**UNIVERSITY OF MISSOURI, KANSAS CITY**

**PYTHON & DEEP LEARNING – CS 5590**

**PART – 2 - DEEP LEARNING**

**LAB – 2 REPORT**

**BHAVYA TEJA GURIJALA**

**STUDENT ID: 16220446**

**CLASS ID: 14**

**TASK:**

Implement the Text Classification with CNN model with new dataset using the Tensor Flow in python.

1. **INTRODUCTION:**

Convolutional Neural Networks (CNN), a technique in the field of Deep Learning, have been a revolutionary strength in the applications of the Computer Vision, exclusively in the past half-decade or so.

CNNs are fundamentally several layers of convolutions with nonlinearfunctions like tanh applied to the results. In a conventional neural network, each input neuron is connected to the output neuron in the subsequent layer. We don’t apply this in CNN. In its place, we use convolutions above the input layer to process the output. Each region in the input is connected to the neuron which leads to the local connection. Every layer applies dissimilar filters, characteristically hundreds or thousands, and syndicates their results. In the training phase, the Convolutional Neural Network learns the values of its respective filters automatically contingent on the task you wish to perform.

1. **OBJECTIVES:**

The objective is to evaluate a CNN architecture on a text classification dataset, mostly comprised of Text Categorization tasks. The CNN architecture succeeds in getting very good performance across the given dataset. The input layer is a sentence comprised of concatenated word2vec word embedding’s. The main moto is to classify the kaggle dataset into 11 classes.

1. **APPROACH:**

The approach is to develop a model of multiple layers where the first layer embeds the words into low dimensional vectors. The consecutive layer performs the convolution of the embedded words using the multiple filter sizes. Then this layer is followed by the implementation of the max pool which converts the convolutional layer into a long feature vector. The predictions are generated using the feature vector from max pooling by computing a matrix multiplication and choosing the class with the highest score.

1. **WORKFLOW:**

The workflow of the entire model building is as follows:

1. The dataset is read by the file path and all the characters are converted to strings.
2. Building the vocabulary by collecting the most frequent words.
3. Using the dictionaries in python, every word and its index is stored as a key value pair.
4. Once the data preprocessing is done, we build a graph using the word2vec model.
5. The process of building the model starts with defining the placeholders for input and output.
6. By defining the weights, we build the model and the loss function. The loss function is optimized using the Gradient Descent Optimizer.
7. We train the model by defining the number of training steps and the weights field.
8. Once the model is build, we plot the graph and visualize it on tensor board.
9. **DATASET:**

