Division of Computing Science and Mathematics
Faculty of Natural Sciences
University of Stirling

Application of Machine Learning Techniques to Animal Health Data: A Scoping Project

Discovering, Ranking and Visualizing Characteristics of BVD Infected Farms in Scotland

Frank Owusu Mensah

Dissertation submitted in partial fulfilment for the degree of
Master of Science in Big Data

September 2018

Abstract

Bovine Viral Diarrhoea (BVD), a cattle disease endemic in the United Kingdom and other parts of the world, has huge reproductive, health and economic impacts. In 2010, the Government embarked on a scheme to eliminate BVD from Scotland. As a result of this scheme, some farms are free from BVD. However, there are some farms that have never been free from the BVD virus. A third group of farms become BVD free for some time only to lose their BVD-free status after some time. This study was conducted to identify, rank and visualise, by the use machine learning techniques, the unique characteristics of Scotland's cattle farm that make them predispose the BVD exposure using data available in the Epidemiology Research Unit's database. The outcomes of this study present opportunities to use customised management programs and preventive management strategies to control the BVD disease. This could accelerate the eradication of BDV from Scotland.

The cross-industry standard process for data mining methodology was adopted to manage this project. An experimental approach in which 10 potential supervised learners and seven datasets were assessed to find which is best suited to achieve the objectives of the project was used. Recursive Feature Elimination with Cross-Validation was run on top of the best algorithm to ensure that the identification and ranking of the optimal risk factors do not suffer from the inter-correlations among the features. In addition, the relative importance of the potential risk factors was assessed and ranked. Probability calibration was run to ascertain whether the principal algorithm requires an Isotonic or Sigmoid calibration to deal with the effect that the bias due to uneven class in the response variable may have on model's performance. In order to improve of the model performance, a grid search was run to find the best hyper-parameters. Finally, three graphical models were produced and evaluated to find which of them could be interpreted easily by a non-data science professional.

The eXtreme Gradient Boost (XGBoost) model and the *Last Status* dataset proved to best suited to achieve the goals of this study. It was observed that the XGBoost model did not need either Isotonic or Sigmoid calibration to handle the effect of the bias that may be introduced by the uneven classes in the dataset. Eight of the 11 farm characteristics were identified and ranked as optimal risk factors by the main model. The surprise risk factor was presence of Pigs on a farm. In terms of relative importance, the number of calves under one-year old had the highest influence. Also, the study found that cattle farms in Scotland are highly open and interconnected. The average farm was found to be influenced by at least seven other farms a year. Again, over 31% of all cattle in 2017 were involved in unidirectional movement. Similarly, a farm in Scotland may be connected to as many as ten countries outside the United Kingdom in a year. Such a high level of 'openness' and interconnectivity increase the opportunity for the spread of the BVD virus. Furthermore, the presence of cattle kept as dairy was found to be the most deciding risk factor on a farm. Finally, the J48 graphics model was found more useful for explaining its decisions to a non-data science expert.

Based on the findings of this study, it is recommended that cattle movement policies be strengthened and rigorously enforced to protect farms from outside influence. Also, more attention be given to farms with dairy breeds and more calves under one-year old. Finally, further study be undertaken to explore why pigs were identified as an important risk factor.

Attestation

I understand the nature of plagiarism, and I am aware of the University's policy on this.

I certify that this dissertation reports original work by me during my University project except for the following:

The J48 Model Building procedures discussed in Section 3.5.1 was developed by me during the second semester in a Machine Learning Assignment of the Data Analytics module 2018, MSc Big Data, submitted to the Computing Science and Mathematics Division, University of Stirling.

The SQL code labelled '1.0 Herd Status.sql' in box 1.1 part B, chapter 1.4 was written by my industrial supervisor, Dr Aaron Reeves after my discussion with him.

The SQL code labelled '1.1 animal_count_2017.sql' in box 1.1 part B and discussed in chapter 1.4 was written by Dr Julie Stirling after my discussion with her, though I made a slight change.

The R code labelled 'bvd_status_count.R' in box 1.1 part B and discussed in chapter 1.4 was written by Dr Andrew Duncan after my discussion with him, though I made a slight change.

The Python code labelled '4.1 Main_ML_Template.py' in box 1.1 part B, 4 and discussed in chapter 1.4 follows and uses Scikit Learn packages and libraries (Pedregosa *et al*.). The calibration part follows Metzen *et al*. The eXtreme Gradient Boost portion (XGBoost) follows and uses XGBoost libraries. (https://xgboost.readthedocs.io/en/latest/index.html).

The Artificial Neural Network code (ann.py) follows and uses Keras libraries (https://keras.io/) which runs on top of TensorFlow.

The Exploratory Data Analysis uses Pandas, a python's external library.

Signature **Date September 7, 2018**

Acknowledgements

I acknowledge the help of God, the Almighty without whose providence I would not have reached this far. I would like to convey my heart-felt gratitude to the following persons and organisations whose contributions made this study a success:

Mrs Ellen A. Mensah, my wife, for her love and for caring for the children. My in-laws Messrs S. T. Aidoo and E. A. Aidoo for providing financial support for my stay in Inverness during the project period and Miss S. A. Aidoo for supporting my children. My parents (Mr. E.G. Asabi-Mensah and Mrs. M. A. Kyerewah) for their morally support.

Professor Carron Shankland, Professor of Computing Science, University of Stirling, my supervisor, for her excellent advice and suggestions, that made me think outside 'the Data Scientist box', that proved to be a landmark effort towards the success of my project and showing care for my situation. Dr. Ansgar Scherp, Associate Professor, University of Stirling, for accepting to be the second reader. Dr Simon Jones for providing timely information and providing all the academic materials needed for this project.

Dr. Aaron Reeves, Quantitative Epidemiologist, Epidemiology Research Unit, Scotland's Rural College (SRUC), Inverness, my industrial supervisor, for accepting me to work on his brain-child project and providing all the equipment and directions that made this project a success as well as reviewing this work. Dr. Jude Eze, Statistician, Epidemiology Research Unit, SRUC, Inverness for his suggestion, readiness to provide technical advice and reviewing this work. Ms. Barbara Gear (MRes, Dip.Eur.Hum.), Research Support Team, with the Future Farming System Research Group, SRUC, Inverness for volunteering to take the pain to proofread all the script. Her work and suggestions were helpful Dr Julie Stirling, Quantitative Scientist and Dr. Andrew J. Duncan for their support. All the staff of the Scotland's Rural College (SRUC) at Inverness, Scotland for the supporting and friendly atmosphere they provided for me during my stay in Inverness both within and outside office hours. Kajetan Stanski, PhD Student, Roslin Institute in Edinburgh and working for EPIC, for his good suggestions.

The Scottish Government for awarding me tuition Scholarship through the Data Lab Programme, The SRUC particularly the Epidemiology Research Unit for providing all the data and office equipment for this project. Scikit Learn (Pedregosa et al.) whose library I used in the machine learning part of the code. Lastly, I am grateful to Pandas, the python library I used for Exploratory Data Analysis.

Table of Contents

List of Table

List of Boxes

Box 0.1

List of Figures

1 Introduction

In Scotland, the Epidemiology Research Unit (ERU) at the Scotland's Rural College (SRUC) is responsible for *"improving and maintaining animal population health and welfare by integrating a range of science, including epidemiology, to develop our understanding of animal disease control"* (SRUC) [1]. To achieve this, the ERU has gathered a large number of diverse datasets ranging from individual animal records to environmental factors in a central database. These data come from a variety of resources including the Centre of Expertise on Animal Disease Outbreaks (EPIC) Data Repository and potentially hold a wealth of valuable insights that could be harnessed by the use of machine learning techniques.

In the past, the ERU has relied on their traditional data analytical techniques to derive knowledge from the data. However, modern data mining / machine learning techniques offer enhanced opportunities for identifying patterns hidden in such datasets. From these patterns prediction, classification or clustering models can be built, sometimes in real-time, to aid decision making. This can be used to benefit the livestock industries and the general population in several ways. For instance, by clustering cattle farms into groups, customised management programs could be applied to control diseases. Also, a classification model could predict which breed of cattle, which herd of cattle, or which regions of Scotland are more vulnerable to (for example) the Bovine Viral Diarrhoea disease. Furthermore, diseases risk factors could be identified and ranked with the aid of a classification model. The outcomes of such predictions could be inputs in preventive management strategies. All these would enhance Biosecurity in Scotland.

Modern data science and big data techniques have been successfully applied in many areas of human life (McAfee and Brynjolfsson) [2]. Their application in veterinary epidemiology could significantly enhance our capability to see and respond to patterns in trends that affect animal health and populations and to devise appropriate interventions (VanderWaal *et a*l.) [3]. There is, therefore, great potential in the application of data science on these relatively unexplored (in machine learning terms) datasets.

Part of this exercise was to take a preliminary view of a variety of datasets, then choose an animal health related problem / project that might be address with an appropriate machine learning technique in order to demonstrate, as a proof of concept, the potential of Data Science techniques to veterinary epidemiology. The success of this project will help inform the ERU of potential future research directions in applied data science for animal health. In consultation with experts at the ERU, the *"discovering, ranking and visualising risk factors of farms that are not negative for Bovine Viral Diarrhoea (BVD)"* was identified as the most promising project because of the following factors:

1. The occurrence of BVD represents a real problem whose solution will have immediate and far reaching impacts;
2. Data regarding BVD and the population in which it is found is available;
3. The study is reproducible with other data resources; and
4. The project has wider scope, in that the approaches used are applicable to other questions in animal disease.

First, BVD affects the health, reproduction and population of cattle (Fray et al.) [4]. This leads to huge economic losses. BVD is also a threat to the biosecurity of Scotland (Vets' Guide) [5]. In recognition of these adverse impacts the Government has embarked on a programme to eradicate BVD from Scotland. The outcomes of this study (identified risk factors and predicted vulnerable herds) present opportunities to use customised management programs and preventive management strategies to control the BVD disease. This could accelerate the eradication of BDV from Scotland and thereby improving the general biosecurity. This would mean that not only animals / herds and farmers will benefit from this project but the wider cattle industry and the general population in Scotland also benefit. Also, the variety of data available in the ERU database present a potential wealth of insights that could be exploited to solve this and future projects. Furthermore, the machine learning algorithms, the solution strategy/procedure used to solve the problem of this project can be used as replicable templates tailored to solve other veterinary or epidemiological problems (The only caution that should be is to ensure that the dataset in converted to matrix form before presenting to the modelling algorithm). Finally, the scope of the project (all cattle herds in Scotland, potentially over 10,000 farms and hundreds of thousands of animals) implies a far reaching impact.

1.1 Background and Context

Bovine Viral Diarrhoea (BVD) is a cattle disease endemic in UK and other parts of the world (EPIC) [6], [5] and (Scottish Government) [7]. It has huge reproductive [4], health [6] and economic impacts. In 2010, the Government embarked on a scheme to eliminate BVD from Scotland [6] and [7]. As a result of this scheme, some farms are now free from BVD. However, there are some farms that are not yet free from the virus. There is yet a third group of farms which become BVD free for some time only to subsequently lose their BVD-free status. Are there some peculiar characteristics of farms that are not yet free from BVD and those whose BVD status keep changing? Can machine learning techniques be employed to discover the hidden characteristics, if any, of these farms? Can these techniques help to predict vulnerable herds? These are the questions whose solution this study seeks to explore.

1.2 Scope and Objectives

The aim of this project is to use machine learning techniques to discover, rank and visualise the *hidden* characteristics or risk factors of cattle farms that are not yet free from BVD disease in Scotland using data available in the ERU's database. To achieve this, the following specific objectives were set up:

1. To explore and extract relevant data from the ERU's database using SQL and determine which machine learning types are applicable
2. To identify or construct features / variables that are potential BVD risk factors and create dataset(s) for the analysis
3. To select some candidate machine learning models and evaluate their performance on the datasets
4. To build a machine learning model to identify, rank and visualise the peculiar characteristics of BVD infected farms

Hypothesis: In management and other applications such as that of this project, it is often required or desired to be able to explain why or how certain decisions were made. This means that one should be able to explain / interpret the decision given by an automated decision making system / model [18]. But different models present different levels of readability / interpretability to non-technical persons. Therefore, it hypothesised that:

> ***some models are more useful for explaining their decisions to a non-data scientist than other***.

In this study, three visual models will be built to test this hypothesis. These models will be presented to non-data science experts and their response / reaction (though subjective) will be used as a measure of a model's level of interpretability. These measures will then be used to ascertain whether or not all the models have the same usefulness in explaining their decisions to non-data science experts.

1.3 Method Employed

The cross-industry standard process for data mining methodology was adopted to manage this project (chapter 3).

1.4 Achievements

This section provides the major achievements, in relation to the objectives, of this study.

Objective 1: Among the extracted data were herd status in terms of BVD, other species and cattle counts (box 1.2, figure 2 and table 3.1). These were used to construct potential inputs (risk factors) to the Machine Learning process.

Objective 2: Nine potential response variables were created from the herd status. Ten potential risk factors were created from cattle number, breed purposes etc. In all, three groups of datasets were created (box 1.2, figure 2 and table 3.1). The main group is the machine learning set which comprises seven datasets These seven datasets differ only by their response variables (Box 1, figure 2 and table 3.1). The machine learning algorithms have been tested on the seven datasets and the best three were selected.

Objective 3: Based on the performance of the 10 candidate Machine Learning algorithms on the seven datasets, the XGBoost model was chosen as the main technique to achieve the project objectives. Based on their responses to the models, the Last Status dataset was found to be the best.

Objective 4: Of the 11 final input farm characteristics, eight were selected and ranked optimal risk factors by the XGBoost model. This included one least expected risk factor (pigs) whose role in BVD will require further investigation to confirm.

Three different visual models were built. Of these, the J48 model was tested to be more easily understood by non-data science experts.

As by-products of this study were reproducible sets of algorithms (in particular the machine learning set, box 1.1) and the solution strategies / procedures that were used to solve the problems of this project. These algorithms and procedures can be used as templates that can be replicated or tailored on different datasets to solve similar or different problems in different fields including veterinary or epidemiological problems.

Box 1.1 Set of deliverable datasets and algorithms

Part A: Three groups of Datasets		
1. Building /Extracted set	**2. Processing / Constructed set**	**3. Machine learning set**
1.0 Herd Status.csv 1.1 animal_count_2017.csv 1.2 nw Farms loc.csv 1.3 total impt 17.csv 1.4 dateofbirths_17.csv 1.5 In-degree off to on_17.csv 1.5 off to on_17=in- 1.6.0 breed.csv 2.0 species.csv 2.1_Pig Farms to add.csv 2.3 SheepGoat 13 to Add.csv 2.3_SheepGoatI.csv degree_2017.csv SheepGoatScot13.csv Pig.csv 1.6 breed impt loos.csv	in-degree_2017.csv Herd Status 11-18.csv farm_breed_purpose.csv confirmative decision.csv farm_breed_purpose.csv total impt 15 17.csv total impt 11 17.csv import_13-17.csv Breed_Size4_Class.csv Breed_Size7_Class.csv N_Breed_Farm_ClassMixed.csv Breed_Farm_ClassMixed.csv Breed_Farm_FlatMixed.csv	3.0 all2.csv 3.1 all_herdStatus.csv 3.2 all_alNegt.csv 3.2b all_alNegt_Cat.csv 3.3 all_alNotNeg.csv 3.3b all_alNotNeg_cat.csv 3.4 all_change.csv 3.5 all_badchange.csv 3.6 all_goodChange.csv 3.7 all_multiChange.csv all.csv all_herdStatus1.csv all_alNeg.csv all_alNot.csv all_mc_breed.csv all_hs.csv all1.csv
Part B: Four different sets Algorithms / Code		
1. Extraction set	**2. Pre-processing set**	**3. Herd Classification set**
1.0 Herd Status.sql (Reeves) 1.1 animal_count_2017.sql (Stirling) 1.2 nw Farms loc.sql 1.3 total impt 17.sql 1.4 dateofbirths_17.sql 1.5 In-degree off to on_17.sql 1.6.0 breed.sql 1.6 breed impt loos.sql 2.1_Pig Farms.sql 2.3_SheepGoatI.sql	Herd status change.py All_Status_Change.py bvd_status_count.R (Duncan) yearly_Status_Change.py All_Status_Change 11-17.py Status_Change13-17.py StatusChange.py EDA.py Inner Joint.py nw total animals.py (joining) in-degre_off to on_17.py Calves_+Import.py	FarmSize_Class.py Breed Classification_7.py Breed.py 2.3 Class_SheepGoat.py import_move.py 2.1 pig.py Calves_+Import.py 3.0 Cattle Brees Class.py native_foreign.py 2.3 S G.py argsort.py
4. Machine learning set		
4.0 PractStepsML_SettingEnv_DataPreprocessig.py 4.1 Main Machine Learning algorithm: Main_ML_Template.py 4.2.1 Data Pre-processing: Final_data.py 4.2.2 Model Comparison: model _selection.py 4.2.2.2 Generating and plotting the Area Under ROC Curve 4.2.4 Recursive Feature Elimination with Cross-Validation: RFE_CV.py 4.2.5 Probability Calibration and Reliability Curve: Probability calibration.py 4.3 Decision Trees 4.4 Ranking Features with Extra Trees 4.5 eXtreme Gradient Boost: XGBoost fx.py 4.6 Artificial Neural Network: ann.py, ann.py 4.7 J48 Model (Weka)		

1.5 Overview of Dissertation

This dissertation is organised into six chapters / sections. Chapter one introduces the project and describes the background and context as well as the scope and objectives of this study. Also, the method employed and the achievements of the study are described.

Chapter two covers the literature review in two different contexts. The first part is dedicated to domain knowledge of the Bovine Viral Diarrhoea disease with emphasis on potential herd-level risk factors. The second section seeks to provide answers to issues relating to the application of machine learning techniques employed in his study. Why were the preferred techniques chosen? In what areas have they been applied? Which of their strengths were exploited? Which weaknesses were overcome?

Chapter three gives details about how the study was executed. Data extraction, data exploration, data processing and modelling are some of the processes covered under this section. Chapter four is dedicated to the evaluation of the principal classifier and the selected datasets. The main findings together with their corresponding discussions are presented in chapter five. Chapter six concludes the project with summary and recommendations.

2 Literature Review

This section reviews what is currently known about Bovine Viral Diarrhoea (BVD), herd characteristics that are potential risk factors for BVD infection (section 2.1) and machine learning techniques (section 2.2).

2.1 Farm Characteristics

Covered under this section are Bovine Viral Diarrhoea (BVD), cattle population parameters and breed types, other species as well as farm interconnectivity metrics.

2.1.1 Bovine Viral Diarrhoea and Herd Status

Bovine Viral Diarrhoea (BVD) is a cattle disease endemic in the United Kingdom and other parts of the world [5] - [7]. It has huge reproductive [4], health and economic impacts [6]. In 2010, the Government embarked on a scheme to eliminate BVD from Scotland [5] – [7]. As a result of this scheme, some farms are free from BVD. However, there are some farms that have never been free from the BVD virus. There is yet a third group of farms which become BVD free for some time only to lose their BVD-free status after some time.

If any cattle on a farm test positive, the farm /herd is assigned BVD Not Negative status. If no cattle on a farm test positive, the farm/herd is assigned BVD Negative status. Farms with Not Negative status are obliged to take certain actions to achieve Negative status (BVD Order 2013) [8], [6], [7].

Among the regulations under the eradication policy are controls on cattle movement, particularly from BVD Not Negative herds [6] – [8]. The aim is to check the spread of the virus and to protect BVD Negative status farms. Another important regulation requires the testing of all new-born calves. This is because

> “Calves that survive infection during the first trimester of pregnancy are born with a persistent and lifelong infection. These persistently infected (PI) animals represent between 1.0% and 2.0% of the cattle population and continuously shed infectious virus [4]”.

For this reason, “persistently infected (PI) cattle are by far the most important method of transmission of the disease” [6] – [8]. Also, ‘a PI mother can only give birth to a PI calf and therefore any calf born to a PI cow will be assumed to be PI’ [5]. Even though some PI live to reproductive age, and some may not be identified, the majority of them do not live to their second year [5]. Hence the interest in this study in the number of calves under one year old.

2.1.2 Cattle Breed Purpose and Population Parameters

It has been observed in Scotland that dairy cattle are more exposed to BVDV than cattle kept for other purposes. Whereas the Not Negative herd rate stood at 12% for Beef herds, the rate for Dairy herds has reduced from 50% to 39% [5]. Also, since the PIs are the most important cattle group in the spread of BVDV [6] – [8], it is expected that herds with more calves are more at risk. In addition to this, since a farm loses its Negative status even if only one animal is tested BVD Not Negative [5], [8], larger farms (in terms of herd size) are more at risk of BVD exposure to BVD than with smaller herd size.

2.1.3 Other Species

An area that perhaps has not been sufficiently explored (to date) is the role of other species. It is known that the BVD virus (BVDV) can infect pigs, sheep goats and deer [5] and (Tao *et al.*) [9]. Natural infection of pigs with BVD was detected as far back as 1976 in the Netherlands

(Terpstra and Wensvoort) [10] and (Wensvoort and Terpstra) [11]. Almeida *et a*l (2017) [12] observed that pigs are susceptible to the BVD virus “under natural conditions” even though the “infection is practically unknown in the realm of pig farming”. But, [9] had previously noticed that “the prevalence of BVDV infection in pig herds has substantially increased in the last several years, causing increased economic losses to the global pig breeding industry” and went on to discuss the

> “historical overview, clinical signs, pathology, source of infection, genetic characteristics, impacts of porcine BVDV infection for diagnosis of classical swine fever virus (CSFV), differentiation of infection with CSFV and BVDV, and future prospects of porcine BVDV infection.”

Also, there is confirmation of an interaction between sheep and cattle in Scotland

> “the pestivirus of sheep, Border Disease Virus (BDV), can infect cattle and result in the generation of persistently infected cattle. In Scotland the close contact between sheep and cattle that occurs on many farms creates the opportunity for BVDV to infect sheep and BDV to infect cattle and while the frequency with which this occurs is unknown, it has been considered unlikely to be of significance in relation to national control” [5].

The presence of other species may therefore be a risk factor.

2.1.4 Farm Interconnectivity

The aim of introducing interconnectivity measures as features in this study was to ascertain whether or not a farm operates an open or closed policy in relation to contacts with other farms as higher levels of interconnectivity present a risk to its BVD status. Interconnectivity has both advantages and disadvantages. In epidemiological sense, the more connected a farm is the more vulnerable to infectious disease it can be (Stattner and Vidot) [13]. Interconnectivity “expresses the susceptibility of a node (farm) through its contacts with others.” (Barabasi) [14]. There are many metrics of network interconnectivity and thus vulnerability. Some of these are In-degree Connectivity, Path, Betweenness, Clustering Coefficient and Eigenvector Centrality. In-degree connectivity of a farm can be defined as the number of connections incident on that farm (Newman, 2003) [15] and (Newman, 2010) [16]. In this study, three aspects of in-degree connectivity are considered, namely (1) the total number of cattle moved to a farm, (2) the number of farms from which a farm has received cattle and (3) the number of non-UK countries from which a farm has imported cattle.

2.2 Machine Learning and other Techniques

What is machine learning and what types of learnings are available? Why were the techniques employed to solve the problem of this study chosen? How have they been used by others? Which of their strengths were exploited? Which weaknesses were overcome? Answers to these are provided by the following sections (sections 2.2.1 to 2.2.9) as much as applicable. Subjects covered include model selection with emphasis on eXtreme Gradient Boost Classifier and interpretability versus accuracy. Furthermore, feature importance, selection and representation, calibration, model optimisation and performance evaluation are also covered.

2.2.1 Machine Learning Background

As data continues to grow, the prospect to gain insight from data for informed management decision-making increases. This is because there are ‘hidden’ patterns inherent in nearly every dataset. Machine Learning (ML) is one of the techniques that offers the opportunity to mine pattern from data. Machine Learning can be defined in simple terms as the process of using data to train or give a machine / computer the ability to **learn** to do a task (Witten *et al.*) [17]

and (Swingler)]18]. The machine is given data and it learns the patterns in the data to perform the task we assign it. If the input data has a desired outcome (output variable), the learning is supervised because the machine learns the rules that map the input variables to the output variable [17], [18]. On the other hand, if there is no output variable in the dataset, the learning is unsupervised [17], [18]. There is also reinforcement learning where the machine is rewarded for doing what is desired and punished for doing what is not desired. Since the datasets have output variable, we will have the luxury of doing supervised learning.

