**PROJECT-EXAM 2 REPORT**

**Introduction**

This Project Exam consists of 5 programs that covers working with Keras, Tensorflow to display Loss and Accuracy, Parameters like Optimizer, learning rate, batch size, activation function, Deep Learning concepts such as Word Embedding, Image Classification with CNN, text generation with LSTM network, Autoencoders etc.

**Objective**

The objective of this project is to gain basic understanding of Deep Learning concepts such as CNN, RNN, LSTM, Autoencoders and apply this knowledge so as to get optimal solution for each program that requires to do the following:

1. Implementing text classification on the review’s sentiment dataset using CNN model.
2. Including Embedding layer in the design of the models and report if that leads to a better performance
3. Plotting loss of the model and report if any overfitting problem.
4. What techniques can be applied to fix overfitting model.
5. Implementing text classification on the 20news\_group dataset using LSTM model.

a. Including Embedding layer in the design of your models and report if that leads to

a better performance.

b. Plotting loss of the model and report if you see any overfitting problem

1. Implementing Image Classification with CNN. Report this with and without scaling. Saving the model and predicting test data. Report the prediction and check if it a correct prediction or not.
2. Training a text generation language model which can be used to generate News Headlines when a sample headline or word is passed.
3. Finding the accuracy and loss in two models and then plotting a graph and make comparisons. and find out which has better accuracy.

**Workflow**

The workflow is common across all the programs:

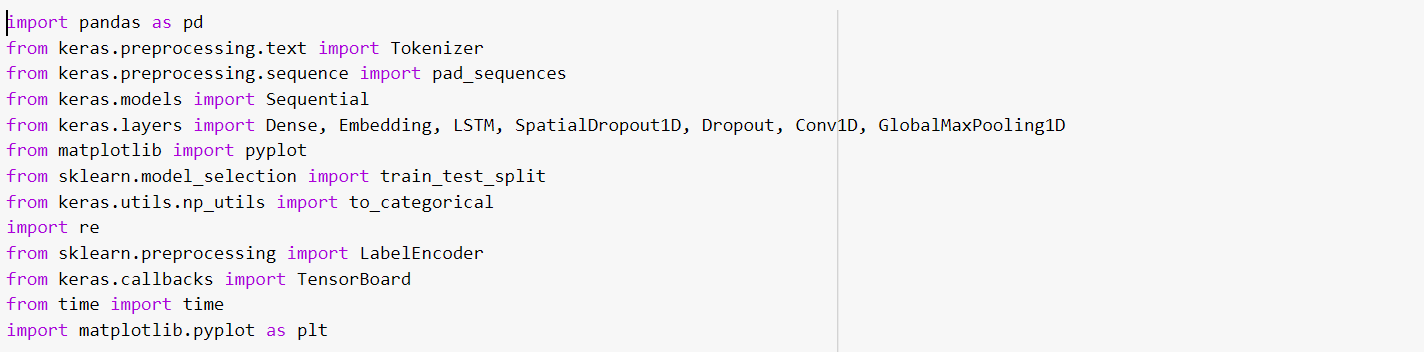
* Importing the required Libraries
* Reading the Dataset
* Data Pre-processing
* Model Creation
* Model Execution
* Model Evaluation

**Approaches**

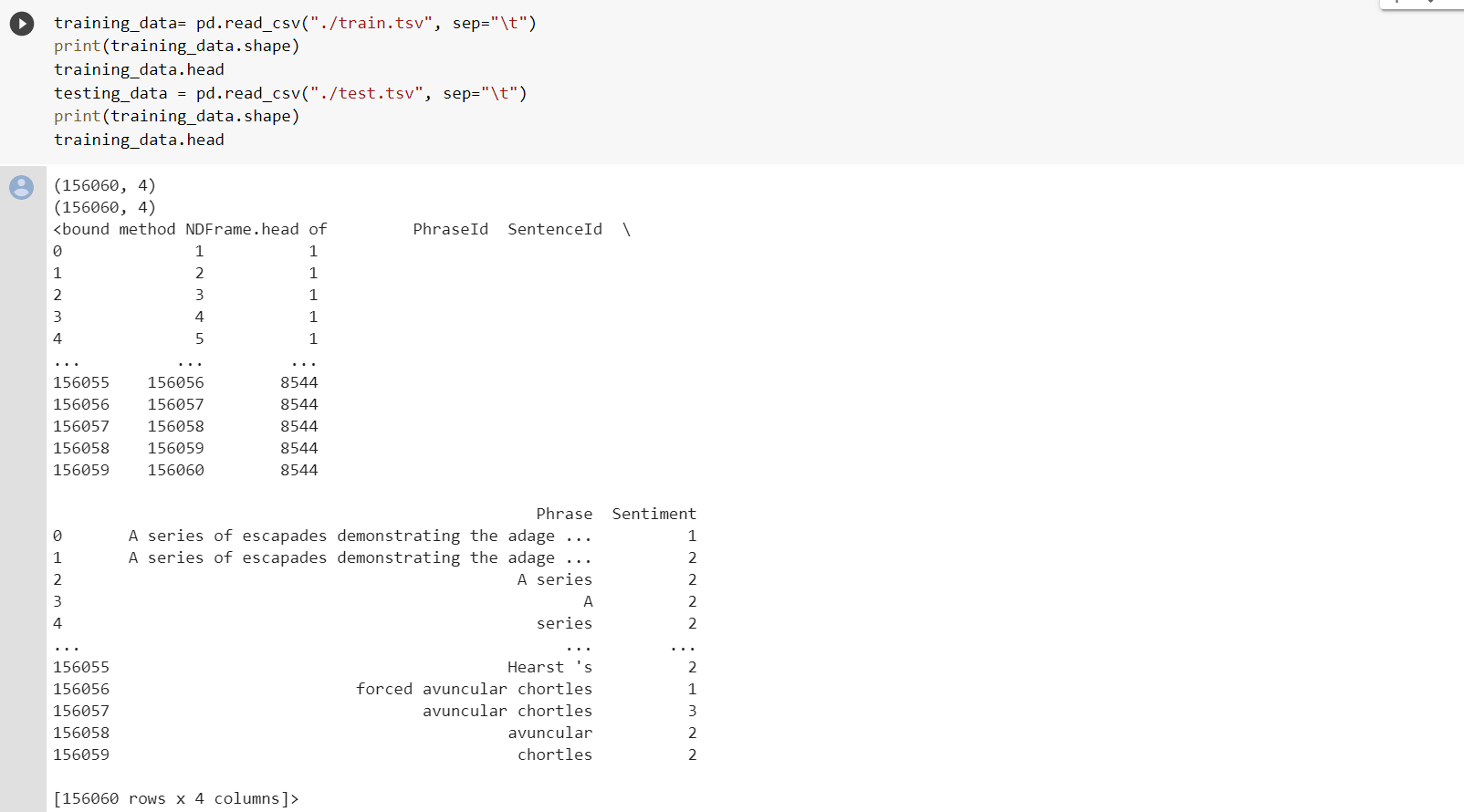
**Task-1:**

**Implementation:**

* Importing all the required libraries.



* Reading training and testing data.
* Displaying the content in train and test data.



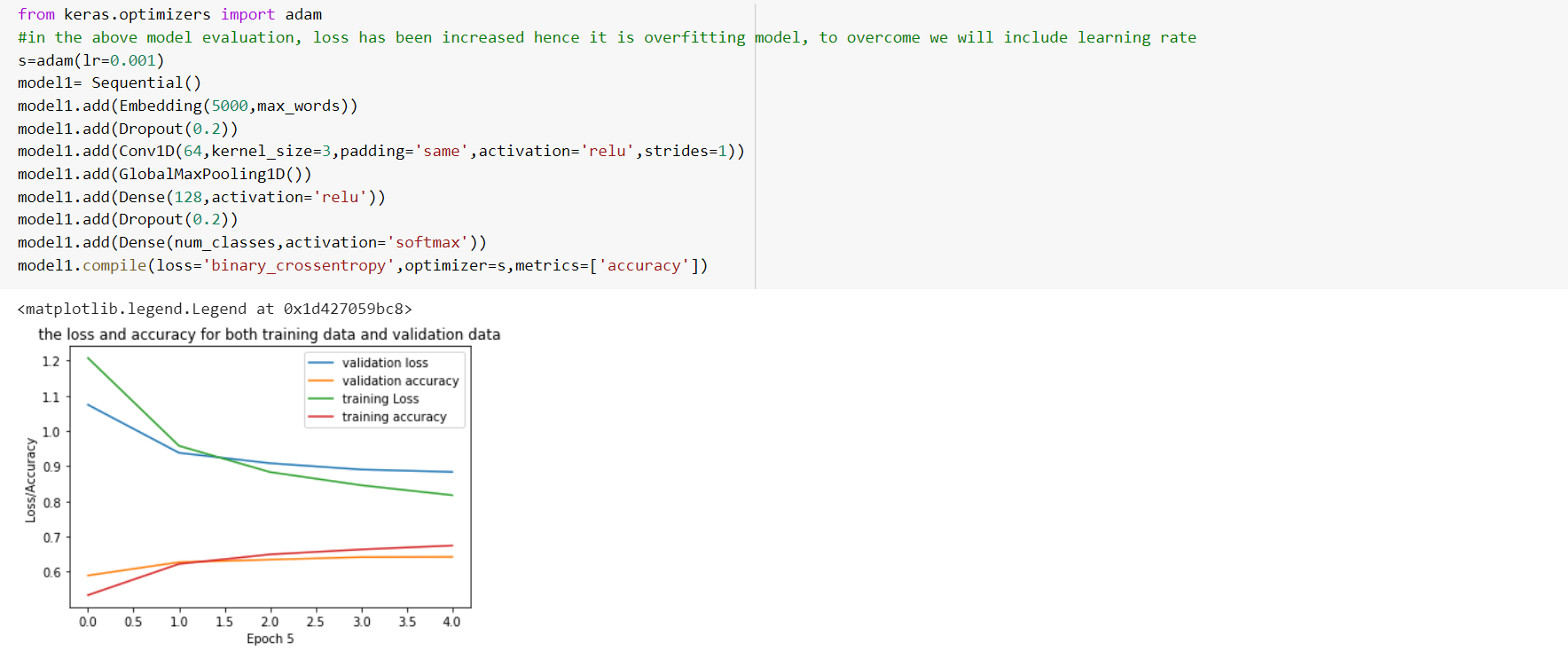
* Dropping multiple columns in pandas dataframe.
* Specifying maximum number of features to be considered.
* Applying tokenizer for train data, to obtain top maximum number of features and filters remaining.
* Repeating the for test data.
* Applying label encoder.
* Fitting the model.



