My Capstone Report – 2

1. My Problem Statement –

My problem statement is that many people suffer from diseases. There are many diseases but in particularly I have chosen heart disease. Sometimes doctors take time to analyze the readings and identify heart diseases and also it requires a lot of time to get the result.

1. My Dataset –

I have chosen my dataset from Kaggle. The link is as follows <https://www.kaggle.com/volodymyrgavrysh/heart-disease>

Age: Age

Sex: Sex (1 = male; 0 = female)

ChestPain: Chest pain (typical, asymptotic, nonanginal, nontypical)

RestBP: Resting blood pressure

Chol: Serum cholestoral in mg/dl

Fbs: Fasting blood sugar > 120 mg/dl (1 = true; 0 = false)

RestECG: Resting electrocardiographic results

MaxHR: Maximum heart rate achieved

ExAng: Exercise induced angina (1 = yes; 0 = no)

Oldpeak: ST depression induced by exercise relative to rest

Slope: Slope of the peak exercise ST segment

Ca: Number of major vessels colored by flourosopy (0 - 3)

Thal: (3 = normal; 6 = fixed defect; 7 = reversable defect)

target: AHD - Diagnosis of heart disease (1 = yes; 0 = no)

For more information please visit this site –

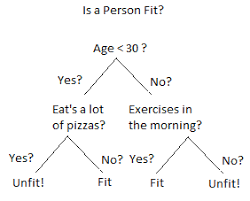
<https://archive.ics.uci.edu/ml/datasets/Heart+Disease>

1. Why require AI to solve this problem ?

Using AI doctors can swiftly identify whether a person has a heart disease or not by giving the machine simple readings like cholesterol , blood pressure etc. After that they can continue with the required medical diagnosis.

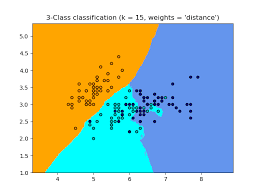
1. Algorithm –

I have done to algorithms. Decision tree classification and K nearest neighbor classification. Decision Tree classification classifies a certain thing based on categories.



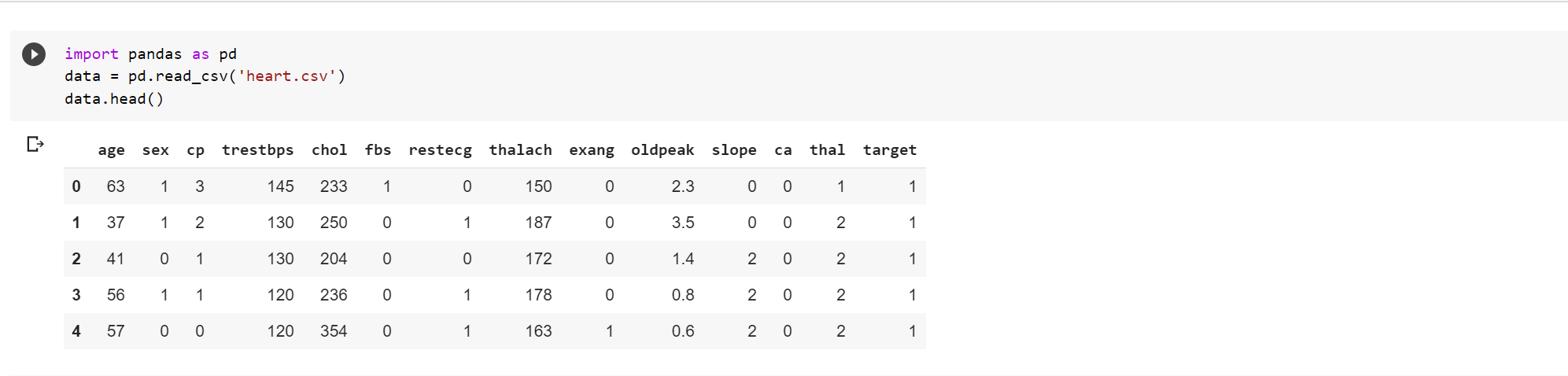
This is an example. The machine classifies the input using similar algorithm like in the picture to arrive at a conclusion. This algorithm is similar to a tree as we can see branches and more branches branching out.

