CS7641 Machine Learning
Assignment 1 : Supervised Learning

Seema Hanji (shanji3)

1 Data Sets

1.1 Red Wine Quality

Location : winequality-red.csv : (https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv)

The data set contains various physicochemical variables (acidity, PH level etc.) as input and provides rating provided by wine experts as output. Since the ratings range between 0-10 , this data set can be challenging classification problem. Using this non-binary output , we can verify performance of various algorithms. It would be interesting to see how different algorithms apply feature selection.

The dataset contains **1,599** instances of wine and each instance is represented by 11-input variables or features. The output classes are ordered , but not balanced.

1.2 Credit Card Payment Default

Location : default_of_credit_card_clients.csv: (https://archive.ics.uci.edu/ml/machine-learning-databases/00350/)

The dataset is to predict if credit card clients in Taiwan, will be defaulting their next month's payment, based on various factors. These factors include client's education, gender, credit balance , past payment history etc. which act as input features to classification problem. The response variable is binary, indicating the probability of client defaulting next payment.

The dataset provides 23 variables as input features and binary response variable. This being a large data set , including 30,000 instances , can be good example to test performance of learning algorithms.

As per [Giudici, 2001] , data mining techniques play a key role in market segmentation fraud detection, credit and behavior scoring, and benchmarking . This data set serves one of those purposes, based on probability of defaulting, can provide appropriate credit score.

2 Learning Algorithms

In all algorithms, dataset was split into 70% training and 30% test data. After training the model with train dataset, test data was used to predict the response of wine quality rating.

2.1 Decision Trees

To test above specified datasets, Decision Tree algorithm from Python library **sklearn.tree** *(DecisionTreeClassifier)* is used. The criteria used for selecting attribute for splitting is 'Gini Index'.

Gini Index is calculated as the amount of probability of a specific feature that is classified incorrectly when selected randomly. And, the attribute with minimum Gini index is chosen to split the tree.

Gini Index $= 1 - \sum_{i=1}^{n} P(i)^2$

Where P(i) is probability of an element being classified for class i.

2.1.1 Classification of Wine Quality dataset

Decision Tree algorithm is applied on ***winequality-red*** data set, with optimized pruning arguments applied as : *max_dept = 6* and *min_samples_leaf = 3.* Here are graphs of learning curve and validation curve.

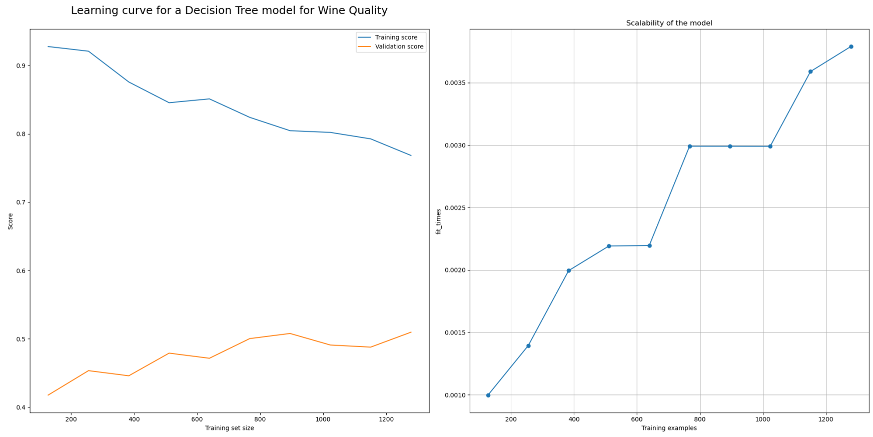

Figure 1: Wine Quality Learning curve

Learning curve indicates that the model is over fitting when number of samples is less, as there is big gap between training accuracy and cross-validation accuracy. As samples size increases, training and validation accuracy start converging, indicating model is generalizing better.

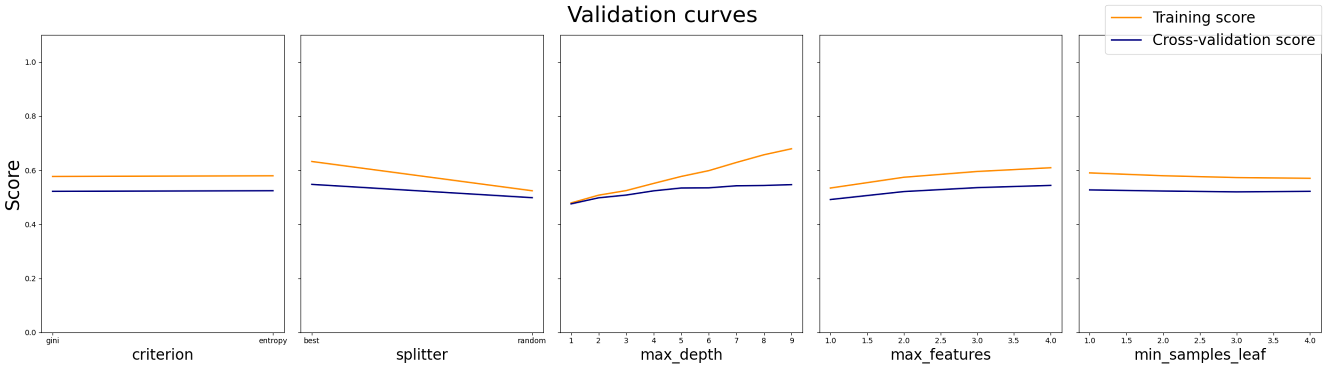

Figure 2: Wine Quality Validation curves

Validation curves for different hyper parameters criteria, max_depth, max_features etc. indicate optimal parameter to use for the model. For all these parameters, the optimal value is when bias is not high and not over fitting. This is at a point where cross validation score has reached max point and there is gap between train and validation scores. As per the chart, best params : 'criterion': 'gini', 'max_depth': 6, 'max_features': 4, 'min_samples_leaf': 3, 'splitter': 'best'

Result was produced with **Accuracy = 60.83%** Elapsed time of train and test : **5.35 sec** Here is comparison of actual wine ratings vs predicted from the test set.

