Classification of hand written numeric digits

TEAM 6 - Saketh Pachika, Sreevidya Baddam, Vivek Nichenametla

4/30/2021

Objective :

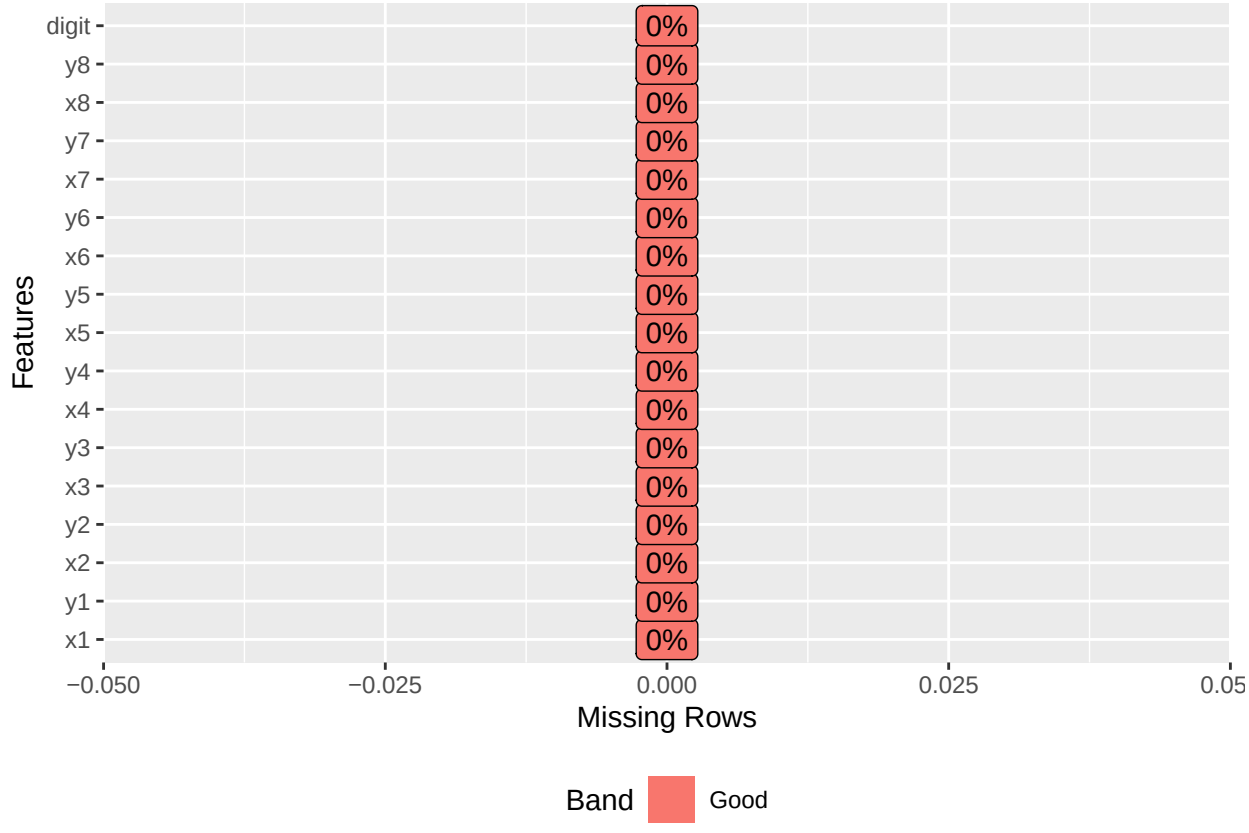
To build a multi-class classification model which can accurately classify the numeric digits written on a pressure sensitive tablet.

Data Overview:

The database created has 250 samples for each of 44 writers, 30 writers are used for training and 14 writers are used for testing purpose, The data-set is normalized and re-sampled such that each digit is represented by a sequence of 8 points as (x,y) coordinates, the data is scaled between 0-100

The scaled data has no missing values and both the training and test data has the least imbalance.

Missing Data Plot:



Model Building:

Different classification models are constructed and trained with the training data and further used them to classify the digits on the test data

Train-Test-Split

Going by the requirement, five random files of zip code digits '14260' have been separated from the training set. A data split of 70-30 is performed on training set, 70% data will be used to build model whereas the rest of the data will be used for validation and optimisation of model by tuning k value. In addition to that , k-fold validation was also performed, but for the little difference in the results. As k-fold is computationally expensive, 70-30 split preferred.

K-Nearest Neighbors:

K-Nearest Neighbors is a classification algorithm that uses feature-similarity to predict the values of new data-points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. It calculates the distance between test data and each row of training data with the help of the methods like Euclidean, based on which a class is assigned to the test point.

