Proyecto Final

Mauricio Hernández Velázquez
Andrés Moguel López Jensen
Francisco Javier Ramírez Martínez
Alfredo Sandoval Rubalcava
Marco Ishai Villacañas Luna

Machine Learning

Docente: Luis Zúñiga

5 de diciembre de 2022

Licenciatura en Actuaría
Otoño 2022
Universidad Iberoamericana

Table of Contents

Planteamiento

Se tiene una base de datos de Prudential en el que se tiene distinta información de productos, historia médica, información de clientes, información de empleo y de asegurados. Se desea crear un modelo que pueda clasificar los tipos de riesgos (1 a 8) de la mejor manera posible. El desarrolló del modelo se hace en Python utilizando las librerías pandas, numpy, sklearn, matplotlib

Análisis Exploratorio de Datos

Carga de Datos e Identificación de Variables

Tras cargar los datos a colab en dataframe, se hizo un análisis de la base de datos. La base de datos original consta de 128 columnas (127 variables regresoras y 1 variable de respuesta), la variable respuesta se identifica como “Response”. De las 128 variables, 18 son de tipo flotantes, 109 de tipo entero y 1 de tipo objeto (cualitativo) Se decide identificar y eliminar la variable de tipo objeto; es “Product_info_2”. Se identifica que la variable “Id” solo es para la etiqueta de datos y por lo tanto no será contemplada para el modelo desarrollado y es eliminada del dataframe.

Identificación de Valores Nulos y Duplicados

Para desarrollo correcto del modelo se revisa el conjunto de datos para identificar datos nulos, faltantes y duplicados. Se encontraron 393,103 datos nulos o faltantes, y se decidió rellenarlos. Esto se hace con los métodos “ffill” y “bfill”, no se encontraron datos duplicados y por lo tanto no es necesario eliminarlos.

Análisis de la Variable Respuesta

Se busca predecir la variable “Response” la cuál indica el tipo de riesgo que es. Esta variable es de tipo categórica; hay 8 tipos de riesgos en la base de datos. La distribución se ve de la siguiente manera:

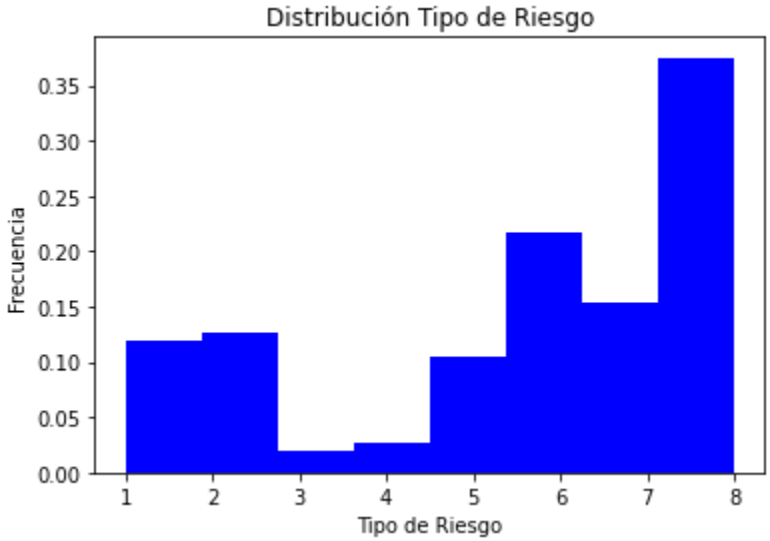

Figura 1: histograma de la variable respuesta.

