Milk Yield Prediction Based on Machine Learning Methods

黃昭學
Team 15, Student ID: 109030605
Department of Chemical Engineering,
National Tsing Hua University,
Hsinchu, Taiwan

***Abstract*—This final project is based on the Dairy Herd Improvement (DHI) database provided by the Dairy Association of Taiwan to predict milk production in different regions of Taiwan. During the competition, we used several different types of machine learning models and attempted to get lower root-mean-squared error (RMSE). At the final stage, we tried to use ensemble method and "combine" all the predictions provided by several models we used previously, and finally lowered the RMSE to about 5.64.**

Keywords—data preprocessing, regression, neuron network

I. INTRODUCTION

With the trend of digital transformation, the farming industry has also started to embrace new technology. In the state of reduction in staff, the average milk production in Taiwan has surpassed that of Australia, Germany, and China. And it is still rising year by year. This final project is based on the Dairy Herd Improvement (DHI) database provided by the Dairy Association of Taiwan to predict milk production in different regions of Taiwan. We hope that we could find some key factors affecting milk production and have contributed to intelligent farm management in Taiwan. In this article, we proposed several data preprocessing and regression as well as neural model to make a more comprehensive prediction to the amount of milk.

II. DATA PREPROCESSING

A. Filling Null Data

For the null values in the data, they could be harmful to the training process, and thus the padding of null data is quite critical, which could be filled by domain knowledge and several numerical methods. For example, the feature "最後配種日期", which has 1563 null data, by applying random forest and regressed by "最近分娩日期" & "泌乳天數" (no null data), the blanks were filled. The error of the prediction was expected to be small compare to the standard deviation, and thus, the feature "最後配種日期" was complete.

B. Feature Selecting

Due to the numerous features of the raw data, it is crucial to extract the most efficient ones, which could lead to the more accurate results. First, we could simply attribute features into label and continuous categories, and then label encoding for labels, which would eventually be converted into one-hot vector.

For continuous features, such as datetime and birth dates, by converting them into the units of "months" would be appropriate, which were in the same order of magnitude of other continuous features.

Additionally, there were other raw datasets, "birth", "breed", "spec", the features from these datasets considered to be useful, for the relation between milk yield. However, due to the large amount of null data exists in these features, which were unpredictable, we should pick the features carefully. Therefore, the addition features from the extra files, we only selected "乾乳日期", whose null data is only 753 and could be regressed by related non-null features. The features selected and preprocessing were shown in Figure 1.

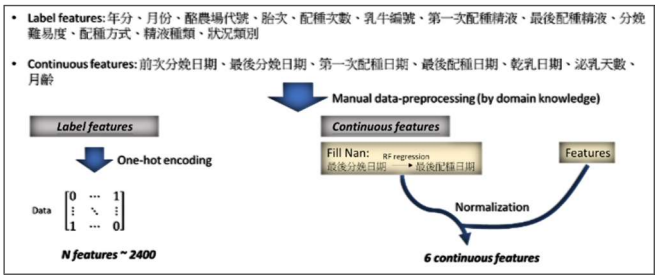

Figure 1. The process of feature extraction and data pre-processing for raw data

III. MODELS

We proposed 3 different types of models, which separately feed by different inputs features extracted from the data after preprocessing.

A. Shallow Model

First, We present 2 simple shallow networks using supervised learning methods. The purpose to implement these regression models is to make a comparison between deep neural network, which was unknown that whether it can achieve a lower loss or not.

