Artificial Intelligence Mini Project

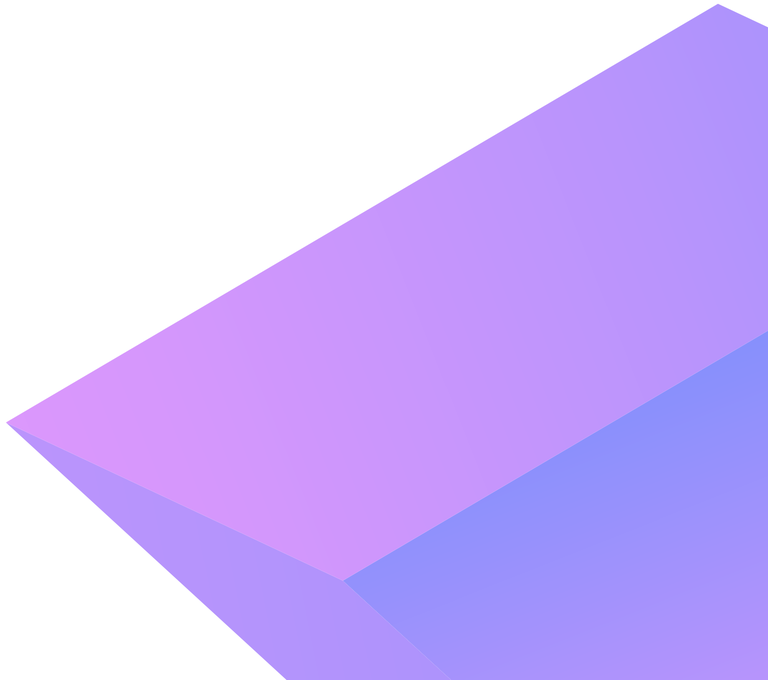
Next Word Predictor

Using Python

Samriddho Das (119)
Madura Sankar S. (114)
Ashwin S. (109)

You must therefore have seen the prediction of the next word on the keyboard of our smartphones which is most of the time precise. How does our keyboard predict the next word? In this article, I will walk you through the task of the next word prediction with Python.

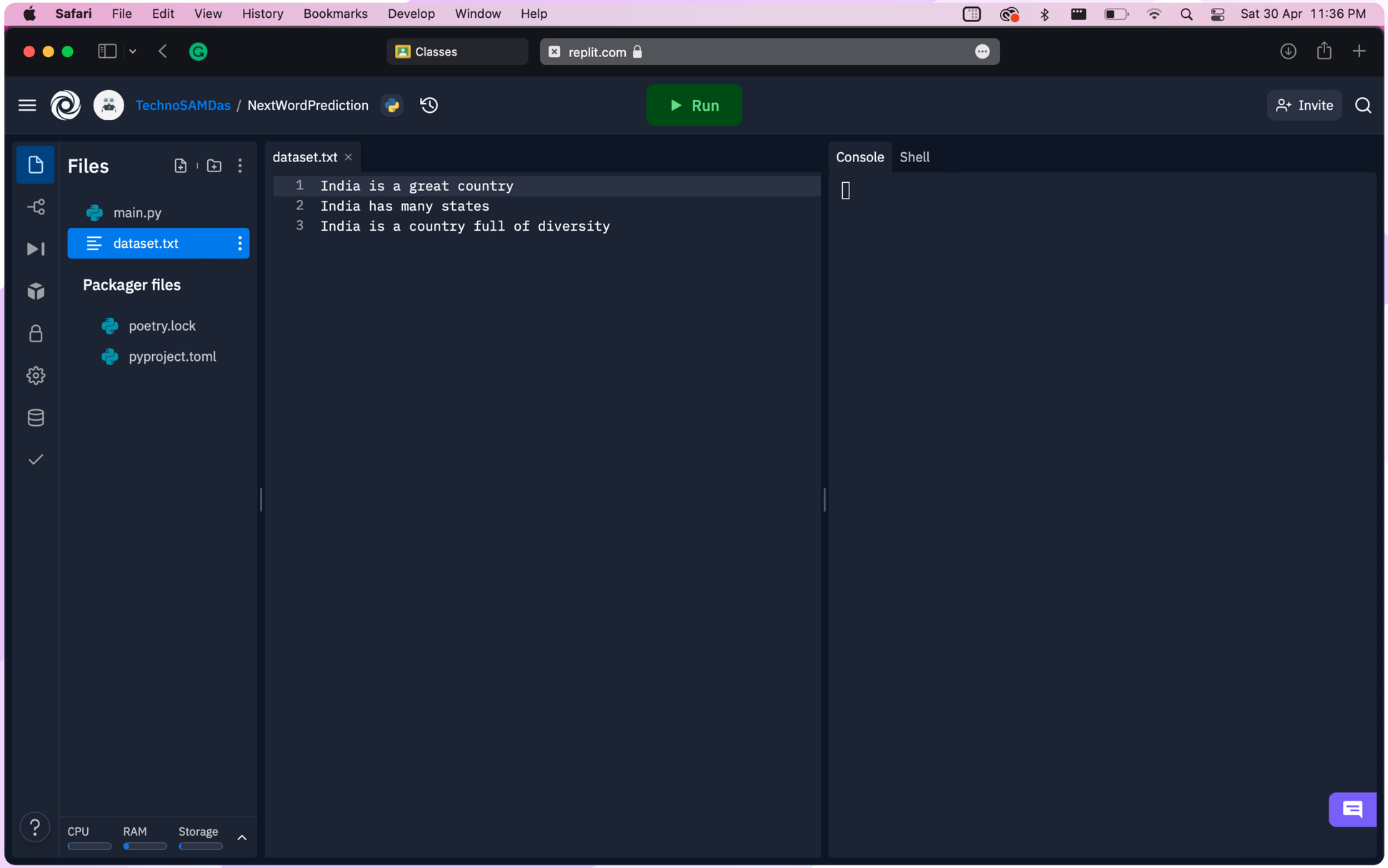
Predicting the next word is a machine learning task, but we can also achieve good results using only Python. Below are the steps you need to take when writing code to predict the next word with Python:



- First, divide the sentence into words
- Then select the last word of the sentence
- Then find the probability of the last word by looking at the vocabulary (dataset)
- Then, finally, select the next possible word

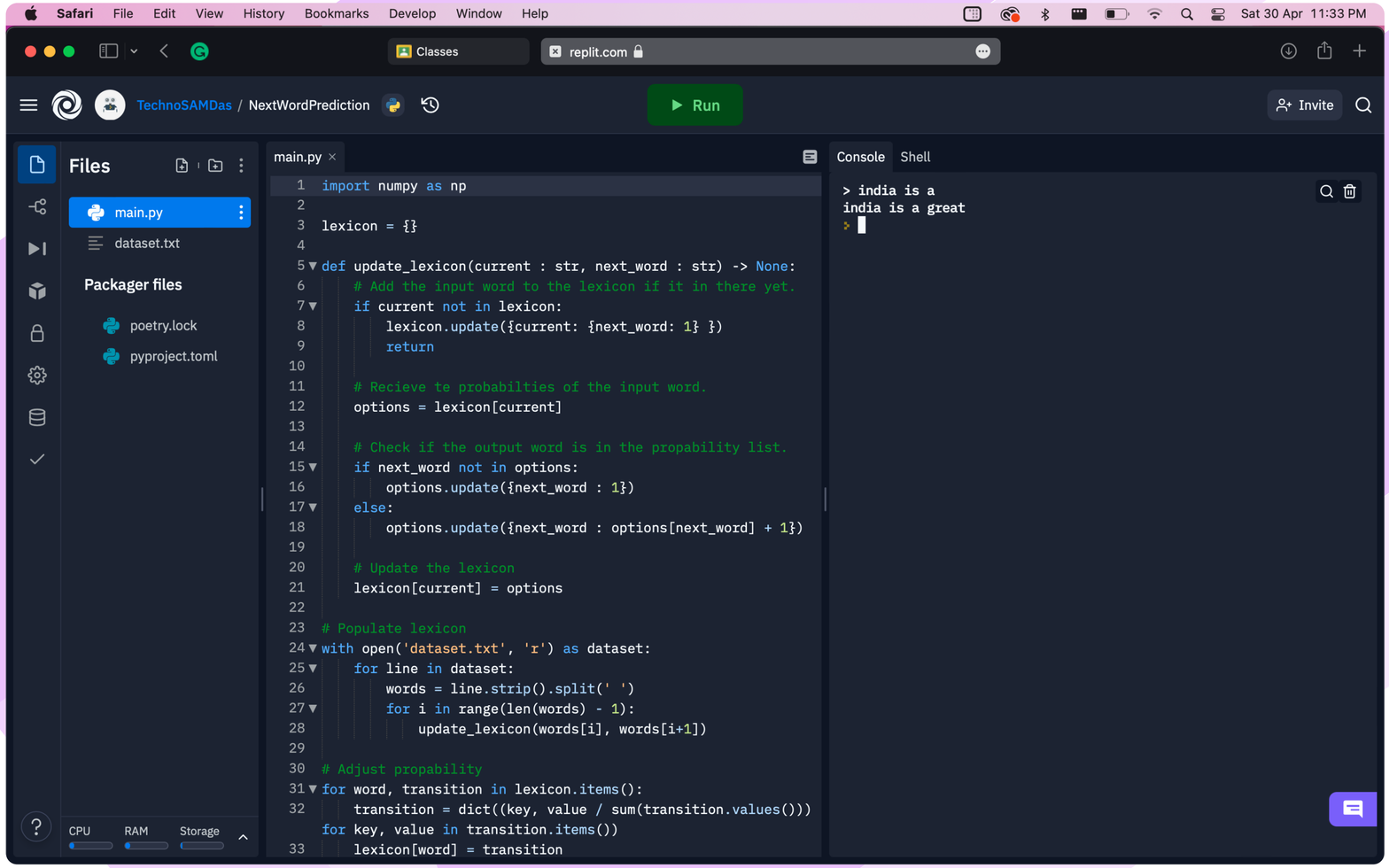
Hands-on Demo

In this section, I'll walk you through a simple Python code to predict the next word using the Python programming language. So, to start with the task of next word prediction with Python, you need to have a vocabulary as shown below.


Safari File Edit View History Bookmarks Develop Window Help
Sat 30 Apr 11:36 PM
Classes
replit.com
TechnoSAMDas / NextWordPrediction
Run
Invite
Files
main.py
dataset.txt
Packager files
poetry.lock
pyproject.toml
CPU
RAM
Storage
dataset.txt
1 India is a great country
2 India has many states
3 India is a country full of diversity
Console
Shell

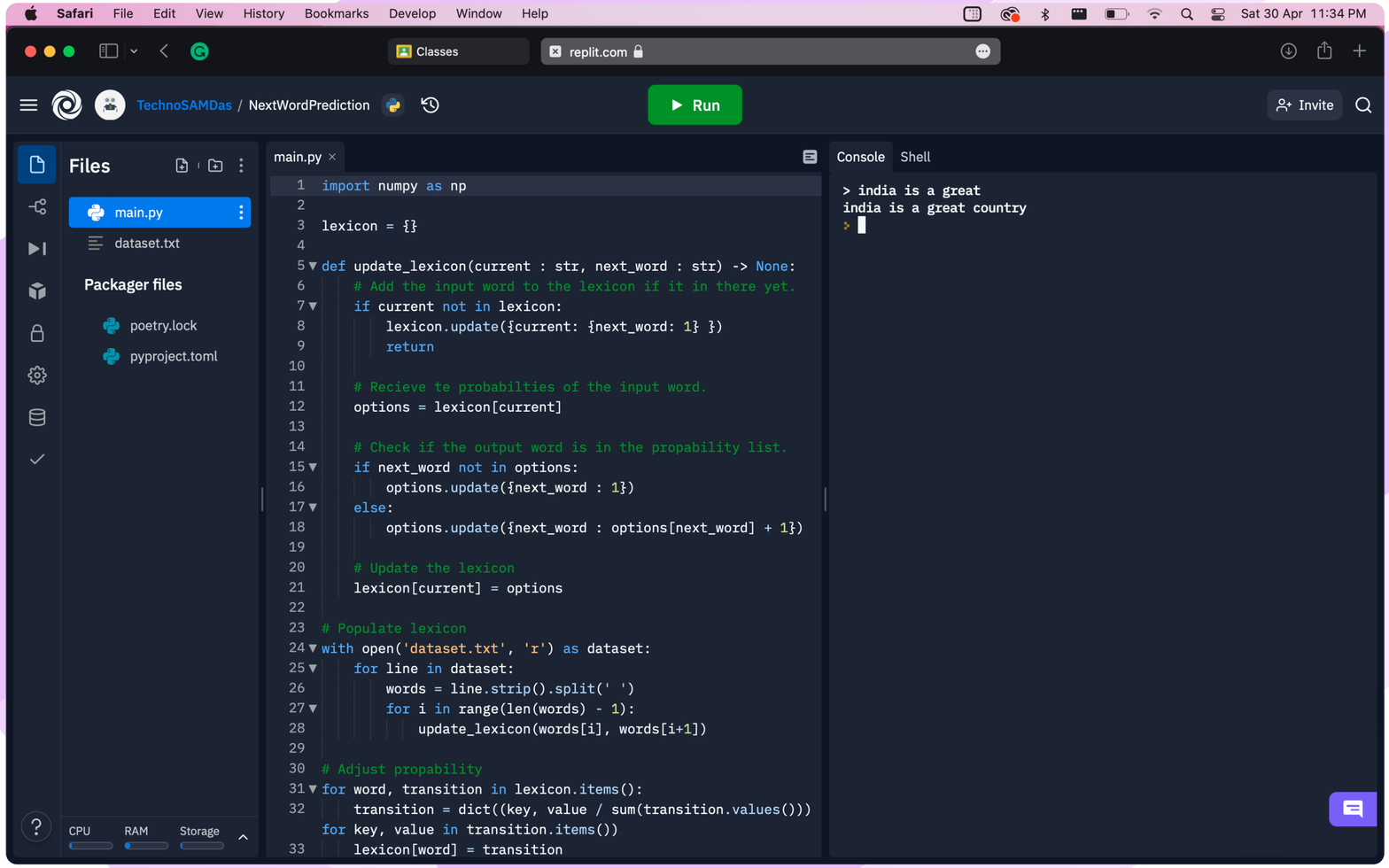
Step 1:

Type the initial portion of any line and press enter. The code will predict a next wird for the line by referencing the given dataset.txt

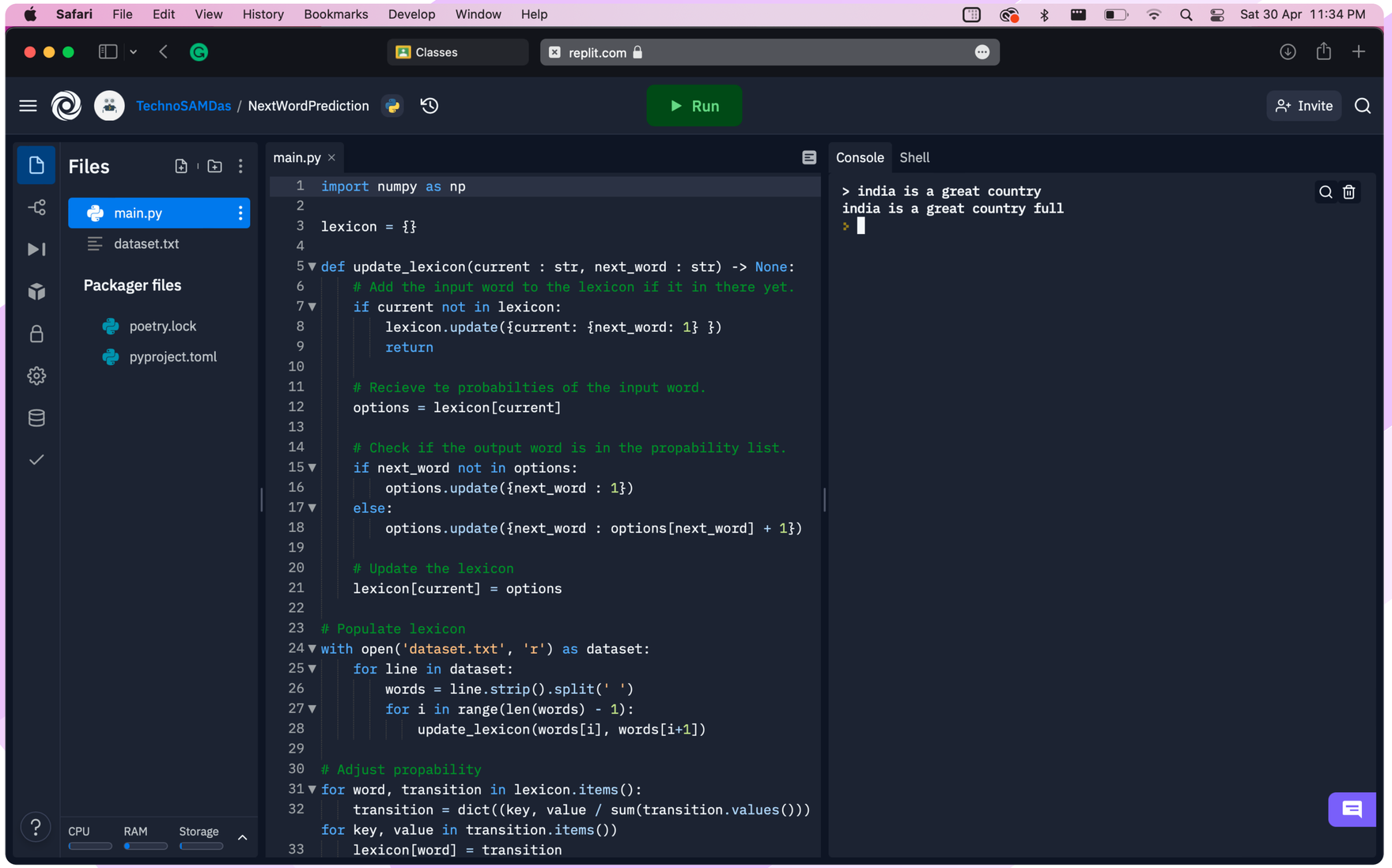


Step 2:

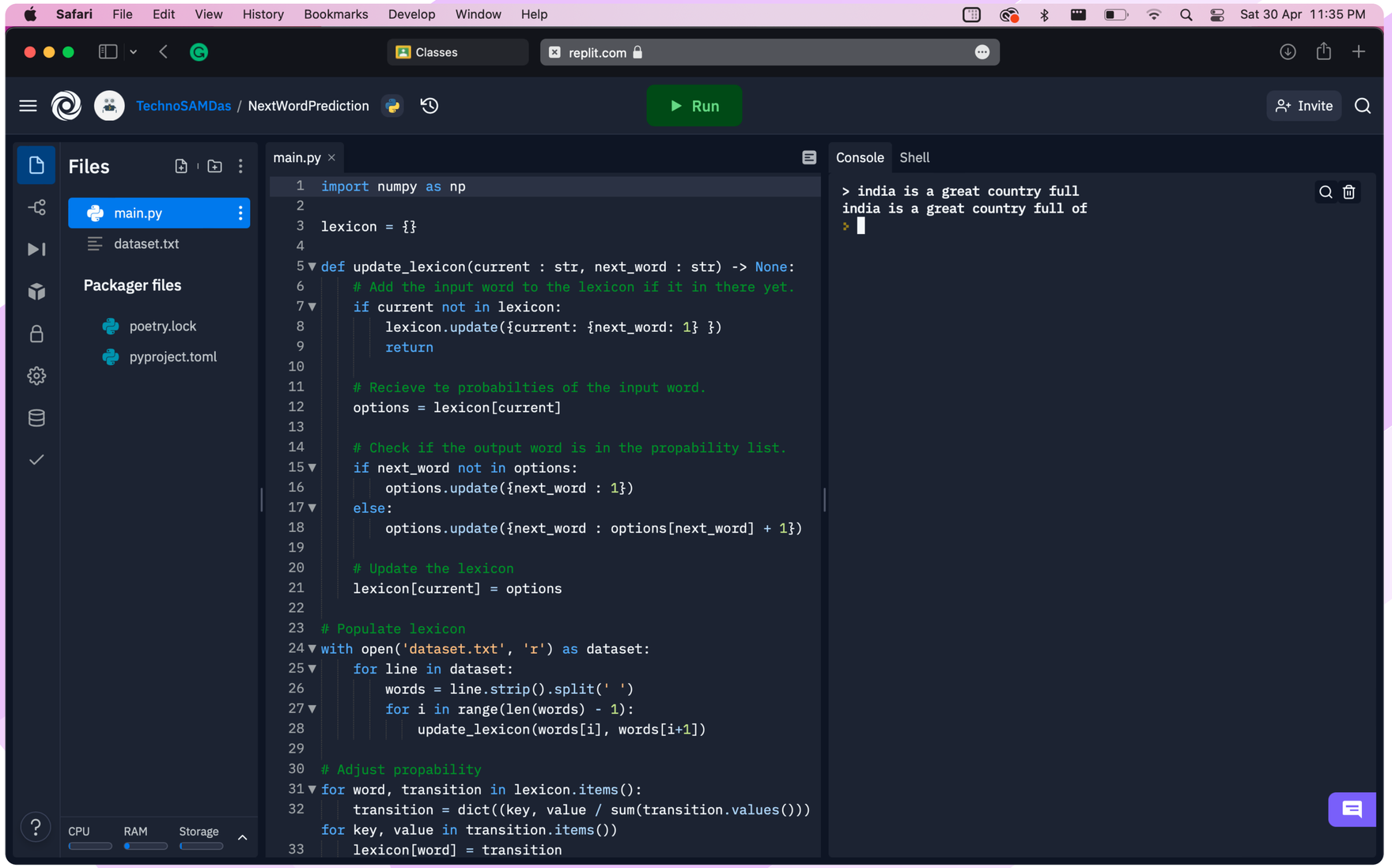
Now copy the output line. Run the code once again and paste it. The code will add a new word to the incomplete sentence. Repeat this process till the whole sentence is complete



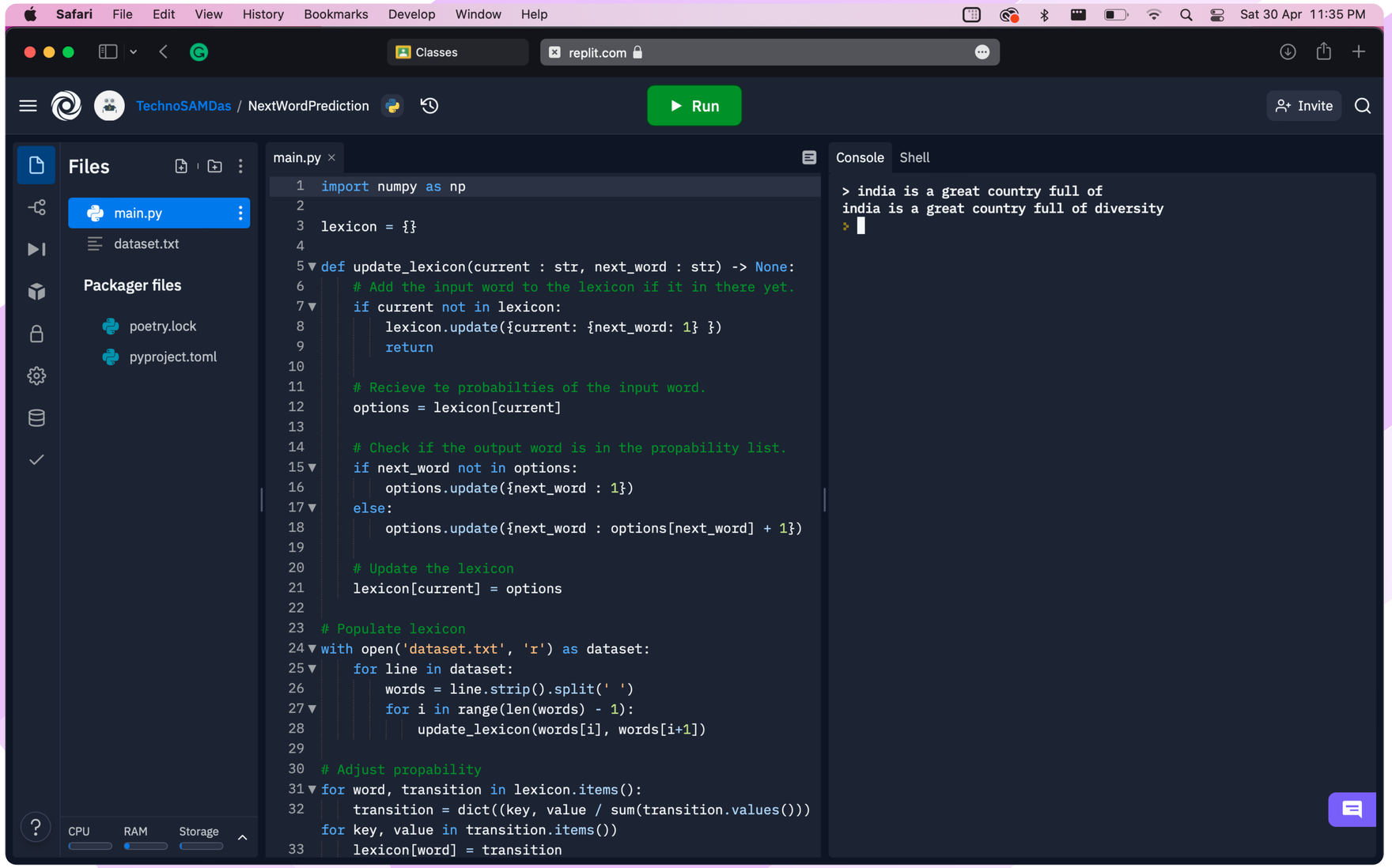
Step 3:


Safari File Edit View History Bookmarks Develop Window Help
Sat 30 Apr 11:34 PM
Classes
replit.com
TechnoSAMDas / NextWordPrediction
Run
Invite
Files
main.py
dataset.txt
Packager files
poetry.lock
pyproject.toml
CPU
RAM
Storage
main.py
Console
Shell
> india is a great country
india is a great country full
import numpy as np
lexicon = {}
def update_lexicon(current : str, next_word : str) -> None:
# Add the input word to the lexicon if it in there yet.
if current not in lexicon:
lexicon.update({current: {next_word: 1} })
return
# Recieve te probabilties of the input word.
options = lexicon[current]
# Check if the output word is in the propability list.
if next_word not in options:
options.update({next_word : 1})
else:
options.update({next_word : options[next_word] + 1})
# Update the lexicon
lexicon[current] = options
# Populate lexicon
with open('dataset.txt', 'r') as dataset:
for line in dataset:
words = line.strip().split(' ')
for i in range(len(words) - 1):
update_lexicon(words[i], words[i+1])
# Adjust propability
for word, transition in lexicon.items():
transition = dict((key, value / sum(transition.values()))
for key, value in transition.items())
lexicon[word] = transition

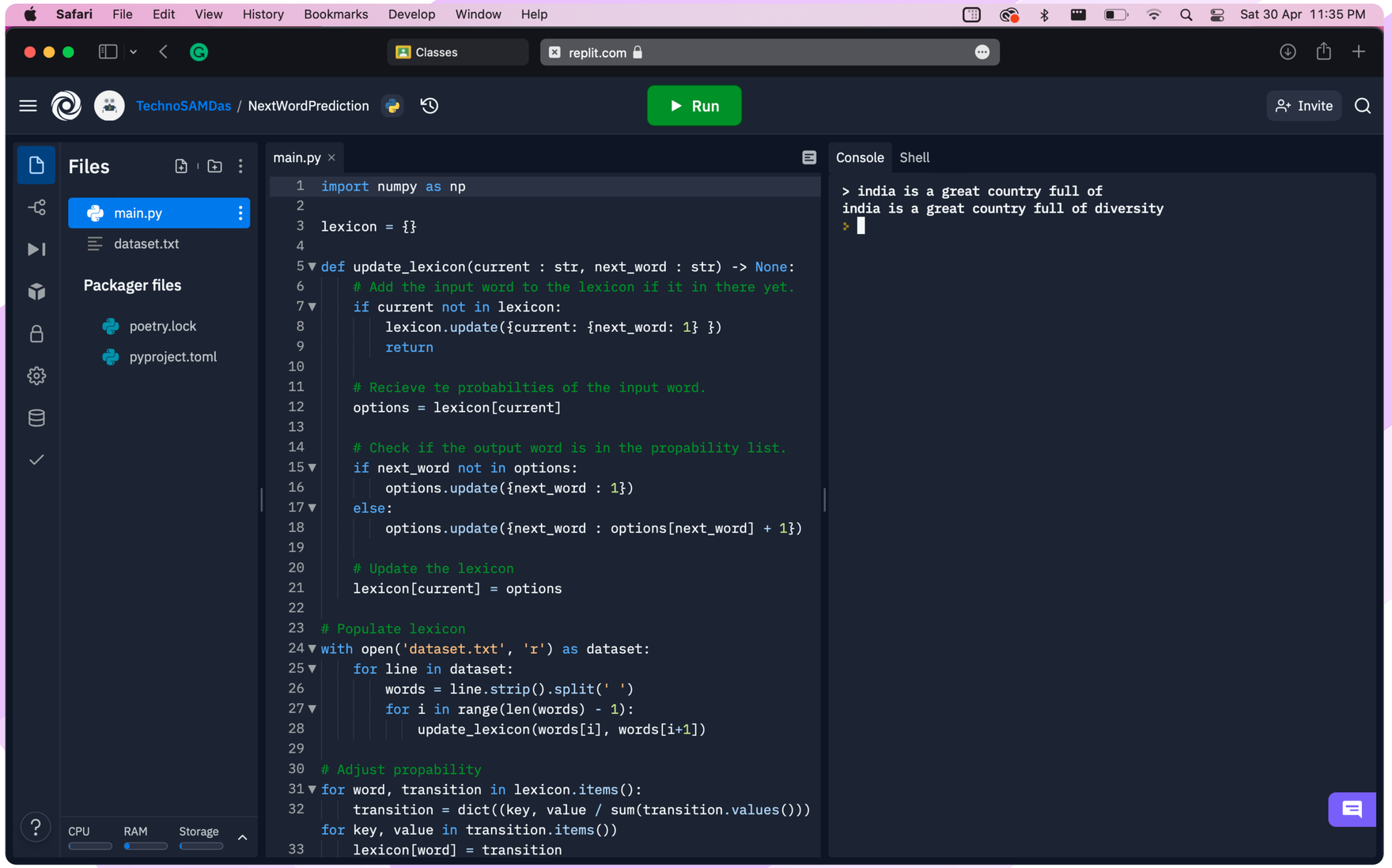
Step 4:


Safari File Edit View History Bookmarks Develop Window Help
Sat 30 Apr 11:35 PM
Classes
replit.com
TechnoSAMDas / NextWordPrediction
Run
Invite
Files
main.py
dataset.txt
Packager files
poetry.lock
pyproject.toml
CPU
RAM
Storage
main.py
Console
Shell
> india is a great country full
india is a great country full of
1 import numpy as np
2
3 lexicon = {}
4
5 def update_lexicon(current : str, next_word : str) -> None:
6 # Add the input word to the lexicon if it in there yet.
7 if current not in lexicon:
8 lexicon.update({current: {next_word: 1} })
9 return
10
11 # Recieve te probabilties of the input word.
12 options = lexicon[current]
13
14 # Check if the output word is in the propability list.
15 if next_word not in options:
16 options.update({next_word : 1})
17 else:
18 options.update({next_word : options[next_word] + 1})
19
20 # Update the lexicon
21 lexicon[current] = options
22
23 # Populate lexicon
24 with open('dataset.txt', 'r') as dataset:
25 for line in dataset:
26 words = line.strip().split(' ')
27 for i in range(len(words) - 1):
28 update_lexicon(words[i], words[i+1])
29
30 # Adjust propability
31 for word, transition in lexicon.items():
32 transition = dict((key, value / sum(transition.values()))
for key, value in transition.items())
33 lexicon[word] = transition

Step 5:


Safari File Edit View History Bookmarks Develop Window Help
Sat 30 Apr 11:35 PM
Classes
replit.com
TechnoSAMDas / NextWordPrediction
Run
Invite
Files
main.py
dataset.txt
Packager files
poetry.lock
pyproject.toml
CPU
RAM
Storage
main.py
1 import numpy as np
2
3 lexicon = {}
4
5 def update_lexicon(current : str, next_word : str) -> None:
6 # Add the input word to the lexicon if it in there yet.
7 if current not in lexicon:
8 lexicon.update({current: {next_word: 1} })
9 return
10
11 # Recieve te probabilties of the input word.
12 options = lexicon[current]
13
14 # Check if the output word is in the propability list.
15 if next_word not in options:
16 options.update({next_word : 1})
17 else:
18 options.update({next_word : options[next_word] + 1})
19
20 # Update the lexicon
21 lexicon[current] = options
22
23 # Populate lexicon
24 with open('dataset.txt', 'r') as dataset:
25 for line in dataset:
26 words = line.strip().split(' ')
27 for i in range(len(words) - 1):
28 update_lexicon(words[i], words[i+1])
29
30 # Adjust propability
31 for word, transition in lexicon.items():
32 transition = dict((key, value / sum(transition.values())) for key, value in transition.items())
33 lexicon[word] = transition
Console
Shell
> india is a great country full of
india is a great country full of diversity

Step 5:


Safari File Edit View History Bookmarks Develop Window Help
Sat 30 Apr 11:35 PM
Classes
replit.com
TechnoSAMDas / NextWordPrediction
Run
Invite
Files
main.py
dataset.txt
Packager files
poetry.lock
pyproject.toml
CPU
RAM
Storage
main.py
1 import numpy as np
2
3 lexicon = {}
4
5 def update_lexicon(current : str, next_word : str) -> None:
6 # Add the input word to the lexicon if it in there yet.
7 if current not in lexicon:
8 lexicon.update({current: {next_word: 1} })
9 return
10
11 # Recieve te probabilties of the input word.
12 options = lexicon[current]
13
14 # Check if the output word is in the propability list.
15 if next_word not in options:
16 options.update({next_word : 1})
17 else:
18 options.update({next_word : options[next_word] + 1})
19
20 # Update the lexicon
21 lexicon[current] = options
22
23 # Populate lexicon
24 with open('dataset.txt', 'r') as dataset:
25 for line in dataset:
26 words = line.strip().split(' ')
27 for i in range(len(words) - 1):
28 update_lexicon(words[i], words[i+1])
29
30 # Adjust propability
31 for word, transition in lexicon.items():
32 transition = dict((key, value / sum(transition.values())) for key, value in transition.items())
33 lexicon[word] = transition
Console
Shell
> india is a great country full of
india is a great country full of diversity

YAAAAAY!!!

The whole sentence is complete.
This is how a simple next word predictor works.

Thank you!!!

Our Teammates

- Samriddho Das (RA1911003010119)
- Madura Sankar S. (RA1911003010114)
- Ashwin S. (RA1911003010109)