En la figura 1 se puede observar que la distribución de los tipos de riesgo está desbalanceada. Hay muchos de tipo 8 (19,489) y de tipo 6 (11,233) hay muy pocos de tipo 3 (1,013) y de tipo 4 (1,428). Este desbalance será considerado en el desarrollo del modelo. A continuación, se desglosa el histograma de forma tabular:

Tipo de Riesgo	**n**
1	6,207
2	6,552
3	1,013
4	1,428
5	5,432
6	11,233
7	8,027
8	19,489

Tabla 1: frecuencia de tipos de riesgo

Desarrollo del Modelo

Análisis de Componentes Principales y Scaling

Se divide el data frame en dos partes: X y Y. X se compone de las 125 variables restantes que no son la respuesta y Y contiene solamente la variable respuesta. Ya que se están considerando una gran cantidad de dimensiones, se utiliza Análisis de Componentes Principales (PCA) para reducir la dimensionalidad de X. El PCA que se aplica retiene el 95% de la información o variabilidad de los datos. Los datos se escalan y se transforman con scaler.

Modelo a Desarrollar

Se decide utilizar regresión logística para clasificar los tipos de riesgos. Los modelos se entrenan y se ajustan, simultáneamente, mediante validación cruzada con 10 pliegues. Todos los modelos incluyen la penalización tipo l2 para regularizar los datos y evitar sobreajuste.

Se desarrollarán 6 modelos de regresión logística:

1. Modelo simple.
2. Modelo con balanceo.
3. Modelo con valor de “C” pequeño.
4. Modelo con valor de “C” mayor a 1.
5. Modelo con valor de “C” pequeño y balanceo.
6. Modelos con valor de “C” mayor a 1 y balanceo.

Modelo Simple

Este primer modelo de regresión logística solamente contempla aplicar regularización de tipo l2. Al ajustar el modelo se obtiene lo siguiente:

Métrica	Valor
Accuracy	0.3282026237348647
F1 Score	0.1621995925946184
Recall Score	0.3282026237348647
Precision	0.10771696222644916

Tabla 2: métricas del modelo simple

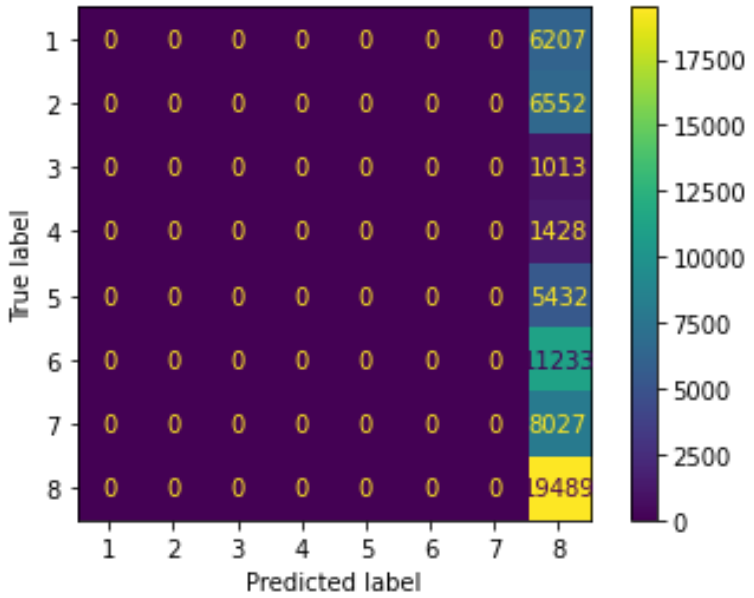

Figura 2: matriz de confusión del modelo simple

La matriz de confusión muestra que el modelo predice todos los riesgos como tipo 8, es decir, que predice correctamente todos los riesgos que, si son tipo 8 pero, incorrectamente todos los demás. Parece ser que el riesgo tipo 8 es un atractor o “imán” y este primer modelo tiende a predecir todos los riesgos como de tipo 8.

