Assignment 5

Data Visualization:

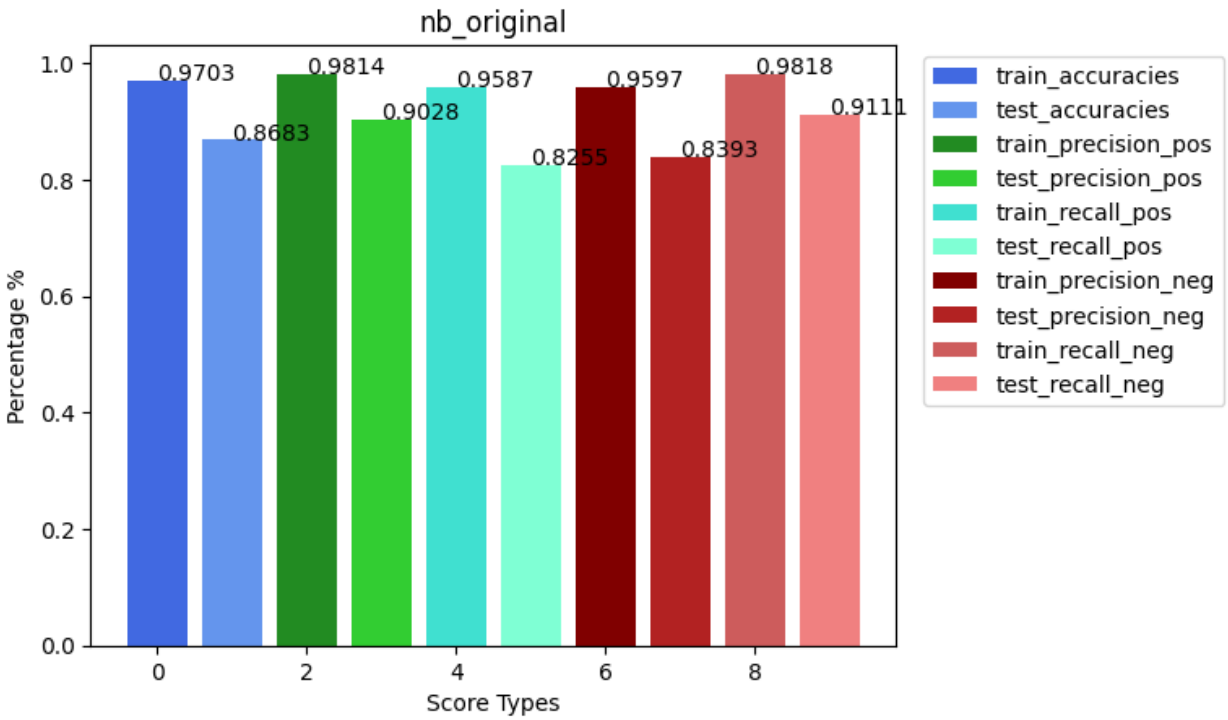
Positive Train Data:

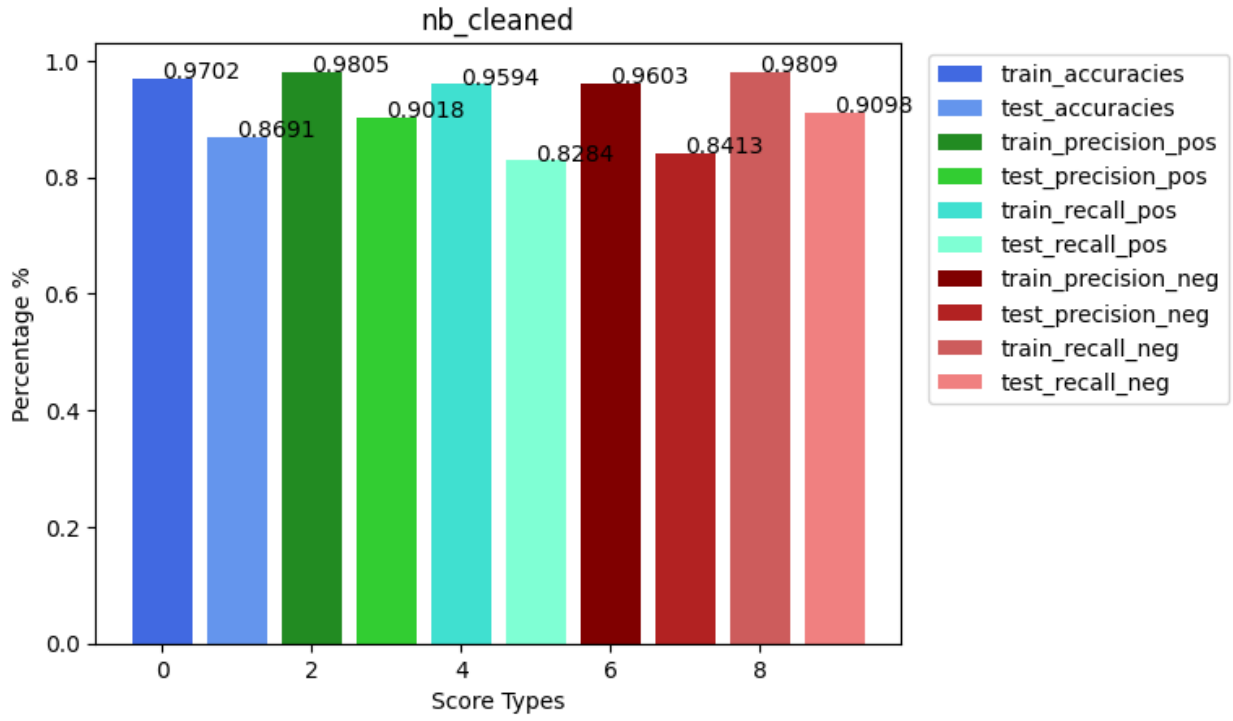
Positive Train Data	Original Words	Frequency	Cleaned Words	Frequency	No stop Words	Frequency	Lemmatized Words	Frequency
0	the	148404	the	149984	film	20335	film	24624
1	and	84265	and	86459	movie	18160	movi	21666
2	a	79423	a	80098	one	13177	one	13682
3	of	75339	of	75615	like	8879	like	10258
4	to	65208	to	65926	good	7552	time	8309
5	is	55356	is	56712	story	6672	good	7670
6	in	45791	in	46950	time	6352	stori	7362
7	that	31939	it	37826	great	6262	see	7229
8	I	28723	I	35603	well	6258	charact	7056
9	it	26976	that	34375	see	5838	make	6935
10	this	25957	s	33601	also	5536	well	6554
11	/><br	24618	this	27190	would	5351	get	6437
12	as	23928	as	24495	really	5308	great	6434
13	with	22030	with	22557	even	4935	watch	6171
14	was	21308	The	21780	much	4617	love	5921
15	for	20866	was	21778	first	4434	also	5536
16	but	16452	for	21390	people	4419	show	5512
17	his	16199	film	20395	get	4242	would	5351
18	The	15943	movie	18508	best	4220	realli	5308
19	on	15385	but	17102	love	4138	even	5088

