ALGORITHM 1: PSEUDO-CODE OF THE MACHINE LEARNING WORKFLOW FOR PREDICTING FOLLOW-UP PERIOD CATEGORY

```
    Input: Diabetes dataset D
    Output: Trained classification models and performance metrics
1   Preprocess D:
2       Handle missing values
3       Apply one-hot encoding
4       Normalize features using RobustScaler
5   Define Feature Selection with SMOTE:
6       FS_set = {Filter method (Mutual Information (MI)),
                  Wrapper method (Sequential Feature Selection (SFS), Genetic Algorithm (GA)),
                  Embedded method (Random Forest feature importance (RFFI), Elastic Net (EN))
                  }
7   for each feature selection method FS_method in FS_set do
8       Apply FS_method on D → obtain selected features FS
9       Initialize K-fold cross-validation (K = 5)
10      for each fold k = 1 to K do
11          Split D into D_train and D_test
12          Apply SMOTE to D_train
13          Extract X_train, y_train using FS
14          Extract X_test, y_test using FS
15          for each model M in {SVM, KNN, RF, ETC, AB, ANN} do
16              Train M on X_train, y_train
17              Predict y_pred = M(X_test)
18              Evaluate performance (Accuracy, F1, AUC, etc.)
19              Store results with tag (FS_method, M)
            end for
        end for
    end for
20  Summarize and compare performance across FS_method × model combinations
21  end
```

Figure S1 Pseudo-code of the machine learning workflow for predicting follow-up period category.

https://dx.doi.org/10.21037/jphe-24-119

ALGORITHM 2: PSEUDO-CODE OF THE FEATURE SELECTION FRAMEWORK

```
Input: Dataset D with feature set F
Output: Reduced feature subsets from each selection method
Define Feature Selection:
    FS_set = {Filter: {Mutual Information (MI)},
              Wrapper: {Sequential Feature Selection (SFS), Genetic Algorithm (GA)},
              Embedded: {Elastic Net (EN), Random Forest Feature Importance (RFFI)}
              }
Initialize result dictionary F_selected = Ø
for each feature selection category C in FS_set do
    for each method M in FS_set[C] do
        if M = MI then
            Compute MI score for each feature in F
            Select top-k features → F_selected[M]
        else if M = SFS or GA then
            Initialize Decision Tree Classifier
            Apply wrapper search (forward selection or genetic strategy)
            Select optimal subset → F_selected[M]
        else if M = EN then
            Train Elastic Net model on D
            Rank features by coefficient magnitude
            Select top-k features → F_selected[M]
        else if M = RFFI then
            Train Random Forest Classifier
            Extract feature importance scores
            Select top-k features → F_selected[M]
        end if
    end for
end for
Output all reduced feature subsets: {F_selected[MI], F_selected[SFS], F_selected[GA], F_selected[EN],
F_selected[RFFI]}
end
```

Figure S2 Pseudocode of the feature selection framework

https://dx.doi.org/10.21037/jphe-24-119

Table S1 Parameter settings of SFS

Parameter	Description	Value
k_features	Number of features to select	10
forward	Direction of feature selection	True
scoring	Evaluation metric	Accuracy
cv	Cross-validation folds	5

Table S2 Parameter settings of GA

Parameter	Description	Value
k_features	Number of features to select	10
population size	Number of individuals in population	50
ngen	Number of generations	40
cxpb	Crossover probability	0.5
mutpb	Mutation probability	0.2
tournsize	Tournament size for selection	3

Table S3 Parameter settings of EN

Parameter	Description	Value
alpha	Regularization strength	[0.01, 0.1, 1, 10, 100]
l1_ratio	L1 to L2 penalty ratio	[0.1, 0.5, 0.7, 0.9, 1.0]
max_iter	Maximum iterations	10000

Table S4 Parameter settings of RF feature importance

Parameter	Description	Value
criterion	The function to measure the quality of a split.	'gini'
max_depth	The maximum depth of the tree.	None
min_samples_split	The minimum number of samples required to split an internal node.	2
min_samples_leaf	The minimum number of samples required to be at a leaf node.	1

https://dx.doi.org/10.21037/jphe-24-119

Table S5 Hyperparameter settings for classification algorithm

Classification Algorithm	Hyperparameter setting
SVM	C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0
RF	n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, max_features='sqrt'
KNN	n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski'
ETC	n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, max_features='sqrt'
AB	n_estimators=50, learning_rate=1.0, algorithm='deprecated'
ANN	hidden_layer_sizes=(100,), activation='relu', solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, max_iter=200,

https://dx.doi.org/10.21037/jphe-24-119