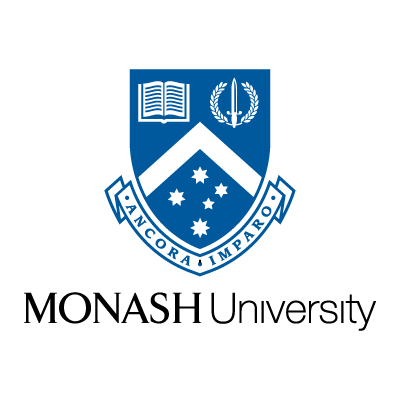
****

**Monash University**

**DELWP Project Final Report**

**Group 13**

**Members:**

**Haoying Zhang 27791912**

**Haoyang Zheng 28764838**

**Xiaoran Gui 27178234**

**Word count**

**Total word(excluding Reference): 8024**

DELWP Project Final Report

Haoyang Zheng

Haoying Zhang

Xiaoran Gui

October 25, 2019

Table of Contents

1. Introduction 4
2. Background 5
   1. Literature review ..…………………………………………………. 5
   2. Summary …………………………………………………………… 8
3. Methodology 8
   1. Introduction ………………………………………...…………….. 10
   2. Data cleaning ………………………………………...…………….. 12
   3. Raster data and shapefile……………..……………..…………….. 13
   4. Separate into different species……………..……………..………… 17
   5. Pseudo absence data generation……………………………….……… 17
   6. Random forest ………………………………………...…………….. 19
   7. Imbalance solution………………………………………...………… 21
   8. Proper evaluation metric………………………………………...…… 21
   9. Oversampling and undersampling……………………………...…… 22
   10. Application implementation……………………………...…………… 24
4. Project management 24
   1. Background ………………………………………………………….. 24
   2. Project management approach ……………………………………… 25
   3. Risk management …………………………………………………… 27
   4. Resource execution and planning …………………………………… 27
   5. Management limitations ……………………………………………… 28
5. Outcomes 30
   1. Hypothesis ………………..………………………………. 30
   2. XGBoost and SVM ………………………………………. 35
   3. Limitations of software ……………………………………… 36
6. Conclusion 36
7. Reference 38

**1 Introduction**

When the Victoria government making decisions about landscape management, fire prevention, and human activities in non-urban areas, animal and plant distributions are some of the most important factors to be considered in. To enrich the distribution dataset, species observation points data from volunteers is also collected, however, it is not appropriate to add all records without verifying them with different reliability, since different records are recorded by different people with different record type. For example, an observation record with a captured record type or picture record type is more reliable than an observation record with a heard record type; a species with most observation records show they tend to live under 100 meters altitude, a new observation record with a 3000 meters altitude location may not very reliable. Hence, before new observation data being stored into the observation database, which is Victorian Biodiversity Atlas, all the observations need to be verified and categorized. Usually, this kind of work is done by the expert verification process.

As there are more and more observation records sending to Victorian Biodiversity Atlas, the workload of verifying and categorizing new observation records is already beyond the capability of the expert verification process. So, a predictive model is needed to verify and categorize new observation records into high and low-reliability groups. Records with high reliability will be more likely to be recorded by Victorian Biodiversity Atlas, while observation records with low reliability are not.

To solve this problem, a new predictive model has been developed, we achieved the categorization of observation records by developing a predictive model. Random forest algorithm, SVM algorithm, and xgboost are used to develop the predictive model, with the observation data and raster data as training and test data, which are provided by DELWP. Along with the predictive model, the corresponding application and UI are also developed. Users can get a binary prediction result(high reliability, low reliability ) of new observation data as input.

In this report, each step of this project will be explained and discussed. Models we tried and the final models we used during the project will be analyzed and evaluated in detail. The project management method and team management method will also be stated. The detail of the application we developed will be discussed, so as the limitations. The achievement and limitation of this project will also be explained in detail.

**2 Background**

**2.1**

**Literature Review updated**

**Introduction**

Species distribution model(SDM) is a common and useful tool for conservation planning, assessing environmental change impacts biodiversity, epidemiology, landscape management, fire management and issuing permits in a certain area (Jetz, McPherson & Guralnick 2012, DELWP 2019). Through the exploration of various papers and literature reviews by other professionals, some key points were noted in building a species distribution model that was suitable for our project.

Also, there exist many different modeling techniques that can be used to predict species distribution such as machine learning algorithms or regression techniques. Choosing the right one is heavily based on the type of data that we have and the result that we are aiming for(Classification/Prediction in this case).

Our research has also found models that are built on presence and absence data are more reliable and provides higher accuracy predictions than those that only have presence data. However,in the real world, compared to the presence data, it is confirmed that true absence data are hard to obtain, especially for mobile species, the level of sampling for true absence data is higher than for true presence data[9]. Thus pseudo-absence is a technique that could be used to generate “true” absence data. This paper reviews the pseudo-absence generation technique and modeling algorithm that is suitable for this project.

**Pseudo-absence data**

Previously, models mostly only used presence only data however from research conducted by Elith et al. 2006, it has been found that model that uses presence only data often underperforms compared to models that use presence**/**absence data[14]. But true absence data is hard to obtain.

Thus, some suggestions are to use pseudo-absence in place of true absence data. Some notable pseudo-absence generation methods for machine learning models such as Random Forest were tested and according to Barbet-Massin, Jiguet, Albert & Thuiller who had done an extensive analysis on the topic, they found that geographical exclusion (‘2°far’ “random selection of any available point located at least two degrees away from any presence point”) yields a better model accuracy with few presence while on the other hand, as the number of presences increases, climatic exclusion (‘SRE’ “random selection of points from all points outside of the suitable area estimated by a rectilinear surface envelope from the presence sample”) is a better option and generally surpasses the performance of ‘2°far’ if the presence is high.

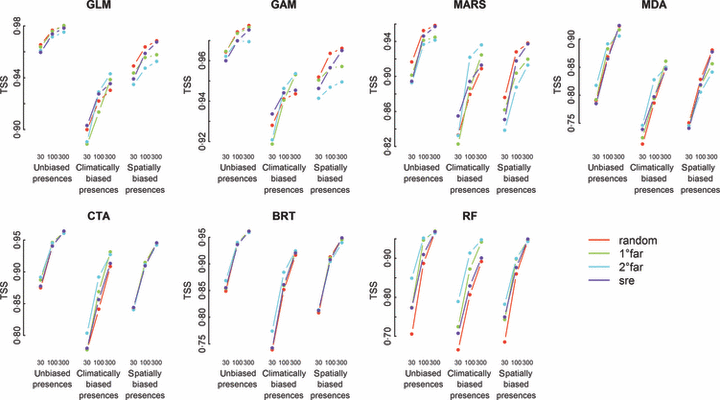


Figure 1: TSS vs Absence data generated for Spatially/climatically/un Biased presence data

In Figure 1 it shows the evaluation results(using TSS score) according to the modeling technique, the number of presence, to the quality of presence and the method used to select pseudo-absence(mean over the different number of pseudo-absences, the weighting scheme, and the random selection of presence)

Another important aspect of generating pseudo absence data is that the ratio of pseudo absence data also affects the accuracy of the model. By the experiments that Barbet-Massin, Jiguet, Albert & Thuiller have done, a total of 1612800 models have been tested and calculated the mean predicted distributions resulting from several (2–20) replicates of pseudo‐absences selection methods for comparison[9]. It has been found that for Random forest or in general machine learning techniques will benefit from a 1:1 ratio of the number of presence and the number of pseudo absence points.

For summary on the pseudo absence generation techniques, random generation of the pseudo absence point will provide a better result for machine learning techniques when the presence data are not climatically biased otherwise the 2degree far method will overtake the random method in performances and the most suitable number of pseudo absence points to generate for machine learning technique in general are a 1:1 ratio.

**Random Forest Modeling Algorithm**

Random Forest is an algorithm that is based on bagging approaches. Random forest generates a set of weak-learners based on a bootstrap of the data but yet it still converges on an optimal solution[7]. That is due to the nature of how it works, random forest randomly draws a selection of the attribute at each of the tree nodes, retaining the attribute that provides the most informational content[11]. The random forest then takes the whole ensemble of decision trees that it has generated and takes a vote between them to be the output as the final prediction. Random Forest is also robust to noise even given a very large number of independent attributes as demonstrated[12].

With rare species, if environmental data are present with the species observations random forest is able to outperform other species distribution models such as maxent, CART and TreeNet.

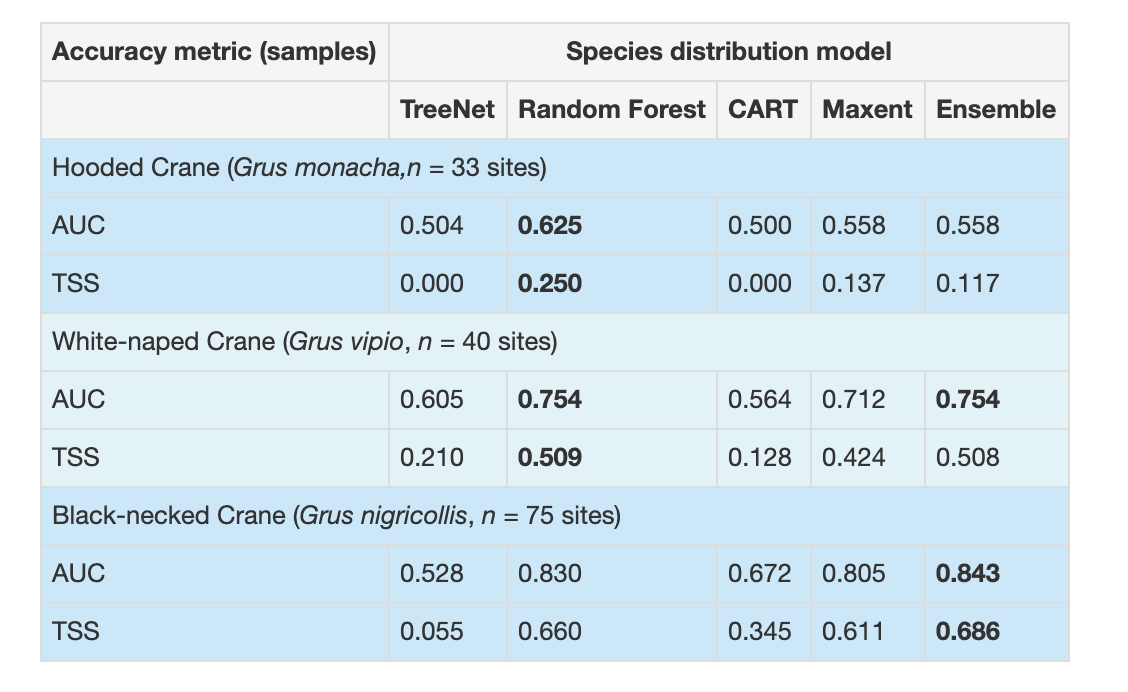


Figure 2: SDM Accuracy Score For Each Species

In figure 2 it shows the AUC score and the TSS of each of the species distribution model tested in the research[13], and it has shown that Random Forest is doing as well or better than other models in terms of accuracy from AUC score for modeling three very rare species with only 33 observation, 40 observations and 75 observations.

In summary, the random forest model with be able to retain the most important attribute from all the decision trees modeled and it is robust to noise and also able to predict rare species which had been handy in our project.

**2.2 summary**

In summary, these two papers were the most important literature review that we considered to be useful for us while we were developing the models, we had planned to adopt these techniques into our model development cycle.

The main changes from last semester was that we have now removed the threshold method due to the fact that we are introducing pseudo absence data into the dataset and previously when we planned to use the threshold method we were planning to not use pseudo absence data but with much consideration we believe we will benefit off generating pseudo absence data, with the dataset given to us by DELWP only containing presence data and the number of data points for some species was quite high but all of them were High reliable which will prevent the data from correctly predicting Low reliable.

Outlier removal was also scrapped since we have done our research on the different types of species that we are predicting and we have found that all the High reliable data points actually match the places that the species are known to exist in. We have also made sure that the data point matched the data point that we were able to get from government websites such as “Atlas of Living Australia” and “Birdlife”. Therefore, since we know that none of these is actual outlier we have no reason to just remove it from the dataset.

**3 Methodology**

**3.1 Introduction**

There are many different ways in building species distribution model, everyone has their own way of developing the models but there are a few important things that almost everyone needs to consider.

This is why in building our species distribution models, we have taken a lot of consideration of the following problems and identified to our best ability the solution to each and every one of them. The following are what the team had considered and have implemented into our models.

