A Comparison of Machine Learning Classifiers for Sign Language Recognition

Author: Ahsan Shehzad

Abstract
In recent years, enormous research is progressing in the field of Computer Vision and Human Computer Interaction where hand gestures play a vital role. Hand gestures are more powerful means of communication for hearing impaired when they communicate to the normal people everywhere in day to day life. As the normal people find little difficulty in recognizing and interpreting the meaning of sign language expressed by the hearing impaired, it is inevitable to have an interpreter for translation of sign language. To overcome this difficulty, an automatic hand gesture recognition system which translates the sign language into text needs to be developed.

In this paper, we focus on the core part of sign language recognition system in which an efficient and effective machine learning model is required. We used a public ally available Sign Language MNIST dataset and apply state of the art classifiers on it and provide a performance comparison of them. KNN, SVM and neural networks provide good results, but deep learning methods out perform among all the classifiers.

Keywords:
Sign Language Recognition, machine learning

Introduction
The usage of computers is found in all the fields nowadays. However, the interaction between man and the machine is carried out only through the conventional input devices such as keyboard, mouse etc., Hence, the computers require some other ways for more convenient and efficient interaction with the human being by using speech or body language which includes gestures and postures. Gestures may be of two types namely static gestures [1] and dynamic gestures [2]. These gestures are shown by finger spelling. Several researchers are working in the area of Human Computer Interaction [3], since the main objective of HCI is to create a simple, visual interface for providing a natural way of communication between human and computer. The visual interface is created using the hand gestures and head movements. Hand gesture recognition is used in many typical applications such as computer game's control, virtual mouse, turning on/off the domestic appliances, and gesture based navigation of medical images during surgery. For hearing impaired, sign language is the only way of communication which employs different signs made by moving the hands in combination with facial expressions and postures of the body. Hand gesture Recognition is one of the areas which has high applicability and helps the community of hearing impaired using image processing techniques and computer vision. In this proposed technique, the signer need not wear any specialized hardware equipment or color gloves. In this paper, the American Sign Language alphabets are recognized using static hand gestures by vision-based approach.

The paper is organized as follows. Section 2 explains the various work done so far in this area. Section 3 describes the proposed system with its architecture. Section 4 comprises the experimental results and discussion. Section 5 concludes with future work.

Related work

Many research works have been done during the past two decades in the area of vision based hand gesture recognition. The overall objective of this subject is to help the hearing impaired people to communicate with the normal people, and replace the conventional language with sign language. Another application of gesture language is human-computer interaction, which uses hand gestures as input data to a computer through webcam. In HCI, a visual interface is created to provide a natural way of communication between man and machine [4].

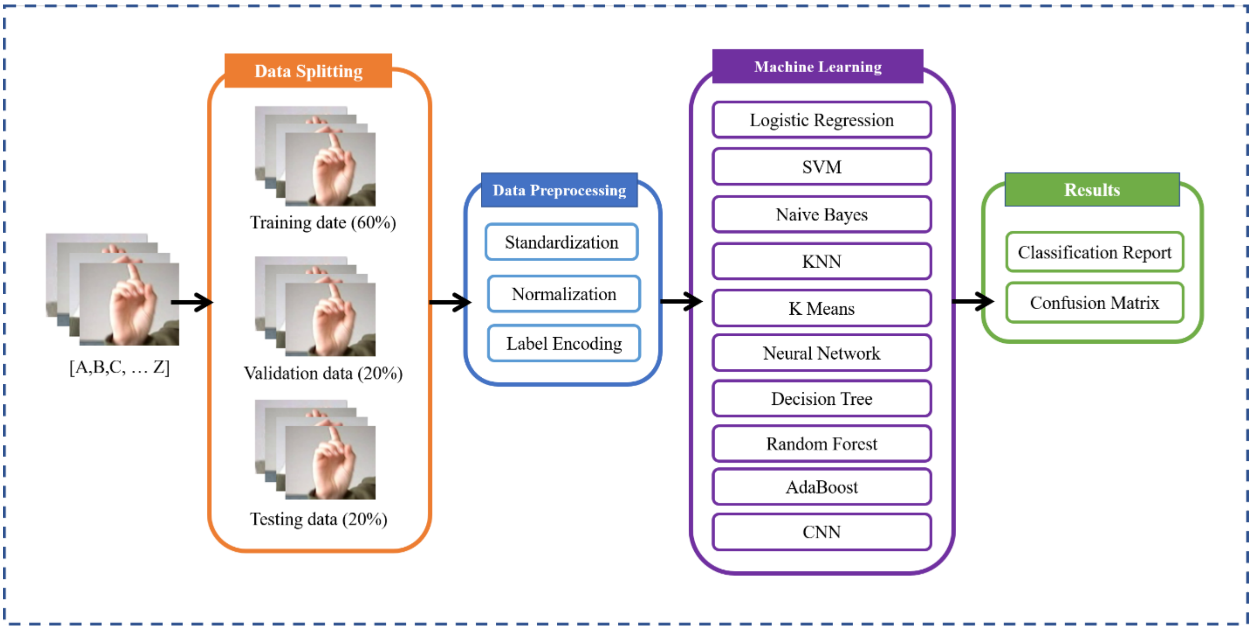
Nasser H.Dardas et al. proposed a novel and real time system [5] for interacting with a video game via hand gestures using SIFT and SURF feature extraction. The hand gesture recognition system involves both uniform and complex background. Many works incorporate skin colour segmentation for segmenting hand from a complex background. Jayashree R. Pansare et al. proposed a system to recognize 26 static hand gestures for ASL alphabets from a complex background [6] using the Euclidean distance measure. [7] Presented a hand gesture recognition system for American Sign Language using back propagation neural network. The neural network is used to classify the hand gestures for alphabets in [8] after segmenting the hand from the input image using the Lab color space and the extraction of peaks and valleys as the features. The hand segmentation technique using background subtraction is given in [9] to segment the hand from the image with uniform and complex background. A robust system [10] invariant to scaling and translation is implemented to recognize the gesture signs using ANN.A novel method of pattern recognition is presented in [11] for recognizing 36 different gestures using SIFT features with PCA and template matching methods.

Even though many techniques have been developed in this field, very few works have been done in recognition of Indian Sign Language. [12] Proposed a system for recognizing ISL gestures using Eigen value weighted Euclidean distance as a classification technique. Gabor filters are convolved with images to acquire the features and the feature space is reduced using PCA [13]. Using the reduced Gabor features, gestures are recognized by SVM. Discrete Wavelet Transform features are used to train the neural network in recognition of Persian Sign Language in [14].

Various color models are used in the segmentation of hand in the gesture recognition system. The perceptual color space is used for hand gesture recognition [15] for wireless robot control applications. The finger spelling is dependent on the specific language of a country. Finger spelling [16] is used in Indian sign language, American Sign Language, British Sign Language, Chinese sign Language, Persian Sign Language, Arabic sign language, Malaysian sign language for recognition of sign language.

Methods

Machine learning consist a lot of supervised and unsupervised estimators. Supervised methods are used to solve the sign language recognition problem. The dataset has been trained and tested on state of the art classification predictors. The dataset is divided into training, validation and testing examples. Class balance is also determined in advanced to evaluate bias and variance. Dataset is also standardized and normalized to get better results. Each machine learning model is applied one by one and detailed classification report with confusion matrix is computed. The overall flow of the process is illustrated in figure /ref{}.



The Dataset Used

The open dataset given at Kaggle called Sign Language MNIST - Drop-In Replacement for MNIST for Hand Gesture Recognition Task [1] which contains set of 28x28 images of all the alphabet, except J and Z, of the standard American Sign Language(ASL). The data contains a total of 27,455 cases. A sampled image set can be observed in Fig. (1). The data in its raw form is provided as a pixel to pixel intensity [0-255] class-wise distributed XLS files. The data preprocessing steps included conversion of the mentioned data to image format using Python's open-source libraries Pandas, Numpy and others to obtain PNG format 28x28 grayscale images.

Data Splitting

Dataset is split into training data (60%), validation data (20%) and testing data (20%). Training data is used for training machine learning models. Validation data is used to validate all models. Testing data is used for testing and confirming accuracy of our classifiers. The sklearn ***sklearn.model_selection.train_test_split*** tool is used for dataset splitting.

Data preprocessing

Dataset is preprocessed to get rid of missing data and standardization. Data is also normalized to get smoother and efficient training of machine learning models.

Standardization

Standardize features by removing the mean and scaling to unit variance. The standard score of a sample x is calculated as:

$$z = \frac{x - u}{s}$$

where u is the mean of the training samples, and s is the standard deviation of the training samples.

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using the transform method. Standardization of a dataset is a common requirement for many machine learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data.

Normalization

All samples of dataset IS normalized to unit norm. Each sample (i.e. each row of the data matrix) with at least one non-zero component is rescaled independently of other samples so that its norm (l1 or l2) equals one. Scaling inputs to unit norms is a common operation for text classification or clustering for instance.

Label Encoding

Labels of the dataset are encoded between 0 and n_classes-1. sklearn.preprocessing.LabelEncoder is used for encoding labels.

Label	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
Code	0	1	2	3	4	5	6	7	8	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Machine learning

The core part of sign language recognition system is to build an efficient and accurate recognition module. Machine learning is the state of the art technique to build any recognition system. We adopted of the shelf top classification models and compare their accuracy for the sign language recognition dataset.

Logistic Regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Support vector machines (SVM)

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N—the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Naive bays

A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task. The crux of the classifier is based on the Bayes theorem. Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other.

KNN

the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. Both for classification and regression, a useful technique can be used to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

K-means

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data. The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.

Neural network

Artificial neural networks are inspired by the human neural network architecture. The simplest neural network consists of only one neuron and is called a perceptron. A perceptron has one input layer and one neuron. The number of nodes in the input layer is equal to the number of features in the input dataset. Artificial neural networks, are a combination of multiple neurons connected in the form a network. An artificial neural network has an input layer, one or more hidden layers, and an output layer.

Decision tree

The decision tree classifiers organized a series of test questions and conditions in a tree structure. In the decision tree, the root and internal nodes contain attribute test conditions to separate records that have different characteristics. All the terminal node is assigned a class label Yes or No. Once the decision tree has been constructed, classifying a test record is straightforward. Starting from the root node, we apply the test condition to the record and follow the appropriate branch based on the outcome of the test. It then leads us either to another internal node, for which a new test condition is applied, or to a leaf node. When we reach the leaf node, the class label associated with the leaf node is then assigned to the record.

Random Forest

Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.

AdaBoost

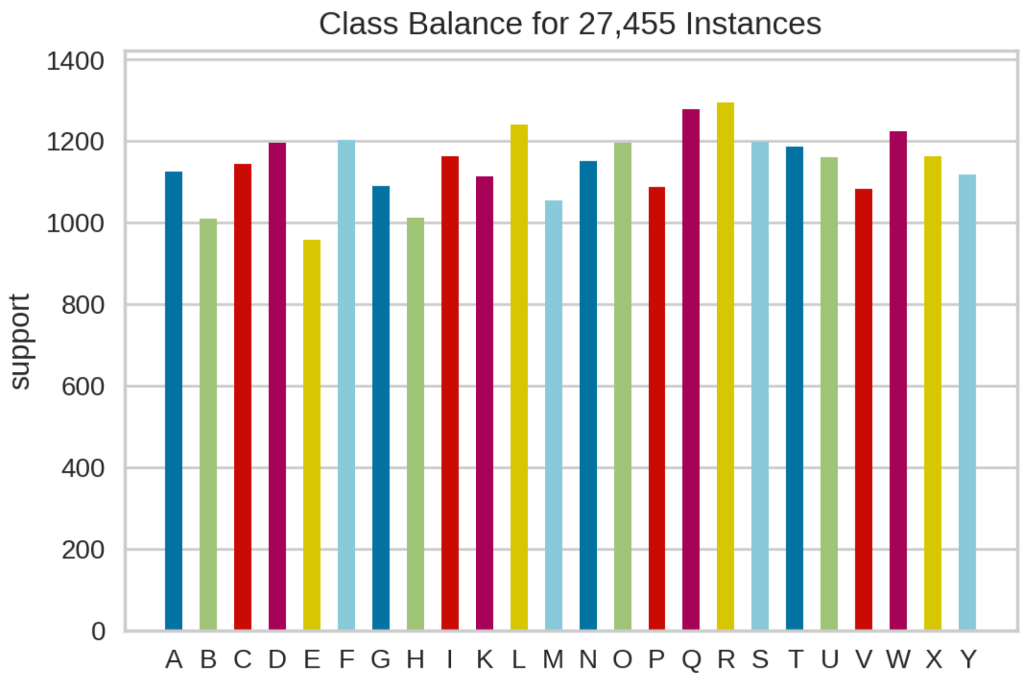
Ada-boost classifier combines weak classifier algorithm to form strong classifier. A single algorithm may classify the objects poorly. But if we combine multiple classifiers with selection of training set at every iteration and assigning right amount of weight in final voting, we can have good accuracy score for overall classifier.

CNN

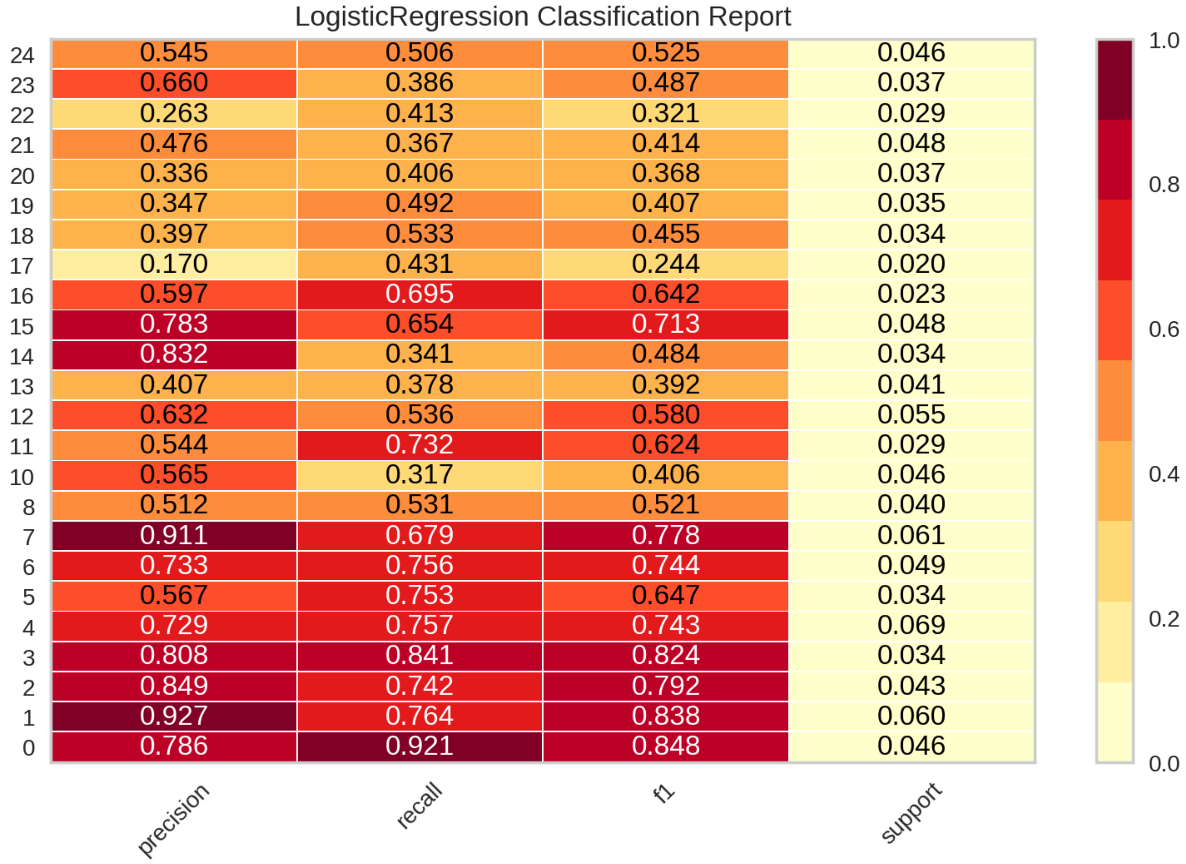
Convolutional neural networks (also refered to as CNN or ConvNet) are a class of deep neural networks that have seen widespread adoption in several computer vision and visual imagery applications. Convolutional Neural Networks consist of multiple layers designed to require relatively little pre-processing compared to other image classification algorithms. They learn by using filters and applying them to the images. The algorithm takes a small square (or 'window') and starts applying it over the image. Each filter allows the CNN to identify certain patterns in the image. The CNN looks for parts of the image where a filter matches the contents of the image.

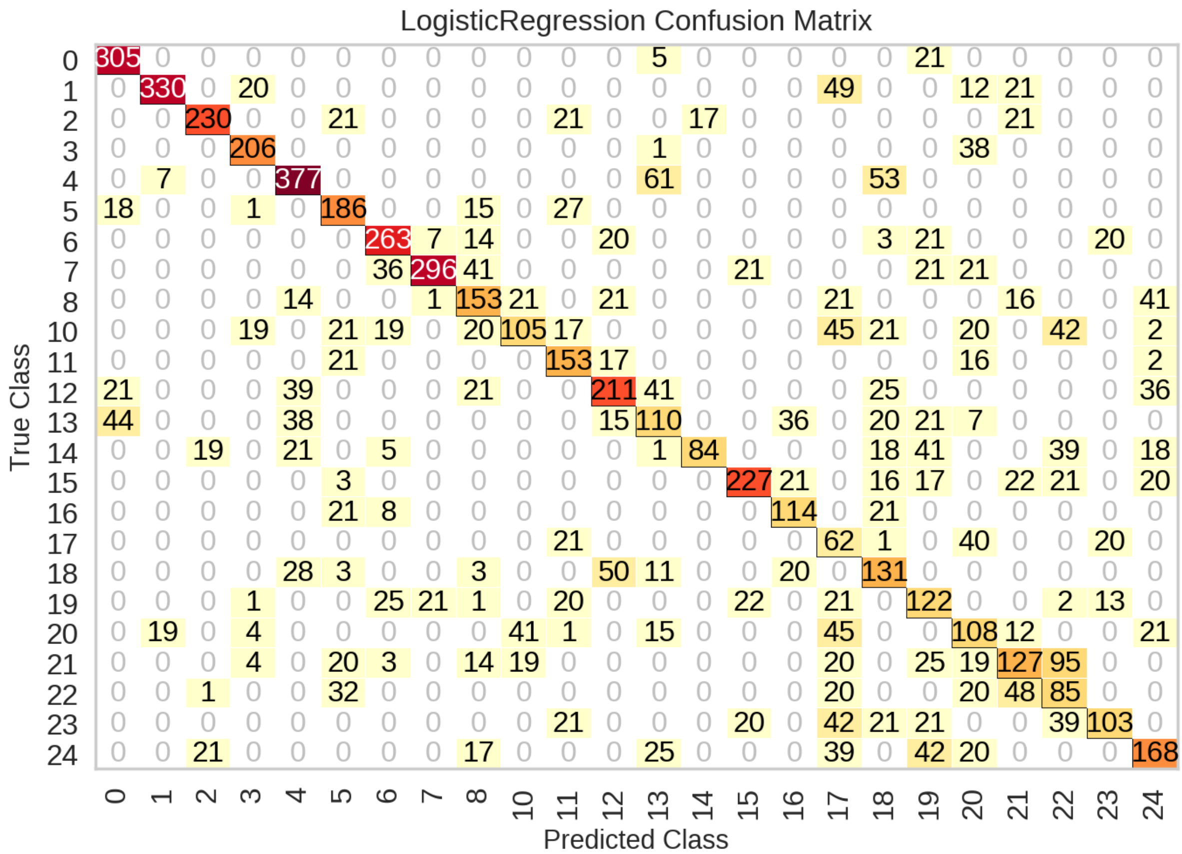
Results

We applied state of the art machine learning classifiers on sign language MNIST dataset. The detailed classification report with the support of confusion matrix for each classifier is computed. The class balance of dataset is illustrated in figure \ref{}.

