Todo el código del proceso en el siguiente repositorio de github

<https://github.com/HernanHGM/duraton/tree/main/geolocalizacion>

Se reciben los datos de los 5 individuos en 5 archivos csv separados. Estos csv provienen directamente de la base de datos de Ornitela. Corresponen a los datos de GPS de los individuos:

IDs

conquista = 192663

Deleitosa = 201255

Gato = 213861

Navilla = 211981

zorita = 201254

Los datos se descargaron el 23 de enero de 2024 y constan de los datos entre el inicio de toma de datos y su final o el 31-12-2023

El proceso de limpieza y enriquecimiento de los datos se hace individualmente para cada individuo.

Proceso de limpieza

* Eliminamos columnas vacías (Irrelevante)
* Renombramos columna device\_id -> ID (Irrelevante)
* Eliminamos los registros cuyas posiciones geográficas (latitud y longitud exclusivamente) sean valores atípicos.
  + Defino rango interquantilico más extremo de lo habitual IQR = Q95-Q5
  + Limite superior valores típicos = Q95 + IQR
  + Limite inferior valores típicos = Q5 – IQR

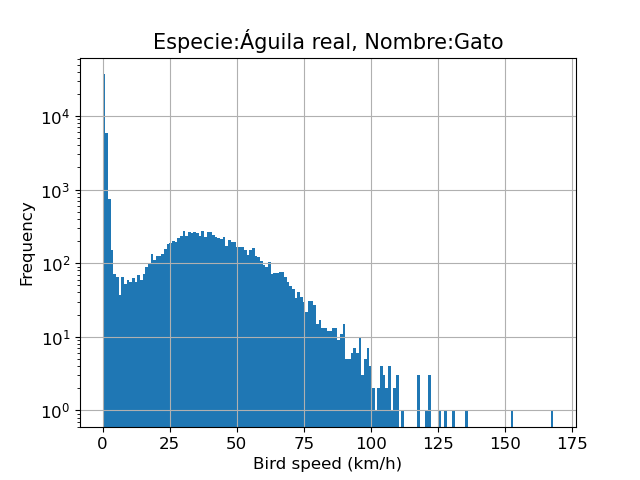
No quería eliminar posiciones atípicas de los individuos quería elimiar posiciones incorrectas. Se daba en repetidas ocasiones que había datos en la posición (0,0) que está en el golfo de guinea aproximadamente. La verdad que podía haber filtrado estos datos de maneras mucho más sencillas, pero bueno es lo que hay.

* Seleccionamos solo las alturas>0, medido a nivel del mar (recordemos que Extremadura está ~400m por lo tanto no debería haber nada por debajo de 400

