APRENDEZAJE ESTADÍSTICO Y TÉCNICAS ANALÍTICAS

2022 - 2023

MODELO DE CLASIFICACIÓN PARA PREDECIR INFARTOS CEREBRALES

JESÚS ROMERO NIETO, NICOLÁS TAPIADOR SOBRINO, MANUEL VÍCTOR GARCÍA-MINGUILLÁN, ROBERTO DE VICENTE DE LA CRUZ, JOSE RAMÓN CORTÉS ALCAIDE

Table of Contents

MODELOS DE CLASIFICACIÓN PARA PREDECIR INFARTOS CEREBRALES.

INTRODUCCIÓN.

CONTEXTO Y JUSTIFICACIÓN DEL TEMA.

Actualmente, la enfermedad cerebrovascular aguda o más conocido como **infarto cerebral** o **ictus**, es la tercera causa de muerte global en España. Este sucede cuando se detiene o disminuye el flujo sanguíneo a parte del cerebro. Al no poder recibir el oxígeno y nutrientes que necesitan, las células cerebrales comienzan a morir en minutos. Esto puede causar un daño grave en el cerebro, llegando a provocar **discapacidad permanente** e incluso la **muerte**. La rapidez en estos casos es fundamental, cuanto antes se detecte, el tratamiento es más específico y eficaz por lo que, los daños mencionados con anterioridad pueden reducirse significativamente. La tendencia demográfica mundial se encamina hacia el envejecimiento de la población debido al aumento de la esperanza de vida, por este motivo, los infartos cerebrales se han convertido en una de las enfermedades más comunes debido a que se producen mayoritariamente en personas mayores.

La prevención y el diagnóstico temprano de los infartos cerebrales es crucial para **reducir** la mortalidad y la discapacidad asociada a este suceso. Por ello, es necesario disponer de **datos** de alta calidad sobre los factores de riesgo y las características que cada individuo tenga de cara a diagnosticar dicha enfermedad. En este estudio, se cuenta con una muestra de pacientes que incluye información sobre su estado de salud y sí han desarrollado o no, un infarto cerebral. Esta información puede servir para el desarrollo de diversas herramientas valiosas que pueda identificar **patrones y tendencias**, evaluar el impacto de las intervenciones preventivas y de tratamiento, así como desarrollar modelos de predicción y asesoramiento clínico. Por todas estas razones, el análisis de una base de datos médica sobre infartos cerebrales puede proporcionar información valiosa para comprender mejor los mecanismos subyacentes, identificar factores de riesgo y protección, y evaluar el **impacto** de las intervenciones preventivas y de tratamiento.

OBJETIVOS.

Una vez expuestos los criterios fundamentales que abarcan este estudio. El **objetivo general** será desarrollar un modelo de clasificación que permitan identificar a los pacientes que, en el momento de ingreso, puedan estar sufriendo un infarto cerebral, de esta manera, la herramienta desarrollada pueda servir de asesoramiento clínico de cara a facilitar a los profesionales sanitarios a tomar decisiones, sobre el tratamiento y el seguimiento de los pacientes con infartos cerebrales.

Este objetivo general vendrá derivado por la elaboración de pequeños **objetivos específicos** que tendrán, relación con el estudio:

- **Determinación de las variables de estudio:** Se tendrá que determinar el número de variables explicativas óptimo para una correcta clasificación del modelo, eligiendo entre los modelos más simples con menos variables y una precisión aceptable a los más complejos y específicos a la hora de clasificar.

- **Proceso de limpieza y preparación de los variables**, para poder aplicar las técnicas estadísticas previas a los modelos.

- Estimar un **modelo de clasificación predictivo** con el que poder diagnosticar a los pacientes, introduciendo los síntomas establecidos.

ESTRUCTURA DEL TRABAJO.

Para poder lograr y conseguir los objetivos marcados, el estudio se ha dividido en diferentes capítulos, concretamente en tres. En el primero de ellos, se hablará de la **metodología** que se llevará a cabo en el estudio, de qué fuentes se extraerán los datos con los que se va a trabajar, la **especificación** de las variables consideradas y un **análisis exploratorio** de la base de datos, aplicándose en este, ciertas técnicas de preparación y limpieza de los datos. Una vez que los datos estén listos, se llevará a cabo la **estimación** de los **modelos de clasificación.**

En el segundo punto se desarrollará todo el proceso de elaboración de los **modelos de clasificación**, así como la introducción de los **resultados** obtenidos en estos. El estudio se cerrará con el ultimo y tercer punto, **comparaciones y conclusiones de los modelos**. Dichas **comparaciones** se llevarán a cabo en base a cuatro indicadores; La **especificidad de los modelos**, la propia **precisión** de los mismos o *accuracy* y por último, una evaluación de la bondad, utilizando la **curva ROC** y el índice o indicador **Kappa**. Por último, las **conclusiones** reunirán las distintas **deducciones** a las que se ha llegado en una forma estructural, en base a todo el tratamiento y trabajo realizado con los datos y las **variables seleccionadas**.

1. METODOLOGÍA.

Teniendo en cuenta que la finalidad de este estudio se centra en la identificación de pacientes con un mayor riesgo de sufrir un infarto cerebral, el diseño de este se ha orientado a la elaboración de **modelos de clasificación**, el cual permitirá, asignar una etiqueta o categoría a una nueva observación, en base a las categorías o atributos seleccionados. Es decir, el modelo de clasificación podrá predecir a qué categoría pertenecerá un nuevo paciente en base a una serie de características que se desarrollarán a lo largo del estudio.

1.1 VARIABLES OBJETO DE ESTUDIO.

Antes de comenzar con el desarrollo del **modelo de clasificación predictivo**, se debe realizar una presentación de las **variables** que van a ser de la partida del estudio, así como un análisis exploratorio. Para que, de esta manera, se pueda comprender con precisión los resultados obtenidos en el modelo. Para ello, en la tabla 1.1 se recogen las distintas **variables** que entrarán en la especificación del modelo.

Tabla 1.1: Variables de estudio

Nombres	Descripción	Fuente
gender	Género de los individuos	Kaggle
hypertension	Si sufre o no de hipertensión	Kaggle
heart_disease	Si sufre o no de enfermedad cardiaca	Kaggle
ever_married	Estado civil de los individuos	Kaggle
work_type	Situación laboral	Kaggle
residence_type	Lugar de residencia	Kaggle
smoking_status	Fumadores o no fumadores	Kaggle
stroke	Si sufre o no infarto cerebral	Kaggle
age	Edad	Kaggle
bmi	Indice de masa corporal	Kaggle
avg_glucose_level	Nivel medio de glucosa en sangre	Kaggle

Como en cualquier conjunto de datos, las **variables** que lo componen presentan ciertas características. Normalmente, las variables pueden dividirse en dos grandes grupos, **variables categóricas** y **variables numéricas**. En primer lugar, con respecto a las variables categóricas, son un tipo de variables que se utilizan para dividir a una población en categorías o grupos. Algunas características que pueden utilizarse para describir una variable categórica es que sus valores pueden ser limitados y dividirse en distintas categorías. En este estudio, existen varias variables que siguen estas características como son; *gender* (Género de los individuos), *hypertension* (Si sufre o no de hipertension), *heart_disease* (Si sufre o no de enfermedad cardíaca), *ever_married* (Estado civil de los individuos), *work_type* (Situación laboral), *residence_type* (Lugar de residencia), *smoking_status* (Tipo de fumadores) y *stroke* (Si sufre o no infarto cerebral).

En segundo lugar, las **variables numéricas** se caracterizan por ser **variables** que se utilizan para medir u observar una cantidad o magnitud. Alguna de las numerosas características que permiten describir a una **variable numérica** pueden ser las siguientes; son **variables** que pueden tomar cualquier valor dentro de un rango determinado (numérica continua), además, pueden tomar también valores específicos o discretos (numérica discreta). Algunas de las **variables especificadas** de este estudio, siguen estas características, concretamente 3 y son las siguientes; *age* (Edad), *bmi* (Índice de masa corporal) y *avg_glucose_level* (Nivel medio de glucosa en sangre).

1.2 MUESTRA.

La **muestra**, se caracteriza por tener un total de **11 variables**, cada una de ellas recoge un total de **4981 observaciones**. Tal y como se ha introducido anteriormente, existe variedad en los datos, es decir, variables con distinto tipo de naturaleza, principalmente dos, categóricas y numéricas. A lo largo del estudio, se aplicarán diferentes **técnicas estadísticas** para poder conocer en mayor profundidad los datos y, en caso de ser necesario, realizar diferentes **transformaciones** en estos, para de esta manera, evitar distorsiones en los **resultados**.

1.3 ANÁLISIS EXPLORATORIO:

Una vez realizada la presentación de las **variables**, así como la descripción de la **muestra**, es fundamental dar paso al **análisis exploratorio**, para así, conocer los datos que se van a tratar antes de comenzar a construir los **modelos de clasificación** pertinentes. Realizar un **análisis exploratorio** de los datos antes de entrenar un modelo de predicción es importante por varias razones:

- **Conocer el conjunto de datos:** El **análisis exploratorio** permite conocer mejor el conjunto de datos con el que se está trabajando, incluyendo el número de **observaciones y variables**, la estructura, forma de los datos y cualquier patrón o tendencia que puedan presentar. Esto ayudará a elegir el **modelo de predicción** adecuado y, dar **solución** al problema de manera más precisa.

- **Detectar problemas de calidad de los datos:** Aplicar **técnicas** de este tipo, ayudará a detectar problemas en estos, tales como, valores atípicos o perdidos. Si no se abordan adecuadamente este tipo de situaciones, no solo puede afectar negativamente al rendimiento del modelo, si no también, a la mera interpretación de los **resultados**.

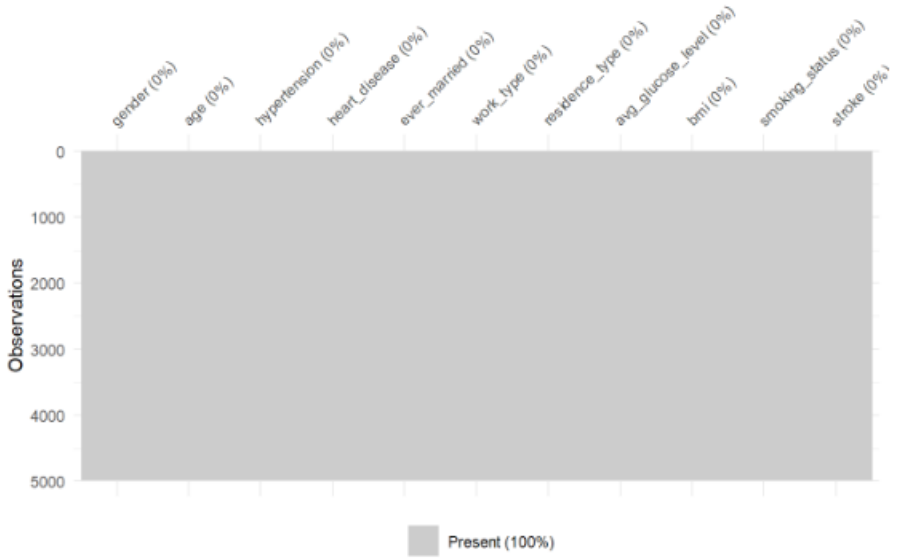
- **Mejorar el rendimiento del modelo:** Identificación de patrones y relaciones en los datos que puede ser útil para mejorar el rendimiento del modelo. Por ejemplo, se pueden encontrar variables que estén altamente correlacionadas entre sí, lo que permitiría simplificar y, por tanto, seleccionar solamente una de ellas en el modelo en lugar de incluir ambas, reduciendo la complejidad de este y a su vez, mejorar su rendimiento.

En resumen, el **análisis exploratorio** es una parte esencial del proceso de modelado de **predicción**, ya que, en base a todos los aspectos mencionados anteriormente, permite conocer mejor el conjunto de datos, detectar problemas de calidad en estos y mejorar el rendimiento del modelo.

1.3.1 Missing Values y Outliers.

Teniendo en cuenta todo lo anterior, se procede a comenzar con el **análisis de la existencia** de **missing values** (valores perdidos) y **outliers** (valores atípicos). Es importante para el estudio, que, en la muestra, todas las variables presenten valores, es decir, que no existan valores perdidos, para ello, todas las variables han sido pasadas por un proceso de filtrado para la detección de este tipo de casos, en el Gráfico 1.1 se presentan los resultados obtenidos en este proceso de detección de **valores perdidos**.

Gráfico 1.1: Representación de Missing values



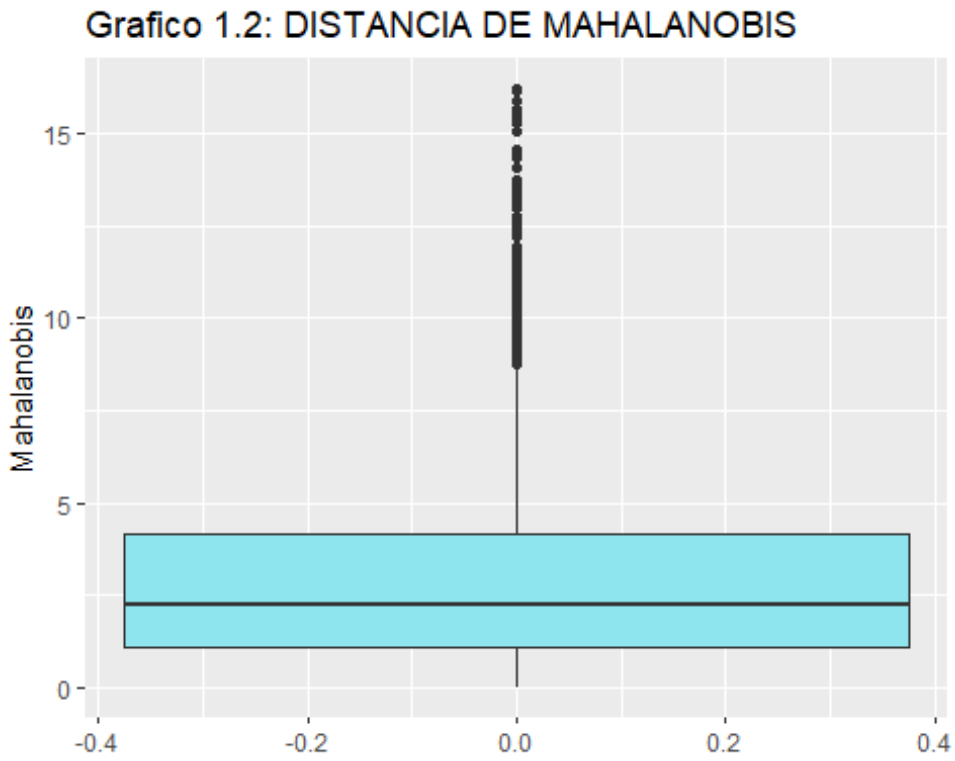
De los resultados del proceso de detección de valores perdidos (missing values) mostrados en el gráfico 1.1, se puede comprobar que no existe ningún valor nulo o perdido en la base de datos, es decir, la base de datos presenta el 100% de las observaciones, punto muy positivo para el análisis, ya que no es necesario la eliminación de ninguna de estas.

1.3.2 Distancia de Mahalanobis

En cuanto al análisis de posibles **outliers o valores atípicos**, cabe destacar que la muestra presenta un gran número de observaciones, punto muy a tener en cuenta ya que existen varias técnicas para la búsqueda de este tipo de datos. En otro tipo de situación (sobre todo en aquellas en las que la muestra tiene pocos datos) lo más óptimo sería realizar una valoración individual para cada variable, pero en este caso, al tener una base de datos que cuenta con un total de 11 variables, esta opción se vuelve inviable en términos de costes y optimización de tiempo.

Es por ello, que se ha decidido aplicar la técnica de la **"Distancia de Mahalanobis"**. Esta técnica describe la distancia entre cada punto de datos y el centro de masa. Cuando un punto se encuentra en el centro de masa, la **distancia de Mahalanobis** es igual a cero y cuando un punto de datos se encuentra distante del centro de masa, la distancia es mayor a cero. Por lo tanto, los puntos de datos que se encuentran lejos del centro de masa se consideran **valores atípicos**.

La **distancia de Mahalanobis** se calcula para cada **observación** en el conjunto de datos, dándole a cada una un peso como inverso de la distancia de Mahalanobis. Las **observaciones** con valores extremos obtienen menores pesos. Finalmente, se ejecuta una **regresión ponderad**a para minimizar el efecto de los valores extremos. En el Gráfico 1.2, se muestran los resultados correspondientes a este proceso de detección de **valores atípicos**.



Como puede observar en el gráfico 1.2, existen muchos casos de **valores atípicos**, dato que a priori puede parecer muy negativo, pero es importante tener en cuenta que, al tratarse de una base de datos médica, estos **casos atípicos**, pueden determinar en gran medida si una persona que acude al hospital y puede sufrir un infarto cerebral. Por todas estas razones, se ha decidido no proceder a su eliminación y, por tanto, seguir con en análisis sin descartarlos, pero teniendo en cuenta el **gran porcentaje** que representan.

1.3.3 Remuestreo

Una vez terminados los procesos de detección de valores perdidos y atípicos, es importante conocer en profundidad de manera individual, la **variable predictora** de este estudio, es decir, la variable *stroke* (Si sufre o no infarto cerebral). En este caso particular, *stroke* cuenta con solo un 5.24% de casos positivos, mientras que el resto, son casos negativos.

Esto puede inducir a problemas a la hora de construir un **modelo de clasificación** por la escasa información del conjunto de datos más importante y del que está centrado este análisis. Además, esto también puede provocar un problema de **interpretabilidad**, debido a la alta posibilidad de obtener un modelo con un rendimiento óptimo para clasificar a los individuos que no van a sufrir infarto cerebral y un pésimo rendimiento para el caso contrario, teniendo un muy buen nivel de precisión global.

Por ello, se va a proceder a aplicar la técnica de remuestreo **Smote** (*Synthetic Minority Oversampling Technique*) en los casos positivos, de forma que la repartición sea más representativa, llevando a los casos positivos a representar un 20% de la muestra.

SMote es una técnica de remuestreo utilizada para tratar desequilibrios de clases en un conjunto de datos. Cuando hay un desequilibrio de clases, significa que hay una clase que es mucho más numerosa que otras, lo que puede afectar negativamente al rendimiento del modelo de clasificación.

Esta técnica funciona creando **observaciones sintéticas** para la clase minoritaria utilizando información de las observaciones existentes de esa clase. Para hacer esto, SMote utiliza un procedimiento que consiste en los siguientes pasos:

1. Selecciona dos observaciones aleatorias de la clase minoritaria.
2. Calcula la diferencia entre las dos observaciones seleccionadas.
3. Genera una nueva observación sintética ubicada a una distancia aleatoria entre las dos observaciones seleccionadas en la dirección de la diferencia calculada.

Repite los pasos 1-3 hasta que se haya generado el número deseado de observaciones sintéticas.

De esta manera, SMote permite **aumentar el tamaño** de la clase minoritaria sin tener que recopilar más datos reales, lo que puede ser útil en casos en los que es difícil o costoso obtener más datos. En la tabla 1.2, se muestran los resultados relacionados con la nueva distribución de casos para la variable *stroke*.

Tabla 1.2: Proporción del remuestreo

	stroke	n	prop
1	No	4733	0.8
2	Yes	1184	0.2

Ahora, el *data.frame* que se usará para predecir con la mayor exactitud posible la pertenencia a un grupo u otro dentro de `stroke` será *brain_oversampled.*

1.3.4 Agrupación de datos

Realizada la **técnica de remuestreo**, es interesante observar y conocer, como se agrupan los datos de las **variables explicativas** con respecto al termino **independiente** (`stroke`). Para ello, se han elaborado una serie de gráficos para facilitar la visualización y comprender la distribución de las variables, además de detectar patrones o tendencias en los datos.

