Autor: Ítalo Garleni Rodríguez

Preprocesamiento y Clasificación

Usuario: ItaloGarleniRodriguez

Contenido

Descripción del dataset

El conjunto de datos a utilizar en este proyecto trata de una recopilación de accidentes en España. Sobre cada accidente se describe con 29 variables las cualidades del mismo, con una última variable clasificación que indica el tipo de accidente. De estas 29 variables, 7 son de tipo numérico y el resto son tipo factor:

- *ANIO, MES, HORA, DIASEMANA* → Este grupo de variables indica la fecha y hora en la que se registró el accidente.

- *PROVINCIA, COMUNIDAD_AUTONOMA, ISLA* → Con estas variables se conoce la localización del accidente en el territorio español.

- *TOT_VICTIMAS, TOT_MUERTOS, TOT_HERIDOS_GRAVES, TOT_HERIDOS_LEVES, TOT_VEHICULOS_IMPLICADOS* → Revelan la cantidad de personas, su estado, y los vehículos implicados.

- *ZONA, ZONA_AGRUPADA, CARRETERA, RED_CARRETERA, TIPO_VIA* → Indica el identificador de carretera y describe el tipo.

- *TRAZADO_NO_INTERSEC, TIPO_INTERSEC* → Señala la morfología del tramo donde ocurrió el accidente (curva, recta, intersección, tipo, etc.)

- *ACOND_CALZADA, PRIORIDAD, ACERAS, MEDIDAS_ESPECIALES* → Da detalles sobre isletas, semáforos, pasos de peatones y aceras posibles en los alrededores del accidente.

- *SUPERFICIE_CALZADA, LUMINOSIDAD, FACTORES_ATMOSFERICOS, VISIBILIDAD_RESTRINGIDA, OTRA_CIRCUNSTANCIA* → Establece las posibles circunstancias que pudieron conllevar el accidente, como visibilidad, lluvia u otros.

- *DENSIDAD_CIRCULACION* → Indica la densidad de circulación media de la carretera.

- *TIPO_ACCIDENTE* → Es la variable clase la cual se va a predecir. Señala el tipo de accidente, existiendo seis posibilidades: *Atropello, Colision_Obstaculo, Colision_Vehiculos, Salida_Via, Vuelco* y *Otro*.

Para llevar a cabo el análisis y la generación del modelo de predicción, se posee de un dataset de entrenamiento de 3002 observaciones y uno de prueba de 19998 observaciones. Cabe destacar que el dataset posee numerosos valores perdidos (NA / missing values), los cuales será necesario llevar a cabo un preprocesamiento para tratar esta carencia de información.

Preprocesamiento

Tratamiento de NA's

Como ya se adelantó, en este conjunto de datos existen numerosos datos con valores perdidos, por lo que es necesario aplicar técnicas que solucionen este problema. Existen varias formas de afrontarlo, pero en este estudio se ha optado por la imputación de valores perdidos en los que exista una cantidad razonable de información.

```
NaTrain
NAvalues
"CARRETERA"                "16690"
"ACOND_CALZADA"            "23699"
"PRIORIDAD"                "8122"
"VISIBILIDAD_RESTRINGIDA"  "10685"
"OTRA_CIRCUNSTANCIA"       "3239"
"ACERAS"                   "3149"
"DENSIDAD_CIRCULACION"     "10710"
"MEDIDAS_ESPECIALES"       "8675"
```

```
NaTest
NAvalues
"CARRETERA"                "11088"
"ACOND_CALZADA"            "15709"
"PRIORIDAD"                "5366"
"VISIBILIDAD_RESTRINGIDA"  "7072"
"OTRA_CIRCUNSTANCIA"       "2093"
"ACERAS"                   "2034"
"DENSIDAD_CIRCULACION"     "7103"
"MEDIDAS_ESPECIALES"       "5711"
```

En otras palabras, se aplicará un filtro en el que se eliminarán aquellos ítems con más de un 40% de valores perdidos en sus variables. Esto evitará que se imputen ítems demasiado incompletos, creando así ítems sintéticos. Luego, se aplicará una imputación a cada variable para reducir el tiempo de imputación, excluyendo las variables con NA menos una en cada imputación. Para este proceso se hará uso de la librería MICE, usando el método PMM.

