**Appendix 1: List of predictor variables, data type, and values**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predictor Variable Name | Data Type | Value |
| Demographics and Social Determinants | Age | Numeric | {0, 1, 2, …} |
| Gender | Categorical | {Female, Male, Other} |
| Race/Ethnicity | Categorical | {White, Black, Hispanic, Asian, …} |
| Marital status | Categorical | {Single, Married, Divorced, …} |
| Preferred language | Categorical | {English, Spanish, …} |
| Primary payor | Categorical | {Medicare, Medicaid, Self-pay, …} |
| Has PCP or not | Binary | {0 = No; 1 = Yes} |
| Visit History\* | Number of completed clinic visits | Numeric | {0, 1, 2, …} |
| Number of ED visits | Numeric | {0, 1, 2, …} |
| Number of hospitalizations | Numeric | {0, 1, 2, …} |
| Number of 72-hour ED returns | Numeric | {0, 1, 2, …} |
| Number of 30-day hospital readmissions | Numeric | {0, 1, 2, …} |
| Number of clinic appointments arranged | Numeric | {0, 1, 2, …} |
| Number of missed clinic appointments | Numeric | {0, 1, 2, …} |
| Disposition of last ED visit | Categorical | {Home, Hospital, Operating Room, Observation Unit, …} |
| Medical Conditions | Number of active medications | Numeric | {0, 1, 2, …} |
| Has chronic disease(s) | Categorical | {No, Single, Multiple} |
| Use of substance | Binary | {0 = No; 1 = Yes} |

\* The past 6 months from the index ED visits

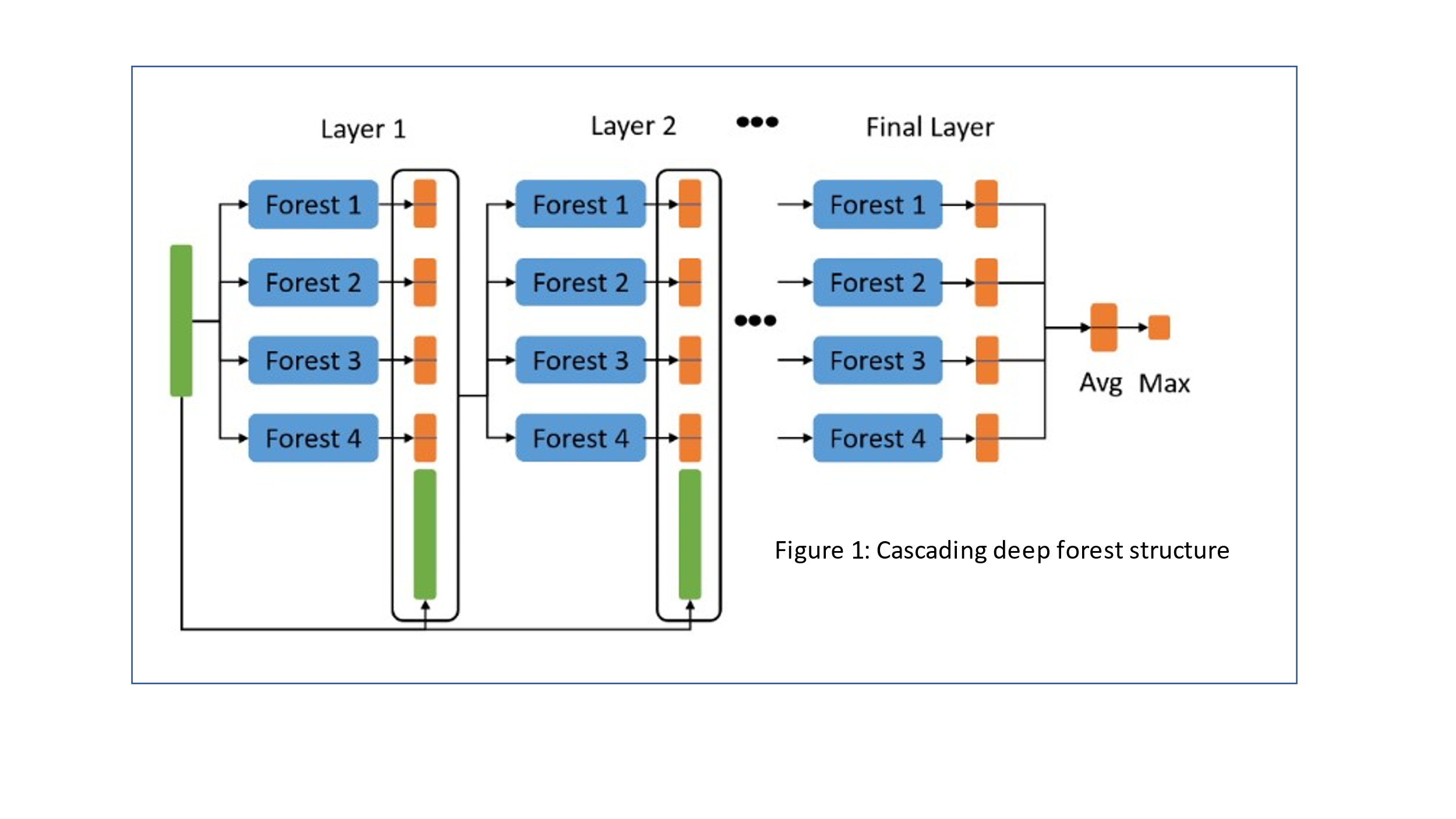
**Appendix 2: Classifiers**

Six classifiers (i.e., prediction models) were used to predict the outcome variable: (1) logistic regression (LR), (2) random forest (RF), (3) deep forest (DF), (4) decision tree (DT), (5) multilayer perception (MLP), and (6) support vector machine (SVM).

