A data mining approach to fraud detection in e-tail

A case study in an online luxury fashion retailer

Nuno Abrunhosa Carneiro

Master's Dissertation

Supervisor: Prof. Dr. Gonçalo Figueira



Integrated Master in Industrial Engineering and Management

January 2016

Dedicated to all the teachers who inspired me.

Abstract

Payment fraud has been a problem for merchants for a long time. More recently, e-commerce has transformed the way that most retailers do business. However, the possibility of reaching anyone in the world through the internet also makes it easier for fraudsters to hide their identity. Payment fraud leads to billions of dollars in losses for merchants. With the development of machine learning algorithms, researchers have been finding increasingly sophisticated ways to detect fraud without ever seeing a fraudster.

We propose a data mining approach to the problem of detecting fraud in e-tail businesses. Our approach is based on extracting the most information from the data available to the merchants, such as customer history, payment details, order data and customer behaviour. An exploratory analysis is done on the data from the online marketplace for luxury fashion Farfetch. Several feature engineering functions are used. Three different machine learning algorithms are applied for the classification problem: Random Forests, Logistic Regression and Support Vector Machines. A deployment solution for classification of incoming orders is proposed and implemented at the case study company.

This dissertation details the many steps needed to develop a fraud detection solution by data mining. We conclude that machine learning algorithms are essential in supporting human workers in classifying orders efficiently. The results show that the features which are used are very important for the performance of the classifier and that Random Forests is a very effective and versatile model to use.

Resumo

A fraude de pagamento é há muito tempo um problema para os comerciantes. O comércio electrónico transformou a forma de como retalhistas fazem negócio, trazendo a possibilidade de chegar a clientes de todo o mundo através da internet. No entanto, esses desenvolvimentos tornaram também mais fácil a ocultação da indentidade de indivíduos que cometem atividades fraudulentas. A fraude de pagamento é responsável por perdas de muitos milhares de milhões de dólares para os comerciantes. Com o desenvolvimento de algoritmos de *machine learning*, têm sido estudados métodos cada vez mais sofisticados de detetar fraude.

Propomos uma abordagem de *data mining* para o problema de deteção de fraude no contexto de retalho online. A nossa abordagem é baseada em extrair a máxima informação a partir dos dados que estão disponíveis ao comerciante, tais como o histórico de cliente, detalhes de pagamento, detalhes da encomenda e comportamento do cliente. É feita uma análise exploratória com os dados disponibilizados pela plataforma online de venda de artigos de luxo Farfetch. São criadas várias variáveis novas através de engenharia de variáveis. São aplicados três algoritmos para o problema de classificação: *Random Forests*, *Logistic Regression* e *Support Vector Machines*. É feita uma implementação da solução proposta para classificação de encomendas na empresa de estudo de caso.

Esta dissertação inclui detalhes sobre os vários passos necessários para o desenvolvimento de uma solução de *data mining* para deteção de fraude. Concluímos que os algoritmos de *machine learning* são essenciais como sistema de apoio à decisão humana para classificação eficiente de encomendas. Os resultados mostram que a decisão de quais variáveis utilizar é muito importante para o desempenho dos modelos e que *Random Forests* é o modelo mais versátil e com a maior eficácia.

Acknowledgements

I would like to thank Farfetch for giving me the opportunity to develop this project. In particular, I owe my gratitude to my supervisor Francisca Marinho who guided me from day one, to Joana Fernandes for showing me the direction to follow, to João Gomes from whom I learned how to be a data scientist, to Miguel Costa for the great technical input which improved the project and for his persistent feedback, to Inês and André for their friendship and to everyone else at Farfetch who supported me in any way.

I would also like to thank my dissertation supervisor, Prof. Gonçalo Figueira, who's help was invaluable for the completion of this document. Thank you for the guidance, for the great feedback and the constant availability. I would like to thank Prof. José Luís Moura Borges, Prof. Henriqueta Nóvoa and Prof. Isabel Horta for their support with statistics.

There were many defining moments in the path which led me from the enrolment at this Master to the conclusion of this dissertation. I would like to appreciate all those professors, friends and family who provided me the right opportunities and motivated me to seek to learn more everyday and become, in some way, a better person.

Contents

Contents i

List of figures v

List of tables vi

1 Introduction 1
- 1.1 Motivation . . . 1
  - 1.1.1 E-commerce: how online payments work . . . 2
  - 1.1.2 The fraud detection problem . . . 3
- 1.2 Objectives and methodology . . . 3

2 State of the Art 5
- 2.1 Different approaches to fraud detection in card-not-present transactions 5
- 2.2 Data mining techniques for fraud detection . . . 6
  - 2.2.1 Review of unsupervised methods for fraud detection . . . 6
  - 2.2.2 Review of supervised methods for fraud detection . . . 7
  - 2.2.3 Final perspective on data mining techniques for fraud detection 7

3 Farfetch Case Study 9
- 3.1 Fraud at Farfetch . . . 9
- 3.2 Current practice of fraud detection at Farfetch (As-Is) . . . 10
  - 3.2.1 List-based evaluation . . . 11
  - 3.2.2 Key performance indicators . . . 11
- 3.3 Resources . . . 11
- 3.4 Proposed approach for Farfetch (To-Be) . . . 12

4 Data understanding and preparation 15
- 4.1 Building the dataset for study . . . 15
  - 4.1.1 Collection of the data . . . 15
  - 4.1.2 How to label the orders as fraudulent? . . . 16
  - 4.1.3 Time to Chargeback – how to choose the data set . . . 18
  - 4.1.4 Training and Testing set . . . 19
- 4.2 Exploratory analysis on the training data . . . 20
  - 4.2.1 Time when an order is placed . . . 20
  - 4.2.2 Product gender . . . 21
  - 4.2.3 Order quantity and value . . . 22
  - 4.2.4 Payment type . . . 22

4.2.5 Browser . . . 23
4.2.6 Interaction with photos . . . 23
4.2.7 Distribution of frauds per country . . . 24

5 Modelling and evaluation 27
5.1 Data transformation . . . 27
5.1.1 Feature selection . . . 27
5.1.2 One-hot-encoding for low-dimensional categorical variables . 28
5.1.3 Risk-level grouping for high-dimensional categorical variables 28
5.1.4 Engineered variables . . . 29
5.1.5 Imputing missing values . . . 29
5.1.6 Standardization of numerical variables using the Min-Max method 30
5.2 Model Selection . . . 30
5.2.1 Performance measures . . . 30
5.2.2 Cross-validation . . . 32
5.2.3 Testing results . . . 36
5.2.4 Discussion of results . . . 38
5.3 Deployment at Farfetch . . . 42
5.3.1 Order Classification Service . . . 42
5.3.2 Results of implementation . . . 43

6 Conclusion and Future Work 45

Appendices 51

A List of features used in training the algorithms 53

B Comparing Random Forests performance with balanced and imbalanced training sets. 55

C Results of testing with additional features 57

List of Figures

1.1 Diagram of online payments steps. Source: Montague [2010] 3
1.2 CRISP-DM reference model phases. Source: Chapman et al. [2000] . 4

3.1 Current process for fraud detection at Farfetch 10
3.2 Proposed process for fraud detection at Farfetch 13

4.1 Criteria for labelling an order as fraudulent. 18
4.2 Cumulative distribution of the number of weeks before a chargeback is communicated (Farfetch data) 18
4.3 Distribution of orders by time of day when they are placed. 21
4.4 Distribution of variable product gender. 22
4.5 Distribution of variables quantity and order value. 22
4.6 Distribution of variable *PaymentType*. 23
4.7 Distribution of variable *Browser*. 24
4.8 Occurrence of fraud and total number of orders when user interacted with product photos before purchase or not. 24
4.9 Fraudulent orders in each destination country. 25

5.1 Example of a Receiver Operating Characteristic curve (ROC) 32
5.2 Example of the division of the dataset for cross-validation and testing 33
5.3 ROC and PR curves of testing results with Random Forests algorithm. 37
5.4 Histograms of score predictions for each of the labels. 38
5.5 Plot of Recall and Specificity for each threshold. 38
5.6 Plot of automation level and fraud. 39
5.7 Diagram of the Order Classification System 43
5.8 Comparison of scores distributions of orders approved vs. cancelled by Order Processing Team. 44

List of Tables

3.1 Data mining goals 13

4.1 Summary of Input Variables 17
4.2 Key statistics of non-binary numeric variables 20
4.3 Division of fraudulent orders between day and night periods 21

5.1 Example of one-hot-encoding: variable *PaymentType* is transformed into 4 dummy variables 28
5.2 Example of confusion matrix 31
5.3 Results of cross-validation 35
5.4 Testing results of the Random Forests model. 36
5.5 Number of legitimate and fraudulent orders whose score falls into each range 37
5.6 Confusion matrix of results when setting threshold at c=0.22. 39
5.7 Confusion matrix of results when automatically approving 80% records with lowest estimated suspicion scores. 40
5.8 Relative importance of the top 10 attributes. 41

A.1 Final list of features used in training. 54

C.1 Testing results with additional features. 57

Chapter 1

Introduction

1.1 Motivation

"Fraud is nothing new to the merchant. Since the beginning of time, man has always looked for the opportunity to defraud others - to gain goods or services without making payment." Montague [2010]

Despite not being new, fraud is a major problem for merchants, particularly in the online sector. According to the 14th annual report of CyberSource [CyberSource, 2013] the total revenue loss for e-commerce merchants in North America amounted to $3.5bi in 2012, which represents 0.9% of the total revenue. The report also states that the fraud rate for orders outside of North America was almost twice as high, averaging 1.6%.

Bhatla et al. [2003] define credit card fraud as the action of an individual who uses a credit card for personal reasons without the consent of the owner of the credit card and with no intention of repaying the purchase made. Contrary to what many consumers believe, merchants are responsible for paying the bill when a fraudster steals goods or services in a consumer-not-present(CNP) transaction. These include all transactions where the physical card is not given to the merchant, such as in mail and telephone orders and in e-commerce. On the other hand, card associations protect the merchants in the case of consumer-present (CP) transactions [Montague, 2010]. In case of e-commerce fraud, when an individual has made an unauthorized purchase with a credit card, the merchant who accepted the transaction will loose the product sold (if it has been shipped), may face chargeback fees and loss of reputation [Bhatla et al., 2003].

In order to avoid losses related to fraud, merchants must implement strategies for fraud prevention and fraud detection. Fraud *detection* is often mistaken for fraud *prevention*. Bolton et al. [2002] make this distinction clear, "Fraud *prevention* describes measures to stop fraud from occurring in the first place (...) In contrast, fraud *detection* involves identifying fraud as quickly as possible once it has been perpetrated."

Measures for fraud prevention are the key to solving problems of fraud. A very effective measure of fraud prevention in credit cards is the chip-and-pin technology. Despite the prevalence of this technology in Europe, it is still not mainstream in most other regions, including North-America. However, even when fraud prevention is highly effective, fraud detection is still required, since even the most advanced system

is prune to fail. Moreover, as fraud detection techniques become more sophisticated, so do fraudsters. Quah and Sriganesh [2008] claim that the key for accurate fraud detection lies in developing dynamic systems that can adapt to new fraud patterns in the e-marketplace. Fraud prevention and detection must therefore evolve hand in hand, faster than fraudsters.

Fraud detection has been explored in several industries with different approaches. When large volumes of data are involved, manual detection becomes impractical. In these cases, the so-called "data mining" techniques, based on statistical methods and artificial intelligence, are imperative. Some studied applications of statistical fraud detection methods include:

- Credit applications
- Telecommunications
- Credit card transactions

The first type of fraud consists in presenting false documentation with the purpose of obtaining credit from a financial institution. The other two applications, telecommunications and credit card transaction fraud were according to Phua et al. [2010] the two most published applications of data mining for fraud detection by 2010. Telecommunications fraud mainly occurs in two forms, subscription fraud (e.g. obtaining a subscription with no intention of paying) or surfing fraud (e.g. mobile phone cloning) [Bolton et al., 2002]. We are concerned about the latter application, credit card fraud detection. Despite the fact that most literature focuses on the bank perspective, this study aims at exploring fraud detection in online payments from the perspective of an e-commerce merchant. Credit card fraud is indeed a big part of the fraud perpetrated in online payments, as explained in the following section.

1.1.1 E-commerce: how online payments work

There are many possibilities for online merchants to accept payments. Montague [2010] mentions examples such as credit card, direct debit, cash alternative payments (such as PayPal, Alipay) or mobile payments. Cash alternative and mobile payments are becoming more popular and are a must for merchants who want to sell in specific overseas regions. However credit card transactions still dominate the market of e-commerce business [Montague, 2010]. Thus, it is fundamental to understand how credit card payments work.

Figure 1.1 describes the steps involved in an online credit card transaction. The merchant first verifies that the card number provided by the customer could be a legitimate card number (*Card Authentication*). After this step, the merchant will seek feedback from the bank who issued the credit card to be sure that there are funds available in the card account (*Card Authorization*). Finally, the merchant requests the *Settlement* of the transaction, the physical transfer of the funds to the merchant's bank. These steps of the transaction, which can be seen left on Figure 1.1, lead to the conclusion of the sale.

Further steps occur only in case a consumer requests a *chargeback* by claiming that he did not get the products or services requested or that the order was placed by a fraudster. In this case, the merchant can dispute the chargeback by providing

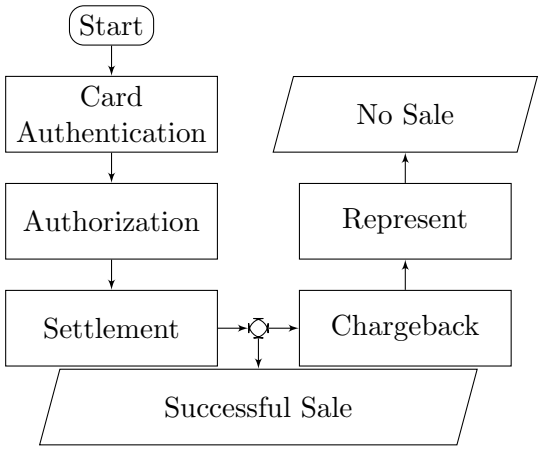


Figure 1.1: Diagram of online payments steps. Source: Montague [2010]

documentation about the order (*Representment*). If the merchant cannot reverse the chargeback, he will have to return the money to the customer's account and the sale is lost. Moreover, merchants can be subject to chargeback fees and fines from card associations if the chargeback rate is above their thresholds [Montague, 2010].

