VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELAGAVI - 590018

Assignment Report

on

EMOJI PREDICTION USING DEEP LEARNING

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by

Santhosh A – 1BG18CS422

Gishnu Govind – 1BG18CS406

Nayan Surya N – 1BG18CS414

Guide

Ms. Brinda

Assistant Professor., Dept. of CSE

BNMIT

Vidyayāmruthamashnuthe

B.N.M. Institute of Technology

Approved by AICTE, Affiliated to VTU, Accredited as grade A Institution by NAAC.

All UG branches – CSE, ECE, EEE, ISE & Mech.E accredited by NBA for academic years 2018-19 to 2020-21 & valid upto 30.06.2021

Post box no. 7087, 27th cross, 12th Main, Banashankari 2nd Stage, Bengaluru- 560070, INDIA

Ph: 91-80- 26711780/81/82 Email: principal@bnmit.in, www. bnmit.org

Department of Computer Science and Engineering

2020-21

B.N.M. Institute of Technology

Approved by AICTE, Affiliated to VTU, Accredited as grade A Institution by NAAC.
All UG branches – CSE, ECE, EEE, ISE & Mech.E accredited by NBA for academic years 2018-19 to 2020-21 & valid upto 30.06.2021
Post box no. 7087, 27th cross, 12th Main, Banashankari 2nd Stage, Bengaluru- 560070, INDIA
Ph: 91-80- 26711780/81/82 Email: principal@bnmit.in, www. bnmit.org

Department of Computer Science and Engineering

CERTIFICATE

Certified that the Mini Project entitled **Emoji Prediction using** carried out by **Santhosh A, Gishnu Govind & Nayan Surya N,** bonafide students of VII Semester B.E., **B.N.M Institute of Technology** in partial fulfillment for the Bachelor of Engineering in COMPUTER SCIENCE AND ENGINEERING of the **Visvesvaraya Technological University**, Belagavi during the year 2020-21. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report. The Project report has been approved as it satisfies the academic requirements in respect of Web Technology Laboratory with Mini Project prescribed for the said Degree.

Ms. Brinda
Assistant Professor
Department of CSE
BNMIT, Bengaluru

Dr. Sahana D. Gowda
Professor and HOD
Department of CSE
BNMIT, Bengaluru

Name & Signature

Examiner 1:

Examiner 2:

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project.

I would like to thank **Shri. Narayan Rao R Maanay**, Secretary, BNMEI, Bengaluru for providing excellent academic environment in the college.

I would like to sincerely thank **Prof. T.J Rama Murthy**, Director, BNMIT, Bengaluru for having extended his support and encouragement during the course of the work.

I would like to express my gratitude to **Prof. Eshwar N Maanay**, Dean, BNMIT, Bengaluru for his relentless support, guidance and assistance.

I would like to thank **Dr. Krishnamurthy G.N**, Principal, BNMIT, Bengaluru for his constant encouragement.

I would like to thank **Dr. Sahana D Gowda**, Professor and Head of the Department of Computer Science and Engineering who has shared her opinions and thoughts which helped me in giving my presentation successfully.

I would like to thank **Ms**. **Brinda**, Assistant Professor of Department of Computer Science and Engineering, BNMIT, Bengaluru for constant guidance's and support and helping us with all the essentials for the completion of this project.

Finally, I take this opportunity to extend my earnest gratitude and respect to my parents, teaching & non-teaching staffs of the department and all my friends, for giving me valuable advices and support at all times in all possible ways.

Santhosh A, Gishnu Govind &
Nayan Surya N

ABSTRACT

Besides alternative text-based forms, emojis became highly common in social media. Given their importance in daily communication, we tackled the problem of emoji prediction in Portuguese social media text. We created a dataset with occurrences of frequent emojis, used as labels, and then compared the performance of traditional machine learning algorithms with neural networks when predicting them. Either considering five or ten of the most popular emojis, an LSTM neural network clearly outperformed RNN in the latter task. At the same time, few studies make emoji recommendation based on Chinese corpus. This divides million-level Chinese micro-blog corpus into different context sets according to emojis, then propose a method to generate emoji-related features by analysis, finally a classification-based recommendation method is proposed by integrating these features. The experimental results show that the proposed method significantly improves the accuracy of emoji recommendation in social media platforms.

Table of Contents

1 INTRODUCTION

1.1 Overview

The whole concept of machine learning is figuring out ways in which we can teach a computer to perform a task without needing to provide explicit instructions. Another way to think about it is that we're trying to "program" intuition in a computer. You and I can look at an email and easily discern whether or not it's spam, but how do you get a computer to do such a task? You could construct a huge convoluted logic infrastructure of "if. then." statements to sort out the spam emails, but it would be a pain to construct and probably wouldn't work too well. Instead, the machine learning approach is to equip the computer with skills to learn on its own and feed it a bunch of examples. Machine learning is exploding as a field right now as people are realizing a multitude of tasks that we can teach computers to perform by feeding it large datasets.

1.2 Problem Statement

Emojis are the incredible way of expressing yourself. Therefore, our machines should also be aware of the appropriate emoji to be used at the right time. Thus, it does the same.
In this project, the model is predicting emoji's according to a given sentence using Deep Learning Models. Text data consisting of different sentences along with the labels. Labels represent a particular emoji representing the sentence.

1.3 Introduction to the machine learning model

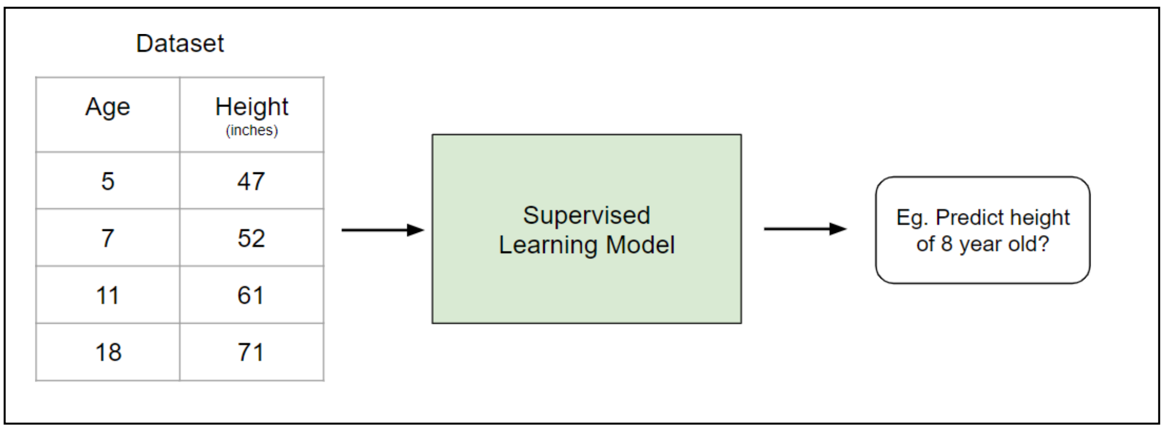
All machine learning models are categorized as either supervised or unsupervised. If the model is a supervised model, it's then sub-categorized as either a regression or classification model. We'll go over what these terms mean and the corresponding models that fall into each category below.

