
# PREDICCIÓN

**Práctica 3:** Préstamos (mejorar)

V. Miguel Sempere Navarro

La presente práctica tiene como objetivo la estructuración, limpieza y creación de un modelo de regresión en base a un conjunto de datos obtenidos en el "Lending Club".

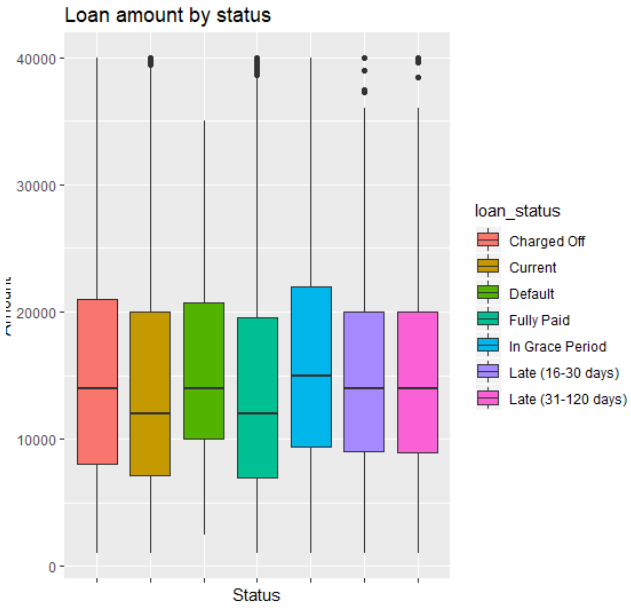
De las más de 100000 observaciones iniciales con 145 variables para cada observación se decide eliminar la información que no aporta valor y trabajar solo con aquellos datos que aporten información suficiente para predecir si un préstamo va a poder devolverse o no.

Tras eliminar la gran parte de las variables, utilizaremos para trabajar las siguientes:

- **Loan status**: Estado actual del préstamo. Nuestra variable dependiente.
- **Grade y Subgrade**: Grado y subgrado asignados al préstamo
- **Open Acc**: Número de líneas de crédito abiertas
- **Pub Rec**: Número de registros públicos derogatorios
- **DTI**: Ratio calculado para el prestatario usando el total de pagos mensuales
- **Delin 2 years**: Número a partir de 30 días de incidencias de morosidad.
- **Inq last 6 months**: Número de consultas en los últimos 6 meses
- **Emp Length**: Duración del empleo expresado en años.
- **Annual income**: Ingresos anuales
- **Home Ownership**: Estado de propiedad de la vivienda del prestatario
- **Purpose**: Propósito del préstamo
- **Addr State**: Estado proporcionado por el prestatario
- **Loan Amount**: Cantidad del préstamo
- **Int Rate**: Ratio de interés para el préstamo
- **Installment**: Pagos mensuales que debe el prestamista
- **Issue**: Mes en el que el préstamo fue concedido
- **Revol Bal**: Balance rotatorio de crédito total
- **Revol Util**: Cantidad de crédito que el prestatario está usando en base al total.

El desarrollo de un modelo de regresión nos permite relacionar una variable dependiente de carácter dicotómico con las demás variables. La idea es la construcción de un modelo de entrenamiento con un 70% de las observaciones y un modelo de test con un 30% que corrobore si el modelo de entrenamiento es válido o no.

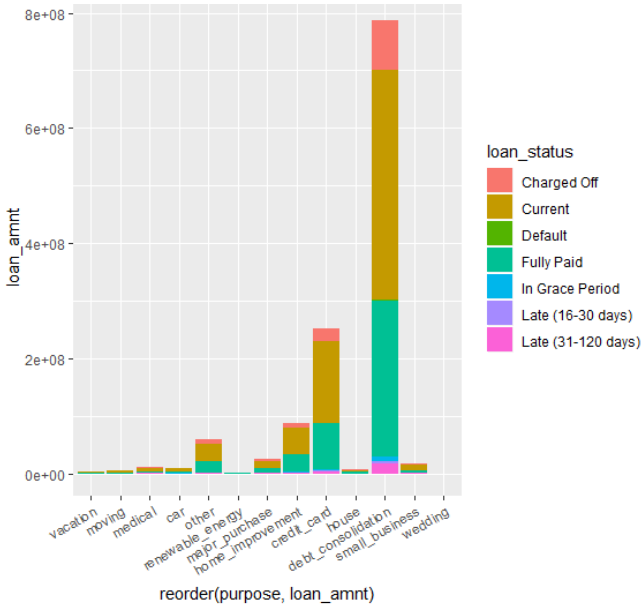
En nuestro nuevo *dataset* aún quedan observaciones que contienen información nula *[NA]* la cual hay que eliminar.




