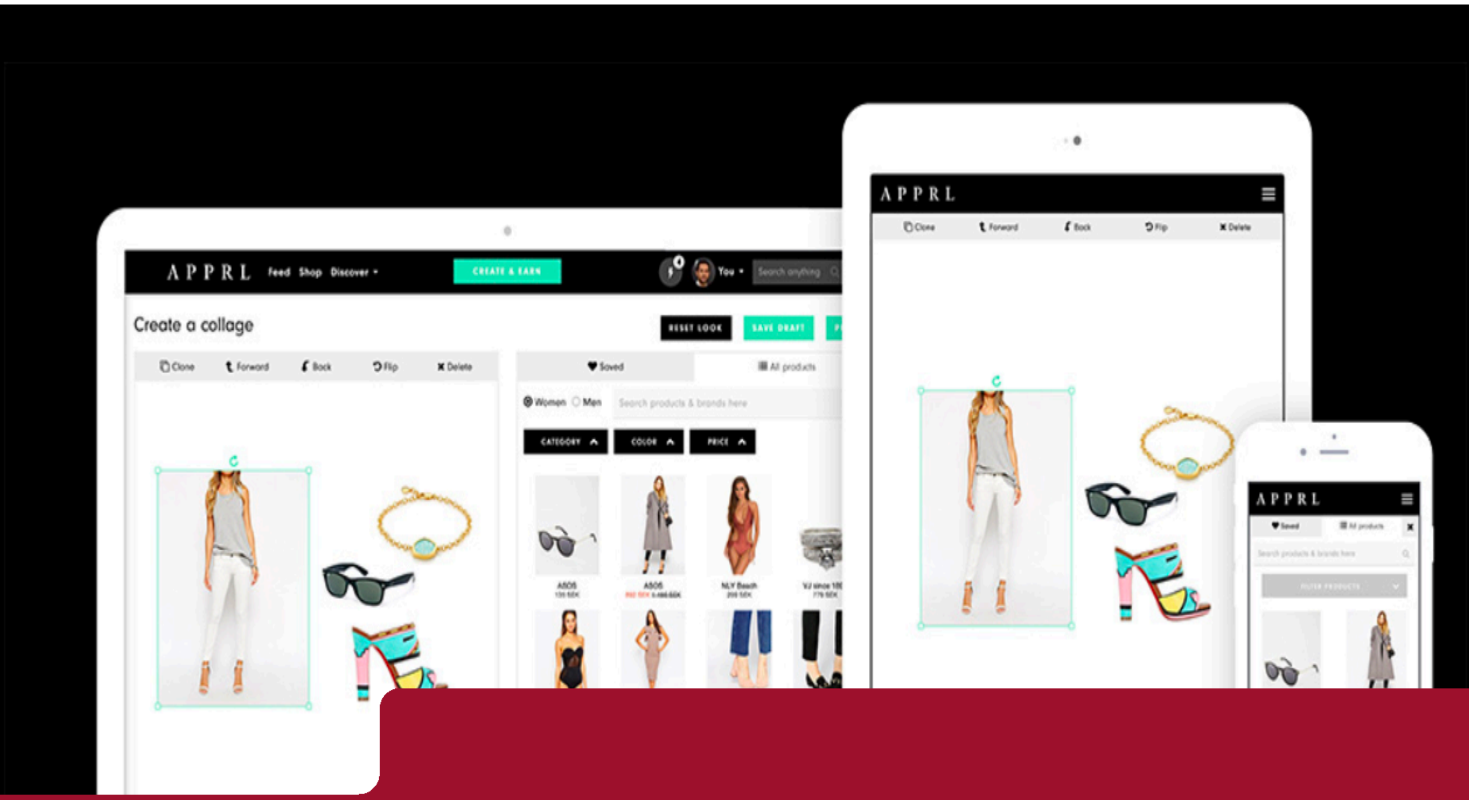
DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2017

Predicting Fashion using Machine Learning techniques

MONA DADOUN



KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION

Predicting Fashion using Machine Learning techniques

Master Degree Project in Computer Science and Communication, Second Cycle
DA225X

Mona Dadoun
dadoun@kth.se

Supervisor at CSC Richard Glassey
Supervisor at Apprl Matti Kotsalainen
Examiner Viggo Kann

KTH Royal Institute of Technology
Stockholm, Sweden
June 18, 2017

Abstract

On a high-level perspective, fashion is an art defined by fashion stylists and designers to express their thoughts and opinions. Lately, fashion have also been defined by digital publishers such as bloggers and online magazines. These digital publishers create fashion by curating and publishing content that is hopefully relevant and of high quality for their readers. Within this master's thesis, fashion forecasting was investigated by applying supervised machine learning techniques. The problem was investigated by training classification learning models on a real world historical fashion dataset. The investigation has shown promising results, where fashion forecasting has been achieved with an average accuracy above 65 %.

Att förutspå mode med maskininlärning

Examensarbete inom Datavetenskap och Kommunikation
Avancerad nivå DA225X

MONA DADOUN
dadoun@kth.se

CSC Examinator Viggo Kann
CSC handledare Richard Glassey
Apprl handledare Matti Kotsalainen

Kungliga Tekniska Högskolan
Stockholm, Sverige

Sammanfattning

På en abstrakt nivå definieras mode av stylister och designers. Dessa väljer att uttrycka sina tankar och åsikter genom att skapa mode. På senare tid har mode också definierats av digitala förlag som bloggare och onlinemagasin. Dessa digitala förlag definierar mode genom att skapa och publicera innehåll som förhoppningsvis är relevant och av hög kvalitet för sina läsare. I den här uppsatsen, undersöktes modeprognoser genom att använda sig av övervakade maskininlärningstekniker. Problemet undersöktes genom att lära klassificeringsinlärningsmodeller på ett verkligt historiskt dataset för mode. Undersökningen har visat lovande resultat där modeprognoser har kunnat nås med en genomsnittlig noggrannhet över 65 %.

Contents

1 Introduction

In the introduction section objectives and motivation to the problem will be presented as well as the problem formulation.

1.1 Fashion

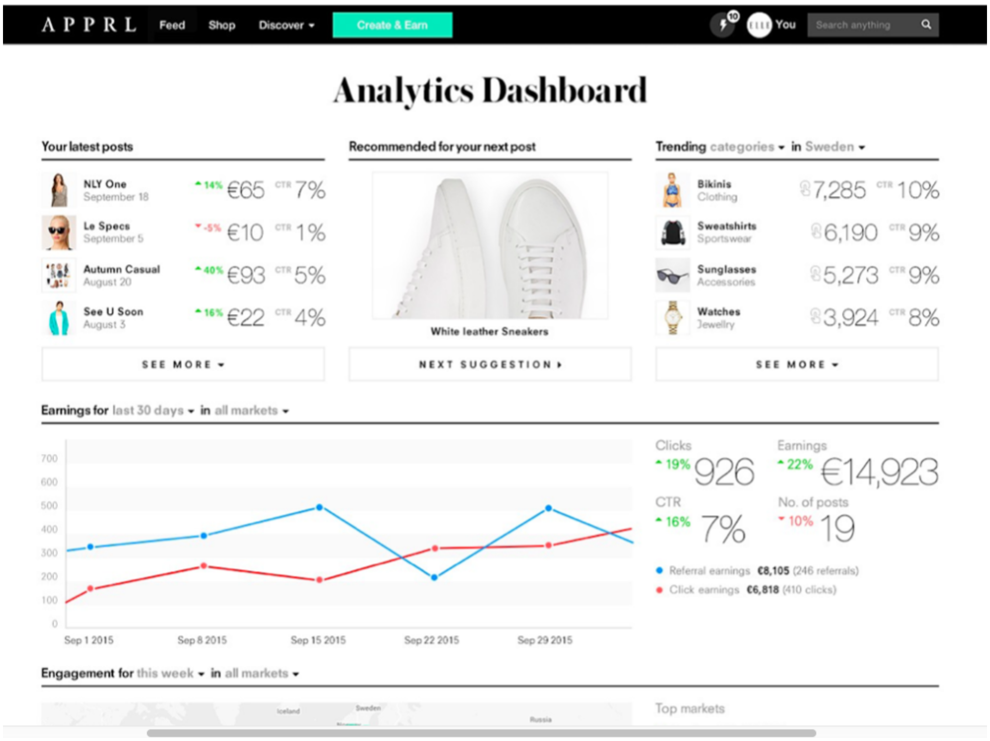
Fashion is, on a high-level perspective, an art where stylists and designers choose to express their thoughts and opinions by using textile as tools (Thomassey, 2014). Lately, digital publishers, such as bloggers and online magazines, have been expressing fashion by curating fashion content. These influencers inspired people on a daily basis, by publishing fashionable contents on social media platforms. Content curation is when an individual pull together, organizes and selects the most relevant and highest quality digital content on a specific topic (Lord, Macdonald, Lyon, & Giaretta, 2004). The recent influx of digital publishers in the fashion industry begs the question: is there anything that the tech industry and machine learning can contribute with to the fashion industry to modernize the logistics of fashion even further?

1.2 Apprl and its objectives

Apprl is a tech startup company that connects online retailers with content of online magazines, blogs and other digital publishers. Apprl enables digital publishers to create different types of curated content such as mini-shops, photos tagged with products, product lists and widgets. For three years, Apprl has been collecting editorial shopping data that contains information on each product for example, sales, colors, brands and categories.

Apprl's aim is to automate the curated content process, by offering a product suggesting system for the digital publishers. The suggested products as shown in figure 1, would be those products predicted by machine learning models that were applied by this degree project. By these product suggestions, the hope is to automate the curated content and make the work easier for digital publishers and retailers to create better content, enhance monetization and better understand their audiences.

Figure 1: Apprl's objective of using the predictions by machine learning



Analytics Dashboard

Your latest posts

NLY One September 18	▲14%	€65	CTR 7%
Le Specs September 5	▼-5%	€10	CTR 1%
Autumn Casual August 20	▲40%	€93	CTR 5%
See U Soon August 3	▲16%	€22	CTR 4%

Recommended for your next post

White leather Sneakers

Trending categories ▾ in Sweden ▾

Bikinis Clothing	7,285	CTR 10%
Sweatshirts Sportswear	6,190	CTR 9%
Sunglasses Accessories	5,273	CTR 9%
Watches Jewellry	3,924	CTR 8%

Earnings for last 30 days ▾ in all markets ▾

Clicks ▲19% 926

Earnings ▲22% €14,923

CTR ▲16% 7%

No. of posts ▼10% 19

● Referral earnings €8,105 (246 referrals)

● Click earnings €6,818 (410 clicks)

Engagement for this week ▾ in all markets ▾

Top markets

1.3 Objectives and problem statement

Fashion forecasting is a term with a variety of meanings. In the context of this master's thesis, it is specifically referring to the sub-problem of generating curated content based on predictions derived from prior curated content. For the digital publishers, content curation is integral to online marketing strategy. Having effective content curation helped positioning as a strong leader in a market and is an economical way to maintain a consistent publishing of quality content. However, manually curating the content in fashion industry could be time-consuming and a challenging problem. To curate effectively in terms of time and relevance thus required automation.

Different machine learning techniques have shown that it is possible to predict for example opinions and trends on multiple domains. However, very little academic research has been done to investigate machine learning's possibilities within the fashion industry. Therefore, this master's thesis aimed to apply supervised machine learning techniques in order to investigate the possibilities of predicting fashion.

Since machine learning performance is dependent on the quality of the input data it is important to extract as much information as possible from the historical dataset (Thomassey, 2014). Therefore, in order to achieve the objective of this master's thesis, it is vital to analyze and collect a dataset that is relevant and of a high quality.

1.3.1 Research question

To what extent is it possible to predict the most popular and unpopular products in terms of number of clicks, sales rates and popularity rates, using machine learning techniques and by leveraging APPRL's dataset?

1.3.2 Data limitations

When Apprl started collecting data, it was not primary intended to be used by machine learning models. Therefore, to extract maximum information from the database, the data is pre-processed, transformed, tested and eventually recollected in an iterative process during this degree project. The data consisted initially of millions of entries, but the final selected dataset had much smaller size. The size got smaller because of removal of non-usable, redundant features and incomplete entries. The size of the dataset has therefore limited the performance and the choice of learning models. For example, deep learning and neural network learning models have not been investigated within this master's thesis because of the small size of the dataset. Further, within this degree project, short and long trends in fashion were not investigated.

1.4 Outline

In the coming sections, this master's thesis provides a theoretical framework investigating several machine learning on a high level and relevant fashion theory in section 2. In section 3, previous and related work will be discussed. In method section, section 4, a description of the process of collection of the data and pre-processing as well as software libraries and machine learning techniques will be presented. Then, in section 5, the result section, all different results for each dataset will be presented, followed by analysis and discussions in section 6. Finally, this master's thesis's conclusion and suggestions for future work will be presented in section 7 and 8.

2 Background

The background section contains an informative view of the area in which this thesis is conducted in. The background section begins with a presentation of the underlying fashion forecast theory in terms of the subproblem curated content. Further, main elements of machine learning such as evaluation and metrics will be presented.

2.1 How curated content become a reality

In ancient times, the most common form of advertising was 'word of mouth' (Dellarocas, 2003). Somehow, in modern time, we were back again to the 'word of mouth' advertising, but this time it is called online word of mouth (Steffes & Burgee, 2009). In the ruins of Pompeii, commercial messages were found, and Egyptians used papyrus to create sales messages and wall posters. In the 16th and 17th century when printing was developed, advertising expanded to appear in handbills. During the 18th century advertisements started to appear in weekly newspapers. And during 20th century, the need for advertising expanded to TV and radio.

Today, advertising is evolving even further and in higher speed than ever. With the growth of world wide web, advertisement expanded to the Internet by using pop-ups and banners within websites. The problem with the pop-ups and banners is that this advertisement is psychologically perceived as not relevant by people (Benini, Batista, & Zuffo, 2005). Therefore, the rise of ad blockers and similar extensions became more common, making these kinds of advertisement disappear from websites (Post & Sekharan, 2015). While pop-up and banners ending era is becoming a reality, blogging and digital publishing became a trending within the world wide web (Bruns, 2009). In human history, it has never been as easy to connect and share news with so many peoples as it is today (Deuze & Bardoel, 2001). Blogging became the new information source for many people (Dearstyne, 2005), where they could read and follow for example, their favorite fashion bloggers.

2.1.1 Curated content within fashion industry

Within the fashion industry, there are fashion creators, for example, Chanel, Filippa K, Nike, Adidas. These fashionistas create fashion based on art and other inspiration sources. Additionally, there are other type of fashion influencers, such as digital publishers, who inspires people on daily basis (Halvorsen, Hoffmann, Coste-Manière, & Stankeviciute, 2013). In comparison to the pop-ups and banners advertisement, the digital publishers' curated contents are perceived as inspirational and relevant material by their audiences (Kretz, 2010). However, there is an eminent difference between advertisement and content curation that has to be pointed out: the contents' relevance.

