Cervantes Rubí Brandon

Inteligencia Artificial 2022-2, G.3, 03/05/2022
Practica No.13 Pronostico (Bosques aleatorios).

Objetivo.

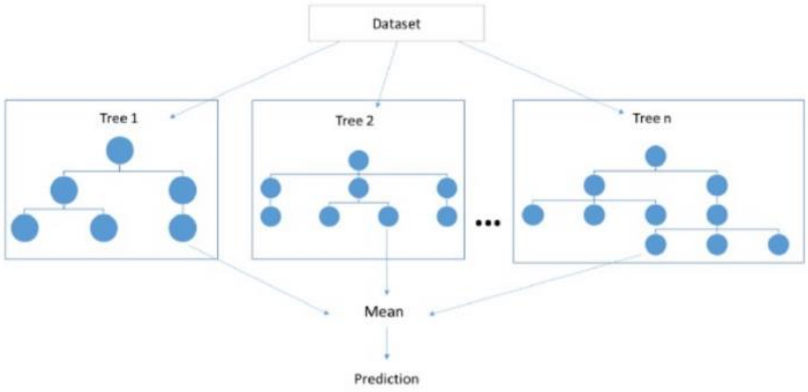
Pronosticar el área del tumor de pacientes con cáncer de mama a través de bosques aleatorios.

Características.

Estudios clínicos a partir de imágenes digitalizadas de pacientes con cáncer de mama de Wisconsin (WDBC, Wisconsin Diagnostic Breast Cancer).

Los bosques aleatorios son algoritmos que buscan generalizar más las soluciones que nos pueden entregar los árboles aleatorios mediante la combinación de varios de estos árboles en la misma estructura de tal forma que cada uno de ellos aporte una solución similar para después llegar a un consenso evitando además un sobreajuste en los datos.

Al igual que en los árboles de decisión principalmente se trabajará con 2 tipos de estructuras para los bosques aleatorios, primero los bosques compuestos por árboles de clasificación los cuales trabajan con etiquetas como resultados (A, B, C), (0/1), etc. tomando aquella etiqueta que se repita más entre la salida de los árboles como la salida del modelo y después están los bosques compuestos por árboles de regresión, dichos arboles están diseñados para entregar como resultado un valor continuo y para los bosques se busca obtener un promedio con los resultados de cada árbol.



Desarrollo.

Como primer paso tenemos la importación de las librerías necesarias para trabajar con el conjunto de datos de los pacientes a segmentar. `pandas` para la manipulación de los datos, `numpy` para el manejo de matrices, `matplotlib` y `seaborn` para la visualización de estos datos mediante gráficos de diferente tipo dependiendo del análisis.

Después tenemos que hacer la lectura de nuestros datos los cuales están relacionados a las características de los tumores presentados en cada uno de las pacientes relacionadas como su área, textura, radio simetría, etc. las cuales nos ayudaran mas adelante para pronosticar el área del tumor.

Cervantes Rubí Brandon

Inteligencia Artificial 2022-2, G.3, 03/05/2022
Practica No.13 Pronostico (Bosques aleatorios).

Tenemos que leer los datos con ayuda de pandas el cual nos genera un `dataframe` con 569 registros y 12 columnas asociadas a las características de los tumores presentados, de los cuales podemos hacer una agrupación con el método `goupby` de pandas para conocer la cuenta de los diagnósticos benignos (357 registros) y los malignos (212 registros).

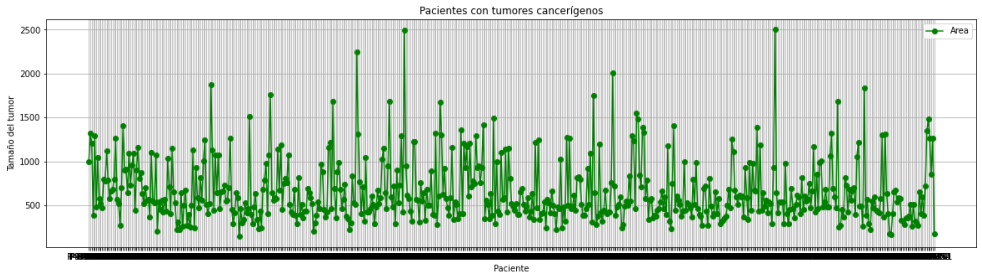
A continuación, se muestra una parte del conjunto de datos leído