The model is trained and tested for k values ranging from 1 to 15 and accuracy is obtained at each k value. Based on the efficiency, the model takes the optimum value of k as 5

KNN Confusion Matrix

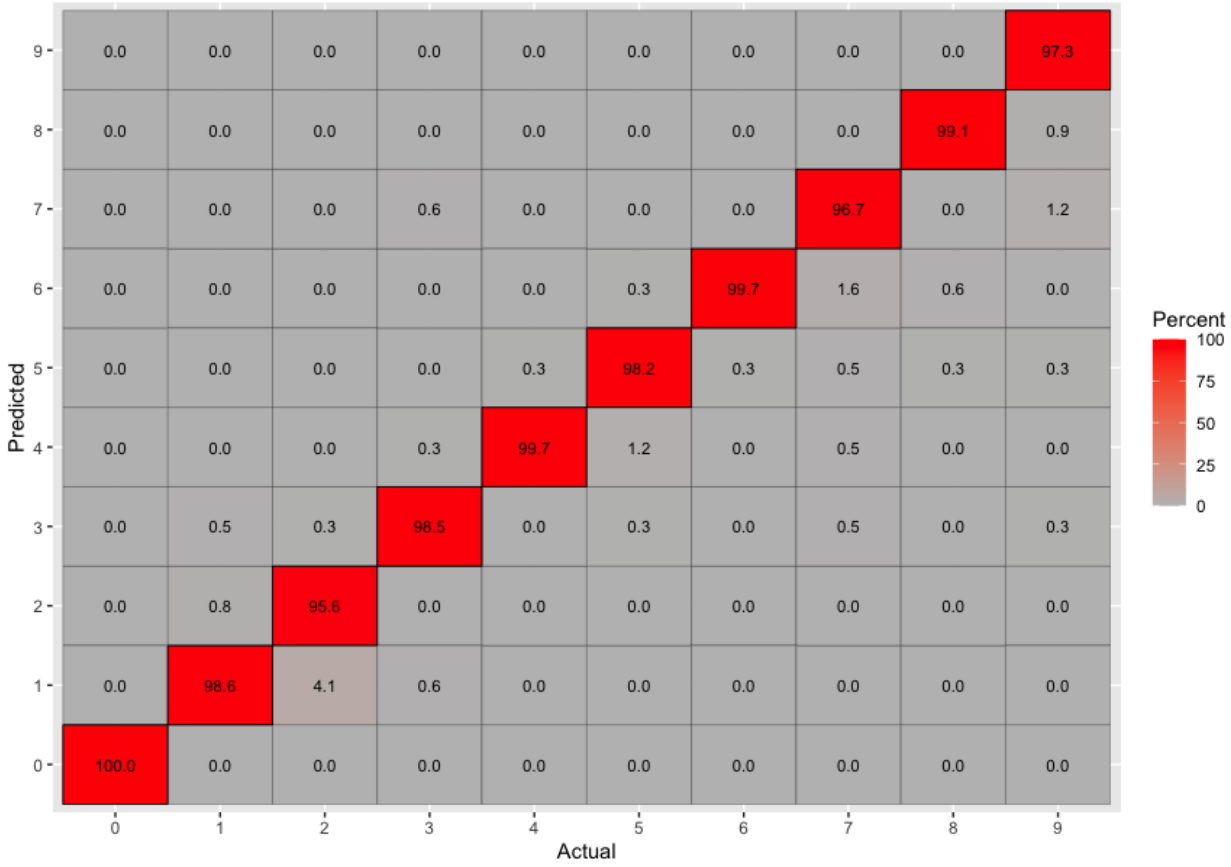


image :

Results:

Knn yielded best result of 99.1 when k is 5 on training set and 97.6 when tested on the testing file

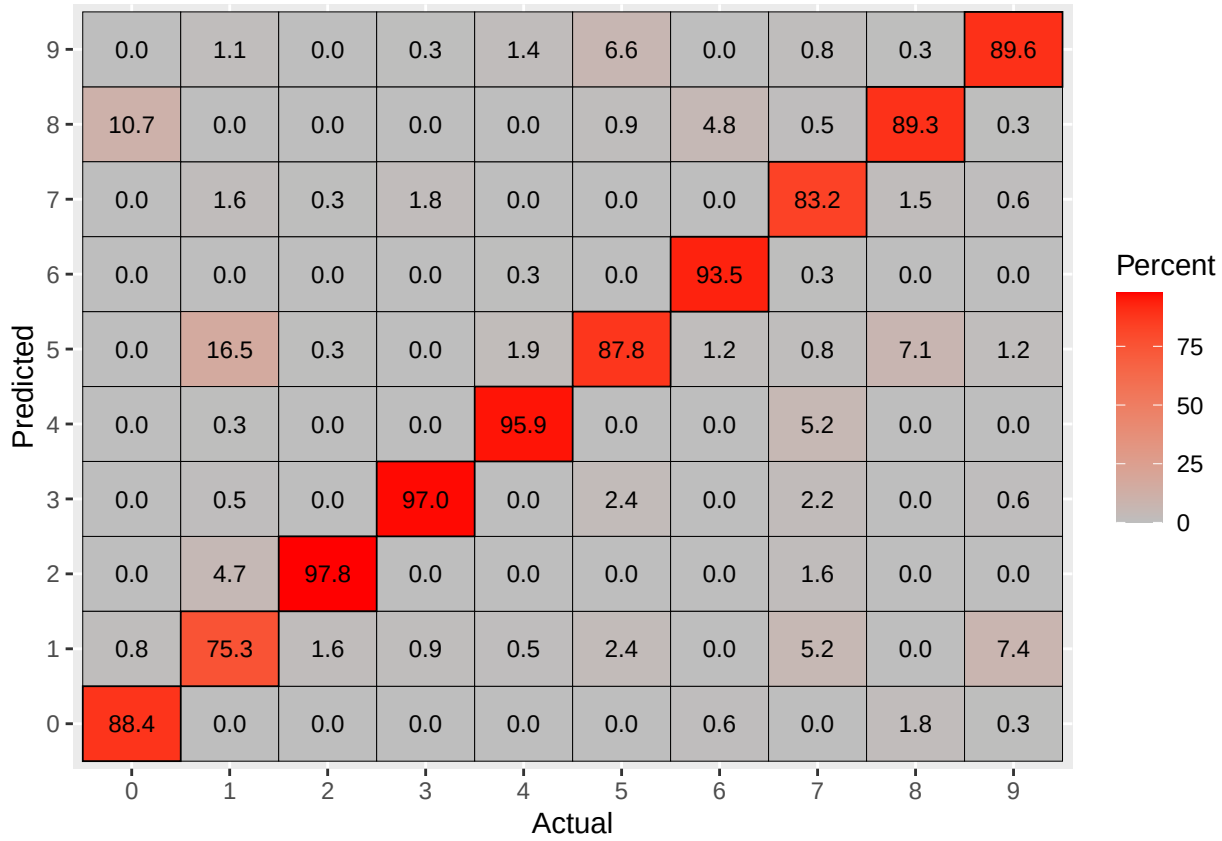
The KNN model accurately predicted the input files associated to digits 14260.

