Predicting Product Returns for a Fashion Business

A study on the ability of machine learning algorithms to predict product returns for a fashion business

Giel Hofstee

Snr: 2047840

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee

Supervisor:	dr. Peter Hendrix
Second reader:	dr. Afra Alishahi
External supervisor:	Stefanie Schouten

Tilburg University
School of Humanities & Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
December 2, 2022

Word count: 8742

Abstract

This study aims to find out how well machine learning algorithms can predict return flows for a fashion business operating online and in retail. Research has been conducted on the topic of return prediction, but the focus is solely laid on e-commerce, and retail returns are ignored. Also, no broad comparison of several different machine learning algorithms has been made so far. This study aims to bridge this gap and contribute with new insights and elaborate on findings in previously conducted research. A dataset of the fashion business Annadiva is used to train and test models. The comparison between the different algorithms gave some interesting findings. The sophisticated algorithms that were expected to outperform the logistic regression baseline did not (overwhelmingly) do so. F1-score performances were similar and did not outperform the baseline with more than 0.0032 for in-store and 0.0365 for webshop orders. The in-store return predictability is significantly higher than that of webshop returns, with mean in-store models F1-score of 0.9638 and 0.7341 for webshop models. The feature importance differed between in-store and webshop models, with only six features appearing in both top 20 lists. It can be stated that it is possible to predict order returns with high certainty, especially for in-store orders.

Source of data, code ethics statement

Work on this thesis did not involve collecting data from human participants or animals. The original owner of the data and code used in this thesis retains ownership of the data and code during and after the completion of this thesis. The author of this thesis acknowledges that they do not have any legal claim to this data or code. The code used in this thesis is not publicly available. All tables and figures used in this thesis were produced by the author.

Contents

1. Introduction

1.1. Project definition

Fashion businesses want to make sure their customers are happy. One way to make this happen is to be flexible about their return policy. Consumers can often return anything they want for free, without even having to return it to a postal office or store themselves. This policy impacts costs and the environment in a negative way. This research aims to compare different means of predicting item returns in the fashion industry.

To achieve this aim, several machine learning algorithms will be compared in terms of performance. These algorithms will be fed with historical data about orders (item data, customer data, order data). Returns will be predicted for each product ordered. So for every individual orderline.

1.2. Motivation

Scientific relevance

Machine learning algorithms have been applied to numerous problem domains, and have proven to be successful, in identifying fake websites (Abbasi et al. 2010), sales forecasting (Choi et al. 2011), detecting credit card fraud (Bhattacharyya et al. 2011), detecting financial fraud (Abbasi et al. 2012), and credit scoring (Zhang et al. 2010).

When looking at return prediction, some studies have been conducted in several fields, like product return volume (Cui et al. 2020), predicting product returns in e-commerce in India (Joshi et al. 2018), predicting product returns for a major German e-tailer (Kranz et al. 2015), predicting order returns in e-tail industry in Bangladesh (Imran & Amin, 2020), and predicting returns before purchase in fashion e-commerce (Kedia et al. 2019).

These studies all focus on their specific domain, but often they are looking at e-commerce order return prediction. Joshi et al. (2018), Kranz et al. (2015), Imran and Amin (2020), and Kedia et al. (2019) all focus on e-commerce return prediction while neglecting the in-store market. Besides, they focus on a specific country or a different domain than fashion. They use machine learning algorithms to make predictions, all with a specific focus on certain algorithms.

This research does not focus on a specific method or algorithm, it aims to predict returns with several machine learning algorithms proven as good models in previously conducted research, to find out how well they perform on the dataset at hand and compare them against each other. The dataset consists of webshop orders as well as in-store orders, so a comparison between these two order sources is made as well.

Societal relevance

Competition in the fashion industry has always been harsh, but with the rapid growth of the online market (24% growth in Western Europe in 2020 (Ambilkar et al. 2021)), consumers demand even more, and retailers and e-tailers need to fulfill these needs. An important component of these consumer demands is return leniency, especially in e-commerce. With a lenient return policy, E-commerce sales will increase (Janakiraman et al. 2016), consumer satisfaction will possibly increase, (Cassill (1998); Dissayanke & Singh (2007)), and customer loyalty (Mollenkopf et al. 2007) and perceived fairness (Pei et al. 2014) will increase. E-tailers are spending lots of money on making sure their return policy is beneficial for consumers, but without knowing which items will be returned, it is difficult to lower these costs.

Besides the cost aspect, there is another reason why it is important to be able to know which items will be returned, namely the environment. Both last-mile delivery (delivery to the final consumer) and a typical shopping trip cause CO2 emissions (Janakiraman et al. 2016; Edwards et al. 2009). Returning goods should not have been shipped to the customer in the first place, so emission for these goods is unnecessary. Research has shown that almost 25 percent of e-commerce CO2 emission is caused by returning goods (Tian & Sarkis, 2021). When an e-tailer can predict if an item will be returned by the customer, they can anticipate on aspects like costs and emissions, by adjusting transportation planning, purchase strategy, and so on.

1.3. Research Questions

The main research question of this study:

How well can product return flows for a fashion business operating in The Netherlands, Belgium, and Germany be predicted with the use of machine learning?

Sub-questions

SQ 1: How does the performance of a set of machine learning algorithms in predicting product return flows for a fashion business operating in The Netherlands, Belgium, and Germany compare?

SQ 2: How does the performance of machine learning algorithms in predicting product return flows for a fashion business operating in The Netherlands, Belgium, and Germany compare for e-commerce and in-store sales?

SQ 3: Which features are the best predictors in predicting product return flows for a fashion business operating in The Netherlands, Belgium, and Germany?

2. Literature review

Since the rise of the internet, and especially since the internet has become available to a huge part of the world population, the e-commerce market has grown extensively. With the extensive growth of the e-commerce market, competition has increased. Nowadays consumers can demand more and more since there is so much competition. Businesses want to keep existing customers and attract new ones by making sure their terms and conditions are appealing. One way to achieve this is by having a lenient return policy. According to Janakiraman et al. (2016), e-commerce sales will increase with a lenient return policy. This, however, leads to a growth in returned goods, which leads to an increase in costs and CO2 emissions, according to Tian and Sarkis (2021). For a business, it is beneficial to know if an item will be returned or not, since it can anticipate the consequences of it and reduce costs and CO2 emission.

Some research has been conducted in the field of return flows, mainly in the area of business process management. The main focus is mostly on the differences between several return process policies (Maxham, 2012), and the impact of returns (Petersen, 2010). There is, however, another research area, on which less focus has been laid so far, which is return prediction.

Return prediction can be placed in the scope of the research field of data science. Return prediction is a broad topic, but not that widely represented in research yet. Existing work on the topic of return prediction mainly focuses on e-commerce returns, and often specific regions or countries, and specific algorithms. It has been shown, however, that with the right algorithms and feature selection, it is possible to predict returns with good performance. Research about return prediction before the purchase in e-commerce, for instance, shows an area under the curve (AUC) of 0.832 with a fully connected neural network (Kedia et al. 2019).

This literature review is split up into three separate sections. One for each sub-question. So firstly, an introduction will be given about the algorithms that have been used in previous research, with their performance. Secondly, the different kinds of features that were used in previous research are reviewed. And lastly, a view will be given on the broadness of current literature in terms of selling platforms (in-store and webshop orders and returns)

To answer the research questions proposed in this paper, a dataset of a fashion business called Annadiva will be used. It consists of two parts and contains useful features to take into account for making predictions.

2.1. Model comparison

Different machine learning techniques have been used to predict returns, with different results. Each paper makes use of a specific selection of algorithms and techniques to compare.

Zhu et al. (2018) propose a method in which similar products and similar customers are clustered to increase predictive capabilities. The aim was to use historical information on customers and products to cluster them and predict customer return behavior concerning specific products. Two “acts” are distinguished; purchase without return, and purchase with return. These acts are assigned to customers and products, who are then clustered to increase return predictability. Predictions are made by finding clusters close to each other and seeing if it is a “return cluster” or “non-return cluster”. This results in a precision score of 0.820, and a recall of 0.388.

Kranz et al. (2015) suggest making use of adaptive boosting (AdaBoost) and random forest. The more than 5,500 dummy features are reduced to 10 numeric features with the use of different methods of dimensionality reduction. The model in which AdaBoost is used is performing best with a precision of 0.867 and recall of 0.069 The random forest classifier performs similar, with a precision of 0.864 and recall of 0.072. These models perform well in predicting true positives (a predicted return being an actual return), while it lacks behind in recall score.

Imran and Amin (2020) propose a comparison of different predictive supervised machine learning algorithms. One study compares several boosting algorithms and a decision tree. The decision tree model performs worst with a ROC-AUC of 0.603, while the other models all have similar predictive capabilities with ROC-AUC between 0.648 and 0.676.

Kedia et al. (2019) use the shopping cart to train the model and predict on individual items. It does so by training a model to predict the return probability score of a cart/customer combination. This probability is used in a binary classification model to predict if an individual item is returned or not. To do so, some different algorithms are used. The fully connected neural network comes out as best performing (AUC of 0.832, precision of 0.740, and recall of 0.340).

