**Group Project Report Phase - IV**

**Team Project**

Otto Group Product Classification

**Approaches Explored**

The project is about classifying the product into 9 categories based on the 93 numerical features provided about the products. We have explored the training sets with basic multiclass algorithms – Multiclass Decision Forest, Multiclass Decision Jungle, Multiclass Logistic Regression and Multiclass Neural Networks. We have worked on doing data preprocessing, identifying any parameters that could be varied and also try with feature selection process to improve on the data fed in to the training models. We have tried multiple ways using the above algorithms in Azure ML to vary a lot of parameters and feature selection process to get a rank of around 2262. After having tried out multiple ways in AZURE ML we explored the XGBOOST Algorithm in R which improved our rank to a great extent to 1050 which was our best result for the competition. The details will be further discussed below with our final approach in AZURE ML and

**Data Preprocessing:**

When training the models we identified that id and target columns are categorical in nature. We used the metadata editor to change them into categorical variables. The id should not be considered while classifying the data as this unique value which can misclassify the categories. The variables are made categorical to take care of the same. Also, the target variable needs to be labeled before we start the next steps in processing. Also we used Descriptive Statistics to see if there was any missing data or extreme outliers but at this point of time we did not find anything to fill up while pre-processing the data. All the variables were also changed to categorical and we chose the top features using the Permutation Feature Importance feature to get the best results.

**Results**:

We trained the data using the below models with default parameters and obtained the below results in tabular format in Azure ML for training and Kaggle for the test data. After varying multiple parameters for the models using multiple models and feature selection process we figured out that the Multiple Logistic Regression Model with selected data pre-processing steps above and the important features using PFI above produced the best results for us in Kaggle using Azure ML. The results using Azure ML and the final approach using R also discussed below.

**Training**:

**Microsoft Azure ML Account**: [skrudrar@asu.edu](mailto:skrudrar@asu.edu)

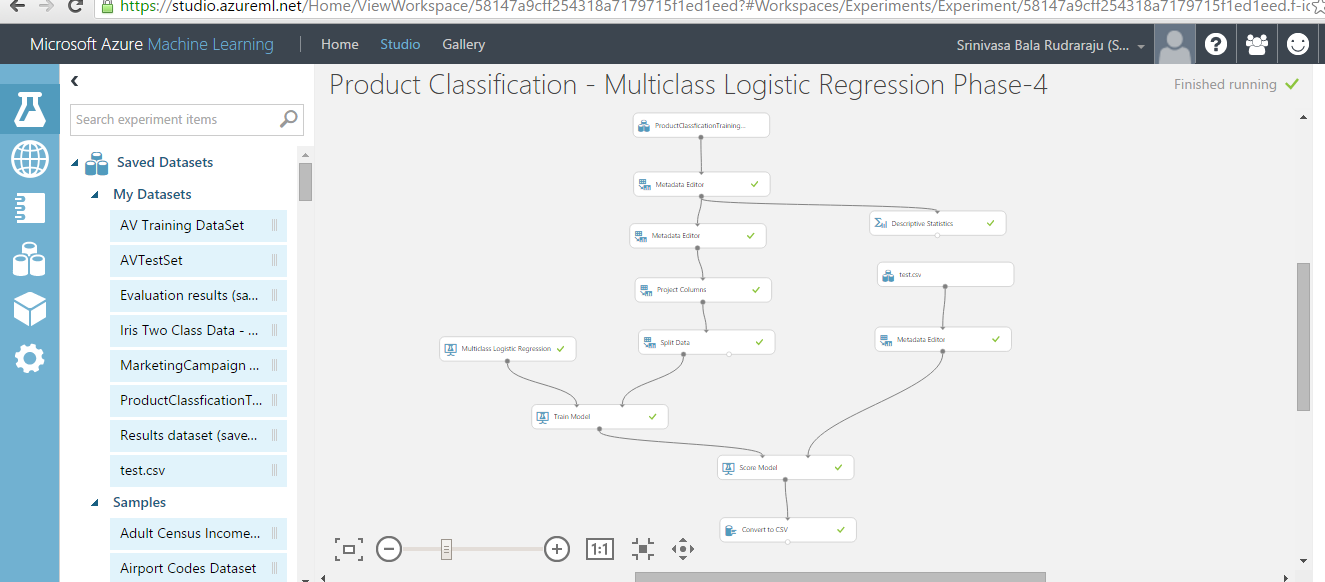
**Test**:

**Kaggle ID**: [skrudrar@asu.edu](mailto:skrudrar@asu.edu)

**Kaggle Team**: CIS508Cohort-ATeam-2

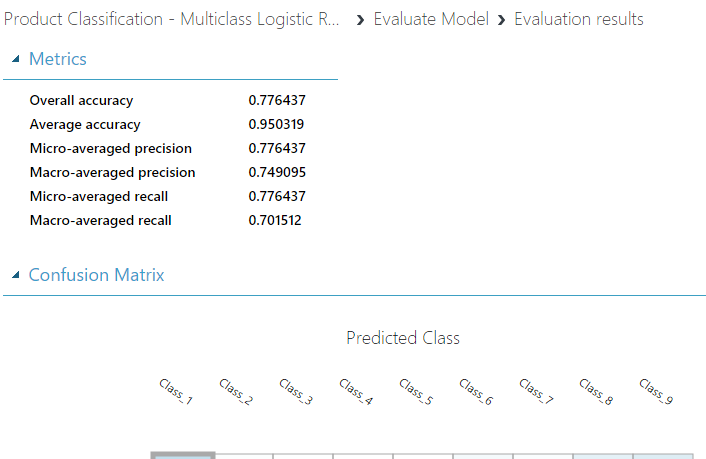
**Multiclass Logistic Regression Model using PFI**

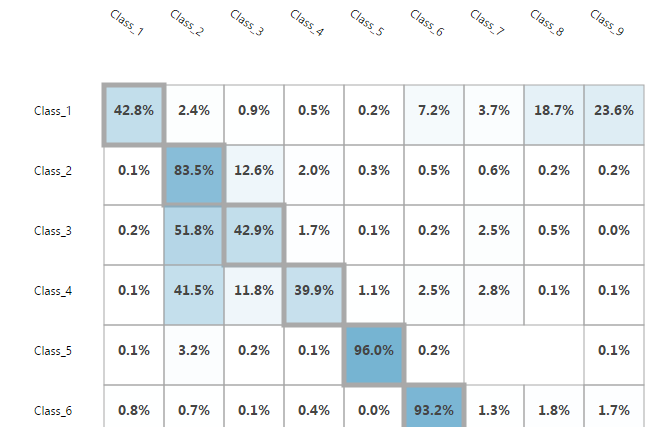
Please see the screenshot for the azure ML model using Multiclass Logistic Regression Model. We used the metadata editor feature to select the categorical variables and label the data in the model. The Project Columns was used to select the important features in the model using PFI which was discussed in detail in Phase-3 of our report. We also tried using Fisher Linear Discriminant Analysis which did not help us with any improvement as far as the log loss was concerned for the problem. The screenshot for the final Azure ML model is shown below.



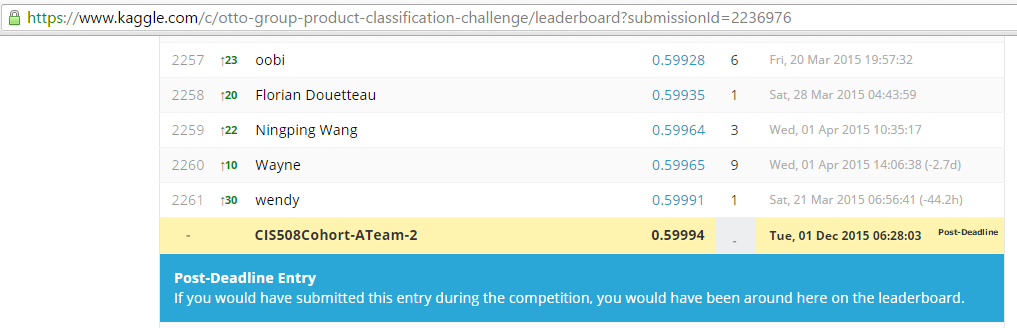
The Kaggle accuracy on the test set is also shown below.

Training:





Test:

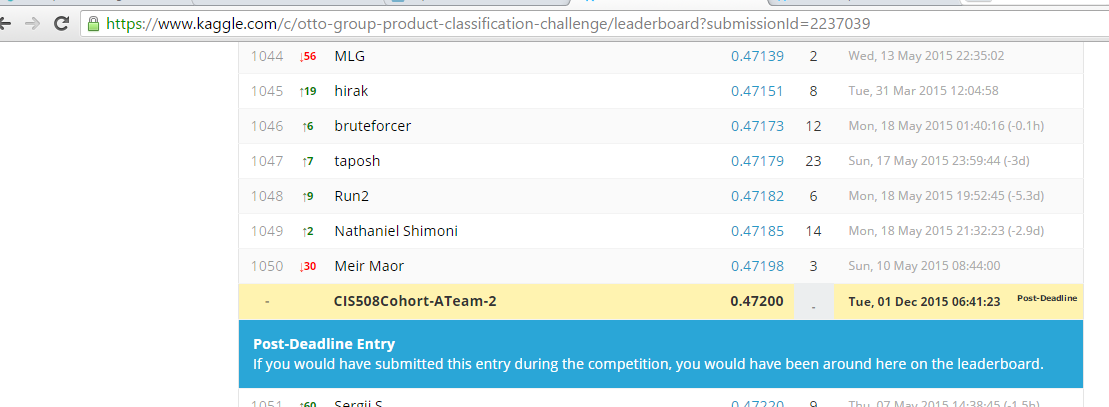


**XGBoost Algorithm Using R:**

After having exhausted with using various options in Azure ML namely all the multiclass classification algorithms with various parameters, feature selection process and ML features like Sweep Parameters we have decided to use XGBoost Algorithm which is a very popular algorithm for multiclass classification problems. It is the implementation of the famous gradient boosting algorithm.

We have written R script which uses the XGBOOST package in R to implement the algorithm. As part of the code we remove the id and target variables in the data. We also rename the Class\_\* variables to numerical from 0 to 8 as the boosting algorithm needs to have the target variables in the digital format. After having done the necessary calculations on the rows and eliminating the id and target columns accordingly we train the data and use the trained model to make the necessary predictions on the target set. We used all the 93 features given in the data set to train the data and varied a variety of hyper parameters as shown below to get the below set of results. We got the best rank using the hyper parameters mentioned in the screenshot below. The Kaggle screenshot is also incorporated for the same.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S No.** | **nfold** | **nround** | **nthread** | **max.depth** | **Public Score** | **Private Score** | **Rank** |
| 1 | 3 | 50 | 8 | default | 0.50673 | 0.50995 | 1491 |
| 2 | 3 | 100 | 8 | default | 0.48054 | 0.48246 | 1185 |
| 3 | 3 | 100 | 8 | 9 | 0.47353 | 0.472 | 1050 |
| 4 | 9 | 200 | 32 | 9 | 0.49942 | 0.50009 |  |
| 5 | 18 | 100 | 32 | 18 | 0.52148 | 0.5231 |  |
| 6 | 3 | 100 | 8 | 3 | 0.54831 | 0.54837 | 1788 |
| 7 | 6 | 100 | 8 | 12 | 0.49073 | 0.49058 | 1265 |
| 8 | 6 | 50 | 8 | 12 | 0.4779 | 0.47818 | 1129 |
| 9 | 9 | 50 | 8 | 12 | 0.4779 | 0.47818 | 1129 |
| 10 | 3 | 50/100 | 8 | 6 | 0.48054 | 0.48246 | 1185 |
| 11 | 6 | 25/50 | 8 | 6 | 0.50673 | 0.50995 | 1492 |
| 12 | 6 | 100 | 8 | 9 | 0.48054 | 0.48246 | 1185 |
| 13 | 3 | 100 | 8 | 12 | 0.49073 | 0.49058 | 1265 |
| 14 | 6 | 100 | 8 | 3 | 0.54831 | 0.54837 | 1788 |



**Learning from the Project:**

We have tried out variety of multiclass classification algorithms and various feature in Azure ML which helped us understand how the algorithms perform on a complex multiclass classification problem like the Otto Competition we took up for this project. We tried our best in Azure ML and also explored XGBOOST algorithm in R which is a very popular technique on multiclass classification algorithms. We hope to enrich our experience and improve our performances with the learning we had in this project to solve even tougher competitions going forward.