In this supervised learning case, the BVD herd status group of variables (figure 2 and table 3.1) are the output (also referred to as target / response/ dependent/ outcome) variables. All the other variables (farm characteristics) are potential BVD risk factors (figure 2 and table 3.1). these group of variables may be referred to as features / predictors / independent / input variables. If the machine finds any pattern or rule that maps or links any/some of the farm characteristics to herd status classes, it means we can use the farm characteristics to predict the herd status. And if the prediction accuracy is good enough, then there is high chance that those farm characteristics are BVD risk factors. This will be confirmed by evaluating the relative importance of the farm characteristics / risk factors to the prediction. The larger the relative importance of the farm characteristic to the prediction of the herd status, the greater the risk of the herd becoming infected.

2.2.2 Model Selection

There are many (supervised) ML techniques available. However, the choice of a suitable technique (i.e. providing a fruitful result) for a specific project depends on knowledge of how that particular model works, knowledge of dataset, goal of the project, resources available and to some extent trial and error (luck). The nature of the dataset will determine the range of potential ML techniques [17], [18] that can be applied to the dataset. Depending on whether the project goal prioritises accuracy or interpretation, the number of potential models can be further reduced. Some algorithms (especially Artificial Neural Networks) are computing intensive and therefore require specialised processing units. Taking all these items into consideration, the best model for a given project is usually not obvious [17]. Hence, the recommendation that the final choice be made on an experimental basis [17], [18] where a number of potential models are tested on the dataset and their performance evaluated to ascertain which model is best suited to the dataset and project goal.

The task of this project is classification. We want to predict which cattle herds are more vulnerable to BVD infection and to identity and rank the associated risk factors. Therefore, interpretability will be intensified at the expense of accuracy, if need be. The following paragraphs briefly explain the selection of candidate classification models.

Dummy and Decision Tree Classifiers: These simple classifiers Whilst Dummy is purposely designed to act as a 'baseline to compare with other classifiers' (Pedregosa *et al.*) [19], Decision Tree can produce a pictorial view of its decision that provide us with the opportunity to interpret the results of its outputs.

K-nearest-neighbours adapts itself well to both data with linearly separable boundaries and those with non-linearly separable boundaries (Hastie *et al.*) [20]. Since it is not yet known whether the unique patterns within the datasets exhibit linearly or non- linearly, this model could be the obvious choice. Also, if it becomes relevant to explore unsupervised learning such as clustering the farms into groups, in addition to the current supervised learning, K-nearest-neighbours would remain an option since it can be used for both types of learning.

Logistic Regression or Naive Bayes is appropriate for ranking the outputs by their probabilities. Logistic Regression is suitable to problems with dichotomous outcome (‘yes’ or ‘no’, ‘infected’ or ‘not infected’) [19 and 18]. Logistic Regression predicts the probabilities of the response variable and pushes all probabilities below certain threshold (usually 0.5) to have a ‘zero’ probability and those above the threshold to have a ‘one’ probability. Since the response variables in the datasets have binary classes, Logistic Regression is expected to produce predictions of higher accuracies. Also, Logistic Regression has been used successfully as a baseline for probability calibration in reliability tests for comparison of classifiers (see chapter 2.2.4) (Dal Pozzolo *et al.*) [21] and (Metzen *et al.*) [22]. This quality will be exploited in this study. For similar reasons, Gaussian Naive Bayes classifier was selected as in addition to its probability qualities, it has been proven to perform well when embedded in a feature selection model (chapter 2.2.3.1) (Granitto *et al.*) [23]. Furthermore, Naive Bayes is a simple but an accurate classifier that works faster [17].

Extra Trees Classifier (ETC) and Random Forest Classifier (RF): ETC and RF are both (ensemble) meta estimators that find the average of some decision trees to ‘improve the predictive accuracy and control over-fitting’. Whereas RF takes several decision tree classifiers, ETC picks randomized decision trees’ (a.k.a. extra-trees), both fit them on various sub-samples of the dataset. [19]. Furthermore, whilst Extra Trees Classifier has been more popular due to its ability to identify and rank important features, Random Forest has been used more in feature calibration that allows comparison of models (Niculescu-Mizil and Caruana) [24]. Although RF gives high accuracy (which was exploited), it is less interpretable. The feature importance identification and ranking qualities of ETC was exploited in this study. Both ETC and RF were used in calibration procedure in this project (see chapter 2.2.4).

Support vector machines (SVM): SVM is a versatile model. By customising the Kernel functions, it can adapt to a wide range of datasets being linearly or non-linearly separable. It has been used to ‘find the optimal separating hyperplane using an SVC for classes that are unbalanced’ [19] which is the situation for this project. However, it is susceptible to overfitting if a regularisation term is not imposed on it.

Artificial Neural Network (Deep Network) and eXtreme Gradient Boost Classifier (XGBoost): these two models are the most complex and often the most accurate. All other things being equal, simple models are usually preferred to complex ones because simple models are better able to generalise and avoid over-fitting than their complex counterparts [17], [18]. Also, simple models are fast since they are less computationally intensive. To select a complex model over a simple one, one must satisfactorily justify one’s decision. All things being equal, one would usually choose Artificial Neural Network (ANN) if the goal of the project requires no interpretation but a higher accuracy performance (such as in self-driven vehicle) and there is evidence that other simpler models are not able to achieve the same [17]. The reason is that, despite their high performance, ANN models have earned the nickname ‘black boxes’ [17 and 18] since they offer no explanation to their decisions. Therefore, the only motivations for including ANN in this study are, firstly, to be able to measure how much accuracy will be lost to the interpretability goal and secondly, to serve as a performance yardstick for comparison with the eXtreme Gradient Boost (XGBoost) model. The XGBoost classifier is an embodiment of high (accuracy) performance, interpretability, fast execution and other qualities (XGBoost) [25] (discussed section 2.2.1.1, next page). The XGBoost model has a good potential to be a single classifier that can help to achieve all the objectives of this project - to discover, rank and visualise the hidden characteristics of cattle farms that are not yet free from BVD.

2.2.2.1 Extreme Gradient Boost Classifier

The eXtreme Gradient Boosting (XGBoost) is a relatively new algorithm that has attracted much attention due to its combination of many desirable qualities such as high performance, fast computational speed and interpretability (figure 1). It comprises trees which are individually weak learners but collectively form strong ensemble models [25]. It presents features as nodes (tree). Its final decision values are pushed into leaves, each with corresponding scores. These scores present "richer interpretations that go beyond classification" [25] in that the higher the value (in the leaf) of a farm characteristic, the greater the risk of BVD infection it presents to the herd.

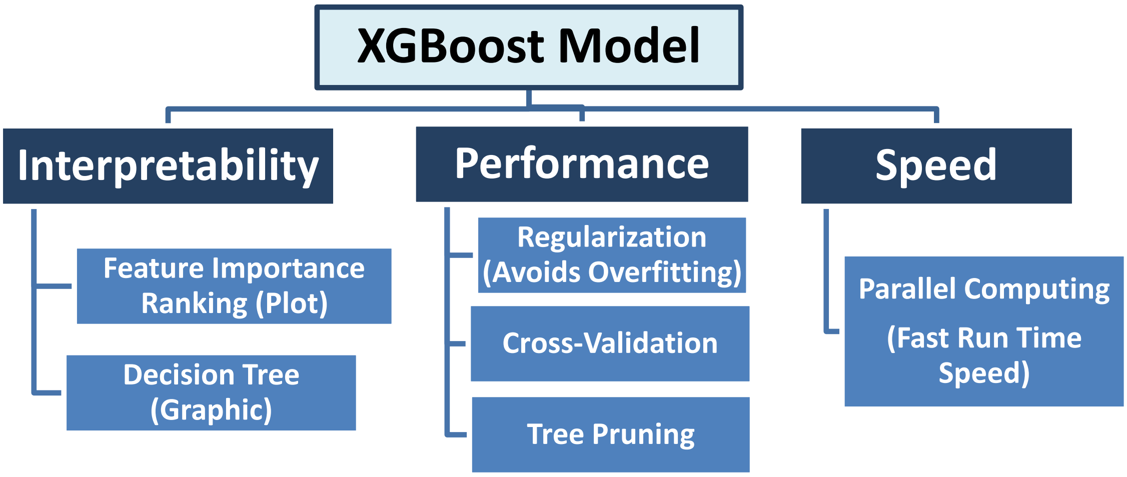


Figure 1. ***Desirable features of eXtreme Gradient Boost model exploited in this project.***

Some of the powerful features of XGBoost are [25]:

1. Regularization: XGBoost has a regularization term which checks overfitting in linear and tree-based models by controlling a model's simplicity and predictability (i.e. bias-variance trade-off);
2. Tree Pruning: It also has a non-greedy tree pruning mechanism which continues pruning even if there is no positive gain;
3. Cross-Validation (CV): Due to its inbuilt CV, its iterations get to the global optimum. Therefore, its results are often reliable and do not necessarily need Grid-search nor hyper-parameter tuning to be optimal, and
4. Parallel Computing: even on a single machine, it parallelises its computation by making efficient use of all available cores and thereby reducing runtime.

2.2.2.2 Interpretability-Accuracy Trade-off

Simple models such as Single Decision Trees can present their outputs in a visually interpretable graphic format [20]. On the other hand, complex models such as Artificial Neural Network, generally give high accuracy but their outputs are more difficult to interpret. Depending on the purpose of the machine learning project, one of these is usually optimised at the expense of the other (García *et al.*) [26] and (Hisao and Yusuke) [27]. For example, in management and other applications such as that of this project, it is often required or desired to be able to explain why or how certain decisions were made. This means that one should be able to explain / interpret the decision given by an automated decision making system / model [18]. One of the desirable advantages of XGBoost classifier is that it combines these two features of interpretability and accuracy [25].

2.2.3 Feature Importance, Selection and Representation

Input features often contribute unequally to the output variable. Whereas some are irrelevant and redundant, others have great influence [20]. Hence the need to select only the important features to improve performance and runtime, measure and rank the importance of farm features (section 2.2.2.1) and, if appropriate, visualise their relative importance (section 2.2.2.2). Whereas recursive feature elimination techniques seek to assess variables on their individual merits, relative importance algorithms judge them on their influence in comparison with others [17]. Another advantage of XGBoost is that it offers an opportunity to extract a feature importance score [25].

2.2.3.1 Recursive Feature Elimination

For every dataset, there exists an optimal subset of features that lead to better performance, generalization and faster convergence [18] and (Abe) [28]. Feature selection mechanisms purge the dataset of redundant and meaningless features. This approach can be used to identify important farm risk factors. Recursive Feature Elimination (RFE) is a reliable tool to identify important risk factors. In addition, the RFE meta-algorithm can be used to rank BVD risk factors [28]. Furthermore, an RFE ensures that ranking of the importance is not sensitive to inter-correlation between the input variables especially when it is run on top of the main algorithm.

The Recursive Feature Elimination is a repetitive means of feature ranking according to certain degrees of their relevance (Guyon *et al.*) [29]. Granitto et al (2006) [23] successfully used Support Vector Machines based RFE on Random Forest. It was however, observed that feature selection can be unstable. To overcome this, it was suggested that the algorithm be run several times and the features with the highest probability of occurrences selected. This weakness is surmounted in this study by using Recursive Feature Elimination with Cross Validation (RFECV) which runs ‘k’ times and selects the optimal features [19] on XGBoost - the principal model. Further, this boosted RFE is run over the three datasets to obtain the best features based on weighted scores. For comparison, RFECV is run again on Extra Trees Classifier. – also a feature selection model.

2.2.3.2 Relative Importance Representation

XGBoost can present its feature selection results in both bar chart (as does the Extra Trees model) and decision tree graphic (as does the Decision Tree model). The decision tree comprises a node (feature) that splits into other nodes each of which in turn splits until a decision node (leaf) is reached. The first node is the most important feature according to the criteria used for splitting, which seeks to reduce the cost function. This could be the amount of information gained (or entropy) [18]. It could also be the relative importance in terms of

> “maximal estimated improvement in squared error risk over that for a constant fit over the entire region. The squared relative importance of a variable is the sum of such squared improvements over all internal nodes for which it was chosen as the splitting variable” [20].

The comparative importance metrics can be “based on the number of times a variable is selected for splitting, weighted by the squared improvement to the model as a result of each split, and averaged over all trees” (Friedman and Meulman) [30]. Thus, the more a feature is used to split, the higher its F Score and hence its importance [30] and (Elith *et al.*) [31]. A weighted importance is calculated for all the features, ranked and plotted on a bar graph whose lengths are proportionate to a feature’s importance, arranged in descending order of importance. This shows the relative importance of each feature (Chen and Guestrin) [32]. This value is scaled to sum to one, such that the larger values imply greater feature contribution to the response variable [30].

2.2.4 Probability Calibration and Reliability Test

Data with uneven class distribution often introduces bias that impact on the performance of a model [17], [18], [21]. There are many options to dealing with such datasets. For instance, the whole variable with imbalance classes can be discarded [18]. Usually, the imbalance can be corrected by either reducing the majority class or increasing the minority class by various techniques [17], [18], [21]. However, one of the fundamental postulations of machine learning is that the entity producing data for training, testing and future business use of a model will remain unchanged. This implies that the dataset used to train, test and even put the model into practical use are presumed to have the same distribution [17], [18], [20], [21]. In view of this, any attempt to balance /correct data with imbalanced classes will often breach this important assumption. One way to deal with imbalanced classes in a dataset that does not violate this assumption is to calibrate that dataset [21].

Dal Pozzolo *et al* (2015) [21] studied the effect of bias introduced by uneven class in datasets using Brier score as a loss measure, G-mean as predictive of accuracy and area under the ROC curve for ranking and observed that this bias influences the "classification accuracy and probability calibration" appreciably even though it did not influence the "ranking order returned by the posterior probability" [21]. Metzen J.H. *et al* (2015) [22], also used probability calibration with isotonic or sigmoid regression on Support Vector Machines (SVP), Gussian Naïve Bayes (GNB) with Logistic regression as baseline [22]. Among other scores, they used Brier score loss to evaluate the performance [19], [22].

In this project, probability calibration with Isotonic and Sigmoid Regressions on XGBoost, Random Forest and Extra Trees Classifiers was tested on our imbalanced datasets with Logistic Regression as the baseline to ascertain whether the principal classifier needs isotonic or sigmoid calibration in dealing with the imbalance classes in the datasets. Brier score loss was used as a performance measure.

Since the Brier score loss is a probabilistic function, it produces a predictive accuracy measure between zero and one. This score is the "mean square difference between the actual outcome and the predicted probability of the possible outcome" [19]. Therefore, the lower the score the more accurate will be the calibrated prediction [19].

2.2.5 Hyperparameter Optimisation

There are two types of parameters. Whereas one set is learnt by the model, the other set is not learnt by the model. The later, hyper-parameter must be fine-tuned by the programmer for optimal performance. Given the number of and the range of possible values each can take, the search space of all possible combinations of hyper-parameters can be huge and inexhaustible, especially for manual tuning [17], [18]. Many means of tuning hyper-parameters exist. One of the best and most successfully used means is Grid-Search with Cross-Validation - a heuristic algorithm that help locate the global optimal hyper-parameters [17], [18]. Hence the adoption, in this project, of the Scikit-learn module (model_selection) a package (GridSearchCV) which exhaustively and automatically searches for the best hyper-parameters values for a model from a range of inputs based on best Bias-Variance Trade-Off [19]. Usually, Grid-Search hyper-parameter optimisation is performed, first, to search for the best model [17], [19] and, second, to improve the performance of the selected model [17] - [19]. Due to the limited recourse availability (in terms of computing power – CPU), an alternative means of model selection will be used. In relation to improving a model's performance, Grid-Search hyper-parameter tuning will be performed to verify the claim that XGBoost (the potential main model) does not necessary require an external optimisation aid since it is believed to exploit its in-built regularisation term and cross-validation to achieve optimal results [25].

2.2.6 Performance and Evaluation

This section briefly explains and justifies some performance metrics used to assess the models and datasets in this study. As much as possible, the terminologies of this project are used to explain these metrics – confusion matrix, accuracy and ROC area, specificity versus sensitivity, precision, recall and f1-score.

2.2.6.1 Confusion Matrix

This is the fundamental classification metric. It shows the number of true and false predicted instances (farms) [18] in a matrix of actual and predicted classes (Negative or Not Negative]. All the other metrics discussed here are derived from the confusion matrix.

2.2.6.2 Accuracy and Receiver Operation Characteristic (ROC) Area

Accuracy expresses the ratio of number of correctly predicted farms to the total number of farms. This gives the basic 'quantitative' measure of an individual model's performance. Often models are judged by their classification accuracies. However, in order to compare the performance of two or more models, standardised and 'qualitative' measures are required. One of these is the area under the ROC curve (AUC) [17], [18]. It appraises a model independent of the error costs [18]. It plots, for example, the True Positive Rates on the vertical axis and the False Positive rates on the horizontal axis. Whilst a diagonal line denotes a random decision, any curve below towards the bottom-right signifies an inadmissibly ruthless decision and an opposite curve advancing up towards the top-left signifies a better decision [18]. The more the ROC curve rises fast up to the top-left and eventually smooths off the better. This implies that, the larger the area under the ROC curve, the better will be the overall performance of the classification model [17]. The ROC area is, therefore, a probability measure of a model that 'ranks an arbitrarily chosen positive test instance above an arbitrarily chosen negative one' [17]. In medical and disease diagnostic fields, the ROC curve is a "commonly used summary for assessing the trade-off between sensitivity and specificity" [20]. In this case, "it is a plot of the sensitivity versus specificity as we vary the parameters of a classification rule" [20].

2.2.6.3 Specificity versus Sensitivity

In this study (and in medical applications), it is important to know the strength of a model in terms of its ability to correctly identify BVD free farms (specificity) or BVD infected farms (sensitivity). This is because a model's ability to 'truly' classify BVD infected farm as presumably infected (Not Negative) in this use-case, is deemed more significant than its capacity to 'truly' predict a BVD free (Negative) herd. The reason for this is that the risk of a mis-classification of an infected farm is higher (disease spread unchecked) than that of misclassifying an uninfected herd (spread of disease is checked even though there is no disease). Whereas specificity measures the proportion of the 'truly' BVD free (Negative) farms that were correctly classified as such by the model, sensitivity measures the proportion of the presumably BVD infected (Not Negative) farms that were rightly predicted as such by the model [17], [18], [20]. In most instances, recall is used as sensitivity. Again, it is interesting to know which dataset lends itself to either specificity or sensitivity.

In relation to these concepts there is an important assumption – a 'closed-world assumption' [17] which stipulates that "if a particular farm is not in class 'Yes', then it must be in class 'No' [17]. For two datasets (*Always Not Negative* and *Always Negative*), this assumption cannot be supported. For instance, for the *Always Not Negative* dataset, if a farm is not classified 'Yes' (infected), it does not necessarily mean it belongs to 'Negative' class (free from BVD). A 'No' class, in this case, only tells that the farm has ever had its BVD status changed. Therefore, I ascribe *Assumed-Specificity* to a 'No' class. Similarly, for the *Always Negative* dataset, if a farm is not classified 'Yes' (free), it does not necessary means it belongs to 'Negative' class (infected). A

‘No’ class, in this case, only shows that a farm has ever had its BVD status changed. Therefore, an *Assumed-Sensitivity* is assigned to a ‘No’ class. However, the *Last Status* dataset follows a ‘closed-world assumption’ [17], i.e. if a farm is not in the ‘Not Negative’ class, then it is in the ‘Negative’ class.

2.2.6.4 Precision-Recall Trade-off: F1-score

Although criticised in some applications [33], f1-score is a better performance measure for datasets with imbalanced classes [34]. F1 score finds a harmony between precision and recall and it is frequently used in dichotomous classifications [34]. While recall is sometimes used as sensitivity, precision is used as a positive predictive value (PPV). Precision is the proportion of positive classifications which are actually positive [17], [18], [20].

3 Methodology

This section covers the phases of a machine learning project. The Cross Industry Standard Process for Data Mining (CRISP-DM) was used to run this project. It consists of six repeating stages that revolve around data. These are business understanding (chapter 1), data literacy (chapter 3.1) and data preparation (chapter 3.2). The rest are modelling (chapter 3.3), evaluation and deployment (chapter 3.4 and 4). First, all platforms (Scikit-learn, TensorFlow, Weka, Python 3, R, pgAdmin II) together with required libraries and packages were installed and the analysis environment was set up.

3.1 Data Literacy

All data for this project were available in the Centre of Expertise on Animal Disease Outbreaks (EPIC) Data Repository. Required data were extracted using PostgreSQL and saved / exported in a *comma separated values* (csv) format for (but not limited to) the following reasons [17]:

1. Almost all applications and programming languages can read, process and write a csv file;
2. Can differential between numeric and categorical data, making the intended data type safe;
3. Compact and fast to write since header is written only once; and
4. Most common way of importing and exporting data to databases.

To allow easy manipulation and thus processing, these csv data files were loaded into the analysis environment in a Pandas *DataFrame* form, a '2-dimensional labeled data structure' in which the columns may have different data types Pandas [35], as the case of this study's data.

Most machine learning algorithms accept only (or at least work well with) matrix data [20] of numeric type. For this reason, the *Dataframe* object must be converted to matrix data. This is archived by converting all nominal data to numeric data. This is so important that may algorithms have their own in-built conversion mechanism (should the programme forgets). The data were presented to the Machine Learning algorithms in a matrix form. This means all the data must the same data type (Buttrey) [36]. For future applications, any 2-dimensional data structure can be used, but care must be taken when presenting the data to the Machine Learning algorithm. All data must be of numeric type since a matrix accepts only one data type [36]. Also, care must be taking to avoid a dummy variable trap (box 3.1).

Box 1.1 and table 3.1 show the type of data extracted from the database or constructed after extraction. The base dataset comprises cattle farms / herds in Scotland in the year 2017. It should be noted that the other species data are not as current as that of cattle. Therefore, the number of sheep, goats and pigs on a farm in 2017 are likely to be different from those available for this study. However, there is no reason to assume that the general distribution of these species on the cattle farms has changed appreciably. Therefore, the impact that this disparity in date (the sheep and goat dataset was dated 2013) will have on the analysis and results of this project, if any, is believed / assumed to be very insignificant. Initially, there were 10,712 unique cattle farms. However, after data processing, each of the final datasets consisted of 8,519 unique farms.

3.2 Data preparation

This section consists of two phases - Exploration Data Analysis (chapter 3.2.1) and Data Preprocessing (chapter 3.2.2). This is the most important and time-consuming stage of machine learning. After cleaning each individual extracted or constructed datasets (box 1.1 and figure 2), they were merged (inner or outer joint) to form consolidated dataset ('all2.csv').

3.2.1 Exploratory Data Analysis

This initial analysis was conducted to understand the consolidated dataset ('all2.csv'. table 3.1) using Python 3 Libraries such as Pandas, Numpy, Matplotlib and Seaborn.