1.3.1 Supervised Learning

Supervised learning involves learning a function that maps an input to an output based on example input-output pairs.

For example, if I had a dataset with two variables, age (input) and height (output), I could implement a supervised learning model to predict the height of a person based on their age.

To re-iterate, within supervised learning, there are two sub-categories: regression and classification.

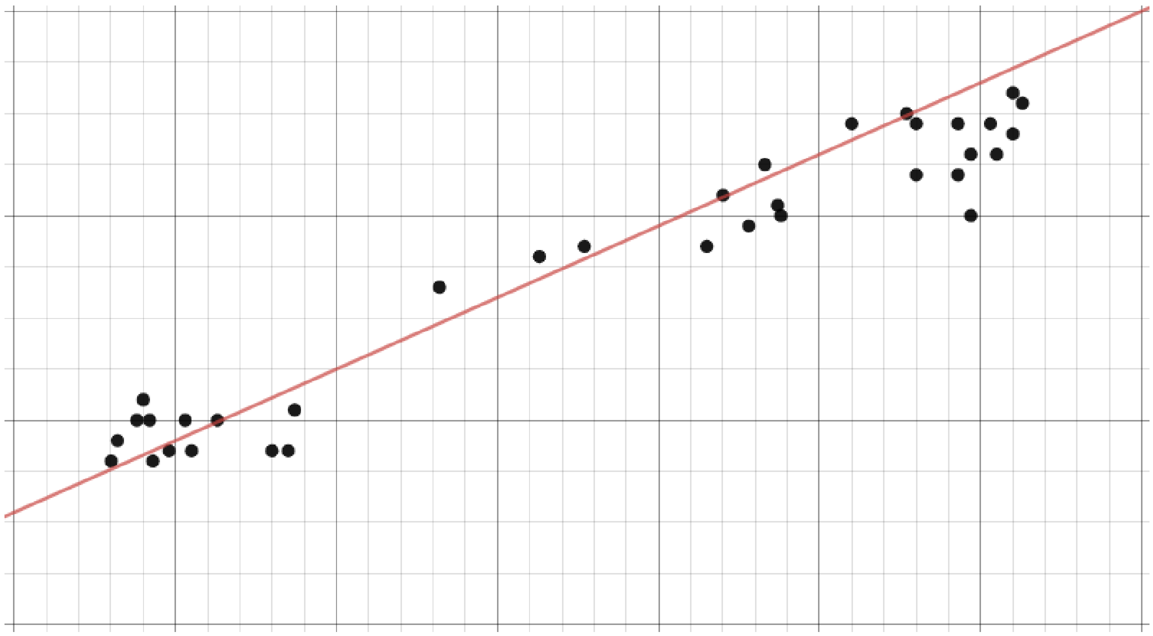


1.3.1.1 Regression

In regression models, the output is continuous. Below are some of the most common types of regression models.

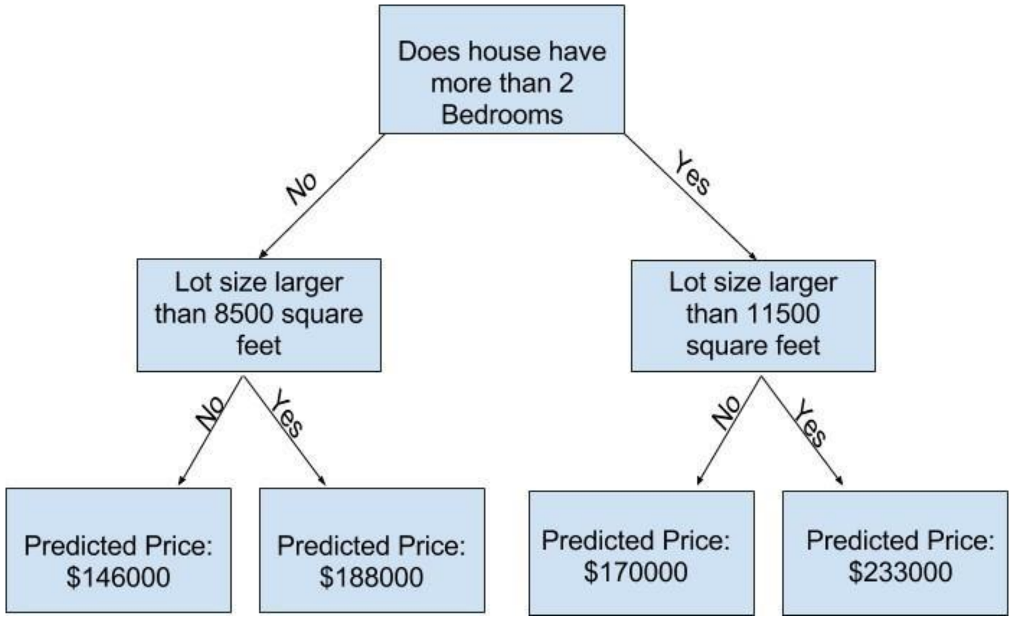
Linear Regression

The idea of linear regression is simply finding a line that best fits the data. Extensions of linear regression include multiple linear regression (e.g. finding a plane of best fit) and polynomial regression (e.g. finding a curve of best fit).



