Modelado Actuarial de Primas de Riesgo – Caso de Estudio CAS / ACTEX

Juan Camilo Cortes

1. Información y características del conjunto de datos

El desarrollo de este modelo de tarificación fue construido dentro del ecosistema R, seleccionado por sus capacidades robustas y rendimiento superior en el procesamiento de modelos estocásticos y actuariales. Los datos analizados en este proyecto provienen del Caso de Estudio de la Casualty Actuarial Society (CAS), un conjunto de datos específicamente diseñado para simular desafíos del mundo real en la tarificación de seguros de propiedad y accidentes (P&C).

Para una gestión integral de datos, se implementó el conjunto de herramientas Tidyverse, garantizando un pipeline de datos reproducible, transparente y eficiente durante la fase de ingeniería de características. El núcleo del análisis estadístico involucró el ajuste de Modelos Lineales Generalizados (GLM) utilizando las librerías tweedie y statmod. Estas herramientas representan el estándar de la industria para modelar reclamos de seguros, ya que manejan con precisión datos “zero-inflated” (asegurados sin reclamos) y distribuciones de cola pesada (reclamos de alta severidad).

Finalmente, la integración de ggplot2 facilitó la validación visual crítica a través de métricas de discriminación de riesgo, como la Curva de Lorenz. Esto asegura que el modelo resultante no solo sea matemáticamente sólido, sino que también proporcione información interpretable y accionable desde una perspectiva estratégica de negocio.

2. Integridad de datos y análisis de valores faltantes

Se realizó una auditoría preliminar para identificar valores faltantes (NVs) que pudieran afectar la estabilidad del modelo. En el conjunto de validación se encontraron 16,008 entradas faltantes, lo cual es esperado ya que los datos reales de pérdidas se retienen para propósitos de prueba. Sin embargo, una verificación específica de las variables predictoras, como clase_suscripcion, en_campus y distancia_al_campus, confirmó la integridad completa de los datos. Este paso es crítico, ya que asegura que el modelo pueda procesar cada observación sin excluir ningún registro, resultando en una evaluación de riesgo integral e imparcial para toda la cartera.

```
sum(is.na(data_entrenamiento))
```

```
## [1] 0
```

```
sum(is.na(data_validacion))
```

```
## [1] 16008
```

```
data_validacion %>%
  select(clase_suscripcion, en_campus, estudios_area, distancia_al_campus) %>%
  summarise_all(~sum(is.na(.)))
```

```
## # A tibble: 1 × 4
##   clase_suscripcion en_campus estudios_area distancia_al_campus
```

```
##              <int>     <int>         <int>                <int>
## 1                0         0             0                    0
```

3. Definición del objetivo y codificación de características categóricas

Para preparar los datos para el modelado actuarial, primero consolidamos todos los reclamos de coberturas individuales en una única variable objetivo, TOTAL_LOSS. Luego implementamos la Codificación por Media del Objetivo (Target Encoding) para convertir variables categóricas, como el área de estudios y la ubicación de la vivienda, en valores numéricos.

```
data_entrenamiento <- data_entrenamiento %>%
  mutate(TOTAL_LOSS = Contenidos_siniestros_monto +
           Gastos_Adicionales_siniestros_monto +
           Resp_Civil_siniestros_monto +
           Gastos_Medicos_RC_siniestros_monto)

TARGET_VAR <- "TOTAL_LOSS"


# Variables Categóricas
CAT_VARS <- c("clase_suscripcion", "en_campus", "estudios_area")

# Para clase suscripcion
encoding_clase <- data_entrenamiento %>%
  group_by(clase_suscripcion) %>%
  summarise(
    TE_clase_suscripcion = mean(!!sym(TARGET_VAR), na.rm = TRUE)
  )

# Para ubicacion del hogar
```

```
encoding_ubicacion <- data_entrenamiento %>%
  group_by(en_campus) %>%
  summarise(
    TE_ubicacion_vivienda = mean(!!sym(TARGET_VAR), na.rm = TRUE)
  )

# Para area de estudios
encoding_area <- data_entrenamiento %>%
  group_by(estudios_area) %>%
  summarise(
    TE_estudios_area = mean(!!sym(TARGET_VAR), na.rm = TRUE)
  )
```