K nearest neighbor classification is classification based on grouping the input based on the nearest similar group.

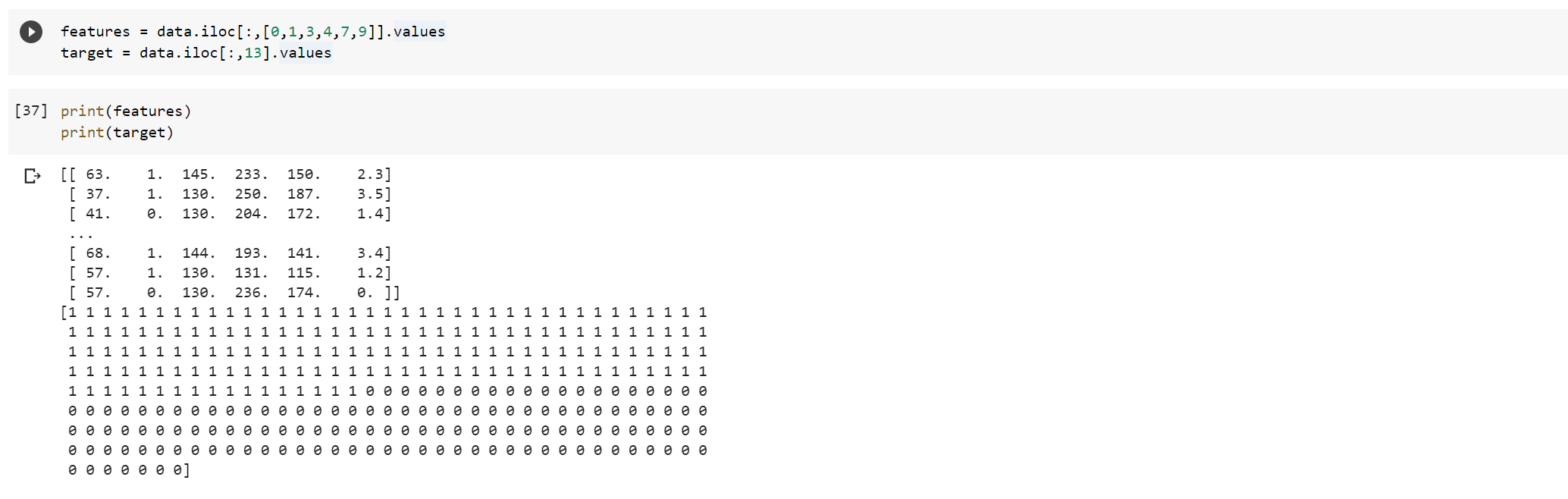


As we can see in the picture the similar dots are grouped in one colored group. This algorithm tries to find similarities in the input data with the data it has and classifies it in one group.

