

**AviClima**

Análisis y predicción de la distribución de aves en función del clima

Índice

1. Descripción del proyecto 3

2. Big Data 4

2.1. Recuperación de datos 4

2.2. Tratamiento de datos 6

2.3. Análisis de datos 8

2.4. Almacenamiento en Hive 12

3. Modelos predictivos 15

3.1. Preparación de los datos 15

3.2. Clustering 16

3.3. Regresión lineal 17

3.4. Red neuronal 18

3.5. One class SVM 19

**Descripción del proyecto**

En el proyecto se analizan los datos de avistamientos de aves en Euskadi y se comparan con los datos climáticos para buscar las posibles relaciones entre ellos.

Para ello se han utilizado datos de las siguientes fuentes:

Laboratorio Cornell de ornitología ([ebird.org](http://www.ebird.org/))

Agencia Estatal de Meteorología ([aemet.es](https://www.aemet.es/))

El proyecto se divide en tres apartados: la recuperación de los datos, el tratamiento y el análisis.

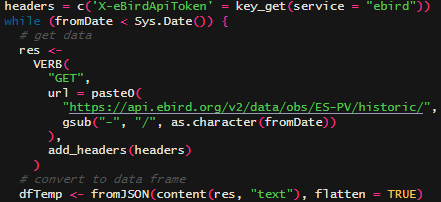
**Big Data**

Recuperación de datos

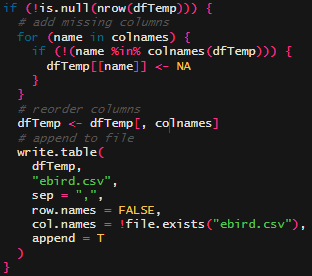
En este paso se han buscado datos en las APIs de ambas fuentes y se han almacenado en archivos csv para su posterior tratamiento y análisis.

En el caso de ebird.org las peticiones a la API se han realizado directamente a través de httr realizando consultas GET a sus URLs.

Por limitaciones de la API ha sido necesario pedir los datos de cada día de forma independiente.

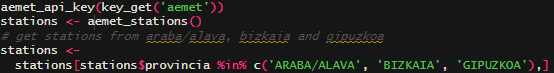


Como no todas las peticiones devuelven el mismo número de columnas es necesario crearlas y ordenarlas para asegurarnos de que el archivo final no tenga errores. Por último, se agregan los datos al archivo.



Para los datos de AEMET disponemos de la librería “climaemet” que nos facilita las peticiones.

Para poder filtrar los datos de Euskadi debemos buscar sus estaciones meteorológicas.



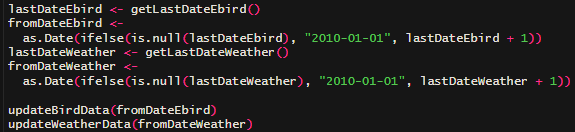
Después pediremos la información meteorológica para las fechas que nos interesen.



Y por último uniremos los datos con la función merge de forma que eliminaremos la información de los territorios que no nos interesan y agregaremos la localización de las predicciones.

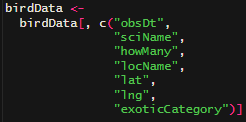


Para ahorrar tiempo en caso de querer actualizar todos estos datos se han creado unas funciones auxiliares que buscaran la última fecha almacenada en los archivos de forma que el código final a ejecutar seria el siguiente.



Tratamiento de datos

A la hora del tratamiento, empezaremos por eliminar columnas innecesarias.



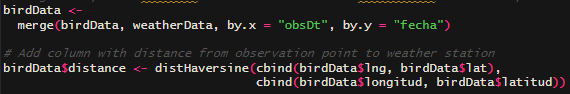
Para cruzar los datos utilizamos la fecha, como esta columna no es única en ninguno de los datasets el resultado de juntarlos multiplicara el número de filas, por cada avistamiento tendremos los datos climáticos de todas las estaciones. Para poder eliminar los resultados innecesarios primero tenemos que añadir identificador a los avistamientos.



Como no todos los resultados meteorológicos incluyen toda la información eliminaremos los que no tengan al menos la temperatura media del día.



Ahora ya podremos unir las tablas, tras lo cual agregaremos una columna con la distancia entre las coordenadas del avistamiento y las de la estación meteorológica, que nos servirá para saber qué datos filtrar más adelante.



