Bachelor Degree Project

UNIVERSITY
OF SKÖVDE

PREDICTING MOVIE RATINGS

A comparative study on random forests and support vector machines

Bachelor Degree Project in Informatics
G2E, 22.5 credits, ECTS
Spring term 2015

Karl Persson

Supervisor: Joe Steinhauer
Examiner: Alexander Karlsson

Abstract

The aim of this work is to evaluate the prediction performance of random forests in comparison to support vector machines, for predicting the numerical user ratings of a movie using pre-release attributes such as its cast, directors, budget and movie genres.

In order to answer this question an experiment was conducted on predicting the overall user rating of 3376 hollywood movies, using data from the well established movie database IMDb. The prediction performance of the two algorithms was assessed and compared over three commonly used performance and error metrics, as well as evaluated by the means of significance testing in order to further investigate whether or not any significant differences could be identified.

The results indicate some differences between the two algorithms, with consistently better performance from random forests in comparison to support vector machines over all of the performance metrics, as well as significantly better results for two out of three metrics. Although a slight difference has been indicated by the results one should also note that both algorithms show great similarities in terms of their prediction performance, making it hard to draw any general conclusions on which algorithm yield the most accurate movie predictions.

Keywords: data mining, machine learning, regression, movie prediction, random forests, support vector machines

Table of contents

1. Introduction

Being able to predict into the future is of great importance in decision making, playing a key role in many areas of science, medicine, finance and industry (Hastie et al., 2009). As our modern computer systems and services have made it possible to continuously gather, process and store an increasing amount of data, our problems in making use of these resources have also increased significantly in complexity and size. As such, only a very small percentage of the data that we gather is utilized, leaving a massive amount of potential information and knowledge buried beneath the surface.

As described by Bramer (2013), machine learning algorithms are generally based upon the notion of finding trends, patterns or anomalies in a set of data, and could be beneficially used for analyzing large and complex datasets. This technology could either be used for grouping certain data points together, finding anomalies in data or for predicting an attribute out of several known parameters, based on trends and patterns found in the training data. Predicting an unknown attribute of an object based on other known attributes and information about it is what is generally known as supervised machine learning. Supervised learning is mainly used to either classify an object into one of several predefined categories, such as vehicle type or a binary yes/no-answer, or in estimating a continuous attribute not bound to any predefined classes, for example the upcoming petrol price, in what is generally known as regression analysis.

While regression analysis and predicting the future might be vital for many different types of applications, the main focus of this study is its use within the domain of movie prediction; spanning from movie recommendation systems, to predicting the box office revenues of hollywood movies prior to release. With the movie databases of today presenting a large amount of information about recent movie productions, as well as overall movie ratings based on the opinions of its users, sparking some interest in what it takes for a movie to become successful in the eyes of its audience and to what degree such aggregated user ratings could be predicted beforehand.

Previous research by Asad et al. (2012) have shown that the general user rating of a movie could be successfully predicted using pre-release attributes such as budget, directors and cast. As such, the goal of this study is to further examine the possibilities of predicting movie ratings using the means of machine learning and regression, as well as to evaluate the use of the two well established machine learning algorithms random forests and support vector machines in doing so. With recent studies showing good prediction performance from random forests, there are some reasons to believe that the algorithm might be suitable also for the purpose of predicting movie ratings. On the other hand support vector machines are considered to be one of the most well established supervised machine learning algorithms, listed by Wu et al. (2008) as being one of the top 10 algorithms in data mining.

In order to investigate the use of machine learning for predicting continuous, numeric movie ratings and whether or not a significantly better prediction performance could be gained from using random forests over support vector machines, an experiment was conducted. The experiment was designed with three main goals in mind: to identify a representative dataset, to prepare the movie data into a model suitable for machine learning, and finally to evaluate and compare the prediction performance of random forests and support vector machines on the specified movie data.

The experiment results show random forest yielding consistently better performance than support vector machines, with significant differences indicated for two out of three performance metrics. It should however also be noted that both algorithms rendered almost identical results over all of the three performance metrics used within the experiment. It is therefore hard to draw any meaningful conclusions on what algorithm yield the best performance based on the single experiment and dataset alone, as a different set of data or an alternative experiment setup might have changed the outcome.

This report is arranged into nine main chapters, starting with the Introduction in presenting a brief summary of the area, problem, experiment and results. The introduction is then followed by a more detailed view of the underlying concepts and theories on which this work is based; including an overview of machine

learning, the two relevant algorithms of support vector machines and random forest, as well as a brief summary of the previous work and research done on comparing the algorithms and within the domain of movie prediction. In the third chapter the problem at hand is defined in further detail, followed by the fourth chapter on the methods used to evaluate the problem as well as their most relevant alternatives. Chapter five present the practical approach of collecting, preparing and mining the dataset, including an explanation of the tools, methods, performance metrics and significance tests included in the experiment. The sixth chapter will then cover a presentation and brief analysis of the experiment results, followed by a discussion on the meaning and meaningfulness of these results. The final chapters present the conclusions of this study and some of the ideas that could be relevant to investigate within future work.

2. Background

The following sections aim to explain the underlying concepts and theories on which the problem of this work is based. Section 2.1 is focused towards giving an understanding of the basic concepts, possibilities and applications of machine learning and data mining, as well as an introduction to supervised machine learning and prediction of continuous data. The goal of Section 2.2 and 2.3 is then to give a further introduction and understanding of the two machine machine learning algorithms evaluated within the frame of this study. Finally, Section 2.4 will focus on the previous work on evaluating and comparing RF and SVM, as well as movie rating prediction and applications and experiments within the movie domain, such as box office predictions and movie recommendation engines.

2.1. Machine learning

As previously noted, machine learning makes it possible to discover trends, patterns or anomalies in a set of data, providing an effective method of analyzing large and complex datasets (Bramer, 2013). Although the concept of machine learning have been established within the area computer science for several decades, the explosion of data in recent years have made the technology highly interesting for mining or extracting information and knowledge from large amounts of data.

Machine learning could be categorized into the two general groups of supervised and unsupervised learning, where the main difference lie in what type of data that is handled by the algorithms. By learning an algorithm to predict the value of an unknown target attribute from a set of learning data where the target is known, e.g. data used for the specific purpose of training the algorithm to predict, we get what is known as supervised learning. Instead relying on the algorithm to find existing groups, relationships or outliers in a previously unspecified dataset of so called unlabelled data is rather what is known as unsupervised learning. Whereas the latter, i.e. unsupervised learning, could be very useful in for example finding what products are most frequently sold together in a supermarket or detecting credit card fraud, the main focus of this work is supervised learning.

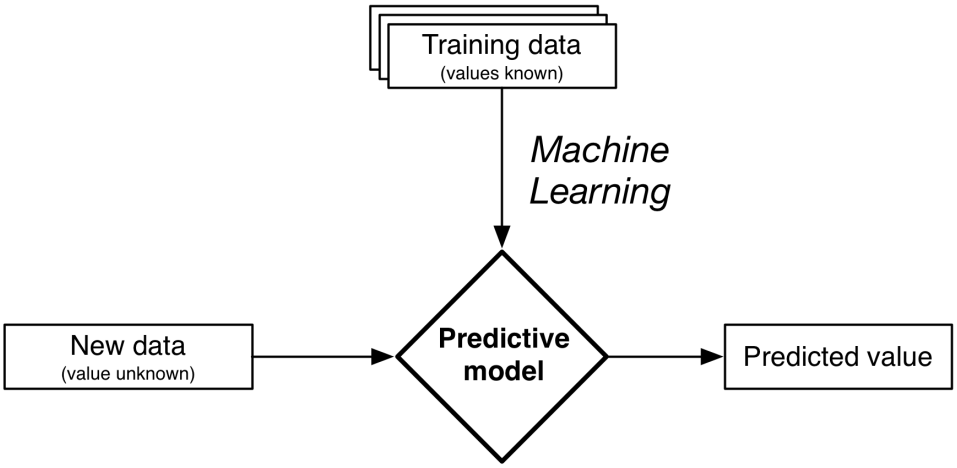

Figure 1. Simplified representation of a supervised learning model.

As previously described supervised learning is based on the notion of being able to predict an unknown attribute of an object based on the attributes we actually know. A supervised learning algorithm could for example be used to predict the value of a house using information that we know, such as the number of rooms, age and location. This type of predictions are made possible by training the supervised machine learning algorithm on data where our target attribute is known, as illustrated in Figure 1. In the previous example the target attribute would be the house value, thus learning the algorithm to predict the value of a house based on previous transactions.

As noted by Bramer (2013) supervised learning could be used for predicting either discrete attributes with a finite number of distinct values or categories through classification, or continuous numerical attributes through what is known as regression. Classification through machine learning might be useful for predicting attributes in the form of a binary true/false value, such as whether or not a student will pass or fail a given exam, or a multiclass value in predicting the origin of wine based on its chemical properties (Lichman, 2013). Accordingly, predicting the target attribute through regression analysis would render in a continuous numeric value rather than a predefined class, which might for example be useful in predicting the value of a house as of the previous example, or the quality of a wine based on its chemical properties (Cortez et al., 2009).

2.2. Support Vector Machines

One of the most frequently used algorithms for supervised learning as identified through the study *Top 10 algorithms in data mining* (Wu et al., 2008) is support vector machines (SVM). SVM is described as one of the most robust and accurate methods among all well-known machine learning algorithms, and even though it was originally intended for classification it has been noted that the algorithm could easily be extended to perform numerical predictions in the form of support vector regression (SVR), as well as time series predictions (Smola and Schölkopf, 2004).

The support vector algorithm dates back to the 1960s, but was in its current form mostly developed during the 1990s for the purpose of optical character recognition (OCR), in converting printed or written characters into machine encoded text. Even though SVM during the last decades have been extended for multiclass classification as well as regression, ranking and time series prediction, it is worth noting that the algorithm was originally developed for binary classification, where the output is categorized into one of two groups as either positive or negative (Yu and Kim, 2012).

