MNIST-Digit-Classification

## A. Creating a Neural Network

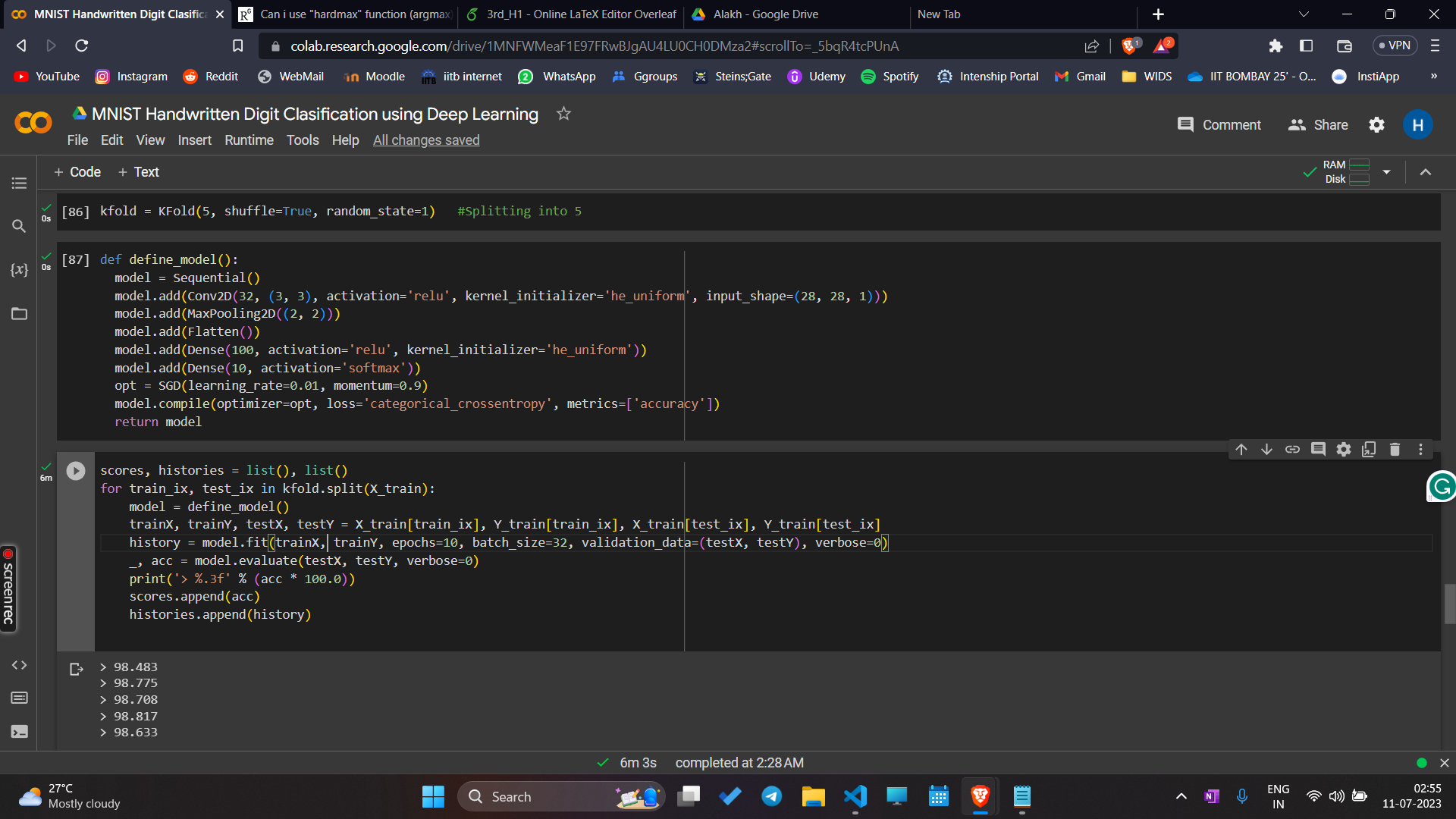
1. First we define out model
   1. model = Sequential()
   2. model.add(Conv2D(32, (3, 3), activation='relu',kernel\_initializer='he\_uniform', input\_shape=(28, 28, 1)))
   3. model.add(MaxPooling2D((2, 2)))
   4. model.add(Flatten())
   5. model.add(Dense(100, activation='relu', kernel\_initializer='he\_uniform'))
   6. model.add(Dense(10, activation='softmax'))

**OR**

* 1. model = keras.Sequential([
  2. keras.layers.Flatten(input\_shape=(28,28)),
  3. keras.layers.Dense(50,activation='relu'),
  4. keras.layers.Dense(50,activation='relu'),
  5. keras.layers.Dense(10,activation='softmax')
  6. ])

1. Then we compile the model
   1. model.compile(optimizer='adam',
   2. loss = 'sparse\_categorical\_crossentropy',
   3. metrics=['accuracy'])
2. Fit the data
   1. model.fit(X\_train,Y\_train,epochs=10)
3. Evaluate the model
   1. loss, accuracy = model.evaluate(X\_test, Y\_test)
   2. print(accuracy)

## B. Using [sklearn.model\_selection](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.model_selection).KFold



K-Folds cross-validation

Provides train/test indices to split data in train/test sets. Split the dataset into k consecutive folds (without shuffling by default).

Each fold is then used once as a validation while the k - 1 remaining folds form the training set.