- Support Vector Regression (SVR)

The support vector machine (SVM) constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. For classification problem, given training vectors $X_i \in R^n, i = 1, \dots, n$ and a vector $y \in \{1, -1\}^n$, the target was to find w and b such that the prediction given by Equation (1) was satisfied to most samples. Here, φ represents feature mapping function. Regression problem could also apply SVM, which depends on only a subset of the training data, neglecting those with small loss, by penalizing samples whose prediction is at least ε away from their true target. These samples penalize the objective by ζ_i, depending on whether their predictions lie above or below the ε tube (Equation (2)).

$$\text{sign}\{w^T\varphi(x_i) + b\} \cdots\cdots\cdots\cdots (1)$$
$$y_i - w^T\varphi(x_i) - b < \varepsilon + \zeta_i \cdots\cdots\cdots (2)$$

- Random Forest (RF)

As one of the ensemble methods, random forest is to build several estimators independently and then to average their predictions. After weighted, the mixed estimator is usually better than any of the single base estimator (individual decision trees) because of variance reduce, and thus, some errors would be dropout (Figure 2).

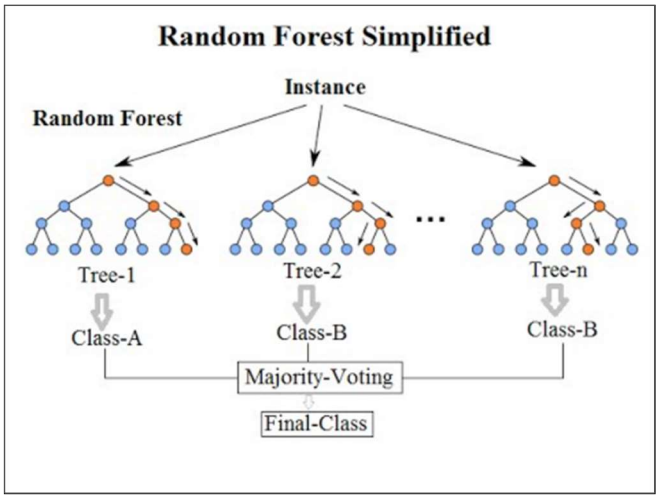

Figure 2. The construction of a random forest

- Input Features and Results

These machine learning methods are applied in regression problem, but the input features of each model are slightly different. For SVM regression model, we only chose “資料年度”, “資料月份”, “酪農場代號”, “胎次”, “泌乳天數”, “最近分娩日期”, “最後配種日期”, “乾乳日期”, these 8 features, for the first 4 features were converted into one-hot vector, and normalized the rest. The result was shown in the first bar in Figure 3, presenting the testing mean-absolute-error with 7.47. The other model, the random forest regression was constructed by 50 estimators, we append 3 large label features, "最後配種精液", "第一次配種精液", "乳牛編號", which had 201, 217, 1991 classes, respectively. However, the support vector regression used here applied non-linear kernel, which was time-consuming than linear one, therefore, the following features should not be consider in SVM regression. Figure 3 present the results of the regression model, the index Cow ID for “乳牛編號”, Last semen for “最後配種精液”, and First Semen for “第一次配種精液”, the testing losses were evaluated on AIDEA. The result of random forest basically present higher performance than SVM, and the number of large label features appended into input data was positively correlated to model performance, the best random forest model could reduce the loss to 6.26.

B. Deep Neuron Network

- Model and Input Features

After data preprocessing, we construct a neural network (NN) model to predict the target. The NN model is quite simple, it is considered that the feature extraction and data preprocessing were more significant to model performance. We present 4-layer neural network, which are all dense layer consisted of 200 hidden units (Figure 4). Same as the regression model by shallow network, we compute the model by appending or deleting features. The basic features defined here were “資料年度”, “資料月份”, “酪農場代號”, “胎次”, “泌乳天數”, “最近分娩日期”, “最後配種日期”, “乾乳日期”, and then separately append the 3 large label features. The result shown in Figure 5, each bar represents the basic features combined with the name of the index. Each training process was at the criterion of mean-absolute-error loss, with learning rate 0.001 and 10 epochs.