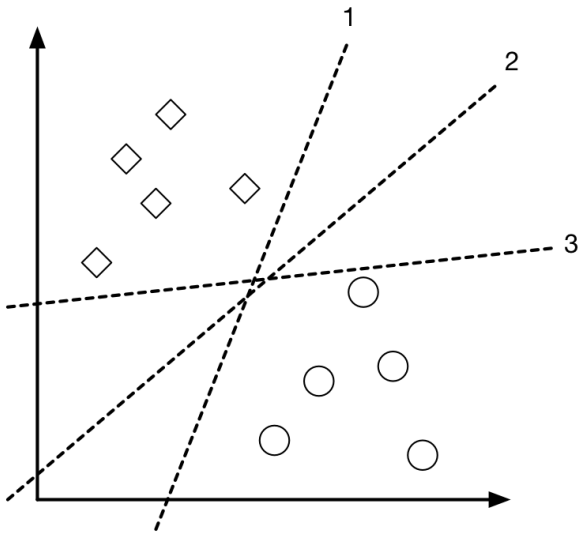

Figure 2. Illustration of linear binary support vector machine classifiers in two dimensions.

Figure 2 above show a simplified representation of such a scenario, where the ten data points are classified into one of the two groups consisting of either diamonds or circles, as each datapoint consist of a two-dimensional vector representing its position in the coordinate system. Just as illustrated, the two groups would have been correctly separated and thus correctly classified by all three linear classifiers, represented by dashed lines and numbers 1-3 in the figure. In order for better generalizations and thereby better predictions on unseen data, the SVM tries to obtain the maximum separation between the two groups by creating a line or hyperplane with the largest possible margin to the nearest data points from both groups (Yu and Kim, 2012).

As the SVM can only separate data linearly using a flat line or hyperplane, separating nonlinear data isn't directly possible. By using a kernel function to map the nonlinear input data into a different, higher dimensional feature space, a linear hyperplane could however still be used for successfully separating the nonlinear dataset into the two classes. Figure 3 below show a simplified illustration of how a two dimensional dataset could be represented by a three dimensional feature space, making it possible to separate the circle objects from the diamonds by the means of a linear hyperplane, despite the objects being linearly inseparable within the original two dimensional input space. According to Chang et al. (2011) one of the most widely used kernels for the SVM is the radial basis function (RBF), mapping data to an infinite dimensional space.

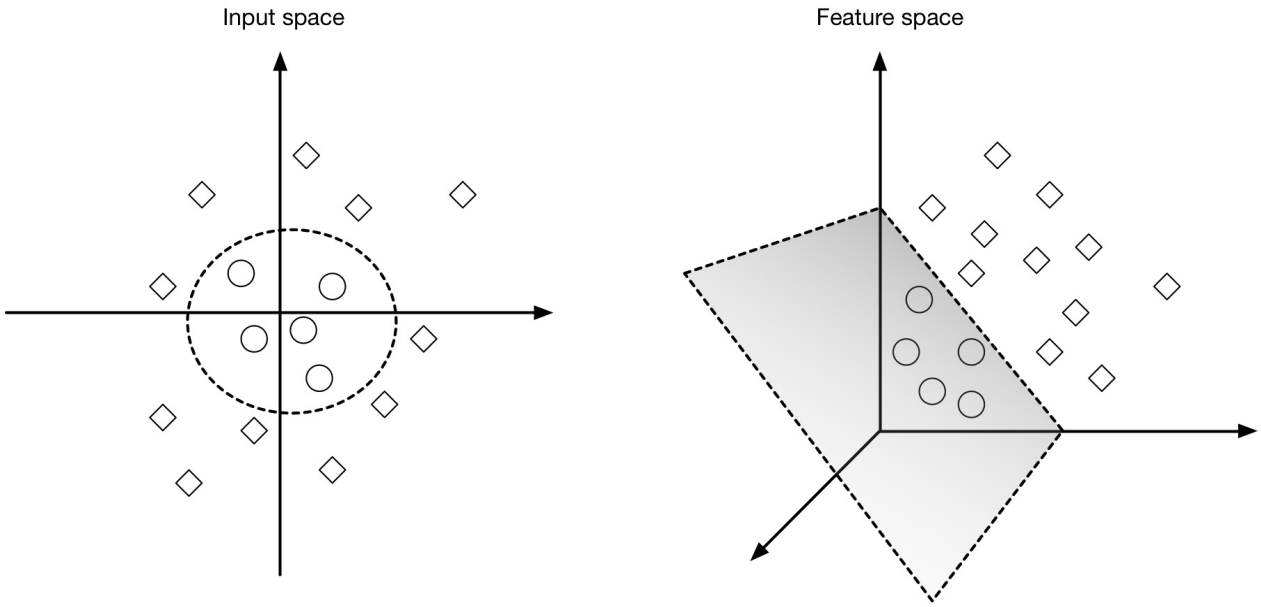

Figure 3. Two dimensional input space mapped into a three dimensional feature space.

Like most other machine learning algorithms and prediction models the SVM require a correct setup in terms of its configuration parameters in order to perform well. As noted by Hsu et al. (2010) there are two relevant parameters to consider for an SVM model based on the RBF kernel: the error penalty parameter *C* and the kernel coefficient *gamma*. As the error penalty parameter *C* controls the margin of misclassification allowed within the prediction model, a value too large will result in the hyperplane being tuned to classify every training example correctly, thus risking overfitting the data causing low generalizability, while a *C* value too small will result in underfitting causing an error prone prediction model. As the *gamma* parameter is used to tune the the mapping between input space and feature space within the RBF kernel, it also require considerable tuning in order to get an optimal prediction model.

Support vector regression works in mostly the same manner as the classifying SVM, utilizing the same margin optimization and kernel functions for nonlinear input data. Basak et al. (2007) argue that support vector regression is the most common application form of all support vector machines, which considering that SVM is regarded as one of the most popular machine learning algorithms (Wu et al., 2008), give the algorithm an important role in many applications.

According to Yu and Kim (2012) two special properties of SVM are that they achieve high generalization by maximizing the margin, and that they support an efficient learning of nonlinear functions using kernels. This is also confirmed by recent studies on the usage of SVM for machine learning applications, where the algorithm yield good prediction performance in comparison to other well respected algorithms such as artificial neural networks and random forests (Were et al, 2015; Bakhtiarizadeh, 2014). As such, the SVM should be able to effectively handle very high dimensional and continuous data.

2.3. Random Forests

Random forests is an ensemble based algorithm, developed by Breiman (2001) with the goal of improving prediction performance over previous models, built upon the idea of combining an ensemble of multiple decision trees through a voting system.

A life analogy of ensemble learning would be to read and weigh in multiple user reviews before purchasing a new product, in order to ensure that we get the right product for our needs. As described by Polikar (2006) the same approach is applied to data analysis through ensemble based learning. Ensemble based machine learning is a generic term for machine learning techniques where multiple classifiers or regressors are used to produce a single prediction. That is, a predictive model based on a number of combined individual models, developed with the goal of improving prediction performance (Rokach, 2010).

Ensemble systems are generally based on the existence of noise, outliers and overlapping data, making the ideal single classifier or regressor virtually impossible to obtain. By however creating a number of classifiers that correctly classify most of the data, the combined result could still improve on that of a single classifier. Ensemble based machine learning therefore rely on the notion of letting the individual classifiers make different errors independent of each other in order for a lower total error, thus requiring each classifier to be as unique as possible, differentiating themselves from the rest.

A diverse set of classifiers could be obtained using several different methods and techniques. A couple of examples considering a decision tree based ensemble stated by Bramer (2013) include using different parameter settings, different training data, different tree generation algorithms and a different subset of attributes, but an ensemble could just as well be constructed from a combination of different types of prediction models.

One of the most popular methods as well as one on which random forests is based is bootstrap aggregating or bagging (Polikar, 2006). Bagging is a method for generating a number of related but still different training sets from a single set of training data, with the goal of improving accuracy and according to Bramer (2013) also reduce overfitting. That is, fitting the machine learning model to noise in the training data instead of the underlying relationship of attributes, resulting in an overly complex model and the consequence of considerably worse prediction performance. The other method for inducing diversity among classifiers or regressors that is utilized by RF is using a random selection of attributes when generating decision trees. This technique was already proposed by Ho (1995) for what's known as *random decision forests*, with the goal of improving generalizability over individual decision trees (Bramer, 2013); later to be used alongside bagging within the RF algorithms developed by Breiman (2001).

Figure 4 below illustrates a small example decision tree for predicting whether or not a passenger of Titanic would have survived the accident. As illustrated the attributes used within the single decision tree are sex, age and the number of siblings of a specific passenger there where on the ship. By generating an ensemble of multiple decision trees like the one below using bagging as a method, slight differences in the trees will occur due to the variations in data between each decision tree. If also generating the ensemble using random subsampling of attributes, the attributes would differ from one tree to another as some trees might for example use sex, age and number of siblings like in Figure 4, while others might be based on another group of attributes such as the number of parents and children, ticket number and passenger class.

A forest of decision trees generated by RF will usually contain a number of hundreds or thousands of unique decision trees depending on the application. Each tree will be generated using different data and attributes and will therefore also predict differently from one another, affecting the final prediction of the RF by casting its vote on the most popular class. Similar to the classification case, Breiman (2001) also note that using RF for the purpose of regression go by similar principles, but instead calculating the average prediction over a set of regression trees (e.g. decision trees whose leaves each represent a continuous numeric value rather than a specific class). As for the reduced risk of overfitting introduced by Ho (1995), the size of decision trees might of course also grow a lot larger than the one in the example above when being part of an ensemble or forest. Likewise, a random forest will generally also benefit from having a large number of

randomized decision trees, although the gains in prediction performance tend to decrease and the computational costs increase as the forest grows.

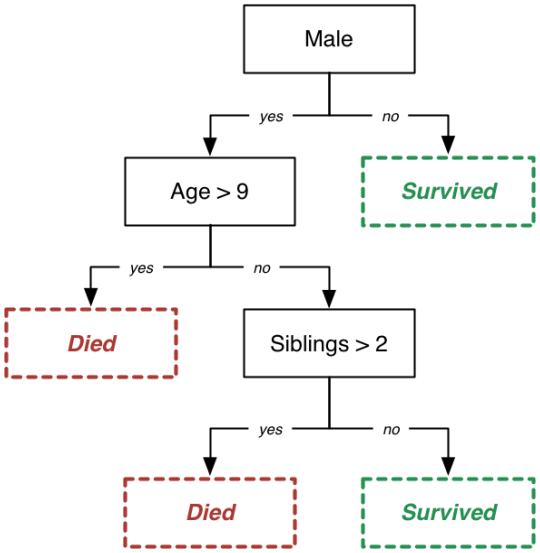

Figure 4. Decision tree for predicting survivors of Titanic.

