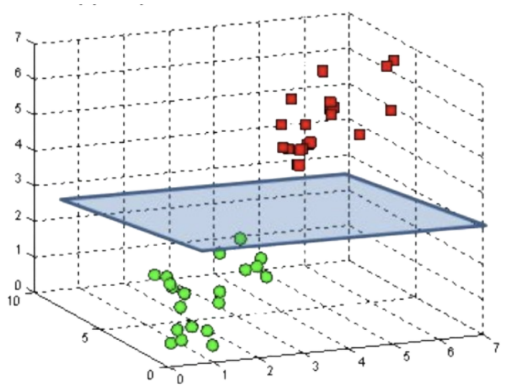
Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Tecnicatura Universitaria en Inteligencia Artificial

Clasificación Supervisada y Métodos de Ensamble



TUIA - IA 4.3 Minería de Datos

Trabajo Práctico Nro. 3

***Fecha*:** 25/11/2023

Docentes:

- *Flavio Spetale*
- *Dolores Biondo*
- *Facundo Vasquez*
- *Sofia Errecarte*

Grupo:

- *Revello Simon*
- *Giampaoli Fabio*

Objetivo:

El objetivo de este trabajo práctico es integrar los conocimientos adquiridos en las unidades 5 (Clasificación supervisada) y 6 (Métodos de Ensamble) en un problema real asociado a la determinación del color de los granos de café mediante la medición atributos característicos

Actividades:

1. Descargar el conjunto de CoffeeRatings.csv, para realizar el trabajo práctico. Analizar los atributos del conjunto de datos (distribuciones, valores, outliers, tipos de datos, etc.).

2. Realizar la predicción del atributo Color utilizando máquinas de vectores con kernel lineal analizando el parámetro costo. Mostrar los resultados sobre los conjuntos de test (Precisión, Exhaustividad y Exactitud) utilizando validación cruzada con k =5.

3. Realizar la predicción del atributo Color utilizando máquinas de vectores con kernel gaussiano analizando los parámetros costo y gama. Mostrar los resultados sobre los conjuntos de test (Precisión, Exhaustividad y Exactitud) utilizando validación cruzada con k=5.
4. Realizar la predicción del atributo Color utilizando Random Forest analizando los parámetros cantidad de estimadores y la máxima profundidad de los árboles. Mostrar los resultados sobre los conjuntos de test (Precisión, Exhaustividad y Exactitud) utilizando validación cruzada con k =5.

Resolución

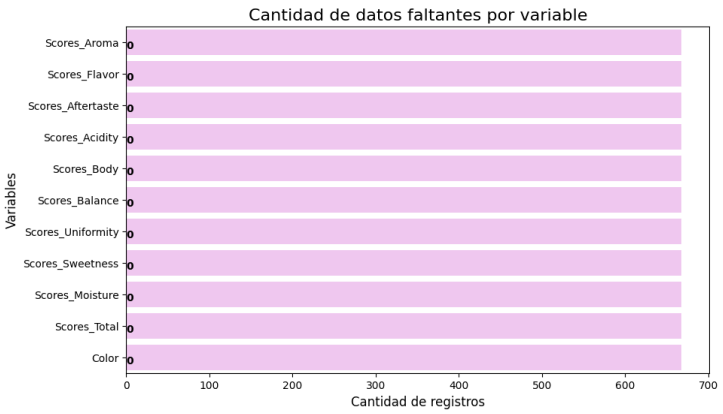
El dataset cuenta inicialmente con un total de 835 filas y 11 columnas. Cada columna representa una variable numérica discreta que representan puntajes de cada característica de un tipo de cadena particular. Pero también posee una variable categórica de tres clases, ‘Color’, que representa el color del grano del café.

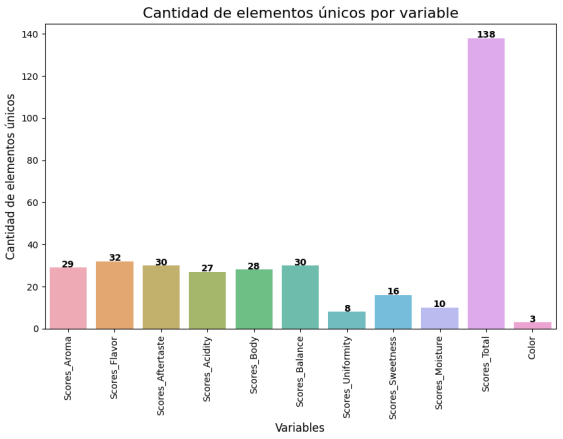
Planteamos la división inicial del conjunto de datos, con el fin de no perturbar el conjunto de validación que deberíamos tratar como si no lo conociéramos de antemano, para que el modelo de clasificación no se contamine con datos que todavía no conoce. Por ello, se dividen las variables explicativas y la variable objetivo (Color), y de ellos se toma el 80% de los registros para entrenamiento del modelo, y el restante para validación de las predicciones.

