Using Machine Learning to Evaluate Food Category Feature Importance in Relation to Change in Blood Glucose Levels mmol/L

Jon Golding
School of Continuing Studies
York University
Toronto, ON, Canada
jgolding31@yahoo.com

ABSTRACT

Diabetes blood glucose monitoring has seen technological advancements over the past two decades to include continuous blood glucose monitoring via sensor inserted under a user's skin, often on the back of the arm, or the stomach region.

While proprietary smart phone apps developed by CGM manufacturers have the functionality to collect sensor data and map descriptive statistics over time such as average blood glucose in mmol/L, or percent time-in-range, there still does not exist a mechanism to allow diabetic CGM-users access to AI insights that demonstrate how certain food choices may relate to directional changes in blood glucose levels.

The objective of this study is to determine if machine learning can be used to predict food feature importance as it relates to predicting change in blood glucose mmol/L levels over time.

9367 time-stamped instances of data were collected from the Author's personal CBM from March 10, 2023 through to July 2, 2023. Over 400 different food categories were logged using text-based notes via sensor-paired phone app for each timestamped instance where food intake was logged.

ML MODELING STRATEGIES :

Two different machine learning models were applied to compare the utility of model-based feature importance:

1. XGBOOST Regressor: A decision-tree-based machine learning algorithm which can be well-adapted for the use-case of modelling multivariate time-series data. XGBOOST uses gradient boosted trees to make y-variable predictions through combining individual decision trees to form a strong learner.[1] The XGBOOST model is able to handle multivariate datasets, as well as non-linear problems
2. Temporal Fusion Transformer Model: A type of deep learning algorithm specifically designed to work with time-series data in order to capture temporal dependencies and patterns across time.[2] The TFT model is also able to handle multivariate datasets, in addition to non-linear problems.

It should be noted that a significant challenge arises when using machine learning to work with time series data: machine learning strategies often shuffle data instances or draw inferences in ways that may not preserve or capture the temporal dependencies between ordered time steps. This can lead to feature leakage where past data points are trained on information extracted from future data points leading to inaccurate modeling and feature importance conclusions.

The methods used in this study take careful measures to capture temporal dependencies between ordered timesteps:

XGBOOST regressor:

1. The timestamp data was parsed into distinct feature column subsets such as 'hour', 'day of week', 'day of month' to capture cyclical seasonality of the data.
2. The integer-based index was replaced with the timestamp itself for a specific-time based reference in date-time format. It should be noted that the date-time-stamp could not be used as in input feature for the model itself given its format; the timestamp was retained as an important reference for each data instance in the index column.
3. A time_idx column was added as a feature variable so that whole number integers were assigned to each data instance in chronological order. This feature proved to be important during the model training processes.
4. Time-based cross validation was applied by progressively using part of the data for training and a subsequent part for testing in an ordered structure. This allowed the

---

[1] Adapted from Grogan, M. (2021, September 8). *XGBoost for Time Series Forecasting: Don't Use it Blindly.* Towards Data Science.

[2] Adapted from OpenAI. (2023). Chat GPT-4. [Large Language Model]

XGBoost model to validate the learning of temporal dependencies of the data during the model training process.[2]

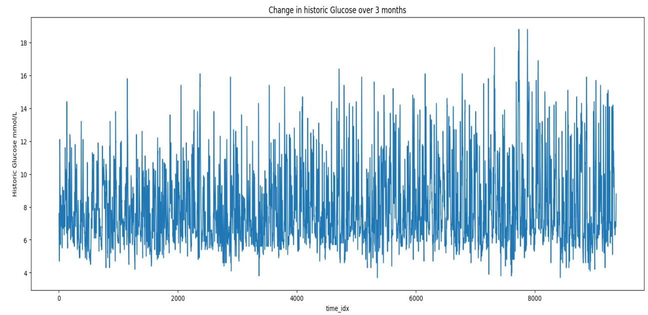
5. Creating lagged variables that were applied to the target variable 'Historic Glucose mmol/L. Each instance of time 't', used 't-1' through to 't-5' lagged variables. This process allows the XGBoost model to learn lagged feature temporal dependency information when making future predictions. This is a rolling window approach to time-series modeling.[3]

