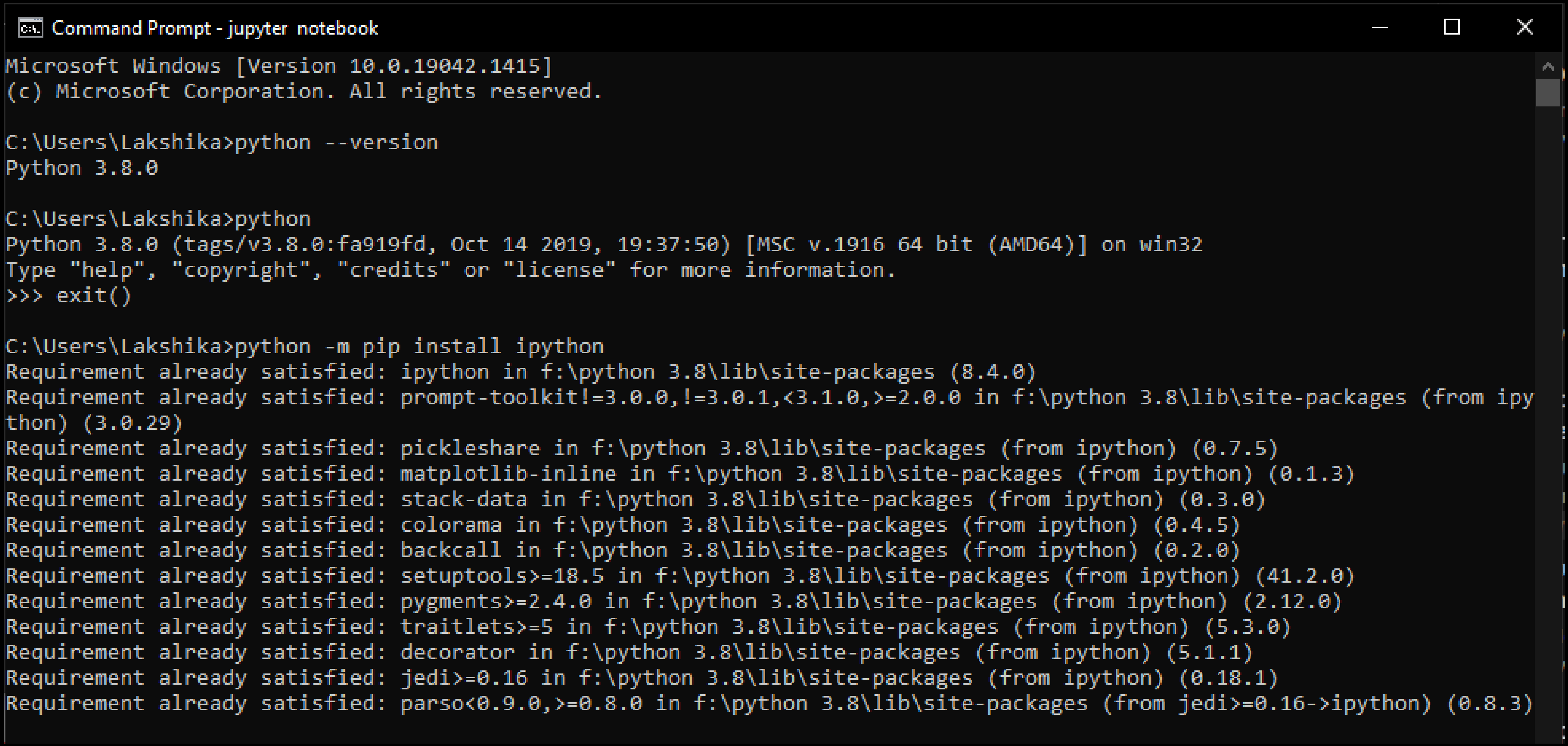
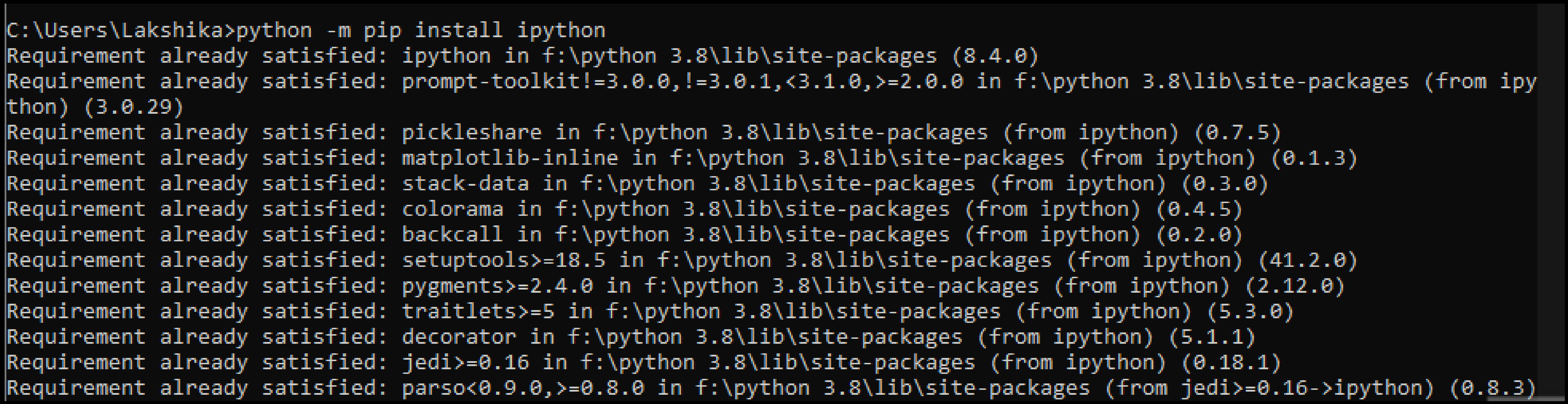
Lab Session 2

