Name : Aditya Milind Vadhavkar
GT username : avadhavkar3
GT Email ID : avadhavkar3@gatech.edu

CS 7641 : Machine Learning
Assignment 1 : Supervised Learning

Description of Classification Problems

1. **German Credit Data :** The German credit data is used to classify people described by a set of attributes as good or bad credit risks. There are 1000 instances and 20 attributes. The attributes are both categorical and numerical. The categorical attributes are further divided into two categories of nominal and ordinal. In this dataset there are nominal categorical attributes such as gender, where the two labels (male and female) cannot be compared with each other. Since, the classification algorithms in the scikit-learn library can only work with numerical attributes, I had to encode the categorical attributes. I had to use one hot encoding for nominal attributes, where we create a new feature for each of the class labels for that attribute and assign a 1 for the row where that label is present and 0 otherwise. To encode the ordinal attributes such as education I directly gave each class label a number with the highest number being assigned to the most educated. The target variable consists of only two classes, 1 and 0 with 1 being as bad risk and 0 being good. The evaluation metric that I chose for this dataset is recall score, because this data has class imbalance, that is, number of samples that are classified as bad credit risks are much lesser than the number of samples that are classified as good (300 instances of bad credit risk and 700 instances of good to be precise). Given the class imbalance problem it is not useful to have accuracy score as the evaluation metric as it will not give an actual idea of how the algorithm is performing. For example, even if any algorithm blindly classifies all the instances as good credit risks, it will still have 70% accuracy. Going further, it is very important to understand that classifying a customer who is actually a bad credit risk as good credit risk is extremely dangerous as it might lead to a loss for the bank if the bank approves a loan for that customer. However, classifying a customer who is actually a good credit risk as bad is not as dangerous as the previous one, although it will be beneficial to avoid classifying good customers as bad so that the bank doesn't lose out on earning interest from the loans provided to good customers. Thus, the aim of any classification algorithm for this dataset should be to minimise the number of false negatives (classifying bad users as good) and thereby maximise the recall score. Considering all the above points, this classification problem is interesting as it has a good mix of categorical and numerical attributes along with nominal and ordinal attributes within the categorical attributes. It also has data imbalance, which can cause problems while evaluating the model if proper evaluation metric is not used (it can be solved by oversampling of the minority class or undersampling of the majority class, although I haven't used these for this assignment). I have carefully selected recall score as my evaluation metric and also used area under the receiver operating

Name : Aditya Milind Vadhavkar
GT username : avadhavkar3
GT Email ID : avadhavkar3@gatech.edu

characteristics curve for evaluating the performance of the different classification algorithms as these metrics are not affected by the data imbalance problem.

2. **Australian Credit Card Approval Data :** This dataset consists of credit card applications and their corresponding labels wherein they have either been approved or denied. There are 690 instances and 14 attributes. The attributes are both categorical and numerical. There are nominal and ordinal attributes among those that are categorical. The encoding of the categorical attributes and their two categories is performed the same way as described for German Credit Data. This dataset does not have data imbalance problem, the instances with label 1 (people whose credit card application got denied) and the instances with label 0 (people whose credit card application got approved) are almost equal in number, that is, 383 instances are with label 0 and 307 instances are with label 1. The evaluation metric that I chose for comparing the different classification algorithms is accuracy for this dataset. Since, the two classes of the target variable are almost equally present in the dataset, accuracy can give a good idea of the performance of each algorithm. Along with accuracy score I am also calculating recall score, area under receiver operating characteristics curve and F1 score. Learning curve for this dataset is plotted for training sizes as a fraction of original training size vs (1 - accuracy score). For this dataset denying credit card to a customer who might be a potential defaulter in future is very important but at the same time approving all the good customers is also very important. Considering all the above points, this classification problem is non-trivial and interesting.

Experimental Settings

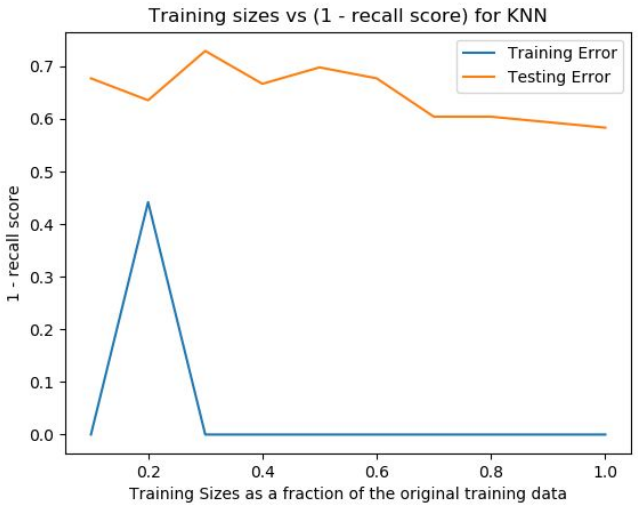
Both the datasets are imported into a pandas dataframe. The original data is split into train and test data with test size as 33% of the original data. For each of the algorithm K fold stratified cross validation is carried out for tuning the corresponding hyperparameters.The hyperparameters tuned for each of the classification algorithms common to both the datasets are as follows:

1. **KNN**: weights = [‘uniform’, ‘distance’], algorithm = [‘auto’, ‘ball_tree’, ‘kd_tree’, ‘brute’], n_neighbors = [3 to 37]
2. **Decision tree**: I have used pre-pruning by tuning the min_samples_leaf = [3 to 200] and max_depth = [5 to 100 with step of 5]
3. **SVM**: gamma = [0.1, 1, 10], kernel = [‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’]
4. **Adaboost**: n_estimators = [90 to 300 with step of 10] and max_depth of base estimator which is decision tree = [5 to 55 with step of 10]
5. **Neural Network**: activation_function = [‘identity’, ‘logistic’, ‘tanh’, ‘relu’], learning_rate = [‘constant’, ‘invscaling’, ‘adaptive’] and number of neurons in hidden layer = [100 to 250 with step of 10]

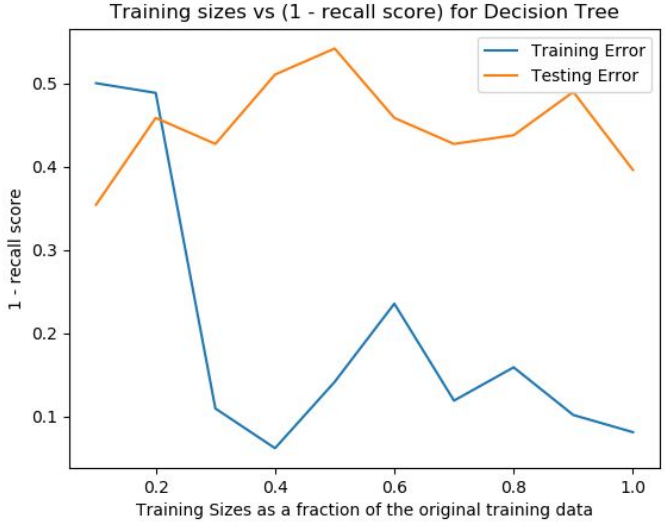
Name : Aditya Milind Vadhavkar
GT username : avadhavkar3
GT Email ID : avadhavkar3@gatech.edu

German Credit Data: The evaluation metric for this dataset is recall score. The number of folds for K fold cross validation are selected as 5. For Decision tree and Adaboost I have used select k best features with k as 15 out of the 24 available features because this gave the best recall score.
Australian Credit Approval Data: The evaluation metric for this dataset is accuracy score. The number of folds for K fold cross validation are selected as 5. All the 14 features are selected for training all the classification algorithms because I observed the best accuracy score was achieved with all 14 features.

