Análisis y Predicción de Experiencia en Apps de Citas

Julio César Velázquez Corona

27 de Noviembre del 2025

Índice

1. Introducción

En el fondo, el objetivo del avance tecnológico que se nos presenta en el día a día es hacernos la vida más facil. Con la creación del teléfono y las apps esta facilidad ha crecido exponencialmente. Debido a ello surgieron las llamadas apps de citas, las cuales prometen cumplir la función de facilitarnos el proceso de encontrar pareja.

Uno podría decir que este tema es reciente y que esta información solo tiene un contexto actual. Sin embargo, esto no es así ya que en los años setentas se reportó el primer uso de la tecnología, mediante portales universitarios, para encontrar pareja.

El motivo de realizar este proyecto surge debido a que el uso de estas tecnologías ha estado creciendo de manera completamente desmedida y es de interes predecir cómo sería nuestra experencia si uno llegara a usar estas aplicaciones. Según un estudio de The Knot, más del 50 % de las parejas en Estados Unidos se conocieron mediante apps de citas y eso es una locura.

Para el desarrollo de este proyecto, se utilizará R, la metodología de CRISP-ML para construir modelos predictivos eficientes y nos acompañaremos del uso de estadística inferencial para explorar la información con la que se cuenta.

2. Entendimiento de los Datos

El objetivo de este proyecto es predecir si nuestra experiencia al usar una app de citas será positiva o negativa. El caso de éxito que se buscará es que la experiencia sea positiva. Los datos con los que se cuenta son provenientes del dataset llamado `online_dating_2023_responses.csv`.

Aquellos datos se obtuvieron mediente una encuesta aplicada a 32 personas en el año 2023 que contenía preguntas acerca de la experiencia que obtuvieron al usar una app de citas. Los resultados obtenidos fueron publicados en la página web dedicada a ciencia de datos llamada Kaggle. Prodeceremos a extraer y mostrar los datos que tenemos:

```
> setwd("C:\\Personal\\primerAnalisis\\latex-homework-template-master\\data")
> data <- read.csv("online_dating_2023_responses.csv", header = TRUE)
> lapply(data,unique)
$Timestamp
"01/12/2023 12:16"   "01/12/2023 13:04"   "01/12/2023 15:17"
"01/12/2023 15:35"   "01/12/2023 15:47"   "01/12/2023 15:59"
"01/12/2023 16:12"   "01/12/2023 16:36"   "01/12/2023 16:38"
"01/12/2023 17:04"   "01/12/2023 17:33"   "01/12/2023 17:35"
"01/12/2023 20:10"   "1-13-2023 1:15:15"  "1-13-2023 9:50:04"
"1-13-2023 10:01:49" "1-13-2023 10:18:13" "1-13-2023 12:57:56"
"1-13-2023 13:11:40" "1-13-2023 14:29:19" "1-13-2023 15:09:13"
"1-13-2023 15:18:54" "1-13-2023 16:25:00" "1-13-2023 17:01:36"
"1-13-2023 17:07:47" "1-13-2023 18:56:25" "1-13-2023 19:49:41"
"1-13-2023 20:15:36" "1-14-2023 7:16:30"  "1-14-2023 22:47:56"
"1-14-2023 23:23:15" "1-15-2023 1:53:31"
$What.is.your.age.
"24 to 34" "18 to 24" "35 +"
$Are.you.
```

```
"Female" "Male"
$Have.you.used.online.dating.app.s..before.
"Yes" "No"
$How.often.do.you.use.online.dating.app.s..
"Frequently"  "Rarely"  ""  "Occasionally"
$What.are.you.looking.for.in.online.dating.app.s..
"Long-term relationship"  "I don't know yet"  ""
"Short-term or casual relationship"
$How.long.have.you.used.online.dating.app.s..
"1-6 months"  "more than 1 year"  ""  "approximately 1 year"
$Do.you.prefer.dating.app.s..to.have.more.interactive.ice.breaker.activities.
"Yes"  "No"  ""
$Do.you.prefer.to.have.a.certain.number.but.selective.profiles.or.
unlimited.profiles.recommended.per.day.
"Certain but selective profiles"  "Unlimited profiles"  ""
$Have.you.experienced.ghosting.or.being.ghosted.
"Both"  "Being ghosted"  ""  "Neither"  "Ghosting"
$What.s.your.overall.experience.using.online.dating.app.s..
"Positive"  "Negative"  ""  "Very negative"
$What.s.the.main.reason.you.haven.t.used.any.dating.app.
""  "Have been in a relationship since I was a minor"
"Prefer real connection over online dating"
"Not interested"
"I'm too old"
"not my style, also, im old school, traditional"
"I used earlier but didn't got any result so I left out"
$Do.you.intend.to.use.any.online.dating.app.
"" "No"  "Yes"
```