La forma más sencilla de limpiar las filas que almacenan datos irrelevantes será ordenar los datos por la distancia que hemos calculado y eliminar las filas que contengan duplicados del identificador. De esta forma nos quedaremos únicamente con una entrada por avistamiento que incluirá la información meteorológica de la estación más cercana.



Por ultimo volveremos a eliminar las columnas auxiliares que hemos creado.

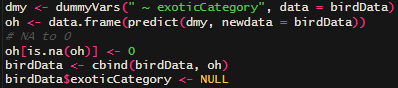


En este punto ya tenemos nuestro dataset unificado y podemos prepararlo para su posterior análisis y uso en sistemas predictivos de inteligencia artificial.

Dividimos la fecha en tres columnas numéricas.



Como la columna exoticCategory es un dato categórico con tres posibles opciones lo transformaremos en tres columnas diferentes con datos binarios.



Como algunas columnas contienen valores nulos tendremos que reemplazarlos, en el caso de howMany los rellenaremos con 1s.



Para la columna velmedia utilizaremos la mediana



Y para prec utilizaremos 0s tanto para los valores nulos como para los marcados como Ip (esto lo hacemos siguiendo las explicaciones de la API de AEMET), además transformaremos toda la columna en datos numéricos.



Por ultimo hemos visto que algunas localizaciones incluían caracteres invisibles que rompían el formato del archivo csv, los reemplazaremos por espacios simples antes de guardar el archivo.

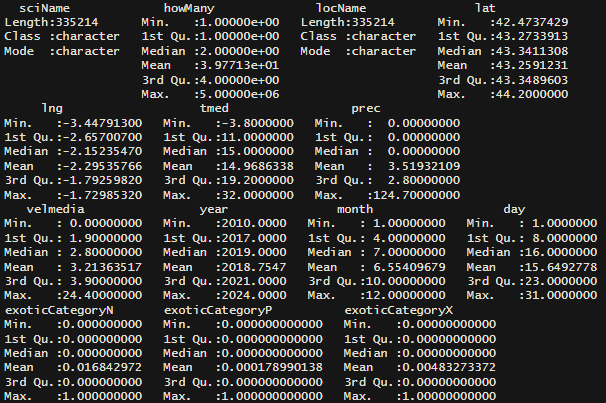


Análisis de datos

En este paso utilizaremos el archivo tratado para buscar información relevante, primero cargaremos el archivo y mostraremos un resumen de sus datos.

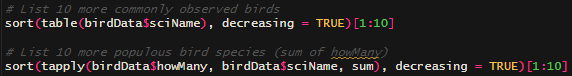


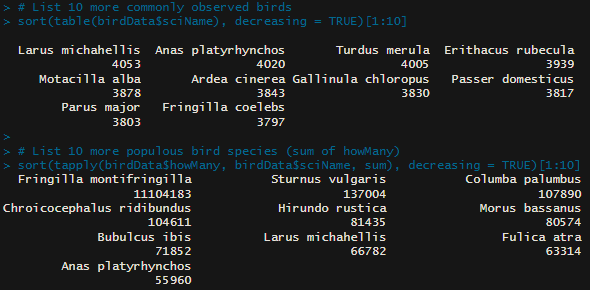
Esto nos dará un primer vistazo con el que podremos ver la cantidad de datos de los que disponemos y su distribución, incluyendo máximos, mínimos, medias, medianas y cuartiles.



Por ejemplo, podremos ver como las columnas howMany y prec tienen una gran diferencia entre sus medias, medianas y máximos, más adelante podremos ver como esto se debe a que, aunque la mayoría de avistamientos tienen datos muy bajos de ambos, hay unos pocos con datos muy desproporcionados que generan la distancia entre media y mediana.

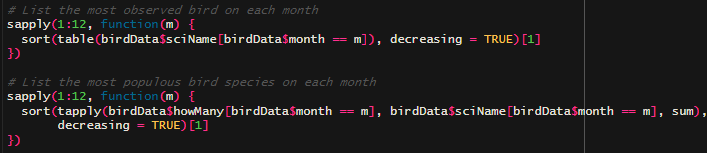
Tras esto hemos listado las 10 especies que más veces se han avistado y las 10 con mayor población.

