PREDICCIÓN DEL RIESGO DE AUSENCIA Y DESERCIÓN EN ESTUDIANTES UNIVERSITARIOS MEDIANTE TÉCNICAS DE MACHINE LEARNING

Carlos Cortés
Dirección de estudiantes
Universidad Central
Bogotá, Colombia
ccortesb@ucentral.edu.co

David Peña
Maestría en analítica de datos
Universidad Central
Bogotá, Colombia
dpenag2@ucentral.edu.co

Diego Villada
Departamento de Matemáticas
Universidad Central
Bogotá, Colombia
dvilladac1@ucentral.edu.co

20 de junio de 2020

RESUMEN

El interés de las instituciones de educación superior en mantener el número de estudiantes matriculados ha llevado a diversos tipos de análisis estadísticos que soporten las campañas de fidelización. Existen estudios que a través del ajuste de un modelo de regresión tratan de explicar el fenómeno; modelos de regresión logística, Maquina de soporte vectorial y árboles de decisión han sido planteados en este documento como una forma de mostrar el fenómeno y poder anticiparse al posible retiro de un estudiante.

Palabras Clave Métodos estadísticos multivariados, Machine learning, regresión logística, árbol de decisión, máquinas de soporte vectorial

1. Planteamiento del problema

Las instituciones de Educación Superior en Colombia han observado este año, alarmadas, la disminución del número de estudiantes matriculados y con ello han aumentado su interés en entender el fenómeno de la deserción estudiantil; aún cuando dicho fenómeno se ha estado monitoreando desde hace algunas décadas por el Ministerio de Educación Nacional (MEN), específicamente, por medio del sistema SPADIES[1]) desde el cual se ha trabajado en el tema de permanencia y graduación con las Instituciones de Eduación superior (IES).

La deserción estudiantil tiene al menos dos importantes facetas, una social, por su efecto en las familias, la sociedad y en la moral del desertor (fracaso) y otra económico- financiera, por el efecto en la productividad nacional y en los resultados de las IES como organizaciones; particularmente en las IES privadas. Ya en el 2010 el SPADIES había dejado claro que el principal problema es la falencia académica de los estudiantes que ingresan a las IES, dejando en segundo lugar las causas económicas y luego las personales e institucionales. (MEN, 2015)

Para medir la dimensión de la deserción existen dos indicadores principales que evidencian su magnitud: las tasas de deserción anual y por cohorte. La primera mide el porcentaje de estudiantes que estaban matriculados un año antes y que figuran como desertores un año después ... la tasa de deserción por cohorte muestra el porcentaje de no culminación de estudios, en tanto ilustra la cantidad de estudiantes que desertan de cada 100 que ingresan a algún programa universitario. (MEN, 2015)

En una investigación muy reciente hecha en los 35 países de la Unión Europea, aunque en mayor detalle en el sistema de educación superior del Reino Unido y Reino de Suecia, se establece que generalmente los investigadores y las universidades no llegan a diferenciar la deserción académica, de aquella "voluntaria" que los lleva hacia otros programas o universidades u otras formas de capacitación o estudio, o al mundo laboral, viajes de aventura, etc (Carlhed,2019)

[1] Sistema para la Prevención de la Deserción en las Instituciones de Educación Superior

La investigación de Carlhed (2019)muestra similitudes entre países, en cuanto a patrones y tendencias, si bien no necesariamente en los porcentajes o resultados exactos, y también encuentra diferencias entre carreras (p. e.: bajas tasas de deserción en ciencias de la salud en contraste con las altas en los programas del área económica e ingeniería), entre estudiantes urbanos y rurales, entre géneros (siendo los hombres más propensos a desertar), entre migrantes y nacionales, entre clases sociales, entre niveles de educación del padre y la madre y entre otros niveles socioeconómicos y culturales.

El auge de las teorías y las tecnologías de análisis de datos, específicamente el machine learning, brinda un cúmulo de técnicas que pueden presentar una solución viable al problema de identificación de estudiantes en riesgo de deserción. Sin embargo, hay que explorar las técnicas de aprendizaje para determinar cuáles son útiles en esta situación y en el marco de la maestría en analítica de datos, este problema representa una oportunidad de mostrar qué tan fiables pueden ser estas técnicas de modelamiento.

2. Objetivos

- Medir el impacto de los resultados académicos (promedios periodo y/o acumulado) en el ausentismo de los estudiantes de pregrado durante 7 periodos consecutivos, de la cohorte 2016-1.
- Establecer si existe algún tipo de relación entre las condiciones socio-económicas del estudiante y su promedio académico en la universidad.
- Determinar, en la medida de lo posible, un modelo de predicción de la deserción estudiantil

3. Justificación

El SAT[2] ha relacionado hasta ahora el riesgo de desertar de los estudiantes a sus resultados académicos, en línea con las conclusiones del Ministerio de Educación (MEN y Qualificar, 2015) pero contiene en su base de datos el registro de otros factores, susceptibles de ser analizados estadísticamente versus la deserción.

Dada la experiencia de la universidad y la literatura disponible tanto del Ministerio de Educación Nacional de Colombia como publicaciones académicas, el SAT ha tomado como criterios para evaluar el riesgo de deserción los resultados académicos obtenidos por los estudiantes durante su vida universitaria. Sin embargo, aún no se ha probado estadísticamente la correlación entre el rendimiento académico previo y la deserción o ausencia de ellos en los siguientes semestres.

Realizar este análisis de los datos del SAT permitiría probar o sino ajustar, las premisas de estos diagnósticos, permitiéndonos tener un mayor impacto en los planes de incremento de la permanencia estudiantil.

4. Marco teórico

4.1. Árboles de decisión

4.1.1. Introducción

En machine learning la metodología de árboles de clasificación o CART[3] por sus siglas en inglés, fue desarrollado por Breiman et al. (1984) [1]. Se pueden definir como un algoritmo de aprendizaje supervisado en donde siempre ha de existir una variable objetivo predefinida que se pretende modelar o clasificar, entendiendo por modelamiento a la elaboración de un modelo de regresión. Es usado principalmente en problemas de clasificación en donde las variables de entrada y salida pueden ser categóricas o continuas. La técnica se basa en dividir el espacio de predictores es decir, el conjunto de variables independientes en regiones distintas y no sobrepuestas.

4.1.2. Funcionamiento

Medidas de selección de atributos
Una medida de selección de atributos, ASM[4] por su siglas en inglés, se usa para seleccionar el criterio de división para los datos de forma óptima. Se conoce también como regla de división porque ayuda a determinar puntos de interrupción

[2]Sistema de Alertas Tempranas
[3]classification and regression tree
[4]Atribute selection measure

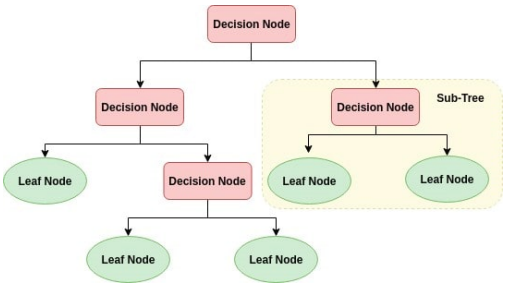


Figura 1: Diagrama ilustrativo de un árbol de decisión

para tuplas en un nodo dado. La ASM proporciona un rango para cada característica (o atributo) al explicar el conjunto de datos dado. El atributo de mejor puntuación se seleccionará como un atributo de división (Fuente).

En el caso de un atributo de valor continuo, los puntos de división para las ramas también deben definirse. Las medidas de selección más populares son Ganancia de información, Proporción de ganancia e Índice de Gini.()

4.1.3. Idea básica de implementación

Detrás del algoritmo de un árbol de decisión se resaltan los siguientes principios a seguir:

1. Seleccionar el mejor atributo utilizando medidas de selección de atributos (ASM) para dividir los registros.
2. Dicho atributo debe representar un nodo de decisión y dividir el conjunto de datos en subconjuntos más pequeños denominados *hojas*. Recursivamente para cada conjunto hoja hasta que cada una de las condiciones coincida:
   - Todas las tuplas pertenecen al mismo valor de atributo.
   - No quedan más atributos.
   - No hay más instancias.

