Standard Operation Procedure

Standard Operating Procedure (SOP): Early Disease Detection Models

Process Name:	Author (Date):	Approver (Date):
Guidelines for EDD Model Development	Andrei Petrus (09/19/2024)	Eric Navarro (XX/XX/2024)
Responsible Audience:	**Revision Number:**	**Revision Date:**
Data Scientist	V0.4	09/11/2025

This SOP's Purpose/Goal
This SOP has been created to serve as a data science guideline for the purpose of developing and evaluating Early Disease Detection Models on retrospective data.

Standard Operation
I. Cohort, Target Indication, and Labelling

1. **Establish preliminary exclusion criteria.**
   a. Exclude patients groups according to the Solution Design Document
      e.g. Males for Breast Cancer screening, Non-Smokers for Lung Cancer screening
   b. Exclude patients with pre-existing conditions that are associated with the target condition or whose clinical expression is associated with the model's input features. Consult the Solution Design Document for further details.
      e.g. exclude patients previously diagnosed with a neoplasm at a different location for a Lung Cancer Model, exclude patients with insulin resistance due to Polycystic Ovary Syndrome for a Diabetes Mellitus Model
2. **Establish preliminary inclusion criteria.**
   a. Include patient groups according to the to the Solution Design Document.

    i. *e.g. Include only patients aged 45 or older, with at least one Complete Blood Count and one Comprehensive Metabolic Panel*

3. **Establish a precise definition for the target indication(s).**
    a. n most cases, the target condition can be defined as ICD-10-CM code(s)
        i. *e.g. C18: Colon Cancer*
    b. Where possible, additional ascertainment should be sought to strengthen the coding confidence – consult with Disease Insights Team for guidance.
        i. *e.g. Z12.11: Encounter for colon cancer screening, Z51.1: Encounter for antineoplastic chemotherapy and immunotherapy*
4. **Create and assign patients the target label.**
    a. In some cases, secondary targets will complement the primary target - consult the Solution Design Document for additional details. Assign labels to positive cases that discriminate between primary and secondary conditions.
        i. *e.g. Primary Target: Malignant Neoplasm of the Colon (C18) - Label 1*
        Secondary Target: Benign Neoplasm of the Colon (D12.2-6) - Label 2
    b. Where different levels of ascertainment confidence exist, mark those in an additional confidence label.
        i. *e.g. <2 encounters with C18 and no evidence of cancer screening/treatment – Label 1, Confidence Low*

            >2 encounters with C18 and encounters for Z51.1 (therapy) – Label 1, Confidence High

II. Feature Selection, Feature Generation, and EDA

1. **Determine preliminary features from medical guidelines, published literature – consult the Solution Design Document for additional details and perform research on similar ML models**
    a. Patient Demographics*
        i. *e.g. age, sex, etc.*
    b. Comorbidities
        i. *e.g. >1 encounter for T2D, >1 encounter for COPD*
    c. Vitals (sequence of time-dependent events)
        i. *e.g. Blood Pressure, SpO2, etc.*
    d. Laboratory Values (sequence of time-dependent events)
        i. *e.g. WBC count, Hematocrit, etc.*
    e. Procedures

    i. *e.g. encounter for Z98.0 (Intestinal Bypass)*
  f. Medications
    i. *e.g. SGLT2 inhibitors, Vitamin K antagonists, etc.*
  g. Polynomial Features
    i. *e.g. CHAD2DS2-VASc Score for Stroke Risk*
  h. *If Race, Ethnicity, Sex and Social Determinants of Health (like employment status) are as features, additional diligence must be performed to ensure the model does not become biased (e.g. only predicts male)*
    i. TBD - what that due dilligence is
2. **Parse and aggregate Time-Dependent Features**
  a. Determine time of index diagnosis for diseased cohort. Further analysis should be cognizant of features that exist before and after index diagnosis.
    i. *e.g. Cases of Heart Failure identified after index diagnosis of Arrhythmia should not be used as input features for Early Disease Detection*
    Experiments should investigate prognostic ability after excluding lab values generated after index diagnosis, or in the few months before index diagnosis
  b. Aggregate Labs and Vitals in monthly/quarterly/yearly bins and generate summary statistics to be used as features
    i. *e.g. mean, median, min, max, stdev, count*
3. **Identify Additional Features from Data (Optional – HIGH EFFORT)**
  a. Create a dataframe containing all the features that would be investigated and the output label
    i. *e.g. all ICD10-CM/PCS Codes in Dataset, all LOINC codes in dataset, and disease status (output label)*
  b. Perform the following statistical methods to search for relevant features:
    i. Categorical Input Feature – Categorical Output Label: *Chi-Squared or Mutual Information*
    ii. Numerical Input Feature – Categorical Output Label: *ANOVA or Kendall Concordance*
    iii. Categorical Input Feature – Numerical Output Label (uncommon): *ANOVA or Kendall Concordance*
    iv. Numerical Input Feature – Numerical Output Label (uncommon): *Pearson's Correlation or Spearman's Rank Coefficient*
  c. Select relevant features with the highest scores
    i. *see scikit SelectKBest or SelectPercentile*
4. **Perform Exploratory Data Analysis as appropriate.**
  a. Identify Percent Missingness by variable (columns) and cases (rows).