1.1.2 The fraud detection problem

Fraud detection is a data mining problem of increased difficulty, because fraudsters make their best efforts to make their behaviour appear legitimate. This undifferentiated behaviour creates a higher challenge in separating good and bad transactions. Another difficulty with fraud detection is that the number of legitimate records is far greater than the number of fraudulent cases [Bolton et al., 2002]. Such imbalanced data sets require additional precautions from the data analyst. In most cases, fraud detection applications can make use of large records of data. Bolton et al. [2002] mention the example of a credit card company carrying 350m transactions each year. However, such data is very rarely available for study.

The aim of a fraud management solution is to minimize the financial losses due to fraud and the overhead costs of preventing such losses [Bhatla et al., 2003]. An efficient solution must balance the trade-off between the cost of reviewing a transaction and the potential savings due to fraud detection [Bhatla et al., 2003].

1.2 Objectives and methodology

The goal of this study is to develop a risk scoring solution for e-tail merchants based on data mining techniques. As it was noted previously, fraud detection problems involve a trade-off between several objectives. On one hand, the solution must be able to quickly evaluate a large number of transactions. Additionally, it must meet two key metrics: maximize the number of fraudulent transactions detected (i.e. number of chargebacks avoided) and minimize the number of customer insults (i.e. number of legitimate transactions refused).

A proof of concept was built around the case study of the e-tail merchant Farfetch. This case study provided a real setting, which is certainly representa-

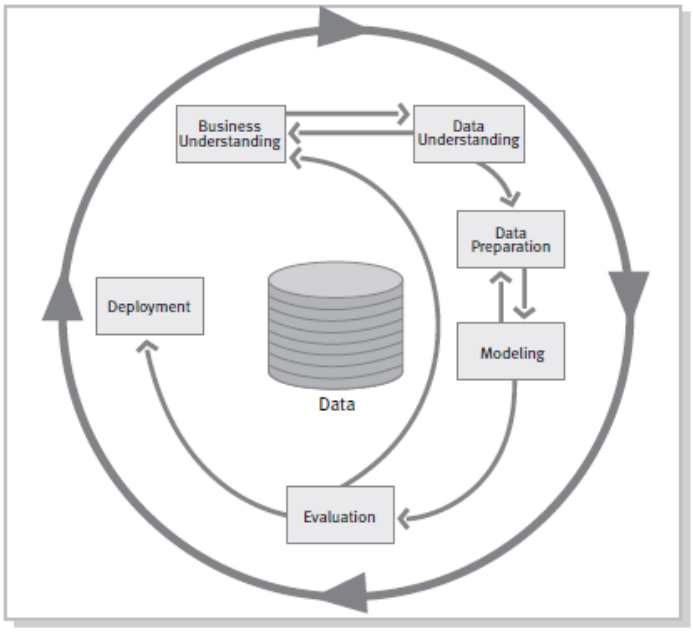


Figure 1.2: CRISP-DM reference model phases. Source: Chapman et al. [2000]

tive of many business situations. The work was developed with access to factual data and the chance to compare the results with current practice. We followed the methodology suggested in the *Cross Industry Standard Process for Data Mining* (CRISP-DM), Chapman et al. [2000]. This methodology was published in the year 2000 as a guide for data mining practitioners. It draws a process model for data mining projects organized into phases, generic tasks, specialized tasks and process instances. In this study we went through all main phases of the data mining project as described in Figure 1.2.

At first, we tried to gain understanding of the key business concepts and objectives involved. Thereafter, we tried to collect and understand the data. From there, the data was prepared and modelled with machine learning algorithms. Lastly, we have evaluated the results and deployed the solution at the company of the case study.

The remainder of the dissertation is organised in five more chapters. Chapter 2 comprises a review of the state of the art in fraud detection. Chapter 3 describes the case study of Farfetch. Chapter 4 reports the steps taken in finding the right data and the outcomes of the exploratory analysis performed on it. Chapter 5 describes the use of the machine learning methods for modelling and the results obtained, as well as the deployment process. Chapter 6 presents conclusions of the work done and gives suggestions for future work in this area.

Chapter 2

State of the Art

2.1 Different approaches to fraud detection in card-not-present transactions

Fraud detection has been the objective of research for a long time. With the increase in online payments, an increase in fraud has followed and merchants and banks must innovate to keep ahead of ever more sophisticated fraud perpetrators. Different approaches have been developed to face the threat of e-commerce fraud. Montague [2010] suggests the following techniques:

- Identity proofing
- Guaranteed payments
- Expert-rules
- Data sharing
- Technology
- Data mining

Montague [2010] defines each of the techniques in the following way. Identity proofing techniques have the aim of verifying that the customer really is who he says he is. These techniques can include e-mail, phone or address authentication through public data to get proof of the buyer's identity or calling the customer to assure that he is the owner of the phone. Guaranteed payments include merchandise insurance or authentication services provided by card associations, which guarantee that the merchant is not responsible for chargebacks. Guaranteed payments are very effective, but implemented in exchange of a fee. Expert-rules comprise all the techniques which are based on rules created from the experience of fraud analysts, including the use of positive and negative lists of users or the manual revision of orders. Such rules can be very effective, but it is hard to maintain them up to date.

Data sharing allows merchants to have more information about the customer, by acquiring such information from an external provider. This data can include credit records, social network activity or address confirmation. Data sharing can be very useful when used in combination with scoring models. Technology services can be used to enhance the available information on the user. A merchant can use

biometrics systems to confirm identity or detect the use of proxy servers to hide the IP address of the device used in the purchase. Although technological advancements help stopping fraud, fraudsters also innovate and make use of them to hide their identity.

Data mining techniques take into account past transactions to estimate the probability of a new transaction being fraudulent, by the use of advanced statistical methods. The output of data mining techniques can be a suspicion score or a direct advise to accept or reject the order. Due to the technical complexity of these techniques, a high experience is needed in order to implement them and they require an organised history of orders to learn from. We will now go into more detail on data mining techniques.

2.2 Data mining techniques for fraud detection

According to Phua et al. [2010], "it is impossible to be absolutely certain about the legitimacy of and intention behind an application or transaction. Given the reality, the best cost effective option is to tease out possible evidences of fraud from the available data using mathematical algorithms." Data mining techniques for fraud detection are divided in two main approaches:

- Supervised methods
- Unsupervised methods

Both of these approaches are based on training an algorithm with a record of observations from the past. Supervised methods require that each of those observations used for learning has a label about which class it belongs to. In the context of fraud detection, this means that for each observation we know if it belongs to the class "fraudulent" or to the class "legitimate". Supervised methods could also be used to learn multi-class problems (more than two classes), but this is usually not the case in fraud detection.

As noted before (cf. Introduction), fraud detection classification problems suffer from the issue of imbalanced class sizes, i.e. the number of fraudulent records is much smaller than that of legitimate transactions. Moreover, existing fraud detection measures usually reduce the number of frauds available for study to an even lower number. Most supervised methods perform best when trained on a balanced data set.

It is common to face data mining problems when we do not know to which class an observation belongs. For example, take the case of an online order which payment was rejected. One will never know whether this was a legitimate order or whether it had been correctly rejected. Such occurrences favour the use of unsupervised methods, which do not require data to be labelled. These methods look for extreme data occurrences or outliers. In order to get the best of two worlds, some solutions combine supervised and unsupervised techniques.

2.2.1 Review of unsupervised methods for fraud detection

Unsupervised methods are often used for exploratory analysis of data and preprocessing. When trying to reduce the dimensionality of samples, it is very useful to

use unsupervised methods such as clustering or principal component analysis to find interesting directions to explore. In fraud detection, unsupervised methods implementations focus on finding outlier transactions, which do not agree with normal patterns. However, such methods are not adequate for complete fraud detection solutions. Indeed, almost all authors studied preferred the use of supervised learning.

Moreau et al. [1999] compared the use of supervised and unsupervised neural networks. Their experiment had results where the unsupervised method performed far below the supervised neural network. Cortes et al. [2001] explored the use of Graph analysis for fraud detection in a telecommunications setting. Given its nature, telecommunications fraud is a logical candidate area to use graph analysis. Quah and Sriganesh [2008] proposed a mixed approach with the use of a self-organising map which feeds a Neural Network if a transaction does not fall into an identified normal behaviour for the given customer.

2.2.2 Review of supervised methods for fraud detection

Most of the published work on statistical methods for fraud detection falls under the category of supervised methods.

Ghosh and Reilly [1994] proposed the use of a neural network for fraud detection at a commercial bank, which proved more accurate than the previously implemented rule based expert system. Fawcett and Provost [1997] studied the use of a profiling approach to telecommunications fraud. The creation of profiles of user behaviour is favoured if there is a high usage of the service, such as in the case of telecommunications. Merchants usually do not have access to payment data other than the transactions made at this specific merchant, which makes it hard to trace a consumer spending profile. Chan and Stolfo [1998] discussed the combination of multiple classifiers in an attempt to create scalable systems which would be able to deal with large volumes of data. In general, the emphasis of research in the late 90s and early 2000s was on Artificial Neural Networks. Bolton et al. [2002] noted in 2002 that the published literature about fraud detection was scarce.

More recently, some other works have been published, making use of newer classification techniques. Srivastava et al. [2008] built a model based on a Hidden Markov Model, with focus on fraud detection for credit card issuing banks. Whitrow et al. [2009] also worked on credit-card fraud detection with data from a bank, in particular addressing the way of pre-processing the data. They studied the use of aggregation of transactions when using Random Forests, Support Vector Machines, Logistic Regression and K-Nearest Neighbour techniques. Bhattacharyya et al. [2011] compared the performance of Random Forests, Support Vector Machines and Logistic Regression for detecting fraud of credit-card transactions in an international financial institution. Random Forest proved to be the most effective and most versatile method in this case.

2.2.3 Final perspective on data mining techniques for fraud detection

Phua et al. [2010] pinpoint two criticisms to the data mining studies of fraud detection: the lack of publicly available data and the lack of published literature on the

topic. Moreover, despite the fact that there are many research papers about statistical methods for fraud detection, Phua et al. [2010] point out that only seven studies claimed to have been implemented in practice by 2010. Most literature on credit card fraud detection has focused on classification models with data from banks. Such data invariably consists of transaction registries, where it is possible to find fraud evidence such as "collision" or "high velocity" events, i.e. transactions happening at the same time in different locations. Some authors have also addressed the techniques for finding the best derived features. Whitrow et al. [2009] proved that transaction aggregation improved performance in some situations, with the aggregation period being an important parameter. However, none of these particularities seems to apply to a case of detecting fraud with data from one single merchant as in our case.

In this study, we chose to use methods of supervised learning for the classification problem, because it is common for fraud detection applications to have labelled data for training. We chose to test three different models. Logistic regression because of its popularity, and Random Forests and Support Vector Machines, which have been used in a variety of applications showing superior performance [Bhattacharyya et al., 2011]. Meyer et al. [2003] showed that Support Vector Machines perform well in classification problems. Bhattacharyya et al. [2011] claim that Random Forests are very attractive for fraud detection due to the ease of application and being computationally efficient.

Chapter 3

Farfetch Case Study

3.1 Fraud at Farfetch

Farfetch is one of the leading online luxury fashion retailers in 2015. The company follows a marketplace business model, selling items from more than 300 partner boutiques on a commission basis. The partner boutiques place their items online on the Farfetch.com portal, which allows small-medium boutiques to reach a global audience with economies of scale. Farfetch provides an omni-channel buying experience through the website, mobile or via telephone order. The value proposition for boutiques includes payment processing, branding, online content creation, out-bound logistics and customer service.

Farfetch was founded in 2008 and had substantial growth since then. Its focus is on retailing luxury items, which results in an average order value of around $650. The gross merchandise revenue in 2014 amounted to $300m and the company expects to grow by 70% in 2015, reaching $500m in sales.

As part of payment processing, fraud detection is included as an added value service provided by Farfetch. Moreover, being acknowledged by customers and card associations as a trustworthy merchant is even more important when doing business online. Farfetch wants to provide its customers high payment acceptance rates in order to convert as many sales as possible, while keeping the number of fraud occurrences to a minimum level. The company's prospects for the mid-term future foresee a sustained growth rate around 50-70%/year. This means that it must find a way to quickly evaluate a large number of orders in order to maintain the efficient service to its growing customer base.

Besides handling its own website, one of the pillars of the strategy of Farfetch is the provision of its platform as a service for brands to manage their e-commerce operations for them. This project, named "Black & White" is planned to launch in 2016. In this case, payment processing and fraud detection are again part of the value proposition of Farfetch.

Farfetch provides a good case study for a data mining solution for fraud detection, because it is representative of an online retailer and there is a large record of orders to study from. Furthermore, the company's interest in exploring this solution in-house is high, since fraud detection is aligned with the company's strategy.

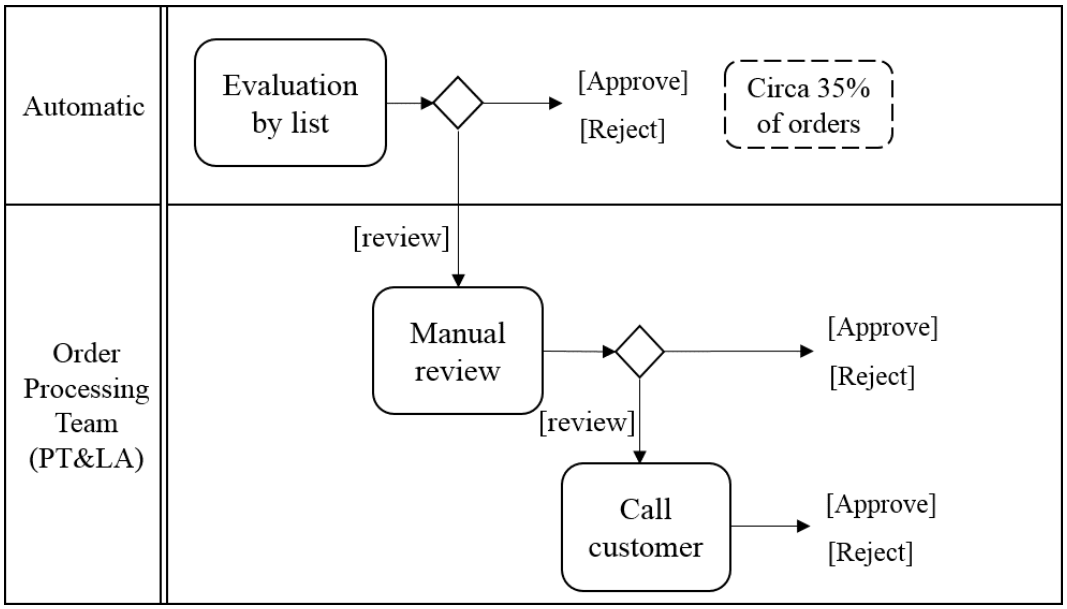


Figure 3.1: Current process for fraud detection at Farfetch

Each case has its own particular characteristics, different areas and merchants are affected by fraud differently. At Farfetch, the history of orders to study from is skewed towards more recent dates due to its high growth rate. This means that there are not so many old orders to learn fraud patterns from, as more recent orders. The evolution of Farfetch's business also presents another challenge: it became a more popular target for fraud as it became more famous.