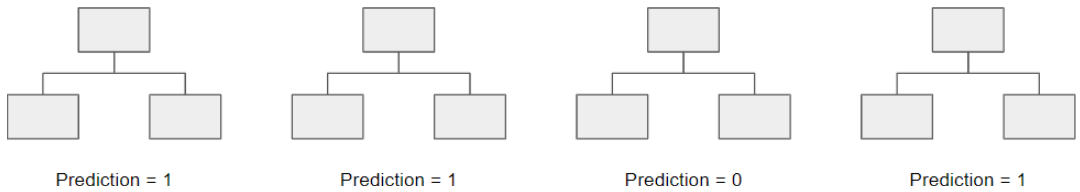
Decision Tree

Decision trees are a popular model, used in operations research, strategic planning, and machine learning. Each square above is called a node, and the more nodes you have, the more accurate your decision tree will be (generally). The last nodes of the decision tree, where a decision is made, are called the leaves of the tree. Decision trees are intuitive and easy to build but fall short when it comes to accuracy.



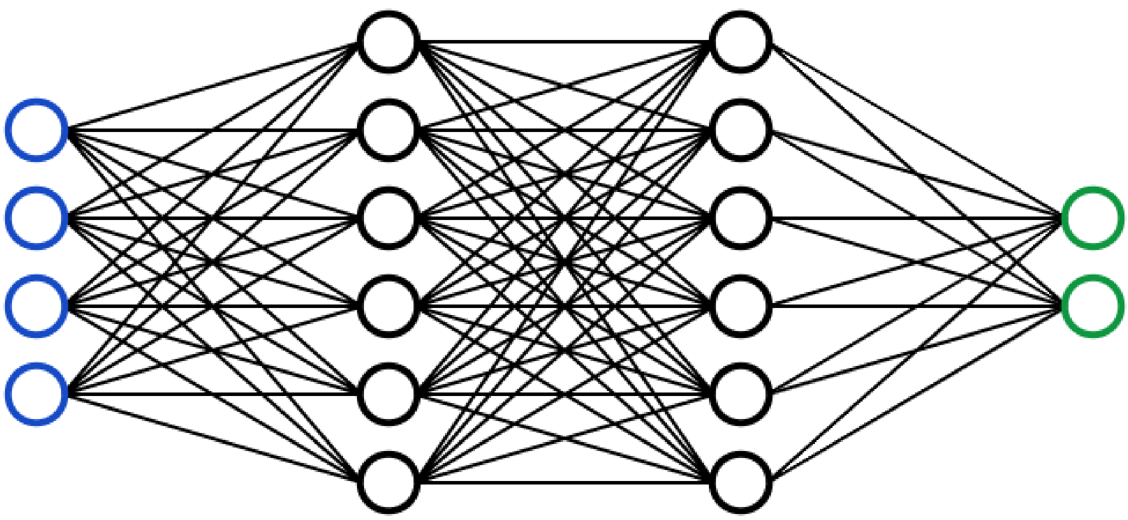
Random Forest

Random forests are an ensemble learning technique that builds off of decision trees. Random forests involve creating multiple decision trees using bootstrapped datasets of the original data and randomly selecting a subset of variables at each step of the decision tree. The model then selects the mode of all of the predictions of each decision tree. What's the point of this? By relying on a "majority wins" model, it reduces the risk of error from an individual tree.



For example, if we created one decision tree, the third one, it would predict 0. But if we relied on the mode of all 4 decision trees, the predicted value would be 1. This is the power of random forests.

Neural Network



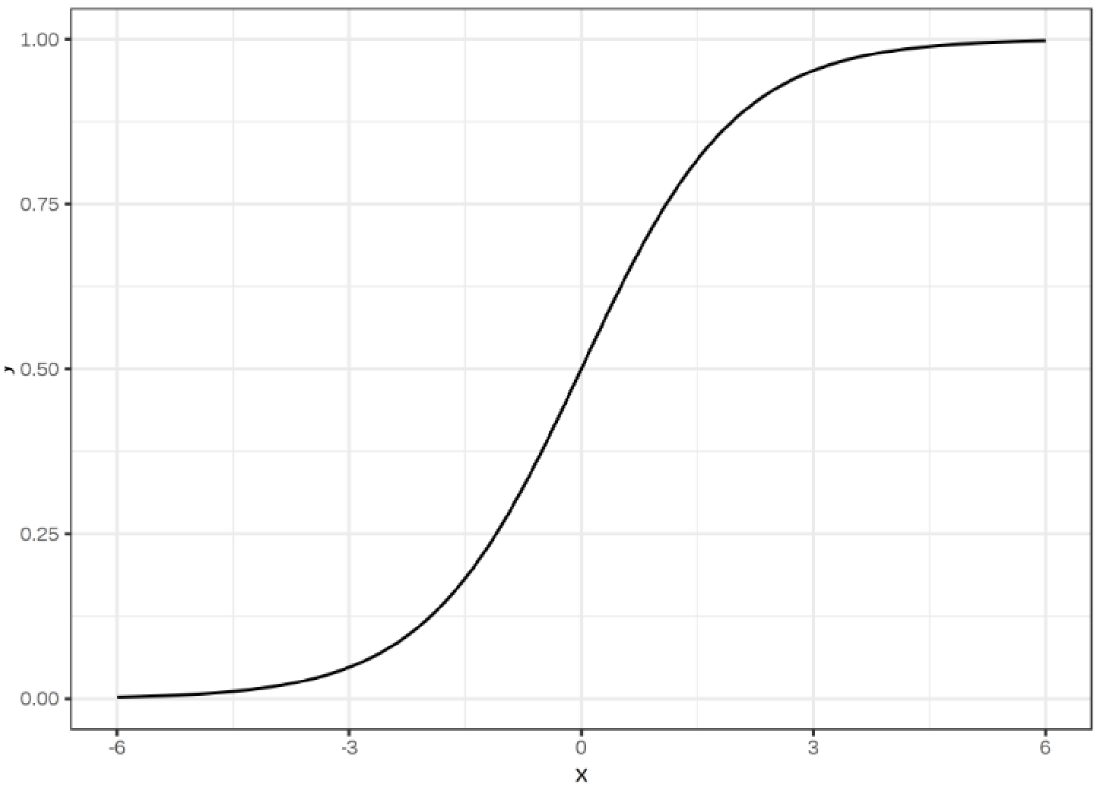
A Neural Network is essentially a network of mathematical equations. It takes one or more input variables, and by going through a network of equations, results in one or more output variables. You can also say that a neural network takes in a vector of inputs and returns a vector of outputs, but I won't get into matrices in this article.

1.3.1.2 Classification

In classification models, the output is discrete. Below are some of the most common types of classification models.

