02450: Introduction to Machine Learning and Data Mining

Project 2

Group 65

Authors:

Gabriel Luo (s221400)

Christian Lund (s203962)

Nil Palau (s222953)

Group Member Contributions:

	Regression	Classification	Discussion	Exam Questions
s221400	30%	30%	40%	33.3%
s203962	30%	40%	30%	33.3%
s222953	40%	30%	30%	33.3%

Table of Contents

Regression

Part a

1. In the regression part, we have decided to predict the number of sales in Japan based on all other variables. This includes NA sales, EU sales, other sales, user score, critic score, genre, rating, platform, and year of release. We are omitting publisher and developer as there are too many different publishers and developers for us to use and encode them. We are applying one-of-K coding to transform rating, genre, and platform. We do this as we believe these values will have an important impact on the sales, and one-of-K is the best way for us to use the data since they are nominal values and can't simply just be assigned a number. Our thoughts are that based on the data from the sales performance in other regions and the characteristics of the game itself, our models will be able to accurately predict the number of sales in Japan.

   If our models work correctly, they could have an interesting real-life application since we could predict how a game would sell in the Japanese region beforehand. This means that companies that are planning to sell games in that region could evaluate if the investment is worth doing in the multi-billion industry.

2. We introduce a regularization parameter λ using cross-validation giving us the graph shown in Figure 1 below. Lambda values from 10^{-5} to 10^{8} were chosen initially, with steps in powers of 10. This graph showed an almost constant error for many of the lower values of lambda before there was an increase in error. To better see the expected behavior of error versus lambda, we focus on the lambda values from 2^{-5} to 2^{9}.

   Here we can see that the training error follows the ideal shape of first slowly dropping before quickly rising as a function of lambda. It shows that from 2^{-5} to 2^{1}, the training error remains relatively unchanged as the model is still under fitted. However, it then starts dropping, before eventually reaching the ideal λ of about 2^{5} or 32. The increase in error that follows is due to the gradual overfitting of the data as lambda increases. It should also be noted that the training error is lower than the test error, which is expected.

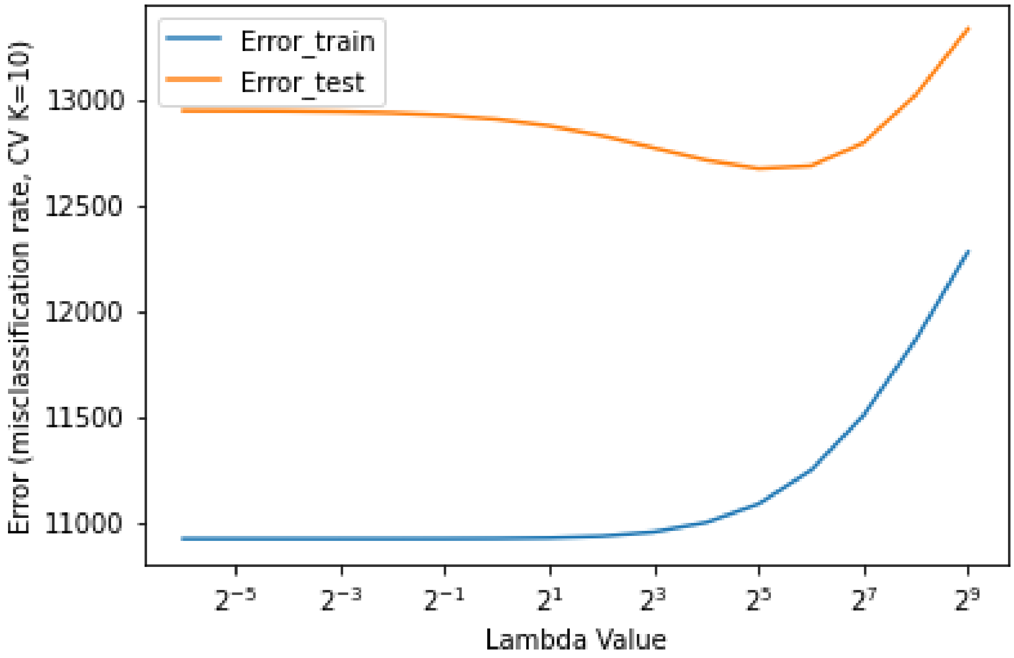

Figure 1: Plot of squared error vs lambda value for linear regression

3. By using the optimal lambda value of 32 as found from the previous question, our model calculates a coefficient or weight for each attribute, a noise, and a y-intercept. After receiving a datapoint, it then multiplies each point by the corresponding weight, before summing all the values. We do this using the following formula:

