# APPLICATION OF MACHINE LEARNING STRATEGIES TO IMPROVE THE PREDICTION OF CHANGES IN THE AIRLINE NETWORK TOPOLOGY

by

**Aleksandra Dervisevic**

**A Thesis**

*Submitted to the Faculty of Purdue University*

*In Partial Fulfillment of the Requirements for the degree of*

**Master of Science in Aeronautics and Astronautics**


School of Aeronautics & Astronautics

West Lafayette, Indiana

December 2020

# THE PURDUE UNIVERSITY GRADUATE SCHOOL
# STATEMENT OF COMMITTEE APPROVAL

**Dr. Daniel DeLaurentis, Chair**

School of Aeronautics and Astronautics

**Dr. Karen Marais**

School of Aeronautics and Astronautics

**Dr. William A. Crossley**

School of Aeronautics and Astronautics

**Approved by:**

Dr. Gregory Blaisdell

*Para Kiki, Mamá, Papá y Marc.*
*Por vuestro apoyo incondicional y*
*por creer en mí cuando perdí la esperanza.*
*Gracias por estar ahí cuando más lo necesitaba.*
*No lo habría conseguido de no ser por vosotros.*
*Os quiero mucho.*

# ACKNOWLEDGMENTS

I would like to thank everyone at Purdue's Department of Aeronautics and Astronautics for almost seven years of dedicated work and effort making my college life so special and meaningful. I am very thankful to all my professors at Purdue, specially Dr. D, Prof. Marais, Prof. Crossley, Prof. Corless and Prof. Kenley for inspiring me to learn and grow as a scientist and as an engineer. I would like to thank my research group, in particular Kushal Moochaldani and Navindran Davendralingam for helping me in the early stages of my thesis. Special thanks to my TA Rohit Tripathy, from the Mechanical Engineering Department, who also played a huge part in the development and understanding of this research.

I would like to acknowledge my great friends, Kolawole Ogunsina, Alejandro Guayaquil and Natasha Crasta for supporting me and being there for me throughout the years. And finally, Purdue's Muay Thai Club and the PAPA Club, for providing me with a space where I was able to grow in other areas outside of academics.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABSTRACT

Predictive modeling allows us to analyze historical patterns to forecast future events. When the data available for this analysis is imbalanced or skewed, many challenges arise. The lack of sensitivity towards the class with less data available hinders the sought-after predictive capabilities of the model. These imbalanced datasets are found across many different fields, including medical imaging, insurance claims and financial frauds. The objective of this thesis is to identify the challenges, and means to assess, the application of machine learning to transportation data that is imbalanced and using only one independent variable.

Airlines undergo a decision-making process on air route addition or deletion in order to adjust the services offered with respect to demand and cost, amongst other criteria. This process greatly affects the topology of the network, and results in a continuously evolving Air Traffic Network (ATN). Organizations like the Federal Aviation Administration (FAA) are interested in the network transformation and the influence airlines have as stakeholders. For this reason, they attempt to model the criteria used by airlines to modify routes. The goal is to be able to predict trends and dependencies observed in the network evolution, by understanding the relation between the number of passengers per flight leg as the single independent variable and the airline's decision to keep or eliminate that route (the dependent variable). Research to date has used optimization-based methods and machine learning algorithms to model airlines' decision-making process on air route addition and deletion, but these studies demonstrate less than a 50% accuracy.

In particular, two machine learning (ML) algorithms are examined: Sparse Gaussian Classification (SGC) and Deep Neural Networks (DNN). SGC is the extension of Gaussian Process Classification models to large datasets. These models use Gaussian Processes (GPs), which are proven to perform well in binary classification problems. DNN uses multiple layers of probabilities between the input and output layers. It is one of the most popular ML algorithms currently in use, so the results obtained using SGC were compared to the DNN model.

At a first glance, these two models appear to perform equally, giving a high accuracy output of 97.77%. However, post-processing the results using a simple Bayes classifier and using the appropriate metrics for measuring the performance of models trained with imbalanced datasets reveals otherwise. The results in both SGC and DNN provided predictions with a 1% of precision and 20% of recall with an $F_1$ score of 0.02 and an AUC (Area Under the Curve) of 0.38

and 0.31 respectively. The low $F_1$ score indicates the classifier is not performing accurately, and the AUC value confirms the inability of the models to differentiate between the classes. This is probably due to the existing interaction and competition of the airlines in the market, which is not captured by the models. Interestingly enough, the behavior of both models is very different across the range of threshold values. The SGC model captured more effectively the low confidence in these results. In order to validate the model, a stratified K-fold cross-validation model was run.

The future application of Gaussian Processes in model-building for decision-making will depend on a clear understanding of its limitations and the imbalanced datasets used in the process, the central purpose of this thesis. Future steps in this investigation include further analysis of the training data as well as the exploration of variable-optimization algorithms. The tuning process of the SGC model could be improved by utilizing optimal hyperparameters and inducing inputs.

# CHAPTER 1. INTRODUCTION

## 1.1 Research Overview

The evolution of the Air Transportation System (ATS) is affected by socioeconomic factors including population growth, government regulations and increase in gross domestic product (Rocha, 2017). Airlines and software companies develop advanced computer models that measure how the profitability of different air routes is influenced by external factors, which affects the route addition and deletion decision-making process. Consequently, the Air Traffic Network (ATN) evolves as a result of the airlines' decisions on route selection, and this evolution can impact the robustness and the flow of traffic (Sha, Moolchandani, Panchal, & DeLaurentis, 2016). The full set of criteria used to measure the profitability of routes, which is affected by all types of external factors, is not openly available to the public. For this reason, researchers have built approximate models with the use of historical data. This historical data is highly skewed, which presents a great challenge when building a predictive model.

Machine learning algorithms are applied successfully to a range of problems in classification, including those that model the evolution of the air traffic network, further explored in Chapter 2. This thesis intends to provide a novel approach to modeling airlines' behavior on route addition and deletion by making use of Gaussian Processes (GPs), a probabilistic approach to learning in kernel machines, and by using performance metrics appropriate for models that run imbalanced data. The GPs are implemented through the modeling technique known as Sparse Gaussian Classification (SGC), used for large datasets, based on the method of Gaussian Process Classification (GPC). GPC is a generalization of the Gaussian probability distribution. It is a non-linear, non-parametric generic supervised learning method. As such, its biggest advantage is that it employs user-specified kernels to provide predictive distributions rather than point predictions. The SGC model will be compared to a DNN model and both will be evaluated using the metrics associated with a confusion matrix in chapter 4.

### 1.1.1 The Evolution of the ATS

In the USA, we saw the most drastic changes in the ATS after the deregulation of the airline markets in the 1970s. Mergers, consolidation and the growth of low-cost carriers have played an

important role in the development of the US airline industry, which contributed to the evolution of the ATS as we know it today.

The first mergers happened shortly after the deregulation in 1978. Nowadays, over 85% of US domestic capacity is controlled by four different entities (Belobaba, Odoni, & Barnhart, 2015). This has allowed merged airlines to apportion their network, eliminating redundant operations, and limiting capacity. This strategy is the reason there has been little growth in capacity in the US markets and also the reason for the recent profit performance of US airlines. However, this is an unstable point for the current ATS and as such it has inconvenienced passengers and stakeholders alike (Belobaba, Odoni, & Barnhart, 2015). The many challenges to overcome in the current ATS will be discussed in this chapter.

Air travel demand is the primary driver of air transportation, along with shipment of goods (Belobaba, Odoni, & Barnhart, 2015). Socioeconomic factors like population size, age, education level and disposable income greatly affect air travel demand. Prices of competing modes of transportation influence air travel demand, especially in short-haul markets that utilize cars, buses or trains as alternative modes of transportation. Cities with industry in similar sectors will require more air travel between them.

As a result of these demand factors, hubs have grown in large metropolitan areas, where demographics and economic activities prevail. The evolution of the system has also shown that these fewer hubs offered even more flights over time and became more important to the resilience of the ATS. The adaptation of the ATS to the society's needs has also benefited the population directly by providing access to remote areas, more continuous flights between large cities and supplying goods by air shipping.

Because of the importance of the US airline industry, many decisions and strategies require the agreement between a number of varied organizations; the legislative entities, environmental and labor organizations, and local governments that aim to protect competition and the consumer. Requiring so many institutions to come to an agreement in many cases results in a troublesome and inefficient way to achieve the improvement of infrastructure and air traffic management systems (Belobaba, Odoni, & Barnhart, 2015).

### 1.1.2 Challenges to overcome for a successful growth of the ATS

Current times present an exception to the historic increase in demand for air travel. The COVID-19 pandemic is the culprit for the unprecedented measures nations around the world have adopted as an attempt to contain the spread of the virus. These measures include the closure of airports, limited flights, and entry banned to international travelers.

However, putting these exceptionally reduced air travel times aside, the historic trend translated to increasing air traffic, and the current ATS was facing many challenges to accommodate this increase. The system suffers from bottlenecks and inefficiencies, ultimately resulting in passengers experiencing delays and air traffic controllers and pilots working in stressful conditions. Some of these difficulties are still felt in the current limited air travel situation, now due to regulations that enforce social distancing and by rerouting flights to certain airports only, sometimes eliminating the possibility to purchase direct flights.

Amongst the most important challenges currently faced by the ATS is the outdated airport infrastructure, which limits airport capacity. Airport capacity constitutes one of the most important long-term constraints on the growth of air traffic (Belobaba, Odoni, & Barnhart, 2015). The runway system plays an important role in this claim. Delays cannot be easily mitigated and they propagate rapidly through the network with weather implications that constrain the number of runways in use and airports operating at a full capacity or near full capacity. Unfortunately, the time-consuming processes like passport control or luggage transport and the underfunded projects make it very difficult to modernize the air traffic management systems and upgrade and expand infrastructures (Belobaba, Odoni, & Barnhart, 2015).

With the increase of air traffic, safety measures will require improvement to achieve faster and more efficient screening of air travelers. These process adaptations need to be exercised with extreme care in order to prevent compromising everyone's safety.

Environmental concerns have also affected the growth and current operation of the ATS. We are becoming more aware of the effects the airline industry has on the environment through greenhouse gas emissions, as well as noise pollution in nearby towns and the environmental disruption through the expansion of airports. New environmental policies could drive the investment and adoption of new technologies, which could also negatively affect carriers that cannot fund the required upgrades.

Modeling the evolution of the ATS can help the population predict the future changes and this way allocate funds efficiently and strategically to support the system's adaptations, which contribute to the population's economic growth and fulfillment of needs.

### 1.1.3 Modeling the Evolution of the ATS

The evolution of the ATS is characterized by the effects of external changes such as government's regulations and policies, environmental matters and society's preoccupations. One of the biggest changes that resulted from this process was the shift from a point-to-point network to a hub-and-spoke network, where fewer airports offer more flights while the majority of airports only offer a few.

Network theories can help us determine topological characteristics of the hub-and-spoke air route network. Combining these characteristics with historical data can help us decipher the evolutionary nature of this complex system. Understanding the underlying framework is key to building efficient models that can more efficiently support and adapt to the changes in economics, transport, infrastructure and demographics.

The motivation of this thesis is to better understand the effect passenger demand has on the decision-making process of addition and deletion of routes. To achieve this, first it is necessary to understand how to process data that is highly imbalanced, because this is a problematic characteristic of the dataset used to train and test any model. In the past, it was reported that results on deletion of routes were more accurate than on addition of routes (Sha, Moolchandani, Panchal, & DeLaurentis, 2016). For this reason, this research focuses on how the deletion of routes is affected by historical data on passenger demand, by using machine learning algorithms a proof of concept.

The time scale of the variability of flight networks is much shorter than that of air route networks. Flight networks can vary from week to week and with the seasons. On the other hand, airways are more stable because of regulations and the physical constrain of available airspace. However, with the implementation in the near future of the concept of free flight traffic control, a more dynamic airway system will result from pilots choosing what airway to fly in certain areas (Rocha, 2017). The implementation of such a system requires an optimized airway network. According to (Rocha, 2017), network modeling efforts in this direction will allow the

implementation of a decentralized system that could be the answer to optimized flight times, avoiding congestion, bottlenecks and bad weather.

As our society becomes aware of the mechanisms that have driven our past decisions, we use machine learning to computerize the process of decision-making so we can predict data more efficiently. In the case of supervised learning, this computerization consists on the analysis of past data to find the patterns that have led to certain outcomes. Historically, these tools have been applied to concrete problems like character or image recognition. More recent applications face the difficulties that come datasets with few data points in the class of greater interest. These are known as imbalanced datasets.

## 1.2 Research Question

- What are the challenges, and means to assess, associated with predictions when applying Machine Learning to transportation data that is imbalanced and using only one independent variable?

Per results obtained in classification experiments where multiple machine learning algorithms are compared, the hypothesis is that assessing the performance of machine learning models that have been trained with imbalanced data needs to include metrics derived from a confusion matrix in order to be complete. Measuring the accuracy of this model by comparing against the test data will not reflect the performance of the model due to the imbalanced nature of the test data.

Gaussian Processes can be used as a tool in the prediction of future changes in the air service provider network topology by modeling of the airlines' decision on route addition and deletion using one independent variable. The performance of the model will rely upon an optimized set of parameters, and how we use this model will depend on specific metrics used to assess its performance.

