Introduction

To eb aligned with previous works, this report used 30% of the whole dataset as testing data. The way to select them is by random state=1. Experiments with different parameters are under cross validation when fold equals to 3.

The train set has no missing values among the target variable and all features. As k-Nearest Neighbors is very sensitive to extreme values, this report transformed features in the preprocessing step. With various range of features, it is necessary to normalize them. Although there is no information about the distribution, given the large scale of dataset, it is safe to take the normal distribution as granted. After Z-scoring, all features are in the range from [0,1].

The following of this report employed k-Nearest Neighbors, Random Forest, Support Vector Machine, and Gradient Boosting to classify the dataset. The key rate for evaluation is the accuracy on the test set. It reveals how well the model predicts the outside data. Besides accuracy, overfitting is also am important issue on tuning parameters. If huge differences between accuracy on testing and training data appear, the associated settings have poor understanding of new information. This report firstly tuned on one or two relative important parameters in each model, plotting validation curves to narrow down the range of these parameters. Then implemented by Grid Search and CV, each model was built on combinations of all necessary parameters. This approach selected the best parameter setting based on the accuracy. With the optimal classifier with each algorithm, learning curves were used to capture the pattern over different training sample size with a specific cross validation. From the trending, one can understand how fast those optimal classifiers learnt from various training size and their performance.

Due to the limitation of available device, in the SVC model, this report didn't touch Grid Search and CV method.

k-Nearest Neighbors

K-Nearest Neighbors collects all training samples at first. The way to determine the category of test data is "voting". In k nearest datapoints of the test data, if the majority of training data belong to a specific category, then the test data is determined to be in the same category. The most important parameter is how many neighbors should be considered. Below is the validation curve over different number of neighbors.

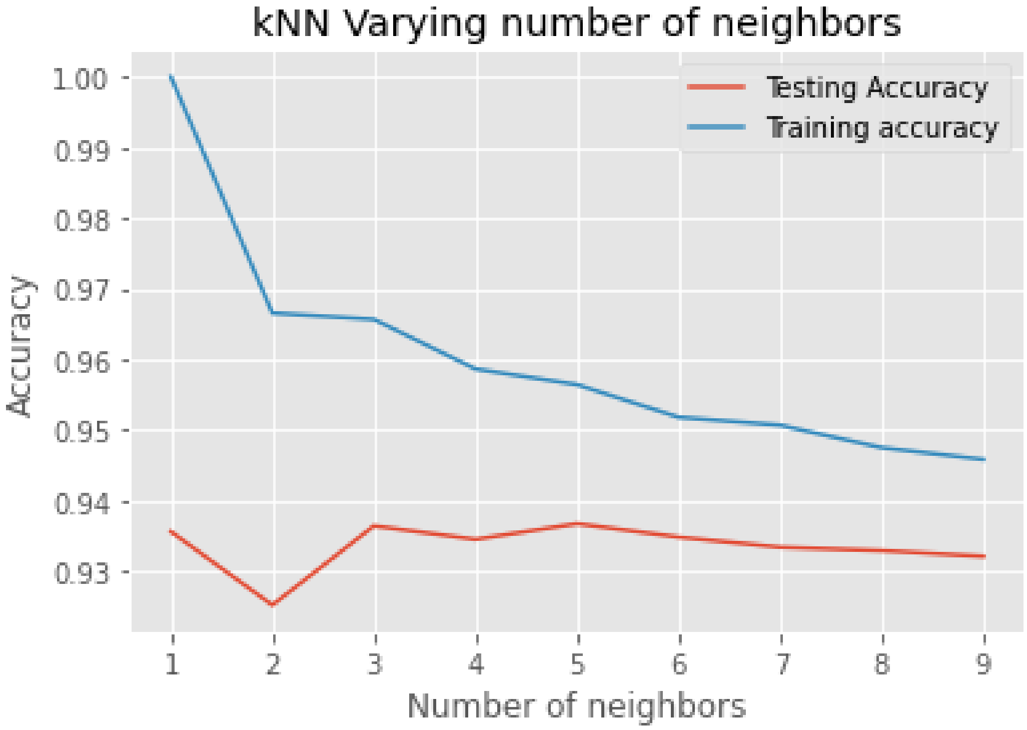


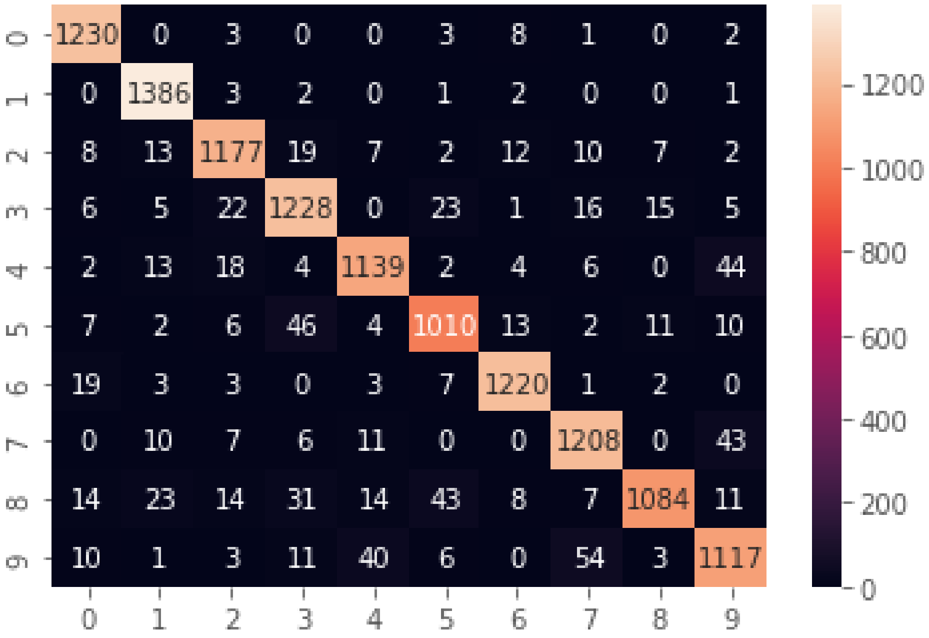
Figure-1 Validation Curve over number of neighbors(k-NN)

The test accuracy is the key rate. From Fig-1, it is clear that when number of neighbors is in the range of 3 to 9 leads to better performance.

