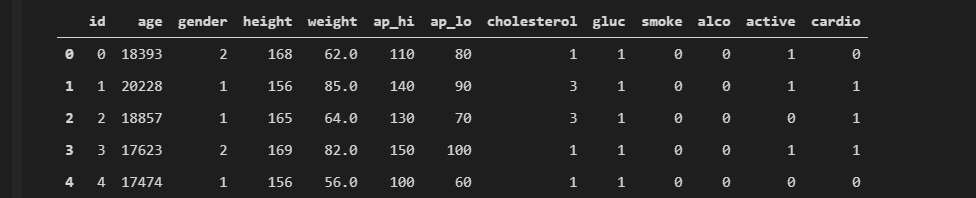
**Decision Tree**

**Problem 2**

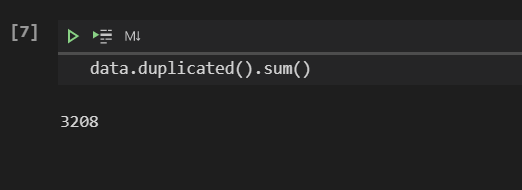
**CART Algorithm**

**Sample For Original data**

****

**I do some processing on data features**

* **The age is given in data so I convert it to years**
* **I don’t use id column so I dropped it**
* **Check on duplicates in data and I found 3208 duplicates**

****

* **Check on outliers That is**

1. **systolic\_bp feature** **>200 or < 80**
2. **diastolic\_bp feature > 180 or <50**

* **Drop outliers**
* **I Thought Height and weight seems uncorrelated with cardio feature But BMI (Body Mass Index) could be more helpful so I replace two columns height and weight with one column called BMI**
* **To make all features take only discrete value I tried to categorize each feature**

1. **I Found For BMI feature**

**BMI between 18.5 and 25 , person Normal**

**if BMI obove 25 , person is obese**

**if BMI less than 18.5 , person is underweight**

**So I categorized it with 0 , 1 , 2**

1. **I Found for systolic blood pressure number**

**Normal: Below 120**

**Elevated: 120-129**

**Stage 1 high blood pressure (also called hypertension): 130-139**

**Stage 2 hypertension: 140 or more**

**Hypertensive crisis: 180 or more**

**So I categorized it with 0 , 1 , 2 ,3 ,4**

1. **I Found for diastolic blood pressure number**

**Normal: Lower than 80**

**Stage 1 hypertension: 80-89**

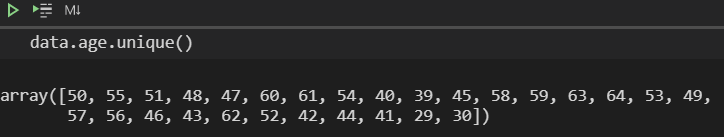
**Stage 2 hypertension: 90 or more\***

**Hypertensive crisis: 120 or more**

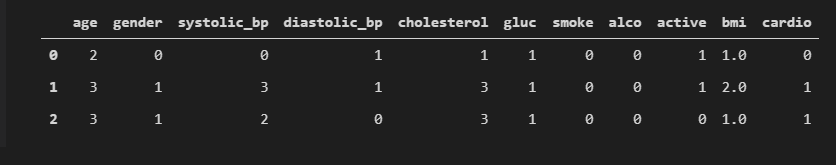
**So I categorized it with 0 , 1 , 2 ,3**