## 1.3 Structure of thesis

This thesis is structured in the following way: first, a literature review on other methods used to perform classification in this field or an applicable field is presented. These methods are divided into optimization-based and network-based. Since this study focuses on the application of GPC (a network-based algorithm), a more in-depth review on network-based algorithms follows. This

category comprises a collection of machine learning algorithms that have been used to model route selection and route formation. Using previous studies and analyses, a comparison on the performance of these methods is summarized. Next, the importance of machine learning in human applications is highlighted, and the GPs are introduced along with the Classification methods that use GPs: GPC, and its large-data-application counterpart, Sparse Gaussian Classification. The advantages of the use of GPs in contrast with other approaches are summarized. In Chapter 4, the specifics of the data used in this project are explained. These include the imbalanced nature of the training and test data. The metrics used to measure the performance of the models are selected specifically taking into account this imbalance. They are explained and used in this chapter to assess the performance of the DNN model and the SGC model. In Chapter 5, the results are validated using a technique called K-fold cross-validation. Lastly, conclusions are drawn and future steps are presented.

# CHAPTER 2. METHODOLOGY

## 2.1 Literature review

Models built to date that attempt to predict trajectories and route selection can be categorized into optimization-based and network-theory-based (Sha, Moolchandani, Panchal, & DeLaurentis, 2016). Optimization methods are characterized by the use of an objective function along with constraints and input variables. The goal is to optimize this objective function and then, multiple methods can be used to refine the results. The biggest downside to this approach is that there is only one defined objective, which is normally cost minimization. In contrast, in network-based approaches, the multi-objective decision-making process of the airlines can be replicated when maximizing the capabilities of the system (Sha, Moolchandani, Panchal, & DeLaurentis, 2016).

Optimization-based approaches include:

- Linear Programming relaxation method (Lohatepanont & Barnhart, 2004)
- Integer Linear Programming (Jaillet, Song, & Yu, 1996)
- Mixed-integer programming (Balakrishnan & Chien, 1990)
- Mixed-integer programming with relaxation methods (Raack, 2012)
- Profit maximization (Lederer & Nambimadom, 1998)

Network-theory-based approaches include:

- Discrete Choice Modeling (Sha, Moolchandani, Panchal, & DeLaurentis, 2016)
- Logistic Regression (Guitton, 2000)
- Random Forests (Diaz-Uriarte, 2004)
- Artificial Neural Networks (Tsoukalas & Uhrig, 1997)
- Fitness Function (Kotegawa, 2012)
- K-nearest neighbor (Murphy, 2012)
- Support Vector Machines (Cortes & Vapnik, 1995)

A brief description of each will be presented next.

In optimization, linear programming (LP) used to be the solution method of choice in the aircraft routing problem when the number of alternatives was relatively small (Balakrishnan &

Chien, 1990). This trend then transitioned to mixed-integer programming methods (MIP). However, integer programming has a high computational cost, and as the number of nodes increases, so does the complexity, making it unusable for large networks computations (Sha, Moolchandani, Panchal, & DeLaurentis, 2016).

In the LP approach, Etschmeier and Richardson, cited in (Balakrishnan & Chien, 1990), make use of LP along with integral routing variables and continuous single-commodity variables, with a very loose LP relaxation. Jaillet et all (Jaillet, Song, & Yu, 1996) presented an ATN design approach without the initial assumption of hub-and-spoke distribution. The flow-based model was designed for capacitated networks and routing policies. Three integer linear programming models are presented, based on the number of stops and connecting flights. The inputs include the set of all cities, the distance between them, the demand measured in passengers who decide to fly from one city to another, and the supply, presented in the form of number of aircraft, their capacity, and cost per mile.

In the MIP category, Balakrishnan and Chien (Balakrishnan & Chien, 1990) use a MIP method along with a Lagrangian relaxation scheme, which allows them to obtain tight upper bounds. The objective in their method is to maximize total profit. The constraints ensure commodity flow as well as a limit on the amount of traffic and capacity, which corresponds to demand. Raack (Raack, 2012) applied mixed integer programming in capacitated network design. Here, a linear objective function is optimized over linear constraints for demand and capacity with integral variables. The MIP method integrates a linear programming (LP) relaxation to increase the solution space and to solve the resulting relaxed problem. Then the problem is re-optimized by adding a cutting plane algorithm, which divides the solution space into smaller subproblems.

Lederer and Nambimadom (Lederer & Nambimadom, 1998) study the choices of networks by performing a profit maximization analysis. In their study they use several assumptions regarding the demand, the frequency of flights, takeoff, landing and ground times, as well as a relaxed assumption on delays. They compare four types of networks: direct, hub-and-spoke, tour, and subtour.

Li and Wang (Li & Wang) use the column-generation method to integrate the fleet assignment and aircraft routing problems. This approach minimizes the cost with respect to constraints that include a maximum utilization of the aircraft to maximize revenue, and time constraints including connection time and flight duration, as well as the maintenance times mandated by the FAA.

Network-based methods allow for multi-objective approaches, which resemble the airline's decision-making process. These methods can be categorized into parametric and non-parametric. Parametric models define an initial type of function, which shapes the model, and the parameters are adjusted to better fit the data. The downside to parametric models is that they depend heavily on the initial assumptions. If the function is not adequate, the data will fit poorly. Models in this category include logistic regression and neural networks. On the other hand, non-parametric models consider a family of functions rather than a single function, so better fitting can be achieved since the modeling process does not depend on the initial assumptions by the researcher. K-nearest neighbor classifier, fitness functions, support vector machines and Gaussian processes are all non-parametric methods.

1. Logistic regression is a generalized linear model used for binomial regression. It creates a probability curve for the occurrence of an event, or dependent variable, based on a set of independent variables, also called predictors (Lemeshow & Hosmer, 2014). It is an analog of linear regression in the classification case because the outcome in logistic regression is usually binary. The most important quantity in regression models is the conditional mean $E(Y|x)$ ($Y$ being the outcome variable and $x$ the independent variable). In the network topology evolution problem, the event represents the addition or removal of a link in the network. In (Kotegawa, DeLaurentis, & Sengstacken, 2010) the iteratively reweighted least squares algorithm (IRLS) is used to train a regression model with historical data. Link addition and removal was modeled with this method.
2. In the Fitness Function (FF) model, nodes with higher importance (or fitness value) receive a higher probability of creating new links with other nodes. The FF uses parameters such as clustering coefficient. This method is used to model the link addition event in the evolution of the network topology (Kotegawa, 2012). In (Kotegawa, DeLaurentis, & Sengstacken, 2010), Kotegawa et all develop an FF algorithm to study new route formation in the ATS. They compare the performance of this algorithm against an artificial neural network method and logistic regression.
3. The artificial neural network (ANN) was developed motivated by the intention to replicate the brain and build a learning machine. In ANNs, connections between elements are created and initialized with random weights. Then, the algorithm is fed with training data until

convergence, which takes place when the desired input-output mapping is achieved. In the binary classification case, an ANN is a logistic regression model applied to a logistic regression model. It produces posterior probabilities through nonconvex optimization in the training process (Murphy, 2012). Naessens (Naessens, 2018) used DNNs in trajectory prediction. In DNNs, multiple layers of elements are stacked to allow for more complex learning. They used a set of historical data to train the model along with a set of predictors. They performed a comparative study between decision trees, random forests, kernel SVM's, K-nearest neighbors and neural networks.

4. The Support Vector Machine (SVM) uses a decision surface to separate the input data according to the margin, which is the minimum distance between the decision boundary and the samples. Hence, the decision surface is such where the margin is maximized (Murphy, 2012). Since the model parameters are defined by convex optimization, any local minimum is also a global optimum (Bishop, 2006). SVM is a decision machine, which means that it does not provide posterior probabilities. SVM is used for link addition. In (Sun & Park, 2017), drivers' route choice behavior is modeled using SVM and ANN for performance comparison.
5. Random forests (RF) learn decision trees based on a randomly chosen subset of the input variables and data cases. These models usually perform really well in terms of predictive accuracy (Murphy, 2012).
6. The K-nearest neighbor (KNN) model is a very simple example of non-parametric models. This model counts the number of points from each class in a set of K amount of points from the training data near the input x, and returns an empirical fraction as the estimate. This method employs memory-based learning (Murphy, 2012).
7. With Discrete Choice Modeling (DCM), the main goal is to maximize utility. Airlines' decisions of route selection can be modeled using discrete choice analysis based on random-utility theory. This utility is composed by the observed utility V and the unobserved utility E. V is affected by explanatory variables that affect the user's decision. E introduces the uncertainty that comes with the randomness due to measurement error, unknown attributes, etc. The intention is to use the decision model to construct a network topology generator. In (Sha, Moolchandani, Panchal, & DeLaurentis, 2016), the strategic

planning process of deciding on origin-destination routes is modeled as a discrete choice problem where the airlines are modeled as a single benevolent entity.

The probability of a decision maker to choose an alternative i over j is defined as the cumulative distribution of $E_j - E_i$. You can find the value of this parameter with the density function of E. In (Sha, Moolchandani, Panchal, & DeLaurentis, 2016), a multinomial logit model is chosen to describe the density function. This is based on the assumption that $E_i$ is independent and identically distributed following a Gumbel distribution. Two reasons that support the use of this model are that the output choice probability can be obtained in closed form, and also that logit models are known to perform well in decision-making problems (Sha, Moolchandani, Panchal, & DeLaurentis, 2016).

## 2.2 Comparison of network-based models

Since this research focuses on the use of a network-based machine-learning algorithm, a comparison of the performance of the different network-based models mentioned above is discussed next.

- DNNs are smaller models in size compared to RFs or SVMs (Kotegawa, 2012; Lederer & Nambimadom, 1998). This is due to the high computational power required for the training process. Consequently, RFs do not scale well with an increase of training data or predictors (Naessens, 2018). However, since it has more degrees of freedom than LR, it is able to output more precise results than the latter (Kotegawa, DeLaurentis, & Sengstacken, 2010).
- In (Murphy, 2012), Murphy includes a discussion on two experimental comparisons of different binary classification methods. The two experiments differ mainly in the amount of features in the problems. The first experiment, considered low-dimensional, used 11 data sets, which were run through 10 binary classification methods. Amongst these 10 methods, 5 of them have been discussed in this project. From these, the best performing algorithm was RF, followed by SVM, ANN, KNN, and lastly, logistic regression. In the second experiment, considered high-dimensional, the best performing algorithm was based on Bayesian neural networks.

- While KNN is a simple enough method that works very well with enough labeled training data and a good distance metric, it is affected by the curse of dimensionality, which translates to poor performance with high dimensional inputs (Murphy, 2012).
- In (Sun & Park, 2017), a drivers' route choice behavior was modeled using SVM and compared with NN. The prediction accuracy between the two methods was similar but SVM was much faster in the computation time. Another study mentioned in (Sun & Park, 2017) modeled travelers' route choice behavior using a dataset from the San Francisco Bay area. They compared the performance of SVM with a multinomial logit (ML) model and NN. Their best performing algorithm was SVM, with comparable prediction capabilities to NN, but better computing efficiency.
- Another analysis on route addition and deletion compared DCM with linear regression (LR) (Sha, Moolchandani, Panchal, & DeLaurentis, 2016). In the DCM, a ML model was used. LR yielded high accuracy results, very close to DCM, but its performance worsened with small networks. The DCM model was still able to perform well with high sensitivity values.
- According to the discussion comparing some ML methods in (Kotegawa, 2012), one of the advantages of RF versus SVM is that it can be used on large datasets even when there were more predictor variables than number of observations. However, this method tends to overfit the data, especially when there is excess noise in the training dataset. This is a behavior also observed in NN, which makes it underperform against SVM (Sun & Park, 2017).
- Integer programming has a high computational cost, and as the number of nodes increases, so does the complexity, making it unusable for large network computations (Sha, Moolchandani, Panchal, & DeLaurentis, 2016).
- Kotegawa et al. compared in (Kotegawa, DeLaurentis, & Sengstacken, 2010) the performance of ANN, FF and LR. ANN performed with the highest precision, but FF and LR provided a good representation of the distribution of new routes and also allowed flexible forecasting. These two algorithms struggled with high-degree nodes, which suggested that additional parameters would be necessary to tune the models.

The following table summarizes the results discussed herein.

Table 2.1 Comparison of network-based models

| *Model* | *Pros* | *Cons* |
|---|---|---|
| *Linear Regression* | • Ability to yield high accuracy results (Sha, Moolchandani, Panchal, & DeLaurentis, 2016)<br>• Good representation of the distribution of new routes and also allows flexible forecasting (Kotegawa, DeLaurentis, & Sengstacken, 2010) | • Struggles with high-degree nodes<br>• Performance worsened with small networks (Sha, Moolchandani, Panchal, & DeLaurentis, 2016)<br>• Suffers from bias towards classes with higher number of instances |
| *Fitness Function* | • Good representation of the distribution of new routes and also allows flexible forecasting (Kotegawa, DeLaurentis, & Sengstacken, 2010) | • Struggles with high-degree nodes (Kotegawa, DeLaurentis, & Sengstacken, 2010) |
| *Artificial Neural Network* | • High precision, works well with high dimensional problems (Murphy, 2012) | • High computational power required (Kotegawa, 2012) |
| *Support Vector Machine* | • Good computing efficiency (Sun & Park, 2017) | • Does not perform well in large datasets (Kotegawa, 2012) |
| *Random Forest* | • Can handle large datasets (Kotegawa, 2012) | • Tends to overfit the data, especially with noisy training datasets (Kotegawa, 2012)<br>• Suffers from bias towards classes with higher number of instances |
| *K-Nearest Neighbor* | • Simple, works well with a good distance metric and enough labeled training data (Murphy, 2012) | • Does not work well with high dimensional inputs (Murphy, 2012) |
| *Discrete Choice Model* | • Performs well with large datasets (Sha, Moolchandani, Panchal, & DeLaurentis, 2016) | • Low accuracy (Sha, Moolchandani, Panchal, & DeLaurentis, 2016) |

# CHAPTER 3. MACHINE LEARNING ALGORITHMS

## 3.1 Machine Learning in Human Applications

Machine learning algorithms are successfully being used in today's applications like regression, spam filtering, image and voice recognition, commute time prediction, data mining, and many more. Some of these applications involve a human decision-making process, like targeted advertisement or virtual assistants. In some of these cases, the outcome is dependent on the irrational nature of the human brain and our cognitive bias, which makes the human decision-making process very complex. We have found that machine learning can constitute a set of powerful tools in the process of modeling this process, usually characterized by a recognition of patterns (Ripley, 1996). As we become aware of the mechanisms that have driven our past decisions, we use machine learning to computerize the process of decision-making so we can predict data more efficiently. In the case of supervised learning, this computerization consists on the analysis of past data to find the patterns that have led to certain outcomes. Historically, these tools have been applied to concrete problems like character or image recognition. However, achieving a machine that could be used to replicate the human's behavior in decision-making processes is of great interest in the scientific community.