La explicación y nombre de cada columna, obtenido con `colnames(data)` se describe a continuación:

- `Timestamp`: la fecha en la que se recibió la respuesta del encuestado.
- `What.is.your.age`: aproximación de la edad del encuestado.
- `Are.you`: género del encuestado.
- `Have.you.used.online.dating.app.s..before`: si el encuestado ha usado antes una app de citas.
- `How.often.do.you.use.online.dating.app.s`: qué tan seguido el encuestado utiliza apps de citas.
- `What.are.you.looking.for.in.online.dating.app.s`: lo que busca el encuestado al usar una app de citas.
- `How.long.have.you.used.online.dating.app.s`: el tiempo que el encuestado lleva usando apps de citas.
- `Do.you.prefer.dating.app.s..to.have.more.interactive.ice.breaker.activities`: si el encuestado preferiría o no, un rompehielos en las apps de citas.
- `Do.you.prefer.to.have.a.certain.number.but.selective.profiles.or.unlimited.profiles.-recommended.per.day`: si el encuestado prefiere ver perfiles recomendados o ilimitados.

- `Have.you.experienced.ghosting.or.being.ghosted`: si el encuestado ha sido o ha "ghosteado", es decir, que corte o que le corten la comunicación sin previo aviso.
- `What.s.your.overall.experience.using.online.dating.app.s`: experiencia del encuestado en las apps de citas.
- `What.s.the.main.reason.you.haven.t.used.any.dating.app`: razón por la cual el encuestado no usa apps de citas.
- `Do.you.intend.to.use.any.online.dating.app`: si el encuestado usaría una apps de citas en algún momento.

Ya con el dataset descrito, hay que considerar limitaciones importantes: el tamaño reducido del dataset nos impone restricciones respecto a la solidez y a la confiabilidad de los resultados que podemos conseguir. La mayoría de las pruebas estadísticas y modelos funcionan mejor y arrojan mejores resultados cuantos más datos tengamos.

Sin embargo, tenemos alternativas, como las pruebas no paramétricas y modelos como el Naive Bayes que funcionan bien con pocas observaciones. Habiendo dicho eso, se debe pedir que invariablemente del resultado obtenido, se contextualice sobre la posibilidad de que la predicción pueda llegar a no ser perfecta.

3. Preparación de los Datos

Nuestro dataset está un poco sucio y se prodecerá a hacer una limpieza. Lo primero que se hará es borrar la columna de tiempo ya que no aporta nada.

```
> data <- subset(data, select = -Timestamp)
```

Los nombres de las columnas son poco claros y demasiado descriptivos así que se les cambiará el nombre por algo más significativo.

```
> data <- data %>%
> rename(
    age = What.is.your.age.,
    gender = Are.you.,
    used_apps_before = Have.you.used.online.dating.app.s..before.,
    apps_usage_rate = How.often.do.you.use.online.dating.app.s..,
    apps_use_objective = What.are.you.looking.for.in.online.dating.app.s..,
    apps_usage_time = How.long.have.you.used.online.dating.app.s..,
    icebreaker_preference =
    Do.you.prefer.dating.app.s..to.have.more.interactive.ice.breaker.activities.,
    selective_or_unlimited_profiles =
    Do.you.prefer.to.have.a.certain.number.but.selective.profiles
    .or.unlimited.profiles.recommended.per.day.,
    ghosting_or_ghosted = Have.you.experienced.ghosting.or.being.ghosted.,
    apps_usage_experience = What.s.your.overall.experience.using.online.dating.app.s..,
    usage_reason = What.s.the.main.reason.you.haven.t.used.any.dating.app.,
    usage_interest = Do.you.intend.to.use.any.online.dating.app.
  )
> colnames(data)
"age"                    "gender"
```

```
"used_apps_before"      "apps_usage_rate"
"apps_use_objective"    "apps_usage_time"
"icebreaker_preference" "selective_or_unlimited_profiles"
"ghosting_or_ghosted"   "apps_usage_experience"
"usage_reason"          "usage_interest"
```

