Predicción de Reclamaciones de Seguro de Automóviles

POR:

Andrés Felipe Calvo Ariza

Andrés Guillermo Toloza Guzmán

MATERIA:

Modelo y Simulación de Sistemas 1

PROFESOR:

Raúl Ramos Pollan

UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERÍA

MEDELLÍN 2023

Contenido

INTRODUCCCIÓN

Este informe se adentra en el análisis detallado de un conjunto de datos integral de seguros de vehículos, que refleja una variedad de atributos de los titulares de las pólizas. Las columnas del dataset proporcionan información meticulosa que abarca desde el período de tenencia de la póliza, la edad del automóvil, la edad del propietario, hasta la densidad poblacional del área de residencia, así como características técnicas del vehículo como la marca, el modelo, la potencia del motor y la presencia de funciones de seguridad y comodidad. Este estudio está diseñado para decodificar las complejidades inherentes a los datos y proporcionar una comprensión precisa de cómo estos factores pueden influir en la propensión de un titular de póliza a presentar una reclamación dentro de seis meses.

1. Planteamiento del problema

La predicción de reclamaciones de seguros de automóviles constituye un desafío significativo debido a la interdependencia y la naturaleza multifacética de las variables involucradas. Con un conjunto de datos que incluye información detallada de los vehículos y sus propietarios, el problema radica en identificar y cuantificar cómo las características específicas y los atributos técnicos del vehículo contribuyen a la probabilidad de ocurrencia de reclamaciones. La capacidad de predecir estas reclamaciones con precisión es de importancia crítica para la aseguradora, ya que permite una tarificación de riesgos más exacta, el diseño de pólizas a medida y la implementación de medidas preventivas efectivas. El desafío se magnifica al considerar la necesidad de equilibrar la rentabilidad y la competitividad en el mercado. Este informe busca establecer un modelo predictivo que pueda identificar con precisión los riesgos de reclamaciones futuras, proporcionando así una base sólida para la toma de decisiones estratégicas en la gestión de pólizas y el desarrollo de productos de seguros.

1.1. Dataset

El dataset contiene información detallada sobre los asegurados, incluyendo los atributos mencionados anteriormente, así como la variable objetivo que indica si el asegurado presentó un reclamo en los próximos 6 meses o no. Los datos se han recopilado previamente y se utilizarán para entrenar y evaluar el modelo predictivo. Cuenta con 44 columnas de las cuales más del 10% son categóricas.

Variable	Descripción
policy_id	Identificador único del titular de la póliza
policy_tenure	Periodo de tiempo de la póliza
age_of_car	Edad normalizada del coche en años
age_of_policyholder	Edad normalizada del titular de la póliza en años
area_cluster	Clúster de área del titular de la póliza
population density	Densidad de población de la ciudad (Ciudad del titular de la póliza)
make	Fabricante o compañía del coche codificado
segment	Segmento del coche (A/B1/B2/C1/C2)
model	Nombre codificado del coche
fuel_type	Tipo de combustible que utiliza el coche
max_torque	Torque máximo generado por el coche (Nm@rpm)
max_power	Potencia máxima generada por el coche (bhp@rpm)
engine_type	Tipo de motor utilizado en el coche
airbags	Número de airbags instalados en el coche
is_esc	Indicador booleano que señala si el coche tiene

