Universidad de Buenos Aires

Maestría en explotación de datos y descubrimiento del conocimiento

Aprendizaje Automático - 1° Cuatrimestre 2021

Trabajo Práctico N°1

Predicción de Accidente cerebro vascular (ACV)

Integrantes:

- Adrian Marino
- Alejandro Szpak
- Claudio Collado

https://github.com/magistery-tps/aa-tp1
https://github.com/magistery-tps/aa-tp1/blob/main/notebooks/tp1.ipynb

1. Resumen

El Accidente Cerebrovascular (ACV) es considerada la epidemia del siglo 21: Este trastorno, caracterizado por la reducción crítica de la llegada de sangre al cerebro, representa la segunda causa de muerte en nuestro país y la primera de discapacidad tanto a nivel local como en el resto del mundo. Una gran proporción de estos son evitables identificando y mitigando en forma temprana los factores de riesgos asociados a su ocurrencia. La aplicación de técnicas de aprendizaje automático en medicina tiene un enorme impacto, ayudando a identificar en forma temprana la susceptibilidad y ocurrencia de este tipo de enfermedades, entre otras.
En línea con lo anterior este trabajo presenta una serie de análisis exploratorios, identificación y selección de atributos relevantes, preprocesamiento y ajuste (tuning) de modelos de árboles de decisión de forma tal de verificar su pertinencia y aplicación en la predicción de ACV.

2. Introducción

El presente trabajo práctico tiene como objetivo principal la aplicación de conceptos y metodologías vistos en la primera parte de la materia *Aprendizaje Automático* perteneciente a la *Maestría en explotación de datos y descubrimiento de conocimiento.*
Para su elaboración se parte de un conjunto de datos suministrados sobre los cuales se aplican técnicas de análisis exploratorio, preprocesamiento y el uso de algoritmos para la generación de árboles de decisión de forma tal de obtener una herramienta de predicción de la ocurrencia de un accidente cerebrovascular en pacientes. A continuación se describe brevemente el contenido de las próximas secciones:

- En la **Sección 3** se describen las principales características y particularidades del conjunto de datos suministrados.
- En la **Sección 4** se describen las metodologías utilizadas sobre los datos y algoritmos.
- En la **Sección 5** se exponen los resultados obtenidos y el análisis correspondiente.
- En la **Sección 6** se enumeran las conclusiones.
- En la **Sección 7** se enumera la bibliografía general y lecturas utilizadas.
- En la **Sección 8** se presentan los Anexos complementarios al cuerpo principal.

3. Datos

A continuación se describen la principales características y particularidades del dataset provisto por la cátedra (*healthcare-dataset-stroke-data.csv*):

a) Se encuentra compuesto por 12 columnas y 5110 instancias (filas). Las columnas presentan las siguientes características:
   - **id**: Identificador único: No utilizado
   - **gender**: Género - Categórica nominal. Presenta un moderado desbalance de clases siendo la clase mayoritaria el género femenino.
   - **age**: Edad - Numérico. Los valores se encuentran en un amplio rango desde escasos meses hasta 82 años. La distribución presenta cola hacia la izquierda.
   - **hypertension**: Presencia o no de hipertensión - Categórica nominal. Presenta un gran desbalance de clases hacia aquellos que no presentan esta patología.
   - **heart_disease**: Presencia o no de enfermedad cardiaca - Categórica nominal. Presenta un gran desbalance de clases siendo la clase mayoritaria aquellos que no presentan esta patología.
   - **ever_married**: Condición de casado - Categórica nominal. Presenta un moderado desbalance de clases siendo la clase mayoritaria los casados.
   - **work_type**: Tipo de trabajo - Categórica nominal. Presenta 5 clases, siendo la clase mayoritaria la correspondiente al tipo de trabajo privado.
   - **Residence_type**: Tipo de residencia - Categórica nominal. Presenta las clases balanceadas.
   - **avg_glucose_level**: Nivel de glucosa medio - Numérico. Presenta una cola hacia la derecha, observándose el grupo con valores normales y aquellos con valores superiores correspondiente a personas con diabetes.
   - **bmi**: Índice de masa corporal - Numérico. Presenta valores extremos cercanos a 100.
   - **smoking_status**: Nivel de fumador - Categórica nominal. Presenta 4 clases, siendo la clase mayoritaria la correspondiente a los que nunca fumaron.
   - **stroke**: Corresponde a la ocurrencia o no de un ACV - Categórica nominal. Esta columna corresponde a las etiquetas (labels) para el entrenamiento del modelo y al target que se pretende predecir. Presenta un gran desbalance de clases, donde la clase mayoritaria corresponde a la no ocurrencia de un ACV con el 95.13% de las etiquetas.