Learning Curves
German Credit Data
KNN

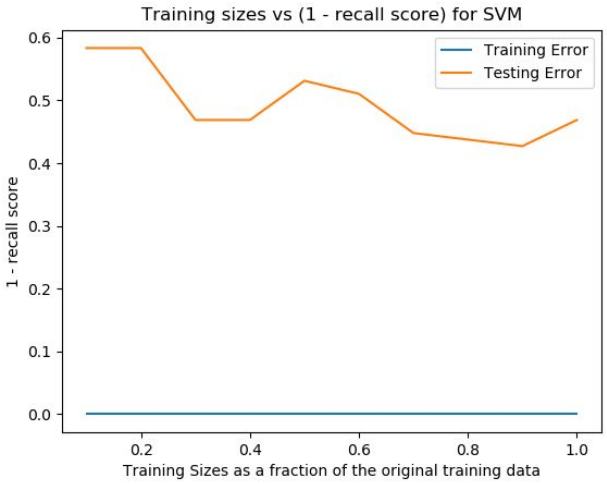


Decision Tree



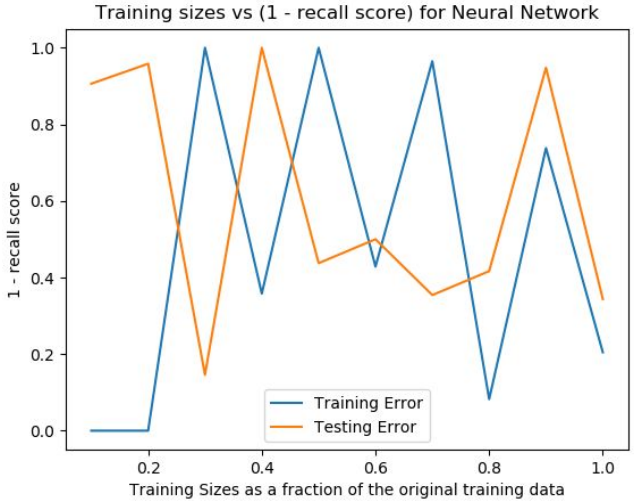
Name : Aditya Milind Vadhavkar
GT username : avadhavkar3
GT Email ID : avadhavkar3@gatech.edu

SVM



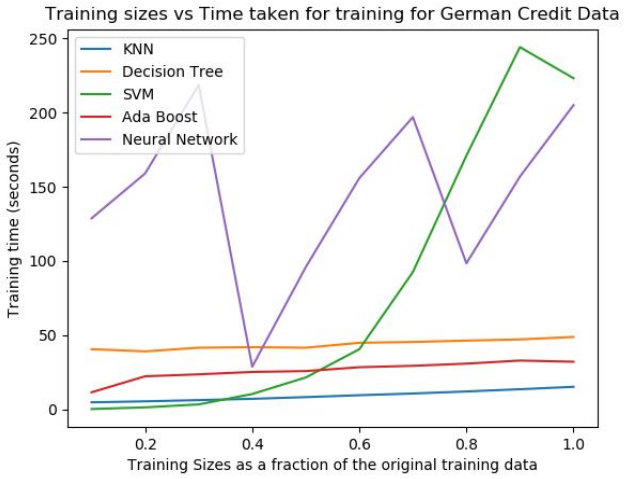
Adaboost

Neural Network



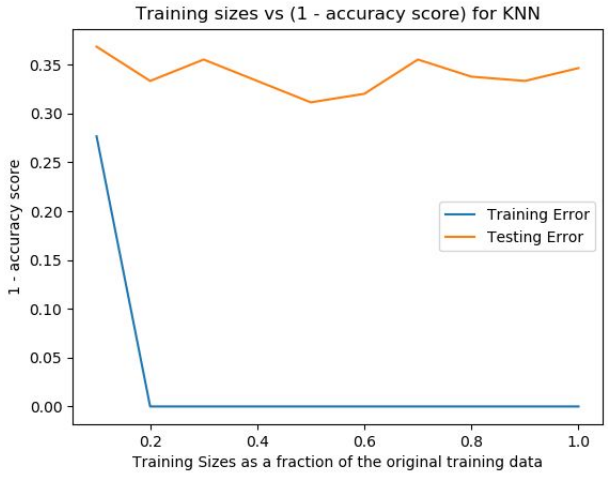
Name : Aditya Milind Vadhavkar
GT username : avadhavkar3
GT Email ID : avadhavkar3@gatech.edu