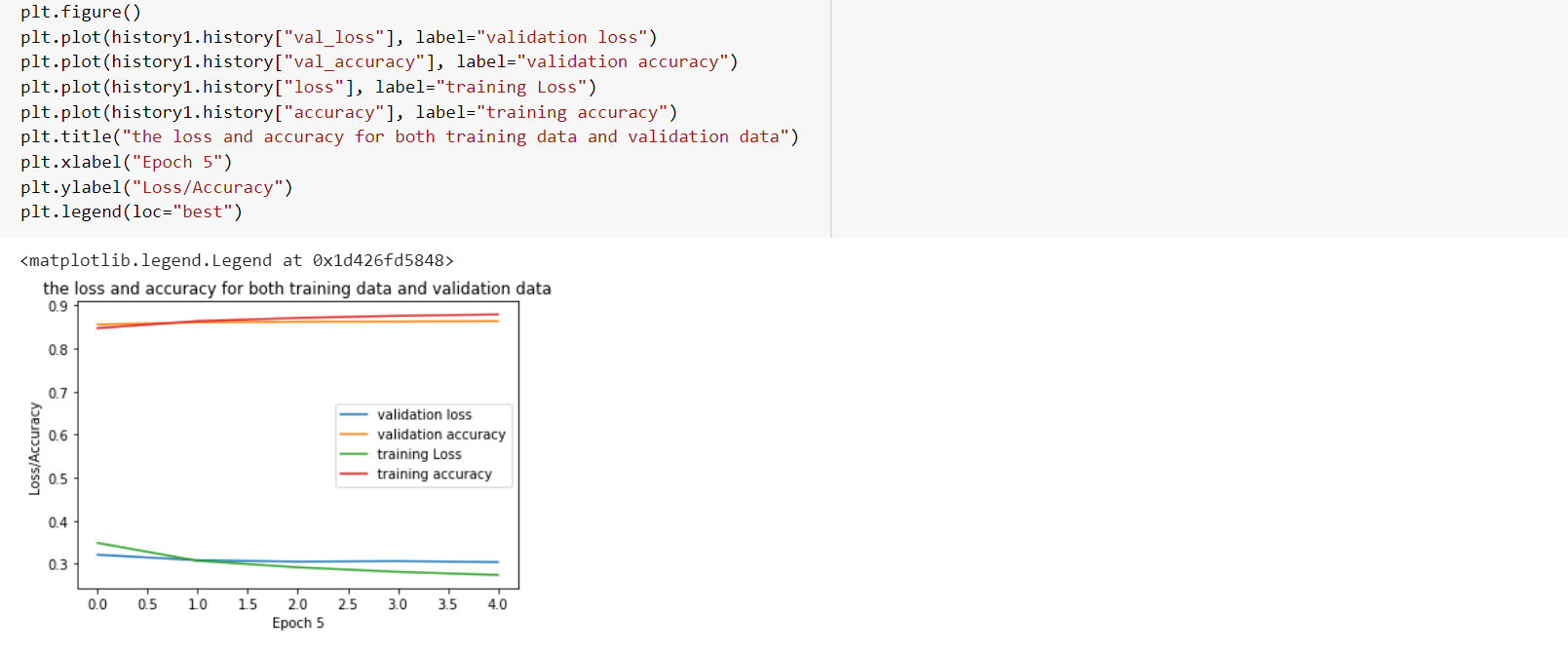
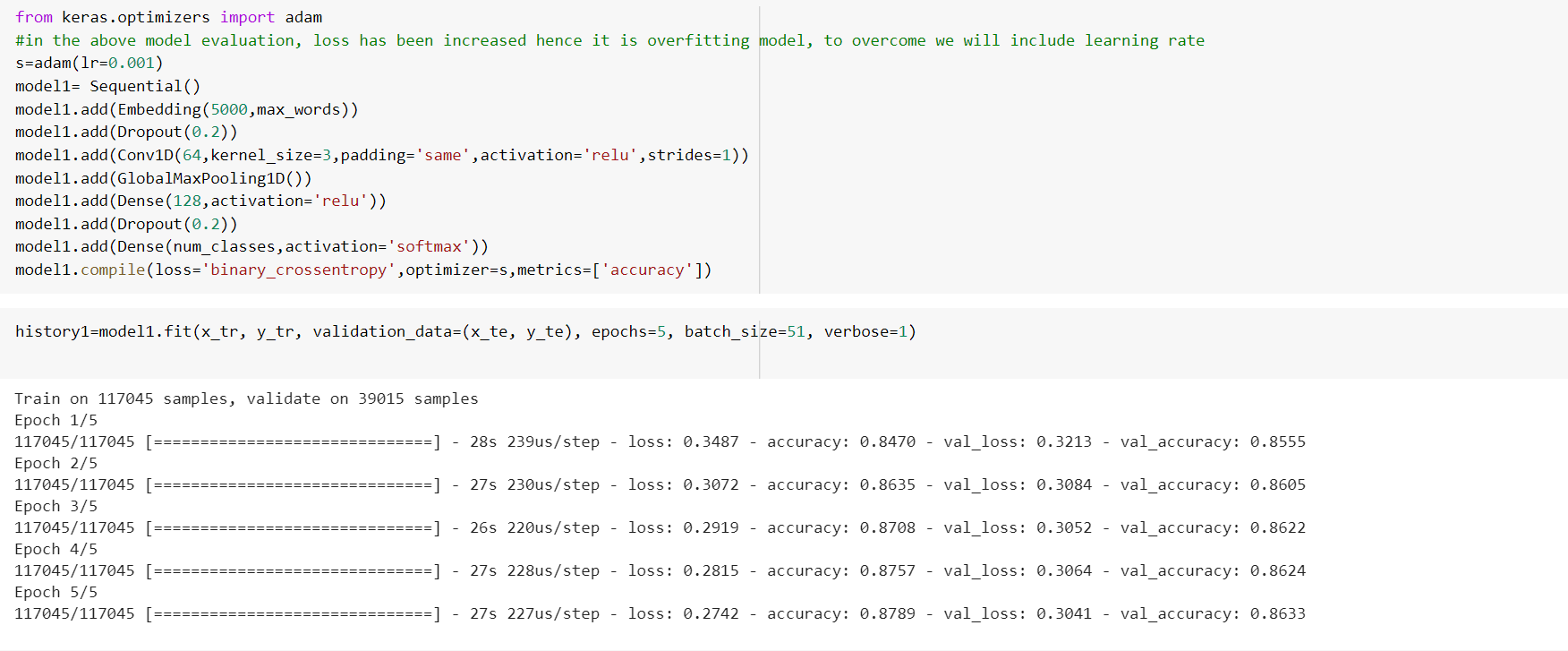
* Splitting the data into train and test data and displaying the content.
* Importing embedding from keras layers.
* Adding embedding hidden layer.
* Fitting the data and compiling the model.
* Calculating accuracy and print accuracy value.



* Plotting loss and accuracy.