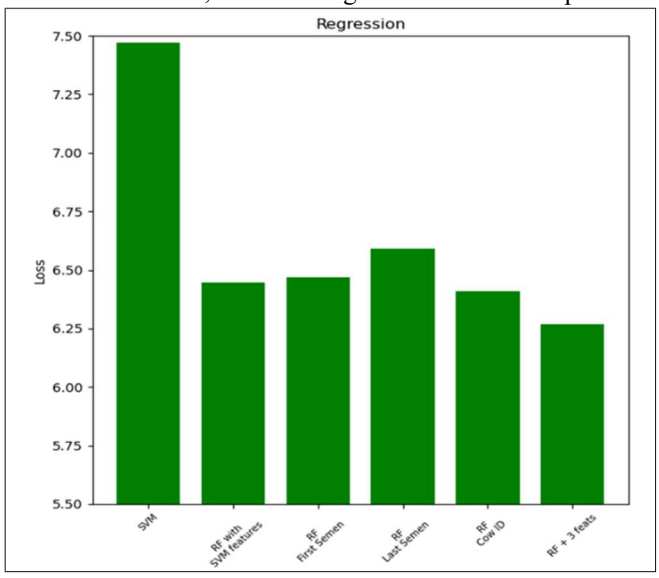

Figure 3. Testing loss vs. the SVM and RF regression model with different input features.

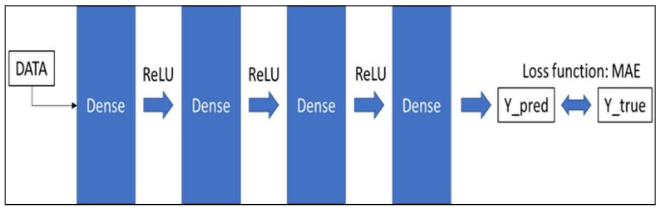

Figure 4. NN model for milk yeld pediction. Each layer contains 200 hidden units, and the activation functions are all rectified linear units (ReLU), loss function is mean absolute error.

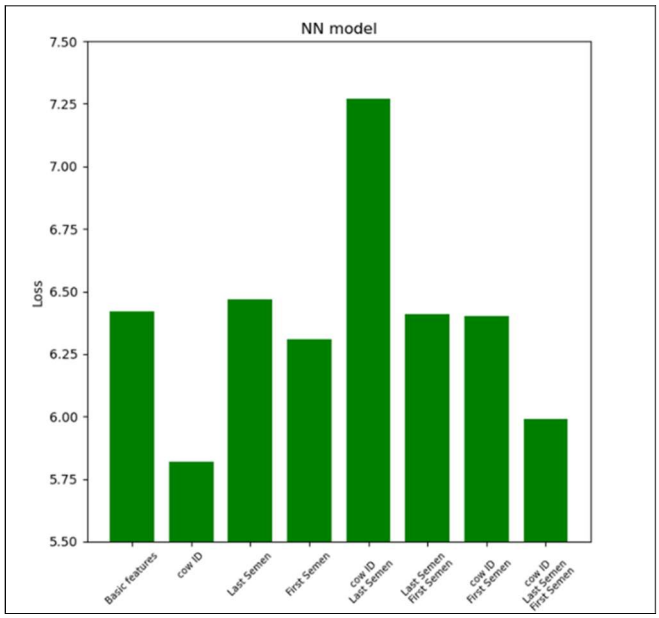

Figure 5. Testing loss vs. NN model with different input features.

- Results

The result of basic features fed into the NN model was 6.42, which was worse than shallow network. Therefore, in the aid of large label features, such as “乳牛編號”, the loss dramatically dropped to 5.82. While combining with either “第一次配種精液” or “最後配種精液”, most of these model slightly reduce the loss (<6.42), and few of them arise the loss. By analyzing the results, we could attribute the error to the null data of “第一次配種精液” and “最後配種精液”, which had the counts of 431 and 1563, and the null data was intractable and unpredictable, and thus, lead to overfitting. Also, as the training epochs increases, the model didn’t enhance the performance and tend to overfit the testing data. (Figure 6)