El gráfico 1 ilustra la división del conjunto de acuerdo con las caracterísitcas expuestas anteriormente.

Toma de decisiones
Suponiendo que el espacio de un atributo se pueda plasmar en un plano cartesiano, la tarea de los árboles de decisión es dividir el espacio de dicho atributo en varias regiones rectangulares simples, divididas en secciones paralelas a los ejes. Para obtener la predicción para una observación particular, se utiliza la media o la moda de las observaciones del conjunto de entrenamiento que se encuentran dentro de la partición a la que pertenece la nueva observación.

Por lo general se puede definir el algoritmo de decisión mediante una fórmula matemática como la siguiente:

$$f(x) = \sum_{m=1}^{M} w_m \phi(x; v_m) \tag{1}$$

Donde w_m es la respuesta media en la región particular, ϕ es una función de decisión para la observación x basada en v_m que es un valor de umbral particular para la clase en cuestión.

4.1.4. Ejemplo

Para ilustrar más claramente el funcionamiento de un árbol de decisión, se recurre a la figura 2 en la que se muestra un subconjunto que contiene nuestros datos de ejemplo. Observa cómo se divide el dominio mediante divisiones paralelas de eje, es decir, cada división del dominio se alinea con uno de los ejes de características.

El concepto de división paralela de ejes se generaliza directamente a dimensiones superiores a dos.

Se crea ahora un árbol de regresión para poder realizar predicciones (Ver figura 3).

La heurística básica para crear un árbol de decisión es la siguiente:

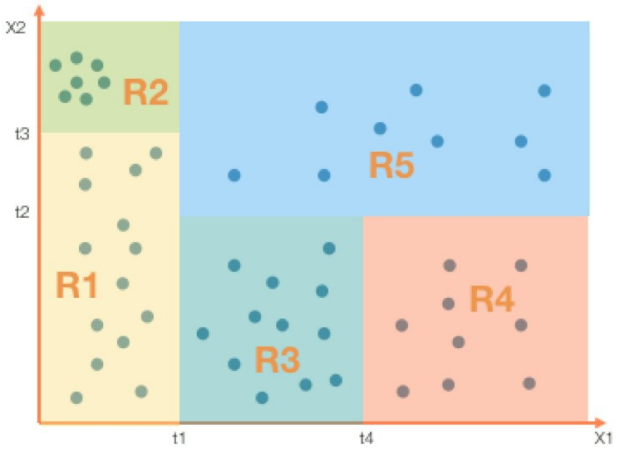


Figura 2: Espacio de características dividido

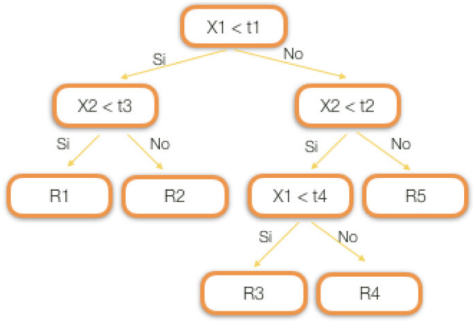


Figura 3: Árbol de decision para el ejemplo

- Dadas las características p, divide el espacio de características p-dimensional, en M regiones mutuamente distintas que cubren completamente el subconjunto del espacio de características y no se superponen. Estas regiones están dadas por $R_1, \ldots, R_m$.
- Cualquier observación nueva que caiga en una partición particular i tiene la respuesta estimada dada por la media de todas las observaciones de entrenamiento pertenecientes a la partición denotada por R_i.

4.2. Bosques aleatorios (Random Forest)

Los modelos de bosques aleatorios, realizan un procedimiento de arbol de decisión a muchas posibles formas del modelo haciendo que al final los tresultados promedio de todos estos árboles de decisión den como resultado una predicción mediada por varias situaciones en conjunto. Suelen ser muy útiles por la simplicidad de su idea general, derivada de la metodología de los árboles de decisión, pero computacionalmente muy costosos en el momento de su aplicación, motivo por el cual, para grandes cantidades de datos no son muy utilizados.

4.3. Máquinas de soporte Vectorial (SVM)

4.3.1. Introducción

El método de clasificación-regresión Máquinas de Vector Soporte o SVMs[5] por sus siglas en inglés, se desarrollaron en la década de los 90 , dentro de campo de la ciencia computacional. Si bien originariamente se desarrolló como un método de clasificación binaria, su aplicación se ha extendido a problemas de clasificación múltiple y regresión. Las SVMs han resultado ser de los mejores clasificadores que se pueden encontrar en los campos de la estadística y machine

[5] Vector Support Machines

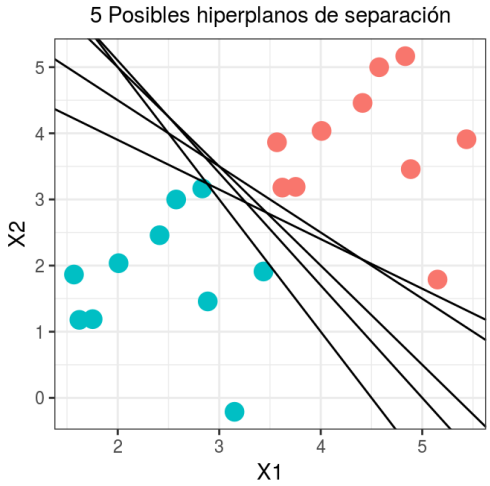


Figura 4: Varias lineas rectas separan los datos en un espacio.

learning debido a que construye un hiperplano sobre un espacio multidimensional para separar las diferentes clases. El SVM genera un hiperplano óptimo de forma iterativa, que se utiliza para minimizar un error. La idea central de SVM es encontrar un hiperplano marginal máximo que mejor divida el conjunto de datos en clases.

Las SVM de clasificación ofrecen una alta precisión en comparación con otros modelos de clasificación como la Regresión Logística y los Árboles de Decisión. Es conocido por su truco de kernel para manejar espacios de entrada no lineales. Es comúnmente utilizado en una variedad de aplicaciones tales como detección de rostros, detección de intrusos, clasificación de correos electrónicos, artículos de noticias y páginas web, entre otros.

En R, las librerías `e1071` y `LiblineaR` contienen los algoritmos necesarios para obtener modelos de clasificación simple, múltiple y regresión, basados en Support Vector Machines.

4.3.2. Funcionamiento

El concepto báísco de la SVM se puede resumir en que se traz un hiper plano en la dimensión p que separa los puntos en dos regiones del hiperespacio de tal forma que al evaluar nuevas entradas en la función de clasificación, se ddecide a qué región pertenece el individuo. En términos matemáticos, si la función de clasificación es:

$$y_i(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p) > 0, \;\; para\; i = 1...n$$

en donde la pertenencia o no a una clase estará determinada por el signo obtenido por la observación al ser evaluada en la función.

Como se muestra en la figura 4, se pueden obtener varias soluciones para este problema, y para resumir el subespacio de soluciones a que dan lugar las funciones que separan los conjuntos, se dibuja una franja establecida entre la línea central de la franja que separa los puntos más cercanos de subconjuntos distintos como se ilustra en la figura 5

4.4. Regresión Logística

La Regresión Logística Simple, desarrollada por David Cox en 1958, es un método de regresión que permite estimar la probabilidad de una variable cualitativa binaria en función de una variable cuantitativa. Una de las principales aplicaciones de la regresión logística es la de clasificación binaria, en el que las observaciones se clasifican en un grupo u otro dependiendo del valor que tome la variable empleada como predictor. [2]

Este tipo de regresión lleva el nombre de la función utilizada en el núcleo del método, la función *logística* es también llamada función *sigmoide* (figura 6) debido a la forma de su gráfica, la cual es muy similar a la letra S. Esta función puede tomar cualquier número de valor real y asignar a un valor entre 0 y 1.[**?**]

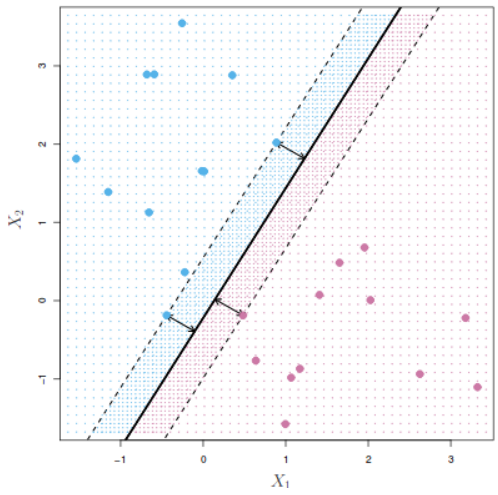


Figura 5: Un subespacio que contiene las posibles rectas que separan a los datos de distintas clases.