The digital publishers became very valuable to the fashion industry and have therefore been working closely with retailers on marketing objectives (Odden, 2012). Digital curating, and data curating is not a new phenomenon (Abbott, 2008). The term "curate" is defined as: "pulling together, sifting through, and selecting for presentation" (Dictionary.com, n.d.). Content curation is when an individual consistently finds, organizes, annotates, and shares the most relevant and highest quality digital content on a specific topic for its target market (Lord et al., 2004).

For the digital publishers, content curation consists of finding material relevant to their audience. This information can be found on a variety of sources (Abbott, 2008). However, contents require curating of strong examples that should be relevant to what they are publishing. The curating process can be time-consuming and cumbersome. Thus, automation of content curation by using machine learning techniques, becomes an important feature. Instead of just picking the generally profitable things, machine learning techniques can help curate the most relevant contents for a specific audience.

2.2 Supervised Machine Learning

Lately, the growth of data size and dimensionality has been high (Tang, Alelyani, & Liu, 2014). As more data becomes available, efficient and elective management of the data becomes increasingly challenging. Therefore, it is no longer sufficient nor practical for the human being to manually manage these vast amounts of data (Tang et al., 2014) nor manually do the analytical tasks such as establishing relations between different functions or attributes, calculating complex models and predicting trends. To solve this problem, machine learning techniques and data mining have been investigated and used to automatically figure out how to perform analytical tasks by generalizing from examples, discovering knowledge and by finding patterns (Domingos, 2012). An original definition of machine learning is given 1959 by Arthur Samuel:

"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed" - Arthur Samuel (1959).

Machine learning models are developed on sets of mathematical and statistical formulas that learn to find patterns and make decisions based on previous experience (Marsland, 2015). Therefore, machine learning techniques are used to perform analytical tasks such as classification and regression (Kotsiantis, Zaharakis, & Pintelas, 2007).

Further, there are two main machine learning approaches, supervised learning and unsupervised learning. Supervised machine learning is the task of inferring a function from labeled training data (Bishop, 2007). Within supervised machine learning, predictive modelers are given a set of feature instances together with corresponding correct outputs (labels). These models forecast future outputs based on prior knowledge by being trained on training data and evaluated on testing data (Mohri, Rostamizadeh, & Talwalkar, 2012).

When dealing with unlabeled data, unsupervised machine learning is used instead. All the observations are assumed to be caused by latent variables. Hence, the goal of unsupervised machine learning is to describe data's structure by organizing it for example by grouping the data into clusters (Jain, Murty, & Flynn, 1999).

2.2.1 Classification and Regression

Supervised machine learning has two main approaches, classification and regression (Bishop, 2007). These approaches of learning are applied on problems depending on the goal of the learning. If the desired output consists of one or more continuous variables, then it is called a regression problem (Bishop, 2007) for example, stock price forecasting. However, when having the goal to predict a category, then the problem is called classification problem. For example, having a set of 'cat' and 'dog' images, each image would be assigned either a label 'cat' or a 'dog'.

2.2.2 Main elements of classification algorithms

Classifier is the name of a learning models used for classification. Classifier's definition is as follows: a mathematical function that map input data to a category (or class), which means a classifier is a system that in general, inputs a vector of discrete or continuous feature values and outputs a single discrete value i.e. the class (Marsland, 2015).

The observations made are usually known as features or classes (also known as labels) and are the possible categories that are predicted by the classifier. Further, classification can be thought of as two separate subjects i.e. binary classification, as the given example with 'cat' or a 'dog' in section 2.2.1, or as multi-class classification; for example, suppose adding one more category i.e. a *goat* to the cat & dog example. Hence, the labels can be translated into the numbers 0,1,2 (James, Witten, Hastie, & Tibshirani, 2013).

2.3 Classifiers

There are multiple classifiers with multiple purposes, developed to solve classification problems. There is rule based classifiers, such as Decision Trees, that can find patterns in qualitative data. Additionally, there are other classifiers such as K Nearest Neighbors and Adaboost that are good in predicting patterns in both qualitative and quantitative types of data (James et al., 2013).

2.3.1 Decision Tree

Decision Tree (DT) classifier is basically defined as a classification procedure that recursively partition data based on a set of rules defined at each tree node (or branch). Each inner node, has a decision rule that determines which instances are assigned to each child node. Further, the class label of the leaf node will be the class predicted by the learning model DT (Friedl & Brodley, 1997).

However, several types of DT's have been developed during time. The most well-known is the ID3 algorithm, which is developed by Ross Quinlan in 1986. The algorithm creates a tree and for each node finds a categorical feature that will yield the largest information gain for categorical targets. Trees are grown to their maximum size and then a pruning step is usually applied to improve the ability of the tree to generalize to unseen data (Marsland, 2015).

2.3.2 Ensemble learning models

In short, ensemble methods use multiple learning algorithms to obtain better predictive performance than can be obtained from one learning model alone (Bishop, 2007). These learning models apply either *bagging* (also known as bootstrap aggregating) (Breiman, 2001) *boosting* (Friedman, Hastie, Tibshirani, et al., 2000). Bagging is a sampling machine learning meta algorithm used to improve learning models' performance Random forest build many individual decision trees, from which a final class assignment is determined.

For instance, Random forest is one of many ensemble learning classifiers, developed to apply *bagging* to build many individual decision trees. It operates by constructing decision trees at training time and outputs the predicted classes. Each tree in the ensemble is built from a sample of the training set using bootstrap method. During the splitting of a node, the split that is picked is the best split among a random subset of features. The randomness results in a slight increase in the bias of the forest but due to averaging, its variance decrease, hence, yielding an overall better model.

Meanwhile, boosting is used by the learning model Adaboost. Adaboost (AB) is a well-known boosting learning algorithm developed by Yoav Freund and Robert Schapire. The basic idea of AB is to fit a sequence of weak learners who performs slightly better than random guessing, iteratively on modified versions of dataset by applying weights to the training set within each boosting iteration (Bishop, 2007). By assigning those training examples that were incorrectly predicted increased weights, and those that were correctly predicted by decreased weights, the weak learners are forced to focus on the examples that were miss predicted by the previous learners in the sequence. Finally, the predictions from all the learners are combined through a weighted majority vote to produce the final prediction (Bishop, 2007).

2.3.3 K Nearest Neighbors

The classifier K Nearest Neighbors (KNN) is one of the simplest non-parametric classifiers (Bishop, 2007). Non-parametric learning models are models that grows in number of parameters grow with the amount of training data. These models can be more flexible but computational heavy (Bishop, 2007). In classification, the idea behind K Nearest Neighbors algorithm is to predict a label based on finding a predefined number of training samples closest in distance to the new point. The distance can be calculated by any metric for example, Euclidean distance which is the most common choice, or Manhattan distance (Marsland, 2015). The K in K Nearest Neighbors, is a user-defined constant.

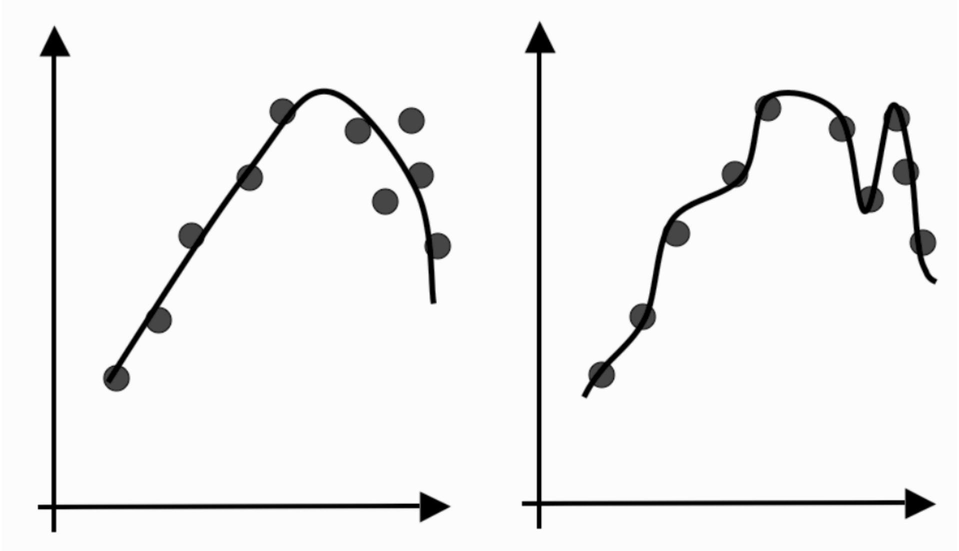
2.3.4 Linear models

Logistic Regression (LR) is a linear learning model used for classification problems, even though the name can be confusing (Marsland, 2015). This learning model is developed specially to predict probabilities of classes. The predictions are made using a logistic function, where a logistic function is a common “S” shape (sigmoid curve) (Ng & Jordan, 2002). Further, logistic regression is known to perform better on smaller datasets (Perlich, Provost, & Simonoff, 2003). Furthermore, Stochastic Gradient Descent (SGD) is another learning model developed to fit linear models. This classifier is especially useful when the number of features is medium large (Marsland, 2015).

2.3.5 Overfitting

Overfitting occurs when a learning model is trained on a limited set of data points into an extent that it learns the details and the noise of the data (Bishop, 2007). For instance, the graph in the right side in figure 2 shows an overfitting example. The line follows the data points exactly while the left graph in figure 2 shows a line that is less fitted which means that the learning model generalizes well (Marsland, 2015). Overfitting can affect the learning models’ performance negatively when predicting on never seen data. Therefore, overfitting should be avoided

Figure 2: Overfitting



by not increasing the complexity of the learning models: trying to avoid fitting each single input data point variation (Bishop, 2007).

2.4 Evaluation and metrics

Before a machine learning model can be used it has to be tested and evaluated for accuracy on test data (data that the model is not trained on). This can often be done by choosing appropriate metrics for evaluating the performance (Bishop, 2007).

2.4.1 Evaluation methods

The goal of learning is to predict the outputs as good as possible. To know how successfully a learning algorithm has performed, prediction made by a classifier can be compared with known labels (Marsland, 2015). The first step would be to partition a dataset into a training and testing dataset, where the training dataset are used to build the initial model, and the testing dataset (or evaluation dataset) are used to evaluate the learning model (Bishop, 2007).

Testing dataset needs to be preprocessed through identical steps as the training dataset (Bishop, 2007). To obtain a well performing model, the training dataset needs to cover the full range of values for all features the model might encounter (Marsland, 2015). To achieve this, there are two commonly used partitioning techniques. The first one is partitioning a dataset by randomly sampling 80 % of the dataset as training data and the left 20 % of the dataset is used as testing data (Bishop, 2007). The second partitioning technique is by sampling data using a technique called K-fold cross validation. The main steps of k-fold cross validation described by the authors Marsland are:

1. Randomly split the dataset into K equal partitions or folds, where K = 2, ..., N.
2. One subset (fold) is used as a testing set, while the algorithm is trained on the union of the other folds (training set)
3. Calculate the testing accuracy
4. Repeat step 1,2,3 K times by using a different fold as the testing set in each iteration.
5. Finally, the model that produced the lowest validation error is tested and used. Use the average testing accuracy as the estimate of out-of-sample accuracy.

K-fold cross validation goal is to minimize the risk for overfitting, and decrement the high variance that can result from partitioning data using the random splitting method (Marsland, 2015). The K in K-fold cross validation can be set manually. Further, for classification problems, stratified sampling is recommended for creating the balanced folds i.e. each fold should hold an equal proportion of each class. Scikit Learn's cross validation method implements this by default.

2.4.2 Metrics

Precision, recall, the F1 score (also called accuracy within this master's thesis) (James et al., 2013) and confusion matrix (Marsland, 2015) are commonly used metrics for classification problems. Considering the possible outputs of the classes, they can all be expressed in the following terms:

- True Positives (TP): an observation of class 1 correctly put into class 1
- True Negatives (TN): an observation of class 2 correctly put into class 2

- False Positives (FP): an observation of class 2 incorrectly put into class 1
- False Negatives (FN): an observation of class 1 incorrectly put into class 2

Precision Precision (also known as positive predictive value), is defined as the ratio of correct positive examples to the number of actual positive examples and is calculated as follow:

$$precision = \frac{TP}{TP + FP}$$

Recall While recall (also known as True positive rate) is the ratio of the number of correct positive examples out of those that were classified as positive (Marsland, 2015):

$$recall = \frac{TP}{TP + FN}$$

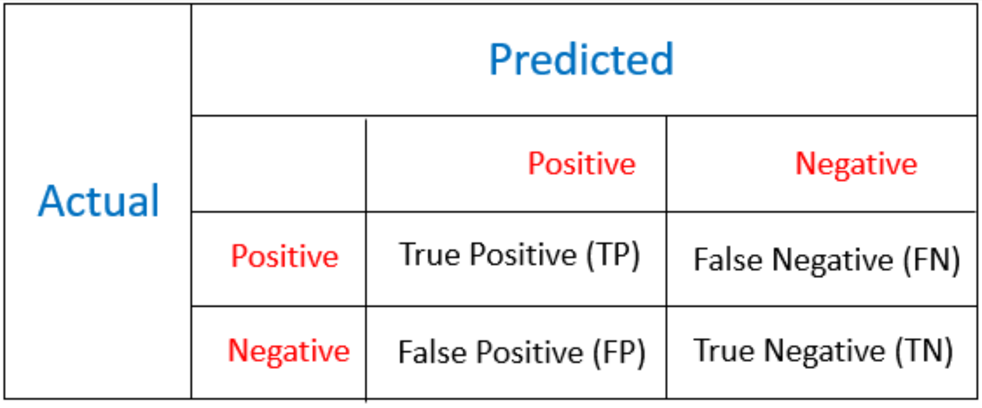
F1-score Finally, F1-score (also called accuracy), is the harmonic mean of precision and recall. The accuracy indicates the classifier's overall performance and is calculated as follows:

$$F_1 = \frac{TP}{TP + \frac{FN+FP}{2}}$$

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Confusion Matrix Confusion matrix (also known as error matrix), is a table that visualize the overall performance of a learning model (Marsland, 2015). This matrix summarizes the metrics discussion in section 2.4.2. Each column of the matrix shown in 3 (grouped by predicted), specifies the instances in a predicted class while each row (grouped by actual) represents the instances in an actual class.

Figure 3: Confusion Matrix



Actual		Predicted	
		Positive	Negative
	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

3 Related work

There few studies that have explicitly studied fashion forecasting using machine learning techniques. However, those that was found have especially focused on using regression models to predict on sales datasets. These models were often complex machine learning models. Other studies, focused on increasing the performance of the learning models that have been studied.