Multinomial logistic Regression

Multinomial logistic regression is used to predict a nominal dependent variable given one or more independent variables. It is an extension of binomial logistic regression for more than two variables. Like binary logistic regression, multinomial logistic regression uses maximum likelihood estimation to evaluate the probability of categorical membership.

The same 70-30 split of training data is performed. The train-test data gave an accuracy of 94%. The same model was run on testing data. The results are shown below.

```
## # weights:  180 (153 variable)
## initial  value 17244.059761
## iter  10 value 3816.152522
## iter  20 value 2388.161850
## iter  30 value 2219.410763
## iter  40 value 2208.495615
## iter  50 value 2204.737860
## iter  60 value 2202.312866
## iter  70 value 2198.129189
## iter  80 value 2162.886211
## iter  90 value 1846.678882
## iter 100 value 1758.988132
## final  value 1758.988132
## stopped after 100 iterations
```



Results:

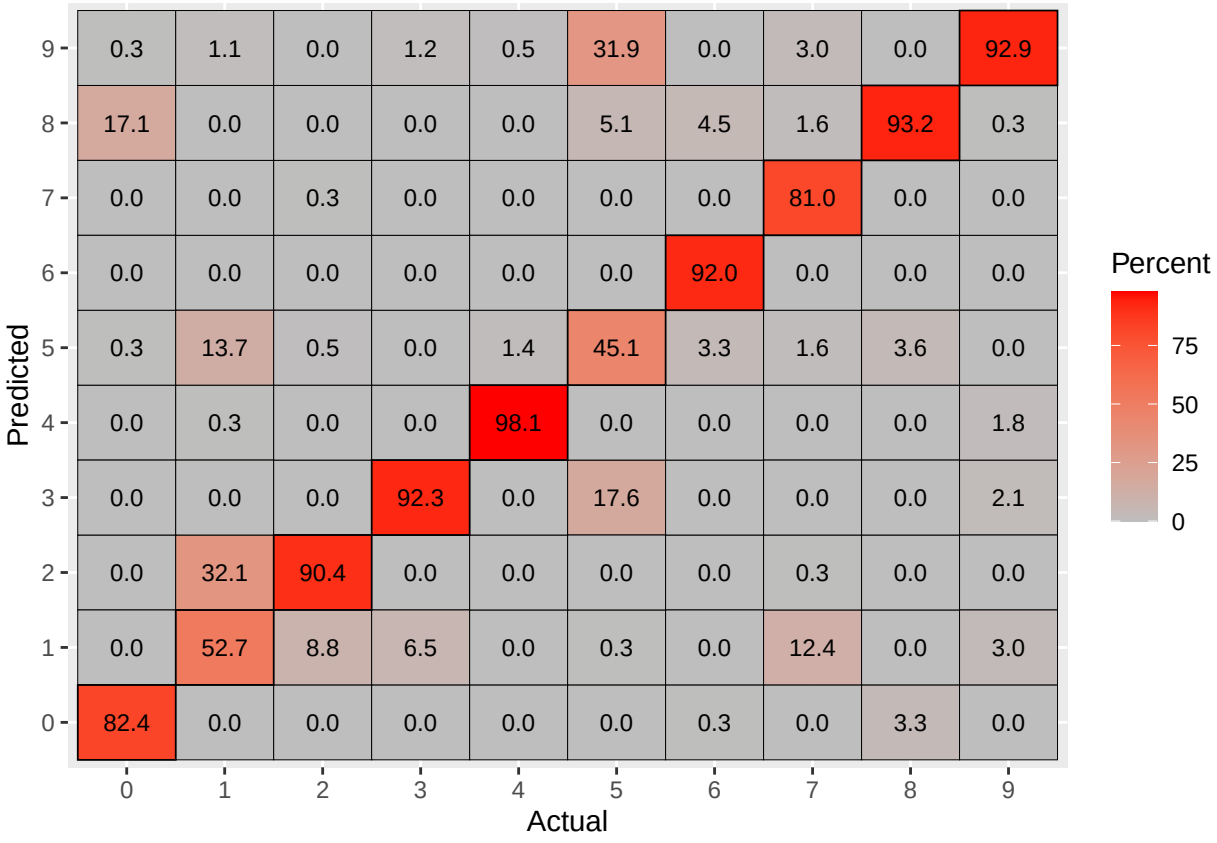
The model gives an accuracy of 94% when tested on training set and 89.7% on the testing data set. MLR considers the data to be liner and it models the data as a liner combination of predictor variabes.

