PREDICTOR DE ACCIDENTES DE TRÁFICO EN MADRID

PREDICCIÓN DE SERIES TEMPORALES

Ana García Palomares
anagarpal123@gmail.com

ÍNDICE

1. EL BIG DATA EN LA PREDICCIÓN DE LOS ACCIDENTES DE TRÁFICO

2. DESCRIPCIÓN DE LOS DATASETS DE ACCIDENTES
   - ORIGEN DE LOS DATOS
   - ESTRUCTURA DE LOS DATOS

3. METODOLOGÍA
   - LIMPIEZA DE DATOS Y DATA ENGINEERING
   - ANÁLISIS EXPLORATORIO DE LA SERIE TEMPORAL
     - ANÁLISIS GRÁFICO
     - ANÁLISIS ESTADÍSTICO DE LAS SERIES TEMPORALES - ESTACIONARIEDAD
   - MODELOS PREDICTIVOS
     - MODELO ESTADÍSTICO TRADICIONAL CON COMPONENTE ESTACIONAL: SARIMA
     - MODELO ESTADÍSTICO BAYESIANO: PROPHET
     - DEEP LEARNING: REDES NEURONALES

4. CONCLUSIONES

5. MANUAL DEL FRONTED

6. REPOSITORIO EN GITHUB

1. EL BIG DATA EN LA PREDICCIÓN DE LOS ACCIDENTES DE TRÁFICO

Es de actualidad el uso del Big Data para la predicción de accidentes gracias a la última campaña de la DGT, donde se intenta luchar contra los accidentes mortales que acontecerán en los desplazamientos de Semana Santa del 2022. Con el lema *"Hagamos que el Big Data se equivoque"*, la organización intenta vencer a los algoritmos que predicen un total de 36 muertes para dicho periodo y los perfiles de las personas que estarán involucradas (link a la website de la campaña *https://los36.dgt.es/).*

No es la primera vez que esta ciencia es utilizada para dicho propósito, la seguridad vial predictiva parte de la premisa de cómo de preciso podemos predecir un accidente de tráfico y modificar los principales factores de riesgo a través del estudio del comportamiento de los conductores: el Big Data guarda y recopila toda la información sobre las carreteras, y la IA predice todas las consecuencias y como las curvas de accidentes evolucionarían si modificamos algunas de las variables involucradas. La aplicación de estos métodos no ha hecho nada más que empezar a demostrar su poder e incrementa su importancia si empezamos a hablar de temas de actualidad como son el vehículo autónomo o las grandes ciudades y la eliminación de los atascos.

Todo esto es posible debido al carácter estacional que recoge la información del tráfico, y por lo tanto, de los accidentes de tráfico: las variables involucradas registran fluctuaciones regulares y cambios a lo largo del año, lo cual facilita las predicciones y los análisis basados en el tiempo. Por estos motivos, los accidentes de tráfico es un buen campo para la aplicación de técnicas de machine learning, las cuales nos arrojaran patrones y predicciones.

PREDICTOR DE ACCIDENTES DE TRÁFICO EN MADRID

Este informe describe el desarrollo de la aplicación del PREDICTOR DE ACCIDENTES DE TRÁFICO EN MADRID. Dicho predictor pronosticará el número de accidentes que acontecerán en un periodo de tiempo en Madrid, y para ello nos hemos basado en la información histórica de Madrid recogida por una entidad pública.

La aplicación tendrá dos puntos básicos:

- Predicción del número de accidentes
- Análisis exploratorio de los datos históricos

El cliente final de esta herramienta será una entidad que dé asistencia en caso de accidentes de coche, como puede ser la Policía Local o servicio de grúas de una empresa de seguros. Sabiendo de antemano la demanda de recursos, les permitirá dar un mejor servicio a sus clientes al mejorar aspectos tales como:

- Distribución de los recursos a lo largo de la ciudad
- Decrementar el tiempo de respuesta
- Tener suficientes recursos disponibles
- Tener servicios de apoyo con externos

2. DESCRIPCIÓN DE LOS DATASETS DE ACCIDENTES

ORIGEN DE LOS DATOS

Este trabajo se ha basado en la información recogida en la web **"*Portal de Datos Abiertos*"** (*https://datos.madrid.es/portal/site/egob*), una iniciativa del Ayuntamiento de Madrid donde se recopilan datos brutos sobre entidades de gestión pública, con el objetivo de dar a la población información para desarrollar aplicaciones, estudios y análisis.

El **PREDICTOR DE ACCIDENTES DE TRÁFICO EN MADRID** se centra en el periodo 2019-2021. Aunque la información recogida en dicho portal abarca un mayor periodo, nosotros solo nos hemos centrado en estos años ya que para los años anteriores hay un cambio de criterio en la estructura de los datos.

La persona responsable del dataset es la Policía Local y el equipo editor del mismo es el Ayuntamiento de Madrid. La frecuencia de actualización es mensual.

El nombre del conjunto de datos es **"Accidentes de tráfico de la Ciudad de Madrid"**, y su descripción es: "accidentes en la ciudad de Madrid con víctimas y/o daños a la propiedad pública, recopiladas por la Policía Local".

La información esta almacenada anualmente, por lo que hay un archivo por año de análisis. El portal te ofrece descargarte los datos tanto en formato Excel como CSV.

Link a los datasets:
https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=7c2843010d9c3610VgnVCM2000001f4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default

ESTRUCTURA DE LOS DATOS

Las bases de datos registran **una línea por persona involucrada** en el accidente, por lo que para un mismo accidente podremos encontrar varias líneas. Aquellas líneas que se refieren a un mismo accidente van a coincidir en la variable "Nº EXPEDIENTE" y en aquellas variables que describen el accidente, pero serán distintas en las variables que describen a las personas involucradas.