A partir de este gráfico se pueden apreciar todos los estados del préstamo que es la variable con la que queremos operar, pero para ello habrá que pasarla a una variable factor con 2 niveles.

Una vez los datos han sido limpiados se construye un modelo de entrenamiento con el 70% de las variables del dataset (seleccionadas de manera aleatoria, pero habiendo establecido una *semilla* previa para poder replicar el modelo tantas veces como deseemos) y, otro modelo de prueba que contendrá el resto, esto es, un 30%.

Partiendo del primero se calcula la regresión, teniendo siempre en cuenta que nuestra variable objeto de estudio es *loan_status*. Los valores AIC y BIC de este son 198.6 y 664.5 respectivamente, valores que se usarán para contrastarlos con el siguiente.
Si nos fijamos por un momento en la variable *purpose y loan_amount* vemos el peso de cada categoría en la valoración global del modelo, lo cual debemos tener en cuenta también. En este caso *debt consolidation* se establece en primer lugar seguido de *credit card*.

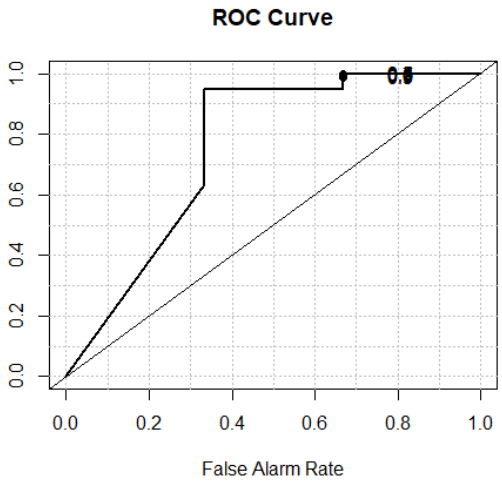


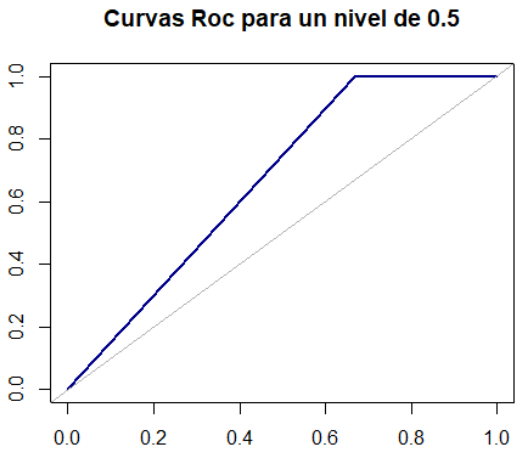

El siguiente paso sería la contrastación del error dentro y fuera de la muestra. El primer caso nos arroja el valor de 0.0023 y el segundo 0.0027. Esto podría indicarnos que nuestro modelo tiene un error bajo y por tanto podría ser válido.
Realizando posteriormente un análisis con un Cut-Off de 0.5 (este valor es a partir del cual podemos dar por válido un modelo), si aumentamos el valor el número de falsos positivos disminuye mientras que el de falsos negativos aumenta, por lo que habría que elegir un corte preciso para el modelo.