   En la Sección A del Notebook realizado para este Trabajo Práctico se observa el conteo de clases y las distribuciones correspondientes a lo detallado anteriormente.

b) La columna **bmi** es la única que contiene valores faltantes (NaN): En total son 201 valores que corresponde al 3.93% del total de esta columna.

4. Metodología

A lo largo del desarrollo del trabajo práctico se aplicaron en diferentes etapas y orden de ejecución las siguientes metodologías:

a) Análisis de correlaciones:
   - Atributos numéricos contra el target: Se realizó por medio de **Point Biserial Correlation**. Para variables categóricas dicotómicas, como es el caso del target, esta metodología es equivalente al resultado obtenido por medio de la correlación de **Pearson.**
   - Atributos categóricos contra el target. Se realizó por medio del coeficiente de **Cramér's**

   En el Anexo 8.1 se observan las matrices de correlaciones y sus correspondientes mapas de calor obtenidas para estos métodos.

b) División del conjunto de datos Inicial: Sobre el conjunto de datos inicial y completo se realizó por única vez la división estratificada con respecto al target de la siguiente forma:
   - Dataset de desarrollo (development): 80% para utilizar en el desarrollo, prueba de algoritmos y búsqueda de hiperparametros.
   - Dataset de evaluación (test): 20% para utilizar al final del proceso de desarrollo para el test de performance de los modelos seleccionados.

   En aquellos casos donde fue necesario también se generó la división del conjunto de datos de desarrollo (development) de la siguiente forma:
   - Dataset de entrenamiento: 80% para utilizar en el entrenamiento de los modelos
   - Dataset de validación: 20% para utilizar en la validación de los diferentes modelos entrenados

c) Imputación valores faltantes (NaN): Los valores faltantes en la columna **bmi** fueron imputados por medio de **IterativeImputer** donde los valores a imputar se modelan como una función de los demás atributos. Se verificó en forma gráfica la similitud de las distribuciones pre/post imputación.

d) Balanceo de clases (over/under sampling): Se utilizó para compensar el gran desbalance de clases existentes en el target identificado anteriormente

e) Codificación de atributos categóricos: Para las variables de tipo categóricas nominales se realizó la codificación correspondiente por medio de **One Hot Encoding**.

f) Identificación de importancia de atributos (feature importance): La importancia de los atributos fueron obtenidos en forma directa como resultado de los árboles utilizados. Al final de trabajo se utilizó **Recursive Feature Elimination (RFE)** para la selección recursiva de atributos

g) Utilización de diferentes semillas (seeds) y **k-fold cross validation** asi como tambien **RandomizedSearchCV** en la búsqueda de hiperparametros.

5. Resultados

5.1 Variables predictoras - Importancia de los atributos

En principio el nivel de importancia de las variables predictores se consideró como el resultado obtenido del análisis de la correlación de Pearson de los atributos numéricos y del coeficiente de Cramer's para los atributos categóricos, ambos con respecto al target. Con esto como referencia se decidió continuar el desarrollo considerando todos los atributos, salvo el id que fue descartado en forma temprana por no considerarse útil. En particular en el ítem 5.3.5 se realiza un análisis específico de importancia de atributos y su impacto en las métricas por medio de la técnica Recursive Feature Elimination (RFE).

5.2 Métrica de Performance

Según lo analizado se consideró importante poder tener una medición cuantitativa del costo de error (Falsos Negativos y Falsos Positivos), para de esta forma tener una medición exacta de la performance de los modelos en análisis. Al no contar con esto consideramos que ambos tipos de Falsos son importantes, pero ligeramente vemos una importancia más elevada a reducir falsos negativos, ya que no advertir un stroke en un paciente tiene potencialmente un riesgo mayor que diagnosticar erróneamente un posible caso futuro de stroke (a menos que requiera diagnosticar un medicamento, en ese caso podría ser equivalente el costo de ambos tipos de errores). Teniendo esto en consideración decidimos seleccionar F1-Score como métrica de performance, dado que esta nos permite tener una media promedio (media armónica) entre Precision y Recall.

