Introduction to Machine Learning and Data Mining
Danmarks Tekniske Universitet
Fall 2023
Group 50

Report 2

Name	Section 1	Section 2	Section 3	Section 4	Section 5
Csaba Hell - s232456	30%	40%	30%	33.3%	33.3%
Gabriel Lanaro - s233541	40%	30%	30%	33.3%	33.3%
Gabriele Turetta - s233124	30%	30%	40%	33.3%	33.3%

Table 1: Contribution of participants in each section

Contents

1 Regression part A

In the first report, we delved into the analysis of a dataset related to the film industry. This dataset, originally crafted by Professor James Gaskin for educational purposes, offers a wealth of information regarding movies. To help the reader better understand this second report, below is a summary table of the features describing each entry in the dataset.

Attribute	**Description**
MovieID	A unique identifier for each movie.
Title	The title of the movie. This attribute is used solely for identification purposes and is not involved in the calculations or specific analyses of the model.
MPAA_Rating	The rating of the movie by the Motion Picture Association of America (MPAA) (G – General Audiences, PG – Parental Guidance Suggested, PG-13 – Parents Strongly Cautioned, R – Restricted).
Budget	The budget allocated for the production of the movie.
Gross	The total box office revenue generated by the movie.
Release_Date	The release date of the movie.
Genre	The genre or category to which the movie belongs (e.g., Action, Comedy, Drama, etc.).
Runtime	The duration of the movie in minutes.
Rating	The average rating from 0 to 10 given to the movie.
Rating_Count	The number of user ratings received by the movie.

Table 2: Attributes of the dataset and their descriptions.

1.1 Introduction

Our primary focus in the first report was to provide a comprehensive understanding of the dataset, including data issues, summary statistics, data visualizations, and a principal component analysis (PCA). Building upon the insights gained from the first report, our second report aims to take a deeper dive into the dataset, exploring regression analysis to predict the variable *gross* (box office earnings) based on the other features and understand the factors influencing a movie's financial performance. Our primal aim is hopefully to provide valuable insights into the financial dynamics of the film industry. In the first report, one of our objectives was also to predict the movie rating through regression. However, since the preliminary analyses conducted have shown that rating prediction is unreliable, in this second report, we have adjusted our goals and focused solely on predicting gross revenue.
To achieve this goal, a preliminary transformation of some nominal features was necessary to work with them. Specifically, the features MPAA_rating and genre were converted, using the one-out-of-k encoding, creating four columns for the MPAA_rating and sixteen for the genre feature.
Furthermore, since regularization was employed for linear regression, the matrix $\mathbf{X}$ was standardized such that each column has a mean of 0 and a standard deviation of 1. We decided to standardize the target variable too. Standardizing the target variable y = gross revenue , is a crucial step in regression analysis for several reasons. Firstly, it ensures that all variables, including the target variable, are on a consistent scale, addressing potential issues where the scale of y is significantly larger than other features. This consistency is important for the convergence of optimization algorithms, as it enhances numerical stability and accelerates the convergence of models like Ridge regression. Standardizing y also facilitates the interpretability of coefficients by making them directly comparable. Additionally, it ensures that the regularization parameter (λ) in Ridge regression is applied consistently across all features, contributing to a more straightforward interpretation. Beyond these technical considerations, standardization aids in the prevention of model bias toward extreme values and outliers, promoting a more robust and reliable regression model.

1.2 Lambda choice

In this section, we focus on the introduction of a regularization parameter, λ. λ plays a crucial role in balancing model complexity and training data fitting. Our goal is to estimate the generalization error across different λ

values. We initially cast a broad net, ranging from 1e-6 to 400. This extensive exploration allowed us to capture the nuanced behaviour of the generalization error concerning λ.
Subsequently, armed with insights from the initial range, we fine-tuned our λ search. Recognizing that the most informative range lay within more constrained bounds, we refined our exploration to λ values between 1 and 300. This targeted adjustment aimed to zoom in on the region where generalization error exhibits a distinct diminishing-then-increasing pattern.
Using a robust 10-fold cross-validation, we rigorously evaluated generalization error for each λ value. The objective was to identify the λ that strikes an optimal balance, enhancing model adaptability and predictive accuracy.

To visually understand how the Mean Squared Error (MSE) changes with different regularization strengths (λ) in linear regression, we created a helpful figure (see Figure 1). On the graph, the y-axis shows the MSE, which measures how accurate our predictions are, and the x-axis represents different λ values.
The graph tells an interesting story. Starting from $\lambda = 1$, the MSE decreases smoothly, reaching its lowest point at $\lambda = 32$ with an impressive value of 0.4747. This low point indicates the best balance between keeping our model simple and staying faithful to the training data, hitting a sweet spot in regularization.
After this point, the MSE starts to rise, highlighting the ongoing trade-off between regularization and how well our model fits the data. The upward trend emphasizes how regularization increasingly influences the model, showing the delicate balance between underfitting and overfitting.

