UNIVERSITÉ DE MONTRÉAL

PREDICTION OF PILOT'S ABSENTEEISM IN AN AIRLINE COMPANY

AMIR HOSEIN HOMAIE SHANDIZI

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE INDUSTRIEL)

MARS 2014

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

PREDICTION OF PILOT'S ABSENTEEISM IN AN AIRLINE COMPANY

présenté par : HOMAIE SHANDIZI Amir Hosein

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. FRAYRET Jean-Marc, Ph.D., président

M. AGARD Bruno, Doct., membre et directeur de recherche

M. PARTOVI NIA Vahid, Doct., membre et codirecteur de recherche

M. GAMACHE Michel, Ph.D., membre et codirecteur de recherche

M. ADJENGUE Luc, Ph.D., membre

DEDICATION

To my lovely sister: Samira

ACKNOWLEDGMENTS

This thesis is the result of a team work and I must thank all the academic and professional advisors without whom I would never be able to finish it. First of all, I would like to express the deepest appreciation to my supervisor Professor Bruno Agard who showed me the correct path of research with his immense knowledge, who also gave me the confidence of attacking hard problems. He also provided me full support and strong motivations. I had the privilege of being his student and benefit from his continuous guidance during my research and study. I could not have imagined having a better supervisor and mentor.

I would also like to offer my special thanks to my co-supervisor Professor Vahid Partovi Nia who generously spent his time for improving the thesis. His advices and critiques with his deep knowledge in theory and practice helped me getting around all obstacles of a scientific research. I wish to acknowledge the help provided by Professor Michel Gamache, my co-supervisor of the research. His encouragement, insightful comments, and his vast knowledge was a big support during research and while writing this thesis.

I had the chance of doing an internship in an airline company and got familiar with the research methods in practice. My sincere thanks also go to all the technical advisors of the project, Jerome-Olivier Ouellet, Mathieu Nonki, Olga Hormaza, Scott Richardson, and Steven Duke. Their comments and discussions improved considerably the results of this thesis. I would like to be thankful to all the other members of operational research group, Karine Lacerte, Jacques Cherrier, and Rodeler Joseph for the friendly environment that they provided and made this project as a memorable experience for me. I am particularly grateful for the assistance given by Jerome-Olivier Ouellet as the technical supervisor and manager of the project. He was the key person in making the results of this thesis applicable in the real situation and this research could not have resulted without his knowledge, support, deep understanding, and management skills.

I also thank the MITACS for the financial support of this thesis which helped me to concentrate completely on this project.

I would also like to thank all my comrades for their support, cups of tea, laughs, and everything; specially Vahid Partovi Nia, thank you for being my best friend. I owned you many things.

Last but not least, I wish to thank wholeheartedly my father, my mother, my sisters, and all my family for their support and encouragement throughout my study and in all steps of my life. They are meaning of life for me and I am so proud of having them. I could not even have started this thesis without their support and encouragement.

RÉSUMÉ

Les compagnies aériennes sont soumises aux nombreuses sources de perturbations pendant les opérations. Il est essentiel pour ce type d'industrie de prédire les origines des perturbations dans les différents niveaux de gestion pour réduire les coûts de rattrapage du calendrier. Une des sources les plus importantes et coûteuses de perturbation dans les compagnies aériennes est l'absentéisme des pilotes au moment de l'opération des vols.

Dans ce mémoire, nous nous concentrons sur l'absentéisme des pilotes pour cause de maladie. Nous proposons une méthode d'apprentissage supervisé qui est capable de prédire la somme mensuelle des heures de maladie chez les pilotes après la publication du calendrier. La méthode proposée utilise les caractéristiques du calendrier mensuel comme les variables explicatives et elle fait la prédiction en utilisant d'un algorithme itératif.

La méthode a été vérifiée avec des données réelles et une amélioration considérable a été observée dans les résultats. Pour rendre la méthode en situation réelle, nous avons créé une interface facile à utiliser comme un système d'aide à la décision. Cette interface automatise l'ensemble du processus de prédiction.

ABSTRACT

Airline companies are subject to a considerable number of disruptions during operations. It is vital for this type of industry to predict the source of disruptions in different levels of management to reduce the costs of schedule recovery. One of the most important and costly source of disruption in the airlines is absenteeism of the pilots at the time of the flights operation.

In this master thesis, we focus on the absenteeism of the pilots because of the sickness. We propose a supervised learning method which is able to predict total monthly sick hours after publishing the schedule. The proposed method uses characteristics of the monthly schedule as the explanatory variables and the prediction is made by using an iterative algorithm.

The model was tested with real data and a substantial improvement was observed in the results. For applying this method in business environment, we created a user-friendly web application as the decision support system. This application automates the whole process of prediction.

CONDENSÉ EN FRANÇAIS

L'objectif principal de ce mémoire est la création d'un système d'aide à la décision capable de prédire la somme des heures de maladie chez les pilotes d'une compagnie aérienne. Pour réaliser cet objectif, nous utilisons une méthode d'apprentissage supervisé dans laquelle l'arbre de décisions est l'outil principal de la méthodologie et les caractéristiques du calendrier mensuel sont des variables explicatives.

La base de données a deux parties : une première partie est l'historique qui décrit les caractéristiques des pairings (rotations) et aussi les événements de maladies associés à chaque pairing, et la seconde partie est le nouveau calendrier qui décrit les caractéristiques des pairings planifiés pour le nouveau mois. Notre objectif est de créer un système d'aide à la décision pour prédire la somme des heures de maladie pour le nouveau mois, ($\hat{s}_{i\,n+1}$), par rapport au calendrier du nouveau mois ($\mathbb{S}_{i\,n+1}$), aux caractéristiques des pairings et à l'histoire de maladies du passé ($\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{in}$). Le processus d'apprentissage proposé suggère d'utiliser une boucle pour choisir les meilleurs arbres de décisions.

Dans cette boucle, nous commençons par fixer un paramètre, appelé a, qui est le nombre de mois consécutifs nécessaires pour bâtir le premier arbre de décision *stable*. Ensuite, nous fusionnons les ensembles de données, $\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{ia}$, dans une base de données unique, notée Γ_{ia}. Dans la première étape de la boucle, un arbre de décisions est construit pour Γ_{ia} et les prédictions des heures de maladie sont calculées pour chaque niveau de cet arbre. Ensuite, l'arbre est coupé au niveau ayant l'erreur minimum de prédiction pour le mois $(a+1)$. L'arbre coupé est appelé $\widetilde{\mathfrak{T}}_{i\,a}$.

Cette boucle se répète pour obtenir $(n-1)$ arbres de décisions coupés $\widetilde{\mathfrak{T}}_{i\,a}, \widetilde{\mathfrak{T}}_{i\,a+1}, \dots, \widetilde{\mathfrak{T}}_{i\,a+n-1}$. L'algorithme de cette boucle s'écrit comme suit :

Choisissez a,

Considérez un ensemble vide comme l'ensemble des arbres de décision,

Pour z de a à $n-1$ faites:

- Fusionnez $\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{iz}$,
- Créez un arbre de décision pour l'ensemble de données fusionné,

- Calculez la prédiction pour le mois suivant en utilisant $\mathbb{S}_{i\,z+1}$ pour chaque niveau de l'arbre obtenu,
- Calculez l'erreur pour le niveau,
- Coupez l'arbre au niveau ayant l'erreur de prédiction minimum,
- Appelez l'arbre coupé $\widetilde{\mathfrak{T}}_{i\,z}$,
- Ajoutez $\widetilde{\mathfrak{T}}_{i\,z}$ dans l'ensemble des arbres de décision.

L'idée essentielle de la méthodologie est de trouver et d'utiliser les meilleurs scénarios des mois passés par rapport à l'historique de maladies chez les pilotes. À la fin de l'algorithme, nous utilisons $(n-1)$ arbres de décision $\widetilde{\mathfrak{T}}_{i\,a}, \widetilde{\mathfrak{T}}_{i\,a+1}, \ldots, \widetilde{\mathfrak{T}}_{i\,a+n-1}$ et le calendrier du nouveau mois ($\mathbb{S}_{i\,n+1}$) pour faire la prédiction des heures de maladies dans le nouveau mois $(n+1)$. Si nous utilisons le calendrier du nouveau mois comme l'entrée de ces arbres, ils donnent $(n-a-1)$ valeurs différentes pour la prédiction de nouveau mois, $\hat{s}\left(\widetilde{\mathfrak{T}}_{i\,a}, \mathbb{S}_{i\,n+1}\right)$, $\hat{s}\left(\widetilde{\mathfrak{T}}_{i\,a+1}, \mathbb{S}_{i\,n+1}\right)$, …, $\hat{s}\left(\widetilde{\mathfrak{T}}_{i\,n-1}, \mathbb{S}_{i\,n+1}\right)$. Chacune de ces prédictions est basée sur les règles d'association qui expliquent le mieux la maladie d'un mois précédent. De cette façon, nous considérons la possibilité d'occurrence des scénarios précédents à l'avenir. Nous considérons une moyenne pondérée de ces estimations comme la prédiction pour le nouveau mois.

La méthodologie proposée a été appliquée dans une compagnie aérienne et les résultats montrent que dans la plupart des cas, les prédictions ont une erreur acceptable et la méthodologie proposée a amélioré d'au moins 13 pourcents les prédictions mensuelles de maladie pour l'année 2012 en comparaison avec la méthode actuelle de prédiction. Cette amélioration est obtenue lorsque l'on considère que le coût de sous-prédiction est égal au coût de sur-prédiction. Si l'on considère un coût de sous-prédiction 1,5 fois le coût de sur-prédiction, plus similaire à la valeur réelle du ratio de coûts dans les compagnies aériennes, l'amélioration de la prédiction augmente à 21 pourcents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF NOTATIONS

$\mathbb{S}_{ij}$	The table of pairings schedule for position *i* in month *j*.
$\mathbb{O}_{ij}$	The operations' record for position *i* and month *j*.
$\mathbb{D}_{ij}$	The database for position *i* and month *j*. The result of merging $\mathbb{S}_{ij}$ and $\mathbb{O}_{ij}$.
i	As an index, denotes the position.
j	As an index, denotes the month.
k	As an index, denotes the pairing.
l	As an index, denotes the flight leg.
I_{ijkl}	Flight sick indicator.
C_{ijkl}	Flight total credit.
c_{ijk}	Pairing total credit.
S_{ijkl}	Flight sick time.
s_{ijk}	Pairing sick time.
s_{ij}	Total sick time for position *i* in month *j*.
y_{ijk}	Response variable, sick indicator for a pairing.
$\mathfrak{T}$	The complete decision tree
m	Number of terminal nodes (regions) of $\mathfrak{T}$
r_m	Region *m* of a decision tree
p_m	Probability of sickness in region *m* of a decision tree.
c_m	Total credit in region *m* of a decision tree.
m	Number of levels of a decision tree.
$\hat{s}(\mathfrak{T}, \mathbb{S}_{i\,n})$	Sickness time prediction for pairing schedule, $\mathbb{S}_{i\,n}$, based on the decision tree, $\mathfrak{T}$.
Γ_{ia}	The result database of merging $\mathbb{D}_{i1}, \mathbb{D}_{i2}, \ldots, \mathbb{D}_{ia}$.

cp	Cost-complexity pruning.
$\mathfrak{T}_{i\,a}$	Decision tree without pruning based on Γ_{ia}.
$\langle\mathfrak{T}_{i\,a}\rangle_{\hbar}$	The pruned tree $\mathfrak{T}_{i\,a}$ at level $\hbar$.
$e_{i\,a_{\hbar}}$	The error of prediction at level $\hbar$ of the original decision tree $\mathfrak{T}_{i\,a}$.
η	The proportion of under-prediction cost on over-prediction cost.
$\widetilde{\mathfrak{T}}_{i\,a}$	Pruned decision tree based on Γ_{ia}.
$e_{i\,a}$	Error of $\widetilde{\mathfrak{T}}_{i\,a}$ in predicting$s_{i\,a+1}$.
$\tau_{n\,a}$	Similarity between schedule of month a and month n in a specific position.
$\hat{s}_{i\,j}$	Final sick hours prediction for position i and month j.

LIST OF AIRLINE TECHNICAL TERMS

Base	The domiciles which can be considered for starting a pairing of an airline.
Block holder	Pilot with a flight schedule for a month, Line pilot.
Captain	Responsible pilot for the flight operations and safety in an aircraft.
Deadhead	A pilot who is assigned to fly as a passenger in a specific flight for transferring to duty airport. The deadhead does not pilot the aircraft.
First Officer	Second pilot in commercial aviation.
Leg	A flight in a pairing.
Non-bidding	Pilots of an airline which is neither block holder nor reserve.
Pairing	Combination of consecutive flight legs which start and end at the same domicile.
PBS	Preferential bidding system, a computer program that optimises airline workforce schedule.
Position	Combination of aircraft type and seat.
Relief Pilot	Third pilot in commercial aviation, used for long distance flights.
Reserve	On-call pilot for substituting the block holder in the necessary cases.
Seat	The seat in a flight deck which determines the arrangement of the pilots.
Type of aircraft	Divided categories of aircrafts for the purpose of certification.

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

In an airline company, crewing costs are second important cost after fuel costs, and pilots are the most important airline crew. Pilots are qualified for just one aircraft type as Captain, First Officer or Relief Pilot. So for big airline companies, in which there are different types of aircraft, having a good prediction of pilot absenteeism helps to manage the operations extensively.

This thesis proposes an efficient way for predicting one of the most important reasons of absenteeism, i.e. pilots' sickness. Before starting a detailed analysis some preliminary subjects need to be explained. Chapter 1 gives a brief review of the reserve crew and crew scheduling process. Assumptions, general and specific objectives and also the structure of the thesis are also explained in this chapter.

1.1 Reserve Crew

In an airline company, in general, a pilot is qualified for one *type of aircraft* and one *seat*. The seat for the pilot, in a hierarchical rank, can be *Captain*, *First Officer* or *Relief Pilot*. This means a first officer of the Airbus A320 cannot be the captain of the same aircraft. The opposite is possible, but it augments the operations costs because the salary of a captain is higher than a first officer. In this study, a *position* is the combination of an aircraft type and a seat, e.g. 320 CA is the captain of Airbus A320.

In each month a pilot, based on his/her work schedule can be *block holder*, *reserve* or *non-bidding*. After publishing the monthly flight schedule, the block holders bid for determining the details of their own working schedule according to the airline's bidding rules. Because of different unexpected conditions (such as weather conditions, pilot's sickness, aircraft maintenance etc), it is impossible to have a fix and unchangeable schedule for monthly flights. Therefore a number of pilots are in reserve in order to take the place of the block holders when the schedule changes.

It is important to have a good prediction for the number of the required reserves. A wrong number of reserves can cause two different extra operational costs for the airline company. First, if the number of reserves is greater than the number of absent pilots, the company must pay some pilots for doing nothing. Second, if the number of reserves is less than the number of absent pilots, then

the airline must pay extra for calling an out-of-duty pilot or even in the worst case it can cause the cancelation of some flights. Therefore, costs of under predictions are higher than those for over predictions.

The reserves are in backup and are used if operations could not be implemented according to the schedule. A pilot could miss his next flight because of a delay in the previous one, a change of aircraft and some other reasons can cause the use of reserve pilots. The most important reason for using the reserves in an airline is the last minute calling sick by the block holders. This covers almost 60 percent of reserves replacements in big airlines.

In this study, we focus on the prediction of absenteeism of pilots when they are calling sick. Hence, we only consider the replacements by reserves that were based on the declared sickness of pilots.

1.2 Crew Scheduling

Crew scheduling for airlines consists of different tasks. Here, we describe an introductory explanation that can help readers to follow future sections. Interested readers are referred to the text books on the airline operations such as Bazargan (2010) or Grosche (2009). For the detailed analysis of preferential bidding systems at airlines see Gamache, Soumis, Villeneuve, Desrosiers, and Gelinas (1998) and Barnhart, Belobaba, Odoni, and Barnhart (2003).

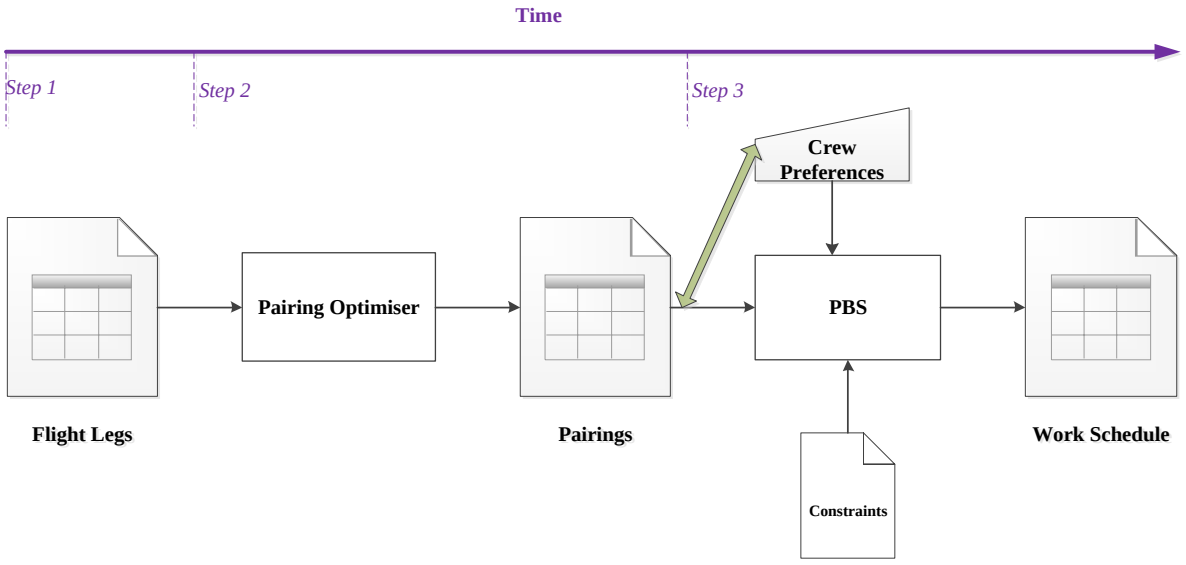


Figure 1-1: Crew scheduling in the airlines.

The first step in crew scheduling process, as can be seen in **Erreur ! Source du renvoi introuvable.**, is publishing the list of monthly flight legs. Based on tactical and strategic decisions, a list of flights for a business month is published by the commercial department of the airline. Each flight has its own planned characteristics such as flight departure, arrival date and time, assigned aircraft type, departure and arrival airports, flight duration, flight credits, etc.

Despite other industries in which working duty consists of shifts or days, in the airlines there is an additional duty period that is called *pairing*. A pairing is a combination of consecutive flight legs which start and end at the same domicile (See Figure 1-2).

Figure 1-2: An example of pairing with 4 legs, started from Montreal.

The domiciles which can be considered in pairings are *airline bases*. Every pilot is assigned to a base for starting and finishing his own duty period. Each airline has its own bases. For example current bases for US Airways are Charlotte, Philadelphia, Washington, and Phoenix. Current bases for Air Canada are Toronto, Montreal, Vancouver, and Winnipeg.

Sometimes it is necessary to move a pilot to his duty location as a passenger because of the limitation in daily flight hours for pilots or for containing all the flight legs in the pairing list. In this case, the pilot is called *deadhead* and he does not pilot the aircraft.

After publishing the pairing table, in the second step, a computer program called *preferential bidding system (PBS)* optimises airline workforce schedule. The PBS is usually executed for a

monthly schedule and its inputs could be *airline operations requirements* (list of the pairings), *crew preferences* (bidding) and some *constraints*.

After pairings are made, in the third step, bidding process starts. In the bidding process each crew requests for a certain schedule or determines his preferences. Then, the PBS assigned the list of the pairings to the pilots. This process must minimize the costs of the operations and matches aircraft type, flying routes, and pairings in a way that each pairing is assigned to only one qualified pilot. This is evident that the PBS must consider pilots' working time and bases, such that there is no overlap in the work schedule of each pilot.

The last input for the PBS is some constraints which are legal crewing solutions that must be considered in the PBS, such as

- Government Regulations,
- Collective Bargaining Agreements,
- Airline Policies.

At the end, the PBS makes the optimization based on the all explained inputs and the airline method of awarding. This method can differ from one airline to another. It could be *honoring seniority* in which the most senior qualified pilot for a position will be awarded by his bidding; the second will be awarded by best matches between his bidding and the remaining pairings; then the third, and so on.

1.3 Assumptions

In this study, we attack the problem of monthly sickness prediction under some minor restrictions. The predictions must be generated after publishing pairings schedule and before the bidding process because at that time the number of the reserves for the following month must be determined. For making the appropriate decision support system, we suppose that the following information exists:

- The monthly list of the pairings.
- At least two years of history for a position.

- The prediction is based on the schedule, so it does not assume changes in schedule during the operations.
- This study considers only sickness among the block holders and not the reserve pilots.
- The format of all the tables that are used in making database is subject to no change.

1.4 General Objective

The general objective of this research is to develop a decision support system for predicting sickness hours in each position based on the monthly list of the pairings, in an airline company, in order to size properly the reserve crew.

1.5 Specific Objective

For achieving the goal of this project, it is necessary to define specific objectives. These specific objectives are as follows:

1- Data pre-processing: Connecting tables of schedules, bidding results and operation records to create a big data base that contains all the information, “correcting mistakes” in the data, here it seems all the data was clean.

2- Developing a method for predicting pilots’ sickness based on the history and monthly schedule of pairings.

3- Determining association rules that helps the airline managers find the characteristics of pairings in which the pilots are less interested.

4- Automating all the processes by making a user-friendly application for implementing the developed method in the business area.

1.6 Thesis Structure

The structure of the rest of the thesis is as follows. In Chapter 2, a literature review is presented. Chapter 3 describes the problem in detail. The solution approach and methodology are proposed in Chapter 4. In Chapter 5, the data pre-processing, implementation and results based on a real dataset are explained. We conclude the thesis and propose further extensions of this subject in Conclusion.

CHAPTER 2 LITERATURE REVIEW

As pilots' absenteeism prediction is a tool for disruption management, in Section 2.1, a brief review of disruption management in airlines is presented. Section 2.2 deals with classification and regression trees which is the main statistical methodology of our prediction algorithm. Section 2.3 introduces the R packages that are used for implementing the methodology of this thesis.

2.1 Disruption Management

Unlike the strategic and tactical problems of an airline company, during flight operations most of problems must be solved in a short period of time. Therefore, managing irregular operations (disruptions) is a subject of considerable interest among many authors.

In the airline industry, disruptions can occur for several reasons: mechanical problems, weather conditions, crew sickness, security, and so on. These kinds of problems may cause flight delays or even flight cancelations. However, in many cases crew reassignment is still feasible.

One of the first works on disruption management in airline discipline was the two minimum-cost flow network models presented by Jarrah et al. (1993) for absorbing the shortages. The first model chooses the set of delayed flights and the second one chooses the set of cancelations. Based on these models, a decision support system (DSS) was implemented at United Airlines and a result of valuable cost saving for using this DSS has been published (Rakshit et al., 1996).

From a different point of view, Bratu and Barnhart (2006) dealt with airline schedule recovery problem and developed an optimal trade-off between airline operations costs and passengers delay costs. They consider either passenger disruption or delay cost.

Kohl et al. (2007) discussed developing a system that uses multiple resource methods, and integrated these resources to improve the quality of decision making. They indicated that developing flexible tools must be considered in research to have added-value contribution in the businesses. They also concluded that emphasizing on finding optimum solution in the strict academic sense without weighting on the operational restrictions cannot be applicable in real situations.

Cauvin et al. (2009) proposed a multi-agent approach to the problem in a *disrupted* and *distributed* environment. Their framework proposed a way for describing the existing methods for managing

disruption. In this framework, it was necessary to identify the actors, their interactions and their consequent activities in the disruptive environment.

Disruption management in airline industry was increasingly active during the last decade, but in most of the cases the proposed solutions consider just one aspect of the problem, e.g. aircraft type, crew, passenger, etc. This is an important field of research because there is a fundamental gap between the proposed prototype tools by software companies and the ideal integrated recovery tool (Clausen et al., 2010).

A model for estimating the number of required reserve crews for covering aircraft delays callout was presented by Gaballa (1979). he minimized costs of both reserve crews and overnight delays. The application of this method resulted in a considerable cost saving at Qantas Airways.

Another example of disruption management system is an automated system that has been implemented at US Airways. This system constructs an optimal scheduling for reserve crew by emphasizing on making good reserve bid lines (Dillon and Kontogiorgis, 1999).