With the smaller range of neighbors, next step is to tune as much as parameters. The optimal setting is:

$$\{neighbor = \ 3, distance = Euclidean, Weight\ function = uniform\}$$

The test accuracy of the above classifier is 0.9364.



	0	1	2	3	4	5	6	7	8	9
0	1230	0	3	0	0	3	8	1	0	2
1	0	1386	3	2	0	1	2	0	0	1
2	8	13	1177	19	7	2	12	10	7	2
3	6	5	22	1228	0	23	1	16	15	5
4	2	13	18	4	1139	2	4	6	0	44
5	7	2	6	46	4	1010	13	2	11	10
6	19	3	3	0	3	7	1220	1	2	0
7	0	10	7	6	11	0	0	1208	0	43
8	14	23	14	31	14	43	8	7	1084	11
9	10	1	3	11	40	6	0	54	3	1117

Figure-2 confusion matrix (k-NN)

Table-1 Performance report(k-NN)

	precision	recall	f1-score	support
0.0	0.95	0.99	0.97	1247
1.0	0.95	0.99	0.97	1395
2.0	0.94	0.94	0.94	1257
3.0	0.91	0.93	0.92	1321
4.0	0.94	0.92	0.93	1232
5.0	0.92	0.91	0.91	1111
6.0	0.96	0.97	0.97	1258
7.0	0.93	0.94	0.93	1285
8.0	0.97	0.87	0.91	1249
9.0	0.90	0.90	0.90	1245
accuracy			0.94	12600
macro avg	0.94	0.94	0.94	12600
weighted avg	0.94	0.94	0.94	12600

The confusion matrix shows most categories has great prediction. In the performance report, F1-score are similar among all categories. So far this k-NN classifier is reliable.

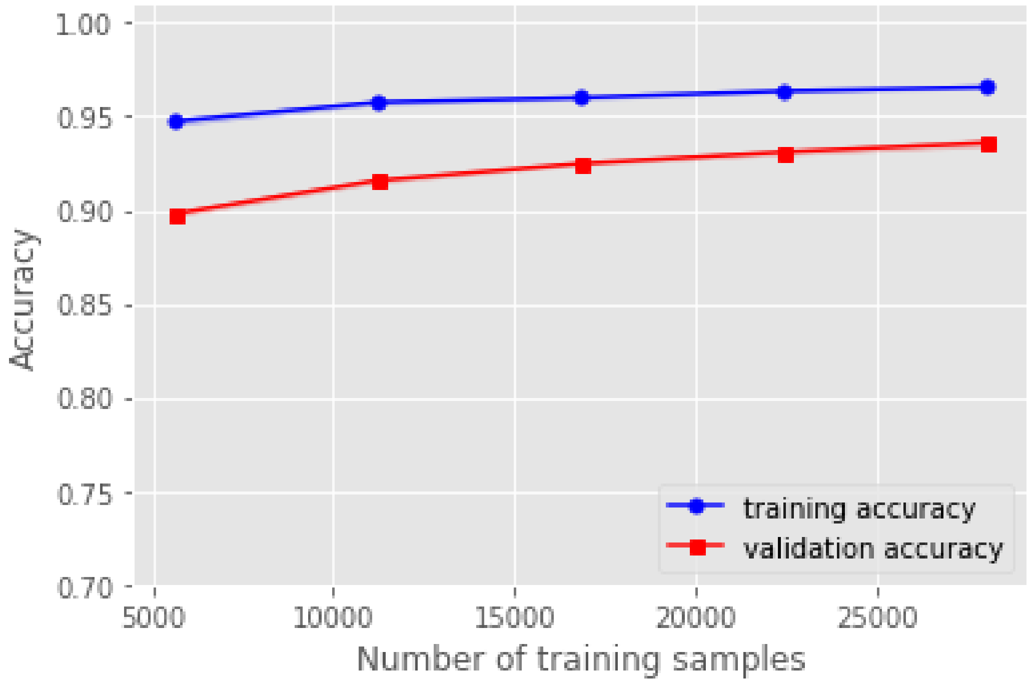


Figure-3 Learning Curve (k-NN)

The learning curve of this model shows possible overfitting issue, especially when training sizes is small. With enlarged datasets, the two learning curves start to converge and the ultimate difference is less than 4%. If possible, this method could be improved by dropping features giving same value to all samples. Based on selected features, the computation time will be limited, and more parameters are allowed to be tuned.

Support Vector Machine

Support Vector Machine is to divide dataset with data points in the closet neighborhood of the “boundary”. In most cases, there is no absolute best division. To include how likely the classifier misunderstand a sample, the cost parameter C is determinant.

