data\_cleaning.r

kenim

2023-05-17

# DATA CLEANING ----  
# Setting working directory  
setwd(  
"C:/Users/kenim/Desktop/School Materials/LUBS5990M - Machine Learning in Practice/Project/Code")  
  
# display numbers in natural form  
options(scipen = 999)  
  
# load libraries  
library("tidyverse")

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 0.3.5  
## ✔ tibble 3.1.8 ✔ dplyr 1.1.0  
## ✔ tidyr 1.2.1 ✔ stringr 1.5.0  
## ✔ readr 2.1.3 ✔ forcats 0.5.2

## Warning: package 'stringr' was built under R version 4.2.3

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library("dplyr")  
library("VIM")

## Loading required package: colorspace  
## Loading required package: grid  
## VIM is ready to use.  
##   
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues  
##   
## Attaching package: 'VIM'  
##   
## The following object is masked from 'package:datasets':  
##   
## sleep

library(mice)

##   
## Attaching package: 'mice'  
##   
## The following object is masked from 'package:stats':  
##   
## filter  
##   
## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(ggplot2)  
  
# Loading data  
data <- read.csv("Data/data.csv")  
  
# Data structure  
str(data)

## 'data.frame': 2767 obs. of 16 variables:  
## $ ID : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ success : chr "N" "N" "N" "Y" ...  
## $ brandSlogan : chr "Is One of Its Kind ERC-20 Decentralized Stable Asset" "The Ultimate Blockchain Gaming Platform" "Simple Automated Investment App Driven by AI & ML" "International Real Estate Crowdfunding Platform" ...  
## $ hasVideo : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ rating : num 4 4.3 4.4 4.3 4.3 4.7 4.1 4.5 4.8 4.2 ...  
## $ priceUSD : num 30 0.13 0.01 NA 0.03 0.1 0.02 2.8 50 0.1 ...  
## $ countryRegion : chr "Singapore" "Malta" "UK" "Netherlands" ...  
## $ startDate : chr "01/10/2019" "07/09/2018" "01/07/2019" "01/10/2019" ...  
## $ endDate : chr "01/10/2019" "12/10/2018" "30/06/2020" "15/12/2019" ...  
## $ teamSize : int 31 20 10 27 14 43 20 31 8 29 ...  
## $ hasGithub : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ hasReddit : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ platform : chr "Ethereum" "XAYA" "Stellar" "Separate blockchain" ...  
## $ coinNum : num 509999 225000000 5000000000 125000000 5000000000 ...  
## $ minInvestment : int 0 1 1 1 1 1 1 1 1 1 ...  
## $ distributedPercentage: num 0.49 0.41 0.4 0.13 0.5 0.5 0.25 0.1 0.05 0.15 ...

# summary of dataset  
summary(data)

## ID success brandSlogan hasVideo   
## Min. : 1.0 Length:2767 Length:2767 Min. :0.0000   
## 1st Qu.: 692.5 Class :character Class :character 1st Qu.:0.0000   
## Median :1384.0 Mode :character Mode :character Median :1.0000   
## Mean :1384.0 Mean :0.7261   
## 3rd Qu.:2075.5 3rd Qu.:1.0000   
## Max. :2767.0 Max. :1.0000   
##   
## rating priceUSD countryRegion startDate   
## Min. :1.000 Min. : 0.00 Length:2767 Length:2767   
## 1st Qu.:2.600 1st Qu.: 0.04 Class :character Class :character   
## Median :3.100 Median : 0.12 Mode :character Mode :character   
## Mean :3.121 Mean : 19.01   
## 3rd Qu.:3.700 3rd Qu.: 0.50   
## Max. :4.800 Max. :39384.00   
## NA's :180   
## endDate teamSize hasGithub hasReddit   
## Length:2767 Min. : 1.00 Min. :0.0000 Min. :0.0000   
## Class :character 1st Qu.: 7.00 1st Qu.:0.0000 1st Qu.:0.0000   
## Mode :character Median :12.00 Median :1.0000 Median :1.0000   
## Mean :13.11 Mean :0.5779 Mean :0.6328   
## 3rd Qu.:17.00 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :75.00 Max. :1.0000 Max. :1.0000   
## NA's :154   
## platform coinNum minInvestment   
## Length:2767 Min. : 12 Min. :0.0000   
## Class :character 1st Qu.: 50000000 1st Qu.:0.0000   
## Mode :character Median : 180000000 Median :0.0000   
## Mean : 8177879989180 Mean :0.4532   
## 3rd Qu.: 600000000 3rd Qu.:1.0000   
## Max. :22619078416800000 Max. :1.0000   
##   
## distributedPercentage  
## Min. : 0.000   
## 1st Qu.: 0.400   
## Median : 0.550   
## Mean : 1.061   
## 3rd Qu.: 0.700   
## Max. :869.750   
##

