Data Glacier

Your Deep Learning Partner

Final Project
Virtual Internship
Classification of Documents
Application of Machine Learning to Classify Emails into Newsgroups

By Betul Mescioglu
betulari@gmail.com
Data Science Graduate Student at Lewis University
Github: betulmesci/DataGlacier_Final_Project (github.com)
Project Deadline: 30-September-22
Submission Date: 27-September-22

PROJECT and DATA DESCRIPTION

Problem Description: In this project, we have a collection of approximately 20,000 emails sent to 20 newsgroups. Our job is to classify them into correct newsgroups with machine learning techniques. These newsgroups with their corresponding target values are:

0 - alt.atheism,
1 - comp.graphics,
2 - comp.os.ms-windows.misc,
3 - comp.sys.ibm.pc.hardware,
4 - comp.sys.mac.hardware,
5 - comp.windows.x,
6 - misc.forsale,
7 - rec.autos,
8 - rec.motorcycles,
9 - rec.sport.baseball,
10 - rec.sport.hockey,
11 - sci.crypt,
12 - sci.electronics,
13 - sci.med,
14 - sci.space,
15 - soc.religion.christian,
16 - talk.politics.guns,
17 - talk.politics.mideast,
18 - talk.politics.misc,
19 - talk.religion.misc

I obtained the data from http://qwone.com/~jason/20Newsgroups/. It came in two folders: training and testing. In each folder there were 20 folders, each of which corresponded to a newsgroup, containing emails sent to that newsgroup. In total, there were 18,846 emails. I formed a new dataset combining training and testing sets, did some exploration and preprocessing on this set, applied bag of words model then finally split it into train and test datasets (Last 10 data points were held to show the performance of the models).

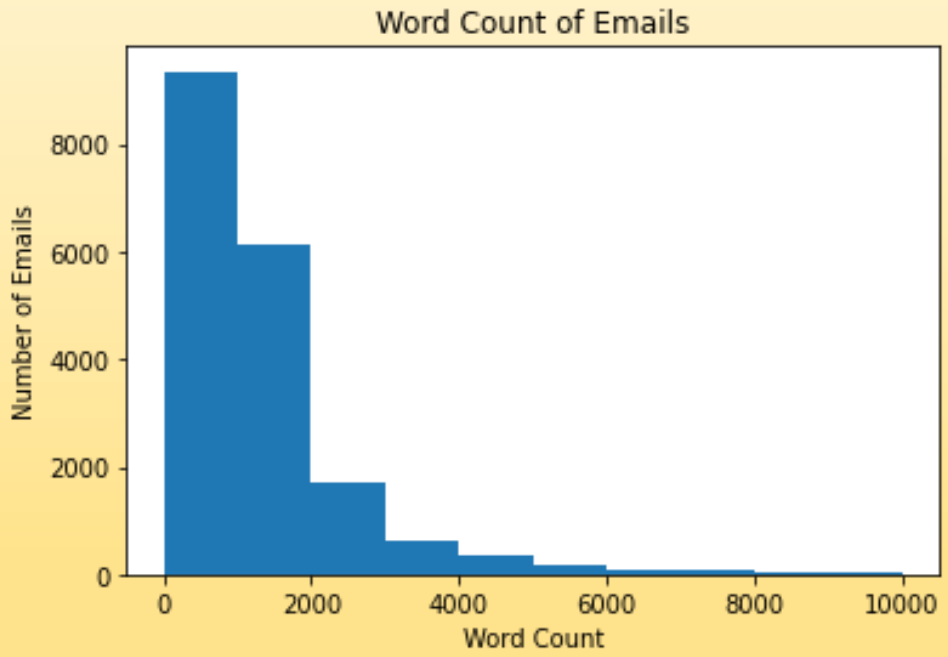
Data Glacier

Your Deep Learning Partner

EXPLORATORY DATA ANALYSIS

The dataset contains emails which is text data. It does not have any empty files. The shortest email is 107 words and longest is 67560 words long. The most frequent word count is 1000 or less. Very few of the emails contain more than 4000 words. On average, there are 1634 words in emails.

