Project Report - Group 10

Letter Recognition

- C.Sravan Chaitanya - S20190020205
- K.Pradyumna Reddy - S20190020226
- B.Manish Kumar - S20190020203

Appendix

Introduction:

The letter recognition has been one of the most popular research Area in pattern recognition in recent years . It is a task which uses Artificial intelligence to identify the letters . It is a task which requires a complex hypothesis to increase the accuracy ,precision and efficiency. We plotted confusion matrix of different types of Classifications like Gaussian , Decision Tree , SVM , KNN , MLP .

Data set information:

The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15. We typically train on the first 16000 items and then use the resulting model to predict the letter category for the remaining 4000

Project:

The dataset has been downloaded from UCI ([Link]) repository . The dataset contains:

1. No. of attributes : 16

2. No. of instances : 20000

3. Type of problem: classification.

Attribute Information:

1. lettr capital letter (26 values from A to Z)
2. x-box horizontal position of box (integer)
3. y-box vertical position of box (integer)
4. width width of box (integer)
5. high height of box (integer)
6. onpix total # on pixels (integer)
7. x-bar mean x of on pixels in box (integer)
8. y-bar mean y of on pixels in box (integer)
9. x2bar mean x variance (integer)
10. y2bar mean y variance (integer)

11. xybar mean x y correlation (integer)
12. x2ybr mean of x * x * y (integer)
13. xy2br mean of x * y * y (integer)
14. x-ege mean edge count left to right (integer)
15. xegvy correlation of x-ege with y (integer)
16. y-ege mean edge count bottom to top (integer)
17. yegvx correlation of y-ege with x (integer).

Libraries :

Pandas , YellowBrick

Methodology :

1. We converted data into .csv file and loaded the provided dataset into the code (mention the path of the data set)
2. Form a set using the above mentioned attributes
3. Now , using pandas we load the values into a variable .
4. We can see the shape of the data matrix formed and can plot it .
5. Now we take X,Y for matrix and vector and split the data into a

   a training set and testing set

6. Based on the type of classification , we name it and form

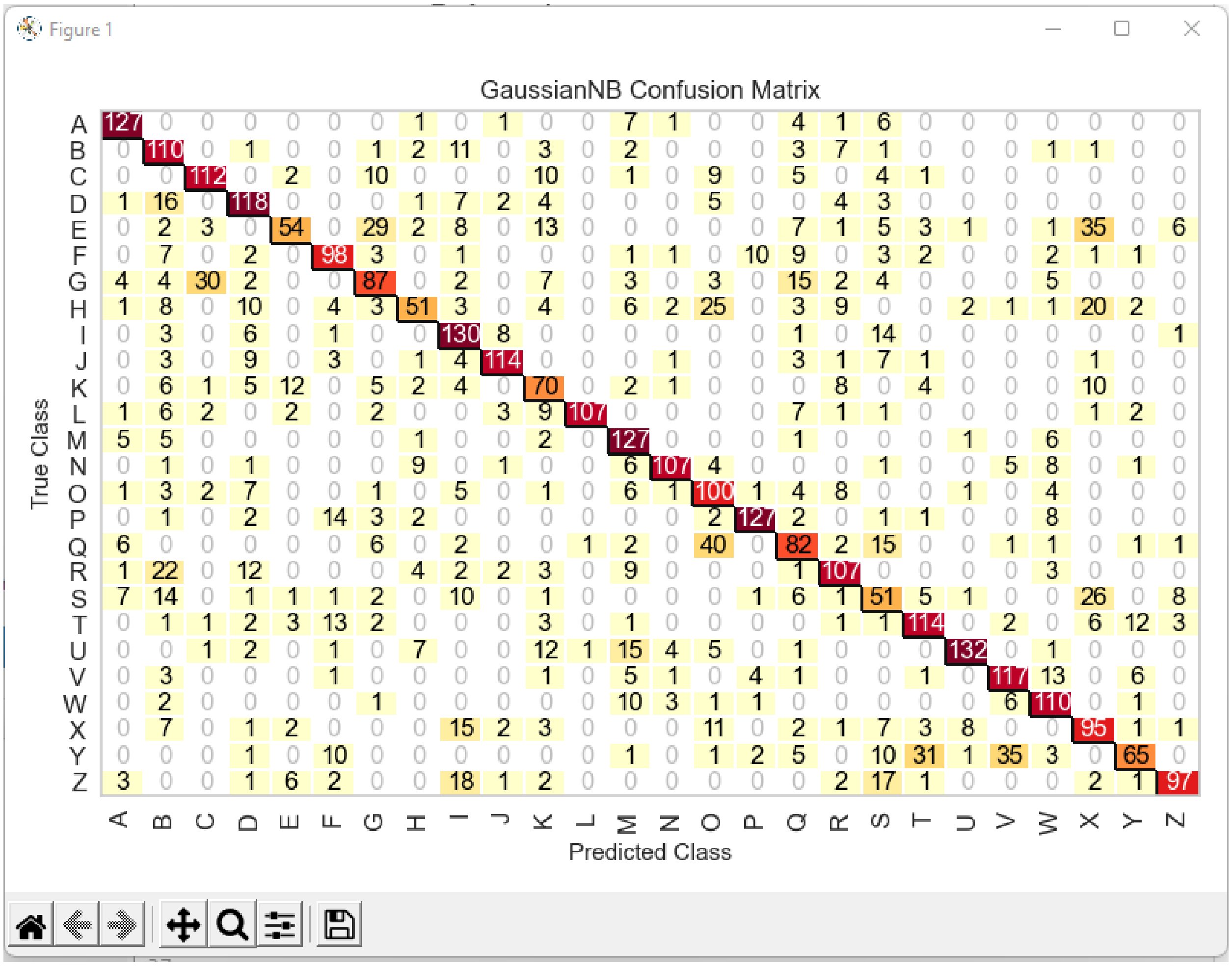
   Confusion matrix for it .

