CSE 291 Pattern Recognition - ASSIGNMENT 3

Ayush Jasuja - A53103338
Nivetha Thiruverahan - A53205226

K Nearest Neighbours

We used the sklearn library's implementation of KNeighborsClassifier to perform the required classification. The underlying algorithm used by the classifier is the ball tree algorithm for efficient indexing and all K nearest neighbours determined based on Euclidean distance are weighted uniformly. The data provided as input to the classifier had their features standardized by removing the mean and scaling to unit variance.
Cross validation experiments were performed to determine the best K - the number of neighbouring points used for classification.

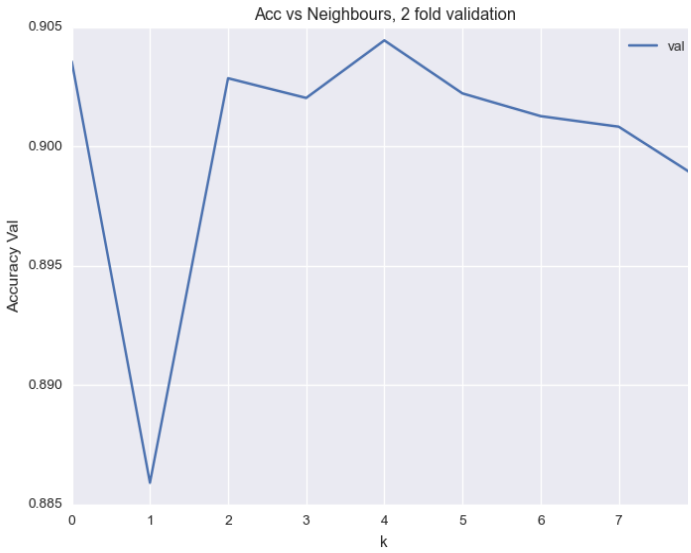
Each dataset was randomly shuffled and 25% of it was allocated as the test set while the remaining was used as the training data. 2-fold, 5-fold and leave one out cross validations were performed on d% subsets of the training data for d in {50,75,100}

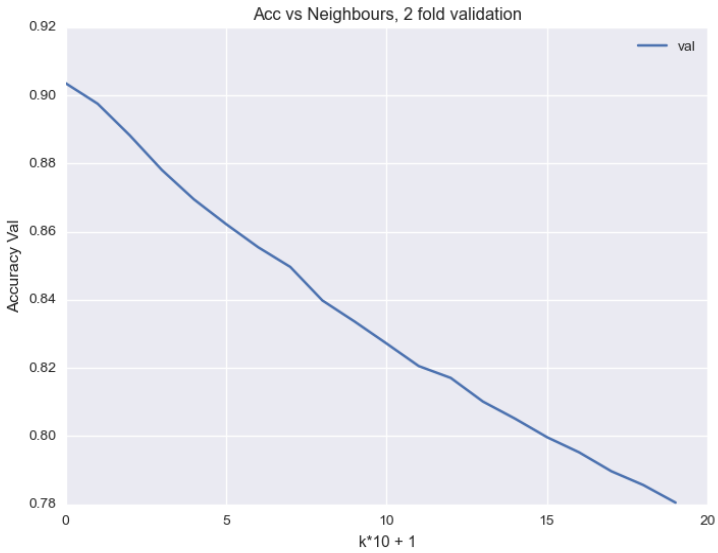
MNIST data :

We tried several values of k from 1 to number of data points-1 initially and witnessed a downward trend in CV accuracy right after just a few neighbours (can be seen in the 2 fold CV graphs shown below, for example) and thus all the following graphs are for a limited range where the maximum accuracy was observed. Note : X axis is k-1 rather than k.

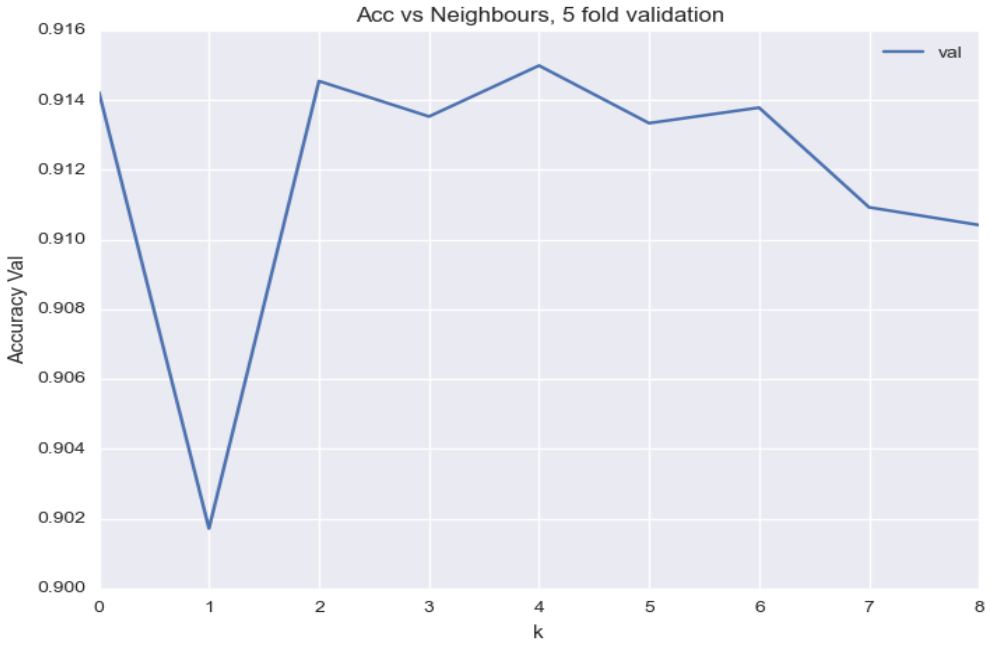
- 50% data:

I. 2-fold cross validation :

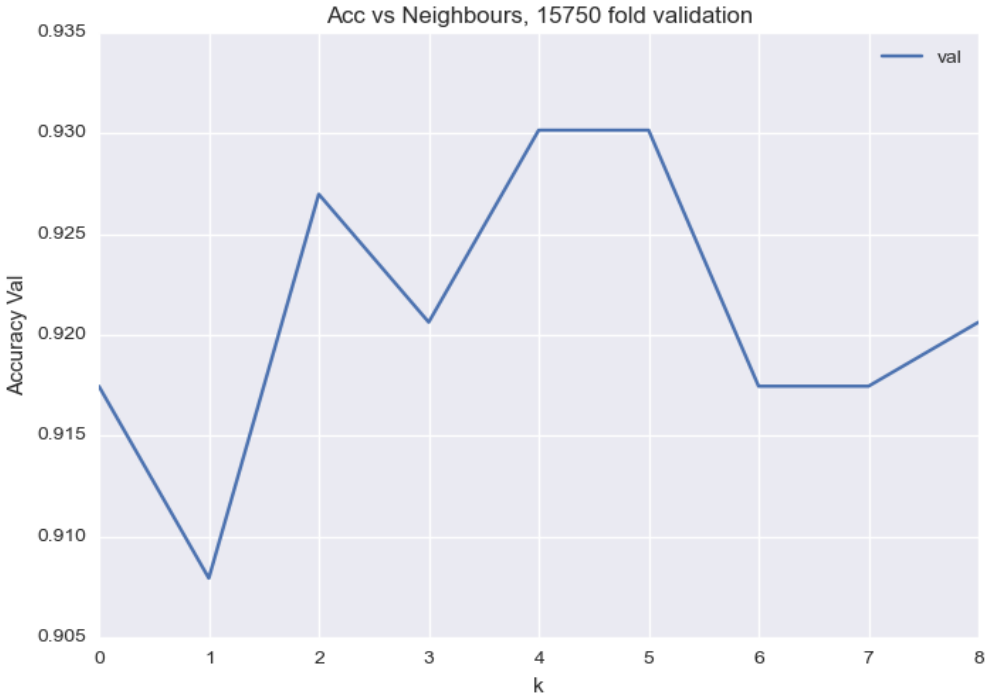




II. 5-fold cross validation :

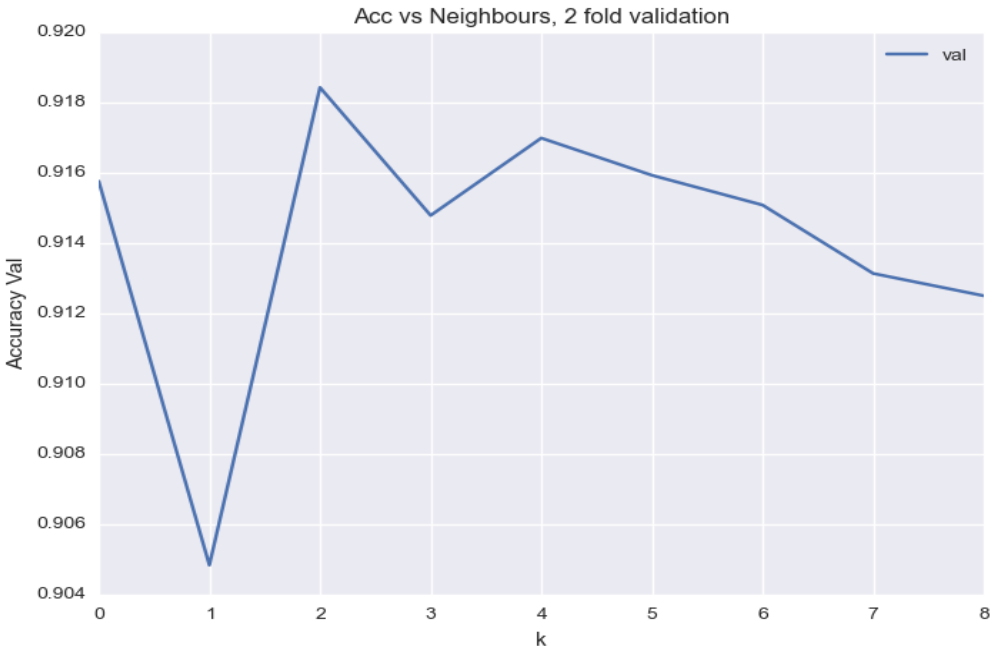


III. Leave one out cross validation :



- 75% data:

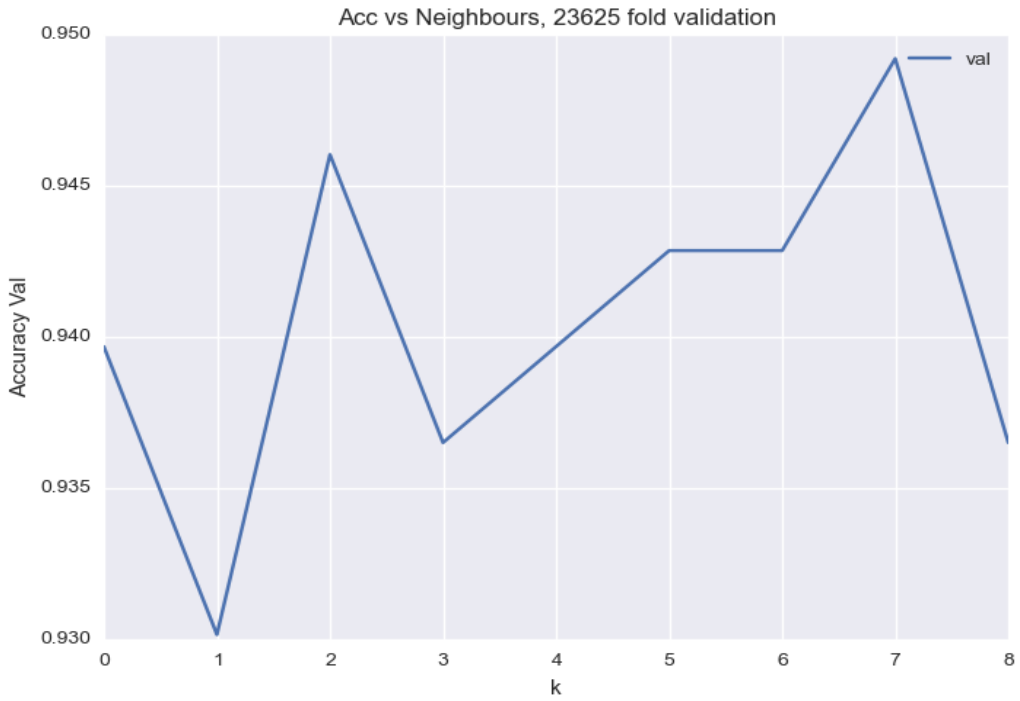
I. 2-fold cross validation :