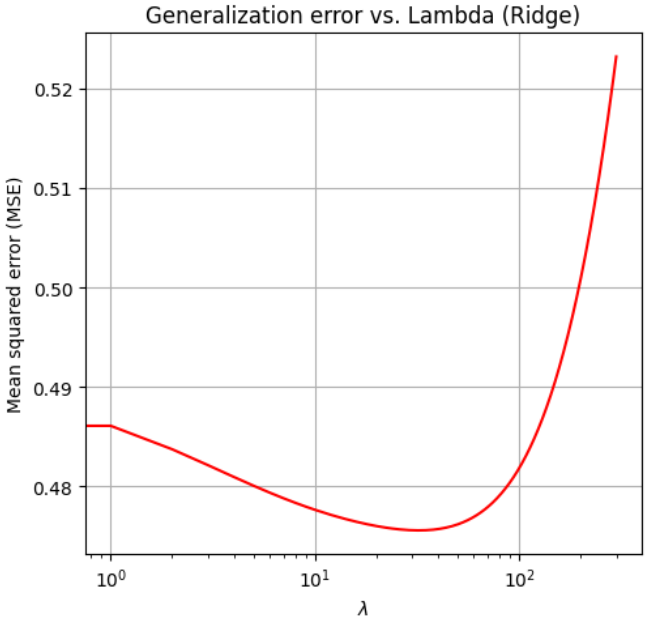

Figure 1: Generalization error for different values of Lambda

1.3 Output computation

To compute the output (y) of the linear model with the lowest generalization error, we relied on the regression coefficients and the intercept term obtained through meticulous model training and cross-validation. The linear model is expressed as:

$$y = \text{Intercept} + \sum_{i=1}^{n} (\text{Coeff}_i \times \text{Feature}_i)$$

Here, the Intercept is the bias term, Coeff_i is the coefficient associated with the i-th feature, and Feature_i is the value of the i-th feature in the input x. The effect of an individual attribute in x on the output y is encapsulated by the corresponding coefficient (Figure 2). Positive coefficients signify a positive impact on the predicted "*Gross*" revenue, while negative coefficients indicate a negative impact. Notably, the chosen features, including *'Budget,' 'release_date'*, and *'rating_count*,' exhibit meaningful influences on the predicted revenue. Figure 3 illustrates

the dynamic changes in various coefficients and their influence on the revenue variable *Gross*. These results are consistent with the observations made in the initial report, where we identified the *budget* as a primary factor influencing the gross, closely followed by *release_ date* and *rating_ count*. Thus, these outcomes were as anticipated. The analysis also provided an interesting perspective on how individual *genres* and *MPAA_ ratings* influence the gross, facilitated by the use of one-hot encoding. This technique allowed us to discern the distinct impact of each genre and *MPAA_ rating* on gross revenue, providing valuable insights into their singular contributions to the overall variability.