Esta es una muestra del conjunto de entrenamiento (concatenación de X e y):

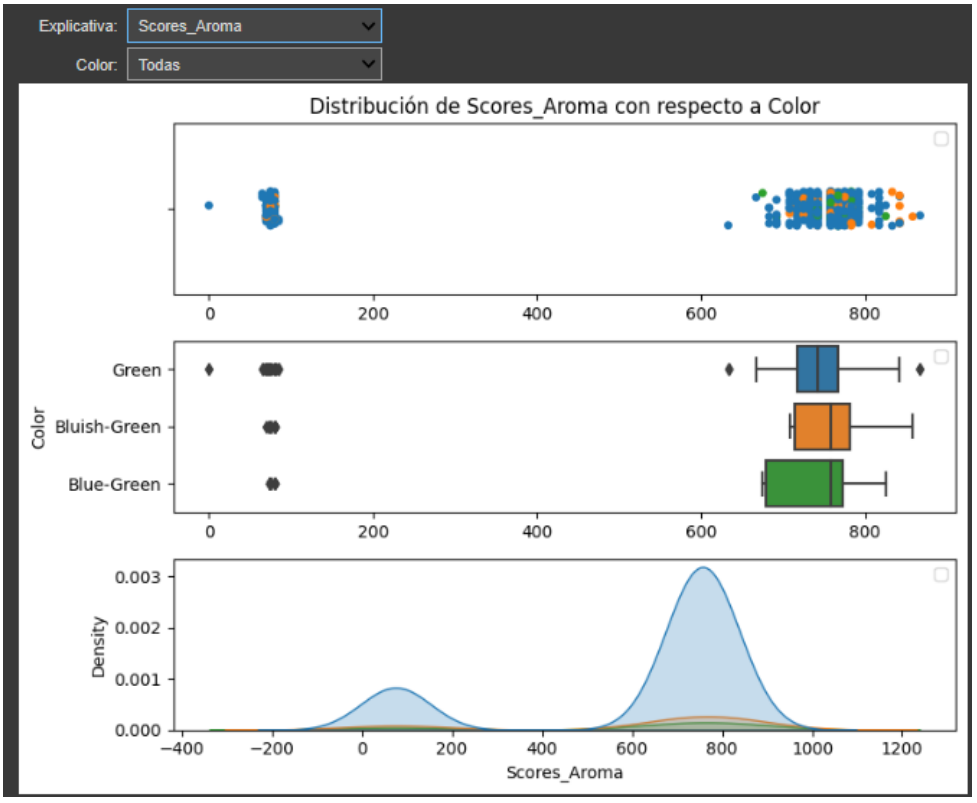
	Scores_Aroma	Scores_Flavor	Scores_Aftertaste	Scores_Acidity	Scores_Body	Scores_Balance	Scores_Uniformity	Scores_Sweetness	Scores_Moisture	Scores_Total	Color
478	758	775	75	758	758	758	100	100	11	8317	Green
346	742	733	75	717	708	833	100	100	1	8233	Green
462	767	767	767	792	75	767	100	100	0	8375	Green
691	792	808	792	808	808	783	100	100	1	8592	Green
302	75	767	767	75	783	717	100	100	11	8333	Green

Notamos primeramente que los datos están pre procesados, ya que no hay valores nulos en el dataset (imagen izquierda). Además podemos notar las proporciones de valores únicos de cada variable (imagen derecha)::



