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 1599 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 dataset , including 30000 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

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.

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

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 pruning arguments applied as : *max_dept = 6* and *min_samples_leaf = 3*.

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.

Result was produced with **Accuracy = 60.83%**

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

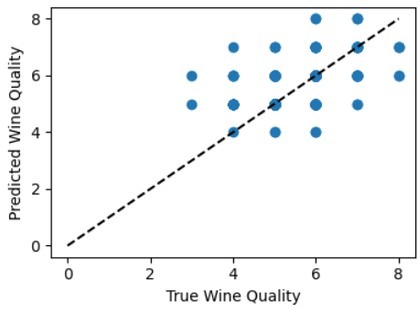


Figure 1: Wine Quality Prediction

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 arguments applied as : *max_dept = 6* and *min_samples_leaf = 3*.

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 for if clients will default credit card or not.

Result was produced with **Accuracy = 81.61%**

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

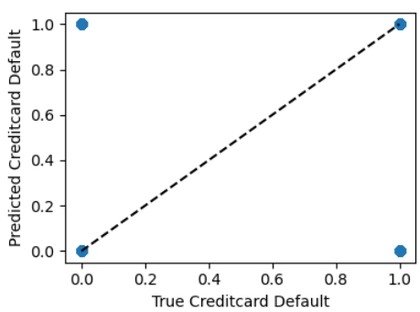


Figure 2: Credit Card Default Prediction

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 max depth is set to 2 and learning rate to 0.2.

2.2.1 Classification of Wine Quality dataset

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

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.

Result was produced with **Accuracy = 62.50%**

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

Predicted Actual	3	4	5	6	7	8	All
3	0	1	1	0	0	0	2
4	1	1	6	8	0	0	16
5	1	3	145	63	2	0	214
6	1	1	47	126	9	1	185
7	0	0	0	30	28	0	58
8	0	0	0	3	2	0	5
All	3	6	199	230	41	1	480

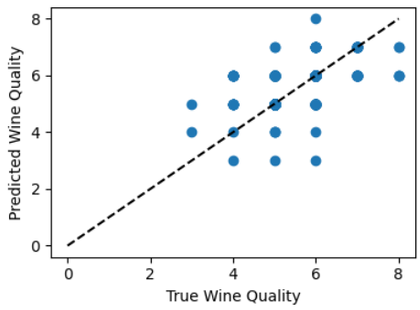


Figure 3: Wine Quality Prediction

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*.

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 for if clients will default credit card or not.

Result was produced with **Accuracy = 82.12%**

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

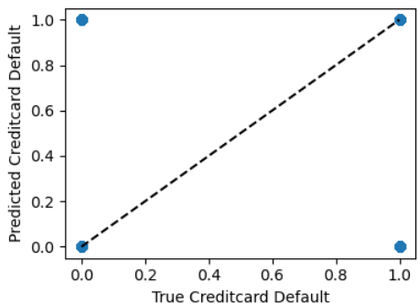


Figure 4: Credit Card Default Prediction

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, with solver used 'adam' and activation method 'tanh' is used.

'adam' - is a stochastic gradient-based optimizer 'tanh' - the hyperbolic tan function, returns f(x) = tanh(x)

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.

Result was produced with **Accuracy = 61.88%**

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

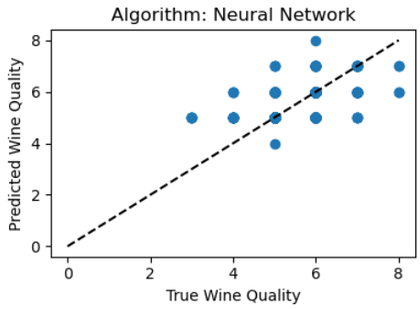


Figure 5: Wine Quality Prediction

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 solver used 'adam' and activation method 'tanh' is used.

'adam' - is a stochastic gradient-based optimizer 'tanh' - the hyperbolic tan function, returns f(x) = tanh(x)

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 for if clients will default credit card or not.

Result was produced with **Accuracy = 77.42%**

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

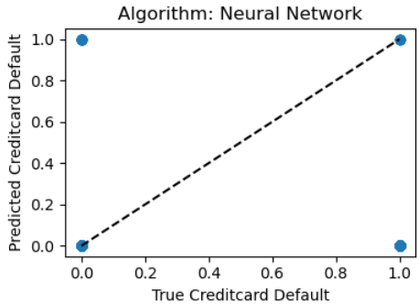


Figure 6: Credit Card Default Prediction

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()).