# save in dataframe  
df <- data  
  
# IDENTIFY DUPLICATED VALUES ----  
duplicated\_row <- df[duplicated(df),]  
  
# COLUMN CLEANING ----  
## ID COLUMN ----  
### dropping ID column ----  
df <- df %>%  
 select(-ID)  
  
## Success column ----  
unique(df$success)

## [1] "N" "Y"

### recoding target feature (success) as factor ----  
df$success <- factor(df$success, levels = c("Y", "N"), labels = c("Yes", "No"))  
   
## brandSlogan ----  
### converting to lower text ----  
df <- df %>%  
 mutate(brandSlogan = tolower(brandSlogan))  
  
### strip irrelevant spaces ---  
df <- df %>%  
 mutate(brandSlogan = trimws(brandSlogan))  
  
## HASVIDEO----  
unique(df$hasVideo)

## [1] 1 0

summary(df$hasVideo)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 1.0000 0.7261 1.0000 1.0000

class(df$hasVideo)

## [1] "integer"

## RATING ----  
summary(df$rating)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.600 3.100 3.121 3.700 4.800

## PRICEUSD ----  
summary(df$priceUSD)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 0.04 0.12 19.01 0.50 39384.00 180

### filter out prices that are equal to 0 ----  
df <- df %>%  
 filter(priceUSD > 0 | is.na(priceUSD))  
  
## COUNTRYREGION ----  
summary(df$countryRegion)

## Length Class Mode   
## 2615 character character

str(df$countryRegion)

## chr [1:2615] "Singapore" "Malta" "UK" "Netherlands" "Mauritius" "UK" ...

### change country names to lower case ----  
df <- df %>%  
 mutate(countryRegion = tolower(countryRegion))  
  
### strip irrelevant spaces ---  
df <- df %>%  
 mutate(countryRegion = trimws(countryRegion))  
  
summary(df$countryRegion)

## Length Class Mode   
## 2615 character character

### check for unique values ----  
sort(unique(df$countryRegion))

## [1] "" "afghanistan"   
## [3] "andorra" "anguilla"   
## [5] "argentina" "armenia"   
## [7] "australia" "austria"   
## [9] "bahamas" "bangladesh"   
## [11] "barbados" "belarus"   
## [13] "belgium" "belize"   
## [15] "bermuda" "bosnia and herzegovina"   
## [17] "brazil" "british virgin islands"   
## [19] "bulgaria" "cambodia"   
## [21] "canada" "cayman islands"   
## [23] "chile" "china"   
## [25] "colombia" "congo"   
## [27] "costa rica" "croatia"   
## [29] "curacao" "curaçao"   
## [31] "cyprus" "czech republic"   
## [33] "denmark" "dominican republic"   
## [35] "ecuador" "egypt"   
## [37] "estonia" "finland"   
## [39] "france" "french polynesia"   
## [41] "georgia" "germany"   
## [43] "ghana" "gibraltar"   
## [45] "greece" "guinea-bissau"   
## [47] "honduras" "hungary"   
## [49] "iceland" "india"   
## [51] "indonesia" "ireland"   
## [53] "isle of man" "israel"   
## [55] "italy" "japan"   
## [57] "kazakhstan" "kuwait"   
## [59] "kyrgyzstan" "latvia"   
## [61] "liechtenstein" "lithuania"   
## [63] "luxembourg" "macedonia"   
## [65] "malaysia" "malta"   
## [67] "marshall islands" "mauritius"   
## [69] "mexico" "méxico"   
## [71] "monaco" "mongolia"   
## [73] "montenegro" "morocco"   
## [75] "netherlands" "new caledonia"   
## [77] "new zealand" "nigeria"   
## [79] "norway" "pakistan"   
## [81] "panama" "peru"   
## [83] "philippines" "poland"   
## [85] "portugal" "puerto rico"   
## [87] "romania" "russia"   
## [89] "saint kitts and nevis" "saint vincent and the grenadines"  
## [91] "samoa" "saudi arabia"   
## [93] "serbia" "seychelles"   
## [95] "sierra leone" "singapore"   
## [97] "slovakia" "slovenia"   
## [99] "south africa" "south korea"   
## [101] "spain" "sweden"   
## [103] "switzerland" "tanzania"   
## [105] "thailand" "timor-leste"   
## [107] "tunisia" "turkey"   
## [109] "uk" "ukraine"   
## [111] "united arab emirates" "usa"   
## [113] "vanuatu" "venezuela"   
## [115] "vietnam" "zimbabwe"

