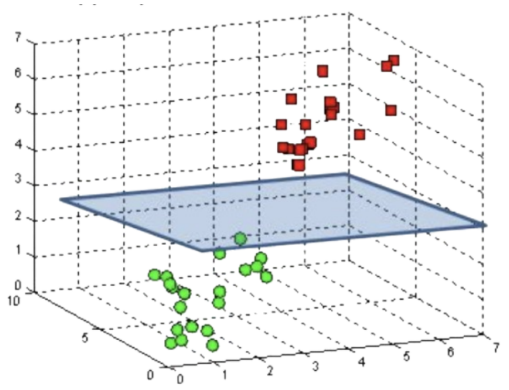
Universidad Nacional de Rosario

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*:** 3/11/2023

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

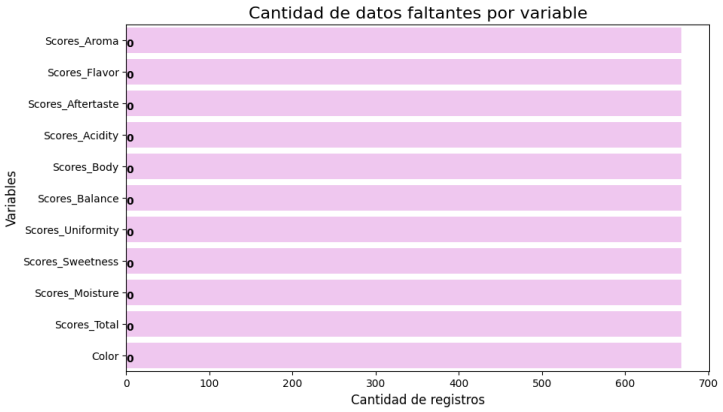
El dataset cuenta inicialmente con un total de 835 filas y 11 columnas. Cada columna representa una variable numérica discreta que representan puntajes de cada característica de un tipo de cadena particular. Pero también posee una variable categórica de tres clases, ‘Color’, que representa el color del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

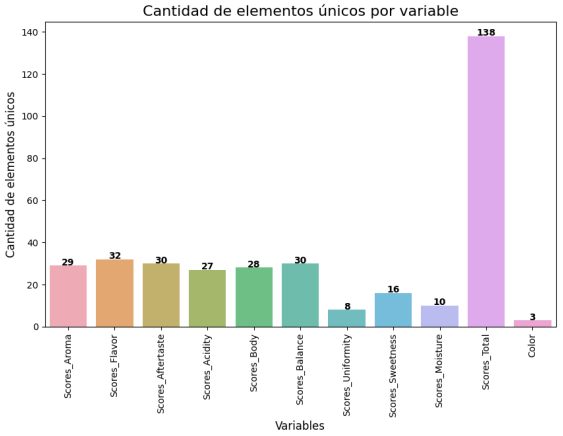
Planteamos la división inicial del conjunto de datos, con el fin de no perturbar el conjunto de validación que deberíamos tratar como si no lo conociéramos de antemano, para que el modelo de clasificación no se contamine con datos que todavía no conoce. Por ello, se dividen las variables explicativas 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):