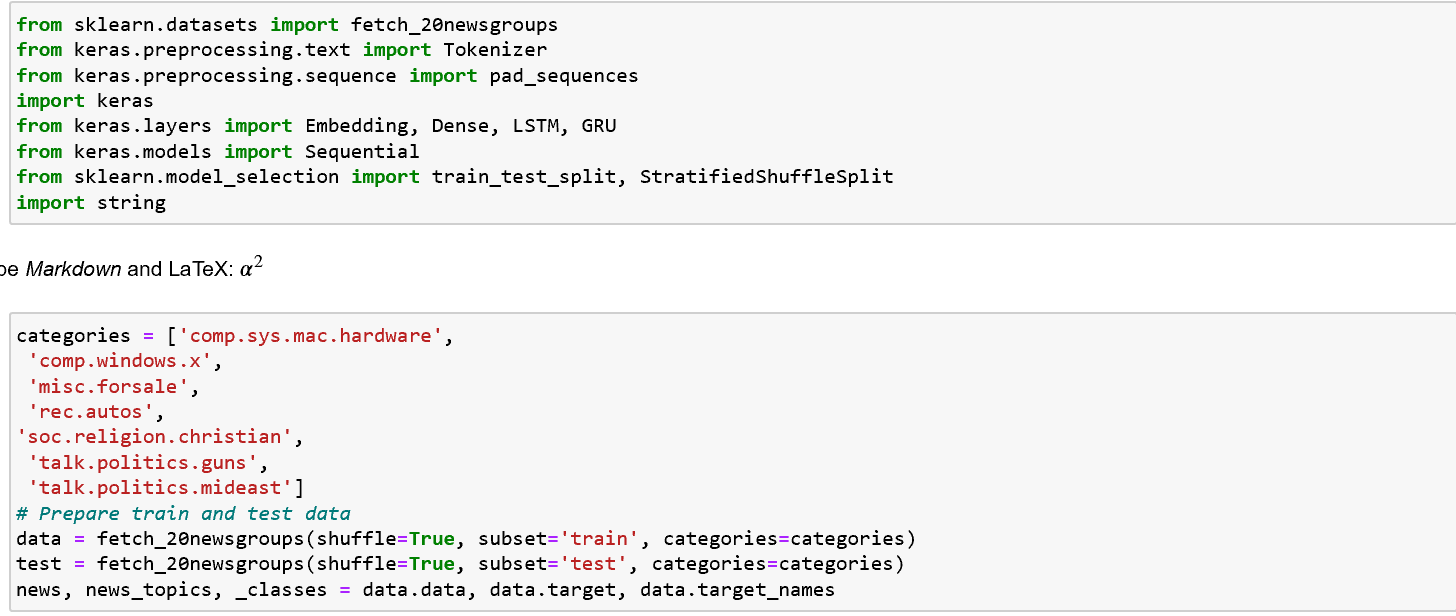
* Adding hidden embedding layers reduces this overfitting problem.



**Task-2:**

**Implementation:**

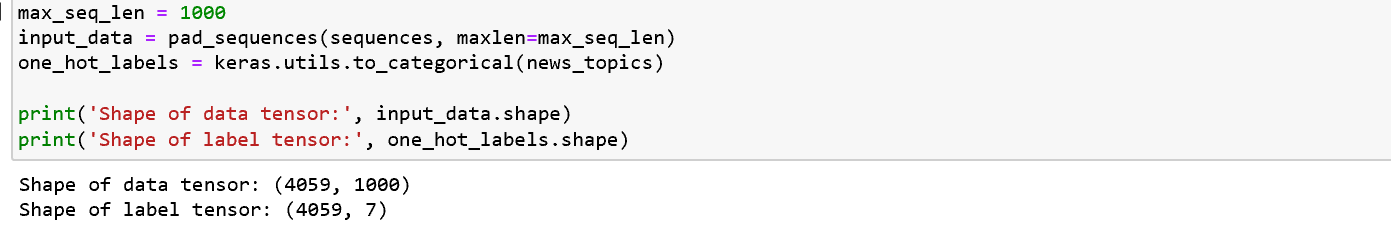
* Importing the required libraries
* Preparing training and testing data from the given dataset.



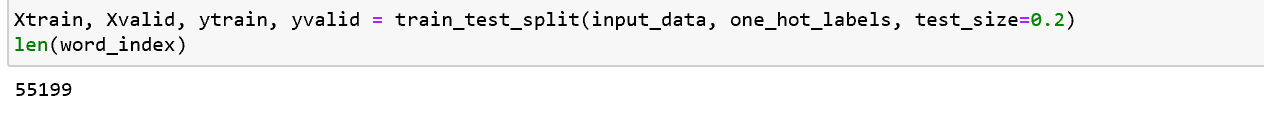
* Applying tokenization and printing unique tokens.



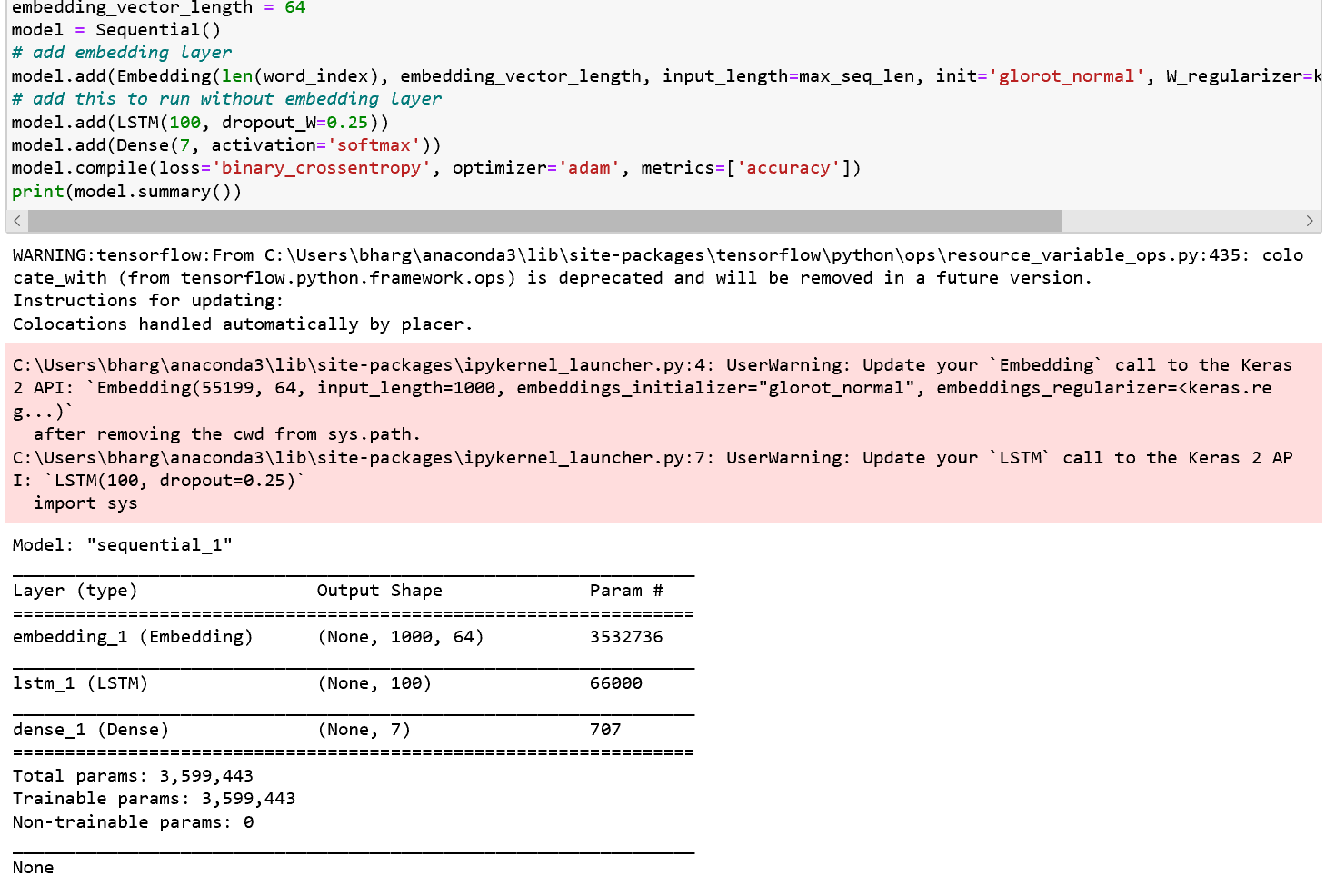
* Applying to\_categorial encoding to convert categorial data to numerical data.