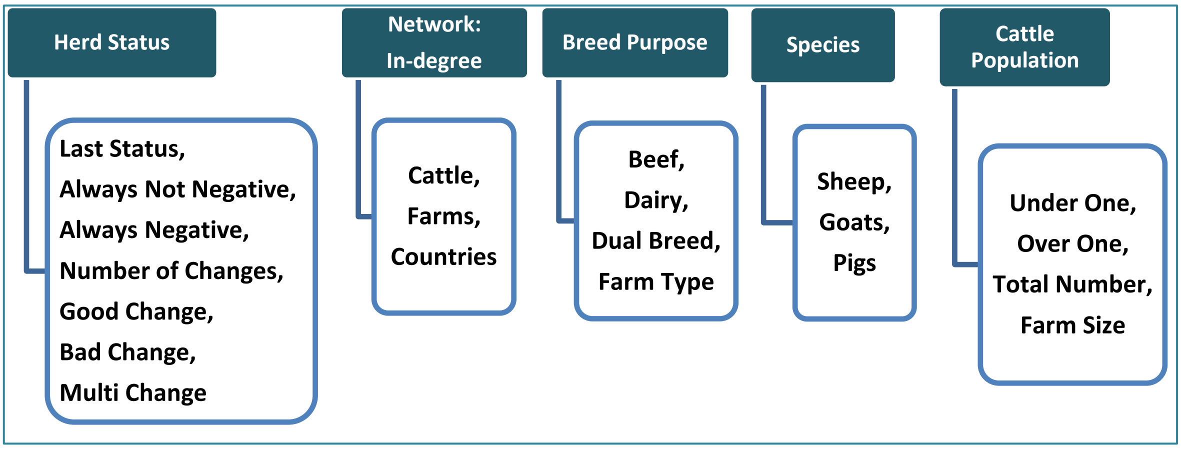


Figure 2. ***Groups of output variables (Herd Status) and input variables (Network metrics, Breed Purpose, Species and Cattle Population) created or extracted for the project.***

Data were visualised to detect data types, invalid data, duplicate records, anomalies (outliers or inliers, missing data, coding inconsistencies, typing errors), correlations, distribution patterns (majority or minority group), distribution types (flat and wide, normal), trends and others. Also, descriptive statistics was performed to reveal means, standard deviation, maximum and minimum values, etc. The variables are grouped into five categories, namely Head Status, Network Metrics, Breed Purpose, Species and Cattle Population (figure 2).

After the data processing, seven datasets (Last Status, Always Not Negative, Always Negative, Number of Changes, Good Change, Bad Change and Multi Change) were formed from the consolidated dataset ('all2.csv'). These seven datasets were used for the modelling phase.

Table 3.1 Description of variables constructed or extracted for the machine learning project showing the name of the variable, data type, unique value or range as well as explanation of the variable.

Variable Name	**Data Type**	**Distinct Values / Values Range**	**Explanation**
cph	numeric	8519	Unique farm identifier
Herd status: dependant variable, target, label, desired outcome			
first_status	Nominal	2	BDV herd status at the beginning of the period
last_status	Nominal	2	BVD herd status at the end of the period
decisions	Numeric	2 - 215	Number of confirmative decisions (Negative or Not Negative) taken on of farm
Negative	Numeric	0 - 95	Number of times a farm has had Negative decisions within the time period.
NotNegative	Numeric	0- 215	Number of times a farm has had Not Negative decisions within the time period
always_Negative	Numeric / boolean	2	Whether a farm's BVD status has remained Negative (BVD free) throughout the period
always_NotNegative	Numeric / boolean	2	Whether a farm's BVD status has remained Not Negative (infected) throughout the period
change	Numeric / boolean	2	Whether a farm's BVD status has changed within period
bad_change	Numeric	0 - 8	Number of times a farm's BVD status has changed from Negative to Not Negative
good_change	Numeric	0 - 8	Number of times a farm's BVD status has changed from Not Negative to Negative
multi_change	Numeric / boolean	2	Whether a farm has experienced both 'good change' and 'bad change' at least once
Cattle: independent variable, feature, predictor			
cat_num	Numeric	1 - 7048	Number of cattle on a farm
cat_one	Numeric	0 - 2615	Number of cattle under one year old on a farm
cat_oneplus	Numeric	0 - 6570	Number of cattle over one year old on a farm
Network measures: independent variable, feature, predictor			
in_degree	Numeric	0 - 6382	Number of cattle moved to a farm
nw_farms	Numeric	0 - 2694	Number of farms from which a farm has received cattle
nw_importcountry	Numeric	0 - 10	Number of countries from which a farm has imported cattle
Farm Breed Type: independent variable, feature, predictor			
Beef	Numeric / boolean	2	Whether a farm has cattle bred for beef
Dairy	Numeric / boolean	2	Whether a farm has cattle bred for dairy
Dual Breed	Numeric / boolean	2	Whether a farm has cattle bred for dual purposes
farm_type	nominal	7	Farm classification based on cattle breed purposes present
Other species: independent variable, feature, predictor			
s_sheep	Numeric	0 - 7393	Number of sheep on a farm
s_gost	Numeric	0 - 172	Number of goats on a farm
s_pig	Numeric	0 - 5500	Number of pigs on a farm

3.2.1.1 General Distribution of the Herd Status

This section takes a general look at the effect of the eradication policy from 2012 to 2017 as well as herd status in 2017.

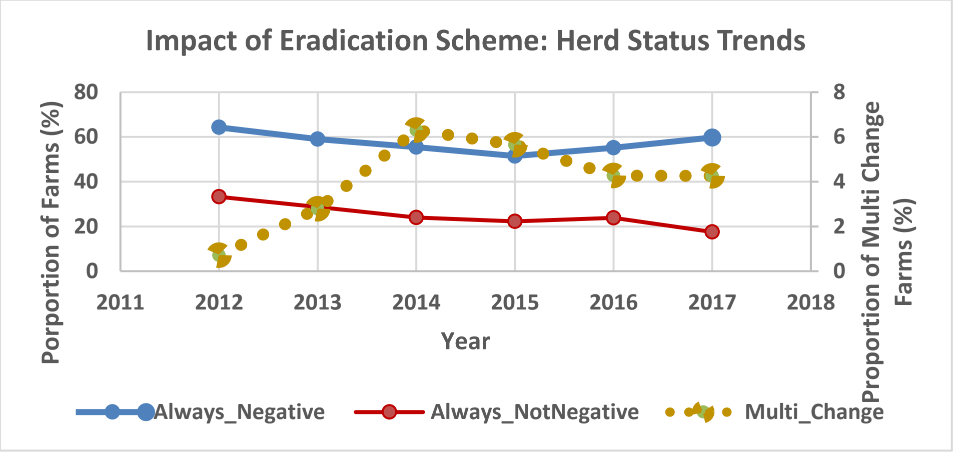


Figure 3. ***Trend of three herd status from 2012 to 2017 showing the impact the eradication policy has had on farms' BVD status.***

From figure 3, it can be seen that the proportion of cattle farms that have remained *Always Negative* was decreasing but this began to improve / increased after 2015. Also, the proportion of farms that have remained *Always Not Negative* consistently decreased / improved with a marked improvement from 2016 to 2017. The proportion of farms that have experienced *Multiple Changes* increased until 2014 after which the situation began to improve / decrease. All these trends show a general improvement in BVD farm status. These observations can be attributed to the impact of the eradication policy on herds' status.

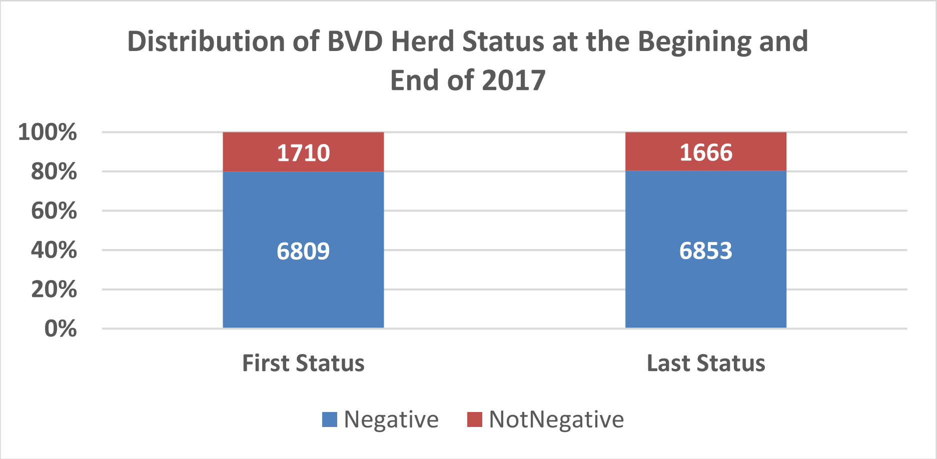


Figure 4. ***Distribution of BVD status at the begin (First Status) and end (Last Status) of 2017 showing the proportion of uninfected (Negative) and infected (Not Negative) farms.***

Figure 4 shows that the percentage of infected farms at the beginning of the 2017 was 20.1% (1710 farms). By the end of the year, this figure had slightly reduced to 19.5 % (1666 farms). This means that 44 farms had their status changed from Not Negative to Negative.

3.2.1.2 Distribution of Classes within Response Variables

The distribution of classes within each variable is of paramount importance to data engineering and modelling decisions as well as performance of a model. The following section looks at the distribution classes within the herd status variable and briefly justifies the pre-processing decisions made.

Figure 5 shows that the Always Not Negative response variable of the *Always Not Negative* dataset is highly imbalanced as only 3 percent (257 of 8519) of the farm had remained always BVD-infected (Always Not Negative) in 2017. This may require calibration to deal with the bias due to uneven class in order to improve accuracy performance [17], [18], [21]. Also, this skewed distribution may affect a model's ability to predict BVD free farms. The rest of the datasets have a fairly balanced class in their respective response variables.

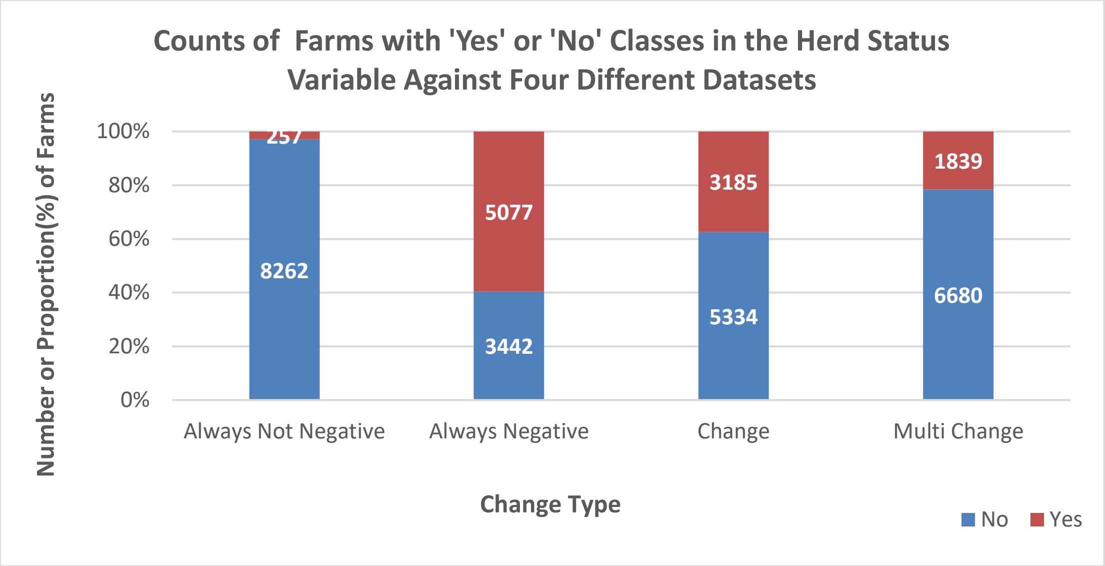


Figure 5. ***Distribution of BVD status classes within the response variables of four machine learning datasets (Always Not Negative, Always Negative, Change and Multi Change) with nominal values.***

Table 3.2 provides the descriptive statistics report of datasets with numeric herd status classes. It can be observed that the number of confirmed (BVD herd status) decisions on a farm range from 2 to 215. Since it was intended to determine the quality (good or bad) of BVD herd status change, all farms whose confirmed decisions were below two were excluded from the dataset. Of the total of 65,187 herd status decisions, 24,068 (36.9%) were confirmed Not Negative (BVD infected) and 41,119 (60.1%) were confirmed BVD Negative (not infected).

The quality of the change farms had experienced in the year 2017 is expressed as either a *good change* or a *bad change* (table 3.1). A farm is said to have experienced *good change* if its BVD status changes from a Not Negative to a Negative. Conversely, if its BVD status changes from a Negative to a Not Negative, a farm is said to have experienced *bad change.* Any farm that experiences both *good change* and *bad change* at least once is said to have experienced a *multi change*. Generally, more farms had experienced a *good change* (3805) than a *bad change* (2881) in 2017 (tables 3.2 & 3.3 and figure 5). This could be attributed to the implementation of BVD Eradication Policy in Scotland. However, a majority of the farms (almost 63%, figure 5) remained in their previous state. Only a little over 37 % (3,185 of 8,519, figure 5) of farms had their status changed in 2017.

Table 3.2 Descriptive statistics report of datasets with numeric herd status classes.

Variable	Minimum	Maximum	Mean	Standard deviation	Total
confirmed decisions	2	215	7.6	8.6	65,187
Not Negative	0	215	2.8	7.9	24,068
Negative	0	95	4.8	3.6	41,119
bad_change	0	8	0.3	0.7	2,881
good_change	0	8	0.4	0.8	3,805

Figure 6 and table 3.3 shows the number of times cattle herds in Scotland experienced either a good change or a bad change in 2017. For the *Good Change* dataset, almost 67% of the farms did not experience a good change (table 3.3). Similarly, for the *Bad Change* dataset, over 74% of the farms did not experience a bad change (table 3.3).

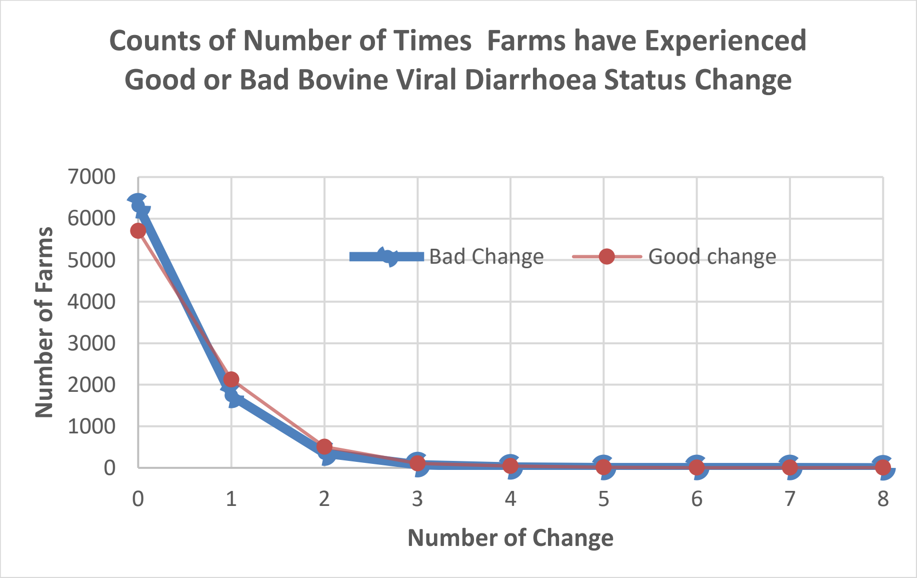


Figure 6. ***Frequencies of number of times farms have experienced good or bad Bovine Viral Diarrhoea status change in 2017.***

Table 3.3 Counts of number of times (and corresponding percentage) farms have experience good or bad change in 2017

Number of changes	Good change		Bad change	
0	5706	66.98 %	6308	74.05 %
1	2126	24.96 %	1744	20.47 %
2	507	5.95 %	350	4.11 %
3	110	1.29 %	73	0.86 %
4	42	0.49 %	23	0.27 %
5	15	0.18 %	8	0.09 %
6	4	0.05 %	7	0.08 %
7	4	0.05 %	4	0.05 %
8	5	0.06 %	2	0.02 %

In order to have a balanced class within these response variables in their respective datasets, the good change and bad change variables were recoded to have binary / Boolean values. This implies that, for the *Good Change* dataset (for example), a '*0*' represents the farm did not experienced a good change and a '*1*' represent the farms experienced a good change. The final *Good Change* dataset subsequently consisted of 66.98% '*0s*' and 33.02% '*1s*'. Similarly, final *Good Change* dataset subsequently consisted of 74.05% '*0s*' and 25.95% '*1s*'.

3.2.1.3 Distribution of Classes within Predictor Variables

The following section considers the distribution of classes within the farm characteristics variable and briefly justifies the pre-processing decisions made.

Breed purpose: From figure 7, it can be seen that the most common cattle breed purpose is Beef as 8371 of the 8519 farms have breeds of cattle used for beef production. The Beef variable is shown to exhibit *majority-group* (98.3%, figure 7) and should normally be excluded from the dataset. However, it was intentionally kept in the dataset as a control. Any algorithm that identifies the Beef variable as an important risk factor would thus be considered ineffective or incompetent.

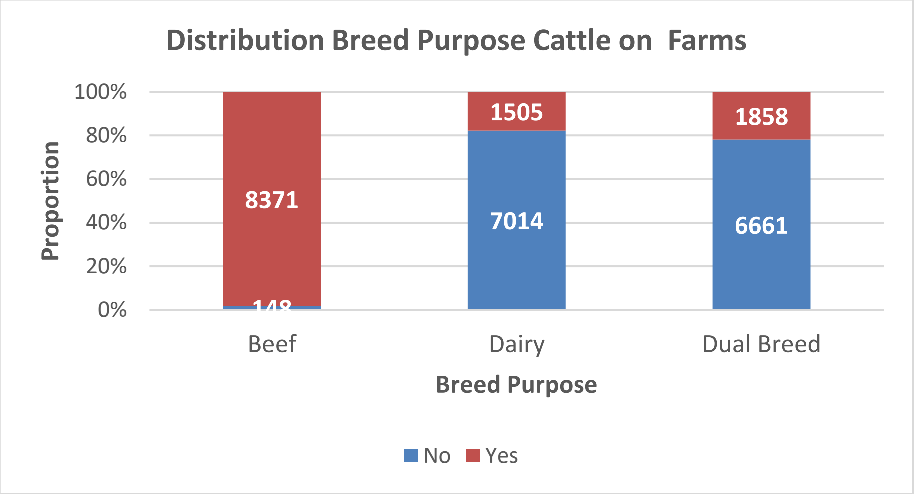


Figure 7. ***Distribution of breed purpose cattle on farms in Scotland.***

Farm Type: It was also interesting to classify farms according to the number of breed types present on each farm (figures 8 & 9). According to this classification, 'Beef' farms are the most common in Scotland (67%). 'Dairy' and 'Dairy-DualBreed' are the less common farms (0.4% and 0.1% respectively).

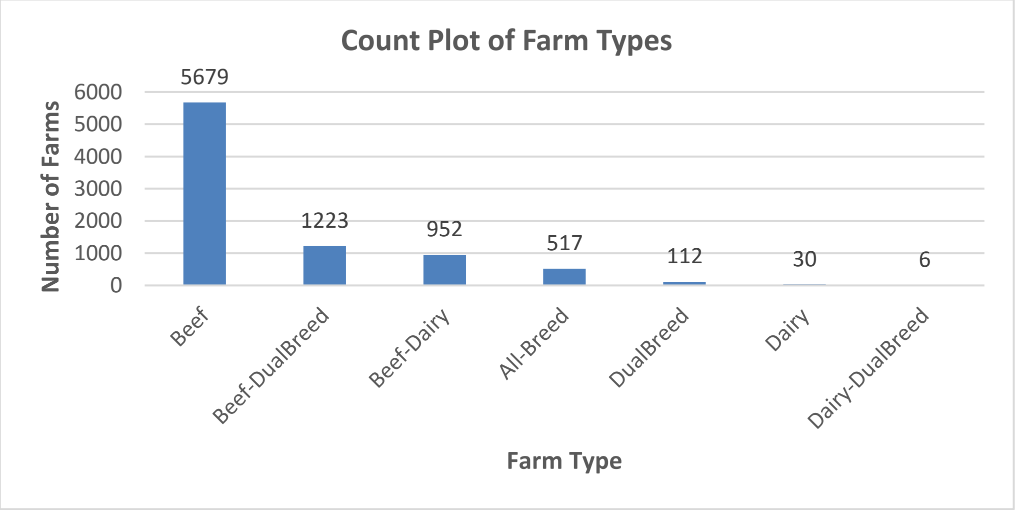


Figure 8. ***Distribution of cattle farm types in Scotland in 2017***

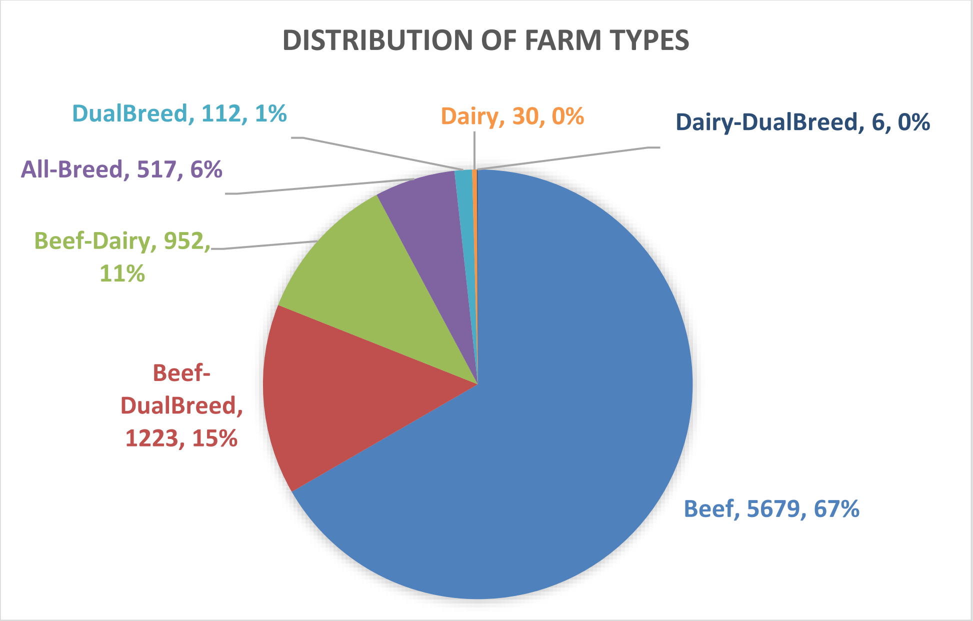


Figure 9. ***Share of cattle farm type in Scotland in 2017***

Other species: From table 3.4 (page 21), it can be seen that there were more sheep (2,434,204) in 2013 than cattle (822,930) in 2017. The presence (59.4%) or otherwise (40.6%) of sheep on farms is almost balanced (figure 10). It can therefore be a good predictor. The average cattle farm in Scotland has about 286 heads of sheep with a range of from zero to 7, 393 animals.

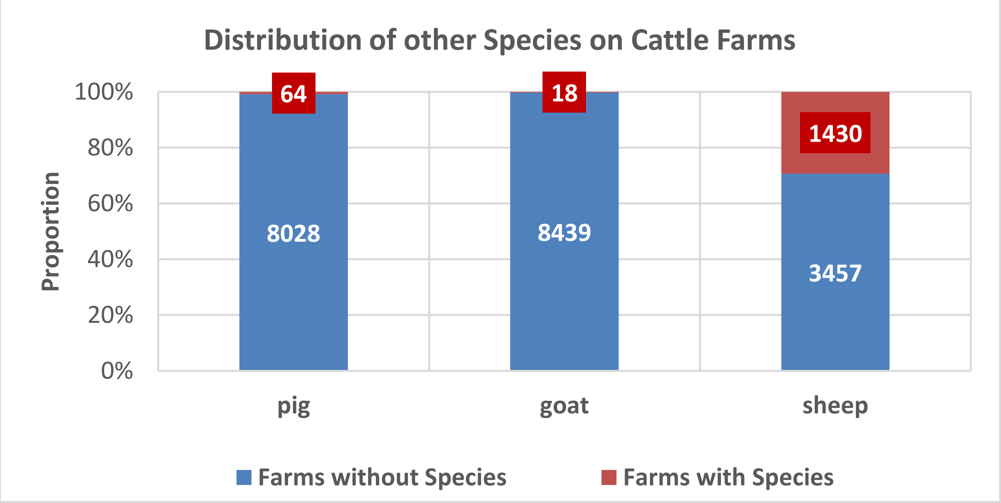


Figure 10. ***Distribution of sheep, goats and pigs on farms in Scotland.***

The number of goats and pigs on a farm range from zero to 172 and zero to 5500 respectively (table 3.4 page 21). It can also be seen that goats are very scarce on cattle farms in Scotland. The total number of goats and pigs was 711 and 55,633 respectively. Also, only 0.9 % and 5.8 % of the farms had goats and pigs respectively (figure 10). These distributions exhibit minority groupings and should not normally be included [17], [18] in the final dataset. However, they were kept in the final datasets for a particular reason. If identified and ranked as an important risk factor, two conclusions can then be drawn:

1. that the model that picks any of them is suspicious / incompetent, or

2. its/their risk(s) is/are really important, especially if the algorithm is proven to be reliable and competent. It can then, confidently, be concluded that it /they has/have very a strong correlation with farm BVD status.