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.

For kernel='rbf'

Result produced with **Accuracy = 50.00%**

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

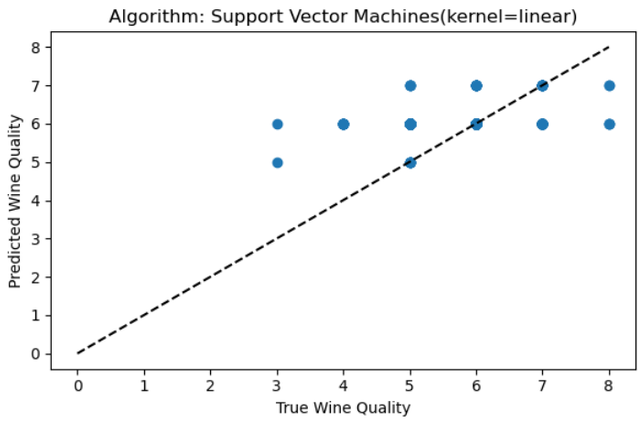
Predicted Actual	5	6	7	All
3	0	2	0	2
4	4	12	0	16
5	83	131	0	214
6	29	156	0	185
7	4	53	1	58
8	0	5	0	5
All	120	359	1	480

For kernel='linear'

Result was produced with **Accuracy = 40.21%**

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



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()).

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 for if clients will default credit card or not.

For kernel='rbf'

Result was produced with **Accuracy = 78.36%**

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

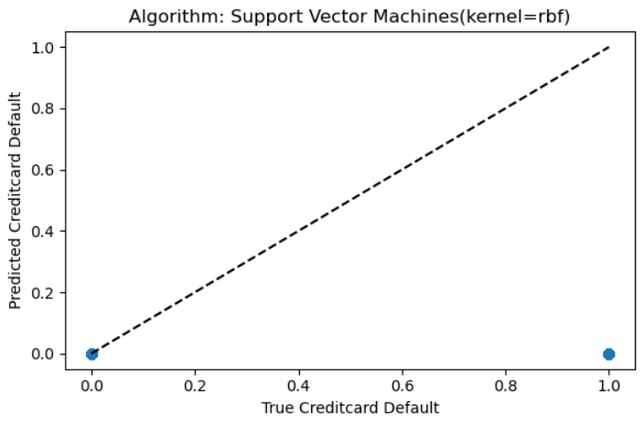


Figure 7: Credit Card Default Prediction

For kernel='linear'
Result was produced with **Accuracy = 52.36%**
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

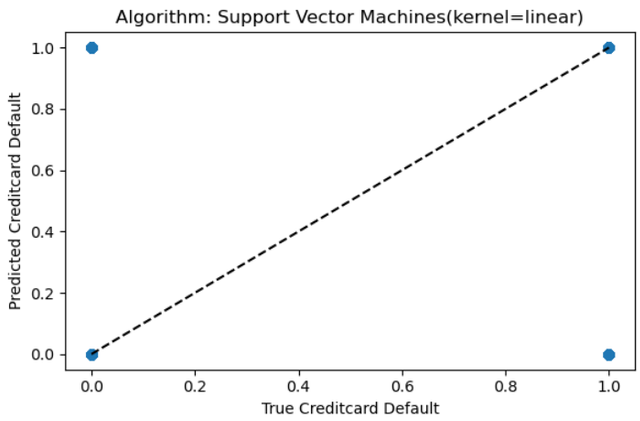


Figure 8: Credit Card Default Prediction

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 : $k = 6$ and $k = 10$.

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.

For k=6
Result was produced with **Accuracy of KNN(k=6) = 52.08%**
Here is comparison of actual wine ratings vs predicted from the test set.

Predicted Actual	3	4	5	6	7	8	All
3	0	0	2	0	0	0	2
4	0	1	8	6	1	0	16
5	0	3	141	63	7	0	214
6	0	1	73	92	18	1	185
7	1	1	12	28	16	0	58
8	0	0	0	5	0	0	5
All	1	6	236	194	42	1	480

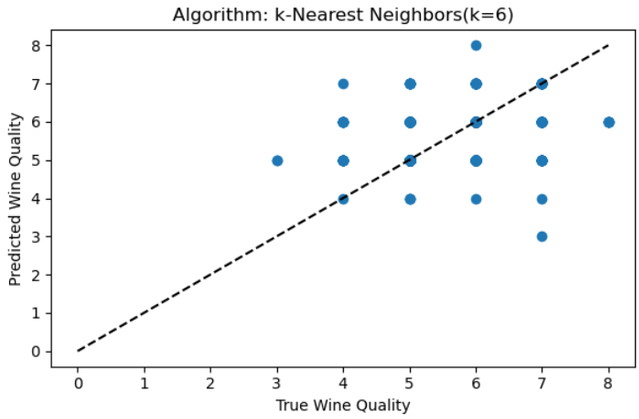


Figure 9: Wine Quality Prediction

For k=10 Result was produced with **Accuracy of KNN(k=10) = 50.00%**
Here is comparison of actual wine ratings vs predicted from the test set.