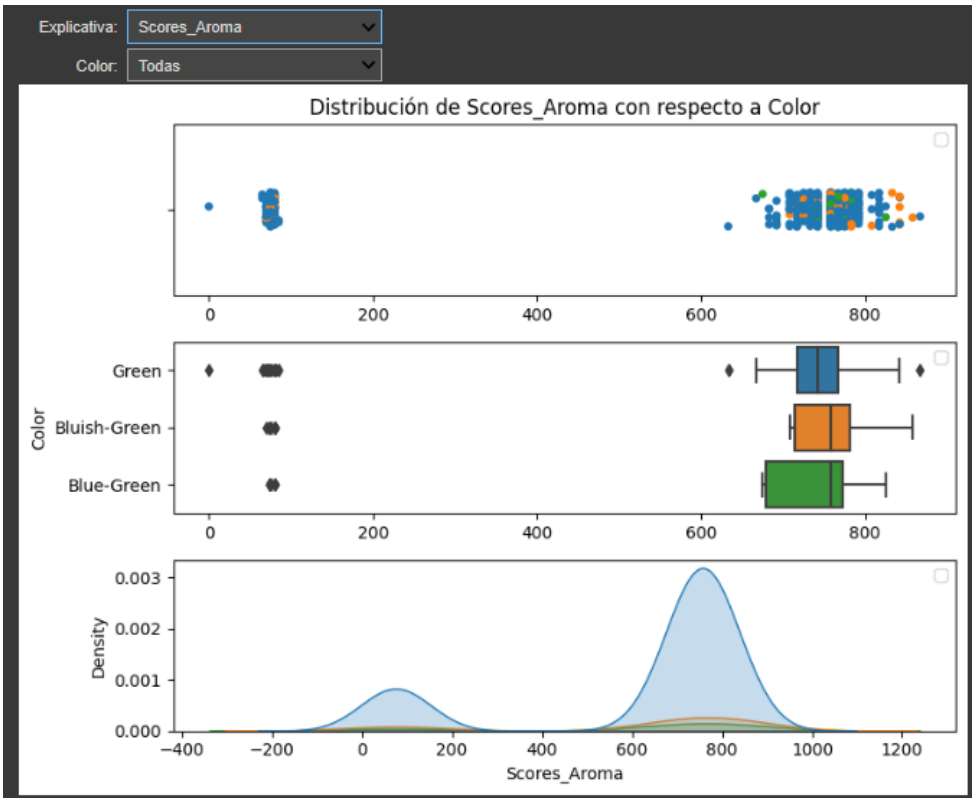
	Scores_Aroma	Scores_Flavor	Scores_Aftertaste	Scores_Acidity	Scores_Body	Scores_Balance	Scores_Uniformity	Scores_Sweetness	Scores_Moisture	Scores_Total	Color
478	758	775	75	758	758	758	100	100	11	8317	Green
346	742	733	75	717	708	833	100	100	1	8233	Green
462	767	767	767	792	75	767	100	100	0	8375	Green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



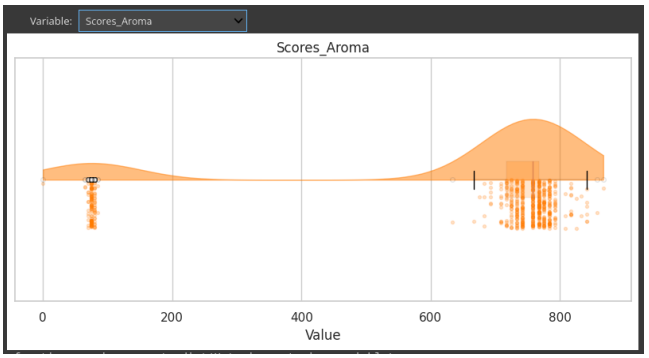


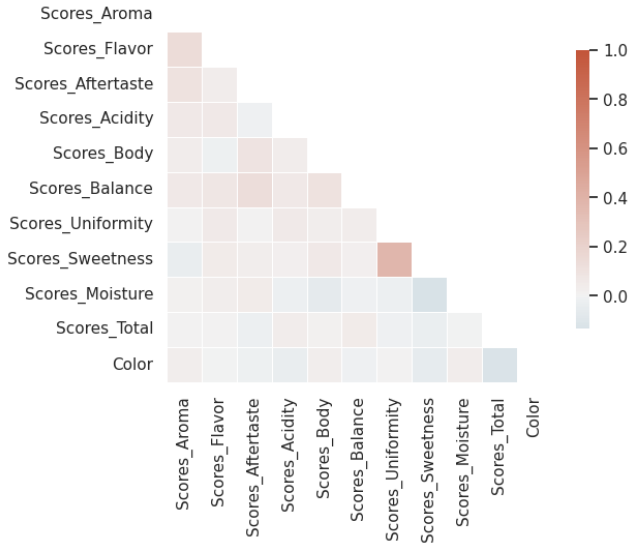
Como parte del análisis descriptivo, podemos comprender la distribución de las variables y de cada clase con un gráfico interactivo que permite la elegir la variable de interés a lo largo del eje x, y la o las clases que se desean visualizar:



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 Support Vector Machines con sus parámetros por default. De esta manera cada conjunto que se le pasa por parámetro obtiene métricas de validación para clasificación.

