**MEMORIA: Ganancias y gastos de familias filipinas**

Artem Vartanov y Mario Baldocchi

Titulos a tamaño 14, texto tamaño 12

**Introducción**

Los [datos](https://www.kaggle.com/datasets/grosvenpaul/family-income-and-expenditure) que estudiaremos reflejan las ganancias y gastos de diversas familias filipinas, además de ciertos aspectos importantes a la hora de reforzar nuestra idea sobre el poder adquisitivo de las mismas.

Esta base de datos se podría emplear en diversos ámbitos laborales, pero supondremos que nuestro cliente es un banco, al que le interesa poder saber cuándo conceder un préstamo o denegarlo. Para poder hacer esto último, una de las principales variables a tener en cuenta es el dinero total que ingresa cada unidad familiar. Sin embargo, en Filipinas mucha gente trabaja en negro, por lo que no puede uno realmente fiarse de lo que el cliente deposita en la cuenta. Es por ello que, basándonos en los gastos y otros datos personales, trataremos de predecir el sueldo anual (en pesos filipinos, que a 30/11/2023 el cambio es de 1₱ = 0,017€) de una unidad familiar.

**Descripción de los datos**

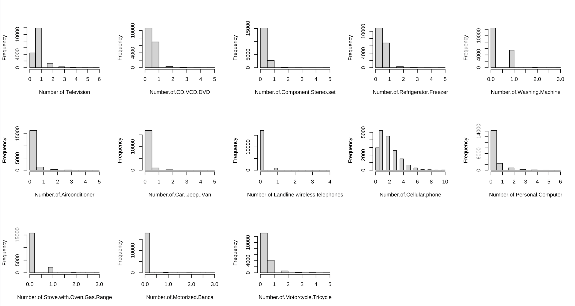
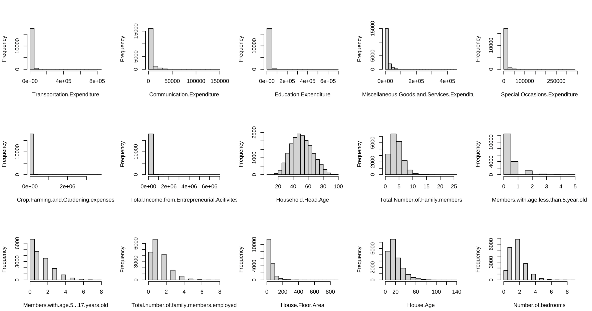
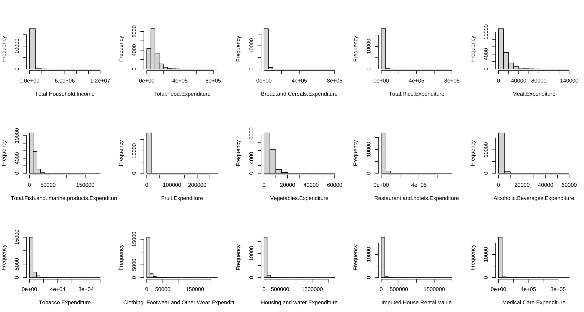
La base de datos se conforma de 18022 clientes y 60 variables, de las cuales 43 son nominales y 17 categóricas. Entre las distintas variables nominales se puede observar que aproximadamente un 25% están relacionadas con desembolsos monetarios, como por ejemplo los gastos médicos, en ropa o en comida. El resto describen la familia y la vivienda en la que residen, así como datos personales de la persona con mayor sueldo.

**Preprocesado**

Una vez importados los datos, lo primero que debe hacerse es echar un vistazo por encima. De lo primero que se percata uno, es que hay variables categóricas "escondidas", entre las que se encuentra, por ejemplo, *Electricity*. Y es que, aunque en esta columna solo haya números (0 y 1), estos son utilizados para indicarnos si el cliente en cuestión tiene electricidad en casa o no, es decir, se podría sustituir el 1 por un "Sí" y el 0 por un "No". Consecuentemente, hay que ir columna por columna comprobando que ninguna variable categórica se haga pasar por otro tipo de variable, y en el caso de localizar alguna, realizaremos su conversión a factor con as.factor <líneas 20-36>.