II. 5-fold cross validation : *

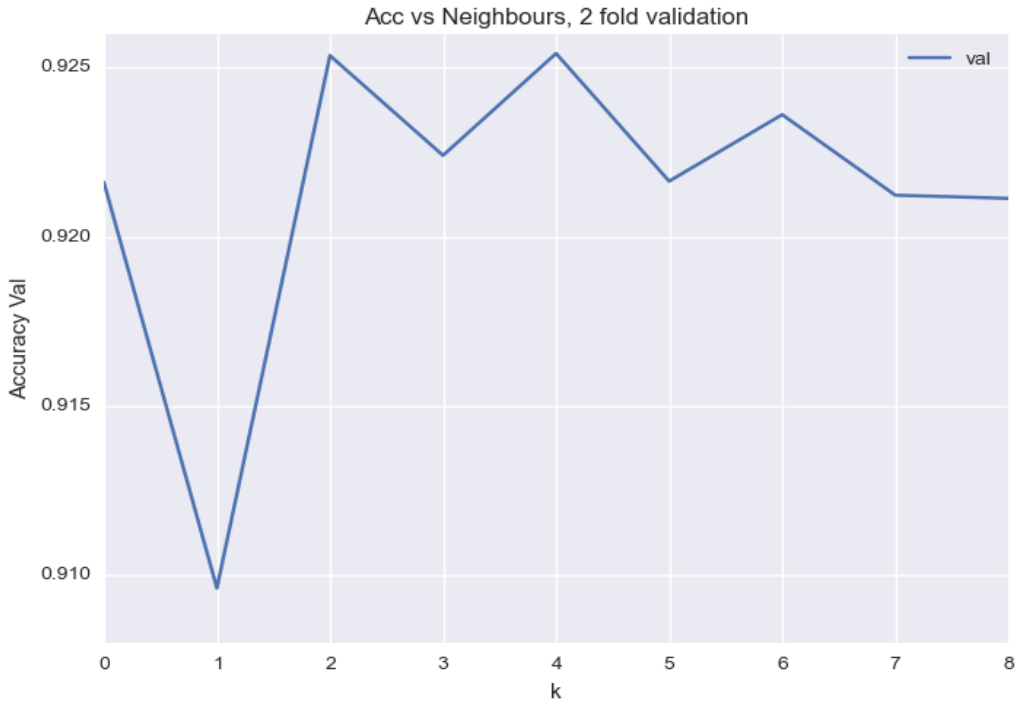
*A previous result was also plotted erroneously. Consider the green line.