* VBA data variable selection
* Raster data/Shape file data
* Separating into different species
* Pseudo absence data
* Random forest model
* Imbalance of data
* Proper evaluation metrics

Consideration of the VBA data

The record data is the very start of the model development process. There are 6 species in the dataset and a total of 18890 rows of observations for the species and a total of 27 columns

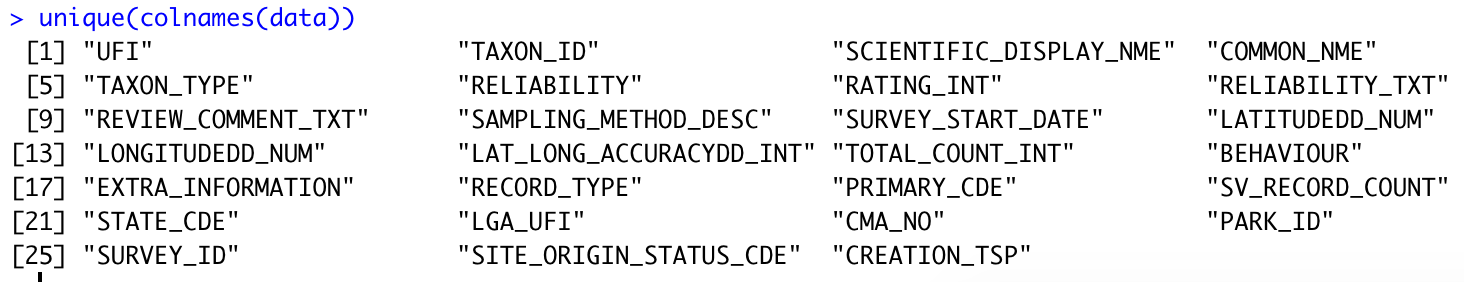


Figure 3: Unique Column Names for Data

With close examination, most of these attribute columns can not be used in our training data. The first consideration was to only keep the LONGITUDEDD\_NUM, LATITUDEDD\_NUM, RELIABILITY\_TXT, COMMON\_NME and SAMPLING\_METHOD\_DESC. Since other columns contain a lot of empty values. We then quickly realized that the SAMPLING\_METHOD\_DESC contains more than 60 levels and from past experience, we knew that the random forest that we wanted to build would not a lot more than 53 levels in one column which means we had to scrap the SAMPLING\_METHOD\_DESC.

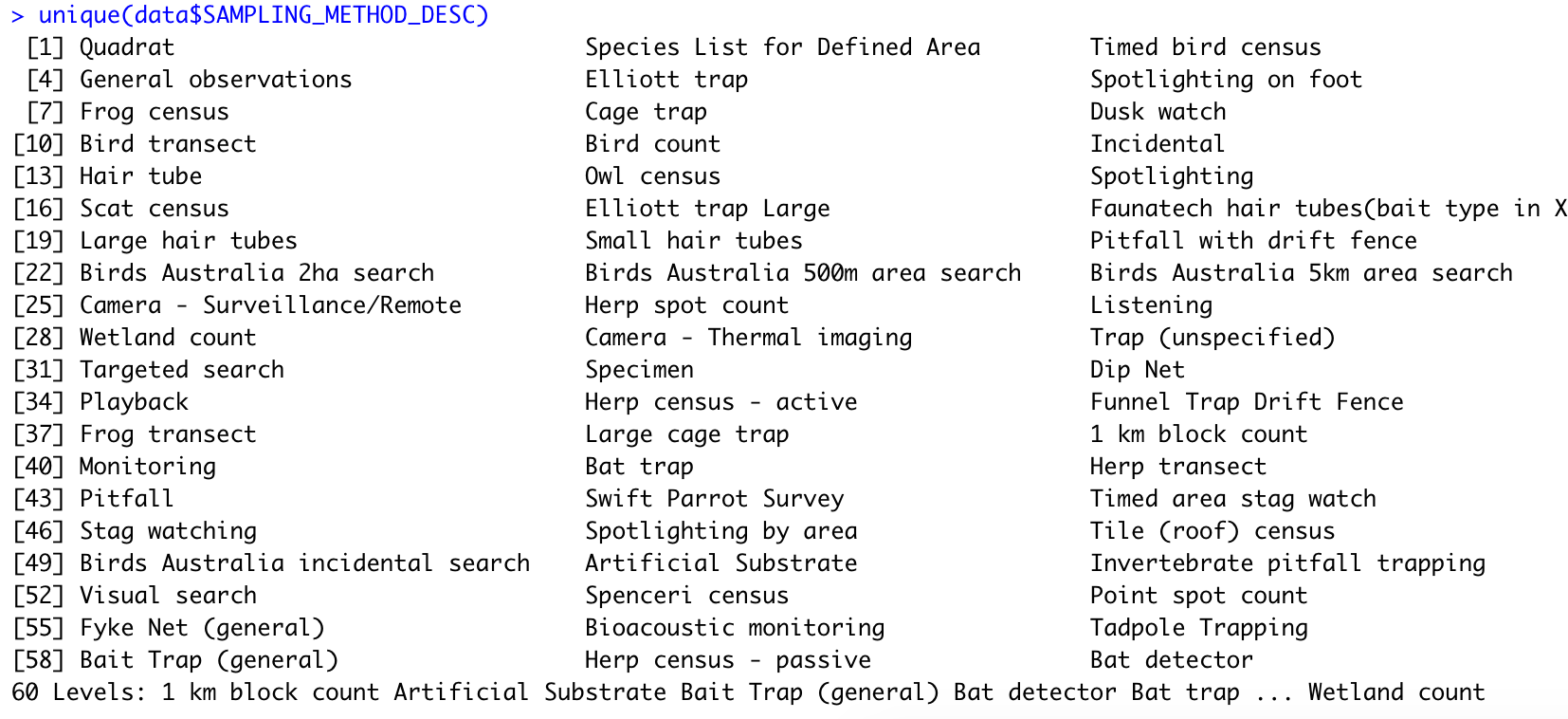


Figure 4: Unique Sampling\_Method\_Desc for Data

The LONGITUDE, LATITUDE, RELIABILITY\_TXT and COMMON\_NME columns were a must for us, as we knew we wanted to separate the species and build a model for each of them on their own. We had decided to keep these four rows only as we would extract raster data and shape file data later on which is what we have found to be

**3.2 Data cleaning**

Before doing anything with the data frame, we have to clean the dataset to make sure that everything is perfect for later use, in the stance of data science, a clean dataset is a key to success. Data cleaning is the main process in data science.

Therefore, the first thing we did was removed duplicate observations.

Before:

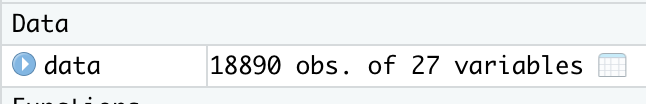


Figure 5: Screenshot of Data frame detail

> data = distinct(data)

Which left us with a total of 17408 rows with 27 variables

After:

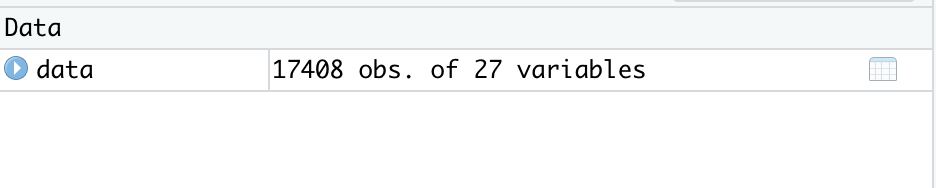


Figure 6: Screenshot of Data frame detail

A total of 1482 rows were removed for being duped.

The following part was also built into a function so that it can be integrated into the user interface.

The NA values in the dataset were changed according to their column type, if the column type were character then the NA values were changed to “None” as a character which fits the column type and similarly for the columns that are of type numeric, the NA values were turned into -1 as a numeric number to keep the column type consistent. Data consistency is another important factor when training model, as we do not want to use incorrect data in our models as it could potentially introduce noise into the model.

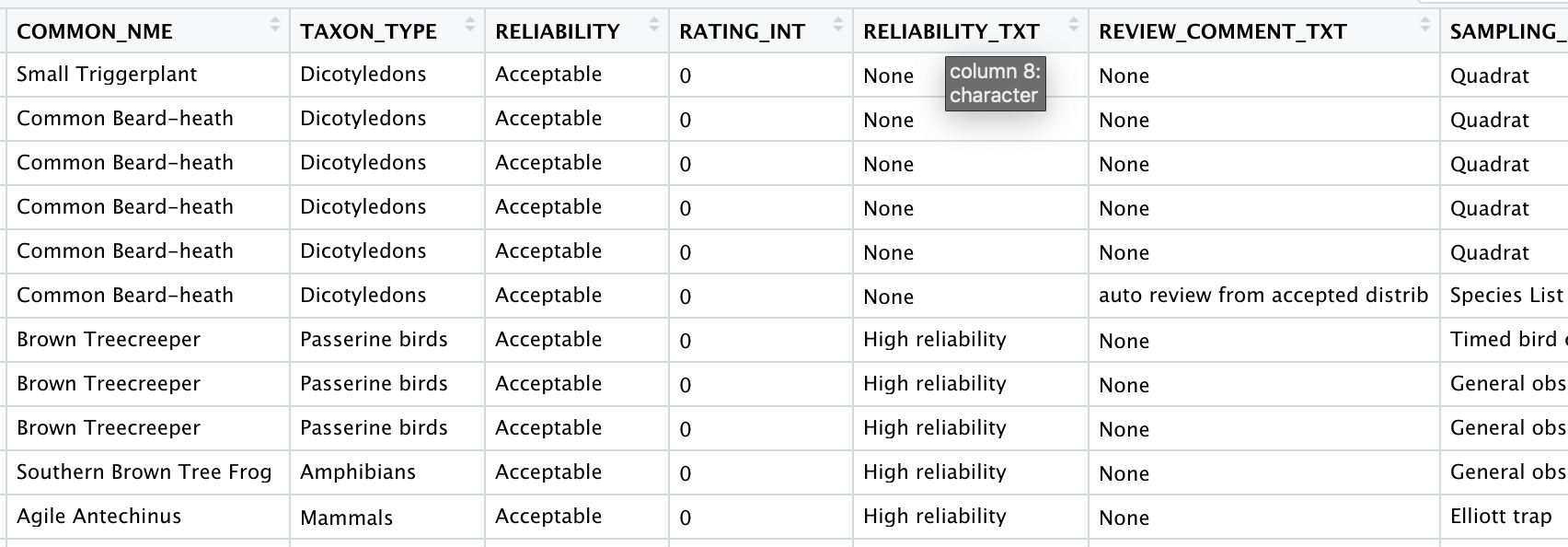


Figure 7: Screenshot of the column detail of a data frame

**3.3 Raster data and shape file**

Raster data were added into the dataset through the use of raster and rgdal libraries, these provided us with function such as readOGR and raster which were used to load the shape files and raster files respectively.

Raster/environmental data are very important for species distribution model especially for data with only presence data such as the data provided to us. Raster/environmental data provides extra information of the distribution of presence data which would allow us to have a better chance at producing a set of more correct “absence data” and thus increasing the reliability of our model’s predictions.

Due to the nature of the projection system on the raster files, we have to convert the dataset into a spatial point data frame and change the projection system of the data frame to match the ones used in the raster file and shape file.

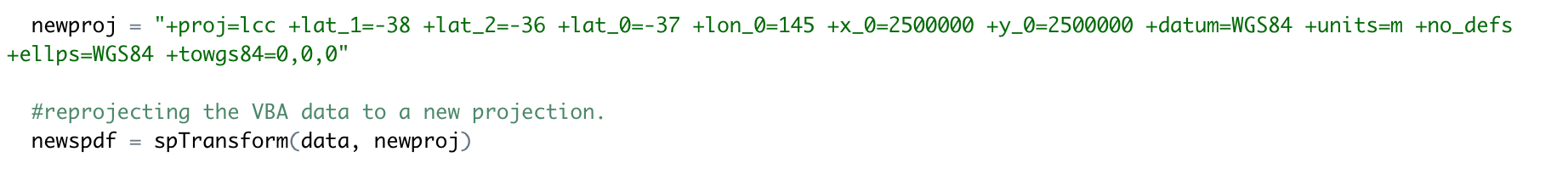


Figure 8: New projection function code

We turned the extraction of raster data and shape data into their own functions which then can be used and implemented into the user interface.