Wei et al. (1997) developed a modeling framework for the crew reassignment by using a heuristic branch-and-bound search algorithm. Their proposed algorithm was more flexible in comparison with the traditional operational research algorithms. They engaged the business rules to bound the solutions.

Lettovský et al. (2000) claimed that it is necessary to reduce the complexity of the problem for crew reassignment during the operations. They applied the fact that the published schedule is optimum and by using a tree-based data structure they generated the integer solutions in a short time.

RESOPT (reserve optimization) is a model developed by Sohoni et al. (2006) which effectively increases reserve availability. The model needs a good estimator for open-time reserve demand to be used as a reserve manpower controller.

Another automated decision support tool is developed by Abdelghany et al. (2004). The tool can be used in large-scale commercial airlines that use the hub-spoke network structure for crew recovering problem. A hub-spoke network is a network in which all the points are connected through spokes to the hubs instead of a point-to-point connection. This tool is flexible to different scenarios and can proactively manage the future disruptions in a chain.

2.2 Classification and Regression Tree

Classification and regression trees was presented by Morgan and Sonquist (1963) as an automatic interaction detection technique. Two decades later Breiman et al. (1984) developed the first modern and comprehensive algorithm for growing trees. Their famous method CART is a fundamental basis for classification and regression trees and the book of Breiman (1993) on the classification and regression trees is a classic reference.

For a long time, classification and regression trees (CART) have been popular for modeling and predicting among statisticians, machine learning experts and data mining practitioners. Like many other methods, these tree-based models are used when there is a response variable, Y, and some predictors $X_1, X_2, \dots, X_p$. Regression tree is used when the response variable is a real number while the usage of classification tree is for the cases with a categorical response. In this section, a brief introduction to classification tree is presented with a review on the literature. We consider here just binary decision tree i.e. the decision trees with a two level response variable. Although decision trees with n-level response variables, called n-ary decision tree, exist in theory and practice, we don't consider this part of literature because the nature of our problem is binary, the pilots are either absent or present. For further reading and an in-depth discussion of this subject see Chapter 9 of Hastie et al. (2009), Chapter 6 of Wittenet al. (2011), and the classic textbook of Breiman et al. (1984).

2.2.1 Growing the tree

The main idea of tree-based models is to partition the variable space into non-overlapping rectangles and fit a constant in each partition. It is a simple but powerful idea, since any function can be approximated by piece-wise constant (step) function. Furthermore, constant model is computationally fast to fit. The fitting process requires only averaging over the response variable of observations belongs to that partition.

Let's consider an example with two explanatory variables X_1 and X_2, each taking values in the unit interval and a class response variable, Y, which is either *True (T)* or *False (F)*. As part *(a)* of Figure 2-1 shows, it is possible to partition the space of X_1 and X_2 so that in each subspace there exists just one kind of response: True or False. However it is difficult to determine the boundary of each region.

A partitioning like the one in Figure 2-1 *(b)* and its related tree structure, as shown in Figure 2-1 *(c)*, is acceptable and applicable. The CART methodology has been developed for simplifying the solution of this problem in an acceptable time and an efficient way with the minimum error.

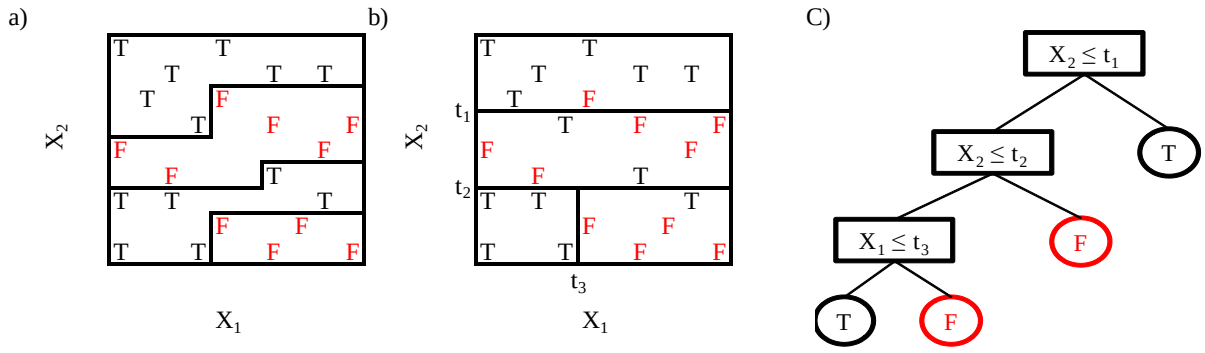


Figure 2-1: Partitioning and CART. General partitioning, (a), CART partitioning, (b), and its tree representation (c).

Consider $\boldsymbol{X}$ as the set of inputs, $X_1, X_2, \dots, X_p$, and Y as the binary response variable. The goal is to find a step function like

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m \mathrm{I}(\mathbf{x} \in \mathrm{R_m})$$

where M is the number of subspaces, R_m is a subspace of the space of inputs, $\boldsymbol{X}$. Here, c_m is the estimated constant, approximating the response variable Y in the region R_m, and $\mathrm{I}(.)$ is the indicator function

$$I(x \in R_m) = \begin{cases} 1, & \text{if } x \in R_m \\ 0, & \text{otherwise.} \end{cases}$$

For explaining the CART algorithm, let's start with all data, for each splitting variable j and each split point s we can partition the space into two subspaces,

$$R_1(j,s) = \{\boldsymbol{X} | X_j \leq s\} \text{ and } R_2(j,s) = \{\boldsymbol{X} | X_j > s\}.$$

Among all the input variables j and split points s, choose the pair that

$$\min_{j,s}\left\{\sum_{x_i\in R_1}(I(y_i=\text{TRUE})-\hat{c}_1)^2+\sum_{x_i\in R_2}(I(y_i=\text{TRUE})-\hat{c}_2)^2\right\}$$

where

$$\hat{c}_m=\frac{1}{N_m}\sum_{x_i\in R_m}I(y_i=\text{TRUE})$$

and N_m is the number of observations in partition region R_m. Now it is possible to partition the root node into two subspaces and repeat the process for each child node.

2.2.2 Splitting and Stopping Criteria

A decision tree has a hierarchical top-down order and at each node just one variable splits the space of inputs. In decision trees, the algorithm chooses a variable and a splitting point at each iteration by using an impurity measure. The splitting criteria with the stopping rule make the growing phase of the decision tree. There are two types of splitting criteria, univariate and multivariate.

Univariate splitting criteria create each node using just one variable, i.e. the discrete splitting function is univariate. These are the well known criteria that are used in almost all the famous algorithms. *Information Gain* (Quinlan, 1987) uses entropy measure as the impurity measure. *Gini Index* (Breiman et al., 1984) is the divergence measure of the probability distribution of response variable. *Likelihood ratio chi-square statistic* (Ciampi et al., 1987) in addition provides statistical inference about the information gain. Normalizing information gain by the entropy, leads to the *Gain Ratio* (Quinlan, 1993). *Twoing Criteria* (Breiman et al., 1984) is the same as Gini Index for binary response model and more accurate generation of it for multi-level response variable.

Multivariate splitting criteria create each node by a linear combination of variables (Breiman et al., 1984) and (Sethi and Yoo, 1994). It is obvious that finding the best solution is more complicated, so multivariate splitting criteria are less popular in practice.

Another important criterion for making a tree is the stopping criteria which are applied at the end of the growing phase. Common stopping criteria are the following (Rokach and Maimon, 2005):

- In each terminal node there exists just a single value of the response.

- The maximum tree depth reaches a pre-specified limit
- Number of cases in the node is less than a pre-specified value
- The best splitting criteria is not greater than a pre-specified threshold.

After a tree is made by using a stopping rule, it is pruned to keep the balance between bias and variance of the model for a better prediction.

2.2.3 Pruning Methods

One of the problems that occurs after growing the tree is the over-fitting, i.e. the training accuracy is high while prediction accuracy is low. In other words, the tree models the data set perfectly, but the fitted model does not work properly for predicting new observations. The reason of this problem is the over-complexity of the model and for simplifying it, pruning is necessary (Bohanec and Bratko, 1994).

There are two types of pruning. First, when it is a part of the tree construction and it is called *pre-pruning*. Second, if the pruning is a separate procedure after the growing phase it is called *post-pruning* (Esposito et al.,1997).

There are many pruning methods (Rokach and Maimon, 2005). Among them *cost-complexity pruning (cp)* is the most popular one. The *cp* is a post-pruning method proposed by Breiman et al. (1984). In this method, all trees extracted from the original one T_0 to the root T_k are created and the best pruned tree is selected with considering the estimation of generalization error.

Let T be a subtree of T_0 which is obtained by pruning T_0. If $|T|$ denotes the number of terminal nodes in T and R_m represents the region that is related to node m. The cost pruning complexity criteria is defined as (Hastie et al., 2009):

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|,$$

Where Q_m is the proper impurity measure and α is a non-negative constant. The first part of the equation is the goodness of fit for the model and the tuning parameter α is used for governing a trade off between this goodness of fit measure and the tree size. The larger the value of α is, the smaller the tree will be.

Usually, CART method for binary response uses one of the following functions as the impurity measure (Hastie et al., 2009):

- Misclassification error: $1 - \max(p, 1-p)$,
- Gini Index: $2p(1-p)$,
- Cross-entropy: $-p\log(p) - (1-p)\log(1-\mathrm{p})$,

where, p is the proportion in the True class for the binary trees. Gini index and cross-entropy are most often used in practice because they are differentiable and also more sensitive to changes in node probabilities.

The *Minimum Error Pruning* is another procedure developed by Niblett and Bratko (1987). This is a bottom-up pruning method i.e. it first checks the internal nodes at the bottom of the tree. The pruning measure is a correction to the simple probability estimation of errors. The tree is pruned at the node that gives the minimum error overall.

Quinlan's *Pessimistic Pruning* (Quinlan, 1993) uses the continuity correction for binomial distribution as the error estimation. An evolution of this method, called *Error-Based Pruning*, is used in the well-known tree making algorithm C4.5. Bohanec and Bratko (1994) introduced an *Optimal Pruning* algorithm. They use the initial tree, T_0, and a measure of accuracy based on the complexity and size of the tree. The accuracy of all the possible pruned trees are calculated and the smallest pruned tree with an accuracy greater than the minimal accuracy is selected as the optimal tree.

2.2.4 Tree Algorithms

Decision tree making needs a lot of computations, and therefore it must have efficient computing algorithm. These make decision tree an interdisciplinary area of research between statistics and computer science. At the same time, statisticians develop the mathematical foundations and program and algorithm developers search for the most efficient algorithms. Some of the famous and most frequently used algorithms are mentioned in the following.

Early decision tree algorithms were in the field of *Automatic Interaction Detection* (*AID*) in the sixties and seventies. *CHAID* or *CHi-square AID* (Kass, 1980) originally handles nominal attributes. For each attribute, CHAID splits its range at the point having the most significant

difference of the response variable. This algorithm uses an F-test, Pearson chi-square or likelihood ratio test for continuous, categorical or ordinal target attribute, respectively.

CART (Breiman et al., 1984) algorithm uses the *Twoing Criteria* as the splitting criterion and the *cp* as the pruning method. It is also able to handle regression trees and this is one of its most notable features.

A very simple decision tree is *ID3* proposed by Quinlan (1986). This algorithm does not have any pruning method. It splits the input dataset according to Information Gain until either best information gain is negative or all the samples in nodes belong to one category.

Quinlan later developed this algorithm as *C4.5* (Quinlan, 1993) with the Gain Ratio as the splitting measure. The algorithm stops growing when the number of splitting samples is less than a pre-determined threshold. After that the growing tree is finished, pruning based on the prediction errors starts.

In the past years, by developing the computation facilities and memory storage, the amount of collected data has grown. Decision trees accordingly must be capable of dealing with such massive data. Chan and Stolfo (1997) suggested a method of partitioning large dataset into several disjoint datasets, and load each of them separately into the memory for inducing the tree. *SLIQ* (Mehta et al., 1996) is an algorithm that does not need to load the whole database into the main memory. *SPRINT* (Shafer et al., 1996) is a similar solution that creates decision tree quickly.

There are many other different approaches to classification trees, from *Bayesian CART* model (Chipman et al., 1998) to *Fuzzy Decision Trees* (Yuan and Shaw, 1995) and *Oblivious Decision Trees* (Almuallim and Dietterich, 1994), in which all the nodes at the same level tests the same attribute.

2.3 R: a Statistical Programming Language

R (R Development Core Team, 2005) is a free software environment for statistical computing and graphics. It was created by Ross Ihaka and Robert Gentleman as a non-commercial of S programming language. Its name indicates the first letter of the first name of the two authors and also a play with the previous statistical computing language S.

R is widely used among statisticians, especially for the academic purposes. The characteristics of R make it an appropriate environment for doing all kind of statistical analyses. Data handling and storage in R are efficient. Operations for calculations on matrices and arrays are fast and easy. Moreover, R has a large and up to date library of packages for implementing the most recent statistical techniques as well as all the classical methods. A wide range of graphical facilities makes it appropriate for data visualization. Its programming language is object-oriented, simple, and efficient. It has a big community of developers all around the world. (Venables et al., 2002)

We use R as the main programming language in this study and therefore some R packages have been used. Here is a list of these packages with a brief introduction to each one.

The *reshape* package (Wickham, 2007) is a powerful tool that makes reconstructing and aggregating data flexible. By using this package, it is possible to change structure of the databases and create pivot table.

The *RODBC* package (Ripley, 2012), is an R package for open database connectivity. This package provides access to different database formats such as Microsoft Access and Microsoft SQL.

The *ggplot2* package (Wickham, 2009) is the R grammar of graphics and it gives a new elegant way of plotting. It provides a way to create multi-layered graphics and complex plots.

The *rpart* package (Therneau, Atkinson, and Ripley, 2010) is a comprehensive package for classification and regression trees. It provides different tools for growing the tree, testing the results, and pruning the resulting tree. The complementary package *rpart.plot* (Milborrow, 2012) plots legible trees with many useful options.

The *gWidgets* package (Verzani, 2012) is an application programming interface for writing graphical user interfaces within R. This package is useful for making widgets for automating R programs.

The *shiny* package (RStudio and Inc., 2013) is a library for creating web applications. It provides an interactive interface for presenting R graphics and tables. By using this package it is possible to automate R codes.

CHAPTER 3 PROBLEM DESCRIPTION

In this chapter, we review description and the importance of the problem from scientific and business viewpoints in Section 3.1. Then, we briefly explain the tables that are available for analyse and give some examples about the differences between schedule and operations pairings in Section 3.2. And finally, in Section 3.3, we introduce pairing characteristics that can be considered as the variables of the final model.

3.1 Problem Overview

Airline companies face many sort of disruptions during operations and they have to spend millions of dollars each year for disruption management. From weather condition to security issues, the airline industry is involved with many uncontrolled situations that may cause changes in their schedule. Flight delays, flight cancellation, passenger dissatisfaction, etc. are few of such challenges.

The loss of budget due to such disruptions has forced airlines to have strategic cost-saving plans for reducing these losses. Airline disruption sources are inevitable, but by different mathematical and engineering tools it is possible to have a continuous improvement in minimizing the loss.

One of the ways of improving decision tools is scientific prediction of future random events. Although, in any situation and for any random event it is impossible to have an exact prediction, having a systematic prediction covers a portion of the current uncertainty. Another reason of using prediction in the industry is related to the nature of the decision-making systems. Having a good prediction causes reductions in the number of decisions that must be taken at the operation level, faster and less optimal in comparison with the tactic level decisions.

In airline industry cabin crew or pilots are especially important. Without them a flight is not even imaginable. It is difficult to replace a pilot with another one because every pilot is qualified for just one position. However, during the operations there are different situations in which a block holder, must be replaced by a reserve. The main reason of these replacements is the pilots' absenteeism caused by sickness. This must be indicated that by sickness we mean both real or fake calling sick.

For each position, the number of pilots on reserve for a block month must be determined in advance, in order to cover the needs of replacement. Obviously, the number of reserve pilots depends on the

monthly schedule. In high seasons, when there are more travel demands, airlines encounter more flight hours so the number of reserve pilots should be more than low seasons. The question then is: how many hours of reserve pilots an airline must consider for a published schedule? In this study we focus on those replacements related to sickness of pilots. The main objective in this thesis is attacking this problem and developing a decision support system for predicting pilots' sickness.

Calling sick depends on many different environmental and personal conditions. However, in this study we do not consider these conditions as the factors for modeling because the prediction must be done exactly after publishing pairing schedule and at that time pilots have not been assigned to the pairings (see Figure 3-1). Therefore, we can only include pairing characteristics as the covariates of the model. The advantages of this approach are the improvement of the prediction and ability of distinguishing mass regions on the space of attributes according to the sick events. This means if there exists undesirable characteristics in pairings that some pilots prefer to use their yearly paid sick days rather than fly, this model is able to distinguish those characteristics.

This is an applied research combined with methodological adjustments and its results have been implemented in a real airline company. The datasets under study have the airline company's format and in the following sections and chapters we describe how the method can be used in other similar companies with minor modifications. It must be considered that many airlines use the same operational procedures and business terms.

3.2 Data Description

3.2.1 Schedule Table

Figure 3-1 shows a simplified flowchart of airline operations from the scheduling phase to the end of the operations. The main objective for our system is predicting monthly total sick hours of pilots in a position after publishing the pairings schedule. This is based on the new month published schedule and by using previous records of the sickness in the past months. Prediction must be implemented without the information about the bidding results (work schedule in Figure 3-1).

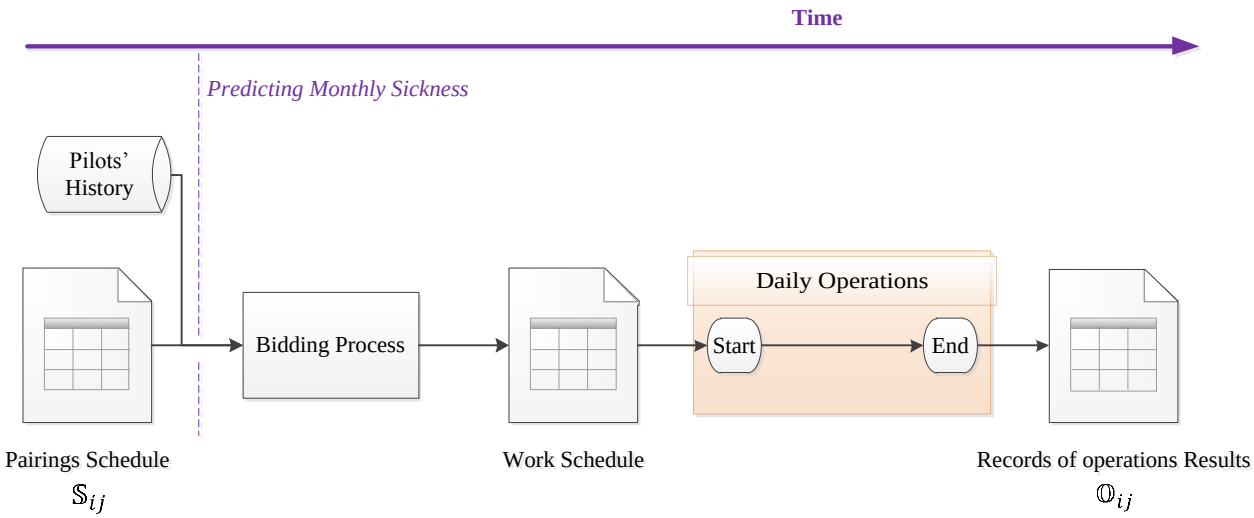


Figure 3-1 : Simplified airline process from scheduling to the end of monthly operations. Our monthly sickness must be done after publishing pairing schedule and before bidding process.

Let $\mathbb{S}_{ij}$ denotes the table of pairings schedule for position *i* in month *j*. Each record in this table is a pairing with its characteristics or attributes. A *pairing number* and *pairing start date* uniquely determine each individual pairing and a combination of these attributes can be used as the *pairing unique code* for distinguishing each individual in the table. In the pairing schedule table, as shown in the Figure 3-1, there is no information about the pilots who will operate each pairing. Pilots bid on this published table and after the bidding period, based on the results of the PBS, monthly work schedule will be published.

3.2.2 Schedule changes during operations

The airline industry is one of those industries where disruptions have a big effect on its schedule and it is almost impossible to operate a pairings schedule without imposed changes. Therefore, all the pairings and flights characteristics have an indicator that determines the attribute belongs to either scheduling phase or operating phase. Schedule attributes indicate planned departure and arrival date and time, flight duration, etc while operated attributes indicate how exactly these flights and pairings have been executed.

Here, we explain two examples of these changes during the operations. The details of these examples show the possibility of changes in every pairing and flight attribute and the necessity of creating different tables for schedule and operations.

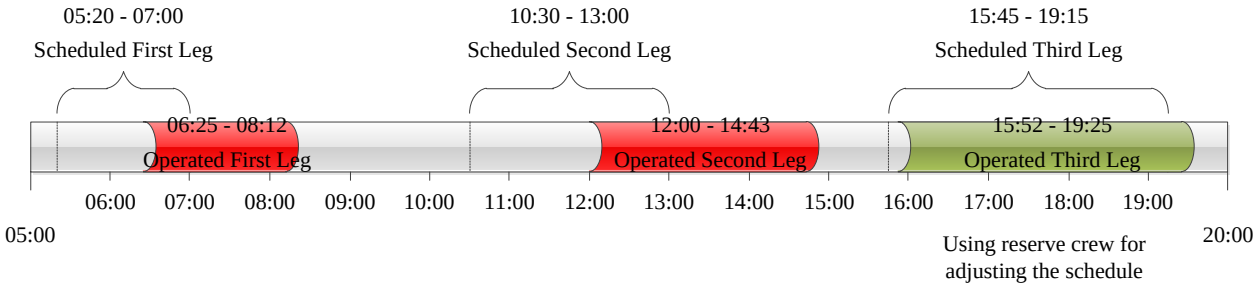


Figure 3-2: Change of scheduled pairing due to a flight delay. First two legs of the pairing were operated by delay and for adjusting the schedule pilot of the third leg was operated by a reserve pilot.

Figure 3-2 and Table 3.1 describe a possible example of changes in schedule because of delay. The pairing is planned for 3 legs. It starts from city A and passes cities B and C and ends in the base of the pairing, City A. The awarded captain for this pairing is ID0015. In the operations, the first two legs of the pairing have been operated with delays. As the second leg arrived later than scheduled (it arrived at 14:43 instead of 13:00, see Arrival time column of Table 3.1), pilots did not have enough resting time between the second and third legs. In this case operations manager had to decide either to operate the third leg with a delay or to change the flight crew. The decision was to change the crew because the crew status, in the second part of Table 3.1, for third leg is deadhead and it is different from the scheduled crew status. Reserve crews operated the last leg of the pairing and the scheduled crew were in the flight as deadhead pilots for coming back to their base. The details and changes are shown with red font in Table 3.1.

Table 3.1: An example of change of scheduled pairing due to a flight delay.

Schedule									
Flight Number	**Departure**	**Arrival**	**Captain ID**	**Departure Date**	**Departure time**	**Arrival Date**	**Arrival time**	**Duration**	**Crew Status**
110	A	B	ID0015	29/09/2013	5:20	29/09/2013	7:00	100	pilot
240	B	C	ID0015	29/09/2013	10:30	29/09/2013	13:00	150	pilot
375	C	A	ID0015	29/09/2013	15:45	29/09/2013	19:15	210	pilot
Column in GREEN fills after bidding process									
Operations Records									
Flight Number	**Departure**	**Arrival**	**Captain ID**	**Departure Date**	**Departure time**	**Arrival Date**	**Arrival time**	**Duration**	**Crew Status**
110	A	B	ID0015	29/09/2013	6:25	29/09/2013	8:12	107	pilot
1420	B	C	ID0015	29/09/2013	12:00	29/09/2013	14:43	163	pilot
375	C	A	ID0015	29/09/2013	15:52	29/09/2013	19:25	213	deadhead

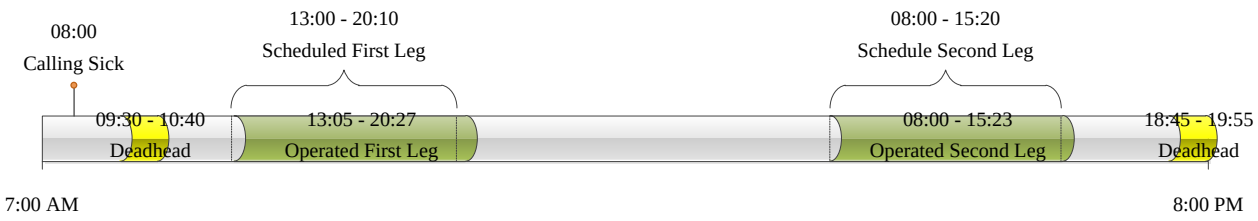


Figure 3-3: Change of scheduled pairing because of sickness. The block holder captain called sick, as there has not been reserve pilot in the pairing base, a reserve pilot has been transferred to the base of the pairing.

