Predicting Cow Health Condition Using Machine Learnings

Evie Cheung

Contents

Executive summary

The aim of this paper is to predict cow sickness in the early detection with automated recorded data. A dataset of 20,154,577 observations involving 10181 cows is used for training models and a sub dataset of 783,494 observations with 409 cows is used for testing. Firstly, in data cleaning upon exploring the data I choose pre and post 14 days window toward the occurrence of sickness and perform log transformation to skewness input variables. Next, I create 1 binary target variable health condition, 1 sickness seasonal variable and 18 cow behavior and sensor variables considering the relative changes to mobs or farm in feature engineering. Then the data are randomly divided into training and validation data sets in a ratio of 50% to 50%. A total of five machine learning models have been used namely Decision Tree (DT), Gradient Boosting (GB), Logistic Regression (LG), Neural Network (NN) and Support Vector Machines (SVM). The evaluation metrics are calculated for determining how well the classifiers have performed, which include misclassification rate, recall, specificity, and precision. Among all the models, Neural Network outperforms the other models with the misclassification rate of 0.30, recall of 0.26, specificity of 0.93 and precision of 0.66.

1 Introduction

The research purpose is to predict which cows are sick and to detect when cows are sick according to the behavior data using machine learning methods. It is crucial to detect cow's sickness earlier since the more accurately we can detect the earlier the farmers can take measures to prevent illnesses from developing into severe conditions, thus reduce economic loss significantly.

Machine learning methods for prediction have been widely used in cow's sickness detection research, such as mastitis prediction (Anglart et al., 2020; Cavero et al., 2008; Ebrahimi et al., 2019) and skin disease prediction (Dofadar et al., 2022). The commonly used methods are Logistic Regression, Decision Tree, Random Forest, Support Vector Classification, K-Neighbors, Gaussian Naïve Bayes, Stochastic Gradient Descent, Gradient Boosting, Light Gradient Boosted Machine, eXtreme Gradient Boosting (XGBoost). Logistic Regression (LR) is an approach to examine the link between multiple independent variables and dependent variables (Kleinbaum & Klein, 2010c, 2010b, 2010a). Decision Tree is a commonly used supervised learning technique since it is not only easily interpreted and visualized as a tree, but also does not require much data processing (Dofadar et al., 2022). Random Forest is a group of decision trees and has a better performance than an individual tree (LeCun et al., 2015). Support Vector Classification finds hyperplanes, which can be called the best decision boundary. These hyperplanes are created to separates the multi-dimensional space and distinctly classify the data points (Dofadar et al., 2022). K-Neighbors algorithm classifies new data points by measuring their similarity to the available data points. This similarity is determined by calculating the Euclidean distance between the K nearest neighbors and the new data point (Dofadar et al., 2022). Gaussian Naïve Bayes assumes the independence of features and each feature is considered to contribute equally to the output. This probabilistic machine learning technique assumes that the continuous values associated with the features follow a Gaussian or Normal distribution (Dofadar et al., 2022; Lewis, 1998). Stochastic Gradient Descent is a variation of Gradient Descent, which is used to minimize a cost function. Instead of using the whole observations, it is calculated using a random little part of the observations (Dofadar et al., 2022). Gradient Boosting employs a boosting technique to create a powerful predictive model by iteratively learning from decision trees. This iterative process helps minimize the bias error, leading to improved prediction accuracy (Dofadar et al., 2022). Light Gradient Boosted Machine is an improved version of the Gradient Boosting technique that incorporates an automated feature extraction mechanism. It prioritizes boosting examples with larger gradients, enabling effective learning (Dofadar et al., 2022). XGBoost is an upgraded version of gradient boosting decision tree algorithm, which can perform more accurately in less amount of time (Dofadar et al., 2022).

The research regarding animal sickness behavior indicates that different kinds of sickness have various behaviors patterns. The most commonly recognized behavior patterns of animals at the onset of sickness are lethargy, depression, anorexia, and reduction in grooming (Hart, 1988). However, unlike the typical sickness behavior, if cows has been diagnosed with mastitis, they do not increase the lying time, but instead stand more and avoid lying on the side of the inflamed udder quarter (Ankinakatte et al., 2013; Siivonen et al., 2011). Besides, as for the lameness, the most important factors are number of meals, average feed intake per meal and average duration of a meal, which means that lame cows feed in fewer and shorter meals with a decreased intake per meal (Grimm et al., 2019). Thus, behavior patterns such as resting, feeding and ruminating play a very important role in predicting sickness type.

2 Materials and Methodologies

2.1 Materials

2.1.1 Data Collection

The data is provided by Halter, which is a pioneering biology technology company in New Zealand leveraging technology for dairy farmers management. The data is collected automatically by the solar-powered collar on the cow's neck across dairy herds and then sent through the air, real-time updates back to Halter's system, and processed. In every minute there will be 3000 data points per cow.

2.1.2 Dataset Description

Totally two datasets are used in the study namely training dataset and test dataset. Training dataset is used for training and validating models. It contains 10181 cows in 155 farms, in which 3624 cows are sick and 6557 cows are healthy. Test dataset is selected alone from a subset of farms to provide an independent score of the best model, which includes 409 cows in 25 farms.

2.1.3 Variable Description

The dataset contains three types of variables: metadata, behavior features, and sensor features. Metadata describes the basic information of the dataset. Behavior features include grazing, resting and ruminating, among which each is measured in minutes per hour and describes the amount of time that a cow spends eating, resting and ruminating. Sensor features illustrate the raw data from the sensors, which contains OBDA, pitch and roll. ODBA (Overall Dynamic Body Acceleration) is a measure of the acceleration a body experiences. This is reported by the collar's accelerometer. Pitch measures how far up or down a cow's neck is tiling. If the pitch is close to zero, then the cow's neck is close to horizontal, and a more negative pitch indicates the cow's neck is tilted down towards the ground. Roll measures the absolute value of roll of the collar to the left or right. This is reported by the collar's accelerometer. It is worth to mention that this is the absolute value of roll, so 0 represents the collar centered and a positive number represents how far off-center the collar is. As the collars are sampling every 25 hertz, 25 times a second, the value of roll-related variables are extremely high in every observation. The following table shows the variable descriptions.