The split-points selected when generating decision trees can also be used to generate a measure on the relative importance of each attribute in the dataset. This type of variable importance measure is usually based on the idea of the most important attributes being the most commonly used within the splits of a decision tree, and that they therefore should be able give some indications on what attributes have the most impact on predictions. Variable importance is also applicable to decision tree-based ensemble algorithms such as RF through averaging the variable importance over decision trees, and could be an effective tool to get an understanding of attributes and data.

Random forests have been proven successful for several applications, and is considered to yield good performance as well as ease of use and transparency considering importance of attributes. One example of this is in analyzing gene expression data, where RF outperformed the majority of algorithms within the comparison, including the previously described SVM for two out of three datasets within the experiment (Pang et al., 2006). Using RF to identify relevant features in high-dimensional gene and protein data Reif et al. (2006 cited by Verikas et al., 2011) also came to the conclusions that random forests are robust to noise and works well for high-dimensional data. Although RF might be considered one of the most prominent ensemble methods, Polikar (2006) also accentuate the growing attention and popularity of ensemble methods in general due to the broad spectrum of application that might benefit from using them.

2.4. Related work

Although the use of RF and SVM within the movie domain seems to be fairly limited, the two algorithms have been applied and evaluated in many applications for the purpose of regression as well as classification. As previously noted, both algorithms have respectively shown good performance over a variety of applications, in several cases outperforming the majority of its competitors.

Within recent study Verikas et al. (2011) have surveyed a number of large as well as small scale comparisons on data mining and machine learning, all of which include the RF algorithm, specifically issuing its prediction performance in comparison to other algorithms as well as the use of the variable importance estimates available from RF. Among the previous applications and algorithm comparisons included by Verikas et al. (2011) are several large scale studies such as Meyer et al. (2003) and Statnikov et al. (2008), evaluating RF and SVM among other algorithms over a number of 33 and 22 datasets respectively. The small

scale studies included in the analysis, spanning over a more specific, smaller number of datasets, also explore a large variety of applications for classification and regression purposes that might be suitable for machine learning; many of which include both RF and SVM.

Verikas et al. (2011) noted that RF outperformed all other techniques including SVM within more studies than RF was outperformed by other techniques, regardless of the number of data attributes. It was however also noted that the variety of techniques outperforming RF was large, and as such concluded that the superiority of RF over other algorithms like SVM seems to be strictly dependent on the problem at hand. Other studies comparing the two such as the previously mentioned by Pang et al. (2006) and Were et al. (2015) also seems give the impression that prediction performance and superiority of RF contra SVM may vary from one problem to another; thus not being generalizable over every application.

Most of the work on movie prediction to this date seems to be directed towards either review based movie recommendation systems, or in classifying whether or not a specific movie will reach its success goal in terms of its box-office income (Asad et al., 2012). While the use of machine learning techniques in predicting the success of a movie stills seems to be a rather unexplored territory in terms of multiclass or continuous variables, especially for predicting user based movie ratings, there are still a few previous studies conducted within this domain.

Through a recent study on predicting movie ratings Asad et al. (2012) propose a scheme for classifying pre and post-release movies based on their inherent attributes, such as its budget, directors and overall movie cast. The study was conducted using movie data from the Internet Movie Database (IMDb) for classifying movies into the four different categories of Excellent, Average, Poor and Terrible, based on their overall IMDb user rating. Asad et al. (2012) concluded that the average rank of directors in a movie played a major role in predicting its rating for both models, followed by budget and the rank of actors, and that 77% of the total number of movies within the pre-production dataset were being correctly classified. Contrary, the post-release dataset based solely on financial information did however fail to show substantial results with only 56% of movies being correctly classified.

Similar studies have also been conducted with the purpose of predicting the success of a movies using reviews and social media (Krauss et al., 2008; Parmar et al., 2014), and in classifying the MPAA ratings (i.e. parental movie guidelines set by the *Motion Picture Association of America*[1]) of a movie using machine learning techniques (Kabinsingha et al., 2012). Krauss et al. (2008) pointed out the importance of the community activity surrounding a specific movie in the success of the same, concerning its box office figures as well as its chances of receiving an Oscar nomination. Likewise, Marovic et al. (2011) also studied the use of machine learning for predicting the movie ratings of a specific user, closely related to recommendation systems in finding the most relevant personal movie recommendations.

As previously noted, predicting the box office figures (i.e. ticket sales revenue) of a movie is another frequently used application within the domain of movie prediction. Ghiassi et al. (2015) estimates that a majority of the approximately 90 films with production budgets over $100 million, released in the United States during the period of 2008 and 2012, failed to reach enough revenue to meet their production costs. Being able to obtain a more accurate decision support system for pre-production analysis would thereby be of most importance to the movie industry. Within their study, Ghiassi et al. (2015) developed a classification model focused around predicting the revenue of a movie during its pre-production phase based on attributes such as advertising expenditures, runtime and seasonality of the movie, demonstrating machine learning through ANNs to be an affective tool for movie revenue forecasting.

Similarly, Kim et al. (2015) successfully developed a set of box office forecasting models spanning over three different stages prior to as well as after the release of a movie using data from social network services. Kim et al. (2015) also provided a comprehensive survey on the forecasting algorithms and movie attributes used in previous studies within the domain of movie prediction, arguing that there has been little progress made in the development of such machine learning based forecasting models.

[1] http://www.mpaa.org/film-ratings/ [2015-07-03]

3. Problem

The purpose of this work is to evaluate the performance of r*andom forests* in contrast to the well established technique of s*upport vector machines* for predicting numerical movie ratings, based on the attributes of a movie, such as actors, directors, budget or language. Hence, the aim of this work is to answer the following question:

"Are random forests more accurate than support vector machines for predicting user ratings, based on specific attributes of movies?"

More specifically the main focus of this paper is to evaluate the performance of the two algorithms in predicting the popularity of a future movie production as of its general user rating, based on analyzing movie data like the one found on popular online services for movie, TV and celebrity content, such as the *Internet Movie Database* (IMDb) or *Rotten Tomatoes*.

3.1. Motivation

As noted by Asad et al. (2012), analyzing the attributes of a movie using machine learning techniques is a relatively unexplored method for predicting its success. Even considering that such information might be of interest not only to the movie sector in the form producers and financiers, but also to academics, service providers and viewers, most of the current work seems to be focused towards user-specific preferences or analysis of movie reviews.

According to Asad et al. (2012), one reason for the sparse amount of previous work within this area could be due to the large number of complex attributes associated with the success of a movie. Despite this, the classification model that was developed did show promising results in predicting the ratings of a pre-release movie based on data from the movie information database IMDb, indicating that the use of machine learning techniques for predicting movie ratings prior to release is a viable approach. Like some previous work based in the area this study should also establish some additional insight into an approach of mining unstructured movie data.

Being able to predict movie success in the form of box office revenue is as previously described closely related to this problem and of large interest for the film industry, as such predictions would enable another level of decision support systems during the pre-production stages of a movie production. Likewise, means for movie prediction might also be of value for user recommendations; either in the form of general recommendations of not yet released films, or as an assisting factor in a user-focused recommendation system. Such systems might for example be found within services for media streaming, or other similar services aimed more specifically towards media discovery and recommendation.

On a wider perspective, systems for decision support based on numerical predictions are a vital part of many fields, including science, medicine, finance and industry. Thus, this study could as well provide new insights and knowledge of some value to similar applications and areas of use, based on machine learning technology for numerical prediction, i.e. regression analysis. More specifically, some additional insights into the use of RF in comparison to SVM for the purpose of regression.

3.2. Objectives

The following list of objectives have been identified in order to answer the main question of this paper:

1. Identify a representative movie dataset.
2. Identify and prepare a suitable model for representing the movie data.
3. Evaluate prediction performance of the machine learning algorithms r*andom forests* and s*upport vector machines* based on the specified movie data.

The first step is to identify a dataset of movie data that's representative and suitable for analysis. Relevant attributes of such data must include general pre-production information regarding film productions such as genre, language and information about the actors and directors involved. Likewise the data must also include some measure of success, such as user originated movie ratings.

Secondly, the relevant dataset has to be prepared and structured in such a way that the data used is representative of the movie scene at large, as well as viable for analysis by the relevant machine learning techniques and algorithms.

Lastly, the prediction performance of the relevant machine learning algorithms has to be evaluated based on the specified dataset. This means that a set of suitable tools has to be acquired, as well as configured for evaluating both algorithms in comparison to each other based on the data, whilst still ensuring equivalence between in measurements. To be able to compare the algorithms based on their prediction performance, suitable measures of this parameter must therefore also be identified.

4. Method

In order to address the previously defined problem and its objectives one or several empirical methods are required. To make sure that the study is conducted in an adequate and appropriate way, the methods at hand has to be carefully evaluated and chosen. Therefore, the aim of this part is to state a clarification and reasoning around the methods used within the study, as well as their general plan of execution.

As described earlier the main aim of this study is to examine and compare the prediction performance of two specific machine learning algorithms in predicting movie ratings during pre-production or prior to release. Therefore only a limited number of independent and dependent variables are relevant for the study, i.e. the two machine learning algorithms and their effect on prediction performance, which according to Berndtsson et al. (2008) makes conducting an experiment an especially suitable method in that there are few factors that have to be investigated. This theory is also supported by the sheer number of experiments conducted within the area of data mining and machine learning, based on algorithm or method comparison (Olson et al, 2012; Rodriguez-Galiano et al, 2015; Loterman et al., 2012).

4.1. Approach

One of the most significant strengths of an experiment is that it is conducted within a controlled environment, making a solid foundation on which more general conclusions can be made. It is thereby also practicable to statistically measure the effect of certain treatments, as well as replicating the experiment itself. Thus, making it possible to conclude the relationship between cause and effect; based on the notion that the experiment is setup correctly (Wohlin et al., 2012).

According to Wohlin et al. (2012) there are however a few drawbacks of conducting a study using this method, mainly concerning threats to the validity of the experiment. That is, the results not being reliable or general, that the observation doesn't reflect the actual theory being studied, or that the experiment show an incorrect relationship between treatment and outcome. An example of the latter might in this particular case be incorrect conclusions of using a particular algorithm regarding the effect on prediction performance, either through incorrect statistical conclusions or in the form of unknown influences affecting the results, such as the machine learning library used in the experiment. It is naturally also of most importance that experiment is modeled in such a way that it correctly represent the studied system. In order to avoid problems regarding the validity of the experiment proper planning is therefore crucial.

