ECE 5970 Final Report (Dec. 5, 2018)

Team member: Jiahao Li (jl3838), Quan Sun (qs84), Xiaowen Wang (xw453)

Dataset: The Alzheimer's Disease Prediction of Longitudinal Evolution (TADPOLE)

Introduction

The Alzheimer's Disease Prediction Of Longitudinal Evolution (TADPOLE) challenge is a competition aiming to find the best model for progress of Alzheimer's disease from the onset of early symptoms to the latest stage. We try to develop models that take all past clinical data as input to predict diagnosis results and important measurements at a future time point of interest.

The input dataset contains all the features, which includes the dates of visit, demographical data, and a large variety of clinical measurements. Training, validation and test datasets contain their corresponding labels. 4 different target values are to be predicted. One is categorical diagnosis including 3 possible states: CN (healthy), MCI (pre-dementia), and AD (Alzheimer's disease). The other 3 are quantitative variables, Mini Mental State Exam (MMSE) test score, Alzheimer's Disease Assessment Scale (ADAS13), and head size-normalized volume of the brain ventricles. We aim to achieve accuracy as high as possible in predicting test targets.

The main challenge for this project has two folds. The first is that a lot of the features and targets suffer from missing data. The reason is that there a wide variety of tests and measurements for physicians to test for patients' status of Alzheimer's disease. When patients have their clinical visit, usually only a couple of them are actually performed. The second part is very specific to this problem. The time series in the input data are quite unique. They are obviously low-frequency. Both the number of visits and interval between visits for each patient are highly variable. Therefore, ideally prediction should have full coverage on continuous time axis.

In this report, we are going to tackle this problem with a variety of popular methods and we compare their performance. For classification of the categorical diagnosis, we used support vector machine (SVM), random forests, GBoost, XGBoost, and neural network, including LSTM for sequence analysis. For regression of quantitative targets Mini Mental State Exam (MMSE) test score, Alzheimer's Disease Assessment Scale (ADAS13), and head size-normalized volume of the brain ventricles, we used linear model, decision tree as well as neural network.

Data Preprocessing and Data Analysis

The goal of data cleansing is to handle the first challenge of missing data, as well as to select important features for training. The TADPOLE dataset containing Input_Data, train_traget, validation_target and test_target files, has 1892 features and a lot of mission data.

Firstly we analyze the data to drop less useful information. We drop the columns consisting of more than 60% missing data and those unknown columns providing invalid information so that just 50 features left. Further we remove features overlapping and to have 33 columns left.

Next step is filling missing data. We fill the baseline features firstly by using same DX_bl object data. Then we fill missing data in DXCHANGE and DX_bl values. We assume that the same values of DXCHANGE and DX_bl means the object getting the same status. So we fill the others by the average of all objects values having same values of DXCHANGE and DX_bl. Then we normalize quantitative data. Ventricles is normalized by dividing the value of ICV.

Before we proceed to model, the pearson correlation and distribution observation are conducted to select better features. We can see there are some features highly correlated, so we select some features to feed our regression models and classification models.