Another example of schedule change is shown in Figure 3-3 and Table 3.2. In this case, the block holder captain (ID0224) called sick before the pairing. Therefore, the same rows of the schedule for this pairing (part 1 of Table 3.2) have been created in the operations' record table (part 2 of Table 3.2). The only difference is in the crew status column which indicates calling sick happened during operations. The pairing was based in city Y and there were no appropriate reserve captain in that city and the operations manager moved a reserve (ID0754) from another city (X) as deadhead to operate this pairing. At the end of the pairing, the reserve captain returns to his base as deadhead. In Table 3.2 the details of this scenario is illustrated.

Table 3.2 : An example of change in scheduled pairing because of sickness.

Schedule									
Flight Number	**Departure**	**Arrival**	**Captain ID**	**Departure Date**	**Departure time**	**Arrival Date**	**Arrival time**	**Duration**	**Crew Status**
857	Y	Z	ID0224	29/09/2013	13:00	29/09/2013	20:10	430	pilot
444	Z	Y	ID0224	30/09/2013	8:00	30/09/2013	15:20	440	pilot
Column in GREEN fills after bidding process									
Operations Records									
Flight Number	**Departure**	**Arrival**	**Captain ID**	**Departure Date**	**Departure time**	**Arrival Date**	**Arrival time**	**Duration**	**Crew Status**
857	Y	Z	ID0224	29/09/2013	13:00	29/09/2013	20:10	430	sick
444	Z	Y	ID0224	30/09/2013	8:00	30/09/2013	15:20	440	sick
Operations Records									
Flight Number	**Departure**	**Arrival**	**Captain ID**	**Departure Date**	**Departure time**	**Arrival Date**	**Arrival time**	**Duration**	**Crew Status**
105	X	Y	ID0754	29/09/2013	9:30	29/09/2013	10:40	70	deadhead
857	Y	Z	ID0754	29/09/2013	13:05	29/09/2013	20:27	442	replaced
444	Z	Y	ID0754	30/09/2013	8:00	30/09/2013	15:23	443	replaced
248	Y	X	ID0754	30/09/2013	18:45	30/09/2013	19:55	70	deadhead

3.2.3 Operations' Record Table

The records of operations are published in a table called *operations' record*. Let $\mathbb{O}_{ij}$ denotes the operations' record for position *i* and month *j*. When some uncontrolled and unpredicted events happen during the operations and the scheduled pairing breaks, the operations managers and their systems consider the problem flight-wise rather than pairing-wise as in the planning phase. This is the reason that in the operations' record table, each record of this table is a flight, and flights must be considered as individuals.

The only information used from the operations' record table in this study is the *sickness indicator*. As the predictions must be based on the scheduled pairings, all the other attributes that are used in modeling and predicting comes from schedule tables $\mathbb{S}_{ij}$. That means, in this study, sickness in a flight is equal to scheduled flight duration if the corresponding planned pilot is reported sick in the operations' record table. For example in the explained case of Table 3.2, time of sickness is considered 430 minutes and 440 minutes for the first leg and for the second leg, respectively. The same as flight scheduled duration (ninth column of the first part of Table 3.2) and not their actual and operated duration (ninth column of the third part of Table 3.2).

3.3 Pairing Characteristics and attributes

In airline industry, pilots bid on the pairings and based on the honouring seniority, it is possible to have some undesired pairings for some pilots in a monthly work-schedule. Therefore we may expect that pairing characteristics have a big effect on their choices. Some pairings are for 4 or 5 consecutive days and some others are for just one day. Some of them contain a lot of deadhead credit and some happen during the weekend and holidays. As there is no available information about the pilots' preferences in this study, we consider pairing characteristics as the only covariates of the model to extract their hidden information during the analysis. A list of these attributes and a brief explanation for each of them is presented in Table 3.3.

Table 3.3: List of the attributes

Row	Attribute	Attribute Name	Description	Type of Attribute
1	UC	Unique Code	Distinguish uniquely each pairing for one pilot.	ID
2	BP	Bid period	Pairing block month.	ID
3	P	Position	Scheduled position of the pairing.	ID
4	LN	Leg Numbers	Number of legs in a pairing.	Integer
5	TC	Total Credit	Total flight minutes during a pairing.	Numeric
6	NC	Night Credit	Night flight minutes during a pairing.	Numeric
7	DC	Day Credit	Day flight minutes during a pairing.	Numeric
8	DH	Deadhead Credit	Flight minutes credited to a pilot to be deadhead during a pairing.	Numeric
9	R	Return to base	Number of legs in a pairing with the same departure as the base.	Integer
10	TT	Total time	Total time of the pairing from departure time of first leg to arrival time of the last leg.	Numeric
11	B	Base	Base of the pairing, departure city of the first leg.	Nominal
12	AS	Actual Seat	The role of the pilot in the flight.	Nominal
13	WT	Weekend Time	Percentage of the pairing's total time that passes during weekend.	Proportion
14	M	Month	Month of the year that pairing belongs to.	Date
15	SH	Start Hour	Hour of day in which pairing starts	Nominal
16	EH	End Hour	Hour of day in which pairing ends	Nominal
17	SD	Start Day	Week day in which pairing starts	Nominal
18	ED	End Day	Week day in which pairing ends	Nominal
19	W	Week	Week of the year that pairing belongs to	Nominal
20	s	Scheduled flying time for sick pilot	Total credit of the pairing in which pilot was sick.	Numeric
21	y	Sick indicator	Was pilot sick during the pairing?	Logical

CHAPTER 4 METHODOLOGY

In this chapter we represent the proposed methodology for predicting sickness. Data pre-processing methods are explained in Section 4.1. In Section 4.2, calculating sickness hours based on a decision tree is presented and the complexity of selecting the best level of the tree is illustrated by an example. The learning process for making a decision tree is presented in Section 4.3 and in Section 4.4 the predicting algorithm is explained.

4.1 Data pre-processing

4.1.1 Merging Tables and Data Cleaning

As explained in Section 3.2, for each position and each month we have 2 tables, pairing schedule, $\mathbb{S}_{ij}$, and operations' record, $\mathbb{O}_{ij}$. The pairing characteristics, which were defined as the variables in the model, are in the pairing schedule table and the sickness indicator is a variable of operations' record table. Hence, we need to merge these two tables for adding sick information to the schedule table.

The steps of the merging process are shown in Figure 4-1. The pairing schedule table is a pairing-wise table and the operations' record table is a flight-wise table so, for merging these two tables we first need to match the individual of these tables. In the first step, the pairing schedule table, $\mathbb{S}_{ij}$, is extracted to its flight legs and we obtain Table 1, which is a flight-wise schedule table. In the second step, we merge Table 1 and the operations' record table flight by flight and obtain Table 2. In the third step, sickness calculation sub-process is applied to Table 2 for adding flight sick minutes, the scheduled flying minutes with a pilot reported as sick. The result is Table 3 which is used in step 4 for the summing-up sub-process. At the last step, the table will be summarized so that each individual of the table is a pairing and each attribute is a pairing characteristic. The final table is our main database and is denoted by $\mathbb{D}_{ij}$.

The reason for merging is adding a sick indicator to the pairings schedule. We remove all other operational information of the table as a data cleaning step to improve the speed of running statistical codes later on. Afterwards, the sickness for each flight will be calculated based on this database. In the next section we explain the sickness calculation step.

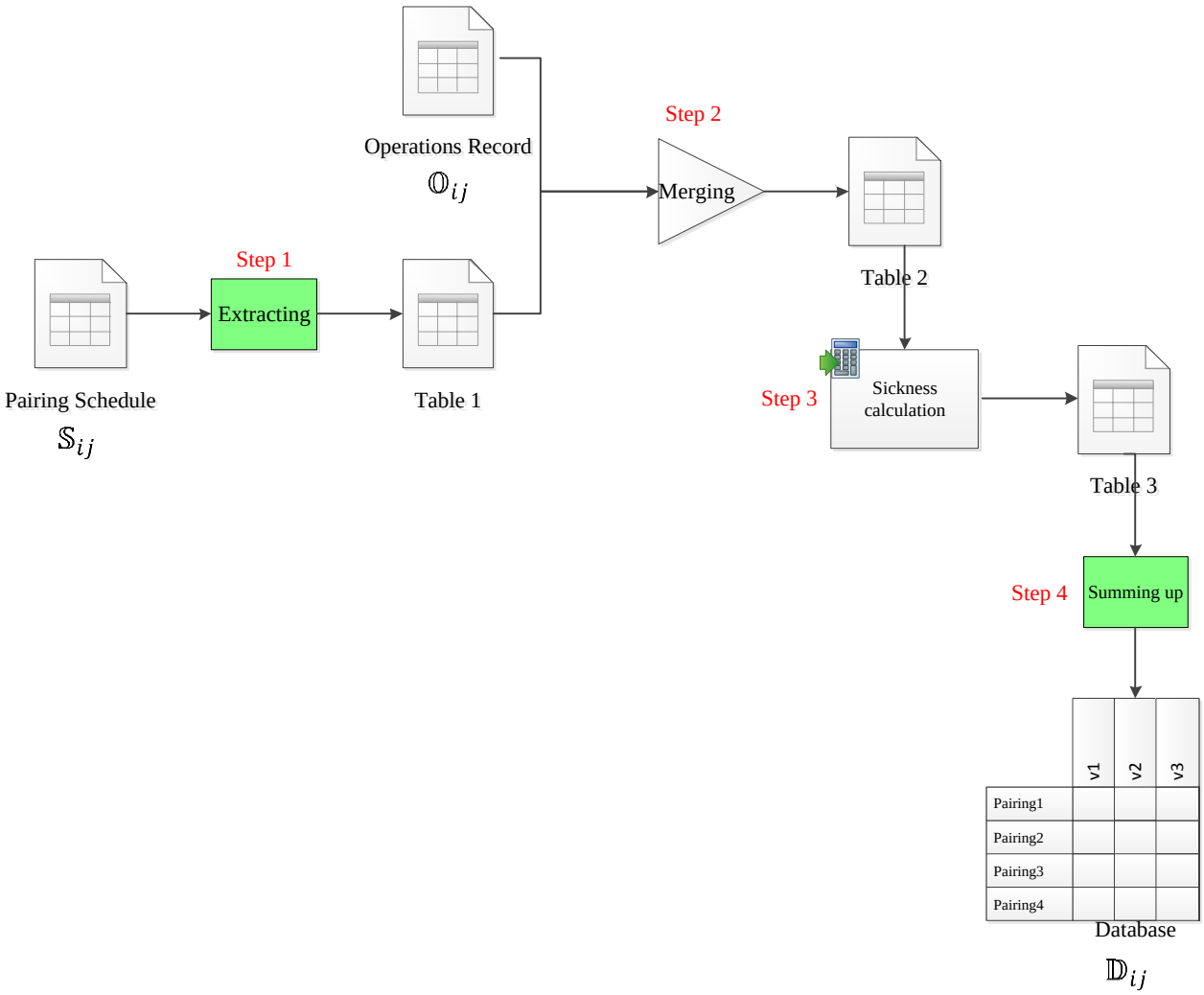


Figure 4-1: Data pre-processing steps. Step 1 is extracting pairing schedule table to its flights. Step 2 is merging obtained table with operations' record table. In step 3, sickness calculation applies to Table 2 and its result is Table 3. Finally by summing up for each variable in Table 3 over each pairing we obtain the final table.

4.1.2 Sickness Calculation and Sick Attributes

We show flight related attributes with upper-case letters and pairing related attributes with lower-case letters, i.e. C denotes total credit of a flight while c denotes the total credit of a pairing.

Suppose that $\mathbb{S}_{ij}$ has K_{ij} pairings $1,2,\dots,K_{ij}$ and pairing k has L_k legs $1,2,\dots,L_k$. After adding the sick indicator, I_{ijkl}, to the schedule pairings table, flight sick minutes, S_{ijkl}, equals to the schedule total credit of the flight (C_{ijkl}) if the pilot in the flight was reported as sick and 0 otherwise, i.e.

$$S_{ijkl} = \begin{cases} C_{ijkl}, & \text{if } I_{ijkl} \text{ is true} \\ 0, & \text{otherwise.} \end{cases} \quad (4\text{-}1)$$

Sick minutes for a pairing is equal to the sum of flight sick minutes, S_{ijkl}, over all its legs. Hereafter s_{ijk} indicates the sum of the flying minutes that have been planned for a pilot during pairing *k* of pairings schedule table for month *j* and position *i*

$$s_{ijk} = \sum_{l=1}^{L_k} S_{ijkl}. \quad (4\text{-}2)$$

And finally the total sickness in month *j* for position *i* is the sum of pairing sickness over all its pairings

$$s_{ij} = \sum_{k=1}^{K_{ij}} s_{ijk}. \quad (4\text{-}3)$$

4.2 Decision tree and its levels

In every data mining study, it is necessary to visualize data and provide some descriptive statistics for having a general idea about the structure of the data. Sometimes visual representation of the data helps a lot in understanding the important information in the data or leads to the appropriate method of analyse. We first started working on these statistics to get familiar with the data, some of the most important and useful data visualization techniques, which we used in this study, are proposed later in Section 5.1.

As it has been explained in the previous section, we have two main databases for predicting new month sickness. Let *n* merged tables be available for position *i*, $\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{in}$. After publishing new month pairing schedule ($\mathbb{S}_{i\,n+1}$), a suitable method for prediction must be able to predict total sickness hours for this new month ($s_{i\,n+1}$).

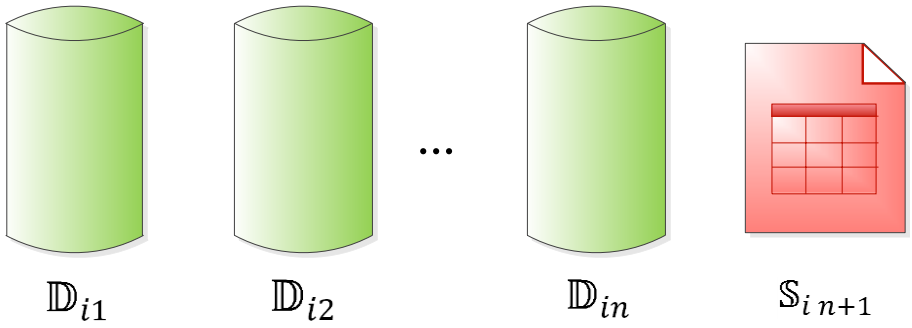


Figure 4-2: Available datasets for predicting new month sickness. All the previous databases ($\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{in}$) and the schedule of new month, ($\mathbb{S}_{i\,n+1}$), can be used in prediction.

In each of the datasets $\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{in}$, the response variable is defined as the *sick indicator*. This variable is denoted by y_{ijk} which indicates the sickness in pairing *k* of position *i* during month *j* and is a binary variable.

$$y_{ijk} = \begin{cases} 1, & \text{if } s_{ijk} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4-4}$$

Let $\mathfrak{T}$ be the complete decision tree obtained by using all the datasets $\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{in}$, where the sick indicator is the response variable and the pairing characteristics (Table 3.3) are the explanatory variables. If $\mathfrak{T}$ has *m* terminal nodes which represent *m* regions on the space of pairing characteristics as $\mathcal{r}_1, \mathcal{r}_2, \dots, \mathcal{r}_m$ and the assigned probability of sickness in each region is $\mathcal{p}_1, \mathcal{p}_2, .., \mathcal{p}_m$; then the estimation of sickness for the new month can be calculated as

$$\hat{s}(\mathfrak{T}, \mathbb{S}_{i\,n+1}) = \sum_{k=1}^{m} \mathcal{p}_k c_k, \tag{4-5}$$

where c_k is the sum of total credits in *k* th region of the pairing schedule of the new month.

This is our proposition for estimating the monthly sickness based on a decision tree and a published pairing schedule. Like any other decision tree, a question arises: how deep must the decision tree be or at which level the decision tree must be pruned?

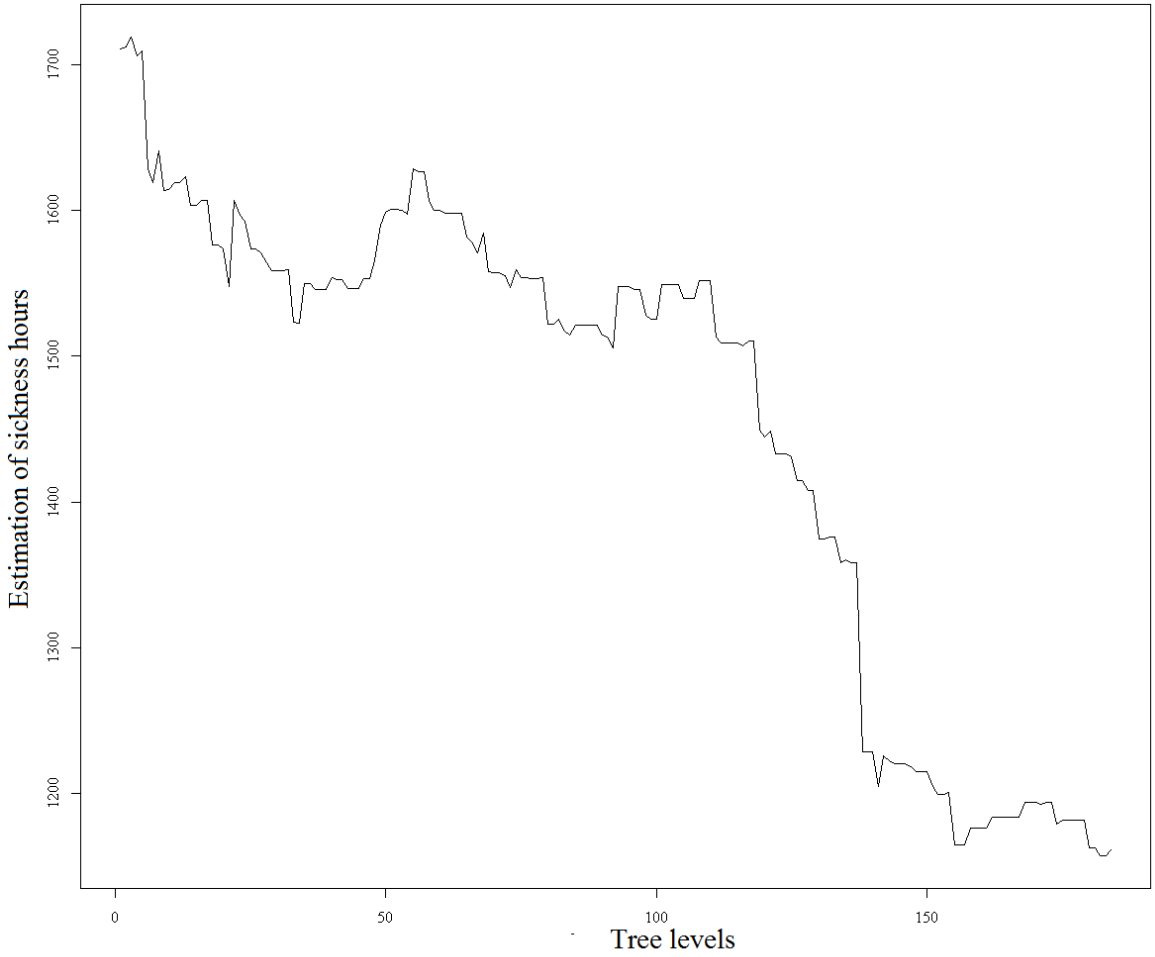


Figure 4-3: Estimation of sickness hours in different levels of the tree. Different values in different levels, in a wide range of variability, make it difficult to prune the tree correctly.

Figure 4-3 shows an example of a tree and the sickness estimations in each level of the tree for a fixed monthly pairing schedule. The estimation of sickness is calculated by pruning the tree at each level and using Formula 4-5. As it can be seen in the figure, the differences between prediction values are big (with a range of more than 500 hours).

For solving this problem, we propose a learning process in which the algorithm of tree growing and tree pruning will be explained.

4.3 Learning process

First, we briefly describe the process and then, in the following subsections, we go into the details for each step of the learning process. Our objective is to create a decision support system for

predicting new month sickness hours ($\hat{s}_{i\,n+1}$) based on the new month pairing schedule ($\mathbb{S}_{i\,n+1}$), pairing characteristics, and sickness history ($\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{in}$). The proposed learning process suggests using a loop for choosing the best decision trees.

In this loop, we start with fixing parameter *a*, which is the number of consecutive months needed to obtain the first *stable* tree. Then, we merge datasets $\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{ia}$ into a unique set, denoted Γ_{ia}. In the first step of the loop, a decision tree is made for Γ_{ia} and the predictions of sickness hours are calculated for each level of this tree by using Equation (4-5) and $\mathbb{S}_{i\,a+1}$ as the input. Then the tree is pruned at the level that gives the minimum *level error* for the month $a+1$. The pruned tree is called $\widetilde{\mathfrak{T}}_{i\,a}$.

This loop will continue monthly to obtain $(n-a)$ pruned trees $\widetilde{\mathfrak{T}}_{i\,a}, \widetilde{\mathfrak{T}}_{i\,a+1}, \dots, \widetilde{\mathfrak{T}}_{i\,a+n-1}$.

4.3.1 Tree growing method

It is necessary to have a stable tree for starting the algorithm. A tree is considered stable when its associated rules do not change considerably and they are acceptable by the experts of company. We fix *a* which is the number of months that must be used for making such a decision tree. Let Γ_{ia} be the dataset obtained by merging $\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{ia}$.

In the airline industry, we suggest to set *a* equal to 12 to have a complete year of history for starting the construction of the tree. The reason for this choice is that explanatory variables (pairing characteristics), which are extracted from pairing schedule, are different in high and low seasons. Another reason for considering 12 months datasets as the preliminary training set is the fact that in the airlines flight schedules are planned at the tactics level for one year.

By using Γ_{ia} as the training dataset, we grow a decision tree. The Gini Index is used as the splitting measure and the stopping rule is applied when either the *cp* is less than a pre-defined threshold or a node is split into child nodes having a population that is less than a pre-defined percent of the number of pairings in Γ_{ia}. The *cp* criteria helps to keep the decision tree at a level where the variance of terminal node (leaves in the tree) is acceptable and the minimum population criteria avoids the creation of terminal nodes that represent very rare cases.

Another parameter being used for growing the tree is the *loss matrix,* same thing called *confusion matrix* in computer science literature, which is a 2×2 matrix with zero on the diagonal elements

and the misclassification cost rates on the off-diagonal elements. In our problem, the cost of misclassifying a sick pairing as non-sick is equal to odds ratio of not being sick. This means if the cost of misclassifying a non-sick pairing as sick is 1, we consider the cost of misclassifying a sick pairing as non-sick equal to $\frac{1-p}{p}$, where p is the proportion of sick pairings in Γ_{ia}. We can write loss matrix as the following

$$\begin{bmatrix} 0 & 1 \\ \frac{1-p}{p} & 0 \end{bmatrix}. \tag{4-6}$$

We use loss matrix because sickness is a rare event and the database of all pairings is large. In this case, the algorithm is not sensitive to small changes of sickness percentages in the child nodes and it returns the root.

Figure 4-4 represents the whole process in a flowchart. We repeat this process month by month to obtain $n-a-1$ trees, $\mathfrak{T}_{i\,a}, \mathfrak{T}_{i\,a+1}, \dots, \mathfrak{T}_{i\,n-1}$. Each of these trees is based on the merged datasets from the first month up to its related month. For example $\mathfrak{T}_{i\,a+1}$ is the decision tree that is obtained from the explained growing method by using $\Gamma_{i\,a+1}$, the merge of datasets $\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{i\,a+1}$, as the train dataset.