Temporal Fusion Transformer Model:

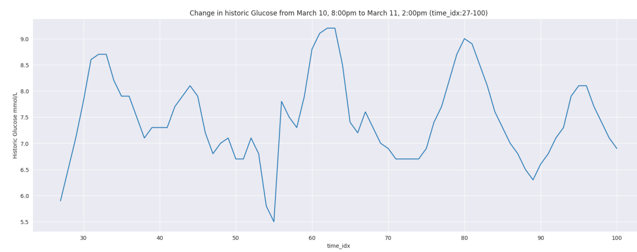
1. The timestamp data was parsed into distinct feature column subsets such as 'hour', 'WeekDay' (day of week), 'day of month' to capture cyclical seasonality of the data.
2. The integer-based index was replaced with the timestamp itself for a specific-time based reference in date-time format. It should be noted that the date-time-stamp could not be used as in input feature for the model itself given its format; the timestamp was retained as an important reference for each data instance in the index column.
3. A time_idx column was added as a feature variable so that whole number integers were assigned to each data instance in chronological order. This feature was shown to have high feature importance ranking during the model training process.
4. The TFT model makes use of a lookback window and prediction window process which is a 'rolling window' approach to time-series modeling. For the TFT model, a variety of window sizes were trialed with varying results. Best modeling results were obtained with a lookback window of 58 * 24 hours (58 days), and a prediction window of 48 hours.
5. TFT modeling takes into account two distinct categories of feature types for input encodings: known real-time variables, and unknown real-time variables. Known real-time variables are features where future values are known at the time of prediction.[4] For the specific use-case of this model, the known real-time features were listed as 'time_idx', 'Hour', 'day_of_month', 'WeekDay' (day of week), and 'Month'. All remaining variables (distinct food category features) were encoded as unknown real-time variables.

EXPLORATORY DATA ANALYSIS:

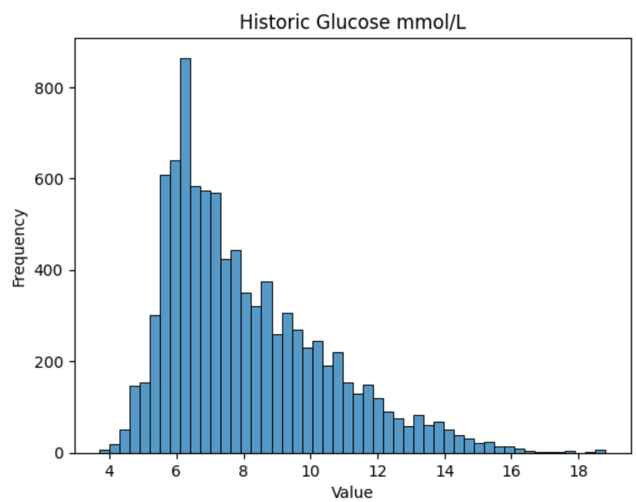
Historic blood glucose mmol/L is plotted as a function of time over a 3-month span showing high volatility:



In order to capture further context of target variable change within a shorter time window, a 30-hour sample of recorded time stamp data was plotted below showing Historic Blood Glucose mmol/L as a function of time:



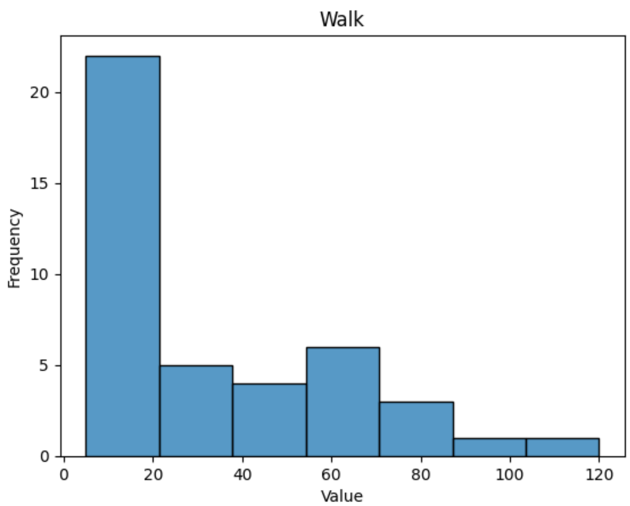
Shown below is a histogram of the target variable data distribution. Given the right skew, a square root transform was later applied when modeling to normalize the distribution:

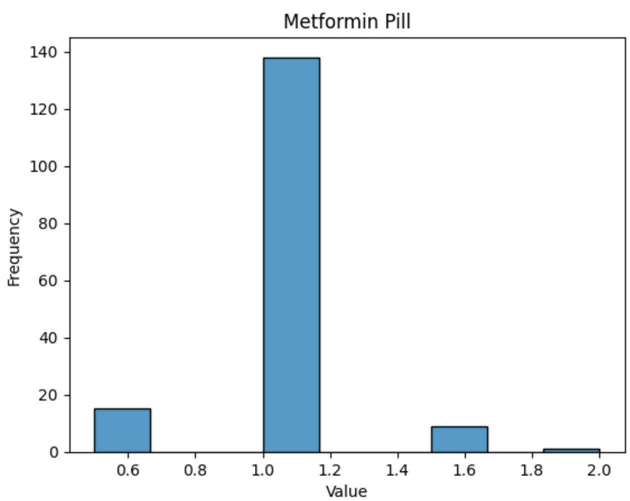


Shown below are the histograms for continuous input feature variables 'Walk' (units are in minutes) and 'Metformin Pill' (units are in 500 mg pills):

[3] Adapted from OpenAI. (2023). Chat GPT-4. [Large Language Model]

[4] Adapted from Larzalere, B. (2023, July 27). *Unlocking Insights from Multivariate Data Using theTemporal Fusion Transformer.* Towards AI.



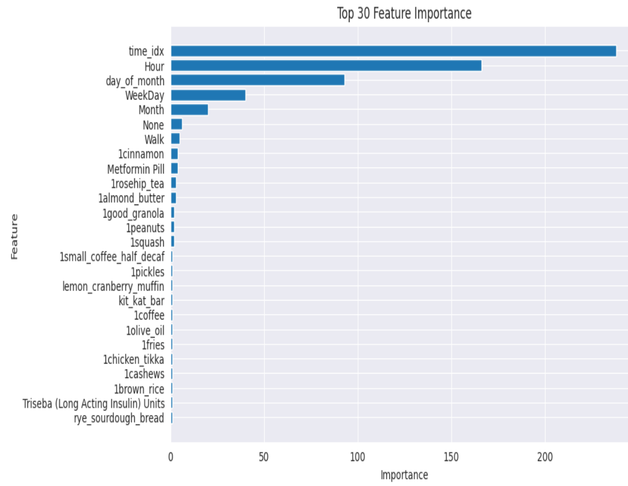


DATA PREPARATION:

Distinct food categories that were logged at the time of each food intake entry (via smart phone app), were later processed into a feature column labelled 'Non-Numeric Food Category'. In order to produce an input language for the machine learning models to process, food category labels were parsed into distinct columns and one-hot-encoded using binary values where 0 represented no food intake, and 1 represented food intake.

The above-mentioned process created over 400 food category columns, and the resulting modeling process was computationally expensive.

In order to address the problem of high compute-time, final food category feature selection was completed using an XGBOOST algorithm and Scikit-learn's Time-Series-Split module to determine ranked feature importance while accounting for temporal dependencies:



FEATURE ENGINEERING:

The following feature columns were extracted from the timestamp and used as features in the data frame matrix to help with model predictions and to account for cyclical seasonality: 'Hour', 'WeekDay'(day of week), 'day_of_month', and 'Month'.

A time_idx column was also created and added to the data frame matrix in order to account for the temporal order of data point instances. Each instance was assigned a whole number integer in chronological order within this feature column.