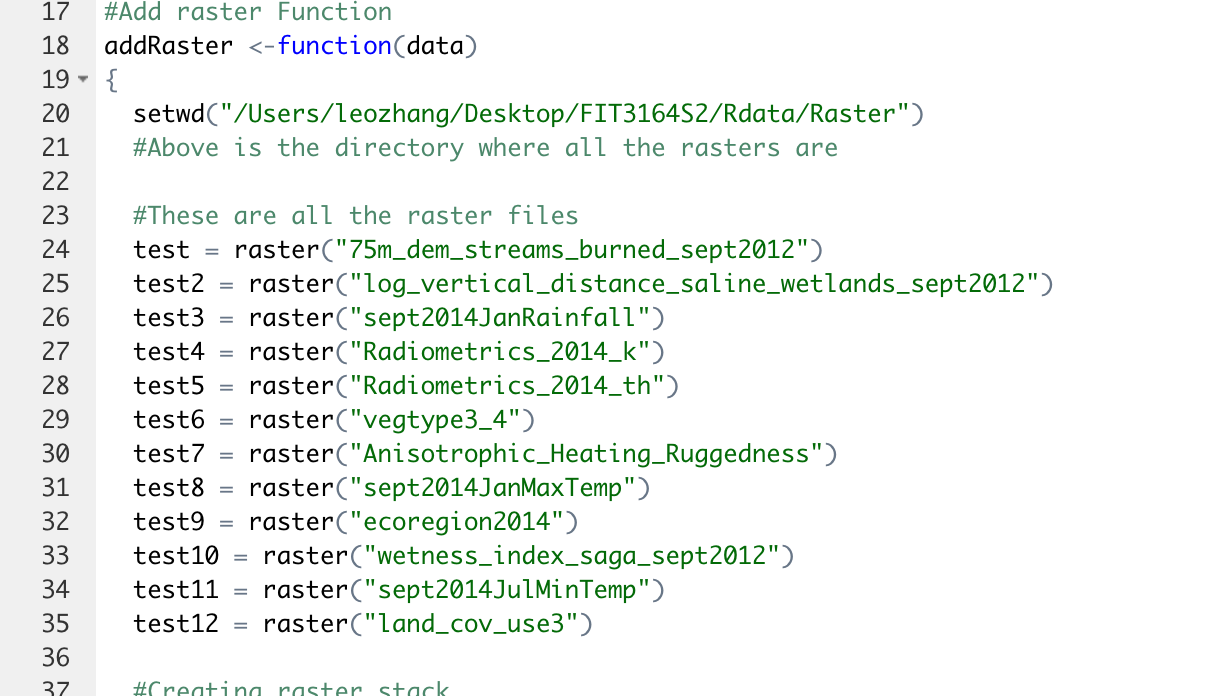


Figure 9: Screenshot of part of addRaster Function

(Part of the code)

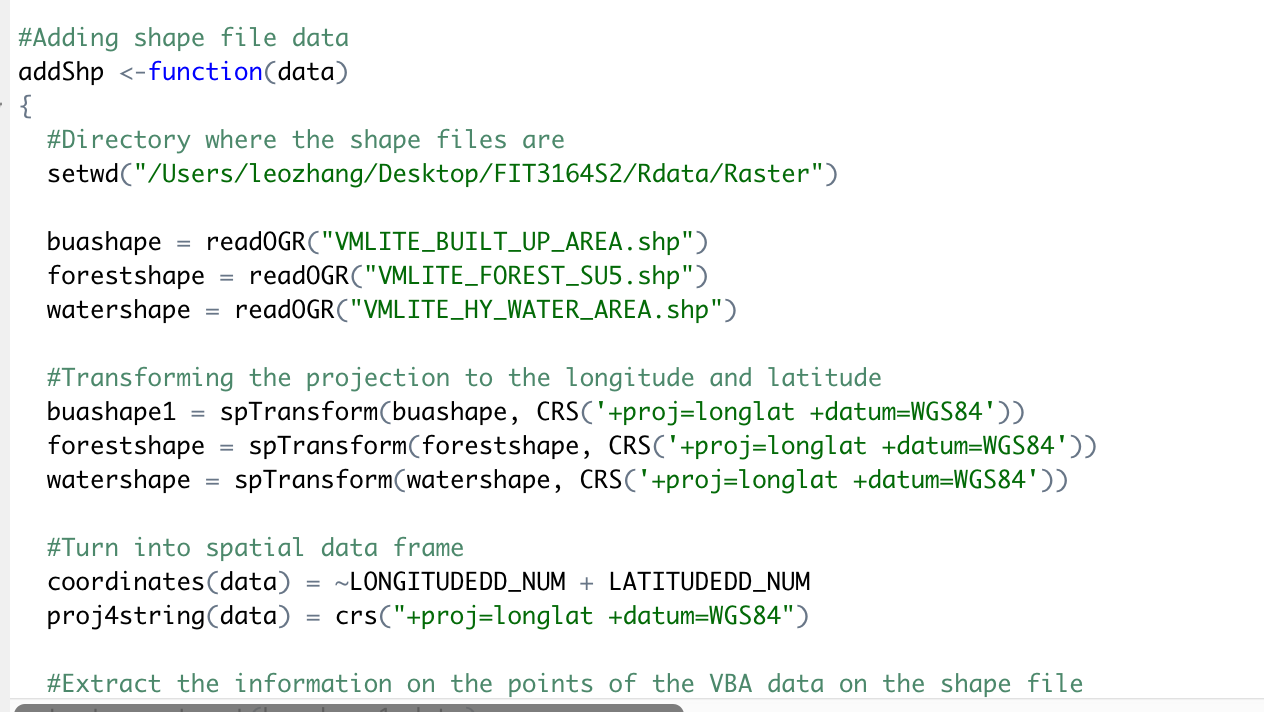


Figure 10: Screenshot of part of addShp function

Once the functions were done, we extracted the information from which the data point overlapped with raster files and then do the same thing with the Shape files using these functions.

There were some limitations in the raster and shape file function, they only include a defined set of raster and shape files because some of the raster files have different extent which means that it is impossible to make a raster stack out of raster files with different extent without doing a costly calculation to transform the files into the same extent, we could have potentially missed out on a key raster file which could have improved our model accuracy.

On the other hand, a similar but rather different problem was faced with the shape files. Some of the files were duplicates:



Figure 11: Screenshot of duplicate files

Multiple files were like this, this is just one of the examples shown. Other shape files were lines or dots which provided no real importance to our dataset in the end, we found three files that could provide useful information and those are the Forest, BuildUpArea, and WaterlandType.

As we were exploring the species to find more information on predicting them, we found a website that shows the native region establishment of one of the species we had to predict: “Common Beard Heath”.

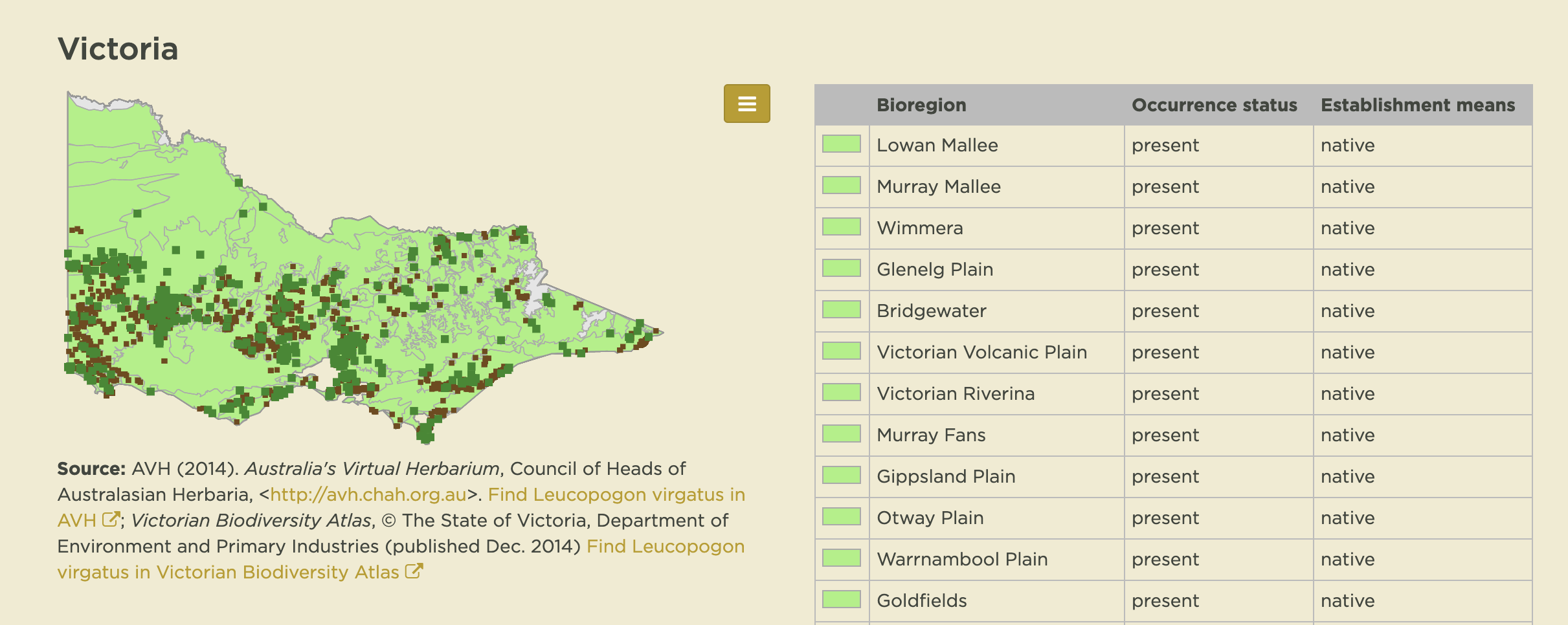
The species distribution for “Common Beard Heath” as shown on “[http://vicflora.rbg.vic.gov.au”](about:blank)

Figure 12: Screenshot of “Common Beard Heath” species distribution in victoria(vicflora 2019)

The bioregions of Victoria:

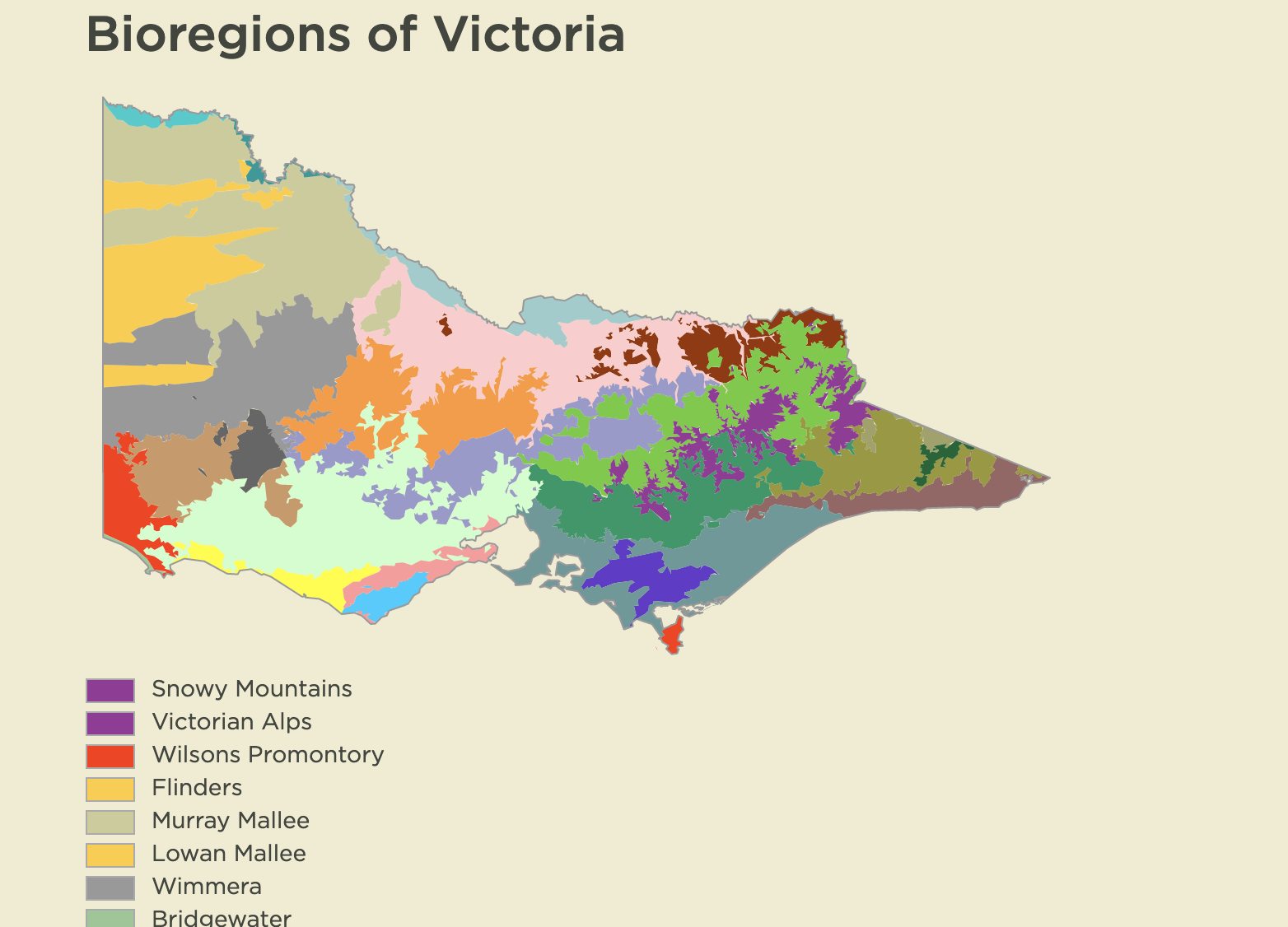


Figure 13: Screenshot of Bioregions of Victoria(vicflora 2019)

In hope of finding the public download for bioregions of victoria, we searched everywhere online for it but was not able to acquire it.