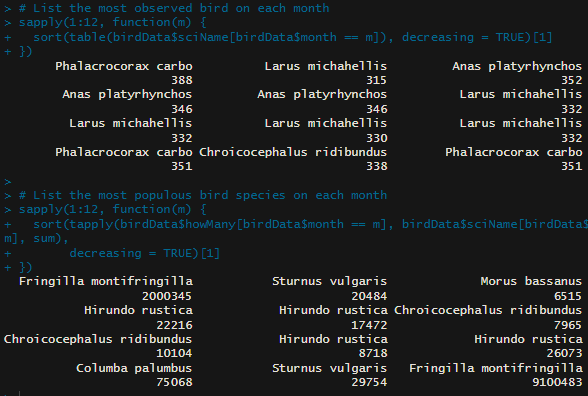




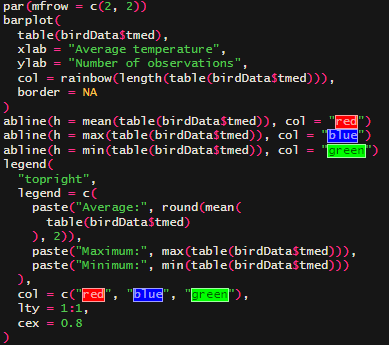
En estos resultados podemos ver como únicamente la especie con más avistamientos aparece en la segunda lista, esto puede deberse a que la mayoría de avistamientos pertenecen a aficionados que únicamente reportan las aves menos comunes mientras unos pocos pertenecen a grupos que realizan recuentos de población y llegan a reportar millones en una sola entrada.

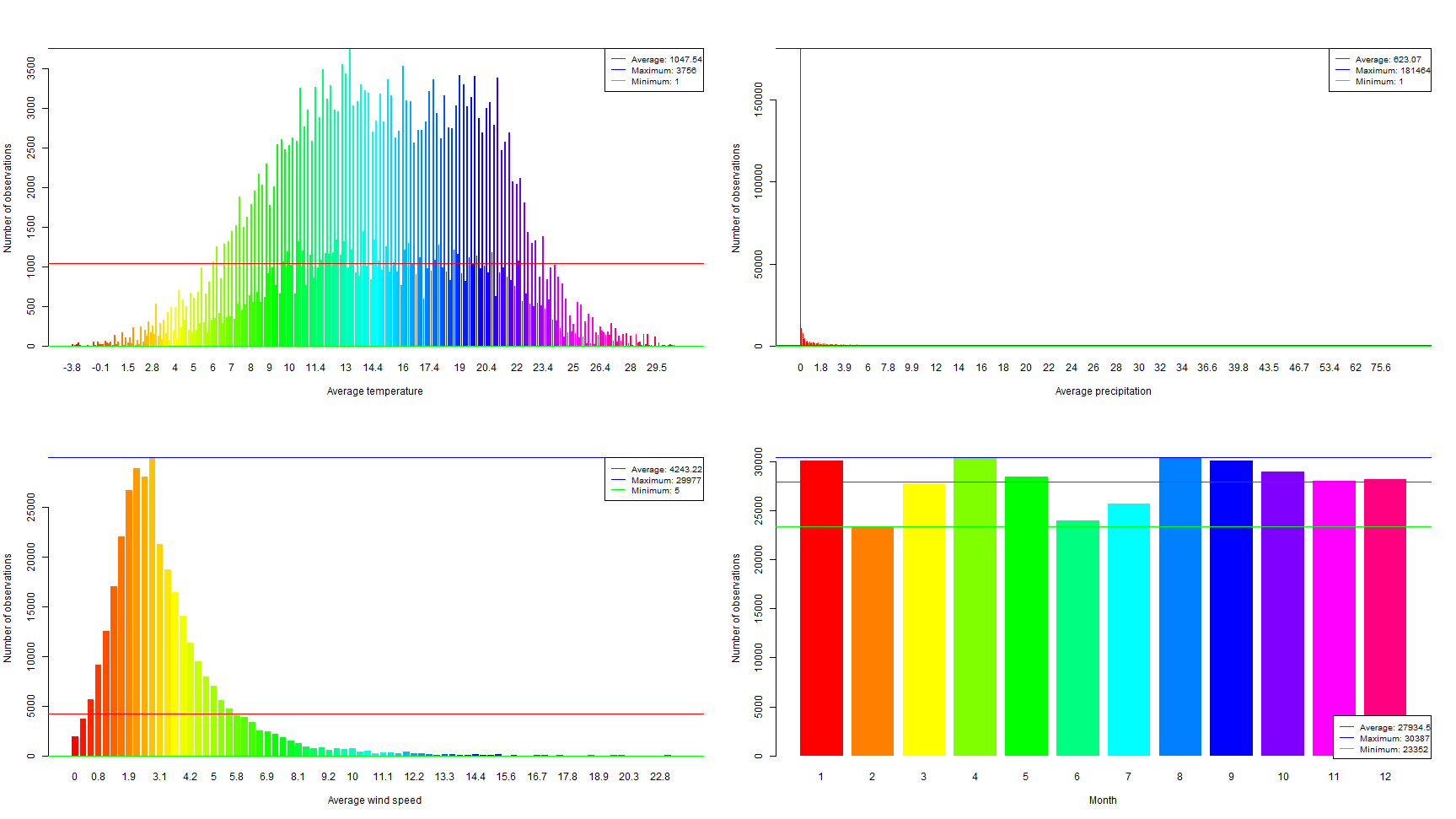
También hemos realizado el listado de la especie más vista y con mayor población de cada mes para comprobar si las migraciones afectan en gran medida a los datos.