The Multinomial regression predicted the 4 digits of the pincode 14260 correctly from the random files data set.

Naive_Bayes

Naive Bayes is a family of probabilistic algorithms that take advantage of probability theory and Bayes' Theorem to predict the tag of a text (like a piece of news or a customer review). They are probabilistic, which means that they calculate the probability of each tag for a given text, and then output the tag with the highest one. The way they get these probabilities is by using Bayes' Theorem, which describes the probability of a feature, based on prior knowledge of conditions that might be related to that feature.

Same training and testing split of the training data has been used. The train-test data gave an accuracy of 87.38%. The same model was run on testing data. The results are shown below.



Results:

The model gives an accuracy of 81.96%. This is expected of naive-bayes model as it is not very efficient in modeling for multi-class classification.

Prediction:

The naive-bayes model has also predicted the pincode lines of the data-set correctly.

Support Vector Machines:

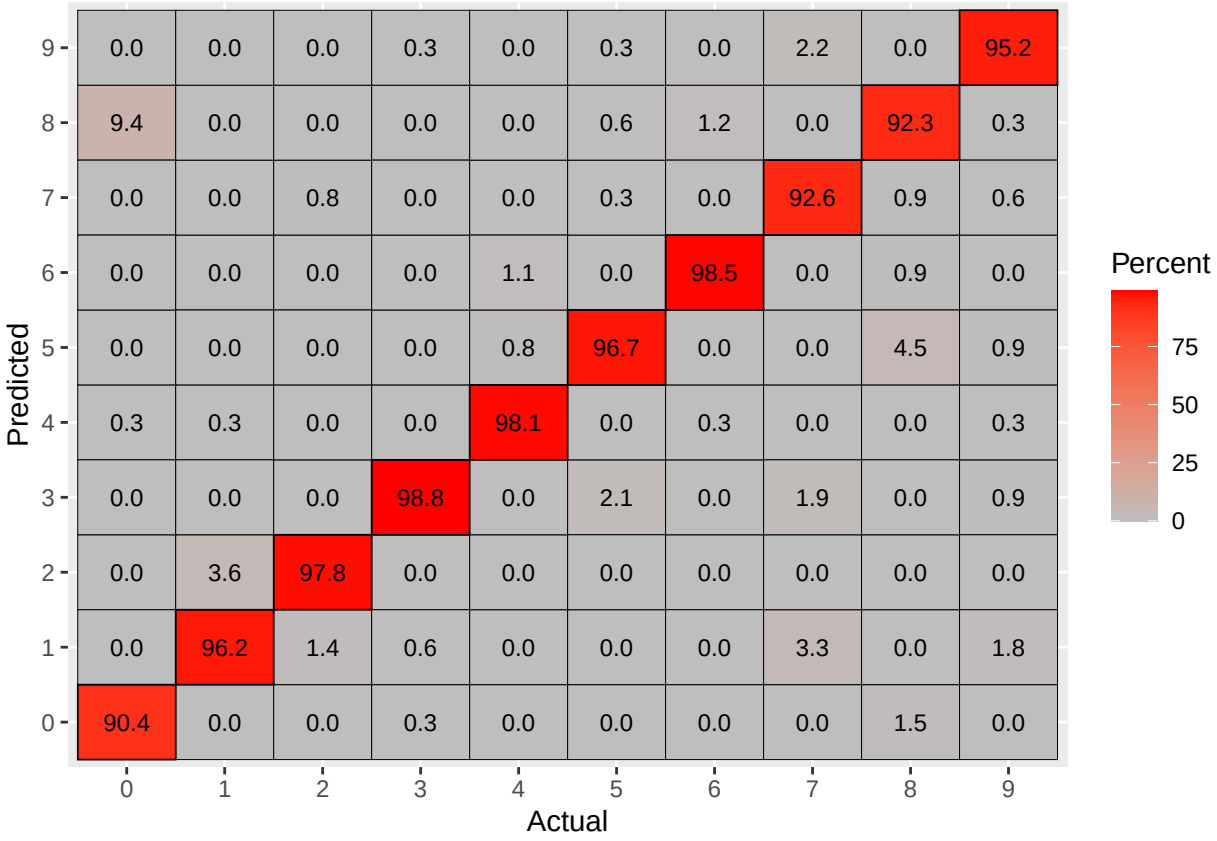
The *SVM classifier* is *a* frontier which best segregates the two classes (hyper-plane/ *line*).