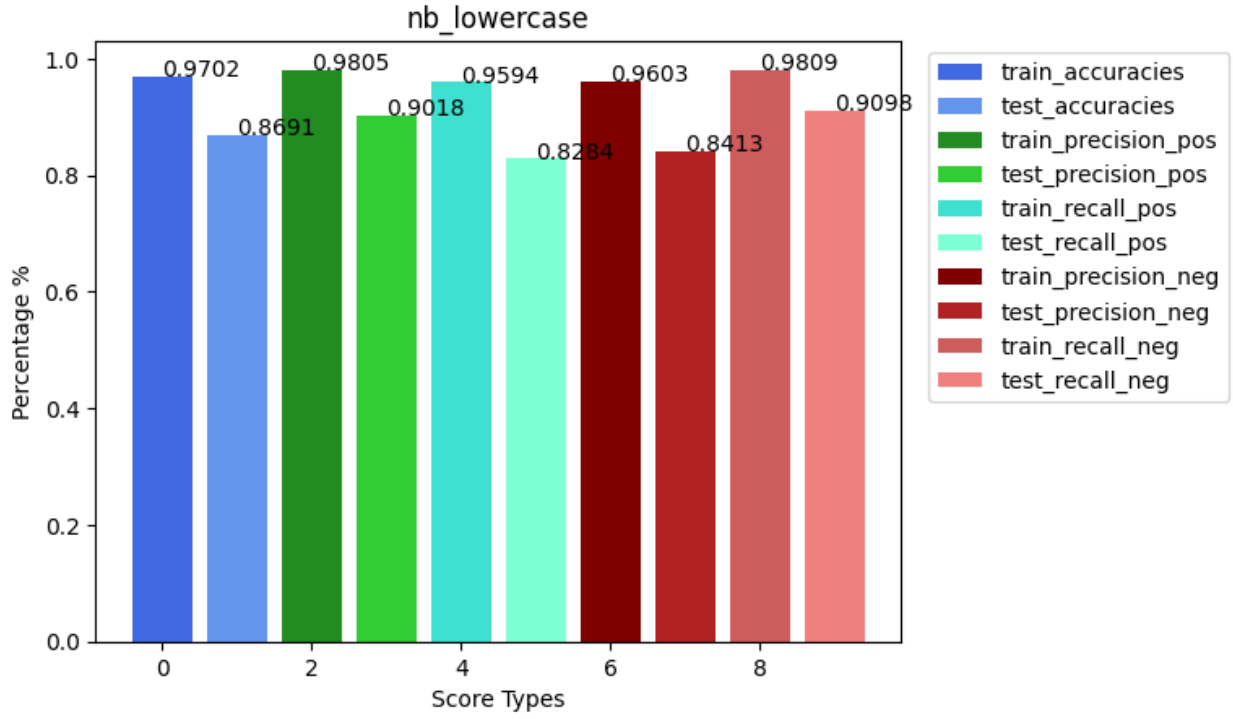
Negative Train Data:

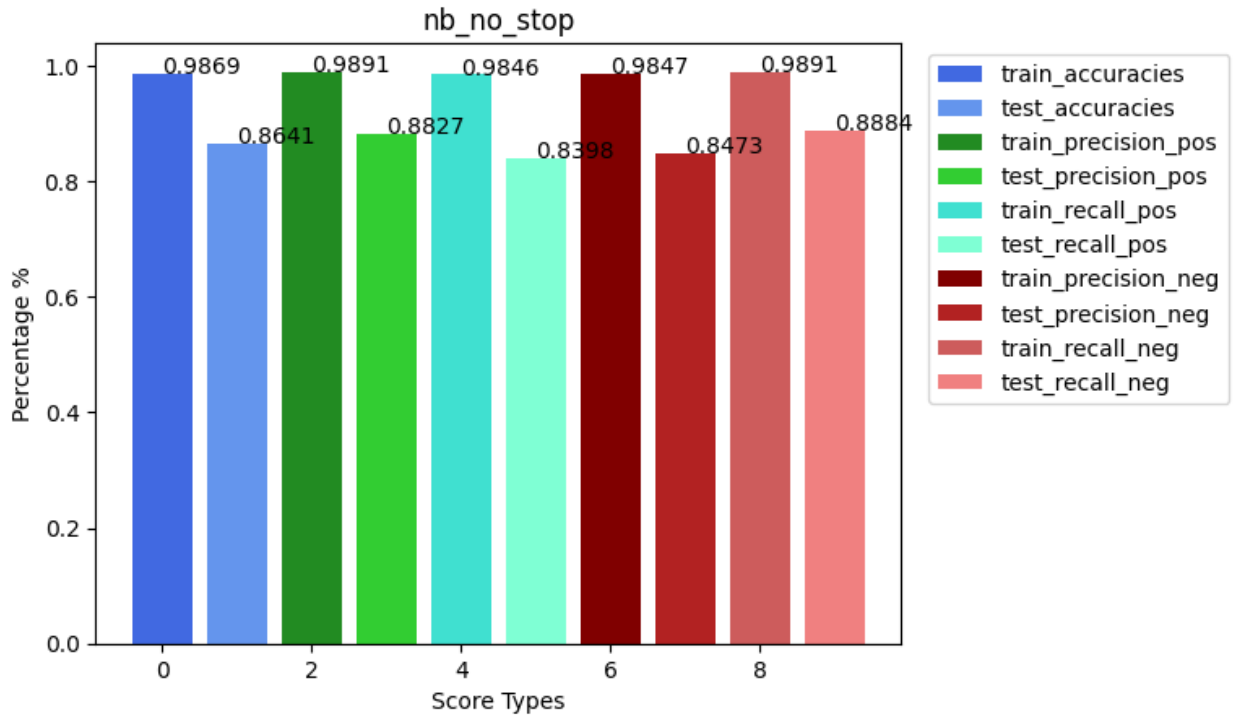
Negative Train Data	Original Words	Frequency	Cleaned Words	Frequency	No stop Words	Frequency	Lemmatized Words	Frequency
0	the	138587	the	140120	movie	23773	movi	27772
1	a	75661	a	76294	film	18580	film	22201
2	and	68373	and	70490	one	12671	one	13102
3	of	67629	to	68017	like	11089	like	12170
4	to	67356	of	67987	even	7629	make	8194
5	is	47869	is	49440	good	7306	bad	7838
6	in	39779	I	41460	bad	7244	even	7735
7	I	32791	in	40831	would	6989	get	7585
8	that	32613	it	38993	really	6087	time	7446
9	this	31102	that	35872	time	5971	good	7405
10	it	27425	this	33226	see	5360	charact	7081
11	/><br	26319	s	31589	story	5134	watch	7036
12	was	25387	was	26027	much	5003	would	6989
13	for	20193	movie	24140	get	4980	see	6491
14	with	19684	The	21662	people	4768	realli	6087
15	as	18575	for	20765	make	4694	look	5804
16	but	17325	with	20336	could	4638	stori	5616
17	movie	17118	t	20160	made	4478	scene	5566
18	The	16338	as	19078	plot	4095	act	5270
19	on	15379	film	18778	well	4080	much	5004