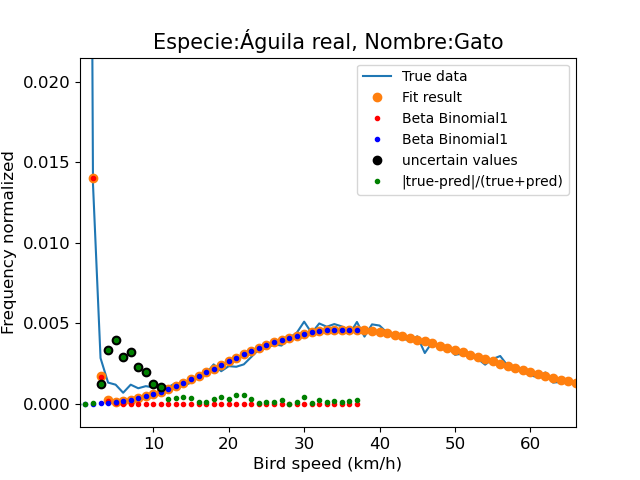
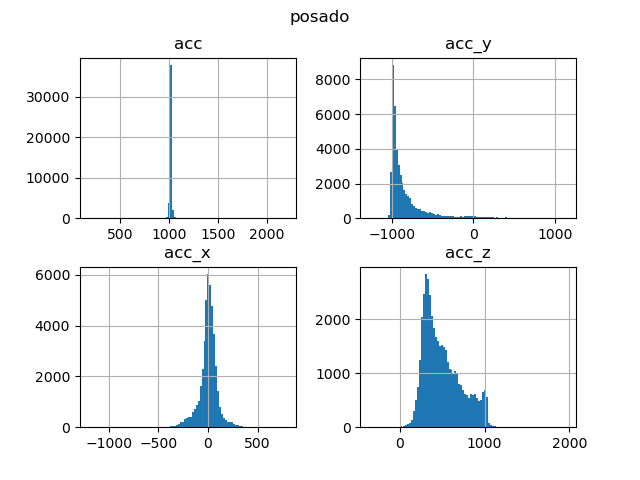
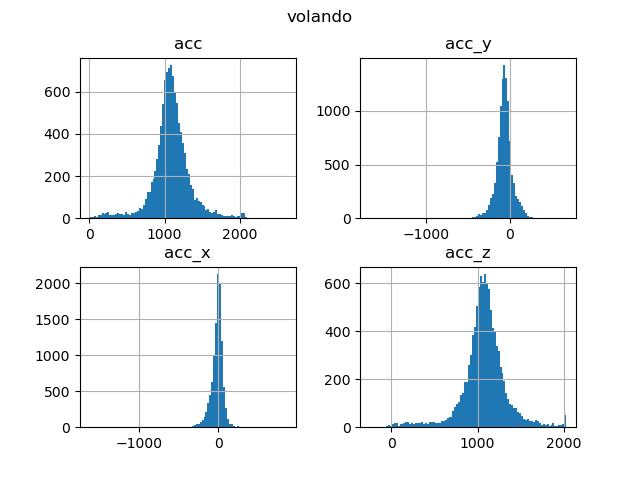
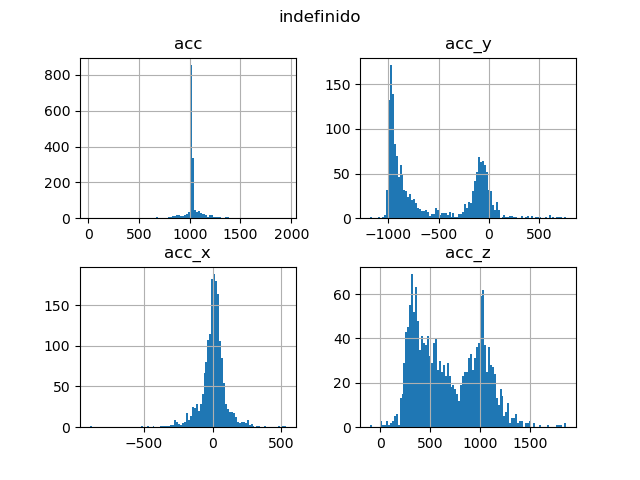
Proceso de enriqucimiento