Grafico 1.3: Agrupación de datos

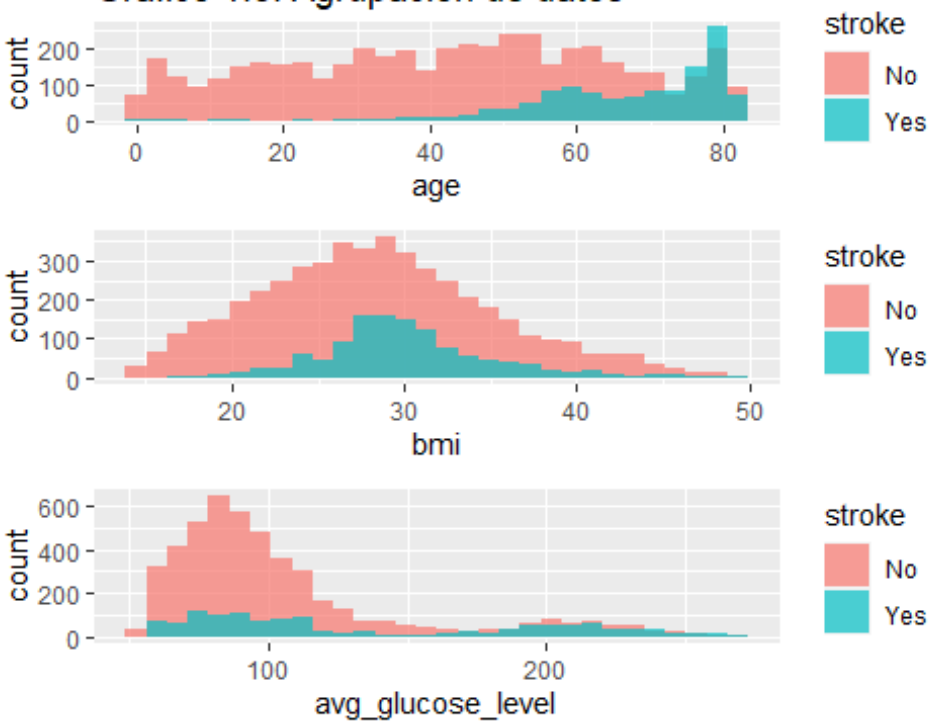


Grafico 1.4: Agrupación de datos

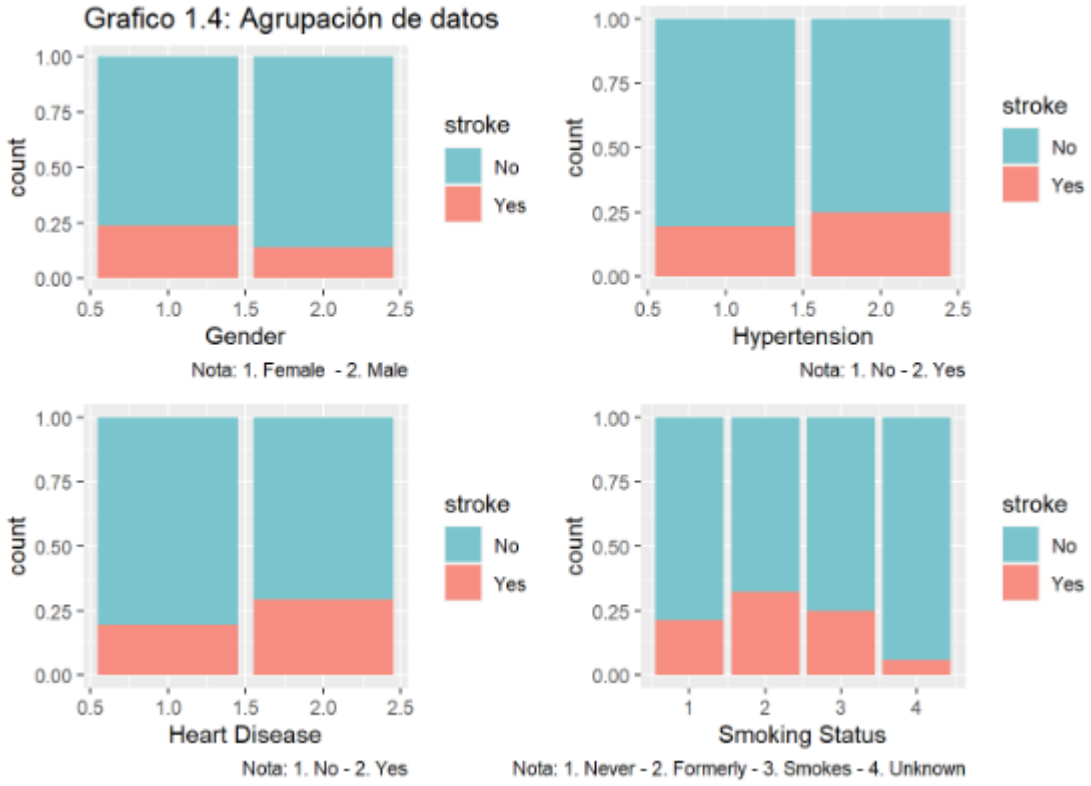
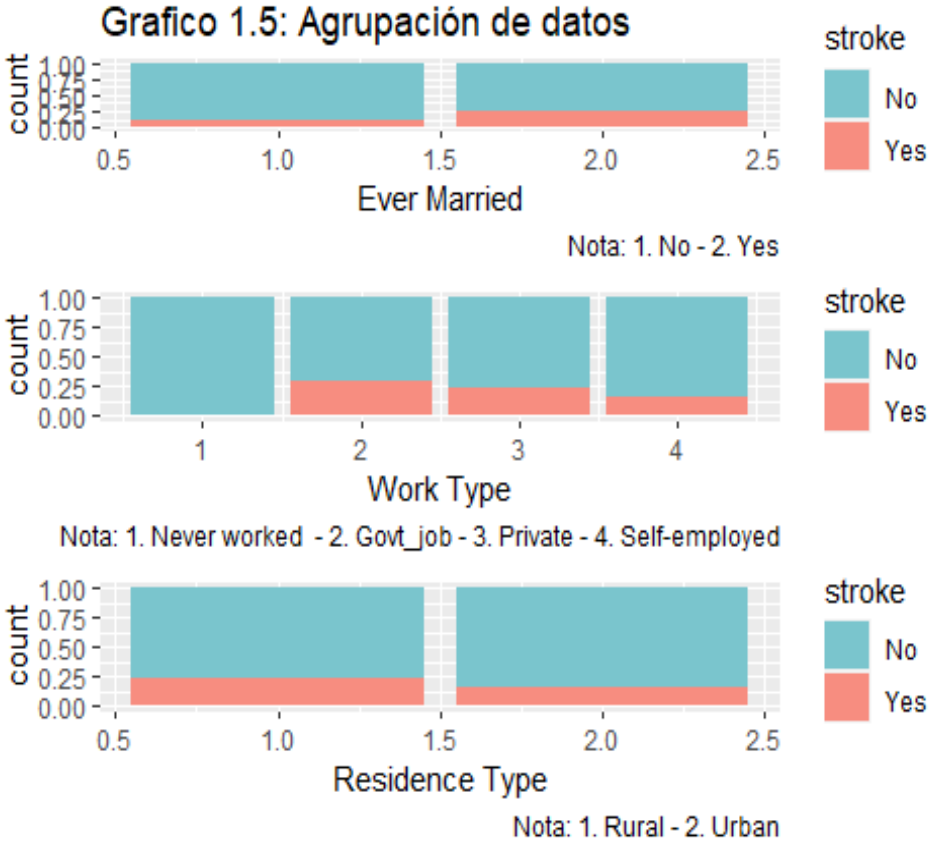


Grafico 1.5: Agrupación de datos



Observando los resultados de los diferentes gráficos y en base a ciertas características como pueden ser la forma de la **distribución**, su **amplitud** y **asimetría**, se puede llegar a varias conclusiones. Con respecto al Gráfico 1.3, la forma de la **distribución** para la variable *age* (edad) no sigue una distribución simétrica para ninguno de los dos casos (si sufre o no infarto cerebral), sin embargo, para la variable *bmi* (índice de masa corporal) es totalmente distinto, los resultados de su distribución si que siguen una estructura más simétrica, aunque con una ligera desviación a la izquierda. Con respecto a la variable *avg_glucose_level*, (Niveles de glucosa media en sangre) se aprecia que su distribución se encuentra totalmente sesgada a la izquierda.

En cuanto a la **amplitud de la distribución**, cada variable ocupa un rango distinto. La variable *age* (edad), tiene un rango general que va desde individuos recién nacidos (meses de vida) hasta individuos de avanzada edad, llegando a superar los 80 años, sin embargo, los datos relaciones a aquellos individuos que si sufren de infarto cerebral siguen una amplitud muy acotada, la mayoría de la muestra se encuentra entre los 40 y 80 años de edad (distribución sesgada a la derecha). Con respecto a la variable *bmi* (índice de masa corporal) la situación es algo distinta, se observa como la **amplitud** de los datos (para los casos que si sufren infarto cerebral) vuelve a estar acotado en unos niveles de 20 y 40 de índice de masa corporal, centrándose la mayoría de los datos en el centro del gráfico, concretamente, en unos valores ligeramente inferiores a 30, en otras palabras, la mayoría de pacientes que sufren de **infarto cerebral** (stroke) se encuentran en una situación de sobrepeso y obesidad. Por último, la variable *avg_glucose_level* (Niveles medios de glucosa en sangre) sigue una **amplitud en su distribución** totalmente distinta a las otras dos variables explicadas anteriormente, la mayoría de casos (individuos que sufren infarto cerebral), se centran en la parte izquierda del gráfico, dato muy curioso, porque son valores que no superan (la mayoria) los **100mg/dl**, valores que se caracterizan por ser considerados normales en una persona saludable.

Siguiendo con el análisis, en los Gráfico 1.4 y 1.5, se observan **distribuciones** y **amplitudes** totalmente distintas. Hay que recordar, que en estos gráficos se está trabajando con **variables categóricas**, por tanto, la estructura visual con respecto al Gráfico 1.3 es diferente.

1.3.5 Matriz de correlación.

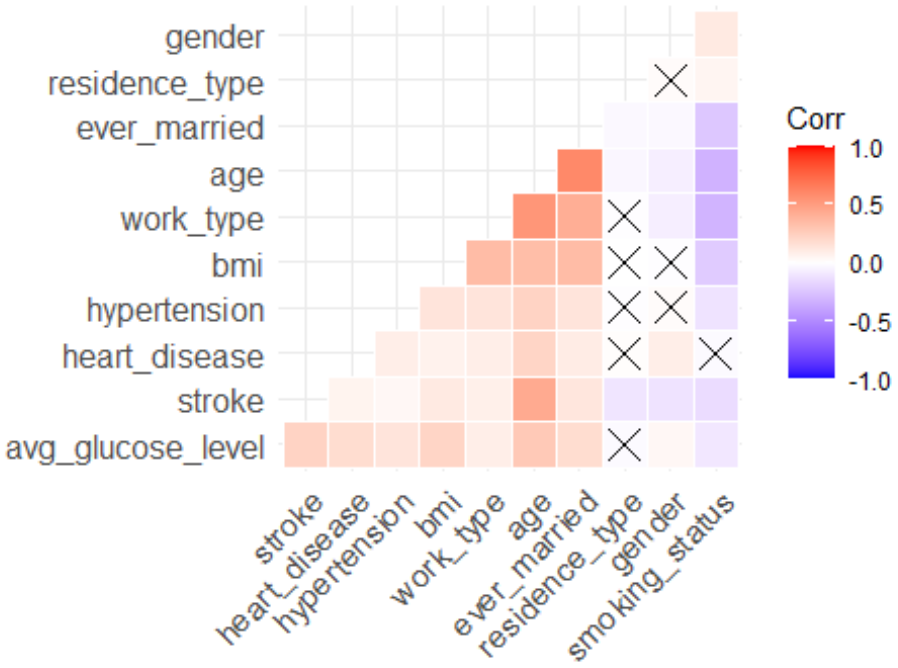
Analizar la **correlación entre las variables** es importante en un **análisis exploratorio** de los datos por varias razones:

- **Ayuda a entender mejor el conjunto de datos:** El análisis de la correlación entre las variables permite ver cómo se relacionan entre sí y si hay patrones o tendencias que puedan ser útiles para entender mejor el conjunto de datos.
- **Identificar variables que pueden ser redundantes:** Si hay dos o más variables altamente correlacionadas entre sí, es posible que solo una de ellas sea necesaria para incluir en el modelo. Esto puede ayudar a reducir la complejidad del modelo y mejorar su rendimiento.

- **Ayuda a elegir el modelo adecuado:** Algunos modelos son más adecuados para trabajar con variables altamente correlacionadas que otros.
- **Ayuda a evitar problemas de multicolinealidad:** La multicolinealidad ocurre cuando dos o más variables están altamente correlacionadas entre sí. Esto puede afectar negativamente al rendimiento del modelo y hacer que los resultados sean menos precisos. El análisis de la correlación entre las variables ayudará a detectar la multicolinealidad y a tomar medidas para abordarla.

Antes de realizar la matriz de correlaciones, es importante que las variables pasen por un proceso de transformación. Hay que recordar que en nuestra base de datos cuenta con variables numéricas y variables categóricas, por lo que, para poder hacer la matriz de correlaciones, todas las variables deben de estar caracterizadas como variables numéricas. Una vez realizado este proceso, se procede a realizar la matriz. En el Gráfico 1.6 se muestran los resultados obtenidos:

Gráfico 1.6: Matriz de correlación



Nota: Las X representan los P-valores no significativos.

Como puede apreciarse de manera general en el gráfico 1.6, la matriz de correlación **no presenta correlaciones altas** (>0.80), además, poniendo el foco a su vez en la matriz de correlaciones de los p-valores, se observa que la variable *residence_type* no aporta ninguna información.

Entrando en detalle y con relación a lo realmente importante del estudio, la variable que guarda mayor correlación con la variable *stroke*, es la variable *age*, un resultado bastante lógico teniendo en cuenta que, a mayor edad, mayor probabilidad existe de que una persona sufra un **derrame cerebral** si no ha seguido unos hábitos saludables

de vida. Otro de los valores importantes a tener en cuenta para el análisis es la alta correlación que presentan *age* y *ever_married*, que podrían contener información redundante en muchos casos.

1.3.6 Normalización de variables

Cuando se estima un modelo de clasificación, es interesante que las variables que sigan una distribución normal por varias razones:

- **Muchos modelos se basan en el supuesto de que las variables siguen una distribución normal:** Varios de los modelos de aprendizaje automático asumen o se basan, en que las variables seguirán una distribución normal, ya que, en caso contrario el rendimiento de este podría verse afectado.

- **Ayuda a evitar problemas de sesgo:** Si las variables no siguen una distribución normal, es posible que algunas de estas tengan más influencia en el modelo que el resto de variables, debido principalmente, a la forma en la que se encuentran distribuidas. Por ejemplo, si una variable tiene una distribución muy sesgada hacia la derecha, es posible que tenga más influencia en el modelo que otras variables. Al utilizar variables que sigue una distribución normal, se evita este problema.

- **Facilita la interpretación de los resultados:** Si las variables siguen una distribución normal, es más fácil realizar una mejor interpretación de los resultados del modelo, ya que permitiría realizar una serie de técnicas estadísticas convencionales, que ayudarían en gran medida a desarrollar una mejor evaluación de la significación de los coeficientes de dicho modelo.

Sin embargo, un aspecto a tener en cuenta es que, aunque la normalidad es importante, **no siempre es posible y necesario** utilizar variables que sigan una distribución normal para que el modelo sea efectivo, es más, existen muchos modelos que pueden funcionar de manera muy eficiente con variables que no siguen este tipo de distribución.

Por tanto, teniendo en cuenta todo lo anterior, el estudio de la **normalidad** en este análisis se centrará únicamente en tres variables, que son precisamente las únicas **variables numéricas** que presenta la base de datos, que son; *age* (Edad), *bmi* (Índice de masa corporal) y *avg_glucose_level* (Nivel medio de glucosa en sangre). Este estudio se divide claramente en dos partes; una primer parte en la que, mediante la aplicación de una serie de pruebas gráficas y contrastes, se analizará si dichas variables siguen o no, una **distribución normal** y, una segunda parte en la que, en caso de obtener unos resultados que devuelvan que algunas de las tres variables no sigue una distribución normal, proceder a forzar la normalidad aplicando a dichas variables un proceso de transformación y, comprobando de nuevo, tanto gráficamente como con contrastes de hipótesis, si se ha conseguido el objetivo de transformación o no.

Para esta primera parte, se van a introducir (para cada variable) dos pruebas gráficas de normalidad, concretamente, histogramas y gráficos quantil-quantil. Los histogramas

son gráficos que representan la frecuencia de cada valor en un conjunto de datos, en caso de que las variables sigan una **distribución normal**, el histograma debería de presentar una forma de campana, con la mayoría de los valores concentrados en el centro del gráfico y, poco a poco disminuyendo hacia los extremos de este, en caso de no presentar dicha forma, podría ser una señal de que los datos no se distribuyen de una manera normal. Con respecto a los gráficos Q-Q, se puede decir que su principal función es la de comparar los quantiles de la distribución de las variables seleccionadas con los quantiles de una **distribución normal**. Si los quantiles de la distribución de las variables forman una línea recta en diagonal, entonces, podría decirse que los datos siguen una **distribución normal**, en caso de no ser así, puede ser un indicio de todo lo contrario. En el gráfico 1.7 que se muestran a continuación, puede observarse todo lo anterior mencionado.

Gráfico 1.7 : Histograma y gráfico Q-Q. Variable age.

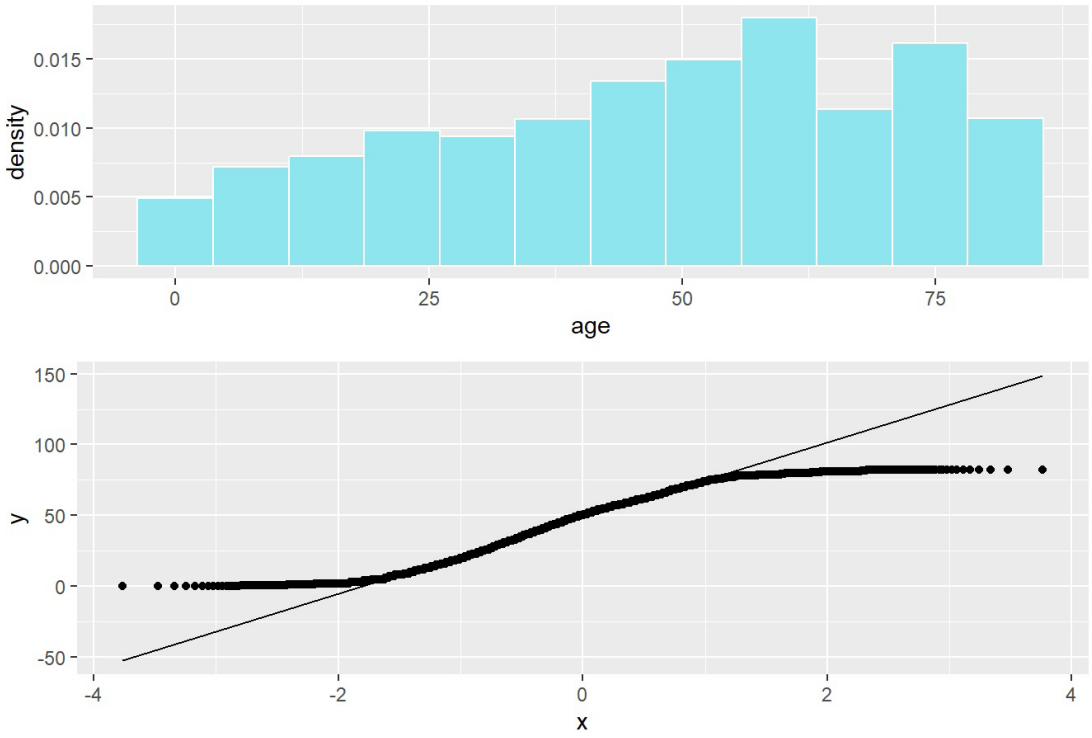


Grafico 1.8 : Histograma y gráfico Q-Q. Variable bmi.