Para que podamos obtener resultados con menos ruido, nos conviene mucho eliminar las filas en donde el encuestado contestó que nunca ha usado una app de citas. Sin embargo, para realizar análisis estadísticos posteriores, se creará un subdataset que contega los datos donde respondieron que sí han usado apps de citas y otro subdataset donde respondieron que no.

```
data_yes <- subset(data, used_apps_before == "Yes")
data_no <- subset(data, used_apps_before == "No")
data <- subset(data, used_apps_before != "No")
```

En `data` ya no ocupamos las columnas llamadas `usage_reason` y `usage_interest` así que serán eliminadas.

```
data <- subset(data, select = -usage_reason)
data <- subset(data, select = -usage_interest)
```

Para que los análisis estadísticos y predictivos sean eficientes, debemos de convertir nuestras variables cualitativas a factores.

```
> data$age <- factor(
  data$age,
  levels = c("18 to 24", "24 to 34"),
  ordered = TRUE
)
> data$gender <- factor(
  data$gender,
  levels = c("Male", "Female"),
  ordered = FALSE
)
> data$used_apps_before <- factor(
  data$used_apps_before,
  levels = c("Yes"),
  ordered = FALSE
)
> data$apps_usage_rate <- factor(
  data$apps_usage_rate,
  levels = c("Rarely", "Occasionally", "Frequently"),
  ordered = TRUE
)
> data$apps_use_objective <- factor(
  data$apps_use_objective,
  levels = c("I don't know yet", "Short-term or casual relationship", "Long-term
  ↪  relationship"),
  ordered = TRUE
)
> data$apps_usage_time <- factor(
  data$apps_usage_time,
```

```
  levels = c("1-6 months", "approximately 1 year", "more than 1 year" ),
  ordered = TRUE
)
> data$icebreaker_preference <- factor(
  data$icebreaker_preference,
  levels = c("Yes", "No"),
  ordered = FALSE
)
> data$selective_or_unlimited_profiles <- factor(
  data$selective_or_unlimited_profiles,
  levels = c("Certain but selective profiles", "Unlimited profiles"),
  ordered = FALSE
)
> data$ghosting_or_ghosted <- factor(
  data$ghosting_or_ghosted,
  levels = c("Neither", "Being ghosted", "Ghosting", "Both"),
  ordered = TRUE
)
> data$apps_usage_experience <- factor(
  data$apps_usage_experience,
  levels = c("Negative", "Positive"),
  ordered = TRUE
)
```

