MSPA PREDICT 420

Extra Credit: Top Words

Introduction

This document presents the results of the extra credit exercise for the Masters of Science in Predictive Analytics course: PREDICT 420.

Assessment

1. Loading the Data

Load the dataset.

In [1]:

```
babble = []
f = open("data/babble-words.txt", "r")
babble = f.read()

print("babble[:500]:\n", babble[:500])
```

```
babble[:500]:
 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla ex risus, porta vitae nisl sit amet, lac
inia feugiat nunc. Duis auctor augue sit amet nulla ultrices ultrices. Sed posuere dictum purus non fauc
ibus. Nullam nec consequat urna. Nunc diam leo, luctus eu tincidunt at, laoreet ac lacus. Duis blandit l
ectus quis massa sagittis consequat. Donec semper quam at ultrices pretium. Morbi varius odio sit amet i
aculis imperdiet. Pellentesque a gravida turpis, eget molestie ligula. Donec lobortis
```

2. Pre-process the Data

Remove punctuation, remove non-printable characters and convert to lowercase.

In [2]:

```
import string

babble = "".join(filter(lambda x: x not in string.punctuation, babble)) # Remove punctuation.
babble = "".join(filter(lambda x: x in string.printable, babble)) # Remove non-printable charact
ers.
babble = babble.lower() # Convert to lowercase.

print("babble[:500]:\n", babble[:500])
```

```
babble[:500]:
 lorem ipsum dolor sit amet consectetur adipiscing elit nulla ex risus porta vitae nisl sit amet lacinia
 feugiat nunc duis auctor augue sit amet nulla ultrices ultrices sed posuere dictum purus non faucibus n
ullam nec consequat urna nunc diam leo luctus eu tincidunt at laoreet ac lacus duis blandit lectus quis
massa sagittis consequat donec semper quam at ultrices pretium morbi varius odio sit amet iaculis imperd
iet pellentesque a gravida turpis eget molestie ligula donec lobortis quis erat at bl
```

3. Word Count

Count up how many times each word occurs.

In [3]:

```
import pandas as pd

babblelist = babble.split() # Convert to dictonary.

worddict = {}
for w in babblelist: # Count words.
    try:
        worddict[w] += 1
    except KeyError:
        worddict[w] = 1

df_wordcounttemp = pd.DataFrame(worddict, index = [0]) # Convert to dataframe.
df_wordcount = df_wordcounttemp.transpose()
df_wordcount.columns = ["wordcount"]
```

In [4]:

```
df_wordcount.head(5)
```

Out[4]:

	wordcount
a	5
ac	7
accumsan	3
adipiscing	1
aenean	1

Output the ten (10) most frequently occurring words, indicating for each word how many times it occurred.

In [5]:

```
df_wordcount.sort_values(by = "wordcount", ascending = False, inplace = True) # Sort dataframe.
```

In [6]:

```
df_wordcount.head(10)
```

Out[6]:

	wordcount
sed	17
ut	15
in	11
nulla	10
amet	8
nec	8
turpis	8
nunc	8
sit	8
et	7