* Se añade hora a partir de la columna UTC\_datetime. Formato HH:00:00
* Se añade la columna UTC\_datetime\_pre que contiene el registro de UTC\_datetime del registro Previo
* Se calcula la columna time\_step\_s como la resta de UTC\_datetime- UTC\_datetiem\_pre. Nos indica cuantos segundos han pasado entre un dato y el previo
* Se añade el nombre del mes: ‘month\_name’ a partir de la columna ‘UTC\_date’
* Se añade el número del mes dentro del año: ‘month number’ a partir de la columna ‘UTC\_date’
* Se añade el número de la semana respecto al año: 1-52 “week\_number’ a partir de la columna ‘UTC\_date’
* Se añade la latitud, Longitud y Altidu del dato previo: columnas Latitude\_lag Longitude\_lag, Altitude\_lag
* Se calculan las distancias entre cada posición y su posición previa
  + 1 se calculan los incrementos en latidudes longitudes y altitudes restnado el dato de una posición menos el de la posición previa.
  + 2 se pasan las diferencias entre lastitudes y longitudes de manera independiente, de grados a kilómetros multiplicando los ángulos por 2\*piº\*6370km/360º (6370km es una estimación del radio terrestre, soy consciente de que la tierra está ligeramente achatado y el radio cambia, pero dado que la región analizada es muy pequeña, la variación del radio terrestre es despreciable)
  + 3 las diferencias de altura se pasan a km
  + Se calculan la distancia 3D, 2D en el plano y diferencia de alturas como la raíz de las suma de los cuadrados (típico cálculo de distancias entre puntos, no hace falta citar nada)
* Se calculan el módulo del acelerómetro y el magnetrómetro como la raíz cuadrada de la suma de sus cuadrados en los ejes x, y ,z
* Se añade la información relativa a cada pájaro, combinando la tabla de los datos con un tabla auxiliar que contiene su especie y nombre por el ID del emisor
* Se añade el periodo reproductor en el que se encuentra siguiendo la siguiente lógica provista por Victor y ratificada por Andreia a la que Ángel y Elvira por omisión dieron su visto bueno:
  + Perdicera
    - incubación: (15-feb - 31-mar)
    - Cría: (1-abril - 31mayo)
    - Dependencia: (1junio-31julio)
    - Fuera de periodo reproductor: (1agosto-14febrero)
  + Real e Imperial
    - incubación: (15Marzo - 31abril)
    - Cría: (1mayo - 10julio)
    - Dependencia: (11julio-31agosto)
    - Fuera de periodo reproductor: (1septiembre-14marzo)

“””””Calculamos”””” si está posado o en vuelo

* Calculamos el histograma de los datos de velocidad de vuelo ‘speed\_km\_h’ con el numero de bins = int(max(data)-min(data)) esto es para que los datos de velocidad se agrupen en valores enteros y tener un valor del histograma para cada uno de estos valores. Esto será especialmente importante cuando trabajemos con los datos interpolados donde las valocidades tomarán valores decimales debido precisamente a esa interpolación)
* 
* En el histograma (eje y logaritmico) podemos ver que hay dos claros grupos de datos, los datos posado, en los que la velocidad es cero, salvo por el error que puede haber en la toma de datos, y la zona de vuelo con velocidades entorno a 20-70km/h.
* Ajustamos los valores del histograma a la suma de dos funciones betabiniomiales, porque después de estar mirando un tiempo eran las que más se me parecieron y el ajuste vas a ver que es brutal. (No he calculado el R2 pero si lo necesitas si o si lo puedo hacer)
* El ajuste lo hacemos con la función ‘curve\_fit’ del paquete scipy.optimize. Las funciones betabinomiales se calculan con la función betabinom del paquete scipy.stats.
* <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.betabinom.html>
* El proceso de optiminación se se inicializa con los parámetros: p0=[C1=0.80, a1=1.5, b1=1000, C2=0.2, a2=5, b2=15]
  + C1: normalizing constant for the first distribution
  + a1: alfa value for the first betabinomial distribution
  + b1: beta value for the first betabinomial distribution
  + C2: normalizing constant for the second distribution
  + a2: alfa value for the second betabinomial distribution
  + b2: beta value for the second betabinomial distribution

He usado estos parámetros buscando a ojo, no tengo ningún método para explicarlo. No son los valores definitivos de ajuste sino unos valores que permiten al método iterar hasta encontrar el valor óptimo.