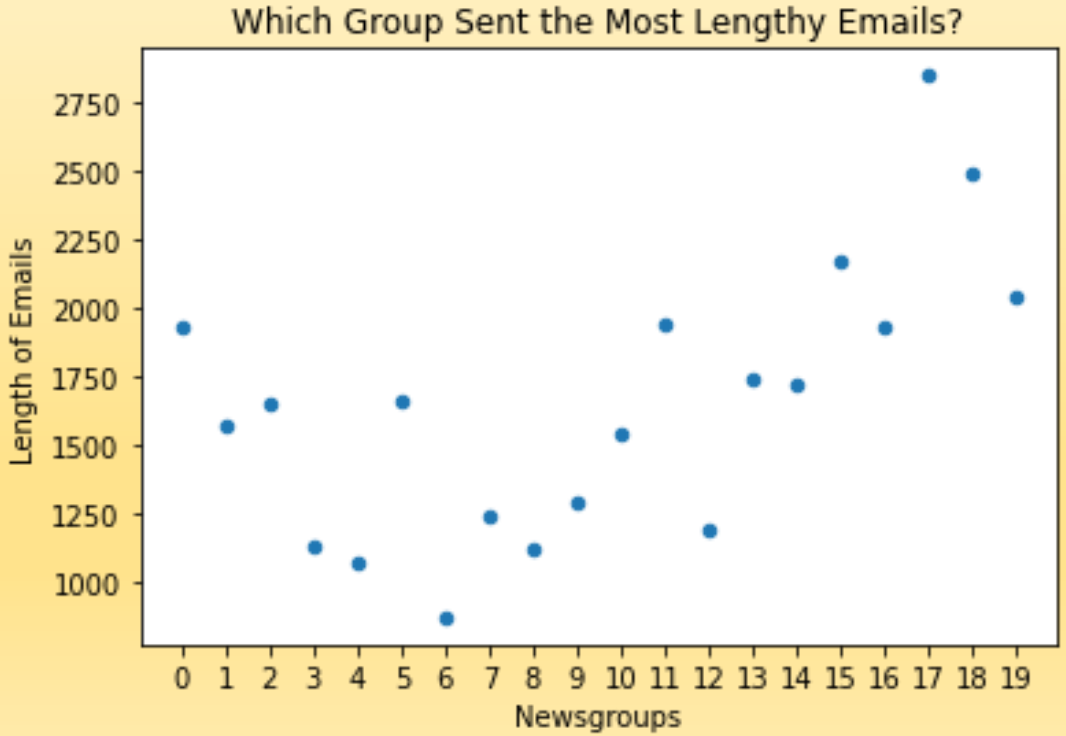
The following histogram shows the word count of the emails:



EXPLORATORY DATA ANALYSIS

I also explored whether there were any difference in email length between newsgroups. On average, Group 17, which corresponds to 'talk.politics.mideast' sent the most lengthy emails. 'talk.politics.misc' and 'talk.religion.misc' are the second and third runner-ups respectively. This is not surprising, as the politics and religion are two topics people are generally very passionate to talk about.

The following graph shows length of emails sent by newsgroups:



0 - alt.atheism,
1 - comp.graphics,
2 - comp.os.ms-windows.misc,
3 - comp.sys.ibm.pc.hardware,
4 - comp.sys.mac.hardware,
5 - comp.windows.x,
6 - misc.forsale,
7 - rec.autos,
8 - rec.motorcycles,
9 - rec.sport.baseball,
10 - rec.sport.hockey,
11 - sci.crypt,
12 - sci.electronics,
13 - sci.med,
14 - sci.space,
15 - soc.religion.christian,
16 - talk.politics.guns,
17 - talk.politics.mideast,
18 - talk.politics.misc,
19 - talk.religion.misc

DATA CLEANING

I encountered two problems in data:
One problem was, there were duplicate emails in group ‘misc.forsale’. So, I dropped one of them.

The other problem was, since these were emails, there were html tags throughout the text which had to be removed.
For example the following email:
“From: I3150101@dbstu1.rz.tu-bs.de (Benedikt Rosenau)\nSubject: Re: Gospel Dating\nOrganization: Technical University Braunschweig, Germany\nLines: 93\n\nIn article <65974@mimsy.umd.edu>\nmangoe@cs.umd.edu (Charley Wingate) writes:\n \n>>Well, John has a quite different, not necessarily more elaborated theology.\n>>There is some evidence that he must have known Luke, and that the content\n>>of Q was known to him, but not in a \'canonized\' form.\n>\n>”
Looked like this after cleaning:
“from i3150101 dbstu1 rz tu bs de benedikt rosenau subject re gospel dating organization technical university braunschweig germany lines 93 in article mangoe cs umd edu charley wingate writes well john has a quite different not necessarily more elaborated theology there is some evidence that he must have known luke and that the content of q was known to him but not in a canonized form ”

I also changed newsgroup names (target values) to numbers from 0 to 19.