What stands out in the discussed research, is that the results solely focus on increasing the predictability of true positives. All research either uses the true positive rate or precision as evaluation metric. In the case of return prediction, predicting a true positive is not more important than predicting a true negative, since a wrong prediction (either false positive or false negative) will lead to certain responses in operation (e.g. predicting a false positive (an orderline predicted as return that is not returned) could lead to unnecessary transportation planning adjustments, and predicting a false negative (an orderline predicted as not returned that is returned) could lead to unnecessary stock replenishment). This is why F1 could be a better measure since it takes into account both precision and recall, so does not focus more on one of the two.

Neural networks, boosting algorithms, and random forest turned out to be best in predicting returns in previous research. Current literature mostly makes use of similar algorithms, so no extensive comparison of different kinds of machine learning algorithms. This paper, however, is extending on current literature by making a comparison of the performance of the algorithms used in existing studies, and does not just focus on one type of algorithm, but aims to compare the best-performing ones, to find the best-performing algorithm on the dataset at hand.

2.2. *Webshop and in-store*

Whereas research mentioned before is focused on e-commerce return prediction, little to no focus has been laid on in-store sales and returns. Four papers issuing research about predicting product return flows focus solely on e-commerce sales (Imran & Amin, 2020; Kedia et al. 2019; Kranz et al. 2015; Zhu et al. 2018), while there are differences in behavior between online and offline customers, according to Sarkar and Das (2017). This research aims to compare the difference in the predictability of returns between e-commerce and in-store sales.

Cao et al. (2020) researched whether a retail business should start selling its products online as well stating that in 2016 the return rate of products sold online was 30% and for products bought physically in a store it was 8.8%. So no research has yet been conducted on the difference between the predictability of product returns for online sales compared to in-store sales, combined with the big difference in return behavior of consumers buying online or in-store.

Current literature is not taking into account that there possibly could be a difference between the predictability of order returns for webshop orders and in-store orders. Focus is laid on just webshop orders. This research, on the other hand, does take into account in-store orders as well. An explicit comparison is made between the predictability of returns for webshop and in-store orders. This is done by splitting the dataset into webshop and in-store orders and running the same algorithms with the same features on both datasets. A comparison is then made between the performance metrics.

2.3. Feature selection

In these mentioned papers' algorithm applications, some features turned out to be more influential than others. Imran and Amin (2020) take an extensive look at the importance of all features used for predictions. They used a metric called feature importance to find out which features contribute to the prediction most. The top five consists of order location, cart size (order size), weekday, order hour, and discount. Four of these five most important features are also available in the dataset at hand.

Kedia et al. (2019) divide the features into three separate categories, namely; product-level, cart-level, and user-level features. They use similar features to the ones available in the dataset at hand, like the product category, cart size, weekday, order time, historical user orders, and the historical user returns. All research conducted makes use of similar features, where they can all be divided into these three categories. Zhu et al. (2018) make a division between, product features like color, size, category; customer attributes like past return rate; and basket information, like the size of the order. Roughly the same features are used by Kranz et al. (2015) who focus on feature extraction based on the importance of these features. They found that these mentioned features are all categorized as best predictors for return prediction within fashion.

It can be concluded that the features available in the dataset at hand are similar to the ones used in previous research, and turned out to be good predictors. Only Imran and Amin (2020) calculated feature importance for their used variables. This gives a good insight into which features contribute most to the predictions made. This research will take an extensive look at the importance of all features in predicting whether an orderline is returned or not. A comparison is made between the webshop and in-store orders as well.

In conclusion, this research aims to use data science techniques to extend and compare previously conducted studies on return prediction by focusing on e-commerce as well as in-store returns and comparing different machine learning algorithms with the use of different features. Splitting up the research into three separate parts leads to a clear overview of the difference between this study and previously conducted studies. It facilitates a framework to make clear divisions between comparisons (webshop with in-store, and the different algorithms and features), but in the end, a general conclusion can be drawn as well.

3. Methodology

In this section, the research design is explained, the data science pipeline is elaborated on, and context is given about the methods used for model training, testing, and evaluation. A visualization of the setup of the methodology of this research is added in Appendix A.

3.1. Algorithms

In current literature various Machine Learning algorithms are used to predict return flows in (e-commerce) fashion. This research will use some of these algorithms to compare them on the dataset of Annadiva and to find out which one performs best, compared to the results of previous research. The best algorithms in current research are selected and used to build models for the in-store and webshop return predictions (Imran & Amin, 2020; Kedia et al. 2019; Kranz et al. 2015). For each algorithm used in this research, an explanation is given about the results of this algorithm in the current literature.

A logistic regression model is used as the baseline in this research. The reason for this is that in conducted research, this turns out the be the worst predictor.

Kranz et al. (2015) found in their research on predicting product returns in e-commerce that for their data the AdaBoost and random forest algorithms performed best, and that logistic regression is performing worse. The features used in their research are similar to the ones available in the dataset at hand, which is why the AdaBoost and random forest algorithms will be used in this research, with a logistic regression model as the baseline.

In a study about predicting returns before purchase, Kedia et al. (2019) found that a neural network was performing best on their dataset. The features used in this research are also similar to the ones available in this study, so the neural network algorithm will be used to make a model for predicting returns.

Bahel et al. (2020) conducted a study in which they compared various binary classification algorithms on different datasets for optimal performance. Naïve bayes turned out to be a well-predicting classification algorithm, however, it depends on the dataset it is used on. So, the Naïve bayes classifier will be used to predict the binary classifications on the dataset at hand to find out if it is a good predictor in predicting returns for a fashion business.

3.2. Hyperparameter tuning

For each machine learning model used in this research, the caret package in R is used to train the model. Some of the algorithms require hyperparameters to be filled in. These hyperparameters can be tuned to an optimal value. The optimal value is based on the F1-score for these settings. Depending on the algorithm used, different hyperparameters can be tuned. See appendix B for the tables of the hyperparameter tuning results.

3.2.1. Logistic regression and Naïve bayes

Both logistic regression and naïve bayes do not have hyperparameters to tune within the caret package. So they will not be discussed further in this part.

3.2.2. Neural network

The neural network hyperparameters that are tuned by the caret package are size and decay. For size, three options are tested, 1, 3, and 5. For decay, also three options are tested, 0, 0.0001, and 0.1. The optimal hyperparameter values for both the in-store and webshop models are a size of 3 and a decay of 0.1.

3.2.3. Adaboost

Adaboost is an algorithm that uses a forest of decision trees to make predictions. The hyperparameters for the AdaBoost algorithm that are used in the caret package are the max tree depth and amount of trees. For the max depth, values of 1, 2, and 3 are tested and for mfinal 50, 100, and 150 trees. The optimal hyperparameter values for the in-store model are a max tree depth of 3 and 150 trees. For the webshop model, it is a max tree depth of 3 and 50 trees.

3.2.4. Random forest

The random forest algorithm has two hyperparameters to tune within the caret package; mtry (the number of variables to randomly sample as candidates at each split). For the split rule, "gini" and "extratrees" are tested. The optimal hyperparameter values for both the in-store and webshop models are a mtry of 32 and "gini" as splitrule.

3.3. Feature importance

To answer the third sub-question, which is about the importance of features used in the models, a feature importance analysis needs to be conducted. For this, an approach proposed by Magdalena et al. (2010) is used. They proposed a feature selection method, specifically for classification problems, using ROC curves. The idea behind their approach is that they calculate the discriminant power of a variable. The discriminant power is the difference between the discriminatory power of the model with all variables except the one for which the discriminatory power is calculated and the model with all variables. For this, the AUC is used. The discriminatory power is equal to the difference in the AUC values of the model with all variables included and the model with all variables except the one for which the discriminatory power is calculated. The method results in a table in which the features are listed with their corresponding discriminant power. This approach will be applied to the best model for in-store orderlines and the best model for webshop orderlines.

4. Experimental setup

The dataset that will be used to conduct this research is provided by a fashion business named Annadiva. The dataset consists of two parts, orders, and orderlines. So for each order, there can be several orderlines with information about the ordered products. Since an order can consist of several orderlines, the data about orders will be added to orderlines.

There are in total 161,018 orderlines, from 01-09-2021 until 01-09-2022. The dependent variable in the dataset is called “return”. There are several features available in the dataset, like discount, product price, order month, order source (webshop or in-store), and product type (color, size, clothing type). The descriptive data analysis is shown and elaborated on in Appendix C.

4.1. Data preprocessing

Some preprocessing steps have been taken to make sure the independent variable is in a good format and is complete. These will be explained briefly below.

4.1.1. Removing orderlines unable to be used

The returned quantity is the dependent variable and is, as told, a binary variable. The values are 1 for a return and 0 for not returned.

The orderlines without a known customer (so without a customer id) are removed. These cannot be taken into account, because orderlines with returns, without a customer id cannot be matched to the original order. This is only necessary for in-store orders since a new orderline is created if something is returned, and for webshop orders, the original orderline is adjusted. About 60% of in-store orders are placed by customers without an id. There is a difference in the return rate between customers with and without customer id. 2.2% of the orders placed by a customer with an id are returned, against 6.6% by customers without an id. This could be a relevant comparison for new research, but considering the scope of this research, it is impossible to take it into account.

Due to a bug in the data management system, there are products without any information about them (color, type, price, etc.). These are of no value in making predictions with ML algorithms. Orderlines with these products are removed from the dataset. For the in-store orders, 1110 of 16,829 rows are removed, and for webshop orders, 10,334 of 144,189 rows are removed.