Para buscar relaciones entre los avistamientos y el clima se han realizado tres graficas de barras en las que se comparan el número de avistamientos con la temperatura, las precipitaciones y la velocidad del viento respectivamente, finalmente se ha añadido otra grafica que muestra los avistamientos mensuales. El código corresponde al primero.



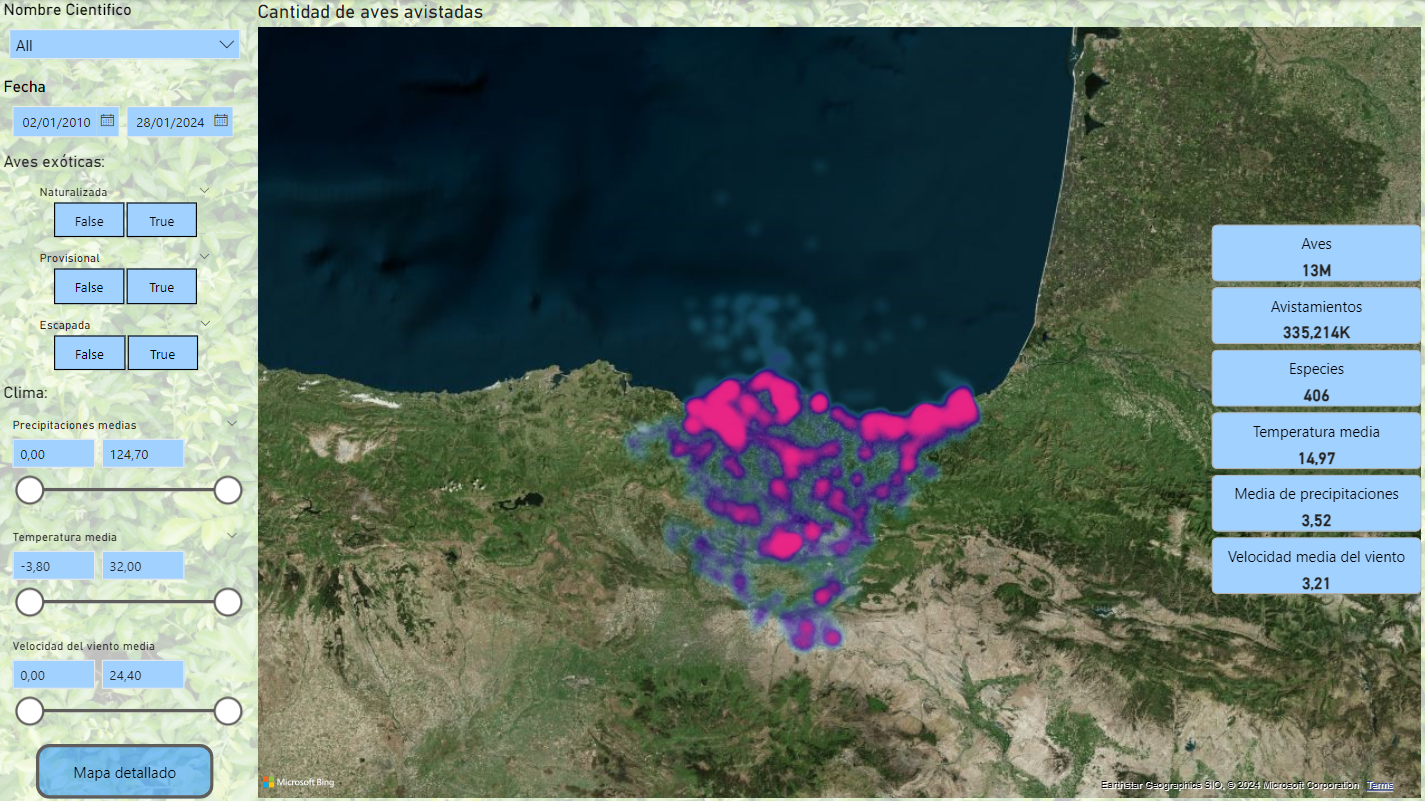


