

**Victor Morgan**

**10/12/25**

**CSC-570**

**Dr. Osunmakinde**

### Conversion to my AI field of choice

For my KNN regressor, I chose to create a program that after being trained on a set of data provides users the option and ability to predict the value of a car. This follows my theme of finance, and would allow users to decide how much they should sell their car for, or if the offer they're being given on their car is fair.

The PEAS are as follows:

**Performance** : The accuracy of the program after being trained and potentially hypertuned.

**Environment** : The car sales dataset which provides various cars' attributes (predictors), and their corresponding prices (target)

**Actuators** : Predicting the value of new cars based on their features after receiving them from the user.

**Sensors** : The csv file used to train and test the model, as well as the input space for users.

### User-Friendly Aspect

For this assignment I focused on the user-friendly aspect of how my AI should function. Not only did this make my program more flexible, it also makes it more reliable and accessible to users who might not be as experienced with systems that require specific input formatting.

When it comes to categorical data, the user is required to enter parameters that correspond to previously existing predictor values. My program uses a function I made, **cat\_input** to dynamically create a list of acceptable values based on which predictor the user is providing information for. It requests the user to input the data on the car. If the user enters a value that is not recognized (appears in the dataset) it provides feedback accompanied by examples of acceptable input to the user before allowing the user to try again. The user will try until they enter a correct(recognized) value.

In addition to **cat\_input**, I made a function called **collect\_int**, that ensures all of the requests for quantitative predictors from the user are met with actual numbers as input. This assists in maintaining the model's integrity as it requests input.

### Conclusion

While I did also modify the data visualization, my main focus was improving the resilience of my program. I wanted to ensure that my program would not fail, no matter what the typical user entered. I've incorporated strong input validation, as well as fleshed out feedback to reduce user

confusion should any be caused. I did contemplate adding the newly generated predictions to the model's dataset. I'm wary as I believe that would allow a user to dismantle the integrity of the dataset and in turn the predictive strength of the model.

For future work I'd like to make it so that my program is able to accept new unique variables, but as of now I haven't figured out how they would be weighed. This would require a more in-depth insight on how the creator of the dataset valued and weighed each parameter. This would make a program that's able to adapt to changing times and overall make it a more optimal choice for real-world application.