**PROJECT REVIEW REPORT**

On

***“*Training a Deep Learning Language Model using Keras and Tensorflow*”***

*Submitted in partial fulfillment of requirements for the award of*

**Bachelor of Technology (BTech.)**

In the department of

**Computer Science & Engineering**

****

*Submitted by*:

**BHARGAV SAIKIA**

**ET16BTHCS027**

**SAGORIKA KALITA**

**ET16BTHCS019**

**MELIU KRONU**

**ET16BTHCS056**

**ANANDA RUPA BARUAH**

**ET16BTHEC009**

**CHRISTINA DAHOTIA**

**ET16BTHEC018**

*Under the Guidance of*

**MR.** **PRANAV KUMAR,**

**Asst. Prof. CSE Dept. (Internal Mentor)**

**MR.** **AADITYA MAHESHWARI**

**(External Mentor)**

**School of Engineering & Technology**

**The Assam Kaziranga University, Jorhat, Assam**

**Nov – 2019**

**Aim of the project:**

Training a Deep Learning Language Model using Keras and Tensorflow.

**Understanding of Project:**

Using deep learning to generate information is a hot area of research and experimentation. In particular, the RNN deep learning model approach has seemingly boundless amounts of areas it can be applied to, whether it be to solutions to current problems or applications in budding future technologies. One of these areas of application is text generation, which is what this Code Pattern introduces. Text generation is used in language translation, machine translation and spell correction. These are all created through something called a language model. This Code Pattern runs through exactly that: a language model using a Long Short Term Memory (LSTM) RNN. For some context, RNNs use networks with hidden layers of memory to predict the next step using the highest probability. Unlike Convolutional Neural Networks (CNNs) which use forward propagation, or rather, move forward through its pipeline, RNNs utilize back propogation, or circling back through the pipeline to make use of the "memory" mentioned above. By doing this, RNNs can use the text inputed to learn how to generate the next letters or characters as its output. The output then goes on to form a word, which eventually ends up as a collection of words, or sentences and paragraphs. The LSTM part of the model allows you to build an even bigger RNN model with improved learning of long-term dependencies, or better memory. This means an improved performance for those words and sentences we're generating!

This model is relevant as text generation is increasingly in demand to solve translation, spell and review problems across several industries. In particular, fighting fake reviews plays a big role in this area of research. Fake reviews are a very real problem for companies such as Amazon and Yelp, who rely on genuine reviews from users to vouch for their products and businesses which are featured on their sites. As of writing this, it is very easy for businesses to pay for fake, positive reviews, which ultimately end up elevating their sales and revenue. It is equally as easy to generate negative reviews for competing businesses. Unfortunately this leads users to places and products fraudulently and can potentially lead to someone having a negative experience or worse. In order to combat these abuses and illegal activity, text generation can be used to detect what a review looks like when it is generated versus a genuine review written by an authentic user. This Code Pattern walks through the steps to create this text generation at a high level