	Control Electrónico de Estabilidad (ESC) o no.
is_adjustable_steering	Indicador booleano que señala si el volante del coche es ajustable o no.
is_tpms	Indicador booleano que señala si el Sistema de Monitoreo de Presión de Neumáticos (TPMS) está presente en el coche o no.
is_parking_sensors	Indicador booleano que señala si los sensores de estacionamiento están presentes en el coche o no.
is_parking_camera	Indicador booleano que señala si la cámara de estacionamiento está presente en el coche o no.
rear_brakes_type	Tipo de frenos utilizados en la parte trasera del coche
displacement	Desplazamiento del motor del coche (cc)
cylinder	Número de cilindros presentes en el motor del coche
transmission_type	Tipo de transmisión del coche
gear_box	Número de marchas del coche
steering_type	Tipo de dirección asistida del coche
turning_radius	Espacio que necesita el vehículo para realizar un giro (Metros)
length	Longitud del coche (Milímetros)
width	Ancho del coche (Milímetros)
height	Altura del coche (Milímetros)
gross_weight	Peso máximo permitido del coche completamente cargado, incluyendo pasajeros, carga y equipo (Kg)
s_front_fog_lights	Indicador booleano que señala si el coche dispone de luces antiniebla delanteras o no.
is_rear_window_wiper	Indicador booleano que señala si el limpiaparabrisas trasero está disponible en el coche o no.
s_rear_window_washer	Indicador booleano que señala si el lavaparabrisas trasero está disponible en el coche o no.
s_rear_window_defogger	Indicador booleano que señala si el desempañador de la ventana trasera está disponible en el coche o no.
__brake_assist	Indicador booleano que señala si la asistencia de frenado está disponible en el coche o no.
is_power_door_lock	Indicador booleano que señala si el coche tiene cierre de puertas eléctrico o no.
is_central_locking	Indicador booleano que señala si el coche cuenta con cierre centralizado o no.
is_power_steering	Indicador booleano que señala si el coche tiene dirección asistida o no.
s_driver_seat_height_adjustable	Indicador booleano que señala si la altura del asiento del conductor es ajustable o no.
is_day_night_rear_view_mirror	Indicador booleano que señala si el coche cuenta con espejo retrovisor día/noche o no.

ecw	Indicador booleano que señala si el coche cuenta con Aviso de Revisión del Motor (ECW) o no.
is_speed_alert	Indicador booleano que señala si el sistema de alerta de velocidad está disponible en el coche o no.
incap_rating	Calificación de seguridad otorgada por NCAP (de 5)
is claim	Resultado: Indicador booleano que señala si el titular de la póliza presentó un reclamo en los próximos 6 meses o no.

1.2. Métrica

El F1-Score es una métrica de evaluación de la calidad de un modelo de clasificación que combina la precisión (precision) y el recall (recuperación) en una sola métrica. Se calcula utilizando la siguiente fórmula:

$$F1 = \frac{2*precision*recall}{precision+recall}$$

Donde:

Precisión es la proporción de verdaderos positivos (TP) sobre la suma de verdaderos positivos y falsos positivos (FP). Se mide la precisión de las predicciones positivas del modelo

$$precision = \frac{TP}{TP + FP}$$

Recall (recuperación) es la proporción de verdaderos positivos (TP) sobre la suma de verdaderos positivos y falsos negativos (FN). Mide la capacidad del modelo para capturar todos los casos positivos.

$$recall = \frac{TP}{TP + FN}$$

Desde una perspectiva de negocio, se evaluará la métrica de "Ahorro Potencial". Esta métrica proporciona una estimación del dinero que la compañía de seguros podría ahorrar como resultado de la implementación del modelo de predicción.
El cálculo del ahorro potencial se basaría en la siguiente fórmula:
Ahorro Potencial = (Número de Reclamos Evitados) x (Costo Promedio de un Reclamo)
Donde:
"Número de Reclamos Evitados" es el número de reclamos que el modelo predice como
probables y, por lo cual se evita asegurar al vehículo de ese cliente.
"Costo Promedio de un Reclamo" es el costo promedio que la compañía de seguros normalmente paga por un reclamo hecho por un asegurado.

1.3. Variable Objetivo

Nuestra columna objetivo se llama is_claim, representa los valores de 1 si el asegurado presento un reclamo del seguro y 0 si no lo hizo dentro de los últimos 6 meses.

2. Exploración de variables

2.1. Análisis de la variable objetivo

Se analiza el comportamiento de la distribución que tiene la variable objetivo, dicho comportamiento de muestra en la *Figura 1,* donde se puede observar que la variable objetivo tiene un desbalance en los datos, este problema será resuelto usando SMOTEN.