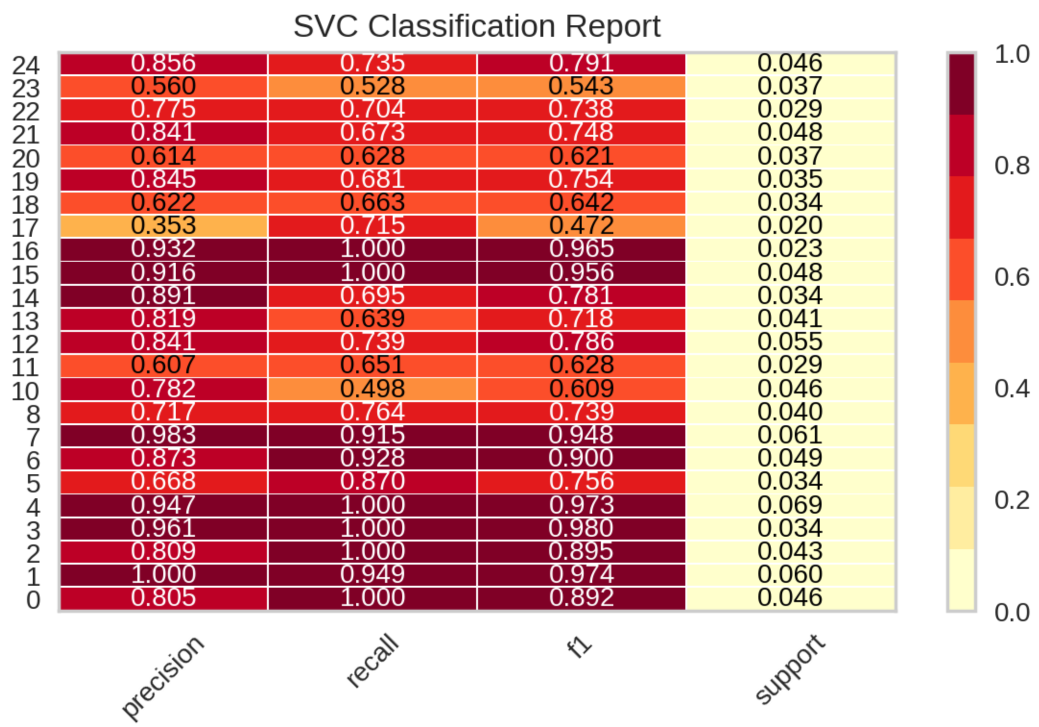


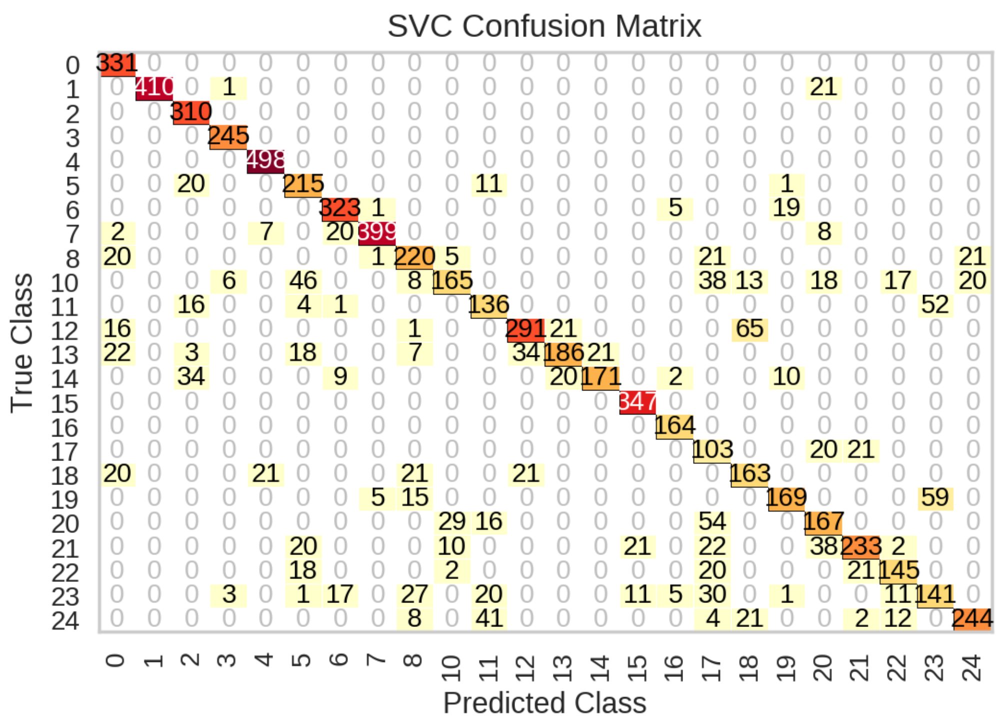
Logistic Regression



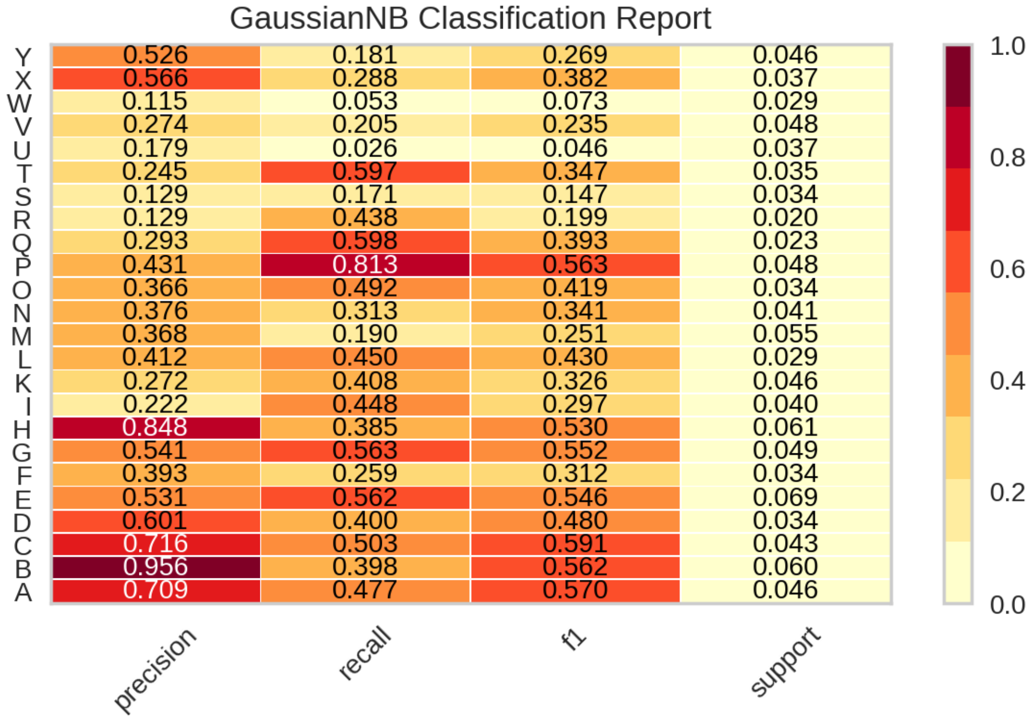


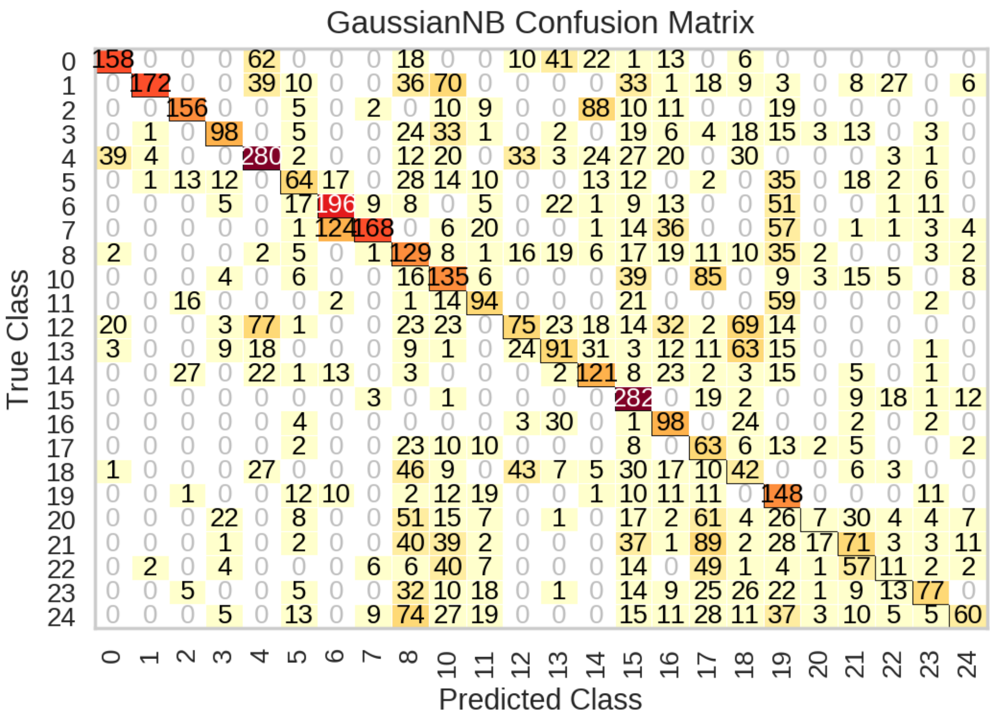
Support vector machines (SVM)



