APFEM(Análisis del Paro, en Función de Elementos Macroeconómicos)



Miembros del grupo:

* Daniel Nasim Santos Ouafki
* David Prats Ulloa
* Jin Wang Xu
* Juan Antonio Escobar de los Ángeles

**Índice**

[**Introducción**](#_aoj0zjp1a1ni) **3**

[**Metodología y Modelo del Datawarehouse**](#_dv9tmkybyc92) **3**

[**Metodología**](#_i5aru84bbhhm) **3**

[**Datasets y licencias**](#_6uiir230rg1o) **4**

[**Estructura Datawarehouse**](#_693kpyjs8w07) **5**

[**Procesos ETL**](#_vil62lptt0jf) **5**

[**Desarrollo**](#_varbctdgy5) **6**

[Modelización 1ª Iteración](#_8z19rrxivd7e) 6

[Modelización 2ª Iteración](#_f85vv74plmbg) 9

[Modelización](#_960gvxw4diaj) 9

[Modelización 3ª Iteración](#_uyqkevsxb28k) 11

[Modelo Principal a Nivel Nacional](#_keo7mv4nk3sd) 11

[Modelo principal pero por Comunidades.](#_dbd7s4lmf2br) 14

[Modelo de regresión con Criminalidad como variable objetivo](#_6bl3p05iv6kn) 16

[Modelo con árboles de decisión](#_rc3i3qg7usw4) 17

[Modelización 4ª Iteración](#_rhibjichbkex) 18

[Modelo con Árboles de Decisión](#_d0uixbj8ut01) 18

[Extra](#_w31ea1fi4ex3) 19

[**Evaluación y resultados**](#_qfxm57jpfqz4) **20**

[Modelo Principal](#_jwq41znte4ro) 20

[Error Medio Teórico](#_vibp1n6k2965) 20

[Validación Cruzada Binaria](#_qc1i55b0jnin) 20

[Árboles de Decisión](#_6i8mxbbw4nut) 21

[Validación simple](#_foex9qatdwm9) 21

[**Conclusiones**](#_bhvsxbe688kb) **22**

[**Bibliografía**](#_3navyxrz1yyp) **23**

[**Material Usado**](#_vfmuhm8sdhen) **23**

# Introducción

**Título del Proyecto**

APFEM (Análisis del Paro, en Función de Elementos Macroeconómicos).

**Dominio del Proyecto**

Paro, causas económicas y factores agravantes, usando datos sobre el paro, turismo, IPC, PIB, precio de la vivienda y criminalidad; a distintos niveles temporales (meses/trimestres) y regionales (comunidades y ciudades autónomas).

**Objetivos del Proyecto (Hipótesis de Trabajo)**

Estudiar las causas macroeconómicas que afectan al paro para así poder hacer una predicción estimada del número de parados en función de unas variables concretas.

Para ello tenemos que unificar todos los datos y clasificarlos de la mejor manera posible para simplificar su posterior análisis. Tras realizar el estudio se podrá decidir qué factores pueden afectar a la economía y el paro de España lo que podría ayudar en la toma de decisiones a nivel político y social de los aspectos a controlar.

# Metodología y Modelo del Datawarehouse

# Metodología

Usamos KDD para la extracción de datos escogiendo cada una de las muestras obtenidas mediante las fuentes previamente seleccionadas, que son lo suficientemente pequeñas para que el procesado sea lo suficientemente rápido, recogiendolas en datasets con el mismo formato. Posteriormente hemos analizado los datos de los archivos .csv y .json obtenidos para elegir cómo vamos a organizar el Datawarehouse usando el proceso ETL (Extracción, Carga y Transformación). Los procesos se han realizado de la siguiente manera:

**Limpieza de datos:** la limpieza de datos se define como la eliminación de datos ruidosos e irrelevantes de la recopilación.

* Limpieza en caso de falta de valores.
* Limpieza de datos ruidosos, donde los ruidos son comentarios sin el formato de los datos.

**Integración de datos**: la integración de datos se define como datos heterogéneos de múltiples fuentes combinadas en una fuente común (DataWarehouse).

* Integración de datos usando el proceso ETL (Extracción-Carga-Transformación).

**Selección de datos**: La selección de datos se define como el proceso en el que los datos relevantes para el análisis se deciden; inicialmente de forma intuitiva y posteriormente mediante la comparación de los diferentes resultados tras usar modelos de árboles de decisión o regresión.

**Transformación de datos**: La transformación de datos se define como el proceso de generación de código necesario para unificar los distintos formatos en uno solo para poder guardarlos en el datawarehouse.

**Minería de datos:** la minería de datos se define como técnicas inteligentes que se aplican para extraer patrones potencialmente útiles. En nuestro caso, el uso de modelos de clasificación.

**Representación del conocimiento:** la representación del conocimiento se define como una técnica que utiliza herramientas de visualización para representar los resultados de la minería de datos.

* Generar informes
* Generar tablas.

Todo ello mediante librerías de R.

