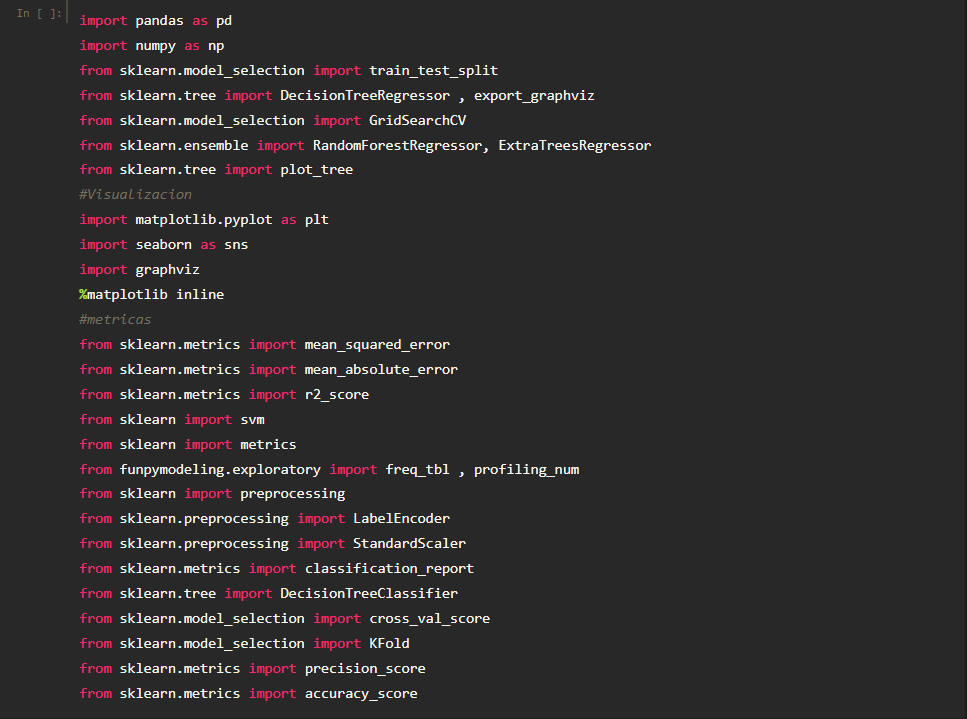
Actividad: Árboles y *random forest* para regresión y clasificación

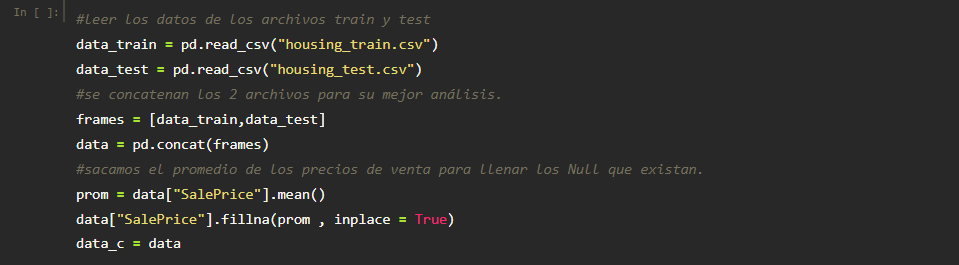
**Librerias a utilizar**

****

**Uso de las librerías**

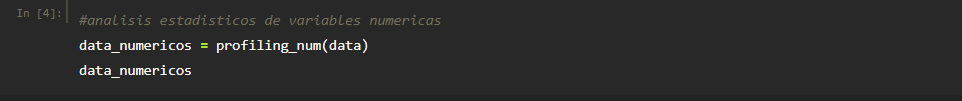
Sklearn fue la opción que se decidió usar, ya que en las practicas que se realizaron para cada tema de la materia, se uso la librería.

**Lectura y primer tratamiento de datos**



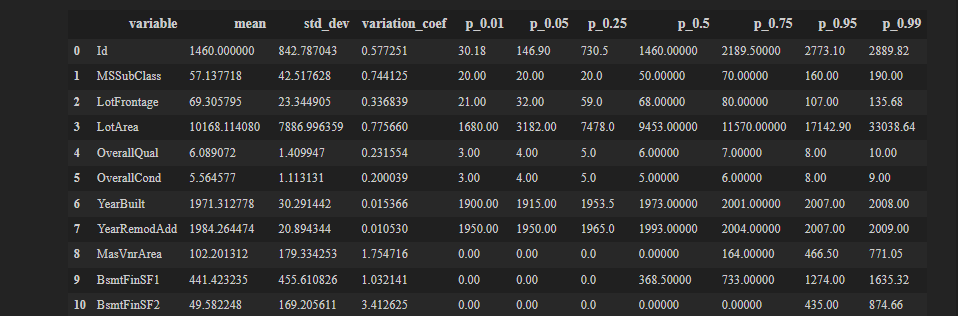
Se decidió que los valores en blanco de la columna SalePrice debian rellenarse con el promedio de los demás valores, ya que esa variable es indispensable al momento de hacer nuestro modelo de regresión y el promedio se acerca un poco mas a la vida real de los precios valorados de inmuebles.

**Primeros Análisis de datos**

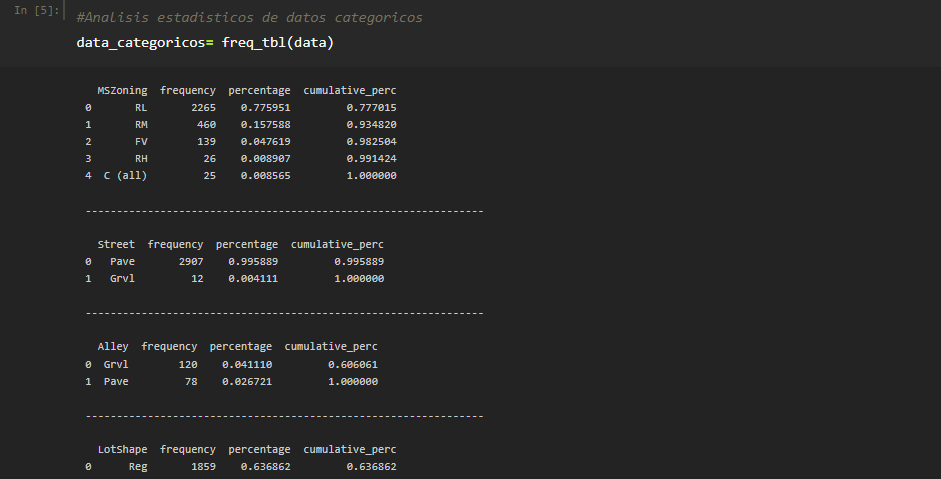
****

Con la librería **funpymodeling** , se pueden obtener fácilmente un análisis de los datos numéricos y categóricos (el detalle completo del análisis se encuentra en el código fuente anexo a este trabajo)

**Datos numéricos**

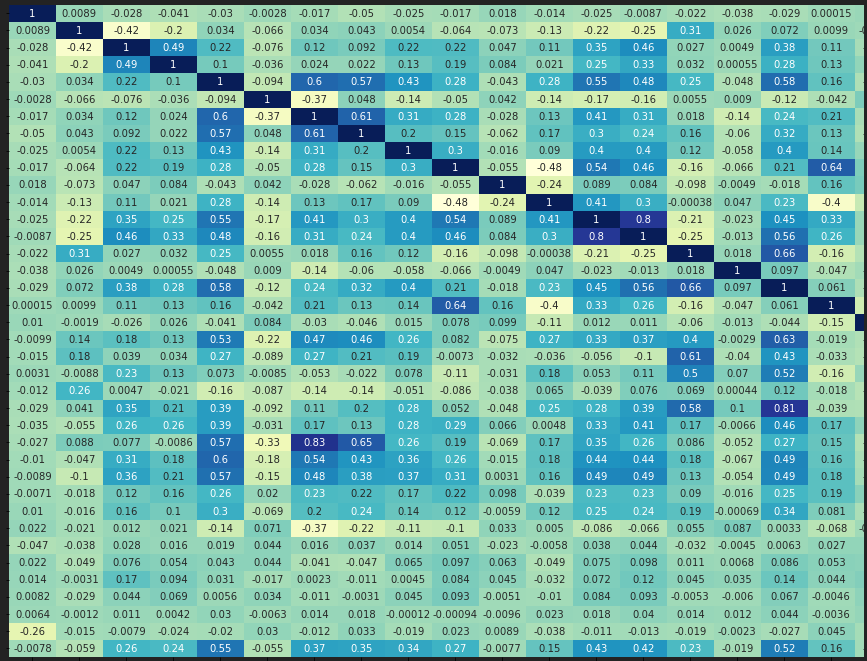


**Datos Categóricos**

****

**Matriz de Correlaciones**

Una matriz de correlación es una tabla que indica los coeficientes de conexión entre los factores. Cada celda de la tabla muestra la conexión entre los dos factores (la matriz completa se encuentra en el código fuente de este trabajo). **Mas adelante en el trabajo se explica mas a detalle esta relación.**

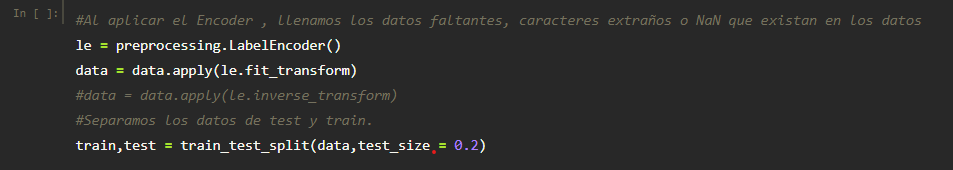
****

**Encoder**

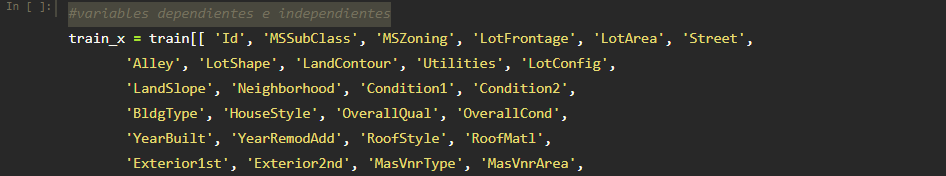
En el proceso de procesamiento de datos, a veces necesitamos digitalizar números discretos o texto, Cuando se usa Python para el procesamiento de datos, es muy fácil usar el codificador para convertir variables ficticias (datos virtuales). El codificador puede convertir el texto en el conjunto de datos en un valor de 0 o 1. LabelEncoder y OneHotEncoder son dos funciones en el paquete scikit-learn, que pueden implementar el proceso de conversión anterior.

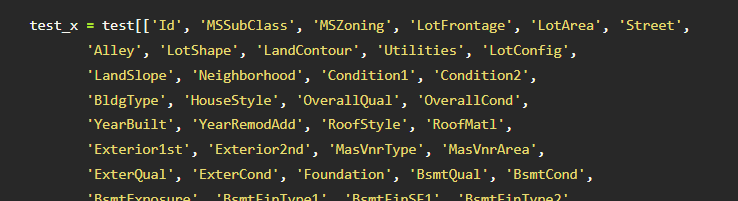
Se uso el encoder para así no borrar ningún dato de la base de datos, cuando aparecen datos categóricos en el conjunto de datos que queremos analizar, los datos en este momento no son ideales, **porque no podemos tratarlos matemáticamente.**

Así de esta forma , no se borra ningún dato de la base de datos y se puede procesar, por que, aunque sean NaN o estén blanco, son datos a procesar que dan una mayor eficacia a nuestro modelo de **árbol y random forest.**



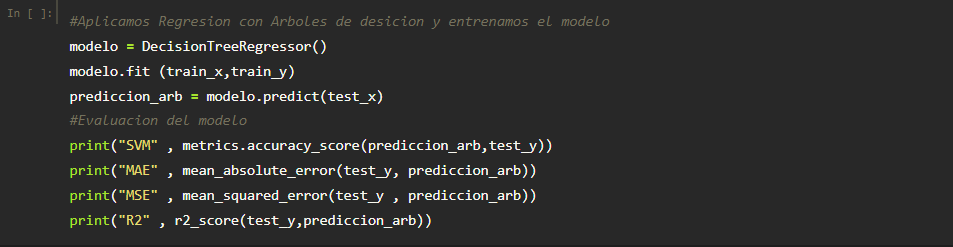
Una vez que se entrenan nuestros datos , separamos nuestras variables dependientes e independientes.





**Árbol de decisiones**

Es un algoritmo de clasificación y regresión para su uso en el modelado predictivo de atributos discretos y continuos. Se crea primero el modelo y se entrena.



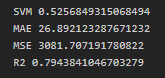
Se obtienen las primeras predicciones y primeros análisis del modelo construido:

**Error cuadrático medio, mean square error (MSE).** - se define como la media de la diferencia entre el valor real y el valor predicho o estimado al cuadrado

**Error absoluto medio, mean absolute error (MAE).** - se define como la diferencia en valor absoluto entre el valor real y el valor predicho.

**Raíz del error cuadrático medio, root mean square Error (RMSE)**. - se define como la raíz cuadrada de la media de la diferencia entre el valor real y el valor predicho o estimado al cuadrado.

**R2.-** el coeficiente de determinación, determina la capacidad de un modelo para predecir futuros resultados

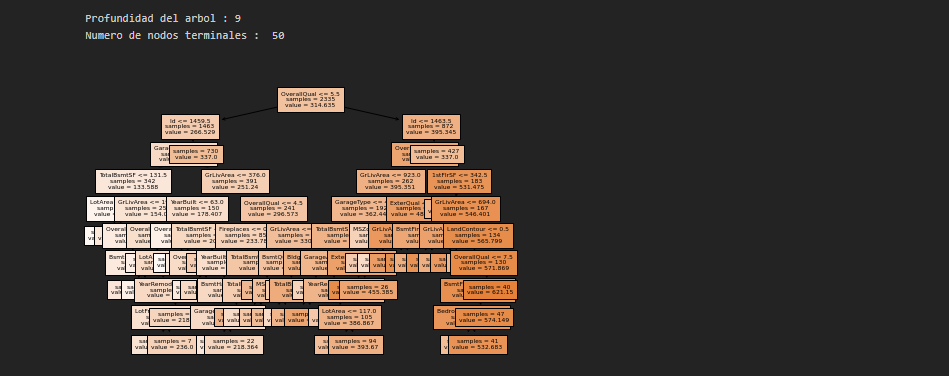
****

Como observamos en el análisis, nuestro modelo de **Árbol de Decisión** tiene un coeficiente de predicción de 0.794384, lo cual se puede considerar alto.

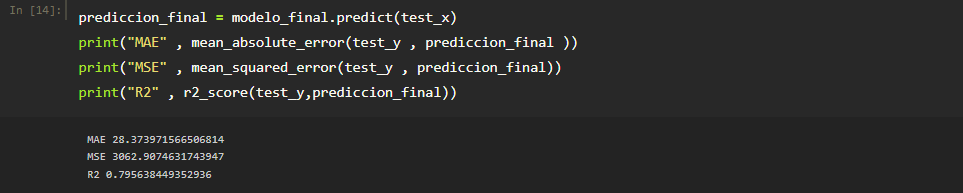
**Podado de árbol y estructura final**

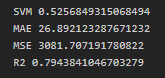
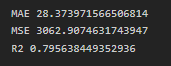
La estrategia para generar árboles más pequeños (poda) y con una menor varianza suele ser generar un árbol muy grande y después podarlo para obtener un sub-árbol. Así solo se enfoca en las variables importantes del modelo.



****

Se vuelven a analizar las métricas del modelo, comparando con el modelo de **Árbol de Decisión sin podar.**



****

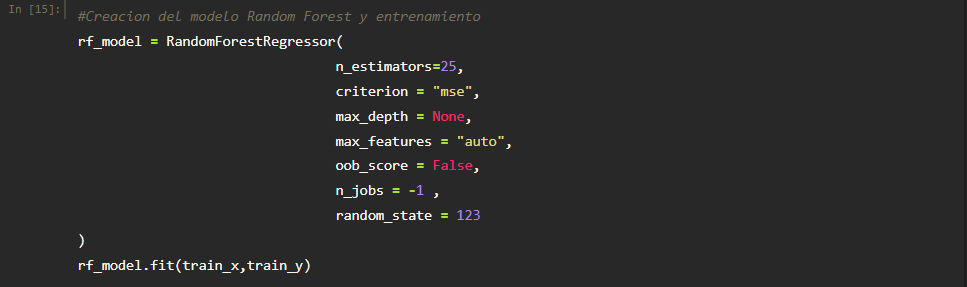
**Conclusión:**

Como se puede observar el modelo de **Árbol de Decisión con poda tiene mejor predicción que el Árbol de Decisión sin podar**

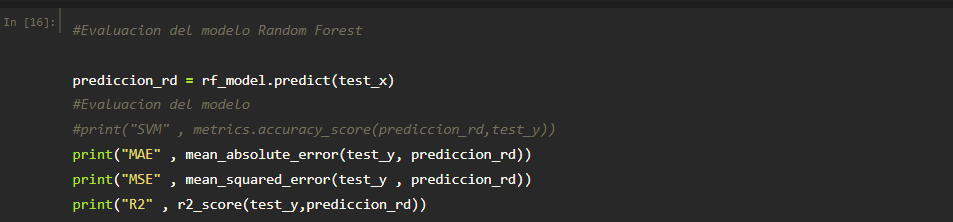
**Random Forest**

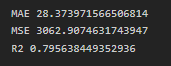
Es un método versátil de aprendizaje automático capaz de realizar tanto tareas de regresión como de clasificación. También lleva a cabo métodos de reducción dimensional, trata valores perdidos, valores atípicos y otros pasos esenciales de exploración de datos. Es un tipo de método de aprendizaje por conjuntos, donde un grupo de modelos débiles se combinan para formar un modelo poderoso. Se puede decir que son varios árboles de decisión en un solo modelo.

Creación del modelo y entrenamiento



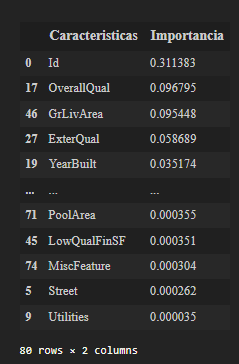
Se obtienen las primeras predicciones y primeros análisis del modelo construido, el R2 indica que **Random Forest** tiene una mayor predicción que los **Arboles de Decisión** para este ejemplo.

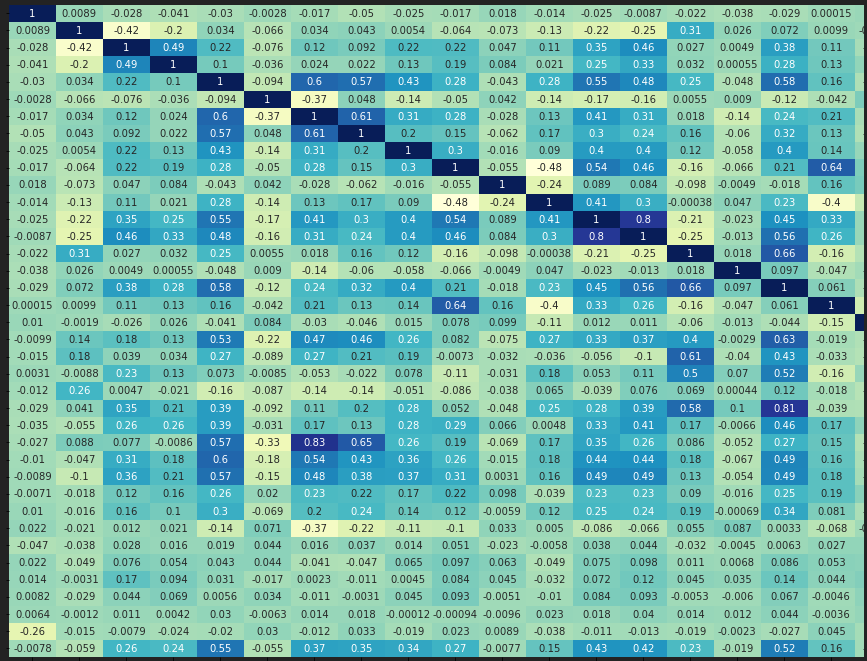
****

****

**Lista de importancia de cada variable para la regresión de Random Forest y Arboles de Decisión.**

Como se dijo anteriormente, la matriz de correlación y la lista de importancia tiene gran significado en el análisis de los datos ya que muestra las variables mas importantes y su relación con el entorno. En la siguiente grafica podemos ver un listado de mayor a menor las variables que se usan en el algoritmo para sus predicciones. (la matriz y el listado completos se encuentra en el código fuente de este trabajo).

****

****

**Comenta las ventajas y desventajas de cada modelo. De acuerdo con los resultados, ¿son realmente útiles los modelos creados para el conjunto de datos propuesto?**

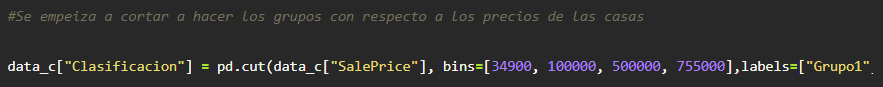
El modelo de **Arboles de Decisión** resulto tener un poco menos de confiabilidad que **Random Forest**, esto debido a la gran cantidad de datos que el algoritmo tiene que usar. Al momento de usar **Random Forest** y clasificarlos en diferentes arboles (y no en uno solo como lo hace el algoritmo de **Arboles de Decisión**), se puede obtener más fiabilidad.

**Algoritmos de Clasificación**

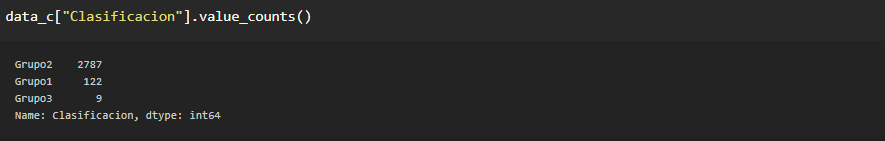
Los algoritmos de clasificación nos permiten, como su nombre lo dice, ordenar información de una manera especial basándonos en un criterio de ordenamiento**.** Se usan cuando el resultado deseado es una etiqueta discreta, en otras palabras, son útiles cuando la respuesta al problema cae dentro de un conjunto finito de resultados posibles.

**Discretizando los datos**

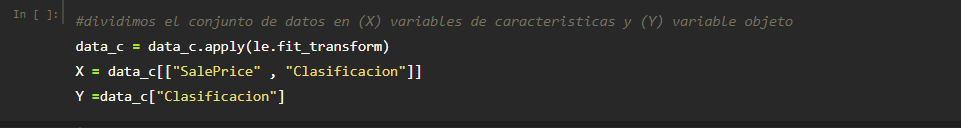
Se empieza a cortar a hacer los grupos con respecto a los precios de las casas

****

Comprobamos que realmente se hizo la clasificación:

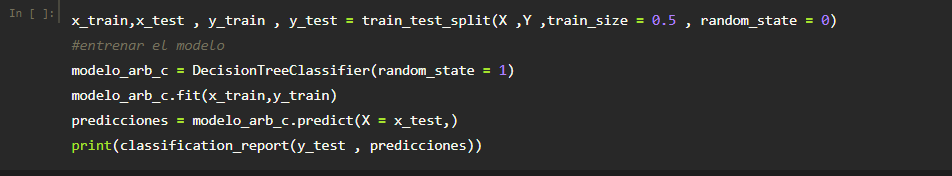


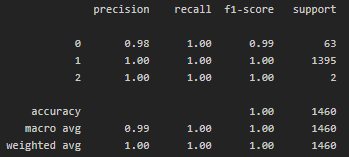
Se divide el conjunto de datos en (X) variables de características y (Y) variable objeto



**Modelo de clasificación**

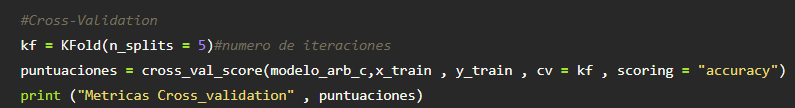
Se empieza por hacer el modelo de clasificación por árboles, se crea el modelo y se entrena, además obtenemos nuestros primeros análisis del modelo





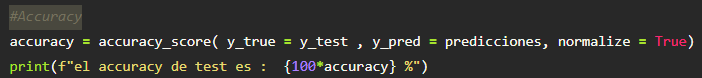
Como se puede observar en el análisis, tenemos un Accuracy y precisión muy altos , del 98% y 99% respectivamente, lo que indica que nuestro modelo puede predecir el grupo sin ningún problema.

**Cross-Validation K-Fold**

****

****

**Accuracy**

****



**Curvas ROC/AUC**

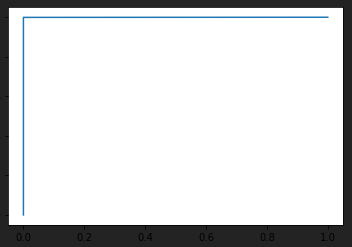
cuando se trata de un problema de clasificación, podemos contar con una curva AUC-ROC. Esta es una de las métricas de evaluación más importante para verificar el rendimiento de cualquier modelo de clasificación. Esta métrica se utiliza

para determinar el balance entre la detección de verdaderos positivos y evitar los

falsos positivos. Para ello, se muestra la proporción de detección de los verdaderos

positivos en el eje vertical y la proporción de los falsos positivos en el eje horizontal,

para un umbral o punto de corte determinado



Conclusiones

Para los algoritmos de regresión usados en este ejercicio , Arboles de Decisión y Random Forest, se obtuvo un rendimiento y una predicción alta, considerando la cantidad de datos usados. Muchos de estos datos se podrían consideran innecesarios (espacios en blanco, caracteres raros o NaN) pero para el algoritmo son necesarios por que tiene una **mas fiabilidad** al momento de determinar el precio de venta de la casa. Para ambos algoritmos se obtuvo una predicción muy alta , para el modelo usado.

En el caso de los algoritmos de regresión, se obtuvo resultados y análisis de predicción muy altos del casi .99%, lo cual indica que el modelo creado es perfecto para el uso de predicciones en categorías para la base de datos.

Bibliografías:

\*Análisis de Datos categóricos con Python. (s. f.). Python. Recuperado 21 de marzo de 2022, de <https://relopezbriega.github.io/blog/2016/02/29/analisis-de-datos-categoricos-con-python/>

\*Método de preprocesamiento de datos para digitalizar datos de categoría-LabelEncoder VS OneHotEncoder - programador clic. (s. f.). Python. Recuperado 21 de marzo de 2022, de <https://programmerclick.com/article/77681608357/>

*\*funpymodeling*. (2020, 16 septiembre). PyPI. <https://pypi.org/project/funpymodeling/>