5.3 Entrenamiento y evaluación de los Modelos

5.3.1 50 Random Seeds (semillas aleatorias)

El primer método de entrenamiento utilizado fue la iteración de 50 Random Seeds (semillas aleatorias): En el Anexo 8.2 se observan los BoxPlots correspondientes a diferentes métricas, focalizando sobre F1 Score ya que fue la métrica seleccionada anteriormente.
En primer lugar vemos un valor bastante bajo de F1 Score (entre 0.15 y 0.25, aproximadamente). Al expandir el análisis hacia las demás métricas, especialmente Precisión y Recall, podemos ver que precisión es la que afecta negativamente al número, con valores entre 0.10 y 0.20 dependiendo la iteración. Al contrario, el Recall da valores altos (entre 0.50 y 0.80). Esto interpretamos se debe a un umbral de predicción alto, lo cual lleva a predecir una mayor cantidad de clases como positivas (1), lo cual eleva el Recall pero disminuye la precisión, ya que impacta directamente en los Falsos Positivos (Recordemos Recall = TP / TP + FN y Precisión = TP / TP + FP).
A partir de esta observación graficamos la curva ROC (Receiver operating characteristic), para observar gráficamente el trade-off entre TPR y FPR a medida que se modifica el umbral de decisión (ver ítem 5.3.3.1).

5.3.2 50-Fold CV (Cross Validation)

Luego de analizar la performance entre los splits de train y validación usando 50 Random Seeds, realizamos el análisis con 50 Fold CV. Inicialmente tuvimos inconvenientes interpretando los resultados, dado que al plotear el BoxPlot con el score de F1 Score para todos los splits, veíamos resultados muy altos, entre 0.87 y 0.99 (ver Anexo 8.3). Sin embargo, al hacer predicciones utilizando el dataset de validación con el modelo resultante vemos nuevamente una caída fuerte de la performance. Dicho esto, entendemos que aunque utilicemos cross-validation para entrenar, la performance del modelo al predecir con datos que no fueron usados en entrenamiento no es buena (overfitting).
En la Sección H del notebook realizado para este trabajo práctico se observa el esquema del árbol resultante y la importancia de los atributos de esta etapa: Se observa un gran tamaño del árbol tanto en apertura lateral como en profundidad. Con respecto a los atributos que más importancia le da estos corresponden a age, avg_glocuse_level y bmi.

5.3.3 10-Fold CV + Poda

Se continuó con el análisis del modelo considerando la poda del mismo y como esta operación afectaba el tamaño del árbol y su performance. A continuación se describen los análisis realizados y conclusiones obtenidas:

1) **Influencia del valor de alfa (α) en las métrica de performance:**
   Se analizó el comportamiento de las métricas de entrenamiento y validación al variar el valor del parámetro alfa (α) correspondiente a la poda:
   - Para el dataset de entrenamiento el valor de la métrica decrece al aumentar el valor de α.
   - Para el dataset de validación al principio la métrica aumenta logrando superar el valor de 0.2 para luego decrecer y estabilizarse en valores cercanos a 0.2 al aumentar el valor de α.

2) **Influencia del valor de alfa (α) en la profundidad del árbol:**
   - El aumento del valor de α impacta directamente en la profundidad del árbol: Partiendo de profundidades superiores a 20 pequeños aumentos en el valor de α generan una gran reducción en la profundidad, observándose que en valores cercanos a α = 0.02 el árbol llega a tamaño mínimo y un aumento en el valor de α ya no genera efecto alguno.

En el Anexo 8.4 se observa gráficamente estos comportamientos.

5.3.3.1 Curva ROC

Motivados por la observación de la diferencia entre Precisión y Recall y cómo esto impacta a la métrica seleccionada (F1 Score), decidimos graficar curva ROC utilizando el modelo entrenado con 10 Fold CV + Poda (ver Anexo 8.5). Adicionalmente calculamos los pares TPR-FPR con un bucle For, calculando y observando el comportamiento de las métricas a medida que aumenta/disminuye el umbral de decisión. Al ver esto pudimos ver visualmente un “punto óptimo” cercano a los 0.50, punto donde pareciera que se optimiza el ratio entre precisión y recall. Al ser el umbral (threshold) por defecto 0.50 en Sklearn, entendimos que modificando este parámetro, no generarían mejores resultados.