* Una vez ajustado el histograma a esta suma de dos funciones betabinomiales se definen dos grupos, aquellos datos cuyo ajuste se considera satisfactorio, y los que no. Los datos ajustados satisfactoriamente, puedes ajustarse a una de las dos betabinomiales, es decir a la de vuelo o a la de posado
* 
* Consideramos que un datos se ajusta satisfactoriamente cuando la diferencia entre el valor real y la media entre el ajuste y el dato real sea menor a 0.4, es decir, si el error en el ajuste es superior al 40% de la media entre el valor real y el ajuste consideramos que no sabemos si está volando o no.
* Además he añadido una función que oblique a que los valores no ajustados satisfactoriamente deban estar seguidos, porque solo queremos analizar esa zona de velocidades donde hay duda entre posado y volando, si están a velocidades menores sabemos que está posado, y a velocidades superiores volando. También aplico el análisis solo a los datos hasta el 90% de los datos acumulados para que el ajuste sea mejor, porque la cola final de datos, estropeaba el ajuste sin aportar calidad al análisis (ya sabemos que no esta posado si viaja a 50km/h) Esta función es pura programación, etá todo subido en un repositorio de github, así que quien lo quiera que lo mire, no tiene valor matemático ni biológico.
* Con los datossatisfactoriamente ajustados vamos a entrenar un modelo de aprendizaje automático al que le daremos como etiquetados los datos de vuelo y posados y de variables ‘acc’ y ‘acc\_y’. Se estudiaron las variables de accelerometro y magnetómetro en sus tre ejes por separado y en módulo, es decir 8 variables. Se seleccionaron finalmente solo el módulo del acelerómetro y el eje y del acelerómetro por ser los que contenían más información con el mínimo de variables
* 
* 
* 
* En la gráfica de los datos indefinidos se observa en acc\_y dos picos claramente diferenciados que coinciden con los picos en acc\_y de los datos posados y volando. En el eje z también se observan esos dos picos pero con menor separación y la correlación con y era alta así que no aportaba nueva información. Acc aporta información en cuanto a la varianza de la distribución.
* Para todos los individuos se entrenaba un modelo propio usando kNeigoburs con 10 vecinos y minmaxscaler para el escalado previo. Se testeaba con los porpios datos de entrenamiento obteniendo en todos los casos una kappa superior a 0.8 usando stratiffeid kFold con 5 splits
* Con el modelo entrenado con los datos ya etiquetados se aplicaba en los datos no etiquetados clasificándolo como posado o vuelo.

Fusión con elevación del terreno

Se combinan los datos de vuelo de los pájaros mediante una unión condicional con las alturas del terreno: <https://dwtkns.com/srtm30m/>

Es una unión condicional porque los datos de terreno son casillas de terreno de un arcosegundo de lado, esto es aproximadamente 30m\*30m, entonces los datos de vuelo si se encuentran dentro de una casilla se les añade la elevación de dicha casilla. Pero no es la típica unión de datos donde unes por varias columnas que contienen los mismos datos (Esto es lo que ralentiza todo)

Una vez añadida la altura del terreno, columna ‘elevation’ calculamos la altura del ave: ‘bird\_altitude’ restando la altitud que nos da el emisor columna: ‘elevation’=‘Altitude\_m’-‘elevation’

