DATA 602: PCS1
December 19, 2021
Jenny Fish, Isha Angadi, Adam Claudy, Samiha Khan, Sandeep Pvn, Mehar Chaturvedi

Project Teal

Ovarian Cancer impacts over 20,000 people nationwide each year. This cancer is known to have a low late stage survival rate at 26%. Additionally, most diagnoses happen during later stages due to an absence of symptoms at the onset of the disease. For this reason it is imperative to find ways to detect early signs. Beginning in 2011, a group of researchers from Soochow University conducted a study that collected blood samples from 349 Chinese patients with Ovarian Cancer tumors over the course of 7 years. The data collected was then used to create a machine learning model designed to predict whether an Ovarian Cancer tumor is benign or malignant based on various biomarkers and demographic features. For our project, we chose to implement this model in python and vary parameters having to do with preprocessing, feature selection, and decision tree criteria. Our team comes from various backgrounds yet we found a common interest in this topic because of the promise in improving cancer detection using machine learning methods we've learned. This write-up describes how Soochow University implemented the model, the parameters our team varied, the performance of each experimental model, and things we would do differently in the future.

The data collected consists of 349 instances with a total of 49 features. 235 instances were reserved for training and 114 for testing. These features constitute various biomarkers with the exception of two non-biomarkers. The first step in processing the data as described in the paper is getting rid of columns with over 50% missing data. There was one such column - biomarker CA72-4. The model goes on to impute missing values with the mean for each feature. This is followed by a feature selection process that uses an algorithm called Minimum Redundancy Maximum Relevance (MRMR). The feature selection results in 8 significant features - Menopause, Age, AFP, CEA, HE4, CA19-9, LYM%, and CO2CP. Lastly, two prediction algorithms were built using logistic regression and decision tree methodology respectively. The decision tree has a depth of two and utilizes gini as a selection criterion.

Our primary goal in this project was to determine the impact of preprocessing steps, feature selection, and decision tree criteria on model accuracy. A secondary goal was to improve the performance of the model by tuning the mentioned parameters. Different python models were constructed to understand the impact of each hyperparameter. In one version, using median and mode for imputation were tested against using mean. In a second version, the

impact of feature selection was tested by running the algorithm with and without it. In a third version, the goal was to see how the decision tree performed upon increasing depth and using entropy instead of gini as the impurity criterion. After obtaining a good understanding of how each hyperparameter impacted the prediction algorithm individually, a loop was constructed that generates 48 different experimental algorithms. Each experiment utilized different combinations of the mentioned model parameters. In this way we could see how the hyperparameters interacted with each other in addition to seeing how they acted independently. A supplementary step incorporated to some experiments was stratified k-cross validation with the goal of seeing whether it increased accuracy. These experimental algorithms were then run using the data reserved for testing. The following metrics were used to measure the performance of each experiment: confusion matrix, sensitivity, specificity, PPV, NPV, F score, overall accuracy, and mean stratified cross-validation accuracy.

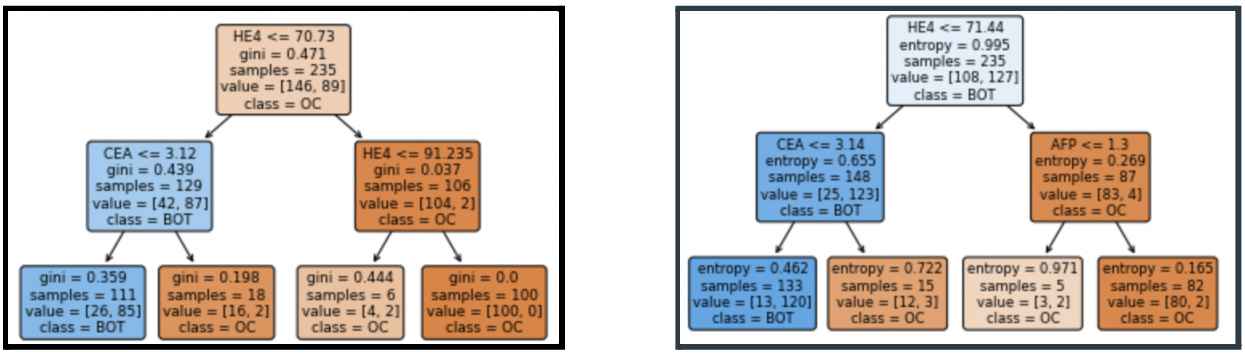

Figure 1: Base Model Decision Tree vs. Team's Decision Tree

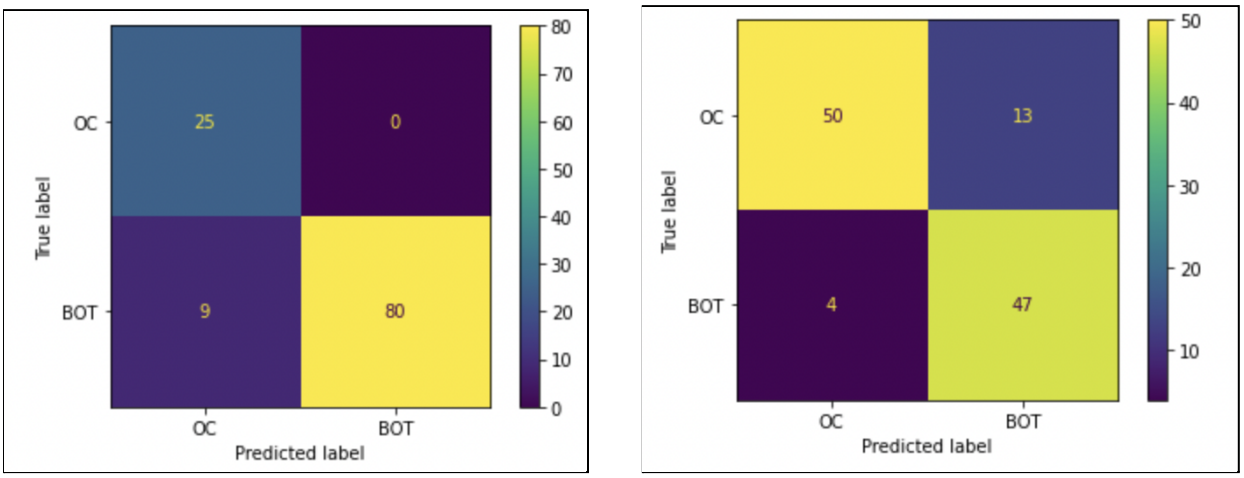

Figure 2: Base Model Confusion Matrix vs. Team's Confusion Matrix

From the results in Version 1, 2, and 3, we learn that the best accuracy when missing data is imputed using the mean. The order of best accuracy is Mean>Mode>Median. Our speculation

is that when imputing with the median or mode, we are not accounting for outliers that may have been very rare but significant for our data. We further speculate that with more time and effective hyperparameter tuning, our model can be more accurate using the median. When mode is employed to impute the missing data, we observe that with a decision tree of only two levels, we face the issue of underfitting with a high bias. To combat this issue, we chose to let the decision tree flow beyond 2 levels up to 20. At this point, we noticed that as we increase the number of levels in our decision tree, the test accuracy decreases because of overfitting. With this experiment, we confirmed that feature selection is required to assess the accuracy of our model.

When comparing gini vs. entropy, we confirmed that the number of features is the deciding factor. When using less features, the results returned when using gini or entropy on test data returns the same results. On increasing the number of features, there is a significant difference in the results obtained. Our results are divided into two categories from the experiments in this criteria:

- Entropy performs better than Gini: In the case of mode and median
- Gini performs better than Entropy: In the case of mean

Upon utilizing stratified K-cross validation with shuffling — our team achieved 82.5% testing accuracy (with the criterias MRMR, 8 features, and entropy). When compared with the stratified K-cross validation without shuffling, our team's test accuracy with the same criteria increases by 16.7% (from 65.8% to 82.5%). On inspection, we concluded that this improvement is caused by a class imbalance found within the train and test datasets.

To further enhance our assessment, we introduced a new metric called the "Teal Score" which was derived by combining specificity and precision using a geometric sum. The formula for this is shown below.

$$Teal\ Score = \frac{1}{1 + \frac{FP}{2}\left[\frac{1}{TN} + \frac{1}{TP}\right]}$$

Figure 3:Teal Score Formula

Specificity and precision were chosen as metrics of interest because they provide a measure of false positives and true negatives. We determined that reducing false positives and true negatives should be a priority. This is because we do not want to report a patient who truly has ovarian cancer as benign. The best accuracies and teal scores of the 48 experiments are

highlighted below. The combination of hyperparameters that provide the best accuracy is that used in the paper. However, our team was able to obtain a close accuracy upon using feature selection, entropy for decision tree criterion, mean for imputation, and using stratified k-cross validation. In conducting these experiments, we identified potential avenues for improving the prediction model in the future.