**Logistic Regression.** Logistic regression is widely used in biostatistical data classification such as justifying whether the disease condition is present or not. It estimates the conditional probability for class K given an input vector . The output of each class remains in and the summation of all classes’ probability is 1. The model to calculate the probability of the second class can be written in the form of:

The and are coefficients of the logistic regression model for the positive class and the probability of the negative class is . And the model uses maximum likelihood to compute the and .

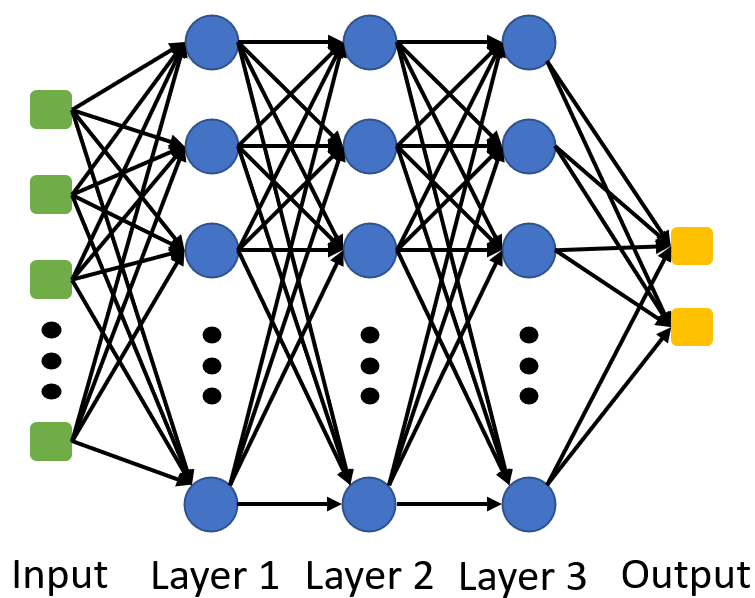
**Random Forest.** We use a random forest classifier to achieve better classification accuracy and more stable output. Plus, the random forest prevents the overfitting of the dataset since it includes multiple trees as an ensemble. The forest can be viewed as a combination of tree classifiers . Each tree in the forest is built by randomly selected features using a random vector . Finally, the trees in the forest vote for the final predictions at an input vector . To make the model more accurate, it needs a method called *bagging* to increase the diversity of prediction errors by using different subsamples of the dataset. Each tree classifier in the random forest is constructed by randomly picked samples (with replacement) from the dataset . On average 63.2% of samples should be included in each tree of the random forest.

**Deep Forest.** It is a novel decision tree ensemble classifier that uses a cascading deep forest structure. Compared to the deep learning model, the deep forest has fewer hyper-parameters for tunning. As demonstrated in figure 1, the input vector (green) is proceeded in the 1st layer by 4 random forest classifiers, and the output of each forest (orange) is a 2-dimensional vector indicating the probability of 2 classes. The original input vector, augmented with the probability vectors generated from the 1st layer, is proceeded to the next layer as the new input vector. The output of each forest will be stacked with the original input vector and be proceeded into the 2nd and 3rd layers respectively. The number of layers will stop increasing until the model converges to a given tolerance or a maximum layer number. The final prediction is given by averaging the probability of the last layer from each forest and taking the class with the largest probability. In general, the total model can be described as a cascade of cascade forests where each layer includes several cascade forests.

**Decision Tree.** A decision tree is a supervised classification method with 3 essential elements: leaf, node, and branch. For a data set with a feature and label , each node test one feature . The branch represents the split of the node by given criteria. The leaf predicts one output of the target or . The node is split by branches with each value of the feature if is a nominal or categorical feature and the node is split by threshold if is a numerical feature. To get the best split, the decision tree algorithm greedily searches for the split choices by computing the information gain from each feature. Given the dataset with Y classes we choose the split that has the largest information gain. Using the entropy and the conditional entropy , the form of information gain can be written as:

**Multilayer Perception.** The multilayer perception (MLP) model is applied in multiple fields for its performance of prediction. The general model can be written as where is an activated linear function. As shown in figure 2, the feature vector (green) is introduced to the hidden layers (blue) and each layer can be written in the form of where is a linear transformation weight vector and bias , and is the sigmoid function in the form: . To calculate the prediction error (loss) of the model with samples, we use the binary cross entropy function with the form:

And we can update the weights of each layer using an algorithm called stochastic gradient descent (SGD). In general, SGD iteratively seeks to minimize an objective function starting from a random initial point given a learning rate where is the loss function in equation 2.



**Fig. 2 Multilayer perception (MLP) structure**

**Support Vector Machine.** The support vector machine (SVM) is an algorithm that can be used for both classification and regression. This algorithm is good at handling high-dimensional datasets efficiently and memory efficient since it only uses part of the training sample to make decisions. In addition, SVM is a customized algorithm since we can use different functions according to the dataset. The general idea of SVM is to compute the margin between 2 classes given the training dataset. Given the training dataset with 2 classes: positive class and negative class, the SVM learns the hyperplane in the form of . We denote as the closest data points to the hyperplane in the positive class and denote as the point to the hyperplane in the negative class. We choose the normalization form and . Then we have the margin:

The class of a new data point can be predicted by computing the output of .