MAIS202 – Project Deliverable 1: Data Selection Proposal

1) The dataset used comes from Kaggle and more specifically is a dataset that contains attributes of almost 54,000 diamonds (such as their carat, cut, colour, clarity and depth) as well as all their prices. (Link: https://www.kaggle.com/shivam2503/diamonds)
   The general idea of the project is to basically predict the price of a diamond given its attributes, so it would be beneficial to have a dataset containing as many points as possible to get the most accurate model. With a dataset containing almost 54,000 entries, I have more than enough data to separate them in three for each stage of the ML pipeline (training, validation and testing). At the minimum, there would be almost 18,000 entries in the training stage, which is quite a lot of data. Furthermore, every diamond in the dataset contains all of its attributes and their corresponding values to make a model calculating its price, which confirms that the dataset is complete and good to use for the project.

2) A) The dataset chosen is very feasible because each of the attributes are separated by columns which can be easily accessed using the Pandas library (the data is all contained within a single .csv file). Every single column will be useful to find the ideal model, though if necessary, I might exclude the diamond's length (x), width (y) and depth (z) since the depth percentage was already calculated for us and uses these three measurements. As stated, I'll be using Pandas and their dataframe methods to extract the information and preprocessing will mostly consist of checking for formatting differences (especially for columns containing strings as values) and primitive data types (some values are stated as floats and others as integers in the same column). Then, for string values, I'll have to convert them in some numerical value to be able to make the algorithm.

   B) As previously stated in the first point, we'd like to predict the price of a diamond given some attributes put in by a client. I will be using two different algorithms to give as precise of a price to the diamond as possible. The first one is the K-nearest neighbours (KNN) classification method which will be great to classify diamonds together by the attributes that contribute the most in their price (such as their carat and cut, for instance). Since it is a non-parametric algorithm, it does not use any assumptions for its implementation and thus it doesn't have a "training step" either, making KNN a relatively easy algorithm to implement.

However, KNN struggles to predict outputs of new data entries as the number of input variables grows (known as the Curse of Dimensionality). The rest of the attributes will be taken care of by our second algorithm: (multivariate) linear regression and gradient descent! This algorithm will be useful for our multiple attributes / input variables and can be even more precise when it's used after KNN. It can be regularized to avoid overfitting and can be easily updated using gradient descent, making it a nice algorithm for our situation.

C) For KNN, accuracy loss is a simple and easily understood evaluation metric. We are only classifying the diamonds by one or two attributes to simplify and refine our (multivariate) linear regression algorithm, so the metric won't be evaluated too deeply upon. However, for the second algorithm, the mean squared error will be a metric that has to be respected. I'm hoping for an error smaller than $1, but expectations can be shifted depending on time-consumption in model training and baseline results.

3) The project will be showcased using a webapp that has its users input either i) the desired attributes of a diamond to find out its price or ii) a price range they can afford to find a list of diamonds of similar prices with a range of their respective attributes. The webapp I'll be using is Flask, as someone who has never worked with webapps before. It also uses Python and will be easy to integrate my models with it.