La descripción de los accidentes y de las personas involucradas están basados en los siguientes campos:

VARIABLE	DESCRIPCIÓN	CATEGORIAS
Nº expediente	AAAASNNNNNN, donde: AAAA: año del accidente S: expediente con acc. NNNNNN: nº de serie por año	-
Fecha	Fecha con formato dd/mm/aaaa	-
Hora	La hora tiene rangos de una hora	-

Localización	Calle 1 – calle 2 (cruce) o calle	-
Número	Nº calle	-
Código distrito	Código distrito	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
Distrito	Nombre distrito	Centro, Carabanchel, Latina, Usera, Moncloa -Aravaca, Moratalaz, Salamanca, Villa de Vallecas, Villaverde, Chamberí, Chamartín, Hortaleza, Ciudad Lineal, Retiro, Fuencarral -El Pardo, Vicálvaro, Puente de Vallecas, Barajar, Arganzuela, Tetuán, San Blas - Canillejas
Tipo de accidente	-	Alcance, atropello a animal, atropello a persona, caída, choque contra obstáculo fijo, colisión frontal, colisión fronto-lateral, colisión lateral, colisión múltiple, despeñamiento, solo salida de la vía, vuelco y otras causas
Estado meteorológico	Condiciones atmosféricas en el momento del accidente	Despejado, granizando, lluvia intense, lluvia débil, nevando, nublado o se desconoce
Tipo vehículo	Tipo de vehículo	Las categorías se analizarán en la limpieza de datos
Tipo persona	-	Conductor, peatón o pasajero
Rango edad	Rango de 5 años en la edad de la persona	-
Sexo	-	Hombre, mujer o Desconocido
Código lesividad	-	1, 2, 3, 4, 5, 6, 7, 14, 77 o blanco
Tipo de lesividad	-	01 - Atención en urgencias sin posterior ingreso LEVE 02 - Ingreso inferior o igual a 24 horas LEVE 03 - Ingreso superior a 24 horas GRAVE 04 - Fallecido 24 horas FALLECIDO 05 - Asistencia sanitaria ambulatoria con posterioridad LEVE 06 - Asistencia sanitaria inmediata en centro de salud o mutua LEVE 07 - Asistencia sanitaria sólo en el lugar del accidente LEVE 14 o En blanco - Sin asistencia sanitaria 77 - Se desconoce
Coordinada X UTM	Coordinada UTM X	-
Coordinada Y UTM	Coordinada UTM Y	-
Positividad alcohol	-	N, S
Positividad droga	-	0, 1

Link a la descripción del dataset:
https://datos.madrid.es/FWProjects/egob/Catalogo/Seguridad/Ficheros/Estructura_DS_Accidentes_trafico_desde_2019.pdf

3. METODOLOGÍA

LIMPIEZA DE DATOS Y DATA ENGINEERING

VARIABLE	TIPO INFO.	NATURALEZA	ELIMINAR?	Desconocido/ Otro CATEGORIA	Nans Nulls	QUE HACEMOS CON Nulls/ Nans?	INCLUIDO EN CSV
num_expediente	Texto	Descriptivo (unico)	-	-	-	-	accidentes.csv accidentes_total.csv
fecha	Datetime	Fecha o tiempo	-	-	-	-	accidentes.csv accidentes_total.csv
hora	Datetime	Fecha o tiempo	-	-	-	-	accidentes.csv accidentes_total.csv
localizacion	Texto	Descriptivo (unico)	**No extrapolable**	-	-	-	-
numero	Numero	Descriptivo (unico)	**No extrapolable**	-	✓	-	-
cod_distrito	Numero	Categoria	-	-	✓	**Dropnan**	codigo_distrito.csv accidentes.csv accidentes_total.csv
distrito	Texto	Categoria	**Duplicado con 'cod_distrito'**	-	✓	-	codigo_distrito.csv
tipo_accidente	Texto	Categoria	-	"Otros"	✓	Nans a "Otros"	accidentes.csv accidentes_total.csv
estado_meteorologico	Texto	Categoria	-	"Se desconoce"	✓	Nans a "Se desconoce"	accidentes.csv accidentes_total.csv
tipo_vehiculo	Texto	Categoria	-	"Sin especificar"	✓	Nans a "Sin especificar" Eliminar categorias no coincidentes	accidentes_total.csv
tipo_persona	Texto	Categoria	-	-	✓	**Dropnan**	accidentes_total.csv
rango_edad	Rango numero	Categoria	-	"Desconocido"	-	-	accidentes_total.csv
sexo	Texto	Categoria	-	"Desconocido"	-	-	accidentes_total.csv
cod_lesividad	Numero	Categoria	-	"77"	✓	Nan es categoria, completar con 14	accidentes_total.csv codigo_lesividad.csv
tipo_lesividad	Texto	Categoria	**Duplicado con 'cod_lesividad'**	"Se desconoce"	✓	-	codigo_lesividad.csv
coordinada_x_utm	Numero	Descriptivo (unico)	-	-	✓	**Dropnan**	accidentes.csv accidentes_total.csv
coordinada_y_utm	Numero	Descriptivo (unico)	-	-	✓	**Dropnan**	accidentes.csv accidentes_total.csv
positiva_alcohol	Texto	Categoria	-	-	✓	**Dropnan**	accidentes_total.csv
positiva_droga	Numero	Categoria	-	-	✓	Nan es categoria, completar con 0	accidentes_total.csv

Como estamos comparando y posteriormente, uniendo tres diferentes bases de datos, la limpieza de datos es un punto importante en el proceso, ya que tenemos que comprobar que todos los ficheros son comparables, al tener las mismas variables y estas a su vez la misma información/categorías.

El proceso de limpieza se resume en los siguientes pasos:

1. Variables
   Chequear que todos los ficheros tienen las mismas categorías (variables) y hacer que todas concuerden
2. Eliminar categorías
   - *Localización* y *numero* son variables no extrapolables, que no aportarán ninguna información en las técnicas de machine learning
   - *Distrito y tipo_lesividad* son redundantes a *código de distrito* y *cod_lesividad:*
     Para futuras referencias, normalizaremos la información y guardaremos la correlación de dichas categorías en los siguientes archivos *cod_distrito.csv* y *cod_lesiv.csv.*
3. Categorías de las variables
   De comprobar que las variables de cada dataset tienen las mismas categorías y el contenido de dichas categorías, tenemos que destacar lo siguiente:
   - Desconocido / Se desconoce / Otros / Sin especificar
     Tipo_accidente, estado_meteorológico, tipo_vehiculo, rango_edad, sexo, cod_lesividad y tipo_lesividad tienen una categoría donde se pueden recoger múltiples opciones. Esto puede dar complicaciones en el machine learning y sus interpretaciones
   - *Tipo_vehiculo*
     Dependiendo del dataset esta variable recoge categorías que no están recogidas en todos los ficheros. Entendemos este hecho como que en los años donde no se recogen dichos tipos de vehículos, los accidentes acontecidos con ellos no fueron recogidos como accidentes del periodo, por lo que eliminaremos aquellas categorías que no son coincidentes en los tres datasets.
4. Existencia de Nans/Nulls
   El procesado de los Nans/Nulls será distinto dependiendo de las variables:
   - ***Tipo_accidente***: Nans asignados a Otros
   - ***Estado_meteorologico:*** Nans asignados a Se desconoce
   - ***Tipo_vehiculo:*** Nans asignados a Sin especificar
   - ***cod_lesividad***: Nan o celdas en blanco son una categoría, con el mismo significado que la categoría 14, por lo que sustituimos el vacío por 14
   - ***positive_drogas***: Nan o celdas en blanco es una categoría, rellenamos las celdas vacías con "0"

   Eliminar las líneas, al haber pocas Nans, para las siguientes variables: *cod_distrito, tipo_persona, coordinada_x_utm, coordinada_y_utm y positiva_alcohol*
5. Unir todos los datasets en uno
6. Nuevas variables
   - *ano_fecha* → *año*
   - *dia_sem* → día de la semana, categoría
   - *mes* → mes, categoría
   - *num_exp* → número de expedientes (líneas del dataset) con el mismo número de expediente
   - *hora* → hora

- *es_finde* → Variable True o False que confirma si el día es fin de semana. Esto es un hecho relevante ya que el comportamiento de los conductores cambia de los días laborables a los de descanso.
- *es_fest* → Variable True o False que confirma si el día es un festivo en Madrid. Los conductores tienen un comportamiento distinto, incrementándose el número de desplazamientos los días en torno a las vacaciones. Para ello incluiremos un archivo adicional, *calendario.csv*, un listado de los días festivos en Madrid.