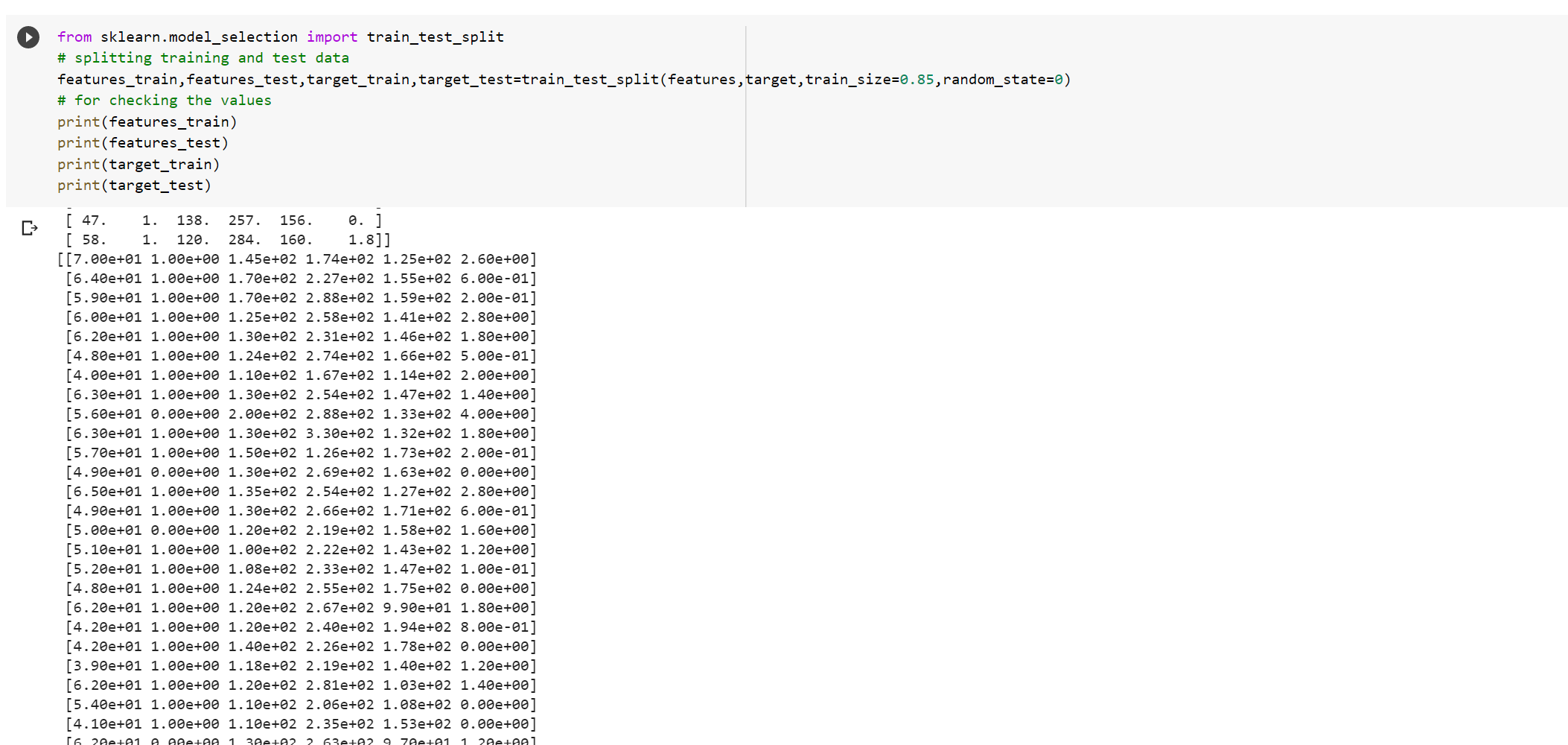
1. Code –



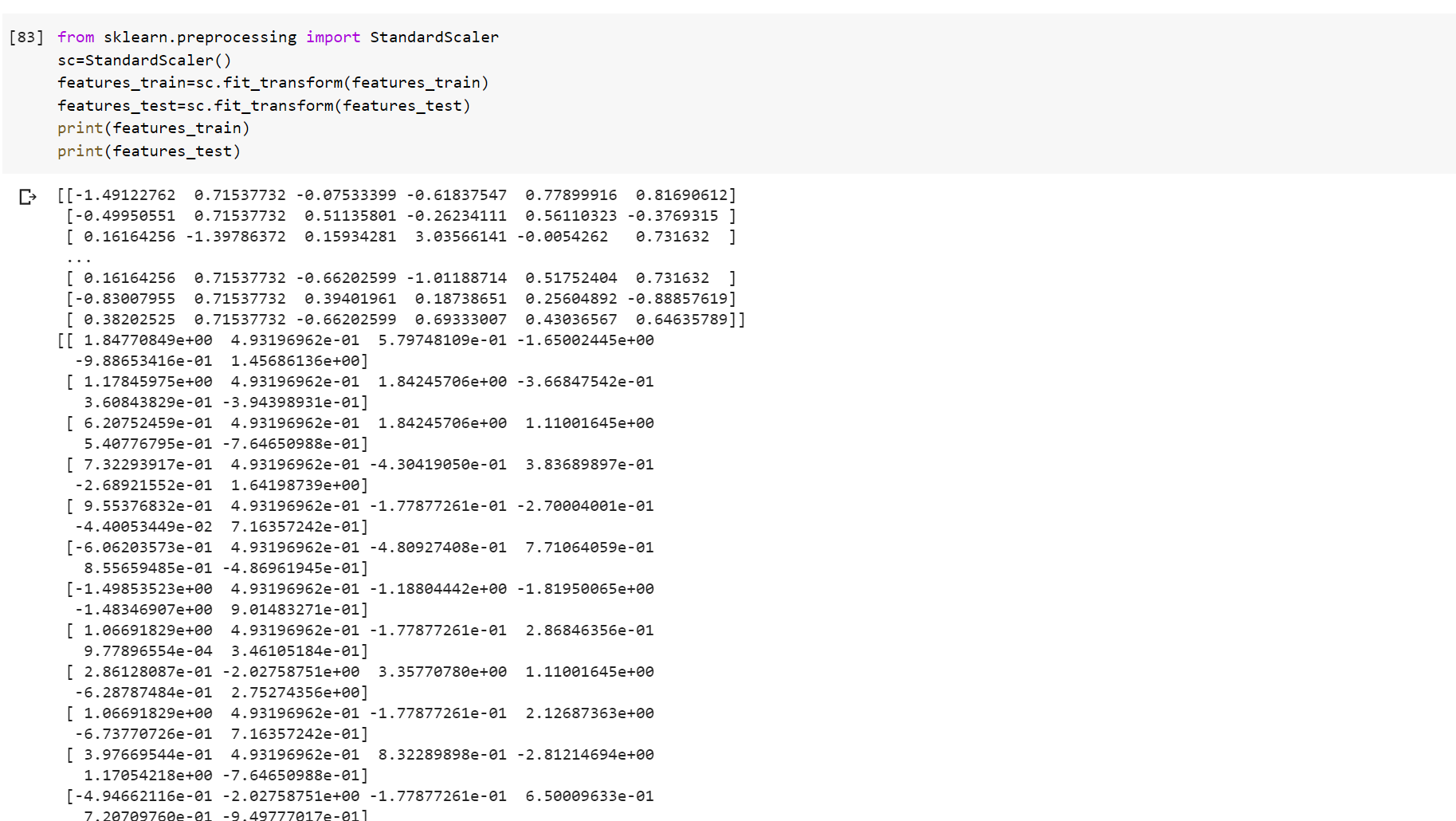
I have imported pandas to open the file in a variable called data. Data.head() helps to display the first few rows of the data.