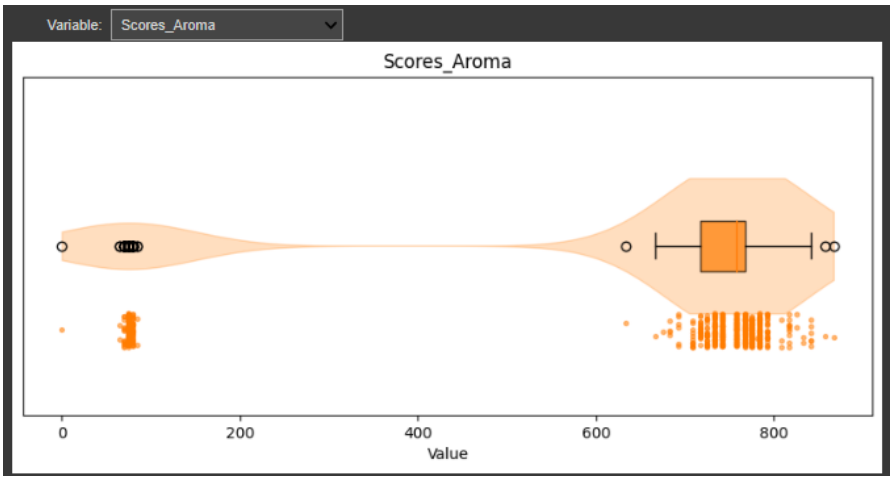


Como parte del análisis descriptivo, podemos comprender la distribución de las variables y de cada clase con un gráfico interactivo que permite la elegir la variable de interés a lo largo del eje x, y la o las clases que se desean visualizar:

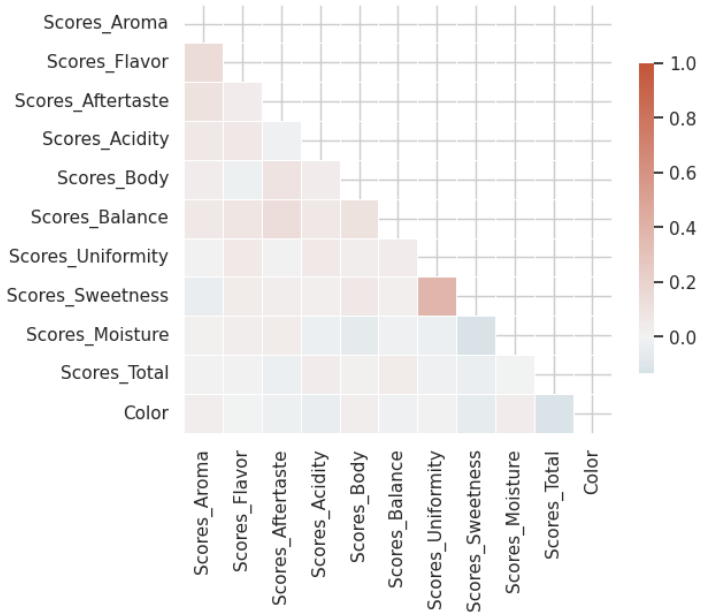


Notamos en su mayoría que las variables cumplen esta distribución de muchos valores concentrados en valores altos o bajos, y algunos grupos de registros en el lado extremo, pero en las tres clases a la vez. Esto lo notamos mejor en un distribuciones de las variables considerando todas las clases, nuevamente en un gráfico interactivo:

Esto sugiere que tendremos que hacer un tratamiento o consideraciones para los valores atípicos, debido a que su presencia perturba las distribuciones de las variables, y posiblemente las estimaciones del modelo de clasificación que no son robustos.



Una matriz de correlación indica la una baja correlación lineal entre todas las variables en general.



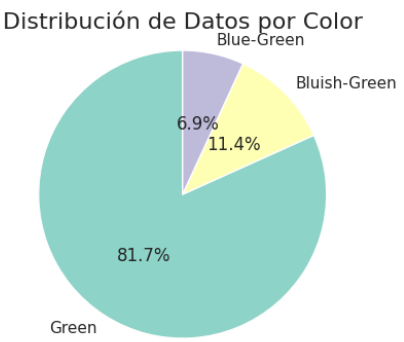
Para controlar que las transformaciones del dataset resulten eficientes y maximicen la explicabilidad de la variable objetivo, se crea un modelo base de clasificación con Soporoso Vector Machines con sus parámetros por default. De esta manera cada conjunto que se le pasa por parámetro obtiene métricas de validación para clasificación.