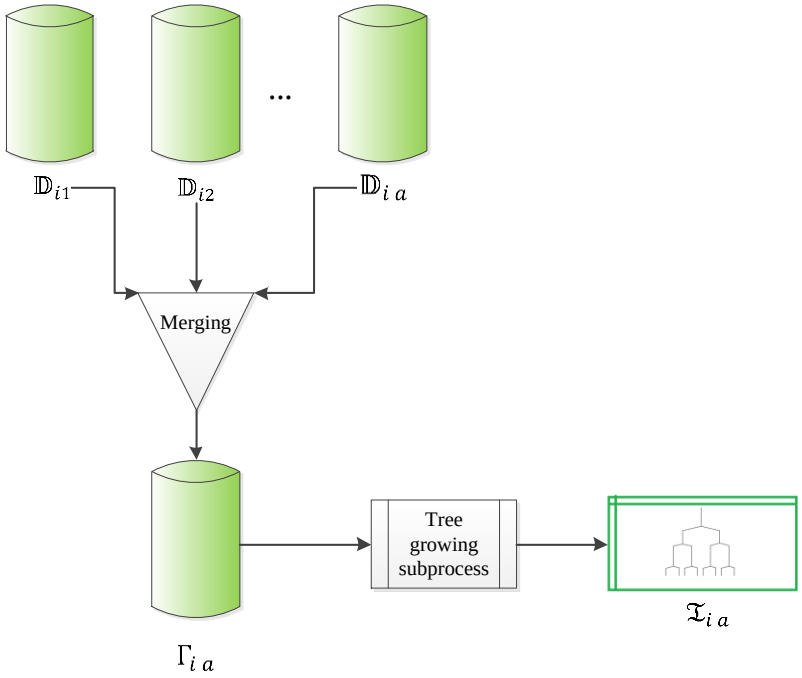


Figure 4-4: Tree growing process. Tree growing sub-process is applied to the result of merging all the databases ($\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{i\,a}$) and create the original decision tree $\mathfrak{T}_{i\,a}$.

4.3.2 Tree pruning method

In the previous step, we obtained $n-a-1$ trees, $\mathfrak{T}_{i\,a}, \mathfrak{T}_{i\,a+1}, \dots, \mathfrak{T}_{i\,n-1}$, each of them describes the association rules that increase or decrease the sickness proportion in its terminal nodes. We want to use them as the predictor of future sickness; therefore in this step of learning the process we prune each of them at the level that gives the best prediction for its following month. In this way, we obtain the best possible scenarios based on the pairing characteristics and on the sickness history and we apply these scenarios for the future.

Let start with $\mathfrak{T}_{i\,a}$ and suppose that it has m levels i.e. its related cp table has m cp values. We can obtain m different trees by pruning the original tree with respect to each of cp values. We denote the pruned tree at level k as $\langle \mathfrak{T}_{i\,a} \rangle_k$.

By using Equation (4-5) and the pairing schedule of the following month, i.e. $\mathbb{S}_{i\,a+1}$, it is possible to calculate the sickness prediction for the following month at each level of the tree, $\hat{s}(\langle \mathfrak{T}_{i\,a} \rangle_k, \mathbb{S}_{i\,a+1})$. The actual value of sickness in month $a+1$, $s_{i\,a+1}$, can be calculated by Equation (4-3). Moreover, the *level error* representing the error of prediction at level k of the original decision tree $\mathfrak{T}_{i\,a}$ is defined as follow

$$e_{i\,a\,k} = \begin{cases} \hat{s}(\langle \mathfrak{T}_{i\,a} \rangle_k, \mathbb{S}_{i\,a+1}) - s_{i\,a+1}, & \text{in the case of over-prediction} \\ \eta\{s_{i\,a+1} - \hat{s}(\langle \mathfrak{T}_{i\,a} \rangle_k, \mathbb{S}_{i\,a+1})\}, & \text{in the case of under-prediction'} \end{cases} \tag{4-7}$$

where η is the proportion of under-prediction cost on over-prediction cost. We consider this ratio because in the airline industry the cost of under-prediction is usually higher than the cost of over-prediction.

The *error* of the tree is defined as the minimum of level errors,

$$e_{i\,a} = \min_{1 \le k \le m} e_{i\,a\,k}. \tag{4-8}$$

We prune the decision tree $\mathfrak{T}_{i\,a}$ at the level where its level error is equal to $e_{i\,a}$ and denote it $\widetilde{\mathfrak{T}}_{i\,a}$. This procedure is applied for all $n-a-1$ trees and the obtained trees, $\widetilde{\mathfrak{T}}_{i\,a}, \widetilde{\mathfrak{T}}_{i\,a+1}, \dots, \widetilde{\mathfrak{T}}_{i\,n-1}$, are used for predicting new month sickness $s_{i\,n+1}$.

4.3.3 Algorithm

Choose *a*,

Consider an empty set as the trees set,

For z from *a* to *n - 1* do:

- Merge $\mathbb{D}_{i1}, \mathbb{D}_{i2}, \dots, \mathbb{D}_{iz}$,
- Grow a decision tree for the merged dataset,
- Calculate the prediction for the next month by using $\mathbb{S}_{i\,z+1}$ for each level of the obtained tree,
- Calculate the level error,
- Prune the tree at the level having the minimum level error,
- Call the pruned tree $\widetilde{\mathfrak{T}}_{i\,z}$,
- Add $\widetilde{\mathfrak{T}}_{i\,z}$ to the trees set.

4.4 Sickness prediction

For predicting new month sickness, we use the pairing schedule table for the new month, $\mathbb{S}_{i\,n+1}$, and all the pruned trees, $\widetilde{\mathfrak{T}}_{i\,a}, \widetilde{\mathfrak{T}}_{i\,a+1}, \dots, \widetilde{\mathfrak{T}}_{i\,n-1}$. These trees explain the best possible scenarios in the past for predicting their following month sickness. If we use the $\mathbb{S}_{i\,n+1}$ table as the input of these trees, they give $n - a - 1$ different values for the new month prediction, $\hat{s}\left(\widetilde{\mathfrak{T}}_{i\,a}, \mathbb{S}_{i\,n+1}\right)$, $\hat{s}\left(\widetilde{\mathfrak{T}}_{i\,a+1}, \mathbb{S}_{i\,n+1}\right)$, …, $\hat{s}\left(\widetilde{\mathfrak{T}}_{i\,n-1}, \mathbb{S}_{i\,n+1}\right)$. Each of them is based on the association rules that best explain a previous month sickness. In this way, we consider the possibility of occurrence previous scenarios in the future. Figure 4-5 represents the procedure of calculating individual estimations.

Based on these estimations, a weighted mean of the individual estimations is the best prediction for the new month when there is no other prior information and the prediction must be done only based on the pairing schedule. Here we explain the weights that we consider for making the final prediction.

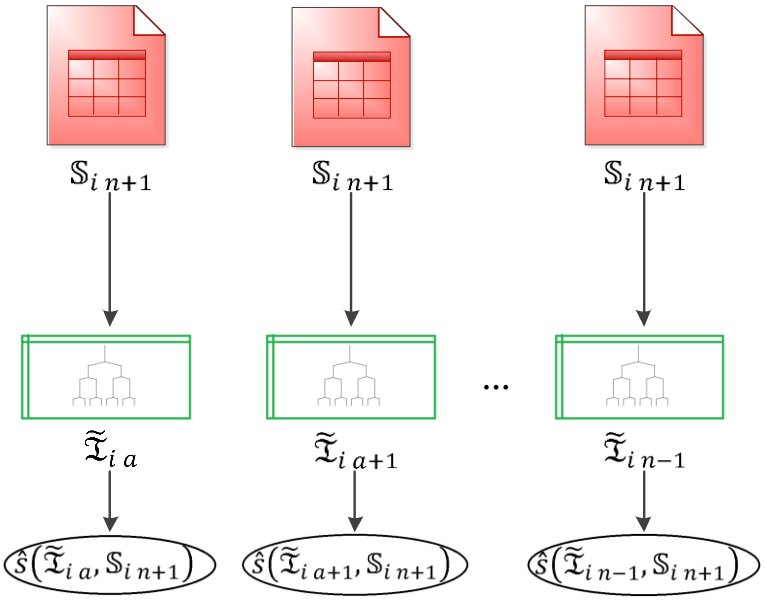


Figure 4-5: Prediction for the new month based on the pruned trees. The pairing schedule of the new month is the input of each available pruned tree for calculating possible values of prediction.

4.4.1 Similarity vector

As it was shown in the previous section, trees $\widetilde{\mathfrak{T}}_{i\,a}, \widetilde{\mathfrak{T}}_{i\,a+1}, \dots, \widetilde{\mathfrak{T}}_{i\,n-1}$ are pruned at the level that gives the best sickness prediction for $\mathbb{S}_{i\,a+1}, \mathbb{S}_{i\,a+2}, \dots, \mathbb{S}_{i\,n}$ respectively. For the new month sickness prediction, we do not have any other information except pairing schedule. So we consider the *similarity* of the new month schedule with the previous month schedules which are used in the tree pruning procedure. We use the intuition that it is more likely to have the same scenario in the months with more similar pairing schedule.

For calculating similarity vector, first we determine the important variables which characterize a pairing schedule. These variables can be pairing characteristics like monthly total time, total credit, day credit, night credit, deadhead credit, weekend credit, etc. If there are z variables, we calculate the value of these variables for the new month schedule and all the months that were used in tree pruning as the test sets. These values can be represented in a matrix,

$$\begin{pmatrix} \Upsilon_{a+1\,1} & \cdots & \Upsilon_{a+1\,z} \\ \vdots & \ddots & \vdots \\ \Upsilon_{n\,1} & \cdots & \Upsilon_{n\,z} \\ \Upsilon_{n+1\,1} & \dots & \Upsilon_{n+1\,z} \end{pmatrix}. \tag{4-9}$$

The matrix in (4-6) is normalized by dividing each column by its maximum value, i.e.

$$\boldsymbol{\upsilon}_k = \boldsymbol{\Upsilon}_k / m_k, \tag{4-10}$$

where $\boldsymbol{\Upsilon}_k$ is the column k of (4-6) and $m_k = \max\{\Upsilon_{a+1\,k}, \dots, \Upsilon_{n\,k}, \Upsilon_{n+1\,k}\}$. Matrix $\boldsymbol{\upsilon}$ can be shown as:

$$\boldsymbol{\upsilon} = \begin{pmatrix} \upsilon_{a+1\,1} = \Upsilon_{a+1\,1}/m_1 & \cdots & \upsilon_{a+1\,z} = \Upsilon_{a+1\,z}/m_z \\ \vdots & \ddots & \vdots \\ \upsilon_{n\,1} = \Upsilon_{n\,1}/m_1 & \cdots & \upsilon_{n\,z} = \Upsilon_{n\,z}/m_z \\ \upsilon_{n+1\,1} = \Upsilon_{n+1\,1}/m_1 & \dots & \upsilon_{n+1\,z} = \Upsilon_{n+1\,z}/m_z \end{pmatrix}. \tag{4-11}$$

Now we define the similarity between pairing schedule of month $n+1$ and pairing schedule of month l $(a+1 \le l \le n)$ as the Euclidian distance between the corresponding rows in matrix (4-8), i.e.

$$\tau_{n+1\,l} = \sqrt{\sum_{i=1}^{z} \left(\upsilon_{n+1\,j} - \upsilon_{l\,j}\right)^2}. \tag{4-12}$$

The similarity vector, $\boldsymbol{\tau}_{n+1} = (\tau_{n+1\,a+1}, \tau_{n+1\,a+2}, \dots, \tau_{n+1\,n})$, is used for calculating the final prediction.

4.4.2 Weighted mean as the prediction

For predicting sickness hours in new month, we use the prediction of trees obtained from $\mathbb{S}_{i\,n+1}$, $\hat{s}\left(\widetilde{\mathfrak{T}}_{i\,a}, \mathbb{S}_{i\,n+1}\right)$, $\hat{s}\left(\widetilde{\mathfrak{T}}_{i\,a+1}, \mathbb{S}_{i\,n+1}\right)$, …, $\hat{s}\left(\widetilde{\mathfrak{T}}_{i\,n-1}, \mathbb{S}_{i\,n+1}\right)$, the similarity vector, $\boldsymbol{\tau}_{n+1}$, and the errors of the trees, $e_{i\,a}$, $e_{i\,a+1}$, …, $e_{i\,n-1}$. Motivated by Kernel regression method (Watson, 1964), the suggested prediction is the *weighted mean of tree based predictions*, i.e.

$$\begin{aligned} \hat{s}_{i\,n+1} = & \frac{1}{2\sum_{k=a}^{n-1} \tau_{n+1\,k+1}} \sum_{k=a}^{n-1} \tau_{n+1\,k+1} \times \hat{s}\left(\widetilde{\mathfrak{T}}_{i\,k}, \mathbb{S}_{i\,n+1}\right) \\ & + \frac{1}{2\sum_{k=a}^{n-1} \left(1/\sqrt{e_{i\,k}}\right)} \sum_{k=a}^{n-1} \frac{1}{\sqrt{e_{i\,k}}} \times \hat{s}\left(\widetilde{\mathfrak{T}}_{i\,k}, \mathbb{S}_{i\,n+1}\right). \end{aligned} \tag{4-13}$$

The Equation (4-13) consists of two parts and each part is a weighted mean of $\left\{\hat{s}\left(\widetilde{\mathfrak{T}}_{i\,k}, \mathbb{S}_{i\,n+1}\right), a \le k \le n-1\right\}$. The weights in the first part of this equation are similarity vector.

We mathematically formulize the intuition that more similarity between two pairing schedules must have more prediction weight. The weights on the second part are a decreasing function of the errors of the trees. We use these errors as the weight for decreasing the effect of outliers on the prediction. When the prediction of a decision tree is better in the learning process, it gets more weight in the calculation of the new month prediction.

In the next chapter, the results of implementing this methodology in an airline company are reported.

CHAPTER 5 IMPLEMENTATION

In this chapter, we present the results of applying the discussed methodology in an airline company. In Section 5.1, descriptive statistics and some plots give an explanation of the real data. In Section 5.2 the methodology is illustrated by one detailed example. The pre-test of the procedure before applying this method in the business is discussed in Section 5.3. Finally the developed decision support system is introduced in Section 5.4.

5.1 Descriptive Statistics

The final table which is our database for analyzing and modeling consists of 3 years schedule pairings and operations' records of an airline company. It contains 13 different positions and 382,202 records for total of 36 months. During 3 years, 379,129 flight hours were replaced due to sickness. This means 7 percent of total flight hours were sick.

Table 5.1: Descreptive statistics for each position and month

Positions	Flying Hours					Sickness Hours				
	Total		Monthly			Total		Monthly		
	Sum	Percent	Min	Mean	Max	Sum	Percent	Min	Mean	Max
Position 1	974025	18%	22308	27056	31382	75950	19%	1120	2110	3099
Position 2	974006	18%	22308	27056	31382	71273	18%	1323	1980	3051
Position 3	123039	2%	2594	3418	3966	9310	2%	93	259	387
Position 4	126170	2%	2630	3505	4213	9314	2%	78	259	473
Position 5	45737	1%	1009	1270	1766	3098	1%	19	86	190
Position 6	427482	8%	9890	11874	14370	33359	8%	607	927	1472
Position 7	430080	8%	9982	11947	14684	26765	7%	470	743	1303
Position 8	167928	3%	2493	4665	5788	12179	3%	126	338	650
Position 9	301711	6%	6855	8381	9409	26271	7%	300	730	1244
Position 10	416858	8%	9144	11579	12755	30795	8%	444	855	1484

Position 11	208665	4%	4021	5796	6771	17647	4%	161	490	896
Position 12	616058	11%	15575	17595	18922	44453	11%	894	1270	1866
Position 13	616009	11%	15575	17594	18922	42154	10%	753	1204	1792

Table 5.15 shows the descriptive statistics for flight and sickness hours for each position. Comparison of minimum and maximum monthly sickness hours shows this variable has a high variation. This high variation is more evident in Figure 5-1 which illustrates monthly sickness hours for Position 1 and Position 2. These are the positions with large number of flight hours in the airline company and they cover 36 percent of total flight hours. In the following figures, Figure 5-1 to Figure 5-7, our examples are from these two positions.

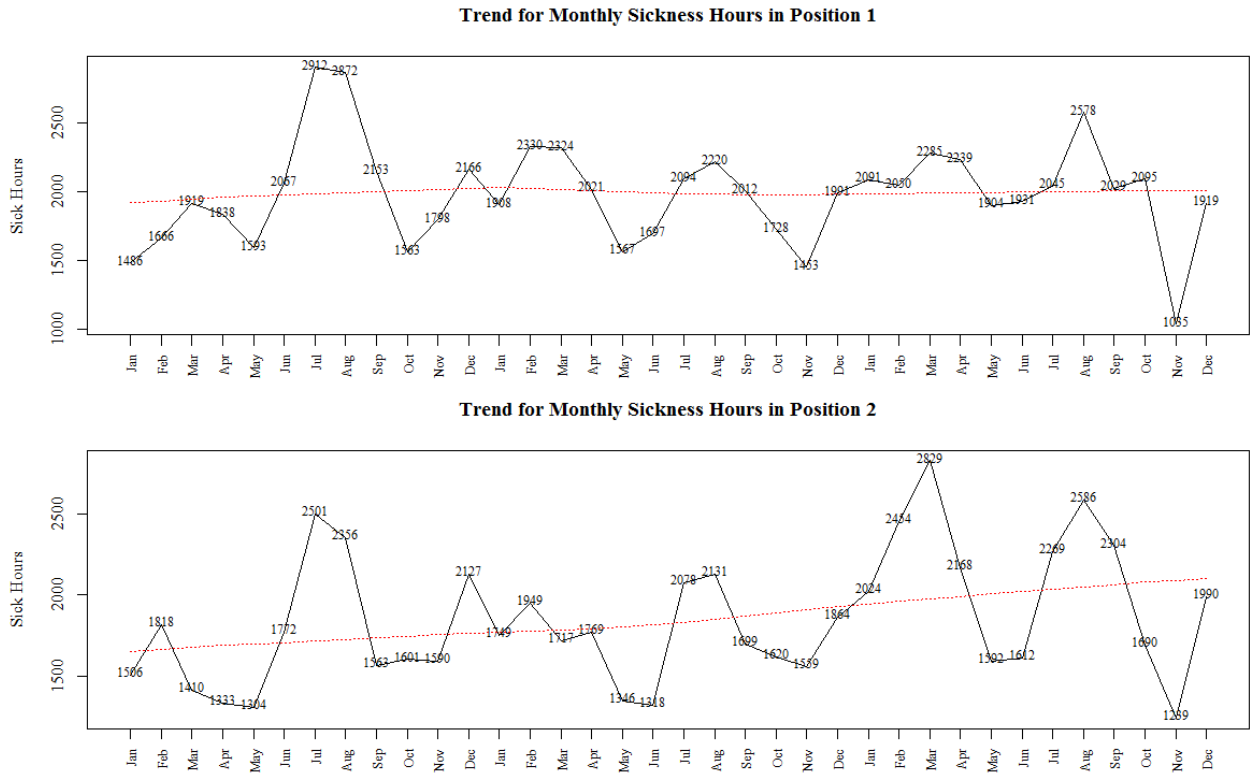


Figure 5-1: Monthly sickness hours for two Positions.

Figure 5-1 shows monthly sick hours for Position 1 and 2 from January 2010 to December 2012. The black line shows the sickness hours and the red dotted line represents the trend of the data. There are some evident outliers, July and August 2010, August and November 2012 for Position 1, and July and August 2010, March and August 2012 for Position 2. The trend for Position 1 is uniform while the figure shows an increasing trend in Position 2.

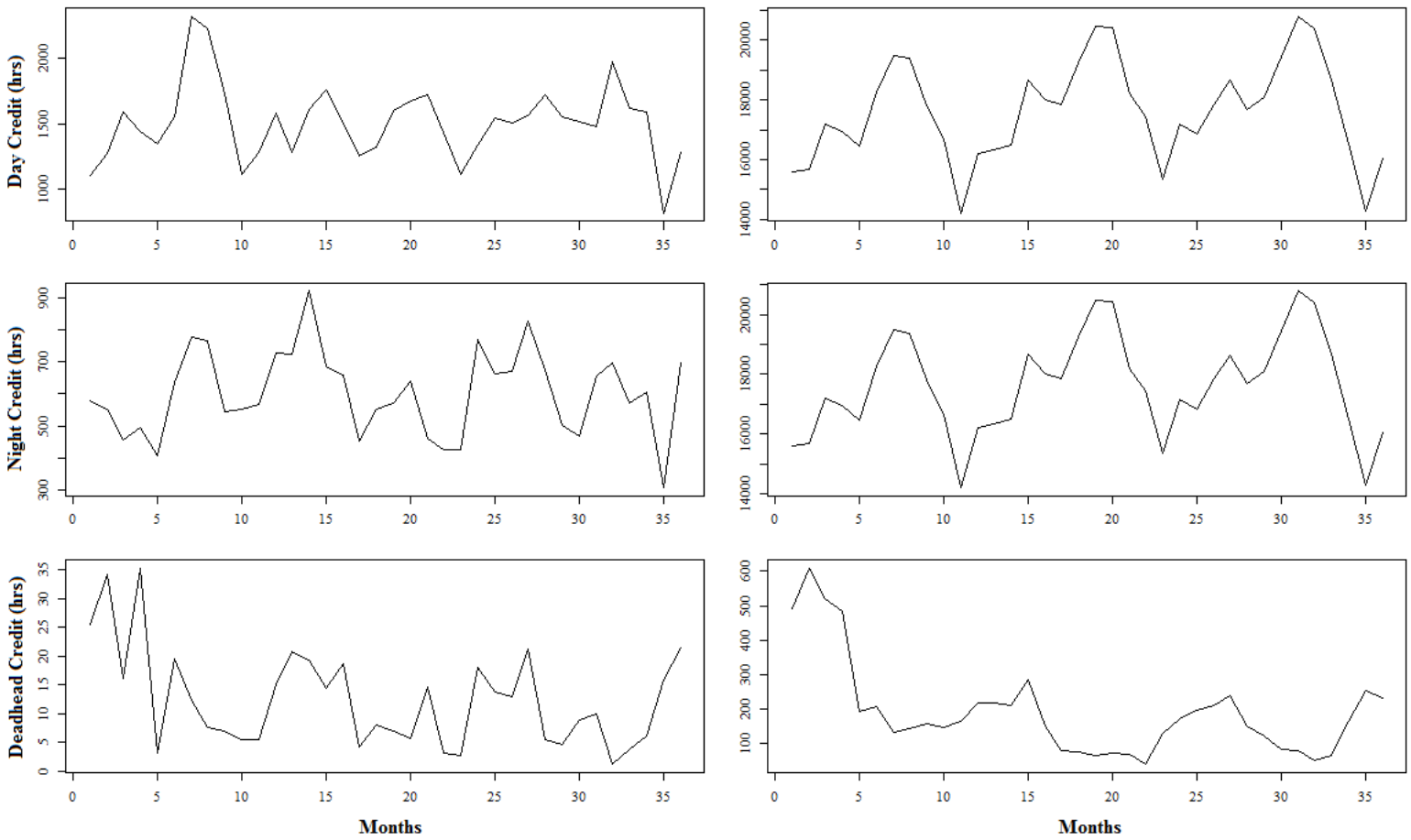


Figure 5-2: Comparison between sick and non-sick pairings.

In Figure 5-2, left and right panel plots show monthly variation of different variables among the sick and non-sick pairings, respectively. The variables top to down are day credit, night credit and deadhead credit. The different pattern in the sick plots (left) in comparison with the non-sick plots (right) makes these variables good candidates as the model covariates.

