Applications of Machine Learning in Remote Sensing
Homework 3

John Smith – johnsmith@rit.edu

https://github.com/johnsmith/repo.git

- In your submission, include **explanation**, **results**, and **the code** for the problem in the same PDF file. Also *separately*, attach solution's codes so I can replicate your results.
- Show your understanding of the problem by providing **explanation**.
- Provide sufficient commenting in your code.
- Ensure all text/images are legible and organized.
- Ensure that your code can reproduce the submitted results.
- Include an *entry point check* so I could run your code easily. This also ensures that your code is not ran accidentally if you were to call these functions from somewhere else.

```
if __name__ == "__main__":
    csv_file = "path/to/data.csv"
    function(csv_file)
```

Groundtruth classes for the Pavia University scene and their respective samples number

#	Class	Samples
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947

Figure 1: Pavia University dataset; classes and number of samples.

Create a directory in your repository and name it `ml`, if you already do not have. The workflows and the scripts created in this homework would go under `ml`.

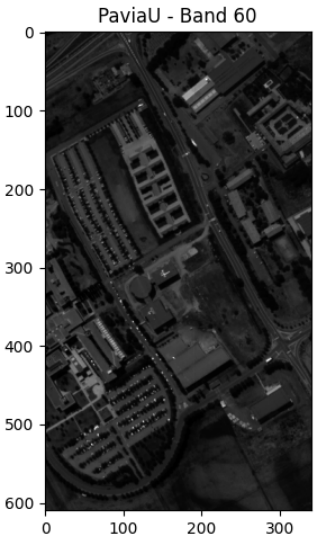
Problem 1: Classification

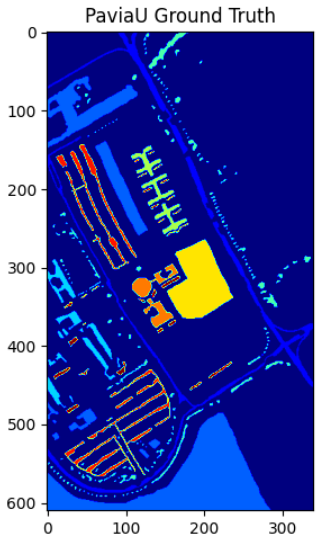
We have discussed binary and multi-class classification in class. Today, we will use the Pavia University remote sensing dataset (**PaviaU.mat and PaviaU_gt.mat)** to perform both types of classification. The Pavia University dataset is a hyperspectral image dataset which gathered by a sensor known as the reflective optics system imaging spectrometer (ROSIS-3) over the city of Pavia, Italy. The image consists of 610×340 pixels with 103 spectral bands. The number of classes and the corresponding number of samples are shown below and as you can see, the dataset is imbalanced. The ROSIS sensor collects between 430-960 nanometer in 5 nano meter spectral spacing.

(1.a)(10 points) Perform all necessary exploratory data analysis (EDA) on the given dataset. Explain what are the important aspect of this dataset that needs to be explored before delving into solving the problem.

(1.b)(15 points) We will perform binary classification using logistic regression to distinguish between Vegetation (trees and meadows) and Non-vegetation (all other classes). For this problem follow these steps:

- Generate training and testing partitions of your data. How would you make sure your original 9 classes are equally represented in the your two class problem? What is a good training/testing split ratio here?
- Separate your dataset into two classes as mentioned above.





- How does the class balance between the two classes look?
- For this problem grab a band in the blue, green, red, red-edge, and near-infrared region (you can assume linear spacing between bands).
- Perform any other preprocessing step necessary, as discussed in the class.
- What is the sample size for the mentioned two classes in training and testing set? What does that tell you about the class imbalance in this problem? Explain.
- Computer and Report mean accuracy and mean per-class accuracy, precision, recall, and F1-score, and explain what precision and recall represent; for both training and testing partition.
- Use these probabilities and plot the ROC curve.
- Compute and report the Area Under the Curve (AUC) on testing partition.
- Explain and interpret your findings.

Note: you would need to dump class 0 in your lables, your total number of samples should be 42776.

(1.c)(15 points) We explained boosting and bagging algorithms in class. One of the most popular boosting algorithms is XGBoost. In this problem, we will demonstrate the performance of a multi-class classification model using the XGBoost algorithm.

For this problem, follow these steps:

- Generate training and testing partitions of your dataset. Ensure that the original 9 classes are equally represented in both sets.
- Use the 5 bands chosen in the previous problem. You are free to also report on the performance of the model by including more features that you think could help the task at hand.
- Apply proper data preprocessing.
- Train an XGBoost model on the dataset and analyze the model's performance on the test set.
- Generate the feature importance plot and interpret the significance of the top features (you can use XGboost functionality for this).
- Show the confusion matrix. Interpret the results by identifying which classes are frequently confused. What might cause this confusion?
- Report and explain the following metrics: Mean Accuracy, Mean Per-Class Accuracy, Precision, Recall, F1-Score
- Since the dataset is imbalanced, implement a strategy to provide a balanced training and testing sets for all 9 classes. Implement the strategies and train your model.
- Compare performance with and without data balancing.
- Discuss your findings.

Note: You must remove class 0 from the labels. The total number of samples after this step should be 42,776.

Problem 2: Regression

In this problem, we will use the dataset (**landis_chlorphyl_regression.npy and landis_chlorphyl_regression_gt.npy)** generated from a combination of the PROSPECT radiative transfer model for leaf reflectance spectra at varying chlorophyll levels and MODTRAN for forward modeling the spectral data to simulate observations from the Landsat next satellite. Ten bands in the visible to near-infrared region has been selected for the purpose of this problem. The bands are as below:

The dataset includes 1000 samples with 10 spectral bands, along with the corresponding chlorophyll content obtained from the original Prospect simulation. This dataset presents a regression task where the goal is to predict chlorophyll content using the spectral band

Band Name	Center Wavelength (nm)
Blue	490
Green	560
Yellow	600
Orange	620
Red 1	650
Red 2	665
Red Edge 1	705
Red Edge 2	740
NIR_Broad	842
NIR1	865

values as features.

For this problem, follow the steps outlined below:
(2.a)(10 points) Perform all necessary exploratory data analysis (EDA) on the given dataset. Explain what are the important aspect of this dataset that needs to be explored before delving into modeling the problem. Do you observe any source of multicolinearity? How does the distribution of your target variable look in terms of balance?

(2.b)(18 points) Linear Regression: We take advantage of Linear Regression for modeling this task.

- Perform necessary preprocessing steps on the dataset.
- Split the dataset into training and testing sets. What is an appropriate ratio for the split?
- Use linear regression from the `scikit-learn` library to predict chlorophyll content. For both the training and testing partitions, report the following metrics: Mean Absolute Error (MAE), R-squared (R^2), Standard Deviation of Residuals
- Generate Regression and residual plots for both the training and testing partitions. Discuss any trends you observe. Do predictions align along the one-to-one line? What patterns do you see in the residuals? How does the spread look in the residuals?
- Report the performance metrics directly on the plots for clarity.
- Analyze and explain your findings.

(2.c)(16 points) Partial Least Squares Regression (PLSR): Next, we are using Partial Least Squares Regression (PLSR). PLSR is conceptually similar to PCA, it takes into account the target variable when performing the transformation. This makes PLSR a powerful linear approach for extract variability across independent variables with respect to target variable.

- Vary the number of components from 1 to 10.
- Report the training accuracy for each component count.
- Identify the number of components that yields the highest training accuracy.
- Using the best-performing component count, generate the same regression and residual plots for both the training and testing partitions.
- Analyze and explain your findings. How do the trends change compared to linear regression? Do you observe any notable differences?

(2.d)(16 points) Multiple Layer Perceptron (MLP): We discussed perceptron in the class as a linear classifier, multi layer perceptron is a class of feedforward neural networks consisting of an input layer, one or more hidden layers, and an output layer. Each layer contains multiple neurons, and each neuron applies a weighted sum followed by a nonlinear activation function (e.g., ReLU, sigmoid) to capture complex patterns in the data. The non-linear activation functions gives us the ability to model more complex non-linear relationships.

- Use MLP with varying number of layers (1-5) and RelU activation function. What happens with the increase in the number of layers?
- Show regression and residual plot for each model separately and compare and contrast the results you observe.
- Report evaluation metrics, similar to previous models (Linear regression and PLSR)
- Examine the trends in the residual plots for each model.
- Analyze how model complexity influences performance.
- Comparing the best performing results from MLP with LR and PLSR, how does bias and variance trade off change when using models with different complexities?
- Analyze and explain your findings.

Note: Make sure all plots are clearly labeled and include appropriate legends. Provide a concise written interpretation of each plot, focusing on any patterns, trends, and potential sources of model error.