Analysis

Jitae Kim

Data Set Introduction

Two data sets are used to implement five learning algorithms: Car Evaluation dataset and Breast Cancer dataset.

- **Car Evaluation Data Set:**

I obtained Car Evaluation dataset from UCI Machine Learning Repository (URL: https://archive.ics.uci.edu/ml/datasets/car+evaluation). It evaluates the target concept(CAR) with 3 other intermediate concepts. PRICE (overall price), TECH (technical characteristic) and COMFORT (comfort). It has six attributes and 1728 instances. I found this data set interesting because it showed how the prices for used cars are calculated. Also, I chose this data because it was very easy to understand what each attribute was.

Table 1: Attribute values

Buying	{v-high, high, med, low}
Maint	{v-high, high, med, low}
Doors	{2,3,4,5- more}
Persons	{2,4, more}
Log_boot	{small, med, big}
Safety	{low, med, high}

Table 2: Class Distribution

Unacceptable	1210
Acceptable	384
Good	69
V-good	65

The file uploaded on that website was text file format, and so I have modified data and saved it in excel file. Also, some attributes are in string such as Buying, Maint, Log_boot, and Safety has value as string. So, I replace all the string values with integers in ascending order. I believe that those data pre-processing makes coding or algorithms easier.

- **Breast Cancer Data Set:**

This data can be obtained from Kaggle (link: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data). This data set is interesting to me because one of my relatives has breast cancer and Breast cancer is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today's society. This data set consists of 32 attributes which include diagnosis results where M stands for malignant and B stands for benign. It has 569 rows where 212 are malign and 357 are benign.

Before using data set, I checked if there are any attributes that are irrelevant to the result, duplicated data, or missing data (using dataset.isnull().sum(), dataset.isna().sum()). Also, I replaced the string values, which are M and B, M for 1 and B for 0 and ignored diagnosis id column because it really does not affect the results.

I picked those data sets because their classifications are distinct and they have different number of attributes. The Breast Cancer Data set is a binary classification and Car Evaluation Data set is a multiclass classification. Comparison between experiments performed on Car Evaluation dataset and Breast Cancer dataset can demonstrate how algorithms behave differently on different types of dataset. Also, the difference in size can also demonstrate how size can influence the performance of classifiers. My initial guess for performance of algorithms was that Car Evaluation Data set has less accuracy than Breast Cancel data set because it has fewer attributes but is a multiclass classification.

- **Data Preprocessing**

As I already mentioned about converting some string values to integers and dropped unnecessary columns, data has been normalized for multi-layer perceptron of Neural Network because multi-layer perceptron is sensitive to feature scaling and Neural Network in Python may have difficulty converging before the maximum number of iterations allowed if the data is not normalized.

Implementation

I implemented five learning algorithms. They are Decision trees with/without pruning, Neural networks, Boosting, Support Vector Machines, and KNN using two data sets. In my code, scikit-learn library was used to implement those five algorithms and Label Encoder to label the categorical data. I randomly split data sets into train sets and test sets in 7:3 ratios (7:3 or 8:2 is general it's because it would fail to generalize our prediction model if a set of train data is relatively small compared to a set of test data). The accuracy score is calculated by testing the algorithm on the test set. I also used k-fold validation with fold size was 5 because it gives better estimator of out-of-sample accuracy and more efficient use of data since each data is used for train and testing. The curves in the graph show the average of five tests.

