Notes from online courses

GenAI course with Langchain and Hugging Face

Siddharth Gehlot

Table of Contents

[Overview 1](#_Toc184056607)

[Getting Started 1](#_Toc184056608)

[Different ways of creating python environment 3](#_Toc184056609)

[3 Different ways to create python environment – 3](#_Toc184056610)

[Install requirements 6](#_Toc184056611)

[Streamlit with python 6](#_Toc184056612)

[ML APP with streamlit 10](#_Toc184056613)

[Machine Learning for NLP 12](#_Toc184056614)

[High-level Overview 12](#_Toc184056615)

[Practical Use case of NLP 12](#_Toc184056616)

[Tokenization and Basic Terminologies 13](#_Toc184056617)

## Overview

To get started right away, just tap any placeholder text (such as this) and start typing.

### Getting Started

* VS code

Create environment

*conda create -p venv python==3.12*

**

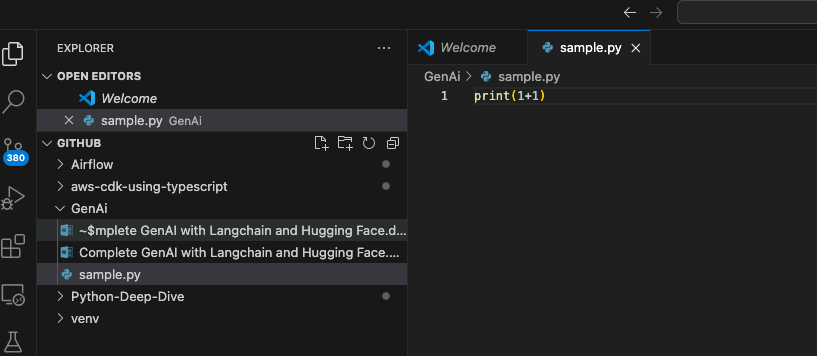
Activate environment

venv/ folder

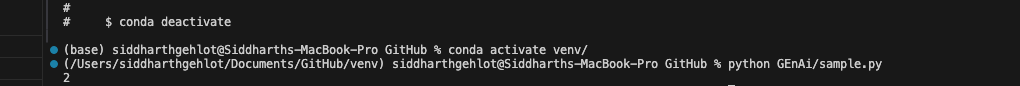
conda activate venv/

This steps allows us to run the python files from python virtual environment.

Sample file to test –



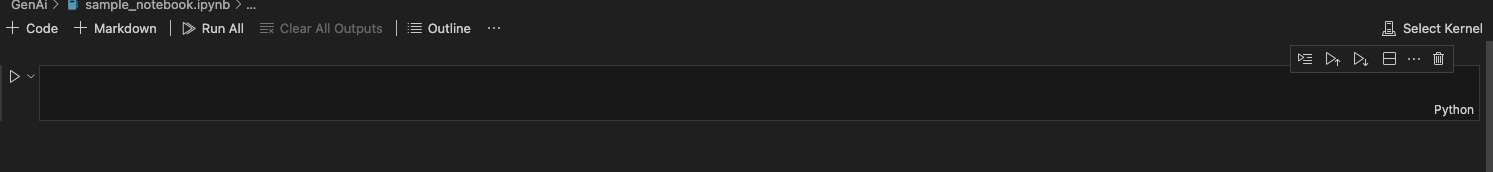
Running the sample file



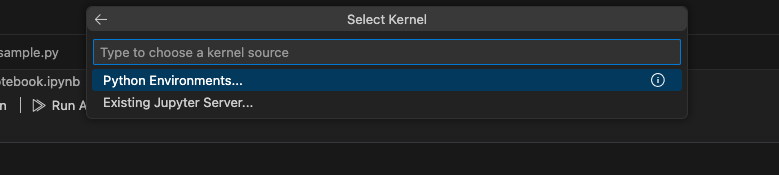
* To run a file from same venv environment.

Create a sample notebook file.

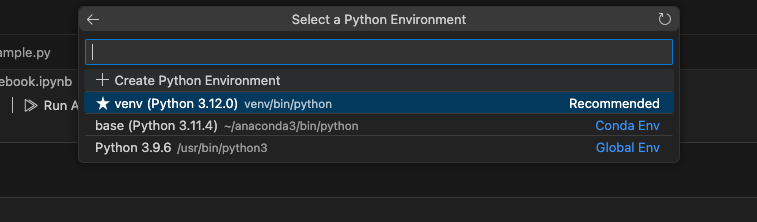
Click on select kernel/Detect kernel as in screen shot below