Figure 3.1 summarizes the process of supervised learning in a predictive model that uses machine learning. The raw data is processed to provide two sets of data; the training dataset and the test dataset. The training dataset is then fed into the machine learning (ML) model, which also goes into an iterative process of parameter optimization, until the user is satisfied through bias and variance measurement. The resulting model is then ran with the test data, and performance metrics are used to evaluate the accuracy of the model. Once these metrics are obtained, they can be compared to other models' metrics, and the user chooses the most appropriate model. New data is fed to the model of choice, which performs the predictions.

No matter how accurate the learning process of a supervised machine learning algorithm is, it is entirely based on the training dataset that it is fed with. Feeding it the right data is a priority if the intent is to achieve accurate predictions. This is especially important when dealing with datasets that are missing information, contain too many features or are imbalanced. Before feeding

raw data to a machine learning algorithm to train, pre-processing the data ensures the training process is maximized to our needs.

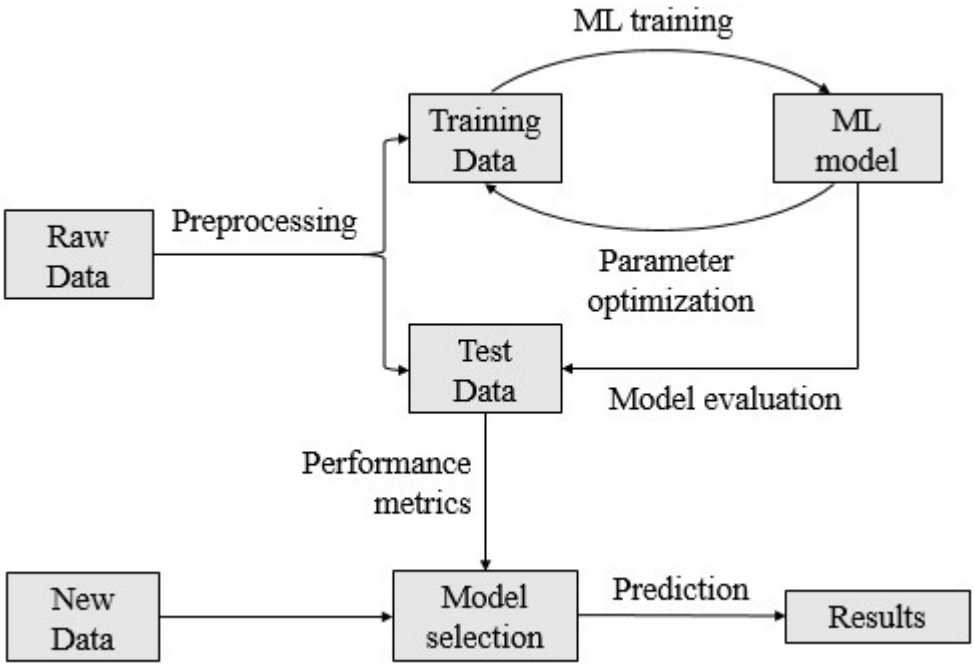



Figure 3.1 Supervised learning diagram

To exemplify this, in (Ripley, 1996), two examples are compared: facial recognition and zip code recognition. The complexity of the first one is what determines the difficulty in achieving high accuracy in the computerized classification. The less we understand the variables that help us make decisions, the less accurately the machine learning algorithm will classify. Since we are not able to access the data that airlines' use to make their decisions on addition or deletion of routes, our predictions are entirely based on public data that is readily available, and it is entirely up to the scientific community to find the connection between the outcome of the airline's decisions and use those as premises for our investigations.

GPs have been known for over 100 years but it is only 10 years ago that they were introduced in the machine learning community. A lack of a readily available introduction to this method along with its probabilistic nature have played a role in its reduced popularity. However, this method has been proven to perform competitively against more popular methods like SVM and DNN in the classification field. GPs have been used successfully in this field, with examples summarized in

Chapter 3. The study of addition or deletion of routes can be simplified as a classification problem, and this is the reason GPs seemed an appropriate approach in the prediction of route addition or deletion.

One of the biggest advantages of this method is the ability to avoid the assumption of a determined function for the classification model. Instead, the process of classification with GPs looks at a family of functions, which contributes to the flexibility of the model. A detailed description of GPs is included in the next section.

### 3.2 Gaussian Processes

A GP is a generalization of the Gaussian probability distribution. It is a collection of random variables, any finite number of which has joint Gaussian distributions (Rasmussen & Williams, 2006). GPs can be used to perform generic supervised learning in the form of regression and probabilistic classification problems. Random fields are probabilities on function spaces, and GPs are a special type of random fields. GPs are defined by a random function, a mean function and a covariance function instead of a random vector, a mean vector and a covariance matrix, in the multivariate normal context.

For a GP function f with a mean *m* and a covariance *K*, the vector of outputs *f* for the vector of inputs *x* will follow the multivariate-normal:

$$f|x_{1:n}, m(\cdot), k(\cdot,\cdot) \sim \mathcal{N}(f|m(x_{1:n}), K(x_{1:n}, x_{1:n})) \quad (1)$$

Where the mean vector is defined as $m(x_{1:n}) = \begin{pmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{pmatrix}$ and the covariance matrix as $K(x_{1:n}, x_{1:n}) = \begin{pmatrix} k(x_1, x_1) & \dots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) \end{pmatrix}$.

The mean and the covariance functions play a central role in our job defining the GP:

- The mean is defined as the expected value of the random variable $f(x)$:

$$m(x) = \mathbb{E}[f(x)] \quad (2)$$

This value varies according to the trends in the response of $f(x)$ for different inputs. The user can define this value depending on the problem: zero, some constant, a linear function,

a set of basis functions, etc. In the presented problem, like in most applications, we will not have any prior knowledge about the mean, so by symmetry this value will be chosen to be zero.

- The covariance function represents the variance of the random variable *f*($x$):

$$k(x,x) = \mathbb{V}[f(x)] = \mathbb{E}\left[\left(f(x) - m(x)\right)^2\right] \tag{3}$$

Consequently, for a pair of different input-output vectors $x$ and $x'$, the covariance is defined as:

$$k(x,x') = \mathbb{C}[f(x), f(x')] = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))] \tag{4}$$

This function defines the similarity between data points. It is based on the assumption that input data points that are closer together will have a similar target value y (Rasmussen & Williams, 2006).

The covariance matrix needs to always be positive definite (i.e., all the eigenvalues of *K* need to be positive). This ensures that the outputs are a multivariate normal distribution. The value of $k(x,x')$ will get smaller as the distance between $x$ and $x'$ grows. The covariance matrix also allows the user to model regularity, invariance, periodicity, etc., on the functions sampled from the probability induced by the GP.

The most commonly used covariance function is the Squared Exponential (SE). This is modeled in the Python package GPy as the Radial Basis Function or RBF, and it is the kernel of choice used in this project due to its popularity and flexibility. The SE formula is the following:

$$k(x,x') = v_0 \exp\left\{-\frac{1}{2}\sum_{i=1}^{d}\frac{(x_i - x_{i'})^2}{l_i^2}\right\} + v_1 \tag{5}$$

where $v_0$ is a parameter that represents the signal strength. This means, the larger this value, the more the GP will vary about the mean. If this value is set too large, the function will follow outliers. On the other hand, the parameter $l$ represents the length scale of the $i$-th input dimension of the

GP. The more we increase this value, the smoother the samples of the random variable along the $i$-th dimension. This means, the slower the values change. Parameter $v_1$ specifies the variance of a zero-mean offset with a Gaussian distribution. Prior to running the algorithm, these parameters need to be defined. The vector containing these parameters is $\theta = (logv_0, logv_1, logw_1, \dots, logw_d)$, where $w_i = \frac{1}{l_i^2}$. These parameters are defined as logarithms since they are positive scale-parameters. The vector of parameters $\theta$ is analogous to the hyperparameters in neural networks.

Choosing the right parameters in your covariance kernel is key to maximize the performance of the GP. In order to estimate the value of $\theta$, one can pursue the maximization of the likelihood through an optimization method like the Broyden-Fletcher-Goldfarb-Shanno algorithm (BFGS), or, instead, use a Bayesian approach via Markov Chain Monte Carlo where a posterior distribution over the parameters is obtained (Williams & Barber, 1998). In this model, the parameters were optimized using the BFGS approach. The user can also select a number of restarts to try for the multi-start approach and the function then chooses the best solution found.

### 3.3 Parametric vs. Non-parametric

The problem we are presented with consists on a finite set of data that needs to be mapped to a function so we can make a point prediction from an input. In order to find this function, we make assumptions about it. Depending on the type of assumptions made, there are two categories of modeling techniques that arise: parametric and non-parametric. In the parametric case, we look at a specific class of functions and we define the parameters that describe the family of distributions. This approach is limited in the sense that if the wrong type of function is assumed, the fitting will be poor. On the other hand, there is the non-parametric modeling, which is able to look at a family of functions for its probability distribution instead of just a set of parameters within a fixed function class.

The GP framework is very powerful because many models used today in machine learning are restricted examples of GP. Since the GP is not a parametric model, there is no need to fit the data in the model. GP techniques look at a family of functions rather than a family of parameters, which makes it very flexible against parametric methods.

### 3.4 Generative vs. Discriminative approach discussion

Generative models constitute a branch of unsupervised learning techniques in machine learning. These models make use of the joint distribution while discriminative models use the conditional distribution. Classification problems can be approached in two different ways, depending on how the joint probability *p(y,x)* (where y is the class label) is decomposed. The first approach consists on modeling the class densities (generative approach) and the second on modeling the conditional probabilities (discriminative approach). Using Bayes theorem, the generative approach decomposes the joint probability as *p(y)p(x|y)* and the discriminative approach as *p(x)p(y|x)*. The generative approach is useful when we are handling missing data or outliers because we have access to *p(x)*. However, density estimation for the class-conditional distribution is a hard problem especially if *x* is high dimensional. In the discriminative approach, we model *p(y|x)* directly, so modeling assumptions do not affect the process as much as in the generative approach (Ripley, 1996).

The posterior probability can be calculated using the following formula:

$$p(y|x) = \frac{p(y)p(x|y)}{\sum_{c=1}^{C} p(C_c)p(x|C_c)} \tag{6}$$

For the generative approach, inference of *p(y)* is straightforward: it consists on the estimation of a binomial probability in the binary case.

The following is a summary table with the multiple approaches described in Chapter 2.

Table 3.1 Model Type Comparison

| *Model name* | *Parametric/Non-parametric* | *Generative/Discriminative* |
|---|---|---|
| *Linear regression* | Parametric | Discriminative |
| *Logistic regression* | Parametric | Discriminative |
| *Fitness Function* | Non-parametric | Discriminative |
| *Neural Network* | Parametric | Discriminative |
| *Support Vector Machine* | Non-parametric | Discriminative |
| *Random Forest* | Non-parametric | Discriminative |
| *K-nearest Neighbor* | Non-parametric | Generative |
| *DCM* | Parametric | Discriminative |
| *Gaussian Process* | Non-parametric | Discriminative |

### 3.5 Gaussian Process Classification

As its name indicates, this method will consist on a learning process of input-output mapping from empirical data (the training dataset). Depending on the characteristics of the output, this approach is known as either regression, for continuous outputs, or classification, when the outputs are discrete. Our problem is a binary classification problem.

In probabilistic classification, test predictions take the form of class probabilities. This means, it results in predictive distributions rather than point predictions, which is the output in regression analysis (Rasmussen & Williams, 2006).

The targets are discrete class labels instead of continuous probabilities. For this reason, Gaussian likelihood does not apply (in GP Regression, we define priors and likelihoods to be Gaussian, which give us GP posteriors). In GPC, the likelihood is not Gaussian, but the posterior process can be approximated by a GP (Rasmussen & Williams, 2006).

The main difference between GP Regression (GPR) and GPC is the way the output data is connected to the underlying function outputs. In GPR this connection is done through a noise process. However, in GPC, there are two steps in the classification of the data because we are interested in the class probabilities. The first step involves the GP in itself. This consists on evaluating the latent function, which represents how the likelihood varies over the x-axis. The second step involves squashing this output through a sigmoid (e.g., logit or probit) function. This schematic from (Ebden, 2008) represents the process just described.

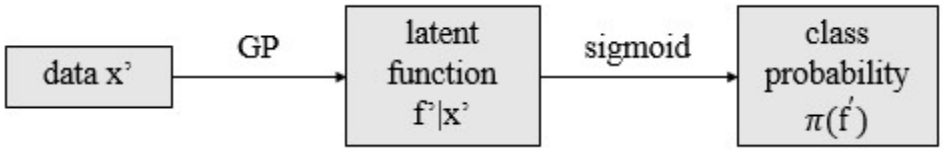



Figure 3.2 Summary schematic of GPC process (Ebden, 2008)

Given a training set $D$ of inputs $x_i$ with binary class labels $y_i \in \{-1, 1\}: D = \{(x_i, y_i)|i = 1, \dots, n\}, X = \{x_i|i = 1, \dots, n\}, Y = \{y_i|i = 1, \dots, n\}$, the goal is to find the correct class label y' for a new data point x'. This is achieved by computing the class probability *p(y'|x', D).* The idea is to transform some real valued latent variable f', which contains the value of some latent function f (·) at x'. For this, we define a GP prior on this function, which means the points from this function will follow a multivariate Gaussian density. Also, we have a set of parameters defined by $\theta$ that will contribute in the definition of our class probability.