La imagen de la derecha muestra métricas base para el dataset sin procesamiento.

```
model = svm.SVC(probability=True)
train_and_evaluate(model, X_train, X_test, y_train, y_test)
```

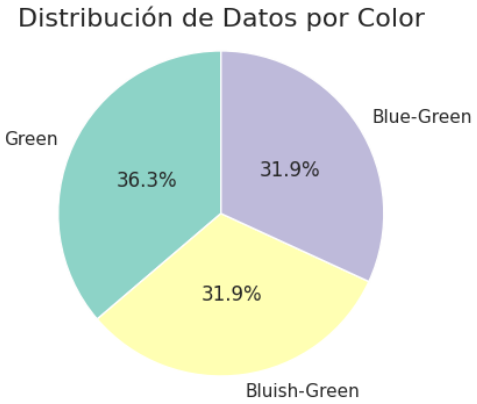
	metric	set	Green	Blue-Green	Bluish-Green
0	accuracy	Train	0.817365	0.817365	0.817365
0	accuracy	test	0.808383	0.808383	0.808383
2	auc	Train	0.566623	0.566623	0.566623
2	auc	test	0.583524	0.583524	0.583524
1	f1	Train	0.000000	0.000000	0.899506
1	f1	test	0.000000	0.000000	0.894040

Notamos principalmente que la combinación de la precisión y exhaustividad (f1) es muy grande para solo una de las clases, y nula para las restantes. Esto sugiere que hay un gran desbalance de las clases en la variable objetivo. Esto lo podemos notar rápidamente en un gráfico de proporciones.



Tratar el desbalance de las clases es importante para que el modelo de clasificación no tome decisiones sesgadas para predecir debido a la falta de entrenamiento en las clases minoritarias. Se propone la utilización de SMOTE para balancear el dataset mediante oversailing por cercanía de datos, aumentando el número de datos para las clases minoritarias a un número dado.

Pero en esta instancia, el balance no es favorable debido a que disminuye las métricas del modelo base, y además causa sobreajuste en los datos. Algo similar ocurre con la técnica de undersampling para balancear.



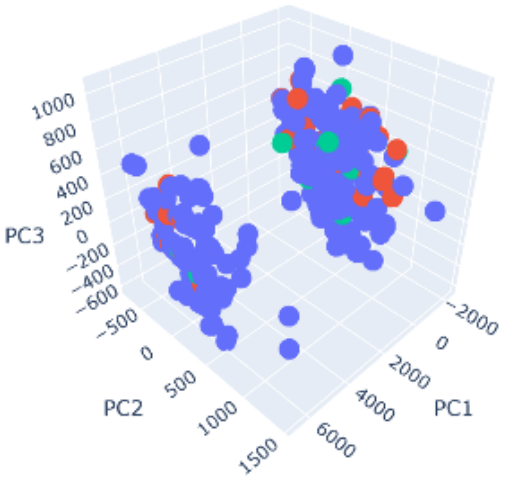

```
model_over = svm.SVC(probability=True)
train_and_evaluate(model_over, X_train_over, X_test, y_train_over, y_test)
```

	metric	set	Green	Blue-Green	Bluish-Green
0	accuracy	Train	0.491368	0.491368	0.491368
0	accuracy	test	0.311377	0.311377	0.311377
2	auc	Train	0.707485	0.707485	0.707485
2	auc	test	0.527066	0.527066	0.527066
1	f1	Train	0.593528	0.453191	0.375155
1	f1	test	0.101266	0.173913	0.451613

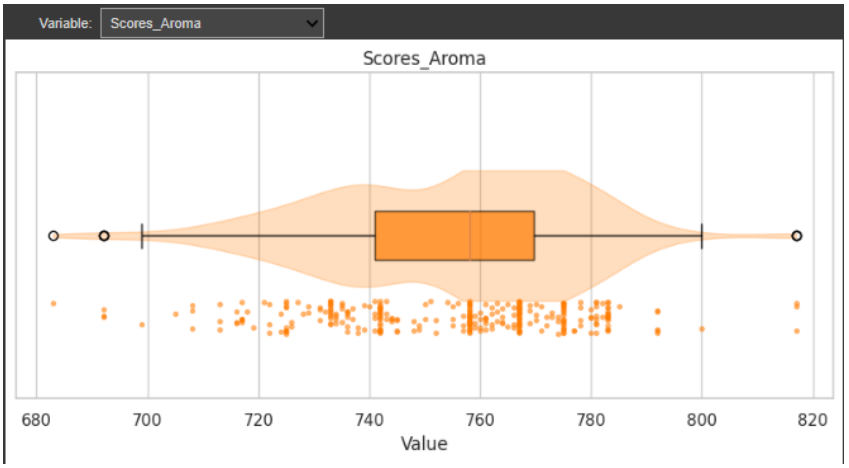
Por lo que se evita continuar este enfoque de tratar el desbalance.

