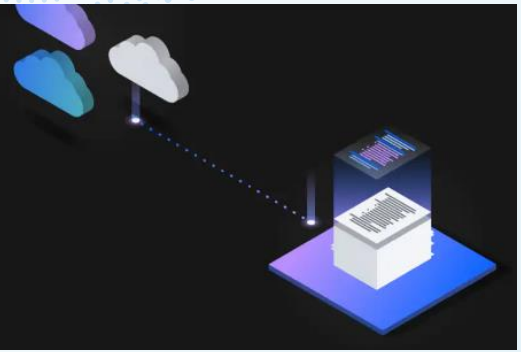
Amazon Transcribe provides quality and affordable speech recognition for efficient speech to text transcription.

Watson Natural Language Understanding

The natural language processing (NLP) service for advanced text analytics

Get started free

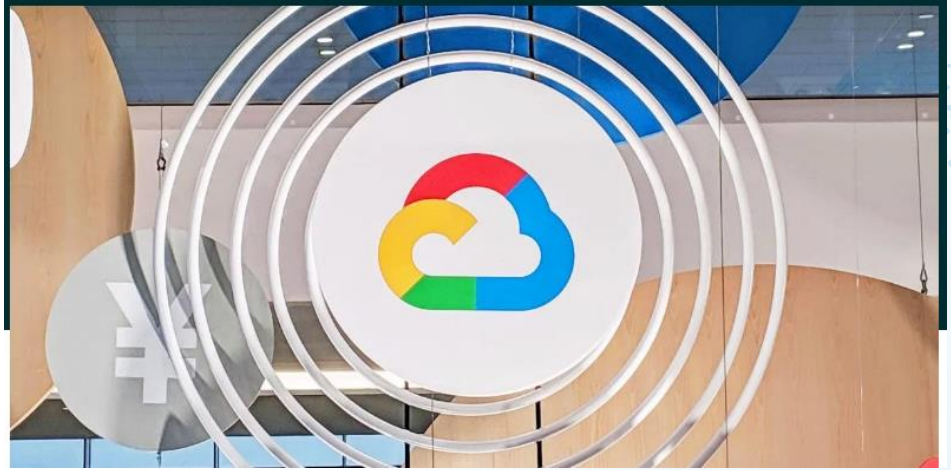
View demo



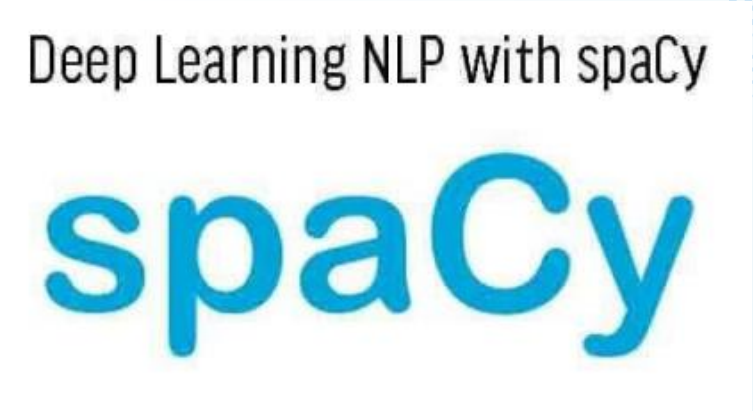
Herramientas actuales

Google launches AutoML Natural Language with improved text classification and model training

Kyle Wiggers @Kyle_L_Wiggers December 12, 2019 9:56 AM AI



Herramientas actuales



Herramientas actuales

Herramientas actuales

Gracias!

Alguna pregunta?

- Maria.limaylla@gmail.com