In 2008, Au, Choi, and Yu presented evolutionary neural network (ENN) to forecast within fashion retail. The problem was presented in a scenario where fashion retailers were posed to highly unpredictable demand of fashion products. These unpredictable demands left the retailers either with high stocks or stock-outs leading to economic issues. Therefore, the aim with this study was to find an ideal network structure for a forecasting system based on a time series apparel sales data. This to help the retailers reduce the inventory burden. However, the study focused on forecasting short term fashion trends. The authors stated that fashion sales are usually influenced by short-term factors, which often last for 2 weeks. Therefore, they considered a two weeks of history data to be enough to provide information for forecasting. The performance of the proposed model, ENN, was compared with a traditional forecasting models called SARIMA and with a basic Neural Network outputs. Considering apparels with the parameters, *low demand* and *weak seasonality trend*. The authors Au et al. found that the proposed model, was useful for fashion retail forecasting which shares the features' short-term trend. The performance of the model was found to be better than the traditional forecasting models.

In 2010, the authors Wong and Guo suggested a hybrid intelligent (HI) sales forecasting model by applying it to medium-term fashion sales real world data. For example, categories are usually the same, while the items in each category frequently change in different selling seasons. A category T-shirt can stand for 150 different models of T-shirts during one season, that will probably be replaced by 100 new models of T-shirts during another season. The HI was developed by integrating a harmony search algorithm with an extreme learning model in order to optimize the fashion forecasting model and generate better performance. However, the HI model was based on a novel Artificial Neural Network algorithm, which generated the initial forecasts. Then, a heuristic fine-tuning process was applied on it, to improve the accuracy of the sales forecasting model. The results have shown that the proposed model performed better than simple linear learning models and novel neural networks on sales fashion forecasting.

In 2012, a study done by Xia, Zhang, Weng, and Ye investigating fashion forecasting models on sales datasets. To avoid stock-out and maintain a high inventory fill rate, fashion retailers were very dependent on an accurate forecasting system. Therefore, the authors of the paper examined a hybrid method based on extreme learning machine model (EML) with the adaptive metrics of inputs, called AD EML. The authors of the study observed that ANN tended to suffer from overfitting of networks especially for fashion retailing sales data. Therefore, they proposed an improvement of the forecasting system based on ELM, that resulted in reduced effect of the overfitting and improvements in the sales forecasting accuracy. The algorithm used was trained on real data from three different fashion retailers based in Hong Kong. However, it was found that the proposed model, AD EML, is practical for fashion retail sales forecasting and that this model outperformed the ANN and ELM (Xia et al., 2012). However, the same authors proposed a better model in 2014 (Xia & Wong, 2014) where they investigated if it was possible to improve forecasting accuracy and overcome the seasonality and limited data by using Grey forecasting models (GM).

Unlike other studies, the authors Choi, Hui, Ng, and Yu, investigated fashion forecasting in context of sales and colors. The study was applied on four years of real world data from a cashmere retailer. They tested various of forecasting methods such as ANN, GM and hybrid models. For example, they found that a hybrid model based on ANN + GM performed best in terms of small amounts of data. They conclude that when forecasting fashion sales with colors, the ANN + GM hybrid model was best to use in particular on the cashmere retailer's dataset.

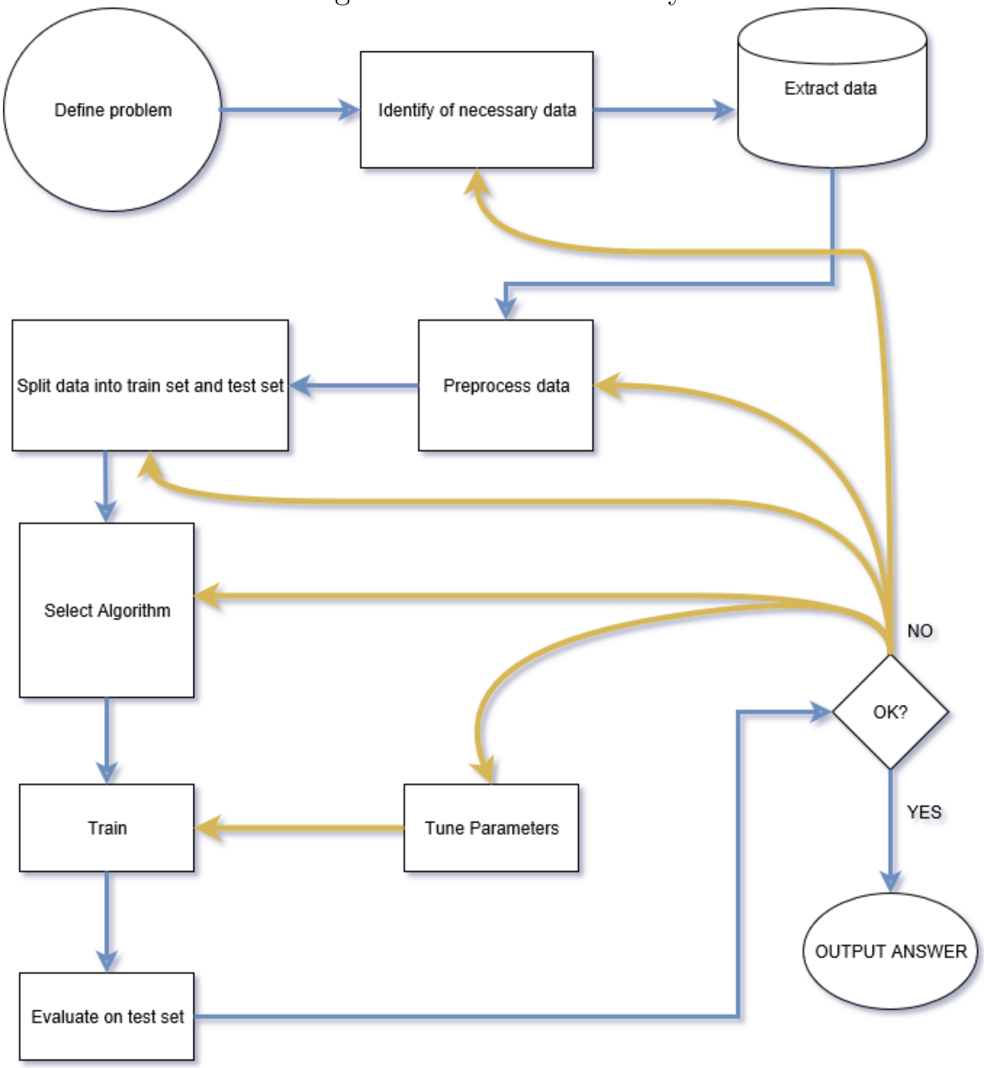
4 Method

In this section, the method for this degree project is described. The aim of the method is to provide an understanding of made choices in order to answer the research question adequately. This section will therefore present chosen datasets, features, labels, machine learning models as well as the parameters of the models.

4.1 Working process

A summary of the method section can be seen in Figure 4. Starting with the definition of the problem, it can be found in section 1.3. Identification of necessary data have been made by getting an understanding of the business model and by analyzing Apprl's dataset. Then the extraction step used was done using an open source SQL database. After the extraction of the dataset, it was preprocessed and split into training and testing set using random split method and cross validation methods. Further, a choice of learning models had to be done, therefore, several learning models have been trained, tested and evaluated by using evaluation methods and measured using metrics described in section 2.4. This process was however performed sequentially, and several times until qualified results were achieved. In figure 4 the yellow arrows represents the sequential steps and blue arrows describes in which order the general steps were performed.

Figure 4: Method summary



4.2 Software and libraries

The classifiers (learning models) investigated within this degree project were developed by Scikit Learn library. Scikit Learn is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems (Pedregosa et al., 2011). Scikit Learn library was chosen to work with since it was one of the most commonly used data science software libraries by data scientists. It was easy to work with, and had minimal dependencies. Since Scikit Learn was coded in Python, this degree projects code base was also written in Python (version 3.6).

4.3 Data Description

Since the data studied within this thesis, was labeled, the conducted empirical experiments applied a supervised learning approach. Further the problem statement concerned a classification problem. The dataset studied within this master's thesis was collected from Apprl's database. Apprl's network which consisted of bloggers and online magazines, collected hundreds of thousands of monthly visitors to online retailers, each one generating unique pieces of data. Apprl started collecting this data in 2012 and initially collected it for other implementation intentions than machine learning. The given historical data generated by Apprl, contains 3 243 806 feature vectors. Each collected data point represented each product's information such as category, brand, publisher, color and vendor.

4.4 Identification of necessary data

The most challenging part during this experiment have been to find relevant datasets to experiment on. However, as in any other machine learning project, several necessary steps have been compulsory to perform during data selection step. These necessary steps will be presented in the following sections.

4.4.1 Data selection process

Description of a small extract from the database, for each column can be seen in table 1.

Table 1: Product Data

Category	**Definition**
Brand	product brand name
Category	General name of category for example shoes, shirts etc.
Colors	product colors: white, gray, black etc.
Currency	EUR
Sales	Sale amount by product
Gender	U = Unisex; W = Woman; M = Man
In stock	If product was in stock, True/False
Vendor	Vendor selling the product
Name	Name of the product
Regular price	Product regular price
Popularity	A popularity rate calculated by product
User	Name of the publisher who published the product
Sales date :	Date when the product was sold
Clicks	Number of clicks a product generated

To select data in first place, it was extracted from a database using PostgreSQL. According to Thomassey, to be able to make good predictions, the goal is to extract greatest information from the historical dataset. To achieve this goal within this master's thesis, the selection process was performed sequentially, by performing all the steps described in figure 4. However, a more detailed description of the final selected data can be found on table 1 and in the result section.

1. *Sales dataset* varies in size depending on included features. An extraction of at most 12 000 feature vectors were possible before any preprocessing.

2. *Click dataset* varies in size depending on number of features included. It was possible to pull out at most 120 000 feature vectors from the database before any preprocessing.

3. *Popularity dataset* consisted of 120 000 feature vectors when it was pulled from the database.

4.5 Preprocessing data

Preprocessing of the data was important for both making the pattern recognition easier and for speeding up computations (Bishop, 2007). With the preprocessing step, the aim was to find useful feature vectors that were fast to compute, and yet preserved useful information.

4.5.1 Feature extraction

Firstly, an intuitive understanding of the business model was necessary to select necessary features. Secondly, a choice of a main approach for feature extraction process was made. The feature extraction process was built upon a combination of *step-wise forward selection and step-wise backward elimination.*

Step-wise backward elimination is when starting with all features and eliminating with the least informative feature.

Step-wise forward selection is when starting with the best feature, and then adding the next beast feature in condition to the first.

The final selected features were:

- Category: for example Shirt, Shorts, Make-up
- Brand: Multiple brands both new and well-established brands were represented in the database
- Gender: Gender could be Woman, Male, Unisex
- Vendor: The name of the vendor of the product
- Publisher: Publisher that would be recommended to use a product within a fashion blog post
- Color: Color of the product

4.5.2 Labels or target vectors

To forecast fashion, the problem was investigated using three approaches, by forecasting sales, number of clicks and popularity. Therefore, the datasets were labeled with three different labels:

- Sales:Sales amount aggregated by each product converted into currency EUR
- Click Count:Click count by each product
- Popularity:Popularity ration by each product. Popularity was a ratio found within the original dataset, calculated by Apprl for each distinct product.

4.5.3 Cleaning the data

- Every data point was unique and all duplicates were removed by using excel macron, duplicates added no value to the learning performance.
- Missing data entries in the data were removed from the final selected datasets.
- Outliers i.e. values that were out of range were removed because these can dominate learning models' performance. Outliers were detected when data was plotted using histograms or scatter plots.

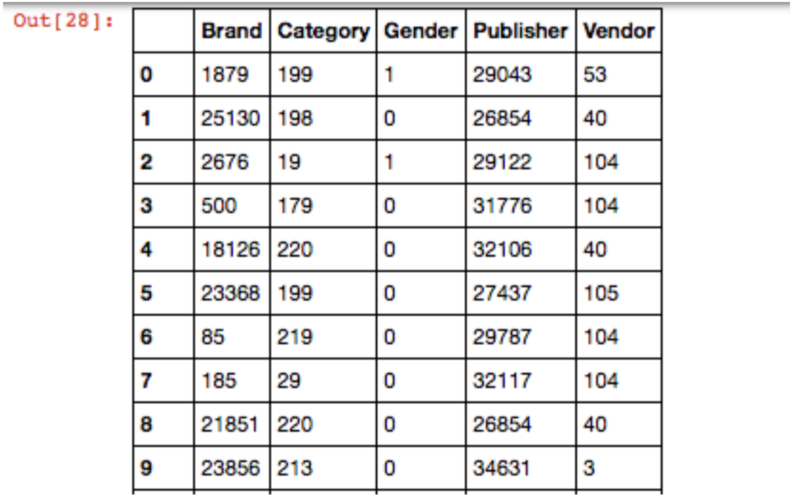
4.5.4 Transformation of the features

The datasets datatype was categorical and nominal. Categorical data types are for example color values and vendor names. Nominal data have no order and thus only gives names or labels to various categories for example, category ids, brand ids and gender. Sensitive information found in the dataset was anonymized in accordance with ethical guidelines. Since the features were nominal, they could neither be scaled, nor normalized. Further, the feature vectors extracted from the database were all nominal (brand id, category id, publisher id, vendor id, etc.) except the column gender that was in string format. This column was transformed into nominal values by assigning gender male zero, the gender female was assigned value one and unisex gender was assigned value two, after the extraction step (extraction step is shown in figure 4). In figure 5, an illustration of a small dataset after feature extraction and transformation of the gender column can be found.

4.5.5 Normalization and discretization of the label vectors

The labels were extracted from the database during query time together with the feature sets. Each label vector was normalized using a statistical

Figure 5: Data after transformation



Out[28]:

	Brand	Category	Gender	Publisher	Vendor
0	1879	199	1	29043	53
1	25130	198	0	26854	40
2	2676	19	1	29122	104
3	500	179	0	31776	104
4	18126	220	0	32106	40
5	23368	199	0	27437	105
6	85	219	0	29787	104
7	185	29	0	32117	104
8	21851	220	0	26854	40
9	23856	213	0	34631	3

formula called z-score:

$$z = \frac{x - \mu}{\sigma}$$

where:
z: the normalized value for each label in target vector
μ: the calculated average value of the target vector
σ: the calculated standard deviation value of the target vector.

After normalization of each label vector, a median value was calculated for the normalized label vector. Finally, if the normalized value was bigger than the normalized median, it was assigned “1” otherwise zero “0”.

For example, in table 2, each feature were labeled with either 0 or 1 depending on the results from the normalization and discretization step:

Table 2: Label Example

Label	Brand	Category	Vendor	Color
0	11	4	32	90
1	3	89	90	23
0	65	7	19	10

4.5.6 One-Hot-Encoding

Decision Trees (DT), Random Forests (RF), Adaboost (AB) and K Nearest Neighbors (KNN) classifiers are known to predict well on categorical data. These classifiers should handle categorical data encoded as numbers. However, since linear learning models are not developed to separate ordered from unordered numerical values, some classifiers such as Logistic Regression (LR) and Stochastic Gradient Descent (SGD), required encoding features into binary numbers. The encoding was necessary according to David. To solve the encoding issue, Scikit Learn library recommended using One-Hot-Encoder (OHE) to enhance the performance of these estimators. OHE transforms each categorical feature with "n" possible values into "n" binary features, with only one active (David, n.d.).