Otro aspecto importante fue ver cómo los cambios entre los distintos valores de la matriz de confusión se dan de a "saltos" y esto se debe a la cantidad baja de valores de clase positiva en el dataset de validación, debido al desbalanceo (alrededor de 40 observaciones de clase = 1). Viendo esto, podríamos pensar que con más datos, podría realizarse un análisis más robusto del umbral óptimo y si existe alguna manera de mejorar la performance del modelo para, de forma iterativa, poder chequear si las probabilidades predichas se "separan mejor" entre ambos valores de clase (0 y 1). Esto entendiendo que siempre el óptimo es que el modelo prediga con probabilidad alta los clase = 1 y con probabilidad baja a la clase = 0, y con esto y un buen uso del umbral de predicción, se podría mejorar la performance considerablemente.
Otro aspecto que vemos es la alta cantidad de Falsos Positivos en relación con la cantidad de Verdaderos Positivos (para TH 0.50 vemos 21 TP vs 131 FP en el dataset de validación), esta es otra visualización más de que nuestro modelo no es bueno, ya que está otorgando alta probabilidad de clase = 1 a los elementos con clase = 0, lo cual genera que tengamos malas predicciones.

5.3.4 Métricas en el conjunto de Evaluación

A continuación se observan los resultados de evaluar los modelos resultantes de 50 Random Seeds, 50 Folds Cv y 10 Fold CV + Poda para el conjunto de evaluación:

	50 Random Seeds	50 Folds Cv	10 Fold CV + Poda
val_precision	12.59%	11.50%	16.77%
val_recall	68.00%	26.00%	54.00%
val_f1_score	21.25%	15.95%	25.59%

5.3.5 Importancia de atributos - Eliminación recursiva

Sobre el árbol sin poda (α = 0) se utilizó Recursive Feature Elimination (RFE) para la eliminación recursiva de atributos. Se obtuvieron los 3 atributos más importantes: **age**, **avg_glucose_leve**l y **bmi**.
A continuación se observan las métricas para ambas condiciones para el conjunto de evaluación:

	Arbol sin Poda	Árbol sin Poda - 3 Features más importantes
val_precision	12.39%	12.50%
val_recall	28.00%	24.00%
val_f1_score	17.18%	16.44%

Se observa que reduciendo la cantidad de features no se genera un impacto considerable en los valores de las métricas.