For the XGBOOST model, 5 time-lag feature columns were created and added to the data frame matrix to account for the historical values of the 'Historic Glucose mmol/L' feature column at different points in the past. These lagged features are able to capture the glucose levels from the previous time steps and provide information about the temporal patterns and dependencies in the data.[5]

In the case of the temporal fusion transformer model, lag-based feature columns were not explicitly created. Instead, the strategy of using encoded lookback and prediction window lengths was used to implicitly account for the specified number of previous time steps when making model predictions.

MODEL EVALUATION METRICS:

[5] Adapted from OpenAI. (2023). Chat GPT-4. [Large Language Model]

Several of iterations of both the XGBOOST and the TFT regressor models were trialed in order to fine tune for higher predictive power when explaining the variance of the target variable y: 'Historic Glucose mmol/L'.

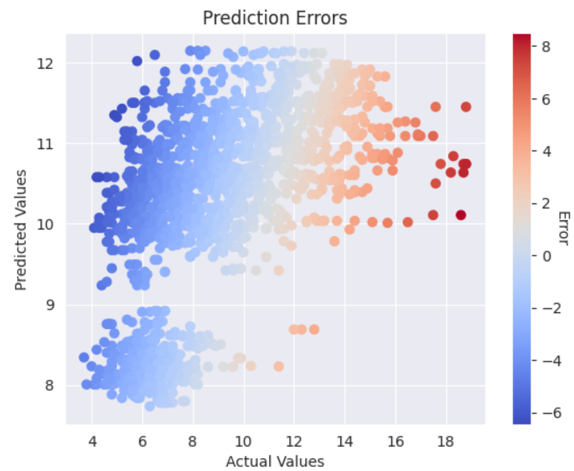
For the XGBOOST Regressor:

1. Modeling was first completed using the following parameters:
   a) Top 26 features were used in input matrix X, including time-based feature columns to capture seasonality.
   b) Scikit-learn's TimeSeries split module was used for cross-validation with n_splits = 5.
   c) Hyperparameters were tuned to obtain the following optimized hyperparameters: learning rate = 0.1, max_depth = 7, colsample_bytree = 0.5, min_child_weight = 5, subsample = 0.5

Evaluation metrics for the above model showed significant overfitting with high variance given the lower r-squared metric seen in the validation dataset when compared to the train set. The negative r-squared result suggests that the model with the above parameters performs slightly worse than a baseline model using the mean of the target variable (Historic Blood Glucose mmol/L when capturing the underlying patterns in the data: [6]

1) Train R-squared: 0.671
   Validation R-squared: -.092
2) Train MAE: 0.952
   Validation MAE: 2.558
3) Train RMSE: 1.264
   Validation RMSE: 3.022

Below a plot of the above model's prediction errors showing no discernable pattern suggesting poor model performance:



XGBOOST Regressor model, 2nd iteration:

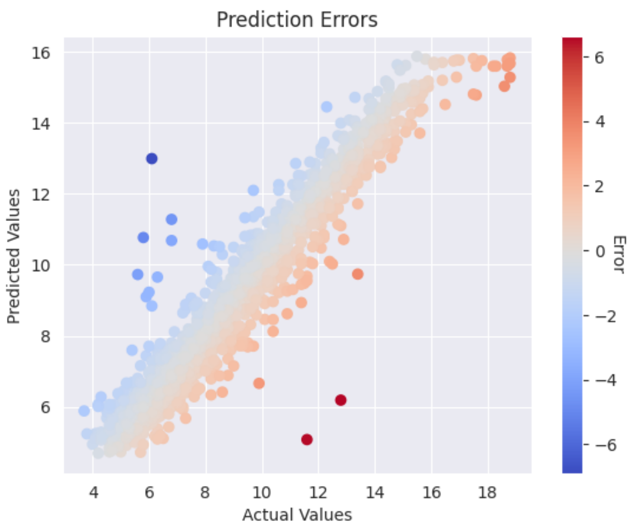
a) 5 lag feature variables were feature engineered and added to the X matrix data frame.
b) With the added lag feature variables, input Matrix X contained 31 features total.
c) Time Series Split module remained unchanged.
d) Previously optimized hyperparameters remained unchanged.