4.1.2. Adjusting and adding variables

Some categorical variables are not binary (either yes or no). For a model to work properly with such variables, they need to be transitioned into dummy variables. A dummy variable is simply an additional feature of either 0 (no) or 1 (yes) for every category in a categorical variable minus 1 since the last category is automatically in place if none of the others is. So for example colors; black, white, and red. When an order is placed for a black product, column black will get a value of 1 and white value of 0 for that row, when an item is red, black and white will get a value of 0. However, since an extensive look will be taken at the importance of all features, all categories must get a corresponding dummy to compare all variables’ importance. Some of the categorical variables have too many categories, like the brand name (134) or series name (937), they are left out, since making dummies is not desirable because the dataset would otherwise get too large. Some features have been added as well:

- Historical amount of orders placed by a customer
- Historical return ratio by a customer
- Size of the order (total amount of different products)

4.1.3. *Webshop, in-store split*

To be able to answer the third sub-question, a division has to be made between webshop orders and in-store orders. This is done by splitting the dataset into two separate datasets, one for webshop orders and one for in-store orders.

4.1.4. *Class imbalance*

The dependent variable (return yes/no) is imbalanced. There are more orderlines without returns than with returns. To make accurate predictions it is important to make sure both classes are evenly represented in the dataset. The imbalance ratio (IR) is used to determine the class imbalance of the dependent variable. IR divides the majority class by the minority class, so if classes are perfectly balanced, IR is 1. The higher it is, the more imbalanced the data is. Figures 1 and 2 show the division of returns and non-returns for both the in-store and webshop orderlines respectively.

For orders placed in-store, there are 15,777 orderlines, with 293 returns (1.9%), so IR = (15,777-293) / 293 = 52.8. The SMOTE method (synthetic minority oversampling technique) is used to make the classes balanced. It is proposed by Bowyer et al. (2002). SMOTE performs data augmentation, which randomly adds small noise to an oversampled instance so that it is not the same as the original instance. In this way, the risk of overfitting is reduced.

For orders placed in the webshop, there are 133,708 orderlines, with 56,889 returns (42.5%), so IR = (133,708 - 56,889) / 56,889 = 1.4. Initially, the same strategy of oversampling was used for webshop orders, to retain consistency. However, when running models with such a large dataset, the computational power of the device is not sufficient. So undersampling is used to reduce the total amount of orderlines to the same size as in-store orderlines.

Figure 1: Class imbalance in-store orderlines

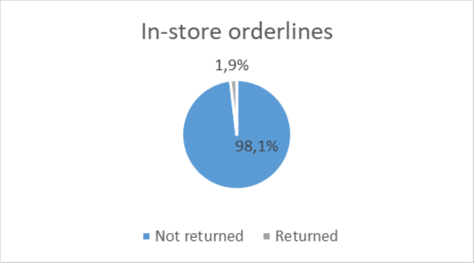
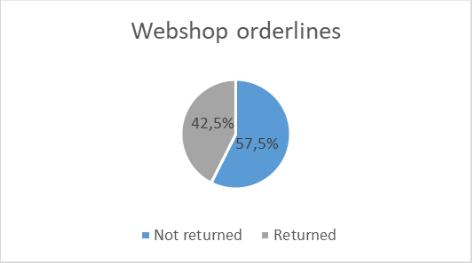


Figure 2: Class imbalance webshop orderlines



4.1.5. *Train-test split*

Firstly both datasets (in-store and webshop) are split into a train and test set, 70% is train data and 30% is test data. Then the models will be fitted on the train set with the usage of k-fold cross-validation, with five folds, due to limitations in the computational power. This technique is used since it repeats the train-test split multiple times and thus reduces the risk of overfitting.

Splitting the data in train and test sets combined with k-fold cross-validation makes sure the results can be better generalized. The risk of overfitting is reduced since the test set is not taken into account when training the model and within cross-validation, one sample is left out as a test set each fold, so there is always a test set that is not used in training the model.

4.1.6. *Features*

The dependent variable in this research is binary(a return yes/no), so binary classification algorithms will be used to predict whether an orderline is returned or not. Features used in making predictions can be separated into three different categories; Customer-related features are historical customer orders, historical customer order return ratio; Order-related features are; order discount, total invoice amount, shipping fee, payment fee, month, time of the day (morning, afternoon, evening/night); Product-related features are product discount, price, color, type, size,

4.2. Software

This research will make use of the programming language R, making use of several data pre-processing/cleaning and machine learning libraries, like tidyverse and caret.

4.3. Evaluation method

Since the predictions made in this research are binary classifications, binary evaluation methods need to be used. Per orderline, a prediction will be made if the product will be returned or not. A true positive in this case is a predicted return, which is returned. While a true negative is predicted to not be returned and is not returned. For comparison of the algorithms, several evaluation metrics will be used.

Accuracy is used as a general metric, it does not show full insights into the performance of the model, so it needs to be accompanied by other metrics as well.

Both precision and recall focus on their specific part of the confusion matrix. Dependent on what is more important, reducing false positives (FP) or false negatives (FN), it is desirable to increase precision and/or recall closer to 1.

F1-score is kind of a mean of precision and recall. If neither precision nor recall is the more important metric F1-score can be used to determine the harmonic mean of the two. F1-score is the best metric to use when data is imbalanced and therefore resampled since it takes into account both precision and recall. Besides, neither optimizing precision nor recall is more important than the other in this study, therefore F1-score is seen as the leading metric in this research but still compared to other metrics for the sake of completeness.

The ROC (receiver operator characteristic) curve gives a plot as output, where the true positive rate is plotted against the false positive rate. The AUC (area under the curve) is used to summarize the ROC curve, AUC gives as output a measurement of the ability of a classifier to make a distinguishment between classes.

5. Results

In this section, the results are presented and elaborated on. To answer the first sub-question, a division is made between the results of all machine learning models' performance. The second sub-question is answered by making a comparison between the webshop and in-store models. The third sub-question is answered with a feature importance analysis of all features used in the models. This is done only for the best-performing model.

The results are displayed per model and within the model, a division is made between in-store performance and webshop performance. To summarize the metric performance across all folds of the k-fold cross-validation, the mean and standard deviation of the performance across these folds are given for all metrics on the in-store and webshop train models. Besides the mean and standard deviation (SD), the metric performances of the best model across the folds are given in columns store train and web train. The performance of these best models on the test set is given in columns store test and web test. For all algorithms, the hyperparameters are tuned. The best hyperparameter settings are used to make predictions on the test set.

5.1. Logistic regression

Table 1 shows the performance of the logistic regression model (baseline) per dataset (train and test for in-store and webshop), per metric. The model is performing better on the in-store data than on the webshop data with a difference in test F1-score of 0.2617. Performance is consistent for train and test sets, with minimal differences in metric scores, which means the model is not overfitting. The mean and SD of the metric scores across the folds of the k-fold cross-validation are displayed in Table 1 as well. All metric means are constant for both the in-store and webshop models, with SD's not higher than 0.0029 for the F1 score.

Table 1: Performance metrics logistic regression

Metric	store train	store train mean	store train sd	store test	web train	web train mean	web train sd	web test
Accuracy	0.9862	0.9686	0.0235	0.9868	0.7346	0.7258	0.0117	0.7299
Precision	0.9999	0.9822	0.0118	0.9996	0.7381	0.7221	0.0107	0.7312
Recall	0.9724	0.9555	0.0225	0.9743	0.7306	0.7137	0.0113	0.7188
F1	0.9860	0.9827	0.0023	0.9867	0.7343	0.7321	0.0029	0.7250
AUC	0.9869	0.9824	0.0030	0.9859	0.7693	0.7627	0.0044	0.7687

5.2. Neural Network

The neural network model performances, based on the optimal hyperparameter settings, are shown in Table 2. This algorithm performs better on the in-store model as well, with a difference in test F1-score of 0.2228, compared to the webshop model. The in-store model performs slightly worse compared to the logistic regression with a difference in test F1-scores of 0.008. The webshop model performs better with a difference in test F1-score 0.0384 compared to the logistic regression model. The performance on train and test sets are similar, with a minimal difference, meaning that the neural network model is not overfitting.

Table 2: Performance metrics neural network

Metric	store train	store train mean	store train sd	store test	web train	web train mean	web train sd	web test
Accuracy	0.9849	0.9669	0.0240	0.9843	0.7465	0.7396	0.0092	0.7483
Precision	0.9979	0.9910	0.0092	0.9976	0.7194	0.7147	0.0063	0.7173
Recall	0.9717	0.9632	0.0113	0.9713	0.8117	0.8059	0.0077	0.8116
F1	0.9846	0.9702	0.0115	0.9843	0.7628	0.7561	0.0068	0.7615
AUC	0.9782	0.9587	0.0260	0.9782	0.7529	0.7444	0.0113	0.7487

5.3. Random forest

The random forest model performances, based on the optimal hyperparameter settings, are shown in Table 3. The random forest algorithm also performs better on the in-store model compared to the webshop model, with a difference in test F1-score of 0.2301. The in-store model performs best compared to all other models, with an F1-score of 0.9899. Compared to the in-store logistic regression baseline, it performs better with a difference in test F1-score of 0.0032. The random forest webshop model is also outperforming the logistic regression baseline, with a difference in test F1-score of 0.0348.