III. Leave one out cross validation :

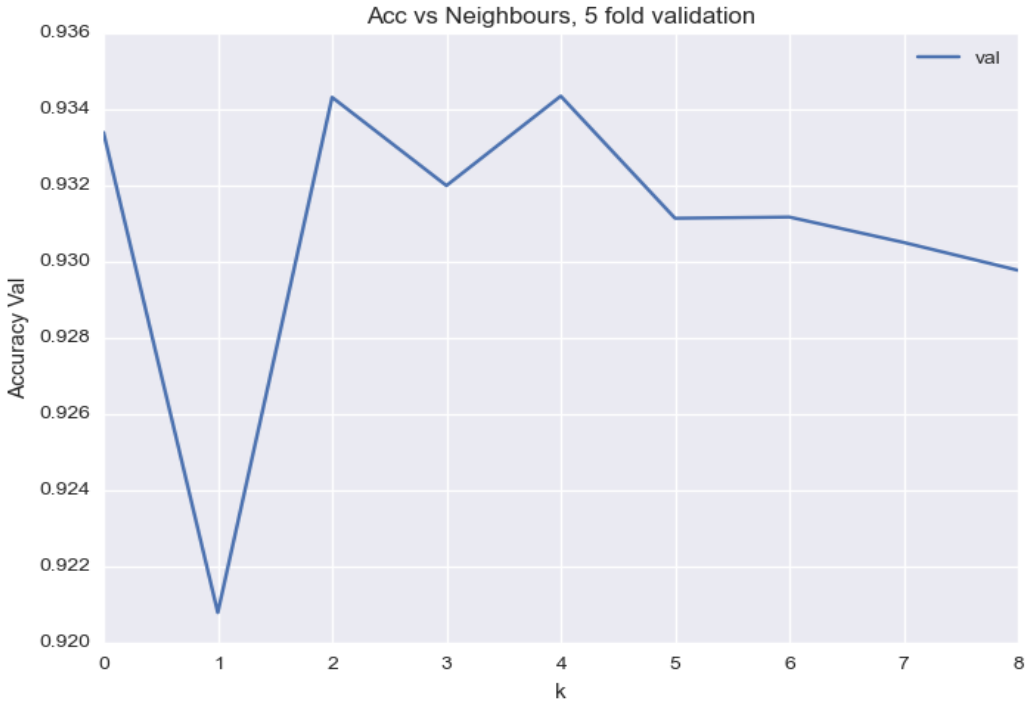


- <u>100% data:</u>

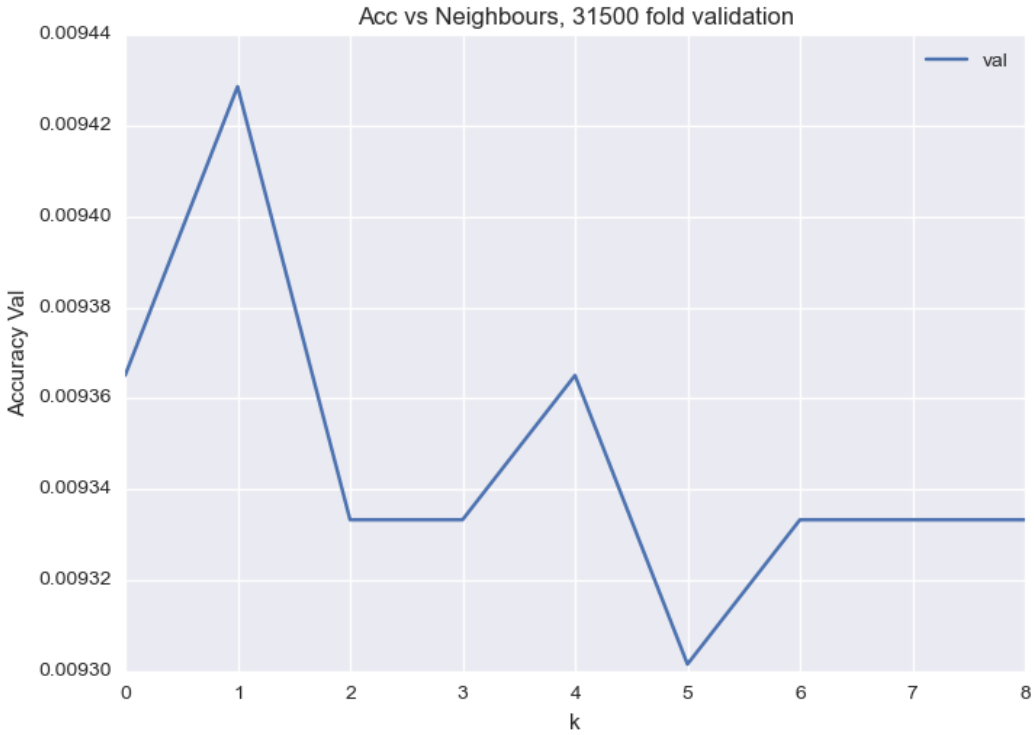
I. 2-fold cross validation :



II. 5-fold cross validation :



III. Leave one out cross validation :



Consider Y axis as accuracy/100. (There was a bug in the program that rendered values in this graph divided by 100 and due to time constraints we couldn't generate the graph again.)

Table I : MNIST data - CV errors for the best K :

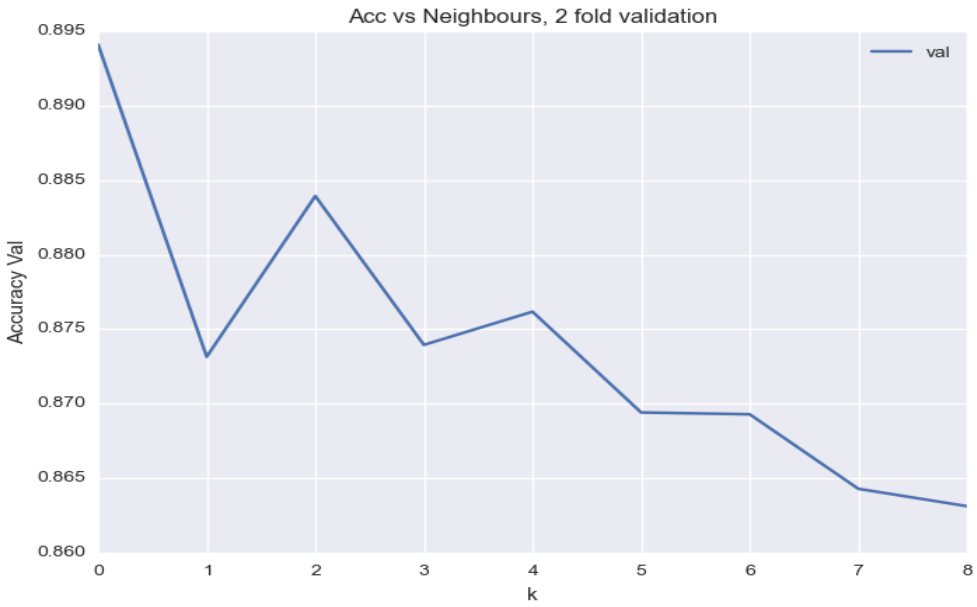
		Best K	CV Error
50% data			
	2-fold	5	9.54%
	5-fold	5	8.50%
	Leave one out	5	7.00%
75% data			
	2-fold	3	8.15%
	5-fold	3	7.20%
	Leave one out	8	5.20%
100% data			
	2-fold	3	7.48%
	5-fold	3	6.59%
	Leave one out	2	5.70%

Covertype data :

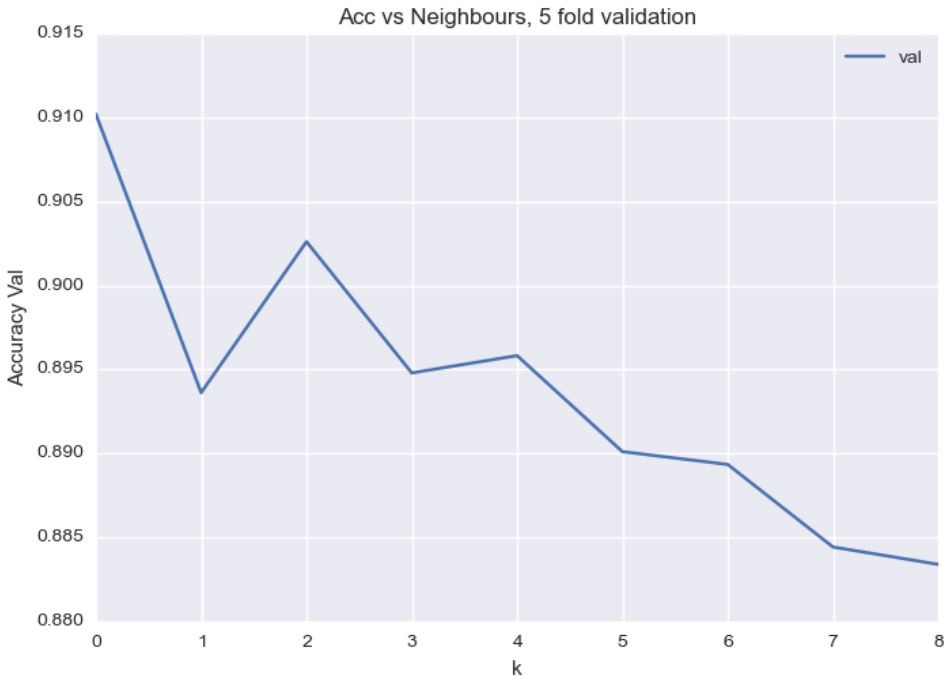
Again, we tried several values of k from 1 to number of data points-1 initially and witnessed a downward trend in CV accuracy right after just a few neighbours and thus all the following graphs are for a limited range where the maximum accuracy was observed. Note : X axis is k-1 rather than k. Due to the size of the dataset, leave one out experiments average CV error across every 100th point instead if every point, that is every 100th point acts as the validation set.

