COMP1433: Introduction to Data Analytics

Assignment

PolyU Spring 2024

Answers Submission Due: 23:59, Apr 26, 2024.

Important Notes.

- This is an *individual* assessment. Therefore, any form of discussion or collaboration with classmates is not allowed. Furthermore, the use of GenAI tools is strictly prohibited for this assignment. Any instances of such behavior being identified will result in a zero mark being assigned.
- Following the syllabus, only the R language is allowed for answering the programming questions.
- Please submit the compressed folder (in zip or rar) with all the answers. Please also name the folder with your student ID, such as "21123456D.zip" or "21123456D.rar".
- In the compressed folder, for non-coding questions (i.e., Question 2(c)), please put your detailed problem-solving steps into a report in PDF format, or you can choose to submit a scanned copy of your handwritten solution. For the coding questions with the index i below, you may create a subfolder named as "Qi" (e.g., Q1), which contains the codes for Q_i and the readme.txt file to explain how to run the code.
- The codes should be well commented for easy reading, and indicate clearly in the comments which part is for which sub-question (if any). It is for the case that the implementation is imperfect (with bugs) and we need to somehow find scores from the codes to see if your solutions are developed in a correct way.
- The compressed folder should be submitted to the *blackboard*.[1] The full mark is 100' and submission entry is: Assessments/Assignment. For Ques-

[1] `learn.polyu.edu.hk`

tion 1 and 3, we have provided the input data in the form of the compressed folder "Assignment_Data" available in the same entry.

- No late submission is allowed and don't forget to double check if the submission is saved successfully before leaving.
- When handling the paths for file loading and saving, please use relative paths for the TAs to run your codes easily in a different environment. It can be assumed that the input data file is stored in the same folder as the codes used to read that data.
- It is assumed that the external library "ggplot2" have already been installed in the R system. For any other external libraries you need to use, please indicate them in the readme.txt file.
- Last but not least, best of the luck for this assignment! :)

Question 1. K-means Clustering [**Coding Question**] In Lecture 3, we have learned the K-means clustering algorithm. In this question, you are required to implement this algorithm from scratch (i.e., without using external packages or the built-in kmeans() function in R) and use it to cluster the samples in the "Customer" dataset. The dataset is stored in the file "Customer.csv", which can be downloaded from the blackboard. Each sample in this dataset comprises 5 attributes: CustomerID, Gender, Age, Annual_Income, and Spending_Score. You are expected to cluster all the data samples into 5 clusters based on the attributes "Annual_Income" and "Spending_Score". At each training iteration, it is required to calculate and record the mean distance of data points to their respective cluster centroid (i.e., $J^{cluster}$ given in Lecture 3 pg 34).

After obtaining the clustering results, it is required to generate figures to visualize the results. In the visualization, you are first required to draw a scatter plot for all the data samples (the x-axis corresponds to the "Annual_Income" while the y-axis corresponds to the "Spending_Score"), and color each sample in red, green, blue, purple, and brown indicating the cluster it has been assigned to. Then, you are required to draw another line plot with the x-axis corresponding to the training iteration, and the y-axis corresponding to the mean distance to the cluster centroids at that iteration.

Please note that no external package or library should be used except "ggplot2" for data visualization. For the initialization, the initial cluster centroids of five classes are (5, 5), (20, 90), (100, 10), (70, 50), and (120, 100), respectively. Please run your K-means clustering algorithm for 100 iterations (early stopping is acceptable if the algorithm converges). (30')

Question 2. Monte Carlo Simulation The log-normal distribution, a continuous probability distribution in probability theory, characterizes a random variable whose logarithm conforms to a normal distribution. Hence, when a random variable X is log-normally distributed, its natural logarithm, denoted as $Y = \ln(X)$, follows a normal distribution. (40')

(a) [**Coding Question**] Apply Monte Carlo simulation techniques to simulate a log-normal distribution with mean $\mu = 0.02$ and standard deviation $\sigma = 0.05$. You are required to simulate 10,000 samples with a random seed of 1234. Please note that the built-in function *rlnorm()* is not allowed to be used in this part. Instead, you should first sample elements from a normal distribution and then apply the exponential transformation to them. Visualize the distribution of all the sampled data points using a histogram. (10')