$$y_n = w_0 + \sum_k w_k x_{n,k} + \epsilon_n$$

The weights for the different attributes can be found in the second column of Table 3, which can be found in the Appendix. When looking at the different weights, several values stand out. Firstly, the year of release has a slight negative weight. This seems counterintuitive as it is expected for games to sell more copies as time moves on. Next, the amount of EU sales has a vastly higher impact on Japanese sales than North America, which has almost no impact.

Looking at which ESRB ratings have the biggest weight, it is unsurprising that the E rating has by far the largest positive weight. In fact, it has the only positive weight among all the ratings. The two ratings with the worst impact on sales are M (Mature) and T (Teen). This also is rather unsurprising as these games of course limit their audiences by not being allowed to sell to the entire population.

When looking at platforms, the results vary widely. Firstly, as expected we see that the 3DS, the Dreamcast, the DS, the GameBoy Advanced, and the GameCube all have highly positive weights. This makes sense as these products are all Japanese. Another unsurprising factor is that all 3 Xbox versions sell very poorly. This again makes sense as the product is American. A bit more surprising however, is the sale rates of the PlayStations and Wiis. The PlayStation and

PlayStation 2 have positive effects on sales, while the PlayStation 3 and 4 have equally negative effects. Additionally, the Wii has a highly negative effect on sales while the Wii U is only slightly positive. This is a big surprise as all these products are Japanese.

Lastly, for the genres, it is clear what kinds of games do best in Japan. Role-Playing games are by far the best genre, with fighting and action games being 2nd and 3rd best. In terms of the negative weights, racing-, sports-, shooter-, platform- and puzzle games all have very negative values.

Part b

1. After implementing the two-level cross-validation we decided on the appropriate range of values for λ by applying the theory given in class, where we should look for a minimum in the test error. The procedure is like what was done for question 2 of part a. By plotting the different test errors depending on the value of lambda, we were looking for a plot where the error would be steady at the beginning and later dip towards a minimum and then increase again. The iterations of the plot from Figure 1 above were made using various lambda ranges to get a range that was able to capture the fall and rise of the error as the lambda value increased. The optimal range of values for λ would be about 2^{-5} to 2^{7}.

   For the ANN we have chosen to use ranges from 1 (as required by the guideline) up to 100. In the first tries, we started with a range of 1 to 10 hidden units but the optimal value for *h* was always the value of the upper bound. After many tries, we realized that the optimal number of hidden neurons is somewhere in the neighborhood of 25 to 50 hidden layers. All the tests were done with a ceiling of 10000 iterations, which logically were fulfilled for the higher numbers of hidden units. We also saw that when many neurons were used (e.g: 1000) the results were somehow non-coherent, where we would get large errors on the validation datasets.

2. Table 1 below shows the results of the two-level cross-validation for the three regression models.

Table 1: Summary of the three regression models and their errors

	Artificial Neural Network		***Linear Regression***		***Baseline***
Outer Fold	h	$\boldsymbol{E_{test}}$	λ	$\boldsymbol{E_{test}}$	$\boldsymbol{E_{test}}$
Outer fold 1	25	6267.30	16	21671.62	16194.54
Outer fold 2	25	5188.37	64	18212.24	22243.48
Outer fold 3	25	6844.09	16	18753.59	22544.70
Outer fold 4	50	7210.58	32	18676.96	22418.40
Outer fold 5	25	7616.63	32	16692.11	18468.90

Outer fold 6	25	6256.59	32	11284.60	14470.89
Outer fold 7	25	5167.68	32	5242.42	6712.46
Outer fold 8	50	7519.58	16	4688.76	4318.27
Outer fold 9	25	6202.26	16	4638.68	2157.54
Outer fold 10	50	5401.61	8	8581.06	670.20
Mean Error		**6367.315**		**12844.2**	**13019.94**

At a glance, the results in this table show that the optimal number of hidden units and optimal lambda values are relatively constant throughout the 10 folds. The square error per observation also suggests that the ANN model has the lowest error, while the next lowest is the linear regression, followed by the baseline. The errors for the linear regression and baseline fluctuate a bit, but this is likely just due to the distribution of data within those specific folds.