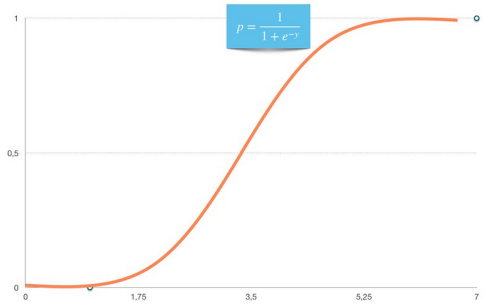


Figura 6: Función sigmoide

4.4.1. Funcionamiento

La idea básica de este tipo de regresión es muy similar a una de tipo lineal. Se estiman los coeficientes de regresión y se calculan los valores a predecir por medio de la función logística obtenida. Como la salida debe ser discreta, se asume que si el resultado obtenido por la estimación es menor a 0.5, la regresión a asignará a la predección el valor cero, de no ser así, el valor asignado es 1.

La fórmula matemática de una función logística es la descrita en la siguiente ecuación

$$f(x) = \frac{1}{1 + e^{-y}} \tag{2}$$

Si se tiene en cuenta que la variable y puede escribirse como una combinación lineal de las variables independientes para un individuo i, es decir

$$y_i = \beta_0 + \sum_{i=k}^{p} \beta_k x_k$$

la función logística puede escribirse en la siguiente forma :

$$f(x) = \frac{1}{1 + e^{-\left(\beta_0 + \sum_{i=k}^{p} \beta_k x_k\right)}} \tag{3}$$

Los valores de predicción para este tipo de modelo resultan siendo probabilidades de ocurrencia del evento positivo.

5. Metodología

5.1. Proceso de modelamiento

El problema de la deserción radica en el hecho del desconocimiento de la realidad del estudiante sobre todo en el ámbito social. Aunque muchas investigaciones se han determinado que los factores que más influyen en el rendimiento académico de un estudiantes son los socioeconómicos, el modelo cambia de una universidad a otra; no hay un modelo general para todas.

Para poder determinar la influencia de dichos factores en la universidad Central, se ha recurrido a una base de datos compuesta de 12352 individuos, observados a través de 49 varibles,

Esta base de datos será analizada utilizando las tres técnicas de machine learnig descritas en la sección **??**. Cada análisis da lugar a resultados que permitirán comparar cuál es el modelo que mejor clasifica a los estudiantes en riesgo de retiro de los que no.

5.2. Software

5.2.1. Librerías

Para la construcción de los algoritmos se han diseñado códigos del software `R` usando los siguientes paquetes:

```
library(tidyverse)
library(summarytools)
library(readxl)
library(caret)
library(VIM)
library(reshape2)
library(reshape)
library(fastDummies)
library(e1071)
library(ipred)
library(xgboost)
```

Los cuatro primeros fueron usados para la manipuplación limpieza y exploracion de los datos, los siguientes se usaron para tareas especificas de clasificación y predisción sobre los modelos.

5.2.2. Carga y limpieza de los datos

Se cargan los datos y se reasignan tipos de variables de acuerdo con su papel en la declaración de los modelos. Esto es, las variables de tipo caracter deben ser convertidas a factores para facilitar su manipulación durante el modelamiento, además de su interpretación. El código para la operación descrita es el siguiente:

```
base_final <- read_excel("base final.xlsx")
base_final$`NIVEL EN CURSO`<-as.factor(base_final$`NIVEL EN CURSO`)
```

La variable NIVEL EN CURSO se ha convertido en factor para hacer su analisis más sencillo buscando convertirlo en una variable dummy posteriormente.

Con la siguiente línea de código se establece una característica dicotómica para la variable objetivo y se presenta una tabla de distribución de los valores 0 (No desertor) y 1 (Posible desertor), en base al promedio semestral obtenido por el alumno:

```
> base_final$Posbile_Desertor<-ifelse(
> base_final$`PROMEDIO SEMESTRAL`< 3 & base_final$`PROMEDIO ACUMULADO`< 3,1,0)
> base_final$Posbile_Desertor <- as.factor(base_final$Posbile_Desertor)
> table(base_final$Posbile_Desertor)
    0     1
10968  1384
```

En la tabla de frecuencias se observa cómo están repartidas las observacioines de acuerdo con la varible objetivo, 10968 sin riesgo de deserción y 1384 en riesgo de acuerdo con sus promedios semestrales y globales.

En la siguiente línea se escogen las variables que se consideran relevantes para la elaboración del modelo final.

las variables finalmente escogidas fueron las relacionadas en el cuadro 5.2.2:

Posbile_Desertor	PROMEDIO ACUMULADO
DURACIÓN DE ANOS EN PLAN DE ESTUDIO	EDAD
CRÉDITOS REQUERIDOS DEL PLAN	NIVEL EN CURSO
CRÉDITOS CURSADOS EN EL PLAN	TIPO DE DOCUMENTO
CRÉDITOS HOMOLOGADOS	CARRERA
CRÉDITOS SUPERADOS EN EL PLAN	GENERO
CRÉDITOS NO SUPERADA EN EL PLAN	ESTADO CIVIL
CREDITOS PENDIENTES EN EL PLAN	Requiere algún apoyo técnico
AVANCE	Situación laboral
ASIGNATURAS CURSADAS EN EL PLAN	Salario
ASIGNATURA HOMOLOGADAS EN EL PLAN	Tipo de vivienda
ASIGNATURA SUPERADA EN EL PLAN	Tenencia de la vivienda
TOTAL ASIGNATURAS NO SUPERADAS	Estrato
PROMEDIO SEMESTRAL	

Cuadro 5.2.2: Variables elegidas para el modelamiento de la deserción.

De acuerdo con el cuadro anterior, se conforma la base de datos que se usará con fines de modelado, haciendo además que las varibles factor se conviertan en entradas tipo dummy para facilitar la lectura e interpretación de los resultados

```
Base_Modelo<- base_final[,
c(1,67,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,24,31,32,33,34,35,44,49)]
Base_Modelo<-dummy_cols(.data = Base_Modelo, remove_first_dummy = TRUE)
Base_Modelo<-as.data.frame(Base_Modelo)
```

Ahora, cómo buena práctica en el ámbito machine learning, se dividen los datos en subconjuntos de entrenamiento y prueba (train y test), de esta manera el modelo se calcula con el primer subconjunto y posteriormente se evalua contra la base de prueba para medir la efectividad de su respuestas

```
set.seed(1234)
index <- createDataPartition(Base_Modelo$Posbile_Desertor, p = 0.7, list = FALSE)
train_data <- Base_Modelo[index, ]
test_data  <- Base_Modelo[-index, ]
```

