Exercise #2 – Classification and Clustering Machine Learning Models

DUE: AS INDICATED on Canvas (before Midnight)

This exercise focuses on understanding supervised classification as well as unsupervised clustering methods to build models using both Python and C.

Turn in your report that is your individual effort. If you get significant help from another classmate or make use of code your find that is open source, or use ChatGPT, please cite this in your report clearly and please review the syllabus to understand what is allowed in terms of collaboration and use of AI tools and open source.

Please thoroughly read chapters 3, 8 and 9 in the on-line text used in the course (Hands-on ML 3rd Edition) that corresponds to the Jupyter notebook examples (https://github.com/ageron/handson-ml3) that are intended to give you an interactive example of how classifying and clustering ML models work. Based on what you learn, you will then be asked to build your own stand-alone models in Python or C/C++, or you can create a new notebook if you prefer. If access to Hands-on ML 3rd Edition is a problem for you, note that Hands-on ML 2nd Edition is also available and when paired with https://github.com/ageron/handson-ml2, this is a good backup and sometimes simpler version of the same concepts and code you may also want to consult.

Note: You may want to install OpenCV on wheels for Python to do the last problem. Please see this example if you plan to use Python - https://www.ecst.csuchico.edu/~sbsiewert/csci581/code/pycv_demo/

It is preferred that you complete all work using ECC-Linux and/or "cscigpu" systems provided by CSU, however you will be able to do work on any system that has Python 3, ability to connect to Google Co-Lab, and any Linux system that can compile and run C/C++.

Note that you can install Python3 modules needed with "pip3 install", for example, "pip3 install opencv-python". You should install Python3 before you use pip3 to install modules used.

Exercise #2 Goals and Objectives:

1) [30 points] The goal for this problem is to use the machine learning model provided and to assess performance using standard ML metrics. Please read and complete the following Jupyter notebook (03_classification.ipynb) and refer to Chapter 3 reading for more background. Once you have completed the reading and notebook, provide a screen shot of the PR curve and ROC, and then please answer the following questions:
   a) What is the difference between precision "P" and recall "R" in terms of the raw performance parameters TP, FP, TN, and FN?

b) Why is PR AUC (area under the PR curve) a good indicator of how well an ML model performs?
c) Why is the ROC useful in addition to a PR curve and how can it be used with a threshold?

2) [20 points] The goal for this problem is to understand the concepts of the "curse of dimensionality" and look at methods to simplify data by creating a lower dimensional version of it. Please read and complete the following Jupyter notebook (08_dimensionality_reduction.ipynb) and refer to Chapter 8 reading for more background. Once you have completed the reading and notebook examples, please provide a screenshot of the simple 3D data projection onto a 2D plane and the more complex Swiss roll manifold dimensional reduction, and then answer the following questions:
   a) Why can't all 3D data sets be projected onto a 2D plane for dimensionality reduction? (e.g., the Swiss roll)
   b) What is the principle behind PCA and what does Figure 8-7 show?

3) [20 points] The goal for this problem is to explore the use of K-means clustering by exploring the application of the method to synthetic data. Please read and complete the following Jupyter notebook (09_unsupervised_learning.ipynb) and refer to Chapter 9 reading for more background. Once you have completed the reading and notebook examples, please provide a screenshot of the best K-means clustering of the "5 blobs" data, and then answer the following questions:
   a) What is meant by "If you plot the cluster's decision boundaries, you get a Voronoi tessellation"?
   b) Why might K-means fail and what are at least two ways to avoid failure?

   Continue using the Chapter 9 notebook and explore DBSCAN, and provide a screenshot of the decision boundary for the two example non-linear clusters and then answer the following questions:
   c) When should DBSCAN be used rather than K-means?
   d) What are at least two disadvantages of DBSCAN compared to other clustering methods?

4) [30 points] Now explore the use of K-means clustering with images in Python and C as a stand-alone module using Python or C and create your own stand-alone application. You can choose to use either this C starter code - https://www.ecst.csuchico.edu/~sbsiewert/csci581/code/kmeans_linux/ and adapt it to segment an image such as the "sun" image or "bricks" image. You may find that this simple example to read and write out a PPM file is helpful for the C code (Mat-rotate-with-read-write/). Or you can choose to install Python OpenCV for Python3 with "pip3 install opencv-python" and segment either image. The Python code found here

(https://machinelearningmastery.com/k-means-clustering-in-opencv-and-application-for-color-quantization/) provides a great example of segmentation using Python3 with OpenCV and documentation on OpenCV K-means is available (https://docs.opencv.org/4.x/d1/d5c/tutorial_py_kmeans_opencv.html). For the "sun" image try to segment water, sky, and sun. For the Legos, try to segment by the color of the bricks. Whichever method you pick, show the original image side-by-side with the segmented image with a screen shot and then answer the following questions:

a) Why did you choose the starter code you selected (C or Python)? What are two or more disadvantages of the programming language you chose?
b) What was the most difficult part of adapting the starter code into a stand-alone k-means segmentation application?

Overall, provide a well-documented professional report of your findings, output, and tests so that it is easy for a colleague (or instructor) to understand what you have done – SEPARATELY UPLODADED from code or any ZIP file for grading. Include any Python source code you write (or modify) and supply instructions for how to load data and run any code you modify. I will look at your report first, so it must be professionally written and clearly address each problem, providing clear and concise responses to receive credit.

In this class, you will be expected to consult the relevant ML Python (or C/C++) library documentation if you use functions, and you should do your best to explain how they work and not simply treat the functions as a black box.

Upload all code and your report completed using MS Word or as a PDF to Canvas and include all source code (ideally example output should be integrated into the report directly, but if not, clearly label in the report and by filename if test and example output is not pasted directly into the report). ***Please zip or tar.gz your CODE with your first and last name embedded in the directory name. Note that I may ask you to walk through and explain your code.***

Grading Rubric

[30 points] Complete Jupyter notebook (03_classification.ipynb) for Chapter 3 reading and answer all questions related to ML model development for classification with performance assessment.

Problem #1	Points Possible	Score	Comments
a) Difference between P and R in terms of raw performance data	10		
b) Use of PR AUC	10		
c) Value of ROC	10		
TOTAL			

[20 points] Complete Jupyter notebook (08_dimensionality_reduction.ipynb) for Chapter 8 reading and answer all questions related to dimensionality reduction.

Problem #2	Points Possible	Score	Comments
a) 3D data sets be projected onto a 2D plane	10		
b) What is PCA?	10		
TOTAL			

[20 points] Complete Jupyter notebook (09_unsupervised_learning.ipynb) for Chapter 9 reading and answer all questions related to K-means clustering and DBSCAN.

Problem #3	Points Possible	Score	Comments
a) Voronoi tessellation	5		
b) Why might K-means fail	5		
c) When should DBSCAN be used rather than K-means	5		
d) What are at least two disadvantages of DBSCAN	5		
TOTAL			

[30 points] Complete https://github.com/ageron/handson-ml3/blob/main/01_the_machine_learning_landscape.ipynb and answer all questions related to ML model development using regression.

Problem #4	Points Possible	Score	Comments
a) Side-by-side segmentation and original image display screenshot	10		
b) Python or C chosen and why?	10		
c) Most difficult part of creating stand-alone K-means app	10		
TOTAL			

Report file MUST be separate from the ZIP file with code and other supporting materials.

Rubric for Scoring for scale 0…10

Score	Description of reporting and code quality
0	No answer, no work done
1	Attempted with minimal work shown, incomplete, does not run with Python or compile with C/C++ compiler
2	Attempted and partial work provided, but unclear, and runs but has errors
3	Attempted with little work provided, but unclear, build warnings for C/C++, runs with no apparent error, but not correct or does not terminate
4	Attempted and more work provided, but unclear, build warnings for C/C++, runs with no apparent error, but not correct or does not terminate
5	Attempted and most work provided, but unclear, build warnings, runs with no apparent error, but not correct or does not terminate
6	Complete answer, but does not answer question well and code has warnings or errors and does not provide expected results
7	Complete, mostly correct, average answer to questions, with code that builds without warnings and runs with average code quality and overall answer clarity
8	Good, easy to understand and clear answer to questions, with easy-to-read code that builds and runs with no warnings (or errors), completes without error, and provides a credible result
9	Great, easy to understand and insightful answer to questions, with easy-to-read code that builds and runs cleanly, completes without error, and provides an excellent result
10	Most complete and correct - best answer and code given in the current class