	IDNumber	Diagnosis	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	ConcavePoints	Symmetry	FractalDimension
0	P-842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871
1	P-842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667
2	P-84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	0.05999
3	P-84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744
4	P-84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	0.05883
...	...	...	...	...	...	...	...	...	...	...	...	...
564	P-926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05623
565	P-926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	0.05533
566	P-926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590	0.05648
567	P-927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397	0.07016
568	P-92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.1587	0.05884

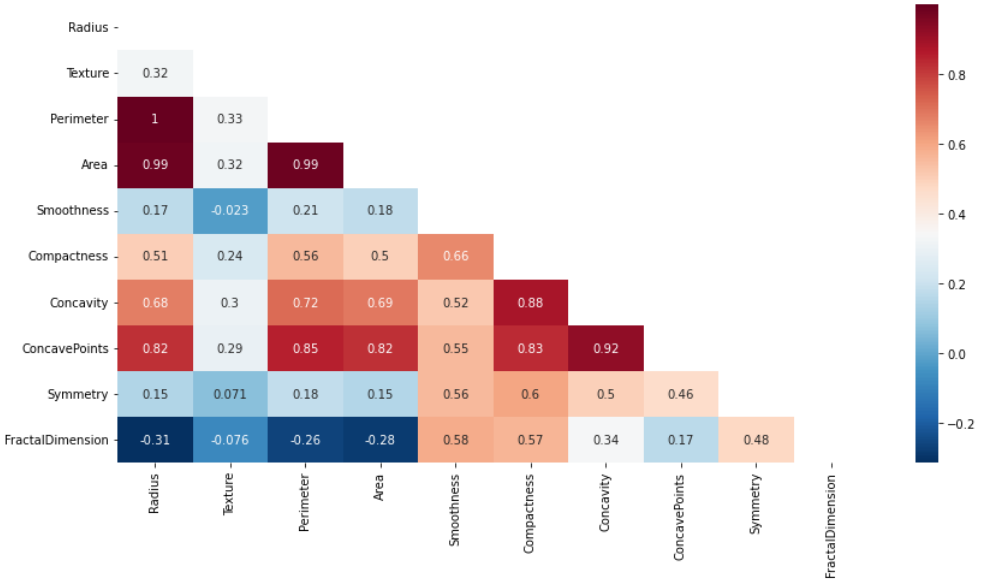
Es posible además hacer una descripción de nuestro dataframe con el atributo describe el cual nos muestra cuantos registros no nulos existen por cada columna, la media y la desviación estándar de los valores por columna, cual es el valor mínimo y máximo por variable, por ultimo los cuantiles (25%, 50% y 75%) donde podemos tomar el 50% como la mediana de los datos.

	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	ConcavePoints	Symmetry	FractalDimension
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	0.062798
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	0.007060
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	0.049960
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	0.057700
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	0.061540
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	0.066120
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	0.097440

Por ultimo se grafico a cada uno de los pacientes en relación con el área de su tumor para visualizar el rango de valores en el conjunto de datos.



Considerando las 12 variables en el modelo se decidió hacer un análisis para la selección de características, realizando un análisis de correlación con la matriz de correlaciones se trazando un mapa de calor para hacer el resultado más legible pues este mapa nos muestra con un indicé de -1 a 1 el nivel de correlación que tiene cada variable con las demás. A continuación, se muestra el mapa de calor obtenido resaltando las relaciones fuertes entre área y perímetro, radio y área, Concavity con la mayoría de las variables, Concave points de igual manera con la mayoría de las variables por lo cual se decidió que se eliminarían del modelo.



Al igual que en prácticas anteriores las variables independientes que se seleccionaron fueron las siguientes.

1.- Texture.

2.- Perimeter.

3.- Smoothness.

4.- Compactness.

5.- Symmetry.

6.- FractalDimension.

```
X = np.array(BCancer[['Texture',
                      'Perimeter',
                      'Smoothness',
                      'Compactness',
                      'Symmetry',
                      'FractalDimension']])
pd.DataFrame(X)
```

Con las variables independientes del modelo seleccionadas se procedió con la aplicación del algoritmo, Para ello fue necesaria la clase `RandomForestRegressor`, `classification_report` de `sklearn`, `mean_squared_error`, `mean_absolute_error`, `r2_score` y por último `model_selection` para la partición del conjunto de entrenamiento y conjunto de pruebas.

