**Escuela de Estudios de Postgrado**

**Facultad de Ingeniería**

**Universidad San Carlos de Guatemala**

**Maestría en Ingeniería para la Industria con**

**Especialización en Ciencias de la Computación.**

**Minería de Datos**

**Ing. Ms. Kevin Lajpop**

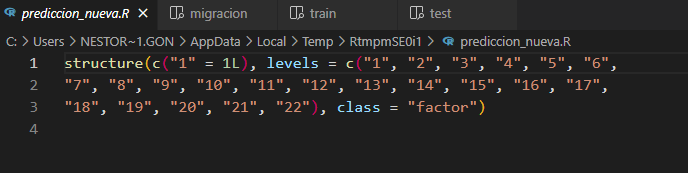
**Tarea Comodín**

<https://github.com/obduliogonzalez/tarea_comodin/tree/main>

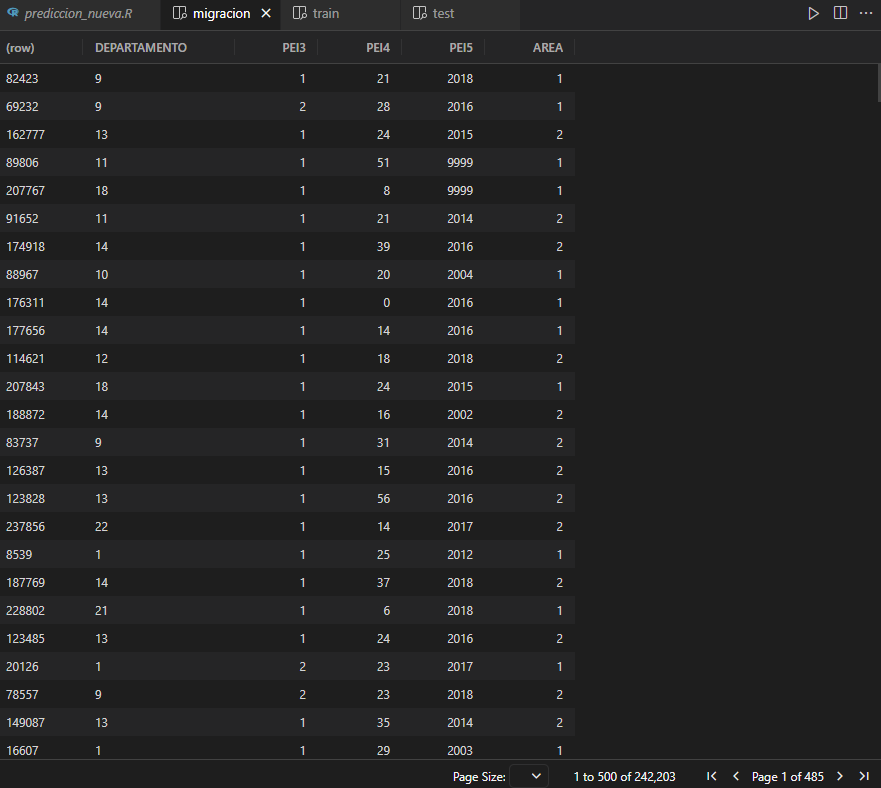
**Néstor Obdulio González López**

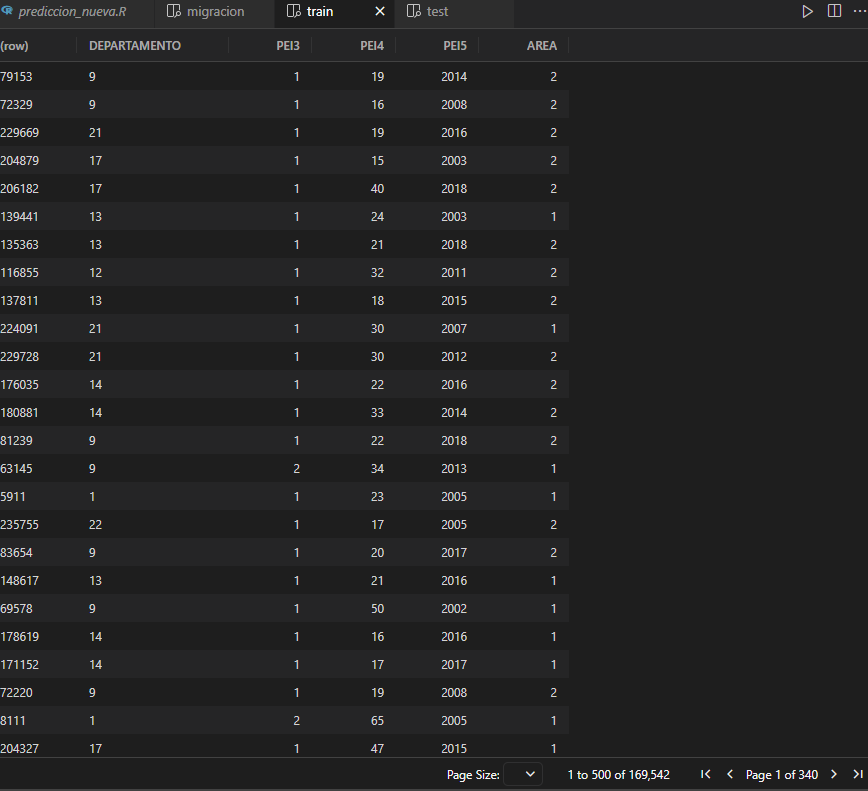
**Carné: 999013150**

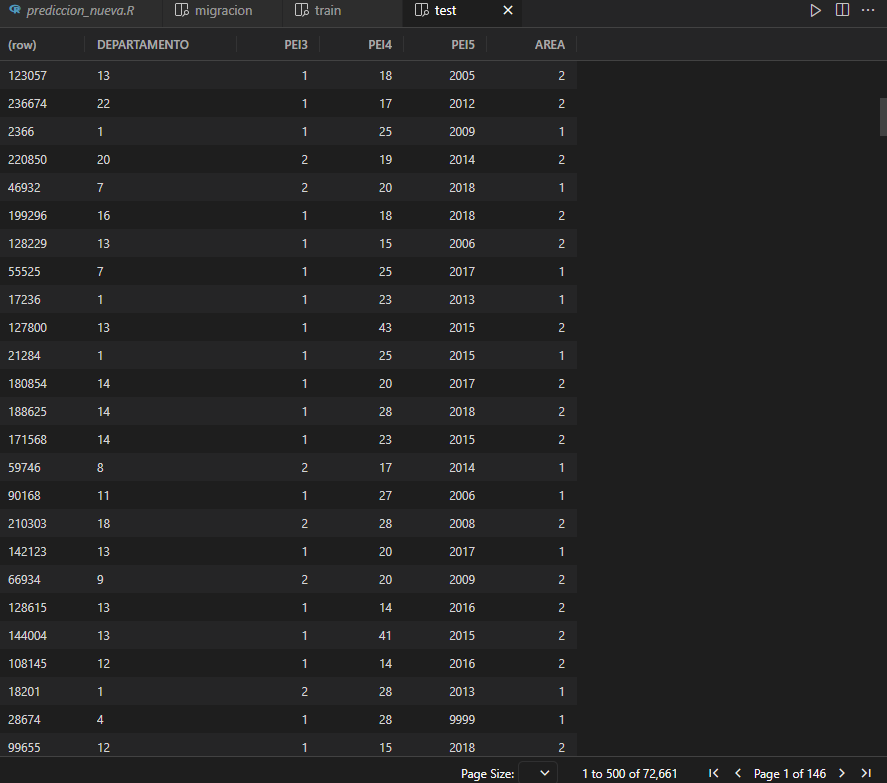
Tarea Comodin

Predicción Nueva

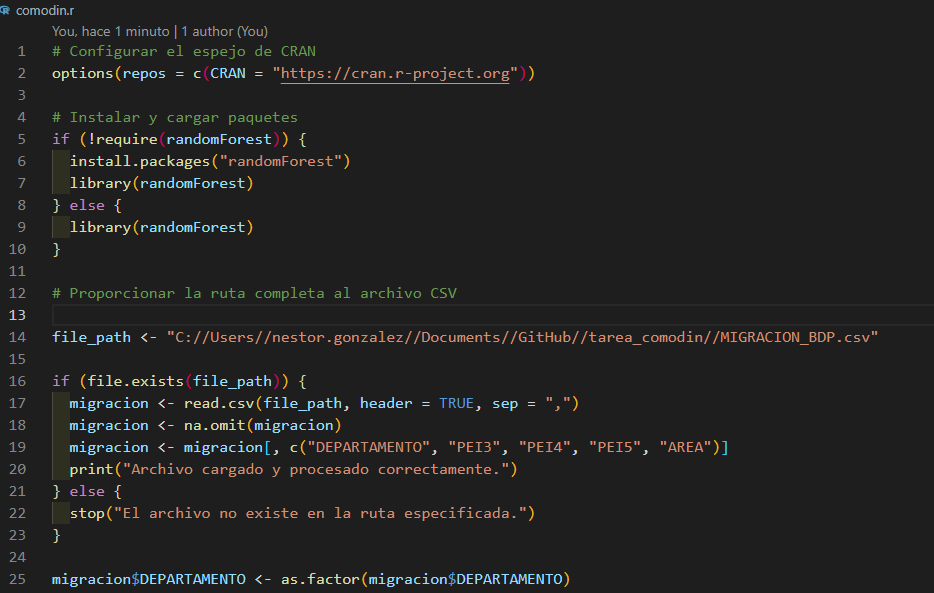
Migración



Train

Test

Ampliación



options(repos = c(CRAN = "https://cran.r-project.org"))

Aquí se define el repositorio de CRAN desde el cual se instalarán los paquetes. Especificas el enlace oficial.

Instalación y carga del paquete randomForest

if (!require(randomForest)) {

install.packages("randomForest")

library(randomForest)

} else {

library(randomForest)

}

Primero, verificar si el paquete randomForest ya está instalado usando require(randomForest).

Si no está instalado, descargar e instalar con install.packages("randomForest") y luego se carga con library(randomForest).