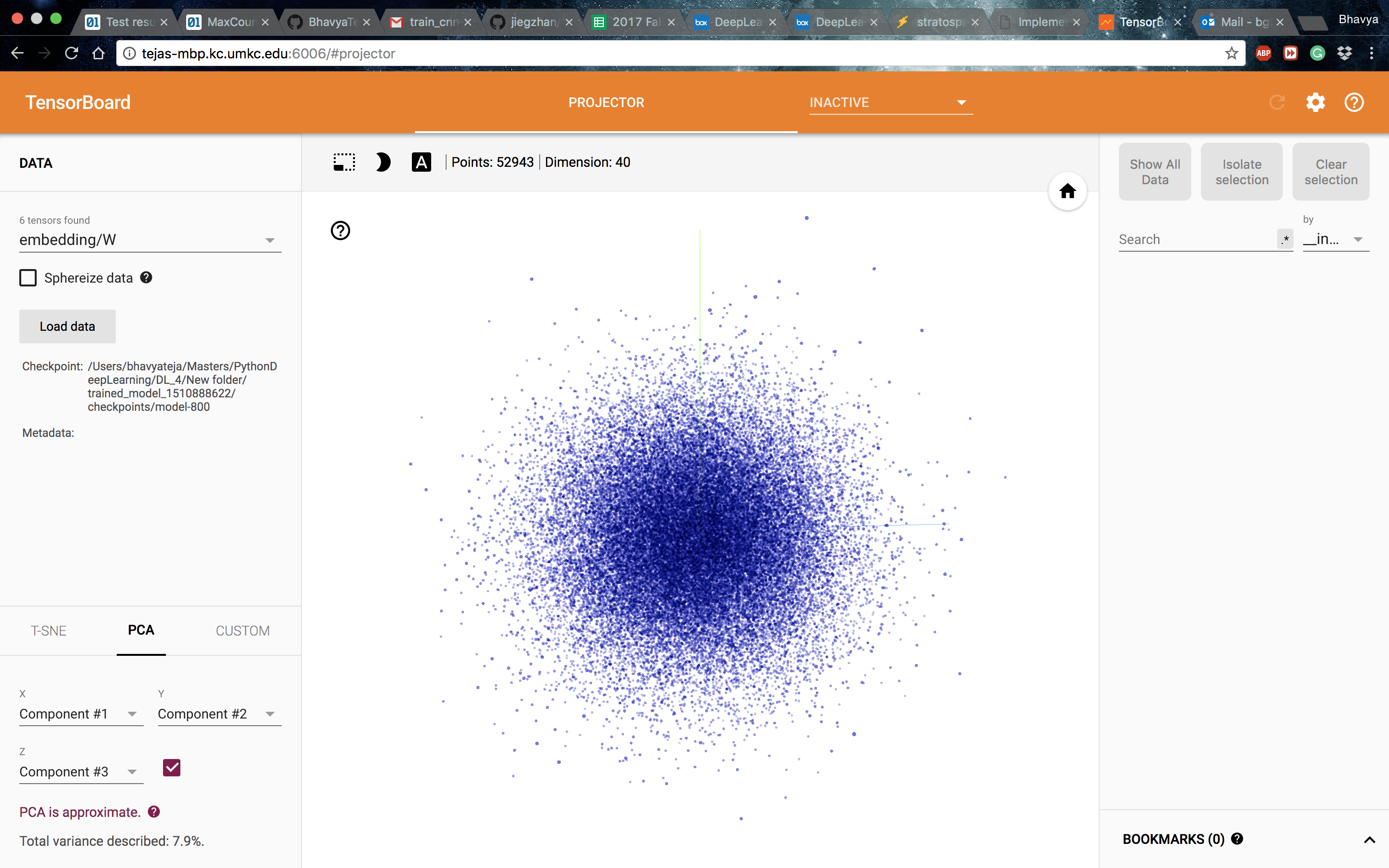
The dataset is a consumer complaint narrative about the financial issues. The dataset is downloaded from kaggle. The output will the product which shows the categories or the classes to which the complaint belongs. By using the convolutional neural networks and word embedding’s on tensor flow, the model was built.

