Informe de Resultados: Informe de resultado basado en un dataset astronómico con 3 modelos de algoritmos de aprendizajes distintos para poder verificar sus potenciales predicciones.

Dataset: https://github.com/matthieuvernier/INFO257_2020/blob/master/unidad1/datos/SDSS-DR14.csv

Modelos Usados:

RandomForest(RF) – Multiclase
SciKitLearn

Arbol de desicion(AD) – Multiclase
SciKitLearn

Regresión logistica Stars(RLS), con 1000 iteraciones:
Usando sigmoide como función de activación/evaluación.
Usando también gradiante descendiente en la optimizacion.
Incorporando intercepto.
Scikit-Learn(penalización l1, solver='liblinear')

Regresión logistica Galaxy(RLG), con 1000 iteraciones:
Usando sigmoide como función de activación/evaluación.
Usando también gradiante descendiente en la optimizacion.
Incorporando intercepto.
Scikit-Learn(penalización l1, solver='liblinear')

Regresión logistica QSO(RLQ), con 1000 iteraciones:
Usando sigmoide como función de activación/evaluación.
Usando también gradiante descendiente en la optimizacion.
Incorporando intercepto.
Scikit-Learn(penalización l1, solver='liblinear')

Regresión Logística(RL) – Multiclase
Scikit-Learn(penalización l1, solver='liblinear')

Pre- procesamiento: para los procesos de importación de los datos, sabemos que vienen estructurados en tabla de filas con columnas, por lo que verificamos si éstos pueden ser usados para los fines destinados al algoritmo de aprendizaje.

```
df2 = pd.read_csv("../datos/SDSS-DR14.csv")
df2.head()
```

	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
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
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
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
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

Una vez importados verificaremos que éstos datos estén en el formato correcto para nuestro uso:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 18 columns):
objid         10000 non-null float64
ra            10000 non-null float64
dec           10000 non-null float64
u             10000 non-null float64
g             10000 non-null float64
r             10000 non-null float64
i             10000 non-null float64
z             10000 non-null float64
run           10000 non-null int64
rerun         10000 non-null int64
camcol        10000 non-null int64
field         10000 non-null int64
specobjid     10000 non-null float64
class         10000 non-null object
redshift      10000 non-null float64
plate         10000 non-null int64
mjd           10000 non-null int64
fiberid       10000 non-null int64
dtypes: float64(10), int64(7), object(1)
memory usage: 1.4+ MB
```

