UNIVERSITAT DE BARCELONA

FUNDAMENTALS OF DATA SCIENCE MASTER'S THESIS

Analysis of altitude sickness by Lake Louise Score prediction and ophthalmological data studies

Author:
Huang CHEN

Supervisor:
Dr. Oriol PUJOL VILA

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamentals of Data Science

in the

Facultat de Matemàtiques i Informàtica

July 3, 2018

Contents

Abstract v

Acknowledgements vii

1 Introduction 1
- 1.1 Project's motivation and introduction . . . 1
  - 1.1.1 Project's motivation . . . 1
  - 1.1.2 Sherpa-Everest'2017 project . . . 1
- 1.2 Objectives . . . 1
- 1.3 Contributions . . . 2
- 1.4 Report layout . . . 2

2 Background information 3
- 2.1 Linear models . . . 3
  - 2.1.1 Support Vector Machine . . . 3
- 2.2 Feature subset selection . . . 5
  - 2.2.1 Best Subset Selection methods . . . 5
    - Brute-force Feature Selection . . . 5
    - Forward Feature Selection . . . 5
    - Backward Feature Selection . . . 6
  - 2.2.2 LASSO . . . 6
  - 2.2.3 Relaxed-LASSO . . . 6
- 2.3 Brief intro to statistic in classification problems . . . 7
  - 2.3.1 Bayesian approach: Beta-binomial distribution . . . 7
  - 2.3.2 Frequentist statistic: Permutation test . . . 8

3 Proposal: "Iterative backward relaxed SVM selection" 9
- 3.1 Problem of Relaxed-Lasso and Best Subset Selection . . . 9
- 3.2 Iterative backward relaxed SVM selection . . . 10
  - 3.2.1 $L1$-norm SVM . . . 10
  - 3.2.2 Iterative backward relaxed SVM selection . . . 10

4 Lake Louise Score analysis 13
- 4.1 Lake Louise Score System . . . 13
- 4.2 Data description . . . 13
- 4.3 Problem modelling and implementation . . . 15
  - 4.3.1 Data cleaning . . . 16
  - 4.3.2 Feature engineering . . . 17
  - 4.3.3 Feture selection . . . 17
  - 4.3.4 Train a classifier for LLS prediction . . . 20
- 4.4 Results . . . 21
- 4.5 Statistical validation of the results . . . 22
  - 4.5.1 Permutation test . . . 22

4.5.2 Beta-binomial distribution support 23

5 Ophthalmic analysis **25**
5.1 Data description 25
5.2 Problem modelling and implementation 27
5.2.1 Data cleaning 27
5.2.2 Feature engineering 27
5.2.3 Modelling hypoxia suffering grades 28
5.2.4 Feature selection 29
5.2.5 Prediction 31
5.3 Results 32
5.4 Statistical validation of the results 34

6 Conclusion and future works **37**
6.1 Conclusion 37
6.2 Future works 37

Bibliography **39**

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Analysis of altitude sickness by Lake Louise Score prediction and ophthalmological data studies

by Huang CHEN

In this work we analyse the data obtained from the Sherpa 2017 Everest project. We focus the study on two problems. First, we study the cause of altitude sickness by analysing the factors that influence the most in the prediction of the Lake Louse Score. Second we study the affection of damage by hypoxia in ophthalmic data. In order to help with this studies, we propose the Iterative Backward Relaxed SVM selection method. This method sorts the factors that are related to the prediction result. With the obtained ordered factors list, we perform the feature selection to remove the uncorrelated factors. The prediction of both Lake Louise Score prediction and the ophthalmic data studies got positive results.

Acknowledgements

First and foremost, I would like to express my most sincere gratitude to my advisor Dr. Oriol Pujol Vila , for his valuable and constructive suggestions during the planning and development of this work, as well as he generously offered his spare time to guide me the philosophy of life and motivate me.

I would also like to extend my thanks to various people for their contribution to this project; Dr. Jose Manuel Soria, for giving me this opportunity to be a part of Sherpa-Everest'2017; Mr. Angel Martinez, for sharing his experience in the medical data analysis with me and discussing the problems with me; Dr. Ricardo Pedro, for contributing his immense ophthalmic knowledge and for giving me useful recommendations; All the participants of the Sherpa-Everest'2017 project, for offering me their data.

Finally, I wish to thank Dr. Alex Haro, for introducing me to Dr. Jose Manuel Soria; my parents, for their support; all the professors of the Foundations of Data Science Master in the University of Barcelona, for teaching me and sharing their knowledge with me; my classmates and friends, for their encouragement throughout my study.

Chapter 1

Introduction

1.1 Project's motivation and introduction

1.1.1 Project's motivation

Hypoxia is a condition in which there is oxygen deficiency in a habitat or a body part, usually due to an insufficient concentration of oxygen in the blood. Generalized hypoxia occurs in healthy people when they ascend to high altitude, where it causes altitude sickness leading to potentially fatal complications: high altitude pulmonary edema (HAPE) and high altitude cerebral edema (HACE). Hypoxia not just occurs in healthy people at high altitude, but also in patient of Chronic Obstructive Pulmonary Disease (COPD) at normal altitude. Hypoxia in COPD results in a relatively focused pattern of impairment in measures of memory function and tasks requiring attention allocation. In this project, we aim to study the cause of generalized hypoxia(Altitude sickness) by understanding what are the most related factors that cause altitude sickness. In order to better understand altitude sickness and model the problem, a high altitude environment is required. Thus, the Sherpa Everest 2017 project was launched.

1.1.2 Sherpa-Everest'2017 project

In 2017, the scientific team of the Sherpa-Everest'2017(SE2017) project arrived to the base camp of Everest to carry out a pioneering study that analyse the genetic, biological and clinical impact of the lack of oxygen (hypoxia) in trekkers, European climbers and Sherpas people during the Everest ascent (8,848 m).

The participants are exactly 11 trekkers, 17 climbers and 28 Sherpas. The 17 climbers are led by Ferran Latorre i Torres, a famous Spanish traveller and mountaineer(*The project description in Ferran Latorre's blog*). Sherpa are one of major ethnic group native to the most mountainous regions of Nepal, the Himalaya. The 11 trekkers are actually 11 well-trained scientist which including geneticists, neurologists, cardiologists, physiologists, biochemists and other doctors from Nepal. Furthermore, as we mentioned before, hypoxia also occurs in COPD. Hence, there were 50 COPD patients who also participate the Sherpa Everest 2017 project by contributing their medical data. However, we don't use any COPD patients data in this study.

1.2 Objectives

As we mentioned previously, we aim to understand and find the most related factors that cause altitude sickness. Thus, we build the project by doing two different analysis: Lake Louise Score(LLS) analysis and Ophthalmic analysis. We started with only LLS analysis. However, the SE2017's scientific team provided us the ophthalmic

data of trekkers afterwards. They consider that ophthalmic study in altitude mountain sickness(AMS) is completely new and we would reach discovering new useful information. For this reason, we added the second analysis and expect to be able to explain the cause of hypoxia in an ophthalmic manner.

1.3 Contributions

Based on the two analysis we mentioned previously, we can summarize the project's contributions as follows:

- We are able to predict the LLS value with an certainly accuracy by given individually physiological data and stage-wise information such as the accumulated distance reached.
- We reach to create an measurement to evaluate how people suffer from high altitude hypoxia by the given ophthalmologic data.
- We propose the Iterative backward relaxed SVM selection method for the feature subset selection process.

With regard to the above three points, we expect the result of this project could help medical researchers to better understand hypoxia, and improve the diagnosis or develop treatments that improves the symptoms and consequences of this illness.

1.4 Report layout

In order to clearly show the pipeline, we organize this report as follows:

- Chapter 2 explains the background information, which are the existing methodologies that we use for this project.
- Chapter 3 points out why the existing methods are not appropriate for the project's analysis, and then proposes a new one.
- Chapter 4 discusses the data we used for the first analysis which is Lake Louise Score analysis, and its implementations. Moreover, the results are also shown in this chapter.
- Chapter 5 shows the ophthalmic data analysis and its corresponding results.
- Chapter 6 ends the report with conclusion and propose other possible future research lines.

Chapter 2

Background information

In this chapter, we aim to provide the background information such as the concept of the used methods for the project in order to easily understand the later implementation chapter. Moreover, we also introduce the small dataset problem that we had to overcome in this project and the methods we used to deal with this kind of problems.

2.1 Linear models

Let E_{in} be in-sample error and E_{out} out of sample error, where in-sample error is the error rate we get on the same dataset we used to build our predictor, whilst out of Sample Error is the error rate we get on a new dataset. The learning process in a Machine Learning algorithm consists of finding the model such that $E_{out} \rightarrow 0$. However, we can not directly measure the E_{out}. But, we can measure the E_{in}. Thus, we can try to obtain $E_{out} \rightarrow 0$ by doing the following two steps: 1) $E_{in} \rightarrow 0$ and 2) $E_{in} \approx E_{out}$. So, it can be summarized to the below formula:

$$E_{in} \rightarrow 0, \quad E_{in} \leq E_{out} \leq E_{in} + O(\sqrt{\frac{C}{N}}),$$

where C is a notation for complexity, the more complex the model is, the higher the value C. N is the amount of data samples. Obviously, we aim to have a small number of $\sqrt{\frac{C}{N}}$ in order to reach the second condition $E_{in} \approx E_{out}$. Hence, an appropriate C along with the given amount of samples N is required.

Since our dataset is relatively a small dataset, a small value of C is needed to keep overfitting in check. That means, a fairly low-complexity model would be suitable for a small size dataset. For this reason, a linear model is suggested, which is a low complexity model that finds the relation between variables $X_i \in \{X_1, ..., X_N\}$ and the observation Y by the following formulation $Y = \sum_{i=1}^{N} w_i X_i + b$, where w_i, b are constants. Thus, Support Vector Machine is introduced in this section.

2.1.1 Support Vector Machine

Support vector machine (known as SVM) is a particularly powerful and flexible class of supervised algorithms for both classification and regression tasks. However, we will develop the project by using SVM in classification problems.

In this algorithm, a hyperplane in a high dimensional space splitting the space into two half-spaces by classifying them as 2 different classes is created. Elements in one of the half-space are positive values and the ones in the other half-space are negative values. Here, we call such hyperplane as classification boundary. Intuitively, a

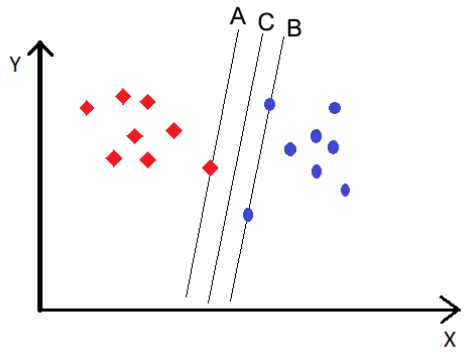

FIGURE 2.1: A SVM classification example

good separation is achieved by the classification boundary that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger is the margin, the lower is the generalization error of the classifier. The margin is calculated as the perpendicular distance from the boundary to only the closest points. So called Support Vectors are the vectors generated by a critical subset of data points, which are the nearest points to the boundary(see figure 2.1). If any of those points disappear the boundary changes, thus we can conclude that boundary only depends on the support vectors.

In fact, there are different types of hyperplanes which are Maximum Margin Hyperplane and Soft-margin Hyperplane. Maximum Margin Hyperplane is introduced by Bernhard E. Boser, 1992 and Soft-margin Hyperplane is given by Corinna Cortes, 1995. Maximum margin works only when data is completely linearly separable without any errors (noise or outliers). On the other hand Soft-margin was proposed to extend the idea of Maximum Margin in order to deal with non-linearly separable data. This is solved by introducing slack variables which penalize misclassifications. Since the given project data are non-separable, we will use Soft-margin Hyperplane. Hence, we give the mathematical formulation of linear Soft-margin below in order to effectively introduced its concepts.

Given a binary classification problem with a training dataset $D = \{(x_i, y_i)\}, i = 1, ..., N, y_i \in \{-1, 1\}$, where x_i are input variables and y_i are their corresponding class that has label either -1 or 1. N represents the number of total input data. The Soft-margin Hyperplane is formed by

$$\begin{aligned} \text{minimize} \quad & \frac{||w||_2^2}{2} + C\sum_{i=1}^{N}\zeta_i \\ \text{subject to} \quad & y_i(w^T \cdot x_i + b) \geq 1 - \zeta_i, \quad i = 1, ..., N \\ & \zeta_i \geq 0 \end{aligned}$$

where b is a constant, w is the coefficients vector, C is the trade-off parameter that roughly balances margin and misclassification rate. The higher the C is, the more influence of the misclassification error.

Furthermore, we present the advantages of SVM (*1.4. Support Vector Machines*) as follows:

- Effective in high dimensional spaces.

- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: can be extended to non-lineal models by means of kernel functions. Different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

2.2 Feature subset selection

Irrelevant features in the dataset do not contribute to the predictive accuracy. Moreover, redundant features do not help to getting a better predictor for that they provide mostly information which is already present in other features. Irrelevant features along with redundant features, severely affect the accuracy of the learning machines. Feature subset selection(FSS) can be viewed as an important process that identifies and removes as many irrelevant and redundant features as possible in order to reduce dimensionality and improve accuracy results, also to get an informative features subset (Evangeline et al., 2013).