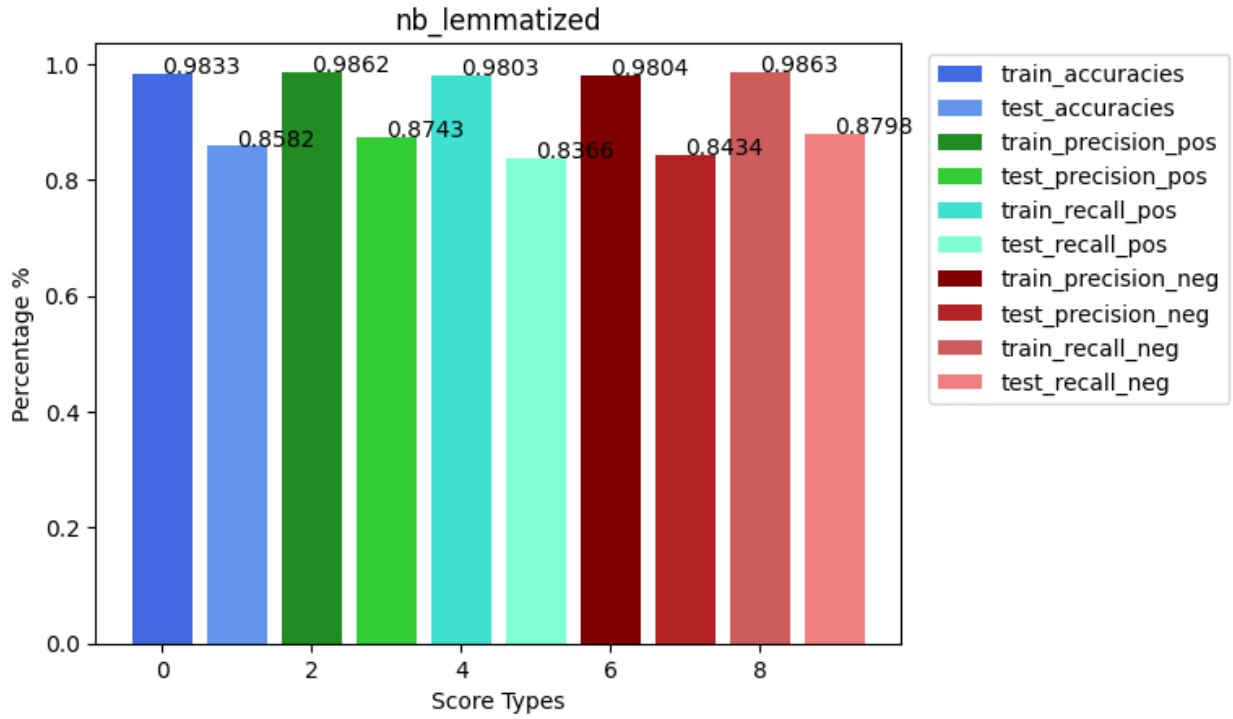
Part 1.7.) A comment on lemmatization: From what I can see of the lemmatized words, it was able to truncate some common words in an attempt to group them together by inflection, but an error that I could potentially see from this is that the base word is not fully recovered. From some of the examples of the words in the lemmatized words data set, I noticed that words like “movi” and “charact” are not complete words themselves. However, the program lemmatization seemed to parse through the majority of words within the first 20 entries without creating word fragments.


