CLASIFICACIÓN ESTELAR

Proyecto Final - Data Science

Alumno: Lucas Gauto

2 de enero de 2022

TABLA DE CONTENIDOS

DESCRIPCIÓN DEL CASO

Historia de los Datos

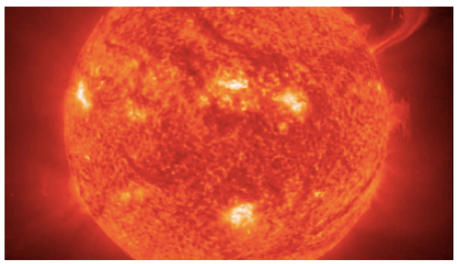
Por mi trasfondo en Ciencias Físicas, decidí trabajar con un proyecto que incorpore algún aspecto de este ámbito científico, y encontré un dataset del **Sloan Digital Sky Survey** (**SDSS**), que es un proyecto de investigación del espacio mediante fotometría y con datos de corrimiento al rojo, realizado en Nuevo México a partir del año 2000.

Sobre Los Datos

La fotometría es, de forma sencilla, el estudio de la luz, y es algo esencial en astronomía. El análisis de las diferentes frecuencias de luz captadas por el telescopio del **SDSS** sirve para determinar diferentes características de los objetos estudiados, como su tamaño, temperatura, luminosidad, masa, composición, velocidad, distancia, etc.

El telescopio ha captado en sus observaciones tres tipos de objetos que se nombrarán a los largo de este documento, estos son:

- **Estrellas:** Esferas gigantes de plasma, como nuestro Sol.
- **Galaxias:** Conjuntos de estrellas, nubes de gas, planetas, polvo cósmico, materia oscura y energía, que se mantienen unidas por acción de la Gravedad. Según su forma pueden ser *espiraladas, elípticas,* o *irregulares*. Por ejemplo, nosotros nos encontramos en la galaxia Vía Láctea. Se espera que en el centro de todas las galaxias haya agujeros negros supermasivos.
- **Cuásares:** Cuando grandes cantidades de materia caen en estos mencionados agujeros negros, por efecto de su campo electromagnético, y su rápida velocidad de rotación, se genera un disco de acreción y un jet de energía, liberada en forma de ondas electromagnéticas de todas las frecuencias. Esto los convierte en los objetos más brillantes del Universo conocido, brillando aún más que galaxias enteras.



Estrella

Galaxia

Cuásar

Variables

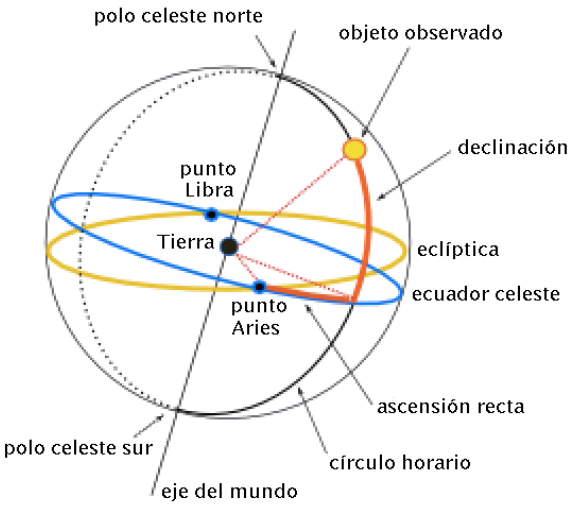
En el dataset seleccionado se incluye información distribuida en varias columnas. Esta es una breve descripción de estas variables trabajadas:

Variable	Descripción
objid	*Object ID* - código de identificación único de cada objeto
ra	Ascensión recta (*right ascension*) - una de las coordenadas ecuatoriales.
dec	Declinación (*declination*) - una de las coordenadas ecuatoriales.
u, g, r, i, z	Filtros
run, rerun, camcold, field	Descriptores de campos dentro de la imagen
class	Indica si el objeto observado es una estrella (STAR), una galaxia (GALAXY), o un cuásar (QSO).
redshift	Corrimiento al rojo
plate	Número de placa
mjd	*Fecha Juliana* o *Modified Julian date of observation* - número de días y fracción transcurridos desde el mediodía del 1 de enero del año 4713 a.C. Es decir, es una forma de expresar fecha y horario de la observación en un sólo número.
fiberid	*Optic fiber ID*

Los conceptos subrayados son explicados a continuación.

Coordenadas Ecuatoriales

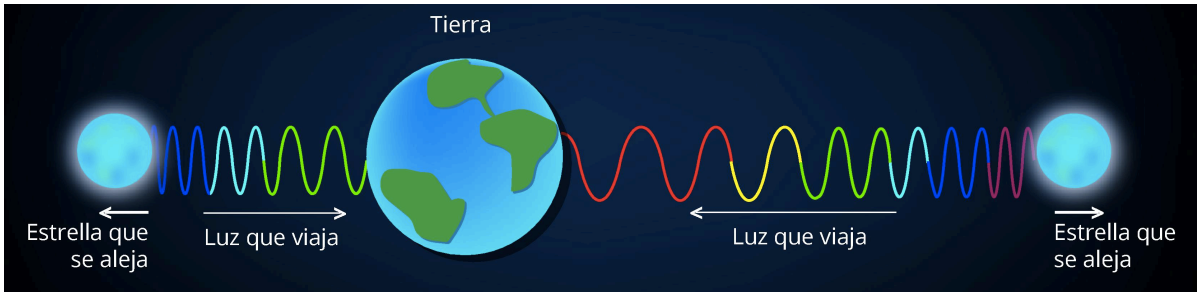
Se trata de un sistema que permite ubicar objetos en la esfera celeste respecto al ecuador y el equinoccio vernal. Las coordenadas ascensión recta y declinación son equivalentes a la latitud y longitud geográficas:



Redshift o Corrimiento al Rojo

El corrimiento al rojo es uno de los temas centrales de este dataset, y de los conceptos astronómicos más importantes. Es un resultado del Efecto Doppler aplicado a ondas de luz, y se conoce como la discrepancia entre la frecuencia electromagnética (luz) emitida por el objeto y la frecuencia electromagnética capturada por los sensores ubicados en la Tierra, observando un desplazamiento al rojo al final del espectro electromagnético - de ahí el nombre.