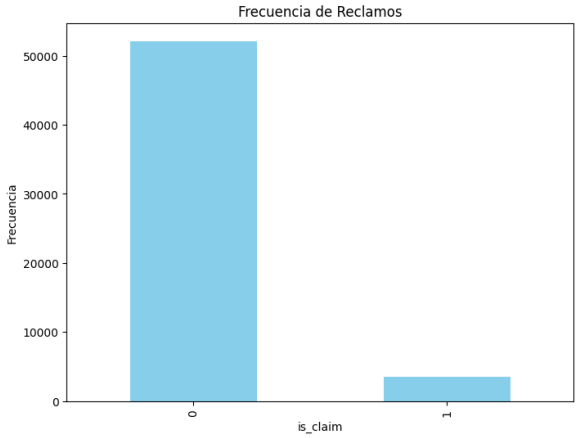

Figura 1. Distribución de la variable objetivo.

2.2. Datos faltantes

Antes de analizar para entrenar un algoritmo es importante buscar datos faltantes en el dataset. En la *Figura 2 y Tabla 3* se muestran la cantidad de datos faltantes de las variables que poseen valores NaN. Existen 3 columnas con 5% de nulos, estas columnas son **“policy_tenure”, “area_cluster”, “age_of_car”**. Por lo que debe tenerse en cuenta a la hora realizar el entrenamiento del modelo.

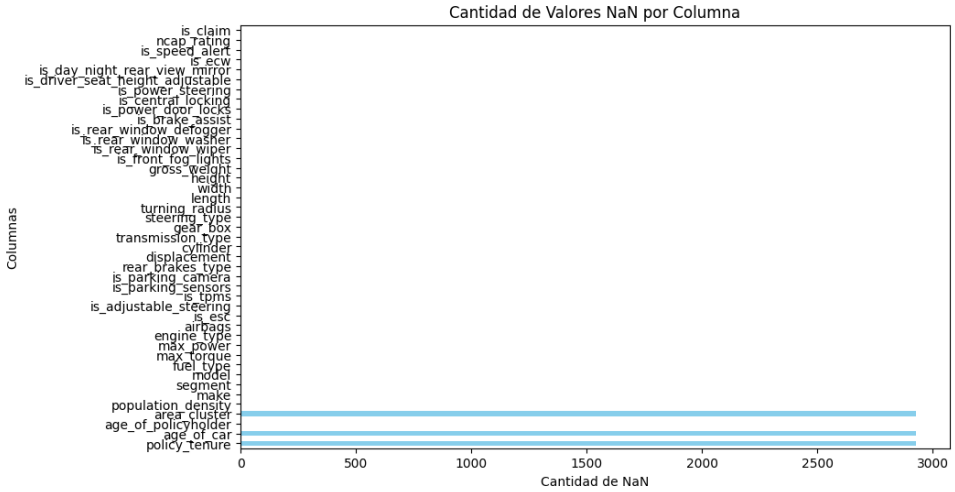

Figura 2. Datos faltantes por variable.

policy_tenure	2929
area_cluster	2929
age_of_car	2929

Figura 3. Datos faltantes por variable.

3. Tratamiento de datos

- **Remoción de policy_id:**
  La columna “policy_id” contiene un id aleatorio asociado a ese registro, para nuestro modelo no nos ofrece ninguna ventaja incluirlo en las variables a analizar.

```
train=pd.read_csv('train.csv')
train.drop('policy_id', axis=1, inplace=True)
```

Figura 4. Código para eliminar las lecturas cero de la variable objetivo.

- **Transformación de variables categóricas:**
  Es necesario realizar una transformación de las variables categóricas a variables numéricas antes de utilizarlas en el entrenamiento de un algoritmo, ya que las variables categóricas no son adecuadas para ser empleadas directamente en dicho proceso. La *Figura 5* muestra cómo se realiza la transformación de dichas variables.

```
#Transformamos los datos de las variables categóricas a una representación numérica
for columns in train.columns:
    if dict(train.dtypes)[columns] == 'object':
        label_encoder = preprocessing.LabelEncoder()
        train[columns] = label_encoder.fit_transform(train[columns])
```

Figura 5. Transformación de variables categóricas a variables numéricas.

4. Modelos

4.1. Selección de modelos

Decidimos trabajar con los siguientes modelos, HistGradientBoostingClassifier, LogisticRegression, RandomForestClassifier, KMeans, PCA. Para estos modelos se hace un estudio para obtener los mejores hiperparametros de las variables.

