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

Neural Network Development for Socio-ecological Modelling of Conservation Conflict

SeyedPouria Modaresi

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

September 2019

Abstract

By 2050, it is expected that Africa will lose 50% of its birds and mammals, and Asian fisheries will completely collapse due to disintegration of the earth. The environmental degradation is a result of consumption of assets such as air, water and soil making nature's turf seen to be pernicious. The expanding human populace, loss of plants and sea life will reduce the Earth's ability to absorb carbon, creating a vicious cycle. Agricultural activities are one of the key reasons for loss of biodiversity as it has a reciprocal relationship with climate change. This project will look into such agricultural practices in terms of land-use issues using a structured dataset generated through a simulated game.

This project has two objectives: Firstly, to generate an AI to replicate the decision-making of a game player (i.e. a farmer). Secondly, predicting system dynamics by simulating socio-agricultural conditions. Consequently, agricultural policies for optimum land-use can be adjusted based on farmer's behaviour.

The two different methods along with machine learning approach and neural network-based approach suitable for data exploration and prediction have been discussed in detail in this project to aid decision-making. Four supervised classification models are investigated; Random Forest, XGBoost classifier, Keras Classifier and multi- layer perceptron and their corresponding results are presented. Further, these models are evaluated before concluding on the best model.

The final models suggested for deployment are XGBoost classifier with 80% accuracy, a machine learning approach if the decision to build model for each cell and Keras classifier with 70% accuracy if the decision is to build a single model for each cell. Also, limitations of the study and the future work are discussed to improve the model accuracy for future use.

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:

- Data Cleaning in R language (section 4.2) has been done by Dr. Brad Duthie, Biological and Environmental Sciences Department, University of Stirling.
- The simulated game (NonCropshare) used as a case study for this project has not been developed by me (section 3.2).
- I have not generated dataset from the game. The data has been provided by a researcher from Biological and Environmental Sciences Department, University of Stirling.

Signature *SeyedPouria Modaresi* **Date** 30-08-2019

Acknowledgements

First of all, Thanks to God for giving me the ability and the power to understand and complete this work.

I acknowledge the support and love of my parents for their moral, financial and emotional support throughout my MSc Big Data degree at the University of Stirling. I would also like to thank Dr. Gabriela Ochoa (academic supervisor) for her timely guidance and support during the tenure of this project. Without her assistance, this project would not have been possible.

I would also like to extend my sincere gratitude to Dr. Brad Duthie, researcher in Biological and Environmental Sciences Department at the University of Stirling who helped me to understand the nature of the problem to come up with an adequate solution and Dr. Simon Jones for timely updating with all the necessary information and his support in making academic materials available for the successful completion of this project.

Last of all, I am thankful to my friends and everyone else for keeping me motivated and always accompanying me on long walks be it for leisure or academic work.

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 Problem Definition

The conservation conflict is becoming a global problem and need to be managed to minimize negative impact on human livelihoods, human well-being and ultimately on biodiversity. It arises when there is clash over conservation objectives between two or more parties (multiple stakeholders) who have different interests, values in different political, historical and cultural settings. Therefore, meeting the needs of each of them lead to conflicts; for instance, use of land for farming or fallowing in case of this project. Farming is part of agriculture which involves growing crops or food for people and animals, growing flowers, manure ,fibres (cotton, wood), biofuels and drugs (weed, opium) whereas when land is left to rest and regenerate is known as fallowing. Fallowing land can be used to pasture livestock. Farmers leave some land fallow because of many reasons such as costly access to inputs or to improve the fertility of the land. As of year the 2000, out of 37% of earth's total agricultural land, 11% has been used for growing the crops. This kind of dualistic approach in terms of land use decision for either farming or fallowing in complex socio-ecological systems requires a more comprehensive understanding of the main components of the conflict, including users, resources, and most importantly the relationships between these factors on spatial and temporal scales (Karimi, Hockings, 2018). Understanding the potential benefits for land use depends on its social and environmental dimensions such as sustainability, accountability and productivity, as well as the potential interaction of these factors that may affect the extent of the conflict.

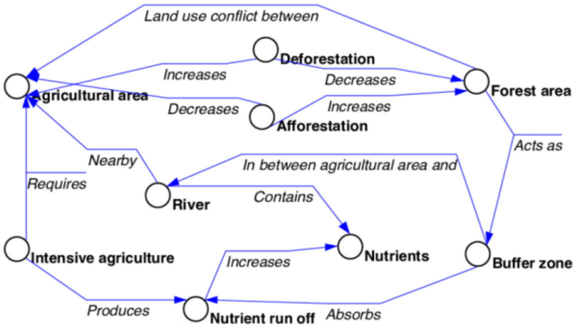


Figure 1: Land-use conflict concept diagram

In the past, researchers and practitioners in this field, used to rely upon games or traditional data analysis techniques to solve such issues. However, modern machine learning techniques provide enhanced opportunities in building AI frameworks to manage conservation conflict that will aid in making most extreme positive effect on critical natural difficulties.

This project has been taken in collaboration with Dr. Brad Duthie from Biological and Environmental Sciences department, University of Stirling. For the purpose of this project under consideration, a coordination game for insect-based ecosystem services has been opted to devise a strategy for land-use decisions in Madagascar agricultural landscape. Games provide an exciting opportunity to address the complexity in managing the conservation conflict that traditional approaches fail to do so. A game being a competitive activity is played just for fun in which an outcome depends upon the action taken by the player. Games act as a powerful research tool to investigate the behaviour of the participants to identify key elements of a conflict and provide a framework for formal analysis of conflicts by exploring the underlying patterns with respect to the stakeholder values, interests and emotions. Hence, games can be helpful in providing insights to the understanding and management of conflicts (Redpath et al, 2018).

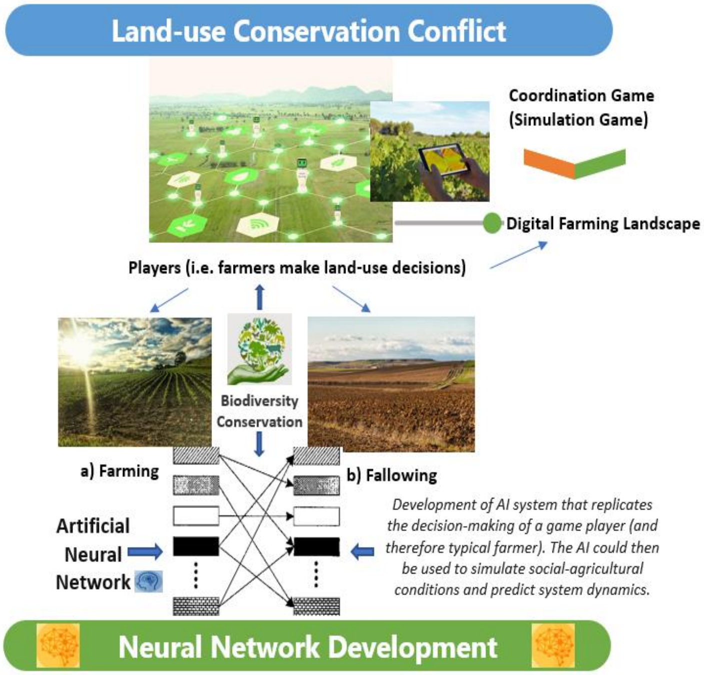

Land-use Conservation Conflict
Coordination Game
(Simulation Game)
Digital Farming Landscape
Players (i.e. farmers make land-use decisions)
Biodiversity
Conservation
a) Farming
b) Fallowing
Artificial
Neural
Network
Development of AI system that replicates the decision-making of a game player (and therefore typical farmer). The AI could then be used to simulate social-agricultural conditions and predict system dynamics.
Neural Network Development

Figure 2: Neural network development to manage land-use conservation conflict

This game revolves around the four farmers who make decision pursuant to farming or fallowing on a 6 *6 grid-cell digital farming landscape. To create "safe" AI, a definitive objective is to guarantee that it progresses toward becoming worth adjusted – that its concept of a decent future is lined up with humankind's qualities, promising safe utilization of the innovation for humankind. This implies governing rules created to guarantee that developing AI frameworks stay "inviting" must consolidate the wellbeing of the common habitat as an essential necessity. The conservation conflict will be investigated with the use of Neural Network by simulating social-agriculture conditions to deal with the difficulties of replicating the decision making of a game player using incentives. This approach will help to understand and analyze the patterns in relation to the problem of conservation conflict with respect to farming and fallowing decision by the stakeholders. The game as a tool to address such kind of conservation conflict provides a novel approach for conservation planning and management. Given the opportunities, this field will be benefitted with the means of data science approach in a changing environment to combat conservation conflict.

1.2 Background and Context

Biological systems are critical to human prosperity. Unsettling influences in geo-biochemical cycles, the loss of biodiversity, and environmental change are possibly irreversible and present genuine dangers to humanity. The Earth is losing its biodiversity at a higher rate than in history. One out of five animal varieties on Earth currently faces extinction, and researchers gauge that this will ascend to half before the century's over except if we take direct action. Since 1500, already **322 species have become** extinct (Dirzo et al., 2014). In the long run, such extinctions have a profound effect on depletion of earth's aesthetic resources and if not properly taken care of will lead to deterioration of our ecosystem (Ceballos et al., 2017). One of the key reasons for loss of biodiversity is the lack of good strategy for *land-use* decision making. 'Land use' means anthropogenically utilizing the land (Islam et al. 2017) to convert natural landscapes like forests into agricultural production (built environment) (Munroe, 2017). Generally, Land-use takes following forms:

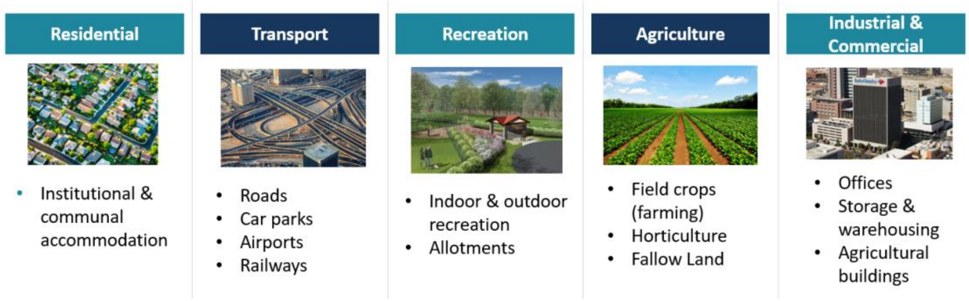


Figure 3: Types of Land-use

Although land use practices, vary from country to country and from region to region, they have similar global impact on the crucial habitat of plants and animals. According to United Nations' Food and Agriculture Organization Water Development Division, the products and benefits we get from the land along with all the activities carried out by humans in order to produce products and benefits forms part of land use concerns and effective decision making. Agriculture being an engine of growth to feed almost 1 billion people (who are undernourished) needs to be preserved for economic development and is vital to environmental services (Diouf, 2011). Despite of many attempts to conserve land, depletion of natural resources especially where agriculture is practiced, continues to happen and poses a tangible threat, which further diminishes the land's ability to produce sufficient and quality food to achieve sustainable development goal of zero hunger in various countries across the world.

1.3 Scope and Objectives

Life on this planet has become hard for all of its wide diversity because of continuously deteriorating environment. Almost 1 million species of plants and animals already face extinction. Such extinction represents a debt- a future ecological cost of current habitat destruction (Tilman et al, 2014).

Human activities have played a major role in these declines with a population of 7.6 billion. Our attempts to pluck natural resources from sensitive ecosystems lead to loss of biodiversity on earth. From the way in which we use the land and oceans, such as for farming and fishing, to logging and mining, humans have a huge contribution in eroding the ecosystem.

Figure 4: Deforestation for agriculture in Madagascar

It fosters the need to take a strict action to protect the environment from further loss. To address this issue, the main aim of this study is to develop a neural network using the results of the game played by the players i.e. farmers. The results have been given in the form of a dataset. The data usually comes in two forms- structured and unstructured data. Structured data is the data that is available in it predefined format such as SQL databases and excel spreadsheets while unstructured data doesn't exist in its predefined format such as data in the form of text, audio or video.

For this study, the data was available in structured form, which will be used as an input to develop a deep learning framework using Keras. Keras is written in python language and is a high level neural networks application programming interface that will be used to classify the players' decision for each cell pursuant to farming or fallowing.

Objectives:

The objective of this study is to make use of artificial neural networks to complex data, which is available in structured form, and lack the research in worldwide application models.

The goal of this project and the research question to investigate into is:

Firstly, generate an AI that replicates the decision-making of a game player (and therefore typical farmer). The AI could then be used to simulate social-agricultural conditions and predict system dynamics.

1.4 Method Employed

For the successful implementation of this project, the Cross-Industry standard process for Data Mining (CRISP-DM) was adopted (Chapter 3).

1.5 Achievements of the Dissertation

The two set objectives of the study as stated in section 1.3 were successfully achieved. To achieve the set objectives, it was very important to understand the nature of the project and the requirements of the researcher in Biological and Environmental Sciences Department at the University of Stirling. For building neural network, the cleaned data was provided on which I have done some data exploration to extract meaningful patterns. The techniques for deep learning approach were researched upon and best techniques were selected to build neural network. To explore more and to conclude with the best method along with the appropriate model, two machine learning algorithms (specifically ensemble methods) were also used. Through rigorous experimentation, it was concluded that for method 1 i.e. Building model for each cell, a machine learning algorithm known as XGBoost is appropriate (section 7.2) and for method 2 i.e. Building a single model for all cells, a neural network (deep learning) approach known as Keras classifier is appropriate (section 7.4). Later, the results were discussed with the researcher who was satisfied with the work done so far and the thoughts for future work (section 8.5).

Other than the set objectives, this project has helped me to improve my personal knowledge in data science techniques and their application, the hurdles that come across data science process and how to overcome such problems to achieve the set objectives. Through continuous research and improving my knowledge through various courses from data camp and other sources, I was able to lead the project in a best direction and contributed to the research in neural network development using simulated data.

1.6 Overview of the Dissertation

This dissertation is organized into eight sections / chapters as follows:

- **Chapter1: Introduction**

  This chapter gives information related to what project is all about, background and context of the project, scope and objectives of the project. Also, the method employed to successfully implement the project and achievements of the project are described in this chapter.

- **Chapter2: Literature Review**

  This chapter covers the Literature Review as introductory section which describes domain knowledge of the land-use conservation conflict with emphasis on the use of Artificial Intelligence field to address this issue and previous related work section which describes about various tools and techniques used in the past to address similar problems. This chapter also provides insight into the motivation for the work proposed and reasons to chosen the preferred techniques.

- **Chapter3: Methodology**

  This chapter comprises of the information about the data mining approach used to execute the study. It also gives insight into the game (NonCropshare, Coordination game) which is used as a case study for this project.

- **Chapter4: Data Preparation**

  This chapter contains detailed information on data collection, data summary, data exploration and data processing.

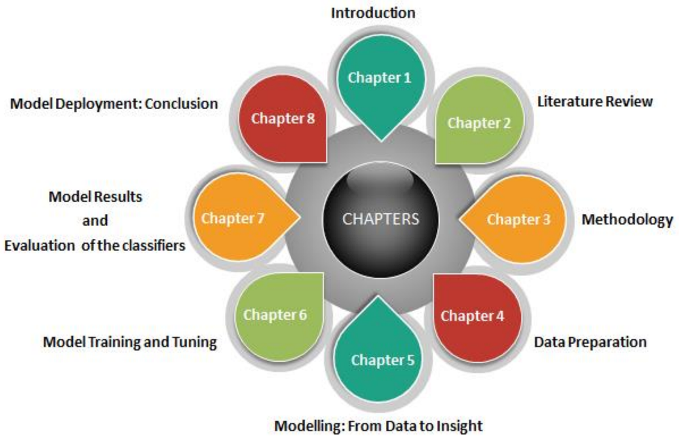


Figure 5: Overview of Dissertation

- **Chapter5: Modelling: From Data to Insight**

  This chapter explains the various machine learning and deep learning algorithms used to build model for this study. It describes the basic structure, learning process, prediction process et al. of the algorithms used.

- **Chapter6: Model Training and Tuning**

  This chapter covers information about the various experiments done on the algorithms proposed to optimize the results of the model after adequate hyperparameter tuning.

- **Chapter7: Model Results & Evaluation of the classifiers**

  This chapter is dedicated to define the results of the principal classifier and their evaluation using appropriate performance metrics.

- **Chapter8: Model Deployment: Conclusion**

  This chapter covers the main findings and conclusion of the project. Finally, recommendations in lieu of the various challenges and limitations are also presented along with the future work.

2. LITERATURE REVIEW

2.1 Introduction

The primary driver of biodiversity loss is due to extensive growth of agriculture which further leads to habitat loss (The Millennium Ecosystem Assessment 2005) (Perrings et al, 2015). More than half of all tropical dry forests, shrublands, temperate broadleaf forests and grasslands and more than two third of Mediterranean forests had already been transformed to agriculture. Too much dependence on agriculture as a food-producing zone makes both water and soil nutrients least abundant, which fosters the need for appropriate trade-off between environmental needs and agricultural production. This gives rise to the conservation conflict that is often complex because of multiple stakeholders of different interests. Though many traditional approaches have been used to address this issue but they failed to meet this challenge and lead to adoption of innovative approach such as the use of games to disentangle it. Games are a powerful research tool to study and clarify the key components of a conflict and provide a solution to engage stakeholders into productive discussion. The data generated from the games are perfect for studying human behaviour because players' every interaction gets recorded. Games such as simulated games provide a platform for deep learning neural networks to solve complex problems like conservation conflict. Deep learning will add value by setting up basic parameters for the data gathered from the game and the trains the computer to recognize the patterns by learning on its own using many layers of processing. The use of deep learning approach will help to develop a predictive system that generalize and adapt the data well and continuously improves as new data arrives.

To sum up, this project involves training the task rather than fitting a model to aid decision making pursuant to land –use dilemma such as, to use it for planting money crops i.e. farming or let it unplanted with high nitrogen crops to return vital nutrients to the soil i.e. fallowing. This approach is effective in this kind of scenario where the task is to meet the objectives of food producers i.e. farmers to maximize their crop yield under complex socio-ecological interactions and uncertainty by building a AI model that replicates the decision-making of stakeholders (players in the game or farmers) and predicting system dynamics (with the help of AI simulating socio-agricultural conditions).

2.2 Previous Related Work

2.2.1 'Conservation games' for conservation of biodiversity

William M. Adams, Bruno Monteferri (2014) in their study regarding Digital Games and Biodiversity Conservation ,discussed about the current interaction between biodiversity conservation and digital gaming, and the potential of what it is called "conservation games. They assessed the potential of conservation games (digital games that promote environmental protection). These include ways in which digital games may be protected by (1) the use of training and behavioural change, (2) raising money, and (3) researching, monitoring, and planning. It is about the danger that games may hide gamers from the real world and their problems, or provide simple misleading narratives on protection issues. They concluded that there is a high potential for protection for the greater use of digital games, provided that the protective games are developed in cooperation with game design specialists, specific goals, not the key objectives, the specific target audience and the protection.

2.2.2 Use of gaming methodology to influence individual farmers' decisions

Gaming for smallholder participation in the design of more sustainable agricultural landscapes study proposed by E.N.Speelmana et al. (2014),tried to evaluate smallholder farmers behaviour and their strategies for using their lands .The main aim of this study was to offer and suggest better services and farming system to local and non-local farmers. To do so, they used gaming methodology to involve farmers and recognize collective decision-making criteria and models through in-depth analysis of game strategies implemented by farmers. Furthermore, the implementation of this game proved to be fruitful to develop a hypothesis in order to find out how communication, leadership and coordination help during the land-use decision-making phenomenon. In this paper, the authors have used simple and stylized game which provides a realistic, coordinated resource based view in a complex agricultural landscape, in which a variety of incentive schemes have been used to influence individual farmers' decisions. Furthermore, the implementation of this game proved to be fruitful to develop a hypothesis in order to find out how communication, leadership and coordination help during the land-use decision-making phenomenon.