7. Train , predict and printing the confusion matrix and accuracy

   Takes place

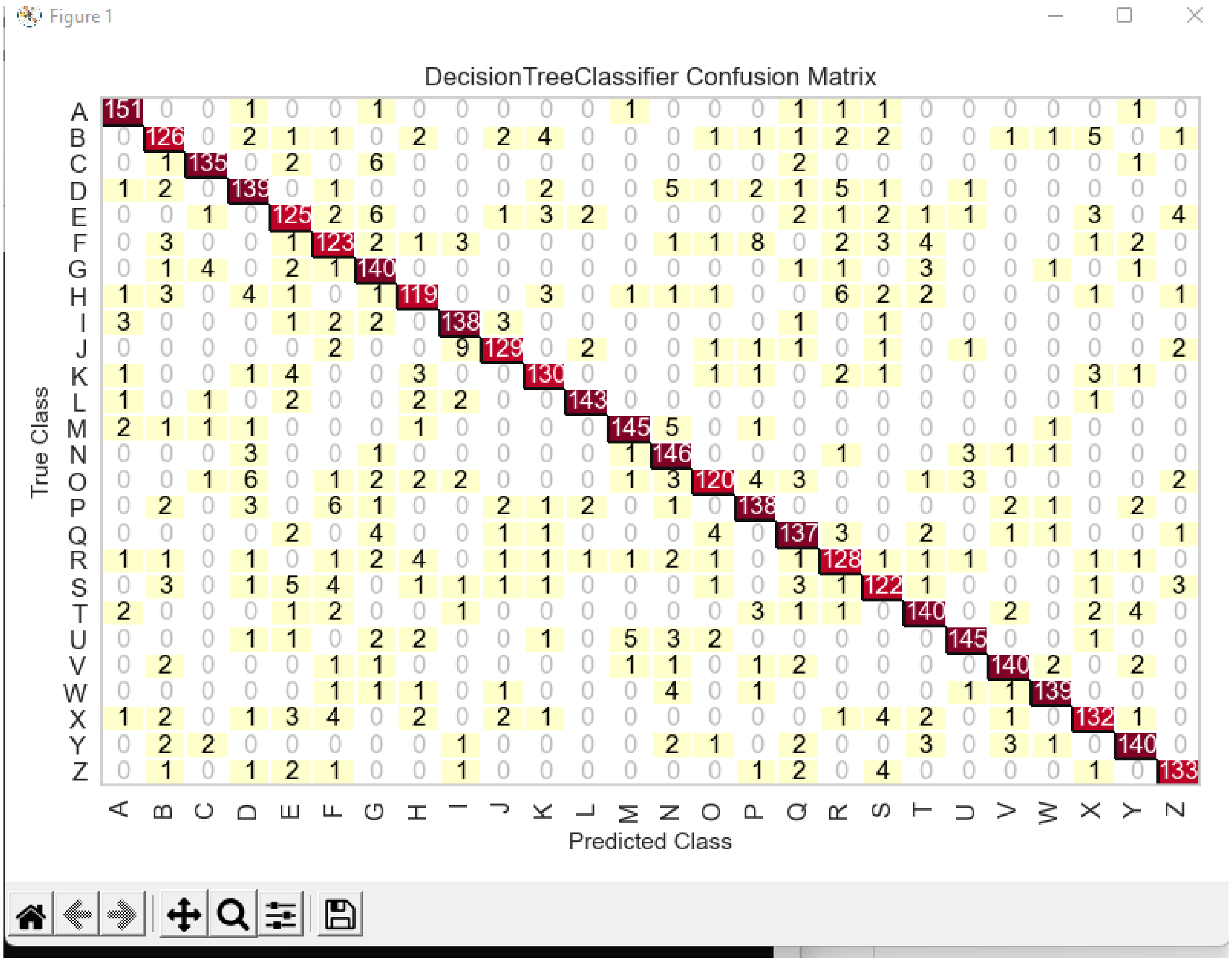
Results

GaussianNB



	precision	recall	f1-score	support
A	0.80892	0.85811	0.83279	148
B	0.49107	0.76923	0.59946	143
C	0.73684	0.72727	0.73203	154
D	0.64481	0.73292	0.68605	161
E	0.65854	0.31765	0.42857	170
F	0.66216	0.69504	0.67820	141
G	0.56129	0.51786	0.53870	168
H	0.61446	0.32903	0.42857	155
I	0.58559	0.79268	0.67358	164
J	0.85075	0.77027	0.80851	148
K	0.47297	0.53846	0.50360	130
L	0.98165	0.74306	0.84585	144
M	0.62255	0.85811	0.72159	148
N	0.87705	0.74306	0.80451	144
O	0.48544	0.68966	0.56980	145
P	0.86986	0.77914	0.82201	163
Q	0.50617	0.51250	0.50932	160
R	0.68590	0.64458	0.66460	166
S	0.33775	0.37500	0.35540	136
T	0.68263	0.69091	0.68675	165
U	0.89796	0.72527	0.80243	182
V	0.70060	0.76471	0.73125	153
W	0.65868	0.81481	0.72848	135
X	0.47980	0.59748	0.53221	159
Y	0.69892	0.39394	0.50388	165
Z	0.82906	0.63399	0.71852	153
accuracy			0.65225	4000