As we mentioned on the previous section, we aim to identify the most relevant factors that are indicators of the predictions result. Thus, we expose a list of FSS methods that we used as a baseline method for the project's implementation.

2.2.1 Best Subset Selection methods

Best Subset Selection finds a feature subset of size S in which a classifier based on these S features has a lowest probability of error. The goal is to choose a subset X_S of the complete set of M input features $X = \{x_1, x_2, ..., x_M\}$ so that the subset X_S can predict the output y with accuracy comparable to the performance of the complete input set X, and with great reduction of the computational cost. The performance of a set of input features can be evaluated by Leave-One-Out cross validation (see *Cross-validation (statistics)*).

Using the same evaluation method, we give a brief explanation of the three different Best Subset Selection algorithms below.

Brute-force Feature Selection

The idea of Brute-force Feature Selection method is to exhaustively evaluate all possible combinations of the input features, and then find the best subset. Obviously, the computational cost of exhaustive search is extremely high, $(2^M - 1)$. Hence, people resort to greedy methods, such as Forward Selection. (*Feature selection*).

Forward Feature Selection

Forward Feature Selection begins by evaluating all single feature subsets, that is $\{\{x_1\}, \{x_2\}, ..., \{x_M\}\}$, where M is the input dimensionality. It can use LOOCV to measure the error of one-component feature subset one by one,$\{x_1\}, \{x_2\}, ..., \{x_M\}$, so that we can reach to find the best individual feature, for instance X_1.

Then, it finds the best pair features that consists of X_1 and the another individual feature in $\{x_2\}, ..., \{x_M\}$. After that, it finds the best three features, four and so on until the best subset feature X_S is found.

Backward Feature Selection

We have seen that Forward Feature Selection begins with one-component and then adds one by one until find the best subset X_s. Backward feature selection also known as Backward Elimination, it conversely builds a model with full features set, then it iteratively removes the least useful predictor, one-at-a-time till the best subset X_S is obtained. Here, the least useful predictor is the one that with which the model has the worst performance.

2.2.2 LASSO

Least Absolute Shrinkage and Selection Operator is known as LASSO, it was first formulated by Robert Tibshirani in 1996 (Tibshirani, 1996). LASSO mainly performs two main tasks for estimation in linear models, which are regularization and feature selection. The definition of LASSO is as follows.

Suppose the given data is $(X, y) = \{(x_i, y_i)\}, i = 1, ..., N$ where $x_i = \{(x_{i1}, x_{i2}, ..., x_{iM})\}$ are predictor variables and y_i are responses, M is the amount of features. We assume that the observations are independent, and the x_{ij} are standardized which means $\sum_i \frac{x_{ij}}{N} = 0, \sum_i \frac{x_{ij}^2}{N} = 1$. Letting $\widehat{\beta} = \{\widehat{\beta}_1, ..., \widehat{\beta_M}\}$, the LASSO estimate is defined as,

$$(\widehat{\alpha}, \widehat{\beta}) = \text{argmin} \sum_{i=1}^{N} (y_i - \alpha - \sum_j \beta_j x_{ij})^2$$
$$\text{subject to } \sum_j |\beta_j| \leq t$$

where $t \geq 0$ is a tuning parameter which controls the amount of shrinkage that is applied to the estimate. Furthermore, for all t, the solution for α is $\widehat{\alpha} = \overline{y}$. Recall $\overline{y}$ is the notation of the mean of y, which we can assume without loss of generality that $\overline{y} = 0$ and hence omit α.

Subsequently, the LASSO is solved in Lagrange form which has the below formulation,

$$\text{minimize}_\beta \sum_{i=1}^{N} (y_i - \alpha - \sum_j \beta_j x_{ij})^2 + \lambda ||\beta||_1$$

where $\lambda \geq 0$ is the parameter that controls the strength of the penalty, the larger the value of λ, the greater the amount of shrinkage. The exact relationship between t and λ is data dependent.

We can summarize the LASSO method as follows. It minimizes the residual sum of squares and sets a constraint on the sum of the absolute values of the model parameters to be less than a constant(t or in Lagrange form λ, which is an upper bound). In order to do so, the method applies a shrinking(regularization) process with which penalizes the coefficients of the regression variables shrinking some of them and sets others to zero. During feature selection process the variables that still have a non-zero coefficient after the shrinking process are selected to be part of the model. Thus, it retain the good features of both subset selection and ridge regression.

2.2.3 Relaxed-LASSO

Relaxed lasso is originally proposed by Meinshausen, 2006 with the aim to overcome the slow convergence of the LASSO in some sparse high dimensional data.

Moreover, it tries to reduce selecting noise variables in case the estimator is chosen by cross-validation. The idea is that Relaxed lasso use the LASSO to select the set of non-zero predictors, and then apply the LASSO again, but using only the selected predictors from the first step. The idea is to use cross-validation to estimate the initial penalty parameter for the LASSO. However it probably includes noise variable. For this reason, performing the LASSO again for a second penalty parameter applied to the selected set of predictors. Since the variables in the second step have less "competition" from noise features, cross-validation will tend to pick a smaller value for the penalization parameter λ, and hence their coefficients will be shrunken less than those in the initial estimate(Trevor Hastie, 2001).

2.3 Brief intro to statistic in classification problems

In this section, we briefly intro a Bayesian approach and a Frequentist inference with which we used to validate and support the project results.

2.3.1 Bayesian approach: Beta-binomial distribution

Bayesian methods solve the following problem, given a prior distribution $p(x)$ and a set of evidences E, compute a posterior distribution on x namely $p(x|E)$. Beta-binomial distribution is one of the conjugate families of distributions. The main property of conjugate families is that the posterior distribution follows the same parametric form as the prior distribution. Also, it is useful because of its closed-form solution for prediction. In other words, the family is closed under evidence-regardless of what we observe, we will continuously believe that the posterior lives in this family. Further, it facilities a huge simplicity in the computation of the posterior.

The density function of the prior Beta distribution $\theta \sim Beta(\alpha, \beta)$ is defined as,

$$f(x;\alpha,\beta) = \frac{t^{\alpha-1}\cdot(1-t)}{B(\alpha,\beta)}, \text{where } B(\alpha,\beta) = \frac{1}{(r+s-1)\cdot(\binom{r+s-2}{r-1})}, \alpha,\beta > 0, r, s \text{ integer}$$

In particular $Beta(1,1) = Unif(0,1)$ uniform distribution.

Binomial distribution is the probability distribution of n independent experiment that answer a yes-no question with an binary question results, and a probability p of success. $X \sim Bin(n,\theta)$ with n parameters and $p \in [0,1]$. The probability of getting k successes in n experiments has a density function like,

$$f(k,n,p) = \binom{n}{k} p^k (1-p)^{n-k}$$

As we mentioned before, the conjugate posterior will be again a Beta distribution.

The posterior distribution contains all the knowledge about the unknown quantity X. Therefore, we can use the posterior distribution to find point or interval estimates of X. One way to obtain a point estimate is to choose the value of x that maximizes the posterior probability density function. This is called the maximum a posteriori (MAP) estimation (*Introduction to probability, statistics, and random processes*).

In order to reflect the uncertainty of statistical results generated through the use of Bayesian statistical methods, the Credibility intervals (CI) can be used. Credibility intervals are used in Bayesian analysis to provide predictive indicators of the distribution of a given outcome (*Credibility Interval [online]*).

2.3.2 Frequentist statistic: Permutation test

Permutation test also called randomization test, is a type of significance tests which is a common statistical tool for constructing sampling distribution by resampling the observed data, that is to shuffle the observed data.It can be sorted as the following three steps(*Permutation and Randomization Tests*):

1. Compute some test statistic using the set of original observation.
2. Rearrange the observations in all possible orders, computing the test statistic each time. Recall that a test statistic is a statistic used in statistical hypothesis testing
3. Calculate the permutation test p-value, which is the proportion of test statistic values from the rearranged data that equal or exceed the value of the test statistic from the original data.

In spite of the fact that the computational cost of Permutation test is intense, but it would not be a problem with the current computers computational capabilities. Moreover, a big advantage of Permutation test is that it is independent to the model compared to many other statistical test methods.

Chapter 3

Proposal: "Iterative backward relaxed SVM selection"

In this chapter, we discuss the problems of the existing feature selection methods that we have already seen in the former chapter. Furthermore, we aim to give a proposal that is able to overcome those problems.

3.1 Problem of Relaxed-Lasso and Best Subset Selection

We aim to obtain the best subset features X_S with the size of S. That is, with a model built by X_{S+1} has roughly the same performance, and the model built by X_{S-1} has a significantly worse performance.

Remind that Relaxed-Lasso method simply applies LASSO twice which in the first time it initializes the penalization term, and the second time eliminates noises variables. Thus, we need to fix the two parameters at each step. That means, we have to tune the λ_1 which is the parameter that controls the strength of the penalty in the first time applying LASSO method, and λ_2 in the second LASSO as well. Thus, we would like to propose an method that can reduce it to only tune the penalty parameter once.

Now, we recall the three Best Subset Selection (BSS) methods, which are Brute-force, Forward and Backward Feature Selection. One of the problem of the BSS methods is that they are greedy methods with respect to performance. Moreover, Brute-force Feature Selection (BFS) consists of building models by using all combination of features which has a unacceptable high computational cost. Even though the Forward and Backward Feature Selection have a computation cost lower than BFS, but they are still computationally expensive. An example of Backward Feature Selection is given in the following, if we want to get the best features subset X_S of size S from the complete features set of size M. Then, it has to build $M + (M-1) + (M-2) + ... + (S+1) + S = \sum_{i=S}^{M} i = \frac{(M+S)\cdot(M-S+1)}{2}$ different models. Note that these models have $M - k$ features in the step k.

Based on the review of the problem of the above methods, we aim to propose a hybrid method of the Relaxed-Lasso and BSS methods that solves the above mentioned problems. The already existent method Improved Variable Selection With Forward-LASSO Adaptive Shrinkage is also a hybrid method of LASSO and Forward Selection (Peter Radchenko, 2011). However, the Relaxed-Lasso method shrinks variable so that we can remove features. Hence, a Backward-LASSO method that shrinks variable and backwardly removes feature one-at-time can be logically considered. Thus, we introduce the Iterative backward relaxed SVM selection method in the next section.

3.2 Iterative backward relaxed SVM selection

In this section, we propose a method that share the same idea as Backward Feature Selection and Relaxed-LASSO method, however it tries to improve them by reducing computational cost and also getting a more relevant feature subset. In order to achieve the above goal, we need to firstly introduce $L1$-norm SVM on below.

3.2.1 $L1$-norm SVM

We have already seen in the former chapter about the standard formulation of Support Vector Machine (SVM) which uses $L2$-norm in the regularization term and hinge loss in the loss function term. We recall the formulation of SVM below:

$$\begin{aligned} \text{minimize} \quad & \frac{||w||_2^2}{2} + C\sum_{i=1}^{N}\zeta_i \\ \text{subject to} \quad & y_i(w^T \cdot x_i + b) \geq 1 - \zeta_i, \quad i = 1, ..., N \\ & \zeta_i \geq 0 \end{aligned}$$

where $||w||_2^2$ is the $L2$-norm penalty (also called *ridge* penalty) which corresponds to the regularization term that imposes a penalty on the complexity of lineal model $f(x) = w^T \cdot x + b$. The loss function is the hinge loss which is $\sum_{i=1}^{N}\zeta_i$ such that $\zeta_i \geq 1 - y_i(w^T \cdot x_i + b) \geq 0$.

The $L1$-norm is also known as *LASSO* penalty which was firstly proposed by P. S. Bradley, 1998 to use $L1$-norm SVM for feature selection as consequence of the resulting sparse solutions (Hai Thanh Nguyen, 2011):

$$\begin{aligned} \text{minimize} \quad & \lambda||w||_1 + \sum_{i=1}^{N}\zeta_i \\ \text{subject to} \quad & y_i(w^T \cdot x_i + b) \geq 1 - \zeta_i, \quad i = 1, ..., N \\ & \zeta_i \geq 0 \end{aligned}$$

where $\lambda = \frac{1}{2C}$, $||w||_1 = \sum_{j=1}^{M}|w_j|$ is the $L1$-norm of w and M is the total number of features. An important property of $L1$-norm penalty is that making the λ sufficiently large will shrink some of the fitted coefficients toward zero. Thus, the $L1$-norm penalty performs a kind of continuous variable selection as long as λ varies, which is not the case of $L2$-norm penalty (Li Wang and Zou, 2006).

3.2.2 Iterative backward relaxed SVM selection

As we mentioned before, Backward Feature Selection needs to select the best subset by comparing $d = \frac{(M+S)\cdot(M-S+1)}{2}$ models. Recall that M is the amount of the complete feature set X_M and we aim to reduce it until the best subset X_S of size S is obtained. Furthermore, d is getting bigger as long as M increases. The idea of the proposal method is to use a Relaxed-LASSO-like methodology along with an iterative backward elimination approach to reach to obtain the best features subset with at most $M - S + 1$ of computational cost, which has a huge enhancement compared to Backward Features Selection in case of high dimensionality. The detail process are as follows.