Modelo con Balanceo de Clases

Este modelo aplica la regularización con penalización de tipo l2 y les asigna un peso balanceado a las clases. Este balance de pesos se hace contemplando el desbalance que existe en la variable respuesta; entre la cantidad de elementos de cada clase. Recordemos que hay muchas respuestas correspondientes a los tipos de riesgo 8 y 6 y muy pocos para los tipos de riesgo 3 y 4. Al ajustar y entrenar el modelo se obtiene lo siguiente:

Métrica	**Valor**
Accuracy	0.13467270675805393
F1 Score	0.13486313084989882
Recall Score	0.13467270675805393
Precision	0.20082047974356945

Tabla 3: métricas del modelo con balanceo de clases

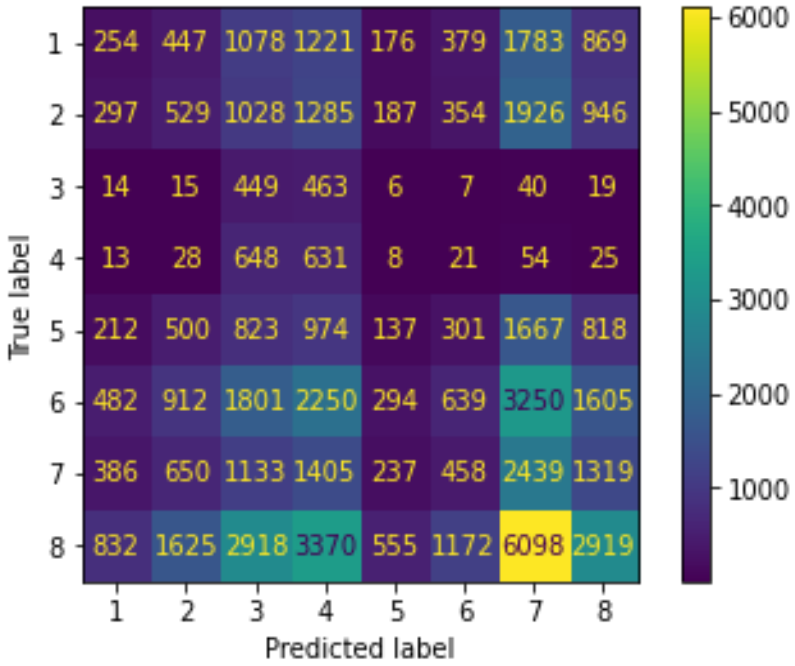

Figura 3: matriz de confusión del modelo con balanceo de clases

Al añadir el balance de clases, el modelo deja de predecir todas las respuestas como 8. Ya predice de mejor manera las otras clases de riesgos que componen la base de datos. Un detalle de esto es que ahora se inclina a predecir muchos 8 como 7. Ahora el atractor o imán, de manera menos intensa, es el 7. Respecto al modelo anterior, disminuye casi

10 puntos el accuracy, pero, casi se duplica la precisión. El f1 score baja 0.03 respecto al modelo anterior. El recall score disminuye en la misma cantidad que el accuracy; respecto al modelo anterior.

Modelo con valor de “C” pequeño

Este modelo aplica la regularización con penalización de tipo l2 y cambia el valor del parámetro C: la inversa de la fuerza de regularización. Un valor de C más pequeño especifica una regularización más fuerte o estricta. En este caso en el modelo se especifica un valor de C igual a 0.1. Al entrenar, ajustar y evaluar el modelo se obtiene lo siguiente:

Métrica	**Valor**
Accuracy	0.3282026237348647
F1 Score	0.1621995925946184
Recall Score	0.3282026237348647
Precision	0.10771696222644916