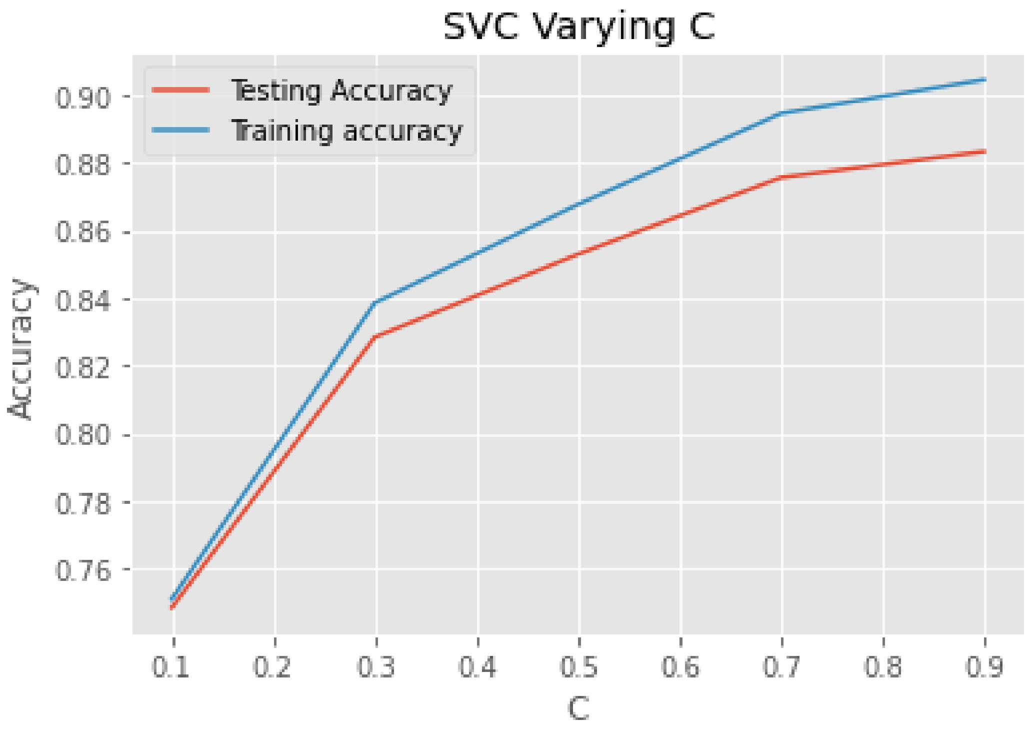
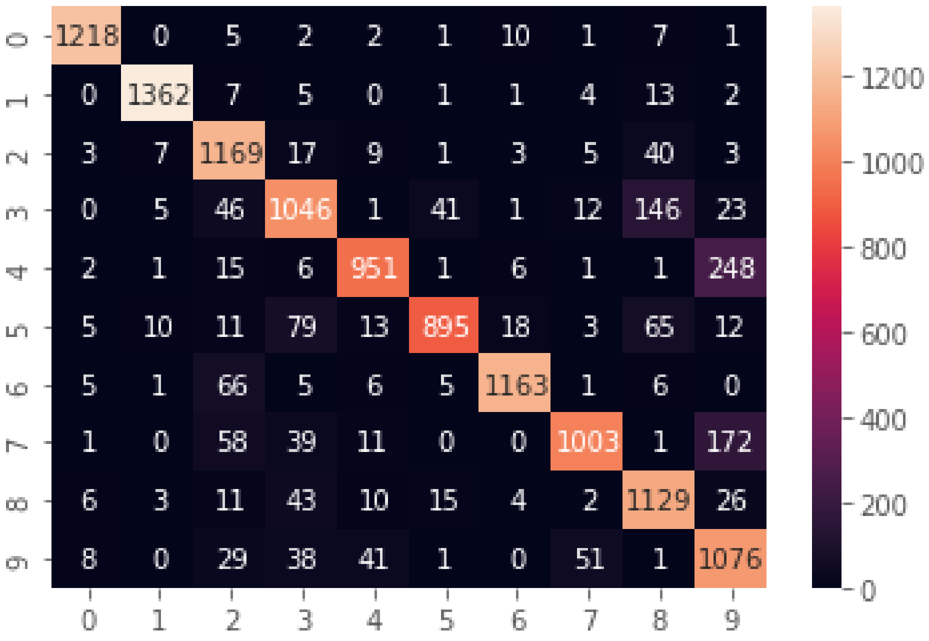


Figure-4 Validation Curve over different values of C(SVC)

Now compare the test accuracies over different values of C. From Fig-4, it is clear that when the cost of misclassification upwards, the accuracy rises up. In the graph, when C=0.8, the accuracy level and the difference of these two curves both achieved a relative better performance.



	0	1	2	3	4	5	6	7	8	9
0	1218	0	5	2	2	1	10	1	7	1
1	0	1362	7	5	0	1	1	4	13	2
2	3	7	1169	17	9	1	3	5	40	3
3	0	5	46	1046	1	41	1	12	146	23
4	2	1	15	6	951	1	6	1	1	248
5	5	10	11	79	13	895	18	3	65	12
6	5	1	66	5	6	5	1163	1	6	0
7	1	0	58	39	11	0	0	1003	1	172
8	6	3	11	43	10	15	4	2	1129	26
9	8	0	29	38	41	1	0	51	1	1076

Figure-5 Confusion matrix (SVC)

Table-2 performance report(SVC)

	precision	recall	f1-score	support
0.0	0.98	0.98	0.98	1247
1.0	0.98	0.98	0.98	1395
2.0	0.82	0.93	0.87	1257
3.0	0.82	0.79	0.80	1321
4.0	0.91	0.77	0.84	1232
5.0	0.93	0.81	0.86	1111
6.0	0.96	0.92	0.94	1258
7.0	0.93	0.78	0.85	1285
8.0	0.80	0.90	0.85	1249
9.0	0.69	0.86	0.77	1245
accuracy			0.87	12600
macro avg	0.88	0.87	0.87	12600
weighted avg	0.88	0.87	0.87	12600

The confusion matrix shows most categories has great prediction. In the performance report, F1-score are around 0.87 among all categories. So far this SVC classifier is reliable.