Input: a list of ordered C's ($C = \frac{1}{2\lambda}$) with a size of p: [C_1,C_2,...,C_p], where C_1<C_2<...<C_p

. **Output:** A subset of the best selected features: X_s

while *the best features subset X_s is not found* **do**

    1) Do grid search over the C list, that is to build p different L1-norm SVM models which each model is created by a different C in the list of C.

    2) In each model, show its indicator features which is the features that help the prediction.

    3) Remove the least important feature which is the one that contributes the last over all models. If there are more than one feature that appears the last, then we do a refined grid search of C, which is to define a list of sorted C's with reduced range. For example [C_1,C_11,C_12,...,C_5] where C_1< C_ij< C_5. Then we remove the least important feature based on the result of refined grid search.

    4) Check whether the remaining features has size S which is the best subset X_s that we aim to find. If the answer is affirmative, the program ends here. Otherwise, we go to next step, so run again from the process 1) to 4).

end

Algorithm 1: Iterative backward relaxed SVM selection

The pseucode as shown in the Algorithm 1 shows the full process in 1 step of backward elimination. In each step, it eliminates the least helpful feature, and repeats the steps until the X_s is obtained. To make it clear, in the process step 1), the grid search of C is simply to fit a list of different values of C to the model. However, in this process we aim to do feature selection that identify the most relevant feature. Hence, we remove the feature that the least less participates the prediction process over the models of different C's. Since we iteratively remove one feature in one step, thus the earliest removed feature is considered as the least helpful feature, and the most recent removed one should be a more helpful feature than the earlier removed ones. With this, we have the best subset features X_S along with an importance-sorted list of the rest features in X_S^C, and its computational cost is $(M - S)$. Note that X_S^C is a complementary set of X_S in the complete feature set X_M.

Chapter 4

Lake Louise Score analysis

In this chapter, we describe the first analysis of altitude sickness which is to predict Lake Louise Score by the given physiological data and stage-wise information. Firstly, the definition of Lake Louise Score System will be given. After that, the detail of data will be explained. Then, we will discuss the implementation. Next, the obtained results will be shown. Lastly, we will validate results in order to supports their certainty.

4.1 Lake Louise Score System

Acute mountain sickness (AMS) is the acute altitude illness that typically occurs in unacclimatized persons ascending to altitudes over 2500 m. However, it can also develop at lower altitudes in highly susceptible individuals. The Lake Louise Scoring System (LLSS) uses an assessment questionnaire and a scorecard to evaluate adults for symptoms of AMS. The questions include asking the involved person whether they have the following 5 symptoms: headache, gastrointestinal symptoms, fatigue and/or weakness, dizziness and/or lightheadedness, difficulty sleeping. Moreover, they need to give a punctuation between 0 to 3 that significantly explains how they are affected by AMS. Regarding the punctuation:

- 0: Non affected
- 1: Mild affected
- 2: Moderate affected
- 3: Severe affected

The sum of 5 scores is total score. A total score under 3 represents not affected by AMS, a total score from 3 to 5 indicates mild AMS and a score of 6 or more signifies severe AMS.

4.2 Data description

In this analysis, the data are stored in an Excel file named *TFM_Sherpa.xlsx* with 3 different sheets: Sheet1, Sheet2 and Sheet3. Sheet1 contains all variables(physiological data, stage-wise information, etc) with their corresponding values. Sheet2 stores the personal information of each individual, however we anonymized them by removing their name and surname in order to protect their privacy. Sheet3 provides the meta-data that describe each feature one-by-one. The given data contains 56 different individuals information with 106 variables. The individuals are classified in 3 different groups: 11 trekkers, 17 climbers, 28 sherpas. While trekkers are those who

only participated from Luckla(2860 meters of altitude) to Base camp(5164 meters), climbers ascended till the highest point of Everest(8848 m) with companions of sherpas. However, the Lake Louise Score data is collected from Luckla to Base Camp due to the undertraining of the scientific team, who were not prepared to ascent to 8848 m. Thus, we only do this analysis by using trekkers data. Nevertheless, there are 8 of climbers who began from Luckla with trekkers so they share 2 different roles: trekkers and climbers. Hence, their LLS data are also collected so that we can include to the analysis dataset. In addition, the data of 4 climbers were only measured in few stages. For the reason of too much missing data, we decided to remove them from the dataset. Therefore, we have originally 11 trekkers data including 4 climbers data afterwards, now the analysis dataset contains 15 different individual stage-wise information.

Beside the individual information, we are also given an Excel file about stage-wise information named *Stage-wise_information.xlsx*. Basically, it provides us the following 6 useful information of each stage: the place of the beginning of the stage, the place where the stage ends, used time (unit: day), the total distance(kilometre), the accumulated ascending meters, the accumulated descending meters. Along with the previously described individual information, now we have in total 112 variables.

In this study, physicians are concerned with the impact of cardiovascular measurements as well as the importance of the stage profile in the assessment of the LLS value. This reduces the considered variables to the following eighteen features:

1. ID: an personal ID for each individual with no repetition.
2. altitude: altitude of the place where the sample was collected, we can also consider them as a stage information. For instance, Barcelona with 20m of altitude can be considered as a stage, because they collected the individual data there. In total, they collected samples in 10 different places with different altitude, which began from Barcelona and ended Barcelona as well. Orderly, they are Barcelona(20m, before going), Luckla(2860m), Monjo(2835m), Namche(3450m), Tengboche(3867m), Pangboche(3965m), Dingboche(4380m), Lobuche(4930m), Gorakshep-EBC(5164m), Barcelona(20, return).
3. Gender: individual's gender. The value are *M* and *F* which respectively represent masculine and feminine.
4. up: total amount of meters ascended in one stage.
5. down: total amount of meters descended in one stage.
6. distance(km): total accumulated distance(kilometres) completed until the current stage.
7. Age: individual's age. The participants have age between 22 to 58.
8. Weight: individual's weight with kilogram(kg) as unit. This variable has values between 47.0kg to 87.0kg.
9. Height: individual's height with centimetre(cm) as unit. The values range from 155cm to 186cm.
10. TAS: systolic Blood Pressure, it is a part of arterial blood pressure. Normal blood pressure is systolic of less than 120 mm Hg. Moreover, blood pressure

is one of the vital signs, along with respiratory rate, heart rate, oxygen saturation, and body temperature (*Vital Signs (Body Temperature, Pulse Rate, Respiration Rate, Blood Pressure)*).

11. TAD: diastolic blood Pressure. Normal blood pressure is diastolic of less than 80 mm Hg.
12. FC: heart rate. The normal pulse for healthy adults ranges from 60 to 100 beats per minute.
13. SO: resting pulse oximetry, it is commonly known as oxygen saturation (SpO_2). Normal pulse oximeter readings usually range from 95 to 100 percent. Values under 90 percent are considered low (*Symptoms Hypoxemia*).
14. HD: headache. It is one of the LLS assessment that has an integer value from 0 to 3. Which 0 represents non affected, 3 means severe affected.
15. GS: gastrointestinal symptoms. The LLS assessment.
16. FN: Fatigue and/or weakness. The LLS assessment.
17. DZ: Dizziness and/or lightheadedness. The LLS assessment.
18. DSL: Difficulty sleeping. The LLS assessment.

In addition, we give an example of only one individual with its data measured along with 10 sorted stages. The exampled individual is the one that has ID *SH01*.

	ID	altitude	Gender	up	down	Age	Weight	Height	TAS	TAD	FC	SO	HD	GS	FN	DZ	DSL	distance(km)
0	SH01	20	M	0.0	0.0	46	62.0	167.0	120.0	72.0	65.0	97.0	NaN	NaN	NaN	NaN	NaN	0.0
1	SH01	2860	M	0.0	0.0	46	NaN	167.0	114.0	78.0	64.0	96.0	0.0	0.0	0.0	0.0	0.0	0.0
2	SH01	2835	M	574.0	604.0	46	NaN	167.0	128.0	84.0	65.0	97.0	0.0	0.0	0.0	0.0	0.0	12.1
3	SH01	3450	M	999.0	208.0	46	NaN	167.0	126.0	NaN	72.0	94.0	0.0	1.0	0.0	0.0	0.0	17.6
4	SH01	3867	M	879.0	629.0	46	NaN	167.0	124.0	95.0	80.0	94.0	0.0	0.0	0.0	1.0	1.0	26.1
5	SH01	3985	M	270.0	143.0	46	NaN	167.0	126.0	95.0	67.0	93.0	0.0	0.0	0.0	0.0	0.0	29.7
6	SH01	4380	M	410.0	75.0	46	NaN	167.0	123.0	91.0	82.0	93.0	0.0	0.0	0.0	0.0	0.0	36.9
7	SH01	4930	M	752.0	149.0	46	62.0	167.0	130.0	89.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	48.1
8	SH01	5164	M	326.0	86.0	46	62.0	167.0	136.0	96.0	77.0	92.0	0.0	0.0	1.0	1.0	0.0	52.4
9	SH01	20	M	0.0	0.0	46	NaN	167.0	110.0	79.0	70.0	98.0	NaN	NaN	NaN	NaN	NaN	0.0

FIGURE 4.1: An explanatory data description of the individual with ID: SH01

Note that the *NaN* values represent missing data.

4.3 Problem modelling and implementation

In this section, we discuss how did we model the problem and what is the exact implementation. We follow the general routine which is to begin with cleaning data. Then, based on the feedback of the scientist team, we did the feature engineering process. Afterwards, we started feature selection in order to sort the importance of features. The last step, we iterative add the more important feature to build a model and use it to make prediction. Then, the model that gets the best result is taken, so it is considered as the definitive model that we use to predict LLS.

4.3.1 Data cleaning

As the title says, this subsection we describe the necessary processes that we did in order to obtain a clean dataset. Before we begin with the cleaning process, we merged together the two Excel files that contain data. Thus, the merged dataset contains both individual information and the stage-wise information. Recall that we just do analysis of the 15 trekkers data.

The first problem we have to deal with is that the given data contain lots of missing data. We consider each feature as a vector of size N, where N is the number of samples. Now, we remove all those features that have missing data in whole vector or the most content in the vector are missing (NaN values). For example, the feature *Forced Vital Capacity teorica* written as *FVC.t*, it has a vector of the individual with ID *SH01* as $(4.32, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN)$. It happens the same with other individuals. Since it does not provide any significant information, thus we remove all the feature like this.

After removing the features that contain missing data such as the above example, we have 18 remained features as the former section shown. However, for those feature vectors that also contain missing data but a few, we do not remove them. What we did in this case is to fill up the missing data with either 0 or the mean value group by the individual with the same ID. We carefully fill up the missing data based on its context. For instance, in Figure 4.1 the 5 LLS parameters collected in the place with altitude of 20m are NaN values, we know that the place with 20m of altitude must be Barcelona which it is not considered as high altitude. Thus, we intuitively think the well-trained healthy individual in Barcelona should report 0 as non affected for all 5 LLS questions. Hence, in this context we fill up them with the value 0. However, the 5 LLS parameters in the altitude of 4930m are also missing. Since 4930m is a relatively high altitude, hence we separately fill up the mean value group by the same ID for each LLS parameter. Despite the parameter *Weight* also contains a lot of NaNs, but the human weight does not really vary too much in few days. Furthermore, the individual with ID $SH01$ has the same weight in the altitude of 20m and 4930m. For this reason, we also fill the missing data of the *Weight* parameter with mean value of the same individual. Based on the same idea as the above example shown, we filled up the mean value group by individuals in all of the following parameters: *Weight*, *TAS*, *TAD*, *FC*, *SO*, *HD*, *GS*, *FN*, *DZ*, *DSL*.

So far, we have explained how did we deal with missing data in a feature vector, which is vertically shown in the table(see Figure 4.1). Now, we describe how to deal with missing entities. An entity is a row in the data table, so here it is horizontally shown. Also, in this project, an entity can also be understand as a row that contain one determined stage of one individual. Hence, as we mentioned previously, there are totally 15 trekkers and 10 stages, it should contain 150 entities. However, there a few individuals were not measured in some stages. For example, the individual with ID *SH11* is a female who had 69% of *SO* in the stage of altitude 3985m which was in a very dangerous situation, so she was evacuated by helicopter. Thus, we remove her last few stages entities due to she did not participate these stages. Furthermore, considering that healthy people in Barcelona do not have any altitude sickness, also Barcelona is not considered as a high altitude. For this reason, we also remove the entities where they were in Barcelona. The remaining entities are 116.

Then, we convert the values of the parameter *Gender* to a binary value, 1 for male and 0 for female. We do this due to the classifiers can only recognize numerical input rather than letters. Furthermore, we think that is obvious altitude sickness depends

on altitude. So, we subsequently remove the *altitude* feature. With this, we end the data cleaning process.

4.3.2 Feature engineering

Feature engineering is the process of using domain knowledge of the data to create features in order to make machine learning algorithms work.