Table 1: variable description

Type	Segment	Variable Name	Description
Metadata	Metadata	UTC Timestamp	The UTC timestamp of the row's observations
		Cattle ID	A UUID specifying the cow which the observations relate to
		Farm ID	A UUID specifying the farm which the cow belongs to
		Sickness Type	The type of sickness the farmer recorded. Common sicknesses include milk fever, lameness, and mastitis.
		Hours Since Sickness	the number of hours before (negative) or after (positive) that row is from when the cow was marked as sick by the farmer
Behavior Features	Grazing	Grazing	the amount of time the cow spent grazing in the specified hour
		Mob Median Grazing	the median amount of time spent grazing in the specified hour, across all cows who were in that cow's mob during the specified hour
		Mob Std Grazing	the standard deviation of time spent grazing in the specified hour, across all cows who were in that cow's mob during the specified hour
		Farm Average Grazing	the average amount of time spent grazing in the specified hour, across all cows on the farm
	Resting	Resting	the amount of time the cow spent resting in the specified hour
		Mob Median Resting	the median amount of time spent resting in the specified hour, across all cows who were in that cow's mob during the specified hour
		Mob Std Resting	the standard deviation of time spent resting in the specified hour, across all cows who were in that cow's mob during the specified hour
		Farm Average Resting	the average amount of time spent resting in the specified hour, across all cows on the farm
	Rumination	Rumination	the amount of time the cow spent grazing in the specified hour

		Mob Median Rumination	the median amount of time spent ruminating in the specified hour, across all cows who were in that cow's mob during the specified hour
		Mob Std Rumination	the standard deviation of time spent rumitating in the specified hour, across all cows who were in that cow's mob during the specified hour
		Farm Average Rumination	the average amount of time spent ruminating in the specified hour, across all cows on the farm
Sensor Features	ODBA	ODBA Average	the average ODBA for the that cow during the specified hour
		ODBA Std	the standard deviation of ODBA for that cow during the specified hour
		Farm Average ODBA Average	The average of ODBA average across all cows on the farm, for the specified hour
		Farm Average ODBA Std	The average of ODBA std across all cows on the farm, for the specified hour
	Pitch	Pitch Average	the average pitch for the that cow during the specified hour
		Pitch Std	the standard deviation of pitch for that cow during the specified hour
		Farm Average Pitch Average	The average of pitch average across all cows on the farm, for the specified hour
		Farm Average Pitch Std	The average of pitch std across all cows on the farm, for the specified hour
	Roll	Roll Average	the average roll for the that cow during the specified hour
		Roll Std	the standard deviation of roll for that cow during the specified hour
		Farm Average Roll Average	The average of roll average across all cows on the farm, for the specified hour
		Farm Average Roll Std	The average of roll std across all cows on the farm, for the specified hour

2.2 Methodologies

2.2.1 Algorithm description

1. Decision Tree (DT)

Decision Tree is a type of supervised learning algorithm that is mostly used in classification problems. It works by creating a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

2. Gradient Boosting (GB)

Gradient Boosting is an ensemble learning method that produces a prediction model in the form of decision trees. It builds the model in a stage-wise manner, and it generalizes them by allowing optimization of loss function.

3. Logistic Regression (LG)

Logistic regression is used in classification problems. It uses the logistic function to find a model that fits with the data points. The function can then predict the likelihood of an instance belonging to a class.

4. Neural Network (NN)

Inspired by the human brain, Neural Network creates an artificial neural network that via an algorithm allows the computer to learn by incorporating new data. They are particularly effective in identifying complex patterns and relationships within the data.

5. Support Vector Machine (SVM)

Support Vector Machine is a type of supervised machine learning classification algorithm. SVMs are based on the idea of finding a hyperplane that best separates the features into different domains.

2.2.2 Evaluation metrics

The primary goal of this challenge is to correctly "alert" as many cows as possible as being sick. So, evaluation metrics such as misclassification rate, recall, specificity, and precision are used to measure the quality and performance of machine learning models. Table 2 shows a comprehensive understanding of the calculation of evaluation metrics.

1. **Misclassification Rate**: This is a metric used to quantify the accuracy of a classification model, which is calculated as the number of incorrect predictions divided by the total number of predictions. A lower misclassification rate is typically better as it indicates a higher accuracy.

2. **Recall (Sensitivity)**: This is the proportion of actual positive cases that are correctly identified by the model. It is calculated as the number of true positives divided by the sum of true positives and false negatives. A higher recall means fewer actual cases of sickness are missed.

3. **Specificity**: This is the proportion of actual negative cases that are correctly identified by the model. It's calculated as the number of true negatives divided by the sum of true negatives and false positives. A higher specificity means fewer healthy cows are incorrectly labeled as sick.

4. **Precision**: This is the proportion of predicted positive cases that are actually positive. In other words, it's the number of correctly predicted sick cows divided by the total number of cows predicted as sick. Higher precision values are better, as they indicate fewer false positives.

Table 2: calculation of evaluation metrics

	Disease +	**Disease -**	**Total**	**Formula**
Test +	True Positive (TP)	False Positive (FP)	TP+FP	**Precision or Positive Predict Value (PPV)**=TP/TP+FP
Test -	False Negative (FN)	True Negative (TN)	FN+TN	**Negative Predict Value (NPV)**=TN/FN+TN
Total	TP+FN	FP+TN	–	
Formula	**Sensitivity**=TP/TP+FN	**Specificity**=TN/FP+TN		

