CENG 463 MACHINE LEARNING:

LAND COVER MAPS PROJECT REPORT

TEAM: Mostapha Alhaj 18030441019

ABSTRACT:

This paper presents a method for generating land cover maps using logistic regression. The dataset used in this study includes aerial imagery from Sentinal-2 data and corresponding land cover labels. The model is trained and tested using Gibraltar as the study area and the results show that logistic regression can be used to produce land cover maps with an overall accuracy of 80% and an F1-score of 0.399. The results also demonstrate that the logistic regression model outperforms other traditional methods while maintain simplicity and can be useful in a variety of land cover mapping applications. Further research is needed to address and resolve low precision scores of a subset of the land cover classes.

GITHUB REPO:

Click here.

INTRODUCTION

Literature Review:

For a long time, the creation of land cover maps has attracted attention in the field of remote sensing. Object-based image analysis, supervised classification, and unsupervised classification are common techniques for creating land cover maps. One of the most popular methods is supervised classification, which typically makes use of training data made up of picture pixels that have been assigned to the correct land cover class. On the other hand, unsupervised classification groups image pixels according to their spectral properties without using training data. A more contemporary technique called object-based image analysis (OBIA) makes use of picture segmentation to recognize and categorize things according to their texture, size, and shape.

Recently, land cover maps have been created using machine learning (ML) techniques. These techniques have been found to be more accurate than conventional techniques and are renowned for their capacity to automatically identify patterns from vast amounts of data. Land cover mapping makes extensive use of machine learning techniques like Random Forest, Support Vector Machine (SVM), and convolutional neural networks (CNNs). These techniques have been tested on numerous varieties of satellite pictures and have shown to be successful in various land cover mapping applications.

Problem Definition:

We will use several training samples to create land cover maps for unobserved data. Gibraltar was chosen as the study location because it has a diverse range of land cover types. We will assess which classification model is best fit for our study and adjust the parameters or use some preprocessing techniques to produce better models. Our target metric is the F1-score.

We won't be considering any deep learning or neural networks approach.

DATA DESCRIPTION

Gibraltar is selected as the study area as there is a wide variety of land cover types. Based on the documentation, The discrete classification map provides 11 classes and is defined using the Land Cover Classification System (LCCS) developed by the United Nations (UN) Food and Agriculture Organization (FAO). The UN-LCCS system was designed as a hierarchical classification, which allows adjusting the thematic detail of the legend to the amount of information available.

Our training data however only has eight classes out of the total 11. The number of observations is 967005 and there are four independent features, Blue, Green, Red and NIR. Our target is the Code feature.

METHOD

First, we observe our training data. The target class has 9 unique values [0,10,20,30,40,50,60,80,90]. However, 0 is not part of that class labels in the documentation, so we remove those values.

We calculate the NDVI and NDWI using the existing features in the dataset, then we concatenate to the training data. Here we also notice that some observations have zeros in all columns, this is an issue because it will return NaN since:

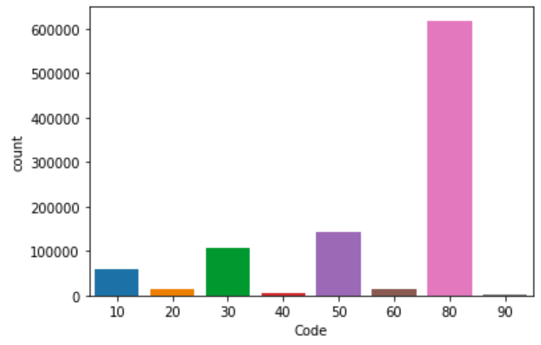
$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{red}}{\mathrm{NIR} + \mathrm{red}}$$

And

$$NDWI = \frac{(Green\ band - NIR\ band)}{(Green\ band + NIR\ band)}$$

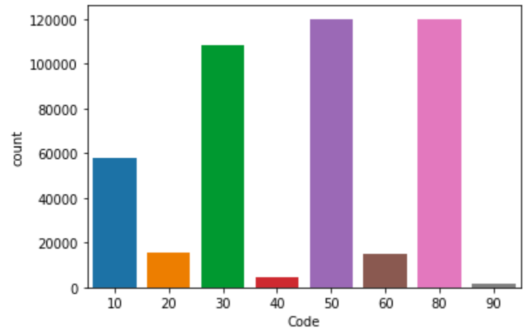
So, we end up removing all rows where all features are zero. The remaining data set has 965282 rows. Now we divide our data to X and y where we will predict our target classes.

Before we proceed, we need to study our training data a bit more. We observe the distribution of the target classes:

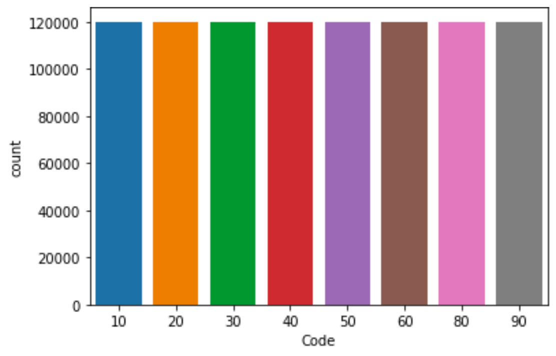


As shown, there is a very large class imbalance, this is not good for logistic regression, so we will be performing some over and under sampling.

After under sampling the majority classes our distribution looks like:



Now we perform over sampling:



Now this isn't ideal, but it will help the model perform better.

Then we split the data after over and under sampling to train and test data with test size 20%. We train our model on the train split (which has 768000 rows) but, we don't test our model on the test split, instead to assess the true performance of the model we test it on our original training data set before over and under sampling.

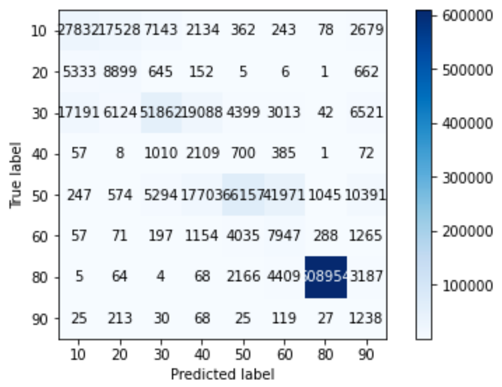
RESULT

First, we observe our model accuracy:

Accuracy of Logistic regression classifier on raining set : 0.59
Accuracy of Logistic regression classifier on test set : 0.80

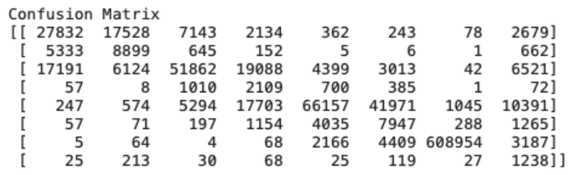
As you see, there is a poor performance whe n training but improvement when testing. Su spected underfitting but that is good given ou r case.

Confusion Matrix:



Because the number of samples for class 80 is exceptionally large, the heat gradient fails to show the model performance across the diagonal. Due to the format, some numbers are unclear so we will discuss more using Console's confusion matrix.

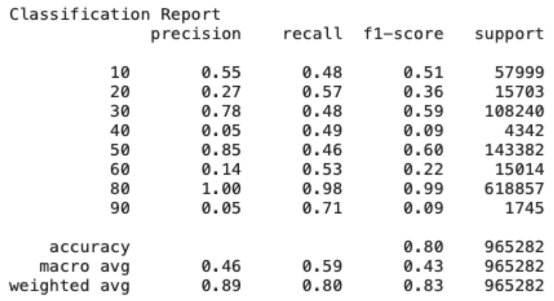
Console Confusion Matrix:



```
Confusion Matrix
[[ 27832  17528   7143   2134    362    243     78   2679]
 [  5333   8899    645    152      5      6      1    662]
 [ 17191   6124  51862  19088   4399   3013     42   6521]
 [    57      8   1010   2109    700    385      1     72]
 [   247    574   5294  17703  66157  41971   1045  10391]
 [    57     71    197   1154   4035   7947    288   1265]
 [     5     64      4     68   2166   4409 608954   3187]
 [    25    213     30     68     25    119     27   1238]]
```

The bulk of values seem to be revolving around the diagonal, indicating that misclassifications usually happen between neighboring classes.

Classification Report:



```
Classification Report
              precision    recall  f1-score   support

          10       0.55      0.48      0.51     57999
          20       0.27      0.57      0.36     15703
          30       0.78      0.48      0.59    108240
          40       0.05      0.49      0.09      4342
          50       0.85      0.46      0.60    143382
          60       0.14      0.53      0.22     15014
          80       1.00      0.98      0.99    618857
          90       0.05      0.71      0.09      1745

    accuracy                           0.80    965282
   macro avg       0.46      0.59      0.43    965282
weighted avg       0.89      0.80      0.83    965282
```

The F1-score is 0.43 which is good given how unbalanced the dataset is. Notice how our weakest values are in the precision. Especially classes 40, 60 and 90 all have precisions under 0.2, if improvements have to be made, this is where they should be.

DISCUSSION

First, the most important question is on why we chose Logistic regression. Logistic Regression is more computationally efficient than Random Forest, which is especially important when working with large datasets. With Random Forest, the bigger the dataset, the more time it will take to train and the bigger the space it will take to store the model. The case is similar with SVM where training the model would be a time-consuming task when dealing with large datasets like the ones in this report.

Since logistic regression is a linear model, this means that it is more interpretable than non-linear models like SVMs or random forests. This can be useful when trying to understand the underlying relationships between the predictors and the response variable in the data.

The current model performance is decent. However, it can be improved if some other techniques where involved, such as decision threshold adjustment and class weight management.

Overall, I can say that I did well enough but can do better if more time and effort were spent.