Informe del modelo 1: Predicción de la Gravedad de Accidentes de Tráfico

En este proyecto desarrollamos un sistema de clasificación binaria para estimar la gravedad de accidentes de tráfico a partir de variables estructuradas. Los datos provienen de un registro oficial de la DGT, que incluye información detallada sobre víctimas, condiciones de la vía y factores ambientales. La variable objetivo, GRAVEDAD_ACCIDENTE, codifica cada incidente en dos niveles: “Leve” (sin heridos), “Grave” (heridos graves o muertes).

1. Preparación y Feature Engineering

Limpieza inicial: Eliminamos las columnas de conteo de víctimas (TOTAL_MU24H, TOTAL_HG24H, etc.) para evitar leakage, identificadores y datos de localización precisa (KM, COD_PROVINCIA), así como el año de registro (ANYO), que no aportan valor predictivo directo. Además, aplicamos un transformador personalizado para sustituir los códigos de falta o invalidez (998, 999) por valores nulos ya que algunos modelos podrían tratarlos como extremo numérico, así nos aseguramos que lo entiende como falta de dato.

Imputación y tipado: Las variables numéricas se imputan con ceros y las categóricas con la cadena "missing". A continuación, las categóricas se convierten a tipo category y se codifican mediante one-hot encoding, generando una matriz dispersa. Las matrices dispersas aprovechan el hecho de que la mayoría de los elementos son cero y almacenan solo sus valores distintos de cero, junto a su ubicación (fila y columna), ahorrando memoria.

Selección univariante: Para reducir dimensionalidad antes de modelar, calculamos tres métricas (chi-cuadrado, ANOVA F-test y Mutual Information) sobre las dummies de cada variable. Agrupamos los valores por variable original, promediamos los scores y extraemos las 20 variables con mayor puntuación agregada.

Reducción de dimensión: A partir del conjunto de características seleccionadas, aplicamos TruncatedSVD para preservar el 99 % de la varianza, lo que requirió 46 componentes. Esta técnica es equivalente a PCA pero trabaja directamente con matrices dispersas sin densificarlas, ahorrando así memoria.

2. Entrenamiento de Modelos y Evaluación

Construimos un pipeline que engloba preprocesado, reducción de dimensión y clasificación. Para contrarrestar el desbalance de clases, integramos SMOTEENN (combina oversampling y undersampling) debido al gran desbalanceo de datos (90% leve, 10% grave) justo después del SVD en cada fold y, para XGBoost, calculamos un vector de sample_weight basado en la frecuencia de cada etiqueta. En el resto de modelos pasamos el parámetro ‘balanced’ a class_weight.

Probamos cinco algoritmos: XGBoost, RandomForest, SGDClassifier, LogisticRegression y SVM. Cada uno se ajustó mediante RandomizedSearchCV , optimizando la métrica F1 ponderada y usando validación estratificada (StratifiedKFold), por el desbalanceo, de 5 folds.

3. Selección del Modelo Final

El modelo final elegido es LogisticRegression (Regresión logística) con un F1 de (0.78). Usamos F1 porque es la media armónica entre precisión y sensibilidad, así tenemos una visión más realista que usando exactitud (accuracy) ya que si el modelo predice siempre “leve” puede tener alta exactitud pero cero utilidad.

Esta medida se debe mayoritariamente a la pobre asociación de las variables con la variable objetivo, teniendo solo 7 columnas por encima de 0.1 en V de Cramer (se considera desde 0.10 a 0.30 que el efecto es pequeño) y siendo ‘TIPO_ACCIDENTE’ la más influyente con una puntuación de 0.16. Esto junto al desbalanceo de clases tan marcado provoca que al modelo le cueste predecir correctamente la clase minoritaria, en este caso ‘grave’.

Informe del modelo 2: Predicción de número de accidentes por provincias y mes

En este proyecto desarrollamos un sistema de predicción de número de accidentes de tráfico por provincia para así optimizar la asignación de recursos de emergencia y diseñar campañas de prevención focalizadas. También podría ser utilizado para la concienciación del público general ya que pueden ver en tiempo real el posible número de accidentes reales que ocurrirán. La aplicación sería una interfaz web que permita seleccionar provincia, horizonte y ventana histórica donde los usuarios puedan experimentar y realizar pronósticos personalizados.

1. Preparación y Feature Engineering

Limpieza inicial: Se elimina el identificador de accidente. Aplicamos un transformador personalizado para sustituir los códigos de falta o invalidez (998, 999) por valores nulos ya que algunos modelos podrían tratarlos como extremo numérico, así nos aseguramos que lo entiende como falta de dato.

Imputación y tipado: Las variables numéricas se imputan con ceros y las categóricas con la cadena "missing". A continuación, las categóricas se convierten a tipo category y se codifican mediante one-hot encoding, generando una matriz dispersa.

Agrupación mensual: Se agruparon mensualmente los accidentes por provincia, se realiza el recuento de accidentes y se calcula el promedio de dummies (columnas después de ser codificadas) y de tráfico.

Conversión de fechas: se convierten la columna ‘MES’ en dos componentes trigonométricos (seno y coseno) para conseguir capturar la estacionalidad, evitar discontinuidades y facilitar el aprendizaje del modelo. El año se escala para que el modelo lo use como indicador de tendencia sin comprimir ese efecto en un bucle periódico.

2. Entrenamiento de Modelos y Evaluación

Se seleccionó un modelo LSTM(Long Short-Therm Memory) para esta tarea. Es un tipo especial de red neuronal recurrente (RNN) diseñada para aprender secuencias y recordar información a largo plazo.

Como es un problema de regresión (predecir número de accidentes) no se ha empleado ninguna función de activación en la última capa pues queremos el dato tal como sale de la red. Sí que se utilizan funciones de activación dentro de cada celda: sigmoide (sigmoid) para las puertas, para decidir cuánto pasa (0 a 1) y hiperbólica tangente (tanh) en el contenido de la celda, para mantener los valores entre -1 y 1 y evitar explosiones de gradiente.

La función de error es MSE (Error Cuadrático Medio) ya que penaliza errores grandes al elevarlos al cuadrado, lo que motiva al modelo a ser más preciso con valores extremos o picos en accidentes.