Predicted Actual	4	5	6	7	8	All
3	0	1	1	0	0	2
4	1	10	4	1	0	16
5	2	146	63	3	0	214
6	2	47	118	16	2	185
7	0	2	26	27	3	58
8	0	0	2	3	0	5
All	5	206	214	50	5	480

Table 1: Prediction Vs. Actual

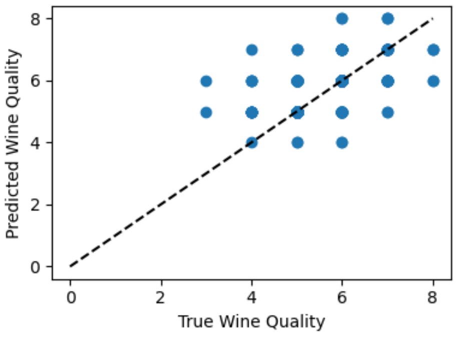

Figure 3: Wine Quality Prediction

2.1.2 Classification of Defaults for Credit Card dataset

Decision Tree algorithm is applied on ***default_of_credit_card_clients*** data set, with pruning

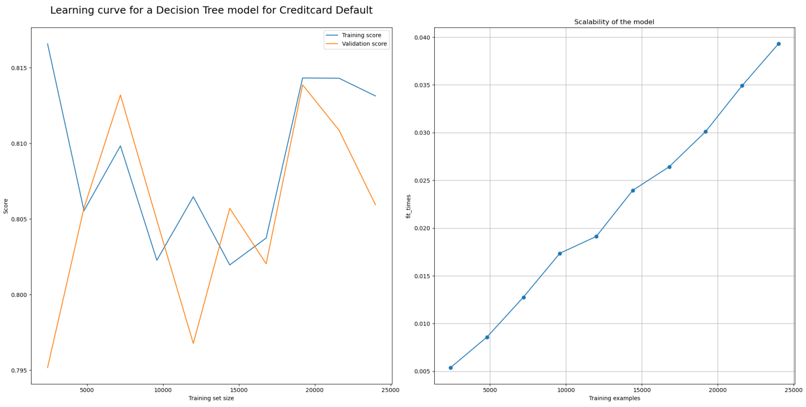

Figure 4: Credit Card Default Learning curve

Learning curve in this model varies a lot based on data sample size. Training accuracy initially starts as high and validation accuracy starts low , as data sample size increases , both curves cross each other and at a point (20,000) both accuracy are a high. This is optimal size to choose for data set split.

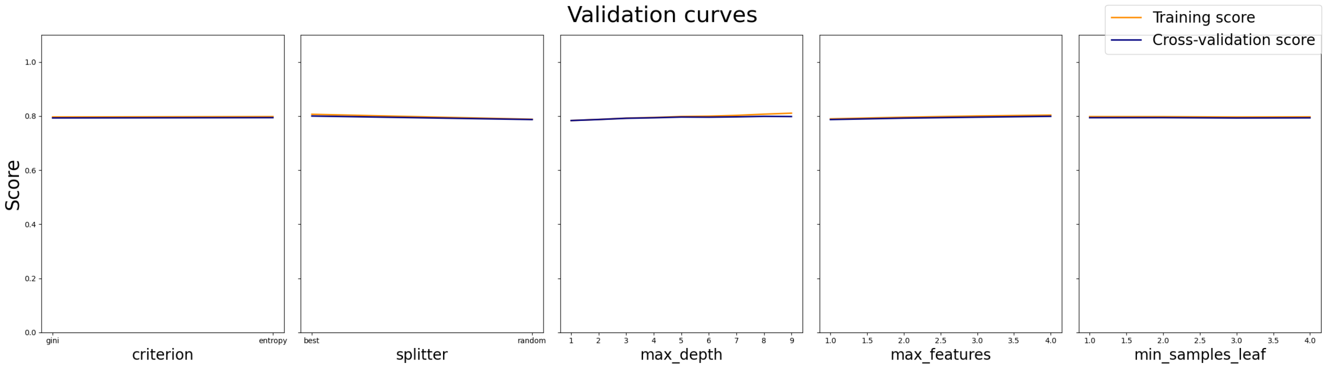

Figure 5: Credit Card Default Validation curves

Validation curves indicate that , there is no real impact of hyper parameters on the model. Training and validation curves have high accuracy and are very close to each other. This indicates that the sampled data might not be well balanced , which is producing a perfect model. The model can be made complex, by sampling data with more balanced classes. Result was produced with **Accuracy = 81.61%** Best params : 'criterion': 'entropy', 'max_depth': 6, 'max_features': 3, 'min_samples_leaf': 3, 'splitter': 'best' Elapsed time of train and test : **25.24 sec** Here is comparison of actual values for defaulted vs predicted from the test set.

Predicted Actual	0	1	All
0	6607	445	7052
1	1210	738	1948
All	7817	1183	9000

Table 2: Prediction Vs. Actual

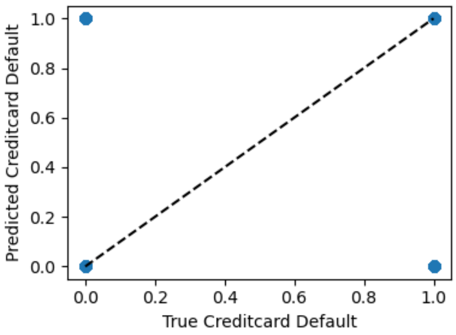

Table 3: Credit Card Default Prediction

2.2 Decision Tree with Boost

Decision Tree with boost, is a method of combining weak learners iteratively to form a strong learner. The kind of boosting algorithm used here is Gradient Boost, which optimizes loss function by ensembling weak learners , here decision trees. Algorithm from Python library **sklearn.ensemble** *(GradientBoostingClassifier)* is used. The optimal max depth and learning rate to found using cross validation using various values of these hyper parameters.

2.2.1 Classification of Wine Quality dataset

Gradient Boosting Decision Tree algorithm is applied on ***winequality-red*** data set.