En la primera grafica podemos ver una distribución normal de los avistamientos con respecto a la temperatura media del día.

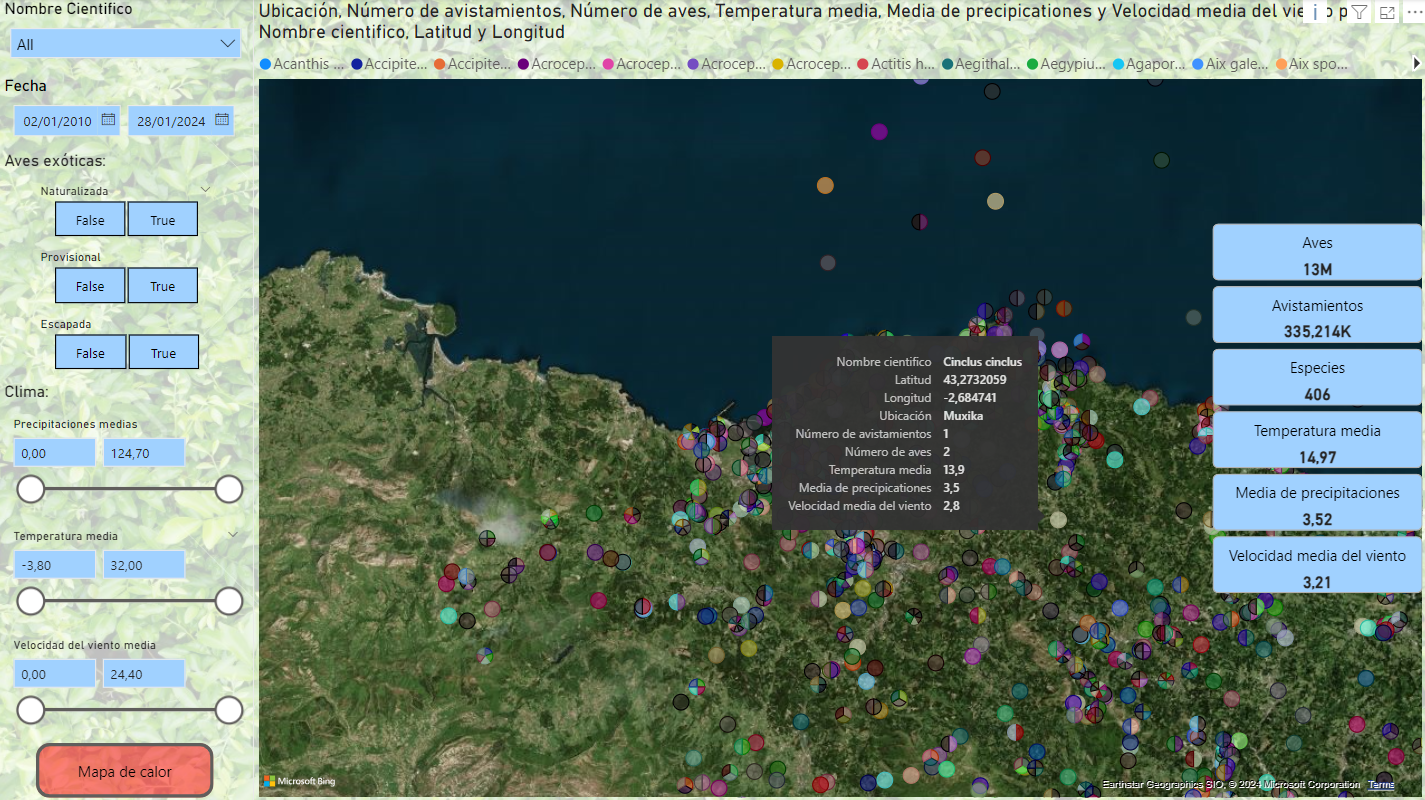
La segunda muestra como apenas se realizan avistamientos con precipitaciones ya que más de la mitad de ellos están en 0

Sobre la velocidad del viento vemos una distribución escorada a la izquierda que nos indica como la mayoría de avistamientos se realizan en días con poco viento.