Mediante tecnica de reduccion de dimensionalidad lineal como PCA, se pueden visualizar las distribucion de los datos, donde los diferentes colores representan las clases de Color.

Notamos nuevamente presencia de valores atípicos pertenecientes a las tres clases, y que parecen manejar un rango de valores particular que los distancia del resto.



Para el tratamiento manual de valores atípicos se usa un enfoque manual de tratarlos, mediante la simple eliminación de los mismos.Esto nos deja con distribuciones de variables más uniformes:

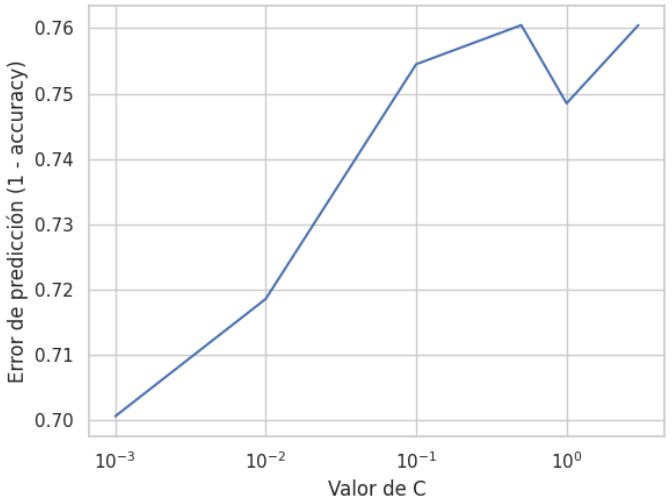


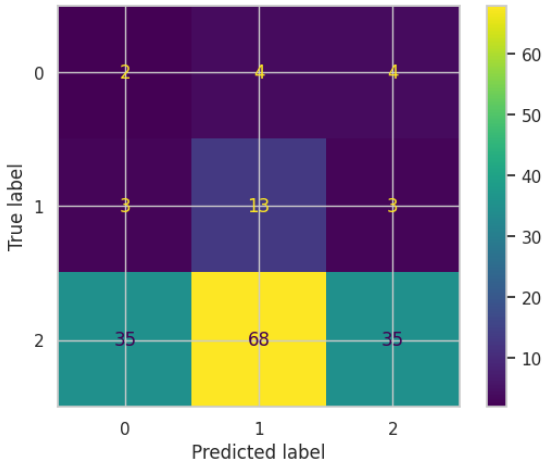
Pero las métricas del modelo base disminuyen fuertemente, por lo que resolveremos este enfoque de tratamiento de valores atípicos mediante parametrización de los modelos, en lugar de tratarlos directamente los datos.

Probamos el primer modelo de clasificación como algoritmo de support vector machine con un kernel lineal, donde le indicamos mediante parámetro que automáticamente assign pesos a cada clases para dar balance a las mismas, sin necesidad de alterar el dataset como tal, si no la forma de interpretar las clases del modelo.

Probamos manualmente y también mediante hiper parametrización con Grid SearCH el ajuste del parámetro de costo del algoritmo de clasificación para minimizar el impacto de los valores atípicos, y obtenemos una gráfica de evolución del error con respecto a este hiper parametro..

El mejor valor para C suelen ser valores muy bajos. En este caso nos quedamos con el valor 0.001 para C.