Table 3: Performance metrics random forest

Metrics	store train	store train mean	store train sd	store test	web train	web train mean	web train sd	web test
Accuracy	0.9999	0.9922	0.0103	0.9898	0.7350	0.7294	0.0075	0.7547
Precision	0.9998	0.9920	0.0104	0.9958	0.7222	0.7042	0.0240	0.7375
Recall	1	0.9821	0.0239	0.9841	0.7673	0.7478	0.0130	0.7834
F1	0.9999	0.9886	0.0017	0.9899	0.7441	0.7367	0.0077	0.7598
AUC	0.9969	0.9919	0.0033	0.9898	0.8253	0.8077	0.0117	0.7556

5.4. AdaBoost

The AdaBoost model performances, based on the optimal hyperparameter settings, are shown in Table 4. The F1-score performance of the AdaBoost in-store model is better than the AdaBoost webshop model, with a difference of 0.2425. The performance of the AdaBoost F1-score is better for both the in-store model and the webshop model, compared to the logistic regression baseline. The in-store model performs better with an F1-score difference of 0.0029. The webshop model performs better with a difference in F1-score of 0.0221

Table 4: Performance metrics AdaBoost

Metric	store train	store train mean	store train sd	store test	web train	web train mean	web train sd	web test
Accuracy	0.9905	0.9858	0.0063	0.9895	0.7566	0.7476	0.0060	0.7505
Precision	0.9983	0.9786	0.0131	0.9962	0.7596	0.7416	0.0240	0.7499
Recall	0.9826	0.9730	0.0064	0.9830	0.7536	0.7480	0.0037	0.7443
F1	0.9904	0.9881	0.0029	0.9896	0.7566	0.7441	0.0069	0.7471
AUC	0.9905	0.9815	0.0120	0.9897	0.7618	0.7518	0.0067	0.7572

5.5. Naïve Bayes

The worst model of all classification models used is the naïve bayes model. As can be seen in Table 5, the test F1-scores are 0.8682 and 0.6826 for the in-store and webshop models respectively. Compared to the logistic regression baseline, the in-store model performs worse with a difference of 0.1185 in test F1-score. The webshop model performs worse with a difference of 0.0424 in test F1-score. Again, the performance of the in-store model is better than the webshop model performance. The difference in F1-score here is 0.1856.

Table 5: Performance metrics naïve bayes

Metric	store train	store train mean	store train sd	store test	web train	web train mean	web train sd	web test
Accuracy	0.8635	0.8467	0.0112	0.8553	0.6092	0.6014	0.0052	0.6061
Precision	0.8090	0.7942	0.0099	0.8053	0.5738	0.5573	0.0110	0.5678
Recall	0.9498	0.9420	0.0104	0.9418	0.8612	0.8567	0.0060	0.8555
F1	0.8737	0.8457	0.0043	0.8682	0.6887	0.6799	0.0025	0.6826
AUC	0.8745	0.8698	0.0063	0.8665	0.6456	0.6408	0.0064	0.6431

5.6. Algorithm comparison

This section compares the predictive performance of the algorithms mentioned in section 3.1 for both the in-store and webshop models. The best-performing algorithm differs for in-store and webshop models. When comparing all in-store models, the random forest algorithm turns out to be the best predictor on the test set, with an F1-score of 0.9899. When comparing all webshop models, the neural network model is the best predictor on the test set, with an F1-score of 0.7615. Table 6 shows the performance of all models on the test set. The logistic regression, neural network, random forest, and AdaBoost in-store models' F1-score test performances are similar, all within a range of 0.0056. The naïve bayes in-store model is the weakest model with an F1-score of 0.1161 below the second weakest model, namely the neural network.

For the webshop models, the difference in the F1-score is larger. The three models with the highest F1-scores are performing within a range of 0.0144. The naïve bayes model is again showing the worst F1-score on the test set, with a score of 0.0424 below the second weakest model, which is the logistic regression baseline in this case. Within the four best models, the logistic regression is doing worse compared to the in-store model, with an F1-score of 0.7250, which is 0.0221 below the third best predictor, namely the AdaBoost model.

Table 6: F1-score performance of all models

Model	Test F1-score in-store	Test F1-score webshop
Logistic regression	0.9867	0.7250
Neural network	0.9843	0.7615
Random forest	0.9899	0.7598
Adaboost	0.9896	0.7471
Naïve bayes	0.8682	0.6826

5.7. In-store and webshop comparison

In this section, a comparison will be made to the predictability of returns for in-store and webshop orderlines. The comparison is made with a statistical test on the performance of all folds of the k-fold cross-validation of all models. So for in-store and webshop orderlines, the performance of all individual folds of all five algorithms is used in a paired samples t-test. In total, both samples have a size of 25, since each of the five models has five folds. The mean of F1-scores for in-store models folds is 0.9638 and for the webshop models it is 0.7341.

- Null hypothesis: “Mean F1-score is the same for the model performances of the in-store and webshop orderlines”.
- Alternative hypothesis: “Mean F1-score is higher for the model performance of the webshop orderlines compared to the in-store orderlines”.

The paired samples t-test (t(24) = 41.40, p <0.0001) shows that there is a significant difference in the mean F1-score between the model performance of the in-store models compared to the webshop models. So it can be concluded that the predictability of returns for in-store orderlines is significantly higher than that of webshop orderlines.

5.8. Error analysis

In this section, an error analysis will be conducted on the two best-performing models. The two best-performing models are random forest (in-store) and neural network (webshop). In this error analysis, the confusion matrices will be analyzed.

Table 7 shows the confusion matrix for the random forest in-store model. It shows that the performance of the model is very good with just 96 false predictions. Of these 96, just 20 are false positives, and 76 are false negatives. This shows that the model performs better in predicting a non-returned orderline than predicting a returned orderline, resulting in a higher precision (0.9958) compared to the recall (0.9841).

Table 7: Confusion matrix random forest in-store

		Actual	
		Yes	No
Predicted	Yes	4701	20
	No	76	4641

Table 8 shows the confusion matrix of the best-performing webshop model. Namely, the neural network model. The webshop model performs worse, with 2,328 false predictions. Of these 2,328, 1465 are false positive, and 863 are false negative. This model predicts false positives more often than false negatives, meaning that the precision (0.7173) is worse compared to the recall (0.8116).

Table 8: Confusion matrix neural network webshop

		Actual	
		Yes	No
Predicted	Yes	3717	1465
	No	863	3205

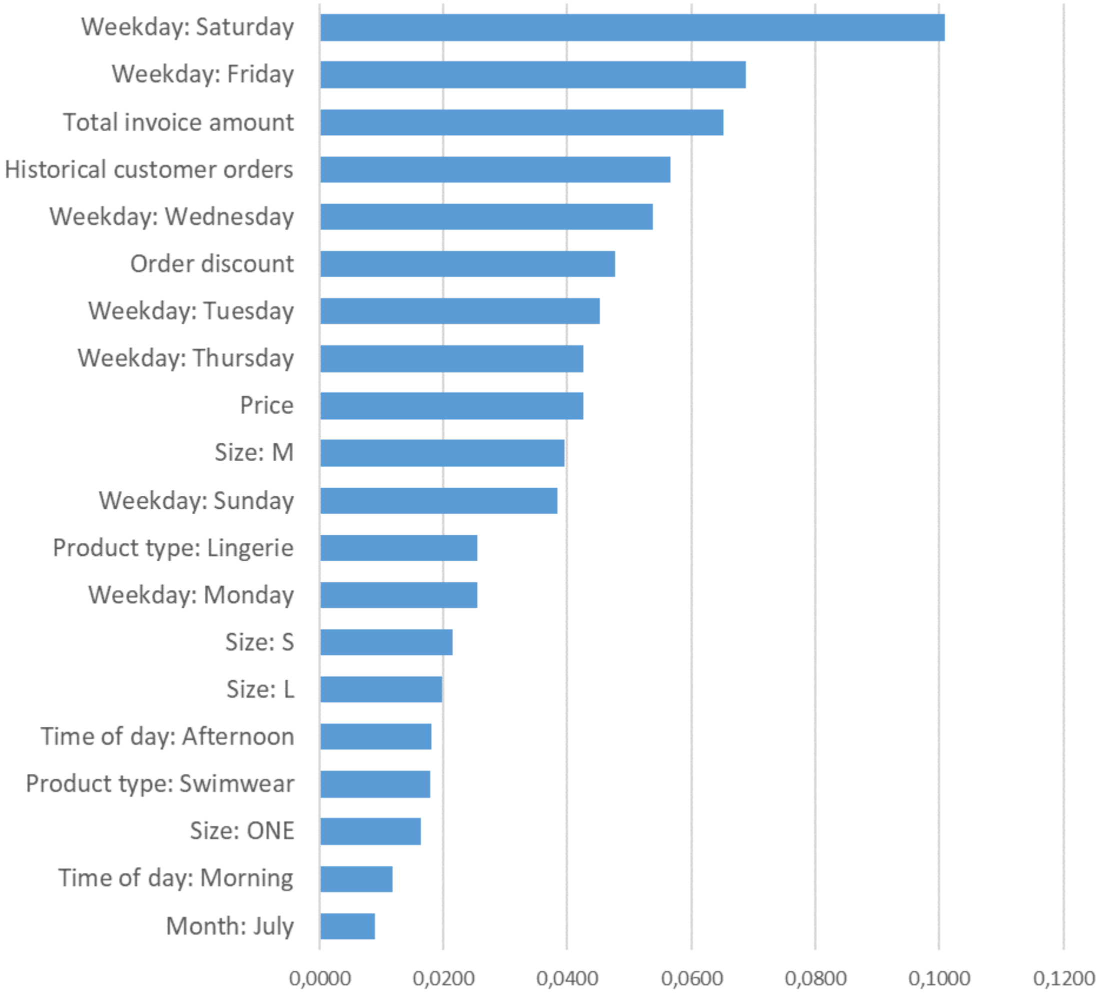
5.9. *Feature importance*

In this section, the difference in feature importance for predicting returns between in-store and webshop orderlines is compared. For this comparison, the best predicting algorithms are used to determine the in-store and webshop models to compare. The best-performing in-store model is the random forest, and the best-performing webshop model is the neural network. So the feature importance of these two models will be compared, as opposed in section 3.3. The 20 features with the highest feature importance are visualized for both models.