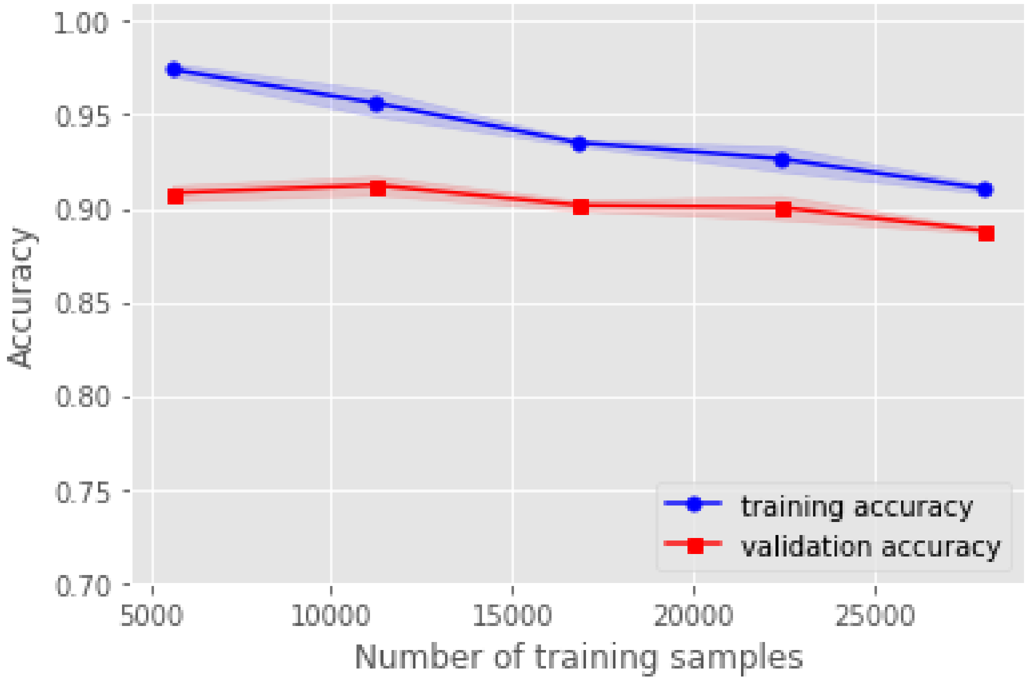


Figure-6 Learning Curve (SVC)

The learning curve of this model reveals less overfitting issue, especially when training sizes is large. With the full datasets, the two learning curves start to converge and the ultimate difference is less than 2%. If possible, this method could be improved by tuning on gamma. The bets separation might not be linear, given the high dimensional features. Based on different values of gamma, it relaxes how to calculate the similarity between datapoints and the way to separate them.

Random Forest

Random Forest is the ensembled method of Decision Tree. To be aligned with homework 8, here similar parameters settings are employed again. It draws some features and forms subsets of features randomly, and then constructs decision trees for each subset. In the last, it ensembles classification rules form all trees. Since the max-depth for each individual tree is relative less important than number of trees, this report focuses more on the number of trees.

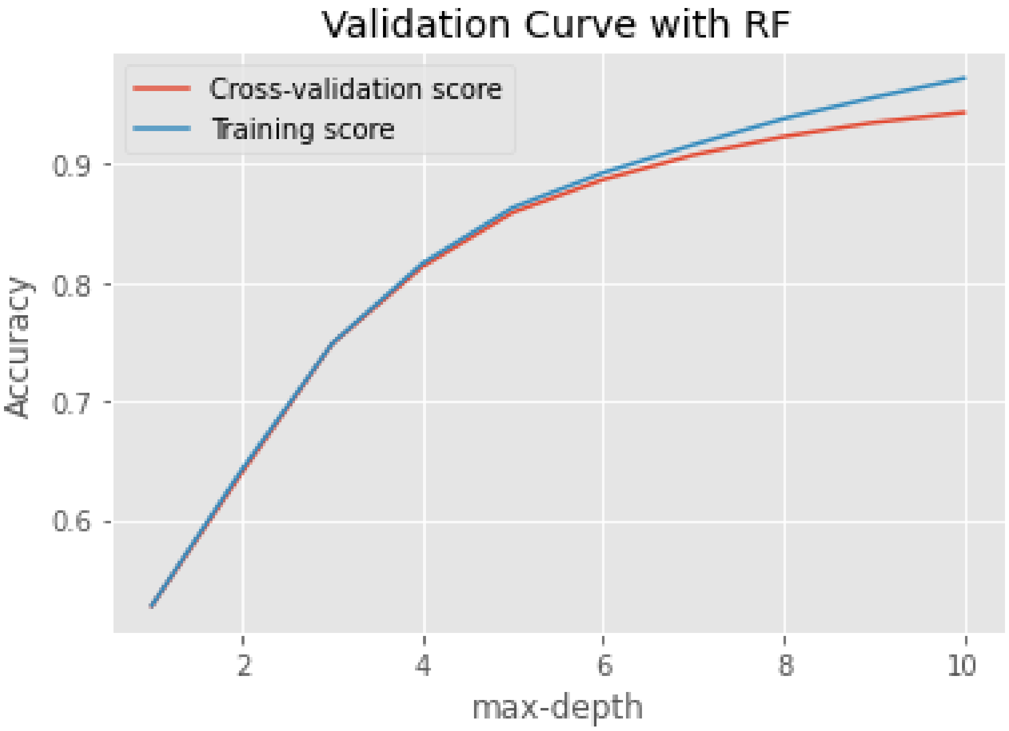


Figure-7 Validation Curve over max-depths(Random Forest)

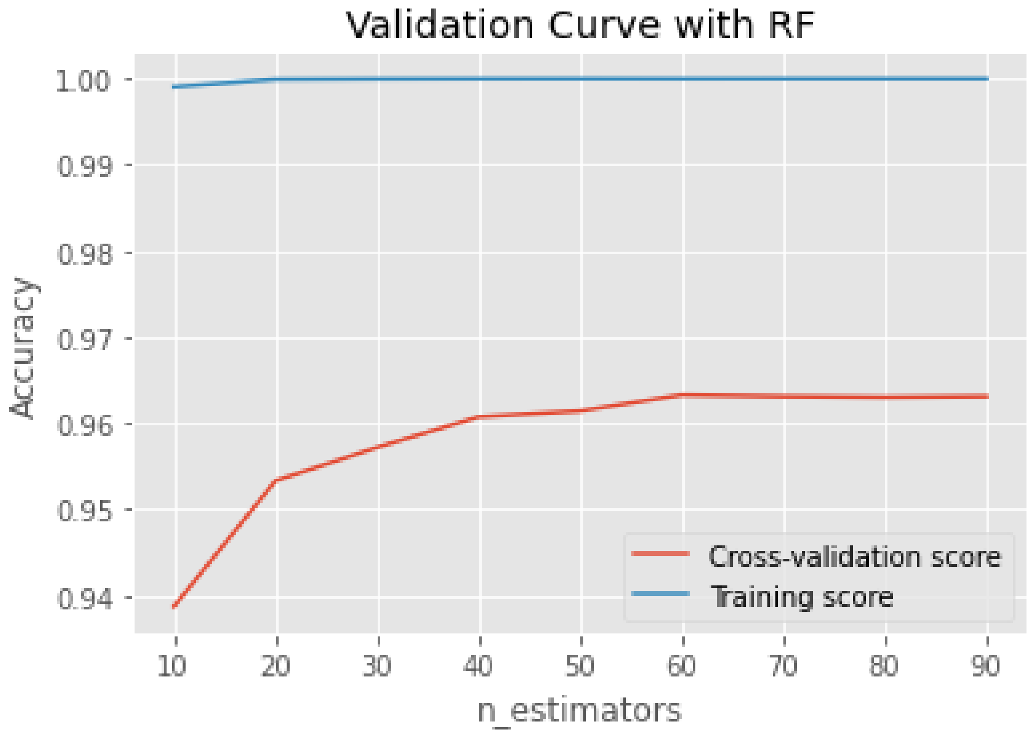


Figure-8 Validation Curve over number of estimators (Random Forest)

From Fig-7, it is clear that when max-depths are in the range of 4 to 8 leads to better performance in accuracy. From Fig-8, the test accuracy reached the highest level when the number of estimators is from 60 to 90.

With the smaller range of max-depths and the number of trees, next step is to tune as much as parameters. The optimal setting is:

$$\{the\ number\ of\ trees = \ 100,\ \max depths = 6, \max \quad features$$
$$= 'log2', criterion =' \ entropy', \qquad min\_samples\_leaf = 786,$$
$$min\_samples\_split = 1572, \}$$

The test accuracy of the above classifier is 0.8077.

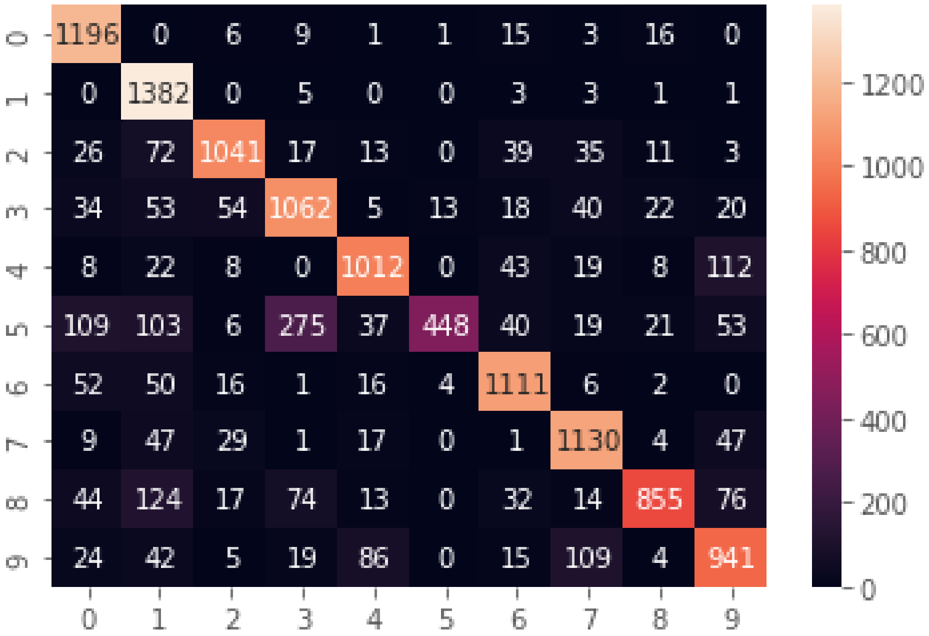


Figure-9 confusion matrix (Random Forest)

Table-3 Performance report(Random Forest)

	precision	recall	f1-score	support
0.0	0.83	0.97	0.90	1247
1.0	0.75	0.98	0.85	1395
2.0	0.86	0.78	0.81	1257
3.0	0.71	0.77	0.74	1321
4.0	0.86	0.77	0.81	1232
5.0	0.94	0.41	0.57	1111
6.0	0.81	0.85	0.83	1258
7.0	0.80	0.89	0.85	1285
8.0	0.86	0.73	0.79	1249
9.0	0.73	0.80	0.76	1245
accuracy			0.80	12600
macro avg	0.82	0.79	0.79	12600
weighted avg	0.81	0.80	0.79	12600

The confusion matrix shows most categories has great prediction, except for samples with label=5. In the performance report, most F1-score are around 0.80. Samples in label=5 has significant lower F1-score. But still, this classifier is not totally unfunctional.

