Structure

Model

* With the weighted vector, logistic regression is applied
  + Split training & test data
  + Check roc\_auc\_score
  + Apply test-public data in the model (kaggle score?)
* Perhaps try another model??

내용 todo

모델 분석?/ 선택이유

-> logistic regression - @sunny

-> Param [tuning...@sunny](mailto:tuning...@sunny)

Model Selection

1. Logistic Regression



After processing the given dataset, we got

-categorical (?) 1/0 => predicting category not a quantity

-labelled

-numeric/ **text** data => are IDs text or numeric data(?)

-linearly separable? -> code exists to check this!

- low dimension(?)

- sparse?

=> Based on the characteristics of the **data** – categorical, labelled, binary and text after data processing, we narrowed our learning model down to K-Nearest Neighbor (**KNN**) and Logistic Regression(**LR**). As the dataset has very low dimension but big sample size, it tended to overfit in KNN model and showed slower computational performance. Hence, we chose LR as our learner.

=>mathematical analysis of LR

=>Stochastic Gradient Descent (as an algorithm to solve LR) -- [s1]SGD [s2]SGD Scikit learn library

To calculate optimised weights of our model, we used Stochastic Gradient Descent (SGD). SGD is an optimised gradient descent algorithm which iteratively calculate weights by minimising the regularised training error where stands for a Loss function and represents a Regularisation(penalty) term. The Regulariser prevents overfitting by restricting the size of weights. Finally, computing the function E with SGD algorithm gives us an optimised model with small loss but low model complexity.

For the LR model, the loss function is squared loss and we picked L2 norm as a penalty with a large alpha value to avoid overfitting as below.

est = make\_pipeline(StandardScaler(), SGDClassifier(loss="squared\_loss", penalty="l2", alpha=0.01))

Furthermore, SGD is sensitive to feature scaling. We normalised the feature values using StandardScaler before fitting the model.

How to choose a model

<https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html>

-linearly separable?

<https://www.quora.com/How-can-I-know-whether-my-data-is-linearly-separable#:~:text=2%2Ddimensional%20space.-,If%20you%20can%20draw%20a%20line%20or%20hyperplane%20that%20can,as%20non%20linearly%20separable%20data>.

Sparse Data

<https://stats.stackexchange.com/questions/267322/difference-between-missing-data-and-sparse-data-in-machine-learning-algorithms>

Reference

[s1]SGD

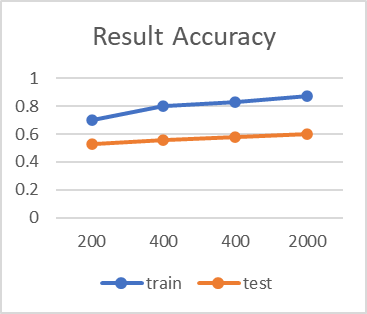
[*Bottou, Léon*](https://en.wikipedia.org/wiki/L%C3%A9on_Bottou); Bousquet, Olivier (2012). [*"The Tradeoffs of Large Scale Learning"*](https://www.google.com/books/edition/_/JPQx7s2L1A8C?hl=en&gbpv=1&pg=PA351). In Sra, Suvrit; Nowozin, Sebastian; Wright, Stephen J. (eds.). Optimization for Machine Learning. Cambridge: MIT Press. pp. 351–368. [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier)) [*978-0-262-01646-9*](https://en.wikipedia.org/wiki/Special:BookSources/978-0-262-01646-9).

[s2]SGD Scikit learn library

[*https://scikit-learn.org/stable/modules/sgd.html#*](https://scikit-learn.org/stable/modules/sgd.html)

*-function doc : param tuning*

[*https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.SGDClassifier.html#sklearn.linear\_model.SGDClassifier*](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier)



As shown in Table 1, our final result performed 0.87 of Accuracy whereas 0-R Baseline showed 0.5. Also, the result showed that the random sample size and the AUC result have propositional relationship. From the lack of our computer resources, however, we could not compute more than 2000 samples. This led an overfitting problem. Small data sample made the algorithm hard to find paths and could not provide enough nodes in the training model.