En los subdatasets `data_yes` y `data_no`, se realizará el mismo proceso de conversion a factores y se eliminarán las columnas innecesarias.

```
## Limpieza de data_yes
> data_yes <- subset(data_yes, select = -usage_reason)
> data_yes <- subset(data_yes, select = -usage_interest)
> lapply(data_yes,unique)
> data_yes$age <- factor(
  data_yes$age,
  levels = c("18 to 24", "24 to 34"),
  ordered = TRUE
)
> data_yes$gender <- factor(
  data_yes$gender,
  levels = c("Male", "Female"),
  ordered = FALSE
)
> data_yes$used_apps_before <- factor(
  data_yes$used_apps_before,
  levels = c("Yes"),
  ordered = FALSE
)
> data_yes$apps_usage_rate <- factor(
  data_yes$apps_usage_rate,
  levels = c("Rarely", "Occasionally", "Frequently"),
  ordered = TRUE
```

```
)
> data_yes$apps_use_objective <- factor(
  data_yes$apps_use_objective,
  levels = c("I don't know yet", "Short-term or casual relationship", "Long-term
  ↪  relationship"),
  ordered = TRUE
)
> data_yes$apps_usage_time <- factor(
  data_yes$apps_usage_time,
  levels = c("1-6 months", "approximately 1 year", "more than 1 year" ),
  ordered = TRUE
)
> data_yes$icebreaker_preference <- factor(
  data_yes$icebreaker_preference,
  levels = c("Yes", "No"),
  ordered = FALSE
)
> data_yes$selective_or_unlimited_profiles <- factor(
  data_yes$selective_or_unlimited_profiles,
  levels = c("Certain but selective profiles", "Unlimited profiles"),
  ordered = FALSE
)
> data_yes$ghosting_or_ghosted <- factor(
  data_yes$ghosting_or_ghosted,
  levels = c("Neither", "Being ghosted", "Ghosting", "Both"),
  ordered = TRUE
)
> data_yes$apps_usage_experience <- factor(
  data_yes$apps_usage_experience,
  levels = c("Negative", "Positive"),
  ordered = TRUE
)
## Limpieza de data_no
> data_no <- subset(data_no, select = -apps_usage_rate)
> data_no <- subset(data_no, select = -apps_use_objective)
> data_no <- subset(data_no, select = -apps_usage_time)
> data_no <- subset(data_no, select = -icebreaker_preference)
> data_no <- subset(data_no, select = -selective_or_unlimited_profiles)
> data_no <- subset(data_no, select = -ghosting_or_ghosted)
> data_no <- subset(data_no, select = -apps_usage_experience)
> data_no$age <- factor(
  data_no$age,
  levels = c("18 to 24", "24 to 34", "35 +"),
  ordered = TRUE
)
> data_no$gender <- factor(
  data_no$gender,
  levels = c("Male", "Female"),
  ordered = FALSE
```

```
)
> data_no$used_apps_before <- factor(
  data_no$used_apps_before,
  levels = c("No"),
  ordered = FALSE
)
> data_no$usage_reason <- factor(
  data_no$usage_reason,
  levels = c("Have been in a relationship since I was a minor",
             "Prefer real connection over online dating",
             "Not interested",
             "I'm too old",
             "not my style, also, im old school, traditional",
             "I used earlier but didn't got any result so I left out"),
  ordered = FALSE
)
> data_no$usage_interest <- factor(
  data_no$usage_interest,
  levels = c("No","Yes"),
  ordered = TRUE
)
```

Al tener en cuenta que nuestra variable de interés es `apps_usage_experience`, podemos tratar de definir variables significativas que pueden tener un impacto importante en nuestro análisis y para ello, se realizarán pruebas no paramétricas de Fisher a cada variable con respecto a nuestra variable de interés para verificar si existe alguna asociación. Las hipótesis se plantean a continuación:

Hipótesis nula: $H_0 : P(A = i, B = j) = P(A = i) \cdot P(B = j)$

Hipótesis alternativa: $H_a : P(A = i, B = j) \neq P(A = i) \cdot P(B = j)$

Se utilizará un nivel de significancia estandar $\alpha = 0.05$ para nuestras regiones de rechazo.

```
> (compara1 <- table(data$apps_usage_experience, data$age))
> fisher.test(compara1)
p-value = 0.3498 # No hay asociación con la edad
```

```
> (compara2 <- table(data$apps_usage_experience, data$gender))
> fisher.test(compara2)
p-value = 0.6499 # No hay asociación con el género
```

```
> (compara3 <- table(data$apps_usage_experience, data$apps_usage_rate))
> fisher.test(compara3)
p-value = 0.1498 # No hay asociación con la frecuencia de uso
```

```
> (compara4 <- table(data$apps_usage_experience, data$apps_use_objective))
> fisher.test(compara4)
p-value = 0.01643 # Si hay asociación con el objetivo de uso
```

```
> (compara5 <- table(data$apps_usage_experience, data$apps_usage_time))
> fisher.test(compara5)
p-value = 0.6563 # No hay asociación con el tiempo de uso
```

```
> (compara6 <- table(data$apps_usage_experience, data$icebreaker_preference))
> fisher.test(compara6)
p-value = 0.3698 # No hay asociación con la preferencia de un rompehielos
```

```
> (compara7 <- table(data$apps_usage_experience, data$selective_or_unlimited_profiles))
> fisher.test(compara7)
p-value = 1 # No hay asociación con la preferencia de recibir perfiles selectivos o
↪  ilimitados
```

```
> (compara8 <- table(data$apps_usage_experience, data$ghosting_or_ghosted))
> fisher.test(compara8)
p-value = 0.6499 # No hay asociación con la conducta del ghosting, qué extraño...
```

