1. **According to our research, a machine learning algorithm has an accuracy of 80%. However, when we applied the algorithm to practical use, the accuracy was below the expected accuracy. Describe two or more causes to this.**

Answer:

The causes of the problem might be due to:

A. Overfitting: It is caused because of following reasons:
   a. Not stopping the learning at sweet spot
   b. Model is too complex
   c. Size of dataset is too small

B. Biased training dataset: For example, we have to train the model for facial expression recognition. But while making the dataset, we only take samples from the white skin color people. In that case, the model can show high accuracy, but in actuality it cannot show the expression of other skin color people.

2. **Please suggest and describe two industry applicable Deep Learning projects you would carry out using a dataset of aerial images.**

Answer:

Two industry applicable deep learning projects due to aerial images are:

A. Traffic surveillance and routing:

   With the help of aerial images, the volume of the vehicle can be found on the roads and, from the same aerial view, the alternative route can be suggested with less traffic.

B. Surveillance of land covered:

   Here we can detect vital information for urban planning, environmental monitoring and geographical change detection with the help of aerial images.

3. **You have a large data set consisting of high-resolution aerial orthophotos. Your objective is to create an API that detects small objects within an orthophoto (e.g. Trees, Cars, People, etc). Please explain how you would create a Deep Learning Pipeline by elaborating on how you would approach the following steps. (No more than 300 words in total)**

   a. **Data Preprocessing/Labeling:**

      I would use the labeling tool, LabelImg, to label the images by selecting the create RectBox option and annotating the segment containing the required objects.

      I will smooth the images down to a lower resolution, which might lead to faster training and higher accuracy. Also, it removes the noise because of the low lighting environment. But the image must not be much lower resolution as we have to detect small objects.

   b. **Model Selection:**
      Deep learning models approaches such as R-CNN, Fast R-CNN, Faster R-CNN, Mask R- CNN, YOLO, RetinaNet, SSD, U-Net are popular for object detection. I will test a bunch of them for a model to select the best of them as our model.

c. Model Training:

Transfer learning can be used for the creation of custom detection models. For this, pre-trained .weights file can be used. After this, the pre-trained model can be trained in for the required number of iterations with our custom dataset to get a custom weights model which can detect objects in the input image.

d. Model Optimization/Hyperparameter Tuning:

For the model optimization and hyperparameter tuning, we have to evaluate the model quality continuously. I would monitor the metrics of the model, change and tune the value of a hyperparameter based on metrics and select the value for which the model gives the better performance. Now, I would save the model with the best result for the deployment proposed.

e. Model Hosting/Deployment/Management:

I would use the flask micro web framework for the model deployment. Web hosting services like AWS can provide us instances for the deployment of the model on the web. I would convert the model into a lite version of the model before deploying it to the low end devices. We can constantly update the models as required.