Assignment 3 Documentation: Machine Learning

For this assignment, we were tasked with creating a python program that would take details of aspects found in diamonds and determine a price for them through the use of regression with varying degrees and compare these predictions to see which is the most accurate version.

Task 1

Before we attempt the prediction, we need to set up numeric values for each of the grades for cut, colour and clarity. After this we get combinations of these 3 variables extracted from the and count each of the combinations present within the CSV file. We achieve this by getting our numeric variables ad placing them in an array that loops around and adds these combinations to the array. We then split these combinations into targets and features to be used later on in the programs. We were asked to only have datasets of combinations that exceeded 800 occurrences, so we loop around again with this in mind

Task 2

This task involves the use of a polynomial function which is to model the relationship between our variables and will takes a degree that will be used with the data in task 1. We pass in the data we had for task 1 as a NumPy array ad begin reiterating the data for the combination at each degree which will increase for each loop like so:

```
for n in range(deg+1):
    for i in range(n+1):
        for j in range(n+1):
            for k in range(n+1):
                if i + j + k == n:
```

Once it passes through, we then return our results to be used later for task 5.

Task 3

The task of linearization takes in the model we developed from the previous task and recalculate it through a linearization point p0. Once we have it we divide it by our Jacobin matrix. This must take in the results for each of the degrees in the last task and the features from the first task. Once it is

completed, we return the values form the previous task that have gone through the linearization as well as a NumPy of the length of the results from the second task and our p0 linearization point.

Task 4

This task involves the use of creating a function that takes in or training target data from the first task and estimating the vector and Jacobian that will be calculated in task 3. We do this through the use if an equation matrix gotten from the np.matmul which will take in 2 arrays of target data we then solve each of these matrixes through the linalg.solve ad return the result to be used for the next task

Task 5

We now use our return values form the past three function to be used within the regression function. Within this function, is where we run each of the three functions for 5 iteration and 0 -4 degrees. Once they all go through, we then begin getting our combinations and shape each of them for the different combinations of our target data recorded before. We then plot the data to a 3D graph; we can see the prediction of the continues target data values being displayed with points that are being used for a best-fit line display.

Task 6

The k-fold cross validation is then preformed, also using the data extracted from task 1. We split the data and iterate loop around 5 times. We display the results in a similar fashion to the pervious task with the 3D graphs and the best-fit line. I was unable to properly get the comparison for the degrees 0-3 and could not get the quality of each of the models or the optimal degree for each of the data sets.

Task 7

Due to this task being tied to the end of the previous task, I could not get the estimate parameters or the predicted price to add to the graph display.