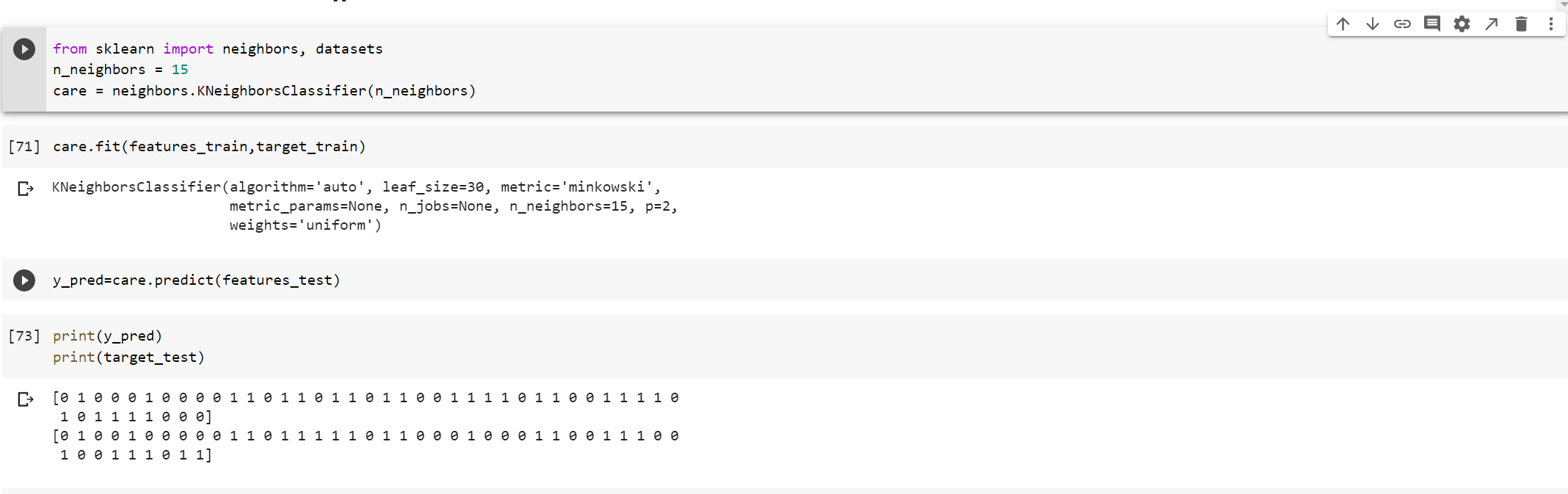
I have created 2 variables named features and target. In the features variable I have given some specific column values to use as my x variable. In the target variable I have added only one column showing the y variable. I have used the .iloc[] function to capture the data I needed.



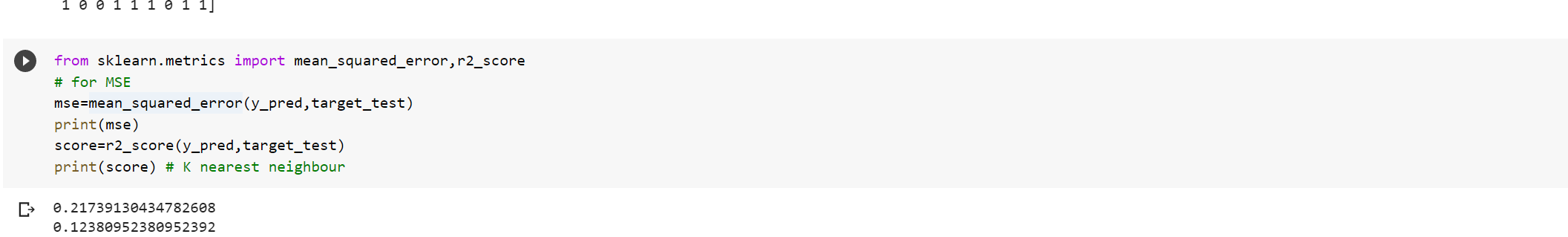
I have imported train test split from scikit learn so that I can split my data into 4 variables namely features\_train , features\_test , target\_train and target\_test. I am going to use features\_train and target\_train to train the model. I am going to use features\_test to test the model. The results will be stored in a variable which I will compare with target\_test.



I have imported standard scaler to transform the values of features\_train and features\_test as they have multiple columns.