Experiment	TP	FP	FN	TN	Sensitivity (Recall)	Specificity	Positive Predictive Value	Negative Predictive Value	F1 score	Accuracy	Teal score	Tree depth
Teal_MRMR_features__gini__mean_	**80**	**0**	**9**	**25**	**0.899**	**1.000**	**1.000**	**0.735**	**0.947**	**0.921**	**1.000**	**2**
Teal_MRMR_features__gini__mean__stratified_k_cross	80	0	9	25	0.899	1.000	1.000	0.735	0.947	0.921	1.000	2
Teal_all_features__gini__mean_	76	0	13	25	0.854	1.000	1.000	0.658	0.921	0.886	1.000	4
Teal_all_features__gini__mean__stratified_k_cross	76	0	13	25	0.854	1.000	1.000	0.658	0.921	0.886	1.000	4
Teal_all_features__entropy__median_	70	0	19	25	0.787	1.000	1.000	0.568	0.881	0.833	1.000	3
Teal_all_features__entropy__median__stratified_k_cross	70	0	19	25	0.787	1.000	1.000	0.568	0.881	0.833	1.000	3
Teal_MRMR_features__entropy__median_	45	0	44	25	0.506	1.000	1.000	0.362	0.672	0.614	1.000	6
Teal_MRMR_features__entropy__median__stratified_k_cross	45	0	44	25	0.506	1.000	1.000	0.362	0.672	0.614	1.000	6
Teal_all_features__gini__mode_	28	1	61	24	0.315	0.960	0.966	0.282	0.475	0.456	0.963	6
Teal_all_features__gini__mode__stratified_k_cross	28	1	61	24	0.315	0.960	0.966	0.282	0.475	0.456	0.963	6
Teal_MRMR_features__entropy__mean_	81	2	8	23	0.910	0.920	0.976	0.742	0.942	0.912	0.947	2
Teal_MRMR_features__entropy__mean__stratified_k_cross	**81**	**2**	**8**	**23**	**0.910**	**0.920**	**0.976**	**0.742**	**0.942**	**0.912**	**0.947**	**2**
Teal_all_features__entropy__mean_	77	2	12	23	0.865	0.920	0.975	0.657	0.917	0.877	0.947	4
Teal_all_features__entropy__mean__stratified_k_cross	77	2	12	23	0.865	0.920	0.975	0.657	0.917	0.877	0.947	4
Teal_MRMR_features__gini__median_	66	2	23	23	0.742	0.920	0.971	0.500	0.841	0.781	0.945	1
Teal_all_features__gini__median_	66	2	23	23	0.742	0.920	0.971	0.500	0.841	0.781	0.945	1
Teal_MRMR_features__gini__median__stratified_k_cross	66	2	23	23	0.742	0.920	0.971	0.500	0.841	0.781	0.945	1
Teal_all_features__gini__median__stratified_k_cross	66	2	23	23	0.742	0.920	0.971	0.500	0.841	0.781	0.945	1
Teal_MRMR_features__gini__mode_	57	3	32	22	0.640	0.880	0.950	0.407	0.765	0.693	0.914	7
Teal_MRMR_features__gini__mode__stratified_k_cross	57	3	32	22	0.640	0.880	0.950	0.407	0.765	0.693	0.914	7
Teal_MRMR_features__entropy__mode_	53	3	36	22	0.596	0.880	0.946	0.379	0.731	0.658	0.912	3
Teal_all_features__entropy__mode_	53	3	36	22	0.596	0.880	0.946	0.379	0.731	0.658	0.912	3
Teal_MRMR_features__entropy__mode__stratified_k_cross	53	3	36	22	0.596	0.880	0.946	0.379	0.731	0.658	0.912	3
Teal_all_features__entropy__mode__stratified_k_cross	53	3	36	22	0.596	0.880	0.946	0.379	0.731	0.658	0.912	3
Teal_MRMR_features__entropy__mean__shuffle	47	13	4	50	0.922	0.794	0.783	0.926	0.847	0.851	0.788	2

Figure 4: 48 experiments run on base model and team's model

One of the biggest challenges we faced during this project was the lack of data. Our goal was to test the model by running additional datasets through it. Due to the limited amount of data, we were unable to thoroughly test our machine learning model. When replicating the paper, the data set was divided into 67% train and 33% test datasets — this further led to limited training data to create the model. To mitigate this challenge, we recommend gathering more data.

There were a number of options we explored to develop this project further. The first thing we considered was obtaining more data to better train the machine learning model. This may help mitigate the class imbalance we observed in the data we used for this project. It may also shed some light on the outliers we observed as well. Another thing we considered was customizing imputation such that each column uses the imputing technique (mean, median, ot mode) that best increases overall accuracy. On top of this, one thing that may help in increasing model efficiency is grid search. Grid search can parallelize the code and compute optimum values of hyperparameters. Other things we would look into with respect to future work are utilizing a neural network, regression testing, and bootstrapping.

Sources

- Bast, Robert C., et al. “Biomarkers and Strategies for Early Detection of Ovarian Cancer.” *Cancer Epidemiology, Biomarkers & Prevention*, American Association for Cancer Research, 1 Dec. 2020, https://cebp.aacrjournals.org/content/29/12/2504.
- Lu, Mingyang, et al. “Using Machine Learning to Predict Ovarian Cancer.” *International Journal of Medical Informatics*, Elsevier, 23 May 2020, https://www.sciencedirect.com/science/article/abs/pii/S1386505620302781.
- Torre, Lindsey A., et al. “Ovarian Cancer Statistics, 2018.” *American Cancer Society Journals*, American Cancer Society, 29 May 2018, https://acsjournals.onlinelibrary.wiley.com/doi/full/10.3322/caac.21456.