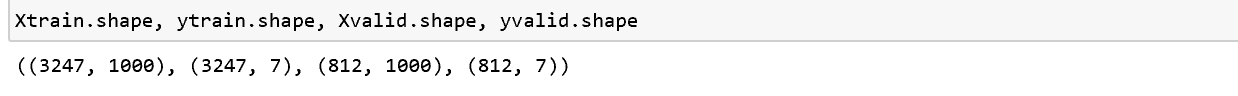
* Splitting to train and test data



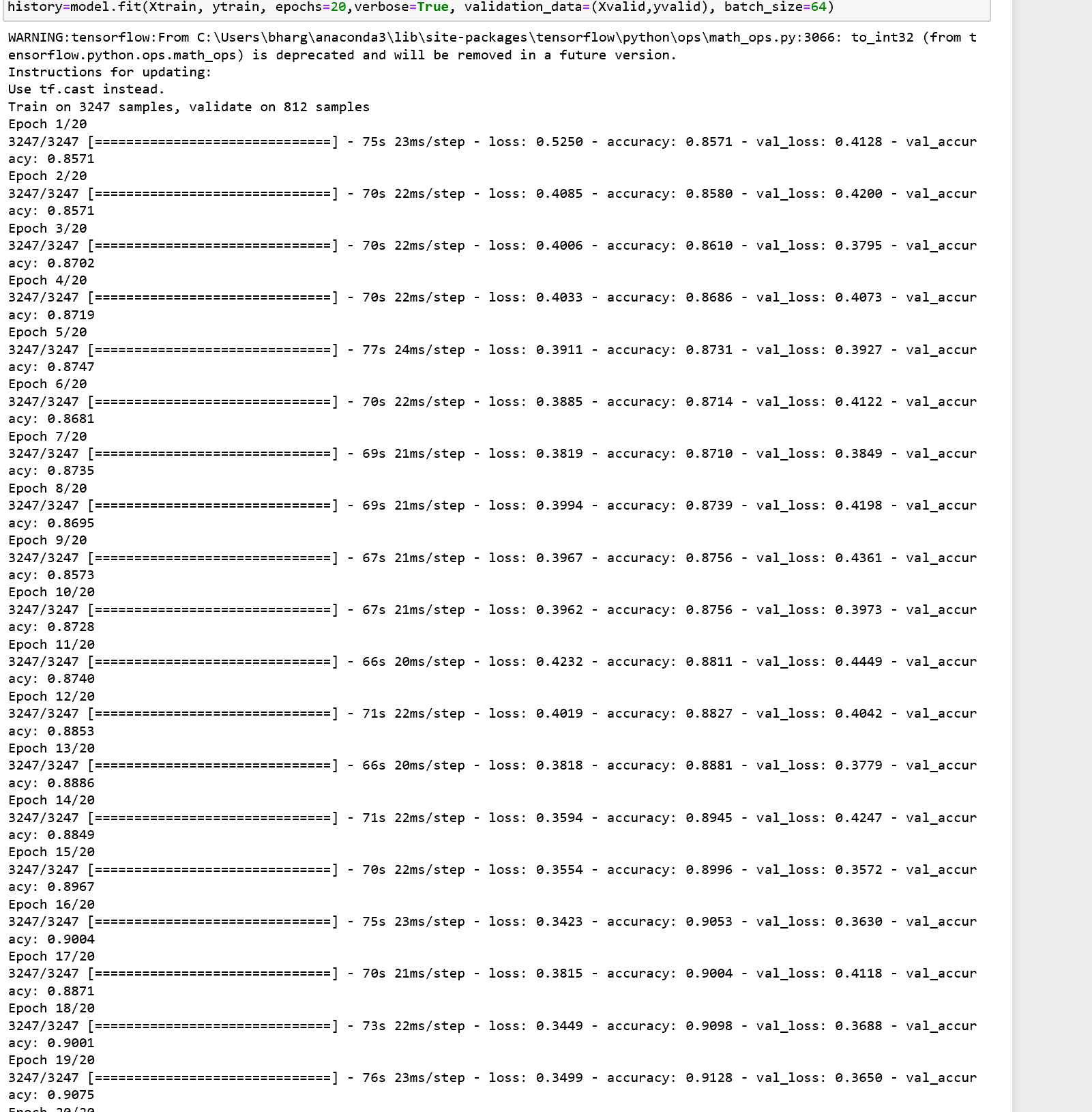
* Creating sequential model and adding embedding layer to it.
* Compiling the model and display the summary.



* Displaying the size of splitted train and test data.



* Creating a history object by model fitting.



* Plotting the accuracy and loss.



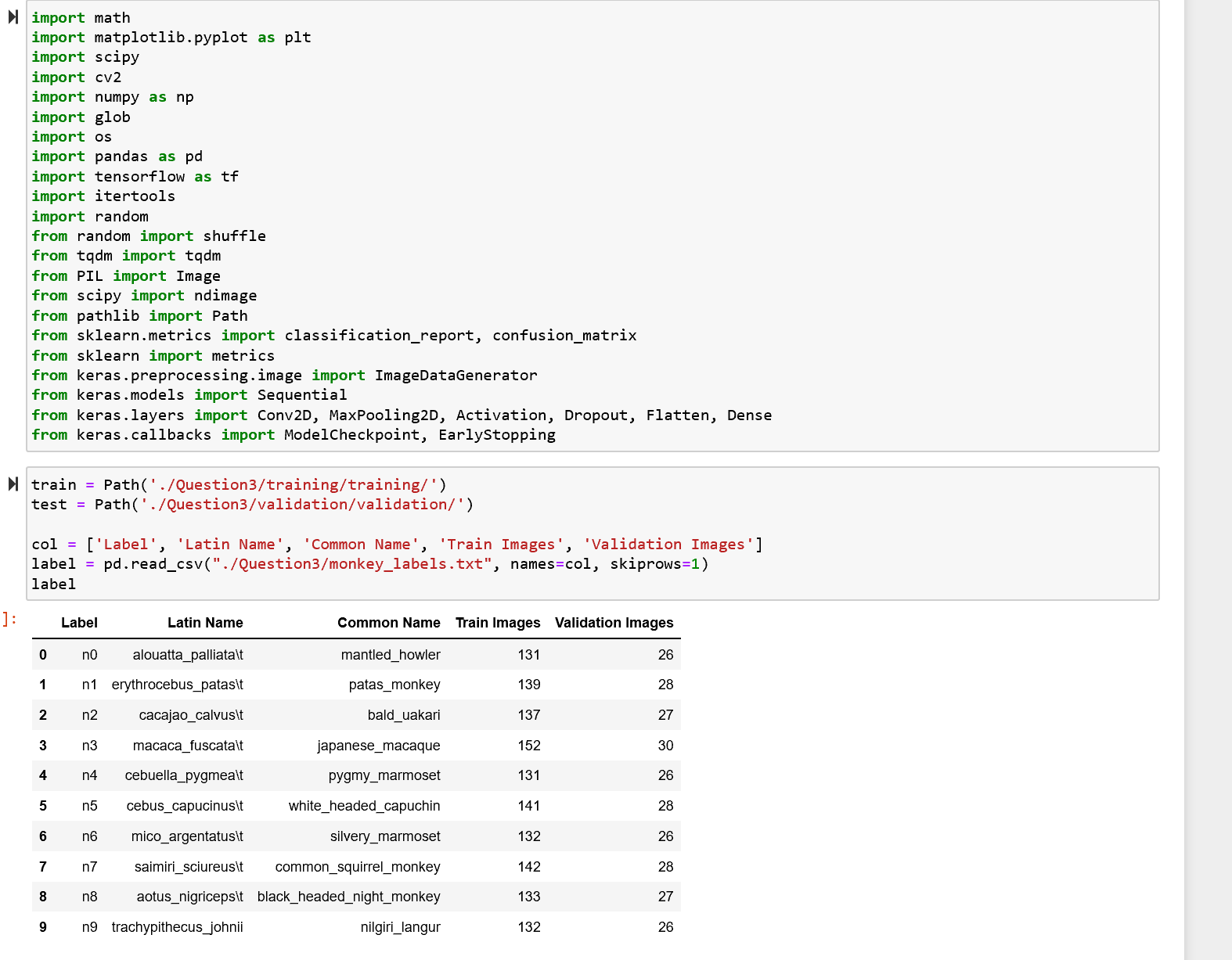
**Task-3:**

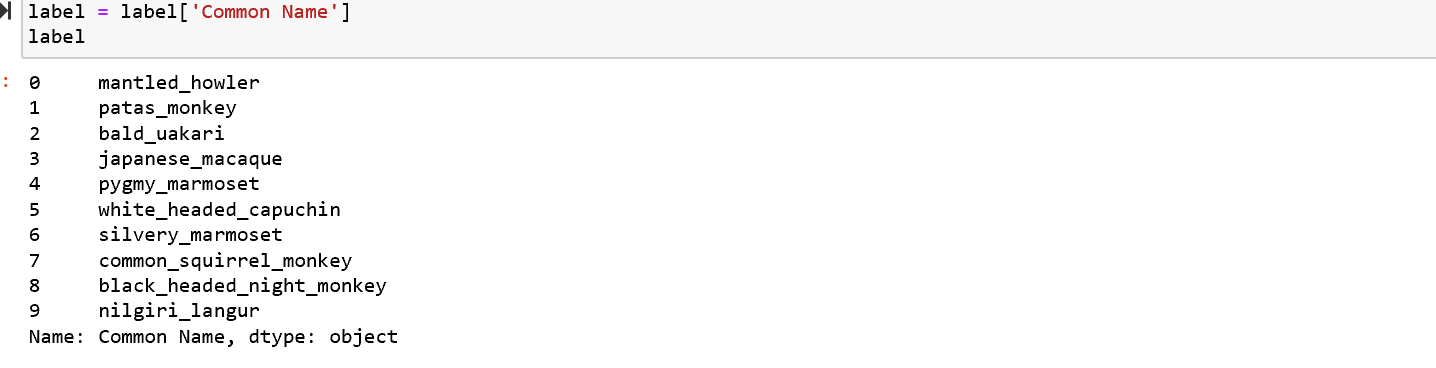
**Implementation:**

* Installing scikit-image and opencv-python libraries.