macro avg	0.66929	0.65441	0.65025	4000
weighted avg	0.67222	0.65225	0.65041	4000

Decision Tree



	precision	recall	f1-score	support
A	0.92073	0.95570	0.93789	158
B	0.84000	0.82353	0.83168	153
C	0.93103	0.91837	0.92466	147
D	0.84242	0.86335	0.85276	161
E	0.81699	0.81169	0.81433	154
F	0.80392	0.79355	0.79870	155
G	0.81395	0.90323	0.85627	155
H	0.85000	0.80952	0.82927	147
I	0.87342	0.91391	0.89320	151
J	0.90210	0.86577	0.88356	149
K	0.87838	0.87838	0.87838	148
L	0.95333	0.94079	0.94702	152
M	0.92949	0.91772	0.92357	158
N	0.83908	0.92994	0.88218	157
O	0.88889	0.79470	0.83916	151
P	0.85185	0.85714	0.85449	161
Q	0.85093	0.87261	0.86164	157
R	0.82581	0.84768	0.83660	151
S	0.84138	0.81879	0.82993	149
T	0.87500	0.88050	0.87774	159
U	0.92949	0.88957	0.90909	163
V	0.92105	0.91503	0.91803	153
W	0.93919	0.92667	0.93289	150
X	0.86842	0.84076	0.85437	157
Y	0.89744	0.89172	0.89457	157
Z	0.90476	0.90476	0.90476	147
accuracy			0.87575	4000
macro avg	0.87650	0.87559	0.87564	4000
weighted avg	0.87638	0.87575	0.87566	4000