Predicted Actual	4	5	6	7	All
3	0	2	0	0	2
4	0	9	7	0	16
5	2	136	73	3	214
6	0	80	91	14	185
7	0	13	32	13	58
8	0	1	3	1	5
All	2	241	206	31	480

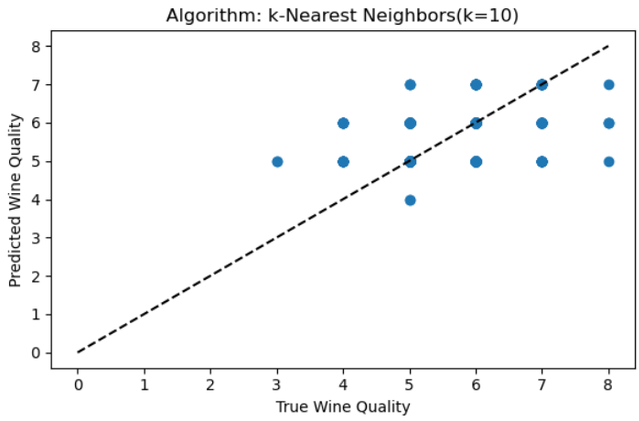


Figure 10: Wine Quality Prediction

2.5.2 Classification of Defaults for Credit Card dataset

KNN algorithm is applied on ***winequality-red*** data set, with neighbours size : $k = 6$ and $k = 7$.

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 for if clients will default credit card or not.

For k=6 Result was produced with **Accuracy of KNN(k=6) = 77.14%**
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

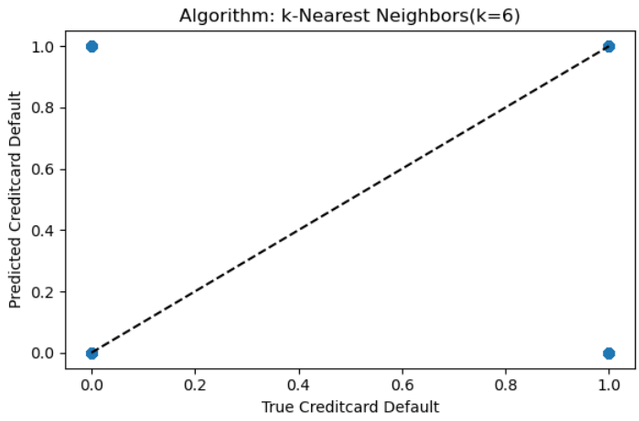


Figure 11: Credit Card Default Prediction

For k=7 Result was produced with **Accuracy of KNN(k=7) = 76.07%**
Here is comparison of actual values for defaulted vs predicted from the test set.

Predicted Actual	0	1	All
0	6556	496	7052
1	1658	290	1948
All	8214	786	9000

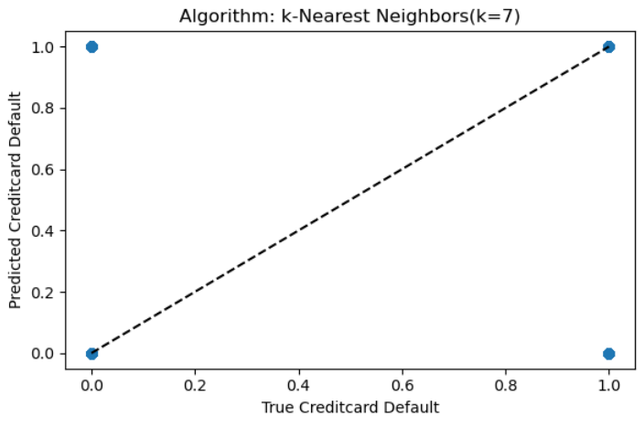


Figure 12: Credit Card Default Prediction

3 Conclusion

Based on the results of applying various learning algorithms for two selected classification problems, *Decision Tree with Gradient Boost* performed fairly better. Also, in general all the algorithms performed better 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.

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.