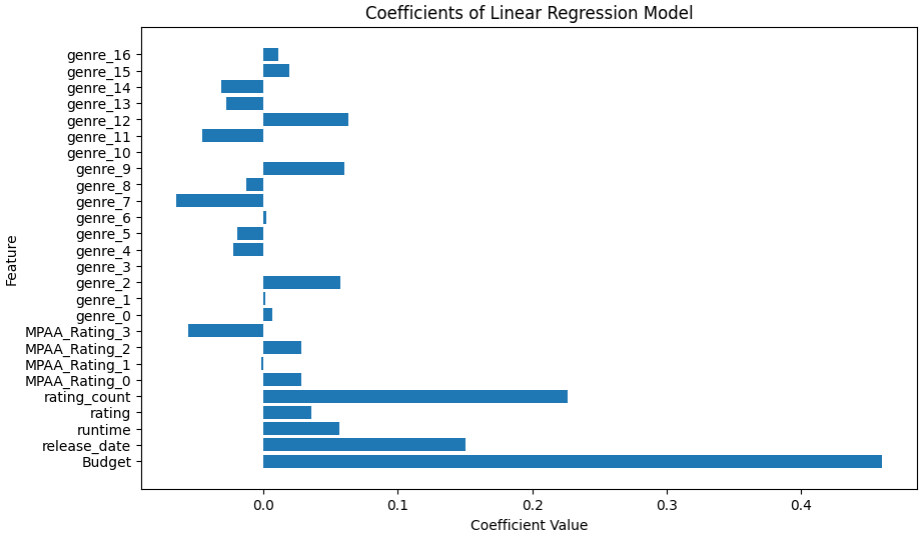

Figure 2: Influence of each coefficient in the linear regression model

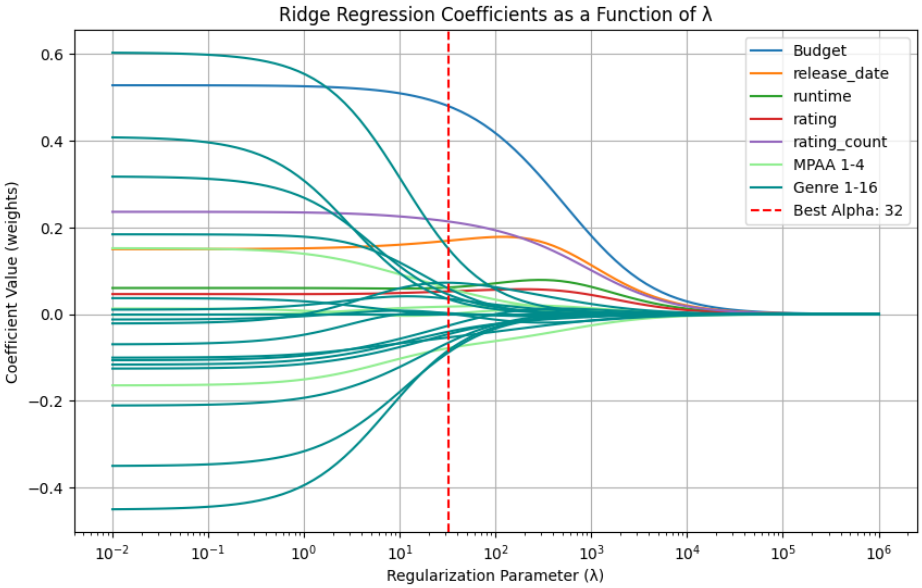

Figure 3: Behavior of the values of the coefficients when increasing the regularization parameter

2 Regression Part B

2.1 Comparison between the 3 models

In the pursuit of a robust comparative analysis, we implemented a two-level cross-validation. This methodology, featuring K1 = K2 = 10 folds, aimed to evaluate and compare three distinct models: Baseline, Ridge Regression, and Artificial Neural Network (ANN).
To establish a solid foundation, the Baseline model was introduced as the simplest reference point. This linear regression model, without any features, computes the mean of y on the training data and utilizes this value for predicting y on the test data. The Baseline's performance, reflected in the Mean Squared Error (MSE), serves as a critical benchmark for evaluating the efficiency of more complex models. Ridge Regression, a linear regression variant, has become a crucial component. This model includes regularization, controlled by the hyperparameter λ, to address potential overfitting. Through thorough testing, we identified a suitable range for λ, in the range from 1 to 200. This range facilitates a systematic exploration of regularization's impact on Ridge Regression across diverse folds. For the ANN model, our focus extended to configuring the number of hidden units h as the complexity-controlling parameter. After conducting preliminary test runs, we established a reasonable range for h, from 1 to 6. This range, including $h = 1$ enables a comprehensive evaluation of ANN's performance under varying levels of complexity.

In this analysis of regression methods applied to predict the "Gross" feature in the Movies dataset, the three distinct training approaches described above were evaluated: Baseline, Ridge Regression, and Neural Network (ANN). The Baseline model, serving as a reference, exhibited a relatively high Mean Squared Error (MSE), as shown in Table 3. Ridge Regression, characterized by the optimal regularization parameter λ, consistently outperformed the Baseline, achieving lower MSE values across different folds. The optimal Lambda values varied among folds but generally fell within the range of 134 to 172. In contrast, the Neural Network's performance showed greater variability, occasionally yielding similar or higher MSE compared to the Baseline. The optimal number of hidden units in the Neural Network varied across folds, indicating a lack of a universally effective configuration. Further exploration and optimization of Neural Network parameters may be needed to enhance its predictive capabilities.
The detailed results, including optimal configurations and generalization errors for each method, are summarized in Table 3.

Fold	**Baseline**	**Ridge**		**ANN**	
	E_i^{test}	λ_i^*	E_i	h_i^*	E_i^{test}
1	1.095818	143	0.649666	5	1.095856
2	1.115021	163	0.740625	3	1.114693
3	0.442819	134	0.641152	5	0.442095
4	1.760649	148	0.677320	2	1.759958
5	1.842502	152	0.657419	1	1.842152
6	0.999482	146	0.663880	4	0.999483
7	0.742797	159	0.665875	1	0.742428
8	0.539400	172	0.653787	2	0.538906
9	0.511695	155	0.653789	4	0.511661
10	1.026668	165	0.690491	3	1.026703