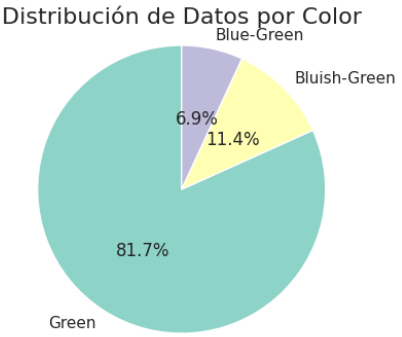
La imagen de la derecha muestra métricas base para el dataset sin procesamiento.

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.

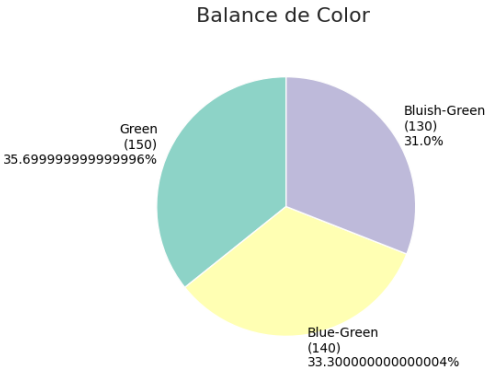
```
base_model = svm.SVC()
train_and_evaluate(base_model, X_train, X_test, y_train, y_test)
```

	precision	recall	f1-score	support
Blue-Green	0.00	0.00	0.00	12
Bluish-Green	0.00	0.00	0.00	20
Green	0.81	1.00	0.89	135
accuracy			0.81	167
macro avg	0.27	0.33	0.30	167
weighted avg	0.65	0.81	0.72	167

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 mezcla de técnicas de undersampling y oversampling para balancear el dataset.



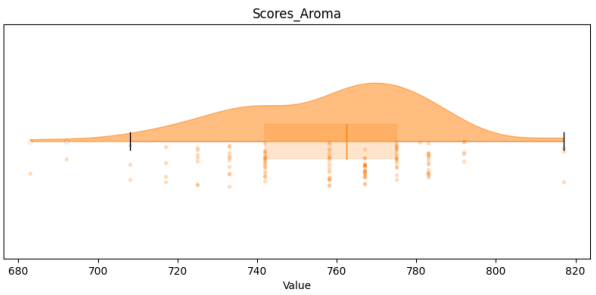

```
model_over = svm.SVC()
train_and_evaluate(model_over, X_train_over, X_test_out, y_train_ov
```

	precision	recall	f1-score	support
Blue-Green	0.09	1.00	0.17	3
Bluish-Green	0.00	0.00	0.00	6
Green	0.00	0.00	0.00	24
accuracy			0.09	33
macro avg	0.03	0.33	0.06	33
weighted avg	0.01	0.09	0.02	33

Pero en esta instancia, el balance no es favorable debido a que disminuye las métricas del modelo base. Tanto submuestrear como sobremuestrear las clases por separado y en conjunto resultan en impactos altamente negativos al modelo base de clasificación.

Por lo que se evita continuar este enfoque de tratar el desbalance, para tratar este tema directamente desde los parámetros del modelo, asignándole pesos tales a cada clase que estabilice la importancia de las mismas a la hora de clasificar.