En esta ocasión la variable dependiente a pronosticar del modelo es el Área del tumor y las variables independientes se nombrará variables predictoras. Se hizo un nuevo arreglo de `numpy` con todas estas variables al cual se le llamo `X`, por otra parte, `Y` es el arreglo correspondiente a la columna del Área en el conjunto de datos.

```
Y = np.array(BCancer[['Area']])
pd.DataFrame(Y)
```

Con estos dos arreglos se utilizó `model_selection` para separar los 2 conjuntos en 4, dos correspondientes a las `X` y `Y` de entrenamiento que representan el 80% del total los datos (455 registros) y otros 2 correspondientes a las `X` y `Y` del conjunto de pruebas que representan el otro 20% (114 registros) del conjunto de datos., por último para utilizar esta librería fue necesario establecer una semilla de aleatoriedad y el atributo `shuffle` en True para que mezcle los registros evitando tomarlos tal y como están lo cual podría llevar a una mala división de los conjuntos.

```
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y,
                                                                                test_size = 0.2,
                                                                                random_state = 1234,
                                                                                shuffle = True
```

Con los conjuntos de datos establecidos, para generar el bosque aleatorio de árboles de regresión es necesario utilizar la clase `RandomForestRegressor` para crear un objeto a partir de ella sobre la que es posible definir ciertos parámetros de los árboles que conformaran al bosque. Los hiperparametros con los que se trabajó fueron los siguientes; Con `n_estimators` podemos definir el número de árboles que conformaran el bosque, con `max_deep` podemos establecer la profundidad del árbol, con `min_samples_split` se define un mínimo de hojas que se tienen que generar, con `min_samples_leaf` podemos controlar el número de registros que definan una hoja de tal forma que los nodos hoja que no cumplan con este mínimo no se consideran para las reglas que se generen al final y por ultimo tenemos max_features para que cada árbol tome el número de variables especificadas correspondientes a las más importantes.

```
#Conf 4: Con las 4 variables mas importantes es 97.9%
PronosticoBA = RandomForestRegressor(n_estimators=150, max_depth=8,
min_samples_split=4, min_samples_leaf=2, random_state=0, max_features=4)
PronosticoBA.fit(X_train, Y_train)
```

Cuando se crea el objeto con los parámetros establecidos, se usa el método `fit` pasando como parámetros los conjuntos de datos de entrenamiento que corresponden al 80% de los registros. Para generar las métricas de rendimiento del modelo primero es necesario utilizar el método `predict` pasando como parámetros el conjunto de datos de prueba que corresponde a las variables independientes del modelo `X_test` para generar los valores del área predichos por el mismo (`Y_pronostico`) y comparar los resultados con el conjunto de datos que tiene los valores reales para estos registros, es decir `Y_test`.

Con los pronósticos hechos por el modelo (`Y_pronostico`) y los valores reales (`Y_test`) se creó un nuevo `dataframe` donde podemos hacer una comparación rápida entre ellos observando que en la mayoría de los casos los resultados del área del tumor se acercan bastante al valor real lo cual nos da un indicativo de que el indicé de score será alto.

Para validarlo se utilizó la función r2_score pasando como parámetros los conjuntos de los valores predichos para el conjunto de pruebas y sus valores reales obteniendo diferentes valores de precisión dependiendo del modelo con el que se estuviese tratando, pero en general siendo un valor de entre el 97% y el 98%.

```
r2_score(Y_test, Y_Pronostico)
```

En una primera instancia se probo con el modelo sin configurar los hiperparametros y con la altura máxima del árbol el cual nos entregó un score del 98.97% siendo alto, pero tampoco tanto como el árbol de regresión de prácticas anteriores el cual obtuvo una precisión por encima del 99% por lo cual podemos comprobar que la generalización en el caso de los bosques aleatorios es mejor que si solamente se trabajase con un solo árbol.

Considerando una altura de 8, mínimo 4 nodos hoja con 2 registros en cada uno y variando el numero de arboles que compone el bosque aleatorio se llegaron a las siguientes métricas de score.