In this firstly LLS analysis, we have parameters such as *ID*, *Weight*, *Height* that all of them represent information about the same individual. In addition, *ID* is just an virtual parameter that we randomly created in order to distinguish individuals, but it cannot provide any useful information for LLS prediction. Hence, we aim to create a feature that can distinguishably represent an unique individual with his/her own value. That occurs the Body Mass Index appear. Body Mass Index (BMI) is a measurement of a person's weight with respect to his/her height. It is more of an indicator than a direct measurement of a person's total body fat (*What is Body Mass Index (BMI)?*). BMI is calculate by the following formula:

$$BMI = \frac{Weight(kg)}{Height^2(m^2)}$$

The normal weight status should have BMI in the range from 18.5 to 24.9. The BMI of the 15 trekkers in the dataset range from 19.01 to 27.44.

Once we created the *BMI* feature, in order to not duplicate the individual information, we remove the features *ID*, *Weight*, *Height*.

Another parameter we created is *lls*, which is the sum of all of 5 LLS parameter's values. Rather than call it as a feature, we call it label which is the parameter that we aim to predict. Recall that the sum of 5 LLS parameters, under 3 means non affected, between 3 to 5 represents mild affected and over 6 indicates severe affected. Due to the fact that we have very small dataset, and few entities have a *lls* over 6. Thus, we decide to transform the *lls* values into a binary value. That is, the sum of 5 LLS parameters under 3 goes to 0, which means non affected, and the sum over 3 will be 1 which indicates the individual was somehow affected by high altitude. After this, we remove the features *HD*, *GS*, *FN*, *DZ*, *DSL*.

After cleaning the dataset and do feature engineering, we still have 10 features remained which are: *BMI*, *Gender*, *up*, *down*, *distance(km)*, *Age*, *TAS*, *TAD*, *FC*, *SO*. In addition, the label parameter is *lls*.

4.3.3 Feture selection

As we mentioned before, our goal is not only be able to predict LLS, but also to find the factors that are indicators of the LLS prediction in order to better understand the cause of altitude sickness. Hence, in this section, we aim to select a subset of features which are the most relevant features so that we can use them to build a more powerful predictive model.

Before selecting a subset of features, we want to know the importance of each feature and its order of importance with respect to the other features. Thus, we use the previously mentioned Iterative backward relaxed SVM selection method to sort the full feature set X_{10}. That is, we apply $L1$-norm SVM in each step so that it shrinks the least important feature's coefficient toward 0. The idea is, from the full feature set X_{10} to iteratively remove one feature at each step until the 1-featured subset X_1 is obtained. Hence, we consider that the earliest removed feature is the least relevant to the LLS prediction, and the later removed features are more relevant than the earlier

removed ones. With this, we can obtain an importance-ordered list of features sorted by the order that it was eliminated during the feature selection process.

However, we also need to discuss the problem we encountered during the process. Recall the pipeline, build a list of models based on the list of different C's (the penalty parameter in SVM). Then we aim to remove the feature that appears the least in the prediction over all models. However, at some step, there were more than 1 feature that are the least appeared at the same time (see Figure 4.2).

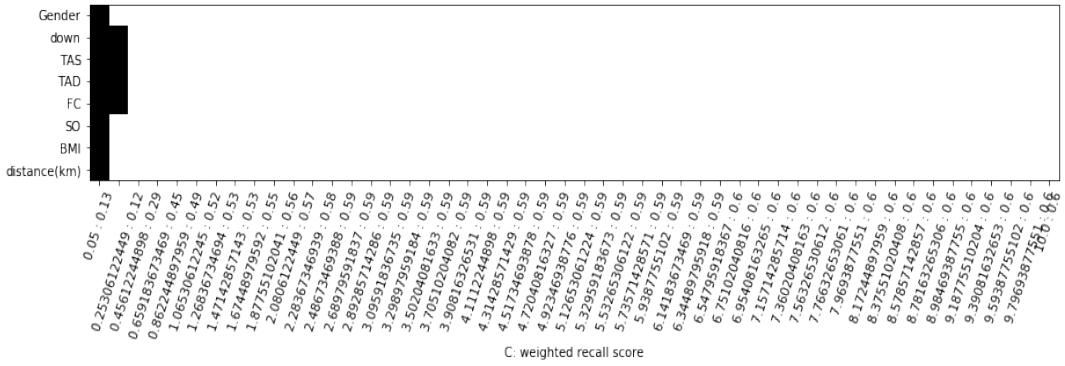

FIGURE 4.2: An example of the problem in feature selection.

The values in X-axis are the recall score of the prediction results obtained by the 50 different SVM models fitted with 50 different C's. Y-axis shows the feature name. The black colour cell reports that the feature was not used for the prediction of the respective model in the X-axis. In the same sense, white colour cell means that the feature is one of the indicator of the corresponding model with the C in the X-axis. For example, the first column cells at the figure 4.2 are black, which its X-axis is 0.05 : 0.13. That means, in the $L1$-norm SVM model with $C = 0.05$ only have 13% of prediction recall, and all of the 8 features in the Y-axis did not participate the prediction in this model. That is, the coefficients of those 8 features are 0, and so it is black colour. Now the problem is, the rest 4 features begin the prediction from the second C in the X-axis, but *down*, *TAS*, *TAD*, *FC* remain. In this case, what we did is to do a refined grid search of C. That is, to do a grid search of a smaller range of C. Then, the refined grid search result shows that *TAS* is the least helpful feature. Thus, we will remove *TAS* at this step. (see Figure 4.3).

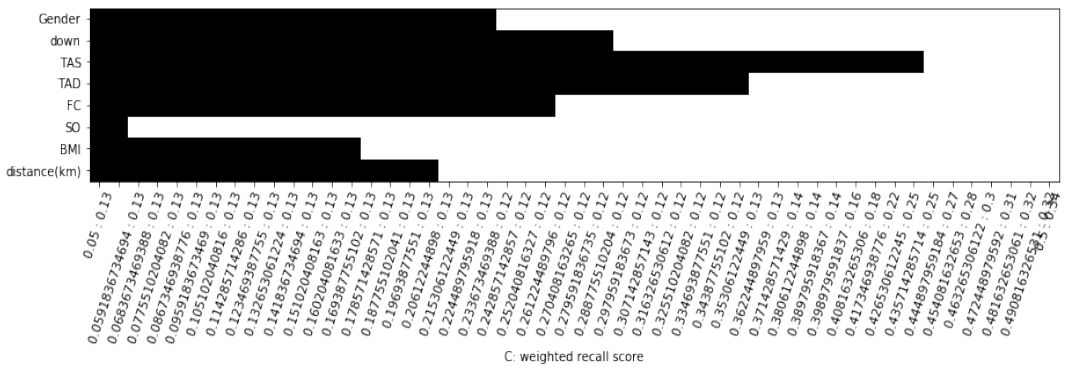

FIGURE 4.3: Refined grid search of a smaller range of C.

The above process results an importance-ordered list of features that are sorted from the most important to the least important for the LLS prediction, which is shown as follows:

$1)SO \quad 2)BMI \quad 3)distance(km) \quad 4)Gender \quad 5)FC \quad 6)down \quad 7)TAD \quad 8)TAS$
$9)Age \quad 10)up$

We have sorted the importance of features so far. However, we have not chosen the subset of feature that we will use to build the predictive model yet. Since we only have 116 entities with 10 features, we can also consider it as a matrix of $116x10$ which is small so we would not have computational cost problem. Furthermore, due to the tiny dataset, we should more carefully select features to ensure that the chosen subset is the most helpful one. Thus, after getting the importance-ordered list of features, we forwardly build predictive model by adding one feature at time beyond its importance order. So, we will have 10 different models. That is, we first build a simple 1-feature $L1$-SVM predictive model, which the used feature is the most relevant one that we had obtained before: *SO*. Then we use the built predictive model to predict LLS and we record its recall score. After this step, we create a 2-features model that contains the most important 2 features: *SO*, *BMI*. We also record its performance. Repeat the same process until we have all 10 models built. Their performance is shown in Figure 4.4 and 4.5.

Number of used features	Best macro recall	The corresponding micro recall	mean of 2 recall scores
1	0.621782	0.637931	0.629857
2	0.765017	0.689655	0.727336
3	0.694719	0.715517	0.705118
4	0.751485	0.715517	0.733501
5	0.737954	0.741379	0.739667
6	0.763696	0.637931	0.700814
7	0.747525	0.560345	0.653935
8	0.762376	0.586207	0.674292
9	0.762376	0.586207	0.674292
10	0.738944	0.594828	0.666886

FIGURE 4.4: The 10 models performance details

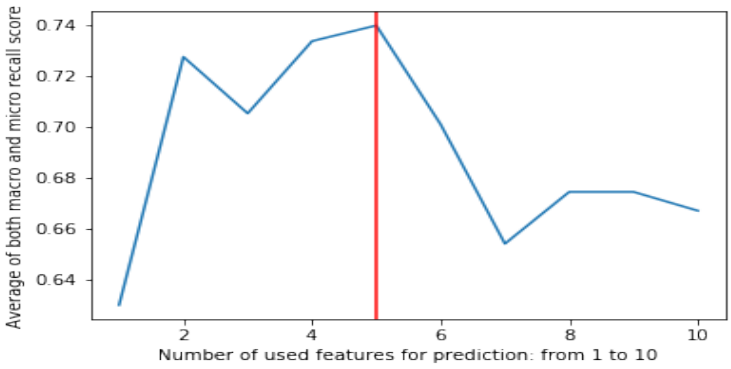

FIGURE 4.5: Plot of the 10 models performance.

In Figure 4.4, it shows the 10 differents model with their performance represented by recall score. Each row represents 1 exact model. The first column shows the number of features used for building model, the second column is the best macro recall scores obtained, the third columns shows the micro recall and the last columns is the average of the previous two recall scores. Figure 4.5 uses the first columns of the previous figure as its X-axis, and the last column for its Y-axis. That is to plot the 10 different model with its corresponding mean recall scores.

Based on the plots of the 10 different models performance. We choose the one that uses the first 5 most important features. Because it has the highest mean of 2 recall scores. Remember that recall is also called sensitivity, which measures the proportion of actual positives that are correctly identified. In other word, it measures that the data in the class -1 are indeed be classified as -1 and the same to the class 1. Now, the difference between micro recall and macro recall is that macro measures the average of the accuracy of recall between 2 classes, whilst the micro is weighted measurement. An example of micro is, we have in total 116 entities which 101 of them belong to the class -1 and 15 belong to 1. Thus, 100% of probability that correctly classifies the entities of class -1 which is $\frac{101}{116} = 87\%$ of accuracy with respect to the entire dataset. This is completely different compared to the class 1 ($\frac{15}{116} = 13\%$). Thus, as micro is a weighted measurement that takes account into the number of sample, it values more to the group with a major number of sample. However, the small group of 15 entities is the class 1 which is the class that people reported they were affected by high altitude ($lls >= 3$). For this reason, we aim to be sensitive to this small group. So, only looking at micro recall is not enough, that is why we consider that we should focus on macro recall, but without ignoring the micro recall. Thus, we compute the average of macro and micro recall, and take the model with the highest mean recall scores.

4.3.4 Train a classifier for LLS prediction

In this last process, we aim to build models with the chosen subset of feature to make prediction about LLS. The chosen feature subset is (*SO*, *BMI*, *distance(km)*, *Gender*, *FC*). As we commented in the Chapter 2, it is appropriate to use models with low complexity due to the small size of dataset. In addition, one of the simple model is SVM that we have used so far. Here, instead of only build SVM model we also create a Random Forest model. A brief description of Random Forest is that it is a supervised learning algorithm that can be used for both classification and regression problem. It creates a forest and make it random. The forest is actually a ensemble of Decision Trees. To summarize, Random forest builds multiple Decision Trees and merges them together to get a more accurate and stable prediction. It is one of the most used algorithm due to its simplicity.

The programming language we use for this project is Python. Here, we create the predictive models by using $scikit - learn$, which is a powerful library of Python for Machine Learning uses. It has both SVM and Random Forest algorithm created. We just need to fit the feature set, label and other parameter to train a desired classifier. The pipeline of this prediction process is below:

1. Given a list of class weights and a list of C's. We build the SVM models with a grid search of class weights and C's in order to find the best performance model, the highest mean recall. Class weight is a parameter that we can fit into the SVM algorithm, it is actually a weight of misclassification rate. We use the same example as the previous subsection, the small size class 1 has only 15

entities, hence one misclassification in this class should weight more than the another class. Thus, we set the class weight of the class 1 as 1, and do a grid search of a list that ranges from 0.01 to 1 for the class -1 in order to somehow balance the misclassification rate.

2. Given a list of class weights and a list of *max_depth*. *max_depth* is a parameter of Random Forest classifier which means the maximum depth of the tree. If its value is None, then nodes are expanded until all leaves are pure. Now, we also do a grid search of Random Forest classifiers that are fitted with the given lists so that we can find the best performed one.

3. We compare between the found best SVM model and Random Forest model. Then, we will choose the one that has highest average between macro and micro recall scores. The chosen model is a definitive model that we will store it and use it for the LLS prediction of new data.

4.4 Results

As we described in the previously implementation section, we have done grid search for finding the best performed SVM model and Random Forest model. The found parameters of those 2 different models are the following.

- SVM: the class weight of the class -1 (the stage where people did not suffered from high altitude) is 0.14655172413793105, the class weight of the class 1 is 1, the C penalty value is 1.3879310344827587.
- Random Forest: the class weight of the class -1 is 0.01, the another class weight is 1, the maximum depth is 3.