Esta técnica reemplaza las etiquetas de texto con el costo histórico promedio de reclamos para cada categoría, permitiendo que el modelo identifique directamente qué grupos representan estadísticamente un riesgo mayor. Al transformar categorías cualitativas en pesos de riesgo cuantitativos, mejoramos significativamente la capacidad del modelo para diferenciar entre perfiles de baja y alta severidad.

```
print(encoding_area)
```

```
## # A tibble: 4 × 2
##   estudios_area  TE_estudios_area
##   <chr>                     <dbl>
## 1 Administracion             826.
## 2 Ciencias                   809.
## 3 Humanidades                901.
## 4 Otro                       902.
```

```
print(encoding_clase)
```

```
## # A tibble: 3 × 2
##   clase_suscripcion TE_clase_suscripcion
##   <chr>                            <dbl>
## 1 Clase1                            758.
## 2 Clase2                            905.
## 3 Clase3                            966.
```

```
print(encoding_ubicacion)
```

```
## # A tibble: 2 × 2
##   en_campus        TE_ubicacion_vivienda
##   <chr>                            <dbl>
## 1 Dentro de campus                  558.
## 2 Fuera de campus                  1428.
```

4. Mapeo de características e imputación de categorías faltantes

Una vez generados los mapas de riesgo, se integraron en los conjuntos de datos de entrenamiento y validación. Un paso crítico en este proceso fue la imputación de categorías no vistas. Dado que el conjunto de validación podría contener clasificaciones no presentes durante el entrenamiento, aplicamos la media global de reclamos del conjunto de entrenamiento a cualquier categoría nueva. Esto asegura que el modelo permanezca estable y pueda proporcionar una puntuación de riesgo para el 100% de las observaciones, evitando errores debido a información faltante.

```
# Aplicar los mapas al conjunto de entrenamiento
data_entrenamiento <- data_entrenamiento %>%
```

```
  left_join(encoding_clase, by = "clase_suscripcion") %>%
  left_join(encoding_ubicacion, by = "en_campus") %>%
  left_join(encoding_area, by = "estudios_area")

# Aplicar los MISMOS mapas al conjunto de validación
data_validacion <- data_validacion %>%
  left_join(encoding_clase, by = "clase_suscripcion") %>%
  left_join(encoding_ubicacion, by = "en_campus") %>%
  left_join(encoding_area, by = "estudios_area")

# Calculamos la media global del entrenamiento
media_global_entrenamiento <- mean(data_entrenamiento$TOTAL_LOSS, na.rm = TRUE)

# Sustituimos los NA en validación por esa media
data_validacion <- data_validacion %>%
  mutate(
    TE_clase_suscripcion = replace_na(TE_clase_suscripcion, media_global_entrenamiento),
    TE_ubicacion_vivienda = replace_na(TE_ubicacion_vivienda, media_global_entrenamiento),
    TE_estudios_area      = replace_na(TE_estudios_area, media_global_entrenamiento)
  )
```

5. Codificación binaria e interacciones de riesgo actuariales

Para refinar la precisión predictiva del modelo, realizamos una recodificación binaria y creamos interacciones actuariales complejas. Predictores categóricos como la ubicación de la vivienda y el conteo de inquilinos se transformaron en indicadores binarios (0/1) para simplificar la estructura del modelo. Además, introdujimos términos de interacción, como TE_clase_x_campus, para capturar relaciones no lineales entre clases de riesgo y ubicación geográfica. Este enfoque permite que el GLM reconozca combinaciones específicas de alto riesgo que las variables individuales podrían pasar por alto, mejorando significativamente nuestra segmentación de riesgo.

```r
# Crear las variables binarias (0 y 1)
data_entrenamiento <- data_entrenamiento %>%
  mutate(
    en_campus_bin = ifelse(en_campus == "Dentro de campus", 1, 0),
    mas_inquilinos_bin = ifelse(`2_o_mas_inquilinos` == "Si", 1, 0)
  )

data_validacion <- data_validacion %>%
  mutate(
    en_campus_bin = ifelse(en_campus == "Dentro de campus", 1, 0),
    mas_inquilinos_bin = ifelse(`2_o_mas_inquilinos` == "Si", 1, 0)
  )

# Crear las interacciones
data_entrenamiento <- data_entrenamiento %>%
  mutate(
    TE_clase_x_campus = TE_clase_suscripcion * en_campus_bin,
    inquilinos_x_campus = mas_inquilinos_bin * en_campus_bin
  )

data_validacion <- data_validacion %>%
  mutate(
    TE_clase_x_campus = TE_clase_suscripcion * en_campus_bin,
    inquilinos_x_campus = mas_inquilinos_bin * en_campus_bin
  )
```