Para terminar, se ha realizado una presentación con PowerBI en la que se puede visualizar toda la información de forma visual en un mapa interactivo.



Como se ve en la imagen disponemos de filtros para cada parámetro del dataset a excepción de la cantidad de aves que será la utilizada para decidir el color del mapa de calor.



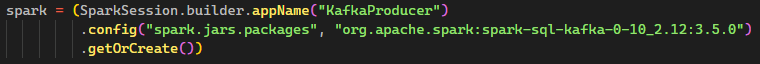
También tenemos un mapa detallado en el que podremos ver los datos específicos de cada punto.

Almacenamiento en Hive

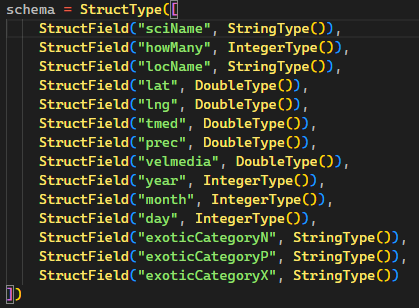
Por ultimo almacenaremos los datos en una base de datos Hive.

Para esto comenzaremos creando un productor de Kafka que enviara los datos de nuestro equipo local a un servidor remoto.

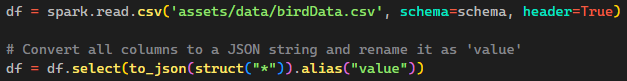
Utilizará una sesión de Spark configurada con las librerías necesarias de Kafka.



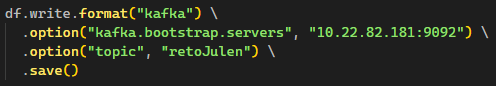
También necesitaremos definir la estructura de nuestros datos de la siguiente manera.



Entonces podremos leer el archivo y transformarlo a un dataframe con una colección de json almacenados en la columna value, también podríamos asignarles identificadores con una columna key.

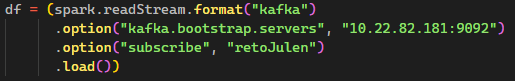


Por ultimo realizaremos el stream contra el servidor.

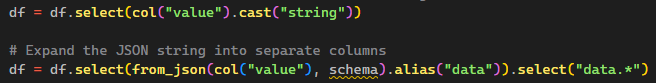


Después haremos un consumidor que ejecutaremos desde nuestra máquina de Hadoop para recibir la información.

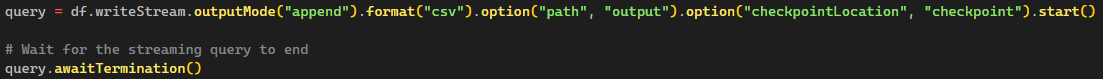
Para ello utilizaremos una sesión de Spark y estructura de datos idénticas a las del productor y leeremos el streaming con la siguiente función.



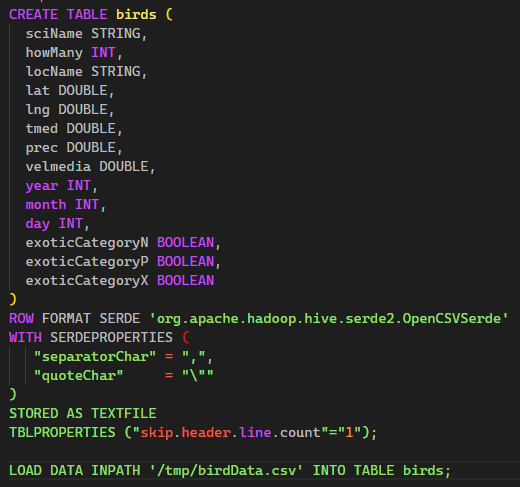
Ahora necesitamos convertir los atributos de los json en columnas.



Por último crearemos el archivo.



Ahora necesitaremos subir esta información al sistema HDFS para finalmente cargar el archivo en la base de datos Hive mediante el siguiente script.