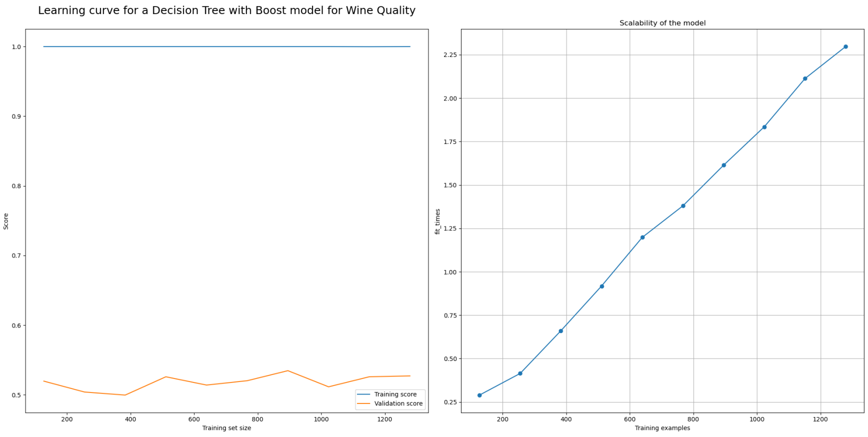

Figure 6: Wine Quality Learning curve

Learning curve indicates that the model is over fitting when number of samples is less, validation accuracy is low and as sample size increases, accuracy also increases, indicating model is performing better. Looking at huge gap between training and validation accuracy score, this model is biased for this dataset.

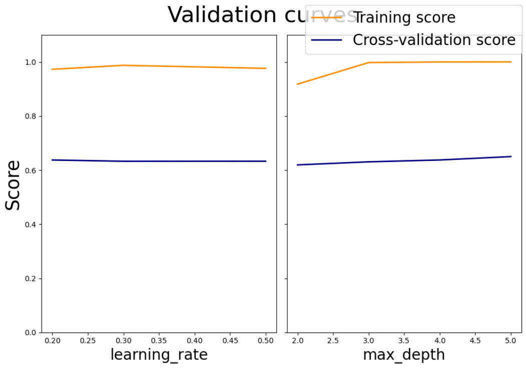

Figure 7: Wine Quality Validation curves

Validation curves for different hyper parameters learning_rate, max_depth. indicate optimal parameter to use for the model. The huge gap between training and CV score indicates model is biased. Max scores in CV indicate optimal paramters.

Result was produced with **Accuracy = 100.00%** Best params : 'learning$_rate' : 0.2,'max_depth' : 4Elapsedtimeoftrainandtest : 75.68Hereiscomparisonofactualwineratingsvspredictedfromthetestset.$

Predicted Actual	3	4	5	6	7	8	All
3	2	0	0	0	0	0	2
4	0	16	0	0	0	0	16
5	0	0	214	0	0	0	214
6	0	0	0	185	0	0	185
7	0	0	0	0	58	0	58
8	0	0	0	0	0	5	5
All	2	16	214	185	58	5	480

Table 4: Prediction Vs. Actual

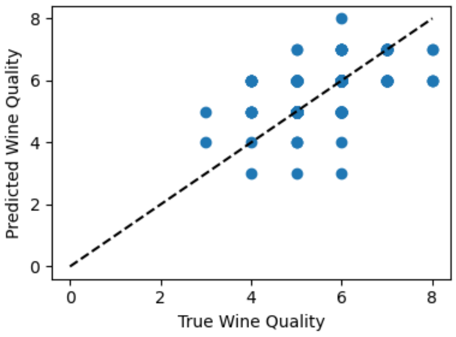

Table 5: Wine Quality Prediction

2.2.2 Classification of Defaults for Credit Card dataset

Gradient Boosting Decision Tree algorithm is applied on ***default_of_credit_card_clients*** data set, with pruning arguments applied as : *max_dept = 2*.

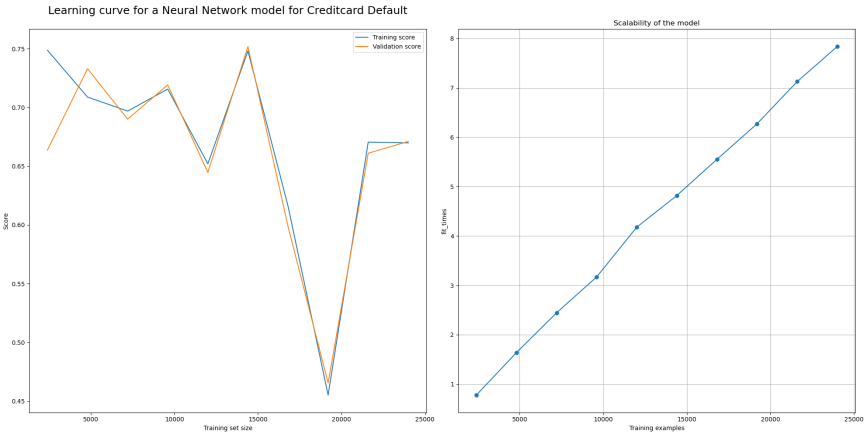

Figure 8: Credit Card Default Learning curve

Learning curve indicates that the model is over fitting when number of samples is less, as there is big gap between training accuracy and cross-validation accuracy. As samples size increases, training and validation accuracy start converging, indicating model is generalizing better.

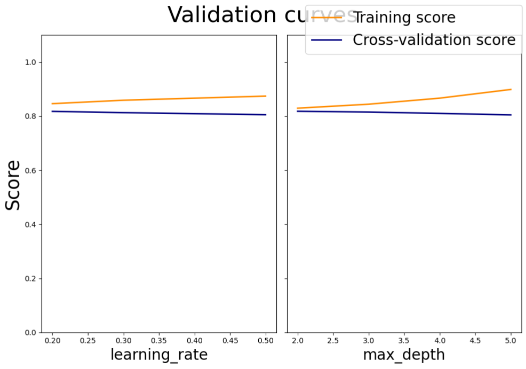

Figure 9: Credit Card Default Validation curves

Validation curves for different hyper parameters learning_rate, max_depth. indicate optimal parameter to use for the model. For all these parameters, the optimal value is when bias is not high and not over fitting. This is at a point where cross validation score start increasing and reaches max.

Result was produced with **Accuracy = 82.56%** Best params : 'learning$_{r}ate'$: $0.2,'max_{d}epth'$: $2Elapsedtimeoftrainandtest$: **320.10 sec**

Here is comparison of actual values for defaulted vs predicted from the test set.