Figure 3 shows the feature importance of the random forest in-store model. Figure 4 shows the feature importance of the neural network webshop model. Something notable is the fact that only six variables are in the top 20 features with the highest importance of both models. So the features that contribute to the model most, are mostly different for the two models.

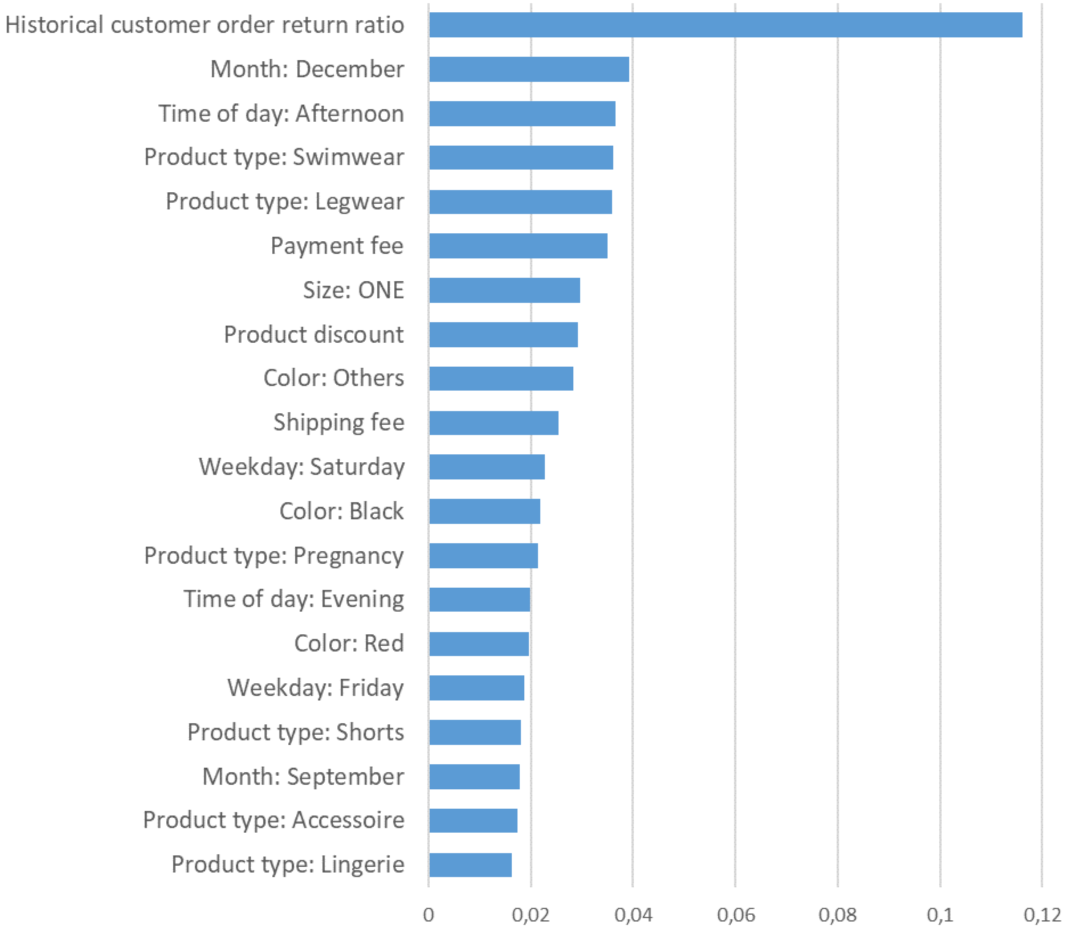
When looking at the feature importance of the random forest in-store model, there is one predictor that stands out in terms of feature importance, the weekday Saturday. With a discriminant power score of 0.1008, it is the feature with the highest importance. What stands out is that all 7 weekday dummies are in the top 13 features with the highest importance.

Figure 3: Feature importance random forest in-store model



The feature with the highest importance for the neural network webshop model, the historical customer order return ratio, has a discriminant power that is almost three times as large as the discriminant power of the next feature, which is the month of December. When compared to the feature importance of the random forest in-store model, the feature importance of the neural network webshop model is less spread across all variables.

Figure 4: Feature importance neural network webshop model



6. Discussion

The research goal of this study was to find out how well product return flows for a fashion business can be predicted with the use of machine learning. To answer this question, a dataset of Annadiva, a fashion business operating in The Netherlands, Belgium, and Germany, with physical stores and a webshop, was used. Information about orders, products, and customers was used to make predictions. The problem this research addressed and aimed to solve was a binary classification problem. The main question was split into three sub-research questions to give the research structure.

The three sub-questions implied a comparison of the predictive ability of different machine learning algorithms, the difference in predictive ability between in-store and webshop sales, and thirdly what the best features are to predict.

6.1. Results discussion

When looking at the main question, it can be concluded that it is possible to predict return flows for a fashion business, given the data that was used. It is however dependent on the sales channel (in-store or webshop). For the comparison of the predictive performance of the different machine learning algorithms, the F1-score was used. The best machine learning algorithm differed for the in-store and webshop datasets.

6.1.1. Algorithm comparison

For the in-store orderlines dataset, the best model was the random forest, which had an F1-score of 0.9889 and an accuracy of 0.9898. In other words, an accuracy this high means that in almost 99% of the cases, the algorithm can predict correctly if an orderline is returned or not. However, the difference between four (logistic regression, neural network, AdaBoost, and random forest) of the total five algorithms' F1-scores on the test datasets are within a range of 0.0056. Only the naïve bayes model had a considerably lower F1-score, namely 0.8682. All models' train and test F1-scores were very close with differences not exceeding 0.01, so all models are robust and can be generalized. When looking at the existing literature on the topic of return prediction, the high degree of consistency for the three non-baseline models is logical, since they were the best predictors in several papers. Kedia et al. (2019) found the neural network to be the best predictor. Kranz et al. (2015) found AdaBoost and random forest as the best predictors. The high performance of the logistic regression baseline was not expected but could be explained by the variable behavior. It could be that the features used to predict are linear, resulting in the fact that a logistic regression model can recognize all correlations in the dataset. This high performance leads to the idea that a machine learning algorithm of higher complexity is not necessary in this situation. The lower performance of the naïve bayes classifier could be explained by the study of Bahel et al. (2020). They conducted a study in which they compared various binary classification algorithms on different datasets. They found that the naïve bayes classifier can be a good predictor, but it depends on the dataset at hand. In general, the high performance on the in-store orderlines data is something that stands out, it was expected that the predictability would not differ from webshop orderlines data as much as it did. The classification problem for in-store orders is not as difficult as expected. Reasons for this could be that there are clear patterns in the behavior of the in-store consumer, that can be well recognized by some machine learning algorithms.

For the webshop orderlines dataset, the best model was the neural network, which had an F1-score of 0.7615 and an accuracy of 0.7483. This means that in almost 75% of the cases, the algorithm can predict correctly if an orderline is returned or not. The difference in the performance of the algorithms' F1-scores is larger for the webshop models in comparison to the in-store models. The range of F1-scores of the three best models (neural network, AdaBoost, random forest) is 0.0144, and the naïve bayes is scoring considerably lower again with an F1-score of 0.6826. When compared to the neural network, AdaBoost, and random forest models, the logistic regression is performing worse for the webshop orders than it did for the in-store orders. For the webshop models, the same holds as for the in-store models, namely, the train and test F1-scores were very close with differences not exceeding 0.01, so all models are robust and can be generalized. As explained for the in-store model performances, it is logical that the neural network, random forest, and AdaBoost perform best, since they were the best predictor in previously conducted studies. The logistic regression is doing a little worse when compared to the in-store data, which could be explained by the fact that the importance of the features used in this model is different compared to the in-store model. For comparison with previously conducted research, precision and recall are used, since those are the only two metrics that all papers use. The neural network model in the study of Kedia et al. (2019) performed with a precision of 0.740 and recall of 0.340, where the precision and recall of the neural network webshop model in this research were 0.7173 and 0.8116 respectively. The AdaBoost and random forest algorithms Kranz et al. (2015) used performed roughly the same with a precision of around 0.846 and recall of around 0.143. The precision and recall for the AdaBoost model in this research were respectively 0.7499 and 0.7443, and for the random forest model, they were 0.7375 and 0.7598 respectively. In this study, the performance on the precision score is lower than in both other mentioned studies, however, recall performance is a lot higher, since this research focuses on optimizing the F1-score, which takes into account both precision and recall. Other papers mainly aim to maximize the true positive class, while that is not more desirable from a scientific or societal perspective. This paper does take into account both of them.