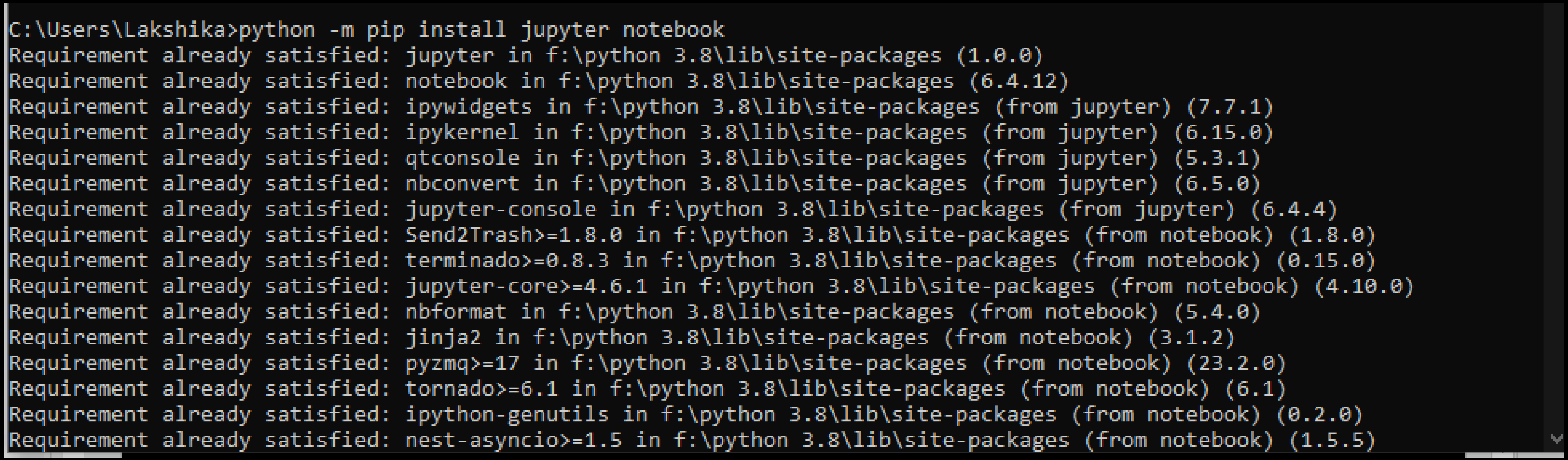
Work with Jupyter notebook



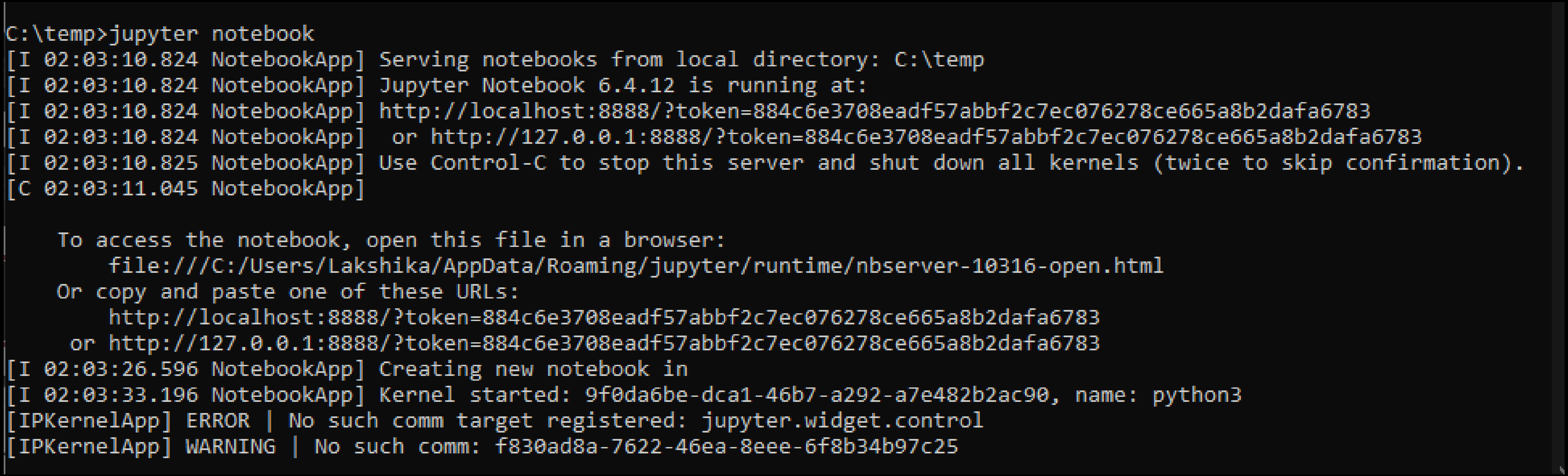
```
Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lakshika>python --version
Python 3.8.0

C:\Users\Lakshika>python
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Users\Lakshika>python -m pip install ipython
Requirement already satisfied: ipython in f:\python 3.8\lib\site-packages (8.4.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in f:\python 3.8\lib\site-packages (from ipy
thon) (3.0.29)
Requirement already satisfied: pickleshare in f:\python 3.8\lib\site-packages (from ipython) (0.7.5)
Requirement already satisfied: matplotlib-inline in f:\python 3.8\lib\site-packages (from ipython) (0.1.3)
Requirement already satisfied: stack-data in f:\python 3.8\lib\site-packages (from ipython) (0.3.0)
Requirement already satisfied: colorama in f:\python 3.8\lib\site-packages (from ipython) (0.4.5)
Requirement already satisfied: backcall in f:\python 3.8\lib\site-packages (from ipython) (0.2.0)
Requirement already satisfied: setuptools>=18.5 in f:\python 3.8\lib\site-packages (from ipython) (41.2.0)
Requirement already satisfied: pygments>=2.4.0 in f:\python 3.8\lib\site-packages (from ipython) (2.12.0)
Requirement already satisfied: traitlets>=5 in f:\python 3.8\lib\site-packages (from ipython) (5.3.0)
Requirement already satisfied: decorator in f:\python 3.8\lib\site-packages (from ipython) (5.1.1)
Requirement already satisfied: jedi>=0.16 in f:\python 3.8\lib\site-packages (from ipython) (0.18.1)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in f:\python 3.8\lib\site-packages (from jedi>=0.16->ipython) (0.8.3)
```