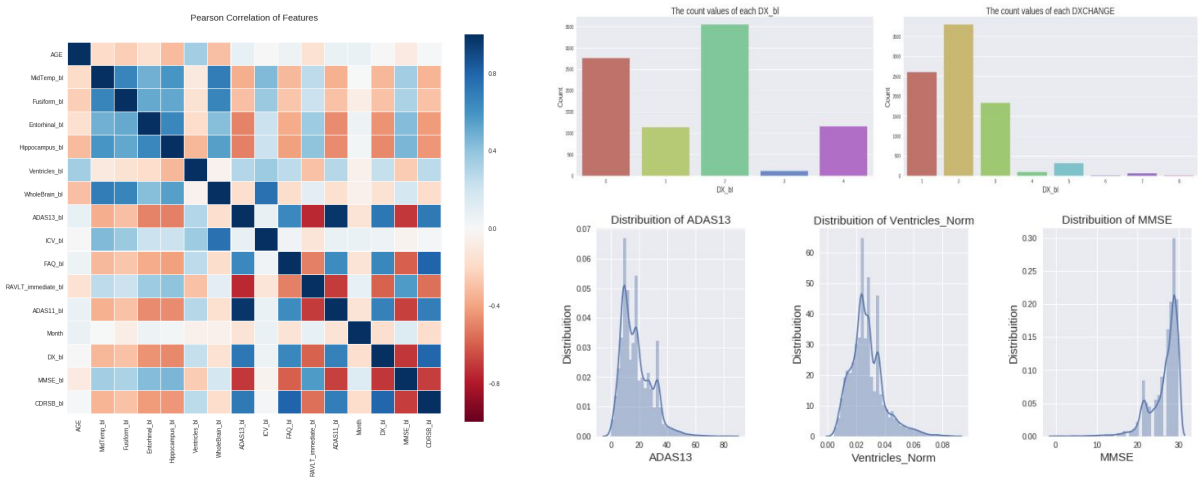

Figure 1. Left: pearson correlation of features selected; Right: Distributions for DX_bl, DXCHANGE, ADAS13, Ventricles_Norm and MMSE

After plotting and analyzing the distributions of DX_bl, DXCHANGE, MMSE, Ventricles_Norm and ADAS13, we calculate the skewness to capture the distribution unbalance. Skewness number is the degree of distortion from the symmetrical bell curve. It differentiates extreme values in one versus the other tail. Positive skewness number means when the tail on the right side of the distribution is longer or fatter, vice versa. The skewness number of ADAS13 is 1.192, of Ventricles_Norm is 1.04, of MMSE is -1.97. Those skewness numbers prove the distributions are not symmetrical.

When predicting ADAS13 and Ventricles_Norm, we choose features: AGE, Hippocampus_bl, Ventricles_bl, WholeBrain_bl, ADAS13_bl, ICV_bl, FAQ_bl, RAVLT_immediate_bl, ADAS11_bl, MMSE_bl, CDRSB_bl, Month, DX_bl, then drop all missing data. When predicting MMSE, we choose features: AGE, DX_bl, Month, Hippocampus_bl, Ventricles_bl,

WholeBrain_bl, ADAS13_bl, ICV_bl, FAQ_bl, RAVLT_immediate_bl, ADAS11_bl, MMSE_bl, CDRSB_bl, then we drop all missing data. However, we choose AGE, Month, DX_bl, MMSE, ADAS13 for the classification features. We plot the diagnosis results distribution for those features in training set as following.