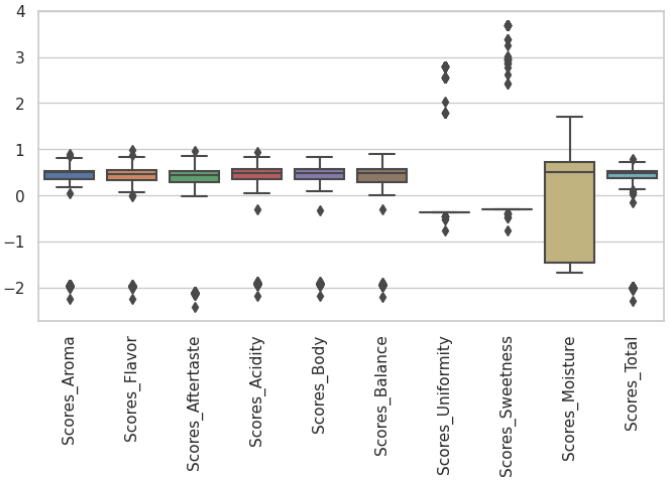
También se propone reducir la influencia de los valores atípicos debido a que perturba la distribución de las variables. Logrando distribuciones más asimétricas, pero esto impacta negativamente al modelo, por lo que se decide no continuar este enfoque, y tratar los valores atípicos directamente desde el parámetro costo del modelo.




```
model_out = svm.SVC(probability=True)
train_and_evaluate(model_out, X_train_out, X_test_out, y_train_out, y_test_out)
```

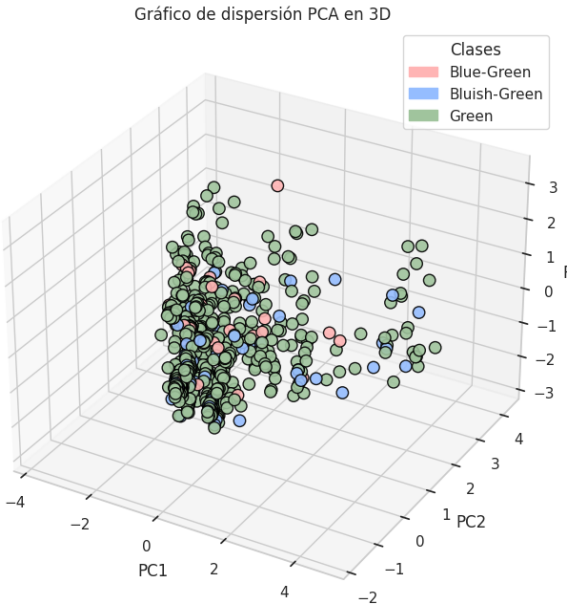
	precision	recall	f1-score	support
Blue-Green	0.00	0.00	0.00	3
Bluish-Green	0.00	0.00	0.00	6
Green	0.73	1.00	0.84	24
accuracy			0.73	33
macro avg	0.24	0.33	0.28	33
weighted avg	0.53	0.73	0.61	33

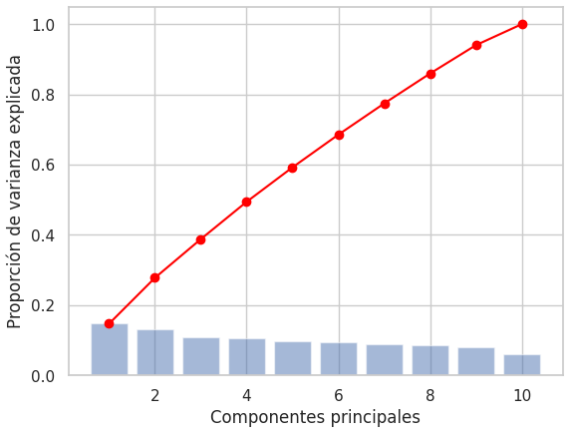
Luego se propone estandarizar los datos para llevar a todas las variables a un rango de valores facil de comprar:



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 la presencia de valores atípicos pertenecientes a las tres clases. Además podemos establecer que difícilmente un método que trace hiperplanos o curvas para lograr la separación de las clases podrá trabajar de forma correcta debido a la superposición de los puntos de las diferentes clases.