6. Conclusiones

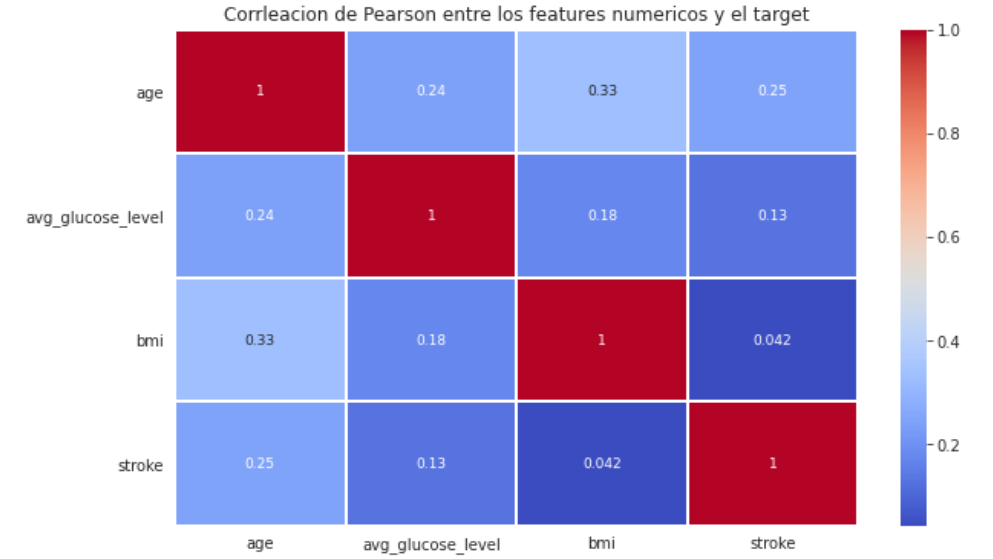
Luego de haber realizado análisis con splits random, usar 50 Fold y 10 Fold Cv y haber recorrido el espectro de alfa (α) en distintos valores de poda, podemos concluir que este dataset y modelo seleccionado, es muy sensible al desbalanceo de clases existentes en el target. Esto lo entendemos al ver que siempre que los modelos de árboles de decisión entrenados se prueban con datos no vistos anteriormente (evaluación) la performance en la métrica seleccionada es muy baja, lo cual se entiende como alto overfitting. Creemos que más allá de la exploración de hiperparametros, el tipo de modelo utilizado y/o el volumen de datos manejado no es suficientemente bueno para producir un modelo lo suficientemente general como para predecir de manera correcta datos no vistos.
En cuanto a las diferentes métricas, vimos una diferencia fuerte entre precision y recall, siendo el recall mucho más alto que la precisión, con lo cual si logramos una alta cantidad de verdaderos positivos, va a ser a costa de predecir muchos falsos positivos, con el correspondiente impacto que ello implica en este dataset (Por ejemplo, diagnosticar probabilidad de stroke a una persona que no va a tenerlo, y tal vez dar medicación errónea con impacto posiblemente negativo en la salud del paciente).