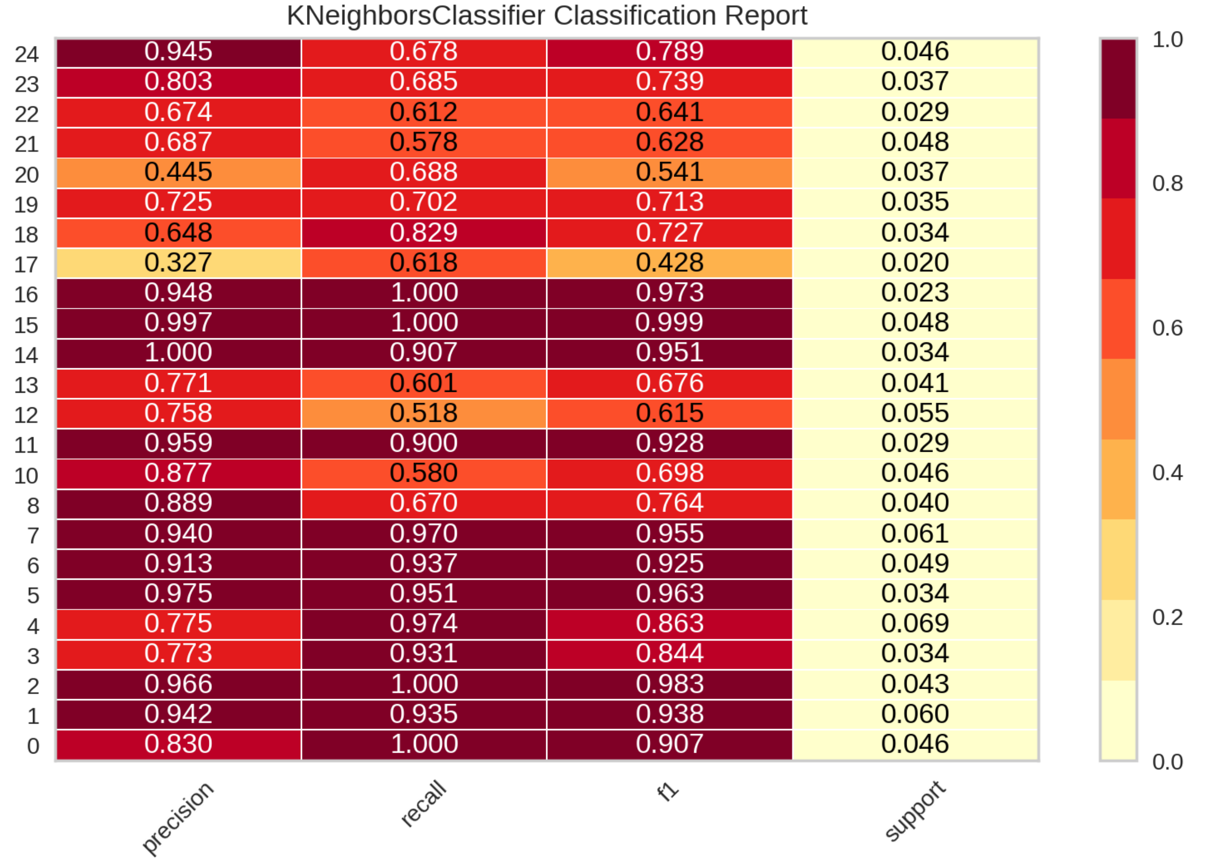


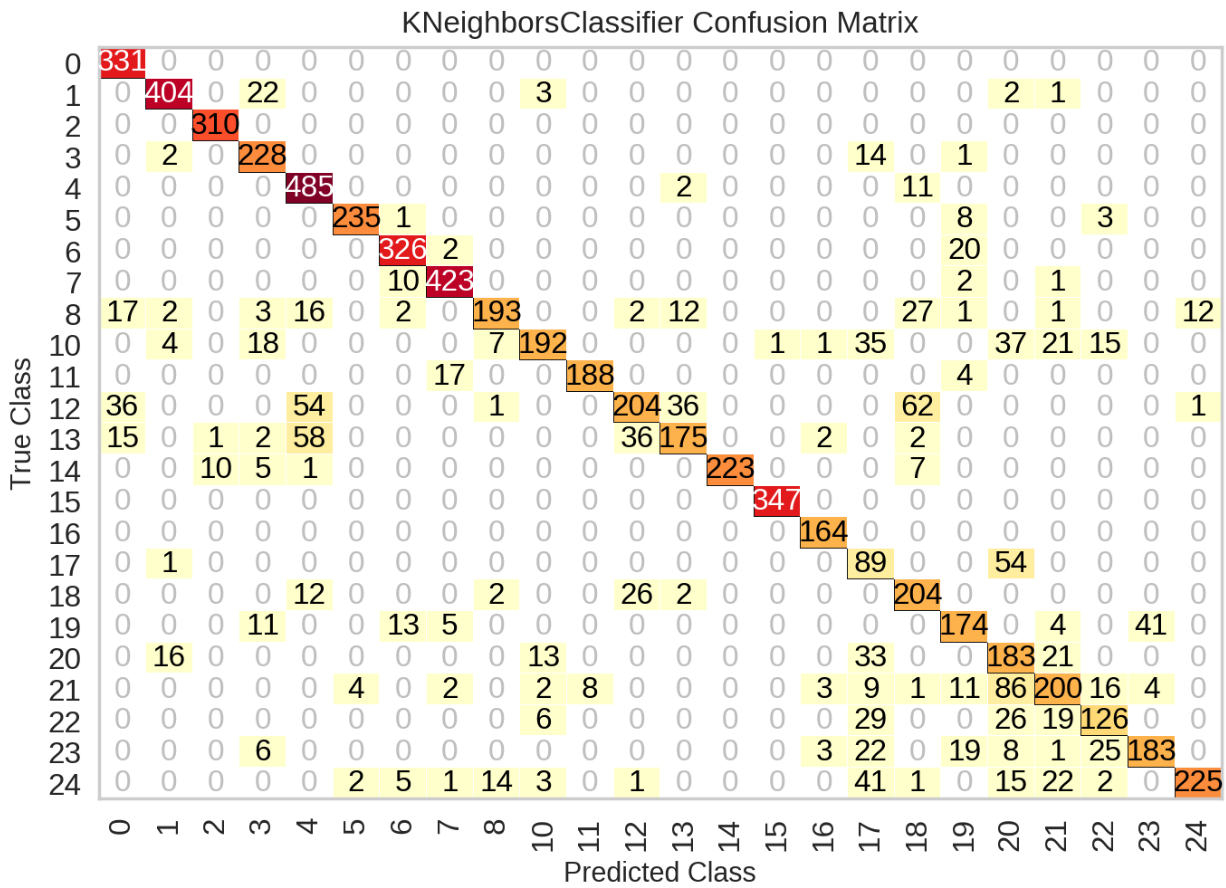
Naive bays


GaussianNB Classification Report
Y 0.526 0.181 0.269 0.046
X 0.566 0.288 0.382 0.037
W 0.115 0.053 0.073 0.029
V 0.274 0.205 0.235 0.048
U 0.179 0.026 0.046 0.037
T 0.245 0.597 0.347 0.035
S 0.129 0.171 0.147 0.034
R 0.129 0.438 0.199 0.020
Q 0.293 0.598 0.393 0.023
P 0.431 0.813 0.563 0.048
O 0.366 0.492 0.419 0.034
N 0.376 0.313 0.341 0.041
M 0.368 0.190 0.251 0.055
L 0.412 0.450 0.430 0.029
K 0.272 0.408 0.326 0.046
I 0.222 0.448 0.297 0.040
H 0.848 0.385 0.530 0.061
G 0.541 0.563 0.552 0.049
F 0.393 0.259 0.312 0.034
E 0.531 0.562 0.546 0.069
D 0.601 0.400 0.480 0.034
C 0.716 0.503 0.591 0.043
B 0.956 0.398 0.562 0.060
A 0.709 0.477 0.570 0.046
precision
recall
f1
support
1.0
0.8
0.6
0.4
0.2
0.0