Logistic Regression

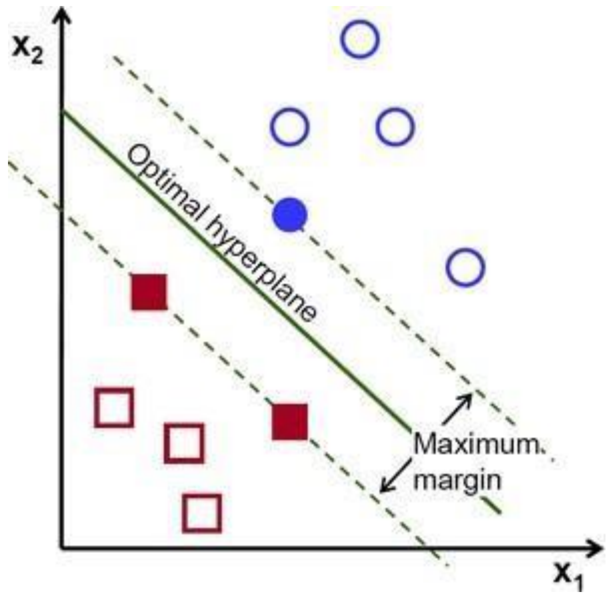
Logistic regression is similar to linear regression but is used to model the probability of a finite number of outcomes, typically two. There are a number of reasons why logistic regression is used over linear regression when modeling probabilities of outcomes (see here). In essence, a logistic equation is created in such a way that the output values can only be between 0 and 1 (see below).



Support Vector Machine

A Support Vector Machine is a supervised classification technique that can actually get pretty complicated but is pretty intuitive at the most fundamental level.

Let's assume that there are two classes of data. A support vector machine will find a hyperplane or a boundary between the two classes of data that maximizes the margin between the two classes (see below). There are many planes that can separate the two classes, but only one plane can maximize the margin or distance between the classes.



Naive Bayes

Naive Bayes is another popular classifier used in Data Science. The idea behind it is driven by Bayes Theorem:

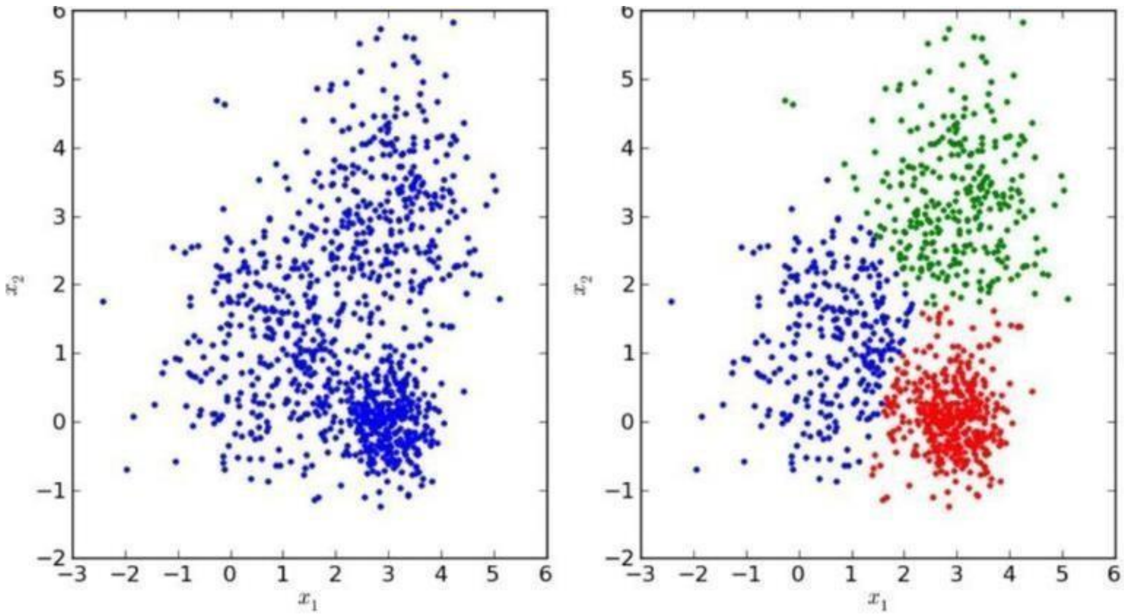
$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)}$$

In plain English, this equation is used to answer the following question. “What is the probability of y (my output variable) given X? And because of the naive assumption that variables are independent given the class, you can say that:

$$P(X|y) = P(x2|y) * P(x2|y) * \dots * P(xn|y)$$

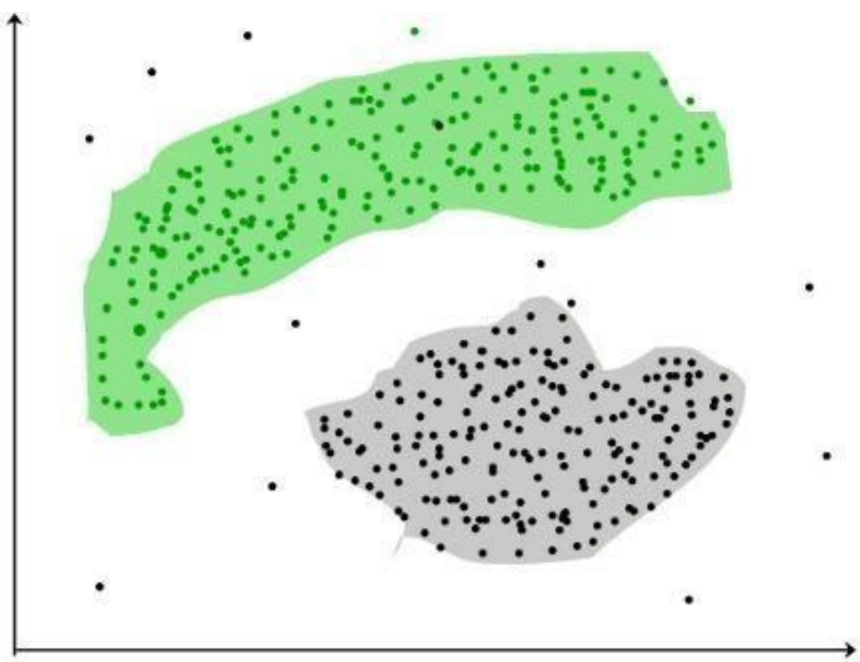
1.. Unsupervised Learning

Unlike supervised learning, unsupervised learning is used to draw inferences and find patterns from input data without references to labeled outcomes. Two main methods used in unsupervised learning include clustering and dimensionality reduction.