Also, we show the performance of both model in Figure 4.6 and 4.7

Macro recall: 0.73795379538
Accuracy score: 0.741379310345

	precision	recall	f1-score	support
-1	0.95	0.74	0.83	101
1	0.30	0.73	0.42	15
avg / total	0.87	0.74	0.78	116

FIGURE 4.6: SVM performance

Macro recall: 0.690759075908
Accuracy score: 0.560344827586

	precision	recall	f1-score	support
-1	0.96	0.51	0.67	101
1	0.21	0.87	0.34	15
avg / total	0.87	0.56	0.63	116

FIGURE 4.7: Random Forest performance

In Figure 4.6 and 4.7, what they show at the first row is the obtained macro recall score. Then, the second one is the accuracy score. Below, the confusion matrix is shown, it tells us what is the precision, recall, f1-score and support of each class. Also, it shows on below the micro total, which is the weighted average values of all the above 4 measurements.

Now, the SVM model has a 0.738 of its macro recall and 0.74 of micro recall, whereas the Random Forest has 0.691 of macro recall and 0.56 of micro recall. Obviously, SVM seems to have higher recall score. Nevertheless, we compute the mean recall of the SVM model which is round of 0.739, and the mean recall of the Random

Forest model is 0.626. We compare these 2 mean recall, the SVM model beats the Random Forest one again. We can also see that the area under the ROC-curve of the SVM model is 0.74 which is also bigger the Random Forest one: 0.69. (Figure 4.8). The plot closer to the left-upper corner, the bigger is the area, thus the better is the result.

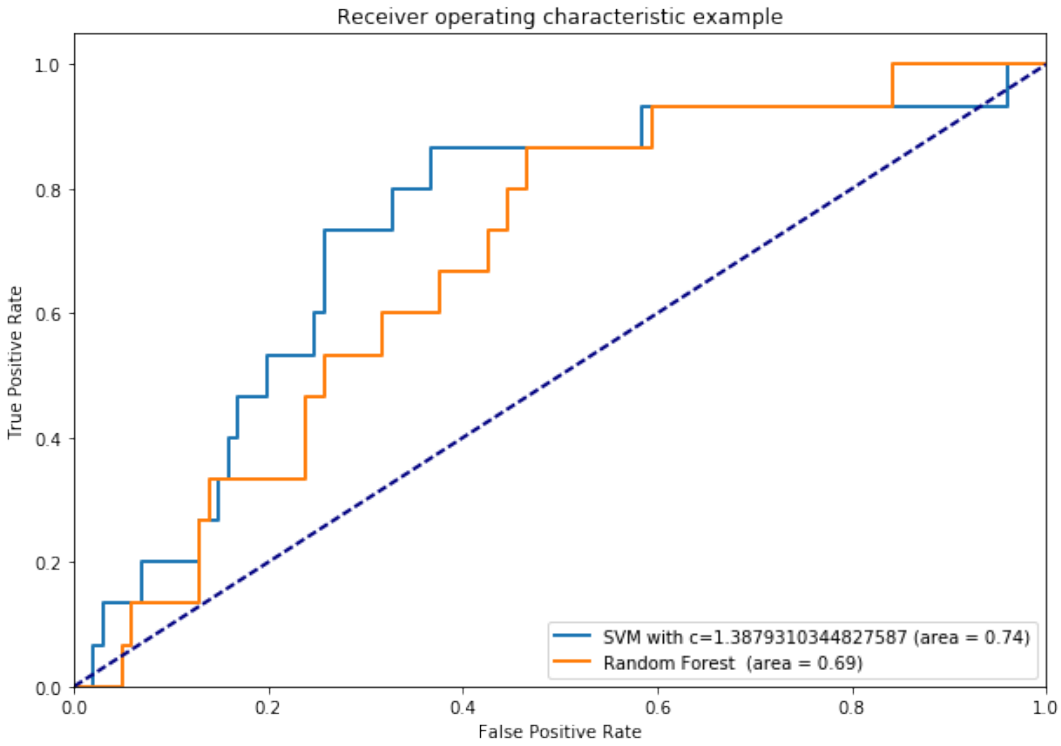

FIGURE 4.8: ROC curves and area under the curve

Thus, we keep the SVM model with the mentioned parameters setting for the LLS prediction, with which we can ensure that it more likely has around 0.74 of recall score.

4.5 Statistical validation of the results

4.5.1 Permutation test

In this last section, we would like to give support to the found results in the previous section. Thus, we did a permutation test of 500 samples. This is, we randomly shuffle the values of label *lls* repeating 500 times. Then we use the same setting model as the we used for prediction, but with the shuffled label to train the classifier. Then we test its performance. Thus, we considered that the prediction result by shuffled label are randomly generated and it should have bad result. For this reason, we give the support of the found result along with Figure 4.9 and 4.10. We have seen in the previous section that the area under the ROC curve is 0.74 which is shown as the red line in Figure 4.9, we can also observe that mostly of the randomly generated sample has an area around 0.50 and other range. Thus, it is validated that random generated models just have a 0.044 of probability that get a better area than our model. With the same reasoning, the mean of macro and micro recall has a 0.21 of p-value, which is reasonable.

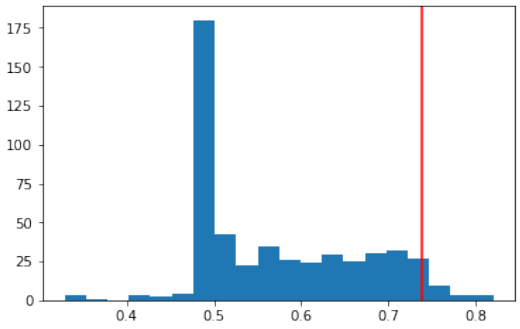

FIGURE 4.9: Permutation test: area under the ROC-curve

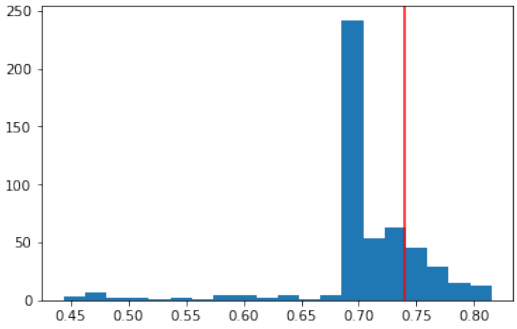

FIGURE 4.10: Permutation test: mean of macro and micro recall

4.5.2 Beta-binomial distribution support

The last validation of result we did is to follow the Bayesian approach. We validate the certainty of the recall for each of the class -1 and 1 by plotting their Beta-binomial distribution. The results are shown below (Figure 4.11 and 4.12). The peak closer to the right side of X-axis, the better is the results.

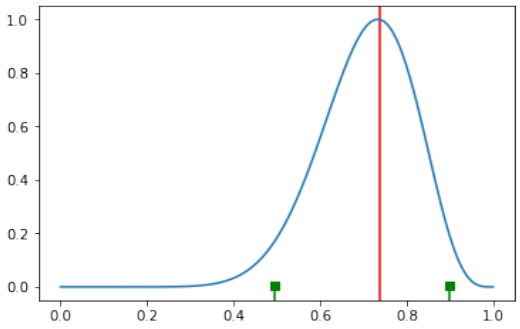

FIGURE 4.11: Beta-binomial: the class -1

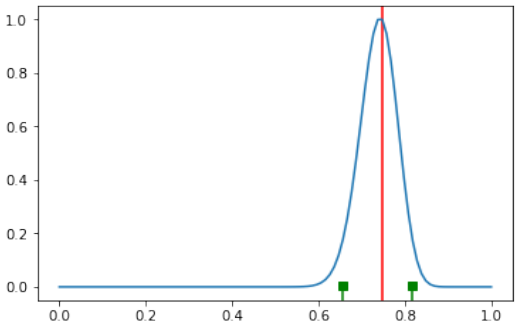

FIGURE 4.12: Beta-binomial: the class 1

The red line shows the mode of the corresponding distribution. Figure 4.11 is the recall distribution of the class -1, which has a mode 0.73737374 and the another figure is the distribution of the class 1 with a mode of 0.74747475. However, the peak of the class 1 seems more narrow than the one of the class -1. Thus, the class 1 has a wider range of confidence interval(CI) than the class -1, which is shown as the green colour marker in the the Figure 4.11 and 4.12. The CI of the class 1 is $[0.495, 0.899]$, whereas the CI of the class -1 is $[0.657, 0.818]$.

Chapter 5

Ophthalmic analysis

In this chapter, we aim to study the high altitude sickness by analysing ophthalmic data. Due to the study from ophthalmic data is an uncommon and quite new method, instead of using the existing measurements such as LLS that we used in the previous analysis, here we carefully and innovatively define measurements that could somehow explain how people were suffered from high altitude by their ophthalmic data.

5.1 Data description

One of the given data files is the same as the one that we had already seen in the previous analysis, which is *TFM_Sherpa.xlsx*. However, due to the other given data files just contain 12 trekkers ophthalmic data, thus we will just analyse these 12 trekkers data.

One of the given Excel files named *visual_acuity.xls*, stores the visual acuity parameter of the trekkers collected before the expedition (in Barcelona) and the return (Barcelona), also 1 month after the return (Barcelona) in order to check their recoveries. To simply notation, the data collected period are written as $V1$, $V2$ and $V3$ where $V1$ represents before the expedition, $V2$ for the return and $V3$ for 1 month after the return. This file contains visual acuity of both right and left eyers, which are integer numbers that ranges from 60 to 99.

There are also another data file named *macula_cfnr.xlsx* which its first sheet contains choroidal macula measurements of both eyes in 3 different stages ($V1$, $V2$ and $V3$), the second sheet stores the information of layer of nerve fibers of the retina(CFNR) which measures the thickness in microns of the optical disk. In order to understand well, we show two explicative scheme about the choroidal macula and CFNR in the figure 5.1 and 5.2 respectively.

The parameters in the above file are *central*, *average*, *volume* which indicates the position where the data were collected for choroidal macula measurements (see Figure 5.1). Moreover, it also contains *total*, *superior*, *inferior*, *temporal* and *nasal* that represent the position where the data were collected for CFNR. Now, *superior* is the superior part that consists of *temporal superior* and *nasal superior* in the figure 5.2. The same sense can explain the *inferior*, *temporal* and *nasal*. However, *total* is the total CFNR that it is composed of all of the other four position.

In addition, we had a meeting with the ophthalmologists of the scientist team. They suggested us to separately analyse 4 quadrants in the figure 5.2, one of reason is because of duplicate data. Also, it is better to know which of these 4 quadrants is the indicator of prediction. Thus, an another file that contains the separate measurements of 4 quadrants for CFNR study is given, and the 4 parameters are named: *Temporal Superior*, *Temporal Inferior*, *Nasal Superior*, *Nasal Inferior*.

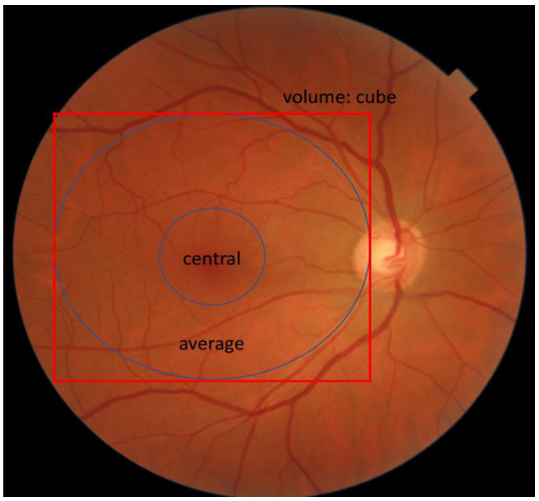

FIGURE 5.1: Choroidal macula

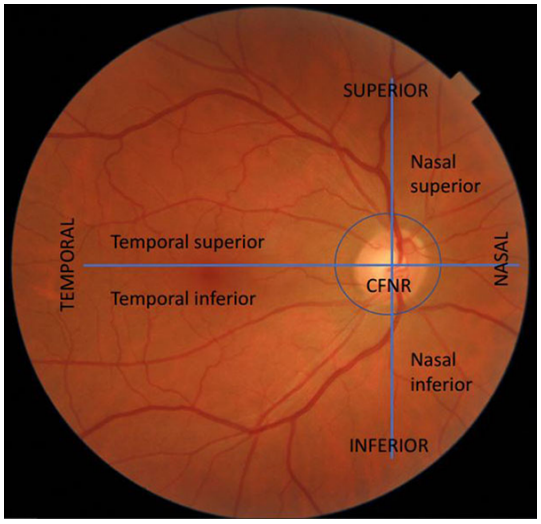

FIGURE 5.2: CFNR

We merge all of those file together to summarize the new obtained data. Note that the individual information are the same as the previous analysis.