- 50% data:

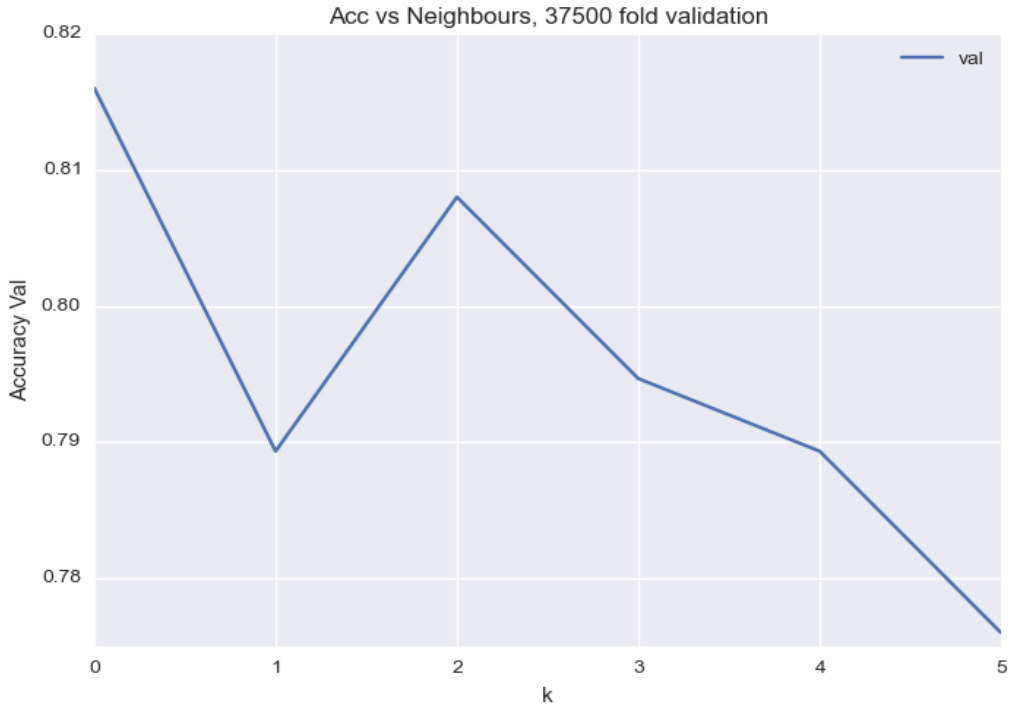
I. 2-fold cross validation :



II. 5-fold cross validation :



III. Leave one out cross validation :

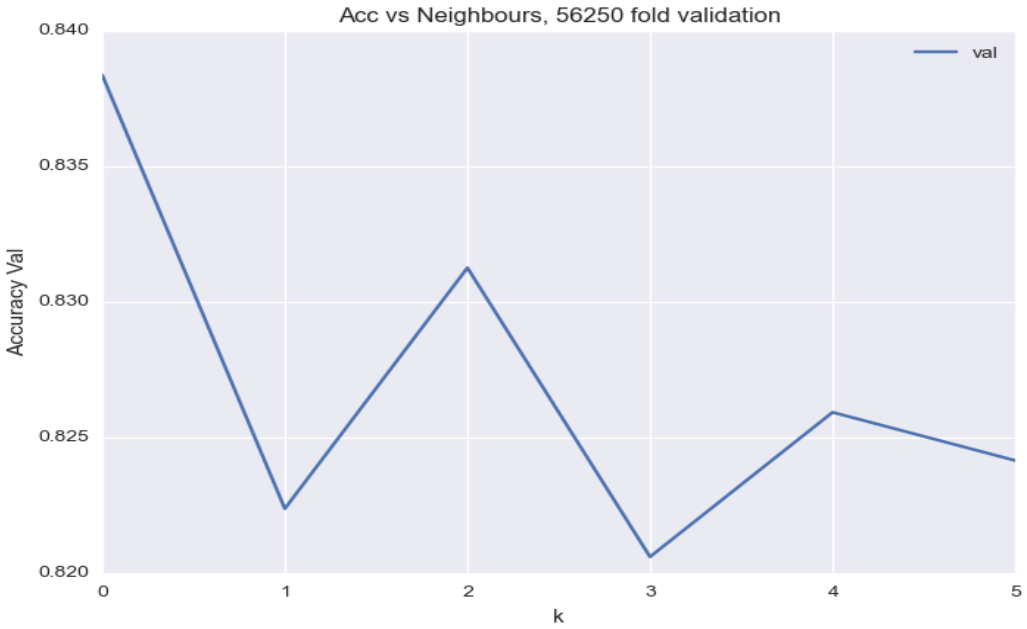


- 75% data:

I. 2-fold cross validation :

II. 5-fold cross validation :

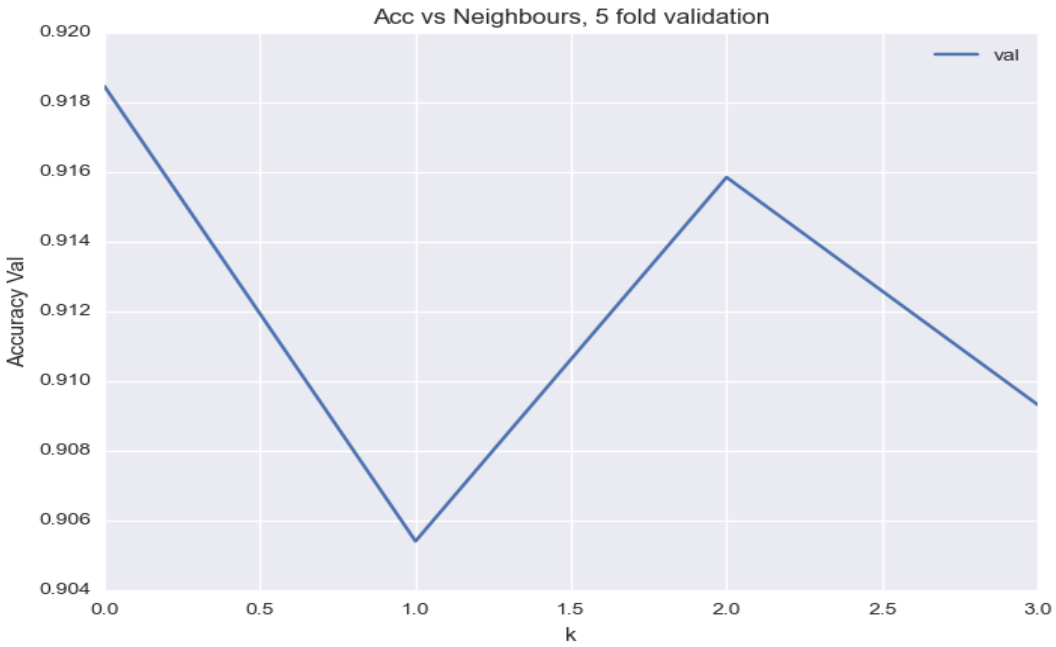
III. Leave one out cross validation :



- <u>100% data:</u>

I. 2-fold cross validation :

II. 5-fold cross validation :



III. Leave one out cross validation :

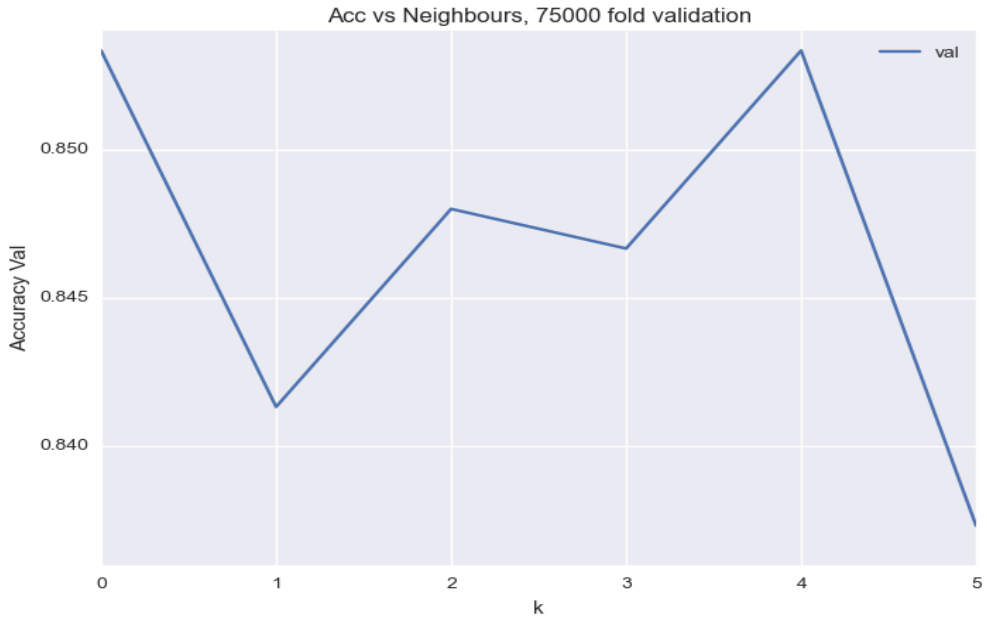


Table II : Covertype data - CV errors for the best K :

		Best K	CV Error
50% data			
	2-fold	1	10.6%
	5-fold	1	9.0%
	Leave one out	1	18.43%
75% data			
	2-fold	1	9.2%
	5-fold	1	7.8%
	Leave one out	1	16.25%
100% data			
	2-fold	1	9.5%
	5-fold	1	8.15%
	Leave one out	5	14.6%

Test Errors :

The test errors corresponding to each subset of training data and the associated best K are given in the following table along with their difference with the corresponding CV error. The cross validation fold that estimated the test error best, that is the one with the lowest difference, has been highlighted.

Table III - MNIST data :

		Best K	Test Error	CV Error	Difference
50% data					
	2-fold	5	7.981%	9.54%	1.559%
	5-fold			8.50%	0.519%
	Leave one out			7.00%	0.981%
75% data					
	2-fold	3	6.962%	8.15%	1.188%
	5-fold			7.20%	0.238%
	Leave one out	8	7.438%	5.20%	2.238%
100% data					
	2-fold	3	6.257%	7.48%	1.223%
	5-fold			6.59%	0.333%
	Leave one out	2	7.772%	5.70%	2.072%

Table IV - Covertype data :

		Best K	Test error	CV Error	Difference
50% data					
	2-fold	1	5.528%	10.6%	5.072%
	5-fold			9.0%	3.472%
	Leave one out			18.43%	12.902%
75% data					
	2-fold	1	5.756%	9.2%	3.444%
	5-fold			7.8%	2.044%
	Leave one out			16.25%	10.494%
100% data					
	2-fold	1	5.922%	9.5%	3.578%
	5-fold			8.15%	2.228%
	Leave one out	5	5.905%	14.6%	8.695%

It can be seen that 5 fold cross validation has estimated the test error best irrespective of the amount of data and the dataset used in this case. One can think of 5 fold cross validation as something in between the extremes of the minimum number folds - 2 folds and maximum number of folds - leave one out, and is therefore in a sense a more reasonable way of performing cross validation tests. This is probably why it is able to match the test error closer than the other two extremes.