Even though this is the recommended and the common step, OHE was impractical to use within this degree project. The frauds of using OHE were the increased dimensionality of the data. For example, the dimensionality extended from five features into about 900 features for the Click dataset. This led to memory errors and time-consuming computations. To solve the dimensionality issue, it was then suggested by David to use principal component analysis, but this worked mainly on small datasets. Moreover, OHE was not used within this degree project because it was unfeasible for several reasons. For example, once the feature vectors were transformed into binaries, it was not possible to revert them back to the original values. Also, even if OHE was tested during this degree project, the results have shown that the linear models performed worse than other classification learning models i.e. the learning models that did not required data to be transformed using OHE.

4.5.7 Partitioning datasets

In this degree project, datasets have been partitioned into training and testing datasets by using two evaluation methods. The first method applied was a random split method: on different training dataset sizes. The second method applied was K-Fold Cross Validation using different values for K. The learning models were evaluated on the test datasets. These methods were used in order to get the best testing dataset. According to Bishop the best testing dataset would be a dataset that contains variables never seen by the learning models during training time. However, the datasets were partitioned using Scikit Learn's random split function: *train and test split method* which looks like this:

```
train_test_split(features, labels, test_size=0.2, \
        random_state=48)
```

where:

- features: represents the feature set, a 2-dimensional array with the shape (nr_samples, nr_features,),
- labels: was the target/label vector, a 1-dimensional array with the shape (n,)
- test_size: was assigned a fraction for example 0.2
- random_state: a randomness level of sampling data. For example, random state = 48

This function returns a training dataset with corresponding training labels and a testing dataset with corresponding testing labels. Further, the machine learning models were evaluated on different training sizes to evaluate the accuracy performance. However, within this project, it was interesting to evaluate the learning models by experimenting with different testing sets. Therefore, the data was partitioned using a fraction range from 0.1 to 0.9.

Nevertheless, using the Random Split method resulted in behaviors as shown in for example figure 6. This figure shows that in many cases the results were not improving for larger training data, in fact they sometimes got worse. This should not be possible if the training data and test data were representative, thus the method was not sufficient. Fortunately, the high variance estimate that was provided by this method, could be avoided by using K-fold cross validation (for a theoretical description see the evaluation section 2.4.1). Therefore, K-fold cross validation evaluation method was used instead as shown in figure 7. The K was set equal to eleven to find the best training and testing dataset.

Figure 6: Average accuracy by each classifier with increasing training set size

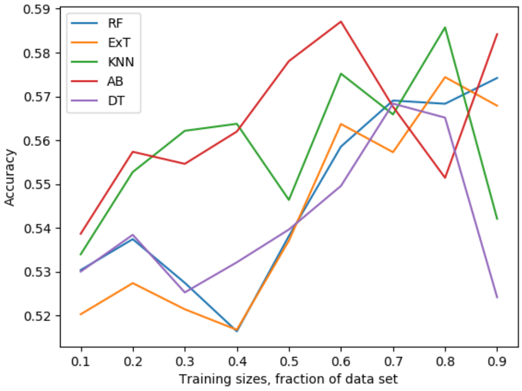
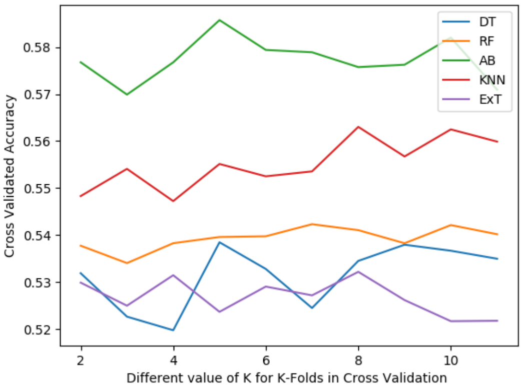


Figure 7: Average accuracy by each classifier with increasing number of K folds



4.6 Algorithm selection

According to the previous work, fashion forecasting has not been studied using classification. Therefore, the choice of classification algorithms has not been easy and was a critical part of this degree project. There was a large amount of learning models to choose between, some of them were applied and evaluated within this degree project.

The classifiers investigated within this degree project were implemented by Scikit Learn. The author of this master's thesis has chosen to work with Scikit Learn library since it is commonly used by Data Scientist as well as commercially (David, n.d.). Some of the steps of classification were easy to generalize as shown below in the function classify.

```
def classify(clf, x_train, y_train, x_test): \label{clf}
    model = clf.fit(x_train, y_train) # Train classifier
    y_predicted = model.predict(x_test) # Predict data
    return y_predicted
```

where:
clf: a classifier imported from scikit library
x_train: training features
y_train: training labels
x_test: testing features
y_predicted: predicted labels

Classifiers used within this degree project (described in section 2.3.1-2.3.4) were initiated by firstly importing them and secondly by calling them. Furthermore, all classifiers were put into an array for simplicity. For instance:

```
def create_classifiers():
    names = ['DT','ExT','AB','RF','KNN','LR','SGD']
    clf = ['DescisionTreeClassifier()',
          'ExtraTreeClassifier()',
          'AdaboostClassifier()',
          'RandomForestClassifier()',
          'KNeighborsClassifier()',
          'LogisticRegression()',
          'SGDclassfier()']
    return clf, names
```

Within a main method, a call to the method classify was made by passing the array of classifiers, the training data and the testing data as arguments. The classifiers were trained and evaluated sequentially.

```
def main():
  # import data from database
  # preprocess data
  # clfs, names = create_classifiers()
  kfolds = [2,3,4,5,6,7,8,9,10,11]
  sizes = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
  RUNS = 10
  for clf, name in zip(clfs, names):
      for fold in k-folds: # or size in sizes for random split
          for run in RUNS:
              # partition data using Cross Validation or
              # or partition using Random Split
              classify(clf, x_train, x_test)
              # calculate F1 Score
      total_score = score / RUNS
```

The evaluation was made by calculating the average accuracy F1-score as described in the background (For more technical details see Scikit Learn Library's home page).

4.7 General notes on Scikit Learn Classifiers

Cross validation was used to get the best parameters for KNN and Adaboost. For other classifiers, the default values defined by Scikit Learn were used. The final parameters used for each classifier can be found in detail in appendix A.

Decision Tree

For the Decision Tree classifier, Scikit Learn use an optimized version of the CART algorithm. CART is the abbreviation for classification and regression trees and similarly implemented as C4.5. CART supports numerical target variables and does not compute rule sets (Loh, 2011). Further, CART constructs binary trees using the feature and threshold that yield the largest information gain at each node (Loh, 2011).

Ensemble Learning models

The Scikit Learn's implementation of ensemble learning models such as Random Forest and Adaboost, combines classifiers by averaging their probabilistic predictions, instead of letting each classifier vote for a single class.

Extremely Randomized Trees (ExT) classifiers, is another example of ensemble learning models implemented by Scikit Learn and used by this degree project. According to Scikit Learn, both Random Forest and ExT use a random of subset of candidate features, but the difference between these ensemble models is when ExT uses randomness to choose the best splitting rule while Random Forest looks at the most discriminative threshold when choosing splitting rules. Further, ExT use averaging to improve the predictive accuracy and to control the overfitting.

K Nearest Neighbors

Within Scikit Learn, the parameters of KNN could be tuned in different ways. For instance, the parameter for weights could be set to 'uniform' which means that all points in each neighborhood were weighted equally. The distance metrics for the tree are by default set to the value: "minkowski" and the K of KNN was set to a default value: K = 5.

Linear Models

Within Scikit Library, Logistic Regression has defined parameters such as penalty that can be set dependently on the goal of the classification. These parameters are supposed to optimize the performance of the algorithm. The implementation of SGD is influenced by the Stochastic Gradient SVM of Bottou. The class SGD Classifier implements a plain stochastic gradient descent learning routine which supports different loss functions and penalties for classification. However, these two linear models were tested and predicted very bad, therefore they were not included in the result section. The drawbacks of these linear models are described in section 4.5.6. The datasets within this degree project, are nominal and these linear models predict best on numerical ordered datasets according to Scikit.

5 Results

To evaluate the method, several datasets and algorithms will be presented. Each section will present a dataset description and the performances of each algorithms trained and tested on the data.

Within this degree project, each dataset was run through the code sequentially. Each dataset was run through each algorithm/classifier mentioned in method chapter. The classifiers are given abbreviations and will be used both in figures and texts for simplicity. The abbreviations are as follow:

- Decision Tree (DT)
- Random Forest (RF)
- Extremely Randomized Tree (ExT)
- K Nearest Neighbor (KNN)
- Adaboost (AB)
- Logistic Regression (LR)
- Stochastic Gradient Descent (SGD)

Further, the datasets were partitioned using two methods: K-fold Cross Validation, and a random splitting method as described in section 2.4. The performance is evaluated by calculation F1 score, observe that F1-score is referred to as accuracy.

The final datasets with selected features can be found summarized in table 3. The Name column presents the name of the datasets. Each dataset has two or more features (columns), as shown in table 3, where 'yes' indicates that the feature is included in the dataset, otherwise 'No' meaning that the feature is not included. The size of each dataset is given in number of feature sets (i.e. rows).

Table 3: Dataset summary

Name	Category	Brand	Gender	Color	Publisher	Vendor	Size
Sales A	Yes	Yes	No	No	No	No	1896
Sales B	Yes	Yes	No	No	No	Yes	2039
Sales C	Yes	Yes	Yes	No	No	Yes	2026
Sales D	Yes	Yes	Yes	No	Yes	Yes	3851
Sales E	Yes	Yes	Yes	Yes	Yes	Yes	3532
Click A	Yes	Yes	No	No	No	No	23 064
Click B	Yes	Yes	No	Yes	No	No	63 641
Click C	Yes	Yes	No	Yes	No	Yes	63 641
Popul B	Yes	Yes	No	No	No	No	122 124
Popul A	Yes	Yes	Yes	No	No	Yes	124 854

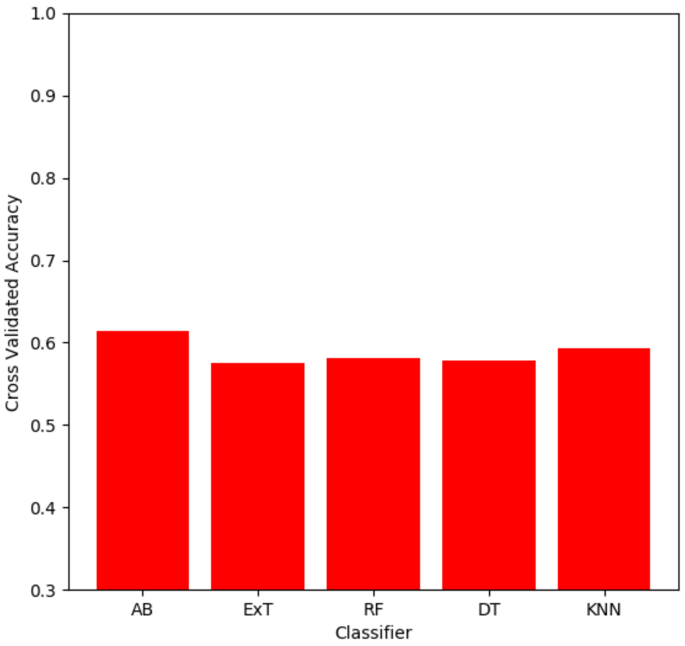
5.1 Click Datasets

Within this section, the datasets were labeled with clicks, i.e. if a product was clicked with more than the median it was labeled with 1 otherwise with 0.

5.1.1 Click dataset A

In figure 8, we can see that the best performing learning model on *Click dataset A* was AB. It predicted with highest cross validated accuracy (61 %).

Figure 8: Cross Validated Average accuracy by each classifier



When comparing the results of the two splitting methods (K fold cross validation and random split method) we can see in figure 9 that with increasing size of training dataset, the accuracy increases for all learning models. By using cross validation evaluation method, in figure 10 a slight increase in accuracy can be seen that is caused by the incremented number of K-folds.

Figure 9: Average accuracy by each classifier with increasing training dataset size

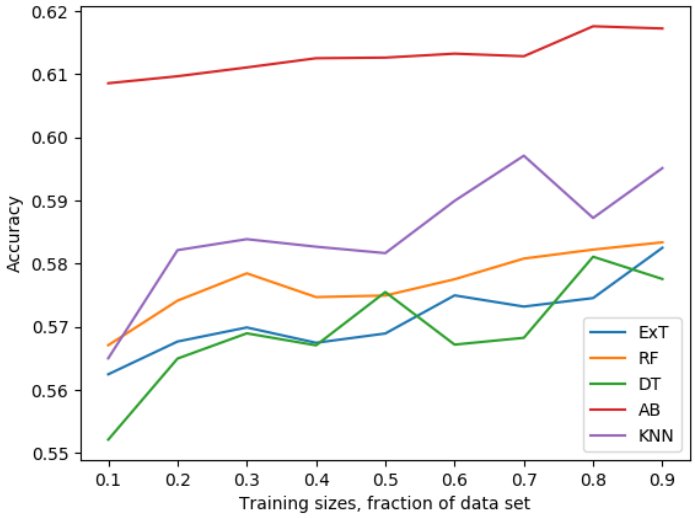
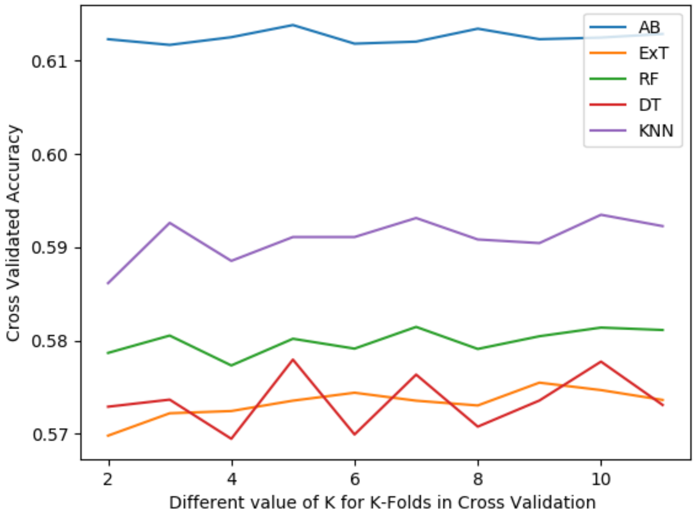


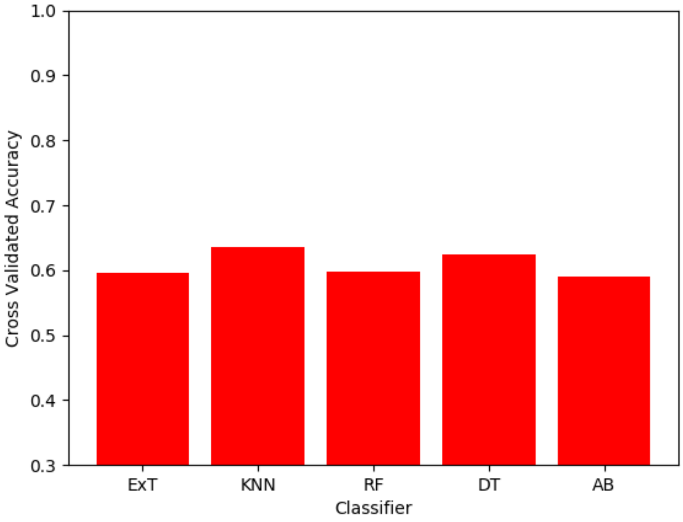
Figure 10: Cross Validated Average accuracy by each classifier with increasing K, i.e. number of folds



5.1.2 Click dataset B

Click dataset B is larger than *Click dataset A* in number of samples and features. KNN prediction accuracy was the best (64%) followed by DT (62 %), as can be seen in figure 11.