Evaluation metrics showed significant model improvement with lower prediction errors and an r-squared of 0.929 as seen for the validation dataset below:

1) Train R-squared: 0.938
   Validation R-squared: 0.929
2) Train MAE: 0.374
   Validation MAE: 0.503
3) Train RMSE: 0.459
   Validation RMSE: 0.766

Below, a plot of the XGBOOST model prediction errors when lag feature variables were engineered and added to input matrix X:

[6] Adapted from OpenAI. (2023). Chat GPT-4. [Large Language Model]



Temporal Fusion Transformer Model:

2. Modeling was first completed using the following parameters:
   a) Manual train-validation split was applied at the 80% point of the ordered data frame (no shuffle was applied in order to preserve the temporal order of the input data).
   b) The following time-varying known reals were encoded separately from the food category feature columns, and used as their own input category group: [‘time_idx’, ‘Hour’, ‘day_of_month’, ‘WeekDay’, ‘Month’.
   c) The remaining feature columns were encoded as ‘time varying unknown reals’: [‘None’, ‘Walk’, ‘1cinnamon’, ‘Metformin Pill’, ‘1rosehip_tea’, ‘1almond_butter’, ‘1good_granola’, ‘1peanuts’, ‘1small_coffee_half_decaf’, ‘1pickles’, ‘lemon_craberry_muffin’, ‘kit_kat_bar’, ‘1coffee’, ‘1olive_oil’, ‘1fries’, ‘1chicken_tikka’, ‘1chashews’,’1brown_rice’, ‘cod’, ‘rye_sourdough_bread’].
   d) The lookback window length was defined as 24 *58 hours. This window length was found to be optimal after several iterations through a trial-and-error process. The lookback window refers to the amount of past data the model considered when making a prediction
   e) The prediction window was defined as 48 hours. This window length was found to be optimal after several iterations through a trial-and-error process. The prediction window defines the number of data points the model is trying to predict after training.
   f) The target variable ‘Historic Blood Glucose mmol/L’ was transformed (to normalize the distribution) using a softplus transformation function.
   g) The following hyperparameters were used as a starting point when training the model: Learning rate =0.001, hidden_size = 128, attention_head_size = 4, dropout = 0.1, hidden_continuous_size = 128, output_size = 7, loss = QuantileLoss, reduce_on_plateau_patience =4.

Evaluation metrics for the above TFT model were performed on the validation dataset. R-squared showed as being slightly better than 0 suggesting the model performed marginally better than a baseline model that predicts the mean of the target variable.[8]

1) Validation dataset R-squared: 0.12
2) Validation dataset Mean Absolute Error (MAE): 1.30
3) Validation dataset Root Mean Squared Error (RMSE): 1.53

Temporal Fusion Transformer Model, 2nd Iteration:

a) Hyperparameter tuning was applied using a Bayesian Optimization algorithm.
b) Optimized hyperparameters were applied to the model parameters: dropout = 0.1, learning_rate = 1e-05, transformation(function) = sqrt.
c) All other model previously used TFT model parameters remained the same.

Evaluation-metrics for the hyperparameter-tuned TFT model showed a noteworthy improvement in the R-squared metric with a score of 0.56. This showed that a substantial portion of the model’s variability can be attributed to the predictor input variables when hyperparameters are tuned. There is, however, enough residual variability remaining that the model still needs to be improved:[8]

1) Validation dataset R-squared: 0.56
2) Validation dataset Mean Absolute Error (MAE): 0.90
3) Validation dataset Root Mean Squared Error (RMSE): 1.08

PLOTTING FEATURE IMPORTANCE:

[8] Adapted from OpenAI. (2023). Chat GPT-4. [Large Language Model]

2 types of feature importance plots were used to evaluate ranking of feature importance for model predictions:

1) SHAP or SHapeley Additive exPlanations was used to plot feature importance for the XGBOOST model. SHAP works well with tree-based models such as XGBOOST because it inherently provides feature importance based on the structure of decision trees.[9]

SHAP has some notable advantages over other methods:

a) Plotting directional feature importance is possible when using SHAP.

b) SHAP is versatile in that it can be used to explain local feature importance for individual predictions, as well as global feature importance for predictions across the entire dataset.

c) SHAP takes into account the interactions between features when assigning feature importance scores.