Unión con datos meteorológicos

* Se han descargado los datos de la web: <https://www.worldweatheronline.com/weather-api/>
* Se ha usado el código que se enceuntra en : <https://github.com/HernanHGM/duraton/blob/main/geolocalizacion/_dev/download_weather.py>
* Fechas: 1/1/2020-21/1/2024
* Ubicaciones:
  + 'Romangordo', 'Deleitosa','Torrecillas de la Tiesa', 'Herguijuela', 'Conquista de la Sierra', 'Zorita', 'Alcollarín', 'Abertura', 'Campo Lugar', 'Higuera de Albalat', 'Casas de Miravete', 'Almaraz', 'Campillo de Deleitosa', 'Aldeacentenera', 'Madroñera', 'Trujillo', 'Garciaz', 'Santa Cruz de la Sierra'
* La ubicación de cada pueblo se ha descargado de la página <https://www.ayuntamiento-espana.es/>
* Ejemplo: <https://www.ayuntamiento-espana.es/ayuntamiento-aldeacentenera.html>
* Los datos meteorológicos descargados tienen frecuencia horaria como la temperatura, humedad o precipitación o diaria como las horas de sol o la hora de salida y puesta del sol.
* Una vez descargados todos los datos se unen los datos de vuelo a los meteorológicos
  + 1) se asigna a cada registro de vuelo, la población más próxima para asegurar los datos más precisos posibles
  + 2) se une por la clave: fecha, hora y población; los datos de vuelo a los meteorológicos con frecuencia horaria
  + 3) se une por la clave: fecha y población; los datos de vuelo a los meteorológicos con frecuencia diaria
  + 4) Se añade la temperatura a la altura de vuelo del ave calculada como se indica en <https://link.springer.com/chapter/10.1007/978-3-030-13307-8_24>
  + A la temperatura a la altitud del pueblo, restamos 10 grados por cada kilómetro de altura ascendida

Agrupado horario

Una vez acabado todo el proceso de enriquecimiento de los datos procedemos a agruparlos para su poseterior análisis.

* Agrupamos por las siguiente variables
  + ['specie', 'name', 'month\_name', 'month\_number', 'week\_number', 'UTC\_date', 'hour', 'flying\_situation']
  + Clave primaria: 'name', 'UTC\_date', 'hour', 'flying\_situation'
  + El resto son paraa facilitar el posterior análisis
* Calculamos los valores medios de las siguiente variables:
  + mean\_columns = ['Latitude', 'Longitude', 'Altitude\_m',
  + 'mag\_x', 'mag\_y', 'mag\_z', 'mag',
  + 'acc\_x', 'acc\_y', 'acc\_z', 'acc',
  + 'bird\_altitude', 'speed\_km\_h',
  + 'elevation', 'direction\_deg',
  + 'tempC', 'DewPointC',
  + 'windspeedKmph', 'pressure', 'visibility', 'cloudcover', 'precipMM',
  + 'humidity', 'maxtempC', 'mintempC', 'avgtempC', 'sunHour',
  + 'totalSunHour', 'uvIndex']
  + Cuando calculemos las variables de distancias entre individuos para la tabla interpolada, a estas variables les añadiremos las distancias entre aves
* Calculamos la suma de los valores que dependen de la duración de los intervalos y por ende no podríamos calcular la media.
  + sum\_columns = ['time\_step\_s', 'distance\_2D', ‘distance\_height’]
  + suma positiva de la distancia ascendida, es decir solo calculamos cuanta distancia ha ascendido, ignorando la distancia descendida.
* Sobre esta primera tabla agrupada, calculamos el tiempo total del que tenemos datos para cada hora, ignorando si esta volando o no.
* Unimos el primero agrupado con todas las métricas a este segundo con el tiempo total
* Calculamos el porcentaje de tiempo en vuelo y posado dividiendo el tiepo en cada situación entre el tiempo total
* Calculamos la distancia recorrida esa hora normalizada respecto al tiempo que ha estado volando o posado. Realizamos también este proceso de normalización para la altura positiva ascencida, y la altura en conjunto ascendida.

Interpolación

Para poder añadir la distancia a ala que se encuentran unos individuos de otros es necesario tener sus posiciones en los mismos instantes. Para ello interpolamos todos los datos a una frecuencia de 5 minutos. Los datos los interpolamos partiendo de la tabla totalmente enriquecida con elevación del terrenos y datos meteorológicos unidos. El cálculo de posado o volando se repite dado que se trata de variables categóricas, al igual que el periodo de cría. A partir de ahí repetimos el proceso de agrupado y cálculo de métricas definitivas. Hay que tener en cuenta que al tratarse de una interpolación no se mantienen las posiciones reales, sino que son estimaciones. Además en el agrupado horario se añaden como variables las distancias entre individuos.