INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

LICENCIATURA EN CIENCIA DE DATOS

UNIDAD DE APRENDIZAJE

DESARROLLO DE APLICACIONES PARA ANÁLISIS DE DATOS

EXAMEN 3ER PARCIAL

NOMBRE DE LOS ALUMNOS:

DE LUNA OCAMPO YANINA

PROFESOR:

OCAMPO BOTELLO FABIOLA

GRUPO:

5CDM1

FECHA:

18/06/2022

Información básica
Objetivo particular
Resumen ejecutivo
Información relevante
Descripción de las variables dadas en el conjunto de datos
Diccionario de datos

Análisis del conjunto de datos de Wine Quality.

Información básica:

Repositorio obtenido de: UC Irvine Machine Learning Repository

Link del conjunto de datos:
https://archive.ics.uci.edu/ml/datasets/Wine+Quality

Autor/es:
* Paulo Cortez, University of Minho, Guimarães, Portugal,
http://www3.dsi.uminho.pt/pcortez
* A. Cerdeira, F. Almeida, T. Matos y J. Reis, Viticulture Commission of the Vinho Verde Region (CVRVV), Porto, Portugal
@2009

Fecha de publicación: 2009-10-07

Objetivo particular
—

Resumen ejecutivo
que para que, pero muy pequeño
De forma particular, el reporte deberá contener un resumen ejecutivo. Dicha sección tiene como objetivo particular describir de forma resumida el objetivo del análisis, los diferentes ejercicios de modelado y los motivos para elegir un modelo sobre otros, de acuerdo al objetivo particular. Esta sección también sirve para atraer al público
interesado en el tema en cuestión a la lectura completa del contenido del reporte.
—

Información relevante:

- El dataset “wine quality-red.csv” está relacionado con las variantes del vino tinto Português Vinho Verde. Para más detalles, consultar: http://www.vinhoverde.pt/en/
- Este dataset puede ser visto como una tarea tanto de clasificación como de regresión porque existen etiquetas preestablecidas dentro de este, por lo que se espera que el modelo aprenda de ellas y al momento de integrar datos no analizados, sea capaz de asignarles las etiquetas aprendidas previamente.
- Las instancias están en orden y no balanceadas, por ejemplo hay más vinos normales, que excelentes o de baja calidad. Esto se debe tomar en cuenta para evitar sesgos durante el análisis.
- Se podrían utilizar algoritmos de detección de datos anómalos para los vinos de baja calidad.

- No todas las variables son relevantes, por lo que sería interesante realizar una selección de características, dado un análisis de correlación.
- El dataset contiene 12 variables con 1599 instancias.
- El dataset no cuenta con valores nulos.

Descripción de las variables dadas en el conjunto de datos:

1. fixed acidity → acidez ajustada
2. volatile acidity → acidez volátil
3. citric acid → ácido cítrico
4. residual sugar → azúcar residual
5. chlorides → cloruros
6. free sulfur dioxide → dióxido de azufre
7. total sulfur dioxide → dióxido de azufre total
8. density → densidad
9. pH → pH
10. sulphates → sulfatos
11. alcohol → alcohol

Variable de Salida:

12. quality → calidad

Diccionario de datos

A continuación, veremos el diccionario de datos de dicho conjunto con el fin de entender un poco más los datos y así proceder a asignar un modelo adecuado a lo analizado.

Nombre	**Significado**	**Tipo**	**Dominio**	**Media**	**Desv Est**	**Valores Nulos**	**Mín**	**Máx**
fixed acidity	Acidez ajustada	Float64 (Numerical)	{ 4.6 , 15.9 }	8.31	1.74	0	4.6	15.9
volatile acidity	Acidez Volátil	Float64 (Numerical)	{ 0.12 , 1.58 }	0.52	0.17	0	0.12	1.58
citric acid	Ácido Cítrico	Float64 (Numerical)	{ 0.0 , 1.0 }	0.27	0.19	0	0	1
residual sugar	Azúcar Residual	Float64 (Numerical)	{ 0.9 , 15.5 }	2.53	1.4	0	0.9	15.5
chlorides	Cloruros	Float64 (Numerical)	{ 0.012 , 0.611 }	0.08	0.04	0	0.01	0.6
free sulfur dioxide	Dióxido de Azufre	Float64 (Numerical)	{ 1.0 , 72.0 }	15.87	10.46	0	1	72
total sulfur dioxide	Dióxido de Azufre total	Float64 (Numerical)	{ 6.0 , 289.0 }	46.46	32.89	0	6	289
density	Densidad	Float64 (Numerical)	{ 0.99007 , 1.00369 }	0.99	0.001	0	0.99	1
pH	pH	Float64	{ 2.74 , 4.01	3.31	0.15	0	2.7	4

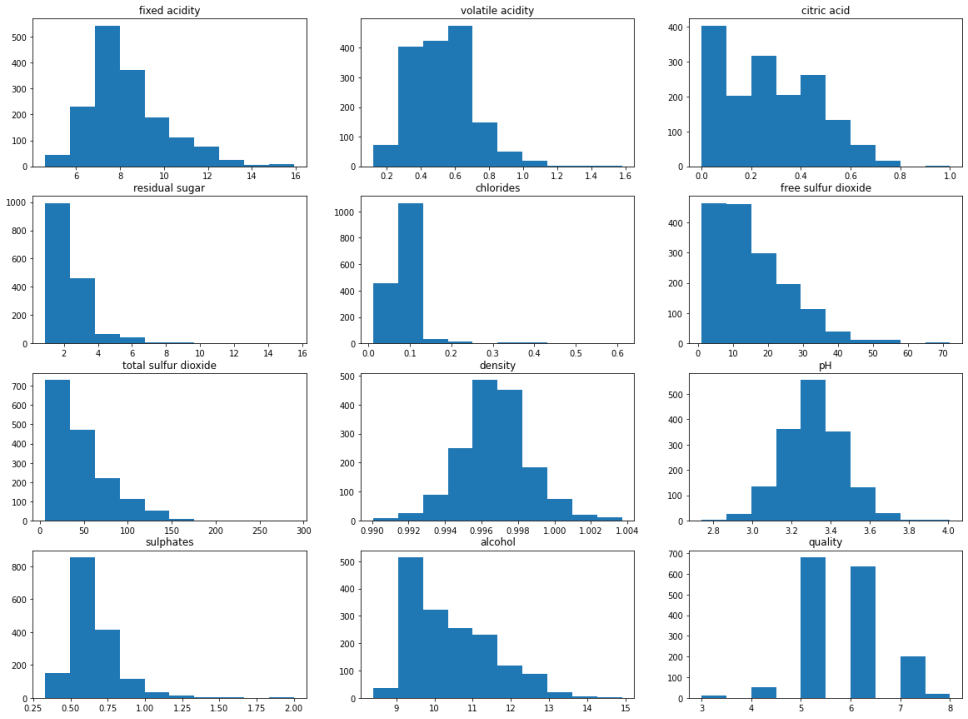
		(Numerical)	}				4	
sulphates	Sulfatos	Float64 (Numerical)	{ 0.33 , 2.0 }	0.65	0.16	0	0.33	2
alcohol	Alcohol	Float64 (Numerical)	{ 8.4 , 14.9 }	10.42	1.06	0	8.4	14.9
quality	Calidad	int (Integer)	{ 3 , 8 }	5.63	0.8	0	3	8

Regresión lineal múltiple

Parte 1, sin manipulación.

Este primer modelo se realizó sin ningún tipo de manipulación de datos. Para todo esto, lo primero que debemos hacer es analizar nuestros datos, obteniendo la cantidad de observaciones, los valores mínimos, máximos, promedio, desviación estándar y quantiles de cada una de las variables presentadas.

Obtenemos los histogramas para ver la distribución que tiene cada variable.



Una vez, teniendo este análisis, como se había mencionado previamente, no se hará ningún tipo de limpieza, por lo que pasaremos a la regresión lineal. Para esta debemos asignar el valor de X y de Y, en este caso la variable “X” es igual a todas las variables excluyendo únicamente la de quality. Y por el contrario, la variable “y” deberá ser igual a quality.

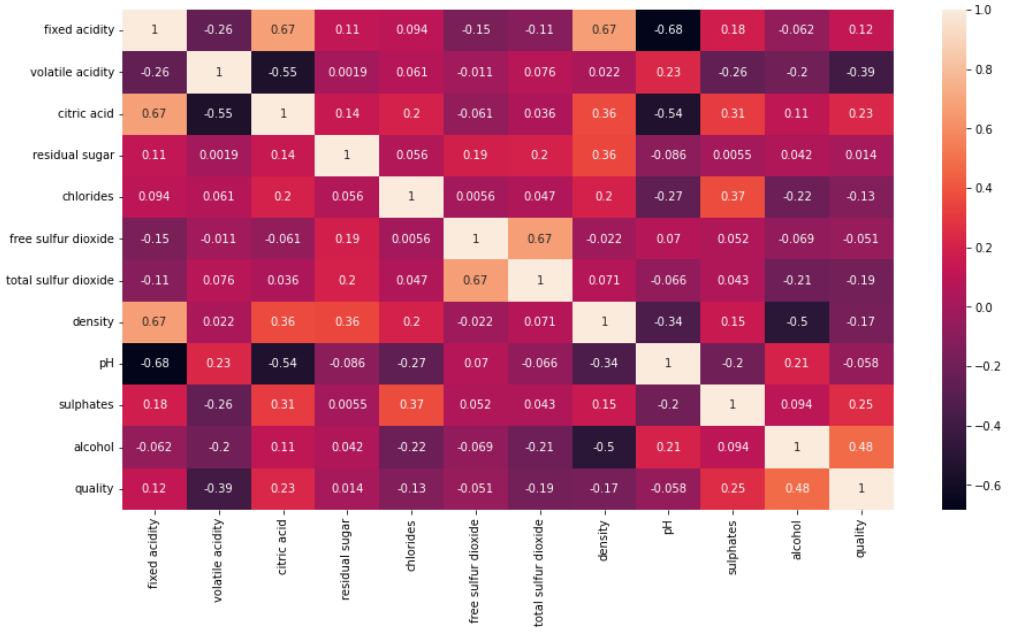
Teniendo esto, procedemos a utilizar la librería de sklearn para importar el modelo de regresión que utilizaremos en este caso. Los resultados obtenidos, son los siguientes:

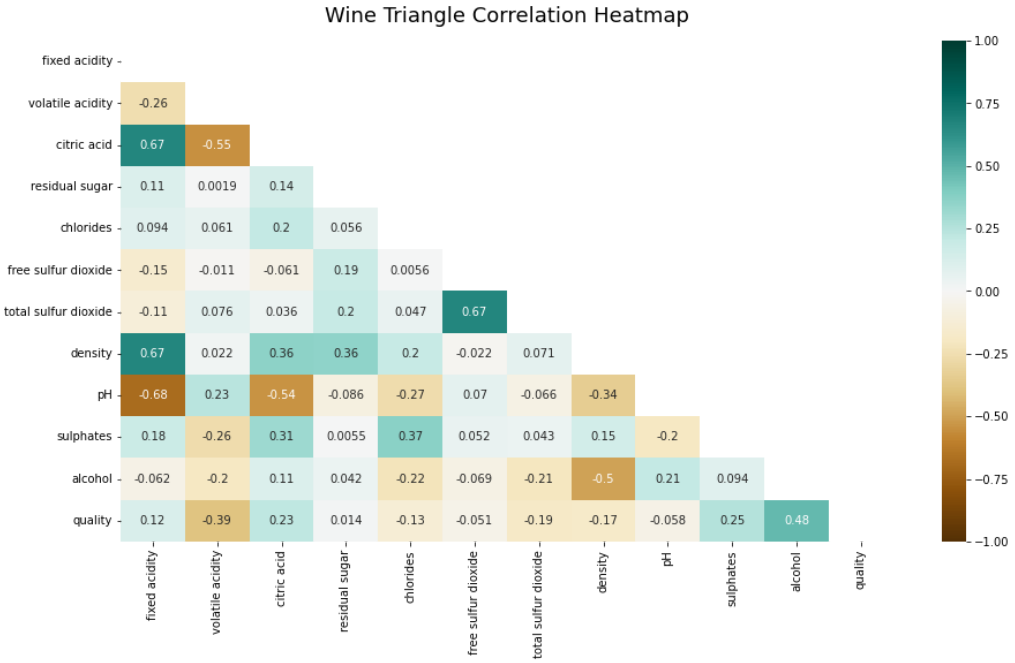
```
Coefficients:
 [ 3.21701286e-02 -1.03467859e+00 -1.53320498e-01  1.23460437e-02
 -1.61715049e+00  5.08258596e-03 -3.32744691e-03 -1.57794200e+01
 -3.84377830e-01  8.10208705e-01  2.88021969e-01]
Intercept:  19.611580823864184
Model:
y = 19.6116 + 0.0322x1 + -1.0347x2 + -0.1533x3

--Metricas--
 Mean squared error: 0.366301
 r2_score: 0.3869
 Mean Absolute Error: 0.4678
 explained_variance_score: 0.388398
 Precision:0.386904
```

Se explicará brevemente las métricas que imprimimos con este modelo:
Coeficientes: son los pesos ponderados de cada variable, que es lo que le da más peso o menos peso dentro de la función.
Intercept: este es el valor mínimo que se espera cuando todas las variables son igual a 0.
Model: función que define a la recta que intenta predecir el comportamiento de los datos.
Mean squared error: mide el promedio de los errores al cuadrado. Entre más este cerca de 0, mejor.
r2_score: representa la proporción de la diferencia o varianza en términos estadísticos para una variable dependiente que puede ser explicada por una variable o variables independientes. En este caso, entre más se acerque a 1, es mejor.
Mean Absolute Error: es la diferencia entre las observaciones y la salida del modelo. En este se ignora el signo para no producir cancelaciones entre valores positivos y negativos.
explained_variance_score: ve la variabilidad de los datos, sin tomar en cuenta los valores anómalos.

Concluimos que en un modelo sin limpieza los valores obtenidos no son óptimos a lo esperado de un modelo de regresión, es un buen parteaguas para comenzar con el análisis esperado, pero siempre será necesario buscar hacer una limpieza para obtener mejores resultados con base en qué es lo que necesitamos y para qué.
Por lo que a continuación veremos cómo se comporta el modelo con una limpieza dentro de estos.

Parte 2, con manipulación.
En este segundo ejercicio generamos un modelo cuyos datos de entrada hayan sido modificados con base en las observaciones de los resultados del primer modelo, como se ha mencionado previamente.
Empezaremos visualizando la correlación de todas las variables para poder empezar a deducir desde ahí.
A continuación veremos los resultados obtenidos en código:

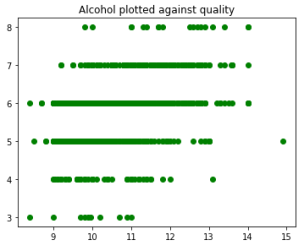


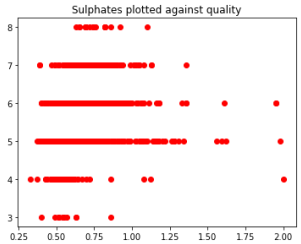


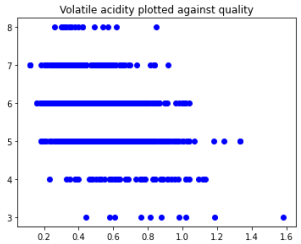
Podemos visualizar dos tipos de gráficas de correlación, ambas con los mismos resultados, esta vemos que no hay variables con una correlación tan alta, pero de entre todas ellas, elegimos las siguientes:

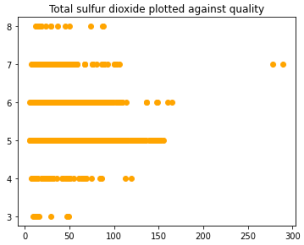
* Alcohol
* Sulphates
* Volatile Acidity
* Total sulfur dioxide

Tomando en cuenta estas, se ha graficado la dispersión con nuestra variable objetivo.









Al momento de poner las variables en nuestra nueva regresión, obtenemos valores demasiados bajos, valores que harían que nuestro modelo fuera poco óptimo en cuanto a los resultados. Por lo que decidimos intentar con diferentes combinaciones de variables y agregar un poco más a las primera seleccionadas, tomando en cuenta las siguientes:

```
   volatile acidity  citric acid  total sulfur dioxide  density  sulphates  \
0              0.70         0.00                  34.0   0.9978       0.56
1              0.88         0.00                  67.0   0.9968       0.68
2              0.76         0.04                  54.0   0.9970       0.65
3              0.28         0.56                  60.0   0.9980       0.58
4              0.70         0.00                  34.0   0.9978       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
```

Teniendo estas ya establecidas, procedemos a normalizar y escalar los datos para que cada una de las variables tuviera el mismo peso que las demás. Obteniendo como resultado lo siguiente una vez aplicado esto.

```
      volatile acidity  citric acid  total sulfur dioxide  density  sulphates  \
0                0.700         0.00                  34.0  0.99780       0.56
1                0.880         0.00                  67.0  0.99680       0.68
2                0.760         0.04                  54.0  0.99700       0.65
3                0.280         0.56                  60.0  0.99800       0.58
4                0.700         0.00                  34.0  0.99780       0.56
...                ...          ...                   ...      ...        ...
1594             0.600         0.08                  44.0  0.99490       0.58
1595             0.550         0.10                  51.0  0.99512       0.76
1596             0.510         0.13                  40.0  0.99574       0.75
1597             0.645         0.12                  44.0  0.99547       0.71
1598             0.310         0.47                  42.0  0.99549       0.66

      alcohol
0         9.4
1         9.8
2         9.8
3         9.8
4         9.4
...       ...
1594     10.5
1595     11.2
1596     11.0
1597     10.2
1598     11.0
```

Separamos nuestros datos, para entrenamiento dejamos el 80% de estos y para prueba solamente el 20%. Ya hecho esto, decidí implementar diferentes modelos para ver cuál era el que resultaba mejor creando una comparativa con los datos normalizados y no normalizados.

- Training Three models
    - Linear Regression
        - No normalizado

```
---- Non Normalized Data ---------

Coefficients:
 [-1.31569299e+00 -9.63737980e-02 -2.04622319e-03  1.03183547e+01
  6.88820125e-01  3.10973694e-01]

Intercept:  -7.510808014745659
y = -7.5108 + -1.3157x1 + -0.0964x2 + -0.0020x3 + 10.3184x4 + 0.6888x5 + 0.3110x6

--Metricas--
 Mean squared error: 0.394285
 r2_score: 0.3112
 Mean Absolute Error: 0.4759
 explained_variance_score: 0.323742
 precision: 0.311245
```

        - Normalizado

```
---- Normalized Data ----

Coefficients:
 [-2.69837905 -0.21503096 -0.77097041  0.22305386  1.3373465   3.79573548]

Intercept:  5.136611521854631
y = 5.1366 + -2.6984x1 + -0.2150x2 + -0.7710x3 + 0.2231x4 + 1.3373x5 + 3.7957x6

--Metricas--
 Mean squared error: 0.394285
 r2_score: 0.3112
 Mean Absolute Error: 0.4759
 explained_variance_score: 0.323742
 Precision:0.311245
```

    - Ridge
        - No normalizado

```
---- Non Normalized Data ----

Fitting 5 folds for each of 7 candidates, totalling 35 fits
Best Score:  0.3349575703701436
Best Params:  {'alpha': 1}
```

        - Normalizado

```
---- Normalized Data ----

Fitting 5 folds for each of 7 candidates, totalling 35 fits
Best Score:  0.3337607058462327
Best Params:  {'alpha': 0.1}
```

- Lasso
  - No normalizado

```
---- Non Normalized Data ---------

Coefficients:
 [-2.294496    0.04884578 -0.73166235  0.          1.20629595  3.57993614]

Intercept:  5.084548043960922
y = 5.0845 + -2.2945x1 + 0.0488x2 + -0.7317x3

--Metricas--
 Mean squared error: 0.379087
 r2_score: 0.3655
 Mean Absolute Error: 0.4828
 explained_variance_score: 0.367687
 precision: 0.365503
```

  - Normalizado

```
---- Normalized Data ----

Coefficients:
 [-2.294496    0.04884578 -0.73166235  0.          1.20629595  3.57993614]

Intercept:  5.084548043960922
y = 5.0845 + -2.2945x1 + 0.0488x2 + -0.7317x3

--Metricas--
 Mean squared error: 0.390430
 r2_score: 0.3180
 Mean Absolute Error: 0.4740
 explained_variance_score: 0.327133
 Precision:0.317980
```

- Random Forest
  - No normalizado

```
---- Non Normalized Data ----

Fitting 5 folds for each of 4 candidates, totalling 20 fits
Best Score:  0.4043239080170958
Best Params:  {'n_estimators': 500}
```

Métricas:

```
---- Non Normalized Data ---------


--Metricas--
 Mean squared error: 0.282020
 r2_score: 0.5280
 Mean Absolute Error: 0.3816
 explained_variance_score: 0.529406
 precision: 0.527969
```

- Normalizado

```
---- Normalized Data ----

Fitting 5 folds for each of 4 candidates, totalling 20 fits
Best Score:  0.4247533828654462
Best Params:  {'n_estimators': 500}
```

Métricas

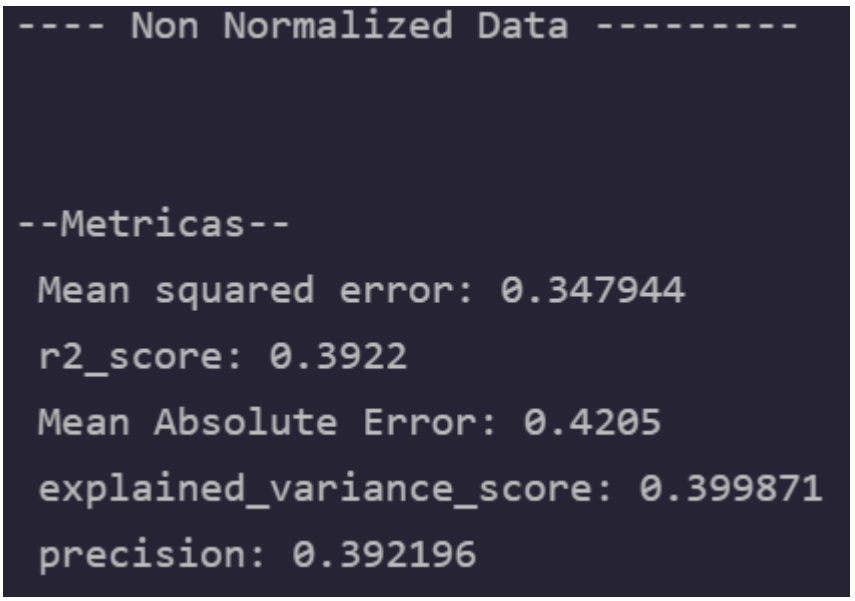
```
---- Normalized Data ----


--Metricas--
 Mean squared error: 0.349602
 r2_score: 0.3893
 Mean Absolute Error: 0.4217
 explained_variance_score: 0.396905
 Precision:0.389300
```

Para cada modelo se entrenó con datos no normalizados y normalizados. Para Ridge, Lasso y Random Forest se utilizó GridSearchCV para encontrar los mejores parámetros.

Parte 3

Similarmente, el tercer ejercicio debe consistir en la creación de un modelo con base en lo aprendido en los resultados de los dos primeros.

Después de realizar una comparativa en los resultados que daban los algoritmos de regresión utilizados en las segunda parte se determina que el algoritmo con el mejor rendimiento predictivo para este conjunto de datos es el modelo de ensamble random forest, cuyo resultado fue:



Usando los datos sin normalizar y con las características determinadas en el análisis de correlación hecho con anterioridad.

En cuanto al algoritmo, después de usar Grid Search CV, se obtiene que el valor del hyperparameter *n_features* debe ser de 500 para obtener el rendimiento óptimo que el algoritmo puede ofrecer al conjunto de datos.

Recordemos que la naturaleza del Algoritmo Random Forest es ser un algoritmo de ensamble que junta a otros algoritmos en este caso árboles de decisión para obtener el mejor rendimiento predictivo.

Las ventajas de usar random forest son:

- Diversidad, debido al método de ensamble bagging el algoritmo crea samples aleatorias con las que entrena a los n árboles de decisión, de esta forma obteniendo árboles diferentes con outputs diferentes.
- No hace Sobreajuste, un poco de la mano con la técnica bootstrap samples permite que cada árbol sea diferente y por lo tanto no permite el ajustarse a los datos en sobremanera como si pasa en los árboles de decisión ordinarios.

Algunas de las desventajas de este algoritmo son:

- Complejidad, debido al numero de arboles de decision que este crea la complejidad de recorrer un árbol es de O(nlogn) esta se multiplica por la cantidad de árboles que se generan.
- No es tan fácilmente interpretable como lo es un árbol de decisión.