7. Estado meteorológico más frecuente

   Sustituiremos el estado meteorológico para aquellos accidentes donde ha sido notificado como "Se desconoce", por el estado meteorológico notificado más frecuente para ese mismo día.

8. Pasar las coordinadas UTM a latitud y longitud
9. Guardar en csv

   En este punto guardaremos el dataset en dos archivos csv's diferentes:

   - *accidentes.csv* – Descripción de los accidentes

     Sabemos por la descripción de los datasets que, para un mismo *num_expediente* tendremos a las personas involucradas en el accidente, por ello hemos creado una nueva variable, *num_exp*. Con esta variable tendremos información de la escala del accidente entendiendo la misma como el número de personas involucradas en el mismo, que será proporcional al número de recursos que necesitaremos que acudan al accidente.

     Como nosotros solo queremos saber el número de accidentes que han pasado en un periodo de tiempo, crearemos un dataset que solo incluya valores únicos de *num_expediente.* Este si mantendrá las variables donde para un mismo num_expediente será coincidente, ya que cada línea describirá un accidente.

     num_expediente
     - fecha
     - hora
     - cod_distrito
     - tipo_accidente
     - estado_meteorologico
     -estado_meteo_mas_freq
     - coordenada_x_utm
     - coordenada_y_utm
     -latitud
     - longitud
     - ano_fecha
     - dia_sem
     - mes
     - hora
     - num_exp
     - es_finde
     - es_fest

   - *accidentes_total.csv* – Accidentes + descripción de las personas involucradas

     Adicional al anterior, se incluyen las siguientes variables:

     - *tipo_vehiculo*
     - *tipo_persona*
     - *rango_edad*
     - *cod_lesividad*
     - *tipo_lesividad*
     - *positiva_alcohol*
     - *positiva_drogas*
     - *sexo*

Al principio de esta sección hemos añadido una tabla con una lista e información detallada de las variables: tipo y naturaleza de la información, eliminación de las variables con información no útil para machine learning, análisis de las categorías y, existencia y procesamiento de Nans/Nulls.
El código completo y las explicaciones del proceso de limpieza están recogidos en el siguiente Jupyter notebook: *ACC_TRAF_limpieza_datos.ipynb.*

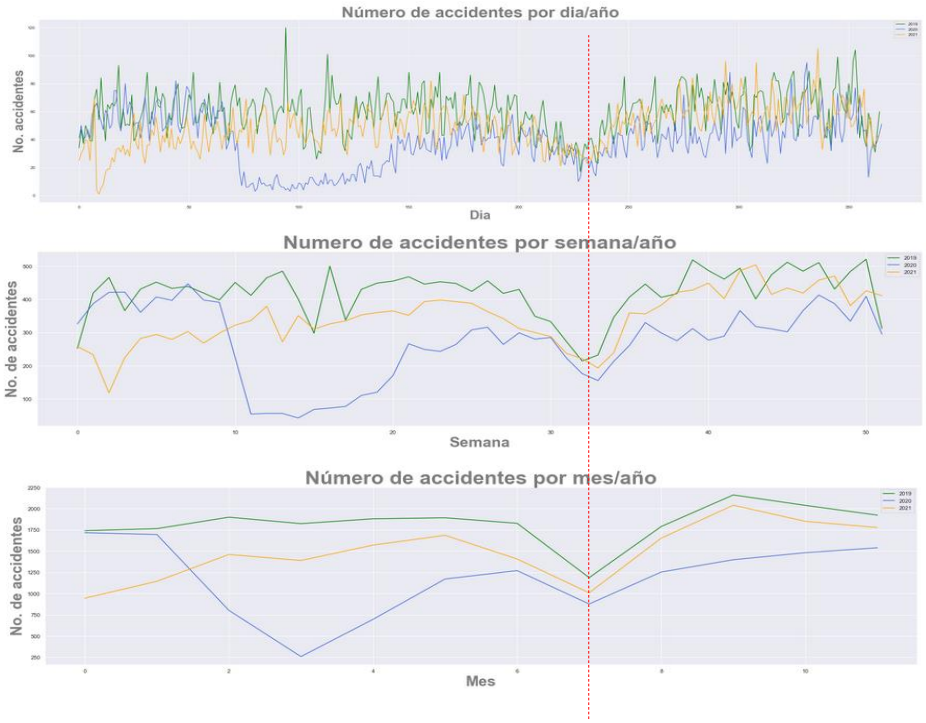
ANÁLISIS EXPLORATORIO DE LA SERIE TEMPORAL

ANÁLISIS GRÁFICO

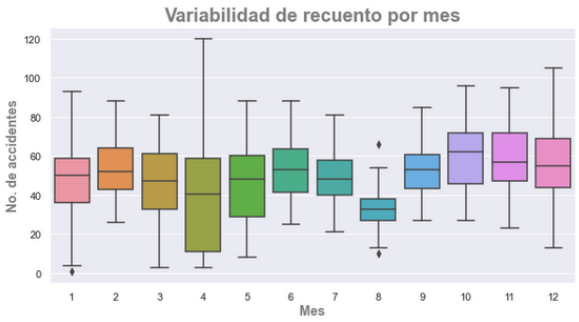
Vamos a hacer un breve análisis gráfico de la frecuencia de siniestros en el tiempo, para poder luego elegir un correcto modelo de forecasting y ajustar el mismo para que tenga en cuenta las características de la muestra.

De la representación lineal por meses del recuento de accidentes podemos apreciar el impacto del **confinamiento**, llegando las observaciones prácticamente a cero, siendo dicho momento un **punto de inflexión de la tendencia**, decrecimiento prácticamente inmediato en el mes de abril 2020 y paulatina recuperación con un crecimiento leve desde dicha fecha hasta final de las observaciones, fechas en las que se alcanzan cifras similares a pre-pandemia.