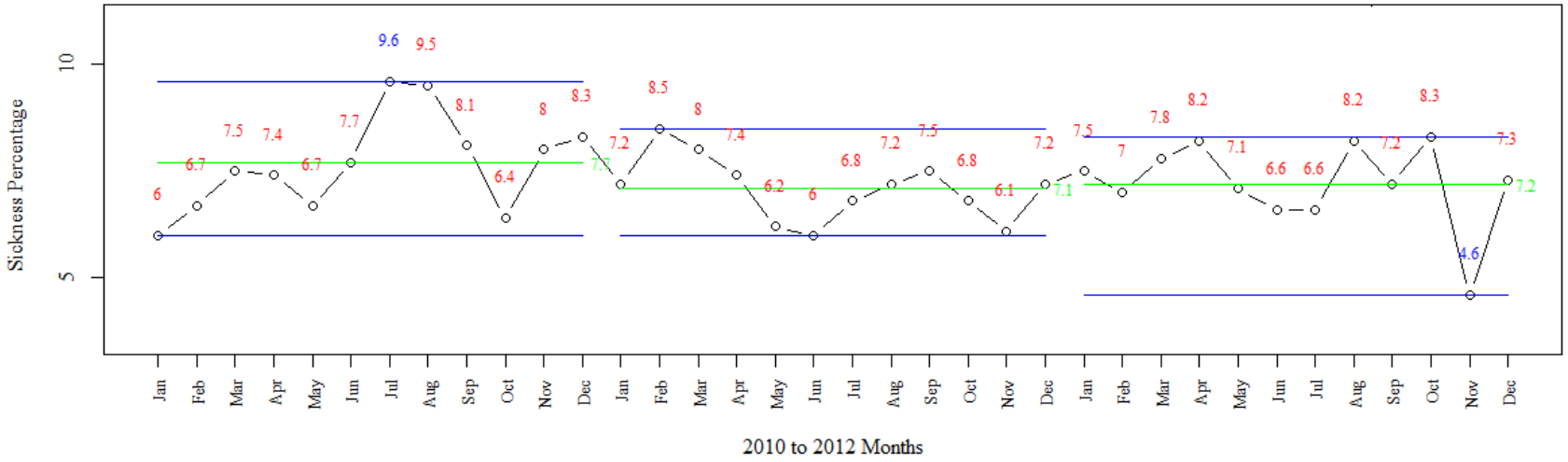


Figure 5-3: Monthly sickness percentage for Position 1.

In Figure 5-3, each block represents sickness percentage for one year from 2010 to 2012. The blue lines determine the range of annually sickness percentage, and the green line is the mean of sickness percentage for each year. The maximum of 2011 happened in February; however for 2010 and 2012 February is not even above the annual mean. The same as the maximum month of 2012, October, which is under the annual mean for 2010 and 2011.

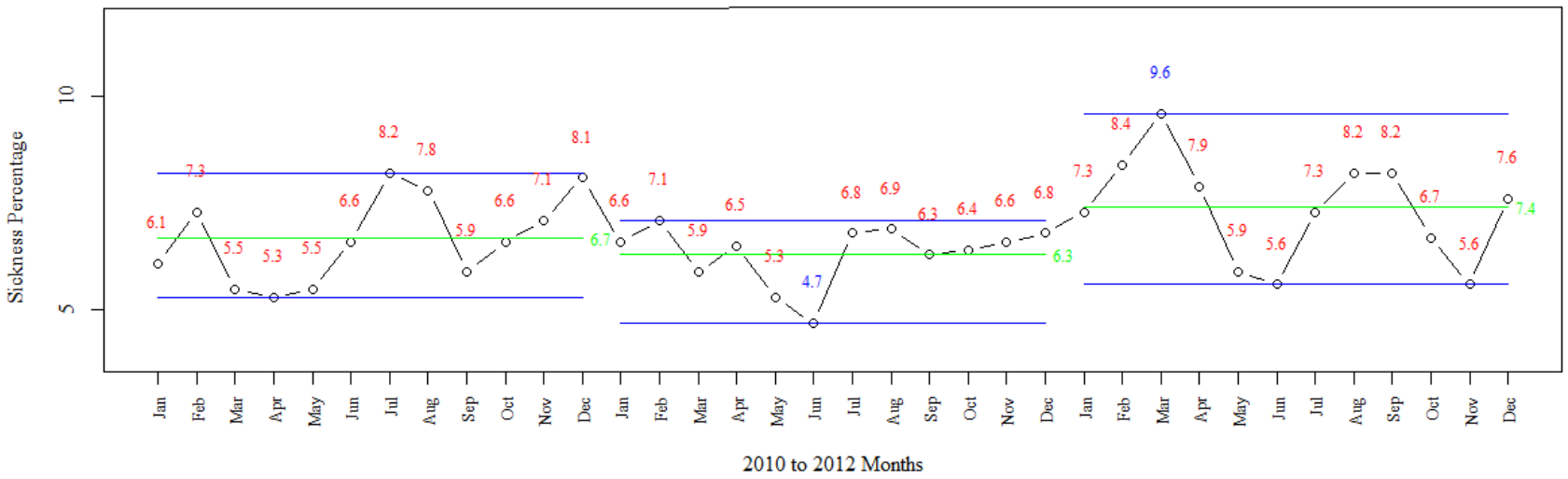


Figure 5-4: Monthly sickness percentage for Position 2

In Figure 5-4, each block represents sickness percentage for one year from 2010 to 2012. The blue lines determine the range of annually sickness percentage and the green line is the mean of sickness percentage for each year. The annual sickness percentage in 2012 is higher than the other two years and also the range of 2012 is clearly higher than the range of 2010 and 2011. This shows the increasing trend in the sickness that we saw in Figure 3-5 (b) with respect to monthly sick hours.

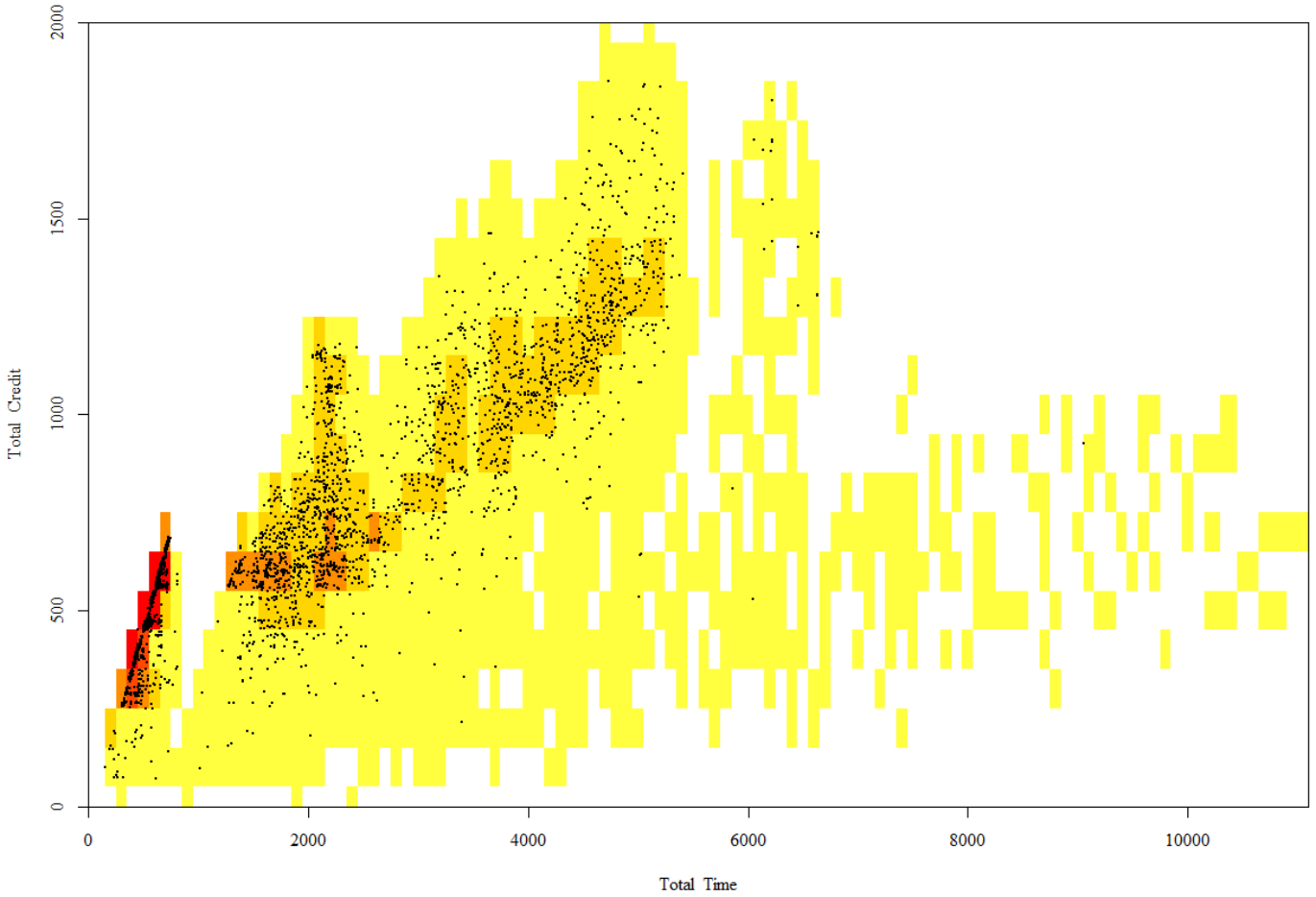


Figure 5-5: Mass plot for sickness, comparing total time against total credit.

The colors in this figure represent the density of pairings in the space of total time and total credit. About 19 percent of the pairings scheduled in the yellow region, 22 percent on the gold region, and 59 percent of the pairings scheduled in the small region with dark orange, orange red or red color, respectively. The dark points show sick pairings. The distribution of sick pairings suggests different patterns of sickness for these two variables, e.g. in red region the pattern of sickness is completely different from other regions.

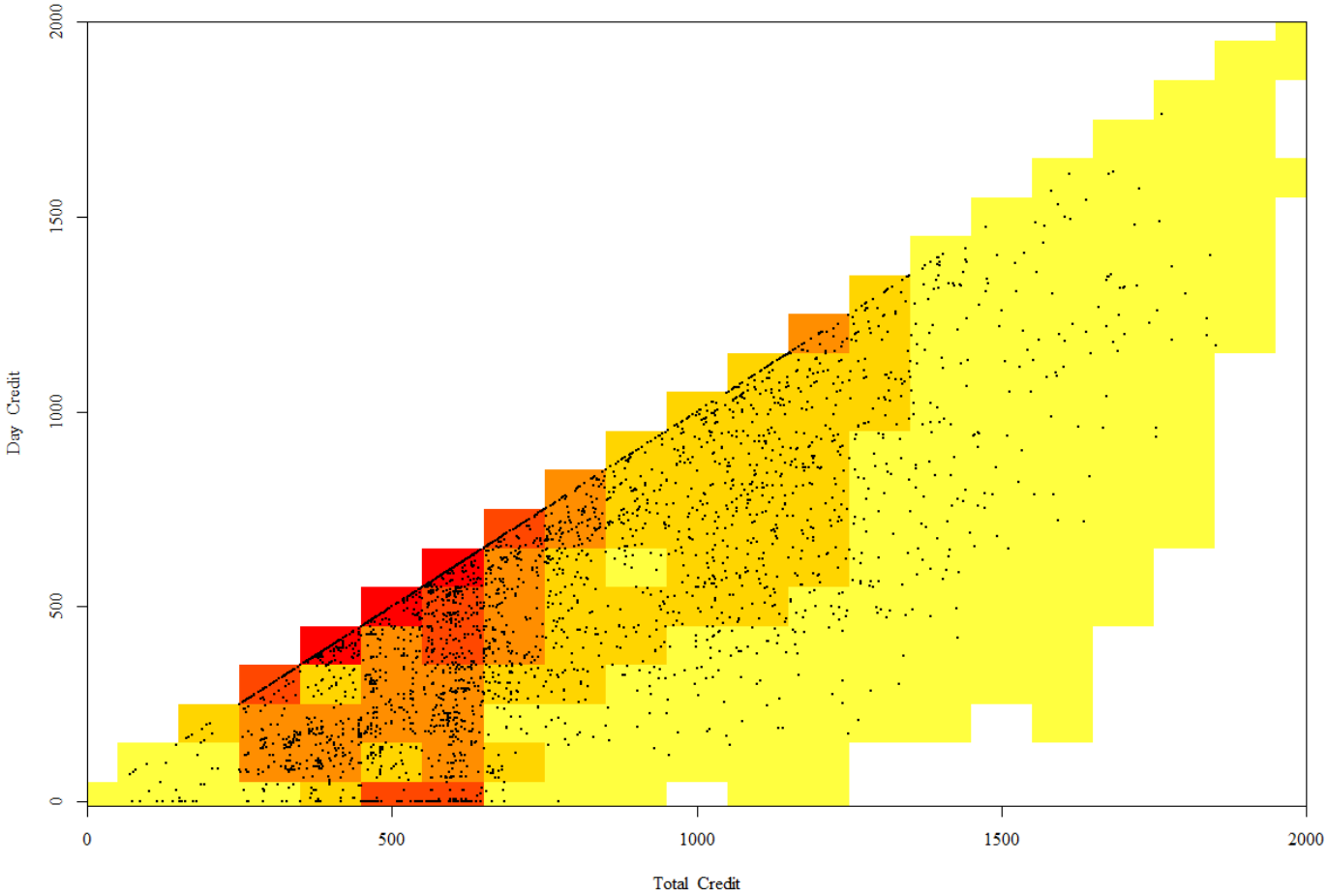


Figure 5-6: Mass plot for sickness, comparing total credit against day credit.

The colors in this figure represent the density of pairings in the space of total credit and day credit. About 9 percent of the pairings scheduled in the yellow region, 22 percent on the gold region, 24 percent on the dark orange region, and 45 percent of the pairings planned in the region with orange red or red, respectively. The dark points show sick pairings. The sicknesses appeared on the line $y = x$, belong to those pairings that all their legs are day flight. Some mass points are evident on orange red and red regions.

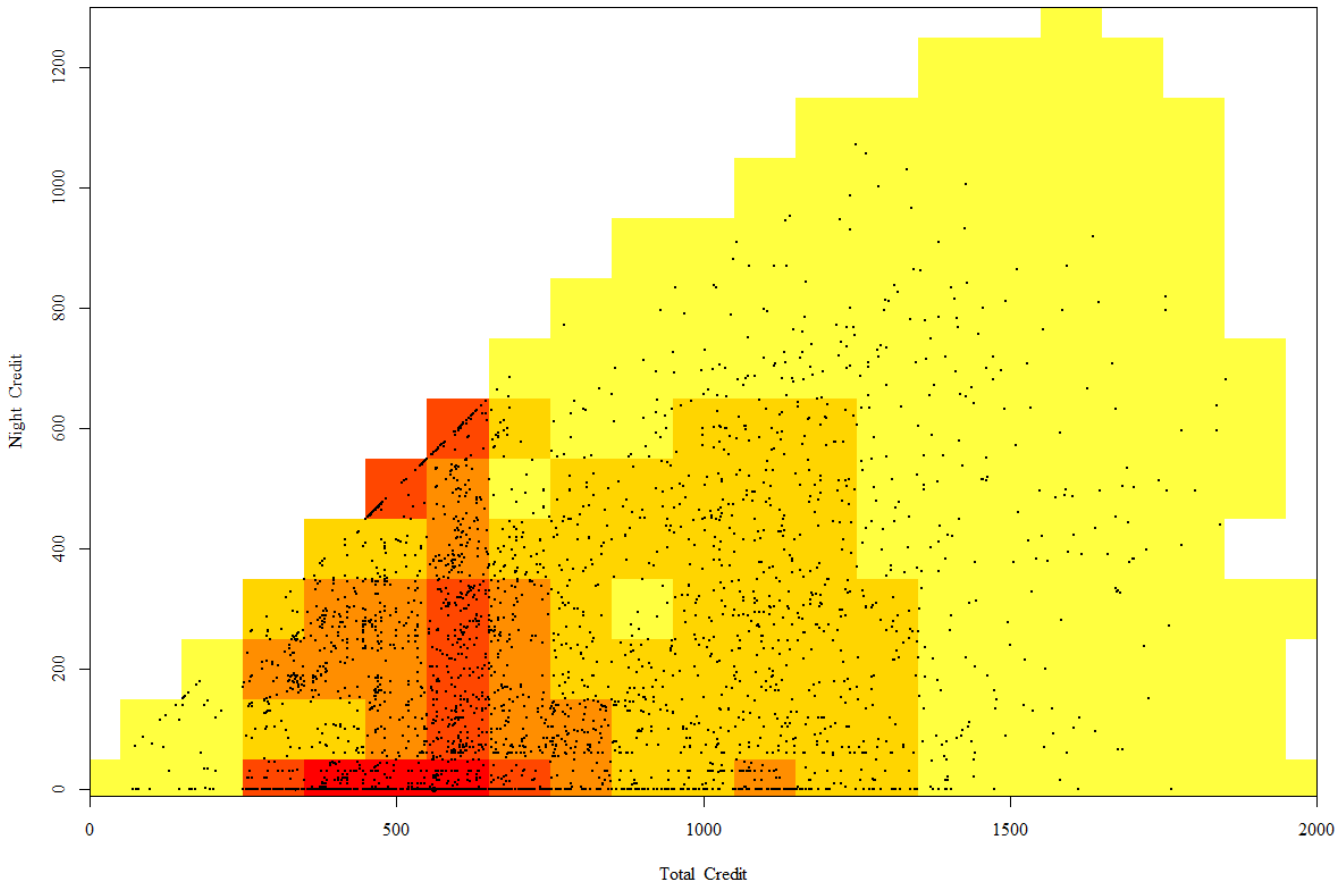


Figure 5-7: Mass plot for sickness, comparing total credit against night credit.

The colors in this figure represent the density of pairings in the space of total credit and night credit. The percentages of pairings in different color regions are the same as Figure 5-6. The dark points show sick pairings. The sick pairings on the line $y = 0$, are those pairings that do not have night credit.

Figure 5-2 compares 3 pairing characteristics (day credit, night credit and deadhead credit) between sick and non-sick pairings in Position 1.

Figure 5-3 and Figure 5-4 show the monthly sickness percentage for Positions 1 and 2. The sickness percent in 2012 for Position 2 is 7.2 percent and is higher than this percentage in 2010 and 2011 (6.7 percent and 6.3 percent respectively). This is an evidence that increasing trend in Figure 5-1 for Position 2 is not because of the increasing of flight hours in this position. It can be seen, in these two figures that the prediction of the sickness based on the months couldn't be applicable. For example in Figure 5-3, the sickness in November 2011 and 2012 is small but in 2010 the percentage is more than average.

The mass plots (Figure 5-5, Figure 5-6, Figure 5-7) present a four-dimensional representation of sickness events, number of pairings and influenced variables for prediction. In Figure 5-5, the space of total time and total credit has been gridded and painted in color ranging from yellow to red. Yellow pixels show the less frequent regions in the term of number of pairings while red pixels are the most frequent regions. The dark points show the sickness events.

These plots demonstrate the idea of the Chapter 4. We search in the space of all variables for relatively frequent regions in terms of number of pairing which sickness occurs in them more than other similar regions.

5.2 Prediction for Position 1

The data that we use here for the analysis and description of the methodology comes from an actual airline company. The data consists of 36 months from January 2010 to December 2012. Therefore, every month related index of variables start from January 2010, e.g. $\mathbb{D}_{1\,1}$ is the dataset of January 2010 in Position 1, $\mathbb{D}_{1\,2}$ is the dataset of February 2010 in Position 1.

Two years of datasets, 2010 and 2011, are used to find the hidden structures in the database and the predictions are made for the last year of available data. This means, the example, figures and tables that are presented here have exactly the same structure as those that can be used in real situation.

In this section, we explain in detail the procedure of predicting sickness for one of the positions. The proposed methodology can be applied for other positions as well. Position 1 is the largest position in the airline company and it has special importance among the managers.

Suppose that we want to predict sickness hours in March 2012 for Position 1. At that time all data from January 2010 to February 2012 and the pairing schedule for March 2012 are available. We use 2010 dataset for making first decision tree. Now we explain the steps of the algorithm presented at Section 4.3.3.

5.2.1 First model

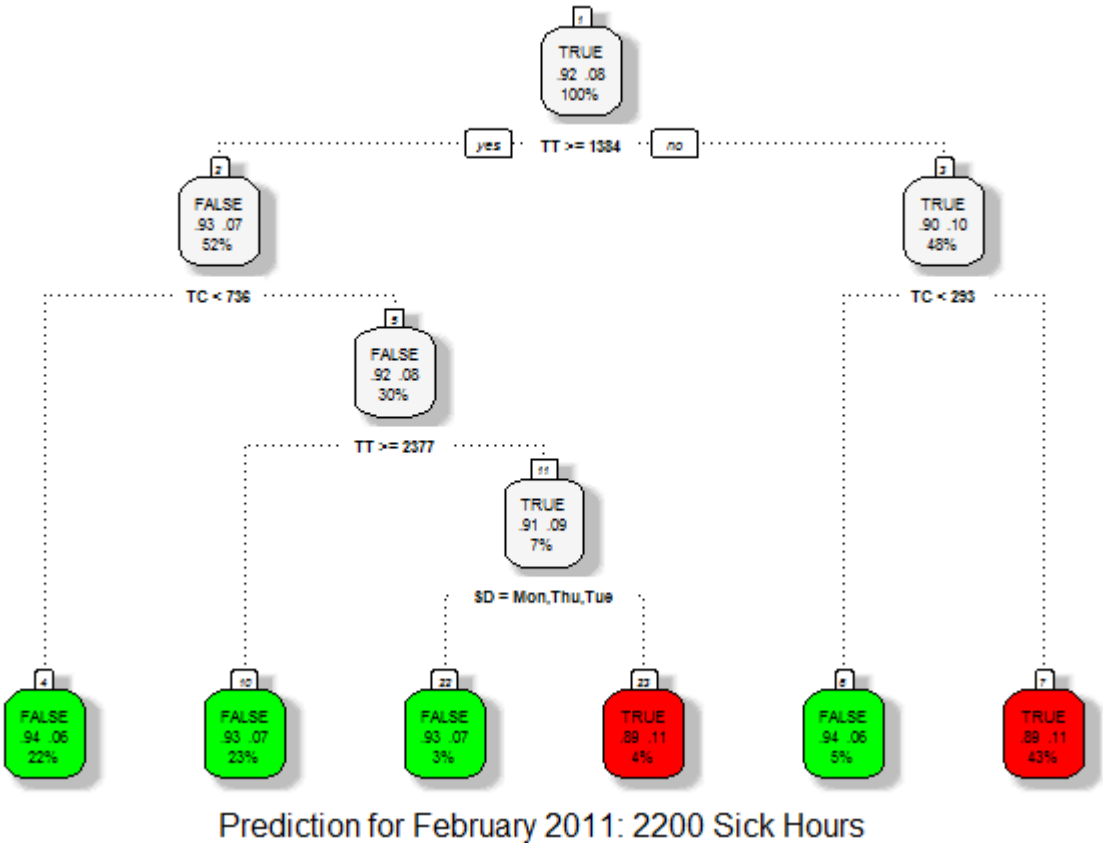


Figure 5-8: Decision tree obtained from 2010 data for Position 1 ($\mathfrak{T}_{1\,12}$). This tree and its sub-trees (Figure 5-9) will be used for making the first model for prediction.

After merging $\mathbb{D}_{1\,1}, \mathbb{D}_{1\,2}, \ldots, \mathbb{D}_{1\,12}$, datasets of 2010, a decision tree is grown by using the growing method of Section 4.3.1. The resulting decision tree is plotted in Figure 5-8. It has 6 terminal nodes and 3 splitting variables, total time (TT), total credit (TC), and pairing start day (SD). In this position, the proportion of being sick in 2010 is 0.08, so highly unbalanced portion of sickness in the red terminal nodes are higher, and in green terminals is less than 0.08. For example, node 7, with the path of total time less than 1334 minutes and total credit more than 293 minutes, consists

of 43 percent of the pairings, and has 11 percent sick pairings. Node 4, with the path of total time more than 1334 minutes and total credit less than 737 minutes, consists of 22 percent of the pairings and has 6 percent sick pairings.

Here is the fitted model as the result of *rpart* package in R. In each line of this report node number, splitting criteria, number of the observations in the node, loss of the node, response value, and probability of the node are presented. Response value in each node is either *TRUE* (sick) or *FALSE* (non-sick). Two probabilities show proportion of non-sick pairings and proportion of sick pairings in a node. Loss of a node, here, is number of misclassified objects multiply by the misclassification cost. As discussed in Section 4.3.1 and Equation (4-6), we considered misclassifying cost for sick pairing equal to 11 ($\frac{1-p}{p} = \frac{0.92}{0.08} = 11$) and misclassifying cost for non-sick pairing as 1.

```
node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 27476 25141 TRUE (0.91501674 0.08498326)
   2) TT>=1384 14276 10659 FALSE (0.93212384 0.06787616)
     4) TC< 736.5 5985  3729 FALSE (0.94335840 0.05664160) *
     5) TC>=736.5 8291  6930 FALSE (0.92401399 0.07598601)
      10) TT>=2377 6370  4939 FALSE (0.92951334 0.07048666) *
      11) TT< 2377 1921  1740 TRUE (0.90577824 0.09422176)
        22) SD=Mon,Thu,Tue 890   704 FALSE (0.92808989 0.07191011) *
        23) SD=Fri,Sat,Sun,Wed 1031   914 TRUE (0.88651794 0.11348206) *
   3) TT< 1384 13200 11834 TRUE (0.89651515 0.10348485)
     6) TC< 293 1397   946 FALSE (0.93843951 0.06156049) *
     7) TC>=293 11803 10523 TRUE (0.89155300 0.10844700) *
```

The decision tree in Figure 5-8 has 4 levels, so for the next step of the algorithm, we prune it in different levels and obtain 3 pruned trees. These decision trees are shown in Figure 5-9. Now by using schedule pairing of the following month, we estimate sickness hours of January 2011 based on each of these trees.