The experiment process begin by capturing a scope in which the problem will be studied, thus defining the objective and goals of the experiment. Further on the design of the experiment itself has to be thoroughly planned, which in this particular case involve acquiring the movie data, stated in the first objective (1), as well as the required tools, what variables to handle and how the results of the experiment should be measured. This also include selection and preparation of data, accordingly to the second objective (2). As previously stated, the data source of movie information to be used within the study must include both certain usable pre-production attributes, such as genre, language, and most importantly attributes for determining the success of a movie. The data must also be prepared in such a way that it is possible to analyze by the means of supervised machine learning algorithms, likely to require some specific tools, as well as for handling the RF and SVM algorithms for regression purposes.

Next up is the actual experiment execution, where a vital part of the previous planning is to be realized. This step includes preparation and execution of the experiment, as well as validation of the collected data in that it doesn't show signs of being unreasonable or erroneous. For a successful experiment execution the two algorithms has to configured and run in such a way that their results are comparable and reflect the general performance of the algorithms. Therefore it is also of most importance that the relevant configuration parameters of each algorithm are optimized to a certain degree, as well as that the experiment itself is configured to enable meaningful and comparable results.

Before any results could be presented, the previously collected data from the experiment execution must also be properly analyzed and interpreted. In order to get a basic understanding of the output data and prediction performance of the algorithms the first objective will be to present and evaluate the output using the means of descriptive statistics and data visualization. To be able to draw any conclusions, the data also has to be analyzed by hypothesis testing, evaluating whether or not the hypothesis is possible to reject. A rejected hypothesis would in this particular case be RF not showing any significant improvement over SVM in regards to prediction performance on predicting the overall user rating of a movie. Thereby, the last objective (3) will be addressed within the operation and analysis parts of the experiment, accordingly producing an answer on whether or not the hypothesis could be confirmed.

This study will be conducted according to existing research ethics, striving towards an honest presentation and communication, objectivity, carefulness, and respect for intellectual property as well as referenced and related sources and work. Data and experiment results will thereby also be handled responsibly, adhering to existing agreements on the use of data and datasets and not in any way manipulating or withholding data or results.

4.2. Alternative methods

Another feasible approach and method for this study would be using a literature analysis. This method would however only be able to cover the problem on a more general level, as the number of articles within the field considering prediction of movie success are very few and generally doesn't focus on any of the relevant algorithms; let alone in comparison to each other. Covering the problem from a wider perspective would therefore likely be the only reasonable way of handling such a comparison, evaluating the two algorithms for the purpose of regression in a general setting rather than within the specific context of movie success or movie rating prediction.

As noted by Berndtsson et al. (2008) it is of most importance that the sources and material used within the study are carefully chosen and interpreted in order for to avoid issues regarding the validity of the results. Likewise, the completeness of the study might also pose an issue in whether or not enough sources are being used in the literature analysis. If thoroughly executed, a literature analysis should however be able to give a good view over the general prediction performance of RF in comparison to SVM, and possibly also whether or not any patterns could be found on one algorithm producing overall better predictions than the other.

While a literature analysis might pose a good alternative, using an experiment is still considered being the most relevant method for this particular type of application and study as an experiment would enable the prediction performance to be properly evaluated within the specific context and of movie prediction; or more specifically predicting the user ratings of a movie.

5. Implementation

The experiment is as previously described aimed at evaluating the prediction performance of RF and SVM for predicting movie ratings, and most importantly to investigate whether or not any significant advantages of using RF can be found.

The following sections will go through the process of collecting, preparing and mining the movie data in order to reliably evaluate the predictive performance of the two algorithms. This includes selection of datasets as well as selection and preparation of data and attributes, but also the practical process of acquiring the results as in selection of metrics, configuration of parameters for the algorithms and design of the experiment itself.

5.1. Data collection

In order to properly evaluate the relevant machine learning algorithms on movie rating predictions a number of relevant datasets has to be considered for the experiment. As previously described, the datasets used within the experiment must include a number of relevant attributes for predicting the success of a movie prior to its release, such as genre, budget, actors starring the movie and most importantly some sort of overall rating. Likewise, the datasets must also include a sufficient number of data points as well as somewhat correct and complete data, in order to be reliably analyzed by the machine learning algorithms, and thus to enable a reasonably valid comparison.

The datasets identified to meet most of these requirements where as follows:

- ***Internet Movie Database (IMDb)****:* Large number of relevant pre-release movie attributes over a very large set of movies. General movie ratings on a scale of 0.0 to 10.0 based on a specified number of user votes. Although the weighted voting system isn't fully disclosed in order to reduce the so called vote stuffing, the voting system is according to the Internet Movie Database (no date) meant to reflect the overall user rating of a movie. The data is available for non-commercial use through plain text data files, to be used within the terms of their copying policy.
- ***Rotten Tomatoes****:* Very similar to IMDb, with several pre-release movie attributes over a large set of movies. Movie ratings are available in the form of 0.0 to 5.0 much like the previous dataset, but also the percentage of critics and users who have given the movie a positive review or rating. Although Rotten Tomatoes seems to deliver largely relevant data for this particular study, ratings, metadata and reviews are only available through their official API, generally restricted by a limited number of API calls per second as well as daily. It should however be noted that increased call limits might be negotiable to a degree (Rotten Tomatoes, no date).
- ***MovieLens****:* Collected by the GroupLens Research Project at the university of Minnesota between 1996 and 2015, differentiating itself from the previous two datasets in that it is focused around user specific movie recommendation (GroupLens, 2014). The MovieLens dataset is easily available, although it doesn't include as many movie attributes. Unlike IMDb and Rotten Tomatoes however, the MovieLens dataset is based on user specific movie ratings from 706 users over 8570 different movies, likely making it less generalizable than the two previous for an application where the user specific ratings of a movie aren't as relevant as the general opinion of its audience.

As noted by Demsar (2006) comparing the prediction performance of the algorithms over a set of multiple datasets would have been the preferred approach, but due to the small number of suitable datasets and their shortcomings considering availability and attributes only a single dataset was eventually chosen for the study; namely the *Internet Movie Database* (IMDb).

The Internet Movie Database provides a dataset consisting of a very large number of movie ratings, in most cases based on many thousand votes, as well as a wide selection of relevant attributes. Although the plain data files available from the well established and widely used movie service might be considered rather

inaccessible due to their structure and distribution over several different files, the dataset seems to be well suited for the experiment. Some of the relevant pre-production and pre-release attributes included in the dataset are for example budget, MPAA-ratings, genres, directors, producers and cast, which should make it suitable for the study in terms of its content.

As the dataset is not regarded freely distributable the experiment will be conducted according to the copyright policy of IMDb.com, inc. Information courtesy of IMDb (http://www.imdb.com). Used with permission. It should also be noted that the data used within the experiment does not include any information about individual users, with the only related information in the database being the total number of votes and the overall user rating of a specific movie. As such, issues regarding the privacy of individual users should not be of any obvious concern.

5.2. Data preparation

As stated in the method section of this paper proper preparation of the data is a necessary step, not just for a valid experiment but also to enable mining of a dataset using the means of machine learning in the first place. That is, a selection of preprocessing steps required for the machine learning framework and algorithms to be able to read and analyze the data, as well as for reducing the dataset to contain the data points and attributes relevant for the study. Likewise it might also be relevant to create or calculate additional attributes from the data, if such derived attributes might be able to aid the analysis and thereby enable better predictions.

The raw IMDb dataset is structured in such a way that most of its attributes and information is organized and stored separately in compressed plain text files. For instance, all of the roughly 600,000 movie ratings from the database are stored in the compressed text file *ratings.list* (e.g. *ratings.list.gz*), which includes textual information about the data as well as a table of film rank, the number of votes and film titles. Thus, some sort of cleaning, integration and preprocessing is likely to be required in order to make good use of the data for the purpose of data mining through supervised machine learning techniques. As such the python package IMDbPY[2] was used in order to convert the dataset into a considerably more manageable relational database.

Like the previous study by Asad et al. (2012), the dataset used for mining was selected as all hollywood movies (e.g. english language and originating from USA) released since year 2000 with more than 1000 user votes. This filtering was done partly to reduce the amount of data as well as to reduce bias. Asad et al. (2012) also suggest that movies from each decade are perceived differently by their viewers due to ongoing psychological changes in society, and thereby a filter on their release date was included when selecting the data. Likewise, hollywood movies seems to have good presence on the movie database used for the experiment.

What factors or attributes that actually influence the success of a movie seems to be widely debated. Previous studies identified by Kim et al. (2015) have used attributes such as *Motion Picture Association of America* (MPAA) ratings and budget, as well as different representations of star power in actresses and actors, for box office predictions. Movie sequels, genre, director and producer are some of the other factors that have been used in previous studies and that could also be relevant for predicting movie ratings prior to the release of a movie. Likewise, Asad et al. (2012) used a set of similar attributes for predicting pre-release movie ratings; namely production budget, average director rank, summed rank of male and female casts as well as the number of votes given by IMDb users.

Attributes identified by Kim et al. (2015) as very commonly used for box office predictions are genre and MPAA rating, as those attributes are the main determinants of the potential market size of a movie. A movie directed towards viewers of all ages within a widely popular genre, will for example target a wider audience than a title that is more restricted in terms of its MPAA-ratings and genres. There are however no consensus on the significance of these attributes for box office predictions, nor if they might have the same importance in predicting movie success in form of user ratings. Likewise, the director and distribution power, measured by factors such as previous box office earnings, number of previous movies and magazine listings, have

[2] http://imdbpy.sourceforge.net [2015-06-12]

neither been consequently reported as meaningful predictors of box office earnings, thus raising some doubt of their usefulness.

For this particular experiment, the attributes chosen as predictor variables where budget, genre, MPAA rating, star power, producer and director, as these factors have seen some previous uses for similar purposes and don't depend on any data only available after the release date; such as reviews, opening week sales or vote figures. Due to the absence of common agreements or standards considering the quantification of star, producer and director power (Kim et al., 2015) each of the three attributes where split into two separate attributes as shown in the table below; mean movie rating and the total number of movie contributions (i.e. number of movie roles played, movies produced or directed, excluding the current production). In order to get the most out of the movie data, ratings and number of previous movies are based on all movies in the dataset rather than just the previous played, excluding the current movie.

Attribute name	**Description**	**Inception (2010)**
Rating	IMDb user rating.	8.8
Budget	Estimated movie budget.	$160 000 000
MPAA	MPAA movie content rating.	PG-13
Actress rating	Average rating of movies played by the actresses and actors.	6.4
Director rating	Average rating of movies directed by the directors.	7.6
Producer rating	Average rating of movies produced by the producers.	6.9
Movies played	Number of movies played by the actresses and actors.	1390
Movies directed	Number of movies directed by the directors.	67
Movies produced	Number of movies produced by the producers.	364
Genre	Genres of the movie.	Action, Mystery, Sci-Fi, Thriller

Table 1. Table of the attributes used within the experiment as well as a practical movie example.