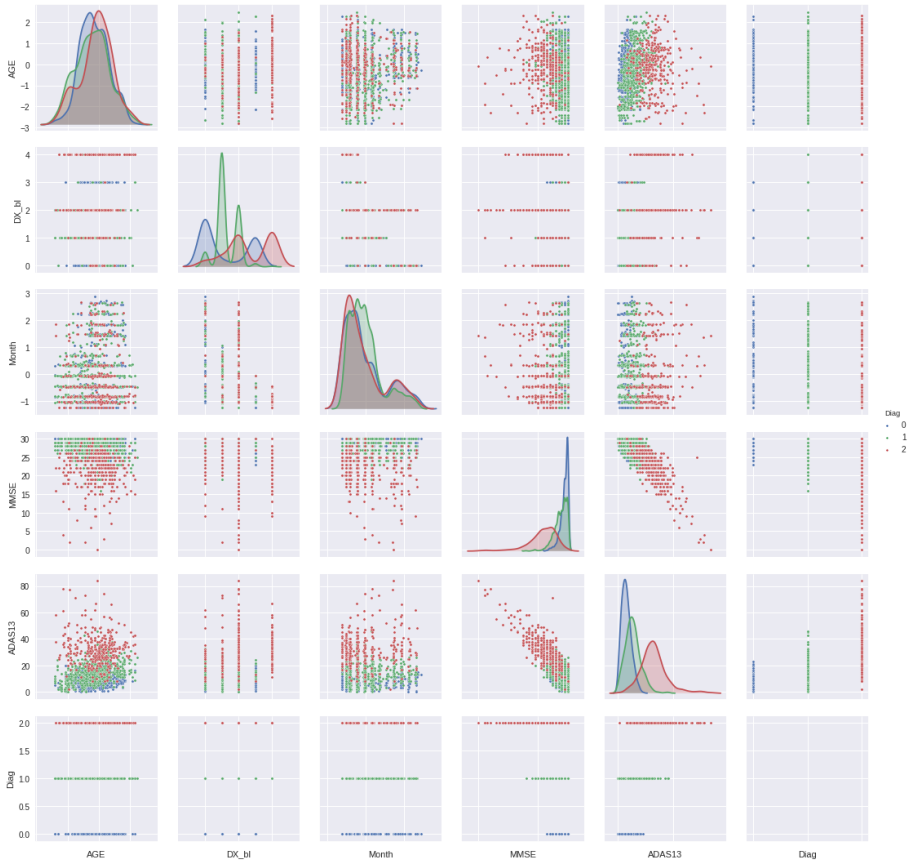

Figure 2. Distributions for diagnosis results on different features

Models

1. Baseline Model

The baseline of this problem is simply copying the labels of last known visit to all the future visits that need to predict, the so-called carry-forward baseline. This is based on the assumption that all labels remain the same. This is obviously too naive. Therefore, any improvement made in our model should outperform this one. For this simple baseline model, we measure it by using cross-entropy and MSE on prediction, classification and regression respectively. The classification cross-entropy is 20.715, MSE(ADAS13) is 166.388, MSE(MMSE) is 15.295.

2. Current Machine Learning Models

2.1 Regression Model

Our regression models for predicting ADAS13, Ventricles_Norm and MMSE are ElasticNetCV, Gradient Boost Regressor and Neural Networks containing 3 hidden layers. We respectively training our models for these three targets.

Evaluation for Models predicting ADAS13:

Model\Metrics	R2 score	RMSE	MSE
ElasticNetCV	-0.136	0.075	\
Gradient Boost	0.118	0.0712	\
Neural Networks	\	\	0.0053

Evaluation for Models predicting Ventricles_Norm:

Model\Metrics	R2 score	RMSE	MSE
ElasticNetCV	-0.642	0.008	\
Gradient Boost	0.887	0.0037	\
Neural Networks	\	\	0.00051

Evaluation for Models predicting MMSE:

Model\Metrics	R2 score	RMSE	MSE
ElasticNetCV	-0.592	0.0888	\
Gradient Boost	-0.516	0.085	\
Neural Networks	\	\	0.0088

2.2 Classification Model

2.2.1 XGBOOST

We prepare five learning models as our first level classification: Random Forest classifier, Extra Trees classifier, AdaBoost classifier, Gradient Boosting classifier and Support Vector Machine. For acquiring the output of the first level predictions we feed the training and validation data into our 5 base classifiers and use the Out-of-Fold prediction function we defined earlier to generate our first level predictions. Then using sklearn method .feature_importance_ to get feature

importances generating from different classifiers. We can find for this classification model the DX_bl and Month are two most important features.

