DA2401 EndSem-Project

R Raghava Reddy DA24B021

October 2025

The Models Used:

- 1.) MultiClassXGB:From the XGB built in Assingnment-2,I've built and one vs all model for each class and made a final prediction
- 2.) RandomForest:An optimised Multiclass randomforest by setting minimum number of data points in each partition to further divide it.
- 3.) PCA:this was used to find direction with least variance
- 4.) KNN and Weighted-KNN:A normal KNN model and weighted KNN model with weight inversely proportional to distance between them
- 5.) Softmax Regression:Even though Softmax isnt an good model it can be used as an combiner in stacking models.

Models accuracy and hyper parameter tuning:

KNN:

Normal KNN and no PCA:

The KNN is expected to work good because the data is in clusters format.For normal KNN and k=5 without the model was giving accuracy of 0.9503 and when the k is increased or decreased from this the accuracy score falls off implying k=5 is best.

KNN with PCA:

As the data is in clusters,PCA is used to remove noise components from the data to make the model more accurate and after running for few components around 50 components the accuracy was looking optimal so I drew an heatmap to find the best parameters around this range and from the heatmap given below best hyperparameters found was k=6 and n=49.

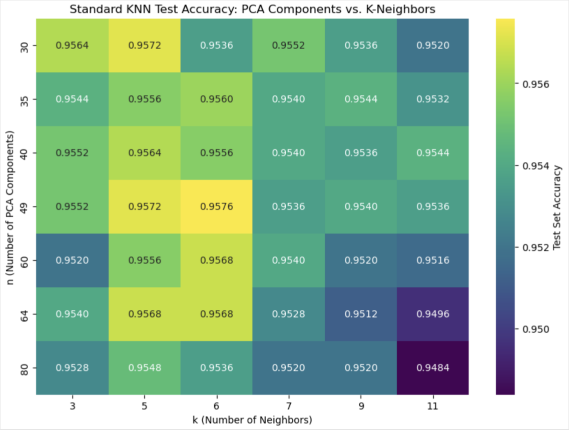


Figure 1: Normal KNN with PCA heatmap

Weighted KNN:

From previous hyper parameters I found ,I tried to run weighted KNN around similar hyperparameters and the heatmap corresponding to it is below

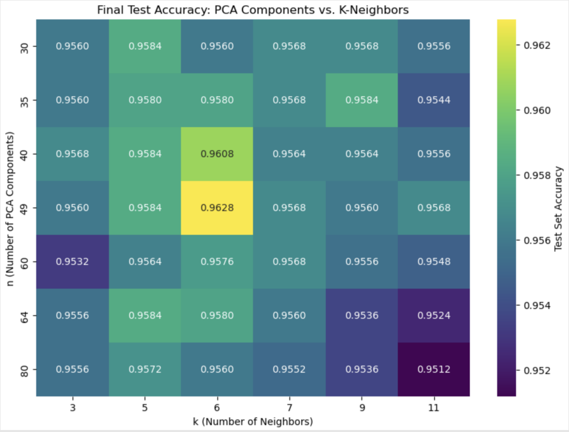


Figure 2: Weighted KNN with PCA heatmap

from the heatmap here and accuracy score we can clearly see that weighted KNN performs better than normal KNN and the best hyperparameters are k=6 and n=49.

RandomForest,XGBoost

RandomForest:

To begin with I tried running the model with small parameters like trees=60,depth=10 for this the model was giving accuracy of around 0.928 ,time taken was 48s .I tried increasing the features to trees=150,depth=15 and the accuracy score was 0.948,timetaken=1m50s the accuracy increased ,so it implies bias was higher so I further the parameters to number of trees=200,depth=20 the accuracy score fell off to 0.943 ,timetaken= 3mins implying increase in variance ,so the best parameters were trees=120 and depth=15.

XGBoost:

I tried running with low parameters n=30 and depth=4 the accuracy was around 0.938 and timetaken was 53s ,so i increased the parameters to n=50 and depth=60 ,the accuracy was fluctuating around 0.954 and it took 2mins 4s so bias was high ,i tried increasing the parameters to n=80 and depth=8 the accuracy score was still around 0.954 and it took 5mins 1s to run so XGBoost was not increasing any further within 5 mins constraint.