From question 2 of part a in the regression, the optimal lambda was found to be 32. When looking at the optimal lambda values for each of the folds in Table 1 above, we can see that the most common value is 32. It is by no means a majority, but it can be seen that when the optimal lambda value does deviate, it is only by one power of 2 except for the last fold with an optimal lambda of 2^3, or 8. The slight deviation of one power of two can simply be attributed to the distribution of data in that specific fold, and overall isn't too big of a difference from 32. Taking the average of the lambda values also gives 26.4, which is quite close to 32. Hence, when using two-level cross-validation, the main optimal lambda value that is outputted matches the optimal lambda value is that found when using single-level cross-validation as was done in question 2 of part a in the regression.

3. Setup I, the paired t-test was used to compare the three models pairwise. The results are as follows:

   Linear regression to ANN

   z = mean(zA – zB) = 391.72

   Confidence interval = (-541.40, 1325.15)

   p-value = 0.41

   From the z value, it is quite large and positive, which would imply that the second model, ANN, is better than linear regression because it has a lower average error. However, when looking at the confidence interval it does overlap with zero, and the p-value is quite large. This means that there is no evidence against the null hypothesis that they are the same, essentially saying that the two models are very similar. Hence, even though the z-value suggests that the ANN model is better, the p-value and confidence interval suggests that the two models are similar, so one model cannot be said to be better than the other. This is somewhat unexpected as intuition would expect that the ANN model would perform better than linear regression, but this result may just be due to the nature/distribution of the dataset. Keep in mind that the statistical results obtained are only for this specific dataset since setup I is being used.

Linear regression to baseline

z = mean(zA – zB) = -1575.00

Confidence interval = (-2530.87, -619.12)

p-value = 0.0013

The z value is very large and negative, which suggests that the first model, the linear regression, is better than the baseline model. The confidence interval doesn't overlap with zero, and the p-value obtained is relatively small. In this case, we do have evidence against the null hypothesis that the two models are similar, so we can say that the two models are statistically different. Hence, the linear regression model performs better than the baseline. This is expected since the baseline is meant to act as a poor model given that it will simply assign the same value to all data points.

ANN to baseline

z = mean(zA – zB) = -1966.72

Confidence interval = (-2575.18, -1358.26)

p-value = $2.90 \cdot 10^{-10}$

The z value is once again very large and negative, which suggests that the first model, ANN, is much better than the baseline model. The confidence interval has a smaller range than the previous comparison and still doesn't overlap with zero. Furthermore, the p-value is very small, so the null hypothesis can be rejected, and the two models are statistically different. This suggests that the ANN model is different and better than the baseline, which once again is expected due to the complex nature of the ANN model which should make better predictions compared to the baseline model which is predicting the same value every time.

Both the linear regression and ANN models are shown to be different and better than the baseline. It cannot be said whether the linear regression model is better than the ANN model because of the large p-value present in the comparison, suggesting that the two models are statistically identical. Based on these results, it is recommended not to use the baseline model since both the linear regression and ANN models are better. However, between the linear regression and ANN the performance appears to be similar for this specific dataset, and so choosing which model to use may come down to other factors such as the time it takes to run, where the linear regression model would be much faster.

Classification

1. For the classification problem, we are going to find the critic score of the game based on the remaining attributes. The remaining attributes include platform, year of release, genre, NA sales, EU sales, JP sales, other sales, critic score, user score and rating. We are going to use a binary classification problem to predict whether a game is above or below the average score value. From our data, around 45% of games are below average and 55% of games are above average. This makes for an effective training and test data set.
2. The method that was chosen to be *method 2* was KNN, k-nearest neighbors classifications. The KNN classifier is most easily influenced by the value of K, the number of nearest neighbors that are used to determine the class of the given point. As such, a few trial runs were run using two-level cross-validation with 10 folds in each level. The first run used values of N that ranged from 1 to 100 in increments of 10. The error values versus the number of nearest neighbors were plotted to see the trend of how the error varied according to the value of N. While there was a relatively large decrease in error between the first two data points (i.e. using 11 neighbors vs 1 neighbor), the error didn't appear to go up much after that as expected by the effects of overfitting. As such, the range was increased significantly to go from 1 to 800 in increments of 25. This produced a plot that more closely reassembled expected behavior; the error decreased, reached a minimum, and began increasing again. To refine the range more, the range was finalized to be from 1 to 500 in increments of 10, providing better resolution in the number of nearest neighbors while still showing the trend of the error and showing where the optimal number of neighbors lies. The final plot capturing the motion of the plot is shown in Figure 2 below. There is a decrease followed by a very gradual increase as the number of nearest neighbors increases.

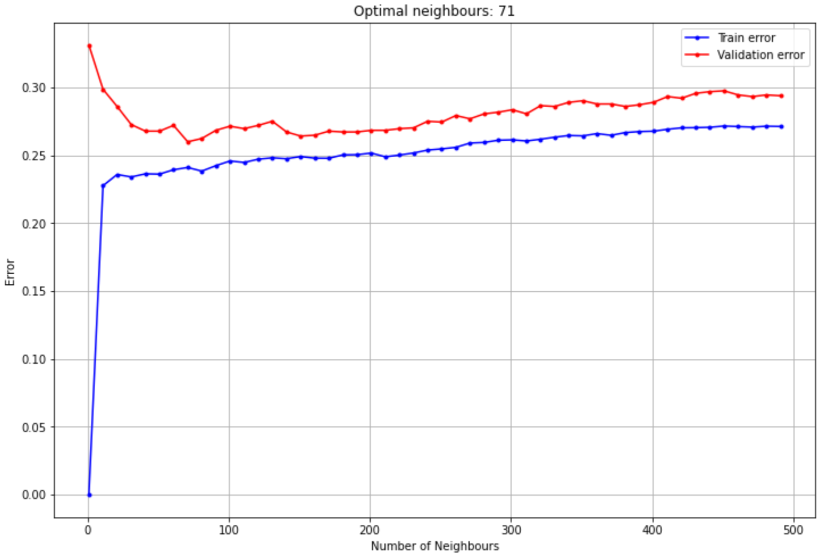

Figure 2: Prediction error versus the number of neighbors used

As stated in the problem statement, the regularization parameter will be used to optimize the logistic regression model, and the baseline model will simply find the largest class out of the training set for the current fold and predict everything to be belonging to that class.

The range of lambda values chosen for the logistic regression model was chosen similar to what was done to the KNN model and for linear regression. The lambda values were first chosen in powers of 10, looking for the characteristic drop and then a rise in error as the value of lambda increases. To further accentuate the effect, a smaller range of lambdas was chosen, and powers of 2 were used instead. After a few trial runs, a similar range of lambda values as was used for linear regression was set, being from 2^{-5} to 2^{8}. This range was able to capture the desired range of motion, where the minima can be seen clearly, indicating the optimal lambda value. Figure 3 in the Appendix shows this trend.

3. Note that the error measure which will be used is the error rate, or simply the number of misclassified observations over the total number of observations in the test split. The results of the two-level cross-validation are shown in Table 2 below.