1. **I found most of unique values in age in range 30 to 60**

**So I categorized it to 5 Catagories**



**Final Data**

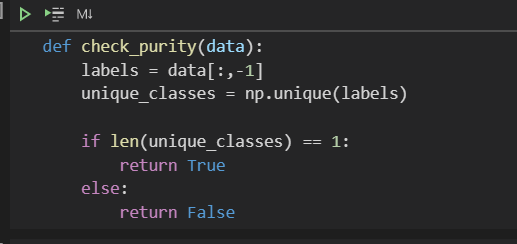
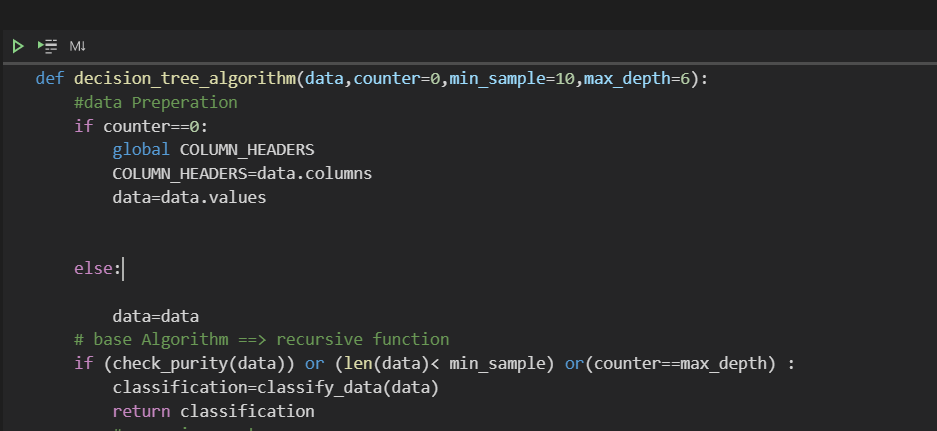
****

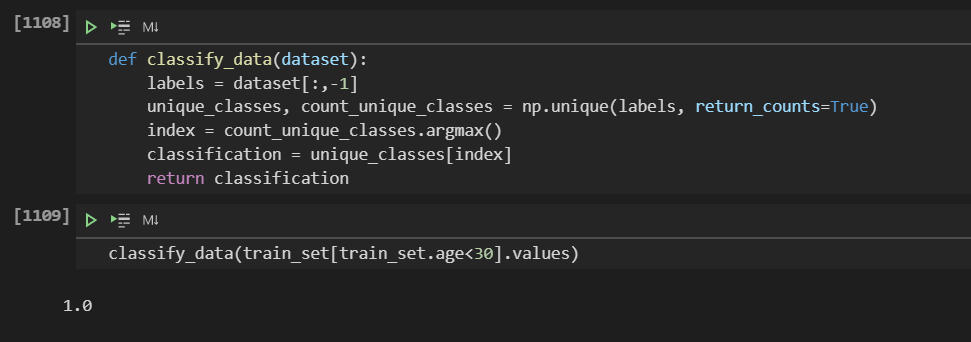
**This is simple Chart That I followed when I was building algorithmDiagram

Description automatically generated**

**Tree Algorithm (Fitting Part)**

1. **Check on Three Criteria To know when Stop**

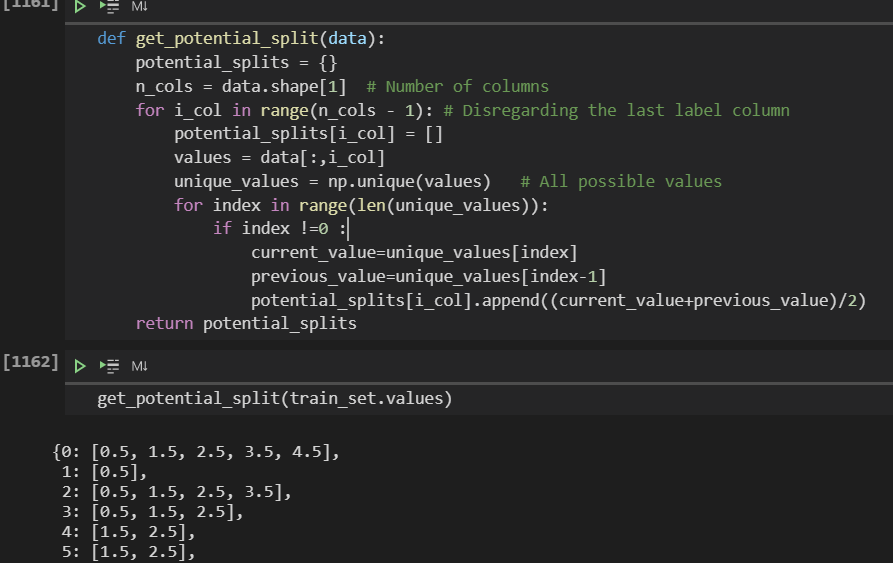
* **Check Purity, if Current node contain only one kind from classes or sample** 
  + ****
* **Check max depth and Number of samples or data that classify that must be at least 10**
* 
* **If condition of any one True, Algorithm will stop and classify**