El resultado final de la imputación se guardará en un fichero CSV con el fin de no tener que realizar una imputación cada vez que se quiera generar un modelo sobre los datos. Como cabe esperar, la imputación se realiza tanto sobre el train como el test, por lo que se generarán dos ficheros: "imputedDataTest.csv" y "imputedDataTrain.csv".

Selección de variables

Después de tratar los valores perdidos, se procederá a hacer un análisis de variables para obtener su influencia sobre la variable clase. De este modo, se puede ahorrar recursos a la hora de generar el modelo, y predecir la variable clase sin necesidad de hacer uso de toda la información.

Para ello, se utilizarán las funciones que provee la librería FSelector, en las que se realiza un análisis de pesos en las variables. Usando varias de éstos algoritmos se podrá obtener un resultado más fiable, por lo que se aplicará un total de cuatro métodos, para luego normalizarlos y obtener una puntuación global sobre las variables.

Los modelos utilizados son: Chi-Squared, Entropy based – information gain, Entropy based – Ratio y Entropy based – Symmetrical uncertainty. Con este grupo de algoritmos se obiene el siguiente resultado:

	attr_importance
TOT_VEHICULOS_IMPLICADOS	1.00000000
CARRETERA	0.45068105
ZONA_AGRUPADA	0.37391518
ZONA	0.29174113
TIPO_VIA	0.20366500
TRAZADO_NO_INTERSEC	0.18573114
RED_CARRETERA	0.18445344
PRIORIDAD	0.18075407
ACERAS	0.17754257
TIPO_INTERSEC	0.13300556
PROVINCIA	0.09992361
DENSIDAD_CIRCULACION	0.09669707
TOT_HERIDOS_LEVES	0.09162691
SUPERFICIE_CALZADA	0.08811986
OTRA_CIRCUNSTANCIA	0.08796713
LUMINOSIDAD	0.08455863
COMUNIDAD_AUTONOMA	0.08392319
VISIBILIDAD_RESTRINGIDA	0.08217781
ACOND_CALZADA	0.07329179
TOT_VICTIMAS	0.07129529
FACTORES_ATMOSFERICOS	0.05573153
HORA	0.05411280
TOT_HERIDOS_GRAVES	0.05097586
MEDIDAS_ESPECIALES	0.04373360
DIASEMANA	0.03973797
TOT_MUERTOS	0.03598215
MES	0.01444897
ISLA	0.01434050
ANIO	0.00000000

Sin embargo, las variables CARRETERA y PRIORIDAD suelen dar algunos problemas en ciertos algortimos, ya que registran numerosos valores perdidos y, por tanto, pueden estar sesgadas o poseer un error acumulado. Por tanto, en futuras clasificaciones se descartará, comprobando que ayuda incluso a mejorar el Accuracy.

Clasificación

En cuanto a la clasificación y envío al servidor de KAGGLE para su evaluación, se han realizado un total de once predicciones sobre los datos preprocesados. En ellas se han utilizado cuatro tipos de modelo: *Random Forest*, *Conditional Inference Classification Tree*, *Bagging* y *Boosting*.