Furthermore, Farfetch ships to anywhere in the world. Such service is of high value to attract customers, but also comes with challenges in terms of fraud prevention and detection. Shipping goods to countries where it is almost impossible to verify addresses is a big risk. The diversity of locations also adds to the complexity of detecting geographical patterns of fraud.

3.2 Current practice of fraud detection at Farfetch (As-Is)

Fraud detection at Farfetch is called order approval, as the final decision to take is whether to approve or reject the order. This is done in 2 main steps:

1. An automatic screening based on lists of trusted users
2. A manual review process for the rest of the orders

Figure 3.1 illustrates the different steps in the order approval process. At first, the user which made the order is compared against a positive list of trusted users and a negative list of fraudulent users. In case the user is found on one of the lists, the order is immediately approved or refused, respectively. The generation of the lists is based on expert-rules and described in Section 3.2.1. This first screening step processes circa 35% of all orders. Very few orders are refused at this step, as

fraudsters quickly learn that the user account they used before has been blocked. Most processed orders are approvals of orders by long-time clients.

The second step of the approval process requires direct human intervention and is handled by the Order Support Team. This team handles all global orders and is divided between the Farfetch office in Porto, Portugal and Los Angeles, USA, providing service 24/7.

The details of the orders are checked by the team for any indications of fraud. The kind of indications that they are looking for is described in Section 3.3. If this team suspects that the order is fraudulent, they will either cancel it or call the customer to get more details.

In the case that doubts remain after calling the customer (no confidence that the order is legitimate nor fraudulent), the fraud analyst can decide to subject the order to an evaluation by a third-party evaluation provider. This external provider acts as an insurance: it can approve the order collecting a share of the sale and it will cover the cost in case of a chargeback. Recurring to this provider is expensive and only a residual number of orders go through that evaluation.

The Farfetch office in Brazil has its own order approval process and is not in the scope of our research. All other orders, except those being shipped to Brazil, are included in this study.

3.2.1 List-based evaluation

The positive list which contains the trusted users, whose orders are automatically approved, is updated on a regular basis. The criteria for considering a user trustworthy is based on her/his history of orders. If its values surpass the established thresholds, this user in put on the positive list. Likewise, any user whose order was flagged as fraudulent will be put on the negative list, whose orders get automatically rejected. It is important to note that not all rejected orders are flagged as fraudulent, only those for which the Order Support Team is certain of fraudulent activity or the orders which originated a chargeback.

3.2.2 Key performance indicators

Farfetch tracks several key performance indicators (KPIs) related to fraud detection in daily, weekly and monthly reports. The main KPIs consist of: the automation level - percentage of orders automatically processed; chargeback level - percentage of orders which originate a chargeback (calculated by order value); rate of refused payments - percentage of payments which are refused (calculated by order value); speed of processing - time it takes to approve or reject an order payment; level of proof of billing - percentage of orders which require a call to the customer.

3.3 Resources

The fraud analysts at Farfetch review most orders manually and thus have an extensive experience about fraud detection. According to their practical observations, the important signs to look for when evaluating an order are:

- Accordance of shipping and billing addresses with the country of the credit card being used (e.g. a credit card from the UK being used in an order to be shipped to Australia would be noticed as a riskier transaction);
- Past fraud occurrences with the same shipping address (e.g. certain cities were considered of higher risk);
- Number of previous orders from the same customer (long time customers were almost always considered legitimate);
- Type and brand of products (specific brands were preferred targets of fraud);
- Number of cards used by customer (a high number of cards tested would be considered highly suspicious);
- Sudden increases in order values (e.g. it was thought that fraudsters would generally attempt to purchase lower value items first and only later place orders on their real targets: high value products);
- Orders placed by other users with the same name or to the same address (might be an indication of multiple accounts used by the same individual);
- Publicly available information on the internet about the customer (trying to understand whether the customer is a real person and not a fake alias).

We were given access to all the data related to transactions, orders and products in the database of Farfetch. However, this data was not organised for fraud detection and must be mined for relevant information.

3.4 Proposed approach for Farfetch (To-Be)

The objective of Farfetch is to develop a process which will result in a higher number of orders approved automatically, while not increasing the chargeback rate (i.e. the number of fraudulent orders approved).

The company wants fraud detection to continue being a part of its value proposition, even with the expected increase in the number of orders to process. The current fraud detection process is not sustainable, because it would require Farfetch to keep growing its Order Processing Team, which brings two challenges: increasing costs and increasing difficulty to coordinate the team.

Hence, the proposed solution should achieve a higher automation rate than the current process, keep the chargeback level stable and not increase the rate of refused payments. In the medium-term the proposed solution should allow Farfetch to reach a level of automation of 80% by the end of 2016, with a chargeback rate under 1% and a payment refused rate under 4.5%. The data mining goals listed in Table 3.1 state in technical terms the desired outcomes to achieve these business goals [Chapman et al., 2000].

The proposed approach consists in building a risk scoring system based on machine learning methods which will estimate a *Fraud Suspicion Score* for each order. This system would replace the current list-based order processing step. As opposed

Table 3.1: Data mining goals

Goal	Indicator
Increase number of orders automatically processed	Number of orders with a suspicion score which falls in the extremes of the range (i.e. close to 0 or close to 1).
Low chargeback rate	Number of false negatives (i.e. Fraudulent orders with a low suspicion score)
Low rate of refused payments	Number of false positives (i.e. Legitimate orders with a high suspicion score)

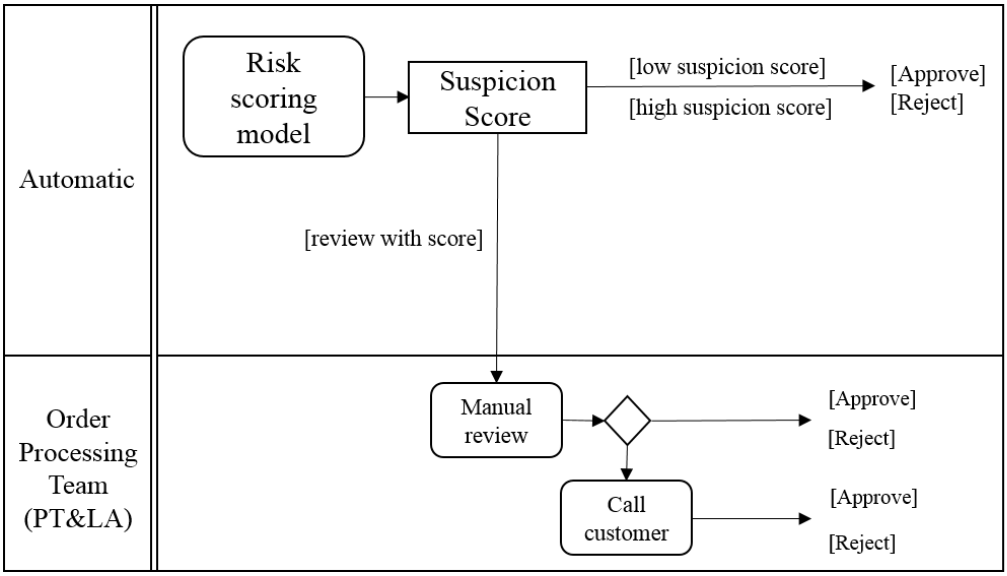


Figure 3.2: Proposed process for fraud detection at Farfetch

to comparing each order with a list of trustworthy users built by expert-rules, the risk scoring should take many more characteristics of the order into consideration (e.g. shipping destination, products being purchased, time of the order, etc.). A diagram of the proposed process can be seen on Figure 3.2.

The suspicion score estimated by the model should be a number between 0 and 1. The risk scoring model would also evaluate whether the score falls below a certain threshold (e.g. lower than 30%), where the order would be automatically approved; or higher than an upper threshold (e.g. higher than 70%), where the order would be automatically rejected. Orders with a score between the lower and the upper thresholds would be manually evaluated by the order processing team, which could also count on the suspicion score for a better evaluation. It is expected that the number of orders whose score fall in the extreme areas is higher than 35% of all orders, in order to reach a higher automation rate than with the currently used list-based evaluation.

As a first step, we tried to gain a complete business understanding of the problem. We had interviews with the involved parties at the company and shadowed their fraud analysts at work. Finally we agreed on the objectives of the study and the expected outcomes for the company.

The second step involved building the data set of orders history. After a short exploratory analysis we went to write a query to extract the data from the databases. A thorough exploratory statistical analysis followed, where we identified possible patterns of fraud by looking at individual variable distributions. The outcome was a report which could be presented to the case study company to guide them in their own analysis.

Next, the work on data modelling and data preparation evolved in parallel. The variables which could not be included in the deployed model were left out; the rest was prepared for modelling. During the modelling of the data we sequentially performed training of three distinct machine learning algorithms: Logistic Regression, Random Forests and Support Vector Machines. This included performing validation to find good model parameters and testing the solutions. The last step was the deployment at Farfetch.

Chapter 4

Data understanding and preparation

4.1 Building the dataset for study

Building the dataset on which to base the study is not a trivial activity and requires decisions which can greatly affect the quality of the data mining project. Which data to use? Where is this data to be found? Is all available data correct? The following sections describe the reasoning behind the building of the data set which was the basis of this study.

The handling of the data was done with the use of the programming language Python [Rossum, 1995], version 2.7 available through the Python Software Foundation at http://www.python.org. Two particular modules for Python, Pandas data structures [McKinney, 2010] and Scikit-Learn [Pedregosa et al., 2011], were especially useful for the modelling of the algorithms.

4.1.1 Collection of the data

Taking in consideration the input from the Order Processing Team described in Section 3.3 and the examples in the literature, we decided that the main unit to analyse would be each individual order. With that in mind, the objective at this stage was to build a table where each row would correspond to one order and each column would represent different attributes of such order.

The bulk of data was queried from a table where each line describes one product item in an order. Hence, a function of aggregation was applied so that each observation would correspond to one order. At this point, several issues arose: each order could have multiple products and for each line to represent an order, one must aggregate the characteristics of the products purchased. Thus, a column was created for each product gender ("Men", "Women" or "Other") and a count of the number of products corresponding to each gender in the order was made. The same was done for the product family ("Clothing", "Shoes", etc.). Aggregating the brand of the product could not be done in the same way, because there were more than 2.500 possible brands – it would imply adding 2.500 columns to the table. To overcome this problem, we chose to create individual columns for the 20 best-selling brands only. Other products attributes such as item price or quantity were simply summed. The aggregation steps were:

1. Read all order-product lines
2. Group by order
   (a) Sum item price, quantity
   (b) Create one column for each gender, category and sum occurrences
   (c) Aggregate product brand
      i. Find the 20 top brands
      ii. Create one column for each of the top 20 brands, count the rest under the column *BrandOther*
3. Read and append data on the transaction (payment provider feedback)
4. Read and append data on user session (e.g. number of pageviews)

Building the data set and aggregating data involves a trade-off. A higher level of aggregation results in a larger (more significant) number of occurrences per category. On the other hand, the loss of information can be substantial if the aggregation level is too high. Therefore, a balance had to be found. The final list of variables in the data set can be read in Table 4.1.

4.1.2 How to label the orders as fraudulent?

One of the challenges in classification is the definition of the class for each data entry. In our case, we faced a binary classification problem. Accordingly, each data entry – an order – should be labelled as belonging to one of two classes: fraudulent or non-fraudulent.

While it is possible to know which orders originated a chargeback transaction, this does not comprise the whole set of fraudulent orders, because fraudulent orders which were rejected did not lead to a transaction and naturally no chargeback. Hence, a broader definition of the class fraudulent had to be found. The available information on each order included:

- Approved or rejected by Order Processing Team
- If rejected, expressly marked as fraudulent by Order Processing Team
- If approved, led (or not led) to a chargeback

Therefore, two criteria are taken into consideration when defining the classes of the dataset and we created a variable which is called *LabelFraud* with a value of 0 or 1, such that:

1. Order originated a chargeback or was marked as fraudulent (see Fig 4.1) → Label as fraudulent (*LabelFraud*=1)
2. All other orders → Label as legitimate transactions (*LabelFraud*=0)

Table 4.1: Summary of Input Variables

Name	Type	Description
LabelFraud	Numeric	Class Label
OrderTimeCustomer	Numeric	Local time at which the order was placed
OrderTimeGMT	Numeric	Absolute time at GMT
OrderValue	Numeric	Value of Order
Quantity	Numeric	Quantity of items in the order
CustomerCurrency	Categorical	Currency used by customer
ShippingType	Categorical	
TimeSinceFirstOrder	Numeric	Time elapsed since user's first order at Farfetch
NumCardsUsed	Numeric	Number of cards used by customer at Farfetch
PaymentType	Categorical	
CardType	Categorical	Only for credit card
Fallback	Numeric	Only for credit card
Session3DUsed	Numeric	Only for credit card
TokenCreated	Numeric	if card data was saved by customer for card re-use
TokenUsed	Numeric	if card saved data was re-used
CardName	Categorical	name on credit card, only for credit card
CardCountry	Categorical	only for credit card
AVSCode	Categorical	Address Verification System
CV2Code	Categorical	CV2 verification system
FraudScore	Numeric	score by payment processor
FamClothing	Numeric	number products family 'clothing'
...	...	(other product families)
GenderWomen	Numeric	number products gender 'Women'
...	...	(other gender categories: Men, Other)
BrandDolceGabbana	Numeric	number of products of the brand D&G
...	...	(other brand categories: top 18 brands)
bAddress	Categorical	address indicated by user for billing
bZip	Categorical	
bCity	Categorical	
bCountry	Categorical	
bRegion	Categorical	
sAddress	Categorical	address indicated by user for shipping
sZip	Categorical	
sCity	Categorical	
sCountry	Categorical	
sRegion	Categorical	
PaymentAttempts	Numeric	Payment attemps by user
IP_country	Categorical	
Sessions	Numeric	number sessions same day by user
Time spent on portal	Numeric	
Page views	Numeric	
ClickedPhotos	Numeric	
Browser	Categorical	
Device type	Categorical	

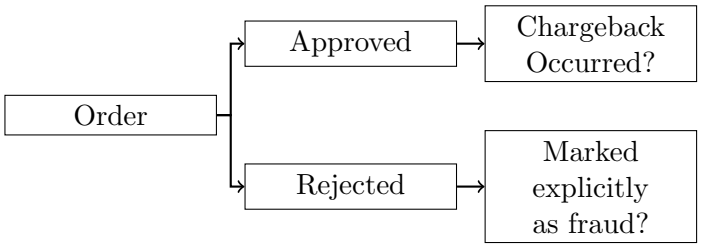


Figure 4.1: Criteria for labelling an order as fraudulent.