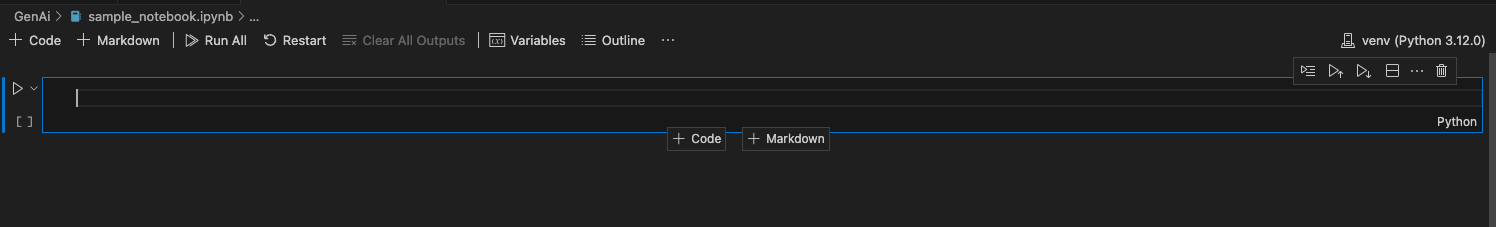
It will pop up following options



Click on Python environment and then venv



Now the notebook will run on given kernel



To run commands on kernel install ipykernel package.

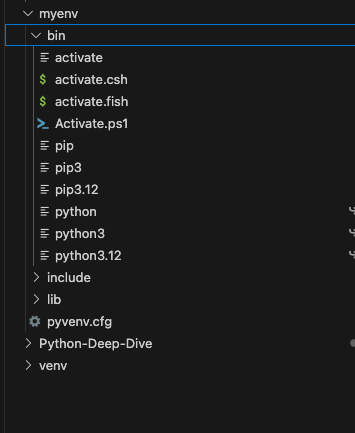
## Different ways of creating python environment

### 3 Different ways to create python environment –

1. Python command(For this approach we need to have python installed)

python -m venv myenv

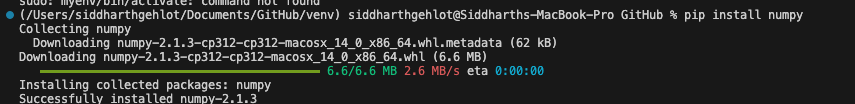
It will create myenv folder as below



Now activate the environment

sudo myenv/bin/activate

and commands can be executed in given virtual environment as below

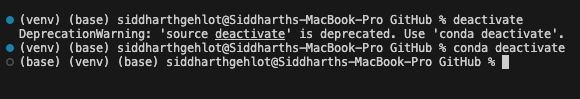


If need the environment with other version, then we need to upsgrade the version.

