Let’s say you are given a large amount of textual data- messages, emails, books, etc. Before performing any operations on this data, it is necessary to clean and preprocess the data (removing unnecessary words or symbols, etc.). Explain how you would go about preprocessing. What different steps would be followed? Why are they necessary?

Data pre-processing means cleaning or cleansing of data. It is necessary for the data to be cleaned before making any model on it or analysing it. To go about pre-processing the data, we will first import the required libraries for example, pandas, numpy, train_test_split, re, stopwords, ntlk, tensorflow. These are the basic libraries to process any data. Then we will download some nltk modules like punkt, stopwords, wordnet, omw-1.4. For a smoother and faster output we will ensure the GPU is used and internet is turned on in our Kaggle workspace. Then we will import our data. Using the re library, we can sort/filter the string for a specific pattern. Suppose we want only the alphabet characters in our data, then we will use the sub function from re module. Re is a regular expression. The sub() takes 3 parameters, first it takes the pattern for which we want to search, second that if any character other than alphabets is found it will be replaced by this, thirdly the data in string form. We can optimize our work more by lowering all the alphabet characters. The next most important step in data pre-processing is Tokenizer. From tensorflow’s library we can use tokenizer to convert our data text into matrix form. We then tokenize our data according to sentences using nltk.sent_tokenizer(). We now fit our text onto the tokens. Further we can access the words from the text using word_index.keys() and store it in a list. If we print this list we can see the tokens. We then have a task to remove the stopwords from our text. Stopwords are basically those words which do not have any relevance or significance in our sentence. They are usually used to express the sentence grammatically correct. But for data analysis they are not required. set() function gives a list of common stop words of whatever language we specify. We then run a loop for our tokens, and if we find that any of these tokens are not matching with the stopwords then we append it in a new list. This filtered list now is called corpus. Corpus contains only words that are required for the model to understand what are we saying. Then our next step is to do lemmatization or stemming. These two are techniques to break down words into more simpler tokens. Lemmatization shortens word into such a thing that its meaning is preserved, whereas stemming breaks word in such a way that it reduces to most possible smaller set of characters. We can also stem by using the PorterStemmer library. We can adjust the use of lemmatization or stemming according to our needs. Normalization is used to convert the data in the scale of a certain range, it can be -1 to 1 or 0 to 1. Now the data which we have finally is cleaned at its best and is suited for further data analysis.

Imagine using a random prompt like "a cat riding a bicycle on Mars" and seeing an AI generate an image that matches your description perfectly. This is made possible by using advanced models like DALL-E, which use various machine learning techniques, including the diffusion processes. For this task, explain what basic diffusion is, how it works, and why it is used in generating such impressive output.

DALL-E are text-to-image models, it has been updated many a times. It is developed by OpenAI. It generates images using various deep learning algorithms using our prompts. One of them is Diffusion models. DALL-E is a neural network and works on transformer model. Diffusion model was based on the concepts of non-equilibrium thermodynamics. Diffusion is the process of taking a real image and adding noises to it such that it fully reduce to noise. Then after that, reversing that noisy image again to produce that real image. The process of adding noise is called the Markov chain. It is a chain of events where the current time step only depends on the previous time stage. This makes sure that there are no cross-dependencies. This can involve small 1000 steps too. We train the model using this technique to convert an image to noise and convert back it to a high resolution image. When the model has created the noisy image, to convert it again to real image, we input that noisy image to convolutional neural network (CNN).