En la configuración de la propia tabla elegimos el formato del csv, separado por comas, con los textos entrecomillados y con las cabeceras en la primera línea.

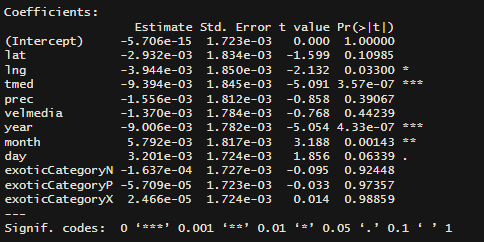
**Modelos predictivos**

Preparación de los datos

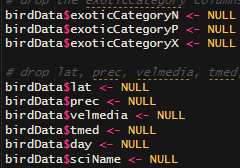
El primer paso para poder desarrollar cualquier modelo predictivo será preparar los datos para su ingesta.

Parte de esta preparación se ha realizado durante la fase del tratamiento de datos: eliminación de valores nulos, transformación en datos en numéricos y one hot encoding.

Tras esto es importante seleccionar las columnas más significativas para evitar que los modelos cometan errores al tratar de utilizar datos irrelevantes.



Una vez localizadas eliminaremos las sobrantes.



También eliminaremos los datos fuera de lo normal.



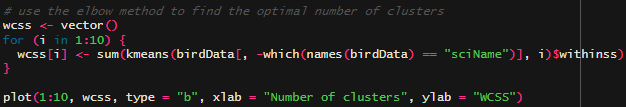
El último paso de preparación de datos será escalar los datos de entrada para que todas ellas puedan ser comparables y sea el modelo el que asigne el peso de las relaciones.

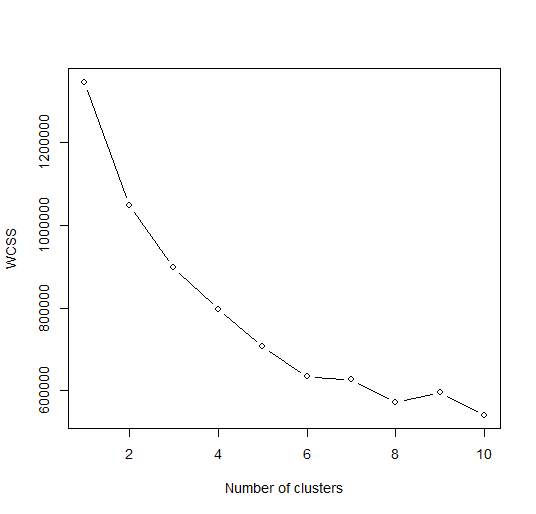


Clustering

El clustering nos ayudará a agrupar los datos sin necesidad de tener un conocimiento previo en la materia. En nuestro caso nos servirá para tratar de agrupar las especies en familias con características similares y así poder clasificarlas.

Para buscar el numero óptimo de clusters utilizaremos el método del codo, en el que utilizaremos la distorsión calculada con diferentes cantidades de clusters.

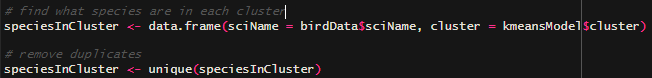




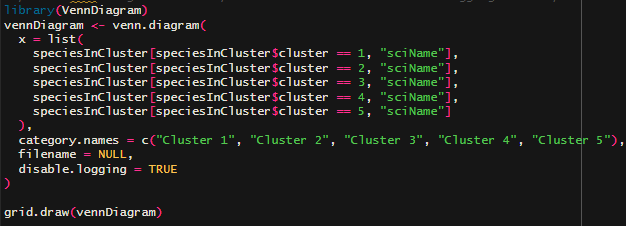
En nuestro caso el resultado óptimo han sido 6 clusters.

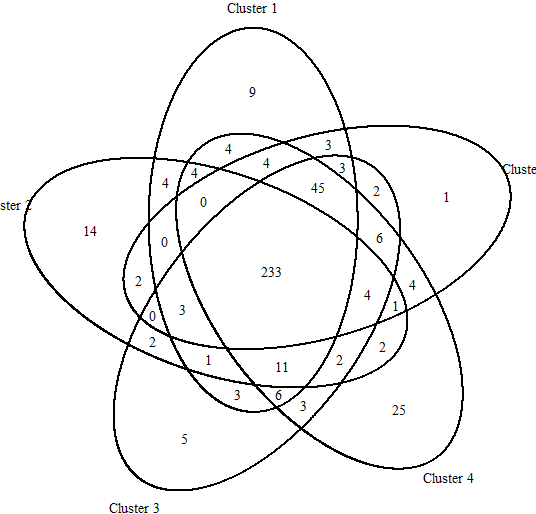
Para comprobar si las especies están directamente relacionadas con estos clusters se han realizado diagramas de venn en los que se puede comprobar de forma sencilla cuantas especies diferentes hay en cada cluster.