1. ID: personal ID
2. eyes: it has values either *left* or *right*, which indicates that the entity data belong to left eye or right eye.
3. Vi_visual_accuity, where $i \in \{1,2,3\}$: that is, we have 3 different parameters V1_visual_accuity, V2_visual_accuity, V3_visual_accuity which indicate the visual accuity value collected in the 3 different stages V1, V2, and V3 respectively.
4. Vi_central (μm), $i \in \{1,2,3\}$: the data collected on the central position for choroidal macula measurement in the stages V1, V2 and V3. It has micrometre (μm) as unit, which is 10^{-6}m.
5. Vi_average (μm), $i \in \{1,2,3\}$: the data collected on the position average (see Figure 5.1) for choroidal macula measurement in the stages V1, V2 and V3. It has μm as unit.
6. Vi_volume ($mm3$), $i \in \{1,2,3\}$: the data collected inside the cube volume (see Figure 5.1) for choroidal macula measurement in the stages V1, V2 and V3. It has cubic millimetre (mm^3) as unit.
7. Vi_total (μm)_CFNR, $i \in \{1,2,3\}$: the data collected over whole CFNR (see Figure 5.2) in the stages V1, V2 and V3. It has μm as unit.
8. Vi_superior (μm)_CFNR, $i \in \{1,2,3\}$: the data collected at the superior of CFNR (see Figure 5.2) in the stages V1, V2 and V3. Recall that it share information of *Temporal Superior* and *Nasal Superior*. It has μm as unit.
9. Vi_temporal (μm)_CFNR, $i \in \{1,2,3\}$: the data collected at the temporal of CFNR (see Figure 5.2) in the stages V1, V2 and V3. Recall that it share information of *Temporal Superior* and *Temporal Inferior*. It has μm as unit.

10. Vi_inferior (μm)_CFNR, $i \in \{1,2,3\}$: the data collected at the inferior of CFNR (see Figure 5.2) in the stages V1, V2 and V3. Recall that it share information of *Nasal Inferior* and *Temporal Inferior*. It has μm as unit.

11. Vi_nasal (μm)_CFNR, $i \in \{1,2,3\}$: the data collected at the nasal of CFNR (see Figure 5.2) in the stages V1, V2 and V3. Recall that it share information of *Nasal Superior* and *Nasal Inferior*. It has μm as unit.

12. Vi_Temporal Superior, $i \in \{1,2,3\}$: the data collected at the Temporal Superior quadrant of CFNR (see Figure 5.2)in the stages V1, V2 and V3.

13. Vi_Temporal Inferior, $i \in \{1,2,3\}$: the data collected at the Temporal Inferior quadrant of CFNR (see Figure 5.2)in the stages V1, V2 and V3.

14. Vi_Nasal Superior, $i \in \{1,2,3\}$: the data collected at the Nasal Superior quadrant of CFNR (see Figure 5.2)in the stages V1, V2 and V3.

15. Vi_Nasal Inferior, $i \in \{1,2,3\}$: the data collected at the Nasal Inferior quadrant of CFNR (see Figure 5.2)in the stages V1, V2 and V3.

5.2 Problem modelling and implementation

5.2.1 Data cleaning

We repeat the same process to clean the physiological data like what we did the LLS analysis. However, we will not merge all of the physiological information with the ophthalmic data. We will only use the following parameters of physiological data: *ID*, *Weight*, *Height*, *Age*.

Furthermore, since we the given ophthalmologic data of 12 trekkers only contain missing data in the stage V3 (1 month after the return), and we will only analysis the data of the stage V1 and V2. Thus, we do not really face the missing data problem in this sense.

Recall that, based on the feedback of ophthalmologists. We will only use the 4 quadrants of CFNR data. Thus, we remove Vi_total (μm)_CFNR, Vi_superior (μm)_CFNR, Vi_temporal (μm)_CFNR, Vi_inferior (μm)_CFNR, Vi_nasal (μm)_CFNR, $i \in \{1,2,3\}$. Furthermore, they also commented that Vi_average (μm) share the same information with Vi_central (μm) (see Figure 5.1). So, we also remove the Vi_average (μm) parameter.

5.2.2 Feature engineering

In this analysis, we also convert the individual's weight and height to BMI. However, since the ophthalmologic data were collected in Barcelona due to the big size of ophthalmological apparatus which were impossible to bring it to Nepal. Also, we just collected the physiological information until the return, we do not have any physiological information after 1 month of return. Thus, we just remain the BMI of the 2 stages in Barcelona, and remove all the rest stage's BMI.

Moreover, in order to correlate the variation of both eyes before the expedition and the return, we create new feature by calculating the difference between all ophthalmologic parameters in V2 and V1. Then, we remove all the single stage data(V1, V2 and V3). The new ophthalmologic features are as follows:

1. visual_accuity_V2_V1: the value of visual accuity collected in V2 minus the value collected in V1.

2. central_V2_V1(μm): V2_central (μm) - V1_central (μm)
3. volume_V2_V1($mm3$): V2_volume ($mm3$) - V1_volume ($mm3$)
4. temporal_superior_V2_V1: V2_Temporal Superior - V1_Temporal Superior
5. temporal_inferior_V2_V1: V2_Temporal Inferior - V1_Temporal Inferior
6. nasal_superior_V2_V1: V2_Nasal Superior - V1_Nasal Superior
7. nasal_inferior_V2_V1: V2_Nasal Inferior - V1_Nasal Inferior

5.2.3 Modelling hypoxia suffering grades

So far, we have not discussed the label that we want to predict to. The question we are trying to answer is, How hypoxia influences the deterioration of human eyes? We rearrange the question as, what is hypoxia? To be able to answer this question, we may need to measure the damage by hypoxia. That is, to measure the suffering level by hypoxia from time series. In this sense, as we have already seen in the previous LLS analysis, SO is the most relevant factor to the LLS prediction, which is one of the assessment of high altitude sickness. Thus, we now define the "suffering level" (the damage by high altitude) based on the parameter SO. Then, we try to predict the suffering level by the given variation of ophthalmologic data.

Originally, we defined 3 suffering grades as follows:

1. Maximum in general suffering (MGS), it is computed by subtracting the maximum SO measured during the 10 stages in the expedition to the measured minimum SO.
$$MGS = max(SO) - min(SO)$$

2. Maximum in local suffering(MLS). This measurement give us a stage-wise suffering information. That is, in which of the 10 stages the individual were suffering the most.
$$MLS = max_{\{\Delta SO_i > 0\}} \quad \Delta SO_i, \text{ where } \Delta SO_i = SO_{i+1} - SO_i, \quad i \in \{1, 2, ..., 10\}$$
SO_i is the SO value measured at the stage$i$.

   Recall that, people are more dangerous as long as the SO decays. Thus, we consider that people only suffered by altitude when their SO value in the current stage is lower than the former stage. Also, we are interested in knowing how deeply people were suffering in only one stage, that is the reason we only take the maximum value of the positive ΔSO_i.

3. Suffering average (SA). It sums the suffering grade of all of the stages that people suffered by altitude ($\{\Delta SO_i > 0\}$), then it divides to the number of stages where people suffered which is written as $|\Delta SO_i > 0|$. With this, we obtain the average suffering grade along with the number of stages of suffering.
$$SA = \frac{\sum_{\Delta SO_i > 0} \Delta SO_i}{|\Delta SO_i > 0|}, \quad i \in \{1, 2, ..., 10\}$$
   where $|\Delta SO >_i 0|$ is the number of stages that people were suffered ($\Delta SO > 0$).

Below, we show the first 10 entities with all the created features and labels (Figure 5.3 and 5.4):

	visual_accuity_V2_V1	central_V2_V1(μm)	volume_V2_V1(mm3)	temporal_superior_V2_V1	temporal_inferior_V2_V1
0	1.0	1.0	-0.18	-23.4	-24.1
1	4.0	0.0	-0.14	-8.6	-9.6
2	-1.0	5.0	0.18	-25.1	-22.8
3	3.0	8.0	0.26	-27.4	-10.9
4	2.0	4.0	0.22	-7.5	6.2
5	-2.0	7.0	0.25	3.7	4.5
6	3.0	3.0	0.14	-0.6	9.4
7	-5.0	-2.0	0.21	20.3	21.1
8	6.0	5.0	0.15	-5.4	-7.8
9	0.0	-1.0	0.15	3.1	15.4

FIGURE 5.3: The first 5 features

	max_suffering	max_local_suffering	suffering_average	Age	BMI
0	5.0	3.0	2.000000	46.0	22.2310
1	5.0	3.0	2.000000	46.0	22.2310
2	16.0	6.0	4.500000	57.0	20.8120
3	16.0	6.0	4.500000	57.0	20.8120
4	10.0	4.0	3.333333	41.0	26.2650
5	10.0	4.0	3.333333	41.0	26.2650
6	12.0	6.0	2.800000	28.0	20.9715
7	12.0	6.0	2.800000	28.0	20.9715
8	18.0	5.0	3.800000	52.0	22.3675
9	18.0	5.0	3.800000	52.0	22.3675

FIGURE 5.4: The remain features and labels

5.2.4 Feature selection

We select the subset of features X_{MGS}, X_{MLS} and X_{SA} for the prediction of the suffering measurements, MGS, MLS and SA respectively. The selection process follows the same approach that we used in the previous analysis, which is to iteratively eliminate 1 feature at time in order to get an importance-ordered features list, then we forwardly create linear model by adding the more important one feature. In the end, we check the model that has the best performance, then the used features at that model will be the selected subset of features.

1. MGS

   The importance-ordered features list that helps the prediction of MGS is below:

   $1)visual\_accuity\_V2\_V1 \quad 2)BMI \quad 3)temporal\_superior\_V2\_V1$
   $4)volume\_V2\_V1(mm3) \quad 5)temporal\_inferior\_V2\_V1 \quad 6)Age$
   $7)nasal\_superior\_V2\_V1 \quad 8)central\_V2\_V1(\mu m) \quad 9)nasal\_inferior\_V2\_V1$

   The list is sorted from the most important feature to least important one. Thus, we can conclude that *visual_accuity_V2_V1* is the factor that correlates the most

with the decrease of SO during the whole expedition.

Then, we follow the above list's order and add one feature at time. The subset of the 5 most important features is selected, which are: visual_accuity_V2_V1, BMI, temporal_superior_V2_V1, volume_V2_V1(mm3), temporal_inferior_V2_V1 (see Figure 5.5).

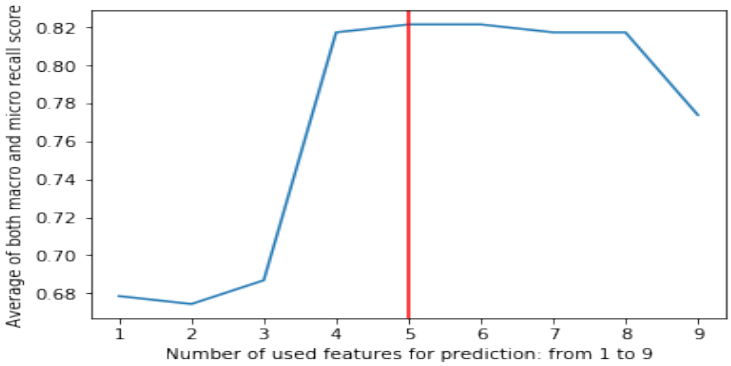

FIGURE 5.5: Feature selection of the MGS prediction

2. MLS

The importance-ordered features list of the MLS prediction.

$1)volume\_V2\_V1(mm3)$ $2)nasal\_inferior\_V2\_V1$ $3)BMI$
$4)temporal\_superior\_V2\_V1$ $5)Age$ $6)temporal\_inferior\_V2\_V1$
$7)nasal\_superior\_V2\_V1$ $8)visual\_accuity\_V2\_V1$ $9)central\_V2\_V1(\mu m)$

The selected subset of features has a size of 4, which are volume_V2_V1(mm3), nasal_inferior_V2_V1, BMI, temporal_superior_V2_V1 (Figure 5.6).

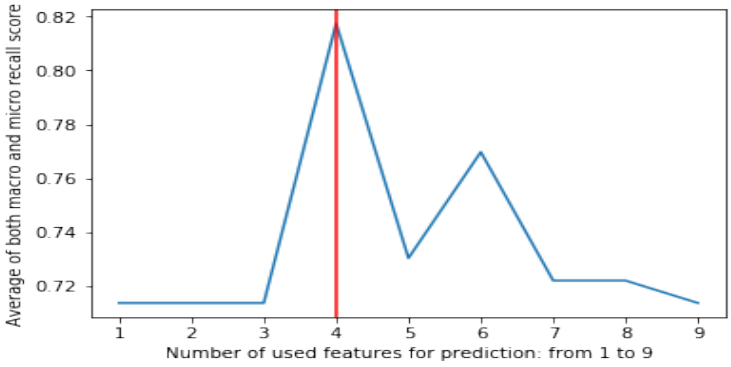

FIGURE 5.6: Feature selection of the MLS prediction

3. SA

The importance-ordered features of the SA prediction.

$1)visual\_accuity\_V2\_V1 \quad 2)BMI \quad 3)volume\_V2\_V1(mm3)$
$4)temporal\_superior\_V2\_V1 \quad 5)Age \quad 6)nasal\_inferior\_V2\_V1$
$7)nasal\_superior\_V2\_V1 \quad 8)temporal\_inferior\_V2\_V1 \quad 9)central\_V2\_V1(\mu m)$

The selected subset of features has a size of 4, which are visual_accuity_V2_V1, BMI, volume_V2_V1(mm3), temporal_superior_V2_V1 (Figure 5.7).