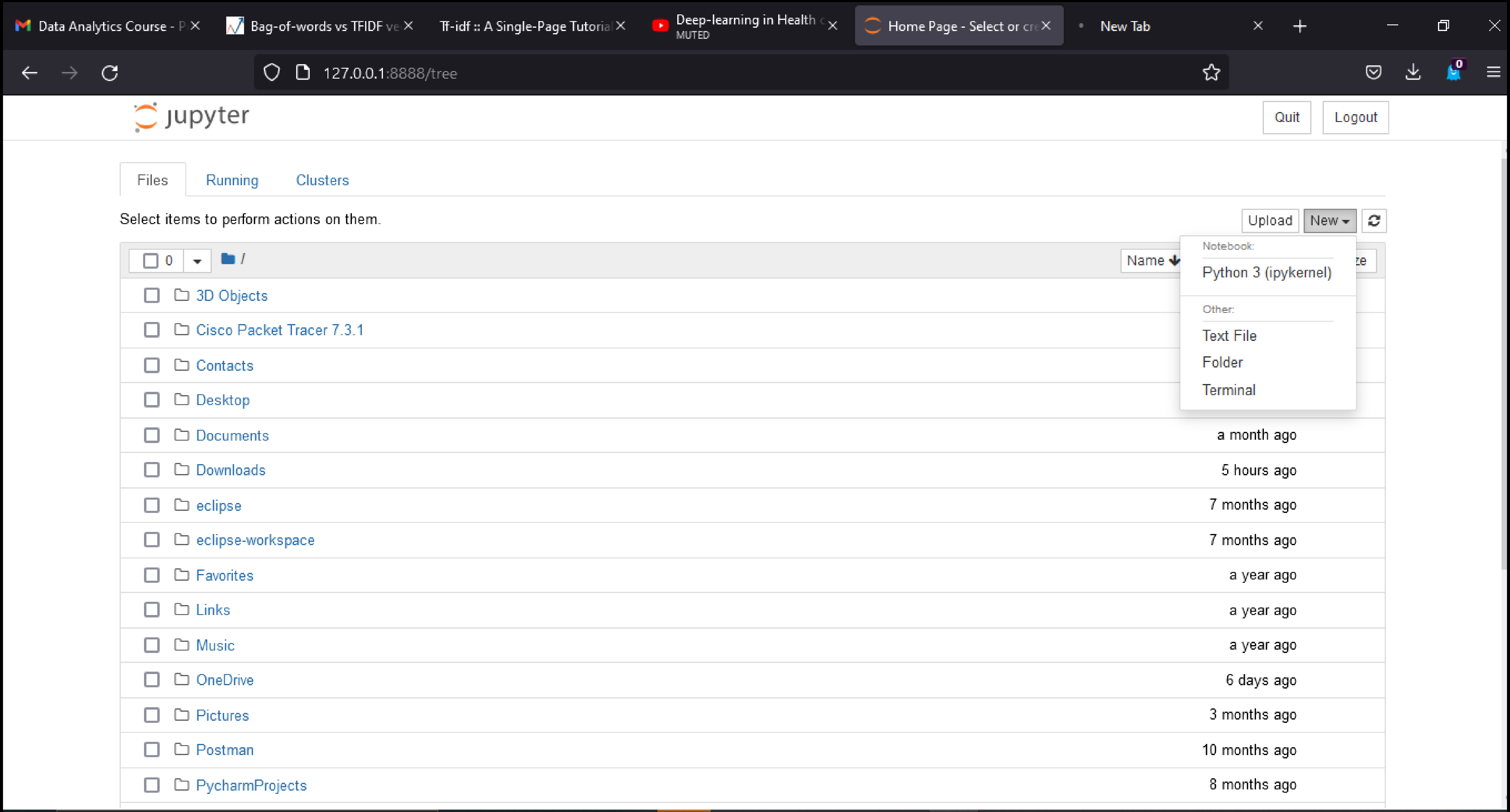
```
C:\Users\Lakshika>python -m pip install ipython
Requirement already satisfied: ipython in f:\python 3.8\lib\site-packages (8.4.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in f:\python 3.8\lib\site-packages (from ipy
thon) (3.0.29)
Requirement already satisfied: pickleshare in f:\python 3.8\lib\site-packages (from ipython) (0.7.5)
Requirement already satisfied: matplotlib-inline in f:\python 3.8\lib\site-packages (from ipython) (0.1.3)
Requirement already satisfied: stack-data in f:\python 3.8\lib\site-packages (from ipython) (0.3.0)
Requirement already satisfied: colorama in f:\python 3.8\lib\site-packages (from ipython) (0.4.5)
Requirement already satisfied: backcall in f:\python 3.8\lib\site-packages (from ipython) (0.2.0)
Requirement already satisfied: setuptools>=18.5 in f:\python 3.8\lib\site-packages (from ipython) (41.2.0)
Requirement already satisfied: pygments>=2.4.0 in f:\python 3.8\lib\site-packages (from ipython) (2.12.0)
Requirement already satisfied: traitlets>=5 in f:\python 3.8\lib\site-packages (from ipython) (5.3.0)
Requirement already satisfied: decorator in f:\python 3.8\lib\site-packages (from ipython) (5.1.1)
Requirement already satisfied: jedi>=0.16 in f:\python 3.8\lib\site-packages (from ipython) (0.18.1)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in f:\python 3.8\lib\site-packages (from jedi>=0.16->ipython) (0.8.3)
```



```
C:\Users\Lakshika>python -m pip install jupyter notebook
Requirement already satisfied: jupyter in f:\python 3.8\lib\site-packages (1.0.0)