- **Decision Tree**

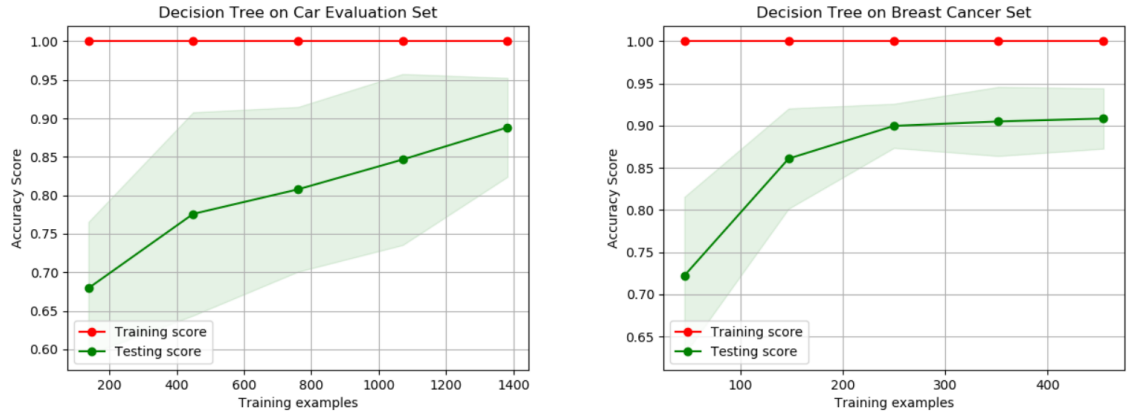

Figure 1.1, 1.2 The graphs show the training and testing curves for the Decision Tree without pruning on Car Evaluation Data set(left) and Breast Cancer Data set(right).

The first algorithm I used was Decision Tree without pruning. It shows that accuracy score for Car Evaluation Data set is **0.893** and for Breast Cancer Data set is **0.903**. There was no significant difference for accuracy score between two data sets. However, it's obvious that there is overfitting issue for both data sets. To solve this issue, we can adopt pruning, which is removing the branches that make use of features having low importance and so reduce the complexity of tree, and thus increasing its predictive power by reducing overfitting. Pruning can start at either root or the leave. I used "reduced error pruning, which starts at leaves and removes each node with most popular class in that leaf, this change is kept if it doesn't deteriorate accuracy.

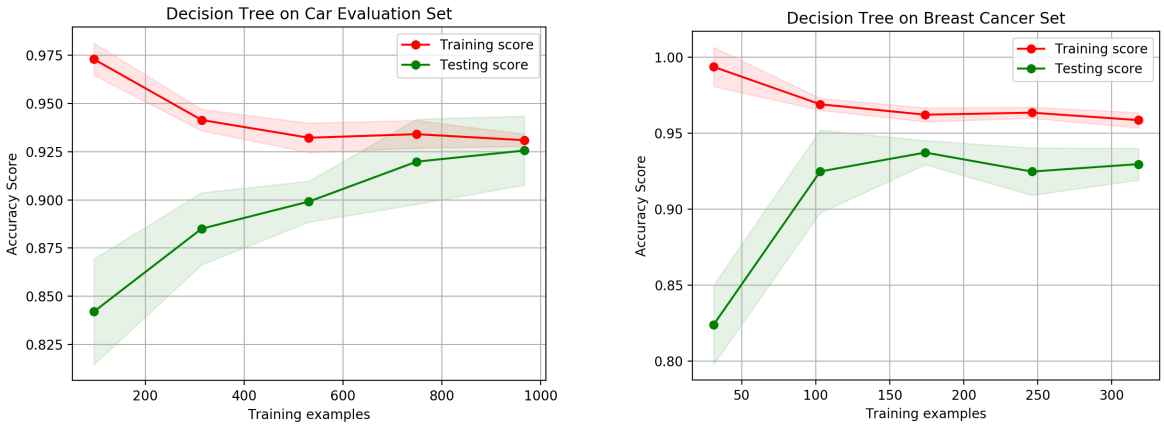

Figure 1.3, 1.4 The graphs show the training and testing curves for the Decision Tree with pruning on Car Evaluation Data set(left) and Breast Cancer Data set(right).

While I was applying pruning method, I also did some experiments with the maximum depth of the tree and threshold. While I was doing this experiments with trial and error, I found that the accuracy largely dropped when threshold got larger. I was kinds of struggling to maximize its accuracy without overfitting because its overfitting when it has small threshold or accuracy

became lower when it has higher threshold. The highest accuracy I got for the Car Evaluation Data set was **0.925** with depth of tree = 7 and threshold = 3. I achieved **0.941** accuracy for the Breast Cancer Data set with depth of tree = 2 and threshold = 20. As you see the graph, you may notice that accuracy for both data sets has increased and pruning solved issue for overfitting. Also, I can conclude that threshold has huge effect on overfitting the decision tree as it goes to small value.