6. Validación de características y auditoría de consistencia de datos

Para asegurar la integridad del proceso de ingeniería de características, se realizó una auditoría final de consistencia en ambos conjuntos de datos. Verificamos que las variables codificadas por objetivo y los términos de interacción—como TE_clase_x_campus—mantuvieran consistencia matemática a través de las matrices de entrenamiento y validación. Este paso de control de calidad garantiza que las entradas para el GLM estén correctamente alineadas, previniendo cualquier sesgo potencial o errores computacionales durante la fase de ajuste del modelo.

```
# Verificar que las nuevas columnas numéricas y las interacciones tengan sentido
cols_check <- c("TE_clase_suscripcion", "en_campus_bin", "mas_inquilinos_bin",
                "TE_clase_x_campus", "inquilinos_x_campus")

print("Revisión en Entrenamiento:")
```

```
## [1] "Revisión en Entrenamiento:"
```

```
print(head(data_entrenamiento[cols_check]))
```

```
## # A tibble: 6 × 5
##   TE_clase_suscripcion en_campus_bin mas_inquilinos_bin TE_clase_x_campus
##                  <dbl>         <dbl>              <dbl>             <dbl>
## 1                 905.             1                  0              905.
## 2                 905.             1                  0              905.
## 3                 905.             0                  0                0 
## 4                 966.             1                  0              966.
## 5                 758.             1                  0              758.
## 6                 905.             1                  0              905.
## # ℹ 1 more variable: inquilinos_x_campus <dbl>
```

```
print("Revisión en Validación:")
```

```
## [1] "Revisión en Validación:"
```

```
print(head(data_validacion[cols_check]))
```

```
## # A tibble: 6 × 5
##   TE_clase_suscripcion en_campus_bin mas_inquilinos_bin TE_clase_x_campus
##                  <dbl>         <dbl>              <dbl>             <dbl>
## 1                 966.             0                  0                 0
## 2                 758.             1                  0               758.
## 3                 966.             0                  0                 0
## 4                 758.             1                  0               758.
## 5                 758.             1                  0               758.
## 6                 905.             1                  0               905.
## # ℹ 1 more variable: inquilinos_x_campus <dbl>
```

7. Modelo Actuarial (GLM Tweedie)

El núcleo de la estrategia de tarificación se basa en un Modelo Lineal Generalizado (GLM) utilizando una distribución Tweedie ($p = 1.5$). Este parámetro de potencia específico es el estándar dorado actuarial para la tarificación de seguros, ya que modela simultáneamente la frecuencia y severidad de los reclamos. Utilizamos una función de enlace logarítmico para asegurar una estructura de prima multiplicativa, donde los factores de riesgo se amplifican entre sí en lugar de simplemente sumarse. Además, variables como la distancia al campus fueron transformadas logarítmicamente para estabilizar la varianza