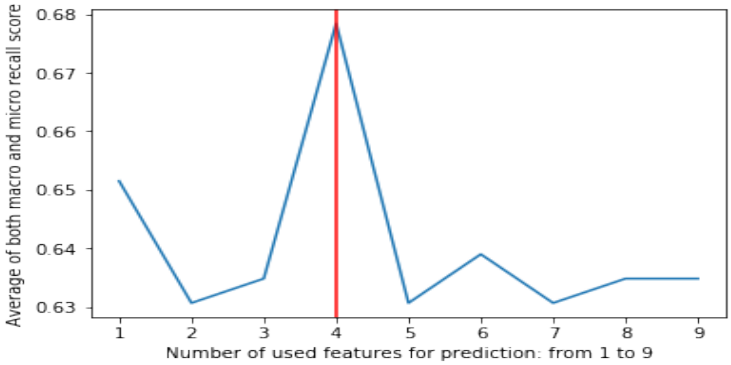

FIGURE 5.7: Feature selection of the SA prediction

5.2.5 Prediction

Again, we follow the same pipeline as we saw in the previous chapter. Firstly, we use the chosen subset features to build models for the prediction of MGS, MLS and SA respectively. Then, we convert the problem into a binary classification problem that splits the values MGS, MLS and SA into 2 classes. The class -1 has a low suffering grade which indicates the individual was not suffered or few suffered. Otherwise, the class -1 is the group that people were suffered by high altitude.

1. MGS prediction

   The distribution of MGS is shown in Figure 5.8, which are integers that range from 5 to 19. Since we aim to convert it into a binary problem, but we do not know how deeply the individuals were suffering with what value of MGS. Thus, we simplify the problem based on the balance of the distribution, we split the MGS into a group of label -1 and another of label 1, as shown in Figure 5.9.

   Then, we follow the same routine. We train a Random Forest model and a SVM model. After that, we compare the performance between them, and choose the best performed one.

2. MLS prediction

   We also show the distribution of MLS in Figure 5.10, the values are integers that range from 3 to 10. Use the same reasoning as before, we split them as shown in Figure 5.11.

3. SA prediction

   The values of SA are real numbers that range from 2.0 to 6.67 (Figure 5.12). We rearrange them as Figure 5.13 shows.

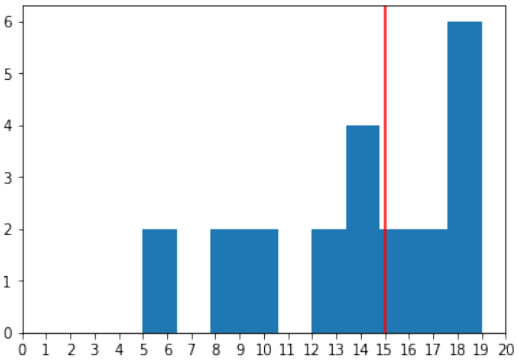

FIGURE 5.8: The distribution of MGS ranges from 5 to 19.

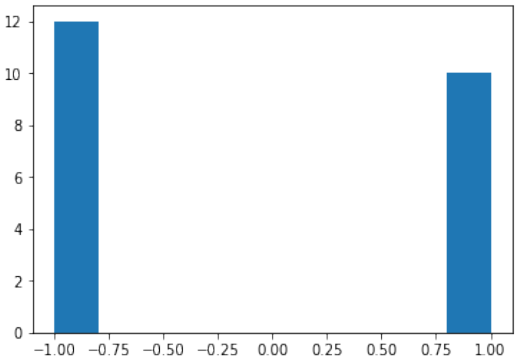

FIGURE 5.9: The rearranged distribution of MGS.

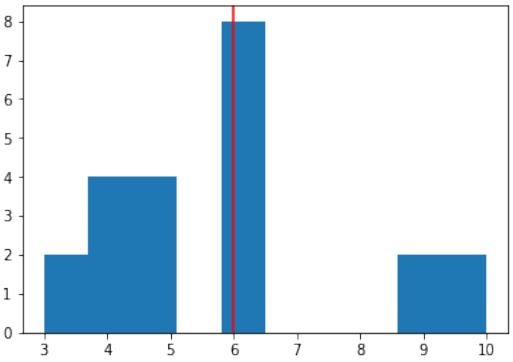

FIGURE 5.10: The distribution of MLS.

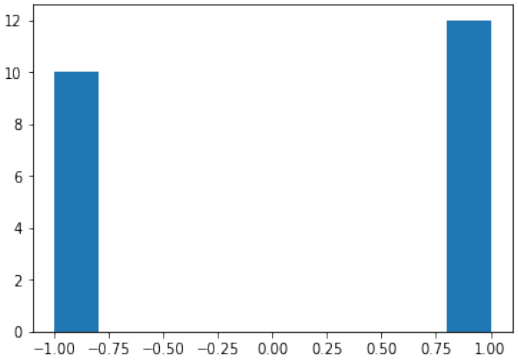

FIGURE 5.11: The rearranged distribution of MLS.

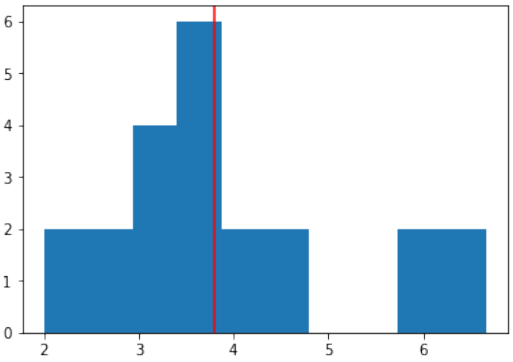

FIGURE 5.12: The distribution of SA.

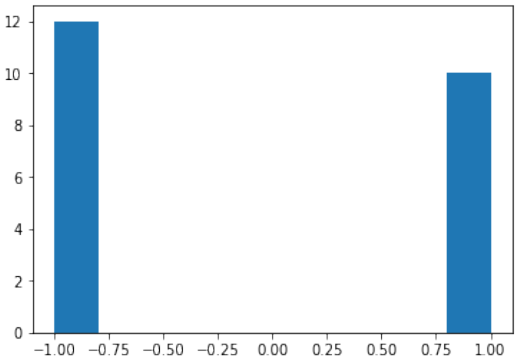

FIGURE 5.13: The rearranged distribution of SA.

5.3 Results

The prediction results of each of the MGS, MLS and SA grade are shown as follows.

1. MGS
   SVM model generates a 0.825 of macro recall and 0.82 of micro (Figure 5.14), whereas Random Forest has 0.867 of macro recall and 0.86 (Figure 5.15). Moreover, the Random Forest model also has an area under the ROC curve bigger than the SVM one (Figure 5.20). Thus, the Random Forest model with the chosen 5 most important features will be used for MGS prediction, which has a mean of 0.865 of recall scores.

Macro recall: 0.825
Accuracy score: 0.818181818182

	precision	recall	f1-score	support
-1	0.90	0.75	0.82	12
1	0.75	0.90	0.82	10
avg / total	0.83	0.82	0.82	22

FIGURE 5.14: MGS-SVM performance

Macro recall: 0.866666666667
Accuracy score: 0.863636363636

	precision	recall	f1-score	support
-1	0.91	0.83	0.87	12
1	0.82	0.90	0.86	10
avg / total	0.87	0.86	0.86	22

FIGURE 5.15: MGS-Random Forest

2. MLS
   The Random Forest model gets an mean recall around 0.869 (0.875 of macro and 0.86 of micro, see Figure 5.17) and its area under the ROC curve is 0.83 (Figure 5.21). However, the SVM model predicts MLS with a lower mean recall which is 0.818, but its area under the ROC curve is higher (0.86). In this sense, we respect the same criterion as we used for feature selection, we choose the Random Forest model because of its higher mean recall score.

Macro recall: 0.816666666667
Accuracy score: 0.818181818182

	precision	recall	f1-score	support
-1	0.80	0.80	0.80	10
1	0.83	0.83	0.83	12
avg / total	0.82	0.82	0.82	22

FIGURE 5.16: MLS-SVM performance

Macro recall: 0.875
Accuracy score: 0.863636363636

	precision	recall	f1-score	support
-1	0.77	1.00	0.87	10
1	1.00	0.75	0.86	12
avg / total	0.90	0.86	0.86	22

FIGURE 5.17: MLS-Random Forest

3. SA
   In the SA prediction, the macro, micro recall and the area under the ROC curve of the SVM model are 0.675, 0.68 and 0.62 respectively (Figure 5.18 and 5.22). Likewise, the scores of Random Forest are 0.85, 0.86, and 0.72 respectively (5.19). With no doubt, we choose Random Forest with the selected 4 features setting.

Macro recall: 0.675
Accuracy score: 0.681818181818

	precision	recall	f1-score	support
-1	0.69	0.75	0.72	12
1	0.67	0.60	0.63	10
avg / total	0.68	0.68	0.68	22

FIGURE 5.18: SA-SVM performance

Macro recall: 0.85
Accuracy score: 0.863636363636

	precision	recall	f1-score	support
-1	0.80	1.00	0.89	12
1	1.00	0.70	0.82	10
avg / total	0.89	0.86	0.86	22

FIGURE 5.19: SA-Random Forest

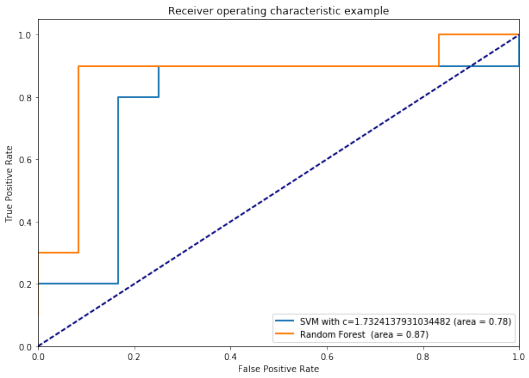

FIGURE 5.20: MGS-ROC and area under the curve

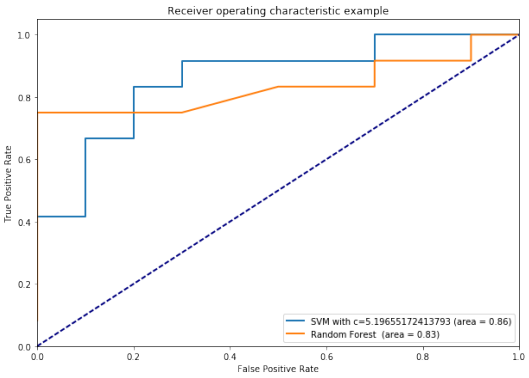

FIGURE 5.21: MLS - ROC and area under the curve

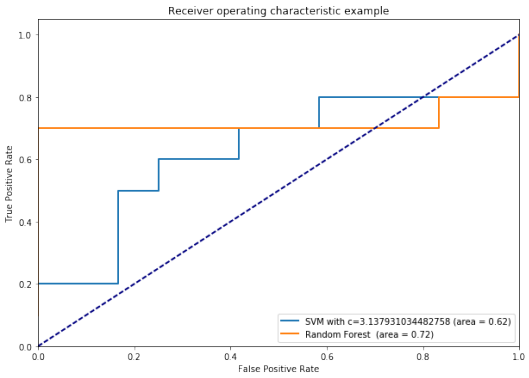

FIGURE 5.22: SA - area under the curve

5.4 Statistical validation of the results

We also do the permutation test and Bayesian approach to support the above results. Overall, both validations give positive supports to the prediction results.

1. MGS
   The chosen model for MGS prediction has really good performance, which it has a high significant p-value in the permutation test of the area under the ROC curve and the mean recall scores (see Figure 5.23 and 5.24). Also, the Beta-Binomial distributions of the class -1 and 1 are plotted with the corresponding credibility interval to validate the prediction results (Figure 5.25 and 5.26).