Respecto a nuestros p-value, podemos darnos cuenta que la variable que tiene más impacto en nuestra variable de interes es `apps_use_objective`, lo cual tiene sentido y ayuda a concluir que esa variable es la que se debe usar en nuestros modelos. Sin embargo, es extraño que se nos haya presentado que la variable `ghosting_or_ghosted` no tiene impacto. Debido al contexto social, esta variable debe resguardarse como una variable asociada independientemente del p-value.

4. Construcción y Afinación

Al tener definida la variable dependiente e independiente, procedemos a elegir modelos candidatos. Las variables son cualitativas y contamos con 31 observaciones, por lo que necesitamos modelos que puedan trabajar eficientemente con esas condiciones. A continuación se presentan los modelos candidatos:

- Naive Bayes: posiblemente la mejor opción, está basado en el Teorema de Bayes y consiste en predecir una clase con respecto a otra asumiendo independencia condicional. Solo necesita contar frecuencias, por lo que funcionará bien con nuestro dataset pequeño. La idea de la que parte es la siguiente:
  - Se plantea el Teorema de Bayes:

    $$P(\Omega_j|x) = \frac{P(\Omega_j)\,P(x|\Omega_j)}{P(x)}$$

  - Se aplica el supuesto de independencia:

    $$P(\Omega_j|x) = \frac{P(\Omega_j)\,P\left((x_1,\ldots,x_p)|\Omega_j\right)}{P(x)} = \frac{P(\Omega_j)\,P(x_1|\Omega_j)\,P(x_2|\Omega_j)\cdots P(x_p|\Omega_j)}{P(x_1)\,P(x_2)\cdots P(x_p)}$$

  - Se omite el denominador para generar constancia con las características:

    $$P(\Omega_j|x_1,\ldots,x_p) \propto P(\Omega_j)\prod_{i=1}^{p} P(x_i|\Omega_j)$$

  - Toma la clase con mayor valor para clasificar:

    $$\hat{y} = \underset{\Omega_j}{\operatorname{argmax}}\left[P(\Omega_j)\prod_{i=1}^{p} P(x_i|\Omega_j)\right]$$

- Regresión logística: es parecido a la regresión lineal, predice una probabilidad en vez de un número como tal. Funciona bien con variables cualitativas y para modelados simples. Tiene la ventaja de que creará un parámetro β_1, el cual nos indicará el aumento o la disminución de la cantidad de nuestra variable dependiente con respecto a la independiente. La idea de la que parte es la siguiente:

- Se plantea el logit, que es el logaritmo de la razón de las probabilidades:

$$\text{logit}(p) = ln\left(\frac{p}{1-p}\right)$$

- Ese logit se modela como una combinación lineal:

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

- Se aplica la función logística para convertir el resultado en una probabilidad entre 0 y 1:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)}}$$

Ya con nuestros modelos candidatos definidos, teóricamente se debe empezar con la división de la muestra. Sin embargo, se utilizará el método de resubstitución debido a que nuestro tamaño de muestra es pequeño. Es decir, se utilizarán todos los datos presentes como el conjunto de entrenamiento (CE).

```
> CE <- subset(data)
```

Procedemos a entrenar un modelo Naive Bayes y por convención, se utilizará la corrección de Laplace.

```
> nb_m <- NaiveBayes(apps_usage_experience ~ apps_use_objective,
                     data = CE, laplace=1)
```

Ahora, entrenaremos un modelo de regresión logística.

```
> rl_m <- glm(apps_usage_experience ~ apps_use_objective,
              data = CE,
              family = binomial)
> summary(rl_m)
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)              0.1122  2617.8736   0.000    1.000
apps_use_objective.L    14.0732  3400.7175   0.004    0.997
apps_use_objective.Q   -23.8261  5436.4227  -0.004    0.997
```

Este modelo no cumplió mis expectativas. El error estandar es altísimo, lo que indica que el modelo enfrentó un problema de separación casi completa con respecto a los puntos. Eso provoca que los parámetros crezcan mucho y basicamente sin limite. Además, los p-values son extremadamente altos, todo indica que no hay confianza con ese modelo.

