DAR/IDAR Coursework 2

- Please submit ONE file to the coursework 2 answer submission portal on moodle: one of the following (.pdf/.html/.doc) files created by RStudio. Please include any R code, plots or results.
- Your files should be named as follows:
  MSc/BSc_CW2_xxxxxxxx_initial_lastname.pdf (.html/.doc)
  where xxxxxxxx is your student ID. For instance, MSc/BSc_CW2_12345678_Wan.pdf.
- Don't forget to write down your programme (MSc or BSc), name and student ID on the first page of your answer sheets as well.
- Each question below has two weightings. The first weighting is for MSc students and the second weighting is for BSc students.

1. Bayesian networks and naïve Bayes classifiers. (20% | 25%)

Marking scheme: - MSc: 5.0% each. - BSc: (a), (c) 7.5% each; (b), (d) 5.0% each.

(a) Given a training dataset including 30 observations and a Bayesian network indicating the relationships between 3 features (i.e. Income, Student and Credit Rate), and the class attribute (i.e. Buy Computer), please create the conditional probability tables by hand.

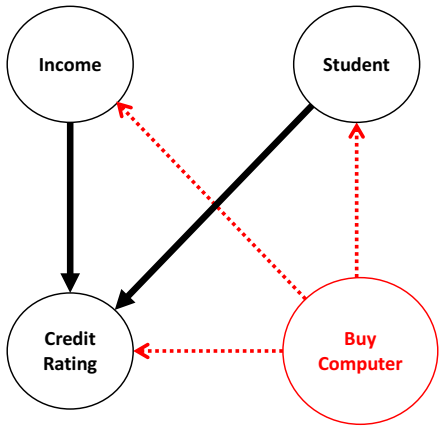
(b) Make predictions for 2 testing observations by using a Bayesian network classifier.

(c) Based on the conditional independence assumption between features, please create the conditional probability tables by hand.

(d) Make predictions for 2 testing observations by using a naïve Bayes classifier.

Training Observations	Income	Student	Credit Rating	Buy Computer
Observation_1	High	True	Fair	No
Observation_2	Low	False	Excellent	No
Observation_3	Low	True	Fair	No
Observation_4	High	False	Fair	No
Observation_5	Low	True	Excellent	Yes
Observation_6	High	False	Fair	Yes
Observation_7	High	True	Excellent	Yes
Observation_8	Low	True	Fair	No
Observation_9	Low	False	Excellent	Yes
Observation_10	Low	True	Excellent	No
Observation_11	High	True	Fair	No
Observation_12	Low	False	Fair	No
Observation_13	Low	True	Fair	No
Observation_14	High	False	Excellent	No
Observation_15	Low	True	Fair	Yes
Observation_16	High	False	Excellent	Yes
Observation_17	High	True	Excellent	No
Observation_18	Low	True	Fair	No
Observation_19	Low	False	Excellent	Yes
Observation_20	Low	True	Excellent	No
Observation_21	High	False	Excellent	Yes
Observation_21	Low	True	Excellent	Yes
Observation_23	High	False	Excellent	No
Observation_24	High	True	Fair	No
Observation_25	Low	False	Fair	Yes
Observation_26	Low	True	Fair	No
Observation_27	Low	True	Fair	No
Observation_28	Low	True	Fair	Yes
Observation_29	Low	False	Fair	No
Observation_30	High	True	Fair	Yes

Testing Observations	Income	Student	Credit Rating	Buy Computer
Observation_31	Low	True	Excellent	?
Observation_32	High	True	Fair	?



2. Predicting room occupancy by using decision tree and random forests classification algorithms. **(20% | 25%)**

Marking scheme: - MSc: 5.0% each. - BSc: (a), (c) 5.0% each; (b), (d) 7.5% each.

(a) Load the room occupancy training and testing datasets that are also used for the 1st coursework. Train a decision tree classifier and evaluate the predictive performance by reporting the classification accuracy obtained on the testing dataset.

(b) Output and analyse the tree learned by the decision tree algorithm, i.e. plot the tree structure and make a discussion about it.

(c) Train a random forests classifier and evaluate the predictive performance by reporting the classification accuracy obtained on the testing dataset. Define set.seed(1).

(d) Output and analyse the feature importance obtained by the random forests classifier.

3. Predicting wine quality by using support vector machine classification algorithm. **(20% | 25%)**

Marking scheme: - MSc: 4.0% each. - BSc: 5.0% each.

(a) Download the full wine quality training and testing datasets from Moodle, and use the training dataset to find out the optimal value of hyperparameter C for a linear kernel-based svm.

(b) Train a svm classifier by using the linear kernel and the corresponding optimal value of hyperparameter C, then make predictions on the testing dataset, report the classification accuracy.

(c) Use the training dataset to find out the optimal values of hyperparameters C and γ for an RBF kernel-based svm.

(d) Train a svm classifier by using the RBF kernel and the corresponding optimal values of hyperparameters C and γ, then make predictions on the testing dataset, report the classification accuracy.

(e) Train a logistic regression model. Then use the testing dataset to conduct an ROC curve analysis to compare the predictive performance of the trained logistic regression model and those two svm classifiers trained by using linear and RBF kernels respectively.

Hint: Define set.seed(1). Given a pre-defined hyperparameter space - C: [0.01, 0.1, 1, 10, 100], and γ: [0.01, 0.1, 1, 10, 100].

4. Hierarchical clustering **(20% | 25%)**

Marking scheme: - MSc: 5.0% each. - BSc: (a-c) 7.0% each; (d) 4.0%.

Consider the `USArrests` data. We will now perform hierarchical clustering on the states.

(a) Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states.

(b) Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters?

(c) Hierarchically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one.

(d) What effect does scaling the variables have on the hierarchical clustering obtained? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed? Provide a justification for your answer.

5. PCA and K-Means Clustering **(20% | 0%)**

Marking scheme: - MSc: (a) 2.0%, (b-c) 4.0% each, (d-g) 2.5% each.

In this problem, you will generate simulated data, and then perform PCA and K-means clustering on the data.

(a) Generate a simulated data set with 20 observations in each of three classes (i.e. 60 observations total), and 50 variables.

Hint: There are a number of functions in `R` that you can use to generate data. One example is the `rnorm()` function; `runif()` is another option. Be sure to add a mean shift to the observations in each class so that there are three distinct classes.

(b) Perform PCA on the 60 observations and plot the first two principal components' eigenvector. Use a different color to indicate the observations in each of the three classes. If the three classes appear separated in this plot, then continue on to part (c). If not, then return to part (a) and modify the simulation so that there is greater separation between the three classes. Do not continue to part (c) until the three classes show at least some separation in the first two principal component eigenvectors.

(c) Perform K-means clustering of the observations with K = 3. How well do the clusters that you obtained in K-means clustering compare to the true class labels?

Hint: You can use the `table()` function in `R` to compare the true class labels to the class labels obtained by clustering. Be careful how you interpret the results: K-means clustering will arbitrarily number the clusters, so you cannot simply check whether the true class labels and clustering labels are the same.

(d) Perform K-means clustering with K = 2. Describe your results.

(e) Now perform K-means clustering with K = 4, and describe your results.

(f) Now perform K-means clustering with K = 3 on the first two principal components, rather than on the raw data. That is, perform K-means clustering on the 60×2 matrix of which the first column is the first principal component's corresponding eigenvector, and the second column is the second principal component's corresponding eigenvector. Comment on the results.

(g) Using the scale() function, perform K-means clustering with K = 3 on the data after scaling each variable to have standard deviation one. How do these results compare to the true class labels? Will the scaling affect the clustering?