Predicted Actual	0	1	All
0	6666	386	7052
1	1223	725	1948
All	7889	1111	9000

Table 6: Prediction Vs. Actual

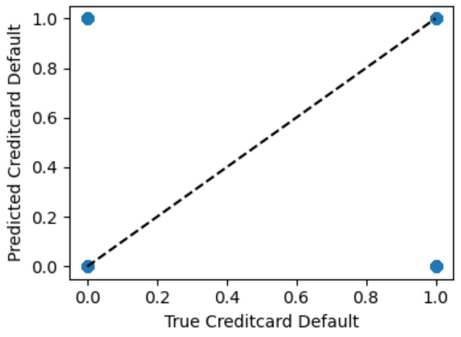

Table 7: Credit Card Default Prediction

2.3 Neural network

Neural network algorithm finds relationships in dataset using a process which mimics the way human brain works. Neural networks contain multiple layers as input , output and hidden. The main job is to transform input into valuable output. The type of neural network algorithm used here is "Multilayer Perceptron" , which uses back propagation for training. It is helpful in non-linear data.

To test above specified datasets, Multilayer Perceptron - Neuron network algorithm from Python library **sklearn.tree** *(MLPClassifier)* is used.

2.3.1 Classification of Wine Quality dataset

MLP Neuron Network algorithm is applied on ***winequality-red*** data set, optimal hyper params used. Some of the solver algorithms used are : 'adam' - is a stochastic gradient-based optimizer 'tanh' - the hyperbolic tan function, returns f(x) = tanh(x)

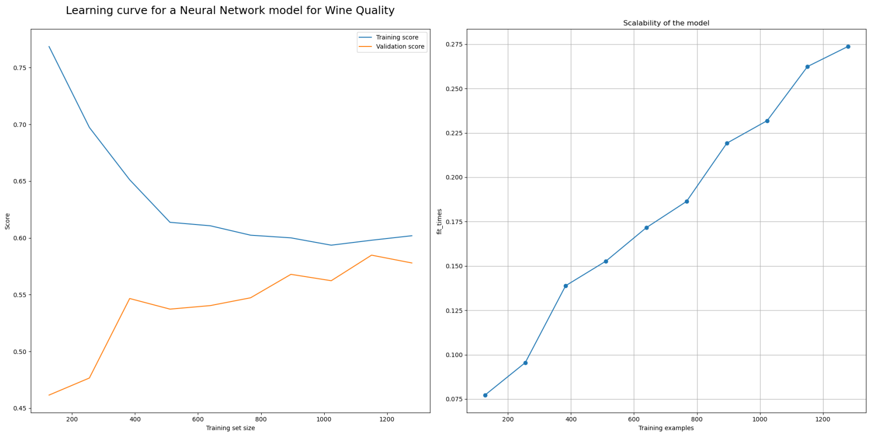

Figure 10: Wine Quality Learning curve

Learning curve indicates that the model is over fitting when number of samples is less, as there is big gap between training accuracy and cross-validation accuracy. As samples size increases, training and validation accuracy start converging, indicating model is generalizing better.

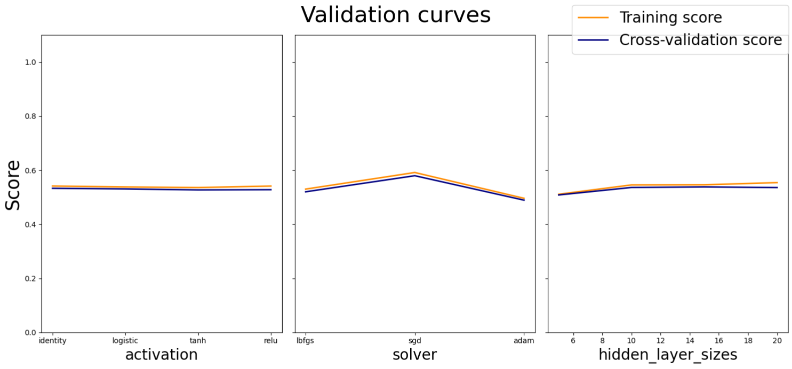

Figure 11: Wine Quality Validation curves

Validation curves for different hyper parameters activation, solver, hidden_layer_sizes. indicate optimal parameter to use for the model. For all these parameters, the optimal value is when bias is not high and not over fitting and cross validation value is at its max. Result was produced with **Accuracy = 58.13%** Best params : 'activation': 'tanh', 'hidden$_layer_sizes' : (20,),' solver' :' lbfgs' Elapsed time of train and test : \mathbf{26.7 sec} Here is comparison of actual wine ratings vs predicted from the test set.$

Predicted Actual	5	6	7	All
3	1	0	0	1
4	15	1	0	16
5	164	51	3	218
6	56	111	20	187
7	4	28	22	54
8	0	2	2	4
All	240	193	47	480

Table 8: Prediction Vs. Actual

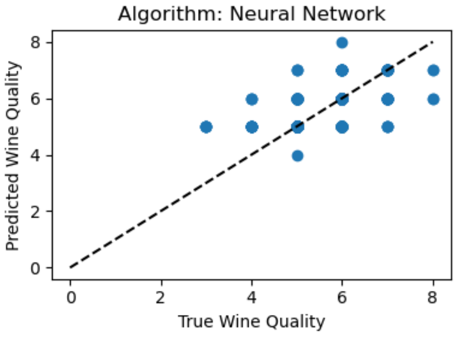

Table 9: Wine Quality Prediction

2.3.2 Classification of Defaults for Credit Card dataset

MLP Neuron Network algorithm is applied on ***default_of_credit_card_clients*** data set, with optimal paramters.

