**PyCaret Model Comparison**

Jean P. Melendez Villanueva

University of Maryland

Data 660 Advanced Topics in Data Science

Professor Ed Herranz

**PyCaret Model Comparison**

**Introduction**

In recent years, the rise of AutoML frameworks has revolutionized the field of machine learning by streamlining the model development process. PyCaret, a low-code machine learning library in Python, is one such tool that simplifies workflows through automation. Its ability to rapidly prototype, compare, and evaluate models makes it especially useful for analysts and data scientists working under time constraints. For this assignment, PyCaret is used to compare classification models built on the same dataset used in Unit 5.

The dataset, derived from a Portuguese bank marketing campaign, includes customer attributes and campaign outcomes to predict whether a client subscribed to a term deposit. This binary classification problem poses challenges due to class imbalance, which can lead to biased model performance toward the majority class. To address this, a variety of models were tested using PyCaret’s compare\_models() function, which evaluates them across multiple metrics including accuracy, recall, and area under the curve (AUC). These models were then evaluated on a hold-out test set to determine their effectiveness in handling imbalanced classes.

By leveraging PyCaret’s automation features, this assignment seeks to identify the models that not only provide strong overall performance but also balance sensitivity and specificity for the minority class. The comparison includes models such as Logistic Regression, LightGBM, Random Forest, XGBoost, and Support Vector Machines. Performance metrics such as accuracy, balanced accuracy, F1-score, and confusion matrix-derived insights were used to guide evaluation. The report concludes with a reflection on PyCaret’s value as an AutoML tool in contrast to the manual ensemble models built in Unit 5.

**Model Results and Comparison**

Using PyCaret’s compare\_models() function, five classification models were automatically trained, evaluated, and ranked based on performance: Logistic Regression, LightGBM, Random Forest, XGBoost, and Support Vector Machines. This comparison was executed on the bank marketing campaign dataset, where the goal was to predict whether a client would subscribe to a term deposit. The models were evaluated using cross-validation metrics such as accuracy, AUC, recall, precision, and F1-score. These metrics provided an initial understanding of how each model performed during training.

As shown in Figure 1, Logistic Regression achieved the highest accuracy (90.21%) with a strong recall and balanced F1-score. LightGBM, however, showed the highest AUC (0.9075) and the best F1-score (0.8929), suggesting its strength in handling both false positives and false negatives. XGBoost followed closely, showing excellent all-around performance with slightly lower accuracy but a strong balance across evaluation metrics. In contrast, the Support Vector Machine showed the weakest performance among the five, particularly in AUC, indicating sensitivity to imbalanced data and feature linearity.

These top-performing models were then tested on a hold-out dataset using PyCaret’s predict\_model() function. The results are summarized in Figure 2, which displays test accuracy, balanced accuracy, average sensitivity, and average specificity. XGBoost yielded the highest balanced accuracy (0.6903) and demonstrated symmetrical sensitivity and specificity, suggesting it was the most effective at recognizing both classes. LightGBM followed with similarly strong balanced metrics, while Logistic Regression, though still robust, lagged in recognizing the minority class as effectively. The Random Forest model and SGDClassifier delivered solid, but slightly weaker, overall performance.

**Handling of Class Imbalance**

A major challenge with the bank marketing campaign dataset is the uneven distribution in the target variable, where most clients did not subscribe to a term deposit. This imbalance can cause misleading results when relying only on accuracy, since a model could achieve high accuracy by simply predicting the majority class and ignoring the minority class. To avoid this issue, evaluation metrics such as balanced accuracy, sensitivity, and specificity were used to provide a more complete picture of how well each model identified both subscribed and non-subscribed clients.

Of all the models tested, XGBoost handled the class imbalance most effectively. As shown in Figure 2, it achieved the highest balanced accuracy, along with equal values for sensitivity and specificity. This means the model was consistent in correctly identifying both positive and negative cases. LightGBM also performed well, showing reliable and balanced performance across evaluation metrics. While Logistic Regression had strong overall accuracy, its sensitivity and specificity were not as balanced, which could lead to missed opportunities in recognizing potential subscribers.

Random Forest and SGDClassifier showed average performance in terms of class balance, with less consistency across sensitivity and specificity. These outcomes highlight why it's important to evaluate models using more than just accuracy, especially when working with imbalanced data. Models like XGBoost and LightGBM, which are designed to adapt to these challenges, proved to be the most dependable options for predicting subscription outcomes in this campaign.

In real-world applications, especially in marketing and finance, the cost of misclassifying the minority class can be significant. In this case, failing to identify potential subscribers could result in missed revenue opportunities and inefficient resource allocation. Therefore, models that prioritize balanced performance, like XGBoost and LightGBM, are more valuable than those that simply optimize for overall accuracy. Their ability to detect subtle patterns in minority class behavior makes them ideal candidates for deployment in environments where every correct prediction can influence customer engagement strategies or business outcomes.

**Reflection on AutoML Tools**

Working with PyCaret in this assignment highlighted the advantages of using automated machine learning tools for developing and evaluating models. One of the clearest benefits was the ability to quickly test and compare multiple models with only a few lines of code. Instead of writing long scripts for data preparation, model training, and evaluation, PyCaret handled most of the process automatically. This saved a considerable amount of time and allowed for greater focus on analyzing the results and understanding how each model performed.

PyCaret also made it easier to evaluate models using meaningful metrics beyond just accuracy. The platform automatically included measures like recall, precision, F1-score, and area under the curve, which are especially helpful when dealing with imbalanced data. Having evaluated all models using the same framework ensured consistency and made the comparison process much more efficient. In addition, PyCaret brought together a wide range of algorithms from trusted libraries like scikit-learn, XGBoost, and LightGBM, all within a simple and intuitive interface.

