Supplemental Digital Content 2 – Machine Learning Model Packages and Tuning Parameters

Machine Learning Model	Implementation	Package(s)	Tuning Parameters
random forest	R	randomForest	nTree (100-10,000) nodesize (1-100) maxnodes (NULL)
extreme gradient boosting	Python, TensorFlow	xgboost scikit-learn	Estimators (10-500) max_depth (5-30) min_child_weight (0-10) instance weights (0-100) gamma (0-10) learning rate (0-1)
support vector machine	Python, TensorFlow	svm scikit-learn	linear kernel regularization constant, C: [0.1, 0.5, 1, 2, 5, 10, 50, 100, 1000]
long short-term memory	Python, TensorFlow	keras.LSTM keras.dense	dropout (0-1) recurrent_dropout (0-1) hidden layers (2-100) hidden_nodes (8-175) activation layers (‘relu’, 'sigmoid')
label-embedding attentive model	Python, TensorFlow	scikit-learn	hidden layers (100, 150, 200, 300) maximum sentence length (10, 20, 30, 50) learning rate (0.0001-0.003)

Feature engineering:

Feature engineering is a method to improve features used in machine learning models. This method relies upon domain expertise (subject matter expertise) and is a time consuming process. Support vector machine, like random forest and extreme gradient boosting, often requires extraction of useful representations of the data prior to model development, feature engineering, which can be time-consuming and requires domain expertise.

Model descriptions:

random forest – an ensemble method often used in classification. As the name implies, the model is a combination of several decision trees from which the result of the model is the majority votes of the individual trees. Each individual tree is built on a random subset of features, all of which can be identified after model development, allowing for maximal transparency as to model classifications. A disadvantage of random forest is that as the number of trees increase the model becomes complex and application may be slow. We used random forest as our first model because of its transparency.

extreme gradient boosting – another ensemble model of decision trees, extreme gradient boosting builds individual trees sequentially, attempting to improve errors with each new build. This method provides strength in classification of unbalanced datasets, where classifications have variable representation, similar to those used in this study. A weakness of this model is that it may take a long time to train and it may overfit the data (meaning the model is too specific to the dataset and less generalizable to new data). In our Current Procedural Terminology code classification, we used extreme gradient boosting to improve the low incidence Current Procedural Terminology codes such as those in the "Burn" body area classification. Using extreme gradient boosting we found an improvement in lower represented classifications but at the detriment of higher represented classifications.

support vector machine – is a unique model which maps examples as points in a high-dimensional space and attempts to find separation between points by a created hyperplane. Support vector machine is the most popular of the "kernel" methods: in finding the maximal margin of the decision boundaries you do not need to know the exact points in space rather only distance between pairs of points in space. This can be calculated using a kernel function which saves computing entire special representations. Once the hyperplane is determined it can be used to classify data much like a function in a 2-dimensional plane. Support vector machine can be sensitive to overfitting. In this Current Procedural Terminology code study, we found support vector machine to be easy to implement but it was labor intensive to tune and memory-intensive to run.

long short-term memory – a recurrent neural network model, long short-term memory is a deep learning model. In these models, data flows sequentially through layers of transformations each modifying input to increasingly complex outputs, all to derive the best representation to differentiate the assigned classification task. A major advantage of deep learning models is that these models automate the feature engineering – a key disadvantage of shallow models. By incrementally adding layers the neural networks use sequential simple transformations to arrive at complex representations to solve difficult problems. A specific advantage for long short-term memory is because it is recurrent – providing dependency on prior features. More specifically, this model looks at the sequence of events, such as in a time series. A major disadvantage of this model is that by forming complex transformations there is less interpretability of the model solutions, limiting the transparency of how the model came to its classifications. Using long short-term memory in the Current Procedural Terminology code classification we were able to use the order of the words in the procedure text to maintain any potential meaning.

label-embedding attentive model – is another deep learning neural network. The label-embedding attentive model was developed to improve text classification, a major advantage in our study where procedure text has significant importance in classification. Word embeddings, a method of representing text in a vector and maintaining relative meaning, are used in many machine learning models. This embedding is almost exclusively used for words and phrases going into a model (features). The label-embedding attentive model uses embeddings of the labels (outputs) during model training, in addition to the features (inputs) of the data, an approach for improving classification. In this study we use Current Procedural Terminology code descriptions as the new embeddings. The label-embedding attentive model then has the advantage of seeing input procedure text and the description text of the Current Procedural Terminology codes it is trying to match. We used the label-embedding attentive model in an attempt to improve accuracy by embedding label descriptions.