**Part 1**

The Class\_6\_MultiModalLLMs.ipynb notebook implements CLIP based on a model in Hugging Face and allows you to try it out. The code loads Pillow which is a friendly fork of the Python Image Library. It then gets a picture of a puppy playing in the snow and sets the caption. The CLIP tokenizer, processor, and model are loaded. Note that “Return\_tensors = “pt”” is for PyTorch Tensors. Then the text caption and the image are processed. It calculates the probability of the text belonging to the image which results in 1.0 – I think that this is 100% but it may be 1%. Then the similarity between the Text and the Image embeddings is determined to be 0.295. This seems low.

Next it uses the SentenceTransformers. This is based on BERT and can work on Text and Images. The code loads a picture of a person taking a picture with a big camera. It sets the text to "Her vocalization was remarkably melodic" which does not seem to be related to the picture. Loads the dog image again and generates text from it. Loads a new picture of a Rorschach Test. Generates the text from the picture. Lastly, it sets things up so you can ask questions about the image.

**Part 2**

Reviewing Class\_6\_TransformerApps\_PyTorch.ipynb I see several steps. It is not clear to me that they are all required by Hugging Face. But assuming that they are, it seems like overkill. Why would I Need to do a translation if I am only working in one language? Other than that, it seems straightforward. It is nice to be able to load a model from Hugging Face into Colab just by name.

The steps as I saw them. The first step is to classify the text. It is classified as negative. The next step is to identify named entities (NER). It seems to get most but there is some overlap. It does not identify entities like the records (Does it only classify entities with proper names?). The next step answers the question: What does the customer want? Next is summarization of the letter. Next the letter is translated to German. Lastly a customer service response is generated.