In summary, only a selective few raster and shape files provided by DELWP were added to the VBA dataset, for future improvement of the models’ accuracy, if the bioregion of victoria can be acquired it would definitely increase the models’ accuracy.

**3.4 Separating into different species**

After some consideration, separation of species was implemented in our system. Different species have different environmental needs, placing all the species together and doing one model for them all does not make sense, as some species could live in water and yet another might live in the desert, which if both were placed together the models will not be able to distinguish the importance of the water and desert attributes.

These species will be passed through custom functions that will choose the pseudo absence data and be combined into the same data set and later on be used to train models.

**3.5 Pseudo-absence data generation**

Pseudo absence data generation as it is the main key point suggested from the pseudo absence literature review paper. Using a 1:1 ratio of presence to pseudo absence data using the random drawing technique will yield the best result for machine learning techniques[9].

On top of just following the paper, we have also made our own modification on the pseudo absence generation method, we provided a raster mask which the pseudo absence generation function can use to generate random points inside a certain area.

The pseudo absence function was embedded into our self-defined function for which we have done unit testing and have been noted in the test report.

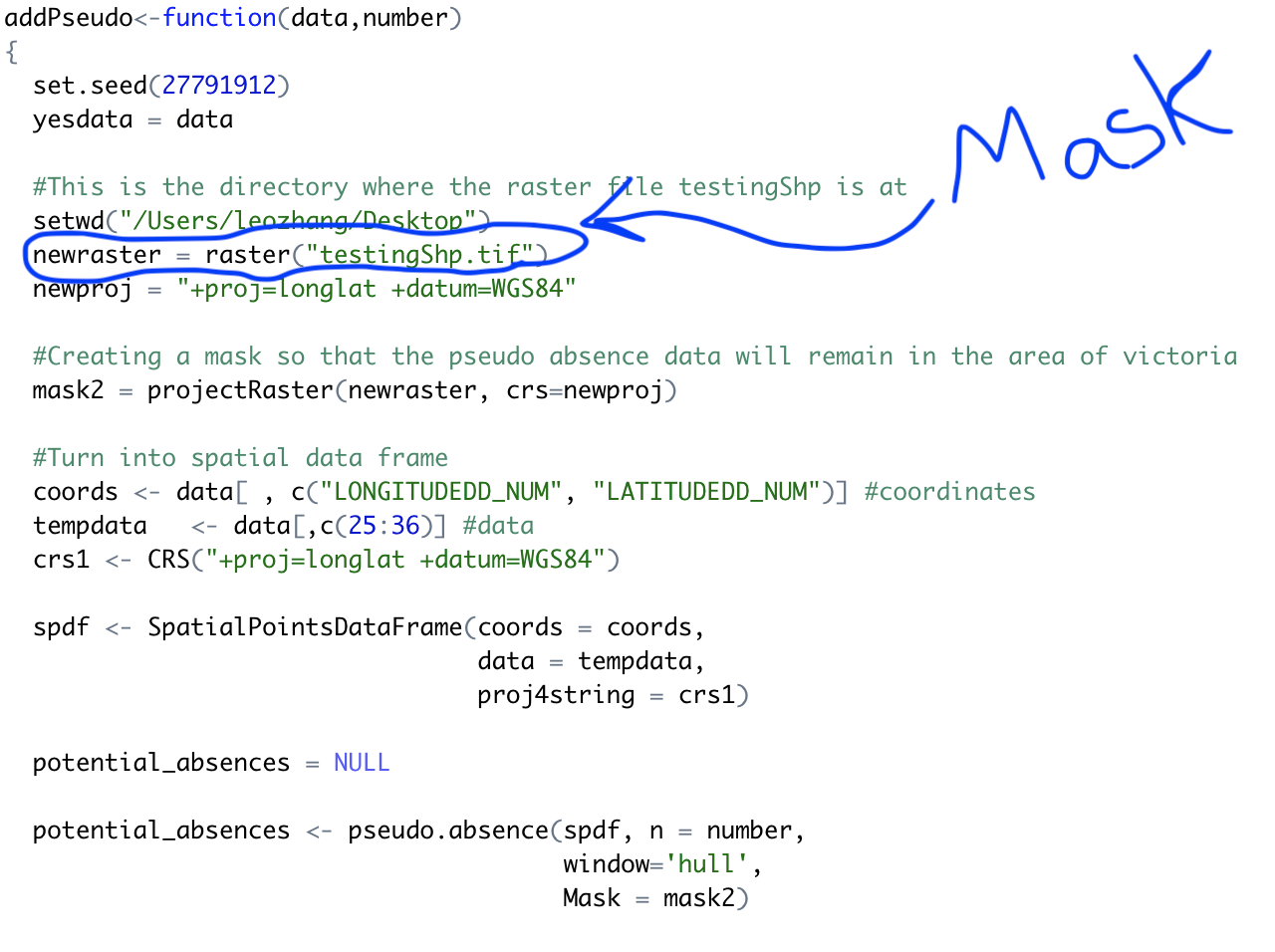


Figure 14: Screenshot of addPseudo function with Raster Mask circled

Result:

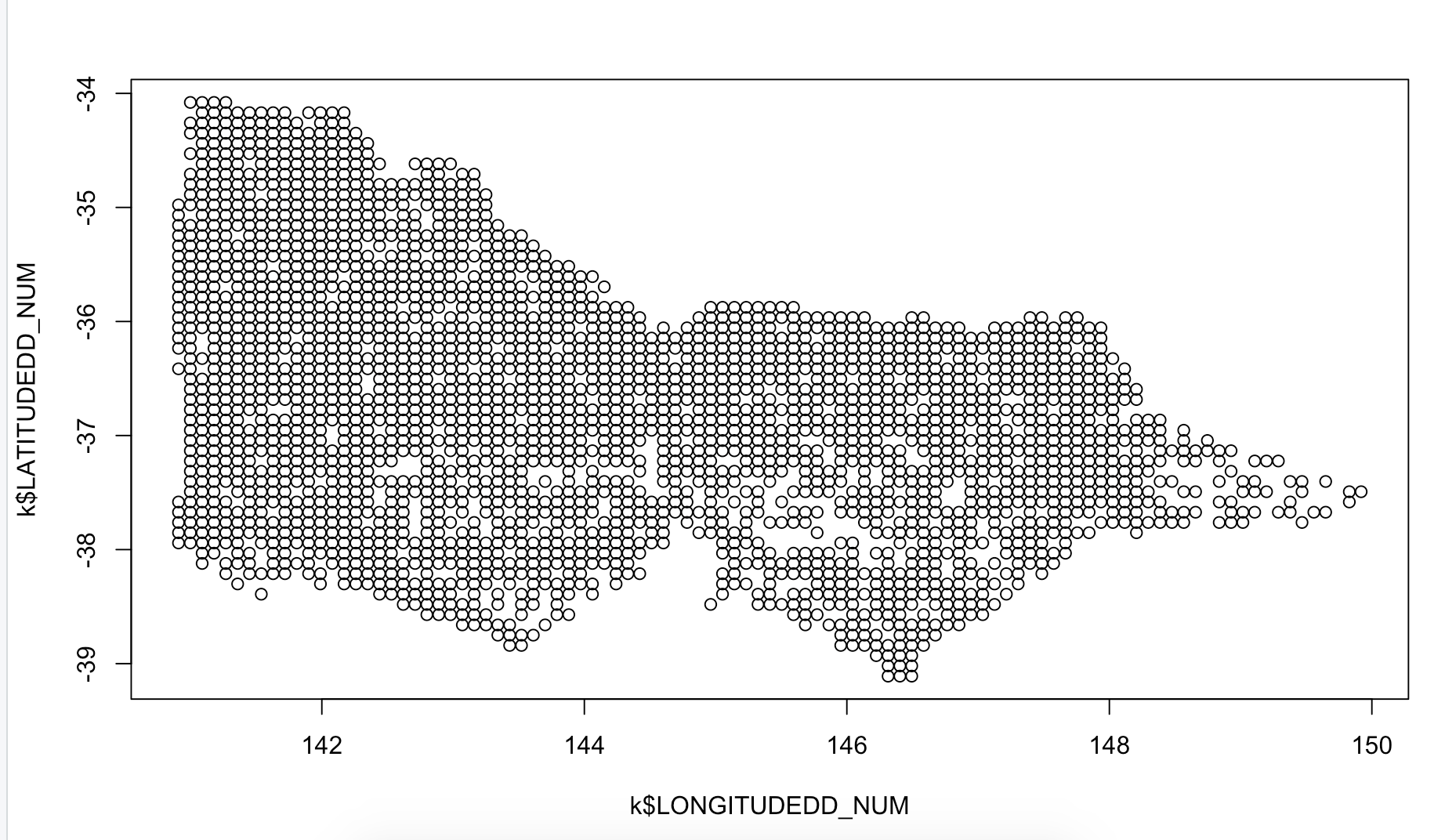


Figure 15: Screenshot of Pseudo absence data plot

The presence data has 2525 points and thus there are 2525 points for the pseudo absence data.

This will be combined with the presence data and then be used to train our models.

**3.6 Random Forest**

Random forest is one of the most popular species distribution modelling techniques, when training a random forest, we have made changes to some of the parameters the random forest function provides. The parameter ntree were focused the most. Ntree controls the number of decision trees that the random forest generates. Different numbers were given and the best one we found were 100, 150 and 250 these three are similar in accuracy.