Table 3.4 Distribution of some Farm Characteristics / Features

	Min	Max	Mean	Standard deviation	Total	Percentage
cat_num	1	7,048	96.6	214.8	822,930	
cat_one	0	2,615	71.2	105.1	607,207	73.8
cat_oneplus	0	6,570	25.3	25.3	215,723	26.2
in_degree	0	6,382	30.3	168.7	258,301	31.4
nw_farms	0	2,694	7.1	48.2	60,495	
nw_importcountry	0	10	0.2	0.6	1,967	
s_sheep	0	7,393	285.7	515.3	2,434,204	
s_goat	0	172	0.1	2.6	711	
s_pig	0	5,500	6.5	137.6	55633	

Cattle population: There were a total of 822,930 cattle within the area and time period explored (table 3.4). Of these, over 73 percent (607,207) were born in 2017. This presents a high risk as discussed in chapter 5.3.2.1 (page 43). The distribution of these and all other variables are wide and varied over individual farms. For example, the range of total number of cattle on a farm was from one to 7048 with 214.8 as the standard deviation. Datasets with such a distribution need to be scaled to ensure that none of the values have an undue influence on the overall learning process, thereby improving the model's performance accuracy and speeding up computation [17], [18]. However, since in this project, the ability to interpret the results overrides other considerations, scaling will be curtailed.

In general, it can be said that the larger the number of cattle on a farm, the greater its risk of being declared Not Negative herd status (figure 11). According to this classification, most farms in the UK are either Mini or Small. Generally, persistently infected (PI) calves are recognised as the most important group in term of BVD transmission [4] – [8]. Even though some PIs live to reproductive age, and some may not be identified, the majority do not life to their second year [5]. Therefore, it was also relevant to separate the calves under one year from the rest of the herd and to check their risk importance in this study.

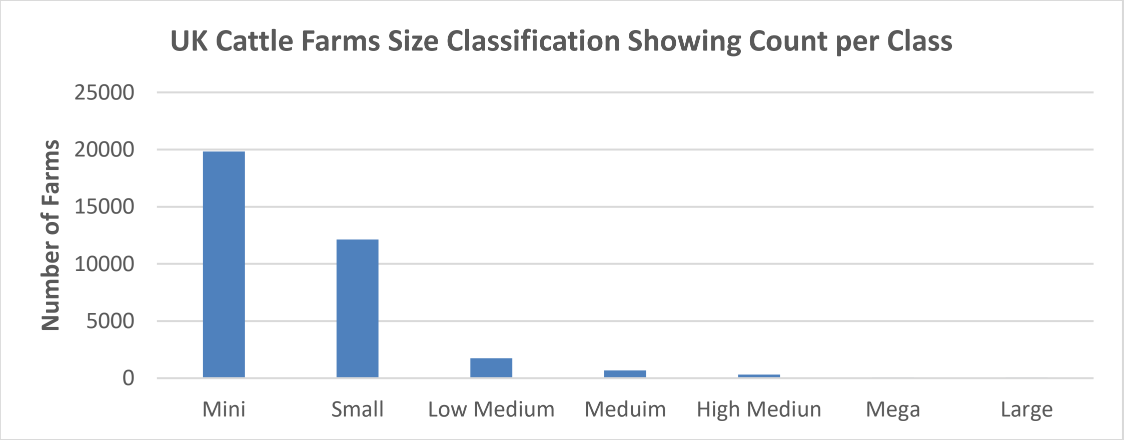


Figure 11. ***Distribution of UK farms by size***

Network metrics: The aim of introducing interconnectivity measures as features in this study was to ascertain whether or not closeness of openness of a farm presents a risk to its BVD status. Three aspects in total in-degree connectivity were considered (table 3.4). Of the 822,930 total cattle in 2017, 258,301 (31.4%), were involved in unidirectional movement to the farms considered (table 3.4). The number of other farms connected to a farm via cattle movement ranged from zero to 2694 in 2017 (table 3.4). This means that farms in Scotland are generally highly interconnected as a farm may be connected to as many as 2694 (31.6%) other farms in Scotland alone. The total number of such unidirectional farm-to-farm connections stood at 60,495. Similarly, a farm in Scotland may be connected to as many as ten countries outside the United Kingdom. The total number of such connections was 1967 (table 3.4) in 2017. In general, the higher the number of connections a farm has the more vulnerable it is to be infected (and re-infected) with the BVDV [13] - [16]. These in-degree network centralities can therefore be very good predictors of farms' BVD status.

3.2.2 Data Pre-processing

The influence of a Data Scientist on the success of machine learning projects invariably depends on the data pre-processing decisions he/she makes. Exploration Data Analysis favourably positions the scientist to apply the correct pre-processing techniques - cleaning and transformation (box 3.1).

3.2.2.1 Data Cleaning

Data cleaning was done to improve the data quality. This included removal of duplicate and irrelevant records, filling of empty cells with appropriate values and recoding nominal values to numeric values as well as rectifying inconsistent values. The following figures (12, 13 & 14) give one example of how the data was cleaned. Note that confirmed BVD herd statuses are Negative and Not Negative. Unconfirmed herd statuses are 'Part Test', 'Not Applicable' and 'Pending'.

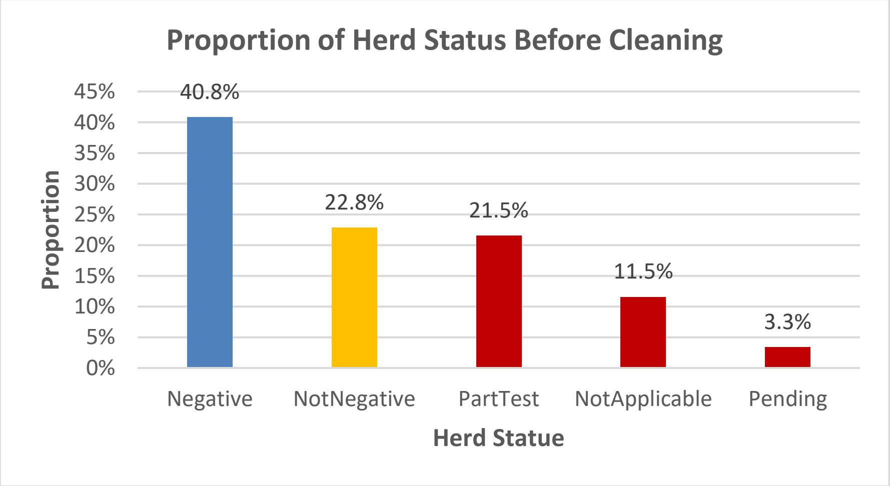


Figure 12. ***Distribution of confirmed and unconfirmed BVD herd status decision in UK before data cleaning.***

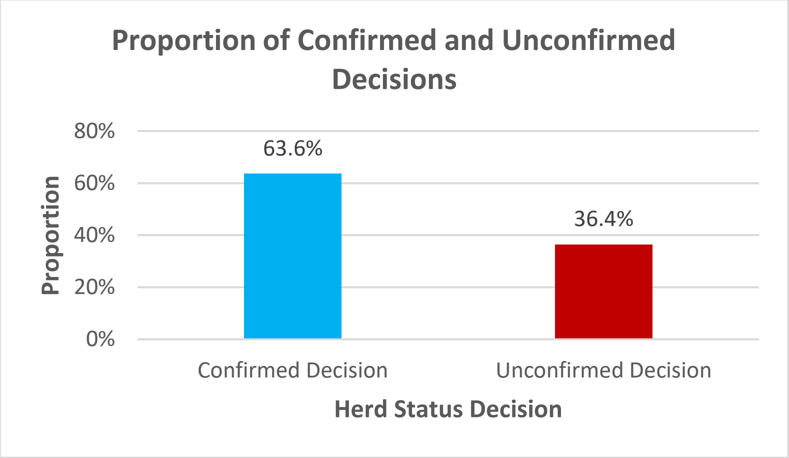


Figure 13. ***Distribution of combined confirmed and unconfirmed BVD herd status decisions in UK***

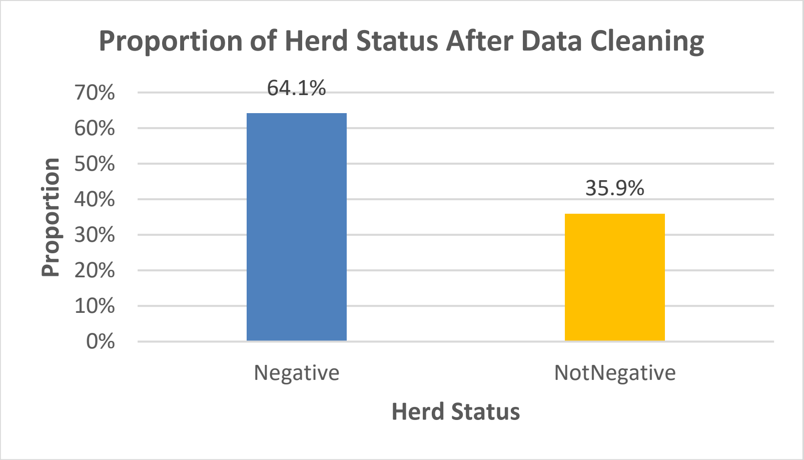


Figure 14. Distribution of combined confirmed BVD herd status decisions in UK after data cleaning

3.2.2.2 Data Transformation

Data transformation renders the data in the form the model can accept and lead to improved performance and converging speeds. Box 3.1 provides a general process for machine learning data pre-processing. New features (farm characteristics or risk factors) were created (figure 2 and table 3.1). All categorical features were converted to numeric values. Any numeric feature whose values were not ordinal were converted to dummy variables and steps were taken to avoid the dummy variable traps.

Box 3.1 Generic machine learning data pre-processing steps.

1. Import required libraries
2. Load the dataset and create feature matrix and target vector
3. Handle missing data
4. Encode variables with nominal values to numeric ones. Further create dummy variables if values are not ordinal and take steps to avoid dummy variable traps
5. Scale features to improve performance (accuracy) and avoid undue influence and to hasten convergence. However, if interpretability is required scaling can be avoided
6. Splitting the dataset into the training-set and test-set. This allows the trained model to be tested on an unseen dataset to assess how it will perform on deployment.

For the Deep Network model, all the farm characteristics (risk factors) were scaled to improve performance and converging speed. However, since interpretability was required scaling was curtailed for the rest of the models. Seven datasets were created to find out which of them were best suited to solve the business problem. Finally, each dataset was divided into training and testing sets. Whereas the training set was used to train the machine learning models, the test set was used to assess the performance of the models.

3.3 Model Building and Analysis

This phase includes selection of some techniques; building and validating model(s) (box 3.2). First, the data was split into training and testing sets (usually 70% and 30 % respectively). When cross-validation is not applied, it is recommended to extract some of the training data set as a validation set. A small number of training models are then built by using different hyper-parameter settings. Alternatively, grid-search may be performed to achieve hyper-parameter tuning. A good bias-variance trade-off must be achieved to avoid model overfitting (or under-fitting). Based on overall performance, the best training model is then selected and saved.

Box 3.2 Generic machine learning modelling process. This follows the pre-processing phase (box 3.1)

1. Import required libraries
2. Create model object and fit it to training dataset
3. Use trained model to predict the test dataset
4. Invoke the performance report: confusion matrix, accuracy, classification report, ROC area, etc.
5. For reliable and robust models, perform k-fold cross-validation and produce mean accuracy
6. Hyper-parameter tuning. For optimal performance, use Grid Search (with k-fold cross-validation) to optimise the unlearnt parameters. Invoke best accuracy and best parameters
7. For reliable and robust model perform k-fold cross-validation
8. Visualise results

3.3.1 Testing the Hypothesis

The hypothesis some models are more useful for explaining their decisions to a non-data scientist than other was effectively tested by presenting three different visual models to non-data scientist experts (see chapter 5.4) and observing their responses to the models. These responses were in the form of preference for particular model during interactions with other experts or reference to a particular model in relation to issues raised during the presentation of the findings of this study. These responses (thought subjective) were used as a measure of a model's easy of interpretability. These measures were then used to ascertain whether or not all the models have the same usefulness in explaining their decisions to non-data science experts.

3.4 Evaluation and Deployment

The last two phases are evaluation and deployment. The selected model is re-evaluated on unseen test dataset to assess its robustness. If it performs satisfactorily, the model is deployed for business use. It is also important to assess the performance of the selected model [17] in order to know its strengths and weaknesses. When deploying the model, it is essential to accompany it with a report of all the processes applied together with its evaluated performance metrics.

3.5 Weka: J48 Decision Tree

In management applications, it is often desirable to be able to interpret the result of a machine learning project to enable customised management interventions to be formulated and applied. The simple Decision Tree model gave detailed interpretable graphics but was inappropriate to the problem to be solved. This is because the Decision Tree model could not deal with the imbalanced classes in the dataset. Its pruned tree produced only 'Negative' classes at the decision/leaf nodes rendering it unsuitable for the management challenge. The XGBoost gave equally information-rich visuals but it requires an external key to interpret the symbols used for the risk factors. A better option is Weka's J48 decision tree which gives a better visualisation and more readily interpretable graphics that can be used to solve the problem for this project. This section briefly describes the procedure to build Weka's J48 decision tree coupled with a brief evaluation. Only the results and evaluations of the principal dataset (*Last Status*) are provided in the main body of this work. The rest are found in appendix 1. The procedure presented here follows Mensah, (2018) [37].

3.5.1 Building the J48 Model

This section provides the process of building a J48 model. A 10-fold Cross-Validation was used throughout this section to ensure reliable results.

3.5.1.1 Learning Curve and Dataset Division

Weka's Filter Classifier was used to conduct an experiment to determine which share of the *Last Status* dataset was best for training the J48 model. This procedure follows [37].

Setup Experimenter: *Weka.Experimenter.Setup> Experimenter configuration mode>Advanced>New>Destination>chose file type>InstanceResultListener>[click in the property editor]>[click on Browser to select output folder]>[give file name]>Ok>Ok>Result generator>Choose>CrossValidationResulProducer>[click in the property editor]>[accept defaults under]>splitEvaluator>Choose>ClassifierSplitEvaluator)>[click in the property editor, under]>classifier>meta>FilterClassifier>[click in the property editor under] classifier>tree>J48 >[under filter]>Choose>unsupervised>instance (filter)>RemovePercentage [remove percentage from the dataset]>OK>OK>OK>Run>1 to 10>Generator properties>Enabled>Advanced properties>spliEvaluator>classifier>filter>percentage >select[Type the hold percentages (10 to 90 steps 5)]>Dataset>Add new>[navigate to dataset]*

Run Experimenter*: Weka.Experimenter>Run>Start*

Analyse Experimenter: *Weka.Experimenter.**Analyse**>Source>Experiment>>Configure test>Comparison field> Percentage_incorrect> Actions>perform test*

The learning curve plotted (figure 15) shows that the incorrect classification keeps decreasing sharply from 10% to 30% of the retained dataset; that it generally slows down from 30% to 55% after which it almost levels off. This implies that 55 % of the dataset is optimum to properly train the J48 model. Therefore 45% of the dataset will be set aside as a test set with which to evaluate the model's performance.

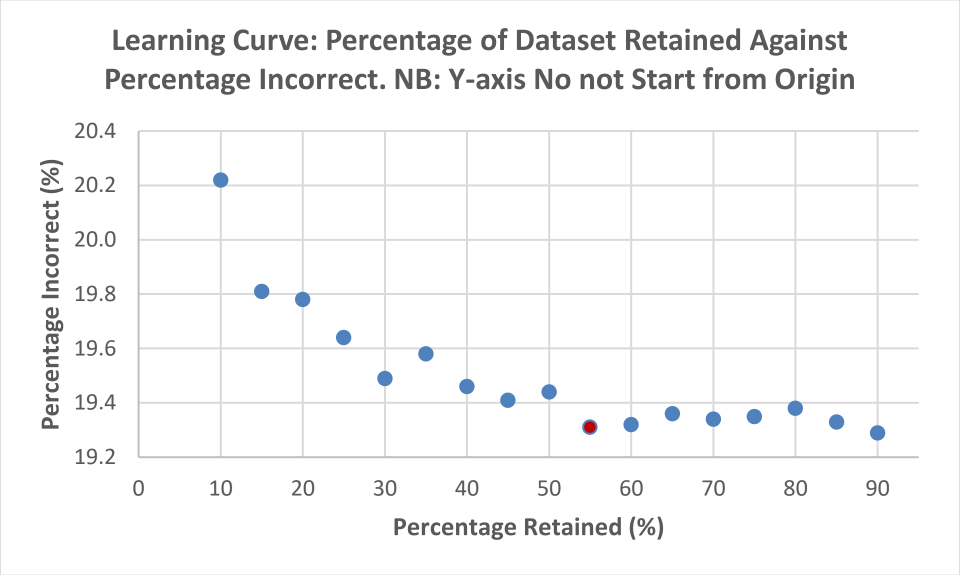


Figure 15. ***Learning curve showing percentage of dataset retained against percentage incorrect***

3.5.1.2 Splitting the Dataset

Based on the learning curve (figure 15), a split in the ratio of *55%:45%* was set to provide the train and test *sets* respectively. This procedure follows [37].

***Randomise**: Weka.preprocess>Open file>[load data]>Filter>Choose>unsupervised>instance>Randomizes>Apply*

***Resample/splitting:** Weka.preprocess>Filter>Choose>unsupervised>instance>Resample>[click to edit properties]> **noReplacement**>True>sampleSizePercent>55>OK>Apply>Save[**all_hs_train**] >[folder]>Undo*
***Test Set** (45% of original)*
*>[click to edit properties]>**invertSelection**>True>noReplacement>True> sampleSizePercent> 55>OK>Apply>Save[**all_hs_test**>[folder]*

Dataset instances / records 100%=8519, 55% train = 4685, 45% test = 3834

3.5.1.3 Hyper-parameter Tuning

The following hyper-parameters were tuned: Pruning, Minimum Number of Objects and Confidence Factor. A 10-fold Cross-Validation was used throughout this section to ensure reliable results.

3.5.1.3.1 Pruning the Decision Trees

Classify>Choose>Weka>classifier>trees>J48>[click property editor]>unpruned>False or True>OK>Start

Table 3.5 Effect of pruning on J48 using 10-fold cross-validation.

Unpruned	Accuracy (%)	FP Rate	Trees	Leaves
False	80.7	0.695	35	18
True	80.4	0.689	87	44

Since pruning gives a simpler tree with 35 nodes (18 leaves) (table 3.5), the model will be pruned in order to generalise with the training dataset.

3.5.1.3.2 Minimum Number of Instances per Leaf

Another way of pruning is to limit the number of training examples that reach a leaf (minNumObj). This parameter was tuned with seven different numbers and the results are shown in table 3.6. The False Positive Rates decreased from 1 to 3 minNumObj after which it began to increase sharply. At the same time, the size of tree decreased sharply from minNumObj of 1 to 3 after which it decreased slightly only to level off after 10. Both table 3.6 and figure 16 show that *minimum number of objects* of 3 is the optimum and that the model generalised well at minNumObj of three.

Table 3.6 Effect of minimum number of objects (*minNumObj)* on performance metrics of J48

minNumObj	1	2	3	5	7	10	15
Accuracy (%)	80.7	80.7	80.9	80.6	80.7	80.7	80.8
Leaves	21	18	14	13	10	5	5
Tree size	41	35	27	25	19	9	9
FP Rate	0.671	0.670	0.670	0.681	0.681	0.678	0.681

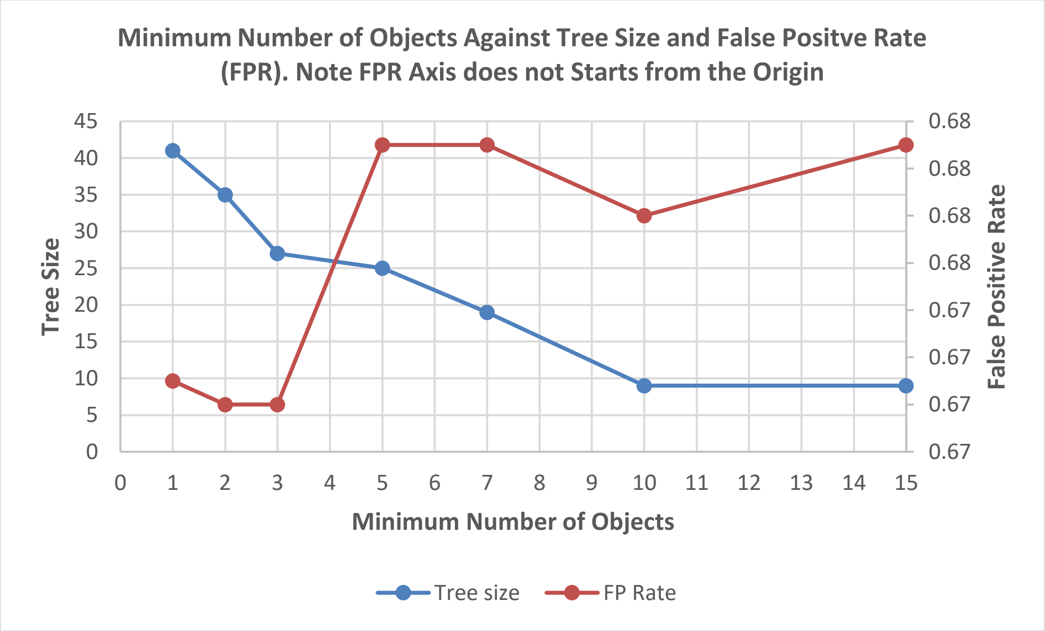


Figure 16. Plot of minimum number of objects against tree size and false positive rates

3.5.1.3.3 Confidence Factor

This hyper parameter takes values between 0 and 1 with smaller values incurring more pruning.

Table 3.7 Effect of confidence factor on number of leaves and accuracy

Confidence factor	0.05	0.10	0.20	0.25	0.50	0.70
Accuracy (%)	80.7	80.8	80.9	80.9	80.6	80.6
Leaves	3	12	14	14	32	40
Tree size	5	23	27	27	63	79
TP Rate	0.708	0.686	0.674	0.670	0.664	0.665

It can be observed from table 3.7 and figure 17 that a confidence factor of 0.20 or 0.25 is the best. The J48 model was therefore built with pruned branches, a *minNumObj* of 3 and 0.25 confidence factor.

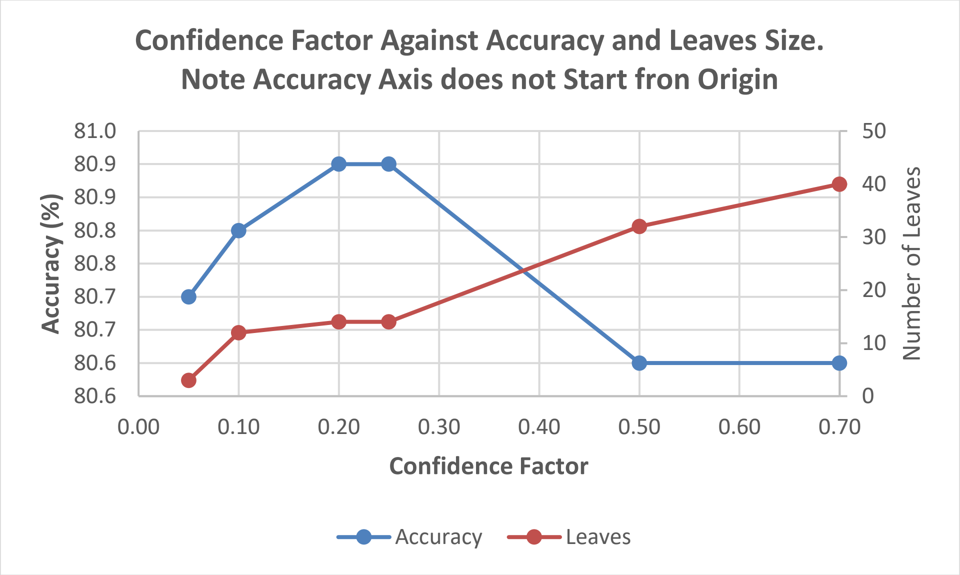


Figure 17. ***A plot of confidence factor against number of leaves and accuracy***

3.5.2 Results

The result of the J48 model (81% accuracy) is summarised in its decision tree (figure 18). This graphics is discussed in chapter 5.4 (page 45).