3 Results

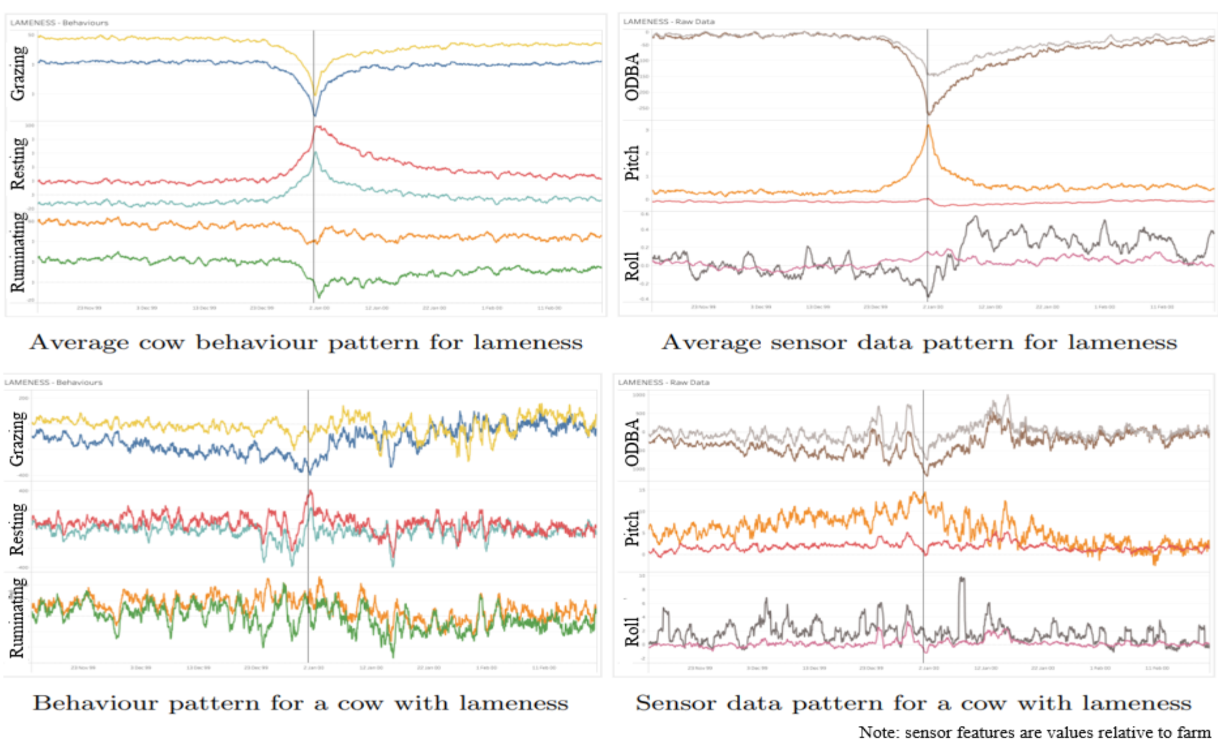
3.1 Data Preprocessing

3.1.1 Data Cleaning

3.1.1.1 Choosing pre and post 14 days window

Figure 1 is provided by Halter, which shows plots based on the timestamp near the timing of sickness occurrences by comparing various features of the average of farms and a single cow. It indicates that if cows are sick their behavior starts to change before approximately two weeks and the same with their recovery. Therefore, a pre and post 14 days window has been extracted from the original train dataset. The original dataset had 20,154,577 observations, and after choosing the 14-day time window, 3,230,524 remained in the dataset. In Figure 1, it also indicates that resting, pitch and roll probably have a positive relationship with health condition. However, grazing, ruminating, and OBDA may have a negative relationship with health condition.

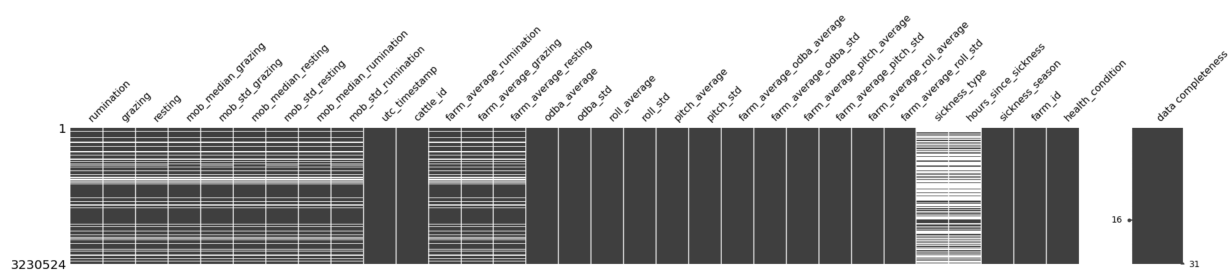
Figure1: the average values of various features when centered on the date the farmer recorded the cow's sickness, compared to the values of the same features for a single cow



3.1.1.2 Data imputation

Figure 2 shows the missing data information of the train dataset, which illustrates that there are 14 variables existing missing data, among which missing data in “sickness type” and “hours since sickness” are all from healthy cows, while in the others most of them are from healthy cows. Therefore, for nominal variables like “sickness type” and “hours since sickness”, the impute method of "Count" is used to replaces missing values with the most frequent category in the non-missing values of that variable. Besides, for the other interval variables, the impute method of "Mean" is used to replaces missing values with the mean value of the non-missing values for that variable.

Figure 2: missing data analysis



3.1.2 Data Transform

Upon exploring the predictor variables, it indicates that Grazing, Rumination, ODBA, Pitch, and Roll input variables show some degree of skewness in their distribution. Thus, I use the log transformation to regularize the skewed distributions. Figure 2 and Figure 3 shows the difference before and after transformation.