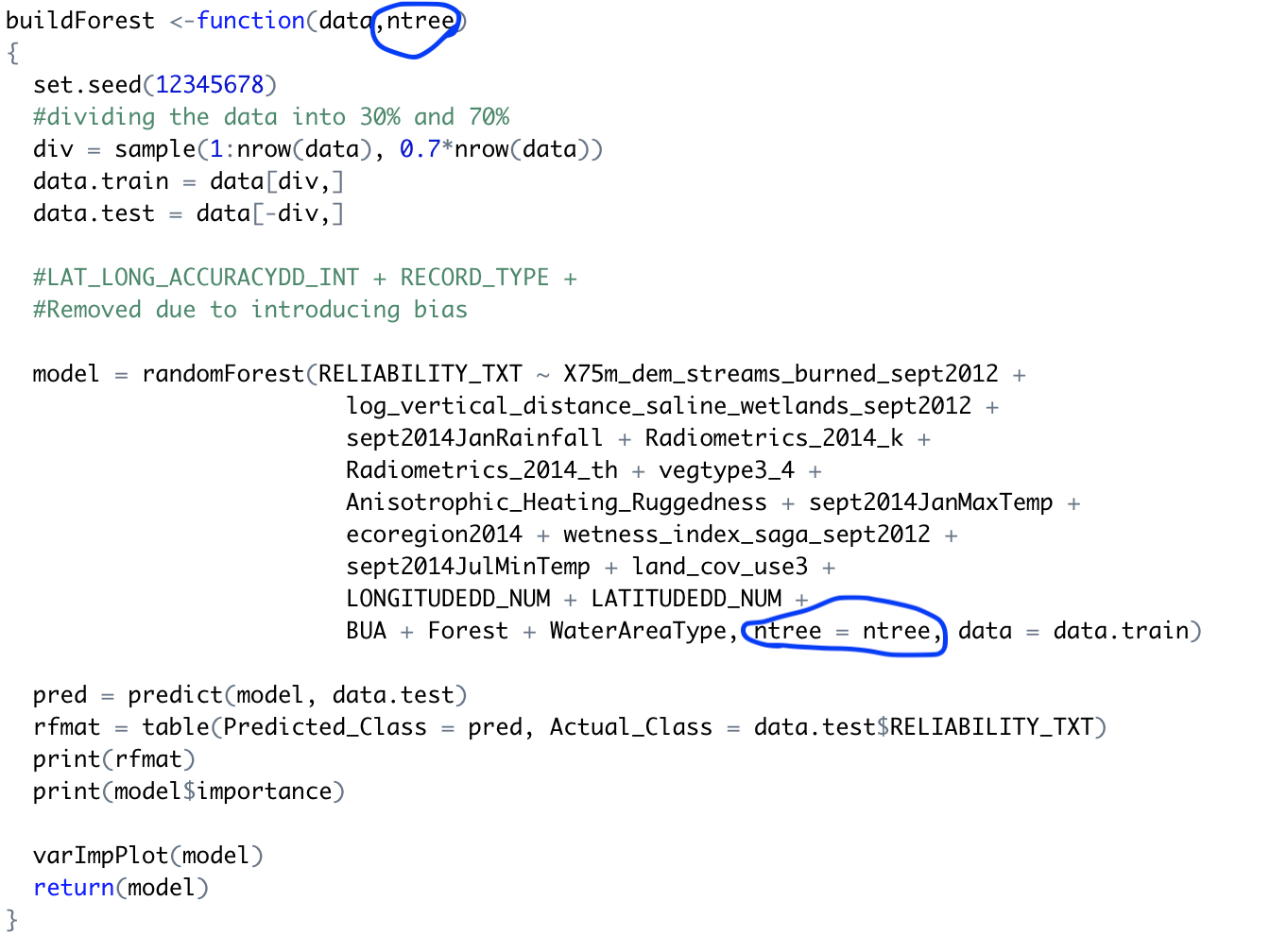


Figure 16: Screenshot of buildForest function with selective parameter circled

(Blue circles are our function arguments when training a random forest)

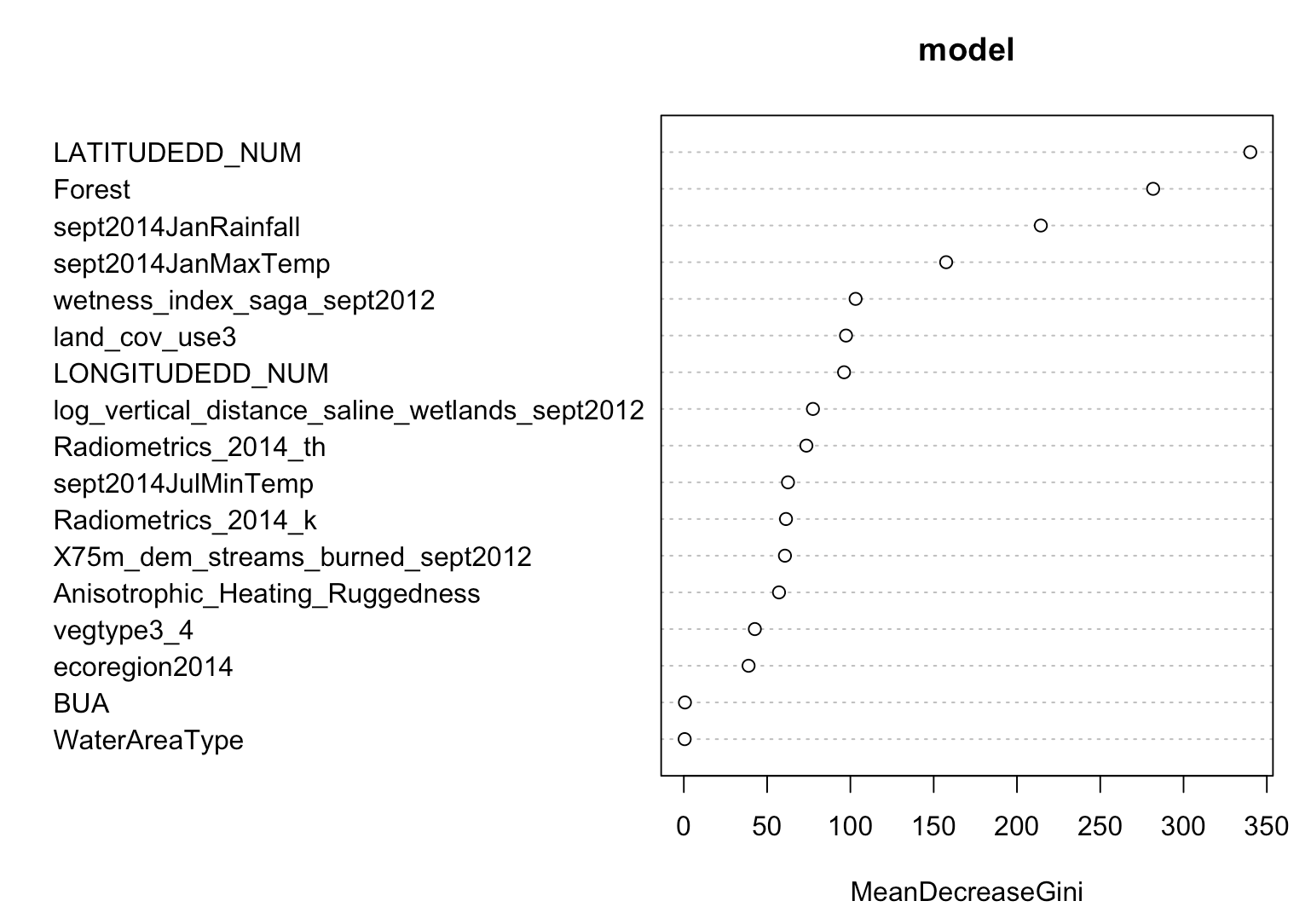


Figure 17: Screenshot of variable importance for “Agile Antechinus” with ntree = 250

The variable importance are plotted on the graphs seems to be predicting well, even though latitude is considered the highest importance predictor, forest comes second and rainfall comes third. From our research it is 100% correct in predicting these two attributes as the more important ones since it is proven that “The agile antechinus inhabits wet or moist forest in the southeastern corner of Australia”(Atlas of Living Australia)

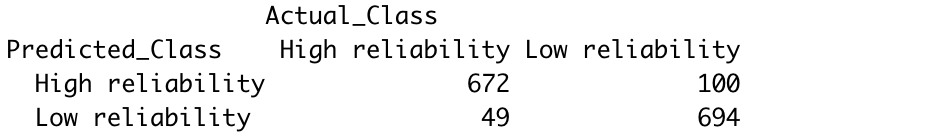


Figure 18: Agile Antechinus random forest Confusion Matrix (Accuracy = 90%)

As seen from above the random forest, is predicting pretty well when the ntree parameter is set to 250.

In summary, we have created a function that allows us to test and play around with the input parameter to find the best possible number and which we have now set to 250 in default.

XGBoost and Svm were not the main focus in this project development, even though XGBoost is a new and upcoming classification model, there is no strong literature paper to support it. The default parameters were used for model training with these models.

**3.7 Imbalance solution**

Imbalanced dataset is one of the critical problems for machine learning. Imbalanced dataset happened when the majority examples hold a very large proportion of the data.

When a model generated by using such data, it is likely to classify any instance to the majority class with a high accuracy, which is clearly not a good classifier. Therefore, without an appropriate method to manipulate the imbalanced data, the model can barely generate a satisfactory result. In this section, the solution to handle the imbalanced data problem will be discussed and explanations will be demonstrated in detail.

**3.8 Proper evaluation metrics**

Since most of the machine learning algorithms are designed to increase the accuracy of the prediction, but not designed to handle the class distribution and class proportion problem. Therefore, the conventional model evaluation methods as follows:

Accuracy of a model = (TP+TN) / (TP+FN+FP+TN),

is no longer appropriate as the model tends to only predict the majority class data. In order to measure a correct accuracy result, we are introduced to measure the Area Under the ROC Curve [1], as it can indicate the trade-off between correctly classified majority and minority instances.

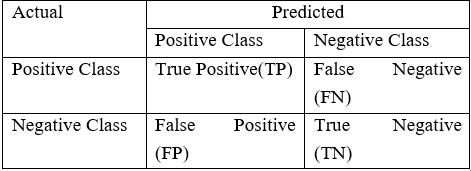
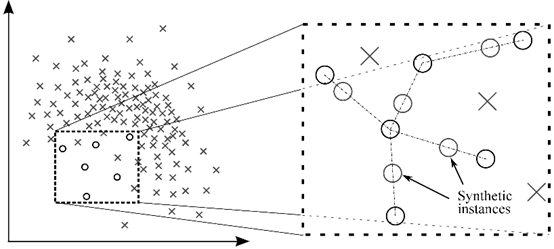


Figure 19: Confusion Matrix

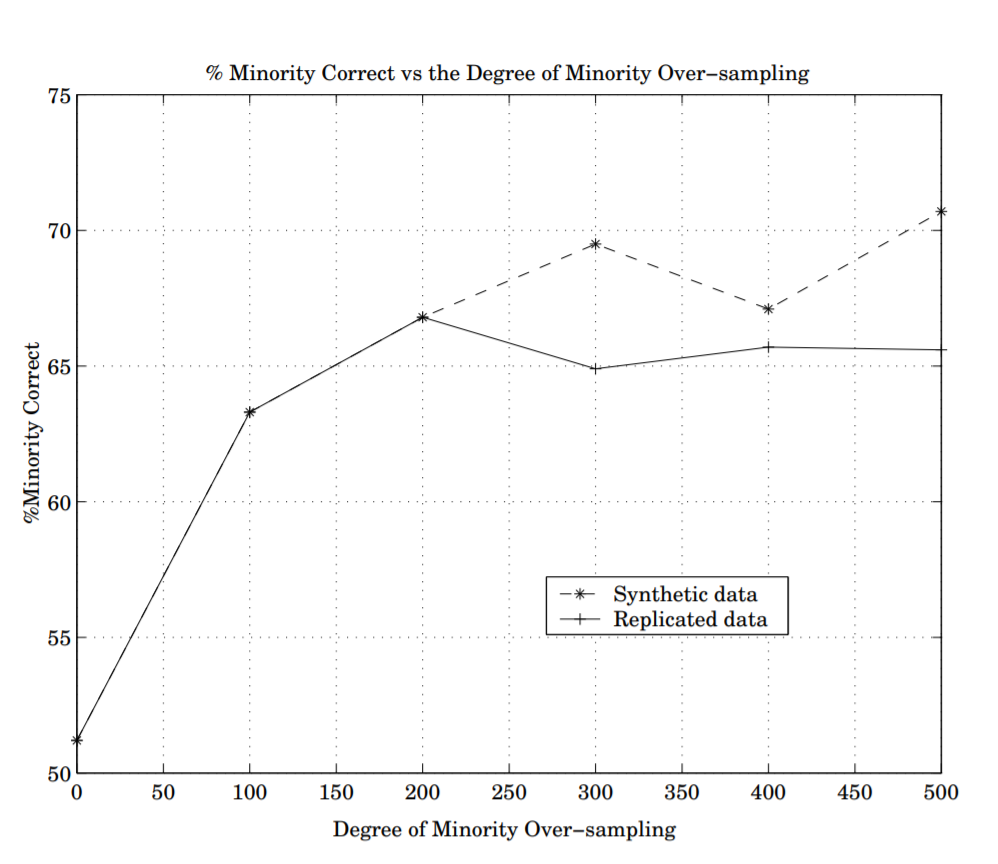
**3.9 Oversampling and Undersampling**

Resampling the data seem to be the most convenient method when faced with an imbalanced problem. Either oversampling or undersampling will be implemented to the data in order to pursue a balanced situation.

The operation principle of undersampling is to randomly eliminate majority class instances, in order to reduce the proportion of the majority class and lead a balanced state of the data. The advantage of this method it helps to improve the model efficiently and reduce run time by shrinking the data observably. However, the shortcoming is also obviously as much potentially useful information is likely to be eliminated and the remaining sample might be a biased sample, hence lead to a huge information loss[4].

On the other hand, Oversampling treats the problem in the opposite way. The Random Oversampling will select the minority class instance randomly and replicate them, hence the number of the minority class instances have increased several times which also increased the minority class’s proportion in the data, and able to achieve balance. Other than undersampling, oversampling will not lead to the information loss problem. But the shortcoming is also existing. As the minority class instances have been multiplied, it means the same sample will be learned multiple times which can cause overfitting to the model. A more reliable approach is called SMOTE (Synthetic Minority Over-sampling Technique)[6]. SMOTE improved the data by applying the K nearest neighbors to all minority class instance independently and introduced the artificial minority instance on the lines between the minority instances and their k nearest neighbors. 

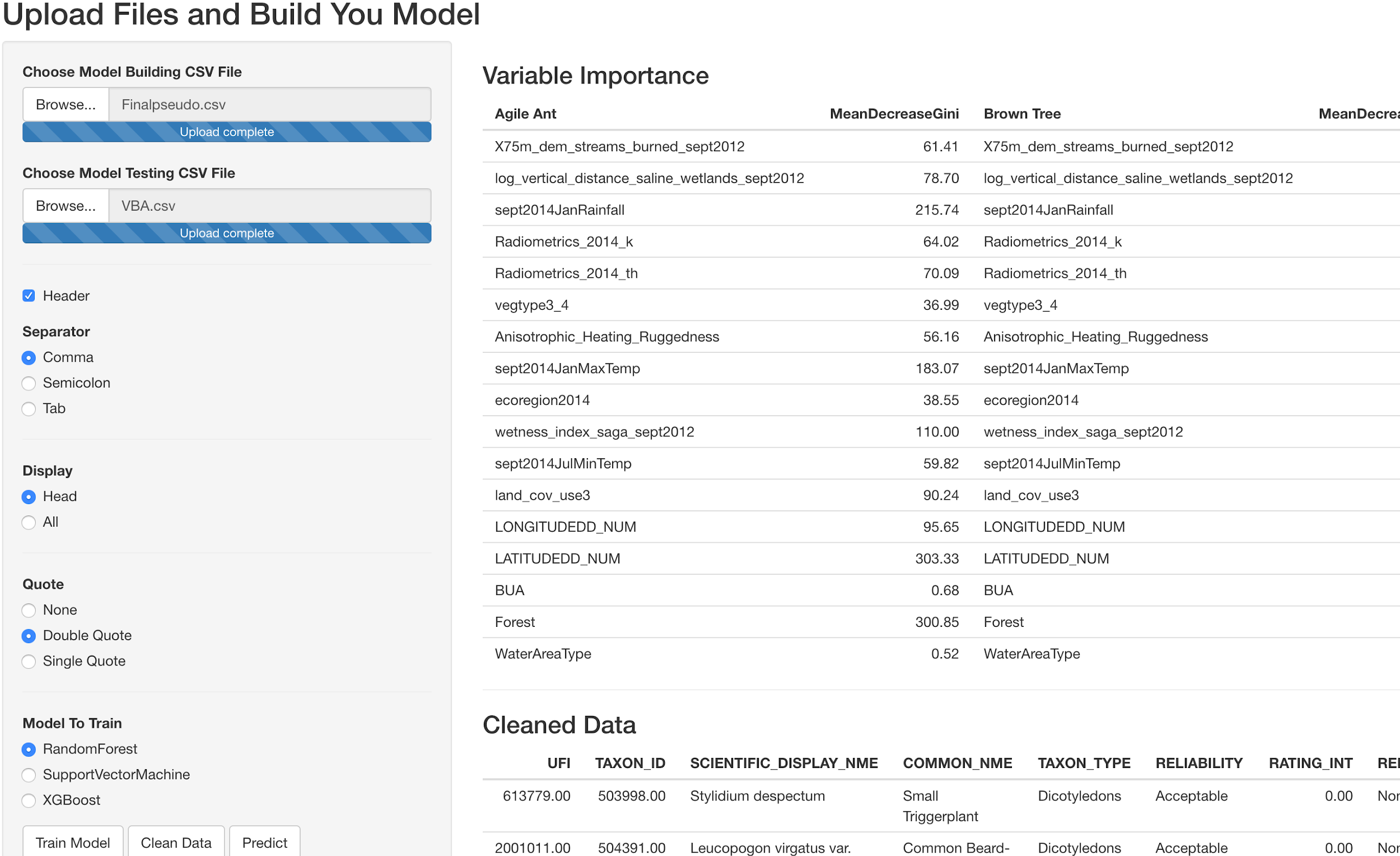
The enhanced technique highly reduces the chance of generating the exact same instance in the same place which can weaken the overfitting and increase the probability of a correct prediction toward the minority class[6].



(Figure: Comparison of decision tree sizes for replicated over-sampling and SMOTE for the Mammography dataset, Adapted from "[SMOTE: Synthetic Minority Over-sampling Techniqu](https://www.semanticscholar.org/paper/SMOTE%3A-Synthetic-Minority-Over-sampling-Technique-Chawla-Bowyer/8cb44f06586f609a29d9b496cc752ec01475dffe)e", by N.V. Chawla, K. W. Bowyer, L. O. Hall, W. Philip Kegelmeyer, 2002, p. 331. )

**3.10 Application implementation**

In this section, the design procedure and decision making of the application will be given. The advantage of the chosen UI design software will be stated. The package used to implement the application will be discussed. And the operating principle will be explained. A screenshot of the application is given below.



The software used to create our project’s user interface is called Shiny. The main reason for it being our final decision is because Shiny is an R package which helps to build an interactive web application with the R script. Since our model is created based on RStudio, hence it becomes a reasonable choice and in return, the compatibility problem with the programing language becomes avoidable.

As the programming language used for this project is R. Many packages other than Shiny is being used in order to achieve the result. Packages “raster”, “rgdal” and “dplyr” is included to handle the data preprocessing problem such as the data extraction and conversion, packages “randomForest” and “ROCR” provide the functionalities of generating model and produce prediction and the packages “ggplot2” helps draw the species prediction map. The functions that were made for Cleaning data, Training model and Predicting model were integrated straight from the Combined.R script into the UI.

To the user interface design, we aim to make it clear, simple and easy to use. Therefore. it only contains a few lines and buttons on the left side panel, but each thing is indicated clearly with an appropriate title or description.

For the usage of the application, when user running the application, only two things will be asked to input which are the training dataset used for build model and the user’s observation dataset, A default training data is provided within the application file, but the user can use their own training dataset if they desire. When the file is selected and the operation status bar shows “File upload completed”, user can select the desired model method then click on the “Train model” button. Then the application will automatically extract the environmental variable from the build-in raster data and combine with the training data according to the longitude and latitude. Furthermore, some pseudo data points will be added in order to balance the sample data and then start model training with the selected method. This action is likely to take dozens of seconds. When training is completed, the model will be stored in the application, after that, the users can click on the “Predict” button and the application will also perform the consistent preprocessing with the “Observation data” then pass it into the predict function along with the model. As a result, the prediction table and a map plot will be printed on the main panel which can be downloaded freely.

**4 Project management**

Project management plays a vital role in project development and implementation. [2]

A suitable project management method can improve the efficiency of the whole project development process. The Agile project management approach is used to manage this project. In this part, the background of project management will be discussed, then the project management approach chosen will be analyzed, the risk management and limitations will be stated in detail as well.

**4.1 Background**

The project management approach was planned at the end of semester one, when we realized that we need an efficient project management approach. When we make plans and assign tasks, a proper project management method can improve efficiency. For example, according to the due date, which is the project delivery time, when should we finished data clean, when should we finished the predictive model development. And for each task we finish, when should we test them and how can we monitor tasks, and for some specific tasks such as UI design, should we seek opinions from the client? To properly management these problems, we select waterfall and Agile as candidates project management approaches. After comparison, we found that Agile is more flexible and easy to monitor, hence we chose to use the Agile approach to manage our project.

**4.2 How was the project conceived and managed?**

In the beginning, a lot of confusion were amongst the team. We were confused as to how the development of the project would go since we have never done a project like this before, most of the other projects we have done actually had a guideline but this one does not and we were required to plan our own milestones and our own path to reach there.

It was a new challenge for our whole team but we think that overall we have done our best for the project. We managed this by asking around other groups and getting a glimpse of what everyone else was doing and then modified it to suit our needs.

We are not experts on programming and we aren't experts on team management but through team effects, we believe we have done our best to finish the project and try to meet all the requirements.

When it comes to managing the progress of the project, we decided to mainly use meeting minutes to monitor and organize what we have done and what we will be doing before the next minute. Meetings are held almost every week so that we can tell each other where we were up to and if there were any problems that we are facing.

For methodology on team management, we considered both Agile and Waterfall as our project management method at the beginning, then we analyzed these two approaches and compared them. For the Agile approach, it is more flexible and easy to monitor and modify if any round of the project cycle goes wrong. We tend to test and review every task after every task is done. For the waterfall approach, we found that it is hard to modify previous steps because there is no monitor step after each task is done, so even a middle step is aberrancy, that step can not be noticed until the final check. [5]. Especially when we develop our application, when we use Agile management, for every function we code, we can test it with test data, if any function we coded is not performing reasonably, or we want to replace it with another function, we can fix it immediately. While if we using the waterfall approach, we can only estimate the whole application after all functions been coded and quite hard to debug and add new functions.

Overall the project was really scary at the start, in semester one we were struggling a lot because we had no idea what we were supposed to do but as time went by the project started to become clearer and clearer. We started to set smaller goals to help us achieve the end goal easier. A bit like building up to the final objective.

Most of the stuff for the whole project were pretty messy because our team doesn’t follow the deadline well enough so every time we set a scrum to do something, we tend to get overtime and thus we have to reduce the time of the next scrum to finish what we are currently doing.

The time constraint did get better at the end as we tried a stricter rule and focing each other to finish the set goal before the next minute. For the user interface, we tried to develop a kind of “user-friendly” interface but ended up making it quite messy because we have so many ideas to put onto the app but have not planned out beforehand so the interface at the start were just functions after functions and not actually thinking about how it would affect the end-user experience. It was improved by communicating between members when we finished a function we would take a vote from everyone as to where they would like the functions to be implemented at.

We have learned so many things along the way from this unit, the main thing we think is time management, organisation, and communication. We think these are the things we lack but the overall relationship between the team members was good and we encourage each other to do our best.

**4.3 Resource execution and planning**

There are many resources we used in this project. We choose R as our programming language, as R can handle most of the functionality we have in mind. We can use R to clean the datasets we need, and R also has many machine learning packages such as random forest, SVM and xgboost, we can use these packages to build our model. There is also a Shiny package which we can use to build our UI, all three of us do not have much experience in building UI, so we choose an app that can perform all the functions we need and need little coding.

The whole project management can be separated into three parts: data clean and extraction management(planned end before week 3), predictive models developing management(planned end before week 8 ) and application and UI developing management(planned end before week 10).

The main resources used are listed below:

Software used: Rstudio, QGIS

Hardware used: Macbook, Windows Laptops, Windows PC.

Human Resources: Leo, Haoyang, Xianran

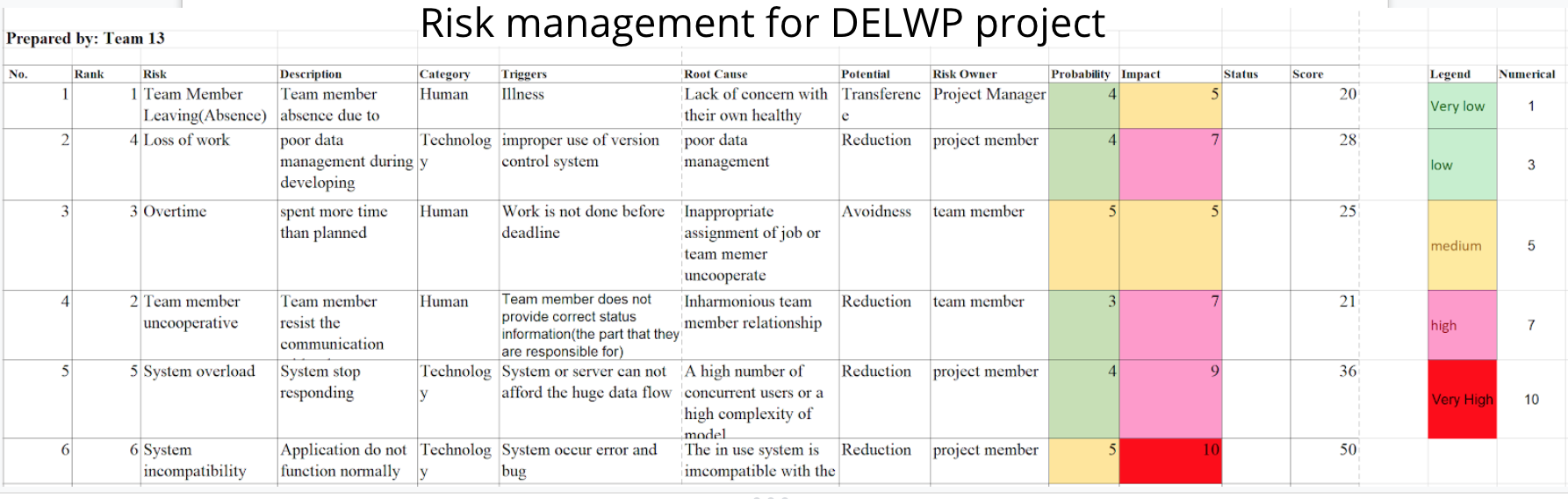
**4.4 Risk management**

Risk management plays a vital role in project management, efficient risk management can ensure the stability of the project process[3].