Since we are basing this mathematical representation of GPC from Rasmussen and Williams' book (Rasmussen & Williams, 2006), we will assume we use a probit function to relate the latent function to the class probability. The probit function will be expressed as:

$$p\big(y = 1\big|f(x)\big) = \pi(x) = \Phi(f(x)) \tag{7}$$

The observations are independent given the latent function f, so the likelihood is defined as:

$$p(y|f) = \prod_{i=1}^{n} p(y_i, f_i) = \prod_{i=1}^{n} \Phi(y_i f_i) \tag{8}$$

The next step is to define a GP prior over the latent function, with mean 0 and covariance K:

$$f|X, \theta \sim \mathcal{N}(0, K) \tag{9}$$

Which is a normalized Gaussian distribution. The posterior then becomes:

$$p(f|D,\theta) = \frac{p(f|X,\theta)p(y|f)}{p(D|\theta)} = \frac{\mathcal{N}(f|0,K)}{p(D|\theta)} \prod_{i=1}^{n} \Phi\,(y_i f_i) \tag{10}$$

Which, since it involves the multiplication of a GP and a probit function, is non-Gaussian.
The last step is to find the latent value f' at the test point x':

$$p(f'|x',D,\theta) = \int p(f'|f,X,x',\theta)p(f|D,\theta)df \tag{11}$$

And finally the predictive class probability is found:

$$p(y'|x',D,\theta) = \int p(y'|f')p(f'|D,\theta,x')df' \tag{12}$$

Both of these integrals are intractable to compute, and here is where the approximation methods come to rescue.

### 3.5.1 Approximation methods

The posterior $p(f|D,\theta)$ is intractable because it involves the product of a Gaussian (the prior) and a product of sigmoids (the likelihood). The reason we perform approximation methods is to replace a non-Gaussian posterior with a Gaussian. There are different methods: the Laplace approximation, Expectation Propagation (EP) and Markov Chain Monte Carlo (MCMC), amongst others (Rasmussen & Williams, 2006). EP is the most accurate method as long as a slightly longer running time is not a constraint (Nickish & Rasmussen, 2008). Several authors have reported it works relatively well with GP models even though convergence is not guaranteed (Rasmussen & Williams, 2006).

A comparison between multiple machine learning methods is presented in (Rasmussen & Williams, 2006). It consists on a binary classification task between images of the digits 3 and 5. In this analysis, two different approaches are compared in the GPC algorithm: Expectation Propagation (EP) and the Laplace approximation. While the test error rates where comparable between the two approaches, the EP approximation produced a higher maximum value of the log marginal likelihood as well as a better test information (Ripley, 1996).

Another example carried out by (Rasmussen & Williams, 2006), this time one-dimensional, demonstrates the superiority of the EP method approximation versus the Laplace method approximation. In this example, the same hyperparameters were chosen to maximize the approximate marginal likelihood for the Laplace method. A plot showing the actual dataset along with the predictive probability approximations by the two methods were drawn and the Laplace approximation performs conservatively compared to the EP approximation. This is because the Laplace method approximation is uncontrolled due to the Hessian, which gives a poor estimation of the true shape of the posterior.

The Python Gaussian Process framework used to build the model, GPy, integrates the EP method. It will be described next in detail.

#### 3.5.1.1 Expectation Propagation (EP)

The approximation of the predictive class probability starts with the use of Bayes' rule with the likelihood, the prior, and a normalization term to obtain the posterior:

$$p(f|X,y) = \frac{1}{Z} p(f|X) \prod_{i=1}^{n} \mathrm{p}\,(y_i|f_i) \tag{13}$$

Where the normalizing term is the marginal likelihood:

$$Z = p(y|X) = \int p(f|X) \prod_{i=1}^{n} \mathrm{p}\,(y_i|f_i) df \tag{14}$$

Up until this point, the same logic is followed for the case of regression in GP. The two methods diverge in the definition of the likelihood. In classification, the likelihood is non-Gaussian, which makes the posterior p(f|X,y) intractable. As mentioned before, we will be using a probit function to define our likelihood, i.e., $p(y_i|f_i) = \Phi(f_i y_i)$. The approach now in EP is to use a local likelihood approximation, defined as:

$$p(y_i|f_i) \simeq t_i\left(f_i \middle| Z'_i, \mu'_i, \sigma'^2_i\right) \triangleq Z'_i \mathcal{N}(f_i|\mu'_i, \sigma'^2_i) \tag{15}$$

Which is nothing but an un-normalized Gaussian function over the latent variable $f_i$. The "site parameters" are $Z'_i, \mu'_i$ and $\sigma'^2_i$ s. t.:

$$\prod_{i=1}^{n} t_i\left(f_i \middle| Z'_i, \mu'_i, \sigma'^2_i\right) = \mathcal{N}(\boldsymbol{\mu}', \Sigma') \prod_i Z'_i \tag{16}$$

Where $\boldsymbol{\mu}'$ is the vector of $\mu'_i$ and $\Sigma'$ is the diagonal with $\Sigma'_{ii} = \sigma'^2_i$.

The idea being pursued is to approximate the non-Gaussian likelihood that normalizes over the targets $y_i$ by an un-normalized Gaussian distribution over the latent variables $f_i$.
The posterior $p(f|X, y)$ is then approximated by $q(f|X, y)$:

$$q(f|X, \triangleq \frac{1}{Z_{EP}} p(f|X) \prod_{i=1}^{n} t_i\left(f_i \middle| Z'_i, \mu'_i, \sigma'^2_i\right) = \mathcal{N}(\boldsymbol{\mu}, \Sigma) \tag{17}$$

Where the rule of the product of Gaussians was used to obtain an un-normalized Gaussian, the approximate posteriors are $\boldsymbol{\mu} = \Sigma\Sigma'^{-1}\boldsymbol{\mu}'$ and $\Sigma = (K^{-1} + \Sigma'^{-1})^{-1}$, and $Z_{EP} = q(y|X)$ is the approximation from the EP algorithm to the normalization term Z from eq. (17).

The next step is to choose the site parameters. The approach consists on updating the value of the local likelihood approximation $t_i$ sequentially, for which the cavity distribution is found. Then, the new un-normalized Gaussian marginal which best approximates the product of this cavity distribution times the likelihood is found. In order to minimize the Kullback-Leibler divergence of *p(x)||q(x)*, the moments of the product between the cavity distribution and the local approximation of $t_i$ are derived, which allows us to obtain the parameters of the approximation $t_i$.

### 3.5.2 GPC vs. Other Methods

Logistic regression can be generalized to give GPC. However, GPC is different from logistic regression in that its decision boundary is non-linear. This allows the method to model non-linear functions (Rasmussen & Williams, 2006).
In (Rasmussen & Williams, 2006), an experiment on handwritten digit and character recognition is performed with different machine learning methods. The task consists on discriminating images of the number 3 with the number 5. The results are compared using the test log predictive probability and the test error percentages. The methods compared in this analysis are GPC, a linear probit model, SVM, K-nearest neighbor and a GP least-squares classifier.

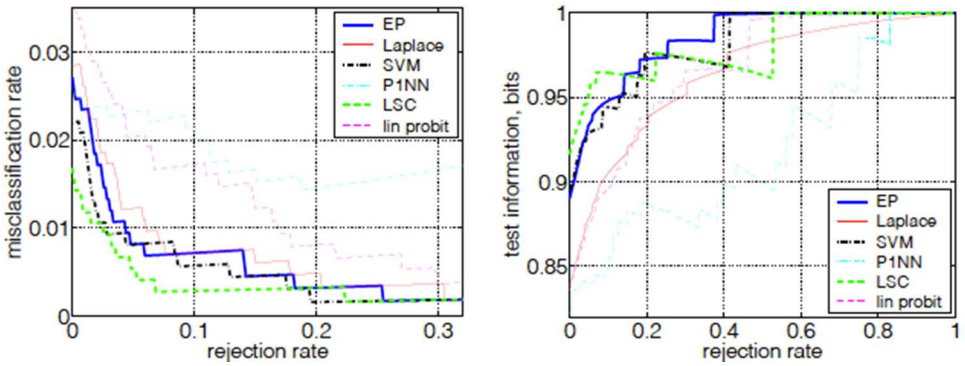



Figure 3.3 Misclassification and test information in terms of rejection rate, using different machine learning methods, in the classification task of digits 3 and 5 (Rasmussen & Williams, 2006)

Figure 3.3 summarizes the error rate and test information for comparison between the different machine learning algorithms used in the binary discrimination example. The figure on the left shows the error-reject curve and the figure on the right presents the amount of information about the test cases with respect to the rejection rate. In the legend, P1NN stands for probabilistic one nearest neighbor, which corresponds to a natural extension for K-nearest neighbor, LSC is for the GP least-squares classifier, and lin probit is for the linear probit classifier. The methods that performed the best were the Expectation Propagation (EP) approximation for GPC, the SVM and the GP least-squarest classifier. The GP least squares classifier performed the best overall. This method treats classification as a regression problem, where a Gaussian noise model is used and the goal is to minimize the squared error function (Rasmussen & Williams, 2006).

One of the advantages of GPC versus SVM is that with the first method you can interpret the result with probabilities; i.e., the output is a prediction for the probability of your value $y$ to be 1 given $x$. In SVM, probabilistic predictions could be obtained by calculating the sigmoid of the function with fitted variables and an unbiased version of the training dataset. However, this method does not take into account the variance of the function of $x$, which results in error-reject curves that correspond to an underperforming classifier (Rasmussen & Williams, 2006).

One of the best aspects of GPs versus other methods is its versatility. Even though common kernels are provided, the user is allowed to input their own kernel, as seen in Chapter 5, which allows for finer tuning. Not only that, the parameters used in GP can be chosen directly from the training data. Other models, like SVM, need cross validation because of the initial choice of kernel.

The biggest disadvantage of GPs is in the loss of efficiency in high-dimensional spaces when the number of features exceeds a few dozens, called the curse of dimensionality. This is a common problem with other advanced methods, such as DNN and SVM. Also, it is not a sparse method, which means they are harder to train with sparse data. This is where Sparse Gaussian Classification comes into the picture (this sparse GP-based algorithm is described more in depth in the next section). Despite these downsides, GPC has been proven to be a very effective classifier. A large-scale comparison study of 12 major classifiers, on 22 benchmark classification problems was performed, where GPC was the best classifier among all (Atiya, 2010).

Rasmussen & Williams speculate in (Rasmussen & Williams, 2006) about why GPs are not more widely used. There are three major reasons highlighted. The first one is that GPs require the use of large matrices, which were computationally tedious to handle in the past; the second reason is that the covariance functions in GP used to be fixed, and not much information on how to choose these functions was available, which is one of the most interesting aspects of the use of GPs. Lastly, GPs are a natural extension of Bayesian linear regression. In statistics, this was never a very popular method in the first place.

### 3.6 Sparse Gaussian Classification

The size of the training database was the main challenge encountered when processing the data with GPs. GPs are unable to process large amounts of training data rapidly, just like the other more advanced machine learning methods discussed in this thesis, ANN or SVM. This is because matrix inversion of a large number of data points makes the application of GPs infeasible for large databases (Tresp, 2001). In order to circumvent this issue, the training database was processed using a modified GP Classification method, called Sparse Gaussian Classification (SGC), where the sparseness relies on the selection of a smaller dataset.

SGC allows the user to process large datasets because the inference complexity is reduced from $O(n^3)$ to $O(nm^2)$, where N is the number of data points and M is a subset of the data that the user is able to select, where M << N. This subset of data is a set of latent variables *u*, called the inducing variables. The name of the model comes from the fact that *u* is the only way the vector of latent functions *f* and the function values *f'* "communicate", and therefore *u* induces the dependencies between the training set and the test case. This method is defined by the following likelihood and inducing prior:

$$p(y|f) = \mathcal{N}(f, \sigma^2_{noise} I) \quad and \quad p(u) = \mathcal{N}(0, K_{u,u}) \tag{18}$$

The inducing variables can be integrated out from the expression for the probability p(*f, f'*) from the joint GP prior p(*f, f', u*) as follows:

$$p(f, f') = \int p(f', f, u)du = \int p(f', f|u)p(u)du, \; where \; p(u) = \mathcal{N}(0, K_{u,u}) \tag{19}$$

Now, the approximation that defines almost all sparse approximations [45] is at the joint prior, where we assume that *f* and *f'* are conditionally independent given *u*, such that:

$$p(f', f) \simeq q(f', f) = \int q(f'|u)q(f|u)p(u)du \tag{20}$$

Per (Quiñonero-Candela, 2006), the selection of inducing variables is a challenging problem in itself, and different methods proposed in literature will be based on variations of the additional assumptions of the two inducing conditionals $q(f'|u), q(f|u)$ in the above equation.

Initially, the selection of the inducing variables had to be from the set of training points. However, Snelson and Ghahramani (Snelson & Ghahramani, 2005) proposed to relax this constraint and treat it as an optimization problem. Snelson and Ghahramani (Snelson & Ghahramani, 2005) maximized the marginal likelihood to learn the inducing inputs just like it is done with the hyperparameters. The marginal likelihood conditioned on the inducing inputs then became:

$$q(y|X_u) = \iint p(y|f)q(f|u)p(u|X_u)dudf = \int p(y|f)q(f|X_u)df \tag{21}$$

where $X_u$ is the vector of inducing inputs and *f* is the vector of latent function values.

This sparse approximation is the result of modifying the algorithms to present "an exact inference with an approximated prior", rather than "an approximate inference with the exact prior" (Quiñonero-Candela, 2006). This way it is possible to express the approximations in terms of prior assumptions about the function, which allows us to understand the consequences of the approximations (Quiñonero-Candela, 2006).

# CHAPTER 4. DATA PROCESSING AND ANALYSIS

## 4.1 Data Processing

As discussed at the beginning of this thesis, the data used to model the airline's decision of deletion of routes was obtained from BTS. According to the conclusions drawn from the analysis performed by Sha et al. in (Sha, Moolchandani, Panchal, & DeLaurentis, 2016), market demand as well as the hub status of an airport affect the addition of a route. Cost to the airline and distance do not affect the addition of a route significantly, and there is no significant interaction between the hub status and the cost of flight.

The data on market demand in the BTS-100 for 2009 through 2013 was used to run the models. This constitutes the single independent variable that will be used in the models. The data contains the market demand for the routes between the main 132 airports in the country in the United States Air Traffic Network, totaling 8,646 possible routes in the database. These are divided into two classes: the routes that were deleted that year (the minority class, or class with less samples, represented with a 1) and the routes that were kept that year (the majority class, or class with more samples, represented with a 0). The members in each class represent a route's market demand for a particular year, and the decision to delete or keep that route with respect to the year before. This database presented an imbalanced number of sample routes. This means that the minority class is the class of greater interest but suffers from the greater cost in the learning process. This type of problem has become more significant in classification in the field of machine learning because of its prevalence in multiple fields (Albisua, 2012), such as medical imaging, insurance claims and fraudulent transactions. In these examples, the imbalance in the datasets can reach values of 98% to 2%.

The data from 2009 to 2013 is characterized by the following proportion:

Class 0: 25,850 Class 1: 647 Proportion: 39.95 : 1

The reason imbalance is problematic in machine learning is because of the accuracy paradox. The classifier will be able to reach high levels of accuracy, meaning, it is able to accurately predict outcomes available in the data set, because it is most often predicting the outcomes observed in the larger class distribution (the majority class). The important outcome here is correct prediction of the minority class outcome. Some algorithms like decision trees and logistic regression have

bias towards classes with higher number of instances. The model can also suffer from overfitting the data in this class distribution or classify all data as the majority class and label the minority class as noise.

The solution to the class-imbalance problem is to process the data with a resampling method. Resampling methods consist on synthesizing a new dataset from the existing training data, where the number of samples for each class is balanced.

However, for this problem, instead of pursuing resampling, inspection of the data revealed that the majority of the data points of interest (those with a value of 1) were located at very low values of market demand, which can be observed in Figures 4.1 and 4.2. These figures represent the majority and minority classes in the dataset. The first plot shows the entire dataset, while the second one applies to market demand under 500 passengers. The distribution of the points is less skewed for the second case.

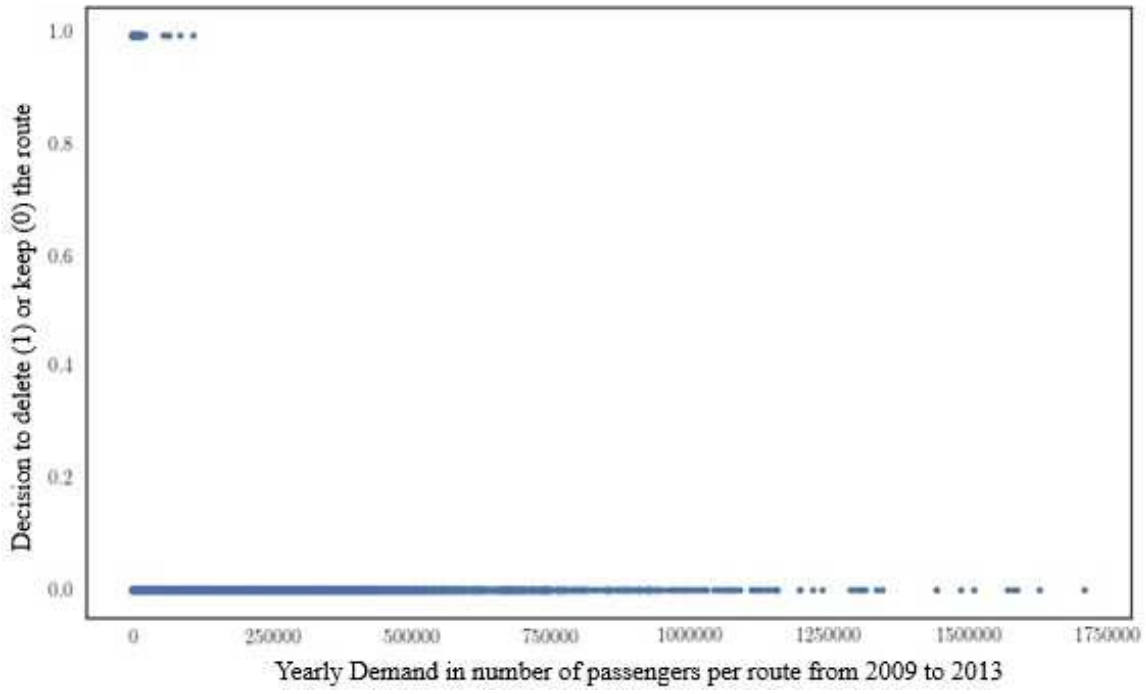



Figure 4.1 Decision to delete (1) or keep (0) a route with respect to demand data per route in the United States, from 2009 to 2013, entire dataset

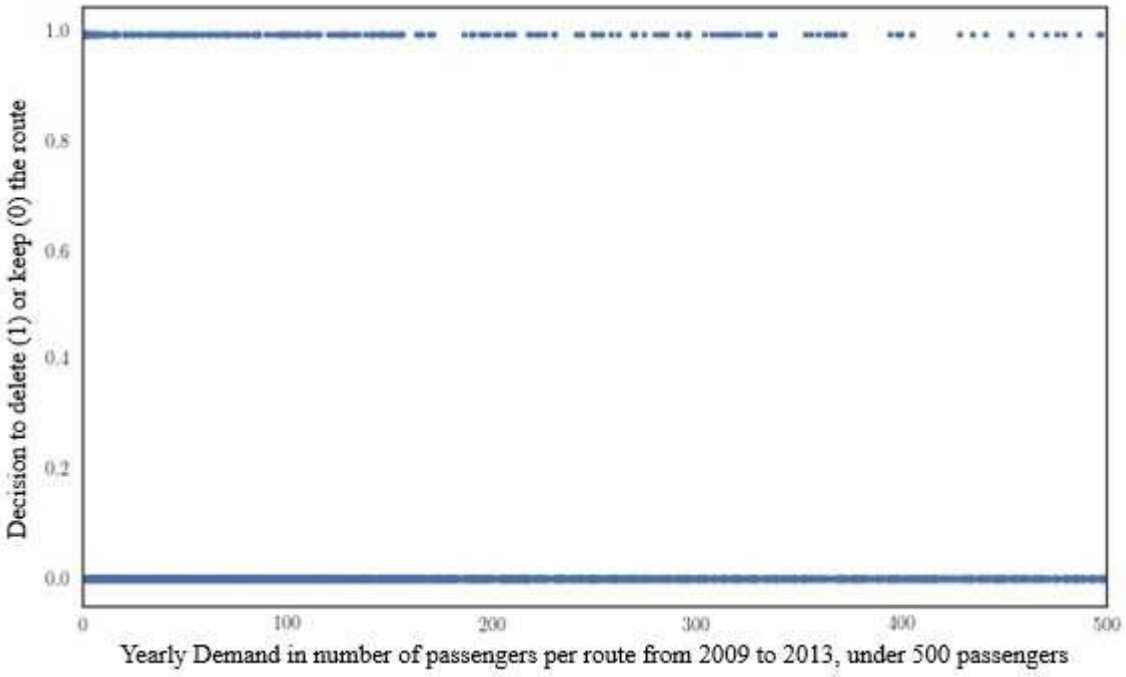



Figure 4.2 Decision to delete (1) or keep (0) a route with respect to demand data per route in the United States, from 2009 to 2013, under 500 passengers

Therefore, instead of running the entire dataset, a threshold was set to select only the points under this threshold and ignore the rest, and this way reduce the proportion of 0's to 1's.

It was observed that the threshold for market demand could be set at 500 without a significant loss of 1's and still reduce the ratio of the majority class to the minority class, so this was the threshold selected. The number of points was reduced from 25,850 to 10,327. The balance of samples of the dataset pre-threshold and post-threshold improved significantly:

Class 0: 10,327 Class 1: 401 Proportion: 25.75 : 1

Once the dataset was reduced to a more manageable size, a special type of cross-validation was used to run the models. This method is explained in Chapter 5.

## 4.2 Metrics

Figures 4.3 and 4.4 demonstrate the skewness of the data collected for this research. With such datasets, accuracy becomes an unsuitable metric to measure the performance of an algorithm (Ng, 2019). A more appropriate approach is to present a confusion matrix along with the precision

and recall metrics to analyze the results. Precision refers to the percentage of results that are relevant while recall represents the percentage of those values that are correctly classified.

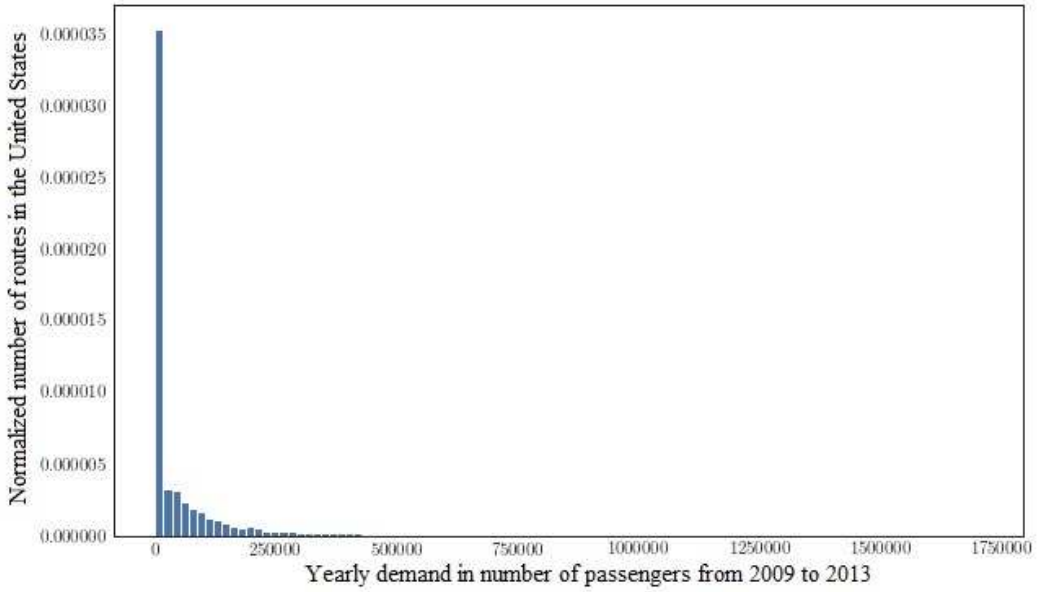



Figure 4.3 Normalized histogram of demand data, measured in number of passengers per route in the United States, from 2009 to 2013, entire dataset

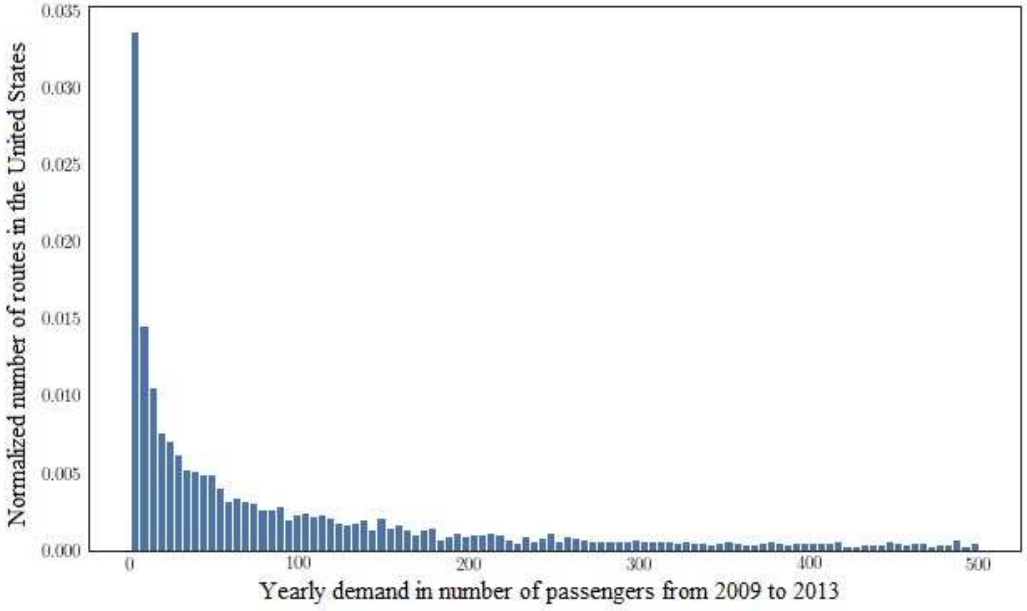



Figure 4.4 Normalized histogram of demand data, measured in number of passengers per route in the United States, from 2009 to 2013, under 500 passengers

### 4.2.1 Confusion Matrix

This table is built to help analyze the performance of a classification model. In the confusion matrix, the true negatives (TN) are the actual negatives that were predicted correctly; the true positives (TP) are the actual positives that were predicted correctly; the actual negative wrongly predicted positive is a false positive (FP) which corresponds to a type error I; and lastly, the actual positive that was wrongly predicted negative is the false negative (FN), corresponding to a type II error.