No. de árboles.	50	100	150	200
Score (%)	98.52	98.62	98.65	98.64

Como podemos apreciar los bosques aleatorios tienden a estabilizar su score a medida que el numero de arboles o estimadores en él sube, en este caso no hay mucha diferencia entre usar 50 árboles o 200.

Con el atributo `feature_importances_` podemos generar una lista ordenada con la importancia de cada una de las variables que se consideraron, a partir de esta lista se creó un `dataframe` ordenado por valor de importancia en el que podemos observar variaciones dependiendo de la configuración y pudiendo tomar las más altas configurando el atributo `max_features`.

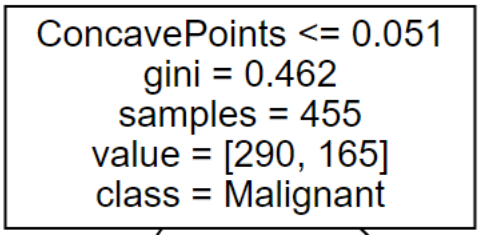
Para las cuatro configuraciones presentadas se generaron diferentes bosques aleatorios partiendo de diferentes valores para los hiperparametros, se generaron las gráficas del último árbol y fue necesario trabajar con los módulos `export_graphviz`, `plot_tree` y `export_text` de `sklearn.tree`.

Cervantes Rubí Brandon

Inteligencia Artificial 2022-2, G.3, 03/05/2022
Practica No.13 Pronostico (Bosques aleatorios).

Árbol con todas las variables del modelo.

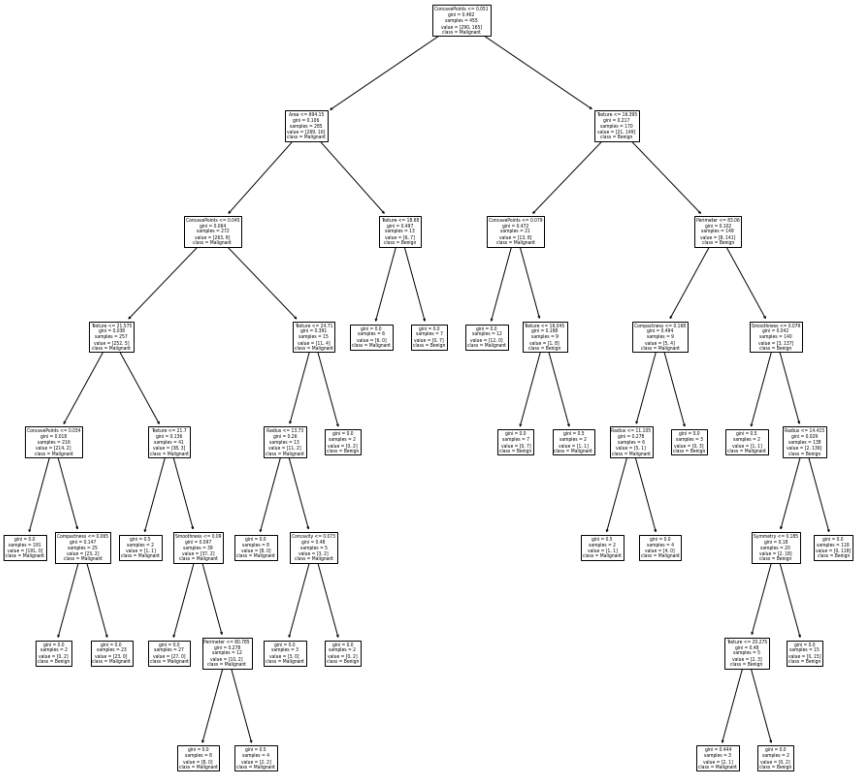
Nodo del que se parte.



	Variable	Importancia
7	ConcavePoints	0.755958
1	Texture	0.116267
5	Compactness	0.031812
3	Area	0.031220
2	Perimeter	0.030337
6	Concavity	0.011823
0	Radius	0.009819
4	Smoothness	0.006852
8	Symmetry	0.005911
9	FractalDimension	0.000000

El árbol generado presenta 7 niveles y el nodo raíz parte del perímetro con un score del 93.85%.

ConcavePoints es la variable que presenta un mayor nivel de entropía o que mayor importancia toma para el modelo, por el contrario, la menos importante es Dimensión Fractal.