Con respecto a los distintos métodos utilizados, vemos que al utilizar cross-validation los resultados tienden a ser más generalizables, y luego al usar 10 Folds en vez de 50 Folds, volvemos a ver mejoras. Al podar, también vemos cierta mejora en la generalización, pero no la suficiente como para generar un modelo con buena performance productiva.

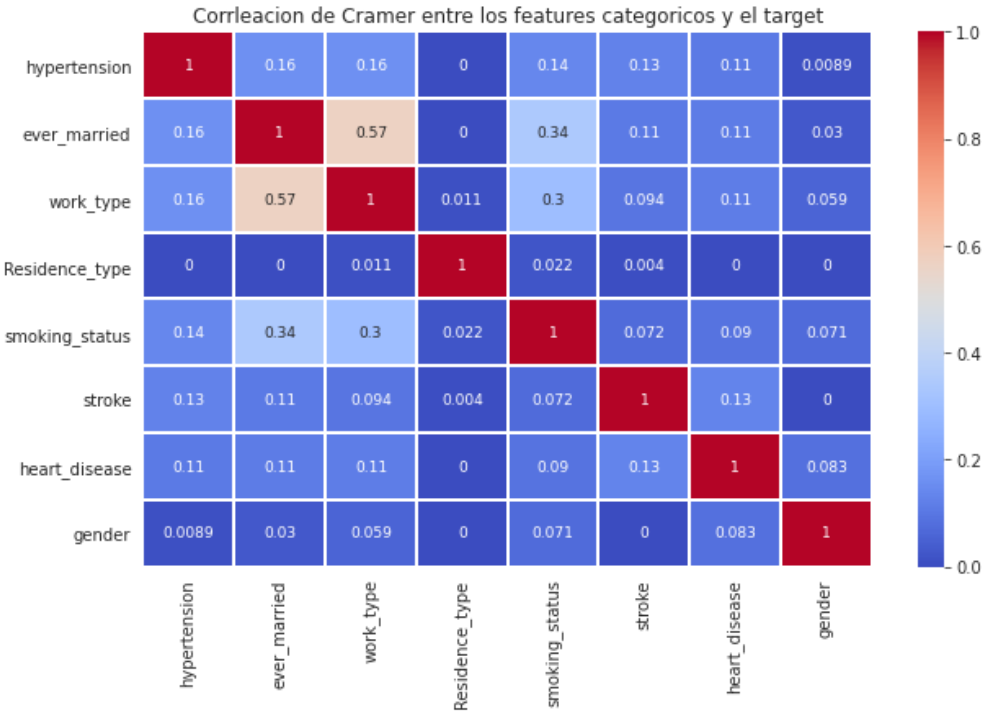
7. Bibliografía y Lecturas

- ***Machine Learning*** - Tom M. Mitchell (1997)
- ***Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*** - Aurelien Geron (2017)
- ***An Introduction to Statistical Learning*** - Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani (2013)
- ***An overview of correlation measures between categorical and continuous variables*** (https://towardsdatascience.com/)
- ***Dealing with Imbalanced Data*** (https://medium.com)
- ***Beyond the F-1 score: A look at the F-beta score*** (https://medium.com)
- ***All about Categorical Variable Encoding*** (https://medium.com)
- ***Feature Selection Techniques*** (https://medium.com)
- ***"MRMR" Explained Exactly How You Wished Someone Explained to You*** (https://medium.com)