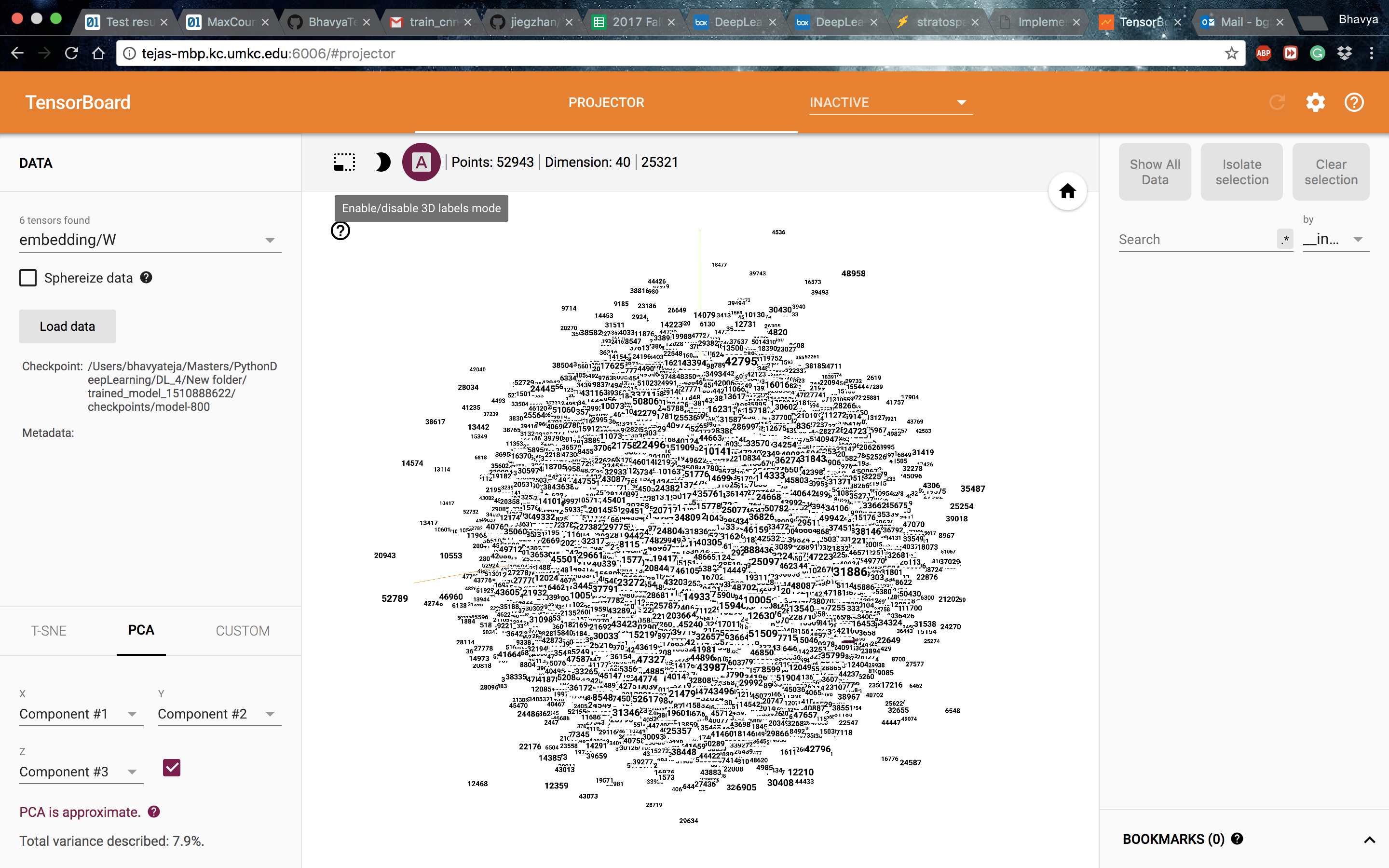
1. **PARAMETERS:**

The parameters to be considered while building the model are the vocabulary size, number of classes, vocabulary size, embedding size, filter sizes, number of filters and the r2 lambda value. We are classifying the model into 11 classes. The labels for this 11 classes are listed below.

1. Bank Account
2. Consumer Loan
3. Credit Card
4. Credit Reporting
5. Debt Collection
6. Money Transfers
7. Mortgage
8. Payday Loan
9. Prepaid Loan
10. Student Loan
11. Other Financial Service
12. **EVALUATION & DISCUSSION:**

The model can be evaluated by the performance graphs portray the PCA of the classes where every word class is treated as a data point.



****

1. **CONCLUSION:**

Eventually we conclude that the metrics are not smooth since we use small batch sizes for training the model.

***References:***

[***https://github.com/jiegzhan/multi-class-text-classification-cnn***](https://github.com/jiegzhan/multi-class-text-classification-cnn)

[***https://www.kaggle.com/cfpb/us-consumer-finance-complaints***](https://www.kaggle.com/cfpb/us-consumer-finance-complaints)

[***http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/***](http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/)

[***https://github.com/dennybritz/cnn-text-classification-tf***](https://github.com/dennybritz/cnn-text-classification-tf)