**Preprocessing steps (I can do this; just including for your reference)**

* Run the kfold.py file to split your testing and training files into 5 folds
* Run the reduce.py file to squash the data into the predefined number of “principal components” (n\_components = X) – I usually start with 1/3 of the original dimensions then scan around that
* Add ID column and headers (id, var1, var2, … , class)

**Vertex AI pipeline steps**

* Create training dataset
  + Make sure you have an ID column and header row in both training and testing data
* Train model
  + Wait until dataset is uploaded and click “Train new model” on the right
* Batch predictions
  + Click “create”
  + Upload your testing set to a Google Cloud location
  + Choose your model and testing set and start the prediction
* Pull results
  + Download gsutil command line tool
  + Click on your batch prediction and go to the export location
  + Select all of the prediction\_results files (usually about 20-30 files)
  + Click “download” and copy the script
  + Open your terminal and navigate to the directory that you wish to download the files into
  + Paste the script that you copied
  + Put all of these files in your Pycharm project and run the joinRows.py file
    - I’ve just been hardcoding the file names so I apologize for the ugly code
  + Open file and remove all of the NANs from the bottom of the file and spot check that the IDs are in ascending order
  + Open one of the original prediction\_results files and check the order of the columns (they shuffle them too – quite annoying…)

A picture containing text, newspaper

Description automatically generated

* + Put them in correct order (ascending classes from left to right)
* Grade results
  + Run the judge.py file
  + Record the average accuracy of the five folds in the Google sheet