nb_original
Percentage %
Score Types
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
0.9703
0.8683
0.9814
0.9028
0.9587
0.8255
0.9597
0.8393
0.9818
0.9111
train_accuracies
test_accuracies
train_precision_pos
test_precision_pos
train_recall_pos
test_recall_pos
train_precision_neg
test_precision_neg
train_recall_neg
test_recall_neg


nb_cleaned
Percentage %
Score Types
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
0.9702
0.8691
0.9805
0.9018
0.9594
0.8284
0.9603
0.8413
0.9809
0.9098
train_accuracies
test_accuracies
train_precision_pos
test_precision_pos
train_recall_pos
test_recall_pos
train_precision_neg
test_precision_neg
train_recall_neg
test_recall_neg


nb_lowercase
Percentage %
Score Types
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
0.9702
0.8691
0.9805
0.9018
0.9594
0.8284
0.9603
0.8413
0.9809
0.9098
train_accuracies
test_accuracies
train_precision_pos
test_precision_pos
train_recall_pos
test_recall_pos
train_precision_neg
test_precision_neg
train_recall_neg
test_recall_neg





Part 2.3.) Each graph displayed above represents a different model used in the program, with five models in total. The five models represented are original words, cleaned words, lowercase words, no stop words, and lemmatized words. Each graph contains 10 values that evaluate the model based on its accuracy, precision detecting positive results, recall detecting positive

results, precision detecting negative results, and recall detecting negative results which were each evaluated on a training data set and a testing data set.

In terms of values, it appears that positive precision and negative recall typically had higher values than their counterparts (positive recall and negative precision). Additionally, it does seem that the test data found larger differences in percentage among the evaluation metrics compared to the train data.

Generally, the results look fairly consistent amongst all models, which indicates that no particular model seems to have a clear advantage in improving the evaluation metrics. Since English is morphologically a simple language, which means words and parts of words can easily combine together to create words with new meanings. Since lemmatization functions by splitting a word into its base components, the value of lemmatization increases the more morphologically simple a language is. English makes new words using base components, while lemmatization breaks down big words back into their base components to evaluate text.

With large dataset sizes, the value of preprocessing also increases. Since larger datasets will inherently contain more information, there will also be an influx of more “fuzzy” information with “useful” data. Preprocessing becomes much more important in order to remove the static data that might interfere with evaluations on accuracy, precision, and recall.

When reviews are written, it is also important to consider the capitalization of certain words within the review. Some examples that come to mind include the beginnings of sentences and pronouns, which could cause inconsistencies in text processing. In reviews, it seems authors tend to employ more professional, first-person perspective, and narrative writing styles which could increase the value of lowercasing, as more capital letter tend to show up in these writing styles.