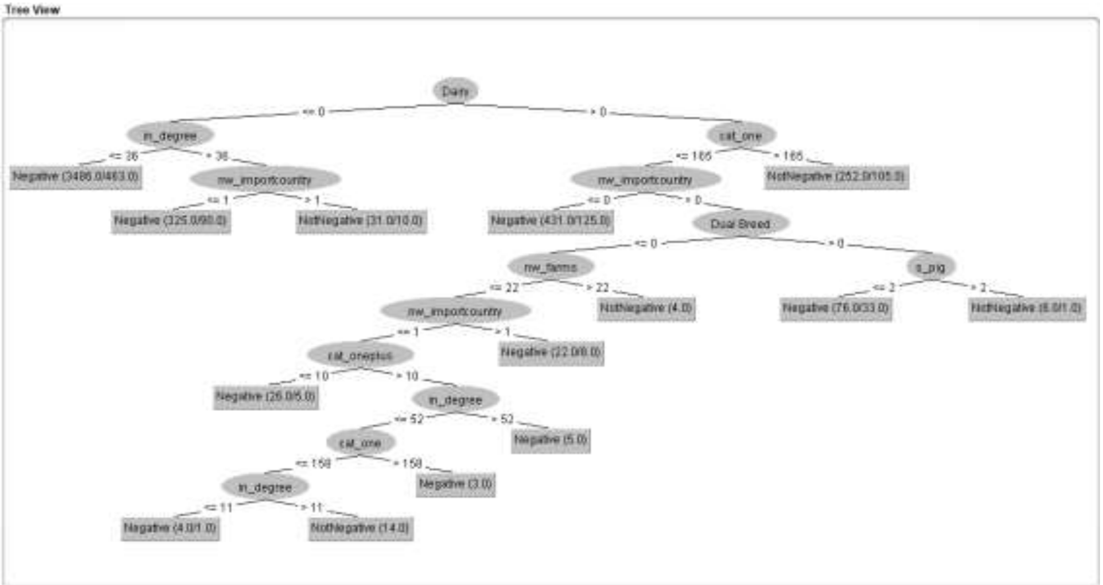


Figure 18. ***Decision tree of J48 built with pruned branches, a minimum number of objects of 3 and 0.25 confidence factor on the Last Status dataset***

3.5.3 Evaluation J48 Model

This section provides brief assessment of the J48 model. The confusion matrix of the J48 model (as presented in table 3.8) shows that the model made more true predictions than false predictions.

Table 3.8 Confusion matrix of the J48 model on the Last Status dataset

		Classified / Predicted	
	Class	Negative	Not Negative
Actual	Negative	2953	131
	Not Negative	606	144

Table 3.9 and figure 19 show that the strengths and weaknesses of both the J48 and XGBoost models are generally similar.

Table 3.9 Comparison of J48 and XGBoost model using performance metrics expressed as percentage

Metrics	Class	Negative	Not Negative	Weighted Avg.
Precision	J48	83%	52%	77%
	XGBoost	81%	54%	76%
Recall	J48	96%	19%	81%
	XGBoost	97%	14%	80%
F-Measure	J48	89%	28%	77%
	XGBoost	88%	23%	75%
ROC Area	J48			67%
	XGBoost			72%
Accuracy	J48			81%
	XGBoost			82%
TP Rate	J48	96%	19%	81%
FP Rate	J48	81%	4%	66%

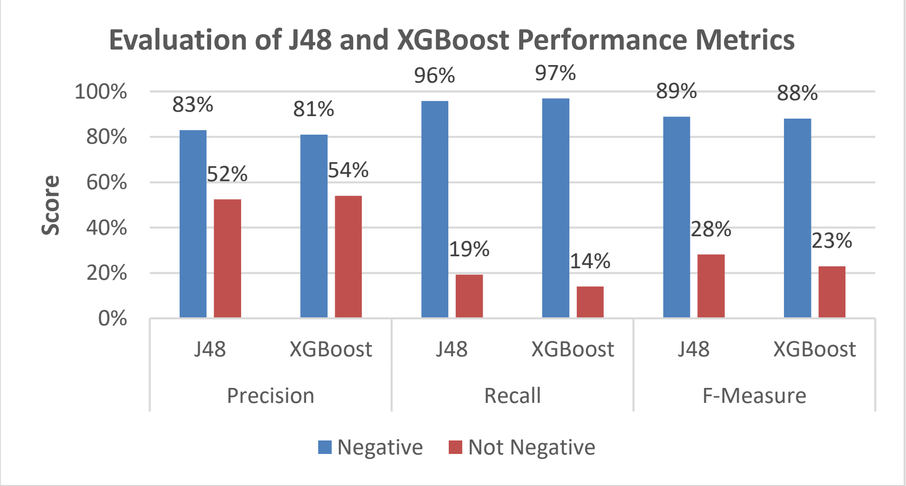


Figure 19. ***Comparison of J48 and XGBoost models on Last Status dataset using performance metrics expressed as percentage***

Figure 20 shows the plot of area under the ROC curve of the J48 model.

ROC Curve

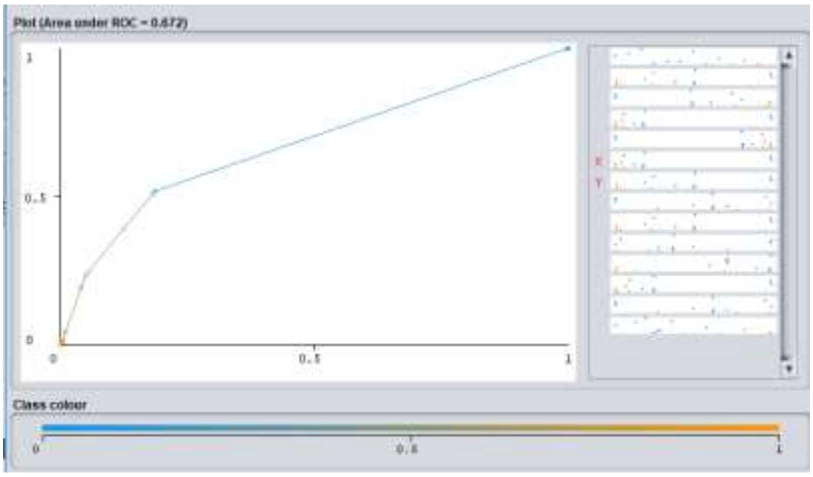


Figure 20. ***Plot of area under the ROC curve of the J48 model on the Last Status dataset***

4 Models and Datasets Evaluation

In machine learning applications, it is important to assess the performance of the selected models [17] in order know their strengths and weaknesses. This section presents a critical analysis of the strengths and weaknesses on the selected datasets, of the selected classification model in relation to others. Performance measures included accuracy and ROC area, specificity versus sensitivity as well as f1-score. As discussed in chapter 2.2.5 (above), these evaluators were chosen for specific reasons. Accuracy was the basic 'quantitative' measure of individual model's performance (chapter 6.1). However, in order to compare two or more models, standardised and 'qualitative' measures devoid of individual error costs are required. In this regard, the area under the ROC curve (AUC) is relevant [17], [18]. This factor is discussed in chapter 6.1. Specificity and sensitivity permit an evaluation of which model or dataset makes the most important prediction or most expensive mistake (chapter 6.2). Since the datasets have uneven classes, the F1-score is a better performance measure than precision alone (chapter 6.3).

4.1 Accuracy and ROC Area

In terms of both accuracy and ROC area, the eXtreme Gradient Boost (XGBoost) classifier is generally slightly better than Artificial Neural Network (ANN) on the selected datasets (figure 21). For example, for the *Last Status* dataset, XGBoost achieved 79.6% accuracy while ANN achieved a 79.1% result. Similarly, *Always Not Negative* and *Always Negative* were indicated to be the best and worst dataset respectively.

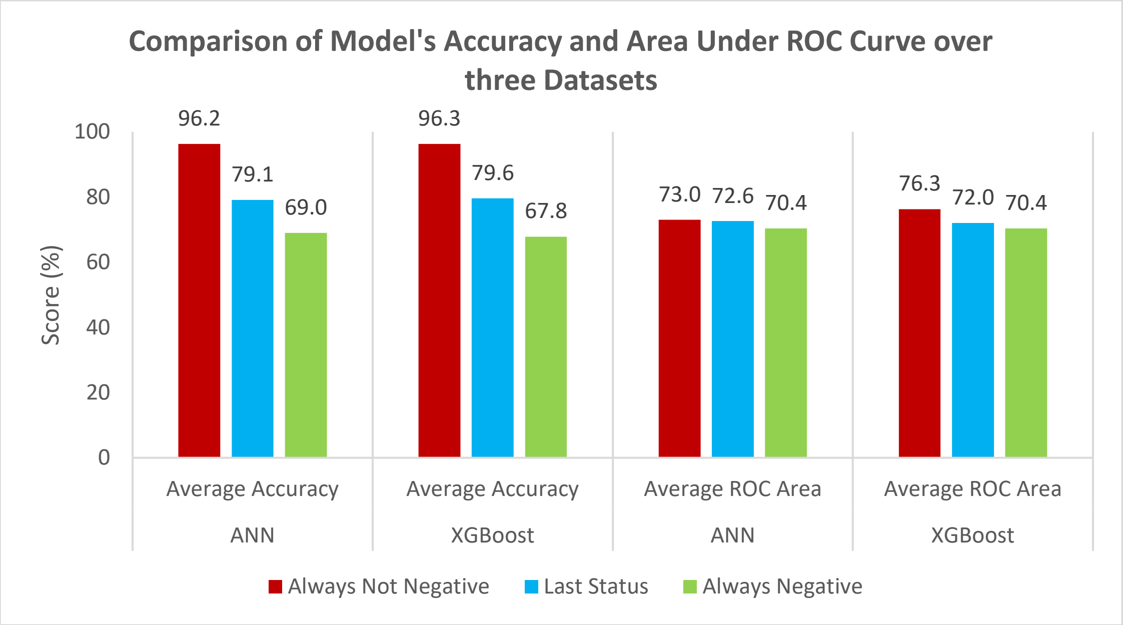


Figure 21. ***Comparison of accuracies and area under Receiver Operation Characteristic (ROC) curve of XGBoost and Artificial Neural Network (ANN) over Always Not Negative, Last Status and Always Negative datasets.***

4.2 Specificity versus Sensitivity

In this use-case, it is more interesting to assess how a model makes important decisions. A model's ability to accurately classify a BVD infected farm as infected (Not Negative) in this use-case, is more important than its capacity to 'truly' predict a BVD free (Negative) herd. This is because the negative impact (cost) of a mis-prediction of infected farms is far higher (disease spread unchecked) than misclassifying uninfected herd (spread of disease is checked even though there is no disease). While specificity measures a model's ability to accurately classify BVD free farms, sensitivity is a measure of how a model is able to correctly predict BVD infected herds [17], [20].

Table 4.1 Consolidated confusion matrix of Decision Tree, Extra Tree, XGBoost, Artificial Neural Network (ANN) classifiers over Last Status, Always Not Negative, Always Negative datasets.

			Predicted							
			Decision Tree		Extra Tree		XGBoost		ANN	
			0	1	0	1	0	1	0	1
Observed	Last Status	Negative = 0	1963	61	1893	131	1959	65	1295	46
		Not Negative = 1	462	70	448	84	456	76	301	62
	Always Not Negative	No = 0	2462	0	2454	8	2461	1	1640	0
		Yes = 1	93	1	92	2	94	0	64	0
	Always Negative	No = 0	349	723	513	559	395	677	309	406
		Yes = 1	115	1369	356	1128	145	1339	126	863

For the *Last Status* dataset, specificity is the number of correctly predicted 'Negatives' divided by the total of actual 'Negatives' (table 4.1 & 4.2). Sensitivity is the number of correctly predicted 'Not Negatives' divided by the total of actual 'Not Negatives' (table 4.1 & 4.2). This dataset follows a 'closed-world assumption' [17], i.e. if a class is not 'Not Negative', then it is 'Negative'. However, the *Always Not Negative* and *Always Negative* datasets cannot lend themselves to a 'closed-world assumption' [17]. For the *Always Not Negative* dataset, the class 'Yes' means a BVD infected farm, but class 'No' does not necessary imply a BVD free farm. It just means that a farm has ever had its BVD status changed. For simplicity, an *assumed-specificity* is assigned to the 'No' class. Consequently, one can confidently compute for sensitivity, (the number of correctly predicted 'Yes' divided by the total actual 'Yes'), but not specificity (table 4.1 & 4.2). Similarly, for the *Always Negative* dataset, the class 'No' does not necessarily imply a BVD infected farm but a 'Yes' class does mean a BVD free farm. Consequently, we can only confidently compute for specificity, (the number of correctly predicted 'Yes' divided by total actual 'Yes'), but not sensitivity (table 4.1 & 4.2). Hence, *assumed-sensitivity* is given to a 'No' class.

Table 4.2 Sensitivity and Specificity Decision Tree, Extra Tree, XGBoost, Artificial Neural Network (ANN) classifiers on the selected datasets

		Decision Tree	Extra Tree	XGBoost	ANN
Last Status	Sensitivity	13.2%	15.8%	14.3%	17.1%
	Specificity	97.0%	93.5%	96.8%	96.6%
Always Not Negative	Sensitivity	1.1%	98.9%	0.0%	0.0%
	Assumed-Specificity	100.0%	0.0%	100.0%	100.0%
Always Negative	Assumed-Sensitivity	92.3%	76.0%	90.2%	87.3%
	Specificity	32.6%	47.9%	36.8%	43.2%

Generally, all the models performed equally on a specific dataset (table 4.2 and figure 22). However, they generally achieved higher specificity than sensitivity This means that they are stronger in predicting BVD free farms than BVD infected farms. This can be attributed to the imbalanced classes of the response variable in favour of BDV free class (Negative 79%, No 96% for *Last Status* and *Always Not Negative* datasets respective, table 4.2).

The datasets, however, responded differently to the models. For example, the Decision Tree classifier achieved 100% assumed-specificity on the *Always Not Negative* dataset. The reason for this is that only all instances were ‘truly’ predicted as BVD free by the Decision Tree classifier and none was ‘falsely’ classified as BVD free (table 4.1). In contrast, no instance was correctly predicted as BVD infected by the XGBoost classifier. Consequently, it achieved zero percent sensitivity (table 4.1) Also, since only one wrongly classified BVD free case was recorded by the XGBoost classifier on the *Always Not Negative* dataset, it recorded an almost 100% specificity. For the Neural Network, no farm was either ‘truly’ classified as infected or ‘falsely’ predicted as free (table 4.1). Therefore, it recorded a zero percent sensitivity and 100% assumed-specificity (table 4.2 and figure 22). These observations show that although the *Always Not Negative* dataset achieved very high accuracy than the rest, its decision is very skewed and can therefore not be trusted.

Also, the *Always Negative* dataset achieved the fairest score for both sensitivity (assumed) and specificity (table 4.2 and figure 22). This could be attributed to the almost even-class of the response variable (59.6% ‘Yes’ versus 40.4% ‘No’, figure 5). The *Always Negative* dataset recorded a markedly higher sensitivity (assumed) than all the other datasets. For instance, whilst the XGBoost managed to achieve only 14.3% and 0.0% sensitivity on the *Last Status* and *Always Not Negative* datasets respectively, it achieved tremendously 90.2% sensitivity (assumed) on the *Always Negative* dataset (table 4.2 and figure 22). This implies that, the *Always Negative* could be the best dataset for revealing BVD infected farm if a ‘closed world’ is assumed.

In terms of specificity, the *Last Status* and *Always Not Negative* datasets the recorded highest scores (which are almost the same). This implies that either of them is very competent in classifying BVD free herds. These observations could be attributed to the imbalanced classes of the response variable in favour of BDV free class (*Last Status* variable: 79% ‘Negative’ and *Always Not Negative* variable: 96% ‘No’, table 4.2). However, given that the *Last Status* dataset is able to predict both sensitivity and specificity (and the decision of the *Always Not Negative* dataset is found to be skewed), it is the recommended dataset as it does not require a ‘closed world’ assumption, although it is stronger in predicting in BVD free farms than BVD infected farms.

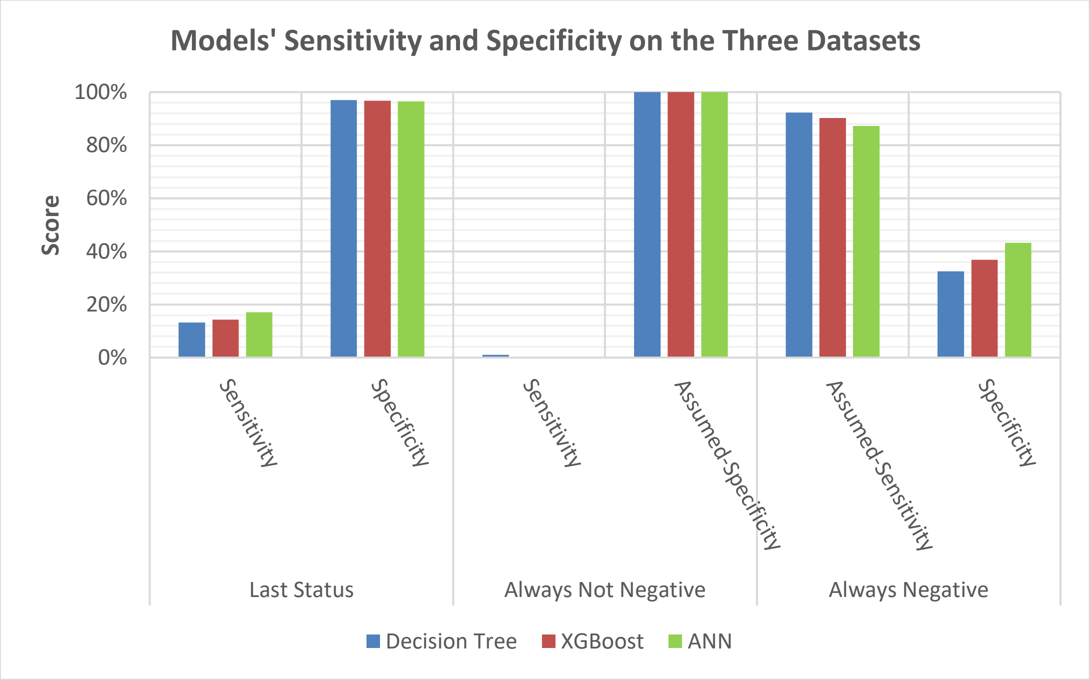


Figure 22. Comparison of the three models’ ability to identify BVD infected farms (sensitivity) and BVD free farms (specificity).

4.3 Precision-Recall Harmonisation: f1-score

Table 4.3: Precision, recall and f1-score information of sensitivity and specificity

Model	Dataset	Last Status			Always Not Negative			Always Negative		
	Class	Negative	Not Negative	Avg./ Total	No	Yes	Avg./ Total	No	Yes	Avg./ Total
Decision Tree	Precision	81%	53%	75%	96%	50%	95%	75%	65%	70%
	Recall	97%	13%	80%	100%	1%	96%	33%	92%	67%
	f1-score	88%	21%	74%	98%	2%	95%	45%	77%	64%
Extra Tree	Precision	81%	39%	72%	96%	20%	94%	59%	67%	64%
	Recall	94%	16%	77%	100%	2%	96%	48%	76%	64%
	f1-score	87%	22%	73%	98%	4%	95%	53%	71%	63%
XGBoost	Precision	81%	54%	75%	96%	0%	93%	73%	66%	69%
	Recall	97%	14%	80%	100%	0%	96%	37%	90%	68%
	f1-score	88%	23%	75%	98%	0%	94%	49%	77%	65%
ANN	Precision	81%	53%	75%	96%	0%	93%	70%	69%	69%
	Recall	96%	15%	79%	100%	0%	96%	46%	86%	69%
	f1-score	88%	24%	74%	98%	0%	94%	56%	76%	68%
	Support	79%	21%	100%	96%	4%	100%	42%	58%	100%

Although it may be criticised in some applications [33], f1-score is an effective performance measure for datasets with imbalanced classes (table 4.3 and figure 4.3) [34]. Although *Always Not Negative* dataset recorded the highest f1-scores for the 'No' class, (98% in all models), it achieved the lowest f1 scores for the 'Yes' class (2%, 4%, 0% and 0% for Decision Tree, Extra Tree, XGBoost and ANN models respectively, table 4.3). This disparity can be ascribed to uneven distribution of the classes in the response variable (96% against 4% support in favour of the 'No' class, table 4.3). The *Always Negative* dataset which has fairly distributed classes of the response variable achieved the most balanced f1-scores for all the classes of the response variable.

4.4 Performance Optimisation

Tables 4.4 and 4.5 show the effect of variable selection and hyper-parameter tuning on XGBoost classifier. Both recursive feature elimination and XGBoost feature ranking showed Beef, Goats and Dual Breed variables as redundant.

Table 4.4 Effect of optimisation techniques on XGBoost accuracy with standard deviation in brackets

Dataset	Last Status	Always Not Negative	Always Negative
Initial	79.61%	96.28%	67.84%
Initial + 10-fold Cross-Validation	81.78% (±0.76%)	97.21% (±0.15%)	69.00% (±0.01%)
Best Variables	79.54%	96.28%	67.53%
Best Variables + 10-fold Cross-Validation	81.69% (±0.75%)	97.22% (±0.15%)	68.89% (±1.10%)
Optimised	81.97%	97.28%	69.13%
Optimised + Re-run	79.70%	96.28%	67.29%
Optimised + 10-fold Cross-Validation	81.97% (±0.59%)	97.28% (±0.10%)	69.13% (±1.60%)

It was expected that their exclusion from the dataset would improve the model's performance metrics. However, XGBoost did not achieve an improvement. For example, the accuracies on *Last Status* dataset with (initial) and without redundant (best) variables was 79.6% and 79.5% respectively (tables 4.4). Furthermore, with or without optimisation, the Negative class of *Last Status* dataset, the precision, recall and f1-scores remained unchanged, 81%, 97% and 88% respectively (table 4.5).

Table 4.5 Effect of optimisation techniques on XGBoost on precision, recall and f1-score

		Last Status			Always Not Negative			Always Negative		
	Class	Negative	Not Negative	Avg./ Total	No	Yes	Avg./ Total	No	Yes	Avg./ Total
Precision	Initial	81.0%	54.0%	75.0%	96.0%	0.0%	93.0%	73.0%	66.0%	69.0%
	Best Variables	81.0%	53.0%	75.0%	96.0%	33.0%	94.0%	72.0%	66.0%	69.0%
	Optimised	81.0%	54.0%	76.0%	96.0%	0.0%	93.0%	72.0%	66.0%	68.0%
Recall	Initial	97.0%	14.0%	80.0%	100.0%	0.0%	96.0%	37.0%	90.0%	68.0%
	Best Variables	97.0%	14.0%	80.0%	100.0%	1.0%	96.0%	36.0%	90.0%	68.0%
	Optimised	97.0%	14.0%	80.0%	100.0%	0.0%	96.0%	36.0%	90.0%	67.0%
F1-score	Initial	88.0%	23.0%	75.0%	98.0%	0.0%	94.0%	49.0%	77.0%	65.0%
	Best Variables	88.0%	22.0%	74.0%	98.0%	2.0%	95.0%	48.0%	76.0%	65.0%
	Optimised	88.0%	23.0%	75.0%	98.0%	0.0%	95.0%	48.0%	76.0%	64.0%

These observations can be explained by the fact that XGBoost has a regularisation term which checks over-fitting, in-built cross-validation and non-greedy tree pruning mechanisms. Therefore, assertions that the XGBoost is robust and its results are reliable are supported [25]. As a result, it does not need hyper-parameter optimisation or cross-validation to give best outputs. In addition, on these datasets, it can be said that XGBoost is robust to both irrelevant and redundant variables.