4.2. Mejores hiperparámetros de los modelos

Después de obtener los modelos que vamos a utilizar, llevamos a cabo un análisis para determinar los hiperparámetros óptimos. Para esto, empleamos un componente de la biblioteca scikit-learn llamado "GridSearchCV". Este componente nos permite explorar diferentes combinaciones de hiperparámetros de nuestro algoritmo de interés utilizando una técnica de validación cruzada (cross-validation). Como resultado, identificamos los hiperparámetros que producen el mejor rendimiento del modelo, y obtenemos el estimador final que incorpora estos ajustes óptimos.

```
# Definir los parámetros a ajustar
param_grid = {
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7]
}
```

Figura 6. Valores de los hiperparámetros para HistGradientBoostingClassifier.

```
# Definir los parámetros a ajustar
param_grid = {
    'C': [0.001, 0.01, 0.1, 1, 10, 100],
    'penalty': ['l1', 'l2']
}
```

Figura 7. Valores de los hiperparámetros para LogisticRegression.

```
param_grid_RFC = {
    'n_estimators': [50, 100, 200],
    'max_depth': [3, 5, 7]
}
```

Figura 8. Valores de los hiperparámetros para RandomForestClassifier.

```
param_grid_kmeans = {
    'n_clusters': [2, 3, 4],
    'init': ['k-means++', 'random']
}
```

Figura 9. Valores de los hiperparámetros para KMeans.

```
param_grid_PCA = {
    'n_components': [3, 5, 7],
    'whiten': [True, False]
}
```

Figura 10. Valores de los hiperparámetros para PCA.

Los mejores hiperparámetros se describen a continuación:

Modelo	Hiperparámetro 1	Hiperparámetro 2
HistGradient BoostingClassifier	Learning_rate: 0.2	Max_depth: 7
LogisticRegression	C: 10	Penalty: 'l2'
RandomForestClassifier	Max_depth: 7	N_estimators: 50
KMeans	Init: Random	N_clusters: 2
PCA	N_components:3	Whiten: True

5. Métodos no supervisados y supervisados

5.1. HistGradientBoostingClassifier

```
Accuracy: 93.48232698671526

              precision    recall  f1-score   support

           0       0.93      0.94      0.94     10426
           1       0.94      0.93      0.93     10425

    accuracy                           0.93     20851
   macro avg       0.93      0.93      0.93     20851
weighted avg       0.93      0.93      0.93     20851
```

Figura 9. Reporte para HistGradientBoostingClassifier.

5.2. LogisticRegression

```
Accuracy
81.69871948587598
              precision    recall  f1-score   support

           0       0.88      0.74      0.80     10426
           1       0.77      0.90      0.83     10425

    accuracy                           0.82     20851
   macro avg       0.82      0.82      0.82     20851
weighted avg       0.82      0.82      0.82     20851
```

Figura 10. Reporte para LogisticRegression.

5.3. KMeans

```
              precision    recall  f1-score   support

           0       0.49      0.61      0.54     10426
           1       0.48      0.36      0.41     10425

    accuracy                           0.48     20851
   macro avg       0.48      0.48      0.48     20851
weighted avg       0.48      0.48      0.48     20851

Accuracy
48.34300513164836
```

Figura 11. Reporte para KMeans.

5.4. PCA y RandomForest

```
Mejores hiperparámetros para PCA: {'n_components': 3, 'whiten': True}
Mejores hiperparámetros para RandomForestClassifier: {'max_depth': 7, 'n_estimators': 100}
Accuracy del modelo RandomForestClassifier + PCA: 84.02474701453167
Classification Report del modelo RandomForestClassifier + PCA:
              precision    recall  f1-score   support

           0       0.88      0.79      0.83     10426
           1       0.81      0.89      0.85     10425

    accuracy                           0.84     20851
   macro avg       0.84      0.84      0.84     20851
weighted avg       0.84      0.84      0.84     20851
```

Figura 12. Reporte para PCA + RandomForest.

5.5. KMeans y RandomForestClassifier

```
Mejores hiperparámetros para RandomForestClassifier : {'max_depth': 7, 'n_estimators': 100}
Accuracy del modelo RandomForestClassifier + KMeans: 88.10129010599012
Classification Report del modelo RandomForestClassifier + KMeans:
              precision    recall  f1-score   support

           0       0.91      0.84      0.88     10426
           1       0.85      0.92      0.89     10425

    accuracy                           0.88     20851
   macro avg       0.88      0.88      0.88     20851
weighted avg       0.88      0.88      0.88     20851
```

Figura 13. Reporte para KMeans + RandomForestClassifier.