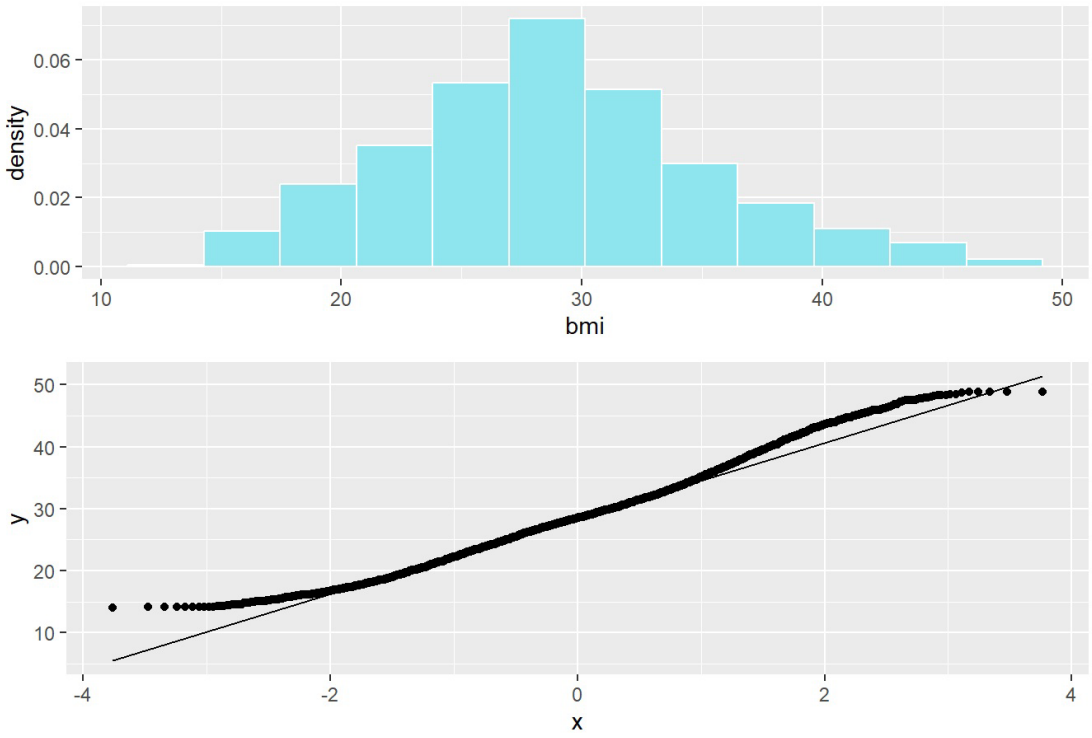
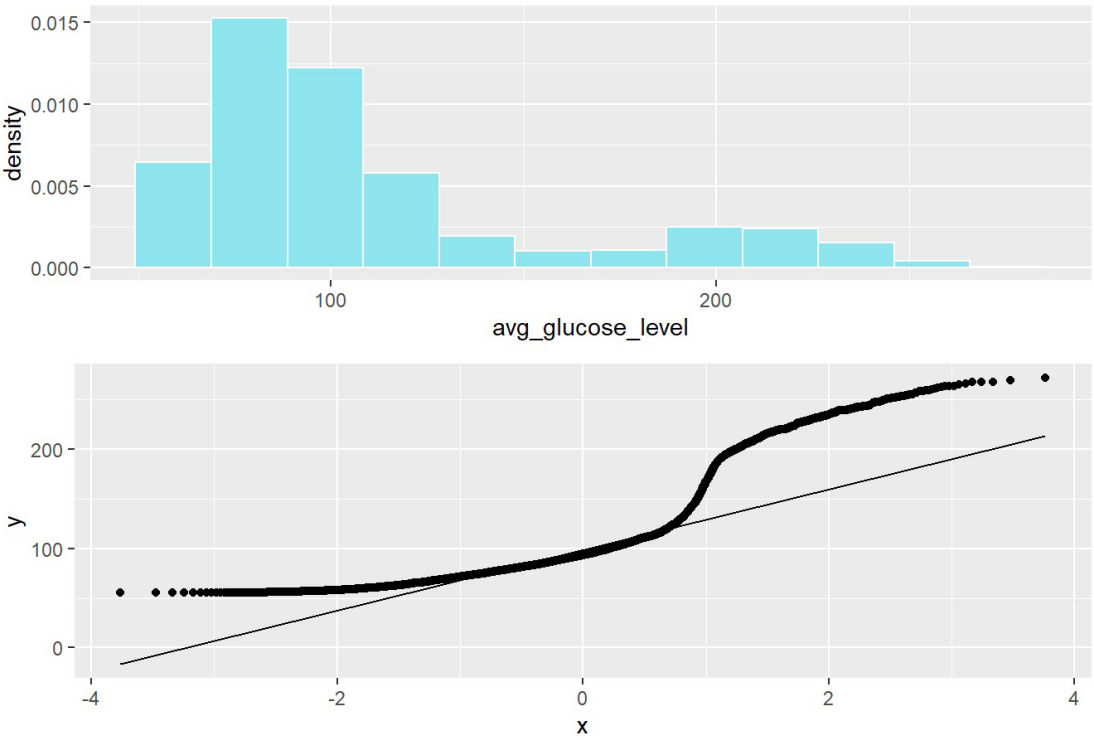


Grafico 1.9 : Histograma y gráfico Q-Q. Variable avg_glucose_level.



Los resultados que muestran los gráficos (1.7, 1.8 y 1.9), se podría llegar a la conclusión de que las variables se encuentran muy lejos de poder acercarse a seguir una **distribución normal**, exceptuando el caso de la variable *bmi* que, en un principio, si parece que se ajusta mejor a una **distribución normal** con respecto al resto de variables. En cualquier caso, el estudio de la normalidad no termina aquí, es importante profundizar más en este. Para ello, es imprescindible aplicar contrastes de hipótesis para examinar más en detalle si existe o no, normalidad en alguna de las variables.

1.3.7 Contraste de Hipotesis:

Como se ha podido observar en el análisis gráfico de la normalidad, ambos métodos son muy interesantes ya que, además de ser visuales, resultan muy fáciles de entender y permiten realizar una evaluación rápida de si una variable, sigue o no, una **distribución normal**. Pero no son suficientes, para poder corroborar al 100% que una variable sigue una distribución normal o no, es necesario ir más allá, es decir, siempre es importante comprobar mediante contrastes de hipótesis que lo que se aprecia visualmente está representando la realidad y no está induciendo a engaño o malas interpretaciones. Para evitar esto, se ha decidido aplicar el método de contraste de hipótesis de **Anderson-Darling** en lugar del famoso contraste de **Shapiro-Wilk**, principalmente, debido a que el *data. frame* tiene más de 5000 observaciones y por tanto, el contraste de **Shapiro-Wilk** se encuentra limitado con tal número de observaciones para poder hacer la prueba de normalidad de manera eficiente.

```
##  Anderson-Darling normality test
##
## data:  brain_oversampled$bmi
## A = 12.374, p-value < 2.2e-16

##  Anderson-Darling normality test
##
## data:  brain_oversampled$age
## A = 66.376, p-value < 2.2e-16

##  Anderson-Darling normality test
##
## data:  brain_oversampled$avg
## A = 418.07, p-value < 2.2e-16
```

Como se puede apreciar, los resultados que aporta el contraste de hipótesis de **Anderson-Darling** son francamente negativos en términos de normalidad. Ninguna de la variables numéricas del *data.frame* sigue una **normal** estrictamente hablando. Como bien se ha mencionado y según la teoría estadística, que las variables del estudio no sigan una **distribución normal** es, por diversas razones, algo que podría causar ciertos problemas, sobre todo en el rendimiento en los modelos de clasificación, por lo que, ante esta situación, es imprescindible pasar a la segunda parte de este análisis de normalidad, que se corresponde básicamente con el intento de transformación de las variables para que sigan una **distribución normal**.

1.3.8 Transformaciones

Para la transformación de las variables se ha seleccionado el método **Box-Cox**. Este método se basa en la idea de que muchas variables no se distribuyen de manera normal porque tienen una escala que no es adecuada para los datos. Por ejemplo, una variable que toma valores entre 0 y 1 no se distribuirá de manera normal porque la escala es muy pequeña. En términos teóricos, **Box-Cox** trabaja basándose en un parámetro llamado lambda, el cual, se utiliza para transformar la variable. Si dicho parámetro lambda es igual a 0, la transformación es una raíz cuadrada, si lambda es igual a 1, la transformación es un logaritmo, si lambda es diferente de 0 o 1, la transformación es una combinación de estos dos casos. El valor de lambda que se utiliza depende siempre de la forma de la distribución de los datos de las variables.

Una vez que se ha aplicado la transformación de **Box-Cox**, se evaluará de nuevo, si las variables se distribuyen de manera **normal** utilizando las mismas técnicas utilizadas anteriormente, como han sido los histogramas y gráficos quantil-quantil. Si los datos siguen una distribución normal después de la transformación, se puede proceder a estimar el modelo de clasificación. Es importante tener en cuenta que el método de **Box-Cox** no siempre produce una **distribución normal**, en algunos casos, es posible que los datos sigan una distribución que no es normal incluso después de la transformación. En estas situaciones, es posible que sea necesario utilizar otras técnicas para transformar los datos, utilizar modelos de clasificación que no requieran que los datos se distribuyan de manera normal o, asumir **normalidad en las variables** (siempre que sea lógica asumirla) para poder aplicar **modelos de clasificación** que requieren de esta característica para ser efectivos.

```
## bcPower Transformation to Normality
##                         Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## brain_oversampled$age     0.954        0.95       0.9175       0.9905
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                               LRT df       pval
## LR test, lambda = (0) 4360.993  1 < 2.22e-16
##
## Likelihood ratio test that no transformation is needed
##                               LRT df   pval
## LR test, lambda = (1) 6.000731  1 0.0143

## bcPower Transformation to Normality
##                         Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## brain_oversampled$bmi    0.4601         0.5       0.3673       0.5529
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                               LRT df       pval
## LR test, lambda = (0) 95.19905  1 < 2.22e-16
##
```

```
## Likelihood ratio test that no transformation is needed
##                             LRT df       pval
## LR test, lambda = (1) 128.4211  1 < 2.22e-16

## bcPower Transformation to Normality
##                                      Est Power Rounded Pwr Wald Lwr Bnd
## brain_oversampled$avg_glucose_level   -1.0058          -1      -1.0728
##                                      Wald Upr Bnd
## brain_oversampled$avg_glucose_level      -0.9389
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                             LRT df       pval
## LR test, lambda = (0) 900.0233  1 < 2.22e-16
##
## Likelihood ratio test that no transformation is needed
##                            LRT df       pval
## LR test, lambda = (1) 3592.94  1 < 2.22e-16
```

Realizadas las **transformaciones**, el siguiente paso es el de comprobar (de nuevo) mediante histogramas, gráficos Q-Q y contrastes de hipótesis, si realmente este método de transformación ha sido efectivo o de lo contrario, se tendrá que asumir normalidad en las variables para poder realizar ciertos modelos de clasificación.

Grafico 1.10 : Histograma y gráfico Q-Q. Variable age transformada.

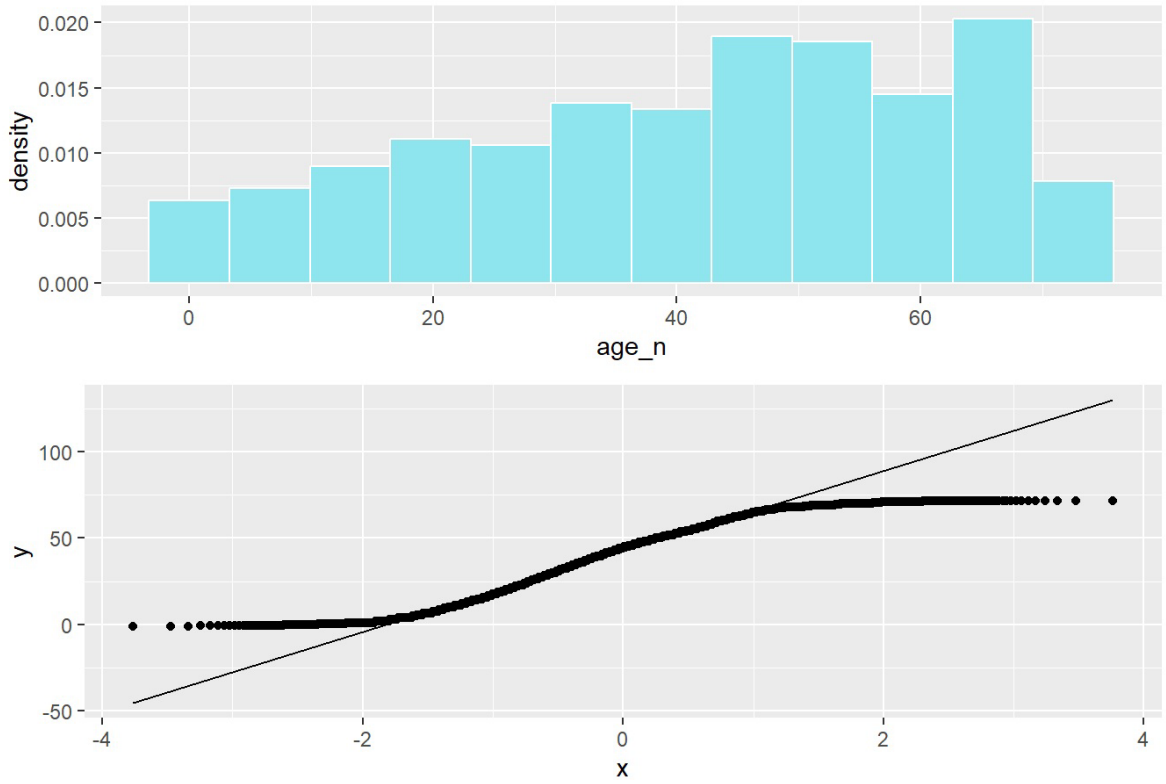


Grafico 1.11 : Histograma y gráfico Q-Q. Variable bmi transformada.

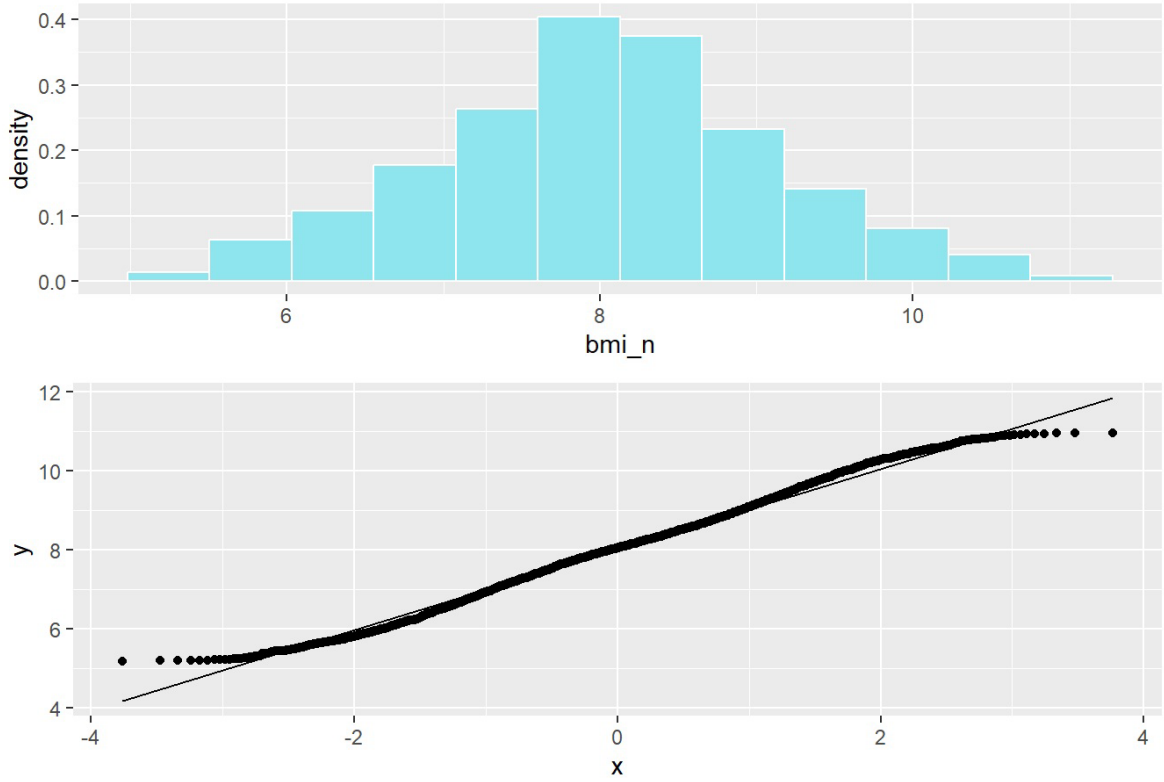
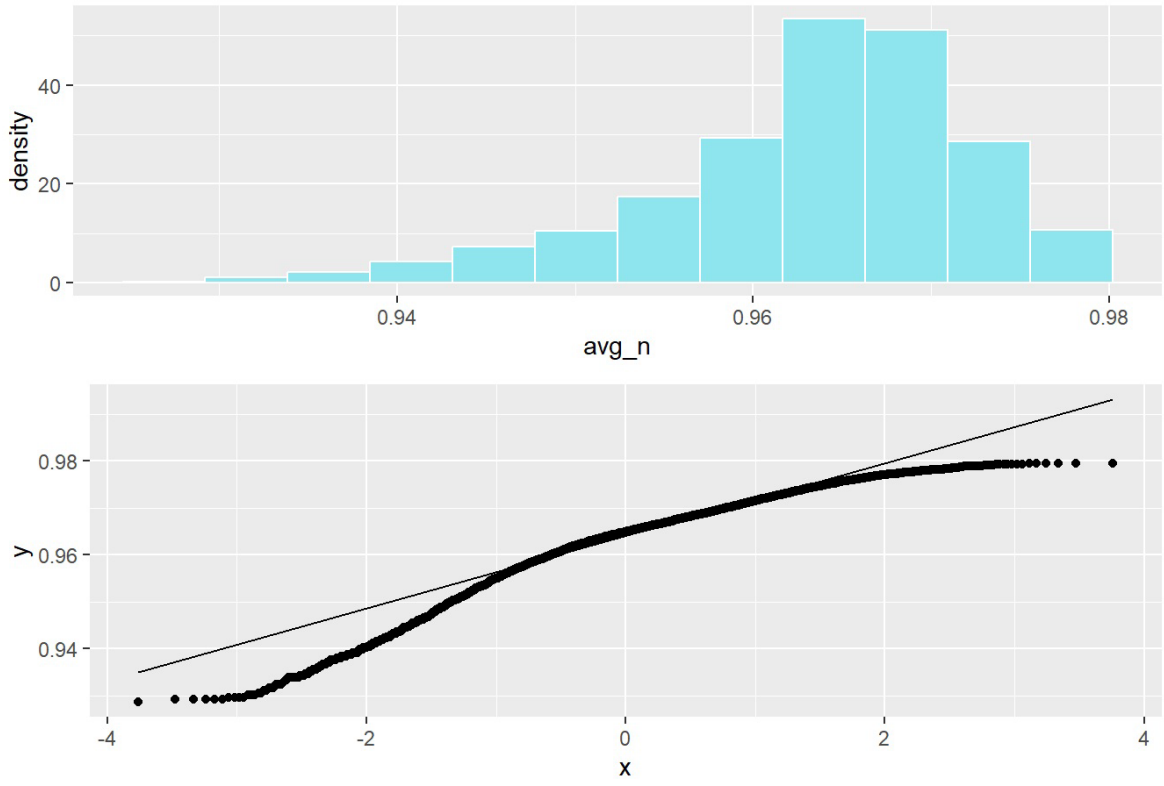


Grafico 1.12 : Histograma y gráfico Q-Q. Variable avg_glucose_level transformada.



1.3.9 Nuevo contraste de hipótesis.

```
##
##  Anderson-Darling normality test
##
## data:  brain_ovn$bmi_n
## A = 5.1872, p-value = 8.253e-13

##
##  Anderson-Darling normality test
##
## data:  brain_ovn$age_n
## A = 68.757, p-value < 2.2e-16

##
##  Anderson-Darling normality test
##
## data:  brain_ovn$avg_n
## A = 87.712, p-value < 2.2e-16
```

Como se puede apreciar en los resultados que aportan los gráficos 1.10, 1.11 y 1.12 y, los contrastes estadísticos, el método **Box-Cox** no ha sido capaz de convertir en normal ninguna de las variables, debido seguramente, a la alta presencia de **outliers** sumado a la gran cantidad de datos. Con todo esto, hay que tener en cuenta algo importante y es que, se tratan de datos pertenecientes a individuos que siempre van a tomar valores dentro de unos límites por lo que, se puede asumir cierta normalidad en ellos. Por tanto, finalmente se seguirá con el estudio asumiendo que las variables siguen **cierta normalidad**, aunque vaya en contra de la teoría estadística y pese a que se ha realizado un intento de transformación, se seguirá el estudio con las variables que no han sido transformadas, se hará así debido a varios motivos:

1. **Perdida de interpretabilidad:** La distribución de los datos ha cambiado, así como su escala y para generar respuestas tras hacer el modelo predictivo es más costoso de interpretar.

2. **Se puede asumir cierta normalidad:** Como se ha comentado anteriormente, los datos siempre van a darse entre ciertos límites y, además, hay una gran cantidad de ellos, por lo que es más difícil que puedan seguir una normal pese a que lo parezca a simple vista, como puede ser el caso de *bmi*.