As indicated in the movie example of table 1, each movie the dataset could be connected to more than one genre, and the genre information must therefore be further processed in order to enable the movie data to be converted into a numerical format that's handled by the tools and machine learning algorithms. A rather straight forward way of doing this conversion was to convert each possible movie genre into a boolean, categorical value (e.g. true or false). After several irrelevant or rarely used genres such as *Reality-TV* and *Experimental* were excluded, a total of 20 different genre attributes were added to the dataset. Likewise, MPAA ratings specified within the database as several text strings of the rating (e.g. G, PG, PG-13, R or NC), often followed by a brief motivation, was converted to numerical values corresponding to each of its classes of 0 to 4 respectively. As with the overall movie ratings, actress, director and producer ratings was represented as continuous attributes on the scale of 0.0 to 10.0, and the number of previous movies played, directed or produced represented by numerical attributes ranging from a few to a few thousand movies.

The large budget span of movies ranging from $50 to a massive $300 000 000 was also cut by 1/1000 in order to reduce the risk of low accuracy when scaling the data. Because SVM isn't scale invariant, scaling or standardizing the attributes into a common form of for example [0, 1] or [-1, +1] is highly recommended (Scikit-learn, no year). By scaling the attributes into a common form, large spanning attributes such as

budget might however lose some of its accuracy; but as such low-budget movies are few and far apart in the dataset and budgets only are represented as rough estimates, reducing the attribute into thousands of dollars should not affect the results negatively.

5.3. Data mining

As previously noted it is highly important that the experiment and the two algorithms are configured in such a way that the results reflect their general performance and enables a fair comparison. This section is therefore focused around the practical process of mining the movie data, including experiment configuration, optimization of algorithm parameters, metrics of prediction performance, significance testing and a description of the toolchain used within the experiment.

5.3.1. Cross-validation

To be able to assess the prediction performance of the two algorithms, a setup of cross-validation over several different combinations of algorithms parameters as well as performance metrics was used for the experiment. As pointed out by Bramer (2013) cross-validation can be used for model selection and model assessment, and have been widely adopted for such purposes as it poses a way of systematically training and testing a model on separate data, thus reducing the risk of *overfitting*. One well known such model is *k-fold cross-validation*, which is based on the notion of splitting the given dataset into a number of *k* separate instances or folds as illustrated in Figure 5 below.

By splitting the dataset into three folds as shown in Figure 5 we are be able to test a predictive model or algorithm on one separate instance of data while training it on the remaining two, for a total of three different sets of test data. As illustrated in Figure 5 each of the three iterations or folds will partition the data in such a way that a new part of the dataset is used for testing (marked as gray), while the rest of the dataset is used for training the model. This way we'll get to use the whole dataset for testing while still avoiding using the same data that the model is currently trained on for testing the same. One of the most common variants of k-fold cross-validation is the 10-fold CV, splitting the data in 10-parts and as such running 10 iterations contrary to the 3 of the previous example.

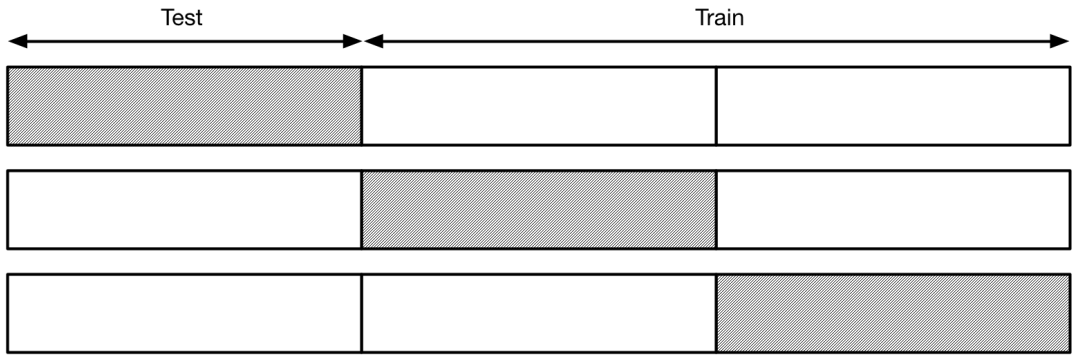

Figure 5. Cross-validation over a single dataset, iterating over the dataset for three different folds.

An especially relevant extension of cross-validation is nested cross-validation, combining model selection with model assessment within a single process. Nested cross-validation might for example be useful if the prediction model require some parameter optimizations prior to assessing the performance of the model. Whereas k-fold CV use only one loop of cross-validation to assess the performance of a prediction model, nested CV adds another cross-validation loop to use on the training data of every iteration called the inner cross-validation. The inner CV is used not for assessing the performance of the model but rather to find its optimal parameters, for example configuring the model using a predefined grid of configuration parameters.

For every iteration of the outer CV-loop the model will be tuned into what seems to be an optimal setup using a so called grid-search of predefined parameters within the inner CV loop. The optimal model based on the outer training data will then be assessed using the outer test data, and the process repeated for each iteration of the outer CV. Figure 6 below show a slightly simplified model of the process, illustrating a single iteration of nested cross-validation with 3 inner folds for assessing the model configuration as well as 3 outer folds for assessing the final model performance. The dark gray part represent the test data of the outer cross-validation and the light gray part represent the test data of the inner cross-validation.

Although the illustration only show one inner CV iteration, the same applies to this model as the previous in that one third of the data is used for testing the model, while two thirds are used for training. Unlike the previous model however, the two thirds of the dataset used for training are as also partitioned into three additional folds, embedding another loop of cross-validation into the model. Within this inner cross-validation one third of the data (light gray) is used for testing the model while the other two thirds are used for training. For every outer fold of cross-validation all of the available parameter combinations specified within the parameter grid will be evaluated via the inner cross-validation scheme, in order to find the optimal model configuration. The configuration that is found to be the most effective is then used to evaluate the prediction model on the outer test data for the current fold (dark gray).

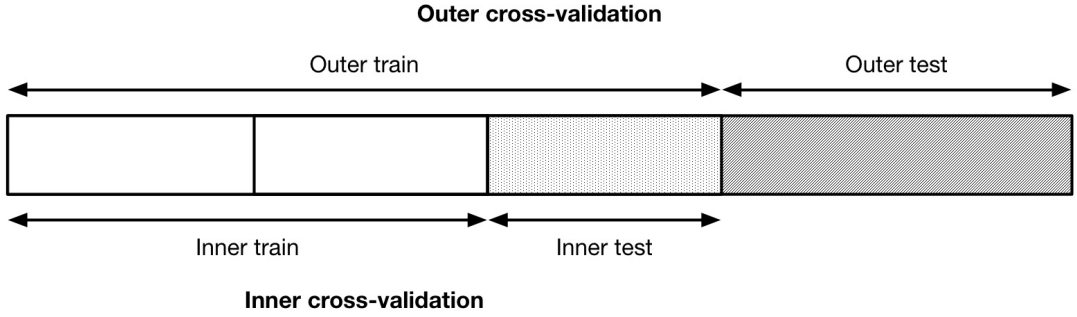

Figure 6. Nested cross-validation.

Further on, Krstajic et al. (2014) propose the importance of repeating nested cross-validation when selecting an optimal model due to additional variance, and that the increased need for computational power become less of a problem as cloud computing gets more common and affordable.

For the purpose of this experiment a setup of five times 10-fold nested cross-validation (e.g. 10 inner folds and 10 outer folds) was used. Although a higher number of iterations of the nested cross-validation might have been desirable, the number of repetitions was kept relatively low due to the high computational requirements and thus high time consumption on the desktop system used for the study.

As some parameter optimizations of the two algorithms was needed, a predefined grid search of suitable parameters where also configured for the experiment. With the SVM being highly dependent on a correct setup, a grid of parameter values was defined for the error penalty parameter *C* as well as for the kernel coefficient *gamma*. The kernel used within the experiment was thereby also set to the well established non-linear RBF. Possible values of the two parameters where chosen based on the suggestion of using exponentially growing sequences by Hsu et al. (2010) represented in the following list; both of which generally having a large impact on prediction performance of an SVM-based prediction model.

- ***C:*** $2^{-5}, 2^{-3}, \dots, 2^{15}$
- ***Gamma:*** $2^{-15}, 2^{-13}, \dots, 2^{3}$

Although the RF algorithm might not be as dependent on a specific setup as SVM, a grid of parameters concerning the number of estimators (e.g. trees) and maximum number of features was still defined for the experiment. The algorithm is generally known to gain from using a large number of trees, but as Segal

(2003) have indicated overfitting for some special cases of noisy data, and that additional parameters for RF didn't add much to the time consumption of the experiment, a number of different configurations concerning the count of estimators was still added. The maximum number of features to consider when looking for the best split was also set to the three different settings of total number of features (auto), square root and binary logarithm.

- ***No. estimators:*** 10, 100, 1000
- ***Max. features:*** total number of features, sqrt, $\log_2$

In order to assess the prediction performance of both algorithms the full cross-validation procedure was executed for each algorithm on the same seeds of data to get comparable results, and the whole process repeated for a number of different prediction metrics as listed below. For each algorithm and round of cross-validation the prediction models were tuned for the current metric, on the same seed and instance of the dataset as the previous one. The Python based machine learning library Scikit-learn was eventually chosen for the experiment, as it supports all of the relevant tools and algorithms out of the box.

5.3.2. Performance metrics

Due to the absence of one single metric able to reflect all different factors affecting the prediction performance of an algorithm, an array of three different metrics were put together for the experiment: *coefficient of determination, root mean squared error and mean absolute error*. All of the metrics seems to be very well represented in previous work on the prediction performance and accuracy of regression models (Erdal & Karakurt, 2013; Loterman et al., 2012), and should be able to give some further information about the performance of the algorithms.