2.2.3 Conservation planning in agricultural landscapes

Shackelford et al. (2015) conducted a study on conservation planning in agriculture landscapes. The purpose of study was to find out the relation between expenses and pros in relation to conservation versus production to come up with an ideal framework to devise a strategy for systematic conservation in agricultural landscapes. They used geographic data of global land available to sample cropland and non-cropland of the world. However, for finding hotspots of conversation conflicts the method called as spatial scan statistic has been used. The results of this study showed that the ideal conversation systems will help to recognize hotspots that are required to be protected among farming landscapes. Also, it has been found that land sharing and sparing should be considered as separate tools to address conservation conflict.

2.2.4 Machine Learning approach for conservation of ecosystem services using Weka Software

Willcock et al. (2018) investigated into the machine learning techniques for conservation of ecosystem services with the use of increasingly available data to scale-up ecosystem service models, analyze and predict flows of these services to competent individuals. To do so, researchers used Weka and Aries applications for having two data-driven modelling (DDM) of firewood in South Africa and Sicily.

2.2.5 Artificial Neural Networks approach to aid environmental problems

Lae et al. (1999) used Neural Network approach to predict fish yield and demonstrated the benefits of using the method of back propagation algorithm of neural networks as a stochastic approach in conservation of aquatic ecosystem. Lek et al. (1995) used multiple linear regression and artificial neural network to predict the food consumptions of fish population. Ozesmi et al. (1999) also used artificial neural network approach for building spatial model for habit selection of bird species with interspecific interaction. The decision pertaining to which neural network model to be used depends upon in-depth understanding of the ecology of system under study. Gavrey et al. (2003) investigated that ANNs are the best tools for solving prediction problems as they help in understanding the ecological phenomenon to find solutions to environmental problems, restore those solutions to improve the environmental conditions for life.

2.2.6 Theoretical games to recognize conservation problems

Colyvan et al. (2011), in their study recognized conversation problems in a real-world and how these problems can be addressed using theoretical games. This study added value to the existing adaptive management methodologies. Barrios et al. (2011) analyzed a version of SIERRA SPRING game (theoretical game) to reflect on individual coordination strategies for land use in small rural watersheds. Through this research, they introduced game as a tool for stakeholders involved in land use planning.

2.2.7 Use of Incentives for conservation of environment

Bell at al. (2016) highlighted the importance of the use of payments for ecosystem services using a NonCropShare coordination game for provision of insect based ecosystem services. Redpath et al. (2018) focused on 100 recent published articles to compare conflict behavior, intervention processes to identify the relationship between them and other geographic variables by using Chi-squared tests method.

2.2.8 Motivation for the work proposed: Importance of Rational Land-use planning

Based on the existing literature, it is evident that conservation conflicts negatively impinge upon biodiversity, livelihoods and human well-being, but has received little attention to figure out the factors or reasons that lead to argument between stakeholders over conservation objectives. From illegal wildlife killing to resource use conflict, the majority of interventions aimed at reducing conservation conflict. Some relevant works in the domain of biodiversity and environmental conservation highlighted the importance of using more advanced techniques such as AI based artificial neural networks over traditional approaches like linear regression and decision trees et al. The research also pointed out that no single factor impacted socio-ecological system but number of factors such as *technical human behavioural interventions* that attempt to change the external environment, *cognitive human behavioural interventions* that attempt to change human behaviour by disseminating important information in lieu of the problem and *structural human behavioural interventions* that attempt to change the context itself in order to discourage certain resource use, For instance, farming instead of fallowing or vice-versa. The existing work is more focused on conservation of biodiversity in general. There has been no much consideration about land-use planning which is very important aspect in terms of environment conservation as it pose many challenges to decision-makers. By 2025, the world population is expected to grow at least 8 billion from 6 billion today (Wrachein, 2003) which highlighted the importance of preserving the environment while achieving food security and maintaining quality of life. The main activity involved here is agriculture that involves production of the food. So, increasing agricultural development in turn, accentuate that sustainable use of the soil is vital for proper land management. Sustainable use of soil means retaining natural fertility of the soil for production of quality food on a long-term basis. It further implies to treat and manage the environment in such a way that the cycles and the energy fluxes among the bodies of water, the soil and the atmosphere are considered, preserved and restored on a continuous basis. Thus, rational land-use planning is fundamental to reduce the risk of environmental degradation and preserve the planet for future generations.

2.2.9 Deep Learning Artificial Neural Network approach to address the issue of land-use planning

To address such kind of earth's environmental challenges which left unguided, Artificial Intelligence (AI) presents transformative opportunities to create positive impact on urgent environmental challenges. Deep learning (a sub-set of machine learning) approach is an innovative way of incorporating a system that considers the health of the environment as a fundamental dimension. Artificial Neural Networks (ANNs) provides a simplified model of the biological neural network inspired by human brain functionality emulating complex functions such as cognition, learning, pattern recognition and decision making. Such models are typically useful in field conditions management and crop management in agriculture. Soil is a heterogeneous natural resource whose temperature alone can give insight into effects on regional yield due to climate change. ANN has the ability to understand the dynamics of ecosystem by studying soil moisture and temperature and the impingement in agriculture. Deep learning and ANNs also help in estimating yield prediction based on soil quality and give meaningful insights on crop quality to reduce waste. AI acts as a 'virtual coach' for providing guidelines on making productive decision. Artificial neural networks help to develop algorithms that can be applied to complex patterns and prediction problems. It proposes a way by which simulated intelligence can help change customary parts and frameworks to address environmental change, convey nourishment and water security, secure biodiversity and reinforce human prosperity (Herd et al., 2018). This concern is firmly connected with the rising inquiry of how to guarantee that AI does not end up unsafe to human prosperity (Mckinley, Cheng, et. al, 2008).

Hence, AI provides the capability to build an autonomous intelligence system that will automate the decision-making of the players (particularly farmers) pursuant to farming or fallowing without the need of actual human (farmers') intervention. AI has immense potential to help unlock solutions to the problem of land use planning for farming or fallowing.

This study focuses on unlocking value from structured dataset where data is gathered from a simulated coordination game by allowing AI to act independently. The existing literature doesn't address this kind of issue specifically. Deep learning approach will be applied on the dataset in question propelling an innovative approach for socio-ecological modelling of conservation conflict.

Also, the proposed techniques have been compared with the ensemble machine learning models such as Random Forest and Extreme Gradient Boosting algorithm (explained in the Modelling section) before making conclusions about the current state- of-the-art.

3. METHODOLOGY

3.1 CRISP-DM: Standard Process for Data Mining

The adoption of data mining is at its hike across various industries because of massive data collected on a daily basis and the need to reveal hidden relations, insightful information and behavioral patterns out of that data to aid decision making process of the organizations. Thus, there is need to adopt a standard and streamlined approach which will help to transform a business problem into data mining task successfully (Wirth, Hipp, 2000). The success of a data mining task using a standard approach will be measured in terms of appropriate data transformation, use of effective data mining techniques with a means to evaluate their results and then finally documenting the experience. There are various methodologies to opt for a particular data mining project such as **Knowledge Discovery Process (KDD)** that involves five stages as Selection, Pre-processing, Transformation, Data mining and Evaluation/Interpretation; SEMMA (Sample, Explore, Modify, Model, Assess) developed by SAS institute and CRISP-DM (CRoss-Industry Standard Process for Data Mining) which is a six stage process that considers both business and technical aspects of a project (Azevedo, Santos, 2008). But the most widely used comprehensive project management approach by data scientists is CRISP-DM which will be used for the project under consideration.

CRISP-DM is a cyclical approach that comprises of six stages which are duly organized, structured and defined. The stages involved in CRISP-DM process are outlined in the below figure:

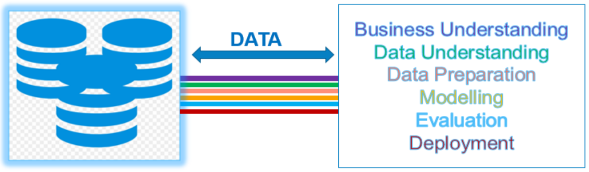


Figure 6: Stages involved in CRISP-DM process

<u>The major phases involved in the implementation of a data mining project using CRISP-DM approach are defined as follows:</u>

a. **Business Understanding:**

The first stage involves defining what you want to accomplish from a business perspective. It refers to creating a balance between the business/project objectives and the constraints involved that influence the outcome of the project results. After gathering knowledge from view point of business, data mining problem is formed and preliminary project plan is designed to achieve the stated objectives.

b. **Data Understanding:**

It involves collection of data in order to get familiar with the data. The idea behind this step is to identify data quality problems and to detect interesting subsets so as hypotheses is formed for hidden information to discover first insights into the data.

c. **Data Preparation:**

Data Preparation means making data ready for analytics by blending, shaping and cleansing the raw data which may come from disparate data sources). Data preparation is very important part of data mining process and the phrase "Garbage In, Garbage Out" is well suited to data preparation as irrelevant and unreliable data (Garbage in) will lead to misleading results (Garbage Out).

Various pre-processing techniques used for the purpose of this project will be explained in detail at a later stage in next section (Modelling section).

d. **Modelling:**

In this stage, various machine learning algorithms such as regression, neural networks, decision tress et al. are a selected and applied after proper hyperparameter tuning to reach at the best model for the problem under consideration.

e. **Evaluation:**

It is important to evaluate the model in-depth that appear to be high quality than other models in order to achieve the project objectives. If the results are in correlation with the stated objectives, revealing no deviation then the decision to use data mining results should be reached.

f. **Deployment:**

The last step in CRISP-DM process is to organize and present the knowledge gained during previous steps in proper format for instance, generating a report so as customer can use the results. Deployment is very crucial for the determination of business success. Thus, it requires a proper strategy to avoid unnecessarily long periods of incorrect usage of data mining results.

3.2 NonCropshare: a Coordination Game as a tool to address land-use conservation conflict

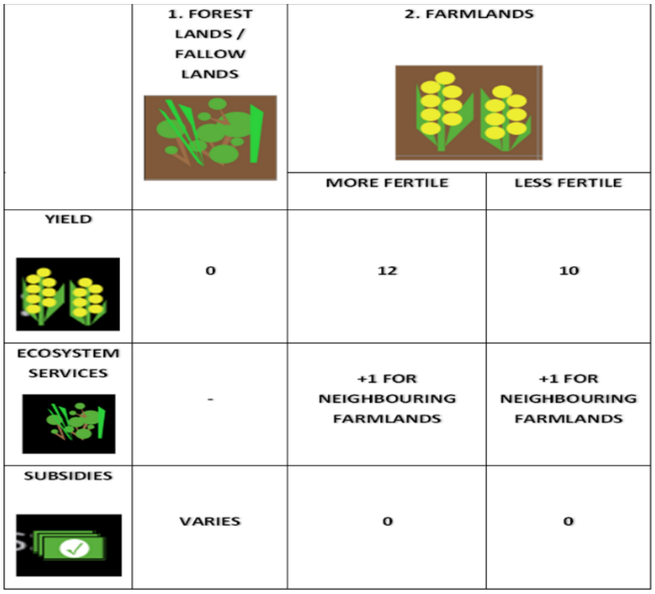
The dataset for this project has been generated from a coordination game known as NonCropshare. It is an experimental field game for insect based ecosystem services (Bell et al. 2013, Bell et al. 2016). The game has been framed to investigate farmers' land-use decision in Madagascar agricultural landscape. The game is played on tablet computers by the four farmers (who make land-use decisions) via a mobile hotspot. The role of the game is to investigate how changes in incentives impact the land-use choices i.e. whether to maintain non-crop habitat or employ pesticides shift. The game consists of 6 x 6 grid-cell digital farming landscape where each grid is formulated as a fallow forest for shifting cultivation.

On each land cell, farmer has following two options:

- Conserve the fallow forest.
- Farm the land for private benefits.

Among above two decisions, farmers will get some yields from farming whereas deciding to fallow the land will boost up crop yields to neighboring cells through ecosystem service provision, For instance, pollination, watershed protection, landslide prevention and soil restoration.

At least six rounds analogous to agricultural years are played in each experimental game session where participants (farmers) have the option to communicate their land use decisions. In each round, farmers decide on the utilization of land squares i.e. whether to fallow or farm the land squares. The return to farming depends upon the conditions of lands as shown in the table below. For example, if land is in a good condition (e.g. intact primary forest), farming gives a payoff of 12 but if land is of low quality (e.g. Secondary regrowth) the payoff will be 10. All fallow forests, if farmed, leads to high yield at the start of each session but if the same fallow square is farmed for two consecutive rounds, the yield dwindles to low level and fosters the need to fallow the lands for at least next two consecutive rounds so as fertility of soil can be restored and high yield can be recovered.



	1. FOREST LANDS / FALLOW LANDS	2. FARMLANDS	
		MORE FERTILE	LESS FERTILE
YIELD	0	12	10
ECOSYSTEM SERVICES	-	+1 FOR NEIGHBOURING FARMLANDS	+1 FOR NEIGHBOURING FARMLANDS
SUBSIDIES	VARIES	0	0

Figure 7*: Working of NonCropshare game*

To influence the decision of farmers, for every square of fallow forests in the digital farming landscape a subsidy of x is given and their overall score in each round is calculated as:

Score = ∑Yield + ∑Ecosystem services + ∑Subsidies

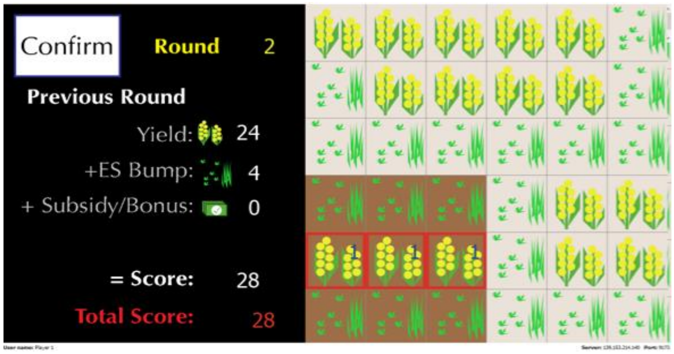
The two game treatments played in random order during each game session form a 2 x 2 design of-

i) Subsidy,

ii) Individual property rights.

1) Treatment T1: Individual property rights, no subsidy (8-10 rounds)

3x3 grid-cell sections of the 6x6 grid-cell digital agricultural landscapes, each participant has to make land use decision on the nine fallow forest patches endowed to him (figure 1). The two main land use decisions are farm or fallow for each player. In first treatment, if participants conserve fallow land, no subsidy is offered to them. All four players make decisions in parallel and select their choice on their tablet computers by cycling through land options available for each cell and the round ends after each player confirms his final choice. The total score is calculated for each cell on whom the choice was made and the cell around it. At the end, each participant can view his score and what has happened across the whole landscape and what yields were achieved in each cell.



Round 2

Previous Round

Yield: 24

+ES Bump: 4

+ Subsidy/Bonus: 0

= Score: 28

Total Score: 28

User name: Player 1

Figure 7: Dashboard of NonCropshare game.

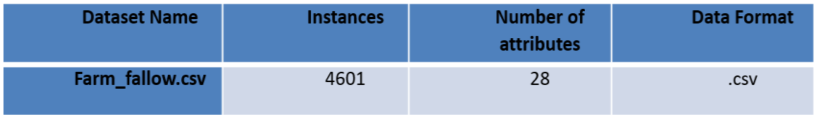
2) **Treatment T2: Individual property rights with subsidy (8-10 rounds)**

3x3 grid-cell sections of the 6x6 grid-cell digital agricultural landscapes, each participant has to make land use decision on the nine fallow forest patches endowed to him (figure 1). The two main land use decisions are farm or fallow for each player. This treatment follows the same procedure as the first treatment except that randomly assigned flat subsidy is offered to the player for conserving fallow land.

This game is used as a case study for this project, to build a neural network using the data generated from the game. The techniques used to build a neural network have been explained at a later stage in the Modelling section.

4. DATA PREPARATION

Data preparation is very important to get reliable results from the models. It has a huge impact on the success of complex data analysis in neural network modeling (Yu et al, 2005). As data is collected from one or more source, so there is a need to make it model ready by transforming its quality prior to use it for building a model. To enrich data quality, data may need to be formatted (such as from relationship database to flat file), cleaned (such as removing outliers, missing values etc.) or sampled (such as over-sampling or under-sampling). It depends upon the nature of data, what steps are required to perform data pre-processing. For the purpose of this project, the data has been provided by a researcher from Leverhulme Trust Early Career Fellow, Biological and Environmental Sciences. Originally, the data was collected on the basis of a coordination name known as NonCropshare (as explained in the methodology section). It has been provided in .csv format and contains 4601x28 matrixes of data where 'x' represents input features while 'y' represents output features.



Dataset Name	Instances	Number of attributes	Data Format
Farm_fallow.csv	4601	28	.csv

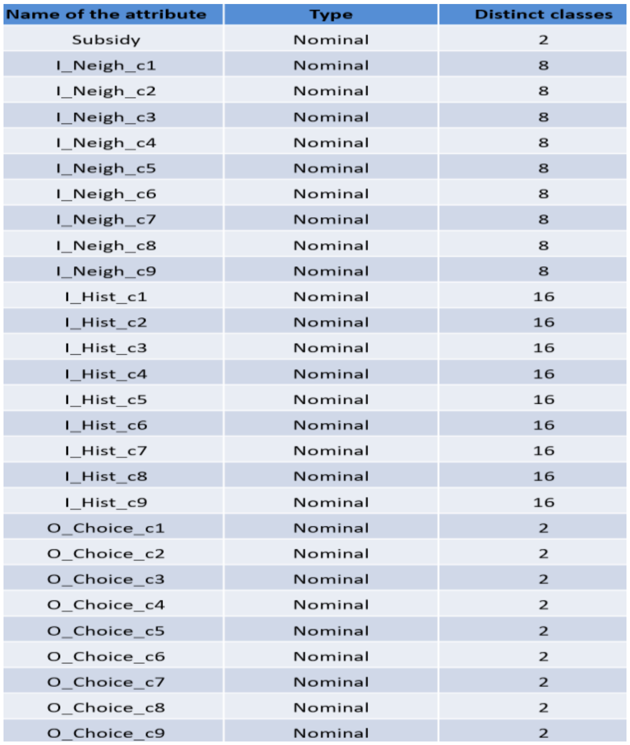
Table 1*:* Dataset Details

4.1 Data Summary

There are 28 feature labels in the given dataset. The definitions of feature labels are as follows:

- **Subsidity Column:** The column I_Subsidy just indicates whether (2) or not (1) players get a subsidy for not farming and allowing a landscape to remain fallow.
- **I_Neigh_cx:** There are 9 neighbouring columns ranging from I_Neigh_c1 to I_Neigh_c9. The columns I_Neigh_cx indicate how many neighbouring landscape cells were fallow cells in the previous time step (potentially affecting payoff).
- **I_Hist_cx:** There are 9 history columns ranging from I_Hist_c1 to I_Hist_c9. The I_Hist_cx columns reveal the previous four time steps of farm v/s fallow decision on a landscape cell.
- **O_Choice_cx:** There are 9 choice columns ranging from O_Choice_c1 to O_Choice_c9. The O_Choice_cx columns reveal the output that farmers will decide to farm or fallow.