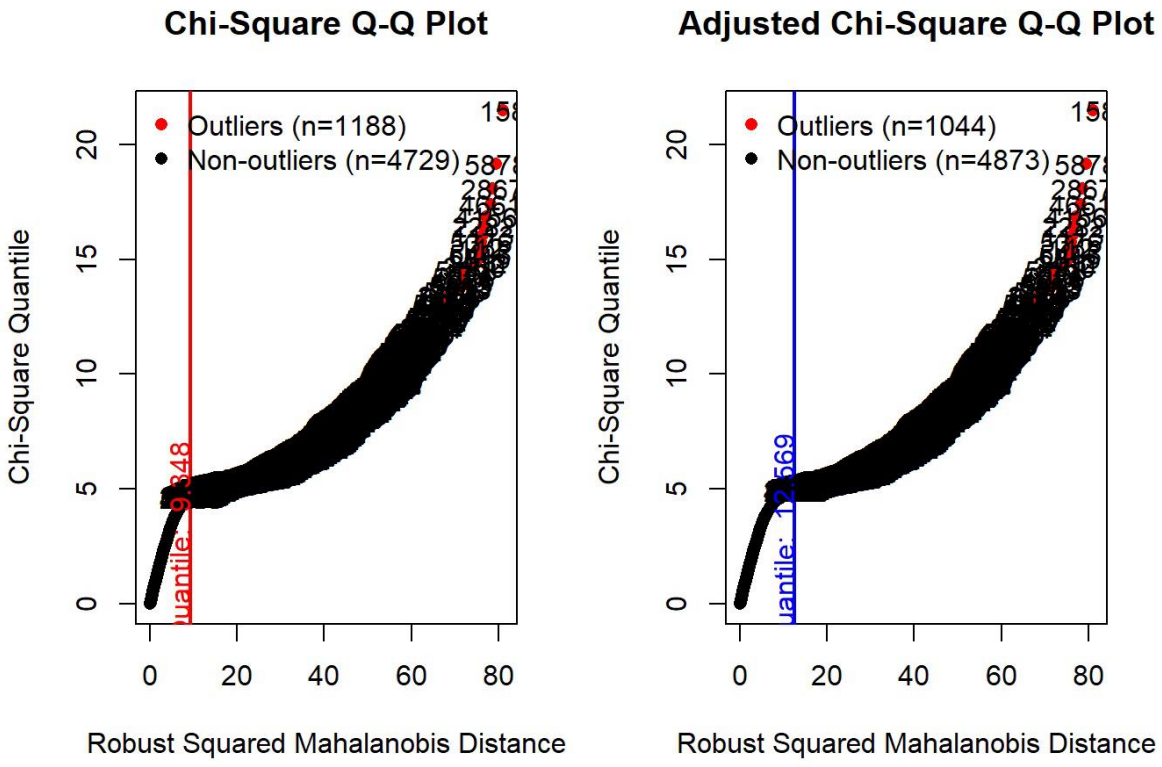
3. **Muchos de los modelos usados son robustos:** Los modelos seleccionados para el análisis presentan, en su mayoría, cierta robustez a la ausencia de normalidad y algunos de ellos en sus procesos incluyen procedimientos de tipificación para eliminar problemas de escala. De hecho, los dos modelos que mejor clasifican `stroke` son modelos bastante robustos.

4. **Pérdida de precisión:** Aunque teóricamente no debería ser así, tras hacer pruebas con estas variables transformadas se pierde tanto precisión como

especificidad (*véase en el anexo*) por lo que pierde aún más sentido continuar con ellas.

1.3.10 Normalidad multivariante.

En el caso de la clasificación predictiva, es posible que **no sea necesario** verificar la normalidad multivariante de los datos antes de entrenar un modelo. Esto se debe a que muchos algoritmos de clasificación, como los árboles de decisión y las máquinas de vectores de soporte (SVM), no asumen que los datos siguen una distribución normal y, por lo tanto, pueden funcionar bien incluso si los datos no son normales. Sin embargo, verificar la **normalidad multivariante** de los datos **puede ser útil** en ciertas circunstancias, como, por ejemplo, si está utilizando un algoritmo de clasificación que asume que los datos son normales o si desea evaluar si los datos cumplen con ciertas condiciones necesarias para utilizar ciertos métodos de análisis. En general, es importante recordar que la **normalidad multivariante** es solo una condición y no necesariamente un **requisito** para el **análisis** o el **aprendizaje automático**.



Se pueden apreciar un 17,64% de valores atípicos o outliers en la muestra. Sabiendo que el 20% de nuestra muestra son casos de ictus se podría incluso pensar que son casos positivos en su mayoría, debido a que por la agrupación de los datos se sabe que se distribuyen de manera parecida.

1.3.11 Contrastes de hipótesis.

Tabla 1.3: Prueba de Mardia

	Test	Statistic	p value	Result
1	Mardia Skewness	3233.65565064135	0	NO
2	Mardia Kurtosis	-2.09587027546379	0.036093700180801	NO
3	MVN			NO

Tabla 1.3: Prueba de Henze-Zirkler

	Test	HZ	p value	MVN
1	Henze-Zirkler	68.0180205611949	0	NO

Después de analizar también los estadísticos **no se puede asumir normalidad** multivariante, seguramente, debido a la falta de normalidad individual. Pero como se ha comentado antes, no supondrá un gran problema para el modelo de clasificación.

1.3.12 Análisis de varianza constante.

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  brain_multiv[, 1:3]
## Chi-Sq (approx.) = 1363.6, df = 6, p-value < 2.2e-16
```

Tampoco se puede afirmar que existe **homogeneidad en las varianzas**, lo que determinará son las elecciones de unos u otros modelos de clasificación, como por ejemplo un modelo discriminante cuadrático frente al lineal debido a la robustez del primero ante la falta de varianza constante (*véase anexo*).

2. MODELOS DE CLASIFICACIÓN.

En una primera etapa, se han estimado diferentes modelos de clasificación, especificando todas las variables explicativas propuestas. Todas las variables, han sido previamente utilizadas en diferentes procesos de limpieza, preparación y análisis exploratorio. Finalmente, tal y como se ha introducido con anterioridad, se ha decidido utilizar las variables no transformadas.

2.1 Estimaciones y selección del mejor modelo de clasificación.

A partir de los resultados, sobre todo tendrá un mayor peso lo referente a la **especificidad** (*Specifity*) ya que la clase negativa es más frecuente que la clase positiva, aunque también se tendrán en cuenta indicadores como la **precisión** (*Accuracy*) y medidores de la bondad del modelo, como el **indicador Kappa** y la **curva ROC**. Como el objetivo del estudio es elaborar modelos de clasificación para detectar cuántas personas sufren un infarto cerebral y la precisión indica el acierto general sabiendo que la mayoría de casos son negativos, no sería un indicador tan **representativo** ya que el valor real del modelo reside en predecir correctamente si un nuevo individuo puede sufrir un infarto cerebral, por ello, la especificidad sería una medida más importante para evaluar el rendimiento del modelo. Por último, para medir la **bondad** del modelo, el índice Kappa junto con la curva ROC aplicado a la tabla de confusión permite evaluar si la clasificación observada es similar (concordante) con la clasificación predecida por el clasificador.

Con respecto a los indicadores de bondad, cabe destacar, que el índice Kappa es una medida tiene en cuenta el **desequilibrio** de clases y mide el grado de acuerdo entre el modelo y una observación humana independiente. Con respecto a la curva ROC, es un indicador construido a partir de los valores del umbral de decisión del modelo, que es el valor a partir del cual se decide si un ejemplo pertenece a la clase positiva o a la clase negativa. Se elabora graficando el **True Positive Rate** (TPR) en el eje Y y el **False Positive Rate** (FPR) en el eje X para diferentes valores del umbral de decisión. El TPR es la proporción de ejemplos positivos que han sido correctamente clasificados por el modelo (es decir, la tasa de verdaderos positivos). El FPR es la proporción de ejemplos negativos que han sido incorrectamente clasificados como positivos (es decir, la tasa de falsos positivos). El modelo será mejor **cuanta más área** haya debajo de la curva.

Una vez que se han desarrollado todos los indicadores a tener en cuenta, en la tabla 2.1, se presentan los resultados de los distintos modelos estimados.

Tabla 2.1: Modelos de clasificación estimados

Indicadores	Modelo1	Modelo2	Modelo3	Modelo4	Modelo5
Accuracy	0,9121	0,9076	0,8399	0,8226	0,7589
Specificity	0,6648	0,6877	0,4384	0,5415	0,4143
Curva ROC	0,9443	0,9434	0,8505	0,9353	0,7979
Kappa	0,6962	0,6894	0,4265	0,4763	0,3192

Modelo 1: XGBoost.
Modelo 2: RandomForest.
Modelo 3: Regresión logística.
Modelo 4: Modelo de análisis discriminante cuadrático (QDA).
Modelo 5: Naive Bayes.

De los resultados del proceso de estimación mostrados en la tabla 2.1 se deduce que el mejor rendimiento, según el valor de la especificidad (*Specificity*) y la precisión es la correspondiente a los **dos primeros modelos**, por tanto, serán los elegidos como modelos de clasificación del estudio. (Modelo 1 y Modelo 2). En estos modelos, la bondad del ajuste, medida mediante el indicador Kappa y la curva ROC son con diferencia los valores más altos, rozando el 0,70 para el indicador Kappa y el 0,95 para el valor de la Curva Roc. El resto de modelos (Modelo 3, Modelo 4 y Modelo 5), en cambio, en todos los indicadores se observan valores muy por debajo de los dos primeros, salvo en el Modelo 3, que a excepción del indicador accuracy, todos los demás presentan valores más altos que su modelo predecesor (Modelo 2).

En cuanto al análisis estructural de los indicadores, cabe destacar principalmente los dos indicadores que han sido determinantes a la hora de seleccionar los dos modelos para realizar el estudio, el indicador *accuracy* que muestra el rendimiento en términos generales del modelo de clasificación, es decir, el valor del indicador nos aporta información en términos generales de como de preciso es el modelo a la hora de predecir si los individuos sufren o no un infarto cerebral. En cambio, la especificidad (*Specificity*) es el estadístico de referencia, ya que indica la tasa de verdaderos positivos del modelo.

2.2 Modelo 1. XGboost

El **XGboost** es uno de los algoritmos de machine learning (ML) que más se está utilizando gracias a que devuelve un grado de precisión bastante alto con poco esfuerzo que en muchos casos iguala o mejora modelos más complejos, sobre todo con datos heterogéneos. **XGboost** significa Extreme Gradient Boosting y está basado en el principio de *boosting*. Este consiste en generar múltiples modelos predictivos "débiles" de forma secuencial, de forma que cada modelo generado este alimentado de los resultados del anterior para crear un modelo robusto, el cual tiene mayor precisión en los resultados. En el proceso de entrenamiento cada modelo débil se intenta ajustar de forma iterativa hasta encontrar el **mínimo de la función objetivo**, si el modelo no es mejor que el anterior se vuelve al que tenía mejores resultados y se ajustan los pesos

para continuar con el proceso. El XGboost utiliza como modelos débiles arboles de decisión de diferente naturaleza en función de los objetivos marcados de regresión o de clasificación.

Se dividirá el conjunto de datos en dos subconjuntos (train/test) con la finalidad de entrenar el modelo y posteriormente evaluarlo.

Conversión en matriz

Para realizar modelos XGboost se requiere que los datos sean matrices, concretamente del tipo *DMatrix* (matriz de datos). Para ello vamos a convertir en matriz los datos exceptuando la variable objetivo, posteriormente aplicaremos una función concreta para el modelo de XGboost

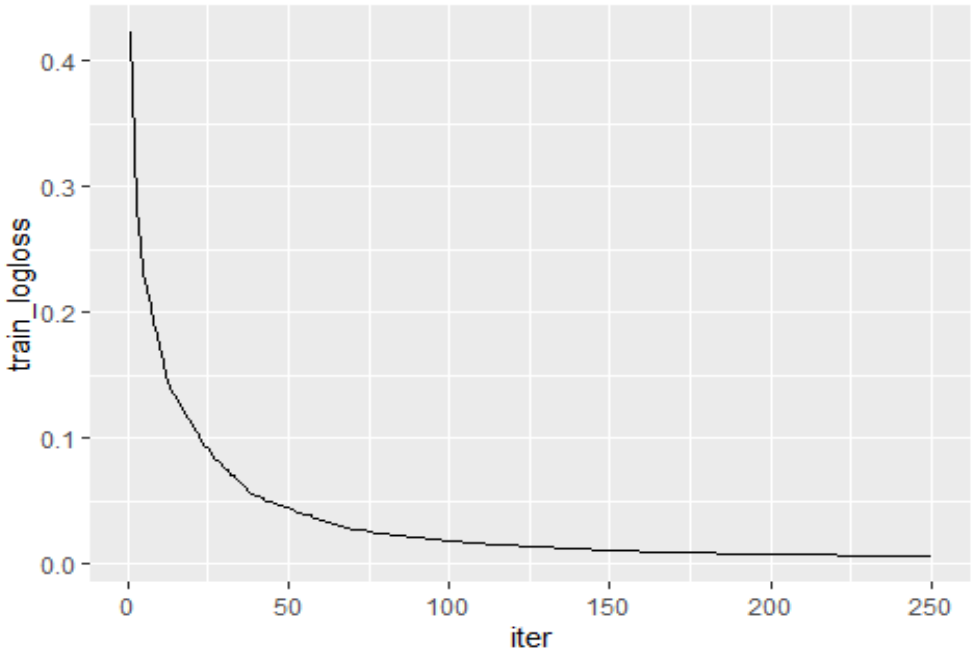
2.2.1 Método 1: Todas las variables

En primer lugar, se van a escoger todas las variables para ver cómo es capaz de predecir el modelo sin ningún tipo de restricción.

2.2.1.1 Entrenamiento.

Se ha escogido para el modelo de aprendizaje la regresión logística binaria para el resultado de salida, devolverá la probabilidad de Stroke entre 0 y 1 de que tenga derrame cerebral. Se ha fijado como 250 interacciones el máximo para entrenar el modelo, se pretende que se ejecute de forma sencilla y no haya *overfitting*. Se ha fijado el número de bifurcaciones a las que están hace por defecto (6), y la tasa de aprendizaje en 0.6 ya que se desea que se llegue un valor más ajustado en el resultado de la función objetivo rápido. Los núcleos usados se han fijado en 4 para los procesos de cálculo.

Gráfico 2.1: Curva de aprendizaje



Para evaluar el modelo se hará mediante el índice **LogLoss**, este mide como de lejos está cada predicción con respecto de la etiqueta real. Los clasificadores más idóneos tienen valores progresivamente más pequeños, por tanto, un menor LogLoss tendrá una mayor precisión.

En el primer modelo se observa que sigue habiendo una mejora, aunque mucho menor que al principio, en las 100 interacciones que ha realizado. Cuando se evalúe el modelo con el conjunto de entrenamiento se observará como de bueno es. Además, se puede considerar que se está ajustando correctamente ya que no hay anomalías en la curva que hace LogLoss.

Importancia de las variables:

Tabla 2.2: Importancia de las variables.

	Feature	Gain	Cover	Frequency
1	age	0.44916832734971	0.344018530044986	0.291335001137139
2	avg_glucose_level	0.21883712164062	0.268448675452313	0.317944052763248
3	bmi	0.146833234923311	0.249443244389735	0.242665453718444
4	smoking_status	0.0537146825216098	0.0362438376425417	0.050943825335456
5	work_type	0.0525460958912068	0.0240477186836568	0.0247896292926996
6	gender	0.0217817347169413	0.0233187318366291	0.0229702069592904
7	residence_type	0.0202819917939263	0.0184856794332892	0.0250170570843757
8	ever_married	0.0198513668085899	0.0161292216237717	0.00932453945872186
9	hypertension	0.00883181537932512	0.00989280567270523	0.00864225608369343
10	heart_disease	0.00815362897475989	0.00997155522037295	0.006367978166932

2.2.1.2 Predicciones.

Se muestran los 5 primeros individuos de la base de datos *test* con el score asignado, posteriormente, se va a realizar una evaluación general del modelo con todos los datos obtenidos. Este procedimiento **se aplicará en todos los modelos realizados.**

```
## [1] 0.185917333 0.002780621 0.064226173 0.613226116 0.003555191
```

2.2.1.3 Evaluación del modelo

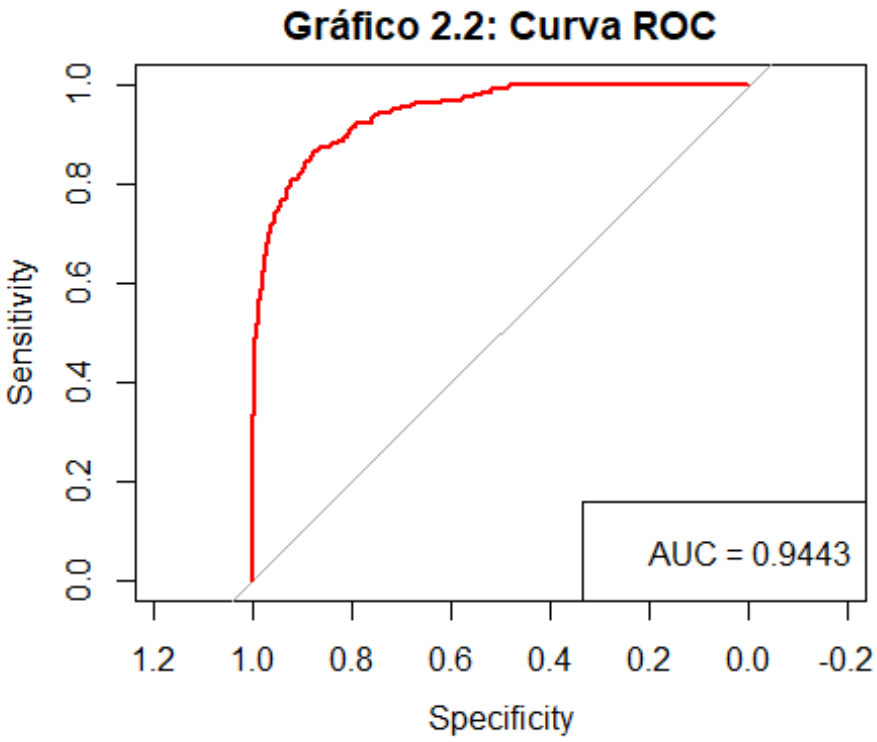
```
## Confusion Matrix and Statistics
##
##    X2
## X1     0    1
##   0 1387  117
##   1   39  232
##
##                Accuracy : 0.9121
##                  95% CI : (0.898, 0.9249)
##     No Information Rate : 0.8034
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6962
##
##  Mcnemar's Test P-Value : 7.051e-10
##
##             Sensitivity : 0.9727
##             Specificity : 0.6648
##          Pos Pred Value : 0.9222
##          Neg Pred Value : 0.8561
##              Prevalence : 0.8034
##          Detection Rate : 0.7814
##    Detection Prevalence : 0.8473
##       Balanced Accuracy : 0.8187
##
##        'Positive' Class : 0
##
```

Se va a situar el score en un 70%, la razón es que sí el procedimiento en una situación así consistiera en la monitorización y suministrar fármacos que no tuvieran efectos secundarios a largo plazo, es más importante acertar en los pacientes que realmente le está dando el infarto. En otro tipo de enfermades se tendría que ajustar los criterios, por ejemplo, en un cáncer, otro ejemplo sería la detección de embarazos. En estos supuestos habría que analizarlos y fijar el score.

El modelo ha obtenido un **91,21**% de precisión cuando se ha evaluado con el conjunto de test, esto quiere decir que de los 1754 pacientes que había, 1387 ha tenido un diagnóstico correcto. Puede parecer buen modelo viendo la precisión, pero si observamos los pacientes que les dio derrame cerebral conseguimos identificar el **66,48**% de los casos. Los pacientes que se podría suponer que se han quedado hospitalizados sin necesidad suponen el 2,73%.

En cuanto al Kappa como medida de bondad se puede ver un valor de casi **70**% que indica un modelo bueno sin llegar a ser excelente.

Curva ROC



El valor de la curva ROC supera el **94**%, lo cual arroja una muy buena bondad. Sabiendo que una curva ROC óptima tiende a seguir la diagonal de la línea de unidad (es decir, una línea que pasa a través del punto (0,0) y (1,1) en el gráfico), lo que indica una alta TPR y una baja FPR. Esto significa que el modelo tiene una alta capacidad para identificar correctamente los casos positivos y minimizar los casos falsos positivos.

2.2.2 Método 2: Modelo refinado.

Se cogerán las variables cuya importancia en el primer modelo cuentan con un **mayor peso** (age, avg_glucose_level, bmi), y se separara el conjunto de datos en 70/30 para realizar este nuevo modelo. Se emplea la misma semilla para que puedan ser comparables.

2.2.2.1 Entrenamiento.

Se procede a entrenar los datos para las variables seleccionadas.

Importancia de las variables.