Table 3: Comparison of Regression Methods

2.2 Statistical Evaluation

In this section, we performed a pairwise statistical evaluation of three models: the baseline, linear regression, and artificial neural network (ANN). We opted for the Setup I and applied the paired t-test method to compare the models and determine if one outperformed the other or if there were significant differences in their performances. We used mean square error loss as a measurement of performance.
There is a notable difference in performance between the baseline and linear regression models. The estimated performance difference falls between approximately 0.39 and 0.66, with a confidence interval well clear of 0. The low p-value ($p < 0.01$) strengthens the argument that this result is not likely due to chance. In conclusion, linear regression demonstrates a significantly better performance than the baseline model.
Surprisingly, the t-test between the baseline and ANN does not provide clear evidence that one model is better than the other. The confidence interval ranges from -0.04 to 0.03 which includes zero, and the associated p-value is high ($p > 0.73$), suggesting no significant difference in performance between the baseline and ANN.
The comparison between linear regression and ANN shows a significant difference in performance. The confidence interval for the performance difference ranges from -0.68 to -0.38, therefore well clear of 0 and the low p-value ($p < 0.01$) indicates the result is not likely to be due to chance. All in all, we can conclude that the linear regression model performs significantly better than the ANN model.
As a result of the statistical evaluation, we can conclude that the regression model outperforms both the baseline and ANN models on the selected evaluation metric. Therefore, the regression model is recommended over both the baseline and ANN. We expected better results by the ANN. Since the dataset is quite small, there is a higher chance that the ANN is overfitted through the learning process. To solve this problem we tried different ways and adjusting parameters to fix this issue. However, it did not lead to improved results.

3 Classification

The initial classification aim was to classify the movies' genre and MPAA rating based on the other features. However, as highlighted in the first report, the PCA revealed that the observations labelled according to their genre and MPAA rating did not appear to separate optimally along the first 2 principal components. Therefore, we adjusted our goals and decided to reduce the number of genres, considering only Action, Comedy, Horror, and War, similarly, for the MPAA rating, we tried to focus only on PG (Parental Guidance Suggested) and R (Restricted). With these restrictions, the data seemed to cluster more visibly. For the purposes of this second report, only one classification example has been implemented, and we have chosen to perform binary classification with respect to the feature MPAA_rating, classifying instances as either PG-rated or R-rated. Therefore, to accomplish this, the initial dataset was limited to films that are classified as R or PG, totalling 303 in all, and the MPAA feature was converted into numerical values using the label encoding technique, assigning a unique integer to each category. More specifically, PG was assigned to value 1 and R to value 3. While we understand that this binary classification may appear limiting or less interesting compared to other options, such as genre-based classification or rating evaluation against a specific threshold, it was chosen for educational purposes to apply the concepts covered during the lessons of the course and hopefully to produce positive classification results. We performed binary classification using three different models:

- Logistic regression
- Baseline
- KNN

A detailed discussion of each model is presented in the following paragraphs.

3.1 Logistic Regression

Logistic Regression is a statistical method used for binary classification tasks, where the goal is to predict the probability that an instance belongs to a particular class using the logistic function. Logistic regression starts with a linear combination of input features. Each feature is multiplied by its corresponding weight, and these products are summed up along with a bias term. The linear combination is then passed through a logistic function, also known as a sigmoid function, that transforms the linear combination into a value between 0 and 1. The output of the logistic function represents the probability that the instance belongs to the positive class or the negative class.

To compare the logistic regression model with the other two models, a 2-level cross-validation with 10 outer folds and 10 inner folds was conducted.
Similarly to the regression model in the previous chapter, a regularization parameter λ was introduced to control overfitting and enhance the model's generalization ability, by adding a penalty term to the model's weights, preventing them from becoming too large.
After choosing a range of λ values, for each outer folder of the cross-validation an inner 10-fold cross-validation was performed for each λ to estimate its error measure, calculated as:

$$E = \frac{\text{number of misclassified observations}}{\text{total number of test samples}}$$

As a result, this process yielded a total of 10 optimal λ values, that correspond to the absolute minimum of the plots shown in figure 6 in the Appendix. Each plot corresponds to the result of an internal 10-cross validation and shows the different values of lambda chosen, and their associated measurement error. After these 10 optimal lambda values were found, each lambda was used to train a logistic regression model on the outer fold training dataset. The same model was then tested on the test set of the outer fold, producing a final measurement error associated with that value of lambda.

At the end of the outer cross-validation, the algorithm therefore provided 10 optimal lambda values (one per fold) and the 10 associated measurement errors, as shown in table 4. It is important to emphasize that the choice of the lambda parameter was made through a trial-and-error approach. We started by selecting a wide range of values, attempting to maintain a pseudo-logarithmic trend within a range from 0.01 to 10000. We then analyzed the obtained plots, focusing on the lambda values in the area where the generalization error first drops and then increases.
The goal was to identify the lambda that provides the best balance between the model's ability to adapt to the data (avoiding overfitting) and its ability to generalize to new data (reducing test error).

3.2 model 2: KNN

K-Nearest Neighbors is a classification algorithm that classifies a data point based on the majority class of its k nearest neighbours in the feature space. The k parameter influences the algorithm's sensitivity to local variations and was therefore used as the complexity controlling parameter. Similar to logistic regression, a 2-level cross-validation with 10 outer folds and 10 inner folds was applied. A range of k values was chosen and for each value, an inner 10-fold cross-validation was applied. The resulting k from each inner cross-validation was then used to train and test a KNN model on the outer fold, resulting in 10 k values (one for each outer fold) along with their respective classification errors, as shown in table 4. In general, the choice of the parameter was made seeking a compromise between a large k, which generally makes the model less sensitive to noise in the dataset, and a small k, which makes the model more responsive to variations in the data but may be more sensitive to noise or outliers.