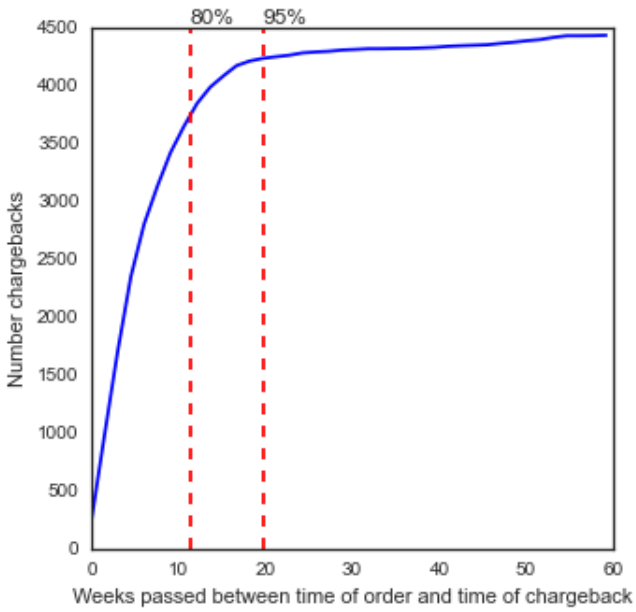


Figure 4.2: Cumulative distribution of the number of weeks before a chargeback is communicated (Farfetch data)

4.1.3 Time to Chargeback – how to choose the data set

It takes a long time for the merchant to realise that a fraudulent transaction has occurred. Most commonly, the alarm is first raised by the legitimate owner of the credit card, who notices a transaction in his bank statement that he did not make. Next, he must place a complaint on that transaction, which will originate a chargeback. It seems natural, that with such lengthy procedures, it takes several weeks between the time that the fraudsters uses the credit card and the merchant is noticed of the chargeback.

This restricts the orders we can use for data mining, because the most recent frauds won't have been detected yet, i.e. the client did not submit a chargeback or it was not yet communicated to the merchant. Hence, the first question to ask is: which data can we study? How long does it take to know if an order was legitimate or fraudulent?

The answer to these questions lies in knowing how long it will take between the time that the order is placed and the chargeback is registered in the system of the merchant. We found the answer by studying the chargeback history at Farfetch. We can see from Figure 4.2 that 80% of the chargebacks arrive within the first 11

weeks after the transaction.

Knowing this, it was decided that the data to be analysed should be no more recent than 16 weeks (circa 4 months). Thus, the data set was trimmed to only include orders older than July 2015. At the deployment stage, this criterion was also included in order not to take the most recent data in consideration when training the algorithm in the future.

4.1.4 Training and Testing set

The data set is split in two: a training and a testing set. Part of the data (the training set) is then used for the algorithm to learn how to classify orders. The remaining data (the testing set) is kept isolated, to be used just at the end of the modelling procedure to estimate the performance of the chosen model, removing the overfitting effect, i.e. the increase in performance that the algorithm has on the instances it has based its learning on, when evaluating new orders, due to the random error present in those particular instances.

According to the literature, the division between the number of observations in the testing set should lay between 20-40% of the total number of data entries. We chose to use 20% of them for testing, hence 80% of the data set was used for training.

Besides deciding the amount of entries which belong in each set, we must also decide which entries they are. The purpose of a testing set is to replicate as close as possible the performance of the model when evaluating new data. It stands to reason that randomly selected observations would emulate new data closer than observations filtered by any variable. However, we chose to take into account the date when the order was placed and consider the most recent 20% of orders for testing. The reasoning behind this was that patterns of fraud change over time and more recent orders would better emulate new orders.

The result of the division is a training set containing the orders placed between 2015-01-01 and 2015-06-21, totalling 347.572 observations; and a testing set containing the orders placed between 2015-06-22 and 2015-08-04, in a total of 86.893 orders.

Table 4.2: Key statistics of non-binary numeric variables

Name	Count	Mean	Std	Skewness	Kurtosis
Quantity	347572	1.84	1.65	7.28	159.82
OrderValue	347572	502.25	682.91	9.62	336.67
TimeCustomer	347572	13.87	6.45	-0.58	-0.51
TimeGMT	347572	12.34	6.63	-0.21	-1.07
TimeSinceFirstOrder (Weeks)	347572	39.48	58.34	1.84	3.23
NumCardsUsed	347572	1.65	2.31	10.19	220.43
FraudScore	347572	1.59	22.77	-9.38	156.89
PaymentAttempts	347572	1.46	1.95	18.03	843.38
Sessions	338579	3.33	3.13	3.17	25.68
TimeSpent	338579	53.71	64.28	3.92	30.31
TotalPageViews	338579	45.79	49.31	3.9	28.8

4.2 Exploratory analysis on the training data

The analysis in this chapter is done with data from the training set only. Before analysing the data, it is important to get an idea about the distributions of the numeric variables. Table 4.2 shows some statistics of the non-binary numeric variables.

The first thing to note is that the variables related to the browser session of the user (sessions, TimeSpent and TotalPageViews) only count 338.579 observations. This means that there are 8.993 orders for which the data on these variables is missing.

Looking at the table we can also note the high values of standard deviation (at the same level as the mean). This suggests that there are very distinct patterns of behaviour. The high kurtosis values indicate a peak around the mean value and a long tail of extreme values. We can also learn from the table that some variables' distributions are highly skewed, which is a sign that those distributions are not symmetric.

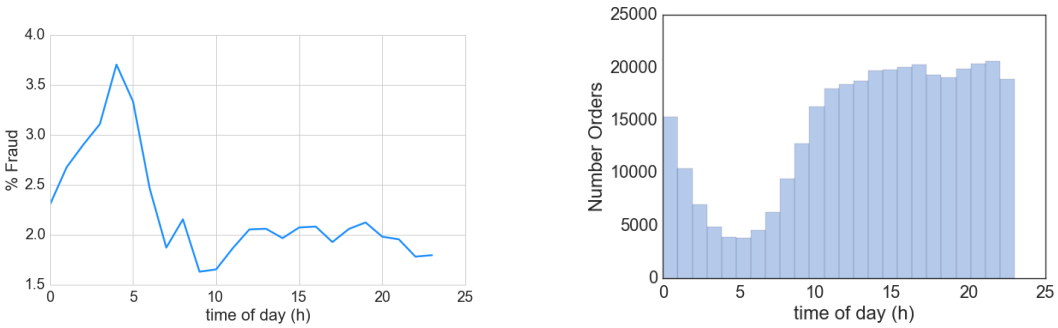
Considering that the orders in our sample correspond to a period of 172 days, we have on average 2020 orders per day. The classification algorithm must be fast enough to evaluate larger amounts of data per day, as the company is growing rapidly, so are its sales.

The following sub-sections present the most relevant results from the exploratory analysis on the relationship between individual variables and the occurrence of fraud.

4.2.1 Time when an order is placed

In order to have a better understanding of the influence of the time dimension in the data, we studied the variable *OrderTimeCustomer* which holds the information about the time at the customer's location when he placed the order.

On Figure 4.3a we can notice that there is a peak in the chart around 4am. This chart plots the ratio of fraudulent orders among all the orders placed in that hour of the day. Hence, the peak tells us that about 3.7% of all orders placed between 4 and 5am were fraudulent. We can tell from this figure that there is a trend for



(a) Fraud by time of day. *(b) Number of orders by time of day.*

Figure 4.3: Distribution of orders by time of day when they are placed.

Table 4.3: Division of fraudulent orders between day and night periods

	23pm to 10am	10am to 23pm
% all fraudulent orders	31%	69%

orders placed between midnight and 6am to have a higher number of fraudulent orders among them.

On the right, on Figure 4.3b, we can see the number of orders placed at each time of the day. A slope in the number of orders placed is noticeable between 1 and 9am. It is interesting to note that the number of orders placed at each hour is relatively stable between 10am and 23pm.

We can conclude that the great majority of fraudulent orders are placed during day time (10-23h) as is confirmed on Table 4.3. However, the orders placed during the night have a higher probability of being fraudulent.

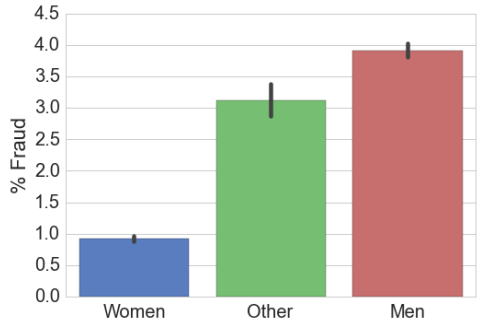
We recommend the fraud analysts to place a special attention to the orders which are placed during the period when it is night at the location of the customer.

4.2.2 Product gender

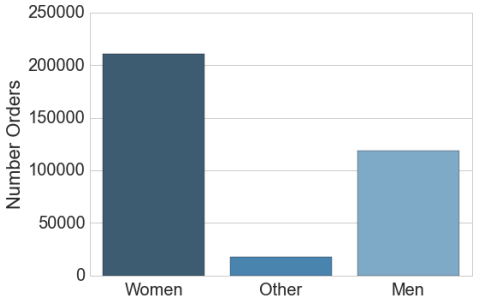
Very few orders include both products for men and for women. Hence, we consider an order of the gender "Women" if the majority of products in that order are of that type (e.g. three items for women and one item for men). The same goes for products of the gender "Men" and "Other", which includes all unisex products. This abstraction makes it easier for us to study the patterns of fraud related to product genders.

There is a noticeable majority of orders whose products' gender is "Women", as can be seen from Figure 4.4b. Despite the majority of orders being for products with gender "Women", two thirds (65%) of all fraudulent orders are of products of the gender "Men".

When taking a closer look at the occurrence of fraud in each of the genders in Figure 4.4a, we notice that only 0.9% of "Women" product orders are fraudulent, while this number reaches 3.9% for "Men" and 3.1% for "Other". One can conclude, that there is a higher prevalence of fraud in orders of products for men and unisex

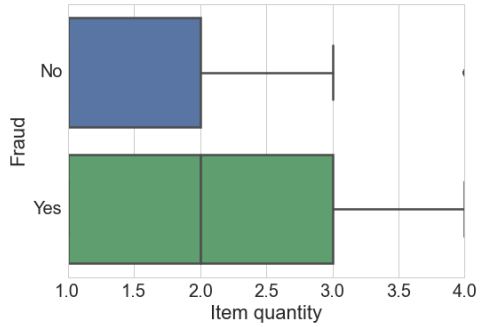


(a) Fraudulent orders by Gender.

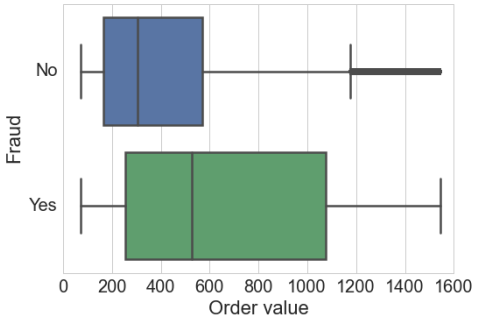


(b) Number of orders by Gender.

Figure 4.4: Distribution of variable product gender.



(a) Distribution of variable quantity in legitimate and fraudulent orders.



(b) Distribution of variable order value in legitimate and fraudulent orders.

Figure 4.5: Distribution of variables quantity and order value.

products.

4.2.3 Order quantity and value

As both the order quantity and value had a very high number of outliers, we applied a winsorizing technique to remove outliers before drawing the chat on Figures 4.5a and 4.5b. The data was only altered for plotting purposes, the data used for training was not subjected to this function.

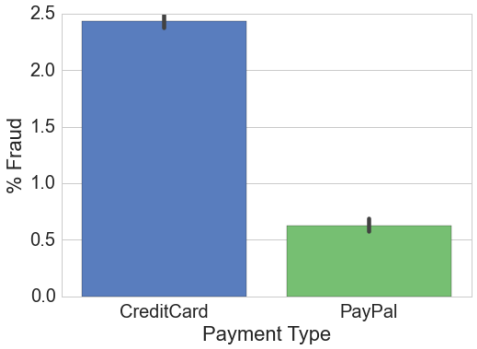
It can be seen on Figure 4.5a that fraudulent orders have a tendency to include a higher quantity of items than legitimate orders. It can be seen from the box plot that only 5% of all legitimate orders have more than three items, while that level corresponds to 25% of the fraudulent orders.

When it comes to the value of the order, we can observe the same pattern: fraudulent orders have on average a higher value. This can been seen on Figure 4.5b, where to two box plot graphs are slightly misaligned.

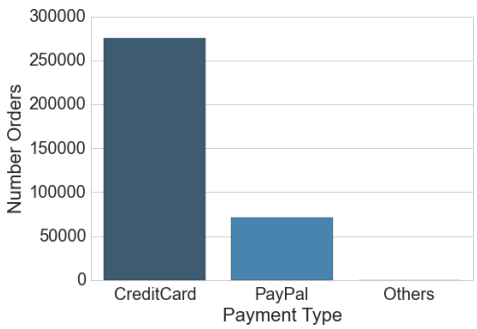
4.2.4 Payment type

Credit card and PayPal are the two main types of payment used in this dataset. Other payment types exist, but are negligible as can be seen on Figure 4.6b.

Figure 4.6a tells us that purchases made using a credit card happen to be



(a) Fraudulent orders by Payment Type.



(b) Number of orders by Payment Type.

Figure 4.6: Distribution of variable PaymentType.

fraudulent more often than purchases using PayPal. In fact, almost 2.5% of all credit card purchases are fraudulent, vs. 0.6% for PayPal. This is mainly due to the security measures implemented by the PayPal service, which make it such a popular alternative to credit cards as was discussed in Chapter 1 (Introduction).

4.2.5 Browser

The next analysis focused on recognizing differences in the level of fraud of orders made with different web browsers. On Figure 4.7a, we notice a significant difference in the level of fraud between browsers. Circa 6% of all orders made with the Firefox browser were fraudulent, while this value only reached circa 1.2% of the orders made with Safari. On an intermediate level, 3% of the orders made with Chrome were fraudulent.