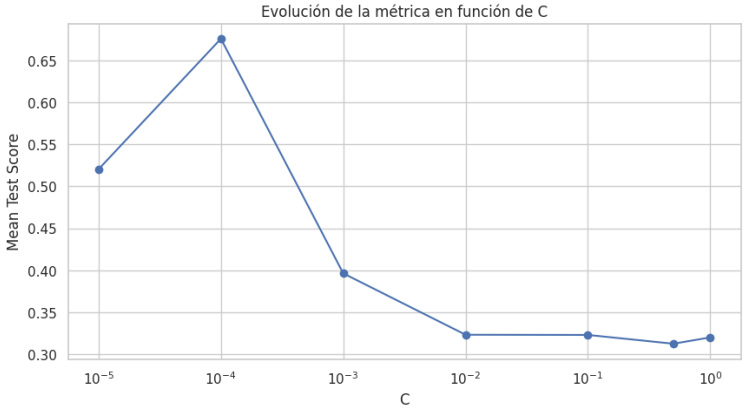




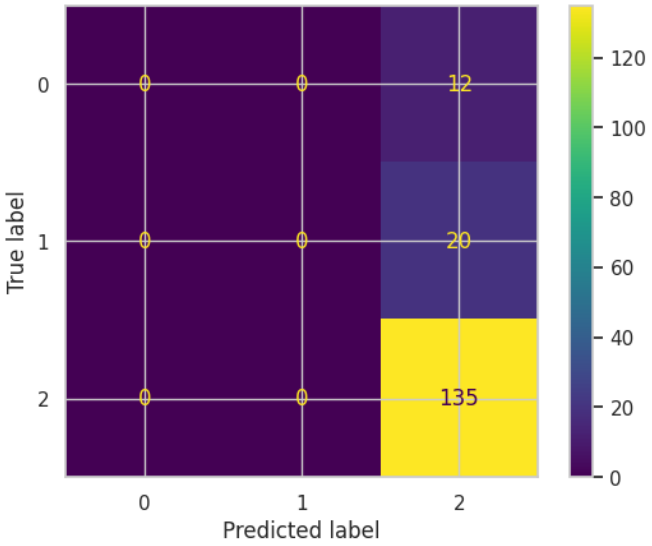
Además un gráfico de explicabilidad de componentes insinúa que no hay atributos ni direcciones que destaques por su explicabilidad de los datos, sino que todos los componentes explican de forma proporcional al conjunto.

Probamos el primer modelo de clasificación como algoritmo de support vector machine con un kernel lineal, donde le indicamos mediante parámetro que automáticamente asigna pesos a cada clase 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 mediante hiper parametrización con GridSearhCV 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 parámetro..



El mejor valor para C suelen ser valores muy bajos. En este caso nos quedamos con el valor 0.0001 para C.

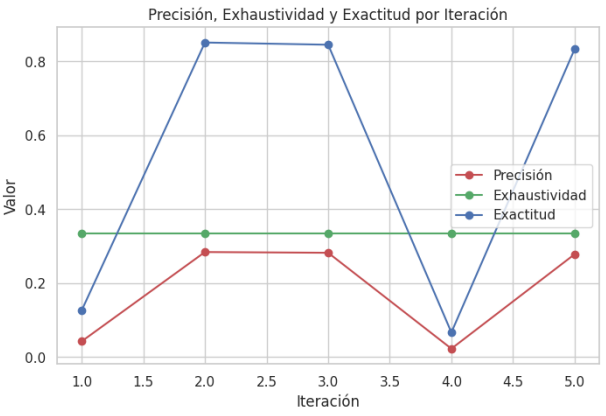


El modelo con kernel lineal balanceado con costo 0.0001 logra una matriz de confusión como la de la izquierda. Notamos que comete errores de falsos positivos (marca 20 veces que debe ser clase 2, cuando en realidad es 1, por ejemplo).