1.3.1.3 Clustering

Clustering is an unsupervised technique that involves the grouping, or clustering, of data points. It's frequently used for customer segmentation, fraud detection, and document classification.



2 SYSTEM REQUIREMENTS

2.1 Hardware and Software Requirements

2.1.1 Hardware Requirements

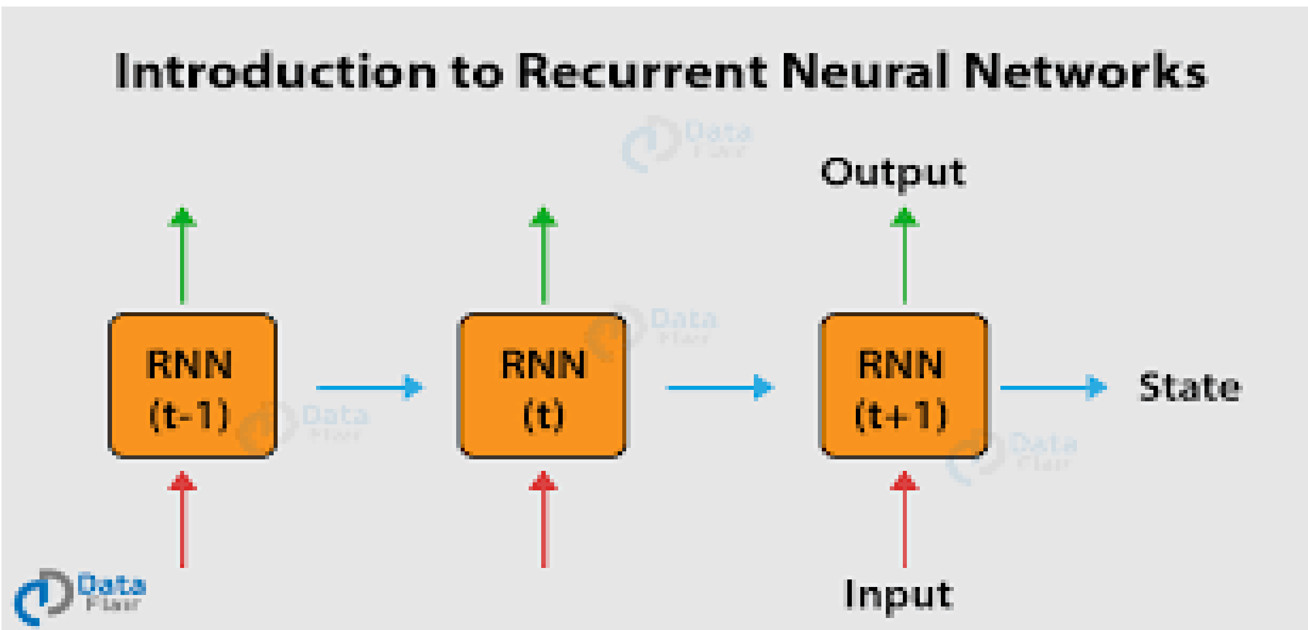
- Processor – Intel Xeon E2630 v4 – 10 core processor, 2.2 GHz with Turboboost upto 3.1 GHz. 25 MB Cache
- RAM – 8 GB
- TB Hard Disk (7200 RPM) + 512 GB SSD
- GPU – NVidia TitanX Pascal (12 GB VRAM)
- Intel Heatsink to keep temperature under control

2.1.2 Software Requirements

- Any suitable python IDE
- NumPy
- Pandas
- Sklearn

3 RNN

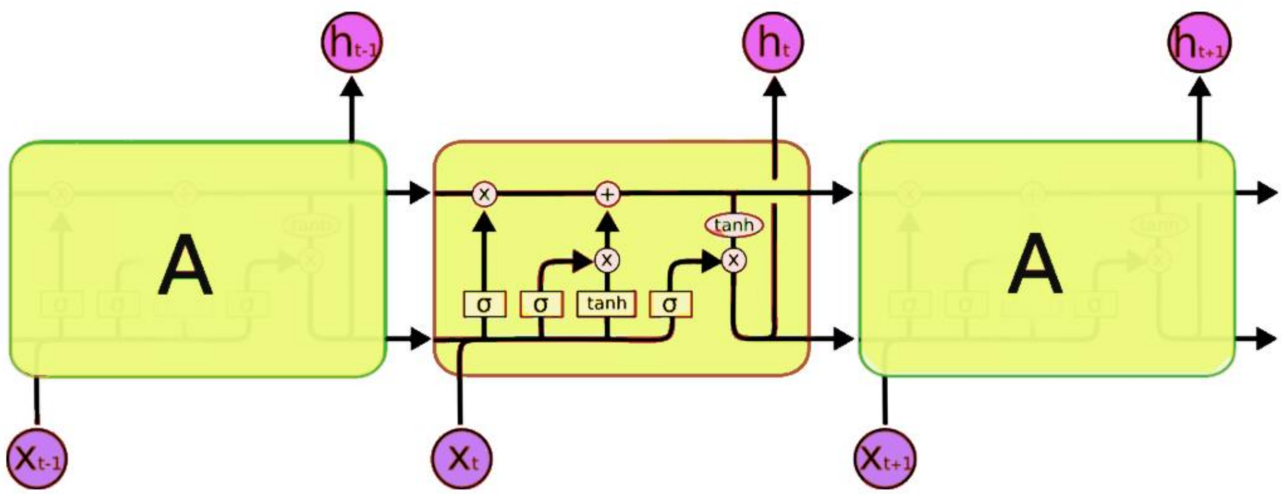
A **recurrent neural network** (**RNN**) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition[4] or speech recognition.



The term "recurrent neural network" is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both classes of networks exhibit temporal dynamic behavior.[7] A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feedforward neural network, while an infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled.

LSTM

Long short-term memory (**LSTM**) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems).



LSTM also improved large-vocabulary speech recognition and text-to-speech synthesis and was used in Google Android. In 2015, Google's speech recognition reportedly experienced a dramatic performance jump of 49% through CTC-trained LSTM.

LSTM broke records for improved machine translation, Language Modeling and Multilingual Language Processing. LSTM combined with convolutional neural networks (CNNs) improved automatic image captioning. Given the computation and memory overheads of running LSTMs, there have been efforts on accelerating LSTM using hardware accelerators.