SVM

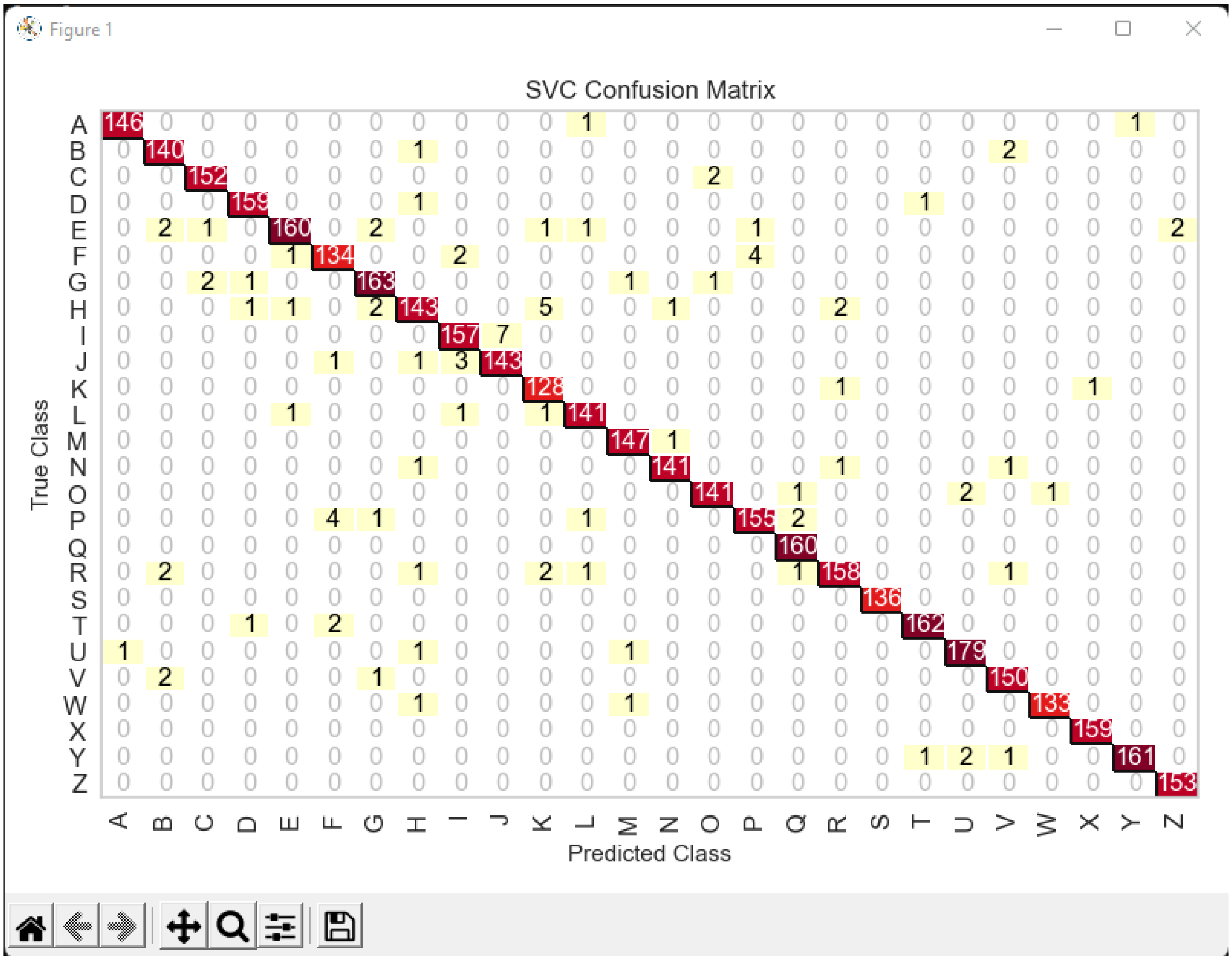


Figure 1
SVC Confusion Matrix
True Class
Predicted Class
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