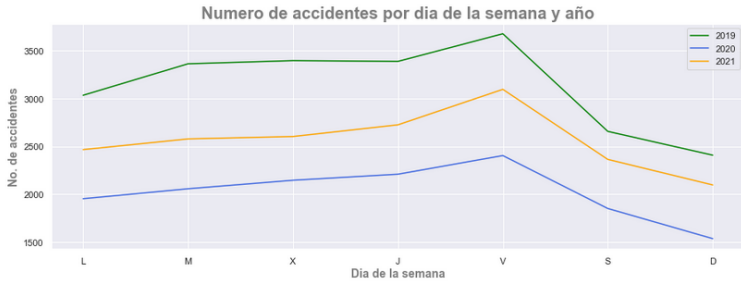
Diferenciando por años se aprecia nuevamente el impacto de la pandemia (2020) y la posterior recuperación que esta sufriendo el recuento. Por otro lado, tambien podemos apreciar el patrón en el recuento de accidentes por meses (estacionalidad anual), siendo los meses de verano aquellos donde menos accidentes se producen y octubre el mes con mayor siniestralidad.

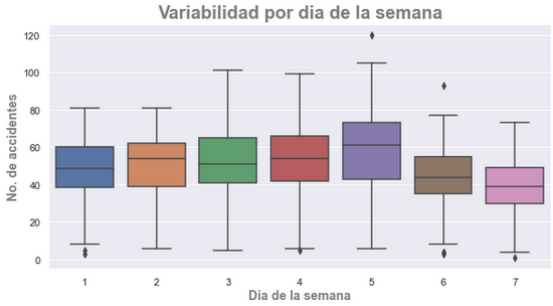


Analizando dicha estacionalidad anual, vemos que el mes con una mayor variabilidad es nuevamente abril 2020 por el confinamiento y sus meses cercanos, siendo aproximadamente constante de +/- 20 para el resto de los meses. Los meses de verano son aquellos que recogen una reducción en la variabilidad, siendo agosto el minimo.

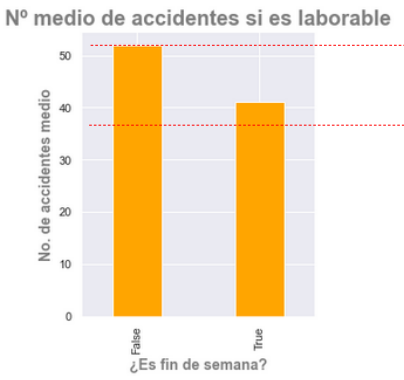


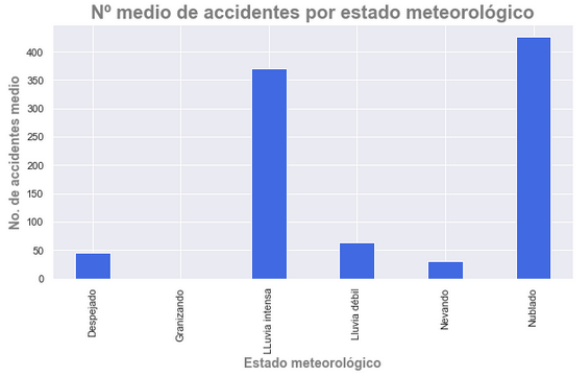
Atendiendo al **dia de la semana** podemos observar una fuerte **estacionalidad semanal**, siendo practicamente **constante a lo largo de la semana**, los **viernes** son el dia con una mayor siniestralidad y una caida en el **fin de semana**, siendo el minimo los domingos. Atendiendo a la variabilidad en el recuento por dia, dichas caracteristicas se repiten, siendo los viernes los dias con mayor dispersión y los sabados y domingos los que menos.





Analizando otras variables exogenas como por ejemplo diferenciando si el día es festivo o no, vemos que el comportamiento de los siniestros tiene un patron incluso más marcado que los fines de semana.





El nº medio de siniestros mayor en función del estado meteorológico es para los dias de nublado, continuado por los dias de lluvia intensa, por lo que concluimos que el estado meteorológico si es un factor determinante en la siniestralidad. Pero no podrá ser usada como variable predictiva, puesto que sería una variable que tambien deberia ser predecida en si misma.

ANÁLISIS ESTADÍSTICO DE LAS SERIES TEMPORALES – ESTACIONARIEDAD

La verificación de la **estacionariedad** (la no dependencia de la serie al tiempo) facilita el modelado de series temporales y es una **suposición subyacente** en muchos métodos de pronóstico de series temporales.

Las características principales de una serie temporal estacionaria son:

- Sus **propiedades estadísticas no varían** con el paso del tiempo (la media, la varianza, y la autocorrelación permanecen constantes a lo largo del tiempo)
- **No tendrá tendencias ni patrones estacionales**

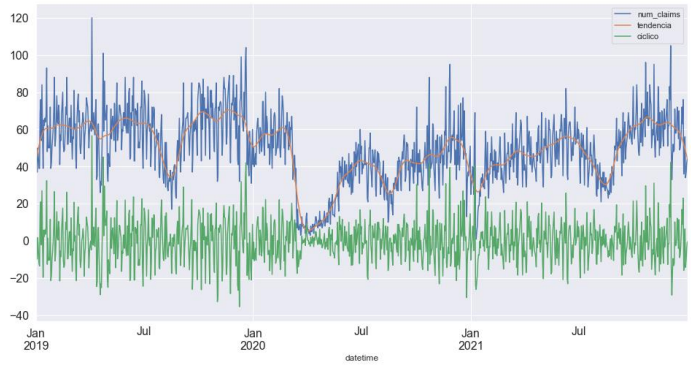
Del análisis exploratorio anterior hemos concluido que la serie recoge tendencia y estacionalidad, vamos a verificar dichas características desde un punto de vista estadístico:

1. **Descomposición**

De la descomposición gráfica de la serie concluimos que la serie muestra:

- **Tendencia**: DC rápido en el periodo del confinamiento, C paulatino desde el mismo hasta final de la muestra, momento en el que se alcanzan niveles similares a la pre-pandemia
- **Estacionalidad**: patrones anuales y semanales

Utilizando el **filtro Hodrick-Prescott**: Método para separar tendencia y componente cíclico



Aunque gráficamente es evidente la tendencia que muestra la serie, aplicamos el **Test Mann Kendall** para verificarlo:

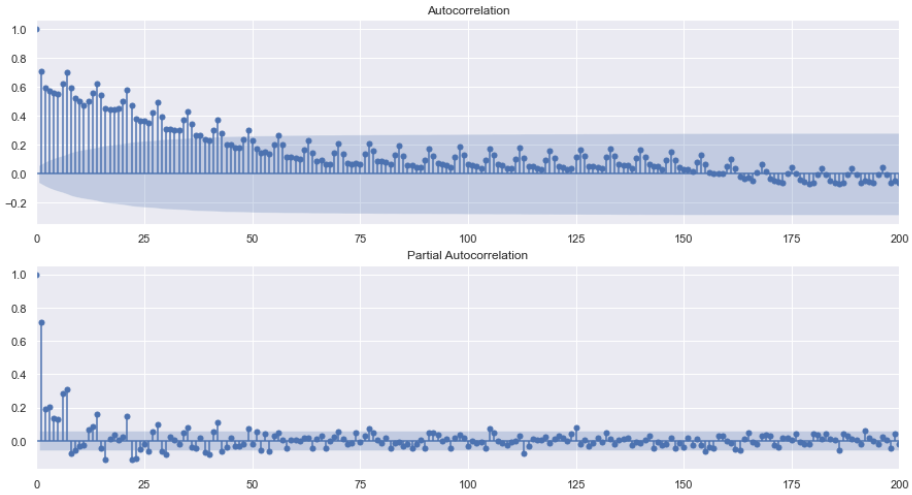
p-value = 2.08 e-06 < 0.05, hay **tendencia** en la serie temporal con un nivel de significancia del 5%

2. Estadísticos constantes

El análisis de los estadísticos por tanto estará afectado por dicha tendencia, siendo excepcional el comportamiento de la muestra en 2020, y similar 2019 y 2021:

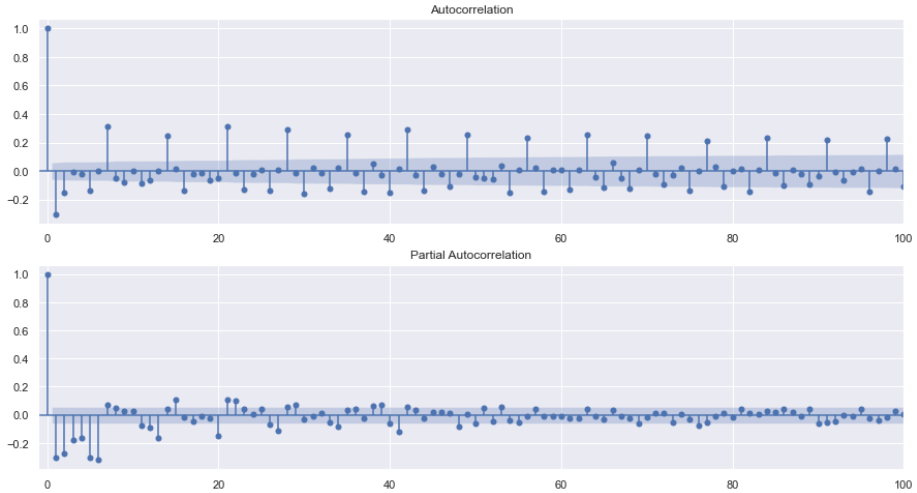
	2019	2020	2021
count	365.000000	366.000000	365.000000
mean	60.063014	38.677596	49.120548
std	15.944749	18.504198	15.104826
min	17.000000	3.000000	1.000000
25%	49.000000	27.000000	38.000000
50%	61.000000	41.000000	50.000000
75%	71.000000	50.000000	59.000000
max	120.000000	95.000000	105.000000

Dicha tendencia también se aprecia en la representación de la autocorrelación, al tener una autocorrelación con tendencia decreciente; también se evidencia la estacionalidad de la muestra.

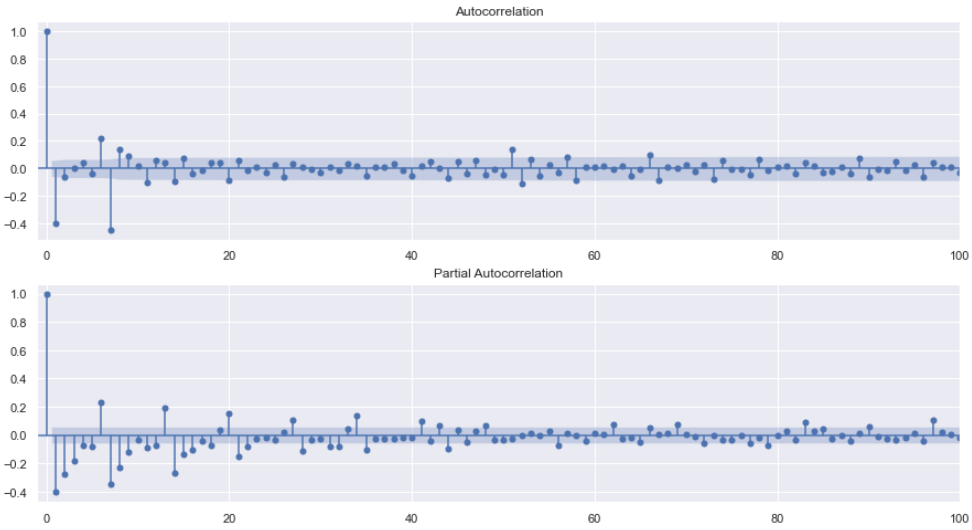


Haciendo una transformación de diferenciación de orden 1, eliminamos la tendencia y de las correlaciones con los rezagos concluimos:

- Dependencia con los días posteriores 1 y 2
- estacionalidad semanal



Haciendo adicionalmente una transformación de diferenciación de orden 7, eliminamos la estacionalidad semanal, pero vemos la dependencia con el día siguiente:



3. Prueba Dickey - Fuller

Aunque la serie muestra tendencia y estacionalidad, vamos a verificar si es estacionaria, ya que es condición imprescindible para la aplicación de modelos predictivos.

Donde con la prueba de Dickey Fuller:
p-valor = 0.032942 < 5% y z*= -3.021416 < z = -2.864237, estamos en la zona de **rechazo de la hipótesis nula (ho: no estacionario)** para un intervalo de confianza del 5%.

CONCLUSIONES:

- ✓ *Tenemos una serie temporal con **tendencia variable** a lo largo del tiempo, y con **estacionalidad semanal y mensual***
- ✓ *Impacto directo del **confinamiento** en la muestra, y corrección paulatina, hasta alcanzar en el final niveles similares a pre-pandemia*
- ✓ *Cambio de comportamiento de la variable dependiendo de si el día en **laboral o festivo***
- ✓ *La serie temporal sin embargo es **estacionaria**, característica corroborada con el test Dickey-Fuller*
- ✓ *La aplicación de los modelos será factible gracias al comportamiento estacionario del mismo, pero deberemos tener en cuenta su tendencia y estacionalidad.*

Para ver una información completa del proceso y el código ver el notebook ACC_TRAF_analisis_descriptivo.ipynb.