Multiclass SVMs are usually implemented by combining several two-class SVMs. The one-versus-all method using winner-takes-all strategy and the one-versus-one method implemented by max-wins voting are popularly used for this purpose.

One-vs-one method was used for this classification. The model was trained and tested on the same train-test data.

The train-test data gave an accuracy of 98.87%. The same model was used to run it for the testing data(.tes file)

Results of the testing-set



Results:

The model gives an accuracy of 95.63% on the testing data. This is one of the most durable and efficient models for classification.

Neural Network:

Artificial Neural Network is a computational algorithm which can be used for pattern recognition and machine learning, these are presented as systems of interconnected neurons which can compute values from inputs

Building an ANN model is very expensive and requires lot of data to train and prone to over-fitting sometimes.

Current classification problem used an ANN with 2 hidden layers and 6,4 neurons per each layer with step-max of 1e+06 using a logistic activation function.

The model required very high processing power and did not converge in the cases with high threshold,In the iterations that were converged the average accuracy was pretty low at ~70%

Neural Network Model Structure:

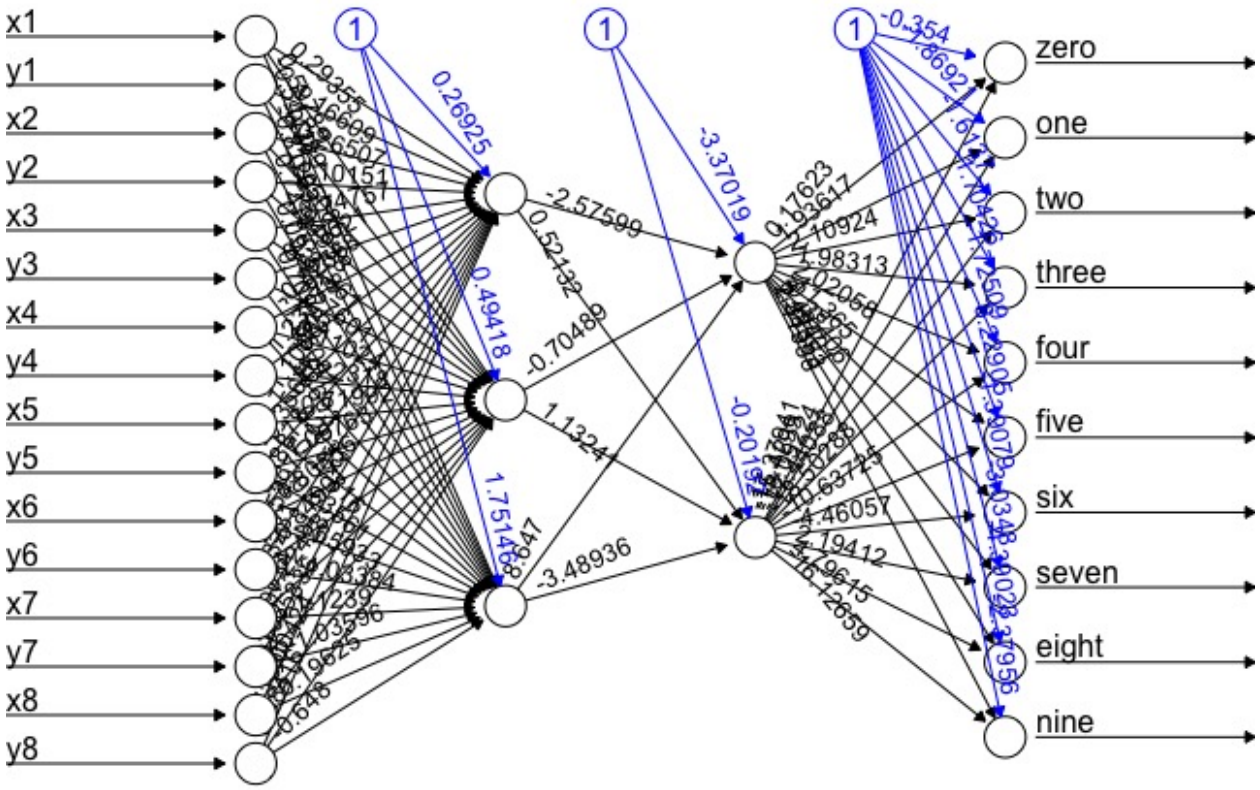


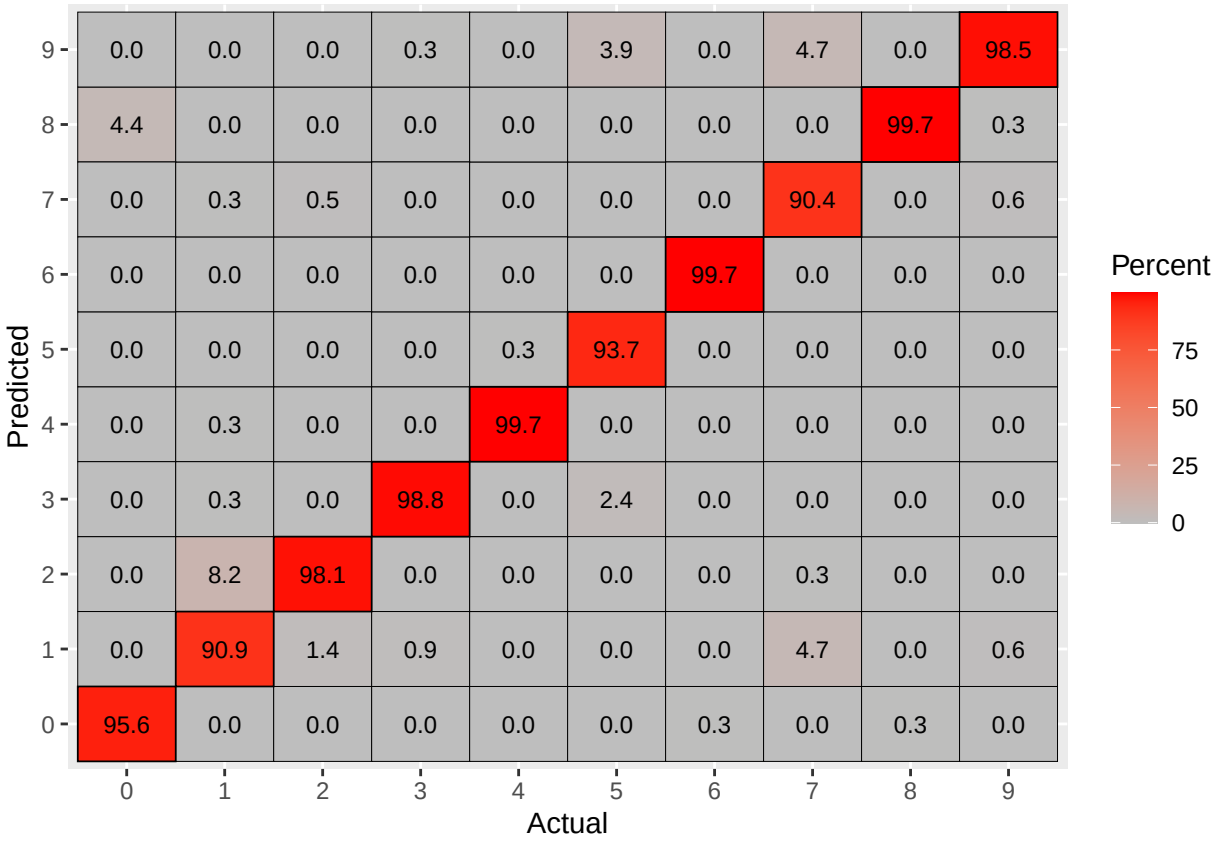
image :

Random Forests:

Random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

The number of trees are taken to be 1000 for stable results, and mtry value was taken as default that is 3(sqrt(9)). The confusion matrix displayed above gave an accuracy of 99.33 % for the test data of the training set. Similar trials for mtry=6 and mtry=9 are performed which yielded the results of 99.07% and 98.53% respectively.

What is mtry? - The number of variables to be considered at each split. When mtry= 16 it is similar to running a model for bagging.



Results:

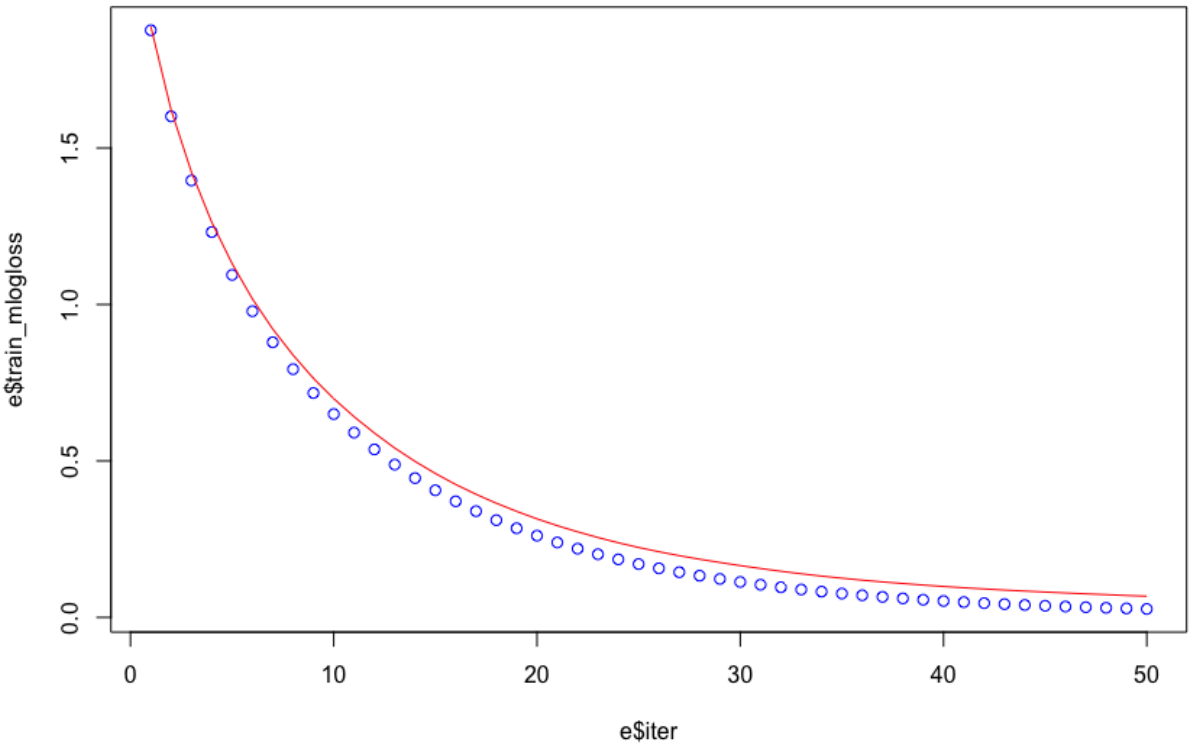
The random forests yielded best result of 99.33 at default random forest model, which considers sqrt(no.of variables) while splitting the trees. The test data-set on the same model gave an accuracy of 96.46%

