Machine Learning and Deep Learning I Homework 2

Instructor: Joonseok Lee

Deadline: 2024/4/30 Tue, 18:00

Instruction

- No unapproved extension of deadline is allowed. Late submission will result in 0 credit.
- Optimize your code as much as you can. We do not guarantee to run unreasonably inefficient codes for grading. Remember, vectorization is important for efficient computation!
- You will be given skeleton files for doing your assignment. Detailed instructions are given in the comments below each method to complete. Please read them carefully before jumping into implementation!
- Each assignment is built and tested under Google Colaboratory. If you work on a local machine, you need to handle version issue on your own.
- Explicitly mention your collaborators or reference (*e.g.*, website) if any. If we detect a copied code without reference, it will be treated as a serious violation of student code of conduct.

What to Submit

Please upload a single zip file named with your student ID (e.g., `2024-00000.zip`) on eTL, containing

- Your complete `hw2_kmeans.ipynb`, `hw2_classification.ipynb` files

Please keep in mind...

- Please erase any unnecessary print codes or comments that you have wrote before submission. This may affect your grade.
- Make sure you run **all** the code cells before submitting the file. We do not guarantee to run the codes that are not executed.

1 k-Means Clustering - Image Compression [50 pts]

Follow the instructions below to complete `hw2_kmeans.ipynb` provided on the ETL.

You will explore the application of k-means clustering in the field of image compression. By applying k-means clustering to image data, we can effectively reduce the number of colors used to represent the image, thus achieving compression while preserving its visual quality.

In this assignment, we will use `img.jpeg` provided on the ETL for image compression. Please read the instructions in `hw2_kmeans.ipynb` carefully for implementations.

It is not allowed to use any libraries such as scikit-learn to directly solve this problem.

(a) Implement `assign_cluster(self, centroid_list)` by calculating the distances between each pixel and centroid and assigning each pixel to the closest centroid. [15 pts]

(b) Implement `update_cluster(self, label)` by updating the centroids. [5 pts]

(c) Implement `calculate_J(self, distances)` by calculating the total sum of distances between each pixel and its assigned centroid. [5 pts]

(d) Implement `kmeans_imgcomp(self)` by updating the centroids for `max_iter` times. In this assignment, we will use `max_iter` times instead of iterating until the convergence. [10 pts]
(No initialization of centroids needed.)

(e) Given the plot of `J`, select your `k` and explain your choice in `hw2_kmeans.ipynb`. [10 pts]

(f) Generate the final compressed image with your selected `k`. [5 pts]

2 Classification [50 pts]

Follow the instructions below to complete `hw2_classification.ipynb` provided on the ETL.

We have learned various machine-learning based classifiers, such as Logistic Regression, Naive Bayes, Random Forest, and more. In this exercise, you are going to train classification models for the 2 datasets you are given on ETL.

Explore and analyze the datasets first, and then fit multiple different models. Optimize as much as you can, by trying regularization methods, model selections, cross validation and other methods that can be applied to your models. Report the best model that you think is the most appropriate for each dataset. You may use the modules from scikit-learn. Write all your answers in the `hw2_classification.ipynb` file.

(a) Given the data `diabetes.csv`, the task is to predict whether the person has diabetes or not. Train at least 2 classification models and briefly explain your choice of models. Report the performance of each model. [7 pts]
(You do not have to analyze the results for this question.)

(b) Report the best model among the models that you trained in (a). Compare the models and explain why you chose the final model. Describe how you optimized the final model to get the best performance. You will get full credit if you plot the process of model optimization. [15 pts]

(c) Given the data `credit_score.csv`, the task is to classify the credit score of a person. Train at least 2 classification models and briefly explain your choice of models. Report the performance of each model.[7 pts]
(You do not have to analyze the results for this question.)

(d) Report the best model among the models that you trained in (c). Compare the models and explain why you chose the final model. Describe how you optimized the final model to get the best performance. You will get full credit if you plot the process of model optimization. [15 pts]

(e) Discuss the differences or commonalities between the two datasets with respect to why you chose the best model for each dataset. [6 pts]