BrisT1D Blood Glucose Prediction Competition

1st place solution

A. MODEL SUMMARY

A1. Background on you/your team

- **Competition Name:** BrisT1D Blood Glucose Prediction Competition
- **Team Name:** Sebastian Cuya
- **Private Leaderboard Score:** 2.3615
- **Private Leaderboard Place**: 1st

Team Member 1:

- **Name:** Sebastian Cuya
- **Location:** Lima. Peru
- **Email:** sebastiancuya20.50@gmail.com

A2. Background on you/your team

- What is your academic/professional background?
  - I'm a Bachelor of Industrial Engineering from the University of Lima.
  - I'm currently working as an AI & Data Senior Consultant
- Did you have any prior experience that helped you succeed in this competition?
  - I had participated in a couple of competitions in the past
- What made you decide to enter this competition?
  - I was interested in putting into practice what I had learning in the MIT MicroMasters Program in Statistics and Data Science, especially for time series forecasting
- How much time did you spend on the competition?
  - About an average of 2 hours per day during these past 2 months (60 hours approximately)
- If part of a team, how did you decide to team up?
  - NA
- If you competed as part of a team, who did what?
  - NA

A3. Summary

- The training method I used is a single LightGBM.
- The most important features were the lags from blood glucose from the past 30 minutes, and

the 5-minute difference between measures

- The tool I used was Python (and associated packages)
- It takes about 5 minutes to prepare the data, 2 hours to train, and 3 second to predict

A4. Features Selection / Engineering

- What were the most important features?
    - Fig.1 and Fig.2 are feature importance by gain and SHAP values on prediction