Figure 2: variables distribution before transformation

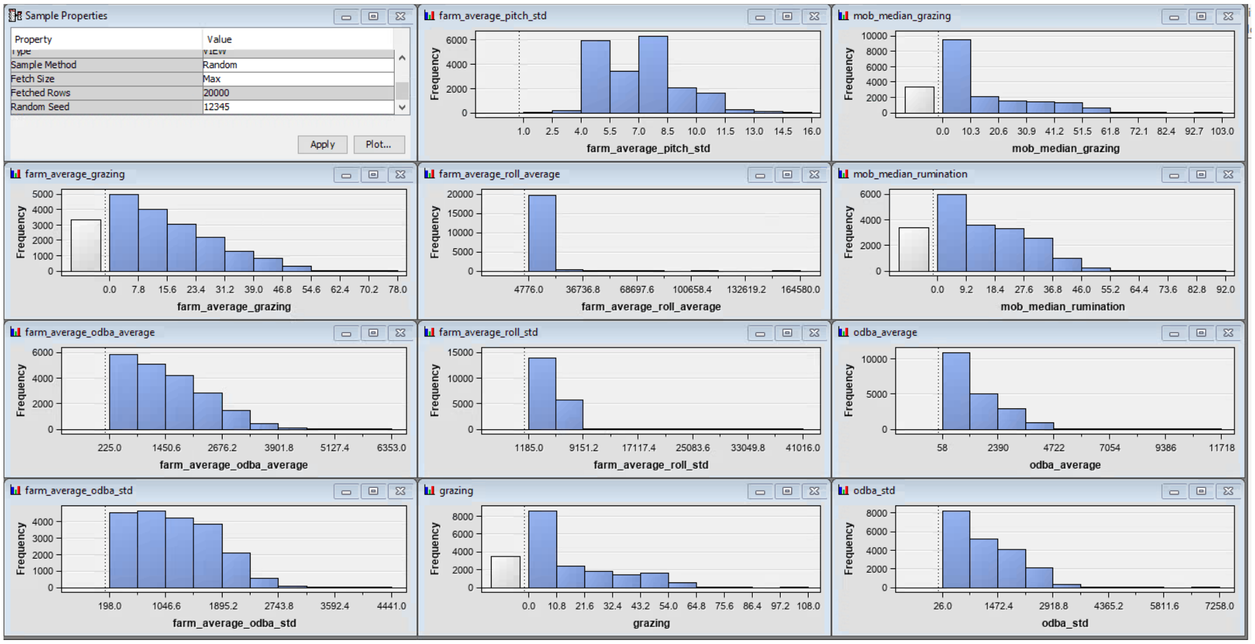


Figure 3: variables distribution after transformation

3.2 Feature Engineering

In the Feature Engineering process, three distinct types of features have been created. Firstly, “health condition” has been incorporated as a binary variable to predict sickness, where 0 represents good health and 1 indicates sickness. Secondly, to consider the influence of seasonality on sickness, a new variable "sickness seasons" has been introduced. This variable is determined by the timing of illness onset, effectively portraying the seasonal implications on health. Lastly, a substantial set of 18 new variables has been incorporated, which can be separated into three categories. The first two categories are cow behavior and sensor features' relative change to farms or mobs. These variables are calibrated in relation to the changes on the farm or within the mobs, primarily owing to the fact that farms exhibit seasonal variations. For example, during spring when the grass grows quickly, providing abundant food for cows and therefore leading to higher grazing time. However, as summer progresses, conditions become drier, leading to a reduction in available feed and consequent grazing time. In winter, the situation is worse, as grass growth stalls significantly, and feed becomes less. Consequently, observing the relative changes in behaviors and sensor features becomes imperative. The third category focuses on analyzing mobs relative to farms, considering that farms operate with distinct mobs. Overall, these comprehensive approach to feature engineering are expected to enhance our predictive capabilities significantly. Table 3 delineates the detail information of the new variables.

Table 3: new variables description

First Category	Second Category	Third Category	Variable Name	Algorithm/Formula
Adding Health Condition	-	-	Health_Condition	if the sickness_type is null, then health_condition is 0; if the sickness_type is not null, then health_condition is 1.
Adding Seasonal Variable	-	-	Sickness_Season	if the UTC Timestamp of sickness falls between Sep to Nov, then it's Spring; if the UTC Timestamp of sickness falls between Dec to Feb, then it's Summer; if the UTC Timestamp of sickness falls between Mar to May then it's Autumn; if the UTC Timestamp of sickness falls between Jun to Aug, then it's Winter.
Adjusting Relative to Farm/mob	Behavior Features Relative to	Grazing	Grazing_Minus_Farm_Avg	Grazing - Farm_Average_Grazing
			Grazing_Minus_Mob_Med	Grazing - Mob_Median_Grazing
			Grazing_Minus_Mob_Std	Grazing - Mob_Std_Grazing
		Resting	Resting_Minus_Farm_Avg	Resting - Farm_Average_Grazing

	Farm/Mob		Resting_Minus_Mob_Med	Resting - Mob_Median_Grazing
			Resting_Minus_Mob_Std	Resting - Mob_Std_Grazing
		Ruminating	Ruminating_Minus_Farm_Avg	Ruminating - Farm_Average_Grazing
			Ruminating_Minus_Mob_Med	Ruminating - Mob_Median_Grazing
			Ruminating_Minus_Mob_Std	Ruminating - Mob_Std_Grazing
	Sensor Features Relative to Farm/Mob	ODBA	OBDA_Minus_Farm_Avg	OBDA_Average - Farm_Average_OBDA_Average
			OBDA_Minus_Farm_Std	OBDA_Std - Farm_Average_OBDA_Std
		Pitch	Pitch_Minus_Farm_Avg	Pitch_Average - Farm_Average_Pitch_Average
			Pitch_Minus_Farm_Std	Pitch_Std - Farm_Average_Pitch_Std
		Roll	Roll_Minus_Farm_Avg	Roll_Average - Farm_Average_Roll_Average
			Roll_Minus_Farm_Std	Roll_Std - Farm_Average_Roll_Std
	Mob Relative to Farm	Grazing	Mob_Minus_Farm_Grazing	Mob_Median_Grazing - Farm_Average_Grazing
		Resting	Mob_Minus_Farm_Resting	Mob_Median_Resting - Farm_Average_Resting
		Ruminating	Mob_Minus_Farm_Ruminating	Mob_Median_Ruminating - Farm_Average_Ruminating

3.3 Models Performance Comparison

After running Decision Tree, Gradient Boosting, Logistic Regression, Neural Network, Support Vector Machines, the results of misclassification rate, recall, specificity, and precision is calculated as shown in Figure 4.

The best misclassification rate for validation data in predicting the cow's health condition is 0.30, which belongs to the Neural Network, followed by Gradient Boosting (0.31), Logistic Regression (0.32), Decision Tree (0.32), and SVM (0.34). The highest value of recall is 0.26 using Neural Network, while the other models are 0.22, 0.17, 0.17, and 0.16 respectively. As for the specificity, Gradient Boosting and Decision Tree shows the highest value of 0.96 and Logistic Regression is the lowest with 0.92. Regarding precision, the best precision is gained by using Gradient Boosting (0.70). The precision of Decision Tree, Neural Network, Logistic Regression, SVM are 0.68, 0.66, 0.61 and 0.54, respectively.