Analizando la matriz de confusión sacada con el anterior valor descrito vemos que la exactitud se sitúa en casi el 99%. Tenemos que tener en cuenta que estamos usando un modelo cuyos datos de *Fully Paid* rozan el 98%, siendo solo un 2% los valores *Charged Off* y por tanto la muestra elegida de manera aleatoria podría solo contener el primer valor y no el segundo.

```
Confusion Matrix and Statistics

   pred_cut_off
      0    1
  0   1    2
  1   1 1068

               Accuracy : 0.9972
                 95% CI : (0.9918, 0.9994)
    No Information Rate : 0.9981
    P-Value [Acc > NIR] : 0.8573

                  Kappa : 0.3987
 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.5000000
            Specificity : 0.9981308
         Pos Pred Value : 0.3333333
         Neg Pred Value : 0.9990645
             Prevalence : 0.0018657
         Detection Rate : 0.0009328
   Detection Prevalence : 0.0027985
      Balanced Accuracy : 0.7490654

       'Positive' Class : 0
```

Realmente para conseguir un punto de corte exacto tendríamos que recurrir a las curvas ROC y observar las gráficas con el objetivo de ajustar el corte de manera más real.



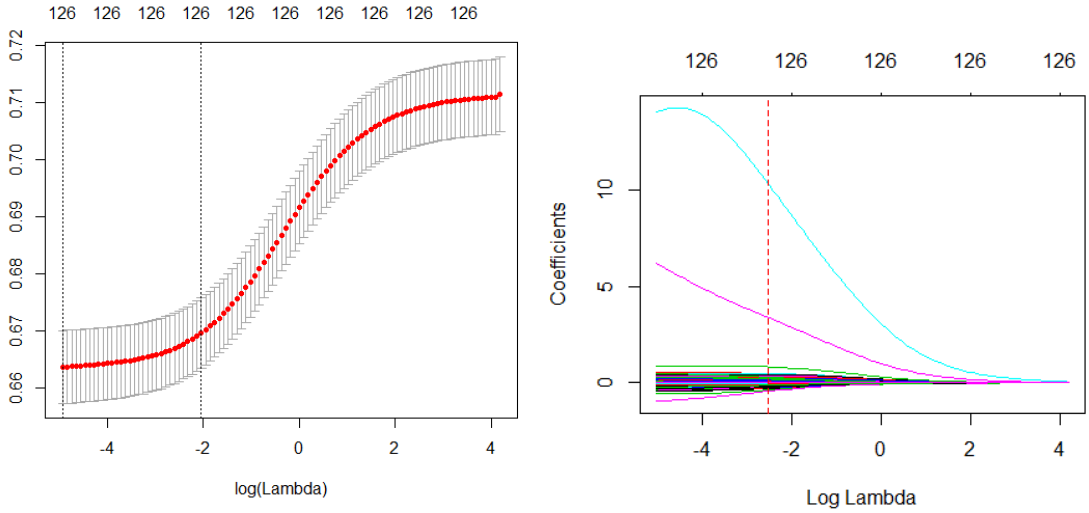




A partir de esto podríamos usar un valor algo superior a 0.6 para el estudio.
Finalmente podemos comprobar que el p valor sacado sea también bajo para verificar la certeza del modelo.

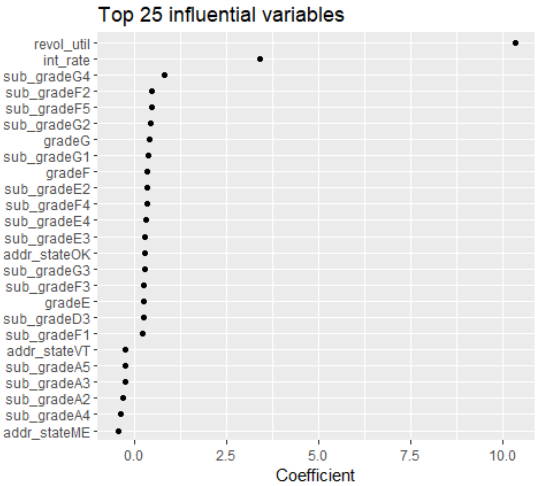
```
> roc.plot(test.data$loan_status == '1', prob.reg.outsample)$roc.vol
     Model       Area    p.value binorm.area
1 Model  1 0.7545993 0.03905598          NA
```

La regresión ridge usa la regularización para penalizar los residuos cuando el parámetro de un modelo de regresión está siendo aprendido. En general el resultado es un modelo que usa la muestra de entrenamiento peor que las regresiones anteriores, pero generaliza mejor porque es menos sensible a la extrema varianza provocada por los outliers en los datos.