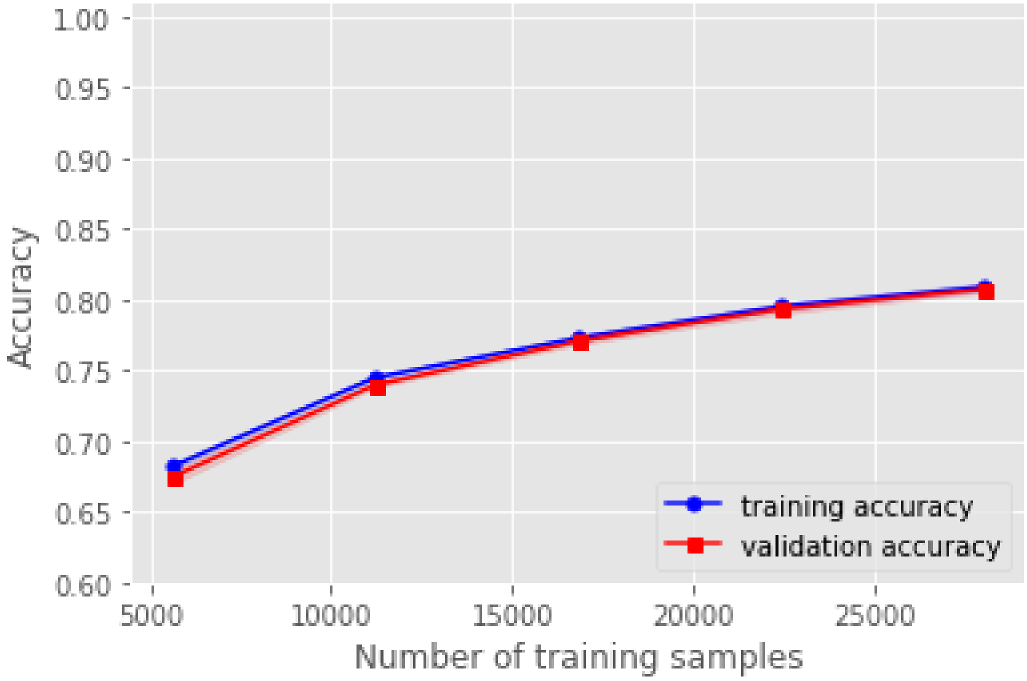


Figure-10 Learning Curve (Random Forest)

The learning curve of this classifier illustrates no overfitting issue, especially when training sizes is large. With the full training samples, the two learning curves converges and the ultimate difference is less than 0.5%. Although this Random Forest has lower accuracy, but its better performance on fitted subject indicates it is a reliable prediction model. If one urges to improve accuracy rates, this report suggests lifting up the number of estimators.

Gradient Boosting

This approach is also an ensemble method. The basic idea is the following: current classifiers learn from previous classifiers, focusing on error produced by previous classifiers. Therefore, the loss goes down in this process of learning. One of the most important parameters is learning rate, which measures in each iteration, how fast classifiers learned from the previous loss.

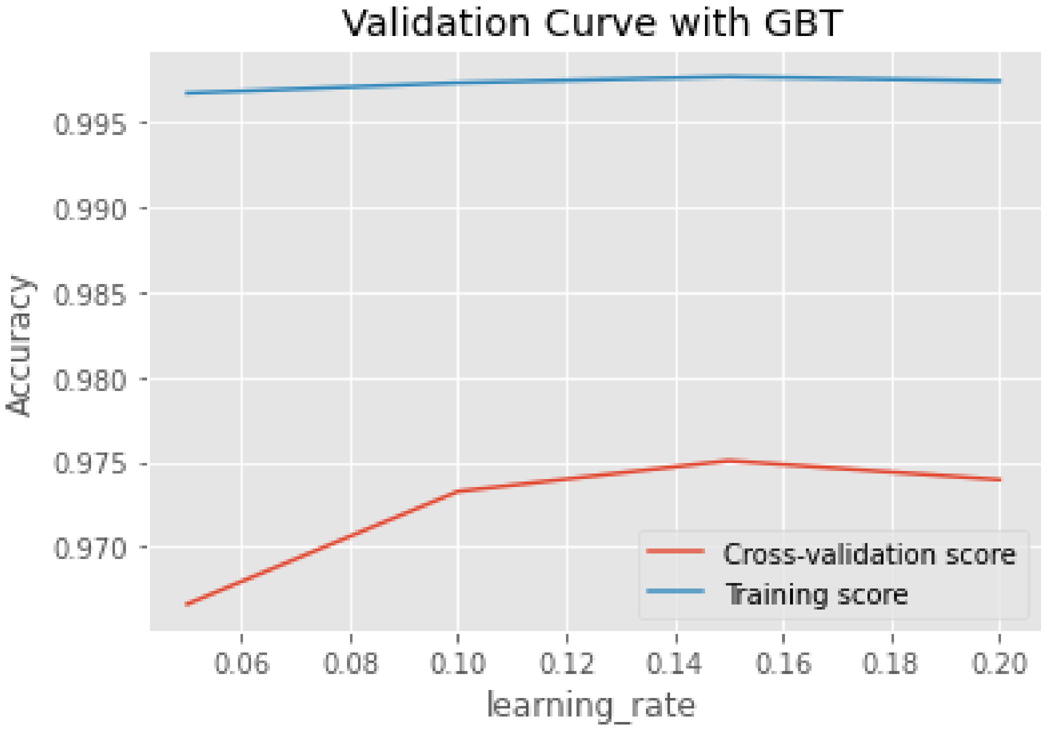


Figure-11 Validation Curve over learning rate (Gradient Boosting)