Seguimos analizando. Vemos que hay solo dos columnas que tienen valores N/A´s, *Household.Head.Occupation* y *Household.Head.Class.of.Worker,* que representan el trabajo del miembro de la familia con las mayores ganancias y para quién trabaja respectivamente. Ante esta situación se nos presentan varias opciones. Por un lado, se puede optar por la eliminación de todas aquellas filas que contengan un N/A (3355 en total). Por el otro, podría ser interesante prescindir de ambas columnas. En este caso, elegiremos la segunda opción por dos motivos principales. En primer lugar, tanto la una como la otra no aportan información relevante frente al resto de variables, por lo que es preferible perder dicha información a "sacrificar" un cliente. En segundo lugar, al ser dos variables categóricas no formarán parte del modelo de regresión, el cual es una de nuestras prioridades. Es verdad que si estuviésemos tratando con un banco que facilitase muy pocos datos de los clientes (no es nuestro caso), cabría la posibilidad de optar por quitar todas aquellas filas con algún N/A para posteriormente convertir las variables categóricas en "dummy", pudiendo así ajustar mejor el modelo de regresión.

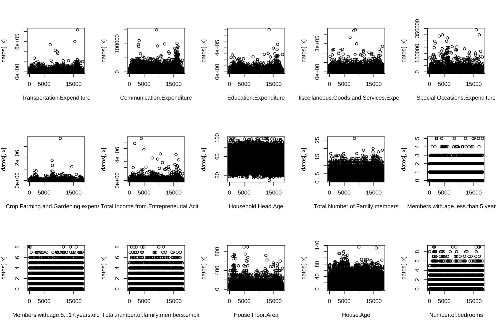
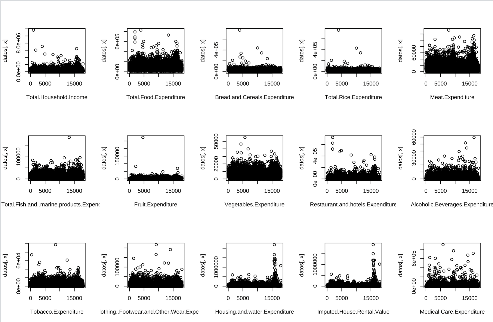
Para finalizar con el preprocesado, es recomendable prestar atención en cómo se distribuyen los datos. Para ello, podemos comenzar generando un histograma de cada una de las variables numéricas.

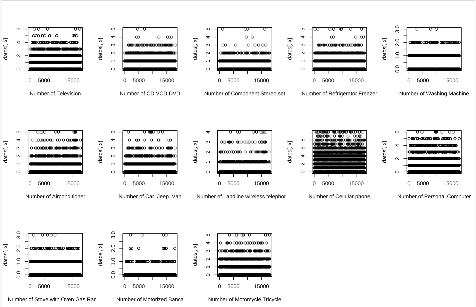


*(Histogramas de todas las variables numéricas)*

De esta manera, podemos percatarnos, por ejemplo, de que todos aquellos atributos que representan algún tipo de gasto tienen algo en común: son asimétricos hacia la derecha. Esto es algo completamente lógico, pues el gasto que realiza una familia solo puede encontrarse en el intervalo (0, ∞).

Además de la simetría, es de especial interés conocer si existen datos atípicos o no. Para esto último, en vez de generar histogramas, generaremos un diagrama de dispersión.





*(Diagramas de dispersión de todas las variables nominales)*

Como era de esperar, diversos atributos presentan outliers. Al estar tratando con variables simétricas y asimétricas, la eliminación de los datos atípicos puede resultar compleja. En el caso de que la variable sea simétrica, simplemente se extraerán todos aquellos datos que no se encuentren en el intervalo [q1 - coefIQR\*iqr, q3 + coefIQR\*iqr], donde coefIQR es igual a 3. Por otro lado, si la variable no es simétrica, hay que realizar un paso previo. Este consiste en transformar la variable, mediante box-cox, a una distribución cercana a una normal, principalmente simétrica.

Una vez realizados todos los pasos anteriormente mencionados, nos queda una base de datos limpia de N/A´s y outliers, donde la categoría de todas las variables es la correcta.