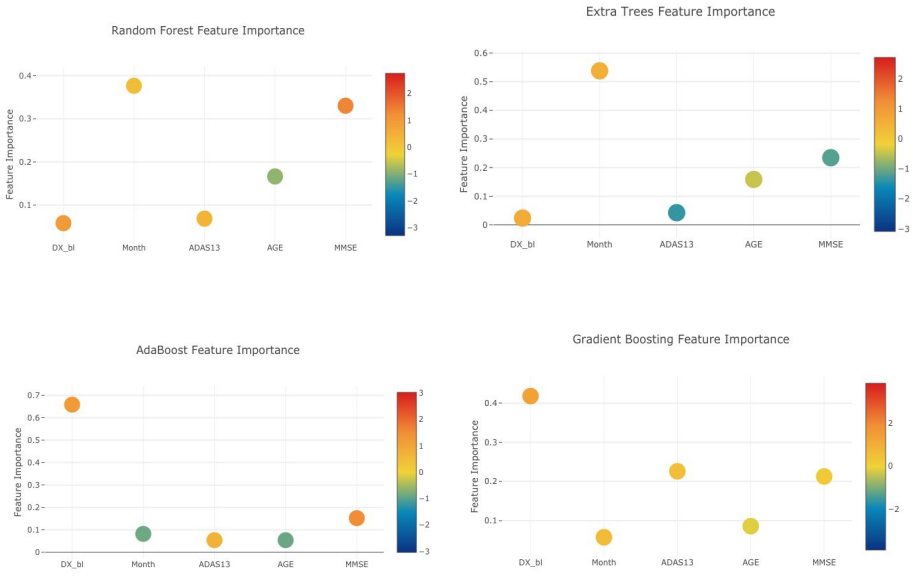

Figure 3. Feature importance for different models

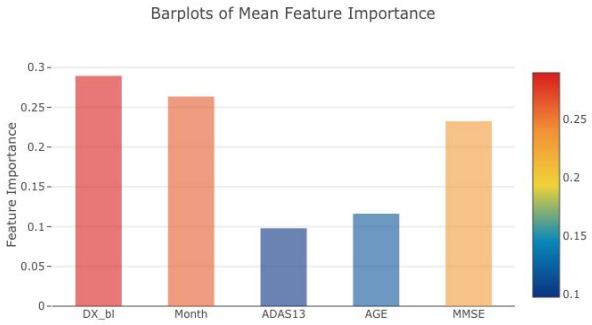

Figure 4. mean feature importance

Predictions\Metrics	Accuracy	Precision	Recall
CN_Diag	92.16%	88.94%	91.36%
MCI_Diag	85.92%	82.67%	82.3%
AD_Diag	92.69%	83.64%	80.00%

Having obtained our first-level predictions, one can think of it as essentially building a new set of features to be used as training data for the next classifier. Concatenating and joining both the first-level train and validation predictions as x_train and x_validation, we can fit a second-level learning model.

2.2.2 Other models

We construct some other models such as SVC, Random Forests, Gradient Boost, K-Neighbors Nearest for predicting the diagnosis results as following:

SVC:

Targets\Metrics	Accuracy	Precision	Recall	AUC
CN_Diag	91.27%	87.01%	91.36%	26.46%
MCI_Diag	85.20%	81.50%	81.86%	90.22%
AD_Diag	92.51%	85.44%	76.52%	25.05%

Random Forests:

Targets\Metrics	Accuracy	Precision	Recall	AUC
CN_Diag	90.91%	88.24%	88.64%	18.90%
MCI_Diag	83.42%	79.04%	80.09%	90.89%
AD_Diag	92.16%	81.98%	79.13%	35.13%

Gradient Boost Classifier:

Targets\Metrics	Accuracy	Precision	Recall	AUC
CN_Diag	89.48%	87.79%	85.00%	23.43%
MCI_Diag	80.75%	75.00%	78.32%	88.59%
AD_Diag	90.20%	76.79%	74.78%	31.89%

K-Neighbor Nearest:

Targets\Metrics	Accuracy	Precision	Recall	AUC
CN_Diag	92.16%	88.94%	91.36%	20.40%
MCI_Diag	85.92%	82.67%	82.30%	89.35%
AD_Diag	92.69%	83.64%	80.00%	35.20%

3. LSTM (Long short-term memory)

As a further exploration, we also try to apply LSTM model for diagnosis prediction. Since our dataset contains patients' sequence data of past visits as well as future prediction, it is always reasonable to predict one's condition based on his previous visit data, which leads us to the utilization of LSTM to capture the influence by previous data.

For LSTM model, we need to give a more precise observation on our data. Considering that each patient has different history visit times as well as different future prediction time points, in addition the time interval between two visit can vary a lot, it is necessary to make some assumptions to simplify the sequence data structure. Thus we choose a fixed time interval and reasonably assume that patients' condition will not change rapidly in this period. The fixed time interval here we choose is 6 months. For those data falls into the same period of 6 months, the first one is remained; for those we have missing data in a 6 month period, we impute it with the last corresponding value, same strategy as moving-forward in benchmark. We also collect statistics on the training and validation sequence length based on the fixed time interval as shown below. Thus an 8 times prediction for each step is defined for our LSTM model.

count	655.000000
mean	4.876336
std	2.121314
min	0.000000
25%	3.000000
50%	5.000000
75%	7.000000
max	9.000000

count	218.000000
mean	5.256881
std	2.096315
min	0.000000
25%	4.000000
50%	6.000000
75%	7.000000
max	9.000000