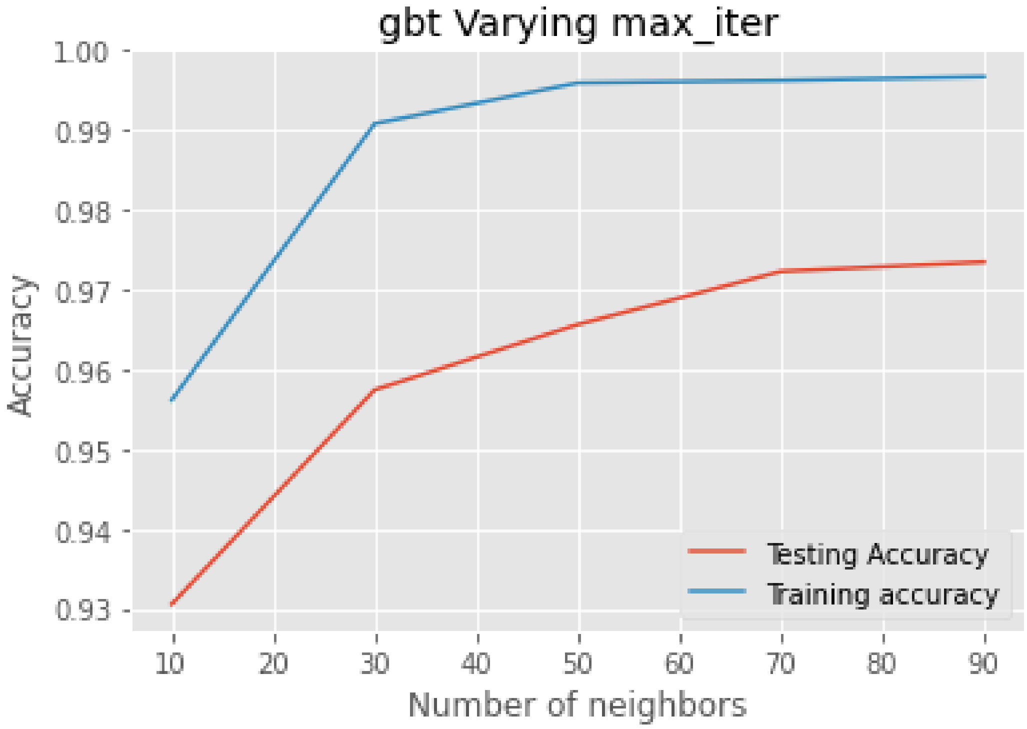


Figure-12 Validation Curve over max iteration (Gradient Boosting)

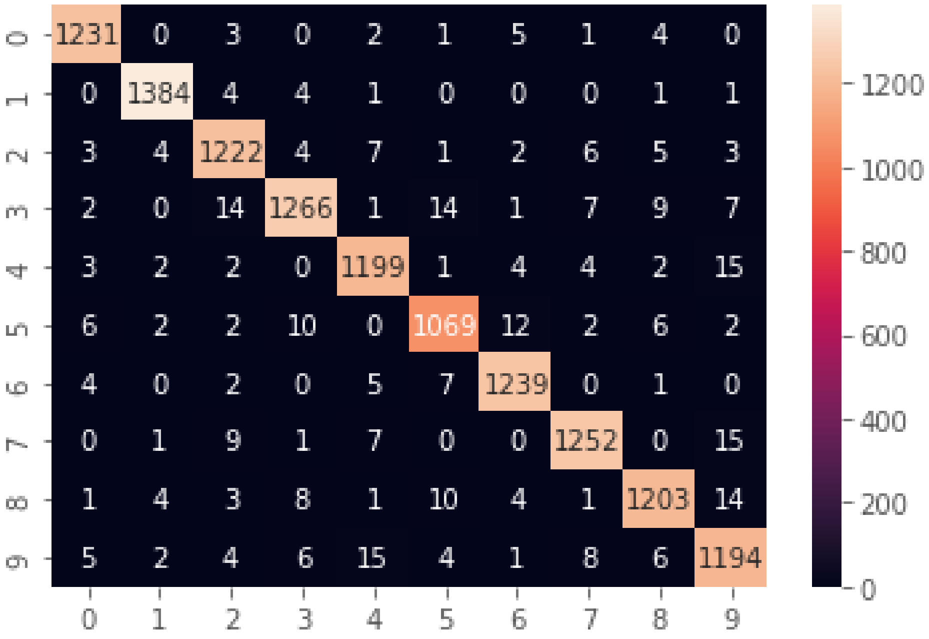
From Fig-11, the test accuracy reached the highest value when learning rate is around

0.15. In the range of 0.13 to 0.17, the differences in accuracy go down, which means overfitting issue is minimized. From Fig-12, these two curves converge as max iteration is greater than 70.

Then tune as much as parameters. The optimal setting is:

$$\{l2\_regularization = 1, learning\_rate = 0.17,$$
$$max\_depth = 6, min\_samples\_leaf = 786,$$
$$\max iteration = 100,\ random\_state = 1\}$$

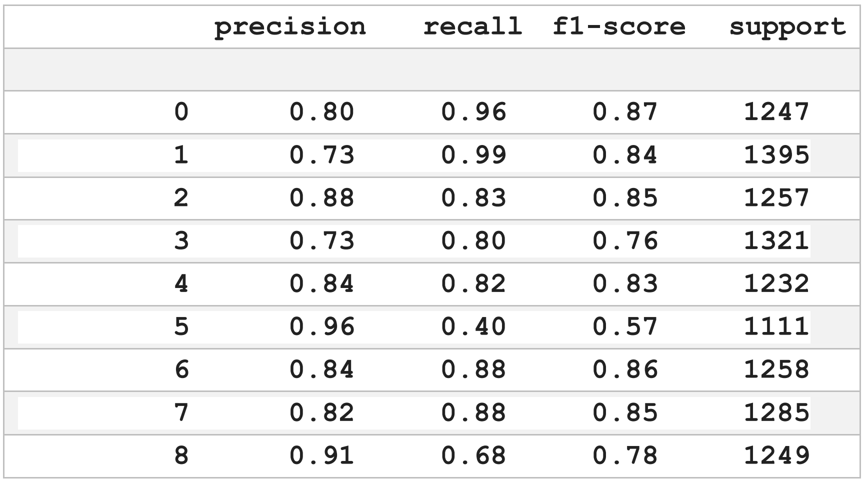
The test accuracy of the above classifier is 0.9729.