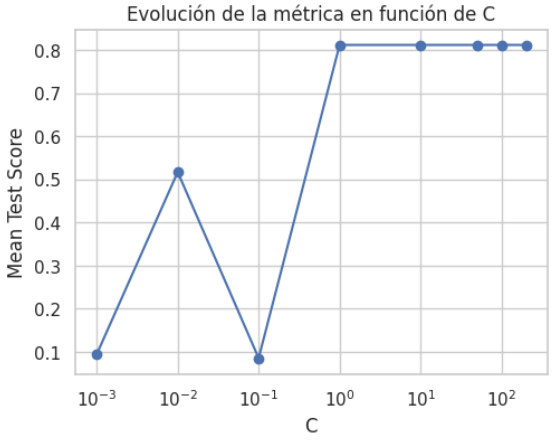
Sus métricas en general son buenas debido a que siempre predice la clase Green para un conjunto de datos que tiene muchos valores de clase Green.

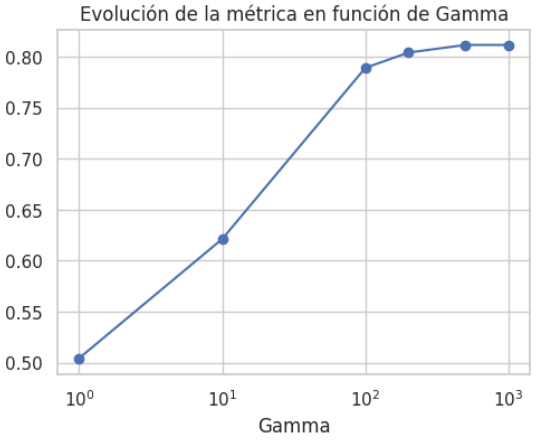
	precision	recall	f1-score	support
Blue-Green	0.05	0.20	0.08	10
Bluish-Green	0.15	0.68	0.25	19
Green	0.83	0.25	0.39	138
accuracy			0.30	167
macro avg	0.35	0.38	0.24	167
weighted avg	0.71	0.30	0.35	167

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.

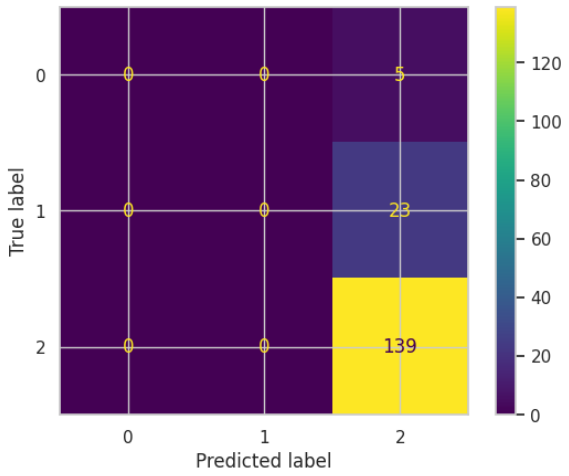


A continuación se intenta resolver el mismo problema con el mismo algoritmo pero utilizando un kernel radial. Esta vez probando la hiper parametrización automática de los parámetros C y gamma para el modelo. Un gráfico de la evolución de C y gamma puede ser útil para comprender el mejor valor para cada parámetro para optimizar los resultados.



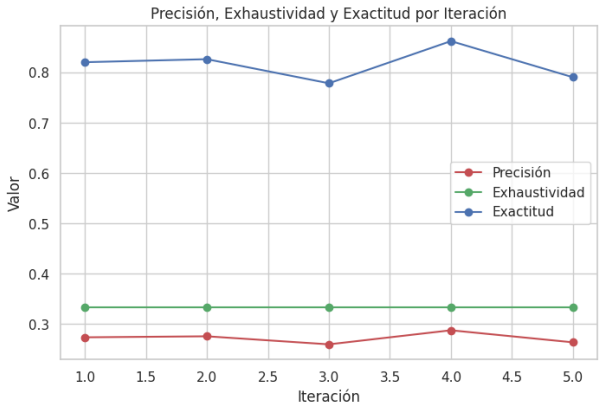


Particularmente para este algoritmo la configuración óptima de de C en 1 y gamma 500, debido a que para este punto, se estabilizan las métricas de validación a su punto máximo. Podemos observar su desempeño en las métricas y en la matriz de confusión resultante.