3.3 Baseline

A baseline is a simple and often naive model that serves as a reference point for comparing the performance of more sophisticated models. It represents the minimum expected level of accuracy and in our case, it consists of a model which computes the largest class on the training data, and predicts everything in the test-data as belonging to that class. Just like for the other two models, a 2-level cross-validation was applied, allowing us to obtain a classification error for each outer fold, as can be seen in table 4.
As expected, the simple baseline model yielded poor results, with classification errors exceeding 40% in each external fold, meaning it incorrectly predicted the MPAA rating for more than 40% of the instances. It is evident that this type of model cannot be used for predictive purposes but only as a benchmark for evaluating the performance of more complex models.

3.4 Comparison between the 3 classifiers

In table 4, the classification errors of each model on each external fold of cross-validation are reported. As anticipated, the baseline consistently demonstrates the highest classification error across all outer folds, ranging from 0.445 to 0.487. This underscores the simplistic nature of the baseline, which predicts the majority class, and its inability to capture the underlying patterns in the data. Thus, predicting the MPAA ratings solely based on the majority class in the training set is not a reliable approach. The baseline serves its only purpose as a benchmark for

Outer fold	**KNN**		**Logistic regression**		**Baseline**
	k	E_i	λ	E_i	E_i
1	3	0.258	0.1	0.226	0.474
2	8	0.258	0.6	0.290	0.452
3	5	0.161	0.1	0.161	0.445
4	7	0.267	0.1	0.300	0.476
5	5	0.133	2	0.167	0.476
6	3	0.200	0.9	0.267	0.480
7	7	0.233	7	0.267	0.487
8	5	0.200	2	0.233	0.469
9	3	0.300	2	0.300	0.465
10	5	0.133	0.7	0.200	0.462

Table 4: Two-level cross-validation table to compare the three models in the classification problem

more complex models, and, looking at the table, it is evident that for each fold both KNN and Logistic Regression consistently outperform it. The classification errors for KNN and Logistic Regression range from 0.133 to 0.300 and from 0.161 to 0.300, respectively, showcasing a notable reduction compared to the baseline's errors ranging from 0.445 to 0.487.
Looking closer at KNN, the model exhibits varying degrees of consistency across folds. The classification errors range from 0.133 to 0.300, suggesting that the performance is somewhat stable but not entirely uniform, probably due to the presence of specific patterns in some portions of the data or the heterogeneous complexity of the dataset. For the same reason also the optimal value of k (number of neighbors) varies across folds, with different folds favoring different values, suggesting sensitivity to the choice of the hyperparameter. It's interesting to notice that KNN's performance is noteworthy as it outperforms both Logistic Regression and the baseline in most of the folds (figure 4). For example, in Fold 5 and 10, KNN achieves a particularly low classification error of 0.133, showcasing its ability to capture local patterns in the data effectively.
Finally, examining the logistic regression, the model behaves similarly to the KNN model demonstrating relatively consistent performance across folds, with classification errors ranging from 0.161 to 0.300. Similarly to KNN, the regularization parameter λ varies across folds, indicating sensitivity to the choice of this hyperparameter.
It is also interesting to note that the error trends across different folds for the logistic regression model are very similar to those of the KNN, as can be seen in Figure 4. This could indicate that both models are robust and generalize well across different portions of the dataset, effectively adapting to the various internal data structures present in different folds. Therefore, the classification problem addressed by our group is characterized by relationships that can be well captured by both a proximity-based model (KNN) and a linear model such as logistic regression.

3.5 Statistical Evaluation

In this section, we performed a pairwise statistical evaluation of three models: the baseline, logistic regression, and k-nearest neighbours (KNN). We opted for the Setup I and applied McNemar's test to compare the models and determine if one outperformed the other or if there were significant differences in their performances.
There is a notable difference in performance between the baseline and logistic regression models, as confirmed by McNemar's test. The estimated performance difference falls between approximately -0.3 and -0.15, with a confidence interval well clear of 0. The low p-value ($p < 1.87e\text{-}10$) strengthens the argument that this result is not likely due to chance. In conclusion, logistic regression demonstrates a significantly better performance than the baseline model.
Similarly, McNemar's test between the baseline and KNN showed a significant difference in performance. The estimated performance difference ranges from approximately -0.33 to -0.18, with a confidence interval that does not include 0. The remarkably low p-value ($p < 1.87e\text{-}10$) provides evidence that this result is not likely due to chance. All in all, KNN demonstrates a significantly better performance than the baseline model.
On the contrary, the comparison between logistic regression and KNN shows no significant difference in performance.

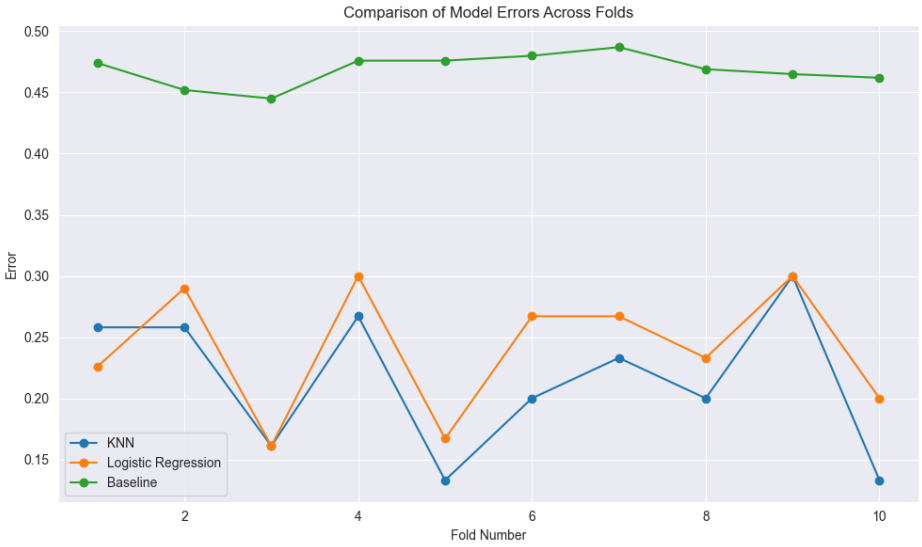

Figure 4: Comparison between the models' errors in each fold

The confidence interval for the performance difference ranges from -0.07 to 0.02 (contains 0), which indicates that there is weak evidence towards logistic regression has relatively higher accuracy than KNN, and the relatively high p-value (p = 0.32) indicates that this result is likely due to chance. In summary, there is no clear evidence favouring one model over the other.
In terms of model accuracy, the baseline achieved 53%, the logistic regression 77% and the KNN reached 80%.
As a result of the statistical evaluation, we can conclude that both logistic regression and KNN outperform the baseline model. However, there is no significant difference in accuracy between logistic regression and KNN. Thus, it is recommended to use logistic regression or KNN over the baseline model for improved accuracy but the choice between logistic regression and KNN requires deeper analysis and additional evaluation metrics.

3.6 Training of the final logistic regression model

After analyzing different values of λ and their respective classification errors, the two λ values with the lowest classification errors are $\lambda = 0.1$ (with error $E = 0.161$) and $\lambda = 2$ (with error $E = 0.167$). Although the classification error for $\lambda = 0.1$ is slightly lower, we chose to train the logistic regression model with the λ value of 2. That's because choosing a lambda value close to zero places less emphasis on the regularization process. A very low lambda value can indeed lead to a more complex model that is tailored to the specific details of the training data, with a risk of overfitting. After training the model on the original dataset, we extracted the coefficients associated with the features. The bar chart in figure 5 indicates the relative importance of different features in predicting the target variable, and it is evident that the relevant features are not the same as in the regression part, where the prediction is positively influenced by budget, release_date and rating_count.
What can be deduced from the logistic regression model training results is that the film's duration is a crucial factor, applying a significant positive influence on the classification. A longer runtime positively correlates with a higher likelihood of receiving an R rating (positive class). Additionally, a high number of ratings increases the probability that a film is classified as Restricted, perhaps suggesting that R-rated films emotionally stir the audience more, prompting them to review the film in question. It's interesting to note that the audience rating negatively influences the classification, indicating that adult-oriented films generally receive lower reviews compared to films suitable for a broader audience.
However, the feature with the most significant negative weight is the budget, showing that films intended for an adult audience tend to have lower budgets.

4 Conclusions

As stated in the first report, the original dataset used for our analysis was originally crafted for educational purposes by Professor James Gaskin, an expert in Information Systems Management at Brigham Young University. Professor