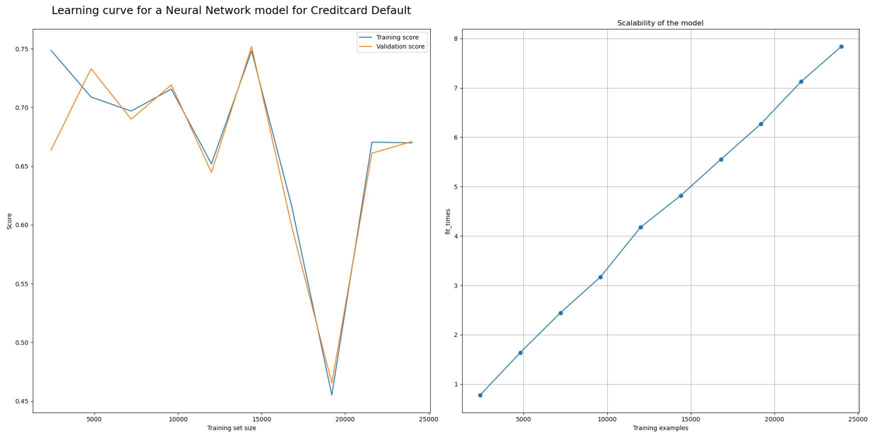

Figure 12: Credit Card Default Learning curve

Learning curve indicates that the model is over fitting when number of samples is less, as there is gap between training accuracy and cross-validation accuracy. The curve indicates, both training and CV accuracy match as sample size increases and there is a point where both are max.

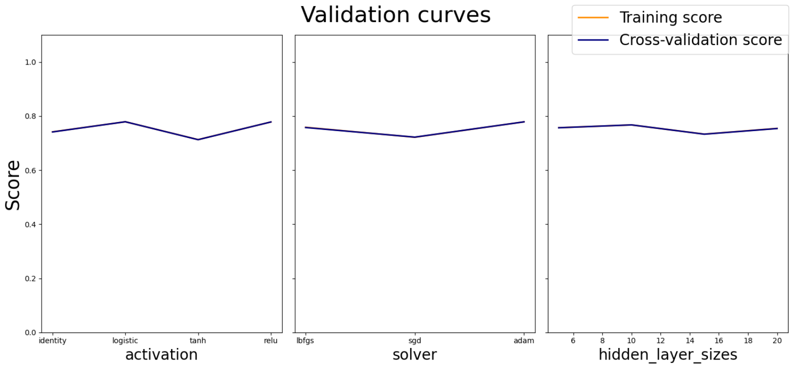

Figure 13: Credit Card Default Validation curves

Validation curves for different hyper parameters activation, solver, hidden_layer_sizes. indicate optimal parameter to use for the model.The point where CV score is max is optimal param. The chart also indicates the model is not biased, as both train and CV scores match across.

Result was produced with **Accuracy = 77.42%** Best params : 'activation': 'logistic', 'hidden_layer_sizes': (5,), 'solver': 'adam' Elapsed time of train and test : **393.43sec**

Here is comparison of actual values for defaulted vs predicted from the test set.

Predicted Actual	0	1	All
0	6952	24	6976
1	2008	16	2024
All	8960	40	9000

Table 10: Prediction Vs. Actual

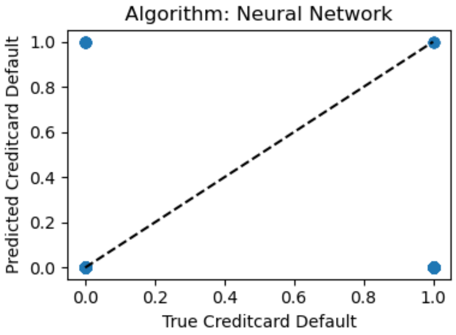

Table 11: Credit Card Default Prediction

2.4 Support Vector Machines

Support Vector Machine (SVM) is a classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. SVM uses a function called to **kernel** to transform low dimensional input space to high dimensional. There are different functions that can be applied as kernel functions.

'rbf'(Radial basis function) - Helpful for non-linear hyper plane 'linear' - Used for linear plane

2.4.1 Classification of Wine Quality dataset

SVM is applied on ***winequality-red*** data set, with kernel functions 'rbf' (SVC(kernel='rbf)) and 'linear' (LinearSVC()).

Validation curves for different hyper parameter gamma. indicate optimal parameter to use for the model. For all these parameter values, the optimal value is when bias is not high and not over fitting.

For kernel='rbf' Result produced with **Accuracy = 51.56%** Best params : 'gamma': 0.001 Elapsed time of train and test : **0.09sec**

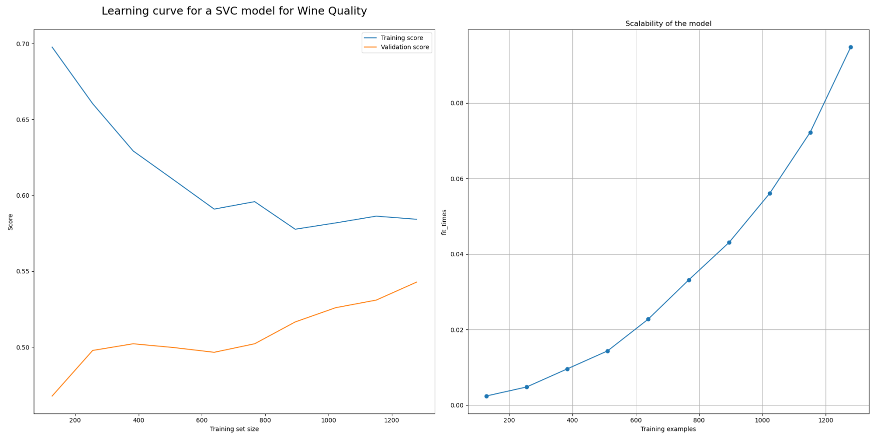

Figure 14: Wine Quality Learning curve

Learning curve indicates that the model is over fitting when number of samples is less, as there is big gap between training accuracy and cross-validation accuracy. As samples size increases, training and validation accuracy start converging, indicating model is generalizing better.

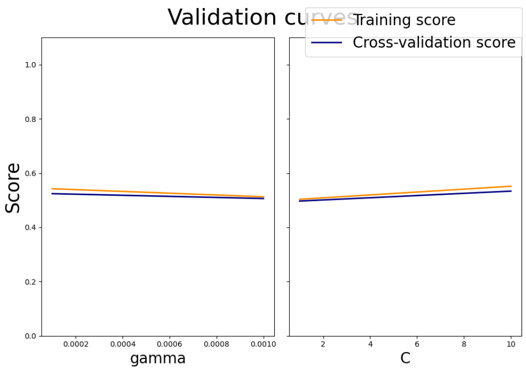

Figure 15: Wine Quality Validation curves

Validation curve indicates model is not biased.