(b) [**Coding Question**] Create a density plot for the log-normal distribution described in question (a), where $\mu = 0.02$ and $\sigma = 0.05$. To obtain the probability density, you can use the built-in function *dlnorm()*. Combine the density plot with the histogram obtained from question (a) and display both of them in the same figure. (10')

(c) [**Non-coding Question**] Derive the theoretical mean and variance of the random variable X, which is log-normally distributed with mean $\mu = 0.02$ and standard deviation $\sigma = 0.05$. You are required to provide the full derivations step-by-step. Compare these theoretical values with the empirical mean and variance that you obtained from the simulation in question (a), with the total number of samples = 10, 100, 1,000, 10,000, and 100,000. Note that you are allowed to submit a scanned copy of your handwritten solution for this question. (10')

(d) [**Coding Question**] Suppose the daily return of a stock follows a log-normal distribution with $\mu = 0.02$ and $\sigma = 0.05$. Apply Monte Carlo simulation to estimate the probability that the stock price will increase by at least 0.05 in a single trading day. (5')

(e) [**Coding Question**] Please use the R built-in functions to obtain the answer to question (d), i.e., what is the probability that the stock price will increase by at least 0.05 in a single trading day? (5')

Question 3. Linear Regression [**Coding Question**] As a data scientist specializing in the real estate market, you have been given the responsibility of working with the dataset named "house_prices_dataset.csv". This dataset comprises various property features, including house area (House_Area), distance to the city center (Distance_to_Center), house age (House_Age), and house condition (House_Condition). In this question, your objective is to build linear regression models that can predict house prices based on these property features. Furthermore, you will conduct analysis to gain insights into the impact of these property

features on house prices. (30')

(a) **Simple Linear Regression Analysis:** In this part, your task is to analyze the impact of each individual property feature (House_Area, Distance_to_Center, House_Age, House_Condition) on the house price (House_Price). To begin, load the dataset named "house_prices_dataset.csv" and use the lm() function to construct four simple linear regression models, treating one property feature as the independent variable and house price as the dependent variable. After constructing the model, you can use the built-in functions summary(), attributes(), and coefficients() to examine various attributes associated with the fitted model. To visualize the relationship between each property feature and house price, create four scatter plots where the x-axis represents the property feature, and the y-axis represents the house price. Additionally, include a line on each plot that represents the obtained simple linear regression model. Atop the line, add a text description for the obtained simple linear regression model (e.g., House_Price = 1982.8 * House_Area + 31499.5). (15')

(b) **Multiple Linear Regression Analysis:** In this part, to enhance the performance of the simple linear regression model you have established earlier, you consider both the House_Area and Distance_to_Center features as independent variables and build a new multiple linear regression model. Following the same procedures in (a) to load the dataset and use the lm() function to construct a multiple linear regression model. Based on the fitted model, please predict the value of the house price when the House_Area is equal to 250 and the Distance_to_Center is equal to 5 using the predict() function. Print your result to the screen. (5')

(c) **Optimal Feature Set for House Price Prediction:** In addition to House_Area and Distance_to_Center features, which were considered in part (b), House_Condition and House_Age are also significant factors influencing house prices. In this part, the aim is to construct and compare multiple linear regression models using different combinations of these features:

- **Model 1.** House_Area and Distance_to_Center;
- **Model 2.** House_Area and House_Age;
- **Model 3.** House_Area and House_Condition;
- **Model 4.** Distance_to_Center and House_Age;
- **Model 5.** Distance_to_Center and House_Condition;
- **Model 6.** House_Age and House_Condition;

To accomplish this, fit the models using the lm() function and evaluate their performance based on the R-squared (R^2) value. After fitting the models, print the

obtained R-squared values on the screen and the ID of the best model using the following format:

"The R-squared values for all the models are aaa, bbb, ccc, ddd, eee, fff. So the best model is Model X" (10')