The description of the attributes is shown in the following table:



Name of the attribute	Type	Distinct classes
Subsidy	Nominal	2
I_Neigh_c1	Nominal	8
I_Neigh_c2	Nominal	8
I_Neigh_c3	Nominal	8
I_Neigh_c4	Nominal	8
I_Neigh_c5	Nominal	8
I_Neigh_c6	Nominal	8
I_Neigh_c7	Nominal	8
I_Neigh_c8	Nominal	8
I_Neigh_c9	Nominal	8
I_Hist_c1	Nominal	16
I_Hist_c2	Nominal	16
I_Hist_c3	Nominal	16
I_Hist_c4	Nominal	16
I_Hist_c5	Nominal	16
I_Hist_c6	Nominal	16
I_Hist_c7	Nominal	16
I_Hist_c8	Nominal	16
I_Hist_c9	Nominal	16
O_Choice_c1	Nominal	2
O_Choice_c2	Nominal	2
O_Choice_c3	Nominal	2
O_Choice_c4	Nominal	2
O_Choice_c5	Nominal	2
O_Choice_c6	Nominal	2
O_Choice_c7	Nominal	2
O_Choice_c8	Nominal	2
O_Choice_c9	Nominal	2

Table 2*:* Data information

In the original dataset, all the feature labels were in the integer form (int64) i.e. numerical forms which have been changed to nominal so as each feature label represent some class. Nominal variables in machine learning consist of discrete categorical values that have no order.

4.2 Data Cleaning in R

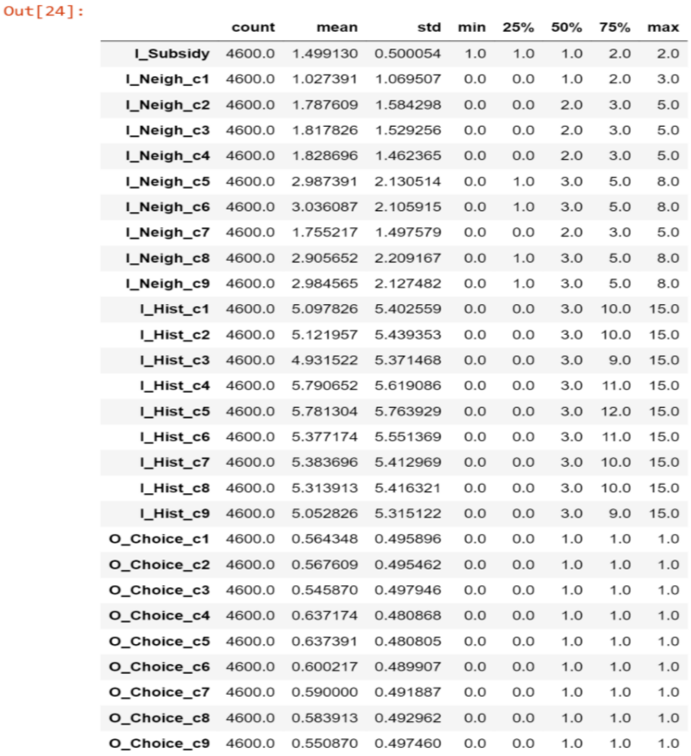
The data generated from the coordination game was imported to R for formatting and cleaning. Data cleansing refers to improve the data quality and model performance. This include deleting the duplicate and irrelevant value, filling missing values with appropriate values and changing nominal values to numeric values and also fixing the issue of inconsistent values. The first step performed was handling and fixing missing values, because the data type was categorical for each cell. It is solved by replacing missing value with previous value. As the **I_Hist_cx** columns reveal the previous four time steps of farm vs fallow decision on a landscape cell. The numbers are converted from binary (e.g., 0 = 0000, meaning that in all of the previous four time steps, the cell was fallowed; or 11 = 1011, meaning that in the previous time step, the cell was farmed, but in the time step before that it was fallowed, and the two steps before it was fallowed it was farmed twice). The data cleaning part in R has been done by the data owner i.e. researcher as mentioned above and handed over for the neural network modelling.

4.3 Exploratory Data Analysis

Before building a model, the data that needs to be fed to machine learning algorithm is required to be refined by understanding the contents of the dataset in order to get the valid results. This process is known as exploratory data analysis.

4.3.1 Detection of outliers in the data

The contents of the dataset in hand have been summarized using a function called as *datainfo* (function in python). The *datainfo* function gives the information such as count, mean, standard deviation, quartile range, minimum and maximum value of each column as shown in the picture below. The significance of summary statistics is to find out the outliers in the data to develop strategies to handle them. The below picture shows the summary statistics of the columns when the data is in its integer form. However, python displays results of all columns using *datainfo* function. There is less difference between min and max which states that there are no outliers in the data.

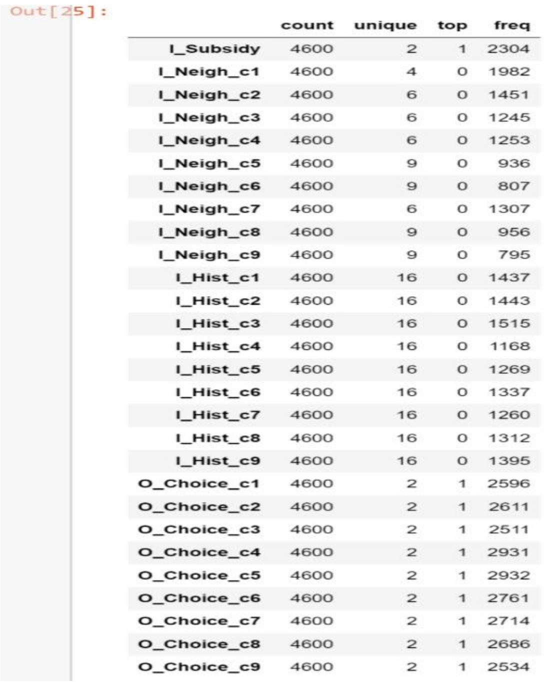


Out[24]:

	count	mean	std	min	25%	50%	75%	max
I_Subsidy	4600.0	1.499130	0.500054	1.0	1.0	1.0	2.0	2.0
I_Neigh_c1	4600.0	1.027391	1.069507	0.0	0.0	1.0	2.0	3.0
I_Neigh_c2	4600.0	1.787609	1.584298	0.0	0.0	2.0	3.0	5.0
I_Neigh_c3	4600.0	1.817826	1.529256	0.0	0.0	2.0	3.0	5.0
I_Neigh_c4	4600.0	1.828696	1.462365	0.0	0.0	2.0	3.0	5.0
I_Neigh_c5	4600.0	2.987391	2.130514	0.0	1.0	3.0	5.0	8.0
I_Neigh_c6	4600.0	3.036087	2.105915	0.0	1.0	3.0	5.0	8.0
I_Neigh_c7	4600.0	1.755217	1.497579	0.0	0.0	2.0	3.0	5.0
I_Neigh_c8	4600.0	2.905652	2.209167	0.0	1.0	3.0	5.0	8.0
I_Neigh_c9	4600.0	2.984565	2.127482	0.0	1.0	3.0	5.0	8.0
I_Hist_c1	4600.0	5.097826	5.402559	0.0	0.0	3.0	10.0	15.0
I_Hist_c2	4600.0	5.121957	5.439353	0.0	0.0	3.0	10.0	15.0
I_Hist_c3	4600.0	4.931522	5.371468	0.0	0.0	3.0	9.0	15.0
I_Hist_c4	4600.0	5.790652	5.619086	0.0	0.0	3.0	11.0	15.0
I_Hist_c5	4600.0	5.781304	5.763929	0.0	0.0	3.0	12.0	15.0
I_Hist_c6	4600.0	5.377174	5.551369	0.0	0.0	3.0	11.0	15.0
I_Hist_c7	4600.0	5.383696	5.412969	0.0	0.0	3.0	10.0	15.0
I_Hist_c8	4600.0	5.313913	5.416321	0.0	0.0	3.0	10.0	15.0
I_Hist_c9	4600.0	5.052826	5.315122	0.0	0.0	3.0	9.0	15.0
O_Choice_c1	4600.0	0.564348	0.495896	0.0	0.0	1.0	1.0	1.0
O_Choice_c2	4600.0	0.567609	0.495462	0.0	0.0	1.0	1.0	1.0
O_Choice_c3	4600.0	0.545870	0.497946	0.0	0.0	1.0	1.0	1.0
O_Choice_c4	4600.0	0.637174	0.480868	0.0	0.0	1.0	1.0	1.0
O_Choice_c5	4600.0	0.637391	0.480805	0.0	0.0	1.0	1.0	1.0
O_Choice_c6	4600.0	0.600217	0.489907	0.0	0.0	1.0	1.0	1.0
O_Choice_c7	4600.0	0.590000	0.491887	0.0	0.0	1.0	1.0	1.0
O_Choice_c8	4600.0	0.583913	0.492962	0.0	0.0	1.0	1.0	1.0
O_Choice_c9	4600.0	0.550870	0.497460	0.0	0.0	1.0	1.0	1.0

Table 3: Summary statistics using datainfo function

As explained earlier that the data has been converted to categorical data, so the summary statistics of the categorical data has been shown in the below picture. It displays the information such as count, frequency and number of unique values.



Out[25]:

	count	unique	top	freq
I_Subsidy	4600	2	1	2304
I_Neigh_c1	4600	4	0	1982
I_Neigh_c2	4600	6	0	1451
I_Neigh_c3	4600	6	0	1245
I_Neigh_c4	4600	6	0	1253
I_Neigh_c5	4600	9	0	936
I_Neigh_c6	4600	9	0	807
I_Neigh_c7	4600	6	0	1307
I_Neigh_c8	4600	9	0	956
I_Neigh_c9	4600	9	0	795
I_Hist_c1	4600	16	0	1437
I_Hist_c2	4600	16	0	1443
I_Hist_c3	4600	16	0	1515
I_Hist_c4	4600	16	0	1168
I_Hist_c5	4600	16	0	1269
I_Hist_c6	4600	16	0	1337
I_Hist_c7	4600	16	0	1260
I_Hist_c8	4600	16	0	1312
I_Hist_c9	4600	16	0	1395
O_Choice_c1	4600	2	1	2596
O_Choice_c2	4600	2	1	2611
O_Choice_c3	4600	2	1	2511
O_Choice_c4	4600	2	1	2931
O_Choice_c5	4600	2	1	2932
O_Choice_c6	4600	2	1	2761
O_Choice_c7	4600	2	1	2714
O_Choice_c8	4600	2	1	2686
O_Choice_c9	4600	2	1	2534

Table 4: Summary statistics (continued) using datainfo function

4.3.2 Checking missing values in the data

To check missing values in the given dataset, *df.isnull().values.any()* function has been used in python. This function returns False as a result which means that there are no missing values in the dataset.

4.3.3 Distribution check of the data

To check distribution of the data, data has been visualized using *sns.countplot()* function. It is useful because it helps to see relationships in data by pointing out the places where there may be significant patterns in order to decide what to dig into next. The below picture displays the visualization of cell 9 using *sns.countplot()* function which shows how subsidy, neighbour, history and choice columns are distributed in the data. However, this function visualized all cells (Appendix1).

In the below picture, the data in neighbour column is normally distributed while for other columns, the function represents the number of classes and their frequency in the data.

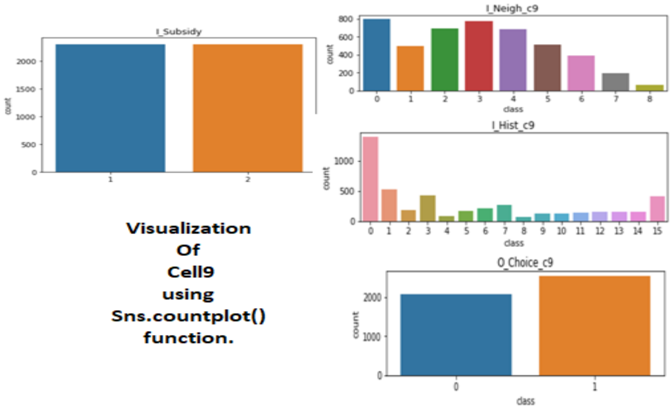


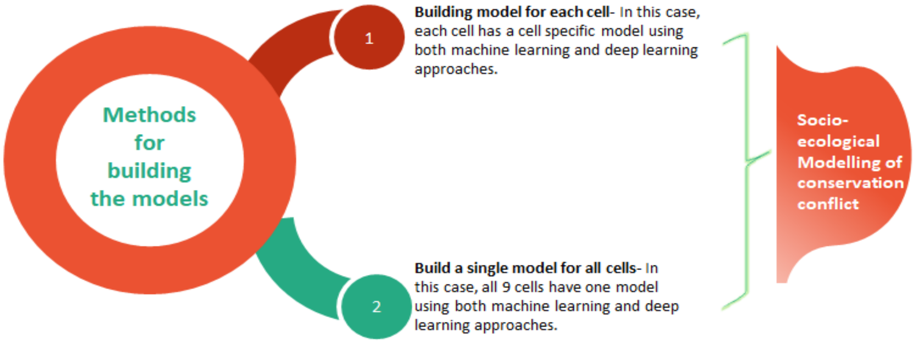
Figure 8: Visualization of Cell9 using Sns.countplot() function.

5. MODELLING: FROM DATA TO INSIGHT

5.1 Introduction

After data preparation is done, the next step is modelling. Modelling in machine learning is a task of feeding training data to machine learning algorithm i.e. learning algorithm to learn from that data. In this stage of CRISP-DM process, machine learning model find the patterns in the training data by mapping attributes of the input data to the target variable and then outputs an ML model that captures the patterns. The result of all the machine learning algorithms is compared and best algorithm is selected for the task in hand. Usually, model selection involves the trade-off between bias and variance. Bias refers to the limits imposed on the selected model and variance measures the sensitivity of the model to the training data. The best model is the one, which has optimal balance between bias and variance, plus the one which is capable of making predictions on the new data.

To build models for replicating decision-making of the players i.e. farmers and predict their behaviour about farming and fallowing to optimize the use of land, deep learning techniques have been opted. Before concluding on the results and model performance, deep learning techniques have also been compared with ensemble machine learning methods to select a model with greater accuracy.
For the problem in question, the models were built using two methods in order to come up with better results as shown in the picture below:


Methods for building the models
1
2
Socio-ecological Modelling of conservation conflict

Building model for each cell- In this case, each cell has a cell specific model using both machine learning and deep learning approaches.

Build a single model for all cells- In this case, all 9 cells have one model using both machine learning and deep learning approaches.

Figure 9: Methods to build the models using machine learning and deep learning approaches.

Below section, explains machine learning and deep learning based Pythonic approach to build the models for the dataset under consideration.

A) Introduction to the algorithms

5.2 Introduction to Machine learning

Machine learning is the science and art of giving computers the ability to learn the data in order to make decision from the data fed to them without being explicitly programmed. Machine learning (a sub-set of artificial intelligence) enables a machine's learning algorithm to identify patterns in the observed data with the help of statistical techniques, build models and predict the better outcomes to aid decision making.

Machine learning techniques can give better insight into the functioning of ecosystem services. This can be achieved by using a general linear model which learns to reproduce the relationship between input variables and the output variables to predict values of target variable. Machine learning has the capability to provide solutions to various environmental problems arising in the face of rapidly changing circumstances such as climate change, globalization, urban sprawl etc. by empowering stakeholders to take data-driven decisions.

Machine learning approach has been divided into: **Supervised learning algorithms** and **Unsupervised Learning algorithms** based on the nature of task in hand.

5.2.1 Supervised learning

In Supervised machine learning, the system tries to learn from the prior examples that have been fed to the system. It is usually used when the data under consideration is a labelled dataset. The data is commonly represented in the table structure. The aim of supervised learning is to build a model that is able to predict the target variable. If the target variable is categorical in nature, the learning task is termed as classification. Alternatively, if the target is continuous in nature, the learning task is termed as regression.

For this project, the task is to classify typical game players' i.e. farmers' decision-making in terms of cells in the digital farming landscape into farming or fallowing. The classification problems involve creating models for a problem with two or more than two classes.

5.2.2 Unsupervised Learning

In Unsupervised machine learning, the task is to learn from the inherent structure of the data without using explicitly provided labels. The goal is to model the underlying structure in order to find previously unknown patterns in the data.

As the data under consideration for this project is a structured and labelled data thus, supervised machine learning algorithms have been used. To build model with greater accuracy, I have used ensemble-learning techniques to build the model.

5.3 Ensemble Machine Learning Models

Ensemble learning techniques are commonly used to improve the model accuracy and thereby performance by using multiple machine learning models instead of one individual model to solve a particular problem.

The total instances in the given dataset are 4600x28 which will be used to build ensemble-learning models.

The types of ensemble learning methods used in this project are briefly described as follows:

Bagging: When a model is built using different subsamples of the training dataset, the technique is called as bagging (bootstrap aggregation). E.g. Random Forest, Bagging meta-estimator etc.

Boosting: When multiple models learn to fix the errors of previous model in a chain, the sequential technique is known as boosting. E.g. XGBoost, Adaboost etc.

The techniques used to build a model using machine learning techniques are Random forest (Bagging) and XGBoost (Boosting) to predict system dynamics by influencing typical farmer's behaviour.

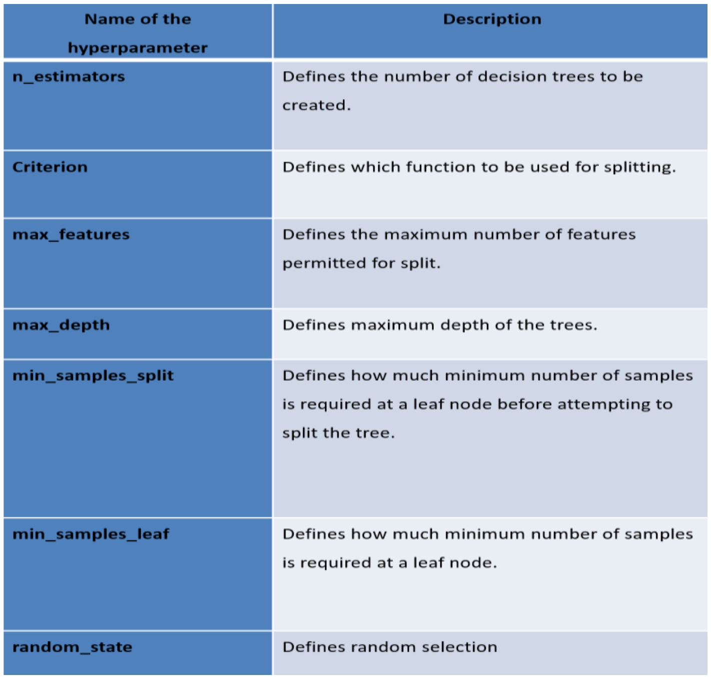
5.3.1 Random Forest

In Random forest, multiple trees (forest) are built by randomly selecting data points and features (used to decide the best split at each node) where base estimators are the decision trees.

Step-by step procedure in Random Forest:

- From the original dataset, random subsets are created.
- Random sets are permitted to decide the best split at each node.
- On each of the subsets, a decision tree model is fitted.
- Final predicted is the average of all the predictions made by a decision tree.

