

AI PROJECT

TORCS



Submitted by: 18i-0453 Faraz Ud din

i181568 Quassain Raza

i170288 Amna Azam

# Introduction:

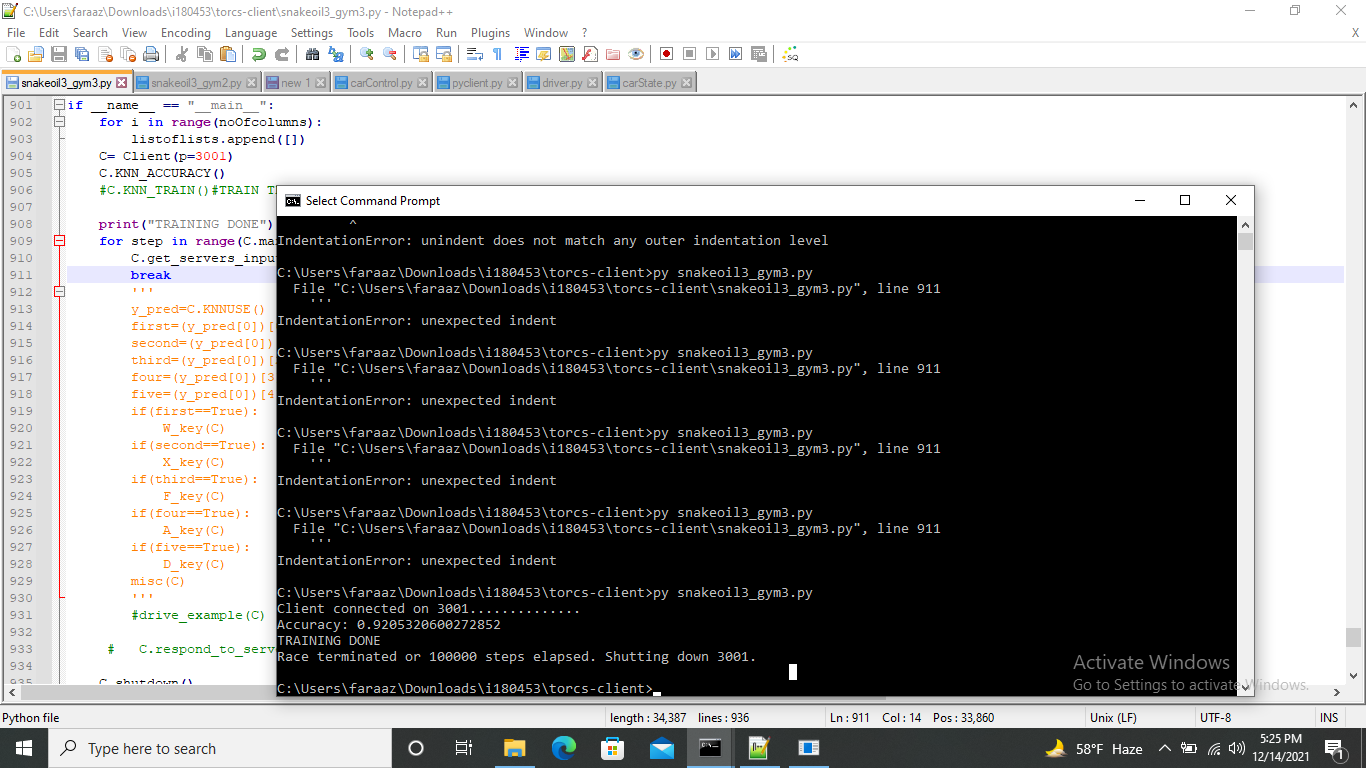
# 

We used KNN classification to make autonomous car. Firstly, we collect the sensors data and how client responds to the server’s input in form of dataset. Then we train our KNN model according to the dataset. One thing to notice here is that the problem was not simpler as it was not a single label classification for one input, we have to predict multiple labels for example if we have to take a turn we have to steer as well as apply brakes. Also, we didn’t go to find the most optimal value of K, we just select K=3 and get labels on the basis of votes of 3 neighbors only.

Also, we didn’t build KNN algorithm from scratch. We used sklearn library to train KNN classifier.

# WHY KNN?

Because clearly it was a classification problem as we have to classify that should we accelerate/apply brakes/take left turn/take right turn for certain sensory input? After going through results of different research papers we found out that KNN performs perfectly fine for TORCS. As well as KNN can also perform multi-label classification so we opt to this option. Also, after training the model, we did test the accuracy of our model and got almost 92% accuracy.



# How KNN Works?

KNN compares the distance of current test set with all points in training set then on the basis of value of K. It does the voting between k closest neighbors and what ever is the result of voting we consider that it should be the output for current test set as well

# KNN with Torcs:

We have some 19 features (sensor values) and 5 labels (Speed, Brake, Boost, Left turn, right turn)

Sensory sends sensory details to client. client compares the sensory details with all dataset it has (we call it training set). Then it finds 3 closest neighbors and then that 3 closest neighbors do voting for speed that should we accelerate or not whatever is the decision of voting we allot that value to that particular label. Then we again do voting for Brake, then for Boost and so on. And in the end, we predict all the label values through this process