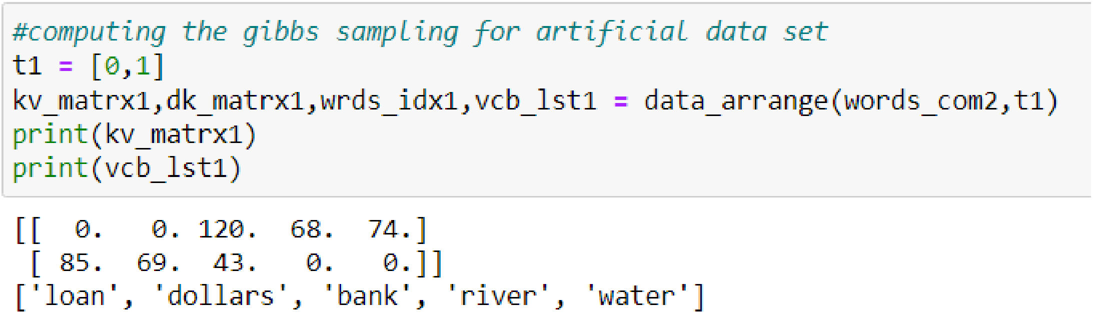
Programming Project 4

Task1: Gibbs Sampling

Testing implementation with artificial data set:

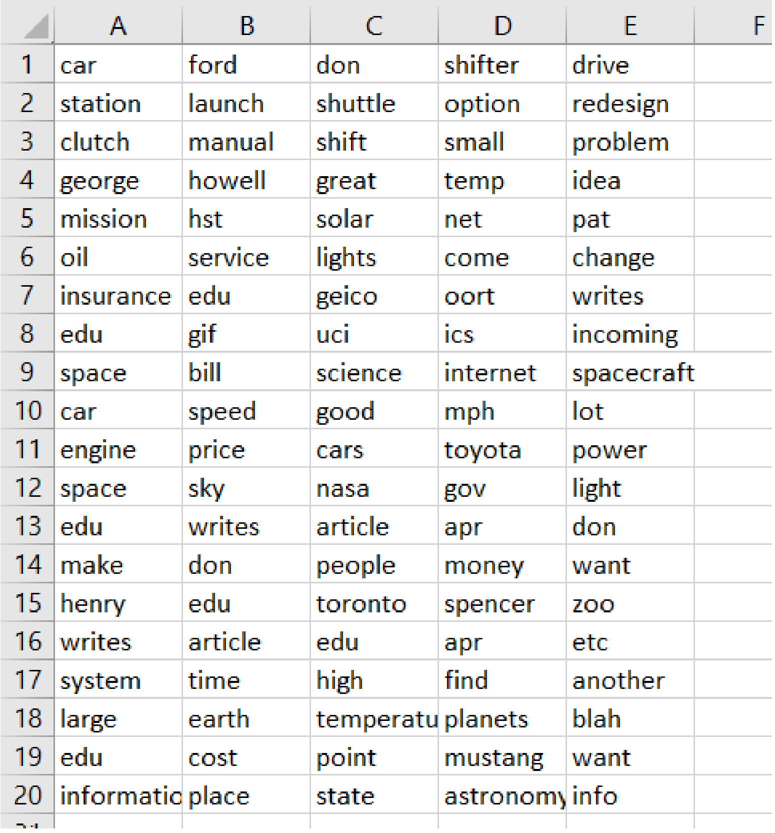


```
#computing the gibbs sampling for artificial data set
t1 = [0,1]
kv_matrx1,dk_matrx1,wrds_idx1,vcb_lst1 = data_arrange(words_com2,t1)
print(kv_matrx1)
print(vcb_lst1)
```

```
[[  0.   0. 120.  68.  74.]
 [ 85.  69.  43.   0.   0.]]
['loan', 'dollars', 'bank', 'river', 'water']
```

Here printed are the counts of KV matrix which has rows as topics and columns as words. From the matrix, it can be understood that bank, river, and water are the three most frequent words in topic1, and loan, dollars, and bank is the three most frequent words in topic2. Hence it proves the successful implementation of Gibbs sampling.

The below output is the one I got finally,



	A	B	C	D	E	F
1	car	ford	don	shifter	drive	
2	station	launch	shuttle	option	redesign	
3	clutch	manual	shift	small	problem	
4	george	howell	great	temp	idea	
5	mission	hst	solar	net	pat	
6	oil	service	lights	come	change	
7	insurance	edu	geico	oort	writes	
8	edu	gif	uci	ics	incoming	
9	space	bill	science	internet	spacecraft	
10	car	speed	good	mph	lot	
11	engine	price	cars	toyota	power	
12	space	sky	nasa	gov	light	
13	edu	writes	article	apr	don	
14	make	don	people	money	want	
15	henry	edu	toronto	spencer	zoo	
16	writes	article	edu	apr	etc	
17	system	time	high	find	another	
18	large	earth	temperatu	planets	blah	
19	edu	cost	point	mustang	want	
20	informatic	place	state	astronomy	info	

Each row is a topic, so 20 topics and the 5 most frequent words in those 20 topics are shown above.