Ensembling Methods:

1.Stacking with all three models:

Here first all three models(KNN,XGBoost and Randomforest) are trained on 70-80% of train data and then using the rest of data i built an softmax model on the outcome of these models with remaining 20-30% of data to find how much weight to give for each model in final vote.After the weights are found all the model are retrained with full data and final predictions are taken with weight found from the softmax model.For this ensembling the accuracy score was around 0.957 ,the hyperparameters are kept in moderate amount to keep the run time under 6 mins.This model is still worse than just KNN so the conclusion is to use KNN ensemblers.

2.Stable KNN ensembling:

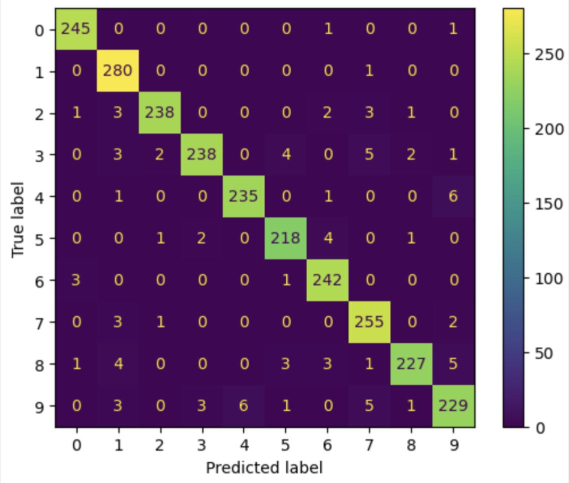
In this ensembling, Ive used multiple KNN for different K and different components.Here first the all models are trained with 60-70% of data and accuracy score is found for all models using rest of data.Now for each model is retrained with full data and then for final vote the weights are sqaured of accuracy scores from previous training.Also for the K=6 and n=49 I gave an additional bias by adding that model multiple times.This indirectly means pick most of the other models agree to an answer but k=6 and n=49 disagrees then pick the majority vote or else pick the k=6 and n=49 model answer.This model was performing slightly better than just k=6 and n=49.
The final chosen ensembling is KNN ensembling as it outperforms the Stacking ensembler significantly.

Other methods tried which werent explicitly allowed:

I implemented some features manipualtions to make more significant features.The manipulation i did was called sobel features add an gradient component to features,this is indirectly adding an edges of shape as features.This model was performing very good for even very basic model.It was giving a score around 0.973 for simple models like SVM and Bayes .But feature manipulation was not allowed in final submission

Summary and conclusion:

- 1.) Even if the algorithm is very good ,there is no ”best algorthm” for all types of setting .In general XGboost model is assumed to best for non-neural network models but here we can clearly see that KNN outperforms the benchmark XGBModel.



True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	245	0	0	0	0	0	1	0	0	1
1	0	280	0	0	0	0	0	1	0	0
2	1	3	238	0	0	0	2	3	1	0
3	0	3	2	238	0	4	0	5	2	1
4	0	1	0	0	235	0	1	0	0	6
5	0	0	1	2	0	218	4	0	1	0
6	3	0	0	0	0	1	242	0	0	0
7	0	3	1	0	0	0	0	255	0	2
8	1	4	0	0	0	3	3	1	227	5
9	0	3	0	3	6	1	0	5	1	229

Figure 3: Confusion Matrix for final model

- 2.) The confusion for final model is given above here we can see that model cant distinguish between 7 and 9,7 and 3 and other somewhat similar numbers.This can be overcome by using feature manipulations and neural networks

- 3.) Generally PCA is used as speedbooster but here PCA is used as noise filter and it significantly increases the accuracy.This also due to curse of dimensionality for KNN model.

- 4.) An ensemble is only as good as its members. Ensembling a strong model with weaker, noisier models can degrade performance.My successful strategy KNNensembler proved it’s better to build an ensemble of diverse variations of your single best-performing algorithm.

- 5.)For the Random Forest, increasing depth from 10 to 15 improved accuracy, showing the initial model was underfit (high bias). However, increasing to a depth of 20 caused accuracy to drop, indicating overfitting (high variance).A similar thing was found for XGBoost, which hit its peak performance around 2 minutes of training and showed no significant improvement with more time, suggesting it had reached its maximum potential on this dataset.