Consider the pruned tree at the first level; it is a simple decision tree with two nodes, one splitting variable, and the probability of being sick equal to 0.08 at the root. Based on this tree, if total time of a pairing is less than 1384, the probability of being sick increases to 0.1034; otherwise it decreases to 0.0679. That means the space of the variables is split into two regions related to total time of the pairing.

By using Equation (4-5), estimation of sick hours for January 2011 is 0.1034 × 9411 + 0.0679 × 17137 = 2137 hours, where 9411 is the sum of flight hours of January 2011 pairings in Position 1

with the pairing total time less than 1384 and 17137 is the same sum for the pairings with total time more than 1384. It can be written as

$$\hat{s}(\langle \mathfrak{T}_{1\,12} \rangle_1, \mathbb{S}_{1\,13}) = \sum_{k=1}^{m} \mathcal{p}_k c_k = 2137 \text{ sick hours.}$$

The actual sickness hour in January 2011 from table $\mathbb{D}_{1\,13}$ is 1907 hours. By putting this observation and the estimation of 2137 sick hour in the Equation (4-7) the first level error for $\mathfrak{T}_{1\,12}$, $e_{1\,12_1}$, is calculated as 2137 – 1907 = 230.

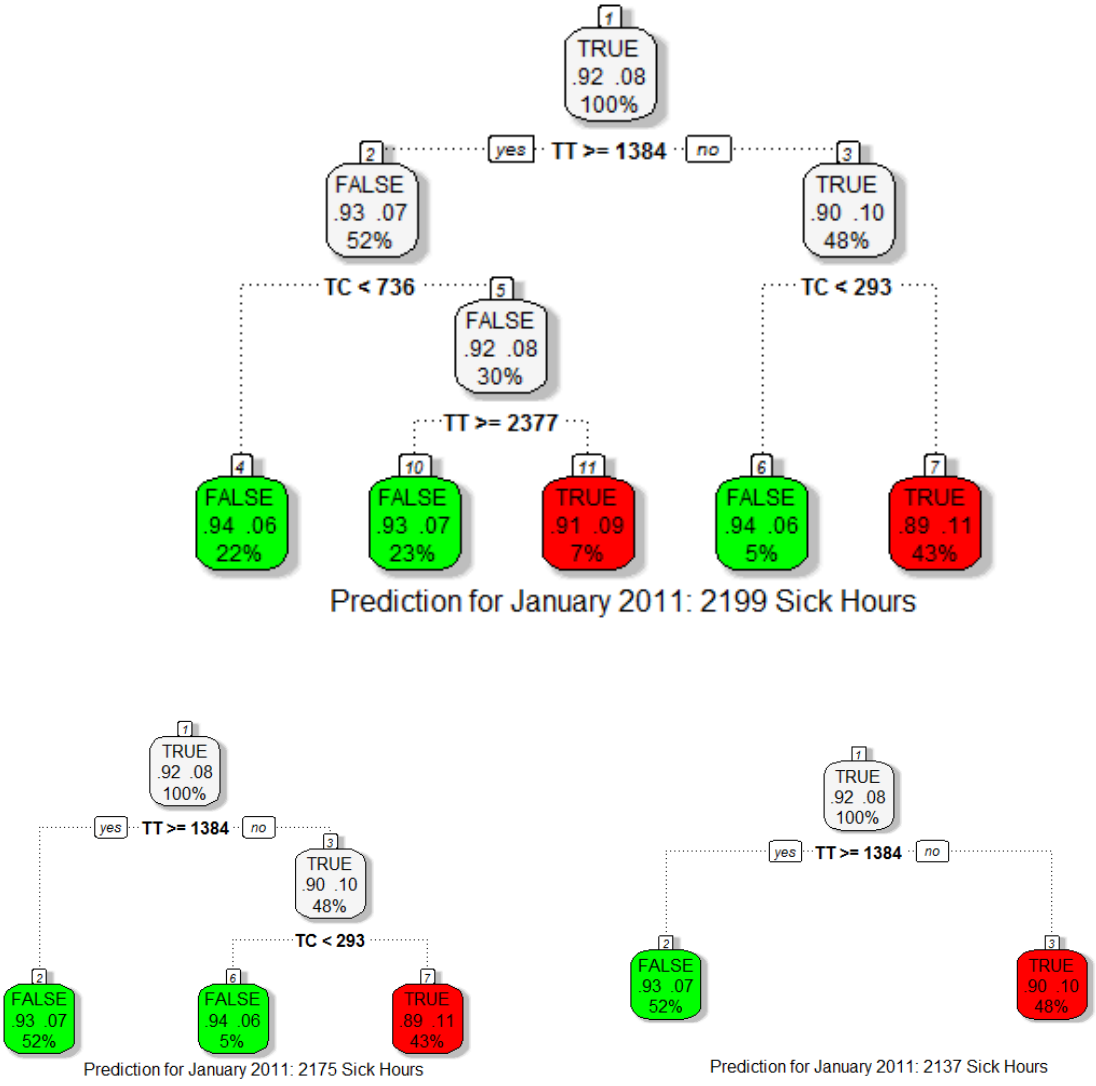


Figure 5-9: Pruning first decision tree (Figure 5-8) at different levels. Top at level 3 ($\langle \mathfrak{T}_{1\,12} \rangle_3$), bottom left at level 2 ($\langle \mathfrak{T}_{1\,12} \rangle_2$), and bottom right at level 1($\langle \mathfrak{T}_{1\,12} \rangle_1$). TT is total time, TC is total credit.

For the second tree ($\langle \mathfrak{T}_{1\,12} \rangle_2$ in Figure 5-9) the space of variable is split in three regions:

- $r_1 = \{\text{pairings with } total\ time\ \geq 1384\}$ with the sick probability equal to 0.0679,
- $r_2 = \{\text{pairings with } total\ time\ \geq 1384\ \&\ total\ credit < 293\}$ with the sick probability equal to 0.0616,
- $r_3 = \{\text{pairings with } total\ time\ \geq 1384\ \&\ total\ credit \geq 293\}$ with the sick probability equal to 0.1084.

The estimation of January 2011 sick hours in Position 1 based on this tree can be calculated as,

$$\hat{s}(\langle \mathfrak{T}_{1\,12} \rangle_2, \mathbb{S}_{1\,13}) = \sum_{k=1}^{m} p_k c_k = 0.0679 \times 17137 + 0.0616 \times 183 + 0.1084 \times 9228$$
$$= 2175 \text{ sick hours.}$$

The values 17137, 183, and 9228 are sum of the flight hours in regions r_1 to r_3, respectively, for the January 2011 schedule pairing in Position 1. The level error for this pruned tree, by using Equation (4-7) and $\eta = 2$, is 2175 – 1907 = 268.

The following table shows the level error for each of the obtained trees from $\mathfrak{T}_{1\,12}$.

Table 5.2: Level errors for $\mathfrak{T}_{1\,12}$.

Level	Sick estimation	Actual sick	Level error
1	2137	1907	**230**
2	2175	1907	268
3	2199	1907	292
4	2200	1907	293

Based on this table the pruned tree at level 1 gives the minimum level error so we keep $\langle \mathfrak{T}_{1\,12} \rangle_1$ as the first model for predicting, say $\widetilde{\mathfrak{T}}_{1\,12}$. It has the corresponding error equals to $e_{1\,12} = 230$ based on the Equation (4-8). The decision tree $\widetilde{\mathfrak{T}}_{1\,12}$ is shown in Figure 5-10. Here we use another plotting method for indicating the difference between pruned trees and monthly models in the learning process.

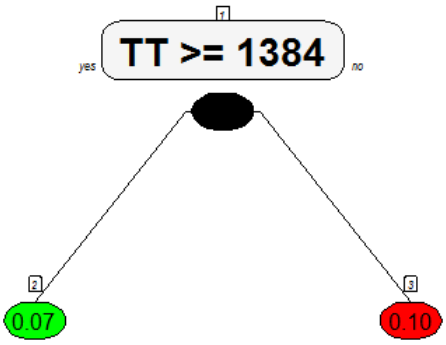


Figure 5-10: First decision tree that is used for predicting, $\widetilde{\mathfrak{T}}_{1\,12}$. TT is total time.

5.2.2 Second Model

Now, by the same process, the second decision tree is created. The original tree, $\mathfrak{T}_{1\,13}$, is grown by using datasets of 2010 and January 2011. This tree (shown in Figure 5-11) has 3 levels and 7 terminal nodes. Applying $\mathfrak{T}_{1\,13}$ to the pairing schedule of February 2011 results an estimation of 2294 sick hours. The *rpart* output of this tree is as follows:

```
node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 29746 27230 TRUE (0.91541720 0.08458280)
   2) TT>=1384 15364 11396 FALSE (0.93256964 0.06743036)
     4) TC< 736.5 6384  4004 FALSE (0.94298246 0.05701754) *
     5) TC>=736.5 8980  7392 FALSE (0.92516704 0.07483296)
      10) Weak< 2.5 675   275 FALSE (0.96296296 0.03703704) *
      11) Weak>=2.5 8305  7117 FALSE (0.92209512 0.07790488)
        22) TT>=2377 6357  5060 FALSE (0.92763882 0.07236118) *
        23) TT< 2377 1948  1761 TRUE (0.90400411 0.09599589)
          46) SD=Mon,Thu,Tue 903   726 FALSE (0.92691030 0.07308970) *
          47) SD=Fri,Sat,Sun,Wed 1045   924 TRUE (0.88421053 0.11578947) *
   3) TT< 1384 14382 12902 TRUE (0.89709359 0.10290641)
     6) TC< 304.5 1509  1045 FALSE (0.93704440 0.06295560) *
     7) TC>=304.5 12873 11488 TRUE (0.89241047 0.10758953) *
```

The tree $\mathfrak{T}_{1\,13}$ can have 2 sub-trees. The first sub-tree, $\langle\mathfrak{T}_{1\,13}\rangle_2$ shown in the left panel of Figure 5-12, is the result of pruning at level 2 with an estimation of February 2011 sick equal to 2238 hours. The second subtree, $\langle\mathfrak{T}_{1\,13}\rangle_1$ shown in the right panel of Figure 5-12, is obtained by pruning $\mathfrak{T}_{1\,13}$ at its first level. The estimation for sickness in February 2011 by using $\langle\mathfrak{T}_{1\,13}\rangle_1$ is 2202 hours.

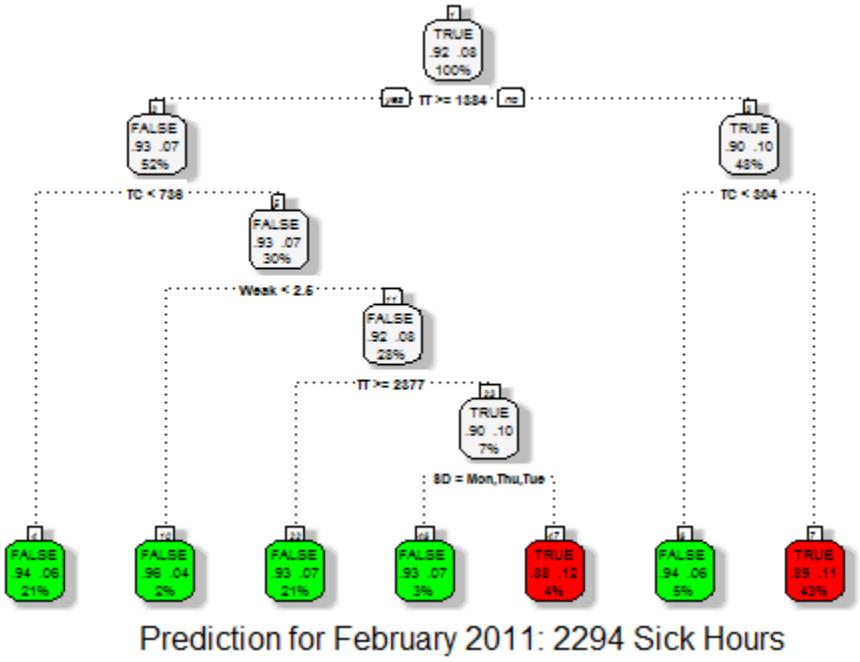


Figure 5-11: Decision tree obtained from first 13 months data for Position 1.

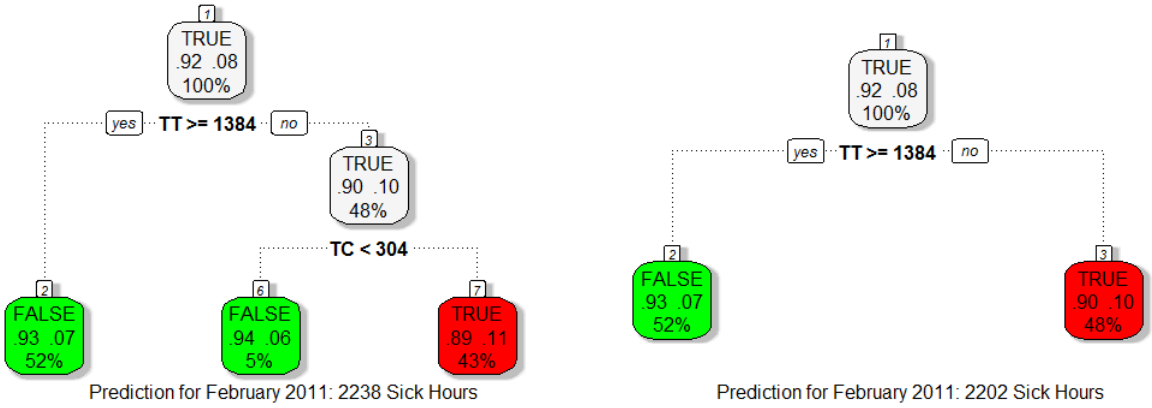


Figure 5-12: Sub-trees of $\mathfrak{T}_{1\,13}$, left pruned tree at level 2 ($\langle\mathfrak{T}_{1\,13}\rangle_2$) and right the pruned tree at level 1 ($\langle\mathfrak{T}_{1\,13}\rangle_1$). TT is total time, TC is total credit and SD is the start day of the pairing.

The actual sickness for February 2011 is 2330 hours. All three estimations are less than real value so the level errors are the absolute value of difference between actual and estimation multiplied by $\eta = 2$. Table 5.3 shows the level errors for $\mathfrak{T}_{1\,13}$.

Table 5.3: Level errors for $\mathfrak{T}_{1\,13}$.

Level	Sick estimation	Actual sick	Level error
1	2202	2330	256
2	2238	2330	184
3	2294	2330	**72**

Based on these level errors the original tree, $\mathfrak{T}_{1\,13}$, is selected as the second model, $\widetilde{\mathfrak{T}}_{1\,13}$, with the error equal to $e_{1\,13} = 72$. Figure 5-6 represents the second model.

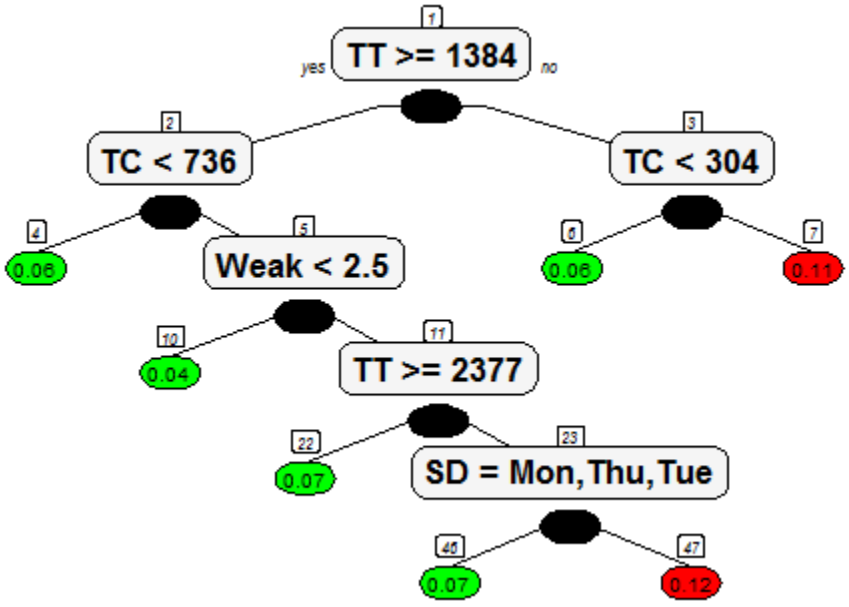


Figure 5-13: second decision tree that is used for predicting, $\widetilde{\mathfrak{T}}_{1\,13}$. TT is total time, TC is total credit and SD is the start day of the pairing.

5.2.3 Other models and prediction

We continue this procedure for making models for all the other months from March 2011 to January 2012. At the end, 14 different models, $\widetilde{\mathfrak{T}}_{1\,12}, \widetilde{\mathfrak{T}}_{1\,13}, \dots, \widetilde{\mathfrak{T}}_{1\,25}$, and 14 different error values, $e_{1\,12}, e_{1\,13}, \dots, e_{1\,25}$. By using these models and pairing schedule of March 2012, $\mathbb{S}_{1\,27}$, the sickness estimation relative to each tree is calculated, $\hat{s}\left(\widetilde{\mathfrak{T}}_{1\,12}, \mathbb{S}_{1\,27}\right), \dots, \hat{s}\left(\widetilde{\mathfrak{T}}_{1\,25}, \mathbb{S}_{1\,27}\right)$.

Then the similarity vector, $\boldsymbol{\tau}_{27} = (\tau_{27\,13}, \tau_{27\,14}, \dots, \tau_{27\,26})$, for Position 1 is created. Note that we calculate similarity between pairing schedule of the goal month, here March 2012, and the pairing schedule of the test set in tree pruning procedure.

Table 5.4: Different sickness estimation, model errors and similarity vector for March 2012.

k	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
$\hat{\boldsymbol{s}}(\widetilde{\mathfrak{T}}_{1\,k}, \mathbb{S}_{1\,27})$	2357	2435	2392	2361	2355	2328	2301	2289	2281	2284	2278	2264	2266	2263
$\boldsymbol{e}_{1\,k}$	230	72	5	154	456	500	279	138	51	237	330	87	67	202
$\boldsymbol{\tau}_{27\,k+1}$	76	76	131	68	35	55	64	57	44	39	39	118	88	110

Now all the components of Equation (4-13) are available and it is possible to calculate sickness prediction for March 2012 in Position 1,

$$\hat{s}_{1\,27} = 2327 \text{ hours.}$$

The actual sickness for Position 1 in March 2012 is 2285 hours and our prediction has 1.8 percent of over-prediction. Figure 5-14 plots estimation values of each model, prediction and actual sickness of Position 1 in March 2012.

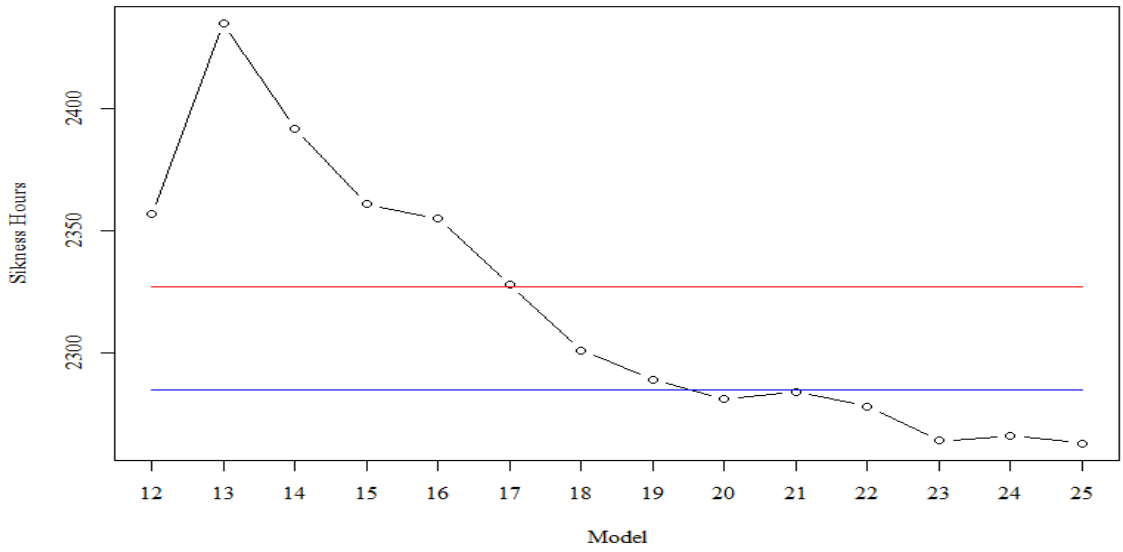


Figure 5-14: Comparison of model estimations, prediction and actual sickness of Position 1 in March 2012. Black circles are the estimations of different models, blue horizontal line is the actual sickness hours, and red horizontal line is the prediction.

5.3 Pre-test

For testing the quality of fit of the proposed methodology and its ability of predicting, we applied the procedure for all the positions and all the months of 2012. This was the first pre-test of the model before applying it in practice.

The prediction procedure has been simulated for 2012 in a way that is similar to the real situation. For this pre-test, only those information that could be available for a real prediction has been used and the sickness observations have been used just for comparison and as a criteria for the model.

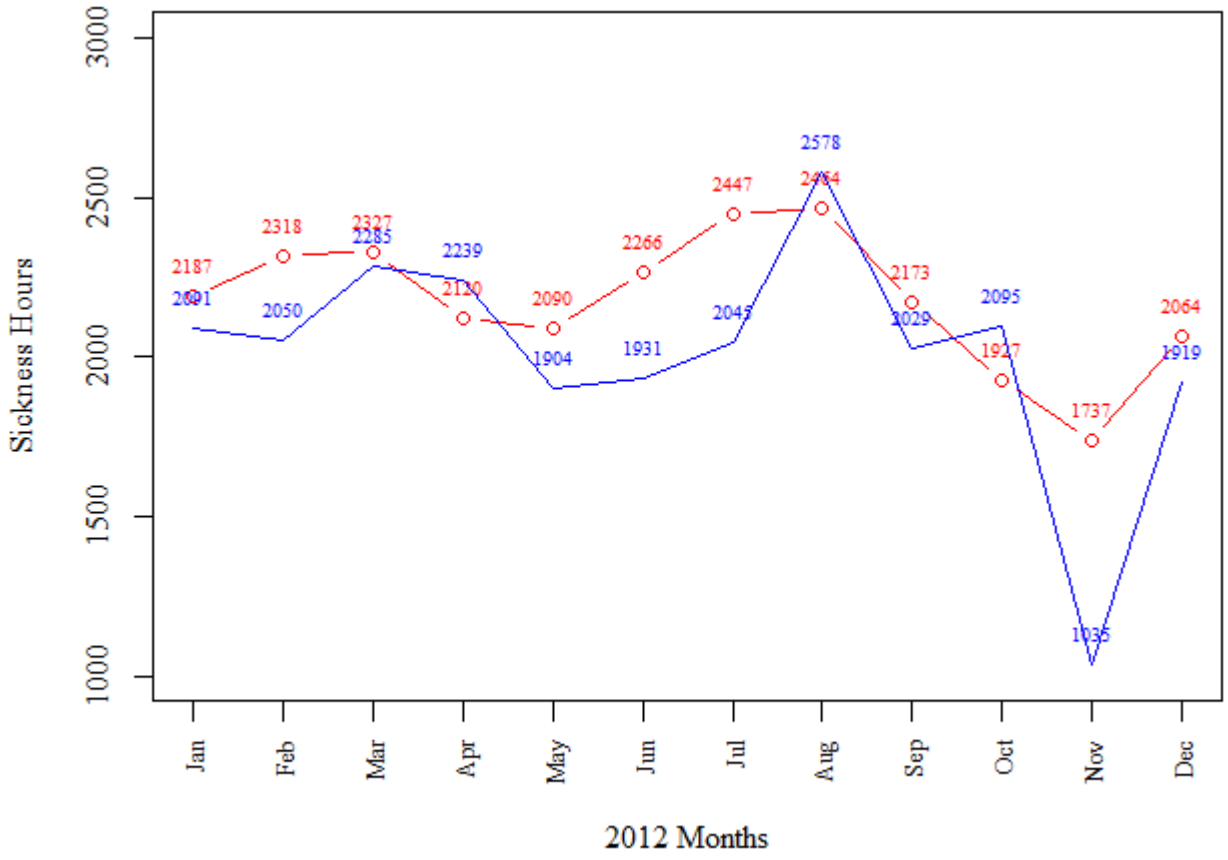


Figure 5-15: Predictions versus Observation for Position 1 in 2012. The blue line is the actual sickness and the red one is sick predciction.