- **Neural Network**

Instead implementing simple Neural Network, I used multi-layer perceptron model of a neural network by adding layers of perceptron because it produces mid-network are less linear with respect to the inputs and more linear with respect to the output for complex problems as well as we can get a smoother transition from the domain to the codomain, improving generalization and reducing overfitting. In this learning algorithm, I was doing some experiments by changing the number of units in hidden layers from 1 to 40 and picked the number of hidden layers with the highest accuracy. Also, I checked how much Neural Network is sensitive to data-normalization.

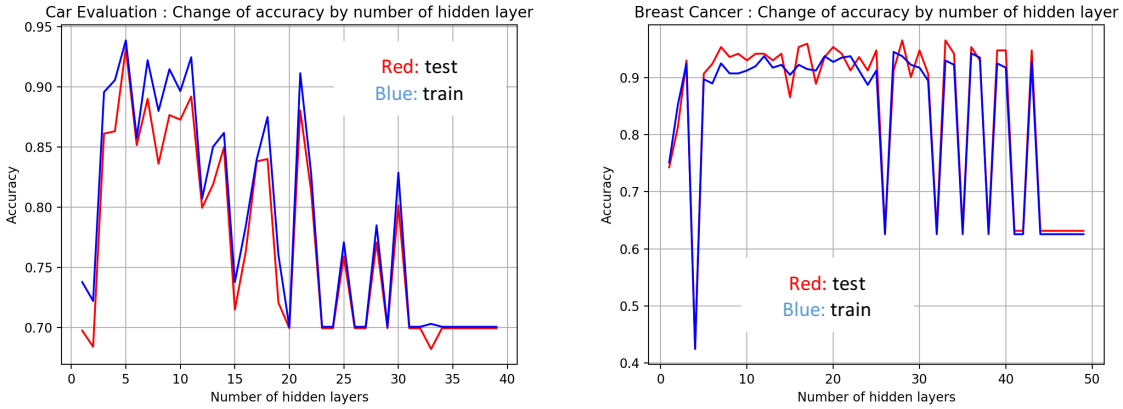

Figure 2.1, 2.2 The graphs show how changing number of units in hidden layer on Car Evaluation Data set(left) and Breast Cancer Data set(right) affect testing and training accuracy without data-normalization.

The above graph shows that how varying number of units in hidden layer impacts the testing and training accuracy. I achieved **0.933** accuracy for the Car Evaluation Data set from 5 units in the hidden layer and **0.937** accuracy for the Breast Cancer Data set from 27 units in the hidden layer. You may observe that there are some fluctuations after 15 units in the hidden layer for Car Evaluation Data set and 25 units in the hidden layer for Breast Cancer Data set.

To overcome those fluctuations and increase accuracy for Neural Network, I applied data normalization before training data. I got **0.956** accuracy for Car Evaluation Data set and **0.988** accuracy for Breast Cancer Data set with data-normalization. I achieved the highest accuracy for both data sets when the number of hidden layers was less than 10 and the number of units in each layer was 15 after testing if the result was consistent based on the number of units in the hidden layer. It turned out that less units in each layer and three additional layers, the accuracy was increased. Thus, we can conclude that number of hidden layer and data-normalization have influence on the accuracy and the Breast Cancer Data set is more complex than Can Evaluation

Data set because the complex problems need more layers as I explained above and the Breast Cancer Data set requires more hidden layers to achieve the best accuracy.