**Correlaciones**

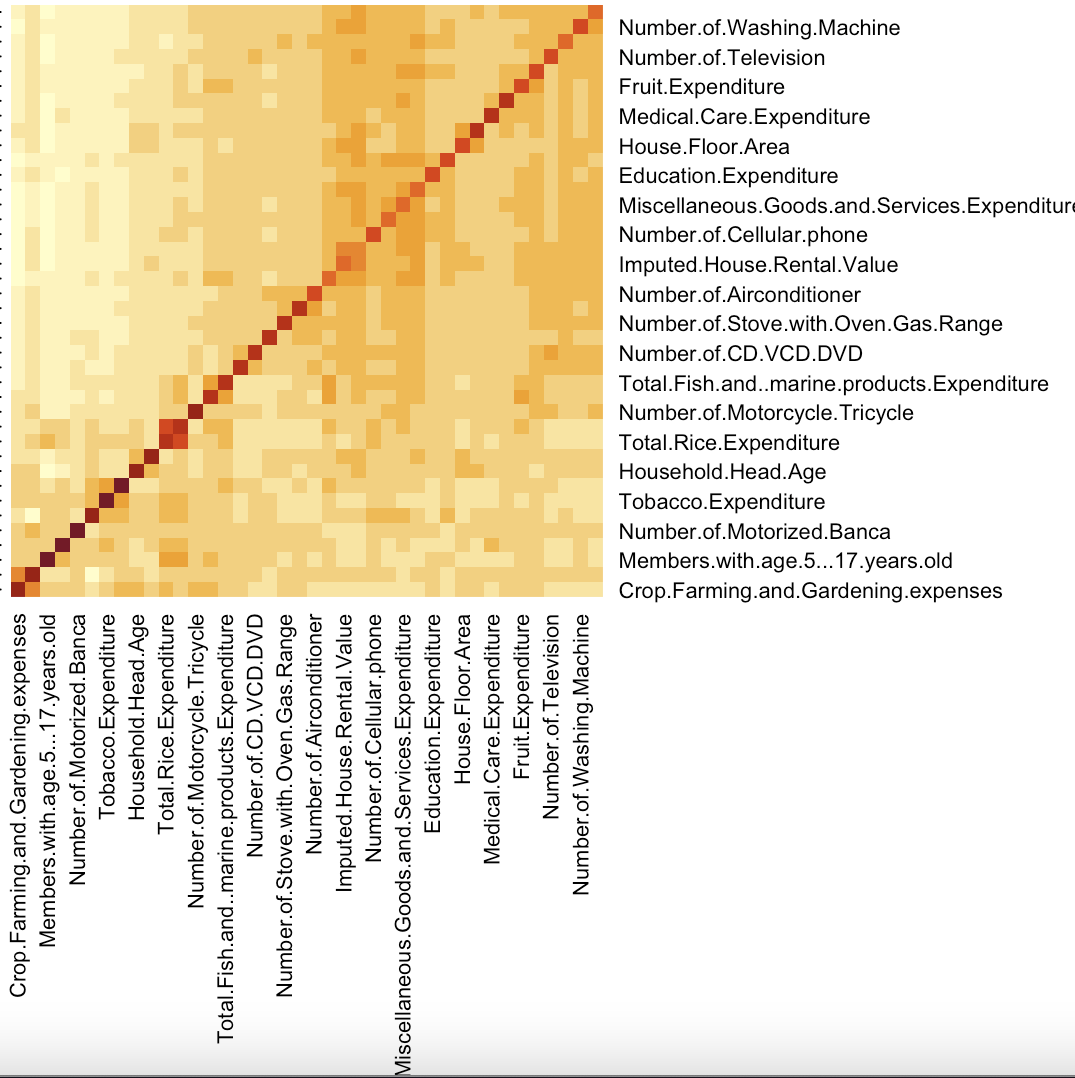
Como ya se ha mencionado anteriormente, las variables descriptivas del modelo lineal múltiple serán numéricas, por lo que lo primero que haremos será eliminar las columnas categóricas.

Uno de los aspectos fundamentales a la hora de construir nuestro modelo de regresión lineal múltiple es que no haya autocorrelación entre las variables descriptivas. Hay varios motivos para esto. En primer lugar, evitamos añadir información redundante

(lo que se podría traducir en una estimación menos eficiente de los coeficientes).

En segundo lugar, aumenta la estabilidad, pues la presencia de variables altamente correlacionadas provoca que el modelo sea sensible a pequeños cambios en los datos.