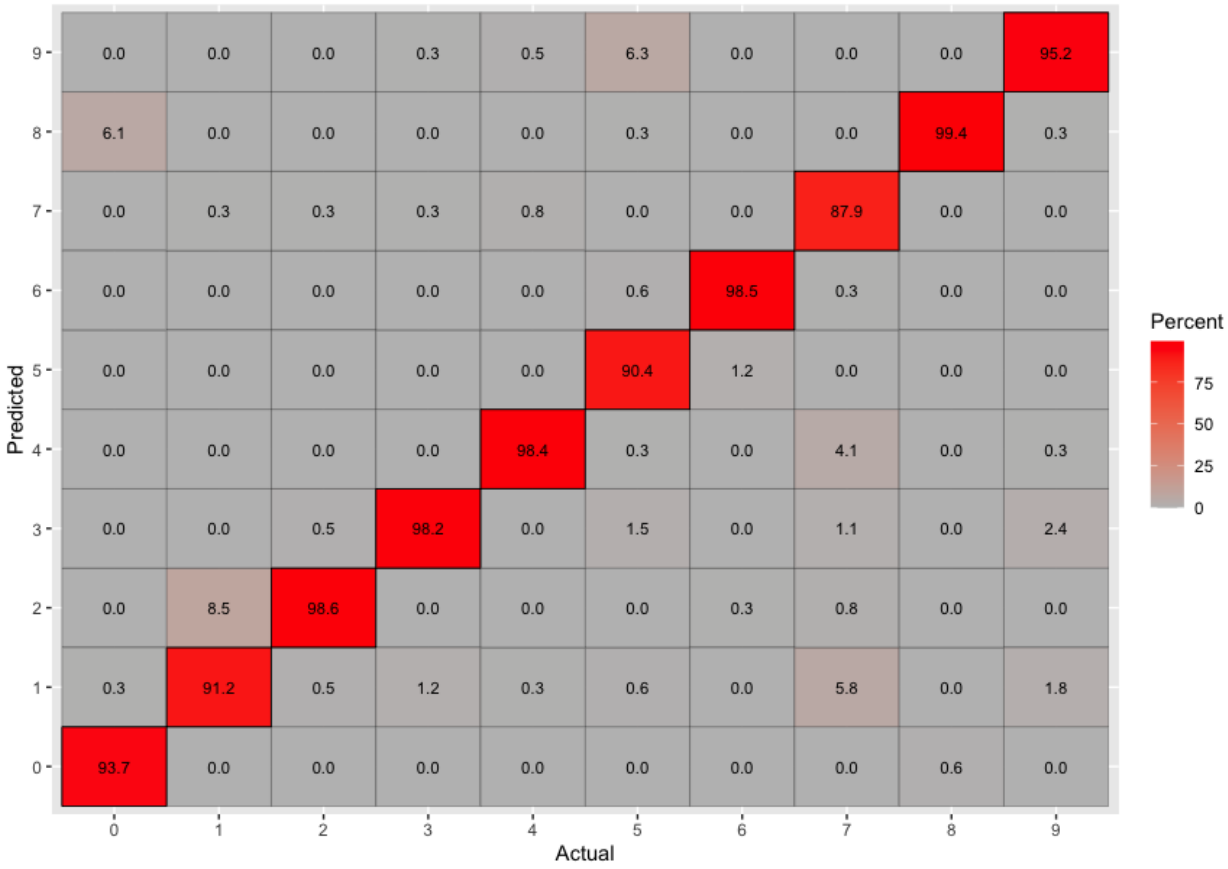
XG Boost:

XG Boost is an implementation of gradient boosted decision trees, three main forms of gradient boosting are supported gradient boosting, stochastic gradient boosting and regularized gradient boosting, it has ability to do parallel computing on a single machine and is faster