Here is comparison of actual wine ratings vs predicted from the test set.

Predicted Actual	5	6	7	All
3	0	2	0	2
4	4	12	0	16
5	83	131	0	214
6	29	156	0	185
7	4	53	1	58
8	0	5	0	5
All	120	359	1	480

Table 12: Prediction Vs. Actual

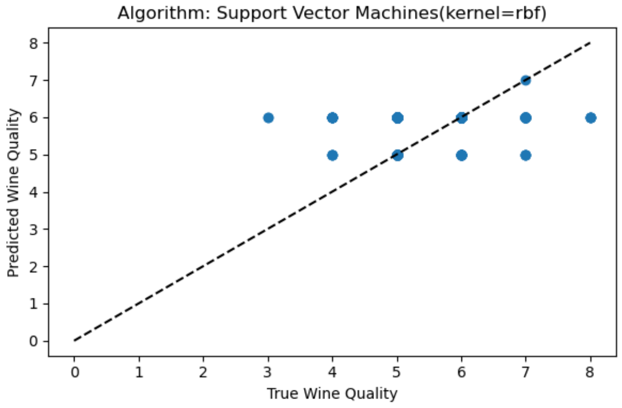

Table 13: Wine Quality Prediction

For kernel='linear' Result was produced with **Accuracy = 40.21%** Elapsed time of train and test : **0.44sec**

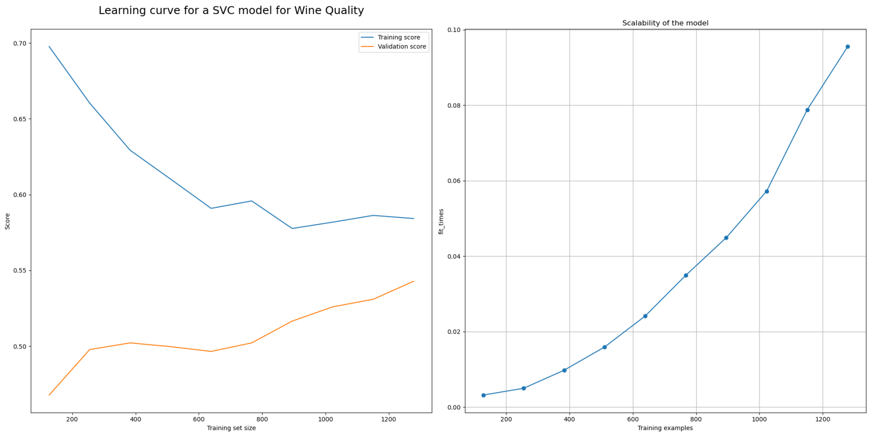

Figure 16: Wine Quality Learning curve

Learning curve indicates that the model is over fitting when number of samples is less, as there is big gap between training accuracy and cross-validation accuracy. As samples size increases, training and validation accuracy start converging, indicating model is generalizing better.

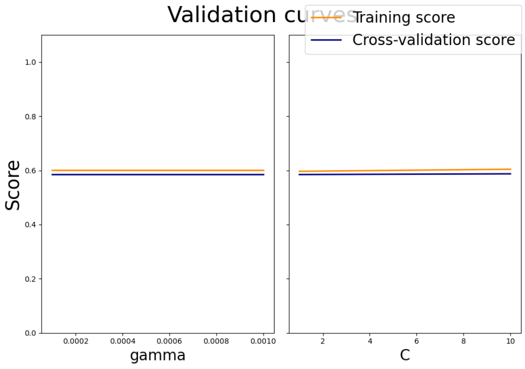

Figure 17: Wine Quality Validation curves

Here is comparison of actual wine ratings vs predicted from the test set.

Predicted Actual	5	6	7	All
3	1	1	0	2
4	0	16	0	16
5	6	204	4	214
6	0	177	8	185
7	0	48	10	58
8	0	3	2	5
All	7	449	24	480

Table 14: Prediction Vs. Actual

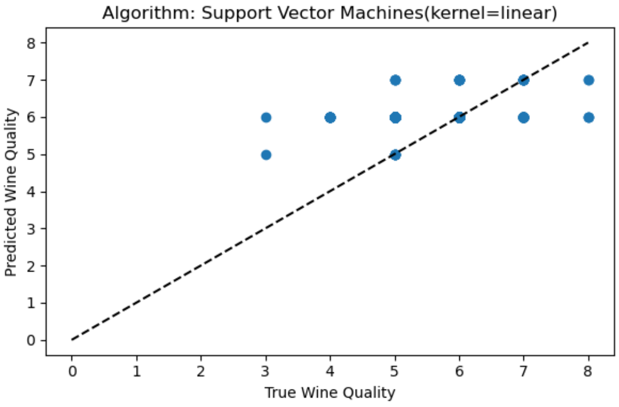

Table 15: Wine Quality Prediction

2.4.2 Classification of Defaults for Credit Card dataset

SVM is applied on ***default_of_credit_card_clients*** data set, with kernel functions 'rbf' (SVC(kernel='rbf)) and 'linear' (LinearSVC()).

For kernel='rbf' Best params : 'gamma': 0.0001 Elapsed time of train and test : **139.74sec** Result was produced with **Accuracy = 78.36%**

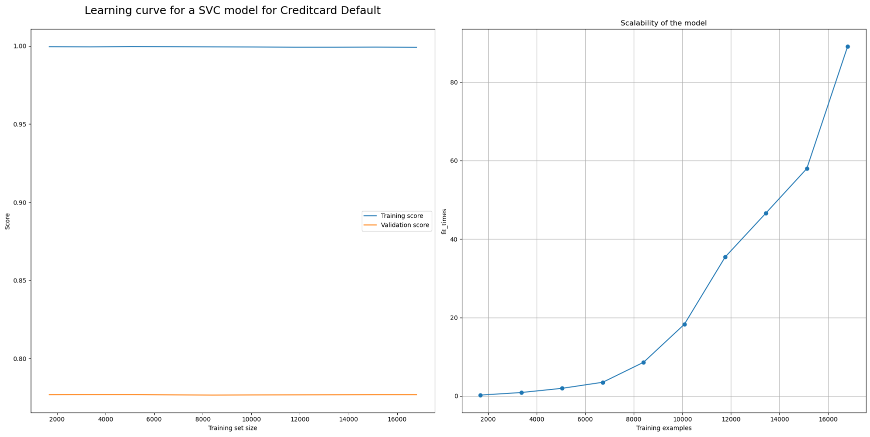

Figure 18: Credit Card Default Learning curve

Learning curve indicates that the model is over fitting as there is big gap between training accuracy and cross-validation accuracy. It doesn't change for sample size.