Figure 5-15 plots the prediction versus observation for sickness hours in Position 1. It can be seen that the model is able to determine the trend of sickness in general. Sickness in November 2012 in Position 1 is extremely lower than other months. This outlier is an exception among the three years

data used in this study. Like many other methods, our model is not able to predict theses kind of outliers.

The monthly predictions are the weighted mean of sickness estimations relative to each model. Table 5.5 shows all these estimations, $\hat{s}(\mathfrak{T}_{1\,j}, \mathbb{S}_{1\,k})$, for Position 1 and 2012 months. The cells in blue are the four best estimations for each month and the cells in red are those cells with an error less than 5 percent.

Table 5.5: Sickness estimations relative to each model ($\mathfrak{T}_{1\,j}, 12 \leq j \leq 34$). Cells in blue determine 4 best estimations for each month and cells with red font are those cells with less than 5 percent of error.

Model	*Jan*	*Feb*	*Mar*	*Apr*	*May*	*Jun*	*Jul*	*Aug*	*Sep*	*Oct*	*Nov*	*Dec*
12	2242	2343	**2357**	**2151**	2127	2310	2501	**2525**	2223	1975	1784	2110
13	**2062**	2422	2435	**2238**	2183	2398	2578	**2602**	2303	1972	1774	2124
14	2274	2376	**2392**	**2177**	2151	2337	2530	**2554**	2246	1984	1792	2140
15	2246	2347	**2361**	**2154**	2130	2314	2505	**2529**	2226	1978	1786	2113
16	2239	2340	**2355**	**2150**	2125	2308	2498	**2521**	2222	1974	1782	2107
17	2213	2314	**2328**	2127	2102	2282	2469	**2492**	2197	1954	1761	2083
18	**2181**	2280	**2301**	2097	2074	2251	2437	**2456**	2167	1934	1743	2061
19	**2177**	2275	**2289**	2091	2067	2244	2428	**2451**	2161	1921	1732	2049
20	**2169**	2267	**2281**	2085	2060	2237	2420	2442	2154	1915	1726	2042
21	**2171**	2270	**2284**	2086	2062	2239	2422	2444	2156	1916	1728	2043
22	**2166**	2264	**2278**	2082	2057	2234	2417	2439	2150	1912	1724	2039
23	**2152**	2250	**2264**	2069	2044	2220	2401	2423	2137	1900	1713	2026
24		2255	**2266**	2064	2040	2215	2396	2419	2132	1895	1709	2022
25			**2263**	2061	2037	2212	2394	2416	**2129**	1893	1707	2019
26				2066	2040	2218	2403	2425	2135	1887	1704	2026
27					2045	2223	2408	2431	2140	1891	1707	2037
28						2218	2400	2422	2135	1898	1712	2025
29							2396	2418	2132	1895	1709	2021
30								2406	**2121**	1886	1700	**2011**
31									2133	1880	1701	2025
32										1890	1702	**2013**
33											1711	2036
34												2022
Actual	***2091***	***2050***	***2285***	***2239***	***1904***	***1931***	***2045***	***2578***	***2029***	***2095***	***1035***	***1919***
prediction	2187	2318	2327	2120	2090	2266	2447	2464	2173	1927	1737	2064
error	4.6%	13.1%	1.8%	-5.3%	9.8%	17.3%	19.7%	-4.4%	7.1%	-8.0%	67.8%	7.6%

This table shows that there is no evident structure for model selection. For example, $\mathfrak{T}_{1\,12}$ is the tree that gives best prediction for January 2011, but applying this model for January 2012 schedule table does not give a good estimation. On the other hand, best subset of estimations for each month,

cells in blue, or estimations with error less than a pre-defined threshold, cells in red font, do not have an evident structure. Therefore, before doing a deep study on model selection for this problem, the weighted mean of all estimations gives the best possible predictions.

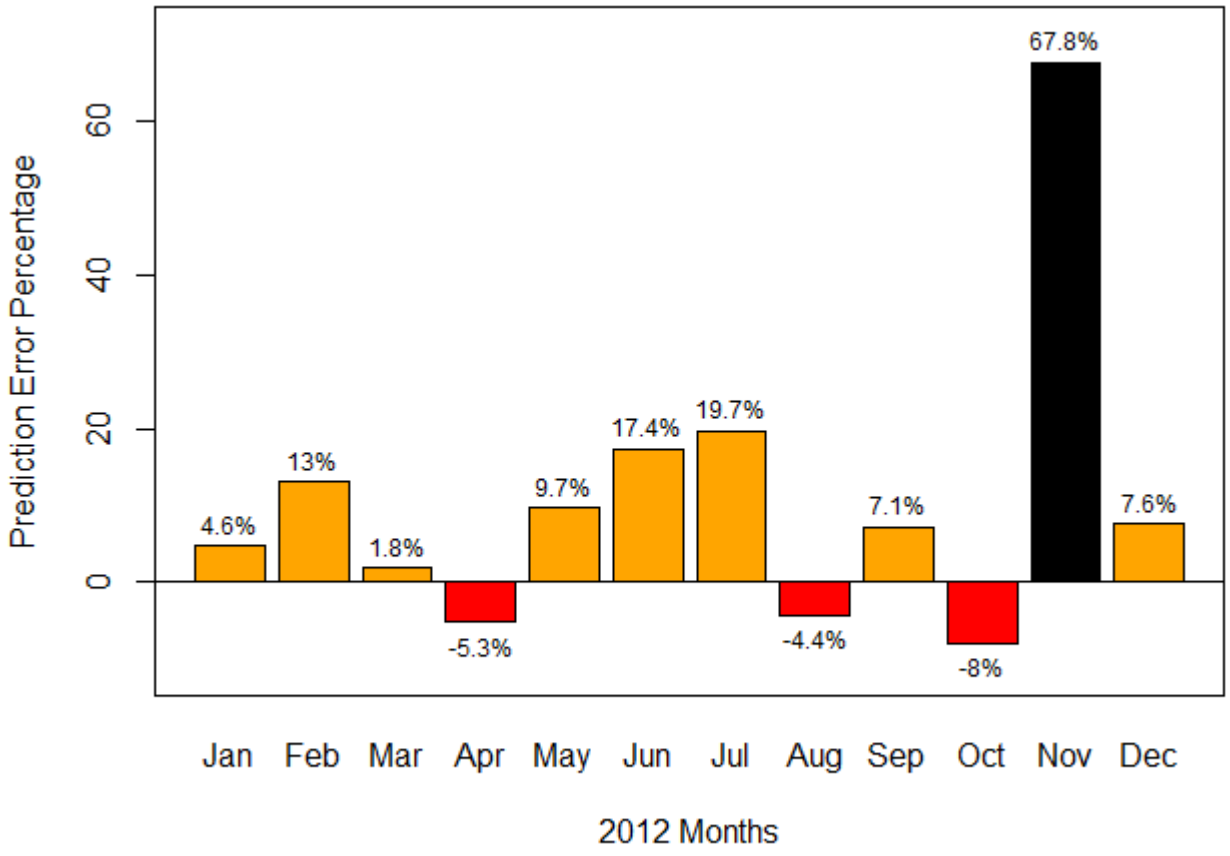


Figure 5-16: Percentage of prediction error for Position 1 in 2012. The orange bars are months with over prediction, the red bars are the months with under prediction, and the black bar is related to one sickness outlier in 2012.

Figure 5-16: Percentage of prediction error for Position 1 in 2012. The orange bars are months with over prediction, the red bars are the months with under prediction, and the black bar is related to one sickness outlier in 2012. is a bar plot for percentage of prediction error for position 1 in 2012. These percentage errors can be calculated as $\frac{(\hat{s}_{ij}-s_{ij})\times 100}{s_{ij}}$. Except the outlier in November, for other months the error of prediction is acceptable, especially because the under prediction errors are small and rare.

The last pre-test for the proposed methodology was a comparison of predictions with current method applied in the airline company. The comparison has been done for 11 positions out of 13 current positions in the airline company. The comparison was not possible in Positions 12 and 13 in Table 5.1 because some schedule tables were not available and the process of remaking these tables was time consuming.

The results of comparison show that in 6 positions the proposed method result better than the current model. These 6 positions cover 84 percent of annual flight hours of the airline company. The other 5 positions are the small positions with less than 100 pilots in each position.

Table 5.6: Comparison of annual prediction error (in hours) between current model of the airline company and new proposed model, for main positions in 2012.

Positions	**Current Model**					**New Model**				
	1	*1.25*	*1.5*	*1.75*	*2*	*1*	*1.25*	*1.5*	*1.75*	*2*
1	3277	3695	4113	4530	4948	2652	2741	2829	2917	3006
2	3890	4732	5574	6416	7258	3461	4081	4702	5323	5944
7	1237	1487	1737	1987	2237	1039	1110	1181	1252	1323
8	1519	1822	2125	2429	2732	1332	1501	1669	1837	2006
9	1605	1841	2078	2314	2550	1356	1499	1642	1786	1929
10	1576	1913	2249	2586	2923	1515	1814	2113	2412	2711
Total	13104	15490	17876	20263	22649	11355	12746	14137	15528	16918
					Improvement in Prediction	**13%**	**18%**	**21%**	**23%**	**25%**

Table 5.6 represents error of the prediction for both models in hours for different cost rate of under prediction (1, 1.25, 1.5, 1.75, and 2). A cost rate of under prediction equals to 2 means that we multiply the prediction error by 2 in the case of under prediction error.

If we consider a cost rate of under prediction equal to 1, i.e. the same cost for under and over prediction error, the proposed method improved 13 percent the predictions and it has 1749 less error hours in comparison with the current method. However the cost of under prediction in the airline companies is higher than the cost of over prediction because of the expensive costs of flight cancelations and expenses related to calling a non-scheduled pilot to fly a pairing. Therefore we can say that the minimum improvement of the proposed method is at least 13 percent.

5.4 Decision support system

As an applied data mining study, the results of this thesis must be applicable in business. That means the methods and procedures must be executed in an automatic way and be able to help managers making better decisions. Monthly sickness predictions are used at the operations level in the airline industry and therefore the proposed method must be automated like an operations level application.

This goal leads us to make a decision support system (DSS). A DSS is an automatic computer program that helps in the process of decision making by providing related information, graphics, and statistics.

We created a user-friendly web application as the DSS for monthly sickness prediction. The codes have been written in R programming language by using *shiny* library. Both server and user interface codes are presented in Appendix 1 and Appendix 2. Screenshots of different tabs of the first version of this application are shown in Appendix 3.

The first version of the application has 3 main tabs dus a help tab. Each main tab consists of a side panel and a main panel. Here, we explain available functions of the application.

1) *Prediction tab*: In the prediction tab of the application, a report for the new month prediction is presented with some plots that determine the performance of the model.
    a) Main Panel:
        i) Prediction for the new month based on the explained methodology in Chapter 4 is reported.
        ii) Comparison between actual sickness and model predictions in the previous months is plotted. The user can determine the period of comparison.
        iii) For all months in the comparison period percentage of over-prediction and under-prediction errors are shown in a table.
        iv) Overall percentages of under and over prediction errors are presented in donut chart.
    b) Side Panel:

i) *Position* is the option to change the report based on different positions. List of available positions is created automatically from the database table.

ii) The user can determine a specific month in the past to have an idea of the behaviour of the method in previous months.

2) *Comparison Tab:* In the comparison tab of the application, it is possible to compare the model with another model and choose the best model.

a) Main Panel:

i) Monthly percentage of prediction error for each of the models is presented in a side by side bar plot. The cost of under-prediction is applied.

ii) Overall percentage of prediction error for the two models is compared in a donut chart. The cost of under-prediction is applied.

b) Side Panel:

i) *Position* is the option for changing the report based on different positions.

ii) *Choose file:* the values for alternative model can be uploaded by using a text file.

iii) The user can determine a specific value for the cost of under-prediction. The percentage of error is multiplied by this value in the case of under-prediction.

3) *Descriptive statistics:* The descriptive statistics tab shows the individual estimations before applying weighted mean method. This is applicable when there is some other prior information and the user can choose the minimum or maximum or any of the grouped mean as the prediction for new month.

a) Main Panel:

i) Descriptive statistics for the selected position and month. These statistics describe the prediction for different models.

ii) The plot of all the predictions based on different models is presented.

iii) Grouped mean is the mean of predictions in every interval shown in the plot of all the predictions.

b) Side Panel:

i) *Position* is the option for changing the report based on different positions.

ii) If the prediction is done for more than one month, the user can determine the month to show the details of the prediction.

CONCLUSION

Unwanted events are always a big challenge for airline companies and having good predictions is a helpful tool for disruption management during operations. Absenteeism of the pilots is one of those unwanted events that may cause flight delay, flight cancelation, customer dissatisfaction, etc besides the costs of substituting a reserve pilot. For these reasons, determining the number of reserve pilots with minimum error is an important task for airlines. Calling sick by the pilots is one of the most important reasons of absenteeism among the cabin crew, hence we focused on predicting monthly sick hours for pilots.

Our method for attacking this prediction problem was considering monthly sickness hours as the sum of the pairing hours in which the pilot is sick. In this approach, we could relate the response variable to the pairing characteristics as the explanatory variables. This means the prediction is based on the monthly schedule and will change as the schedule changes. The proposed iterative algorithm determines the best possible scenarios of previous months and predicts a future monthly sickness as the weighted mean of the results of applying these scenarios to the new schedule.

Results of applying this method to real data show that in most cases the predictions have an acceptable error and the proposed methodology improved at least 13 percent the monthly sickness predictions for 2012 in comparison with the current method of prediction in the airline. This means, for the pre-test period, whole year of 2012, the error of prediction of our model was 1749 hours less than the error of prediction in the current model of airline. This improvement is achieved when we consider that the cost of under prediction is equal to the cost of over prediction. In other words, when the cost rate of under prediction (η) is 1. Table 5.6 shows that the improvement percentage in the error of prediction is increasing with respect to the cost rate of under prediction (η). Hence, we can say that the improvement of prediction is 21 percent with a cost ratio of under prediction equal to 1.5, which is more close to the real value of cost ratio used in the airlines.

The decision support system that is created for this methodology makes its application considerably easy and completely automatic. It provides extra information for helping the managers in making the best possible decision in a user friendly manner. The algorithm is able to determine outliers automatically and decrease the effect of the outliers on the future predictions. The proposed decision support system is the main contribution of this thesis in the applied area.

Although the proposed model has the previously explained advantages and improvements, like any other methodology, it has its own limits and restrictions. The predictions are over smooth and they do not behave well in the case of outliers. Also the pre tests show that we cannot use this method for the small positions with a large variation in the monthly sick hours.

This field of research can be developed in different aspects. One of the open problems is the daily prediction for the sick hours. This could be done by considering the awarded pairing table and at the time that we know which pilot is assigned to which pairing.

As it can be seen in Table 5.5, among the different trees that exist for each month, some of them give better predictions. It can be concerned to apply model selection methods for choosing the best set of the trees for each month. This can be done by developing the similarity vector discussed in Section 4.4.1.

Focusing on the outliers and developing a model for predicting outliers is another possible improvement of the current model. As mentioned before, the methodology is not sensible enough in the case of outliers and the predictions are over smooth, so it is a field for future developments. It must be noticed that in the airlines, the certainty of the outlier predictor must be extremely high because of the nature of the business and high cost of under prediction of absenteeism.

BIBLIOGRAPHY

Abdelghany, A., Ekollu, G., Narasimhan, R., and Abdelghany, K. (2004). A proactive crew recovery decision support tool for commercial airlines during irregular operations. *Annals of Operations Research, 127*(1–4), 309–331.

Almuallim, H., and Dietterich, T. G. (1994). Learning Boolean concepts in the presence of many irrelevant features. *Artificial Intelligence, 69*(1), 279–305.

Barnhart, C., Belobaba, P., Odoni, A. R., and Barnhart, C. (2003). Applications of Operations Research in the Air Transport Industry (Vol. 37, pp. 368–391).

Bazargan, M. (2010). *Airline operations and scheduling* (2nd ed.). Farnham, Surrey, England: Ashgate.

Bohanec, M., and Bratko, I. (1994). Trading accuracy for simplicity in decision trees. *Machine Learning, 15*(3), 223–250.

Bratu, S., and Barnhart, C. (2006). Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling, 9*(3), 279–298.

Breiman, L. (1993). *Classification and regression trees*. Belmont, California: CRC press.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees. Wadsworth & Brooks*. Monterey, CA: Wadsworth International Group.

Cauvin, A., Ferrarini, A., and Tranvouez, E. (2009). Disruption management in distributed enterprises: A multi-agent modelling and simulation of cooperative recovery behaviours. *International Journal of Production Economics, 122*(1), 429–439.

Chan, P. K., and Stolfo, S. J. (1997). On the accuracy of meta-learning for scalable data mining. *Journal of Intelligent Information Systems, 8*(1), 5–28.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian CART model search. *Journal of the American Statistical Association, 93*(443), 935–948.

Ciampi, A., Chang, C.-H., Hogg, S., and McKinney, S. (1987). Recursive Partition: A Versatile Method for Exploratory Data Analysis in Biostatistics. *Biostatistics, 38*, 23–50.

Clausen, J., Larsen, A., Larsen, J., and Rezanova, N. J. (2010). Disruption management in the airline industry—concepts, models and methods. *Computers & Operations Research, 37*(5), 809–821.

Dillon, J. E., and Kontogiorgis, S. (1999). US Airways optimizes the scheduling of reserve flight crews. *Interfaces, 29*(5), 123–131.

Esposito, F., Malerba, D., Semeraro, G., and Kay, J. (1997). A comparative analysis of methods for pruning decision trees. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19*(5), 476–491.

Gaballa, A. (1979). Planning Callout Reserves for Aircraft Delays. *Interfaces, 9*(2-Part-2), 78–86.

Gamache, M., Soumis, F., Villeneuve, D., Desrosiers, J., and Gelinas, E. (1998). The Preferential Bidding System at Air Canada. *32*(3), 246–255.

Grosche, T. (2009). *Computational intelligence in integrated airline scheduling*. Berlin, Germany: Springer-Verlag.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, US: Springer.

Jarrah, A. I., Yu, G., Krishnamurthy, N., and Rakshit, A. (1993). A decision support framework for airline flight cancellations and delays. *Transportation Science, 27*(3), 266–280.

Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Applied statistics, 29*(2), 119–127.

Kohl, N., Larsen, A., Larsen, J., Ross, A., and Tiourine, S. (2007). Airline disruption management—perspectives, experiences and outlook. *Journal of Air Transport Management, 13*(3), 149–162.

Lettovský, L., Johnson, E. L., and Nemhauser, G. L. (2000). Airline crew recovery. *Transportation Science, 34*(4), 337–348.

Mehta, M., Agrawal, R., and Rissanen, J. (1996). SLIQ: A fast scalable classifier for data mining. *Lecture Notes in Computer Science, 1057*, 18–32.

Milborrow, S. (2012). rpart.plot: Plot rpart models. An enhanced version of plot.rpart. R package version 1.4-3. Retrieved from http://CRAN.R-project.org/package=rpart.plot

Morgan, J. N., and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association, 58*(302), 415–434.

Niblett, T., and Bratko, I. (1987). *Learning decision rules in noisy domains.* Paper presented at the Proceedings of Expert Systems' 86, The 6Th Annual Technical Conference on Research and development in expert systems III.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

Quinlan, J. R. (1987). Simplifying decision trees. *International journal of man-machine studies, 27*(3), 221–234.

Quinlan, J. R. (1993). *C4. 5: programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

R Development Core Team. (2005). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org/

Rakshit, A., Krishnamurthy, N., and Yu, G. (1996). System operations advisor: a real-time decision support system for managing airline operations at united airlines. *Interfaces, 26*(2), 50–58.

Ripley, B. (2012). RODBC: ODBC Database Access (2011). R package version 1.3-3. Retrieved from http://CRAN.R-project.org/package=RODBC

Rokach, L., and Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 35*(4), 476–487.

RStudio and Inc. (2013). shiny: Web Application Framework for R. R package version

0.6.0 Retrieved from http://CRAN.R-project.org/package=shiny

Sethi, I. K., and Yoo, J. H. (1994). Design of multicategory multifeature split decision trees using perceptron learning. *Pattern Recognition, 27*(7), 939–947.

Shafer, J., Agrawal, R., and Mehta, M. (1996). *SPRINT: A scalable parallel classifier for data mining.* Paper presented at the International Conference on Very Large Data Bases.

Sohoni, M. G., Johnson, E. L., and Bailey, T. G. (2006). Operational airline reserve crew planning. *Journal of Scheduling, 9*(3), 203–221.

Therneau, T. M., Atkinson, B., and Ripley, B. (2010). rpart: Recursive partitioning. R package version 4.1-1. Retrieved from http://CRAN.R-project.org/package=rpart

Venables, W. N., Smith, D. M., and Team, R. D. C. (2002). *An introduction to R* Retrieved from http://cran.wustl.edu/doc/manuals/R-intro.pdf

Verzani, J. (2012). gWidgets: gWidgets API for building toolkit-independent, interactive GUIs. R package version 0.0-52. Retrieved from http://CRAN.R-project.org/package=gWidgets

Watson, G. S. (1964). Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, *26*(4), 359–372.

Wei, G., Yu, G., and Song, M. (1997). Optimization model and algorithm for crew management during airline irregular operations. *Journal of Combinatorial Optimization, 1*(3), 305–321.

Wickham, H. (2007). Reshaping Data with the reshape Package. *Journal of Statistical Software, 21*(12), 1–20.

Wickham, H. (2009). *ggplot2: elegant graphics for data analysis*. New York, NY, US: Springer.

Witten, I., Frank, E., and Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. US: Morgan Kaufmann.

Yuan, Y., and Shaw, M. J. (1995). Induction of fuzzy decision trees. *Fuzzy Sets and systems, 69*(2), 125–139.