Stability of Results :

MNIST Data :

From Table I results we can see that the best K is the same when considering 75% and 100% data to 3, at least when considering the K obtained out of 2-fold and 5-fold cross validation. experiments. Also in the case of leave one out cross validation, when considering 75% data, it can be seen from the graph that the difference in accuracy between considering 3 neighbours and 8 neighbours is minimal of the order of 0.4% rendering 3 neighbours as good enough in the face of increasing computation for minimal performance improvement. The leave one out case in 100% data has a difference of only 1% between K=3 and K=2. In contrast, the best K result from 50% data is different from that of 100% data.
In Table III, the difference in test errors between 75% data and 100% data is < 1% and the difference between CV error and test error across the folds is closely mirrored from 75% data to 100% data with very minute difference, which is not the case for 50% data that has > 1% difference.
Thus it can be said that stability of result - K and the test error, and thereby stability of the model has been reached at 75% data.
So, in light of the humongous computation involved that is not linear in datapoints, it would be enough to consider 75% data to train the model to obtain a stable model.

Covertype Data :

From Table II results we can see that the best K is 1 when considering each of 50%,75% and 100% data and across the different cross validation experiments except the leave one out experiment considering 100% data. But it can be seen that the difference in accuracy between 1 neighbour and 5 neighbours is minimal of the order of 0.1%.
From Table IV, it can be seen that the test errors across the models with 50%,75% and 100% data don't vary too much with difference in the order of 0.2%. The small scale of accuracy differences witnessed in MNIST data is not observed here. Nevertheless going by the values here, one can see the difference in test error compared to the CV error across the folds using 75% data and 100% data are much closer than 50% data and 100% data.
So, in terms of model performance, training with merely 50% of randomly shuffled data will result in the same K value and performance as training with 100% data and thus can be termed as a 'stable' model because the test error rendered will be the same. If we are looking at stability in terms of how well the cross validation experiments are able to predict the test error, a stable model is reached at 75% data and it will be unnecessary to spend additional computation power training on 100% data.

Parameter choices :

Distance metric :

The metric used to find the K nearest neighbours that will influence the class of a data point is the distance between the point of consideration and other points in KNN. There are many different distance metrics though, that determine "nearness". Euclidean distance is a commonly used metric , however, has the disadvantage that it assumes each feature equally and can be affected by non-standardised data. Mahalanobis distance on the other hand automatically accounts for the scaling and corrects for correlation between the different features. But the covariance matrices used in Mahalanobis distance calculation can be hard to determine accurately, and the memory and time requirements grow quadratically rather than linearly with the number of features. Thus we decided, given the number of features in our dataset, to normalise the data features and apply Euclidean distance metric for determining the nearest neighbours, which seems a reasonable alternative to using Mahalanobis distance.

Choice of K :

Cross validation experiments are used to determine the optimal number of neighbours. In F fold cross validation, the training data is divided into almost equal sized F sets. F-1 sets are used for training the model and validation is done on the remaining one set. This is done F times, that is each set gets a chance to be the validation set. Leave one out is the case of N fold where N is the number of data points. The average CV accuracy/error is determined. The experiment is done for different values of K and the best K is obtained by graphing the average cross validation accuracy/ error against K and looking for the K that gives the highest/lowest value respectively.

Bias-variance tradeoff :

In general, the tradeoff between bias and variance is a compromise between under-fitting and over-fitting. Complex models capture variance, have low bias and over fit whereas simple models are highly biased with less variance resulting in an under fit. There is usually a sweet spot to balance variance and bias.

In KNN, increasing k results in more points influencing a decision, as K increases to a large value, the distinction between points blur with almost points considered as neighbours to all points reducing the variance and increasing bias. Whereas on decreasing K, the prediction is based only on few neighbours resulting in more variance being captured and less bias.

Let's take a look at the CV error trend graphs. For the MNIST data, it can be seen that decreasing K below 3, results in a drop in accuracy as high variance kicks in and increasing K beyond 3 results in a drop in accuracy as high bias kicks in. The sweet spot of tradeoff between bias and variance occurs at K=3.

It's difficult to note this two way trend in case of Covertype data since the optimal K is 1 and we can look at the case of decreasing K. However, accuracy drops on increasing K as bias increases.

SVM

We used the sklearn library's implementation of SVC - Support Vector Classifier in which multi class support is handled via one-vs-one scheme and the type of kernel, rbf (radial basis function) , linear, polynomial or sigmoid is passed as a parameter. Optimal hyper parameters, C - penalty for the error term and gamma - kernel coefficient for each of the classifiers is determined through sklearn's grid search function.

Due to the extensive size of Covertype data and the large computation times of SVM, we consider a randomly shuffled 50% of the dataset as our whole dataset to perform the experiments. Each dataset was randomly shuffled and 25% of it was allocated as the test set while the remaining was used as the training data. 2-fold and 5-fold cross validations were performed on d% subsets of the training data for d in {50,75,100} and the four type of kernels to determine the best kernel.

The methodology of finding best parameters is outlined below :

1. Take a subset of data for training

2. Apply PCA and reduce the feature space to 15 features (mean reconstruction error: 0.00954496859281).

3. For parter estimation, since running grid search with SVC on the complete training dataset takes a lot of time, we decided to divide the dataset into multiple smaller subsets (each subset of size 5000).

4. We find best parameters for different types of kernels on these smaller subsets and report the mean of those parameters as the final solution for the complete dataset.

5. When testing, we make multiple predictions about a data point based on training over multiple smaller subsets and report the mode classification as the final class of that point.

Note: For MNIST we are not following this methodology as learning parameters in that dataset takes a lot lesser time. But we do compress the feature space to 30 dimensions based on our previous assignments.

MNIST data : The best kernel in terms of average CV error and corresponding average CV error is reported for each case.

- 50% data:

I. 2-fold cross validation

RBF	2.86%

II. 5-fold cross validation :

RBF	2.48%

- 75% data:

I. 2-fold cross validation

RBF	2.54%

II. 5-fold cross validation :

RBF	2.01%

- 100% data:

I. 2-fold cross validation

RBF	2.19%

II. 5-fold cross validation :

RBF	1.88%

It was observed that the RBF kernel outperformed all the remaining kernels in all cases.