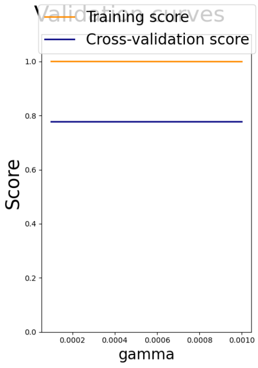

Figure 19: Credit Card Default Validation curves

Validation curves for different hyper parameter gamma indicate optimal parameter to use for the model. For all these parameters, the optimal value is when bias is not high and not over fitting. Model indicates again it is biased.

Here is comparison of actual values for defaulted vs predicted from the test set.

Predicted Actual	0	All
0	7052	7052
1	1948	1948
All	9000	9000

Table 16: Prediction Vs. Actual

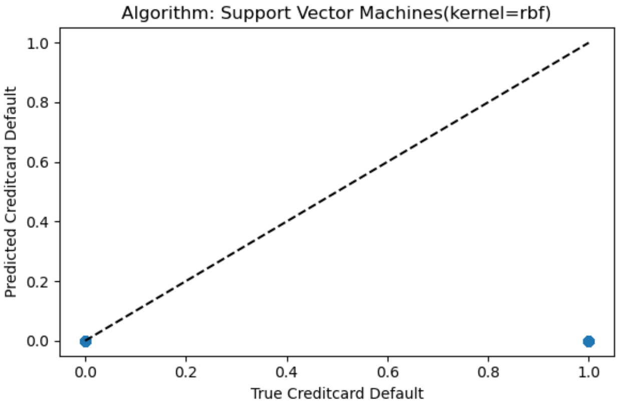

Table 17: Credit Card Default Prediction

For kernel='linear'

Result was produced with **Accuracy = 52.36%** Elapsed time of train and test : **3.05sec**

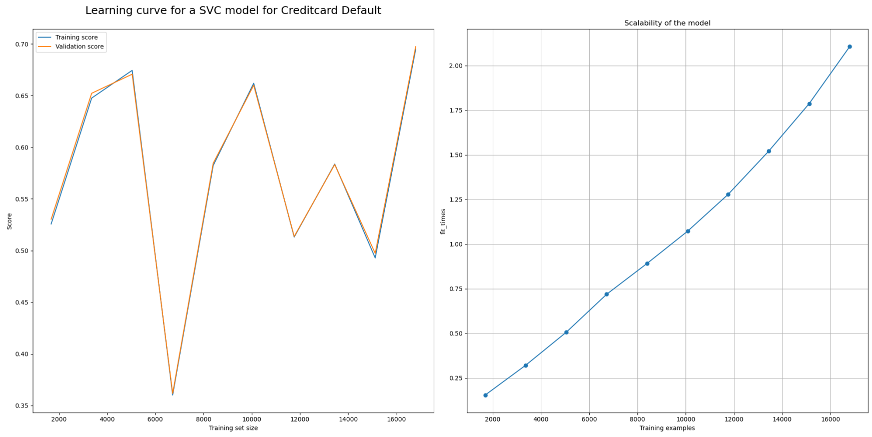

Figure 20: Credit Card Default Learning curve

Learning curve indicates that the model is not over-fitting. As samples size increases, training and validation accuracy vary a lot and there are some points where they reach max, which is ideal sample size.

For SVM with kernel 'linear' , there hyper paramters tuning doesn't impact performance of the model and hence there is no need of cross validation.

Here is comparison of actual values for defaulted vs predicted from the test set.

Predicted Actual	0	1	All
0	3455	3597	7052
1	691	1257	1948
All	4146	4854	9000

Table 18: Prediction Vs. Actual

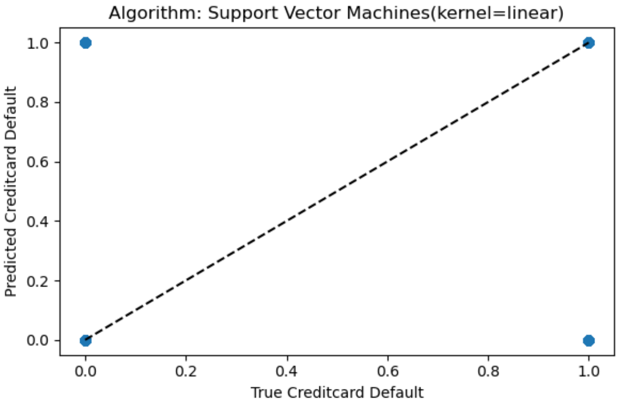

Table 19: Credit Card Default Prediction

2.5 k-Nearest Neighbors

k-Nearest neighbors (KNN) is a popular classification algorithm. It stores the whole data set and their corresponding distance to each other and based on this classifies new data. To test above specified datasets, KNN algorithm from Python library **sklearn.neighbors** *(KNeighborsClassifier)* is used, with different k (number of neighbors to use for querying).

2.5.1 Classification of Wine Quality dataset

KNN algorithm is applied on ***winequality-red*** data set, with neighbours size ranging between : *k = 1* and *k = 10.*

Optimal n_neighbours(k) = 1

Result was produced with **Accuracy of KNN(k=1) = 56.04%** Elapsed time of train and test : **4.20sec**

Here is comparison of actual wine ratings vs predicted from the test set.

