**PROJECT PLAN**

This assessment provides you with an opportunity to reflect on concepts in machine learning in the context of an open-ended research problem, and to strengthen your skills in data analysis and problem solving. The idea behind the project is for you to correctly implement general principles of statistical machine learning, while exploring data and algorithms of your interest. The goal of this project is not to obtain the best metric (e.g., accuracy) per se, but to perform different steps of machine learning in the proper way, according to what you have learnt in this subject. You should be clear about what is the research question in your project, what you plan to try, and what insights you might be planning to get. Then in terms of the results you get, you should discuss what worked or what did not work, and explain the possible reasons in light of what you learnt in class.

**By submitting this plan, you confirm that you have access to the required computational resources and tools to execute this plan. This plan (or any subsequent change) will be reviewed, to ensure that all plans have a similar level of complexity. If your plan is not reviewed, it might likely not be at the level of what we expect, which will end up impacting your grade. Thus, it is in your best interest to have your plan reviewed.**

1. Student names. (The project is to be done in groups of 3 students.)

Ximing Wan

Yang Jin

Lanye Shao

1. [Up to 5 lines] Definition of the problem and research question, possibly relevant to your interests.

This project aims to develop a machine learning algorithm to predict the number of hospital days a patient will experience in the next 12 months, based on claims, lab tests, and prescription data from the past three years. This aligns with the COMP90051 project requirements, which focus on solving an open-ended research problem through data preprocessing, algorithm selection, and model tuning. The research question is how to effectively predict hospital days using machine learning and which algorithms and features contribute most to the prediction accuracy, along with their advantages and limitations.

1. [Up to 5 lines] Description of the dataset (or datasets) to be used, e.g., number of tables, rows, columns, type of data (discrete/categorical, continuous, sentences, images, etc.). Datasets should be already publicly available, since there is not enough time for you to collect data. Possible datasets are for instance: [ADHD 200 (Whole Brain Data)](http://www.nitrc.org/plugins/mwiki/index.php/neurobureau:AthenaPipeline#Whole_Brain_Data), [Labeled Faces in the Wild](http://vis-www.cs.umass.edu/lfw), [Heritage Health Prize](https://www.kaggle.com/c/hhp), [Yahoo Bidding (A1)](http://webscope.sandbox.yahoo.com/catalog.php?datatype=a), [Yahoo Music (C15)](https://webscope.sandbox.yahoo.com/catalog.php?datatype=c). **You should choose a challenging dataset. Small datasets are not allowed. (Please also see question 5).**

The dataset is from the Heritage Health Prize competition, aimed at predicting the number of hospital days members will spend in the next 12 months. It spans 48 months and includes multiple tables, such as member details (age, gender), claims data (provider, place of service, length of stay), lab tests, and prescription information. Additionally, there are hospitalization tables that record the number of hospital days for each member in Year 2 and Year 3. We need to use this data to train their algorithms and predict hospital days for Year 4 for submission.

1. URL where the above dataset(s) is(are) available.

https://foreverdata.org/1015/

1. [up to 5 lines] Which feature construction and preprocessing of the dataset will be performed, e.g., converting several tables to a single table, counting, summing, one-hot encoding, etc. **You are not allowed to just read a single data table and used it. Remember this is a Masters level subject and we require some level of complexity. You are allowed to either implement this from scratch or use third-party code.**

1)one-hot encoding of sex in Members.csv. Convert the age group in the AgeAtFirstClaim column to the average for each segment.

2)Filter the claims data for the first and second years in claim.csv and calculate the number of claims for each member in that year.

3)Count the number of unique values for each member for a particular feature, such as Providers.

4)In the member's claim data, counts are taken for the length of stay and SupLOS characteristics, and these counts are added as new columns.

This is just a preliminary preprocessing, and we will try to add more features for the accuracy of the results.