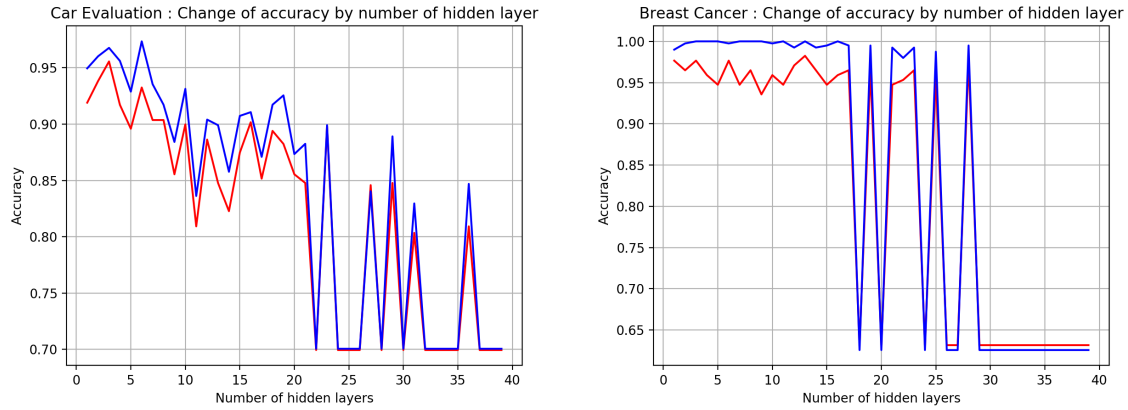

Figure 2.3 2.4 The graphs show how changing number of units in hidden layer on Car Evaluation Data set(left) and Breast Cancer Data set(right) affect testing and training accuracy with data-normalization.

- **Boosting**

In this learning algorithm, I applied AdaBoost classfier because it is ensemble, which combines several decision trees to produce better predictive performance than utilizing a single decision tree. So, I thought it would work better than Decision Tree I implemented as the first learning algorithm and give best result for the Car Evaluation Data set because AdaBoost is best used to boost the performance of decision trees on binary classification problems.

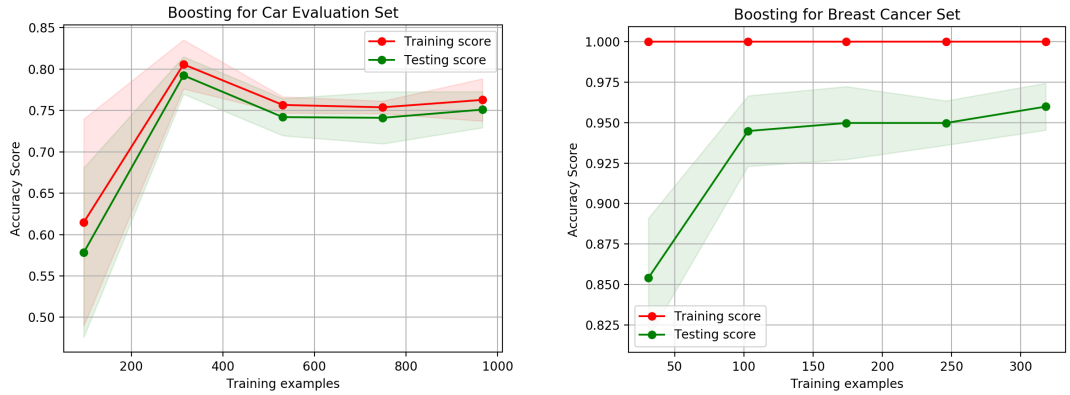

Figure 3.1, 3.2 The graphs show the training and testing curves for Boosting with on Car Evaluation Data set(left) and Breast Cancer Data set(right).

I got **0.789** for Car Evaluation Data set and **0.962** for Breast Cancer Data set. The results surprised me because the accuracy for Car Evaluation Data set was way below than I expected and Decision Tree learning algorithm. You may notice that training and testing curve are converging as number of training samples increased for the Car Evaluation Data Set. However, the standard deviation was too large because of its small data size. There are some possible reasons why it gives low accuracy. It's probably because of outliers or quality data. Adaboost is sensitive and highly affected by outliers because it tries to fit each point perfectly and the

ensemble method continues to attempt to correct misclassifications in the training data. Thus, we may conclude that Breast Cancel Data set has less noisy data or outliers, and more clear data than Car Evaluation Data set.

