Johan Sebastian Castillo Mazo

Documento técnico sobre las acciones tomadas para el análisis
Ciencia de datos con peludos

1. Manejo de datos

Para el manejo de los datos decidí utilizar las librerías por excelencia pandas y numpy en cuanto al almacenamiento decidí utilizar google drive junto con su API de integración de Google Colab para poder montar mi drive como un disco al cual podía acceder desde el notebook fácilmente.

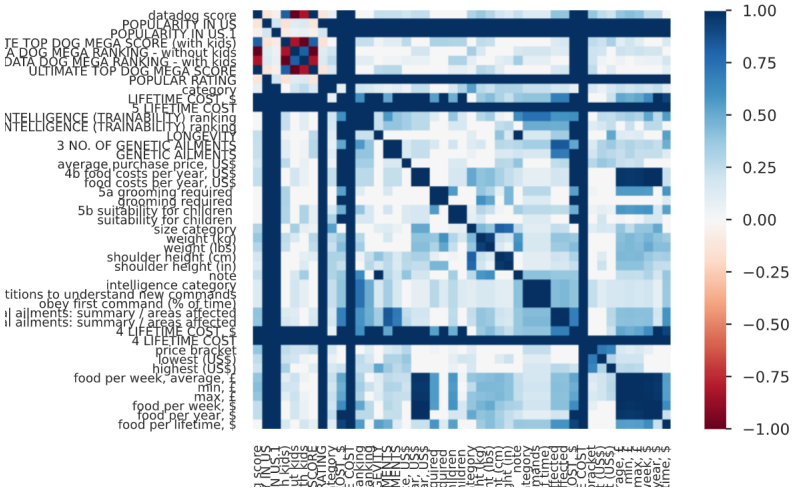
2.Limpieza de datos

Como relata en el pdf de presentación el dataset aun tenia algunos problemas especialmente varias columnas con el nombre ‘’Unnamed’’ por lo que utilice el atributo loc del Data Frame para acceder a las filas, en su primer parámetro las seleccione todas con ‘’:’’ y luego en el segundo parámetro seleccione todas las columnas las cuales tuvieran por nombre ‘’Unnamed’’ y negué esa condición para seleccionar todas las columnas que precisamente no se llamaran así, con ello conseguí retirar todas las columnas no deseadas del dataset.

Luego de esto aun el dataset contaba con una fila extra que contenía información adicional sobre cada variable por lo que esto podría afectar mis análisis futuro entonces la retire seleccionando todas las filas saltandome la 0 y esta la guarde en una variable de información adicional para consultarla cuando quisiera.

3.Analisis superficial

Para el análisis superficial decidí utilizar la librería de Pandas Profiling y aunque me ayudo a ver las variables del dataset mas a detalle también muchas de estas las identificó como categóricas cuando a mi parecer están mejor en la categoría de numéricas, además de ello al ser tantas variables y algunas con nombres bastante largos ver las correlaciones o los valores nulos era un desastre.



4.Analisis Extenso

Para el análisis principal decidí dividirlo en varias preguntas que me llegaron a la mente después de explorar el dataset y de varias que surgieron mientras investigaba estas primeras.

1. ¿Cuál es el top 5 de perros más populares en USA?

Para esto utilice la variable POPULARITY IN US.1 y filtre el dataset en sólo aquellos registros que tuvieran datos en esta variable, dejando los NaN por fuera.
Después de esto convertí la columna a tipo int64 para poder ordenar los valores según su lógica numérica con el método astype de pandas.
Terminando ordene los registros del dataset según esta variable y escogí los primeros 5 registros.

2.Cuál es el promedio de vida de los perros entre todas sus razas?

En esta también filtra los datos a partir de los valores de la variable 2 LONGEVITY para eliminar los datos NaN.

Después de filtrar los datos me di cuenta que aún había un dato anómalo una combinación entre letras y números para la raza de perro Black Russian Terrier la cual decía ‘’1.83 - really?’’, seguramente el creador del dataset estaba igual de confundido que yo con ese valor.

Así que estuve buscando en otras fuentes y vi que el promedio de vida de esta raza estaba entre los 10 y 11 años, así que decidí reemplazarlo con el método replace.

Luego convertí la columna a int64 para poder graficarla correctamente con altair y también para poder sacar el promedio entre las razas.

Al graficarla con un gráfico de barras de altair pude ver mejor la distribución de longevidad entre las razas, y al final con el método mean sobre la columna pude obtener el promedio de edad.

Con esta pregunta se me ocurrió la siguiente.

3.Hay una correlación entre el tamaño y longevidad de los perros?

Para esta utilize la misma columna de longevidad (2 LONGEVITY) y la contraste con la variable (weight (kg)) que define el peso promedio de cada raza, y dividiendo por colores por la variable (size category) la cual categoriza el tamaño del perro entre grande, pequeño y mediano todo esto en un diagrama de dispersión.