In terms of the current experiment, an FP would be a route that was actually not removed but predicted to be removed; and an FN would be a route that was actually removed but was not predicted to be removed. FPs are the least desirable type of error because we prefer a conservative decision-making process, which we will talk about more later in this section.

Table 4.1 corresponds to the confusion matrix for the values obtained when the threshold for the resulting probabilities is set to 0.495. These values are then used to compute the precision and recall.

Table 4.1 Confusion Matrix

| | *Predicted negative (0)* | *Predicted positive (1)* |
|---|---|---|
| *Actual negative (0)* | TN = 96.11% | FP = Type I error = 99.03% |
| *Actual positive (1)* | FN = Type II error = 3.89% | TP = 0.96% |

$$\text{Precision: } \frac{TP}{TP+FP} = 0.01 \tag{22}$$

$$\text{Recall: } \frac{TP}{TP+FN} = 0.20 \tag{23}$$

We are capable of controlling the trade-off between the precision and recall by applying a Bayes classifier with a threshold probability. The model will choose a class once it reaches this threshold. Varying this threshold can help achieve a classifier with higher precision and lower recall, because we want to make sure the number of routes the model predicts to be deleted are actually deleted in the data and not deleting a route that was not deleted in the data. This translates into a lower false positive.

The metric that will be used to measure the performance of the algorithm given threshold is called the $F_1$ score, which uses the prediction (P) and recall (R) values.

$$F_1\ score = \frac{2PR}{P+R} \tag{24}$$

The $F_1$ score ranges from 0 to 1, 1 indicating a perfect precision and recall, and 0 indicating one of the two values is zero.

### 4.2.2 ROC Curve and AUC

The ROC curve (Receiver Operating Characteristic curve) is used to evaluate the performance of a classification model for the range of predictive probability thresholds. The curve is achieved by plotting the precision against the recall. The AUC is the Area Under the Curve, and it represents the area encased under the ROC. It is a measure of separability, which means it is capable of telling us how good our model is at distinguishing between classes.

The AUC value ranges from 0 to 1. The closer to 1, the more capable our model is at distinguishing between classes. An AUC value of 0.5 means it is not capable of distinguishing the classes. An AUC closer to 0 means the model can distinguish the classes, but it is assigning them inversely; i.e., the 1s are 0s and the 0s are 1s.

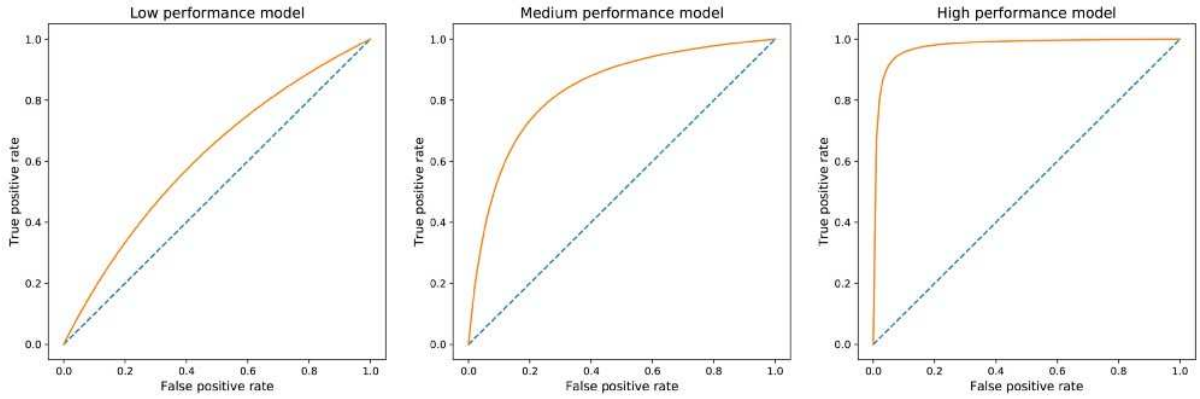



Figure 4.5 ROC curve examples (Rocca, 2019)

Figure 4.5 represents three different ROC curves. A curve that resembles a diagonal across the plot, from the bottom left to the top right, will indicate an AUC of 0.5, equivalent of a model incapable of discerning between classes. The first figure in 4.5 exemplifies this. The middle figure

corresponds to an average classifier. The last one is obtained from a good performing classifier. The better the model, the closer the line will be to the top left corner.

The scikit learn library in Python was used in this thesis in order to plot the ROC curve and find the AUC for the SGC and the DNN models.

### 4.3 SGC Results

The varying prediction threshold results are plotted in Figure 4.6. Here, the accuracy of the model increases as we raise the threshold whereas the precision, recall and $F_1$ score decrease.

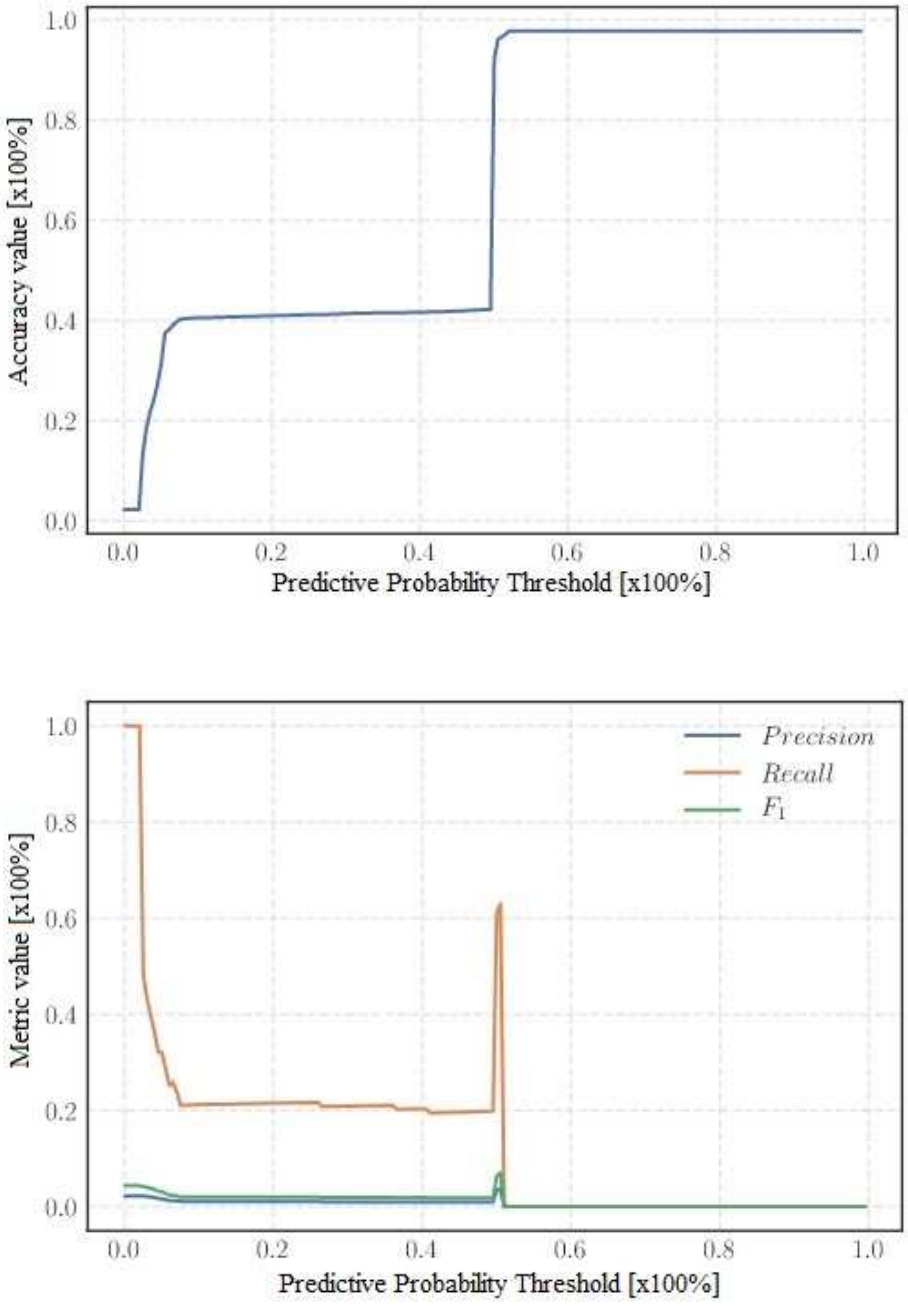



Figure 4.6 Performance metrics comparison with varying predictive probability threshold for the SGC model, using the demand data for the years 2009 to 2013 in the United States.

There is a very noticeable split in the data at the 50% predictive probability threshold. What this tells us is that the majority of the 1s returned by our model have a predictive probability below 50%, specifically between 0 and 10%. As we increase the threshold, it is more difficult to select a route to delete; hence, the model returns more 0s. Since there are more 0s in the dataset, the accuracy reflects this positively. As we increase the prediction probability threshold, the precision and recall stay relatively unaltered. At the 0.495 threshold, the model returns a 42% accuracy. At the 0.5 threshold all metrics have a spike, which is especially noticeable in the accuracy and the recall. After this value, the accuracy shoots up because there are very few real 1s afterwards. This means, if the model labeled all results as 0, the percentage of values that are 1s could just be labeled as noise, bringing the accuracy to its reported 97.77%. This is why looking at the precision and recall is important. These are indicative that the results obtained in this range are meaningless. Overall, the $F_1$ score stays relatively unaffected until the 0.49 threshold where it spikes and then decreases in a similar fashion to the precision. The spike can be interpreted as the inability of the model to select a class for the data that sits at the 50% probability threshold. The subsequent decrease is indicative of less positives actually marked positive, which means, more routes are marked as not actually deleted.

Figure 4.7 represents the ROC obtained for the SGC. The AUC is 0.375. This value confirms the inability of the model to discern between the classes.

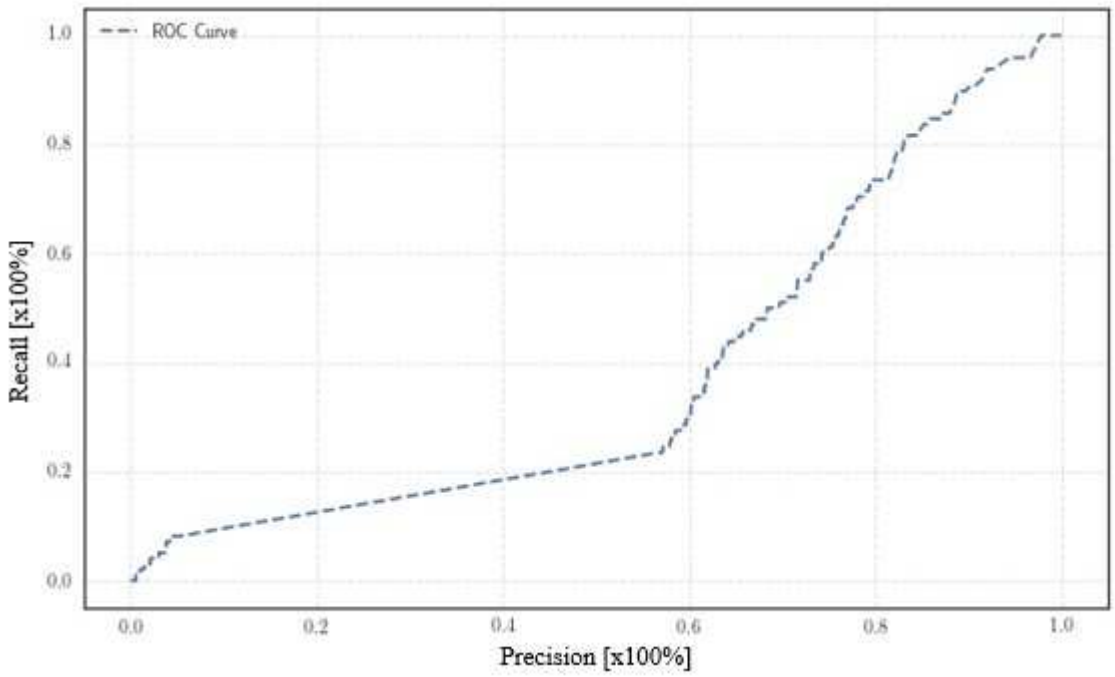



Figure 4.7 ROC curve for the SGC model, across the range of probability thresholds from 0 to 1, using the market demand test data for 2014 in the United States

It was reported in (Sha, Moolchandani, Panchal, & DeLaurentis, 2016) that regarding route addition, the accuracy in the results returned is lower compared to route deletion. This is because airlines prefer to stay conservative due to the large initial investment to operate a new route (Sha, Moolchandani, Panchal, & DeLaurentis, 2016). There are more uncertainties associated with adding routes than deleting them and airlines are more intolerant to risk in this case.

## 4.4 DNN Results

To provide some context, the SGC model was compared to a similarly built DNN model. The same historical data to train and validate the model was used as well as the same optimizer as the SGC.