Figure 11: Cross Validated Average accuracy by each classifier



Both figure 12 and figure 13 shows a normal behavior: the models performs better when trained on larger training sizes, and cross validated with larger K. However, AB, RF and ExT learning models had low accuracies.

Figure 12: Average accuracy by each classifier with increasing training dataset size

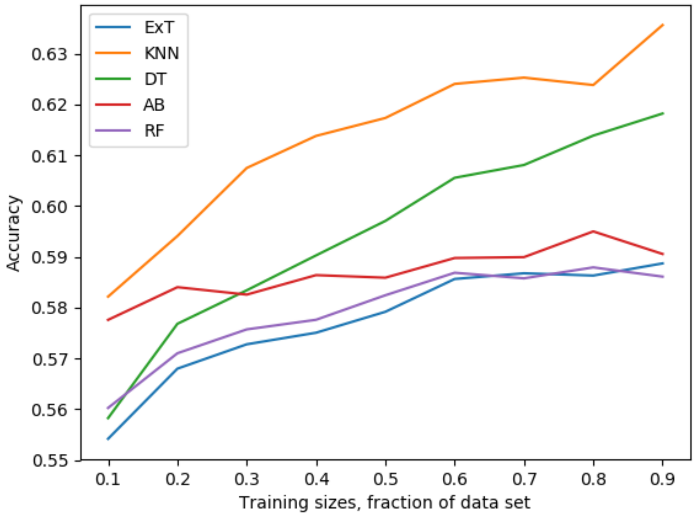
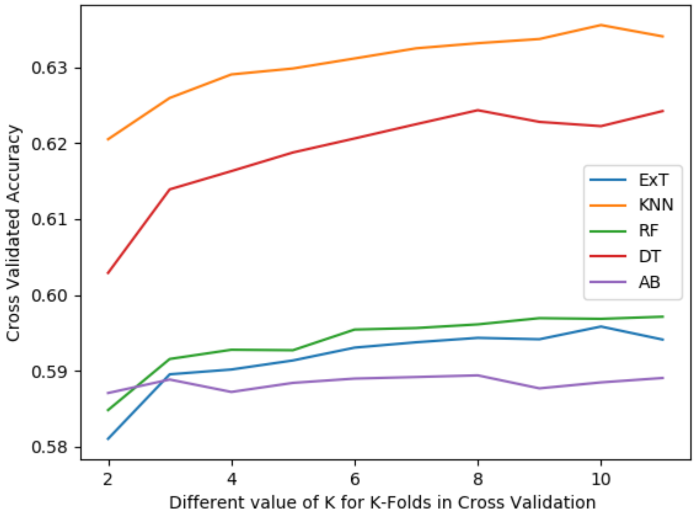


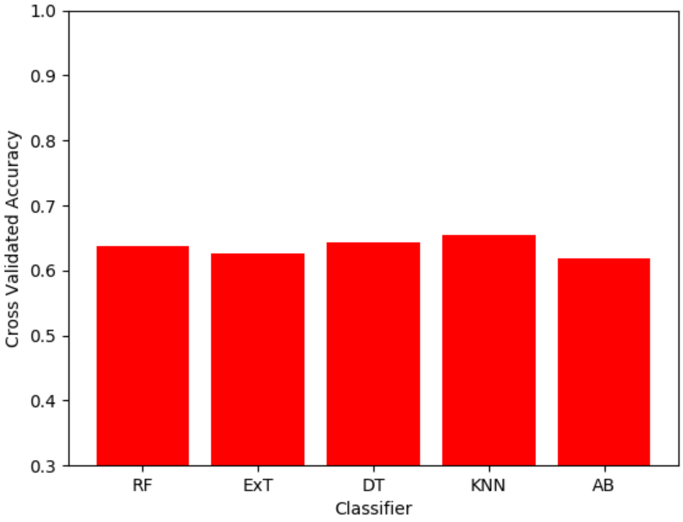
Figure 13: Cross Validated Average accuracy by each classifier with increasing K, i.e. number of folds



5.1.3 Click dataset C

Click dataset C, in comparison to *Click dataset B*, had one more feature (vendor). Figure 14 shows that the best performing classifier was KNN with $\mathbf{K} = \mathbf{5}$ and a test cross validated accuracy of about 66 %.

Figure 14: Cross Validated Average accuracy by each classifier



In figure 15, we can see that the accuracy of the models increase while the size of the training data increase and that the Adaboost is performing suspiciously. In figure 16, shows that when using cross validation, smoother trends were achieved than when using random split method.

Figure 15: Average accuracy by each classifier with increasing training dataset size

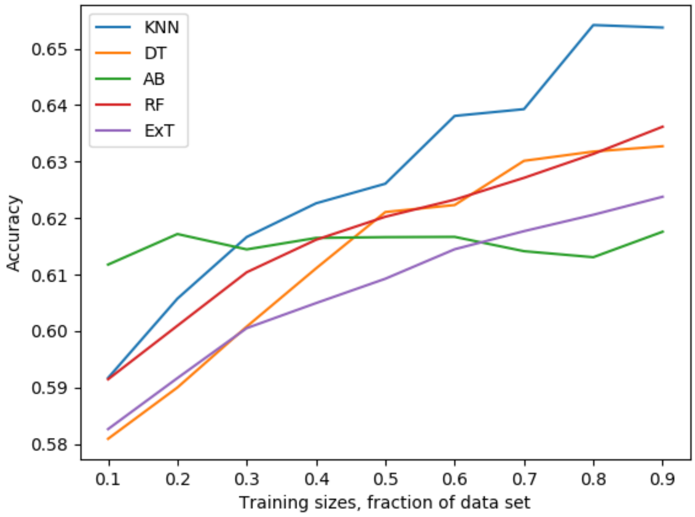
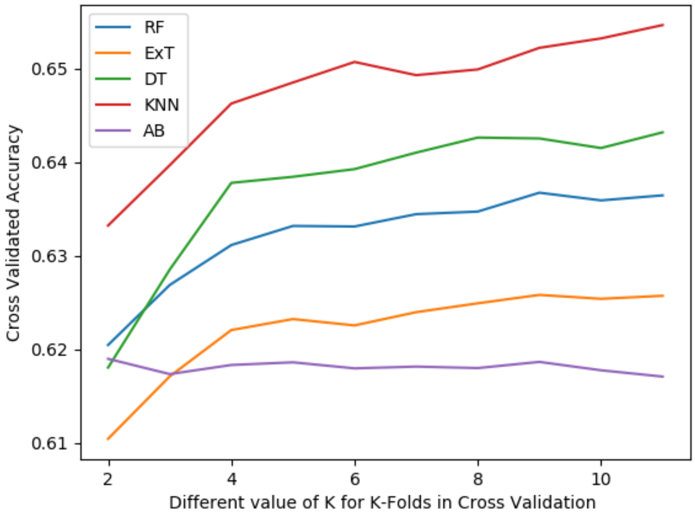


Figure 16: Cross Validated Average accuracy by each classifier with increasing K, i.e. number of folds



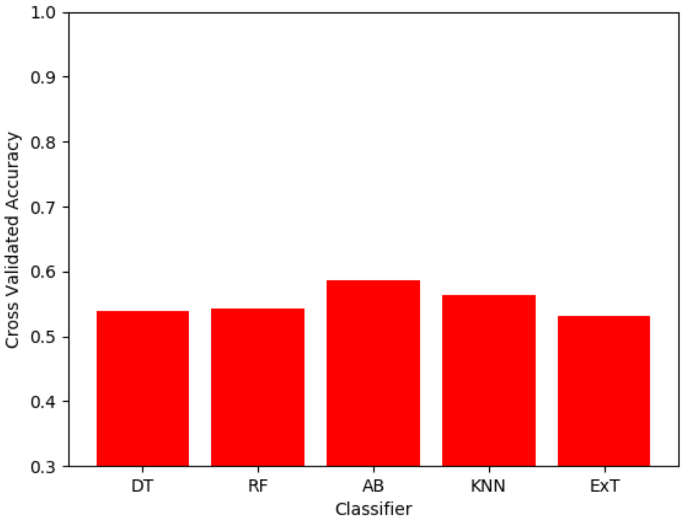
5.2 Datasets labeled with sales

Within this section, the datasets were labeled with sales, i.e. if a product was selling better than the median it was labeled with 1 otherwise 0.

5.2.1 Sales dataset A

Sales dataset A is the smallest dataset that have been studied within this degree project. It contains 1895 unique feature vectors and two features, category and brand. The accuracy 59% was the best test cross validated accuracy yielded by AB, followed by 56 % in second place by KNN as can be seen in figure 17.

Figure 17: Cross Validated Average accuracy by each classifier



In figure 18, using random split method, we can see that AB has a maximum accuracy of 62 %, and that the performance decreases with decreasing training size. However, all other learning models acts strange and performs bad: the performance increases with decreasing training set size. Therefore, this was evaluated by using cross validation as well. Figure 19 shows that the best achieved result was no longer 62 % for AB but 59 % when number of K folds where equal to five.

Figure 18: Average accuracy by each classifier with increasing training dataset size

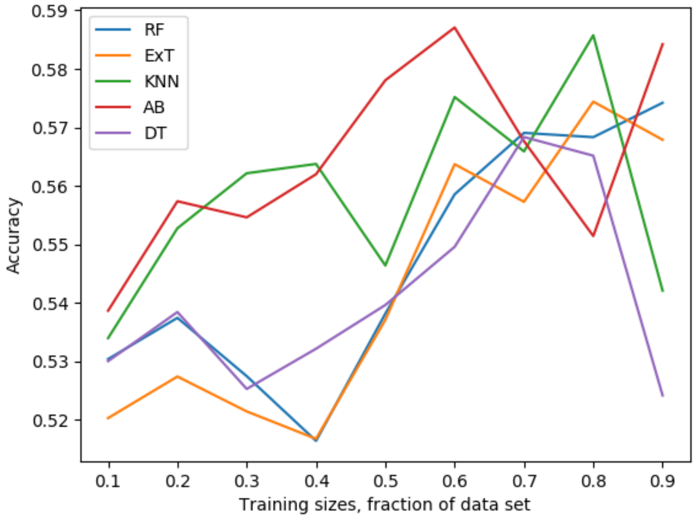
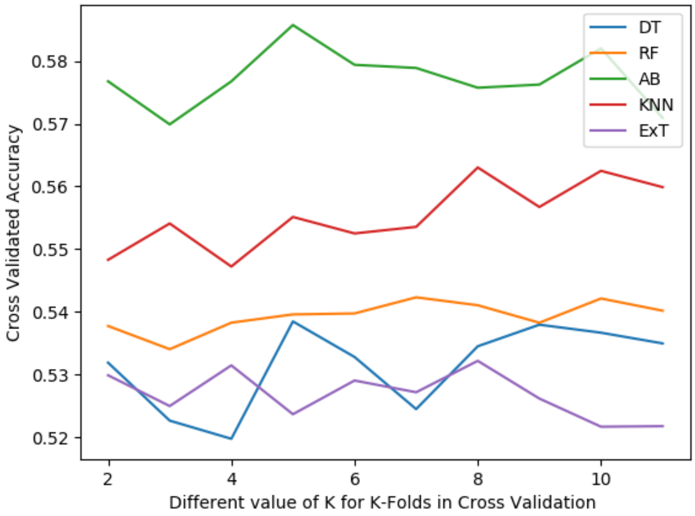


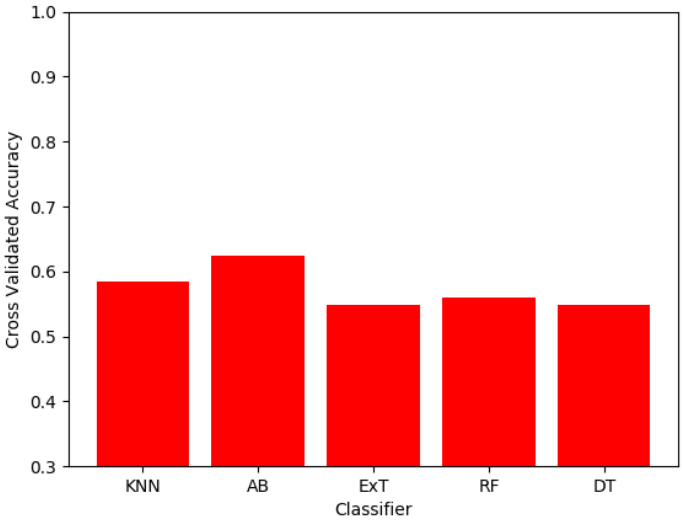
Figure 19: Cross Validated Average accuracy by each classifier with increasing K, i.e. number of folds



5.2.2 Sales dataset B

Sales dataset B was best predicted by AB, with a test cross validated accuracy of 63 %. KNN predicted with 58 % and RF with 56 % accuracy, which can be seen in figure 20.

Figure 20: Cross Validated Average accuracy by each classifier



In figure 21, AB peaks in performance at a 80 % training data size with exactly 62 % accuracy. In figure 22, an increase can be seen when the number of folds is incremented, and for AB the best K was nine.