Even though Breast Cancer Data set gives pretty high accuracy, you may notice that training data is too perfect and it shows that the data is overfitting. To overcome this issue, I was trying to tune the most important parameters, which are n_estimators, and learning_rate. The n_estimators is the number of weak learners to train iteratively, and learning_rate contributes to the weights of weak learners. The default learning_rate is 1 and it showed that accuracy was decreased as it's increased. The best learning_rate I found was 0.9. Also, the number of estimators has direct relationship with accuracy and overfitting. I found that it's accuracy was increased as the number of estimators was increased and if it's too large such as over 15 or 20, then it makes model overfitting as well as higher accuracy. Thus, number of estimators = 10 and learning_rate = 0.9 prevents overfitting and maximizes the accuracy, which was **0.965**.

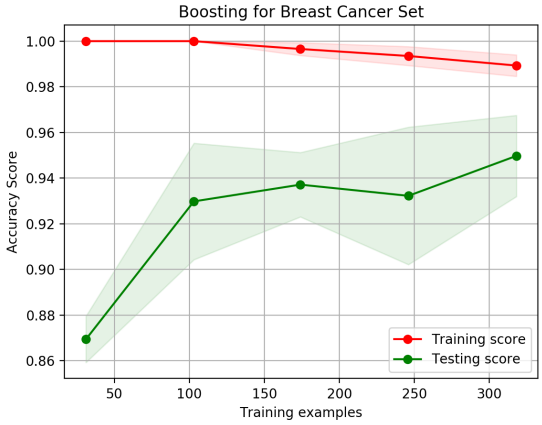

Figure 3.3 The graphs show the training and testing curves for Boosting on breast cancer set with number of estimator = 10 and learning_rate = 0.9.

The Figure 3.3 shows that the training and testing curve converging, which is a sign that overfitting is not occurred.

- **Support Vector Machine**

In this section, the SVM algorithm is implemented using a various kernel, which transforms an input data space into the required form and takes a low-dimensional input space and transforms it into a higher dimensional space (this is called kernel trick). I was going to play with Linear, Polynomial, and Radial Basis Function kernel to see how different kernels affect the accuracy for SVM because we really don't know which kernel may work best. My initial guess for accuracy of Car Evaluation Data set was that it might not work well with linear kernel because we noticed that Car Evaluation Data set was not really clean and had some outliers from Boosting learning algorithm and the data was too small.

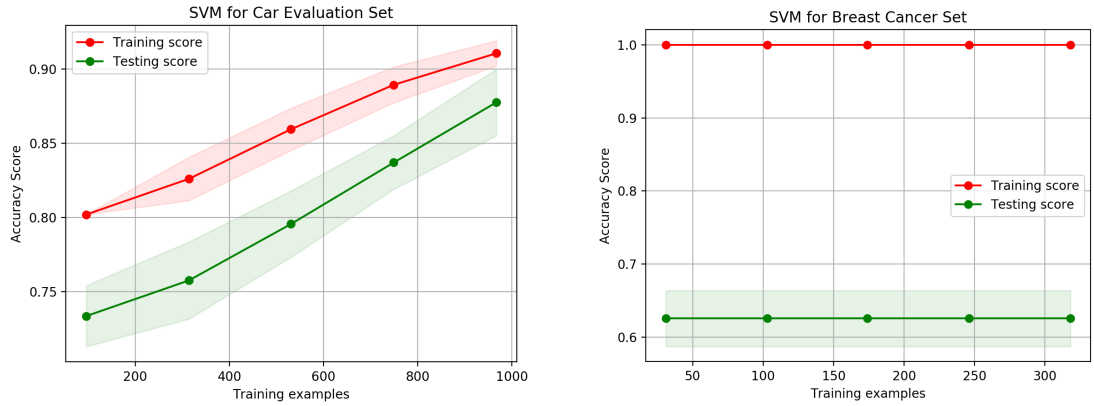

Figure 4.1, 4.2 The graphs show the training and testing curves for SVM with radial basis function kernel on Car Evaluation Data set(left) and the Breast Cancer Data set(right).