DATA TRANSFORMATION

BAG of WORDS:

We need to convert each email into a vector containing numbers to make them suitable for machine learning algorithm. We used bag of words approach, where each unique word in a text is represented by one number. We tokenized words in text, removed stop words (commonly used English words that do not contribute in decision making) and normalized words into their base forms (lemmas). In vectorization process, first we count how many times a word occurs in each document, then weight the counts, so that frequent tokens get lower weight. Finally, normalize the vectors to unit length. Each vector has as many dimensions as there are unique words in the document. (CountVectorizer() was used in this process.)

As a second approach, we used TfidfVectorizer(). After cleaning emails. We applied this. Applying Logistic Regression to resulting sets, gave approximately same testing accuracy.

PROPOSED MODELS

We applied the following models:

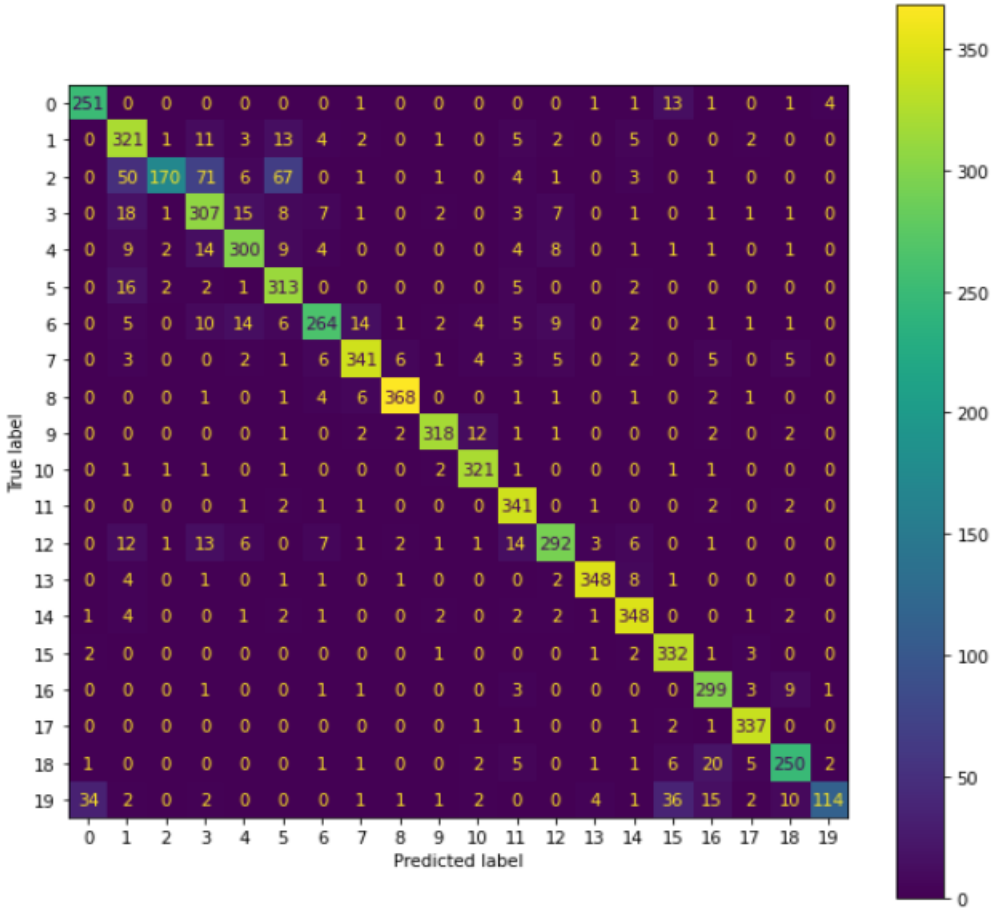
- Multinomial Naïve Bayes
- Logistic Regression
- Support Vector Classification
- Stochastic Gradient Descent
- K-Neighbors