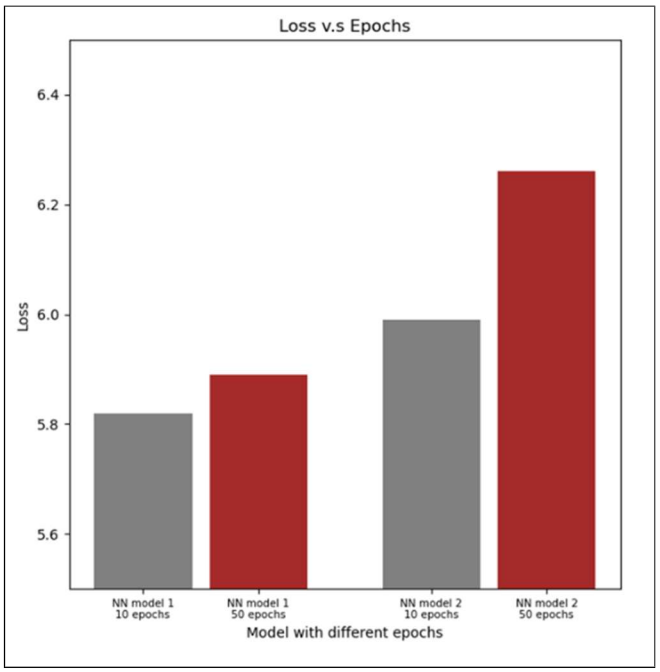

Figure 6. Testing loss vs. NN model with different training epochs. NN model 1 represents the NN model with some basic features. NN model 2 represents another NN model, which input features are some basic features and three large label features.

C. *Residual Neuron Network (ResNet)*

Because the outstanding performance of some well-known neuron network models on computer vision topics, we looked forward to find a technique to convert non-image data into image format. After doing some literature searching, we found a methodology, DeepInsight[1], to transform non-image data to image from Scientific Reports. The general concept of this method is putting the features that are highly “similar” to each other as neighbors on the image. (Figure 7-a)

Recently, the most popular methods to measure the similarity between features within dataset are t-distributed stochastic neighbor embedding (t-SNE). We choose to apply t-SNE to dataset as the same method used in the essay. After t-SNE processing, we could get a 2D plane with lots of points that represent the location of features with respect to their similarities. Therefore, using the relative location between features, we could create a transformation between raw data and images. (Figure 7-b)

However, because of the limitation of t-SNE algorithm, we found that if we keep the more “categorical” features as the input into the DeepInsight pipeline above, the more possibility of “curse of dimensionality” and “crowding problem” we would occur. In order to solve this problem, we tried to focus on thinking deeper to the meaning of “similarity” between different features.

The most popular definition of similarity between two features may be the Pearson correlation coefficient. Therefore, we turned to calculate the correlation matrix of all features and used this information to create the transformation between raw data and images. For example, once doing the process of filling null data and feature selecting, we might get the data with 16 features. Second, we can easily calculate the Pearson correlation matrix between features and plotted it into heatmap image to see the similarity between features. (Figure 8-a) Last but not least, we can sort features by the Pearson correlation coefficients from high to low (using C0 feature as the center), and make those “similar” features get together. (Figure 8-b)

Once we get the image-type of training data, we could put them into some well-known convolutional neuron networks or ResNet. Nevertheless, because of some unknown reasons, the performance of two ResNets (ResNet 50 and ResNet 101) are both very undesirable. (See two ResNet losses at Figure 9) We presumed this phenomenon might be caused by the redundancy of features which are highly correlated.