**Carga del archivo CSV**

Proporcionar la ruta del archivo CSV:

file\_path <- "C:/Users/xxxx.xxxx/GitHub/tarea\_comodin/MIGRACION\_BDP.csv"

Especifica la ruta completa donde se encuentra el archivo MIGRACION\_BDP.csv.

Comprobación de la existencia del archivo:

if (file.exists(file\_path)) {

migracion <- read.csv(file\_path, header = TRUE, sep = ",")

migracion <- na.omit(migracion)

migracion <- migracion[, c("DEPARTAMENTO", "PEI3", "PEI4", "PEI5", "AREA")]

print("Archivo cargado y procesado correctamente.")

} else {

stop("El archivo no existe en la ruta especificada.")

}

file.exists(file\_path): Verifica si el archivo existe en la ruta indicada.

Si el archivo existe:

Lo cargas en un data.frame llamado migracion utilizando read.csv. Usas header = TRUE para indicar que la primera fila contiene los nombres de las columnas y sep = "," para especificar que el separador es una coma.

Eliminar las filas con valores NA (faltantes) usando na.omit.

Filtrar el data.frame para quedar solo con las columnas "DEPARTAMENTO", "PEI3", "PEI4", "PEI5", "AREA".

Imprimir mensaje indicando que el archivo fue cargado correctamente.

Si el archivo no existe, detienes la ejecución con un mensaje de error stop.

Conversión de datos

