Enrollment Predictions using AutoGluon

Shani Melbourne and Denise Anciola • Professor Michael Soltys • Capstone 499

Introduction

Machine learning is a subset of artificial intelligence in which an algorithm is trained to detect patterns within a dataset, making predictions based on what it has learned and gradually improving its accuracy. It allows a computer to think and learn similar to a human, with multiple applications over numerous fields. A recent project focused on machine learning model training, utilizing the Amazon Web Service's (AWS) machine learning service Sagemaker and the XGBoost framework to predict the likelihood of an admitted students enrollment at California State University, Channel Islands. XGBoost, short for eXtreme Gradient Boosting, is a gradient boosted trees algorithm framework that generates its predictions through combining the estimates produced from creating a set of simpler but weaker models.

The benefit of determining these predictions is that it allows for for the enrollment office to direct more resources towards those more likely to enroll and better prepare for the upcoming academic year. Our project is a continuation of that project that instead uses the machine learning framework AutoGluon, an open source machine learning library that can train models and evaluate data in as little as two lines of code. AutoGluon differs from many other machine learning frameworks by generating different kinds of models and combining it with the data to produce highly accurate models without much user input needed. The goal of our project is to compare the results and functionality of the two frameworks in order to evaluate benefits of each framework.

Methodology

AutoGluon is a powerful yet simple to utilize machine learning framework that differs from other algorithms by training multiple models, weighing and combining their results to produce high quality models to make predictions on.

- Before the model was constructed, labels were added to the training set due to Autogluon being a supervised machine learning framework, which means it needed a target label to know which column to predict
- Building and training the predictive model is accomplished with one line of code:

```
predictor = TabularPredictor(label=label, path=save_path, eval_metric='roc_auc').fit(train_data=train_data,presets='best_quality')
```

- Important features:
  - **TabularPredictor() -** predicts values in a column of a tabular dataset
  - **fit() -** fit models to predict a column of a data table based on the other columns
  - **eval_metric -** the metric used for prediction evaluation. roc_auc is short for Area Under the Receiver Operating Characteristic Curve
  - **presets -** set to best quality, which means it aims for the best predictive accuracy without considering time of disk usage
- After the model is trained, it can be ran on a test dataset with

  `predictor.predict_proba(test_data)`

  to produce the probabilities of each class, which is then used for data analysis

AutoGluon Model Training

AutoGluon handles everything from data preprocessing to model fitting without any need to set up parameters beyond the target column and training data to use. In fact, not setting hyperparameters for the predictor maximizes the predictive accuracy of the created model. We set the evaluation metric to roc_auc in order to make it easier to compare to the model training done with xgboost, which used auc for it's evaluation metric. The preset *best_quality* is the only other parameter set in order to produce the best possible model accuracy. Once training is initiated, AutoGluon automatically determines the data type for each column before it fits the data onto 13 different models. During this process, autogluon takes note of the evaluation scores for each model it generates in order to present the best model for the predictor.

Results

Using the parameter *best_quality* meant that AutoGluon does not worry about time spent or disk usage, and it takes a few minutes for model training to complete. While this comparable to the amount it took to train the model on XGBoost using Sagemaker, AutoGluon was able to train more models (one of the models being XGboost itself) with less setup required.

The predictions generated by the AutoGluon model were fairly accurate and in fact were more accurate that then predictions made by the XGBoost model, as it can be seen below in the confusion matrices comparing the predicted enrollment status to the actual status:

XGBoost

	Not Enroll	Enroll
Not Enroll	1992	596
Enroll	131	167

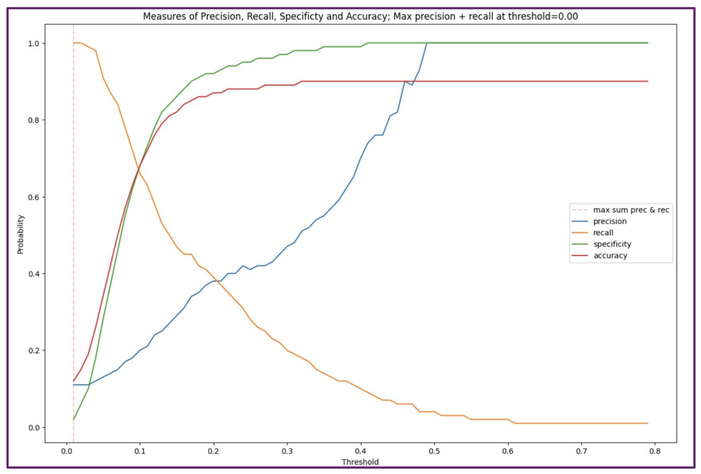
AutoGluon

	Not Enroll	Enroll
Not Enroll	2001	560
Enroll	124	174

The matrices suggest that AutoGluon is decently better at predicting true positive results (in the case of our dataset, positives are not enrolled and negatives are enrolled) and slightly better predicting true negatives compared to XGBoost.

This can also be seen when calculating the precision, recall, specificity, and accuracy of the two models based on the test data, with the precision and recall of the AutoGluon model on the test data was slightly higher than the XGBoost model, while their specificity and accuracy were equivalent.

The model was then ran on the admission data from 2022 to predict the enrollment likelihood of the student given the other features of the student. The result was that AutoGluon predicted on average a higher likelihood of enrollment within a smaller range compared to XGBoost.



Conclusions

AutoGluon either matched or exceeded XGBoost in terms of predictive and evaluation performance on the dataset, proving it to be a powerful framework for machine learning. With its automatic hyperparameter tuning, data processing, and model selection, it is easy to use while still producing high quality models and data analysis. In addition, AutoGluon features XGBoost as one of the model it trains without needing any model tuning to produce a quality model, making it the clear choice between the two machine learning frameworks.

However, it should be noted that the previous project was ran using a cloud EC2 instance while this project was ran entirely on a local machine, meaning there is a possible time variable that may change the value of using AutoGluon over XGBoost. It may be that AutoGluon may have had a longer computational time than XGBoost if using Sagemaker instead of its own Python library due to all of the different models it has to train. Regardless, AutoGluon being a more user friendly framework to use and the fact it never underperformed compared to XGBoost makes it the more attractive framework to utilize.

Technologies Used

- Juypter Notebook - a web-based documents that combine computer code with rich-text elements
- Python - a high-level, general purpose programming language.
- AutoGluon - an open-source machine learning library for Python 3.8+. This project used AutoGluon.tabular due to the tabular format of the dataset
- Pandas - a data manipulation and analysis library for Python
- Sklearn - a machine learning library for Python. Sklearn.metrics was used for creating the confusion matrices.

Further Readings

AutoML for Image, Text, Time Series, and Tabular Data. https://auto.gluon.ai/stable/index.html

Machine learning with AutoGluon, an open source AutoML library. https://aws.amazon.com/blogs/opensource/machine-learning-with-autogluon-an-open-source-automl-library/

AutoGluon vs. XGBoost — Will AutoML Replace Data Scientists? https://towardsdatascience.com/autogluon-vs-xgboost-will-automl-replace-data-scientists-dc1220010102

Soltys M, Dang H, Reilly GR, Soltys K. Enrollment Predictions with Machine Learning. *Strategic Enrollment Management Quarterly*. 2021;9(2):11-18. https://proquest.ezproxy.csuci.edu/login?url=https://www.proquest.com/scholarly-journals/enrollment-predictions-with-machine-learning/docview/2606939441/se-2