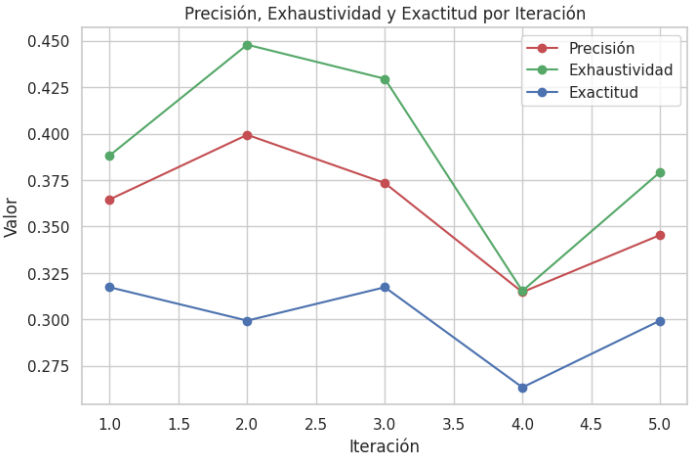


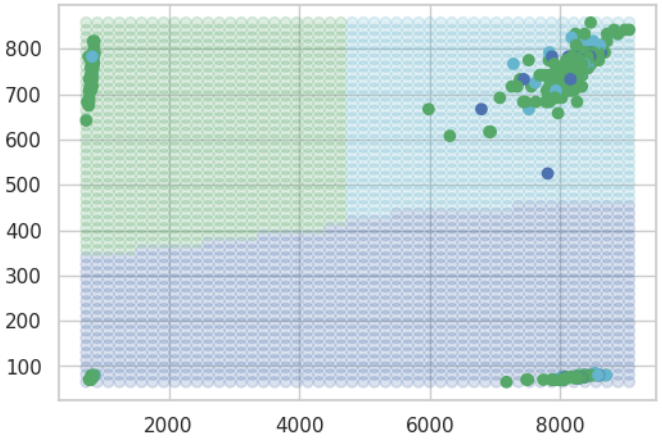


El modelo con kernel lineal balanceado con costo 0.001 logra una matriz de confusión como la de la izquierda. Notamos que comete muchos errores de falsos positivos (marca 68 veces que debe ser 1, cuando en realidad es 2, por ejemplo).

Sus métricas en general tampoco son buenas. Un gráfico de las métricas para cada set en el proceso de validación cruzada demuestra que para ciertas combinaciones de los datos a veces se logran resultados mejores, que otras iteraciones.

	precision	recall	f1-score	support
Blue-Green	0.05	0.20	0.08	10
Bluish-Green	0.15	0.68	0.25	19
Green	0.83	0.25	0.39	138
accuracy			0.30	167
macro avg	0.35	0.38	0.24	167
weighted avg	0.71	0.30	0.35	167





El gráfico de la izquierda puede ser de utilidad para ver como el modelo de clasificación divide en el espacio las clases.

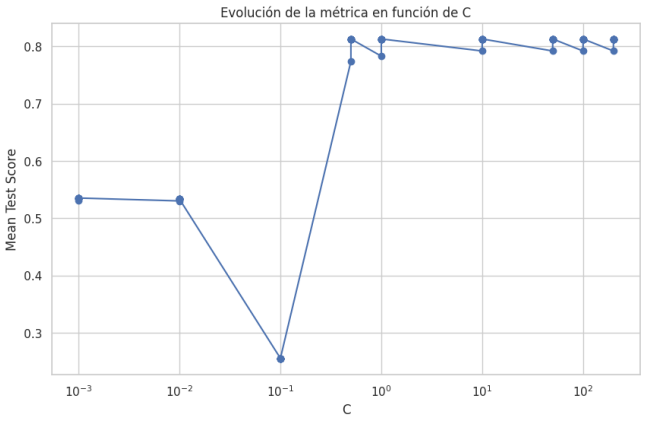
Los datos son reales de train solo para dos variables para facilidad de visualización, y los colores del fondo indican la clase a la que pertenece ese espacio.

Notamos que no realiza un buen trabajo este algoritmo para la clasificación.

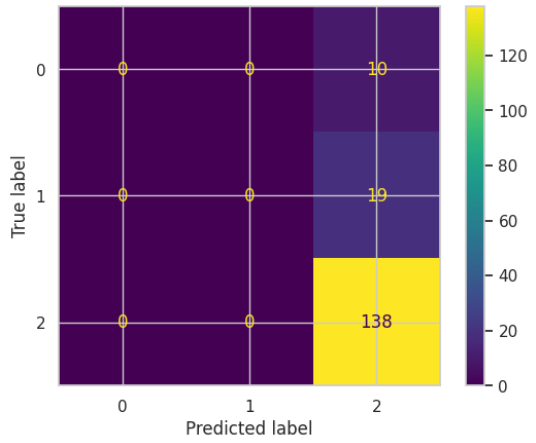
Por lo que ahora se hace un prueba con el mismo algoritmo pero utilizando un kernel gaussiana. Esta vez probando la hiper parametrización automática de los parámetros C y gamma para el modelo.

Un gráfico de la evolución de C puede ser útil para comprender el mejor valor.