SHAP does have the disadvantage that is not especially well-suited for TFT model applications due to the complex architecture and inner workings of the neural network layers. As seen below, TFT modeling makes use of a custom-tailored feature importance package built into the TFT model itself.[10]

2) The Temporal Fusion Transformers package (in python) includes built-in feature importance capabilities for predictive modeling. Model output is interpreted using the sum reduction technique.

Advantages of the built-in TFT package:

a) A ready-made TFT package component that fits the use-case of the model architecture without having to retrofit an outside feature importance explanation method.

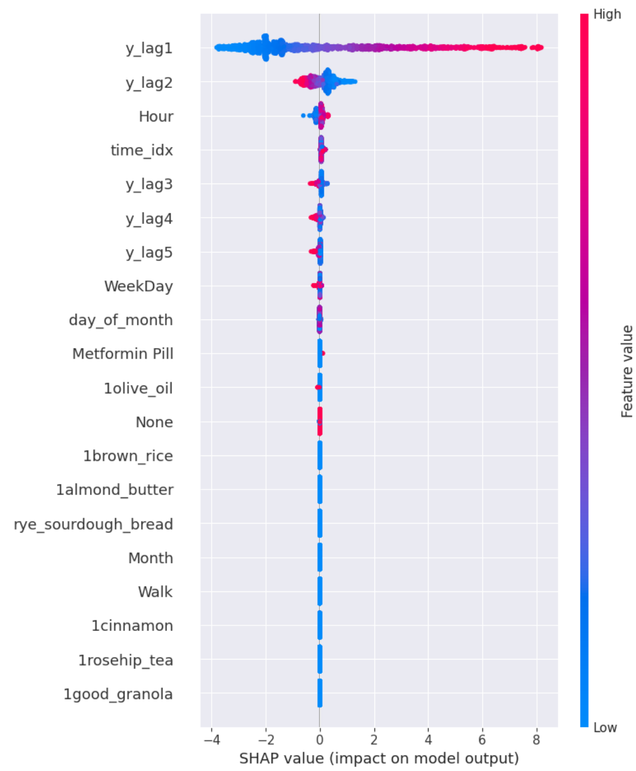
b) The plots are easily interpretable.

c) The feature importance plot of the time varying known reals (time-based features) is plotted separately from the remaining feature variables. This can also be interpreted as a potential disadvantage, but is noted as an idiosyncrasy within the TFT package.

One disadvantage of the TFT package is that it does not have the capability to plot directional feature importance, which is a capability that SHAP does have.

SHAP FEATURE IMPORTANCE IN THE XGBOOST MODEL:

Shown below is a plot of directional feature importance using SHAP for the final XGBOOST model where an R-squared of 0.929 was obtained on the validation set. The plot contains time-based features, lag-based features, and food intake features. Feature importance is ranked using the validation dataset which is an indicator of how the model would potentially attribute feature importance on unseen data with similar input features:

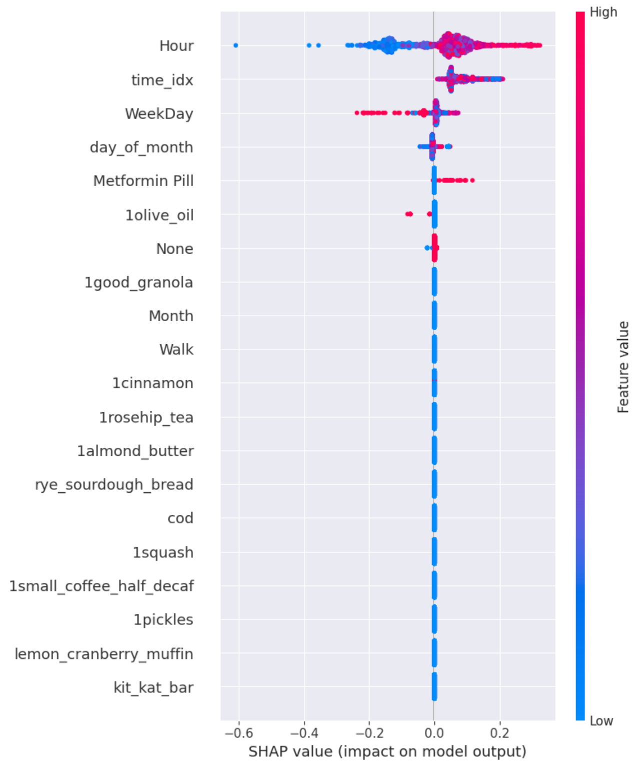


From the plot above, it is apparent that the food-based feature variables are weighted with less importance compared to the time-lag features and the seasonality time-based features. This was an important discovery with insight into how the XGBOOST model attributed feature importance rankings when making its predictions on blood glucose levels.

In the following SHAP plot, lag-based variables were filtered from the plot and the scale was decreased to show a more granular view of the food category variables. It should be noted that while the plot was filtered to remove the lag variables, the XGBOOST model itself remained unchanged:

[9] Adapted from OpenAI. (2023). Chat GPT-4. [Large Language Model]

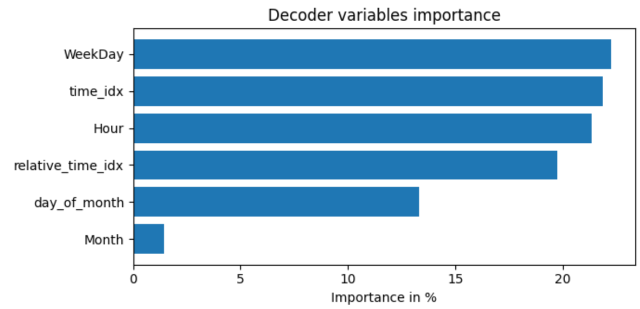
[10] Adapted from OpenAI. (2023). Chat GPT-4. [Large Language Model]

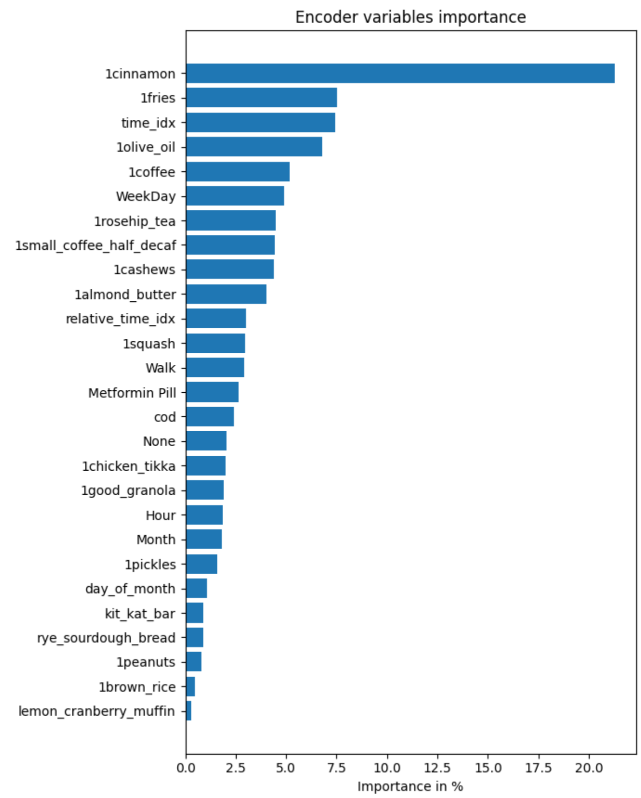


Given the granularity of the above SHAP values (impact on model prediction outputs), it is difficult to attribute definitive directional feature importance for the selected food category features as they relate to the target variable 'Blood Glucose mmol/L.' The possibility that the directional attributions for the food category features being influenced by model noise remains high given the level of granularity shown above (see x-axis unit scale). While the data used in the feature importance plot was from the validation set, the plot was congruent with similar SHAP plots produced on the training set.

