

- **ARQUITECTURA**

- *diagrama_arquitectura.png*: Diagrama de la arquitectura final del proyecto.

- **BÚSQUEDA Y DATOS**

- *descarga_accidentes_2010-2020.ipynb*: Notebook de Python con la descarga de los dataframes relativos a los accidentes de tráfico en Madrid.
- *descarga_estado_meteorologico_2010-2020_aemet_api.ipynb*: Notebook de Python con la descarga de los dataframes relativos al estado meteorológico de AEMET.
- DOCUMENTACIÓN ACCIDENTES:
 - *accidentes_trafico_2010-2018.pdf*: Documentación relativa a la base de datos sobre los accidentes de tráfico entre los años 2010 y 2018.
 - *accidentes_trafico_2019-2020.pdf*: Documentación relativa a la base de datos sobre los accidentes de tráfico entre los años 2019 y 2020.

- **LIMPIEZA DE DATOS**

- *eficiencia_dask.ipynb*: Notebook de Python con un experimento para medir la eficiencia del uso de Dask en nuestro proyecto.
- *accidentes_data_cleaning_paralelo.ipynb*: Notebook de Python con el data-cleaning del dataframe con los accidentes de tráfico en Madrid.
- *estado_meteorologico_data_cleaning_paralelo.ipynb*: Notebook de Python con el data-cleaning del dataframe con el estado meteorológico de AEMET.
- *union_accidentes_estado_meteorologico.ipynb*: Notebook de Python con la unión de los dataframes relativos a los accidentes de tráfico de Madrid y el estado meteorológico de AEMET.
- *procesamiento_paralelo_data_cleaning_accidentes.png*: Task graph del procesamiento en paralelo para la obtención del conjunto de datos de accidentalidad en la ciudad de Madrid.
- *procesamiento_paralelo_data_cleaning_estado_meteorologico.png*: Task graph del procesamiento en paralelo para la obtención del conjunto de datos del estado meteorológico en la ciudad de Madrid.
- *procesamiento_paralelo_data_cleaning_accidentes_estado_meteorologico.png*: Task graph del procesamiento en paralelo para la obtención del conjunto de datos final con la accidentalidad en la ciudad de Madrid y la información meteorológica de dicho día.

- **MODELOS ANALÍTICOS**

- **RANDOM FOREST**

- *Random_Forest_accidentes.r*: Script escrito en R con la creación de un modelo, utilizando la técnica Random Forest, para predecir la lesividad de la persona involucrada en un accidente.

- **XGBOOST**

- *XGBoost_model_to_predict_LESIVIDAD.ipynb*: Notebook en Python con la creación de un modelo, utilizando la técnica Extreme Gradient Boosting para predecir la lesividad de la persona involucrada en un accidente.
 - *matriz_confusion_xgboost.png*: Matriz de confusión sobre el conjunto de datos de prueba del modelo XGBoost implementado.
 - *importancia_variables.png*: Gráfico de barras con el listado de la importancia de las diferentes variables sobre el modelo XGBoost implementado.

- **REGRESIÓN LOGÍSTICA**

- *Regresión_logística_accidentes.r*: Script escrito en R con la creación de un modelo, utilizando la técnica Regresión Logística, con el objetivo de predecir la gravedad de los individuos involucrados en un accidente, dividido en Accidente Grave y Accidente Leve.

- **VISUALIZACION**

- *analisis_exp_r.R*: Script con el análisis exploratorio de las variables numéricas.
 - *edad.pbix*: Dashboard con gráficos relativos a la edad.
 - *lesividad_distribucion.pbix*: Dashboard que contiene el análisis exploratorio de la variable LESIVIDAD y algunos gráficos relativos a las variables relacionadas con las fechas.
 - *mapa.pbix*: Dashboard con el mapa interactivo y la distribución de la variable DISTRITOS.
 - *meteo.pbix*: Dashboard con el análisis exploratorio de las variables cualitativas relativas a la meteorología.
 - *tablas.pbix*: Dashboard con algunas tablas relativas a algunas variables expuestas en el apartado del análisis exploratorio.