Concluimos que el modelo que se elegirá para la evaluación es claramente Naive Bayes. Tengo la teoría de que este modelo funcionó por la independencia que asume entre el predictor dado la clase. Como solo utilizamos una variable independiente, la regresión logística asoció por completo una categoría con una clase y eso provocó un problema grave de separación completa.

5. Evaluación

Para que alguien pueda utilizar el modelo solo tiene que contestarse lo siguiente: ¿Qué buscas en una app de citas? Invariablemente de la respuesta, solo se puede caer en tres categorias:

- Quiero algo ”bien” (Long-term relationship).

- Quiero algo "de rapido" (Short-term or casual relationship).
- No sé qué quiero en mi vida (I don't know yet).

Dependiendo de la respuesta, se correrá el modelo para crear una prueba de cada resultado:

```
> ## Quiero algo "bien"
> predict(nb_m, newdata = data.frame(
            apps_use_objective = factor(
              'Long-term relationship', levels = levels(CE$apps_use_objective))))
$posterior
      Negative  Positive
[1,] 0.4166667 0.5833333
```

```
> ## Quiero algo "de rapido"
> predict(nb_m, newdata = data.frame(
            apps_use_objective = factor(
              'Short-term or casual relationship',
              levels = levels(CE$apps_use_objective))))
$posterior
        Negative  Positive
[1,] 0.003322259 0.9966777
```

```
> ## No sé qué quiero en mi vida
> predict(nb_m, newdata = data.frame(
            apps_use_objective = factor(
              'I don\'t know yet', levels = levels(CE$apps_use_objective))))
$posterior
     Negative    Positive
[1,] 0.998004 0.001996008
```

6. Conclusiones

Los resultados son interesantes: una persona que entra para buscar una relación seria y estable tiene una probabilidad equilibrada de obtener una experiencia positiva o negativa. Sin embargo, una persona que busque algo casual, está prácticamente asegurado de que obtendrá una experiencia positiva. Por último, una persona que no sabe lo que quiere solo entra a perder el tiempo y está prácticamente asegurada de obtener una experiencia negativa.

Si contextualizamos socialmente los resultados, en términos generales y por experiencia propia, se puede asumir que la mayoría de la gente busca algo casual. Por lo que se concluye lo siguiente: si buscamos la mejor experiencia posible, definitivamente debemos de saber lo que queremos y no debemos asumir que la gente busca compromisos tan duraderos.

Contenido adicional

Con los subdatasets `data_yes` y `data_no` se tenía planeado realizar pruebas no paramétricas. Sin embargo, se presentaron múltiples dificultades debido a que los datos son cualitativos y se omitió esta accion.