Many steps in this project contain risks that may affect the final completion of the whole project, especially time management and technique issues.

Time management will directly affect the quality of a project, since not enough workload on some vital functions may affect the performance of the project[2]. For every task in the data preparation step and model development step, we estimate the time needed, and assign 120% time to each task. For tasks completed over time, we chose to reduce time in the next step. For example, If data preparation and model development part take too much more time than expected, we will consider cut some functions of the application and UI, to make sure the completion of the project.

Technique issue risks is also a big risk we concern. In every implementation of code and design, we may run into some problems and stuck for a while, we plan to seek help and advice from classmates and tutors if we get stuck. Also, we will analyze the difficulty and complexity of planned models and functions, we deleted some unrelated functions and some not major but very difficult functions.

****

(risk management for DELWP project)

The main risk we have encountered which we had hoped not to happen was the System incompatibility. It was rated high on impact because it is. Previously we had tried to avoid it by reducing the chance of it happening by sharing the code between everyone of the team and whenever we would implement a new function, we have to pass it to everyone and make sure that it works on everyone’s computer. The worst thing happens at the end and we had no choice but to give up, an app breaking bug happened on Windows two days before we had to deploy the system. It suddenly stopped working on Windows due to some unforeseen errors. So the method we chose was to give in to the error and make it a MacOS dependent software.

**4.5 Limitations**

There are many limitations occurred during the whole process of this project. In this part, three main limitations will be discussed, as well as some approaches to solve them.

The first limitation is poor communication, our team members only meet once a week, and we did not communicate a lot through chat software such as messenger and Wechat. Due to poor communication, we often finish tasks just before the deadline, with quality far worse than we planned. One way to solve this problem is to increase the times we meet each week, communicate more about the tasks of each team member during meetings, and encourage the members to speed up if he is falling behind.

The second limitation is the poor time management during the whole project, because of lack of experience, we were too optimistic at the beginning. We should not evenly assign time to tasks with different difficulty levels, which is the cause of poor task difficulty estimation. We can solve this by analyzing each task more precisely, for example, when we want to extract data from QGIS in some specific format, before we assign time to this task, we need to consider if any of our team members know how to do this already or we need to study this software from scratch, then make estimation about the time might need to solve this task.

The third limitation is team collaboration, we could save much time by collaboration. During the whole project process, in every meeting, we tend to assign tasks to each member, and each team member just focuses on their own tasks, if get stuck no one asks help until planned due time, and no one tends to care about the tasks assigned to other members. All three team members trust each other too much and feel shy to seek help.

Because of three team members studied different units in the last two years, we have different knowledge backgrounds. For example, Xiaoran is good at Python and math modeling, while Haoying and Haoyang are good at R, for many tasks such as data cleaning and application development, if we do it together, we can improve the efficiency a lot.

**5 Outcomes**

At the end of the project, a Species Distribution Prediction Application which contains 3 different prediction model methods with accuracy around 85 to 95 percent is obtained. Implemented with a simple and clear user interface and a method guide is given in the source file which aims to make it easy-to-use. User only have to provide their own observation data and as a result of the application, the prediction of the species exist within the training data will be presented in a table, the extra environmental information of the location and the importance of those variables will be given, and eventually a diagram will be plotted to indicate the correction of species.

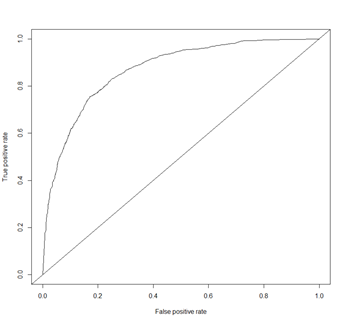
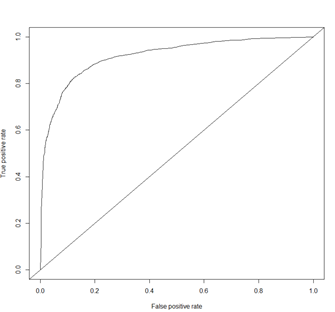
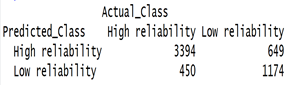
**5.1 Hypothesis**

At the beginning of this project, the major hypothesis is derived about the impact of using imbalanced dataset. The first hypothesis is using the imbalanced data will cause an impact on accuracy. Then we prepared two sets of data. Both had the factor called the “RELIABILITY\_TXT” which can either be “High reliability” or “Low reliability”. One of the data is manipulated to set as unbalanced data which contains about 12000 of “High reliability” records but only 6000 of “Low reliability” records. The other dataset being the control group, which used the same dataset as the previous group initially but a SMOTE function shown below is then applied to it.

Which oversampling the minority class and the “Low reliability” records had increased to 9000 rows. Then they were learnt by the Random Forest model and the confusion matrix was recorded and the model accuracy was calculated using the AUC shown below.

As a result, the model with balanced dataset achieved 92% accuracy while the imbalanced dataset one only had 87% of accuracy. It seems the difference was not huge. However, if we look carefully at the confusion matrix, we can see that the model with the balanced dataset has double amounts of the correct prediction than the imbalanced one. The reason that it is important is because when the model handling imbalanced data, it is more likely to be “lazy” or “guessing” while being trained due to most of the outcomes will fall into the majority class as it occupies a large proportion. Therefore, the model trained by the imbalanced data will tend to predict the “High reliability” and achieved a relatively high accuracy in this situation. On the other hand, the model with balanced datasets was more willing to predict the minority class, but it also achieved a higher score. Hence, it supported our hypothesis that using imbalanced dataset will cause an impact on accuracy.

（Left: RF model with Imbalanced dataset, Right:RF model with balanced dataset)



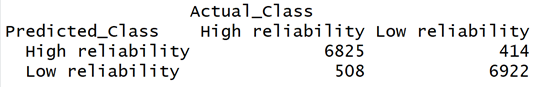
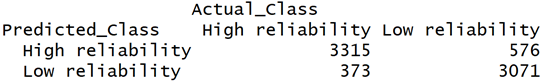
 

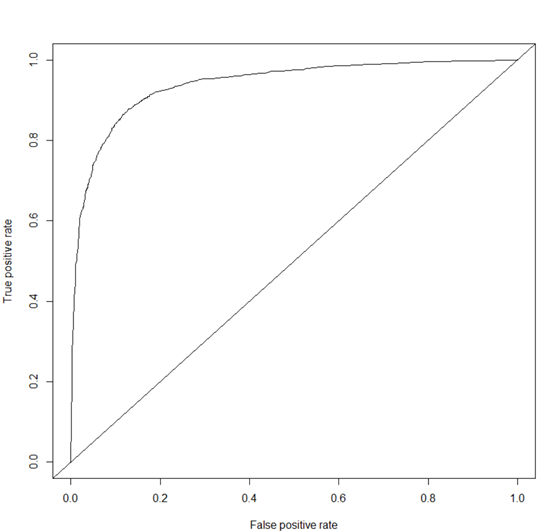
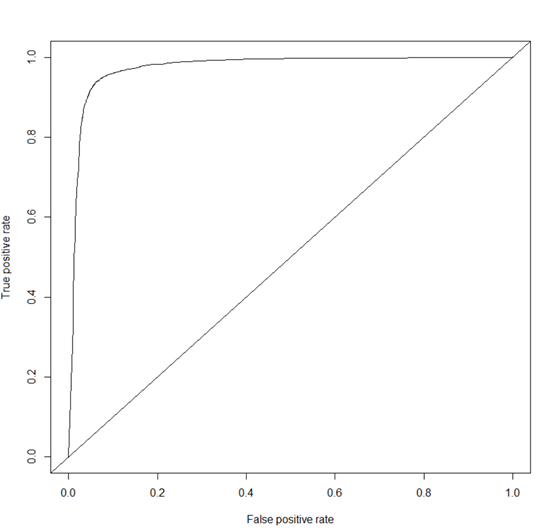
(The float number shown above is the accuracy of the model)

In addition, the second hypothesis we have made is the larger dataset being used, the higher the accuracy of the model we obtained. To process this test, we set up two datasets, which also contain the “RELIABILITY\_TXT” as factor type and “High reliability” and “Low reliability” as variables. The first data had 24448 rows, half of them were the “High reliability” records and the rest of them were the “Low reliability” records. And the other data also had the exact same proportion for the two classes but the size was 2 times larger than the previous one. Those datasets were manipulated to the perfect balance state. Which aim to minimize the other factors that might influence the result and emphasized the impact of the size. We started the test by putting them into the Random Forest model and using AUC to calculate the accuracy.

The result is shown below , the model generated by the smaller size data given a slightly smaller result than the other. Since all the variables we predicted was removed, but also due to the characteristic of the Random Forest method which made it had strong resistance to overfitting. Hence, from the improved result of the model with larger dataset. It proved that the more information was given, the higher accuracy the model able to achieve.

（Left: RF model with smaller dataset, Right:RF model with larger dataset)



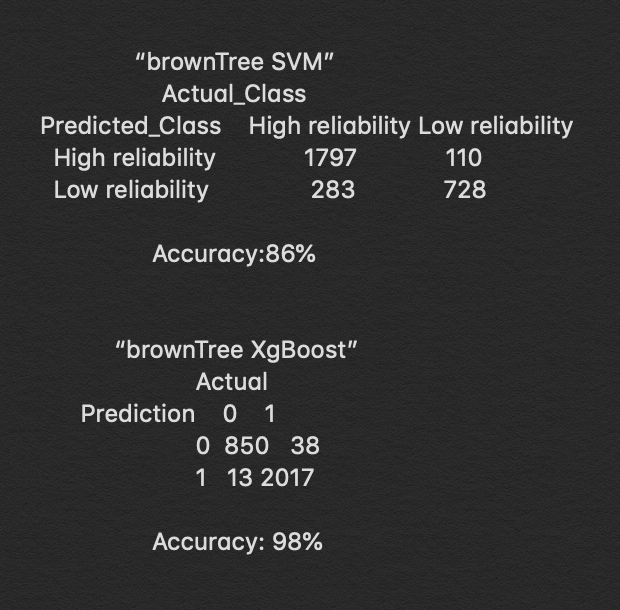
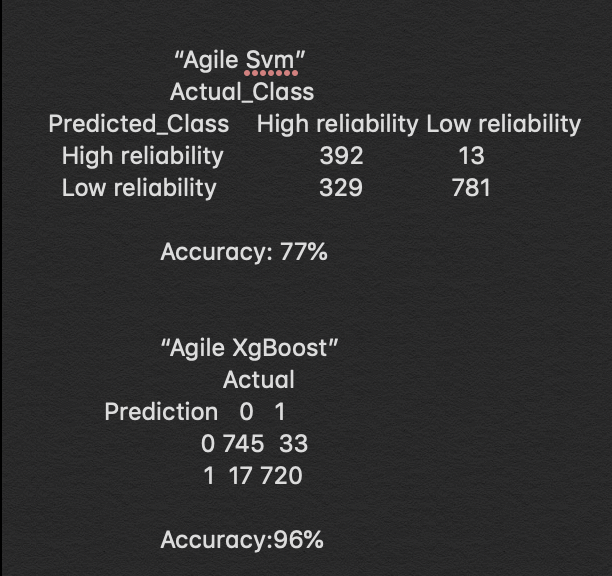
(The float number shown above is the accuracy of the model)

**5.2 XGBoost and SVM:**

The confusion matrix score is calculated to test the accuracy for XGBoost and SVM with only pseudo absence data with no SMOTE.