Figure 21: Average accuracy by each classifier with increasing training dataset size

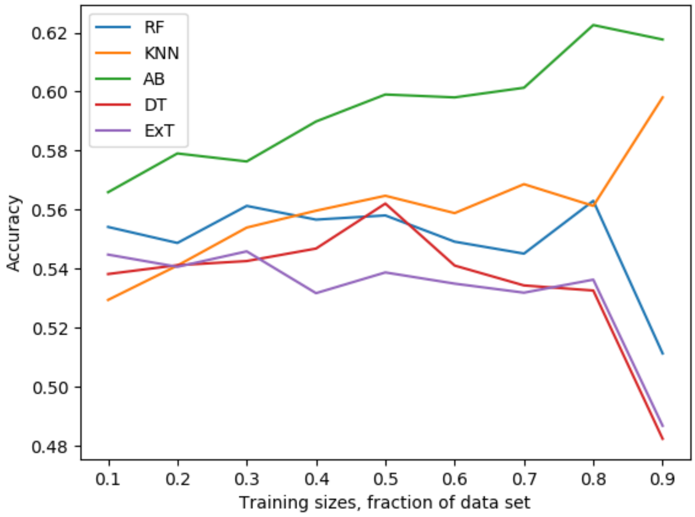
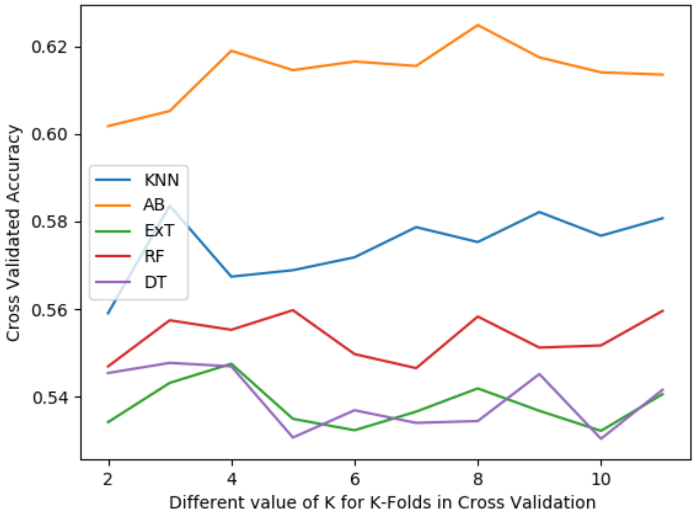


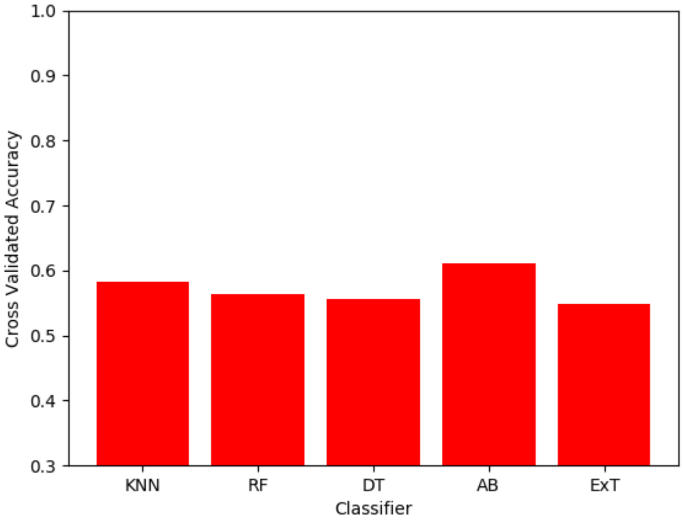
Figure 22: Cross Validated Average accuracy by each classifier with increasing K, i.e. number of folds



5.2.3 Sales dataset C

Sales dataset C was, again, best predicted by AB, with a test cross validated accuracy of 61 %. KNN predicted with 58 % and RF with 56 % accuracy, which can be seen in figure 20.

Figure 23: Cross Validated Average accuracy by each classifier



In figure 24, shows suspicious trends. However, in figure 25, the high variance that was introduced by the random split method is reduced. An very small increase can be seen when the number of folds is incremented and for AB the best K was six.

Figure 24: Average accuracy by each classifier with increasing training dataset size

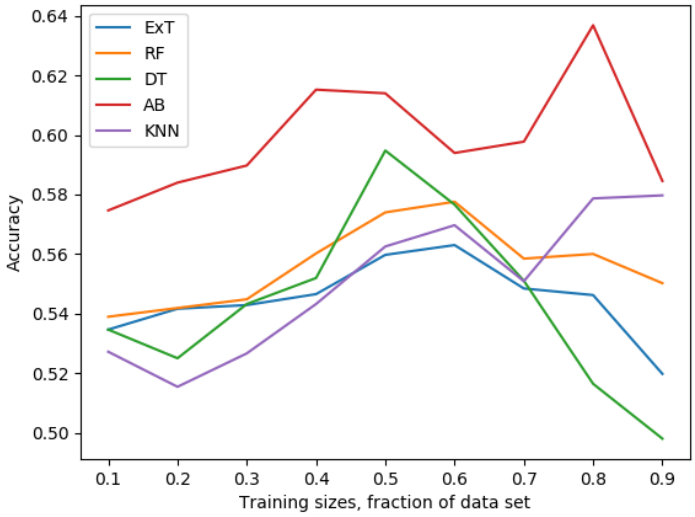
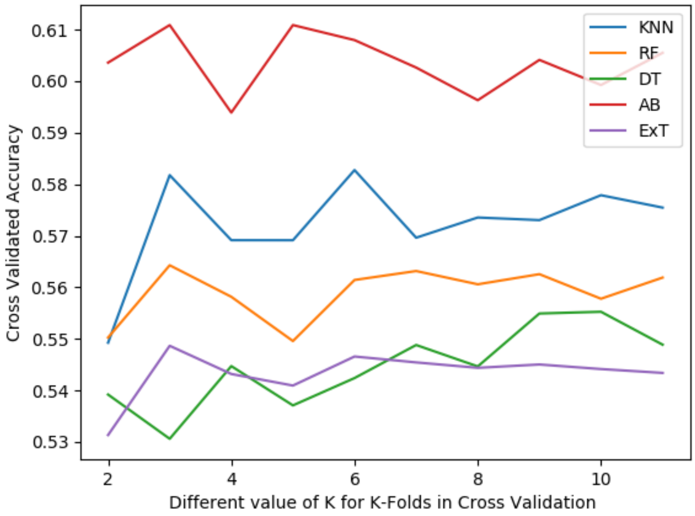


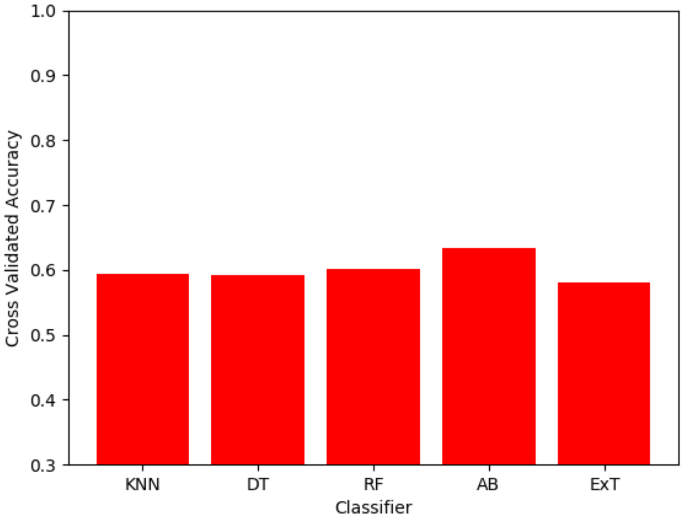
Figure 25: Cross Validated Average accuracy by each classifier with increasing K, i.e. number of folds



5.2.4 Sales dataset D

Sales dataset B was best predicted by AB, with a test cross validated accuracy of 64 %. RF predicted with 60 % and KNN with 59.8 % accuracy, DT with 59 % and at last ExT with a cross validated accuracy of 58 % (see figure 20).

Figure 26: Cross Validated Average accuracy by each classifier



In figure 27, AB peaks in performance at a 70 % training data size with an accuracy of 62.5 %. However, in figure 28, an increase can be seen when the number of folds is incremented, and for AB the best accuracy was achieved when K was set to six.

Figure 27: Average accuracy by each classifier with increasing training dataset size

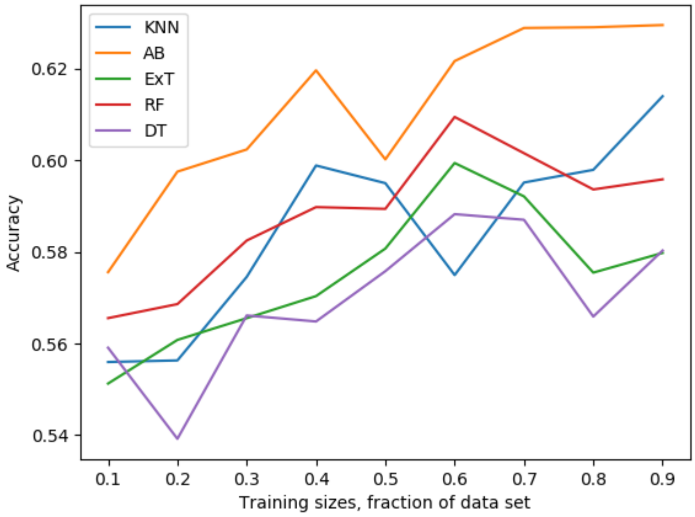
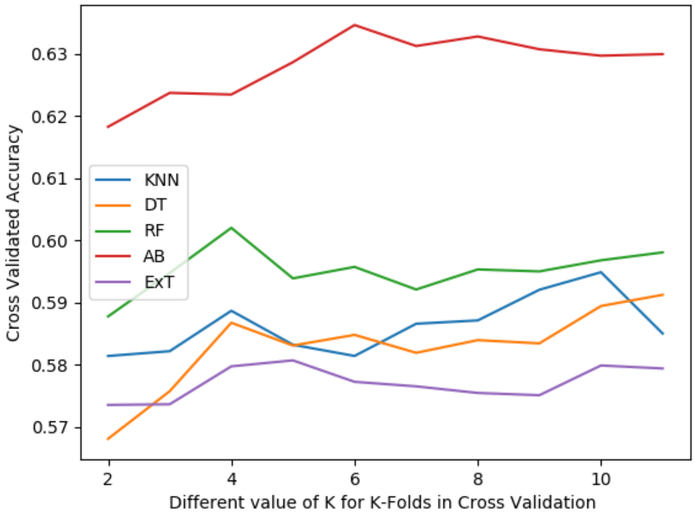


Figure 28: Cross Validated Average accuracy by each classifier with increasing K, i.e. number of folds

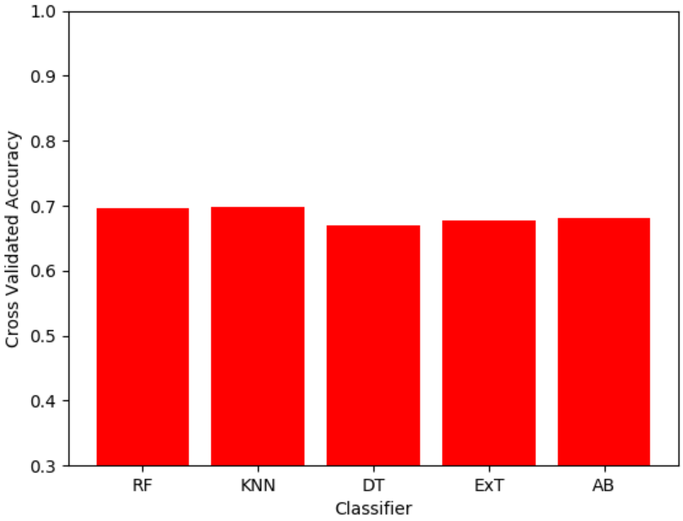


5.3 Popularity datasets

5.3.1 Popularity dataset A

Popularity dataset A was the largest studied dataset within this degree project. It was best predicted by KNN, with a test cross validated accuracy of 70 %. RF predicted with 69.6% and AB with 68 %. The rest had an accuracy between 67.1 and 67.7 % accuracy, which can be seen in figure 29.

Figure 29: Cross Validated Average accuracy by each classifier



In figure 30, KNN peaks in performance at a 90 % training data size with 69.3 % accuracy. In figure 31, an increase can be seen when the number of folds is incremented, and for KNN, it gets even better when the K is larger than five. The best accuracy for KNN was achieved when K was set to seven. The worst performing model was DT.

Figure 30: Average accuracy by each classifier with increasing training dataset size

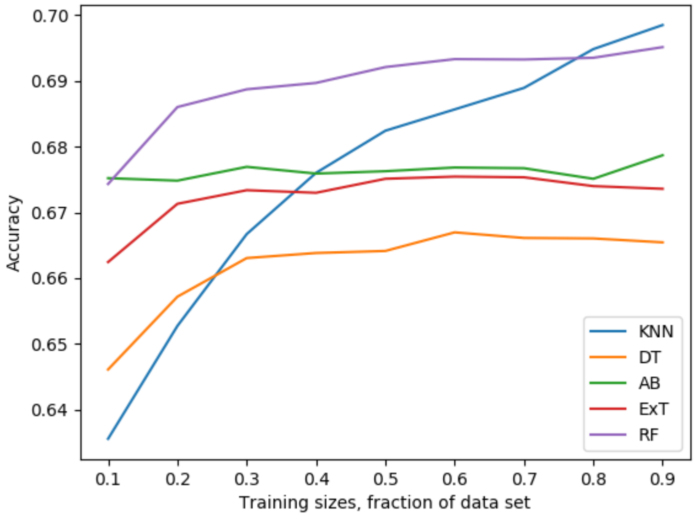
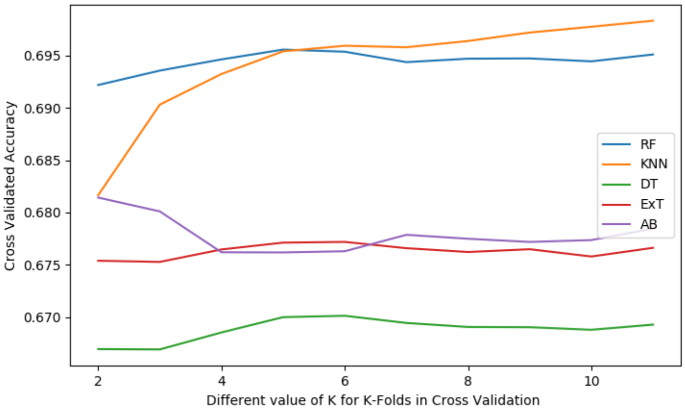


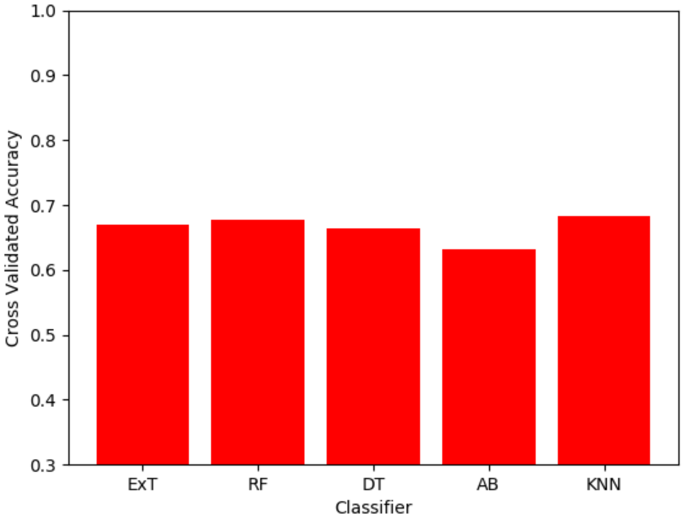
Figure 31: Cross Validated Average accuracy by each classifier with increasing K, i.e. number of folds



5.3.2 Popularity dataset B

Popularity dataset B holds the basic features category and brand. It was best predicted by KNN, with a 68.3 % test cross validated accuracy. RF predicted with 67.7 % accuracy and ExT with 66.8 % accuract (see figure 32).