Primero calcularemos las especies que hay en cada cluster.



Y con estos datos realizaremos el diagrama.



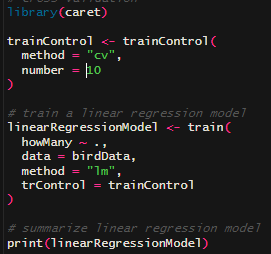


La conclusión de este diagrama es que no podemos clasificar las especies por la relación entre sus avistamientos y el clima.

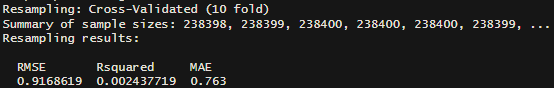
Regresión lineal

Abandonada la opción de clasificarlas especies trataremos de realizar una regresión lineal con la que predecir la cantidad de aves que serán avistadas en un día y lugar concretos.

Para poder comprobar la calidad de nuestro modelo utilizaremos cross validation, separando nuestros datos en 10 conjuntos y entrenando el modelo con diferentes combinaciones de datos de entrenamiento y pruebas.



Este ha sido el resultado de nuestro entrenamiento.



De nuevo a pesar de haber limpiado los datos y haber utilizado únicamente los parámetros más significativos el resultado ha sido pobre.

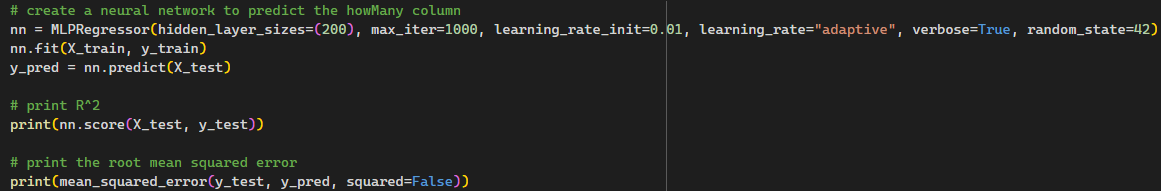
El RMSE y el MAE podemos considerarlos altos porque nuestros datos únicamente varían entre 1 y 5, de forma que un error de casi 1 representa una gran diferencia.

R^2 es bajo cuando este parámetro nos muestra de forma porcentual como nuestro modelo es capaz de definir los datos, siendo 1 el valor al que debería acercarse.

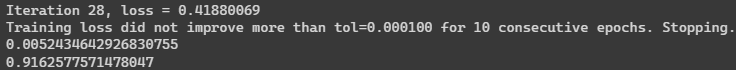
Red neuronal

El siguiente modelo a entrenar será una red neuronal con la que intentaremos mejorar los resultados de la regresión lineal.

En este caso, por agilidad únicamente se ha utilizado un grupo de entrenamiento y otro de prueba.



Estos han sido los resultados

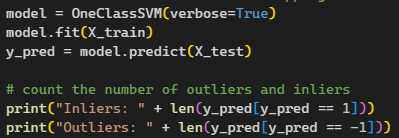


Podemos ver que R^2 ha mejorado respecto al obtenido en la regresión lineal, aunque sigue siendo muy bajo y el RMSE es prácticamente idéntico.

Tras haber obtenido resultados similares entrenando con diferentes parámetros la conclusión a la que podemos llegar es que los datos no tienen una relación suficientemente fuerte para tratar de realizar este tipo de predicciones.

One class SVM

Como última prueba intentaremos entrenar un modelo que prediga únicamente si un avistamiento es posible en unas condiciones concretas. Para esto utilizaremos one class SVM, un sistema utilizado para detectar outliers en conjuntos de datos.



Este es el resultado no habiendo introducido ningún outlier en los datos de prueba:



Claramente el modelo no ha sido capaz de generalizar los datos, reforzando de nuevo la conclusión de que los datos no tienen una relación significante entre sí.