TO deactivate the env

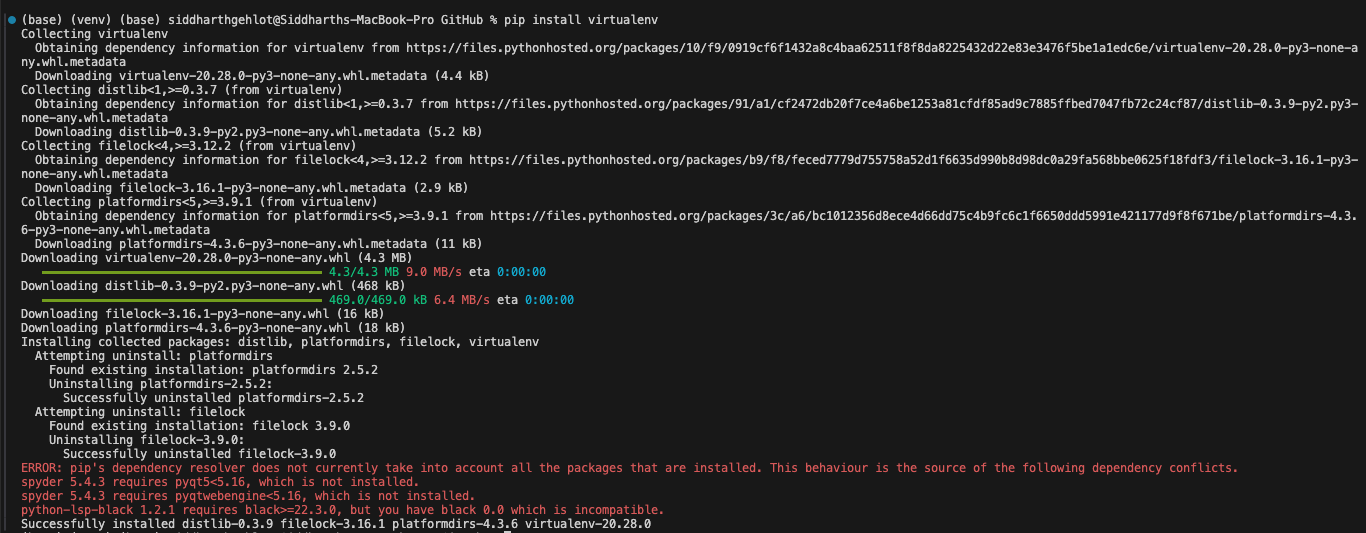
Command is

conda deactivate



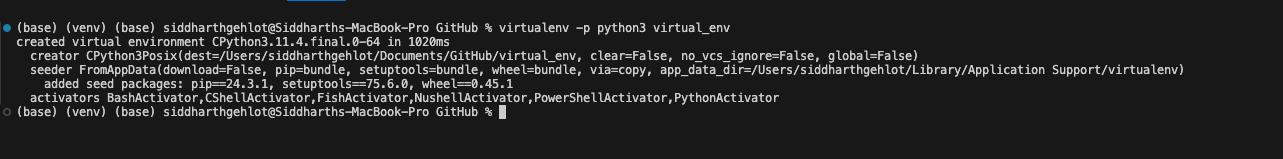
1. Linux command approach for creating environment

pip install virtualenv



Create virtual environment using command

virtualenv -p python3 virtual\_env



Activate environment

sudo virtual\_env\bin\activate

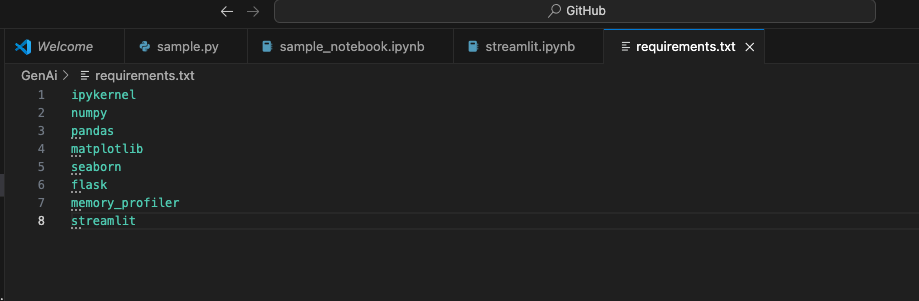
1. Conda create to create environment

This approach is already covered in section above

## Install requirements

pip install -r GenAi/requirements.txt

Requriements.txt will be as below –

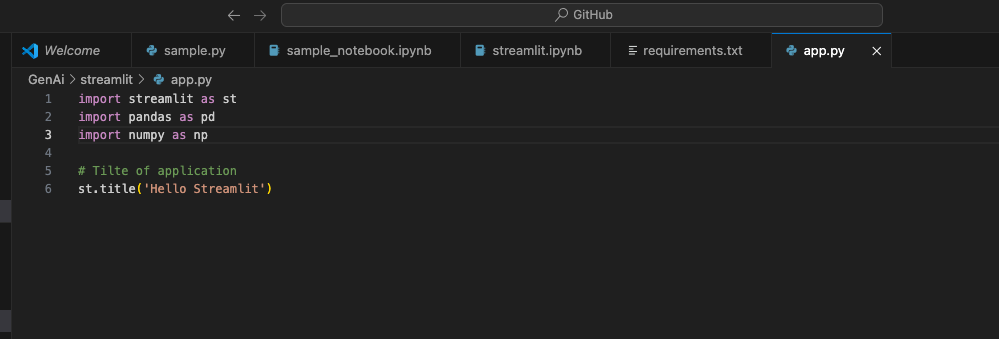


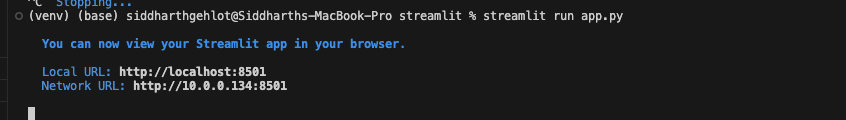
## Streamlit with python

Streamlit is an open-source app framework for machine learning and Data science projects. It allows you to create beautiful web applications for your machine learning and data science projects with simple Python scripts.