Figure 5 shows the evaluation metrics before model optimization, which is the one represented in the group presentation using four machine learning models (LG, DT, GB, and NN) without implementation of the log transformation and feature engineering. The misclassification rates of Logistic Regression, Decision Tree, Gradient Boosting, and Neural Network are 0.32, 0.32, 0.33, and 0.31respectively. Neural Network has the highest recall of 0.28, along with 0.21 of Decision Tree, 0.19 of Logistic Regression, and 0.10 of Gradient Boosting. Gradient Boosting achieves the highest specificity of 0.98, whereas Logistic Regression and Decision Tree have the value of 0.93 as well as Neural Network has the value of 0.91. As for the precision, Gradient Boosting, Neural Network, Logistic Regression, and Decision Tree has the value of 0.75, 0.62, 0.61, and 0.60.

By comparing Figure 4 and Figure 5, it can be shown that the overall performance of the model after optimization has improved over the pre-optimization. For instance, the misclassification rate of Neural Network decreases from 0.31 to 0.30 while Gradient Boosting from 0.33 to 0.31. Besides, the recall of GB and LG improve 0.07 and 0.03 respectively. The specificity of DT and NN increase 0.03 and 0.02. Finally, the precision of DT, and NN also increase from 0.60 to 0.68, and 0.62 to 0.66.

Figure 4: evaluation metrics of models after optimization (performing transformation and feature engineering)

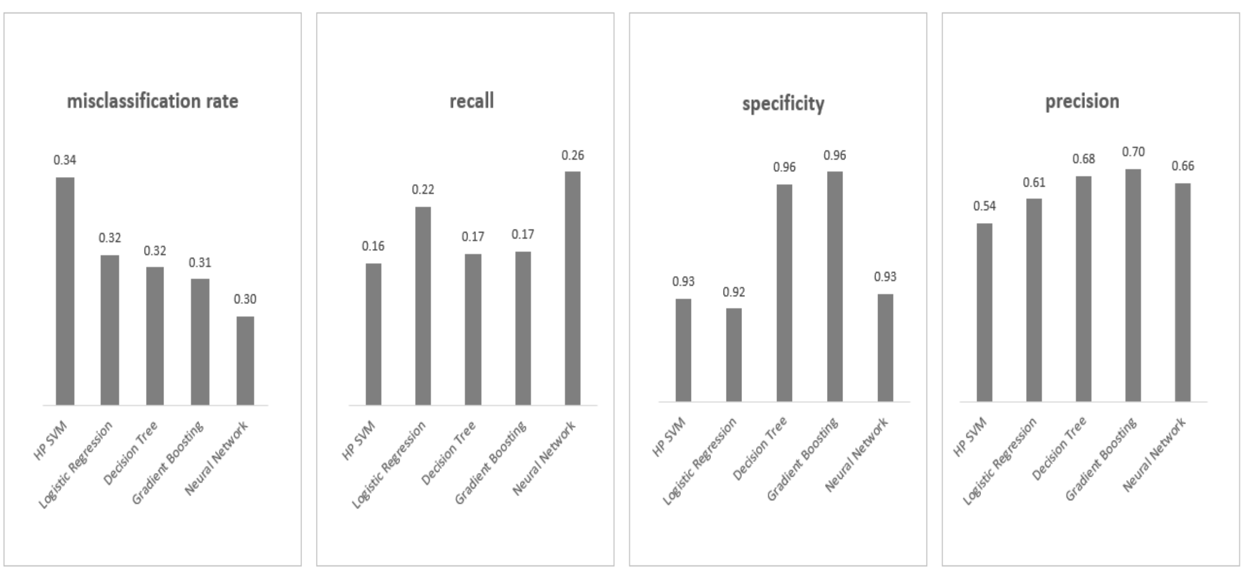
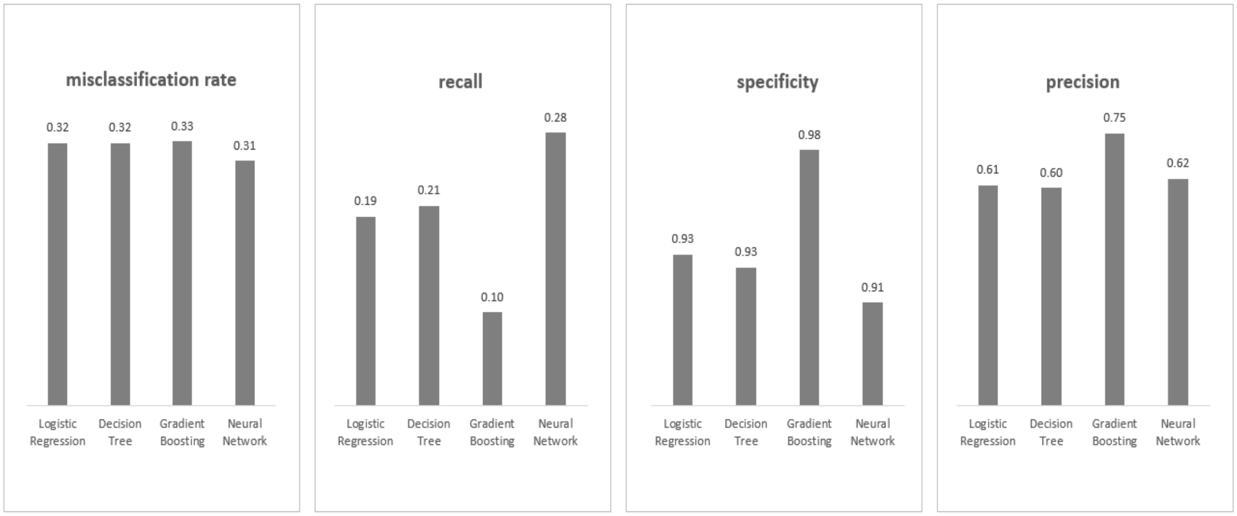