migracion$DEPARTAMENTO <- as.factor(migracion$DEPARTAMENTO)

Paso a paso:

1. **Semilla para reproducibilidad y desordenar datos:**

set.seed(100) # Semilla para reproducibilidad

migracion <- migracion[sample(1:nrow(migracion)), ] # Desordenar datos

set.seed(100): Establece una semilla para garantizar que los resultados sean reproducibles cada vez que se corra el código.

sample(1:nrow(migracion)): Reordena aleatoriamente los índices de las filas del data.frame.

El conjunto migracion queda desordenado.

1. **División del conjunto de datos en entrenamiento y prueba:**

index <- sample(1:nrow(migracion), 0.7 \* nrow(migracion)) # 70% de los datos para entrenamiento

train <- migracion[index, ] # Datos de entrenamiento

test <- migracion[-index, ] # Datos de prueba

sample(1:nrow(migracion), 0.7 \* nrow(migracion)): Selecciona aleatoriamente el 70% de las filas como índice para el conjunto de entrenamiento.

train: Contiene las filas seleccionadas para el entrenamiento.

test: Contiene el 30% restante para evaluar el modelo.

1. **Entrenamiento del modelo Random Forest:**

bosque <- randomForest(DEPARTAMENTO ~ PEI3 + PEI4 + PEI5 + AREA, data = train, ntree = 100, mtry = 4)

randomForest(): Se utiliza para entrenar un modelo de bosque aleatorio.