# Datasets y licencias

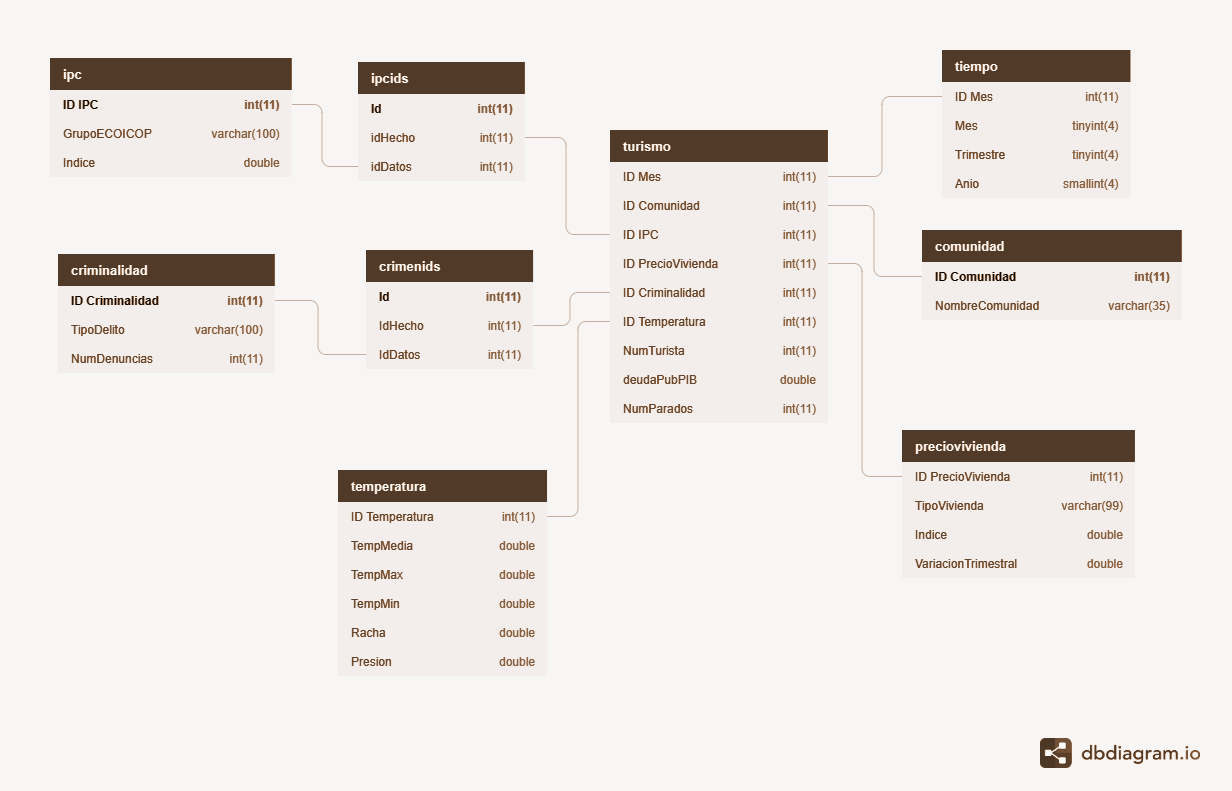
Los datasets seleccionados estaban en formato .csv y .json. En general, salvo el dataset de números de turistas, tuvimos una gran cantidad de datos, abarcando años desde 1995 al 2020, pero de uso útil desde el 2005 para la prueba de los modelos.

Todos los datasets utilizados disponen de una licencia de uso Open Data en las que el requisito mínimo es nombrar a sus propietarios y respetar ciertos criterios de uso. Hemos extraído datos de:

* **Temperatura**, obtenida de [**AEMET**](http://www.aemet.es/es/nota_legal) a través su API con script Python.
* **Criminalidad** del [**Ministerio de Interior**](http://www.interior.gob.es/web/interior/aviso-legal)**.**
* **Paro** de [**EPDATA**](https://www.ine.es/ss/Satellite?L=0&c=Page&cid=1254735849170&p=1254735849170&pagename=Ayuda%2FINELayout)**(Datos obtenidos del INE).**
* **Precio Vivienda, Deuda pública, IPC y Turismo** obtenido del [**INE**](https://www.ine.es/ss/Satellite?L=0&c=Page&cid=1254735849170&p=1254735849170&pagename=Ayuda%2FINELayout)**.**

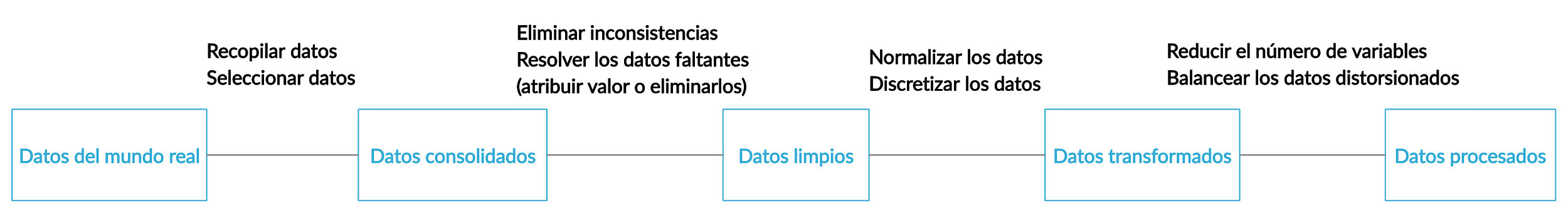
En los enlaces adjuntos se puede ver en detalle los términos de uso de los datos utilizados en nuestro proyecto.

# Estructura Datawarehouse

Nuestro datawarehouse unifica todos los datos ya limpiados y preprocesados en un modelo multidimensional con una tabla de hechos y varias tablas de dimensiones que son las del tiempo,comunidad, precio vivienda, temperatura, criminalidad e IPC, que representan las puntas de la estrella y una tabla de hechos en el centro, que contiene datos de número de turistas, deuda pública y número de parados, además de los ids de las otras tablas para poder relacionarse con ellas. Las claves primarias en las tablas de las dimensiones son los ids únicos y en la tabla de hechos la clave principal se compone de las claves primarias de las tablas de dimensiones.

# Procesos ETL

El proceso de extracción se ha llevado a cabo utilizando python recorriendo las fuentes de datos y extrayendo los datos llevando a cabo la limpieza de los datos erróneos y los vacíos, de archivos con extensión json y csv, unificandolos a través de una transformación en archivos .csv para posteriormente hacer el proceso de carga al datawarehouse con un programa en R que procesa los datos y los guarda.



# Desarrollo

## Modelización 1ª Iteración

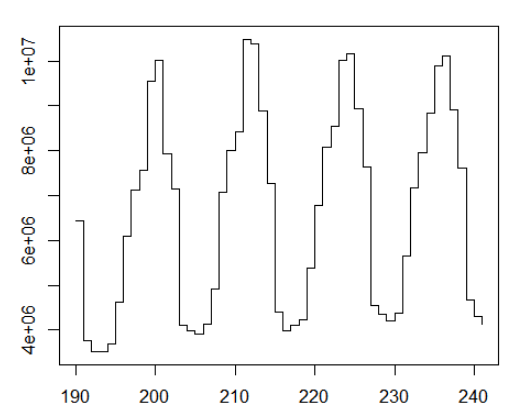
Durante la 1º Iteración se implementaron los procesos ETL, ya explicados anteriormente, necesarios para la limpieza y la carga de los datasets en el Datawarehouse. Una vez terminada la carga del Datawarehouse se procedió a la prueba de un modelo simple, en este caso, una predicción del turismo a lo largo del tiempo.

**Modelo implementado**

Se implementó una regresión lineal con la función **lm** que proporciona R.