```
df2.isnull().sum()
```

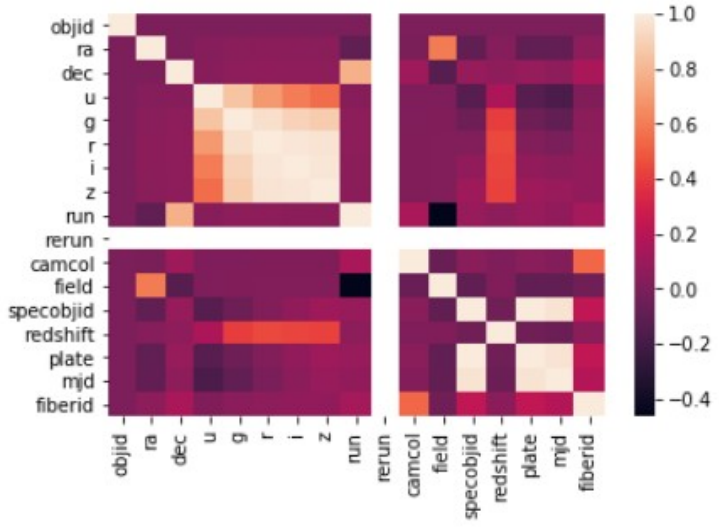
```
objid        0
ra           0
dec          0
u            0
g            0
r            0
i            0
z            0
run          0
rerun        0
camcol       0
field        0
specobjid    0
class        0
redshift     0
plate        0
mjd          0
fiberid      0
dtype: int64
```

Así podemos ver que tenemos una categoría “clase” de tipo no numérica, cual trabajaremos con ella más adelante, junto con ésto ademas verificamos que todo el conjunto no posea datos nulos.

Análisis de distribución:

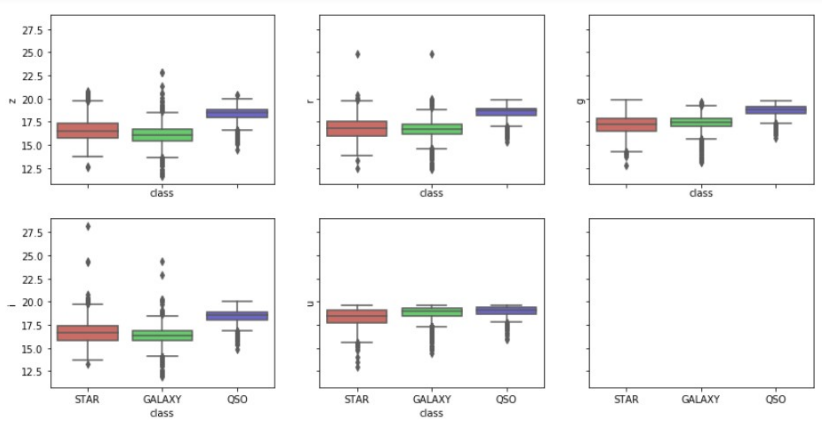
Revisaremos la correlación de datos, que nos permita verificar cierta dependencia entre las características con un mapa de calor:

```
sb.heatmap(df2.corr())
df2_data = df2
```

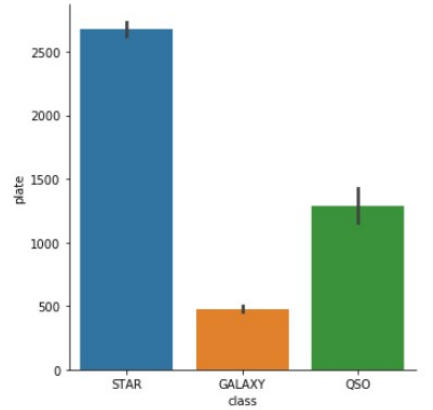


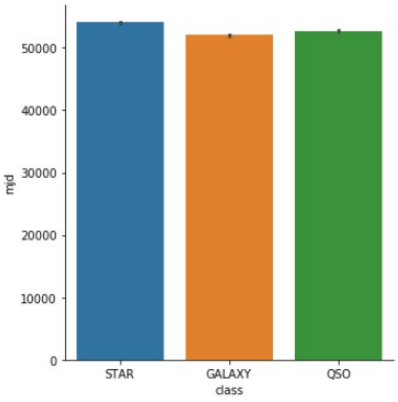
Pudimos visualizar que la categoría “rerun” poseía una dependencia fuerte con todas las otras variables, indicando que ésto nos puede generar problemas en la predicción de nuestros datos a la hora del aprendizaje, por lo que optamos en RandomFores, DesitionTree y sólo en Regresión logística de QSO eliminar ésta carácterística, para éste último modelo explicaremos más adelante la restriccón aplicada para éste criterio.

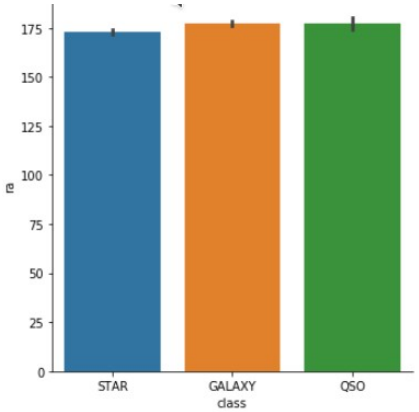
Basado en los datos anteriores, desplegamos los parámetros de luz para saber que tan alejadas están sus cotas medias, y extremos que nos digan si éstos nos puedan presentar algun problema de disparidad den los datos:



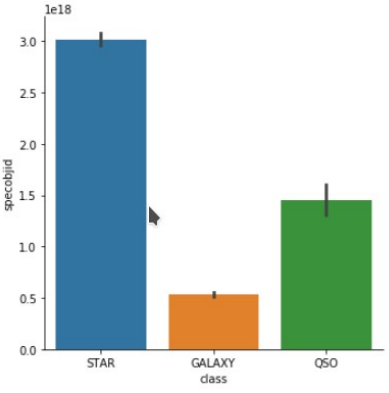
Según lo visto anteriormente también analisamos otras medidas para ver en que nos podían afectar en la categoría “plate”, “mjd”, “specobjid”, “ra” cuales en el gráfico de calor nos indican una dependencia directa entre algunas variables, por lo que revisaremos a través de las clases si éstas pueden ser un posible problema.







Para efectos del modelo aplicado a la



Regresión logística de QSO, eliminaremos las categorías “plate”, “specobjid”, debido que en este

modelo presenta problemas para la clasificación por observarse desbalance al final del proceso, donde efectivamente para éste modelo en específico eliminaremos los objetos de la clase “STAR”, ya que en éstas 2 caracteristicas presenta una evidente disparidad de datos permitiendonos, además hacer una comparación binaria.

Para los demás modelos aplicados RF, AD sólo se transformaron las variables categóricas a numéricas para usarlas en procesos multiclase, en cuanto al modelo RLS quitamos las características y clases de QSO y las convertimos en dummies las restantes, eliminamos las clases GALAXY de las dummies y ocupamos las categorias dummies de estrellas para comparar, éste método lo ocupamos para poder conservar las carácteristicas de la clase galaxia así no disminuyendo el dataset a entrenar.

Para RLG se usó la misma metodología pero sin eliminar las caracterísiticas de QUASAR dejándo solo la clase GALAXIA para clasificar.

Para RLQ eliminamos las caracterisitas y clases de “Estrella” quedándonos con los datos de galaxia y QUASAR para poder clasficar, según los ya filtrados anteriormente.

Proceso de Entrenamiento y evaluación:

Random Forest: se escoge un tamaño para test un 25% de los datos y un 75 para entrenamiento.

Clasificación:

GALAXIA = 0
ESTRELLA= 1
QUASAR= 2

```
[[1199    3    6]
 [   4 1087    0]
 [  23    0  178]]
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      1208
           1       1.00      1.00      1.00      1091
           2       0.97      0.89      0.92       201

    accuracy                           0.99      2500
   macro avg       0.98      0.96      0.97      2500
weighted avg       0.99      0.99      0.99      2500
```

F1 Score, nos indica que los datos presentados, no presentaron problemas de desbalance por clase.
Precision: nos indica que tan agrupados en la clasificación óptima se encuentran los datos, por lo que nos entregan buenos resultados.
Recall: nos dice el % de datos correctamente seleccionados por cada clase, cual sin problemas selecciona.
Accuracy: la correlación de los datos nos dice que están fuertemente apegados al modelo de clasificacion

Desition Tree: se escoge un tamaño para test un 25% de los datos y un 75 para entrenamiento.

Clasificación:

GALAXIA = 0
ESTRELLA= 1
QUASAR= 2

```
[[1194    2   12]
 [   6 1083    2]
 [  22    0  179]]
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      1208
           1       1.00      0.99      1.00      1091
           2       0.93      0.89      0.91       201

    accuracy                           0.98      2500
   macro avg       0.97      0.96      0.96      2500
weighted avg       0.98      0.98      0.98      2500
```

F1 Score: para cada clase que fueron descubiertas no encuentra problemas de desbalance.
Precision: se logra obtener un buen % de agrupación de datos relacionados a la solución.
Recall: nos presenta un buen % aislando un poco la clase 2, aún así se logra clasificar sin problemas.
Accuracy: Nos indica de igual manera que los datos están fuertemente correlacionados al modelo.

RLS: iteraciones = 1000, función=sigmoide, optimizador= gradiante descendiente
Clasificación:
Estrella=1
No Estrella=2