Tabla 4: métricas del modelo con valor de “C” igual a 0.1

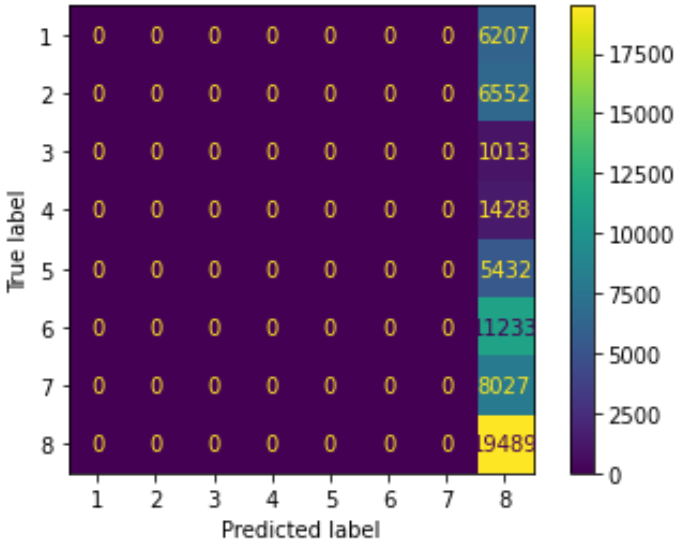

Figura 4: matriz de confusión del modelo con valor de “C” igual a 0.1

Es interesante resaltar que la figura 1 y la figura 4 son idénticas; se tiene la misma matriz de confusión. También se obtienen las mismas métricas de evaluación en ambos

modelos. Cambiar el valor del parámetro C de 1 (como en el modelo simple) a 0.1, no ayuda a mejorar el rendimiento del modelo.

Modelo con valor de “C” mayor a 1

Este modelo aplica la regularización con penalización de tipo l2 y cambia el valor del parámetro C. En este caso en el modelo se especifica un valor de C igual a 2. Al entrenar, ajustar y evaluar el modelo se obtiene lo siguiente:

Métrica	**Valor**
Accuracy	0.3282026237348647
F1 Score	0.1621995925946184
Recall Score	0.3282026237348647
Precision	0.10771696222644916

Tabla 5: métricas del modelo con valor de “C” igual a 2.

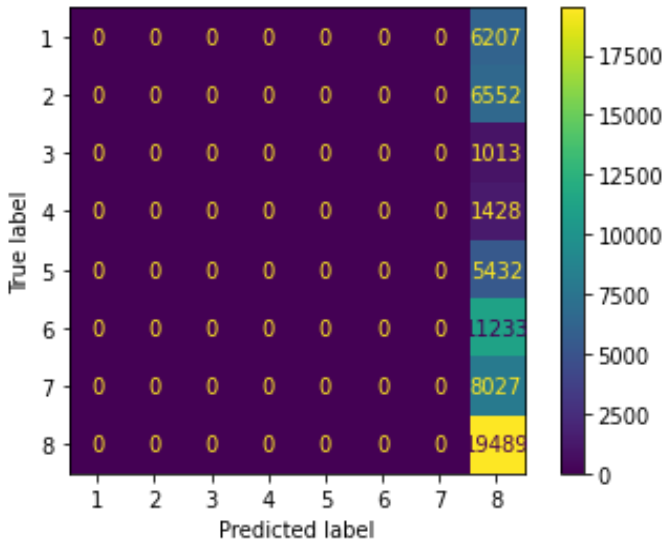

Figura 5: matriz de confusión del modelo con valor de “C” igual a 2

Es interesante resaltar que la figura 1 y la figura 5 (se da el mismo caso que con la figura 4) son idénticas; se tiene la misma matriz de confusión. También se obtienen las mismas métricas de evaluación en ambos modelos. Cambiar el valor del parámetro C de 1 (como en el modelo simple) a 2, no ayuda a mejorar el rendimiento del modelo.

Modelo con valor de “C” pequeño y balanceo

Este modelo aplica la regularización con penalización de tipo l2, cambia el valor del parámetro C y balancea las clases. En este caso en el modelo se especifica un valor de C igual a 0.1 y el class_weight igual a “balanced”. Al entrenar, ajustar y evaluar el modelo se obtiene lo siguiente:

Métrica	**Valor**
Accuracy	0.13460534514406966
F1 Score	0.13472508407657063
Recall Score	0.13460534514406966
Precision	0.20077494732140996

Tabla 6: métricas del modelo con valor de “C” igual a 0.1 y balanceo de clases.