Nº Test	**Información del clasificador**	**Accuracy 10CFV**	**Accuracy KAGGLE (public)**	**Accuracy KAGGLE (private)**
1	RandomForest, 100 trees, TOT_VEHICULOS_IMPLICADOS * ZONA_AGRUPADA * ZONA * TIPO_VIA	0.8203448	0.81930	0.81541
2	RandomForest, 2000 trees, TOT_VEHICULOS_IMPLICADOS * ZONA_AGRUPADA * ZONA * TIPO_VIA	0.8202781	0.81930	0.81541
3	RandomForest, 500 trees, TOT_VEHICULOS_IMPLICADOS * ZONA_AGRUPADA * ZONA * TIPO_VIA * TRAZADO_NO_INTERSEC * RED_CARRETERA * ACERA	0.8216112	0.82158	0.81632
4	RandomForest, 500 trees, - CARRETERA - ISLA - MEDIDAS_ESPECIALES	(takes too long)	0.82158	0.81632
5	RandomForest, 50 trees, - CARRETERA - ISLA - MEDIDAS_ESPECIALES	0.8281116	0.82859	0.82584
6	CassificationTree, TOT_VEHICULOS_IMPLICADOS * ZONA_AGRUPADA * ZONA * TIPO_VIA * TRAZADO_NO_INTERSEC	0.8214445	0.81891	0.81521
7	ClassificationTree, TOT_VEHICULOS_IMPLICADOS * ZONA_AGRUPADA*ZONA *TIPO_VIA *TRAZADO_NO_INTERSEC* RED_CARRETERA*ACERAS *PRIORIDAD *TIPO_INTERSEC *PROVINCIA * TOT_HERIDOS_LEVES * SUPERFICIE_CALZADA	0.8250111	0.82573	0.82078
8	Boosting, mfinal = 2, maxdepth = 10, TOT_VEHICULOS_IMPLICADOS + ZONA_AGRUPADA + ZONA + TIPO_VIA + TRAZADO_NO_INTERSEC + RED_CARRETERA	0.8191449	0.81891	0.81511

9	ClassificationTree, TOT_VEHICULOS_IMPLICADOS * ZONA_AGRUPADA * ZONA * TIPO_VIA * TRAZADO_NO_INTERSEC * RED_CARRETERA	0.8214445	0.81891	0.81521
10	RandomForest, 500 trees, TOT_VEHICULOS_IMPLICADOS * ZONA_AGRUPADA * ZONA * TIPO_VIA * TRAZADO_NO_INTERSEC * RED_CARRETERA	0.8209446	0.81930	0.81571
11	Bagging, Maxdepth = 5, minsplit = 15, TOT_VEHICULOS_IMPLICADOS + ZONA_AGRUPADA + ZONA + TIPO_VIA + TRAZADO_NO_INTERSEC + RED_CARRETERA	0.8191449	0.81891	0.81511

El proceso a seguir ha sido el de combinación de los modelos de predicción más conocidos, con la selección de variables que anteriormente se nombró. En algunos casos se utilizaba únicamente las variables con mejor puntuación para generar el modelo. Sin embargo, en otras se pretendía utilizar el dataset completo. Esto último no era posible, ya que las variables CARRETERA, ISLA y MEDIDAS_ESPECIALES daban error y no permitían el análisis. Por tanto, dichos modelos que excluyan dichas variables, simplemente pretendían probar el modelo con el máximo número de variables posibles.

Tras probar diferentes modelos, se comprobó que el mejor que funcionaba era el Random Forest. Sin embargo, el aumento de árboles no beneficia en gran medida a la tasa de acierto, sino que generan un sobreajuste a los datos de Train. Es por esto que el mejor modelo obtenido era un Random Forest con 50 árboles, aunque dicho modelo no fuera el que mejor Accuracy tenía a priori.

Funcionamiento de los R-Scripts

Como punto final, en este apartado se describirá el funcionamiento de los scripts en R que se han utilizado para la realización de este análisis. Éstos se dividen en diferentes módulos, que se reparten las tareas, y una función principal denominada **"analyzer.R"**. De este modo se podrán reciclar los módulos para futuros trabajos de análisis, y dando la posibilidad de modificar módulos sin que esto afecte a los otros.

En el script principal se puede encontrar el recorrido que se ha aplicado, paso a paso, para poder generar los modelos de clasificación. Además, cada modelo de clasificación se encuentra en un módulo, donde, después de una pequeña preparación de variables de entorno, aplica aun 10cfv para calcular un Accuracy relativamente fiable a través de la función getAccuracy(errors,hits) declarada en "CrossFoldSetup.R".

Por último, para poder obtener el mejor modelo que se ha encontrado a través de este proceso no es necesario ejecutar el script completo (con imputación). Basta con comenzar a ejecutar desde la sección { ## Classification Algorithms ## }, la cual lee los ficheros previamente nombrados donde se encuentran los datasets imputados y genera los modelos.

Esta sección posee un limpiado de espacio de trabajo, más un pequeño preparativo del mismo. Luego, de esto, basta con escoger el Test que se quiere reproducir, para finalmente en la última línea de código generar el fichero de predicción para KAGGLE "kagglePrediction.csv".