PROPOSED MODELS: MULTINOMIAL NAÏVE BAYES

Multinomial Naïve Bayes Results with Confusion matrix:

Train accuracy score: 96.13%
Test accuracy score: 86.85%

	precision	recall	f1-score	support
0	0.87	0.92	0.89	273
1	0.72	0.87	0.79	370
2	0.96	0.45	0.61	375
3	0.71	0.82	0.76	373
4	0.86	0.85	0.85	354
5	0.74	0.92	0.82	341
6	0.88	0.78	0.82	339
7	0.91	0.89	0.90	384
8	0.97	0.95	0.96	386
9	0.96	0.93	0.95	341
10	0.93	0.97	0.95	330
11	0.86	0.97	0.91	351
12	0.88	0.81	0.85	360
13	0.97	0.95	0.96	367
14	0.90	0.95	0.93	367
15	0.85	0.97	0.90	342
16	0.84	0.94	0.89	318
17	0.95	0.98	0.96	343
18	0.88	0.85	0.86	295
19	0.94	0.51	0.66	225
accuracy			0.87	6834
macro avg	0.88	0.86	0.86	6834
weighted avg	0.88	0.87	0.86	6834



Data Glacier

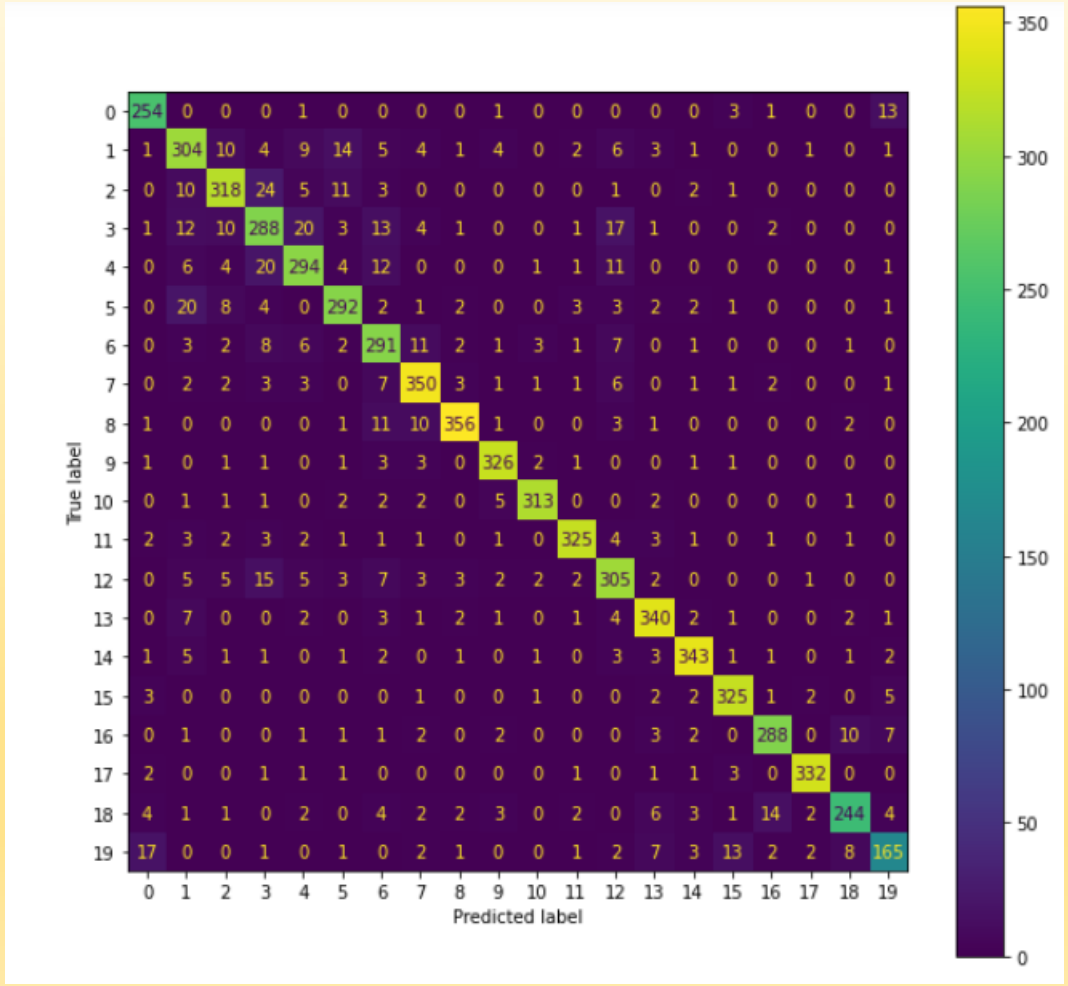
Your Deep Learning Partner

PROPOSED MODELS: LOGISTIC REGRESSION

Logistic Regression Results with Confusion matrix:

Train accuracy score: 100.00%
Test accuracy score: 88.57%

	precision	recall	f1-score	support
0	0.89	0.93	0.91	273
1	0.80	0.82	0.81	370
2	0.87	0.85	0.86	375
3	0.77	0.77	0.77	373
4	0.84	0.83	0.83	354
5	0.86	0.86	0.86	341
6	0.79	0.86	0.82	339
7	0.88	0.91	0.90	384
8	0.95	0.92	0.94	386
9	0.94	0.96	0.95	341
10	0.97	0.95	0.96	330
11	0.95	0.93	0.94	351
12	0.82	0.85	0.83	360
13	0.90	0.93	0.92	367
14	0.94	0.93	0.94	367
15	0.93	0.95	0.94	342
16	0.92	0.91	0.91	318
17	0.98	0.97	0.97	343
18	0.90	0.83	0.86	295
19	0.82	0.73	0.77	225
accuracy			0.89	6834
macro avg	0.89	0.88	0.88	6834
weighted avg	0.89	0.89	0.89	6834