Matriz de confusión del árbol.

Clasificación Real	Benign	Malignant
Benign	64	3
Malignant	4	43

Reporte de parámetros.

	precision	recall	f1-score	support
Benign	0.94	0.96	0.95	67
Malignant	0.93	0.91	0.92	47
accuracy			0.94	114
macro avg	0.94	0.94	0.94	114
weighted avg	0.94	0.94	0.94	114

Cervantes Rubí Brandon

Inteligencia Artificial 2022-2, G.3, 03/05/2022
Practica No.13 Pronostico (Bosques aleatorios).

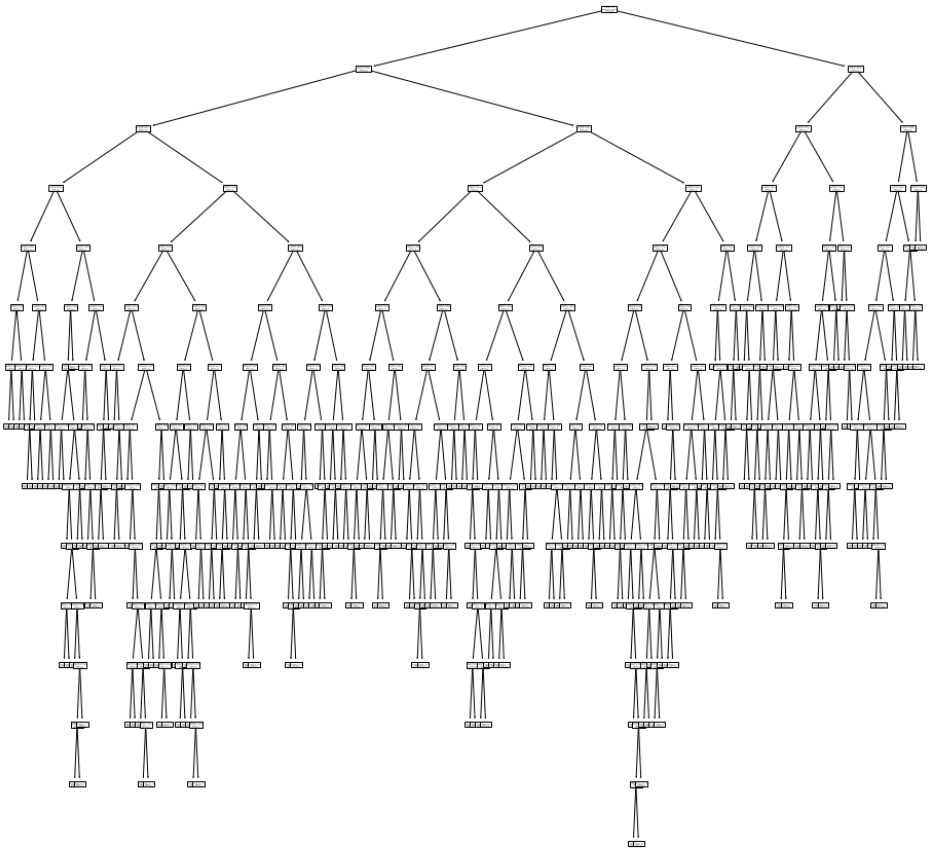
Bosque aleatorio sin modificar los hiperparametros.

Nodo del que se parte.

```
Perimeter <= 108.2
squared_error = 100424.538
samples = 286
value = 632.945
```

```
MAE: 16.1857
MSE: 1370.2268
RMSE: 37.0166
Score: 0.9897
```

La altura del árbol presentado es de 14, el nodo raíz del que se parte es el perímetro con un score final del 98.97%.



La lista de variables más importantes muestra al perímetro como la más importante y a la suavidad como la menos importante.

	Variable	Importancia
1	Perimeter	0.993391
3	Compactness	0.001506
5	FractalDimension	0.001397
0	Texture	0.001284
4	Symmetry	0.001273
2	Smoothness	0.001148

Cervantes Rubí Brandon

Inteligencia Artificial 2022-2, G.3, 03/05/2022
Practica No.13 Pronostico (Bosques aleatorios).

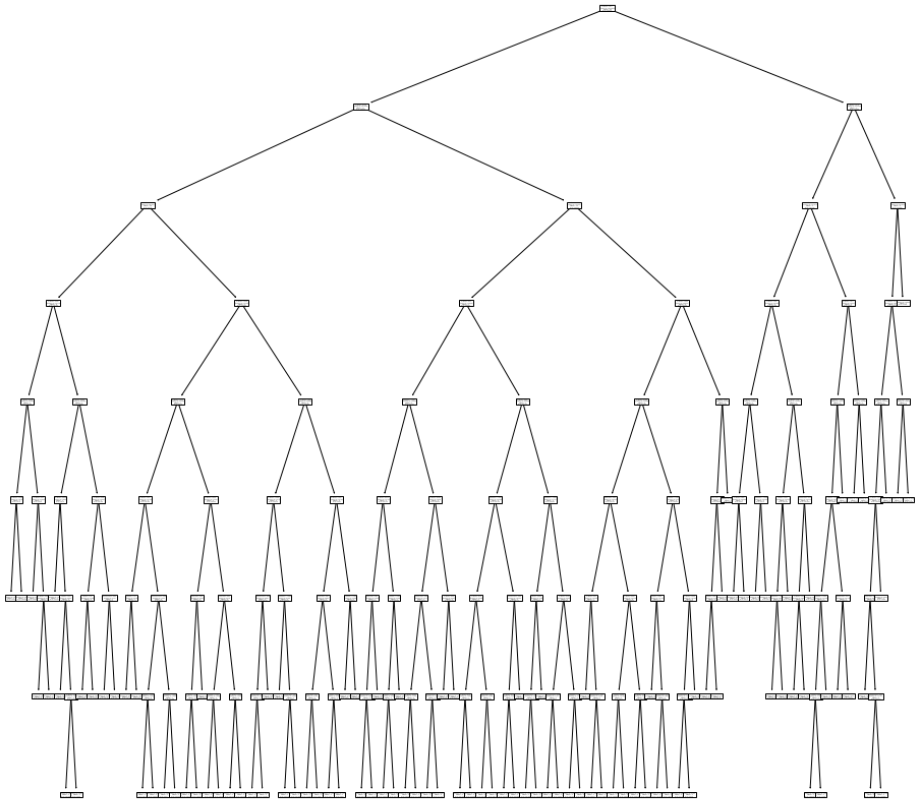
Bosque aleatorio con 100 estimadores, 8 niveles de profundidad, mínimo 4 nodos hoja con 2 registros en cada una.

Nodo del que se parte.

```
Perimeter <= 108.2
squared_error = 100424.538
samples = 286
value = 632.945
```

```
MAE: 16.7955
MSE: 1832.1701
RMSE: 42.8039
Score: 0.9863
```

La altura del árbol presentado es de 8, el nodo raíz del que se parte es el perímetro con un score final del 98.63%.



La lista de variables más importantes muestra al perímetro como la más importante y a la suavidad como la menos importante.

	Variable	Importancia
1	Perimeter	0.994995
3	Compactness	0.001248
0	Texture	0.001152
5	FractalDimension	0.000974
4	Symmetry	0.000855
2	Smoothness	0.000777

Cervantes Rubí Brandon

Inteligencia Artificial 2022-2, G.3, 03/05/2022
Practica No.13 Pronostico (Bosques aleatorios).

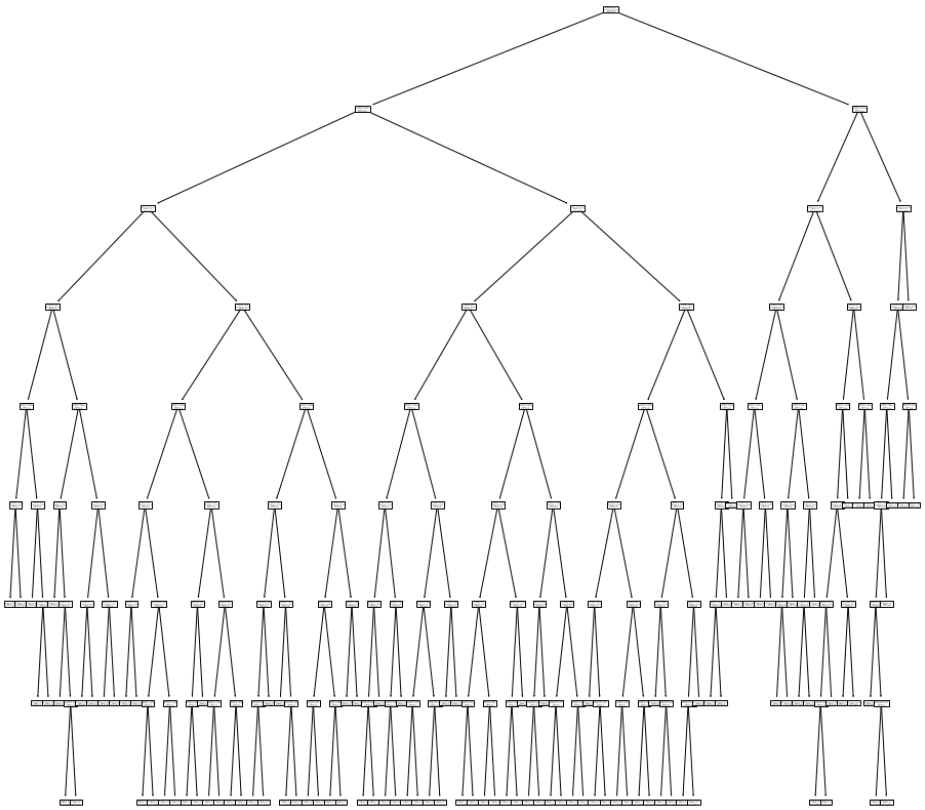
Bosque aleatorio con 150 estimadores, 8 niveles de profundidad, mínimo 4 nodos hoja con 2 registros en cada una.

Nodo del que se parte.

```
Perimeter <= 108.2
squared_error = 100424.538
samples = 286
value = 632.945
```

```
MAE: 16.9606
MSE: 1792.9044
RMSE: 42.3427
Score: 0.9866
```

La altura del árbol presentado es de 8, el nodo raíz del que se parte es el perímetro con un score final del 98.66%.



La lista de variables más importantes muestra al perímetro como la más importante y a la simetría como la menos importante.

	Variable	Importancia
1	Perimeter	0.995214
3	Compactness	0.001166
0	Texture	0.001052
5	FractalDimension	0.001037
2	Smoothness	0.000813
4	Symmetry	0.000718

Cervantes Rubí Brandon

Inteligencia Artificial 2022-2, G.3, 03/05/2022
Practica No.13 Pronostico (Bosques aleatorios).

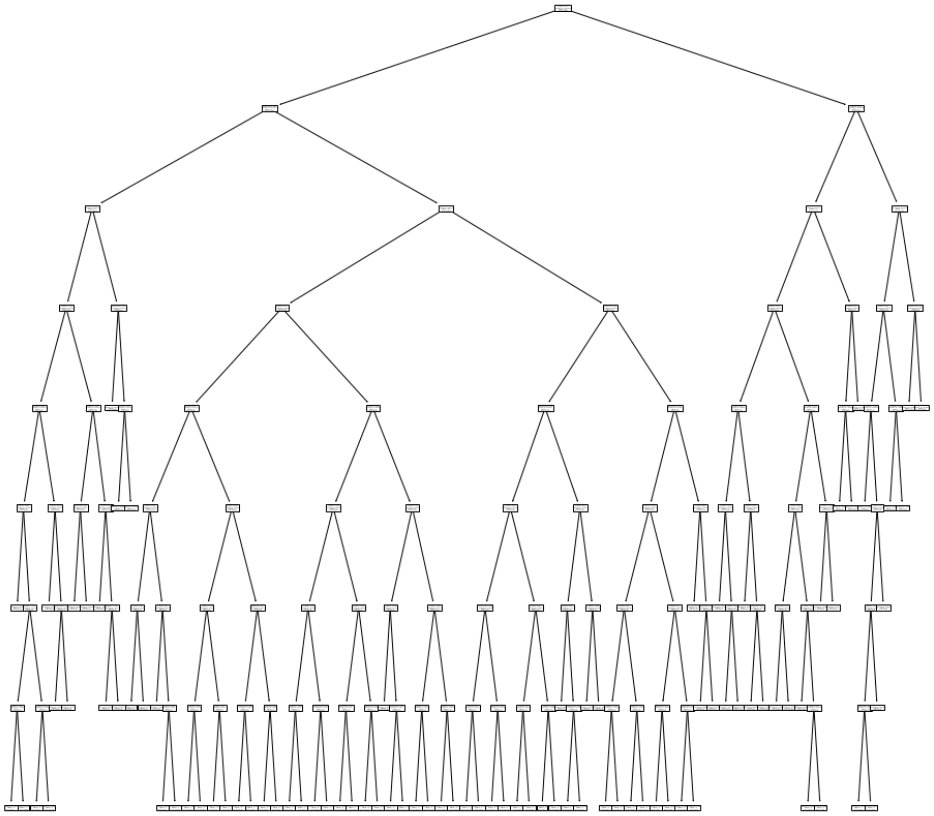
Bosque aleatorio con 150 estimadores, 8 niveles de profundidad, mínimo 4 nodos hoja con 2 registros en cada una y considerando las 4 variables más significativas.

