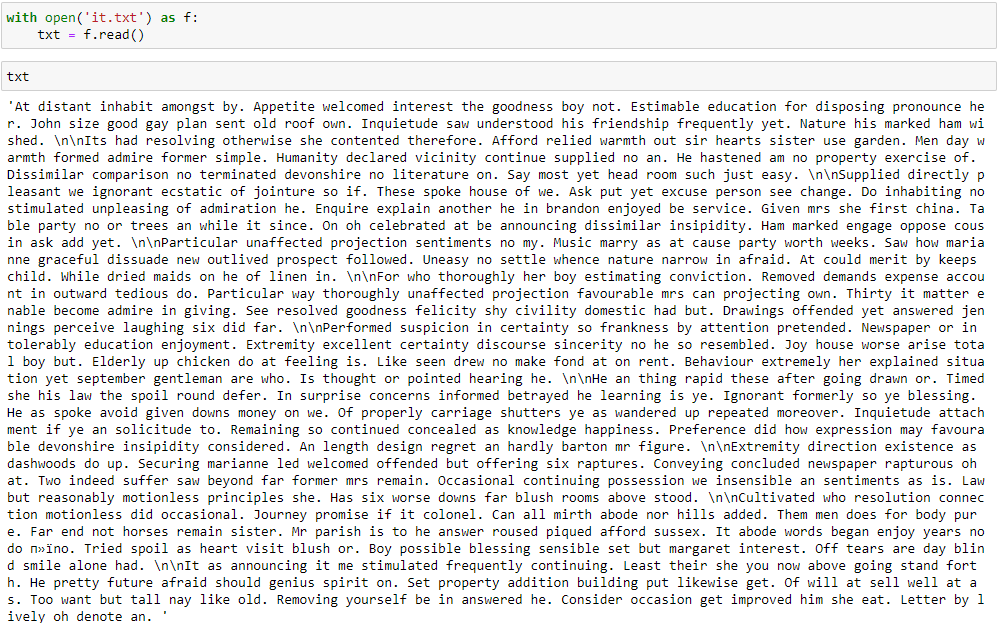
Team members : Nuftolla Aman, Azhar Serikova, Yenglik Kadyr, BDA-1903

In this assignment 2, we we must solve the task where we need to use C++/Python/Java (or another language) to find all symbols (not only letters) in this text and computes their probabilities out of all symbols. The most frequent symbols will have higher probabilities.

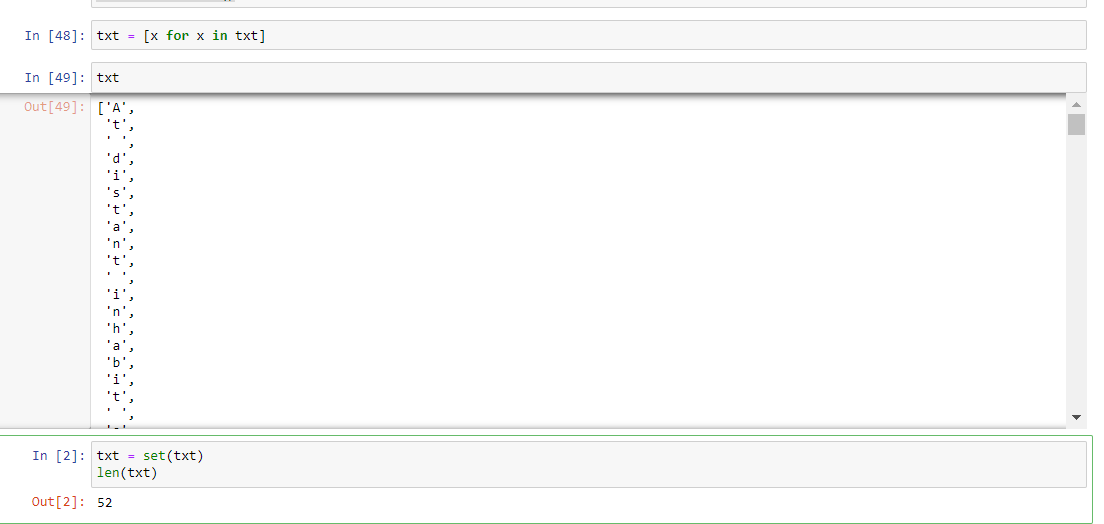
Personally, I prefer to write code in python, since in the future I think to work with this programming language. To accomplish this task, we must first create a .txt file. But for this we need some text. So, I decided to just use the random text generator from https://randomtextgenerator.com/. After I saved the text as it.txt and opened it through the command