* Importing all other required libraries.
* Read training and the testing data.
* From labels.txt, add the label names to the column.
* Printing the content in 'common name' column to check.





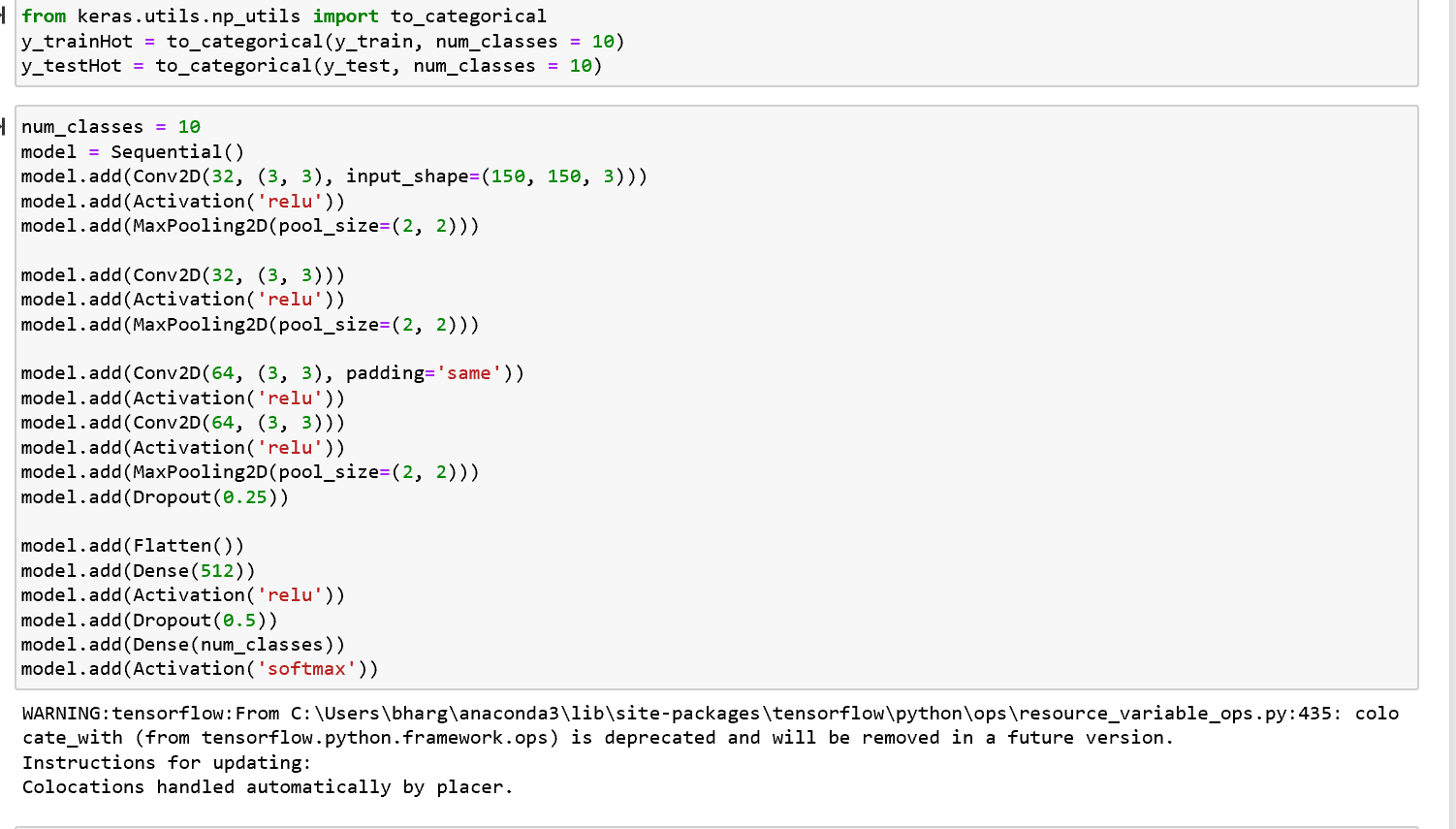
* Now Print the images with the different labels.



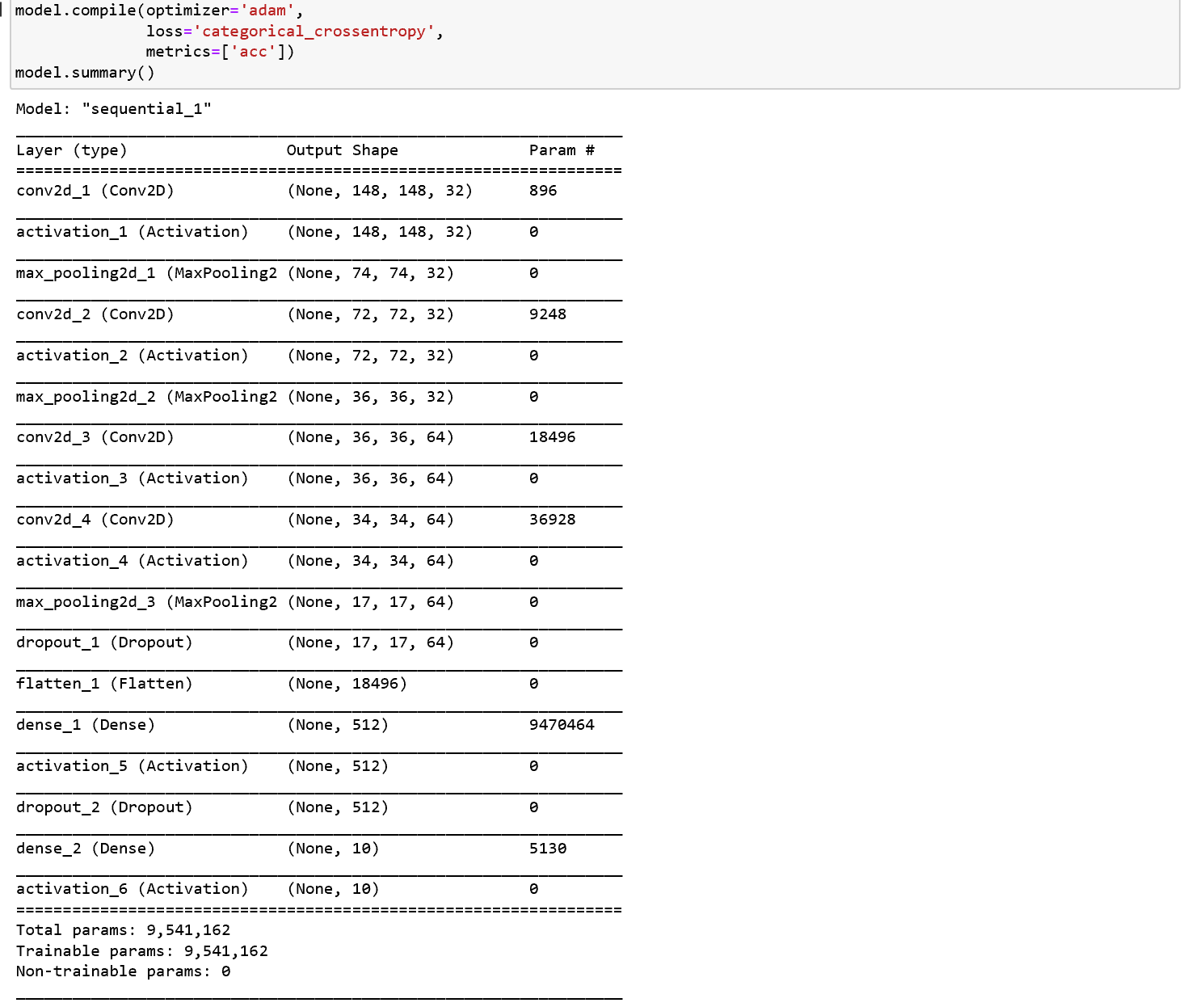
* Importing tqdm and then append all the folder names to read the image data and store in array.



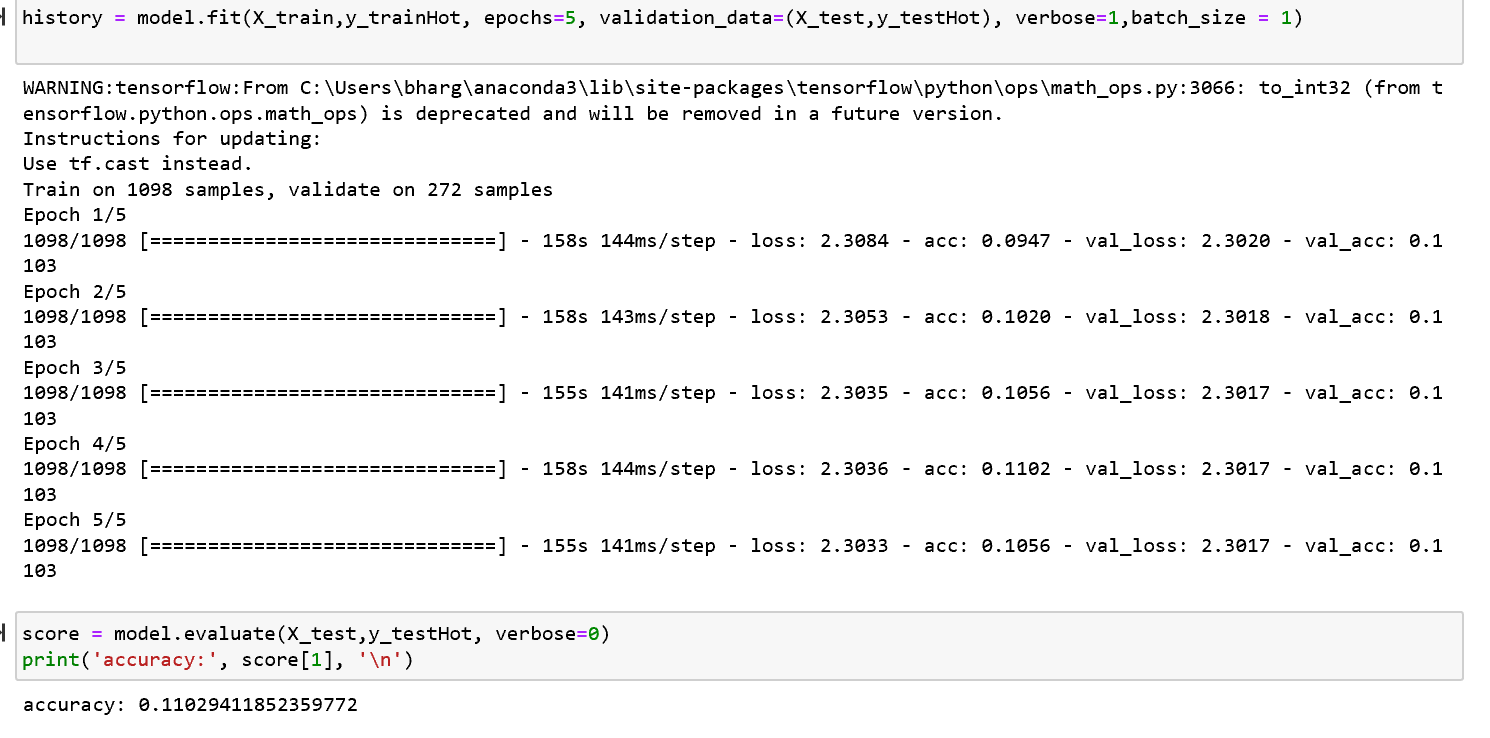
* Using hot encoder, convert categorical columns to numerical columns.
* Creating the sequential model.



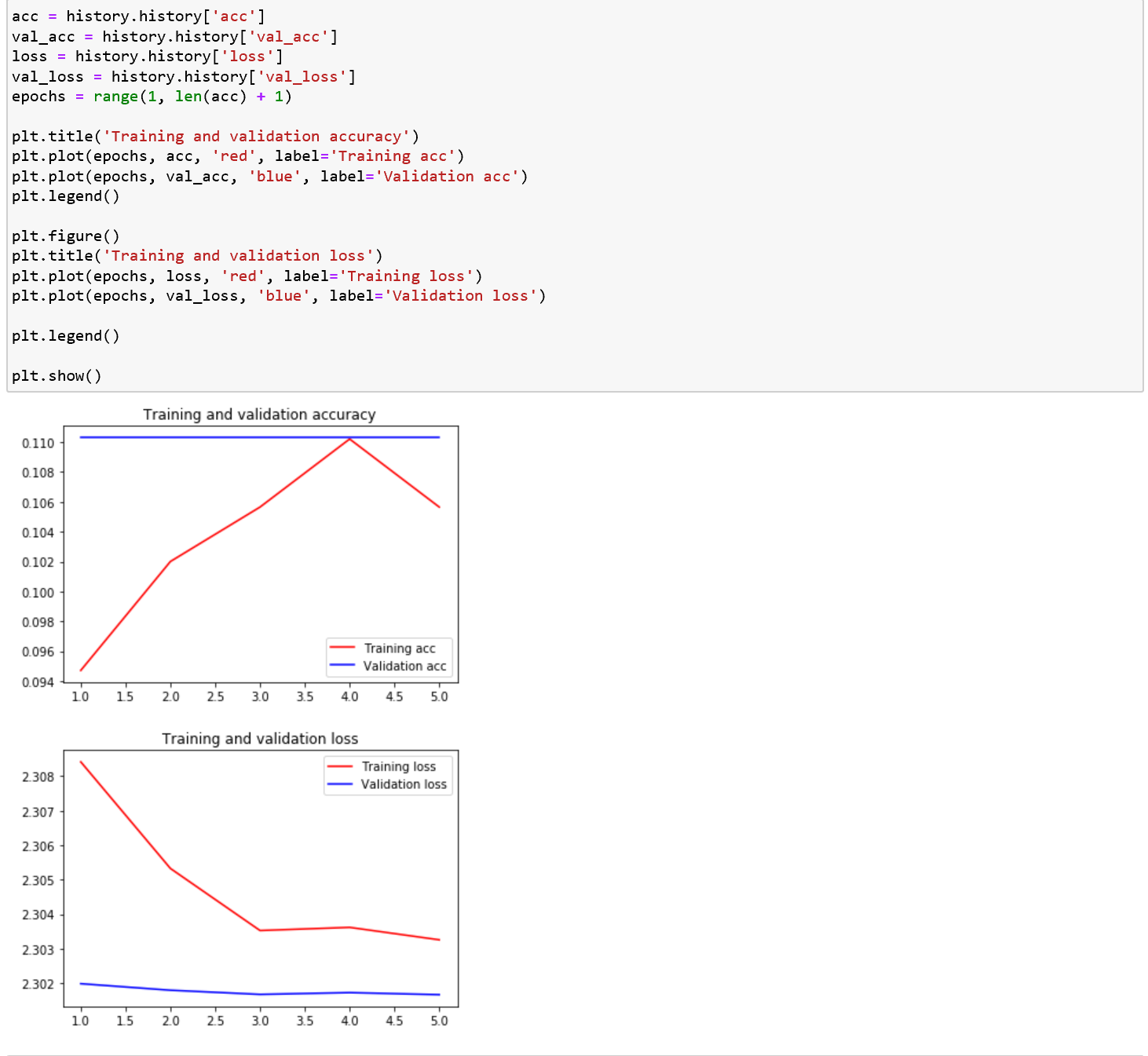
* Compiling the model and get the summary.



* Applying the history object and predicting the accuracy and loss.

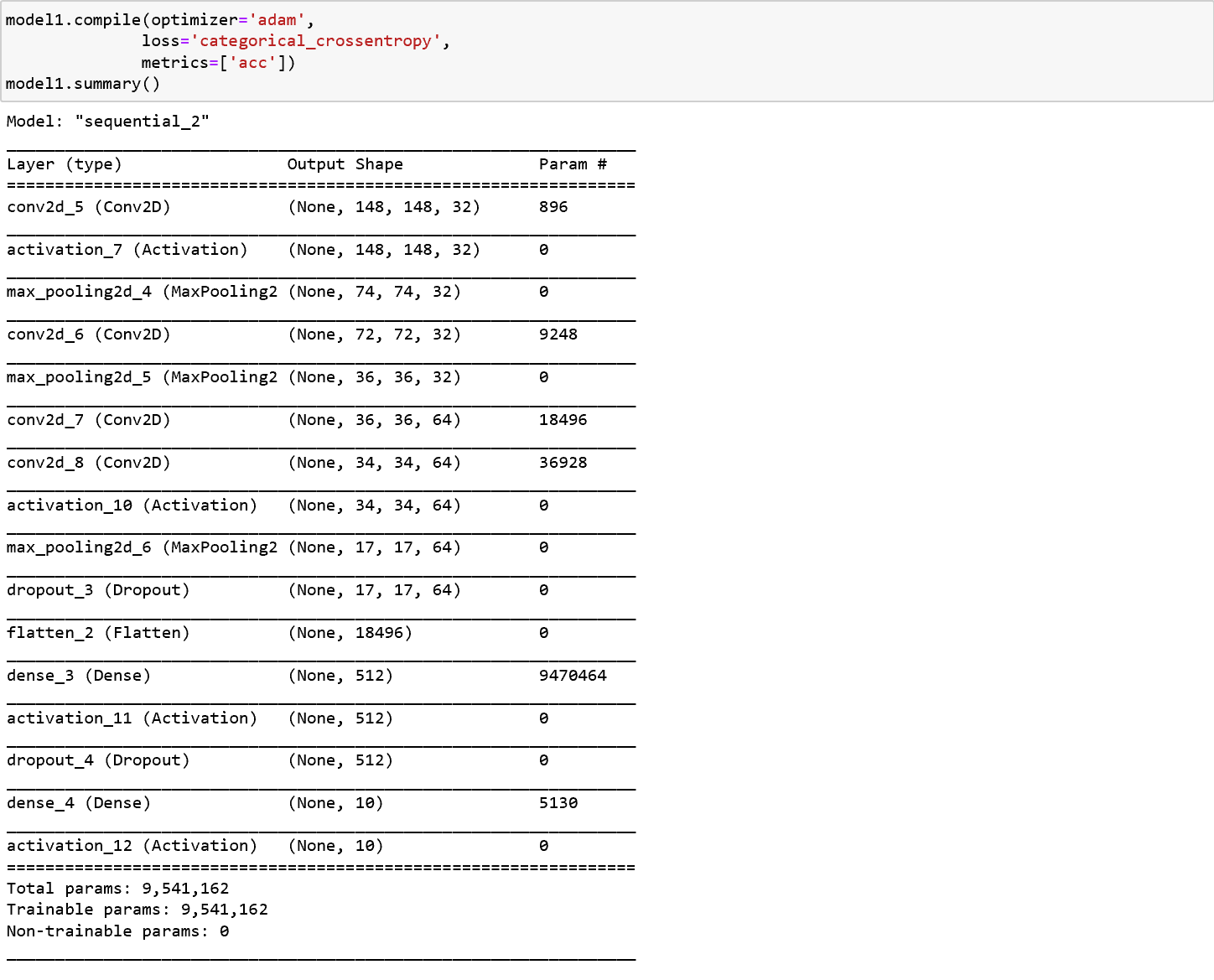


* Plotting the loss and accuracy.

