code (using the format: Courier New), results and figures to the Word document.

We will mark your submission mostly using your word document. However, you need to make sure your R markdown file is executable in case we need to check your code.

Task A: Modelling - Classification

Build a model using the data in LoanData (The data comes from The Lending Club).

Data Set Information: The data LoanData_train.csv and LoanData_test.csv is on loans given by The Lending Club. A lot of information on these loans is publicly available. We have taken 10,000 of these loans and simplified the data to 11 variables, (you can download the original data at https://www.lendingclub.com/info/download-data.action):

- loan_amnt: The amount of money loaned (in US dollars).
- term: The length of time of the loan (in months)
- int_rate: The interest rate.
- installment: The amount due to be paid in each installment (in US dollars)
- grade: An assessment of the 'quality' of the loan (based on some estimate of the likelihood that the loan will be repaid, with A being a loan most likely to be repaid and F being the least likely to be repaid).
- home_ownership: A variable indicating whether the borrower owns their home, has a mortgage or rents.
- annual_inc: The income of the borrower (in US dolloars)

- verification: Whether the income was verified by The Lending Club

- loan_status: Whether the loan was repaid in full "Fully Paid", or the loan was not repaid "Charged Off".
- delinq_2yrs: The number of times in the past 2 years that the borrower has been more than 30 days late in the payment of a debt.
- pub_rec: The number of derogatory records of the borrower.

A.1 We wish to use this data to build a model in order predict whether a loan will be repaid (`loan_status`) based on the other variables. Which technique from the lectures is most appropriate for this problem: linear regression, logistic regression? Explain why your technique is most appropriate by explaining why your chosen technique will work and why the other technique will not. (You may want to generate a summary of the data before answering this question).

A.2 Consider the grade variable. Note that R treats it as a categorical variable but we could change grade into a numeric variable using the R code `Loan$grade<-as.numeric(Loan$grade)`. Do you think this will improve the model you will achieve? Why or why not? If you think that this will improve predictive accuracy make the change. (Note, you are not expected to analyse the data for this question, just base this on which assumptions you think are more reasonable for this data).

A.3 Train a model using your chosen technique on the data in LoanData_train using all the predictors. Print the summary of your model. Explain the meaning of the "Estimate" and "Std. Error" columns in the Coefficients table. Why are there more rows in the coefficients table than there are variables in the data?

A.4 Load the test data Loandata_test.csv into a data frame called `loan_test`. You can use the R code

$$\texttt{prob} = \texttt{predict}(\texttt{model}, \texttt{loan}_\texttt{t}\texttt{est}, \texttt{type} =' \texttt{response}')$$

to generate probabilities from your model. By investigating the signs of the coefficients of the predictors in the model you trained, or otherwise, determine whether these are probabilities that the loan is paid (`loan_status= "Fully Paid"`) or that the loan is defaulted on (`loan_status="Charged Off"`).

A.5 Now convert the probabilities calculated in the previous question to estimates of the loan_status . For example if the probabilities are probabilities that the loan is defaulted on then you can convert estimates above 0.5 to "Charged Off" and below to "Fully Paid", by first creating a logical vector indicating whether `prob` is greater than half `pred<- prob>0.5`, then converting to a factor using `as.factor()` and changing the names for the levels using `levels()`. What proportion of the test data does the model you trained predict correctly?

A.6 (Optional) Now consider a model which simply predicts that all loans will be fully paid. What proportion of the test data does this model predict accurately? If the simpler model predicts whether loans will be paid correctly on the test data, more often than the more complicated model, does that mean we should prefer the simpler model? Why, why not? (Hint: consider other possible measures of success for classifiers).

Task B: Modelling - Regression

In this task you will use the auto_mpg data to build a linear regression model in order to predict a car's fuel efficiency based on its other characteristics: "The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes." (Quinlan, 1993)

1. mpg: continuous, target variable
2. cylinders: multi-valued discrete
3. displacement: continuous
4. horsepower: continuous
5. weight: continuous
6. acceleration: continuous
7. model year: multi-valued discrete
8. origin: multi-valued discrete
9. car name: string (unique for each instance)

More details of the dataset is available at
`https://archive.ics.uci.edu/ml/datasets/Auto+MPG` .

Tasks

B.1: (Optional) In auto_mpg_train.csv there are some missing values listed as "?". Describe your strategy for treating missing values and update (edit by hand) the file accordingly.

B.2: *If you did B.1 use your modified auto_mpg_train.csv for the remaining questions. Otherwise use the auto_mpg_clean_train.csv data for the remainder of this section.*
Pair plot mpg vs. the other variables to visualise the relationships (note that `plot()` does this by default when given a dataframe). Based on your pair plots, discuss which variables would be good to include in a multiple linear regression model to predict `mpg`. On this basis propose on initial set of predictors to use for a multiple linear regression to predict `mpg`.

B.3: With predictors of your choice build the model using the `lm()` routine in R, and then print the summary of the model to get the R diagnostics. Briefly explain the R^2 value, and p-value statistics in the summary. What do these imply about the predictors for your model?

B.4: Test the fitted model using the "auto_mpg_test.csv", and calculate the MSE on the test set, reporting it. Note the test set has no missing values.

B.5: Now make one change to your model in an attempt to improve your model. You might try removing a predictor you think might be a cause of overfitting, or adding a predictor which is the ratio or product of good predictors which you also expect might interact with each other. Before training your new model describe the change you wish to make and explain why you expect it to perform better than your previous model. Train your new model on the training set and calculate the MSE on the test. Did it perform better or worse than your previous model?