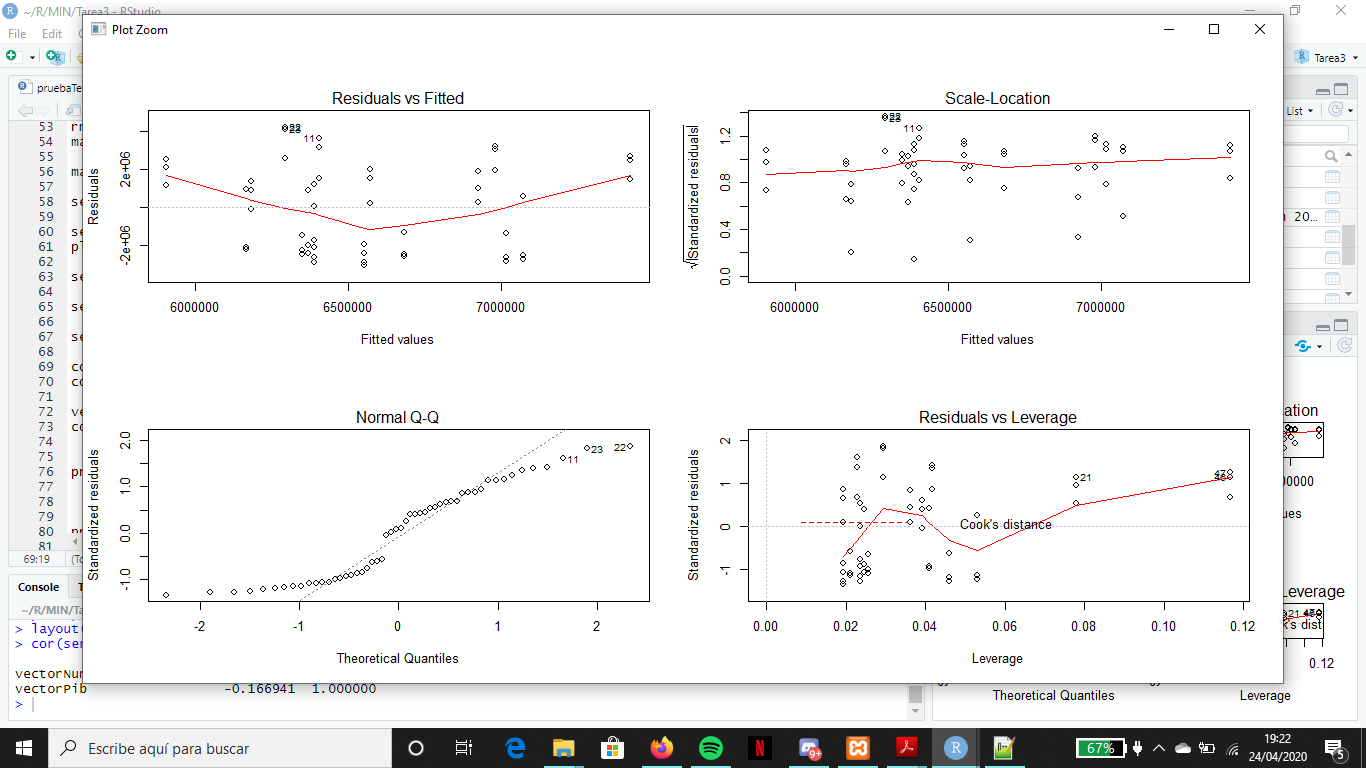
|  |
| --- |
| Coefficients:  Estimate Std. Error t value Pr(>|t|) (Intercept) 5940351 671577 8.845 8.43e-12 t$Mes 94312 88923 1.061 0.294 |
|  |

Viendo el valor del coeficiente, pudimos observar que el modelo no era efectivo de esta manera.

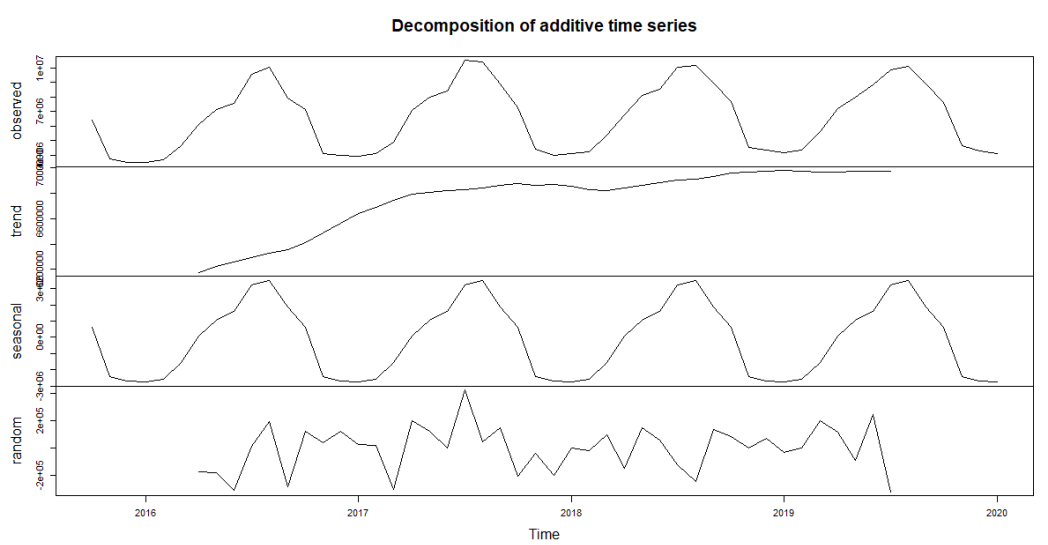


Viendo esta gráfica se puede observar cómo fluctúa el turismo (eje Y), según el mes (eje X). En este último eje la unidad de medida es el id del mes. Podemos ver que turismo sigue una función periódica que coincide con el verano. Teniendo esto en cuenta decidimos usar otras funciones de R.