	precision	recall	f1-score	support
A	0.99320	0.98649	0.98983	148
B	0.95890	0.97902	0.96886	143
C	0.98065	0.98701	0.98382	154
D	0.98148	0.98758	0.98452	161
E	0.98160	0.94118	0.96096	170
F	0.95035	0.95035	0.95035	141
G	0.96450	0.97024	0.96736	168
H	0.95333	0.92258	0.93770	155
I	0.96319	0.95732	0.96024	164
J	0.95333	0.96622	0.95973	148
K	0.93431	0.98462	0.95880	130
L	0.97241	0.97917	0.97578	144
M	0.98000	0.99324	0.98658	148
N	0.98601	0.97917	0.98258	144
O	0.97917	0.97241	0.97578	145
P	0.96875	0.95092	0.95975	163
Q	0.97561	1.00000	0.98765	160
R	0.97531	0.95181	0.96341	166
S	1.00000	1.00000	1.00000	136
T	0.98780	0.98182	0.98480	165
U	0.97814	0.98352	0.98082	182
V	0.96774	0.98039	0.97403	153
W	0.99254	0.98519	0.98885	135
X	0.99375	1.00000	0.99687	159
Y	0.99383	0.97576	0.98471	165
Z	0.98710	1.00000	0.99351	153
accuracy			0.97525	4000
macro avg	0.97512	0.97561	0.97528	4000
weighted avg	0.97534	0.97525	0.97521	4000