FEATURE IMPORTANCE USING THE TFT MODEL:

Shown below are the feature importance plots for the final hyperparameter-tuned TFT model where an R-squared of 0.56 was obtained on the validation dataset:





It was encouraging that encoder variable importance plot showed a more robust feature importance weighting for the various food categories in the TFT model when compared to the XGBOOST model. Some important considerations, however, need to be taken into account:

1) Given that only time-based features are shown in the decoder variable feature importance plot, this suggests that much like XGBOOST, the TFT model is largely reliant on time-based features to make future predictions.[11]

2) Both time-based features and food category features were present with robust percentage importance values as found on the x-axis of the encoder-variables importance plot. This suggests that food categories are important when evaluating historical input data that the

[11] Adapted from OpenAI. (2023). Chat GPT-4. [Large Language Model]

TFT model trains on in order to capture temporal patterns and dependencies.[12]

3) While the R-squared metric was overall lower on the validation dataset for the TFT model when compared the XGBOOST model, the prediction task for the TFT model was more complex given the addition of the prediction window size of 48 hours.

RECOMMENDATIONS FOR NEXT STEPS:

a) Implement further hyperparameter tuning for the TFT model parameters such as number of neurons and hidden layers. Minimal hyperparameter tuning was initially completed for 'dropout', 'learning rate', and transformation function parameters resulting in noticeable model improvements as seen in the TFT evaluation metrics.

b) Consider data augmentation techniques to see if the organic data can be captured and augmented in a way that adds robustness and value for the TFT and XGBOOST models to further train on.

c) Consider using LIME (Local Interpretable Model Agnostic Explanations) and SHAP for plotting local feature importance rankings with direction for the higher ranked food category features such as 'cinnamon' found in the TFT encoder variables plot.

CONCLUSION:

Determining directional feature importance for regression models in the domain of time series algorithms is not commonly found as a primary task or end-goal in itself; rather the emphasis is often on the goal of producing model predictions. Built-in model packages for feature importance or custom applied feature importance algorithms are then applied after model training in order to plot feature importance rankings. The task of considering directional feature importance as a primary goal when looking at target variable change in time series regression models, does shift how modeling use-cases and problem definitions are defined. More work in this emerging area of machine learning could prove useful for future research applications across multiple domains, including that of personalized food category feature importance in diabetes research.

ACKNOWLEDGMENTS

The use of a pre-defined temporal fusion transformer model architecture that could be specifically adapted for the use-cases found of this study, was essential to the author's completion of many of the above tasks and subsequent conclusions.

Coding templates and TFT modeling architecture were used and adapted from Brent Larzalere's article: '*Unlocking Insights from Multivariate Data with the Temporal Fusion Transformer'* published in Towards AI.

XGBOOST model code was generated and iteratively adapted through a process of prompt engineering using OpenAI's Chat GPT-4.

REFERENCES

[1] Larzalere, B. (2023, July 27). *Unlocking Insights from Multivariate Data Using the Temporal Fusion Transformer*. Towards AI. https://pub.towardsai.net/unlocking-insights-from-multivariate-data-with-the-temporal-fusion-transformer-b711f0396f57

[2] Grogan, M. (2021, September 8). *XGBoost for Time Series Forecasting: Don't Use it Blindly.* Towards Data Science. https://towardsdatascience.com/xgboost-for-time-series-forecasting-dont-use-it-blindly-9ac24dc5dfa9

[3] Cerliani, M. (2020, March 17). *Feature Importance with Time Series and Recurrent Neural Network*. Towards Data Science. https://towardsdatascience.com/feature-importance-with-time-series-and-recurrent-neural-network-27346d500b9c

[4] OpenAI. (2023). Chat GPT-4. [Large Language Model] https://chat.openai.com/?model=gpt-4

[12] Adapted from OpenAI. (2023). Chat GPT-4. [Large Language Model]