Es decir, la luz proveniente de un objeto que se aleja respecto a la Tierra es captada como “más roja” que lo que realmente es. Y mientras más lejos esté el objeto, mayor será esta discrepancia. Para fines físicos, el redshift es una de las mayores evidencias en cuanto refiere a la expansión universal.



En la imagen se ilustra el corrimiento al rojo: una estrella que se aleja de nosotros emite luz azul que captamos como luz verde (a la izquierda), y otra estrella que también se aleja pero esta vez más lejana emite luz violeta que captamos como roja (a la derecha). En ambos casos, la luz emitida no es la misma que la percibida, se ha “corrido” en el espectro, hacia el lado del rojo.

Es decir, si se tiene la distancia a la que se encuentra un objeto, se puede calcular cuál sería su redshift asociado; y lo mismo puede hacerse de manera inversa (resultando más provechoso): si se tiene su redshift, se puede calcular su distancia.

Para este trabajo en particular, el redshift será una de las variables más importantes a analizar, puesto que no se tiene como dato la distancia de los objetos estudiados, pero sí se puede lograr inferir en base a esta información.

OBJETIVOS DEL PROYECTO

Objetivos Generales

Al tratarse de un proyecto final referente al curso de Data Science, se tienen como objetivos generales el uso de todas las herramientas adquiridas durante el mismo, destacando: la exploración de grandes volúmenes de data, la limpieza de datos, el análisis variado (ya sea univariado, bivariado, o multivariado) de los mismos, el armado de gráficos relevantes, el uso de modelos de Machine Learning, el análisis de los modelos elegidos, la optimización de los mismos, y la elección justificada del mejor modelo para los datos.

Objetivos Específicos

Para este dataset y proyecto en particular, se tuvieron en cuenta los siguientes objetivos:

- Analizar y limpiar el archivo de datos original para obtener un dataset prolijo que me dé la información necesaria.
- Encontrar una forma de determinar la distancia de cada objeto en base a su Redshift.
- En base a lo anterior, y utilizando también los datos de coordenadas ecuatoriales, armar un gráfico que me muestre la distribución de los objetos según su distancia la Tierra (en coordenadas polares).
- Encontrar relaciones entre las columnas del dataset.
- Crear un modelo de Machine Learning que me permita clasificar un objeto en base a las demás variables. Es decir, que a priori pueda identificar si lo observado es una estrella, una galaxia, o un cuásar. Para esto adelanto que tendré como objetivo la columna "**class**" del dataset.

DATA WRANGLING

Base de Datos

Comenzamos a trabajar el dataset.
Para tener una idea de lo que hablamos, el dataset original tenía este aspecto:

	objid	ra	dec	u	g	r	i	z	run	rerun	camcol	field	specobjid	class	redshift	plate	
0	1.237650e+18	183.531326	0.089693	19.47406	17.04240	15.94699	15.50342	15.22531	752	301	4	267	3.722360e+18	STAR	-0.000009	3306	5
1	1.237650e+18	183.598371	0.135285	18.66280	17.21449	16.67637	16.48922	16.39150	752	301	4	267	3.638140e+17	STAR	-0.000055	323	5
2	1.237650e+18	183.680207	0.126185	19.38298	18.19169	17.47428	17.08732	16.80125	752	301	4	268	3.232740e+17	GALAXY	0.123111	287	5
3	1.237650e+18	183.870529	0.049911	17.76536	16.60272	16.16116	15.98233	15.90438	752	301	4	269	3.722370e+18	STAR	-0.000111	3306	5
4	1.237650e+18	183.883288	0.102557	17.55025	16.26342	16.43869	16.55492	16.61326	752	301	4	269	3.722370e+18	STAR	0.000590	3306	5
5	1.237650e+18	183.847174	0.173694	19.43133	18.46779	18.16451	18.01475	18.04155	752	301	4	269	3.649550e+17	STAR	0.000315	324	5
6	1.237650e+18	183.864379	0.019201	19.38322	17.88995	17.10537	16.66393	16.36955	752	301	4	269	3.232870e+17	GALAXY	0.100242	287	5
7	1.237650e+18	183.900081	0.187473	18.97993	17.84496	17.38022	17.20673	17.07071	752	301	4	269	3.722370e+18	STAR	0.000315	3306	5
8	1.237650e+18	183.924588	0.097246	17.90616	16.97172	16.67541	16.53776	16.47596	752	301	4	270	3.638290e+17	STAR	0.000089	323	5
9	1.237650e+18	183.973498	0.081626	18.67249	17.71375	17.49362	17.28284	17.22644	752	301	4	270	3.243690e+17	GALAXY	0.040508	288	5
10	1.237650e+18	183.979195	0.135998	19.29772	17.80227	17.18266	16.92335	16.79928	752	301	4	270	3.722360e+18	STAR	-0.000035	3306	5
11	1.237650e+18	184.085331	0.112110	18.83307	17.51785	16.94273	16.71418	16.60521	752	301	4	271	3.722380e+18	STAR	0.000623	3306	5
12	1.237650e+18	184.102098	0.191511	19.56250	18.19113	17.65759	17.47573	17.39203	752	301	4	271	3.722370e+18	STAR	0.000055	3306	5
13	1.237650e+18	184.160510	0.075645	19.57990	17.72815	16.98740	16.68076	16.50426	752	301	4	271	3.722380e+18	STAR	0.000008	3306	5
14	1.237650e+18	184.189574	0.099482	19.25667	17.54869	16.63578	16.14922	15.76639	752	301	4	271	3.243660e+17	GALAXY	0.072087	288	5