Hyperparameters: The hyperparameters for random forest are explained below:



Name of the hyperparameter	Description
n_estimators	Defines the number of decision trees to be created.
Criterion	Defines which function to be used for splitting.
max_features	Defines the maximum number of features permitted for split.
max_depth	Defines maximum depth of the trees.
min_samples_split	Defines how much minimum number of samples is required at a leaf node before attempting to split the tree.
min_samples_leaf	Defines how much minimum number of samples is required at a leaf node.
random_state	Defines random selection

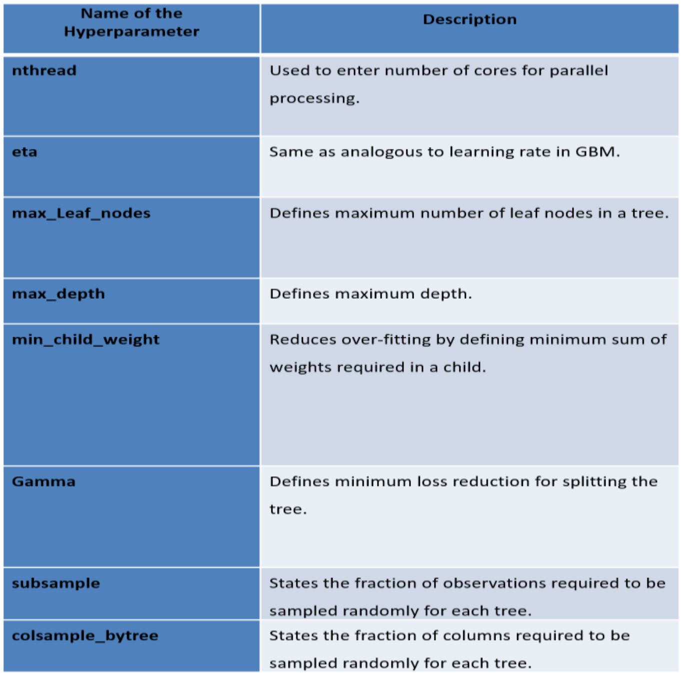
Table 5: Hyperparameters for a random forest algorithm.

5.3.2 Extreme Gradient Boosting (XGBoost)

XGBoost is an extension of gradient boosting algorithms and is known as regularized boosting technique because it reduces model over-fitting and improves the model performance.

It implements parallel processing and is 10 times faster than other gradient boosting techniques. It has in-built function to handle missing values and for cross-validation of the model at each iteration of the boosting process.

Hyperparameters: The hyperparameters for XGBoost are explained below:



Name of the Hyperparameter	Description
nthread	Used to enter number of cores for parallel processing.
eta	Same as analogous to learning rate in GBM.
max_Leaf_nodes	Defines maximum number of leaf nodes in a tree.
max_depth	Defines maximum depth.
min_child_weight	Reduces over-fitting by defining minimum sum of weights required in a child.
Gamma	Defines minimum loss reduction for splitting the tree.
subsample	States the fraction of observations required to be sampled randomly for each tree.
colsample_bytree	States the fraction of columns required to be sampled randomly for each tree.

Table 6: Hyperparameters for XGBoost algorithm.

5.4 Neural Network

The key goal of the project is to build a neural network (AI based approach) to replicate the decision making of player and predict system dynamics (non-linear behaviour pursuant to farming and fallowing). The aim is to understand if incentives (i.e. subsidy) could influence farmers' land-use decision or not. As farming goes digital, machine learning proves to be effective to solve complicated tasks such as yield prediction in the future, estimated harvest time et al. But using deep learning as an approach to develop a probability model emulates complex functions; one of which is decision-making.

5.4.1 Deep learning

Deep learning is the use of powerful neural networks to model complex non-linear relationships. The main reason behind Deep Learning is the idea that Artificial Intelligence should inspire the human brain. Deep neural network can learn from hierarchy of layers that identify the input features and creates new features based on the dataset, just as human brain. In neural network, there are three types of layers which have been explained as follows in lieu of the project under consideration:

a) Input layer:

The "Input layer" includes neurons that do nothing but receive inputs and pass them to other layers. The number of layers in the input layer must be equal to the "attributes" or "features" in the dataset.

Method 1: Building model for each cell- In this case, there are 3 input layers i.e. Subsidy, History and Neighbour.

Method 2: Building a single model for all cells- In this case, there are 19 input layers i.e. 9 for neighbour, 9 for history and 1 for subsidy.

b) Output layer:

The "Output Layer" is a predicted property; this layer depends essentially on the type of model being built.

Method 1: Building model for each cell- In this case, there is one output i.e. choice.

Method 2: Build a single model for all cells: In this case, there are 9 outputs i.e. choice for each cell (9 cells).

c) Hidden layer:

Between the input and output layers, the layers that exist are termed as the "hidden layers" which depends upon on the model being built. The hidden layers include a wide range of neurons. The neurons in the hidden layer apply transformations before they pass the inputs on them. By training the network, the weights are updated to be more prominent.

To find out the correct number of neurons to use in the hidden layer, various rule-of-thumb methods to determine it have been used such as the following:

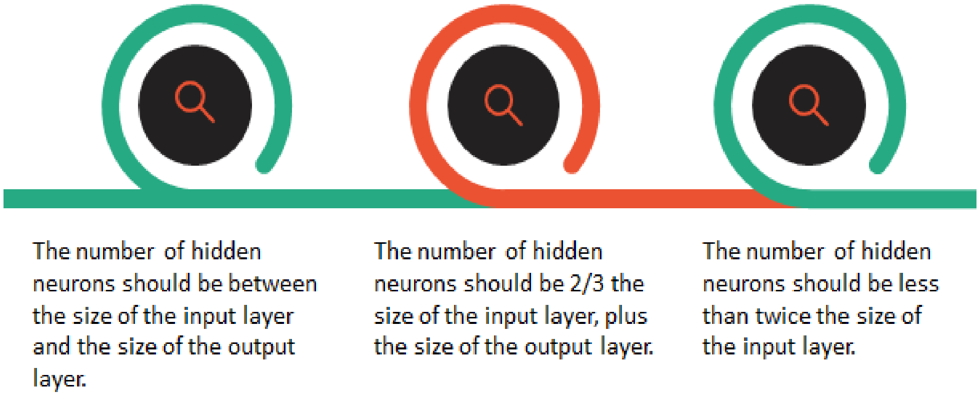

The number of hidden neurons should be between the size of the input layer and the size of the output layer.
The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
The number of hidden neurons should be less than twice the size of the input layer.

Figure 10: Rule-of-thumb methods to find out number of hidden layers

5.4.2 Hyperparameters for Deep Learning Neural Network

These are the hidden elements that can be tuned to control the behaviour an algorithm and align the model accuracy to the project/business goals. Hyperparameters are the values that are set before training occurs. Therefore, anytime we refer to a parameter as being manually set, we are referring to hyper-parameters. Hyper-parameter tuning consists of selecting hyper-parameter to test and then running a specific type of model with hyper-parameters. For each time we run the model, we keep track of how well the model did for a specify accuracy metric, as well as keep track of the hyper-parameters that were used. One of the hardest parts of this process is selecting the right hyper-parameter to tune and specifying the appropriate value ranges for each hyper-parameter.

There are two ways to optimize the hyperparameters for models:

a) **Grid Search:** In this method, combination of parameters from the list of all parameters is taken into consideration to run the model.

b) **Random Search:** This method is used to combine the parameters. KerasClassifier (scikit-learn API) is used to run a random search with Keras and RandomizedSearchCV class for cross validation (for evaluating the model) random search.

For the dataset under consideration, the Grid Search method has been chosen to find the value in a fastest way with the fewest function evaluations.

The hyperparameters for neural network has been explained below:

a) **Neurons:**

The core concept of the neural networks is "artificial neurons," imitating human brain neurons. These neurons are simple and powerful computing units which have input-weighted signals that generate an output signal using an activation function. Neurons are released in several layers in the neural network.

b) Learning ratio:

The amount of cost reduction per repetition is called learning. Simply put, the cost-cutting speed is the same as learning. The learning ratio has to be carefully selected so that it does not go far enough to reject the optimal mode and not to the extent that the network learning takes years.

c) Batch:

When we teach a neural network, instead of sending the entire input, we divide it into small packages of the same size. When we send data as batches, it makes the model more comprehensive than the model that received all the information at one place.

d) Epoch:

A period (Epoch) refers to a back propagation in the network i.e. a period equal to an input sweep across the entire network. The number of courses you use to learn the network is completely in your own hands. It is true that having more courses leads to more precision in the network, but also increases network learning time. In addition, it should be noted that if the number of courses is too high, it may not be digested in a heavy network, or so-called "over-fit" network.

e) Activation function:

The Activation Function is the total weight Input that maps to the output of the neuron. For this reason, the transfer function is said to control the start of the activation of the neuron and give the power to the output signal. It is represented as:

$$Y=\sum(\text{weight}*\text{input})+\text{bias}$$

There are many activation passages among which there are- "rectified linear unit" (Rectified Linear Unit , ReLU), (tanh) and (SoftPlus) are mostly used.

For the given dataset, Sigmoid and Rectified Linear Unit (ReLU) hyperparameters have been tuned because task is of classification nature.

f) Cost function and Gradient Decent:

The "Cost Function" of the gauge is to determine whether a nerve skeleton has been used for the training set and how good is the expected output. This function also depends on features such as "weights" and "biases". The cost function is a single value and does not exclude, since this function is a function of the goodness of the function of the neural network as a whole. Using gradient optimization algorithms, weights gradually increase after each epoch.

In mathematical terms, "Sum of Squared Errors | SSE" is calculated as follows:
J(W)=1/2

The amount and direction of weight update are calculated by taking a step in the opposite direction to the cost gradient as represented follows:

$$\Delta \mathrm{wj} = f(z) = \frac{1}{2}\sum_{i=0}^{\infty}((\mathrm{target})^{i} - (\mathrm{output})^{1})^{2}$$

Where Δw is a vector containing the updated weights for each coefficient of weight w, which is computed as follows:

$$\Delta \mathrm{wj} = f(z) = \frac{1}{2}\sum_{i=0}^{\infty}((\mathrm{target})^{i} - (\mathrm{output})^{i})X_{j}{}^{2}$$

5.5 Neural networks libraries in python used for this project

5.5.1 Keras

"Keras" is a high-level library used for "Neural Networks", it was created by Francois chollet. *Keras* runs over the TensorFlow library and is developed by Google. *Keras* is high-level Deep learning framework. It is an open source deep learning library that enables fast experimentation with neural networks. It also runs on top of other frameworks like TensorFlow, Theano or CNTK.

The question here is why use keras instead of other low-level libraries like TensorFlow?

With keras we can build industry-ready model in less time with less code, it’s suitable for both beginners and experts. It allows for quickly and easily checking if using a neural network solves the problem in question or not. With keras, we can build any architecture, from simple network to more complex ones .Keras’ model also can deploy in multiple platforms. Keras user Interface is an API designed for humans, not machines. This user experience opens up and down the center. Keras follows best practices for reducing cognitive load: it provides consistent and simple APIs, minimizes the number of user actions required for common use, and provides clear and enforceable feedback on user error. Slowly, the modularity of a model is understood as a sequence or a graph of independent and fully customizable modules that can be connected with limited constraints and may be attached. Specifically, the nerve layers, cost functions, optimizers, initialization schemes, activation functions and settings are all independent modules that you can combine to create new models. The easy development of new modules to add (as new classes and functions) is simple, and existing modules offer many examples. To make it easy to create new modules, it's fully explained and makes Keras suitable for advanced research. Working with Python, there are no configuration files for separate models in a notice template. The models are described in Python code, which are more compact and easier to debug and allow for ease of development.(https://keras.io/)

5.5.2 Multilayer perceptron (MLPClassifier)

A multi-layer perceptron MLP is a feed-forward artificial neural network model where a set of input data is mapped onto a set of appropriate outputs. It consists of multiple layers or nodes in where each layer is fully connected to the next one in the form of a directed graph. MLPRegressor can be used for regression problems and MLPClassifier can be used for classification problems.

5.6 Model validation

In machine learning, we cannot solely rely on trained model to assume that the same model is going to work well on test data as well. Thus, there is a need to validate our model for the assurance of the accuracy of the predictions made by the training model on the unseen data. This process of deciding whether the results (that quantify hypothesised relationships between variables) generated by the trained model are acceptable as descriptions of data or not acceptable is known as validation.

For performance evaluation of the model, it is usually tested on unseen data to analyse whether the model is under-fitted/Over-fitted/Well-generalised. To test the effectiveness of a machine learning model, Cross validation (CV) is one of the technique which is performed by keeping a sample/portion of the data aside which is not used for training the model but is used later for testing/validating.

<u>The techniques used for cross validation for dataset in hand are described as follows:</u>

a) **Train_Test Split approach:**

Out of the total 4600 instances in the dataset, 70% data has been used as training dataset and 30% is used as test dataset. The task has been accomplished using two methods: first, for each cell, the features and targets have been extracted. For instance, in cell1, we have three columns i.e. subsidy, neighbor_c1 and I_Hist_c1 as features and one column i.e.I_choice_c1 as target. In second method, the first 19 columns have been extracted as features columns and the last 9 columns (choice columns) as target.

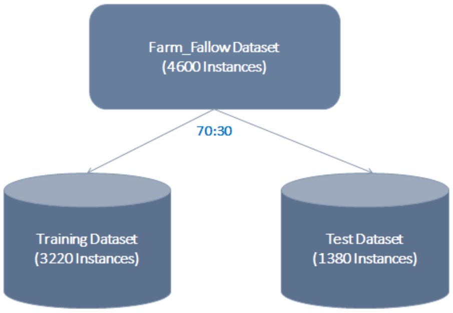


Figure 11: Train/Test split approach

b) **K-Folds Cross Validation:**

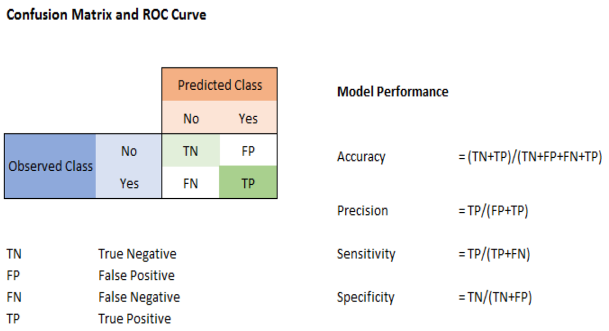
To achieve model results with less amount of bias, K-Fold is the best and easy to use method because it ensures that every observation has a chance of appearing in the training and test dataset from the original dataset.

5.7 Performance Metrics

To evaluate a model, its performance is measured against some metric and the metric used for the models for dataset under consideration is confusion matrix because accuracy is not a reliable metric as it can give misleading results.

5.7.1 Confusion matrix

In the "Classification" discussion of a "Data Set" using categorization methods, the goal is to achieve the highest possible accuracy in categorizing and recognizing categories. In such cases, the precision of detecting a bunch is more important than the overall detection accuracy; the concept of "confusion matrix" comes in handy. The collation table or matrix displays the classification results based on the actual information available as shown in the picture below:



Confusion Matrix and ROC Curve

		Predicted Class	
		No	Yes
Observed Class	No	TN	FP
	Yes	FN	TP

TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

Model Performance

Accuracy	=(TN+TP)/(TN+FP+FN+TP)
Precision	=TP/(FP+TP)
Sensitivity	=TP/(TP+FN)
Specificity	=TN/(TN+FP)

Figure 12: Analysis of confusion matrix

- The sensitivity criterion, which is also called "True Positive Rate." Sensitivity means a proportion of the positive cases that classifiers have correctly identified as positive.
- In some cases, it may be important to accurately detect a negative class. Of the most commonly used parameters, which are usually considered alongside sensitivity,is the Specificity parameter, which is also called True Negative Rate. The property means a proportion of the negative cases that tested them correctly as a negative example.
- There is another important parameter called F-Measure, which is very useful for assessing the performance of clusters and combining two sensitivity and positive predictive values. With the explanation that the parameter of positive predictive value is called **precision**, and sensitivity is called **recall**, The " F-Measure " is defined as:

*F-measure= 2 * (Recall * Precision) / (Recall + Precision)*

6. MODEL TRAINING AND TUNING

For the following section, GridSearch method has been used to find the right hyperparameters and a 5-fold Cross-Validation has been used throughout the modelling to achieve the reliable results. The below function has been used to achieve the same:

grid=GridSearchCV(estimator=model,param_grid=param_grid,cv=5)

grid_result = grid.fit(X_train,y_train)

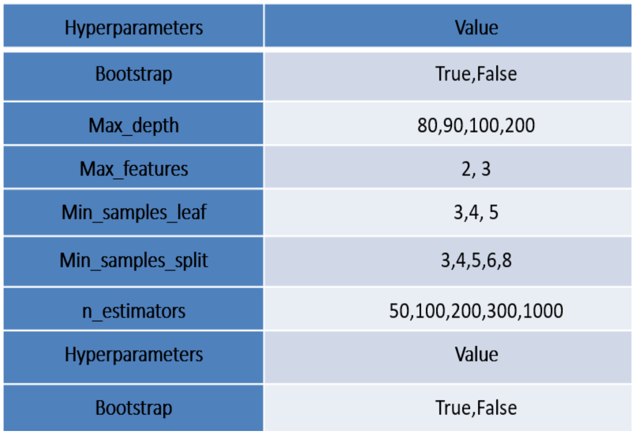
The hyperparameters and their value have been stored as a dictionary as '*param_grid=dict'*.

6.1 Machine Learning

Random Forest and XGBoost algorithms machine learning algorithms have been used to predict the outcome variable (dependent variable) from a given set of predictors (independent variables). The training process continues until models are properly tuned to achieve best accuracy. Thus, learning process is controlled through hyperparameter tuning. The hyperparameter tuning for the above machine learning algorithms to achieve the desired results have been explained in the following section:

6.1.1 Random Forest

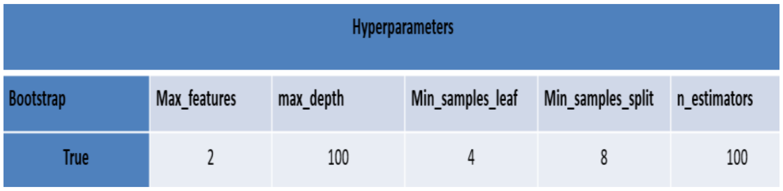
The hyperparameters were tuned as follows:



Hyperparameters	Value
Bootstrap	True,False
Max_depth	80,90,100,200
Max_features	2, 3
Min_samples_leaf	3,4, 5
Min_samples_split	3,4,5,6,8
n_estimators	50,100,200,300,1000
Hyperparameters	Value
Bootstrap	True,False

Table 7*:* Various Combinations of Hyperparameter values for Random Forest

With 69% accuracy, the following values of chosen hyperparameters were used to build the final model:



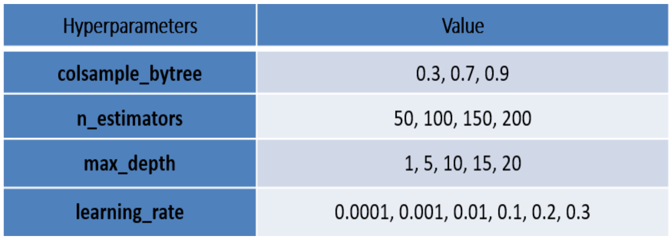
Hyperparameters					
Bootstrap	Max_features	max_depth	Min_samples_leaf	Min_samples_split	n_estimators
True	2	100	4	8	100

Table 8: Random Forest Hyperparameters for building the model

6.1.2 XGBoost Classifier

The hyperparameters for this classifier were tuned as follows:

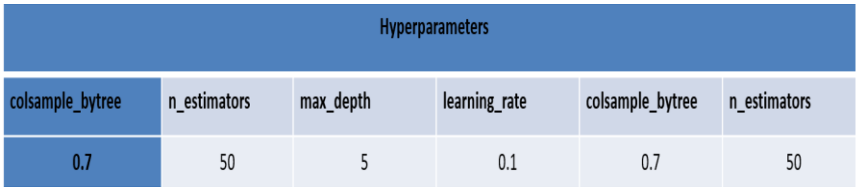
Using Grid search method, following right hyperparameters have been find out and tuned to achieve the model training results (Appendix 1).