4.5 Evaluating the Visual Models

Since part of the objectives of this study was visualise and interpret the results of the model(s), three different graphics models were built and tested on non-data scientist experts to ascertain the models' usefulness for explaining their decisions (see chapter 5.4). Based on the datasets and the responses from other professionals, the J48 model was found to offer the easiest explaining of its decision (see chapter 5.4.3).

5 Results and Discussions

This chapter presents the salient findings of the study together with accompanying discussions. Each subsection begins with a bulletin/summary of the findings before moving to the main discussion.

5.1 Model Selection

The eXtreme Gradient Boost (XGBoost) model proved to be the best for the datasets, followed by the Logistic Regression. The *Last Status* dataset proved to be the best general-purpose dataset.

Table 5.1 presents the results of model selection trials. As expected, the baseline algorithm, Dummy Classifier, generally achieved the least accuracy over all the datasets (except on two occasions where it achieved better results than the Decision Tree model). Surprisingly, the Logistic Regression, a simple classifier, achieved relatively high accuracies, outperforming the Artificial Neural Network (ANN) on three of the seven occasions. This observation can be attributed to the binary nature of the response variable [18], [19]. The outstanding performance of the XGBoost and the ANN was anticipated because they are complex algorithms. The seeming superiority of XGBoost over the ANN was not, however, anticipated. A weighted score was calculated to find the overall performance of the models and the datasets and results are shown below (Table 5.1).

Table 5.1 Performance (accuracy) of ten classification models over the seven datasets

Classifier	Change	Always Negative	Always Not Negative	Last Status	Multi Change	Good Change	Bad change
Dummy	52.6%	52.2%	94.2%	70.0%	70.0%	60.0%	60.0%
Decision Tree	58.8%	58.9%	93.9%	72.5%	68.8%	61.8%	64.1%
K-Nearest-Neighbours	62.1%	62.3%	96.9%	78.3%	75.0%	64.2%	70.7%
Gaussian Naive Bayes	64.5%	65.1%	94.6%	80.4%	74.2%	67.5%	70.5%
Support. Vector Machines	62.2%	58.9%	97.0%	80.4%	78.0%	66.7%	73.6%
Extra Trees	62.2%	61.5%	96.6%	77.6%	74.6%	64.4%	70.1%
Random Forest	62.6%	62.4%	96.8%	78.6%	75.4%	65.6%	71.1%
Logistic Regression	67.2%	68.3%	97.0%	80.8%	78.2%	70.0%	73.9%
XGBoost	67.3%	67.8%	97.0%	81.0%	78.2%	70.0%	73.8%
Neural Network (11,6,6,1)	70.0%	69.1%	96.2%	79.0%	78.6%	69.9%	69.9%

Figure 23 portrays the weighted score of the 10 models' performance over the seven datasets. As anticipated, the baseline classifier, Dummy, obtained the lowest weighted score (18.6%). Surprisingly, the Artificial Neural Network model (74.3% score) performed less well than either XGBoost (93%) or Logistic Regression (93%) classifiers. The Outstanding performance of Logistics Regression can be ascribed to the binary nature of the dependent variables. The success of XGBoost could be explained by its ability to handle imbalanced classes in the dataset (table 5.3). Also, as a strong ensemble learner, it is able to form a good decision. Its inbuilt mechanisms such as regulation term [25] contributed to its success. XGBoost and some others were selected to extend and improve evaluation by use of the areas below the Receiver Operation Character (ROC) curves (table 5.2 and figure 24).

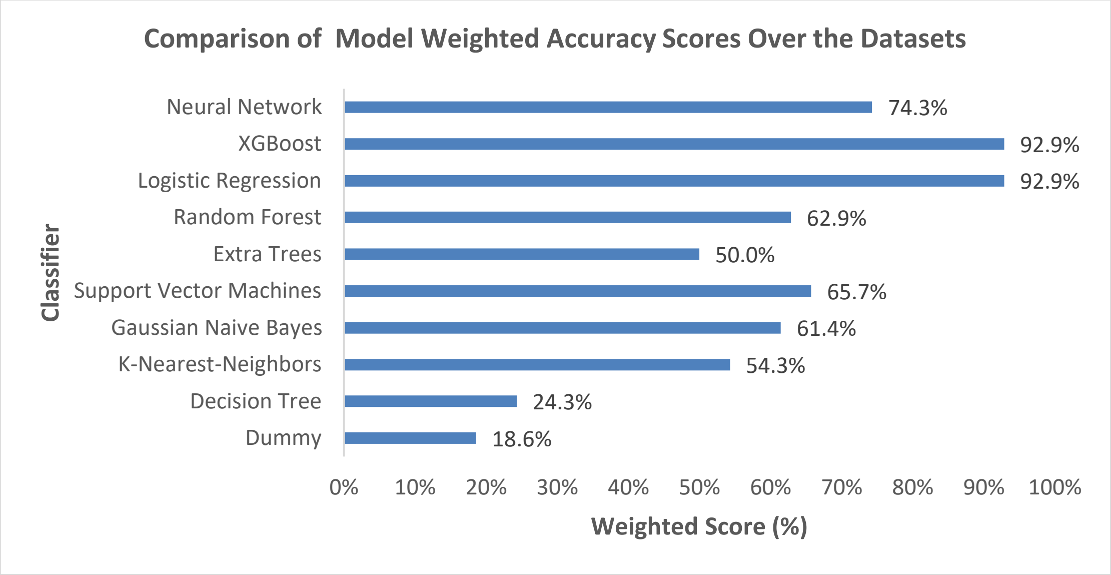


Figure 23. Comparison of the models using their weighted accuracy scores over the seven datasets

The XGBoost and Logistic Regression still out-performed the Neural Network in terms of the weighted areas under ROC curves (91.4%, 87.1% and 85.7% in that order, figure 24, next page). Given the superior performance coupled with the ability to produce ranked feature importance and a pictorial decision tree, the XGBoost was selected as the main classification model. However, performance of XGBoost will be periodically reviewed and compared with any classifier which is known to have certain comparative advantages and required characteristics.

Table 5.2 Area under ROC curve (expressed as percentage) of selection classification models over various datasets.

ROC	Always Not Negative	Herd Status	Always Negative	Change	Multi Change	Good Change	Bad change
Extra Tree	68.3%	65.0%	66.4%	62.8%	60.9%	61.8%	60.3%
Logistic Regression	59.3%	72.2%	70.9%	66.5%	65.2%	65.5%	65.2%
XGBoot	76.3%	72.0%	70.4%	66.4%	65.4%	65.7%	64.9%
Neural Network	74.4%	69.9%	70.7%	66.9%	63.5%	65.6%	64.4%

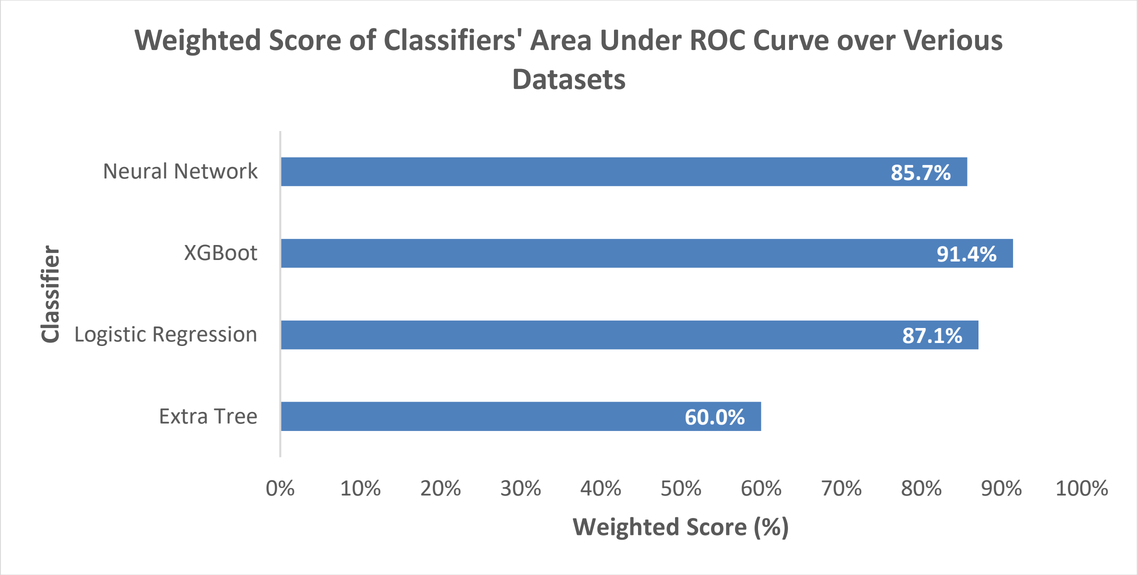


Figure 24. Weighted score of classifiers' areas under ROC curve over various datasets

The suitability of the datasets for this project was assessed in a manner similar to the models. Table 5.1 and figure 25 show their performance in terms of accuracy. The weighted score (figure 25) suggests that the *Always Not Negative* dataset was the best (100%) and the *Always Negative* dataset the worst (46%). However, in terms of the area under the ROC curve (table 5.2, previous page and figure 26, next page), the Always Not Negative, the *Always Negative* and the *Last Status* datasets showed similar results, achieving the highest weighted score of 87.5%. The classic ROC area achievement of the *Always Negative* dataset supports the fact that accuracy should not always be the priority evaluator of models. The XGBoost classifier will be used to compare the strength of these datasets (the *Always Not Negative*, the *Always Negative* and the *Last Status*) in relation to their ability to identify farm risk characteristics.

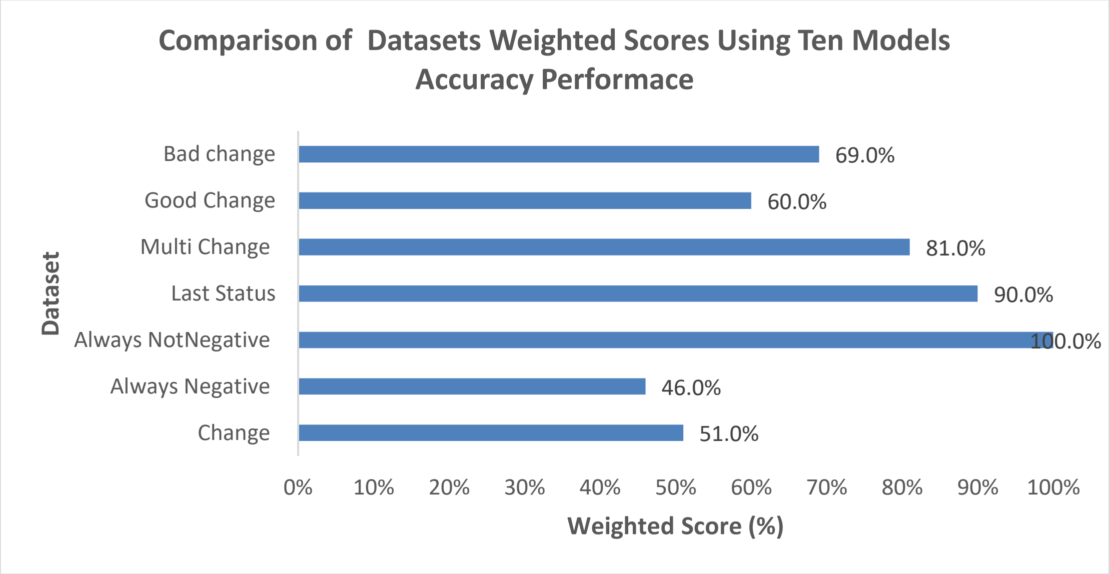


Figure 25. Comparison of the datasets using the weighted accuracy scores of the ten models

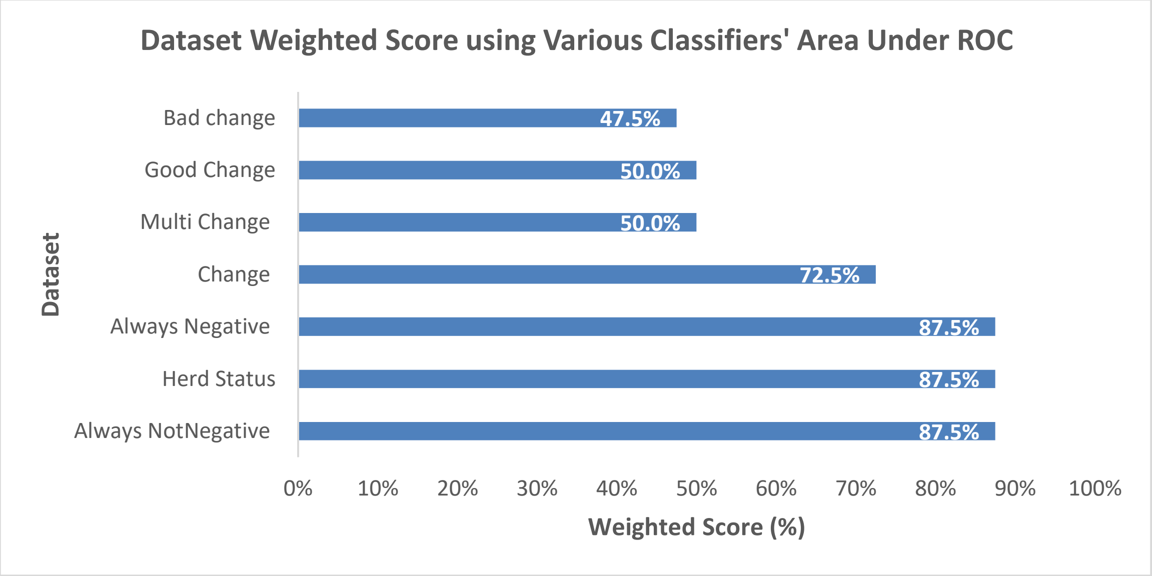


Figure 26. Weighted score of the seven datasets using various classifiers' areas under ROC curves

5.2 Probability Calibration

The XGBoost Model does not need a calibration mechanism to perform well.

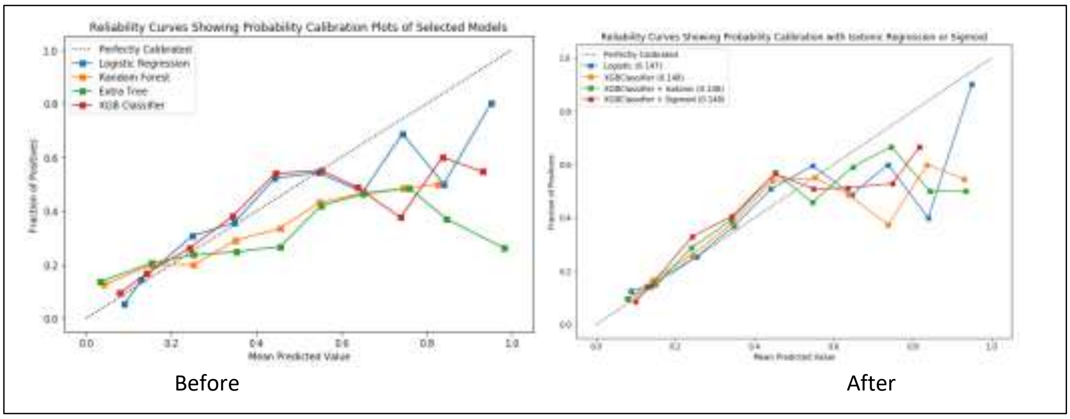
It was noted that the target variables and other features have imbalanced classes and that probability calibration was performed to improve the performance of the model by mitigation of the bias that an uneven class dataset may have on the performance of a model [21]. Since the calibration is probabilistic, the Logistic Regression model is used as a standard for comparison. Any score appreciably greater than that of Logistic Regression score would imply that an Isotonic or Sigmoid calibration may be useful. In addition, a Brier score loss can be used to evaluate this effect; smaller Brier scores are better than bigger scores [19] (chapter 2.2.4).

Table 5.3 gives the results of the effect of calibration on the XGBoost, Random Forest and Extra Trees classifiers and on the three selected datasets. The results show that while the Random Forest and Extra Trees classifiers improve with both Isotonic and Sigmoid calibration, the XGBoost model (generally) does not improve with either Isotonic or Sigmoid calibration. This means that XGBoost, the principal classifier, is able to handle the effect of the bias that may be introduced by the uneven classes in the dataset and that it does not require an Isotonic or Sigmoid calibration.

Table 5.3 Probability Calibration with Brier Score Loss

Classifier	Last Status	Always Negative	Always Not Negative
Logistic	0.147	0.212	0.032
XGBoost	0.148	0.211	0.032
XGBoost with Isotonic	0.148	0.211	0.032
XGBoost with Sigmoid	0.148	0.210	0.032
Random Forest	0.160	0.231	0.033
Random Forest with Isotonic	0.155	0.225	0.033
Random Forest with Sigmoid	0.152	0.217	0.032
Extra Tree:	0.171	0.244	0.035
Extra Tree with Isotonic:	0.157	0.221	0.033
Extra Tree with Sigmoid:	0.154	0.220	0.033

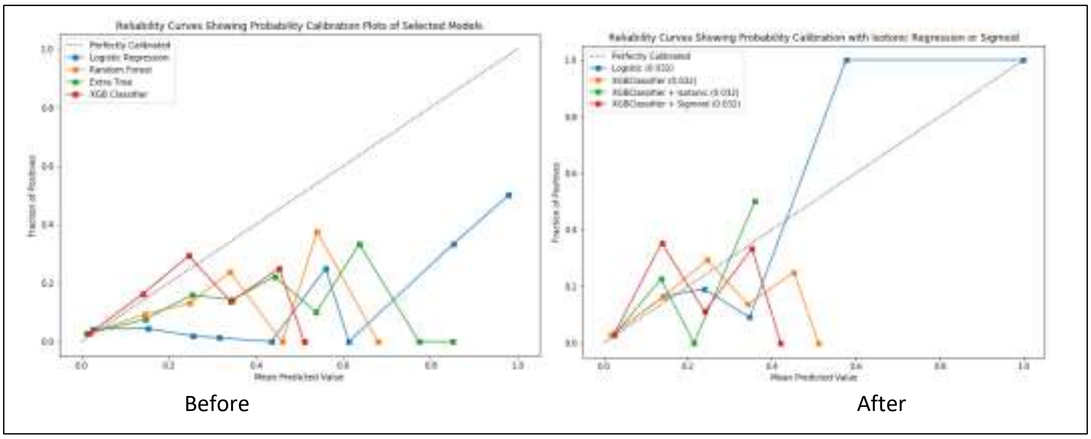
The reliability curves, notwithstanding, indicates that the datasets respond differently to calibration (figures 27, 28 & 29). These results further show the poorest dataset is the *Always Not Negative* dataset (figure 28). The reason for this could be the fact that the *Always Not Negative* dataset has the most imbalanced classes in the response variable (3% to 97%, figure 5, chapter 3.2.1.2).



Before

After

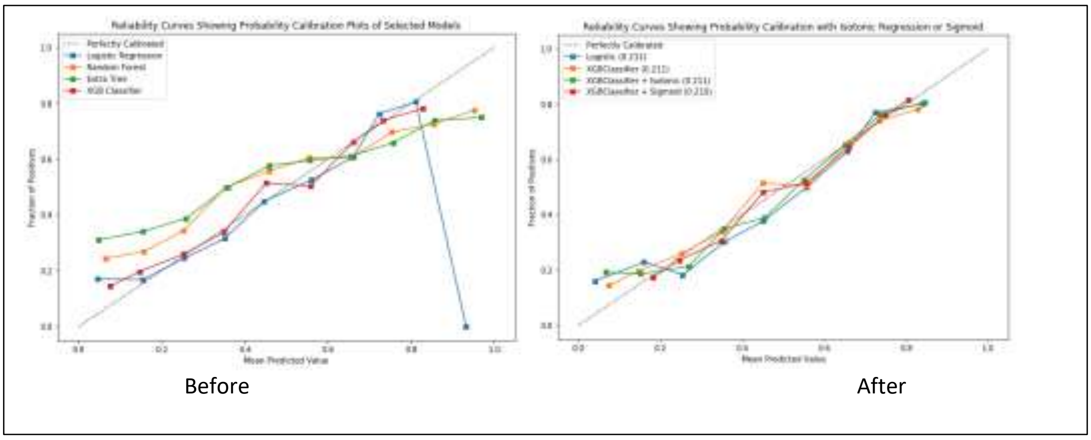
Figure 27. Reliability curve of XGBoost on Last Status dataset before and after calibration



Before

After

Figure 28. Reliability curve of XGBoost on Always Not Negative dataset before and after calibration



Before

After

Figure 29. Reliability curve of XGBoost on Always Negative dataset before and after calibration

5.3 Farm Characteristics: Risk Importance and Rank

This section shows the results of BVD risk identification and ranking. Whereas section 5.3.1 discusses the (risk) importance of farm characteristics on their individual merits, section 5.3.2 shows their relative (risk) importance. Among the optimal ranked risks is ‘number of pigs’.

5.3.1 Herd Characteristics: Individual Risk Importance

To ensure that the identification of farm risk characteristics and subsequent ranking of the main model (XGBoost) does not suffer from and is not sensitive to any inter-correlation between the input farm characteristics, a meta-algorithm (Recursive Feature Elimination (RFE)) was run on top of the XGBoost. Again, to ensure that the identification of risk characteristics and ranking were stable and reliable [28], an advanced RFE, (Recursive Feature Elimination with Cross-Validation (RFECV)) was employed.

Table 5.4 shows that XGBoost identified a minimum of 8 farm characteristics as first ranked BVD risk factors. These are Cattle under one-year old, Number of Sheep, Cattle over one-year old, In_degree (Cattle), In_degree (Farms), In_degree (Countries), Dairy and Number of Pigs.

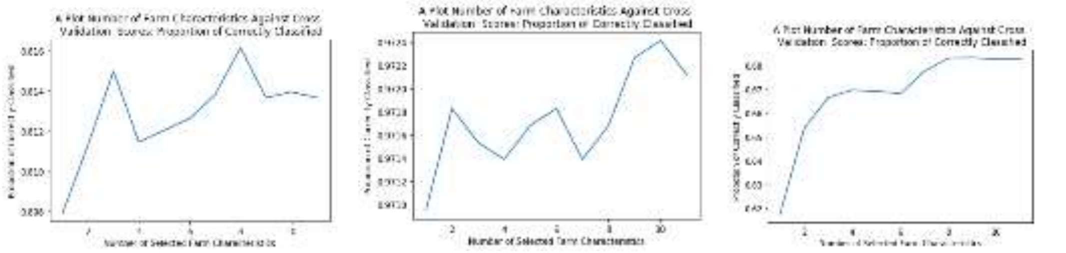
Table 5.4 Selection and ranking of BVD risk factors by XGBoost and Extra Trees Classifier embedded in Recursive Feature Elimination with Cross-Validation

	XGBoost			Extra Trees		
	Last Status	Always Not Negative	Always Negative	Last Status	Always Not Negative	Always Negative
Cattle under one-year old	1	1	1	1	1	2
Number of Sheep	1	1	1	3	4	1
Cattle over one-year old	1	1	1	4	3	4
In_degree (Cattle)	1	1	1	2	2	3
In_degree (Farms)	1	1	1	5	5	5
In_degree (Countries)	1	1	1	7	6	8
Dairy	1	1	1	6	8	6
Number of Pigs	1	1	1	8	7	7
Dual Breed	2	1	2	9	9	9
Beef	3	1	3	11	11	11
Number of Goats	4	2	1	10	10	10

Table 5.5 Average risk factor ranking

	XGBoost Average	Extra Tree Average	Combined Average
Cattle under one-year old	1.0	1.3	1.2
Number of Sheep	1.0	2.7	1.8
In_degree (Cattle)	1.0	2.3	1.7
Cattle Over One	1.0	3.7	2.3
In_degree (Farms)	1.0	5.0	3.0
In_degree (Countries)	1.0	7.0	4.0
Dairy	1.0	6.7	3.8
Number of Pigs	1.0	7.3	4.2
Dual Breed	1.7	9.0	5.3
Number of Goats	2.3	10.0	6.2
Beef	2.3	11.0	6.7

Figure 30 plots the number of risk factor against share of correctly classified farms. This shows that the *Always Not Negative* dataset achieved the highest correctly classified farms.