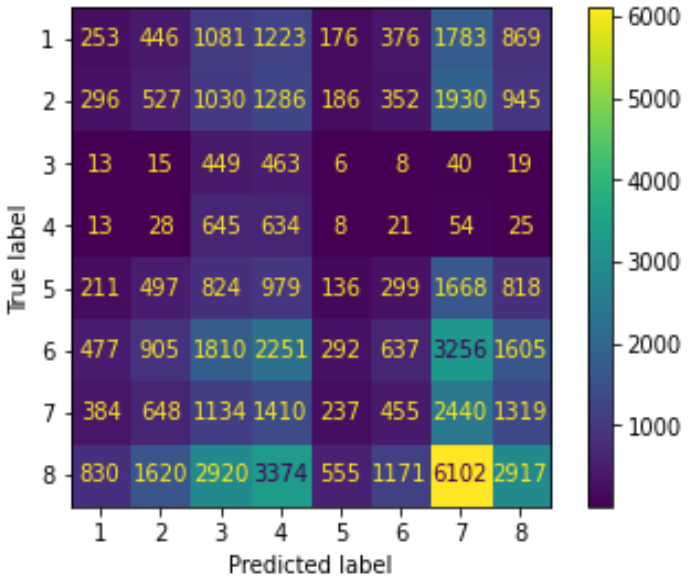

Figura 6: matriz de confusión del modelo con valor de “C” igual a 0.1 y balanceo de clases

Este modelo ajusta ligeramente peor que el modelo con balance de clases. Todas las métricas de evaluación son levemente menores que las del segundo modelo presentado.

Modelo con valor de “C” mayor a 1 y balanceo

Este modelo aplica la regularización con penalización de tipo l2, cambia el valor del parámetro C y balancea las clases. En este caso en el modelo se especifica un valor de C igual a 2 y el class_weight igual a “balanced”. Al entrenar, ajustar y evaluar el modelo se obtiene lo siguiente:

Métrica	**Valor**
Accuracy	0.13468954716155
F1 Score	0.13488777871712504
Recall Score	0.13468954716155
Precision	0.20085621099427406

Tabla 6: métricas del modelo con valor de “C” igual a 2 y balanceo de clases.

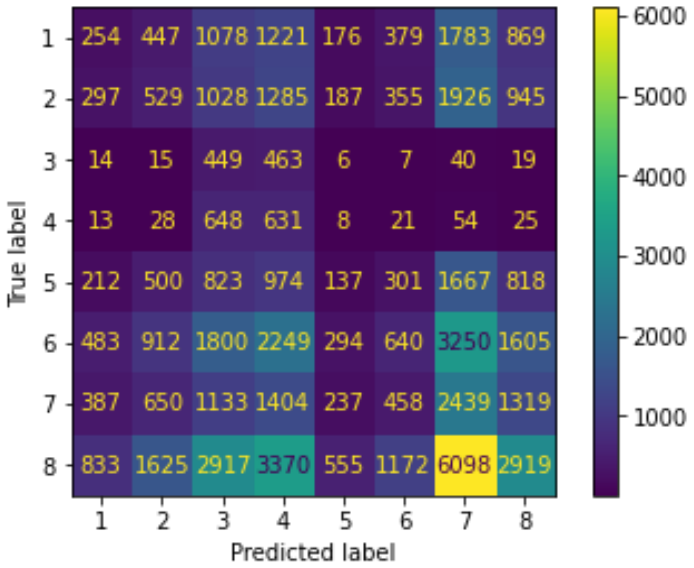

Figura 6: matriz de confusión del modelo con valor de “C” igual a 2 y balanceo de clases

Este modelo ajusta ligeramente mejor que el modelo con balance de clases. Todas las métricas de evaluación son levemente mayores que las del segundo modelo presentado.

Conclusiones

Es importante entender que no existe un mejor modelo. Todos los modelos tienen sus ventajas y sus desventajas. Siempre hay algún tipo de costo de oportunidad. Hay veces que se sacrifica accuracy por precisión o viceversa. Para este conjunto de datos se pudieron utilizar otro tipo de modelos como redes neuronales. Se sugiere revisar su implementación en algún trabajo futuro.