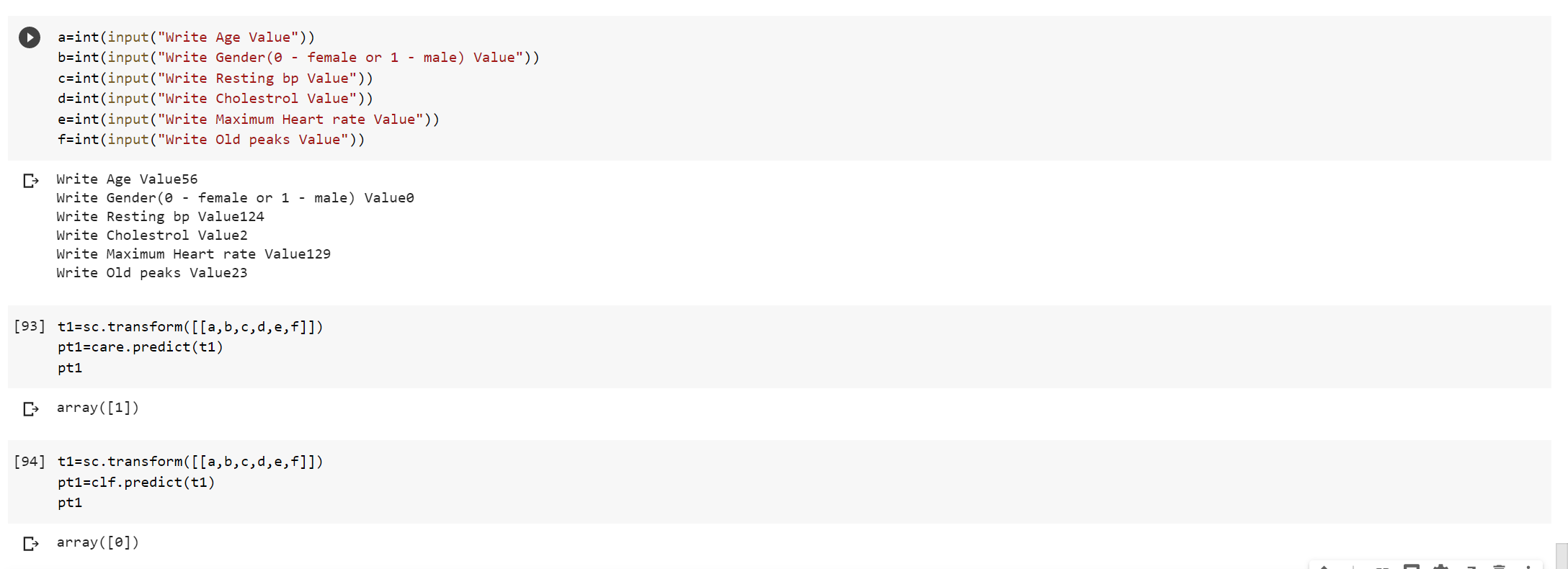
In these three steps I have imported a k nearest neighbor classifier model and set the n\_neighors value to 15. I have declared the name of the model as care which is a k nearest neighbor classifier model. After that I have trained the model with my features\_train and target\_train variables. Then I have predicted the outcome of the features\_test variable which I have assigned to a variable called y\_pred. Then to compare the results I have first printed the machine’s guess and then the correct results.



Now I have imported mean squared error and r2 score to check the accuracy of my KNN classifier model. The output is as shown above in the picture. The first value is for the mean squared error and the second value is for the r2 score.



Now I have used another algorithm which is Decision Tree algorithm for just a comparative analysis. So I have imported a Decision Tree classifier model and gave it a name which is clf. Then I have set the max depth to 5 and the criterion to gini. Then I have trained the model and predicted the outcome for features\_test and compared it with the target\_test variable. Then I have taken the mean squared error and the r2 score.



Now using the user based value I have predicted the output. The first output (93 number code cell) is the output of a KNN classification model and the second output (94 number code cell) is the output of a Decision Tree classification model.

Conclusion –

By comparing the mean squared errors and r2 scores of both models I have concluded that the KNN classifier is more efficient than the Decision Tree classifier in this dataset. The KNN classifier has a good mean squared error and also a good r2 score whereas the decision tree classifier does not.