Fórmula (DEPARTAMENTO ~ PEI3 + PEI4 + PEI5 + AREA): DEPARTAMENTO es la variable dependiente, mientras que PEI3, PEI4, PEI5 y AREA son las variables independientes.

data = train: Los datos utilizados para entrenar el modelo son el conjunto de entrenamiento.

ntree = 100: El modelo utilizará 100 árboles para el bosque.

mtry = 4: Especifica que se considerarán 4 variables en cada división.

1. **Predicción en el conjunto de prueba:**

predicciones <- predict(bosque, test)

predict(): Usa el modelo entrenado para predecir DEPARTAMENTO en el conjunto de prueba test.

1. **Predicción en un nuevo dato:**

dato\_nuevo <- data.frame(PEI3 = 2, PEI4 = 40, PEI5 = 2000, AREA = 2)

prediccion\_nueva <- predict(bosque, dato\_nuevo)

data.frame(): Crea un nuevo conjunto de datos con los valores especificados (PEI3 = 2, PEI4 = 40, PEI5 = 2000, AREA = 2).

predict(bosque, dato\_nuevo): Predice el valor de DEPARTAMENTO para este nuevo dato utilizando el modelo entrenado.

1. **Semilla para reproducibilidad y desordenar datos:**

set.seed(100) # Semilla para reproducibilidad

migracion <- migracion[sample(1:nrow(migracion)), ] # Desordenar datos

set.seed(100): Establece una semilla para garantizar que los resultados sean reproducibles cada vez que corras el código.

sample(1:nrow(migracion)): Reordena aleatoriamente los índices de las filas del data.frame.

El conjunto migracion queda desordenado.

1. **División del conjunto de datos en entrenamiento y prueba:**

index <- sample(1:nrow(migracion), 0.7 \* nrow(migracion)) # 70% de los datos para entrenamiento

train <- migracion[index, ] # Datos de entrenamiento

test <- migracion[-index, ] # Datos de prueba

sample(1:nrow(migracion), 0.7 \* nrow(migracion)): Selecciona aleatoriamente el 70% de las filas como índice para el conjunto de entrenamiento.

train: Contiene las filas seleccionadas para el entrenamiento.

test: Contiene el 30% restante para evaluar el modelo.

1. **Entrenamiento del modelo Random Forest:**

bosque <- randomForest(DEPARTAMENTO ~ PEI3 + PEI4 + PEI5 + AREA, data = train, ntree = 100, mtry = 4)

randomForest(): Se utiliza para entrenar un modelo de bosque aleatorio.

Fórmula (DEPARTAMENTO ~ PEI3 + PEI4 + PEI5 + AREA): DEPARTAMENTO es la variable dependiente, mientras que PEI3, PEI4, PEI5 y AREA son las variables independientes.

data = train: Los datos utilizados para entrenar el modelo son el conjunto de entrenamiento.

ntree = 100: El modelo utilizará 100 árboles para el bosque.

mtry = 4: Especifica que se considerarán 4 variables en cada división.

1. **Predicción en el conjunto de prueba:**

predicciones <- predict(bosque, test)

predict(): Usa el modelo entrenado para predecir DEPARTAMENTO en el conjunto de prueba test.

1. **Predicción en un nuevo dato:**

dato\_nuevo <- data.frame(PEI3 = 2, PEI4 = 40, PEI5 = 2000, AREA = 2)

prediccion\_nueva <- predict(bosque, dato\_nuevo)

data.frame(): Crea un nuevo conjunto de datos con los valores especificados (PEI3 = 2, PEI4 = 40, PEI5 = 2000, AREA = 2).

predict(bosque, dato\_nuevo): Predice el valor de DEPARTAMENTO para este nuevo dato utilizando el modelo entrenado.

1. **Mostrar resultados:**

# Mostrar las predicciones

View(predicciones)

View(prediccion\_nueva)

# Mostrar las tablas de datos

View(migracion)

View(train)

View(test)

View(): Abre una vista tabular en RStudio para inspeccionar:

predicciones: Resultados de las predicciones para el conjunto de prueba.

prediccion\_nueva: Resultado de la predicción para el nuevo dato.

migracion, train, test: Los conjuntos de datos principales, de entrenamiento y de prueba.