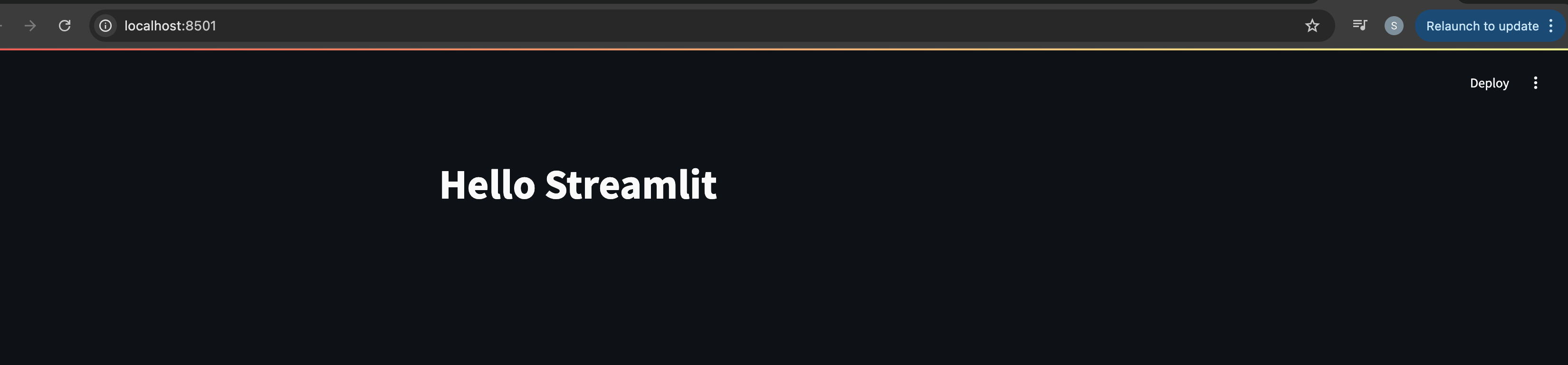
* Create a simple Streamlit app then to call streamlit app(app.py) use command