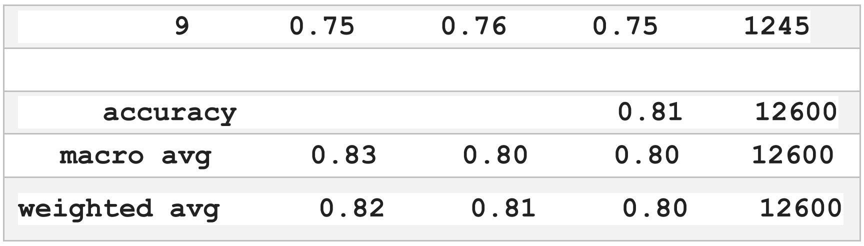
	0	1	2	3	4	5	6	7	8	9
0	1231	0	3	0	2	1	5	1	4	0
1	0	1384	4	4	1	0	0	0	1	1
2	3	4	1222	4	7	1	2	6	5	3
3	2	0	14	1266	1	14	1	7	9	7
4	3	2	2	0	1199	1	4	4	2	15
5	6	2	2	10	0	1069	12	2	6	2
6	4	0	2	0	5	7	1239	0	1	0
7	0	1	9	1	7	0	0	1252	0	15
8	1	4	3	8	1	10	4	1	1203	14
9	5	2	4	6	15	4	1	8	6	1194

Figure-13 Heat map (Gradient Boosting)

Table-4 Performance report(Gradient Boosting)



	precision	recall	f1-score	support
0	0.80	0.96	0.87	1247
1	0.73	0.99	0.84	1395
2	0.88	0.83	0.85	1257
3	0.73	0.80	0.76	1321
4	0.84	0.82	0.83	1232
5	0.96	0.40	0.57	1111
6	0.84	0.88	0.86	1258
7	0.82	0.88	0.85	1285
8	0.91	0.68	0.78	1249



9	0.75	0.76	0.75	1245
accuracy			0.81	12600
macro avg	0.83	0.80	0.80	12600
weighted avg	0.82	0.81	0.80	12600

The confusion matrix shows most categories has great prediction. In the performance report, most F1-score are around 0.80. Samples in label=5 has significant lower F1-score.This method is convincing so far(based on the accuracy rate).

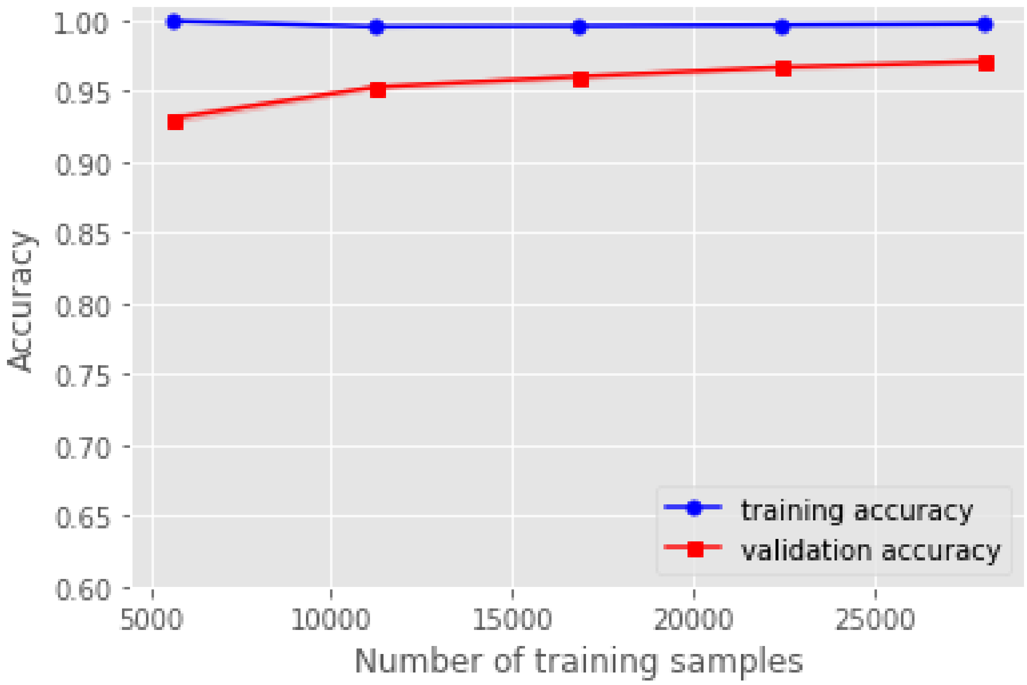


Figure-14 Learning curve (Gradient Boosting)

The learning curve of this classifier shows limited overfitting issue, especially when training sizes is large. With the full training samples, the two learning curves converges and the ultimate difference is about 3%. This Gradient Boosting has the highest accuracy, and limited overfitting issue, which implies it is a reliable prediction method.

Conclusion

GBT test accuracy is 0.9729

RF test accuracy is 0.8077

KNN test accuracy is 0.9364

SVC test accuracy is 0.8739

Above all, Gradient Boosting has the highest test accuracy. It makes sense because this is an ensemble method which learns from previous error. In the end, the final rules lead to barely no misclassification. The accuracy in turn upwards.

K-Nearest Neighbor method has the most significant overfitting issue. This is a reasonable conclusion. Due to its basic idea is distance, this method is greatly influenced by extreme values. Also notice that there are several features has 0 values among all sample. Those features increase the difficulty of computation and unnecessary information to the model.

Random Forest tends to have overfitting issue. Since this report controls the number of trees and max depth for each individual tree, this problem could be ignored. Support vector machine has a decent test accuracy and converged learning curve, but it is too time expensive to tune all desired parameters. In the future, this report recommends Gradient Boosting given the consideration of accuracy, overfitting, and availability on common device.