### change "curaçao" to curacao ----  
df <- df %>%  
 mutate(countryRegion = ifelse(countryRegion == 'curaçao',  
 'curacao', countryRegion))  
  
### testing ----  
df %>%  
 filter(countryRegion == 'curaçao') %>%  
 count()

## n  
## 1 0

### change méxico to mexico ----  
df <- df %>%  
 mutate(countryRegion = ifelse(countryRegion == 'méxico', 'mexico',  
 countryRegion))  
  
## START AND END DATE COLUMNS ----  
### convert startDate and endDate columns to date type ----  
df$startDate <- as.Date(df$startDate, format = "%d/%m/%Y")  
df$endDate <- as.Date(df$endDate, format = "%d/%m/%Y")  
  
### calculate duration of ICO campaign using startDate and endDate ----  
df <- df %>%  
 mutate('ico\_duration' = endDate - startDate)  
  
### convert ico\_duration to numeric ----  
df$ico\_duration <- as.numeric(df$ico\_duration)  
  
### testing ----  
class(df$ico\_duration)

## [1] "numeric"

### filter out observations with negative ico\_duration days ----  
df <- df %>%  
 filter(ico\_duration >= 0)  
  
## TEAMSIZE ----  
summary(df$teamSize)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 1.00 7.00 12.00 13.08 17.00 75.00 150

## GITHUB ----  
summary(df$hasGithub)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 1.0000 0.5745 1.0000 1.0000

## REDDIT ----  
summary(df$hasReddit)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 1.0000 0.6356 1.0000 1.0000

## PLATFORM ----  
### Identify unique values ---  
sort(unique(df$platform))

## [1] " " " Ethereum"   
## [3] "​Komodo" "Acclaim"   
## [5] "Achain" "AION"   
## [7] "Akroma" "Apollo Blockchain"   
## [9] "Ardor" "BEP2"   
## [11] "Bitcoin" "BitForex"   
## [13] "Bitshares" "BitShares"   
## [15] "Bitsmo" "Blockchain"   
## [17] "BTC" "ChainRepublik Blockchain"   
## [19] "Coffe" "Coincart"   
## [21] "Counterparty" "CryptoNight"   
## [23] "CryptoNote-based Blockchain" "DAG"   
## [25] "DECOIN Blockchain" "DPoS"   
## [27] "DPOS" "ENLTE PLATFORM"   
## [29] "ENTRY" "EOS"   
## [31] "ERC20" "ETH"   
## [33] "Ethererum" "Ethereum"   
## [35] "Ethereum " "Ethereum "   
## [37] "Ethereum " "Ethereum "   
## [39] "Ethereum " "Ethereum, Waves"   
## [41] "Etherum" "Fiber"   
## [43] "Filecoin network" "GoChain"   
## [45] "Graphene" "Hard-Fork of Litecoin"   
## [47] "Hybrid" "Hyperledger"   
## [49] "ICON" "Infinity Blockchain"   
## [51] "iOlite Blockchain" "IOV Blockchain"   
## [53] "IronGeekChain" "ISL-Blockchain"   
## [55] "JPMorganChase" "Keccak"   
## [57] "Komodo" "Lisk"   
## [59] "Litecoin" "MAHRA platform "   
## [61] "Monero" "Multichain"   
## [63] "MultiChain" "Native"   
## [65] "Neblio " "Nem"   
## [67] "NEM" "NEO"   
## [69] "Neurochain" "New Blockchain"   
## [71] "Newton" "Nilechain"   
## [73] "NXT" "Pivx"   
## [75] "PivX" "POS"   
## [77] "POS + POW" "POS,POW"   
## [79] "PoW" "pow/pos"   
## [81] "PoW/PoS" "QRC"   
## [83] "QTUM" "Ripemd160"   
## [85] "RSK" "Scrypt"   
## [87] "Separate blockchain" "Separate Blockchain"   
## [89] "Separate Blockchain " "Separate Blockchain "   
## [91] "SHA256 Coin" "Slatechain"   
## [93] "SmartX" "ST20"   
## [95] "StartEngine" "Steem"   
## [97] "Stellar" "Stellar Protocol"   
## [99] "Stratis" "STRATIS"   
## [101] "TEZOS" "Tomochain"   
## [103] "Tron" "TRON"   
## [105] "Tron " "Tron "   
## [107] "TTchain " "UNIVERSA"   
## [109] "VASYA" "VeChainThor VIP180"   
## [111] "Ventureon" "Wanchain"   
## [113] "Waves" "WAVES"   
## [115] "WizeBit" "X11"   
## [117] "X11 blockchain" "X13"   
## [119] "x13 " "XAYA"   
## [121] "xDAC" "YouToken"   
## [123] "Zilliqa" "Zuum"