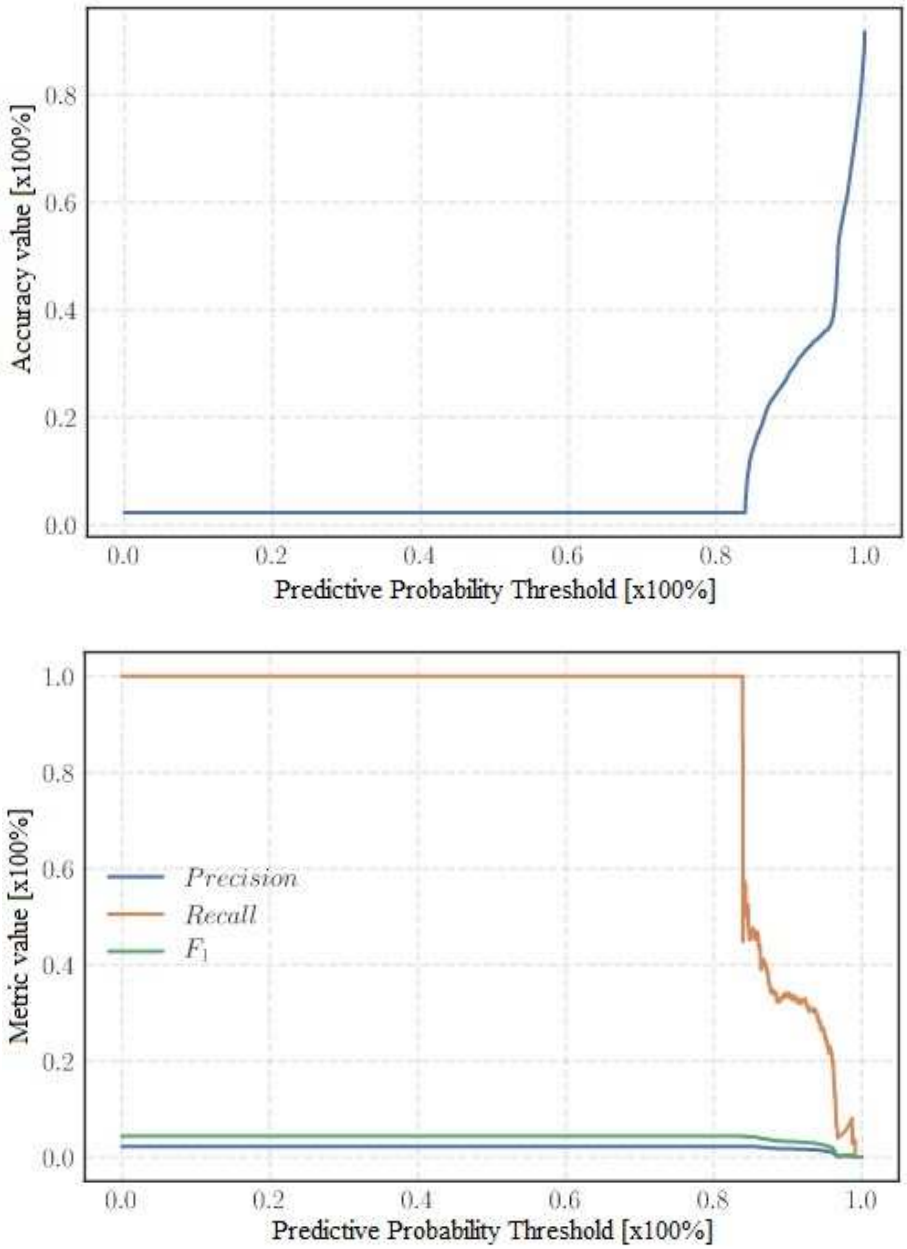



Figure 4.8 Performance metrics comparison with varying predictive probability threshold for the DNN model, using the demand data for the years 2009 to 2013 in the United States.

Figure 4.8 shows that the accuracy of the model is unaffected until the 0.85 threshold, where it starts increasing. The precision, recall and $F_1$ score are unaffected until they start dropping at the 0.85 threshold as well. Accuracy is lower in the lower threshold values because there are too many routes predicted to be deleted, which translates to more zeroes predicted erroneously as 1s. There are more 0s than 1s in the dataset, so decreasing the amount of 1s that can be returned by increasing

the threshold affects the accuracy positively once again. At the predictive probability of 0.85 we see a rapid increase of the accuracy. This is due to an increasing number of 1s around this value that will turn to 0 as the threshold increases. The majority of 1s returned come with a posterior probability of over 90% (the second part of the increase, where the slope is steeper), which is a very high chance returned by the model of the route being deleted. Therefore the recall decreases in this range and continues decreasing sharply until the end of the threshold range. The precision and $F_1$ score remain unchanged up to 0.85, and then they start to decrease as well.

Figure 4.9 represents the ROC obtained from the DNN predictions after running the market demand data for 2014. The resulting AUC is 0.314. This value confirms the inability of the model to discern between the classes.

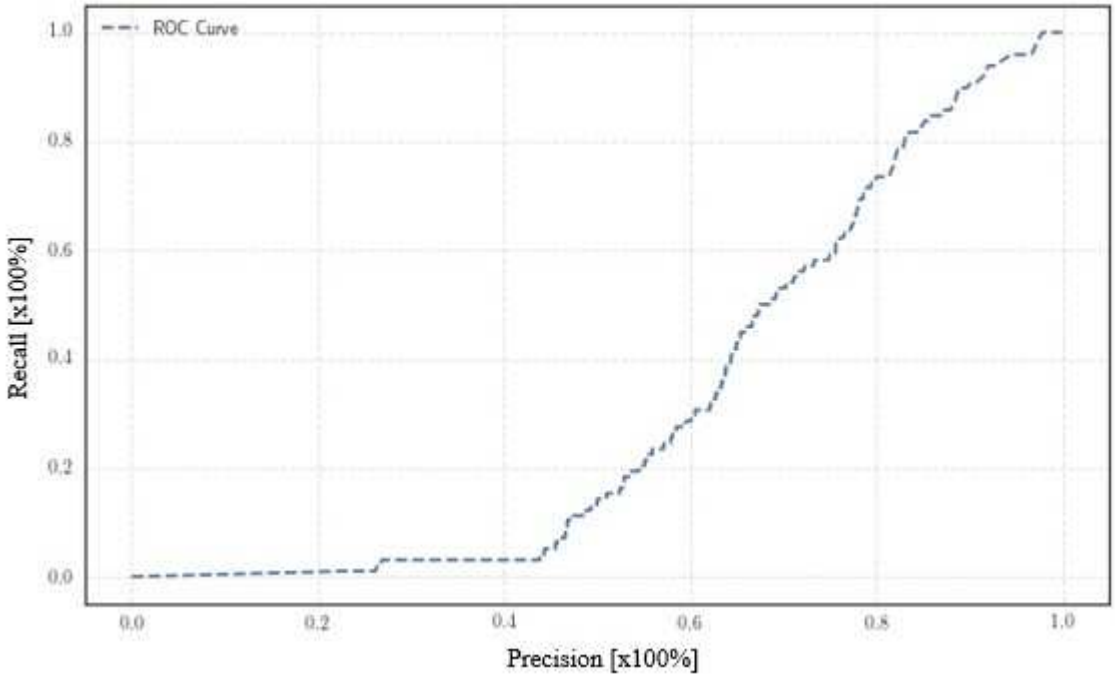



Figure 4.9 ROC curve for the DNN model, across the range of probability thresholds from 0 to 1, using the market demand test data for 2014 in the United States

## 4.5 Comparison of results

The SGC and the DNN models are compared:

Table 4.2 SGC and DNN metrics comparison

| | $SGC$ | $DNN$ |
|---|---|---|
| $Threshold$ | 49% | 96% |
| $Accuracy$ | 42% | 40% |
| $Precision$ | 0.01 | 0.01 |
| $Recall$ | 0.20 | 0.20 |
| $F_1\ score$ | 0.02 | 0.02 |
| $AUC$ | 0.38 | 0.31 |

- At a 49% threshold, the SGC model has an accuracy of around 42%, precision of 0.01, recall of 0.20, $F_1$ score of 0.02 and AUC of 0.38.
- At a 96% threshold, the DNN model has an accuracy of around 40%, precision of 0.01, recall of 0.20, $F_1$ score of 0.02 and AUC of 0.31.

Without the threshold modification, these two models would appear to perform equally, given the high accuracy output of 97.77% in both cases. However, using a predictive probability threshold to fine-tune our results and the appropriate metrics for imbalanced problems can help in the selection of the appropriate model for the problem, as well as support the decision-making process. From these results it can be concluded that these models had a very similar performance.

However, it is interesting to see how differently they behave along the range of threshold values. Most of the cases of route deletion have a predictive probability of over 90% for the DNN model while for the SGC they stayed under 50%. Unfortunately, in both models, the precision was lower than the recall, which is undesirable for this problem. As stated earlier, a higher precision translates to more correctly classified values, which in the case of route deletion it is important to account due to the high costs associated to initiating a new route. The poor performance of the models suggest that more data should be used to train the models, as well as new data about other parameters that seem to affect the deletion of routes.

The models output similar ROC curves, which yielded an AUC of 0.31 and 0.38, meaning the models are incapable of differentiating between the classes. The low probabilities returned by the SGC confirm these suspicions.

# CHAPTER 5. VALIDATION

## 5.1 Stratified K-fold cross-validation

A popular method used in machine learning to compare different algorithms is K-fold cross-validation. It consists on splitting the data set into k folds or subsets, training each model with (k-1) folds and running them against the remaining fold, until every fold has been used as the test set.

Stratified K-fold cross validation consists of a modified K-fold cross validation. The difference relies on the arrangement of the data points in each fold. The stratified method ensures that each fold contains a representative amount of data from each class. This is especially significant in highly unbalanced datasets, making it more suitable for the dataset used in this research. The k error in each fold will contribute to the performance metric for the specific model.

The user can vary the value of k depending on the outcome. Bias and variance are inversely proportional so a lower K will result in more bias, a higher K will produce larger variance. K=10 has been shown through experimentation across fields to provide a model skill estimate with acceptable bias and variance (Kohavi, 1995). Twenty folds were conducted to compare the validation models but it became too computationally intensive so the simulation was stopped. Five folds produced acceptable results. Ten folds yielded similar results, slightly less biased. Therefore, in this model, k will be set to 10.

## 5.2 Results

A DNN stratified 10-fold cross-validation was run to validate the results from both models. This operation produced 10 folds of the complete data set from 2009 to 2014. The stratification ensured a balanced amount of data points from each label for every fold.

The accuracy, $F_1$ score, precision and recall numbers were plotted for every fold against the range of values for thresholds. The values for a predictive probability threshold of 0 to 0.96 stayed constant until the observed changes in the region of interest. The four plots in figure 5.1 show the values on the region of interest, from threshold 0.96 to 0.985. This region of interest is where the majority of the predictive probabilities are returned from the model.

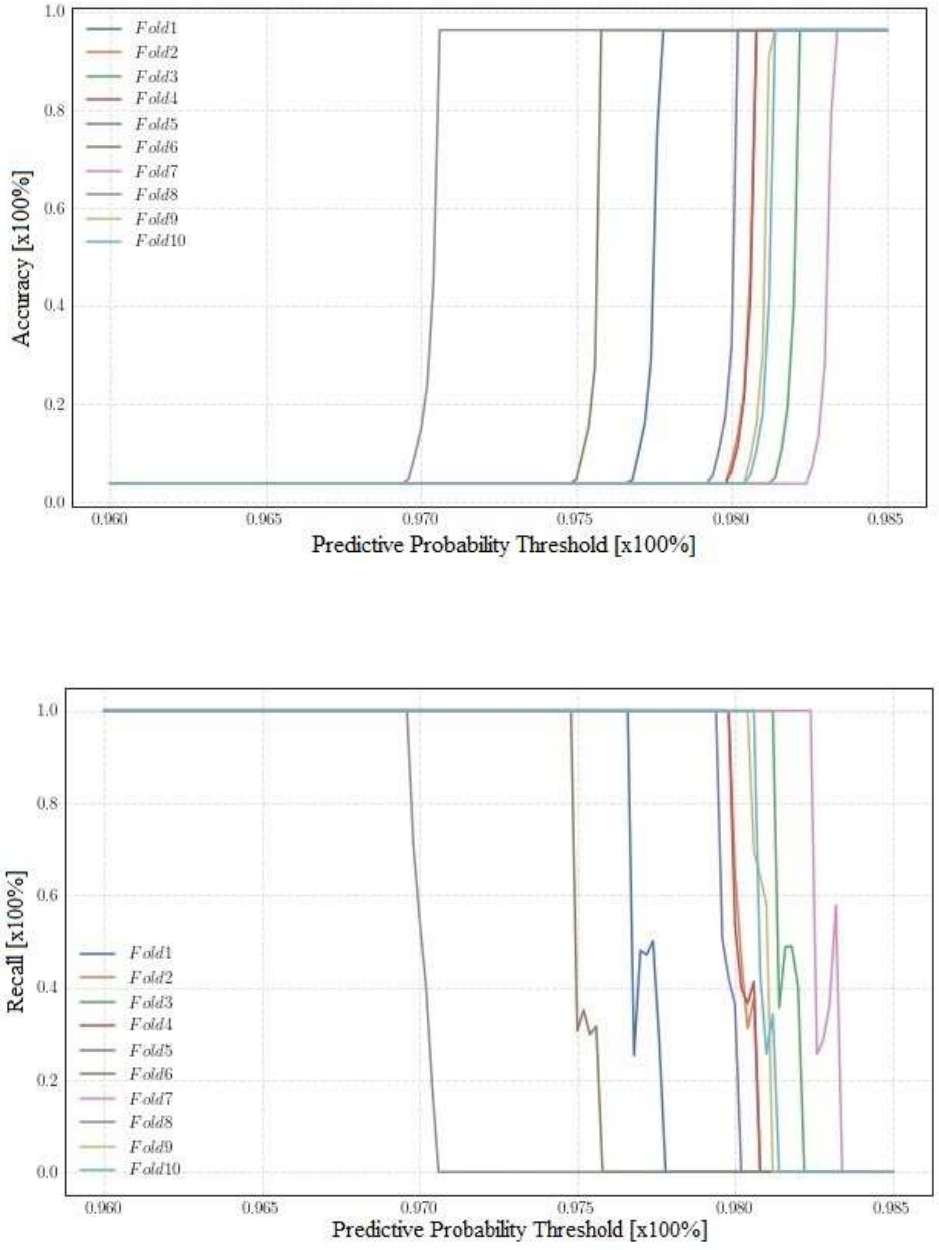
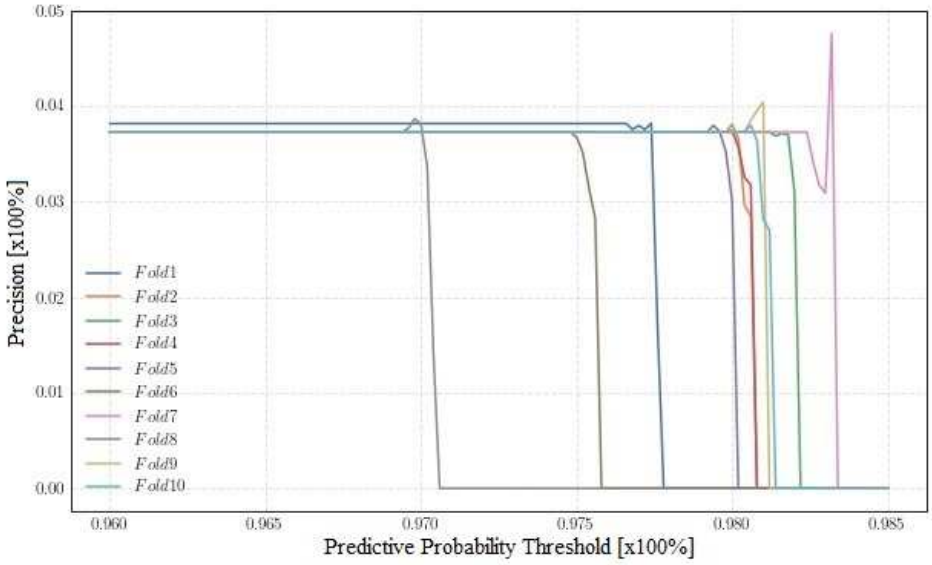
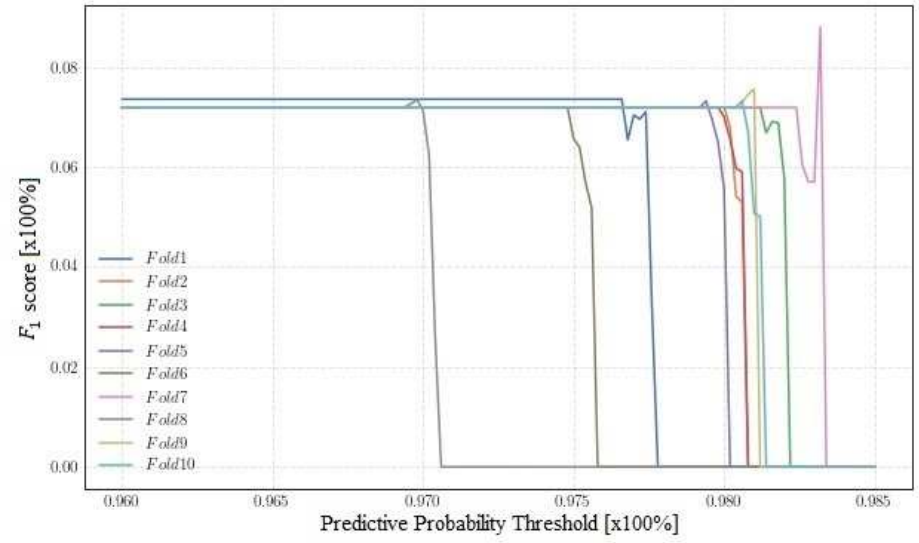