****

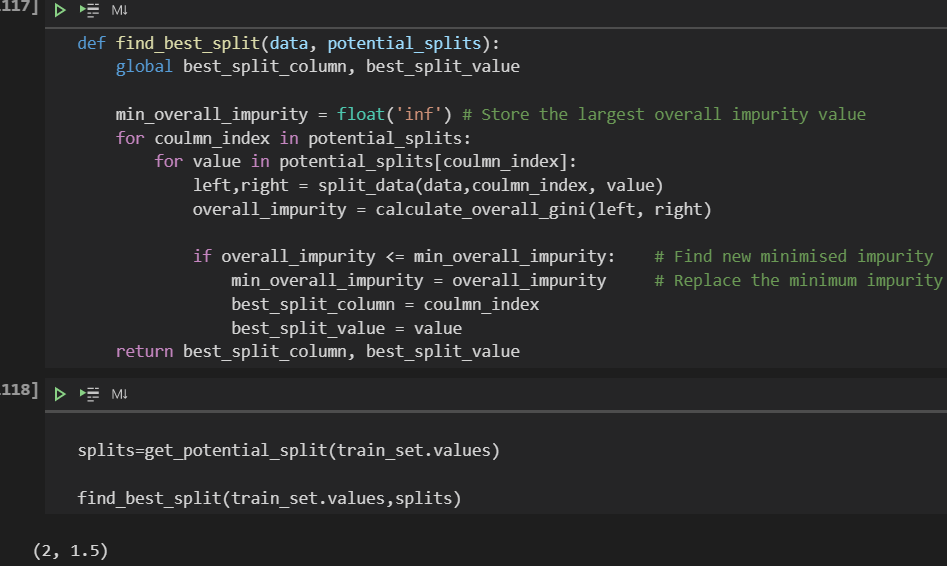
* **This is Classify function that can classify based on majority class**

1. **After Checking on three criteria if condition false, it will follow scenario that I mentioned in picture above**

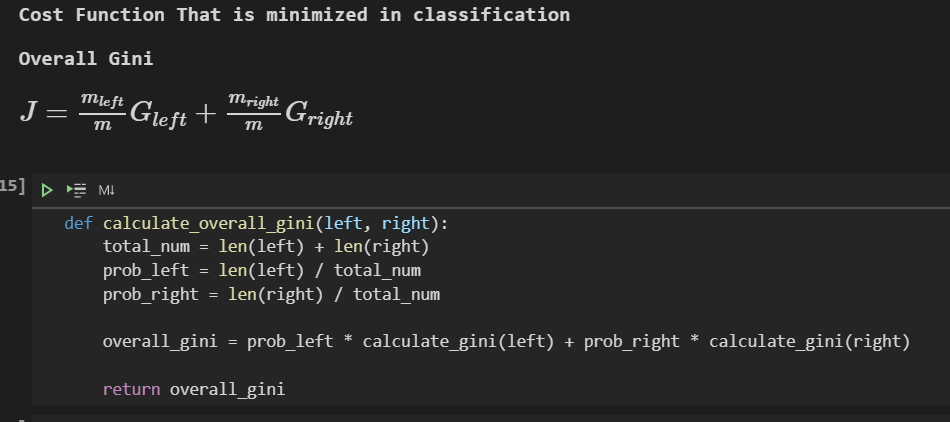
* **Get all possible values for all features of data that can split at**