THE LSTM ALGORITHM

- Prepare the data
- Feature Scaling (Preprocessing of data)
- Split the dataset for train and test
- Converting features into NumPy array and reshaping the array into shape accepted by LSTM model
- Build the architecture for LSTM network
- Compile and fit the model (Training)
- Evaluate the performance of model (Test)

Building the LSTM

In order to build the LSTM, we need to import a couple of modules from Keras.

1. Sequential for initializing the neural network.
2. Dense for adding a densely connected neural network layer.
3. LSTM for adding the Long Short-Term Memory layer.
4. Dropout for adding dropout layers that prevent overfitting.

Summary:

We can say that, when we move from RNN to LSTM (Long Short-Term Memory), we are introducing more & more controlling knobs, which control the flow and mixing of Inputs as per trained Weights. So, LSTM gives us the most Control-ability and thus, Better Results. But also comes with more Complexity and Operating Cost.

4 IMPLEMENTATION

4.1 Module Description

The following modules were used to prepare the KNN model:

- **Pandas**

Pandas is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.

- **NumPy**

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- **Sklearn**

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms.

4.2 Source Code


```
{
 "nbformat": 4,
 "nbformat_minor": 0,
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.7.6"
     },
```

```
   "colab": {
    "name": "Emoji Prediction.ipynb",
    "provenance": [],
    "include_colab_link": true
   }
  },
  "cells": [
   {
    "cell_type": "markdown",
    "metadata": {
     "id": "view-in-github",
     "colab_type": "text"
    },
    "source": [
     "<a href=\"https://colab.research.google.com/github/bhargav-joshi/Emoji-
Prediction/blob/master/Emoji_Prediction.ipynb\" target=\"_parent\"><img
src=\"https://colab.research.google.com/assets/colab-badge.svg\" alt=\"Open In Colab\"/></a>"
    ]
   },
   {
    "cell_type": "code",
    "metadata": {
     "id": "qgmc-lS1PDBZ",
     "colab_type": "code",
     "colab": {},
     "outputId": "70fba510-6092-447a-a167-6fafd2673814"
    },
    "source": [
     "pip install emoji"
    ],
    "execution_count": null,
    "outputs": [
     {
      "output_type": "stream",
      "text": [
      ],
      "name": "stdout"
     }
    ]
   },
   {
    "cell_type": "code",
    "metadata": {
     "id": "Osw-IDDaPDBe",
     "colab_type": "code",
     "colab": {}
    },
    "source": [
     "import emoji\n",
     "import pandas as pd\n",
     "import numpy as np"
```

],
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"metadata": {
"id": "uNuB-v33PDBg",
"colab_type": "code",
"colab": {},
"outputId": "8ea0703b-85bf-41b3-d660-40921857fca9"
},
"source": [
"emoji.EMOJI_UNICODE"
],
"execution_count": null,
"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"{':1st_place_medal:': 'ðŸ¥‡',\n",
" ':2nd_place_medal:': 'ðŸ¥ˆ',\n",
" ':3rd_place_medal:': 'ðŸ¥‰',\n",
" ':AB_button_(blood_type):': 'ðŸ†Ž',\n",
" ':ATM_sign:': 'ðŸ• §',\n",
" ':A_button_(blood_type):': 'ðŸ…°',\n",
" ':Afghanistan:': 'ðŸ‡¦ðŸ‡«',\n",
" ':Albania:': 'ðŸ‡¦ðŸ‡±',\n",
" ':Algeria:': 'ðŸ‡©ðŸ‡¿',\n",
" ':American_Samoa:': 'ðŸ‡¦ðŸ‡¸',\n",
" ':Andorra:': 'ðŸ‡¦ðŸ‡©',\n",
" ':Angola:': 'ðŸ‡¦ðŸ‡´',\n",
" ':Anguilla:': 'ðŸ‡¦ðŸ‡®',\n",
" ':Antarctica:': 'ðŸ‡¦ðŸ‡¶',\n",
" ':Antigua_&_Barbuda:': 'ðŸ‡¦ðŸ‡¬',\n",
" ':Aquarius:': 'â™’',\n",
" ':Argentina:': 'ðŸ‡¦ðŸ‡·',\n",
" ':Aries:': 'â™ˆ',\n",
" ':Armenia:': 'ðŸ‡¦ðŸ‡²',\n",
" ':Aruba:': 'ðŸ‡¦ðŸ‡¼',\n",
" ':Ascension_Island:': 'ðŸ‡¦ðŸ‡¨',\n",
" ':Australia:': 'ðŸ‡¦ðŸ‡º',\n",
" ':Austria:': 'ðŸ‡¦ðŸ‡¹',\n",
" ':Azerbaijan:': 'ðŸ‡¦ðŸ‡¿',\n",
" ':BACK_arrow:': 'ðŸ”™',\n",
" ':B_button_(blood_type):': 'ðŸ…±',\n",
" ':Bahamas:': 'ðŸ‡§ðŸ‡¸',\n",
" ':Bahrain:': 'ðŸ‡§ðŸ‡',\n",
" ':Bangladesh:': 'ðŸ‡§ðŸ‡©',\n",
" ':Barbados:': 'ðŸ‡§ðŸ‡§',\n",

```
                  " ':Belarus:': 'ðŸ‡§ðŸ‡¾',\n",
"outputs": [
 {
  "output_type": "stream",
  "text": [
   "I want to eat\n",
   "ðŸ• ´\n",
   "he did not answer\n",
   "ðŸ˜“\n",
   "he got a raise\n",
   "ðŸ˜“\n",
   "she got me a present\n",
   "ðŸ˜• \n",
   "ha ha ha it was so funny\n",
   "ðŸ˜• \n",
   "he is a good friend\n",
   "ðŸ˜• \n",
   "I am upset\n",
   "ðŸ˜“\n",
   "We had such a lovely dinner tonight\n",
   "ðŸ˜• \n",
   "where is the food\n",
   "ðŸ• ´\n",
   "Stop making this joke ha ha ha\n",
   "ðŸ˜• \n",
   "where is the ball\n",
   "âš¾\n",
   "work is hard\n",
   "ðŸ˜• \n",
   "This girl is messing with me\n",
   "â• ¤ï¸• \n",
   "are you serious ha ha\n",
   "â• ¤ï¸• \n",
   "Let us go play baseball\n",
   "âš¾\n",
   "This stupid grader is not working\n",
   "ðŸ˜“\n",
   "work is horrible\n",
   "ðŸ˜“\n",
   "Congratulation for having a baby\n",
   "ðŸ˜• \n",
   "stop messing around\n",
   "ðŸ• ´\n",
   "any suggestions for dinner\n",
   "ðŸ• ´\n",
   "I love taking breaks\n",
   "â• ¤ï¸• \n",
  "you brighten my day\n",
  "â• ¤ï¸• \n",
  "I boiled rice\n",
  "ðŸ• ´\n",
```