**Technology Stack - Application Development Architecture:**

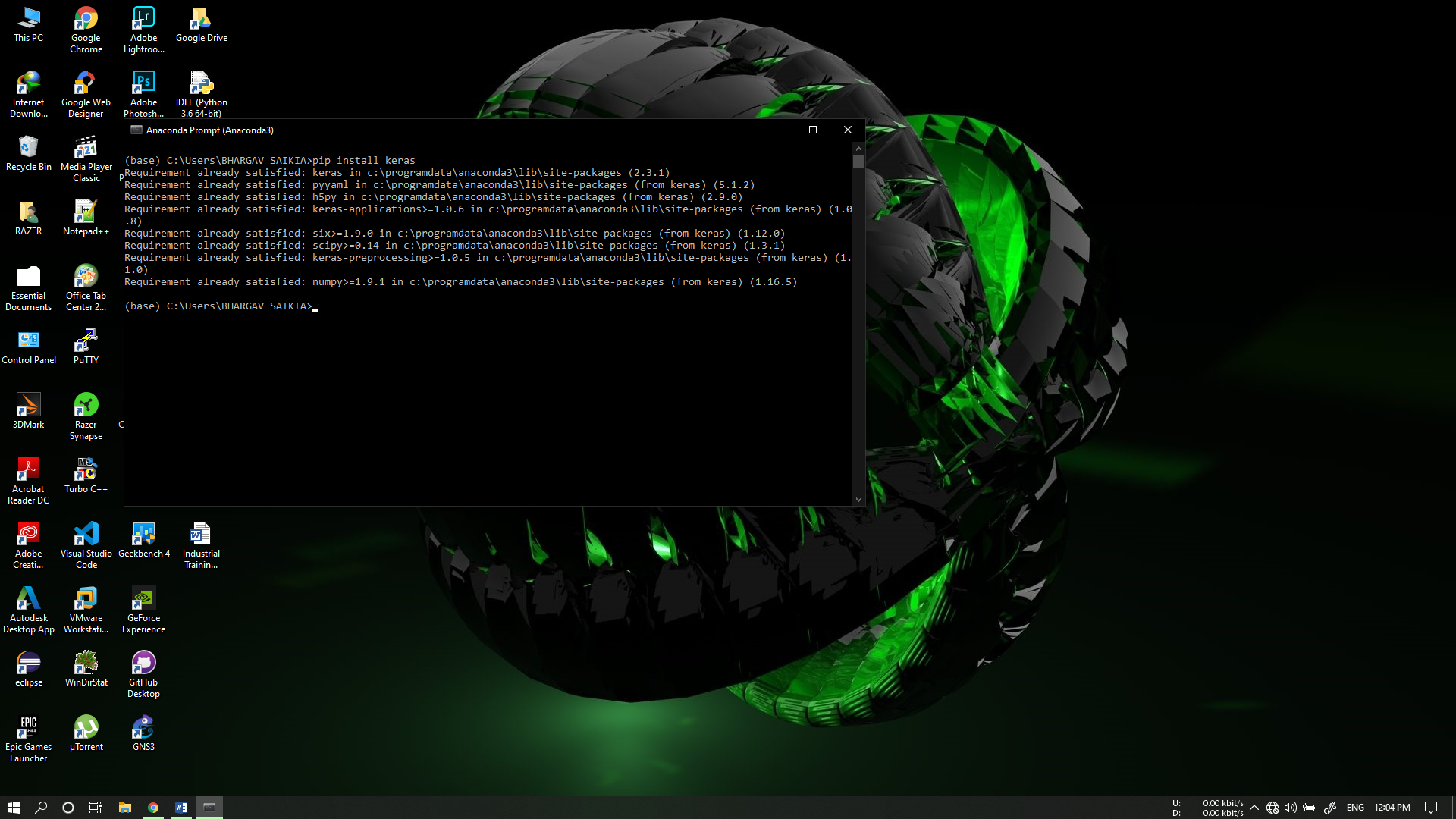
1. **Anaconda 3 :** open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X
2. **Juptyer Notebook:** The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.
3. **Keras:** Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.
4. **Tensorflow:** TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

**Datasets:**

We have used writings of **Friedrich Nietzsche** as our training data to train the model. We got the dataset from “https://s3.amazonaws.com/text-datasets/nietzsche.txt”