Task C: Sampling

C.1: Pick one of the probability density functions $p(x)$ below:

$$p(x) = \frac{1}{1-e^{-4}} 2e^{-2x} \qquad \text{for } x \in [0,2]$$
$$p(x) = 2e^{-2x} \qquad \text{for } x \geq 0$$

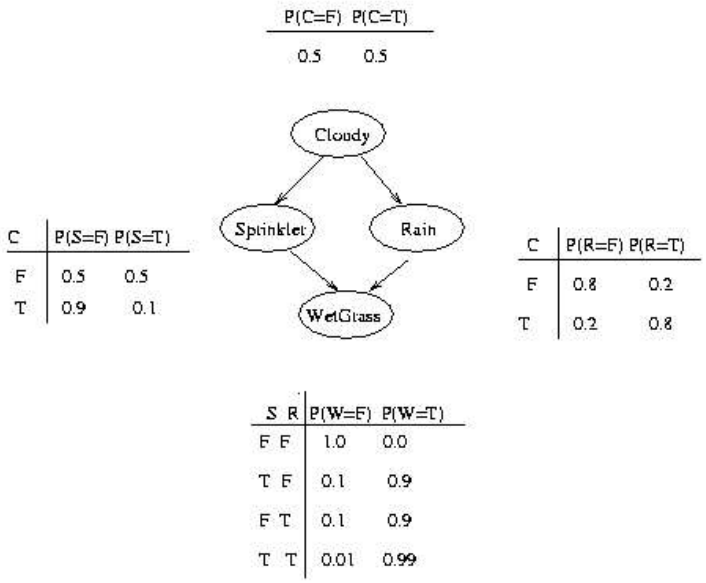

Figure 1: Simple Bayesian Network

In R, develop a function to generate samples from your chosen probability distribution using runif() as the only builtin source of (pseudo) random numbers. In your report explain whether you use rejection or inverse sampling. Using your function sample 200 values from the distribution and store them in a vector. Then using the `hist()` with `breaks=10` plot a histogram of your samples.

C.2 Consider the simple Bayesian network of Figure 1, representing the relationship between four variables (Note T and F in the tables represent 'true' and 'false'):

1. Cloudy: indicating whether the weather is cloudy on a given day

2. Rain: Indicating whether it rains on a given day

3. Sprinkler: Indicating whether the sprinklers are turned on a given day

4. WetGrass: Indicating whether the grass is wet on a given day.

Now briefly answer the following questions

1. Write the joint probability distribution

$$p(cloudy, sprinkler, rain, wetgrass) = p(C = cloudy, S = sprinkler, R = rain, W = wetgrass)$$

as a product of probabilities in the tables in the diagram. (Note that we are using $cloudy, sprinkler, rain$ and $wetgrass$ as variables which can be either T or F).

2. On this model which variables are independent of Rain? Explain your reasoning.

C.3 In R construct a function `p_w_given_scr` which, when given the values of the Boolean variables S,C,R and W returns the conditional probability of the value of W given S,C,R. For example

$$\texttt{p\_s\_given\_crw}(\texttt{S} = \texttt{TRUE}, \texttt{C} = \texttt{FALSE}, \texttt{R} = \texttt{TRUE}, \texttt{W} = \texttt{FALSE})$$

should return the value of $p(S = T|C = F, R = T, W = F)$. Use your function to calculate $p(S = T|C = F, R = T, W = F)$.

In order to do this question it helps to have the the probability tables given in figure loaded into R. For this you can use the code:

$$\begin{aligned}
\texttt{cpt_c} &= c(0.5, 0.5) \\
\texttt{cpt_s_given_c} &= matrix(c(0.5, 0.5, 0.9, 0.1), 2, 2, byrow = F) \\
\texttt{cpt_r_given_c} &= matrix(c(0.8, 0.2, 0.2, 0.8), 2, 2, byrow = F) \\
\texttt{cpt_w_given_sr} &= matrix(c(1, 0.1, 0.1, 0.01, 0, 0.9, 0.9, 0.99), 2, 4, byrow = T)
\end{aligned}$$

So, for instance `cpt_s_given_c[1,1]` is $p(S = F|C = F)$, while `cpt_s_given_c[2,1]` is $p(S = T|C = F)$. (Note that the rows of the tables in Figure 1 are the columns in the R matrices).

Hint: If we were to do this for the Cloudy variable we would first recall from lectures that

$$p(cloudy|sprinkler, rain, wetgrass) \propto p(cloudy)p(rain|cloudy)p(sprinkler|cloudy)$$

To calculate the actual probability we have to normalise by dividing by the sum of probabilities over all possible values of C, in this case:
$p(C = T)p(R = rain|C = T)p(S = sprinkler|C = T)$
$+ p(C = F)p(R = rain|C = T)p(S = sprinkler|C = T)$

C.4 Describe how we could use Gibbs sampling to estimate $p(C = T|W = T)$. Which probability distributions would we need to sample from (note: you don't need to describe how to calculate these)? What do you do with the values you have sampled?

C.5 (Optional) Create a Gibbs sampler to estimate $p(C = T|W = T)$. In order to generate your estimate, run the sampler for 1000 cycles throw away the first 100 samples, and then take every 10th sample to generate your estimate (if you do this question and provide comments in your code explaining what you are doing, this will also serve as an answer to question C.4).