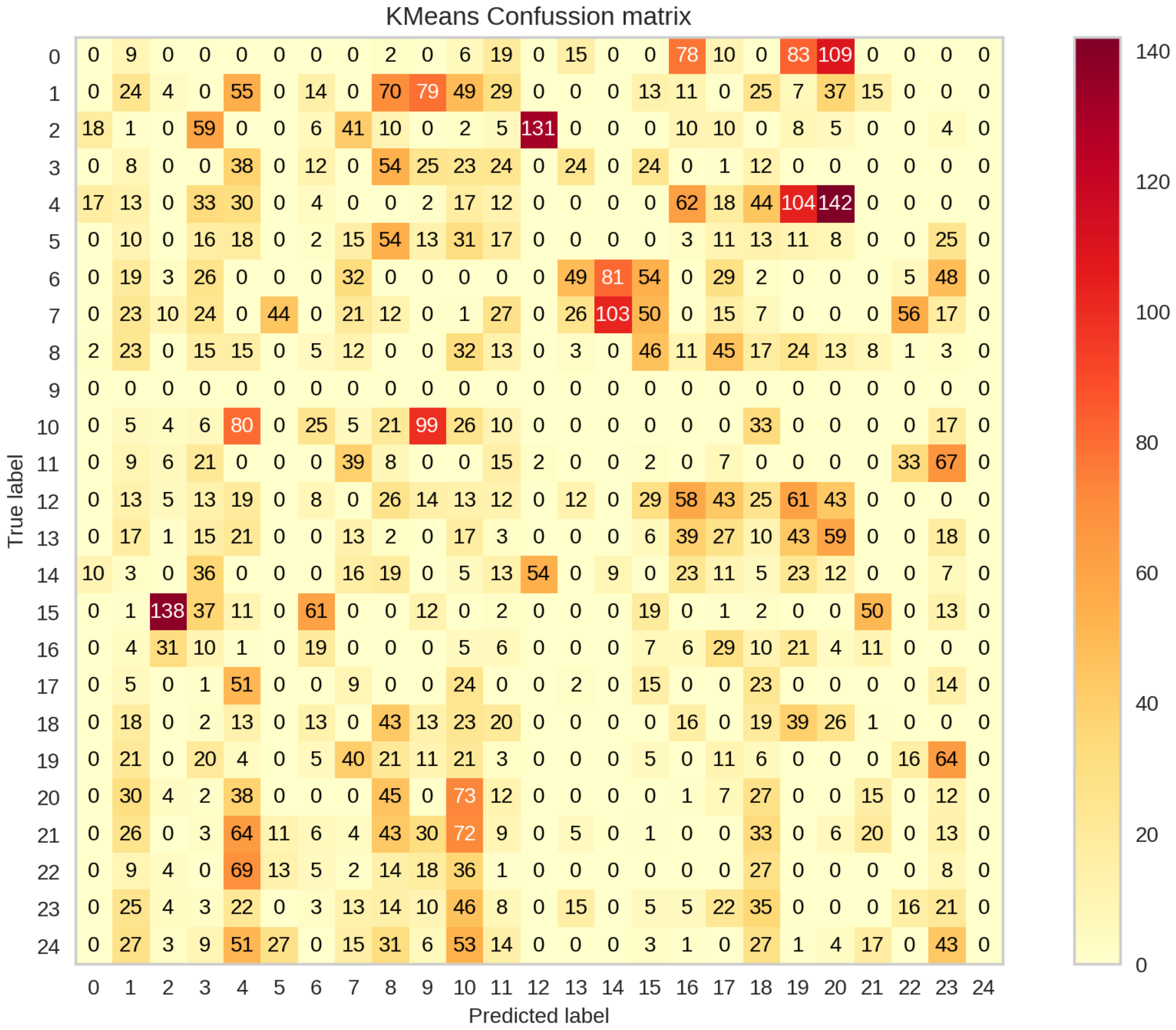

GaussianNB Confusion Matrix
True Class
Predicted Class

KNN

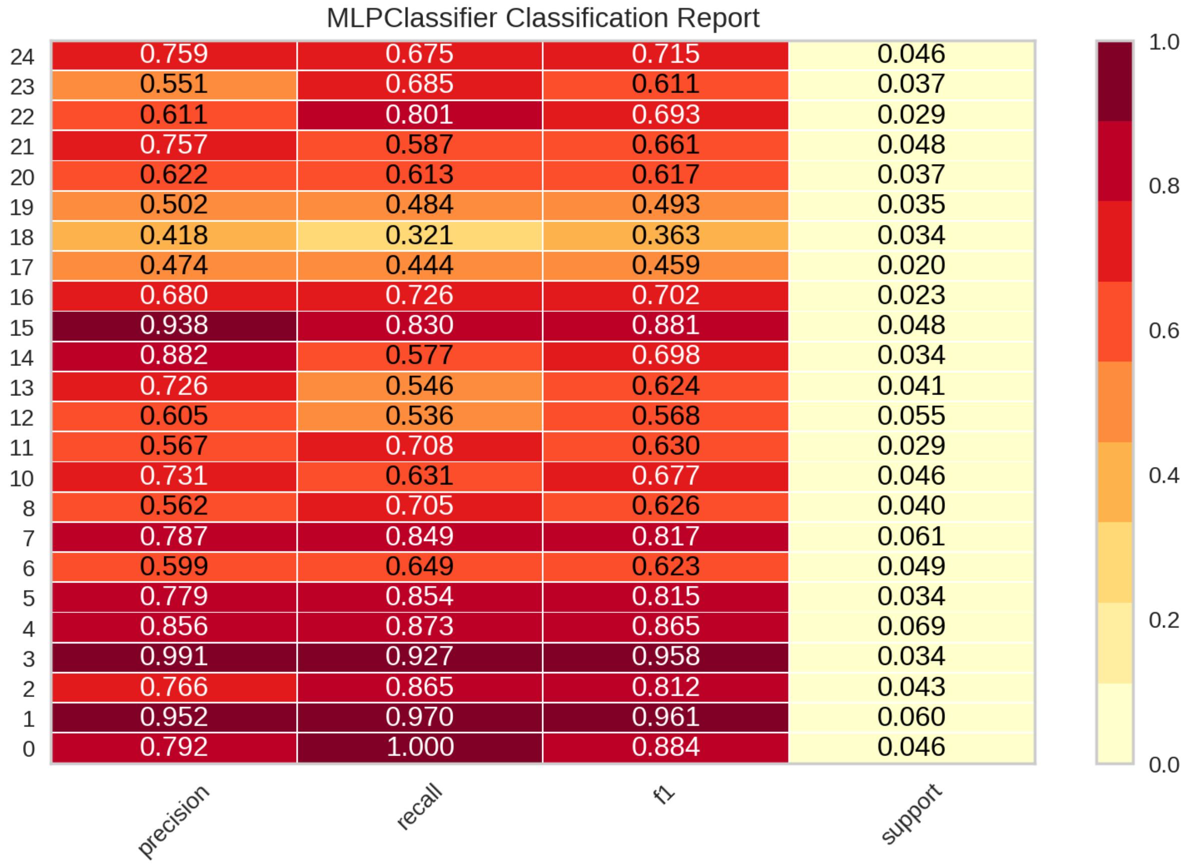

KNeighborsClassifier Classification Report
precision
recall
f1
support

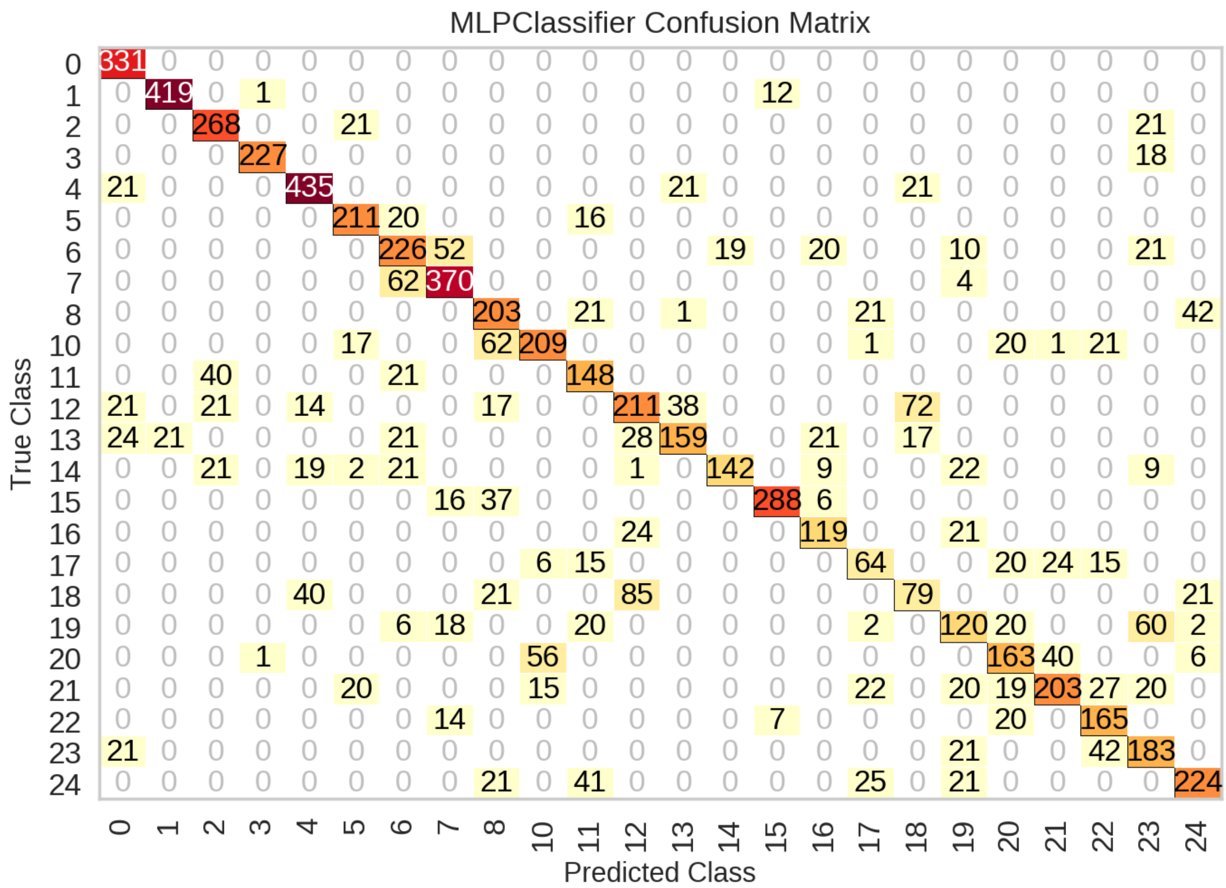

KNeighborsClassifier Confusion Matrix
True Class
Predicted Class

K-means

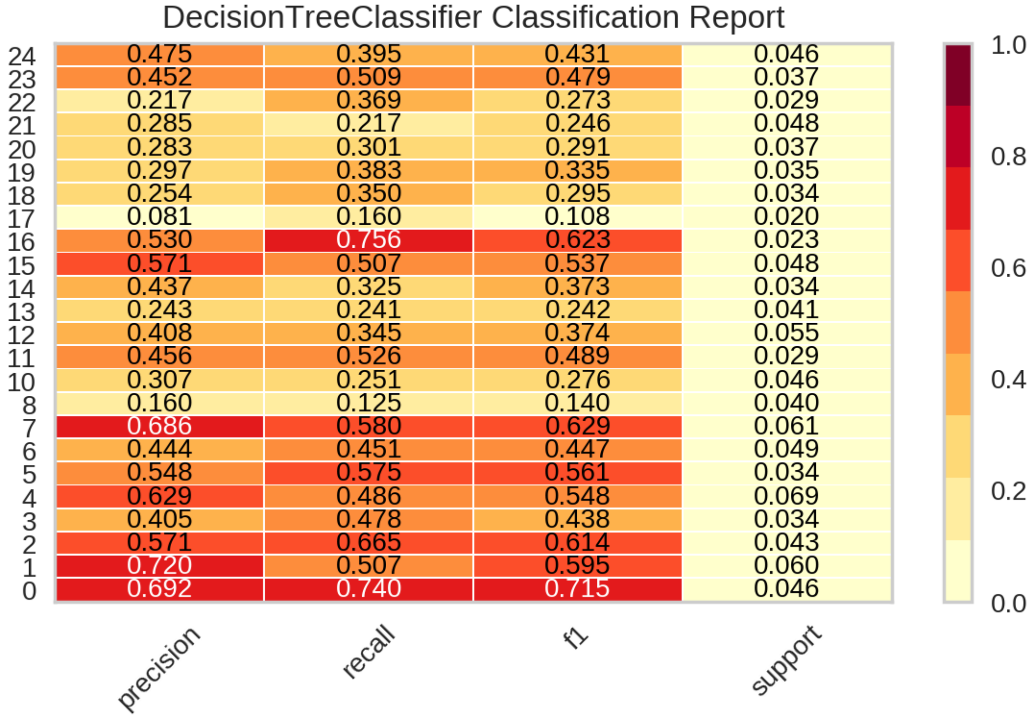

KMeans Confussion matrix
True label
Predicted label
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
0 9 0 0 0 0 0 0 2 0 6 19 0 15 0 0 78 10 0 83 109 0 0 0 0
0 24 4 0 55 0 14 0 70 79 49 29 0 0 0 13 11 0 25 7 37 15 0 0 0
18 1 0 59 0 0 6 41 10 0 2 5 131 0 0 0 10 10 0 8 5 0 0 4 0
0 8 0 0 38 0 12 0 54 25 23 24 0 24 0 24 0 1 12 0 0 0 0 0 0
17 13 0 33 30 0 4 0 0 2 17 12 0 0 0 0 62 18 44 104 142 0 0 0 0
0 10 0 16 18 0 2 15 54 13 31 17 0 0 0 0 3 11 13 11 8 0 0 25 0
0 19 3 26 0 0 0 32 0 0 0 0 0 49 81 54 0 29 2 0 0 0 5 48 0
0 23 10 24 0 44 0 21 12 0 1 27 0 26 103 50 0 15 7 0 0 0 56 17 0