```
[[4857  141]
 [1029 3123]]
              precision    recall  f1-score   support

           0       0.83      0.97      0.89      4998
           1       0.96      0.75      0.84      4152

    accuracy                           0.87      9150
   macro avg       0.89      0.86      0.87      9150
weighted avg       0.88      0.87      0.87      9150
```

F1 Score: el % de desbalance no es bajo para el conjunto de datos que se sesgaron anteriormente, aún así no presenta inconvenientes en la validación.
Precision: en la relación de los datos no presenta problemas para la clase estrella.
Recall: nuestro % de encontrar positivos la NO estrella es alto, x lo que nos ayuda a discriminar nuestra clasificación, la precisión de nuestros datos falsos.
Accuracy: el % nos indica que la correlación de los datos con respecto al modelo es aceptable

RLS: Iteraciones=1000, penalizacion= l1, solver='liblinear'
Estrella=1
No Estrella=2

```
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      1498
           1       0.99      1.00      0.99      1247

    accuracy                           0.99      2745
   macro avg       0.99      0.99      0.99      2745
weighted avg       0.99      0.99      0.99      2745
```

F1 Score: para cada clase que fueron descubiertas no encuentra problemas de desbalance.
Precision: se logra obtener un buen % de agrupación de datos relacionados a la solución.
Recall: nos presenta un buen % aislando un poco la clase 1, aún así se logra clasificar sin problemas.
Accuracy: Nos indica de igual manera que los datos están fuertemente correlacionados al modelo.

RLG: iteraciones = 1000, función=sigmoide, optimizador= gradiante descendiente
Clasificacion:
Galaxia=1
No Galaxia=0

```
[[3584 1418]
 [1001 3997]]
              precision    recall  f1-score   support

           0       0.78      0.72      0.75      5002
           1       0.74      0.80      0.77      4998

    accuracy                           0.76     10000
   macro avg       0.76      0.76      0.76     10000
weighted avg       0.76      0.76      0.76     10000
```

F1 Score: el % de balance para la concentración de los datos muestra ser considerable, lo podemos atribuir a que las carácterísticas de estrellas mostraban más predominancia por sobre las galaxias.
Precision: en la relación de los datos indican ser aceptables en la clasificacion de Galaxia como también como para determinar resultados que no pertenezcan al conjunto.
Recall: nuestro % de encontrar positivos presentada muestra ser aceptables para los conjuntos que deseamos clasificar.
Accuracy: el % nos indica que la correlación de los datos con respecto al modelo es aceptable

RLG: Iteraciones=1000, penalizacion= l1, solver=’liblinear’

Clasificacion:

Galaxia=1

No Galaxia=0

	precision	recall	f1-score	support
0	0.94	0.87	0.91	1292
1	0.88	0.95	0.91	1208
accuracy			0.91	2500
macro avg	0.91	0.91	0.91	2500
weighted avg	0.91	0.91	0.91	2500

F1 Score: para cada clase que fueron descubiertas no encuentra problemas de desbalance.
Precision: se logra obtener un buen % de agrupación de datos relacionados a la solución.
Recall: nos presenta un buen % aislando un poco la clase 1, aún así se logra clasificar sin problemas.
Accuracy: Nos indica de igual manera que los datos están fuertemente correlacionados al modelo.

RLQ: iteraciones = 1000, función=sigmoide, optimizador= gradiante descendiente

Clasificación:

Quasar=1

No Quasar=0

