TOOLS BASICS

* import pandas as pd
* from pandas import DataFrame
* from datetime import datetime, timedelta, date
* from pandas.plotting import autocorrelation\_plot
* from pandas import read\_csv
* from matplotlib import pyplot as plt
* import warnings;
* warnings.filterwarnings('ignore')
* import seaborn as sns

TOOLS DE NLP

* import re
* import nltk
* nltk.download('punkt')
* nltk.download('wordnet')
* nltk.download('stopwords')
* nltk.download('averaged\_perceptron\_tagger')
* from nltk.corpus import stopwords
* from nltk.tokenize import word\_tokenize, sent\_tokenize

# ANALITICA DE TEXTO

* from wordcloud import WordCloud
* from wordcloud import STOPWORDS

# ANALITICA DE TEXTO

* from wordcloud import WordCloud
* from wordcloud import STOPWORDS

#Statistical LTV

* from lifetimes import BetaGeoFitter, GammaGammaFitter
* from lifetimes.utils import calibration\_and\_holdout\_data, summary\_data\_from\_transaction\_data
* from sklearn.model\_selection import train\_test\_split, RandomizedSearchCV
* from sklearn.metrics import accuracy\_score
* from sklearn.ensemble import RandomForestRegressor
* from sklearn.svm import LinearSVC
* from sklearn import preprocessing
* from kmodes.kprototypes import KPrototypes
* from pprint import pprint
* import numpy as np
* import plotly.graph\_objects as go

TOOLS ALGORITMO 2

# ML approach to LTV

* import tensorflow as tf
* #import tensorflow\_probability as tfp
* from tensorflow import keras
* from tensorflow.keras import layers
* import tensorflow\_docs as tfdocs
* import tensorflow\_docs.modeling as tfmodel
* import tensorflow\_docs.plots

# Ploteo

* import matplotlib
* matplotlib.use('TkAgg')
* from sklearn.model\_selection import train\_test\_split
* from sklearn.preprocessing import StandardScaler

# Evaluación

* from sklearn.metrics import r2\_score
* from sklearn.metrics import mean\_absolute\_error
* print(f'TensorFlow Version: {tf.\_\_version\_\_}')
* print(f'Keras version: {keras.\_\_version\_\_}')
* print('Physical devices:')
* tf.config.list\_physical\_devices()

%load\_ext tensorboard

!rm -rf ./logs/

#set the directory to save fits:

import datetime

log\_dir = "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")

#define the callback:

tensorboard\_callback = tf.keras.callbacks.TensorBoard(log\_dir=log\_dir, histogram\_freq=1)

#start tensorboard:

%tensorboard --logdir logs/fit