- ***Coefficient of determination (R^2):*** Indicates how well the data fits a prediction model in the amount of the total variance explained by the model. R^2 is commonly used for evaluating the performance of regression models, but its usefulness have also been questioned for not being as relevant for predictive models (Razi & Athappilly, 2005).
- ***Root mean squared error (RMSE):*** Measures the average magnitude of errors in the predicted values. That is, the average distance of a data point from the fitted line. Being a quadratic measure RMSE is most affected by large errors, thus useful for when large errors are especially undesirable.
- ***Mean absolute error (MAE):*** Measures the average magnitude of errors irrespective of their direction. Contrary to RMSE, MAE is a linear measure and therefore isn't as affected by larger errors. As noted by Erdal & Karakurt (2013) MAE should be useful for measuring how close predictions are to the outcomes. Some also seems to argue that MAE is better suited for measuring average model performance than RMSE (Chai & Draxler, 2014).

5.3.3. Significance testing

As previously noted, properly testing the hypothesis by the means of a significance test is crucial for the experiment. Due to the continuous nature of the predicted values, using a test based on the number of correct and incorrect predictions such as McNemar's test isn't directly applicable, as such a test would require additional definitions of what type of predictions should be considered correct or incorrect. Thereby, the type of tests considered to further evaluate the hypothesis of the experiment are first and foremost continuous statistical tests such as the paired t-test and Wilcoxon signed-rank test.

Demsar (2006) argue that the Wilcoxon signed-rank test should be considered safer in comparison to the paired t-test when the assumptions of the paired t-test aren't fully met, among other factors since the latter assume a normal distribution of the data; something that for example may occur due to small data samples. Although it might not necessarily be as powerful as the paired t-test, the Wilcoxon signed-rank test was still considered being the more sensible alternative for the experiment as cross-validation might violate the underlying assumptions of the t-test (Diettrich, 1998).

In general the Wilcoxon signed-rank test works by calculating the differences between each observation, ranking them from the largest to the smallest absolute differences between observations. In this particular case each observation will consist of one measurement from RF and one measurement from SVM, based on the same train and test data from the movie dataset. The smallest of the summed positive ranks and the summed negative ranks is used to determine the significance level, where a smaller summed rank indicate a larger difference between the algorithms, and will as such result in a smaller p-value. Thereby equally distributed positive and negative summed ranks would also fall under the null hypothesis (McDonald, 2007).

In order to reduce the risk of an incorrect rejection of the null-hypothesis (e.g. RF incorrectly showing a significant improvement in prediction performance over SVM) the significance will only be tested over a single repetition of cross-validation, contrary to the fully repeated set of data. As noted by Diettrich (1998), resampling over the same data will likely yield a very high risk of type 1 error (i.e. an incorrectly rejected null-hypothesis). While a single run of 10-fold cross-validation still have been reported to result in a somewhat elevated probability of type 1 error, paired up with the Wilcoxon signed-rank test the method should still be able to give reasonably reliable results.

6. Results

The following section aim to present an overview of the experiment results, comparing the prediction performance of RF and SVM over the IMDb movie dataset, consisting of 3376 hollywood movies released since the year 2000. As previously stated, all results from the experiment are based on a setup of 10-fold nested cross-validation combined with grid search of model parameters, in order to optimize the parameters of each algorithm to better reflect its performance.

In order to first and foremost establish a better understanding of the predicted movie data, as well as the general differences and most of all similarities of the two machine learning algorithms, a combined scatter plot over the predicted values of each algorithm is illustrated in Figure 7. Further on, the scatter plot is followed by several box plots for each performance metric (i.e. R^2, *RMSE and MAE*), as well as the corresponding significance tests.

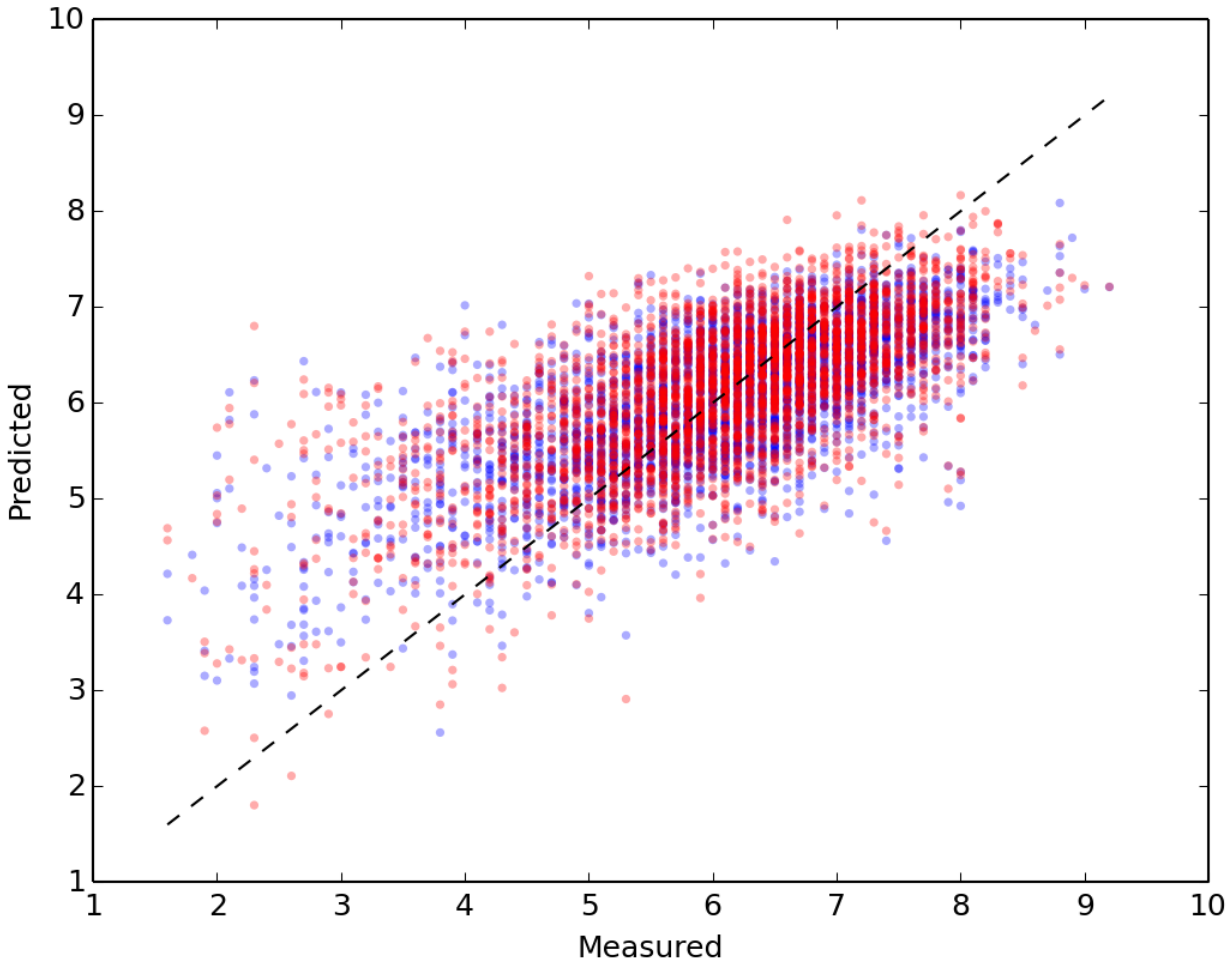

Figure 7. Scatter plot of the full predicted dataset. RF represented by blue, SVM by red.

Figure 7 above illustrate the full dataset of 3376 movies predicted by both algorithms using the same nested cross-validation as the rest of this experiment. It should therefore be noted that all of the data points in the illustration are predicted through a combination of 10 splits of training and test data, using the all same partitioning of data for both algorithms. Thereby, the algorithm parameters chosen for each iteration of the outer cross-validation loop, e.g. model assessment, might differ slightly from iteration to iteration as the partitioning of training data to some degree will vary. It should also be noted that the models used for these predictions are tuned using the *coefficient of determination*, R^2, the default scoring metric of Scikit-learn.

Worth pointing out is that the X-axis of the scatterplot denote the measured user ratings of a movie while the Y-axis mark the predicted values. Thus, the dashed diagonal line represent the optimal and fully correct rating, while data points below or above is either too low or too high.

The first thing to notice in the scatter plot is likely to be the similarities of the two algorithms in terms of their predictions. Neither of the two algorithms seems to be able to fully explain the user rating of a movie based on the given attributes, as the predicted values often seems to substantially differentiate themselves from the measured movie ratings. It have also become clear that the predictions of movies with a high measured user rating tend to be somewhat underestimated, falling under the dashed line, and especially that movies with a low user rating tend to be overestimated. Movies with low measured ratings also seems to yield less accurate predictions than ones with higher average user ratings. Although the predictions of the SVM might seem to be slightly higher and more widespread than the ones of the RF, it is hard to draw any general conclusions based on the scatter plot alone as it present a large amount of similar predictions from both algorithms.

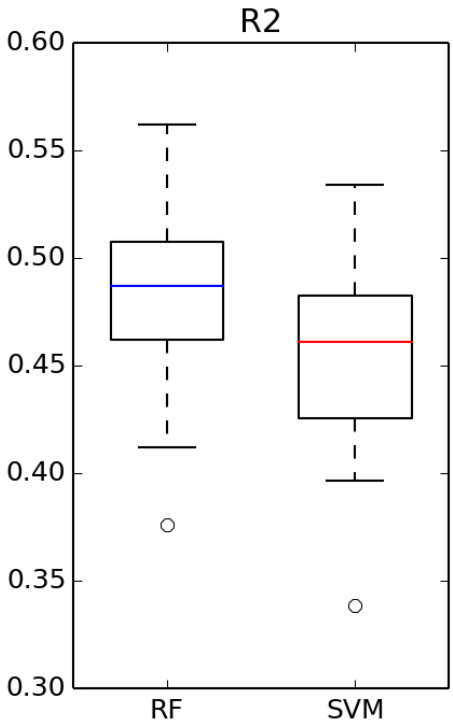

Figure 8. Box plot over R^2, based on the full five repetitions of 10-fold nested CV.

The box plot in Figure 8 is based on the full five times repeated 10-fold cross-validation for each algorithm, and tuned for optimal R^2 values. As previously noted, the *coefficient of determination* indicates how well the data fits a prediction model. The higher the R^2 value, the better the model.

As shown in Figure 8, the box plot seems to indicate a slight edge to RF in comparison to the SVM. Random forests showing slightly better data overall, with overlapping boxes but also a notably higher median for RF. Still, the box plot show that the two algorithms perform relatively closely, with mean R^2 vales of 0.48 (+/- 0.04) and 0.46 (+/- 0.04) respectively for all 50 data points. Looking at the significance tests, the results from all five repetitions consistently show p-values below a 5% significance level, thus indicating a significant difference between the two algorithms with a slight advantage to RF.