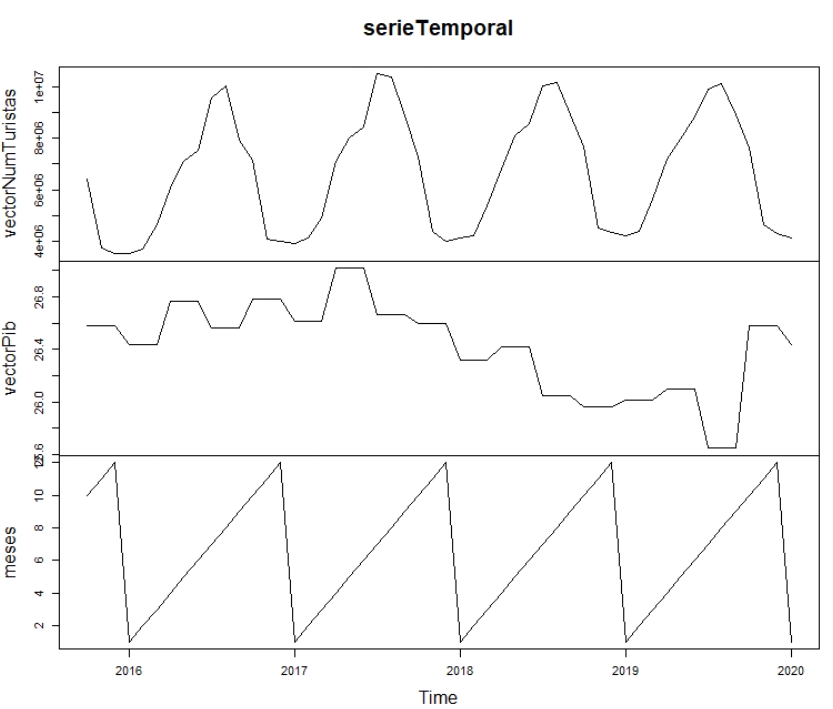
Durante las pruebas del modelo se obtuvieron los siguientes coeficientes a la hora de calcular el error en la operación.



Tras usar el modelos de predicción anterior, decidimos probar las funciones que ofrecen las series temporales en R. Con dichas funciones podemos observar la tendencia del turismo y su periodicidad por estaciones.



A continuación probamos a correlacionar en las series temporales el PIB y los meses.

Después de ver este tipo de resultados, llegamos a la conclusión de que relacionar el turismo con el PIB no tenía sentido, puesto que el turismo varía según una época del año y el PIB no depende de eso. Es fácil verlo en la gráfica de arriba en la que el turismo fluctúa debido a la llegada de turistas en verano, y baja en invierno. Por esta causa, descartamos el uso del turismo en las siguientes iteraciones.

## Modelización 2ª Iteración

Durante las 2º Iteración se corrigió el script de carga del Datawarehouse para que abarcase un periodo temporal más extenso y así para poder mejorar las pruebas de modelos.

Además se cambió totalmente el objetivo principal aplicado en la 1º iteración, llevando a cabo una regresión lineal para ver la relación entre el IPC y el número de parados de 2005 a 2020.

Este cambio se debe a la escasez de datos presentes de turismo, en este caso solo se poseen datos desde el 2015, teniendo por lo tanto apenas 5 años de datos sobre turismo con los que hacer pruebas. Con este cambio conseguimos un un periodo temporal de unos 15 años con los que podemos realizar modelos más robustos.

### **Modelización**

Como ya se ha nombrado anteriormente se realizó una regresión lineal con el objetivo de predecir el IPC a partir del número de parados.

En **vectorIPC** y **vectorParados** se guardan los datos para la regresión lineal, que juntaremos en una matriz para tenerlos juntos.

|  |
| --- |
| matriz <- cbind(vectorIPC, vectorParados) |

Se aplica una semilla para hacer las pruebas: **set.seed(1)**

Para los datos de prueba se ha escogido una muestra del 90%(**trainData**) y el 10% para test(**testData**)

En esta imagen se puede observar la correlación de la serie temporal que se ha construido previamente, compuesta por IPC y los parados.

|  |
| --- |
| > cor(serieTemp)  vectorIPC vectorParados vectorIPC 1.0000000 0.6592044 vectorParados 0.6592044 1.0000000 |
|  |

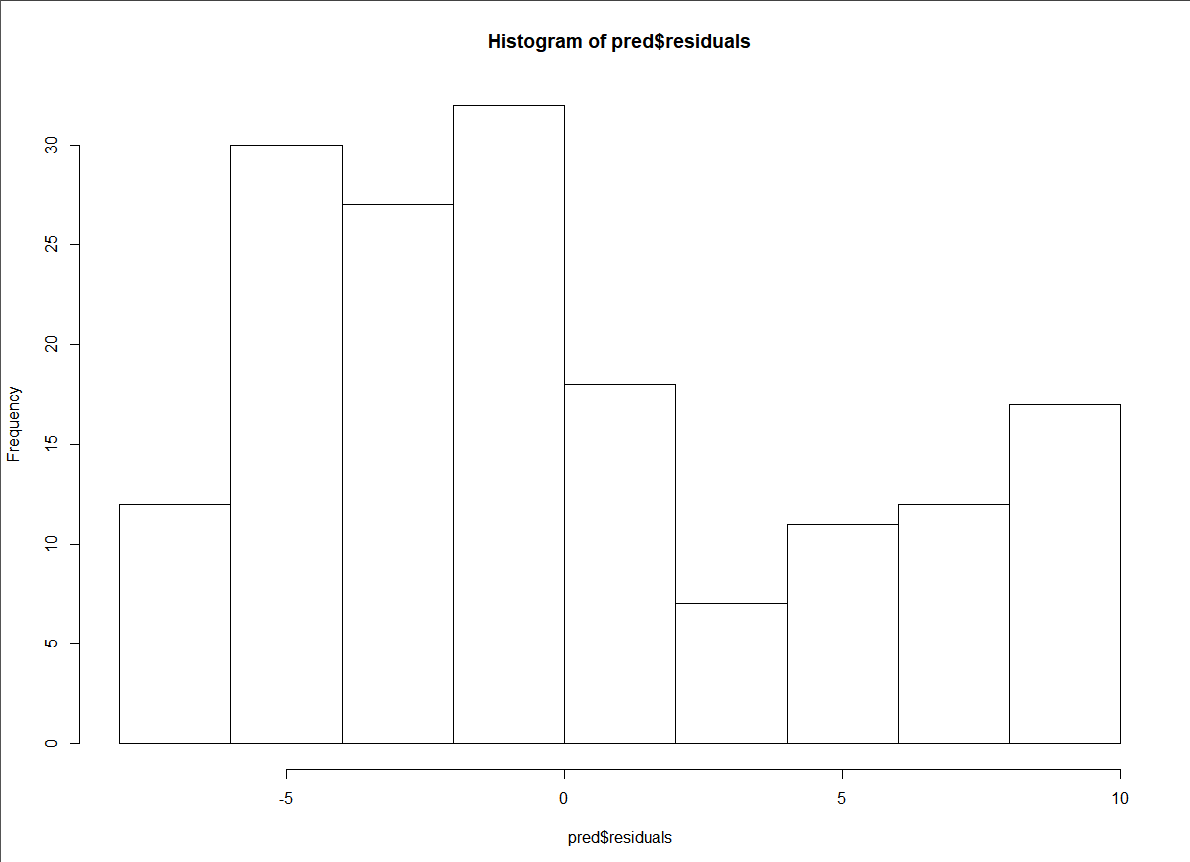
Este es el resultado de la aplicación de la regresión mediante la llamada a **lm**. Donde se puede observar la fórmula aplicada

|  |
| --- |
| pred <- lm(formula = ipcTrain ~ paradosTrain) |

|  |
| --- |
| Call: lm(formula = ipcTrain ~ paradosTrain)  Residuals:  Min 1Q Median 3Q Max  -7.611 -4.040 -1.109 3.261 9.842  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 8.045e+01 1.495e+00 53.80 < 2e-16 \*\*\* paradosTrain 4.265-06 4.071e-07 10.48 < 2e-16 \*\*\*  --- Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1 Residual standard error: 4.875 on 164 degrees of freedom Multiple R-squared: 0.4009, Adjusted R-squared: 0.3972  F-statistic: 109.7 on 1 and 164 DF, p-value: < 2.2e-16 |