streamlit run app.py



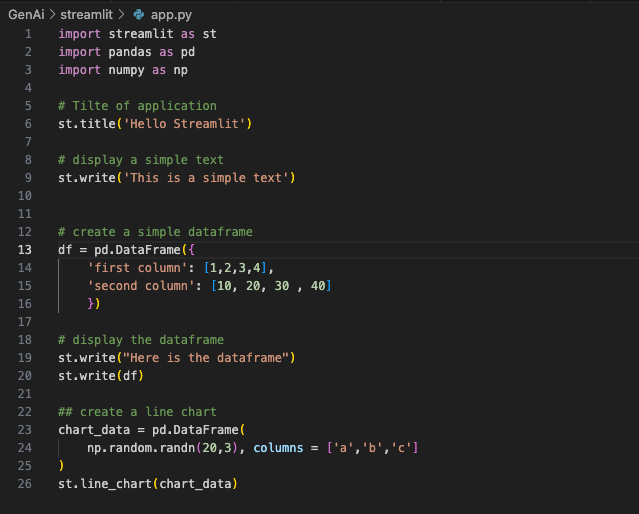


Web application will be as follows

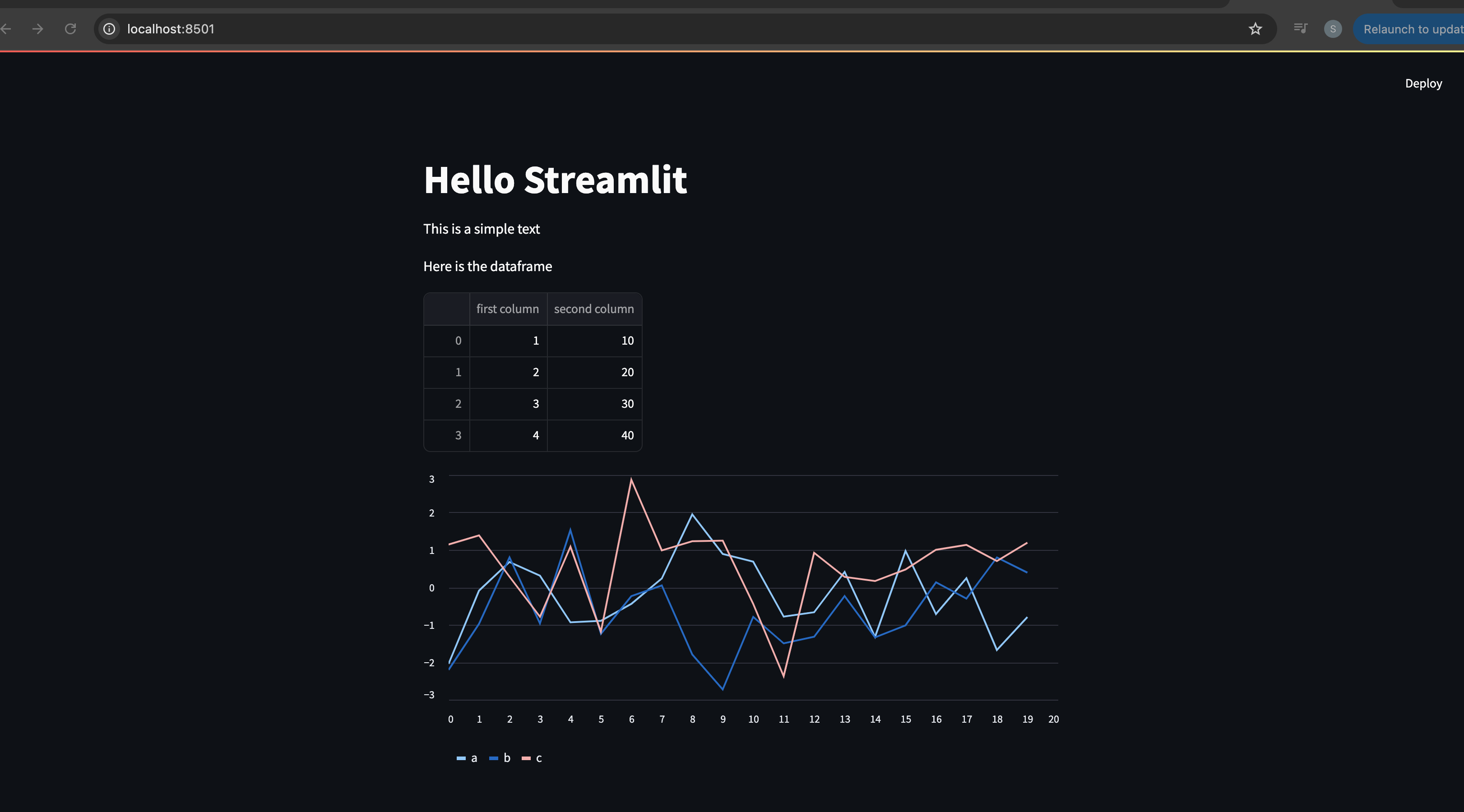


Further display chart in streamlit

Code –

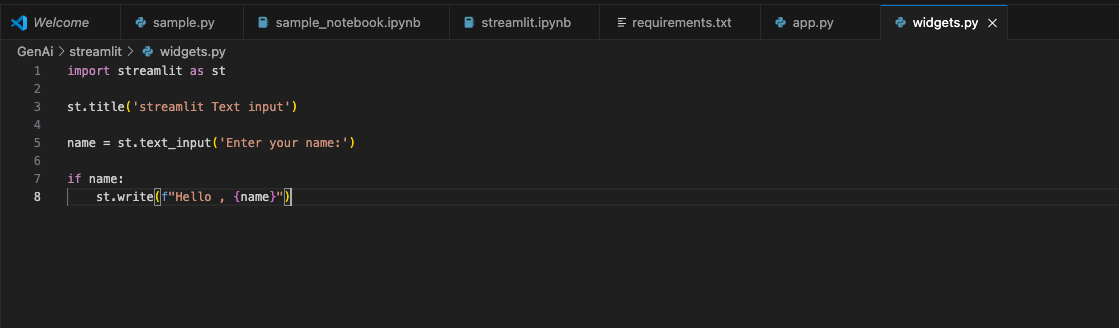


Web app –