5.3. Formulación de los modelos

5.3.1. Modelo de bosque aleatorio

En el codigo siguiente se especifican las variables que se usarán para encontrar el modelo y se detalla el método de entrenamiento elegido, `rf` que significa random forest, un preprocesamiento a los datos que consiste en escalarlos y centrarlos, tal como se hace cuando se aplican metodos de reducción de dimensionalidad como el PCA[6], el remuestreo, mediante bootstraping en 4 etapas con subconjuntos de datos para las muestras ("down").

```
Modelo_ARBOL1 <- train(Posbile_Desertor~`DURACIÓN DE ANOS EN PLAN DE ESTUDIO`+
`CRÉDITOS REQUERIDOS DEL PLAN`+  `CRÉDITOS CURSADOS EN EL PLAN`+
`CRÉDITOS HOMOLOGADOS`+  `CRÉDITOS SUPERADOS EN EL PLAN`+
`CRÉDITOS NO SUPERADA EN EL PLAN`+  `CREDITOS PENDIENTES EN EL PLAN`+
`AVANCE`+  `ASIGNATURAS CURSADAS EN EL PLAN`+
`ASIGNATURA HOMOLOGADAS EN EL PLAN`+  `ASIGNATURA SUPERADA EN EL PLAN`+
`TOTAL ASIGNATURAS NO SUPERADAS`+  `PROMEDIO SEMESTRAL`+
`PROMEDIO ACUMULADO`+  `EDAD`+
`NIVEL EN CURSO_2`+  `NIVEL EN CURSO_3`+
`NIVEL EN CURSO_4`+  `NIVEL EN CURSO_5`+
`NIVEL EN CURSO_6`+  `NIVEL EN CURSO_7`+
`NIVEL EN CURSO_8`+  `NIVEL EN CURSO_9`+
`NIVEL EN CURSO_10`+  `TIPO DE DOCUMENTO_Cedula de Extranjería`+
`TIPO DE DOCUMENTO_Pasaporte`+  `TIPO DE DOCUMENTO_Tarjeta de Identidad`+
`CARRERA_ARTE DRAMÁTICO`+  `CARRERA_BIOLOGÍA`+
`CARRERA_CINE`+  `CARRERA_COMUNICACIÓN SOCIAL Y PERIODISMO`+
`CARRERA_CONTADURÍA PÚBLICA`+  `CARRERA_CREACIÓN LITERARIA`+
`CARRERA_DERECHO`+  `CARRERA_ECONOMÍA`+
`CARRERA_ESTUDIOS MUSICALES`+  `CARRERA_INGENIERÍA AMBIENTAL`+
`CARRERA_INGENIERÍA DE SISTEMAS`+  `CARRERA_INGENIERÍA ELECTRÓNICA`+
`CARRERA_INGENIERÍA INDUSTRIAL`+  `CARRERA_INGENIERÍA MECÁNICA`+
`CARRERA_MATEMÁTICAS`+  `CARRERA_MERCADOLOGÍA`+
`CARRERA_PUBLICIDAD`+  `CARRERA_TRABAJO SOCIAL`+
`GENERO_Masculino`+  `ESTADO CIVIL_Divorciado(a)`+
```

[6]Principal component analisys

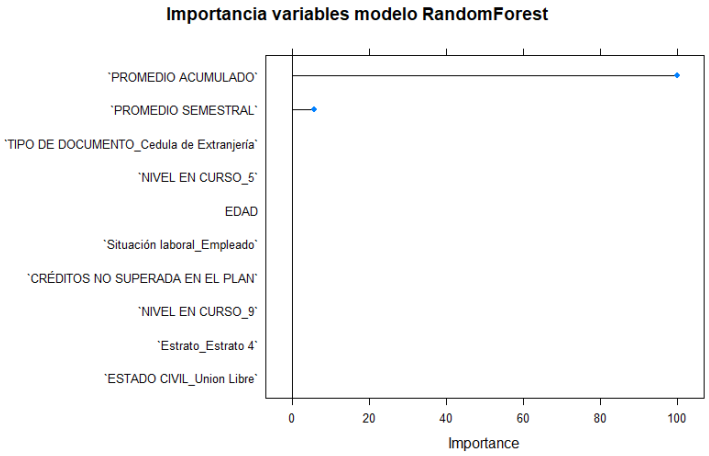


Figura 7: Importancia de las variables en el ajuste del modelo RF

```
‘ESTADO CIVIL_Madre soltera‘+  ‘ESTADO CIVIL_Separado(a)‘+
‘ESTADO CIVIL_Sin Informacion‘+
‘ESTADO CIVIL_Soltero(a) (Nunca se ha casado ni ha vivido en unión libre)‘+
‘ESTADO CIVIL_Union Libre‘+  ‘ESTADO CIVIL_Viudo(a)‘+
‘Requiere algún apoyo técnico_Sí‘+  ‘Situación laboral_Dueño o socio‘+
‘Situación laboral_Empleado‘+  ‘Situación laboral_Independiente‘+
‘Salario_1 a 2 SMMLV‘+  ‘Salario_2 a 5 SMMLV‘+
‘Salario_5 a 10 SMMLV‘+ ‘Salario_Más de 10 SMMLV‘+
‘Tipo de vivienda_Casa‘+  ‘Tipo de vivienda_Casa lote‘+
‘Tipo de vivienda_Finca‘+  ‘Tipo de vivienda_Habitación‘+
‘Tenencia de la vivienda_Cedida‘+  ‘Tenencia de la vivienda_Familiar‘+
‘Tenencia de la vivienda_Propia con pagos pendientes‘+
‘Tenencia de la vivienda_Propia totalmente pagada‘+
‘Tenencia de la vivienda_Vivienda estudiantil‘+
‘Estrato_Estrato 2‘+  ‘Estrato_Estrato 3‘+  ‘Estrato_Estrato 4‘+
‘Estrato_Estrato 5‘+  ‘Estrato_Estrato 6‘,
data =train_data,
method="rf",
preProcess = c("scale", "center"),
trControl=trainControl(method ="boot",
number =4,
verboseIter = FALSE,
sampling = "down"))
```

El resultado de este entrenamiento se muestra a continuación:

```
> Modelo_ARBOL1$finalModel

Call:
randomForest(x = x, y = y, mtry = param$mtry)
Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 75

OOB estimate of  error rate: 0.05%
Confusion matrix:
0   1 class.error
0 968   1 0.001031992
1   0 969 0.000000000
```

La tasa de error es muy baja, muestra que fallo el 0.05 % de la clasificación de los datos, lo que implica una precisión de casi 100 % en la clasificación de los datos. Contrario a lo que se pueda pensar este resultado solo da a entender que el modelo presenta sobreajuste, dejando en el ambiente una sensación de incredulidad que no puede ni debe ser aceptada.

El siguiente pedazo de código produce el gráfico muestra el aporte de las variables al ajuste del modelo:

```
varImp(Modelo_ARBOL1)
plot(varImp(Modelo_ARBOL1,scale = TRUE),10,main="Importancia variables modelo RandomForest")
```

Como se observa en el gráfico, solamente las variables de los promedios semestral y actual representan un verdadero aporte en el modelamiento de la posible deserción, las demás no contribuyen al modelo.

Ahora se contrasta este modelo con los datos de prueba. La siguiente línea de código explica cómo le va al modelo en términos de predicción.

```
> pred_test_random_forest<-predict(object = Modelo_ARBOL1,newdata = test_data,type = "raw")
> Matriz_confusion_test_RandonForest<-confusionMatrix(test_data$Posbile_Desertor,pred_test_random_forest)
> Matriz_confusion_test_RandonForest$table
Reference
Prediction    0    1
0  3290    0
1     0  415
```

La matriz de confusión muestra resultados similares a los de entrenamiento, para corroborarlo se muestran los indices complementarios emandos de la misma:

```
> Matriz_confusion_test_RandonForest$overall
Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
1.000000e+00   1.000000e+00   9.990048e-01   1.000000e+00   8.879892e-01

AccuracyPValue  McnemarPValue
7.087210e-192            NaN
```

Los índices de ajuste de los datos de prueba muestran una precisión del 100 % como en los datos de entrenamiento. Esto significa que el modelo no es convincente, porque las tasas de error y precision son demasiado altas, perfectas de por si, lo que da indicios de un sobreajuste en el modelo.