Se puede ver que el coeficiente R cuadrado es igual a 0.4 por lo tanto se concluyó que el modelo no era adecuado.

Generando el histograma de los valores residuales podemos concluir la ineficiencia del modelo.



Se puede observar que los residuos no siguen ninguna distribución, lo que reafirma que el modelo no es válido para la predicción del IPC.

Después de esta iteración, al ver que no tenía sentido intentar predecir el IPC por la naturaleza de la variable, decidimos que en las siguientes iteraciones cambiaríamos de variable dependiente.

## Modelización 3ª Iteración

La tercera iteración, en comparación a la segunda, es muy distinta ya que obtenemos mejores resultados y con un objetivo más práctico.

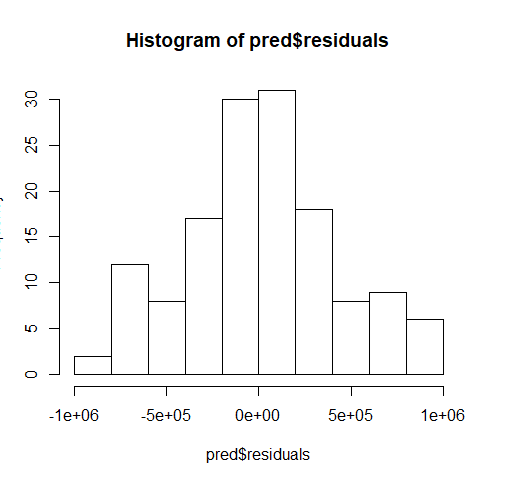
Hemos implementado varios modelos que se explica a continuación:

### Modelo Principal a Nivel Nacional

El modelo principal de la tercera iteración está basada en una regresión con el número de parados como variable objetivo, y 3 nuevas variables independientes: IPC, deuda pública y el precio de las viviendas.

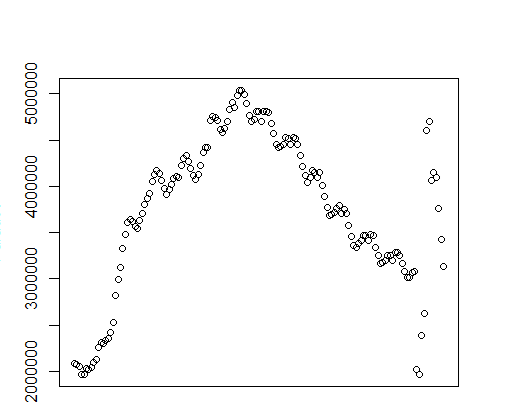
|  |
| --- |
| Call: lm(formula = paradosTrain ~ ipcTrain + deudaTrain + precioViviendaTrain)   Residuals:  Min 1Q Median 3Q Max -919344 -260418 6595 220070 994931   Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 382394 2351013 0.163 0.871  ipcTrain 152875 24951 6.127 8.93e-09 \*\*\* deudaTrain -211636 18343 -11.538 < 2e-16 \*\*\* precioViviendaTrain -66837 4284 -15.602 < 2e-16 \*\*\* --- Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1   Residual standard error: 417700 on 137 degrees of freedom Multiple R-squared: 0.7402, Adjusted R-squared: 0.7345 F-statistic: 130.1 on 3 and 137 DF, p-value: < 2.2e-16 |

Fijándonos en el **coeficiente de determinación (R2)** observamos que es de 0.74 (0.73 ajustado) como dicho valor tiende a 1 podemos decir que es un buen ajuste para el modelo.

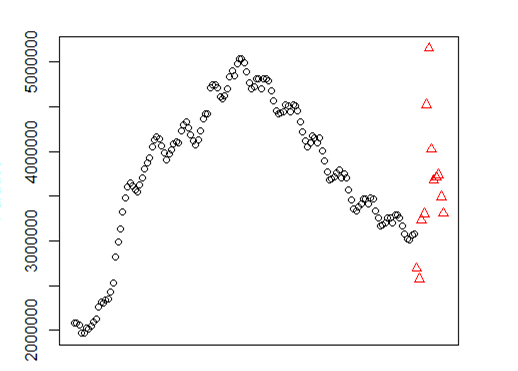
****

En el histograma podemos apreciar que los residuos siguen una **Distribución Normal**, la mayoría de los errores son cercanos a cero.

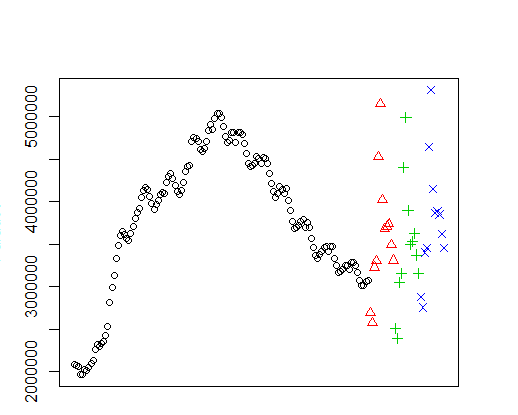
**Predicciones obtenidas con el modelo**

****

**Evolución Real del Número de Parados (datos de entrenamiento y test)**

****

**Predicción del Número de Parados a partir de los datos de entrenamiento**

****

**Predicción del Número de Parados a partir de los datos de entrenamiento, con datos ajustados, máximos y mínimos**

**Otros Modelos**

En la 3º iteración aparte del modelo principal ya comentado, exploramos más modelos con distintos objetivos.