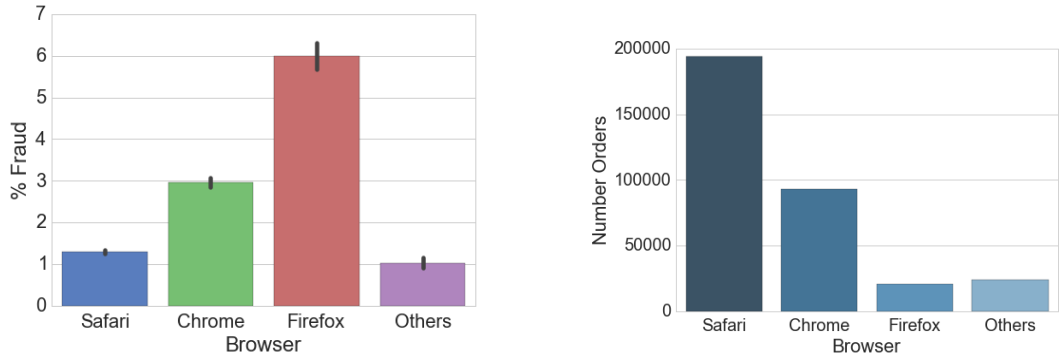
In order to know how many orders (legitimate and fraudulent) were made with each browser, we studied Figure 4.7b. About half of all orders were made with Safari (of which only circa 1.2% were fraud, as seen before).

These results tell us that the Browser used is probably a very explanatory variable. Before this study, the information about Browser used to make the purchase was not available for the Order Processing Team. We recommend that this variable be taken in consideration by this team, in particular that special attention is paid to orders made with the Browser Firefox.

4.2.6 Interaction with photos

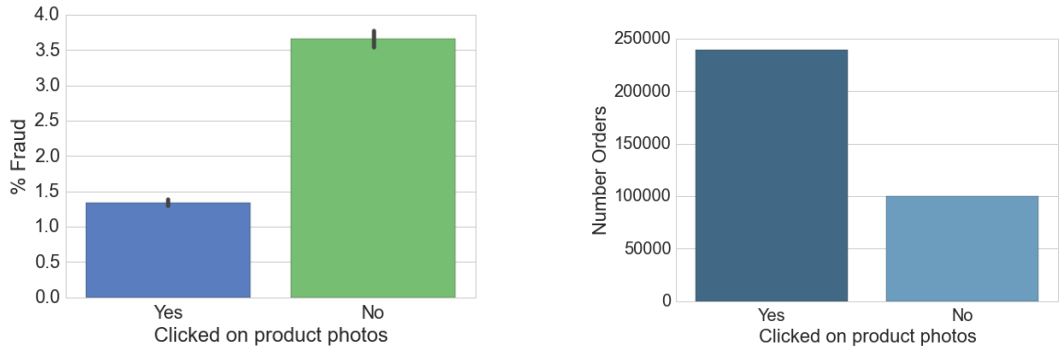
In the scope of studying the variables about the behaviour of the user on the website before making the purchase, we studied the variable *InteractedWithPhotos* which tells us whether the user has clicked on photos of products. The rationale behind this, was the intuition that fraudsters are not as interested in the product they are buying as legitimate users and might add items to their basket without inspecting their images.

In fact, orders where the user did not interact with photos on the day of the purchase have a higher probability of fraud. On Figure 4.8a we see that there are circa 3.6% of fraudulent orders out of all those where there was no interaction with photos, while only 1.4% in the case that inspection of the photos had happened.



(a) Occurrence of frauds by browser. *(b) Number of orders by browser.*

Figure 4.7: Distribution of variable Browser.



(a) Occurrence of fraud. *(b) 'Number of orders.*

Figure 4.8: Occurrence of fraud and total number of orders when user interacted with product photos before purchase or not.

Figure 4.8b shows the occurrence of each of the two cases in the total amount of orders studied. We see that circa one third of all orders are not preceded by interaction with the photos.

As with the information about the Browser used in the purchase, the behaviour of the user towards photographs, was not taken into account or even available to the Order Processing Team. After our analysis, we recommended that this information be displayed in the order approval dashboard and that the fraud analysts consider no interaction with photos as a suspicious sign of fraud.

4.2.7 Distribution of frauds per country

In order to understand the geographical dimension of fraud, we studied the occurrence of frauds in orders shipped to different countries. First, we calculated the ratio of fraudulent orders over the total number of orders in each country i:

$$Fraudratio_i = \frac{FraudulentOrders_i}{TotalOrders} \tag{4.1}$$

Countries with less than 30 orders, have been considered not significant and excluded from this analysis.

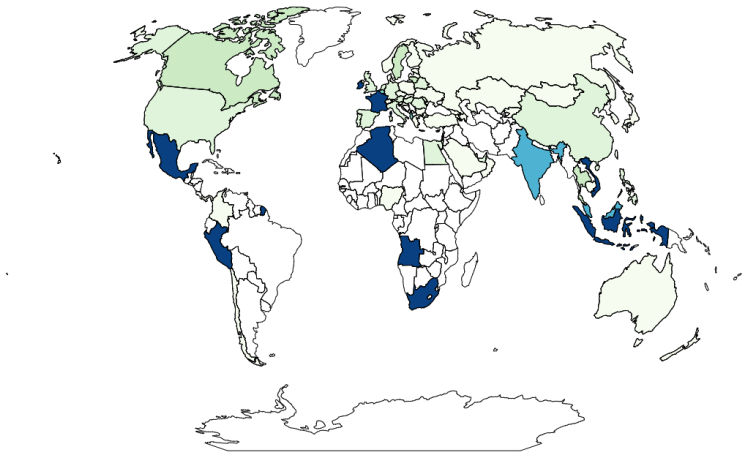


Figure 4.9: Fraudulent orders in each destination country.

The result of equation (4.1) is a list of pairs (country,ratio). We have plotted this list on a map of the world, as seen on figure 4.9. The darker colours represent countries where a bigger share of orders shipped to this country were fraudulent. The lighter colours represent the countries with a lower fraud level and countries in white are those which were excluded.

Mexico, Peru, Algeria, Angola, France, Ireland, South-Africa and Indonesia are some of the countries with a higher rate of frauds. This result is in line with the empirical experience of the Order Processing Team. When an order for any of these countries is placed, a fraud analyst must pay especial attention to it and look for other indications of fraud.

Chapter 5

Modelling and evaluation

5.1 Data transformation

The exploratory analysis presented in the previous chapter gives a first insight regarding the relevance and behaviour of each variable when detecting fraud. In this chapter we show how to transform and include the studied data set in machine learning algorithms, which will automatically combine those variables in a systematic and effective way. At the end, models are compared and the best is chosen for deployment in the case study.

In order to use machine learning algorithms such as Support Vector Machines, one must transform all categorical variables into numerical. Moreover, these must also be transformed through a standardization process, so that all values have the same scale [Miner et al., 2009].

We begin by filtering the variables which could not be used in the model, as described in Section 5.1.1. Next we perform transformations in order to make sure that all variables have a numeric representation, through one-hot-encoding (Section 5.1.2) and risk-level grouping (Section 5.1.3). We engineer new features through the use of several functions, detailed in Section 5.1.4.

It was seen on Section 4.2 that some variables had missing values. This is corrected by imputing a likely value, as described in Section 5.1.5. Lastly, the variables are standardized to the same value range, which is done according to Section 5.1.6.

All these transformations were applied separately on the training and testing set in order not to "contaminate" the data. For instance, the parameters for standardization of the training set were saved and used for the standardization of the testing set, as it would not be possible to calculate such parameters from a single new observation.

5.1.1 Feature selection

Unfortunately, not all variables which were studied in Chapter 4 could be used in the deployment stage. This constraint comes from the difficulty in calculating such variables when evaluating a new order. In particular, the variables related to the session of the user: *device and browser used, number of page views, number of sessions, time spent on the website and interaction with photos*, could not be

Table 5.1: Example of one-hot-encoding: variable PaymentType is transformed into 4 dummy variables

Order	Payment Type
1	Type A
2	Type B
3	Type C
4	Type A
5	Type A
6	Type D

→

Order	Type A	Type B	Type C	Type D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	1	0	0	0
5	1	0	0	0
6	0	0	0	1

accessed from the database in real time.

It would be possible to use them to train the algorithm, because they exist in the order history. However, we chose not to count on them in order to use a model which would work as closely as possible to the reality. The use of theses variable is an option for a future upgrade of the model.

The categorical variables which were considered for feature engineering, but had no meaning by themselves, were also deleted. This included all the address and name information. Other categorical variables were transformed as seen in sections 5.1.2 and 5.1.3.

5.1.2 One-hot-encoding for low-dimensional categorical variables

Categorical variables with few categories were transformed into several features through one-hot-encoding. One-hot-encoding is a method which consists of encoding a variable through binary numbers. In our case, we created one column for each of the categories of the variable. An example of this transformation can be seen on Table 5.1. This technique will allow the model to fit such variables, but as the dimensionality grows the model looses generality [Miner et al., 2009]. Hence, one-hot-encoding should not be applied to variables with an extensive number of categories, as to do so would increase the dimensionality of the sample too much.

5.1.3 Risk-level grouping for high-dimensional categorical variables

Variables such as the address country include too many categories (more than 200 different countries) to be transformed into individual features through one-hot-encoding. The choice was to transform the variables *sCountry* and *sCity* into features which translate the risk associated with each of the countries.

We calculated the value of the *fraud ratio* for each country as described in Section 4.2.7. Following that, the countries have been clustered in four groups. The clustering technique used is called Jenks natural breaks optimization (one-dimensional variant of k-means)and was developed by the cartographer George Jenks [of Kansas. Dept. of Geography and Jenks, 1977]. This technique seeks to minimize within-group variance, while maximizing between-group variance.

The number of clusters (four) was chosen by considering what could be a reasonable number of risk levels in such a problem. Alternatively, an iterative method for choosing the number of classes could have been used (North [2009] proposes such

a solution). Further alternatives could include creating individual features only for each of the most represented categories (e.g. US as the most represented category of variable *sCountry*).

The chosen method, to group categories into risk-level groups, does not take into account any multivariate effect, because the label "fraud" is the only criterion used for this risk evaluation. The method was however, the most practical way to transform the categories without greatly increasing the dimensionality of the sample.

This transformation was calculated using data from the training set only. When transforming an entry from the testing set, we check which risk-level had been assigned to that category in the training. Had such a category not been assigned a risk-level before (e.g. a country where no order from the training set was shipped to), the risk-level is set to the intermediate level 2.

5.1.4 Engineered variables

In order to get more significant variables to train the models with, we engineered new variables through abstraction and combination of variables.

The first engineered variables were created to represent the degree of similarity between certain pairs of categorical variables. We created a binary function which generated a new feature with the value of 1 in case of a match and 0 in case of no match. This function was applied to the pairs of variables {(billing country, shipping country), (billing country, card country), (shipping country, card country)}. These were all predefined country fields, which ensured the match could be done by a simple comparison of strings.

On the other hand, the variables referring to names (e.g. name on card, user name) would often not match because the names were spelled in different order. Hence, we used another function which calculates the similarity between two names. This function is based on n-gram similarity and its output is a continuous number in [0,1]. It was applied to the pairs of variables {(UserName, CardName), (billing city, shipping city), (billing zip-code, shipping zip-code)}.

A function which confirms that the customer has entered a valid telephone number was also created, by checking whether the input of the user is a number.

Lastly, a function summarizing the customer's recent buying behaviour was created. This function counts the number of orders by this customer in the ten days prior to the current purchase. The complete list of variables which were used in training can be seen in Table A.1.

5.1.5 Imputing missing values

In order to get the best performance of the machine learning algorithms, the data must be clean and complete. As was seen in section 4.2, there were missing values in the data set. We chose to complete these data entries by imputing a value in the missing variables, as suggested by Miner et al. [2009].

In the case of categorical variables, a missing value would correspond to a zero in each of the corresponding columns after the one-hot-encoding transformation. Hence, no further action was required.

In the case of numerical variables, the missing value was replaced by the mean of the sample. As an example, a missing value for *OrderValue* would be substituted

by the average order value of the sample of the training set. These parameters were saved and used for imputing missing values in the testing set.

5.1.6 Standardization of numerical variables using the Min-Max method

Support Vector Machines and other machine learning methods require data variables to be in the same range. In our case, variables which are measured in different units such as *Quantity* or *OrderValue* were standardized to values between 0 and 1, so that they can more easily be compared.

Among the several techniques for standardization, we chose to use Min-Max. This technique was preferred over the commonly used Z-value standardization, because it transforms variables to the range [0,1], which is consistent with the range of the many binary variables in the model. Indeed, Shalabi et al. [2006] compared three different techniques and concluded that the Min-Max method is the best design for standardization in data mining projects. Equation (5.1) describes the transformation applied to each numeric variable through the Min-Max scaling.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.1}$$

5.2 Model Selection

This section describes our work in selecting the model and parameters to use for the order classification. The models for study were Logistic Regression, Random Forests and Support Vector Machines. The choice of the models and respective parameters was done by applying cross-validation on the training set.

5.2.1 Performance measures

Before evaluating the results of each model, we had to decide which would be the performance measures to compare.

For each observation X, we have an associated real class label from the set $\{\mathbf{0,1}\}$ and a corresponding predicted label. Observations which are classified correctly as belonging to class 1 are called *true positives*, while observations correctly classified as class 0 are called *true negatives*. There are two other possible outcomes in which the prediction incurs in an error. The *type I error* occurs when the positive class is predicted, but the observation label is 0, these predictions are called *false positives*. The *type II error* occurs when the prediction of class 0 does not agree with the observation's true class which would be 1. Such errors are called *false negatives*. The two types of error can have different implications. In the case of fraud detection, a false positive error would correspond to a legitimate observation being labelled as fraud. A false negative error would happen in the case of a fraudulent transaction being classified as legitimate.

Table 5.2: Example of confusion matrix

		Real class	
		0	1
Predicted	0	True Negative	False Negative
	1	False Positive	True Positive

The most common way to visualize the error rate in classification problems is to draw a confusion matrix. This matrix depicts four quadrants with the number of true negatives, false negatives, false positives and true positives, respectively.

Several performance measures can be derived from the confusion matrix:

$$\text{True Positive Rate (known as } \textit{Recall}) = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

$$\text{True Negative Rate (known as } \textit{Specificity}) = \frac{\text{True Negatives}}{\text{True Negatives + False Positives}}$$

$$\text{Negative Predictive Value} = \frac{\text{True Negatives}}{\text{True Negatives + False Negatives}}$$

$$\text{Positive Predictive Value (known as } \textit{Precision}) = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

However, we can build a confusion matrix only if we can map each prediction to one of the classes. In the case of classifiers such as Random Forests, the output is a continuous value, that represents a score which tells us how close the observation is to 0 or 1. Therefore, a threshold must be defined in order to determine the final class (0 or 1), based on the real value obtained with the classifier.

ROC curves The *Receiver Operating Characteristic* curve (ROC) offers a way of visualizing different outcomes. Fawcett [2006] describes ROC curves as depicting the relative trade-offs between benefits (true positives) and costs (false positives). Observations with a score under the threshold are classified as class 0, while a score above the threshold would predict that the observation belongs to class 1. A ROC curve plots the *true positive rate* and *false positive rate* for each threshold between $-\infty$ and $+\infty$ [Fawcett, 2006].