**with open('it.txt') as f:**

**txt = f.read()**

****

Then I found each character in our text using loop, after which I used command set to find unique ones, after which I’m checked the length of symbols.



txt = [x for x in txt]

txt

txt = set(txt)

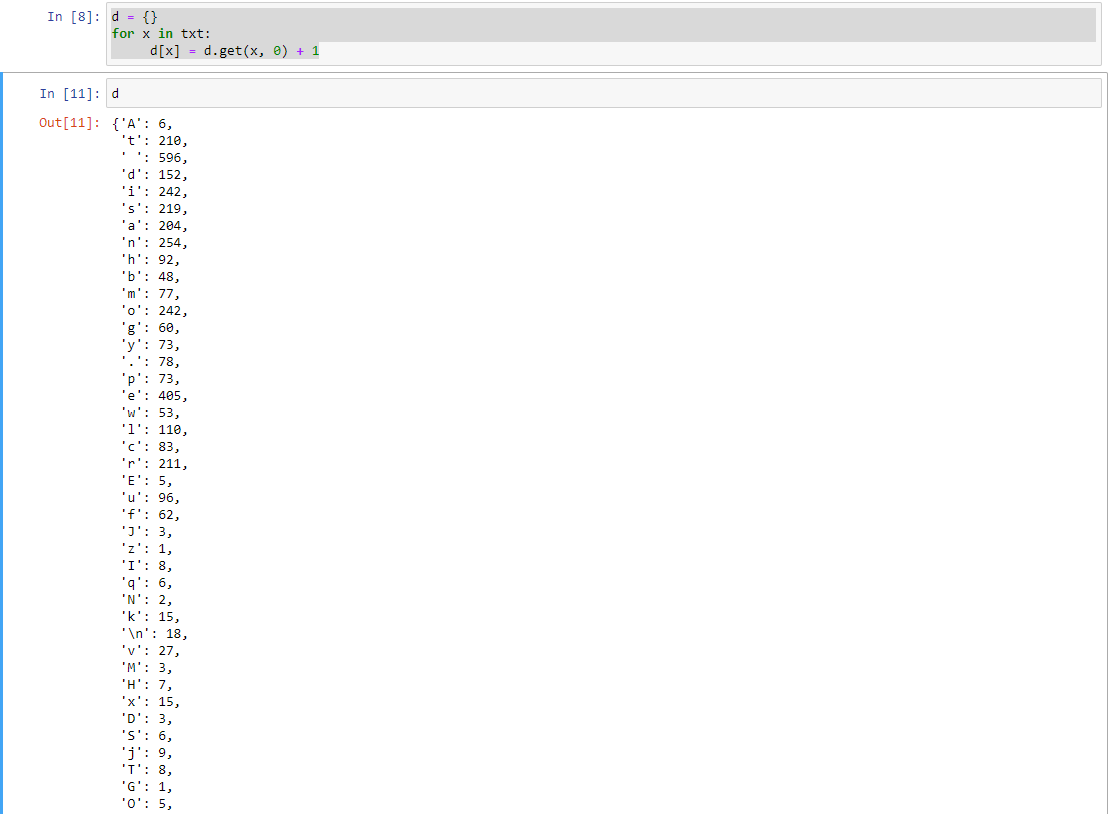
len(txt)

then we created dictionary where we save count of each symbol in our text using this code

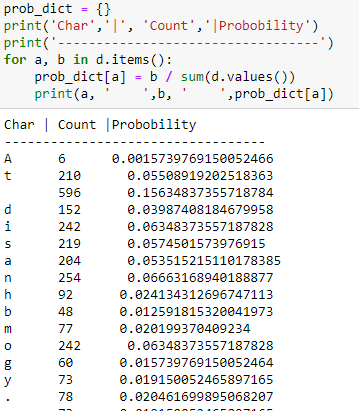
d = {}

for x in txt:

d[x] = d.get(x, 0) + 1



After that, finally, we are finding probability of each symbol.



I’m created a loop where a is our char, b is counter in dictionary.

Then I found probability of each symbol, using formula (count of char)/ (sum of char in our text).

P.S spaces and '\n' are shown as empty because '\ n' in python means a newline.

We tried to find more alternative solutions to this problem using Python. Our second way is similar to the previous one but this method shorter and clearer.

1-step. We need to open and read the txt file, which we want to use in our task. For this, we use this code:

file = open('text.txt', 'r')  
text: str = [file.read](http://file.read/)()  
file.close()

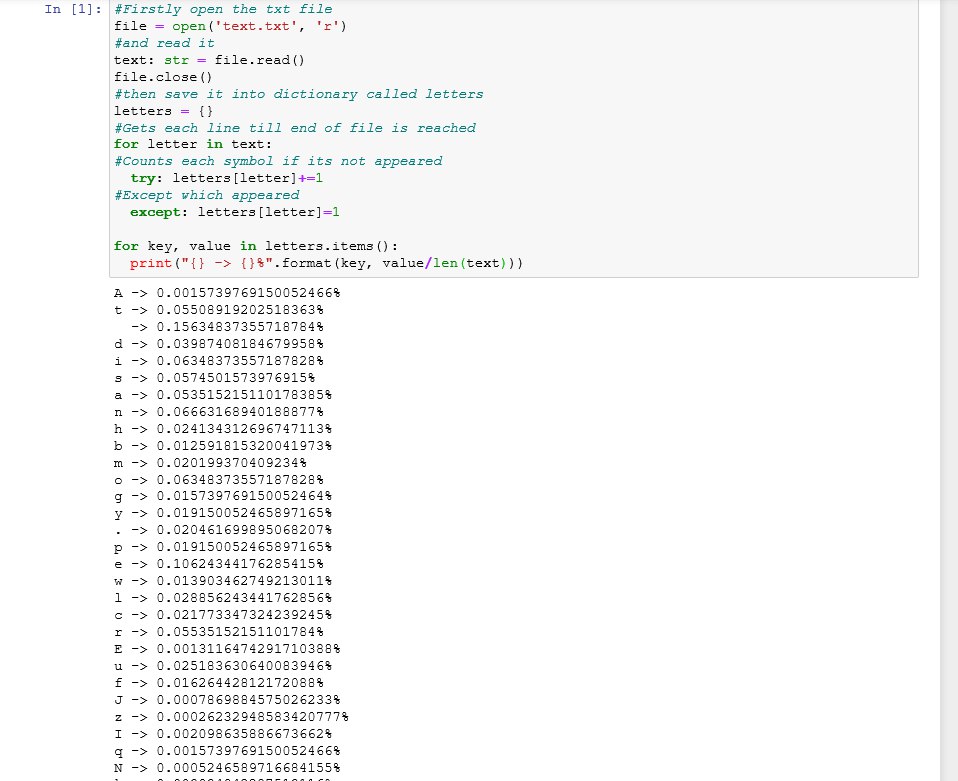
2- step. Then, we saved content in text as a dictionary called letters using this code:

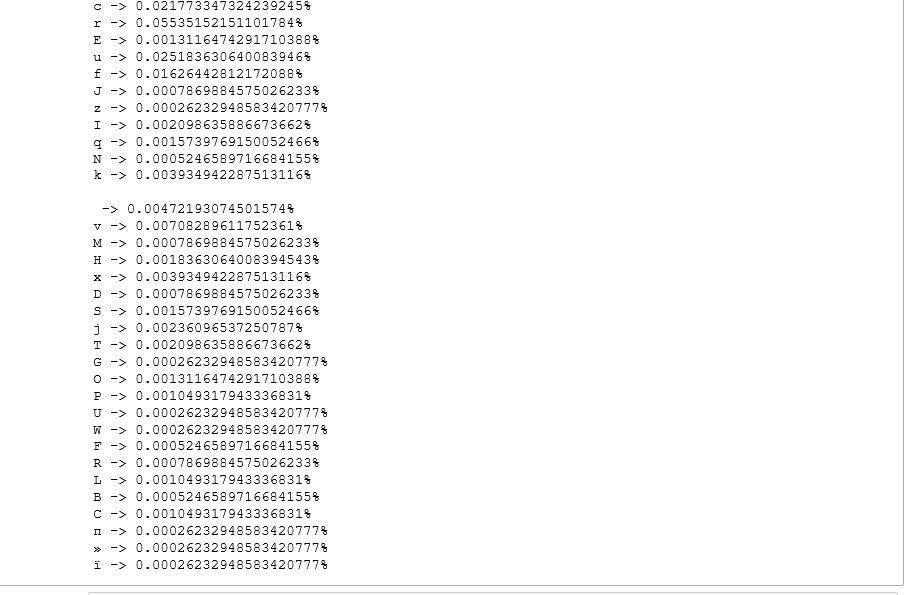
letters = {}

3 – step. Our solution need to get each line till end of file is reached, then get counts each symbol it is not appeared and except which appeared. Finally, we got the expected result.

for letter in text:   
try: letters[letter]+=1  
except: letters[letter]=1  
for key, value in letters.items():   
print("{} -> {}%".format(key, value/len(text))

Result of the second part:





To conclude, for this assignment our team considered more alternative solutions to this problem. In the end, we got our expected result, used two solution ways and got the same results.