6.1.2. In-store & webshop comparison

This research contributes to current literature with a comparison between in-store and webshop return prediction. To compare the total in-store model performance with the webshop performance, a paired samples t-test was conducted. It was done by using the F1-score of all folds of the k-fold cross-validation for both in-store and webshop data as paired samples. The mean F1-score for the in-store models' folds was 0.9638 and for the webshop models, it was 0.7341. The result was that the in-store F1-score is significantly higher than that of the webshop models. The in-store models' predictive ability is a lot higher than the webshop models' predictive ability. The physical consumers' behavior can be better predicted than that of an online consumer. This could be explained by looking at the feature importance of the different models. The features with large impacts differ for in-store and webshop models, which could explain the difference in performance. Besides, webshop customers return more in general, which could make it less predictable. It could be the case that the returned in-store orders are much more alike (in terms of feature values) than the returned webshop orders. When looking back at the literature review, it was found that there were four papers (Imran & Amin, 2020; Kedia et al. 2019; Kranz et al. 2015; Zhu et al. 2018) focusing solely on webshop returns and ignoring in-store returns, while the comparison is relevant from a scientifical perspective and societal point of view.

6.1.3. Feature importance

The feature importance is analyzed for the best-performing model for in-store data and webshop data. For the in-store data, it stands out that the weekday of the sale is an important feature since all 7 weekday dummies occur in the top 15 features with the highest importance, compared to only 2 weekday dummies in the top 20 of the webshop model. Besides, only six features occur in both top 20 features with the highest importance. Since an order return is a human decision, predicting a return is predicting human behavior directly. This makes it interesting to think about reasons for differences in performance. They probably mostly are related to the difference in behavior between in-store customers and webshop customers, like the fact that someone in a store sees the product already and knows better whether they will keep it or not. Lantz and Hjort (2013) found that when the return policy is lenient, webshop customers will order more frequently, and increased probability of return, meaning that webshop customers care less if an item is right for them, since they can return it anyway. This could be a reason for more randomness in the webshop return data, which could lead to the model performing worse.

6.2. Limitations and future research

In this part limitations of this study and improvements toward future work are discussed. The computational power of the device with which this research was conducted was not sufficient enough to work with all data that was available. This resulted in the fact that part of the webshop orders had to be left out through undersampling. This is not ideal since part of the information in the data is lost, which could have been relevant. For future work, it is suggested to make use of a computational device that is sufficient enough to deal with datasets of these sizes.

A substantial part of the in-store data in this study had to be left out since it was not usable because the returns without customer id could not be matched with the original order. This is something to take into account when researching this topic. A complete dataset with all returns having a corresponding original order is important since there could be differences in return behavior between registered and non-registered customers (in this study 2.2% of the orders placed by registered customers were returned, and 6.6% of the orders played by non-registered customers were returned).

The data used in this research is from a fashion business operating in The Netherlands, Germany, and Belgium. However, the in-store data all comes from stores located in The Netherlands, since there are no stores in other countries. Webshop data comes from all three different countries. There is a possibility that the difference in the predictability of returns between in-store and webshop orders is caused by the fact that the in-store orders mainly consist of Dutch consumers and the webshop orders consist of Dutch, German, and Belgian consumers.

A post-hoc analysis has been conducted to check whether there is a difference in the predictability of the webshop returns between the three countries. For this analysis, the neural network model is used, since this was the best-performing model for webshop orders. As can be seen in table 9, the F1-score differs per country, so the return predictability is different for orders from these different countries. For future research, it is recommended to look further into these differences and possibly find out the cause of them, to make sure the geographical information does not influence the predictions without knowing it.

Table 9: Post-hoc webshop neural network

	Train F1-score	Test F1-score
The Netherlands	0.7230	0.7173
Germany	0.5790	0.6000
Belgium	0.8377	0.8463

The error analysis for the best in-store model shows that it has more difficulties in predicting the minority class (a return) compared to the majority class (no return) since there are more false negatives than false positives. In this study, the method used to reduce the imbalance between the majority and minority classes was SMOTE, but it could be that a larger or more balanced dataset would have been better for the sake of even errors for the minority and majority classes.

This study has broadened the scope of what has been researched on the topic of return prediction before. The focus was solely on e-commerce, while, according to Young (2022), consumers still buy more than 80% of their purchases in physical stores. So for the topic of return prediction, it is important to keep a broader scope and narrow it down throughout the study, to make sure all relevant information and differences can be taken into account. Besides, focusing on just one country can be interesting, but the post-hoc analysis conducted in this discussion shows that there are differences in return predictability, and thus possibly return behavior, between countries. Results based on one country cannot be generalized to other countries, this is something to take into account when conducting new research. Another important aspect to consider is the use of adequate performance metrics. Is it more important to accurately predict a return in favor of a non-returned order or vice versa? if not, a metric should be used that takes into account both the return and non-return, like the F1-score.

For society, and especially fashion businesses like Annadiva, this study can help with the implementation of new business strategies that help reduce costs and CO2 emissions. When a business can make accurate predictions on whether an ordered item will be returned or not, it can anticipate in various ways. It could for instance improve its inventory management, packaging strategies, route planning, and communication to customers with for example warning messages about high return probabilities.

7. Conclusion

The main research question of this study is: "How well can product return flows for a fashion business operating in The Netherlands, Belgium, and Germany be predicted with the use of machine learning?" In answering this question, this study contributes to the current body of research in the field of order return prediction. It does so by answering three sub-questions.

Firstly, it is found that several machine learning algorithms make good predictions about order returns. For in-store orders, the random forest algorithm performed best, and for webshop orders, the neural network algorithm performed best. However, the differences between the models were very small, especially for in-store orders, so a logistic regression or AdaBoost are also suitable models for this problem. Using sophisticated machine learning algorithms does not seem to be necessary for this research problem.

The predictability of in-store returns is significantly higher than that of webshop returns. Whether an in-store order is returned or not is predicted better than whether a webshop order is returned or not, with an accuracy that is 24% higher.

Several features influence the final predictions of the machine learning models. The more important features differ for in-store and webshop orders, with only six features being in the top 20 features of both in-store and webshop orders.

Altogether, it can be concluded that it is possible to accurately predict order returns for a fashion business with the use of machine learning. Depending on the type of sales channel and the available features to make predictions.

Acknowledgments

A word of thanks and appreciation goes out to the granter of the dataset used in this research, namely Annadiva.

References

Abbasi, A., Albrecht, C., Vance, A., & and Hansen, J. (2012). MetaFraud: A Meta-Learning Framework for. *MIS Quarterly, 36*(4), 1293-1327. doi:10.2307/41703508

Abbasi, A., Zhang, Z., Zimbra, D., Chen, H., and Nunamaker, J., & F, J. (2010). Detecting Fake Websites: The Contribution of Statistical Learning Theory. *MIS Quarterly, 34*(3), 435-461. doi:10.2307/25750686

Ambilkar, P., Dohale, V., Gunasekaran, A., & Bilolikar, V. (2021). Product returns management: a comprehensive review and future research. *International Journal of Production Research, 60*(12), 3920-3944. doi:10.1080/00207543.2021.1933645

Bahel, V., Pillai, S., & Malhorta, M. (2020). A Comparative Study on Various Binary Classification Algorithms and their Improved Variant for Optimal Performance. *Institute of Electrical and Electronics Engineers*. doi:10.1109/TENSYMP50017.2020.9230877

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems, 50*(3), 602-613. doi:10.1016/j.dss.2010.08.008

Bowyer, K. W., Chawla, N. V., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research, 16*(1), 321-357. doi:10.1613/jair.953

Cao, K., Xu, Y., Cao, J., Xu, B., & Wang, J. (2020). Whether a retailer should enter an e-commerce platform. *International Transactions in Operational Research, 27*(6), 2878-2898. doi:10.1111/itor.12768

Cassill, N. L. (1998). Do customer returns enhance product and shopping experience satisfaction? *The International Review of Retail Distribution and Consumer Research, 8*(1), 1-13. doi:10.1080/095939698342841

Choi, T. M., Yu, Y., & Au, K. F. (2011). A hybrid SARIMA wavelet transform method for sales forecasting. *Decision Support Systems, 51*(1), 130-140. doi:10.1016/j.dss.2010.12.002

Cui, H., Rajagopalan, S., & Ward, A. R. (2020). Predicting product return volume using machine learning methods. *European Journal of Operational Research, 281*(3), 612-627. doi:10.1016/j.ejor.2019.05.046

Dissayanke, D., & Singh, M. (2007). Managing Returns in E-Business. *Journal of Internet Commerce, 6*(2), 35-49. doi:10.1300/J179v06n02_04

Edwards, J. E., McKinnon, A. C., & Cullinane, S. L. (2009). Comparative analysis of the carbon footprints of conventional and online retailing. *Logistics Research Centre, 40*(1-2), 103-123. doi:10.1108/09600031011018055