An example of a ROC curve can be seen on Figure 5.1. The point (0,1) in the ROC graph would correspond to the perfect classifier, one which would not incur in any false positive classification and would get 100% of the true positive observations. On the contrary, the point (1,0) corresponds to the worst possible classifier. The closer the curve is to the top left corner, the better the performance of that classifier. A way of interpreting this graph is to use the area under the curve (AUC) as a measure of performance. Hanley and McNeil [1982] showed that the AUC is equivalent to the probability that the classifier will give a higher score to

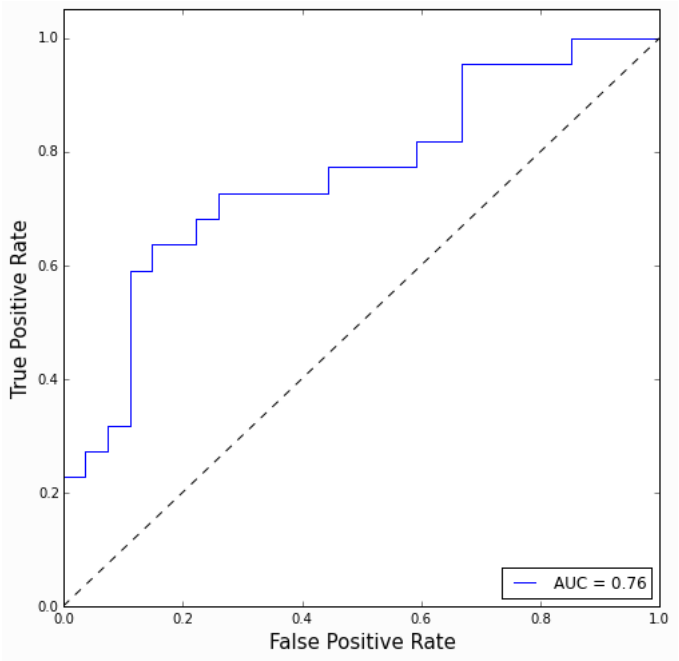


Figure 5.1: Example of a Receiver Operating Characteristic curve (ROC)

a randomly chosen observation of class 1 than to a randomly chosen observation of class 0. Fawcett [2006] notes that the AUC performs very well in practice as a general measure of classifier performance.

Precision-recall Precision-Recall (PR) curves have been used as an alternative performance measure to ROC. In a PR graph, we plot the precision and recall for each threshold. Davis and Goadrich [2006] studied the relationship between PR and ROC curves. One of the conclusion of this study is that optimizing the AUC-ROC will not guarantee the best result in AUC-PR. Therefore, we chose to use AUC-PR as the last measure of comparison between the performance of the different hypothesis in cross-validation.

5.2.2 Cross-validation

We want to state a hypothesis about a model which can make the best predictions about the distribution of our data. In order to do so we choose three families of models: Logistic Regression, Support Vector Machines and Random Forests. From these, we want to know which set of parameters generates a hypothesis which yields the best performance.

The whole data set was previously split into training and testing sets. Now a further split is performed on the training set. Part is used in actual training, while the rest, which we call the validation set, is used to help us find the best parameters of each model. We train the algorithm on a part of the training set and use the rest of the observations – the validation set – to test that hypothesis. The advantage in using a validation set is that the rest of the data in the training set has already been used in training the algorithm and is therefore not appropriate to be used in evaluating that same algorithm. The main negative aspect of validation is that it

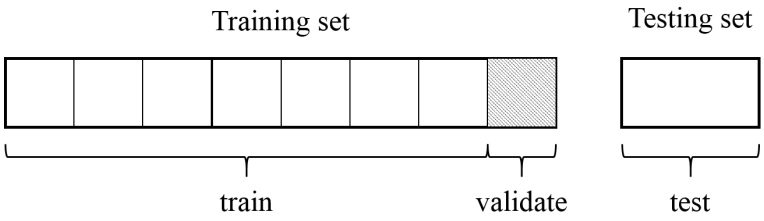


Figure 5.2: Example of the division of the dataset for cross-validation and testing

reduces the size of the training set.

Cross-validation is a modified way of doing validation. Instead of splitting the training set in two parts (training and validation), we can estimate the performance by repeating the training-validation procedure multiple times. One type of cross-validation is called the *leave-one-out* technique. With this technique we would train the algorithm on all data points N-1 and validate the hypothesis on the one point which was left out. We would then repeat this procedure N times and the performance measures would be estimated from that sample of results of size N. However, this approach suffers from the problem of requiring great computational time and it is not used in practice.

K-fold cross-validation is another type of cross-validation, which mitigates the problem of computational time. The training set is divided in K folds, with each of the folds being left out at a time for the training-validation sequence, as ilustrated on Figure 5.2. Abu-Mostafa et al. [2012] suggest 10-fold cross-validation as the common practice.

Algorithm 1: Cross-validation

```
Data: training dataset
Result: Estimate of performance of models with different parameters
initialization;
foreach Parameter set do
    Load classifier with new parameters
    Split into K-fold validation sets
    for k ∈ K do
        Train on train set
        Test on validation set
        Calculate performance measures - AUC-ROC and AUC-PR
    end
    Calculate mean and variance of the performance measures in the K
     validation runs
end
```

Algorithm 1 describes the steps implemented to do cross-validation. This procedure was repeated for the three models: Random Forests, Support Vector Machines and Logistic Regression. We validated several combinations of parameters for each model. The choice of the parameters to test was done by grid search, a technique which consists of performing an exhaustive search through a predefined

space of parameters. In some cases, a refined grid search was additionally performed after finding a suitable subset of the parameter space.

We also compared results when training on a balanced sample (i.e. equal number of fraudulent and legitimate records) achieved by randomly under-sampling the legitimate records vs. an unbalanced sample of all records. Each training set had 347.572 records of which 312.814 were used for training and 34.758 for validation at a time. In the case of under-sampling, the records used for training were reduced to a number around 13.000 (circa two times the number of fraudulent cases in nine tenths of the records), while the validation was done on all 34.758 records from the validation fold.

Random Forests and Logistic Regression were validated with both balanced and unbalanced data sets. Support Vector Machines has a much higher computational complexity and therefore it was just trained on the balanced sets generated by under-sampling.

Random Forests Random Forests is an ensemble method proposed by Breiman [2001] which consists of creating many decision trees and combining their estimates. The parameters which were varied were the minimum number of features to split each node of the tree (*Min. Split*), the criterion for quality of node split (ginni impurity or entropy), the number of trees, and whether a balanced set was used (*under-sampling*). The number of trees is not a true parameter, as more trees will always lead to a higher performance, but also more computational time. In fact, the better results were obtained with the highest number of trees we used (1500). However, the performance was practically the same as the performance when using 1200 or 900 trees. The top five results of this grid search can be seen on Table 5.3.

Under-sampling did not lead to a better performance. In fact, there is no indication that using a balanced sample by under-sampling will achieve a higher performance (see Appendix B). Regarding the criterion for split, it is clear that the best performance is achieved by using the entropy criterion.

Support Vector Machines Support Vector Machines is an advanced statistical classifier, which can make use of a kernel trick to map the data to a high dimensional feature space. We used an implementation of Support Vector Machines which yields a continuous probabilistic output. We employed a radial basis function kernel (*RBF*). The kernel coefficient Gamma and regularization term C were varied in a logarithmic scale. A high regularization term represents a weaker regularization. The best results were obtained for a value of C=10, as can be seen from Table 5.3.

Logistic Regression Logistic Regression is a widely used method for classification and regression. Due to memory limitations, we could not train a logistic regression on a higher order representation of the features (e.g. second or third order transformation). Like in Support Vector Machines, we varied the regularization term. The only solver used was an implementation of the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (*lbfgs*) optimization algorithm.

Validation results From Table 5.3, we can conclude that the performance of Support Vector Machines and Logistic Regression is similar. The best performance

Table 5.3: Results of cross-validation

Panel A: Grid search results for Random Forests

Random Forests					
Num trees	Min. Split	Criterion	Under-sampling	AUC-PR	AUC-ROC
1500	10	entropy	No	0,479	0,935
1500	12	entropy	No	0,477	0,935
1500	11	entropy	No	0,475	0,935
1500	6	entropy	No	0,475	0,935
1500	16	entropy	No	0,474	0,935

Panel B: Grid search results for Support Vector Machines

Support Vector Machines					
C	Gamma	Kernel	Under-sampling	AUC-PR	AUC-ROC
10	0,0464	RBF	Yes	0,337	0,906
10	0,0215	RBF	Yes	0,336	0,902
10	0,01	RBF	Yes	0,319	0,896
10	0,1	RBF	Yes	0,317	0,903
1	0,1	RBF	Yes	0,305	0,895

Panel C: Grid search results for Logistic Regression

Logistic Regression				
C	Solver	Under-sampling	AUC-PR	AUC-ROC
100,00	lbfgs	No	0,360	0,907
3,16	lbfgs	No	0,357	0,905
1,00	lbfgs	No	0,349	0,902
0,32	lbfgs	No	0,324	0,895
3,16	lbfgs	Yes	0,313	0,903

is achieved by Random Forests. The AUC-ROC is slightly higher than in the other two models and the AUC-PR is considerably higher. Taking this results into account we conclude that the best model for this problem is Random Forests.

Table 5.4: Testing results of the Random Forests model.

Classifier	AUC-PR	AUC-ROC
Random Forests	0,333	0,880

5.2.3 Testing results

After choosing the model, we want to estimate the performance of this algorithm on new data. We finally come to use the *testing set* which we separated from the rest of our data as described in Section 4.1.4. The performance in cross-validation has an optimistic bias, thus we expected to get a lower performance in testing.

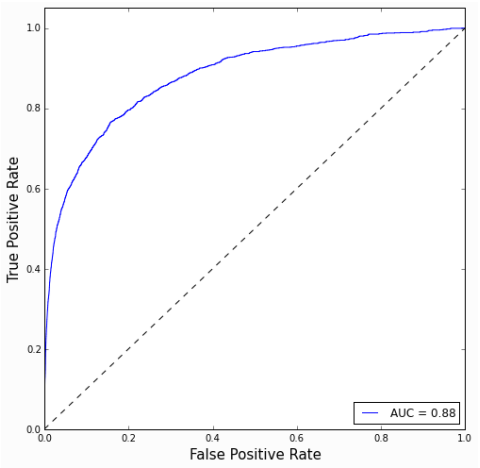
The purpose of testing is to estimate the performance of the model which we chose in training – Random Forests – with the parameters which achieved the best performance as seen in Section 5.2.2. This time, the complete data from the *training set* was used to train the classifier.

Table 5.4 shows the results of testing. Contrary to the performance results in validation (see Section 5.2.2), the value of the area under the precision-recall curve (AUC-PR) was not so satisfactory. However, Random Forests still presented a high value for the area under the receiver operating characteristic curve (AUC-ROC). The complete curves can be seen on Figure 5.3, on Figure 5.3a we can see the ROC curve. To the right on Figure 5.3b, we have the plot of the precision-recall curve. We will further explore these results.

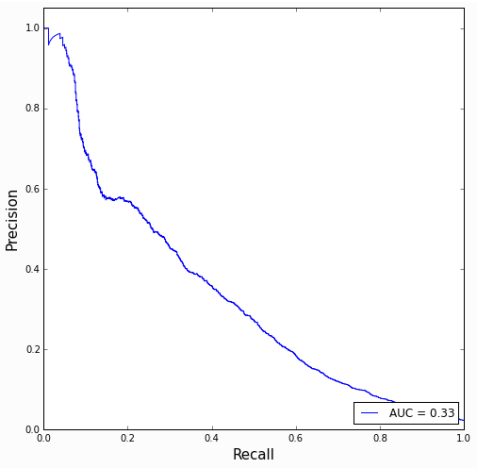
Table 5.5 shows the results of the suspicion scores estimated by Random Forests on the testing data, divided by bins of 5 basis points. The ratio of frauds per total orders increases monotonously with the score, which is a sign that higher scores represent a higher probability of fraud. The distribution of scores for legitimate and for fraudulent orders can be seen on the histograms on Figure 5.4. On the left, we see the distribution of the scores which were estimated for legitimate orders. On the right the scores estimated for fraudulent orders. The first thing to notice is that the distributions are different, which is a sign that the classifier recognized the two classes. There is a high occurrence of low scores for legitimate orders, which is what we would expect. On the other hand, fraudulent orders have a rather even distribution of scores. It would be expected that more fraudulent orders had high scores (near the value 1). This is a consequence of the imbalanced data set used for training, as the classifier can very well identify legitimate orders but has more difficulties with fraudulent observations.

Alternative results are shown in Appendix C. A higher AUC-PR indicates that using certain additional features in training, would increase performance. However, this case was only tested to provide an incentive for the company to develop their systems in order to enable the use of such features.

We must define a score threshold for considering a transaction legitimate or fraudulent based on its suspicion score. Once the threshold is defined, we can calculate business metrics such as the number of chargebacks (false negatives) and the number of refusals of legitimate payments (false positives). One approach to define the threshold is to look for the value which originates the point of the curve closest to the top left corner in the ROC space. The intersection of the Recall and Specificity curves in Figure 5.5 is mapped to the threshold value (0.22) which achieves



(a) ROC curve.

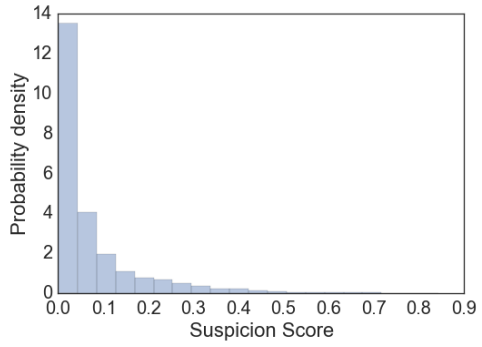


(b) Precision-Recall curve.

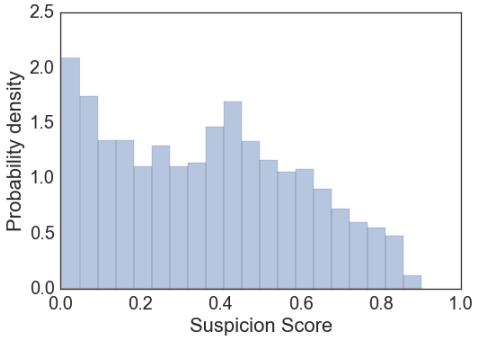
Figure 5.3: ROC and PR curves of testing results with Random Forests algorithm.