5.3.2. Máquina de soporte vectorial

Para la máquina de soporte vectorial se han elegido las mismas variables que se usaron en el random forest. El modelo se formula en R mediate las siguiente líneas de código

```
Modelo_SVM<-svm(Posbile_Desertor~‘DURACIÓN DE ANOS EN PLAN DE ESTUDIO‘+
CRÉDITOS REQUERIDOS DEL PLAN‘+
.
.
.
+ Estrato_Estrato 6‘,
data =train_data,
      probability = TRUE)
```

Es un modelo más ágil en términos computacionales, requiere menos exigencias de máquina y ese sentido puede realizar más iteraciones en busca de un número elevado de vectores de clasificación. De los modelos ususalmente utilizados para la clasificación, se encontró que el modelo que utiliza un kernel de tipo radial es escogido para la tarea que este ejercicio exige .

utiliza un kernel radial y la matriz de confusión muestra una tasa de error menor por lo que la accuracy también aumenta y el modelo parece ser más convincente.

```
summary(Modelo_SVM)

Parameters:
SVM-Type:  C-classification
SVM-Kernel:  radial
cost:  1

Number of Support Vectors:  1660

( 941 719 )


Number of Classes:  2

Levels:
0 1
```

El modelo muestra que de los posibles ajustes producidos, una fución con kernel radial es la más adecuada. Se estableció un costo por defecto de 1, formando así 1660 vectores de los cuales 941 corresponden a la categória de estudiantes sin riesgo de deserción, en tanto que 719 fueron elegidos para la complementaria.

En términos de los estadísticos respecto a la matríz de confusión se encuentra lo siguiente

```
>Modelo_SVM_pred_train<-predict(newdata = train_data,
object =Modelo_SVM,type = "class" )
```

```
>confusionMatrix(train_data$Posbile_Desertor,Modelo_SVM_pred_train)

          Reference
Prediction    0    1
0 7592   86
1  240  729

Accuracy : 0.9623
95% CI : (0.9581, 0.9662)
No Information Rate : 0.9057
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.7964

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9694
Specificity : 0.8945
Pos Pred Value : 0.9888
Neg Pred Value : 0.7523
Prevalence : 0.9057
Detection Rate : 0.8780
Detection Prevalence : 0.8879
Balanced Accuracy : 0.9319
```

La salida del programa muestra que la precisión del modelo llega a ser de un 96.23 % soportado por una sensibilidad del 96.94 % y una especificidad del 98.88 %. Realizando el mismo ejercicio para el conjunto de prueba utilizando el modelo ajustado se tiene lo siguiente:

```
Modelo_SVM_pred_test<-predict(newdata = test_data,object =Modelo_SVM,type = "class" )
confusionMatrix(test_data$Posbile_Desertor,Modelo_SVM_pred_test)
```

Con el conjunto de prueba se obtiene una precision del 94.87 % y la sensibilidad disminuye a 95.94 %.

5.4. Regresión logística

Para realizar este modelo ha de usarse la función glm del paquete base de R, el cual realiza una regresión basada en la familia binomial de distribuciones con los siguientes resultados

```
Modelo_Logistico <- glm(
Posbile_Desertor~.
,data = train_data,family = binomial)

summary(Modelo_Logistico)
```

Para resumir la salida, los coeficientes más significativos de esta estimación pertenecen a las variables que se presentan en la salida editada, en donde han sido removidas todas aquellas variables que de acuerdo con la tabla de estadísticos resultan ser no significativas para el ajuste de la regresión continuación:

```
(Intercept)                                                    ***
`NIVEL EN CURSO`6                                              .
`NIVEL EN CURSO`7                                              *
`NIVEL EN CURSO`8                                              .
`CRÉDITOS CURSADOS EN EL PLAN`                                 *
`CRÉDITOS NO SUPERADA EN EL PLAN`                              ***
`ASIGNATURAS CURSADAS EN EL PLAN`                              ***
`PROMEDIO SEMESTRAL`                                           ***
`PROMEDIO ACUMULADO`                                           ***
EstratoEstrato 4                                               *
EstratoEstrato 5                                               *
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 6066.82  on 8646  degrees of freedom
Residual deviance:  773.46  on 8570  degrees of freedom
```

```
AIC: 927.46

Number of Fisher Scoring iterations: 15
```

El deviance de los residuos resulta ser de 773.46 con 8570 grados de libertad, una cantidad relativamente baja si se tiene en cuenta la cantidad de observaciones que se tenián en total. El AIC [7] da como resultado 927.46 el cual se puede comparar con el número de variables que hay en el modelo y clarmente está muy alejado del valor original que es 67.

Para obtener la matriz de confusión han de tomarse los valores predichos por el modelo y reclasificarlos de acuerdo a un threshold de 0.2, en donde toda probabilidad predicha por el modelo menor a 0.2 será clasificada como 0, es decir, en bajo riesgo de deserción, y las que superen esta probabilidad se entenderán como 1 o sea en riesgo de deserción. Las siguiente líneas de código producen la transformación de la variable y de la misma manera, la matriz de confusión respectiva al modelo logístico:

```
> Pred_Logistica_Train <- predict(object = Modelo_Logistico,newdata = train_data,
type = "response" )

> Pred_Logistica_Train <- ifelse(Pred_Logistica_Train>0.2,1,0)
> Pred_Logistica_Train <- as.factor(Pred_Logistica_Train)
> confusionMatrix(train_data$Posbile_Desertor, Pred_Logistica_Train)$table
          Reference
Prediction    0    1
          0 7524  154
         1   20  949

>confusionMatrix(train_data$Posbile_Desertor, Pred_Logistica_Train)$overall
    Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
9.798774e-01   9.046466e-01   9.766931e-01   9.827324e-01   8.724413e-01
AccuracyPValue  McnemarPValue
5.792276e-290   6.588752e-24
```

La clasificación del modelo logístico en los datos de entrenamiento se produce una precisión del 97.98 %. Para probar el modelo en los datos de prueba, se utiliza el procedimiento descrito en el párrafo anterior y las líneas de código son las siguientes:

```
> Pred_Logistica_Test<-predict(object = Modelo_Logistico,newdata = test_data,type = "response" )
> Pred_Logistica_Test<-ifelse(Pred_Logistica_Test>0.2,1,0)
> Pred_Logistica_Test<-as.factor(Pred_Logistica_Test)
> confusionMatrix(test_data$Posbile_Desertor,Pred_Logistica_Test)$table
           Reference
Prediction     0    1
         0  3219   71
         1    20  395

> confusionMatrix(test_data$Posbile_Desertor,Pred_Logistica_Test)$overall
    Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
9.754386e-01   8.828235e-01   9.699287e-01   9.801797e-01   8.742240e-01
AccuracyPValue  McnemarPValue
2.070389e-109   1.593419e-07
```

El modelo logístico en los datos de prueba muestra una precisión del 97.54 %. Lo que representa un porcentaje alto de acierto en la predicción de los posibles desertores y que lo avala como el modelo con mejor desempeño en términos de la prediccción.

5.5. Refinamiento de los modelos

El siguiente

```
Base_Modelo$Probabilidad_Logistica<-predict(Modelo_Logistico, newdata = Base_Modelo,type = "response")
Base_Modelo$Clasificacion_Logistica<-ifelse(Base_Modelo$Probabilidad_Logistica>0.2, "Posible Desertor","No Desertor")

Base_Exportar<-left_join(base_final,Base_Modelo[,c(1,91,92)])
Base_Exportar$Posbile_Desertor<-ifelse(Base_Exportar==0,"No Desertor", "Posible Desertor")
Base_Exportar$Probabilidad_Logistica<-as.numeric(Base_Exportar$Probabilidad_Logistica)

library(writexl)
write_xlsx("D:/DEMOS/Proyecto fin curso DP/base_TABLEAU.xlsx",
x =Base_Exportar )
```

[7] Akaike's Information Criteria

6. Resultados

6.1. Resultados iniciales

En este sentido se presentan los resultados de los corrimientos de los dos métodos cuando se remueven las variables promedio acumulado y promedio semestre .

6.2. Modelos refinados

En esta sección se discuten los resultados obtenidos teniendo en cuenta las modificaciones que se han realizado a los diferentes modelos. Por ejemplo para el modelo de Random Forest y las máquinas de soporte vectorial han sido eliminadas las variables promedios semestre y promedio acumulado teniendo en cuenta que la variable objetivo `posible\_desertor` ha sido construida como una combinación lineal de las mencionadas anteriormente.