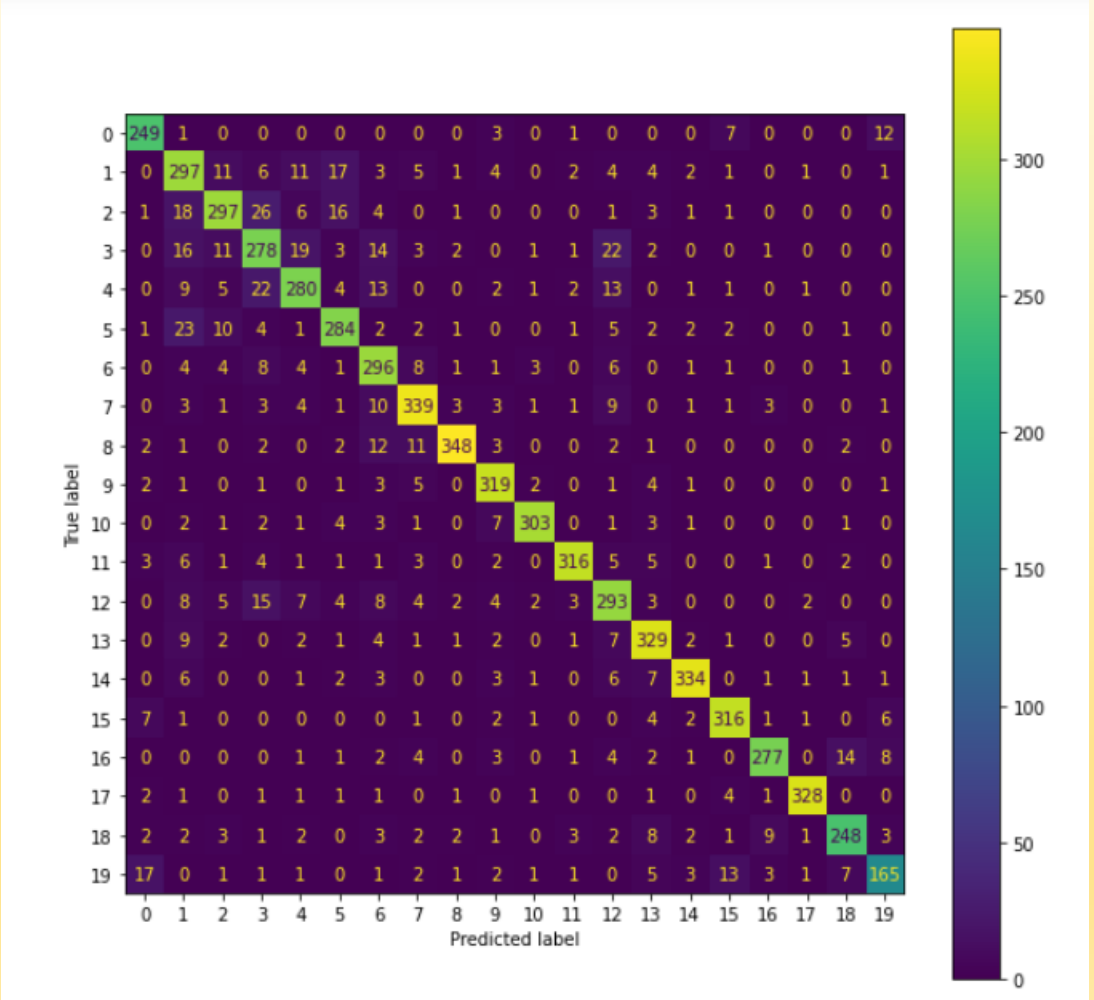


PROPOSED MODELS: SUPPORT VECTOR CLASSIFICATION

Support Vector Classification Results with Confusion matrix:

Train accuracy score: 99.76%
Test accuracy score: 86.27%

	precision	recall	f1-score	support
0	0.87	0.91	0.89	273
1	0.73	0.80	0.76	370
2	0.84	0.79	0.82	375
3	0.74	0.75	0.74	373
4	0.82	0.79	0.80	354
5	0.83	0.83	0.83	341
6	0.77	0.87	0.82	339
7	0.87	0.88	0.87	384
8	0.96	0.90	0.93	386
9	0.88	0.94	0.91	341
10	0.96	0.92	0.94	330
11	0.95	0.90	0.92	351
12	0.77	0.81	0.79	360
13	0.86	0.90	0.88	367
14	0.94	0.91	0.93	367
15	0.91	0.92	0.91	342
16	0.93	0.87	0.90	318
17	0.98	0.96	0.97	343
18	0.88	0.84	0.86	295
19	0.83	0.73	0.78	225
accuracy			0.86	6834
macro avg	0.87	0.86	0.86	6834
weighted avg	0.87	0.86	0.86	6834