Dataset original (hay columnas ocultas)

Las columnas fueron explicadas en una sección anterior, y se cuenta con 10000 (diez mil) datos con valores en cada columna, por lo que es necesario un trabajo de los datos en general; un análisis individual sería imposible.

Limpieza del Dataset

Como primer medida, se buscó la cantidad de datos nulos del dataset para su posterior limpieza y desarrollo. Como se ve en la imagen, no había ningún dato nulo, por lo que este proceso no fue necesario y ahorró tiempo para el posterior análisis.

```
Data columns (total 18 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   objid      10000 non-null  float64
 1   ra         10000 non-null  float64
 2   dec        10000 non-null  float64
 3   u          10000 non-null  float64
 4   g          10000 non-null  float64
 5   r          10000 non-null  float64
 6   i          10000 non-null  float64
 7   z          10000 non-null  float64
 8   run        10000 non-null  int64
 9   rerun      10000 non-null  int64
 10  camcol     10000 non-null  int64
 11  field      10000 non-null  int64
 12  specobjid  10000 non-null  float64
 13  class      10000 non-null  object
 14  redshift   10000 non-null  float64
 15  plate      10000 non-null  int64
 16  mjd        10000 non-null  int64
 17  fiberid    10000 non-null  int64
dtypes: float64(10), int64(7), object(1)
```

Además se analizan detalles del dataset para ver información relevante (eran de interés mínimos y máximos principalmente):

	count	mean	std	min	25%	50%	75%	max
objid	10000.0	1.237650e+18	1.577039e+05	1.237650e+18	1.237650e+18	1.237650e+18	1.237650e+18	1.237650e+18
ra	10000.0	1.755300e+02	4.778344e+01	8.235100e+00	1.573709e+02	1.803945e+02	2.015473e+02	2.608844e+02
dec	10000.0	1.483615e+01	2.521221e+01	-5.382632e+00	-5.390350e-01	4.041660e-01	3.564940e+01	6.854227e+01
u	10000.0	1.861936e+01	8.286560e-01	1.298897e+01	1.817804e+01	1.885309e+01	1.925923e+01	1.959990e+01
g	10000.0	1.737193e+01	9.454572e-01	1.279955e+01	1.681510e+01	1.749513e+01	1.801015e+01	1.991897e+01
r	10000.0	1.684096e+01	1.067764e+00	1.243160e+01	1.617333e+01	1.685877e+01	1.751268e+01	2.480204e+01
i	10000.0	1.658358e+01	1.141805e+00	1.194721e+01	1.585370e+01	1.655499e+01	1.725855e+01	2.817963e+01
z	10000.0	1.642283e+01	1.203188e+00	1.161041e+01	1.561829e+01	1.638995e+01	1.714145e+01	2.283306e+01
run	10000.0	9.810348e+02	2.733050e+02	3.080000e+02	7.520000e+02	7.560000e+02	1.331000e+03	1.412000e+03
rerun	10000.0	3.010000e+02	0.000000e+00	3.010000e+02	3.010000e+02	3.010000e+02	3.010000e+02	3.010000e+02
camcol	10000.0	3.648700e+00	1.666183e+00	1.000000e+00	2.000000e+00	4.000000e+00	5.000000e+00	6.000000e+00
field	10000.0	3.023801e+02	1.625778e+02	1.100000e+01	1.840000e+02	2.990000e+02	4.140000e+02	7.680000e+02
specobjid	10000.0	1.645022e+18	2.013998e+18	2.995780e+17	3.389248e+17	4.966580e+17	2.881300e+18	9.468830e+18
redshift	10000.0	1.437257e-01	3.887740e-01	-4.136078e-03	8.090000e-05	4.259070e-02	9.257851e-02	5.353854e+00
plate	10000.0	1.460986e+03	1.788778e+03	2.660000e+02	3.010000e+02	4.410000e+02	2.559000e+03	8.410000e+03
mjd	10000.0	5.294353e+04	1.511151e+03	5.157800e+04	5.190000e+04	5.199700e+04	5.446800e+04	5.748100e+04
fiberid	10000.0	3.530694e+02	2.062981e+02	1.000000e+00	1.867500e+02	3.510000e+02	5.100000e+02	1.000000e+03

Función “describe” para ver detalles del dataset

De estas primeras observaciones (y el análisis de algunas columnas que no son agregados a este documento por una cuestión de extensión) se concluye qué columnas tienen o no relevancia en el dataset. Luego, se hace una primer limpieza, directamente borrando las columnas “**objid**”, “**specobjid**”, “**run**”, “**rerun**”, “**camcol**”, y “**field**”.
El dataset ahora está listo para ser trabajado y pasa a tener este aspecto:

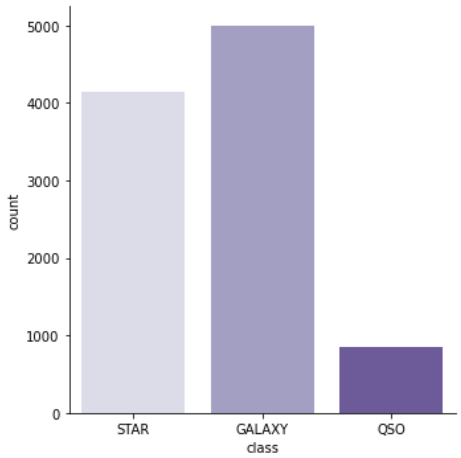
	ra	dec	u	g	r	i	z	class	redshift	plate	mjd	fiberid
0	183.531326	0.089693	19.47406	17.04240	15.94699	15.50342	15.22531	STAR	-0.000009	3306	54922	491
1	183.598371	0.135285	18.66280	17.21449	16.67637	16.48922	16.39150	STAR	-0.000055	323	51615	541
2	183.680207	0.126185	19.38298	18.19169	17.47428	17.08732	16.80125	GALAXY	0.123111	287	52023	513
3	183.870529	0.049911	17.76536	16.60272	16.16116	15.98233	15.90438	STAR	-0.000111	3306	54922	510
4	183.883288	0.102557	17.55025	16.26342	16.43869	16.55492	16.61326	STAR	0.000590	3306	54922	512
5	183.847174	0.173694	19.43133	18.46779	18.16451	18.01475	18.04155	STAR	0.000315	324	51666	594
6	183.864379	0.019201	19.38322	17.88995	17.10537	16.66393	16.36955	GALAXY	0.100242	287	52023	559
7	183.900081	0.187473	18.97993	17.84496	17.38022	17.20673	17.07071	STAR	0.000315	3306	54922	515
8	183.924588	0.097246	17.90616	16.97172	16.67541	16.53776	16.47596	STAR	0.000089	323	51615	595
9	183.973498	0.081626	18.67249	17.71375	17.49362	17.28284	17.22644	GALAXY	0.040508	288	52000	400
10	183.979195	0.135998	19.29772	17.80227	17.18266	16.92335	16.79928	STAR	-0.000035	3306	54922	506
11	184.085331	0.112110	18.83307	17.51785	16.94273	16.71418	16.60521	STAR	0.000623	3306	54922	547
12	184.102098	0.191511	19.56250	18.19113	17.65759	17.47573	17.39203	STAR	0.000055	3306	54922	544
13	184.160510	0.075645	19.57990	17.72815	16.98740	16.68076	16.50426	STAR	0.000008	3306	54922	546
14	184.189574	0.099482	19.25667	17.54869	16.63578	16.14922	15.76639	GALAXY	0.072087	288	52000	389

Dataset pasado en limpio

EXPLORATORY DATA ANALYSIS (EDA)

Columna Class

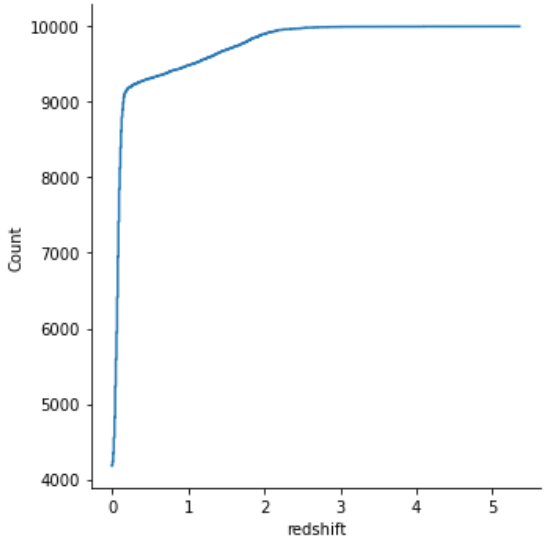
Las columnas “class” y “redshift” son las principales del dataset. Cualquier error en ellas provocaría que el posterior uso del modelo sea inefectivo. Comienzo viendo si la columna contiene algún valor anómalo (es decir, si tiene algún objeto que no sea estrella, galaxia, o cuásar), y que no haya una marcada diferencia entre la cantidad de valores de cada clase.



Afortunadamente, no es el caso. No hay valores anómalos y si bien hay menos objetos de tipo QSO (cuásar), esto se debe a que son más raros de observar. Esto me indica que no hay problemas en esta columna.

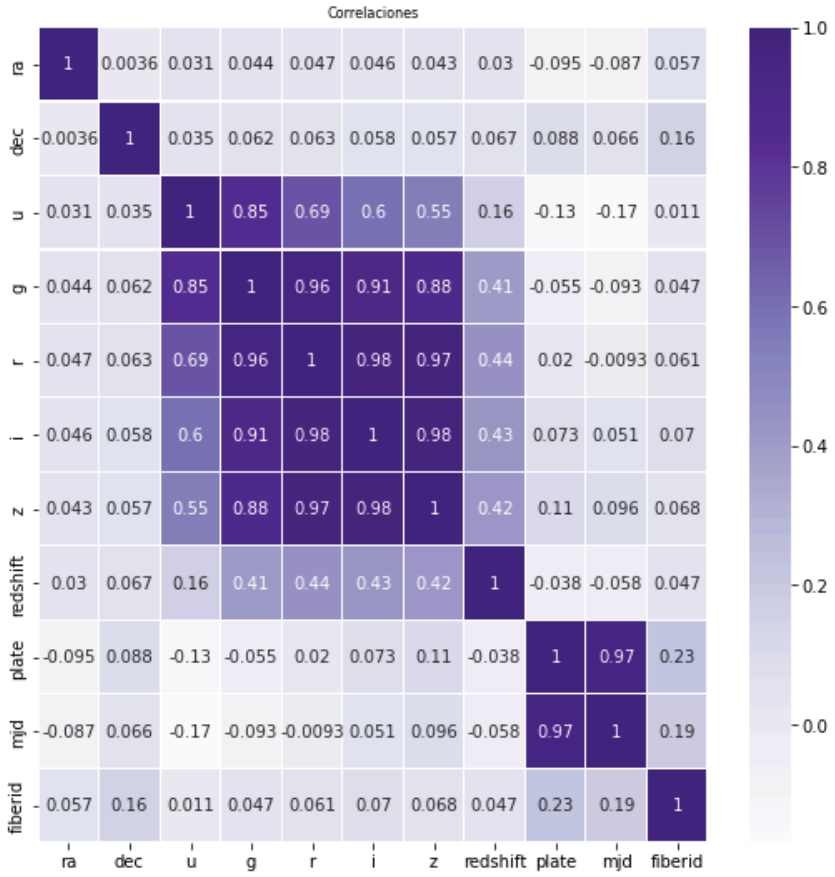
Columna Redshift

Cuando un histograma de los redshifts probó ser ineficiente a la hora de comunicar la cantidad de valores, busqué una mejor forma de visualizar su distribución. Este gráfico indica que la mayoría de los redshifts están entre 0 y 2, con mayor concentración de 0 a 0.3 (valor aproximado):



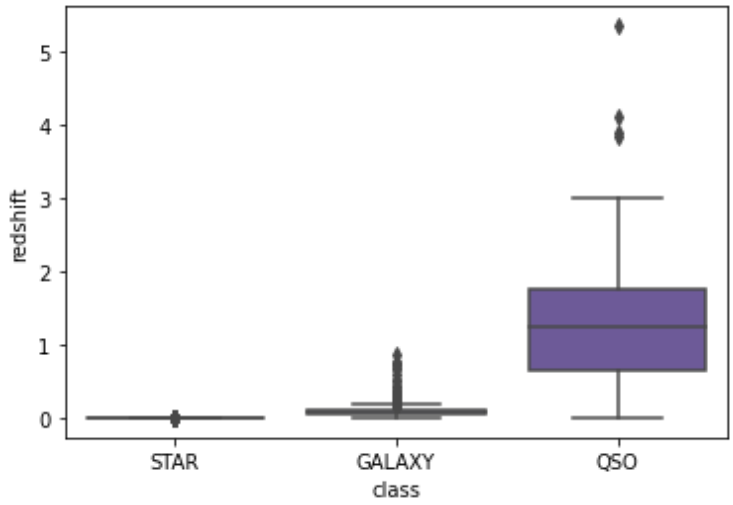
Correlaciones

Ya trabajando con un análisis que relacione variables, se obtuvo la matriz de correlaciones de los datos, sintetizada en el siguiente *heatmap*, donde los colores más oscuros corresponden a una mayor correlación:



Redshift vs Class

Busco relaciones entre las columnas más importantes del dataset utilizando un gráfico *boxplot*:

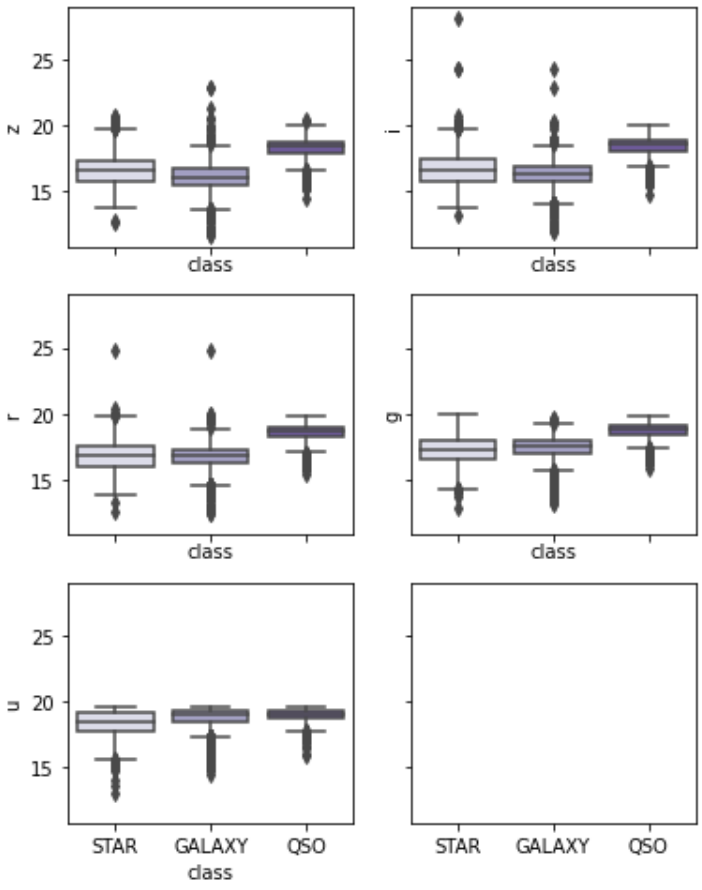


Redshift de los objetos según su clase

Se observa que la distribución de los redshifts es bastante clara: las estrellas poseen los valores más bajos, seguidas de las galaxias, y por último los cuásares (QSO) presentan los valores más altos.
Se pueden sacar de este gráfico conclusiones tempranas pero muy importantes: teniendo en cuenta que la medida del redshift puede ser traducida a una medida de distancia, se entiende que las estrellas observadas son los objetos más cercanos, luego las galaxias, y por último los cuásares serán los objetos más alejados. Esta conclusión es importante, porque es congruente con lo que sucede en la realidad. Además, por su cercanía menor a otras galaxias, podemos intuir que seguramente las estrellas observadas corresponden, en su mayoría, a estrellas pertenecientes a la Vía Láctea.