Imran, A., & Amin, M. (2020). Predicting the Return of Orders in the E-Tail Industry Accompanying with Model Interpretation. *Procedia Computer Science, 176*(1), 1170-1179. doi:10.1016/j.procs.2020.09.113

Janakiraman, N., Syrdal, H., & Freling, R. (2016). The Effect of Return Policy Leniency on Consumer Purchase and Return Decisions: A Meta-analytic Review. *Journal of Retailing, 92*(2), 226-235. doi:10.1016/j.jretai.2015.11.002

Joshi, T., Mukherjee, A., & Ippadi, G. (2018). One Size Does Not Fit All: Predicting Product Returns in E-Commerce Platforms. *Institute of Electrical and Electronics Engineering*. doi:10.1109/ASONAM.2018.8508486

Kedia, S., Madan, M., & Borar, S. (2019). *Early Bird Catches the Worm: Predicting Returns Even.* doi:10.48550/arXiv.1906.12128

Kranz, J., Urbanke, P., & Kolbe, L. (2015). Predicting Product Returns in E-Commerce: The Contribution of Mahalanobis. *International Conference on Information Systems*. Retrieved from https://www.researchgate.net/publication/283271154_Predicting_Product_Returns_in_E-Commerce_The_Contribution_of_Mahalanobis_Feature_Extraction_Completed_Research_Paper

Lantz, B., & Hjort, K. (2013). Real e-customer behavioral responses to free delivery. *Electronic Commerce Research, 13*, 183-198. doi: 10.1007/s10660-013-9125-0

Magdalena, R., Martín, J. D., Soria, E., Serrano, A. J., & Gómez, J. (2010). Feature selection using ROC curves on classification problems. *International Joint Conference on Neural Networks*. doi:10.1109/IJCNN.2010.5596692

Maxham, J. G. (2012). Return Shipping Policies of Online Retailers: Normative Assumptions and the Long-Term Consequences of Fee and Free Returns. *Journal of Marketing, 76*(5). doi:10.1509/jm.10.0419

Mollenkopf, D. A., Rabinovich, E., Laseter, T., & Boyer, K. (2007). Managing internet product returns: a focus on effective service operations. *Journal of Operations Management, 38*(2), 215-250. doi:10.1111/j.1540-5915.2007.00157.x

Pei, Z., Paswan, A., & Yan, R. (2014). E-tailer's return policy, consumer's perception of return policy fairness. *Journal of Retailing and Consumer Services, 21*(3), 249-257. doi:10.1016/j.jretconser.2014.01.004

Petersen, J. A., & Kumar, V. (2010). Can Product Returns Make You Money? *MIT Sloan, 51*(3). Retrieved from https://apprissretail.com/wp-content/uploads/sites/4/2017/02/Can-Returns-Make-You-Money_White-Paper.pdf

Sarkar, R., & Das, S. (2017). Online Shopping vs Offline Shopping : A Comparative Study. *International Journal of Scientific Research in Science and Technology, 3*(1), 2395-6011. Retrieved from https://d1wqtxts1xzle7.cloudfront.net/63954907/Publishedarticle620200718-105382-7b6ukj-with-cover-page-v2.pdf?Expires=1669838071&Signature=KMcDiNoH2CQMcBdUc5CHSMHcml1CuFhRT0~V7eVNKCpiYxUDgoRQMuvpEBXhMUgJZVfr7BeUzR4Apmim3hn4X47Kx~1N5ClU8Y4bSsrZicwtP18Xvoqk

Tian, X., & Sarkis, J. (2021). Emission burden concerns for online shopping. *Joint Institute of Inclusive and Sustainable Industrial Development, 21*(1), 2-3. doi:10.1038/s41558-021-01246-9

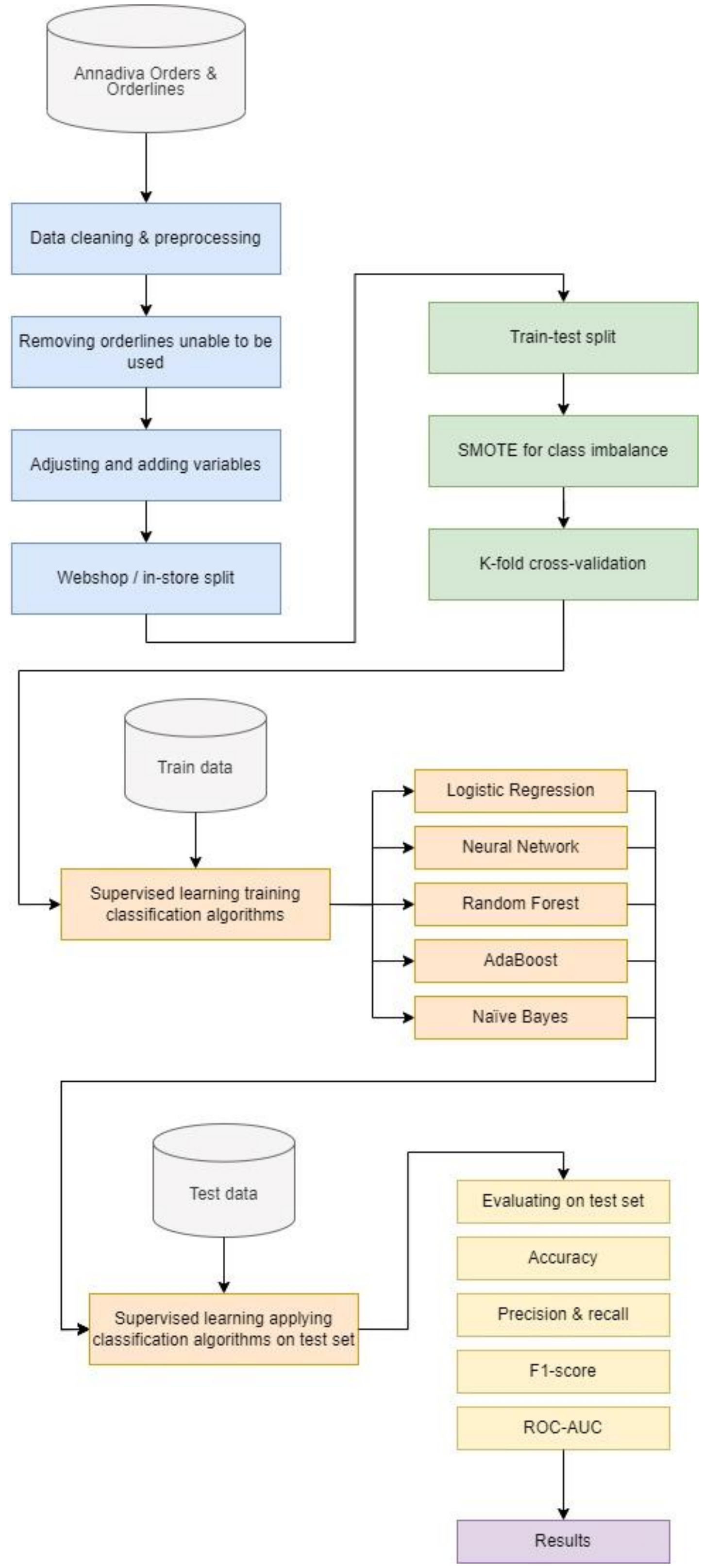
Young, J. (2022, March 10). *A decade in review: Ecommerce sales vs. total retail sales 2012-2021*. Retrieved December 1, 2022, from https://www.digitalcommerce360.com/article/e-commerce-sales-retail-sales-ten-year-review/#:~:text=This%20means%20ecommerce%20now%20accounts,significantly%20from%2015.5%25%20in%202019.

Zhang, D., Zhou, X., Leung, S. C., & and Zheng, J. (2010). Vertical bagging decision trees model for credit scoring. *Expert Systems with Applications, 37*(12), 7838-7843. doi:10.1016/j.eswa.2010.04.054

Zhu, Y., Li, J., He, J., Quanz, B. L., & Deshpande, A. A. (2018). A Local Algorithm for Product Return Prediction in E-Commerce. *IBM Research*, 3718–3724. doi:10.24963/ijcai.2018/517

Appendix A: Methodology workflow visualization

Figure 1: Methodology workflow visualization


Annadiva Orders & Orderlines
Data cleaning & preprocessing
Removing orderlines unable to be used
Adjusting and adding variables
Webshop / in-store split
Train-test split
SMOTE for class imbalance
K-fold cross-validation
Train data
Supervised learning training classification algorithms
Logistic Regression
Neural Network
Random Forest
AdaBoost
Naïve Bayes
Test data
Supervised learning applying classification algorithms on test set
Evaluating on test set
Accuracy
Precision & recall
F1-score
ROC-AUC
Results

Appendix B: Descriptive data analytics

Exploratory data analysis is conducted on some of the variables to give a view of what the dataset looks like. Figure 1 shows the return rate per product type. The types of accessories and legwear both have a much lower rate of return than the rest of the types. For categories, this could be explained by the fact that they often do not have sizes, so accessories usually fit. Figure 2 shows the return rate for every weekday. There is a dip in return rate for Friday and Saturday, all other days are relatively consistent in their return rates.