1. [Up to 5 lines] Which 3 machine learning algorithms are going to be compared? You should list 3 different algorithms and with different model class complexity, i.e., simple, medium, complex. **You are allowed to either implement this from scratch or use third-party code. At least one of the 3 algorithms should be from a research paper in a conference or journal, e.g.,** [**NeurIPS**](https://proceedings.neurips.cc/)**,** [**ICML/UAI/AISTATS/JMLR**](https://proceedings.mlr.press/)**,** [**ICLR**](https://openreview.net/group?id=ICLR.cc)**,** [**TMLR**](https://openreview.net/group?id=TMLR)**, etc. (Since this research-paper algorithm might take you more time to figure out, you can let us know this later.)**

* **Simple:** Logistic Regression – a basic linear model that provides baseline performance.
* **Medium:** Random Forest – a tree-based ensemble method with moderate complexity and strong interpretability.
* **Complex**: A neural network model based on a research paper.

1. [Up to 5 lines] Cross-validation technique, e.g., training/validation/testing, k-fold cross-validation, bootstrapping. **You MUST implement this from scratch. At least 20 independent repetitions should be run, so that means/variances can be computed for proper comparison between algorithms.**

We will implement k-fold cross-validation from scratch, using 5 folds repeated 20 times to compute the mean and variance for robust comparison. The dataset will be split into training, validation, and testing sets in each fold. Bootstrapping techniques may also be used to account for variance in smaller subsets of the data.

1. [Up to 10 lines] Which hyperparameter(s) is(are) going to be tuned for each of the 3 algorithms above, and what method is going to be used for the nested cross-validation. **You MUST implement this from scratch. Every algorithm should have at least one hyperparameter to be tuned and such hyperparameter should be expected to affect the results significatively.**

For Logistic Regression, we will tune the regularization strength (C) to control the balance between model complexity and overfitting. For Random Forest, we will tune the number of trees (n\_estimators) and maximum tree depth (max\_depth), which directly influence both model accuracy and overfitting. For the Neural Network (RNN), hyperparameters such as the number of hidden layers, hidden units per layer, and learning rate will be tuned to optimize its temporal learning capabilities.

We will implement nested cross-validation with a 5-fold outer loop for model evaluation and a 5-fold inner loop for hyperparameter tuning. Grid search will be used for hyperparameter tuning in the inner loop, ensuring thorough exploration of the parameter space for each model. This approach helps ensure fair comparison between models while preventing data leakage.

1. [Up to 15 lines] Description of the experimental results, e.g., learning curves, ROC curves, plots of different datasets, etc. **You MUST implement this from scratch. Error bars should be computed across repetitions (at least 20 as described in question 7) and reported for proper comparison between algorithms.**

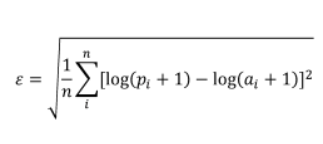
**Learning Curves**: We will plot learning curves for each model, showing the training and validation error as a function of the training dataset size. This will help identify if models are underfitting or overfitting.

**ROC Curves**: For each algorithm, we will compute and plot the Receiver Operating Characteristic (ROC) curve to evaluate classification performance. The Area Under the Curve (AUC) will be used to compare the algorithms' ability to differentiate between classes.

**Error Bars**: We will run each algorithm at least 20 times, and compute the mean and variance for key performance metrics (e.g., accuracy, precision, recall, and AUC). Error bars will be plotted to show the variability across these repetitions.

**Feature Importance**: For models like Random Forest, we will visualize feature importance to provide insights into which features contribute most to the predictions.

**Error Metric**: We will use the error metric from the Heritage Health Prize competition to calculate the accuracy of predicting hospital days. This metric computes the mean squared difference between the natural logarithms of the predicted and actual hospital days:



1. Which programming language are you going to use? (Python, Jupyter, C++, MATLAB, Java are allowed.)

We will use Jupyter & Python(optional).