Filtros u, g, r, i, z

Como se ha visto antes, las columnas "**u**", "**g**", "**r**", "**i**", y "**z**" del dataset corresponden a filtros usados en el telescopio para capturar la luz de los objetos. Se realizan gráficos boxplot que analicen la relación entre estos filtros:

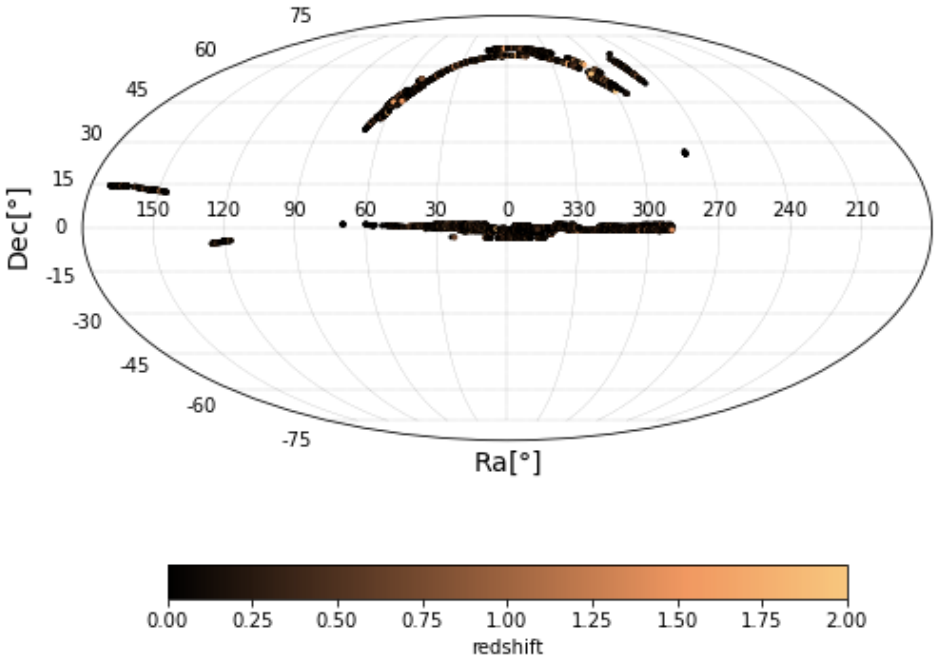


Boxplot de los filtros u g r i z. Ignorar el último cuadro en blanco.

Se ve que los gráficos para todas las variables son extremadamente similares, con excepción del correspondiente al filtro "**u**". Entonces, sería posible condensar toda esta información: creando una columna con los valores de los filtros **g r i z** , y otra columna para el valor de **u**, que seguiría siendo independiente. Últimamente se optó por no tomar esta ruta, pero la información que se perdería de haberlo hecho sería mínima, y conllevaría un tiempo de análisis menor para el algoritmo de Machine Learning en pasos siguientes.

Coordenadas Ecuatoriales

En un primer intento para comenzar a interpretar mejor esta información espacialmente hablando, y para comenzar a hacer gráficos un poco más complejos, se recurrió a un gráfico de los objetos según sus coordenadas ecuatoriales (información que sacamos de las columnas "**ra**" y "**dec**"), en una proyección *Mollweide*:



Proyección Mollweide, distribución de los objetos según sus coordenadas ecuatoriales (con información de redshift)

Si bien este gráfico presenta información útil, no es el objetivo al que se apunta, que es una representación la distribución de los objetos en base a su distancia a la Tierra. En todo caso, este gráfico representa a los objetos según su distribución en la esfera celeste - es decir, su distribución vista desde la Tierra. Esta información es un poco rebuscada y difícil de ver a simple vista, especialmente si no se está en tema o acostumbrado a pensar en coordenadas ecuatoriales -que vale la pena reconocer no es algo que suceda frecuentemente.