That said, there are a few drawbacks to keeping in mind. PyCaret can sometimes hide important details about how models are built, such as how hyperparameters are tuned or how data is preprocessed behind the scenes. This lack of transparency might limit customization and make it harder for users to fully understand what is happening under the surface. Relying too much on automation can also reduce learning opportunities for those trying to deepen their understanding of machine learning. Still, PyCaret proved to be a valuable tool, offering a practical and efficient approach to model comparison and performance evaluation.

**Conclusion**

This assignment highlighted the practical value of using PyCaret to streamline the process of building and evaluating machine learning models. Applying PyCaret to the bank marketing campaign dataset allowed for quick experimentation with multiple classification models, all evaluated using meaningful metrics. Among the models tested, XGBoost and LightGBM stood out not only for their strong accuracy but also for their balanced performance in identifying both subscribed and non-subscribed clients. Their ability to handle uneven class distributions made them especially effective for this task.

Compared to the ensemble models created manually in Unit 5, PyCaret provided a more efficient and organized approach. It significantly reduced the time and effort needed to train and compare different models, while still producing high-quality results. The tool’s ability to handle complex tasks, like managing class imbalance and computing advanced performance metrics made it a helpful resource for developing well-rounded models without writing extensive code.

Although there are limitations to using automated tools, such as reduced control over model tuning and less insight into certain behind-the-scenes processes, the benefits of PyCaret were clear. It offers a powerful and user-friendly way to explore machine learning models, especially when working with structured data and classification problems. Overall, this experience showed that PyCaret can be a reliable and efficient tool for improving both the speed and quality of model development.

These insights are particularly relevant in real-world business environments where timely and accurate decision-making can drive customer engagement and improve marketing outcomes. For instance, banks and financial institutions often rely on predictive models to target clients with personalized campaigns. In such cases, identifying the small group of clients likely to subscribe to a product is more important than achieving high overall accuracy. The models identified through PyCaret, especially XGBoost and LightGBM, offer the precision and balance needed to support these goals and demonstrate how automated machine learning tools can contribute directly to data-driven business strategies.

**References**

Dancker, D. (2025). *Generative AI in data science*. University of Maryland Global Campus Library.

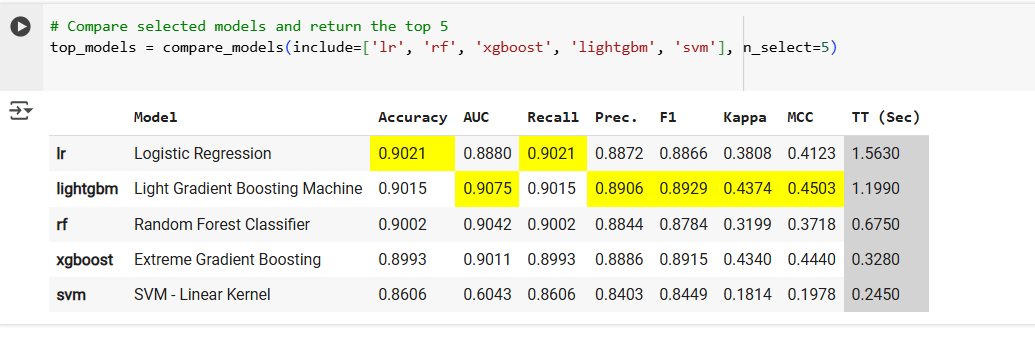
Marr, B. (2024). *How generative AI will change jobs in healthcare*. Forbes. https://www.forbes.com/sites/bernardmarr/2024/06/05

PyCaret. (2025). *PyCaret documentation*. https://pycaret.gitbook.io/docs/

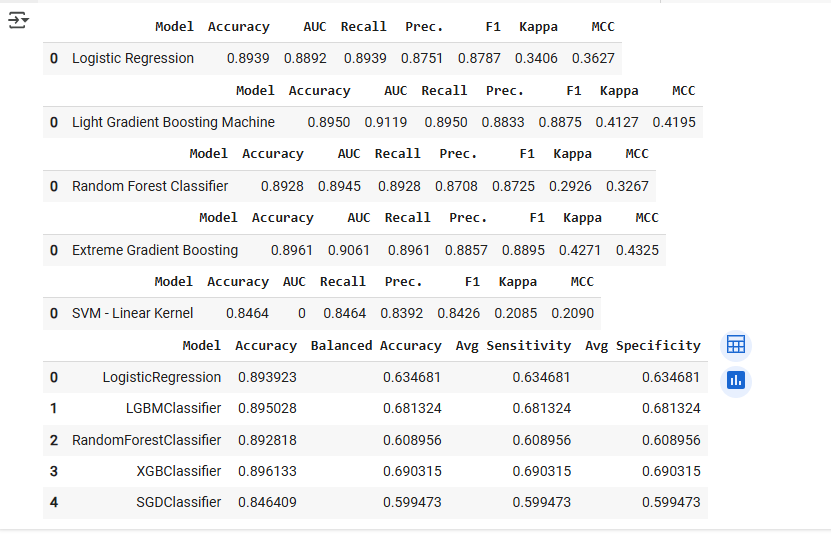
**Appendix**

**Appendix A –** Refer to attached .ipynb code

**Appendix B -** Screenshots



(Figure 1) Top 5 models ranked by cross-validation metrics (accuracy, AUC, F1-score) using PyCaret's compare\_models().



(Figure 2) Test set performance of the top five PyCaret models. Metrics include accuracy, balanced accuracy, sensitivity, and specificity, highlighting each model’s ability to handle class imbalance.