Root mean squared error (*RMSE*) show a similar although less pronounced trend concerning the magnitude of errors in the data. As shown in box plot in Figure 9 below, RF seems to yield an overall just so slightly lower *RMSE* than SVM, with a mean of 0.84 (+/- 0.04) compared to 0.86 (+/- 0.05) of the SVM. It should

also be noted that the outlier of the SVM box plot might implicate a greater difference between the lowest and thus best *RMSE* values of each algorithm in the experiment than there really are, as RF show a similarly low best value. Still, the array of significance tests on *RMSE* for each algorithm show a just as consistent significant edge towards RF in comparison to the SVM, although the difference in performance doesn't seem to be especially large at all considering the *RMSE* values of the two algorithms.

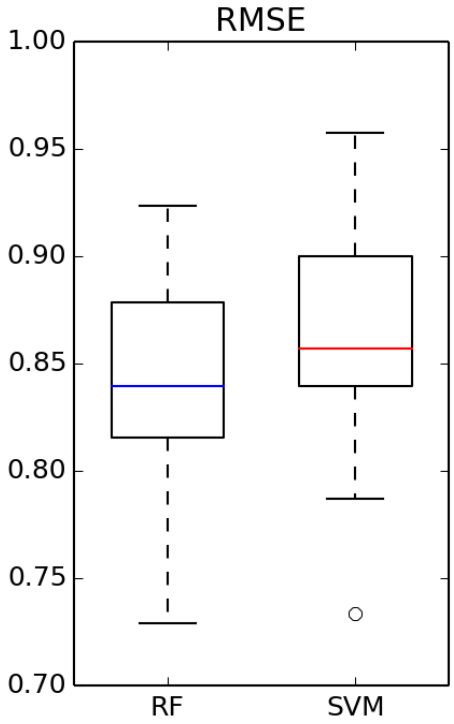

Figure 9. Box plot over RMSE, based on the full five repetitions of 10-fold nested CV.

As previously noted, *mean absolute error* (*MAE*) is in contrast to *RMSE* a linear metric, and therefore isn't as affected by larger errors as the latter one. Looking at the box plot in Figure 10 below it also becomes clear that the difference between RF and SVM is even smaller for this metric. Although RF again is showing slightly better numbers compared to the SVM their performance seems to be very close; not only considering the major overlap of the two box plots, but also the close mean *MAE* values of 0.65 (+/- 0.03) for RF and 0.66 (+/- 0.03) for SVM. Like the previous box plots concerning *RMSE* (Figure 9) an outlier is identified for the SVM, implying better performance than might actually be the case, maxing out close to 0.74; a little higher than the maximum *MAE* value for RF.

Contrary to the previous results of R^2 and *RMSE*, it is hard to state any significant difference of *MAE* on the two algorithms. Out of the five runs of 10-fold nested cross-validation, only three show a significant difference whereas the remaining two cross-validation runs rendered p-values of well above the 5% significance level. Considering the large overlap of the box plots and even smaller span of *MAE* measurements, the less significant results are hardly surprising.

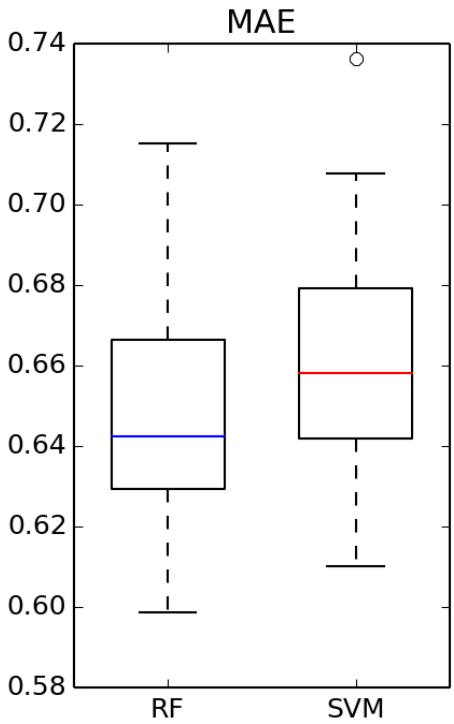

Figure 10. Box plot over MAE, based on the full five repetitions of 10-fold nested CV.

7. Discussion

Based on the previously presented results it becomes quite clear that RF and SVM yield a rather similar prediction performance, not only when looking at the output but also pretty close in the terms of the tested performance metrics. Still, the experiment also seems to show a slight edge in performance in favor to the RF over the given movie dataset, with both R^2 and *RMSE* indicating a slight but significant difference. Concerning *MAE* however, the significance level of 5% were only met for three out of five seeds of the dataset, making it hard to fully reject the null hypothesis and thus conclude any significant improvements of using RF in comparison to SVM for one out of the three evaluated performance metrics.

It should still be noted that the SVM is highly dependent on a correct configuration in terms of algorithm and kernel parameters, considerably more so than RF. With the experiment only conducted using a limited number of possible parameter values, SVM would likely be the algorithm most affected by a non-optimal parameter choice. Thereby SVM would probably be the one algorithm most likely to underperform within the study, making it hard to fully ensure generalized results outside of the particular experiment setup. Still, the prediction performance of RF have also been reported to be highly affected by having few estimators, as well as an incorrect number of variables for each estimator.

It is also relevant to point out some validity threads related to the setup of the experiment. As previously noted, Diettrich (1998) reported the 10-fold cross-validation having a somewhat elevated probability of type 1 error, which might well affect the results of incorrectly indicating a significant difference in the prediction performance of the two algorithms, and should be worth considering. The benefits and drawbacks of using cross-validation techniques in model evaluation also seems to be a widely debated subject, questioning the meaningfulness of cross-validation and significance testing overall (Vanwinchelen & Blockeel, 2012). Whether or not the same trends concerning prediction performance would stand true also for other similar datasets might as well be hard to fully generalize, as the comparison this far only covers a single dataset.

Another interesting and most important aspect to reflect over is the predictability of movie data in general, as the factors and attributes studied in this experiment only seems explain one part of what makes a movie successful in the eyes of its audience and viewers. To this date Kim et al. (2015) claim that there has been little progress in forecasting models within the domain of motion pictures, and there seems to be no general consensus on the importance of certain factors and attributes in predicting the success of a movie; let alone the user rating of one. It is thereby also a risk that the attributes used within this experiment aren't able to give enough relevant information and details about the movie data in order to separate the two algorithms in terms of their prediction performance. With a better and more nuanced set of predictors one of the algorithms might have been able to find a wider range of relations in the movie data than the other, thus likely to render better prediction performance. As for the current setup it is therefore still plausible that both algorithms where able to pick up the existing relationships in the data, making them perform very similarly within the experiment.

The bar chart in Figure 11 below show an overview of the variable importance as interpreted by the RF configuration according to the variable importance procedure previously described in Section 2.3, and should be able to give some insight into the relative importance of each attribute used within the experiment. Looking at the relative importance of each attribute within the experiment it becomes clear that the average actor and actress user ratings play a key role in predicting user rating of a movie; considerably more so than all of the other attributes. At half the variable importance level relative to the actress ratings are the producer and director ratings, closely followed the number of previous produced, played and directed movies, as well as budget and surprisingly enough whether or not the movie is considered being a drama. At below 10% of the variable importance level of the average actress rating is the MPAA-classification, tightly followed by the remaining set of genres, with for example war, western and history being considered almost completely irrelevant attributes for the RF-based prediction model.

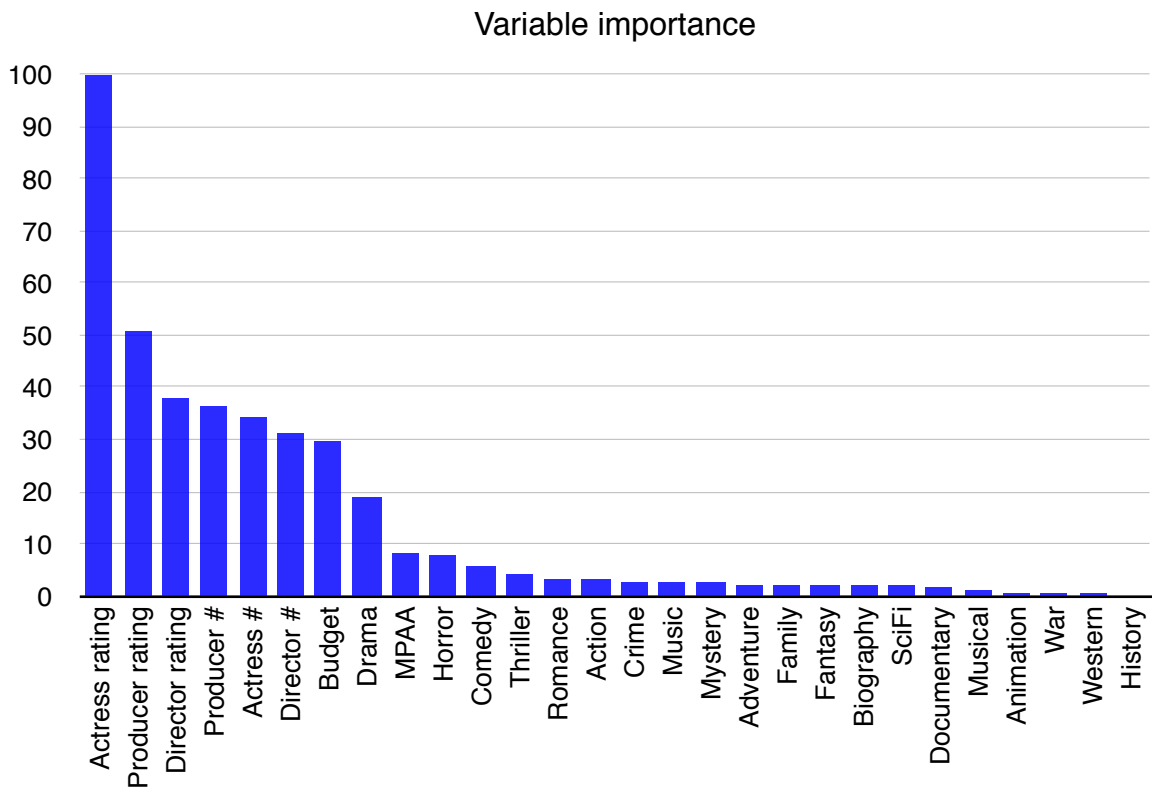

Figure 11. Chart of the relative importance of attributes, generated by the RF prediction model.

Although not being the main focus of this work, looking at the algorithms from other perspectives it could sometimes be hard not to notice some of their differences. As touched upon within Section 5.3.1, SVM + RBF seems to require more extensive parameter optimizations than RF in order to perform well. Closely related is also the need for computational power and scaling over multiple processors or cores, as the SVM implementation used within the experiment did not only require a more extensive parameter grid search, but also seems to consume significantly more processing time than RF (Verikas et al., 2011). This despite an additional parallelization over cross-validation runs, which would otherwise not be available for the SVM + RBF-kernel implementations. Contrary, the basic structure of RF enables the algorithm to make good use of multicore or multiprocessor systems, although it should still be noted that a larger forest (i.e. higher number of estimators) would require a higher computation time (e.g. linearly growing) and therefore might not yield the same benefits. As previously noted RF also propose a way of evaluating the importance of attributes within a dataset by assessing the attributes and structure of its decision trees.

8. Conclusion

As previously noted, the purpose of this work was to evaluate the performance of the random forests algorithm in contrast to the well established support vector machines on predicting numerical movie ratings, based on a number of relevant pre-release attributes of the movie such as its actors, budget etc. More specifically the aim was to provide an answer on whether or not random forests could be considered better performing than support vector machines for predicting user based movie ratings, as stated in the question below.

”Are random forests more accurate than support vector machines for predicting user ratings, based on specific attributes of movies?”

In order to answer this question an experiment was conducted with the goal of evaluating and comparing the prediction performance of the two algorithms on movie data. The experiment was conducted using the method of repeated nested cross-validation on a set of 3376 hollywood movies from the well established movie database IMDb, where the two algorithms were evaluated side by side for three commonly used metrics for assessing performance and error in predictive models.

The results from the comparisons show some differences between the two algorithms, to an extent favoring RF for all three metrics on prediction performance, but most importantly also a showing a high resemblance in the predictions and prediction performance made by RF and SVM. As it turns out only part of the trends in the general user ratings of a movie is explained by the prediction models, where high scores seems to be consistently underestimated and low scores consistently overestimated and overall harder to predict.

Significance testing consistently indicated improvements in RF over SVM for two out of three metrics used within the study, with significant differences on both R^2 and *RMSE* measurements but not concerning *MAE*. While some significant differences where found in support of the hypothesis, both algorithms still performed very similarly throughout the whole experiment. As such, different datasets or model parameters might affect the outcome of the results, making it difficult to conclude any definitive answers concerning the hypothesis and what algorithm yield the best prediction performance or predictive accuracy. One possible explanation to the almost identical prediction performance between algorithms might be the limited predictability of the movie attributes used within the experiment, possibly not providing good enough predictors to clearly separate the two methods in terms of prediction performance.

9. Future work

Looking back at this experiment and results but also the movie domain as a whole it becomes clear that there are still work to be done within the area of movie prediction. This section aim to present an overview of the relevant future work that has been identified throughout the study.

The first and possibly the most obvious improvement that come to mind is the movie attributes used within the study. As noted in previous sections, the current set of movie attributes only seems to describe part of the general user rating of a movie, making to believe that further improvements and research on the factors and relations within movie data would be highly beneficial for the overall quality of movie predictions. When looking at the variable importance of the experiment data identified by the RF algorithm (Figure 11) the power of the movie cast seems to be an important factor to account for when predicting movie ratings, and might therefore be worth to develop further; especially considering that there seems to be no consensus on how to represent this type of information within previous studies. Kim et al. (2015) also suggest that factors related to word-of-mouth (WOM), such as blogposts, play a key role in predicting the success of a movie within previous studies on predicting box office. WOM-based attributes might therefore also be useful as predictors within the domain of movie prediction in order to further improve prediction performance on such datasets.

Further on it might also be relevant to evaluate whether or not the prediction performance and results are generalizable over multiple datasets as well as over larger and smaller datasets in order to be able to draw more general conclusions, as the current experiment setup only incorporated a single though well established dataset. Likewise it might also be of interest to evaluate whether or not the same is applicable to box office predictions, to further widen the scope of the study.

Another possibility would naturally also be to include a wider range of algorithms and algorithm configurations, as the methods included in the current study were narrowed down to fit the given time frame. Previous studies have for example shown artificial neural networks (ANNs) to be a good alternative to SVM and RF for box office predictions and regression purposes in general (Verikas et al., 2011; Kim et al., 2015), but alternative ensemble methods and SVM kernels might also be feasible alternatives. Such machine learning techniques are of course not only applicable to movie prediction, but might as well be used for other purposes within the scope of machine learning and regression.

10. References

Asad, K.I., Ahmed, T. & Rahman, M.S. (2012). Movie popularity classification based on inherent movie attributes using C4.5, PART and correlation coefficient. Informatics, Electronics & Vision (ICIEV), International Conference on.

Bakhtiarizadeh, M. R., Moradi-Shahrbabak, M., Ebrahimi, M., & Ebrahimie, E. (2014). Neural network and SVM classifiers accurately predict lipid binding proteins, irrespective of sequence homology.

Basak, D., Pal, S. & Patranabis, D. (2007). Support vector regression. Neural Information Processing. 11, 203-224.

Berndtsson, M., Hansson, J., Olsson, B., Lundell, B. (2008). Thesis projects: a guide for students in computer information systems. 2nd ed. London, Springer.

Bramer, M.A. (2013). Principles of data mining. Springer.

Breiman, L. (2001). Random Forests. Machine Learning. 45, 5-32.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geoscientific model development. 7, 1247-1250.

Chang, Y.-W., Hsieh, C.-J., Chang, K.-W., Ringgaard, M., & Lin, C.-J. (2011). Training and testing low-degree polynomial data mappings via linear svm. Journal of Machine Learning Research. 11, 1471-1490.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T. & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems. 47, 547-553.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of machine learning research. 7, 1-30.

Diettrich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. Neural computation. 10, 1895-1923.

Erdal, H. I., & Karakurt, O. (2013). Advancing monthly streamflow prediction accuracy of CART models using ensemble learning paradigms. Journal of hydrology. 477, 119-128.

Ghiassi, M., Lio, D., & Moon, B. (2015). Pre-production forecasting of movie revenues with a dynamic artificial neural network. Expert Systems with Applications. 42, 3176-3193.

GroupLens. (2014). MovieLens. https://movielens.org. [2015-06-10]

Hastie, T., Tibshirani, R. & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. 2nd ed. Springer.

Ho, T. K. (1995). Random decision forests. International Conference on Document Analysis and Recognition. 1, 278.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2010). A practical guide to support vector classification.

Internet Movie Database. (no year). The vote average for film "X" is clearly wrong! Why are you displaying another rating? http://www.imdb.com/help/show_leaf?votes. [2015-06-10]

Kabinsingha, S., Chindasorn, S., & Chantrapornchai, C. (2012). A movie rating approach and application based on data mining. International Journal of Engineering and Innovative Technology.

Kim, T., Hong, J., & Kang, P. (2015). Box office forecasting using machine learning algorithms based on SNS data.

Krauss, J., Nann, S., & Simon, D. (2008). Predicting movie success and academy awards through sentiment and social network analysis. European Conference on Information Systems.

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Loterman, G., Brown, I., Martens, D., Mues, C., & Baesens, B. (2012). Benchmarking regression algorithms for loss given default modeling. International Journal of Forecasting. 28, 161.

Marovic, M., Mihokovic, M., Miksa, M., Pribil, S., & Tus, A. (2011). Automatic movie ratings prediction using machine learning. Convention on Information & Communication Technology Electronics & Microelectronics.

McDonald, J. H. (2007). The handbook of biological statistics. http://www.biostathandbook.com/wilcoxonsignedrank.html. [2015-07-07]

Meinhausen, N. (2006). Quantile regression forests. Journal of Machine Learning Research. 7, 983-999.

Meyer, D., Leisch, F., & Honk, K. (2003). The support vector machine under test. Neurocomputing. 55, 169-186.

Olson, D.L., Delen, D., & Meng, Y. (2012). Comparative analysis of data mining methods for bankruptcy prediction. Decision Support Systems. 52, 464-473.

Pang, H., Lin, A., Holford, M., Enerson, B.E., Lu, B., Lawton, M.P., Floyd, E., Zhao, H. (2006). Pathway analysis using random forests classification and regression. Bioinformatics (Oxford, England). 22, 2028-36.

Parmar, H., Bhanderi, S., & Shah, G. (2014). Sentiment mining of movie reviews using random forest with tuned hyperparameters.

Polikar, R. (2006). Ensemble based systems in decision making. IEEE Circuits and Systems Magazine.

Razi, M. A., & Athappilly, K. (2005). A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree (CART) models. Expert systems with applications. 29, 65-74.

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chicha-Olmo, M., & Chica-Rivas, M. (2015). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. Ore Geology Reviews.

Rokach, L. (2010). Ensemble-based classifiers. Artificial Intelligence Review. 33, 1-2.

Rotten Tomatoes. (no date). Rotten Tomatoes API. http://developer.rottentomatoes.com. [2015-06-10]

Scikit-learn. (No year). Documentation of scikit-learn 0.16.1. http://scikit-learn.org/dev/user_guide.html. [2015-07-07]

Segal, M. R. (2003). Machine learning benchmarks and random forest regression.

Smola, A. J., Schölkopf, B. (2004). A tutorial on support vector regression*. Statistics and Computing. 14, 199-222.

Statnikov, A., Wang, L., & Aliferis, C.F. (2008). A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics. 9, 1-10.

Vanwinckelen, G., & Blockeel, H. (2012) On estimating model accuracy with repeated cross-validation.

Verikas, A., Gelzinis, A., & Bacauskiene, M. (2011). Mining data with random forests: A survey and results of new tests. Pattern recognition. 44, 330-349.

Were, K., Bui, D., Dick, O. & Singh, B. (2015). A comparative assessment of support vector regression, artificial neural networks, and random forests for predicting and mapping soil organic carbon stocks across an Afromontane landscape. Ecological Indicators. 52, 394-403.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A. (2012). Experimentation in software engineering. New York, Springer.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P., Zhou, Z., Steinbach, M., Hand, D.J., Steinberg, D. (2008). Top 10 algorithms in data mining. Knowledge and Information Systems. 14, 1-37.

Yu, H., & Kim, S. (2012). SVM Tutorial — Classification, regression and ranking.