Particularmente para este algoritmo la configuración óptima de de C en 0.5 y gamma 0.1. Esto obtenido mediante Grid Search CV,

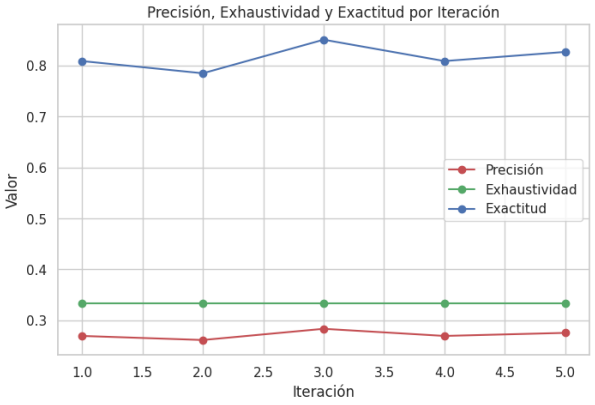


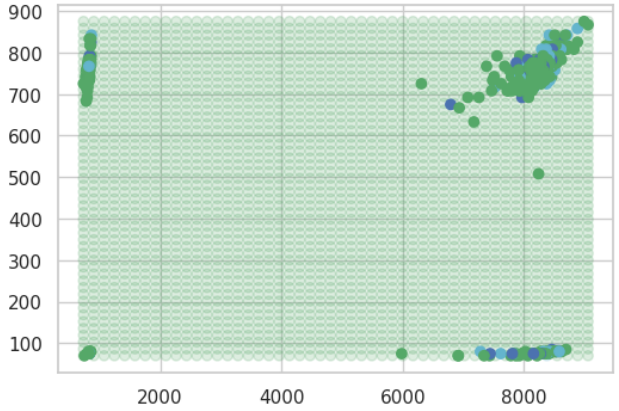
Si bien las métricas han mejorado debido a que el modelo se equivoca menos, ahora tenemos un sesgo debido a que clasifica correctamente solo la clase mayoritaria a pesar de haber configurado el parámetro de los pesos como balanceado.



	precision	recall	f1-score	support
Blue-Green	0.00	0.00	0.00	10
Bluish-Green	0.00	0.00	0.00	19
Green	0.83	1.00	0.90	138
accuracy			0.83	167
macro avg	0.28	0.33	0.30	167
weighted avg	0.68	0.83	0.75	167

En la validación cruzada podemos notar la evolución de las métricas en cada iteración de los conjuntos de train y test de la siguiente forma (izquierda). Y también visualizar cómo trabaja la clasificación utilizando solo dos variables explicativas

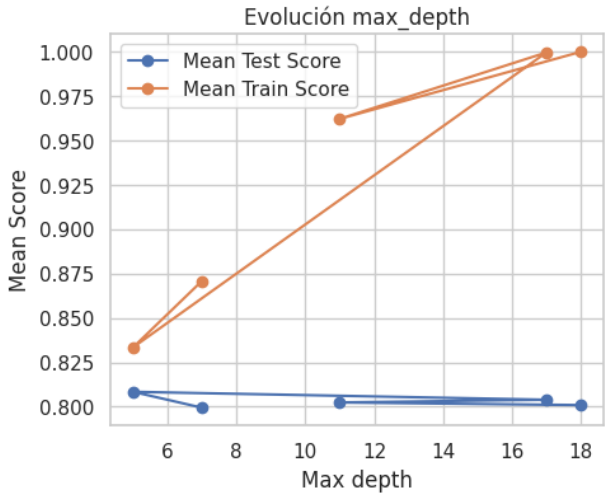


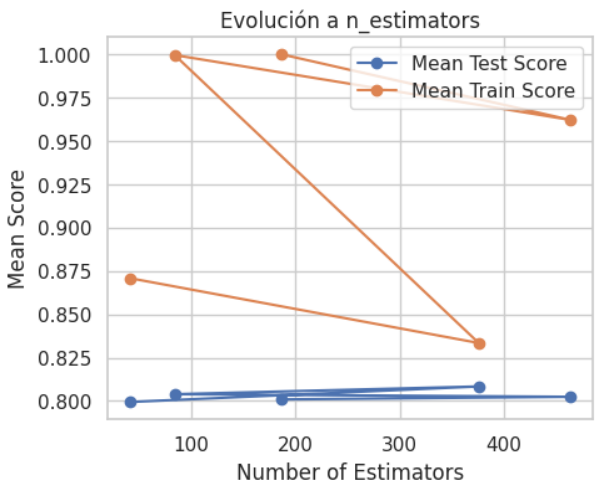


No parece clasificar en lo absoluto. De hecho se corresponde a los que indicaba la métrica. Predice mayor o totalmente en la clase Green y no en las demás. O a lo mejor no podemos trabajar debido a una limitación de la dimensionalidad.