APPENDIX A – R CODES FOR WEB APPLICATION: SERVER

```
library(shiny)
library(rpart)
library(ggplot2)
load('Trees.RData')
load('NewPairingFinal.RData')
tree <- Trees[[1]]
Data <- NewPairingByPosition[[1]]
#FUNCTIONS
Prediction <- function(PredictData,fit,similar){
  Forecast <- vector()
  Predict <- list()
  for(j in 1:length(levels(PredictData$BP))){
    test <- PredictData[which(PredictData$BP==levels(PredictData$BP)[j]),]
    temp <- sapply(1:length(fit$error), function(k)
      round(sum(predict(fit$FinalFit[[k]],
                        newdata=test,type="prob")[,2]*test$TC)/60,1))
    weights <- 1/sqrt(fit$error)/sum(1/sqrt(fit$error))
    weights <- 0.5*weights+0.5*similar[[j]]
    Forecast[j] <- sum(weights*temp)
    Predict[[j]] <- temp
  }
  Forecast <- round(Forecast)
  Predict$Forecast <- Forecast
  return(Predict)
}
AlterPredict <- function(inFile,tree,Position){
  file <- read.csv(inFile$datapath, header=T,sep='\t')
  names(file)[-1] <- paste(substr(names(file)[-1],1,3),
                           substr(names(file)[-1],5,8))
  file <- file[,-(which(!names(file)[-1] %in%
                          names(tree$observation))+1)]
  observation <- tree$observation
  Npredict <- tree$Simulation[which(names(observation) %in%
                                      names(file))]
  observation <- observation[which(names(observation) %in%
                                     names(file))]
  Opredict <- file[which(file[,1]==Position),-1]
  Opredict <- c(Opredict,recursive=T)
  New <- Npredict-observation
  Old <- Opredict-observation
  OldNew <- t(as.matrix(data.frame(Cur.=Old,New,observation)))
  return(OldNew)
}
# Define required server logic
shinyServer(function(input, output) {
  output$selectPO <- renderUI({
    selectInput('Position','Position',
                names(NewPairingByPosition))
  })
  output$selectSS <- renderUI({
    selectInput('StartSimulation', 'Simulate the predictions for the
Months From',
```

```
                names(Trees[[1]]$observation)[-(1:6)],
                names(Trees[[1]]$observation)[13])
  })
  output$selectAM <- renderUI({
    radioButtons('DetailedMonth','Show the Detailes for the Prediction of
the Month',
                 levels(NewPairingByPosition[[1]]$BP))
  })
  # Return the requested dataset
  datasetInput <- reactive({
    switch(EXPR=input$Position,
           NewPairingByPosition[[input$Position]])
  })
  TreeInput <- reactive({
    switch(EXPR=input$Position,
           Trees[[input$Position]])
  })
  SimilarityInput <- reactive({
    switch(EXPR=input$Position,
           Similarity[[input$Position]])
  })
  # Generate a summary of the dataset
  output$summary <- renderPrint({
    Data <- datasetInput()
    tree <- TreeInput()
    similarity <- SimilarityInput()
    Forecast <- Prediction(PredictData=Data,
                           fit=tree,similar=similarity)
    Forecast <- Forecast[['Forecast']]
    PMonths <- as.character(levels(Data$BP))
    FLMonth <- c(as.character(names(tree$observation)[1]),
                 as.character(tail(names(tree$observation),1)))
    Forecast <- paste('For the Bidding Period',PMonths,':',
                      Forecast,' HOURS')
    Forecast <- c(paste('The Prediction for the Position',
                        input$Position),
                  paste('(Based on the Collected History from ',
                        FLMonth[1],' to ',FLMonth[2],')',sep='')
                  ,Forecast)
    print(as.data.frame(Forecast),row.names=F)
  })
  ssInput <- reactive({
    which(names(tree$observation)==input$StartSimulation)
  })
  output$THPlot <- renderPlot({
    Data <- datasetInput()
    tree <- TreeInput()
    similarity <- SimilarityInput()
    ss <- ssInput()
    Forecast <- Prediction(PredictData=Data,
                           fit=tree,similar=similarity)
    Forecast <- Forecast[['Forecast']]
    observation <- round(tree$observation)
    PredictHour <- tree$Modeling[1:(ss-1)]
```

```
    PredictHour <- c(PredictHour,tree$Simulation[-(1:(ss-1))])
    PredictHour <- c(PredictHour,Forecast)
    PredictHour <- round(PredictHour)
    Ylim <- range(range(PredictHour)+c(-10,10),
                  range(observation)+c(-10,10))
    COLOR <- rep(c('red','brown4'),times=c(
      length(observation),length(Forecast)))
    PCH <- rep(c(1,18),times=c(
      length(observation),length(Forecast)))
    LABLES <- names(tree$observation)
    LABLES <- c(LABLES,levels(NewPairingByPosition[[1]]$BP))
    LABLES <- paste(substr(LABLES,1,3),substr(LABLES,7,8),sep="-")
    plot(1:length(PredictHour), PredictHour, type="b", col=COLOR,
         xlab="Month", ylab="Sick Hours",
         main=paste('Observation vs. Prediction for the
Position:',input$Position),
         ylim=Ylim, xaxt = "n",pch=PCH)
    axis(1, at=1:length(PredictHour), labels=LABLES, las=3,cex.axis=0.75)
    text(1:length(PredictHour), PredictHour+round(.02*(Ylim[2]-Ylim[1])),
         PredictHour, col=COLOR, cex=0.6)
    lines(1:length(observation), observation, col="blue")
    text(1:length(observation), observation+round(.02*(Ylim[2]-Ylim[1])),
         observation, col="blue", cex=0.6)
    rect(-5,-1000,ss-1,10000,density=10,col="lightgreen")
    box()
    legend("topright",c("Pre.","Obs."), lty=c(1,1),
           col=c("red","blue"), lwd=c(2.5,2.5),cex=0.5)
  })
  output$view <- renderTable({
    tree <- TreeInput()
    ss <- ssInput()
    observation <- tree$observation[-(1:ss-1)]
    Simulation <- tree$Simulation[-(1:ss-1)]
    OUPrediction <- (Simulation - observation)*100 / observation
    OUPrediction <- round(OUPrediction,1)
    OverPrediction <- ifelse(OUPrediction>0,
                             paste(OUPrediction,'%',sep=''),'')
    UnderPrediction <- ifelse(OUPrediction<0,
                              paste(OUPrediction,'%',sep=''),'')
    OUPrediction <- t(cbind.data.frame(OverPrediction,UnderPrediction))
  })
  output$PNError <- renderPlot({
    tree <- TreeInput()
    ss <- ssInput()
    observation <- round(tree$observation)[-(1:ss-1)]
    Simulation <- round(tree$Simulation)[-(1:ss-1)]
    OUPrediction <- (Simulation - observation)
    PError <-
sum(OUPrediction[which(OUPrediction>0)])*100/sum(observation[which(OUPredi
ction>0)])
    PError <- round(PError,1)
    NError <- -
sum(OUPrediction[which(OUPrediction<0)])*100/sum(observation[which(OUPredi
ction<0)])
```

```
    NError <- round(NError,1)
    dat1 = data.frame(ymin=c(0,100-NError,100,100+PError)/2,
                      ymax=c(100-NError,100,100+PError,200)/2,
                      category=factor(c('','Under Pred.','Over
Pred.','')))
    p <- ggplot(dat1, aes(fill=category, ymax=ymax, ymin=ymin, xmax=4,
xmin=3)) +
      geom_rect()+
      scale_fill_manual(values=c('white','navy','red')) +
      guides(fill=guide_legend(title=NULL)) +
      coord_polar(theta="y",start=pi/2) +
      annotate("text", x=3.5, y=dat1[4,1]+PError/5,
               colour='navy', label=paste(PError,'% of O.P.E.',sep=''))+
      annotate("text", x=3.5, y=dat1[2,1]-NError/5,
               colour='red', label=paste(NError,'% U.P.E.',sep=''))+
      xlim(c(0, 4)) +
      theme_bw() +
      theme(panel.grid=element_blank()) +
      theme(axis.text=element_blank()) +
      theme(axis.ticks=element_blank()) +
      xlab("") +
      ylab("") +
      ggtitle("Total Prediction Error Percentages for the Simulation
Period") +
      theme(plot.title = element_text(vjust=1,lineheight=.8, face="bold"))
    print(p)
  })
  output$monthlyCompare <- renderPlot({
    inFile <- input$file
    if(ncol(inFile) > 13){
      inFile <- inFile[,c(1,(ncol(inFile)-11):ncol(inFile))]
    }
    tree <- TreeInput()
    OldNew <- AlterPredict(inFile=inFile,tree=tree,
                           Position=input$Position)
    OldNew <- ifelse(OldNew>0, OldNew,(-input$cost)*(OldNew))
    OldNew[1,] <- round(OldNew[1,]*100/OldNew[3,],1)
    OldNew[2,] <- round(OldNew[2,]*100/OldNew[3,],1)
    OldNew <- OldNew[1:2,]
    b <- barplot(OldNew, main="Monthly Error Percentages of Current and
New Methods",
                 col=c("green","red"), ylim = c(0,max(OldNew)+10),
                 xlim=c(0,3*ncol(OldNew)+3),legend=rownames(OldNew)
,beside=TRUE)
    text(x=b, y=OldNew+1, labels=OldNew, col='black',cex=0.75)
    box()
  })
  output$totalCompare <- renderPlot({
    inFile <- input$file
    if(ncol(inFile) > 13){
      inFile <- inFile[,c(1,(ncol(inFile)-11):ncol(inFile))]
    }
    tree <- TreeInput()
    OldNew <- AlterPredict(inFile=inFile,tree=tree,
```

```
                            Position=input$Position)
    OldNew <- ifelse(OldNew>0, OldNew,(-input$cost)*(OldNew))
    OldNew <- rowSums(OldNew)
    Old <- round(OldNew[1]*100/OldNew[3],1)
    New <- round(OldNew[2]*100/OldNew[3],1)
    dat = data.frame(ymin=c(0,100-Old,100,100+New)/2,
                     ymax=c(100-Old,100,100+New,200)/2,
                     category=factor(c('','Cur.','New','')))
    p <- ggplot(dat, aes(fill=category, ymax=ymax, ymin=ymin, xmax=4,
xmin=3)) +
      geom_rect()+
      scale_fill_manual(values=c('white','green','red')) +
      guides(fill=guide_legend(title=NULL)) +
      coord_polar(theta="y",start=pi/2) +
      annotate("text", x=3.5, y=dat[4,1]+New/5,
               colour='red', label=paste(New,'%',sep=''))+
      annotate("text", x=3.5, y=dat[2,1]-Old/5,
               colour='green', label=paste(Old,'%',sep=''))+
      xlim(c(0, 4)) +
      theme_bw() +
      theme(panel.grid=element_blank()) +
      theme(axis.text=element_blank()) +
      theme(axis.ticks=element_blank()) +
      xlab("") +
      ylab("") +
      labs(title="Total Error Percentages of Old and New Methods") +
      theme(plot.title = element_text(vjust=1,lineheight=.8, face="bold"))
    print(p)
  })
  AM <- reactive({
    which(levels(Data$BP)==input$DetailedMonth)
  })
  output$DescTabTitle <- renderText({
    paste("Descriptive Statistics for Position",
          input$Position, "in", input$DetailedMonth)
  })
  output$MonthSummary <- renderPrint({
    Data <- datasetInput()
    tree <- TreeInput()
    similarity <- SimilarityInput()
    am <- AM()
    Forecast <- Prediction(PredictData=Data,
                           fit=tree,similar=similarity)
    summary(Forecast[[am]])
  })
  output$MonthSummaryPot <- renderPlot({
    Data <- datasetInput()
    tree <- TreeInput()
    similarity <- SimilarityInput()
    am <- AM()
    Forecast <- Prediction(PredictData=Data,
                           fit=tree,similar=similarity)
    mean <- mean(Forecast[[am]])
    sd <- sd(Forecast[[am]])
```

```
    l <- length(Forecast[[am]])
    YLIM <- 0.025 * max(Forecast[[am]])
    plot(Forecast[[am]],col=c('red','green','blue','brown4')
         [as.numeric(cut(x=Forecast[[am]],
                         breaks=c(0,mean-
sd,mean,mean+sd,10000),labels=1:4))],
         pch=19,ylab='Predicted Sick Hours', xlim=c(0,l+4),
         ylim=c(min(Forecast[[am]])-YLIM,max(Forecast[[am]])+YLIM),
         main="All Predictions Based on Different Models",xlab='Model')
    lines(cbind(1:l, mean), lty=2)
    lines(cbind(1:l, mean+sd), lty=2)
    lines(cbind(1:l, mean-sd), lty=2)
    text(cbind(l+2,c(mean-sd,mean,mean+sd)+5),
         c("mean-sd","mean","mean+sd"))
  })
  output$groupedMean <- renderTable({
    Data <- datasetInput()
    tree <- TreeInput()
    similarity <- SimilarityInput()
    am <- AM()
    Forecast <- Prediction(PredictData=Data,
                           fit=tree,similar=similarity)
    mean <- mean(Forecast[[am]])
    sd <- sd(Forecast[[am]])
    GMean <- tapply(Forecast[[am]],
                    cut(x=Forecast[[am]],
                        breaks=c(0,mean-sd,mean,mean+sd,10000),
                        labels=1:4),mean)
    GMean <- round(GMean)
    names(GMean) <- paste('Group', names(GMean))
    GMean <- as.data.frame(GMean)
    GMean <- t(GMean)
    row.names(GMean) <- 'Grouped Mean'
    GMean <- ifelse(is.na(GMean),'-',as.character(GMean))
    GMean
  })
})
```

APPENDIX B – R CODES FOR WEB APPLICATION: UI

```
library(shiny)

# Define UI
shinyUI(pageWithSidebar(

  # Application title
  headerPanel("Prediction Sickness Hours for the Pilots"),

  sidebarPanel(
    img(src="air-canada-logo.jpg"),
    img(src="Poly_100.png"),
    conditionalPanel(
      condition = "$('li.active a').first().html()!=='Help'",
      htmlOutput("selectPO")
    ),
    br(),
    conditionalPanel(
      condition = "$('li.active a').first().html()==='Prediction'",
      htmlOutput("selectSS")
      ),
    conditionalPanel(
      condition = "$('li.active a').first().html()==='Descriptive'",
      htmlOutput("selectAM")
    ),
    conditionalPanel(
      condition = "$('li.active a').first().html()==='Comparison'",
      fileInput('file', 'Choose text/CSV File of the Current
                Prediction Values',
                accept=c('text/csv',
                         'text/comma-separated-values,text/plain'))),
    conditionalPanel(
      condition = "$('li.active a').first().html()==='Comparison'",
      sliderInput("cost",
                  "Cost of Under Prediction:",
                  value = 1, min = 1, max = 2, step = 0.05))
  ),

  mainPanel(
    tabsetPanel(
      tabPanel('Prediction',
               verbatimTextOutput("summary"),
               plotOutput("THPlot"),
               br(),
               HTML("<h5><center>Percentage of Errors in Simulation
Months</center></h5>"),
               tableOutput("view"),
               plotOutput("PNError")),
      tabPanel('Comparison',
               plotOutput("monthlyCompare"),
               plotOutput("totalCompare")
               ),
      tabPanel('Descriptive',
```

```
                h5(textOutput('DescTabTitle')),
                verbatimTextOutput('MonthSummary'),
                br(),
                plotOutput('MonthSummaryPot'),
                br(),
                HTML("<h5>Grouped Means</h5>"),
                tableOutput('groupedMean')
                ),
      tabPanel('Help',
                helpText(HTML('<p><b><font color="red">Position</font></b>
                              : The position for which you want to see the
analysis.</p>
                              <p><h4>Prediction tab</h4></p>
                              <p>The <i>forecast </i>for desired months
will be estimated as the weighted mean
                              of the result of each model that exists in
database. </p>
                              <p>The <i>Observation vs. Prediction plot</i>
in this tab presents a comparison
                              between observed and predicted sick hours
(blue and red lines resp.).  The plot
                              is divided in two parts It is possible
                              to change the place of the border by changing
the <b><font color="red">
                              Start Month</font color></b>. In the green
area of the plot, the observed values of
                              sickness have been used
                              in modeling but in the white part the
prediction values are <b><font color="blue">
                              BLIND PREDICTIONS</font color="blue"> </b>,
which means the predictions have been
                              calculated without considering the observed
value, exactly the same as a future
                              prediction. In other words, a  <b><font
color="blue">SIMULATION</font color="blue">
                              </b> for the past months gives an idea of the
goodness of prediction in future.</p>
                              <p>The <i>Percentage of Errors in Simulation
Months</i> table presents the Error
                              Percentage for each month of the SIMULATION
period. The error percentages have been
                              calculated as<b><i>(Prediction - Observation)
/ Observation</i></b>, for both Over
                              Prediction and Under Prediction errors.</p>
                              <p>The <i>Total Prediction Error Percentage
plot</i> shows both under and over
                              prediction errors in the simulation
period.</p>
                              <br>
                              <p><h4>Comparison tab</h4></p>
                              <p>This tabs help compare the New prediction
with the Current Air Canada
                              prediction.</p>
```

```
                              <p>If the cost of under prediction (c.u.p.)
is greater than the cost of over
                              prediction error, it can be adjusted by using
the <b><font color="red">Cost of
                              under Prediction</b></font color="red">
slider.</p>
                              <p>The percentage of error is
<b><i>(Prediction - Observation) / Observation</i>
                              </b>, in the case of over prediction and is
<b><i>(Observation - Prediction) *
                              c.u.p. / Observation</i></b>, in the case of
under prediction.</p>
                              <p>The <i>Monthly Error Percentage</i> plot
shows the error percentage for
                              current and new models month by month; and
the <i>Total Error Percentage</i>
                              shows the comparison for all the months.</p>
                              <br>
                              <p><h4>Descriptive tab</h4></p>
                              <p>When there is prior information about the
sickness in the future month,
                              maximum or minimum or even a grouped mean can
be a better prediction in comparison
                              with the proposed weighted mean.</p>
                              <p>So in this tab, <i>All Predictions Based
on Different Models</i> have been
                              plotted in a graphic. The values have been
classified in 4 groups based on the
                              mean and standard deviation.</p>
                              <p>The <i>Descriptive Statistics</i> and
<i>Grouped Means</i> have been presented
                              in two tables. </p>
                              <br>
                              <br>'))
               )
      )
    )
))
```

APPENDIX C – SHINY WEB APPLICATION SCREENSHOTS

Prediction Sickness Hours for the Pilots

AIRLINE COMPANY

POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE

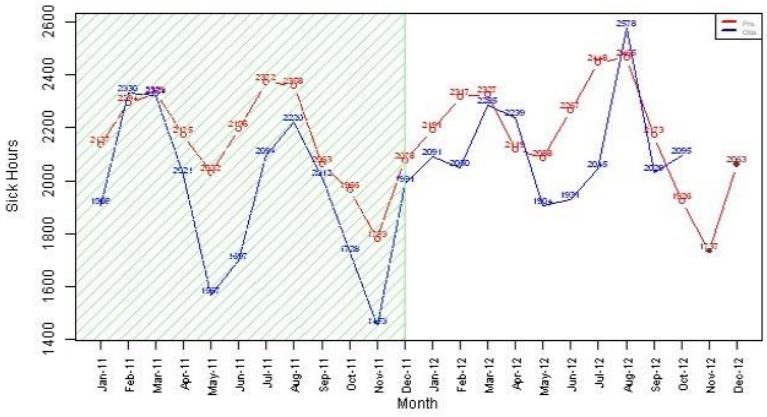
Position
320 CA

Simulate the predictions for the Months From
Jan 2012

Prediction | Comparison | Descriptive | Help

Forecast
The Prediction for the Position 320 CA
(Based on the Collected History from Jan 2011 to Oct 2012)
For the Bidding Period Nov 2012 : 1737 HOURS
For the Bidding Period Dec 2012 : 2063 HOURS

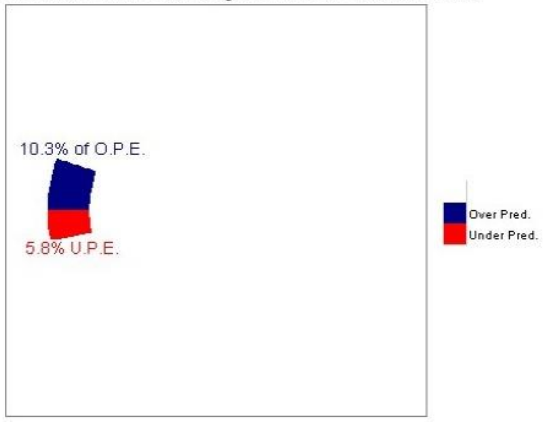
Observation vs. Prediction for the Position: 320 CA



Percentage of Errors in Simulation Months

	Jan 2012	Feb 2012	Mar 2012	Apr 2012	May 2012	Jun 2012	Jul 2012	Aug 2012	Sep 2012	Oct 2012
OverPrediction	4.8%	13%	1.8%		9.6%	17.4%	19.7%		7.1%	
UnderPrediction				-5.4%				-4.4%		-8%

Total Prediction Error Percentages for the Simulation Period



Prediction Sickness Hours for the Pilots

AIRLINE COMPANY

POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE

Position

320 CA

Choose text/CSV File of the Current Prediction Values

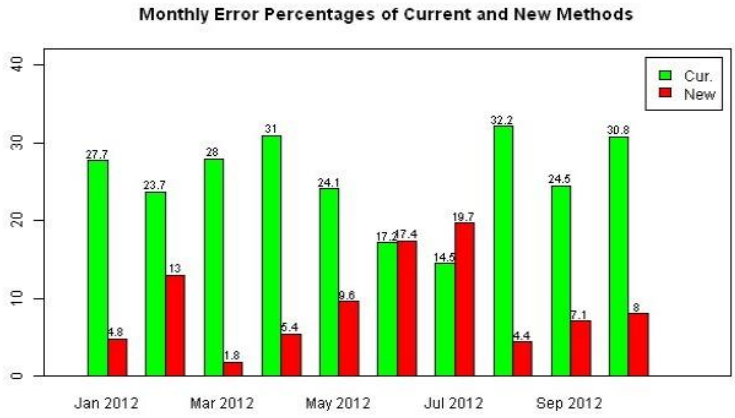
Choose File prediction.txt

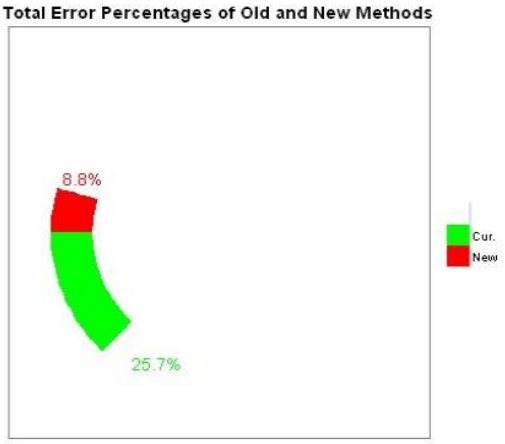
Upload complete

Cost of Under Prediction:

1 2

Prediction | Comparison | Descriptive | Help





Prediction Sickness Hours for the Pilots

AIRLINE COMPANY

POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE

Position

320 CA

Show the Detailes for the Prediction of the Month

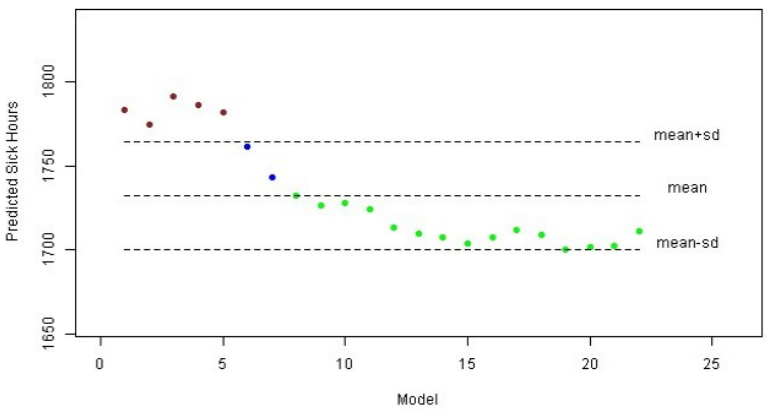
- (•) Nov 2012
- ( ) Dec 2012

Prediction | Comparison | Descriptive | Help

Descriptive Statistics for Position 320 CA in Nov 2012

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1700    1708    1718    1732    1757    1792
```

All Predictions Based on Different Models



Grouped Means

	Group 1	Group 2	Group 3	Group 4
Grouped Mean	-	1712	1752	1783

Prediction Sickness Hours for the Pilots

Prediction | Comparison | Descriptive | Help

Position : The position for which you want to see the analysis.

Prediction tab

The *forecast* for desired months will be estimated as the weighted mean of the result of each model that exists in database.

The *Observation vs. Prediction plot* in this tab presents a comparison between observed and predicted sick hours (blue and red lines resp.). The plot is divided in two parts It is possible to change the place of the border by changing the **Start Month**. In the green area of the plot, the observed values of sickness have been used in modeling but in the white part the prediction values are **BLIND PREDICTIONS** , which means the predictions have been calculated without considering the observed value, exactly the same as a future prediction. In other words, a **SIMULATION** for the past months gives an idea of the goodness of prediction in future.

The *Percentage of Errors in Simulation Months* table presents the Error Percentage for each month of the SIMULATION period. The error percentages have been calculated as***(Prediction - Observation)*** / ***Observation***, for both Over Prediction and Under Prediction errors.

The *Total Prediction Error Percentage plot* shows both under and over prediction errors in the simulation period.

Comparison tab

This tabs help compare the New prediction with the Current Air Canada prediction.

If the cost of under prediction (c.u.p.) is greater than the cost of over prediction error, it can be adjusted by using the **Cost of under Prediction** slider.

The percentage of error is ***(Prediction - Observation)*** / ***Observation*** , in the case of over prediction and is ***(Observation - Prediction) * c.u.p.*** / ***Observation***, in the case of under prediction.

The *Monthly Error Percentage* plot shows the error percentage for current and new models month by month; and the *Total Error Percentage* shows the comparison for all the months.

Descriptive tab

When there is prior information about the sickness in the future month, maximum or minimum or even a grouped mean can be a better prediction in comparison with the proposed weighted mean.

So in this tab, *All Predictions Based on Different Models* have been plotted in a graphic. The values have been classified in 4 groups based on the mean and standard deviation.

The *Descriptive Statistics* and *Grouped Means* have been presented in two tables.