	precision	recall	f1-score	support
Blue-Green	0.00	0.00	0.00	5
Bluish-Green	0.00	0.00	0.00	23
Green	0.83	1.00	0.91	139
accuracy			0.83	167
macro avg	0.28	0.33	0.30	167
weighted avg	0.69	0.83	0.76	167

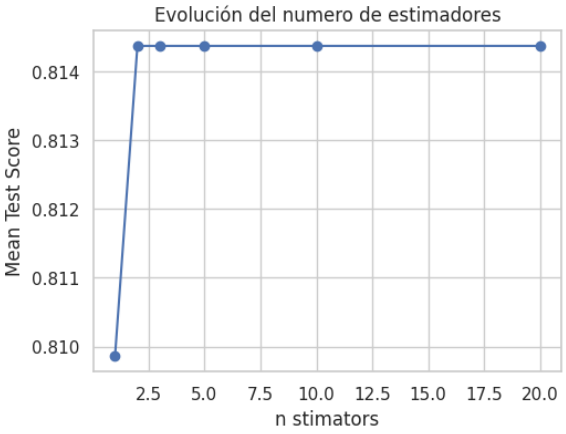
Si bien las métricas han mejorado debido a que el modelo se equivoca menos, ahora tenemos un sesgo debido a que clasifica correctamente solo la clase mayoritaria a pesar de haber configurado el parámetro de los pesos como balanceados.

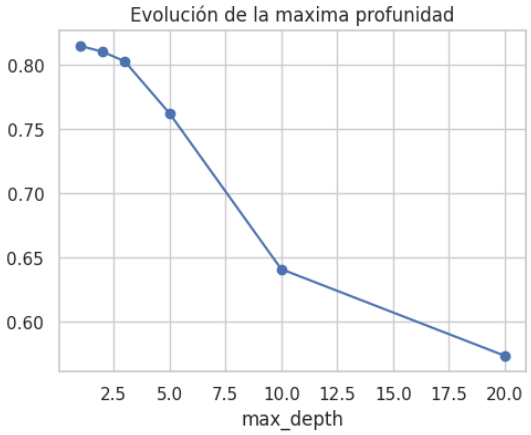


Podemos notar un proceso de entrenamiento relativamente estable para las diferentes combinaciones de conjuntos de entrenamiento y validación, lo que resulta un ventaja del modelo anterior.

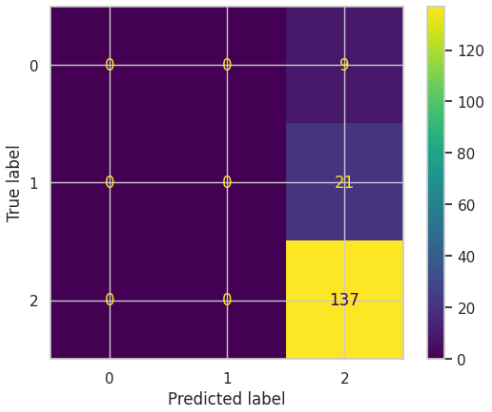
Probamos un nuevo algoritmo de clasificación basado en árboles que no se ve afectado por la presencia de valores atípicos ni desbalanceos, por lo que no es necesario parametrizar ni transformar los datos en este caso. El método de bosques aleatorios ensambla distintos árboles de decisión para aumentar la robustez del entrenamiento.