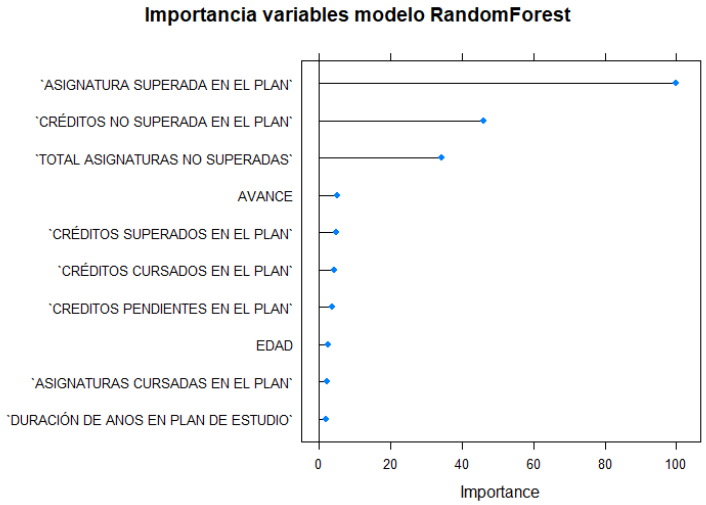
Los siguientes son los resultados de los modelos luego de las modeficaciones descritas arriba.

6.2.1. Bosque aleatorio

Una vez removidas las variables mencionadas en el párrafo anterior se modifican las líneas de código de tal manera que el modelo ahora a correr da como resultado la siguiente matriz de confusión:

```
> Modelo_ARBOL2$finalModel

Call:

randomForest(x = x, y = y, mtry param$mtry)

Type of random forest: classification
                      Number of trees: 500

No. of variables tried at each split: 73
OOB estimate of  error rate: 5.62%

Confusion matrix:
          0   1 class.error

  0 900  69  0.07120743

  1  40 929  0.04127967
```

Como se observa en la salida del programa, La tasa de error aumentado a 5.62 % lo que indica también una disminución en la precisión que no necesariamente es nociva para la consideración del modelo. Todo lo contrario, en este momento es más creíble que el modelo tenga algunas reservas con respecto a las predicciones en el entrenamiento en contraste con aquellas emanadas del conjunto de prueba.



El análisis gráfico muestra que las *asignaturas superadas en el plan* y los *créditos aún no superados en el plan* son las variables que más importancia tienen en la varianza del modelo seguidas de total asignaturas no superadas avance y créditos superados en el plan. Este resultado da a entender que ninguna de las variables socioeconómicas consideradas para la elaboración del modelo tienen relevancia a la hora decidir como estudiante si se retira o no.

```
> Matriz_confusion_test_RandonForest2$table
Reference
Prediction    0    1
0 3047  243
1   22  393

> Matriz_confusion_test_RandonForest2$overall
Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
9.284750e-01   7.083176e-01   9.196992e-01   9.365673e-01   8.283401e-01

AccuracyPValue  McnemarPValue
7.424452e-72   1.284382e-41
```

Por otro lado, la matriz de confusión encontrar para el grupo prueba muestra que existen 243 observaciones qué siendo reales fueron mal clasificadas por el modelo en tanto que 22 de las clasificadas como no desertoras reales fueron predichas cómo riesgo de predicción . así las cosas la matriz de confusión presenta una precisión del 92.84 % lo que lo deja en un mejor plano a considerar a la hora de decidirse por esta metodología para el modelamiento de la deserción en la Universidad Central.

6.2.2. Máquinas de soporte vectorial

Análogo al ejercicio realizado con la metodología Random Forest, en la máquina de soporte vectorial también se han removido las variables promedio acumulado y promedio semestre el resultado de esta variación en el modelo se presenta en la siguiente línea de código :

```
Modelo_SVM2_pred_train<-predict(newdata = train_data,
object =Modelo_SVM,type = "class" )
confusionMatrix(train_data$Posbile_Desertor,Modelo_SVM2_pred_train)

Confusion Matrix and Statistics

Reference
Prediction    0    1
0 7592   86
1  240  729

Accuracy : 0.9623
95% CI : (0.9581, 0.9662)
```

```
No Information Rate : 0.9057
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.7964

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9694
Specificity : 0.8945
Pos Pred Value : 0.9888
Neg Pred Value : 0.7523
Prevalence : 0.9057
Detection Rate : 0.8780
Detection Prevalence : 0.8879
Balanced Accuracy : 0.9319

'Positive' Class : 0
```

Los resultados de la matriz de confusión y la salida del paquete R muéstrame 96 observaciones mal clasificadas como estudiantes sin riesgo de deserción y 240 estudiantes que no están en riesgo de deserción y el modelo clasifica cómo que sí no está. Estos resultados son prácticamente idénticos a los obtenidos con el primer modelo generado antes de sacar las variables promedio acumulado y promedio semestre lo que indica que la máquinas de soporte vectorial es invariante a la eliminación de estas dos variables .

```
Confusion Matrix and Statistics

Reference
Prediction    0    1
0 3237   53
1  137  278

Accuracy : 0.9487
95% CI : (0.9411, 0.9556)
No Information Rate : 0.9107
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.7172

Mcnemar's Test P-Value : 1.729e-09

Sensitivity : 0.9594
Specificity : 0.8399
Pos Pred Value : 0.9839
Neg Pred Value : 0.6699
Prevalence : 0.9107
Detection Rate : 0.8737
Detection Prevalence : 0.8880
Balanced Accuracy : 0.8996

'Positive' Class : 0
```

Los resultados para y conjunto de prueba evidencia 53 personas en riesgo de deserción mal clasificadas por el modelo y siento 37 personas que no lo estaban clasificadas ahora como en posible riesgo de deserción.

6.2.3. Modelo de Regresión logística

Una vez removidas las varibles promedio acumulado y promedio semestre, los resultados del modelo de regresión logística se vuelcan sobre las demas variables, las cuales son referidas en la siguiente salida del programa:

```
Coefficients: (1 not defined because of singularities)
                                          Estimate Std. Error z value Pr(>|z|)
    `CRÉDITOS CURSADOS EN EL PLAN`         -0.03132    0.01108  -2.827 0.004704 **
    `CRÉDITOS HOMOLOGADOS`                 -0.34564    0.06443  -5.365 8.12e-08 ***
    `CRÉDITOS NO SUPERADA EN EL PLAN`       0.09142    0.02413   3.788 0.000152 ***
    `ASIGNATURAS CURSADAS EN EL PLAN`       0.21448    0.07465   2.873 0.004064 **
```

```
    ‘ASIGNATURA HOMOLOGADAS EN EL PLAN‘                      1.22830    0.21124   5.815 6.07e-09 ***
‘ASIGNATURA SUPERADA EN EL PLAN‘                            -1.40388    0.08969 -15.652  < 2e-16 ***
     EDAD                                                    0.03996    0.02162   1.848 0.064531 .
‘NIVEL EN CURSO_4‘                                           1.74214    0.62711   2.778 0.005468 **
‘NIVEL EN CURSO_5‘                                           3.57151    0.82624   4.323 1.54e-05 ***
‘NIVEL EN CURSO_6‘                                           5.35823    1.04879   5.109 3.24e-07 ***
‘NIVEL EN CURSO_7‘                                           6.84339    1.25797   5.440 5.33e-08 ***
‘NIVEL EN CURSO_8‘                                           7.25031    1.53687   4.718 2.39e-06 ***
‘NIVEL EN CURSO_9‘                                           7.18200    1.74320   4.120 3.79e-05 ***
‘NIVEL EN CURSO_10‘                                          7.31069    1.88470   3.879 0.000105 ***
‘CARRERA_COMUNICACIÓN SOCIAL Y PERIODISMO‘                   -1.35716    0.77920  -1.742 0.081553 .
‘CARRERA_CONTADURÍA PÚBLICA‘                                  1.23881    0.48877   2.535 0.011258 *
 CARRERA_DERECHO                                             1.16408    0.60902   1.911 0.055953 .
‘Estrato_Estrato 5‘                                          1.84210    0.97976   1.880 0.060087 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 6066.8  on 8646  degrees of freedom
Residual deviance: 1589.4  on 8574  degrees of freedom
AIC: 1735.4

Number of Fisher Scoring iterations: 15
```

Como principal resultado de esta salida editada, hay que decir que el valor del AIC ha aumentado considreablemente, pues en el modelo anterior el llamado saturado, el AIC presentaba un valor de 927.46 versus 1735.4 del modelo actual. La remoción de las variables Promedio_Acumulado y Promedio_Semestre produjo un súbito aumento en el valor del criterio de informacion de Akaike que obliga a reformular el modelo, esta vez insertando solo las variables que se mostraron significativas en este segundo modelo evaluado. Un dato no menor es que el intercepto no es un coeeficiente significativo.