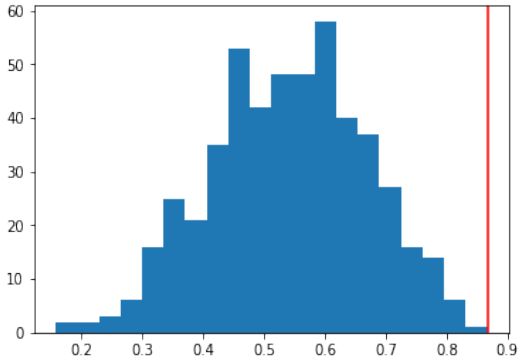

FIGURE 5.23: Permutation test (MGS): Area under the ROC curve

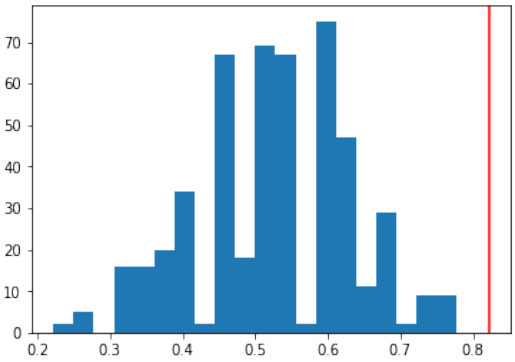

FIGURE 5.24: Permutation test (MGS): mean recall scores

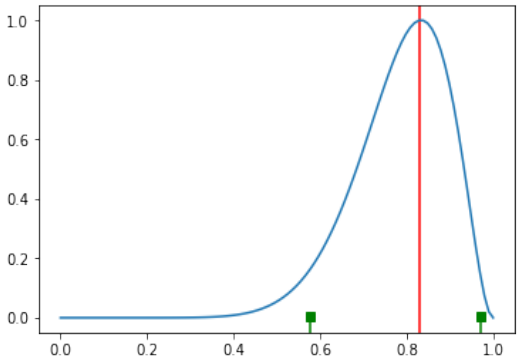

FIGURE 5.25: Cred. intervals(MGS) of class-1:$[0.576, 0.960]$

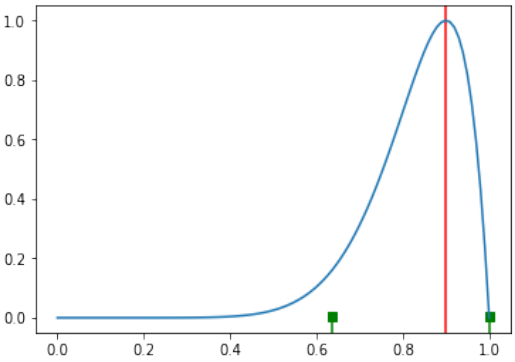

FIGURE 5.26: Cred. intervals(MGS) of class 1:$[0.636, 1.0]$

2. MLS
   The permutation test of MLS is shown in the Figure 5.27 and 5.28 which has the corresponding p-values 0.002 and 0.004 of the area under the ROC curve and the mean recall scores. The Bayesian validation is shown in Figure 5.29,5.30.

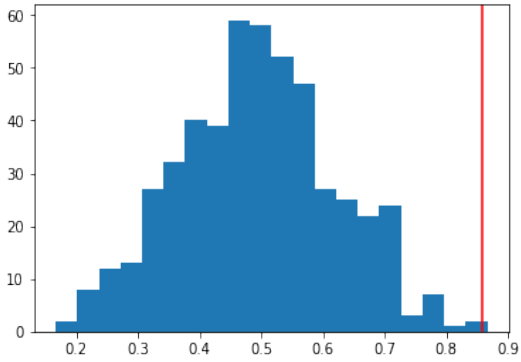

FIGURE 5.27: Permutation test (MLS): Area under the ROC curve

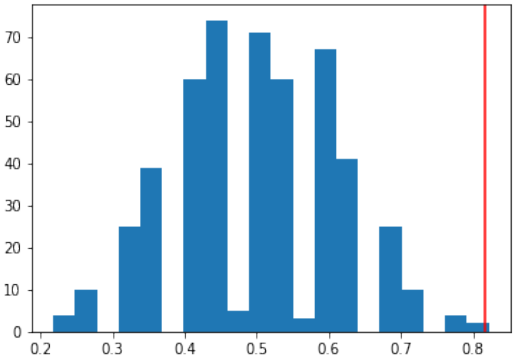

FIGURE 5.28: Permutation test (MLS): mean recall scores

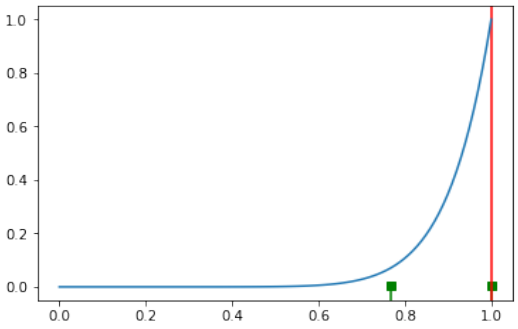

FIGURE 5.29: Cred. intervals(MLS) of class−1: [0.768, 1.0]

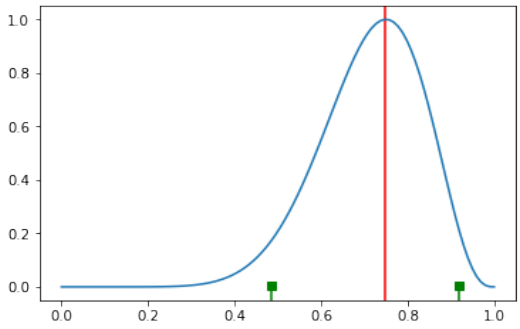

FIGURE 5.30: Cred. intervals(MLS) of class1: [0.485, 0.920]

3. SA
   The obtained p-value of the area under the ROC curve and the mean recall score in the permutation test of SA are 0.012 and 0.048, respectively (Figure 5.31 and 5.32). See Figure 5.33 and 5.34 for Beta-Binomial distribution validation.

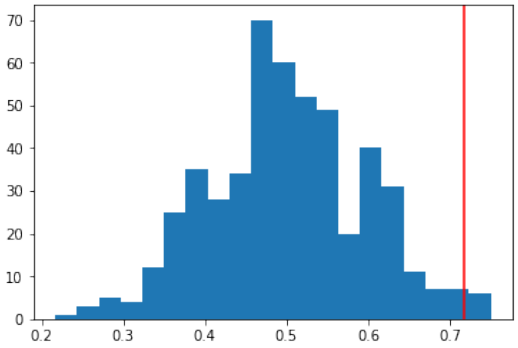

FIGURE 5.31: Permutation test (SA): Area under the ROC curve

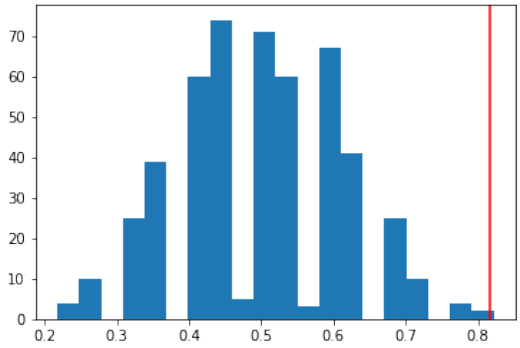

FIGURE 5.32: Permutation test (SA): mean recall scores

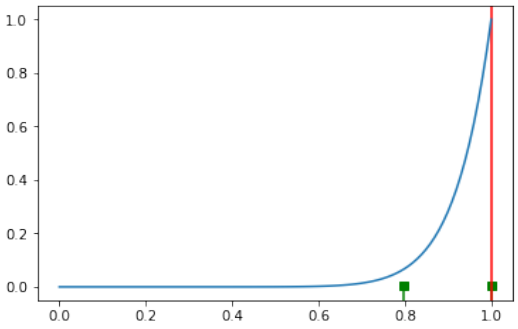

FIGURE 5.33: Cred.intervals(SA) of class−1: [0.798, 1.0]

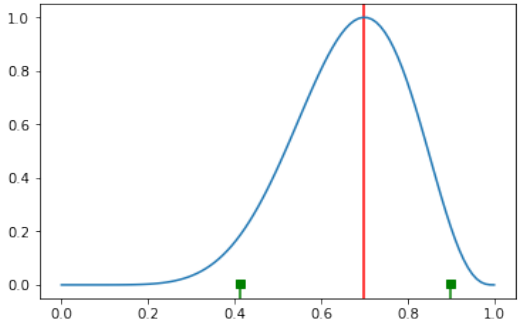

FIGURE 5.34: Cred. intervals(SA) of class1: [0.414, 0.899]

Chapter 6

Conclusion and future works

In this last chapter, we detail the project's conclusion and propose the possible future works.

6.1 Conclusion

Based on the results obtained at the previous two chapters, we can conclude the project as the following aspects.

First of all, we recall that our objective of the project is to study the cause of generalized hypoxia which is altitude sickness. Thus, we split the study into two different analysis: Lake Louise Score (LLS) analysis and Ophthalmic data analysis. However, to understand well the root of hypoxia, it is not enough only find the factors that cause altitude sickness, but also which factor is more relevant than the others. Hence, an importance-ordered list of the factors that cause altitude sickness is also required.

Now, in the first LLS analysis. We are able to predict the LLS by the given physiological data and stage-wise information. Moreover, the importance-ordered list is also found, which is below: SO, BMI, distance(km), Gender, FC, down, TAD, TAS, Age, up. However, with only the 5 most important factors, we are able to predict LLS with a mean of 0.74 of the recall scores.

Regarding the Ophthalmic data analysis, we define 3 different measurements which can somehow explain how people were damaged by hypoxia in a global sense (MGS), a stage-wise sense(MLS) and an average suffering sense(SA) during the Everest expedition. Then, we use the given physiological data merged with stage-wise information and ophthalmic data to predict MGS, MLS and SA. The obtain results are positive, which all of them are over 70% of certainty in their recall scores. Also, the results are supported by doing two different validation tests which are Permutation test and Bayesian approach. Thus, we consider that the obtained results are trustworthy.

Because of the obtained positive results on both of the analysis, and also the results are given support from validation tests. Thus, we can conclude that our goal is fulfilled.

6.2 Future works

Due to the complex structure of human body, in this project we predict LLS in order to explain what are the factors that affect most to human in a high altitude. However, we hope this can be somehow correlated to other Hypoxia studies in future, such as COPD. Furthermore, we defined 3 measurements that grade the suffering level in a global, stage-wise and average sense. Despite they were separately worked well,

which we gain a high accuracy of the predictions. But, we consider that a merged sense can be also helpful, that means we can merge the 3 grades together such as sum or other mathematical operations.

In the end, we matter any required and necessary improvements on this project that can help the medical researchers to improve the diagnosis and treatment of hypoxia.

Bibliography

Bernhard E. Boser Isabelle M. Guyon, Vladimir N. Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Proceeding COLT '92 Proceedings of the fifth annual workshop on Computational learning theory Pages 144-152*. URL: https://dl.acm.org/citation.cfm?id=130401.

Brunner, Jerry. *Permutation and Randomization Tests*. URL: http://www.utstat.toronto.edu/~brunner/oldclass/appliedf12/lectures/2101f12PermutationRandomization.pdf.

Corinna Cortes, Vladimir Vapnik (1995). "Support-Vector Networks". In: URL: http://image.diku.dk/imagecanon/material/cortes_vapnik95.pdf.

Credibility Interval [online]. URL: http://www.yhec.co.uk/glossary/credibility-interval/.

Cross-validation (statistics). URL: https://en.wikipedia.org/wiki/Cross-validation_(statistics)#Leave-one-out_cross-validation.

Deng, Kan. *Feature selection*. URL: https://www.cs.cmu.edu/~kdeng/thesis/feature.pdf.

Evangeline, D. Preetha et al. (2013). "Feature subset selection for irrelevant data removal using Decision Tree Algorithm". In: *2013 Fifth International Conference on Advanced Computing (ICoAC)*. URL: https://ieeexplore.ieee.org/document/6921962/.

Hai Thanh Nguyen Katrin Franke, Slobodan Petrovi'c (2011). "On General Definition of L1-norm Support Vector Machines for Feature Selection". In: *International Journal of Machine Learning and Computing*. URL: http://www.ijmlc.org/papers/41-L0204.pdf.

Latorre, Ferran. *The project description in Ferran Latorre's blog*. URL: http://www.ferranlatorre.com/es/sherpa-everest-2017-project/.

learn, scikit. *1.4. Support Vector Machines*. URL: http://scikit-learn.org/stable/modules/svm.html.

Li Wang, Ji Zhu and Hui Zou (2006). "THE DOUBLY REGULARIZED SUPPORT VECTOR MACHINE". In: *Statistica Sinica*. URL: http://www3.stat.sinica.edu.tw/statistica/oldpdf/A16n214.pdf.

Library, Health. *Vital Signs (Body Temperature, Pulse Rate, Respiration Rate, Blood Pressure)*. URL: https://www.hopkinsmedicine.org/healthlibrary/conditions/cardiovascular_diseases/vital_signs_body_temperature_pulse_rate_respiration_rate_blood_pressure_85,P00866.

Mandal, Ananya. *What is Body Mass Index (BMI)?* URL: https://www.news-medical.net/health/What-is-Body-Mass-Index-(BMI).aspx.

MayoClinic. *Symptoms Hypoxemia*. URL: https://www.mayoclinic.org/symptoms/hypoxemia/basics/definition/sym-20050930.

Meinshausen, Nicolai (2006). "Relaxed Lasso". In: URL: https://stat.ethz.ch/~nicolai/relaxo.pdf.

P. S. Bradley, O. L. Mangasarian (1998). "Feature Selection via Concave Minimization and Support Vector Machines". In: *Proceedings of the Fifteenth International Conference (ICML)*. URL: https://pdfs.semanticscholar.org/14b8/cbbaf08d1f00145f83c0270833a03e4pdf.

Peter Radchenko, Gareth M. James (2011). "IMPROVED VARIABLE SELECTION WITH FORWARD-LASSO ADAPTIVE SHRINKAGE". In: URL: https://arxiv.org/pdf/1104.3390.pdf.

Pishro-Nik, H. *Introduction to probability, statistics, and random processes*. URL: https://www.probabilitycourse.com/chapter9/9_1_2_MAP_estimation.php.

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Department of Statistics and Division of Biostatistics, Stanford University*. URL: https://cs.nyu.edu/~roweis/csc2515-2006/readings/lasso.pdf.

Trevor Hastie Robert Tibshirani, Jerome Friedman (2001). *The elements of statistical learning*. Reading, Massachusetts: Springer.