Requirement already satisfied: notebook in f:\python 3.8\lib\site-packages (6.4.12)
Requirement already satisfied: ipywidgets in f:\python 3.8\lib\site-packages (from jupyter) (7.7.1)
Requirement already satisfied: ipykernel in f:\python 3.8\lib\site-packages (from jupyter) (6.15.0)
Requirement already satisfied: qtconsole in f:\python 3.8\lib\site-packages (from jupyter) (5.3.1)
Requirement already satisfied: nbconvert in f:\python 3.8\lib\site-packages (from jupyter) (6.5.0)
Requirement already satisfied: jupyter-console in f:\python 3.8\lib\site-packages (from jupyter) (6.4.4)
Requirement already satisfied: Send2Trash>=1.8.0 in f:\python 3.8\lib\site-packages (from notebook) (1.8.0)
Requirement already satisfied: terminado>=0.8.3 in f:\python 3.8\lib\site-packages (from notebook) (0.15.0)
Requirement already satisfied: jupyter-core>=4.6.1 in f:\python 3.8\lib\site-packages (from notebook) (4.10.0)
Requirement already satisfied: nbformat in f:\python 3.8\lib\site-packages (from notebook) (5.4.0)
Requirement already satisfied: jinja2 in f:\python 3.8\lib\site-packages (from notebook) (3.1.2)
Requirement already satisfied: pyzmq>=17 in f:\python 3.8\lib\site-packages (from notebook) (23.2.0)
Requirement already satisfied: tornado>=6.1 in f:\python 3.8\lib\site-packages (from notebook) (6.1)
Requirement already satisfied: ipython-genutils in f:\python 3.8\lib\site-packages (from notebook) (0.2.0)
Requirement already satisfied: nest-asyncio>=1.5 in f:\python 3.8\lib\site-packages (from notebook) (1.5.5)
```

