In this exercise, the default parameters for Random Forest Classifier initially performed better than running Auto Sklearn Classifier. After further exploring the dataset, some modifications were made to improve Auto Sklearn performance

Before performance was fixed, the Auto Sklearn function was explored. Here, Auto Sklearn uses the sci-kit library to automatically search for the correct learning algorithm and optimized parameters. Within this process, Auto Sklearn uses meta-learning to pick out similar datasets and then learns from the past (previous training information). Meta-learning can be described as Auto Sklearn learning to learn. Knowing this, in order for Auto Sklearn to perform better, its information it is using to learn must be looked at and refined.

To look at the dataset, the dataset was explored using the .info() and .describe() function. Then, the random state value was changed to observe the affect of the performance. After changing the random state number a couple of times, it was observed that the performance outcome changed notably, so the value counts of the y_train and y_test values were evaluated. It was seen here

that the data was indeed imbalanced (Table 1).

Because the dataset is imbalanced, Auto Sklearn is using its meta-learning to tune parameters based off of imbalanced training data. Additionally, classification algorithms predict based off of the training set they are given. When that training set includes all data, the classifier prediction places learned/incoming data to the highest prevalent class in the dataset. With an imbalanced dataset, this means the minor classes of information are getting left out and or underrepresented. Thus, to fix this, Synthetic Minority Oversampling Technique (SMOTE) and train_test_split stratification were used before Auto Sklearn was performed.

.

Before performing SMOTE, reproducible results across multiple executions was accomplished by setting the random state to 42 within the dataset. This was accomplished using np.random.RandomState. Next, SMOTE was used to improve representation of data from the minor class. Here, SMOTE works by randomly choosing a point within the minor class and computing a k-nearest neighbor. The computed point is then added as a synthetic data point, and the

number of data points in the underrepresented data increases. In the execution here, the SMOTE function was used, then the model with k-nearest neighbor points was fit. Next, within the train_test_split, stratification was turned on. Stratification in the train_test_split allows for an even distribution of y values in each split that is made. This helps balance the imbalanced dataset. With these three things in place, the Random Forest Classification and Auto Sklearn Classifier were ran again. Before Auto Sklearn, the RF accuracy (subset accuracy showing the fraction of correctly classified samples) was 0.88 (Table 2). After Auto Sklearn, the RF accuracy was 0.96 (Table 2). In conclusion, improving how the data is trained and split is key to improving Auto Sklearn performance.

Table 1

y_train	y_test
class 5 0.431193 6 0.391159 7 0.125938 4 0.033361 8 0.010842 3 0.007506 Name: proportion, dtype: float64	class 6 0.4225 5 0.4100 7 0.1200 4 0.0325 8 0.0125 3 0.0025 Name: proportion, dtype: float64

Table 2	
Before Auto Sklearn Classifier	0.88
After Auto Sklearn Classifier	0.96

References:

1 . "10. Common Pitfalls and Recommended Practices." Scikit, scikit-learn.org/stable/common_pitfalls.html. Accessed 12 Apr. 2024.

2. "Sklearn.Metrics._score." Scikit, scikit-learn.org/stable/modules/generated /sklearn.metrics.accuracy_score.html. Accessed 12 Apr. 2024.

3. Analyticsvidhya.Com, www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/. Accessed 12 Apr. 2024.

4. Sadaiyandi, Jeyabharathy, et al. "Stratified Sampling-Based Deep Learning Approach to Increase Prediction Accuracy of Unbalanced Dataset." MDPI, Multidisciplinary Digital Publishing Institute, 27 Oct. 2023, www.mdpi.com/2079-9292/12/21/4423.

5. "Auto-Sklearn." AutoML, www.automl.org/automl-for-x/tabular-data/auto-

sklearn/

: :text=

Built%20around%20the%20scikit%2Dlearn,focus%20on%20the%20real%20problem. Accessed 12 Apr. 2024.

6. Analyticsvidhya.Com, www.analyticsvidhya.com/blog/2022/09/meta-learning-structure-advantages-examples/: :text=Meta%2Dlearning%2C%20described%20as%20%E2%80%9C,the%20results%20of%20the%20experiment. Accessed 12 Apr. 2024.

7. GfG. "Python: Pandas Series.Value_counts()." GeeksforGeeks, GeeksforGeeks, 29 Jan. 2019, www.geeksforgeeks.org/python-pandas-series-value_counts/.

8. Ogunbiyi, Ibrahim Abayomi. "What Is Stratified Random Sampling? Definition and Python Example." freeCodeCamp.Org, freeCodeCamp.org, 15 Nov. 2022, www.freecodecamp.org/news/what-is-stratified-random-sampling-definition-and-python-example/.

9. Brownlee, Jason. "Train-Test Split for Evaluating Machine Learning Algorithms."

MachineLearningMastery.com, 26 Aug. 2020, machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/: :text=Stratified%20Train%2DTest%20Splits,-One%20final%20considerationtext=As%20such%2C%20it%20is%20desirable,a%20stratified0train%2Dtest%20split.