Figure 32: Cross Validated Average accuracy by each classifier



In figure 33, all models seem to act normally except AB, that is performing worse and does not change performance when decreasing the training dataset which is strange. In figure 34, an increase can be seen when the number of folds is incremented. However, AB was the worst performing learning model even when using cross validation with different K.

Figure 33: Average accuracy by each classifier with increasing training dataset size

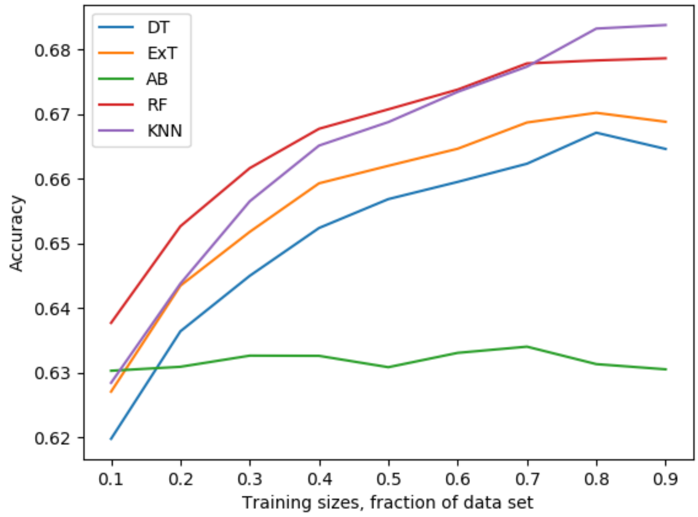
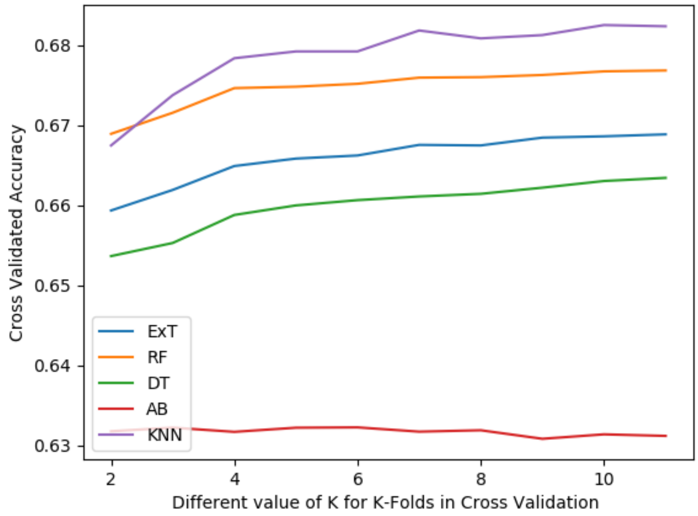


Figure 34: Cross Validated Average Accuracy by each classifier with increasing K, i.e. number of folds



6 Discussion

Within this master's thesis, the main difference from the state of the art (previous work) was that Artificial Neural Network (ANN) was not possible to apply on the investigated datasets, since the sales datasets were very small.

6.1 The Click datasets

The click datasets were binary and labeled with clicks where class 0 represented products that generates a small amount while class 1 generates a large number of clicks. The goal with Click datasets was to predict most clickable products and it meant products that generated more than three clicks. The experiments were conducted on three different combinations of feature vectors and will be discussed below.

- Discussion on *Click data A*, section 5.1.1
  Click dataset A was the most basic Click dataset since it contained only two features, the product category and the product brand. Training models on this dataset resulted in a maximum accuracy of 61 %. The AB classifiers performed best, followed by KNN. Looking at figure 9, AB is mostly unchanged no matter training size. The strange behavior of the AB can be explained by its nature to find complex boundaries from small training sizes. In general, KNN performed better than AB, as shown figure 9, where the accuracy decreased accordingly with decreasing training set size. However, this dataset was not large enough to draw any conclusions on, and was therefore discarded.

- Discussion on *Click data B and C*, section 5.1.2 - 5.1.3
  Both Click dataset B and C consisted of large amounts of data: 63 641 feature vectors. The goal with these two datasets was to find the best feature combination and the results have shown that the feature set that achieved highest accuracy consisted of the features category, brand, color and vendor of a product. Both Click dataset B and C have shown in figures 12 and 15 smooth decreasing trends while having a decreasing amount of training data. In figures 13 and 16 have shown smooth increasing trends when the number of iterations i.e. K-folds increased. Therefore, conclusions on most clickable products could be best drawn by predicting on products with the characteristics color, brand, category, and vendor with a cross validated average accuracy of 66 %.

6.2 The Sales datasets

The Sales datasets described in section 5.2 were shamelessly small. The datasets were labeled with sales where class 0 represented the worst selling products and the label 1 represented the bestselling products. The goal with these datasets was to predict the most sellable products.

- Discussion on *Sales data A, B and C*, section 5.2.1-5.2.3
  These datasets prediction accuracies were surprisingly good (59 % - 61 %). Since the datasets were very small it is difficult to draw any conclusions. Especially since figure 21 and 24 have shown that the datasets were noisy i.e. not representative. Therefore, these datasets were discarded. However, the results indicated that the cross-validation evaluation method were more likely to give less noisy results than the random split method. Worth noting is that the range on the Y-axis is quite small, meaning that the strange fluctuations observed were often smaller than 2 %

- Discussion on *Sales data D*, section 5.2.4
  Sales dataset D includes the features brand, category, gender, and vendor. However, with this dataset the goal was to find out whether it was possible to predict most sellable product by taking publishers into account. The cross validated accuracy achieved was 64 % by the learning model RF. This increase in accuracy is somewhat biased, since it was found that some of the publishers were overrepresented within the dataset.

6.3 The Popularity dataset

- Discussion on Popularity data A and B, section 5.3.1, 5.3.2 Popularity dataset B consisted of the two basic features category and brand. The learning models' performance were all good with an accuracy above 65 %. The best performing learning model on Click dataset B was KNN which yielded an accuracy of 67.7 %. Since this dataset resulted in good performance considering only two features were used, it was interesting to study the popularity dataset with more features. Therefore, the popularity dataset A was investigated once more by adding two more features, gender and vendor.

  The Popularity dataset A was the largest dataset studied within this degree project with approximately 124 000 unique feature vectors. The goal with these popularity datasets was to investigate if it was possible to predict most popular products. The popularity ratio was given by Apprl. Surprisingly, this dataset resulted in the highest test cross validated average accuracy within this degree project. The accuracy yielded by the learning model KNN was about 70 %. Since the dataset was large enough, the learning models could be trained on larger training datasets in comparison to Sales datasets. For example, in figure 31, KNN's learning curve decreases from 70 % to approximately 63 % when the training data size decreases with 90 %. The accuracies were still high even though having smaller fractions of training data sizes, because the size was still larger than for example an entire sales dataset. With Popularity dataset, it is possible to predict most popular products using features such as product brand, category, gender and vendor.

6.4 The best and the worst classifiers

Since very little academically research have been investigating classification learning models' performance on fashion datasets, this degree project investigated several of them. The goal was to find out which classifiers that performed best. The weakest classifiers amongst all that were studied in this degree project were the classifiers SGD and LR. These learners predicted worst and were not included in the result section. The reason why these models performed bad was explained in detail in section 4.5.6. In short, the data type of the datasets investigated within this degree project was categorical data and these weak learners were not developed to support categorical data. There were alternative ways to overcome these problems e.g. by using One-Hot-Encoder (OHE). Even though the data was transformed into binary using OHE the weak learners did not perform better, and these were therefore excluded from the results

However, rule based learning models and ensemble models were known to perform better on categorical data. Therefore, the focus has been in investigating them. In the results, AB and KNN was the best performing classifiers. Adaboost performed good since it is developed to draw complex decision boundaries even in small datasets. This could be clearly seen in figure 9 and figure 10, for example. In my opinion and based on the results, KNN performed in overall best. Studying KNN in the results, the figures have shown clear flu trends fluctuating in comparison to rule based models, such as DT, RF, AB and ExT. For example when training sizes decreased the performance for KNN decreased, and when the number of K folds increased the performance of this models increased as well (see for example figure 12, 13, 16, 24).

6.5 Ethical Aspects

This master's thesis touched several ethical aspects which will be discussed in this section. To achieve a successful forecasting model, it was necessary to collect initial data, holding all kind of information. To fulfill the ethical guidelines, sensitive information that was found in the dataset was therefore anonymized by transforming the data into digits, which limit the risk of potential harm. Furthermore, there were several other ethical concerns that this master's thesis touched, such as the role of machine learning within the fashion industry. Is it ethical that computers replace humans by automating the process of curating contents and predicting fashion? There is no easy answer to the question. However, there are risks that machine learning models can predict worse for instance when looking at minority groups. These bad predictions occur due to unbalanced datasets. Nevertheless, the author of this master's thesis believes that machine learning could be used to complement the humans. Fashion creators could take advantage from using machine learning by letting computers do the heavy computations and find patterns. While machine learning does the heavy work, fashion creators could benefit from focusing on fine tuning the process by for example complementing with their experiences of fashion making. The author of this master's thesis believes that machine learning could provide general modeling and guidelines.

6.6 Sustainability

Within this master's thesis, one of the most important sustainability aspects was the economical aspect. As studied in previous work, the authors Au et al. (and others) found that retailers are very dependent on fashion forecasting systems. he retailers needs accurate fashion forecasting system in order to maintain and balance the products in stock.

Since the fashion industry is known to be volatile, the retailers could either have a stock full of products in case of low demand, or be out of products in case of high demand. This of course leads to economic losses in both ways. Having a technical solution for this such as using machine learning for predicting fashion, retailers can then adapt the products in stock to the market demand.

7 Conclusion

7.1 Conclusion and Criticism

To what extent is it possible to predict the most popular and unpopular products in terms of number of clicks, sales rates and popularity rates, using machine learning techniques and by leveraging APPRL's dataset?

As stated in the problem formulation, several machine learning techniques have shown that forecasting have been possible within multiple domains. For example, within image recognition, text opinions, stock prices and diseases. Though, fashion forecasting has not been investigated in extent and especially not using a classification approach. In the related work section, similar studies were presented. Most of these studies based their investigations on time series datasets using regression or unsupervised learning approaches such as extreme learning models Artificial learning networks. The main differences between this degree project experiments and related study was therefore the machine learning approach taken and datasets used.

To answer the research question adequately, it was necessary to experiment with several perspectives of Apprl's dataset using different supervised classification learning models. The dataset was therefore labeled with Clicks, Sales and Popularity. By experimenting with these different datasets, the results have shown that Sales datasets were not possible to draw any conclusions from since they were very small. Even though, different combinations of feature sets were used, the data was too noisy. The results have also shown that both Click datasets and Popularity datasets could be used to predict on since the data was less noisy and large enough. The high yielded accuracies have shown that these datasets contained a representative amount of information. Cross validation evaluation method eliminated was preferably used rather than the random split method since it decreased the amount of noise that was introduced by the random split method. The results have also shown that linear models, were not suitable to use within this degree project since these was developed to predict on other data types than categorical data. The best performing learning model was the KNN, and Adaboost. Adaboost was found to perform good even on smaller datasets, since it is developed to find complex decision boundaries.

The author of this master's thesis believes that the forecasts made in this degree project, in the context of curating content, could best be used as guidelines when curating contents. For example, by suggesting most popular products or most clickable products to digital publishers. Digital publishers can use these suggestions to fine tune its experiences and curating process. However, as shown in results and discussed, the predicted popular and most clickable products were not personalized. This step was tested by including the publisher feature in sales dataset, but it was later found that some of the publishers were more representative in the dataset. This biased on the learning models' performance.

The results have shown that it was possible to forecast fashion using Apprl's dataset with an average accuracy between 65% and 70%. The highest accuracy was achieved by predicting on Popularity dataset A. The best feature set contained the features category, brand, gender, color and vendor.

7.2 Contributions

The four main contributions of this thesis were:

- the conclusion that it was possible to predict fashion based upon Apprl's dataset.
- an automated curated shopping content that could be used by influencers.
- theoretical investigation of curated content and fashion forecast.
- preprocessed fashion dataset for future work.

7.3 Future work

Future work should focus on collecting larger sales datasets. Also, future work should focus on methods that already works well, such as ANN and deep learning, by improving them for fashion predictions. Furthermore, one could redefine the problem by using regression learning models instead of classification e.g. to estimate how many clicks on product can get or how big return a product can yield.

References

Abbott, D. (2008). Dcc briefing paper: What is digital curation?

Au, K.-F., Choi, T.-M., & Yu, Y. (2008). Fashion retail forecasting by evolutionary neural networks. *International Journal of Production Economics*, *114*(2), 615–630.

Benini, M. J., Batista, L. L., & Zuffo, M. K. (2005). When marketing meets usability: the consumer behavior in heuristic evaluation for web. In *Proceedings of the 2005 latin american conference on human-computer interaction* (pp. 307–312).

Bishop, C. (2007). Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn. *Springer, New York.*

Bottou, L. (n.d.). *Stochastic gradient descent.* Retrieved 2017-05-21, from `http://leon.bottou.org/projects/sgd`

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Bruns, A. (2009). News blogs and citizen journalism: New directions for e-journalism. *e-Journalism: New Media and News Media*, 101–126.

Choi, T.-M., Hui, C.-L., Ng, S.-F., & Yu, Y. (2012). Color trend forecasting of fashionable products with very few historical data. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), 1003–1010.

David, P.-S. C. f. D. S. N. C. U., Google. (n.d.). *About scikit.* Retrieved 2017-05-21, from `http://scikit-learn.org/stable/supervised_learning.html#supervised-learning`

Dearstyne, B. W. (2005). Blogs: the new information revolution? *Information Management*, *39*(5), 38.

Dellarocas, C. (2003). The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management science*, *49*(10), 1407–1424.

Deuze, M., & Bardoel, J. (2001). Network journalism: converging competences of media professionals and professionalism.

