Assessment 1 - Linear Regression

In this assessment, you need to answer all the questions about KNN, Linear Regression, Regularization, Logistic Regression, K-fold cross-validation, and other concepts covered in Module 1-3. R studio is recommended to use to complete your assessment. All codes need comments to help markers to understand your idea. If no comment is given, you may have a 10% redundancy on your mark. Please refer to weekly activities as examples for how to write comments. After you have answered all the questions, please knit your R notebook file to HTML or PDF format. Submit both .rmd file and .html or .pdf file to assessment 1 dropbox via the link on the Assessment page. You can compress your files into a zip file for submission. The total mark of this assessment is 100, which worths 30% of your final result.

hint: Please review all reading materials in Module 1-3 carefully, especially the activities.

Question 1 - KNN (20 marks)

In this question you are required to implement a KNN classifier to predict the class of iris plants. The well-known iris dataset is used in this question. Detailed description of this data set can be found at https://archive.ics.uci.edu/ml/datasets/iris.

Specifically, you need to:

1. Split the data set into a training and a test set with the ratio of 7:3. (1 mark)
2. Implement a KNN classifier. (5 marks)
3. Investigate the impact of different K (from 1 to 6) values on the model performance (ACC) and the impact of different distance measurements (euclidean, manhattan, canberra, and minkowski) on the model performance (ACC). Visualize and discuss your findings. (14 marks)

```
library(datasets)
data(iris)
```

```
# start your answer here ...
```

Question 2 - Linear Regression (35 marks)

In this question you need to implement a linear regression model to predict health care cost. The data set used in this question can be found in 'insurance.csv'. The data set has 7 features, which are summarized as below.

- Age: insurance contractor age, years
- Sex: insurance contractor gender, [female, male]
- BMI: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9
- Children: number of children covered by health insurance / Number of dependents
- Smoker: smoking, [yes, no]
- Region: the beneficiary's residential area in the US, [northeast, southeast, southwest, northwest]
- Charges: Individual medical costs billed by health insurance, $ #predicted value

Specifically, you need to:

1. Perform data pre-processing, including removing invalid data and transfromatting the categorical features to numerical features. (4 marks)

2. Split the data set into a training set and a test set, with ratio of 7:3. (2 mark)
3. Implement a linear regression model and train the model with your training data. Visualize the parameter updating process, test error (RMSE) in each iteration, and cost convergence process. Please be advised that built-in models in any realeased R package, like glm, is not allowed to use in this question. You can choose your preferred learning rate and determine the best iteration number. (8 marks)
4. Evaluate your model by calculating the RMSE, and visualizing the residuals of test data. Please note that explanation of your residual plot is needed. (5 marks)
5. Does your model overfit? Which features do you think are not significant? Please justify your answers. For example, you can analyze the significance of a feature from correlation, variance, etc. (8 marks)
6. Use the glmnet library to biult two linear regression models with Lasso and Ridge regularization, respectively. In comparison to your model, how well do these two models perform? Do the regularized models automatically filter out the less significant features? What are the differences of these two models? Please justify your answers. (8 marks)

```
data = read.csv('insurance.csv')
```

```
# start your answer here ...
```

Question 3 - Logistic Regression (45 marks)

In this question, you are required to implement a Logistic Regression model to classify whether a person donated blood at a Blood Transfusion Service Center in March 2007. Please read the sub-questions below carefully for the deteailed instructions.

1. Check out the Blood Transfusion Service Center Data Set at https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center.
2. Perform data preprocessing to determine and remove invalid samples. Split the data into a training set and a test set with a ratio of 7:3. (2 marks)
3. Develop a Logistic Regression model that use batch gradient descent for optimization. Visualize the parameter updating process, test error (ACC) in each iteration, and the cost convergence process. Please note that you need to develop your model step-by-step, built-in models in any realeased R package, like glm, is not allowed to use in this question. (10 marks)
4. Invesitigate the influence of different learning rate to the training process and answer what happend if you apply a too small or a too large learning rate. (5 marks)
5. Expermently compare batch gradient descent and stochastic gradient descent and discuss your findings (e.g., convergence speed). Visualize the comparison in terms of updating process and the cost convergence process. (6 marks)
6. Develop a K-fold (K = 5) cross validation to evaluate your model in step 3. Please note that you need to write R codes to explicitly show how you perform the K-fold cross validation. Built-in validation methods are not allowed to use. Different metrics, e.g. ACC, Recall, precision, etc, should be used to evaluate your model. (8 marks)
7. Use different values of K (from 5 to N, where N denotes the sample number) and summarize the corresponding changes of your model performances. Visualize and explain the changes. (6 marks)
8. How can you modify the cost function to prevent overfitting? Discuss the possibility of adding regularization term(s) and summarize the possible changes in the gradient descent process. (8 marks)

```
# start your answer here ...
```