Table 5.5: Number of legitimate and fraudulent orders whose score falls into each range

Range	Total	Fraud	OK	Ratio
0 - 5	52469	180	52289	0,34
5 - 10	13510	159	13351	1,18
10 - 15	6810	127	6683	1,86
15 - 20	3828	115	3713	3
20 - 25	2941	111	2830	3,77
25 - 30	2197	109	2088	4,96
30 - 35	1445	106	1339	7,34
35 - 40	1022	112	910	10,96
40 - 45	914	161	753	17,61
45 - 50	564	129	435	22,87
50 - 55	365	110	255	30,14
55 - 60	213	86	127	40,38
60 - 65	194	102	92	52,58
65 - 70	190	85	105	44,74
70 - 75	111	57	54	51,35
75 - 80	59	51	8	86,44
80 - 85	43	42	1	97,67
85 - 90	17	17	0	100
90 - 95	1	1	0	100
95 - 100	0	0	0	-



(a) Scores of legitimate orders.



(b) Scores of fraudulent orders.

Figure 5.4: Histograms of score predictions for each of the labels.

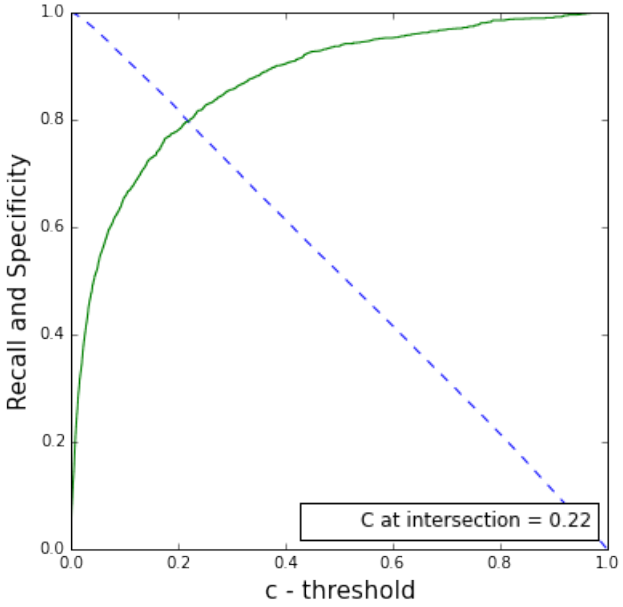


Figure 5.5: Plot of Recall and Specificity for each threshold.

the compromise between the two measures.

Fixating the threshold at 0.22, we can draw the confusion matrix of results, which can be seen on Table 5.6.

5.2.4 Discussion of results

Table 5.6 shows a high number of false positives (9.09%). In case this classifier would be processing all orders, the estimated number of chargebacks would be of 0.71% (number of false negatives), while 1.42% of all orders would be correctly rejected as fraudulent. However, it is not realistic that such a classifier would be implemented by itself in an e-tail situation such as in our case study.

It is important to relate the performance measures to the data mining objectives of the project and the business goals. Hence, we must also estimate the performance for the case of using the classifier together with manual review. This approach can give us much more interesting results.

Table 5.6: Confusion matrix of results when setting threshold at c=0.22.

	Legitimate (actual)	Fraud (actual)	Total
Legitimate (predict)	77129 (88.76%)	618 (0.71%)	77747
Fraud (predict)	7904 (9.09%)	1242 (1.42%)	9146

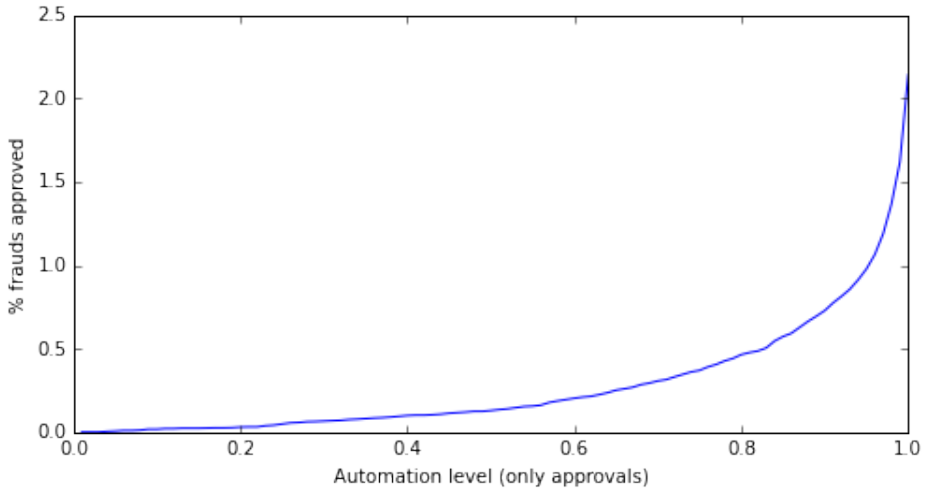


Figure 5.6: Plot of automation level and fraud.

The new approach is based on the assumption that all orders below the threshold would be automatically approved, while the rest (orders with higher score) would be manually reviewed. Then we chose the score threshold based on the number of orders which can be manually reviewed. In accordance with the business objectives, we determined the level of automation to be at 80%. Hence, the threshold was set at the value which is higher than 80% of the score of all orders. The advantage of this approach is that it can be easily adapted to new business requirements. If the company is interested in taking a lower risk, it can choose to decrease the number of automatically approved orders. Figure 5.6 shows how many fraudulent orders would be automatically approved if we vary the level of automation.

When choosing the level of automation, we can build a confusion matrix based on a few assumptions, such as:

- The score threshold splits the orders which would be automatically approved (score under the threshold) from the ones which would be reviewed (score above the threshold);
- When a fraudulent order is manually reviewed, there is a 75% probability that it will be refused;
- When a legitimate order is manually reviewed, there is a 90% probability that it will be accepted.

Table 5.7: Confusion matrix of results when automatically approving 80% records with lowest estimated suspicion scores.

	Legitimate (actual)	Fraud (actual)	Total
Legitimate (predict)	83441 (96.02%)	768 (0.88%)	83914
Fraud (predict)	1592 (1.83%)	1092 (1.25%)	2684

Automation level:	80%
Recall:	0.587
Specificity:	0.981
Fallout:	0.019
Precision:	0.407

Hence, the number of items approved can be calculated as:

$$\begin{aligned}\text{True Negatives} =& \text{Legitimate orders automatically accepted} \\ &+ 0.9 \times \text{Legitimate orders manually reviewed}\end{aligned} \tag{5.2}$$

$$\begin{aligned}\text{False Negatives} =& \text{Fraudulent orders automatically accepted} \\ &+ (1 - 0.75) \times \text{Fraudulent order manually reviewed}\end{aligned} \tag{5.3}$$

$$\text{False Positives} = (1 - 0.9) \times \text{Legitimate orders manually reviewed} \tag{5.4}$$

$$\text{True Positives} = 0.75 \times \text{Fraudulent orders manually reviewed} \tag{5.5}$$

Table 5.7 shows the results of the combination of automatic classifier with manual review, under the assumptions mentioned before. The results show a high value of specificity. Specificity can be interpreted as the fraction of legitimate orders which are approved. With our results, circa 98% of legitimate orders would be approved. On the other hand, the recall or sensitivity value is rather low. Only circa 59% of the fraudulent orders would be refused. Precision is the fraction of fraudulent orders out of all which were refused. We can see that 41% of the refused orders were actually fraudulent. Fallout can be interpreted as the "false alarm" rate, i.e. the probability of falsely rejecting a legitimate order. This value is equal to 1 minus the value of specificity, at circa 2%.

The combination of the Random Forests classifier with the manual revision of orders, seems to yield very good results in increasing the automation level and having a low rate of customer insults (fallout). The maximization of the number of chargebacks avoided (recall) is not so successful. However, we believe that this is a limitation of the problem of fraud detection itself, as fraudsters work hard to hide their intentions, as previously mentioned.

The Random Forests models also allows us to estimate the relative importance of each of the features. We can see the importance of the top 10 attributes on Table 5.8. Attributes which were represented by many features such as the product brand have been aggregated into one attribute in order to have a better overview

Table 5.8: Relative importance of the top 10 attributes.

Importance	Feature	Description
13,3%	TimeSinceFirstOrder	Time elapsed since user's first purchase at Farfetch
10,3%	OrderValue	Value of order
7,6%	Brand	All the features describing the product brand
6,3%	sCity_RiskLevel	Engineered variable - risk level associated with shipping city
6,1%	Quantity	Quantity of items in the order
5,9%	Gender	Gender of items in the order
5,4%	Similarity (UserName/CardName)	Engineered variable - similarity between username and name on credit card
4,9%	NumCardsUsed	Number of cards used by customer at Farfetch
4,9%	PaymentAttempts	Payment attempts by user for this order
4,6%	OrderTimeGMT	Absolute time at which the order is placed

of their importance. The importance of each variable is calculated as the "mean decrease impurity", described in Breiman et al. [1984].

We can note that the time since the customer's first order is the most important attribute. The value and quantity of items in the order also appear to be relevant features. On the product perspective, its brand and gender are together responsible for the same level of importance as the customer's first order date. The risk associated with the city where the order will be shipped to is the fourth most important attribute. A few features related to the payment itself also show significant importance. These are the similarity between the name on the credit card and the name of the customer, the number of cards used by the customer in all his orders at this merchant and the number of repeated payments attempts for this order.

Other variables such as the currency used or the similarity between billing and shipping address ranked lower in terms of relative importance. The information we get from this analysis can be of use to improve the manual review process and in building new models in the future.

5.3 Deployment at Farfetch

The deployment of the selected model involves additional issues, such as the way the model will be updated (i.e. periodically trained with new data) and the integration with Farfetch's existing systems and services. This section explores these issues and shows some of the first results Farfetch is getting from using our solution.

5.3.1 Order Classification Service

In order to maintain the relevance of the classifier, it is important to update the data used for training. We created a recurring job on the database server to update the table where the data set is saved. This job runs two procedures: one which aggregates the information about the new orders and appends it to the table, and a second procedure which updates the variable *LabelFraud* in case there was a chargeback on the payment of an order.

The final model is to be integrated in the order classification service. This application will evaluate new orders and return a suspicion score. It contains two main procedures:

1. Training the algorithm

2. Classifying orders

Figure 5.7 illustrates the main parts of the system. Training the algorithm happens once every week. After the data set in the database has been updated, the training script will download the records. As was seen in Section 4.1.3, it takes several weeks a chargeback is submitted. Thus, the training script will only use data from orders up to four months before the current time. As there is no need for a *testing set*, all downloaded data is used in training. The output of the training is a set of parameters for the transformation of the data and a classifier which will generate the suspicion score.

The classification service runs continuously, only stopping when loading new parameters after the training occurs. When a new order is placed on the Farfetch portal, it will be submitted for evaluation through the classification service's application programming interface (API). At first, the service will parse the information on the request to map each field to the respective variable. Not all variables used in classification are inherent to the order. This practical issue prevents all required information to be sent together with the request. The application must make its own request to the database to query the extra information, such as the time of the first purchase of this user. Finally, the service will apply the transformation functions to the data, so that it will generate the exact same features as in training. The preprocessing parameters saved from the training, such as risk-level mapping of countries are used in this step.

With all the features in the right format, the classifier is called to estimate the suspicion score for the order. This score is returned as the answer of the classification request.

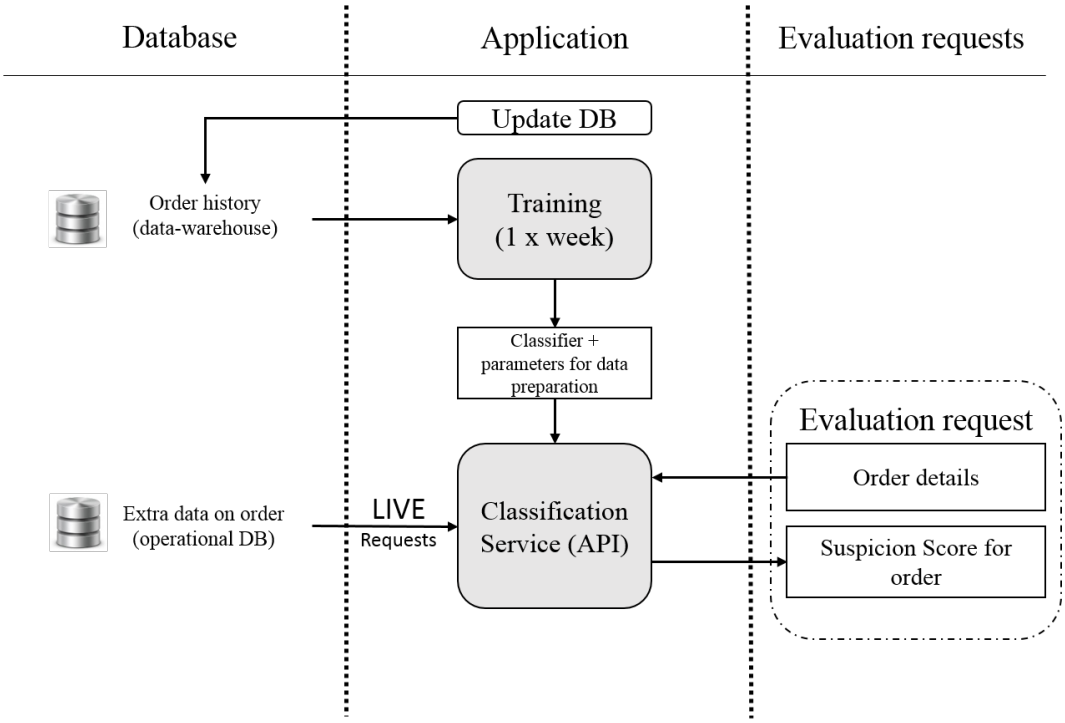
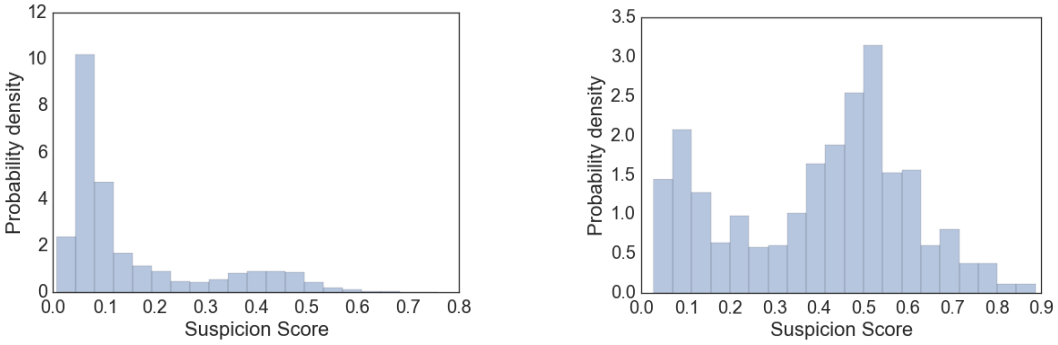


Figure 5.7: Diagram of the Order Classification System

5.3.2 Results of implementation

The time between a issuing a single request and getting the answer from the Order Classification Service amounts to circa 1200ms. During out tests, batches of fifty requests were processed in circa 1min30s. This is more than enough for a prompt response.