This graph shows that I got **0.913** accuracy for Car Evaluation Data set, which was little higher than I expected but it was acceptable because we knew that Car Evaluation Data set was not linearly separable. However, I got the worst accuracy, which was **0.632** for Breast Cancer Data set, which may imply that this data was linearly separated. We can check if those data were linearly separable or not with linear kernel on SVM.

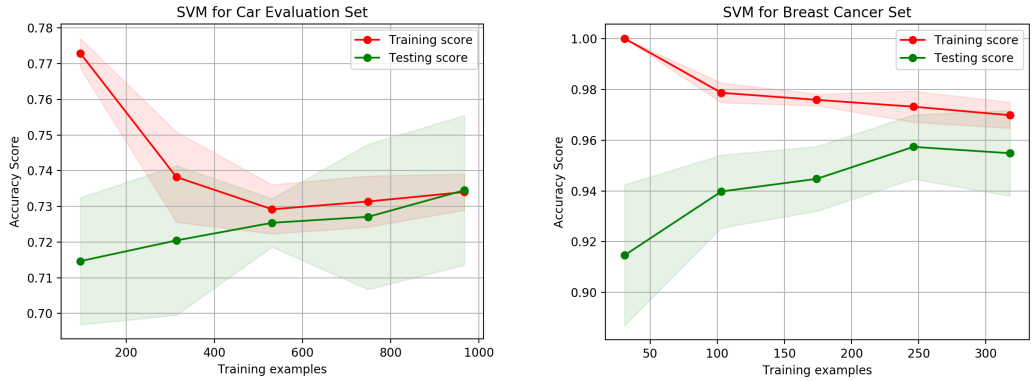

Figure 4.3, 4.4 The graphs show the training and testing curves for SVM with linear kernel on Car Evaluation Data set(left) and the Breast Cancer Data set(right).

From the graph, we can see that Car Evaluation Data set works poor on linear kernel with SVM (**0.705** accuracy) and Breast Cancer Data set works very good on linear kernel with SVM (**0.960** accuracy). Thus, we can conclude that Car Data set is non linearly separable and Breast Cancer Data set is linearly separable.

The polynomial kernel took 600 times longer than the running time of other kernels. Even the polynomial kernel doesn't give the best accuracy. While googling about SVM with kernels, I found out that polynomial kernel is not really powerful as we already observed from the graph. To improve accuracy, I was trying to tune the most important parameters, C, which is related to the simplicity of the decision surface, and gamma value defining how far a single instance

influences, which means that it's inversely related to the radius of influence samples. With several experiments with varying C,

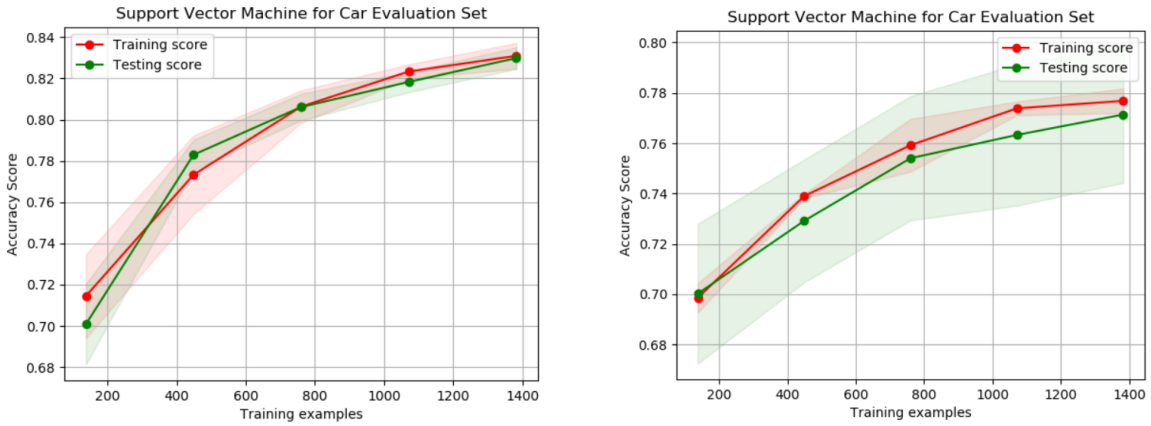

Figure 4.5, 4.6 The graphs show the training and testing curves for Support Vector Machine Car Evaluation Set with polynomial kernel on Car Evaluation Data set(left) and the Breast Cancer Data set (right).