Hyperparameters	Value
colsample_bytree	0.3, 0.7, 0.9
n_estimators	50, 100, 150, 200
max_depth	1, 5, 10, 15, 20
learning_rate	0.0001, 0.001, 0.01, 0.1, 0.2, 0.3

Table 9: Various Combinations of Hyperparameter values for XGBoostclassifier

With 71% accuracy, the following values of chosen hyperparameters were used to build the final model:



Hyperparameters					
colsample_bytree	n_estimators	max_depth	learning_rate	colsample_bytree	n_estimators
0.7	50	5	0.1	0.7	50

Table 10*:* XGBoost Hyperparameters for building the model

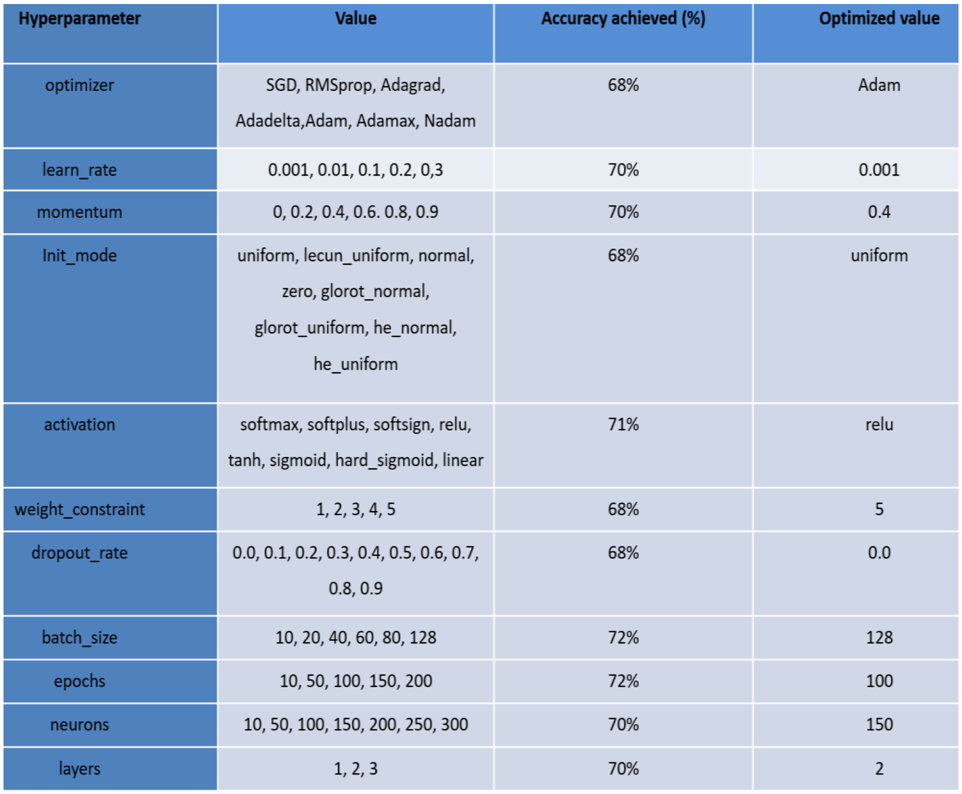
6.2 Deep Learning

Keras Classifier and MLP deep learning algorithms have been used to predict the outcome variable (dependent variable) from a given set of predictors (independent variables). The training process continues until models are properly tuned to achieve best accuracy. Thus, learning process is controlled through hyperparameter tuning.

Out of these two classifiers, MLP is a classical deep learning classifier but Keras is more advanced one. The technique used to find out hyperparameters for Keras classifier was to find out optimum value of one hyperparameter and then use the appropriate hyperparameter along with the value found for it to further discover the other hyperparameters. For Instance, learn_rate=0.001 was found using Optimizer=Adam as shown in the table below. But for MLP classifier, all the hyperparameters were l found in a parallel way. The hyperparameter tuning for the deep learning algorithms to achieve the desired results have been explained in the following section:

6.2.1 Keras Classifier

The hyperparameters for Keras classifier were tuned using the technique described in the above section. The description of the chosen hyperparameters to build a model has been stated in the below table:

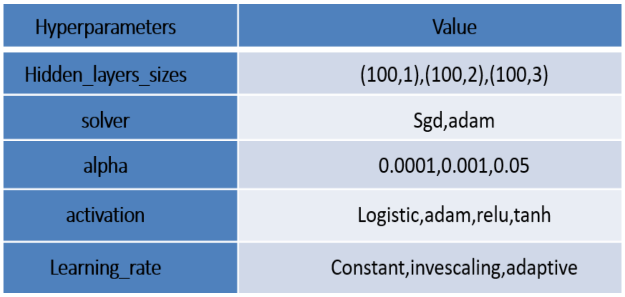


Hyperparameter	Value	Accuracy achieved (%)	Optimized value
optimizer	SGD, RMSprop, Adagrad, Adadelta,Adam, Adamax, Nadam	68%	Adam
learn_rate	0.001, 0.01, 0.1, 0.2, 0,3	70%	0.001
momentum	0, 0.2, 0.4, 0.6. 0.8, 0.9	70%	0.4
Init_mode	uniform, lecun_uniform, normal, zero, glorot_normal, glorot_uniform, he_normal, he_uniform	68%	uniform
activation	softmax, softplus, softsign, relu, tanh, sigmoid, hard_sigmoid, linear	71%	relu
weight_constraint	1, 2, 3, 4, 5	68%	5
dropout_rate	0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9	68%	0.0
batch_size	10, 20, 40, 60, 80, 128	72%	128
epochs	10, 50, 100, 150, 200	72%	100
neurons	10, 50, 100, 150, 200, 250, 300	70%	150
layers	1, 2, 3	70%	2

Table 11: Kerasclassifier Hyperparameters for building the model

6.2.2 Multilayer Perceptron

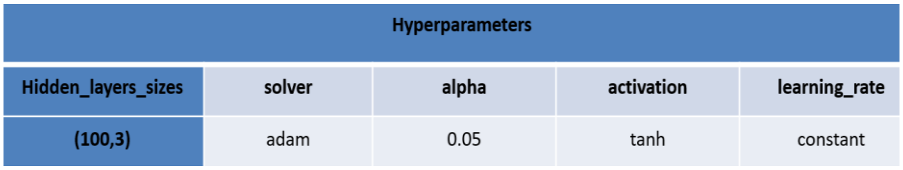
The hyperparameters of MLP classifier were tuned as follows:



Hyperparameters	Value
Hidden_layers_sizes	(100,1),(100,2),(100,3)
solver	Sgd,adam
alpha	0.0001,0.001,0.05
activation	Logistic,adam,relu,tanh
Learning_rate	Constant,invescaling,adaptive

Table 12: various Combinations of Hyperparameter values for MLP

With 68% accuracy, the following values of chosen hyperparameters were used to build the final model:



Hyperparameters				
Hidden_layers_sizes	**solver**	**alpha**	**activation**	**learning_rate**
(100,3)	adam	0.05	tanh	constant

Table 13: MLP Hyperparameters for building the model

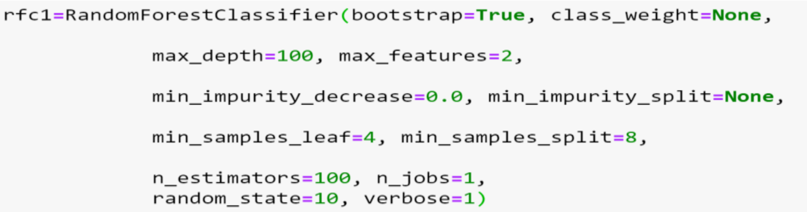
7. MODEL RESULT AND EVALUATION OF THE CLASSIFIERS

As two methods have been used to build a model (as explained in modelling section), model results attained using both methods have been explained in the following section:

7.1 Machine Learning algorithms

7.1.1 Random Forest

To build model using Random Forest, *RandomForestClassifier* has been imported from sklearn package in python and after optimum hyperparameter tuning as explained in above chapter, the model has been built. The dataset has been trained using model as shown in the picture below:



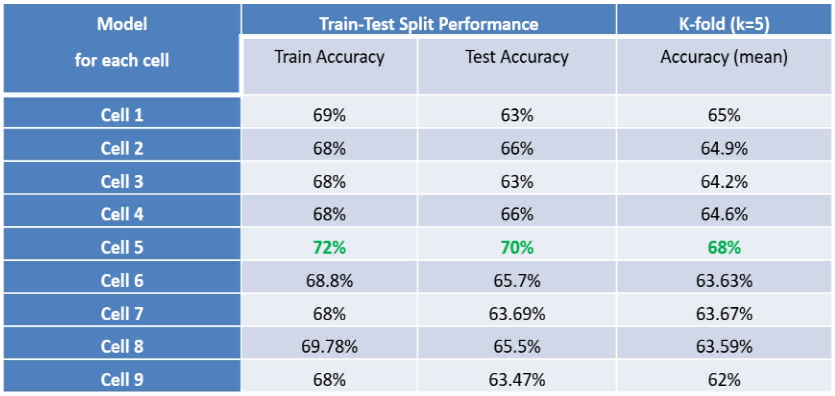
```
rfc1=RandomForestClassifier(bootstrap=True, class_weight=None,

            max_depth=100, max_features=2,

            min_impurity_decrease=0.0, min_impurity_split=None,

            min_samples_leaf=4, min_samples_split=8,

            n_estimators=100, n_jobs=1,
            random_state=10, verbose=1)
```

Figure 13: Model for Random Forest

I) Building model for each cell- Using this method, machine learning model has been built for each cell as shown in the result section below:

A) Results:

When the model was built for each cell, the best model was selected based on its high train-test split and cross validation performance because the successful model is the one which not only works best on the data, it is trained upon but also on the unseen data and should adequately be capable of getting tailored to the future data.



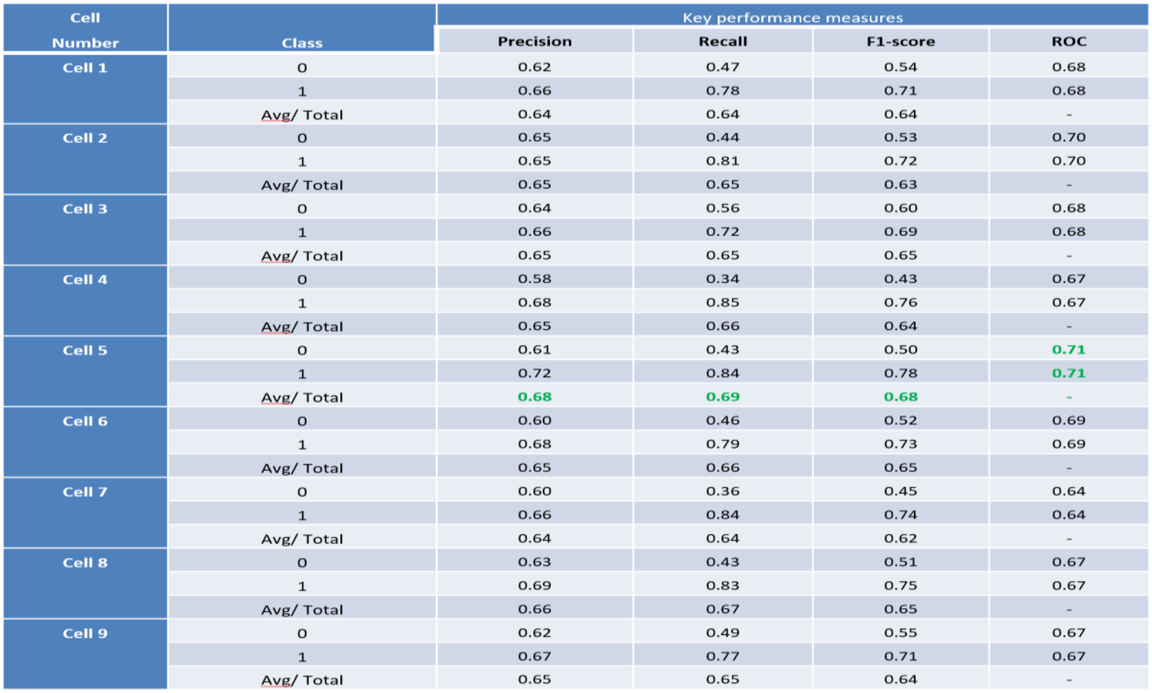
Model for each cell	Train-Test Split Performance		K-fold (k=5)
	Train Accuracy	Test Accuracy	Accuracy (mean)
Cell 1	69%	63%	65%
Cell 2	68%	66%	64.9%
Cell 3	68%	63%	64.2%
Cell 4	68%	66%	64.6%
Cell 5	72%	70%	68%
Cell 6	68.8%	65.7%	63.63%
Cell 7	68%	63.69%	63.67%
Cell 8	69.78%	65.5%	63.59%
Cell 9	68%	63.47%	62%

Table 14: Random Forest Result for one model for each cell

The result in the above table shows that model built for cell5 has a better accuracy than for other cells (Appendix 2). This model is appropriate to be used for land-use decision making by the farmers.

B) Performance Metrics: Confusion Matrix

As explained in section 5.8, we cannot solely reply upon accuracy to prove that the particular model is the best one and need to measure it against metrics beyond accuracy (see table below).



Cell Number	Class	Key performance measures			
		Precision	Recall	F1-score	ROC
Cell 1	0	0.62	0.47	0.54	0.68
	1	0.66	0.78	0.71	0.68
	Avg/ Total	0.64	0.64	0.64	-
Cell 2	0	0.65	0.44	0.53	0.70
	1	0.65	0.81	0.72	0.70
	Avg/ Total	0.65	0.65	0.63	-
Cell 3	0	0.64	0.56	0.60	0.68
	1	0.66	0.72	0.69	0.68
	Avg/ Total	0.65	0.65	0.65	-
Cell 4	0	0.58	0.34	0.43	0.67
	1	0.68	0.85	0.76	0.67
	Avg/ Total	0.65	0.66	0.64	-
Cell 5	0	0.61	0.43	0.50	0.71
	1	0.72	0.84	0.78	0.71
	Avg/ Total	0.68	0.69	0.68	-
Cell 6	0	0.60	0.46	0.52	0.69
	1	0.68	0.79	0.73	0.69
	Avg/ Total	0.65	0.66	0.65	-
Cell 7	0	0.60	0.36	0.45	0.64
	1	0.66	0.84	0.74	0.64
	Avg/ Total	0.64	0.64	0.62	-
Cell 8	0	0.63	0.43	0.51	0.67
	1	0.69	0.83	0.75	0.67
	Avg/ Total	0.66	0.67	0.65	-
Cell 9	0	0.62	0.49	0.55	0.67
	1	0.67	0.77	0.71	0.67
	Avg/ Total	0.65	0.65	0.64	-

Table 15: Performance metrics for Random Forest

ROC curve is usually plotted with TPR (True Positive Rate) on y-axis against FPR (False Positive Rate) on x-axis. The ROC curve for cell 5 which has better accuracy than other cells (Appendix 3) is displayed in the below picture. An excellent model has ROC near to 1 and cell 5 has the value closet to 1 than other cells i.e. 0.71. Also average precision, recall and F1-score is higher than the value for other cells.

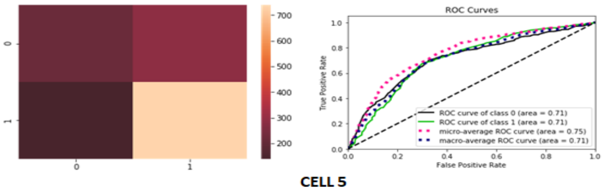


Figure13: Confusion Matrix and ROC for cell 5

II) Building a single model for all cells- The model results based on train-test and cross validation performance and evaluation in terms of set performance metrics has been explained in below sections.

A) Results

Using Random Forest classifier, the training accuracy of 81% has been achieved which means that model works well on trained data and when checked on test data, 61% accuracy was achieved. With cross validation approach, 60% was achieved as shown in table below.

Model for each cell	Train-Test Split Performance		K-fold (k=5)
	Train Accuracy	Test Accuracy	Test standard deviation
Cell 1 to Cell 9	0.81	0.61	0.60

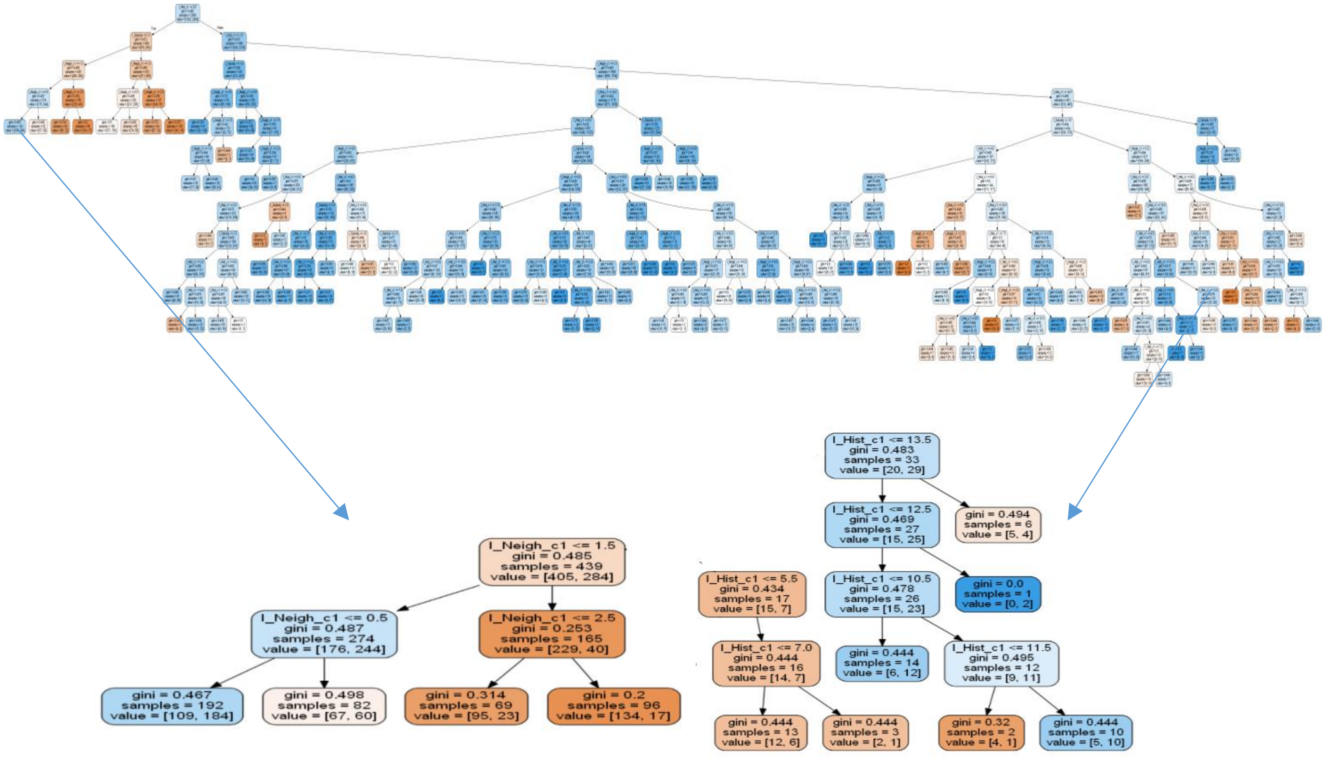
Table 16: Random Forest Result for one model for all cells

B) Performance Metrics: Confusion Matrix

The below table shows the accuracy/performance evaluation of the model built, in terms of precision, recall and F1-score.

Cell Number	Output	Key performance measures		
		Precision	Recall	F1-score
Cell 1 to Cell9	1	0.67	0.68	0.68
	2	0.67	0.77	0.67
	3	0.66	0.58	0.62
	4	0.71	0.73	0.72
	5	0.71	0.79	0.75
	6	0.73	0.65	0.69
	7	0.66	0.74	0.70
	8	0.68	0.73	0.70
	9	0.66	0.63	0.64
	Avg/ Total	0.688	0.692	0.685

Table 17: Performance metrics for one model for all cells

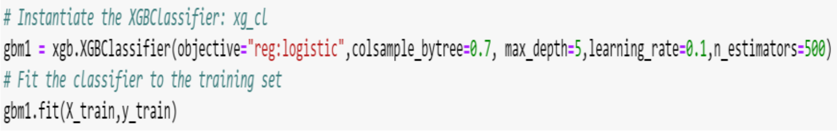


I_Neigh_c1 <= 1.5 gini = 0.485 samples = 439 value = [405, 284]
I_Neigh_c1 <= 0.5 gini = 0.487 samples = 274 value = [176, 244]
I_Neigh_c1 <= 2.5 gini = 0.253 samples = 165 value = [229, 40]
gini = 0.467 samples = 192 value = [109, 184]
gini = 0.498 samples = 82 value = [67, 60]
gini = 0.314 samples = 69 value = [95, 23]
gini = 0.2 samples = 96 value = [134, 17]
I_Hist_c1 <= 13.5 gini = 0.483 samples = 33 value = [20, 29]
I_Hist_c1 <= 12.5 gini = 0.469 samples = 27 value = [15, 25]
gini = 0.494 samples = 6 value = [5, 4]
I_Hist_c1 <= 5.5 gini = 0.434 samples = 17 value = [15, 7]
I_Hist_c1 <= 10.5 gini = 0.478 samples = 26 value = [15, 23]
gini = 0.0 samples = 1 value = [0, 2]
I_Hist_c1 <= 7.0 gini = 0.444 samples = 16 value = [14, 7]
gini = 0.444 samples = 14 value = [6, 12]
I_Hist_c1 <= 11.5 gini = 0.495 samples = 12 value = [9, 11]
gini = 0.444 samples = 13 value = [12, 6]
gini = 0.444 samples = 3 value = [2, 1]
gini = 0.32 samples = 2 value = [4, 1]
gini = 0.444 samples = 10 value = [5, 10]

Figure 14: Visualization of Random Forest

7.1.2 XGBoost Classifier

To build model using XGBoost Classifier, *XGBClassifier* has been imported from sklearn package in python and after optimum hyperparameter tuning as explained in above chapter, the model has been built. The dataset has been trained using model as shown in the picture below:



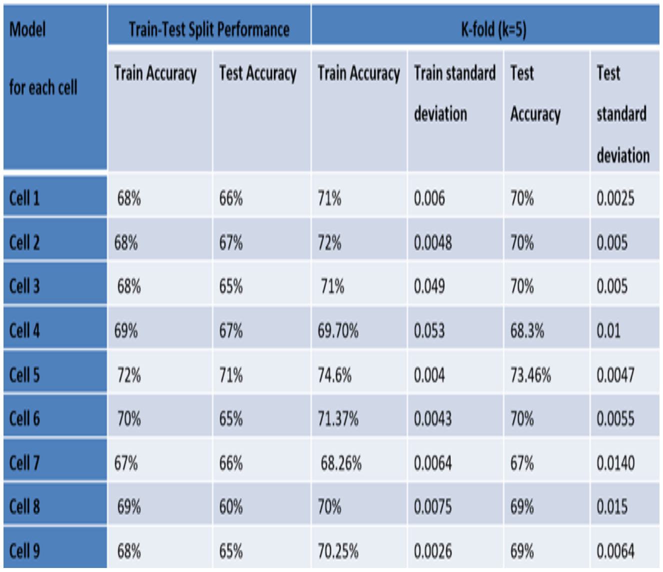
```
# Instantiate the XGBClassifier: xg_cl
gbm1 = xgb.XGBClassifier(objective="reg:logistic",colsample_bytree=0.7, max_depth=5,learning_rate=0.1,n_estimators=500)
# Fit the classifier to the training set
gbm1.fit(X_train,y_train)
```

Figure 15: Model for XGBoostClassifier

I) Building model for each cell- Using this method, machine learning model has been built for each cell as shown in the result section below:

A) Results

The table below shows the accuracy achieved through train-test split and cross validation approach. The training accuracy ranges between 67%-72% from Cell 1 to Cell 9; model for Cell5 being the highest performer, test accuracy ranges between 60%-71%. In case of cross validation approach, training and test accuracy ranges between 68.26% -74.6% and 67%-73.46% respectively. The standard deviation for training accuracy in K-fold approach ranges between 0.004-0.053 which means that the models have few outliers because the range of standard deviation is close to 1 that further demonstrates that data points are close to the mean.



Model for each cell	Train-Test Split Performance		K-fold (k=5)			
	Train Accuracy	Test Accuracy	Train Accuracy	Train standard deviation	Test Accuracy	Test standard deviation
Cell 1	68%	66%	71%	0.006	70%	0.0025
Cell 2	68%	67%	72%	0.0048	70%	0.005
Cell 3	68%	65%	71%	0.049	70%	0.005
Cell 4	69%	67%	69.70%	0.053	68.3%	0.01
Cell 5	72%	71%	74.6%	0.004	73.46%	0.0047
Cell 6	70%	65%	71.37%	0.0043	70%	0.0055
Cell 7	67%	66%	68.26%	0.0064	67%	0.0140
Cell 8	69%	60%	70%	0.0075	69%	0.015
Cell 9	68%	65%	70.25%	0.0026	69%	0.0064

Table 18: Performance metrics for XGBoost classifier for all cells

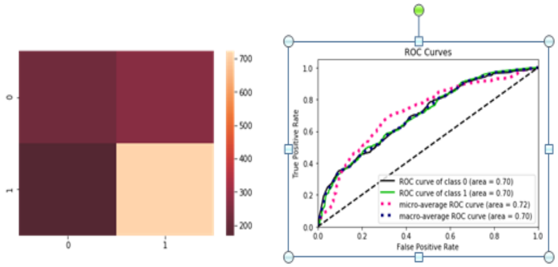
B) Performance Metrics: Confusion Matrix:

Further the accuracy analysis in terms of precision, recall and F1-score along with plotting of ROC curve has been shown in table below:

Cell Number	Class	Key performance measures			
		Precision	Recall	F1-score	ROC
Cell 1	0	0.68	0.40	0.50	0.68
	1	0.64	0.86	0.73	0.68
	Avg/ Total	0.66	0.65	0.63	-
Cell 2	0	0.70	0.45	0.54	0.70
	1	0.66	0.84	0.74	0.70
	Avg/ Total	0.67	0.67	0.65	-
Cell 3	0	0.68	0.40	0.50	0.67
	1	0.61	0.84	0.71	0.67
	Avg/ Total	0.65	0.63	0.61	-
Cell 4	0	0.60	0.36	0.45	0.69
	1	0.71	0.86	0.78	0.69
	Avg/ Total	0.67	0.68	0.66	-
Cell 5	0	0.57	0.45	0.50	0.70
	1	0.73	0.81	0.77	0.70
	Avg/ Total	0.67	0.68	0.67	-
Cell 6	0	0.60	0.44	0.51	0.69
	1	0.68	0.80	0.73	0.69
	Avg/ Total	0.65	0.66	0.64	-
Cell 7	0	0.63	0.37	0.47	0.65
	1	0.65	0.84	0.74	0.65
	Avg/ Total	0.64	0.65	0.63	-
Cell 8	0	0.62	0.38	0.47	0.67
	1	0.64	0.82	0.72	0.67
	Avg/ Total	0.63	0.63	0.61	-
Cell 9	0	0.64	0.40	0.50	0.66
	1	0.61	0.81	0.70	0.66
	Avg/ Total	0.63	0.62	0.60	-

Table 19: Performance metrics for XGBoost Classifier

ROC curve is usually plotted with TPR (True Positive Rate) on y-axis against FPR (False Positive Rate) on x-axis. The ROC curve for cell 5 which has better accuracy than other cells (Appendix 4) is displayed in the below picture. An excellent model has ROC near to 1 and cell 5 has the value closet to 1 than other cells i.e. 0.70. Also average precision, recall and F1-score is higher than the value for other cells.



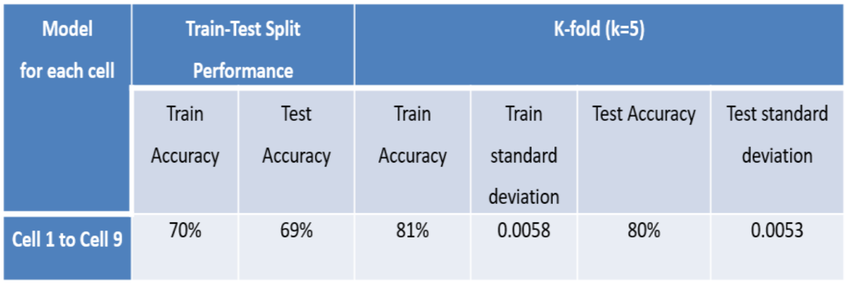
CELL 5

Figure 16: Confusion Matrix and ROC for cell 5

II) Building a single model for all cells-

A. Results

Using this method, training accuracy of 70%, test accuracy of 69% was achieved using train-test split approach. Also, training and test accuracy of 81% and 80% respectively has been achieved. The standard deviation= 0.0053 which portrays that data points are close to the mean and only few outliers exist.

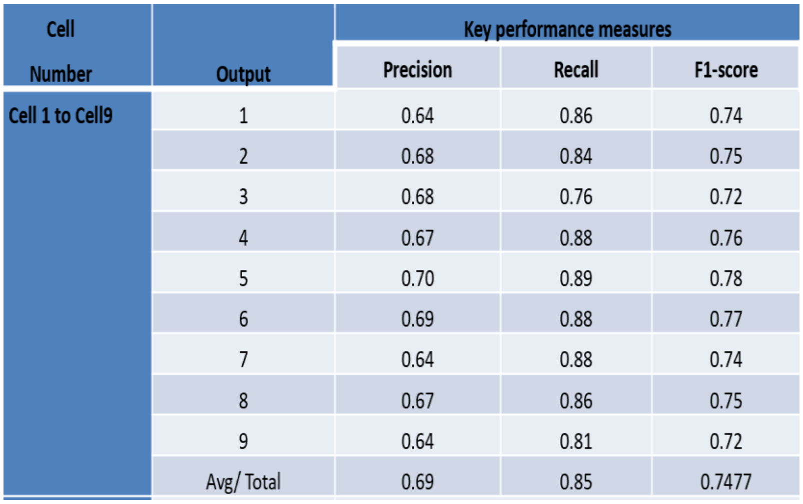


Model for each cell	Train-Test Split Performance		K-fold (k=5)			
	Train Accuracy	Test Accuracy	Train Accuracy	Train standard deviation	Test Accuracy	Test standard deviation
Cell 1 to Cell 9	70%	69%	81%	0.0058	80%	0.0053

Table 20: XGBoost classifier Result for one model for all cells

B. Performance Metrics: Confusion Matrix

The evaluation of accuracy achieved by the model in terms of Precision, recall and F1-score has been shown in table below. The F1- score (i.e. weighted average of precision and recall) ranges between 0.72-0.78 (very close to 1) which means that the classifier has correctly identified positive results to attain the true positive rate.



Cell Number	Output	Key performance measures		
		Precision	Recall	F1-score
Cell 1 to Cell9	1	0.64	0.86	0.74
	2	0.68	0.84	0.75
	3	0.68	0.76	0.72
	4	0.67	0.88	0.76
	5	0.70	0.89	0.78
	6	0.69	0.88	0.77
	7	0.64	0.88	0.74
	8	0.67	0.86	0.75
	9	0.64	0.81	0.72
	Avg/ Total	0.69	0.85	0.7477

Table 21*:* XGBoostclassifier Performance metrics for one model for all cells

7.2 Conclusion for Machine Learning approaches

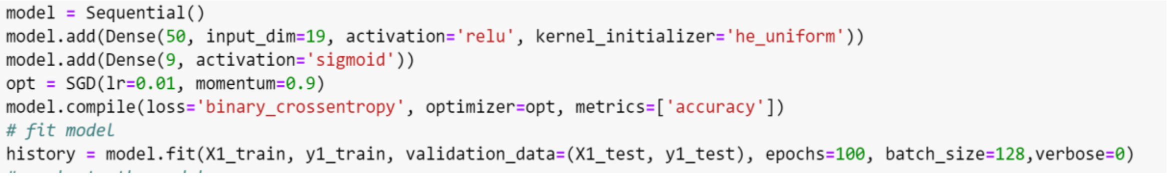
After analyzing the results of the machine learning approach using Random Forest and XGBoost classifier on the basis of two different methods, it is recommended to use XGBoost classifier as the accuracy results of this model outperforms the results achieved by Random Forest model. XG attained higher accuracy than random forest in case of both methods: 67%-73.46% in case of building model for all cells (method1) and 80% in case of building a single model for all cells whereas it is 62%-68% in case of method1 and 60% in case of method2 for random forest model.

Hence, for both methods to attain high classification accuracy in order to influence farmers' behaviour for farming or fallowing, XGB classifier can adequately predict farmers' decision in terms of land-use by correctly identifying the positive class which means high true positive rate.

7.3 Deep Learning algorithms

7.3.1 Keras Classifier

To build model using Keras, Keras Classifier, Dense, Activation, SGD.Adam, relu, sigmoid, model has been imported from keras.wrappers.scikit_learn package, kera.optimizer, keras.layers, keras.models packages in python and after optimum hyperparameter tuning as explained in above chapter, the model has been built. The dataset has been trained using model as shown in the picture below:



```
model = Sequential()
model.add(Dense(50, input_dim=19, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(9, activation='sigmoid'))
opt = SGD(lr=0.01, momentum=0.9)
model.compile(loss='binary_crossentropy', optimizer=opt, metrics=['accuracy'])
# fit model
history = model.fit(X1_train, y1_train, validation_data=(X1_test, y1_test), epochs=100, batch_size=128,verbose=0)
```

Figure 17*:* Model for Keras Classifier

A) Results

I) Building model for each cell- Using this method, the model has been built for each cell as shown in the table below. The neural network built using this method has been visualized using nnv package in python is shown below:

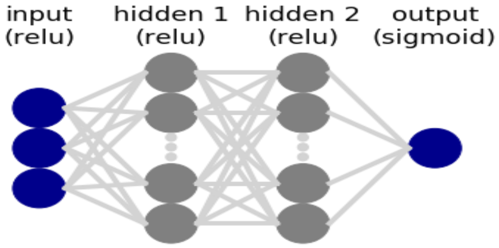
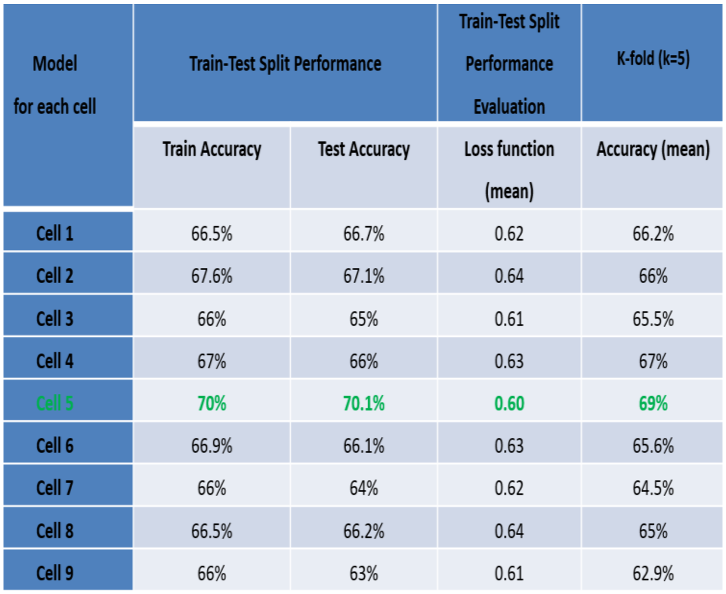


Figure 18: Neural network using Keras Classifier

In the above diagram, when the neural network has been built for each cell, the network has 3 input layers, two hidden layers and one output layer. After adequate training of the model, the results achieved are as follows:



Model for each cell	Train-Test Split Performance		Train-Test Split Performance Evaluation	K-fold (k=5)
	Train Accuracy	Test Accuracy	Loss function (mean)	Accuracy (mean)
Cell 1	66.5%	66.7%	0.62	66.2%
Cell 2	67.6%	67.1%	0.64	66%
Cell 3	66%	65%	0.61	65.5%
Cell 4	67%	66%	0.63	67%
Cell 5	70%	70.1%	0.60	69%
Cell 6	66.9%	66.1%	0.63	65.6%
Cell 7	66%	64%	0.62	64.5%
Cell 8	66.5%	66.2%	0.64	65%
Cell 9	66%	63%	0.61	62.9%

Table 22: Keras Classifier Result

The above table shows that there is not much difference in the train-test split performance approach and k-fold method which ranges between 66%-70% and 63%-70.1% for both training and test accuracy respectively in case of train-test split approach and 69% accuracy in case of K-fold approach. It means that the model is neither over-fitted nor under-fitted. The cell 5 performs better in terms of high accuracy and low loss function (69% accuracy and 0.60 loss function) than other cells (Appendix1) as shown in the below figure which means that cell5 is a better classifier to predict the given set of predictors in the future.

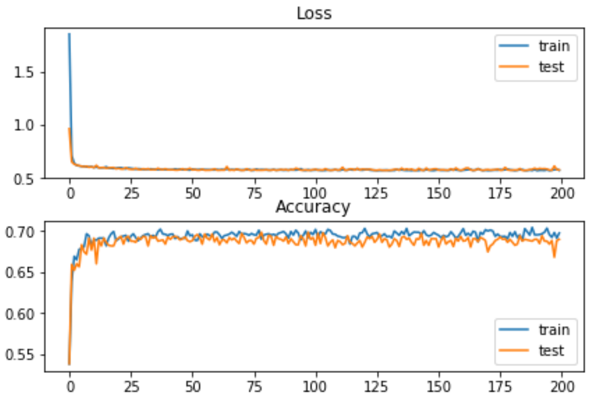


Figure 19: The result of model based on cell number 5

C) Performance Metrics: Confusion Matrix

The below table evaluate the accuracy achieved by keras classifier in term sof precision, recall and F1-score.