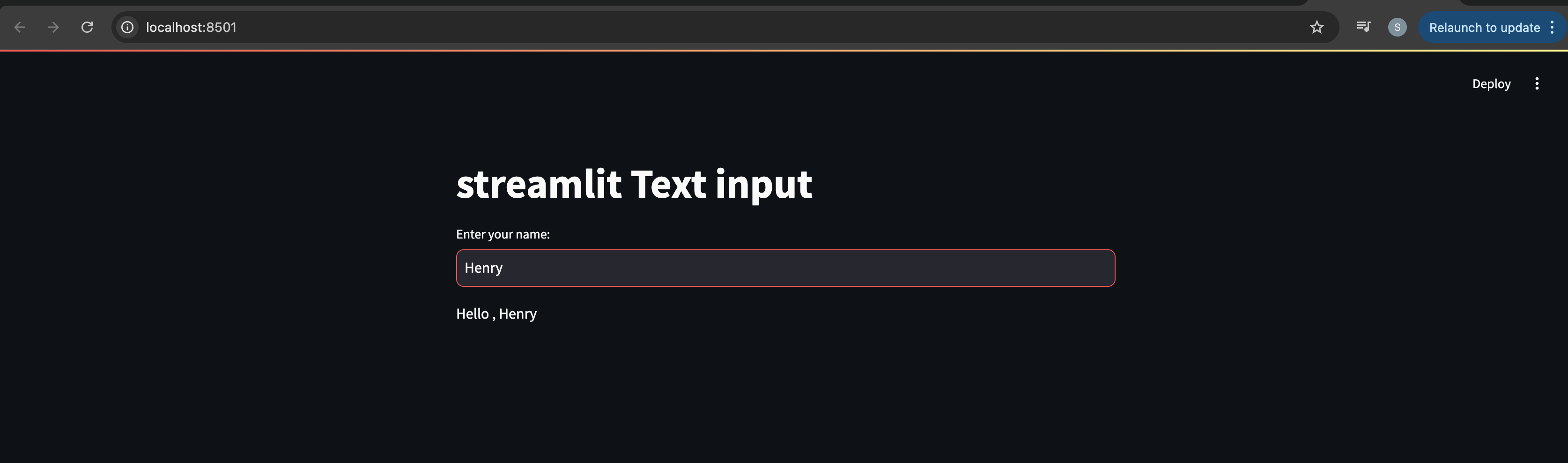


* Streamlit widgets

Code –

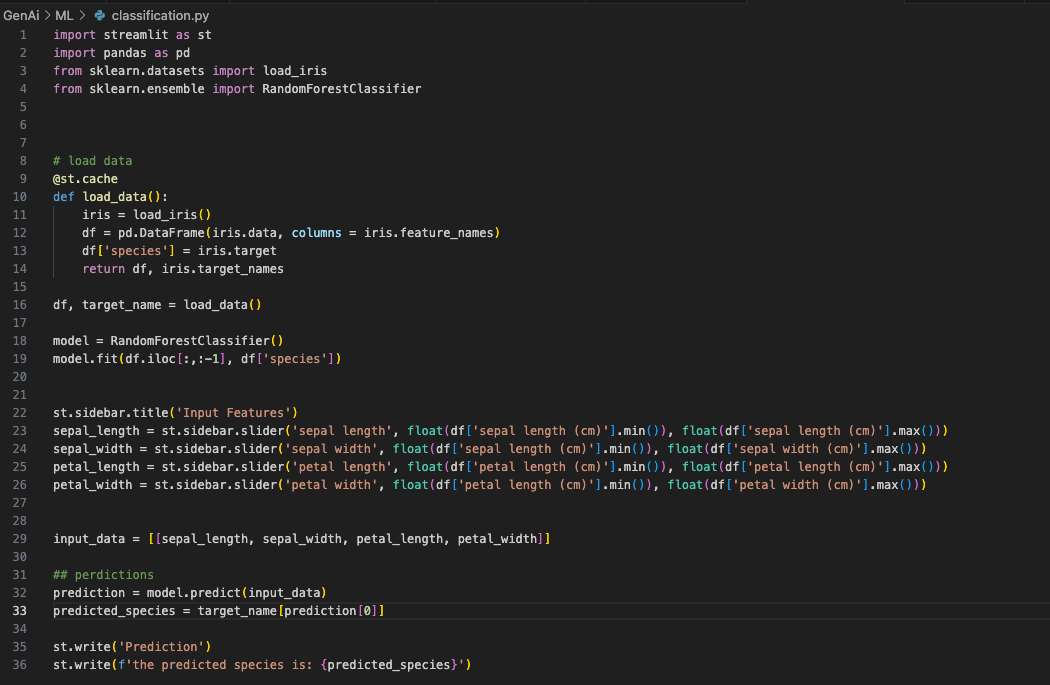


Web app –

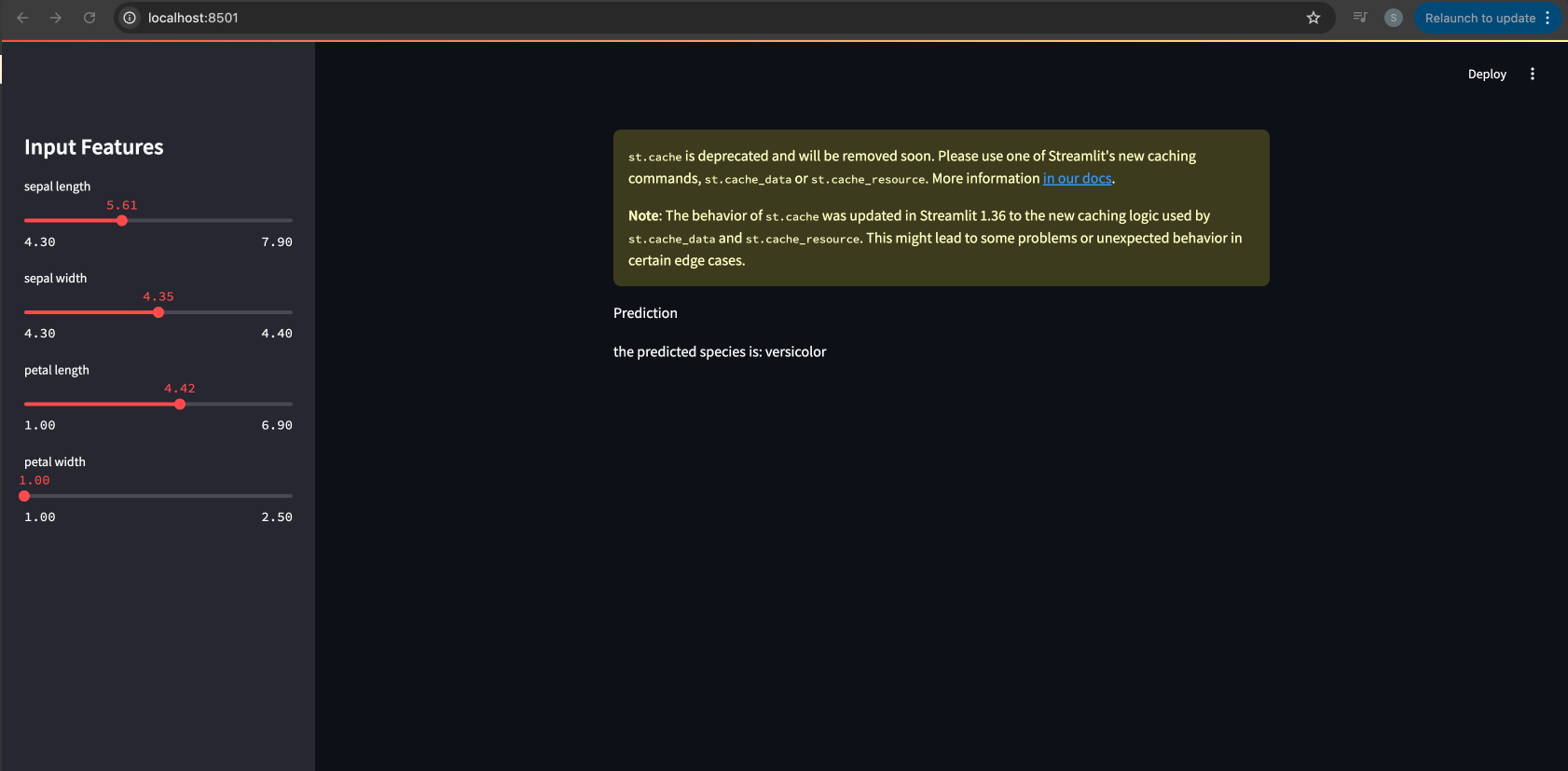


## ML APP with streamlit

* Install scikit-learn library
* Create a ML model. In the example, RandomForsetClassification model is created on iris data. Model predicts species based on provided input for sepal length/width and petal length/width.
* Streamlit takes input as slidebar and provides predictions with st.write