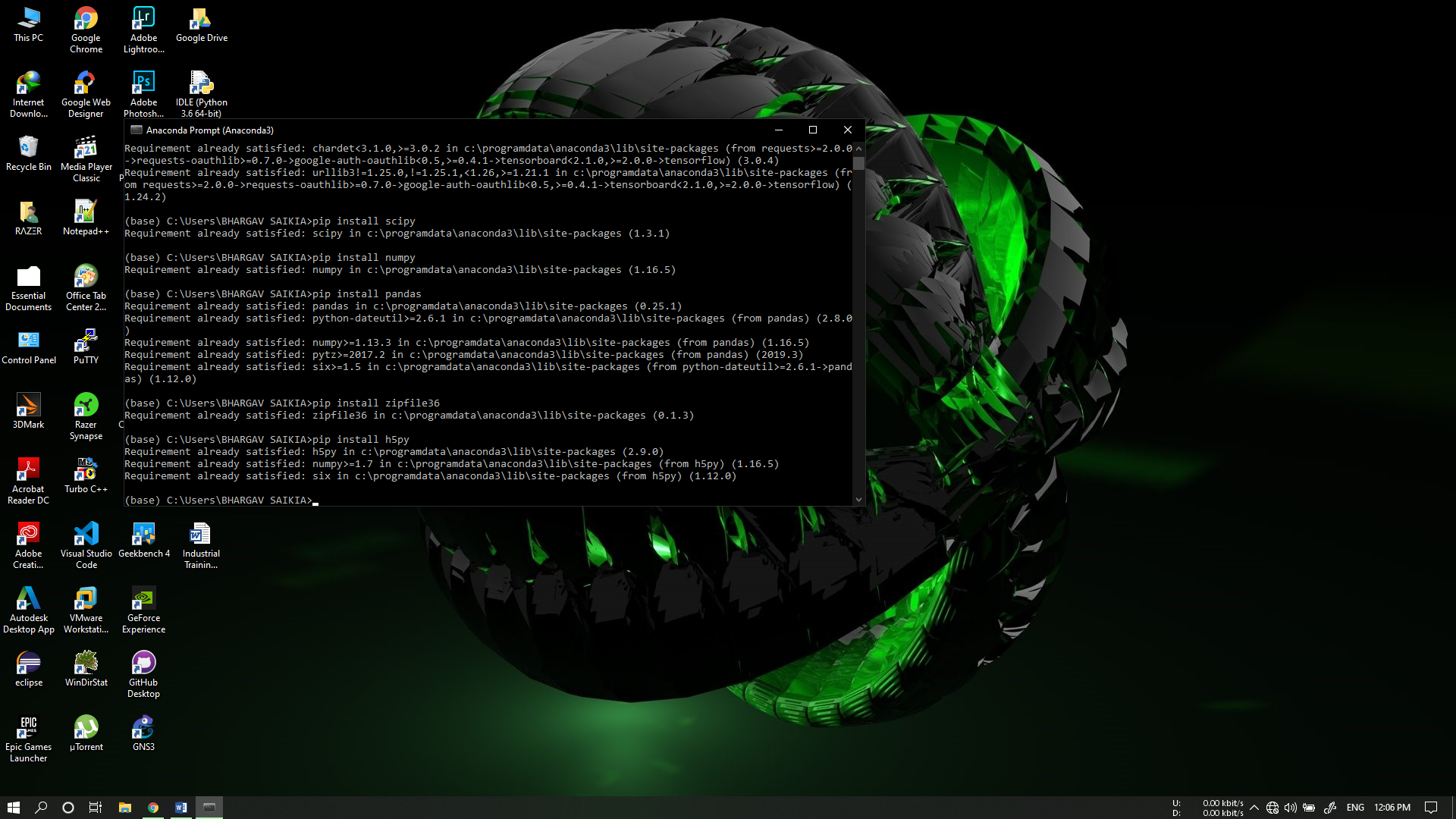
**Procedure:**

1. Open Anaconda Prompt, install Keras and Tenssorflow.

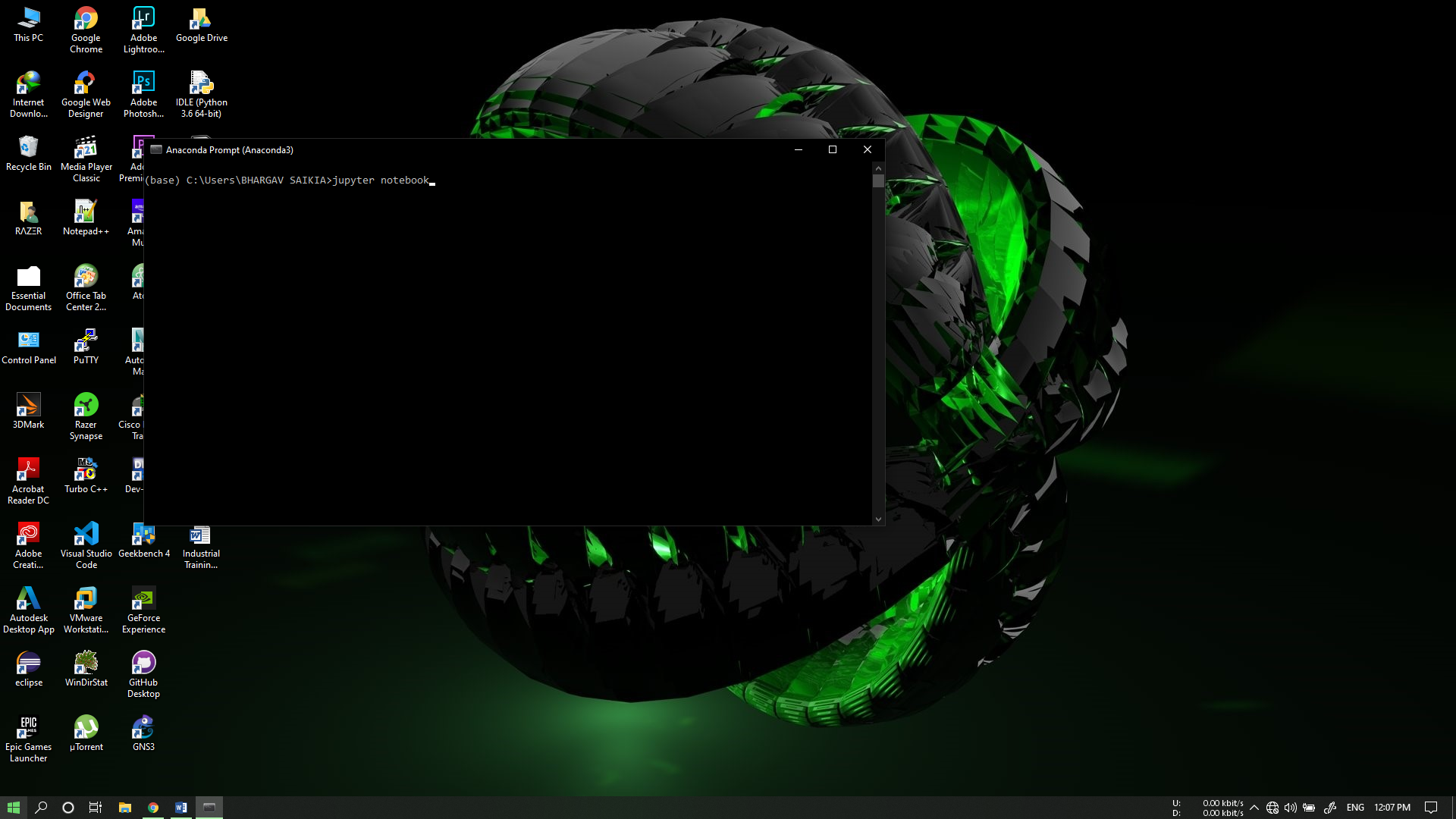
****

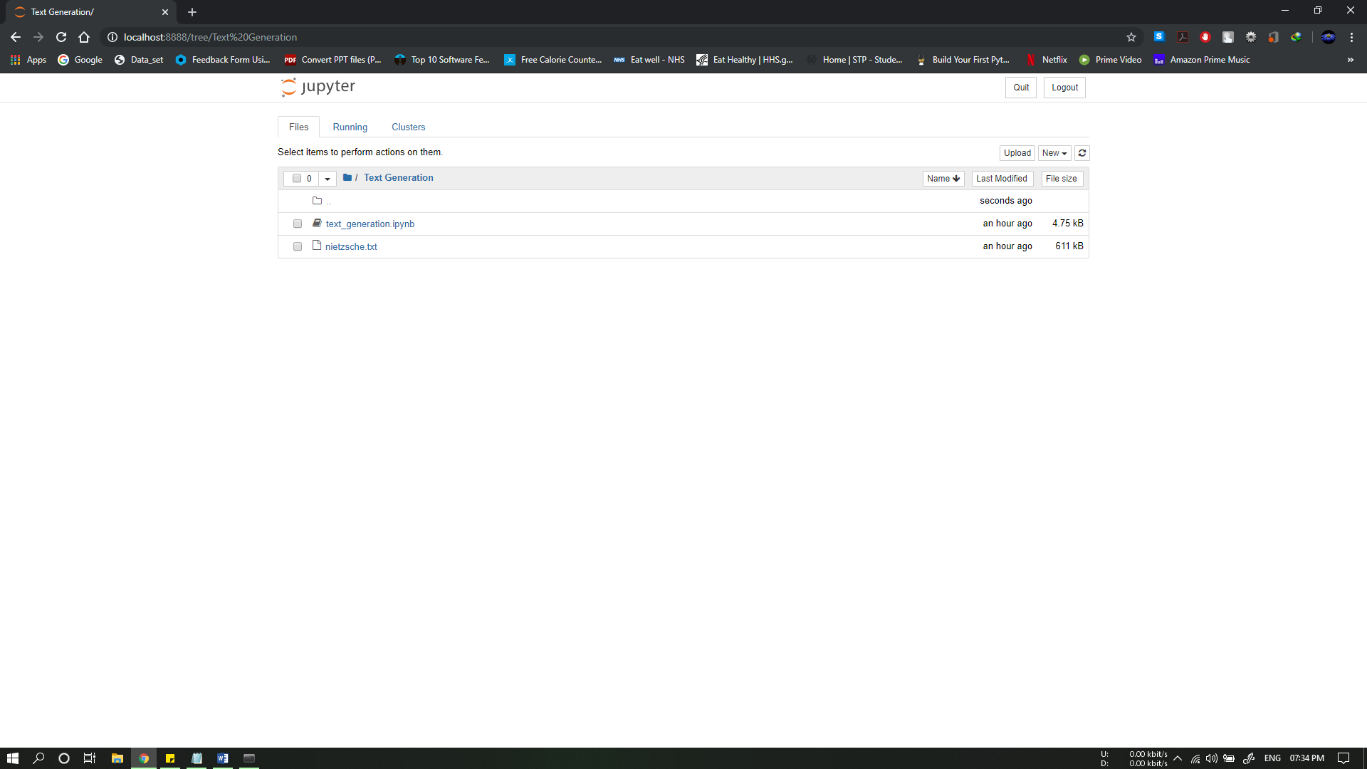
****

1. Installing required python libraries.

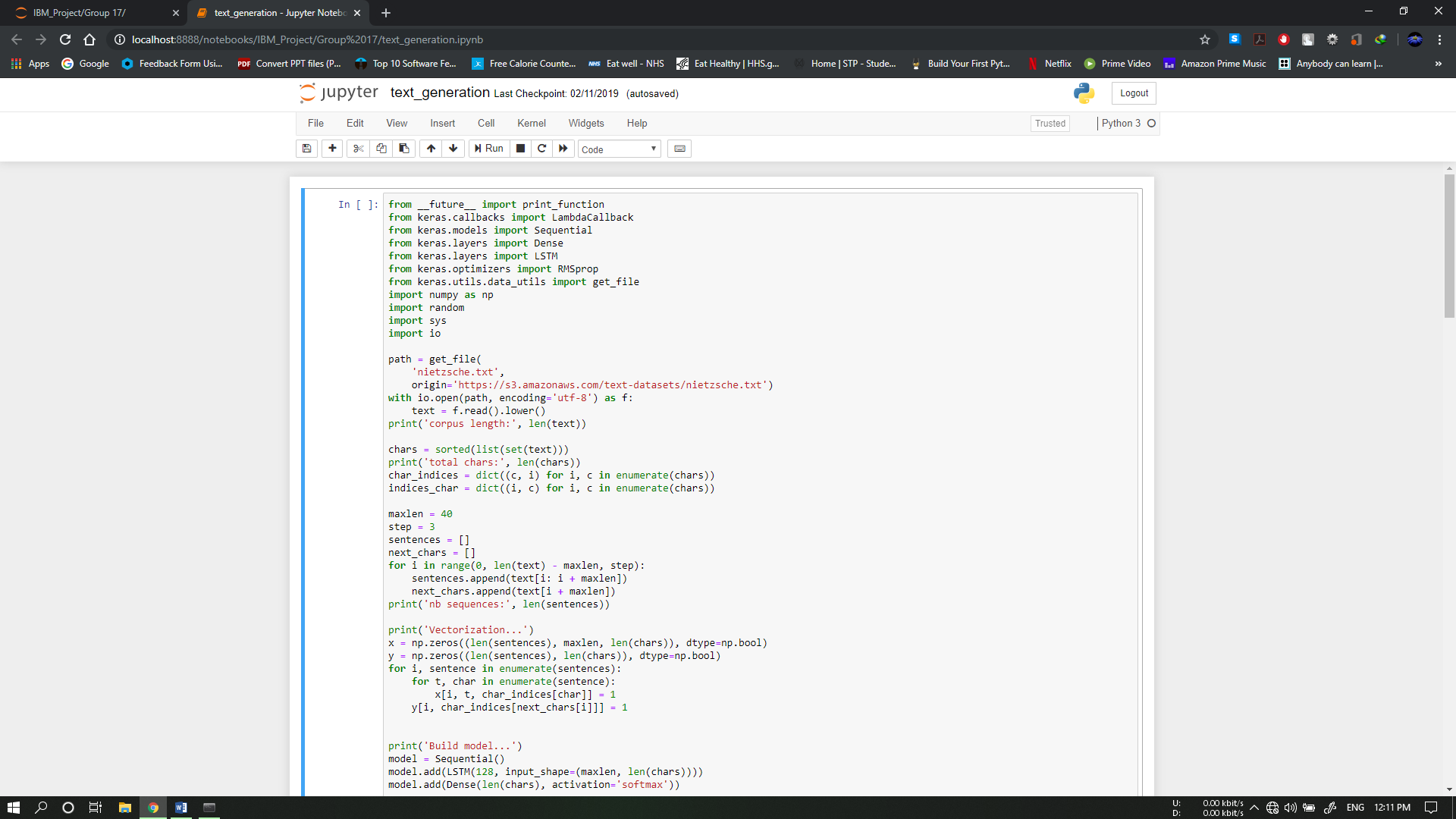
****

1. After installing the requisites, Keras and Tensorflow, we execute the notebook