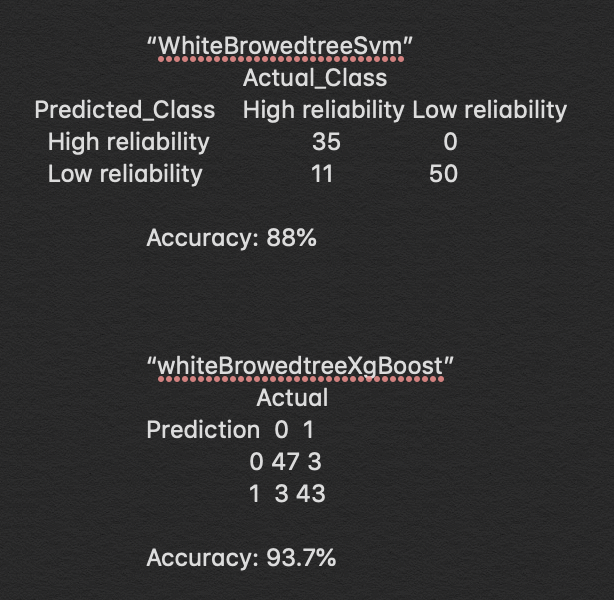
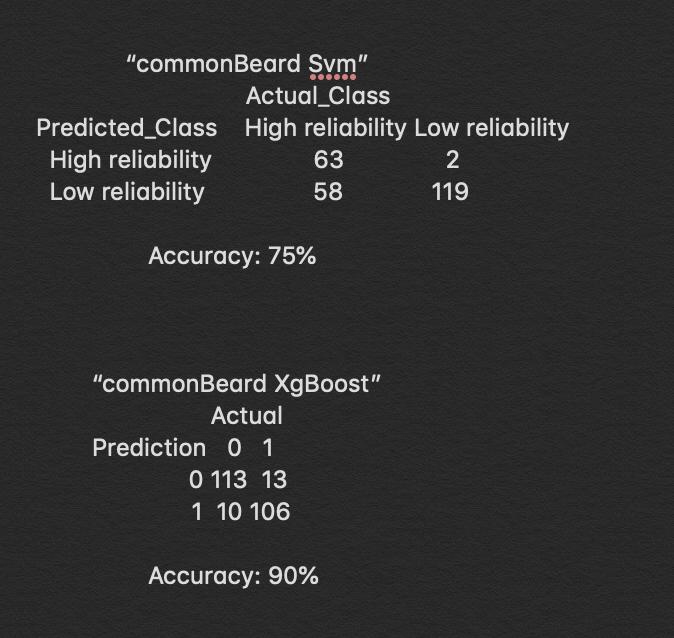
The accuracy of XGBoost is very high compared to SVM, with both using the default parameters. We can see that for “Agile Antechinus species”, SVM only has an accuracy of 77 while XGBoost has reached 96%

For the “Brown Treecreeper” SVM has achieved 86% and XGBoost has gotten 98%

****

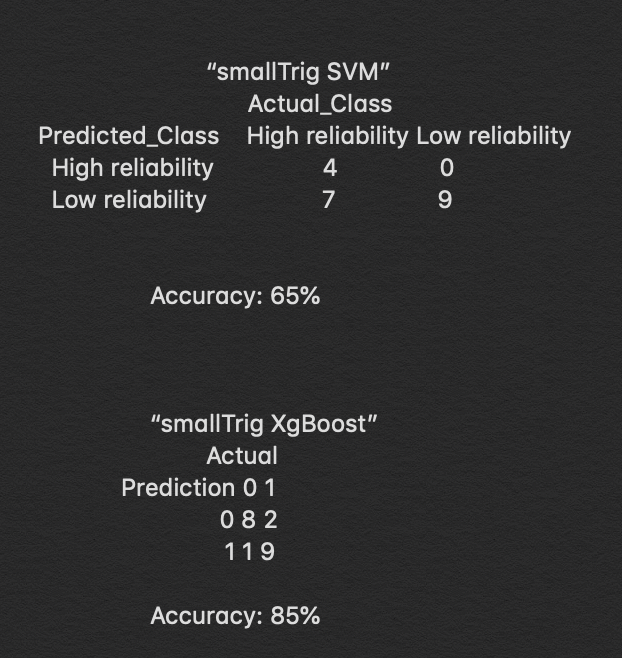
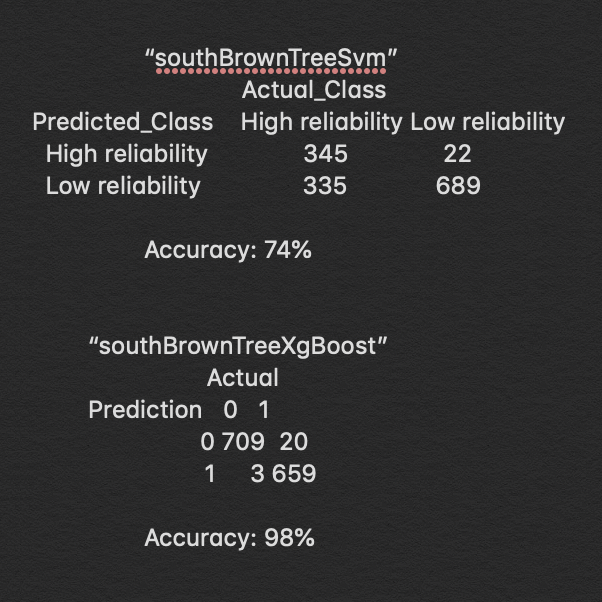
For “Common Beard-heath” Svm’s accuracy: 75 comparing to XGboost which is miles in accuracy with a 90% accuracy.

For “White-browed Treecreeper” Svm has reached 88% while XGBoost has reached 93.7%



For “South Brown Tree Frog” the Accuracy for SVM is only 74% while XGBoost has achieved 98%

And lastly for the species “Small Triggerplant” the accuracy of SVM is only 65% and the XGBoost has achieved 85%



From the comparison of the two models we can clearly see that the XGBoost is way better at predicting the species than SVM even for the Small Trigger plant species where there were not many data points at all, the XGBoost algorithm still predicts 85%. And for species with more datapoint, XGBoost can get to 98%.

Though the evaluation of the models is not the focus on this topic, the deliverables were the main User Interface App that we have created. And the functionality that our system can provide on the Shiny app.

**5.3 Limitations of** **software**

Since the application is built up with many open source tools, especially the machine learning function. Therefore, the operability of the function is limited. Hence more difficult to tune the hyperparameter. Consequently, imbalanced data problem can be hard to solve by adjusting the weight of each coefficient or adding l2-regularization to the model. which leads to solving this problem by using oversampling.

Oversampling can also bring limitations to performance. As long as the absence data is generated artificially rather than the true absence data collected by observation, it is likely to impact the calculation of probability of presence as the noisy might be replicated or generated can cause overfitting. Due to the true absence data is not provided in DELWP project, this limitation might be unavoidable. However, this limitation has been minimized as we using an enhanced and more reliable method “SMOTE” to generate absence data, and the model methods we implemented have a strong resistance when facing overfitting.

The long performance-period is another limitation. Since we aim to improve the model accuracy and generate more reliable pseudo data. The large amounts of environmental information from both raster data and shape file provided by DELWP are being widely used and become the most essential factors to determine a point. In order to achieve this, a data preprocess will be applied to every data the application received. Particularly, for each receiving dataset, the program will extracting the raster data and shape file, and combine the received data with the extracted environmental information according to the longitude and latitude. Thus, multiple times of extracting the massive raster data can be time-consuming.

Furthermore, as we mentioned before, the raster data contains much useful information, consequently, the size of the raster will also be large and will occupy much storage. Which is very user-unfriendly. An appropriate solution to this problem is making it becomes an online service that will receive the uploaded file, manipulate in the local server then send back the preprocessed data. However, due to the time limit and lack of networking experience, the solution becomes unprocurable.

Besides, the species that are not mentioned in the training data will become unpredictable. Because the models are created separately for each species. Which means each model is exclusively used for a specific species. Therefore, the cross species prediction using maximum likelihood becomes infeasible.

Lastly, the main limitation of the whole project is a docker container. Even though we have implemented a working user interface to the end-user, it is a mess to get the end user to implement our system because our system is MacOS based which means that users on a windows operating system will not be able to run our app and get our output. If there were time, we would definitely build a docker container so that no matter which operating system the user is using, they can always be able to run our product.

**6 Conclusion**

This report concludes the details and explanations of every step of the DELWP project. Overall, our main modeling method random forest has proven to be a good species distribution model, it has produced the best overall accuracy of 97% for random forest using pseudo absence data and SMOTE to balance the data. By using raster and rgdal packages, we successfully extract the environmental variables such as rainfall and temperature from the raster data according to the longitude and latitude value, which provide more useful information and help to generate a more reliable result. The using of the random forest algorithm to build the predictive model clearly explained, followed by the implementation of SVM and xgboost. We found that xgboost performs best among three machine learning algorithms, which is 93% accuracy, Random forest performs 90% accuracy, while SVM only achieved 82% accuracy, but xgboost gets overfit frequently. The deliverable is packaged with the Shiny user interface. The user interface is designed to be as simple as possible and user-friendly. The button is presented clearly with an appropriate naming. On the other hand, an instruction manual is given in the source file called “Readme.txt” which will guide the user. The number of implemented model methods can be easily switched through the user interface. The result of the prediction will be shown in the table form which is able to be downloaded, a scatter diagram indicated by the prediction data will be given to visualize the distribution of presence and absence of each species. In addition, since the application had passed multiple unit tests and had been working on both Windows and MacOS operating system, hence it seems to be quite stable and consistent and adapt to any working environment. Also, the user- friendly implement structure will automatically rollback when incorrect files have been uploaded, hence, the crashing caused by misoperation becomes avoidable and consequently less limitation to the user.

Besides, in this report, how is the project management conceived and how is this project managed is stated clearly, project development resource and project plan are also been explained. The implementation of the Agile project management approach is also detailed discussed in this project, and the risk management and limitations of the way our team worked are discussed as well. From the project management experience, we found that we need to improve our communication skills, and time management skills need to be improved too.

As for the team experience, even though we had a rollercoaster of a ride, there were ups and downs through the whole year period, but team members were always there to cheer each other up and did not get mad and anyone, so in general, we think we had quite a good experience during the whole project development process, we all trust each other and everyone tries their best for each task, even for some tasks we did not perform very well, such as the mid-semester presentation, we can notice the problem and try to improve it immediately, and make sure we can do better next time.

References:

[1]A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognit. 30 (1997) 1145– 1159.

[2]Burke, R. (2013). Project management: planning and control techniques. *New Jersey, USA*, *26*.

[3]Chapman, C., & Ward, S. (1996). *Project risk management: processes, techniques and insights*. John Wiley.

[4]D Anyfantis, M Karagiannopoulos, S Kotsiantis, and P Pintelas. Robustness of learning techniques in handling class noise in imbalanced datasets. In Artificial intelligence and innovations 2007: From theory to applications, pages 22. Springer, 2007.

[5]Highsmith, J. (2009). *Agile project management: creating innovative products*. Pearson education

[6]N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, SMOTE: synthetic minority oversampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[7]Evans J.S., Murphy M.A., Holden Z.A., Cushman S.A. (2011) Modeling Species Distribution and Change Using Random Forest. In: Drew C., Wiersma Y., Huettmann F. (eds) Predictive Species and Habitat Modeling in Landscape Ecology. Springer, New York, NY

[8]Iturbide, M., Bedia, J., Herrera, S., del Hierro, O., Pinto, M., Gutierrez, J.M., 2015. A framework for species distribution modeling with improved pseudo-absence generation. Ecological Modelling.

[9]Barbet-Massin, M., Jiguet, F., Albert, C., & Thuiller, W. (2012). Selecting pseudo-absences for species distribution models: how, where and how many?. Methods In Ecology And Evolution, 3(2), 327-338. doi: 10.1111/j.2041-210x.2011.00172.x

[10]Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[11]Breiman L (2001a). Statistical modeling: the two cultures. Stat Sci 16:199-231

[12]Hastie T, Tibshirani R, Friedman J(2009) The elements of statistical learning: data mining, inference, and prediction. 2nd edition. Springer, New York.

[13]Mi C, Huettmann F, Guo Y, Han X, Wen L. 2017. Why choose Random Forest to predict rare species distribution with few samples in large undersampled areas? Three Asian crane species models provide supporting evidence. *PeerJ* 5:e2849

[14]Elith, J. *et al*. 2006. Novel methods improve prediction of species’ distributions from occurrence data. *Ecography* 29: 129– 151.