AutoML - AWS Sagemaker Studio

- Using every feature in the dataset with little to no cleaning for predicting total_lift resulted in a very inaccurate model (huge RSME, low R^2)
- After feature engineering and data cleaning, model performance was much better, though still comparable to manual model selection/training.
- Data pre-processing is still just as important with tools like these because the full set of features may not be conducive to high model metrics.
- Top 5 features:
    - Weight (~27%)
    - Fran (~20%)
    - Grace (~17%)
    - Gender (~16)
    - Height (8%)
- There exists a model leaderboard for the AutoML job – which includes performance and latency – but difficult/impossible to understand the type of model attached to the metrics (screenshot available).

H2O AutoML – Local

- Once again, utilizing a subset of features collected during data engineering/pre-processing yielded best results.
- Model accuracy and R^2 metrics are still comparable to the manual model selection process
- **Top 3 (Performance**)
    - StackedEnsembleAll (**RMSE**: 95.66)
    - StackedEnsembleFamily (**RMSE**: 95.93)
    - GBM1 (**RMSE**: 97.00)
- **Top 3 (Latency)**
    - GBM5 (**Training Time**: 109 ms)
    - GBM1 (**Training Time**: 193 ms)
    - GBM2 (**Training Time**: 227 ms)

- AutoML in Sagemaker Studio is completely no-code. With just a UI, I could join datasets, perform basic data cleaning/engineering, and run an AutoML job. H2O, on the other hand, was full-code. I utilized the H2O Python module to create an AutoML job and analyze the leaderboard after training.