```
[[4998    0]
 [ 850    0]]
```

	precision	recall	f1-score	support
0	0.85	1.00	0.92	4998
1	0.00	0.00	0.00	850
accuracy			0.85	5848
macro avg	0.43	0.50	0.46	5848
weighted avg	0.73	0.85	0.79	5848

F1 Score: el % de balance para la concentración de los datos muestra inconsistente, lo podemos atribuir a que las carácterísticas de galaxias mostraban más predominancia por sobre las Quasares, no muestra indicios de mostrarse para detectar Quasares

Precision: en la relación de los datos indican ser aceptables en la clasificacion de No Quasares, indicando que el desbalance para detectarlos no muestra resultado alguno

Recall: la ventaja de nuestro modelo es que según nuestro modelo podemos clasificar con datos positivos los No Quasares, ya que no presenta alguno como detectado.

Accuracy: el % nos indica que la correlación de los datos con respecto al modelo es aceptable.

RLQ: Iteraciones=1000, penalizacion= l1, solver=’liblinear’

Clasificación:

Quasar=1

No Quasar=0

	precision	recall	f1-score	support
0	0.98	1.00	0.99	1512
1	0.97	0.88	0.92	243
accuracy			0.98	1755
macro avg	0.98	0.94	0.96	1755
weighted avg	0.98	0.98	0.98	1755

F1 Score: para cada clase que fueron descubiertas no encuentra problemas de desbalance.
Precision: se logra obtener un buen % de agrupación de datos relacionados a la solución.
Recall: nos presenta un buen % aislando un poco la clase 1, aún así se logra clasificar sin problemas.
Accuracy: Nos indica de igual manera que los datos están fuertemente correlacionados al modelo.

Regresión Logistica: Iteraciones=1000, penalizacion= l1, solver=’liblinear’

Clasificación:

GALAXIA = 0

ESTRELLA= 1

QUASAR= 2

```
[[1429   20    4]
 [  14 1286    0]
 [  23    0  224]]
```

	precision	recall	f1-score	support
0	0.97	0.98	0.98	1453
1	0.98	0.99	0.99	1300
2	0.98	0.91	0.94	247
accuracy			0.98	3000
macro avg	0.98	0.96	0.97	3000
weighted avg	0.98	0.98	0.98	3000

F1 Score: para cada clase que fueron descubiertas no encuentra problemas de desbalance.
Precision: se logra obtener un buen % de agrupación de datos relacionados a la solución.
Recall: nos presenta un buen % aislando un poco la clase 2, aún así se logra clasificar sin problemas.
Accuracy: Nos indica de igual manera que los datos están fuertemente correlacionados al modelo.

Comparaciones y Analisis de los modelos usados

Según los modelos que ocupamos evidentemente, los modelos basados en árboles nos da mejores resultados debido que la metodología a traves de ganancias probabilísitcas con soporte de preguntas dan muchas mejores chances de identificar datos que puedan ser sesgados por parámetros o aproximacione numéricas. A diferencia como lo muestran los modelos logísticos, que si bien se tuvo que separar en distintos modelos para verificar su clasificacion individualmente, éstos resultó dificil para encotrar una solución factible y aproximada, que además dependían mucho de la cantidad de iteraciones, la funcion de evaluación usada y de estrategia usar para optimizar, debido que se depende mucho de los parámetros o pesos que se aprenden para ajustarse al modelo de regresión.

Si bien fue algo que no consideramos desde un principio al modelar la regresión logística, finalmente pudimos cambiar éstos mismos criterios para evaluar los modelos a través de scikit-learn cambiando dichos criterios para clasificar, entregándonos resultados mucho más satisfactorios de lo esperado, mostrando mejoras notables para detectar Quasar. Aunque se debe acotar que para el proceso de clasificacion la funcion a utilizar de Library for Large Linear Classification, indican que para gran cantidad de datos el rendimiento puede verse afectado.