Se realiza la reformulación del modelo, usando las variables significativas de este último encontrando lo siguiente:

```
Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept)                                -1.286194   0.342546  -3.755 0.000173 ***
‘CRÉDITOS CURSADOS EN EL PLAN‘              0.025879   0.009025   2.867 0.004138 **
‘CRÉDITOS HOMOLOGADOS‘                     -0.149742   0.060793  -2.463 0.013773 *
‘CRÉDITOS NO SUPERADA EN EL PLAN‘           0.038085   0.021064   1.808 0.070598 .
‘ASIGNATURAS CURSADAS EN EL PLAN‘           0.366975   0.063635   5.767 8.08e-09 ***
‘ASIGNATURA HOMOLOGADAS EN EL PLAN‘         1.019456   0.194978   5.229 1.71e-07 ***
‘ASIGNATURA SUPERADA EN EL PLAN‘           -0.958969   0.072781 -13.176  < 2e-16 ***
EDAD                                       0.076599   0.014799   5.176 2.27e-07 ***
‘NIVEL EN CURSO_4‘                         1.499811   0.355246   4.222 2.42e-05 ***
‘NIVEL EN CURSO_5‘                         2.719741   0.447137   6.083 1.18e-09 ***
‘NIVEL EN CURSO_6‘                         4.263612   0.550488   7.745 9.55e-15 ***
‘NIVEL EN CURSO_7‘                         5.988288   0.646159   9.268  < 2e-16 ***
‘NIVEL EN CURSO_8‘                         6.355305   0.841688   7.551 4.33e-14 ***
‘NIVEL EN CURSO_9‘                         5.616654   1.072768   5.236 1.64e-07 ***
‘NIVEL EN CURSO_10‘                        7.380378   1.191732   6.193 5.90e-10 ***
‘CARRERA_COMUNICACIÓN SOCIAL Y PERIODISMO‘ -0.875500   0.276965  -3.161 0.001572 **
‘CARRERA_CONTADURÍA PÚBLICA‘               0.185738   0.231989   0.801 0.423344
CARRERA_DERECHO                            0.120552   0.363591   0.332 0.740221
‘Estrato_Estrato 5‘                        1.200511   0.827873   1.450 0.147026
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 6066.8  on 8646  degrees of freedom
Residual deviance: 1905.5  on 8628  degrees of freedom
AIC: 1943.5

Number of Fisher Scoring iterations: 10
```

Aunque el AIC aumentó de 1735.4 a 1943.5, es de anotar que la remoción de las varibles significativas parace ser una solución adecuada desde el punto de vista de la propiedad de la parsimonia en ciencias, en el cual, si un modelo necesita pocos argumentos para su explicación es más confiable. En este caso, el aumento en el AIC representa un cambio aceptable en el que esta última combinación de variables puede ser la mejor para explicar el comportamiento de las variables.

En cuanto a los resultados de la matriz de confusión, las siguientes líneas de código muestran el comportamiento del modelo en los datos tanto de entrenamiento como de prueba:

```
> Pred_Logistica_Train2<-predict(object = Modelo_Logistico2,newdata = train_data,
```

```
+                             type = "response" )
> Pred_Logistica_Train2<-ifelse(Pred_Logistica_Train2>0.2,1,0)
> Pred_Logistica_Train2<-as.factor(Pred_Logistica_Train2)
> confusionMatrix(train_data$Posbile_Desertor,Pred_Logistica_Train2)$table
Reference
Prediction    0    1
         0 7266  412
         1   73  896
> confusionMatrix(train_data$Posbile_Desertor,Pred_Logistica_Train2)$overall
Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull AccuracyPValue  McnemarPValue
9.439112e-01   7.555254e-01   9.388521e-01   9.486685e-01   8.487337e-01  4.580298e-170   3.664626e-53
```

Los datos de entrenamiento muestran que la precisión de este modelo está cerca al 94.39 % que es un resultado aceptable teniendo en cuenta que se han eliminado las variables que producían multicolinealidad y a su vez dependencia y sobre ajuste. El resultado sobre el conjunto de prueba desvela lo siguiente:

```
> Pred_Logistica_Test2<-predict(object = Modelo_Logistico2,newdata = test_data, type = "response" )
> Pred_Logistica_Test2<-ifelse(Pred_Logistica_Test2>0.2,1,0)
> Pred_Logistica_Test2<-as.factor(Pred_Logistica_Test2)
> confusionMatrix(test_data$Posbile_Desertor,Pred_Logistica_Test2)$table
          Reference
Prediction    0    1
         0 3102  188
         1   37  378
> confusionMatrix(test_data$Posbile_Desertor,Pred_Logistica_Test2)$overall
Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull AccuracyPValue  McnemarPValue
9.392713e-01   7.365968e-01   9.310931e-01   9.467476e-01   8.472335e-01   8.147976e-68   1.523971e-23
```

La redución en la precisión de predicción respecto al modelo de entrenamiento fus muy pequeña, se pasó de un 94.39 % en el entrenamiento a un 93.92 % en el conjunto de prueba, lo que representa un desempeño uniforme en ambos conjuntos de datos por parte del modelo y práctimente

7. Análisis y discusión de resultados

El problema es principalmente de clasificación, por ello, fue abordado desde la metodología de machine learning teniendo como métodos de contraste tres tipos de modelos usados con bastante regularidad en este tipo de ejercicios: Random Forest, Máquina de Soporte Vectorial y Regresión Logística

En términos generales, el modelo Random Forest representó un modelamiento sobre ajustado a los datos teniendo en cuenta que la variables predictoras fueron reducidas a únicamente dos: PROMEDIO ACUMULADO y PROMEDIO SEMESTRE. Este mismo problema se presentó en los demás modelos de clasificación, motivo por el cual hubo que remover las variables en aras de buscar un mejor aprovechamiento de las mismas al momento de calcular sus coeficientes.

7.1. Tasas de precisión

En cuanto a la clasificación de los modelos sí encontró que cuando no habían sido removidas las variables promedio acumulado y promedio semestre las tasas de error eran demasiado bajas y en consecuencia la precisión muy alta esto da cuenta de un sobreajuste en los modelos indeseado. de esta manera y para corregir el error, han sido removidas estas dos últimas variables a fin de producir modelos más acertados por lo menos desde el punto de vista predictivo.

El modelo Random Forest se pasó de una precisión del 100 % a una del 92.79 % cuando se comparan las matrices de confusión para los modelos antes y después de remover estas variables redundantes, las cual es en primera instancia no mostraban diferencia alguna en la precisión obtenido a partir de las matrices de confusión de los conjuntos entrenamiento y prueba. Posteriormente, al remover las variables La diferencia entre los conjuntos de entrenamiento y prueba pasó de una precisión del 93.86 % a una precisión del 92.79 % en el conjunto de prueba.

Ya para el modelo máquinas de soporte vectorial la precisión del conjunto de entrenamiento fue el 98.6 % mientras que en el conjunto de prueba del 97.14 %, en contraste a esto, una vez removidas las variables, la diferencia en la precisión entre los conjuntos de entrenamiento y prueba Pasó del 96.23 % a 94.87 % respectivamente, dejando entrever que la remoción de las variables no afectó de manera significativa precisión y agregando una confianza al modelo tras la remoción de las variables redundantes. Sin embargo la diferencia entre la precisión de los conjuntos de entrenamiento y prueba manifiesta una falta de consistencia que ha de contrastarse con el ajuste de los otros dos modelos.

