Paweekorn Soratyathorn
65070501037

In [ ]:
```
from bs4 import BeautifulSoup
import requests

import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk import WordNetLemmatizer
```

In [ ]:
```
url = 'https://stackoverflow.com/questions/38488442/razor-framework-backend-or-fronten
page = requests.get(url)
soup = BeautifulSoup(page.text, 'html')
```

Question Part

In [ ]:
```
div_tags = soup.find_all('div', class_='s-prose js-post-body', attrs={'itemprop': 'tex
question = div_tags[0].text.strip('\n')
print(question)
```

```
What kind of framework is Razor? Is it backend or frontend?
What is the difference between the two types of frameworks?
I'm trying to learn a little bit more about backend and frontend frameworks and since
I usually work with Visual Studio Asp.net MVC was wondering about it.
```

Text Cleansing

1. Remove Punctuation

In [ ]:
```
def clean_punctuation(text):
    punctuation_pattern = re.compile(r'[^\w\s]|_')
    cleaned_text = re.sub(punctuation_pattern, '', text)

    return cleaned_text

question = clean_punctuation(question)
question = question.replace('\n', ' ')
question
```

Out[ ]:
```
'What kind of framework is Razor Is it backend or frontend What is the difference bet
ween the two types of frameworks Im trying to learn a little bit more about backend a
nd frontend frameworks and since I usually work with Visual Studio Aspnet MVC was won
dering about it'
```

1. Stopwords removal

In [ ]:
```
stop_words = stopwords.words('english')
question = re.sub('\n', '', question)
words = question.split(' ')

words = [word for word in words if word.lower() not in stop_words]
```

```
question = ' '.join(words)
print(question)
```

```
kind framework Razor backend frontend difference two types frameworks Im trying learn
little bit backend frontend frameworks since usually work Visual Studio Aspnet MVC wo
ndering
```

In [ ]:
```
question = question.replace('Im', '')
question = ' '.join([word for word in question.split(' ') if word != ''])
question
```

Out[ ]:
```
'kind framework Razor backend frontend difference two types frameworks trying learn l
ittle bit backend frontend frameworks since usually work Visual Studio Aspnet MVC won
dering'
```

1. Lemmatization

In [ ]:
```
lemmatizer = WordNetLemmatizer()
words = question.split(' ')
cleaned_text = ' '.join([lemmatizer.lemmatize(word, pos='n') for word in words])
cleaned_text = ' '.join([lemmatizer.lemmatize(word, pos='v') for word in words])
print(cleaned_text)
```

```
kind framework Razor backend frontend difference two type frameworks try learn little
bite backend frontend frameworks since usually work Visual Studio Aspnet MVC wonder
```

Data Masking

In [ ]:
```
def mask_framework(text):
    prog_pattern = re.compile(r'Razor|MVC')
    masked_text = re.sub(prog_pattern, '[FRAMEWORK]', text)

    return masked_text

def mask_microsoft(text):
    prog_pattern = re.compile(r'Visual Studio Aspnet')
    masked_text = re.sub(prog_pattern, '[MICROSOFT IDE]', text)

    return masked_text
```

In [ ]:
```
cleaned_question = mask_framework(cleaned_text)
cleaned_question = mask_microsoft(cleaned_question)
cleaned_question
```

Out[ ]:
```
'kind framework [FRAMEWORK] backend frontend difference two type frameworks try learn
little bite backend frontend frameworks since usually work [MICROSOFT IDE] [FRAMEWOR
K] wonder'
```

Answer Part

In [ ]:
```
ans_zone = soup.find_all('div', class_='answercell post-layout--right')
answers = [elem.find_all('p') for elem in ans_zone]
answers
```

Out[ ]:
```
[[<p>It is not a framework . I think you're misinterpreting certain concepts. Razor i
s a server side view engine, and it uses C # or VB.NET to generate dynamic content.</
p>,
  <p><a href="http://haacked.com/archive/2011/01/06/razor-syntax-quick-reference.asp
x/" rel="noreferrer">Razor Syntax Quick Reference</a></p>],
 [<p>This question is a couple of years old, but I'm going to add my two cents.</p>,
  <p>People struggle to give an answer to this question because the terms 'front-end'
and 'back-end' aren't formally defined anywhere. Because of that, any answer is purel
y subjective.</p>,
  <p>That being said, it is my <em>opinion</em> that the relationship between front-e
nd/back-end and client-side/server-side isn't necessarily one-to-one</p>,
  <p>I think it helps to think of it like this: <strong>client-side and server-side a
re run-times</strong>, while <strong>front-end and back-end are a separation of conce
rns</strong>.</p>,
  <p>'Client-side' always refers to code executing on the client's machine and 'serve
r-side' always refers to code executing on the server. A 'front-end' developer deals
with displaying data to the user and getting data from the user, while a back-end dev
eloper deals with storing, manipulating, and retrieving that data.</p>,
  <p>Consider a front-end developer who is tasked with building a UI. Much of the cod
e they write will be the typical HTML / CSS / JS. However, they will also have to dea
l with the data that is passed to the front-end from the backend. This is where Razor
comes into play. The front-end developer will write the Razor code (which executes on
the server-side) to display the data. </p>,
  <p>That is, <strong>the front-end developer will write server-side code to help gen
erate the UI, in addition to writing the client-side code that really defines the UI
</strong>.</p>,
  <p>Now, I can't imagine a scenario where a back-end developer will write client-sid
e code.</p>,
  <p>So, to answer your question, <strong>Razor is a front-end technology that execut
es on the server-side runtime</strong>. It's only purpose is to generate the UI, whic
h is the concern of the front-end.</p>],
 [<p>Razor is for writing dynamic html page which is front end and c# is for writing
backend logic. Although you could move all the backend logic inside razor but its hig
hly not recommended. </p>],
 [<p>Razor allows you on the <strong>back-end</strong> more easily create views (.csh
tml in C#).
  It is more like templating system...</p>,
  <p><a href="http://www.asp.net/web-pages/overview/getting-started/introducing-razor
-syntax-c" rel="nofollow">http://www.asp.net/web-pages/overview/getting-started/intro
ducing-razor-syntax-c</a></p>]]
```

Text Cleansing

1. Remove HTML Tags

In [ ]:
```python
def clean_html_tags(text):
    clean_text = text.get_text(separator=' ')
    clean_text = re.sub(r'\s+' , ' ' , clean_text).strip()

    return clean_text
```

In [ ]:
```python
answer = []
for list in answers:
    text = []
    for p in list:
        text.append(clean_html_tags(p))
    answer.append(' '.join(text))
```

```
answer = ' '.join(answer)
print(answer)
```

```
It is not a framework . I think you're misinterpreting certain concepts. Razor is a s
erver side view engine, and it uses C # or VB.NET to generate dynamic content. Razor 
Syntax Quick Reference This question is a couple of years old, but I'm going to add m
y two cents. People struggle to give an answer to this question because the terms 'fr
ont-end' and 'back-end' aren't formally defined anywhere. Because of that, any answer
is purely subjective. That being said, it is my opinion that the relationship between
front-end/back-end and client-side/server-side isn't necessarily one-to-one I think i
t helps to think of it like this: client-side and server-side are run-times , while f
ront-end and back-end are a separation of concerns . 'Client-side' always refers to c
ode executing on the client's machine and 'server-side' always refers to code executi
ng on the server. A 'front-end' developer deals with displaying data to the user and 
getting data from the user, while a back-end developer deals with storing, manipulati
ng, and retrieving that data. Consider a front-end developer who is tasked with build
ing a UI. Much of the code they write will be the typical HTML / CSS / JS. However, t
hey will also have to deal with the data that is passed to the front-end from the bac
kend. This is where Razor comes into play. The front-end developer will write the Raz
or code (which executes on the server-side) to display the data. That is, the front-e
nd developer will write server-side code to help generate the UI, in addition to writ
ing the client-side code that really defines the UI . Now, I can't imagine a scenario
where a back-end developer will write client-side code. So, to answer your question, 
Razor is a front-end technology that executes on the server-side runtime . It's only 
purpose is to generate the UI, which is the concern of the front-end. Razor is for wr
iting dynamic html page which is front end and c# is for writing backend logic. Altho
ugh you could move all the backend logic inside razor but its highly not recommended.
Razor allows you on the back-end more easily create views (.cshtml in C#). It is more
like templating system... http://www.asp.net/web-pages/overview/getting-started/intro
ducing-razor-syntax-c
```

1. Stop word Removal

In [ ]:
```
stop_words = stopwords.words('english')

words = answer.split(' ')
filtered_word = [word for word in words if word.lower() not in stop_words]
answer = ' '.join(filtered_word)
print(answer)
```

```
framework . think misinterpreting certain concepts. Razor server side view engine, us
es C # VB.NET generate dynamic content. Razor Syntax Quick Reference question couple
years old, I'm going add two cents. People struggle give answer question terms 'front
-end' 'back-end' formally defined anywhere. that, answer purely subjective. said, opi
nion relationship front-end/back-end client-side/server-side necessarily one-to-one t
hink helps think like this: client-side server-side run-times , front-end back-end se
paration concerns . 'Client-side' always refers code executing client's machine 'serv
er-side' always refers code executing server. 'front-end' developer deals displaying
data user getting data user, back-end developer deals storing, manipulating, retrievi
ng data. Consider front-end developer tasked building UI. Much code write typical HTM
L / CSS / JS. However, also deal data passed front-end backend. Razor comes play. fro
nt-end developer write Razor code (which executes server-side) display data. is, fron
t-end developer write server-side code help generate UI, addition writing client-side
code really defines UI . Now, can't imagine scenario back-end developer write client-
side code. So, answer question, Razor front-end technology executes server-side runti
me . purpose generate UI, concern front-end. Razor writing dynamic html page front en
d c# writing backend logic. Although could move backend logic inside razor highly rec
ommended. Razor allows back-end easily create views (.cshtml C#). like templating sys
tem... http://www.asp.net/web-pages/overview/getting-started/introducing-razor-syntax
-c
```

1. Remove Punctuation

In [ ]:
```python
def clean_punctuation(text):
    punctuation_pattern = re.compile(r'[^\w\s#]|_')
    cleaned_text = re.sub(punctuation_pattern, '', text)

    return cleaned_text

ans_no_punk = clean_punctuation(answer)
check = [word for word in ans_no_punk.split(' ') if word != '']
ans_no_punk = ' '.join(check)
print(ans_no_punk)
```

```
framework think misinterpreting certain concepts Razor server side view engine uses C
# VBNET generate dynamic content Razor Syntax Quick Reference question couple years o
ld Im going add two cents People struggle give answer question terms frontend backend
formally defined anywhere that answer purely subjective said opinion relationship fro
ntendbackend clientsideserverside necessarily onetoone think helps think like this cl
ientside serverside runtimes frontend backend separation concerns Clientside always r
efers code executing clients machine serverside always refers code executing server f
rontend developer deals displaying data user getting data user backend developer deal
s storing manipulating retrieving data Consider frontend developer tasked building UI
Much code write typical HTML CSS JS However also deal data passed frontend backend Ra
zor comes play frontend developer write Razor code which executes serverside display
data is frontend developer write serverside code help generate UI addition writing cl
ientside code really defines UI Now cant imagine scenario backend developer write cli
entside code So answer question Razor frontend technology executes serverside runtime
purpose generate UI concern frontend Razor writing dynamic html page front end c# wri
ting backend logic Although could move backend logic inside razor highly recommended
Razor allows backend easily create views cshtml C# like templating system httpwwwaspn
etwebpagesoverviewgettingstartedintroducingrazorsyntaxc
```

1. Lemmatization

In [ ]:
```python
lemmatizer = WordNetLemmatizer()
words = ans_no_punk.split(' ')
cleaned_text = ' '.join([lemmatizer.lemmatize(word, pos='n') for word in words])
cleaned_text = ' '.join([lemmatizer.lemmatize(word, pos='v') for word in words])
print(cleaned_text)
```

```
framework think misinterpret certain concepts Razor server side view engine use C # V
BNET generate dynamic content Razor Syntax Quick Reference question couple years old
Im go add two cents People struggle give answer question term frontend backend formal
ly define anywhere that answer purely subjective say opinion relationship frontendbac
kend clientsideserverside necessarily onetoone think help think like this clientside
serverside runtimes frontend backend separation concern Clientside always refer code
execute clients machine serverside always refer code execute server frontend develope
r deal display data user get data user backend developer deal store manipulate retrie
ve data Consider frontend developer task build UI Much code write typical HTML CSS JS
However also deal data pass frontend backend Razor come play frontend developer write
Razor code which execute serverside display data be frontend developer write serversi
de code help generate UI addition write clientside code really define UI Now cant ima
gine scenario backend developer write clientside code So answer question Razor fronte
nd technology execute serverside runtime purpose generate UI concern frontend Razor w
rite dynamic html page front end c# write backend logic Although could move backend l
ogic inside razor highly recommend Razor allow backend easily create view cshtml C# l
ike templating system httpwwwaspnetwebpagesoverviewgettingstartedintroducingrazorsynt
axc
```

Data Masking

In [ ]:
```
def mask_programming_lang(text):
    prog_pattern = re.compile(r'[cC].?#|\w+\.?(NET)|(net)|JS')
    masked_text = re.sub(prog_pattern, '[PROG_LANG]', text)

    return masked_text


def mask_website(text):
    prog_pattern = re.compile(r'http\w+')
    masked_text = re.sub(prog_pattern, '[WEBSITE]', text)

    return masked_text

def mask_framework(text):
    prog_pattern = re.compile(r'[rR]azor')
    masked_text = re.sub(prog_pattern, '[FRAMEWORK]', text)

    return masked_text
```

In [ ]:
```
cleaned_answer = mask_website(cleaned_text)
cleaned_answer = mask_programming_lang(cleaned_answer)
cleaned_answer = mask_framework(cleaned_answer)
print(cleaned_answer)
```

```
framework think misinterpret certain concepts [FRAMEWORK] server side view engine use
[PROG_LANG] [PROG_LANG] generate dynamic content [FRAMEWORK] Syntax Quick Reference q
uestion couple years old Im go add two cents People struggle give answer question ter
m frontend backend formally define anywhere that answer purely subjective say opinion
relationship frontendbackend clientsideserverside necessarily o[PROG_LANG]oone think
help think like this clientside serverside runtimes frontend backend separation conce
rn Clientside always refer code execute clients machine serverside always refer code
execute server frontend developer deal display data user get data user backend develo
per deal store manipulate retrieve data Consider frontend developer task build UI Muc
h code write typical HTML CSS [PROG_LANG] However also deal data pass frontend backen
d [FRAMEWORK] come play frontend developer write [FRAMEWORK] code which execute serve
rside display data be frontend developer write serverside code help generate UI addit
ion write clientside code really define UI Now cant imagine scenario backend develope
r write clientside code So answer question [FRAMEWORK] frontend technology execute se
rverside runtime purpose generate UI concern frontend [FRAMEWORK] write dynamic html
page front end [PROG_LANG] write backend logic Although could move backend logic insi
de [FRAMEWORK] highly recommend [FRAMEWORK] allow backend easily create view cshtml
[PROG_LANG] like templating system [WEBSITE]
```

In [ ]: