M.Sc. Thesis
Master of Science in Engineering

DTU Compute
Department of Applied Mathematics and Computer Science

Big Data and Success of Hollywood Movies

Enrique Yegros Miller (s171839)

Kongens Lyngby 2020

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Matematiktorvet
Building 303B
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary

The ever growing research that is being done on predicting success reflects the great interest surrounding the topic of motion-pictures. There are plenty of studies done on predicting motion-pictures success, but none seem to be center on success of featured actors. The main focus of this thesis is to understand and predict the success of an actor, defining the success of an actors as the actor having been featured in a motion-picture. This undertaking is accomplished by using a large dataset of motion-pictures extracted from the Internet Movie Database, utilizing Random Forest classification techniques and considering ordered sequences of the actors previews movies with information regarding each motion-picture including if the actor was featured in these movies or not. The results show that a Random Forest model where the hyperparameter have been tuned can predict if the actor will be featured in the next movie to certain degree, it also shows what are the most important features when considering this approach. This thesis will also focus on predicting the success of motion-pictures using Random Forest classification technique utilizing information pertaining to the motion pictures and classifying the motion-pictures according to the box-office it generates.

Preface

This Master's thesis was prepared at the department of Applied Mathematics and Computer Science at the Technical University of Denmark in fulfilment of the requirements for acquiring a Master's degree in Mathematical Modelling and Computation.

Kongens Lyngby, March 8^{th}, 2020

Enrique Yegros Miller (s171839)

Acknowledgements

Firstly, I want to thank my supervisors Sune Lehmann Jørgensen and Roberta Sinatra for their advice, support and patience over the course of the thesis. I also want to thank my friends Þorsteinn Gunnar Jónsson and Alma Salnaja for the support and help in proof reading my thesis. I want to give special thanks to my mother Melanie, uncle Patrick, brother Matias and the rest of my family for their unconditional support throughout my studies.

Contents

CHAPTER 1

Literature

The Lumière brothers were perhaps the ones that propelled the Motion Picture industry in their public screening of short films into a worldwide success, but now the movie industry alone generates billions of USD worldwide every year. In Figure 1.1 is possible to observe the Global Box Office from years 2014 to 2018.

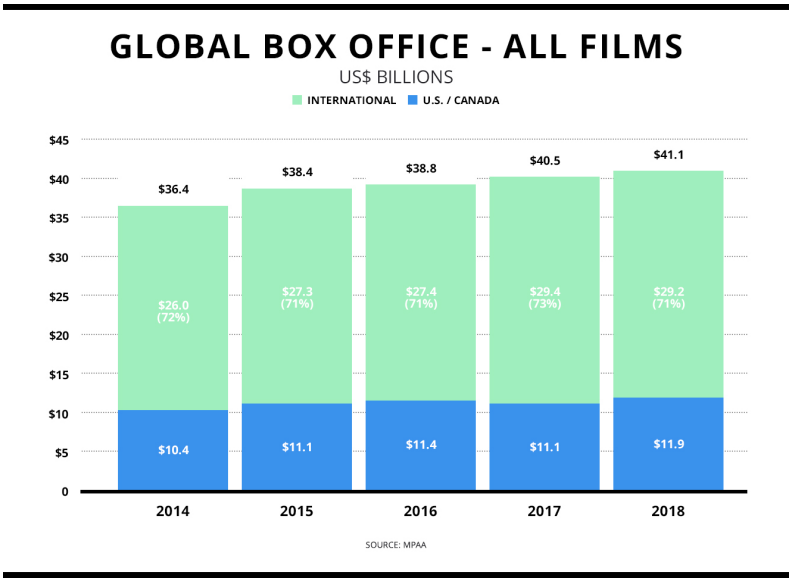


Figure 1.1: Global Box Office of all films from 2014 until 2018 [1]. The bars are divided into U.S./Canada and the rest of the world, International.

The monetary revenue incentive in the motion-picture industry makes production companies eager to produce motion pictures that will not only entertain the audience with a great film, but also maximize the monetary success of the films they produce. This motivation for success leads individuals and companies into the prediction of the next big motion-picture and the questions of what actors, directors, producers and other crew combinations are the most likely to produce a top rated film as well as a film that surpasses the cost of making and leaves the greatest profit. This also motivates actors to become successful and to be part of this motion-pictures, therefore

leading to the question of the factors involving the success of an actor.

The interest in success prediction in the motion-picture industry and of actors careers are not unique cases, there are plethora of fields where measuring and predicting success is of great interest, this also showing the growing curiosity in fame and success prediction topic. Wang et al. [2] talk about the success prediction of book sales before publication using a machine learning approach, they take into account author features, book features, publishing features and then quantifying the features and finding what are the forces contributing to the success of a book.

One approach towards early prediction of Box-Office success using a minimalistic predictive model constructed on the online user collective activity is done by Mestyán et al. [3]. In this paper they show how the popularity of a movie can be predicted prior to the release of the motion-picture by analyzing the degree of activity done by editors and viewers of the motion-picture Wikipedia page.

An IMDb rating prediction approach is done by Oghina et. al. [4] utilizing a cross-channel prediction task using various social media signals.

An special example is shown by Janosov et. al. [5] where they predict the success and luck in creative careers, they convey the idea of luck frequently influencing the career significance in the movie, music and book industries therefore improving the insight of the main elements in success prediction.

A different approach towards box-office revenue success related to motion-pictures and IMDb data is done by Sreenivasan S. [6]. In this paper Sreenivasan attempts to explore the revenue generated by a motion-picture using the influence of novelty scores based on the occurrence probabilities of keywords.

Pan et. al. [7] argue that the popularity of motion-pictures can be explained by, total income, the opening week income, and the income per week by theater thus suggesting that the success may be the outcome of a linear multiplicative stochastic process.

An attempt to quantify and predict success in show business is done by Williams et. al. [8]. They analyze a large database with information regarding films and television, they study the success of people working in the entertainment industry. They claim that two out of three actors are one-hit wonders, and that the success of an actor has a period of repeated successes. One of the main assertion made by Williams et. al. is that the highest productivity of an actor is at the start of their career and that the most productive years can be predicted.

The hitherto examples mentioned and the incentives that motivates the motion-picture industry as well as the actors impetus are the main propulsors to the enthusiasm towards predicting fame of actors careers and success of motion-pictures. The main focus of this thesis is understanding and predicting careers of stars in motion-pictures using data from the Internet Movie Database, the second focus is understanding and predicting the success of motion-pictures utilizing the features available in the data.

Predicting success is not a simple task, there are plenty of challenges and factors involved. One of the main issues with the undertaking of this project is the requirement of processing large amounts of data and to extract information pertaining thousands

of motion-pictures. After processing the data an additional hurdle arises, that is quantifying a score for actors, cast, crew, production company and others according to the data available. To achieve this a formula using the motion-picture score, a feature available in the data, was created and used in a similar way for the actors, cast and crew. The idea of using the cast order as a means to portray the fame of an actor was first consider, but inconsistencies in the data lead to other ways. Therefore to express and quantify if an actor is famous on a motion-picture the term "Featured Actor" is used, this meaning that an actor is featured if it appears in a list of stars of the motion pictures, this also is part of the data of each motion-picture.

This thesis will continue in the following manner to accomplish the main goal. In chapter 2 the theory behind the methods and models will be explained. In chapter 3 an exploration of the data will be presented in order to have a better insight of it, this will be followed by the manipulation of the data to constitute the format necessary for the machine learning method. The results, comments and a discussion regarding the results and methods will be presented in chapter 4. Finally in chapter 5 ideas on how to improve and possible future work will be discussed.

CHAPTER 2

Theory

In this chapter the machine learning models and methods to be used will be explained. The chapter will start with a brief explanation regarding decision trees in order to continue and describe one of the most useful methods, that is Random Forest.

The main motivation of machine learning is to interpret and recognize patterns in intricate data, create models in order to generate accurate predictions as well as to analyze the data. It is not enough to produce accurate predictions, the predictions need to be employed in smart ways for it to be useful.

There are plethora of classification algorithms used in machine learning that work better according to the data at hand, a few of them are. The research and the usefulness of machine learning algorithms and models lead to the creation and modification of models and algorithms. One of the most useful and popular models in the world is Random Forest [9], this is due to the versatility of the method and the accurate and reliable results one can obtained from it while been able to use different types of data with the algorithm.

Since Random Forest is plainly a collection of decision trees where the outcomes are combined into a final result, then to understand a Random Forest model first one needs to comprehend how decision trees function, a brief explanation follows.

2.1 Decision Trees

Decision Trees or just DTs are used for both classification and regression. A decision tree is a decision support tool that utilizes a model resembling a tree. They are a non-parametric supervised learning model since it does not make assumptions regarding the distribution of the data, therefore it yields a higher adaptability to handle different types of data-sets. Decision trees mimic a tree in their shape, and also similarly to a tree it follows a path from the root(root nodes) to the leaves(leaf nodes). The main objective of implementing a Decision Tree is to predict the class or value of the target variable using a training model by acquiring simple decision rules knowledge deduced from the training data.

Classification And Regression Tree (CART) [10] refers to both types of decision trees and was introduced by Leo Breiman et al. in 1984 [11]. The most common concepts used in Decision trees are the following,

- **Root Node:** It describes entire population or sample. It gets divided into homogeneous sets.
- **Splitting:** Action of dividing a node into sub-nodes.
- **Decision Node:** If a sub-node splits into more sub-nodes then this is referred as a decision node.
- **Leaf/Terminal Node:** The nodes that do not split are referred as Leafs or Terminal.
- **Pruning:** Discarding sub-nodes of the decision node is called pruning. The opposite of the splitting process.
- **Branch/Sub-Tree:** A part of the whole tree is referred as a branch or sub-tree.
- **Child and Parent Node:** If a node is divided into sub-nodes, then this is referred as a Parent node and the sub-nodes are the child nodes of the parent node.

There are two types of decision trees:

- Categorical variable decision tree, classification trees.
- Continuous variable decision tree, regression trees.

A basic decision tree is built by recursive partitioning, starting from the Root Node, where the node can be split into child nodes, the child nodes can be split even further and they become parent nodes to the new splits child nodes. The focus will be restricted to recursive binary partition.

2.1.1 Regression Trees

Suppose our data is composed of p inputs and a response, for every of N observations, then (x_i, y_i) for $i = 1, 2, 3, ..., N$ with $x_i = (x_{i1}, x_{i2}, x_{i3}, ..., x_{ip})$. The algorithms need to naturally decide on the splitting variables and split points. Assume that the data has been partitioned into M regions R_1 to R_M, and the response is modelled as a constant c_m in each region.

$$f(x) = \sum_{m=1}^{M} c_i \cdot I(x \in R_m). \tag{2.1}$$

Assuming as basis minimization of the sum of squares

$$\sum_{i=1}^{n} (y_i - f(x_i))^2, \tag{2.2}$$

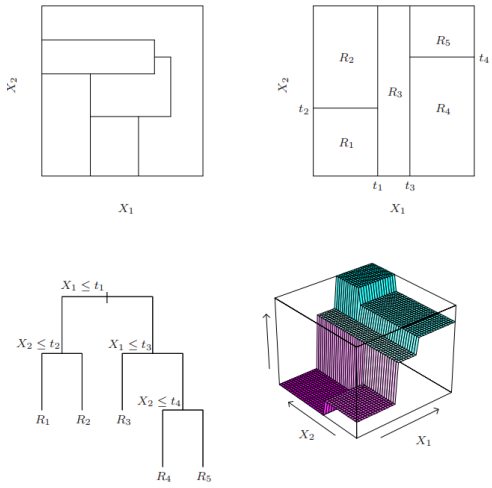


Figure 2.1: Partitions and CART [10]. The top left panel displays a general partition that is impossible to get with recursive binary partition. Top right panel displays a recursive binary splitting partition of a two dimensional feature space. The bottom left panel displays a tree equivalent to the partition on the top right panel. The bottom right panel shows a predictive surface.

then in order to minimize the sum of squares the best $\hat{c}_m$ needs to be chosen

$$\hat{c}_m = ave(y_i|x_i \in R_m), \tag{2.3}$$

thus the average of y_i in region R_m. Now which variable to split and where to split it needs to be consider. Finding the optimal binary partition when considering the minimum sum of squares is normally computationally infeasible. Therefore a greedy algorithm approach is considered. Consider a splitting variable $j \in 1, ..., p$ and splitting point $s \in \mathbb{R}$ and define the pair of half-planes:

$$R_1(j,s) = x \in \mathbb{R}^p : x_j \leq s, R_2(j,s) = x \in \mathbb{R}^p : x_j > s. \tag{2.4}$$

Then j and s are picked to minimize the following:

$$\min_{j,s}[\min_{c_1 \in \mathbb{R}} \sum_{x_1 \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2 \in \mathbb{R}} \sum_{x_1 \in R_2(j,s)} (y_i - c_2)^2]. \tag{2.5}$$

For any possible j and s, the inner minimization is done by,

$$c_1 = ave(y_i|\hat{x}_i \in R_1(j,s))\text{and}c_2 = ave(y_i|\hat{x}_i \in R_2(j,s)). \tag{2.6}$$

It is possible to do the determination of the splitting point s swiftly. Therefore deciding the optimal choice (j,s) is feasible. This process is repeated in the two

regions that were partitioned and so on all the following resulting regions. Now the main question is how large should a tree be, and when the process should stop. If a tree is too large it might over-fit the data and if the opposite is true then it might not capture the important structure. The size of the tree is a tuning parameter controlling the complexity of the model and the optimal size of the tree should be adjustable depending on the data. Generally the process of splitting the tree is stopped when there are five or less observations left in that region. The pruning process using *cost-complexity pruning* will now be described. Consider a large tree T_0 then let T be a sub-tree of T_0 thus $T \subset T_0$. Let $|T|$ be the number of terminal nodes in T, then for $\alpha > 0$ let

$$C_\alpha(T) := \sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}R_m)^2 + \alpha|T|. \tag{2.7}$$

Thus pick a sub-tree $T_\alpha \subseteq T_0$ for each α that minimizes $C_\alpha(T)$ where $\hat{y}R_m$ is the average response for observations in R_m. The trade-off between tree size and its goodness of fit to the data is controlled by the tuning parameter $\alpha \geq 0$. As α gets bigger the trees T_α become smaller and vice versa, when $\alpha = 0$ the full tree T_0 is obtained [10].

2.1.2 Classification Trees

Regression trees with continuous output were examined in the last subsection, now the technique will be adjusted in order to predict categorical output, that is classification trees. To achieve this only small changes need to be made to our splitting and pruning criteria in the tree algorithm. In the case of continuous variables a constant for each box R_i is chosen in order to minimize the sum of squares in that region.

$$\min_{c \in \mathbb{R}} \sum_{x_i \in R_i} . \tag{2.8}$$

In consequence the following is chosen,

$$\hat{c_i} = \frac{1}{N_i} \sum_{x_k \in R_i} y_k, \tag{2.9}$$

where N_i indicates the number of observations in R_i. This last approach is not applicable for classifications but similarly, when the output is categorical, it is possible to enumerate the proportion of class k observation in node m. Let R_m represent a region with N_m observations for a node m, then let

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k). \tag{2.10}$$

Thus the observations is classified in node m utilizing a majority vote,

$$k(m) = \arg\max_k \hat{p}_{mk}. \tag{2.11}$$

Different measures are usually used to ascertain how acceptable a given partition is and how to split a given partition thus:

Misclassification error:

$$\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk}(m). \tag{2.12}$$

Gini index:

$$\sum_{k \neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{2.13}$$

Cross-entropy or deviance:

$$-\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk}). \tag{2.14}$$

Considering two classes, let p be the proportion in the second class, then the measures mentioned above are the following:

$$\begin{aligned} \textbf{Misclassification error} &= 1 - \max(p, 1-p) \\ \textbf{Gini index} &= 2p(1-p) \\ \textbf{Cross-entropy or deviance} &= -p\log(p) - (1-p)\log(1-p) \end{aligned} \tag{2.15}$$

The three measures are similar, but Gini Index and Cross-entropy are differentiable, therefore susceptible to numerical optimization, shown in 2.2.

It is important to mention that either Gini Index or Cross-Entropy should be used when growing a tree, this is due to the fact that they are more sensitive to changes in the node probabilities than the misclassification rate. All three measures should be used in order to guide cost-complexity pruning, but normally the misclassification rate is used. There are many other factors and issues regarding introduction to Decision Trees focusing on classification and regression tree(CART).

2.1.3 Conclusion

The advantages of decision trees are many. Just to mention a few:

- Most of the times copies human decision making process.
- Easy to explain and interpret.
- Useful for both regression and classification.

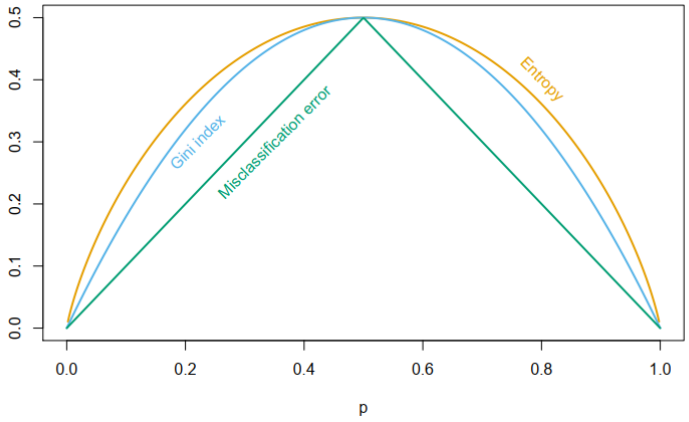


Figure 2.2: Node Impurity measures [10] for classification of two classes using proportion p as a function in class 2.

The are also some disadvantages,

- The most rudimentary implementation is usually not competitive when comparing to other methods.

Considering the advantages and disadvantages then aggregating many decision trees and utilizing other variants in order to improve the decision tree method needs to be examined. This is where Random Forest comes into play, but by aggregating many trees the straightforward interpretability of a decision tree might be more difficult or even lost in some cases.

2.2 Random Forest

Random Forest is a supervised learning algorithm that can be used for classification and regression. The forest part in the name gives us a clue regarding this algorithm and that it contains many trees. Decision trees were previously explained in section 2.1. The focus of this section will be ensemble method (bagging or bootstrap aggregation), that is Random Forest. The main concept behind bagging is to average plenty of noisy although generally unbiased models, to reduce variance. In the case of classification many such single decision trees are taken into account to make predictions. The algorithm creates de-correlated trees on randomly selected data samples, continues by obtaining the predictions from each tree in order to select the best solution by a manner of voting. For regression the same regression tree is fitted many times to bootstrap sample versions of the training data and then it averages the result. An ensemble method outperforms single decision trees since it reduces over-fitting

(thus solving one of the major significant disadvantages of the single decision tree) by averaging the result.

The Random Forest algorithm for regression and classification goes as follows (from "The Elements of Statistical Learning" by Hatie et al. [10]):

Algorithm 1 Random Forest algorithm for Regression and Classification.

1. For b = 1 to B:
   a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   b) Grow a Random Forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$

To make predictions at a new point x:
Regression: $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$
Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth Random Forest tree. Then $\hat{C}_{rf}^B(x) =$ majority vote $\{\hat{C}_b(x)\}_1^B$

For bagging, trees are the perfect contenders due to the fact that they are able to seize complex interaction structures in the data and have a rather low bias if grown deep enough. Every tree that is generated in bagging is identically distributed, therefore the expectation of B such trees is identical as the expectation of each one of them. With this in mind then the bias of bagged trees equals the bias of individual bootstrap trees, therefore variance reduction is the means to make it better. The Random Forest algorithm 1 has the purpose to increase the reduction in variance of bagging via the curtail of the correlation among trees while maintaining the variance from increasing excessively. To accomplish this it is important to observe at the random selection of input variables in the tree-growing step.
"Before each split, select $m \leq p$ for p input variables at random as candidates for splitting"[10]. The value of m would depend on the problem at hand, but in general there are some basic criteria that could be useful,

- **Classification:** In the case of classification the standard value of m is $\lfloor\sqrt{p}\rfloor$ and the lowest node size is one.
- **Regression:** In the case of regression the standard value of m is $\lfloor\frac{p}{3}\rfloor$ and the lowest node size is five.

2.2.1 Methods

2.2.1.1 Out of Bag Samples and Cross-validation

Out-of-bag (OOB) samples is a meaningful feature of Random Forest. The error estimate of OOB is alike one can produce using N-fold cross-validation. Therefore similarly to other nonlinear estimators it is possible to perform cross-validation on a single sequence fit of Random Forest while training the Random Forest. When the OOB error gets to a constant state, then the training can be stopped. From the The Elements of Statistical Learning by Hatie et al [10]:

"For each observation $z_i = (x_i, y_i)$, construct its Random Forest predictor by averaging only those trees corresponding to bootstrap samples in which z_i did not appear."

2.2.1.2 Feature Importance

It was mentioned that a single decision tree has a fairly straightforward interpretation, but it becomes more difficult when a Random Forest model is considered. In a Random Forest model the importance of the input predictor variables impacting the predicting response are crucial. It could be the case that just a few of the input predictors have an impact on the response, but then removing the non-important input predictors should be considered since it they cloud the model, it is also the case that adding more input variables helps the response. Hence, seeing the importance of these variables is a key part of Random Forest. Breiman et al. (1984) [12] suggest the following when considering a single tree to measure relevance of each predictor variable X_l,

$$I_l^2 = \sum_{t=1}^{J-1} \hat{i}_t^2 I(v(t) = l). \tag{2.16}$$

The summation is for all $J - 1$ internal nodes pertaining the tree. For each node t, a single variable input $X_{v(t)}$ is considered in order to partition the region associated with the node into two additional subregions and inside them another constant is fit to the response values. The chosen variable grants the maximal estimated improvement $\hat{i}_t^2$ in terms of square error risk over that for a invariant fit over the whole region. For variable X_l the square relative importance is the sum of such square improvements for all internal nodes that it was chosen for as the splitting variable. The importance measure is generalized in the following manner for additive tree expansions.

$$I_l^2 = \frac{1}{M} \sum_{m=1}^{M} I_l^2(T_m). \tag{2.17}$$

In the case of multiple classes K-class. K independent models $f_k(x)$,$k = 1, ..., K$ are induced, and every one consisting of a sum of trees

$$f_k(x) = \sum_{m=1}^{M} T_{km}(x). \tag{2.18}$$

This can be generalized to

$$I_{lk}^2 = \frac{1}{M} \sum_{m=1}^{M} I_l^2(T_{km}). \tag{2.19}$$

The relevance of X_l in separating the class k observations from the rest of the classes is I_{lk}. By averaging over all the classes the overall relevance of X_l is obtained.

$$I_l^2 = \frac{1}{K} \sum_{k=1}^{K} I_{lk}^2. \tag{2.20}$$

Compared to Boosting where some of the variables are ignored entirely, Random Forest does not. At every single split of each tree, the refinement in the split-criterion is the importance measure attribute of the splitting variable, this is accumulated over all trees in the forest independent for every variable The candidate split-variable selection enlarge the chances that any single variable becomes included in a Random Forest.

2.2.1.3 Random Forest and Overfitting

The Random Forest model could perform poorly with a small m if the number of variables are numerous, but the number of important variables are few. This is because the chances of selecting the useful variables are low at each split. As the number of relevant variables grows the robustness of the Random Forest performance surges to an increase in the number of noise variables. From the The Elements of Statistical Learning by Hatie et al [10]:

"This robustness is largely due to the relative insensitivity of misclassification cost to the bias and variance of the probability estimates in each tree."

2.2.1.4 Measures of performance

Once the model is built, evaluating how good the model performs is perhaps the foremost important question. Therefore evaluating the model performance is one of the most crucial tasks of any project. In order to understand the accuracy, recall, ROC/AUC and F1 first it is important to understand the confusion matrix, since the other measures are calculated using this matrix.

True Positives (TP) - The correctly predicted positive values. The values of class that is yes and the predicted class is also yes.

	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Figure 2.3: Confusion Matrix depicting True Positives, True Negatives, False Positives and False Negatives.

True Negatives (TN) - The correctly predicted negatives values. The values of class that is no and the predicted class is also no.

False Positives (FP) – The incorrectly predicted negative values. The values of class that is no and the predicted class is yes.

False Negatives (FN) – The incorrectly predicted positive values. The values of class that is yes and the predicted class is no. The following measures are calculated using the confusion matrix values shown in Table 2.3 ,

- Accuracy: The accuracy is the most intuitive performance measure. It is the ratio of observations correctly predicted to the total number of observations. This is a great tool to use when the data is balanced.

$$\frac{TP + TN}{TP + FP + FN + TN} \tag{2.21}$$

- Precision: The precision refers to the ratio of positive (negative) observations that are correctly predicted to the total number of predicted positive (negative) observations.

$$\frac{TP}{TP + FP} \tag{2.22}$$

- Recall: The Recall is also called Sensitivity and Probability of Detection. The Recall is the ratio of positive (negative) predicted observations that were classify correctly to all the observations in the class.

$$\frac{TP}{TP + FN} \tag{2.23}$$

- F1 score: The F1 score refers to the weighted average of both the Precision and Recall. If there is an uneven class distribution the F1 score is a better metric than Accuracy, that is if the cost of false negatives and false positives differs by a considerable value then the Recall and Precision are the better metrics and hence F1 score since it combines both of them.

$$\frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \tag{2.24}$$

- ROC - AUC: ROC stand for Receiver Operating Characteristic curve and AUC for the Area Under the ROC curve. The AUC provides an aggregated performance measure across all possible classification thresholds, therefore the higher the AUC score, or area under the ROC, the better the classifier performs. To generate the ROC-AUC curve the True positive rate (TRP) or Recall and the False Positive Rate (FPR) are needed. The $FPR = 1 - \text{Specificity}$. The Specificity is defined as follows,

$$\frac{TN}{TN + FP} \tag{2.25}$$

  The ROC-AUC is graphed by plotting the TPR against the FPR. The closer the AUC score is to the value of 1 the better the model is distinguishing between the 2 classes.

CHAPTER 3

Data

In order to do predictions data is necessary. In this chapter, the data will be introduced. How the data was obtained, cleaned and manipulated will be discussed, also an exploratory data analysis will be done to exhibit any patterns and to have a better understanding of the data.

3.1 Data Acquisition

The original IMDb [13] data encompassed 703242 files, with each text file containing information about a single movie. An example of one motion-picture can be seen in in Table 3.1, it is important to point out that not all files were this complete or clean, some of the text file had the wrong information in some lines or in some cases this information was non-existent. This data-set was scraped from the Internet Movie Database (IMDb) website over the summer of 2017 by Milan Janosov [5] and shared in August 2019. The text files obtained were approximately 3.2 gigabytes.

The basis to select files to scrape from the extensive library of the IMDb website was if the movie, series or video possessed a rating value, IMDb score. An analysis of the text files showed that some of them were missing information:

- 8648 text files were completely blank, had no movie ID or had only information regarding the crew (cast, director, producer, etc).
- 50758 text files were missing cast.
- 70451 text files were missing director.

There are certain information/variables that we deemed important in order to train a random forest model, some of the files lacking this information were not taken into consideration and a subset of the whole data set was used. There is some overlapping between the files that were fully blank and the ones containing no director and no cast. The total number of overlapping files are 2903. The total number of files missing information totaled 127327, thus the number of files we consider were 575901.

movie_id	tt0120737
title	Le seigneur des anneaux: La communauté de l'anneau (2001) - IMDb
original_title	The Lord of the Rings: The Fellowship of the Ring (original title)
review_count_user	5075
review_count_critic	295
metascore	92
rating_value	8.8
rating_count	1266990
date_published	2001-12-19
duration	2h58min
titleYear	2001
summary_text	A meek Hobbit from the Shire and eight companions set out on a journey to destroy the powerful One Ring and save Middle Earth from the Dark Lord Sauron.
genre_tags	Adventure Drama Fantasy
plot_keywords	ring quest hobbit elf orc
writer_names	J.R.R. Tolkien (novel) Fran Walsh (screenplay)
writer_ids	nm0866058 nm0909638
director_names	Peter Jackson
director_ids	nm0001392
star_names	Elijah Wood Ian McKellen Orlando Bloom
star_ids	nm0000704 nm0005212 nm0089217
Country	New Zealand|USA
Language	English|Sindarin
Release Date	19 December 2001 (France)
Filming Locations	Fort Dorset, Miramar, Wellington, New Zealand
Budget	$93,000,000 (estimated)
Opening Weekend	£11,058,045
Gross	$313,837,577 (USA)(5 December 2003)
Production Co	New Line Cinema, WingNut Films, The Saul Zaentz Company
Runtime	178 min|208 min (Special DVD Extended Edition) |228 min (Blu Ray Extended Edition)
Color	Color
Aspect Ratio	2.35 : 1
Motion Picture Rating	None
. . .	. . .
Directed by	nm0001392
Writing Credits(WGA)	nm0866058 nm0909638 nm0101991 nm0001392
Cast(in credits order) verified as complete	nm0397102 nm0032370 nm0000276 nm1019674 nm0000293 nm0000949 ...
Produced by	nm0001392 nm1088153 nm0649507 nm0651614 nm0691815 ...
Music by	nm0006290
Cinematography by	nm0504226
Film Editing by	nm0003016
Production Design by	nm0538194
Art Direction by	nm0088054 nm0377172 nm1134464 nm0653695 nm0732314
Costume Design by	nm0225699 nm0853050
Makeup Department	nm0040069 nm0277515 nm1182298
. . .	. . .

Table 3.1: Data text file example for "The Lord of the Rings: The Fellowship of the Ring". Here we show the best case of a text file data where all of the information was available. There are omitted parts since they did not contain any pertinent information.

3.2 Data Exploratory Analysis

In the last section it was discussed how the data was obtained, also the data was examined for missing information. In this section an exploratory analysis of the data will be done in order to find patterns, anomalies and recognize the behavior of the variables of the motion-pictures. After reading the numerous lines in all 575901 text files the information that will form the variables necessary for the machine learning models were extracted and saved in a Pandas data-frame [14], each row in the data-frame represents a movie. The following are the information extracted, the columns of the data-frame followed by an explanation of each.

- **ID:** The movie unique Id identifier, e.g. $tt0120737$.
- **cast:** List of unique cast ID's, e.g. $[nm0397102, nm0032370, ...]$.
- **director:** List of unique directors ID's, similar to cast.
- **producer:** List of unique producers ID's, similar to cast.
- **writer:** List of unique writers ID's.
- **cinematography:** List of unique Cinematographer ID's.
- **metacritic:** Metacritic [15] numerical score for the movie, value from 0 to 100.
- **imdbScore:** Aggregated numerical rating value giving by IMDb users, value from 0.0 to 10.0.
- **boxOffice:** Gross earnings in U.S. dollars. This amount does not account for inflation.
- **budget:** Cost of making the motion-picture, it is the cost of producing/shooting the motion-picture and in some cases other expenses related to the motion-picture. This amount does not account for inflation, and is important to specify that only the amounts in U.S. dollars were consider.
- **genre:** The genres of the motion picture.
- **star:** List of the 3 stars featured in the motion picture.
- **alph0_nonalph1:** Binary value showing whether the cast is in alphabetical order, 0, or non-alphabetical order,1. Non-alphabetical meaning that the first 3 in the list are the featured actors shown in 'stars' and continue with the most known actors to less known actors at the time the film was released.
- **date_released:** Date year-month-day of the released of the film as it appears in the text files. If no month or day exist January-01 is used. This is used as the based to have the films in chronological order.

- **country:** Country or countries where the production companies for the title are based, and therefore where the financing originated.
- **plot_keywords:** Main words describing the motion picture.
- **no_reviews_imdb:** The number of people that voted for the title, contributing to the aggregated IMDb score.
- **no_reviews_critics:** The number of professional critic reviews from newspapers, magazines and other publications regarding the title.
- **no_review_users:** The number of user reviews where IMDb users explained the reasons why liking or disliking a title.
- **production_co:** List of the production companies involved in the motion-picture.

The cleaned data set after removing blank files, files with no cast and files with no directors consists of 575901 motion-pictures. We created subsets of the dataset out of the total number of files/motion-pictures/data-points since a big number of them were missing data in some of the columns mention above. An overview of the number of files is shown in Table 3.2. It is important to point out that each data set mention is a subset of the preceding one.

Data set type	Number of titles
Motion-Pictures with IMDb score (full-data set)	581848
Motion-Pictures with Box Office values	12527
Motion-Pictures with Metacritic score	8111
Motion-Pictures with Budget	6030

Table 3.2: Showing the full data set 'Data with IMDb score (full-data set)' (The one after removing files with no cast, directors or blank text files) and the different subsets created with them with more complete information regarding the title motion picture/data point. 'Data with Box Office values' is the data set containing IMDb score and box office values, but missing metacritic score and the titles budget. 'Data with Metacritic score' is the sub-data set containing IMDb score, box office values and metacritic score, but missing budget for the title. Finally Data with Budget is the one containing all values mention above in the description of the columns of the data set.

Now that the data and the subsets of the data are in a data-frame it is possible to analyzing it.

The exploration of the data will start with 'plot_keywords', and for this the plot keywords will be visualized using 'Wordcloud' [16] to see the most significant and most frequent words for all the motion-pictures, thus it is possible to visualize it in Figure 3.1.

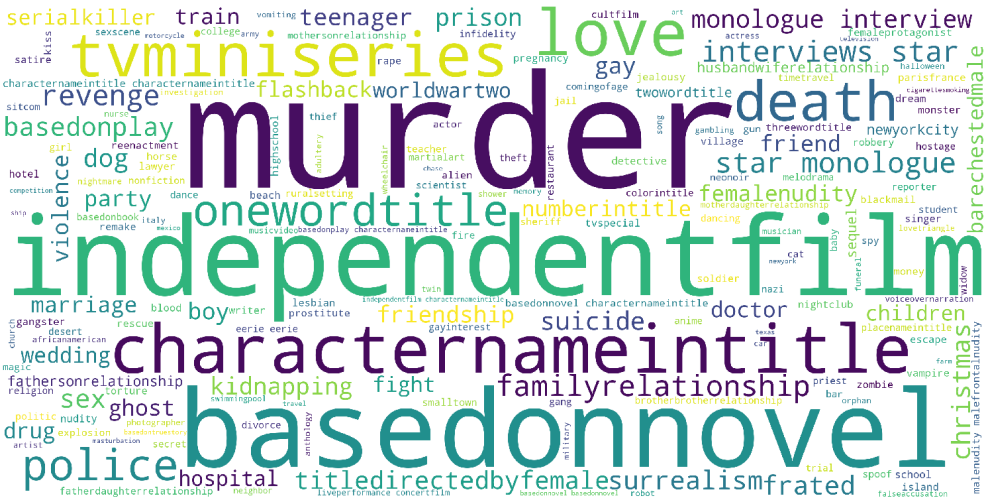


Figure 3.1: Wordcloud of the plot-keywords for the 581848 motion-pictures showing the most important and frequent words used.

The wordcloud for all the motion-pictures shows that the most important and frequent word is murder' followed by 'character name in title', 'based on novel' and 'independent motion-picture'. It is interesting to see what the plot for majority of motion-pictures are based on, though for our models we do not use the plot-keywords this data was available to us.

Now the focus will be given to the distribution of motion-pictures according to country per year, to be more precise the country where the production companies and the majority of funding of the motion-picture are based at. Our data-set encompasses a total of 229 countries, including countries that no longer exist, such as 'The Soviet Union', 'West Germany', 'The Federal Republic of Yugoslavia' and 'Korea'. It also includes territories belonging to other countries, such as 'Puerto Rico' and 'Guadeloupe'. The range of years for motion pictures of our data set starts at 1888 and ends in 2018, with only 26 motion-pictures from 2017 and 1 motion picture from 2018. As previously mentioned, the total number of motion pictures is 581848.

A table is shown in Figure 3.2 some countries, with the years as index and the number of motion pictures per year.

The table shown in Figure 3.2 shows a number of motion-pictures made by many countries that no longer exist, even some motion-pictures made after their dissolution. This could be explained by motion-pictures that have been used in other motion-pictures or films that have been remastered.

The table shown in Figure 3.3 shows the total number of motion pictures made by country. Not all the countries are shown in the figure, but a subset of them with the countries with the most motion-pictures.

Index	USA	UK	France	Canada	Germany	India	Japan	Italy	Spain	Australia	West Germany
1984	1413	438	192	106	3	221	105	114	91	135	191
1985	1761	483	191	149	0	242	123	103	83	100	154
1986	1752	468	192	141	4	222	140	116	64	103	146
1987	1948	450	196	195	3	192	134	145	73	98	160
1988	1860	486	212	207	6	217	124	184	81	126	159
1989	2117	538	222	225	5	249	173	176	87	77	178
1990	2152	487	231	204	114	226	142	156	114	88	58
1991	2080	567	255	195	188	240	149	146	113	62	9
1992	2134	563	312	225	239	246	155	134	106	78	4
1993	2307	571	296	223	269	251	162	126	82	82	1
1994	2611	584	323	307	286	231	173	120	88	104	1
1995	2893	649	351	346	355	210	205	123	114	109	0
1996	3181	590	376	418	367	196	258	147	132	101	0
1997	3472	683	372	448	452	220	232	150	120	106	0
1998	3662	700	410	505	490	221	274	164	132	123	0
1999	3870	696	421	608	583	232	291	188	136	147	1
2000	3979	798	502	611	624	206	280	185	176	159	1
2001	4395	929	535	655	689	235	362	207	193	215	0
2002	4935	1003	598	752	714	263	425	205	243	220	0
2003	5046	1109	641	711	708	299	448	221	261	236	0
2004	5606	1409	654	781	791	310	499	269	341	255	0
2005	6381	1653	629	790	852	355	566	241	378	285	1
2006	7808	1743	679	1040	931	359	735	254	335	514	1
2007	7616	1879	673	951	954	360	614	275	354	496	0
2008	8345	1838	679	1073	968	392	567	263	371	399	2
2009	9681	1998	787	1040	983	432	576	319	391	420	1
2010	9157	2000	830	939	991	500	548	293	449	502	2
2011	9373	2146	818	1081	950	515	613	318	465	530	1
2012	9051	2138	870	1106	938	568	533	344	442	455	3
2013	9186	2031	872	1115	942	674	523	311	452	501	4
2014	8938	2004	875	1019	747	708	517	330	367	373	1
2015	8291	1962	840	1043	720	706	545	320	353	369	1

Figure 3.2: This table shows the number of motion-pictures some of the countries in the dataset made throughout the years. It is interesting to see how many motion pictures countries that no longer exist made and how it drastically changed after their dissolution. The color scheme in the table represents higher and lower values, the darker (purple) the color the higher the value, the lighter pink to red/orange, the lower the value. The color range is unique for each column, thus representing the contrast for the years the country made the most movies.

Country	Number of Movies	Country	Number of Movies
USA	215357	Norway	2246
UK	47984	Ireland	1995
France	23697	Iran	1808
Canada	21605	Czechoslovak…	1758
Germany	18879	Israel	1622
India	14939	Yugoslavia	1596
Japan	14712	Romania	1558
Italy	13009	Taiwan	1444
Spain	10436	Czech Republic	1429
Australia	8897	Egypt	1424
West Germany	6802	New Zealand	1193
Mexico	5453	Croatia	1162
Sweden	5272	South Africa	1008
Turkey	4928	Indonesia	983
Soviet Union	4755	Bulgaria	949
Netherlands	4562	Thailand	831
Hong Kong	4471	Chile	822
Brazil	4392	East Germany	761
Denmark	4069	Colombia	753
Argentina	4031	Serbia	620
Russia	3646	Iceland	541
Greece	3503	Malaysia	527
Finland	3477	Venezuela	500
Belgium	3421	Ukraine	492
Austria	3131	Pakistan	487
Poland	2892	Singapore	419
South Korea	2829	Luxembourg	395
Hungary	2565	Bangladesh	376
Switzerland	2441	Cuba	372
Philippines	2431	Slovenia	366
China	2322	Estonia	351
Portugal	2249	Bosnia and Herzegovina	318

Figure 3.3: Total number of motion pictures made by individual countries, the are countries that not longer exist or changed names in the list. The color range of the table represents the higher and lower values, the darker, purple, the color bigger the value and the opposite for the lighter going from pink to red. The columns are in descending order with the second column been the continuation of the first. It is possible to see how the USA is the leading country in the overall making of motion-pictures by more than 4 times the following country.

The graph in Figure 3.4 shows an area graph with 19 countries from years 1950 to 2016 with the area with the number of motion-pictures they made in this time frame. It is another way to visualize the distribution of motion-pictures for countries per year.

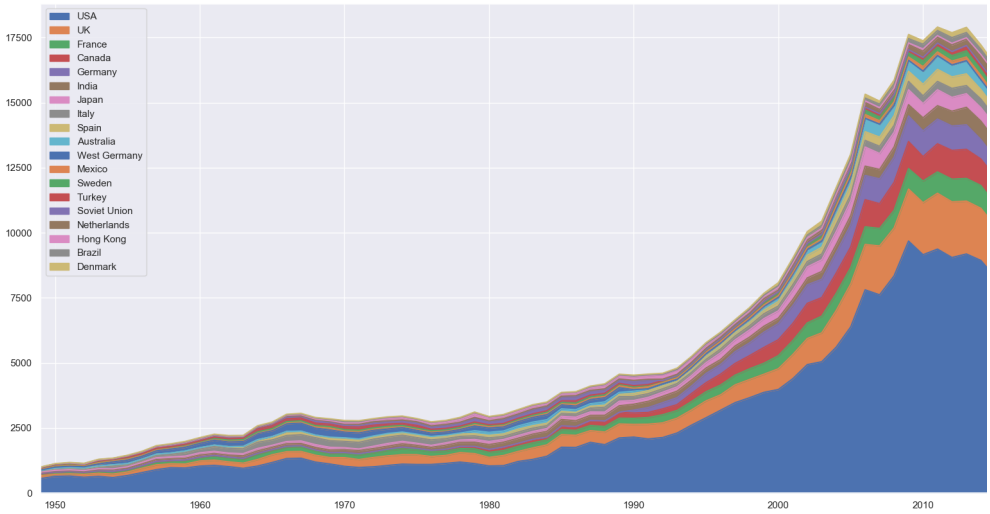


Figure 3.4: Area graph for the number of motion-pictures for 19 countries from year 1950 to 2016. We can see that the USA is the leading country in terms of making motion-pictures followed by the UK, France and Canada. Also, it is interesting to see when West Germany stops making motion pictures and when Germany starts making them.

The graph in Figure 3.5 shows the world map with a heat-map of the numbers of motion-picture each country made, since the USA has the biggest number of motion-pictures made compared to the rest of the world we used a log transformation in order to make the data visualization less skewed [17].

When looking at Figure 3.5 we notice that the USA is the leading country when it comes to making motion-pictures, this is stated already in previews paragraphs. We can also notice that 'The Republic of South Sudan' and 'The Republic of Kosovo' appear to be countries with a high number of motion-picture, white color, but this is due to the fact that there is still some conflicts in South-Sudan and Kosovo and the map does not take them into account yet. Although in this regard the map does include Palestine (only West Bank) as a country, Palestine having 101 motion-pictures. The other interesting fact from the Figure is that Antarctica seems to have produced a few movies, three to be exact. After examining at the Motion-Picture in the IMDb website [13] for the motion-pictures Antarctica appears as country, it is obvious to see the error made by users inputting the information. The filming location is Antarctica, but it should not be mentioned in the Country section since this reflects where the

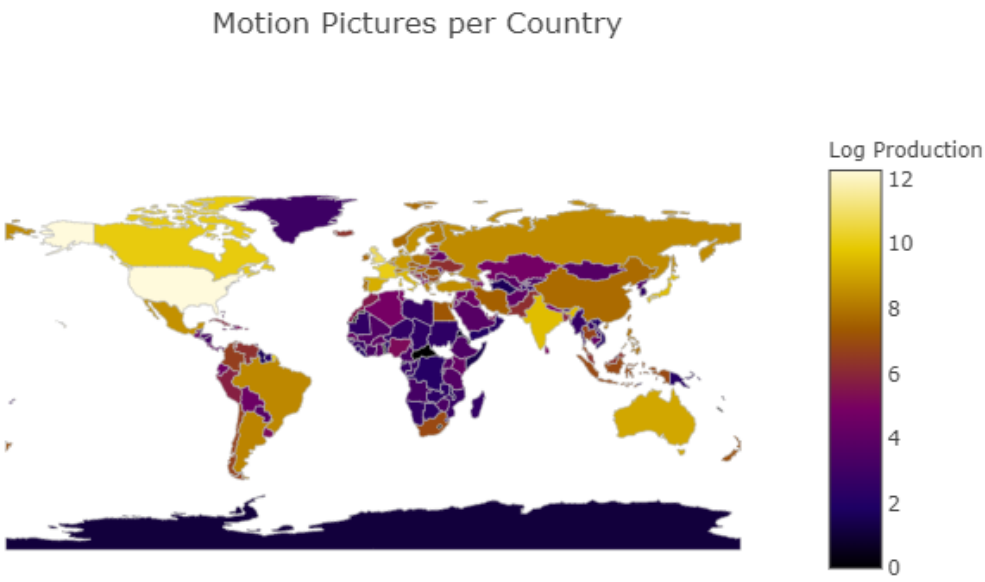


Figure 3.5: This is a good representation of the distribution of the number of motion-picture made per country in total. The total number for each country has been normalized with a log transformation since the difference between the total number of motion-pictures between the some of the countries are overwhelming, see Table 3.3. This map only shows current countries and does not account for countries that no longer exist.

funding for the motion-picture originates and where the production companies are located.

Next we will investigate the different genres of the full dataset. The unique genres of the motion-pictures from our dataset are the following: **“Action”, “Adult”, “Adventure”, “Animation”, “Biography”, “Comedy”, “Crime”, “Documentary”, “Drama”, “Family”, “Fantasy”, “Film-Noir”, “Game-Show”, “History”, “Horror”, “Musical”, “Music”, “Mystery”, “News”, “Reality-TV”, “Romance”, “Sci-Fi”, “Short”, “Sport”, “Talk-Show”, “Thriller”, “War”, “Western”.** We can observe the distribution of genres across all motion-pictures, shown in Figure 3.6.

We also want to investigate how the different genres do over different years. Figure 3.7 shows the distribution of genres for motion-pictures from 1964 to 2016. We choose these years since they are the years with the most motion-pictures.

The next thing to investigate is the ‘Gross’ or ‘Box Office’ amounts for the motion-pictures as well as the ‘Budget’. It was mentioned before that the total number of motion-pictures with ‘Box Office’ values were 12527 and the ones with ‘Budget’ were 6030. The graph shown in Figure 3.8 shows Box-Office for the motion-pictures that

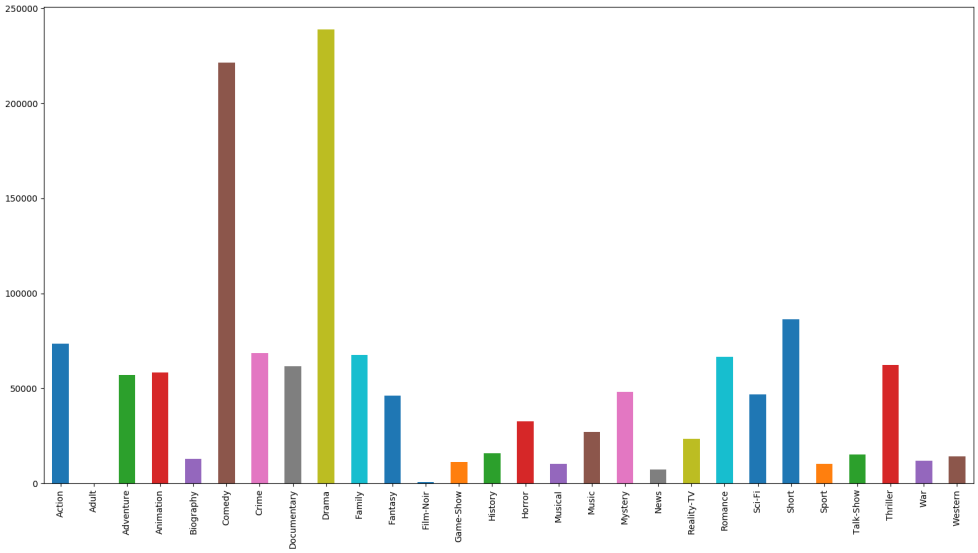


Figure 3.6: Distribution of all motion-pictures for genres, a motion picture can have multiple genres. Comedy and Drama surpass the other genres by a large margin. Adult only have a few motion-pictures, therefore it does not appear in the figure

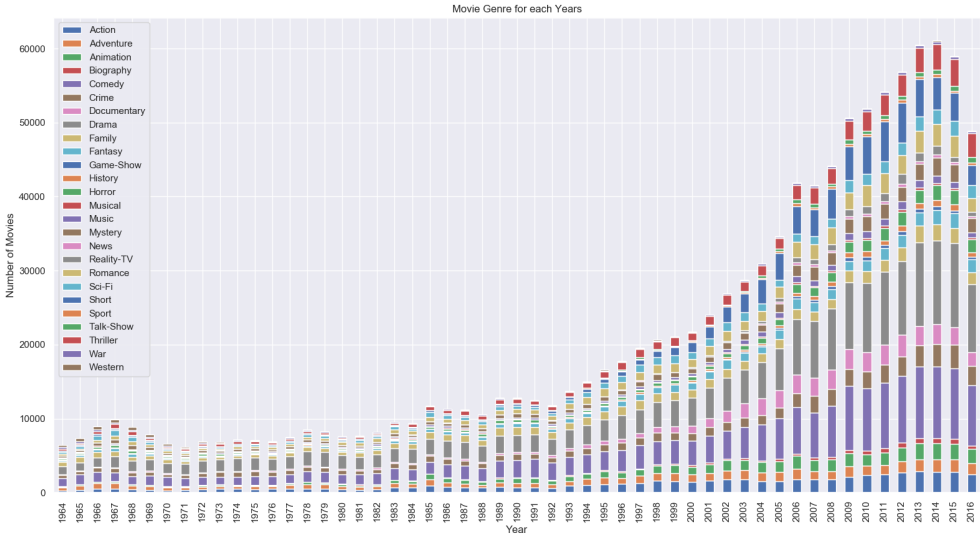


Figure 3.7: Genre distribution for motion-pictures between the years 1964 and 2016. The bottom of the stacked bars shows action, then adventure then animation and so on. As we can observe and stated before drama and comedy are the bigger ones for any year and as the years progress the number of motion-picture increased drastically.

have this value and also includes the Adjusted Box-Office to adjust the value for inflation to 2019 prices, this was accomplish using the package CPI [18] in Python. Similarly the adjusted budget was found and modified to year 2019 as base using the CPI package [18] the graph depicting this Figure 3.9. It is important to mention that there were many more motion-pictures that contained budget information, but the budget information of these motion-pictures were not in USD. It was possible to translate some of the currencies into a single one, USD, but it was not possible to account for inflation since there is not a consumer price index for all countries and currencies, including countries or currencies that no longer exist.

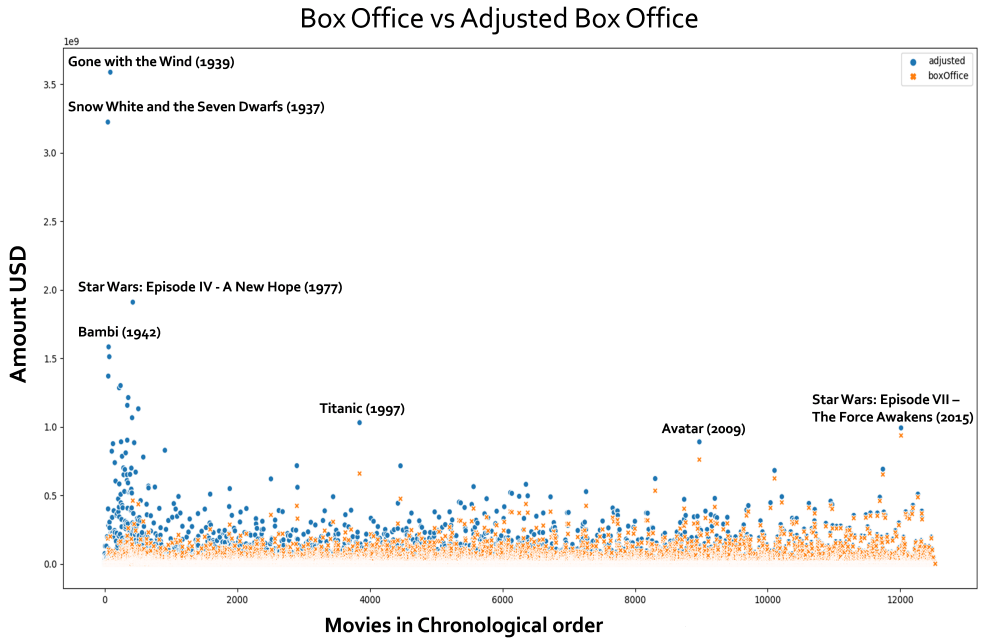


Figure 3.8: The Box-Office of motion-pictures in orange with their respective Adjusted-Box-Office accounting inflation. The movies are ordered according to the released year therefore we can see as we continue going to the right the difference between the regular Box-Office and Adjusted is diminishing since we get closer to the current value of USD$. We can observe that the motion-picture "Gone with the Wind" from 1939, has the highest adjusted Gross/Box Office at around 3.4 billion USD, the original Box-Office of the motion picture is around 400 million USD.

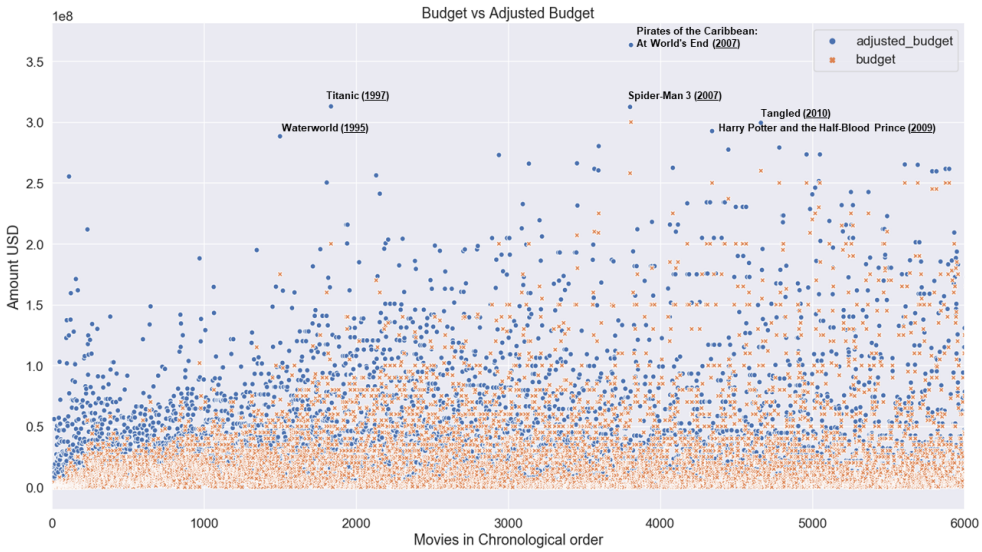


Figure 3.9: The Budget of motion-pictures in orange with their respective Adjusted-Budget accounting inflation in Blue. The movies are ordered according to the released year appearing in the text files therefore we can see as we continue going to the right the difference between the regular Budget and Adjusted Budget is diminishing since we get closer to the current value of USD$. We can observe that the motion-picture "Pirates of the Caribbean: At World's End" from 2007, has the highest Adjusted Budget at around 360 million USD, the original Budget of the motion picture is around 300 million USD.

3.3 Data Preparation

It was shown where the data was obtained, the number of files that were missing crucial information deemed important for our task as well as an exploratory data analysis to find patterns and anomalies. We also shown all the different sub-sets of the dataset we could create with the information available in the files. Now the data will be manipulate in order to fit into our machine learning method.

The next step with the data is to give a score to every actor, director, producer, writer, cinematographer and production company for every single motion-picture. This is accomplished by using the IMDb score of all previous motion-pictures done by and individual (actor, director, writer, etc) until that point. Thus the score for the N^{th} motion-pictures is based on $N-1$ motion-pictures, if it is the first motion-picture then we used the average of all IMDb scores of all motion-pictures in the dataset, we denote this average as 'IAS'(IMDb Average Score). The total number of titles of any person involved in a motion picture is denoted as M. Rigorously, we compute X_N^i,

the score of the N^{th} movie of individual i as:

$$X_N^i = \begin{cases} \text{IAS}, & \text{if N} = 1 \\ \frac{\sum_{n=1}^{N-1} X_{N-1}^i}{N-1}, & \text{if } 2 \leq \text{N} \leq \text{M} \end{cases} \tag{3.1}$$

In case of the production company, we create a dictionary with all the production companies mentioned in the full-dataset and count the number of motion-pictures in chronological order, thus for each motion-picture we can give a score of the total number of motion-pictures the production company has done thus far. If there are multiple production companies, we use the value of the production company with the highest number of motion-pictures done until that point. We use the scores of individuals to create a dictionary for the actors, using the actors unique identifier as keys and a list of all motion-pictures the actor did as values, with every motion-picture having the following numerical information,

- **movie_id:** Motion-picture unique identifier.
- **IMDb:** IMDb aggregated score.
- **meta:** Metacritic score.
- **actor_score:** Score-value for actor.
- **cast_score:** Average-score of entire cast, looking at the scores for each single one until this point in time and finding the average.
- **director_score:** Score value for director or average-score if multiple directors (similarly to cast_score).
- **writer_score:** Score value for writer or average-score if multiple writers (similarly to cast_score:).
- **producer_score:** Score value for producer or average-score if multiple producers (similarly to cast_score).
- **cinematography_score** Score value for cinematographer or average-score if multiple cinematographers (similarly to cast_score).
- **star_1:** Score given to first actor in the list of feature stars of the motion-picture.
- **star_2:** Score given to second actor in the list of feature stars of the motion-picture.
- **star_3:** Score given to third actor in the list of feature stars of the motion-picture.

- **score_production_co**: Production company score.
- **no_reviews_imdb:** The number of people that voted for the IMDb score.
- **no_reviews_critics:** The number of professional critic reviews from newspapers, magazines and other publications regarding the title.
- **no_reviews_users:** The number of user reviews, written essays, where IMDb users explained the reasons why liking or disliking a title, in some cases offering some criticism.
- **budget:** Cost of making the motion-picture in U.S. dollars adjusted for inflation.
- **box_office:** Gross/Box Office earnings in U.S. dollars adjusted for inflation of the motion-picture.
- **feature_binary:** Binary value signifying whether an actor is featured or not in a motion-picture, Equation 3.2.

Every single item mentioned above has a single numerical value, thus one for every motion-picture and it differs from the items mentioned at the start of section 3.2 as they contain lists of the identifiers of the individuals that are part of the motion-picture as well as information regarding a motion-picture, i.e. genre, released date, country, keywords of the plot.

The 'binary_value_actor' is declared in the following manner: $star_{j^N}$ is the list of three actors j featured in the motion-picture N. X_N^i is an actor i in the cast list of motion-picture N thus:

$$\text{binary\_value\_actor} = \begin{cases} 1, & \text{if } X_N^i \in star_{j^N} \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

An example of the binary features for some famous actors for the motion-pictures they have done can be seen in Figure 3.10, and an example of the binary feature of random selected actors can be seen in Figure 3.11 for all the motion pictures they have done in our dataset. The original idea was to use the cast list order as hierarchy since most films have the main actors of the motion-pictures on top, but there are inconsistencies with some motion-pictures using an alphabetical order, this lead to the use of the star list as 'feature actor'.

We have given a numerical score to most of the variables in our data in order to train and test models with it. In the following chapter we will be showing some of the results we obtained using this data.

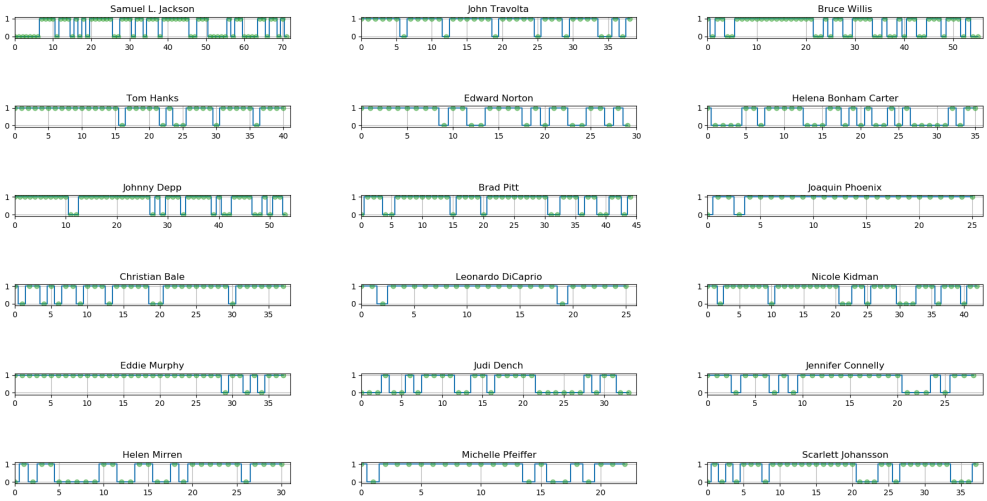


Figure 3.10: Here we can see the binary features of famous actors for the motion-pictures they were part off in chronological order. If the value is zero then the actor was not in the list of stars of the motion-picture, hence not featured, on the other hand if the value is one then the actors were in list of stars of the motion-picture therefore featured.

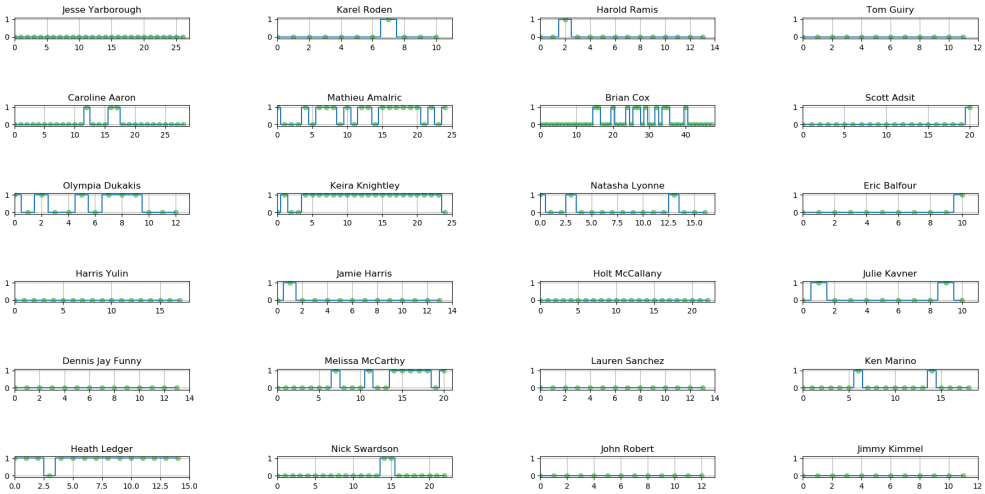


Figure 3.11: Here we can see the binary features of random actors for the motion-pictures they were part off in chronological order. If the value is zero then the actor was not in the list of stars of the motion-picture, hence not featured, on the other hand if the value is one then the actors were in list of stars of the motion-picture therefore featured.

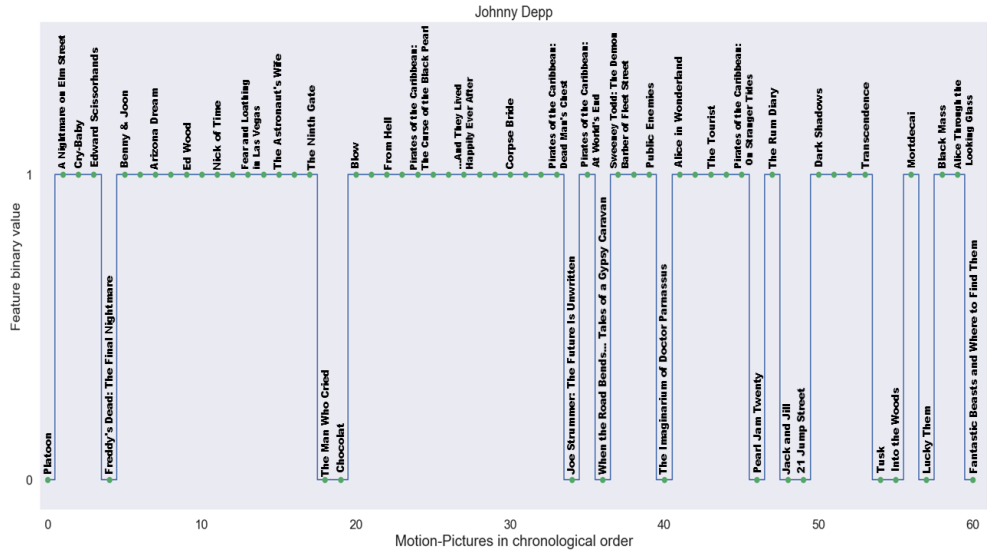


Figure 3.12: The binary features of our data for the acting career of 'Johnny Depp' with the motion-pictures where he was featured (1) and the ones where he was not featured (0). The motion-pictures are in chronological order according to the released date that appears in the text files. The number of motion-pictures showing here differs from the one in Figure 3.10, this is the subset of the data set that contains box office information.

CHAPTER 4

Results and Discussion

In the previous chapter the characteristics of the data have been shown and how it was processed in order for it to be used in Random Forest models. In this chapter the models will be applied and discussed, they will be trained and tested and the results will be presented alongside with some comments regarding the results and models. This chapter will be divided into 3 main sections; Actors, Motion-Pictures and a general discussion. In the Actors section we will focus on the dictionaries of actors (with all the motion-pictures each actor made) we created for the different subsets of dataset. In the second section we will look at the motion pictures dataset with all its features. The models are created using the scikit-learn [19] package, for the data handling pandas package [14] was used, the results were graph using Matplotlib [20].

The metrics that will be used to measure the models are presented with a small description of them,

- Accuracy: To see the overall number of correct predictions the model obtained.
- OOB score: As a validation metric of our model. To obtain error estimates of the Random Forests model.
- AUC score: To check the overall performance of the model, to measure the models capabilities of distinguishing between classes, the ROC-AUC curve is also used for most models.
- Precision: It is the ratio of correctly predicted class observations to the total predicted class observations.
- Recall: It is the ratio of correctly predicted of the class observations to all observations in actual class.
- F1-score: It is a weighted average of Precision and Recall.
- Confusion Matrix: A matrix showing the number of data that were classified correctly and incorrectly for both classes, this is used to calculate precision, recall and F1-score.
- Feature Importance: Table/Figure showing relevance of variables of the models, that is the features used to train the model. It gives an insight of the what features are the most relevant and less relevant and accordingly changes can be make to the model or some knowledge about the model is discover.

4.1 Actors

In this section we will look at the actors. The section will be further divided into two sub-sections, how actors become famous and how actors stay famous/unfamous. In the sub-section regarding how actors become famous we will use the dictionaries created for the unique actors using the "Motion-Picture with IMDb score" (full cleaned data) as stated in Table: 3.2. In the sub-section about how actors stay famous/unfamous we will use the dictionaries created for the unique actors using the subsets of the data, that is the "Motion-Pictures with Box-Office values", the "Motion-Pictures with Metacritic score" and the "Motion-Picture with Budget" shown in Table: 3.2.

4.1.1 Prediction how actors become famous

This sub-section will focus on predicting how actors become famous. The full cleaned data set will be used in this case, that is 581848 motion-pictures with around 2.5 million actors. From all the actors dictionaries are created with the actors unique identifiers as keys and a list of all the motion-pictures they played as values. Using the lists of motion-pictures done by the actors three consecutive motion-pictures where the actor is not a 'Featured Actor' are chosen and a combination of both featured (binary feature = 1) and not featured (binary feature = 0) for the 4^{th} motion-picture is picked to train/test the model. Thus the main difference between the section where we try to train/predict how actors stay famous/unfamous will be in the sequences use to train the model. In this sub-section only two sequences will be used, that is the sequences that go from zero to one and zero to zero, 000_1 and 000_0. Table 4.1 shows the distribution of sequences. It was only divided into three groups, since only two of them will be used for the model.

Sequence	**No. of data points**
000_0	5759791
000_1	365421
Other	2342839
Total	**8468051**

Table 4.1: Here the number of data according to the sequence they belong to in agreement with the binary values representing if the actor was featured in the motion-picture or not is shown. The total number of data points is of around 8.5 million and they are distributed in 16 distinct sequences, but only three sequences are shown, the aggregated of the 13 other sequences are in 'Other'. It is possible to observe that the majority of data-points have the sequence '000_0', 5759791 in total. This big difference will generate issues in the training and predictions since it will create a class imbalance. Therefore a random subset of the data points will be chosen with the sequence 000_0 in order to amend the class imbalance.

The total number of sequences used to train and test our model were 731421 with a total of 45 attribute/variables, these variables are shown in Table 4.2. From the 731421, 658278 were used to train and 73143 were used to test the model.

Attributes			
IMDb	writer_score	score_production_co	star_1
actor_score	producer_score	no_reviews_imdb	star_2
cast_score	cinematography_score	no_reviews_users	star_3
director_score	binary_value_actor	no_reviews_critics	

Table 4.2: The attributes shown here are the ones that are used in our RF-Model-0. It is important to highlight that these attributes correspond to one motion-picture and that each individual motion-picture has the same attributes, therefore we show here 15 attributes and the total number of attributes used in the model are $3 \times 15 = 45$. We add the number 1, 2, 3 or 4 preceded by '_' at the end of each attribute in order to differentiate to what motion-picture it corresponds to in the sequence.

Only two models were trained and tested with this data set, this is due to the size of the data and the available computational power. The best model characteristics, RF-Model-0, is shown next.

- **RF-Model-0:** $n\_estimators = 1750$, $max\_features = None$, $criterion = entropy$

The classification report and the top 6 most important features are shown in Table 4.3. The confusion matrix showing the number of correct and incorrect classification for the classes is shown in Figure 4.1. It is possible to see all the importance of each feature in Figure 4.2, the table with the exact values can be found in the appendix Table A.4.

Classification Report RF-Model-0			
Accuracy:			60%
OOB SC:			59.48%
AUC SC:			60%
Class	**Precision**	**Recall**	**F1-SC**
0	0.62	0.54	0.57
1	0.58	0.66	0.62

Feature Importance	
cast_score_3	0.02861
star_1_1	0.02783
actor_score_1	0.02782
star_3_3	0.02776
star_3_1	0.02773
star_1_2	0.02771

Table 4.3: Classification report and top 6 feature importance for RF-Model-0

This model is different from the rest that will be used since it only takes into account two of the sequences, but similar as it takes three motion pictures and tries to predict the binary value of the 4^{th}. The accuracy of this model is not great, but it balances with the Recall and F1-score for class 1. This tells us that the model can predict better when an actor will become famous. Most of the feature importance

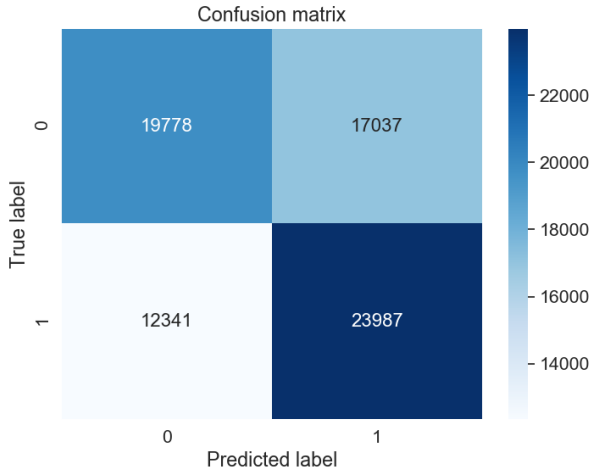


Figure 4.1: Confusion Matrix for RF-Model-0 found with random search

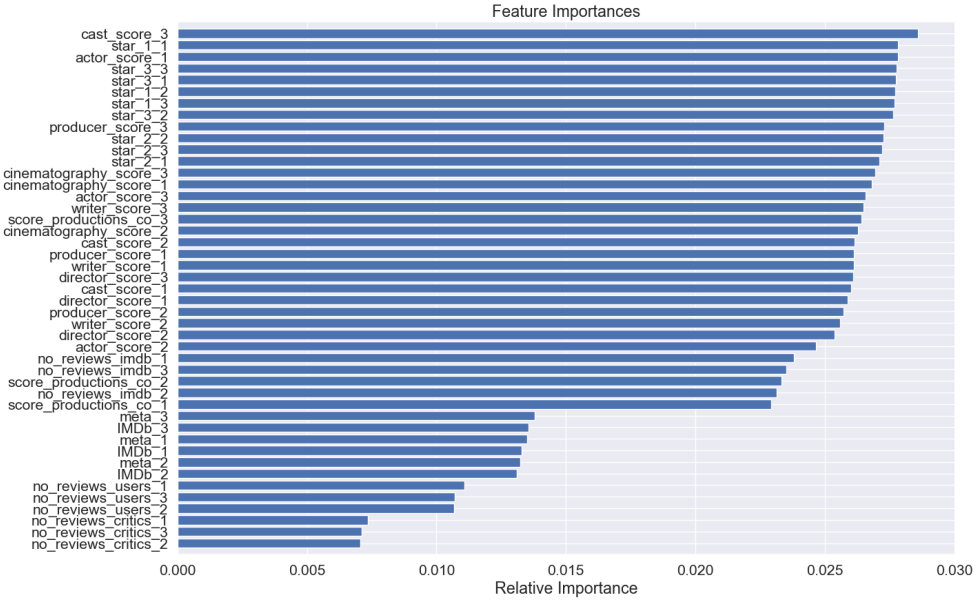


Figure 4.2: Feature Importance for RF-Model-0 found with random search

values have similar scores, but it is interesting to see that amongst the most important features are the 'stars' of the movies, the list of three actors used to create the binary feature. In the bottom of the graph it is possible to see the number of reviews done by critics, which is the least important feature that the model takes into account for all three motion-pictures with an importance of around 0.7%. The same variables for all 3 motion-pictures seem to be grouping together, having a similar impact on the overall model-prediction.

4.1.2 Predicting how stars stay famous/unfamous

In this sub-section we are going to predict how actors stay famous/unfamous. We will be looking at the binary feature showing if the actor is featured in the motion-picture or not, if the actor appears in the list of three stars. We will be training Random Forest models using only sub-sets of the data with complete information regarding all fields (see Section 3). When we say that we are going to predict how an actor stays famous/unfamous we are referring to the combinations of binary features (if actor is featured or not in the motion-picture) of the sequences of motion-pictures we feed our Random Forest model, an example of the different combinations when looking at a sequence of three motion-pictures while training and predicting on the fourth binary feature can be seen in Table: 4.4. Here we trained/tested our model with all the different combinations in comparison to the section regarding 'how to become famous' where we only used two of the combinations, a more explicit explanation of becoming famous can be found at the start of that subsection.

4.1.2.1 Predicting binary-value of actors with sequences of motion-pictures that have budget value

In this sub-section we will look at chronological sequences of variables of motion-pictures done by actors and try to predict the binary feature of the next motion-picture of the actor. We will use the dataset containing 6030 movies and that contain information regarding the budget of the motion-pictures, from these 6030 motion-pictures we obtain 185245 number of unique actors. Using the information of the motion-pictures each actor were a part of we start by creating sequences of three consecutive motion-pictures and getting the binary feature of the 4^{th} thus creating sequences shown in Table 4.4, the number actors that have three or more motion-pictures are 17666. The total number of sequences/data-points we can get are 83120, but as we see in Table 4.4 there is a big class-imbalance. Since we are looking at three motion-pictures and their attributes and using the binary value of the 4^{th} motion-picture to train and later predict of an actor, the values can be either zero or one therefore we can see that there will be $2^4 = 16$ combinations of sequences. In Table 4.4 we show the number of data-points each combination has.

Since there exists a class imbalance in our dataset we will randomly select 3000 data-point sequences from the sequence 000_0, while selecting the rest of the data

Sequence	No. of data points
000_0	54417
001_0	2859
010_0	2813
100_0	3016
011_0	1310
110_0	1461
101_0	1168
111_0	1562
Total	**68606**

Sequence	No. of data points
000_1	2969
001_1	1333
010_1	1182
100_1	1203
011_1	1496
110_1	1390
101_1	1369
111_1	3572
Total	**14514**

Table 4.4: Here we are showing the number of data points according to the sequence they belong according to the binary values representing if the actor was featured in the motion-picture or not. We have a total of 83120 data-points distributed in 16 sequences. The different sequences for the binary values shows if the actors are featured or not on 4 motion-pictures, training and testing on the variables of three motion-pictures and the 4_{th} motion-picture binary value. We can observe the majority of data-points that have the sequence '000_0', 54417 in total, and also the total number of data points that have a zero in the 4_{th} motion picture, 68606, compared to the sequences with 1's in the 4_{th} motion-picture, 14514. This big difference will generate issues in our training and predictions since it will create a class in-balance. Therefore we will select a random subset of the data-points with sequence 000_0 in order to amend the class in-balance.

points from all the other sequences and randomly mix them. We use the under sampling method [21], only selecting 3000 data points because the similar sequence 000_1 has 2969, Table: 4.4. In order to train a Random Forest model with our data we will need to select certain model attributes, two of the main hyperparameters are the number of trees ensemble to create the Random Forest, and the other is the number of features to sample and pass onto each tree. We select the three of the most common values for feature sampling for a Random-Forest classification model, the square root of the total number of features sampled($m = \lfloor\sqrt{p}\rfloor$), the logarithm to the base two of the total number of features sampled($m = \lfloor\log_2 p\rfloor$) and the total number of features sampled($m = p$). In order to see what combination is the best we look at the OOB error rate for a range of tree sizes, 100 to 2500 and the number of features mentioned, figure shows the results Table 4.3. The attributes each motion picture has are shown in Table 4.5, for a description of each attribute refer back to Chapter 3.

We will examine the OOB error in order to see what is the best number of trees (n_estimators) to used in this model and the best number of features (max_features) to consider when looking for the best split. Figure 4.3 shows the OOB error for models with n_estimators ranging from 100 to 2500 trees using three different max_features, the total number of models trained with our data were 7200. We are looking at

Attributes			
IMDb	director_score	score_production_co	star_1
meta	writer_score	no_reviews_imdb	star_2
actor_score	producer_score	no_reviews_users	star_3
cast_score	cinematography_score	no_reviews_critics	
budget	box_office	binary_value_actor	

Table 4.5: The attributes shown here are the ones that we used in our Random Forest model. It is important to highlight that this attributes corresponds to one motion-picture and that each individual motion-picture has the same attributes, therefore we show here 18 attributes and the total number of attributes used in the model are $3 \times 18 = 54$. We add the number 1,2 or 3 precede by '_' at the end of each attribute in order to differentiate to what motion-picture it corresponds in the sequence.

the point where the OOB stabilizes, we can see this happening at around 1250 trees (n_estimators) for 'max_feature = None'(this means selecting all input variables as candidates for splitting), the difference in the OOB error is minimal between the trees at this point, but since it seems that the lowest OOB error is at 'max_features = None' with number of trees ranging between 1500 and 2000 and in most cases more trees are better[22], we will choose 1750 trees for our model. We stated in Chapter 2 that the standard value of variables m is $\lfloor \sqrt{p} \rfloor$, where p is the total number of feature, but we also stated that the value of m would depend on the problem at hand, therefore we will investigate both cases thus the Random Forest models with the following hyperparameters:

- **RF-Model-1**: $n\_estimators = 1750,\ max\_features = None,\ criterion = gini$
- **RF-Model-2**: $n\_estimators = 1750,\ max\_features = sqrt(n\_features)$ $criterion = gini$

The Random forest model has many hyperparameters, the only hyperparameters changed were the ones showing in RF-Model-1 and RF-Model-2, the default values were used in the rest of the hyperparameters. We trained the model with 90% of the dataset that is 28532 data-points each with 54 variable/attributes columns and tested the model with the 10% left, that is 3171 data points also with 54 variables/attributes.

The classification report for our models are shown in Table 4.6 and Table 4.7. The confusion matrix showing the number of true and predicted labels for both classes for the two models are shown in Figure 4.4 and Figure 4.5. The models most relevant and least relevant variables are shown in Figure 4.6 and Figure 4.7, the top seven feature importance values for both models can be seen in Table 4.8 (the table containing all values is in the Appendix in Table A.1).

The results yielded by our models are not the most favorable in terms of accuracy, but even less favorable when we focus on the Recall of out binary class 1. Considering

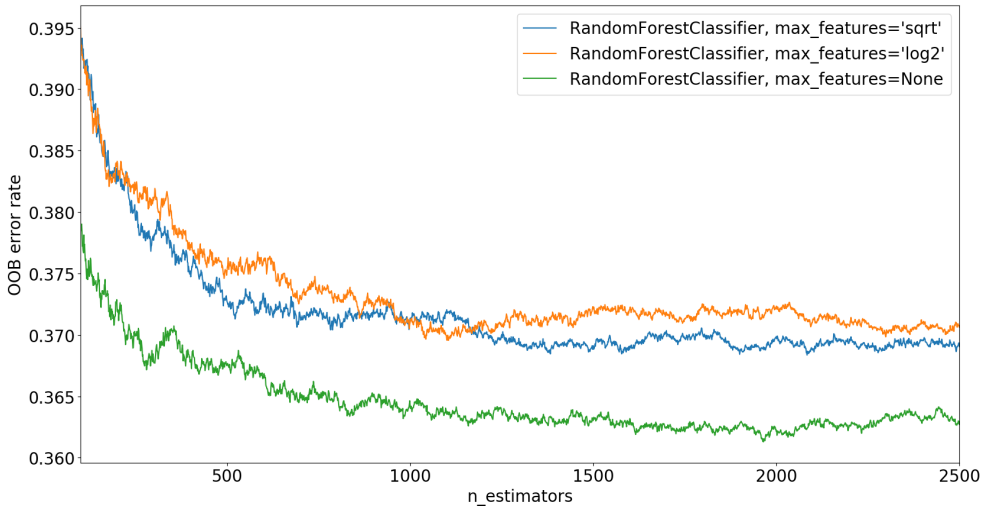


Figure 4.3: The out-of-bag (OOB) error estimate

Classification Report RF-Model-1			
Accuracy :			63.67%
OOB score:			63.61%
AUC score:			67%
Class	**Precision**	**Recall**	**F1-score**
0	0.64	0.76	0.69
1	0.63	0.49	0.55

Table 4.6: The classification report for the RF-Model-1.

Classification Report RF-Model-2			
Accuracy :			62.82%
OOB score:			63.13%
AUC score:			67.5%
Class	**Precision**	**Recall**	**F1-score**
0	0.62	0.81	0.70
1	0.65	0.41	0.50

Table 4.7: The classification report for the RF-Model-2.

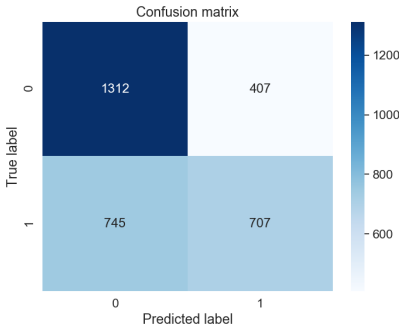


Figure 4.4: Confusion Matrix: RF-Model-1

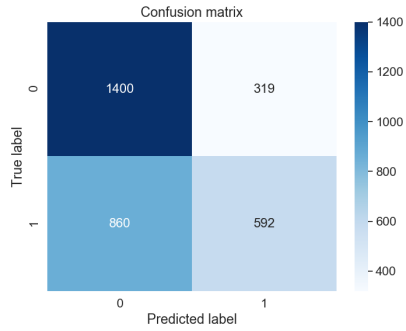


Figure 4.5: Confusion Matrix: RF-Model-2

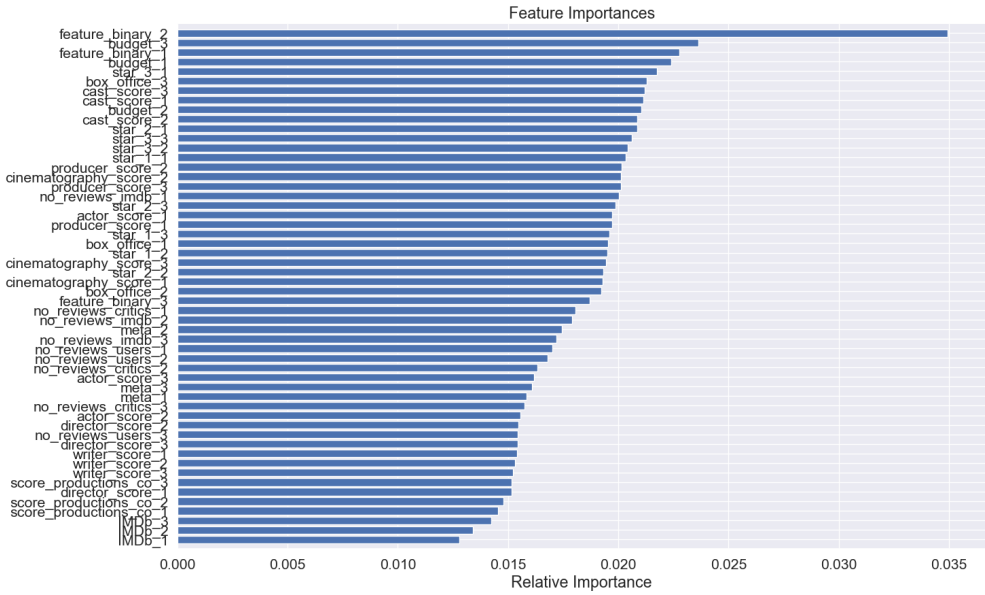


Figure 4.6: Feature Importance for RF-Model-1.

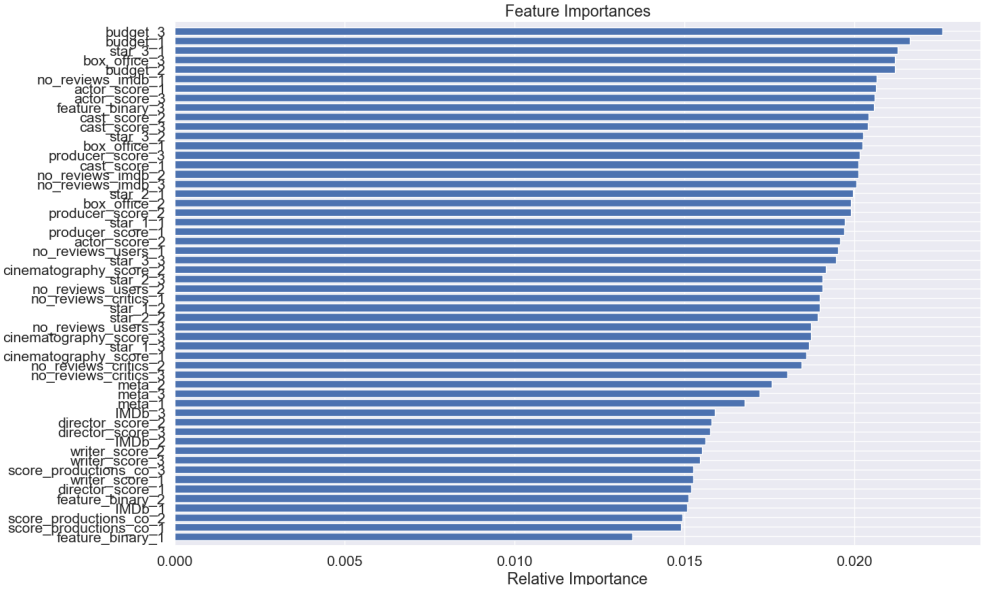


Figure 4.7: Feature Importance for RF-Model-2.

Feature Importance			
RF-Model-1		**RF-Model-2**	
Feature	**Value**	**Feature**	**Value**
feature_binary_2	0.034928	budget_3	0.022584
budget_3	0.023643	budget_1	0.021633
feature_binary_1	0.022763	star_3_1	0.021282
budget_1	0.022404	box_office_3	0.021199
star_3_1	0.021764	budget_2	0.021189
box_office_3	0.021281	no_reviews_imdb_1	0.020648
cast_score_3	0.021184	actor_score_1	0.020631

Table 4.8: Top seven Feature Importance values for **RF-Model-1** and **RF-Model-2** shown in Figures 4.6 and Figures 4.7, table with full values can be found in the Appendix, Table A.1.

this the RF-Model-1 shows better results than RF-Model-2 since the Recall for the class predicting 1's is higher, the Accuracy is also slightly superior, but not significant. We can confirm that in both models the OOB score is slightly higher than the Accuracy, but almost the same, thus confirming the theory behind OOB and that it can be used as a test-set of the model and that it functions as a N-fold cross-validation.
The Feature importance for RF-Model-1 seems fairly intuitive in the main features, we would expect to see most of motion-picture 3 features higher, but this is not the case. We can observe that the 'feature_binary' of motion picture 1 and 2 are on the top and the 'feature_binary' for motion picture 3 is not that far. What is interesting to notice is that they all have a similar score overall and none of them have a score higher than 3.5%. We also notice the trade off between Recall and Precision when comparing the 2 models.
Since Accuracy does not seem the be the best way to asses our model we decided to use AUC - ROC scores to have a better understanding, since it measures the performance for the classification problem at different threshold settings. Receiver operating characteristic 'ROC' is a probability curve and Area Under The Curve 'AUC' shows the degree or measure of separability. It exhibits the capability of the model to distinguish between the classes, therefore the higher the AUC the better the model at predicting the zeros and ones in our model, the closer the AUC to 1 the better. In Figures 4.8 and 4.9 we show the ROC plot for RF-Model-1 since it is the model that yielded the best results, the average $AUC = 0.67$.

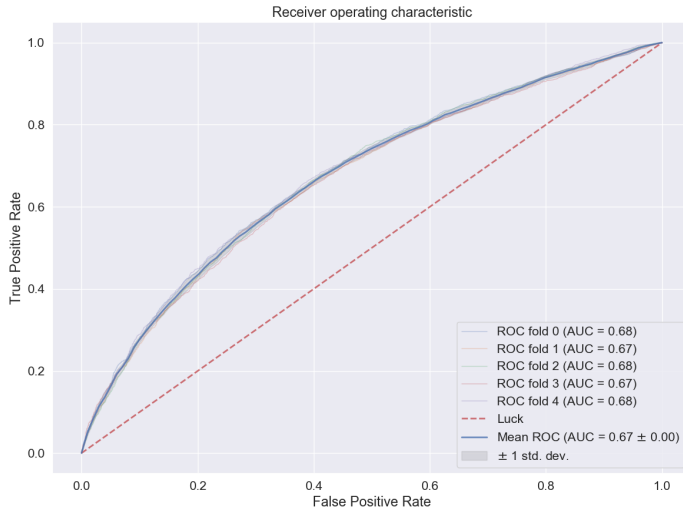


Figure 4.8: Receiver operating characteristic plot for **RF-Model-1**. The True Positive Rate, 'TRP' is plotted against the False Positive Rate, 'FPR'. The AUC values are calculated for 5 different ROC folds of the data and the 50% luck base-line is set in red. The closer to the value of AUC to 1 the better.

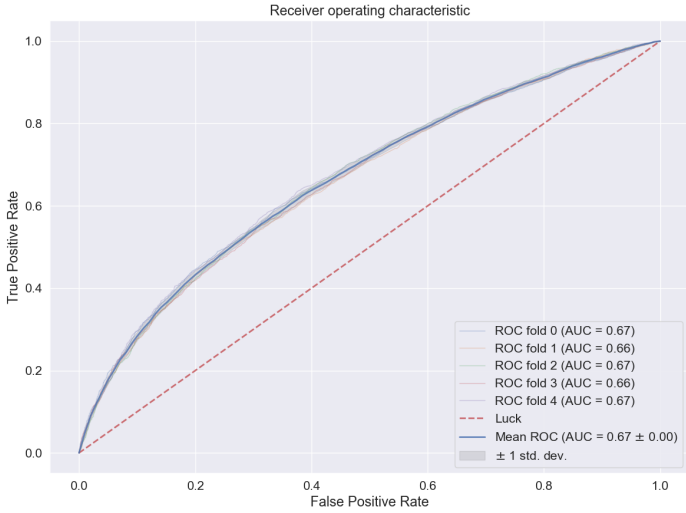


Figure 4.9: Receiver operating characteristic plot for **RF-Model-2**. The True Positive Rate, 'TRP' is plotted against the False Positive Rate, 'FPR'. The AUC values are calculated for 5 different ROC folds of the data and the 50% base-line is set in red 'luck'. The closer to the value of AUC to 1 the better.

So far we have seen two models with mostly default hyperparameters, normally these default values perform well with large enough number of trees, but we will see if we could improve this by trying to tune some of the hyperparameters [23].
We will observe if changing some of the hyperparameters could improve the model and increase the Accuracy, Recall and maybe show a more pronounce difference in the importance of the variables. The first method we had in mind to accomplish our hyperparameter tuning for our model 'GridSearch'[19] in python with different attribute combinations, this creates models with all possible combinations one decides to try and according to a scoring parameter set beforehand (e.g., Accuracy, Recall, AU) and it returns the a model with the best attributes, this method is time consuming as well as computationally expensive therefore we decided against it and instead we will try random search, "randomly chosen trials are more efficient for hyper-parameter optimization than trials on a grid" [24].

The following are the hyperparameter which were chosen to tune to try to improve the model.

- 'max_depth'
- 'min_samples_leaf'
- 'min_samples_split'
- 'criterion' ('gini', 'entropy').

Some of these values were chosen at random, by increasing the default values, but for two of them we decided to test them first individually, these 2 attributes are 'max_depth' and 'min_sample_split' and the results for different values are shown in Table 4.9. In this table we can see that increasing 'min_sample_split' does not help the model and the accuracy seems to fluctuate, in the other hand 'max_depth' seem to have an slight improvement as it increases.

m_s_s	Accuracy	Recall
10	64.142967	0.418026
15	64.104535	0.413734
20	63.950807	0.412876
25	64.142967	0.411159
100	63.989239	0.399142
200	64.027671	0.395709
400	63.835511	0.392275

max_depth	Accuracy	Recall
1	56.802460	0.528547
3	61.683321	0.427336
5	62.413528	0.389273
7	63.182168	0.401384
9	63.797079	0.413495
11	63.758647	0.415225
14	63.797079	0.425606

Table 4.9: Here we try different values for min_sample_split, 'm_s_s', of our RF-Model-1 and the max_depth for RF-Model-1 in order to have a better idea of what values to chose when using GridSearch.

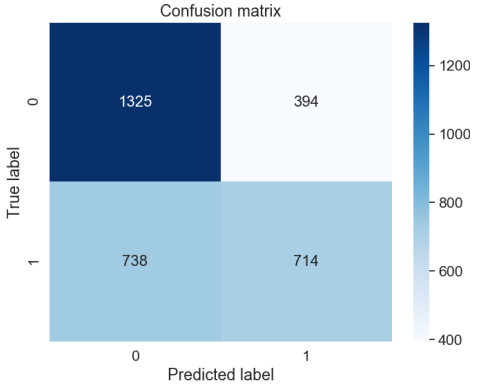
Many models were tested with different values for the hyperparameters, many of the the results were similar to the first models we trained. The best model we found using random search is:

- **RF-Model-3:** $n\_estimators = 1250$, $max\_features = None$, $criterion = entropy$, $min\_samples\_leaf = 16$, $max\_depth = 24$, $min\_samples\_split = 10$

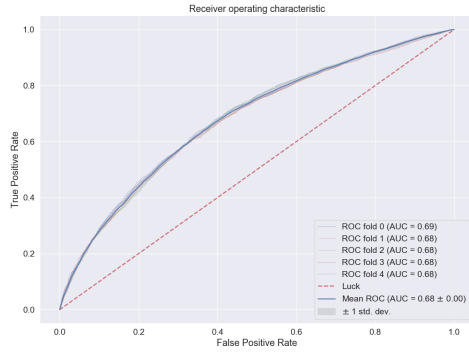
The classification report for RF-Model-3 is shown in Table 4.10, the confusion matrix and ROC-AUC plot are shown in Figure 4.10. Finally the Feature Importance graph is shown in Figure 4.11. We also included the top seven feature importance in a table with the corresponding values, this can be seen in Table 4.11, and the full table can be found in the Appendix, in Table A.2.

Classification Report RF-Model-3			
Accuracy :			64.30%
OOB score:			63.86%
AUC score:			68%
Class	**Precision**	**Recall**	**F1-score**
0	0.64	0.77	0.70
1	0.64	0.49	0.56

Table 4.10: The classification report for the RF-Model-3.



(a) Confusion Matrix for RF-Model-3.



(b) Receiver operating characteristic plot for RF-Model-3.

Figure 4.10: Confusion Matrix for Rf-Model-3(left) and Receiver operating characteristic, ROC-AUC(right) plots for Rf-Model-3 found with Random Search.

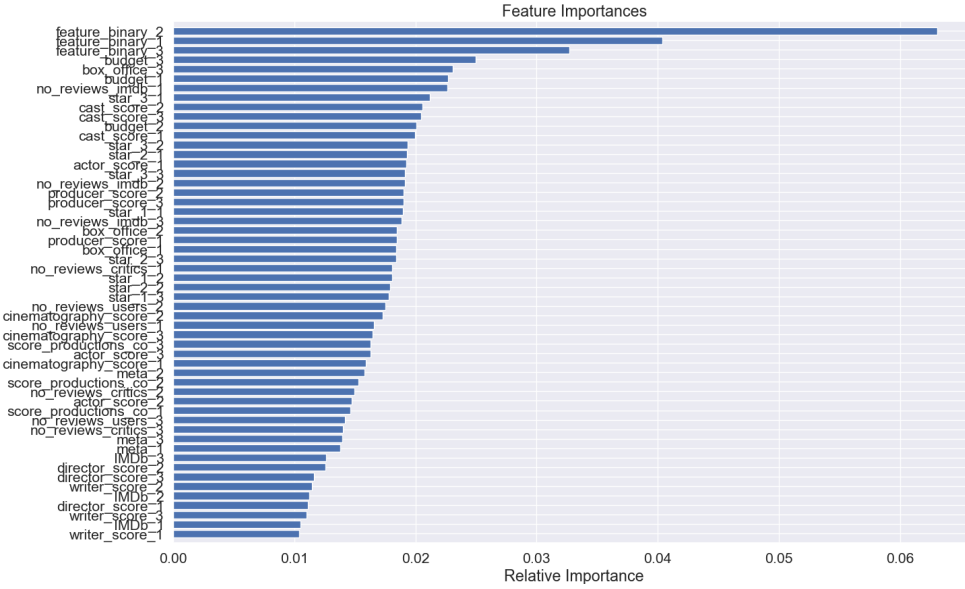


Figure 4.11: Feature Importance for Rf-Model-3 found with random search.

Feature Importance	
RF-Model-3	
Feature	**Value**
feature_binary_2	0.063047
feature_binary_1	0.040366
feature_binary_3	0.032714
budget_3	0.024993
box_office_3	0.023062
budget_1	0.022701
no_reviews_imdb_1	0.022617

Table 4.11: Feature Importance values for RF-Model-3.

The extensive computational and time consuming search for a model that performs better did not yield the desirable outcome, but the new model seems to be performing slightly better in terms of Accuracy and the Recall for both classes. It is important to mention that this new model used '$n\_estimators = 1000$', thus it takes less time to train, it also uses 'entropy' as criterion instead of 'gini'. The Feature importance shown in Figure 4.11 seems more intuitive with the features one will imagine seen first since it uses 'feature_binary' of the last 3 motion-pictures, though 'feature_binary_2' is the highest with 6.3%. These low numbers and the somewhat evenly distributes values that the other features have might be due to some correlation between the features, but is good to see that the model not only depends on the 'feature_binary' of the movies and that the other features play a role in it, especially 'budget' and 'box-office' of the third motion-picture.

We have seen different models with a subset of the sequence '000_0', we want to also show what would happen if the full dataset is train (74808 data-points) and tested (8312 data-points) and show the class-imbalance issue. We named the model as **RF-Model-4** and added most of the results in the Appendix, here we will only show the classification report and the top-5 most important features, both table can be seen in Table 4.12. In the appendix is the full table of feature importance, Table A.2, also the Feature Importance Figure A.1 and the Confusion Matrix Figure A.2.

Classification Report RF-Model-4			
Accuracy:			85.54%
OOB sc:			85.88%
AUC sc:			68.12%
Class	**Precision**	**Recall**	**F1-score**
0	0.89	0.95	0.92
1	0.61	0.42	0.50

Feature Importance	
feature_binary_3	0.2884
feature_binary_2	0.1037
feature_binary_1	0.0784
budget_1	0.0146
budget_3	0.0142
budget_2	0.0131

Table 4.12: Classification report and Top 6 Feature Importance for model using all the dataset that contains 'budget', Rf-Model-4.

We can see that 'feature_binary_3', 'feature_binary_2' and 'feature_binary_1' are the predominant features, this makes sense since around 63% of the data have the sequence '000_0' therefore the Random-Forest models has an inclination to use this to predict, and predicts that most of the binary values will be zero. We can observe that in the Recall for class 1 it under-performs comparing to RF-Model-3, thus showing that considering the class imbalance and using a subset of the sequence with the most data was a reasonable choice.

Now that we have looked at the subset of the full dataset that contains the budget information, we will continue and look at a bigger dataset containing more data-points with the gross/box-office of the motion-pictures, but without the budget information of the motion-pictures.

4.1.2.2 Predicting binary value of actors with motion-pictures that have Gross/Box Office

In the last sub-section we created a model that takes6032 motion-pictures and extracted information of the sequences of four motion-pictures of every actor. The next step is creating a similar model, but this time having a bigger number of motion-pictures as dataset and one less feature, that is budget.
Similarly to what has been done in that last subsections we will also look at the motion-pictures in chronological order sequence and train a model using information of consecutive motion-pictures of an actor and try to predict the binary feature of the next motion-picture, but since we have more data-points in this section we will see if adding more movies to the sequence improves the model. The attributes/variables of each motion-picture we are gonna use in our model for each motion-picture are shown in Table 4.13.
We intended to use the subset of the dataset that contained metacritic information, 8111 motion-pictures, but since this is a subset of the 12527 motion-pictures that have 'gross/box-office' and since 'metacritic' and 'IMDb-score' are somewhat related in many motion-pictures then we use the 'IMDb-score' for those motion-pictures that were missing 'Metacritic' and change in the correct format of hundreds instead of decimal, therefore obtaining more data-points for our models. The total number of actors in the subset of 12527 motion-pictures is 307143.
This subsection will be divided into another two subsections in order to differentiate the two models we will train and test. The first model will be trained and tested using the attributes of three consecutive motion-pictures and the binary feature of the 4^{th} similarly to the last subsection. The second model will be trained and tested using the attributes of four consecutive motion-pictures and the binary feature of the 5^{th} motion picture.

Model using features of 3 consecutive motion-pictures and predicting the 4^{th}.

The first model we will train and test will be the one using the attributes of three

Attributes			
IMDb	director_score	score_production_co	star_1
meta	writer_score	no_reviews_imdb	star_2
actor_score	producer_score	no_reviews_users	star_3
cast_score	cinematography_score	no_reviews_critics	
box_office	binary_value_actor		

Table 4.13: The attributes shown here are the ones that we used in our Random Forest model. It is important to highlight that these attributes corresponds to one motion-picture and that each individual motion-picture has the same attributes, therefore we show here 17 attributes and the total number of attributes used in the model are $3 \times 17 = 51$, $4 \times 17 = 68$. We add the number 1, 2, 3, 4 or 5 preceded by '_' at the end of each attribute in order to differentiate to what motion-picture it corresponds in the sequence.

consecutive motion pictures and predicting the 4^{th}. This is almost identical to the last subsection where we had budget information about the motion-pictures, but here we will have more data-points to train and test on. Out of the 307143 actors in the 12527 motion pictures only 28247 of them have four or more motion pictures. The total number of sequences extracted from the 28247 actors with 4 consecutive motion pictures is 145057, we can see the sequences distribution in Table 4.14.
We also have the issue of class-imbalance here where the sequence of binary feature '000_0' is extremely large when comparing to the other sequences, therefore a subset of this sequence is used in the model in order to avoid a high accuracy that is mostly based on people not been featured. We will used the number of data-points in the sequence '000_1' as a baseline for picking the subset of data-points from sequence '000_0', thus we use the under sampling method [21] again and choose only 6000 data-points of this sequence at random.

We stated before that the most efficient way to find the best model is to randomly choose trails [24], therefore we will also do the same here and use the hyperparameters of the best model of the last section as a base in order to find the hyperparameters that yield the best results for the dataset in this section. After training many models with different combinations of hyperparameters we found the following,

- **RF-Model-5:** $n\_estimators = 1750$, $max\_features = None$, $criterion = gini$

The classification report can be seen in Table: 4.15 alongside the top six feature importance table.

Again the results are not the most favorable, we can see that **RF-Model-3** outperforms this model in terms of Recall for class 1, AUC score and Accuracy. This shows that the variable 'budget' of each motion-picture helps the model predict better and that adding more data-points not necessarily helps improving the model. To

Sequence	No. of data points
000_0	89175
001_0	5773
010_0	5665
100_0	5862
011_0	2542
110_0	2677
101_0	2461
111_0	2960
Total	**117115**

Sequence	No. of data points
000_1	6196
001_1	2659
010_1	2445
100_1	2314
011_1	2813
110_1	2733
101_1	2653
111_1	6129
Total	**27942**

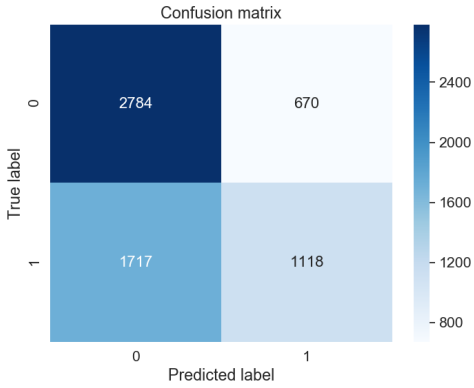
Table 4.14: Here we are showing the number of data points according to the sequence they belong according to the binary values representing if the actor was featured in the motion-picture or not. We have a total of 83120 data-points distributed in 16 sequences. The different sequences for the binary values shows if the actors are featured or not on 4 motion-pictures, training/testing on the variables of three motion-pictures and the 4_{th} motion-picture binary value. We can observe the majority of data-points that have the sequence '000_0', 54417 in total, and also the total number of data points that have a zero in the 4_{th} motion picture, 68606, compared to the sequences with 1's in the 4_{th} motion-picture, 14514. This big difference will generate issues in our training and predictions since it will create a class imbalance. Therefore we will select a random subset of the data-points with sequence '000_0' in order to amend the class in-balance.

Classification Report RF-Model-5			
Accuracy:			62.05%
OOB SC:			62.69%
AUC SC:			60.02%
Class	**Precision**	**Recall**	**F1-SC**
0	0.62	0.81	0.70
1	0.63	0.39	0.48

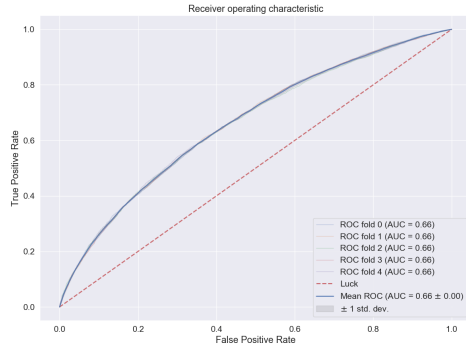
Feature Importance	
feature_binary_2	0.025692
box_office_3	0.024475
star_3_1	0.023398
cast_score_2	0.023236
star_3_2	0.023077
box_office_2	0.023052

Table 4.15: Classification report and Top 6 Feature Importance for model Rf-Model-5.

finish this subsection we also added the Confusion matrix and the Receiver operating characteristic plot in Figure 4.12. Also we can see the Feature Importance graph in Figure 4.13. The feature importance values are not that different, the values are close, though 'feature_binary' of the second motion-picture and 'box_office' of the third movie are the top two. We try models where we take the last attributes from the list, but this did not help improving the model, on the contrary it made it worse, similarly to this model compared to the model where we used budget.



(a) Confusion Matrix for RF-Model-5.



(b) Receiver operating characteristic plot for RF-Model-5.

Figure 4.12: Confusion Matrix for Rf-Model-5(left) and Receiver operating characteristic, ROC-AUC(right) plots for Rf-Model-5 found with Random Search.

Now that we have seen a model where we have a bigger data-set but one less feature we will see if adding more motion-pictures to the sequence could improve our predictions.

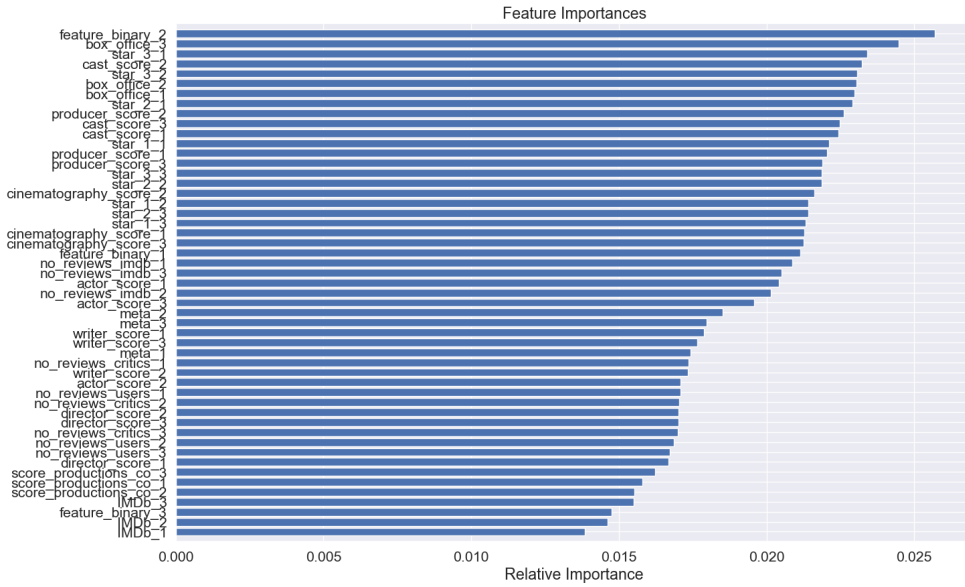


Figure 4.13: Feature Importance for Rf-Model-5 found with random search

Model using features of 4 consecutive motion-pictures and predicting the 5^{th}.

We have trained and tested our Random Forest Model with attributes of three motion-picture sequences with the binary values of the 4^{th}. We have not seen great results and we can also notice the decrease in performance from the model where we included the budget attribute of the motion-pictures, considering this the next step in the search of improving our model is increasing the sequence number of motion-pictures, thus instead of using the features of three motion-pictures use the features of four motion-pictures while trying to predict the 5^{th}. Since we are increasing the sequence of motion pictures then we will have less actors that were protagonist in five or motion pictures motion-pictures therefore having a smaller data-set to train and test on. The number of actors with 5 or more motion pictures are 19884 and the total number of sequences we obtained from these actors are 116810. As stated before we also have a class imbalance here, but the distinct sequences are $2^5 = 32$. We have seen before that class 0 is the class with the biggest number of data-points and that the sequence with all zeros is the biggest one, taking this into account instead of showing 32 distinct sequences we will only show 5, this can be seen in Table 4.16.

We found the Random Forest model with the best hyperparameters using random search, as we did in the last section, thus the model is the following,

- **RF-Model-6:** $n\_estimators = 1750,\ max\_features = None,$ $criterion = entropy$

Sequence	No. of data points
0000_0	64122
0000_1	3929
1111_1	4234
1111_0	1610
Other	42915
Total	**116810**

Table 4.16: Here we are showing the number of data according to the sequence they belong taking into account the binary values representing if the actor was featured in the motion-picture or not. We have a total of 116810 data-points that should be distributed in 32 distinct sequences, but we only show the number of 4 sequences and the aggregated of the 28 other sequences in 'Other'. We can observe the majority of data-points that have the sequence '0000_0', 64122 in total. This big difference will generate issues in our training and predictions since it will create a class in-balance. Therefore we will select a random subset of the data-points with sequence 0000_0 in order to amend the class imbalance.

The classification report for RF-Model-6 is shown in Table: 4.17, the table also includes the six most important features of the model. The full table is attached in the appendix A.3.

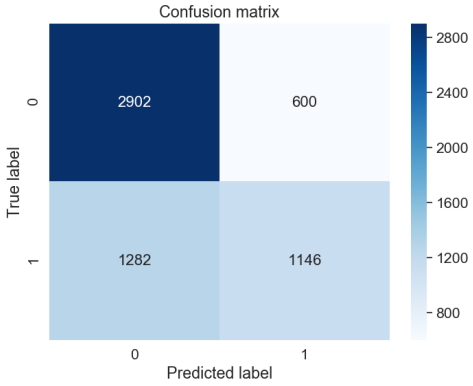
Classification Report RF-Model-6			
Accuracy:			68.26%
OOB SC:			67.57%
AUC SC:			72%
Class	**Precision**	**Recall**	**F1-SC**
0	0.69	0.83	0.76
1	0.66	0.47	0.55

Feature Importance	
box_office_4	0.020748
star_4_1	0.018627
cast_score_1	0.018261
cast_score_4	0.018207
box_office_3	0.018203
cast_score_3	0.018166

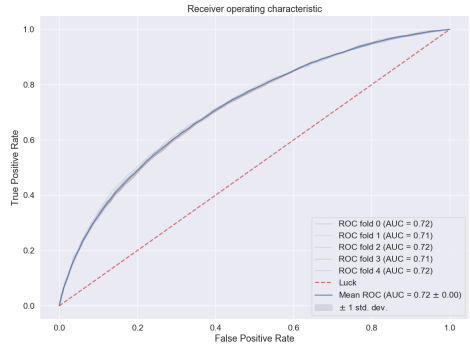
Table 4.17: Classification report and top 6 feature importance for Rf-Model-6.

We can see the improvement from RF-Model-5, 4.15, and also a slight overall improvement from RF-Model-3, 4.10. RF-Model-6 is the model with the highest Accuracy and Recall combination thus far. The confusion matrix plot and the receiver operating characteristic plot for RF-Model-6 can be seen in Figure 4.14 and the feature importance plot with the highest 34 features can be seen in Figure 4.15.

So far it has been shown models training and predicting on actors sequences of motion-pictures while trying to predict if the actor will be featured on the next motion-picture. The next step is using the motion-picture and training/predicting the gross income it will generate, this is done in the next section.



(a) Confusion Matrix for RF-Model-6.



(b) Receiver operating characteristic plot for RF-Model-6.

Figure 4.14: Confusion Matrix for Rf-Model-6(left) and Receiver operating characteristic, ROC-AUC(right) plots for Rf-Model-6 found with Random Search and 5-fold cross validation.

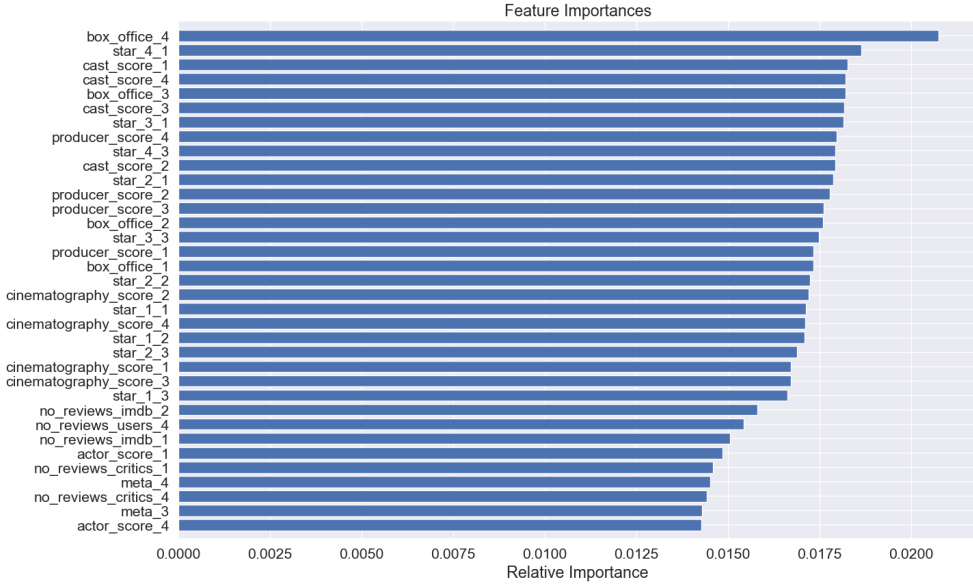


Figure 4.15: Top 34 Feature Importance for Rf-Model-6 found with random search.

4.2 Motion-Pictures

In this section we will train a Random Forest model with the motion-pictures that have box-office and budget information. The goal is to predict the Gross/Box-Office of the motion pictures. It was stated in Chapter 1 1 an approach towards predicting success of box-office of motion-pictures, but in that case Mestyán et. al [3] used the activity of a motion-picture Wikipedia page in order to predict the the success of the motion-picture prior to its release. In our case we will used the attributes of the motion pictures shown in Table 4.18. In order to predict the Gross/Box-Office of the motion pictures and since there will be as many classes as there are data points we will divide the box-office values into terciles, meaning that the box-office information of all motion pictures will be separated into three classes. By dividing the Box-Office values into terciles three classes will be created and the model will try to predict to what class each motion-picture belongs to according to the variables available. In total we have 6030 movies that have both box-office and budget values. A portion of them, 90% will be used to train the model and 10% will be used as the test set.

Attributes			
IMDb	producer_score	score_production_co	star_1
cast_score	cinematography_score	no_reviews_imdb	star_2
director_score	budget	no_reviews_users	star_3
writer_score	binary_value_actor	no_reviews_critics	

Table 4.18: The attributes shown are the ones that to be used in our RF-Model-7. It is important to highlight that these attributes corresponds to one motion-picture and that each individual motion-picture has the same attributes, the total number of attributes per motion-picture is 15.

To find the best model a random search was applied, as done in previous sections, and the best model found is one using the following hyperparameters with default values on the other hyperparameters,

- **RF-Model-7:** $n\_estimators = 1250$, $max\_features = None$, $criterion = entropy$

Since we have divided the box-office values into terciles, three different classes, then the best metrics for the model performance are the confusion matrix showing the number of correct classification for each class, the variable importance, and the classification report. The confusion matrix for RF-Model-7 can be seen in Figure 4.16, the variable importance graph is shown in Figure 4.17 also the table showing the specific values for each feature is shown in Table 4.20 and finally the classification report can be seen in Table 4.19.

The model predicts fairly well for the three classes, not perfectly, but much higher than just choosing the class by chance. The 2 extremes, class 0 and class 2 are

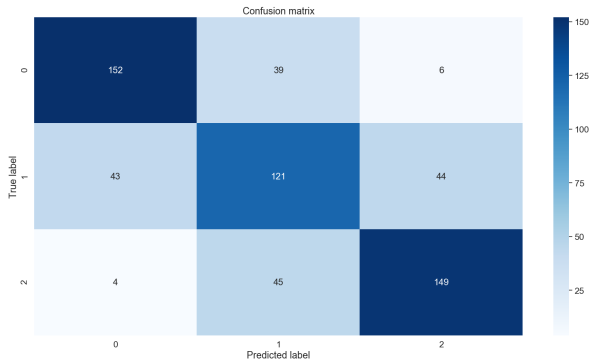


Figure 4.16: Confusion Matrix for RF-Model-7 found with random search.

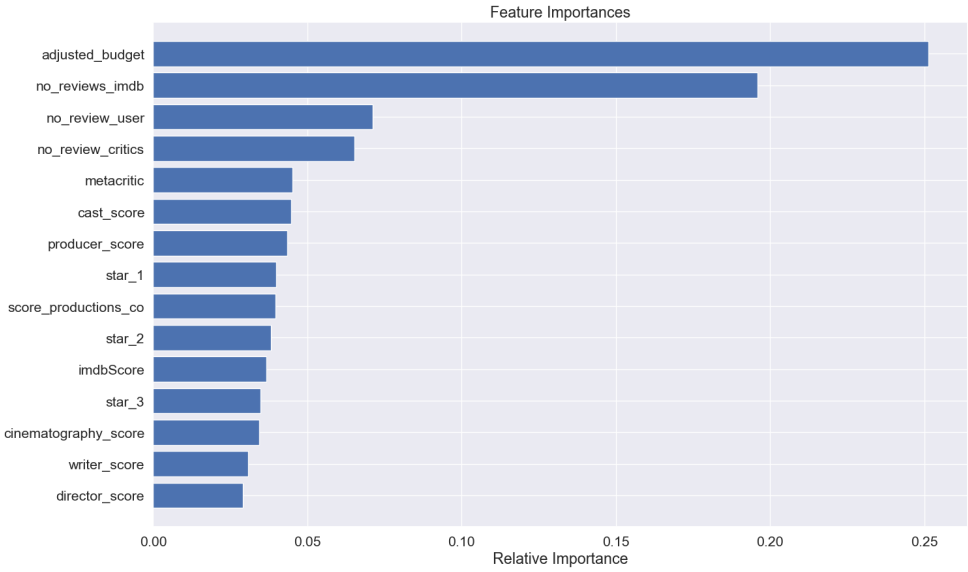


Figure 4.17: Feature Importance graph for RF-Model-7 found with random search.

Classification Report RF-Model-7			
Accuracy:			70%
OOB score:			70.8%
Class	**Precision**	**Recall**	**F1-score**
0	0.76	0.77	0.77
1	0.59	0.58	0.59
2	0.75	0.75	0.75

Table 4.19: Classification report for RF-Model-7 using all the data set that contains 'budget'.

RF-Model-7	
Feature	**Importance**
adjusted_budget	0.2514
no_reviews_imdb	0.1958
no_review_user	0.0711
no_review_critics	0.0652
metacritic	0.0451
cast_score	0.0446
producer_score	0.0435
star_1	0.0398
score_productions_co	0.0397
star_2	0.03812
imdbScore	0.03662
star_3	0.0348
cinematography_score	0.0344
writer_score	0.0307
director_score	0.0290

Table 4.20: The feature importance for RF-Model-7.

predicted better than class 1, the model seems to have issues in this class prediction, this is expected since it is in the middle of the other two classes and the model finds it harder to classify it correctly. This is most likely an issue that arises when the data is separated into terciles, since the total number of data points is divided evenly among the three classes, and the model finds it hard to predict the points that fall to closely to the other classes.

The variable with the highest importance is 'budget', this seems intuitive since movies that invest the most would have the most to gain compared to motion-pictures that have a lower investment. All three variables involving the number of reviews are also higher since if a movie is popular it also means it will have a higher gross. This leads to some issues that needs addressing. People need to see the motion-pictures in

order to review it, therefore these features will not be available for a motion-picture in the making. If the objective is to predict the box-office before the movie is released then another model needs to be trained without the mentioned features. The director, writer and cinematographer score are the variable with the lowest importance, though they are still contributing to the overall prediction of the variables. It is interesting and intuitive to see that the first featured actor is the most important, then the second and finally the third, nevertheless they all have similar importance.

4.3 Discussion

There were countless models trained and tested during the period of this project, but only 8 models are shown here. These models were the ones that gave the best results or showed something that helped understand the prediction of a featured actor or success of the motion-pictured. A summary of the classification report for all the models can be found in Table 4.21.

The results section was divided into two main sections. The actors and the prediction of feature actors were considered in the first part. The movies and the success of the motion-picture according to the box-office was considered in the second part.

The first part was further divided into two subsections. In this first subsection a model was trained and tested, RF-Model-0, to predict and understand when an actor would become featured after having not been featured in three motion-pictures with a combination of both been featured and not been featured in the fourth motion-picture. The Accuracy of this model was the lowest of all the models shown, but the recall for class 1 was the highest at 66% and F1 score of 62% for class 1. It is difficult to compare this model with the others since it uses a very particular data to train and test.

The models RF-Model-1 to RF-Model-6 are easier to compared since for most of them the same data is used.

RF-Model-1, RF-Model-2 and RF-Model-3 are using the exact same data to train and test, the only difference is in the hyperparameters used. These models also have an extra attribute, that is budget. RF-Model-3 is the one that gives the best results in all terms. This model was found using random search and required training certain number of models to find the best hyperparameters. The time and computational power used to find the best hyperparameters outbalanced the metrics gain, therefore it does not seem to be of great help in this case. The most important features for RF-Model-3 were the ones showing if the actor was featured or not on the last three films, the other features had similar importance and were between 1% and 3%. For this model it is important if the actor was featured in the past or not, we can see the different distributions of featured and not featured actors for the four motion pictures used in the model in Table 4.4. The sequences with most data points are for the ones that actors are never featured and the ones that are always featured. The overall performance of this model is not bad, but it still lack the capability of predicting

class 1 correctly, it only did for 49% of the test data points. The F1 score is still better than RF-Model-1 and RF-Model-2 with 56%.

For most of the models balanced combinations of sequences of the actors which have been featured in previous movies was used in order to avoid class imbalance, but in the case of RF-Model-4 all sequences were used. Using all the data of sequences made the model computationally expensive to train and test. The model performance was not great, but it fell under similar performance than RF-Model-1 and RF-Model-2. The accuracy for this model was the highest among all the models, but this metric is misleading since most of the data are of Class 0, we can see that the recall score for class 1 was 42% and the F1 score for the same class was 50%. This shows that the model does not perform very well when predicting class 1 observations. The feature importance shows that the model considers almost 30% if the actor was featured or not in the third movie of the sequence of the data, around 11% if the actor was feature in the second to last movie of the data sequence and around 8 if the actor was featured in the first movie of the sequence used. With the budgets of the 3 motion pictures as the next feature with most importance all with similar scores of around 1.4%. This clearly shows that using a subset of the sequences with the greatest number of data was a good choice.

The models RF-Model-5 was similar to RF-Model-2, RF-Model-3 RF-Model-4 but with more actors, thus more sequences. The same methodology was used as the previous 3 models, but the consequences of having a bigger data set was not having the budget feature of the movies. This made RF-Model-5 perform poorly and showed that budget was an important predictor variable, hence showing that having more data points with less important features makes the model perform worst.

To offset the poor performance of RF-Model-5 model more movies were added into the sequence, that is four movies instead of three and predicting if the actor is featured in the fifth motion-picture instead of the fourth. This made the total number of data points decrease but increased the number of predictor variables, this model is RF-Model-6. The results of this model were among the best, but not as great as RF-Model-3. This showed that by adding more movies into the sequence to predict if the actor will be feature increases the score of the classification metrics of the model.

Finally a model where the success of movies is considered instead of predicting actor success. The success of a movie in this case is defined as the box-office amount the movie generated. the box-office amounts were divided into terciles, 3 groups, with class 2 been the highest and class 0 the lowest. Model RF-Model-7 was trained and tested on thousands of variables of unique motion-pictures with the goal of predicting success of movies according to the box-office. This model performed well, the accuracy of the model is 70% while the lowest F1 score was 59% for class 1 and between $75-77\%$ for the other 2 classes. These classification results are promising, but there could be a potential problem with the features of importance, those are the budget and the reviews received. The reviews are connected to the number of people that watch the movies, thus the model will be less accurate with movies that are not released yet, nor with movies that were released recently.

In conclusion it is possible to say that the performance of the models to pre-

dict if an actor will be a featured actor in his/hers next role using Random-Forest classification techniques are ordinary when considering the metrics of Accuracy, Precision, Recall, F1-score as a base. Regardless of the results of the models this thesis has broaden the knowledge and understanding of actors success and success of motion-pictures. It also expanded our understanding of Random Forest classification techniques and hyperparameter tuning for models.

RF-M	Acc	OOB	AUC		Precision	Recall	F1
0	60%	59.48%	60%	**Class 0**	0.62	0.54	0.57
				Class 1	0.58	0.66	0.62
1	63.67%	63.61%	67%	**Class 0**	0.64	0.76	0.69
				Class 1	0.63	0.49	0.55
2	62.82%	63.13%	67.5%	**Class 0**	0.62	0.81	0.70
				Class 1	0.65	0.41	0.50
3	64.30%	63.86%	68%	**Class 0**	0.64	0.77	0.70
				Class 1	0.64	0.49	0.56
4	85.54%	85.88%	68.12	**Class 0**	0.89	0.95	0.92
				Class 1	0.61	0.42	0.50
5	62.05%	62.69%	60.02%	**Class 0**	0.62	0.81	0.70
				Class 1	0.63	0.39	0.48
6	68.26%	67.57%	72%	**Class 0**	0.69	0.83	0.76
				Class 1	0.66	0.47	0.55
7	70%	70.8%		**Class 0**	0.76	0.77	0.77
				Class 1	0.59	0.58	0.59
				Class 2	0.75	0.75	0.75

Table 4.21: Overview of the classification report for all the models shown in the project. Acc is the Accuracy and RF-M are the RF-Models.

CHAPTER 5

Future work

The random forest models along side the data exploratory analysis provided valuable insight regarding the problem of predicting success of actors and movies. The results obtained by the models were not the most satisfactory, but a great deal of knowledge and experienced was gained with it.

Considering the results, the data and the knowledge gained there are a number of possible changes and ideas that could be implemented to improve the original goal and prediction of feature actors.

One of this improvements could be the creating of a program similar to consumer price index, CPI [18], in order to take into account other currencies and be able to adjust this currencies for inflation, moreover be able to translate the adjusted currencies into a single currency. The Table 3.8 showed the adjusted budget for movies that had USD as currency, but there were many motion-pictures that contained budget feature but were in currencies not supported by CPI. This approached would make our original dataset bigger with one of the most important features seen in the models.

In the data exploratory analysis chapter we seen that there were a few incorrect countries/region mentioned in the country field of the text files, Antarctica. Thinking in terms of incorrect data, the possibility that Box-Office values were incorrect is there. Therefore an idea of finding other sources such as The Numbers [25], to get the missing or corrected values for box-office along side other information pertaining the motion-picture.

Another possible approach using the current models or other models is to use the cast list of the motion-pictures that are not in alphabetical order and start with the most famous actor of the on the top of the list to the least famous actor on the bottom and quantify this in order to have more precise predictions. This could lead to understanding the actors career, the peak moments and maybe predict the loss of success in actors careers as they move away from the first places and what possible reasons can the data and model can provide.

The hitherto mentioned approaches only consider the data. There are also other machine learning techniques that could be used. A more sophisticated approach using Deep Learning is Long Short Term Memory or LSTM [26]. Recurrent neural networks with Long Short-Term Memory is an effective and scalable model used for learning problems related to sequential data [27].

Since we have the movies every actor was part of in chronological order and the newer movies would have a greater impact in the future success of an actor compared

to older ones then LSTM seems the perfect fit.

These are a few ideas that could be used to improve the prediction of success of actors and motion-pictures.

APPENDIX A

Appendix

Feature Importance			
RF-Model-1		**RF-Model-2**	
Feature	**Value**	**Feature**	**Value**
feature_binary_2	0.034928	budget_3	0.022584
budget_3	0.023643	budget_1	0.021633
feature_binary_1	0.022763	star_3_1	0.021282
budget_1	0.022404	box_office_3	0.021199
star_3_1	0.021764	budget_2	0.021189
box_office_3	0.021281	no_reviews_imdb_1	0.020648
cast_score_3	0.021184	actor_score_1	0.020631
cast_score_1	0.021130	actor_score_3	0.020592
budget_2	0.021029	feature_binary_3	0.020569
cast_score_2	0.020851	cast_score_2	0.020419
star_2_1	0.020844	cast_score_3	0.020389
star_3_3	0.020601	star_3_2	0.020248
star_3_2	0.020426	box_office_1	0.020228
star_1_1	0.020345	producer_score_3	0.020167
producer_score_2	0.020155	cast_score_1	0.020124
cinematography_score_2	0.020122	no_reviews_imdb_2	0.020120
producer_score_3	0.020108	no_reviews_imdb_3	0.020067
no_reviews_imdb_1	0.020042	star_2_1	0.019953
star_2_3	0.019883	box_office_2	0.019908
actor_score_1	0.019733	producer_score_2	0.019893
producer_score_1	0.019717	star_1_1	0.019715
star_1_3	0.019592	producer_score_1	0.019703
box_office_1	0.019538	actor_score_2	0.019573
star_1_2	0.019508	no_reviews_users_1	0.019516
cinematography_score_3	0.019441	star_3_3	0.019456
star_2_2	0.019331	cinematography_score_2	0.019158
cinematography_score_1	0.019280	star_2_3	0.019065
box_office_2	0.019211	no_reviews_users_2	0.019057
feature_binary_3	0.018688	no_reviews_critics_1	0.018978
no_reviews_critics_1	0.018070	star_1_2	0.018977

no_reviews_imdb_2	0.017893	star_2_2	0.018929
meta_2	0.017440	no_reviews_users_3	0.018724
no_reviews_imdb_3	0.017194	cinematography_score_3	0.018715
no_reviews_users_1	0.017020	star_1_3	0.018668
no_reviews_users_2	0.016785	cinematography_score_1	0.018591
no_reviews_critics_2	0.016316	no_reviews_critics_2	0.018446
actor_score_3	0.016166	no_reviews_critics_3	0.018024
meta_3	0.016085	meta_2	0.017570
meta_1	0.015849	meta_3	0.017220
no_reviews_critics_3	0.015741	meta_1	0.016783
actor_score_2	0.015554	IMDb_3	0.015896
director_score_2	0.015456	director_score_2	0.015789
no_reviews_users_3	0.015445	director_score_3	0.015765
director_score_3	0.015432	IMDb_2	0.015627
writer_score_1	0.015401	writer_score_2	0.015510
writer_score_2	0.015316	writer_score_3	0.015461
writer_score_3	0.015210	score_productions_co_3	0.015268
score_productions_co_3	0.015166	writer_score_1	0.015253
director_score_1	0.015152	director_score_1	0.015193
score_productions_co_2	0.014784	feature_binary_2	0.015124
score_productions_co_1	0.014544	IMDb_1	0.015077
IMDb_3	0.014247	score_productions_co_2	0.014942
IMDb_2	0.013408	score_productions_co_1	0.014908
IMDb_1	0.012785	feature_binary_1	0.013476

Table A.1: Feature Importance values for **RF-Model-1** and **RF-Model-2**.

Feature Importance			
RF-Model-3		**RF-Model-4**	
Feature	**Value**	**Feature**	**Value**
feature_binary_2	0.063047	feature_binary_3	0.288410
feature_binary_1	0.040366	feature_binary_2	0.103711
feature_binary_3	0.032714	feature_binary_1	0.078384
budget_3	0.024993	budget_1	0.014626
box_office_3	0.023062	budget_3	0.014170
budget_1	0.022701	cast_score_1	0.013364
no_reviews_imdb_1	0.022617	budget_2	0.013110
star_3_1	0.021184	cast_score_2	0.012995
cast_score_2	0.020557	cast_score_3	0.012857
cast_score_3	0.020457	box_office_3	0.012604
budget_2	0.020092	star_3_1	0.012415
cast_score_1	0.019962	box_office_2	0.012128
star_3_2	0.019361	producer_score_3	0.012083

star_2_1	0.019327
actor_score_1	0.019251
star_3_3	0.019158
no_reviews_imdb_2	0.019115
producer_score_2	0.019038
producer_score_3	0.019000
star_1_1	0.018988
no_reviews_imdb_3	0.018834
box_office_2	0.018475
producer_score_1	0.018452
box_office_1	0.018429
star_2_3	0.018407
no_reviews_critics_1	0.018101
star_1_2	0.018078
star_2_2	0.017933
star_1_3	0.017825
no_reviews_users_2	0.017517
cinematography_score_2	0.017300
no_reviews_users_1	0.016588
cinematography_score_3	0.016466
score_productions_co_3	0.016321
actor_score_3	0.016311
cinematography_score_1	0.015904
meta_2	0.015780
score_productions_co_2	0.015275
no_reviews_critics_2	0.014984
actor_score_2	0.014763
score_productions_co_1	0.014609
no_reviews_users_3	0.014213
no_reviews_critics_3	0.014040
meta_3	0.013957
meta_1	0.013815
IMDb_3	0.012616
director_score_2	0.012598
director_score_3	0.011649
writer_score_2	0.011465
IMDb_2	0.011232
director_score_1	0.011140
writer_score_3	0.011044
IMDb_1	0.010505
writer_score_1	0.010384

producer_score_2	0.011965
star_1_1	0.011965
star_2_1	0.011955
star_3_2	0.011936
star_2_2	0.011720
no_reviews_imdb_1	0.011698
star_3_3	0.011659
star_1_2	0.011633
actor_score_1	0.011529
no_reviews_critics_1	0.011523
box_office_1	0.011477
producer_score_1	0.011351
no_reviews_imdb_2	0.011039
star_1_3	0.010933
cinematography_score_2	0.010735
star_2_3	0.010590
no_reviews_imdb_3	0.010403
cinematography_score_1	0.010177
cinematography_score_3	0.010050
no_reviews_users_2	0.009905
score_productions_co_3	0.009871
no_reviews_users_1	0.009614
no_reviews_critics_2	0.009501
actor_score_3	0.009480
score_productions_co_2	0.009393
no_reviews_users_3	0.009345
score_productions_co_1	0.009258
meta_2	0.009254
no_reviews_critics_3	0.008904
actor_score_2	0.008886
meta_1	0.008853
meta_3	0.008614
director_score_2	0.007941
IMDb_3	0.007461
writer_score_2	0.007444
director_score_3	0.007346
director_score_1	0.007135
IMDb_2	0.007028
writer_score_3	0.006784
writer_score_1	0.006761
IMDb_1	0.006025

Table A.2: Feature Importance values for **RF-Model-3** and **Model-4**.

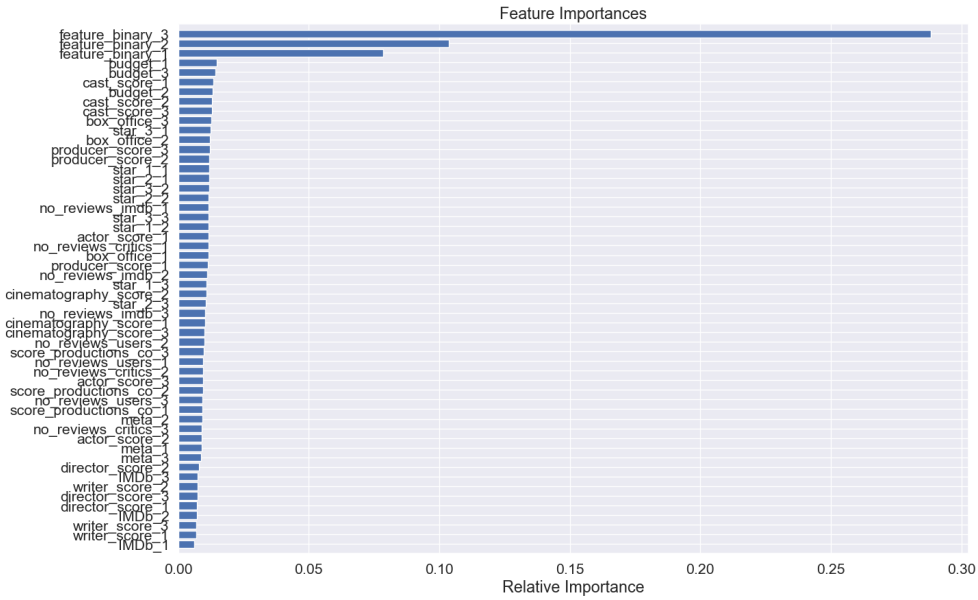


Figure A.1: Feature Importance for Rf-Model-4, this is suing the full data set for the motion-pictures that have budget.

Feature Importance	
RF-Model-6	
Feature	**Value**
box__office__4	0.0207479
star__4__1	0.0186271
cast__score__1	0.01826064
cast__score__4	0.01820685
box__office__3	0.0182034
cast__score__3	0.018166
star__3__1	0.0181485
producer__score__4	0.017963
star__4__3	0.017926
cast__score__2	.017925
star__2__1	0.017867
producer__score__2	0.017784
producer__score__3	0.0176110
box__office__2	0.017582
star__3__3	0.017467
producer__score__1	0.017334
box__office__1	0.017317

star_2_2	0.017226
cinematography_score_2	0.017201
star_1_1	0.0171149
cinematography_score_4	0.017108
star_1_2	0.017082
star_2_3	0.016881
cinematography_score_1	0.0167068
cinematography_score_3	0.016702
star_1_3	0.016627
no_reviews_imdb_2	0.0157879
actor_score_1	0.0148467
no_reviews_critics_1	0.01459
meta_4	0.0145167
no_reviews_critics_4	0.014412
meta_3	0.014287
actor_score_4	0.01427
meta_2	0.013978
meta_1	0.0136687
writer_score_4	0.013602
writer_score_2	0.013547
writer_score_1	0.013514
no_reviews_users_2	0.0134962
score_productions_co_3	0.0134656
no_reviews_critics_2	0.0134142
no_reviews_users_1	0.0132957
no_reviews_users_3	0.0132862
director_score_4	0.013246
writer_score_3	0.0132267
score_productions_co_4	0.0131006
no_reviews_critics_3	0.0130657
score_productions_co_2	0.01304814
director_score_3	0.013027364
director_score_2	0.0130022
director_score_1	0.0128121
actor_score_2	0.0126196
IMDb_4	0.012599
score_productions_co_1	0.012551
feature_binary_4	0.012288
actor_score_3	0.01192
IMDb_3	0.011803
IMDb_2	0.011683

feature_binary_2	0.011616
IMDb_1	0.01157
feature_binary_1	0.010957
star_3_2	0.009978
star_3_2	0.009958
feature_binary_3	0.00992
no_reviews_imdb_3	0.00891
no_reviews_imdb_3	0.00886

Table A.3: Feature Importance values for RF-Model-6. From most important(top) to less important (bottom).

Feature Importance	
RF-Model-0	
Feature	**Value**
cast_score_3	0.0286
star_1_1	0.0278
actor_score_1	0.0278
star_3_3	0.0277
star_3_1	0.0277
star_1_2	0.0277
star_1_3	0.0277
star_3_2	0.0276
producer_score_3	0.0273
star_2_2	0.0273
star_2_3	0.0272
star_2_1	0.0271
cinematography_score_3	0.0269
cinematography_score_1	0.0268
actor_score_3	0.0266
writer_score_3	0.0265
score_productions_co_3	0.0264
cinematography_score_2	0.0263
cast_score_2	0.0261
producer_score_1	0.0261
writer_score_1	0.0261
director_score_3	0.02608
cast_score_1	0.0260
director_score_1	0.0259
producer_score_2	0.02573
writer_score_2	0.0256

director_score_2	0.0254
actor_score_2	0.0246
no_reviews_imdb_1	0.02379
no_reviews_imdb_3	0.0235
score_productions_co_2	0.0233
no_reviews_imdb_2	0.0231
score_productions_co_1	0.0229
meta_3	0.0137
IMDb_3	0.0135
meta_1	0.0135
IMDb_1	0.0132
meta_2	0.0132
IMDb_2	0.0131
no_reviews_users_1	0.0111
no_reviews_users_3	0.0107
no_reviews_users_2	0.0106
no_reviews_critics_1	0.0073
no_reviews_critics_3	0.0071
no_reviews_critics_2	0.0071

Table A.4: Feature Importance values for RF-Model-0. From most important(top) to less important (bottom).

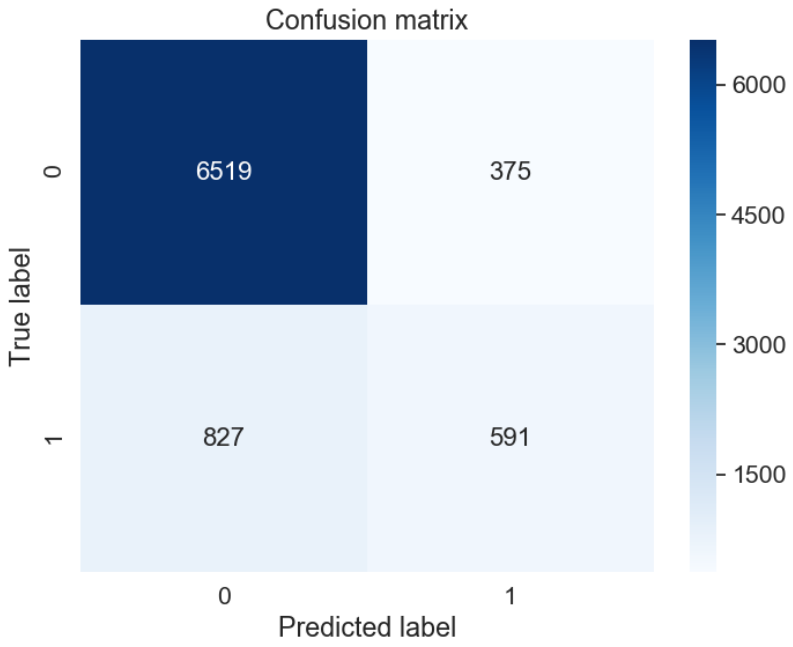


Figure A.2: Confusion Matrix for Rf-Model-4, this is suing the full data set for the movies that have budget.

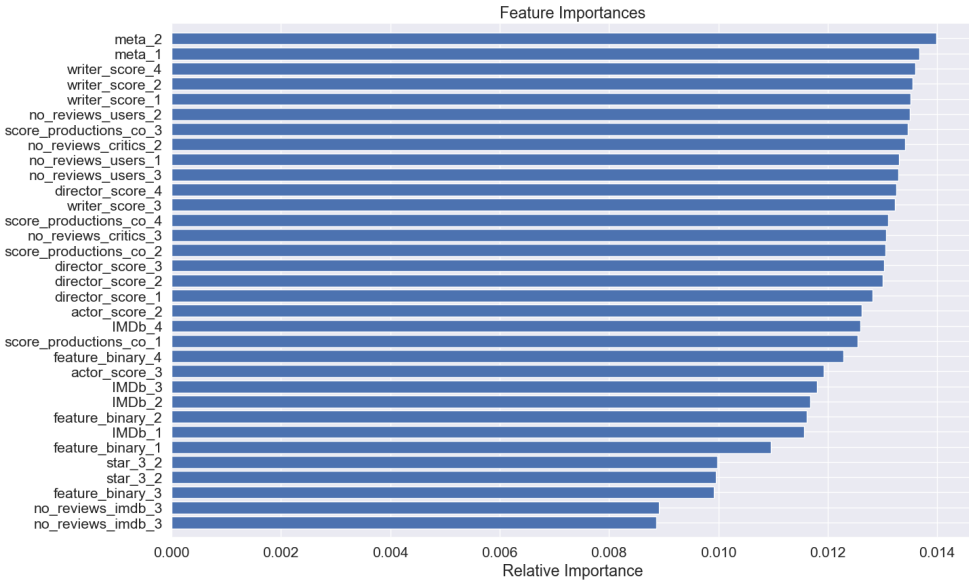


Figure A.3: Bottom 34 Feature Importance for Rf-Model-6 found with random search

Bibliography

[1] Motion Picture Association, INC., *Mpaa theme report 2018*. [Online]. Available: https://www.motionpictures.org/.

[2] X. Wang, B. Yucesoy, O. Varol, T. Eliassi-Rad, and A.-L. Barabási, "Success in books: Predicting book sales before publication," *EPJ Data Science*, 2019.

[3] M. Mestyán, T. Yasseri, and J. Kertész, "Early prediction of movie box office success based on wikipedia activity big data," *PLOS ONE*, volume 8, number 8, 2013. DOI: 10.1371/journal.pone.0071226.

[4] A. Oghina, M. Breuss, M. Tsagkias, and M. de Rijke, "Predicting imdb movie ratings using social media," *Springer-Verlag*, 2012, 34th European Conference on Information Retrieval.

[5] M. Janosov, F. Battiston, and R. Sinatra, "Success and luck in creative careers," 2019. [Online]. Available: http://arxiv.org/abs/1909.07956.

[6] S. Sreenivasan, "Quantitative analysis of the evolution of novelty in cinema through crowdsourced keywords," 2013. [Online]. Available: https://arxiv.org/pdf/13040786.pdf.

[7] R. K. Pan and S. Sinha, "The statistical laws of popularity: Universal properties of the box-office dynamics of motion pictures," *New Journal of Physics*, volume 12, November 2010.

[8] O. E. Williams, L. Lacasa, and V. Latora, "Quantifying and predicting success in show business," 2019. [Online]. Available: https://arxiv.org/pdf/1901.01392.pdf.

[9] L. Breiman, "Random forests," in *Machine Learning*, 2001, pages 5–32.

[10] T. Hastie, R. Tibshirami, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2017, ISBN: 9780387848587.

[11] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Wadsworth & Brooks/Cole Advanced Books & Software, 1984, ISBN: 978-0-412-04841-8.

[12] L. Breiman and R. Ihaka, "Nonlinear discriminant analysis via scaling and ace," Statistics Department Yale University, Tech. Rep., December 1984, https://digitalassets.lib.berkeley.edu/sdtr/ucb/text/40.pdf.

[13] IMDb, *Internet movie database*, Accessed: 2019-09-01. [Online]. Available: https://www.imdb.com/.

[14] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pages 51–56.

[15] CBS Interactive, *Metacritic*. [Online]. Available: http://www.metacritic.com.

[16] M. Andreas Christian Mueller, *Wordcloud*, 2012. [Online]. Available: https://github.com/amueller/word_cloud.

[17] D. M. Lane and Rice University, *Online statistics education: A multimedia course of study*. [Online]. Available: http://onlinestatbook.com.

[18] Los Angeles Times Data and Graphics Department, *Cpi*, 2017. [Online]. Available: https://github.com/datadesk/cpi.

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

[20] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, volume 9, number 3, pages 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[21] G. Haixiang, Y. Li, J. Shang, G. Mingyun, H. Yuanyue, and B. Gong, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, volume 73, December 2016. DOI: 10.1016/j.eswa.2016.12.035.

[22] P. Probst and A.-L. Boulesteix, "To tune or not to tune the number of trees in random forest?," 2017. [Online]. Available: https://arxiv.org/pdf/1705.05654.pdf.

[23] P. Probst, M. Wright, and A.-L. Boulesteix, "Hyperparameters and tuning strategies for random forest," 2019. [Online]. Available: https://arxiv.org/pdf/1804.03515.pdf.

[24] Y. Bengio and J. Bergstra, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, volume 13, pages 281–305, February 2012. DOI: 10.1016/j.eswa.2016.12.035.

[25] Nash Information Services, LLC., *The numbers - where data and movies meet*. [Online]. Available: https://www.the-numbers.com/.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, volume 9(8), 1997.

[27] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Elsevier journal "Physica D: Nonlinear Phenomena"*, volume 404, March 2020.