Do the topics obtained make sense for the dataset?

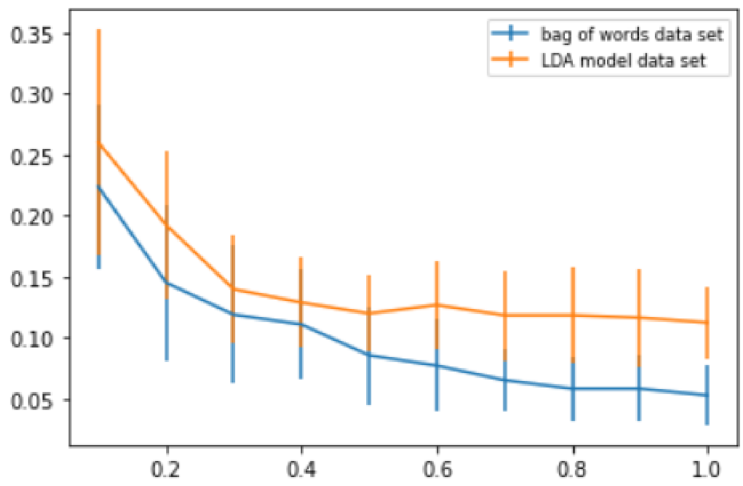
I have tried the sampling three times each of 500 iterations. I infer that some words always form the same topic. For example, edu, gif, uci, ics, and incoming always featured a single topic in all three topicwords.csv files. If you consider 1st topic, in which the words are more relevant and related. Car, ford, shifter, and drive all related to cars and came under a single topic. Similarly, the 11th topic words are also very related. The topic 13 words edu, write, article always occurred together in all the iterations, which shows that these are related in the documents under the same topics overall.

Also from topics, we could understand the overall dataset, from topics 20, 12, and 18 we can infer that one of the main topics in the data set is related to space. From topics, 11, 10, and 1 another main topic could be related to cars. And from 9, 18 rows another possibility could be related to general science. So, the overall topic does help to know what the data set could talk mainly about.

Task 2: Classification

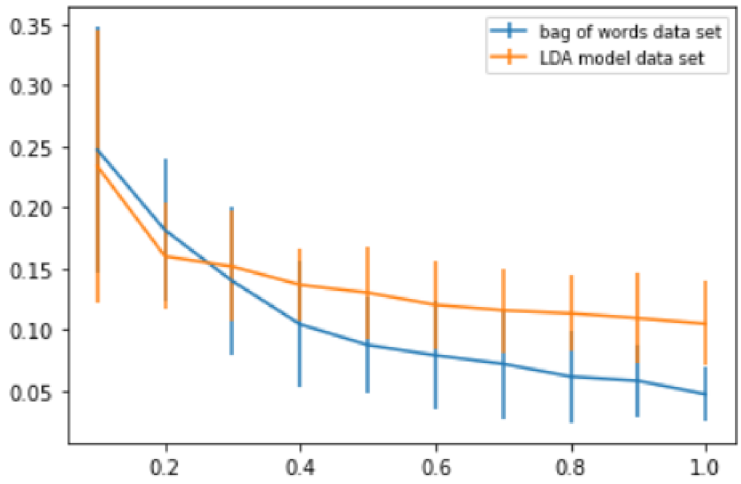
Plot the learning curve performance of the logistic regression algorithm (with error bars) on the two representations. Then discuss your observations on the results obtained, as well as the run time of the algorithms.

```
For data set 20newsgroups
```



When logistic regression alone ran again, the plot changed to,

```
For data set 20newsgroups
```

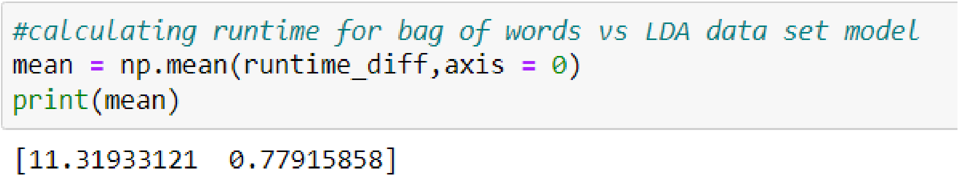


The plot with the x-axis as increasing training size and the y-axis as error rate shows the learning curve for the bag of words data set and LDA model data set.

For both data sets, as the training size increases the error rate decreases, and at the complete training, data set the error rate is very minimal on test data. Also, for both data sets, the variance of the error rate decreases as the training data size increases.

The LDA model has a slightly higher error rate compared to the bag of words model data set, but both data sets nearly have around 0.1 error rate with complete training size.

Runtime difference:



```
#calculating runtime for bag of words vs LDA data set model
mean = np.mean(runtime_diff,axis = 0)
print(mean)
```

```
[11.31933121  0.77915858]
```

For the Bag of the words data set, the time taken is 11 seconds but for the LDA model data set system took less than one second on average, to calculate the error rate per iteration of 30 iterations. So, the LDA model took nearly 1/10th of the time the bag of words model took. If there are very large data sets then the time difference would be so high. So, the LDA model would be very less time-consuming.