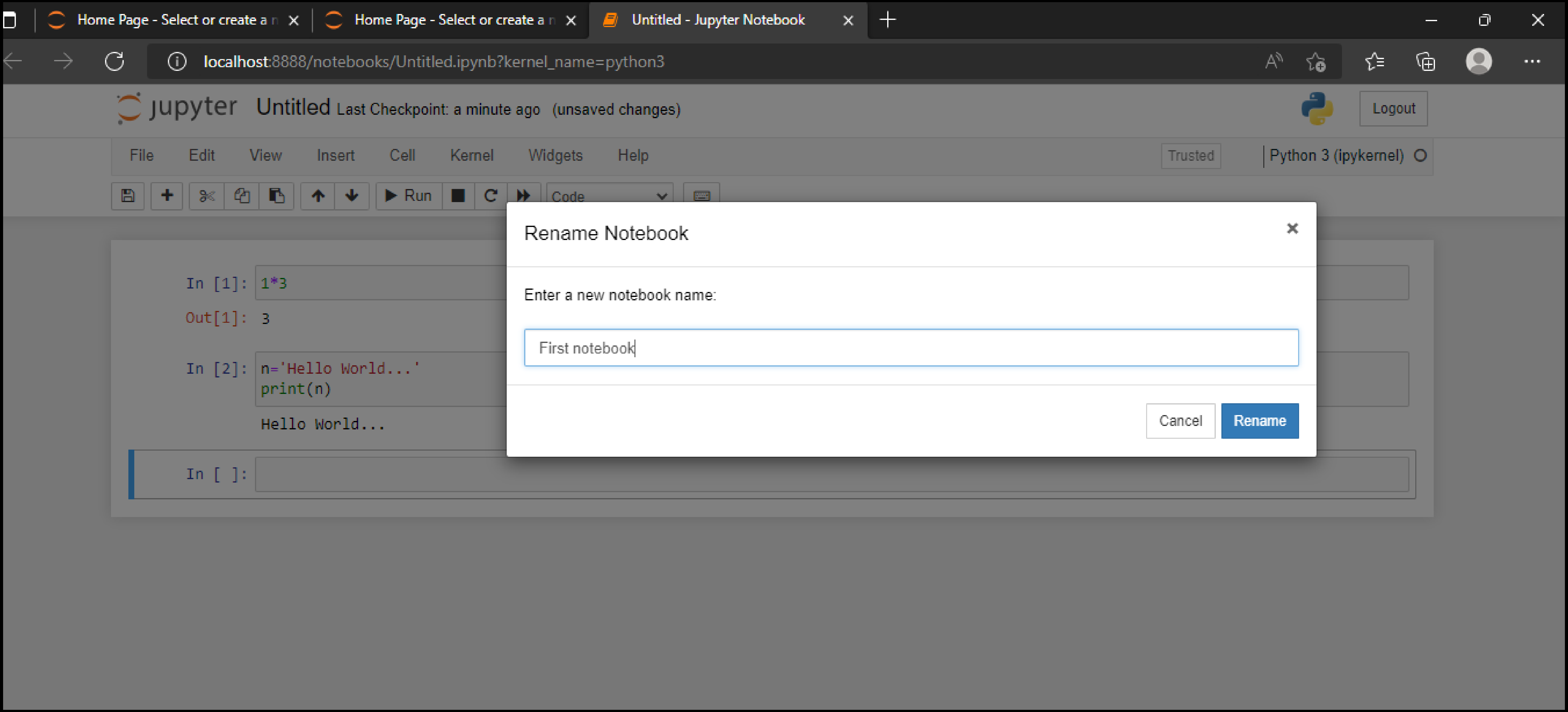


```
C:\temp>jupyter notebook
[I 02:03:10.824 NotebookApp] Serving notebooks from local directory: C:\temp
[I 02:03:10.824 NotebookApp] Jupyter Notebook 6.4.12 is running at:
[I 02:03:10.824 NotebookApp] http://localhost:8888/?token=884c6e3708eadf57abbf2c7ec076278ce665a8b2dafa6783
[I 02:03:10.824 NotebookApp]  or http://127.0.0.1:8888/?token=884c6e3708eadf57abbf2c7ec076278ce665a8b2dafa6783
[I 02:03:10.825 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 02:03:11.045 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/Lakshika/AppData/Roaming/jupyter/runtime/nbserver-10316-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=884c6e3708eadf57abbf2c7ec076278ce665a8b2dafa6783
     or http://127.0.0.1:8888/?token=884c6e3708eadf57abbf2c7ec076278ce665a8b2dafa6783
[I 02:03:26.596 NotebookApp] Creating new notebook in
[I 02:03:33.196 NotebookApp] Kernel started: 9f0da6be-dca1-46b7-a292-a7e482b2ac90, name: python3
[IPKernelApp] ERROR | No such comm target registered: jupyter.widget.control
[IPKernelApp] WARNING | No such comm: f830ad8a-7622-46ea-8eee-6f8b34b97c25
```



Select items to perform actions on them.

☐ 0 /	Name
☐ 3D Objects	
☐ Cisco Packet Tracer 7.3.1	
☐ Contacts	
☐ Desktop	
☐ Documents	a month ago
☐ Downloads	5 hours ago
☐ eclipse	7 months ago
☐ eclipse-workspace	7 months ago
☐ Favorites	a year ago
☐ Links	a year ago
☐ Music	a year ago
☐ OneDrive	6 days ago
☐ Pictures	3 months ago
☐ Postman	10 months ago
☐ PycharmProjects	8 months ago



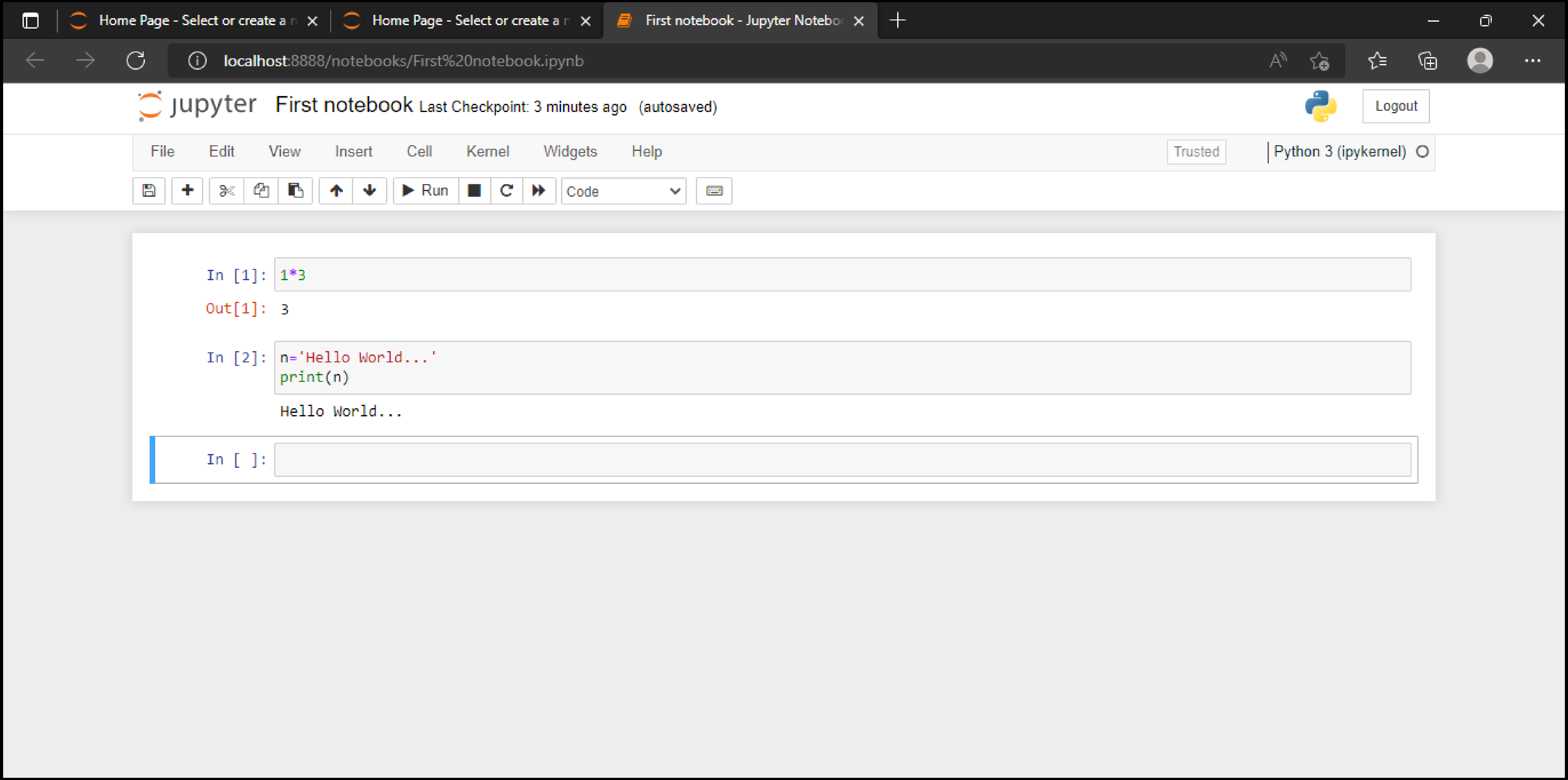
Home Page - Select or create a n
Home Page - Select or create a n
Untitled - Jupyter Notebook
localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3
jupyter
Untitled
Last Checkpoint: a minute ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel) ○
Run
Code
Cancel
Rename

```
In [1]: 1*3
Out[1]: 3

In [2]: n='Hello World...'
        print(n)

        Hello World...

In [ ]:
```

Rename Notebook

Enter a new notebook name:

First notebook


Home Page - Select or create a n
Home Page - Select or create a n
First notebook - Jupyter Notebo
localhost:8888/notebooks/First%20notebook.ipynb
jupyter
First notebook Last Checkpoint: 3 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [1]: 1*3

Out[1]: 3

In [2]: n='Hello World...'
        print(n)

        Hello World...

In [ ]:
```

TFIDF - Vectorization

- Text ----------------> Numeric Form
- We can use preprocessing techniques such as TFIDF and Bags-of-words

For Reading:

Whenever we apply any algorithm to textual data, we need to convert the text to a numeric form. Hence, there arises a need for some pre-processing techniques that can convert our text to numbers. Both bag-of-words (BOW) and TFIDF are pre-processing techniques that can generate a numeric form from an input text.

Term Frequency Inverse Document Frequency (TFIDF)

TF

How frequency term occurs in the document

IDF

The weight of rare words.
The words that occur rarely in the documents. High IDF score.

TFIDF = TF * IDF

TFIDF

For Reading:

TFIDF works by proportionally increasing the number of times a word appears in the document but is counterbalanced by the number of documents in which it is present.

Hence, words like ‘this’, ’are’ etc., that are commonly present in all the documents are not given a very high rank.

However, a word that is present too many times in a few of the documents will be given a higher rank as it might be indicative of the context of the document.

Term Frequency (TF)

$$TF(t) = \frac{\text{Number of times term } t \text{ appears in the document}}{\text{Total number of terms appear in the document}}$$

Ex: Text processing is necessary. Text processing is easy to learn.

TF(Text) = 2 / 10

INVERSE DOCUMENT FREQUENCY (IDF)

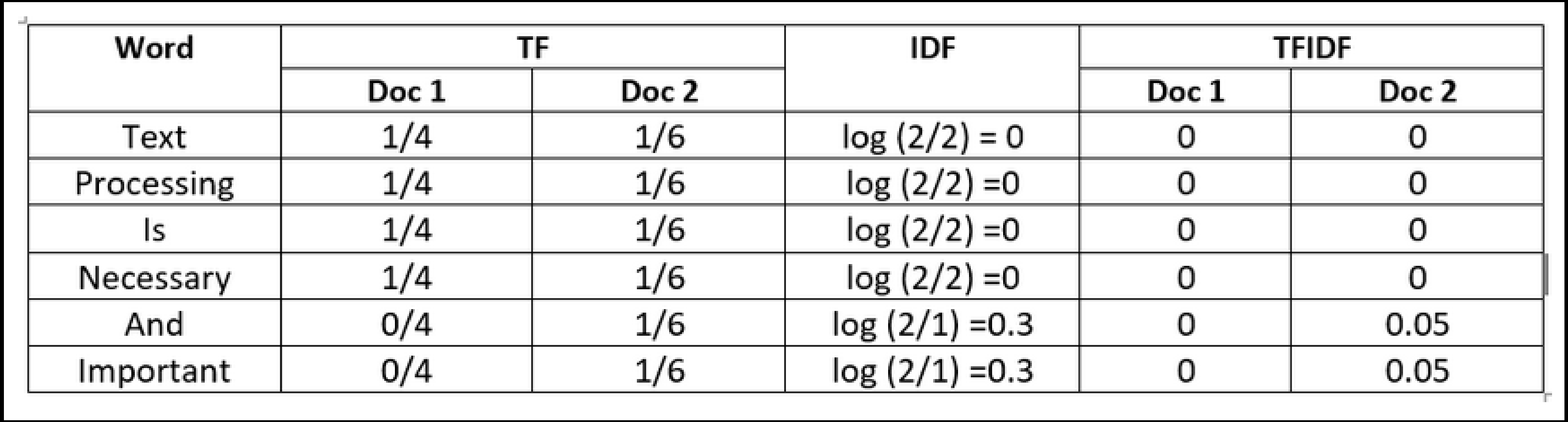
$$\text{IDF}(t) = \log\left[\frac{\text{Total Number of Documents}}{\text{Number of documents with term } t \text{ in it}}\right]$$

Ex:

Document 1 = Text processing is necessary.

Document 2 = Text processing is necessary and important.

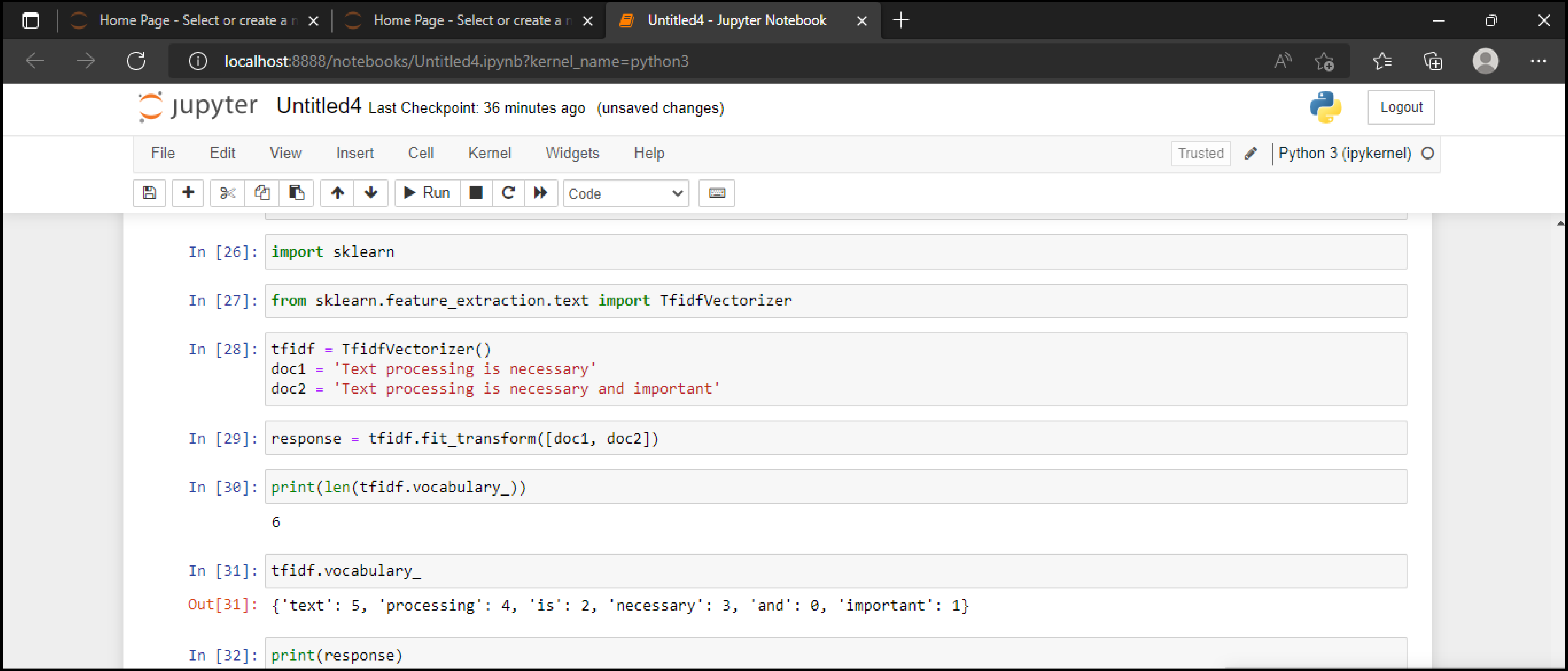
$$\text{IDF(Text)} = \log(2/2)$$



Word	TF		IDF	TFIDF	
	Doc 1	Doc 2		Doc 1	Doc 2
Text	1/4	1/6	log (2/2) = 0	0	0
Processing	1/4	1/6	log (2/2) =0	0	0
Is	1/4	1/6	log (2/2) =0	0	0
Necessary	1/4	1/6	log (2/2) =0	0	0
And	0/4	1/6	log (2/1) =0.3	0	0.05
Important	0/4	1/6	log (2/1) =0.3	0	0.05

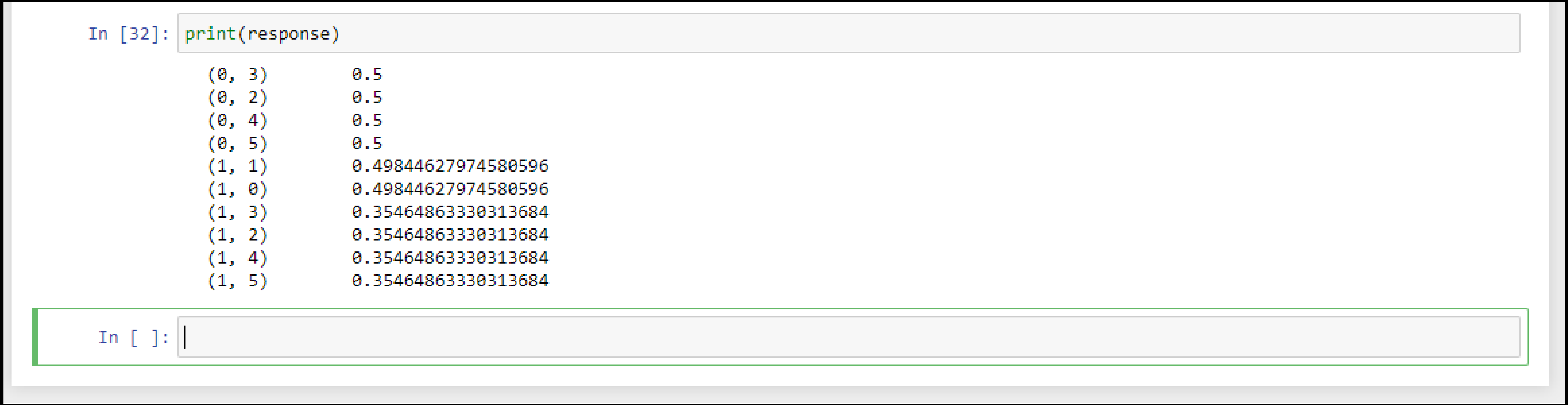
The above table shows how the TFIDF of some words are zero and some words are non-zero depending on their frequency in the document and across all documents.

The limitation of TFIDF is again that this vectorization doesn't help in bringing in the contextual meaning of the words as it is just based on the frequency.



Home Page - Select or create a n
Home Page - Select or create a n
Untitled4 - Jupyter Notebook
localhost:8888/notebooks/Untitled4.ipynb?kernel_name=python3
jupyter Untitled4 Last Checkpoint: 36 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [26]: import sklearn

In [27]: from sklearn.feature_extraction.text import TfidfVectorizer

In [28]: tfidf = TfidfVectorizer()
         doc1 = 'Text processing is necessary'
         doc2 = 'Text processing is necessary and important'

In [29]: response = tfidf.fit_transform([doc1, doc2])

In [30]: print(len(tfidf.vocabulary_))

         6

In [31]: tfidf.vocabulary_

Out[31]: {'text': 5, 'processing': 4, 'is': 2, 'necessary': 3, 'and': 0, 'important': 1}

In [32]: print(response)
```



```
print(response)
```

```
  (0, 3)	0.5
  (0, 2)	0.5
  (0, 4)	0.5
  (0, 5)	0.5
  (1, 1)	0.49844627974580596
  (1, 0)	0.49844627974580596
  (1, 3)	0.35464863330313684
  (1, 2)	0.35464863330313684
  (1, 4)	0.35464863330313684
  (1, 5)	0.35464863330313684
```