Data Glacier

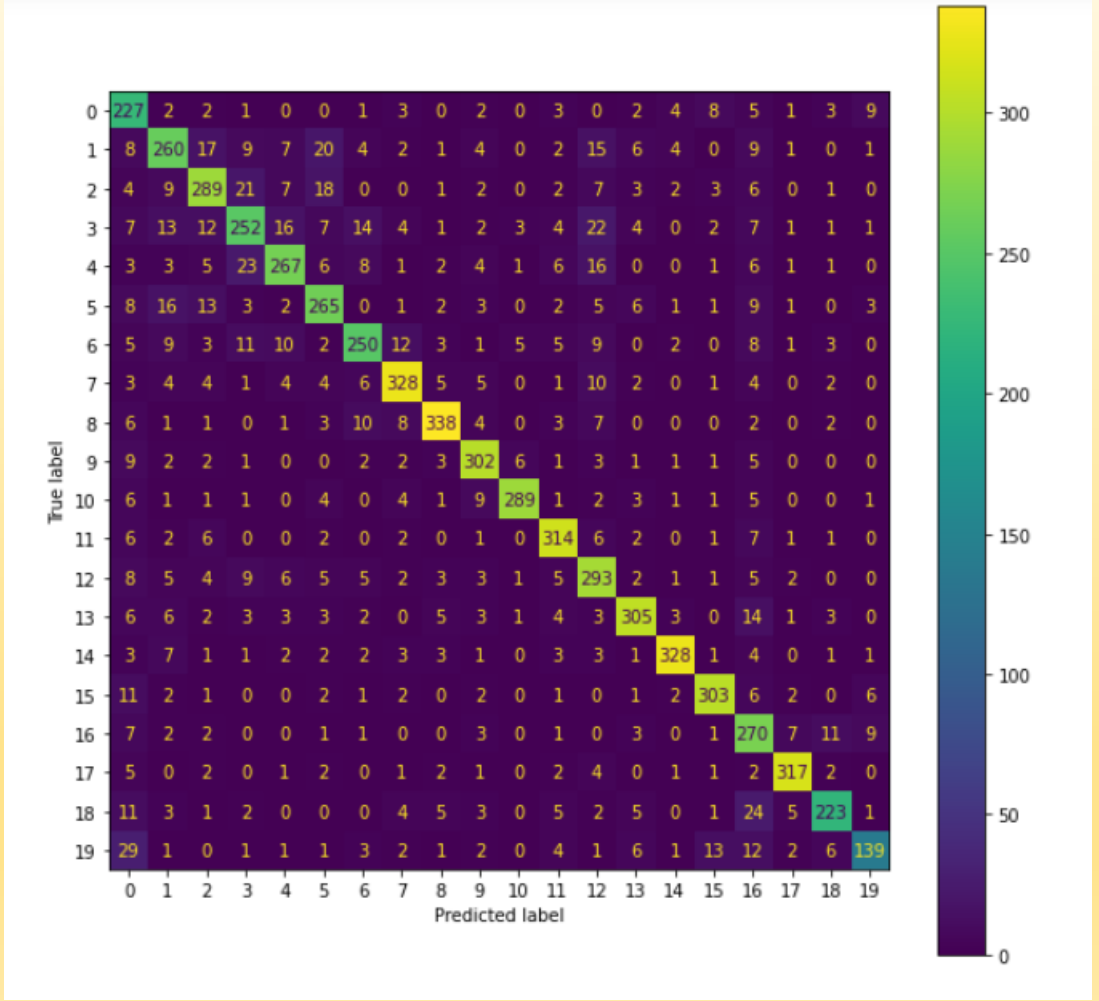
Your Deep Learning Partner

PROPOSED MODELS: STOCHASTIC GRADIENT DESCENT

Stochastic Gradient Descent Classification Results with Confusion matrix:

Train accuracy score: 99.78%
Test accuracy score: 81.34%

	precision	recall	f1-score	support
0	0.61	0.83	0.70	273
1	0.75	0.70	0.72	370
2	0.79	0.77	0.78	375
3	0.74	0.68	0.71	373
4	0.82	0.75	0.78	354
5	0.76	0.78	0.77	341
6	0.81	0.74	0.77	339
7	0.86	0.85	0.86	384
8	0.90	0.88	0.89	386
9	0.85	0.89	0.87	341
10	0.94	0.88	0.91	330
11	0.85	0.89	0.87	351
12	0.72	0.81	0.76	360
13	0.87	0.83	0.85	367
14	0.93	0.89	0.91	367
15	0.89	0.89	0.89	342
16	0.66	0.85	0.74	318
17	0.92	0.92	0.92	343
18	0.86	0.76	0.80	295
19	0.81	0.62	0.70	225
accuracy			0.81	6834
macro avg	0.82	0.81	0.81	6834
weighted avg	0.82	0.81	0.81	6834