"she is a bully\n",
"â• ¤ï¸• \n",
"Why are you feeling bad\n",
"ðŸ˜“\n",
"I am upset\n",
"ðŸ˜“\n",
"I worked during my birthday\n",
"ðŸ˜• \n",
"My grandmother is the love of my life\n",
"â• ¤ï¸• \n",
"enjoy your break\n",
"âš¾\n",
"valentine day is near\n",
"ðŸ˜• \n",
"I miss you so much\n",
"â• ¤ï¸• \n",
"throw the ball\n",
"âš¾\n",
"My life is so boring\n",
"â• ¤ï¸• \n",
"she said yes\n",
"ðŸ˜• \n",
"will you be my valentine\n",
"â• ¤ï¸• \n",
"he can pitch really well\n",
"âš¾\n",
"dance with me\n",
"ðŸ˜• \n",
"I am starving\n",
"ðŸ˜“\n",
"See you at the restaurant\n",
"ðŸ• ´\n",
"I like to laugh\n",
"ðŸ˜• \n",
"I will go dance\n",
"ðŸ˜• \n",
"I like your jacket\n",
"â• ¤ï¸• \n",
"i miss her\n",
"â• ¤ï¸• \n",
"what is your favorite baseball game\n",
"âš¾\n",
"Good job\n",
"ðŸ˜• \n",
"I love to the stars and back\n",
"ðŸ˜• \n",
"What you did was awesome\n",
"ðŸ˜“\n",
"ha ha ha lol\n",
"ðŸ˜• \n",
"I want to joke\n",

"ðŸ˜“\n",
"go away\n",
"âš¾\n",
"yesterday we lost again\n",
"ðŸ˜“\n",
"family is all I have\n",
"ðŸ˜• \n",
"you are failing this exercise\n",
"ðŸ˜“\n",
"Good joke\n",
"ðŸ˜• \n",
"You totally deserve this prize\n",
"ðŸ˜“\n",
"I did not have breakfast\n",
"ðŸ• ´\n"
],
"name": "stdout"
}
]
},

"source": [
"for w in x_query.split():\n",
"   print(w)\n",
"   emb = embedding_index[w]\n",
"   embedding_x_query.append(emb)"
],
"execution_count": null,
"outputs": [
{
"output_type": "stream",
"text": [
"i\n",
"like\n",
"data\n",
"science\n"
],
"name": "stdout"
}
]
},
{
"cell_type": "code",
"metadata": {
"id": "GtHc5VD1PDC5",
"colab_type": "code",
"colab": {}
},
"source": [
"embedding_x_query = np.array(embedding_x_query)"
],

```
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "metadata": {
   "id": "MaVGZNT3PDC6",
   "colab_type": "code",
   "colab": {},
   "outputId": "59899f75-1ca5-4538-df3b-88cbdd3bd1e9"
  },
  "source": [
   "embedding_x_query.shape"
  ],
  "execution_count": null,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "(4, 50)"
     ]
    },
    "metadata": {
     "tags": []
    },
    "execution_count": 88
   }
  ]
 },
 {
  "cell_type": "code",
  "metadata": {
   "id": "kyfwLvtmPDC8",
   "colab_type": "code",
   "colab": {}
  },
  "source": [
   "if embedding_x_query.shape[0]:\n",
   "    embedding_x_query = np.vstack((embedding_x_query , np.zeros((10 - len(x_query.split()), 50))))"
  ],
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "metadata": {
   "id": "25wqZgxWPDC-",
   "colab_type": "code",
   "colab": {}
  },
```

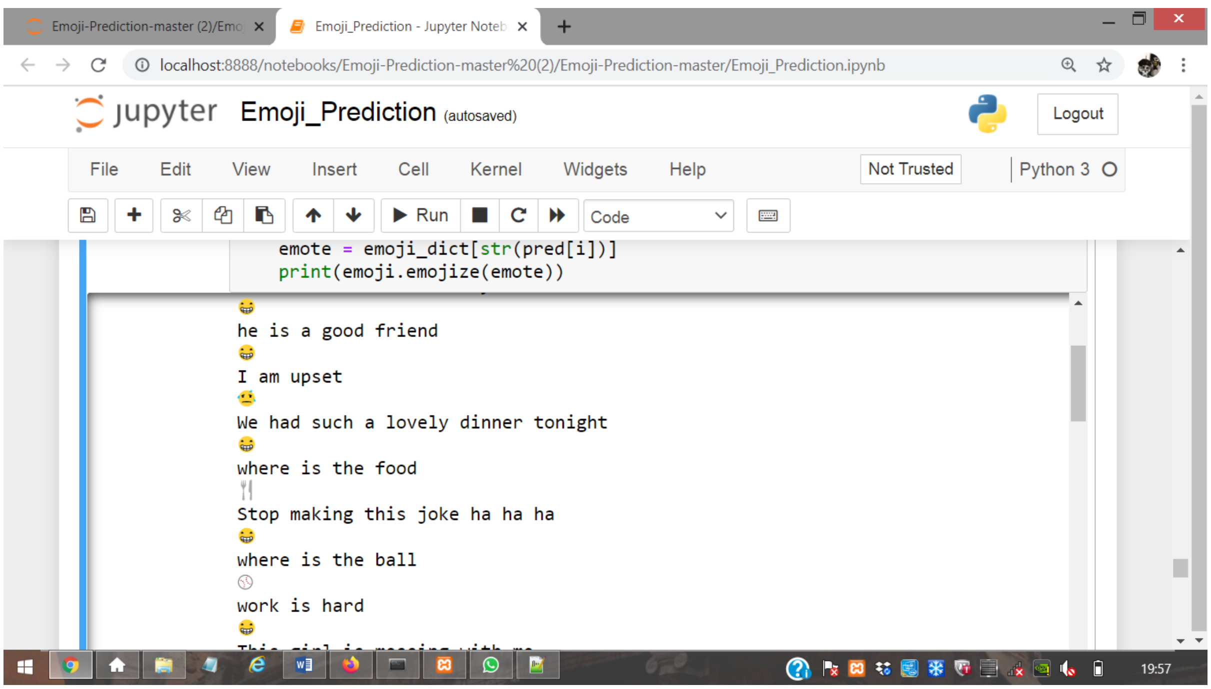
"source": [
"embedding_x_query = embedding_x_query.reshape(1, 10, 50)"
],
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"metadata": {
"id": "2qF4gXNjPDDA",
"colab_type": "code",
"colab": {},
"outputId": "62a70d35-5984-4b58-8f4a-f747cbb40988"
},
"source": [
"embedding_x_query.shape"
],
"execution_count": null,
"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"(1, 10, 50)"
]
},
"metadata": {
"tags": []
},
"execution_count": 91
}
]
},
{
"cell_type": "code",
"metadata": {
"id": "O0jiDOh7PDDC",
"colab_type": "code",
"colab": {}
},
"source": [
"p = model.predict_classes(embedding_x_query)"
],
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"metadata": {
"id": "1yIxCcRSPDDF",
"colab_type": "code",