Table 2: Summary of the three classification models and their errors

	K-Nearest Neighbor		***Logistic Regression***		***Baseline***
Outer Fold	N_h	E_{test}	λ	E_{test}	E_{test}
Outer fold 1	61	0.13	16	0.13	0.13
Outer fold 2	151	0.22	32	0.20	0.23
Outer fold 3	131	0.33	8	0.25	0.26
Outer fold 4	*91*	0.29	8	0.22	0.35
Outer fold 5	*151*	0.24	16	0.25	0.49
Outer fold 6	*71*	0.35	16	0.31	0.51
Outer fold 7	*121*	0.26	16	0.21	0.52
Outer fold 8	*71*	0.31	32	0.29	0.61
Outer fold 9	*71*	0.38	32	0.37	0.74
Outer fold 10	*71*	0.31	8	0.32	0.72
Mean Error		**0.282**		**0.234**	**0.456**

The table shows that the optimal number of nearest neighbors varies between 61 and 151. This is a relatively large range when considering that the steps were in increments of 10. The values of lambda for logistic regression were within the range of 8 to 32, or 2^3 to 2^5. This in contrast is a much more reasonable range since it is only two steps. In both cases, the variation in the optimal parameter value can be attributed to the distribution of the data in the specific fold. From the error values, the logistic regression and KNN models have a similar mean error, while the baseline model has a much higher error rate. This is expected as the two computed models should have better performance than the baseline. One interesting aspect is that the error for the baseline gets worse with the fold number, and this may simply just be because of the division between test and training data for fold 10, the testing data had most of its data in one class while the test data had most in the other class, hence producing a high error rate. The inverse is likely true for the first fold, producing a low error rate. In other words, the data is simply distributed a bit strangely, but given that the mean error is close to 50%, this matches the expected error rate for the baseline model.

4. The test that has been chosen is setup I, the McNemera's test. The models are compared pairwise as follows:

   Logistic regression to KNN:

   Confidence interval = [θ_L, θ_U] = (0.012, 0.044)

   p-value = 0.00078

   $\theta = \theta_A - \theta_B = 0.028$

   We first compare logistic regression to KNN. We can see that theta is positive. As logistic regression is our "A" model, this shows us that our logistic regression model is a more effective predictor than our KNN model. Additionally, the confidence interval for theta doesn't include zero and does include theta. This, along with the fact that our p-value is significantly small, means that we can reject the null hypothesis that the two models are identical. In other words, the two models are statistically different from each other, and their performance can be compared.

   Logistic regression to baseline:

   Confidence interval = [θ_L, θ_U] = (0.17, 0.23)

   p-value = $1.09 \cdot 10^{-43}$

   $\theta = \theta_A - \theta_B = 0.20$

   Comparing our logistic regression model to the baseline model, it is no surprise that the logistic regression is much better than the baseline. The p-value is once again very small, with the confidence interval for theta being way above zero. This means that we can again reject the null hypothesis that the two models are identical.

   KNN to baseline:

   Confidence interval = [θ_L, θ_U] = (0.14, 0.20)

   p-value = $3.26 \cdot 10^{-30}$

   $\theta = \theta_A - \theta_B = 0.17$

Comparing our logistic KNN model to the baseline, we see the same thing we saw when comparing our logistic regression. The theta value is positive and is significantly enough away from 0, suggesting the KNN model is superior to the baseline. The p-value is once again very small, with the confidence interval for theta not including zero. This means that we can again reject the null hypothesis that the two models are identical.

Based on our findings, we would recommend never using the baseline as a model as both the logistic regression and ANN models were shown to be statistically better than the baseline. Furthermore, we would recommend using logistic regression rather than KNN as the statistical values show that logistic regression has better performance. However, that is not to say that KNN is a bad model, as KNN is still miles better than the baseline. All three of our models could be compared with each other without issues and none of the p-values were high enough to indicate that any two models were similar to each other.

5. The logistic regression model uses a similar function to the regression model. This time however, the logistic function of the sum of the products of the weights and their corresponding values is calculated. This looks as follows:

$$y_n = logistic(w_0 + \sum_k w_k x_{n,k}) + \epsilon_n$$

The weights for the logistic regression model can be found in the third column of Table 3, which can be found in the Appendix. Comparing the weights to those from the previous exercise it is unsurprising that they don't match. The biggest standout is, unsurprisingly, NA sales. While the NA sales were negatively weighted when predicting the Japanese sales, it is one of the highest weighted positive weights for the score of the games. This of course makes sense as highly rated games tend to sell more copies. Another big standout is the user score. The user score is by far the biggest weight, being nearly 3 times as big as the second biggest weight.