Last Status: 8 optimal

Always NotNegative 10 optimal

Always Negative: 9 optimal

Figure 30. A plot of the number of farm characteristics against proportion of correctly classified by XGBoost on the three datasets

5.3.1.1 Other Species

The rankings of risk factors were averaged on all the datasets for each model (table 5.5). These averages suggest that Dual Breed may be equally important, but that Beef and Number of Goats are less important to the BVD status of a farm (tables 5.6 &5.7 and figure 31). Tables 5.6, 5.7 and figure 31 show the rankings of Extra Trees and XGBoost algorithms without the influence of the meta-algorithm (RFECV) over the three datasets. These lend support to the fact that Beef and Number of Goats are not important BVD risk factors as these farm characteristics always ranked as last and next to last.

The elimination / poor ranking of Beef and Number of Goats (table 5.4 to 5.7, figure 31) were anticipated. Statistically, their results required their exclusion from the dataset. Beef was the majority group (98.3%) and Number of Goats was the minority group (0.9%). These behaviours would normally contribute nothing to the learning [18]. However, they were included as a control and it can therefore be concluded that the XGBoost algorithm is reliable.

However, whilst Number of Pigs had minority distribution (5.8%), XGBoost ranked it among the optimal classes on all the datasets. Beef, Number of Goats and Number of Pigs variables were kept in the datasets for the reason that if an algorithm identified and ranked any of them as being an important risk factor, we can conclude either:

1. that the model could be suspicious, or
2. that the risk posed by that feature is very important, especially if the algorithm is proven to be reliable. It may have a strong correlation with a farm's BVD status.

Since the Number of Goats variable was eliminated and the Number of Pigs variable was also picked as an important BVD risk factor, this when coupled with the fact that the prevalence of BVD Virus natural infection in pigs is on the increase [5], [9] – [12], the question of the possible importance of pigs is raised, even though there is no report that suggests that pigs can infect cattle with the BVD Virus. Again, figures 34 and 40 further show that the presence of a pig on a cattle farm matters.

5.3.1.2 Interconnectivity

Cattle farms in Scotland are highly open (page 22 and table 3.4). An average farm was influenced (via cattle movement alone) by at least seven other farms a year. A farm may be connected (in-degree) to as many as 2,694 (31.6% of all farms) other farms per year in Scotland alone. The total of such unidirectional farm-to-farm connections was 60,495 in 2017. Again, an average farm brings in over 30 cattle per year, but a farm may bring in as many as 6,382 head of cattle from other farms. Furthermore, of the 822,930 total cattle in 2017, 258,301 (31.4%), were involved in unidirectional movement to the farms under consideration. Similarly, a farm in Scotland may be connected to as many as ten countries outside the United Kingdom in a year. The total number of such connections was 1967 in 2017. Such a high level of '*openness*' and interconnectivity increase the opportunity for the spread of BVDV. This is because the higher the number of connections a farm has the more susceptible it is to infectious diseases [13] - 16]. This assertion is supported by the findings of this study. All the network metrics included in this project were considered optimal risk factors on their individual merits (tables 5.4 & 5.5) as well as their presence among the top-ranking risks in terms of relative importance (tables 5.6 & 5.7 and figure31)

5.3.2 Herd Characteristics: Relative Risk Importance

This section shows the relative risk importance of farm characteristics.

Table 5.6 Risk factors rankings by XGBoost without the Recursive Feature Elimination algorithm

	Last Status			Always Not Negative			Always Negative		
Risk	Rank	F1	Importance	Rank	F1	Importance	Rank	F1	Importance
Cattle Under 1	1	187	0.287	1	188	0.307	1	171	0.279
Num. of Sheep	2	156	0.239	2	132	0.119	2	119	0.194
Indeg (Cattle)	3	75	0.115	3	76	0.124	4	59	0.096
Cattle Over 1	4	72	0.110	4	73	0.090	3	95	0.155
Indeg. (Farms)	5	54	0.081	5	55	0.051	5	59	0.096
Indeg (Countr.)	6	33	0.051	6	31	0.021	6	41	0.067
Number of Pigs	7	30	0.046	8	17	0.003	8	26	0.042
Dairy	8	23	0.035	7	24	0.039	7	28	0.046
Dual Breed	9	15	0.023	10	2	0.002	11	0	0.000
Beef	10	5	0.008	9	13	0.216	10	2	0.003
Num. of Goats	11	2	0.003	11	1	0.028	9	14	0.023

Table 5.7 Risk factors rankings by Extra Trees without the Recursive Feature Elimination algorithm

	Last Status		Always Not Negative		Always Negative	
Risk	Rank	Importance	Rank	Importance	Rank	Importance
Cattle Under One	1	0.276	1	0.263	1	0.279
Number of Sheep	2	0.210	2	0.163	2	0.226
In-degree (Cattle)	3	0.136	3	0.162	3	0.125
Cattle Over One	4	0.129	4	0.154	4	0.119
In-degree (Farms)	5	0.108	5	0.135	5	0.094
Dairy	6	0.063	8	0.020	6	0.082
In-degree (Countries)	7	0.033	6	0.055	8	0.026
Number of Pigs	8	0.028	7	0.027	7	0.031
Dual Breed	9	0.010	9	0.013	9	0.009
Number of Goats	10	0.004	10	0.006	10	0.007
Beef	11	0.003	11	0.001	11	0.003

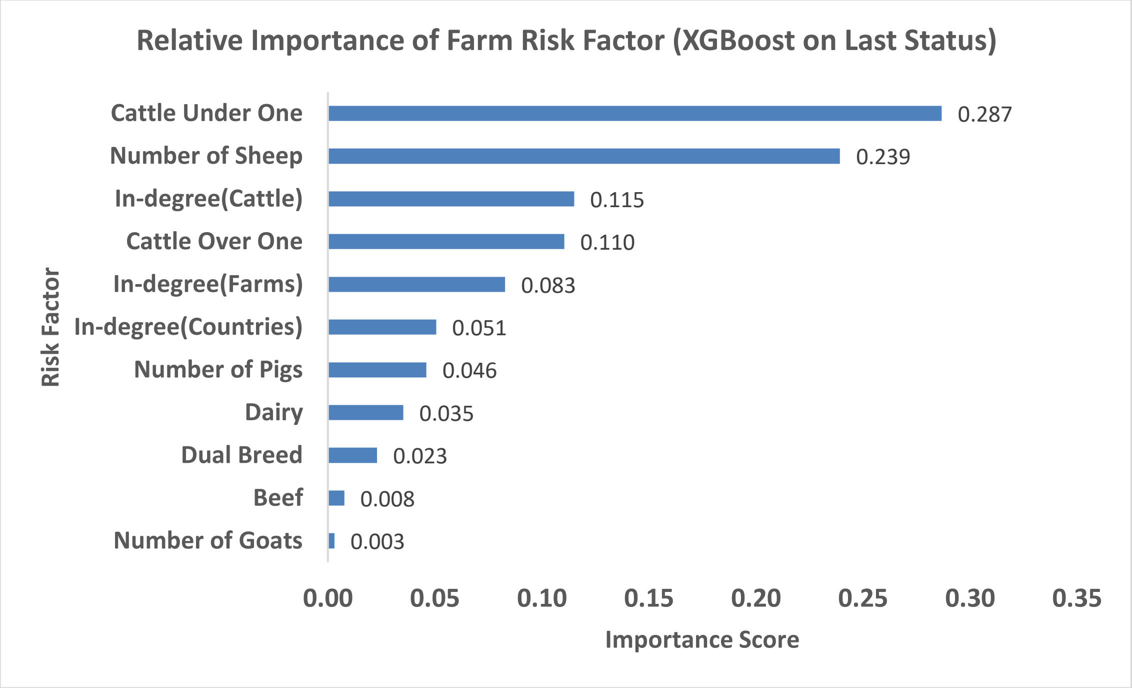


Figure 31. Risk factors rankings by XGBoost without the Recursive Feature Elimination algorithm on the Last Status dataset

5.3.2.1 Calves under One Year Old

This population group is among those optimally ranked (tables 5.4 & 5.5) by XGBoost embedded in the meta-algorithm (RFECV) which ensured no inter-correlation between the potential risk factors. Both XGBoost and Extra Trees models assessing the relative importance of the potential risk factors, consistently ranked this farm characteristic as having the most relative importance (tables 5.6 & 5.7 and figure 31). The number of calves under one year old is most likely to correlate with the number of persistently infected (PI) cattle on a farm than any other factor considered in this study. Since most PIs do not live to year two [5], it can be assumed that the majority of them are under one year old. Given that PIs are recognised as the most important group in term of BVD transmission [4] – [7], [9], it was expected that they would rank as the highest risk farm factor. It should be noted that of the number of cattle considered in this study, over 73 percent calves were under one-year old (table 3.4) and that the average herd in Scotland has over 71 percent in this group.

5.4 Visualising the Models

This section provides graphic representations of the models (part of the fourth objective of this project) in the form of decision trees. It also evaluates their strengths and weaknesses (based on datasets of this study) of the graphic models given by the Decision Tree, XGBoost and J48 Classifiers with emphasis on their usefulness for explaining their decisions to non-data science experts. A mention is also made on how these experts responded to these models. Since the *Last Status* dataset is the "general purpose" dataset of the three, only its graphics will be presented here. Others are shown appendix 1

5.4.1 The Decision Tree Model

This model is the most informative (figure 32) as it provides the most important predictor variable (risk factor) in term of information gain (entropy), the Gini coefficient, number of samples at each node, the values considered and the final decision (class of the response variable). The Statistician related well to this model due to its detailed information. For instance, following the left-most logic, one can interpret the model as:

> If a herd has Dairy breed cattle and has more than 163 heads of calves under one-year old and further has more than 1006 heads of sheep, then it is predicted to have a BVD negative status with 72.5% certainty (accuracy).

However, the pruned tree contains only the Negative class at both the tree and the leaf nodes. It is not able to predict any Not Negative class. This makes it unhelpful to this project. Given such a level of details, the Decision Tree model could have been recommended but for its inability to predict both classes of the response variable.

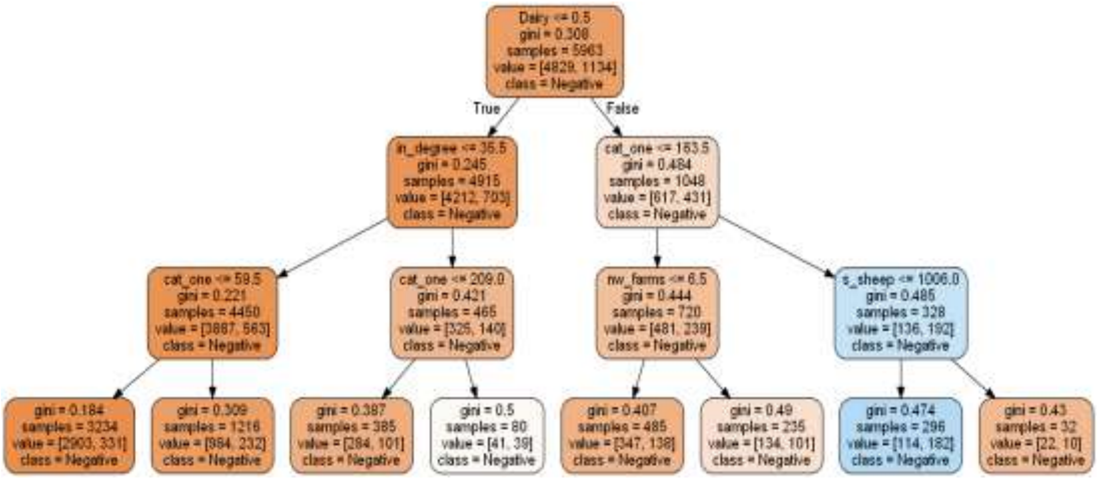


Figure 32. Graphical model of the Decision Tree Classifier on the Last Statue dataset

5.4.2 The XGBoost Model

This model (figure 33) is less informative than that of the Decision Tree. It gives logical decisions in Boolean forms. Each leaf node shows the relative importance measure of that decision. The decision following the upper-most 'blue' line can be interpreted as:

> If f6 (Dairy, box 5.1) is present, and f2 (the total number of cattle brought to the farm) is more than 16 heads of cattle, and there are more than 28 f0s (heads of calves less than one year old) then the BVD risk is the highest (relative risk importance of 0.112).

It can be realised that the XGBoost model graphic uses positional indices (not the names) of the predictor variable. The chronological list (box 5.1) of the input variables must be available in order to interpret the model. Again, the model does not give a prediction of the BVD status (Negative or Not Negative). It does, however, predict the relative risk importance of the logical decision. This was discussed in previous sections.

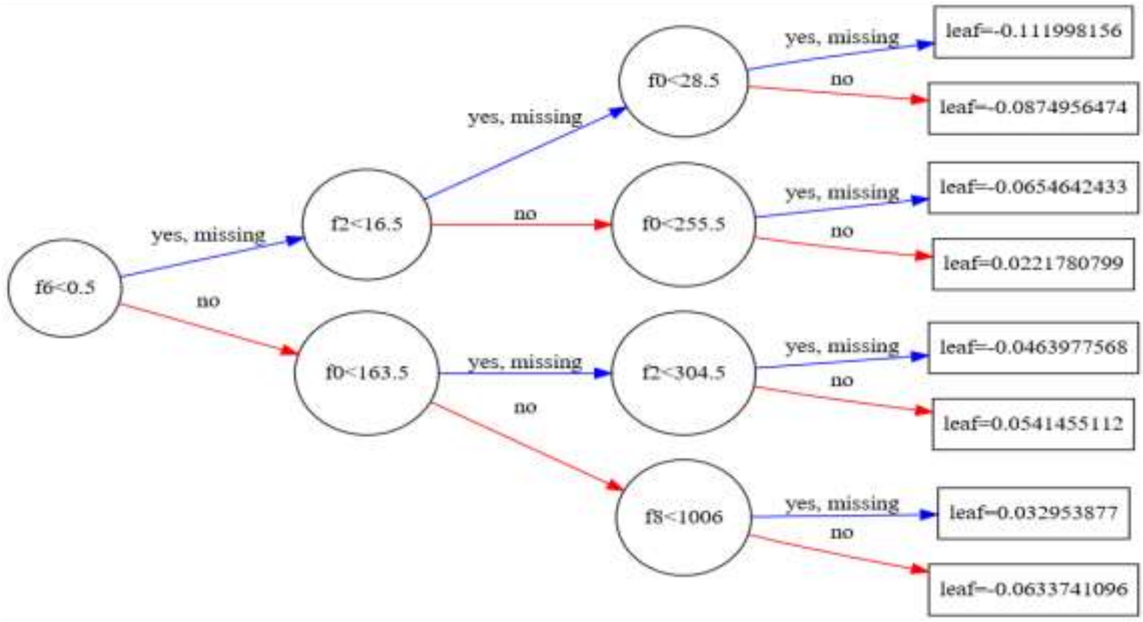


Figure 33. Graphical model of the XGBoost Classifier on the Last Status dataset.

Box 5.1 Positional indexes of the farm characteristics

0='cat_one', 1='cat_oneplus', 2='in_degree', 3='nw_farms', 4='nw_importcountry', 5='Beef', 6='Dairy', 7='Dual Breed', 8='s_sheep', 9='s_goat', 10='s_pig'

5.4.3 The J48 Model

This model (figure 34), in common with the others is intuitive and easily legible.

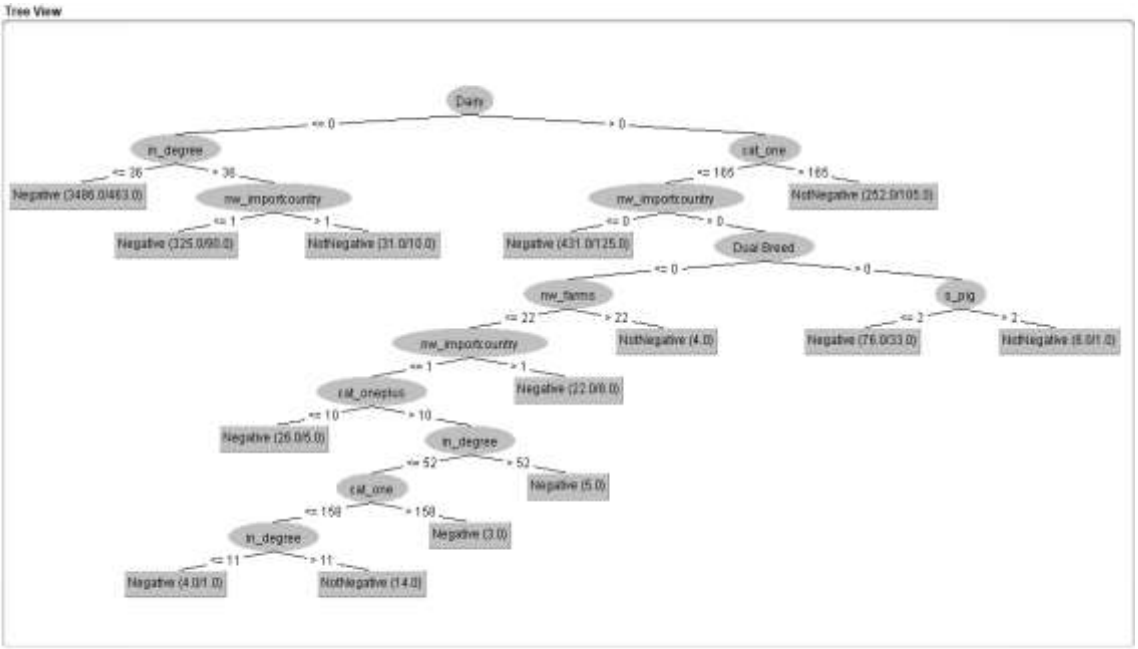


Figure 34. Graphical model of the J48 Classifier on the Last Status dataset

Although some of the technical information is hidden from the reader (unlike the Decision Tree) and it does not show the relative importance measure of the risk (as does XGBoost), it is able to capture all the classes of the response variable (BVD Negative or Not Negative) at its decision node (the leaf). It is the one that can be most easily interpreted by a non-technical reader. This could be due to its simple manner of presenting the most important information to the reader. With this model, it is possible to conclude that, if certain logical conditions are met, a farm is likely to be classified as having, for example, a Negative BVD status. One striking feature that makes this model easy to follow is the information at each edge (relationship) which tells the value at which the instances of a variable decision were separated. For instance, following the left-most logic, one can interpret the model as:

> if a herd has Dairy breed cattle and has more than 165 heads of calves under one-year old, then it is predicted to be exposed to the BVD virus. On the other hand, if a farm has less than 165 heads of calves under one-year old and it has ever imported cattle from outside the United Kingdom, and further keeps Dual Breed cattle and has more than two heads of pigs, then it is likely to be classified as BVD infected with 81% certainty (accuracy).

Of the three graphic models and based on the datasets, the J48 is best suited to the goals of this project as it was tested to be the one whose decisions were easy to read and understand by a non-data scientist professional. During my interactions with the experts (non-data scientist) at the Epidemiology Research Unit of the Scotland's Rural College (SRUC), preference was showed to this model. This was further confirmed during the present of this study to a wider audience of the SRUC staff (comprising a range of different experts) at Inverness as all the issues that were raised made reference to the J48 model.

5.4.4 Dairy Cattle

In spite of their relative strengths and weaknesses, it can be seen that Dairy breed sits on top of all the three models. This shows the importance of the presence or absence of Dairy cattle on a farm. It also confirms the fact that Dairy cattle are more exposed to BVDV than other breed purposes in Scotland. Whereas Not Negative Beef herds rate stood at 12%, the rate for Dairy has reduced from 50% to 39% [5].

Despite the length of time that the implementation of the eradication policy has been in effect, the herd exposure to BVD rate in Scotland is still high (19.5 %, figure 4 at the end of 2017) although there has been a general decline in the herd status levels from 2012 to 2017 (figure 3).

6 Conclusion

The salient findings of this study together with some recommendations are presented in this chapter.

6.1 Summary

Given the length of time of the implementation of the eradication policy, the herd exposure to BVD rate in Scotland is still high (19.5 %, figure 4 at the end of 2017) although there has been a general improvement in the herd status levels from 2012 to 2017 (figure 3).

Furthermore, this study has shown that Cattle farms in Scotland are highly open. Such a high level of openness and interconnectivity increases opportunities for the spread of the BVDV. Open farms are likely to to be BVD infected. Specifically, the more a farm received cattle from outside, the higher the risk to BVD [13] – [16].

Also, Dairy cattle are confirmed to be more exposed to the BVDV than other breed purposes in Scotland. This means that farms with dairy cattle are more at risk than those without them.

It was further discovered that herds with more calves under one-year of age are likely to be have a BVD Not Negative status than farms with fewer cattle under one-year old. This is because calves under one-year old are ranked the highest risk of farm factors. It should be noted, however, that of the number of cattle considered in this study, over 73 percent were calves under one-year old.

The unexpected revelation of this study is that farms with more than two pigs are classified as being BVD infected. Although no report has been seen that alludes that pigs can infect cattle with the BVD Virus, given the increased incidences of natural BVDV infection in pigs [6], [9] – [12], it becomes difficult to ignore the possible importance of pigs to a cattle herd's BVD status.

In terms of the machine learning techniques, the eXtreme Gradient Boost (XGBoost) emerged as the best model on all the datasets and it passed all tests performed on it. Based on this study, it is confirmed and concluded that the XGBoost model is robust, accurate [25] and reliable. However, the Weka's J48 Decision Tree graphic was found to be more understandable to a non-data scientist than the XGBoost graphics, even though it could not give a graphic for the *Always Not Negative* dataset. All the other profeesionals who asked questions or raised issues during the presenting of this study made reference to only the J48 visual model. This shows that they understoond the J48 model more than the other models. Therefore, the hypothseis that some models are more useful for explaining their decisions to a non-data scientist than other is valid and acceptable based on the conditions of this study. Unexpectedly, Logistic Regression classifier outperformed the Artificial Neural Network on these datasets. As it is a simple classifier, I would recommend the Logistic Regression model in future work on these and similar datasets provided that interpetation is not a prime goal.

Whereas the *Last Status* dataset captures both farms with BVD Negative and BVD Not Negative herd statuses, the *Always Not Negative* dataset and *Always Negative* dataset specialise in capturing BVD infected herds (sensitivity) and BVD free farms (specificity) respectively. Given its general 'purpose' quality together with its accuracy (80%) and high specificity (96.8%), the *Last Status* dataset is recommended for future studies. However, if future projects aim to maximise sensitivity (BVD infected farms), the best dataset in such circumstances would be the *Always Negative* provided a 'closed world'[17] is assumed.

Finally, given the experimental approach, together with the rigorous model testing, adopted by this study, I foresee no difficulty in either replicating this project on different dataset or tailoring it to a different machine learning (classification) project in any field of study. The data processing algorithms are wide-ranging enough to deal with any common data anomalies and make them acceptable to the common models. To match well with the algorithms, it is recommended (though optional) that future study have the datasets structured in a 2-dimensional form (such as a csv or a Pandas DataFrame form). However, the data must be presented to the Machine Learning algorithm in a matrix form by converting all nominal data to numeric data. Also, care must be taking to avoid a dummy variable trap. It should be noted that the templates are not to be followed blindly. There are certain (tailored) decisions that needs to be made. Details of these are presented in the codes as comments. For example, whereas conversion of all categorical data into numeric data are a prerequisite, scaling of data depends on the goal of the project.

During the presentation of the results, the XGBoost accuracy, selection and ranking of the risk factors were well accepted. However, the J48 visual model was found to be more understandable to a non-data scientist than that of the XGBoost. The role of the network metrics was enlightening but the importance of dairy cattle was found to be no new -information. However, the role of pigs provoked controversy that can only be resolved, in my view, by a further investigation.