```
## xgb.DMatrix  dim: 4142 x 3  info: label  colnames: yes

## xgb.DMatrix  dim: 1775 x 3  info: label  colnames: yes
```

Gráfico 2.3: Curva de aprendizaje

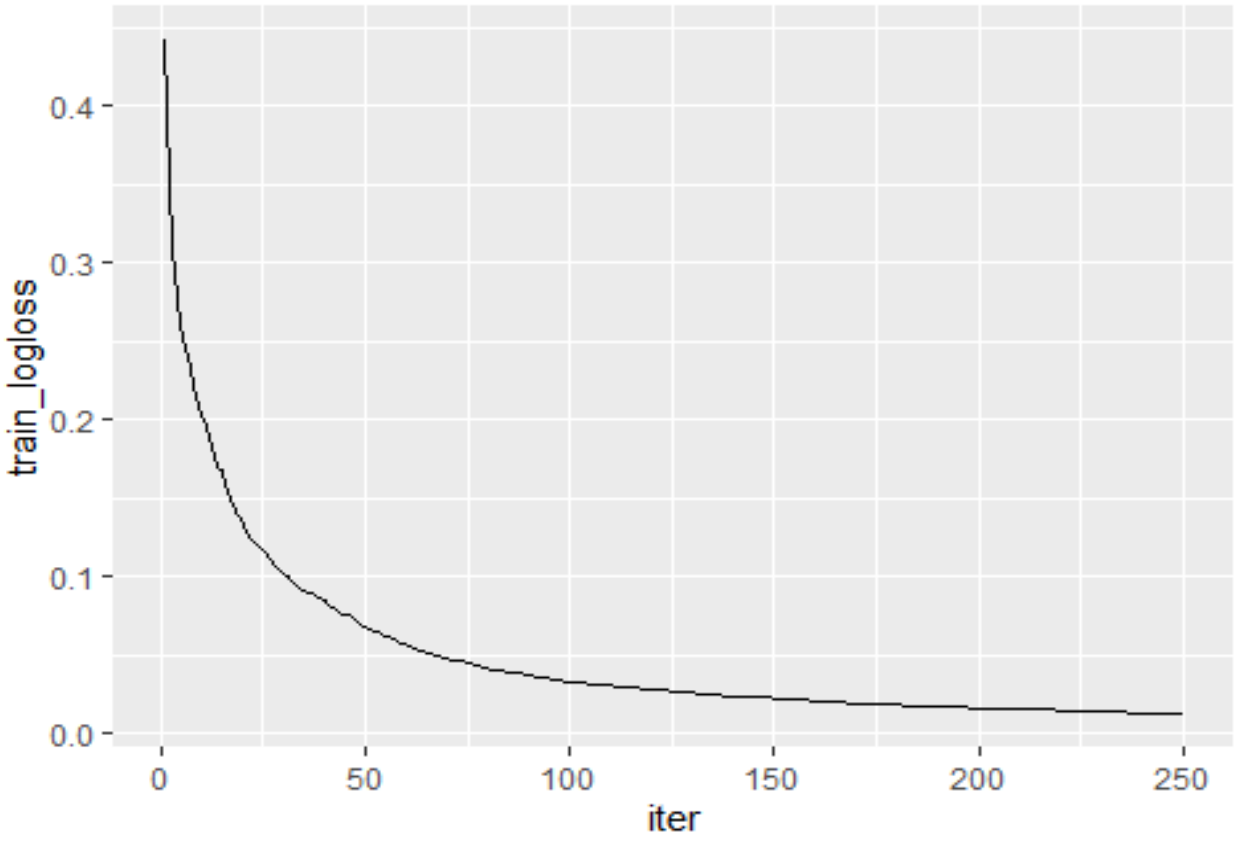


Tabla 2.3: Importancia de las variables

	Feature	Gain	Cover	Frequency
1	age	0.62212963051173	0.455871161326874	0.315430131998179
2	avg_glucose_level	0.219850991049821	0.293876406925147	0.374146563495676
3	bmi	0.158019378438449	0.250252431747979	0.310423304506145

2.2.2.2 Predicción

```
## [1] 0.98460358 0.01488991 0.00439485 0.20666440 0.04480074
```

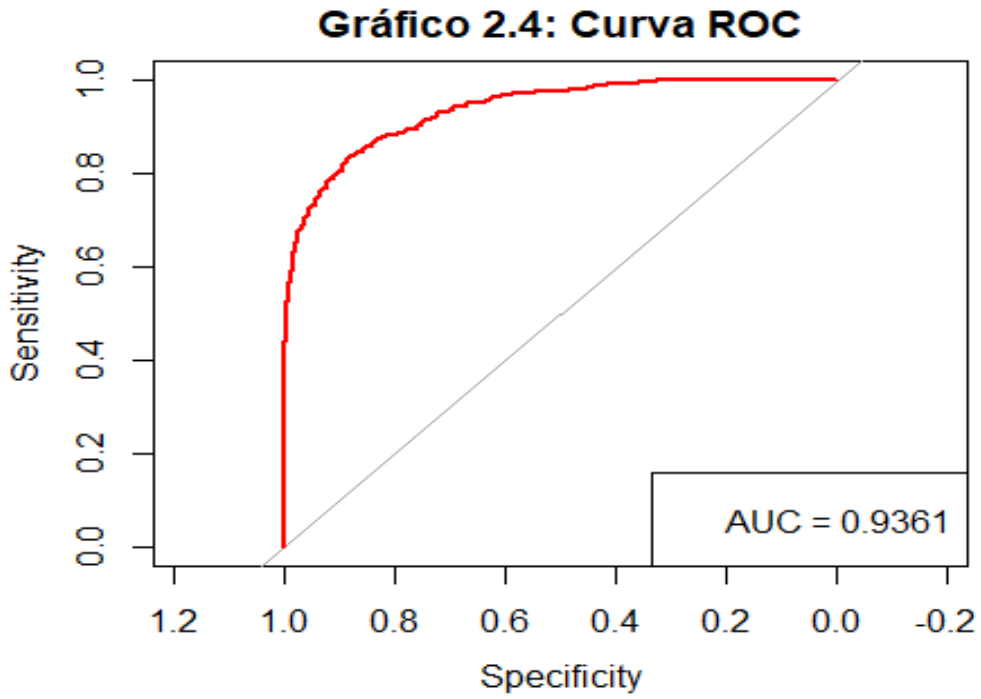
2.2.2.3 Evaluación del modelo

```
## Confusion Matrix and Statistics
##
##    X2
## X1     0    1
##   0 1396  126
##   1   30  223
```

```
##
##                Accuracy : 0.9121
##                  95% CI : (0.898, 0.9249)
##     No Information Rate : 0.8034
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6896
##
##  Mcnemar's Test P-Value : 2.825e-14
##
##             Sensitivity : 0.9790
##             Specificity : 0.6390
##          Pos Pred Value : 0.9172
##          Neg Pred Value : 0.8814
##              Prevalence : 0.8034
##          Detection Rate : 0.7865
##    Detection Prevalence : 0.8575
##       Balanced Accuracy : 0.8090
##
##        'Positive' Class : 0
```

En este caso se puede apreciar que, pese a que la precisión sigue igual (0,9121) la sensibilidad y especificidad han variado, habiendo **decrecido** la segunda. El kappa también ha disminuido un poco (menos de un punto), pero en general el modelo es casi idéntico, por lo que se podría decir que es una buena simplificación del modelo.

Curva ROC.



La bondad calculada por la curva ROC también es algo menor, aunque también es casi idéntico. Pese a que se podría decir que es una buena simplificación del modelo con todas las variables, este estudio concreto trata de salvar vidas a individuos gracias a diagnosticar bien un caso de infarto cerebral y casi todas las variables son **fácilmente recolectables** en un entorno sanitario (como el IMC, la existencia de hipertensión, enfermedad cardíaca o el género) son datos que residen en la historia clínica del paciente. Por tanto, el modelo que se considera válido será el primero planteado que de 100 pacientes con ictus diagnostica correctamente 66.

2.3 Modelo 2. Random Forest.

Random Forest es un método de aprendizaje automático utilizado en tareas de clasificación y regresión. Se basa en la idea de crear un conjunto de *árboles de decisión*, cada uno de los cuales es entrenado con un subconjunto aleatorio de las características y ejemplos de entrenamiento. Los árboles de decisión son modelos de clasificación que toman un conjunto de características como entrada y devuelven una predicción de clase para cada ejemplo.

Este método combina las predicciones de todos los árboles de decisión para hacer predicciones más precisas. La idea es que, al combinar muchos modelos diferentes entrenados con diferentes subconjuntos de datos, el resultado final será **más preciso y robusto** que si solo se utilizara un árbol de decisión.

Random Forest es un método muy popular en el aprendizaje automático debido a su capacidad para manejar un gran número de características y para generalizar bien a nuevos datos.

2.3.1 Método 1: Todas las variables.

Primero, se va a optar a ver cómo clasifica el modelo usando todas las variables del estudio.

2.3.1.1 Entrenamiento.

Para el random forest, en vez de usar los hiperparámetros predeterminados, se realiza un **bucle** para elegir los **hiperparámetros** que minimizan el error. El bucle esta acotado entre los hiperparámetros óptimos. En el anexo está el bucle con el rango completo.

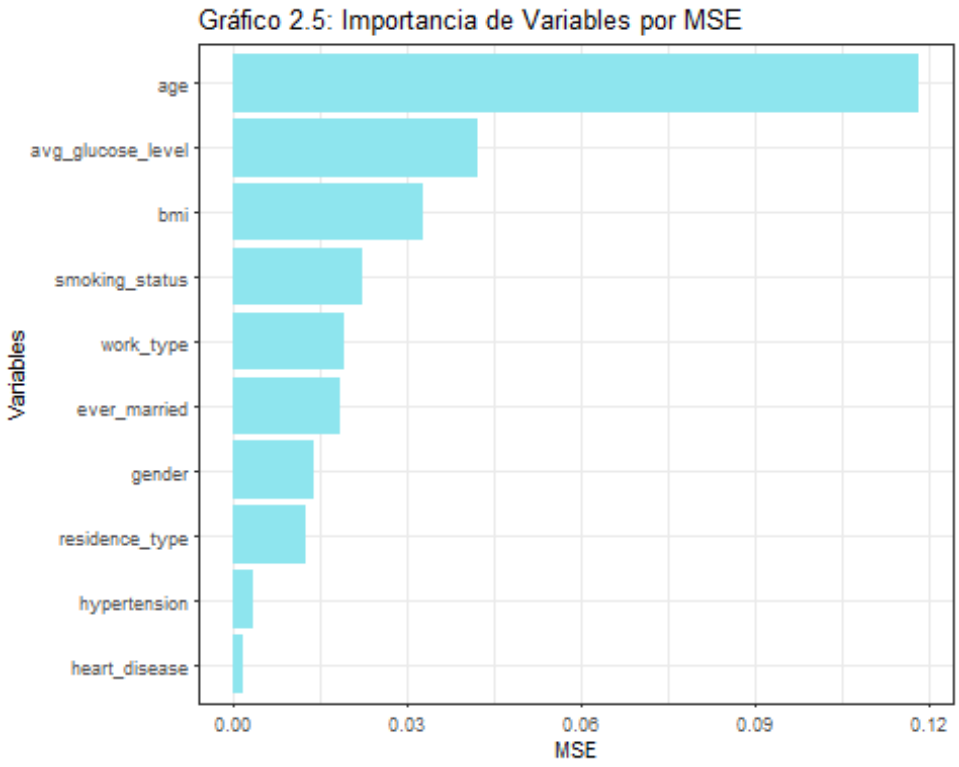
Los hiperparámetros que vamos a modificar son *mtry*, que indica el número de variables cogidas aleatoriamente en cada división y *nodesize*, que recoge cuantas observaciones vamos a tener en los nodos terminales.

Hiperparámetros que minimizan el error:

```
## [1] 4

## [1] 2
```

Incluimos los parámetros y la **importancia de las variables** en el modelo, ya que nos brinda una mejor interpretabilidad.

```
##
## Call:
##  randomForest(formula = stroke ~ ., data = train_r, mtry = 4,      nod
esize = 2, ntree = 2500, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 2500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 7.99%
## Confusion matrix:
##       No Yes class.error
## No  3196 111  0.03356516
## Yes  220 615  0.26347305
```



La gráfica nos indica el incremento del error medio cuadrático, en caso de que los datos de la variable cambien, es decir, la importancia de la variable en el modelo.

La variable más importante es *age*. Esta destaca por encima de las demás. La sigue *avg_glucose_level* y *bmi*. Las demás variables tienen una importancia similar y hay dos que tienen una importancia bastante menor, que son *hypertension* y *heart_disease*.

2.3.1.2 Predicción.

En RandomForest se muestra si la predicción va a ser cierta o falsa clasificándolas entre las categorías *Yes* y *No*.

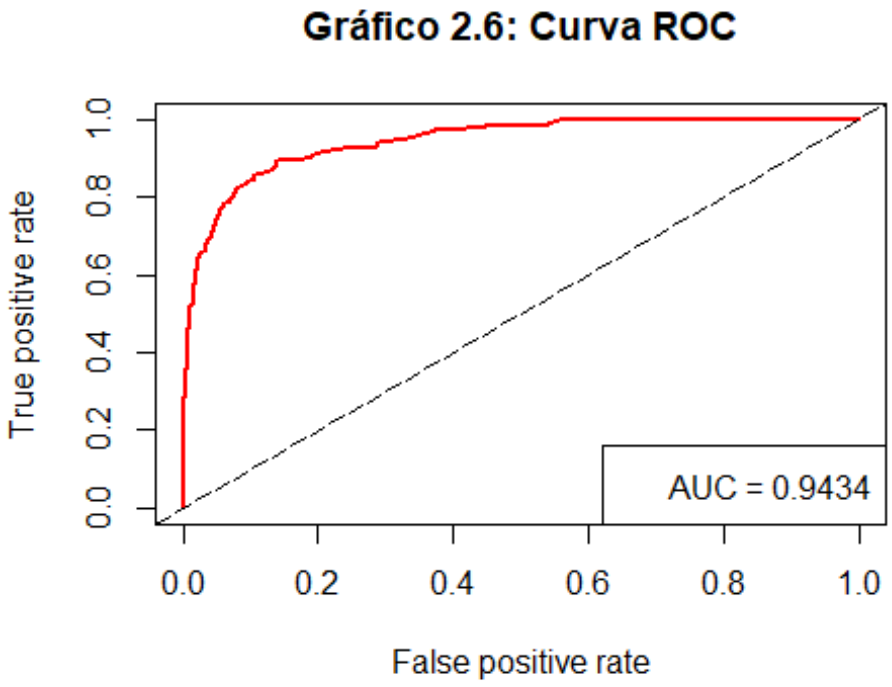
```
##   1   2   3   4   5
##  No  No Yes  No  No
## Levels: No Yes
```

2.3.1.3 Evaluación.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1371  109
##        Yes   55  240
##
##                Accuracy : 0.9076
##                  95% CI : (0.8932, 0.9207)
##     No Information Rate : 0.8034
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6894
##
##  Mcnemar's Test P-Value : 3.494e-05
##
##             Sensitivity : 0.9614
##             Specificity : 0.6877
##          Pos Pred Value : 0.9264
##          Neg Pred Value : 0.8136
##              Prevalence : 0.8034
##          Detection Rate : 0.7724
##    Detection Prevalence : 0.8338
##       Balanced Accuracy : 0.8246
##
##        'Positive' Class : No
##
```

Como se puede apreciar el modelo tiene una buena precisión general del **90,76**%. Siendo muy bueno en detectar casos negativos (*sensibilidad*) y bueno, aunque mejorable al detectar los casos objeto de estudio, los positivos (especificidad). Pero en líneas generales el modelo acertará entre un **89**% y un **92**% de las veces. Es importante resaltar que el principal riesgo a tener en cuenta es el de los individuos que clasifica en el grupo de no y realmente pertenecen, al contrario, ya que suman 109. En cuanto al Kappa, se observa un valor de **0,68**, lo cual conduce a pensar que la tasa de buena clasificación es buena.

Curva ROC



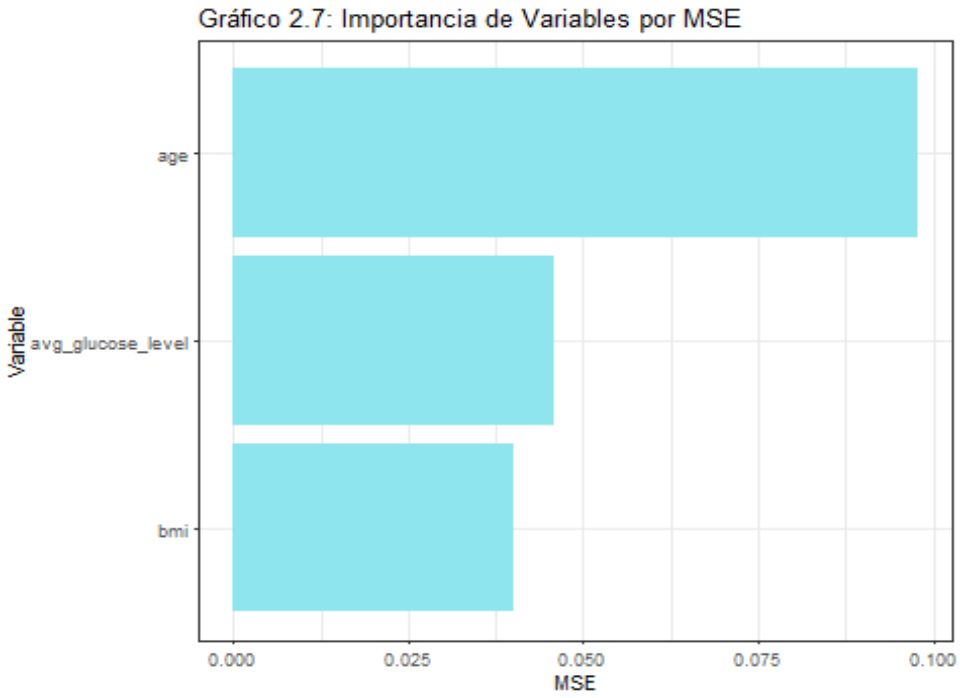
Se observa que el área debajo de la curva es **amplia** lo que reafirma que el modelo es bastante sólido y bueno prediciendo.

2.3.2 Método 2: Modelo refinado

Se va a intentar simplificar el modelo utilizando las variables más significativas por su influencia en el error medio cuadrático. En el gráfico anterior se pudo observar que las variables más importantes eran *age, bmi y avg_glucose_level*. Se va a proceder a hacer el modelo con esas tres variables debido a que se puede emplear también como un **modelo selector de variables**. Simplificar un modelo es importante no solo porque incurre en una mejor interpretabilidad sino porque también es más fácil recolectar los datos objeto del estudio.

2.3.2.1 Entrenamiento

Se escoge el nuevo data.frame con las 3 variables más importantes según el método.



Se observa que la importancia de las variables en el error medio cuadrático se mantienen similares.

2.3.2.2 Predicción.

```
##   1   2   3   4   5
## Yes  No  No Yes  No
## Levels: No Yes
```

2.3.2.3 Evaluación.

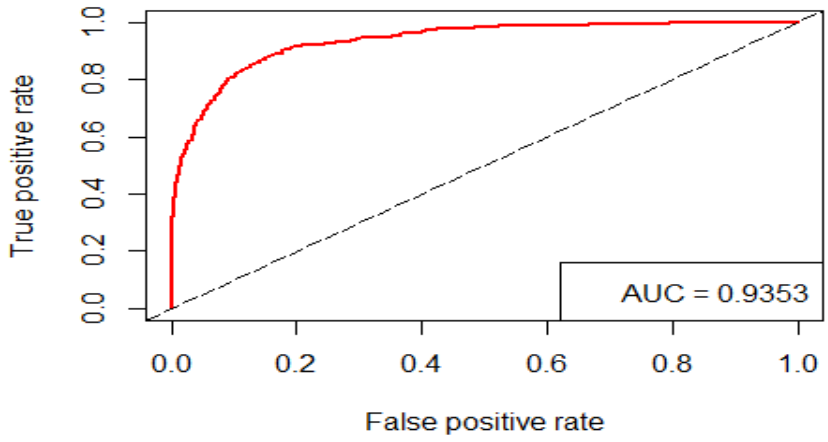
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1369  125
##        Yes   57  224
##
##                Accuracy : 0.8975
##                  95% CI : (0.8824, 0.9112)
##     No Information Rate : 0.8034
##     P-Value [Acc > NIR] : < 2.2e-16
##