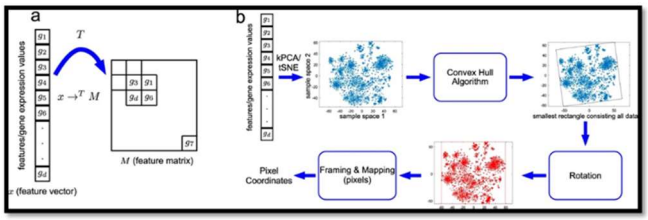

Figure 7. The concept of transformation from feature vector to feature matrix. [1]

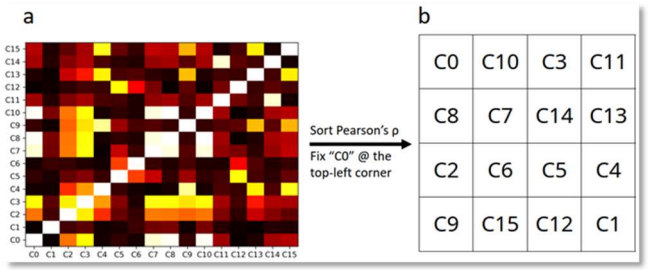

Figure 8. The transformation from Pearson correlation matrix to feature matrix.

D. *Overall Results and Bagging Algorithm*

Constructing 3 different types of models, though they presented deviation of the overall performance, however, each model had it own advantages of extract the efficient features, such as the filter in the ResNet, which could be regarded as feature extraction process. Therefore, the best model from each model type was selected, the best one from random forest regression, the top two of NN model, and the top two from ResNet model. Additionally, applying bootstrap aggregating (bagging) algorithm, the overfitting issue could be significantly decreased, **which eventually compute the RMSE loss about to 5.59 at AIDEA public leaderboard (Figure 12) and 5.83 at private leaderboard. (Figure 13)**

IV. CONCLUSION

In this article, we proposed 3 types of models, which had its own dominance as well as drawbacks. ResNet generated the worst performance, however, the simple neural network and shallow network had better prediction on the target. Obviously, these models were extremely overfitting on testing datasets, by observing on Figure 10, which represented different neural training models. The training loss and validation loss were all lower than 5.5, however, the testing loss could come to 5.9. The behavior had no exception on the random forest regression and ResNet model.

To deal with overfitting problem, though these models had different results individually, by applying ensemble method, we believed the efficient features and parameters could be obtained, and thus, leading to the final loss, which was much lower than that of three models, separately. The process overview could be visualized by the flow chart. (Figure 11)

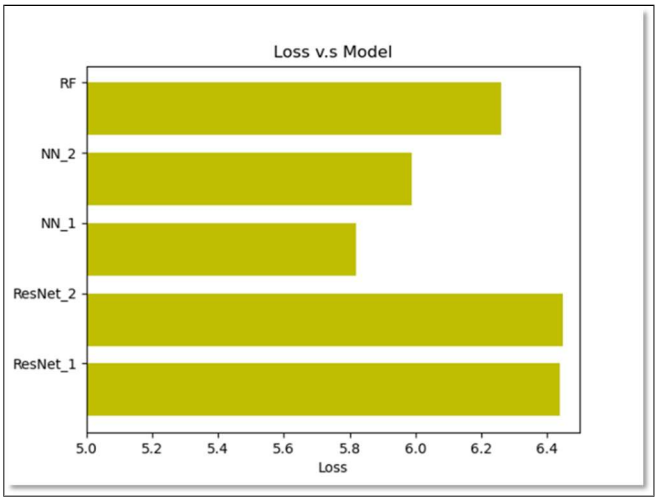

Figure 9. Voting five models from each model type, and generate the overall model by bagging method, the loss was efficiently dropped to 5.64.

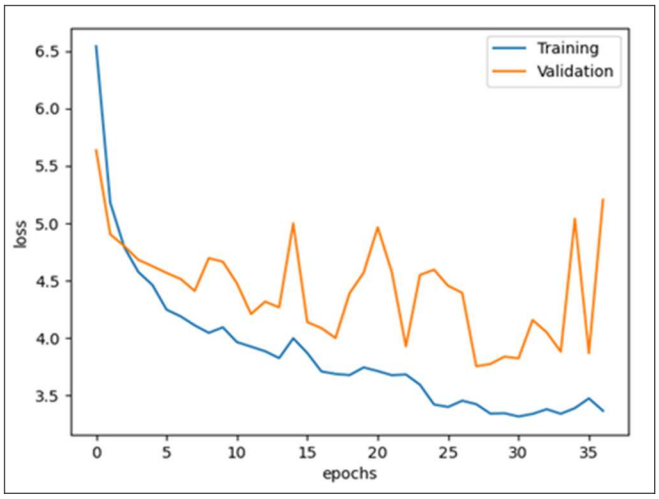

Figure 10. Training loss and validation loss between different epochs of NN model.