2 23 0 15 15 0 5 12 0 0 32 13 0 3 0 46 11 45 17 24 13 8 1 3 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 5 4 6 80 0 25 5 21 99 26 10 0 0 0 0 0 0 33 0 0 0 0 17 0
0 9 6 21 0 0 0 39 8 0 0 15 2 0 0 2 0 7 0 0 0 0 33 67 0
0 13 5 13 19 0 8 0 26 14 13 12 0 12 0 29 58 43 25 61 43 0 0 0 0
0 17 1 15 21 0 0 13 2 0 17 3 0 0 0 6 39 27 10 43 59 0 0 18 0
10 3 0 36 0 0 0 16 19 0 5 13 54 0 9 0 23 11 5 23 12 0 0 7 0
0 1 138 37 11 0 61 0 0 12 0 2 0 0 0 19 0 1 2 0 0 50 0 13 0
0 4 31 10 1 0 19 0 0 0 5 6 0 0 0 7 6 29 10 21 4 11 0 0 0
0 5 0 1 51 0 0 9 0 0 24 0 0 2 0 15 0 0 23 0 0 0 0 14 0
0 18 0 2 13 0 13 0 43 13 23 20 0 0 0 0 16 0 19 39 26 1 0 0 0
0 21 0 20 4 0 5 40 21 11 21 3 0 0 0 5 0 11 6 0 0 0 16 64 0
0 30 4 2 38 0 0 0 45 0 73 12 0 0 0 0 1 7 27 0 0 15 0 12 0
0 26 0 3 64 11 6 4 43 30 72 9 0 5 0 1 0 0 33 0 6 20 0 13 0
0 9 4 0 69 13 5 2 14 18 36 1 0 0 0 0 0 0 27 0 0 0 0 8 0
0 25 4 3 22 0 3 13 14 10 46 8 0 15 0 5 5 22 35 0 0 0 16 21 0
0 27 3 9 51 27 0 15 31 6 53 14 0 0 0 3 1 0 27 1 4 17 0 43 0
140
120
100
80
60
40
20
0