### Modelo principal pero por Comunidades.

Cambiamos la hipótesis principal a **cómo afectaría el número de parados** en relación del ipc, deuda pública, precio vivienda y **la comunidad autónoma exceptuando Ceuta, entre los años 2007 y 2019.**

De esta forma aumentamos el número de datos y obtenemos unos resultados más precisos.

Para ello, hemos planteado la fórmula con **4 vectores como variables independientes**: IPC, deuda pública, precio vivienda y la comunidad autónoma, y la **variable dependiente el número de parados**.

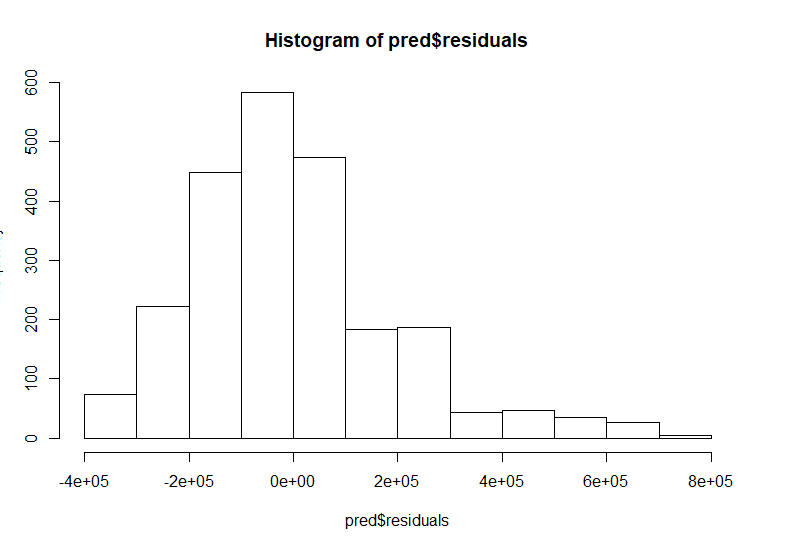
La división para el entrenamiento hemos optado a dividirlo 90 % entrenamiento y 10 % testeo.

|  |
| --- |
| **ind <- sample(2, nrow(matriz), replace=TRUE, prob=c(0.9, 0.1))** |

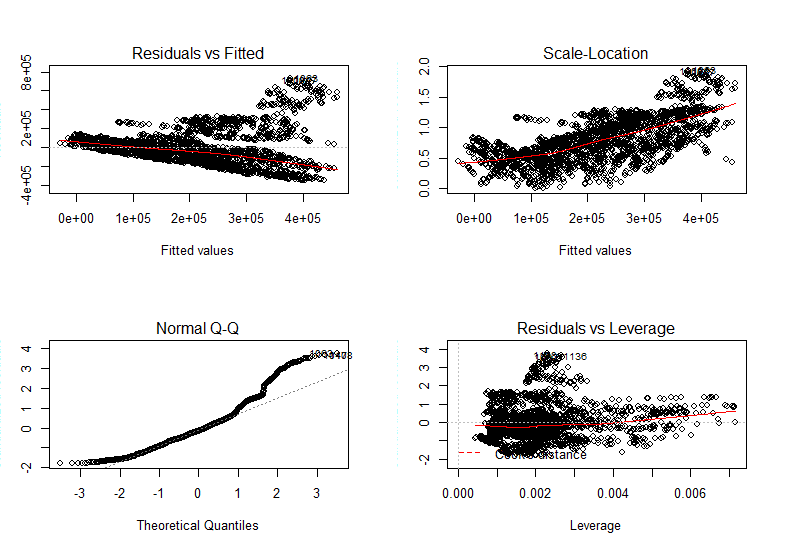
Hemos utilizado la función lm de la librería stats y hemos obtenido los siguientes resultados

|  |
| --- |
| Call: lm(formula = paradosTrain ~ ipcTrain + paradosTrain +   precioVivienda + comunidadTrain)   Residuals:  Min 1Q Median 3Q Max  -352323 -122779 -29537 87110 734214   Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 2074979.3 176176.9 11.778 < 2e-16 \*\*\* trainData[, 1] -14277.3 1561.3 -9.145 < 2e-16 \*\*\* trainData[, 3] 4787.8 631.0 7.587 4.7e-14 \*\*\* trainData[, 4] -2940.8 316.9 -9.279 < 2e-16 \*\*\* trainData[, 5] -19808.1 830.3 -23.857 < 2e-16 \*\*\*  Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1 Residual standard error: 194900 on 2323 degrees of freedom Multiple R-squared: 0.245, Adjusted R-squared: 0.2437  F-statistic: 188.4 on 4 and 2323 DF, p-value: < 2.2e-16 |

Objetivamente en comparación al modelo principal, el **coeficiente de determinación (R2)** es de 0.24, mucho más inferior y llegamos a la conclusión de que los valores macroeconómicos a nivel de comunidades, no es viable.



En este histograma podemos apreciar que los residuos siguen una Distribución de Poisson, la mayoría de los errores son cercanos a cero.



Podemos observar con la normal, que al trazar los dos conjuntos de cuantiles, forman una recta exceptuando al final con un desvío, tendiendo a una distribución de Poisson.

### Modelo de regresión con Criminalidad como variable objetivo

En este nuevo modelo, cambiamos la hipótesis para abarcar nuevas posibilidades.

En este caso, aplicamos con un modelo de regresión lineal, la relación de la criminalidad frente al número de parados, deuda pública, precio vivienda e ipc.

La variable dependiente es un vector de los datos de criminalidad.

4 variables independientes: Número de parados, IPC, deuda Pública y precio vivienda.

**lm(formula = criminalidadTrain ~ paradosTrain + ipcTrain + deudaTrain + precioViviendaTrain)**

|  |
| --- |
| Call: lm(formula = criminalidadTrain ~ paradosTrain + ipcTrain + deudaTrain + precioViviendaTrain) Residuals:  Min 1Q Median 3Q Max  -1199501 -627934 -61612 563244 1393984   Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) -6.315e+06 2.966e+07 -0.213 0.83247  paradosTrain -3.208e+00 1.255e+00 -2.557 0.01457 \*  ipcTrain 6.442e+05 1.644e+05 3.918 0.00035 \*\*\* deudaTrain -4.298e+05 1.737e+05 -2.475 0.01779 \*  precionViviendaTrain -3.293e+05 1.175e+05 -2.803 0.00784 \*\*  --- Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  Residual standard error: 774700 on 39 degrees of freedom Multiple R-squared: 0.4923, Adjusted R-squared: 0.4402  F-statistic: 9.453 on 4 and 39 DF, p-value: 1.928e-05 |

Observando los valores obtenidos, es un modelo con resultados mediocres basándonos en la R cuadrado, debería tender a 1, y apenas supera el 0,5. Es un modelo que no cumple los requisitos suficientes para considerarlo una iteración del todo válida.