MLP

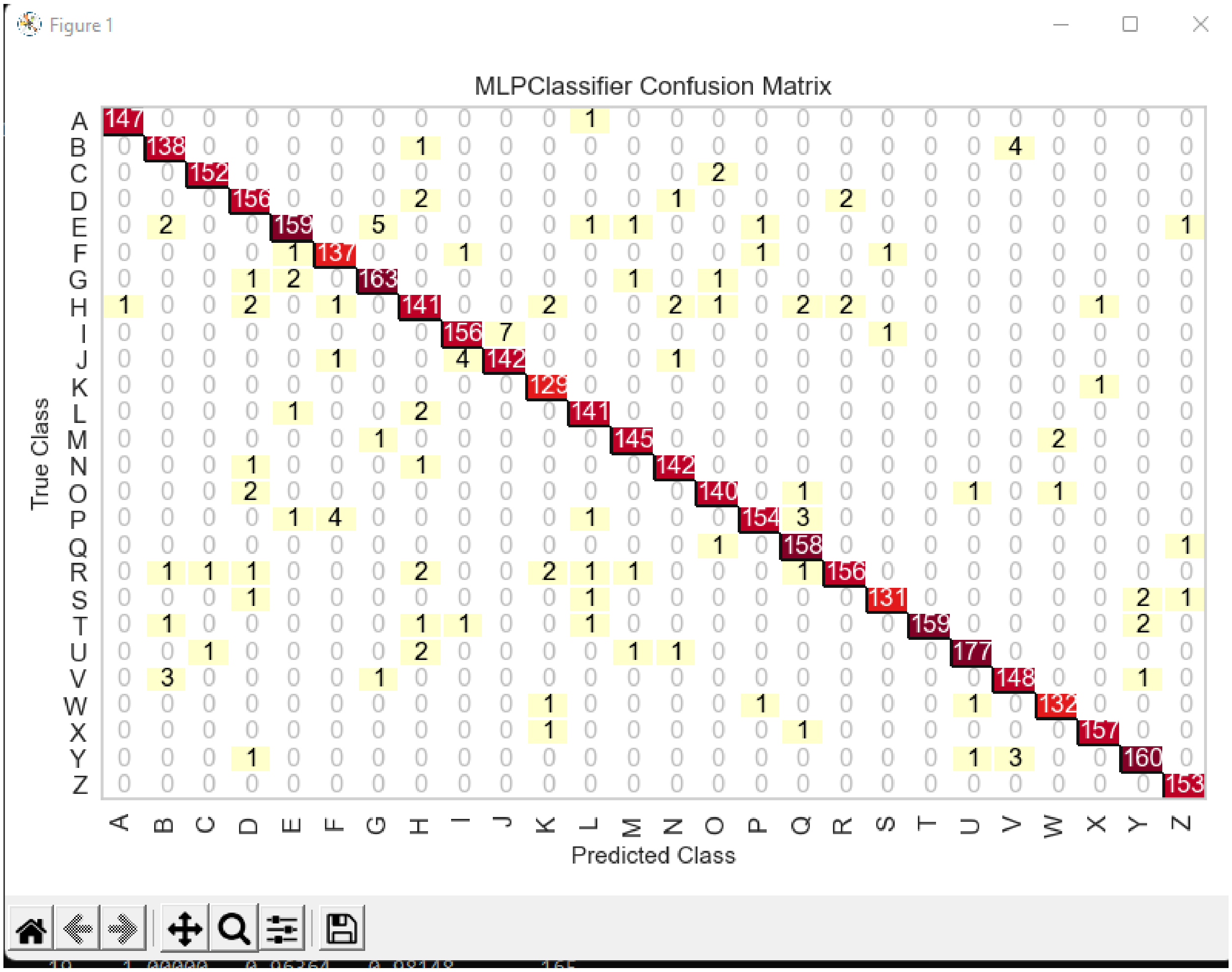


Figure 1
MLPClassifier Confusion Matrix
True Class
Predicted Class
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 0 0 0 0 0 0 153]]

	precision	recall	f1-score	support
0	0.99324	0.99324	0.99324	148
1	0.95172	0.96503	0.95833	143
2	0.98701	0.98701	0.98701	154
3	0.94545	0.96894	0.95706	161
4	0.96951	0.93529	0.95210	170
5	0.95804	0.97163	0.96479	141
6	0.95882	0.97024	0.96450	168
7	0.92763	0.90968	0.91857	155
8	0.96296	0.95122	0.95706	164
9	0.95302	0.95946	0.95623	148
10	0.95556	0.99231	0.97358	130
11	0.95918	0.97917	0.96907	144
12	0.97315	0.97973	0.97643	148
13	0.96599	0.98611	0.97595	144
14	0.96552	0.96552	0.96552	145
15	0.98089	0.94479	0.96250	163
16	0.95181	0.98750	0.96933	160
17	0.97500	0.93976	0.95706	166
18	0.98496	0.96324	0.97398	136
19	1.00000	0.96364	0.98148	165
20	0.98333	0.97253	0.97790	182
21	0.95484	0.96732	0.96104	153
22	0.97778	0.97778	0.97778	135
23	0.98742	0.98742	0.98742	159
24	0.96970	0.96970	0.96970	165
25	0.98077	1.00000	0.99029	153
accuracy			0.96825	4000
macro avg	0.96820	0.96878	0.96838	4000
weighted avg	0.96841	0.96825	0.96821	4000