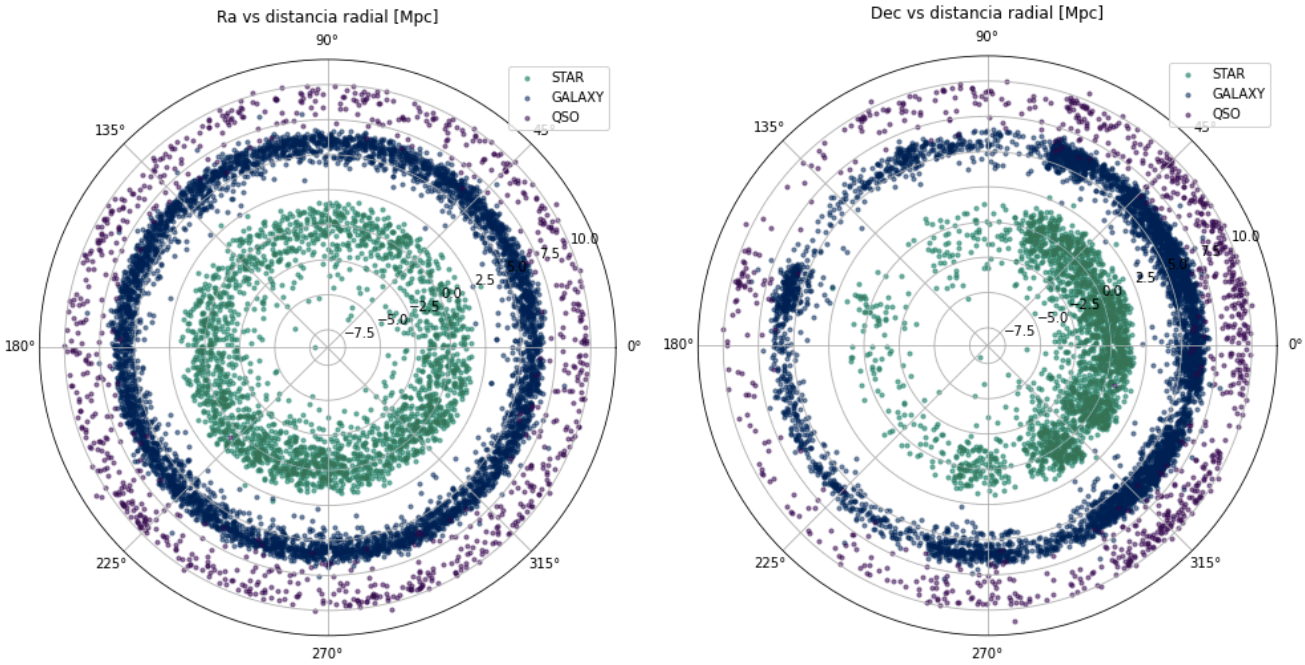
Por lo tanto, teniendo en cuenta todos estos factores, se optó por intentar una visualización diferente.

Coordenadas Polares

Como se vio antes, la clave para solucionar este problema radica en pasar el redshift a distancia. Para eso hay que utilizar alguna librería externa. En este caso:

```
# Librería que me permite pasar redshift a distancia
from astropy.cosmology import WMAP9 as cosmo
```

Utilizando este recurso, se puede obtener la distancia radial de cada objeto, y utilizando además su ascensión recta o declinación, es posible obtener los siguientes gráficos:



Es importante aclarar que conseguir estos gráficos, especialmente el de **dec vs distancia radial** (el de la derecha), era uno de los objetivos principales de este proyecto.

Desafío 3D

Al momento de exponer este trabajo en clase, surgió la idea de graficar la ubicación de los objetos en un gráfico 3D, y acepté el desafío. Al principio pensé que iba a ser algo desafiante pero lograble, y luego encontré el primer inconveniente: las coordenadas. Los valores de **ra**, **dec**, y **distancia** (en base a redshift) cumplen el papel de θ , φ y r en coordenadas esféricas respectivamente; y no hay gráficos de *seaborn* que utilicen coordenadas esféricas, sólo hay posibilidad de hacerlos en coordenadas cartesianas (x y z). Pensé que seguramente podría transformarlas (ayudándome de Internet y trigonometría), pero resultó ser más complicado de lo esperado puesto que no son exactamente lo mismo que coordenadas esféricas.
Además, está el problema de las distancias: la mayoría de los objetos se encuentran muy espaciados unos de otros, así que el impacto visual se perdería un poco.
En resumen: lo intenté, pero no resultó.

MODELO DE MACHINE LEARNING

Variable Objetivo y Modelos

Como se ha anticipado, el objetivo principal de la parte correspondiente a Machine Learning (ML) es el de utilizar un modelo que me permita determinar la clase de objeto observado en base a las demás variables. Dicha información se encuentra resumida en la columna "**class**", que será nuestra variable objetivo. Se pretende que el modelo usado indique si el objeto es una estrella, una galaxia, o un cuásar.
Antes de comenzar, será necesario pasar la columna "**class**" a valores numéricos. Lo hago con una función simple que utiliza un condicional **if**:

```
def en_numeros(x):
    if x =='STAR':
        return 0
    elif x =='GALAXY':
        return 1
    else:
        return 2
```

Los modelos a estudiar para llevar a cabo este proceso serán:

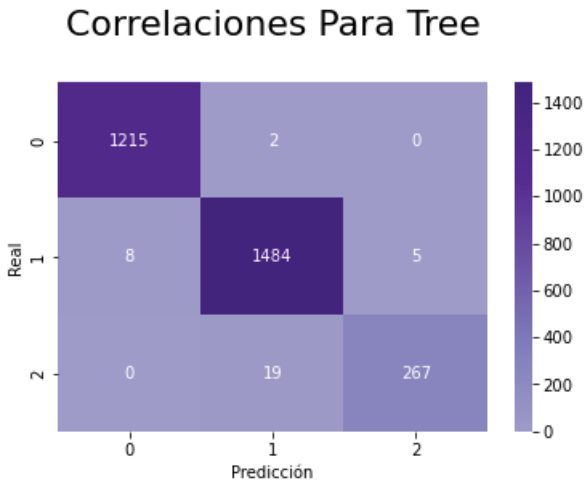
- Tree Classifier
- Forest Classifier
- KNN

Fue imprescindible para llegar a esta instancia lograr una limpieza del dataset, y definir variables importantes. Luego, se compararán los resultados de los modelos para seleccionar el mejor.

