Project Group: Eastern Standard Time

# Meghana: Random Forest, XgBoost, and K-fold Cross Validation:

## Data Pre-Processing

Random Forest and XgBoost Classifier models were trained on Ontario and British Columbia data extracted from Sentinel-2 hub and NRCan. Carried out Data standardization on features.

Data was split 80/20 with 80% being used for training and 20% being used for testing. While performing K-fold Cross validation the training dataset was further split 75/25 with 25% being used for validation.

## Model Structure

Random forest was trained on Ontario and British Columbia data, after ensuring all possible land covers were contained in these regions. However, considering the limitations with Random Forest Classification, XGboost classification was preferred with the following advantages.

1. XGBoost always gives more importance to functional space when reducing the cost of a model while Random Forest tries to give more preferences to hyperparameters to optimize the model
2. XGboost prunes the tree with a score called “Similarity score” before entering into the actual modeling purposes, while considering gain of node. If the gain from a node is found to be minimal then it just stops constructing the tree to a greater depth which can overcome the challenge of overfitting to a great extent.
3. XGBoost is a good option for unbalanced datasets but we cannot trust random forest in these types of cases. In XGBoost, when the model fails to predict the anomaly for the first time, it gives more preferences and weightage to it in the upcoming iterations thereby increasing its ability to predict the class with low participation.
4. When the model is encountered with a categorical variable with a different number of classes then there lies a possibility that Random forest may give more preferences to the class with more participation.

**XGboost Classifier**

### Important features

Due to the unbalanced nature of the dataset, class\_weight from sklearn’s ‘utils’ library was incorporated to deal with bias and overall low accuracy due to misclassification.

### 1 Learning Rate : To combat the low validation accuracy, a low learning rate of 0.01 allowed for training and test accuracy to increase at a similar rate, and less variance in validation and test accuracies.

### 2. Gamma: Set to value 1 Minimum loss reduction required to make a further partition on a leaf node of the tree.

3. Max\_depth: Between 6 and 10 to allow the model to train a deep tree, preventing overfitting.

4. Subsample: set between 0.5 to 1.0: XGBoost would randomly sample half of the training data prior to growing trees.

5.ColSample\_bytree: Set to 0.8 , i.e the subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed.

6. Reg\_lambda: Set to 1: L2 regularization term on weights.

7. Tree\_method: The tree construction algorithm used in XGBoost. gpu\_hist was used for GPU implementation of hist algorithm.

8. Eval\_metric: default metric assigned to the validation data, according to the learning objective. ‘Merror’ used for Multiclass classification error rate ‘Auc’ operating Characteristic Area Under Curve.

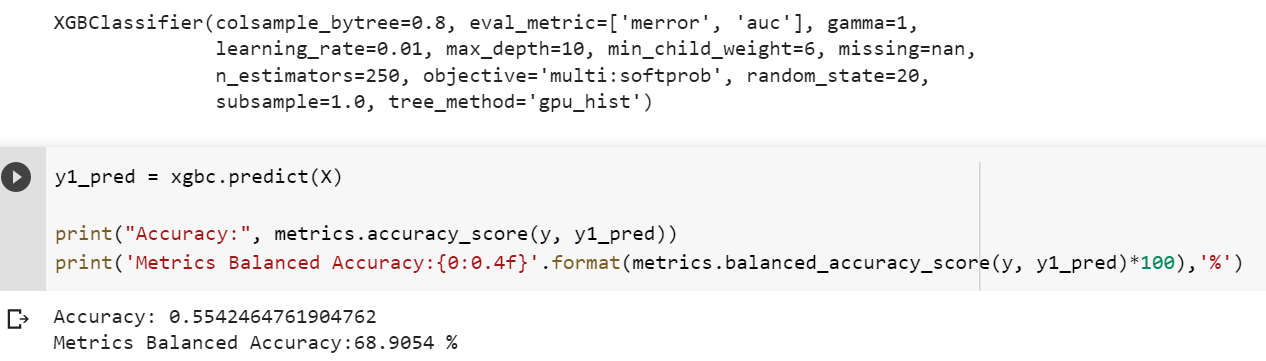
**K-fold Cross validation**

K-fold Cross validation was performed over XGboost to build a more robust model with XGboost. The following CV parameters were added. XGBoost supports k-fold cross validation using the cv() method.

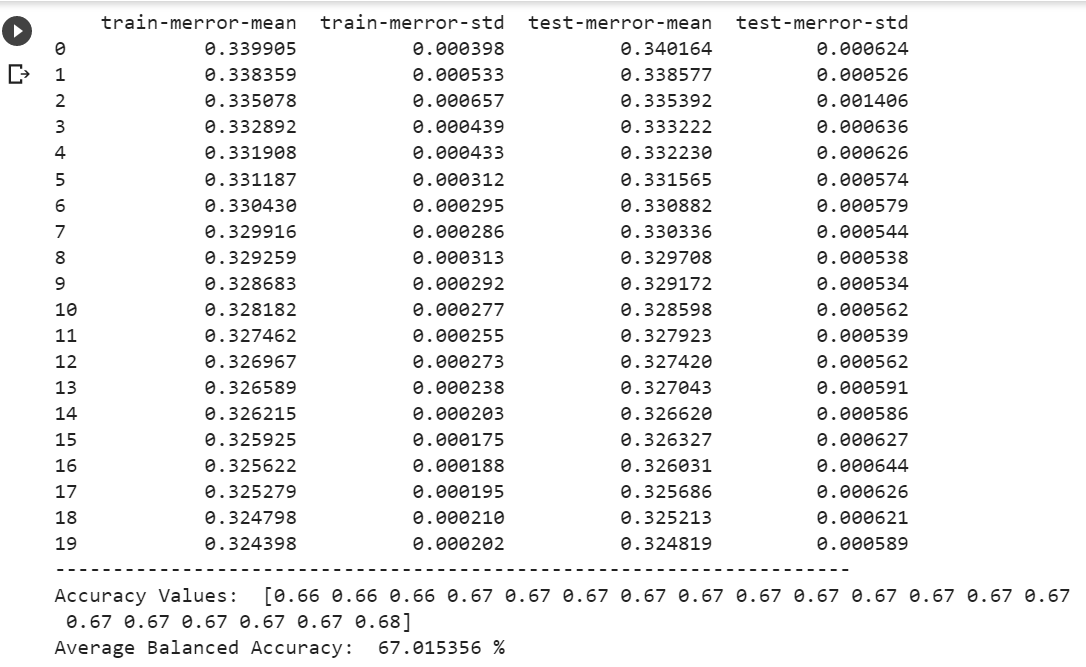
1. **Objective**: To specify the learning task and the corresponding learning objective ‘Multi:softmax’- XGBoost does multiclass classification using the softmax objective, you also need to set num\_class(number of classes). Num\_class was set to 20
2. **nfolds** - Set to 5. This parameter specifies the number of cross-validation sets we want to build.
3. **num\_boost\_round** - Set to 20. It denotes the number of trees we build.
4. **metrics** - Set to ‘Auc’ It is the performance evaluation metrics to be considered during CV.
5. **as\_pandas** - It is used to return the results in a pandas DataFrame.
6. **early\_stopping\_rounds** - Set to 5. This parameter stops training of the model early if the hold-out metric does not improve for a given number of rounds.
7. **seed** - Set to 42. This parameter is used for reproducibility of results.

Prediction Accuracies:

Xgboost model predicted the test and validation data with a balanced accuracy score of 68.9%



Accuracies with K-fold Cross validation are as below.



# Hafsa: Artificial Neural Network

## Data Pre-Processing

The Artificial Neural Network was trained on Ontario and British Columbia data, after ensuring that all possible land covers were contained in these regions.

As the data was loaded, the band data was cast to float64, and the land cover data was cast as int64, and then the data was standardized.

The data was shuffled to ensure that each land cover had an equal chance of being present in each of these sub-datasets. The data was split 80/20, with 20% being used for testing, and 80% being used for training. The training dataset was then further split 75/25 with 25% to be used for validation purposes, with the remaining 75% to be used for training.

## Model Structure

The ANN went through rigorous experimentation with different hyperparameters. The overall structure of the model contained an input layer, three hidden layers, and output layer.

### Activation Functions

ReLu was the initial choice for the input layer activation function, however, after observing a large variance in accuracy, there was a suspicion that all ReLu neurons were not being activated. ReLu was then switched out for “LeakyReLu” to ensure the activation of all neurons (leading to better model training), as well as faster training.

After experimenting with activation functions like Sigmoid, and Tanh, ReLu seemed to produce the best results in the three hidden layers.

Since this is a multiclass classification problem, Softmax was used in the output layer.

Using 100 as the hidden layer size was a happy medium between training speed and accuracy.

### Epochs

Training on 300 epochs was found to be the most optimal to give the model plenty of opportunities to update the weights.

### Learning Rate

To combat large loss metric and low validation accuracy, a low learning rate of 0.000025 allowed for training and validation accuracy to increase at a similar rate, and less variance in validation accuracy.

### Important features

Due to the unbalanced nature of the dataset, class\_weight from sklearn was incorporated to deal with bias and overall low accuracy due to misclassification.

## Running the model

For pre-processing of the data and post-processing of results, please see the included python file, named **“pre\_and\_post\_process\_for\_picked\_model.py”**. It is intended to be used with the included **“Eastern\_Standard\_Team\_Final\_Classifier.sav”** file. For more details, and explanation, check out my GitHub repository at:

https://github.com/hafsa-masood/EDS-Neural-Network

—

# 

# Tryambak: Convolution Neural Network

Data Pre-Processing

The objective of the project was to develop correlation between satellite images with the land-cover type. Satellite images of Sentinel2 dataset consisted of 12 bandwidths which served as features for the ML model. The target values for the ML model were land\_cover type as published by NrCAN. 2 different sites, each for test and validation of ML model, from Ontario and British Columbia were used in the current analysis.

Analysis

Two different deep-learning models CNN and stacked CNN with LSTM were evaluated for accuracy and balanced accuracy. Categorical\_crossentropy and Adam optimizer were used to evaluate and converge the losses, respectively.

Result

The highest balanced accuracy of 60% was achieved with CNN model without weights. The CNN-LSTM stacked model achieved an accuracy score of 45%.

The train and test dataset had imbalanced data and thus the model accuracy can be significantly lower in actual use-cases.

Conclusion

ANN and XGBoost performed better than CNN possibly due to lower dimension of dataset. A third dimension possibly in time could have resulted in CNN performing better than ANN.