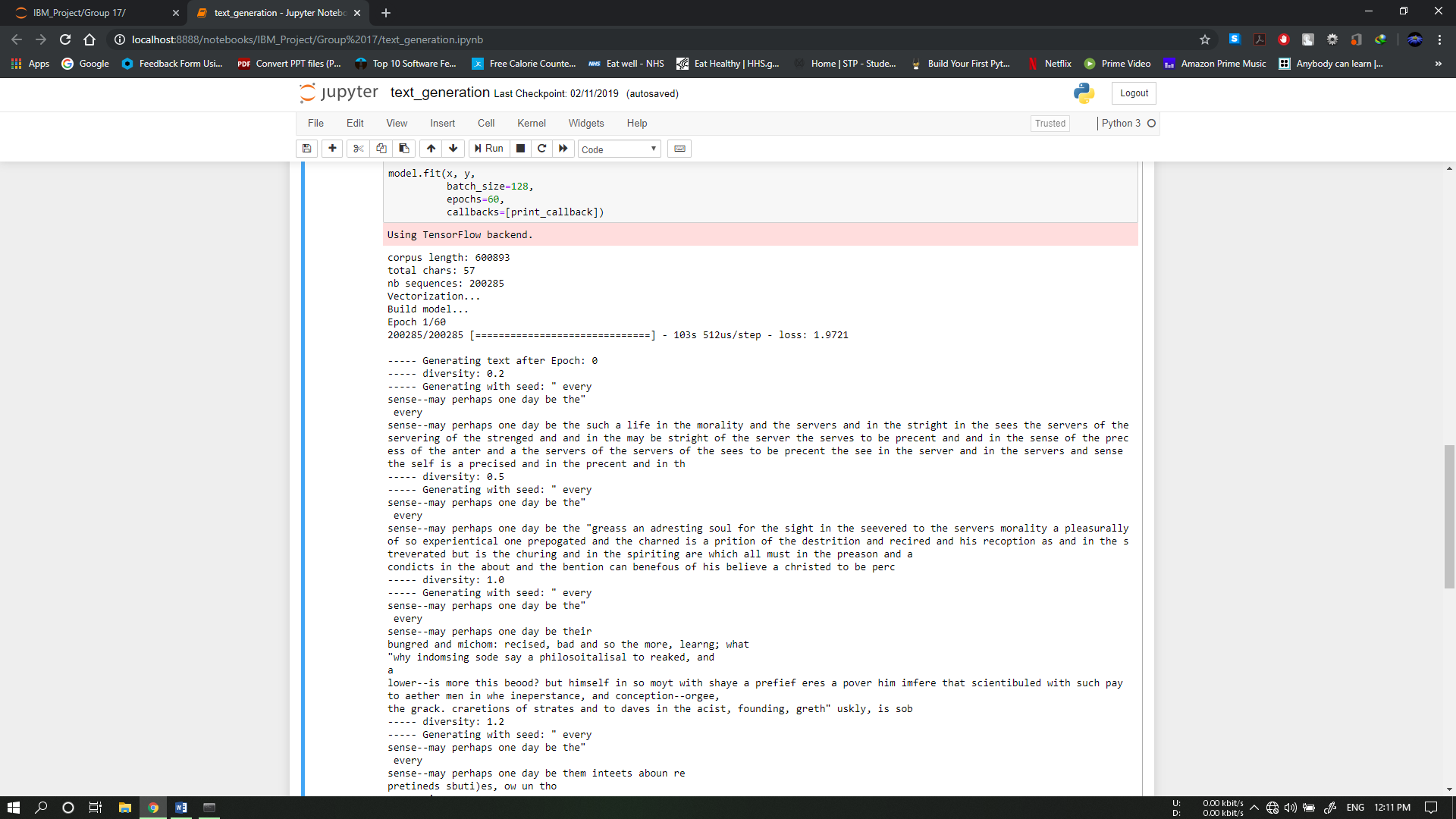
****

****

1. The training data is used to train a language model

****

1. New text is generated based on the model and returned as output.

****

**Scalability Scope of the Project:**

This model is relevant as text generation is increasingly in demand to solve translation, spell and review problems across several industries. In particular, fighting fake reviews plays a big role in this area of research. Fake reviews are a very real problem for companies such as Amazon and Yelp, who rely on genuine reviews from users to vouch for their products and businesses which are featured on their sites. As of writing this, it is very easy for businesses to pay for fake, positive reviews, which ultimately end up elevating their sales and revenue. It is equally as easy to generate negative reviews for competing businesses. Unfortunately this leads users to places and products fraudulently and can potentially lead to someone having a negative experience or worse. In order to combat these abuses and illegal activity, text generation can be used to detect what a review looks like when it is generated versus a genuine review written by an authentic user. This Code Pattern walks through the steps to create this text generation at a high level

**Conclusion:** Using Keras and Tensorflow, we have trained the language model using the data set and the output is generated based on the analysis.