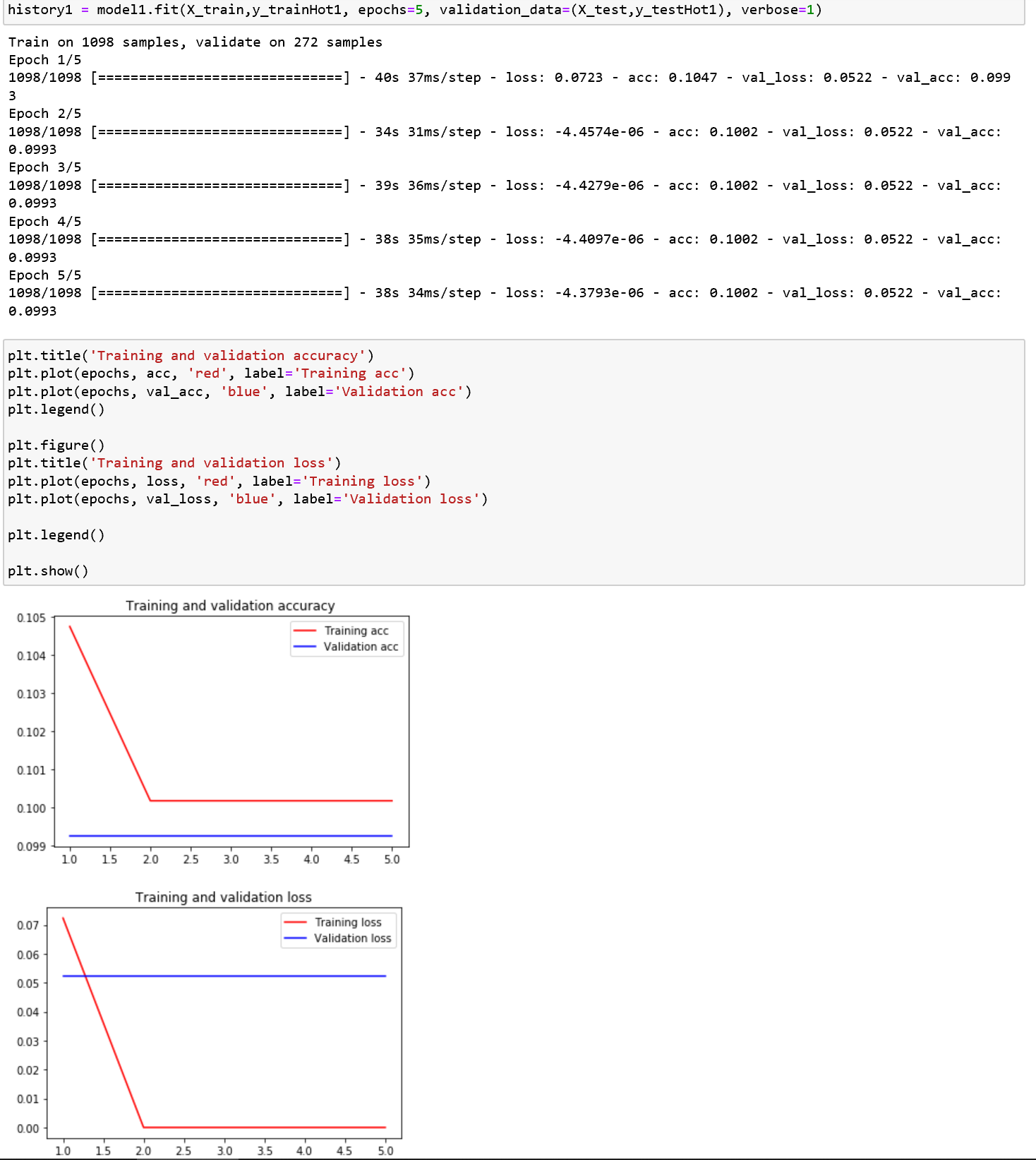


* Fitting it to the standard scaler and initializing the model.

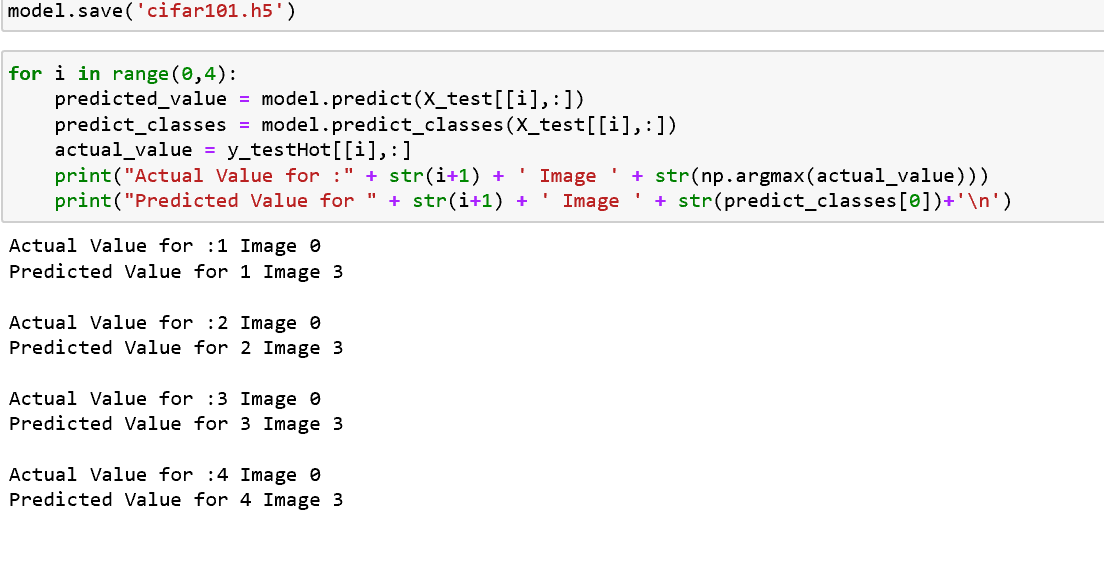




* Applying the history object and Predicting the loss and accuracy.
* Plotting the loss and accuracy.



* Saving the model and displaying the actual and predicted values for images.



**Task-4:**

**Implementation:**

* In this program, first we import the required libraries and load the dataset of news headlines.

A screenshot of a social media post

Description automatically generated

* Next, we move on to Dataset preparation where we perform text cleaning of the data which includes removal of punctuations and lower casing all the words.

A screenshot of a social media post

Description automatically generated

* Then, we apply N-gram Tokenization where we convert data to sequence of tokens.

A screenshot of a cell phone

Description automatically generated

* Next, we use pad\_sequence function to pad the sequences of the obtained tokenized data. To input this data into a learning model, we need to create predictors and label. We create N-grams sequence as predictors and the next word of the N-gram as label.

A screenshot of a social media post

Description automatically generated

* We now create a LSTM model for our program where we add three layers: input (Takes the sequence of words as input), lstm layer with 100 neurons and dropout 10% and output layer that uses softmax activation function. We compile the model and fit it with 100 epochs.

A screenshot of a cell phone

Description automatically generated

A picture containing table

Description automatically generated

* Next we write the function to predict the next word based on the input words. When we give the input, it first tokenizes it and pads the sequence and then passes it into the trained model to get the predicted word.

