

1. La utilización de todos los registros si nos puede entregar un modelo mas completo ya que pueden realizarse pruebas con diferentes valores que aumenten la probabilidad de que la princesa se cure
2. Es importante aplicar el Featuring engeniring ya que nos ayuda a poder tener los datos de una manera más clara, nos ayuda a manejar todo tipo de dato como los categóricos y transformarlos al momento de poder entrenar el modelo, dando una mejor adaptabilidad para los requerimientos.
3. Se puede elegir de una manera objetiva observando cuales registros tienen la menor cantidad de datos faltantes para usar mas esos y pasar a un segundo plano las que si tienen un mayor número de datos faltantes que solo pueden entorpecer que el modelo aprenda de manera correcta llegando a cargar datos que no sean útiles.
4. En este caso se podría utilizar en vez del Accuracy el Precision ya que nos da una proporción entre las correctas que fueron predichas ya que nos ayuda a poder minimizar los fallos o falsos correctos o positivos.
5. Utilizar todos los datos en el modelo puede ser contraproducente ya que al ser una gran cantidad de datos puede generar overfitting y el modelo solo se aprendera lo que pasara y cuando se ingresen nuevos datos el modelo puede fallar y no predecir de forma correcta.
6. La métrica que mejor se acerca para poder medir el modelo es F1 ya que es una combinación entre la precisión y la sensibility dando un equilibrio entre los dos anteriores para poder acercarse mas al objetivo

## Parte 1

```
title: "AnPar2"
author: "Luis Tujab 1103920"
date: "2023-11-07"
output:
  word_document: default
  pdf_document: default
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r}
library(corrplot)
library(caret)
library(dplyr)
library(ggplot2)
library(e1071)
```
```

Luis Pablo Tujab Xuc  
1103920

```
#Cargar data
```{r}
data <- read.csv2("ingredientes.csv")
data
summary(data)
```

#Crear la funcion para determinar NA si es que existen
```{r}
DetectarNA <- function(data, tipo)
{

  #Determinar que columnas tienen datos faltantes
  colNA <- colnames(data)[!complete.cases(t(data))]
  # 1.2 substraer las columnas con datos faltanes
  IncompleteData <- data %>%
    select(colNA)

  #1.3 calcular porcentaje de datos faltantes apply(datos faltantes, 1 fila o 2 column, funcion )
  PorcentajeNA <- as.data.frame(apply(IncompleteData,2,function(col) sum(is.na(col))/ length(col) ))
  colnames(PorcentajeNA) <- c("Porcentaje")
  PorcentajeNA

  #Determinar columnas que si son procesables (las que cumplen con valores abajo del 5%)

  colPros <- PorcentajeNA %>%
    filter(PorcentajeNA <= 0.05)
}
```

```
colPros

if(tipo == 0)
{
  return(ColumnasProcesables)
}
else
{
  return(ColumnasNoProcesables)
}
}

data2 <- data
DataImputatio <- function(data2, colname, tipo) {
  if (tipo == "media"){
    data2[,colname] <- ifelse(is.na(data2[,colname]),
                             mean(data2[,colname],na.rm=T),
                             data1[,colname])
  }
  else
  {
    data2[,colname] <- ifelse(is.na(data2[,colname]),
                             median(data2[,colname],na.rm=T),
                             data2[,colname])
  }
}
}
```

Luis Pablo Tujab Xuc  
1103920

```
```{r}
set.seed(123)
trainingsamples<-
  createDataPartition(y =
data$Phoenix.Feather.Unicorn.Horn.Dragon.s.Blood.Mermaid.Tears.Fairy.Dust.Goblin.Toes.Witch.s.Brew.Griffin.Claw.Troll.Hair.Kraken.Ink.Minotaur.Horn.Basilisk.Scale.Chimera.Fang.Cured, p=0.8, list = FALSE)

traindata <- data[trainingsamples,]
testdata <- data[-trainingsamples,]
```
#Experimentacion con la variable dependiente
```{r}

model <- glm(Cured~HunicornHorn+ KrakenInk,data = traindata, family = binomial)
summary(model)
```
```

```
l
```{r}
predictionvalue <- model %>% predict(testdata,type = "response")
predictionvalue <- ifelse(predictionvalue>=0.5,1,0)

predictionvalue <- as.factor(predictionvalue)
testdata$cured <- as.factor(testdata$cured)

example <- confusionMatrix(data=predictionvalue, reference=testdata$cured, mode="everything")
example
```
```