Supplemental Material:

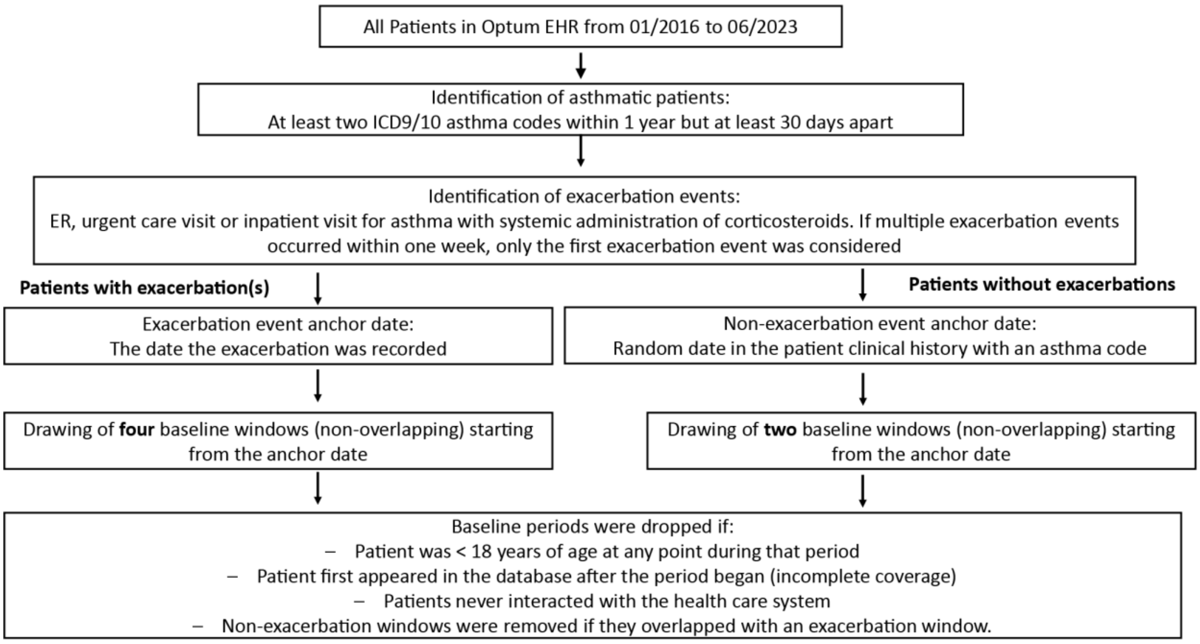
Predicting Asthma Exacerbations Using Machine Learning Models

Gianluca Turcatel[1], Yi Xiao[1], Scott Caveney[2], Gilles Gnacadja[1], Julie Kim[1] and Nestor A. Molfino[2]

[1] Digital Health and Innovation, Amgen Inc., Thousand Oaks, CA, USA; [2] Global Development, Amgen Inc., Thousand Oaks, CA, USA

Corresponding author: Nestor A. Molfino, MD, MSc, Global Development, Amgen Inc., One Amgen Center Dr, Thousand Oaks, CA 91320. Email: nmolfino@amgen.com

Supplemental Figure 1. Flowchart on patient selection, event and anchor date identification, and baseline periods generation and filtering



EHR, electronic health record; ER, emergency room; ICD, International Classification of Disease

Supplemental Methods

ML Model Details

XGBoost is a sequence of models that correct mistakes made by the previous step in such a sequence. The first model is trained with data and the next model will improve it, with the third model improving the outputs from the second and successively thereon. Hence, overall predictions are improved.[1]

LSTM is a special recurrent neural network (RNN) that allows information to persist, thereby addressing the vanishing gradient problem.[2]

Transformer is an architecture that uses an "attention mechanism" to create global dependencies between the input data and outputs without relying on RNNs of convolutional neural networks (CNNs).[3]

LSTM and Transformers algorithms can digest time series data to make predictions. Meanwhile, XGBoost expects structured data as input and is not able to leverage the temporal nature of our clinical data, at least without complicated data manipulation. Therefore, data preprocessing was tailored to the specific algorithm's requirements and limitations.

Model Training

For training of ML models, data were split into 5:1:1 training, validation, and test sets based on the patients' unique identification number. All three models were validated on a holdout set. For LSTM and Transformer models, the training was performed with early stopping and learning rate decay, and hyperparameters tuning was accomplished with Kera-tuning. For the XGBoost model, hyperparameters tuning was carried out with GridSearchCV. Given the dataset's imbalance, we assigned greater weight to misclassifications made by the model on the minority class during training. This weight was calculated as follows:

$$W = \frac{n\ samples\ in\ the\ majority\ class}{n\ samples\ in\ the\ negative\ class}$$

Model performance was reported on the hold out test set.

Model Evaluation

Models were evaluated on the holdout test set and generated a probability of exacerbation. To convert probability to binary class values, the threshold that generated the highest F1-score was chosen. Precision, recall, accuracy, and NPV were calculated using the threshold.

Model Prediction Explanation with SHAP

The predictions of the XGBoost model were explained using SHAP.[4] SHAP values are a game theoretic approach to explain the output of any ML model. Shapley values, which are used to explain how much each feature in a model contributes to the model's output are calculated by comparing the output of the model with and without each feature. The difference in the output is then attributed to the feature, and the SHAP value for that feature is the average of these differences over all possible combinations of features. SHAP values can be used to explain the output of any ML model, including black box models such as neural networks. SHAP values are also local and can be used to explain the output of the model for a single data point. Features with positive SHAP values are associated with a higher probability of exacerbation. Meanwhile features with negative SHAP values have been associated with lower probability of exacerbation. The absolute mean SHAP values of a feature is indicative of how important that feature is in driving model predictions.

Tools and Platforms

Data preparation, XGboost training, and SHAP analysis were carried out on Databrick. Transformers and LSTM-based models were trained and evaluated on AWS SageMaker.

Supplemental References

1. Guestrin TCaC. XGBoost: A Scalable Tree Boosting System. 2016.
2. Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Computation.* 1997;9(8):1735-1780.
3. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. Proceedings of the 31st International Conference on Neural Information Processing Systems; 2017; Long Beach, California, USA.
4. Lundberg SM, Erion G, Chen H, et al. From Local Explanations to Global Understanding with Explainable AI for Trees. *Nature machine intelligence.* 2020;2(1):56-67.