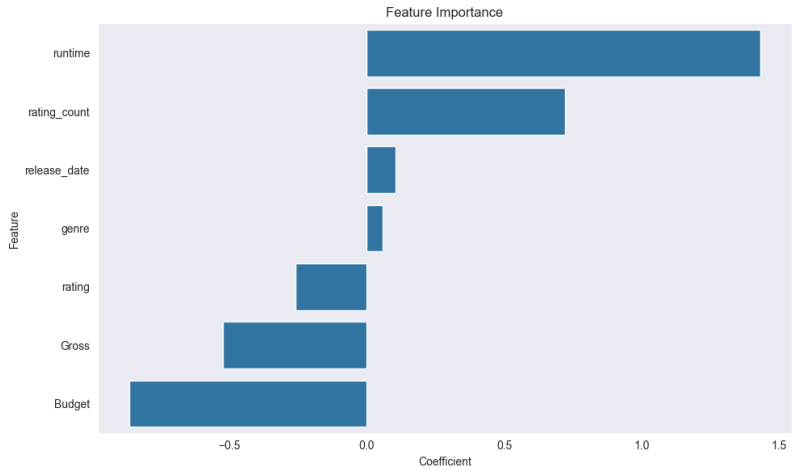

Figure 5: Influence of each coefficient in the logistic regression model

Gaskin employed this dataset to clarify the concepts of relational databases to his students. Despite the abundance of information within this dataset, unfortunately, there is no prior analysis or comprehensive study available to compare our results with, since the dataset was mainly used for teaching purposes by Professor Gaskin. The lack of previous references prompted us to resort to a more independent evaluation methodology, formulating conclusions that we hope can still hold value.

In evaluating regression models for predicting the 'Gross' feature within the Movies dataset, three key models—Baseline, Ridge Regression, and Neural Network (ANN)—were evaluated. Ridge Regression consistently outperformed the Baseline with lower Mean Squared Error (MSE) values across different folds, while the performance of the Neural Network varied, occasionally matching or exceeding the Baseline's MSE. Statistical evaluations emphasized that linear regression significantly outperformed the Baseline, showcasing a substantial performance difference. Comparisons between the Baseline and ANN did not show clear evidence of superiority, and linear regression consistently outperformed the ANN model. In conclusion, the statistical evaluations suggest that linear regression is a more effective model for predicting the 'Gross' feature in the Movies dataset, despite initial expectations for improved performance with the ANN model. In light of the unexpected performance of the ANN, it's notable to mention that the dataset's relatively small size might have contributed to the model's overfitting during the learning process. Despite our efforts to fine-tune and adjust parameters, the ANN's performance did not improve significantly. This overfitting issue highlights a challenge with small datasets impacting the generalization ability of complex models like ANN.

Regarding the classification, the analysis conducted has yielded positive results, confirming the expectations from the initial report. The goal of binary classification for movies with respect to MPAA ratings (as PG or R) was consciously chosen, despite its limited utility. This decision was influenced by the PCA results from the first report, revealing an interesting split between movies categorized as "PG" and those categorized as "R." By using this subset of the dataset to train classification models, we anticipated positive outcomes due to the artificial nature of the considered set.

The training of Logistic Regression, KNN, and Baseline models demonstrated that the Baseline, the simplest model, consistently underperformed compared to the other two across all external folds of cross-validation, with an error ranging from 0.445 to 0.487. This was confirmed by McNemar's test conducted in the Statistical Evaluation.

Logistic Regression and KNN, on the other hand, exhibited errors ranging between 0.161 and 0.300 and between 0.133 and 0.300, respectively. These models showed somewhat stable but not entirely uniform performance, possibly due to specific patterns in some portions of the data. Moreover, the similar error trends across external folds in KNN and Logistic Regression indicate that both models are robust and generalize well across different portions of the dataset.

In the end, the training of the classification model allowed us to uncover which features of a film most significantly influence the classification between "PG" and "R." It was discovered that the duration of a film and the number of ratings heavily influence this discrimination. Additionally, budget plays a crucial role in film classification, with low-budget films having a higher likelihood of being classified as Restricted. This aligns with observations from the initial report, where PCA highlighted the same result, emphasizing that the horror genre has always been the preferred choice for first-time movie directors aiming to make a film with a very low budget.

5 Exam problems for the project

1. **C**. The ROC curve shows how the model performs when you change the threshold that determines which class an instance is assigned to. Looking at candidate C and setting a threshold in $y_i \approx 0.5$ (the instance is classified as positive if $y_i <= 0.5$), we get $TPR = 0.25$ and $FPR = 0.5$. This value is consistent with the ROC curve shown in the figure. Similarly, by adjusting the threshold to other points, we obtained values that confirm C as the correct answer.

2. **C**. The question asks about the impurity gain of the split $x_7 = 2$. Looking at Table 3, it can be inferred that after the analyzed split, only one instance classified as class y = 2 falls into the left branch, while in the right branch, there are 37 instances belonging to class 1, 30 to class 2, 33 to class 3, and 34 to class 4. Given the formula for the impurity gain and using the classification error as the impurity measure $I(r)$, the result is:

   $PurityGain = I(r) - \sum(\frac{N(v_k)}{N(r)} \cdot I(v_k)) = \left(1 - \frac{37}{135}\right) - \left[\frac{1}{135} \cdot 0 + \frac{134}{135}\left(1 - \frac{37}{134}\right)\right] \approx 0,0074$

3. **A**. With 7 input nodes, a hidden layer of 10 neurons, and 4 output nodes, the number of weights to be trained to fit the neural network is calculated as