Por último, se mejora la interpretación del modelo, ya que nos permite conocer con mayor facilidad el impacto individual de cada variable descriptiva en la variable respuesta.

A continuación, estudiaremos la correlación entre todas

las variables menos la variable respuesta. Para empezar, vamos a utilizar un mapa de calor (Fig 1) para echar un vistazo por encima. Al tratarse de una base de datos con tantos atributos, es muy complicado sacar conclusiones.

El siguiente paso es seleccionar toda aquella columna que esté altamente correlacionada (el coeficiente de correlación de Spearman debe ser mayor de 0,7 en valor absoluto) con alguna otra.

Una vez realizado este proceso, obtenemos las siguientes variables: *Housing.and.water.Expenditure*, *Imputed.House.Rental.Value*, *Total.Rice.Expenditure,*

*Fig 1*

*Bread.and.Cereals.Expenditure* y *Communication.Expenditure*.

Generamos otro mapa de calor para estas cinco

Imagen que contiene Gráfico

Descripción generada automáticamente

columnas (Fig 2). Como se puede observar,

*Bread.and.Cereals.Expenditure* y *Total.Rice.Expenditure*

están muy correlacionadas. Eliminaremos la que tenga una

menor correlación con la variable respuesta, es decir,

*Total.Rice.Expenditure.* Por otro lado, podemos percatarnos

de que *Housing.and.water.Expenditure* está correlacionada

tanto con *Imputed.House.Rental.Value* como con

*Communication.Expenditure*, por lo que también la extraemos

de la base de datos.

*Fig 2*

Al volver a analizar la tabla de correlaciones, vemos que

ya no hay variables cuyo coeficiente de correlación de

Spearman sea mayor que 0.7 en valor absoluto.

Por último, de las variables descriptivas resultantes, seleccionaremos aquellas que más fuertemente se correlacionan con la variable respuesta y las guardaremos para usarlas más adelante.

Nota: Utilizamos el coeficiente de correlación de Spearman porque no todas las variables siguen una distribución normal (lo vimos en preprocesado).

**Regresión lineal múltiple**

**Modelo completo (RLM\_FULL)**

Generamos el modelo de regresión múltiple que usará todas las variables numéricas de nuestra base de datos (incluyendo columnas autocorrelacionadas). En la *Figura 3* aparece el resumen del modelo completo. Vemos que no todas las variables tienen un p-valor < 0.05 (valor prefijado para rechazar hipótesis nulas). Esto quiere decir que no podemos estar seguros de que el coeficiente de esa variable no sea 0, en otras palabras, que debería formar parte del modelo.

Cabe destacar que el R2 ajustado, que mide la proporción de la varianza total de la variable respuesta explicada por el modelo teniendo en cuenta una penalización basada en el número de predictores, es 0.798. Esto se debe a que tenemos 42 predictores, lo que deriva en una penalización bastante alta, pues muchos de ellos no mejoran significativamente la capacidad predictiva del modelo.

Después de calcular la raíz cuadrada del error del modelo nos sale 132910.2, lo cual parece ser un error crítico.

**Modelo mejorado (RLM\_BEST)**

Antes de comenzar, es importante recordar que cada vez que vayamos a generar un nuevo modelo, es necesario dividir el conjunto de datos que vayamos a emplear en dos grupos:

-train: Representa el 80% de los datos y será utilizado para entrenar el modelo.

-test: Está conformado por todos lo datos que no se encuentran en

el grupo “train”. Se emplea para comprobar el modelo.

Para comenzar, construiremos un modelo a partir de los atributos altamente correlacionados con la variable respuesta (los cuales ya habíamos guardado cuando hicimos el apartado de correlaciones). El error medio de la predicción es 272143,5 (más del doble que nuestro modelo anterior). Intentamos mejorarlo quitando observaciones influyentes a partir de residuos estudentizados, pero el error sale todavía peor, 272512.1. Pero quitando observaciones de alto leverage si que sale mejorar el error, 269037.3, pero sigue siendo altísimo por lo que descartamos este modelo.