7. Referencias

S. F. Eslava. *Apuntes de Estadística Inferencial.* Universidad Anáhuac México Campus Norte, 2025.

C. V. Patiño. *Apuntes de Machine Learning.* Universidad Anáhuac México Campus Norte, 2025.

F. L. Pontecorvo. *Apuntes de Big Data.* Universidad Anáhuac México Campus Norte, 2025.

8. Apéndice

```
library(dplyr)
library(e1071)
library(MASS)
library(klaR)

# Lectura de datos
setwd("C:\\Personal\\primerAnalisis\\latex-homework-template-master\\data")
data <- read.csv("online_dating_2023_responses.csv", header = TRUE)
lapply(data,unique)

# Explicacion de columnas
View(data)
colnames(data)

# Limpieza de datos
## Borrar timestamp
data <- subset(data, select = -Timestamp)
## Cambiar nombre de columnas
data <- data %>%
  rename(
    age = What.is.your.age.,
    gender = Are.you.,
    used_apps_before = Have.you.used.online.dating.app.s..before.,
    apps_usage_rate = How.often.do.you.use.online.dating.app.s..,
    apps_use_objective = What.are.you.looking.for.in.online.dating.app.s..,
    apps_usage_time = How.long.have.you.used.online.dating.app.s..,
    icebreaker_preference =
    ↪  Do.you.prefer.dating.app.s..to.have.more.interactive.ice.breaker.activities.,
    selective_or_unlimited_profiles =
    ↪  Do.you.prefer.to.have.a.certain.number.but.selective.profiles.or.unlimited.
    profiles.recommended.per.day.,
    ghosting_or_ghosted = Have.you.experienced.ghosting.or.being.ghosted.,
    apps_usage_experience = What.s.your.overall.experience.using.online.dating.app.s..,
    usage_reason = What.s.the.main.reason.you.haven.t.used.any.dating.app.,
    usage_interest = Do.you.intend.to.use.any.online.dating.app.
  )
colnames(data)
## Crear subdatasets para analisis despues y quitar los 'no' del original
data_yes <- subset(data, used_apps_before == "Yes")
```

```
data_no <- subset(data, used_apps_before == "No")
data <- subset(data, used_apps_before != "No")
## Borrar columnas basura en data
View(data)
data <- subset(data, select = -usage_reason)
data <- subset(data, select = -usage_interest)
## Pasar a factores los valores de data
lapply(data,unique)
unique(data$age)
data$age <- factor(
  data$age,
  levels = c("18 to 24", "24 to 34"),
  ordered = TRUE
)
data$gender <- factor(
  data$gender,
  levels = c("Male", "Female"),
  ordered = FALSE
)
data$used_apps_before <- factor(
  data$used_apps_before,
  levels = c("Yes"),
  ordered = FALSE
)
data$apps_usage_rate <- factor(
  data$apps_usage_rate,
  levels = c("Rarely", "Occasionally", "Frequently"),
  ordered = TRUE
)
data$apps_use_objective <- factor(
  data$apps_use_objective,
  levels = c("I don't know yet", "Short-term or casual relationship", "Long-term
  ↪  relationship"),
  ordered = TRUE
)
data$apps_usage_time <- factor(
  data$apps_usage_time,
  levels = c("1-6 months", "approximately 1 year", "more than 1 year" ),
  ordered = TRUE
)
data$icebreaker_preference <- factor(
  data$icebreaker_preference,
  levels = c("Yes", "No"),
  ordered = FALSE
)
data$selective_or_unlimited_profiles <- factor(
  data$selective_or_unlimited_profiles,
  levels = c("Certain but selective profiles", "Unlimited profiles"),
  ordered = FALSE
```

```
)
data$ghosting_or_ghosted <- factor(
  data$ghosting_or_ghosted,
  levels = c("Neither", "Being ghosted", "Ghosting", "Both"),
  ordered = TRUE
)
data$apps_usage_experience <- factor(
  data$apps_usage_experience,
  levels = c("Negative", "Positive"),
  ordered = TRUE
)
## Conversion a factores en data_yes y eliminacion de columna basuras
data_yes <- subset(data_yes, select = -usage_reason)
data_yes <- subset(data_yes, select = -usage_interest)
lapply(data_yes,unique)
data_yes$age <- factor(
  data_yes$age,
  levels = c("18 to 24", "24 to 34"),
  ordered = TRUE
)
data_yes$gender <- factor(
  data_yes$gender,
  levels = c("Male", "Female"),
  ordered = FALSE
)
data_yes$used_apps_before <- factor(
  data_yes$used_apps_before,
  levels = c("Yes"),
  ordered = FALSE
)
data_yes$apps_usage_rate <- factor(
  data_yes$apps_usage_rate,
  levels = c("Rarely", "Occasionally", "Frequently"),
  ordered = TRUE