b. Remove Features with extremely high missingness (<5% within diseased cohort) and flag features with high missingness (<30% within diseased cohort). Imputation not advisable at this point.
c. Ensure unit consistency and encode categorical variables
d. Detect and Remove outliers
    i. *e.g. using acceptable lab ranges or 5 stdev away from mean*
e. Calculate Dataset Imbalance
f. Perform Multicollinearity Test and remove highly correlated features

5. **Feature Distribution Comparison Between Data Sets**
    a. Identify gaps between different data sets (Logicstream vs DO).
        i. In patient vs out patient
    b. Prune features that aren't generalize-able
6. **Document Features.**
    a. Create a Feature Dictionary
    b. Log all features identified in a spreadsheet, including the reasoning and provenance (EDA, clinical SME, published literature and prior art)
    c. Feature naming convention: Features should not include any special characters aside from _'s used to separate words. Feature names should be uppercase. If a feature was derived from a formulate, e.g. an average that should be included in the name. Be sure to use the human readable names for LOINCS and Labs.
        i. Example feature names 😀
            1. MEDIAN_HEMATOCRIT
            2. DELTA_HEART_RATE
            3. RACE_WHITE
            4. PULMONARY_EMBOLISUM
            5. BMI

III. Prepare Data for Training

1. **Prepare data to train 3 or 4 models**
    a. Logistic Regression (LR)
    b. Random Forest with Gradient Boosting (XGBoost)
    c. TRisk2 (Oxford transformer model)
    d. Optional *(if other architectures under perform)* : vanilla RNN (tanh activation), LSTM or GRU
2. **Create a first hold-out validation set containing one or two smaller sites.**

3. **Randomize data masking of 1 week, 1 month, 3 months, 6 months from diagnosis (DX) date so the model doesn't focus entirely on right before DX date.**
4. **Create a second hold-out set containing 5-10% of the remaining dataset (all sites - 10% of diseased cases and 10% of controls)**
5. **For LR and MLP, imputation is likely necessary.**
    a. *Simple and Iterative Imputer will suffice for most cases.*
    b. *Impute only after splitting the data and add a binary indicator to merk imputed data.*
6. **For LR, MLP and RNNs (RNN-vanilla, LSTM or GRU), scaling of numerical features is likely necessary.**
    a. Standard Scaler will suffice for most cases
    b. Calculate Scaler only on training data and apply it to testing and validation data
7. **RNNs require data to be preprocessed in feature vectors corresponding to each event. Each vector will have missing values due to data sparsity, however LSTM and GRU can handle sequences of varying length. Nonetheless, if imputation is desired, the following strategies are recommended:**
    a. Forward-filling (using the last known value)
    b. Using a special "missing" token or using the masking approach
    c. Domain-specific imputation (e.g., using normal ranges for lab values)
8. **Initiate a cross-validation to split data into train and test using 10 folds.**
    a. *Stratified K-fold is recommended. Shuffling is likely not necessary. Set random state to 42 in all experiments for reproducibility.*