Looking at the negative weights, PlayStation and the PSV have the worst weights in terms of the critic score. This is very unlike the sales, where they both had quite high weights.

Overall, we can conclude that our classification and regression largely value different qualities. This is interesting, as you’d expect the high-selling games to have higher critic scores. However, it is possible that Japan doesn’t weigh critic scores as highly as the rest of the world, leading to the large difference. It could be that the critics were from more western countries and that Japanese critic scores would have aligned more with the sales.

Discussion

In the regression part, we tried to predict the sales of the region of Japan based on all the other attributes such as scores and sales in other countries. Since many videogame sales did not reach over 100.000 sales in that region, we had a large group of data points that once we deleted outliers by value, our dataset was considerably reduced. After a group discussion, we decided that we could have also dropped other regions to predict the sales in that specific region, preventing the loss of a large number of data points.

After creating a baseline model, an ANN and a linear regression model, we observed easily that the last two mentioned models outperformed the baseline model. It is true that punctually the generalized error for the linear regression model was lower than the one for the ANN, but it is important to mention that this second one would perform in a more consistent fashion. We do still believe that the ANN could find complex relationships between attributes that could not be found by other models, explaining why it performed more consistently.

For the classification part, we tried to predict the score of the different games given by the critics. Originally the attribute was on a scale of zero to a hundred. Since the mean was quite a high number (70/100) we decided to turn it into a binary problem, where any critic score above the mean would be considered as a good score or bad if it would fall under the mean. In this section, we also believed that the KNN would outperform the logistic regression, which after analyzing the results did not happen.

It is visible that using the most common class approach for the baseline model, due to the way we encoded the variable we wanted to predict, it would be a trivial task to outperform it. The great results obtained on the classification models could be explained since we used attributes that have a strong link with the goal attribute.

In the following paper [1] we find a regression problem solved by Alice Yufa et. al. who estimated the global sales by using all attributes provided by the dataset except the ones involving other sales since, as we stated before, they are intrinsic of the global sales attribute. They obtained the following formula to predict the global sales of a videogame:

Global sales predicted = -1.225 + 0.023 (Critic_Count) + 0.021 (Genre) + 0.025 (Critic_score) - 0.082 (User_Score)

Concluding that the Genre and Critic score has a positive impact but also mentioning the uncertainty of the reason why the user score has a negative impact, as we also stated in the previous report 1.

Problems

Problem 1

Option C: If we set the threshold after the furthest left data point, computing the FPR and TPR of predictions B and D, we can see that the point is not on the ROC curve, abling us to discard them. Looking at the next data point going from left to right, we can compute that for prediction A the TPR is ½ and the FPR is 1 which is also not on the ROC curve. By discarding, we can say that Prediction C is the correct one.

Problem 2

Option C: To solve this exercise we will compute the impurity gain for x_7 = 2 using the classification error impurity method.

$$\Delta = 1 - max\, p(y = c|v) - \frac{N(x_7 = 2)}{N}(1 - max\, p(y = c\,|x_7 = 2\,)) - \frac{N(x_7 \neq 2)}{N}(1 - max\, p(y = c\,|x_7 \neq 2\,))$$

$$\Delta = 1 - \frac{37}{135} - \frac{1}{135}(1 - 1) - \frac{134}{135}(1 - \frac{37}{135}) = 1 - \frac{37}{135} - \frac{134}{135}(1 - \frac{37}{135}) = 0.0\underline{074}$$

Problem 3

Option C: Since we have a NN with only one hidden layer, the total amount of weights can be divided into two sets. The first set will be the weights that go from the input layer (size = 7) to the hidden layer (size = 10), by multiplying the number of nodes we can state that the first "set" will have 70 weights. Moving on to the second set, we will go from a 10 nodes hidden layer to the output layer, which has 4 nodes since there are four possible outputs, creating a total amount of 40 weights. In total, our NN will have 70+40 = 110 weights.