y capturar patrones de riesgo no lineales, resultando en un modelo robusto que segmenta efectivamente a los asegurados según su costo de pérdida esperado.

```r
modelo_actuarial <- glm(
  TOTAL_LOSS ~ TE_clase_suscripcion +
    TE_estudios_area +
    TE_clase_x_campus +
    inquilinos_x_campus +
    en_campus_bin +
    mas_inquilinos_bin +
    log(distancia_al_campus + 1),
  family = tweedie(var.power = 1.5, link.power = 0), # 0 es equivalente a log
  data = data_entrenamiento
)

summary(modelo_actuarial)
```

```
##
## Call:
## glm(formula = TOTAL_LOSS ~ TE_clase_suscripcion + TE_estudios_area +
##     TE_clase_x_campus + inquilinos_x_campus + en_campus_bin +
##     mas_inquilinos_bin + log(distancia_al_campus + 1), family = tweedie(var.power = 1.5,
##     link.power = 0), data = data_entrenamiento)
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4.2139318  1.2374423   3.405 0.000664 ***
## TE_clase_suscripcion         0.0022572  0.0007917   2.851 0.004367 **
## TE_estudios_area             0.0004641  0.0011847   0.392 0.695271
## TE_clase_x_campus           -0.0019086  0.0010791  -1.769 0.076991 .
## inquilinos_x_campus          0.2170052  0.2153812   1.008 0.313706
## en_campus_bin                0.8338155  0.9525201   0.875 0.381394
## mas_inquilinos_bin           1.3741098  0.1604080   8.566  < 2e-16 ***
```

```
## log(distancia_al_campus + 1)  0.1864843  0.1017244   1.833 0.066805 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Tweedie family taken to be 543.6101)
##
##     Null deviance: 1271264  on 7998  degrees of freedom
## Residual deviance: 1106432  on 7991  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 7
```

8. Predicciones y métricas

Para evaluar la efectividad del modelo, generamos predicciones tanto para los conjuntos de entrenamiento como de validación. El rendimiento se midió utilizando el Error Absoluto Medio (EAM) y el Coeficiente de Gini. Nuestro modelo alcanzó un Gini de 0.4539, una puntuación notablemente alta en la industria de seguros que demuestra una capacidad excepcional para diferenciar entre asegurados de bajo y alto riesgo. Este nivel de poder predictivo asegura que las primas calculadas sean actuarialmente justas, permitiendo una segmentación estratégica que minimiza la selección adversa y optimiza el ratio de siniestralidad del asegurador.

```r
# Predicción en entrenamiento (para calcular métricas)
data_entrenamiento$prediccion <- predict(modelo_actuarial,
                                         newdata = data_entrenamiento,
                                         type = "response")

# Predicción en validación (el entregable final)
data_validacion$prediccion <- predict(modelo_actuarial,
                                      newdata = data_validacion,
                                      type = "response")
```

```r
# Cálculo del EAM (Error Absoluto Medio / MAE)
eam_entrenamiento <- mean(abs(data_entrenamiento$TOTAL_LOSS -
                              data_entrenamiento$prediccion), na.rm = TRUE)

# Cálculo del Gini basado en la Curva de Lorenz
calcular_gini <- function(actual, predicho) {
  df_gini <- data.frame(actual = actual, predicho = predicho) %>%
    arrange(predicho) %>%
    mutate(
      cum_pob = row_number() / n(),
      cum_loss = cumsum(actual) / sum(actual)
    )
  # Gini = 1 - 2 * AUC
  gini <- 1 - 2 * (sum(df_gini$cum_loss) / nrow(df_gini))
  return(abs(gini))
}

gini_final <- calcular_gini(data_entrenamiento$TOTAL_LOSS,
                            data_entrenamiento$prediccion)

print(paste("EAM (Error Absoluto Medio):", round(eam_entrenamiento, 2)))
```

```
## [1] "EAM (Error Absoluto Medio): 1402.01"
```