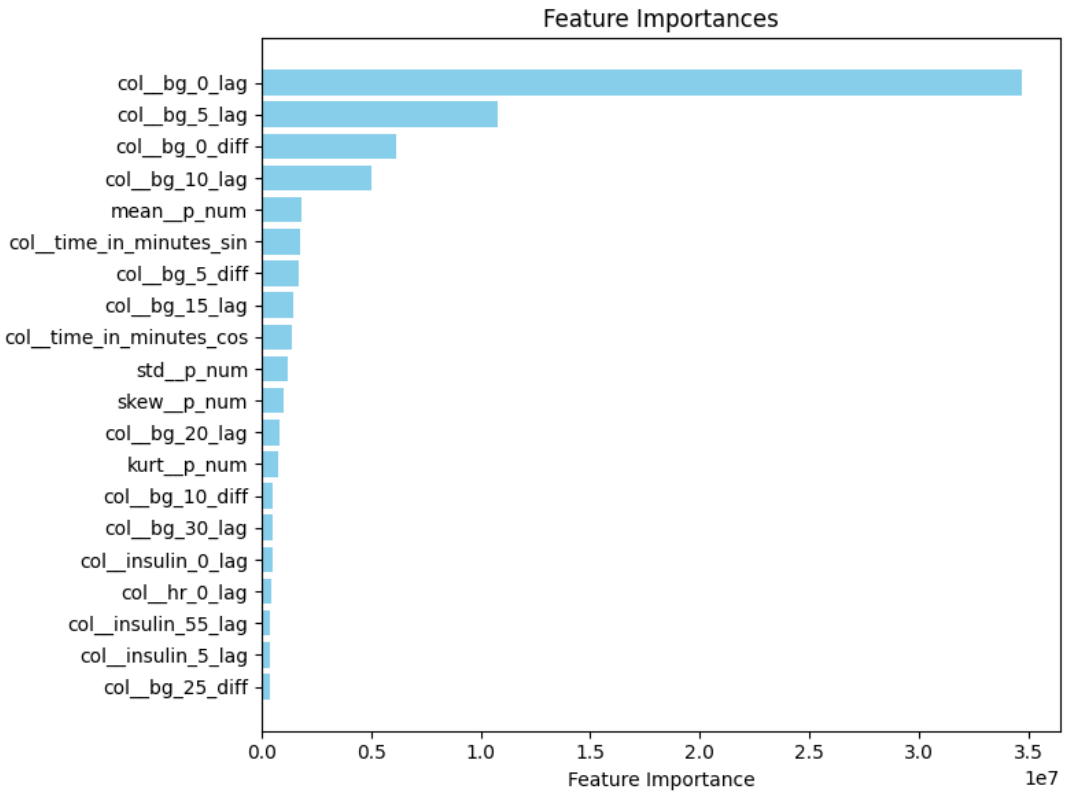

Fig.1 variable importance plot by gain

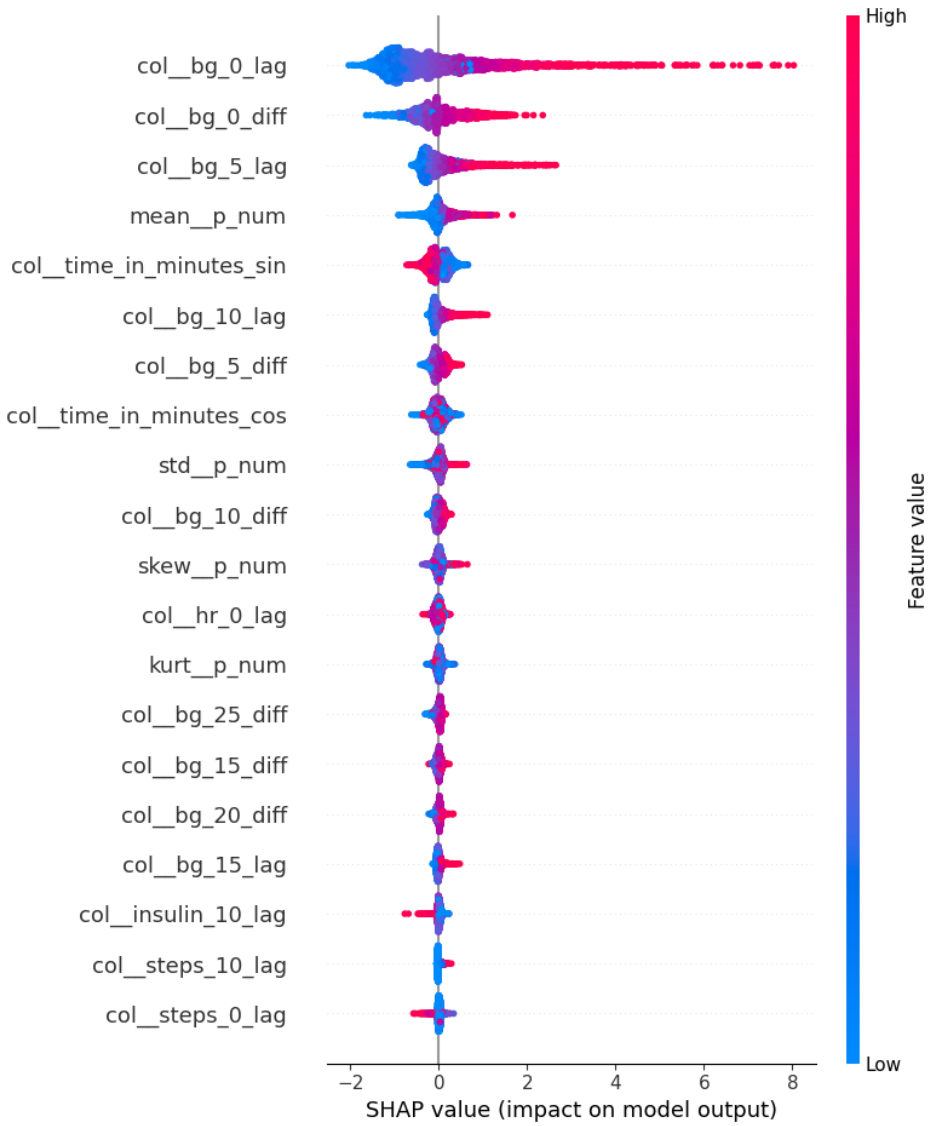

Fig.2 SHAP feature importance

- How did you select features?
    - Through statistical analysis: Stationarity analysis; autocorrelation, partial autocorrelation and cross correlation plots.
    - Through common feature engineering techniques (target encoding, indicators)
    - Through feature importance plot and SHAP values after initial tests
    - Through domain knowledge
- Did you find any interesting interactions between features?
    - Blood glucose was long-term stationary but short term non-stationary
    - Autocorrelation and partial autocorrelation plots showed strong predictive power for blood glucose from the 2 immediate previous lags (5-minute gap). Something similar,