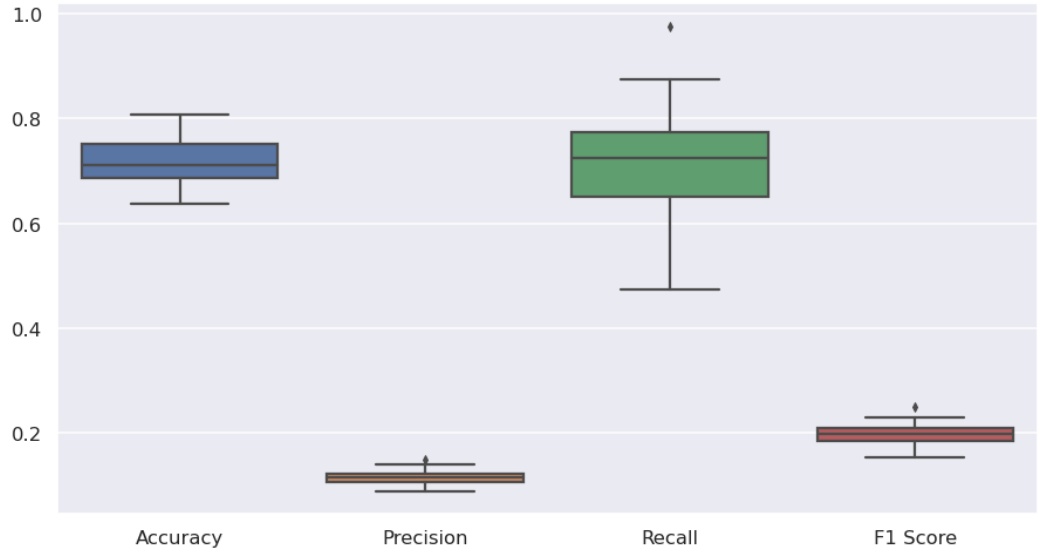
8. Anexos

8.1 Correlaciones y Mapas de Calor

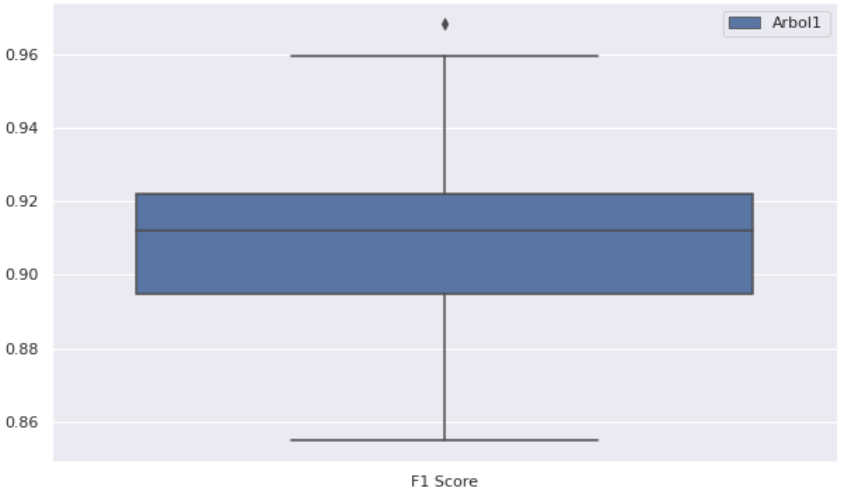




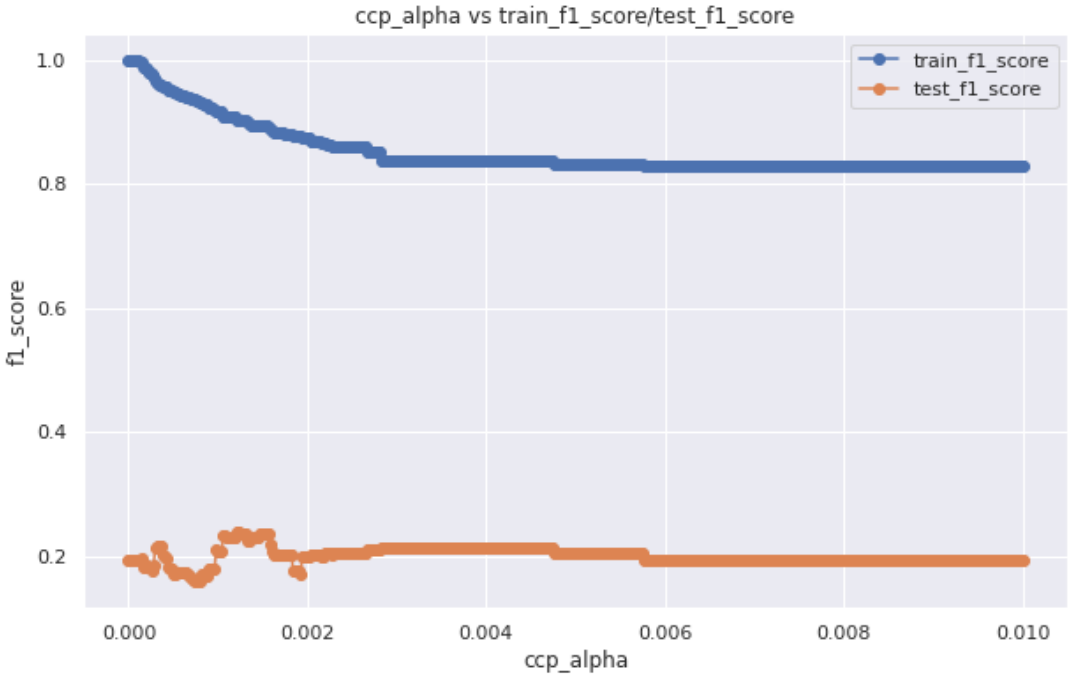
8.2 50 Random Seeds - BoxPlot de Métricas