Figure 5.1 Metrics measured for the stratified 10-Fold Cross-Validation results using the DNN model, for the predictive probability threshold range of 0.96 to 0.985.

Figure 5.1 continued




Each plot shows the range of values for every fold for each different metric: accuracy, recall, precision and $F_1$ score. Overall, all folds appear to perform very similarly. The predictive probabilities range from ~0.973 to ~0.980. The recall plots show the same bump we had seen in the previous section plots. This sudden change in slope is due to an increase of misclassified values.

Observing the point right before the bump, we can note that the recall varies between 0.3 and 0.6 for the folds. At this same threshold, the precision is measured to be between 0.01 and 0.02, the $F_1$ score between 0.03 and 0.07, and the accuracy between 0.2 and 0.6. The plots shapes match those that resulted from the DNN model in the previous section, which validates the results obtained for the year tested.

An attempt was made to run a Sparse Gaussian Classification Stratified 10-fold Cross-Validation model. The model turned out to be very computationally expensive for this algorithm and for that reason fewer threshold values were run. With fewer values, the plot was not significant for comparison with the original model where we tested against data from 2014.

# CHAPTER 6. CONCLUSION AND FUTURE STEPS

The objective of this thesis is to identify the challenges, and means to assess, the application of machine learning to transportation data that is imbalanced and using only one independent variable. Machine learning is a powerful tool used to understand the relationship between different variables that lead to certain decisions that maximize profit or minimize loses. It has found broad applications in finance, healthcare, technology and engineering. However, we have learnt that there is no universal machine learning algorithm. Existing models that use a machine learning algorithm set a precedent for resolving new problems, but testing new approaches result in more efficient models or more accurate results. Our decision on what algorithm to implement is impacted by the type of data collected as well as the characteristics we are interested in studying. More recently, we have encountered across many fields of study skewed datasets that affect the performance of the algorithm, by resulting in the accuracy paradox.

The demonstration application centers creating a prediction model that relates the number of passengers per flight leg as the single independent variable and the airline's decision to keep or eliminate that route (dependent variable). One of the difficulties faced in the modeling process originated from the imbalance of the data. This is a problem currently observed in fields where there is a lack of data in the class of interest. The datasets used for training can become unmanageable due to the size, because the low percentage of cases in the class of interest forces the user to select a bigger dataset to obtain more samples. The problem then requires the use of a sparse method to build a model. The pre-processing of the imbalanced dataset can also include resampling which introduces bias and overfitting or causes the loss of important information.

This thesis represents a step in the process for finding the best performing algorithm in a classification problem for an imbalanced dataset. In this type of problems, the outputs of the model are discrete. There are existing categories and the data is classified following an input-output mapping that results from the training process. The number of problems in classification is continuously increasing, and there have been many instances of successfully applying a machine learning algorithm to create this mapping. The addition and deletion of routes by airlines represent a binary classification problem. Routes are deleted –or not—as a result of a decision-making process affected by many variables, and studying how they may affect the outcome is a powerful approach to better understand the factors that affect the network topology evolution. A simplistic

approach where this decision-making process is dependent on a single variable (in the case of our model, demand) is not representative of the outcome.

The use of GPs seems appropriate in this problem because of the reported high prediction accuracy in binary classification problems in (Rasmussen & Williams, 2006). Properly interpreting the accuracy in the results is another key part of machine learning. The application of the SGC and DNN algorithms to this particular problem, which uses an imbalanced dataset, teaches us that sometimes accuracy results can be deceiving, and further investigation is required for validation.

Using the appropriate metrics for imbalanced problems allowed us to conclude that a much more thorough study of the variables that affect the airlines' decision-making process is required in order to use a machine learning algorithm successfully. The output from both the SGC and the DNN models showed a precision lower than the recall, which means they did not perform accurately. This may seem to contradict the accuracy value returned of over 97%, but when a model is trained and tested with imbalanced data, the accuracy on its own is not sufficient to evaluate the performance of the model. The use of appropriate threshold values for the predictive probabilities plays an important part in the analysis of the results from these models. Varying the threshold is a way to avoid relying on the models' results completely and to restrict the number of cases where a route is predicted to be deleted. Since creating a route incurs a large investment, deleting it should be a decision that comes with high confidence, and the appropriate threshold allows that precisely.

The SGC model returned probabilities of under 50%, which confirms the inability of the model to perform predictions with confidence. The DNN model returned similar results to the SGC model but the probabilities reported were higher than 90%. Since the $F_1$ score was very low in both cases, the high threshold probability returned by the DNN is undesirable. The AUC was 0.38 for the SGC and 0.31 for the DNN classifier. These values come close to 0.5, which indicate their inability to discern between the classes.

A properly optimized SGC model can represent an aid tool in the decision-making process of adding or deleting an air route. The model in this thesis represents the first step towards the successful application of Gaussian Processes to the problem. This model will need to be optimized by exploring other choices of covariance matrices. The output of the model is highly impacted by this matrix, which makes this investigation crucial in the process of obtaining a good performing

algorithm. Another task that will greatly benefit the model is the optimization of the inducing inputs. This will allow us to maximize the capabilities of the sparse data.

Understanding these limitations and the metrics used lays the grounds for future application of Gaussian Processes in decision-making. The next steps in this study include research to find more parameters that could affect the airlines' decision-making process (i.e., the urban development in the origin/destination city, financial performance of the country in that year, rate of growth of other modes of transportation, etc.). This will allow the scientist to create a more complex model that can accurately represent the reality of the air traffic network evolution. A stratified K-fold cross-validation technique is an efficient way to confirm the validity of the model, as seen in this thesis.

Further research in the topic should include the exploration of variable-optimization algorithms. Recently developed Gradient Descent Optimization algorithms proved to work very well in machine learning (Ruder, 2017). When the data readily available is scarce or hard to obtain, the improvement of the model through optimization methods can result in less computational effort and improved use of computational memory.

# REFERENCES


Albisua. (2012, November 24). The quest for the optimal class distribution: an approach for enhancing the effectiveness of learning via resampling methods for imbalanced data sets.

Atiya, A. (2010). Gaussian Processes for Classification. *ANNPR*.

Balakrishnan, A., & Chien, T. W. (1990). Selecting Aircraft Routes for Long-Haul Operations: A Formulation and Solution Method. *Transportation Research Part B: Methodological, 24*(1), 57-72.

Belobaba, P., Odoni, A., & Barnhart, C. (2015). *The Global Airline Industry.* Wiley & Sons.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Chan, S. (2001). Complex Adaptive Systems. *ESD.83 Research Seminar in Engineering Systems*.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273-297.

Diaz-Uriarte, R. (2004, December). *Variable selection from random forests: application to gene expression data.* Retrieved from http://ligarto.org/rdiaz/Papers/jornadas.bioinfo.randomForest.pdf

Ebden, M. (2008, August). Gaussian Processes for Classification: A Quick Introduction.

Guitton, A. (2000). *Huber Solver versus IRLS Algorithm for Quasi L1 Inversion.* Retrieved 2019, from http://sepwww.stanford.edu/public/docs/sep103/antoine2/paper_html/index.html

Jaillet, P., Song, G., & Yu, G. (1996). Airline Network Design and Hub Location Problems. *Location Science, 4*(3), 195-212.

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint COnference on Artifical Intelligence (IJCAI).*

Kotegawa, T. (2012, January). Analyzing the Evolutionary Mechanisms of the Air Transportation System-of-Systems Using Network Theory and Machine Learning Algorithms. West Lafayette, IN: Purdue University.

Kotegawa, T., DeLaurentis, D. A., & Sengstacken, A. (2010). Development of Network Restructuring Models for Improved Air Traffic Forecasts. *18*(6), 937-949.

Lederer, P. J., & Nambimadom, R. S. (1998). Airline Network Design. *Operations Research, 46*(6), 785-804.

Lemeshow, S., & Hosmer, D. (2014). Logistic Regression. In N. Balakrishnan, *Methods and Applications of Statistics in Clinical Trials: Planning, Analysi, and Inferentials Methods.* John Wiley & Sons, Inc.

Li, & Wang. (n.d.). Integration of Fleet Assignment and Aircraft Routing.

Lohatepanont, M., & Barnhart, C. (2004). Airline Schedule Planning: Integrated Models and Algorithms for Schedule Design and Fleet Assignment. *Transportation Science, 38*(1), 19-32.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective.* MIT Press.

Naessens. (2018). Predicting Flight Routes with a Deep Neural Network in the Operational Air Traffic Flow and Capacity Management System.

Ng, A. (2019). *Machine Learning. Trading off Precision and Recall.* (Stanford, Producer) Retrieved from Coursera: https://www.coursera.org/lecture/machine-learning/trading-off-precision-and-recall-CuONQ

Nickish, H., & Rasmussen, C. E. (2008). Approximations for Binary Gaussian Process Classification. *Journal of Machine Learning Research.*

Quiñonero-Candela, R. (2006). A Unifying View of Sparse Approximate Gaussian Process Regression.

Raack, C. (2012). Capacitated Network Design – Multi-Commodity Flow Formulations, Cutting Planes, and Demand Uncertainty.

Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Ripley, B. (1996). *Pattern Recognition and Neural Networks.* Cambridge, UK: Cambridge University Press.

Rocca, B. (2019, January 27). *Handling imbalanced datasets in machine learning*. Retrieved from towards data science : https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28

Rocha, L. E. (2017). Dynamics of air transport networks: A review from a complex systems perspective. *Chinese Journal of Aeronautics*, 469-478.

Ruder, S. (2017). An overview of Gradient Descent Optimization Algorithms.

Sha, Z., Moolchandani, K., Panchal, J. H., & DeLaurentis, D. A. (2016). Modeling Airline's Decisions on City-Pair Route Selection Using Discrete Choice Models. *Journal of Air Transportation, 24*(3).

Snelson, E., & Ghahramani, Z. (2005). Sparse Gaussian Processes using Pseudo-Inputs.

Sun, B., & Park, B. B. (2017). Route Choice Modeling with Support Vector Machine. *World Conference on Transport Research.* Shanghai: Elsevier.

Tresp, V. (2001). Scaling kernel-based systems to large data sets. *Data Mining and Knowledge Discovery*(5), 197-211.

Tsoukalas, L. H., & Uhrig, R. E. (1997). *Fuzzy and Neural Approaches in Engineering*. New York, NY: John Wiley and Sons, Inc.

Williams, C., & Barber, D. (1998). Bayesian Classification With Gaussian Processes. 1342-1351.