Al final logré ver una leve correlación negativa en cuanto al tamaño/peso de un perro con su longevidad, a más peso menos años de vida.

4.Cuáles son los perros más baratos de mantener?

Aqui volvi a limpiar los datos de valores NaN o valores extraños en las variables que me interesaban las cuales eran (food per lifetime, $) gasto promedio total en comida a través de toda la vida del perro, (Other regular costs, total per lifetime, $) gasto promedio total en varias categorías como veterinaria, peluqueria juguetes y demás a través de toda la vida del perro y (Dog breed) raza de perro.

Con la pregunta en mi mente decidí hacer 2 data frames para dividir entre tipo gastos comida u otros y por cada raza de perro. La variable de cantidad de gastos en dólares contenía comas ‘’,’’ lo que hacía difícil su manipulación así que las elimine combinando el método apply con el método replace.

Luego de eso los uni y los grafique con altair añadiendo el parámetro sort al eje x y especificando que el ordenamiento dependiera del eje y.

Con esto obtuve una gráfica ordenada de menor a mayor según el gasto total promedio (dividido en comida y otros gastos) a través de toda la vida de una raza de perro.

Gracias a este gráfico también puede resolver la siguiente pregunta.

5. ¿Cuáles son los perros más caros de mantener?

Que al final es la misma gráfica pero invertida.

Con estas 2 preguntas pude observar algo extraño es común que el gasto en comida de un perro sea un cuarto del total de gastos pero en algunos perros este gasto sobrepasa ese cuarto y se aproxima a la mitad total de los gastos, estos son el irish wolfhound, saint bernard y giant schnauzer.
Este raro comportamiento me hizo plantearme 2 nuevas preguntas.
Acaso se debía al gran tamaño de estos y por eso comen más lo que conlleva a un aumento en el gasto de la comida? o acaso tienen mejor salud y se enferman menos? lo que bajaría el porcentaje de gasto en salud.

No tenía datos para responder la primera, pero intente responder la segunda, el dataset contiene una variable la cual recolectaba las posibles enfermedades genéticas que podría desarrollar una raza en especifico (3 NO. OF GENETIC AILMENTS). Con esto en mente me formule las siguientes 2 preguntas.

6.Cuáles razas de perro pueden desarrollar más o menos enfermedades genéticas?

Para esto limpie los datos de valores NaN o de perros con posibles enfermedades genéticas de 0.

Luego grafique con altair un diagrama de barras con el eje x siendo la raza, el eje y siendo las posibles enfermedades, el color siendo la categoría de tamaño del perro, con el eje x siendo ordenado por el valor del eje y en orden descendiente.

Con esto pude ver la distribución de posibles enfermedades entre los diferentes tamaños de perros, pero no me quedó realmente muy claro y en los primeros puestos con más enfermedades había tanto perros grandes como pequeños. Por lo que se me ocurrió otra pregunta.

7.Cómo se divide la cantidad de enfermedades entre los diferentes tipos de tamaño?

Para esto agrupe los datos según el tamaño del perro, luego de esto sume las posibles enfermedades de cada uno de los grupos, después de esto sume todas las enfermedades para obtener el total.
Divide el total de cada uno de los grupos por el total de enfermedades para obtener el porcentaje total de ese grupo.
Luego grafique un pie chart con altair y trate por todos los medios de colocarle labels personalizadas con los porcentajes a cada tajada del pie pero no lo logre.

Con todo esto pude darme una idea mejor de la distribución de las posibles enfermedades genéticas entre los diferentes tamaños de perro, al final los perros medianos y pequeños quedaron con un porcentaje de 30% cada uno y los perros grandes con el restante 40% al final esto contradice mi pensamiento de que los perros grandes se enfermaban menos y por lo tanto su gasto en salud de reducia, pero también este análisis tiene un problema y es que al agrupar por tamaños puede que haya perros que compartan enfermedades lo que conlleva a enfermedades repetidas en el conteo y porcentaje de cada grupo, me hubiera gustado filtrar las enfermedades repetidas y seguir indagando pero no tuve más tiempo para desarrollar esta parte, por lo tanto creo que no saco nada realmente concluyente de esta última pregunta.

Pero como no quería irme con mal sabor de boca se me vino a la mente una última pregunta también relacionada con el tamaño.

8. El tamaño de un perro influye en su inteligencia/entrenabilidad?

Aquí de nuevo limpie los datos NaN y aquí me di cuenta que tal vez debí haber limpiado el dataset desde el principio, pero ya que el dataset tenía datos faltantes en varias columnas diferentes no quería perder toda una fila donde faltaba algo con lo cual no estaba trabajando.

Luego elimine el símbolo ‘’%’’ de la variable INTELLIGENCE (TRAINABILITY) ranking para que esta fuera más fácil de graficar.