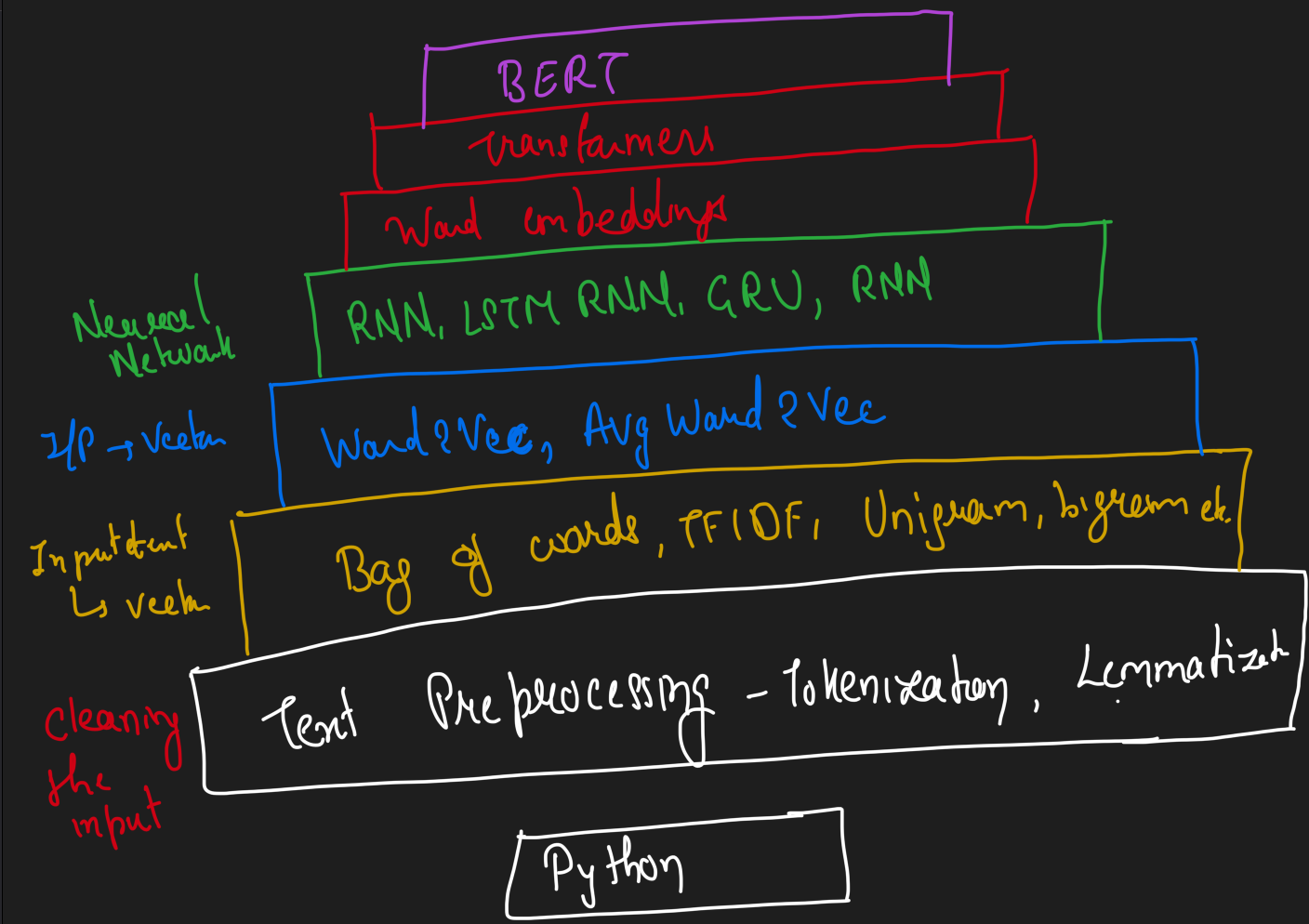


Web app –



## Machine Learning for NLP

### High-level Overview



### Practical Use case of NLP

* Auto spelling check/correct
* Auto reply
* Google translate
* Text to image
* Text to Video
* And many more…..

### Tokenization and Basic Terminologies

* Corpus – Paragraph
* Documents – Sentences
* Vocabulary – Unique words
* Words

#### Tokenization

Example -

“My name is Siddharth. A student of GenAI and Data science. Also a Data Engineer.”

This is called corpus.

Token{corpus}

On tokenizing corpus, it will create sentences as below

My name is Siddharth

A student of GenAI and Data science

Also a Data Engineer

Token{sentences}

Tokenizing sentence will create words i.e. unique word.

#### Stemming

Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as lemma.

1. PorterStemmer

For some of the word it might not return the word with exact meaning

For example,

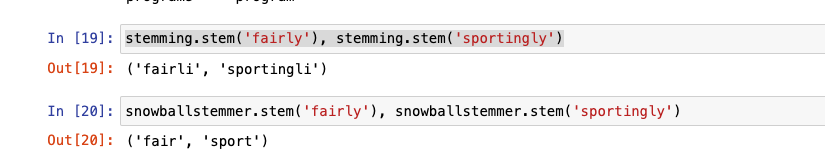
History changes to Histori after PorterStemmer

1. RegexpStemmer class

Pass regular expression to Stem accordingly

1. SnowballStemmer

Its better than PorterStemmer.



Disadvantage –

Even with all of above 3 stemming techniques, it is possible that stemming returns incorrect word.   
Resolve –

To overcome this disadvantage, lemmatization is used.

#### Lemmatization

1. Wordnet Lemmatizer

Lemmatization technique is like stemming. The output we will get after lemmatization is called ‘lemma’ which is a root word rather than root stem, the output of stemming. After lemmatization, we will be getting a valid word that means the same thing.

#### Other Pre-processing steps

* Stopwords
* Parts of Speech Tagging
* Named Entity Recognition
* One Hot Encoding
* Bag of words
* N Grams
* TF-IDF (Term Frequency – Inverse Document Frequency)

TF = No. of repetition of words in sentence/no. of words in sentence

IDF = ln(No. of sentences/no. of sentences containing the word)

* Word Embedding

Encodes the meaning of word such that the words that are closer in the vector space are expected to be similar in meaning.

Types –

1. **Count/Frequency** – One hot encoding, bag of words, TF-IDF
2. **Deep learning trained model –** Word2Vec

* Word2Vec

Word2Vec is a technique for NLP, it uses a neural network model to learn word association form a large corpus of text. Once trained such a model can detect synoynous or suggest additional word for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of number called a vector.

Types –

CBOW (continues back of words)  
 Skipgram

## Deep Learning For NLP

## ANN

Requirements –

Packages –

tensorflow==2.15.0

pandas

numpy

scikit-learn

tensorboard

matplotlib

streamlit

scikeras

pip3 install ipykernel