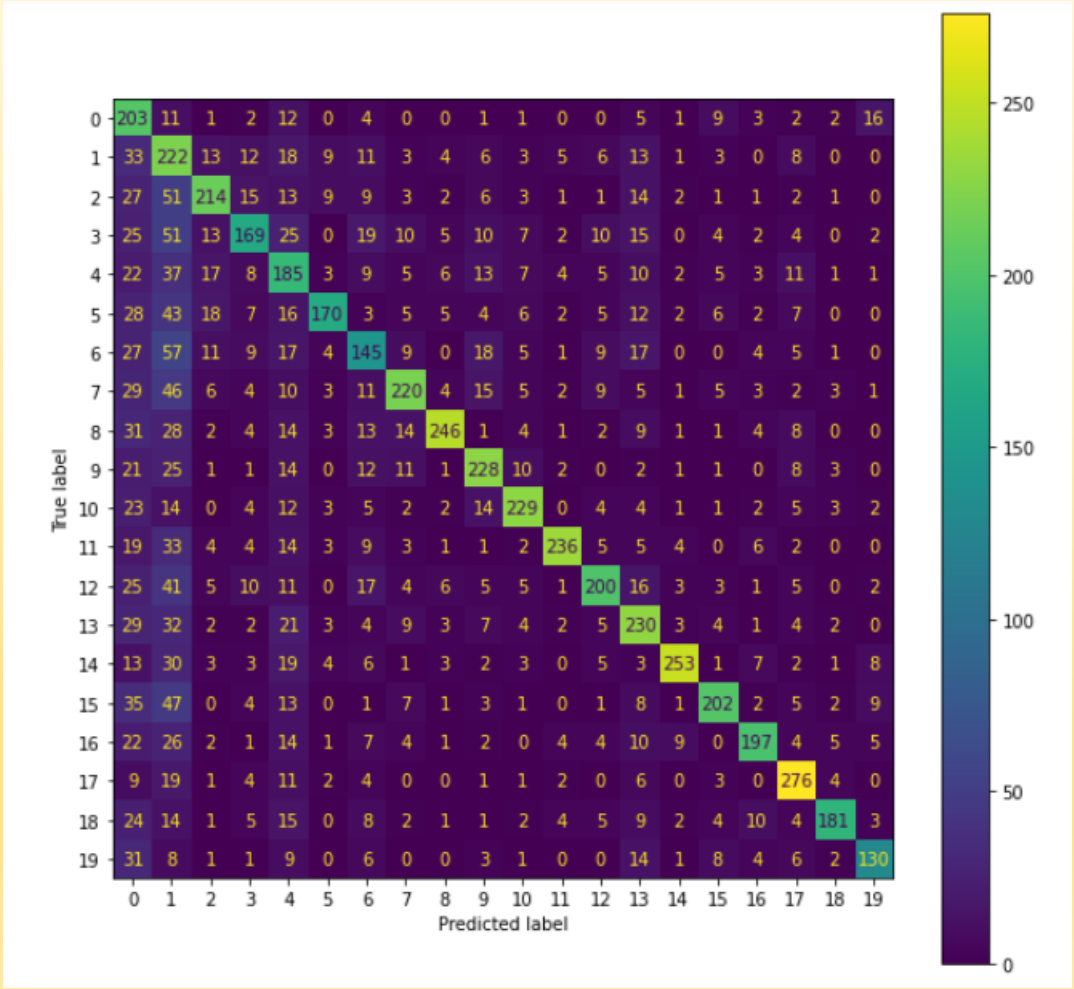


PROPOSED MODELS: K-NEIGHBORS

K-Neighbors Classification Results with Confusion matrix:

Train accuracy score: 100.00%
Test accuracy score: 60.52%

	precision	recall	f1-score	support
0	0.30	0.74	0.43	273
1	0.27	0.60	0.37	370
2	0.68	0.57	0.62	375
3	0.63	0.45	0.53	373
4	0.40	0.52	0.45	354
5	0.78	0.50	0.61	341
6	0.48	0.43	0.45	339
7	0.71	0.57	0.63	384
8	0.85	0.64	0.73	386
9	0.67	0.67	0.67	341
10	0.77	0.69	0.73	330
11	0.88	0.67	0.76	351
12	0.72	0.56	0.63	360
13	0.57	0.63	0.59	367
14	0.88	0.69	0.77	367
15	0.77	0.59	0.67	342
16	0.78	0.62	0.69	318
17	0.75	0.80	0.77	343
18	0.86	0.61	0.72	295
19	0.73	0.58	0.64	225
accuracy			0.61	6834
macro avg	0.67	0.61	0.62	6834
weighted avg	0.67	0.61	0.62	6834



Conclusion

	Training Accuracy	Testing Accuracy
Multinomial Naïve Bayes	95.62%	86.65%
Logistic Regression	99.99%	87.78%
Support Vector Classification	99.72%	86.16%
Stochastic Gradient Descent	98.42%	80.93%
K-Neighbors	100%	61.33%

It seems Logistic Regression is the best performing algorithm with almost 88% and 100% accuracy with testing and training sets respectively. Second best would be Support Vector Classification, followed by Multinomial Naive Bayes. The worst performing was K-Neighbors. It's probably due to the curse of dimensionality. K-Neighbors performs best with fewer number of features. If the number of features is high, it requires more data. When there is more data, it is prone to overfitting, which is exactly what we see here with 100% training accuracy and only 61.33% testing accuracy. This is due to the fact that we have 130,656 features which is enormous.