)
data_yes$apps_use_objective <- factor(
  data_yes$apps_use_objective,
  levels = c("I don't know yet", "Short-term or casual relationship", "Long-term
  ↪  relationship"),
  ordered = TRUE
)
data_yes$apps_usage_time <- factor(
  data_yes$apps_usage_time,
  levels = c("1-6 months", "approximately 1 year", "more than 1 year" ),
  ordered = TRUE
)
data_yes$icebreaker_preference <- factor(
  data_yes$icebreaker_preference,
  levels = c("Yes", "No"),
```

```
  ordered = FALSE
)
data_yes$selective_or_unlimited_profiles <- factor(
  data_yes$selective_or_unlimited_profiles,
  levels = c("Certain but selective profiles", "Unlimited profiles"),
  ordered = FALSE
)
data_yes$ghosting_or_ghosted <- factor(
  data_yes$ghosting_or_ghosted,
  levels = c("Neither", "Being ghosted", "Ghosting", "Both"),
  ordered = TRUE
)
data_yes$apps_usage_experience <- factor(
  data_yes$apps_usage_experience,
  levels = c("Negative", "Positive"),
  ordered = TRUE
)
## Conversion a factores en data_no y eliminacion de columna basuras
lapply(data_no,unique)
data_no <- subset(data_no, select = -apps_usage_rate)
data_no <- subset(data_no, select = -apps_use_objective)
data_no <- subset(data_no, select = -apps_usage_time)
data_no <- subset(data_no, select = -icebreaker_preference)
data_no <- subset(data_no, select = -selective_or_unlimited_profiles)
data_no <- subset(data_no, select = -ghosting_or_ghosted)
data_no <- subset(data_no, select = -apps_usage_experience)
data_no$age <- factor(
  data_no$age,
  levels = c("18 to 24", "24 to 34", "35 +"),
  ordered = TRUE
)
data_no$gender <- factor(
  data_no$gender,
  levels = c("Male", "Female"),
  ordered = FALSE
)
data_no$used_apps_before <- factor(
  data_no$used_apps_before,
  levels = c("No"),
  ordered = FALSE
)
data_no$usage_reason <- factor(
  data_no$usage_reason,
  levels = c("Have been in a relationship since I was a minor",
             "Prefer real connection over online dating",
             "Not interested",
             "I'm too old",
             "not my style, also, im old school, traditional",
             "I used earlier but didn't got any result so I left out"),
```

```
  ordered = FALSE
)
data_no$usage_interest <- factor(
  data_no$usage_interest,
  levels = c("No","Yes"),
  ordered = TRUE
)
## Explicacion de variables mas significativas
View(data)
(compara1 <- table(data$apps_usage_experience, data$age))
fisher.test(compara1)
(compara2 <- table(data$apps_usage_experience, data$gender))
fisher.test(compara2)
(compara3 <- table(data$apps_usage_experience, data$apps_usage_rate))
fisher.test(compara3)
(compara4 <- table(data$apps_usage_experience, data$apps_use_objective))
fisher.test(compara4)
(compara5 <- table(data$apps_usage_experience, data$apps_usage_time))
fisher.test(compara5)
(compara6 <- table(data$apps_usage_experience, data$icebreaker_preference))
fisher.test(compara6)
(compara7 <- table(data$apps_usage_experience, data$selective_or_unlimited_profiles))
fisher.test(compara7)
(compara8 <- table(data$apps_usage_experience, data$ghosting_or_ghosted))
fisher.test(compara8)

# Construccion de modelos y afinacion
## Asignación de CE y CP
head(data)
CE <- subset(data)
CP <- subset(data)
## Naive Bayes
nb_m <- NaiveBayes(apps_usage_experience ~ apps_use_objective,
                   data = CE, laplace=1)
print(nb_m)
## Regresion logistica
rl_m <- glm(apps_usage_experience ~ apps_use_objective,
             data = CE,
             family = binomial)
summary(rl_m)

# Evaluacion
table(CE$apps_use_objective, CE$apps_usage_experience)
## Quiero algo "bien"
predict(nb_m, newdata = data.frame(
          apps_use_objective = factor(
            'Long-term relationship', levels = levels(CE$apps_use_objective))))

## Quiero algo "de rapido"
```

```
predict(nb_m, newdata = data.frame(
  apps_use_objective = factor(
    'Short-term or casual relationship', levels = levels(CE$apps_use_objective))))

## No sé qué quiero en mi vida
predict(nb_m, newdata = data.frame(
  apps_use_objective = factor(
    'I don\'t know yet', levels = levels(CE$apps_use_objective))))
```