****

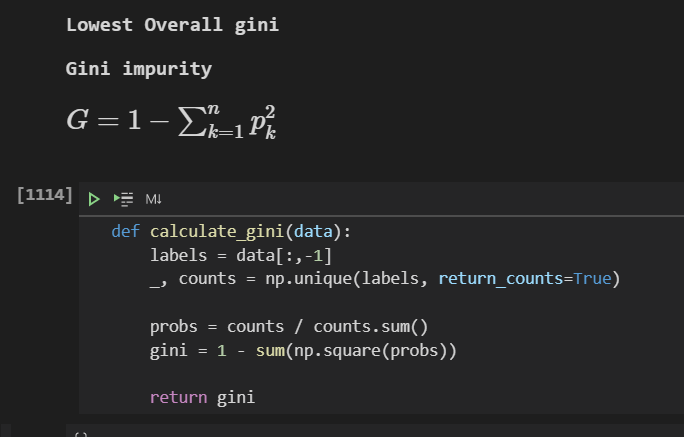
* **This is function that can get all possible splits for all features and its output is dictionary where key is column number and value is list with all possible splits for column**
* **After that calculate over all gini impurity at each value for each feature and based on the lowest gini, algorithm can know the best value at best feature that can split at (Node)**

****

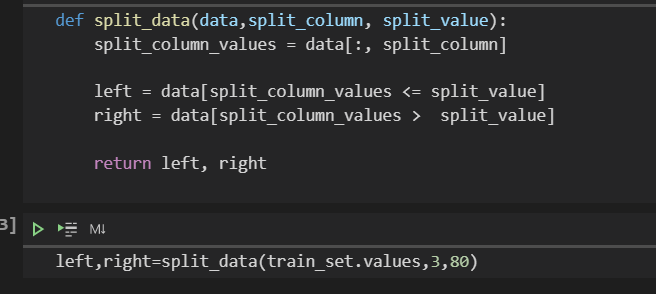
* **This function to get best value and best feature to split based on gini impurity equation where it split at each value and calculate gini then compare if gini at current value less than previous value, it replaced it and repeat that until get best value at best feature**

****

* **This overall gini Function**

****

* **This Gini value**
* **After getting the best feature and best value ,splitting Node to left and right and doing recursion at each part untill stop at certain layer depend on criteria**

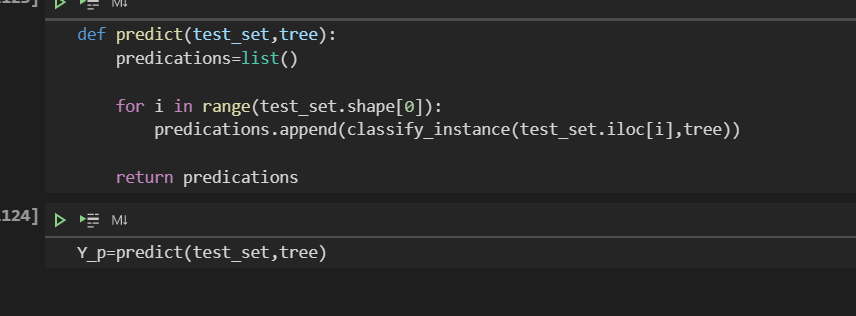
****

* **This function to split Node based on best value that calculated at previous step**