Al final hice un diagrama de dispersión en el cual el eje x era el peso, el eje y la entrenabilidad entre más alta mejor, y el color dividido en el tamaño, con esto no vi ninguna correlación evidente entre el tamaño y la entrenabilidad, pero si pude observar que los perros extremadamente grandes suelen ser más difíciles de entrenar o suelen ser talvez mas tontos? ¿Tal vez el creador de Marmaduke se inspiró en esto?

Con esas últimas conclusiones terminé mi análisis y empecé con la parte de Machine Learning.

Machine Learning

Estuve pensando en maneras de implementar un modelo de deep learning con los datos que había estado trabajando pero en realidad no se me ocurrió nada. Por lo tanto decidí buscar uno nuevo y me encontré con un dataset el cual tenía 15000 fotos de perros categorizados entre feliz, triste, relajado y enojado con esto empecé a crear mi modelo.

1.Tecnologias

Me decidí por utilizar Tensor Flow/Keras porque he de admitirlo aunque me encanto el mundo del machine learning en este momento estoy literal así:

Con esto en mente supe que la abstracción de Keras me ayudaría mucho.

2.Almacenamiento

Para el almacenamiento de este nuevo dataset seguí utilizando drive con su API de google colab.

3.Carga de imágenes

Para la carga de imágenes decidí usar la librería path lib que hace que trabajar con directorios en python sea bastante fácil, estaba pensando en una manera óptima de cargar las imagenes cuando entre en la documentación de Tensor Flow y me di cuenta que ya tenían un método para la carga de estas el (tf.keras.utils.image_dataset_from_directory), par utilizarlo tenía que definir algunos parámetros como la carpeta donde estan las imagenes, en cuanto voy a dividir el dataset de validación y cuanto le voy a dejar al de entrenamiento, la semilla para controlar la aleatoriedad con la cual escoge esto, el tamaño con al cual va a redimensionar las imágenes al cual le coloque 256 * 256 y luego me arrepentiría de ello, y el batch size que por lo que entiendo es el número de muestras que van ha entrar simultáneamente al modelo durante el entrenamiento.

Con los 2 datasets el de entrenamiento y el de validación listos, saque el nombre de las clases con el atributo class names de los datasets resultantes.

Luego utilice unas funciones de optimizado de memoria en los datasets, que no me quedó muy claro qué es exactamente lo que hacen pero Tensor Flow lo recomendaba.

4.Arquitectura del modelo

El modelo es uno secuencial compuesto por 10 capas la primera de redimensión, el siguiente grupo de 6 que está compuesto por una capa Con v2D que por lo que lei es una capa convolucional bastante buena imagenes luego una Max Pooling 2D la cual parece que cambia los valores de salinidad a ser un tipo diferente como calculando los valores máximos de las diferentes regiones de la imagen? La verdad esta aun no la comprendo muy bien, pero estas 2 se intercalan 3 veces mientras las neuronas de la siguiente capa se duplican. Luego de este proceso la data pasa por una capa flatten la cual aplana los vectores para convertirlos en un solo vector de una dimensión, por último pasa por una capa densa de 128 neuronas y una última la capa de salida la cual es del tamaño de las clases que queremos clasificar. Todas estas capas utilizan una función de activación relu.

Para optimizar escogí el algoritmo ‘’adam’’ por recomendación de tensor flow, para el error decidir también utilizar por recomendación de tensor flow el SparseCategoricalCrossentropy que por lo que lei es preferible cuando se está trabajando clasificación, y para la métrica decidir utilizar la precisión.

5.Entrenamiento del modelo

Finalmente empecé a entrenar el modelo durante 100 épocas pero para mi sorpresa la primera época decía que iba a tardar 15 minutos! Y además de eso después de 30 minutos recibí el mensaje en google colab te has quedado sin memoria ram!, ahí fue cuando me puse a revisar el modelo en más detalle y utilice el método summary para verlo mejor, y oh por dios cuando vi mi modelo tenía la cantidad aproximada de 16 millones de parametros, ahí dije algo va mal asi que decidi probar con menores tamaños de imagen hasta que las cosas no fueran tan monumentales. Al final me quede con imágenes de 32 * 32 las cuales no eran muy pequeñas ni tampoco muy grandes.

Después de entrenar al modelo por 100 épocas con estas nuevas dimensiones, logre un accuracy del 98%

6.Prueba del modelo

Testee el modelo con diferentes imágenes que nunca había visto y me dio muy buenos resultados!, me gustaria tener mas tiempo para probarlo mas o ver si hay algún tipo de overfitting pero ya no tengo tiempo.

7.Uso del modelo

Para terminar hay un notebook llamado usa_el_modelo en el cual están todos los pasos para descargar el modelo, cargarlo en memoria, cargar la foto y ejecutar la predicción.

Gracias al equipo de codigofacilito por todo su trabajo en este bootcamp.

Johan Sebastian Castillo Mazo