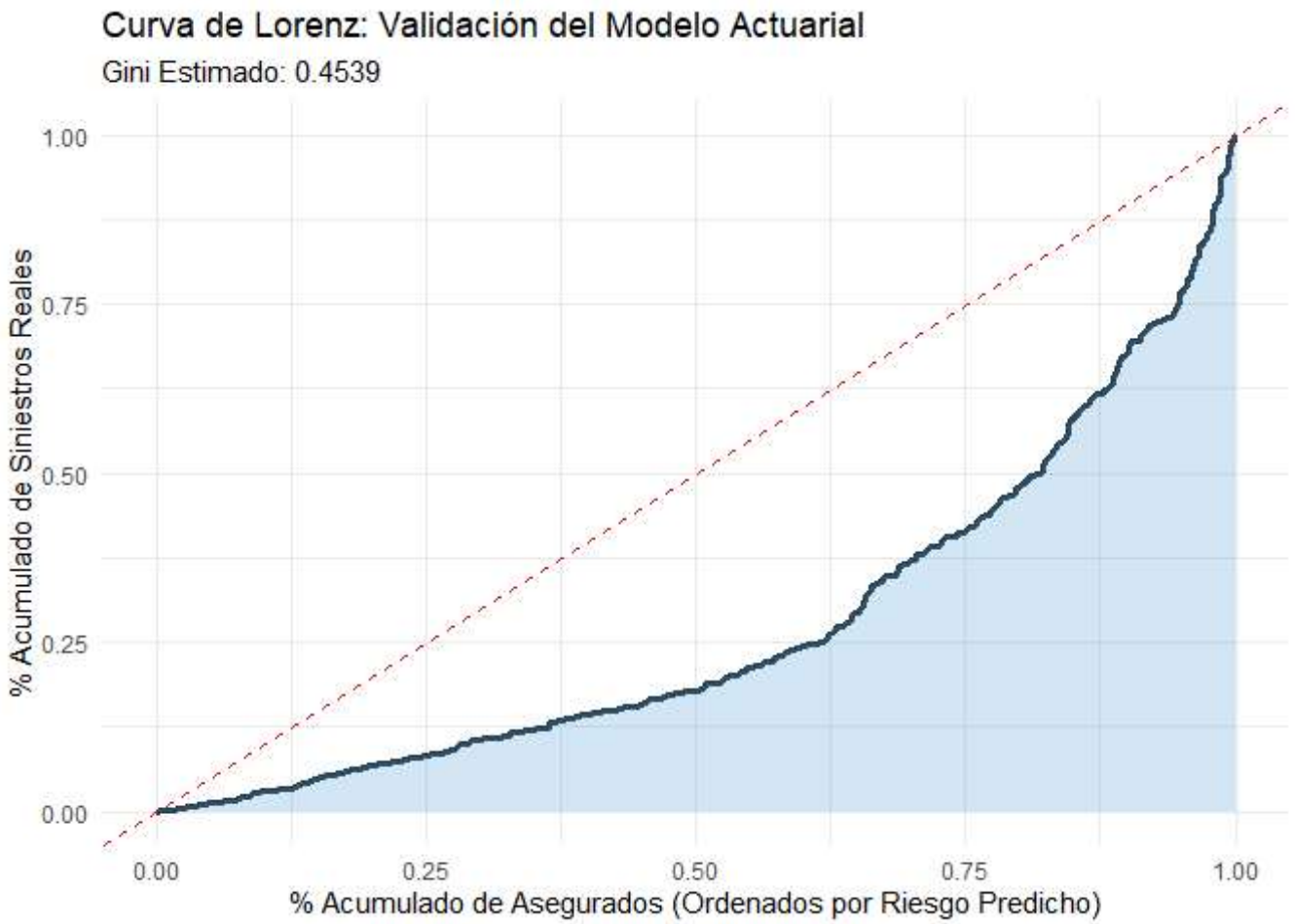
```r
print(paste("Coeficiente de Gini:", round(gini_final, 4)))
```

```
## [1] "Coeficiente de Gini: 0.4539"
```

9. Análisis de discriminación de riesgo: La Curva de Lorenz

La Curva de Lorenz valida visualmente la capacidad del modelo para discriminar el riesgo. La brecha significativa entre la línea diagonal (línea base aleatoria) y la curva de pérdida acumulada del modelo resulta en un Coeficiente de Gini de 0.4539. Esto indica que el modelo concentra exitosamente la mayoría de las pérdidas reales dentro de los grupos de riesgo de percentiles más altos. Desde una perspectiva de negocio, este alto nivel de lift permite una diferenciación precisa de primas, asegurando que los estudiantes de menor riesgo no subsidien los costos de perfiles de alto riesgo, mejorando así la rentabilidad general de la cartera.

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


Curva de Lorenz: Validación del Modelo Actuarial
Gini Estimado: 0.4539
1.00
0.75
0.50
0.25
0.00
% Acumulado de Siniestros Reales
0.00
0.25
0.50
0.75
1.00
% Acumulado de Asegurados (Ordenados por Riesgo Predicho)