For classification we used gbtree as the booster

Error Vs Iteration Plot


e$train_mlogloss
0.0
0.5
1.0
1.5
0
10
20
30
40
50
e$iter



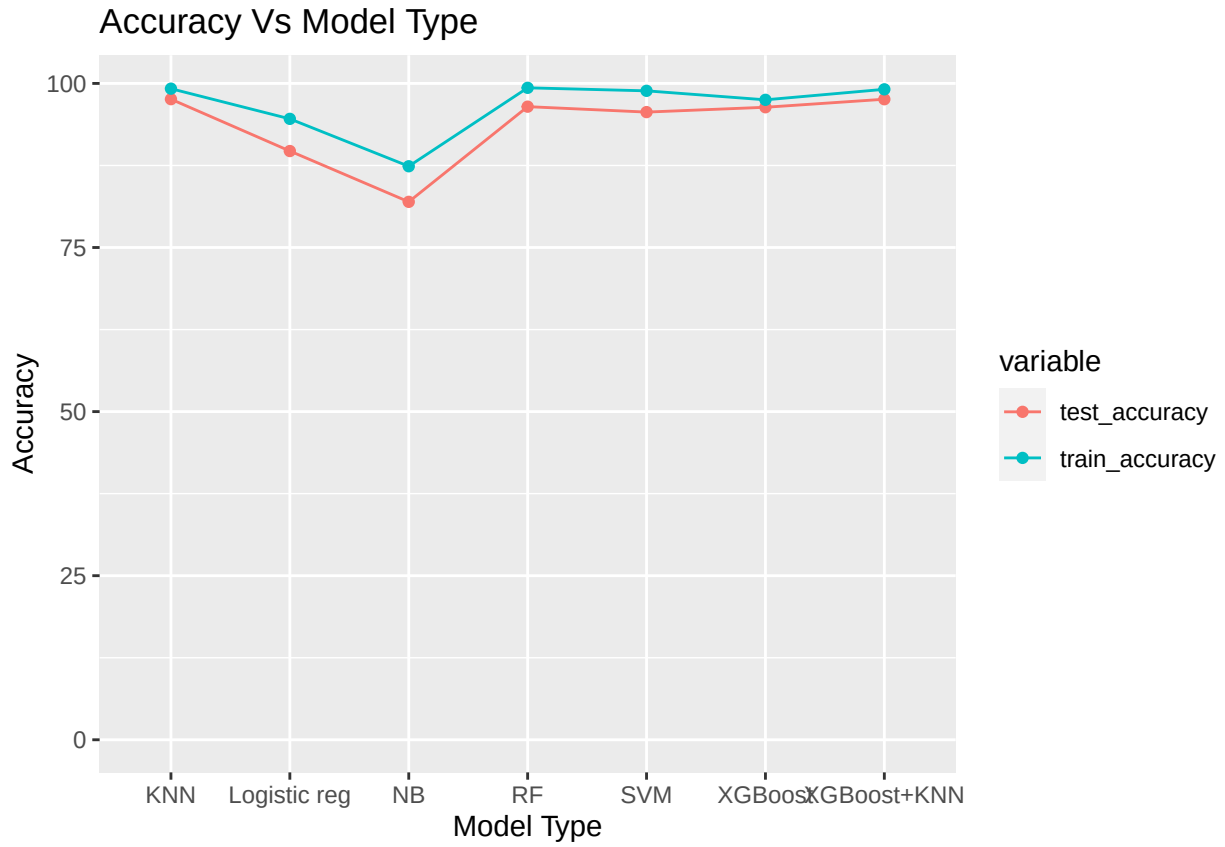
Results:

The XGBoost yielded an accuracy of ~ 96.4% on the test data

Ensemble Model:

Tried combining different classification models in order to get better accuracy on the test data, The Ensemble between XGBoost and KNN classifier has an improved accuracy over the individual models by 0.3%, base model of XGBoost is considered and KNN is applied to the predictors where the probability of the prediction <=0.7

Training and Test Accuracies of Various Models:



Conclusion:

The classification accuracy for most of the models constructed were above ~95%, among these Ensemble (XGBoost + KNN) and KNN performed best.Random Forests and SVM have also performed good with efficiencies of 96% and 95% respectively. As expected multi-class logistic regression, naive-bayes did not produce desirable results for multi-class classification.

Challenges:

Neural networks could not converge for the given data-set for many iterations mainly due to lack of computational resources. Going forward, neural network models could be leveraged as we have good amount of data.

Contribution:

Saketh Pachika- Gathered prior information for the project, implemented neural network, XG-Boost algorithms.

Sreevidya Baddam- Implemented knn algorithm, logistic regression algorithms.

Vivek Nichenametla- Random Forests, SVM and Naive-Bayes algorithms.

All the three have contributed to the project report rendering and additional research.

Citations:

[1] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2012). e1071: Misc Functions of the Department of Statistics (e1071), TU Wien. R package version 1.6-1. http://CRAN.R-

project.org/package=e1071 [2] Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0 [3] Keysers et al. “Deformation Models for Image Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 8. August 2007, pg. 1430. [4] A Survey of Handwritten Character Recognition with MNIST and EMNIST