A screenshot of a social media post

Description automatically generated

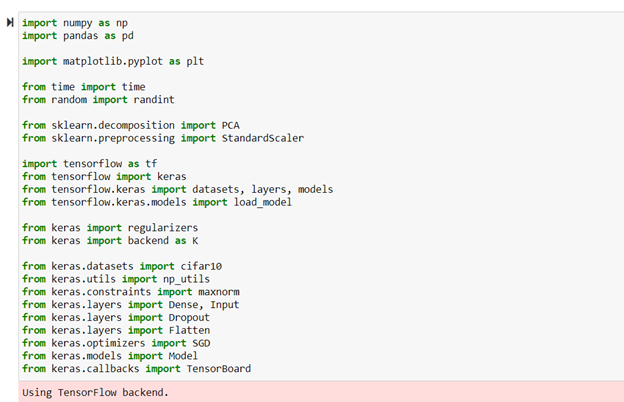
**Task-5:**

**Implementation:**

**Question 5:**

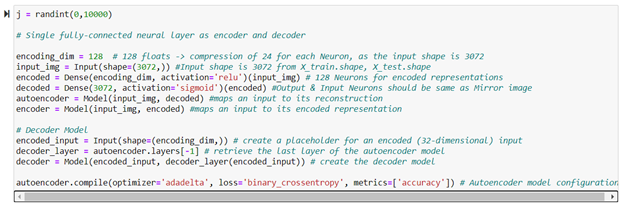
**Apply Autoencoder on theCifar\_10datasetand then pass the result of Autoencoder to CNN or LSTM or three layers model to classify data**

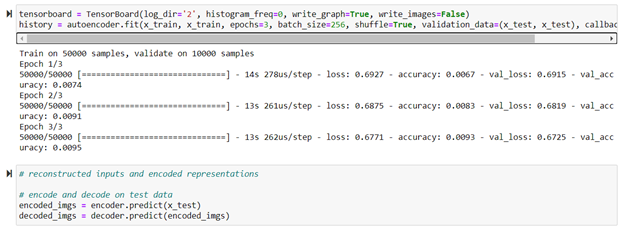
* First import all the necessary libraries.
* Normalize train and test data and then reshape.



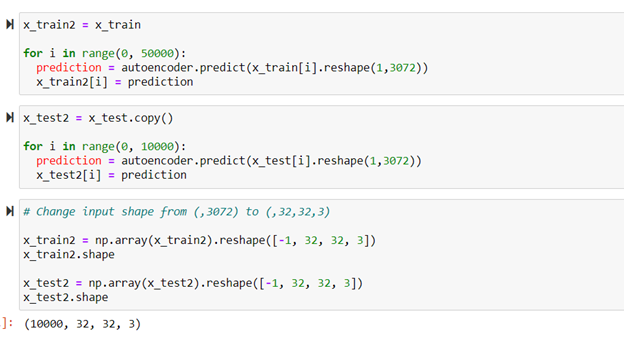


* Add the encoding layer with padding same.
* Also add the decoding layer with activation sigmoid.

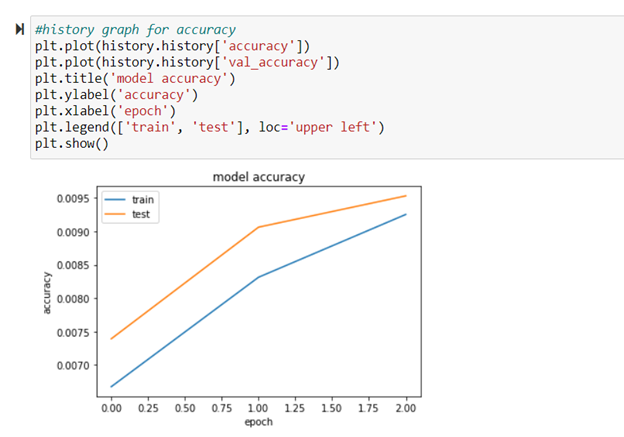


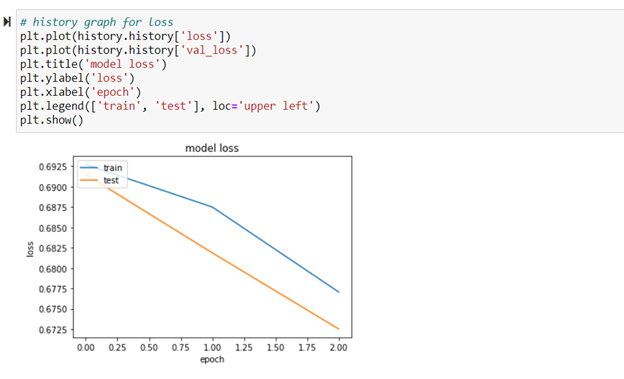


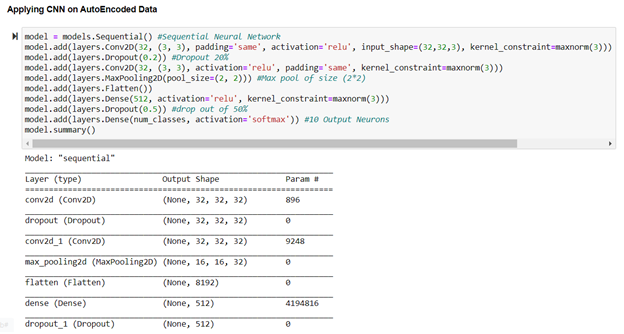
· Now compile the autoencoder and print the summary.

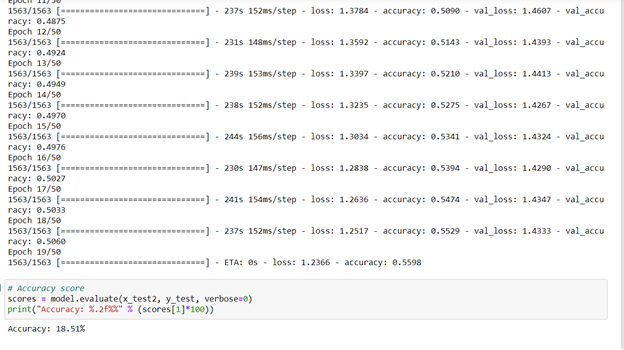


* · Plot the history object and then predict the loss and accuracy.







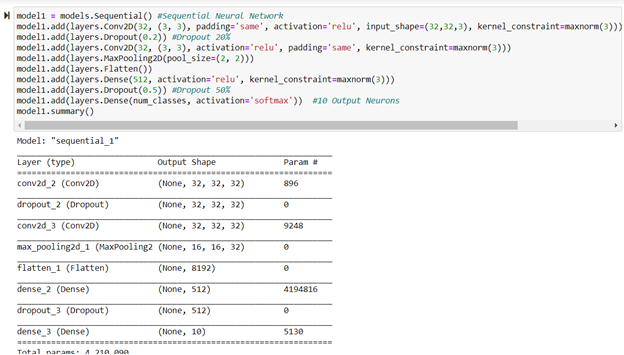


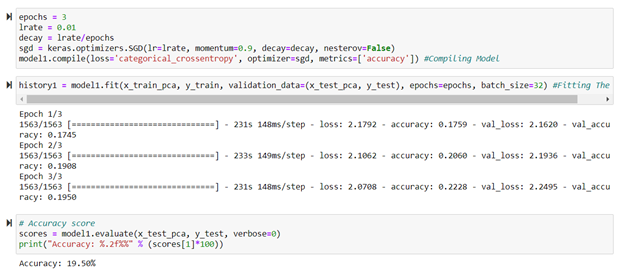
**a) Repeat the same thing with PCA (apply PCA on the dataset and then pass the result to CNN or LSTM or three layers model)**

* Import the necessary libraries.

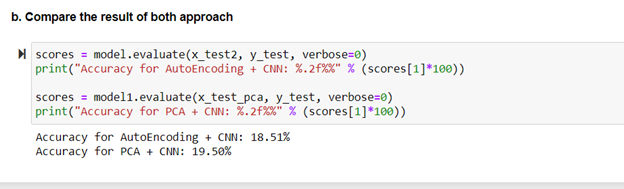


* Read the train and the test data.
* Reshape the train and the test data into two dimensional to fit into PCA.





* Now import PCA and then fit x\_train\_flat and x\_test\_flat into PCA
* Initialize the sequential model with the convolutional layers.
* Compile the model
* Plot the history object and then predict the loss and accuracy.



**Datasets**

1. https://umkc.app.box.com/s/nfaji3a8c86yfy5f4f9pzidmurvj82dz
2. https://umkc.app.box.com/s/qhtpjz7hdb8vdzmwscp7alvs6ilt0zf2
3. <https://www.kaggle.com/slothkong/10-monkey-species>
4. <https://umkc.app.box.com/s/v9d6l40pmqsu4x4h6edhvgxx8yn8h47d>

**Evaluation**

1. Loss has increased and hence it is an overfitting model.
2. Performance increased after adding embedding layer.
3. Accuracy has been changed before scaling and after scaling. The predicted value is different from the actual value.
4. The model has produced a fine output for given seed text US, India and Coronavirus.
5. The accuracy is high in PCA+CNN than AutoEncoding+CNN

**Conclusion**

We have successfully applied the concepts and obtained the required accurate outputs to all the programs. This project Exam has enabled us to gain deeper understanding on the working of Deep Learning methods in practice.

**Submission By:**

**Team 4-**

1. **Pooja Reddy Gopu – 16**
2. **Tejaswi Guturu – 18**
3. **Mounika Rachamadugu – 37**
4. **Pujitha Sakhavarapu - 40**