Figure 5.8 shows the first results of scores evaluated by this service in real time at Farfetch. At this stage, the classification service is taking no direct decision to approve or reject orders, but estimating the suspicion score. We will not know for sure which orders were fraudulent before a few months have passed, but we can compare the scores estimated for orders which were cancelled by the Order Processing Team for suspicion of fraud, against those of orders which have been accepted. Figure 5.8a shows the distribution of scores for all those orders which were accepted. On the right, Figure 5.8b shows this distribution of scores for the orders which were not accepted. The clear difference in the distribution of scores in the two cases, indicates that the classifier has distinguished between these orders. A further analysis should compare the results of orders which originated chargebacks.



(a) Orders not cancelled. *(b) Orders cancelled for fraud suspicion.*

Figure 5.8: Comparison of scores distributions of orders approved vs. cancelled by Order Processing Team.

Chapter 6

Conclusion and Future Work

This work addressed the problem of fraud detection for retailers doing business online. The project had the goal of developing a risk scoring solution through the use of data mining techniques, which was met with the implementation of such a system at the case study e-tail merchant Farfetch.

One of the limitations encountered was that a single online retailer does not have a lot of information about the customer it is doing business with. For instance, it would not know if this customer has been banned from other websites. For this reason, it is imperative that the retailer uses the available data as best as possible. The choice of which variables to use was very important and could be done thanks to the empirical experience provided by the fraud analysts at the case study company. Besides, it was important to explore the database and understand which other variables could be used, which were not obvious at first.

The most relevant features turned out to be features which were engineered out of one or multiple base variables. The time since the customer's first order, the similarity measures between names or the grouping of cities by risk are examples of such features. The exploratory analysis showed which variables are particularly relevant on their own, as an indicator of fraud. The patterns identified in this analysis can be used to improve fraud detection through manual review. Future work comparing different transformation functions and providing guidance on which features to engineer would be very useful for this area. We recommend studying the possibility of grouping addresses, which would require matching addresses that are spelled differently.

We confirmed that supervised learning methods are applicable to fraud detection in an e-tail merchant setting. The three machine learning algorithms tested, Logistic Regression, Support Vector Machines and Random Forests, provided good results. Random Forests achieved the highest performance. This algorithm seems to be very adequate to be used for fraud detection due to its good performance, combined with ease of implementation and fast computation time even with large datasets. We concluded that using a balanced set of observations through undersampling of legitimate records, was not significantly different in performance than using the much bigger imbalanced full set of records. We would suggest future work to explore whether oversampling of fraudulent records would lead to a different conclusion.

The testing results with the Random Forests algorithm showed that such a model would perform well enough to be of practical application. The advantages

of a data mining approach are the possibility of automatically processing a large volume of orders, allowing e-tail businesses to expand sustainably. When using a continuous classifier such as Random Forests, it is critical to choose the score threshold for considering an order to be legitimate or fraudulent. We came up with an approach based on the assumption that the merchant will manually review orders with a score above the threshold, instead of directly cancelling them. We provide the merchant the possibility to define the threshold by choosing the share of orders which should be automatically processed. We recommend the exploration of other ways to choose the score threshold, such as methods based on the business targets for fraud tolerance and customer insult rates. Such a cost-based performance measure could be used in training the algorithm in order to further direct learning towards the intended business outcomes.

The results obtained, under the assumption that 80% of the orders in the testing set with the lowest scores would be automatically approved and the rest reviewed manually, showed a very high number of legitimate orders approved and a false alarm rate of only circa 2%. On the other hand, the probability of rejecting a fraudulent order was only of circa 59%. Still, the number of chargebacks would be at 0.88% which is well under the industry average of 1.6% for international orders in North America [CyberSource, 2013].

The contribution for Farfetch was the implementation of this approach as a risk scoring application. The system developed extracts the data from the database, prepares it, trains the algorithm and runs a web application which gets requests with order details and estimates the risk score for the order. At the end of this project, the system was implemented at Farfetch and the first results of real-time order evaluations are now being assessed.

Finally, this work details the many steps that need to be taken to develop a complete application of fraud detection. We hope that our case study provided some insights into fraud detection in the perspective of merchants and that others will be motivated by it to further explore the problem of presenting solutions for fraud detection in e-tail.

References

Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. Learning from data. AML-Book, 2012.

T. P. Bhatla, V. Prabhu, and A. Dua. Understanding Credit Card Frauds. Cards business review, 1(6):1–15, 2003.

S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland. Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3):602–613, 2011. ISSN 01679236. doi: 10.1016/j.dss.2010.08.008. URL http://linkinghub.elsevier.com/retrieve/pii/S0167923610001326.

R. J. Bolton, D. J. Hand, F. Provost, L. Breiman, R. J. Bolton, and D. J. Hand. Statistical Fraud Detection: A ReviewCommentCommentRejoinder. Statistical Science, 17(3):235–255, 2002. ISSN 08834237. doi: 10.1214/ss/1042727940.

L. Breiman. Random forests. Machine learning, pages 5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL http://link.springer.com/article/10.1023/A:1010933404324.

L. Breiman, J. Friedman, C. Stone, and R. Olshen. Classification and Regression Trees. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. ISBN 9780412048418. URL https://books.google.pt/books?id=JwQx-WOmSyQC.

P. K. Chan and S. J. Stolfo. Toward Scalable Learning with Non-uniform Class and Cost Distributions : A Case Study in Credit Card Fraud Detection 1 Introduction. n Proceedings of the Fourth In- ternational Conference on Knowledge Discovery and Data Mining, pages 164–168, 1998.

P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. CRISP-DM 1.0 step-by-step data mining guide. Technical report, The CRISP-DM consortium, Aug. 2000. URL http://www.crisp-dm.org/CRISPWP-0800.pdf.

C. Cortes, D. Pregibon, and C. Volinsky. Communities of interest. Springer, 2001.

CyberSource. 2013 online fraud report. Technical report, CyberSource, 2013. URL http://www.cybersource.com.

J. Davis and M. Goadrich. The Relationship Between Precision-Recall and ROC Curves. Proceedings of the 23rd International Conference on Machine learning – ICML'06, pages 233–240, 2006. ISSN 14710080. doi: 10.1145/1143844.1143874. URL http://portal.acm.org/citation.cfm?doid=1143844.1143874.

T. Fawcett. An introduction to ROC analysis. Pattern Recognition Letters, 27(8): 861–874, 2006. ISSN 01678655. doi: 10.1016/j.patrec.2005.10.010.

T. Fawcett and F. Provost. Adaptive Fraud Detection. Data Mining and Knowledge Discovery, 316(1):291–316, 1997. ISSN 1384-5810, 1573-756X. doi: 10.1023/A:1009700419189.

Ghosh and Reilly. Credit card fraud detection with a neural-network. 1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences, 3:621–630, 1994. doi: 10.1109/HICSS.1994.323314.

J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology, 143(1):29–36, 1982.

W. McKinney. Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 1697900(Scipy):51–56, 2010. URL `http://conference.scipy.org/proceedings/scipy2010/mckinney.html`.

D. Meyer, F. Leisch, and K. Hornik. The support vector machine under test. Neurocomputing, 55(1–2):169 – 186, 2003. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/S0925-2312(03)00431-4. URL `http://www.sciencedirect.com/science/article/pii/S0925231203004314`. Support Vector Machines.

G. Miner, R. Nisbet, and J. Elder IV. Handbook of statistical analysis and data mining applications. Academic Press, 2009.

D. Montague. Essentials of Online payment Security and Fraud Prevention. Essentials Series. Wiley, 2010. ISBN 9780470915141. URL `https://books.google.pt/books?id=3IJCmhWztBIC`.

Y. Moreau, E. Lerouge, H. Verrelst, C. Stormann, P. Burge, and K. U. Leuven. A hybrid system for fraud detection in mobile communications. Neural Networks, (April):447–454, 1999.

M. a. North. A Method for Implementing a Statistically Significant Number of Data Classes in the Jenks Algorithm. 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, pages 35–38, 2009. doi: 10.1109/FSKD.2009.319. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5358673`.

U. of Kansas. Dept. of Geography and G. Jenks. Optimal data classification for choropleth maps. 1977.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

C. Phua, V. Lee, K. Smith, and R. Gayler. A comprehensive survey of data mining-based accounting-fraud detection research. International Conference on Intelligent Computation Technology and Automation, 1:50–53, 2010. doi: 10.1109/ICICTA.2010.831.

J. T. Quah and M. Sriganesh. Real-time credit card fraud detection using computational intelligence. Expert Systems with Applications, 35(4): 1721–1732, 2008. ISSN 09574174. doi: 10.1016/j.eswa.2007.08.093. URL http://linkinghub.elsevier.com/retrieve/pii/S0957417407003995.

G. Rossum. Python reference manual. Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.

L. A. Shalabi, Z. Shaaban, and B. Kasasbeh. Data Mining: A Preprocessing Engine. Journal of Computer Science, 2(9):735–739, 2006. ISSN 15493636. doi: 10.3844/jcssp.2006.735.739.

A. Srivastava, A. Kundu, S. Sural, and S. Member. Credit Card Fraud Detection Using Hidden Markov Model. Ieee Transactions on Dependable and Secure Computing, 5(1):37–48, 2008. ISSN 15455971. doi: 10.1109/TDSC.2007.70228.

C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams. Transaction aggregation as a strategy for credit card fraud detection. Data Mining and Knowledge Discovery, 18(1):30–55, 2009. ISSN 1384-5810. doi: 10.1007/s10618-008-0116-z. URL http://www.springerlink.com/index/10.1007/s10618-008-0116-z.

Appendices

Appendix A

List of features used in training the algorithms

Table A.1: Final list of features used in training.

#	Name	Values
1	LabelFraud	{0,1}
2	OrderTimeGMT	[0,23]
3	OrderValue	$\mathbb{R}_{>0}$
4	CV2Code_match	{0,0.5,1}
5	AVSCode_match	{0,0.5,1}
6	Quantity	$\mathbb{Z}_{>0}$
7	FamClothing	$\mathbb{Z}_{\geq 0}$
-	...	
16	FamOther	$\mathbb{Z}_{\geq 0}$
17	GenderMen	$\mathbb{Z}_{\geq 0}$
18	GenderWomen	$\mathbb{Z}_{\geq 0}$
19	GenderOther	$\mathbb{Z}_{\geq 0}$
20	BrandDOLCEGABBANA	$\mathbb{Z}_{\geq 0}$
-	...	
41	BrandOther	$\mathbb{Z}_{\geq 0}$
42	CustomerCurrency:EUR	{0,1}
-	...	
52	CustomerCurrency:USD	{0,1}
53	PaymentType:A	{0,1}
54	PaymentType:B	{0,1}
55	PaymentType:C	{0,1}
56	PaymentType:D	{0,1}
57	NumCardsUsed	$\mathbb{Z}_{\geq 0}$
58	PaymentAttempts	$\mathbb{Z}_{\geq 0}$
59	isValidUserPhone	{0,1}
60	bCountryCode=sCountryCode	{0,1}
61	bCountryCode=CardCountry	{0,1}
62	sCountryCode=CardCountry	{0,1}
63	bRegion=sRegion	{0,1}
64	TimeSinceFirstOrder	$\mathbb{R}_{\geq 0}$
65	3gramDistance(UserName/CardName)	[0,1]
66	3gramDistance(bCity/sCity)	[0,1]
67	3gramDistance(bZip/sZip)	[0,1]
68	sCountryCode_RiskLevel	{1,..,4}
69	sCity_RiskLevel	{1,..,4}
70	BinaryShippingExpress	{0,1}
71	NumLast10DaysSameUser	$\mathbb{Z}_{\geq 0}$

Appendix B

Comparing Random Forests performance with balanced and imbalanced training sets.

While doing the grid search for the best parameters of the Random Forests algorithm, we have trained all parameter combinations on both a balanced and an imbalanced data set. The imbalanced data set was the original data set, the nine folds of the training set used in each run of the 10-fold cross-validation. The balanced data set was a sample of the same nine folds obtained by undersampling the data. The data was undersampled in such a way that all fraudulent orders, n, were sampled and only the same number n of legitimate orders were sampled. Due to this sampling approach, the balanced data set was much smaller than the imbalanced set. A balanced set has the advantage of not being dominated by legitimate transactions, but the disadvantage of having only a fraction of the data points.

We calculated the difference between the performance measures AUC-PR and AUC-ROC obtained by the same set of parameters in the case of training with a balanced vs. imbalanced set. In case one of the sets would perform better on average, we would expect that this difference be different from zero. Hence, we made a hypothesis test with a t-test under the assumption that the distributions of differences are Gaussian. The hypotheses are shown on equation B.1.

$$\begin{aligned} H_0 &: \mu_1 = 0 & \qquad H_0 &: \mu_2 = 0 \\ H_1 &: \mu_1 \neq 0 & \qquad H_1 &: \mu_2 \neq 0 \end{aligned} \tag{B.1}$$

where:
μ_1 = average difference AUC-ROC
μ_2 = average difference AUC-PR

We obtained a p-value of 0.61 for the test on the average difference of the AUC-ROC and of 0.48 on the average difference of the AUC-PR. On a confidence level of 95%, we cannot reject any of the null hypotheses. This means that there is no evidence that there is a difference in the performance when using a balanced set obtained by undersampling.

Appendix C

Results of testing with additional features

Some of the variables which were studied in the exploratory analysis had to be left out of training, because they could not be calculated in real-time due to practical restrictions at the company as described in 5.1.1. These variables showed promising potential to be indicative of fraud. Therefore, we chose to make a simulation of the results which could be obtained by training and testing the Random Forests algorithm on a data set from which the variables *Device, Browser and ClickedPhotos* have not been removed. The aim is to show that the inclusion of these variables improves the model performance, so that the company invests in the infrastructure to make these variables available in real-time.

We see on table C.1 that there is a significant increase in the area under the precision-recall curve (AUC-PR) from the 0.333 obtained before. The increase in the area under the ROC curve is smaller, only up from 0.880. We would need to repeat the testing with multiple testing sets to be sure of the significance of the higher performance. However, these results seem to indicate that there is a reason to improve the infrastructure to be able to include such variables.

Table C.1: Testing results with additional features.

Classifier	AUC-PR	AUC-ROC
Random Forests	0,404	0,895