I figured out the fact that smoothing the decision surface aggravates the performance of those data sets while changing the gamma value improved the accuracy a lot. Also, I applied scaling data before SVM to improve its performance because it helps to avoid attributes in greater numeric ranges dominate those in smaller numeric ranges and numerical difficulties during the calculation. As you can see the results from the table 1, it shows pretty drastic improvement in performance from a scaling.

	Linear Kernel	RBF Kernel
Running time with Scaling	`0.0023307800293`	`0.00628590583801`
Running time without Scaling	`1.06743311882`	`0.0111088752747`

Table 1 shows that running time differences between data with Scaling and without Scaling for Breast Cancer Data Set.

- **K Nearest Neighbor**

K-Nearest Neighbor (KNN) is a non-parametric method used for classification. kNN is also instance-based learning where the function is only approximated locally and all computation is deferred until classification. In this learning algorithm, I tested its accuracy with varying k, the number of neighbors, from 1 to 50 because research has shown that no optimal number of neighbors suits all kind of data sets. As you see the results from the graph, the cross-validation performance decreases as k increases over some point for both data set. kNN is proficient in dealing with noise in the data set as the number of neighbors, k increase. When comparing the two curves from the Figure 5.1 and 5.2, we can observe that the curve on the left is much smoother than the curve on the left, which implying that Car Evaluation Data set has much less noise rather than the Breast Cancer Data set. I got the highest accuracy, **0.923** with k = 7 for Car Evaluation Data set and **0.962** with k = 10.

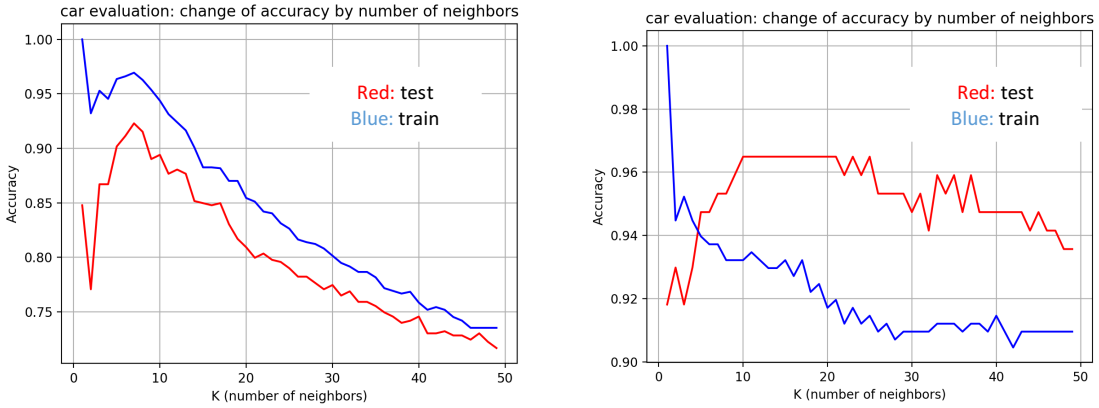

Figure 5.1, 5.2 The graphs show how changing number neighbors on Car Evaluation Set(left) and Breast Cancer Set(right) affect testing and training accuracy in K Nearest Neighbor.

I applied scaling of data before training model to see how it affects the accuracy. As you see the results from the Figure 5.3 and 5.4, scaling data before training model gives higher accuracy (**0.936** for Car Evaluation Data set and **0.977** for Breast Cancer Data set) for both data set. KNN requires scaling of data because KNN uses the Euclidean distance between two data points to find nearest neighbors. Euclidean distance is sensitive to magnitudes. The features with high magnitudes will weight more than features with low magnitudes.