Modelo	Accuracy	F1-Score
HistGradientBoostingClassifier	93.48%	0.93
LogisticRegression	81.69%	0.81
KMeans	48.43%	0.48
PCA + RandomForest	84.024%	0.84
KMeans + RandomForest	88.102%	0.88

6. Curvas de aprendizaje

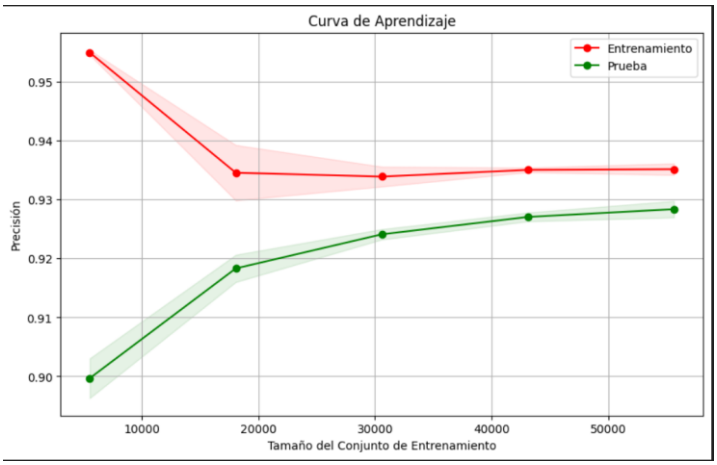

Figura 14. Curva de aprendizaje para HistGradientBoostingClassifier.

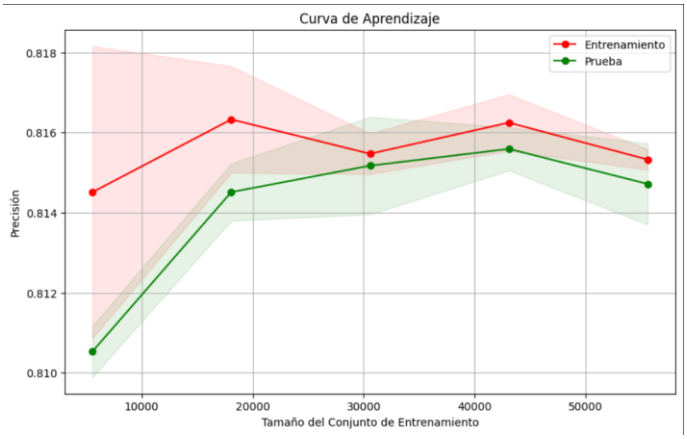

Figura 15. Curva de aprendizaje para LogisticRegression.

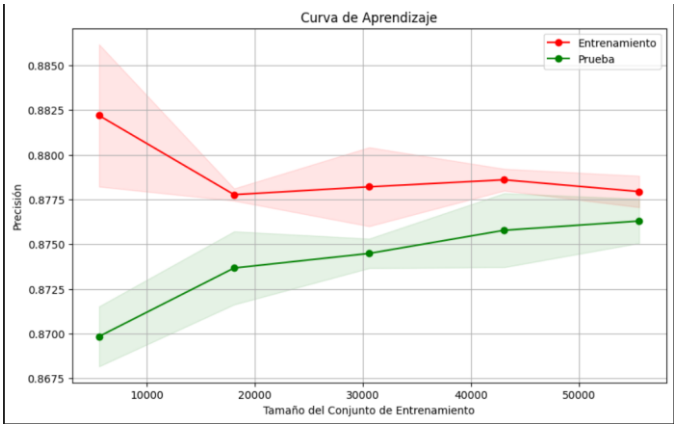

Figura 16. Curva de aprendizaje para RandomForestClassifier.