MODELOS PREDICTIVOS

En la aplicación de todos los modelos hemos supuesto que se utilizará el mes de diciembre 2021 como set de datos de test, y el resto para el entrenamiento.

Para el pronóstico de la serie temporal vamos a utilizar los siguientes modelos:

1. MODELO ESTADÍSTICO TRADICIONAL CON COMPONENTE ESTACIONAL: SARIMA
2. MODELO ESTADÍSTICO BAYESIANO: PROPHET
3. DEEP LEARNING: REDES NEURONALES

1. MODELO ESTADÍSTICO TRADICIONAL CON COMPONENTE ESTACIONAL: SARIMA

Como ya sabemos nos encontramos con un proceso estocástico de sucesión de variables aleatorias que dependen del parámetro tiempo. Del análisis exploratorio observamos que la serie no es naturalmente estacionaria, por lo que para la aplicación del modelo tendremos que ajustarla por medio de sus parámetros.

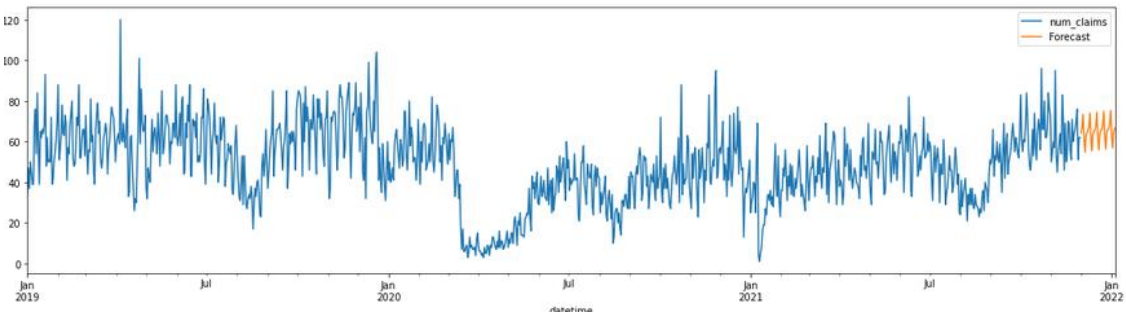
Dentro de las posibilidades de los modelos estadísticos tradicionales hemos elegido el modelo SARIMA, que considera la dependencia de un suceso y su media móvil con los de momentos anteriores (observado en las gráficas ACF y PACF). Este modelo incluye un parámetro adicional para corregir el fuerte componente estacional con frecuencia semanal.

Los parámetros del mismo (SARIMA (p, d, q) (P, D, Q, S)):

- s = 7 → estacionalidad semanal
- d = 1 → eliminación de la tendencia con la diferenciación de orden 1
- AR (p, P) y MA (q, Q) → búsqueda de los parámetros óptimos por medio de un grid

Una vez que hemos encontrado aquellos parámetros que mejor ajustan el modelo a la muestra (aquel con menor AIC), analizamos la distribución de los residuos, para comprobar que efectivamente se ajusta correctamente al sacar las siguientes conclusiones:

- La media de los residuos fluctúa alrededor de cero y tiene una varianza uniforme, por lo que podemos pensar que la predicción no tiene sesgo
- El histograma de la función de densidad sugiere una distribución normal, reforzando la idea de que la media es cero
- Normal Q-Q: los resultados están casi en la línea recta, por lo q concluimos que siguen una tendencia a la normal y el residuo no está skwed
- No hay correlación en los residuos, por lo que no hemos dejado información en ellos, siendo por tanto ruido blanco



Evaluación del modelo:

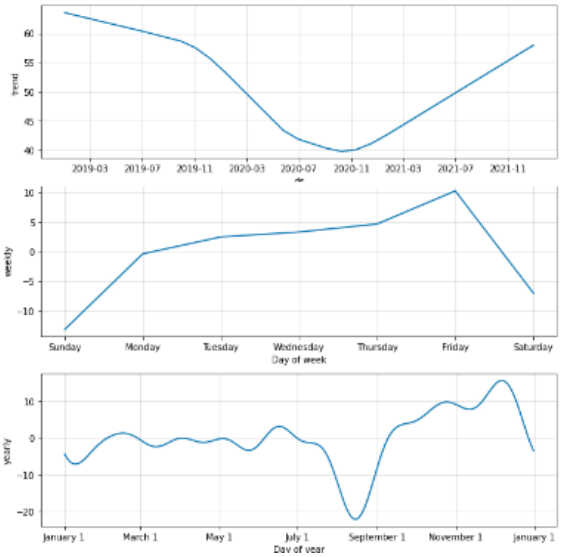
- MSE → 266.43
- RMSE → [17.14916762, 18.5277278 , 22.47101184, 16.3256234 , 15.86005949, 16.64062739, 17.28903432, 17.464337 , 18.84379996, 22.93978473, 16.15282645, 15.76877987, 16.82197174, 17.54454354, 17.67488511, 19.09444458, 23.26390073, 16.26571837, 15.72473979, 16.9873017 , 17.74710437, 17.88299821, 19.3495304 , 23.59125369, 16.38963351, 15.69293636, 17.16233568, 17.95817687, 18.09943072, 19.61118424, 23.92223092]
- MAE → 13.09
- MAPE → 0.2743

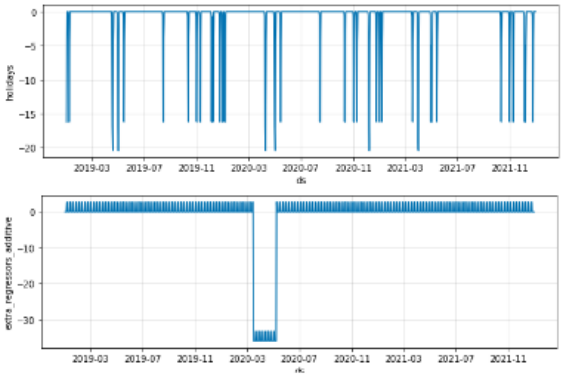
2. MODELO ESTADÍSTICO BAYESIANO: PROPHET

Premisas que hemos tenido en cuenta en la implementación del modelo:

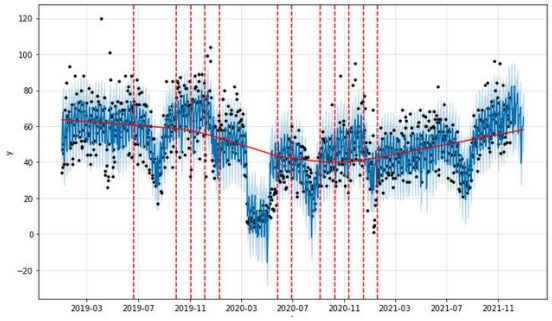
- Incluimos un regresor que indica si el día es **fin de semana o no**, ya que el comportamiento de la variable varía notablemente por dicha condición
- Hemos añadido los **festivos** de la ciudad de Madrid como variable exógena, ya que el comportamiento de los accidentes varía, registrando un comportamiento similar al de los fines de semana. Adicionalmente, hemos incluido una ventana de (-1, 1) días, ya que los festivos generan desplazamientos (y posibles accidentes) no solo en el día mismo, sino también en los días colindantes
- Dado el carácter excepcional del periodo de confinamiento y el impacto sobre la variable hemos generado un regresor que indique el **periodo del confinamiento**

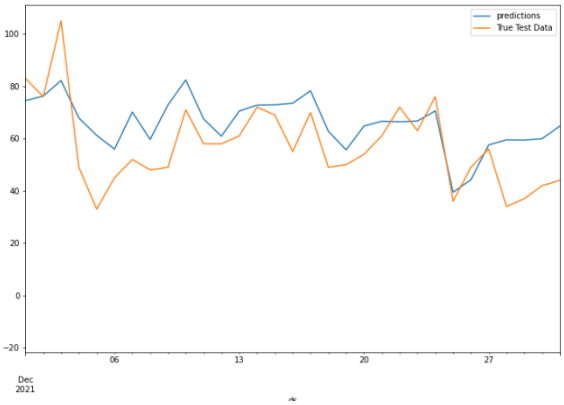
Prophet nos genera una descomposición de la serie temporal por los siguientes elementos: tendencia, estacionalidad semanal y anual, impacto de los festivos e impacto del periodo de confinamiento.





También el modelo genera una gráfica interesante donde muestra los cambios más drásticos de la tendencia en el periodo estudiado.





La representación gráfica de las estimaciones para el último mes con respecto a los datos reales es la siguiente:

La evaluación del modelo la realizamos por medio del **cross-validation**, el cual hemos parametrizado:

- parallel processes de todos los posibles, ya que nuestra muestra no es muy grande
- initial: hemos querido descartar el año de la pandemia
- period: 7 dias
- horizon: periodo de forecasting, diciembre 2021

	horizon	mse	rmse	mae	mape	mdape	smape	coverage
0	4 days	99.923565	9.996178	7.511357	0.253801	0.119839	0.178505	0.824866
1	5 days	120.450342	10.974987	8.570638	0.221389	0.148337	0.190712	0.789439
2	6 days	141.071785	11.877364	9.567845	0.217159	0.171996	0.202156	0.731283
3	7 days	178.977282	13.378239	10.602118	0.277266	0.184658	0.241263	0.691845
4	8 days	185.090300	13.604790	10.936297	0.466557	0.216370	0.278630	0.683155

Intentamos mejorar un poco los resultados al aplicar un grid de búsqueda para optimizar los hiperparámetros de nuestro modelo (changepoint_prior_scale : 0.5, seasonality_prior_scale : 0.01), y aunque los resultados han mejorado, no los consideramos aceptables para un modelo de forecasting.

3. DEEP LEARNING: REDES NEURONALES

Hemos aplicado las redes neuronales desde tres enfoques:

- Considerando únicamente **los accidentes de los 7 días anteriores**
- Considerando las variables **accidentes de los 7 días anteriores, mes y día de la semana**
- Considerando las variables **accidentes de los 7 días anteriores, y las variables mes y día de la semana como embeddings**

Consideraciones comunes en los tres enfoques para que puedan ser comparables:

- ✓ La función de actividad será el número de accidentes los 7 días anteriores, la cual escalaremos al intervalo (-1 , 1) **tangente hiperbólica,** facilitando el funcionamiento de la red
- ✓ Debido a la fuerte estacionalidad de los datos vamos a considerar que la red va a tener **7 neuronas** o vectores
- ✓ Transformamos el dataset a supervisado con una función de **backpropagation**
- ✓ Aplicamos el modelo con **optimizador Adam** y **métrica de pérdida el Mean Absolute Error**
- ✓ **40 EPOCHS**

Para los tres modelos vemos que la métrica de loss desciende y se mantiene estable, lo cual significa que el modelo está aprendiendo; y además no hay overfitting ya que la curva de train y de validate son distintas.

De los tres enfoques en redes neuronales, aquel que recoge mejores resultados es el que toma como variables los accidentes de los 7 días anteriores, y las variables mes y día de la semana como embeddings, ya que es el que obtiene mejores resultados en las medidas analizadas (mse, rmse, mae y mape).

Para ver una información completa del proceso y el código ver el notebook ACC_TRAF_forecasting.ipynb

4. CONCLUSIONES

De todos los modelos aplicados, la predicción la hemos realizado con aquel que mejores resultados nos ha dado en relación a las métricas MSE, RMSE, MAE y MAPE, la **red neuronal con la variable de los 7 días anteriores y embeddings para mes y día de la semana**. Con dicho modelo hemos predecido el número de accidentes que acontecerán la semana siguiente al final de la muestra (primera semana de enero 2022).

La calidad de predicción del modelo no la consideramos aceptable para la subida a un entorno productivo, siendo por tanto este trabajo como la antesala en la búsqueda de un modelo más óptimo que pudiera ser aplicado en el producto para el cliente final.

Actualmente existen aplicaciones y librerías que analizan la muestra y buscan automáticamente el modelo que mejor se ajusta a la misma y que por tanto arrojaría las mejores predicciones (Data Robot, Pycaret), sin embargo, dichas aplicaciones no las hemos aplicado puesto que el objetivo de este trabajo era conocer detalladamente todos los pasos en el análisis de una serie temporal y conocer las consideraciones iniciales en la elección de un modelo. Por otro lado, he utilizado diferentes tipologías de modelos de predicción de series temporales, puesto que quería conocer la mejoría que ha supuesto la evolución de los modelos de predicción.

Otras formas para mejorar la predicción sería ampliar el tamaño de la muestra con un mayor número de años de histórico, sin embargo, no hemos podido ampliar dicha muestra puesto que tal como se enuncia en la fuente de los datos, hubo un cambio de criterio en la recogida de la muestra, por lo que consideramos que los datos podrían no ser equivalentes.

Adicionalmente se podría mejorar el modelo por medio de la inclusión de más variables explicativas, que pudiesen tener un efecto sobre la variable accidentes y estas sean a su vez previas al acontecimiento del evento. Dichas variables (como podría ser el estado meteorológico) no las hemos podido meter, porque su inclusión en el modelo generaría en si mismas otras variables a predecir, complicando por lo tanto el nivel de dificultad del trabajo.