Covertype data :

- 50% data:

I. 2-fold cross validation

RBF	40.68%

II. 5-fold cross validation :

RBF	38.37%

- 75% data:

I. 2-fold cross validation

Linear	32.28%

II. 5-fold cross validation :

Linear	30.19%

- 100% data:

I. 2-fold cross validation

Linear	32.25%

II. 5-fold cross validation :

Linear	31.56%

It was noted that the RBF kernel performed best for 50% data and the Linear kernel performed better for 75% and 100% data in case of Covertype.

Test Errors :

The test errors corresponding to each subset of training data and the associated best K are given in the following table along with their difference with the corresponding CV error. The cross validation fold that estimated the test error best, that is the one with the lowest difference, has been highlighted.

Table V - MNIST data :

		Best Kernel	Test Error	CV Error	Difference
50% data					
	2-fold	rbf	2.12%	2.86%	0.74%
	5-fold			2.48%	0.36%
75% data					
	2-fold	rbf	1.75%	2.54%	0.79%
	5-fold			2.01%	0.26%
100% data					
	2-fold	rbf	1.55%	2.19%	0.64%
	5-fold			1.88%	0.33%

Table VI - Covertype data :

		Best Kernel	Test Error	CV Error	Difference
50% data					
	2-fold	rbf	24.74%	40.68%	15.94%
	5-fold			38.37%	13.63%
75% data					
	2-fold	linear	29.19%	32.28%	3.09%
	5-fold			30.19%	1.00%
100% data					
	2-fold	linear	29.73%	32.25%	2.52%
	5-fold			31.56%	1.83%

It can_be seen that 5 fold cross validation has estimated the test error best irrespective of the amount of data and the dataset used in this case as well.

Stability of Results :

MNIST Data :

From Table V, it can be seen that the best kernel is determined as rbf kernel right from 50% data and the test error has stabilised from 50% data, as well, since the difference in test error when considering 50% and 100% data is < 1%. The difference in CV error and test error has also remained almost the same. Thus sufficiently stable results can be obtained with an SVM model built with 50% of MNIST data to get almost equivalent results to considering 100% data and maybe 75% of the data for even more accurate results close to the 100% data case.

Covertype Data :

From Table VI, it can be seen that the best kernel determined by 75% data matches with the one determined by 100% data. Also note that the test errors differ by only around 5% between the two and the difference in the CV error estimate and test error is < 2% if 5-fold cross validation is considered. The same is not true when comparing 50% data results and 100% data results wherein stark differences are present in terms of kernel and error estimates. Thus, stable results can be obtained by training a model with 75% of randomly shuffled data to obtain results similar to having used 100% data.

Parameter choices :

C-penalty parameter and Gamma- kernel coefficient:

As mentioned, the methodology in the introduction is used to determine the best kernel parameters C and gamma for each type of kernel for each subset of the data, using the sklearn gridsearch implementation's default search range. The best kernel parameters are as follows :

MNIST data	Kernel	C	Gamma
50% data			
	Linear	1E-09	0.01
	Poly	6.85E-06	0.041
	RBF	8.65E-07	9.55
	Sigmoid	6.4E-08	86.05
75% data			
	Linear	1E-09	0.01
	Poly	7.6E-06	0.064
	RBF	9.1E-07	9.4
	Sigmoid	5.44E-08	174.4

MNIST data	Kernel	C	Gamma
100% data			
	Linear	1E—09	0.01
	Poly	7.943E-06	0.0563
	RBF	9.228E-07	6.657
	Sigmoid	5.886E-08	102.314

Covertype data	Kernel	C	Gamma
50% data			
	Linear	1E-09	7.719E-06
	Poly	1E-05	1E-05
	RBF	1.194E-07	571646.31
	Sigmoid	1E-09	1E-09
75% data			
	Linear	1E-09	0.0262
	Poly	1E-05	0.064
	RBF	2.590E-07	0.0557
	Sigmoid	8.674E-07	0.0361
100% data			
	Linear	1E—09	0.0024
	Poly	1E-05	0.0611
	RBF	1E-09	1E-06
	Sigmoid	6.1035E-07	0.0391

Best kernel :

Cross validation experiments are used to determine the best kernel. As mentioned before, in F fold cross validation, the training data is divided into almost equal sized F sets. F-1 sets are used for training the model and validation is done on the remaining one set. This is done F times wherein each set gets a chance to be the validation set. Leave one out is the case of N fold where N is the number of data points. The average CV accuracy/error is determined. The experiment is done for the 4 different kernels and the best kernel is obtained by comparing the average cross validation accuracy/error for the one that gives the highest/lowest value respectively.

Bias-variance tradeoff :

As mentioned before, the tradeoff between bias and variance is a compromise between under-fitting and over-fitting. Complex models capture variance, have low bias and over fit whereas simple models are highly biased with less variance resulting in an under fit. There is usually a sweet spot to balance variance and bias.

In the RBF kernel, C is the cost of misclassification. Large values of C penalise misclassifications a lot and hence have low bias and high variance leading to overfitting, whereas small values of C have high bias and low variance.

The Gaussian radial basis function is given by

$$K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2), \gamma > 0$$

where Gamma is a free parameter used to handle non linear classification.
Intuitively, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. In other words, a small value of gamma implies a decision boundary whose curve is not much resulting in a broad decision region leading to low variance and high bias and a large value of gamma implies very curved decision boundaries which overfits the data leading to high variance and low bias.

There will exist a sweet spot that balances bias and variance in terms of both parameters C and gamma and that is determined by grid search as discussed before.

Comparison of KNN and SVM :

In terms of performance, it is apparent from tables III, IV, V and VI, that SVM (RBF kernel) performs better for the MNIST data classification task and KNN performs much better, in fact really well for the Covertype data classification task. It makes intuitive sense for Covertype, since the covertype of a tree will be similar to that of the neighbours and that will be an excellent way to model the dataset.

KNN has a better learning time than SVM. SVM training is particularly hard because finding a good boundary to separate different classes across different types of kernels is difficult for multiple parameter values (C and gamma). Since KNN is completely dependent on the nearest neighbor classes, that is non parametric , its learning rate is better.

KNN requires all the training points to be stored in memory for classification whereas the memory requirements in the case of SVM is not so much. However, efficient storage and indexing in KNN is done with the help of algorithms like Ball tree and KD tree.