Cell Number	Class	Key performance measures		
		Precision	Recall	F1-score
Cell 1	0	0.72	0.45	0.55
	1	0.66	0.86	0.75
	Avg/ Total	0.69	0.68	0.66
Cell 2	0	0.66	0.46	0.54
	1	0.67	0.82	0.74
	Avg/ Total	0.67	0.67	0.66
Cell 3	0	0.68	0.43	0.53
	1	0.64	0.84	0.72
	Avg/ Total	0.66	0.65	0.63
Cell 4	0	0.65	0.26	0.38
	1	0.67	0.92	0.78
	Avg/ Total	0.67	0.67	0.63
Cell 5	0	0.61	0.39	0.47
	1	0.71	0.86	0.78
	Avg/ Total	**0.67**	**0.69**	**0.67**
Cell 6	0	0.68	0.29	0.41
	1	0.65	0.92	0.76
	Avg/ Total	0.66	0.65	0.61
Cell 7	0	0.67	0.35	0.46
	1	0.64	0.87	0.73
	Avg/ Total	0.65	0.64	0.62
Cell 8	0	0.66	0.37	0.47
	1	0.66	0.87	0.75
	Avg/ Total	0.66	0.66	0.63
Cell 9	0	0.67	0.42	0.51
	1	0.61	0.81	0.70
	Avg/ Total	0.64	0.63	0.61

Table 23*:* Keras classifier Performance metrics for one model for each cell

Based on the key metrics' values in the above table, it can be interpreted that model for cell5 performs better than models for other cells in terms precision, recall and f1-measure values. High precision means algorithm returns low false positive rate which is 0.67 in case of cell5, High Recall means algorithm returned most of the relevant results which is 0.69 in case of cell5 and F1-score takes in account both false positives and false negatives and return 0.67 i.e. low misclassification rate.

II) Building a single model for all cells-

A) Results

Using this method, the model has been built for all cells as shown in the table below. The neural network built using this method has been visualized using nnv package in python is shown below:

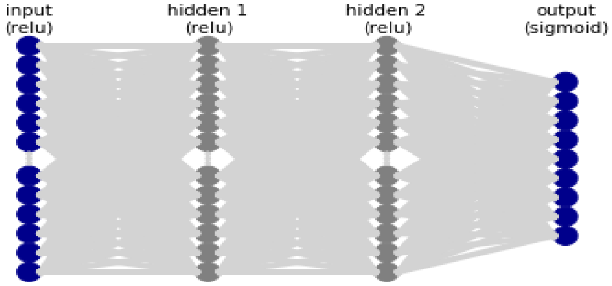
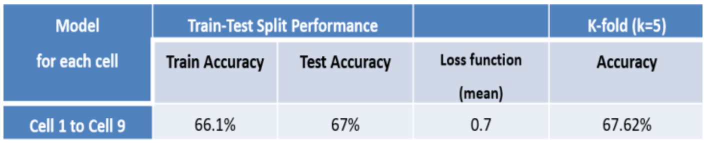


Figure 20*:* Neural Network model for all cells

In the above diagram, when the neural network has been built for all cells together, the network has 19 input layers, two hidden layers and nine output layers. After adequate training of the model, the results achieved are as follows:



Model for each cell	Train-Test Split Performance			K-fold (k=5)
	Train Accuracy	Test Accuracy	Loss function (mean)	Accuracy
Cell 1 to Cell 9	66.1%	67%	0.7	67.62%

Table 24*:* Keras classifier result for all cells

The above table shows that there is not much difference in the train-test split performance approach and k-fold method which means that the model is neither over-fitted nor under-fitted. The training accuracy and test accuracy is 66.1% and 65.1% in case of train-test split approach and 67.62% in case of K-fold approach. As objective is to minimize the loss function to improve the model accuracy, in the figure below it is less for both train and test model i.e. 0.70.

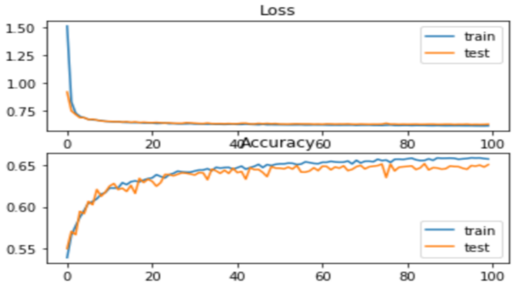
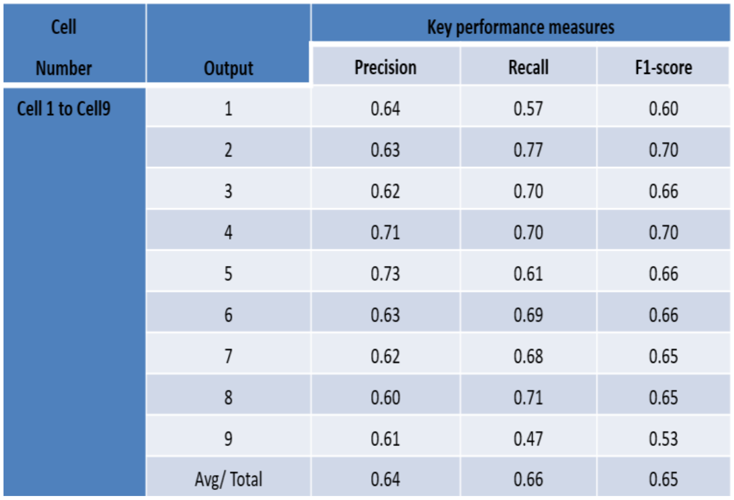


Figure 21: Model result for all cells for Keras classifier

B) Performance Metrics: Confusion Matrix

On a binary classification problem, the commonly used metrics to evaluate the model performance are precision, recall and F1-score other than test accuracy. The Classifier's prediction is represented in terms of 'positive' and 'negative' while actual values are represented as 'true' and 'false'. Precision is the following table means classifier’s ability not to label a negative sample as a positive one, Recall refers to the ability of a classifier to find all the positive samples and F1-score is the weighted harmonic mean of both precision and recall.



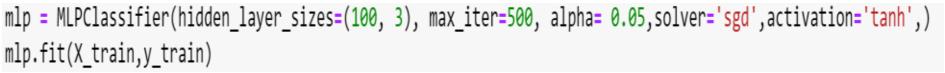
Cell Number	Output	Key performance measures		
		Precision	Recall	F1-score
Cell 1 to Cell9	1	0.64	0.57	0.60
	2	0.63	0.77	0.70
	3	0.62	0.70	0.66
	4	0.71	0.70	0.70
	5	0.73	0.61	0.66
	6	0.63	0.69	0.66
	7	0.62	0.68	0.65
	8	0.60	0.71	0.65
	9	0.61	0.47	0.53
	Avg/ Total	0.64	0.66	0.65

Table 25: Keras classifier Performance metrics for one model for all cells

Based on the key metrics’ values in the above table, it can be interpreted that when a model has been built for all cells the average precision score is 0.64, recall score is 0.66, f-1 score is 0.65.

7.3.2 Multi-layer Perceptron

To build model using MLP, *MLPClassifier* has been imported from sklearn.neural_network package in python and after optimum hyperparameter tuning as explained in above chapter, the model has been built. The dataset has been trained using model as shown in the picture below:



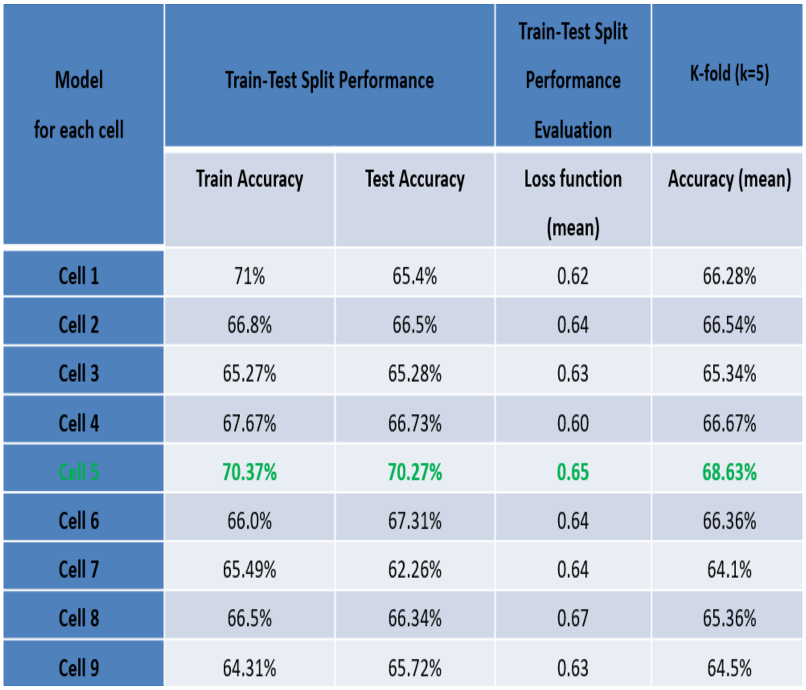
```
mlp = MLPClassifier(hidden_layer_sizes=(100, 3), max_iter=500, alpha= 0.05,solver='sgd',activation='tanh',)
mlp.fit(X_train,y_train)
```

Figure 22: Model for MLP Classifier

I) Building model for each cell- Using this method, machine learning model has been built for each cell and result achieved is shown in the table below:

A) Results

Using MLP classifier, training accuracy lies between 64.31%-70.37% and test accuracy of 62.26%-70.27% has been achieved with a loss function of 0.65. Also, k-fold accuracy lies between 64.1%-68.63%



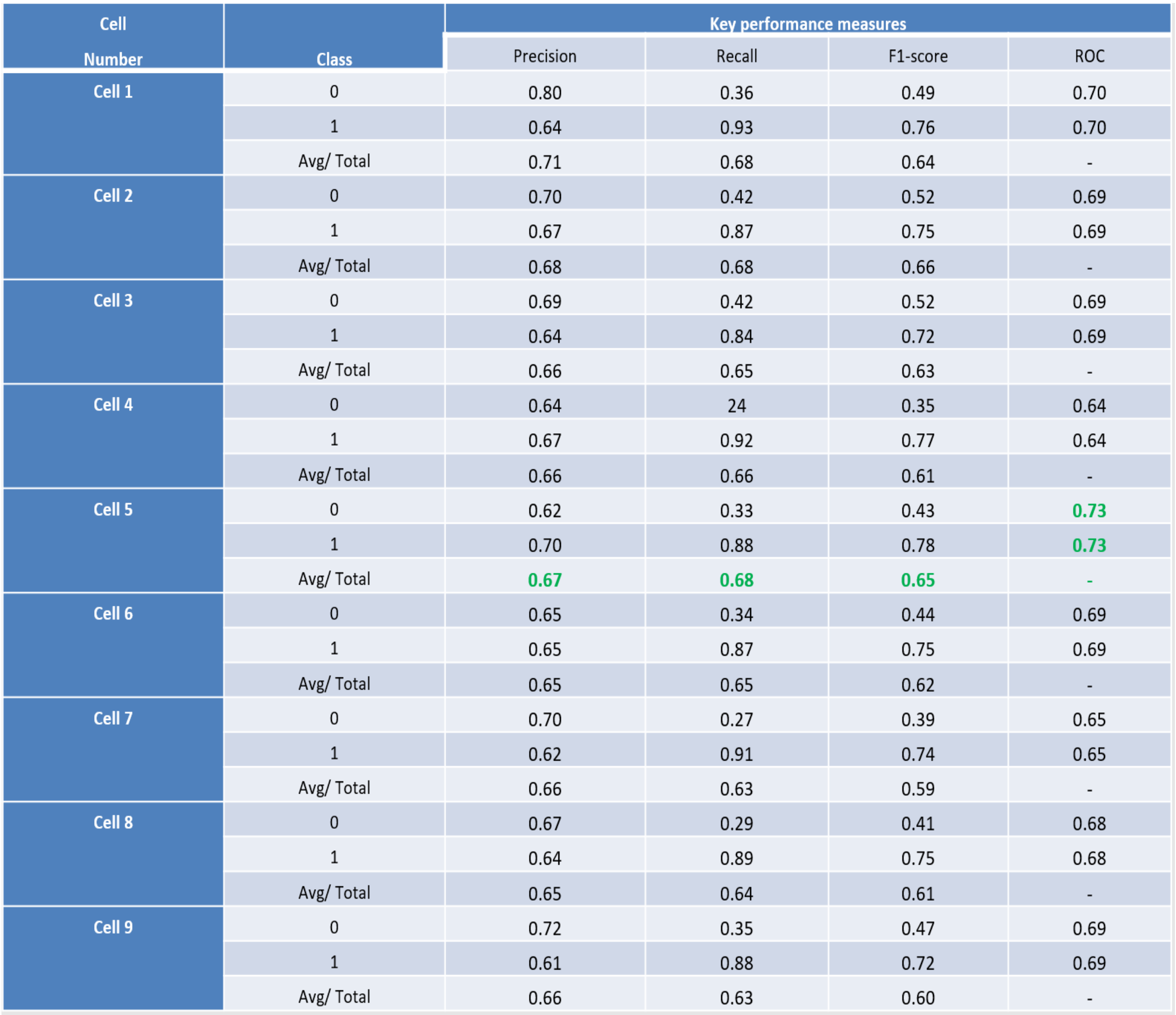
Model for each cell	Train-Test Split Performance		Train-Test Split Performance Evaluation	K-fold (k=5)
	Train Accuracy	Test Accuracy	Loss function (mean)	Accuracy (mean)
Cell 1	71%	65.4%	0.62	66.28%
Cell 2	66.8%	66.5%	0.64	66.54%
Cell 3	65.27%	65.28%	0.63	65.34%
Cell 4	67.67%	66.73%	0.60	66.67%
Cell 5	**70.37%**	**70.27%**	**0.65**	**68.63%**
Cell 6	66.0%	67.31%	0.64	66.36%
Cell 7	65.49%	62.26%	0.64	64.1%
Cell 8	66.5%	66.34%	0.67	65.36%
Cell 9	64.31%	65.72%	0.63	64.5%

Table 26: MLP Classifier model's Result for each cell

The above table shows that there is not much difference in the train-test split performance approach and k-fold method which means that the model is neither over-fitted nor under-fitted. Also, like other classifiers, model for cell 5 performs better than models for other cells.

B) Performance Metrics: Confusion Matrix

For MLP, performance metrics taken into account are Precision, Recall, F1-score and ROC curve. Receiver operating characteristic (ROC) curve is very important metrics for checking classification model's performance. It is a probability curve that measures the capability of model in terms of distinguishing between classes. The F1-score for MLP classifier is 0.65 which means is close to 1 that means model has correctly predicted positive results to determine the true positive rate. The following table displays the precision, Recall, F1-score and ROC curve score for MLP classifier:



Cell Number	Class	Key performance measures			
		Precision	Recall	F1-score	ROC
Cell 1	0	0.80	0.36	0.49	0.70
	1	0.64	0.93	0.76	0.70
	Avg/ Total	0.71	0.68	0.64	-
Cell 2	0	0.70	0.42	0.52	0.69
	1	0.67	0.87	0.75	0.69
	Avg/ Total	0.68	0.68	0.66	-
Cell 3	0	0.69	0.42	0.52	0.69
	1	0.64	0.84	0.72	0.69
	Avg/ Total	0.66	0.65	0.63	-
Cell 4	0	0.64	24	0.35	0.64
	1	0.67	0.92	0.77	0.64
	Avg/ Total	0.66	0.66	0.61	-
Cell 5	0	0.62	0.33	0.43	0.73
	1	0.70	0.88	0.78	0.73
	Avg/ Total	0.67	0.68	0.65	-
Cell 6	0	0.65	0.34	0.44	0.69
	1	0.65	0.87	0.75	0.69
	Avg/ Total	0.65	0.65	0.62	-
Cell 7	0	0.70	0.27	0.39	0.65
	1	0.62	0.91	0.74	0.65
	Avg/ Total	0.66	0.63	0.59	-
Cell 8	0	0.67	0.29	0.41	0.68
	1	0.64	0.89	0.75	0.68
	Avg/ Total	0.65	0.64	0.61	-
Cell 9	0	0.72	0.35	0.47	0.69
	1	0.61	0.88	0.72	0.69
	Avg/ Total	0.66	0.63	0.60	-

Table 27: Performance Metrics for MLP

ROC curve is usually plotted with TPR (True Positive Rate) on y-axis against FPR (False Positive Rate) on x-axis. The ROC curve for cell 5 which has better accuracy than other cells (Appendix4) is displayed in the below picture. An excellent model has ROC near to 1 and cell 5 has the value closet to 1 than other cells i.e. 0.73. Also average precision, recall and F1-score is higher than the value for other cells.

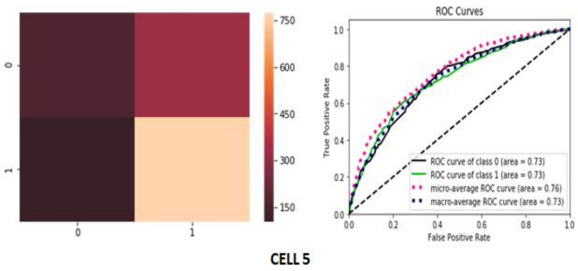
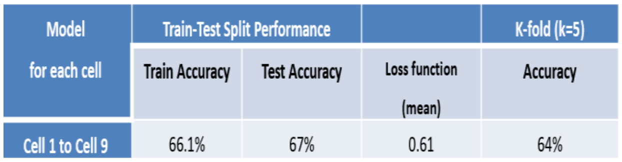


Figure 23: Confusion Matrix and ROC curve for Cell5

II) Building a single model for all cells:

A) **Results**

B) Using this method, training accuracy of 65% and test accuracy of 67% with loss function being 0.61 has been achieved in case of train-test split approach and accuracy of 64% in case of K-fold approach.

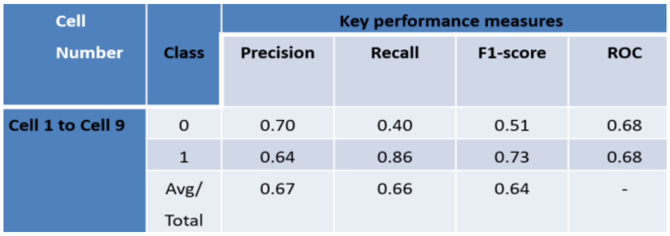


Model for each cell	Train-Test Split Performance			K-fold (k=5)
	Train Accuracy	Test Accuracy	Loss function (mean)	Accuracy
Cell 1 to Cell 9	66.1%	67%	0.61	64%

Table 28: MLP result for all cells.

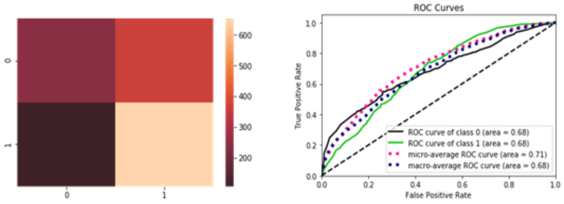
C) **Performance Metrics: Confusion Matrix**

The accuracy achieved by the MLP classifier was later evaluated in terms of precision, recall and F1-score as follows:



Cell Number	Class	Key performance measures			
		Precision	Recall	F1-score	ROC
Cell 1 to Cell 9	0	0.70	0.40	0.51	0.68
	1	0.64	0.86	0.73	0.68
	Avg/ Total	0.67	0.66	0.64	-

Table 29: MLP Performance Metrics for all cells



Confusion Matrix and ROC curve in case of one model for all cells

Figure 24: Confusion Matrix and ROC curve in case of one model for all cells.

7.4 Conclusion for Deep Learning Approaches

After analyzing the results of the neural network (deep learning) approach using Keras classifier and MLP classifier on the basis of two different methods, it is recommended to use Keras classifier as the accuracy results of this model outperforms the results achieved by MLP model. Keras classifier attained higher accuracy than MLP model in case of both methods: 62.9%-69% in case of building model for all cells (method1) and 67% in case of building a single model for all cells whereas it is 64.1%-68.63%% in case of method1 and 64% in case of method2 for random forest model.

Therefore, to achieve high classification accuracy in case of both methods, Keras classifier is a better performer than MLP classifier and can adequately predict farmers' decision in terms of land-use by correctly identifying the positive class which means high true positive rate. Also, it is a better classifier to influence farmers' behaviour for farming or fallowing.