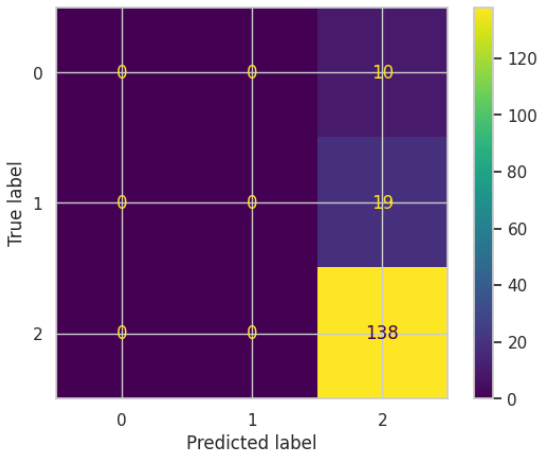
Probamos un nuevo algoritmo de clasificación basado en árboles que no se ve afectado por la presencia de valores atípicos ni desbalanceos, por lo que no es necesario parametrizar ni transformar los datos en este caso.

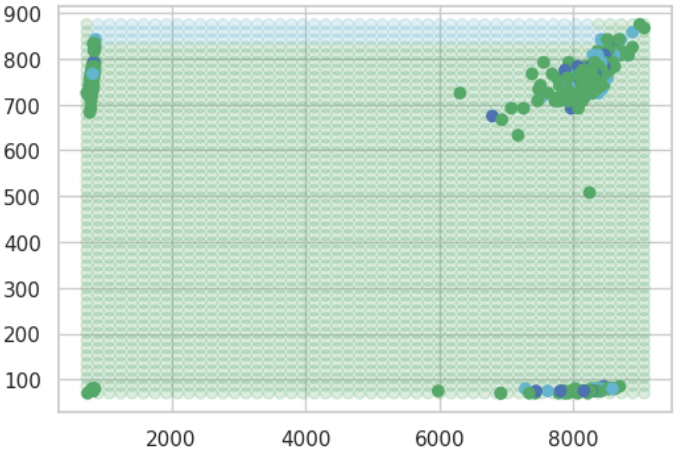
Se hace una selección automática de los mejores hiper paramétricos mediante una técnica de Randomized Search CV. Obtenemos las gráficas de la evolución de las métricas de máxima profundidad y número de estimadores:





El método de búsqueda establece que los mejores parámetros refieren al uso de 376 estimadores y una profundidad máxima de 5 niveles. Notamos un rendimiento prácticamente igual al método de Support Vector machine del tercer ejercicio, usando un kernel gaussiano y balanceando las clases. Presentando nuevamente un modelo preciso, pero sesgado a la clase mayoritaria.





Nuevamente visualizamos solo con dos dimensiones de los datos como trabaja la clasificación este algoritmo de bosques de decisión, y notamos cierta diferencia con el método anterior.En este caso parece detectar una pequeña región de la clase azul, pero no representa efectivamente las clases, al menos para estas variables.

Conclusiones

Los diferentes métodos de clasificación tanto los basados en máquinas de soporte vectorial como los basados en árboles de decisión han demostrado un rendimiento no tan excelente para el conjunto de datos de entrada, debido principalmente al gran desbalance que poseen las clases.

Pero en general, pudimos notar un mejor rendimiento en el tercer (Support Vector Machine Gaussiano) y cuarto modelo (RandomForest Classifier) en que han acertado su mayoría las predicciones debido a que predecían generalmente la clase mayoritaria. Por lo que resultaban modelo sesgados. Al modelo de clasificación de Support vector machines con kernel lineal no le ocurría este sesgo debido a su balance en los pesos del modelo, pero sí demostró equivocarse mucho más en sus estimaciones, teniendo a clasificaciones que resultan incorrectas.

Por lo que si tuviéramos que escoger un modelo de los tres propuestos para esta tarea específica con estos datos, sería cualquiera de los últimos dos debido a que su rendimiento es prácticamente el mismo en ambos.