Dictionary.com. (n.d.). *Definitions.* Retrieved 2017-05-23, from `http://www.dictionary.com/browse/curated`

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87.

Friedl, M. A., & Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. *Remote sensing of environment*, *61*(3), 399–409.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, *28*(2), 337–407.

Halvorsen, K., Hoffmann, J., Coste-Manière, I., & Stankeviciute, R. (2013). Can fashion blogs function as a marketing tool to influence consumer behavior? evidence from norway. *Journal of Global Fashion Marketing*, *4*(3), 211–224.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264–323.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 6). Springer.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). *Supervised machine learning: A review of classification techniques.*

Kretz, G. (2010). Pixelize me!: A semiotic approach of self-digitalization in fashion blogs. *NA-Advances in Consumer Research Volume 37*.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(1), 14–23.

Lord, P., Macdonald, A., Lyon, L., & Giaretta, D. (2004). From data deluge to data curation. In *Proceedings of the uk e-science all hands meeting* (pp. 371–375).

Marsland, S. (2015). *Machine learning: an algorithmic perspective.* CRC press.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning (adaptive computation and machine learning series).* Cambridge, MA: MIT Press.

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *2*, 841–848.

Odden, L. (2012). *Optimize: How to attract and engage more customers by integrating seo, social media, and content marketing.* John Wiley & Sons.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Perlich, C., Provost, F., & Simonoff, J. S. (2003). Tree induction vs. logistic regression: A learning-curve analysis. *Journal of Machine Learning Research*, *4*(Jun), 211–255.

Post, E. L., & Sekharan, C. N. (2015). Comparative study and evaluation of online ad-blockers. In *Information science and security (iciss), 2015 2nd international conference on* (pp. 1–4).

Steffes, E. M., & Burgee, L. E. (2009). Social ties and online word of mouth. *Internet research*, *19*(1), 42–59.

Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, 37.

Thomassey, S. (2014). Sales forecasting in apparel and fashion industry: a review. In *Intelligent fashion forecasting systems: Models and applications* (pp. 9–27). Springer.

Wong, W., & Guo, Z. (2010). A hybrid intelligent model for medium-term sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm. *International Journal of Production Economics*, *128*(2), 614–624.

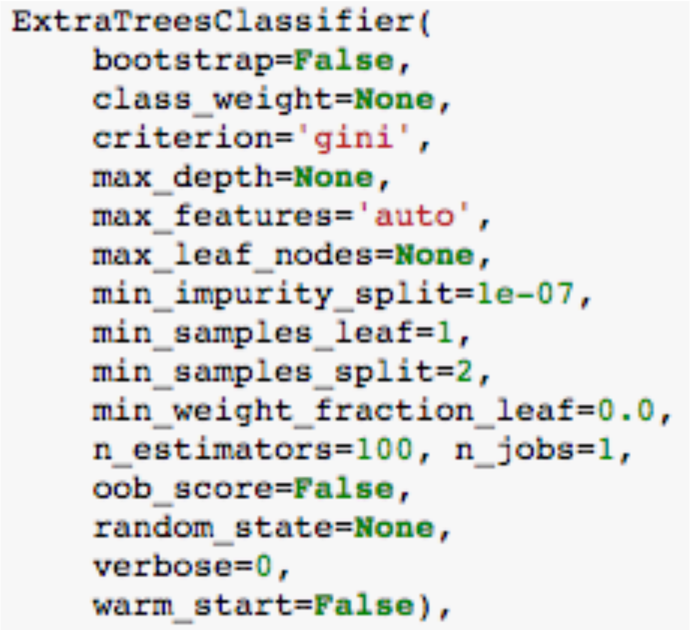
Xia, M., & Wong, W. (2014). A seasonal discrete grey forecasting model for fashion retailing. *Knowledge-Based Systems*, *57*, 119–126.

Xia, M., Zhang, Y., Weng, L., & Ye, X. (2012). Fashion retailing forecasting based on extreme learning machine with adaptive metrics of inputs. *Knowledge-Based Systems*, *36*, 253–259.

Appendices

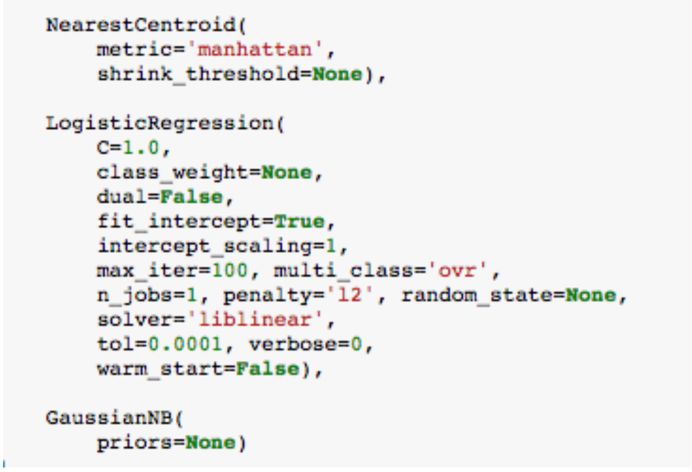
A Parameter Tuning

Figure 35: Parameters for chosen extra tree classifiers



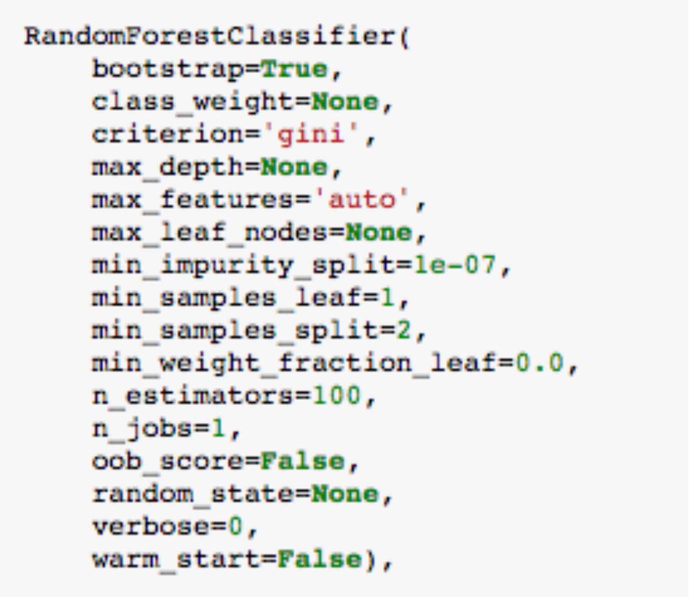
```
ExtraTreesClassifier(
    bootstrap=False,
    class_weight=None,
    criterion='gini',
    max_depth=None,
    max_features='auto',
    max_leaf_nodes=None,
    min_impurity_split=1e-07,
    min_samples_leaf=1,
    min_samples_split=2,
    min_weight_fraction_leaf=0.0,
    n_estimators=100, n_jobs=1,
    oob_score=False,
    random_state=None,
    verbose=0,
    warm_start=False),
```

Figure 36: Parameters for chosen Nearest Centroid, Gaussian NB and Logistic Regression classifiers



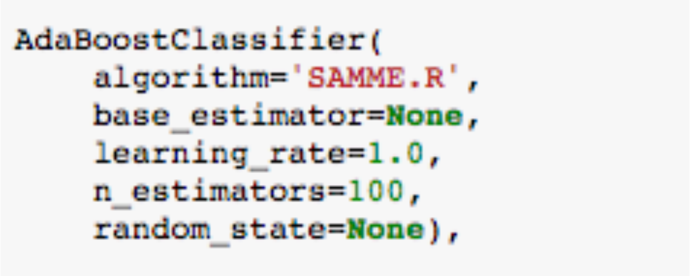
```
NearestCentroid(
    metric='manhattan',
    shrink_threshold=None),

LogisticRegression(
    C=1.0,
    class_weight=None,
    dual=False,
    fit_intercept=True,
    intercept_scaling=1,
    max_iter=100, multi_class='ovr',
    n_jobs=1, penalty='l2', random_state=None,
    solver='liblinear',
    tol=0.0001, verbose=0,
    warm_start=False),

GaussianNB(
    priors=None)
```

Figure 37: Parameters for chosen Randomforest classifier



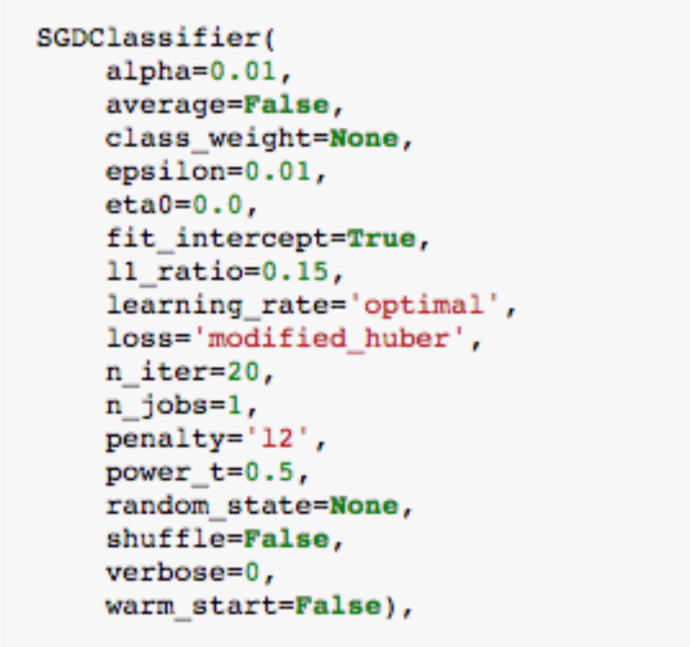
```
RandomForestClassifier(
    bootstrap=True,
    class_weight=None,
    criterion='gini',
    max_depth=None,
    max_features='auto',
    max_leaf_nodes=None,
    min_impurity_split=1e-07,
    min_samples_leaf=1,
    min_samples_split=2,
    min_weight_fraction_leaf=0.0,
    n_estimators=100,
    n_jobs=1,
    oob_score=False,
    random_state=None,
    verbose=0,
    warm_start=False),
```

Figure 38: Parameters for chosen adaboost classifier



```
AdaBoostClassifier(
    algorithm='SAMME.R',
    base_estimator=None,
    learning_rate=1.0,
    n_estimators=100,
    random_state=None),
```

Figure 39: Parameters for chosen SGD classifiers

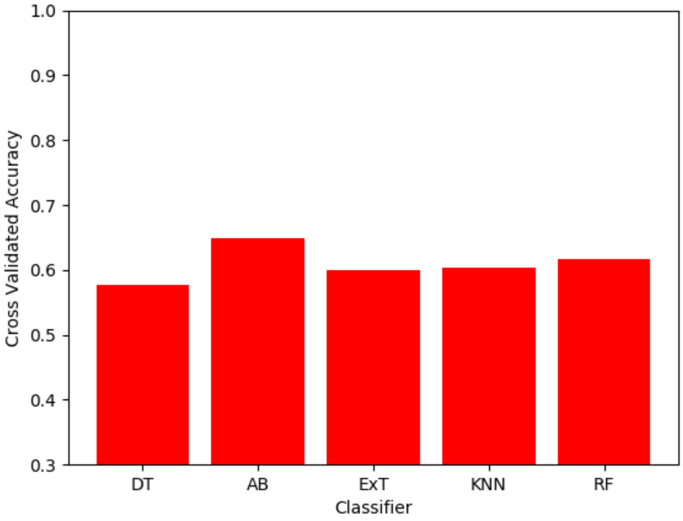


```
SGDClassifier(
    alpha=0.01,
    average=False,
    class_weight=None,
    epsilon=0.01,
    eta0=0.0,
    fit_intercept=True,
    l1_ratio=0.15,
    learning_rate='optimal',
    loss='modified_huber',
    n_iter=20,
    n_jobs=1,
    penalty='l2',
    power_t=0.5,
    random_state=None,
    shuffle=False,
    verbose=0,
    warm_start=False),
```

B Sales dataset E

Sales dataset E was best predicted by AB, with a test cross validated accuracy of 65 %. RF predicted with 62% and KNN with the rest had an accuracy between 57 and 60 % accuracy, which can be seen in figure 40.

Figure 40: Cross Validated Average accuracy by each classifier



Again, the learning models acted strange, in figure 41, AB peaks in performance at a 40 % training data size with exactly 64 % accuracy. In figure 42, an increase can be seen when the number of folds is incremented, and for AB the best K was 7. The worst performing model was DT.

Figure 41: Average accuracy by each classifier with decreasing training set size

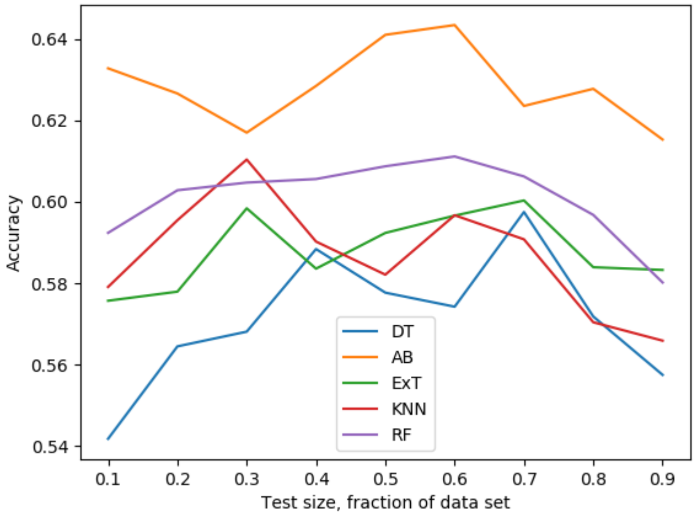
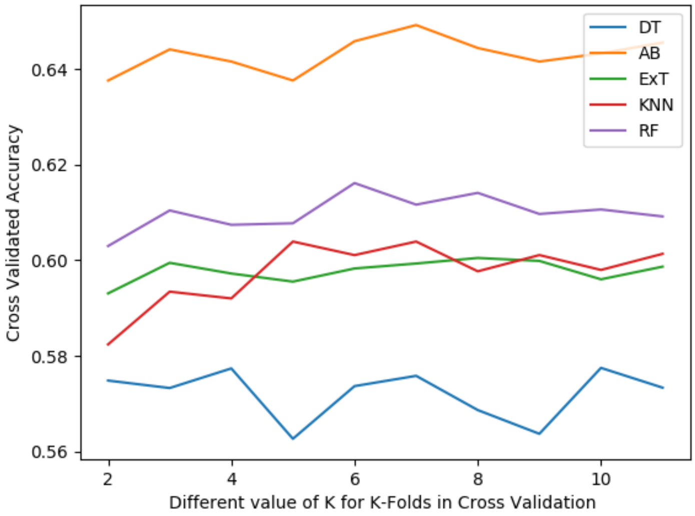


Figure 42: Cross Validated Average accuracy by each classifier with increasing K, i.e. number of folds



www.kth.se