En el caso de la regresión logística, se pudo observar que el modelo completo aún con las variables redundantes mostrar una precisión del 97.99 % en la data de entrenamiento, mientras que el conjunto de prueba este indicador fue el

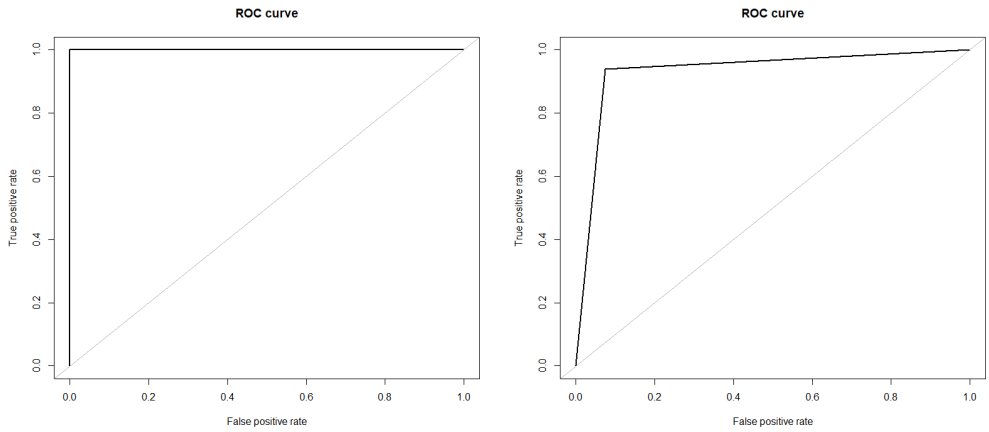


Figura 8: Comparación de las curvas ROC para el modelo Random Forest antes y después de la remoción de variables redundantes.

97.54 %. los resultados obtenidos remover las variables redundantes Y las variables que no salieron significativas en el primer modelo mostraron una disminución en la precisión para el conjunto de entrenamiento del 94.39 % y Asimismo evaluando en el conjunto de prueba la precisión solamente mostró una disminución de 0.04 % es decir pasó al 93.92 %. esta diferencia tan leve en la precisión del modelo logístico cuando se contrastan entrenamiento y prueba hace que este modelo se comporte como el más consistente entre los 3 que hemos probado y haciendo las modificaciones que se han comentado anteriormente.

7.2. Curvas ROC

En cuanto a las curvas roc un estadístico correcto para evaluar su ajuste es el área debajo de la misma. para el modelo de Random Forest se contrataron las curvas roc antes y después de quitar las variables redundantes los resultados que pasamos de un área de 100 % a 93.3 % una vez removidas dichas variables lo que representa una diferencia del 6.7 % . En el modelo de máquinas de soporte vectorial (SVM), el área bajo la curva antes de la remoción de las variables fue del 89.3 % en tanto que el área bajo la curva después de la remoción pasó al 82.7 % , no mente una diferencia de alrededor de 6.6 %. en la regresión logística se pudo observar una disminución menos significativa pasó del 96.5 % al 92.7 %, es decir una reducción de 3.8 % del área qué representa el poder de predicción del modelo y desde otra perspectiva, es resaltar el hecho qué son las áreas bajo la curva más altas encontradas para los diversos modelos descartando claramente al modelo Random Forest inicial que tenía un área de 100 %.

Las graficas 8, 9 y 10 muestran en comparación las curvas ROC de cada modelo antes y después de la remoción de las varibles redundantes PROMEDIO_ACUMULADO y PROMEDIO_SEMESTRE:

8. Conclusiones

Como conclusiones principales de este ejercicio se han encontardo las siguientes observaciones:

8.1. Sobre los datos

La base de datos usada para este ejercicio inició con poco más de 50000 registros de los cuales tras un proceso de limpieza bastante largo, se pudieron obtener 12352 con la información necesaria para el analisis posterior.

La cantidad de variables obtenidas es información valiosa para la caracterización de la población de los estudiantes universitarios.

8.2. Sobre los modelos

Los modelos escogidos se pensaron como contraste de los metodos supervisados versus no supervisados en el marco de trabajo desde el machine learning.

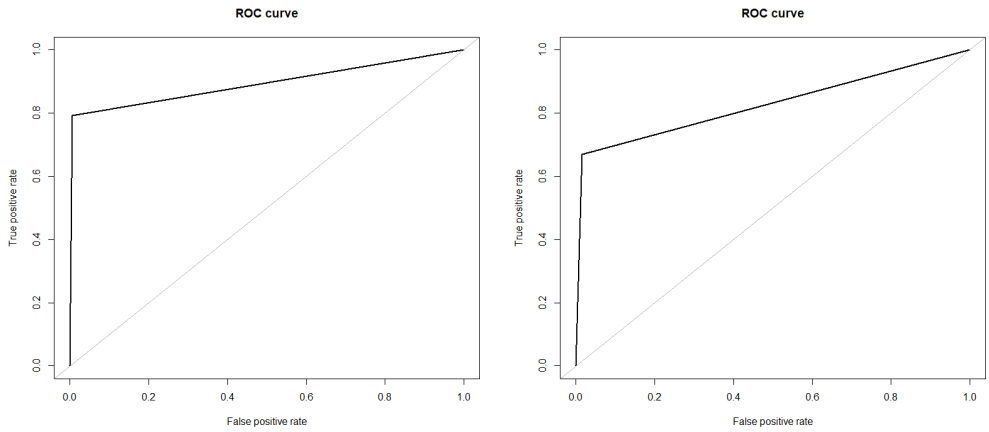


Figura 9: Comparación de las curvas ROC para el modelo Random Forest antes y después de la remoción de variables redundantes.

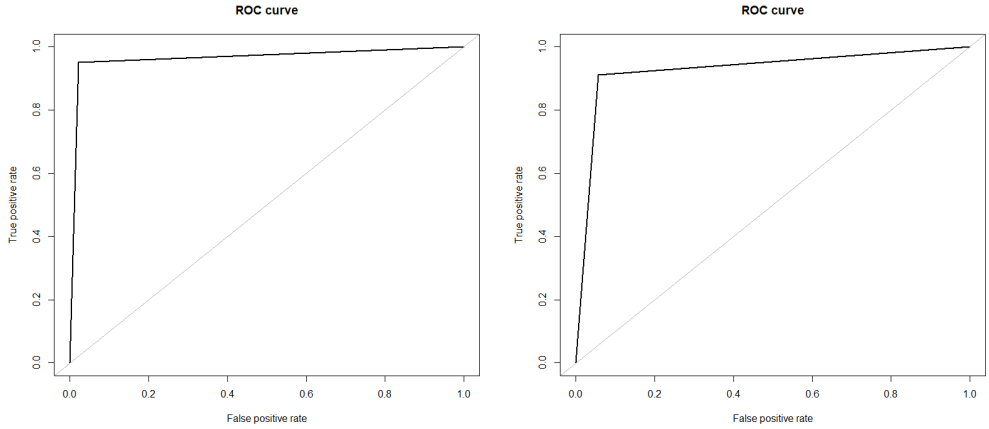


Figura 10: Comparación de las curvas ROC para el modelo Random Forest antes y después de la remoción de variables redundantes.

Para todos los modelos se eligió el mismo conjunto de variables a fin de mirar el comportamiento de las mismas respeccto a la predeicción de la condición de desersión de un estudiante, obteniendo resultados similares en todos los casos.

Según los resultados obtenidos, las variables más relevantes terminaron siendo las relacionadas con el promedio acumulado y por semestre dando a entender que el desempeño académico es lo que impulsa al aestudiante a tomar la decisión de abandonar a los estudios al final del semestre. Sin embargo, uan vez removidas estas variables de la base de datos para volver a correr los modelos

8.3. Trabajo futuro

Es de resaltar el hecho de que la información que se tiene podría aportar mejores resultados si se tuviera información concreta del momento exacto en que un estudiante deja de asistir a clases, datos que son muy complicados de obtener más aún cuando la metodología de registro de fallas no está completamente establecida por parte de la unniversidad. Ante esta situación, y sobre la base de que en este momento se encuentran en desarrollo tantas herramientas tecnológicas que permitirían dicha recolección de información, se sugiere que una vez implementadas estas estrategias, se realice nuevamente el ejercicio para determinar qué factores sociales llevan a un estudiante a tomar la decisión del retiro.

Referencias

[1] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984.

[2] Regresión logística simple y múltiple.