Está regresión incluye un hiperparámetro que es lambda. La función que vamos a usar generará valores por nosotros, aunque es una práctica común establecer el nuestro propio.

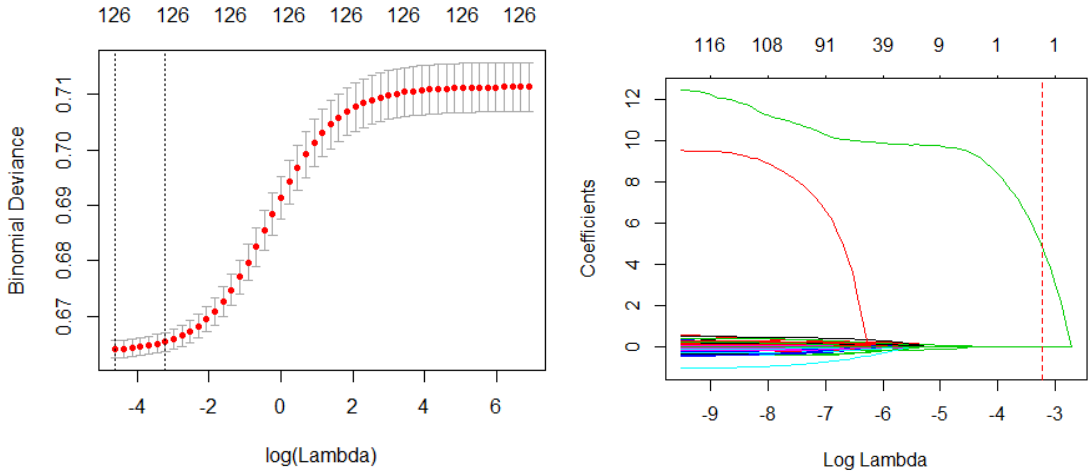



El punto mínimo en la primera gráfica indicará el lambda óptimo, en nuestro caso ha sido necesario extraerlo con otra función aparte la cuál nos arroja un valor de 0.01. A medida que aumenta lambda, el error cuadrático medio aumenta, lo cual es malo. La primera línea es el mínimo y la segunda en la línea de corte que se elige, que son estadísticamente iguales. Se puede ver gráficamente cuando cambia la pendiente cuando tenemos que elegir el valor.
Aplicando un plot directo para ver las 25 variables que más influencia tienen tenemos a revol_util en primer lugar, seguida de int_rate.