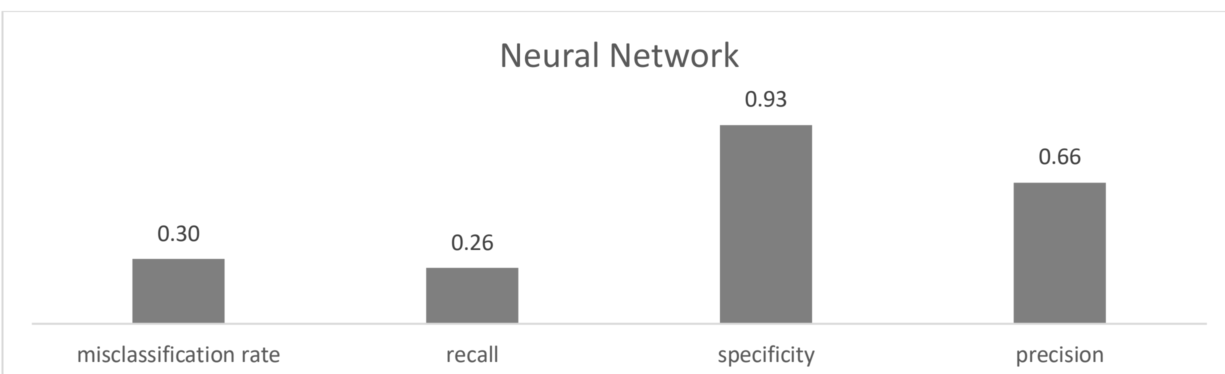
Figure 5: evaluation metrics of models before optimization (the one showed in the presentation)



3.4 Model Implementation

Utilizing the best model Neural Network on the sub test dataset, the evaluation metrics is shown as Figure 6. It can be identified that the misclassification rate is 0.30, recall is 0.26, specificity is 0.93, and precision is 0.66.

Figure 6: evaluation metrics of Neural Network on test data



3.5 Variable Importance

Table 4 shows the top five important input variables of DT, GB, NN, and LG. It can be identified that different models have various variable importance. To be more specific, both DT and GB have the most importance variables of sickness season, ODBA relative to farms, and rumination of mobs relative to farms, while DT specializes the variable grazing relative to mobs and GB prefers the variable pitch relative to farms. NN's most important variables are sickness season, mob standard variance of resting, mob grazing median relative to that of farms, grazing relative to farms, and resting relative to mobs. LG values all the log transformed variables.

Table 4: variable importance of different models

Sequence	Decision Tree	Gradient Boosting	Neural Network	Logistic Regression
1	sickness_season	sickness_season	sickness_season	IMP_LOG_farm_average_grazing
2	IMP_ODBA_minus_farm_average	IMP_ODBA_minus_farm_average	M_mob_std_resting	IMP_LOG_farm_average_odba_averag
3	M_LOG_farm_average_grazing	IMP_mob_minus_farm_rum_med	IMP_mob_minus_farm_grave_med	IMP_LOG_farm_average_odba_std
4	IMP_mob_minus_farm_rum_med	IMP_ODBA_minus_farm_std	IMP_graze_minus_farm_av	IMP_LOG_farm_average_pitch_std
5	M_graze_minus_mob_med	IMP_pitch_minus_farm_std	M_rest_minus_mob_std	IMP_LOG_farm_average_roll_averag

4 Discussion

Three perspectives are delineated in the evaluation of model performance that are models' horizonal comparison, models' vertical improvement and variable importance. Firstly, from the perspective of the horizonal comparison of the models, it is necessary to consider multiple evaluation metrics to get a comprehensive understanding of the model's performance. Thus, Neural Network is the best approach in predicting cow's health condition based on the behavior and sensor features with the lowest misclassification rate 0.30, highest recall 0.26, and relatively high specificity 0.93 and precision 0.66.

What's more, from the perspective of the vertical improvement of the models, the misclassification, recall, specificity and precision of DT, GB, LG, and NN have improved in different degrees. It indicates that it is useful to perform log transformation in skewness variables and creating 1 sickness season variable, 15 new cow behavior and sensor variables relative to mobs or farms, and 3 behavior variables of mobs relative to farms. Furthermore, from the variable importance table of four models, these new variables are among the most important variables in models. Therefore, it is crucial to perform transformations, consider the seasonal impact to cow sickness, and adjust relative changes to mobs and farms.

Finally, as for the variable importance, it is reasonable that different models have differentiated variable importance due to the difference in model algorithm and model complexity. For example, decision tree and decision tree-based algorithm like Gradient Boosting and Random Forest determine variable importance based on how much a variable contributes to reducing impurity in the tree splits. Logistic Regression examines the magnitude of the coefficients associated with each variable to assess the variable importance. Neural Network determines variable importance in a more complex way due to the non-linear and distributed nature. Therefore, different models may yield varying results of variable importance.

Although the study succeed in predicting the cow's health condition, there are still three aspects needing improvements in the future that are response variable replacement, predictor variable optimization, and diversity of model implementation. First of all, in addition to the binary response variable "health condition", nominal variable "sickness type" can be considered as different sicknesses show varying behavior patterns in resting and grazing. Besides, two more predictor variables features can be added which are factor adjusting for the time of year and farm effect mentioned by the Halter data scientist in a precious business view. The reason why considering adjusting the time of year is because sickness like milk fever and mastitis usually come through in July or August, cows prefer to sit in a paddock in a health condition before lactating or giving birth. So, adding some sort of factor that adjusts for the time of year could be an important part in the model. Besides, as for farm effect, some farms have hundreds of sicknesses, while some farms have one. So, it would be interesting to see whether there is some farm effect to the cow sickness. Lastly, more kinds of machine learning models like Random Forest, Ensemble models and Naïve Bayes can be conducted using another predicting tool like Python.

5 Conclusion

The research aims to predict cow sickness in the early detection based on five different machine learning models namely DT, LG, GB, NN and SVM. As a result, NN shows the best performance in evaluation metrics. In the future, there still needs three aspects of research which are the replacement of target variable, optimization in predictor variables and more diversity in model implementation.