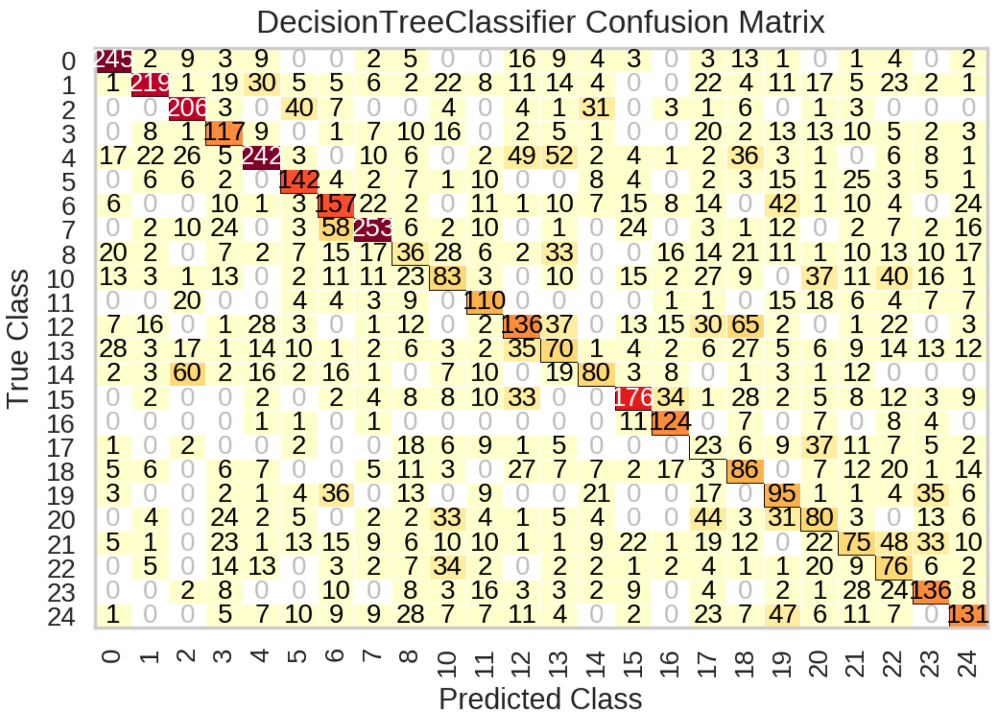
Neural network



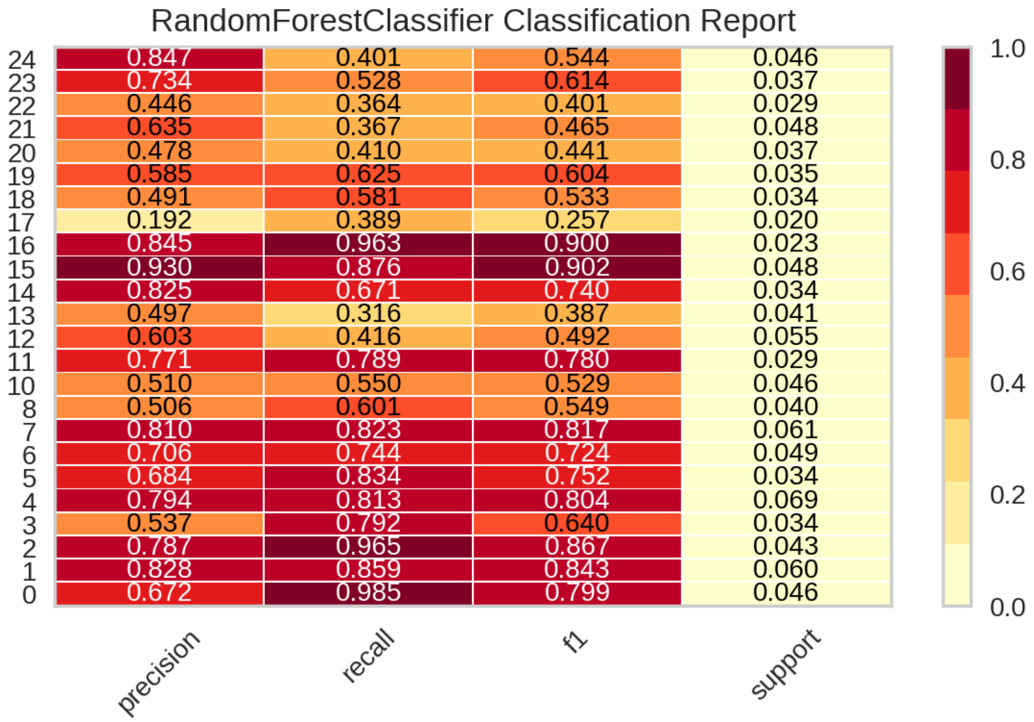


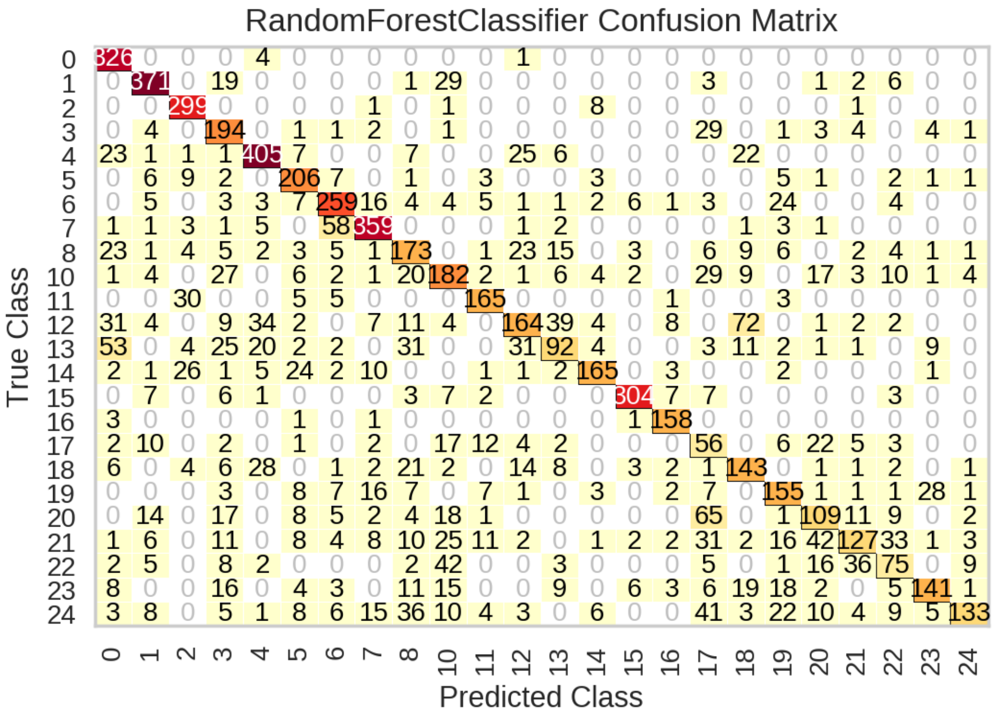
Decision tree



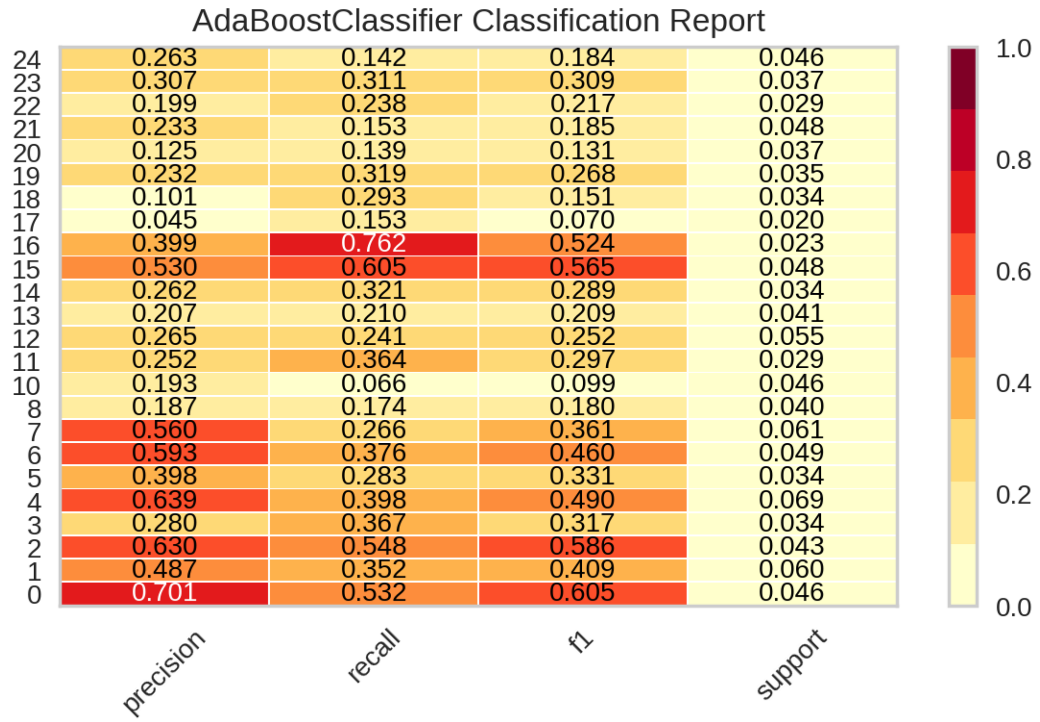


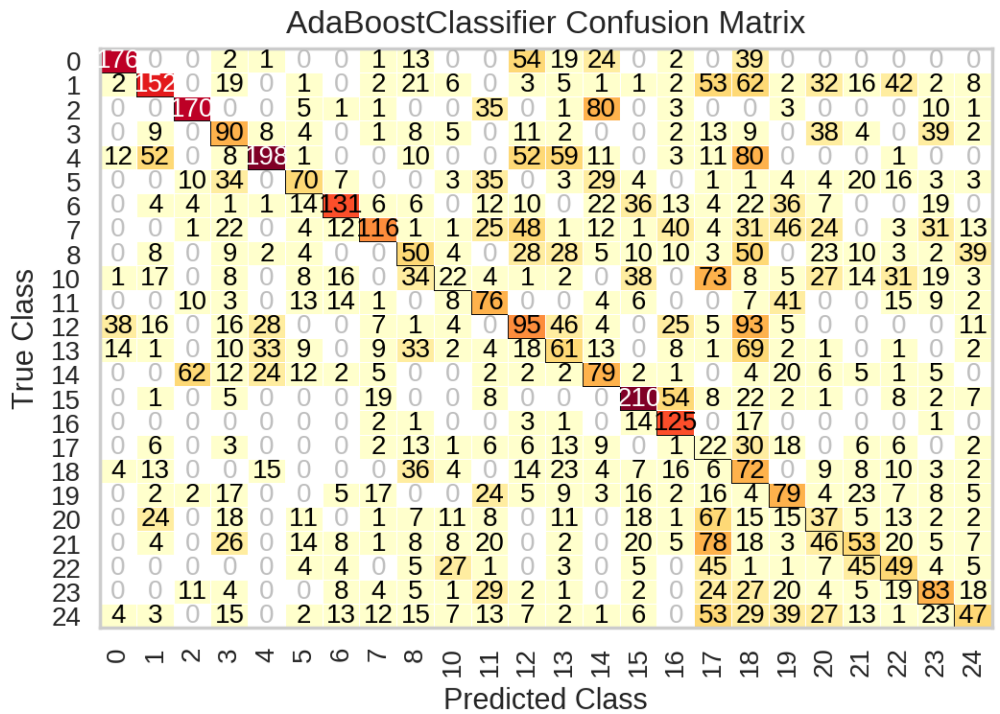
Random Forest





AdaBoost


AdaBoostClassifier Classification Report
precision
recall
f1
support


AdaBoostClassifier Confusion Matrix
True Class
Predicted Class

CNN

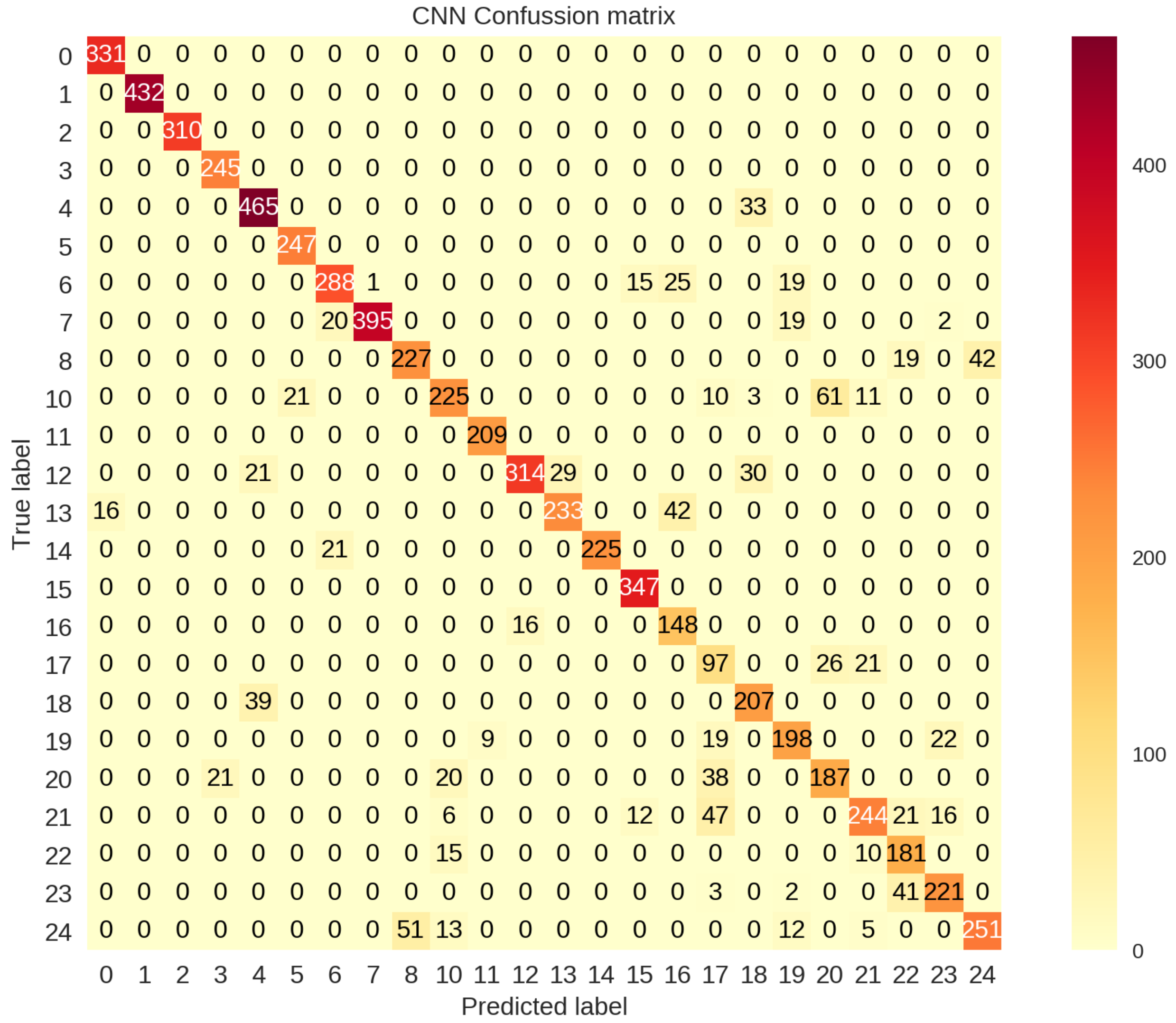
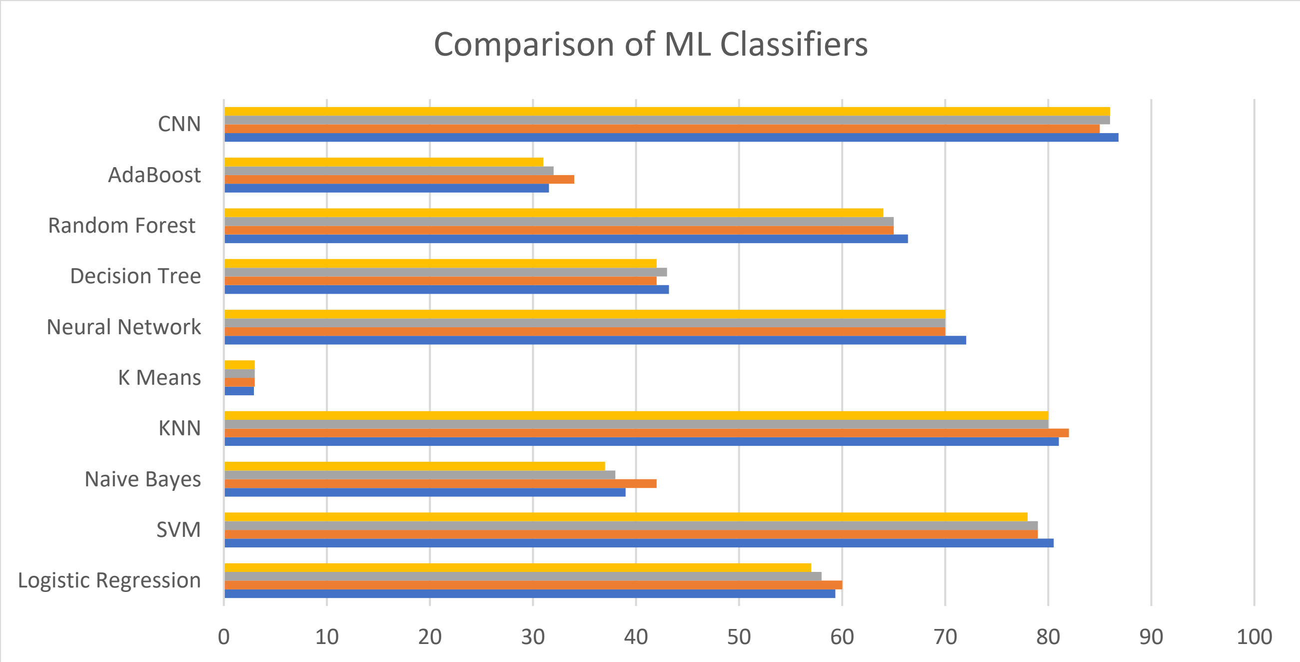


Table 1 Comparison of ML Classifiers

Classifier	**Accuracy**	**Precision**	**Recall**	**F1 - Score**
Logistic Regression	59.34	0.60	0.58	0.57
SVM	80.53	0.79	0.79	0.78
Naive Bayes	38.98	0.42	0.38	0.37
KNN	81.03	0.82	0.80	0.80
K Means	02.92	0.03	0.03	0.03
Neural Network	72.04	0.70	0.70	0.70
Decision Tree	43.19	0.42	0.43	0.42
Random Forest	66.38	0.65	0.65	0.64
AdaBoost	31.55	0.34	0.32	0.31
CNN	86.82	0.85	0.86	0.86



	Logistic Regression	SVM	Naive Bayes	KNN	K Means	Neural Network	Decision Tree	Random Forest	AdaBoost	CNN
F1 - Score	57	78	37	80	3	70	42	64	31	86
Recall	58	79	38	80	3	70	43	65	32	86
Precision	60	79	42	82	3	70	42	65	34	85
Accuracy	59.34	80.53	38.98	81.03	2.92	72.04	43.19	66.38	31.55	86.82

References

[1] Singha.J, and Das.K, "Hand Gesture Recognition based on Karhunen-Loeve Transform", Mobile and Embedded Technology International Conference (MECON), January 17-18, 2013, pp 365-371.

[2] Joyeeta Singha, and Karen Das, "Recognition of Indian Sign Language in Live Video", International Journal of Computer Applications, Vol.70, No.19, May 2013.

[3] T.Kapsciinski and M.Wysocki, "Hand Gesture Recognition for Man-Machine interaction", Second Workshop on Robot Motion and Control, October 18-20, 2001, pp.91-96.

[4] I.G.Incertis, J.G.G.Bermejo, and E.Z.Casanova, "Hand Gesture Recognition for Deaf People Interfacing", The 18th International Conference on Pattern Recognition (ICPR), 2006.

[5] Nasser H.Dardas, Nicolas D.Georganas, "Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques", IEEE Transactions on Instrumentation and Measurement, Vol.60, No.11, November 2011.