##
##                   Kappa : 0.6497
```

```
##
##  Mcnemar's Test P-Value : 6.822e-07
##
##             Sensitivity : 0.9600
##             Specificity : 0.6418
##          Pos Pred Value : 0.9163
##          Neg Pred Value : 0.7972
##              Prevalence : 0.8034
##          Detection Rate : 0.7713
##    Detection Prevalence : 0.8417
##       Balanced Accuracy : 0.8009
##
##        'Positive' Class : No
```

Se ha simplificado mucho el modelo sin perder apenas un **1**% de **precisión** para el conjunto test. Aunque igual que el modelo anterior, es algo más débil en la especificidad, acertando el **64,18**% de los casos. El valor de kappa en esta ocasión sigue reflejando que es un modelo **moderadamente bueno**, aunque también ha bajado un poco y los falsos negativos han aumentado en unos 20 individuos. Todo esto es importante tenerlo en cuenta a la hora de ver si simplificar o no el modelo.

Curva ROC

Gráfico 2.8: Curva ROC



El área bajo la curva es **prácticamente idéntica** a a la anterior, por lo que se podría asumir que es una buena simplificación del modelo.

3. COMPARACIÓN Y CONCLUSIÓN DE LOS MODELOS.

Antes de comenzar, se visualizará una tabla comparando los estadísticos de los dos modelos seleccionados. Ambos son con todas las variables en primer lugar y una variante con las variables más importantes (*age*, *bmi* y *avg_glucose_level*).

Tabla 3.1: Modelos de clasificación estimados

Indicadores	Modelo1	Modelo2	Modelo1.1	Modelo2.1
Accuracy	0,9121	0,9076	0,9121	0,8975
Specificity	0,6648	0,6877	0,6390	0,6418
Curva ROC	0,9443	0,9434	0,9361	0,9353
Kappa	0,6962	0,6894	0,6896	0,6497
Sensitivity	0,9727	0,9614	0,9790	0,9600

Modelo 1: XGBoost usando todas las variables.
Modelo 2: RandomForest usando todas las variables.
Modelo 1.1: XGBoost según el MSE.
Modelo 2.1: Random Forest según el MSE.

En primer lugar, se han realizado los modelos de **XGBoost** y **RandomForest** con todas las variables, debido a que son **modelos robustos** y toleran bien aspectos como la no transformación de las variables. Posteriormente, se ha realizado una variante de estos modelos seleccionando las variables según su **error cuadrático medio** (MSE).

Ambos modelos presentan una tasa de **acierto general** alta (*accuracy*) rondando el 91% y una **especificidad** (*specificity*) del 66 y 68% respectivamente. Según los **estadísticos de bondad** (*Curva ROC y Kappa*), ambos modelos son, por lo general, estimadores buenos. En los modelos refinados los resultados obtenidos rondan el 90% de precisión general y una reducción en torno a 2,5 y 4 puntos respectivamente de la especificidad. Por último, los estadísticos de bondad se mantienen en rangos considerados como aceptables en el índice Kappa y óptimos en la curva ROC.

Desde una visión de negocio, el desarrollo de estos modelos puede tener **varios objetivos**, desde la **minimización** de costes hasta la **optimización** de algún resultado concreto. En este estudio, el objetivo principal se enfoca en tener una tasa de acierto elevada de los pacientes que estén sufriendo un derrame cerebral, sin importar tanto una variación extrema en los costes de desarrollo del modelo. Por ello, el modelo que se considerará como el más idóneo es el elaborado mediante el método de **RandomForest** con todas las variables, ya que se pueden predecir de forma correcta 69 casos de ictus por cada 100.

Si el enfoque fuese orientado hacia la reducción de costes, se tendrían en cuenta aspectos como la tasa de aciertos general o la reducción de **falsos positivos** (medido como 1-*sensitivity*), buscando el modelo que mejor puntúe en este estadístico. En este caso, la elección del mejor modelo pasaría a ser el **modelo refinado de XGBoost**, ya que solo deja a un 2% de los pacientes hospitalizados que no deberían estar hospitalizados, manteniendo el resto de estadísticos en rangos adecuados.

El análisis se ha realizado con datos ya recogidos y devolviendo unos resultados bastante buenos. El modelo seleccionado, tendría que **integrarse** dentro de un organismo (sistema sanitario) que fuese introduciendo nuevos datos priorizando el criterio médico y re-entrenándolo pasado un período de tiempo de prueba para ver si generaliza bien en un ambiente de total incertidumbre. Esto es **esencial** en todo modelo estadístico, debido a que los factores pueden variar a lo largo del tiempo, en especial cuando se trata de características atribuibles al ser humano. Con esto se pretende la **no obsolescencia** del modelo y la **máxima generalización** posible.

ANEXO:

Modelo Regresión Logística

En estadística, la **regresión logística** es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica en función de las variables independientes o predictoras. Es útil para modelar la probabilidad de un evento ocurriendo en función de otros factores. Las **condicones** que se deben de dar para obtener la mayor precisión en el análisis con una regresión lineal son:

• Respuesta binaria: La variable dependiente ha de ser binaria. • Independencia: las observaciones han de ser independientes.

• Multicolinealidad: se requiere de muy poca a ninguna multicolinealidad entre los predictores (para regresión logística múltiple).

• Linealidad entre la variable independiente y el logaritmo natural de odds.

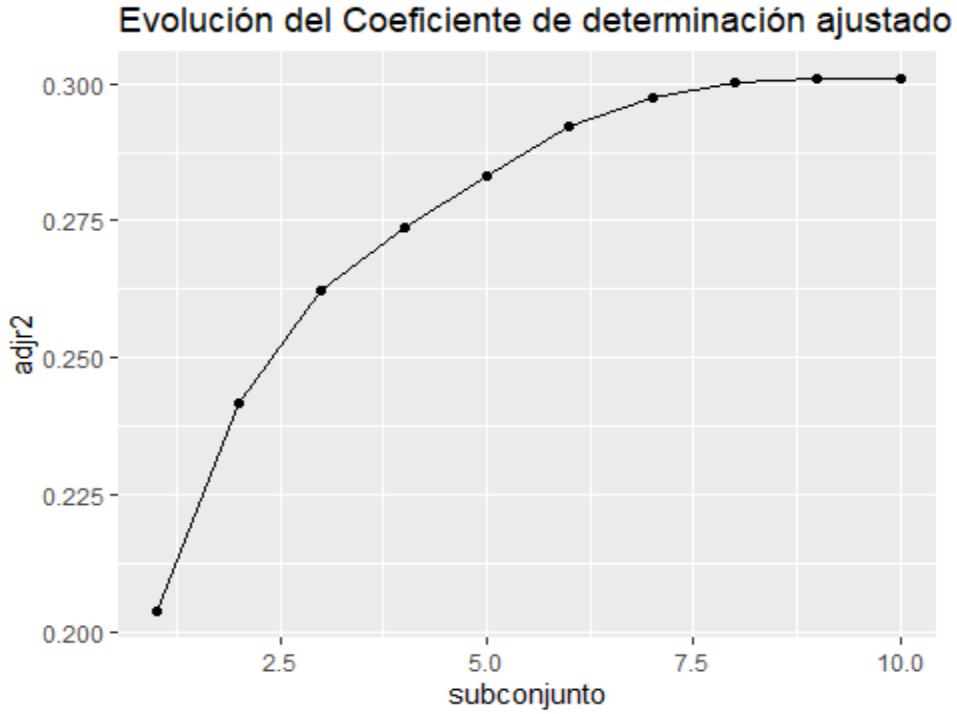
Método 1: Best Subset.

El proceso de **best subset selection** consiste en evaluar todos los posibles modelos que se pueden crear por combinación de los predictores disponibles. El algoritmo a seguir para k predictores es: Se genera lo que se conoce como modelo nulo (M0), que es el modelo sin ningún predictor. El **coeficiente de determinación ajustado** es la medida que define el porcentaje explicado por la varianza de la regresión de acuerdo con la varianza experimentada por las variables aplicadas. Este penaliza la inclusión de aquellas variables que no resultan trascendentales para la variable real.

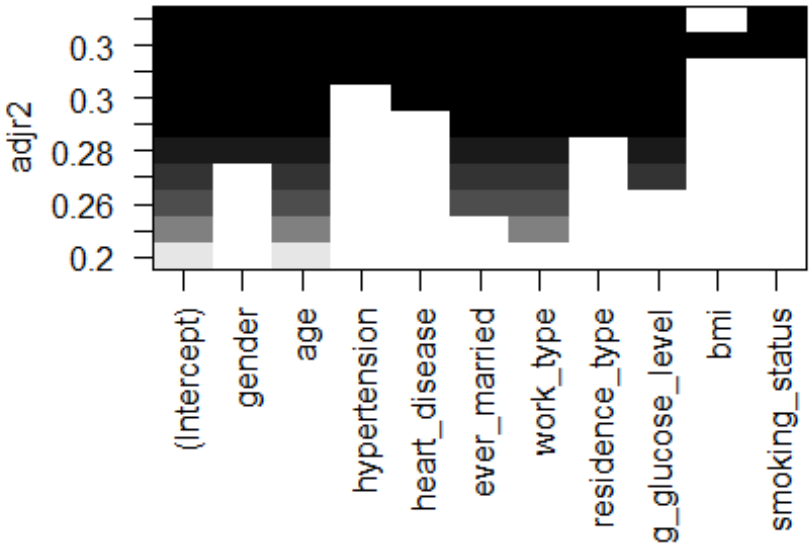
División de los datos en dos subconjuntos (train/test) con la finalidad de entrenar el modelo y posteriormente evaluarlo.

```
## Subset selection object
## Call: regsubsets.formula(stroke ~ ., data = rlog_brain$train, method =
"exhaustive",
##     nvmax = 10)
## 10 Variables  (and intercept)
##                    Forced in Forced out
## gender                 FALSE      FALSE
## age                    FALSE      FALSE
## hypertension           FALSE      FALSE
## heart_disease          FALSE      FALSE
## ever_married           FALSE      FALSE
## work_type              FALSE      FALSE
## residence_type         FALSE      FALSE
## avg_glucose_level      FALSE      FALSE
## bmi                    FALSE      FALSE
## smoking_status         FALSE      FALSE
```

```
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive



##
##           gender age hypertension heart_disease ever_married work_type
## 1  ( 1 )  " "    "*" " "          " "           " "          " "
## 2  ( 1 )  " "    "*" " "          " "           " "          "*"
## 3  ( 1 )  " "    "*" " "          " "           "*"          "*"
## 4  ( 1 )  " "    "*" " "          " "           "*"          "*"
## 5  ( 1 )  "*"    "*" " "          " "           "*"          "*"
## 6  ( 1 )  "*"    "*" " "          " "           "*"          "*"
## 7  ( 1 )  "*"    "*" " "          "*"           "*"          "*"
## 8  ( 1 )  "*"    "*" "*"          "*"           "*"          "*"
## 9  ( 1 )  "*"    "*" "*"          "*"           "*"          "*"
## 10  ( 1 ) "*"    "*" "*"          "*"           "*"          "*"
##           residence_type avg_glucose_level bmi smoking_status
## 1  ( 1 )  " "            " "               " " " "
## 2  ( 1 )  " "            " "               " " " "
## 3  ( 1 )  " "            " "               " " " "
## 4  ( 1 )  " "            "*"               " " " "
## 5  ( 1 )  " "            "*"               " " " "
## 6  ( 1 )  "*"            "*"               " " " "
## 7  ( 1 )  "*"            "*"               " " " "
## 8  ( 1 )  "*"            "*"               " " " "
## 9  ( 1 )  "*"            "*"               " " "*"
## 10  ( 1 ) "*"            "*"               "*" "*"
```



En este caso, la mejor elección se sitúa entre el modelo de 7 u 8 variables, para realizar el modelo se seleccionarán, por tanto, ocho variables que es el que más explicativo.



Como se puede observar en el gráfico anterior, se excluirán las variables *bmi* u *smoking_status*. Además, debido a la condición de multicolinealidad, se ha omitido la variable *work_type* ya que presentaba un alto grado de correlación con la variable *age*.

```
## # A tibble: 2 × 3
##   stroke     n  prop
##   <fct>  <int> <dbl>
## 1 No      4733   0.8
## 2 Yes     1184   0.2
```

División de los datos en dos subconjuntos (train/test) con la finalidad de entrenar el modelo y posteriormente evaluarlo.

Entrenamiento

```
##
## Call:
## glm(formula = stroke ~ gender + age + hypertension + heart_disease +
##     ever_married + avg_glucose_level + residence_type, family = binomi
al,
##     data = rlog_brain$train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.2955  -0.5478  -0.2603  -0.1002   3.3886
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -2.1311497  0.3790637  -5.622 1.89e-08 ***
## gender            -0.6459488  0.1030828  -6.266 3.70e-10 ***
## age                0.0832722  0.0033913  24.555  < 2e-16 ***
## hypertension      -0.5768735  0.1432870  -4.026 5.67e-05 ***
## heart_disease     -0.8383017  0.1816618  -4.615 3.94e-06 ***
## ever_married      -0.8259895  0.1327242  -6.223 4.87e-10 ***
## avg_glucose_level  0.0068083  0.0008439   8.067 7.18e-16 ***
## residence_type    -0.6543629  0.0957365  -6.835 8.20e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4163.6  on 4141  degrees of freedom
## Residual deviance: 2867.6  on 4134  degrees of freedom
## AIC: 2883.6
##
## Number of Fisher Scoring iterations: 6
```

Como se puede observar, las variables más representativas de este modelo son *age*, *avg_glucose_level* y *hypertension*.

Obtención de los coeficientes:

```
##       (Intercept)            gender               age      hypertensio
n
##      -2.131149651      -0.645948834       0.083272194      -0.57687347
5
##     heart_disease      ever_married avg_glucose_level    residence_typ
e
##      -0.838301719      -0.825989453       0.006808331      -0.65436294
8

##                      Estimate   Std. Error   z value      Pr(>|z|)
## (Intercept)       -2.131149651 0.3790636649 -5.622142  1.886044e-08
## gender            -0.645948834 0.1030828024 -6.266310  3.697031e-10
## age                0.083272194 0.0033912963 24.554680 3.854788e-133
## hypertension      -0.576873475 0.1432869869 -4.026000  5.673358e-05
## heart_disease     -0.838301719 0.1816618007 -4.614628  3.937991e-06
## ever_married      -0.825989453 0.1327241578 -6.223354  4.866369e-10
## avg_glucose_level  0.006808331 0.0008439329  8.067385  7.182001e-16
## residence_type    -0.654362948 0.0957364954 -6.835042  8.198112e-12

##       (Intercept)            gender               age      hypertensio
n
##      1.886044e-08      3.697031e-10     3.854788e-133      5.673358e-0
```

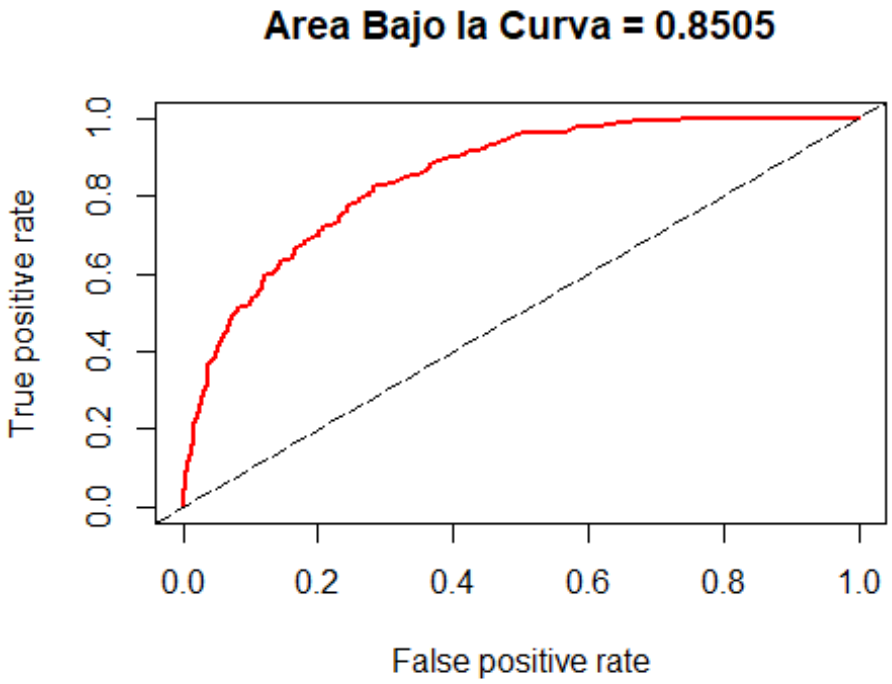
```
5
##     heart_disease      ever_married avg_glucose_level    residence_typ
e
##      3.937991e-06      4.866369e-10      7.182001e-16      8.198112e-1
2
```

Predicciones

```
##
## modelo1_glm.pred   No  Yes
##              No  1337  196
##              Yes   88  153

## [1] 0.8399098

## [1] 0.1600902
```

Evaluación

```
## Confusion Matrix and Statistics
##
##
## modelo1_glm.pred   No  Yes
##              No  1337  196
##              Yes   88  153
##
##                Accuracy : 0.8399
##                  95% CI : (0.822, 0.8567)
##     No Information Rate : 0.8033
##     P-Value [Acc > NIR] : 4.002e-05
##
##                   Kappa : 0.4265
##
##  Mcnemar's Test P-Value : 2.163e-10
##
##             Sensitivity : 0.9382
##             Specificity : 0.4384
##          Pos Pred Value : 0.8721
##          Neg Pred Value : 0.6349
##              Prevalence : 0.8033
##          Detection Rate : 0.7537
##    Detection Prevalence : 0.8641
##       Balanced Accuracy : 0.6883
##
##        'Positive' Class : No
##
```



En cuanto, al *accuracy* del modelo, este ha obtenido un 82,37% de precisión cuando se ha evaluado con el conjunto de test. Pero la especificidad de este modelo tan solo es del 36,39%, por tanto, este modelo que tiene una precisión medianamente elevada no es un buen modelo debido a la especificidad que presenta.

Método 2: Refinado

Este modelo será construido tan solo con las variables que, como se ha expuesto en el apartado anterior, tienen una mayor importancia en el modelo. Siguiendo el mismo procedimiento que en el anterior y las mismas condiciones, para después poder realizar una comparación verídica de ambos modelos.

Entrenamiento.

```
##
## Call:
## glm(formula = stroke ~ age + hypertension + avg_glucose_level,
##     family = binomial, data = rlog_brain$train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7642  -0.6011  -0.2693  -0.0922   3.4349
##


## Coefficients:
```

```
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -5.7638563  0.2488883 -23.158  < 2e-16 ***
## age                0.0772384  0.0032717  23.608  < 2e-16 ***
## hypertension      -0.6135807  0.1380070  -4.446 8.75e-06 ***
## avg_glucose_level  0.0053851  0.0007976   6.752 1.46e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4163.6  on 4141  degrees of freedom
## Residual deviance: 3029.3  on 4138  degrees of freedom
## AIC: 3037.3
##
## Number of Fisher Scoring iterations: 6
```

Obtención de los coeficientes:

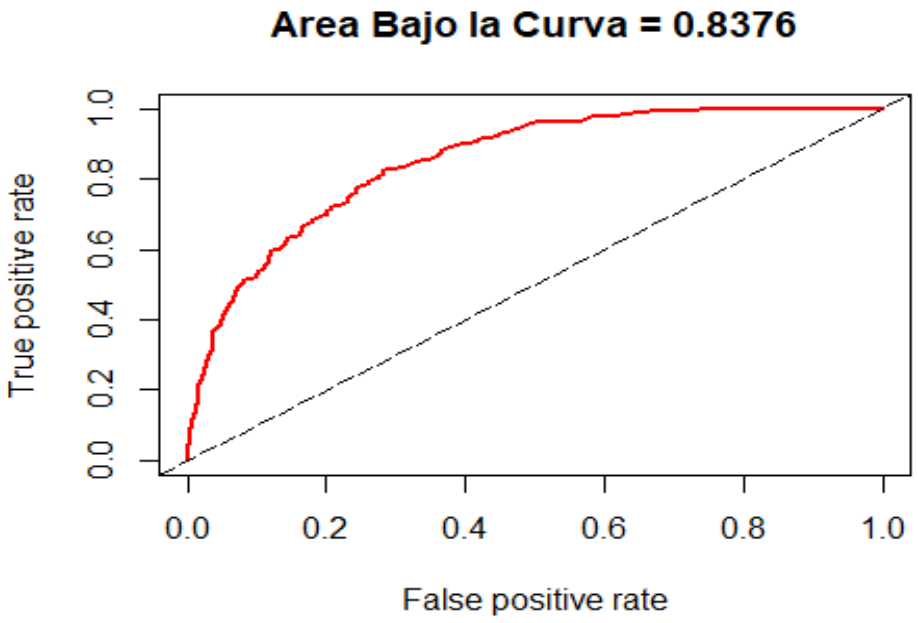
```
##       (Intercept)               age      hypertension avg_glucose_leve
l
##      -5.763856336       0.077238391      -0.613580653       0.00538510
3

##                       Estimate   Std. Error    z value      Pr(>|z|)
## (Intercept)       -5.763856336 0.2488883342 -23.158403 1.196286e-118
## age                0.077238391 0.0032716983  23.608042 3.186672e-123
## hypertension      -0.613580653 0.1380070170  -4.446011  8.747966e-06
## avg_glucose_level  0.005385103 0.0007975996   6.751636  1.461870e-11

##       (Intercept)               age      hypertension avg_glucose_leve
l
##     1.196286e-118     3.186672e-123      8.747966e-06      1.461870e-1
1
```

Predicciones.

```
##
## modelo2_glm.pred   No  Yes
##              No  1325  204
##              Yes  100  145

## [1] 0.8286359

## [1] 0.1713641
```

Evaluación.

```
## Confusion Matrix and Statistics
##
##
## modelo2_glm.pred   No  Yes
##              No  1325  204
##              Yes  100  145
##
##                Accuracy : 0.8286
##                  95% CI : (0.8103, 0.8459)
##     No Information Rate : 0.8033
##     P-Value [Acc > NIR] : 0.003499
##
##                   Kappa : 0.3891
##
##  Mcnemar's Test P-Value : 3.474e-09
##
##             Sensitivity : 0.9298
##             Specificity : 0.4155
##          Pos Pred Value : 0.8666
##          Neg Pred Value : 0.5918
##              Prevalence : 0.8033
##          Detection Rate : 0.7469
##    Detection Prevalence : 0.8619
##       Balanced Accuracy : 0.6726
##
##        'Positive' Class : No
##
```



El modelo formado con las variables de mayor importancia cuenta con una precisión del 82,02% y una especificidad de 0,3438%. Por tanto, se puede concluir que el mejor modelo de regresión logística sería el modelo 2, es decir, el modelo compuesto por las variables más importantes. Ya que este presenta, prácticamente, la misma precisión y tan solo un 2% menos de especificidad. Con lo cual, por el principio de parsimonia será mejor el modelo 2.

Modelo de análisis discriminante cuadrático

Se va a utilizar un **LDA o QDA** como método de clasificación a modo de observar cómo estima el modelo. Antes de comenzar es relevante tener en cuenta ciertas condiciones que tener en cuenta en este tipo de modelos de clasificación.

Las condiciones que se deben cumplir para que un Análisis Discriminante Lineal sea válido son:

- Cada predictor que forma parte del modelo se distribuye de forma **normal** en cada una de las clases de la variable respuesta. En el caso de múltiples predictores, las observaciones siguen una distribución normal multivariante en todas las clases.
- La **varianza** del predictor es **igual** en todas las clases de la variable respuesta. En el caso de múltiples predictores, la matriz de covarianza es igual en todas las clases. Si esto no se cumple se recurre a *Análisis Discriminante Cuadrático (QDA)*.

Cuando la condición de normalidad no se cumple, los modelos pierden precisión, pero aun así puede llegar a clasificaciones **relativamente buenas**.

Como se ha visto anteriormente, no se puede asumir normalidad univariante ni multivariante, así como varianza constante, por lo que se va a optar por usar el *QDA* debido a su mayor robustez y porque incurrirá en menos sesgo.

El *análisis discriminante cuadrático* es un método de clasificación utilizado en estadísticas y machine learning para separar dos o más grupos o clases basándose en un conjunto de variables predictoras. Este método se basa en el análisis discriminante lineal, pero utiliza una función discriminante cuadrática en lugar de lineal para separar las clases.

El análisis discriminante cuadrático es útil cuando se tiene un conjunto de datos con dos o más clases y se quiere predecir a qué clase pertenece un nuevo caso basándose en un conjunto de variables predictoras. Es especialmente útil cuando las **clases no están perfectamente separadas** por una línea recta y se necesita una función más flexible para separarlas, que es justamente lo que pasa en el caso de la base de datos, lo que **justifica** el uso de este modelo.

Método 1: Usando todas las variables.

En primera instancia, se usarán todas las variables para clasificar la variable `stroke` con el fin de analizar el modelo y después se intentará refinarlo para obtener un modelo lo más preciso posible con el mínimo de variables posibles.

Entrenamiento.

Se escogen un 80% de las observaciones para el conjunto de entrenamiento o *train* y el resto para el de prueba o *test*

Predicción.

```
## Quadratic Discriminant Analysis
##
## 4733 samples
##   10 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 4733, 4733, 4733, 4733, 4733, 4733, ...
## Resampling results:
##   Accuracy   Kappa
##   0.8128374  0.4636618
```

Evaluación

```
## Confusion Matrix and Statistics
##           Reference
## Prediction  No Yes
##        No  824 127
##        Yes  83 150
##                Accuracy : 0.8226
##                  95% CI : (0.7997, 0.844)
##     No Information Rate : 0.766
##     P-Value [Acc > NIR] : 1.28e-06
##
##                   Kappa : 0.4763
##
##  Mcnemar's Test P-Value : 0.003004
##
##             Sensitivity : 0.9085
##             Specificity : 0.5415
##          Pos Pred Value : 0.8665
##          Neg Pred Value : 0.6438
##              Prevalence : 0.7660
##          Detection Rate : 0.6959
##    Detection Prevalence : 0.8032
##       Balanced Accuracy : 0.7250
##        'Positive' Class : No
```

Como se puede observar con todas las variables se obtienen resultados que a priori se pueden considerar buenos, como la **sensibilidad** que es la tasa de verdaderos positivos del modelo. En este caso, el modelo ha identificado correctamente el 90.57% de las veces a las instancias que pertenecen a la clase positiva (1), en este caso es cómo es de bueno de modelo identificando casos negativos y se puede observar que es bastante preciso. La **precisión** que se refiere a la tasa de aciertos del modelo. En este caso, el modelo ha acertado en el 81.42% de los casos. Sin embargo, la **especificidad** que es la tasa de verdaderos negativos del modelo. En este caso, el modelo ha identificado correctamente el 52.3% de las veces a las instancias que pertenecen a la clase negativa (2). En este caso es el más importante porque es el objetivo principal del estudio, identificar correctamente los verdaderos positivos en cuanto a infarto cerebral se refiere. El modelo es bastante escaso en este ámbito.

Como métodos de evaluación, han sido probado más modelos a parte de los que guían el trabajo que también pueden ser válidos según el contexto, un ejemplo es *la validación cruzada*. La cual, es una técnica utilizada en machine learning para evaluar el rendimiento de un modelo de clasificación. Se utiliza para evaluar cómo el modelo se desempeña en datos que no se han utilizado para entrenarlo, lo que proporciona una medida más precisa del rendimiento del modelo en la práctica.

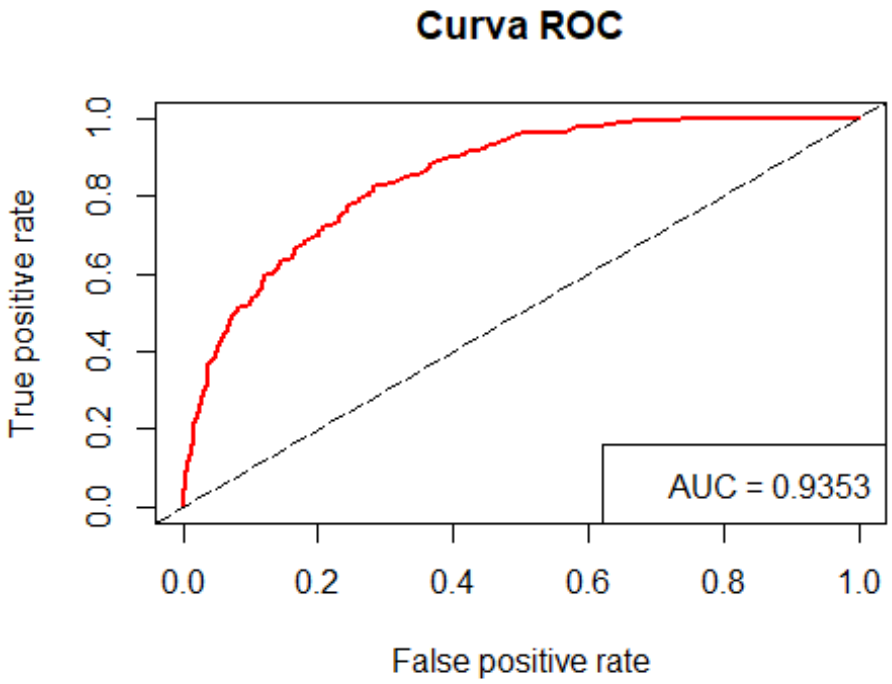
En la validación cruzada, se **dividen** los datos de entrenamiento en varios conjuntos o "pliegues" y se entrena el modelo utilizando un conjunto de datos y se evalúa su rendimiento en el conjunto de datos restante. Esto se repite varias veces, cada vez utilizando un conjunto diferente de datos como el conjunto de prueba y el resto como el conjunto de entrenamiento. Al final, se calcula una medida promedio del rendimiento del modelo en todos los pliegues, lo que proporciona una evaluación más precisa del rendimiento del modelo en datos que no se han utilizado para entrenarlo.

La validación cruzada es útil para evitar el **sobreajuste del modelo**, es decir, la capacidad del modelo para funcionar bien en los datos de entrenamiento, pero mal en datos nuevos. Al evaluar el rendimiento del modelo en datos que no se han utilizado para entrenarlo, se puede tener una idea más precisa de cómo se desempeñará el modelo en la práctica. Además, la validación cruzada es útil para comparar el rendimiento de diferentes modelos y elegir el que mejor se adapte a los datos.

También se usará la curva ROC como se ha venido haciendo anteriormente

```
## [1] 0.1100724 0.1100639
```

Se observa que los delta asociados son de 0,11 lo que a priori parece poco teniendo en cuenta que es un modelo de clasificación binomial pero sin compararlo con otros modelos no es posible tener una idea clara.



Se puede observar un área por debajo de la curva bastante amplia sin llegar a ser muy buena, reflejo claro de los estadísticos analizados anteriormente

Método 2: Usando Best-Subsets

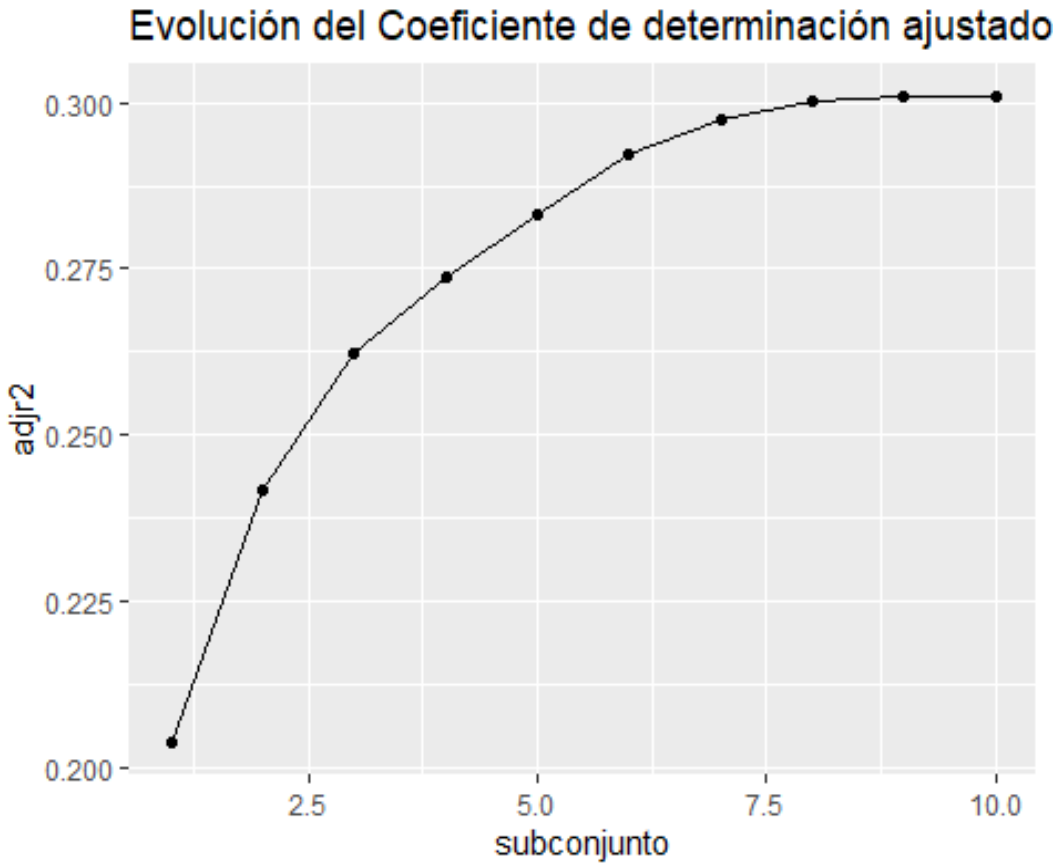
El método de selección de variables de best subset es una técnica utilizada en estadísticas y machine learning para seleccionar el **subconjunto óptimo** de variables de un conjunto más grande para utilizar en un modelo de clasificación. Este método se basa en la idea de evaluar diferentes combinaciones de variables y elegir la que mejor se ajuste al modelo y tenga el mejor rendimiento en términos de medidas de evaluación como la precisión o la sensibilidad.

Para utilizar el método de selección de variables de best subset, se siguen los siguientes pasos:

1. Seleccionar un conjunto de **variables candidatas** a incluir en el modelo.
2. Evaluar todas las posibles **combinaciones** de variables y seleccionar la que tenga el **mejor rendimiento** en términos de medidas de evaluación.
3. **Repetir** el proceso para **diferentes tamaños** de subconjunto de variables, desde el subconjunto más pequeño hasta el más grande.
4. Seleccionar el subconjunto de variables que tenga el mejor rendimiento en **todos los tamaños** de subconjunto evaluados.

```
## Subset selection object
## Call: regsubsets.formula(stroke ~ ., data = train_selection, method =
"exhaustive",
##     nvmax = 10)
## 10 Variables  (and intercept)
##                   Forced in Forced out
## gender                FALSE      FALSE
## age                   FALSE      FALSE
## hypertension          FALSE      FALSE
## heart_disease         FALSE      FALSE
## ever_married          FALSE      FALSE
## work_type             FALSE      FALSE
## residence_type        FALSE      FALSE
## avg_glucose_level     FALSE      FALSE
## bmi                   FALSE      FALSE
## smoking_status        FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##           gender age hypertension heart_disease ever_married work_type
## 1  ( 1 )  " "    "*" " "          " "           " "          " "
## 2  ( 1 )  " "    "*" " "          " "           " "          "*"
## 3  ( 1 )  " "    "*" " "          " "           "*"          "*"
## 4  ( 1 )  " "    "*" " "          " "           "*"          "*"
## 5  ( 1 )  "*"    "*" " "          " "           "*"          "*"
## 6  ( 1 )  "*"    "*" " "          " "           "*"          "*"
## 7  ( 1 )  "*"    "*" " "          "*"           "*"          "*"
## 8  ( 1 )  "*"    "*" "*"          "*"           "*"          "*"
## 9  ( 1 )  "*"    "*" "*"          "*"           "*"          "*"
## 10  ( 1 ) "*"    "*" "*"          "*"           "*"          "*"
##           residence_type avg_glucose_level bmi smoking_status
## 1  ( 1 )  " "            " "               " " " "
## 2  ( 1 )  " "            " "               " " " "
## 3  ( 1 )  " "            " "               " " " "
## 4  ( 1 )  " "            "*"               " " " "
## 5  ( 1 )  " "            "*"               " " " "
## 6  ( 1 )  "*"            "*"               " " " "
## 7  ( 1 )  "*"            "*"               " " " "
## 8  ( 1 )  "*"            "*"               " " " "
## 9  ( 1 )  "*"            "*"               " " "*"
## 10  ( 1 ) "*"            "*"               "*" "*"
```

En forma de tabla, se muestra qué variables escoger en función del número de variables que se quieran seleccionar. En este caso, se va a optar por intentar resumir el modelo lo máximo posible sin perder precisión.



En este gráfico se puede observar que *R^2* no aumenta a penas a partir de la selección de 7 variables por lo que serán las seleccionadas para hacer predicciones.

```
##       (Intercept)            gender               age      hypertensio
n
##      1.5683904780     -0.0709732692      0.0112114077     -0.080315228
0
##     heart_disease      ever_married         work_type    residence_typ
e
##     -0.1332703800     -0.1602031838     -0.0981638504     -0.074741459
4
## avg_glucose_level               bmi    smoking_status
##      0.0010674292     -0.0004644025     -0.0099493766
```

Entrenamiento.

Se mantendrán los criterios del modelo QDA 1.

Predicción.

```
## Quadratic Discriminant Analysis
##
## 4733 samples
##    7 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 4733, 4733, 4733, 4733, 4733, 4733, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.806529  0.4331416
```

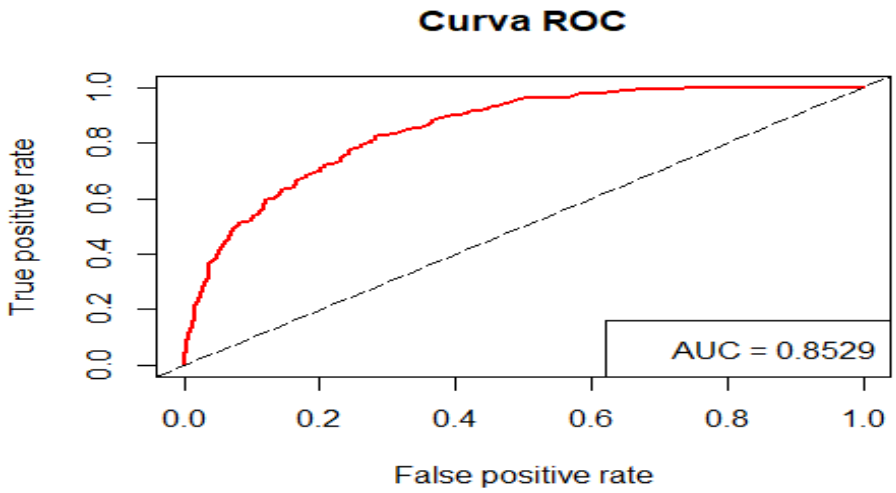
Evaluación.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  799 152
##        Yes  86 147
##
##                Accuracy : 0.799
##                  95% CI : (0.775, 0.8215)
##     No Information Rate : 0.7475
##     P-Value [Acc > NIR] : 1.720e-05
##
##                   Kappa : 0.4256
##
##  Mcnemar's Test P-Value : 2.516e-05
##
##             Sensitivity : 0.9028
##             Specificity : 0.4916
##          Pos Pred Value : 0.8402
##          Neg Pred Value : 0.6309
##              Prevalence : 0.7475
##          Detection Rate : 0.6748
##    Detection Prevalence : 0.8032
##       Balanced Accuracy : 0.6972
##
##        'Positive' Class : No
##
```

Como se puede apreciar se ha perdido un poco de precisión general en el modelo a costa de reducir un poco la complejidad de este. Pero debido a las variables que faltan (*bmi, gender y residence_type*) no merece mucho la pena excluirlas porque son datos relativamente sencillos de extraer.

```
## [1] 0.1135525 0.1135466
```

Además, se incurre en un poco más de error por lo que no sería un modelo preferible ante seleccionar todas las variables.



Además, se percibe a simple vista una pérdida considerable en el área por debajo de la curva ROC. Lo que confirma que el método de selección utilizado no es el más idóneo. Además, se incurre en un poco más de error por lo que no sería un modelo preferible ante seleccionar todas las variables.

Método 3: Usando variables normalizadas.

Se van a usar las variables normalizadas para comprobar si hay una mejora de la precisión y justifica la pérdida de interpretabilidad o no.

Entrenamiento.

Se mantendrán los criterios del modelo QDA 1.

Predicción.

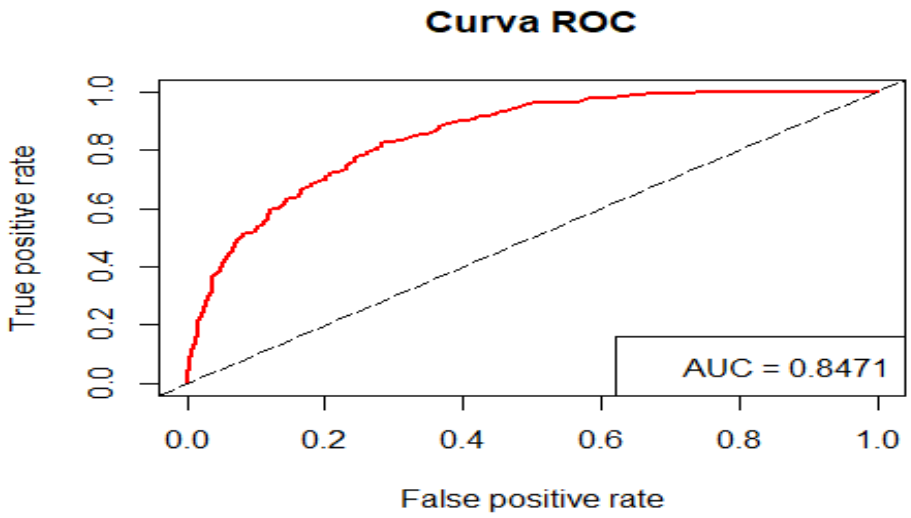
```
## Quadratic Discriminant Analysis
##
## 4733 samples
##   10 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 4733, 4733, 4733, 4733, 4733, 4733, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7941371  0.4562735
```

Evaluación.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  763 188
##        Yes  67 166
##
##                Accuracy : 0.7846
##                  95% CI : (0.7601, 0.8077)
##     No Information Rate : 0.701
##     P-Value [Acc > NIR] : 5.698e-11
##
##                   Kappa : 0.4304
##
##  Mcnemar's Test P-Value : 5.705e-14
##
##             Sensitivity : 0.9193
##             Specificity : 0.4689
##          Pos Pred Value : 0.8023
##          Neg Pred Value : 0.7124
##              Prevalence : 0.7010
##          Detection Rate : 0.6444
##    Detection Prevalence : 0.8032
##       Balanced Accuracy : 0.6941
##
##        'Positive' Class : No
##
```

Se puede observar que, pese a tener las variables normalizadas no aumenta la especificidad ni la precisión general del modelo, así como cualquier otro estadístico, como el Kappa, que ha empeorado y más teniendo en cuenta de que no se partía de un buen modelo con respecto a este estadístico.

```
## [1] 0.1096950 0.1096862
```



El error sí se puede apreciar que disminuyen, pero no lo suficiente, analizando el resto de estadísticos como el kappa, para justificar la pérdida de interpretabilidad.

Método 4: Usando RandomForest.

Una de las **principales ventajas** de Random Forest es su capacidad para seleccionar automáticamente las variables más importantes en el conjunto de datos y utilizarlas para realizar predicciones precisas.

El método de selección de variables basado en Random Forest funciona de la siguiente manera:

1. Se construye un **conjunto de árboles de decisión** utilizando una muestra aleatoria del conjunto de datos y un conjunto aleatorio de variables.

2. Se evalúa la **importancia de cada variable** en el conjunto de datos utilizando la poda de nodos y la reducción de la impureza en cada árbol.

3. Se seleccionan las variables más importantes basándose en su **importancia relativa** en el conjunto de árboles. Estas variables se utilizan entonces para construir el modelo final de clasificación.

```
##                   MeanDecreaseGini
## gender                    48.01660
## age                      644.10512
## hypertension              36.26422
## heart_disease             24.90043
## ever_married              48.89121
## work_type                 99.18859
## residence_type            44.06714
## avg_glucose_level        396.50541
```

```
## bmi                          299.22058
## smoking_status               117.64973
```

Se pueden apreciar la pureza de los nodos en todas las variables. Destacando dos que sobrepasan los 100 puntos. Para la selección se escogerán aquellas que sobrepasen los 90 ya que son evidentemente las más puras y que además son las más correlacionadas con stroke.

Entrenamiento.

Se mantendrán los criterios del modelo QDA 1.

Predicción.

```
## Quadratic Discriminant Analysis
##
## 4733 samples
##    4 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 4733, 4733, 4733, 4733, 4733, 4733, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7753169  0.2720808
```

Evaluación.

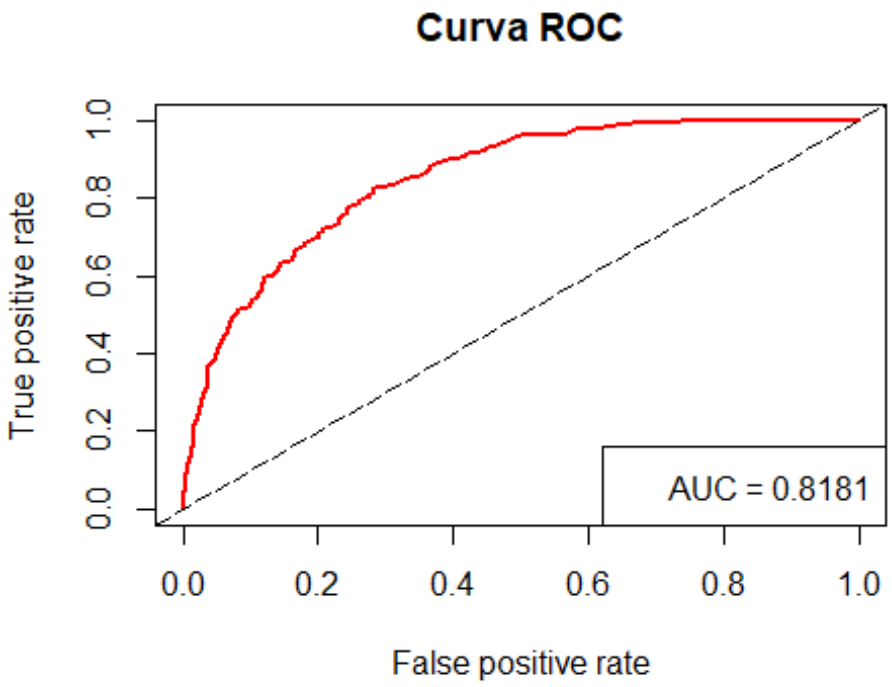
```
## Confusion Matrix and Statistics
##           Reference
## Prediction  No Yes
##        No  821 130
##        Yes 137  96
##
##                Accuracy : 0.7745
##                  95% CI : (0.7496, 0.798)
##     No Information Rate : 0.8091
##     P-Value [Acc > NIR] : 0.9987
##
##                   Kappa : 0.2785
##
##  Mcnemar's Test P-Value : 0.7135
##
##             Sensitivity : 0.8570
##             Specificity : 0.4248
##          Pos Pred Value : 0.8633
##          Neg Pred Value : 0.4120
##              Prevalence : 0.8091
##          Detection Rate : 0.6934
```

```
##    Detection Prevalence : 0.8032
##       Balanced Accuracy : 0.6409
##        'Positive' Class : No
##
```

Se puede observar nuevamente que se ha perdido precisión general (alrededor de 3 puntos), pero esta vez tenemos un modelo muy simple. Sería una posible elección si es necesario simplificar el modelo usando el mínimo de variables. Es necesario a tener en cuenta, que la **especificidad** ahora es inferior al 50% por lo que se podría acertar más en los casos de ictus positivos de forma totalmente aleatoria.

```
## [1] 0.1197767 0.1197735
```

El aumento del error en estos datos de nuevo no parece ser muy significativo, aunque se debe tener en cuenta que es el mayor de todos los utilizados.



Se aprecia que el índice ha bajado casi 10 puntos con respecto a sus predecesores, pero con menos variables escogidas, al estar por encima del 80% se podría llegar a la conclusión de que es una buena simplificación del modelo, sin llegar a afirmar en ningún caso, que se dispone de un buen modelo para predecir infartos cerebrales.

Modelo Random Forest

Método 1: Con hiperparámetros predeterminados

Se realiza un modelo random forest con los hiperparámetros sin modificar para comparar. Para clasificaciones los parámetros son los siguientes: mtry = $\sqrt{p}$ = 3 y nodesize = 1, ntree = 500.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1372  113
##        Yes   54  236
##
##                Accuracy : 0.9059
##                  95% CI : (0.8914, 0.9191)
##     No Information Rate : 0.8034
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6819
##
##  Mcnemar's Test P-Value : 7.184e-06
##
##             Sensitivity : 0.9621
##             Specificity : 0.6762
##          Pos Pred Value : 0.9239
##          Neg Pred Value : 0.8138
##              Prevalence : 0.8034
##          Detection Rate : 0.7730
##    Detection Prevalence : 0.8366
##       Balanced Accuracy : 0.8192
##
##        'Positive' Class : No
##
```

Tanto la precisión, como la especifidad para predecir positivos es menor que el modelo random forest con los hiperparámetros modificados.

Método 2: Con variables normalizadas

Se elabora un random forest con las variables normalizadas para ver si consigue mejorar la capacidad de predicción de los modelos.

Entrenamiento.

Se crea un data.frame supliendo las variables sin transformar por las transformadas y se entrena el modelo.

Predicción.

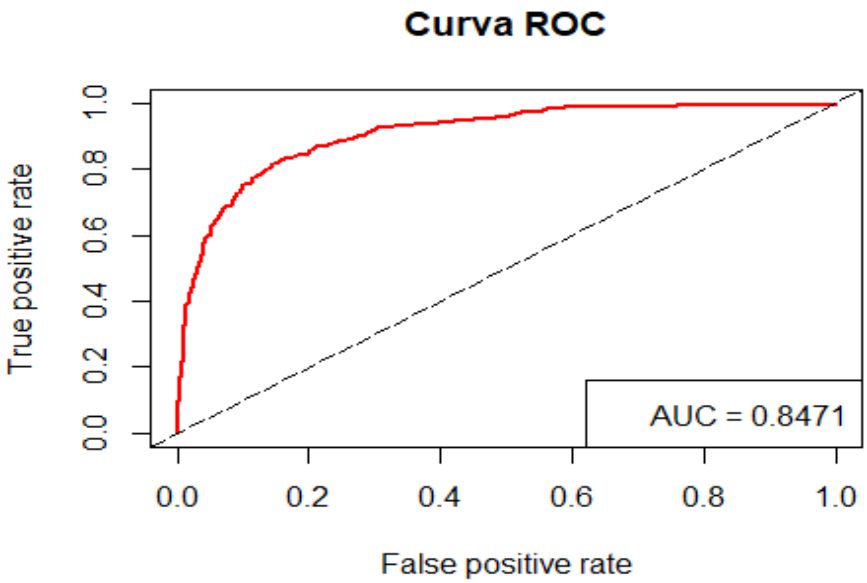
```
##  1  2  3  4  5
## No No No No No
## Levels: No Yes
```

Evaluación.

```
confusionMatrix(randompred2, test_n$stroke)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1331  140
##        Yes   71  209
##
##                Accuracy : 0.8795
##                  95% CI : (0.8633, 0.8944)
##     No Information Rate : 0.8007
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5922
##
##  Mcnemar's Test P-Value : 2.85e-06
##
##             Sensitivity : 0.9494
##             Specificity : 0.5989
##          Pos Pred Value : 0.9048
##          Neg Pred Value : 0.7464
##              Prevalence : 0.8007
##          Detection Rate : 0.7601
##    Detection Prevalence : 0.8401
##       Balanced Accuracy : 0.7741
##
##        'Positive' Class : No
##
```

Han bajado la precisión y la especificidad respecto a los modelos random forest con las variables no normalizadas.



El área por debajo de la curva representa un menor coeficiente con respecto a las variables sin transformar por lo que se desecha la elección de las variables normalizadas para el análisis debido a la suma de la pérdida de interpretabilidad.

Bucle para hiperparámetros

Se ha planteado un código que optimiza los hiperparámetros pero tarda una hora aproximadamente en ejecutarse, si se desea, se puede acudir al código adjuntado y en el *chunk name* cambiar el argumento *eval = FALSE* por *eval= TRUE* (chunk 152)

Modelo de NAIVE BAYES.

El método de clasificación de Naive Bayes es una técnica de aprendizaje automático muy utilizada en problemas de clasificación. Algunas cosas que se pueden tener en cuenta al utilizar este método son las siguientes:

- **La suposición de independencia condicional**: El método de Naive Bayes asume que los distintos atributos del conjunto de datos son independientes entre sí, lo que significa que el valor de un atributo no está correlacionado con el valor de los demás atributos. Esta suposición puede ser una simplificación excesiva en algunos casos y puede afectar al rendimiento del modelo, por lo tanto, es un aspecto muy a tener en cuenta.

- **La necesidad de una distribución normal**: El método de Naive Bayes suele prestar un mejor rendimiento cuando los atributos del conjunto de datos siguen una distribución normal o gaussiana. Si las variables seleccionadas no siguen una distribución normal, es posible que sea necesario aplicar algún tipo de transformación para tratar de normalizar los datos.

- **El tratamiento de atributos faltantes**: Naive Bayes no permite tratar atributos faltantes de manera directa. Si hay atributos faltantes en el conjunto de datos, es necesario aplicar algún tipo de técnica para tratar de imputar o estimar los valores perdidos (missing values).
- **La elección del tipo de distribución**: Permite elegir entre diferentes tipos de distribución para modelar los atributos del conjunto de datos. Algunos de los tipos de distribución más comunes son la distribución normal, la distribución Bernoulli y la distribución multinomial. Es importante elegir el tipo de distribución más adecuado para cada variable en función de su naturaleza y distribución.
- **La elección de los parámetros del modelo**: Naive Bayes permite ajustar algunos parámetros del modelo, como la métrica de distancia o la suavización de Laplace. Es importante elegir los parámetros adecuados para cada problema y evaluar el impacto de cada uno en el rendimiento del modelo.

Método 1: Usando todas las variables.

Entrenamiento.

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        No       Yes
## 0.7984066 0.2015934
##
## Conditional probabilities:
##      gender
## Y         [,1]      [,2]
##   No  1.411854 0.4922433
##   Yes 1.256287 0.4368440
##
##      age
## Y         [,1]     [,2]
##   No  41.95819 22.39513
##   Yes 68.04636 11.64892
##
##      hypertension
## Y         [,1]      [,2]
##   No  1.084971 0.2788811
##   Yes 1.122156 0.3276618
##
```

```
##      heart_disease
## Y         [,1]      [,2]
##   No  1.047777 0.2133273
##   Yes 1.071856 0.2584046
##
##      ever_married
## Y         [,1]      [,2]
##   No  1.643484 0.4790427
##   Yes 1.805988 0.3956752
##
##      work_type
## Y         [,1]      [,2]
##   No  2.739946 0.8865386
##   Yes 2.894611 0.5655219
##
##      residence_type
## Y         [,1]      [,2]
##   No  1.514968 0.4998515
##   Yes 1.367665 0.4824584
##
##      avg_glucose_level
## Y         [,1]     [,2]
##   No  104.3214 43.41427
##   Yes 134.0216 61.31425
##
##      bmi
## Y         [,1]     [,2]
##   No  28.42477 6.940451
##   Yes 30.14692 4.704360
##
##      smoking_status
## Y         [,1]      [,2]
##   No  2.399153 1.2636571
##   Yes 1.953293 0.9452509
```

Predicción.

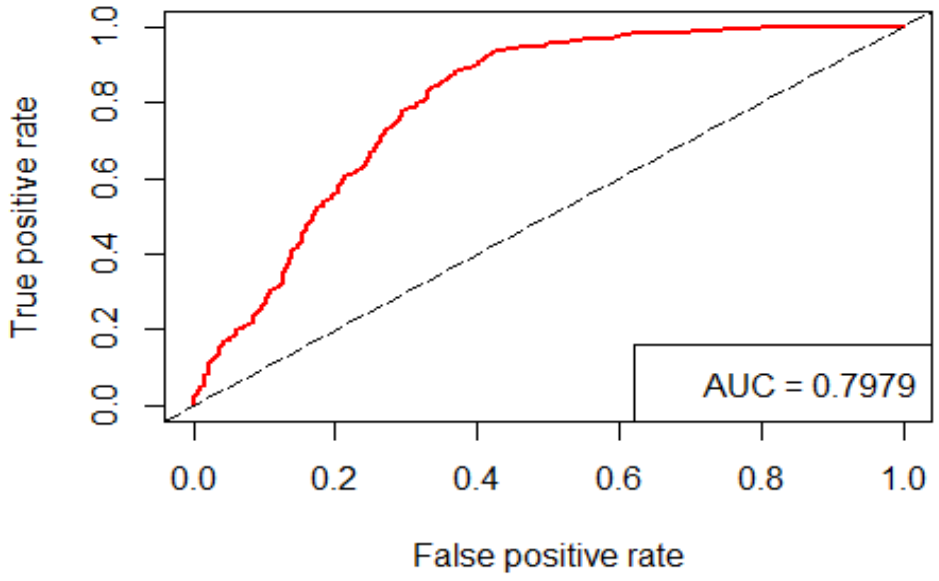
```
## [1] Yes Yes No  Yes No
## Levels: No Yes
```

Evaluación.

```
##
## nb.class   No  Yes  Sum
##      No  1156  158 1314
##      Yes  270  191  461
##      Sum 1426  349 1775
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1156  270
##        Yes  158  191
##
##               Accuracy : 0.7589
##                 95% CI : (0.7383, 0.7786)
##    No Information Rate : 0.7403
##    P-Value [Acc > NIR] : 0.03847
##
##                  Kappa : 0.3192
##
##  Mcnemar's Test P-Value : 8.078e-08
##
##             Sensitivity : 0.8798
##             Specificity : 0.4143
##          Pos Pred Value : 0.8107
##          Neg Pred Value : 0.5473
##              Prevalence : 0.7403
##          Detection Rate : 0.6513
##    Detection Prevalence : 0.8034
##       Balanced Accuracy : 0.6470
##
##        'Positive' Class : No
##
```

Gráfico 2.6: Curva ROC



Se puede observar un nivel de precisión general del casi 76% y una especificidad del 41,43% (se acierta más aleatoriamente), por lo que no es un modelo muy bueno para este conjunto de datos. Las medidas de bondad (Kappa y curva ROC) reflejan un modelo regular, sin llegar a arrojar resultados muy fiables.

Método 2: Usando Random Forest.

Siguiendo la estructura del resto de modelos se hará una clasificación usando el método de selección de variables, Random Forest.

```
##
## Call:
##  randomForest(formula = stroke ~ ., data = brain_oversampled,      imp
ortance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 7.82%
## Confusion matrix:
##       No Yes class.error
## No  4574 159  0.03359392
## Yes  304 880  0.25675676

##                              No         Yes MeanDecreaseAccuracy
## gender             0.002566628 0.071449375          0.016334597
## age                0.039058626 0.407144920          0.112617484
## hypertension       0.002205170 0.013359849          0.004438171
## heart_disease      0.001006481 0.007151975          0.002235251
## ever_married       0.013058960 0.046823469          0.019807885
## work_type          0.005211840 0.078680944          0.019905777
## residence_type     0.001034792 0.067285333          0.014274821
## avg_glucose_level -0.002705044 0.240792086          0.045948610
## bmi               -0.003648055 0.182629654          0.033573349
## smoking_status    -0.002350538 0.135513975          0.025201014
##                   MeanDecreaseGini
## gender                       46.62410
## age                         643.85670
## hypertension                 35.31946
## heart_disease                25.57781
## ever_married                 49.28706
## work_type                   101.34237
## residence_type               44.96537
## avg_glucose_level           396.31426
## bmi                         299.40398
## smoking_status              116.94817
```

Como puede apreciarse en la tabla de resultados que nos proporciona Random Forest, las variables que destacan con mucha diferencia y las cuales se aplicarán en este modelo son; age, work_type, avg_glucose_level, bmi y smoking_status.

Entrenamiento.

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        No       Yes
## 0.7984066 0.2015934
##
## Conditional probabilities:
##      age
## Y         [,1]     [,2]
##   No  41.95819 22.39513
##   Yes 68.04636 11.64892
##
##      work_type
## Y         [,1]      [,2]
##   No  2.739946 0.8865386
##   Yes 2.894611 0.5655219
##
##      avg_glucose_level
## Y         [,1]     [,2]
##   No  104.3214 43.41427
##   Yes 134.0216 61.31425
##
##      bmi
## Y         [,1]     [,2]
##   No  28.42477 6.940451
##   Yes 30.14692 4.704360
##
##      smoking_status
## Y         [,1]      [,2]
##   No  2.399153 1.2636571
##   Yes 1.953293 0.9452509
```

Predicción.

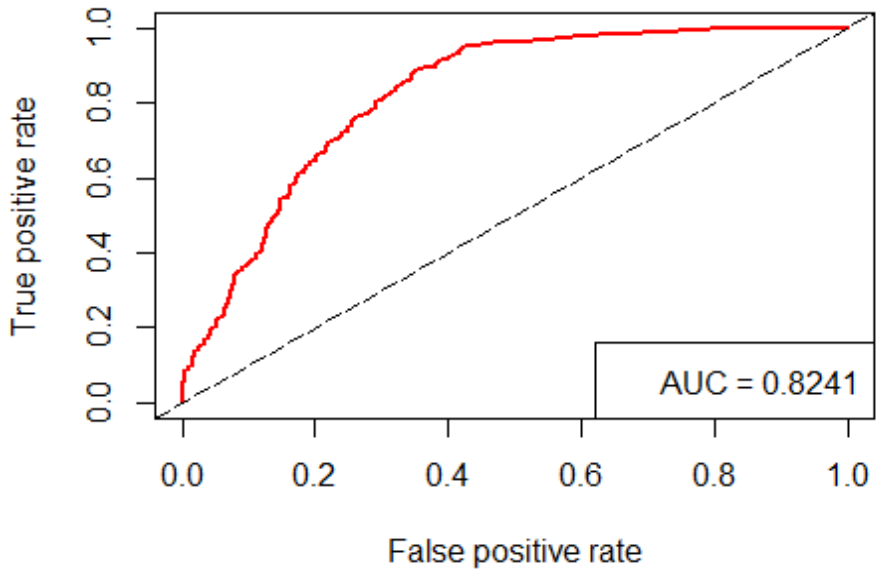
```
## [1] Yes Yes No  Yes No
## Levels: No Yes
```

Evaluación.

```
##
## nb.class   No  Yes  Sum
##      No  1211  159 1370
##      Yes  215  190  405
##      Sum 1426  349 1775
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1211  215
##        Yes  159  190
##
##                Accuracy : 0.7893
##                  95% CI : (0.7696, 0.8081)
##     No Information Rate : 0.7718
##     P-Value [Acc > NIR] : 0.041310
##
##                   Kappa : 0.3712
##
##  Mcnemar's Test P-Value : 0.004455
##
##             Sensitivity : 0.8839
##             Specificity : 0.4691
##          Pos Pred Value : 0.8492
##          Neg Pred Value : 0.5444
##              Prevalence : 0.7718
##          Detection Rate : 0.6823
##    Detection Prevalence : 0.8034
##       Balanced Accuracy : 0.6765
##
##        'Positive' Class : No
```

Gráfico 2.6: Curva ROC



Se puede observar un nivel de precisión general del casi 78,93% y una especificidad del 46,91%. Es decir, es un modelo que siendo simplificado predice y generaliza bastante major que incluyendo todas las variables. El índice Kappa arroja uno valor algo pobre de 37,12%, lo cual indica que los resultados no son fiables. La curva ROC, por el contrario, arroja mejores resultados, pero no se puede asegurar que sea buen modelo para el conjunto de datos (principalmente porque Naive Bayes funciona bien para conjuntos de datos con observaciones reducidas).