Figure 1: Return rate per product type

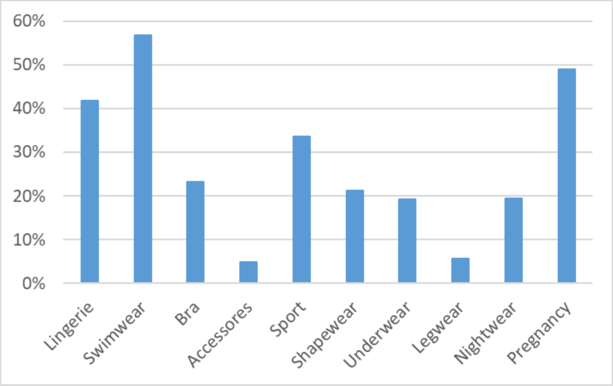


Figure 2: Return rate per weekday

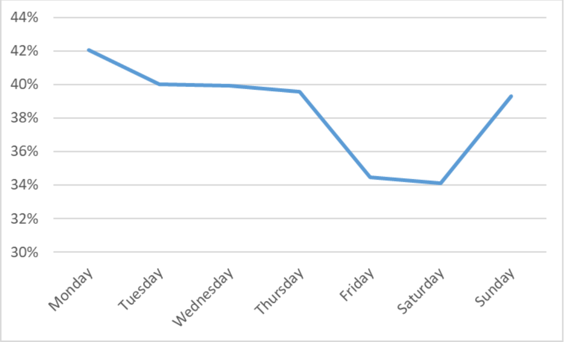


Figure 3 shows the return rate for each month throughout the year. There are quite some differences in the return rate, especially with a dip in November, which could be caused by the fact that a lot of presents (for Sinterklaas and Christmas) are ordered in that period and fewer items are returned due to this. Figure 4 shows that in the morning and afternoon, fewer items are returned than in the evening. This could be the cause of webshop orders that are returned after a working day.

Figure 3: Return rate per month

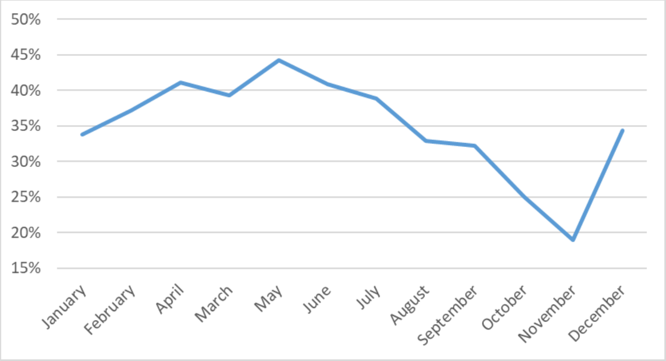
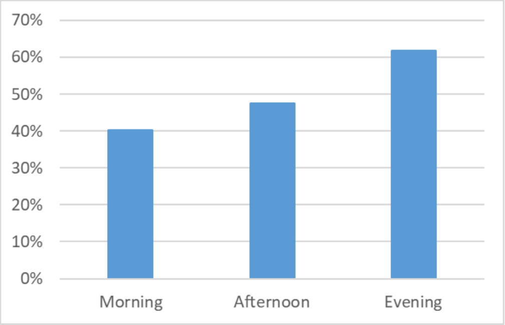


Figure 4: Return rate per time of the day



Appendix C: Hyperparameter tuning

Table 1: Neural network instore

size	decay	AUC	Precision	Recall	F	AUCSD	PrecisionSD	RecallSD	FSD
1	0	0.1452	0.9142	0.8033	0.8552	0.3104	0.0477	0.4361	0.4040
1	0.0001	0.1620	0.7636	0.9959	0.8644	0.3295	0.2385	0.0086	0.1622
1	0.1	0.9356	0.9670	0.9990	0.9827	0.0857	0.0170	0.0007	0.0088
3	0	0.7581	0.9441	0.9979	0.9703	0.2413	0.0324	0.0020	0.0167
3	0.0001	0.6013	0.9403	0.9678	0.9538	0.4082	0.0286	0.0666	0.0465
3	0.1	0.9859	0.9979	0.9717	0.9846	0.0053	0.0052	0.0007	0.0029
5	0	0.9541	0.9649	0.9967	0.9806	0.0392	0.0097	0.0024	0.0044
5	0.0001	0.7878	0.9605	0.9982	0.9790	0.4321	0.0185	0.0020	0.0093
5	0.1	0.9783	0.9569	0.9838	0.9702	0.0171	0.0332	0.0300	0.0314

Table 2: Neural network webshop

size	decay	AUC	Precision	Recall	F	AUCSD	PrecisionSD	RecallSD	FSD
1	0	0.0972	0.7062	0.1465	0.2427	0.2132	0.0142	0.3277	0.0060
1	0.0001	0.1785	0.7527	0.2814	0.4097	0.2810	0.0302	0.3860	0.0100
1	0.1	0.8016	0.7755	0.6815	0.7254	0.0064	0.0226	0.0276	0.0109
3	0	0.5244	0.7483	0.5668	0.6451	0.3646	0.0396	0.3206	0.0090
3	0.0001	0.7856	0.7727	0.6771	0.7218	0.0087	0.0164	0.0218	0.0098
3	0.1	0.7529	0.7194	0.8117	0.7628	0.0145	0.0169	0.0231	0.0107
5	0	0.7829	0.7678	0.6887	0.7261	0.0729	0.0131	0.0227	0.0078
5	0.0001	0.7881	0.7615	0.6927	0.7255	0.0590	0.0424	0.0419	0.0103
5	0.1	0.8097	0.7620	0.6888	0.7236	0.0182	0.0187	0.0145	0.0132

Table 3: Random forest in-store

mtry	splitrule	AUC	Precision	Recall	F	AUCSD	PrecisionSD	RecallSD	FSD
2	gini	0.9955	0.9709	0.9996	0.9850	0.0016	0.0034	0.0004	0.0017
2	extratrees	0.9910	0.9472	1	0.9729	0.0035	0.0055	0	0.0029
32	gini	0.9969	0.9998	1	0.9999	0.0067	0.0041	0.0010	0.0020
32	extratrees	0.7491	0.9835	0.9963	0.9899	0.0134	0.0037	0.0011	0.0021
62	gini	0.5640	0.9833	0.9908	0.9871	0.0216	0.0033	0.0016	0.0012
62	extratrees	0.6240	0.9845	0.9939	0.9891	0.0117	0.0037	0.0008	0.0019

Table 4: Random forest webshop

mtry	splitrule	AUC	Precision	Recall	F	AUCSD	PrecisionSD	RecallSD	FSD
2	gini	0.8012	0.7515	0.6915	0.7202	0.0092	0.0064	0.0174	0.0107
2	extratrees	0.7801	0.7313	0.6502	0.6884	0.0125	0.0083	0.0153	0.0118
32	gini	0.8253	0.7222	0.7673	0.7441	0.0096	0.0090	0.0148	0.0109
32	extratrees	0.8286	0.7536	0.7218	0.7373	0.0096	0.0058	0.0180	0.0109
62	gini	0.6209	0.7578	0.7115	0.7340	0.0062	0.0081	0.0143	0.0105
62	extratrees	0.8276	0.7524	0.7210	0.7364	0.0108	0.0080	0.0183	0.0128

Table 5: AdaBoost in-store

max tree depth	#trees	AUC	Precision	Recall	F	AUCSD	PrecisionSD	RecallSD	FSD
1	50	0.9436	0.8803	0.8019	0.8393	0.0040	0.0115	0.0202	0.0096
2	50	0.9888	0.9489	0.9546	0.9517	0.0021	0.0070	0.0068	0.0052
3	50	0.9918	0.9704	0.9728	0.9716	0.0015	0.0037	0.0040	0.0026
1	100	0.9743	0.9194	0.8837	0.9012	0.0015	0.0066	0.0197	0.0072
2	100	0.9922	0.9576	0.9724	0.9649	0.0015	0.0011	0.0026	0.0018
3	100	0.9933	0.9795	0.9799	0.9797	0.0015	0.0024	0.0015	0.0013
1	150	0.9838	0.9298	0.9391	0.9344	0.0011	0.0052	0.0085	0.0018
2	150	0.9931	0.9660	0.9769	0.9714	0.0015	0.0027	0.0024	0.0011
3	150	0.9905	0.9983	0.9826	0.9904	0.0017	0.0030	0.0012	0.0016

Table 6: AdaBoost webshop

max tree depth	#trees	AUC	Precision	Recall	F	AUCSD	PrecisionSD	RecallSD	FSD
1	50	0.7357	0.7580	0.7168	0.7368	0.0060	0.0209	0.0273	0.0072
2	50	0.7449	0.7583	0.7251	0.7413	0.0072	0.0164	0.0257	0.0132
3	50	0.7518	0.7596	0.7536	0.7566	0.0055	0.0052	0.0127	0.0059
1	100	0.7411	0.7578	0.7272	0.7422	0.0064	0.0144	0.0156	0.0066
2	100	0.7479	0.7608	0.7310	0.7456	0.0052	0.0132	0.0121	0.0069
3	100	0.7525	0.7538	0.7456	0.7497	0.0056	0.0100	0.0082	0.0057
1	150	0.7426	0.7338	0.7630	0.7481	0.0074	0.0331	0.0465	0.0081
2	150	0.7484	0.7615	0.7319	0.7464	0.0072	0.0140	0.0070	0.0073
3	150	0.7540	0.7505	0.7531	0.7518	0.0049	0.0167	0.0281	0.0086