### Modelo con árboles de decisión

Hemos probado por último un modelo con árboles de decisión alternativa al modelo de regresión lineal.

Para realizar pruebas con el modelo, hemos utilizado la función rpart() de la librería “rpart”.

Discretizamos en intervalos de 5 niveles.

Observamos los resultados obtenidos con el printcp().

|  |
| --- |
| CP nsplit rel error xerror xstd 1 0.246835 0 1.00000 1.00000 0.029688 2 0.097890 1 0.75316 0.75316 0.027116 3 0.037975 4 0.45949 0.47848 0.022752 4 0.036709 5 0.42152 0.42911 0.021734 5 0.034177 6 0.38481 0.40127 0.021119 6 0.031646 8 0.31646 0.37215 0.020441 7 0.023418 9 0.28481 0.32658 0.019297 8 0.021519 11 0.23797 0.26076 0.017434 9 0.017722 12 0.21646 0.24557 0.016961 10 0.015190 14 0.18101 0.20633 0.015646 11 0.010000 17 0.13544 0.15316 0.013596 |

Como podemos observar, tras hacer la función printcp(), se genera 11 niveles en el árbol de decisión. La tabla se compone principalmente de tres tipos de errores, el error relativo, xerror y xstd, siendo la xerror la más importante ya que es la necesaria para saber en qué nivel del árbol empieza a empeorar.

La xerror es el error de la validación cruzada creada. Se supone que cada vez que aumenta el nivel en el árbol, el valor xerror disminuye ya que a medida que bajamos de nivel, las decisiones se vuelven más precisas. Tras observar que en nuestro caso disminuye, concluimos que no sirve como modelo para la predicción

## Modelización 4ª Iteración

En esta última iteración, se realizaron actividades similares a la anterior con el objetivo de mejorar las evaluaciones de los modelos. Por lo tanto, no se ha visto la necesidad de modificar los procesos **ETL** y se han refinado los modelos realizando algunas validaciones sobre los mismos.

En el caso del modelo principal de la 3º iteración, los resultados son bastantes buenos, por tanto no es necesario realizar una iteración extra.

Sin embargo, para el modelo secundario de árboles de decisión, realizamos una mejora notable en la 4º iteración pero no suficientes para sustituir el modelo principal.

## Modelo con Árboles de Decisión

Con este modelo nuestro objetivo fue clasificar el número de parados en función del IPC, la deuda pública, el precio de la vivienda y las comunidades y ver si existía relación entre ellos.

Para ello hemos discretizado la variable en 5 niveles utilizando 5 intervalos distintos cuya longitud se ha calculado en función del número de valores (223415), teniendo una amplitud cada intervalo de 44683, se han analizado los datos y hemos observado que del primer nivel, los que tienen menos parados, son más numerosos, como se ve en el análisis de frecuencias absolutas y relativas.

Después se han dividido los datos en dos conjuntos, los datos de prueba para probar el modelo (10%) y los datos de entrenamiento para crearlo (90%).

Después, se ha utilizado la librería rpart para crear un árbol de decisión cuya variable dependiente es el número de parados por niveles y cuyas variables independientes son el IPC,la deuda pública, el precio de la vivienda y las comunidades.

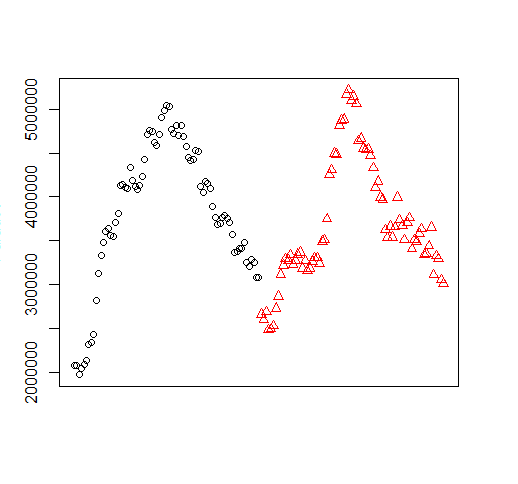
### 

### Extra

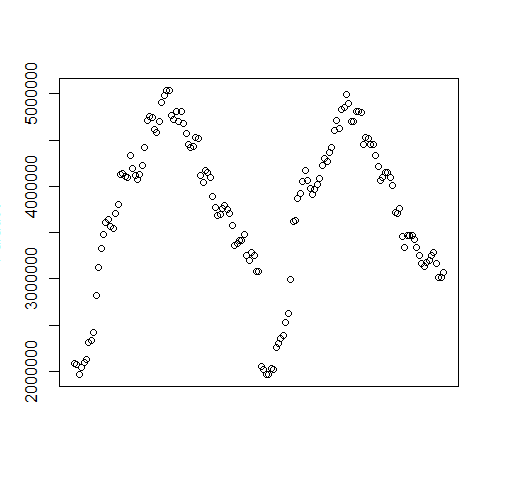
**Modelo con 50% de los datos**

|  |
| --- |
| Call: lm(formula = paradosTrain ~ ipcTrain + deudaTrain + precioViviendaTrain)  Residuals:  Min 1Q Median 3Q Max  -794994 -264141 -50458 248573 950804   Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 3637110 3251120 1.119 0.266976  ipcTrain 120542 34953 3.449 0.000945 \*\*\* deudaTrain -193474 26642 -7.262 0.000000000363 \*\*\* precioViviendaTrain -70399 6246 -11.271 < 0.0000000000000002 \*\*\* |

**Predicción con la primera mitad de los datos**

****

**Datos Reales**

****

# Evaluación y resultados

La evaluación tuvo lugar en nuestra última iteración, se hicieron dos valoraciones sobre el **modelo principal** presentando en la **última iteración** y sobre el modelo basado en **árboles de decisión.**

## Modelo Principal

### Error Medio Teórico

|  |
| --- |
| > #----Calculo del Error Cuadrático Medio Teórico de la Regresión > ECMaPriori <- sum((residuals(pred)^2))/length((residuals(pred))) >  > # Cálculo del error medio absoluto "teórico" o "a priori" de la regresión > EMAaPriori <- sqrt(ECMaPriori); EMAaPriori [1] 411724.4 >  > # Cálculo del error medio relativo (o porcentual) "teórico" o "a priori" de la regresión > EMRaPriori <- EMAaPriori / mean(paradosTrain); EMRaPriori [1] 0.108683 |