Predicted Actual	3	4	5	6	7	8	All
3	1	1	0	0	0	0	2
4	0	0	6	8	2	0	16
5	0	8	123	68	13	2	214
6	0	3	44	115	20	3	185
7	0	0	6	19	30	3	58
8	0	0	0	4	1	0	5
All	1	12	179	214	66	8	480

Table 20: Prediction Vs. Actual

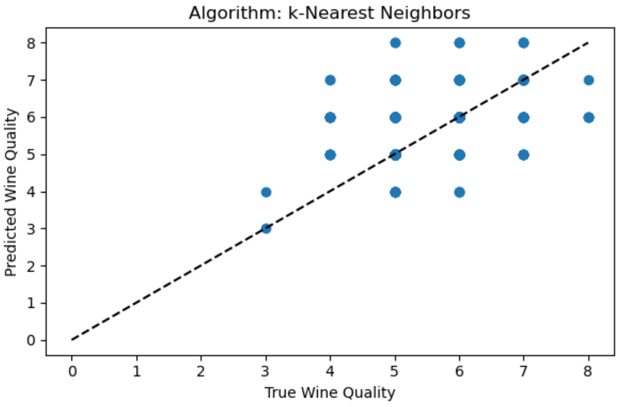

Table 21: Wine Quality Prediction

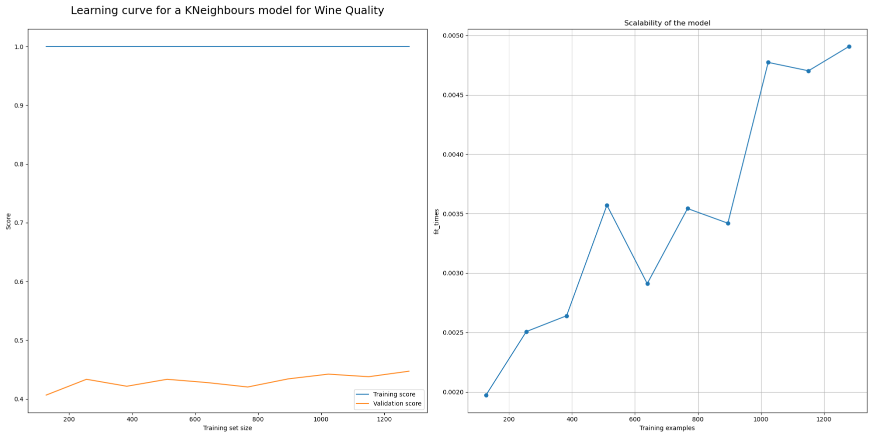

Figure 21: Wine Quality Learning curve

Learning curve indicates that the model is over fitting , as there is big gap between training accuracy and cross-validation accuracy.

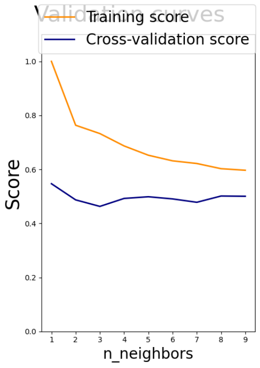

Figure 22: Wine Quality Validation curves

Validation curves for different hyper parameter n_neighbours indicates optimal parameter to use for the model. Optimal value is one where CV score is high.

2.5.2 Classification of Defaults for Credit Card dataset

KNN algorithm is applied on ***winequality-red*** data set, with neighbours size ranging between : k = *1* to k = *10*.

Optimal k=8

Result was produced with **Accuracy of KNN(k=8) = 77.66%** Elapsed time of train and test : **42sec**

Here is comparison of actual values for defaulted vs predicted from the test set.

Predicted Actual	0	1	All
0	6757	295	7052
1	1762	186	1948
All	8519	481	9000

Table 22: Prediction Vs. Actual

images/knn6_2.png

Table 23: Credit Card Default Prediction

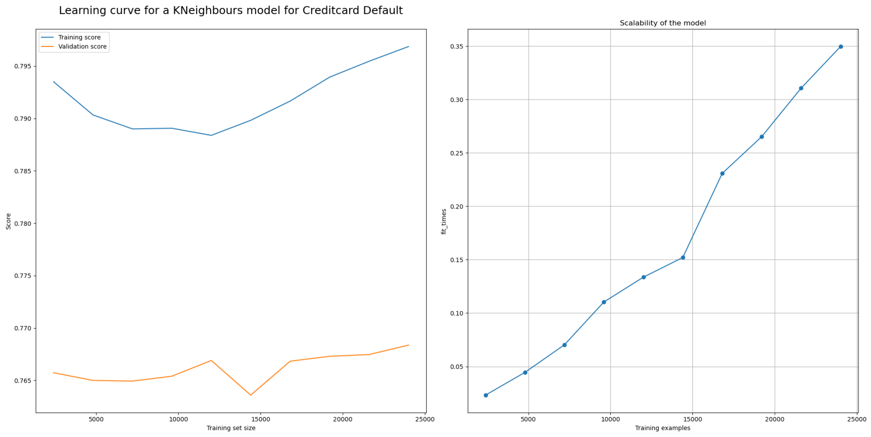

Figure 23: Credit Card Default Learning curve

Learning curve indicates that the model is over fitting ,as there is big gap between training accuracy and cross-validation accuracy. As samples size increases, training and validation accuracy start increasing, indicating optimal size.

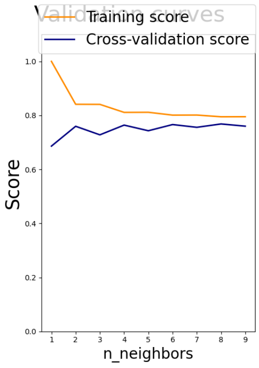

Figure 24: Credit Card Default Validation curves

Validation curves for different hyper parameter n_neighbours indicates optimal parameter to use for the model. Optimal value is one where CV score is high.

3 Conclusion

Based on the results of applying various learning algorithms for two selected classification problems, *Decision Tree with Gradient Boost* performed fairly better, with accuracy of *100%* for Wine quality dataset and accuracy of *82%*for credit card default dataset. Also, in general all the algorithms performed better in terms of accuracy on the "Credit Card Default" data set, which is binary classification problem in comparison to "Red Wine Quality" data set which is multi-class classification problem.

With respect to execution time, SVM performed very well with execution time under *1 second* for Wine quality dataset. But, it didn't perform well on Creditcard defaults dataset.

These experiments indicate that performance of a learning algorithm varies , based on the data set type, size as well as hyper parameters used.

References

[Giudici, 2001] Giudici, P. (2001). Bayesian data mining, with application to benchmarking and credit scoring. *Applied Stochastic Models in Business and Industry*, 17(1):69–81.