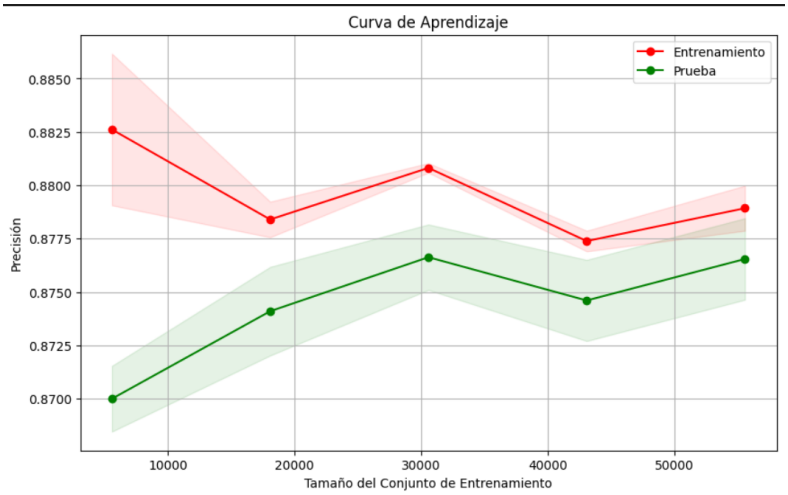

Figura 17. Reporte para KMeans + RandomForestClassifier.

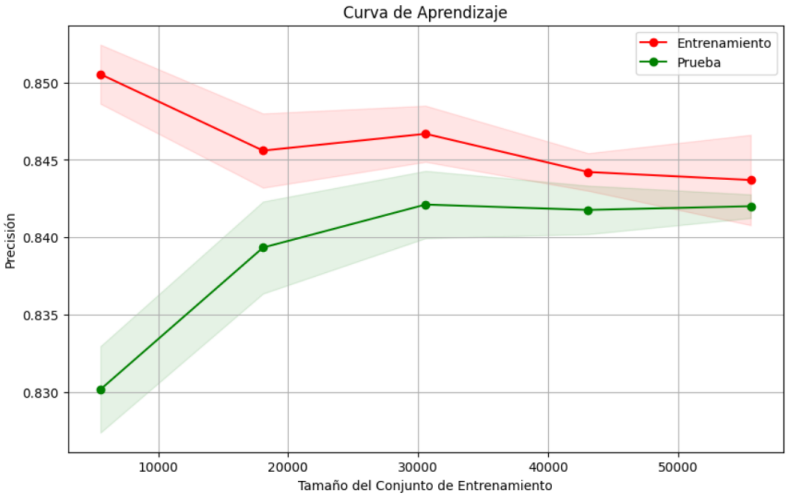

Figura 13. Reporte para PCA + RandomForestClassifier.

7. Retos y consideraciones de despliegue

Para la implementación en producción, seleccionaremos modelos con una precisión superior al 90%, como el HistGradientBoostingClassifier. La integración se hará a través de APIs o puntos finales que recibirán los datos para la predicción. Además, crearemos un dashboard en Power BI que nos permitirá monitorear semanalmente el rendimiento y las métricas del modelo, junto con las curvas de aprendizaje. Este seguimiento semanal nos ayudará a evitar que el modelo funcione por periodos prolongados sin supervisión, previniendo así posibles pronósticos inexactos que podrían impactar negativamente en el presupuesto y las estrategias relacionadas con las reclamaciones de seguros de los clientes.

8. Conclusiones

Después de evaluar diversos modelos predictivos, encontramos que el modelo HistGradientBoosting es el más eficaz para predecir las reclamaciones de seguros de automóviles. Este modelo no solo muestra la precisión más alta, sino que también las gráficas de ROC y las curvas de aprendizaje indican que se ajusta bien a los datos sin caer en sobreajuste. Se observa que mejora su desempeño con la adición de nuevos datos, lo que es esencial para nuestra meta de prever reclamaciones futuras.

El análisis detallado reveló que la preparación de los datos es crucial. Inicialmente, un modelo de regresión logística parecía prometedor, pero un ajuste fino de los hiperparámetros y el manejo del desbalance de clases nos llevó a un modelo más robusto y confiable, listo para ser implementado en un entorno de producción, lo cual es esencial en la industria de seguros.

9. Bibliografía

- Car Insurance Claim Prediction | Kaggle. Retrieved 22 November 2023, from https://www.kaggle.com/datasets/ifteshanajnin/carinsuranceclaimprediction-classification/data