7.5 Final Thoughts

Based on above analysis in terms of machine learning and neural network based approaches in terms of two different methods to build a model, it is recommended to use XGBoost classifier (with accuracy of 73.46%) in case of building model for each cell and Keras classifier (with accuracy of 67.62%) in case of building a single model for all cells. However, XGBoost classifier's (machine learning approach) accuracy in case of method2 is more than keras classifier i.e. 80% but it is not chosen as best approach for method2 because the given dataset contains 9 output columns which are represented by Choice columns. For method2 i.e. to build a single model for all 9 cells, neural network based approach is most appropriate because machine learning algorithms will treat it as a task of classifying the instances into one or more classes (multi-class problem and 9 classes for this project) and will recognize these 9 output columns as 9 different classes which is not true in relation to the nature of dataset in hand and the set objectives. Therefore, neural network based approach is better in case of method2 as such algorithms are extended for multi-output data where one output node is maintained for each class label and weights can be updated in a way so that the algorithm leads to correct label ranking (Sorower, 2010).

8. MODEL DEPLOYMENT: CONCLUSION

The final step using CRISP-DM methodology is to deploy the best model with highest accuracy the model to achieve the objectives of this project/business under consideration. The primary objective of this study was to build a neural network model to predict the behaviour of the players (i.e. typical farmers) in terms of land-use whether to use it for farming or keep it fallow to improve the fertility of the soil as well as to earn incentives in the form of subsidy. However, before concluding on the power of neural networks over machine learning models, the comparison between machine learning and deep learning approaches have been performed due to the nature of data in hand. Let's look at the summary in below section to review the project once again.

8.1 Summary

Once the data was supplied to python's jupyter notebook, it was subject to an adequate exploratory data analysis (section 4.3) which aims to detecting any outliers in the data, checking for missing values, and distribution check of the data to identify traits of the data and locate any potential errors before building the models. For the smooth running of the project, CRISP-DM (section 3.1) methodology was opted. The data used for this project has been derived from a simulated game known as NoncropShare (a coordination game) (Section 3.2).To achieve the project objectives, it was decided to use two different methods such as *Building model for each cell* and *Building a single model for all cells* (section 5.1). Then, four different types of classification models (machine learning and deep learning models) were implemented using a grid search method to come up with the best model. There were a total of 9 cells to categorize into farming and fallowing behaviour of the farmers. The data was splitted into training and test set in a proportion of 70:30. On the basis of model performance it was discovered that for method 1 i.e. *Building model for each cell*, a machine learning algorithm known as XGBoost was performing well (section 7.2.1) and for method 2 i.e. *Building a single model for all cells,* a neural network (deep learning) approach known as Keras classifier was performing well (section 7.3.1).

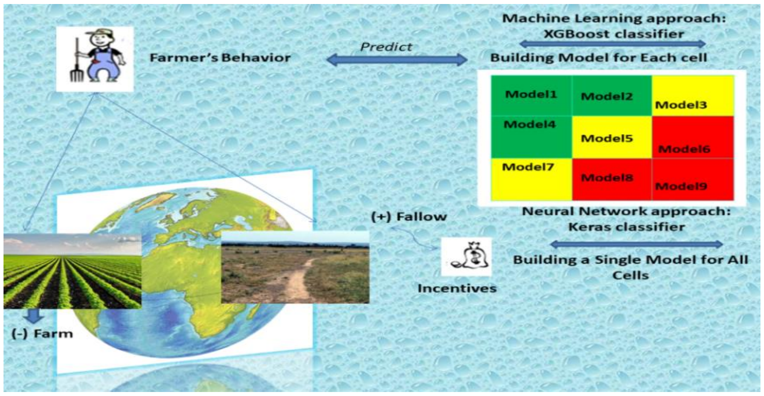


Figure22: Recommendation in terms of model implementation

8.2 Challenges and Limitations

As mentioned in the above section that XGBoost classifier and Keras classifier performed well in terms of building model for each cell and building a single model for all cells respectively where XGBoost classifier was able to achieve accuracy of 80% and Keras Classifier was able to achieve accuracy of 70%. Though these models successfully achieved the objectives of this project but further critical evaluation has been done in terms of dataset to achieve better accuracy in the future. The given data is not a big data and for generating best results using neural networks approach, it is very important that massive amount of data should be fed to the algorithm for training and thereby testing upon. Also, to improve the scalability of neural network, availability of abundance of unlabelled data is vital to the performance of the model both when it is trained and then tested on unseen data. Working with this kind of data was a key challenge which required adequate hyperparameter tuning and comparing the model results not only on the basis of accuracy but on the basis of suitability of the approach for each method used. This is because though XGBoost classifier performed well in case of method2 as well but Keras classifier is a better predictor based on the working principle of method2 which requires treating *9 choice columns* as *9 outputs* not *9 varied classes*.

8.3 Evaluation

The unexpected revelation of this study is that on the basis of the comparison made between machine learning and deep learning algorithms, though the desired approach was to use neural network for the dataset in hand, but the extreme Gradient Boosting classifier (Appendix 1) among machine learning algorithms (XGBoost) with 80% accuracy emerged as the best model on all the datasets (training, test and validation) and it passed all tests performed on it using adequate hyperparameter tuning. The results obtained from XGBoost classifier shows that Deep Learning is currently over-estimated and it is not suited to solve all classification problems.

8.4 Recommendations

Based on the results achieved through this study, it is confirmed and concluded that the machine learning method *(XGBoost model for the project under consideration)* is accurate, robust and reliable than the neural networks when the task is to build a model for each cell (method1) while Keras Classifier (though accuracy is less than XGBoost classifier) is a better predictor of farmer's decision making when the task is to build a single model for each cell. Therefore, I would recommend using neural network based approach *(Keras classifier for the project under consideration)* which is more reliable, accurate and suitable in case of multi-output classification to build models on the given dataset or datasets of similar kind. Also, building a single model makes the process fast, easy to analyze and more logical results to predict the farmer's behaviour.

8.5 Future Work

As farmers are the key stakeholders in agricultural and land-use issues. To understand their behaviour is an essential priority to design appropriate policy for agricultural practices keeping in view the sustainable development and to maintain the ecological biodiversity. With the technological advancements, a system which can predict farmer's behaviour provides an advantage in terms of adjusting the agricultural policies for optimum land-use.

Therefore, through the use of experimental approach along with the rigorous model testing adopted to conduct this study, I foresee that in order to avoid difficulty in either replicating this project on different dataset or tailoring it to a different machine learning (classification) or deep learning project in any field of study, the amount of data provided should be more than the present one. Also, to match well with the objectives of the study, the data provided should be unstructured data. Furthermore, it is discovered that few more features are required to build more reliable models. It is a good idea to include more numerical features along with categorical ones. *Examples of features can be the type of produce in each cell like wheat, corn et al., how much is the total produce in each cell, age of the farmer, amount of pesticides and fertilizers used on the land.*

Due to the time constraint involved in conducting this project, the above mentioned piece of work has not been taken into consideration but the set objectives of this project have successfully been achieved.

References

Apeksha Jain., Radovan Kavicky. and Ankit Dixit. (2017). *Ensemble Machine Learning*. Packt Publishing.

Azevedo, A.I.R.L. and Santos, M.F., 2008. KDD, SEMMA and CRISP-DM: a parallel overview. *IADS-DM*. https://pdfs.semanticscholar.org/7dfe/3bc6035da527deaa72007a27cef94047a7f9.pdf

Baynham-Herd, Z., Redpath, S., Bunnefeld, N., Molony, T. and Keane, A., 2018. Conservation conflicts: Behavioural threats, frames, and intervention recommendations. *Biological conservation*, *222*, pp.180-188.
https://www.sciencedirect.com/science/article/pii/S0006320718301022

Bell, Andrew; Zhang, Wei; Bianchi, Felix; and vander Werf, Wopke. 2013. NonCropShare- a coordination game for provision of insect-based ecosystem services. IFPRI Biosight Program. Version 2 (September 29, 2014). Washington, D.C.: International Food Policy Research Institute (IFPRI). https://www.ifpri.org/publication/noncropshare-coordination-game

Bell, A., Zhang, W., Nou, K., 2016. Pesticide use and cooperative management of natural enemy habitat in a framed field experiment. Agric. Syst. 143, 1–13.
https://www.sciencedirect.com/science/article/pii/S0308521X15300524?via%3Dihub

Bell, A. and Zhang, W., 2016. Payments discourage coordination in ecosystem services provision: evidence from behavioral experiments in Southeast Asia. *Environmental Research Letters*, *11*(11), p.114024.
https://iopscience.iop.org/article/10.1088/1748-9326/11/11/114024/meta

Bosnjak, Z., Grljevic, O. and Bosnjak, S., 2009, May. CRISP-DM as a framework for discovering knowledge in small and medium sized enterprises' data. In *2009 5th International Symposium on Applied Computational Intelligence and Informatics* (pp. 509-514). IEEE.
https://ieeexplore.ieee.org/abstract/document/5136302

Brownlee, J. (2019). Keras Tutorial: Develop Your First Neural Network in Python Step-By-Step. [online] Machine Learning Mastery.
https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/

Brownlee, J. (2019). How to Calculate Precision, Recall, F1, and More for Deep Learning Models. [Blog] Machine Learning Mastery.
https://machinelearningmastery.com/blog/

Cannon, J. (2019). *'Unprecedented' loss of biodiversity threatens humanity, report finds*. [online] Mongabay Environmental News
https://news.mongabay.com/2019/05/unprecedented-loss-of-biodiversity-threatens-humanity-report-finds/

Ceballos, G., Ehrlich, P.R. and Dirzo, R., 2017. Biological annihilation via the ongoing sixth mass extinction signaled by vertebrate population losses and declines. *Proceedings of the National Academy of Sciences*, *114*(30), pp.E6089-E6096.
https://www.pnas.org/content/pnas/114/30/E6089.full.pdf

Chollet, F. (2018). *Deep learning with Python*. Shelter Island, N.Y: Manning.

DataCamp Community. (2019). *Evaluation Metrics : Precision,Recall and F1 score*. https://www.datacamp.com/community/news/evaluation-metrics-precisionrecall-and-f1-score-mogk4k1su0d

Dirzo, R., Young, H.S., Galetti, M., Ceballos, G., Isaac, N.J. and Collen, B., 2014. Defaunation in the Anthropocene. science, 345(6195), pp.401-406.
https://science.sciencemag.org/content/sci/345/6195/401.full.pdf

De Wrachien D. (2003) Land Use Planning: A Key to Sustainable Agriculture. In: García-Torres L., Benites J., Martínez-Vilela A., Holgado-Cabrera A. (eds) Conservation Agriculture. Springer, Dordrecht
https://link.springer.com/chapter/10.1007/978-94-017-1143-2_57

Food and Agriculture Organization of the United Nations (FAO) (2011). *THE STATE OF THE WORLD'S H LAND AND WATER RESOURCES FOR FOOD AND AGRICULTURE*. NewYork: The Food and Agriculture Organization of the United Nations and Earthscan, p.308.
http://www.fao.org/3/a-i1688e.pdf

Gevrey, M., Dimopoulos, I. and Lek, S., 2003. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological modelling*, *160*(3), pp.249-264.
https://www.sciencedirect.com/science/article/pii/S0304380002002570

Islam, K., Jashimuddin, M., Nath, B. and Nath, T.K., 2018. Land use classification and change detection by using multi-temporal remotely sensed imagery: The case of Chunati wildlife sanctuary, Bangladesh. *The Egyptian Journal of Remote Sensing and Space Science*, *21*(1), pp.37-47.
https://www.sciencedirect.com/science/article/pii/S1110982316301594

Karimi, A., Brown, G. and Hockings, M., 2015. Methods and participatory approaches for identifying social-ecological hotspots. *Applied Geography*, *63*, pp.9-20.
https://www.sciencedirect.com/science/article/pii/S0143622815001381

Lek, S., Belaud, A., Dimopoulos, I., Lauga, J. and Moreau, J., 1995. Improved estimation, using neural networks, of the food consumption of fish populations. *Marine and Freshwater Research*, *46*(8), pp.1229-1236.
http://www.publish.csiro.au/mf/MF9951229

Laë, R., Lek, S. and Moreau, J., 1999. Predicting fish yield of African lakes using neural networks. *Ecological modelling*, *120*(2-3), pp.325-335.
https://www.sciencedirect.com/science/article/pii/S030438009900112X

McKinley, P., Cheng, B.H., Ofria, C., Knoester, D., Beckmann, B. and Goldsby, H., 2008. Harnessing digital evolution. *Computer*, *41*(1), pp.54-63.
https://ieeexplore.ieee.org/abstract/document/4445603/

Özesmi, S.L. and Özesmi, U., 1999. An artificial neural network approach to spatial habitat modelling with interspecific interaction. *Ecological modelling*, *116*(1), pp.15-31.

https://www.sciencedirect.com/science/article/pii/S0304380098001495

Perrings, C. and Halkos, G., 2015. Agriculture and the threat to biodiversity in sub-saharan Africa. *Environmental Research Letters*, *10*(9), p.095015.
https://iopscience.iop.org/article/10.1088/1748-9326/10/9/095015/meta

Redpath, S.M., Keane, A., Andrén, H., Baynham-Herd, Z., Bunnefeld, N., Duthie, A.B., Frank, J., Garcia, C.A., Månsson, J., Nilsson, L. and Pollard, C.R., 2018. Games as tools to address conservation conflicts. *Trends in ecology & evolution*, *33*(6), pp.415-426.
https://www.sciencedirect.com/science/article/pii/S0169534718300594

Scikit-learn.org. (2019). 1. Supervised learning — scikit-learn 0.20.3 documentation.
https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

Sorower, M.S., 2010. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, *18*, pp.1-25.
https://pdfs.semanticscholar.org/6b56/91db1e3a79af5e3c136d2dd322016a687a0b.pdf

Tilman, D., May, R.M., Lehman, C.L. and Nowak, M.A., 1994. Habitat destruction and the extinction debt. *Nature*, *371*(6492), p.65.
https://www.nature.com/articles/371065a0

Weston, J. and Watkins, C., 1999, April. Support vector machines for multi-class pattern recognition. In Esann (Vol. 99, pp. 219-224).
http://tka4.org/materials/lib/Articles-Books/Speech%20Recognition/from%20Nickolas/SVM%20for%20multiclass%20pattern%20recognition.pdf

Willcock, S., Martínez-López, J., Hooftman, D. A., Bagstad, K. J., Balbi, S., Marzo, A., ... & Villa, F. (2018). Machine learning for ecosystem services. *Ecosystem services*, *33*, 165-174.
https://www.sciencedirect.com/science/article/pii/S2212041617306423

Wirth, R. and Hipp, J., 2000, April. CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (pp. 29-39). Citeseer.
https://pdfs.semanticscholar.org/48b9/293cfd4297f855867ca278f7069abc6a9c24.pdf

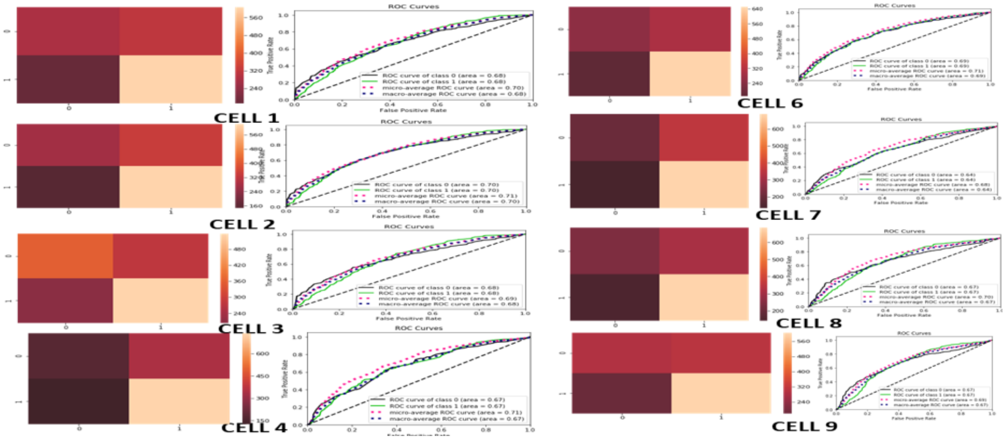
Yu, L., Wang, S. and Lai, K.K., 2005. An integrated data preparation scheme for neural network data analysis. *IEEE Transactions on Knowledge and Data Engineering*, *18*(2), pp.217-230.
https://ieeexplore.ieee.org/abstract/document/1563984

Appendices

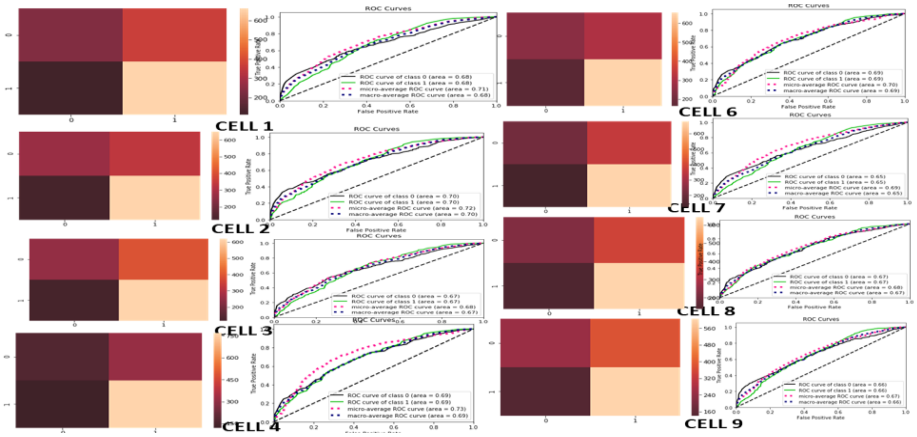
Appendix 1: The whole code can be found using following github link-

https://github.com/pouriamo66/MSc-Big-data-Dissertation-

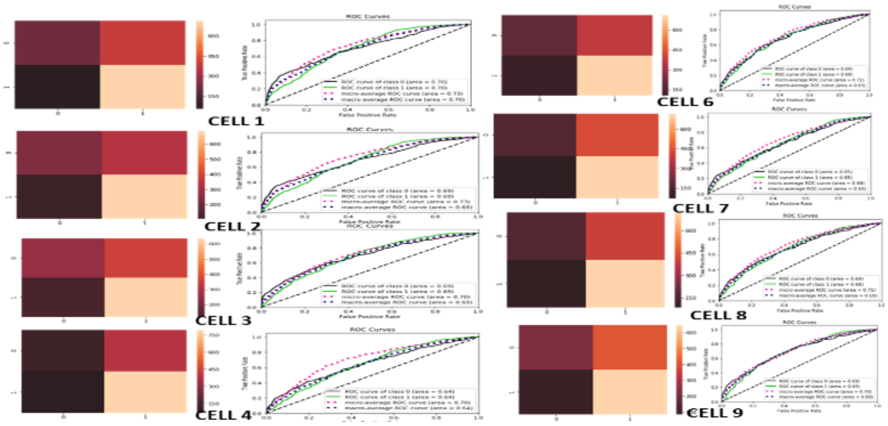
Appendix 2: Confusion Matrix and ROC curve for Random Forest Cell1 to cell 9 except for cell5.



Appendix 3: Confusion Matrix and ROC curve for XGBoost classifier Cell1 to cell 9 except for cell5.



Appendix4: Confusion Matrix and ROC curve for MLP classifier Cell1 to cell 9 except for cell5.


ROC Curves
CELL 1
CELL 2
CELL 3
CELL 4
CELL 6
CELL 7
CELL 8
CELL 9