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





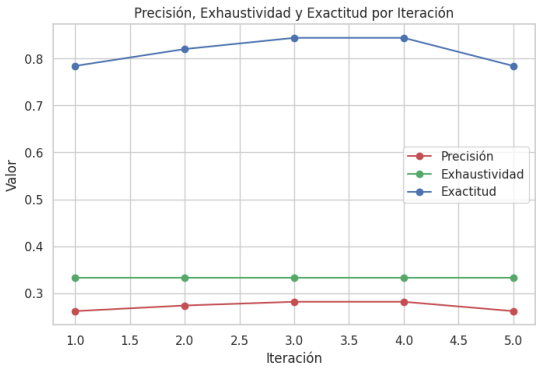
El método de búsqueda establece que los mejores parámetros refieren al uso de 2 estimadores y una profundidad máxima de 1 nivel. Ya que luego de esto las métricas se estabilizan o caen en errores más grandes. Esto significa que para este problema el algoritmo es más efectivo cuando toma decisiones cortas y rápidas, en lugar de expandir árboles en profundidad para determinar a qué clase pertenece un valor.



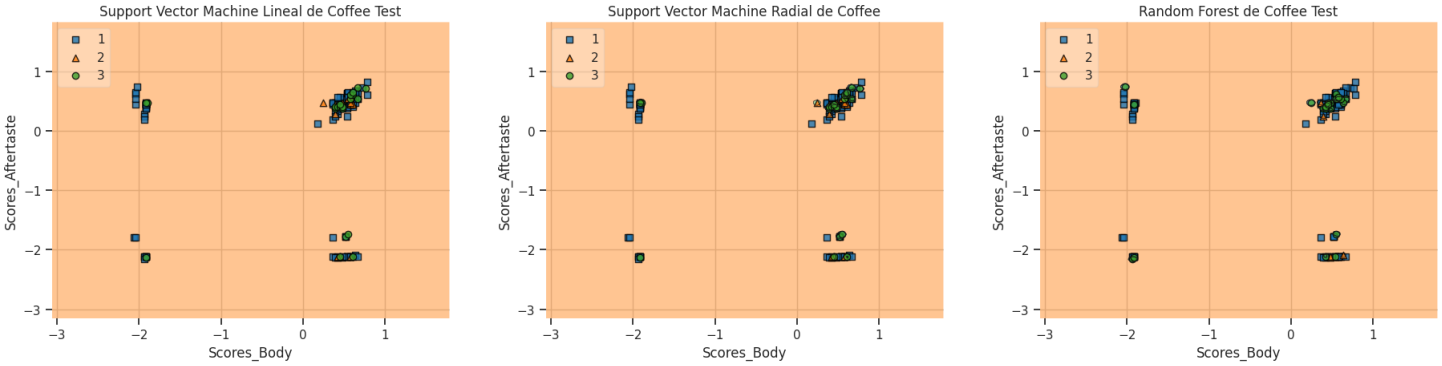
Nuevamente notamos que tiende a clasificar correctamente la clase de los registros que tiene clase mayoritaria, y que de hecho sólo preside para todos los valores esta misma clase ya que no puede distinguir las clases las unas de las otras.

Notamos que su entrenamiento tiene un comportamiento estable para las distintas combinaciones de conjuntos de datos.

	precision	recall	f1-score	support
Blue-Green	0.00	0.00	0.00	9
Bluish-Green	0.00	0.00	0.00	21
Green	0.82	1.00	0.90	137
accuracy			0.82	167
macro avg	0.27	0.33	0.30	167
weighted avg	0.67	0.82	0.74	167



Por último, notemos una comparación de cómo clasifican los tres modelos el conjunto de validación en un gráfico donde los colores del fondo representan la clase que predice para los puntos en esa posición, y los puntos están pintados por los colores de sus clases originales.



Notamos que los tres modelos tienen el mismo modelo: estiman todos los valores como si fueran la misma clase. Notar que esto solo está adaptado a dos de las variables explicativas. Es posible que la limitación de visualización no permita identificar las separaciones de las clases.

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 segundo modelo (Support Vector Machine Gaussiano) en que ha acertado su mayoría las predicciones de verdaderos positivos a pesar de sus predicciones de falsos positivos. Por lo que resulta en un modelo sesgado.

Si tuviéramos que escoger un modelo de los tres propuestos para esta tarea específica con estos datos, sería este modelo debido a que su rendimiento es levemente superior a los otros dos enfoques.