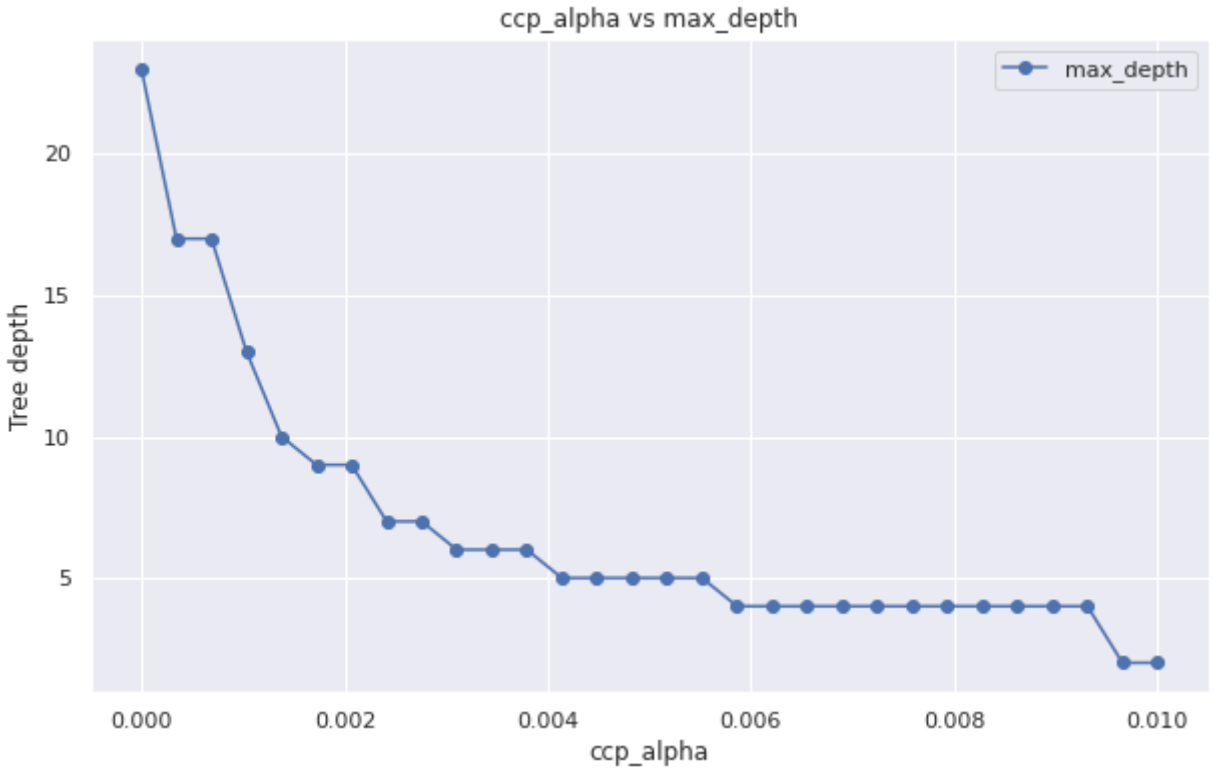


8.3 50 fold Cv - Boxplot F1 Score

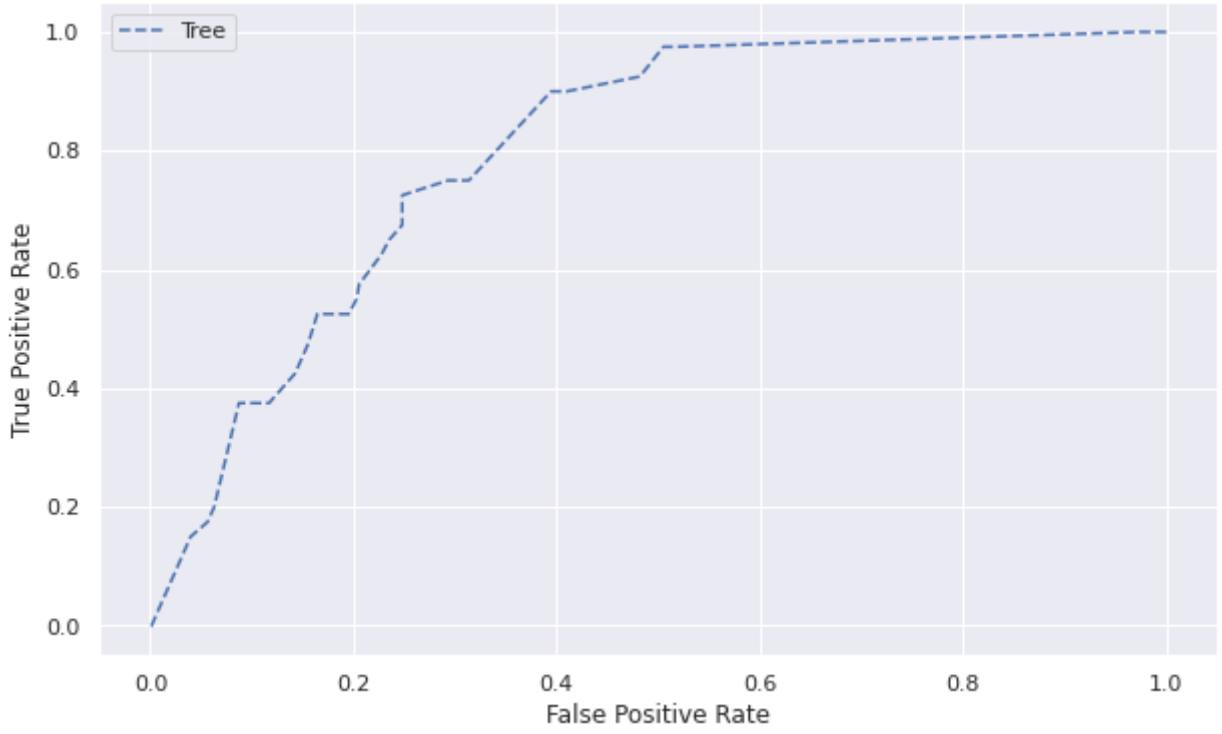


8.4 10-Fold CV + Poda




ccp_alpha vs max_depth
max_depth
Tree depth
20
15
10
5
0.000
0.002
0.004
0.006
0.008
0.010
ccp_alpha

8.5 Curva ROC


Tree
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate