**DECISION TREE FOR REGRESSION PROBLEMS**

Ora vedremo un implementazione di un albero di decisione in Python, prima di tutto importiamo le librerie che andremo ad utilizzare :



Successivamente andremo a recuperare i dati da un file in formato csv

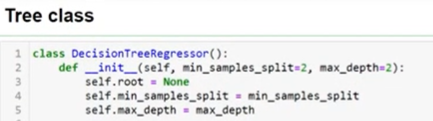


Ora iniziamo creando la classe nodo detta Node che servirà per la costruzione dell’albero binario dei decisione :

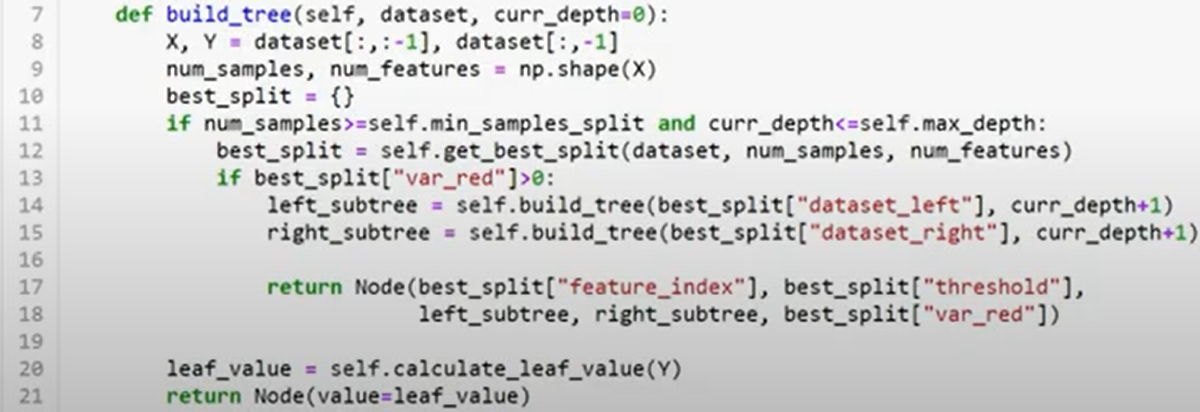
Graphical user interface, text, application

Description automatically generated

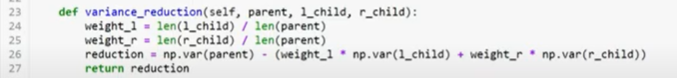
Adesso andiamo a definire la classe per creare il modello di apprendimento dell’albero di decisione:



Il primo metodo che vediamo risulta essere il costruttore della classe, il quale definisce il root che servirà per la navigazione dell’albero e le 2 condizione di stop nella costruzione dell’albero ovvero un numero minimo di esempi all’interno di ciascuna partizione dei nodi figli oppure una profondità massima per l’albero che vogliamo costruire.

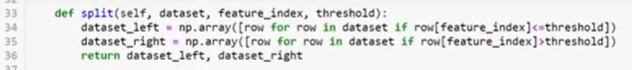


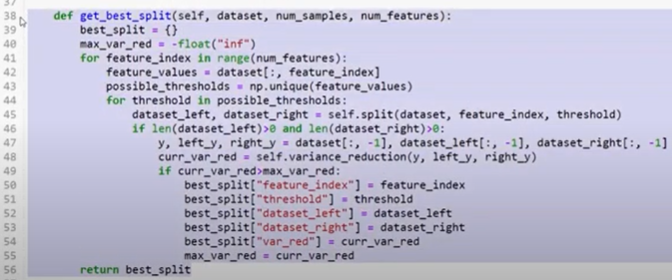
Questa è la funzione di ricorsione che va a costruire l’albero di decisione, nel secondo IF viene chiamata la funzione best\_split che in base al valore della riduzione di varianza se essa è maggiore di zero continuerà a generare test per dividere gli esempi

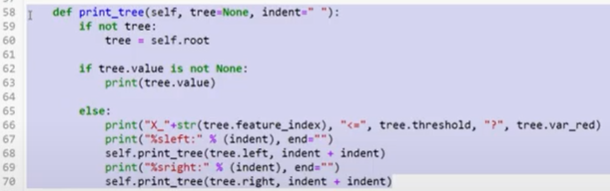


Questa funzione va a calcolare l’impurità di un nodo tramite la formula della riduzione di varianza.









Questa funzione serve a dare una rappresentazione grafica dell’albero di decisione.



Text, letter

Description automatically generated

Funzione di predizione

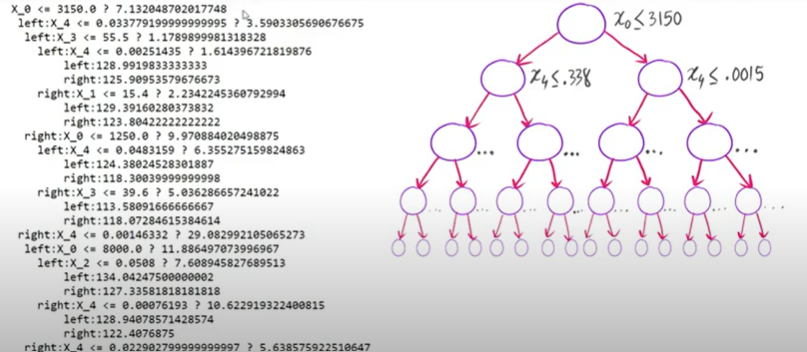


Questa funzione va a completare la precedente sul processo di predizione.

Adesso il modello di albero di decisione per la regressione è pronto e necessità di ricevere un data set in particolare un training set per la fase di addestramento e in seguito seguirà la parte di validazione e test, per farlo usiamo sklearn:

Text, letter

Description automatically generated



Quello appena visto è lo schema dell’albero costruito.

Text, letter

Description automatically generated

Infine testiamo il modello e ne controlliamo l’efficienza con la media dell’errore quadratico.