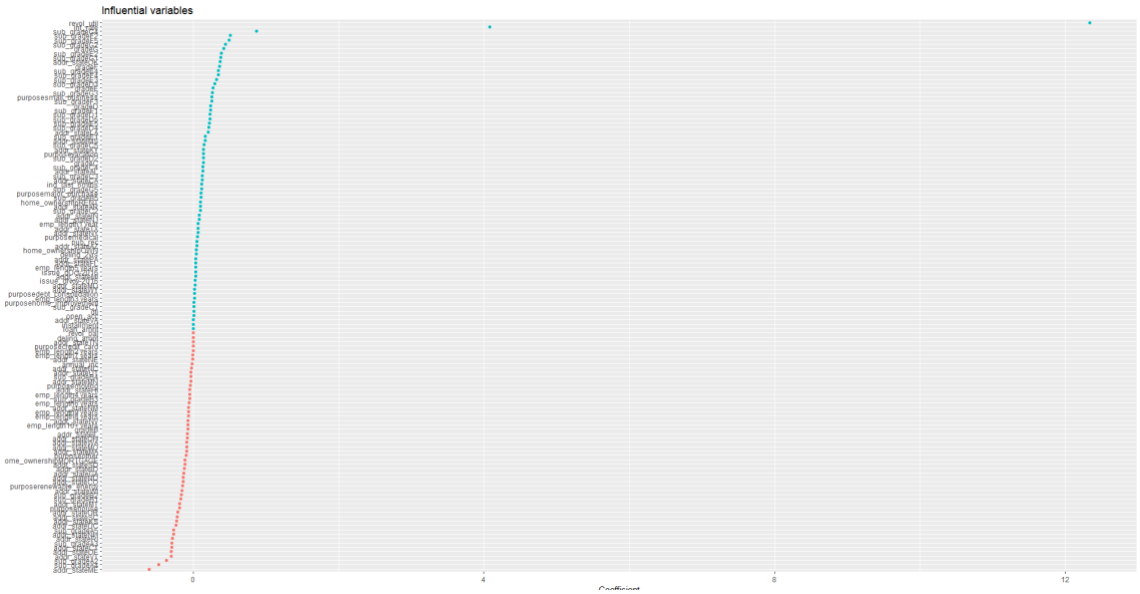



Mientras que en la regresión ridge el efecto penalizador es reducir los coeficientes que contribuyen más a error, en la regresión lasso es directamente establecerlos a cero, lo

que significa que está regresión actúa como un selector que coge los coeficientes más importantes (los más predictivos y los que tienen un p-valor más bajo)

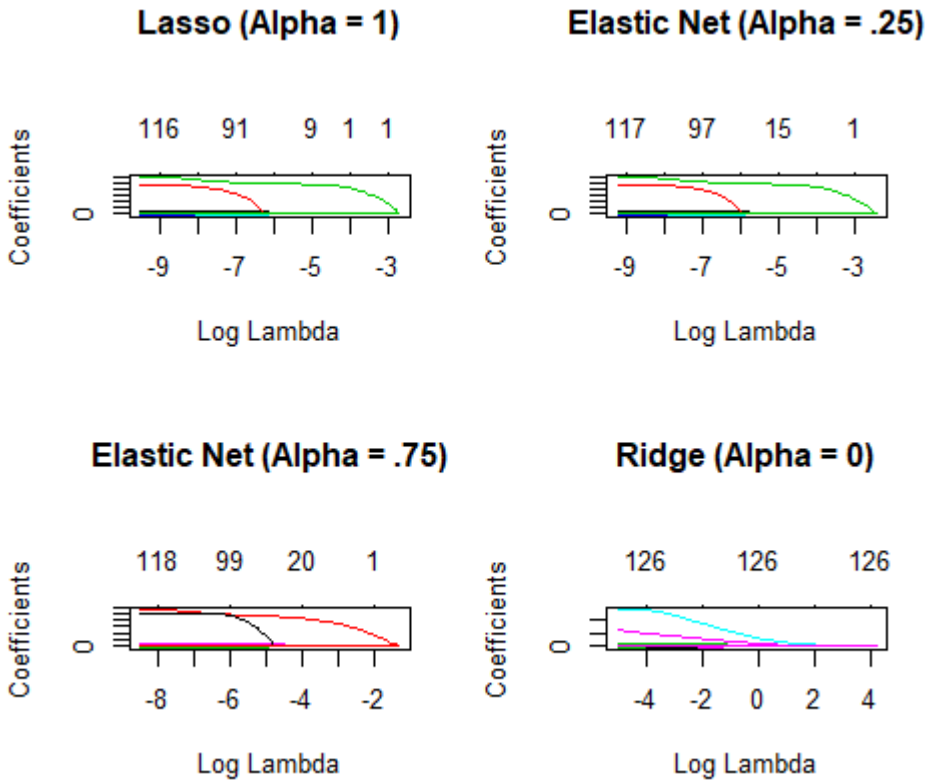



Al igual que en el modelo anterior se establece un punto mínimo de manera gráfica fuera de la vista por lo que para hallarlo volveremos a usar fórmulas. En este caso vuelve a ser 0.01 para un error medio del modelo de 0.664, el cual lleva un lambda de 0.039




Aunque no pueda apreciarse muy bien, en la parte alta encontramos los mismos valores que con la regresión ridge: *revol_util* y *int_rate.*

Para tener una visión global y no incluir solo las penalizaciones propias de los modelos ridge y lasso, se puede ampliar el rango del hiperparametro lambda para tener 4 observaciones en un gráfico.




Gracias a los datos de Lending Club, hemos creado un modelo de regresión para predecir si un préstamo será devuelto o no. Se han filtrado los datos para crear un modelo más limpio y que induzca a menos errores.

En base al modelo de entrenamiento y el de prueba y junto con los diferentes test realizados nos aseguramos que los resultados sean lo más precisos posible.