6.2 Recommendations

Based on the above conclusions, it is recommended that:

1. cattle movement policies (should) be strengthened and rigorously enforced to protect farms from outside influence;
2. more attention (can) be given to farms with dairy breeds and more calves under one-year old; and
3. further study be undertaken to explore why pigs were identified as an important risk factor.

6.3 Future Work

Betweenness is an important network metric that can be a potential BVD risk factor. Due to time constraints, this was not considered in this study, but the scope of this study could beneficially be extended to include this metric.

Future study could also usefully investigate the role of pigs and farm BVD status.

References

[1] Scotland’s Rural College (SRUC 2018) Epidemiology Research Unit. [online], available: https://www.sruc.ac.uk/info/120249/epidemiology_research_unit. [viewed 31.05.2018].

[2] A. McAfee and E. Brynjolfsson, Big data: the management revolution. Harv Bus Rev (2012) 90:61–8. ***in*** K. VanderWaal, R. B. Morrison, C. Neuhauser, C Vilalta and A. M. Perez, Translating Big Data into Smart Data for Veterinary Epidemiology. Front. Vet. Sci. 4:110. 2017.

[3] K. VanderWaal, R. B. Morrison, C. Neuhauser, C Vilalta and A. M. Perez, Translating Big Data into Smart Data for Veterinary Epidemiology. Front. Vet. Sci. 4:110. 2017.

[4] M. D. Fray, D. J. Paton and S. Alenius, The effects of bovine viral diarrhoea virus on cattle reproduction in relation to disease control. Anim Reprod Sci 60–61 2000 615–627, Elsevier 2000. www.elsevier.comrlocateranireprosci. Also available: https://doi.org/10.1016/S0378-4320(00)00082-8. July 2018.

[5] Vets’ Guide to: Enhanced BVD Screening BVD Eradication Scheme Phase 4. The Scottish BVD Eradication Scheme: A Guide to Enhanced Annual Screening for Vets in Practice June 2015. APS Group Scotland. ISBN: 978-1-78544-363-3.

[6] Epidemiology, Population health and Infectious disease Control (EPIC 2017) - The Scottish Government's Centre of Expertise on Animal Disease Outbreaks. [viewed 31.05.2018]. Available from: http://www.epicscotland.org. Also EPIC Home Page , http://www.epicscotland.org/our-research/case-studies/bvd-eradication-programme-in-scotland/

[7] Scottish Government Home Page https://www.gov.scot/Topics/farmingrural/Agriculture/animal-welfare/Diseases/disease/bvd/eradication, August 2018.

[8] BVD Order 2013. The Bovine Viral Diarrhoea (Scotland) Order 2013, SCOTTISH STATUTARY INSTRUMENT, 2013 No. 3, ANIMALS, ANIMAL HEALTH, http://www.legislation.gov.uk/ssi/2013/3/made, August 2018

[9] J. Tao, J. Liao, Y. Wang, X. Zhang, J. Wang and G. Zhu, Bovine viral diarrhoea virus (BVDV) infections in pigs. Vet. Microbiol. Vol. 165, Iss 3–4, 30 August 2013, Pages 185-189. https://doi.org/10.1016/j.vetmic. 2013.03.010 Vet. Microbiol [26 Mar 2013, 165(3-4):185-189] DOI: 10.1016/j.vetmic.2013.03.010. https://europepmc.org/abstract/MED/23587625

[10] C. Terpstra and G. Wensvoort, Natural infections of pigs with bovine viral diarrhoea virus associated with signs resembling swine fever. Res. Vet. Sci. [01 Sep 1988, 45(2):137-142], also available: https://europepmc.org/abstract/med/2848298, 1988.

[11] G. Wensvoort and C. Terpstra, Bovine viral diarrhoea virus infections in piglets born to sows vaccinated against swine fever with contaminated vaccine. Res. Vet. Sci. [01 Sep 1988, 45(2):143-148], also available: https://europepmc.org/abstract/MED/2848299 1988.

[12] H. M. S. Almeida, I. R. H. Gattodos, A. C. R. Santos, D. A. Pereira, K. A. Nascimento and T. G. Baraldi *et al.*, Bovine viral diarrhea virus infections in pigs: why is this situation important for Brazilian herds?. Arq. Inst. Biol. [online]. 2017, vol.84 [cited 2018-08-16], e0322016. Available from: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1808-16572017000100406&lng=en&nrm=iso>. Epub Feb 01, 2018. ISSN 1808-1657. http://dx.doi.org/10.1590/1808-1657000322016.

[13] E. Stattner and N. Vidot. (2011). Social network analysis in epidemiology: Current trends and perspectives. Proceedings - International Conference on Research Challenges in Information Science. 1 - 11. 10.1109/RCIS.2011.6006866.

[14]-Barabasi. A Network Science Book~~.~~ http://barabasi.com/networksciencebook/ [viewed 31.06.2018]

[15] M. E. J. Newman, The structure and function of complex networks. https://arxiv.org/pdf/cond-mat/0303516.pdf, 2003 [viewed 01.07.2018]

[16] M. E. J. Newman, Networks: An Introduction, Oxford University Press, Oxford, 2010 [viewed 01.07.2018]

[17] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques. Third Edition. AMSTERDAM, 2011 Elsevier (Morgan Kaufmann Publishers is an imprint of Elsevier)

[18] K Swingler, 2018 Machine Learning (Unpublished source, lecture material)

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion and O. Grisel, *et al*., Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011. [online], available: http://scikit-learn.org/stable/model_selection.html#model-selection, [viewed 22.06.2018]

[20] T. Hastie, R. Tibshirani and J. Friedman, 2008. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, Springer Series in Statistics. Springer

[21] A. Dal Pozzolo, O. Caelen, R.A. Johnson and Bontempi, G. Calibrating Probability with Undersampling for Unbalanced Classification. 2015 IEEE Symposium Series on Computational Intelligence

[22] J. H. Metzen, and A. Gramfort, M. Blondel, and B. Kegl, 2015-04-14 python classification [online] machine-learning. https://jmetzen.github.io/2015-04-14/calibration.html. Also on http://scikit-learn.org/stable/modules/calibration.html#calibration [viewed 24.07.2018]

[23] P. M. Granitto, C. Furlanello, F. Biasioli. and F. Gasperi, 2006. Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. Chemometrics and Intelligent Laboratory Systems. Elsevier. Vol. 83, Issue 2, 15 September 2006, Pages 83-90. https://doi.org/10.1016/j.chemolab.2006.01.007.

[24] A. Niculescu-Mizil and R. Caruana, Predicting Good Probabilities with Supervised Learning, ICML 2005 ***in*** Pedregosa et al., Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.

[25] XGBoost, eXtreme Gradient Boost Home Page https://xgboost.readthedocs.io/en/latest/tutorials/model.html, June 2018

[26] S. García, A. Fernández, J. Luengo and F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput - A Fusion of Foundations, Methodologies and Applications 2009. Vol. 13 Iss. 10, April 2009 Pages 959-977. Springer-Verlag Berlin. Published online: 20 December 2008 https://doi.org/10.1007/s00500-008-0392-y

[27] I. Hisao and M. Yusuke, Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. INT J APPROX REASON. Vol. 44, Iss. 1, January 2007, Pages 4-31. https://doi.org/10.1016/j.ijar.2006.01.004. Also available at: https://www.sciencedirect.com/science/article/pii/S0888613X06000405 [viewed 01.08.2018]

[28] S. Abe, (2010) Feature Selection and Extraction. In: Support Vector Machines for Pattern Classification. Advances in Pattern Recognition. Springer, London. DOI https://doi.org/10.1007/978-1-84996-098-4_7

[29] I. Guyon, J. Weston, S. Barnhill. and V. Mach. Vapnik, Learn., 46 (2002), pp. 389-422 ***in*** P. M. Granitto, C. Furlanello, F. Biasioli and F. Gasperi, 2006. Recursive feature elimination

with random forest for PTR-MS analysis of agroindustrial products. Chemometrics and Intelligent Laboratory Systems. Elsevier. Vol. 83, Iss. 2, 15 September 2006, Pages 83-90. https://doi.org/10.1016/j.chemo lab.2006.01.007.

[30] J. H. Friedman and J. J. Meulman, Multiple additive regression trees with application in epidemiology. Statistics in Medicine, 22, 1365–1381. 2003 ***In*** Elith et al. 2008, A working guide to boosted regression trees Journal of Animal Ecology 2008, 77, 802–813. (Pages 808). British Ecological Society 2008

[31] J. Elith, J. R. Leathwick and T. Hastie, A working guide to boosted regression trees, J Anim Ecol July 2008, 77, 802–813. (Pages 808). British Ecological Society 2008

[32] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", 22nd SIGKDD Conference on Knowledge Discovery and Data Mining, 2016, https://arxiv.org/abs/1603. 02754] https://cran.r-project.org/web/packages/xgboost/xgboost.pdf [viewed 02.08.2018]

[33] D. M. W. Powers, Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. JMLT [online]. vol. 2, iss 1 , pp-37-63, 2011 (page 44) Available http://www.bioinfo.in/contents.php?id=51. ISSN: 2229-3981 & ISSN: 2229-399X,

[34] Wikipedia. Precision and recall, https://en.wikipedia.org/wiki/Precision_and_recall. This page was last edited on 3 August 2018, View August 2018

[35] Pandas, Pandas Home Page https://pandas.pydata.org/pandas-docs/stable/dsintro.html [View 02/09/2018]

[36] S. E. Buttrey, Matrices, [online] http://faculty.nps.edu/sebuttre/home/R/matrices.html Last updated: Apr. 26, 2016. Viewed 02.09.2018

[37] F. O. Mensah, Using Data Mining techniques to build predictive model to foretell which new customer is likely to pay back a loan. Assignment submitted to University of Stirling, MSc Big Data, ITNPBD6 – Data Analytics Assignment 2018.

Appendix 1. Graphical Model of the Decision Tree, eXtreme Gradient Boost and J48 Classifiers on the Always Negative and Always Not Negative Datasets

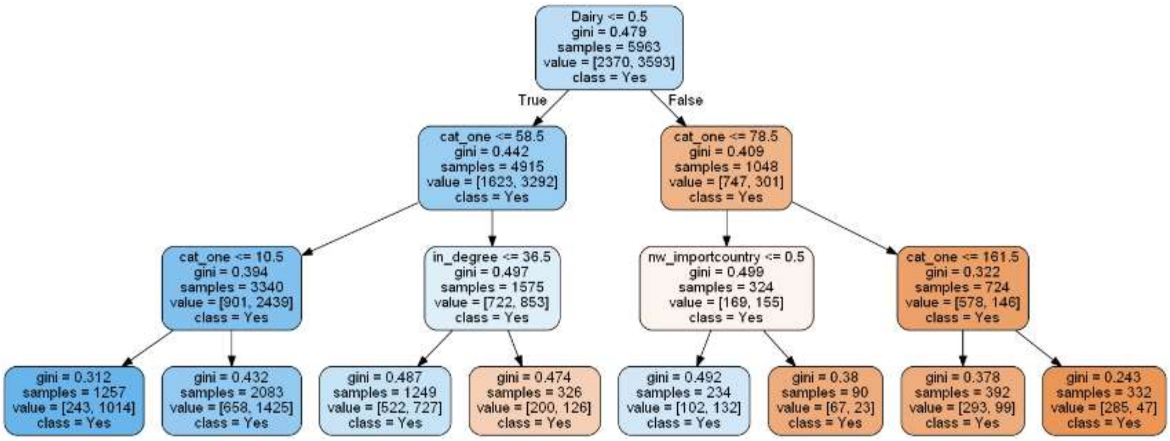


Figure 35. Graphical model of the Decision Tree Classifier on the Always Negative dataset

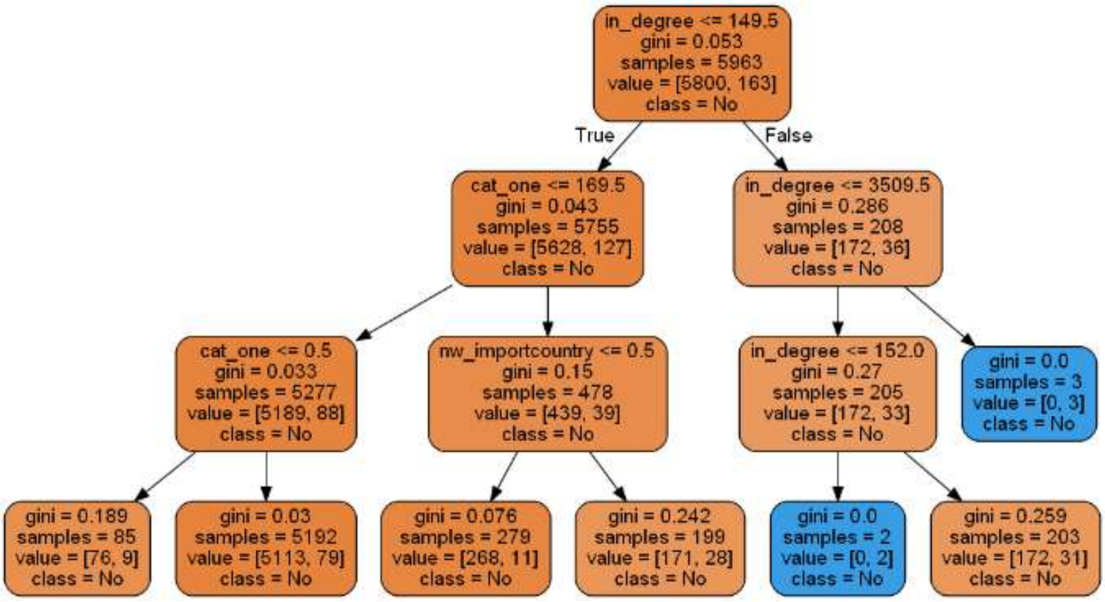


Figure 36. Graphical model of the Decision Tree Classifier on the Always Not Negative dataset

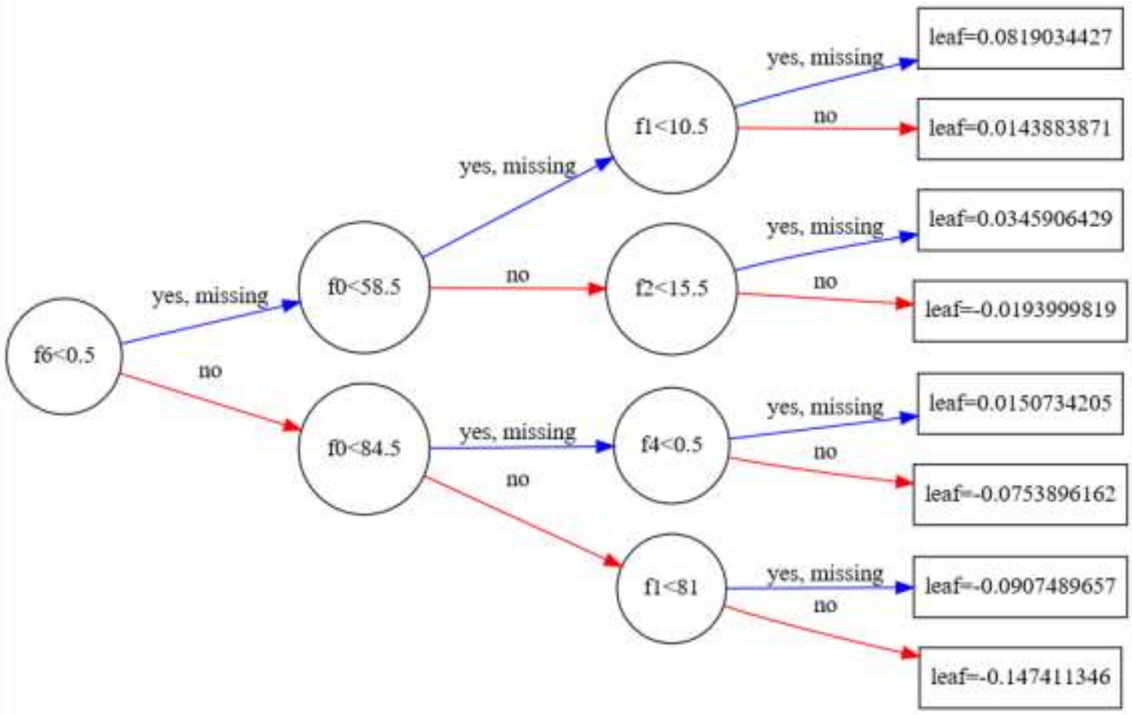


Figure 37. Graphical model of the XGBoost Classifier on the Always Negative dataset

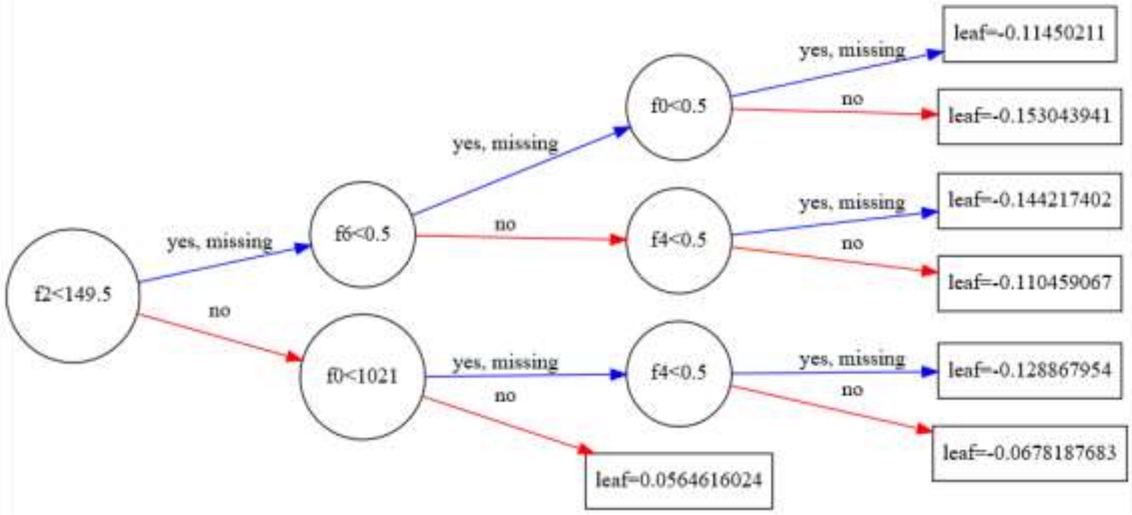


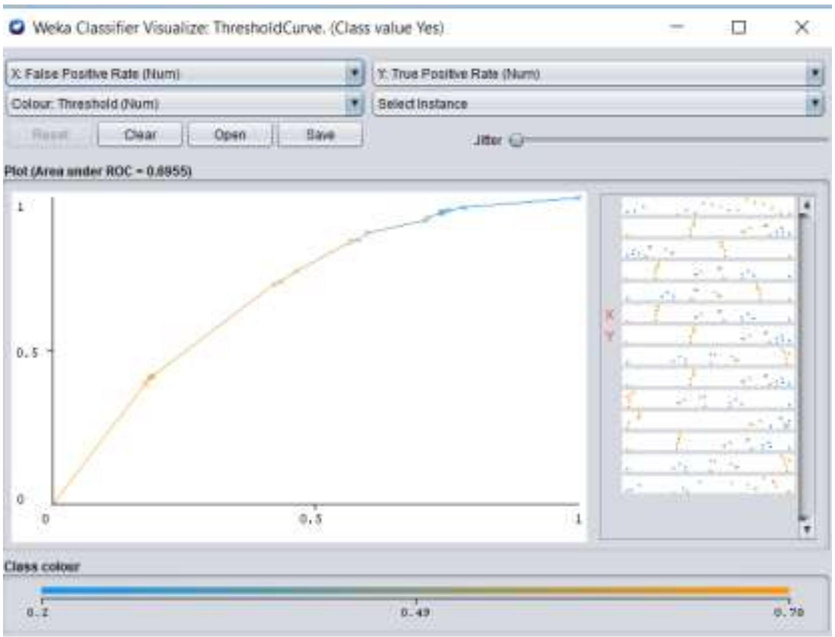
Figure 38. Graphical model of the XGBoost Classifier on the Always Not Negative dataset

==

Performance Metrics J48 Classifier on the Always Negative Dataset (minNumObj 10)

```
=== Re-evaluation on test set ===
=== Summary ===
Correctly Classified Instances       2629            68.5707 %
=== Detailed Accuracy By Class ===
TP Rate  FP Rate  Precision  Recall  F-Measure  MCC    ROC Area  PRC Area  Class
  0.854   0.566    0.694      0.854    0.765     0.320   0.696     0.727    Yes
  0.434   0.146    0.664      0.434    0.525     0.320   0.696     0.604    No
  0.686   0.398    0.682      0.686    0.669     0.320   0.696     0.678    Weighted Avg
```

```
=== Confusion Matrix ===
    A      b   <-- classified as
 1964  337 |    a = Yes
  868  665 |    b = No
```



Weka Classifier Visualize: ThresholdCurve. (Class value Yes)

X: False Positive Rate (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

Select Instance

Plot (Area under ROC = 0.6955)

Figure 39. ***Plot of area under the ROC curve of the J48 model on the Always Not Negative dataset***

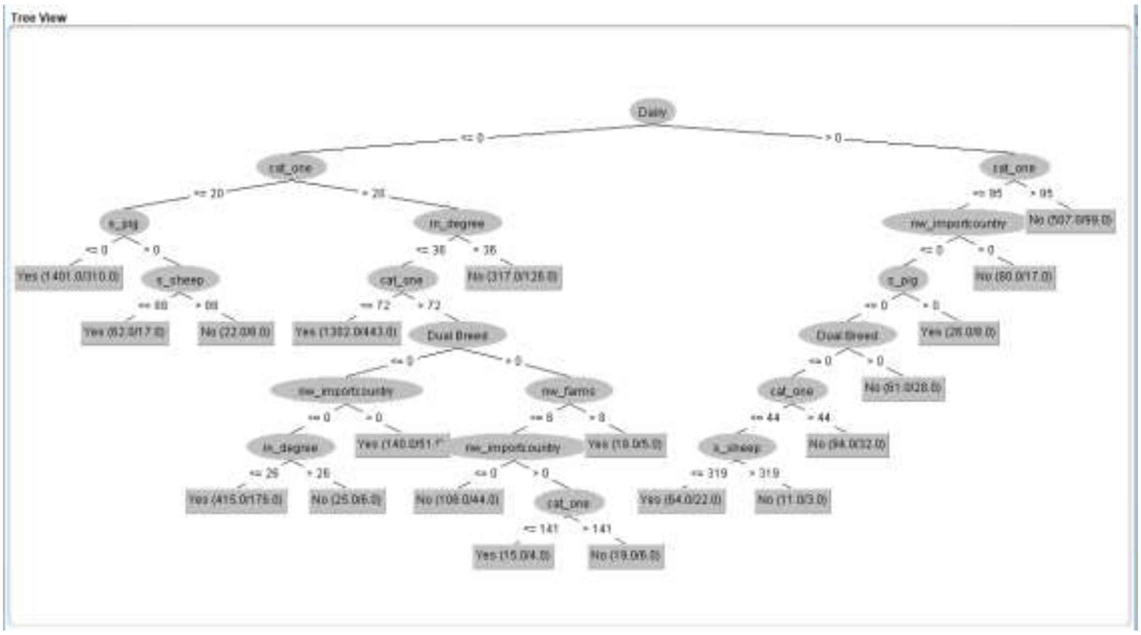


Figure 40. Graphical model of the J48 Classifier on the Always Negative dataset (minNumObj 10)

Performance Metrics J48 Classifier on the Always Not Negative Dataset

Note: J48 is not able to produce any meaningful model for the Always Not Negative dataset

```
=== Summary ===
Correctly Classified Instances       4546           97.0331 %
Relative absolute error             99.637  %
```

```
=== Detailed Accuracy By Class ===
TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Class
1.000      1.000    0.970      1.000      0.985          ?          0.496        0.970       No
0.000      0.000       ?         0.000         ?              ?           0.496       0.029       Yes
0.970      0.970       ?         0.970         ?              ?           0.496        0.942      Weighted Avg.
=== Confusion Matrix ===
   a       b   <-- classified as
 4546    0 |    a = No
   139    0 |    b = Yes
```