Se obtiene un **error medio teórico** del **10,8%.**

### Validación Cruzada Binaria

Primero se dividieron los datos en 2 partes iguales para entrenar

|  |
| --- |
| #Semilla y división para entrenamiento set.seed(1) ind <- sample(2, nrow(matriz), replace=TRUE, prob=c(0.5, 0.5)) head(ind, 10) trainData <- matriz[ind==1,] ; dim(trainData) testData <- matriz[ind==2,] ; dim(testData) |

Siguiendo los pasos necesarios para la validación:

1. Se entrena el modelo con los datos de A; se valida con los datos de B se calcula el error asociado.
2. Se entrena el modelo con los datos de B; se valida con los datos de A; se calcula el error asociado.
3. Se entrena el modelo con todos los datos (A+B); se valida con todos los datos (A+B); se calcula el error asociado.
4. El error del clasificador se obtiene como la media de los tres errores anteriores

El **error clasificador** es el siguiente; para este caso con un 11,49% de error media entre los distintos modelos

|  |
| --- |
| > errorFinalCruzBin = (EMRaPrioriA+EMRaPrioriB+EMRaPrioriF) / 3 > errorFinalCruzBin [1] 0.114918 |

## Árboles de Decisión

### Validación simple

|  |
| --- |
| trainPredRpart <- predict(arbol\_rpart, newdata = trainData, type = "class") |

|  |
| --- |
| > tablaPriori <- table(trainPredRpart, trainData$corte) ; tablaPriori   trainPredRpart nivel 1 nivel 2 nivel 3 nivel 4 nivel 5  nivel 1 1587 30 0 0 0  nivel 2 33 306 8 0 0  nivel 3 1 15 226 0 0  nivel 4 0 0 4 57 3  nivel 5 0 0 0 0 58 |

|  |
| --- |
| > sum(diag(tablaPriori))/sum(tablaPriori) [1] 0.959622 |

|  |
| --- |
| testPredRpart <- predict(arbol\_rpart, newdata = testData, type = "class") |

> tablaPosteriori <- table(testPredRpart, testData$corte) ; tablaPosteriori

|  |
| --- |
| testPredRpart nivel 1 nivel 2 nivel 3 nivel 4 nivel 5  nivel 1 186 5 0 0 0  nivel 2 3 34 3 0 0  nivel 3 1 4 22 0 0  nivel 4 0 0 1 3 1  nivel 5 0 0 0 0 10 |

|  |
| --- |
| > sum(testPredRpart == testData$corte) / length(testData$corte) \* 100 [1] 93.40659 > sum(diag(tablaPosteriori))/sum(tablaPosteriori) [1] 0.9340659 |

|  |
| --- |
| > precisionPorCategorias [1] 0.9789474 0.7906977 0.8461538 1.0000000 0.9090909 |

|  |
| --- |
| > precisionMedia <- mean(precisionPorCategorias); precisionMedia [1] 0.904978 |

Se puede apreciar que la precisión se acerca en todos los casos al 90%, pero para ello se ha dividido la variable dependiente en 5 niveles para poder crear el árbol de decisión.

# Conclusiones

Tras realizar las distintas evaluaciones surge una pregunta, dado los resultados obtenidos ¿Qué modelo resulta mejor para una predicción del paro, el modelo basado en regresión lineal o en árboles de decisión?

Para contestar a esta pregunta tenemos que tener en cuenta que los árboles de decisión se han implementado en base a una categorización del número de parados, creando así intervalos del mismo tamaño, por lo tanto, en la prueba del modelo por árboles de decisión obtenemos como resultado un intervalo con 44683 posibles valores, es decir, se pierde precisión al categorizar la variable **número de parados.**

En cambio, para el modelo de regresión no se ha discretizado la variable asociada al número de parados, eso sumando a un error de solo el 11% hace que este modelo sea la mejor opción para nuestro proyecto.

Este modelo es muy útil para organizaciones gubernamentales ya que podrían tomar medidas para actuar sobre posibles subidas en el número de parados y evadiendo crisis económicas y sociales.

De esta manera se proporcionaría un beneficio social y económico a todos los ciudadanos del país.

# Bibliografía

* <https://www.machinelearningplus.com/machine-learning/complete-introduction-linear-regression-r/>
* <http://www.sthda.com/english/articles/40-regression-analysis/166-predict-in-r-model-predictions-and-confidence-intervals/>
* <https://www.rdocumentation.org/packages/car/versions/3.0-7/topics/Predict>
* <https://cv4.ucm.es/moodle/pluginfile.php/6900487/mod_collaborate/intro/Tema%203.3.4%20-%20Evaluaci%C3%B3n.pdf>
* <https://cv4.ucm.es/moodle/pluginfile.php/6900280/mod_collaborate/intro/2%29%20Tema%203.3.3.1%20-%20%C3%81rboles%20de%20decisi%C3%B3n%20y%20bosques%20aleatorios.pdf>
* <https://cv4.ucm.es/moodle/pluginfile.php/6900245/mod_collaborate/intro/Tema%203.3.1%20%281%29%20-%20Importaci%C3%B3n%2C%20preprocesamiento%20y%20exportaci%C3%B3n%20de%20datos%20con%20R.pdf>
* <https://cv4.ucm.es/moodle/pluginfile.php/6900245/mod_collaborate/intro/Tema%203.3.1%20%284%29%20-%20Lectura%20de%20datos%20de%20formularios%20HTML%20con%20R.zip>
* <https://cv4.ucm.es/moodle/pluginfile.php/6725926/mod_folder/content/0/Tema%203.2%20-%20Metodolog%C3%ADas%20de%20desarrollo%20en%20MD%20%28KDD%2C%20SEMMA%2C%20CRISP-DM%29.pdf?forcedownload=1>

# Material Usado

* <https://stackoverflow.com/questions/22712409/putting-functions-into-a-data-frame>
* <https://stackoverflow.com/questions/1660124/how-to-sum-a-variable-by-group>
* <https://www.youtube.com/watch?v=CLKPK2JN1XE>
* <https://www.youtube.com/watch?v=ttkxy1bMkgQ>
* <https://www.youtube.com/watch?v=mgh4KdbYHv0>
* <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/predict.lm.html>
* <https://www.dummies.com/programming/r/how-to-predict-new-data-values-with-r/>
* <https://itelligent.es/es/10-ventajas-la-mineria-web/>
* <http://rtdibermatica.com/?tag=kdd>

|  |
| --- |
|  |