References

Anglart, D., Hallén-Sandgren, C., Emanuelson, U., & Rönnegård, L. (2020). Comparison of methods for predicting cow composite somatic cell counts. *Journal of Dairy Science*, *103*(9), 8433–8442. https://doi.org/10.3168/jds.2020-18320

Ankinakatte, S., Norberg, E., Løvendahl, P., Edwards, D., & Højsgaard, S. (2013). Predicting mastitis in dairy cows using neural networks and generalized additive models: A comparison. *Computers and Electronics in Agriculture*, *99*, 1–6. https://doi.org/10.1016/j.compag.2013.08.024

Cavero, D., Tölle, K.-H., Henze, C., Buxadé, C., & Krieter, J. (2008). Mastitis detection in dairy cows by application of neural networks. *Livestock Science*, *114*(2), 280–286. https://doi.org/10.1016/j.livsci.2007.05.012

Dofadar, D. F., Abdullah, H. Md., Khan, R. H., Rahman, R., & Ahmed, Md. S. (2022). A Comparative Analysis of Lumpy Skin Disease Prediction Through Machine Learning Approaches. *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 1–4. https://doi.org/10.1109/IICAIET55139.2022.9936742

Ebrahimi, M., Mohammadi-Dehcheshmeh, M., Ebrahimie, E., & Petrovski, K. R. (2019). Comprehensive analysis of machine learning models for prediction of sub-clinical mastitis: Deep Learning and Gradient-Boosted Trees outperform other models. *Computers in Biology and Medicine*, *114*, 103456. https://doi.org/10.1016/j.compbiomed.2019.103456

Grimm, K., Haidn, B., Erhard, M., Tremblay, M., & Döpfer, D. (2019). New insights into the association between lameness, behavior, and performance in Simmental cows. *Journal of Dairy Science*, *102*(3), 2453–2468. https://doi.org/10.3168/jds.2018-15035

Hart, B. (1988). Hart, B. L. Biological basis of the behavior of sick animals. Neurosci. Biobehav. Rev. 12, 123-137. *Neuroscience and Biobehavioral Reviews*, *12*, 123–137. https://doi.org/10.1016/S0149-7634(88)80004-6

Kleinbaum, D. G., & Klein, M. (2010a). Important Special Cases of the Logistic Model. In D. G. Kleinbaum & M. Klein (Eds.), *Logistic Regression: A Self-Learning Text* (pp. 41–71). Springer. https://doi.org/10.1007/978-1-4419-1742-3_2

Kleinbaum, D. G., & Klein, M. (2010b). Introduction to Logistic Regression. In D. G. Kleinbaum & M. Klein (Eds.), *Logistic Regression: A Self-Learning Text* (pp. 1–39). Springer. https://doi.org/10.1007/978-1-4419-1742-3_1

Kleinbaum, D. G., & Klein, M. (2010c). *Logistic Regression*. Springer. https://doi.org/10.1007/978-1-4419-1742-3

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), Article 7553. https://doi.org/10.1038/nature14539

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In C. Nédellec & C. Rouveirol (Eds.), *Machine Learning: ECML-98* (pp. 4–15). Springer. https://doi.org/10.1007/BFb0026666

Siivonen, J., Taponen, S., Hovinen, M., Pastell, M., Lensink, B. J., Pyörälä, S., & Hänninen, L. (2011). Impact of acute clinical mastitis on cow behaviour. *Applied Animal Behaviour Science*, *132*(3), 101–106. https://doi.org/10.1016/j.applanim.2011.04.005

Appendix

Table 5: Variable Importance of Decision Tree

Variable Name	Label	Importance
sickness_season		1.00
IMP_ODBA_minus_farm_average	Imputed ODBA_minus_farm_average	0.88
M_LOG_farm_average_grazing	Imputation Indicator for LOG_farm_average_grazing	0.59
IMP_mob_minus_farm_rum_med	Imputed mob_minus_farm_rum_med	0.53
M_graze_minus_mob_med	Imputation Indicator for graze_minus_mob_med	0.42
IMP_ODBA_minus_farm_std	Imputed ODBA_minus_farm_std	0.38
IMP_pitch_minus_farm_std	Imputed pitch_minus_farm_std	0.29
IMP_mob_std_grazing	Imputed mob_std_grazing	0.29
IMP_mob_std_rumination	Imputed mob_std_rumination	0.27
IMP_rumi_minus_farm_avg	Imputed rumi_minus_farm_avg	0.25
M_graze_minus_mob_std	Imputation Indicator for graze_minus_mob_std	0.24
IMP_mob_minus_farm_rest_med	Imputed mob_minus_farm_rest_med	0.23
IMP_roll_std	Imputed roll_std	0.24
IMP_roll_average	Imputed roll_average	0.22
IMP_LOG_farm_average_roll_std	Imputed: Transformed farm_average_roll_std	0.14
IMP_rumination	Imputed rumination	0.16
IMP_pitch_minus_farm_average	Imputed pitch_minus_farm_average	0.15
IMP_mob_std_resting	Imputed mob_std_resting	0.11
IMP_LOG_odba_std	Imputed: Transformed odba_std	0.15
IMP_farm_average_resting	Imputed farm_average_resting	0.07
IMP_mob_median_resting	Imputed mob_median_resting	0.07
IMP_LOG_odba_average	Imputed: Transformed odba_average	0.07