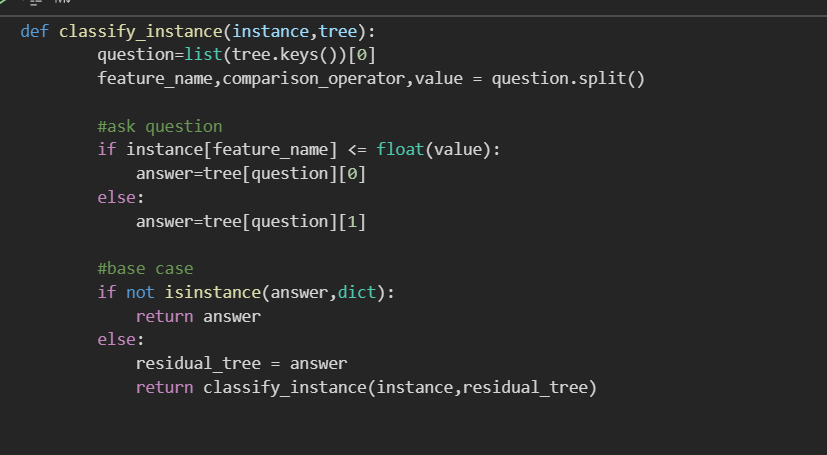
1. **Output of Decision Tree Algorithm is Tree or model**

* **Tree is dictionary where key is question that consist of best feature name and operator <= and best split value for this feature**

**Prediction Part**

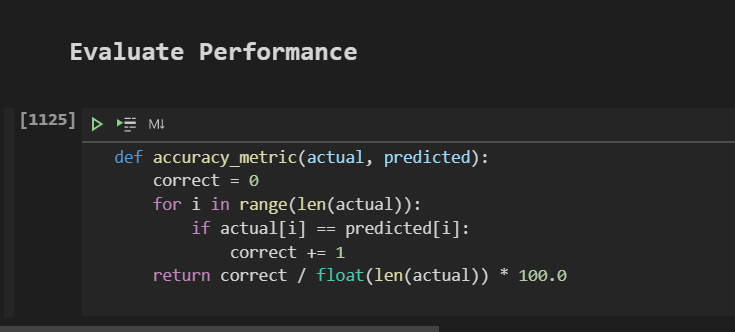
****

1. **Predict function loop on classify\_instace which its input is tree or model that I built and only one instance and can classify it**
2. **Classify\_instance function**

****

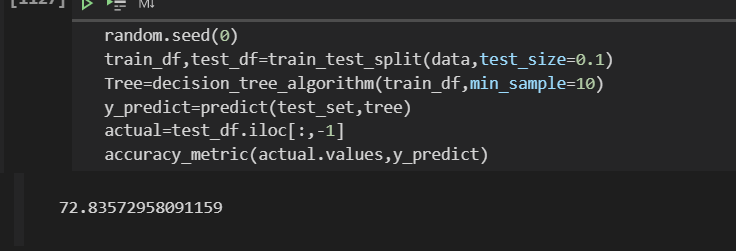
**Based on tree key (question that I formed before when I build tree and consist of feature name and operator and value) I check if value of feature for instance less than best value that I split at, to determine where can I go whether in left tree or right**

**Evaluation Performance**

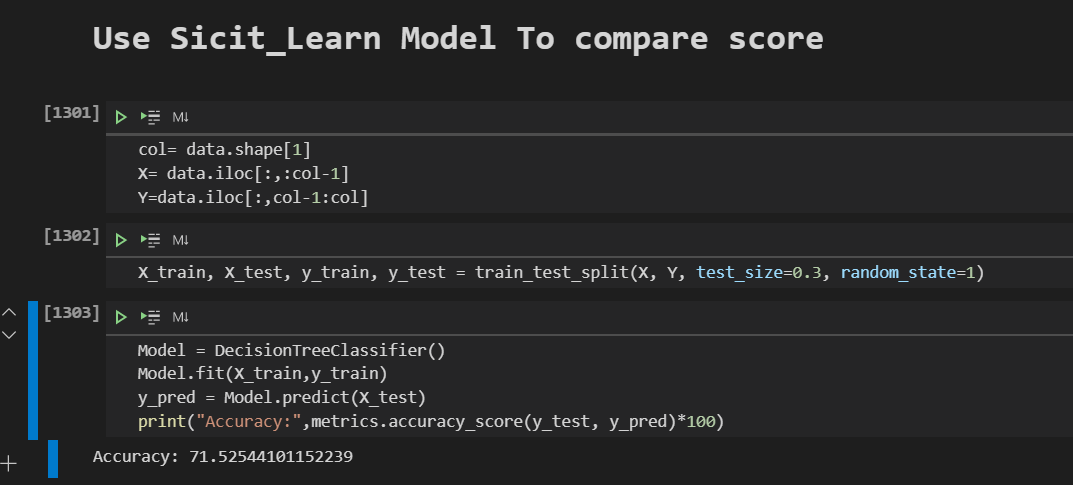


**Accuracy\_metric calculate score of model , by see how many instances can classify correctly**

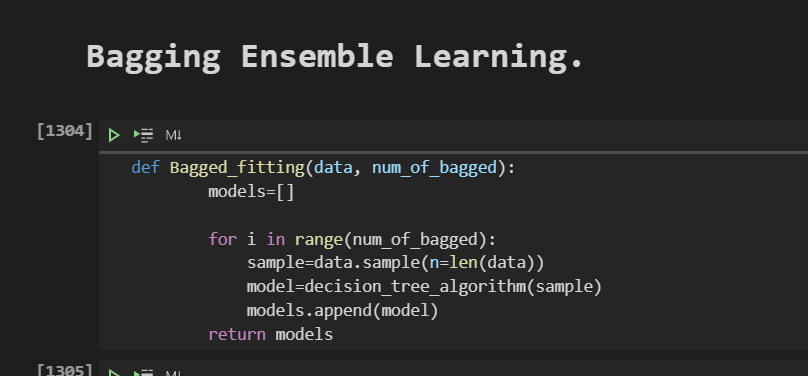
**Result of My Model**

****

**It is similar to score of model of sicit\_learn**

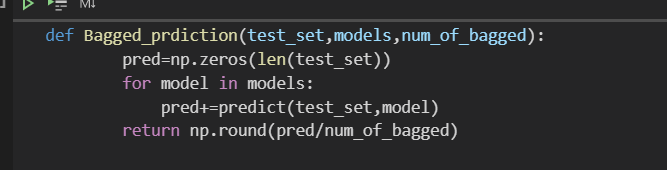
****

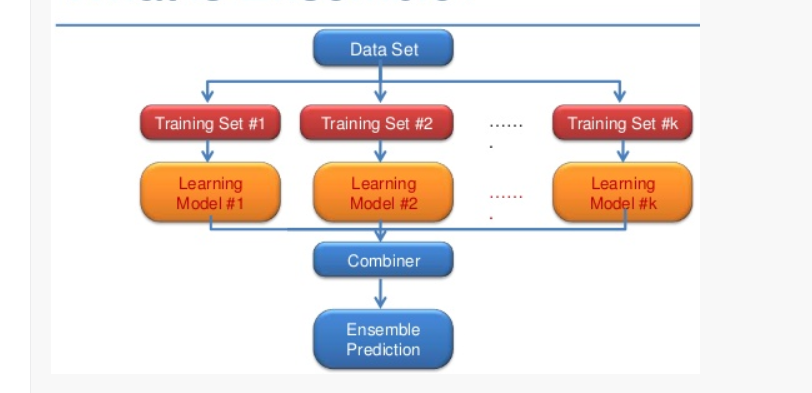
**Bagging Ensemble Learning**

****

1. **IN Bagged\_fitting Function, input is data and number of bagged or models**
2. **I used My Decision Tree Algorithm**
3. **I sample or select from original data randam data with size L= len(data)**
4. **Output of Bagged\_fitting Function is trees or models that I build**

**Bagging Prediction**

****

****

**Based on my understanding for this technique, I Used all models that I built and produced Y\_prediction for each model and added all Y\_prediction list OF all models to make one prediction**

**Bagging Result**