German Credit Data time taken for training vs training sizes



Australian Credit Approval Data

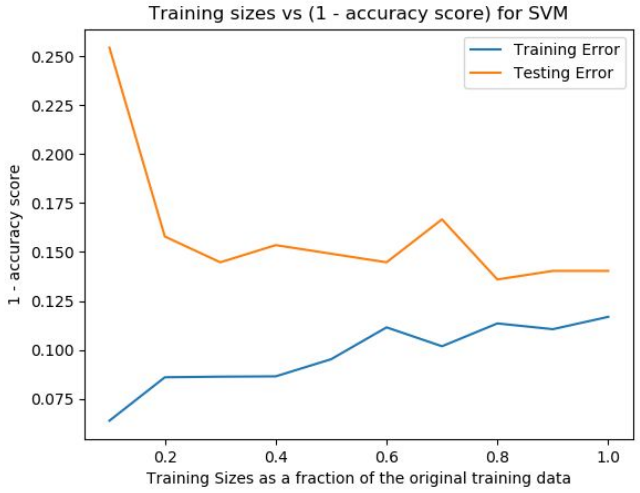
KNN



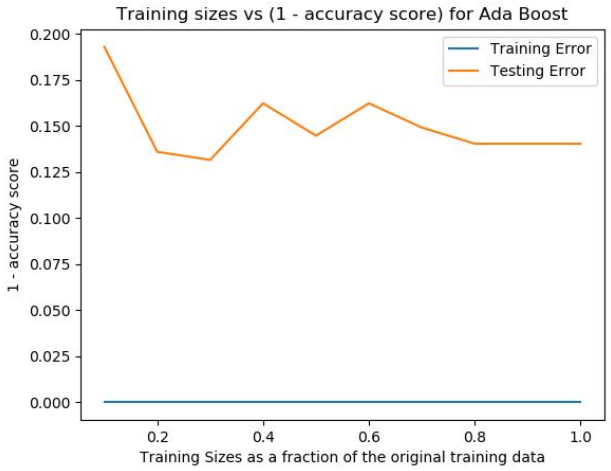
Decision Tree

Name : Aditya Milind Vadhavkar
GT username : avadhavkar3
GT Email ID : avadhavkar3@gatech.edu

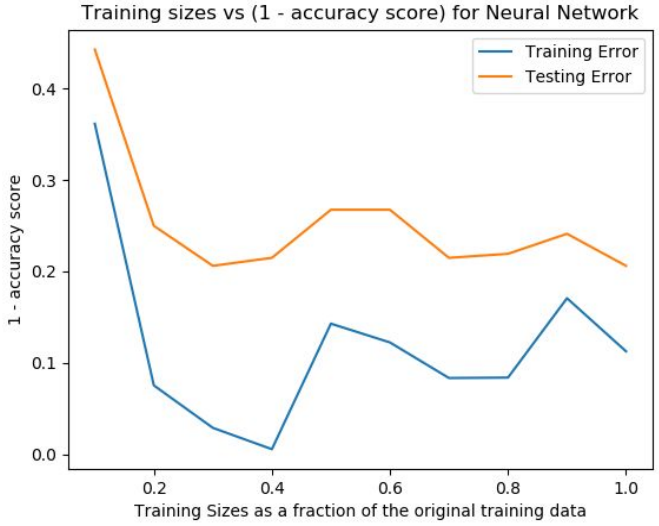
SVM



Adaboost

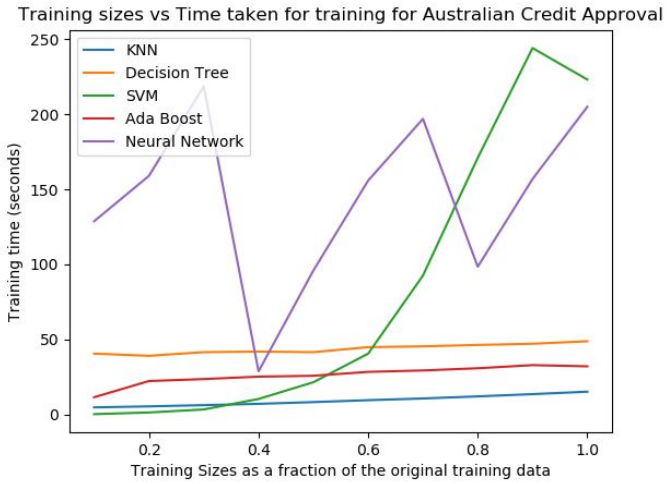


Neural Network



Name : Aditya Milind Vadhavkar
GT username : avadhavkar3
GT Email ID : avadhavkar3@gatech.edu

Australian Credit Approval data time taken for training vs training sizes



Results:

German Credit Data

```
German-Credit-Data
------------------------------------------------------------
KNN - Testing Data
Best paramaters = {'algorithm': 'auto', 'n_neighbors': 4, 'weights': 'distance'}
True negative = 181, False Positive = 53, False Negative = 56, True Positive = 40
Accuracy score = 0.6696969696969697
Recall score = 0.4166666666666667
AUC ROC score = 0.5950854700854702
F1 score = 0.42328042328042326

Decision Tree - Testing Data
Best paramaters = {'max_depth': 20, 'min_samples_leaf': 3}
True negative = 186, False Positive = 48, False Negative = 38, True Positive = 58
Accuracy score = 0.7393939393939394
Recall score = 0.6041666666666666
AUC ROC score = 0.6995192307692307
F1 score = 0.5742574257425742

SVM - Testing Data
Best paramaters = {'gamma': 0.1, 'kernel': 'poly'}
True negative = 178, False Positive = 56, False Negative = 45, True Positive = 51
Accuracy score = 0.693939393939394
Recall score = 0.53125
AUC ROC score = 0.6459668803418803
F1 score = 0.5024630541871922

AdaBoost - Testing Data
Best paramaters = {'base_estimator__max_depth': 15, 'n_estimators': 170}
True negative = 183, False Positive = 51, False Negative = 38, True Positive = 58
Accuracy score = 0.7303030303030303
Recall score = 0.6041666666666666
AUC ROC score = 0.6931089743589743
F1 score = 0.5658536585365854

Neural Network - Testing Data
Best paramaters = {'activation': 'tanh', 'hidden_layer_sizes': (240,), 'learning_rate': 'invscaling'}
True negative = 187, False Positive = 47, False Negative = 33, True Positive = 63
Accuracy score = 0.7575757575757576
Recall score = 0.65625
AUC ROC score = 0.7276976495726496
F1 score = 0.6116504854368932

------------------------------------------------------------
------------------------------------------------------------
```