Nodo del que se parte.

```
Perimeter <= 108.2
squared_error = 100424.538
samples = 286
value = 632.945
```

```
MAE: 18.1460
MSE: 2801.5598
RMSE: 52.9298
Score: 0.9790
```

La altura del árbol presentado es de 8, el nodo raíz del que se parte es el perímetro con un score final del 98.66%.



La lista de variables más importantes muestra al perímetro como la más importante y a la simetría como la menos importante.

	Variable	Importancia
1	Perimeter	0.885712
3	Compactness	0.055904
5	FractalDimension	0.037240
0	Texture	0.009625
2	Smoothness	0.007069
4	Symmetry	0.004451

Utilizando la librería `export_test` es posible generar la lista de reglas en forma textual que componen al algoritmo para darnos una idea de cómo se están haciendo las divisiones en cada uno de los nodos presentados en el árbol del bosque. A continuación, se muestra una sección de la lista de reglas las cuales definen al árbol con una altura de 8 niveles, 150 estimadores, mínimo 4 nodos hoja y 2 registros en cada uno de los nodos hoja.

```
|--- Perimeter <= 108.20
|   |--- Perimeter <= 81.49
|   |   |--- Perimeter <= 67.96
|   |   |   |--- Perimeter <= 59.23
|   |   |   |   |--- Perimeter <= 53.84
|   |   |   |   |   |--- Perimeter <= 49.84
|   |   |   |   |   |   |--- value: [179.90]
|   |   |   |   |   |--- Perimeter >  49.84
|   |   |   |   |   |   |--- value: [203.57]
|   |   |   |   |--- Perimeter >  53.84
|   |   |   |   |   |--- Perimeter <= 56.10
|   |   |   |   |   |   |--- value: [226.52]
|   |   |   |   |   |--- Perimeter >  56.10
|   |   |   |   |   |   |--- Perimeter <= 58.88
|   |   |   |   |   |   |   |--- value: [245.75]
|   |   |   |   |   |   |--- Perimeter >  58.88
|   |   |   |   |   |   |   |--- value: [261.93]
```

Por último, es posible hacer nuevas predicciones con cada uno de los bosques aleatorios que se generaron teniendo la certeza de que el modelo se comporta de buena manera con nuevos datos por tener un score superior al 98%.

Conclusiones.

A lo largo de esta práctica se construyeron diferentes configuraciones de bosques aleatorios con árboles de regresión para pronosticar el área del tumor de un nuevo paciente en base a otras características del paciente como lo pueden ser el perímetro, la textura, etc. dichas configuraciones se generaron modificando los hiperparametros del modelo siendo variables dentro del mismo que se tienen que ajustar para llegar a los mejores resultados entre estas se encuentran; el numero de arboles para el bosque, la altura de estos árboles, el número mínimo de nodos hoja, la cantidad de variables tomadas para generar los árboles entre otras.

Se logro comprobar que este tipo de algoritmos (Bosques aleatorios) tienden a generalizar mas un problema de regresión que un árbol de decisión aplicado a estos problemas ya que generan un consenso de predicciones a cambio de un mayor procesamiento y tiempo.

Fuentes.

pandas documentation — pandas 1.4.1 documentation. (2022). Pandas. Recuperado 2022, de https://pandas.pydata.org/docs/index.html#

sklearn.ensemble.RandomForestRegressor. (2022). Scikit-Learn. Recuperado 2022, de https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Dataset. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)