KNN

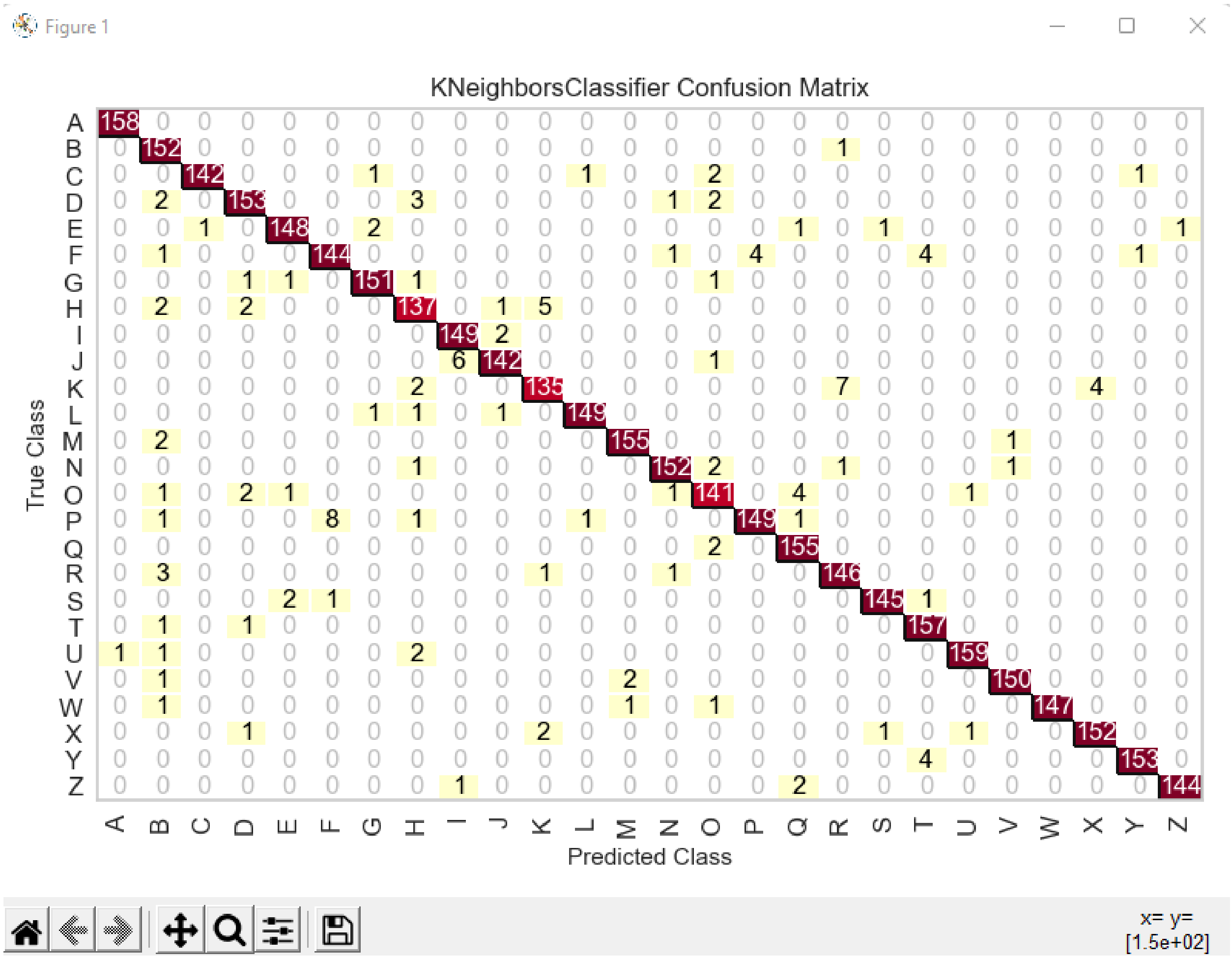


Figure 1
KNeighborsClassifier Confusion Matrix
True Class
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Predicted Class
x= y=
[1.5e+02]

	precision	recall	f1-score	support
A	0.99371	1.00000	0.99685	158
B	0.90476	0.99346	0.94704	153
C	0.99301	0.96599	0.97931	147
D	0.95625	0.95031	0.95327	161
E	0.97368	0.96104	0.96732	154
F	0.94118	0.92903	0.93506	155
G	0.97419	0.97419	0.97419	155
H	0.92568	0.93197	0.92881	147
I	0.95513	0.98675	0.97068	151
J	0.97260	0.95302	0.96271	149
K	0.94406	0.91216	0.92784	148
L	0.98675	0.98026	0.98350	152
M	0.98101	0.98101	0.98101	158
N	0.97436	0.96815	0.97125	157
O	0.92763	0.93377	0.93069	151
P	0.97386	0.92547	0.94904	161
Q	0.95092	0.98726	0.96875	157
R	0.94194	0.96689	0.95425	151
S	0.98639	0.97315	0.97973	149
T	0.94578	0.98742	0.96615	159
U	0.98758	0.97546	0.98148	163
V	0.98684	0.98039	0.98361	153
W	1.00000	0.98000	0.98990	150
X	0.97436	0.96815	0.97125	157
Y	0.98710	0.97452	0.98077	157
Z	0.99310	0.97959	0.98630	147
accuracy			0.96625	4000
macro avg	0.96661	0.96613	0.96618	4000
weighted avg	0.96668	0.96625	0.96628	4000

Conclusion :

We are able to produce different types of confusion matrices for different types of classifications and their accuracies by training the model using training sets and we are able to predict and plot the responses for the test dataset. The model can also calculate the macro average and weighted average of the obtained data. From the obtained results the accuracy of SVM model is higher than the other models

Contribution :

C.Sravan chaitanya	- GaussianNB , Decision Tree and Report
K.Pradyumna Reddy	- SVM , MLP and PPT
B.Manish kumar	- SVM , KNN , Theory and References

References:

https://link.springer.com/article/10.1007/BF00114162

Code link / Outputs

https://drive.google.com/drive/folders/1gxoVpKSeozuWMBep0h-qREKqTyBK1nB8?usp=sharing