IV. Train Models

1. **Initialize GridSearchCV where applicable to simplify hyperparameter tuning.**
2. **Set up and tune hyperparameters, Activation/Loss Functions, and Performance metrics:**
    a. **Logistic Regression**
        i. Performance Metrics: AUROC, AUPRC, F1, F2, F0.5, Brier Score, G-Mean, Accuracy
        ii. Hyperparameters: *C (regularization strength) Penalty: l1, l2, elasticnet Solver: liblinear, saga, etc.*
        iii. Activation Function: Sigmoid (implicit)
        iv. Loss Function: Cross-Entropy (implicit)
    b. **XGBoost**
        i. Performance Metrics: AUROC, AUPRC, F1, F2, F0.5, Brier Score, G-Mean, Accuracy
        ii. Hyperparameters: *n estimators, learning rate, max depth, min child weight, subsample, colsample by tree, Lambda (l2), Alpha (l1)*
        iii. Loss Function: logistic (binary) or softprob (multi)

c. **MLP Neural Network**
   i. Performance Metrics: AUROC, AUPRC, F1, F2, F0.5, Brier Score, G-Mean, Accuracy
   ii. Hyperparameters: *num layers, num neurons, learning rate, batch size, epochs, dropout rate*
   iii. Activation Function: ReLU (vanilla or leaky) for hidden layers; Sigmoid for output layer
   iv. Optimizer: Adam, RMSProp, SGD
   v. Loss Function: Coss-Entropy

d. **RNNs**
   i. Performance Metrics: AUROC, AUPRC, F1, F2, F0.5, Brier Score, G-Mean, Accuracy
   ii. Hyperparameters: *num layers, num units, batch size, epochs, dropout rate*
   iii. Activation Functions: tanh for hidden states; Sigmoid for output layer
   iv. Optimizer: Adam, RMSProp, Adagrad
   v. Loss Function: Cross-Entropy

e. TRisk2
   i. TODO

3. **Train Models in sequence with cross-validation**
   a. Plot the Loss Function over epochs when applicable.
   b. Plot the Performance Metrics and compare between training and testing.
   c. Capture and Plot Feature Importance.
   d. Save models with a unique identifier of format: Disease_Architecture_Date_BuildNo – e.g. MASH_LR_01012024_03 (ONNX or json file desired)
   e. Save input data header with a unique identifier format:
      i. ex: Disease_DataModel_Architecture_Date_BuildNo – e.g. MASH_OMOP_LR_01012024_03
   f. Log all trained models in a spreadsheet, containing the unique identifier for model & training data header, target indication, cohort details (inclusion/exclusion criteria) and a brief description of the training protocol
4. **If performance or loss are raising alerts, reinitiate the EDA process, tune features and retune hyperparameters.**

V. Validate Models

1. **Validate the models on the holdout sets.**
   a. Plot the Performance Metrics and compare between training and testing.
2. **Use Bootstrapping to estimate standard deviation and 95% confidence interval of model performance.**

3. **If performance metrics are raising alerts, reinitiate the EDA process, tune features and retune hyperparameters.**
4. **Investigate any data dissimilarity between train set and holdout sets and measure Drift.**
    a. Feature Drift Measures:
        i. *KS Test, Chi-Squared Test, Frobenius Norm, PSI, KL Divergence, JS Divergence*
    b. Concept Drift Measures:
        i. *Page-Hinkley Test, ADWIN*
5. **Measure Bias and Fairness across different demographic segments.**
    a. Fairness Metrics: *Demographic Parity, Equal Opportunity, Equalized Odds*
6. **Measure Calibration and Parity.**
    a. Calibration: Ensure that predicted probabilities match the true likelihood of outcomes. Use reliability diagrams and calibration plots.
    b. Parity: Check for parity in prediction distributions across different groups. Evaluate the dispersion of predicted outcomes.
7. **Perform SHAP Feature importance analysis and validate that feature importance propagates across training and holdout sets.**

VI. Improve Model Robustness

1. **Perform Time Travel Experiments.**
    a. Models should prognosticate the probability of disease diagnosis days/weeks/months in advance of typical index diagnosis.
    b. Train and validate the models by masking all of the data generated by the diseased patients in the time window before index diagnosis. Consult the Solution Design Document for further information of detection-windows.
    c. Match the time controls by trimming records in order to have a similar distribution of timestamps of 'model run' with the diseases cases timestamps.
2. **Perform Data Augmentation Techniques.**
    a. Masking**:** Randomly replace some data points with masks (on a feature-level or event-level) – not applicable to demographics
    b. Noise: Gaussian Noise to some numerical features; Flip some categorical
3. **Reduce Complexity.**
    a. Dropping Features: quasi-randomly remove features
    b. Investigate reducing the feature list while maintaining model performance. E*.g., SelectFromModel, results from SHAP, Hierarchical feature clustering, etc.*
    c. Include a tab within the feature spreadsheet created in II, 5 that includes the final feature list to be used by the MLOps engineers. This feature list could be the same as the list of all

identified features, but in most cases will be the list resulting from the Feature Pruning step of the process.