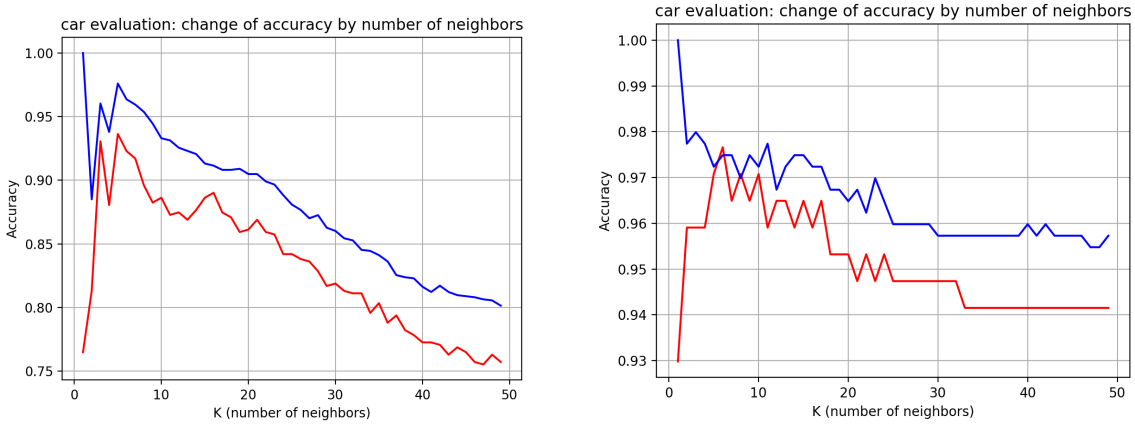

Figure 5.3, 5.4 The graphs show how changing number neighbors on Car Evaluation Set(left) and Breast Cancer Set(right) affect testing and training accuracy in K Nearest Neighbor with scaling data.

Conclusion

	Car Evaluation Data set	Breast Cancer Data set
Decision Tree	**0.00306**	**0.00284**
Neural Network	1.4032	0.0439
Boosting	0.0388	0.0465
SVM	0.0312	0.00289
KNN	1.2203	0.6523

Table 2 Running time comparison for five learning algorithms using two data sets

	Car Evaluation Data set	Breast Cancer Data set
Decision Tree	0.925	0.941
Neural Network	**0.956**	**0.988**
Boosting	0.789	0.965
SVM	0.913	0.960
KNN	0.936	0.977

Table 3 Accuracy comparison for five learning algorithms using two data sets

The two table shows that five learning algorithm's running time and accuracy. As we expected, the Decision Tree algorithm has the shortest running time because it's simplest. I was tweaking all parameters, doing data preprocessing, and normalization to make training less sensitive to the scale of features, and to achieve better performance. The best accuracy I got was Neural Network with normalizing data. The normalization of data boost up accuracy around 3% and 5% each for Car Evaluation Data set and Breast Cancer Data set and we found out that it's good to model the non-linear data with large number of input features; however, multi-layer neural networks are usually hard to train, and require turning lots of parameter to achieve higher accuracy. Contrary to the Neural Network algorithm, Decision Tree algorithm was quite easy to implement, non-parametric, and no needed to worry about outliers or whether the data is linearly separable. Also, it gives quite good performance compared to the other four learning algorithms in terms of its running time.

The worst learning algorithm based on its accuracy was Boosting with AdaBoost Classifier. Maybe, it was my bad to use AdaBoost Classifier because it's too sensitive to noisy data and outliers and too much relying on data and weak learner, which means that sometimes too complex leads to overfitting or low margins. With some experiments such as SVM with linear kernel gives 0.705 accuracy, which was worse than Boosting algorithnm, for the Car Evaluation Data set, we found that the Car Evaluation Data set has some noisy or outliers on it, and thus it was worst choice to use AdaBoost Classifier for Boosting learning algorithm. As we learned from class, it's recalling me "No Free Lunch" theorem which basically states that no one Machine Learning algorithm is best for all problems. Thus, the performance of different ML algorithms strongly depends on the size, structure of data, tuning parameters and preprocessing data.