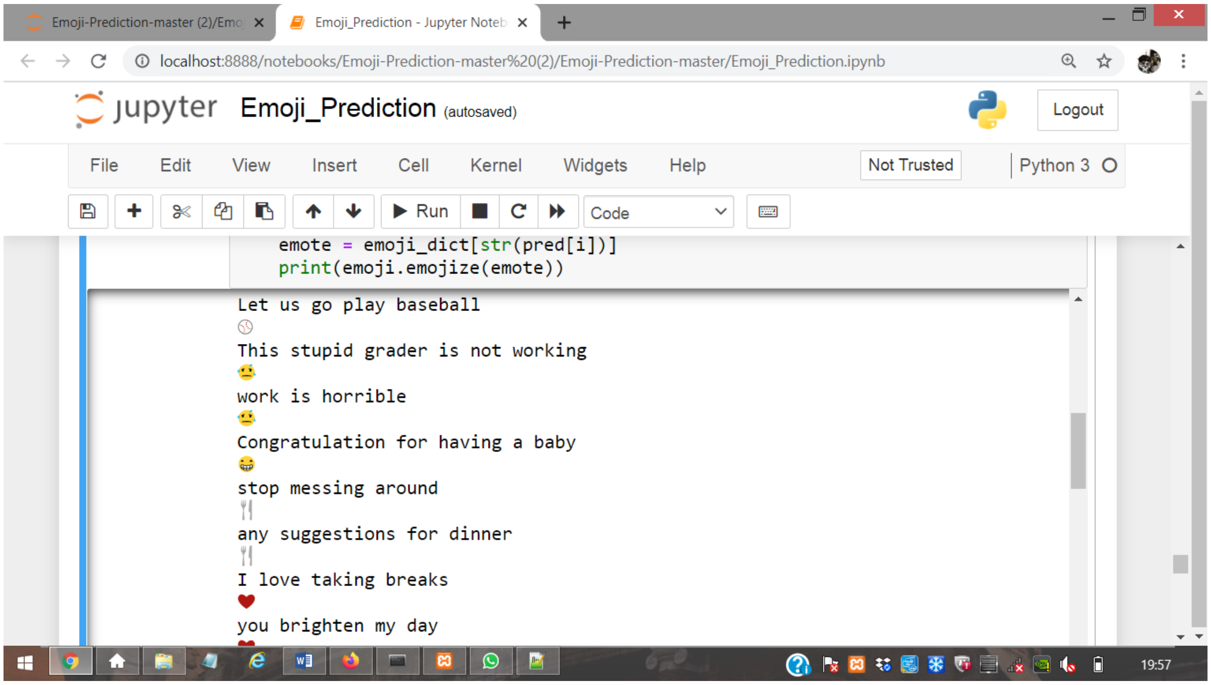
```
        "colab": {},
        "outputId": "829dee4b-41af-4fb0-dd24-c28f914c4c9d"
      },
      "source": [
        "print(emoji.emojize(emoji_dict[str(p[0])]))"
      ],
      "execution_count": null,
      "outputs": [
        {
          "output_type": "stream",
          "text": [
            "ðŸ˜• \n"
          ],
          "name": "stdout"
        }
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "nDuVSzDIPDDI",
        "colab_type": "text"
      },
      "source": [
        "### End !!"
      ]
    }
  ]
}
```

4.3 Accuracy of the model

LSTM worked better than RNN i.e., accuracy of the model is more by using LSTM than RNN. When we move from RNN to LSTM, we are introducing more & more controlling knobs, which control the flow and mixing of Inputs as per trained Weights. And thus, bringing in more flexibility in controlling the outputs. So, LSTM gives us the most Control-ability and thus, Better Results. But also comes with more Complexity and Operating Cost.

5 RESULTS





Emoji-Prediction-master (2)/Emo
Emoji_Prediction - Jupyter Noteb
localhost:8888/notebooks/Emoji-Prediction-master%20(2)/Emoji-Prediction-master/Emoji_Prediction.ipynb
jupyter Emoji_Prediction (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
emote = emoji_dict[str(pred[i])]
print(emoji.emojize(emote))
Let us go play baseball
This stupid grader is not working
work is horrible
Congratulation for having a baby
stop messing around
any suggestions for dinner
I love taking breaks
you brighten my day
19:57

6 CONCLUSION

When compared to RNN, LSTM algorithm which is the artificial neural of RNN's is more efficient, because accuracy of the model is more. When we move from RNN to LSTM, we are introducing more & more controlling knobs, which control the flow and mixing of Inputs as per trained Weights. And thus, bringing in more flexibility in controlling the outputs. So, LSTM gives us the most Control-ability and thus, Better Results with more reliability in predicting the emojis. We train several models based on Long Short-Term Memory networks (LSTMs) in this task. Our experimental results show that our neural model outperforms two baselines as well as humans solving the same task, suggesting that computational models are able to better capture the underlying semantics of emojis.

7 FUTURE ENHANCEMENT

We are still at an infant state in the field of Machine Learning. There are a lot of advancements to achieve in this field. One of them that will take Machine Learning to the next level is Quantum Computing. It is a type of computing that uses the mechanical phenomena of quantum such as entanglement and superposition. By using the quantum phenomenon of superposition, we can create systems (quantum systems) that can exhibit multiple states at the same time. On the other hand, entanglement is the phenomenon where two different states can be referenced to each other. It helps in describing the correlation between the properties of a quantum system.