d. Dimensionality Reduction: PCA

4. **Time Wrapping (only for RNNs).**
   a. Randomly Stretch or Shrink the time axis in parts of the data
5. **Generate Synthetic Data – Optional.**
   a. Use SMOTE or GAN
6. **Add additional Inclusion/Exclusion Criteria to improve model performance when compared to baseline (e.g. age filtering, comorbidity-filtering, etc.)**
7. **Measure the effect of each data augmentation technique on the hold-out set.**
8. **Finalize model architecture, feature selection, hyperparameter tuning and inclusion/exclusion criteria. If necessary, create an ensemble wrapper with a voting mechanism. Include a json version of the final model in GitHub for use by the MLOps engineers.**

VII. Produce a Model Specification Report

1. **Record the final model version, the final set of features, and the final inclusion criteria in report, alongside a description of the training and validation protocols.**

2. **Plot Feature Importance and determine which features are required and optional (based on previous experiments on model sensitivity to feature dropping).**

3. **Calculate and plot the final performance metrics: AUROC, AUPRC, F1, F0.5, F2, Accuracy, Breier Score, G-Mean, for both all and require-only features.**

4. **Calculate Sensitivity, Specificity, PPV, NPV, Odds-ratio at different predefined Specificity values (e.g. 93%, 95%, 90%) for both all and required-only features – consult the Solution Design Document for further details.**

5. **Calculate the Risk Threshold Cut-off to achieve the desired Specificity Values for both all and required-only features.**

VIII. Bias Mitigation with Equity and Fairness Metrics

Fairness metrics must be calculated for each group in the dataset.
A “group” is defined as: Sex, Gender (if used), Race, Ethnicity, Socio-economic background, or any other subset of the population that could be misrepresented due to bias.
These metrics must be reported in the Chai model cards for each model we develop as well as any final model documentation where results are shared.

1. **Parity Difference (Demographic Parity Difference)**
   Difference in selection rate (fraction of positive predictions) for a given group compared to the overall population.
   Application: Compute the predicted positive rate per group and subtract the overall rate. Values close to 0 indicate parity; larger values suggest bias.
2. **Equalized Odds Difference**
   Difference in true positive rates (TPR) and false positive rates (FPR) between a group and the overall population.
   Application: Compute TPR and FPR per group, compare to overall. Values close to 0 indicate fairness.
3. **AUROC per group (Training and Validation)**
   Area Under the ROC curve calculated separately for each group in both training and validation sets.
   Application: Report AUROC per group with confidence intervals.
4. **Sensitivity (True Positive Rate) per group (Training and Validation)**
   Proportion of actual positives correctly identified.
   Application: Report sensitivity per group with confidence intervals.
5. **Specificity (True Negative Rate) per group (Training and Validation)**
   Proportion of actual negatives correctly identified.
   Application: Report specificity per group with confidence intervals.
6. **Reporting Guidance**
   - All fairness metrics must be presented per group and relative to the overall population.
   - Confidence intervals are required for AUROC, Sensitivity, and Specificity.
   - Large disparities must be highlighted in the model card along with proposed mitigations (e.g., rebalancing, subgroup analysis, data augmentation).
   - These metrics and results must be included in final model documentation.

Revision History:	
V0.1	Document Created. Topics Covered: I. Cohort, Target Indication, and Labelling II. Feature Selection, Feature Generation, and EDA III. Prepare Data for Training IV. Train Models V. Validate Models VI. Improve Model Robustness VII. Produce a Model Specification Report
V0.2	II. 5. c. How to capture final feature names. VI 3 b, c Feature pruning step. Documenting finalized feature list for engineering. VI 8. Include pickled file of final model into GitHub.
V0.3	**II.** 4 Feature comparison between data sets **III**. 4 Data masking strategy
V0.4	**VIII**. Bias Mitigation with Equity and Fairness Metrics