Table 6: Variable Importance of Gradient Boosting

Obs	NAME	LABEL	VIMPORTANCE
1	sickness_season		1
2	IMP_ODBA_minus_farm_average	Imputed ODBA_minus_farm_average	0.78
3	IMP_mob_minus_farm_rum_med	Imputed mob_minus_farm_rum_med	0.72
4	IMP_ODBA_minus_farm_std	Imputed ODBA_minus_farm_std	0.67
5	IMP_pitch_minus_farm_std	Imputed pitch_minus_farm_std	0.53
6	M_LOG_farm_average_grazing	Imputation Indicator for LOG_farm_average_grazing	0.51
7	IMP_rumi_minus_farm_avg	Imputed rumi_minus_farm_avg	0.46
8	IMP_farm_average_rumination	Imputed farm_average_rumination	0.37
9	IMP_mob_std_grazing	Imputed mob_std_grazing	0.35
10	IMP_LOG_farm_average_roll_averag	Imputed: Transformed farm_average_roll_average	0.33
11	M_graze_minus_mob_med	Imputation Indicator for graze_minus_mob_med	0.31
12	IMP_pitch_std	Imputed pitch_std	0.30
13	IMP_mob_minus_farm_grave_med	Imputed mob_minus_farm_grave_med	0.29
14	IMP_pitch_average	Imputed pitch_average	0.29
15	IMP_farm_average_pitch_average	Imputed farm_average_pitch_average	0.17
16	IMP_farm_average_resting	Imputed farm_average_resting	0.14
17	IMP_roll_std	Imputed roll_std	0.15
18	IMP_roll_minus_farm_avg	Imputed roll_minus_farm_avg	0.12
19	IMP_LOG_mob_median_grazing	Imputed: Transformed mob_median_grazing	0.14
20	IMP_LOG_farm_average_odba_averag	Imputed: Transformed farm_average_odba_average	0.14
21	IMP_mob_minus_farm_rest_med	Imputed mob_minus_farm_rest_med	0.12
22	IMP_mob_std_resting	Imputed mob_std_resting	0.11
23	IMP_pitch_minus_farm_average	Imputed pitch_minus_farm_average	0.11
24	IMP_rest_minus_mob_med	Imputed rest_minus_mob_med	0.12
25	IMP_LOG_grazing	Imputed: Transformed grazing	0.12
26	IMP_LOG_farm_average_grazing	Imputed: Transformed farm_average_grazing	0.11
27	IMP_graze_minus_farm_avg	Imputed graze_minus_farm_avg	0.10
28	IMP_rumination	Imputed rumination	0.06
29	IMP_LOG_farm_average_pitch_std	Imputed: Transformed farm_average_pitch_std	0.04

Table 7: Variable Importance of Neural Network

Parameter	Estimate Absolute
BIAS_H13	2.05
BIAS_H11	1.35
H13_health_condition1	0.93
H11_health_condition1	0.85
H12_health_condition1	0.64
sickness_seasonsummer_H11	0.40
M_mob_std_resting0_H11	0.38
IMP_mob_minus_farm_grave_med_H13	0.35
IMP_graze_minus_farm_avg_H13	0.33
sickness_seasonsummer_H12	0.32
M_rest_minus_mob_std0_H12	0.30
IMP_mob_minus_farm_rum_med_H13	0.30
IMP_farm_average_resting_H11	0.27
M_roll_std0_H11	0.25
IMP_LOG_mob_median_grazing_H11	0.25
_DUP12	0.25
_DUP	0.25
M_farm_average_rumination0_H13	0.24
IMP_mob_std_grazing_H11	0.24
IMP_ODBA_minus_farm_average_H11	0.24
IMP_pitch_std_H12	0.24
M_mob_minus_farm_rest_med0_H12	0.23
sickness_seasonsummer_H13	0.23
M_rumi_minus_mob_med0_H12	0.23
IMP_rumi_minus_mob_std_H12	0.23
M_LOG_farm_average_grazing0_H12	0.23
M_pitch_average0_H13	0.23
M_rumi_minus_farm_avg0_H13	0.22
IMP_ODBA_minus_farm_std_H12	0.22
M_LOG_farm_average_grazing0_H13	0.21
IMP_pitch_std_H11	0.21
M_rumi_minus_mob_med0_H13	0.21
M_mob_std_grazing0_H12	0.21
IMP_pitch_minus_farm_average_H13	0.21
M_graze_minus_mob_std0_H12	0.21
IMP_LOG_odba_average_H13	0.21

Table 8: Variable Importance of Logistic Regression

Effect	Wald Chi-Square	Pr > ChiSq
IMP_LOG_farm_average_grazing	12.4106	0.0004
IMP_LOG_farm_average_odba_averag	233.9931	<.0001
IMP_LOG_farm_average_odba_std	153.3143	<.0001
IMP_LOG_farm_average_pitch_std	51.1283	<.0001
IMP_LOG_farm_average_roll_averag	231.1786	<.0001
IMP_LOG_farm_average_roll_std	169.274	<.0001
IMP_LOG_grazing	76.103	<.0001
IMP_LOG_mob_median_grazing	1127.4511	<.0001
IMP_LOG_mob_median_rumination	137.4665	<.0001
IMP_LOG_odba_average	331.2458	<.0001
IMP_LOG_odba_std	528.1768	<.0001
IMP_ODBA_minus_farm_average	1383.9322	<.0001
IMP_ODBA_minus_farm_std	271.0174	<.0001
IMP_farm_average_pitch_average	341.901	<.0001
IMP_farm_average_resting	34.1733	<.0001
IMP_farm_average_rumination	540.3904	<.0001
IMP_graze_minus_mob_med	5.2926	0.0214
IMP_graze_minus_mob_std	82.0769	<.0001
IMP_mob_minus_farm_grave_med	34.4001	<.0001
IMP_mob_minus_farm_rum_med	80.4379	<.0001
IMP_mob_std_grazing	746.9428	<.0001
IMP_mob_std_resting	438.0958	<.0001
IMP_mob_std_rumination	156.159	<.0001
IMP_pitch_minus_farm_average	898.7444	<.0001
IMP_pitch_minus_farm_std	103.1857	<.0001
IMP_pitch_std	5.4521	0.0195
IMP_rest_minus_mob_med	258.6968	<.0001
IMP_roll_average	41.833	<.0001
IMP_roll_minus_farm_avg	48.1536	<.0001
IMP_roll_minus_farm_std	46.5267	<.0001
IMP_roll_std	67.592	<.0001
IMP_rumi_minus_mob_std	1255.5007	<.0001
M_LOG_farm_average_grazing	39.8114	<.0001
M_graze_minus_mob_med	796.8967	<.0001
M_graze_minus_mob_std	630.6685	<.0001
sickness_season	3081.872	<.0001

Figure 7: SAS EM diagram of Halter project