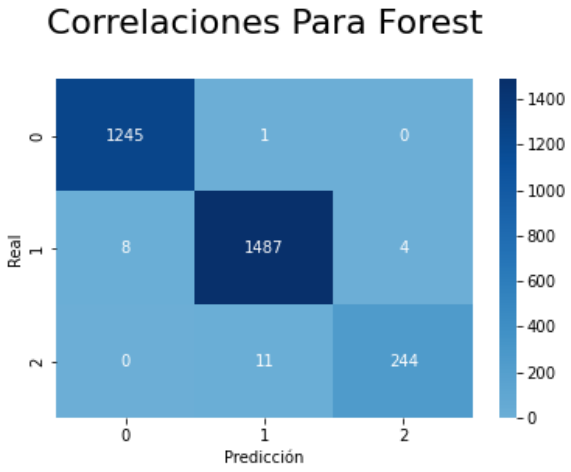
Accuracy y Matrices de Confusión

Una vez hechos y entrenados los modelos (cuyo desarrollo se encuentra en el archivo *notebook* adjunto a este documento), se obtienen los siguientes resultados de accuracy, acompañados de las matrices de confusión de cada modelo:

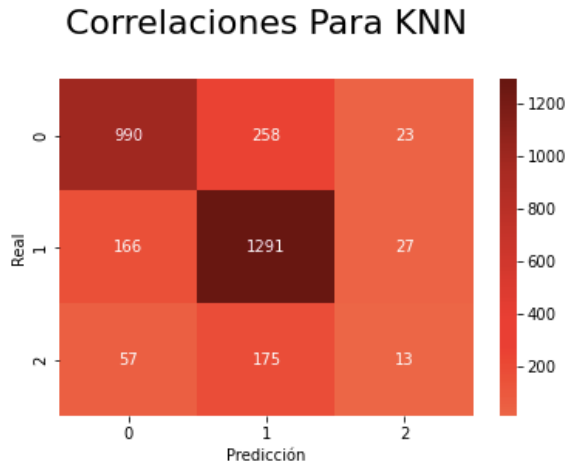
```
TREE ACCURACY: 0.9886666666666667
```



```
FOREST ACCURACY: 0.992
```




```
KNN ACCURACY: 0.7646666666666667
```



Tanto la matriz de confusión como el accuracy de los modelos deja en evidencia que el modelo KNN es el menos útil para este problema, y que los modelos de Tree y Forest tienen un porcentaje muy alto de certeza, con el modelo **Forest** siendo superior.

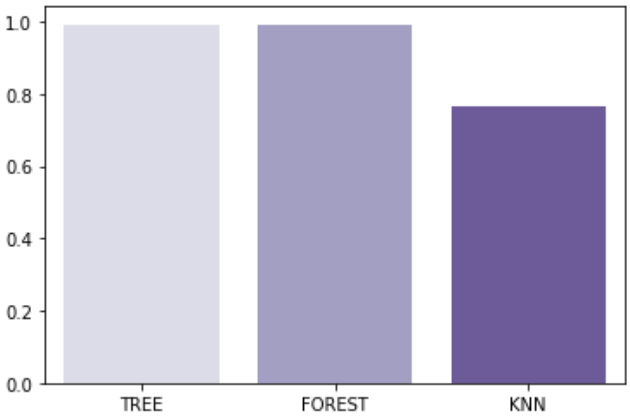
Métricas de los Modelos

Tener el parámetro de precisión de cada modelo es importante pero no suficiente. Agrego también el cálculo de los parámetros Recall y F1 (que combina las medidas de precisión y recall de los modelos en una sola). Los resultados son los siguientes:

```
TREE RECALL: 0.9886666666666667
FOREST RECALL: 0.992
KNN RECALL: 0.7646666666666667
```

```
TREE F1: 0.9886666666666667
FOREST F1: 0.992
KNN F1: 0.7646666666666667
```

Las medidas de Precisión, Recall, y (por tanto) F1 de los tres modelos son exactamente iguales. Ya que son las mismas, incluyo el gráfico de una sola:



En el eje Y podrían ir Precisión, Recall, o F1. El gráfico es el mismo para todos estos parámetros.

Como todo lo demás en este documento, el código puede revisarse en la *notebook* adjunta.
Lo que me indican estos valores es que tanto el modelo de Tree o el Forest serán excelentes elecciones para el objetivo que busco, con el Forest siendo ligeramente superior -y por tanto preferido.

Además, al ser el valor de accuracy tan alto para el modelo de **Forest** (el elegido en base a todo lo discutido), pierde necesidad realizar optimizaciones en dicho modelo.

CIERRE

Futuras Líneas

Primero que nada, me gustaría mencionar que quizás con un poco más de paciencia y experiencia podría lograr la representación 3D que mencioné en la página 13, pero actualmente encuentro que es algo que me supera un poco.
Dicho eso, en cuanto a la implementación del modelo **Forest**, si bien es algo prácticamente imposible dado que el **SDSS** es un programa de investigación espacial totalmente ajeno a mí, me parece importante aclarar que fue entrenado con un 30% de los datos del dataset, y puesto que por tanto no tiene un overfitting a los datos, su implementación podría ser algo provechoso para la investigación -suponiendo que les es relevante.

Conclusiones

El resultado fue completamente satisfactorio y óptimo: primero, en cuanto a los objetivos generales, estoy orgulloso de haber podido realizar un trabajo así de principio a fin considerando que antes de iniciar el curso mis conocimientos de programación eran prácticamente nulos.

En cuanto a los objetivos específicos del trabajo, considero están cumplidos: el gráfico en coordenadas polares (página 13) resume la distribución de los objetos de forma clara y fácil de entender, y el modelo utilizado de Machine Learning (**Random Forest**) muestra métricas que indican su excelente rendimiento. En términos concretos, 24 errores de predicción en un dataset de 10000 es un error insignificante.