Problem 5

Option C: To compute the total time to create the table, we have to take into account that the size of the outer fold is 5, the size of the inner fold is 4 and there are 5 different parameters that we want to test out. The training and testing time will be used twice in the equation since we do it for the inner and outer folds. The time will be computed as follows:

$$Total\ time = 5 * (4 * 5 * (20 + 5) + 25) + 5 * (4 * 5 * (8 + 1) + 9) = 3570\, ms$$

References

1. Yufa, A. (n.d.). *Predicting global video-game sales - quest journals*. Retrieved November 10, 2022, from https://www.questjournals.org/jrbm/papers/vol7-issue3/I07036064.pdf

Appendix

Figure 3 shows a plot of the prediction error versus the lambda value for the logistic regression model. The range of lambda used is the final range chosen to optimally show the behavior of the error.

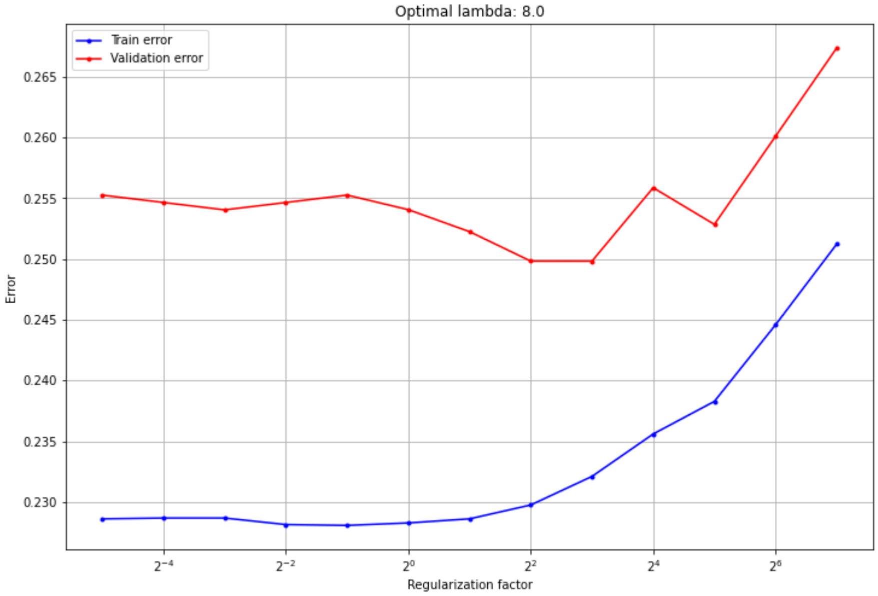

Figure 3: Prediction error versus the value of lambda

Table 3 below shows the coefficients that were obtained for the optimal linear and logistic regression models.

Table 3: Coefficients for the linear and logisitic regression models

Attribute Name	**Linear Regression weight**	**Logistic Regression weight**
Offset	114.19	N/A
Year of Release	-12.7	0.111
NA Sales	1.52	0.412
EU Sales	14.7	0.413
JP Sales	N/A	-0.060
Other Sales	5.8	0.108
Critic Score	-4.1	N/A
User Score	12.41	1.209
AO	0.0	0.0
E	17.2	-0.046
E10+	-3.82	0.067

K-A	0.0	0.0
M	-6.6	0.045
RP	0.0	0.0
T	-6.78	-0.067
3DS	51.3	-0.015
DC	21.52	0.284
DS	13.99	0.205
Gameboy Advanced	25.05	0.215
GameCube	22.88	-0.150
PC	-1.89	0.002
PS	16.96	0.012
PS2	12.2	-0.439
PS3	-4.79	0.059
PS4	-13.41	0.079
PSP	-8.64	-0.055
PSV	3.53	-0.075
Wii	-19.55	-0.478
WiiU	3.6	-0.088
Xbox 360	-68.88	0.239
Xbox	-29.5	0.086
Xbox One	-24.38	0.119
Action	14.23	-0.235
Adventure	6.64	-0.071
Fighting	28.6	0.145
Miscellaneous	8.56	0.081
Platform	-19.99	-0.131
Puzzle	-10.15	-0.036
Racing	-39.49	0.140
Role-Playing	56.96	-0.033
Shooter	-16.98	-0.007
Simulation	-0.28	-0.188
Sports	-26.49	0.161

Strategy	-1.61	0.174