[6] Jayashree R.Pansare, Sharavan H.Gawande, Maya Ingle, "Real-Time Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background", Journal of Signal and Information Processing, 2012, Vol.3, pp 364-367.

[7] Md.AtiqurRahman, Ahsan-Ul-Ambia,Md.Aktaruzzaman, "Recognition of Static Hand Gestures of Alphabet in ASL", IJCIT, Vol.2, Issue 1, 2011.

[8] Rajesh Mapari, Dr.Govind Kharat, "Hand Gesture Recognition using Neural Network", International Journal of Computer Science and Network, Vol.1, Issue 6, December 2012.

[9] A.A.Randive, H.B.Mali, S.D.Lokhande, "Hand Gesture Segmentation", International Journal of Computer Technology and Electronics Engineering, Vol.2, Issue 3, June2012.

[10] Vaishali S.Kulkarni, Dr.S.D.Lokhande, "Appearance Based Recognition of American Sign Language Using Gesture Segmentation", International Journal of Computer Science and Engineering, Vol.2, No.3, 2010, pp 560-565.

[11] Deval G.Patel, "Point Pattern Matching algorithm for recognition of 36 ASL gestures", International Journal of Science and Modern Engineering, Vol.1, Issue 7, June 2013.

[12] Joyeeta Singha, Karen Das, "Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique", International Journal of Advanced Computer Science and Applications, Vol.4, No.2, 2013.

[13] D.Y.Huang, W.C.Hu, and S.H.Chang, "Vision based Hand Gesture Recognition Using PCA+Gabor filters and SVM", IEEE Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009, pp 1-4.

[14] Ali Karami, Bahman Zanj, Azadeh Kiani Sarkalesh, "Persian sign language (PSL) recognition using wavelet Transform and neural Networks", ELSEVIER Journal of Expert Systems with Applications 38, 2011, 2661-2667.

[15] Manigandan. M, I. M. Jackin, "Wireless Vision based Mobile Robot control using Hand Gesture Recognition through Perceptual Color Space", IEEE International Conference on Advances in Computer Engineering, 2010, pp.95-99.

[16] S.Saengsri, V.Niennattrakul, and C.A.Ratana mahatana, "TFRS: Thai Finger-spelling Sign Language Recognition System", IEEE, 2012, pp 457-462.

[17] ASL Dataset. https://www.kaggle.com/datamunge/sign-language-mnist/, Sign Language MNIST, Kaggle, 2017.