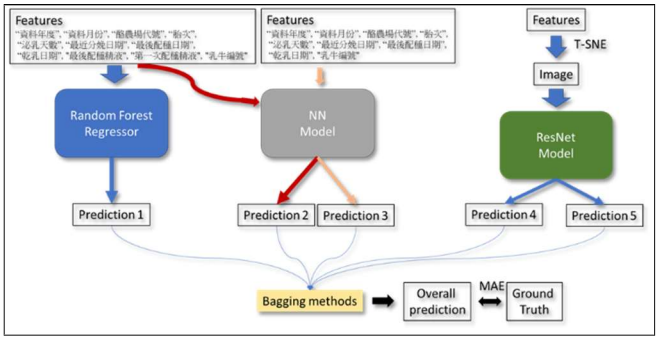

Figure 11. The flow chart of the overall processes we used.

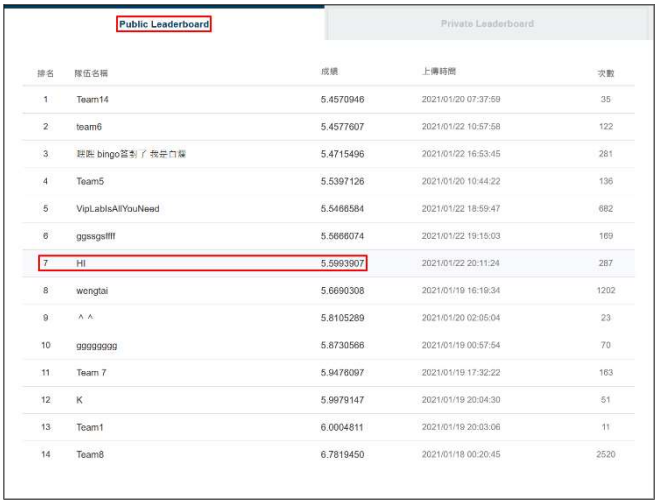

Figure 12. The final rank of the competition (public LB).

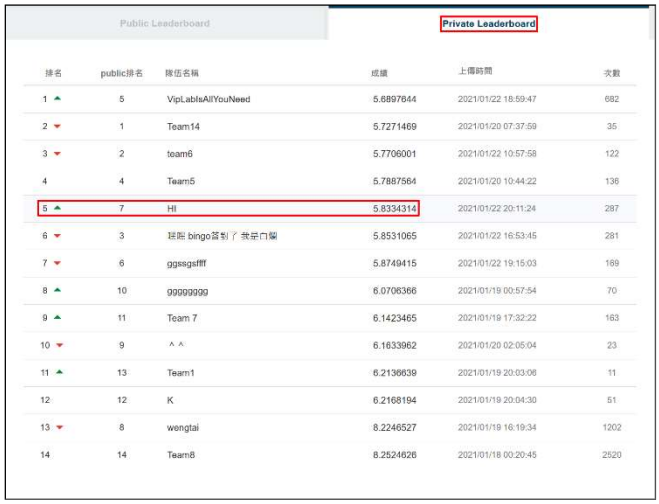

Figure 13. The final rank of the competition (private LB).

V. REFERENCES

[1] Sharma, A., Vans, E., Shigemizu, D. et al. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. Sci Rep 9, 11399 (2019). https://doi.org/10.1038/s41598-019-47765-6