Aunque ha sido mencionado numerosas veces en el informe, es importante destacar el impacto del COVID. Este evento excepcional tiene un impacto directo y de gran tamaño en la muestra, ya que el comportamiento de los conductores cambió no solo en el momento del confinamiento, sino también en la paulatina recuperación posterior de los siniestros.

El impacto del COVID no afecta únicamente a este trabajo, este evento ha distorsionado las series temporales de numerosas variables que cambiaron su tendencia en dicho periodo, siendo por tanto un evento disruptivo que ha afectado a numerosos estudios y que está suponiendo un reto de actualidad en el mundo del machine learning.

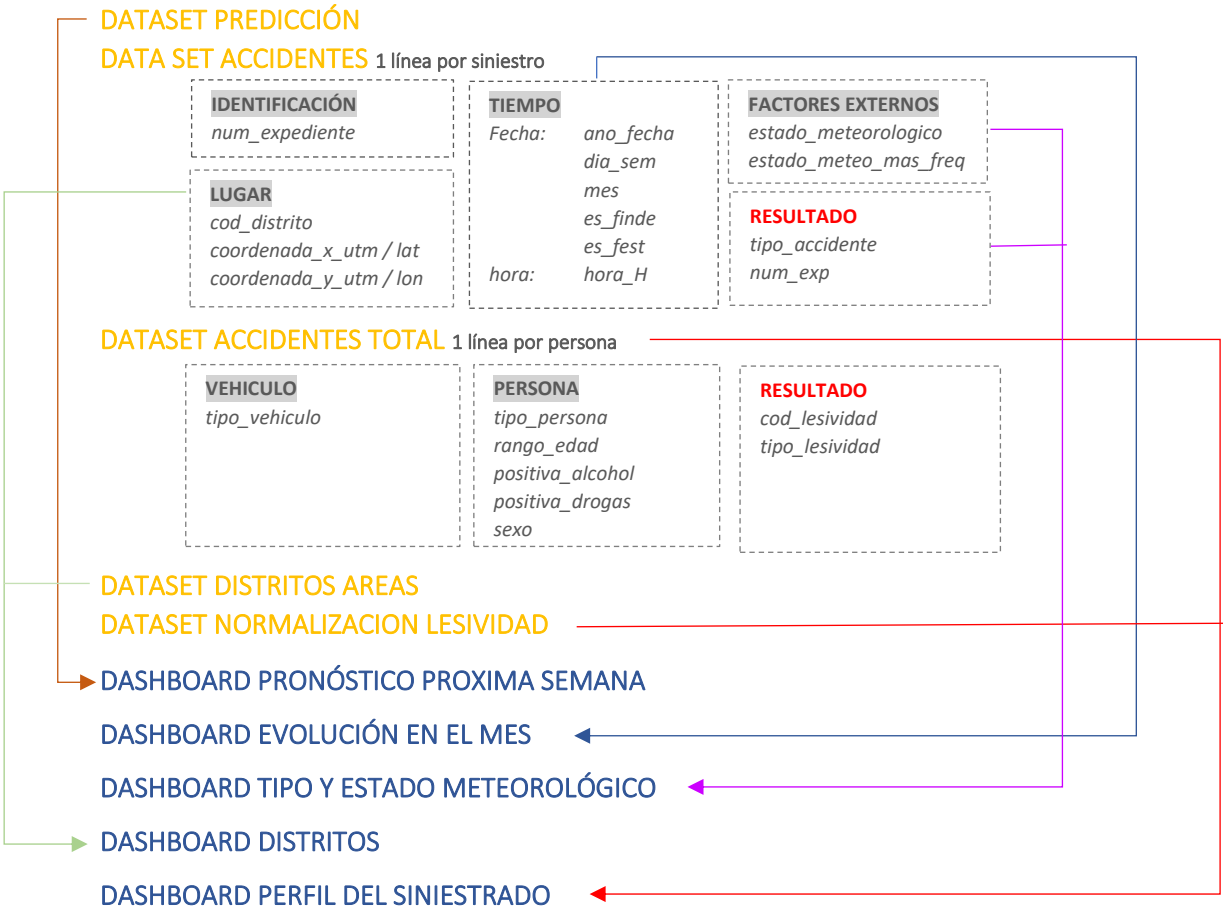
5. MANUAL DEL FRONTED

El ***predictor de accidentes en Madrid*** va a ser una herramienta para un usuario final que necesite cuantificar el número de siniestros que van a acontecer en un corto plazo (una semana en este caso) y tener una idea aproximada de cómo/dónde/a quién… le podrán ocurrir. La información para la predicción ha sido obtenida por medio de modelos predictivos, mientras que la descripción de los mismos podrá ser obtenida por el análisis del histórico.

Todo ello estará plasmado en unos dashboards en la herramienta Tableau Public, que se estructuraran de la siguiente forma:

- PRONÓSTICO PRÓXIMA SEMANA: Predicción de número de accidentes para la próxima semana
- EVOLUCIÓN EN EL MES: Evolución mensual del número de siniestros y estado meteorológico
- TIPO Y ESTADO METEOROLÓGICO: comparativa mensual entre estado meteorológico y tipo de accidente
- DISTRITOS: Número de accidentes y estadísticas por distritos, visualización en mapa
- PERFIL DEL SINIESTRADO

Para entender como están montados los dashboards tenemos la siguiente grafica que hace un resumen de la fuente de los datos y el flujo en su visualización:



Acceso a los dashboards:
https://public.tableau.com/app/profile/ana.garcia.palomares

6. REPOSITORIO EN GITHUB

Toda la información pertinente al proyecto se ha almacenado en el repositorio GitHub: *https://github.com/AnaGarciaPalomares/TFM3.*

El contenido del proyecto se estructura de la siguiente forma:

- Predictor de accidentes de tráfico en Madrid_Memoria

- Requisitos

- NOTEBOOKS: carpeta donde se almacenan los Jupyter notebooks del código, donde estarán:
  1. *ACC_TRAF_limpieza_datos.ipynb*
  2. *ACC_TRAF_analisis_descriptivo.ipynb*
  3. *ACC_TRAF_forecasting.ipynb*

- DATOS_PROCESADOS: de la ejecución del código se generarán los siguientes archivos:
  1. *accidentes.csv*
  2. *accidentes_total.csv*
  3. *cod_distrito.csv*
  4. *cod_lesiv.csv*
  5. *pronostico_embeddings.csv*

- DATOS_BRUTOS: carpeta con los documentos csv descargados de la fuente de información:
  1. *2019_Accidentalidad.csv*
  2. *2020_Accidentalidad.csv*
  3. *2021_Accidentalidad.csv*
  4. *calendario.csv*
  5. *Distritos_map.csv* → archivo de áreas de los distritos usado para el fronted