Para el siguiente modelo, aplicaremos la validación cruzada. Esta vez dispondremos de todas las variables nominales entre las que no exista ninguna correlación. El primer paso consiste en dividir nuestro conjunto “train” en 4 subconjuntos, para evitar “overfitting” (puede producirse cuando el modelo es muy complejo). Esto último provoca que el modelo, más que descubrir patrones que relacionan las variables, se “aprenda” los datos pertenecientes a “train”, dando lugar a una mala predicción cuando se aplica con nuevos datos (el conjunto “test”). //.

Una vez generado el modelo, comprobamos si es bueno o no. El error medio de la predicción es 147132, 9. Intentamos mejorarlo quitando observaciones influyentes a partir de residuos estudentizados, llegando a un resultado de 141859.3. Además, quitando observaciones de alto leverage sale mejorar el error hasta 140866.3. Puede llamar la atención que siga prediciendo peor, pero no se está siendo del todo justo a la hora de comparar nuestro modelo actual con el modelo completo (RLM\_FULL). Y es que el modelo actual está realizando la predicción utilizando datos nuevos, mientras que el modelo completo predice empleando datos que han sido usados en su entrenamiento (sólo funciona con los datos de los que disponemos, pero empeoraría si el banco nos proporcionase otra base de datos con nuevos clientes).

Es momento de pasar a comprobar la normalidad, homocedasticidad e independencia de los residuos. Tanto para la normalidad como para la homocedasticidad, los residuos no superan las pruebas. Sin embargo, si cumplen el test de independencia. Para arreglarlo realizamos la transformación Box-Cox a la variable respuesta y quitamos los outliers, pues la variable ahora es simétrica, usando IQR. Pero no mejora el p valor del test de normalidad. Nuestra última esperanza recae sobre las transformaciones de las variables explicativas asimetricas, quitando los outliers a partir del IQR. Hechas las operaciones necesarias para conseguir lo que queremos, nos damos cuenta que el p valor a subido un poco a 1.508e-12, pero sigue siendo muy pequeño. Quitamos observaciones influyentes a partir de los residuos estudentizados y volvemos a probar los tests. Esta vez el p-valor es mayor que 0.05, por tanto acepta la hipótesis de normalidad. Calculamos el error de nuevo modelo: 54325792, un error cósmico.

**Modelos regularizados**

**Regularización de Tíjonov (Ridge)**

Recordemos que en este tipo de regularización se añade una penalización (siempre que λ sea distinto de 0) al método de mínimos cuadrados para hacer que los valores de coeficientes se aproximen a 0, pero no llega a eliminar predictores. En primer lugar, vamos a trazar un modelo de regresión “Ridge” en el conjunto “train”. Posteriormente, realizamos la validación cruzada para encontrar el valor de λ que minimiza el error. Por último, utilizando el modelo que acabamos de generar, vamos a realizar las predicciones tomando el valor de λ que minimiza el error.

En conclusión, observamos que el error medio nos da 97497,05. Este modelo predice mejor que el modelo de mínimos cuadrados, pero no nos convence del todo puesto que no elimina predictores.

**Regularización Lasso**

La regularización “Ridge” es muy similar a la regularización Lasso, con la diferencia de que la segunda llega a eliminar predictores menos relevantes. El proceso es el mismo que el mencionado en la regresión “Ridge” (creación de un modelo -> búsqueda, mediante validación cruzada, del valor λ que minimice el error -> predicciones).

Una vez finalizado el proceso, uno se puede percatar de que no tan solo el error medio ha disminuido a 97336,29 (lo que indica unas mejores predicciones), sino que también lo han hecho el número de variables descriptivas (nos quedamos con 28 de 38).

**Penalización Elastic net**

Esta penalización consiste en una regularización que es producto de la combinación de los dos métodos anteriormente mencionados (busca sacar lo mejor de ambos). Según el valor del α, esta regularización se “inclinará” más hacia una regresión Lasso (si α = 1 no habría diferencia) o una Ridge (si α = 0 no habría diferencia).

En este caso, no trazaremos un único modelo de regresión, sino 20 (los cuales guardaremos en una lista), donde en cada uno de ellos el valor que tome α (α ∈ [0,1]) será diferente, empezando por α = 0 hasta α = 1. Entre un modelo y su sucesivo, α se incrementará en 0,05. Volvemos, a través de la validación cruzada, a buscar el λ que minimice el error. Finalmente, nos queda una tabla compuesta por 20 filas (una por cada modelo) y 3 columnas (el α del modelo, el λ que minimiza el error para el α dado, y el error del modelo teniendo en cuenta como parámetros el α y el λ anteriores). Quedándonos con el error mínimo de la tabla obtenemos 97256,58, por lo que podemos concluir que con la penalización Elastic net hemos obtenido el mejor modelo. Como curiosidad, podemos saber que el mejor modelo encontrado con Elastic net no es una regresión Lasso ni una Ridge, porque, a parte de que el error medio es diferente, el número de predictores es 31 (más que en la Lasso y menos que en la Ridge).