Figure 5.Statistical summary of prediction sequence length with fixed time interval. (Left: training; right: validation)

Our LSTM takes a sequence input from the past visit data, using features of `CDRSB`, `ADAS11`, `RAVLT_immediate`, `Hippocampus`, `WholeBrain`, `Entorhinal`, `MidTemp`, `APOE4`, `AGE`, `ICV`, `DX_b`, and outputs predictions on the following 8 time points' diagnosis based on current input time point. For our problem on TADPOLE, the model we build will only concerns about the last network output as our prediction. Another challenge is that each patient has various past visit times, in other words, the input sequence data can be different. So we take advantage of dynamic LSTM model on Tensorflow, which allows to take as input different length of sequence. For each batch, we only need to pad the input to the same length.

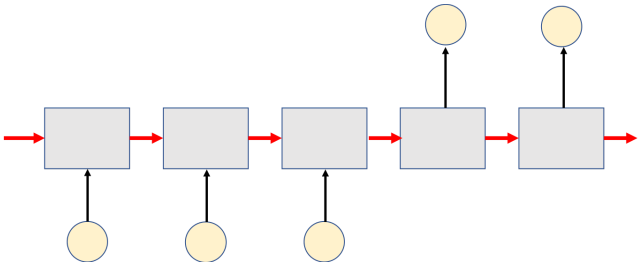

Figure 6. Dynamic LSTM structure for TADPOLE diagnosis prediction

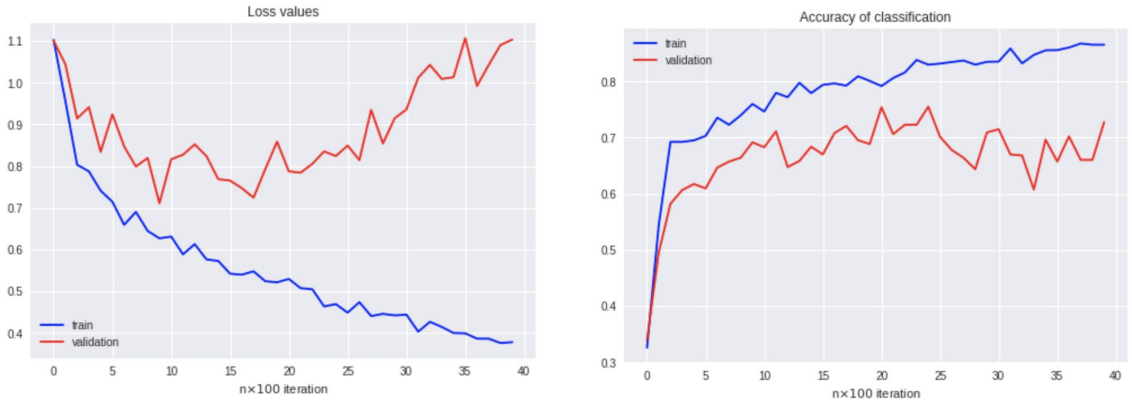

Figure 7. Loss values and accuracy for each epoch

We put an additional layer before softmax to make the basic LSTM model adaptive to multiple values output. Using training data and validation data, we set the appropriate hyperparameters of cell state size and additive layer state size. The loss values and accuracy are plotted below. From the plots, around 2000 epoches the training is going to be overfitting, and we find the optimal training with the classification accuracy of 77.1%.

Conclusion

For our final project we try to build models on TADPOLE challenge using classical machine learning methods as well as deep network such as LSTM. We compare different models' performance on accuracy for classIfication and RMSE for regression. Benchmark method applies moving-forward with the assumption that the probability will be constant along time, which is rather rational considering clinical practice. Our classical machine learning methods include linear models, decision trees with ensemble learning for classification and regression, which takes baseline data and corresponding time difference to yield future prediction. In order to take advantage of properties of sequence data, we try LSTM as a sequential model to capture correlation with previous influence. As a discussion, deep neural networks performs better on regression problem, and LSTM is a great model for time series data on classification problem.