   $num\_input\_nodes * num\_hidden\_layer\_nodes +$
   $+ num\_hidden\_layer\_nodes * num\_of\_output\_nodes =$
   $= 7 * 10 + 10 * 4 = 110$ weights

   while the number of biases is calculated as $num\_hidden\_layer\_nodes + num\_output\_nodes = 10 + 4 = 14$ biases. Thus, the total number of parameters to be trained is $110 + 14 = 124$

4. **D**. Looking at Figure 3, for split C, which allows for the definitive classification of congestion level 4, the only plausible option is 'D' because it takes into consideration the split $b1 >= 0.16$

5. **C**. Given the number of outer folds $k1 = 5$, the number of inner folds $k2 = 4$ and the number of models $L = 5$, we have: $k1 * (LK2 + 1) = 5 * (5 * 4 + 1) = 105$ trainings and 105 tests. Thus, the total time taken to train and test all the models is:
   -for the ANN : $105 * 20ms + 105 * 5ms = 2625ms$
   -for the LR : $105 * 8ms + 105 * 1ms = 945ms$
   The total time to build the table is in conclusion $2625 + 945 = 3570ms$

6. **B**. Because given $\mathbf{x} = [1 - 0.6 - 1.6]$, and the 3 weights $\mathbf{w1}, \mathbf{w2}, \mathbf{w3}$, we obtain
   $\mathbf{x} * \mathbf{w1} = -2.66$,
   $\mathbf{x} * \mathbf{w2} = -2.42$,
   $\mathbf{x} * \mathbf{w3} = -1.56$
   Thus, calculating the probability that the observation belongs to class number 4, given the per-class probability function, the result is P = 0.73. There's no need to calculate the per-class probability for the other classes.

A Appendix

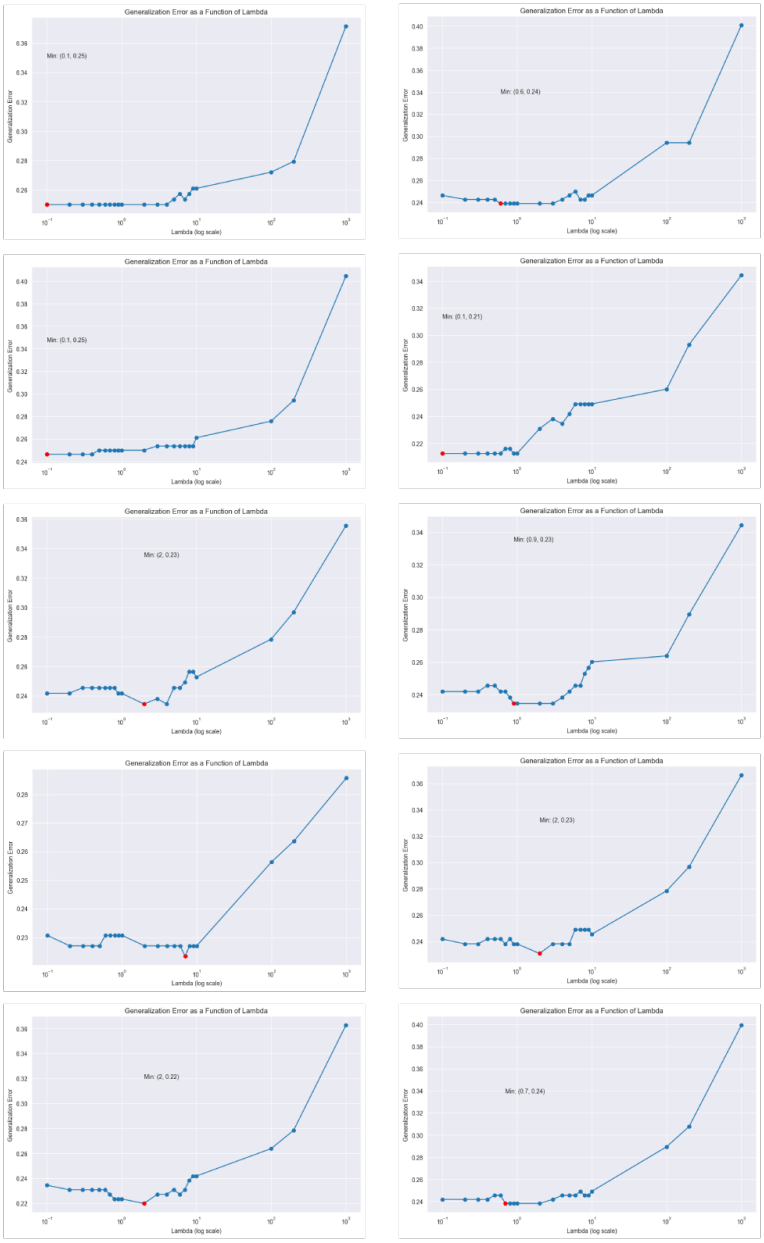

Figure 6: Plot of the tested delta values and their classification errors obtained with internal cross-validation