**Modelos PLS y PCR**

Antes de realizar cualquiera de los dos siguientes modelos, recuerda que todos los predictores del conjunto de datos deben ser cuantitativos. Además, no hace falta eliminar variables que estén correlacionadas, pues no supone un problema para este tipo de métodos.

**PCR**

A partir de la validación cruzada, queremos encontrar el modelo con el menor número de componentes que mejor explique la variabilidad de los datos. Nos da el modelo con xxx componentes.

**PLS**

A partir de la validación cruzada, queremos encontrar el modelo con el menor número de componentes que mejor explique la variabilidad de los datos. Nos da el modelo con xxx componentes

**Regresión Logística**

La semana pasada se ha producido uno de los peores terremotos de la historia de Filipinas, dejando gran parte del país destrozado. Consecuentemente, muchas familias han perdido sus casas. Este acontecimiento ha hecho que el banco haya visto una oportunidad de oro para aumentar su capital. Es por ello que ha puesto en marcha el PRVF (Plan para la reconstrucción de viviendas en Filipinas), un préstamo de 90000₱ (1485€) al que cualquier familia filipina puede acceder siempre y cuando cumpla una condición: la renta per cápita mensual (ingresos mensuales de la familia / número de miembros en la familia) debe ser mayor o igual al 5% (4500₱) del préstamo.

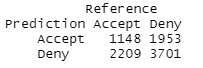
Para poder realizar el modelo de regresión logística utilizaremos la misma base de datos que hasta ahora, con una pequeña diferencia, y es que *Total Household Income* se transforma en una variable categórica, donde los únicos posibles valores que puede tomar son “Accept”, si la familia cumple la condición, o “Deny”, en caso contrario.

Antes de comenzar a entrenar el modelo, es necesario dividir nuestros datos en dos conjuntos:

-train: se utiliza para entrenar el modelo.

-test: se emplea para comprobar el modelo.

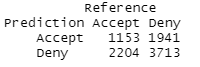
En el primer modelo vamos a tener en cuenta todos los atributos nominales (que no estén correlacionados) disponibles en la base de datos. El modelo resultante se aproxima al lanzamiento de una moneda, por lo que no es interesante.





*(Datos del modelo)*

Para tratar de mejorarlo, podemos quitar las variables con un p-valor muy alto. Sin embargo, esta acción tan solo mejora el modelo en un 0,19%, tal y como se puede ver en la siguiente imagen.





*(Datos del modelo)*

Antes de conformarnos y quedarnos con este modelo, comprobemos si hay uno mejor. Para ello aprovechémonos de la validación cruzada, la cual realizaremos con 20 grupos (k = 20). Se puede observar como el modelo ha mejorado muy notablemente.





*(Datos del modelo obtenido mediante validación cruzada)*

Y es que no solo es bueno por la exactitud a la hora de clasificar a los clientes del banco, sino también porque tiene una mayor tendencia a clasificar a los “Accept” como “Deny”(falsos negativos) que los “Deny” como “Accept”(falsos positivos). Esto último es de especial relevancia para el banco, pues se asegura de que el modelo realiza predicciones más “conservadoras”, disminuyendo así la probabilidad de que el préstamo no sea devuelto, lo que se traduce en un menor riesgo de pérdida de capital.

Tabla de resultados

|  |  |  |
| --- | --- | --- |
| MODELO | ERROR | N componentes |
| RLM\_FULL | N/A | N/A |
| RLM\_HIGH\_CORR | N/A | N/A |
| RLM\_REGSUBSETS |  |  |
| RLM\_RIDGE |  |  |
| RLM\_LASSO |  |  |
| RLM\_ELASTIC |  |  |
| RLM\_PCR |  |  |
| RLM\_PCR |  |  |
| RLM\_LOGISTICA |  |  |