Name : Aditya Milind Vadhavkar
GT username : avadhavkar3
GT Email ID : avadhavkar3@gatech.edu

Australian Credit Approval Data

```
Australian-Credit-Approval
------------------------------------------------------------
KNN - Testing Data
Best paramaters = {'algorithm': 'auto', 'n_neighbors': 12, 'weights': 'distance'}
True negative = 94, False Positive = 33, False Negative = 46, True Positive = 55
Accuracy score = 0.6535087719298246
Recall score = 0.5445544554455446
AUC ROC score = 0.6423559678802526
F1 score = 0.582010582010582

Decision Tree - Testing Data
Best paramaters = {'max_depth': 75, 'min_samples_leaf': 3}
True negative = 114, False Positive = 13, False Negative = 30, True Positive = 71
Accuracy score = 0.8114035087719298
Recall score = 0.7029702970297029
AUC ROC score = 0.8003040461526467
F1 score = 0.7675675675675676

SVM - Testing Data
Best paramaters = {'gamma': 0.1, 'kernel': 'linear'}
True negative = 112, False Positive = 15, False Negative = 17, True Positive = 84
Accuracy score = 0.8596491228070176
Recall score = 0.8316831683168316
AUC ROC score = 0.8567864660481795
F1 score = 0.84

AdaBoost - Testing Data
Best paramaters = {'base_estimator__max_depth': 5, 'n_estimators': 220}
True negative = 113, False Positive = 14, False Negative = 18, True Positive = 83
Accuracy score = 0.8596491228070176
Recall score = 0.8217821782178217
AUC ROC score = 0.8557729788726903
F1 score = 0.8383838383838383

Neural Network - Testing Data
Best paramaters = {'activation': 'logistic', 'hidden_layer_sizes': (200,), 'learning_rate': 'invscaling'}
True negative = 110, False Positive = 17, False Negative = 30, True Positive = 71
Accuracy score = 0.793859649122807
Recall score = 0.7029702970297029
AUC ROC score = 0.7845560146565839
F1 score = 0.7513227513227512
```

Name : Aditya Milind Vadhavkar
GT username : avadhavkar3
GT Email ID : avadhavkar3@gatech.edu

Analysis of the obtained results

KNN algorithm performed poorly on both the datasets due to overfitting. I tried selecting fewer features but the performance of KNN degraded even further. For German Credit Data SVM performed poorly as compared to Decision tree, Adaboost and Neural Networks because of the data. The data is too intertwined, the samples are mixed in such a way that SVM is not able to find a decision boundary that separates them in a clean way. The performance of Decision tree and Adaboost was somewhat comparable to that of Neural Network which performed the best among all the other algorithms. In the case of Australian Credit Approval Data, SVM performed the best in terms of the accuracy score with Adaboost classifier very close to it. Decision tree had relatively low performance as it suffered from overfitting. I have used Stratified Cross Validation for tuning hyperparameters using the GridSearchCV method of scikit-learn with the scoring metric as recall_score for German Credit Data and accuracy for Australian Credit Approval Data for all the algorithms. For the German Credit Data, I think oversampling of the minority class might help improve the recall score.

For both the datasets, KNN took the least amount of time to train over all the training sizes. Time taken to train SVM increased exponentially with increase in the size of the training data. The time taken to train Neural Network increased as the size of the training data increased, although the increase was not consistent. Training time for Decision tree and Adaboost increased almost linearly with increase in the size of the training data.

For Decision tree I have used pre-pruning by tuning the number of samples per leaf and the maximum depth of the tree. For Adaboost I have used pruning by tuning the number of instances of the base estimator (decision tree) and further I have tuned the maximum depth of the base estimator.

From the above results I concluded that for German Credit Data Neural Networks with activation function ‘tanh’, learning_rate ‘invscaling’ and number of neurons in the hidden layer as 240 gave the best result as it had the highest recall score and the least number of false negatives. For Australian Credit Approval, SVM with gamma as 0.1 and ‘linear’ kernel gave the best performance with the highest accuracy and area under the receiver operating characteristics curve.