### change text to lower  
df <- df %>%  
 mutate(platform = tolower(platform))  
  
### remove unnecessary space ----  
df <- df %>%  
 mutate(platform = trimws(platform))  
  
### convert btc to bitcoin  
df <- df %>%  
 mutate(platform = ifelse(platform == 'btc', "bitcoin", platform))  
  
### convert eth, ethererum, etherum, to ethereum  
df <- df %>%  
 mutate(platform = ifelse(platform == 'eth' | platform == 'ethererum' |   
 platform == 'etherum', "ethereum", platform))  
  
# convert x11 blockchain to x11  
df <- df %>%  
 mutate(platform = ifelse(platform == 'x11 blockchain', 'x11', platform))  
  
# convert stellar protocol to stellar  
df <- df %>%  
 mutate(platform = ifelse(platform == 'stellar protocol', 'stellar', platform))  
  
# convert "pos + pow", "pos,pow", "pow/pos" to "pos\_pow"  
df <- df %>%  
 mutate(platform = ifelse(platform == "pos + pow" | platform == "pos,pow" |  
 platform == "pos/pow" | platform == "pow/pos",   
 "pos\_pow", platform))  
  
## COINNUM ----  
summary(df$coinNum)

## Min. 1st Qu. Median Mean   
## 12 45000000 150000000 8688147209225   
## 3rd Qu. Max.   
## 500000000 22619078416760300

## MININVESTMENT ----  
summary(df$minInvestment)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.4493 1.0000 1.0000

## DISTRIBUTEDPERCENTAGE ----  
summary(df$distributedPercentage)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.400 0.560 1.097 0.700 869.750

### filter out values that are greater than 1 ----  
df <- df %>%  
 filter(!distributedPercentage > 1)  
  
summary(df)

## success brandSlogan hasVideo rating   
## Yes: 968 Length:2594 Min. :0.0000 Min. :1.000   
## No :1626 Class :character 1st Qu.:0.0000 1st Qu.:2.600   
## Mode :character Median :1.0000 Median :3.100   
## Mean :0.7255 Mean :3.118   
## 3rd Qu.:1.0000 3rd Qu.:3.700   
## Max. :1.0000 Max. :4.800   
##   
## priceUSD countryRegion startDate   
## Min. : 0.01 Length:2594 Min. :2010-01-10   
## 1st Qu.: 0.05 Class :character 1st Qu.:2018-03-01   
## Median : 0.14 Mode :character Median :2018-06-25   
## Mean : 20.36 Mean :2018-08-03   
## 3rd Qu.: 0.54 3rd Qu.:2018-11-30   
## Max. :39384.00 Max. :2020-06-06   
## NA's :179   
## endDate teamSize hasGithub hasReddit   
## Min. :2016-05-31 Min. : 1.00 Min. :0.0000 Min. :0.0000   
## 1st Qu.:2018-04-30 1st Qu.: 7.00 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :2018-09-01 Median :12.00 Median :1.0000 Median :1.0000   
## Mean :2018-10-12 Mean :13.09 Mean :0.5748 Mean :0.6353   
## 3rd Qu.:2019-03-10 3rd Qu.:17.00 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :2020-08-10 Max. :75.00 Max. :1.0000 Max. :1.0000   
## NA's :150   
## platform coinNum minInvestment   
## Length:2594 Min. : 12 Min. :0.0000   
## Class :character 1st Qu.: 45000000 1st Qu.:0.0000   
## Mode :character Median : 150360000 Median :0.0000   
## Mean : 8721639903230 Mean :0.4487   
## 3rd Qu.: 500000000 3rd Qu.:1.0000   
## Max. :22619078416800000 Max. :1.0000   
##   
## distributedPercentage ico\_duration   
## Min. :0.0000 Min. : 0.00   
## 1st Qu.:0.4000 1st Qu.: 29.00   
## Median :0.5600 Median : 46.00   
## Mean :0.5451 Mean : 70.53   
## 3rd Qu.:0.7000 3rd Qu.: 90.00   
## Max. :1.0000 Max. :3722.00   
##

# Saving cleaned data to csv file ----  
write.csv(df, file = "Data/data\_cleaned.csv", row.names = FALSE)