but less evident was shown for 15-minute gaps. Consequently, that pattern was weaker for larger gaps (30-minute, 1-hour)
    - Cross correlation plots showed medium predictive power from up to the previous 2 to 3 hours of insulin doses and 30 min – 1 hour of carbohydrate intake.
    - Cross correlation plots show low predictive power from up to the previous hour of activity-related features
    - Given that the forecast horizon was 1 hour into the future, these interactions were extrapolated in order to have predictive power throughout the missing hour in the forecast horizon.
- Did you use external data? (if permitted)
    - No

A5. Training Method

- What training methods did you use?
    - A single LightGBM.
- Did you ensemble the models?
    - No
- If you did ensemble, how did you weight the different models?
    - NA

A6. Interesting findings

- What was the most important trick you used?
    - Applied target encoding but with different statistics to outline blood glucose distributions (mean, standard deviation, skewness, kurtosis)
    - Applied trigonometric transformation to the time (in minutes)
    - Created an indicator for the sampling time gap of blood glucose per patient (5 or 15 minutes)
    - Increased by a factor the number of estimators for the final model based on the trend of the cross validation early stopping rounds.
- What do you think set you apart from others in the competition?
    - Expanded test data and used it with the training data to have a bigger dataset. Windows of 1 hour of data were generated.
    - Created a custom cross validation procedure whose folds were aligned with the test data's sampling procedure and distribution, making it robust and reliable
- Did you find any interesting relationships in the data that don't fit in the sections above?
    - Insulin dose was a good feature up to the last 2 hours, there was a decision to make

between expanding more data and having 2 reliable hours of insulin dose. There was a tradeoff, however, since most of the predictive power was within the last hour, just that hour was considered.

- Carbohydrate intake showed more immediate effects (30 min or so contained predictive power); however, since it was correlated with insulin doses spikes (bolus), the entire first hour back was kept, and provided better results.
- Activity description and calories, when taken together as features, provided better results, meaning that there could have been a strong relationship between both

A7. Simple Features and Methods

- Is there a subset of features that would get 90-95% of your final performance? Which features?
  - Even though the model uses 89 features, it just depends on 1 single model and LightGBM is fast by design, making the model already perform well.
  - To achieve 90-95% of results, I would remove all activity-related features (steps, hear rate, calories, activity description). The number of features would decrease from 89 to 41 features
  - If I had to keep at most 10 features, I would just keep:
    - Blood glucose from the last 15 minutes (point-in-time and difference)
    - Blood glucose mean as a target encoding feature
    - Time trigonometric features
- What model that was most important?
  - I used a single LightGBM model
- What would the simplified model score?
  - At most 10 features (this will be considered the simplified model):
    - 2.4330 public (3.76% error increase)
    - 2.4489 private (3.70% error increase)

A8. Model Execution Time

- How long does it take to train your model?
  - About 2 hours of hyperparameter tuning + 1 minute of fitting
- How long does it take to generate predictions using your model?
  - About 3 seconds
- How long does it take to train the simplified model?
  - About 20 minutes of hyperparameter tuning + 10 seconds of fitting
- How long does it take to generate predictions from the simplified model?

- About 3 seconds

A9. References

[1] https://pmc.ncbi.nlm.nih.gov/articles/PMC10135844/pdf/bioengineering-10-00487.pdf

[2] https://pmc.ncbi.nlm.nih.gov/articles/PMC8224858/pdf/pone.0253125.pdf

[3] https://www.nature.com/articles/s41598-021-03341-5